SIEMENS

SINUMERIK

SINUMERIK 840D sl/828D
基本機能

機能マニュアル

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
6FC5397-0BP40-6TA2

まえがき
基本的な安全に関する指示事項 1
A2: NC/PLC 間の各種共通インタフェース信号と各種機能 2
A3:軸監視機能 3
A5:プロテクションゾーン 4
B1: 連続軌跡モード、イグザクトストップ、先読み: 5
B2:加速度 6
F1:突き当て点停止 7
G2: 速度、指令値/フィードバック回路、閉ループ制御 8
H2: PLC への補助機能出力 9
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 10
K2: 軸タイプ、座標系、フレーム 11
N2: 非常停止 12
P1:径方向軸 13
P3: SINUMERIK 840D sl 用 PLC 基本プログラム 14
P4 : SINUMERIK 828D 用 PLC 15
R1:レファレンス点復帰 16
S1:主軸 17
V1:送り速度 18
W1:工具補正 19
Z1:NC/PLC インタフェース信号 20
付録 A

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: 6FC5397-0BP40-6TA2
℗ 08/2018 変更する権利を留保

Copyright © Siemens AG 1994 - 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。

- プロジェクトエンジニア
- 工作機械メーカの技術者
- セットアップエンジニア(システム/機械)
- プログラマ

本書の目的

本書は、対象者が機能を理解して選択できるようにするために、各機能について説明しています。 対象者には、各機能の実行に必要な情報が提供されます。

標準仕様

この取扱説明書には、標準仕様の機能についてのみ記載されています。 工作機械メーカが実施した拡張または変更については、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、コントローラ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コントローラ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、簡略化のため、本説明書では製品のすべてのタイプの詳細を記載していません。また、取り付け、運転および保守において想定されるすべてのケースを網羅したものではありません。

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。

構成と内容に関する情報

構成

この機能マニュアルは、以下のように構成されています。

- 機能マニュアルという表題、機能マニュアルのこの版が適用されるソフトウェアとバージョンおよび SINUMERIK コントローラ、ならびに個々の機能説明の一覧が記述された内部の見出しページ(3 ページ)
- アルファベット順(A2、A3、B1 などの略号順)の機能説明
- 付録:
  - 略語の一覧
  - 本書の概要
- 用語のインデックス

注記

データとアラームの詳細説明については、以下を参照してください。

- マシンデータとセッティングデータ
  マシンデータの詳細説明
- NC/PLC インタフェース信号
  『NC 変数とインタフェース信号』リストマニュアル
- アラーム:
  『診断マニュアル』

システムデータの表記法

本書では、システムデータに対して以下の表記法をが適用されます。

信号/データ	表記法	例
NC/PLC インタフェース信号	...NC/PLC インタフェース信号: <信号アドレス> (<信号名称>)	新しいギヤが入ると、以下の NC/PLC インタフェース信号が PLC プログラムによって設定されます。 DB31、... DBX16.0-2(実際のギヤ選択 A ～ C) DB31、... DBX16.3 (ギヤ切替済)
マシンデータ	...マシンデータ: <タイプ><番号> <識別子全体> (<意味>)	メイン主軸は、下記のマシンデータに設定された主軸です。 MD20090 $MC_SPIND_DEF_MASTER_SPIND (チャネルのメイン主軸の初期設定)
セッティングデータ	...セッティングデータ: <タイプ><番号> <識別子全体> (<意味>)	論理メイン主軸は、下記のセッティングデータに含まれます。 SD42800 $SC_SPIND_ASSIGN_TAB[0] (主軸再割り当て)

注記

信号アドレス

機能説明書に NC/PLC インタフェース信号の<信号アドレス>として記載されているのは、SINUMERIK 840D sl に有効なアドレスだけです。SINUMERIK 828D の信号アドレスは、特定の機能マニュアルの最後にあるデータリスト「送信/受信信号」から入手してください。

構造の大きさ

NC/PLC インターフェースに関する説明は、以下のコンポーネントについて考えられる最大数に基づいています。

- モードグループ(DB11)
- チャネル(DB21 など)
- 軸/主軸(DB31 など)

データタイプ

制御装置は、パートプログラムでのプログラミングに使用できる次のデータタイプを提供します。

タイプ	意味	値の範囲
INT	符号付き整数	-2,147,483,648 ... +2,147,483,647
REAL	小数点付き数値	≈ ±5.0*10^{-324} … ≈ ±1.7*10^{+308}
BOOL	ブール値	TRUE (≠0)、FALSE (0)
CHAR	ASCII 文字およびバイト	0 ... 255 または-128 ... 127
STRING	文字列、NULL で終了	最大 400 文字の文字+ /0 (特殊文字は使用できません)
AXIS	軸名称	コントロールシステムに存在するすべての軸名称
FRAME	座標系の平行移動、座標回転、スケーリングおよびミラーリングのための幾何学的なパラメータ	---

配列

配列は、同種の基本データタイプからのみ形成できます。最大 3 次元の配列が使用可能です。

例:`DEF INT ARRAY[2, 3, 4]`

番号システム

次の番号システムが使用可能です。

- 10 進数: `DEF INT number = 1234` or `DEF REAL number = 1234.56`
- 16 進数: `DEF INT number = 'H123ABC'`
- 2 進法: `DEF INT number = 'B10001010010'`

REAL 変数の問い合わせ

NC プログラムとシンクロナイズドアクションでの REAL 変数または DOUBLE 変数の問い合わせを、制限値の評価としてプログラムすることを推奨します。

例:特定の値の軸の現在値の問い合わせ

プログラムコード	コメント
`DEF REAL AXPOS = 123.456`	
`IF ($VA_IM[<軸>] - 1ex-6) <= AXPOS <= ($VA_IM[<軸>] + 1ex-6)`	`; 現在位置`
`...`	`== AXPOS`
`ELSE`	

プログラムコード	コメント
...	<> AXPOS
ENDIF	

目次

まえがき 3

1 基本的な安全に関する指示事項 41

1.1 一般的な安全に関する指示事項 41

1.2 アプリケーション例に対する保証と責任 41

1.3 産業セキュリティ 42

2 A2: NC/PLC 間の各種共通インタフェース信号と各種機能 45

2.1 概略説明 45

2.2 NC/PLC インタフェース信号 45
2.2.1 一般情報 45
2.2.2 PLC への準備完了信号 47
2.2.3 PLC への状態信号 48
2.2.4 操作パネルへ/からの信号 51
2.2.5 チャネルへの信号 54
2.2.6 軸/主軸への信号 54
2.2.7 軸/主軸からの信号 65
2.2.8 軸/主軸への信号(ディジタルドライブ) 66
2.2.9 軸/主軸からの信号(ディジタルドライブ) 68

2.3 各種機能 70
2.3.1 過負荷の場合の画面の更新動作-840D sl のみ 70
2.3.2 インボリュート補間の設定- 840D sl のみ 71
2.3.3 初期設定メモリの起動- 840D sl のみ 74
2.3.4 PLC 変数の読み取りおよび書き込み- 840D sl のみ 74
2.3.5 パスワードおよびキースイッチを介したアクセス保護 78
2.3.5.1 パスワード 80
2.3.5.2 キースイッチ位置(DB10、DBX56.4 ～ 7) 82
2.3.5.3 パラメータ設定可能な保護レベル 83
2.3.6 モータ/ドライブデータセットの切り替え 84
2.3.6.1 一般事項 84
2.3.6.2 フォーマットインターフェース 85
2.3.6.3 要求インターフェース 86
2.3.6.4 表示インタフェース 86
2.3.6.5 例 87
2.3.6.6 インターフェースの概要 88
2.3.6.7 必要条件 89

2.4 例 90
2.4.1 パラメータセットの切り替え 90

2.5 データリスト....93
2.5.1 マシンデータ....93
2.5.1.1 表示マシンデータ....93
2.5.1.2 NC 別マシンデータ....93
2.5.1.3 チャネルマシンデータ....93
2.5.1.4 軸/主軸マシンデータ....94
2.5.2 システム変数....95
2.5.3 信号....95
2.5.3.1 NC への信号....95
2.5.3.2 NC からの信号....95
2.5.3.3 操作パネルへの信号....96
2.5.3.4 操作パネルからの信号....97
2.5.3.5 チャネルへの信号....97
2.5.3.6 チャネルからの信号....97
2.5.3.7 軸/主軸への信号....98
2.5.3.8 軸/主軸からの信号....98

3 A3:軸監視機能....101

3.1 輪郭監視....101
3.1.1 輪郭誤差....101
3.1.2 追従誤差の監視....102

3.2 位置決め監視、停止状態監視およびクランプ監視....104
3.2.1 位置決め監視、停止状態監視およびクランプ監視の関係....104
3.2.2 位置決め監視....104
3.2.3 停止状態監視....106
3.2.4 パラメータセットに依存するイグザクトストップと停止許容移動範囲....107
3.2.5 クランプ監視....107
3.2.5.1 機能....107
3.2.5.2 マシンデータ....108
3.2.5.3 NC/PLC インタフェース信号....109
3.2.5.4 故障応答....110
3.2.5.5 「クランプを解除するための自動停止」クランプ機能....110
3.2.5.6 「時間を優先したクランプの解除」クランプ機能....112
3.2.5.7 「クランプを設定するための自動停止」クランプ機能....113
3.2.5.8 必要条件....115

3.3 速度指令監視....118

3.4 フィードバック速度監視....119

3.5 検出器監視....120
3.5.1 エンコーダ限界周波数監視....123
3.5.2 絶対値エンコーダの妥当性チェック....124
3.5.3 カスタマイズされたアラーム応答....127

3.6 リミットスイッチ監視....130

3.6.1 ハードウェアリミットスイッチ....130
3.6.2 ソフトウェアリミットスイッチ....131
3.7 作業領域リミット監視....134
3.7.1 全般....134
3.7.2 BCS におけるワーキングエリアリミット....136
3.7.3 WCS/SZS における作業領域リミット....139
3.7.4 例:WCS/SZS における作業領域リミット....142
3.8 機械軸のパーキング....145
3.9 パッシブな位置検出器のパーキング....147
3.9.1 機能....147
3.9.2 必要条件....152
3.9.3 例:直接位置検出器の装着ヘッドの交換....152
3.9.4 例:2 台の直接位置検出器の装着ヘッドの交換....157
3.9.5 例:範囲の一部にエンコーダがない場合の検出器の切り替え....162
3.10 エンコーダデータセットの切替え....165
3.11 データリスト....169
3.11.1 マシンデータ....169
3.11.1.1 NC 別マシンデータ....169
3.11.1.2 チャネルマシンデータ....169
3.11.1.3 軸/主軸マシンデータ....170
3.11.2 セッティングデータ....172
3.11.2.1 軸/主軸セッティングデータ....172
3.11.3 信号....172
3.11.3.1 軸/主軸への信号....172
3.11.3.2 軸/主軸からの信号....173
4 A5:プロテクションゾーン....175
4.1 機能....175
4.2 セットアップ....183
4.2.1 マシンデータ....183
4.3 プログラミング....184
4.3.1 プロテクションゾーンの定義(CPROTDEF、NPROTDEF)....184
4.3.2 プロテクションゾーンの起動/解除(CPROT、NPROT)....188
4.3.3 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)....193
4.4 特別な事例....207
4.4.1 プロテクションゾーンの一時移動許可....207
4.4.2 AUTOMATIC と MDI モードでの動作....207
4.4.3 JOG モードでの動作....208
4.5 境界条件....212
4.6 例....213

4.6.1 旋盤のプロテクションゾーン...213
4.6.2 パートプログラムでのプロテクションゾーンの定義...214
4.6.3 システム変数によるプロテクションゾーンの定義...214
4.6.4 プロテクションゾーンの起動:...224

4.7 データリスト...225
4.7.1 マシンデータ...225
4.7.1.1 NC 別マシンデータ...225
4.7.1.2 チャネルマシンデータ...225
4.7.2 信号...225
4.7.2.1 チャネルへの信号...225
4.7.2.2 チャネルからの信号...226

5 B1: 連続軌跡モード、イグザクトストップ、先読み:...227

5.1 概略説明...227

5.2 イグザクトストップモード...230

5.3 連続軌跡モード...235
5.3.1 共通動作...235
5.3.2 過負荷係数による速度低減...237
5.3.3 丸み付け...240
5.3.3.1 軌跡条件に応じた丸み付け(G641)...242
5.3.3.2 定義した許容範囲に適合する丸み付け(G642/G643)...245
5.3.3.3 使用可能な最大軸ダイナミック応答による丸み付け(G644)...249
5.3.3.4 接線の変化が連続的なブロック遷移部の丸み付け(G645)...252
5.3.3.5 丸み付けと再位置決め(REPOS)...254
5.3.4 先読み...255
5.3.4.1 標準機能...255
5.3.4.2 自由形式面モード: 拡張機能...261

5.4 ダイナミック補正...265
5.4.1 軌跡速度のスムージング...265
5.4.2 ダイナミック軌跡応答の補正...269
5.4.3 ダイナミック応答制限値の決定...274
5.4.4 「軌跡速度のスムージング」および「軌跡のダイナミック応答の補正」機能の連携...275
5.4.5 軌跡補間のダイナミック応答モード...279
5.4.6 自由形式面モード:基本機能...281

5.5 コンプレッサ機能...285
5.5.1 直線ブロック、円弧ブロック、および早送りブロックの圧縮...285
5.5.1.1 機能...285
5.5.1.2 セットアップ...288
5.5.1.3 プログラミング...291
5.5.1.4 必要条件...292
5.5.2 短いスプラインブロックの圧縮...292

5.6 輪郭/向きの許容範囲...294

5.6.1 セットアップ....294
5.6.1.1 パラメータの割り付け....294
5.6.2 プログラミング....295
5.6.2.1 輪郭/旋回の許容範囲のプログラミング(CTOL、OTOL、ATOL)....295
5.6.2.2 輪郭/旋回の許容範囲のプログラミング(CTOL、OTOL、ATOL)の追加情報....298

5.7 早送り移動....300
5.7.1 機能....300
5.7.1.1 早送り....300
5.7.1.2 早送り移動の軌跡軸の補間動作....301
5.7.1.3 早送り移動の許容範囲....303
5.7.1.4 早送りオーバライド....303
5.7.2 セットアップ....304
5.7.2.1 パラメータの割り付け....304
5.7.3 プログラミング....305
5.7.3.1 早送り(G0)の有効化....305
5.7.3.2 早送り移動の直線補間のオン/オフを切り替えます(RTLION、RTLIOF)....306
5.7.3.3 早送り移動(STOLF)の許容範囲係数を調整します....308

5.8 リセット動作....310

5.9 必要条件....311
5.9.1 ブロック変更と位置決め軸....311
5.9.2 ブロック変更の遅延....311

5.10 データリスト....312
5.10.1 マシンデータ....312
5.10.1.1 一般マシンデータ....312
5.10.1.2 チャネルマシンデータ....312
5.10.1.3 軸/主軸別のマシンデータ....314
5.10.2 セッティングデータ....315
5.10.2.1 チャネルセッティングデータ....315
5.10.3 信号....316
5.10.3.1 チャネルからの信号....316
5.10.3.2 軸/主軸からの信号....316

6 B2:加速度....317

6.1 概略説明....317
6.1.1 一般情報....317
6.1.2 特徴....317

6.2 機能....319
6.2.1 加々速度制限のない加減速(BRISK/BRISKA) (チャネル/軸別)....319
6.2.1.1 一般事項....319
6.2.1.2 パラメータ設定....321
6.2.1.3 プログラミング....322
6.2.2 定速移動区間(チャネル別)....323
6.2.2.1 一般事項....323

6.2.2.2 パラメータ設定...325
6.2.3 加速度調整(ACC) (軸別)...325
6.2.3.1 一般事項...325
6.2.3.2 プログラミング...325
6.2.4 加速マージン(チャネル別)...326
6.2.4.1 一般事項...326
6.2.4.2 パラメータ設定...327
6.2.5 軌跡加速度制限(チャネル別)...327
6.2.5.1 一般事項...327
6.2.5.2 パラメータ設定...327
6.2.5.3 プログラミング...327
6.2.6 リアルタイムイベントの軌跡加速度(チャネル別)...329
6.2.6.1 一般事項...329
6.2.6.2 プログラミング...330
6.2.7 早送り(G00)がプログラムされたのときの加速度(軸別)...331
6.2.7.1 一般事項...331
6.2.7.2 パラメータ設定...331
6.2.8 加々速度制限が有効な加速度(SOFT/SOFTA) (軸別)...332
6.2.8.1 一般事項...332
6.2.8.2 パラメータ設定...332
6.2.9 接線の変化が不連続なブロック遷移の過剰加速度(軸別)...333
6.2.9.1 一般事項...333
6.2.9.2 パラメータ設定...333
6.2.10 法線方向加速度に対応した加速マージン(チャネル別)...334
6.2.10.1 一般事項...334
6.2.10.2 パラメータ設定...335
6.2.11 軌跡補間のときの加々速度制限(SOFT) (チャネル別)...336
6.2.11.1 一般事項...336
6.2.11.2 パラメータ設定...338
6.2.11.3 プログラミング...339
6.2.12 単独軸補間の加々速度制限(SOFTA) (軸別)...339
6.2.12.1 パラメータ設定...339
6.2.12.2 プログラミング...340
6.2.13 軌跡加々速度制限(チャネル別)...341
6.2.13.1 一般事項...341
6.2.13.2 パラメータ設定...341
6.2.13.3 プログラミング...342
6.2.14 リアルタイムイベントの軌跡加々速度(チャネル別)...343
6.2.14.1 一般事項...343
6.2.14.2 プログラミング...344
6.2.15 早送り(G00)がプログラムされたのときの加々速度(軸別)...345
6.2.15.1 一般事項...345
6.2.15.2 パラメータ設定...346
6.2.16 一定した曲率を持たないブロック遷移に対する過剰加々速度(軸別)...346
6.2.16.1 一般事項...346

6.2.16.2 パラメータ設定...346
6.2.17 速度依存加々速度調整(軸別)...347
6.2.18 加々速度フィルタ(軸別)...349
6.2.18.1 一般事項...349
6.2.18.2 パラメータ設定...352
6.2.19 膝形加減速特性曲線...353
6.2.19.1 機能:モータ特性曲線の補正...353
6.2.19.2 機能:軌跡加速度に対する影響...353
6.2.19.3 機能:代替曲線...354
6.2.19.4 パラメータ設定...357
6.2.19.5 プログラミングチャネル別の起動(DRIVE)...360
6.2.19.6 プログラミング軸別の起動(DRIVEA)...361
6.2.19.7 境界条件...361
6.2.20 JOG 移動の加速度と加々速度...362
6.2.20.1 パラメータ設定...362
6.2.20.2 必要条件...364

6.3 例...365
6.3.1 加速度...365
6.3.1.1 軌跡速度特性...365
6.3.2 加々速度...367
6.3.2.1 軌跡速度特性...367
6.3.3 加速度と加々速度...368
6.3.4 膝形加速特性曲線...370
6.3.4.1 起動...370

6.4 データリスト...371
6.4.1 マシンデータ...371
6.4.1.1 NC 別マシンデータ...371
6.4.1.2 チャネルマシンデータ...371
6.4.1.3 軸/主軸マシンデータ...372
6.4.2 セッティングデータ...373
6.4.2.1 チャネルセッティングデータ...373
6.4.3 システム変数...373

7 F1:突き当て点停止...375

7.1 概略説明...375

7.2 詳細説明...376
7.2.1 プログラミング...376
7.2.2 機能の処理...379
7.2.2.1 選択...379
7.2.2.2 突き当て点に達しました。...380
7.2.2.3 突き当て点に未達...383
7.2.2.4 解除...384
7.2.3 ブロック検索時の動作...387
7.2.4 リセットと機能の中止の動作...390

7.2.5 その他の機能に関する動作...392
7.2.6 セッティングデータ...392
7.2.7 システム変数...394
7.2.8 アラーム...395
7.2.9 トルク/推力を制限した移動(FOC)...397
7.3 例...400
7.4 データリスト...402
7.4.1 マシンデータ...402
7.4.1.1 軸/主軸マシンデータ...402
7.4.2 セッティングデータ...403
7.4.2.1 軸/主軸セッティングデータ...403
7.4.3 信号...403
7.4.3.1 軸/主軸への信号...403
7.4.3.2 軸/主軸からの信号...403
8 G2: 速度、指令値/フィードバック回路、閉ループ制御...405
8.1 概略説明...405
8.2 速度、移動範囲、精度...405
8.2.1 速度...405
8.2.2 移動範囲...407
8.2.3 コントロールシステムの位置決め精度...408
8.2.4 最小入力/表示単位、計算分解能...409
8.2.5 マシンデータとセッティングデータの物理量の単位変換...411
8.3 単位系、メトリック/インチ...414
8.3.1 機能...414
8.3.1.1 パラメータ設定およびプログラムされた単位系...414
8.3.1.2 拡張単位系機能...416
8.3.1.3 ユーザーインタフェースでの単位系の切り替え...416
8.3.2 セットアップ...419
8.3.3 プログラミング...423
8.3.3.1 単位系の切り替え(G70/G71/G700/G710)...423
8.4 指令/フィードバック回路...428
8.4.1 一般情報...428
8.4.2 指令値とエンコーダの割り当て...431
8.4.3 モータ/負荷側減速比の設定...436
8.4.4 速度指令値出力...439
8.4.5 フィードバック回路のマシンデータ...441
8.4.6 フィードバック値の分解能...442
8.4.6.1 フィードバック値の分解能のマシンデータ...442
8.4.6.2 例: リニアスケールを使用した直線軸...446
8.4.6.3 例: モータ側にロータリエンコーダを使用した直線軸...446
8.4.6.4 例: 機械側にロータリエンコーダを使用した直線軸...448
8.4.6.5 例: モータ側にロータリエンコーダを使用した回転軸...449

8.4.6.6 例: 機械側にロータリエンコーダを使用した回転軸 ..450
8.4.6.7 例: エンコーダが工具に接続されている中間ギヤ ..451

8.5 閉ループ制御 ..452
8.5.1 一般情報 ..452
8.5.2 位置コントローラのパラメータセット ..455

8.6 制御の調整 ..457
8.6.1 位置コントローラ、位置指令フィルタ:バランスフィルタ ..457
8.6.2 位置コントローラ、位置指令フィルタ: 加々速度フィルタ ..462
8.6.3 位置コントローラ、位置指令フィルタ: 位相フィルタ ..464
8.6.4 位置コントローラ: 位置偏差の入力 ..466
8.6.5 比例積分動作コントローラによる位置制御 ..468

8.7 データリスト ..470
8.7.1 マシンデータ ..470
8.7.1.1 表示マシンデータ ..470
8.7.1.2 NC 別マシンデータ ..470
8.7.1.3 チャネルマシンデータ ..471
8.7.1.4 軸/主軸マシンデータ ..471
8.7.2 信号 ..474
8.7.2.1 NC からの信号 ..474
8.7.2.2 NC への信号 ..474

9 H2: PLC への補助機能出力 ..475

9.1 概略説明 ..475
9.1.1 機能 ..475
9.1.2 補助機能の定義 ..476
9.1.3 補助機能の概要 ..477

9.2 事前定義補助機能 ..483
9.2.1 一覧: 事前定義補助機能 ..484
9.2.2 概要:出力動作 ..499
9.2.3 パラメータ設定 ..504
9.2.3.1 グループ割り当て ..504
9.2.3.2 タイプ、アドレス拡張、値 ..505
9.2.3.3 出力動作 ..506

9.3 ユーザー定義補助機能 ..510
9.3.1 パラメータ設定 ..512
9.3.1.1 ユーザー定義補助機能の最大数 ..512
9.3.1.2 グループの割り当て ..512
9.3.1.3 タイプ、アドレス拡張、値 ..513
9.3.1.4 出力動作 ..514

9.4 関連補助機能 ..515

9.5 タイプ別の出力動作 ..517

9.6 パラメータに設定された出力動作の優先順位...519
9.7 補助機能のプログラミング...520
9.8 プログラム可能な出力時間...522
9.9 PLC への補助機能出力...523
9.10 ブロック変更遅延のない補助機能...524
9.11 自動先読み停止を伴う M 機能...525
9.12 オーバーストアへの応答...526
9.13 ブロック検索中の動作...527
9.13.1 タイプ 1、2、および 4 のブロック検索中の補助機能出力...527
9.13.2 複数のグループへの補助機能の割り当て...529
9.13.3 有効な M 補助機能のタイムスタンプ...530
9.13.4 出力順序の決定...531
9.13.5 主軸用補助機能の出力抑制...532
9.13.6 タイプ 5 のブロック検索(SERUPRO)による補助機能の出力...536
9.13.7 SERUPRO 終了時の ASUB...542
9.14 自動出力補助機能...549
9.15 情報表示手段...551
9.15.1 グループ別モーダル M 補助機能の表示...551
9.15.2 システム変数の確認...552
9.16 必要条件...555
9.16.1 一般制約...555
9.16.2 出力動作...556
9.17 例...558
9.17.1 事前定義補助機能の拡張...558
9.17.2 補助機能の定義...560
9.18 データリスト...564
9.18.1 マシンデータ...564
9.18.1.1 NC 別マシンデータ...564
9.18.1.2 チャネルマシンデータ...564
9.18.2 信号...565
9.18.2.1 チャネルへの信号...565
9.18.2.2 チャネルからの信号...566
9.18.2.3 軸/主軸への信号...568
9.18.2.4 軸/主軸からの信号...568
10 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...569
10.1 概略説明...569
10.2 モードグループ...572
10.2.1 モードグループ停止...574

10.2.2 モードグループリセット..575
10.3 モードタイプおよびモードタイプ変更..576
10.3.1 各モードの監視機能およびインタロック..582
10.3.2 モード変更..583
10.4 チャネル..584
10.4.1 起動禁止、グローバルおよびチャネル別..588
10.5 プログラムテスト..589
10.5.1 指令値の出力のないプログラム実行..590
10.5.2 シングルブロックモードでのプログラムの実行..591
10.5.3 ドライラン送り速度でのプログラムの実行..594
10.5.4 パートプログラムブロックのスキップ..597
10.6 ワークシミュレーション..598
10.7 ブロック検索、タイプ 1、2、および 4:..599
10.7.1 機能の説明..601
10.7.2 ブロックサーチの他の NCK 機能への関係..603
10.7.2.1 ブロックサーチ中およびブロックサーチ後の ASUB..603
10.7.2.2 ブロックサーチ後の PLC 動作..605
10.7.2.3 ブロック検索後の主軸機能..605
10.7.2.4 ブロック検索のためのシステム変数の読み取り..607
10.7.3 ブロックサーチ後の ASUB の自動起動..607
10.7.4 多段ブロックサーチ..609
10.7.5 計算ありのブロック検索の例..611
10.7.6 必要条件..615
10.7.6.1 コンプレッサ機能(COMPON、COMPCURV、COMPCAD)..615
10.8 ブロック検索タイプ 5 (SERUPRO)..615
10.8.1 機能の説明..615
10.8.2 輪郭への再位置決め(REPOS)..619
10.8.2.1 制御された REPOS による再位置決め..629
10.8.3 ブロック検索の加速..630
10.8.4 SERUPRO ASUB..632
10.8.5 自動 SERUPRO..635
10.8.6 「輪郭での加工継続」のプログラム区間のロック..637
10.8.7 POWER ON、モード変更および RESET 中の動作..641
10.8.8 必要条件..641
10.8.8.1 ターゲットブロックの STOPRE..641
10.8.8.2 ターゲットブロックでの SPOS..642
10.8.8.3 突き当て点停止(FXS)..642
10.8.8.4 トルク/推力を制限した移動(FOC)..643
10.8.8.5 主軸同期..644
10.8.8.6 連結とマスタスレーブ..644
10.8.8.7 軸機能..648
10.8.8.8 ギヤ選択切替..650

10.8.8.9 重畳動作....651
10.8.8.10 NC/PLC インタフェース信号....651
10.8.8.11 よりフレキシブルな初期設定の作成....652
10.8.8.12 コンプレッサ機能(COMPON、COMPCURV、COMPCAD)....652
10.8.9 システム変数....653
10.9 プログラム運転....653
10.9.1 初期設定....654
10.9.1.1 マシンデータ....655
10.9.1.2 プログラミング....656
10.9.2 NC プログラムの選択と起動....660
10.9.3 プログラムの中断....661
10.9.4 チャネルリセット....664
10.9.5 プログラム状態....665
10.9.6 チャネル状態....667
10.9.7 オペレータアクションおよびプログラムアクションへの応答....668
10.9.8 プログラム実行のタイミングチャートの例....669
10.9.9 プログラムジャンプ....670
10.9.9.1 プログラムの先頭への復帰ジャンプ(GOTOS)....670
10.9.10 プログラムセクションの繰り返し....673
10.9.10.1 プログラミング....673
10.9.11 イベント起動プログラム呼び出し(PROG_EVENT)....679
10.9.11.1 機能....679
10.9.11.2 パラメータ設定....683
10.9.11.3 プログラミング....689
10.9.11.4 境界条件....690
10.9.11.5 例....690
10.9.12 停止イベントに及ぼす停止遅延領域の影響....692
10.9.12.1 機能....692
10.9.12.2 パラメータ設定....696
10.9.12.3 プログラミング....696
10.9.12.4 必要条件....699
10.10 非同期サブプログラム(ASUB)....700
10.10.1 機能....700
10.10.1.1 プログラムモードの ASUB の実行シーケンス....702
10.10.1.2 REPOSA による ASUB....703
10.10.1.3 NC 動作....705
10.10.2 試運転:マシンデータ....706
10.10.2.1 NC 別:モードグループ別の NC/PLC インターフェース信号および操作モード切り替え....706
10.10.2.2 NC 別:ASUP 起動イネーブル....707
10.10.2.3 NC 別:パラメータ設定した起動イネーブルの有効性....709
10.10.2.4 チャネル別:原点確立されていない軸にかかわらない起動イネーブル....709
10.10.2.5 チャネル別:読み込み停止にかかわらない起動イネーブル....710
10.10.2.6 チャネル別:シングルブロックにかかわらない連続実行....710

10.10.2.7 チャネル別:表示の更新...711
10.10.3 プログラミングシステム変数...711
10.10.3.1 REPOS オプション($P_REPINF)...711
10.10.3.2 起動イベント($AC_ASUP)...712
10.10.4 プログラミング(SETINT、PRIO)...712
10.10.5 制限事項...715
10.10.6 例...716

10.11 RET および REPOS 用のユーザー固有 ASUB...717
10.11.1 機能...717
10.11.2 パラメータ設定...717
10.11.3 プログラミング...719

10.12 ユーザーアラーム用の ASUB 起動の実行...720
10.12.1 機能...720
10.12.2 起動...721
10.12.3 例...722
10.12.3.1 リセットからのユーザー ASUB -例 1...722
10.12.3.2 リセットからのユーザー ASUB -例 2...723
10.12.3.3 M0 を含むユーザー ASUB...724
10.12.3.4 停止を含むユーザー ASUB...726
10.12.3.5 停止からのユーザー ASUB...727

10.13 シングルブロック...728
10.13.1 パラメータ設定...729
10.13.2 プログラミング...731
10.13.2.1 シングルブロック加工の無効化/有効化(SBLOF、SBLON)...731
10.13.2.2 必要条件...732
10.13.2.3 例...733
10.13.3 モードグループ別シングルブロックタイプ A/B...735
10.13.4 必要条件...737
10.13.4.1 SBL2 シングルブロックタイプおよびブロック関連のシンクロナイズドアクション...737
10.13.4.2 プログラムされた停止(M0)、シングルブロック、およびシングルブロックタイプの切り替え...737

10.14 プログラム制御...738
10.14.1 ユーザーインターフェースまたは PLC ユーザープログラムからの機能選択...738
10.14.2 スキップレベルの起動...739
10.14.3 補間バッファサイズの調整...741
10.14.4 追加の基本ブロック表示を介したプログラム表示モード...743
10.14.5 ShopMill/ShopTurn の基本ブロック表示...744
10.14.6 DIN ブロックの構成...747
10.14.7 外部からの実行...751
10.14.8 外部サブプログラムの実行(EXTCALL)...753

10.15 外部記憶からの実行(EES) (オプション)...756
10.15.1 機能...756
10.15.2 セットアップ...760

10.15.2.1 ドライブの設定...760
10.15.2.2 グローバルパートプログラムメモリ(GDIR)...762
10.15.2.3 EES のパートプログラムでのファイル処理の設定...764
10.15.2.4 メモリ構成...766
10.15.3 必要条件...767

10.16 Process Datashare - 外部の機器/ファイルへの出力...767
10.16.1 機能...767
10.16.2 セットアップ...769
10.16.3 プログラミング...772
10.16.4 必要条件...777

10.17 電源投入、RESET/パートプログラム終了およびパートプログラム起動のシステム設定...778
10.17.1 方向座標変換による電源投入後の工具後退...783

10.18 サブプログラムによる機能の置換...786
10.18.1 概要...786
10.18.2 M、T/TCA、および D/DL 機能の置換...787
10.18.2.1 M 機能の置換...787
10.18.2.2 T/TCA と D/DL 機能の置換...790
10.18.2.3 システム変数...793
10.18.2.4 例: M 機能の置換...795
10.18.2.5 例: T 機能と D 機能の置換...798
10.18.2.6 競合時の動作...800
10.18.3 主軸機能の置換...801
10.18.3.1 一般情報...801
10.18.3.2 M40 ～ M45 (ギヤ選択切替)の置換...803
10.18.3.3 SPOS、SPOSA、M19 (主軸位置決め)の置換...804
10.18.3.4 システム変数...805
10.18.3.5 例:ギヤ選択切替...806
10.18.3.6 例:主軸位置決め...808
10.18.4 サブプログラムのプロパティ...812
10.18.5 制限事項...813

10.19 NC 命令の名前変更/ロック...814

10.20 プログラム実行時間/ワークカウンタ...817
10.20.1 プログラム実行時間...817
10.20.1.1 機能...817
10.20.1.2 セットアップ...823
10.20.1.3 必要条件...825
10.20.1.4 例...825
10.20.2 ワークカウンタ...827
10.20.2.1 機能...827
10.20.2.2 セットアップ...828
10.20.2.3 必要条件...829
10.20.2.4 例...830

10.21 データリスト..832
10.21.1 機能..832
10.21.2 マシンデータ..833
10.21.2.1 一般マシンデータ..833
10.21.2.2 チャネルマシンデータ..835
10.21.2.3 軸/主軸マシンデータ..839
10.21.3 セッティングデータ..840
10.21.3.1 チャネルセッティングデータ..840
10.21.4 信号..840
10.21.4.1 NC への信号..840
10.21.4.2 モードグループの信号..840
10.21.4.3 NC への信号..841
10.21.4.4 チャネルへの信号..841
10.21.4.5 チャネルからの信号..842
10.21.4.6 NC への信号..844
10.21.4.7 軸/主軸からの信号..844

11 K2: 軸タイプ、座標系、フレーム..845

11.1 概略説明..845
11.1.1 軸..845
11.1.2 座標系..847
11.1.3 フレーム ..849

11.2 軸..853
11.2.1 概要..853
11.2.2 機械軸..854
11.2.3 チャネル軸..855
11.2.4 ジオメトリ軸..855
11.2.5 付加軸..856
11.2.6 軌跡軸..856
11.2.7 位置決め軸..857
11.2.8 メイン軸..858
11.2.9 同期軸..859
11.2.10 軸設定..861
11.2.11 リンク軸..863

11.3 原点とレファレンス点..864
11.3.1 作業空間のレファレンス点..864
11.3.2 座標系とレファレンス点の位置..866

11.4 座標系..867
11.4.1 概要..867
11.4.2 機械座標系(MCS)..870
11.4.2.1 原点確立状態をリセットして現在値を設定(PRESETON)..872
11.4.2.2 原点確立状態をリセットしない現在値の設定(PRESETONS)..876
11.4.3 基本座標系(BCS)..883

11.4.4 基本ゼロオフセットシステム(BZS)....885
11.4.5 設定可能ゼロオフセットシステム(SZS)....887
11.4.6 ワーク座標系(WCS)....889
11.4.7 追加オフセット....889
11.4.7.1 外部ゼロオフセット....889
11.4.7.2 DRF オフセット....891
11.4.7.3 リセット応答....892
11.4.8 軸別重畳($AA_OFF)....892
11.4.8.1 機能....892
11.4.8.2 セットアップ....893
11.4.8.3 プログラミング軸別の重畳の解除(CORROF)....893

11.5 フレーム....897
11.5.1 フレームタイプ....897
11.5.2 フレームコンポーネント....898
11.5.2.1 座標移動....898
11.5.2.2 仕上げオフセット....899
11.5.2.3 回転の概要(ジオメトリ軸のみ)....901
11.5.2.4 オイラー角による回転: ZY'X"形式(RPY 角)....902
11.5.2.5 オイラー角による回転: ZX'Z"形式....908
11.5.2.6 任意の平面の回転....909
11.5.2.7 スケーリング....911
11.5.2.8 ミラーリング....911
11.5.2.9 結合演算子....912
11.5.2.10 プログラム可能な軸名称....912
11.5.2.11 座標変換....913
11.5.3 データ管理フレームと有効フレーム....914
11.5.3.1 概要....914
11.5.3.2 データ管理フレームの起動....916
11.5.3.3 NCU グローバルフレームとチャネル別フレーム....918
11.5.4 フレーム結合と座標系....918
11.5.4.1 概要....918
11.5.4.2 相対座標系....920
11.5.4.3 選択可能な SZS....922
11.5.4.4 WCS または SZS でのジオメトリ軸の手動移動($AC_JOG_COORD)....923
11.5.4.5 フレームのマスク....924
11.5.5 フレーム結合のフレーム....926
11.5.5.1 概要....926
11.5.5.2 設定可能フレーム($P_UIFR[<n>])....926
11.5.5.3 研削フレーム$P_GFR[<n>]....929
11.5.5.4 チャネル別基本フレーム[<n>]....933
11.5.5.5 NCU グローバル基本フレーム$P_NCBFR[<n>]....935
11.5.5.6 有効な基本フレーム一式$P_ACTBFRAME....936
11.5.5.7 プログラマブルフレーム$P_PFRAME....938
11.5.5.8 チャネル別システムフレーム....940

11.5.6 自動的なフレーム変更...943
11.5.6.1 ジオメトリ軸の切り替え...943
11.5.6.2 座標変換の選択と選択解除:概要...947
11.5.6.3 座標変換の選択と選択解除:TRANSMIT...947
11.5.6.4 座標変換の選択と選択解除:TRACYL...954
11.5.6.5 座標変換の選択と選択解除:TRAANG...960
11.5.6.6 有効なフレームの補正...966
11.5.6.7 マッピングされたフレーム...967
11.5.7 事前定義されたフレーム機能...971
11.5.7.1 逆フレーム...971
11.5.7.2 フレーム結合内の追加フレーム...974
11.5.8 動作...975
11.5.8.1 原点の設定、ワーク計測および工具計測...975
11.5.8.2 軸別外部ゼロオフセット...976
11.5.8.3 工具ホルダ...977
11.5.9 SAVE 属性を持つサブプログラム(SAVE)...989
11.5.10 データバックアップ...990
11.5.11 座標系の位置...991
11.5.12 コントロールシステムの応答 ...991
11.5.12.1 POWER ON...991
11.5.12.2 モード変更...992
11.5.12.3 チャネルリセット/パートプログラム終了...993
11.5.12.4 パートプログラム起動...998
11.5.12.5 ブロックサーチ...999
11.5.12.6 REPOS...1000

11.6 ワーク基準のフィードバック回路...1000
11.6.1 概要...1000
11.6.2 ワーク基準のフィードバック回路の使用...1000
11.6.3 特殊な応答...1003

11.7 制限事項...1005

11.8 例...1005
11.8.1 軸...1005
11.8.2 座標系...1009
11.8.3 フレーム ...1011

11.9 データリスト...1013
11.9.1 マシンデータ...1013
11.9.1.1 表示マシンデータ...1013
11.9.1.2 NC 別マシンデータ...1014
11.9.1.3 チャネルマシンデータ...1015
11.9.1.4 軸/主軸マシンデータ...1016
11.9.2 セッティングデータ...1016
11.9.2.1 チャネルセッティングデータ...1016
11.9.3 システム変数...1017

11.9.4 信号 1019
11.9.4.1 チャネルからの信号 1019
11.9.4.2 軸/主軸への信号 1019
11.9.4.3 軸/主軸からの信号 1019

12 N2: 非常停止 1021

12.1 概略説明 1021
12.2 関連規格 1021
12.3 非常停止制御部品 1022
12.4 非常停止処理 1023
12.5 非常停止応答 1025
12.6 データリスト 1028
12.6.1 マシンデータ 1028
12.6.1.1 軸/主軸マシンデータ 1028
12.6.2 信号 1028
12.6.2.1 NC への信号 1028
12.6.2.2 NC からの信号 1028
12.6.2.3 モードグループへの信号 1028

13 P1:径方向軸 1029

13.1 機能 1029
13.2 パラメータ設定 1033
13.3 プログラミング 1037
13.4 必要条件 1038
13.5 例 1039
13.6 データリスト 1040
13.6.1 マシンデータ 1040
13.6.1.1 チャネルマシンデータ 1040
13.6.1.2 軸/主軸マシンデータ 1041

14 P3: SINUMERIK 840D sl 用 PLC 基本プログラム 1043

14.1 概略説明 1043
14.2 PLC CPU のキーデータ 1045
14.3 PLC オペレーティングシステムのバージョン 1046
14.4 PLC モード選択 1047
14.5 予約リソース(タイマ、カウンタ、FC、FB、DB、I/O) 1047
14.6 PLC CPU のハードウェアコンフィグレーションのセットアップ 1048
14.7 PLC プログラムのセットアップ 1048

14.7.1 基本プログラムのインストール ... 1048
14.7.2 基本プログラムの用途 ... 1049
14.7.3 バージョンコード ... 1050
14.7.4 機械プログラム ... 1051
14.7.5 データバックアップ ... 1052
14.7.6 PLC 一括セットアップ、PLC アーカイブ ... 1052
14.7.7 ソフトウェアのアップグレード ... 1056
14.7.8 I/O モジュール(FM、CP モジュール) ... 1057
14.7.9 トラブルシューティング ... 1059

14.8 PLC CPU の連結 ... 1060
14.8.1 一般情報 ... 1060
14.8.2 PLC CPU の特性 ... 1061
14.8.3 内蔵 PLC とのインタフェース ... 1061
14.8.4 PLC の診断バッファ ... 1064

14.9 インタフェース構造 ... 1064
14.9.1 PLC/NCK インタフェース ... 1064
14.9.2 PLC/HMI インタフェース ... 1073
14.9.3 PLC/MCP/HHU インタフェース ... 1078

14.10 基本プログラムの構造とファンクション ... 1080
14.10.1 NCK PLC の起動および同期 ... 1082
14.10.2 サイクリック運転(OB1) ... 1082
14.10.3 時間割り込み処理(OB35) ... 1085
14.10.4 プロセス割り込み処理(OB 40) ... 1086
14.10.5 診断アラーム、モジュール故障処理(OB82、OB86) ... 1086
14.10.6 NCK 故障に対する応答 ... 1087
14.10.7 ユーザープログラムから呼び出される基本プログラムのファンクション ... 1088
14.10.8 インタフェース DB によるユーザープログラムのシンボルプログラミング ... 1092
14.10.9 リストに応じた M デコード ... 1094
14.10.10 PLC マシンデータ ... 1098
14.10.11 機械操作パネル、ハンドヘルドユニット、ダイレクトキーの構成 ... 1103
14.10.12 機械操作パネル、ハンドヘルドユニットの切り替え ... 1115

14.11 Safety Integrated 用の SPL ... 1117

14.12 割り付け一覧 ... 1117
14.12.1 割り付け: NCK/PLC インタフェース ... 1117
14.12.2 割り付け:FB/FC ... 1118
14.12.3 割り付け:DB ... 1118
14.12.4 割り付け:タイマ ... 1120

14.13 HMI 用の PLC 機能(DB19) ... 1120
14.13.1 チャネルの選択 ... 1120
14.13.2 プログラムの選択 ... 1122
14.13.3 キーロックの有効化 ... 1125
14.13.4 操作エリア番号 ... 1125

14.13.5 画面番号 1126
14.13.5.1 画面番号:JOG、Manual Machine 1127
14.13.5.2 画面番号:基準点アプローチ 1133
14.13.5.3 画面番号:MDI 1134
14.13.5.4 画面番号:AUTOMATIC 1134
14.13.5.5 画面番号:パラメータ操作エリア 1135
14.13.5.6 画面番号:プログラム操作エリア 1137
14.13.5.7 画面番号:プログラムマネージャ操作エリア 1137
14.13.5.8 画面番号:アラーム診断操作エリア 1138
14.13.6 HMI モニタ 1138

14.14 内蔵 PROFIBUS 上のドライブコンポーネントの PLC 機能 1140
14.14.1 一覧 1140
14.14.2 セットアップの実行 1140
14.14.3 例 1141

14.15 PLC 基本プログラムの必要メモリ 1142

14.16 必要条件および NC VAR セレクタ 1147
14.16.1 必要条件 1147
14.16.1.1 プログラミングとパラメータ設定ツール 1147
14.16.1.2 必要な SIMATIC 取扱説明書 1149
14.16.1.3 関連 SINUMERIK 文書 1149
14.16.2 NC VAR セレクタ 1150
14.16.2.1 概要 1150
14.16.2.2 機能説明 1153
14.16.2.3 セットアップ、インストール 1162

14.17 ブロックの説明 1162
14.17.1 FB1:RUN_UP - 基本プログラム、起動セクション 1162
14.17.2 FB2:GET - NC 変数の読み取り 1173
14.17.3 FB3:PUT - NC 変数の書き込み 1184
14.17.4 FB4:PI_SERV - PI サービス要求 1193
14.17.4.1 使用可能な PI サービスのリスト 1199
14.17.4.2 PI サービス ASUB 1201
14.17.4.3 PI サービス CANCEL 1202
14.17.4.4 PI サービス CONFIG 1203
14.17.4.5 PI サービス DIGION 1203
14.17.4.6 PI サービス DIGIOF 1203
14.17.4.7 PI サービス FINDBL 1204
14.17.4.8 PI サービス LOGIN 1204
14.17.4.9 PI サービス LOGOUT 1205
14.17.4.10 PI サービス NCRES 1205
14.17.4.11 PI サービス SELECT 1205
14.17.4.12 PI サービス SETUDT 1206
14.17.4.13 PI サービス SETUFR 1207
14.17.4.14 PI サービス RETRAC 1207

14.17.4.15 PI サービス CRCEDN ... 1208
14.17.4.16 PI サービス CREACE ... 1209
14.17.4.17 PI サービス CREATO ... 1209
14.17.4.18 PI サービス DELECE ... 1210
14.17.4.19 PI サービス DELETO ... 1210
14.17.4.20 PI サービス MMCSEM ... 1211
14.17.4.21 PI サービス TMCRTO ... 1212
14.17.4.22 PI サービス TMFDPL ... 1214
14.17.4.23 PI サービス TMFPBP ... 1215
14.17.4.24 PI サービス TMGETT ... 1217
14.17.4.25 PI サービス TMMVTL ... 1218
14.17.4.26 PI サービス TMPOSM ... 1220
14.17.4.27 PI サービス TMPCIT ... 1221
14.17.4.28 PI サービス TMRASS ... 1222
14.17.4.29 PI サービス TRESMO ... 1222
14.17.4.30 PI サービス TSEARC ... 1223
14.17.4.31 PI サービス TMCRMT ... 1227
14.17.4.32 PI サービス TMDLMT ... 1228
14.17.4.33 PI サービス POSMT ... 1229
14.17.4.34 PI サービス FDPLMT ... 1230
14.17.5 FB5:GETGUD - GUD 変数の読み取り ... 1232
14.17.6 FB7:PI_SERV2 - PI サービス要求 ... 1242
14.17.7 FB9:MtoN - 操作パネルの切り替え ... 1243
14.17.8 FB10:安全リレー(SI リレー) ... 1249
14.17.9 FB11:ブレーキテスト ... 1252
14.17.10 FB29:信号レコーダおよびデータトリガ診断 ... 1258
14.17.11 FC2:GP_HP - 基本プログラム、サイクリックセクション ... 1263
14.17.12 FC3:GP_PRAL - 基本プログラム、割り込み起動セクション ... 1264
14.17.13 FC5:GP_DIAG - 基本プログラム、診断アラーム、およびモジュール故障 ... 1267
14.17.14 FC6:TM_TRANS2 - 工具管理およびマルチツール用伝送ブロック ... 1269
14.17.15 FC7:TM_REV - リボルバの工具交換伝送ブロック ... 1271
14.17.16 FC8:TM_TRANS - 工具管理機能用伝送ブロック ... 1275
14.17.17 FC9:ASUB - 非同期サブプログラムの起動 ... 1284
14.17.18 FC10:AL_MSG - 異常メッセージおよび操作メッセージ ... 1287
14.17.19 FC12:AUXFU - 補助機能のユーザー用呼び出しインターフェース ... 1289
14.17.20 FC13:BHGDisp - ハンドヘルドユニット用表示制御 ... 1291
14.17.21 FC17:YDelta - スター/デルタ切り替え ... 1297
14.17.22 FC18:SpinCtrl - 主軸制御 ... 1301
14.17.23 FC19:MCP_IFM - MCP 信号のインターフェースへの伝送 ... 1314
14.17.24 FC21:転送 - データ交信 NC/PLC ... 1322
14.17.24.1 機能 ... 1322
14.17.24.2 ファンクションの宣言 ... 1323
14.17.24.3 仮パラメータの説明 ... 1323
14.17.24.4 機能 1、2:チャネルへ/チャネルからのシンクロナイズドアクション通知 ... 1324
14.17.24.5 機能 3、4:PLC-NC 間での高速データ交換 ... 1325

14.17.24.6 機能 5:チャネルへの制御信号更新 1330
14.17.24.7 機能 6:軸への制御信号更新 1331
14.17.24.8 機能 7:軸への制御信号更新 1331
14.17.25 FC22:TM_DIR - 工具管理用の方向選択 1332
14.17.26 FC24:MCP_IFM2 - インターフェースへの MCP 信号の伝送 1335
14.17.27 FC25:MCP_ IFT - インターフェースへの MCP/OP 信号の伝送 1339
14.17.28 FC26:HPU_MCP - インターフェースへの HT 8 信号の伝送 1343
14.17.28.1 HT8 の NC/PLC インタフェース信号の一覧 1348
14.17.28.2 HT8 の NC/PLC インタフェース信号の一覧 1350
14.17.29 FC1005:AG_SEND - Ethernet CP へのデータの伝送 1350
14.17.30 FC1006:AG_RECV - Ethernet CP からのデータの受信 1352

14.18 信号/データの説明 1354
14.18.1 NCK/PLC、HMI/PLC、MCP/PLC の各インタフェース信号 1354
14.18.2 デコードされた M 信号 1354
14.18.3 G 命令 1355
14.18.4 DB 2 のメッセージ信号 1356

14.19 STEP 7 のプログラミングに関する注意事項 1357
14.19.1 データのコピー 1358
14.19.2 ANY および POINTER 1358
14.19.2.1 FC での POINTER および ANY の使用 1359
14.19.2.2 FB での POINTER および ANY の使用 1360
14.19.2.3 FC または FB への伝送用 POINTER または ANY 変数 1361
14.19.3 マルチインスタンス DB 1364
14.19.4 文字列 1365
14.19.5 データブロック構造のオフセットアドレスの決定 1366
14.19.6 FB 呼び出し 1367

14.20 データリスト 1368
14.20.1 マシンデータ 1368
14.20.1.1 表示マシンデータ 1368
14.20.1.2 NC 別マシンデータ 1369
14.20.1.3 チャネルマシンデータ 1369
14.20.2 信号 1369
14.20.2.1 操作パネルからの信号 1369

15 P4 : SINUMERIK 828D 用 PLC 1371

15.1 概要 1371
15.1.1 PLC ファームウェア 1371
15.1.2 PLC ユーザーインタフェース 1371
15.1.2.1 周期的に交信されるデータ 1374
15.1.2.2 アラームとメッセージ 1374
15.1.2.3 保持データ 1375
15.1.2.4 非保持データ 1375
15.1.2.5 PLC マシンデータ 1375
15.1.3 PLC 主要データ 1375

15.1.4 PLC I/O、高速オンボード入力/出力 ... 1376
15.1.5 PLC ツールボックス ... 1377
15.1.5.1 スター/デルタ切り替え ... 1377
15.2 高速オンボード入出力部 ... 1377
15.3 ラダー図、ラダーエディタおよびラダーアドオンツール ... 1379
15.3.1 一覧 ... 1379
15.3.2 パラメータ設定 ... 1381
15.4 PLC プログラミングツール ... 1382
15.5 データインタフェース ... 1382
15.5.1 PLC-NC インタフェース ... 1383
15.5.1.1 モード信号 ... 1383
15.5.1.2 NC チャネル信号 ... 1385
15.5.1.3 軸および主軸信号 ... 1386
15.5.1.4 一般 NC 信号 ... 1387
15.5.1.5 PLC-NC 高速データ交換 ... 1387
15.5.2 PLC-HMI 間インタフェース ... 1388
15.5.2.1 プログラムの選択 ... 1389
15.5.2.2 操作エリア番号 ... 1392
15.5.2.3 画面番号 ... 1393
15.5.2.4 HMI モニタ ... 1405
15.6 機能インタフェース ... 1406
15.6.1 NC 変数の読み取り/書き込み ... 1406
15.6.1.1 ユーザーインタフェース ... 1406
15.6.1.2 ジョブの指定 ... 1407
15.6.1.3 ジョブの管理: ジョブの開始 ... 1409
15.6.1.4 ジョブの管理: ジョブの終了の待機 ... 1409
15.6.1.5 ジョブの管理: ジョブの完了 ... 1410
15.6.1.6 ジョブの管理:フローチャート ... 1411
15.6.1.7 ジョブの評価 ... 1412
15.6.1.8 操作可能な変数 ... 1414
15.6.1.9 選択された NC 変数の指定 ... 1424
15.6.2 プログラムインスタンスサービス(PI サービス) ... 1426
15.6.2.1 ジョブの指定 ... 1426
15.6.2.2 ジョブのフィードバック ... 1427
15.6.2.3 PI サービス ASUB ... 1428
15.6.2.4 PI サービス LOGOUT ... 1430
15.6.2.5 PI サービス DATA_SAVE ... 1431
15.6.2.6 PI サービス TMMVTL ... 1431
15.6.2.7 PI サービス: タイミングチャート ... 1433
15.6.3 PLC ユーザーアラーム ... 1433
15.6.3.1 ユーザーインタフェース ... 1433
15.6.3.2 ユーザーアラームの起動インタフェース ... 1434
15.6.3.3 ユーザーアラームの変数インタフェース ... 1436

15.6.3.4 ユーザーアラームの設定 1437
15.6.3.5 有効なアラーム応答と解除条件のエクスポート 1440
15.6.3.6 ユーザーアラームの応答インタフェース 1440
15.6.3.7 HMI へのインタフェース 1441
15.6.4 PLC 軸制御 1441
15.6.4.1 一般情報 1441
15.6.4.2 ユーザーインタフェース: PLC 軸としての NC 軸の準備 1443
15.6.4.3 ユーザーインタフェース:機能 1444
15.6.4.4 主軸位置決め 1446
15.6.4.5 主軸回転 1448
15.6.4.6 主軸揺動 1450
15.6.4.7 軸割り出し 1453
15.6.4.8 軸位置決め、メトリック 1455
15.6.4.9 軸位置決め、インチ 1457
15.6.4.10 手動パルス発生器の移動量重畳による軸位置決め、メトリック 1459
15.6.4.11 手動パルス発生器の移動量重畳による軸位置決め、インチ 1461
15.6.4.12 自動ギヤ選択切替を使用した主軸回転 1462
15.6.4.13 周速一定制御[m/min]を使用した主軸回転 1464
15.6.4.14 周速一定制御[ft/min]を使用した主軸回転 1466
15.6.4.15 エラーメッセージ 1468
15.6.5 ASUB の起動 1470
15.6.5.1 ジョブの開始 1470
15.6.5.2 ジョブ結果 1471
15.6.5.3 信号フロー 1473
15.6.6 HMI でのチャネル選択 1474

15.7 CNC ロック機能(オプション) 1476
15.7.1 機能 1476
15.7.2 必要条件 1476
15.7.3 制限事項 1477
15.7.4 不正使用からの保護 1478
15.7.5 最初の CNC ロック機能の作成 1478
15.7.6 CNC ロック機能の拡張 1482
15.7.7 CNC ロック機能の解除 1484
15.7.8 故障したコントロールシステムハードウェア(PPU)の交換 1487
15.7.9 故障した CF カードの交換 1490
15.7.10 OEM PIN を忘れた場合 1492
15.7.11 その他の情報 1493

16 R1:レファレンス点復帰 1497

16.1 概略説明 1497

16.2 軸別原点確立 1498

16.3 チャネル別の原点確立 1500

16.4 パートプログラムによるレファレンス点復帰(G74) 1502

16.5 インクリメンタルエンコーダによる原点確立 1503
16.5.1 ハードウェア信号 1503
16.5.2 原点マーク選択 1506
16.5.3 タイミング図 1507
16.5.4 フェーズ 1:レファレンス点スイッチへの移動 1508
16.5.5 フェーズ 2:原点マークとの同期 1511
16.5.6 フェーズ 3: レファレンス点への移動 1516

16.6 絶対番地化原点マークによるレファレンス点復帰動作 1518
16.6.1 概要 1518
16.6.2 基本パラメータ設定 1519
16.6.3 タイミング図 1521
16.6.4 フェーズ 1: 同期をともなう原点マークの通過 1521
16.6.5 フェーズ 2:目標点への移動 1523

16.7 フィードバック値の位置合わせによる原点確立 1526
16.7.1 フィードバック値の原点確立検出器への位置合わせ 1526
16.7.2 絶対番地化原点マークのある検出器へのフィードバック値の位置合わせ 1526

16.8 フォローアップモードでの原点確立 1528

16.9 絶対値エンコーダの原点確立 1531
16.9.1 位置合わせに関する情報 1531
16.9.2 レファレンス点オフセットの入力による位置合わせ 1533
16.9.3 レファレンス点位置の入力による位置合わせ 1535
16.9.4 プローブによる自動校正 1537
16.9.5 BERO による位置合わせ 1538
16.9.6 絶対値エンコーダによるレファレンス点復帰 1539
16.9.7 等価原点マークによるロータリー絶対値エンコーダでのレファレンス点復帰 1540
16.9.8 検出器の起動 1543
16.9.9 サポートされない原点確立方法 1544

16.10 機械原点の自動復元 1545
16.10.1 自動原点確立 1546
16.10.2 実位置の復元 1547

16.11 必要条件 1549
16.11.1 広範な移動範囲 1549

16.12 データリスト 1550
16.12.1 マシンデータ 1550
16.12.1.1 NC 別マシンデータ 1550
16.12.1.2 チャネルマシンデータ 1550
16.12.1.3 軸/主軸マシンデータ 1551
16.12.2 信号 1552
16.12.2.1 モードグループへの信号 1552
16.12.2.2 モードグループからの信号 1553
16.12.2.3 チャネルへの信号 1553

16.12.2.4 チャネルからの信号 1553
16.12.2.5 軸/主軸への信号 1553
16.12.2.6 軸/主軸からの信号 1554

17 S1:主軸 1555

17.1 概略説明 1555

17.2 モード 1556
17.2.1 概要 1556
17.2.2 モード変更 1557
17.2.3 コントロールモード 1558
17.2.4 揺動モード 1562
17.2.5 位置決めモード 1562
17.2.5.1 一般機能 1562
17.2.5.2 回転からの位置決め 1571
17.2.5.3 停止状態からの位置決め 1576
17.2.5.4 工具交換用の「主軸インポジション」信号 1580
17.2.6 軸モード 1582
17.2.6.1 一般機能 1582
17.2.6.2 軸モードへの自動遷移 1585
17.2.7 初期主軸状態 1589
17.2.8 フローティングチャックのないタッピング 1590
17.2.8.1 機能 1590
17.2.8.2 プログラミング 1591
17.2.8.3 例:G331/G332 のタッピング 1592
17.2.8.4 例:現在のギヤ選択でプログラム指令穴あけ速度を出力 1593
17.2.8.5 例:2 番目のギヤ選択データブロックの用途 1593
17.2.8.6 例:速度はプログラムされておらず、ギヤボックス選択が監視されています 1594
17.2.8.7 例:ギヤボックスの選択を変更できません、ギヤボックスの選択は監視されています 1594
17.2.8.8 例:SPOS を使用しないプログラミング 1595
17.2.8.9 特記事項:NC プログラムの NC/PLC インタフェース信号による回転方向の反転 1595
17.2.9 フローティングチャックを使用したタッピング 1597
17.2.9.1 機能 1597
17.2.9.2 プログラミング 1597

17.3 原点確立/原点同期 1599

17.4 設定可能なギヤ調整 1604
17.4.1 主軸のギヤ選択とギヤ選択切替 1604
17.4.2 主軸ギヤ選択 0 1615
17.4.3 主軸ギヤ選択の特定 1618
17.4.4 ギヤ選択切替中のパラメータセット選択 1619
17.4.5 中間ギヤ 1622
17.4.6 応答のないギヤ選択切替 1624
17.4.7 揺動モードでのギヤ選択切替 1625
17.4.8 固定位置でのギヤ選択切替 1632
17.4.9 M70 の設定可能なギヤ選択 1639

17.4.10 ドライラン、プログラムテスト、および SERUPRO に対するギヤ選択切替のマスク. 1642
17.5 設定可能な主軸機能の追加の調整 1644
17.6 主軸再割付け 1647
17.7 プログラミング 1651
17.7.1 パートプログラムによるプログラミング 1651
17.7.2 シンクロナイズドアクションを介したプログラミング 1656
17.7.3 FC18 による PLC を介した主軸制御のプログラミング - 840D sl のみ 1657
17.7.4 NC/PLC インタフェース信号を使用してプログラミング 1657
17.7.4.1 機能 1657
17.7.4.2 試運転:マシンデータ 1659
17.7.4.3 試運転:NC/PLC インタフェース信号 1660
17.7.4.4 速度指令値 (SD43200) 1660
17.7.4.5 周速一定制御の入力(SD43202) 1661
17.7.4.6 メイン主軸の主軸速度タイプの入力(SD43206) 1663
17.7.5 外部プログラミング(PLC、HMI) 1663
17.8 主軸監視 1665
17.8.1 許容速度範囲 1665
17.8.2 軸/主軸停止 1666
17.8.3 主軸指令一致 1666
17.8.4 ギヤ選択の最高/最低速度 1668
17.8.5 主軸速度制限の診断 1669
17.8.6 最高主軸速度 1671
17.8.7 最大エンコーダ限界周波数 1673
17.8.8 終点監視 1674
17.8.9 M40: 設定された切り替えレベルを超えた速度の自動ギヤ選択 1676
17.9 SMI 24 付き主軸(ワイス主軸) 1678
17.9.1 一般事項 1678
17.9.2 センサデータ 1679
17.9.3 クランプ済み状態 1681
17.9.4 追加ドライブパラメータ 1683
17.10 必要条件 1683
17.10.1 制御パラメータの変更 1683
17.11 例 1684
17.11.1 自動ギヤ選択(M40) 1684
17.12 データリスト 1685
17.12.1 マシンデータ 1685
17.12.1.1 NC 別マシンデータ 1685
17.12.1.2 チャネルマシンデータ 1686
17.12.1.3 軸/主軸マシンデータ 1686
17.12.2 セッティングデータ 1688
17.12.2.1 チャネルセッティングデータ 1688

17.12.2.2 軸/主軸別セッティングデータ 1689
17.12.3 信号 1689
17.12.3.1 軸/主軸への信号 1689
17.12.3.2 軸/主軸からの信号 1690

18 V1:送り速度 1693

18.1 概略説明 1693

18.2 軌跡送り速度 F 1694
18.2.1 送り速度タイプ G93、G94、G95 1696
18.2.2 G96、G961、G962、G97、G971 送り速度のタイプ 1700
18.2.3 ねじ切りの送り速度(G33、G34、G35、G335、G336) 1705
18.2.3.1 G33 による送り速度 1705
18.2.3.2 G34 と G35 による単調増加/単調減少の可変ねじピッチ 1706
18.2.3.3 G33、G34、および G35 の軸の加速度特性 1708
18.2.3.4 G33、G34、および G35(DITS、DITE)でプログラムされた切り始めと切り上げの距離 1711
18.2.3.5 ねじ切り時の高速リトラクト 1713
18.2.3.6 凸型ねじ(G335、G336) 1718
18.2.4 フローティングチャックなしのタッピングの送り速度(G331、G332) 1723
18.2.5 フローティングチャック付きタッピングの送り速度(G63) 1725
18.2.6 FGROUP と FGREF 1726

18.3 位置決め軸の送り速度(FA) 1729

18.4 送り速度の制御 1730
18.4.1 送り速度無効および送り速度/主軸停止 1730
18.4.2 機械操作パネルでの送り速度オーバライド 1732
18.4.3 プログラム可能な送り速度オーバライド 1737
18.4.4 ドライラン送り速度 1738
18.4.5 1 ブロックでの複数送り速度値 1741
18.4.6 固定送り速度値 1748
18.4.7 プログラム可能な送り速度特性 1749
18.4.8 面取り/丸み付け FRC、FRCM の送り速度 1751
18.4.9 ノンモーダル送り速度 FB 1754
18.4.10 単独軸のダイナミック応答に対するの影響 1755

18.5 必要条件 1760

18.6 データリスト 1761
18.6.1 マシンデータ 1761
18.6.1.1 NC 別マシンデータ 1761
18.6.1.2 チャネルマシンデータ 1762
18.6.1.3 軸/主軸マシンデータ 1763
18.6.2 セッティングデータ 1763
18.6.2.1 チャネルセッティングデータ 1763
18.6.2.2 軸/主軸セッティングデータ 1764
18.6.3 信号 1764

18.6.3.1 チャネルへの信号 1764
18.6.3.2 チャネルからの信号 1764
18.6.3.3 軸/主軸への信号 1765
18.6.3.4 軸/主軸からの信号 1765

19 W1:工具補正 1767

19.1 概略説明 1767

19.2 工具 1770
19.2.1 一般情報 1770
19.2.2 補正メモリ構造 1773
19.2.3 工具補正の計算 1775
19.2.4 NC アドレス T および M のアドレス拡張 1775
19.2.5 D 番号の任意割り当て 1777
19.2.6 工具交換中の異常発生時の補正ブロック 1784
19.2.7 工具パラメータの働きの定義 1787

19.3 フラット D 番号構造 1788
19.3.1 一般情報 1788
19.3.2 新しい D 番号(補正ブロック)の作成 1789
19.3.3 D 番号のプログラミング 1790
19.3.4 T 番号のプログラミング 1792
19.3.5 M6 のプログラミング 1793
19.3.6 プログラムテスト 1793
19.3.7 工具管理機能または「フラット D 番号構造」 1794

19.4 工具刃先 1795
19.4.1 一般情報 1795
19.4.2 工具パラメータ 1:工具タイプ 1797
19.4.3 工具パラメータ 2:刃先位置 1801
19.4.4 工具パラメータ 3 ～ 5: ジオメトリ - 工具長 1803
19.4.5 工具パラメータ 6 ～ 11: ジオメトリ - 工具形状 1805
19.4.6 工具パラメータ 12 ～ 14: 摩耗 - 工具長 1807
19.4.7 工具パラメータ 15 ～ 20: 摩耗 - 工具形状 1807
19.4.8 工具パラメータ 21 ～ 23: 工具ベース寸法/アダプタ寸法 1808
19.4.9 工具パラメータ 24: 逃げ角 1809
19.4.10 関連する工具刃先位置を持つ工具 1811

19.5 2 次元工具径補正(2D-WRK) 1811
19.5.1 一般情報 1811
19.5.2 TRC の選択(G41/G42) 1813
19.5.3 アプローチおよび後退動作(NORM/KONT/KONTC/KONTT) 1814
19.5.4 滑らかなアプローチと後退 1819
19.5.4.1 機能 1819
19.5.4.2 パラメータ 1819
19.5.4.3 速度 1827
19.5.4.4 システム変数 1829

19.5.4.5 必要条件 1830
19.5.4.6 例 1831
19.5.5 TRC の解除(G40) 1834
19.5.6 外側コーナでの補正 1834
19.5.7 補正と内側コーナ 1839
19.5.8 衝突監視(「ボトルネック検出」) 1841
19.5.8.1 機能 1841
19.5.8.2 パラメータ設定 1842
19.5.8.3 プログラミング 1843
19.5.8.4 必要条件 1843
19.5.8.5 例 1844
19.5.9 スロット形状認識(オプション) - 840D sl のみ 1845
19.5.10 可変補正値のあるブロック 1848
19.5.11 アラーム応答 1850
19.5.12 多項式用の交点処理 1851
19.5.13 G461/G462 アプローチ/後退方法の拡張 1851
19.6 工具径補正の継続 1855
19.7 旋回工具ホルダ 1860
19.7.1 一般情報 1860
19.7.2 キネマテックの伝達と機械構成 1868
19.7.3 キネマティック結合による工具ホルダ 1878
19.7.4 3 + 2 軸による傾斜面加工 1888
19.7.5 回転作業テーブルを備える機械 1890
19.7.6 旋回工具ホルダの使用手順 1894
19.7.7 プログラミング 1899
19.7.8 向きのための必要条件とシステム応答 1901
19.7.9 例 1904
19.7.9.1 例: 旋回工具ホルダ 1904
19.7.9.2 回転テーブルのある旋回工具ホルダの例 1905
19.7.9.3 位置別およびワーク別の補正値の計算 1908
19.7.9.4 例:キネマティック結合による方向付け機能を備えた工具キャリア 1910
19.8 旋回工具のオフセットデータの変更 1913
19.8.1 はじめに 1913
19.8.2 旋削工具の旋回 1914
19.8.2.1 旋削工具の刃先位置、切削方向、および角度 1914
19.8.2.2 旋削工具の回転時の修正 1917
19.8.3 フライス工具と穴あけ工具の回転 1921
19.8.3.1 フライス工具およびタッピング工具の刃先位置 1921
19.8.3.2 フライス工具およびタッピング工具の回転時の修正 1922
19.8.4 セットアップ 1923
19.8.4.1 パラメータ割り当て 1923
19.8.5 プログラミング 1927
19.8.5.1 オリエンテーションの計算(ORISOLH) 1927

19.8.5.2 オリエンテーションの計算(ORISOLH):詳細情報 1934
19.8.5.3 旋回工具のオフセットデータの変更の起動(CUTMOD、CUTMODK) 1941
19.8.5.4 旋回工具のオフセットデータの変更の起動(CUTMOD、CUTMODK)詳細情報 1945
19.8.6 例 1949

19.9 インクリメンタルでプログラムされた補正値 1951
19.9.1 G91 拡張 1951
19.9.2 工具方向(MOVT)への移動 1952

19.10 ジオメトリ軸への工具長成分の割り当て 1954
19.10.1 工具タイプおよび作業平面に応じた割り当て 1954
19.10.2 平面変更時の割り当て 1954
19.10.3 工具タイプに依存しない割り当て 1956

19.11 近軸工具オリエンテーション 1957
19.11.1 基本工具オリエンテーション 1957
19.11.2 平面変更の場合の工具オリエンテーション 1958

19.12 パラメータ設定可能な基本工具のオリエンテーション 1959
19.12.1 機能 1959
19.12.2 セットアップ 1960
19.12.2.1 起動 1960
19.12.2.2 パラメータ設定 1961
19.12.3 プログラミング 1962
19.12.4 例 1964

19.13 工具補正の特殊処理 1966
19.13.1 関連するセッティングデータ 1966
19.13.2 工具長のミラーリング 1967
19.13.3 摩耗長のミラーリング 1968
19.13.4 向きを考慮した WCS の工具長 1970
19.13.5 工具方向の工具長オフセット 1970
19.13.6 旋回工具ホルダの特徴 1975

19.14 サムオフセットとセットアップオフセット 1976
19.14.1 一般情報 1976
19.14.2 機能説明 1977
19.14.3 起動 1981
19.14.4 例 1988
19.14.5 工具長定義の改良 1989
19.14.5.1 位置別およびワーク別の補正値の計算 1989
19.14.5.2 個々の摩耗値の動作 1995

19.15 工具環境の使用 1998
19.15.1 工具環境の保存(TOOLENV) 1999
19.15.2 工具環境の削除(DELTOOLENV) 2002
19.15.3 T 番号、D 番号、および DL 番号の読み出し (GETTENV) 2003
19.15.4 保存した工具環境に関する情報の読み取り($P_TOOLENVN、$P_TOOLENV) 2004
19.15.5 工具長、工具長成分、またはその両方の読み取り(GETTCOR) 2005

19.15.6 工具成分の変更(SETTCOR)..... 2013
19.16 座標軸への工具長 L1、L2、L3 の割り当ての読み取り(LENTOAX)..... 2029
19.17 必要条件..... 2032
19.17.1 フラット D 番号構造..... 2032
19.17.2 SD42935 の拡張..... 2033
19.17.3 接触計測..... 2033
19.18 データリスト..... 2033
19.18.1 マシンデータ..... 2033
19.18.1.1 NC 別マシンデータ..... 2033
19.18.1.2 チャネルマシンデータ..... 2034
19.18.1.3 軸/主軸マシンデータ..... 2036
19.18.2 セッティングデータ..... 2036
19.18.2.1 チャネルセッティングデータ..... 2036
19.18.3 信号..... 2037
19.18.3.1 チャネルからの信号..... 2037
20 Z1:NC/PLC インタフェース信号..... 2039
A 付録..... 2041
A.1 略語の一覧..... 2041
A.2 マニュアルの一覧..... 2053
用語集..... 2055
索引..... 2081

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。

A2: NC/PLC 間の各種共通インタフェース信号と各種機能

2

2.1 概略説明

内容

PLC/NC インタフェースは、一方ではデータインタフェースを、他方では機能インタフェースを構成します。データインターフェースには、状態と制御信号、補助機能およびG命令が含まれる一方、PLC から NC へのジョブの転送に機能インターフェースが使用されます。

この章では、機能全般に関連するものの、その他の機能説明の章に含まれないインタフェース信号の動作について説明します。

- 非同期イベント
- 状態信号
- PLC 変数(読み取りおよび書き込み)

2.2 NC/PLC インタフェース信号

2.2.1 一般情報

NC/PLC インタフェース

NC/PLC インタフェースは、以下の要素から構成されます。

- データインタフェース
- 機能インタフェース

データインタフェース

データインタフェースは、以下のコンポーネントの協調に使用されます。

- PLC ユーザープログラム
- NC

- HMI (操作部コンポーネント)
- MCP (機械操作パネル)

データ交信は、PLC 基本プログラムにより実行されます。

サイクリック信号交信

以下のインタフェース信号は、周期的に、すなわち OB1 の時間区切りに合わせて PLC 基本プログラムにより転送されます。

- NC と操作パネル用信号
- モードグループ別信号
- チャネル別信号
- 軸/主軸別信号

NC と操作パネル用信号(DB10)

PLC から NC:

- CNC 入出力に影響を与えるための信号
- キースイッチ信号(およびパスワード)

NC から PLC:

- CNC 入力の現在値
- CNC 出力の指令値
- NC と HMI からの準備完了信号
- NC 状態信号(アラーム信号)

チャネル別信号(DB21、...)

PLC から NC:

- 制御信号「残移動距離削除」

NC から PLC:

- NC 状態信号(NC アラーム有効)

軸/主軸別信号(DB31 など)

PLC から NC:

- 軸/主軸への制御信号(フォローアップモード、サーボイネーブルなど)
- ドライブへの制御信号(バイト 20、21)

NC から PLC:

- 軸/主軸からの状態信号(位置コントローラ有効、電流コントローラ有効など)
- ドライブからの制御信号(バイト 93、94)

機能インタフェース

機能インタフェースは、ファンクションブロック(FB)とファンクションコール(FC)により生成されます。軸の移動などの機能要求は、機能インタフェースを介して PLC からNC に送信されます。

参照先

- PLC 基本プログラムの説明
  →機能マニュアル、基本機能、PLC 基本プログラム(P3)
- イベント起動型信号交信の説明(補助機能と G 命令)
  →機能マニュアル、基本機能、補助機能出力(H2)
- すべてのインタフェース信号、機能とデータインタフェースの一覧
  →リストマニュアル 2

2.2.2 PLC への準備完了信号

DB10 DBX104.7 (NC-CPU 準備完了)

NC CPU は準備完了であり、そのことを PLC に周期的に通知します。

DB10 DBX108.3 (オペレーティングソフトウェア準備完了)

SINUMERIK Operate は準備完了であり、そのことを NC に周期的に通知します。

DB10 DBX108.5 (ドライブがサイクリック運転)

必要条件:NC のすべての機械軸に対して、関連するドライブがサイクリック運転を行うこと。つまり、NC と PROFIdrive メッセージを周期的に交換すること。

DB10 DBX108.6 (ドライブ準備完了)

必要条件:NC のすべての機械軸に対して、PROFIBUS 経由で関連するドライブだけでなく他社製ドライブも準備完了であることが通知されること。

DB31、... DBX93.5 == 1 (ドライブ準備完了)

DB10 DBX108.7 (NC 準備完了)

NC が準備完了です。

2.2.3 PLC への状態信号

DB10 DBX103.0 (リモート診断有効)

HMI コンポーネントは、リモート診断(オプション)が有効であること、すなわち、外部PCによって制御が行われることを PLC に報告します。

DB10 DBX109.6 (周囲温度アラーム)

周囲温度またはファン監視機能が異常を検出しました。

DB10 DBX109.7 (NC バッテリアラーム)

バッテリ電圧が下限値未満に低下しました。コントローラはまだ操作できます。コントロールシステムのシャットダウンや電源電圧の低下は、データ消失の原因になります。

DB10 DBX109.0 (NC アラーム未処置)

少なくとも 1 つの NC アラームが未処置であることを NC が通知しています。チャネル別のインタフェースを調べると、どのチャネルが関係するのかと、このアラームが加工停止の原因になるかどうかを確認できます。

DB21、... DBX36.6 (チャネル別 NC アラームが未処置)

NC はこの信号を PLC に送信して、少なくとも 1 つの NC アラームが、対応するチャネルに対して未処置であることを通知します。

DB21、... DBX36.7 (加工停止を伴う NC アラーム未処置)

NC は PLC にこの信号を送信して、現在のパートプログラム処理を中断または中止した少なくとも 1 つの NC アラームが、関連するチャネルに対して有効であることを通知します。

DB21、... DBX39.1 (プログラムストップを伴う NC アラーム)

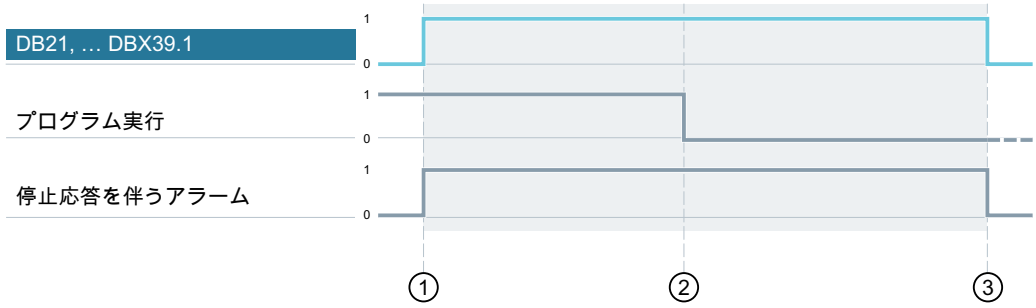
インタフェース信号 DB21、... DBX39.1 (プログラムストップを伴う NC アラーム)を使用して、NC は、対応するチャネルに対して、プログラムの進行を妨げている少なくとも 1 つの NC アラームが未処置であることを PLC に通知します。

この信号を起動したアラーム応答が有効でなくなると、この信号は直ちにリセットされます。これは、アラームのリセット条件に依存します。

停止応答を伴うアラームの信号チャート

インタフェース信号 DB21、... DBX39.1 は、停止応答を伴うアラームが発生すると、常に設定されます。以下に示す、さまざまな停止応答の区別は行われません。

- 軌跡上で停止し、信号「NC 準備完了」(DB10 DBX108.7)をリセット。
- モードグループのすべての軸を停止し、信号「モードグループ準備完了」(DB11 DBX6.3)をリセット。
- 軌跡上で即時停止。
- インタープリタが停止し、IPO バッファの内容が処理される。
- ブロック終点で停止。



① 停止応答を伴うアラームが発生します。
対応するチャネルに対して NC/PLC インタフェース信号 DB21、... DBX39.1 (プログラムストップを伴う NC アラーム)が設定されます。

② このアラームにより、プログラムの実行が停止されます(加工停止)。

③ アラームがリセットされると、インタフェース信号 DB21、... DBX39.1 もリセットされます。

図 2-1 停止応答を伴うアラームの信号チャート

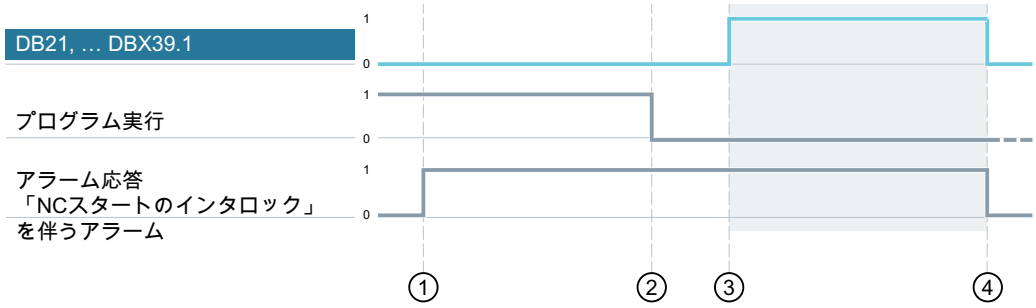
アラーム応答「NC スタート禁止」を伴うアラームの信号チャート

アラーム応答「NC スタート禁止」を伴うアラームも考慮されます。

この応答を伴うアラームは、以下の処理の実行を妨げます。

- パートプログラムの起動/ティーチングの起動/[NC スタート]キーによるジョグ移動の継続
- 別のチャネルのパートプログラムからのパートプログラムの起動
- (JOG または AUTOMATIC モードでの)「リセット」状態からの ASUB の起動
- JOG モードでの「中断」状態からの ASUB の起動
- 正常に完了した検索後の多段検索
- リセット状態からの検索(検索タイプ 1,2,4)の起動
- 「中断」状態からの検索(検索タイプ 1,2,4)の起動
- SERUPRO (検索タイプ 5)の起動
- シミュレーション検索(検索タイプ 102)の起動

アラーム応答「NC スタート禁止」を伴う未処置のアラームに応じてこれらの処理のいずれかが中止された場合、インタフェース信号 DB21.DBX39.1 が設定されます。



① アラーム応答「NC スタート禁止」を伴うアラームが発生します。

② NC プログラムが最後まで実行されます。

③ NC スタート/ASUB の起動/検索がトリガされます。アラームがまだクリアされていないため、対応するチャネルに対して NC/PLC インタフェース信号 DB21、... DBX39.1 (プログラムストップを伴う NC アラーム)が設定されます。

アラームがリセットされると、インタフェース信号 DB21、... DBX39.1 もリセットされます。

図 2-2　アラーム応答「NC スタート禁止」を伴うアラームの信号チャート

注記

ASUB の起動を許可

マシンデータ MD20194 $MC_IGNORE_NONCSTART_ASUP を使用すると、アラーム応答「NC スタート禁止」を伴うアラームが未処置である場合でも、割り当て済みの ASUB が起動されるように割り込み入力をパラメータ設定できます(「ユーザーアラーム用の ASUB 起動の実行 (ページ 720)」の章を参照)。この場合、インタフェース信号 DB21、... DBX39.1 (プログラムストップを伴うアラーム)は設定**されません**。

2.2.4　操作パネルへ/からの信号

DB19 DBX0.0 (画面表示)

画面消去を解除します。

DB19 DBX0.1 (画面消去)

操作パネルの画面を消します。

インタフェース信号により画面が消されている場合、以下の点に注意してください。

- キーボードで画面を再度表示させることはできなくなります(以下を参照してください)。
- 操作パネル上での最初のキーストロークだけで、オペレータ操作が実行されます。

注記

インタフェース信号を介して画面が消されているときに、偶発的なオペレータ操作を防止するために、**同時に**キーボードを無効にすることを推奨します。

DB19 DBX0.1 = 1 かつ DB19 DBX0.2 = 1 (キーロック)

キーボード/自動スクリーンセーバーによる画面消去

下記の割り当て時間(初期設定 = 3 分)内に操作パネルでボタンが押されなかった場合:
MD9006 $MM_DISPLAY_SWITCH_OFF_INTERVAL (画面消去までの時間)、
画面は自動的に消されます。

消された後で最初にボタンが押されたときに、画面は再度表示されます。画面を表示するためにボタンを押しても、オペレータ操作は実施されません。

パラメータ設定:

- DB19 DBX0.1 = 0
- MD9006 $MM_DISPLAY_SWITCH_OFF_INTERVAL > 0

DB19 DBX0.2 (キーロック)

接続されたキーボードを介したすべての入力が禁止されます。

DB19 DBX 0.3 / 0.4 (キャンセルアラームの解除/リコールアラームの解除)

キャンセルまたはリコールの解除条件を持つ現在未処置のすべてのアラームを解除するための要求。アラームの解除は、次のインターフェース信号を介して通知されます。

- DB19 DBX20.3 (キャンセルアラーム解除済み)
- DB19 DBX20.4 (リコールアラーム解除済み)

DB19 DBX0.7 (WCS/MCS の現在値)

機械座標系とワーク座標系の間の現在値表示の切り替え

- DB19 DBX0.7 = 0:機械座標系(MCS)
- DB19 DBX0.7 = 1:ワーク座標系(WCS)

DB19 DBB13 (ファイル転送の制御)

PLC のプログラム選択を制御するための命令バイト。以下のインタフェース信号によって指定されたプログラムに関連する命令:

- DB19 DBX16.0-6 (PLC からのプログラム選択:プログラムリストのインデックス)
- DB19 DBB17 (PLC からのプログラム選択:プログラムリストのプログラムインデックス)

DB19 DBB16 (PLC からのプログラム選択:プログラムリストのインデックス)

プログラムリストを指定するための制御バイト。

DB19 DBB17 (PLC からのプログラム選択:プログラムリストのプログラムインデックス)

プログラム番号を指定するための制御バイト。

DB19 DBB26 (PLC からのプログラム選択:状態信号)

現在のデータ送受信ステータスのステータスバイト。

DB19 DBB27 (PLC からのプログラム選択:異常 ID)

データ転送の異常値の出力バイト。

参照先

PLC のプログラム選択の詳細については、以下を参照してください:

- プログラムの選択 (ページ 1122)SINUMERIK 840D sl 用
- プログラムの選択 (ページ 1389)SINUMERIK 828D 用

2.2.5 チャネルへの信号

DB21、... DBX6.2 (残移動量削除)

インタフェース信号の立ち上がりにより、現在有効な軌跡加速度で、対応する NC チャネルにプログラムされた軌跡上に停止します。 その後、軌跡の残移動量が削除され、次のパートプログラムブロックへのブロック変更が可能になります。

2.2.6 軸/主軸への信号

DB31、... DBX1.0 (ドライブテスト移動許可)

通知
ドライブテストの指定 ドライブテスト中に作業者や機械に危険を及ぼさずに機械軸を移動できるようにするための適切な対策実施/適切なテストの実行は、工作機械メーカ/システムセットアップエンジニアそれぞれの責任です。

機械軸は「関数発生器」などの特別なテスト機能により移動する場合、移動のために、下記の信号によるドライブテスト用の許可が別に必要となります。

DB31、... DBX61.0 = 1 (ドライブテスト移動要求)

移動は、許可されると、実行されます。

DB31、... DBX1.0 == 1 (ドライブテスト移動許可)

DB31、... DBX1.3 (軸/主軸無効)

停止軸

インターフェース信号が停止軸に設定されている場合は、この時点以降、すべての移動要求が無視されます。

移動要求は保持されます。移動要求が未処置の状態で軸無効をキャンセル、つまり DB31、... DBX1.3 = 0 にすると、移動が実行されます。

移動軸

インターフェース信号が移動軸に設定されている場合は、軸が、現在有効なブレーキ特性を使用して停止します。移動要求が存在している間にインターフェース信号がリセットされた場合は、移動が継続されます。

主軸

- 開ループ制御モード:速度指令値 0 をすぐに出力
- 位置決めモード:前述の「停止軸」/「移動軸」を参照

DB31、... DBX1.4 (フォローアップモード)

インターフェース信号は、DB31, ... DBX2.1 (コントローライネーブル)インターフェース信号でのみ有効です。

DB31、... DBX2.1	DB31、... DBX1.4	モード
1	無効	閉ループ制御
0	1	フォローアップ
0	0	保持

フォローアップ

フォローアップモードでは、機械軸の指定位置を連続的にフィードバック位置に合わせます(指定位置 = フィードバック位置)。

フィードバック:DB31、... DBX61.3 == 1 (フォローアップ有効)

フォローアップモードでは、クランプ監視や停止状態監視は**無効**です。

注記

DB31、... DBX2.1 (コントローライネーブル)インターフェース信号がフォローアップモード時に設定され、最後にプログラムされた位置への再位置決めは、NC プログラムが有効なときに REPOSA (すべての軸による直線でのアプローチ)で実行されます。その他の場合には、すべての後続の移動は実位置から開始されます。

保持

「保持」モードでは、機械軸の指定位置はフィードバック位置に従いません。機械軸を指定位置から遠ざけた場合は、指定位置とフィードバック位置の差が広がり続けます(追従誤差)。軸加速特性に適合しないコントローライネーブルが設定されている場合は、追従誤差が突然、修正されます(速度ジャンプ)。

フィードバック:DB31、... DBX61.3 == 0 (フォローアップ有効)

「保持」の間、クランプ監視や停止状態監視は**有効**です。

注記

軸加速特性に適合しないコントローライネーブルが設定されている場合は、追従誤差が突然、修正されます(速度の不連続変化)。

用途例

クランプ後に「コントローライネーブル」が設定された場合の機械軸 Y の位置決め応答。クランピングにより、機械軸が現在位置 Y_1 からクランプ位置 Y_k に動かされたものとします。

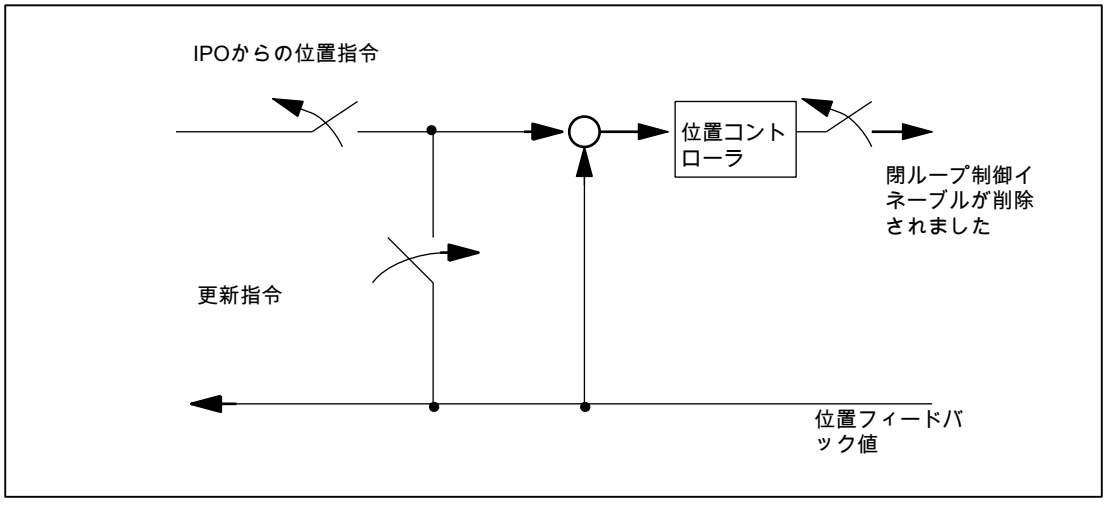


図 2-3　コントローライネーブルとフォローアップモードの働き

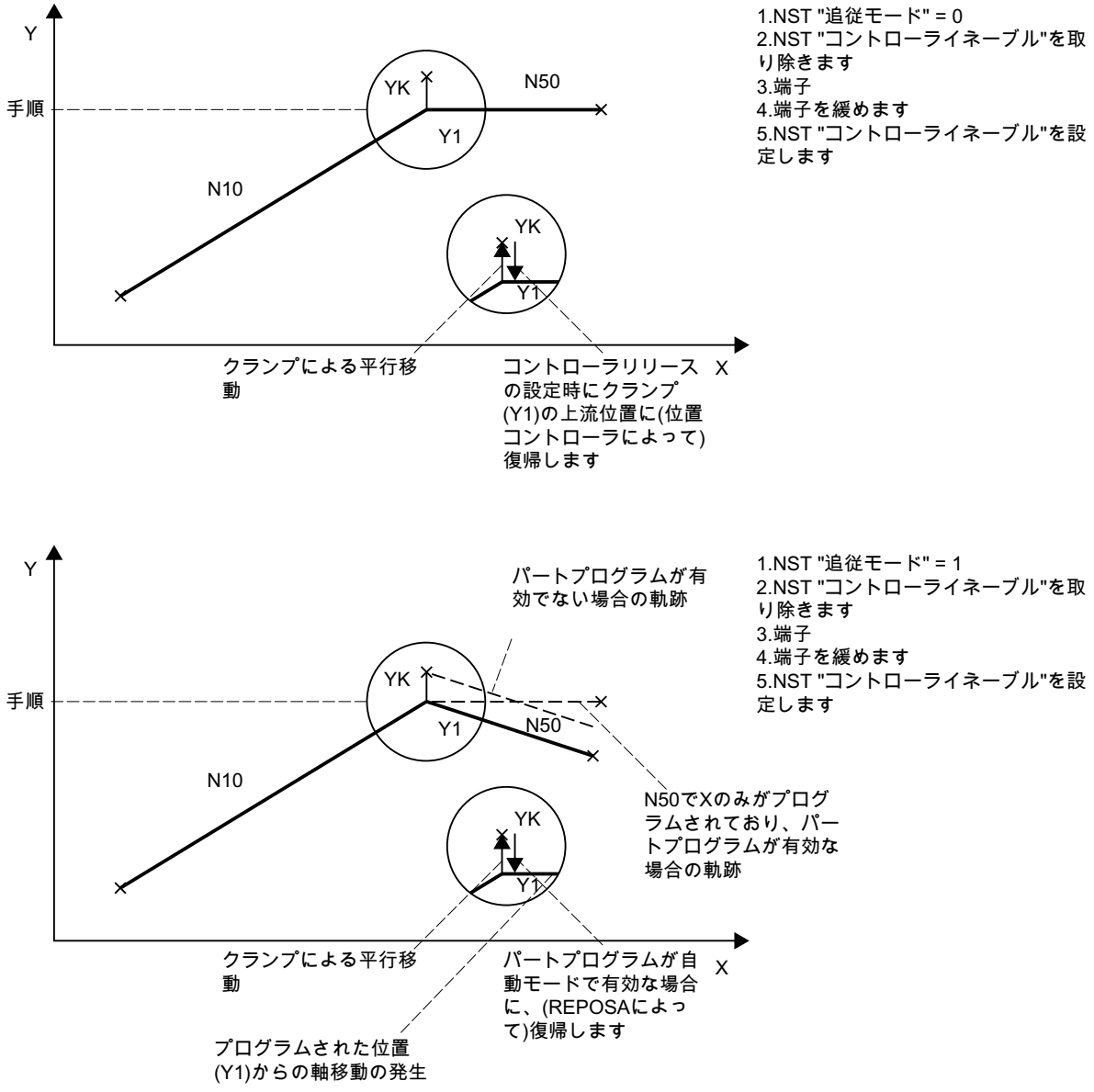


図 2-4　クランプと「保持」による軌跡

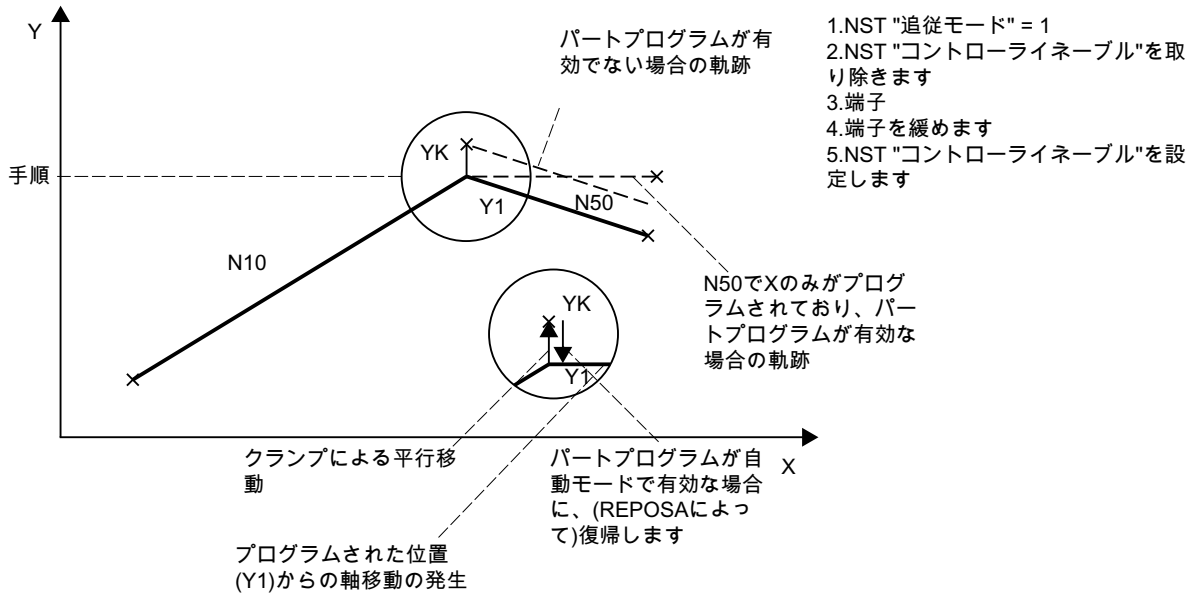


図 2-5　クランプと「フォローアップ」による軌跡

アナログ指令インタフェースを備えたドライブ

アナログ指令インタフェースを備えるドライブは、外部指令値を使用して機械軸を移動できます。機械軸で「フォローアップモード」が設定されている場合は、実位置が引き続き取得されます。フォローアップモードがキャンセルされた後は、原点確立は必要ありません。

以下の手順を推奨します。

1. フォローアップモードの起動
   DB31、... DBX2.1 = 0 (コントローライネーブル)
   DB31、... DBX1.4 = 1 (フォローアップモード) (同一または前の OB1 周期で)
   → 軸/主軸はフォローアップモードで動作しています。
2. 外部コントローライネーブルと外部速度指令値を無効にします。
   → 軸/主軸は、外部指令値を使用して移動します。
   → NC はフィードバック位置の検出を継続し、指令位置をフィードバック位置に合わせます。
3. 外部コントローライネーブルを無効にし、外部速度指令値をキャンセルします。
   → 軸/主軸は停止します。
4. フォローアップモードのキャンセル
   DB31、... DBX2.1 = 1 (コントローライネーブル)
   DB31、... DBX1.4 = 0 (フォローアップモード)
   → NC はフィードバック位置に一致させます。この位置から次の移動動作が開始します。

注記

「フォローアップモード」は「コントローライネーブル」と組み合わせた場合にのみ有効になるため、キャンセルする必要はありません。

フォローアップモードのキャンセル

フォローアップモード中に有効な検出器の許容できる最大のエンコーダ限界周波数を超えなかった場合は、フォローアップモードがキャンセルされたときに、機械軸を再度原点確立する必要はありません。エンコーダ限界周波数を超えた場合、コントローラによりこれが検出されます。

- DB31、... DBX60.4 / 60.5 = 0 (原点確立済み/原点同期済み 1/2)
- アラーム:「21610 エンコーダ周波数を超えました」

注記

有効な座標変換(TRANSMIT など)の一部である機械軸に対して「フォローアップモード」が解除された場合、これにより座標変換に関係する他の機械軸の再位置決め(REPOS)の一環としての移動を生成できます。

監視機能

機械軸がフォローアップモードの場合、以下の監視メカニズムは機能しません。

- 停止状態監視
- クランプ監視
- 位置決め監視

他のインタフェース信号に対する影響

- DB31、... DBX60.7 = 0 (精密イグザクトストップ位置に到達)
- DB31、... DBX60.6 = 0 (汎用イグザクトストップ位置に到達)

DB31、... DBX1.5/1.6 (位置検出器 1/2)

1 つの機械軸に 2 つの検出器を接続できます。たとえば、

- 間接モータ検出器
- 負荷側の直接検出器

一度に有効にできる検出器は 1 つだけです。機械軸に関するすべての閉ループ制御、位置決め運転などは、常に有効な検出器に対応しています。

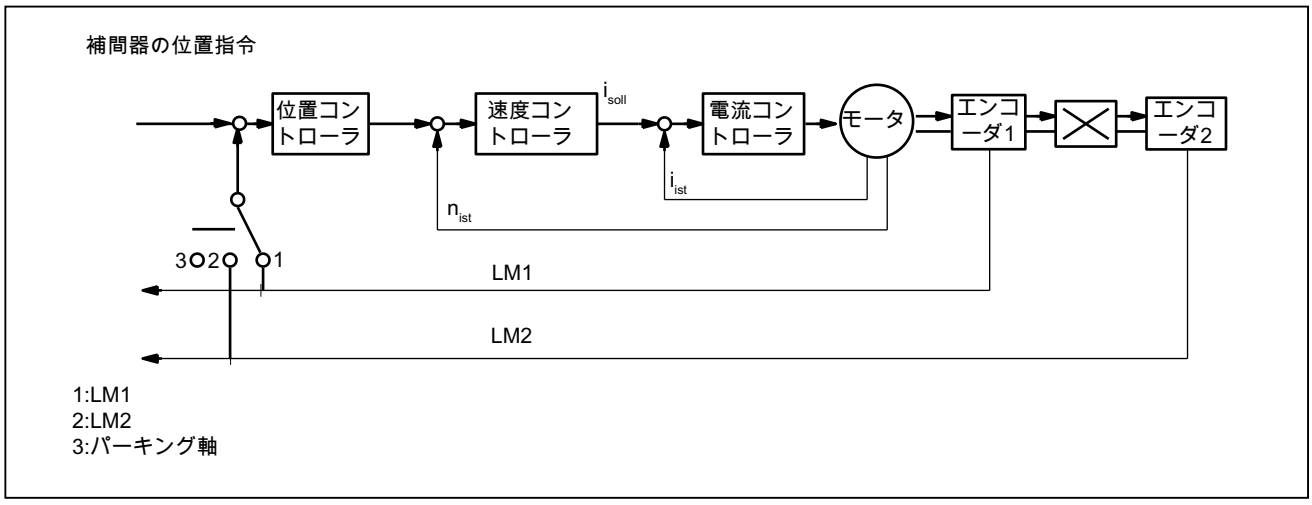


図 2-6　位置検出器 1 と 2

「コントローライネーブル」と組み合わせた「位置検出器 1 / 2」インターフェース信号の機能

DB31、... DBX1.5	DB31、... DBX1.6	DB31、... DBX2.1	機能
1	任意	1	位置検出器 1 有効
0	1	1	位置検出器 2 有効
0	0	0	「パーキング」有効
0	0	1	位置検出器のない主軸(速度制御)
1 → 0	0 → 1	1	切り替え:位置検出器 1 → 2
0 → 1	1 → 0	1	切り替え:位置検出器 2 → 1

DB31、... DBX2.1 (コントローライネーブル)

コントローライネーブルを設定すると、機械軸の位置制御ループが接続されます。機械軸は位置制御モードになります。

DB31、... DBX2.1 == 1

コントローライネーブルをキャンセルすると、機械軸位置制御ループが切断されて、少し遅れて機械軸の速度制御ループが切断されます。

DB31、... DBX2.1 == 0

起動方法

機械軸の閉ループ制御イネーブルは、以下の影響を受けます。

- NC/PLC インタフェース信号:
  - DB31、... DBX2.1 (コントローライネーブル)
  - DB31、... DBX21.7 (パルスイネーブル)
  - DB31、... DBX93.5 (ドライブ準備完了)
  - DB10、DBX56.1 (非常停止)
- NC 内部
  機械軸のコントローライネーブルをキャンセルするアラーム。アラーム応答については以下を参照してください。
  参照先:
  『診断マニュアル』

機械軸が停止状態のときのコントローライネーブルのキャンセル

- 機械軸の位置制御ループが切断されます。
- DB31、... DBX61.5 == 0 (位置コントローラ有効)

機械軸が移動しているときのコントローライネーブルのキャンセル

機械軸が補間軌跡移動または連結の一部で、これに対するコントローライネーブルがキャンセルされた場合、関連するすべての軸が急停止により停止し(速度指令値 = 0)、アラームが表示されます。

アラーム:「21612 移動中のコントローライネーブルのリセット」

- 機械軸は、下記のパラメータに設定された、急停止による異常状態に対する減速カーブの持続時間を考慮して減速されます(速度指令値 = 0)。
  MD36610 $MA_AX_EMERGENCY_STOP_TIME (異常時の減速カーブの最大持続時間)
  アラームが表示されます。
  アラーム:「21612 移動中のコントローライネーブルのリセット」

  注記

  コントローライネーブルは、遅くとも下記の遮断遅れの時間が経過したときにキャンセルされます。
  MD36610 $MA_AX_EMERGENCY_STOP_TIME

- 機械軸の位置制御ループが切断されます。インタフェース信号を介したフィードバック:
  DB31、... DBX61.5 == 0 (位置コントローラ有効)
  下記のマシンデータに設定されたコントローライネーブルの遮断遅れの時間が、開始されます。
  MD36620 $MA_SERVO_DISABLE_DELAY_TIME (コントローライネーブルの遮断遅延)
- 実速度がゼロ速度範囲に到達するとすぐ、ドライブのコントローライネーブルがキャンセルされます。インタフェース信号を介したフィードバック:
  DB31、... DBX61.6 == 0 (速度コントローラ有効)
- 機械軸の位置フィードバック値が、コントローラにより継続して取得されます。
- 減速動作の終了時に、関連の NC/PLC インタフェース信号に関係なく、機械軸がフォローアップモードに切り替わります。停止状態監視とクランプ監視は無効です。下記インタフェース信号については、前述の記述を参照してください。
  DB31、... DBX1.4 (フォローアップモード)

フィードバック値の原点同期(レファレンス点復帰)

機械軸が位置制御モードでなかったときに検出器の許容できる最大の限界周波数を超えなかった場合、コントローライネーブルが設定されたときに、機械軸のフィードバック位置を再度原点同期(レファレンス点復帰)する必要はありません。

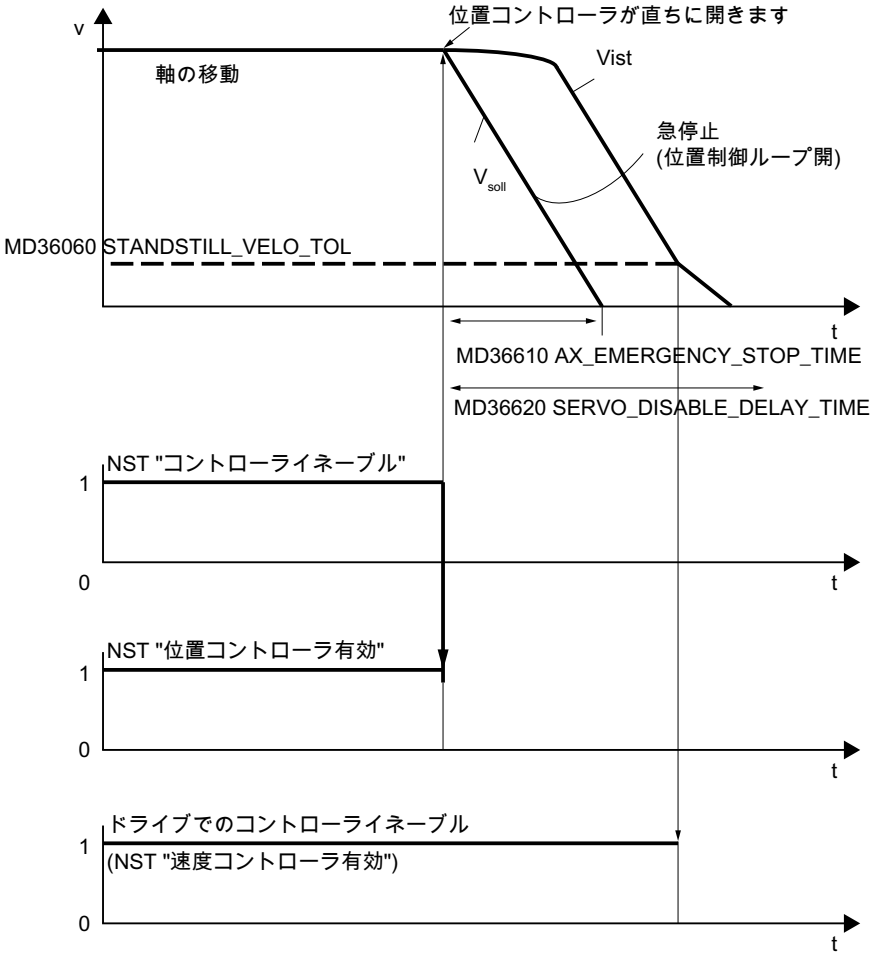


図 2-7　機械軸が移動しているときのコントローライネーブルのキャンセル

DB31、... DBX2.2 (残移動距離/主軸リセット(軸/主軸別))

「残移動距離削除」は、位置決め軸と組み合わせて AUTOMATIC および MDI モードでのみ有効になります。位置決め軸は、現在のブレーキ特性に従って減速停止します。軸の残移動距離が削除されます。

主軸リセット

主軸リセットの詳しい説明については、「S1:主軸 (ページ 1555)」の章を参照してください。

DB31、... DBX9.0 / 9.1 / 9.2 (制御パラメータセット)

指定した制御パラメータセットの起動の要求。

制御パラメータセット	DBX9.2	DBX9.1	DBX9.0
1	0	0	0
2	0	0	1
3	0	1	0
4	0	1	1
5	1	0	0
6	1	0	1
7	1	1	0
8	1	1	1

パラメータセット切り替えは、下記のマシンデータによって有効にしてください(主軸には必要ありません)。

MD35590 $MA_PARAMSET_CHANGE_ENABLE = 1 または 2

パラメータセット切り替えについての詳細は、「ギヤ選択切替中のパラメータセット選択 (ページ 1619)」の章を参照してください。

機械軸が移動しているときのパラメータセット切り替え

パラメータセット切り替えに対する応答は、閉ループ制御回路ゲイン係数 K_V において結果として発生する変化に従います。

MD32200 $MA_POSCTRL_GAIN (サーボゲイン係数)

- 「同じサーボゲイン係数(K_V)」または「位置制御が無効」の場合
  NC は、パラメータセット切替を直ちに検出します。パラメータセットは、移動中に変更されます。
- 「等しくないサーボゲイン係数(K_V)」または「位置制御が無効」の場合
  相対的に加々速度一定の切替を維持するために、軸が"停止"するまで、つまり割り当てられた停止状態の速度に達するか、それを超えるまで、パラメータセットの切替は実行されません。
  DB31、... DBX61.4 = 1 (軸/主軸停止)
  MD36060 $MA_STANDSTILL_VELO_TOL (「軸/主軸停止」検出速度/回転数)

パートプログラムからのパラメータセット切り替え

パートプログラムからのパラメータセット切り替えについては、ユーザー(工作機械メーカ)が、対応するユーザー専用の補助機能を定義し、PLC ユーザープログラムで処理して

ください。そして、PLC ユーザープログラムにより、対応するパラメータセットの切り替え要求を設定します。

補助機能出力についての詳細は、「H2: PLC への補助機能出力 (ページ 475)」の章を参照してください。

DB31、... DBX9.3 (NC によるパラメータセット指定無効化)

インターフェース信号を指定した場合、パラメータセットの切替要求(DB31、... DBX9.0/9.1/9.2)は無視されます。

2.2.7 軸/主軸からの信号

DB31、... DBX61.0 (ドライブテスト移動要求)

機械軸は「関数発生器」などの特別なテスト機能により移動する場合、移動のために、下記の信号によるドライブテスト用の許可が別に必要となります。

DB31、... DBX61.0 == 1 (ドライブテスト移動要求)

移動は、許可されると、実行されます。

DB31、... DBX1.0 == 1 (ドライブテスト移動許可)

DB31、... DBX61.3 (フォローアップ有効)

機械軸はフォローアップモードになります。

DB31、... DBX61.4 (軸/主軸停止($n < n_{min}$)

以下の場合、NC により「軸/主軸停止」が設定されます。

- 出力すべき新たな指令がない、**かつ**
- 機械軸の実速度が、下記のパラメータに設定されたゼロ速度未満である。
  MD36060 $MA_STANDSTILL_VELO_TOL (軸/主軸停止検出速度)

DB31、... DBX61.5 (位置コントローラ有効)

機械軸位置制御ループが接続され、位置制御が有効です。

DB31、... DBX61.6 (速度コントローラ有効)

機械軸速度制御ループが接続され、速度制御が有効です。

DB31、... DBX61.7 (電流コントローラ有効)

機械軸電流制御ループが接続され、電流制御が有効です。

DB31、... DBX69.0 / 69.1 / 69.2 (パラメータセットサーボ)

下記の信号に応じた有効なパラメータセットコード:

DB31、... DBX9.0 / 9.1 / 9.2 (制御パラメータセット選択)

DB31、... DBX76.0 (潤滑パルス)

コントローラの電源投入/リセットの後は、信号状態は 0 (FALSE)です。

「潤滑パルス」は、機械軸が潤滑用に下記のパラメータに設定された移動距離を進むとすぐに、反転(エッジ変化)されます。

MD33050 $MA_LUBRICATION_DIST (PLC による潤滑の距離)

2.2.8 軸/主軸への信号(ディジタルドライブ)

DB31、... DBX21.0 ～ 4 (モータデータセット/ドライブデータセット:選択)

PLC は、ドライブに対する、新しいモータデータセットとドライブデータセットへの切り替え要求を発行します。

インターフェースは、以下を使用して柔軟に設定することができます。
DB31、... .DBX130.0 - 4 (「モータ/ドライブデータセットの切り替え (ページ 84)」の章を参照)

有効なモータデータセットとドライブデータセットが、以下を使用して示されます。
DB31、... DBX93.0 - 4

注記

メイン主軸ドライブ(Main Spindle Drive)

メイン主軸ドライブの場合、スターとデルタ結線の切り替えの際に有効なのは、モータデータセット 1 および 2 のみです。

- モータデータセット 1:スター
- モータデータセット 2:デルタ

DB31、... DBX21.5 (モータが選択されています)

PLC ユーザープログラムにより、この信号がドライブに送信され、モータ選択が完了したことが通知されます。その場合、駆動パルスがドライブにより許可されます。

DB31、... DBX21.6 (積分停止の要求、速度コントローラ)

PLC ユーザープログラムにより、ドライブに対して速度コントローラの積分器を無効にします。速度コントローラは、PI 制御から P 制御に切り替わります。

速度コントローラの積分停止が有効になったとき、特定の用途で補正が実行されることがあります(停止時に積分器が既に負荷を保持している場合など)。

以下を介したフィードバック信号 DB31、... DBX93.6 = 1 (積分停止、速度コントローラ)

DB31、... DBX21.7 (要求パルスイネーブル)

ドライブモジュールのパルスイネーブルは、以下のすべてのイネーブル信号(ハードウェアとソフトウェア)が処理待ちの場合のみ要求されます。

- トリガ装置イネーブル
- コントローラとパルスイネーブル
- パルスイネーブル(セーフオペレーティングストップ)
- 記憶されたハードウェア入力

- 指令値有効
- 「運転準備完了状態」
  - ドライブアラームなし(DClink1 エラー)
  - DC リンクが接続済み
  - 加速完了

以下を介したフィードバック信号:DB31、... DBX93.7 (パルスイネーブル中)

2.2.9 軸/主軸からの信号(ディジタルドライブ)

DB31、... DBX92.1 (ランプ関数発生器の無効化有効)

ドライブはランプ関数発生器の急停止が有効であることを PLC へアンサーバックします。ドライブは、加減速機能を使用しないで停止します(速度指令値が 0)。

DB31、... DBX93.0 ～ 4 (モータデータセット/ドライブデータセット:表示)

ドライブは、有効なモータデータセットとドライブデータセットを PLC にアンサーバックします。

インターフェースは、以下を使用して柔軟に設定することができます。
DB31、... .DBX130.0 - 4 (「モータ/ドライブデータセットの切り替え (ページ 84)」の章を参照)

モータデータセットとドライブデータセットの切り替え要求が、以下を使用して実行されます。DB31、... DBX21.0 - 4

DB31、... DBX93.5 (ドライブ準備完了)

ドライブが準備完了であることを示すアンサーバック信号。軸/主軸の移動に必要な条件が満たされています。

DB31、... DBX93.6 (積分停止、速度コントローラ)

速度コントローラ積分器が無効です。このため、速度コントローラが PI 制御から P 制御に切り替わりました。

DB31、... DBX93.7 (パルスイネーブル中)

ドライブモジュールに対するパルスイネーブルが有効です。この状態で軸/主軸を移動できます。

DB31、... DBX94.0 (モータ温度警告)

モータ温度が設定されたモータ温度警告レベル(ドライブパラメータ p0604)を上回っています。

後述の「DB31、... DBX94.1 (ヒートシンク温度警告)」の注記も参照してください。

DB31、... DBX94.1 (ヒートシンク温度警告)

電源ユニットのヒートシンクの温度が許容範囲を超えています。過熱が続くと、約 20 秒後にドライブは遮断されます。

注記

温度警告 DB31、... DBX94.0 および DBX94.1

インタフェース信号は、以下の周期的なドライブメッセージ信号から取得されます。

- ケース 1:メッセージワードでの温度警告
  - DB31、... DBX94.0 = MELDW、ビット 6(モータ過熱警告なし)
  - DB31、... DBX94.1 = MELDW、ビット 7(電源ユニットの熱過負荷警告なし)
- ケース 2:警告クラス B の警告(インタフェースモード「SIMODRIVE 611U」、p2038 = 1 の場合のみ)
  DB31、... DBX94.0 == 1 および DBX94.1 == 1、以下に該当する場合:
  周期的なドライブメッセージ、ステータスワード 1(ZSW1)、ビット 11/12 == 2(警告クラス B)

メッセージワードに特定の情報がない場合、警告クラス B の警告からインタフェース信号が取得されます。

アラームが表示されます。アラーム番号= 200.000 + アラーム値(r2124)

モータ温度監視設定の詳しい説明については、以下の資料を参照してください。

参照先:

- S120 試運転マニュアル、「試運転」の「SINAMICS モジュールの温度センサ接続」の章
- S120 機能マニュアル、「監視と保護機能」の章
- S120 リストマニュアル
  - MELDW、ビット 6 ≙ BO:r2135.14 → ファンクションダイアグラム:2548, 8016
  - MELDW、ビット 7 ≙ BO:r2135.15 → ファンクションダイアグラム:2548, 2452, 2456, 8016

DB31、... DBX94.2 (起動完了)

速度指令値の変更後に、速度フィードバック値が再度、パラメータで割り付けされた許容範囲内になります。これで、起動手順は完了です。

この後で負荷の変化などにより許容範囲を超える速度変動があっても、インタフェース信号には影響を及ぼしません。

DB31、... DBX94.3 ($|M_d| < M_{dx}$)

現在のトルク$|M_d|$の絶対値が、パラメータ設定されたトルク検出値 M_{dx}(トルク検出値 2、p2194)未満です。

トルク検出値は、現在の速度依存トルク制限の百分率(%)として設定されます。

DB31、... DBX94.4 ($|n_{act}| < n_{min}$)

速度フィードバック値 n_{act} が n_{min} 未満です(速度検出値 3、p2161)。

DB31、... DBX94.5 ($|n_{act}| < n_x$)

速度フィードバック値 n_{act} が n_x 未満です(速度検出値 2、p2155)。

DB31、... DBX94.6 ($n_{act} = n_{set}$)

速度フィードバック値は速度設定値の許容範囲内(p2163)です。

DB31、... DBX95.7(警告クラス C の警告が未処置)

ドライブは警告クラス C の警告が未処置であることを通知します。

2.3 各種機能

2.3.1 過負荷の場合の画面の更新動作-840D sl のみ

先読み運転で新しいブロックが提供されるまでメインランを待機させるパートプログラムがあります。

先読み運転と画面の更新の両方に NC のプロセッサ時間が必要になります。

以下のマシンデータを使用して、先読み運転に時間がかかりすぎる場合の NC の動作を設定できます。

MD10131 $MN_SUPPRESS_SCREEN_REFRESH

値	意味
0	チャネルの先読み運転に時間がかかりすぎる場合は、すべてのチャネルで画面が更新されません。
1	チャネルの先読み運転に時間がかかりすぎる場合は、時間的に厳しいチャネルでのみ画面が更新され、先読み運転にプロセッサ時間を使用します。
2	画面の更新は抑制されません。

2.3.2 インボリュート補間の設定- 840D sl のみ

はじめに

円のインボリュートは、円からほどける「一本の糸」の先端により描かれる曲線です。インボリュート補間により、インボリュート曲線に沿った移動が可能です。

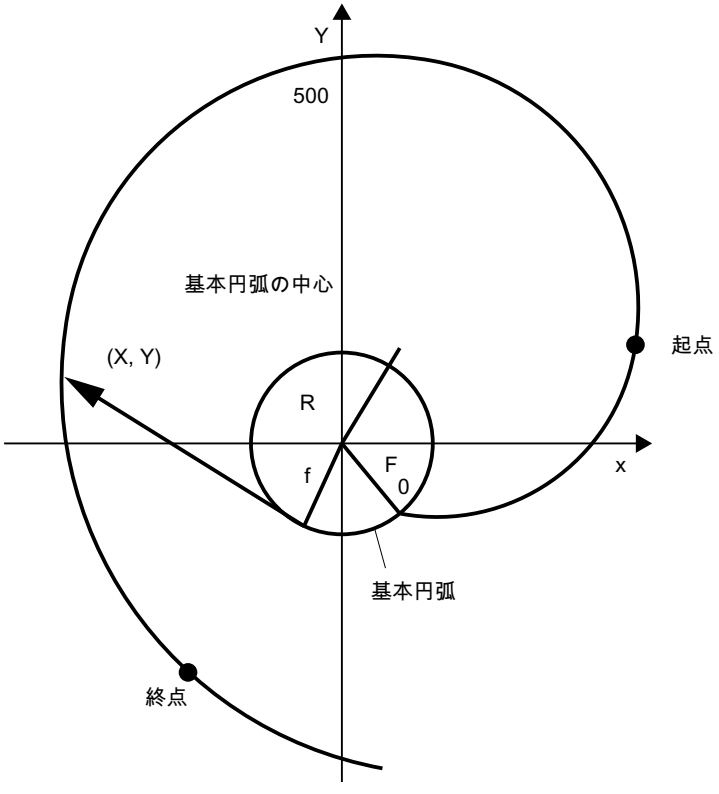


図 2-8 インボリュート曲線(基礎円からほどいた)

プログラミング

インボリュート補間のプログラム指令方法の概要については、以下に説明があります。

参照先:
『プログラミングマニュアル、基本編』

プログラム指令パラメータに加えて、マシンデータがインボリュート補間の 2 つの事例に関連しています。このデータは、工作機械メーカー/エンドユーザーにより設定する必要があります。

精度

プログラム指令終点が開始点により定義されたインボリュート曲線上に正確に配置されていない場合、開始点と終点により定義された 2 つのインボリュート曲線の間で補間が実行されます(下図を参照してください)。

終点の誤差の最大値は、マシンデータにより決定されます。

MD21015 $MC_INVOLUTE_RADIUS_DELTA(インボリュート曲線の終点監視)

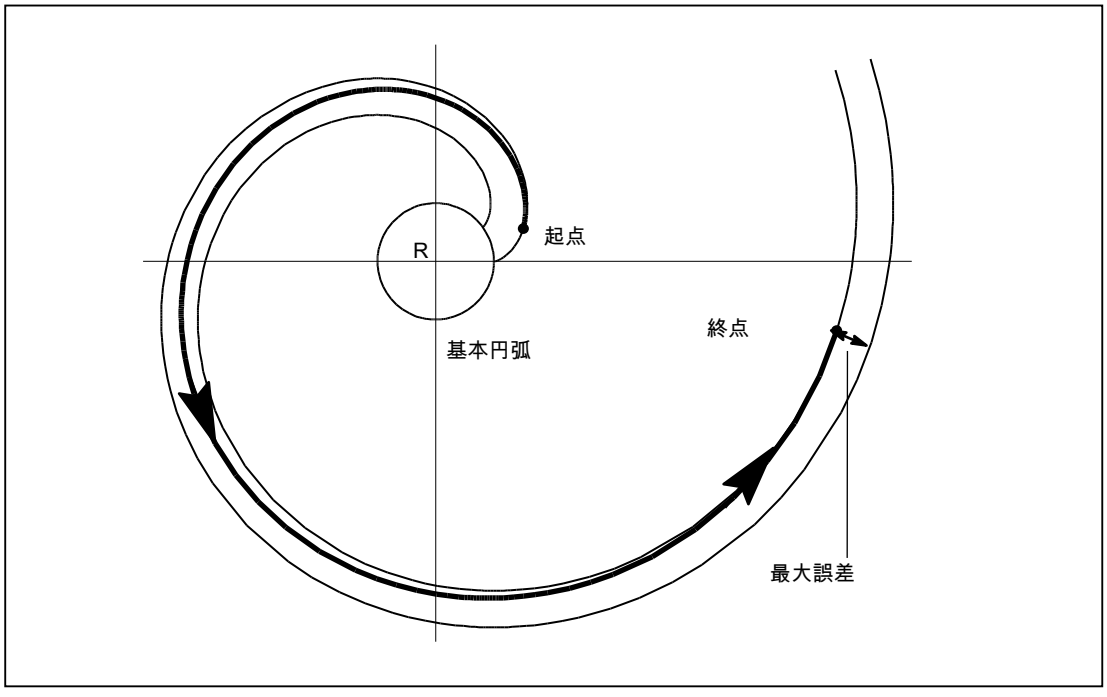


図 2-9　MD21015 は、許容される誤差の最大値を指定します。

制限角度

基礎円に向かうインボリュート曲線をプログラム指令するために AR が使用され、その回転角度が最大許容値より大きい場合、アラームが出力され、プログラムの実行が中止されます。

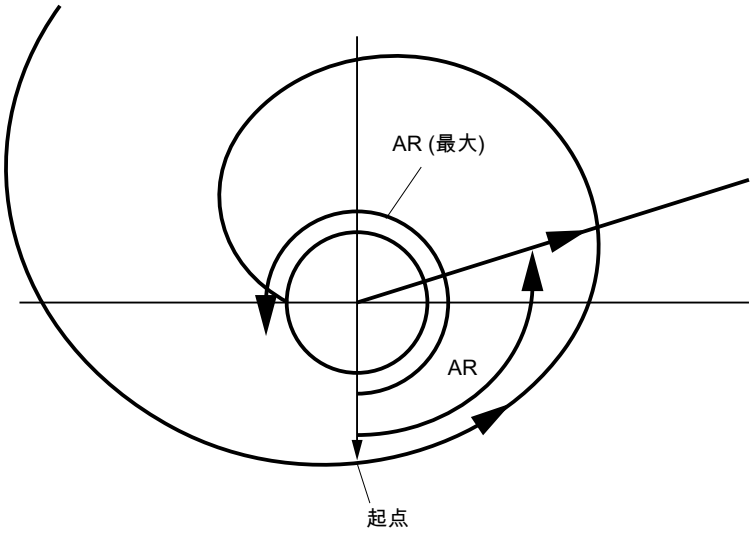


図 2-10　基礎円に向かう回転の制限角度

以下のパラメータ設定を使用して、アラーム表示をマスクできます。

MD21016 $MC_INVOLUTE_AUTO_ANGLE_LIMIT = TRUE (インボリュート補間用自動角度制限)

プログラム指令の回転の角度は自動的に制限され、補間軌跡はインボリュート曲線が基礎円と交わる点で終了します。 これは、たとえば、基礎円の外側の点から出発して、基礎円上で終了するインボリュート曲線のプログラム指令を容易にします。

工具径補正

2-1/2 D 工具径補正は、インボリュート曲線に使用できる唯一の工具径補正機能です。3D 工具径補正が有効の場合(外周と正面の両方のフライス削り)で、インボリュート曲線がプログラム指令されると加工がアラーム 10782 により中断します。

2-1/2D 工具径補正を使用する場合、インボリュート曲線の平面は補正平面に配置してください。そうしないと、別のアラーム 10781 が発生します。 ただし、補正平面においてインボリュート曲線に対する追加のヘリカル要素のプログラム指令は可能です。

ダイナミック応答

基礎円上で開始または終了するインボリュート曲線は、この点での曲率は無限です。 工具径補正が有効のときに、この点で速度が適切に制限され、他の点で減速しすぎないようにするために、下記の MD で「速度制限プロファイル」機能を有効にしてください。

MD28530 $MC_MM_PATH_VELO_SEGMENTS > 1 (軌跡速度を制限するためのメモリ要素の数)

5 の設定が推奨されます。 比較的小さい範囲で変わる曲率半径を持つインボリュート曲線の一部が使用される場合、この設定をおこなう必要はありません。

2.3.3 初期設定メモリの起動- 840D sl のみ

GUD 開始値

The DEF... / REDEF...NC 命令を使用して、グローバルユーザーデータ(GUD)にデフォルト設定を割り当てることができます。電源投入後に、たとえば、属性 INIPO で、これらの初期値をパラメータ設定済初期化時に使用可能にするには、これらを固定値としてシステムに保存してください。次のマシンデータを使用して必要なメモリ領域を有効にしてください。

MD18150 $MM_GUD_VALUES_MEM (GUD 値用の不揮発性メモリ領域)

参照先:

- 『機能マニュアル、上級機能』; S7;「メモリ構成」
- プログラミングマニュアル、上級編

2.3.4 PLC 変数の読み取りおよび書き込み- 840D sl のみ

高速データチャネル

PLC と NC の間での情報の高速交信のために、メモリ領域がこれらのモジュールの通信バッファに予約されています(デュアルポート RAM)。任意のタイプ(I/O、DB、DW、フラグ)の変数をこのメモリ領域内で交信できます。

PLC は「ファンクションコール」(FC)を使用して、NC はシステム変数を使用して、このメモリにアクセスします。

メモリ領域の構成

このメモリ領域を構成する責任は、ユーザーのプログラムエンジニア(NC と PLC)にあります。

該当するデータ書式に応じた制限(すなわち、DWORD は 4 バイトの制限、WORD は 2 バイトの制限など)が選択されている限り、メモリ内の任意の保存位置を指定することができます。

メモリは、データタイプとメモリ領域内の位置オフセットを介してアクセスされます。

NC からのアクセス

システム変数を使用して、NC で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 PLC 変数に高速にアクセスすることができます。データの読み取り/書き込みは NC によって直接おこなわれます。データタイプはシステム変数の識別子になります。メモリ領域内の位置は、インデックスとしてバイト単位で指定されます。

システム変数	データタイプ	数値の範囲
$A_DBB[<インデックス>]	バイト(8 ビット)	0 <= x <= 255
$A_DBW[<インデックス>]	ワード(16 ビット)	-32768 <= x <= 32767
$A_DBD[<インデックス>]	ダブルワード(32 ビット)	-2147483648 <= x <= 2147483647
$A_DBR[<インデックス>]	浮動小数点(32 ビット)	$\pm(1.5 \cdot 10^{-45} <= x <= 3.4 \cdot 10^{38})$

PLC からのアクセス

PLC は、ファンクションコール(FC)を使用してメモリにアクセスします。データの読み取りと書き込みは、PLC 周期の開始時ではなく、FC により DDR で直ちにおこなわれます。データタイプとメモリ領域内の位置は、パラメータとして FC に伝送されます。

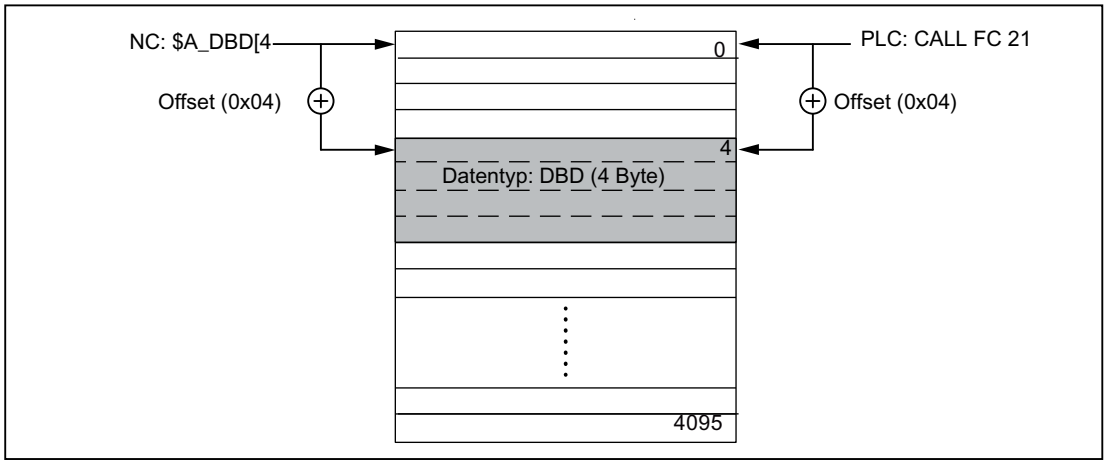


図 2-11　NC/PLC 通信のための通信バッファ(DPR)

必要条件

- DPR メモリ領域を構成する責任はユーザーにあります。設定の矛盾に対するチェックはおこなわれません。
- 入力と出力方向で、合計 4096 バイトが使用できます。
- 単一ビット操作はサポートされていません。ユーザーによりバイト操作に戻してください。
- 変数の内容は通信バッファで直接操作できるので、変数が数回に分けて参照されたり変数が連結されている場合の複数のアクセス操作の結果として、値に中間変化が発生することをユーザーは心に留めておいてください(すなわち、値を一時的にローカル変数または R 変数に格納するか、セマフォを組み込む必要があります)。
- ユーザーのプログラムエンジニアに、異なるチャネルからの通信バッファへのアクセスを調整する責任があります。
- データの一貫性は、最大 16 ビット(バイトとワード)のアクセス操作の場合のみ保証されます。ユーザーは、32 ビット変数(ダブルと実数)の一貫性のある伝送を確保する責任があります。このために、PLC で簡単なセマフォメカニズムを使用できます。

- PLC は、データを「リトルエンディアン」書式で DPR に格納します。
- $A_DBR によって転送される値は、データ換算の影響を受けるので、精度が落ちます。NC での浮動小数点数のデータ書式は DOUBLE (64 ビット)ですが、PLC では FLOAT (32 ビット)のみです。デュアルポート RAM での保管に使用される書式は、FLOAT です。デュアルポート RAM での保存の前/後それぞれで換算が実行されます。デュアルポート RAM で NC から変数への読み取り/書き込みアクセスがおこなわれた場合、換算は 2 度実行されます。データは両方の書式で保存されるため、読み取られた値と書き込まれた値の間の差をなくすことはできません。
  例
  「EPSILON」(微小偏差)での比較による問題の回避

プログラムコード

```
N10          DEF REAL DBR
N12          DEF REAL EPSILON = 0.00001
N20          $A_DBR[0]=145.145
N30          G4 F2
N40          STOPRE
N50          DBR=$A_DBR[0]
N60          IF ( ABS(DBR/145.145-1.0) < EPSILON ) GOTOF ENDE
N70          MSG ( "エラー" )
N80          M0
N90          END:
N99          M30
```

起動

同時に書き込み可能な出力変数の最大数は、以下の MD によって設定できます。
MD28150 $MC_MM_NUM_VDIVAR_ELEMENTS (PLC 変数書き込み用の要素数)

例

タイプ WORD の変数は、PLC から NC に伝送されます。

NC 入力(PLC 出力エリア)内の位置オフセットは、4 番目のバイトとなります。位置オフセットは、データ幅の整数倍にしてください。

PLC からの書き込み:

プログラムコード	コメント
. . .	
CALL FC21 (	
Enable :=M10.0,	;if TRUE, then FC21 active

プログラムコード	コメント
Funct :=B#16#4,	
S7Var :=P#M 104.0 WORD1,	
IVAR1 :=04,	
IVAR2 :=-1,	
Error :=M10.1,	
ErrCode :=MW12);	
. . .	
)	

パートプログラムでの読み取り

プログラムコード	コメント
. . .	
PLCDATA = $A_DBW[4];	; ワードを読み取り
. . .	

POWER ON、ブロック検索中の動作

DPR 通信バッファは、「POWER ON」時に初期化されます。

「ブロック検索」中に、PLC 変数出力が収集され、DPR 通信バッファにアプローチブロックとともに転送されます(アナログとディジタル出力の書き込みと同等)。

他の状態遷移は、この点に関して影響はありません。

参照先

PLC による FC 21 とのデータ交信に関する詳しい説明は、以下の資料にあります。

SINUMERIK 840D sl:「機能 (ページ 1322)」の章

2.3.5 パスワードおよびキースイッチを介したアクセス保護

アクセス権

機能、プログラムおよびデータへのアクセスは、ユーザーに応じて、8 階層の保護レベルを介して制御されます。 これらは、以下のものに細分されます。

- 当社、工作機械メーカおよびエンドユーザー用のパスワードレベル
- エンドユーザー用のキースイッチ位置

多階層のセキュリティコンセプト

アクセス権を規制するための多階層のセキュリティコンセプトが、パスワードレベルとキースイッチ設定の形で使用できます。

保護レベル	タイプ	ユーザー	アクセス対象(例)
0	パスワード	当社	すべての機能、プログラムおよびデータ
1	パスワード	工作機械メーカ: 開発	定義された機能、プログラムおよびデータ; 例: オプション入力
2	パスワード	工作機械メーカ: セットアップエンジニア	定義された機能、プログラムおよびデータ; 例: マシンデータの大部分
3	パスワード	エンドユーザー:サービス	割り付けられた機能、プログラムおよびデータ
4	キースイッチ位置 3	エンドユーザー:プログラマ、機械セットアップ担当	保護レベル 0 ～ 3 未満の場合; 工作機械メーカまたはエンドユーザーにより決定
5	キースイッチ位置 2	エンドユーザー: プログラミング知識のない熟練オペレータ	保護レベル 0 ～ 3 未満の場合; エンドユーザーにより決定
6	キースイッチ位置 1	エンドユーザー: プログラミング知識のない教育済みオペレータ	例: プログラム選択のみ、工具磨耗入力およびワークオフセット入力
7	キースイッチ位置 0	エンドユーザー:非熟練オペレータ	例: 入力およびプログラム選択は不可、 機械操作パネルのみ操作可能

アクセス保護の特徴

- 保護レベル 0 は最大のアクセス権、保護レベル 7 は最小のアクセス権を提供します。
- ある保護レベルに特定のアクセス権が付与された場合、これらのアクセス権はより高い保護レベルに自動的に適用されます。
- 逆に、特定の保護レベルのアクセス権は、より高いレベルからのみ変更できます。
- 保護レベル 0 ～ 3 のアクセス権は、当社により固定的に割り付けられ、変更はできません(初期設定)。
- アクセス権は、現在のキースイッチ位置の照会および入力されたパスワードの比較により設定できます。 パスワードが入力されると、キースイッチ位置のアクセス権が上書されます。
- オプションは、各保護レベルで保護できます。 ただし、オプションデータは保護レベル 0 および 1 だけで入力できます。
- 保護レベル 4 ～ 7 のアクセス権は単なる提案で、工作機械メーカおよびエンドユーザーにより変更できます。

2.3.5.1 パスワード

パスワードの設定

保護レベル(0 ～ 3)のパスワードは、HMI 操作画面を介して入力します。

操作エリア[セットアップ|パスワード|パスワードの設定]

パスワードの削除

パスワードを設定することで割り付けられるアクセス権は、そのパスワードの削除によって別途アクセス権が破棄されるまで有効です。

操作エリア[セットアップ|パスワード|パスワードの削除]

パスワードの変更

保護レベル(0 ～ 3)のパスワードは、HMI 操作画面を介して変更します。

操作エリア[セットアップ|パスワード|パスワードの変更]

注記

ウォーム再起動

アクセス権とパスワードの状態(設定/削除)は、ウォーム再起動によって影響を受けることはありません。

新しいパスワード

パスワードには、最大 8 文字まで使用できます。パスワードを選択する際は、操作パネルで入力できる文字だけを使用してください。

パスワードが 8 文字に満たない場合、不足の桁分は空白とみなされます。

新しいパスワードを割り当てる場合、安全なパスワードを割り当てるための規則を考慮する必要があります。

注記

安全なパスワードの割り当て

新しいパスワードを割り当てる場合には、次の規則を遵守します。

- 新しいパスワードを割り当てる場合、容易に推測できるパスワードを割り当てないでください(容易に推測できる簡単な単語、キーの組合わせなど)。
- パスワードには常に、数字、特殊文字、大文字と小文字の組合わせを含める必要があります。パスワードは、少なくとも 8 文字で構成する必要があります。PIN は、任意の数字シーケンスで構成する必要があります。
- 可能な場合は常に、IT システムによってサポートされる場合に、パスワードは可能なかぎり複雑な文字シーケンスであることが必要です。

安全なパスワードを割り当てるための規則の詳細については、情報セキュリティに関する政府機関にお問い合わせください(Bundesamt fur Sicherheit in der Informationstechnik、BSI) (https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/International/GSK_15_EL_EN_Draft.pdf?__blob=publicationFile&v=2)

パスワードを扱う場合、パスワード管理を支援するプログラムを使用できます。これによって、暗号化された形式でパスワードと PIN を保存および管理し、安全なパスワードを生成できます。

初期設定

標準では、保護レベル 1 ～ 3 で次のパスワードが有効です。

- 保護レベル 1:SUNRISE
- 保護レベル 2:EVENING
- 保護レベル 3:CUSTOMER

通知
コールド再始動時には、パスワードは初期設定にリセットされます コールド再始動(NC が電源投入し、標準マシンデータがロードされます)の後、保護レベル 1 ～ 3 のパスワードは初期設定に再度リセットされます。セキュリティ上の理由から、コールド再始動の後、パスワードの初期設定を変更することを強くお勧めします。

参照先

- 『SINUMERIK 840D sl 基本ソフトウェアおよび操作ソフトウェア』試運転マニュアル、セクション「SINUMERIK Operate (IM9)」>「一般的な設定」>「アクセスレベル」
- 試運転マニュアル『CNC 試運転:NC、PLC、Drive』、「セットアップの必要条件」セクション > 「電源オンおよび電源投入」および「アクセスレベル」

2.3.5.2 キースイッチ位置(DB10、DBX56.4 ～ 7)

キー操作スイッチ

キースイッチ位置は 4 つあり、それぞれの位置に保護レベル 4 ～ 7 が割り付けられています。キースイッチには色の異なる複数のキーがあり、それぞれ異なるスイッチ位置に設定できます。

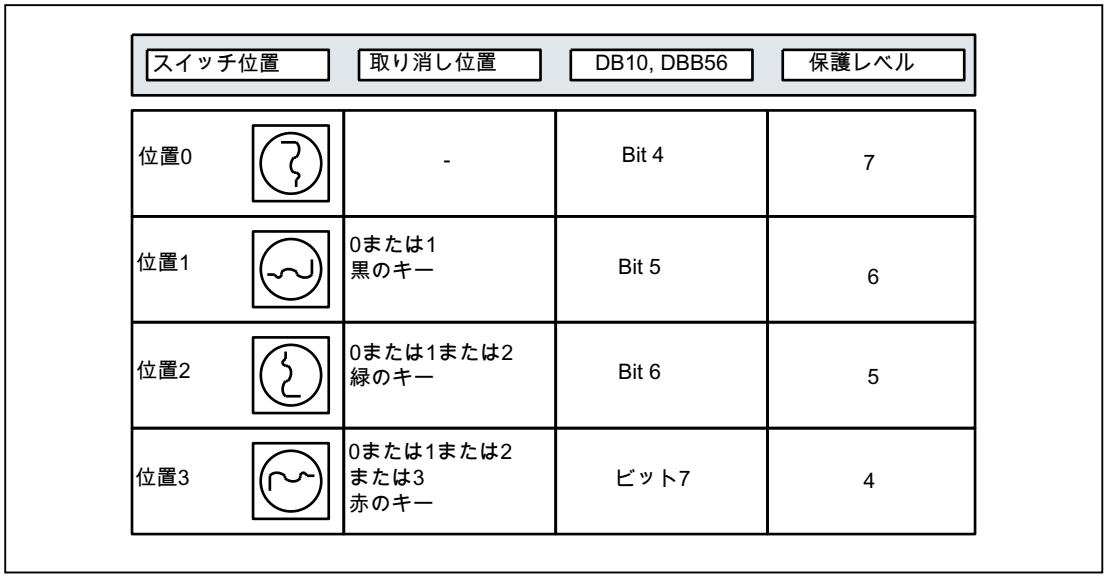


図 2-12 スイッチ位置 0 ～ 3

スイッチ位置

スイッチ位置 0 では、アクセス権が最小になります。スイッチ位置 3 では、アクセス権が最大になります。

DB10、DBX56.4 / .5 / .6 / .7 (スイッチ位置 0 / 1 / 2 / 3)

プログラム、データおよび機能に対する機械固有のアクセス許可を各スイッチ位置に割り付けることができます。詳細については、以下の資料を参照してください。

参照先

- CNC 試運転マニュアル:「NC、PLC、ドライブ、基本編」、
  セクション:保護レベルの基礎の章
- 『試運転マニュアル SINUMERIK Operate (IM9)』;「一般的な設定」、
  :アクセスレベルの章

PLC ユーザープログラムによる初期設定

キースイッチ位置は、PLC 基本プログラムによって NC/PLC インタフェースに伝送されます。対応するインタフェース信号は、PLC ユーザープログラムを使用して変更できます。これに関して、NC 側からすれば、常に 1 つのスイッチ位置のみ有効であることが望まれます(つまり、対応するインタフェース信号を 1 に設定します)。NC 側からみて複数のスイッチ位置が同時に有効である場合、スイッチ位置 3(つまり、最大のアクセス権を意味するキースイッチ位置)が NC によって内部的に有効になります。

2.3.5.3 パラメータ設定可能な保護レベル

パラメータ設定可能な保護レベル

いくつかの機能とデータ領域に対して保護レベルを自由に設定できます。 次のように、操作パネルのマシンデータを介して保護レベルを設定します。 $MM_USER_CLASS_<*機能_データ領域コード*>

例:

$MM_USER_CLASS_READ_TOA	工具オフセットの読み取り
$MM_USER_CLASS_WRITE_TOA	工具オフセットの書き込み
$MM_USER_CLASS_READ_PROGRAM	パートプログラムの読み取り
$MM_USER_CLASS_WRITE_PROGRAM	パートプログラムの書き込み/編集

初期値

少数の例外はあるものの、出荷時または標準セットアップ時には、保護レベルの初期値は 7(つまり最低保護レベル)に設定されます。

2.3.6 モータ/ドライブデータセットの切り替え

2.3.6.1 一般事項

モータとドライブデータセット

特定の加工状態への最適な調整のため、または機械構成が異なるために、モータ、インバータパラメータおよびエンコーダに対して 1 つのドライブで複数の異なるデータセットを使用できるようにする必要がある場合があります。ドライブオブジェクトの基本データセットの作成は、起動時に「ドライブウィザード」を使用しておこないます。

注記

参照先

試運転マニュアル:CNC:NC、PLC、ドライブ、「NC 制御ドライブの試運転」の章

後述のデータセットの複製と管理は、下記のユーザーインタフェースを介しておこないます。

SINUMERIK Operate:操作エリア[セットアップ|ドライブシステム|ドライブ装置|データセット]

特定の加工状態の機械軸に必要なモータデータセット(MDS)またはドライブデータセット(DDS)の起動は、PLC ユーザープログラムから下記のインタフェースを介しておこなってください。

軸の NC/PLC インタフェース

モータとドライブデータセットを切り替えるための軸の NC/PLC インタフェースのインタフェースは、3 つの領域に分割されます。

- フォーマットインターフェース (ページ 85)
- 要求インターフェース (ページ 86)
- 表示インターフェース (ページ 86)

2.3.6.2 フォーマットインターフェース

書式

フォーマットインターフェースを使用して、要求インターフェースと表示インターフェースのどのビットを使用して、モータデータセット(MDS)のアドレスを指定し、どれを使用してドライブデータセット(DDS)のアドレスを指定するかを指定します。

DB31、... DBX130.0- 4、ビット x = <値>

<値>	意味
0	モータデータセット(MDS)のビット位置または無効なビット位置
1	ドライブデータセット(DDS)のビット位置

モータデータセットと、ドライブ内のドライブデータセット

フォーマットは、モータデータセット(MDS)と、ドライブ内のドライブデータセット(DDS)の数によって異なります。この数は、以下のドライブパラメータを使用して決定できます。

- p0130 (モータデータセットの数)
- p0180 (ドライブデータセットの数)

インターフェースの有効範囲

電源投入後、必要なすべての情報をコントローラがドライブから受け取って NC で評価するとすぐに、要求インターフェース (ページ 86)と表示インターフェース (ページ 86)が有効として示されます。

DB31、... DBX130.7 == 1 (要求インターフェースと表示インターフェースが有効)

ドライブから伝送される情報がないか、互換性のない情報が伝送された場合、要求インターフェースと表示インターフェースは無効として表示されます。

注記

要求インターフェースと表示インターフェースが無効な状態でも、無効な要求インターフェースと表示インターフェースを介してデータセット切り替えをおこなう場合は、やはり、ユーザー/工作機械メーカの責任となります。

下記も参照

例 (ページ 87)

インターフェースの概要 (ページ 88)

2.3.6.3 要求インターフェース

新しいモータデータセット(MDS)とドライブデータセット(DDS)がインターフェースを介して要求されます。

DB31、... DBX21.0 - .4 = <MDS/DDS インデックス>

値の範囲

モータデータセットまたはドライブデータセット n (n = 1, 2, 3, ...)のアドレス指定は、そのインデックス i (i = n - 1 = 0, 1, 2, ...)に基づいて実行されます。

- モータデータセット:MDS[0, 1, 2, ... 15]
- ドライブデータセット:DDS[0, 1, 2, ... 31]

フォーマットインターフェース

要求インターフェースのフォーマット、つまり、どのビットを使用して、モータデータセット(MDS)のアドレスを指定し、どれを使用してドライブデータセット(DDS)のアドレスを指定するかは、フォーマットインターフェース (ページ 85)で指定します。

モータデータセットと、ドライブ内のドライブデータセット

モータデータセット(MDS)と、ドライブ内のドライブデータセット(DDS)の数は、以下のドライブパラメータを使用して決定できます。

- p0130 (モータデータセットの数)
- p0180 (ドライブデータセットの数)

主軸ドライブのモータデータセット(MDS)

主軸ドライブの場合は、以下の関連付けが適用されます。

- MDS[0]→スター運転
- MDS[1]→デルタ運転

2.3.6.4 表示インタフェース

有効なモータデータセット(MDS)とドライブデータセット(DDS)が、インターフェースを介して表示されます。

DB31、... DBX93.0 - .4 == <MDS/DDS インデックス>

値範囲と書式は、要求インターフェース (ページ 86)と同じです。

2.3.6.5 例

各モータデータセットについて、ドライブで 2 つのモータデータセット(MDS)と 2 つのドライブデータセット(DDS)を使用できます。これは図 2-13 モータ/ドライブデータセットの切り替えの原理 (ページ 89)に示す可能なデータセットの組合わせの「No.」:9 に対応します。

書式

ドライブデータセットの切り替え(DDS)のビット位置:

- DB31、... DBX130.0 == 1

モータデータセットの切り替え(MDS)のビット位置:

- DB31、... DBX130.1 == 0

無効なビット位置:

- DB31、... DBX130.2 == 0
- DB31、... DBX130.3 == 0
- DB31、... DBX130.4 == 0

ドライブデータセット(DDS)のインタフェース

要求と表示インタフェースの関連するビット位置:

- DB31、... DBX21.0 / DBX93.0
  - DB31、... DBX21.0 / DBX93.0 == 0 ⇒ 1 番目のドライブデータセット DDS[0]
  - DB31、... DBX21.0 / DBX93.0 == 1 ⇒ 2 番目のドライブデータセット DDS[1])

モータデータセット(MDS)のインタフェース

要求と表示インタフェースの関連するビット位置:

- DB31、... DBX21.1 / DBX93.1
  - DB31、... DBX21.1 / DBX93.1 == 0 ⇒ 1 番目のモータデータセット MDS[0]
  - DB31、... DBX21.1 / DBX93.1 == 1 ⇒ 2 番目のモータデータセット MDS[1])

無効なビット位置(MDS/DDS)

要求と表示インタフェースの無効なビット位置:

- DB31、... DBX21.1 / DBX93.2 == 0
- DB31、... DBX21.1 / DBX93.3 == 0
- DB31、... DBX21.1 / DBX93.4 == 0

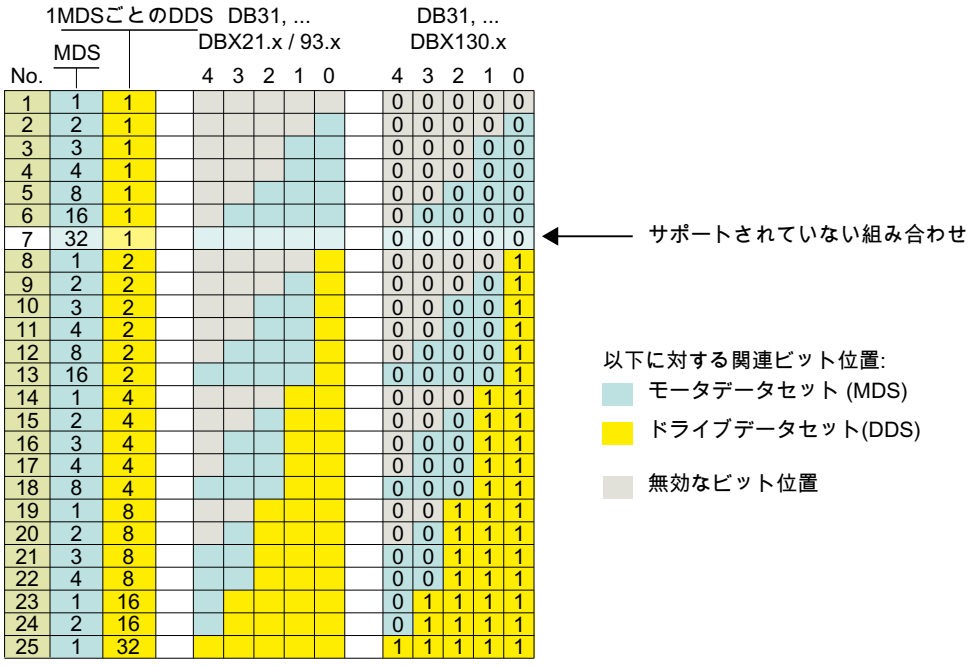
下記も参照

インターフェースの概要 (ページ 88)

2.3.6.6 インターフェースの概要

表 2-1 設定可能な MDS/DDS の組合わせ

MDS (モータ)の数	1MDS 毎の DDS (ドライブ)の数
1	1 ... 32
2	1, 2, 4, 8, 16
3	1, 2, 4, 8
4	1, 2, 4, 8
5	1, 2, 4
6	1, 2, 4
7	1, 2, 4
8	1, 2, 4
9	1, 2
10	1, 2
11	1, 2
12	1, 2
13	1, 2
14	1, 2
15	1, 2
16	1, 2



MDS	モータデータセットの数
1MDS 毎の DDS	モータデータセット毎のドライブデータセットの数
DB31、... DBX21.x	要求インターフェース
DB31、... DBX93.x	表示インタフェース
DB31、... DBX130.x	フォーマットインターフェース

図 2-13　モータ/ドライブデータセットの切り替えの原理

2.3.6.7　必要条件

「最後の」モータデータセットに対するドライブデータセットの可変数

「最後の」モータデータセットは、最上位の数またはインデックスを持つモータデータセットです。

一般的には、ドライブ内の各モータデータセットに同じ数のドライブデータセットが作成されます(「1MDS 毎の DDS」の数)。「最後の」モータデータセットのみ、これとは異なり、任意の数のドライブデータセットを設定できます。

1 ≤ a ≤ (「1MDS 毎の DDS」の数)

例

モータデータセット毎に 4 つのモータデータセット(MDS)と 8 つのドライブデータセット(DDS) (1MDS 毎の DDS)が設定されています。これは図 2-13 モータ/ドライブデータ

セットの切り替えの原理 (ページ 89)に示す可能なデータセットの組合わせの「No.」:22 に対応します。

- モータデータセット:MDS[0]、MDS[1]、... MDS[3] (「最後の」モータデータセット)
- モータデータセット毎のドライブデータセット:DDS[0] ... DDS[7]

つまり、個々のモータデータセットに対するドライブデータセットの数は、次のようになります。

モータデータセット (MDS)	モータデータセット(MDS)毎のドライブデータセット(DDS)の数
MDS[0] ... MDS[2]	8
MDS[3]	1 - 8

切り替えのタイミング:ドライブパラメータセット

原則として、ドライブパラメータセットは、どの時点でも切り替えることができます。軸の移動中は、特に速度コントローラパラメータとモータ速度合わせを切り替えるときに、トルクの不連続変化/ステップが発生する場合があります。このため、ドライブパラメータセットは、特に軸が停止状態のときは、停止状態のためにのみ切り替えることを推奨します。

下記も参照

インターフェースの概要 (ページ 88)

2.4 例

2.4.1 パラメータセットの切り替え

パラメータセットの切り替え

パラメータセットの切り替えは、機械軸 X1 の位置制御ゲイン(K_V 係数)を K_V = 4.0 から K_V = 0.5 に変更するためにおこないます。

必要条件

次のマシンデータによってパラメータセットの切り替えを有効にしてください。

MD35590 $MA_PARAMSET_CHANGE_ENABLE [AX1] = 1 または 2 (パラメータセットの変更可能)

機械軸 X1 の第 1 パラメータセットは、以下の NC/PLC インタフェースにより、インデックス「0」のマシンデータに従って設定されます。

DB31、… DBX9.0 - DBX9.2 = 0 (制御パラメータセット)

パラメータセット関連のマシンデータ

パラメータセット関連のマシンデータの設定は、次のとおりです。

マシンデータ	コメント
MD32200 $MA_POSCTRL_GAIN [0, AX1] = 4.0	パラメータセット 1 の K_V 設定
MD32200 $MA_POSCTRL_GAIN [1, AX1] = 2.0	パラメータセット 2 の K_V 設定
MD32200 $MA_POSCTRL_GAIN [2, AX1] = 1.0	パラメータセット 3 の K_V 設定
MD32200 $MA_POSCTRL_GAIN [3, AX1] = 0.5	パラメータセット 4 の K_V 設定
MD32200 $MA_POSCTRL_GAIN [4, AX1] = 0.25	パラメータセット 5 の K_V 設定
MD32200 $MA_POSCTRL_GAIN [5, AX1] = 0.125	パラメータセット 6 の K_V 設定
MD31050 $MA_DRIVE_AX_RATIO_DENOM [0, AX1] = 3	パラメータセット 1 の負荷側ギヤボックスの分母
MD31050 $MA_DRIVE_AX_RATIO_DENOM [1, AX1] = 3	パラメータセット 2 の負荷側ギヤボックスの分母
MD31050 $MA_DRIVE_AX_RATIO_DENOM [2, AX1] = 3	パラメータセット 3 の負荷側ギヤボックスの分母
MD31050 $MA_DRIVE_AX_RATIO_DENOM [3, AX1] = 3	パラメータセット 4 の負荷側ギヤボックスの分母
MD31050 $MA_DRIVE_AX_RATIO_DENOM [4, AX1] = 3	パラメータセット 5 の負荷側ギヤボックスの分母
MD31050 $MA_DRIVE_AX_RATIO_DENOM [5, AX1] = 3	パラメータセット 6 の負荷側ギヤボックスの分母
MD31060 $MA_DRIVE_AX_RATIO_NUMERA [0, AX1] = 5	パラメータセット 1 の負荷側ギヤボックスの分子
MD31060 $MA_DRIVE_AX_RATIO_NUMERA [1, AX1] = 5	パラメータセット 2 の負荷側ギヤボックスの分子

マシンデータ	コメント
MD31060 $MA_DRIVE_AX_RATIO_NUMERA [2, AX1] = 5	パラメータセット 3 の負荷側ギヤボックスの分子
MD31060 $MA_DRIVE_AX_RATIO_NUMERA [3, AX1] = 5	パラメータセット 4 の負荷側ギヤボックスの分子
MD31060 $MA_DRIVE_AX_RATIO_NUMERA [4, AX1] = 5	パラメータセット 5 の負荷側ギヤボックスの分子
MD31060 $MA_DRIVE_AX_RATIO_NUMERA [5, AX1] = 5	パラメータセット 6 の負荷側ギヤボックスの分子
MD35130 $MA_AX_VELO_LIMIT [0...5, AX1]	各パラメータセットの設定*)
MD32800 $MA_EQUIV_CURRCTRL_TIME [0..5, AX1]	各パラメータセットの設定*)
MD32810 $MA_EQUIV_SPEEDCTRL_TIME [0..5, AX1]	各パラメータセットの設定*)
MD32910 $MA_DYN_MATCH_TIME [0...5, AX1]	各パラメータセットの設定*)
*) 当該の行は、適用される構成規則に従って、各パラメータセットについて、それぞれ指定する必要があります。	

切り替え

位置制御ゲインを切り替えるために、機械軸 X1 に関して第 4 パラメータセットが PLC ユーザープログラムによって選択されます。

- PLC ユーザープログラムによる要求:
  DB31、… DBX9.0 – DBX9.2 = 3 (パラメータセットサーボ)
  - 機械軸 AX1 に関して、第 4 パラメータセットへの切り替え要求が送信されます。
  - 遅延時間が経過すると、パラメータセットが切り替わります。
  - これで、インデックス「3」のマシンデータに従って、パラメータセット 4 が有効になります。
- NC によるフィードバック:
  DB31、… DBX69.0 – DBX69.2 = 3 (パラメータセットサーボ)
  - NC がパラメータセットの切り替えを確認/通知します。

2.5 データリスト

2.5.1 マシンデータ

2.5.1.1 表示マシンデータ

番号 SINUMERIK Operate	識別子: $MM_	説明
9000	LCD_CONTRAST	コントラスト
9001	DISPLAY_TYPE	モニタタイプ
9004	DISPLAY_RESOLUTION	最小表示単位
9006	DISPLAY_SWITCH_OFF_INTERVAL	画面消去の時間

2.5.1.2 NC 別マシンデータ

番号	識別子: $MN_	説明
10350	FASTIO_DIG_NUM_INPUTS	有効なディジタル NC の入力バイト数
10360	FASTIO_DIG_NUM_OUTPUTS	有効なディジタル NC の出力バイト数
10361	FASTIO_DIG_SHORT_CIRCUIT	ディジタル入/出力部の短絡
11120	LUD_EXTENDED_SCOPE	グローバルプログラム変数(PUD)の有効化
11270	DEFAULT_VALUES_MEM_MSK	有効。 機能:GUD 初期値の保存
18150	MM_GUD_VALUES_MEM	GUD 用メモリ空間の確保

2.5.1.3 チャネルマシンデータ

番号	識別子: $MC_	説明
21015	INVOLUTE_RADIUS_DELTA	インボリュートの終点監視
21016	INVOLUTE_AUTO_ANGLE_LIMIT	インボリュート補間用自動角度制限
27800	TECHNOLOGY_MODE	チャネルの用途

番号	識別子: $MC_	説明
28150	MM_NUM_VDIVAR_ELEMENTS	PLC 変数の書き込み要素の数
28530	MM_PATH_VELO_SEGMENTS	ブロックでの軌跡速度を制限するための記憶要素の数

2.5.1.4 軸/主軸マシンデータ

番号	識別子: $MA_	説明
30350	SIMU_AX_VDI_OUTPUT	シミュレーション軸の軸信号の出力
33050	LUBRICATION_DIST	潤滑パルス距離
35590	PARAMSET_CHANGE_ENABLE	PLC からのパラメータセット定義可能
36060	STANDSTILL_VELO_TOL	軸/主軸が停止状態のときの最高速度/回転数
36610	AX_EMERGENCY_STOP_TIME	異常時の減速カーブの長さ
36620	SERVO_DISABLE_DELAY_TIME	サーボイネーブル遮断遅れ

2.5.2 システム変数

名前	説明
$P_FUMB	未使用のパートプログラムメモリ(Free User Memory Buffer)
$A_DBB[n]	PLC 上のデータ(データタイプ BYTE)
$A_DBW[n]	PLC 上のデータ(データタイプ WORD)
$A_DBD[n]	PLC 上のデータ(データタイプ DWORD)
$A_DBR[n]	PLC 上のデータ(データタイプ REAL)

2.5.3 信号

2.5.3.1 NC への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
キースイッチ設定 0 ～ 3	DB10.DBX56.4 - 7	DB2600.DBX0.4 - 7

2.5.3.2 NC からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
リモート診断有効(HMI アラームが未処置)	DB10.DBX103.0	-
AT ボックス準備完了	DB10.DBX103.5	-
HMI 温度制限	DB10.DBX103.6	-
HMI バッテリアラーム	DB10.DBX103.7	-
NC 準備完了	DB10.DBX104.7	-
操作パネル 2:「準備完了」	DB10.DBX108.1	-
MPI の操作パネル:「準備完了」	DB10.DBX108.2	-
OPI の操作パネル:「準備完了」	DB10.DBX108.3	DB2700.DBX2.3
ドライブがサイクリック運転	DB10.DBX108.5	DB2700.DBX2.5
ドライブ準備完了	DB10.DBX108.6	DB2700.DBX2.6
NC 準備完了	DB10.DBX108.7	DB2700.DBX2.7
NC アラームが有効です	DB10.DBX109.0	DB2700.DBX3.0

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
NCU ヒートシンク温度アラーム	DB10.DBX109.5	-
周囲温度アラーム	DB10.DBX109.6	DB2700.DBX3.6
NC バッテリアラーム	DB10.DBX109.7	-

2.5.3.3 操作パネルへの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
画面表示	DB19.DBX0.0	-
画面消去	DB19.DBX0.1	DB1900.DBX5000.1 [1)]
キーロック	DB19.DBX0.2	DB1900.DBX5000.2
キャンセルアラームの解除	DB19.DBX0.3	-
リコールアラームの解除	DB19.DBX0.4	-
WCS の現在値、(1) / MCS (0)	DB19.DBX0.7	DB1900.DBX5000.7
パートプログラム:アンロード	DB19.DBX13.5	-
パートプログラム:ロード	DB19.DBX13.6	-
パートプログラム:選択	DB19.DBX13.7	DB1700.DBX1000.7
ファイルシステムが有効(0) /無効(1)	DB19.DBX14.7	-
PLC からのプログラムの選択:プログラムリストのインデックス	DB19.DBB16 [2)]	DB1700.DBB1001 [2)]
PLC からのプログラムの選択:プログラムリストのプログラムインデックス	DB19.DBB17	DB1700.DBB1002
モード変更無効	DB19.DBX44.0	-

[1)] SINUMERIK 828D の場合は、DB1900.DBX5000.1 を使用して、明暗の制御がおこなわれます。

DB1900.DBX5000.1=**0**:画面表示

DB1900.DBX5000.1=**1**:画面消去

DB1900.DBX5000.0 は、スクリーンセーバー機能にとっては無意味です。

[2)] ビット 7:常時 1

2.5.3.4 操作パネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
画面消灯中	DB19.DBX20.1	-
操作画面:シミュレーション有効	DB19.DBX20.6	DB1900.DBX0.6
MCS/WCS の切り替え	DB19.DBX20.7	DB1900.DBX0.7
PLC 状態信号のプログラム選択:ジョブの完了	DB19.DBX26.1	DB1700.DBX2000.1
PLC 状態信号のプログラム選択:異常	DB19.DBX26.2	DB1700.DBX2000.2
PLC 状態信号のプログラム選択:有効	DB19.DBX26.3	DB1700.DBX2000.3
PLC 状態信号のプログラム選択:アンロード	DB19.DBX26.5	-
PLC 状態信号のプログラム選択:ロード	DB19.DBX26.6	-
PLC 状態信号のプログラム選択:選択	DB19.DBX26.7	DB1700.DBX2000.7
FC9:「Jog での計測」の起動	DB19.DBX42.0	-
FC9 出力:Active	DB19.DBX45.0	-
FC9 出力:Done	DB19.DBX45.1	-
FC9 出力:Error	DB19.DBX45.2	-
FC9 出力:StartErr	DB19.DBX45.3	-

2.5.3.5 チャネルへの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
残移動距離削除(チャネル別)	DB21、... .DBX6.2	DB320x.DBX6.2

2.5.3.6 チャネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
チャネル別 NC アラームが未処置	DB21、... .DBX36.6	DB330x.DBX4.6
加工停止を伴う NC アラームが未処置	DB21、… .DBX36.7	DB330x.DBX4.7
プログラムストップを伴う NC アラーム	DB21、… .DBX39.1	DB330x.DBX7.1
オーバストア有効	DB21、... .DBX318.7	-

2.5.3.7 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
軸/主軸無効	DB31、... .DBX1.3	DB380x.DBX1.3
フォローアップモード	DB31、... .DBX1.4	DB380x.DBX1.4
位置検出器 1	DB31、... .DBX1.5	DB380x.DBX1.5
位置検出器 2	DB31、... .DBX1.6	DB380x.DBX1.6
コントローライネーブル	DB31、... .DBX2.1	DB380x.DBX2.1
残移動距離削除(軸別)/主軸リセット	DB31、... .DBX2.2	DB380x.DBX2.2
モータ/ドライブデータセット:選択	DB31、... DBX21.0 - 4	DB380x.DBX4001.0 - 4
選択されているモータ	DB31、... .DBX21.5	DB380x.DBX4001.5
積分停止要求、速度コントローラ	DB31、... .DBX21.6	DB380x.DBX4001.6
パルスイネーブル要求	DB31、... .DBX21.7	DB380x.DBX4001.7

2.5.3.8 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
原点確立済み/原点同期済み、エンコーダ 1/2	DB31、... .DBX60.4/5	DB390x.DBX0.4/5
移動指令マイナス/プラス	DB31、... .DBX64.6/7	DB390x.DBX4.6/7
フォローアップ有効	DB31、... .DBX61.3	DB390x.DBX1.3
軸/主軸停止($n < n_{min}$)	DB31、... .DBX61.4	DB390x.DBX1.4
位置コントローラ有効	DB31、... .DBX61.5	DB390x.DBX1.5
速度コントローラ有効	DB31、... .DBX61.6	DB390x.DBX1.6
電流コントローラ有効	DB31、... .DBX61.7	DB390x.DBX1.7
潤滑パルス	DB31、... .DBX76.0	DB390x.DBX1002.0
ランプ関数発生器の無効化が有効	DB31、... .DBX92.1	DB390x.DBX4000.1
モータ/ドライブデータセット:表示	DB31、... DBX93.0 - 4	DB390x.DBX4001.0 - 4
ドライブ準備完了	DB31、... .DBX93.5	DB390x.DBX4001.5
積分停止、速度コントローラ	DB31、... .DBX93.6	DB390x.DBX4001.6
パルス許可中	DB31、 DBX93.7	DB390x.DBX4001.7
モータ温度警告	DB31、 DBX94.0	DB390x.DBX4002.0
ヒートシンク温度警告	DB31、 DBX94.1	DB390x.DBX4002.1

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
起動完了	DB31、 DBX94.2	DB390x.DBX4002.2
$\|M_d\| < M_{dx}$	DB31、 DBX94.3	DB390x.DBX4002.3
$\|n_{act}\| < n_{min}$	DB31、 DBX94.4	DB390x.DBX4002.4
$\|n_{act}\| < n_x$	DB31、 DBX94.5	DB390x.DBX4002.5
$n_{act} = n_{set}$	DB31、 DBX94.6	DB390x.DBX4002.6
アラームクラス C のアラームが有効	DB31、... .DBX95.7	-
モータ/ドライブデータセット:書式	DB31、... DBX130.0 - 4	DB390x.DBX4008.0 - 4
モータ/ドライブデータセット:フォーマットが有効	DB31、...DBX130.7	DB390x.DBX4008.7

A3:軸監視機能

3

3.1 輪郭監視

3.1.1 輪郭誤差

輪郭誤差は、位置制御ループの信号ひずみによって発生します。

信号ひずみは、線形または非線形のいずれかです。

線形信号ひずみ

線形信号ひずみは、次のような原因で発生します。

- 適切に設定されていない速度と位置コントローラ
- 軌跡の生成に関連する送り軸の異なるサーボゲイン係数(K_V)
  2 つの直線補間軸のサーボゲイン係数(K_V)が等しい場合、実位置は、時間的な遅れがあるものの、指定位置と同じ軌跡を通って移動します。サーボゲイン係数(K_V)が等しくない場合は、指定された軌跡と実際の軌跡の間に平行なオフセットが発生します。
- 送り用ドライブのダイナミック応答の不一致
  ドライブのダイナミック応答が等しくない場合、特に輪郭の変化時に軌跡誤差が発生します。2 つの送り用ドライブのダイナミック応答が等しくないと、真円が楕円に変形します。

非線形信号ひずみ

非線形信号ひずみは、次のような原因で発生します。

- 加工領域内での電流制限値の起動
- 指令速度制限の起動

- 位置制御ループ範囲内/外のバックラッシ
  円弧軌跡の移動中には、輪郭誤差は主に反転誤差と摩擦によって発生します。
  直線に沿った移動中には、輪郭誤差は位置制御ループ範囲外の反転誤差、たとえば傾斜フライス主軸によって発生します。これにより、実際の輪郭と指令された輪郭の間に平行なオフセットが発生します。直線の傾きが小さいほど、オフセットが大きくなります。
- スライドガイドの非線形の摩擦特性

3.1.2 追従誤差の監視

機能

制御工学の観点からすれば、機械軸の移動によって、ある程度の追従誤差(つまり、指令位置と実位置の間のずれ)が必ず発生します。

発生する追従誤差の程度は、次の要因によって決まります。

- 位置制御ループゲイン
  MD32200 $MA_POSCTRL_GAIN (サーボゲイン係数)
- 最高加速度
  MD32300 $MA_MAX_AX_ACCEL (最高軸加速度)
- 最高速度
  MD32000 $MA_MAX_AX_VELO (最高軸速度)
- フィードフォワード制御が有効な場合:
  軌跡モデルの精度とパラメータ:
  MD32610 $MA_VELO_FFW_WEIGHT (速度フィードフォワード制御用の係数)
  MD32800 $MA_EQUIV_CURRCTRL_TIME (フィードフォワード制御の電流制御ループ等価時定数)
  MD32810 $MA_EQUIV_SPEEDCTRL_TIME (フィードフォワード制御の速度制御ループ等価時定数)

加速段階では、まず、機械軸の移動による追従誤差が増えます。次に、位置制御ループのパラメータ設定に対応する時間が経過すると、追従誤差が一定になります(理想的な場合)。外乱の影響で、程度の差はあるものの、加工中に追従誤差の変動が必ず発生します。追従誤差監視では、このような追従誤差の変動によってアラームが発生するのを回避するために、追従誤差が変化してもよい許容範囲を定義する必要があります。

MD36400 $MA_CONTOUR_TOL (軌跡誤差監視許容範囲)

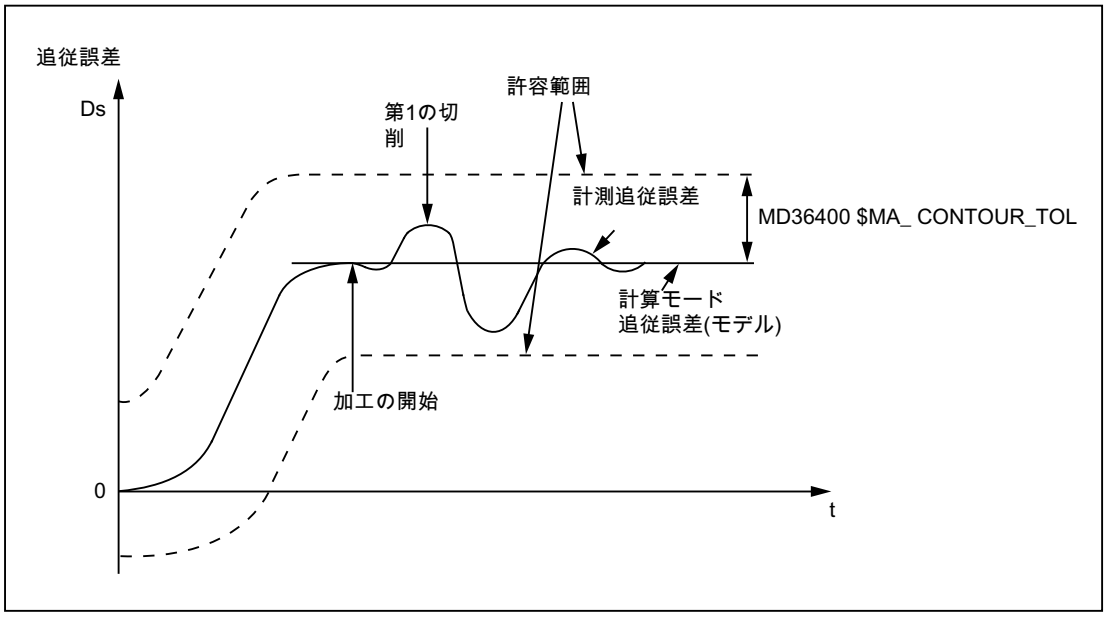


図 3-1　追従誤差の監視

有効性

追従誤差監視は、次の軸タイプで位置制御が有効である場合についてのみ機能します。

- 直線軸(フィードフォワード制御の有無を問わない)
- 回転軸(フィードフォワード制御の有無を問わない)
- 位置制御主軸

異常検出

設定された許容範囲を超えると、次のアラームが出力されます。

25050「軸<軸名称>軌跡誤差監視」

対応する軸/主軸が、下記の MD に設定された減速カーブによってフォローアップモードで停止します。

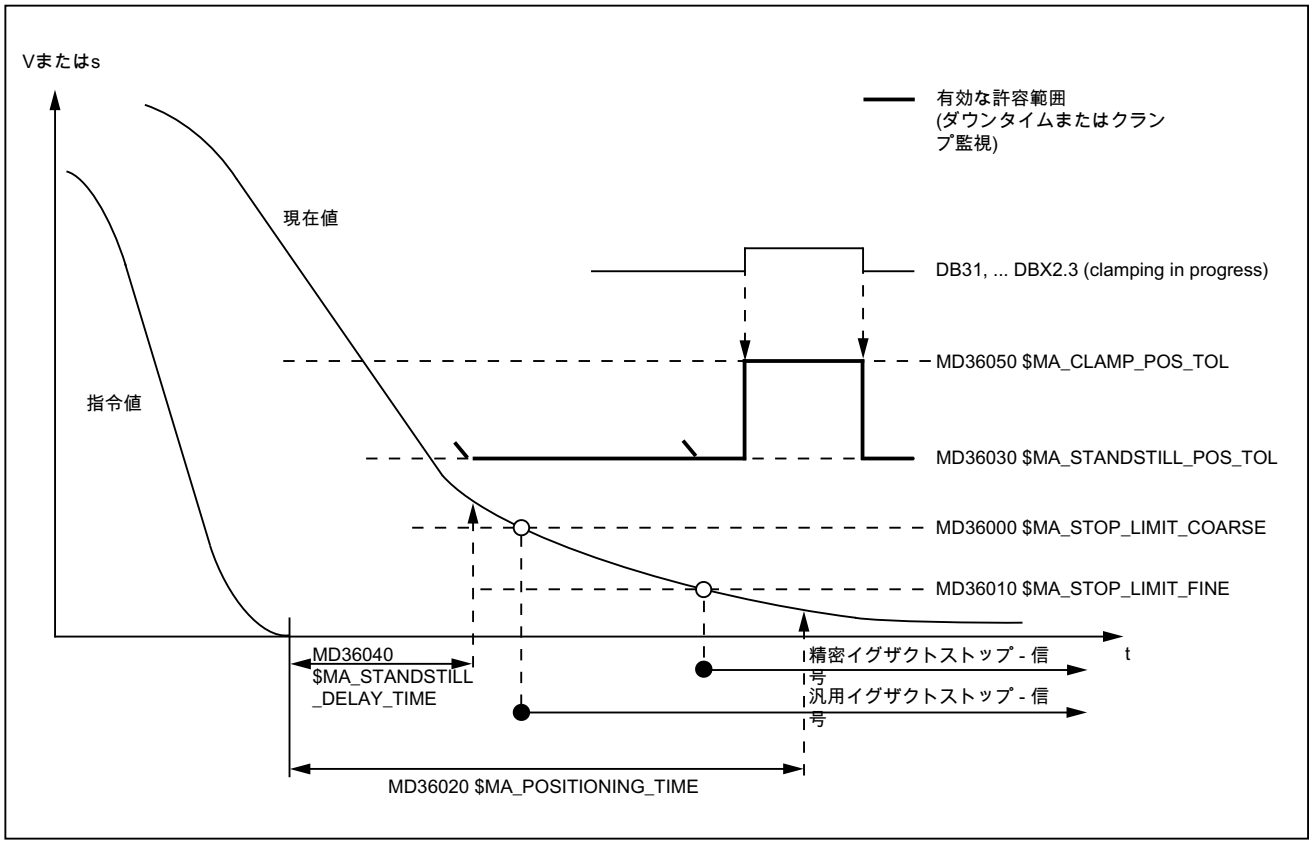
MD36610 $MA_AX_EMERGENCY_STOP_TIME
(異常発生時の減速カーブの最大時間)

3.2 位置決め監視、停止状態監視およびクランプ監視

3.2.1 位置決め監視、停止状態監視およびクランプ監視の関係

概要

次の図は、位置決め、停止状態およびクランプの各監視機能の関係を示しています。



3.2.2 位置決め監視

機能

位置決め動作終了時、すなわち:

- 指令速度 = 0 かつ
- DB31、... DBX64.6/7 (マイナス/プラスの移動命令) = 0

位置監視が有効になり、遅延時間内に関連する各機械軸の追従誤差が精密イグザクトストップの許容範囲よりも小さくなったかどうかを確認します。

MD36010 $MA_STOP_LIMIT_FINE (精密イグザクトストップ)

MD36020 $MA_POSITIONING_TIME (精密イグザクトストップの遅延時間)

「精密イグザクトストップ」に達すると、位置決め監視は無効になります。

注記

精密イグザクトストップの許容範囲が小さいほど、位置決め動作の時間が長くなり、ブロック切り替えまでの時間が長くなります。

MD 設定の規則

MD36010 $MA_STOP_LIMIT_FINE	MD36020 $MA_POSITIONING_TIME
大	比較的短い時間を選択できます。
小	比較的長い時間を選択してください。

MD32200 $MA_POSCTRL_GAIN (サーボゲイン係数)	MD36020 $MA_POSITIONING_TIME
小	比較的長い時間を選択してください。
大	比較的短い時間を選択できます。

有効性

位置決め監視は、次の軸タイプで位置制御が有効である場合についてのみ機能します。

- 直線軸
- 回転軸
- 位置制御主軸

異常検出

設定された位置決め監視時間を超えると、次のアラームが出力されます。

25080「軸<軸名称>位置決め監視」

対応する軸が、下記の MD に設定された減速カーブによってフォローアップモードで停止します。

MD36610 $MA_AX_EMERGENCY_STOP_TIME
(異常発生時の減速カーブの最大時間)

3.2.3 停止状態監視

機能

位置決め動作終了時、すなわち:

- 指令速度 = 0 かつ
- DB31、... DBX64.6/7 (マイナス/プラスの移動命令) = 0

停止状態監視が有効になり、遅延時間内に関連する各機械軸の追従誤差が停止時許容移動範囲よりも小さくなったかどうかを確認します。

MD36040 $MA_STANDSTILL_DELAY_TIME (停止状態監視遅延時間)

MD36030 $MA_STANDSTILL_POS_TOL (停止時許容移動範囲)

求められるイグザクトストップ状態に達すると、位置決め動作が完了します。

DB31、... DBX60.6/7 (汎用/精密イグザクトストップ位置に到達) = 1

位置決め監視機能が無効になり、停止状態監視に切り替わります。

停止状態監視では、停止時許容移動範囲が順守されているかどうか監視されます。 新しい移動要求が受信されない場合、機械軸は停止時許容移動範囲を逸脱してはなりません。

有効性

停止状態監視は、次の軸タイプで位置制御が有効である場合についてのみ機能します。

- 直線軸
- 回転軸
- 位置制御主軸

異常検出

遅延時間または停止時許容移動範囲、あるいはその両方を超えると、次のアラームが出力されます。

25040「軸<軸名称>停止状態監視」

対応する軸が、下記の MD に設定された減速カーブによってフォローアップモードで停止します。

MD36610 $MA_AX_EMERGENCY_STOP_TIME
(異常発生時の減速カーブの最大時間)

3.2.4 パラメータセットに依存するイグザクトストップと停止許容移動範囲

さまざまな加工状況や軸の応答性、たとえば

- 動作状態 A: 高精度、長時間の加工
- 動作状態 B: 低精度、短時間の加工
- ギヤ切替後の質量比率の変化

などに適応するために、位置決め許容範囲

- MD36000 $MA_STOP_LIMIT_COARSE (汎用イグザクトストップ)
- MD36010 $MA_STOP_LIMIT_FINE (精密イグザクトストップ)
- MD36030 $MA_STANDSTILL_POS_TOL (停止時許容移動範囲)

をパラメータセットに依存する共通係数で重み付けすることができます。

MD36012 $MA_STOP_LIMIT_FACTOR (汎用/精密イグザクトストップと停止許容値の係数)

この係数は 3 つの位置決め許容範囲すべてに共通して適用されるため、許容値間の関係は一定に保たれます。

3.2.5 クランプ監視

3.2.5.1 機能

位置決め動作完了と同時に機械的にクランプされた機械軸の場合は、クランプ処理の結果として、位置指令からの軸のずれが発生することがあります。NC/PLC インターフェース信号 DB31、... DBX2.3 (クランプ中)を設定すると、クランプ処理の間、停止許容移動範囲(MD36030 $MA_STANDSTILL_POS_TOL)ではなく、クランプ許容移動量 (MD36050 $MA_CLAMP_POS_TOL)が監視されます。クランプ許容移動量がオーバシュートされた場合は、アラーム 26000「クランプ監視」が発行されます。

アラーム遅延時間

クランプ許容移動量のオーバシュートが期限付きで許容される場合は、マシンデータ MD36051 $MA_CLAMP_POS_TOL_TIME を使用して、アラームの遅延時間を指定してください。これで、クランプ許容移動量がオーバシュートされたときに、設定した時間の経過後にのみアラームが出力されます。その時間の経過後にクランプ許容移動量が再度減少した場合は、アラームは出力されません。この時間は、クランプ許容移動量が再度オーバシュートされると、再開されます。

アラームの遅延時間の経過前に、クランプ許容移動量のオーバシュートに応答できるようにするには、軸別の NC/PLC インターフェース信号 DB31、... DBX102.3 (クランプ許容移動量のオーバシュート)を設定します。この信号は、クランプ許容移動量が減少すると、再度リセットされます。

3.2.5.2 マシンデータ

クランプ許容移動量

停止許容移動範囲より大きいクランプ許容移動量がマシンデータに入力されます。

MD36050 $MA_CLAMP_POS_TOL[<軸>] = <クランプ許容移動量>

アラーム遅延時間

クランプ許容移動量のオーバーシュートが期限付きで許容される場合、許容可能な最大アラーム遅延時間をマシンデータで指定する必要があります。

MD36051 $MA_CLAMP_POS_TOL_TIME[<軸>] = <アラーム遅延時間>

クランプ許容移動量を超過した場合、アラーム遅延時間の経過後にのみアラーム 26000「クランプ監視」が表示されます。

アラーム遅延時間の経過前にクランプ許容移動量を再度下回った場合、アラームは出力されません。

クランプ許容移動量を再度上回ると、アラーム遅延時間は再度開始されます。

特殊クランプ機能

パートプログラムの実行シーケンスの解除と設定を自動化する特殊クランプ機能は、マシンデータを使用してビット単位で有効になります。

MD36052 $MA_STOP_ON_CLAMPING[<Achse>], <ビット> = <値>

<ビット>	<値>	意味
0	クランプを解除するための自動停止	
	0	無効
	1	有効
1	最適化されたクランプリリース	
	0	無効
	1	有効、必要条件:ビット 0 == 1
2	クランプを設定するための自動停止	
	0	無効
	1	有効

3.2.5.3 NC/PLC インタフェース信号

クランプ監視の有効化

クランプ監視は、次の NC/PLC インターフェース信号を設定すると有効になります。

DB31、... DBX2.3 = 1 (クランプ中)

クランプ許容移動量のオーバシュート

クランプ許容移動量のオーバシュートは、次の NC/PLC インターフェース信号で示されます。

DB31、... DBX102.3 == 1 (クランプ許容移動量のオーバシュート)

信号は、クランプ許容移動量がアラームの遅延時間内にオーバシュートすると**設定されます**。

信号は、クランプ許容移動量がアラームの遅延時間内に減少するか、軸でフォローアップモードが有効になっていると**リセット**されます。

3.2.5.4 故障応答

クランプ許容移動量を上回った場合の故障応答:

- アラーム 26000「クランプ監視」が示されます。
- 設定された最大加減速で軸が停止状態になります。
  MD32300 $MA_MAX_AX_ACCEL
  これにより、異常状態の減速カーブの最大時間が監視されます。
  MD36610 $MA_AX_EMERGENCY_STOP_TIME
- 軸でフォローアップモードが有効になります。
  DB31、... DBX61.3 == 1
- 「クランプ許容移動量の超過」信号がリセットされます。
  DB31、... DBX102.3 == 0

3.2.5.5 「クランプを解除するための自動停止」クランプ機能

「クランプを解除するための自動停止」クランプ機能により、連続軌跡モードのクランプ軸の移動ブロックの前に NC 内部停止が追加されます。

ブロック変更の前にクランプ軸のコントローライネーブル信号(DB31、... DBX2.1)が設定されている場合は、停止することも、連続軌跡モードの中断が発生することもありません。

ブロック変更の前にクランプ軸のコントローライネーブル信号が**設定されていない**場合は、停止が機能します。

パラメータ設定

MD36052 $MA_STOP_ON_CLAMPING[<クランプ軸>] = 'H01'

必要条件/前提条件

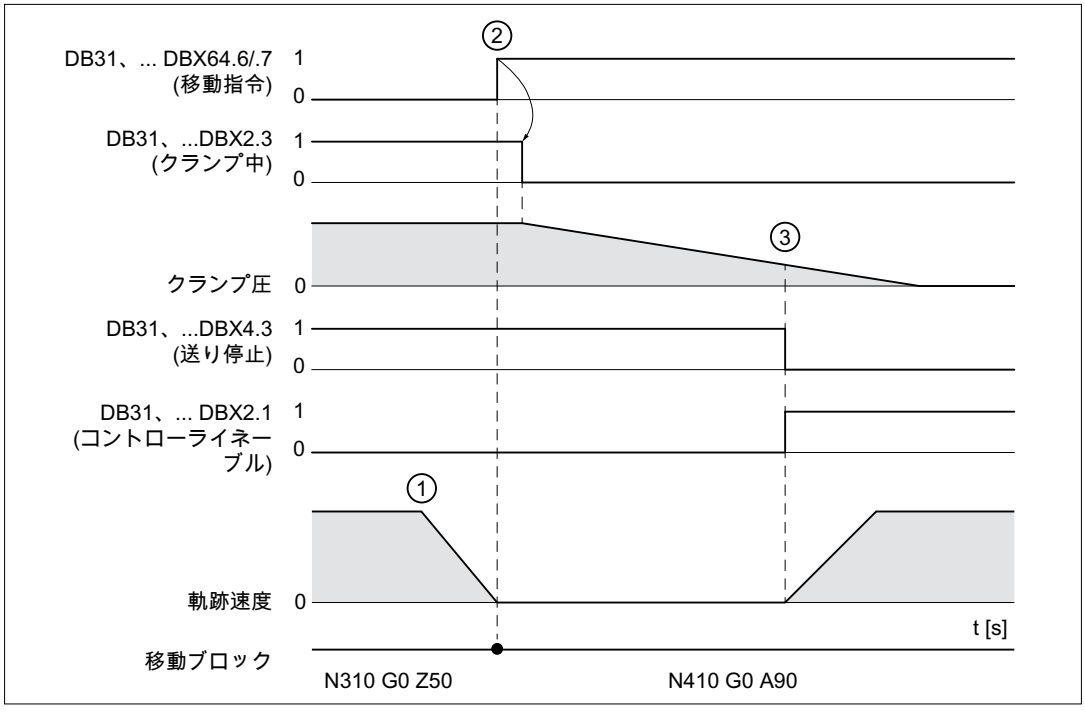
- クランプ軸に対して**移動指令**(DB31, ... DBX64.6 / .7)がある場合、PLC ユーザープログラムによりクランプが**解放**されます。
- コントローライネーブル信号(DB31, ... DBX2.1)とクランプ軸のクランプとの間には、次の関係が存在する必要があります。

コントローライネーブル	⇒	クランプ軸
未設定	⇒	クランプ済み
セット	⇒	未クランプ

例:

プログラムコード	コメント
N100 G0 X0 Y0 Z0 A0 G90 G54 F500	; 初期状態にアプローチ
N101 G641 ADIS=.1 ADISPOS=5	; 連続軌跡モードを有効化
N210 G1 X10	; 移動ブロック
N220 G1 X5 Y20	; "
N310 G0 Z50	; 位置決めブロック
N410 G0 A90	; " (**クランプ軸**)
N510 G0 X100	; "
N520 G0 Z2	; "
N610 G1 Z-4	; 移動ブロック
N620 G1 X0 Y-20	; "

NC/PLC インターフェース信号と、N310 および N410 ブロックの状態の変化の図:



① NC:自動挿入された停止により N310 ブロックエンドで停止します。

② NC → PLC:ブロック変更の後、クランプ軸の移動指令が設定されます。
PLC:移動指令に基づいてクランプが解除されます。

③ PLC → NC:クランプ圧が適切に取り除かれます。クランプ軸で移動が有効になります。

3.2.5.6 「時間を優先したクランプの解除」クランプ機能

連続軌跡モードで「時間を優先したクランプの解除」クランプ機能と「クランプを解除するための自動停止」クランプ機能と組み合わせると、クランプ軸の移動指令の先読み設定によって、NC 内部クランプの解除が要求されます。移動指令は、クランプ軸の移動までに位置決め(G0 ブロック)は実行されるが、処理(G1 ブロック)は実行されない場合にのみ設定されます。

クランプ軸の移動ブロックへの参照を取得するために、移動指令により、2 個までの高速処理ブロック (G0) (内部で生成された中間ブロックを含む) が移動ブロックの先頭に付加されます。

起動

MD36052 $MA_STOP_ON_CLAMPING[<クランプ軸>] = 'H03'

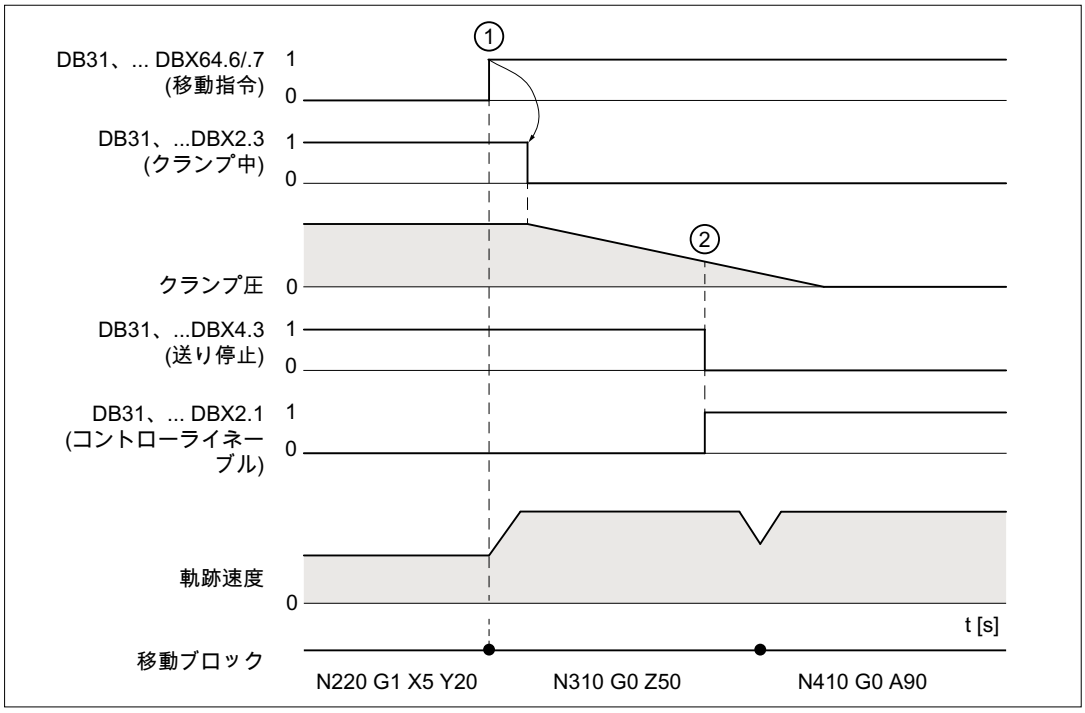
必要条件/前提条件

- クランプ軸に対して**移動指令**(DB31, ... DBX64.6 / .7)がある場合、PLC ユーザープログラムによりクランプが**解放**されます。
- 他の軸が高速処理(G0)で移動している間、クランプ軸をクランプすることは**できません**。

例:

プログラムコード	コメント
N100 G0 X0 Y0 Z0 A0 G90 G54 F500	; 初期状態にアプローチ
N101 G641 ADIS=.1 ADISPOS=5	; 連続軌跡モードを有効化
N210 G1 X10	; 加工ブロック
N220 G1 X5 Y20	; "
N310 G0 Z50	; 位置決めブロック
N410 G0 A90	; " (**クランプ軸**)
N510 G0 X100	; "
N520 G0 Z2	; "
N610 G1 Z-4	; 加工ブロック
N620 G1 X0 Y-20	; "

NC/PLC インターフェース信号と、N220 ブロックから N410 ブロックへの状態の変化の図:



① NC → PLC:ブロック変更により、クランプ軸の移動指令が設定されます。
PLC:移動指令に基づいてクランプが解除されます。

② PLC → NC:クランプ圧が適切に取り除かれます。クランプ軸で移動が有効になります。

3.2.5.7 「クランプを設定するための自動停止」クランプ機能

クランプ処理には時間がかかります。連続軌跡モードでは、加工開始前にクランプが確実に有効になるように、プログラミング(たとえば、G09、G60、補助機能出力など)によって移動を明確に停止する必要があります。

「クランプを設定するための自動停止」クランプ機能により、連続軌跡モードで移動が自動的に停止します。クランプ軸が依然としてクランプしていない場合、クランプ動作は次の加工ブロック(早送り G0 なしの移動ブロック)の**前**または**その途中**で停止します。クランプが発生し、追加移動動作が有効である条件は、0%に一致しない PLC ユーザープログラム(DB21, ... DBB4 ≠ 0%)によるチャネル別送り速度オーバライドの設定に対応します。

起動

MD36052 $MA_STOP_ON_CLAMPING[<クランプ軸>] = 'H04'

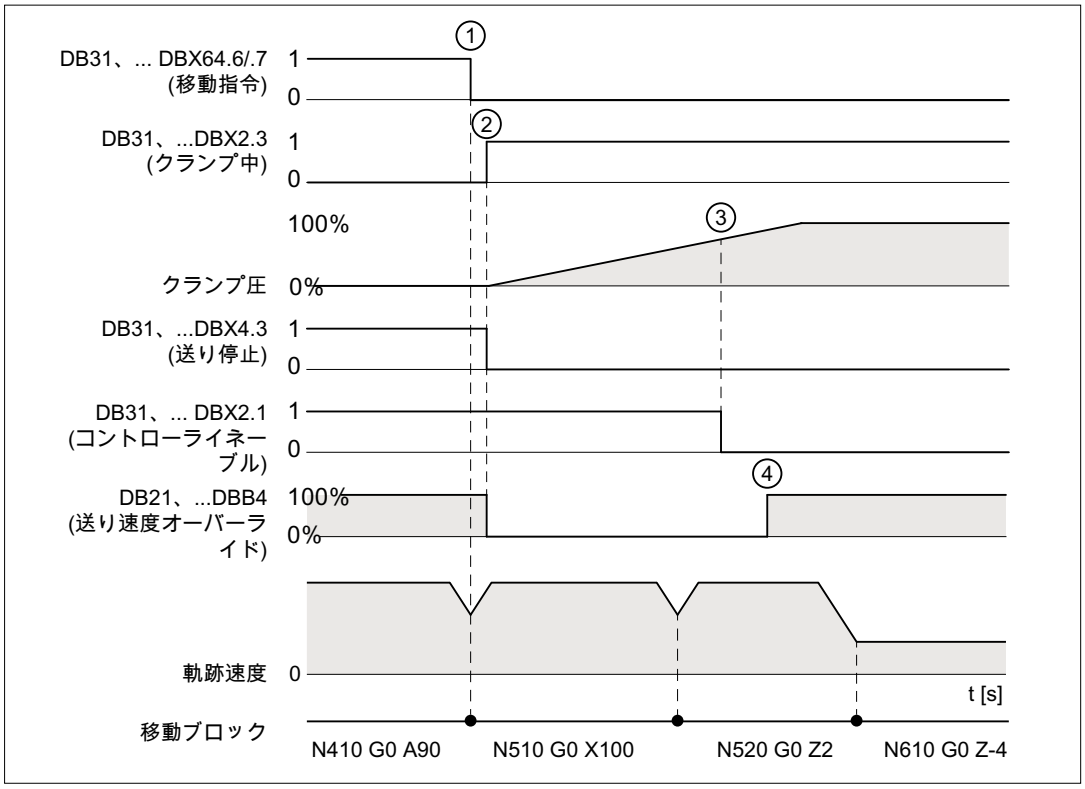
必要条件/前提条件

- クランプ軸に対して移動指令(DB31, ... DBX64.6 / .7)が**ない**場合、PLC ユーザープログラムによりクランプが**閉じ**ます。
- 他の軸が高速処理(G0)で移動している間、クランプ軸をクランプすることは**できません**。
- チャネル別送り速度オーバライドが **0%に一致しない場合** (DB21, ... DBB4 ≠ 0%)、クランプ軸は**クランプされます**。

例

プログラムコード	コメント
N100 G0 X0 Y0 Z0 A0 G90 G54 F500	; 初期状態にアプローチ
N101 G641 ADIS=.1 ADISPOS=5	; 連続軌跡モードを有効化
N210 G1 X10	; 加工ブロック
N220 G1 X5 Y20	; "
N310 G0 Z50	; 位置決めブロック
N410 G0 A90	; " (**クランプ軸**)
N510 G0 X100	; "
N520 G0 Z2	; "
N610 G1 Z-4	; 加工ブロック
N620 G1 X0 Y-20	; "

NC/PLC インターフェース信号と、N410 ブロックから N610 ブロックへの状態の変化の図:



① NC → PLC:ブロック変更により、クランプ軸の移動指令がリセットされます。

② PLC:クランプが開始されます

③ PLC → NC:コントローライネーブルをリセットするために十分なクランプ圧があります

④ PLC → NC:チャネル別送り速度オーバライドが 0%以外に設定され、N610 加工が有効になります。

3.2.5.8 必要条件

中断された連続軌跡モード

前述のクランプ機能の間に、移動しないブロック(M 機能 M82/M83 の出力)により、連続軌跡モードと「先読み」機能が中断した場合は、各機能が次のように動作します。

- **クランプ機能:「時間を優先した軸クランプの解除」**
  (MD36052 $MA_STOP_ON_CLAMPING[<軸>] = 'B011')
  移動指令は、連続軌跡モードが有効なブロックについてのみ先読みモードで設定されるため、機能が動作しなくなります。以下のサンプルプログラムのブロック N320 への M 機能 M82 の出力により、移動が停止し、そのために連続軌跡モードが中断します。
  機能による N410 での先読み停止は不要になります。いずれにしても停止が発生するからです。
- **クランプ機能:「クランプを設定するための自動停止」:**
  (MD36052 $MA_STOP_ON_CLAMPING[<axis] = 'B100')
  機能は、M83 に関係なく、停止を生成します。これは、「送り速度オーバライド 0%」に応じて実行されます。したがって、最初の加工ブロックの前で軸が停止されます。

注記

クランプなしのクランプ機能の使用

以下のクランプ機能は、軸のクランプとは無関係に使用することもできます。

- 「クランプを解除するための自動停止」:
  MD36052 $MA_STOP_ON_CLAMPING[<軸>] = 'B001'
  動作:設定された**<軸>**のコントローライネーブル(DB31、... DBX2.1)が **設定されていない**が、以下のブロックのいずれかで移動する場合に、軌道で実行中のブロックで停止が発生します。
- 「クランプを設定するための自動停止」:
  MD36052 $MA_STOP_ON_CLAMPING[<軸>] = 'B100'
  動作:高速処理ブロック(G0)から移動ブロック(G1)への移行でチャネル別の送り速度オーバライド(DB21、... DBB4)が 0%の場合に、軌道で実行中のブロックで停止が発生します。

いずれの場合も、当該ブロック内だけでなく、そのブロックの開始前にも、連続軌跡モードの軌跡移動が確実に停止されるようになっています。

表 3-1　サンプルプログラム:中断された連続軌跡モード

プログラムコード	コメント
N100 G0 X0 Y0 Z0 A0 G90 G54 F500	; 初期状態にアプローチ
N101 G641 ADIS=.1 ADISPOS=5	; 連続軌跡モードを有効化
N210 G1 X10	; 移動ブロック
N220 G1 X5 Y20	; "
N310 G0 Z50	; 早送りブロック
N320 M82	; 連続軌跡モードの**中断**
N410 G0 A90	; 早送りブロック

プログラムコード	コメント
N420 M83	; 連続軌跡モードの**中断**
N510 G0 X100	; 早送りブロック
N520 G0 Z2	; "
N610 G1 Z-4	; 移動ブロック
N620 G1 X0 Y-20	; "

ブロック切り替え条件:クランプ許容移動量

クランプ監視の起動(DB31、... DBX2.3 = 1)の後は、ブロックの最後で停止する移動ブロックのブロック変更条件では、イグザクトストップ条件ではなく、クランプ許容移動量が軸に対して機能します。

MD36050 $MA_CLAMP_POS_TOL (インタフェース信号「クランプ有効」のときのクランプ許容移動量)

クランプの解除の動作

クランプ軸は、クランプ処理によって位置指令から移動された場合は、クランプを解除し、(DB31, ... DBX2.1)コントローライネーブル信号を設定した後、NC によって位置指令に戻されます。再位置決めの方法は、以下のように、クランプ処理中に「フォローアップモード」が軸に対して有効になっているかどうかによって異なります。

- DB31、... DBX1.4 == 0 (フォローアップモードが有効でない) ⇒ 位置コントローラによる急変
- DB31、... DBX1.4 == 1 (フォローアップモードが有効) ⇒ 補間方式

注記

以下のデータは、フォローアップモード有効化(DB31、... DBX1.4)の条件として PLC ユーザープログラムで使用できます。

- DB31、... DBX60.6 = 7 (汎用/精密イグザクトストップ位置に到達)
- クランプ軸の現在位置

フォローアップモード

クランプ監視はフォローアップモードでは無効です。

DB31、... DBX1.4 == 1 (フォローアップモード)

3.3 速度指令監視

機能

速度指令値は、以下のものから構成されています。

- 位置コントローラの速度指令値
- フィードフォワード制御の速度指令値分配(フィードフォワード制御が有効である場合のみ)
- ドリフト補正(アナログ指令インタフェースを伴うドライブのみ)

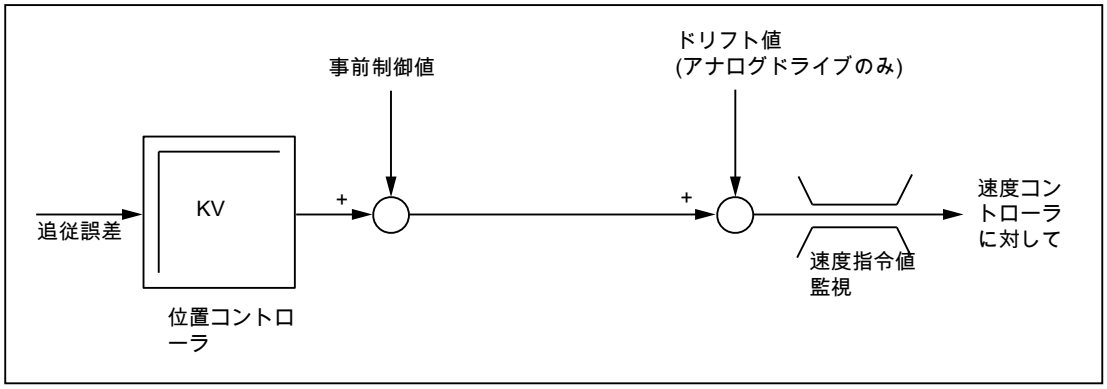


図 3-2　速度指令値の演算

速度指令監視では、ドライブの物理的限界値を超えないように制御信号または出力信号が下記の MD 設定値に制限されます(アナログ指令インタフェースでは 10 V、ディジタルドライブでは定格速度)。

MD36210 $MA_CTRLOUT_LIMIT (最大速度指令値)

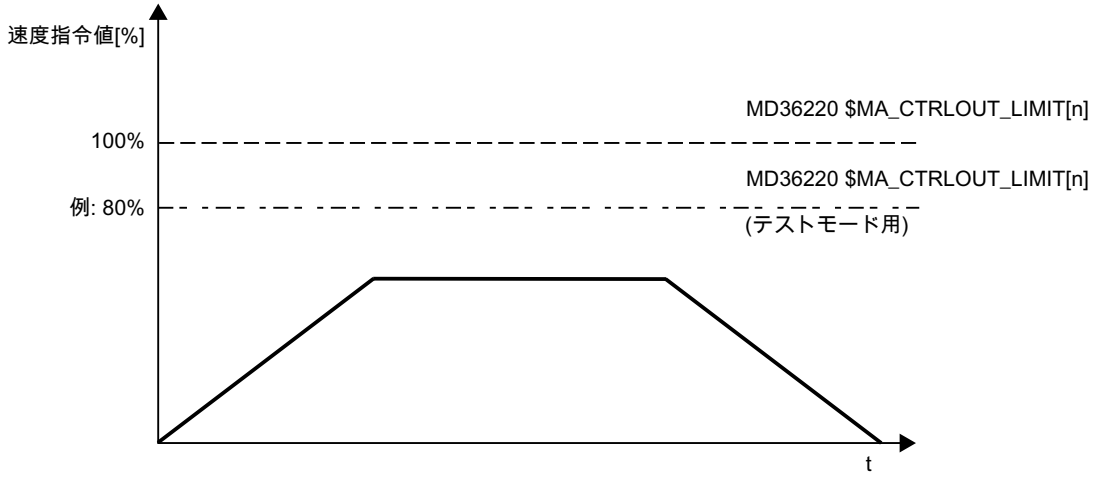


図 3-3　速度指令値制限

速度指令監視の遅延

一つ一つの速度制限超過すべてに対してアラーム応答が行なわれるのを回避するために、遅延時間を設定できます。

MD36220 $MA_CTRLOUT_LIMIT_TIME (速度指令監視の遅延)

設定された時間よりも長い期間、速度制限が必要な場合のみ、対応するアラーム応答が行なわれます。

有効性

速度指令監視は閉ループ位置制御軸についてのみ有効であり、無効にすることはできません。

異常検出

設定された遅延時間を超えると、次のアラームが出力されます。

25060「軸<軸名称>速度指令値制限」

対応する軸が、下記の MD に設定された減速カーブによってフォローアップモードで停止します。

MD36610 $MA_AX_EMERGENCY_STOP_TIME
(異常発生時の減速カーブの最大時間)

注記

速度指令監視の制限値に達すると、それによって、軸の位置フィードバックループが非線形になります。 軸が輪郭の生成にかかわる場合、輪郭誤差が発生します。

3.4 フィードバック速度監視

機能

フィードバック速度監視では、機械軸/主軸のフィードバック速度が、下記の MD に設定されているスレッシホールドを超えていないことが確認されます。

MD36200 $MA_AX_VELO_LIMIT(速度監視レベル)

この検出値は、設定されている最高速度より 10 ～ 15%大きくします。

- 軸の場合:
  MD32000 $MA_MAX_AX_VELO (最高軸速度)
- 主軸の場合:
  MD35110 $MA_GEAR_STEP_MAX_VELO_LIMIT[n](ギヤ選択の最高速度)

この設定を使用すると、通常は速度が速度監視レベルを超えません(例外: ドライブ異常)。

起動

フィードバック速度監視は、動作中の検出器によって有効なフィードバック値(エンコーダ限界周波数を超えていない)が返された直後に有効になります。

有効性

フィードバック速度監視は、次の軸タイプで位置制御が有効である場合についてのみ機能します。

- 直線軸
- 回転軸
- 開ループ制御と位置制御されている主軸

異常検出

検出値を超えると、次のアラームが表示されます。

25030「軸<軸名称>フィードバック速度アラームリミット」

対応する軸が、下記の MD に設定された減速カーブによってフォローアップモードで停止します。

MD36610 $MA_AX_EMERGENCY_STOP_TIME
(異常発生時の減速カーブの最大時間)

3.5 検出器監視

NC が検出器のハードウェアにダイレクトアクセスしないため、検出器監視は主としてドライブソフトウェアによっておこなわれます。

ドライブの監視機能

- ハードウェアの故障の監視(検出器の故障、ケーブル断線など)
- ゼロマーク監視

参照先
『ドライブの機能 SINAMICS S120』

ドライブで実行される検出器監視は、NC アラーム(アラーム 25000 およびそれ以降のアラーム)または NC 応答(原点確立の中止または高速計測)にマッピングされます。NC の正確な動作は、次のマシンデータの設定に依存します。

MD36310 $MA_ENC_ZERO_MONITORING

値	意味	
= 0	ハードウェア障害の監視:	ON 有効な検出器でハードウェア障害が検出されると、POWER ON アラーム 25000 が表示されます。 「軸<軸名称>有効なエンコーダのハードウェアの故障」 対応する軸が、下記の MD に設定された減速カーブによってフォローアップモードで停止します。 MD36610 $MA_AX_EMERGENCY_STOP_TIME (障害発生時の減速カーブの最大時間) 不使用検出器でハードウェア障害が検出されると、アラーム 25001 が表示されます。 「軸<軸名称>不使用エンコーダのハードウェアの故障」 これ以外のアラーム応答はありません。
	原点マーク監視:	OFF アラーム 25020 と 25021 (下記を参照してください)がマスクされます。
= 100	原点マーク監視がおこなわないばかりでなく、すべてのエンコーダ監視機能が隠されます(つまり、アラーム 25020 (25021)に加えて、アラーム 25000 (25001)と 25010 (25011)がマスクされます)。	

値	意味	
> 0、ただし< 100	ハードウェア障害の監視:	ON (上記を参照してください)
	原点マーク監視:	ON 有効な検出器で原点マーク監視が異常を検出すると、アラーム 25020 が表示されます。 「軸<軸名称>有効なエンコーダの原点マーク監視」 対応する軸が、下記の MD に設定された減速カーブによってフォローアップモードで停止します。 MD36610 $MA_AX_EMERGENCY_STOP_TIME (障害発生時の減速カーブの最大時間) 不使用検出器で原点マーク監視が異常を検出すると、アラーム 25021 が表示されます。 「軸<軸名称>不使用エンコーダの原点マーク監視」 これ以外のアラーム応答はありません。
> 100	ハードウェア障害の監視:	弱められたエラーメッセージで ON POWER ON アラーム 25000 がリセットアラーム 25010 に置き換えられ、リセットアラーム 25001 がキャンセルアラーム 25011 に置き換えられます。
	原点マーク監視:	ON (上記を参照してください)

アラームの詳細については、以下を参照してください:

参照先:
『診断マニュアル』

注記
ハードウェア障害の場合は、機械軸の原点確立状態がリセットされます。
DB31、... DBX60.4/5(原点確立済み/原点同期済み 1/2) = 0

NC の監視機能

- エンコーダ限界周波数監視
- 絶対値エンコーダの妥当性チェック

3.5.1 エンコーダ限界周波数監視

機能

NC エンコーダ限界周波数監視は、ドライブの設定とメッセージ情報に基づいています。 エンコーダ周波数が設定されたエンコーダ限界周波数を超えていないことを監視します。

MD36300 $MA_ENC_FREQ_LIMIT(エンコーダ限界周波数)

エンコーダ限界周波数監視では、NC/PLC インタフェースで選択された有効な検出器が常に参照されます。

DB31、... DBX1.5/1.6 (位置検出器 1/2)

有効性

エンコーダ限界周波数は次の軸に対して有効です。

- 直線軸
- 回転軸
- 開ループ制御と位置制御されている主軸

異常検出

エンコーダ限界周波数を超えると、次の処理がおこなわれます。

- PLC への通知:
  DB31、... DBX60.2 または 60.3 = 1(エンコーダ限界周波数超過 1 または 2)
- 主軸
  主軸は停止せず、速度制御で回転を継続します。
  エンコーダ周波数がエンコーダ限界周波数を下回る程度に主軸速度が低下すると、主軸のフィードバック回路が自動的に再原点同期されます。
- 軸
  次のアラームが表示されます。
  21610「チャネル<チャネル番号>軸<軸名称>エンコーダ<エンコーダ番号>周波数超過」
  対応する軸が、下記の MD に設定された減速カーブによってフォローアップモードで停止します。
  MD36610 $MA_AX_EMERGENCY_STOP_TIME
  (異常発生時の減速カーブの最大時間)

注記

エンコーダ限界周波数を超えた場合は、位置制御されている機械軸を再原点確立してください(「R1:レファレンス点復帰 (ページ 1497)」の章を参照してください)。

3.5.2 絶対値エンコーダの妥当性チェック

機能

絶対値エンコーダ(MD30240 $MA_ENC_TYPE = 4)では、検出器により提供されたアブソリュート値がフィードバック値の妥当性のチェックに使用されます。

確認時には、NC は、エンコーダからのインクレメンタル情報に基づき位置制御周期で入手している位置データと、アブソリュート情報およびインクレメンタル情報から直接生成される新しい位置データを比較し、算出された位置の違いが、許容偏差を超えていないことを確認します。

MD36310 $MA_ENC_ZERO_MONITORING (絶対値エンコーダとインクリメンタルエンコーダのトラック間の 1/2 汎用インクリメントの許容偏差)

注記

絶対値エンコーダの妥当性チェックでは特に、アブソリュート値伝送時の異常またはアブソリュート値トラックの汚れを原因とするすべての偏差が検出されます。しかし、インクレメンタルトラックでの小さなエラー(バースト干渉、インパルスエラー)は検出されません。このような事例では、妥当性チェックはミリメータ領域の偏差だけを検出します。したがってこの監視形態は、アブソリュート位置異常の診断を支援する追加監視の役割を果たします。

注記

ロータリー絶対値エンコーダ

妥当性チェックをロータリー絶対値エンコーダに使用する場合、モジュロ範囲(MD34220 $MA_ENC_ABS_TURNS_MODULO)を設定するときに SINAMICS パラメータ p0979 を考慮してください。

注記

NC はソフトウェアのアップグレード

絶対値エンコーダで妥当性チェックが起動中(MD36310 > 0)の場合は、既存の MD36310 の設定値をチェックしてください。また、必要に応じて、NC ソフトウェアのアップグレード中に設定値を増やしてください。

原点マーク診断

絶対値エンコーダでは、セットアップ中の妥当性チェックのための許容偏差を決定してください。これは、次のマシンデータを介しておこなうことができます。

MD36312 $MA_ENC_ABS_ZEROMON_WARNING (原点監視警告レベル)

値	意味
0	原点マーク診断なし
> 0	絶対値エンコーダとインクリメンタルエンコーダのトラック間の 1/2 汎用インクリメントの許容偏差

システムのセットアップ時の手順:

1. 原点マーク監視を解除:
   MD36310 $MA_ENC_ZERO_MONITORING = 0
2. 原点マーク監視を起動:
   MD36312 $MA_ENC_ABS_ZEROMON_WARNING = 1
3. 軸を移動し、システム変数$VA_ENC_ZERO_MON_ERR_CNT (検出された制限値違反の回数)を監視します。
4. If $VA_ENC_ZERO_MON_ERR_CNT ≠ 0:
   MD36312 の値を増やして手順 3 を繰り返します。
5. $VA_ENC_ZERO_MON_ERR_CNT = 0(より長時間にわたる)の場合:
   MD36310 の正しい値が見つかりました。MD36312 の値を MD36310 に適用し、MD36312 を「0」に設定します。

注記

機械の剛性(最小負荷量/慣性モーメントが最適かどうか)とコントローラの設定内容に応じて、コントローラの遊びは、さまざまな強度で「振動」します。これを考慮するために、機械毎の制限値を MD36310 に入力してください。

異常時

アラーム 25020

有効な検出器で妥当性チェックが異常を検出すると、アラーム 25020 が表示されます。

「軸<軸名称>有効なエンコーダの原点マーク監視」

対応する軸が、下記の MD に設定された減速カーブによってフォローアップモードで停止します。

MD36610 $MA_AX_EMERGENCY_STOP_TIME
(異常発生時の減速カーブの最大時間)

アラーム 25021

不使用検出器で妥当性チェックが異常を検出すると、アラーム 25021 が表示されます。

「軸<軸名称>不使用エンコーダの原点マーク監視」

これ以外のアラーム応答はありません。

注記

障害が発生すると、絶対値エンコーダの位置合わせが失われ、軸は原点確立済みではなくなります。絶対値エンコーダを再調整してください(「絶対値エンコーダの原点確立 (ページ 1531)」の章を参照してください)。

注記

振幅監視で検出できないインクレメンタルトラックの異常によって、ミリメートル領域の位置偏差が発生する可能性があります。この偏差は、格子ピッチ/ライン数と、異常発生時の軸の移動速度に依存します。

完全な位置監視は、独立した第 2 の検出器との比較のような、多重化によってのみ可能です。

3.5.3 カスタマイズされたアラーム応答

カスタマイズされた原点マーク監視

原点監視の規定のアラームと応答動作を、システム変数を使用して絶対値検出器(MD30240 $MA_ENC_TYPE = 4)で設定することができます。 これによって、シンクロナイズドアクションまたは OEM アプリケーションを使った独自の監視をおこなうことができ、このアプリケーションで使用可能な次のような応答オプションをすべて使用できます。

- アラームの送信
- サイクルの使用(工具交換位置へのアプローチなど)
- ...

例:

アラームの結果として軸が停止すると損傷する恐れがある高価なワークを加工する場合、ワークの加工品質が評価されるまで加工が停止するように、ユーザが適切なシンクロナイズドアクション命令を使用してアラームと応答動作を調整できます。

有効性

カスタマイズされた監視は、マシンデータの設定に応じて、標準原点マーク監視と同時または置き換えて有効にできます。

MD36310 $MA_ENC_ZERO_MONITORING

値	意味
0	ユーザー専用の監視だけを実現する場合、規定の原点監視を無効にしてください。 MD36310 = 0 と MD36312 = 0
> 0	カスタマイズされた監視と標準原点マーク監視が同時動作
100	すべてのエンコーダ監視機能が無効

両方の監視機能が有効な場合 (MD36310 > 0) 、**階段型の監視**をおこなうことができます。

例:

値が MD36310 で指定された検出値を下回ると、カスタマイズされた監視によって警告が出力されます。標準原点マーク監視は、検出値を超えた場合だけ障害を検出し、その後自動的に無効にします。

システム変数

次のシステム変数を使用して、カスタマイズされたアラーム応答を実施できます。

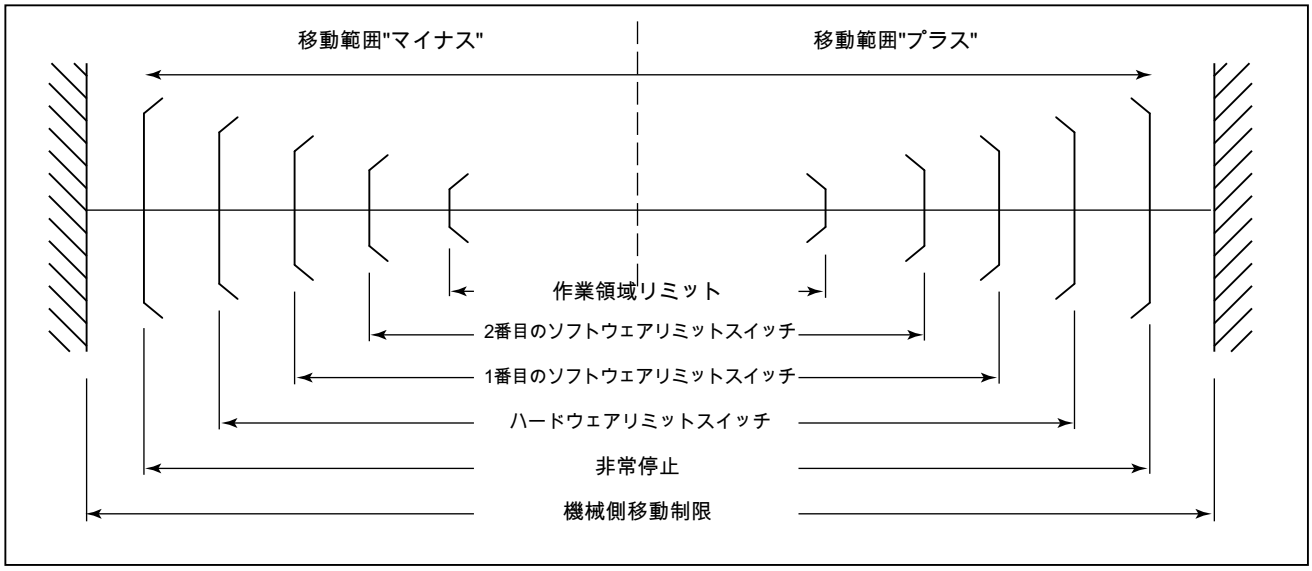
システム変数	意味
$VA_ENC_ZERO_MON_ERR_CNT[<n>,<軸>]	検出された制限値違反の回数 絶対値エンコーダとインクリメンタルエンコーダのトラックを比較したときに検出された現在の制限値違反の回数が含まれます。 値は以下の場合に 0 にリセットされます: ● 電源投入 ● パーキングの選択/選択解除 リセットによってリセットされることはありません。
$VA_ABSOLUTE_ENC_DELTA_INIT[<n>,<軸>]	絶対値エンコーダの初期差分 スタティック NC メモリ内の最後に保持されたアブソリュート位置と現在のアブソリュート位置との間の初期差分が含まれます。 差分値のフォーマット: 内部分割数 (MD10200 $MN_INT_INCR_PER_MM or MD10210 $MN_INT_INCR_PER_DEG を参照してください) 値は以下の場合に更新されます: ● 電源投入 ● ウォーム再起動 ● パーキングの選択解除 ● エンコーダ限界周波数を下回ったことを返すときリセットでリセットはおこなわれません。

<n>: エンコーダ番号

<軸>:軸名称

3.6 リミットスイッチ監視

終端と使用できるリミットスイッチ監視の概要



3.6.1 ハードウェアリミットスイッチ

機能

ハードウェアリミットスイッチは通常、機械軸の移動範囲の終端に取り付けられます。これは、機械軸がまだ原点確立済みではないときに、機械軸が誤ってその最大移動範囲を超えて移動することを防止する役割を果たします。

ハードウェアリミットスイッチが動作すると、工作機械メーカーが作成した PLC ユーザプログラムによって、対応するインタフェース信号が設定されます。

DB31、... DBX12.0/1 = 1 (ハードウェアリミットスイッチ マイナス/プラス)

パラメータ設定

ハードウェアリミットスイッチへの到達直後の機械軸の減速動作は、マシンデータを介して設定できます。

MD36600 $MA_BRAKE_MODE_CHOICE (ハードウェアリミットスイッチでの減速動作)

値	意味
0	設定されている軸加速度での減速
1	急停止(指令速度 = 0)

有効性

ハードウェアリミットスイッチ監視は、コントローラが起動した後、すべてのモードで有効になります。

働き

ハードウェアリミットスイッチに達すると、次の処理がおこなわれます。

- アラーム 21614「チャネル<チャネル番号>軸<軸名称>ハードウェアリミットスイッチ<方向>」
- 設定されている減速動作に従って、機械軸を減速します。
- 軸/主軸が他の軸/主軸との補間に関与する場合は、それらに設定されている減速動作に従って減速されます。
- 方向に基づいて、対応する機械軸の移動キーが無視されます。

3.6.2 ソフトウェアリミットスイッチ

機能

ソフトウェアリミットスイッチは、機械軸の移動範囲を制限する役割を果たします。機械軸毎および移動方向毎に、2 つ(1 番目と 2 番目)のソフトリミットスイッチが使用可能です。

MD36100 POS_LIMIT_MINUS (1 番目のマイナスのソフトウェアリミットスイッチ)

MD36110 POS_LIMIT_PLUS (1 番目のプラスのソフトウェアリミットスイッチ)

MD36120 POS_LIMIT_MINUS2 (2 番目のマイナスのソフトウェアリミットスイッチ)

MD36130 POS_LIMIT_PLUS2 (2 番目のプラスのソフトウェアリミットスイッチ)

初期設定では、1 番目のソフトウェアリミットスイッチが有効です。2 番目のソフトウェアリミットスイッチは、PLC ユーザプログラムによって、特定の方向について有効にできます。

DB31、... DBX12.2 / 12.3 (2 番目のソフトウェアリミットスイッチ マイナス/プラス)

有効性

次の条件で、ソフトウェアリミットスイッチが有効になります。

- 機械軸が正常に原点確立された直後から
- すべての運転モードで

必要条件

- ソフトウェアリミットスイッチでは機械座標系が基準になります。
- ソフトウェアリミットスイッチは、ハードウェアリミットスイッチの内側に設定してください。
- 機械軸は、有効なソフトウェアリミットスイッチの位置まで移動できます。
- PRESET
  PRESET 機能の使用後は、ソフトウェアリミットスイッチ監視が無効になります。最初に機械を再原点確立してください。
- 移動量制限のない回転軸
  移動量制限のない回転軸に対しては、ソフトウェアリミットスイッチ監視は実行されません。
  MD30310 $MA_ROT_IS_MODULO == 1 (回転軸と主軸のモジュロ変換)
  例外:セットアップ回転軸

動作

自動運転モード(AUTOMATIC、MDI)

- 座標変換なし、重畳移動なし、ソフトウェアリミットスイッチ変更なしの場合:
  ソフトウェアリミットスイッチのオーバーランをもたらす移動動作がプログラム指令されたパートプログラムブロックはスタートしません。
- 座標変換がある場合:
  座標変換のタイプに応じて異なる応答が発生:
  - 上記のとおりの動作
    または
  - ソフトウェアリミットスイッチのオーバーランをもたらす移動動作がプログラム指令されたパートプログラムブロックはスタートします。対応する機械軸が、有効なソフトウェアリミットスイッチ上で停止します。移動動作に関わる他の機械軸は減速します。この処理では、プログラム指令輪郭が保たれます。
- 重畳動作がある場合
  ソフトウェアリミットスイッチのオーバーランをもたらす移動動作がプログラム指令されたパートプログラムブロックはスタートします。重畳動作で移動している、または重畳動作で移動した機械軸が、該当する有効なソフトウェアリミットスイッチ上で停止します。移動動作に関わる他の機械軸は減速します。この処理では、プログラム指令輪郭が保たれます。

手動運転モード

- 座標変換のない JOG
  機械軸はソフトウェアリミットスイッチ上で停止します。
- 座標変換のある JOG
  機械軸はソフトウェアリミットスイッチ上で停止します。移動動作に関わる他の機械軸は減速します。この処理では既定の軌跡が保たれます。

一般事項

- ソフトウェアリミットスイッチの変更(1 番目 ↔ 2 番目のソフトウェアリミットスイッチ)
  変更後の機械軸の現在位置がソフトウェアリミットスイッチの外側にある場合は、機械軸が最高許容加速度で停止します。
- JOG モードでのソフトウェアリミットスイッチのオーバーラン
  ソフトウェアリミットスイッチの位置に達していて、新たに移動ボタンを押すとその方向にさらに移動することになる場合は、下記のアラームが表示され、軸はそれ以上移動しません。
  アラーム 10621「チャネル<チャネル番号>軸<軸名称>はソフトウェアリミット<スイッチ方向>上にあります」

3.7 作業領域リミット監視

3.7.1 全般

機能

「作業領域リミット」機能を使用すると、チャネルのジオメトリ軸と付加軸の移動範囲を、許容動作範囲内に制限できます。 この機能は、AUTOMATIC モードと JOG ジョグモードの両方で、作業領域リミットに適合しているかを監視します。

次の種類が使用可能です。

- 基本座標系(BCS)での作業領域リミット
  移動範囲制限を、基本座標系に対して指定します。
- ワーク座標系(WCS)または設定可能原点系(AZS)での作業領域リミット
  移動範囲制限を、ワーク座標系または設定可能原点系に対して指定します。

この 2 つのタイプの監視は互いに独立しています。 これらが同時に動作中の場合は、移動方向に応じて、アクセスを最大に制限する移動範囲制限が有効になります。

工具のレファレンス点

工具データ(工具長と工具半径)と、作業領域リミットの監視時の工具のレファレンス点をどのように考慮するかは、チャネルでの座標変換状態により異なります。

- **座標変換をおこなわない場合**
  有効な工具での移動動作時に座標変換がおこなわれない場合は、工具先端 P の位置が監視されます。つまり、監視中には工具長が自動的に考慮されます。
  工具径の考慮は別個に有効にしてください。
  MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS(作業領域リミットでの工具径の考慮)
- **座標変換有効**
  特定の座標変換がおこなわれる場合は、下記のように、作業領域リミットの監視が座標変換のない場合の動作と異なることがあります。
  - 工具長が座標変換の成分のとき ($MC_TRAFO_INCLUDES_TOOL_X = TRUE):
    この場合、工具長は考慮されません。つまり、監視では工具ホルダのレファレンス点を基準にします。
  - 向きの変更をともなう座標変換のとき:
    座標変換で向きを変更した場合は常に、工具中心点に基づいて監視します。
    MD21020 は影響を及ぼしません。

  注記

  マシンデータ$MC_TRAFO_INCLUDES_TOOL_...は、特定の座標変換のみで解析されます。 考えられる使用の条件は、基本座標系に対する工具の向きが、座標変換によって変更できない場合です。 標準座標変換の場合は、「傾斜軸」形式の座標変換の条件のみが満たされます。

動作

自動運転モード

- 座標変換あり/なしの両方
  作業領域リミットのオーバーランをもたらす移動動作がプログラム指令されたパートプログラムブロックは実行されません。
- 重畳動作がある場合
  重畳移動によって作業領域リミットに違反した軸は、最高加速度で、加々速度制限なしで減速され(BRISK)、作業領域リミットの位置で停止します。 移動に関与するその他の軸は、現在の加速動作に従って減速されます(SOFT など)。 軌跡の相関関係は、減速加速度が異なるために失われる場合があります(経路干渉)。

手動運転モード

- 座標変換あり/なし両方の JOG
  軸は作業領域リミットの位置に移動して停止します。

電源投入動作

作業領域リミットの起動時に、許容される作業領域外に軸が移動した場合、軸は最高許容加速度で直ちに停止します。

JOG モードでの作業領域リミットのオーバーラン

JOG モードでは、コントロールシステムが、軸をその作業領域リミットを超えて移動することはありません。 移動ボタンを再度押すと、アラームが表示され、軸がそれ以上は移動しません。

ジオメトリ軸切替

次のマシンデータを介して、ジオメトリ軸の切り替え時に有効な作業領域リミットを保持するか無効にするかを選択できます。

MD10604 $MN_WALIM_GEOAX_CHANGE_MODE = <値>

<値>	意味
0	ジオメトリ軸の切り替え時に作業領域リミットが無効になります。
1	ジオメトリ軸の切り替え時に作業領域リミットが有効なままになります。

3.7.2 BCS におけるワーキングエリアリミット

用途

「BCS での作業領域リミット」を使用すると、周囲の機器(工具リボルバ、測定ステーションなど)が損傷から保護されるように、工作機械の作業領域が制限されます。

作業領域リミット

各軸の作業領域リミットの上限と下限は、セッティングデータを介して設定されるか、パートプログラム命令を介してプログラムします。

セッティングデータによる作業領域リミット

設定は、即時有効になる軸セッティングデータを介しておこなわれます。

SD43420 $SA_WORKAREA_LIMIT_PLUS (正の作業領域リミット)

SD43430 $SA_WORKAREA_LIMIT_MINUS (負の作業領域リミット)

プログラム指令された作業領域リミット

このプログラミングは G 命令を使用しておこなわれます。

G25 X… Y… Z… 作業領域リミットの下限

G26 X… Y… Z… 作業領域リミットの上限

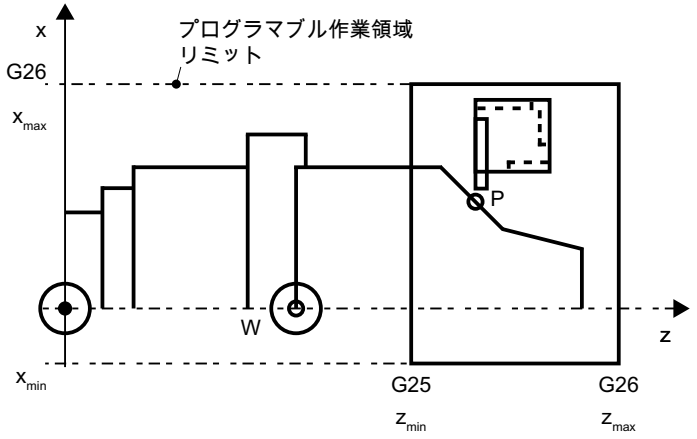


図 3-4　プログラム指令された作業領域リミット

プログラム指令された作業領域リミットが優先され、SD43420 と SD43430 に入力された値が上書きされます。

起動/解除

セッティングデータによる作業領域リミット

各軸の作業領域リミットの起動/解除は、即時有効になるセッティングデータを介して、方向別に行われます。。

SD43400 $SA_WORKAREA_PLUS_ENABLE (正方向の作業領域リミットが有効)

SD43410 $SA_WORKAREA_MINUS_ENABLE (負方向の作業領域リミットが有効)

値	意味
0	正方向または負方向の作業領域リミットが**オフ**になります。
1	正方向または負方向の作業領域リミットが**有効**になります。

プログラム指令された作業領域リミット

「BCS での作業領域リミット」全体の起動または解除は、以下のパートプログラム命令によって行われます。

WALIMON	作業領域リミットオン
または	
WALIMOF	作業領域リミットオフ

作業領域リミットの変更

セッティングデータによる作業領域リミット

HMI 操作画面:操作エリア「パラメータ」

- 自動モード:
  - 変更:リセット状態のみで可能
  - 有効性:即時
- 手動運転モード:
  - 変更:常に可能
  - 有効性:次の移動動作の開始時

プログラム指令された作業領域リミット

作業領域リミットは、G25 または G26 **<**軸名称**> <**値**>**を使用してパートプログラムで変更することができます。この変更は即時有効になります。

新しい作業領域リミット値は、SD43420 と SD43430 用の NC 保持データ記憶域でバックアッププロセスを起動した場合は、NC リセットと電源投入の後も保持されます。

MD10710 $MN_PROG_SD_RESET_SAVE_TAB[0] = 43420

MD10710 $MN_PROG_SD_RESET_SAVE_TAB[1] = 43430

リセット設定

作業領域リミットのリセット設定(WALIMON または WALIMOF)は、以下を使用して構成できます。

MD20150 $MC_GCODE_RESET_VALUES (G グループのリセット設定)

3.7.3 WCS/SZS における作業領域リミット

用途

「WCS/SZS の作業領域リミット」により、ワーク座標系(WCS)または設定可能ゼロオフセットシステム(SZS)におけるチャネル軸の移動範囲をチャネル別に、ワーク毎に柔軟に制限することができます。.これは主として、従来の旋盤で使用するためのものです。

必要条件

チャネル軸を原点確立してください。

作業領域リミットグループ

座標変換の切り換えや有効なフレームのオン/オフ時などのように、軸割り付けを変更するときに、すべてのチャネル軸に対して軸固有の作業領域リミットを書き換えなくてもよいように、作業領域リミットグループが用意されています。

作業領域リミットグループには、以下のデータが含まれます。

- すべてのチャネル軸の作業領域リミット
- 作業領域リミットの基準座標系

作業領域リミットグループの数は、次のマシンデータでチャネル別に設定されます。

MD28600 $MC_MM_NUM_WORKAREA_CS_GROUPS

チャネルごとに設定可能な作業領域リミットグループは最大で 10 個です。

作業領域リミットの設定

チャネル内の作業領域リミットは、各チャネル軸について、以下のシステム変数によって定義されます。

- $P_WORKAREA_CS_LIMIT_PLUS[<WALimNo>, <Ax>]
- $P_WORKAREA_CS_LIMIT_MINUS[<WALimNo>, <Ax>]

意味: <WALimNo> = 作業領域リミットグループ
値の範囲: 0 (グループ 1) ... 9 (グループ 10)
<Ax> = チャネル軸の名称

作業領域リミットの有効化

チャネル内の作業領域リミットは、各チャネル軸について、以下のシステム変数によって有効になります。

- $P_WORKAREA_CS_PLUS_ENABLE[<WALimNo>, <Ax>]
- $P_WORKAREA_CS_MINUS_ENABLE[<WALimNo>, <Ax>]

意味: **<WALimNo> =** 作業領域リミットグループ

値の範囲: 0 (グループ 1) ... 9 (グループ 10)

<Ax> = チャネル軸の名称

方向別に有効化を実行すると、軸の作業範囲を 1 方向のみに制限できます。

イネーブルによる適用はありません。

基準座標系の選択

チャネル内の作業領域リミットの基準座標系は、以下のシステム変数によって設定されます。

$P_WORKAREA_CS_COORD_SYSTEM[<WALimNo>] = *<値>*

意味: **<WALimNo> =** 作業領域リミットグループ

値の範囲: 0 (グループ 1) ... 9 (グループ 10)

<値>	意味
1	基準座標系は WCS です。
3	基準座標系は SZS です。

作業領域リミットの適用

作業領域リミットグループの作業領域リミットは、G 命令 WALCS<n>によってパートプログラムで適用されます。

意味: <n> = 作業領域リミットグループの番号

値の範囲: 1 ... 10

作業領域リミットの解除

チャネルで有効な作業領域リミットグループの作業領域リミットは、G 命令 WALCS0 によってパートプログラムで解除されます。

作業領域リミットの変更

作業領域リミットは、上述のシステム変数を使用していつでも変更することができます。変更は、作業領域リミットグループが次に適用されたときに(WALCSn)有効になります。

データ記憶

作業領域リミットのシステム変数は、NC のスタティックメモリに確実に保存されます。

注記

直線軸の制限値の記憶については、単位系の初期設定が考慮されます(MD10240 $MN_SCALING_SYSTEM_IS_METRIC)。

データバックアップ

作業領域リミットのシステム変数は、別々のファイルにバックアップできます。

- _N_CHx_WAL
  チャネル x のシステム変数値を保存します。
- _N_COMPLETE_WAL
  すべてのチャネルのシステム変数値を保存します。

注記

作業領域リミットのシステム変数は、"_N_INITIAL_INI"ファイルに入っています。

JOG モードでの動作

初期状態:

- JOG モードでは、**複数の** ジオメトリ軸が同時に移動します(たとえば、複数のハンドルを使用して)。
- **回転**フレームは、基本座標系(BCS)と作業領域リミットの基準座標系(WCS またはSZS)の間で有効になります。

作業領域リミットの検出時の動作:

- 影響を受けないジオメトリ軸の移動動作は継続されます。
- 影響を受けるジオメトリ軸は、作業領域リミットで停止します。

初期設定の設定

電源投入時、リセット時、またはパートプログラムの終了時と開始時に有効になる作業領域リミットグループは、次のマシンデータによってチャネル別に事前設定されています。

MD20150 $MC_GCODE_RESET_VALUE[59] = <n>

意味: **<n> =** 作業領域リミットグループの番号

値の範囲: 1 ... 10

次の設定により、事前に選択された作業領域リミットが、電源投入時とリセット時またはパートプログラムの終了時に作動するかどうかが決定されます。

MD20152 $MC_GCODE_RESET_MODE[59] = *<値>*

<値>	意味
0	作業領域グループは、MD20150(初期設定)に従って有効になります。
1	最後に有効になった作業領域グループが、そのまま有効です。

3.7.4 例:WCS/SZS における作業領域リミット

前提条件

チャネル軸

チャネル内では、次の 4 つの軸が定義されています。

- 直線軸:X、Y、Z
- 回転軸:A (モジュロなし)

必要条件

チャネル軸

チャネル内では、次の 4 つの軸が定義されています。

- 直線軸:X、Y、Z
- 回転軸:A (モジュロなし)

ワーキングエリアリミットグループ

3 つの作業領域リミットグループがチャネルで有効であることが必要です。

MD28600 $MC_MM_NUM_WORKAREA_CS_GROUP = 3

以下では、この 3 つの作業領域リミットグループから、2 つのグループを定義しています。

座標系

- 作業領域リミットグループ 1:設定可能ゼロオフセットシステム(**AZS**)の作業領域リミット。
- 作業領域リミットグループ 2:ワーク座標系(**WCS**)の作業領域リミット

作業領域リミットグループ 1

- X 軸のプラス方向:10 mm
- X 軸のマイナス方向:制限なし
- Y 軸のプラス方向:制限なし
- Y 軸のマイナス方向:25 mm
- Z 軸のプラス方向:制限なし
- Z 軸のマイナス方向:制限なし
- A 軸のプラス方向:10°
- A 軸のマイナス方向: -40 度

NC プログラムのシステム変数による定義。

プログラムコード

```
;作業領域リミットグループ 1
$P_WORKAREA_CS_COORD_SYSTEM[1] = 3          ; SZS の作業領域リミット
$P_WORKAREA_CS_PLUS_ENABLE[1,X] = TRUE
$P_WORKAREA_CS_LIMIT_PLUS[1,X] = 10
```

プログラムコード

```
$P_WORKAREA_CS_MINUS_ENABLE[1,X] = FALSE
$P_WORKAREA_CS_PLUS_ENABLE[1,Y] = FALSE
$P_WORKAREA_CS_MINUS_ENABLE[1,Y] = TRUE
$P_WORKAREA_CS_LIMIT_MINUS[1,Y] = 25
$P_WORKAREA_CS_PLUS_ENABLE[1,Z] = FALSE
$P_WORKAREA_CS_MINUS_ENABLE[1,Z] = FALSE
$P_WORKAREA_CS_PLUS_ENABLE[1,A] = TRUE
$P_WORKAREA_CS_LIMIT_PLUS[1,A] = 10
$P_WORKAREA_CS_MINUS_ENABLE[1,A] = TRUE
$P_WORKAREA_CS_LIMIT_MINUS[1,A] = -40
```

作業領域リミットグループ 2

- X 軸のプラス方向:10 mm
- X 軸のマイナス方向:制限なし
- Y 軸のプラス方向:34 mm
- Y 軸のマイナス方向: -25 mm
- Z 軸のプラス方向:制限なし
- Z 軸のマイナス方向: -600 mm
- A 軸のプラス方向:制限なし
- A 軸のマイナス方向:制限なし

NC プログラムのシステム変数による定義。

プログラムコード

```
;作業領域リミットグループ 2
$P_WORKAREA_CS_COORD_SYSTEM[2] = 3          ; WCS の作業領域リミット
$P_WORKAREA_CS_PLUS_ENABLE[2,X] = TRUE
$P_WORKAREA_CS_LIMIT_PLUS[2,X] = 10
$P_WORKAREA_CS_MINUS_ENABLE[2,X] = FALSE
$P_WORKAREA_CS_PLUS_ENABLE[2,Y] = TRUE
$P_WORKAREA_CS_LIMIT_PLUS[2,Y] = 34
$P_WORKAREA_CS_MINUS_ENABLE[2,Y] = TRUE
$P_WORKAREA_CS_LIMIT_MINUS[2,Y] = -25
$P_WORKAREA_CS_PLUS_ENABLE[2,Z] = FALSE
$P_WORKAREA_CS_MINUS_ENABLE[2,Z] = TRUE
$P_WORKAREA_CS_LIMIT_PLUS[2,Z] = -600
$P_WORKAREA_CS_PLUS_ENABLE[2,A] = FALSE
```

プログラムコード

```
$P_WORKAREA_CS_MINUS_ENABLE[2,A] = FALSE
```

起動

作業領域リミットグループは、NC プログラムで、命令 WALCS<x>を使用して起動します。x:作業領域リミットグループの番号

3.8 機械軸のパーキング

機械軸を「パーキング」状態にすると、この特定の軸に対してエンコーダなしの現在値が取得され、前の章に記載したすべての監視機能(検出器、停止状態、クランプ監視など)が無効になります。

起動と解除

パーキングの起動

位置検出器およびコントローライネーブルの軸別 NC/PLC インタフェース信号を**リセット**することにより、機械軸に対して[パーキング]ファンクションを**起動**します。

- DB31、... DBX1.5 = 0 (位置検出器 1)
- DB31、... DBX1.6 = 0 (位置検出器 2)
- DB31、... DBX2.1 = 0 (コントローライネーブル)

これで、軸の位置検出器のエンコーダ状態は「原点未確立」に設定されます。

- DB31、... DBX60.4 = 0 (原点確立済み/原点同期済み、位置検出器 1)
- DB31、... DBX60.5 = 0 (原点確立済み/原点同期済み、位置検出器 2)

さらに次の NC/PLC インターフェース信号もリセットされます。

- DB31、... DBX61.5 = 0 (位置コントローラ有効)
- DB31、... DBX61.6 = 0 (速度コントローラ有効)
- DB31、... DBX61.7 = 0 (電流コントローラ有効)
- DB31、... DBX93.7 = 0 (パルスイネーブル中)
- DB31、... DBX102.5 = 0 (位置検出器 1 有効)
- DB31、... DBX102.6 = 0 (位置検出器 2 有効)

パーキングの解除

位置検出器およびコントローライネーブルの軸別 NC/PLC インタフェース信号を**設定**することにより、機械軸に対して[パーキング]ファンクションを**解除**します。

- DB31、... DBX1.5 (位置検出器 1) = 1
  または
  DB31、... DBX1.6 = 1 (位置検出器 2) = 1
- DB31、... DBX2.1 = 1 (コントローライネーブル) = 1

機械軸の位置制御が現在の位置で再度有効になります。

位置検出器のエンコーダ状態は、検出器のタイプによって異なります。

- インクリメンタル位置検出器⇒「原点未確立」状態
  - DB31、... DBX60.4 = 0 (原点確立済み/原点同期済み、位置検出器 1)
  - DB31、... DBX60.5 = 0 (原点確立済み/原点同期済み、位置検出器 2)
- アブソリュート位置検出器⇒「原点確立済み/原点同期済み」状態
  - DB31、... DBX60.4 = 1 (原点確立済み/原点同期済み、位置検出器 1)
  - DB31、... DBX60.5 = 1 (原点確立済み/原点同期済み、位置検出器 2)

次の NC/PLC インターフェース信号も再設定されます。

- DB31、... DBX61.5 = 1 (位置コントローラ有効)
- DB31、... DBX61.6 = 1 (速度コントローラ有効)
- DB31、... DBX61.7 = 1 (電流コントローラ有効)
- DB31、... DBX93.7 = 1 (パルスイネーブル中)
- DB31、... DBX102.5 = 1 (位置検出器 1 有効)
- DB31、... DBX102.6 = 1 (位置検出器 2 有効)

インクリメンタル位置検出器

「パーキング」状態を無効にした後、「原点確立済み」エンコーダ状態になる前にインクリメンタル位置検出器を原点確立してください。

⚠ 警告

実際の機械軸位置のオフセットを原因とする、位置検出器の誤った同期

「パーキング」中に、設定されたマシンデータの変更が必要となるような変更(別のエンコーダの取り付けなど)が位置検出器に対して行われた場合などは、位置検出器を完全に再測定して原点確立してください(「R1:レファレンス点復帰 (ページ 1497)」の章を参照)。

位置検出器のない機械軸

位置検出器のない機械軸(速度制御主軸)の場合は、コントローライネーブルを取り消すと、「パーキング」に相当する状態になります。

- DB31、... DBX2.1 = 0 (コントローライネーブル)

3.9 パッシブな位置検出器のパーキング

3.9.1 機能

機械軸のすべての位置検出器が無効になる「機械軸のパーキング (ページ 145)」機能とは逆に、「パッシブな位置検出器のパーキング」機能を使用すると、アクティブな位置検出器が作動中のままで、機械軸のパッシブな位置検出器のみを「パーキング」する(つまり、エンコーダ評価、およびドライブとコントローラでの監視を無効にする)オプションを使用できます。

注記

アクティブ/パッシブな検出器の説明については、「指令/フィードバック回路 (ページ 428)」を参照してください。

用途

「パッシブな検出器のパーキング」機能は、次のような場合に使用できます。

- 内蔵エンコーダ付きまたは内蔵エンコーダなしの装着ヘッドの交換
  「パッシブな位置検出器のパーキング」機能を使用すると、内蔵エンコーダ付きまたは内蔵エンコーダなしの装着ヘッドを交互に取り付けて、ドライブとコントロールの障害を開始するエンコーダ信号が欠落することなく、主軸でのさまざまな機械加工処理を行うことができます。
  以下も参照ください:
  - 例:直接位置検出器の装着ヘッドの交換 (ページ 152)
  - 例:2 台の直接位置検出器の装着ヘッドの交換 (ページ 157)
- 機械軸の移動範囲全体では使用できないリニア位置検出器の使用
  「パッシブな位置検出器のパーキング」機能を使用すると、ドライブとコントロールの障害を開始するエンコーダ信号が欠落することなく、リニア位置検出器の範囲を通過することができます。
  以下も参照ください:
  - "例:範囲の一部にエンコーダがない場合の検出器の切り替え (ページ 162)".

起動と解除

起動

機械軸の位置検出器は、次の場合にパーキングされます。

- 「パッシブな位置検出器のパーキング」機能は、次の検出器で有効です。
  MD31046 $MA_ENC_PASSIVE_PARKING[<n>] = 1
  ここで、<n> = 0 (位置検出器 1)または 1 (位置検出器 2)

注記

MD31046 は、次の場合は**有効 ではありません**。

- 軸のエンコーダが 2 つ未満:
  MD30200 $MA_NUM_ENCS < 2
- シミュレートされたエンコーダ:
  MD30240 $MA_ENC_TYPE = 0

注記

モータ測定システムとして使用される位置検出器の場合は、「パッシブな位置検出器のパーキング」機能を無効にしてください(MD31046 = 0)。

および

- ユーザーが次の NC/PLC インターフェース信号を「0」に設定している:
  DB31、... DBX1.5 (位置検出器 1) = 0
  または
  DB31、... DBX1.6 (位置検出器 2) = 0

その後、コントローラによって位置検出器の有効化状態が「0」に設定される:

DB31、... DBX102.5 (位置検出器 1 有効化済み) = 0

または

DB31、... DBX102.6 (位置検出器 2 有効化済み) == 0

これで、位置検出器の監視と更新が行われなくなります。

解除

ユーザーが位置検出器を有効にしている場合は、「パーキング」が無効になります。

DB31、... DBX1.5 (位置検出器 1) = 1

または

DB31、... DBX1.6 (位置検出器 2) = 1

その後、コントローラによって位置検出器の有効化状態が「1」に設定されます。

DB31、... DBX102.5 (位置検出器 1 有効化済み) = 1

または

DB31、... DBX102.6 (位置検出器 2 有効化済み) = 1

注記

パーキングされた位置検出器への切り替えは、パーキングされていない位置検出器への切り替えより時間がかかります。時間がかかるため、軸が停止している間に切り替えることを推奨します。

位置検出器の配置

アブソリュート位置検出器

アブソリュート位置検出器の場合、「パーキング」を無効にした後の位置は、実際のアブソリュートエンコーダ位置に対応します。

位置検出器は原点確立されています。

DB31、... DBX60.4 (原点確立済み/原点同期済み、位置検出器 1) == 1

または

DB31、... DBX60.5 (原点確立済み/原点同期済み、位置検出器 2) == 1

インクリメンタル位置検出器

インクリメンタル位置検出器の場合、「パーキング」を無効にした後の位置は常に、位置検出器を最後に無効にした位置に対応します。

パーキングした位置検出器への切り替えは、2 つの位置検出器の実際値の間のパラメータ設定された許容誤差(MD36500 $MA_ENC_CHANGE_TOL を参照)を超えない場合にのみ実行されます。そうでない場合は、ユーザーが「機械軸のパーキング」機能を適用する必要がありますが、そのチェックは行われません。

位置検出器は**原点確立されていません**。

DB31、... DBX60.4 (原点確立済み/原点同期済み、位置検出器 1) == 0

または

DB31、... DBX60.5 (原点確立済み/原点同期済み、位置検出器 2) == 0

位置転送を行うインクリメンタル位置検出器

他にも、インクリメンタル位置検出器の場合は、「パッシブな位置検出器のパーキング」機能が有効(MD31046 $MA_ENC_PASSIVE_PARKING[<n>] = 1)なときは、「パーキング」を無効にして位置を転送することができます。該当する場合は、以前に有効であった位置検出器から「原点確立済み」状態を転送できます。

機械軸のすべての位置検出器について、次のマシンデータを使用して、この機能を有効にすることができます。

MD34210 $MA_ENC_REFP_STATE[<n>]

ここで、<n> = 0 (位置検出器 1)または 1 (位置検出器 2)

値	意味
1	以前に有効であった位置検出器からの位置のみが転送されます。 位置検出器は**原点確立されていません**。 DB31、... DBX60.4 (原点確立済み/原点同期済み、位置検出器 1) == 0 または DB31、... DBX60.5 (原点確立済み/原点同期済み、位置検出器 2) == 0
2	位置と「原点確立済み」状態が、以前に有効であった位置検出器から転送されます 位置検出器は原点確立されています。 DB31、... DBX60.4 (原点確立済み/原点同期済み、位置検出器 1) == 1 または DB31、... DBX60.5 (原点確立済み/原点同期済み、位置検出器 2) == 1

注記

以前に有効であった位置検出器の位置と「原点確立済み」状態は、MD34210 $MA_ENC_REFP_STATE[<n>]に対応して、インクリメンタル位置検出器の場合にのみ、および「パッシブな位置検出器のパーキング」機能が有効(MD31046 $MA_ENC_PASSIVE_PARKING[<n>] = 1)である場合にのみ転送されます。

転送される位置の精度は、以前に有効であった位置検出器と同じです。精度が適切でない場合は、位置検出器を原点確立してください。

⚠ 警告

実際の機械軸位置のオフセットを原因とする、位置検出器の誤った同期

「パーキング」中に、設定されたマシンデータの変更が必要となるような変更(別のエンコーダの取り付けなど)が位置検出器に対して行われた場合などは、位置検出器を完全に再測定して原点確立してください(「R1:レファレンス点復帰 (ページ 1497)」の章を参照)。

3.9.2 必要条件

「デュアル位置検出」との連携

「パッシブな位置検出器のパーキング」機能は、「デュアル位置検出」機能(MD32960 $MA_POSCTRL_DUAL_FEEDBACK_TIME > 0)と組み合わせて使用することは**できません**。

「位置差異入力」との連携

「パッシブな位置検出器のパーキング」機能は、「位置差異入力」機能(MD32950 $MA_POSCTRL_DAMPING > 0)と組み合わせて使用することは**できません**。

APC(SINUMERIK 840D sl のオプション)との連携

「パッシブな検出器のパーキング」機能は、「高度な位置制御(APC)」ドライブ機能と組み合わせて使用することは**できません**。

エンコーダ安全保護コンセプトとの連携

「パッシブな検出器のパーキング」機能と組み合わせて使用できるエンコーダ安全保護コンセプトは、1 つ**だけ**です。

DRIVE-CLiQ での接続時と接続切断時の連携

SMC とモータモジュール間などで、エンコーダケーブルではなく、DRIVE-CLiQ ケーブルの接続を切断してから接続した場合、このエンコーダのパーキングを正常に解除するには、「機械軸のパーキング (ページ 145)」機能を使用してください。

3.9.3 例:直接位置検出器の装着ヘッドの交換

初期状態

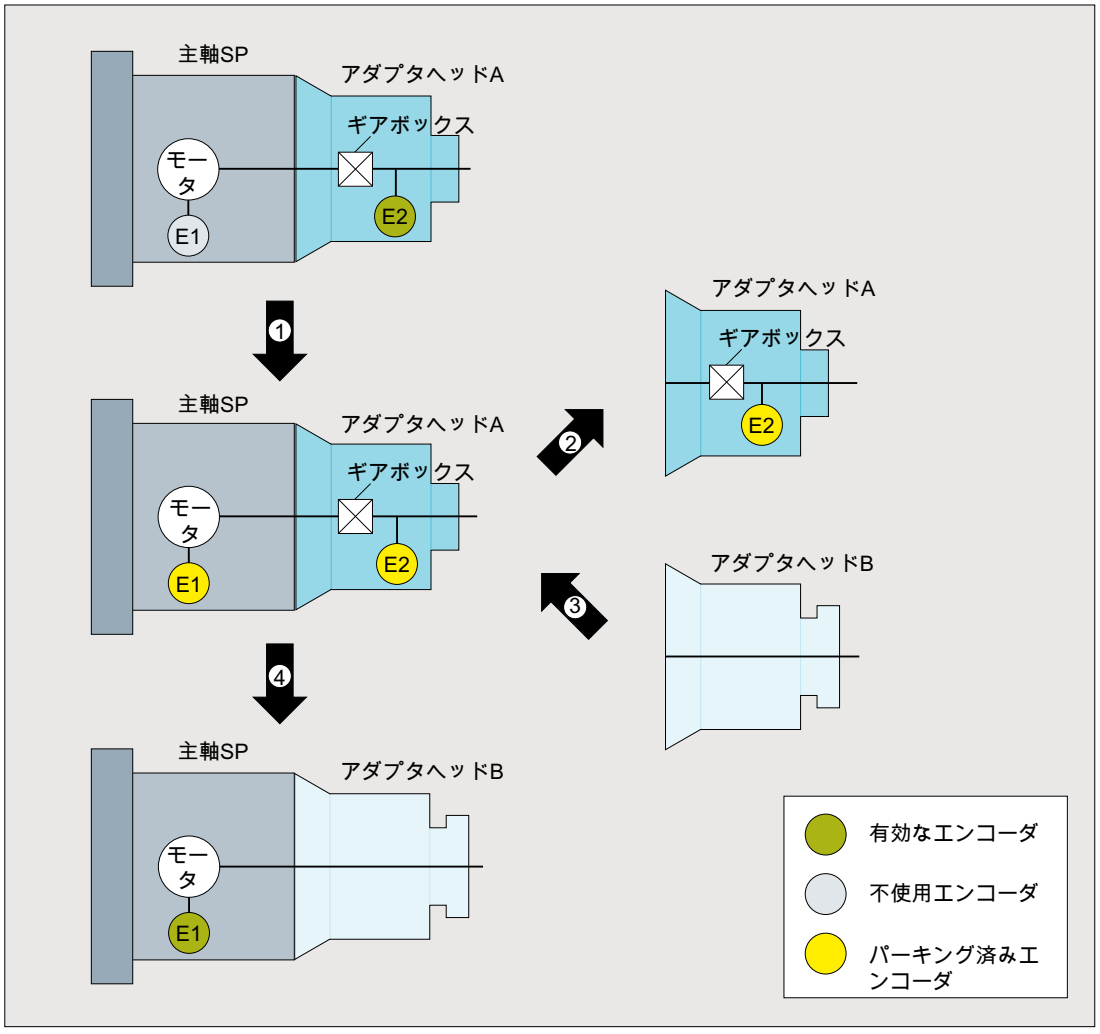
- 装着ヘッド「A」にエンコーダ E2 があります。
- 装着ヘッド「B」にはエンコーダがありません。
- 主軸「SP」にエンコーダ E1 があります。

- 以下の 2 つの通信形式のいずれかが、MD13060 $MN_DRIVE_TELEGRAM_TYPE (PROFIdrive の標準通信形式)に設定されています。
  - メッセージ 116 (モータエンコーダ+別置きエンコーダ)
    または
  - メッセージ 136 (モータエンコーダ+別置きエンコーダ、およびトルクフィードフォワード補償)
- 以下の位置検出器が主軸「SP」で設定されています。
  - 位置検出器 1 としてモータエンコーダ E1
  - 位置検出器 2 として直接エンコーダ E2
- エンコーダ E2 を備えた装着ヘッド「A」は、現在、主軸に取り付けられています。
- 位置検出器 2 が、有効な検出器です。
  DB31、... DBX1.6 = 1
  位置検出器 1 はパッシブです。
- 「パッシブな位置検出器のパーキング」機能は、以下のとおりです。
  - 位置検出器 1 で有効です。
    MD31046 $MA_ENC_PASSIVE_PARKING [0] = 0
  - 位置検出器 2 で有効です。
    MD31046 $MA_ENC_PASSIVE_PARKING [1] = 1

目的

ユーザーが、装着ヘッド「A」を装着ヘッド「B」に交換しようとしています。

実行



① 装着ヘッドを交換する前にユーザーは、「機械軸のパーキング (ページ 145)」機能を使用して、機械軸のすべての位置検出器を無効にしてください。

DB31、... DBX1.5 (位置検出器 1) = 0

DB31、... DBX1.6 (位置検出器 2) = 0

その後、コントローラが、位置検出器の状態信号をリセットします。

DB31、... DBX102.5 (位置検出器 1 有効化済み) = 0

DB31、... DBX102.6 (位置検出器 2 有効化済み) == 0

② ユーザーは、状態信号を待ち、主軸から装着ヘッド「A」を取り外します。これで、装着ヘッド「A」と連結部の間のエンコーダケーブルの電源接続が切断されます。エンコーダ E2 が存在していなくても、NC とドライブのいずれにも異常は発生しません。

③ 装着ヘッド「B」が主軸に取り付けられました。

④ ユーザーは、位置検出器 1 だけを有効にします。

DB31、... DBX1.5 (位置検出器 1) = 1

その後、制御装置が状態信号を設定します。

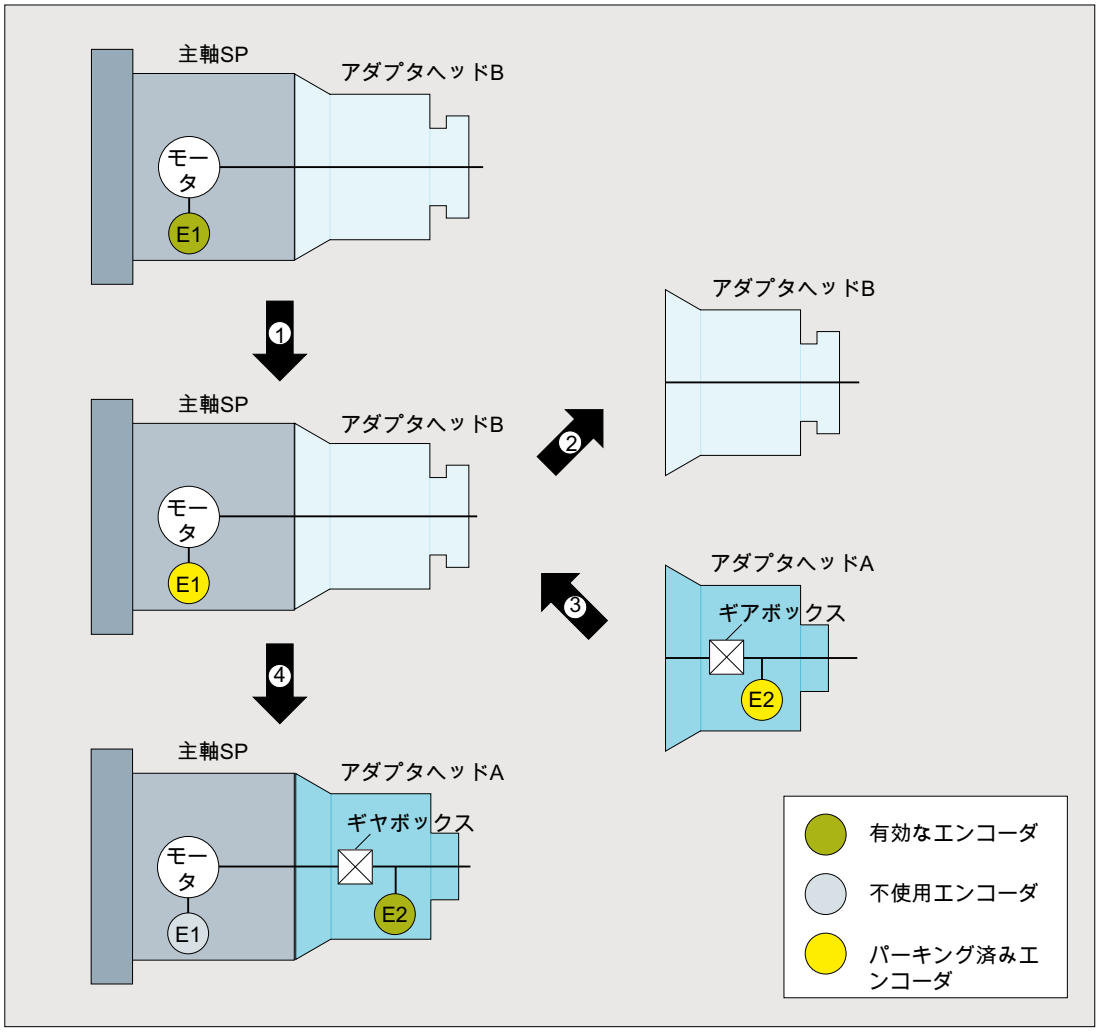
DB31、... DBX102.5 (位置検出器 1 有効化済み) = 1

「パッシブな位置検出器のパーキング」機能が位置検出器 2 で有効です。つまり、位置検出器 2 はパッシブにならず、「パーキング」状態のままです。

目的

ユーザーが、装着ヘッド「A」を再度取り付けようとしています。

実行



① ユーザーが「機械軸のパーキング」機能を使用して、位置検出器 1 を無効にします。

DB31、... DBX1.5 (位置検出器 1) = 0

その後、コントローラが、位置検出器の状態信号をリセットします。

DB31、... DBX102.5 (位置検出器 1 有効化済み) = 0

② ユーザーは、状態信号を待ち、主軸から装着ヘッド「B」を取り外します。

③ 装着ヘッド「A」が主軸に取り付けられました。

④ ユーザーは位置検出器 2 を有効にします。

DB31、... DBX1.6 (位置検出器 2) = 1

結果として、位置検出器 1 も同時に有効になります。これは、「パッシブな位置検出器のパーキング」機能が位置検出器 1 (モータ検出器)で有効でないためです。位置検出器 1 が、パッシブな位置検出器になります。

その後、コントローラが、位置検出器の状態信号を設定します。

DB31、... DBX102.5 (位置検出器 1 有効化済み) = 1

DB31、... DBX102.6 (位置検出器 2 有効化済み) = 1

3.9.4 例:2 台の直接位置検出器の装着ヘッドの交換

初期状態

- 装着ヘッド「A」にエンコーダ E3 があります。
- 装着ヘッド「B」にはエンコーダがありません。
- 主軸「SP」に 2 つのエンコーダ、E1 および E2 があります。
- 以下の 2 つの通信形式のいずれかが、MD13060 $MN_DRIVE_TELEGRAM_TYPE (PROFIdrive の標準通信形式)に設定されています。
  - メッセージ 118 (2 つの別置きエンコーダ)
    または
  - メッセージ 138 (2 つの別置きエンコーダ、およびトルクフィードフォワード補償)

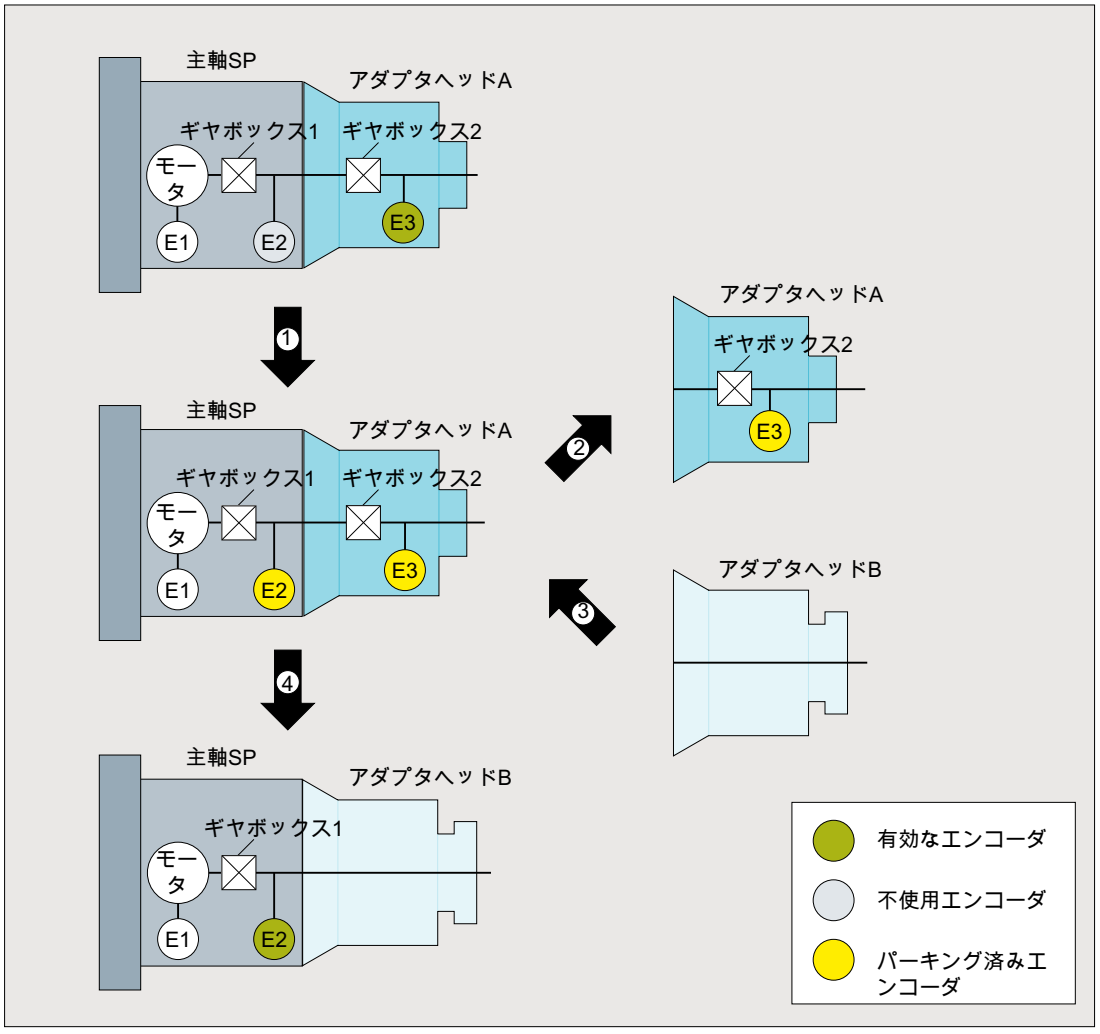
  参照先:
  エンコーダ割り当てについては、以下を参照してください。
  『試運転マニュアル、CNC 試運転:NC、PLC、ドライブ』
  「NC とドライブ間の通信」の「ドライブ:軸の割り当て」の章
- 以下の位置検出器が主軸「SP」で設定されています。
  - 位置検出器 1 として直接エンコーダ E2
  - 位置検出器 2 として直接エンコーダ E3
- エンコーダ E3 を備えた装着ヘッド「A」は、現在、主軸に取り付けられています。

- 位置検出器 2 が、有効な検出器です。
  DB31、... DBX1.6 = 1
  位置検出器 1 はパッシブです。
- 「パッシブな位置検出器のパーキング」機能は、以下のとおりです。
  - 位置検出器 1 で有効です。
    MD31046 $MA_ENC_PASSIVE_PARKING [0] = 0
  - 位置検出器 2 で有効です。
    MD31046 $MA_ENC_PASSIVE_PARKING [1] = 1

目的

ユーザーが、装着ヘッド「A」を装着ヘッド「B」に交換しようとしています。

実行



① 装着ヘッドを交換する前にユーザーは、「機械軸のパーキング (ページ 145)」機能を使用して、機械軸のすべての位置検出器を無効にしてください。

DB31、... DBX1.5 (位置検出器 1) = 0

DB31、... DBX1.6 (位置検出器 2) = 0

その後、コントローラが、位置検出器の状態信号をリセットします。

DB31、... DBX102.5 (位置検出器 1 有効化済み) = 0

DB31、... DBX102.6 (位置検出器 2 有効化済み) == 0

② ユーザーは、状態信号を待ち、主軸から装着ヘッド「A」を取り外します。これで、装着ヘッド「A」と連結部の間のエンコーダケーブルの電源接続が切断されます。エンコーダ E3 が存在していなくても、NC とドライブのいずれにも異常は発生しません。

③ 装着ヘッド「B」が主軸に取り付けられました。

④ ユーザーは、位置検出器 1 だけを有効にします。

DB31、... DBX1.5 (位置検出器 1) = 1

制御装置が状態信号を設定します。

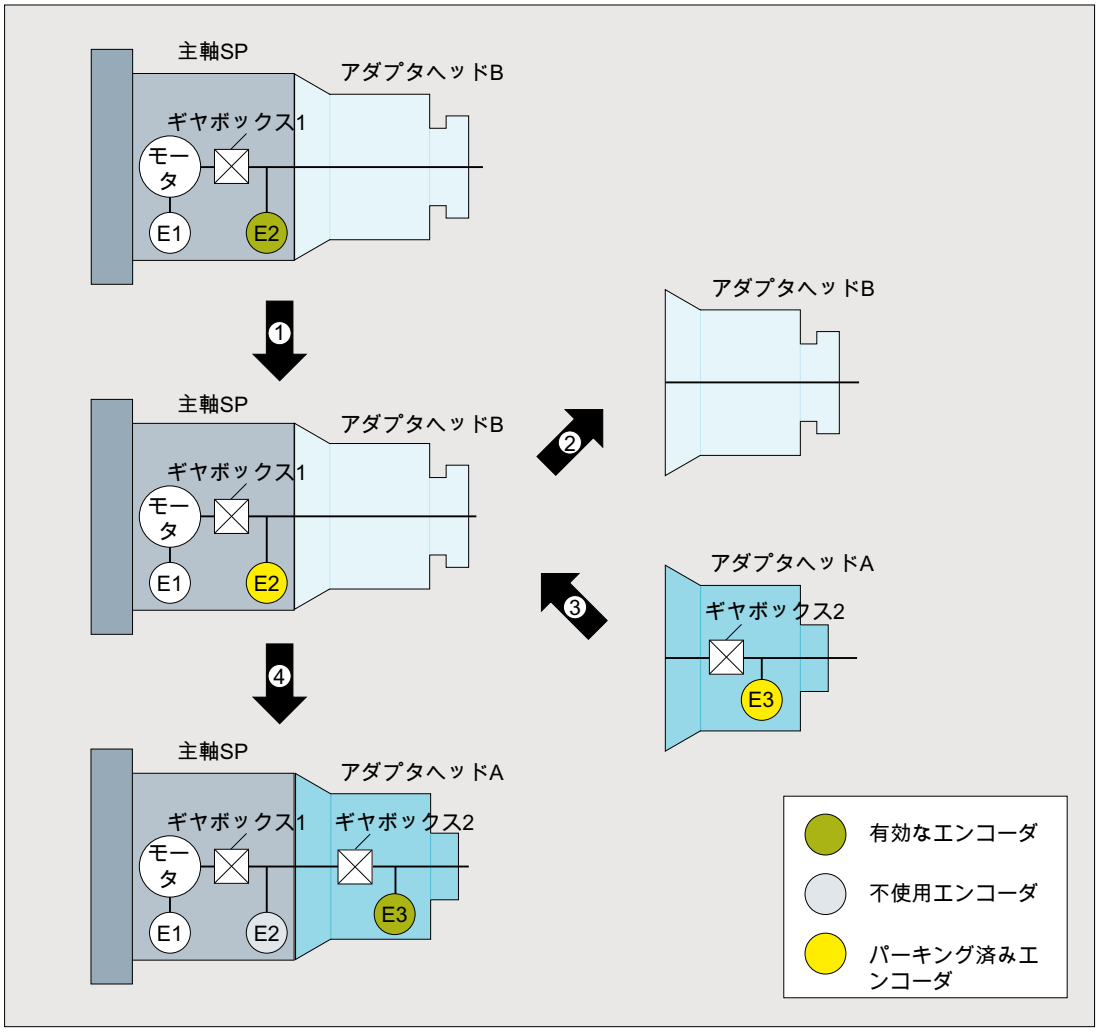
DB31、... DBX102.5 (位置検出器 1 有効化済み) = 1

「パッシブな位置検出器のパーキング」機能が位置検出器 2 で有効です。つまり、位置検出器 2 はパッシブにならず、「パーキング」状態のままです。

目的

ユーザーが、装着ヘッド「A」を再度取り付けようとしています。

実行



① ユーザーが「機械軸のパーキング」機能を使用して、位置検出器 1 を無効にします。

DB31、... DBX1.5 (位置検出器 1) = 0

その後、コントローラが、位置検出器の状態信号をリセットします。

DB31、... DBX102.5 (位置検出器 1 有効化済み) = 0

② ユーザーは、状態信号を待ち、主軸から装着ヘッド「B」を取り外します。

③ 装着ヘッド「A」が主軸に取り付けられました。

④ ユーザーは位置検出器 2 を有効にします。

DB31、... DBX1.6 (位置検出器 2) = 1

結果として、位置検出器 1 も同時に有効になります。これは、「パッシブな位置検出器のパーキング」機能が位置検出器 1 で有効でないためです。位置検出器 1 が、パッシブな位置検出器になります。

その後、コントローラが、位置検出器の状態信号を設定します。

DB31、... DBX102.5 (位置検出器 1 有効化済み) = 1

DB31、... DBX102.6 (位置検出器 2 有効化済み) = 1

3.9.5 例:範囲の一部にエンコーダがない場合の検出器の切り替え

以下の例では、加工ゾーン内のみで直接リニア位置検出器を使用できるのに対し、加工ゾーン外で使用できるのはモータ検出器のみです。

初期状態

- 直線軸「X」には 2 つのインクリメンタルエンコーダがあります。
  - モータエンコーダ E1
  - 直接リニアエンコーダ E2
- 直接リニアエンコーダ E2 は、加工ゾーン内でのみ使用できます。
- 以下の 2 つの通信形式のいずれかが、MD13060 $MN_DRIVE_TELEGRAM_TYPE (PROFIdrive の標準通信形式)に設定されています。
  - メッセージ 116 (モータエンコーダ+別置きエンコーダ)
    または
  - メッセージ 136 (モータエンコーダ+別置きエンコーダ、およびトルクフィードフォワード補償)
- 以下の位置検出器が機械軸に設定されています。
  - 位置検出器 1 としてモータエンコーダ E1
  - 位置検出器 2 として直接リニアエンコーダ E2
- 機械が電源投入されているときに、ユーザーが両方の位置検出器を有効にします。
  - DB31、... DBX1.5 (位置検出器 1) = 1
  - DB31、... DBX1.6 (位置検出器 2) = 1

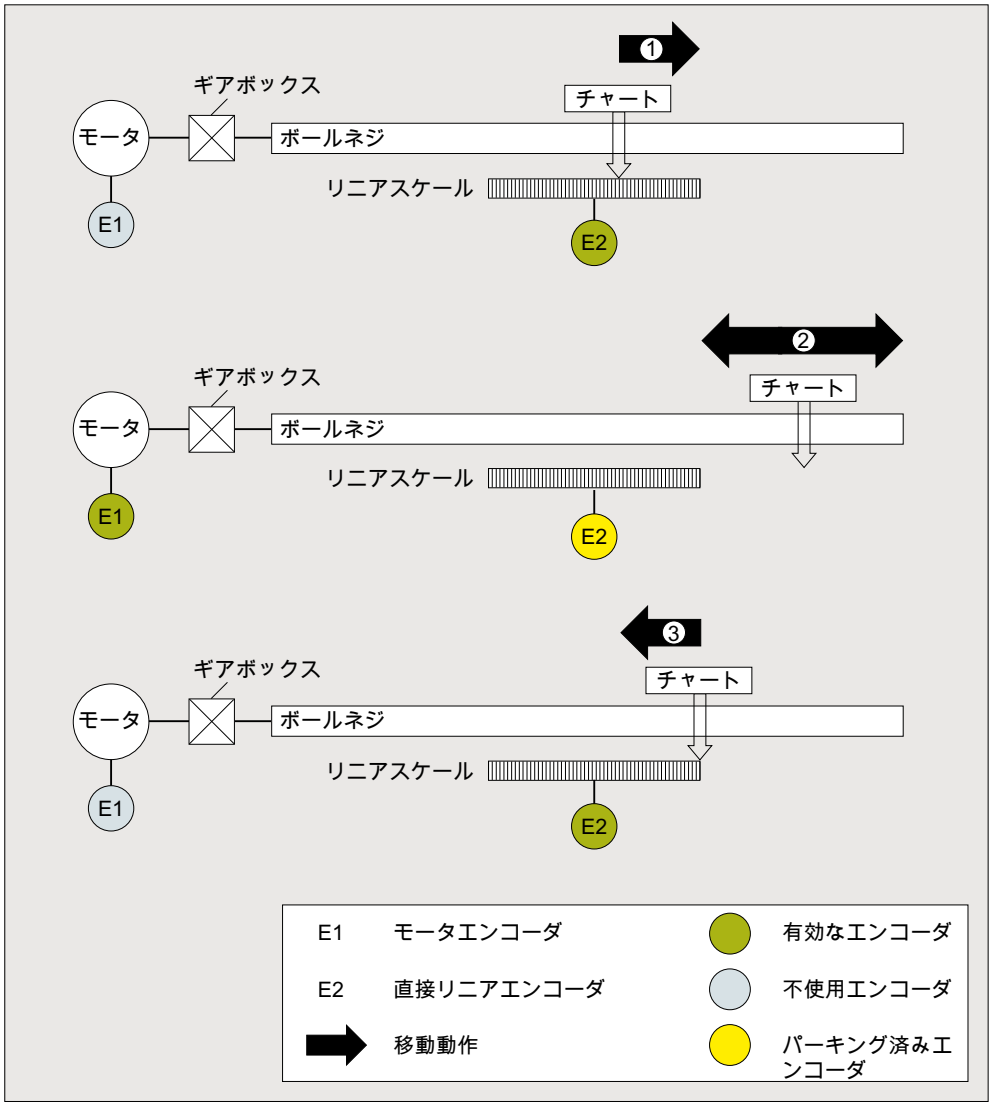
  両方の位置検出器を同時に有効にすると、制御装置は、有効な位置検出器として位置検出器 1 を選択します。

- 位置検出器 2 が加工用に選択されます。
  - DB31、... DBX1.5 (位置検出器 1) = 0
  - DB31、... DBX1.6 (位置検出器 2) = 1
- 「パッシブな位置検出器のパーキング」機能は、以下のとおりです。
  - 位置検出器 1 で有効です。
    MD31046 $MA_ENC_PASSIVE_PARKING [0] = 0
  - 位置検出器 2 で有効です。
    MD31046 $MA_ENC_PASSIVE_PARKING [1] = 1
- 位置と「原点確立済み」状態の転送が位置検出器 2 で有効です。
  MD34210 $MA_ENC_REFP_STATE[1] = 2

目的

リニア位置検出器 E2 の外部の範囲を通過中に、エンコーダ信号がないことにより、ドライブと制御装置で異常が発生しないようにします。

実行



① テーブルが、リニア位置検出器の最後に到達する前に、モータ検出器に切り替える必要があります。これは、ユーザーが両方の位置検出器を有効にして実行します。

DB31、... DBX1.5 (位置検出器 1) = 1

DB31、... DBX1.6 (位置検出器 2) = 1

「パッシブな位置検出器のパーキング」機能により、検出器の切り替え後にパッシブになるリニア位置検出器が制御装置によってパーキングされます。

制御装置が状態信号をリセットします。

DB31、... DBX102.6 (位置検出器 2 有効化済み) == 0

ユーザーは状態信号を待機し、その後、リニア位置検出器の外部の範囲での移動を続行します。

② モータ検出器による、リニア位置検出器の外部の範囲の通過。

③ テーブルが停止状態のときにリニア位置検出器の範囲に戻った場合は、ユーザーがモータ検出器からリニア位置検出器に切り替えます。

DB31、... DBX1.5 (位置検出器 1) = 0

DB31、... DBX1.6 (位置検出器 2) = 1

制御装置が状態信号を設定します。

DB31、... DBX102.6 (位置検出器 2 有効化済み) = 1

モータ位置検出器が、パッシブな位置検出器になります。

結果

両方の位置検出器が原点確立されます。リニア位置検出器の位置は、切り替えの時点では、モータ検出器の位置に対応します。位置の精度が十分でない場合は、リニア位置検出器を原点確立してください。

3.10 エンコーダデータセットの切替え

用途

異なる加工タスクを実行するために、まったく同一の電源ユニットで異なる取り付けヘッドを連続して使用できます。

モータまたはエンコーダを装備する各取り付けヘッドについて、1 つのモータデータセット(MDS)および 1 つのエンコーダデータセット(EDS)をプログラムする必要があります。これらのデータセットは、取り付けヘッドの変更時に常に、PLC プログラムで切替える必要があります。この件については、ドライブデータセット(DDS)の切替えによってのみ、MDS/EDS の切替えを間接的に実行できます。

機能

通知
機械の損傷 以下のドライブパラメータおよびマシンデータの設定が異なる場合、軸がプログラムされた速度でまたはプログラムされた位置に移動しないことがあります。 このマシンデータは、パーキング状態のエンコーダデータの切替えと同時に、矛盾なく変更する必要があります。

制御でのエンコーダデータの切替えは、以下の SINAMICS ドライブパラメータに制限されています。

- p0408 (ロータリエンコーダ パルス数)
- p0418 (高分解能エンコーダエミュレーション Gx_XIST1 (ビット単位))
- p0419 (高分解能絶対値 Gx_XIST2 (ビット単位))

これらのパラメータでは、同じエンコーダタイプでもエンコーダパルス数の異なるエンコーダデータセットの切替えが可能です。

制御での切替えは、以下のマシンデータで有効になります:

MD31700 $MA_ENC_EDS_ACTIVE (EDS 使用の有効化)

値	意味
0	エンコーダデータセット切替え EDS は使用されません。
1.	エンコーダデータセット切り替え EDS が使用されます。

ドライブパラメータ p0408、p0418、および p0419 のそれぞれに対して、マシンデータ項目を指定します。以下の有効なエンコーダデータセットに従ってパラメータを割り付ける必要があります。

- MD31710 $MA_ENC_RESOL_EDS (EDS 使用時の 1 回転当たりのエンコーダパルス)
- MD31720 $MA_ENC_PULSE_MULT_EDS (EDS 使用時のエンコーダの組み合わせ(高分解能))
- MD31730 $MA_ABS_INC_RATIO_EDS (絶対値エンコーダ:EDS 使用時のアブソリュート分解能とインクリメンタル分解能)

有効性

If MD31700 $MA_ENC_EDS_ACTIVE = 1、以下のマシンデータは無効になりました:

- MD30260 $MA_ABS_INC_RATIO (絶対値エンコーダ:アブソリュート分解能とインクリメンタル分解能の比率)
- MD31020 $MA_ENC_RESOL (1 回転あたりのエンコーダパルス)
- MD31025 $MA_ENC_PULSE_MULT (エンコーダの組み合わせ(高分解能))

DB3x.DBX21.0 - 4 のドライブデータの切替えにより、エンコーダデータセットも切替わります。切替えは、パーキング状態で実行されます(章「機械軸のパーキング (ページ 145)」および「パッシブな位置検出器のパーキング (ページ 147)」を参照)。

注記

MD31700 $MA_ENC_EDS_ACTIVE = 1 である場合、ドライブで設定された値と制御で設定された値の間で信頼性チェックは実行されません。

NC/PLC インタフェース信号:

- DB31、... DBX21.7 (パルスイネーブル)
- DB31、... DBX21.6 (積分停止、速度コントローラ)
- DB31、... DBX21.5 (モータ選択)
- DB31、... DBX21.0 - 4 (モータデータセットとドライブデータセットの切り替え要求)
  インターフェースは、以下を使用して柔軟に設定することができます。DB31、... DBX130.0 - 4

必要条件

SINUMERIK 必要条件

- ロータリーエンコーダ用のマシンデータのみが使用可能である。
- 絶対値エンコーダ MD30270 $MA_ENC_ABS_BUFFERING = 0 の移動範囲は、拡張できません。

SINAMICS の必要条件については、以下の取扱説明書を参照してください。

参照先

SINAMICS S120 機能マニュアルドライブ機能、
「ドライブシステムに関する一般事項」>「データセット」

- 章「DDS:ドライブデータセット」
- 章「EDS:エンコーダデータセット」

その他のマシンデータ

軸監視機能のマシンデータに加えて、次のマシンデータを設定またはチェックする必要があります。

すべての機械軸

- MD31030 $MA_LEADSCREW_PITCH (送りねじピッチ)
- MD31050 $MA_DRIVE_AX_RATIO_DENOM (負荷ギヤボックス分母)
- MD31060 $MA_DRIVE_AX_RATIO_NUMERA (負荷側ギヤボックス分子)
- MD31070 $MA_DRIVE_ENC_RATIO_DENOM (検出器ギヤボックス分母)
- MD31080 $MA_DRIVE_ENC_RATIO_NUMERA (検出器ギヤボックス分子)
- MD32810 $MA_EQUIV_SPEEDCTRL_TIME (フィードフォワード制御の速度制御ループ等価時定数)
- エンコーダ分解能:「G2: 速度、指令値/フィードバック回路、閉ループ制御 (ページ 405)」の章を参照してください。

アナログ速度指令値インタフェース付き機械軸

- MD32260 $MA_RATED_VELO (定格モータ速度)
- MD32250 $MA_RATED_OUTVAL (定格出力電圧)

3.11 データリスト

3.11.1 マシンデータ

3.11.1.1 NC 別マシンデータ

番号	識別子: $MN_	説明
10604	WALIM_GEOAX_CHANGE_MODE	ジオメトリ軸切り替え時のワーキングエリアリミット
10710	PROG_SD_RESET_SAVE_TAB	更新するセッティングデータ

3.11.1.2 チャネルマシンデータ

番号	識別子: $MC_	説明
20150	GCODE_RESET_VALUES	G グループの初期設定
21020	WORKAREA_WITH_TOOL_RADIUS	ワーキングエリアリミットのときの工具半径の取り扱い
24130	TRAFO_INCLUDES_TOOL_1	第 1 座標変換が有効な場合の工具の取り扱い
24230	TRAFO_INCLUDES_TOOL_2	第 2 座標変換が有効な場合の工具の取り扱い
24330	TRAFO_INCLUDES_TOOL_3	第 3 座標変換が有効な場合の工具の取り扱い
24426	TRAFO_INCLUDES_TOOL_4	第 4 座標変換が有効な場合の工具の取り扱い
24436	TRAFO_INCLUDES_TOOL_5	第 5 座標変換が有効な場合の工具の取り扱い
24446	TRAFO_INCLUDES_TOOL_6	第 6 座標変換が有効な場合の工具の取り扱い
24456	TRAFO_INCLUDES_TOOL_7	第 7 座標変換が有効な場合の工具の取り扱い
24466	TRAFO_INCLUDES_TOOL_8	第 8 座標変換が有効な場合の工具の取り扱い
24476	TRAFO_INCLUDES_TOOL_9	第 9 座標変換が有効な場合の工具の取り扱い
24486	TRAFO_INCLUDES_TOOL_10	第 10 座標変換が有効な場合の工具の取り扱い
25106	TRAFO_INCLUDES_TOOL_11	第 11 座標変換が有効な場合の工具の取り扱い
25116	TRAFO_INCLUDES_TOOL_12	第 12 座標変換が有効な場合の工具の取り扱い
25126	TRAFO_INCLUDES_TOOL_13	第 13 座標変換が有効な場合の工具の取り扱い
25136	TRAFO_INCLUDES_TOOL_14	第 14 座標変換が有効な場合の工具の取り扱い

番号	識別子: $MC_	説明
25146	TRAFO_INCLUDES_TOOL_15	第 15 座標変換が有効な場合の工具の取り扱い
25156	TRAFO_INCLUDES_TOOL_16	第 16 座標変換が有効な場合の工具の取り扱い
25166	TRAFO_INCLUDES_TOOL_17	第 17 座標変換が有効な場合の工具の取り扱い
25176	TRAFO_INCLUDES_TOOL_18	第 18 座標変換が有効な場合の工具の取り扱い
25186	TRAFO_INCLUDES_TOOL_19	第 19 座標変換が有効な場合の工具の取り扱い
25196	TRAFO_INCLUDES_TOOL_20	第 20 座標変換が有効な場合の工具の取り扱い
28600	MM_NUM_WORKAREA_CS_GROUPS	座標系用のワーキングエリアリミットの数

3.11.1.3 軸/主軸マシンデータ

番号	識別子: $MA_	説明
30200	NUM_ENCS	エンコーダの数
30240	ENC_TYPE	フィードバック値取得のエンコーダタイプ(位置フィードバック値)
30260	ABS_INC_RATIO	絶対値エンコーダ:アブソリュート値とインクリメンタル値の分解能の比率
30270	ENC_ABS_BUFFERING	絶対値エンコーダ:移動範囲の拡張
30310	ROT_IS_MODULO	回転軸/主軸のモジュロ変換
30800	WORK_AREA_CHECK_TYPE	作業領域リミットのチェックのタイプ
31020	ENC_RESOL	1 回転当たりのエンコーダパルス数
31025	ENC_PULSE_MULT	エンコーダの倍率(高分解能)
31046	ENC_PASSIVE_PARKING	パッシブな位置検出器のパーキング
31700	ENC_EDS_ACTIVE	EDS 使用の有効化
31710	ENC_RESOL_EDS	使用時の 1 回転当たりのエンコーダパルス数
31720	ENC_PULSE_MULT_EDS	EDS 使用時のエンコーダの倍率(高分解能)
31730	ABS_INC_RATIO_EDS	絶対値エンコーダ:EDS 使用時のアブソリュート分解能とインクリメンタル分解能の比率
32200	POSCTRL_GAIN [n]	K_V 係数
32250	RATED_OUTVAL	定格出力電圧
32260	RATED_VELO	定格モータ速度
32300	MAX_AX_ACCEL	最高軸加速度

番号	識別子: $MA_	説明
32800	EQUIV_CURRCTRL_TIME	フィードフォワード制御の電流制御等価時定数
32810	EQUIV_SPEEDCTRL_TIME	フィードフォワード制御の速度制御ループ等価時定数
32910	DYN_MATCH_TIME [n]	ダイナミック応答調整の時定数
34210	ENC_REFP_STATE	エンコーダ状態
35160	SPIND_EXTERN_VELO_LIMIT	PLCC による主軸速度制限
36000	STOP_LIMIT_COARSE	汎用イグザクトストップ
36010	STOP_LIMIT_FINE	精密イグザクトストップ
36020	POSITIONING_TIME	精密イグザクトストップ遅延時間
36030	STANDSTILL_POS_TOL	停止許容移動範囲
36040	STANDSTILL_DELAY_TIME	遅延時間停止状態監視
36050	CLAMP_POS_TOL	IS「クランプ有効」によるクランプ許容移動量
36052	STOP_ON_CLAMPING	クランプ軸用応用機能
36060	STANDSTILL_VELO_TOL	最高速度「軸/主軸停止」
36100	POS_LIMIT_MINUS	1 番目のソフトウェアリミットスイッチマイナス
36110	POS_LIMIT_PLUS	1 番目のソフトウェアリミットスイッチプラス
36120	POS_LIMIT_MINUS2	2 番目のソフトウェアリミットスイッチマイナス
36130	POS_LIMIT_PLUS2	2 番目のソフトウェアリミットスイッチプラス
36200	AX_VELO_LIMIT	速度監視の検出値
36210	CTRLOUT_LIMIT	最高速度指令値
36220	CTRLOUT_LIMIT_TIME	速度指令監視の遅延時間
36300	ENC_FREQ_LIMIT	エンコーダ限界周波数
36302	ENC_FREQ_LIMIT_LOW	エンコーダ再原点同期のエンコーダ限界周波数
36310	ENC_ZERO_MONITORING	ゼロマーク監視
36312	ENC_ABS_ZEROMON_WARNING	原点マーク監視警告レベル
36400	CONTOUR_TOL	軌跡誤差監視許容範囲
36500	ENC_CHANGE_TOL	位置フィードバック値切り替えの最大許容範囲
36510	ENC_DIFF_TOL	検出器間許容誤差
36600	BRAKE_MODE_CHOICE	ハードウェアリミットスイッチの減速動作
36610	AX_EMERGENCY_STOP_TIME	障害に対する減速カーブの最大時間
36620	SERVO_DISABLE_DELAY_TIME	コントローライネーブル遮断遅延

3.11.2 セッティングデータ

3.11.2.1 軸/主軸セッティングデータ

番号	識別子: $SA_	説明
43400	WORKAREA_PLUS_ENABLE	ワーキングエリアリミットがプラス方向に有効
43410	WORKAREA_MINUS_ENABLE	ワーキングエリアリミットがマイナス方向に有効
43420	WORKAREA_LIMIT_PLUS	ワーキングエリアリミットプラス
43430	WORKAREA_LIMIT_MINUS	ワーキングエリアリミットマイナス

3.11.3 信号

3.11.3.1 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
フォローアップモード	DB31、... .DBX1.4	DB380x.DBX1.4
位置検出器 1/2	DB31、... .DBX1.5/6	DB380x.DBX1.5/6
コントローライネーブル	DB31、... .DBX2.1	DB380x.DBX2.1
クランプ中	DB31、... .DBX2.3	DB380x.DBX2.3
速度/主軸速度制限	DB31、... .DBX3.6	DB380x.DBX3.6
送り停止/主軸停止	DB31、... .DBX4.3	DB380x.DBX4.3
ハードウェアリミットスイッチマイナス/ハードウェアリミットスイッチプラス	DB31、... .DBX12.0/1	DB380x.DBX1000.0/1
第 2 ソフトウェアリミットスイッチマイナス/第 2 ソフトウェアリミットスイッチプラス	DB31、... .DBX12.2/3	DB380x.DBX1000.2/3
モータ/ドライブデータセット:	DB31、... DBX21.0 - 4	DB380x.DBX4001.0 - 4
選択されているモータ	DB31、... .DBX21.5	DB380x.DBX4001.5
速度コントローラ積分停止	DB31、... .DBX21.6	DB380x.DBX4001.6
パルスイネーブル	DB31、... .DBX21.7	DB380x.DBX4001.7

3.11.3.2 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
エンコーダ限界周波数超過 1 / 2	DB31、... .DBX60.2/3	DB390x.DBX0.2
原点確立済み/原点同期済み 1/2	DB31、... .DBX60.4/5	DB390x.DBX0.4/5
移動指令マイナス/プラス	DB31、... .DBX64.6/7	DB390x.DBX4.6/7
位置検出器 1/2 有効化済み	DB31、...DBX102.5/6	DB390x.DBX5006.5/6
モータ/ドライブデータセット:	DB31、... DBX130.0 - 4	DB390x.DBX4008.0 - 4

A5:プロテクションゾーン

4

4.1 機能

プロテクションゾーンは、機械内の 2 次元または 3 次元の静止または稼動部分で、機械の要素を衝突から保護します。またこれは、ユーザーが定義する必要があります。

次の要素を保護できます。

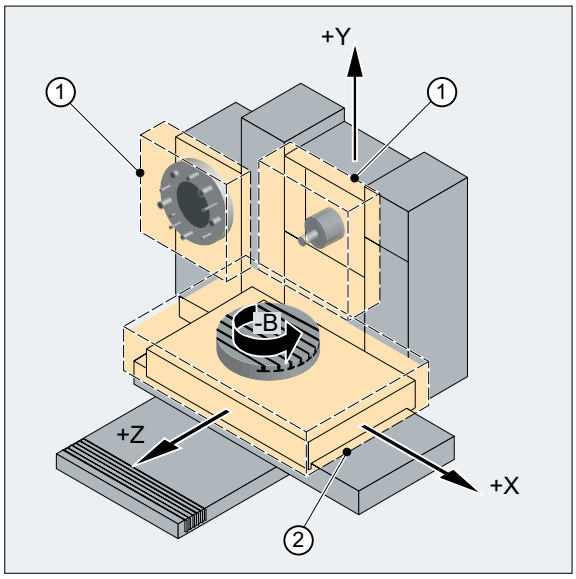
- 固定機械要素(たとえば、工具マガジン、旋回可能なプローブなど)。
- 工具に属する機械要素の移動(工具、工具ホルダなど)
- ワークに属する機械要素(ワーク部分、クランプ台、クランプシュー、主軸チャック、心押し台など)の移動

機械要素を確実に保護するために、保護する要素が完全に覆われるように、プロテクションゾーンを定義する必要があります。

プロテクションゾーン監視はチャネル単位で行われます。つまり、チャネルの有効なプロテクションゾーンすべてによって、衝突の有無が相互に監視されます。

AUTOMATIC モードまたは MDI モードでのパートプログラムの自動実行中、NC が、プログラム指令軌跡に沿った移動時にプロテクションゾーン間の衝突が発生する可能性があるかどうかを、すべてのパートプログラムブロックの開始時にチェックします。

有効なプロテクションゾーンを手動で無効にした後で、このプロテクションゾーン内を移動することができます。プロテクションゾーンから出ると、プロテクションゾーンは自動的に再び有効になります。



① **工具**関連プロテクションゾーン

② **ワーク**基準プロテクションゾーン

図 4-1　フライス盤におけるプロテクションゾーンの例

プロテクションゾーンの定義

プロテクションゾーンは 2 次元または 3 次元で、輪郭要素として最大 10 個のコーナー点と円弧を含む連続線を使用して定義します。この定義は、パートプログラムの命令またはシステム変数を使用して作成できます(「プロテクションゾーンの定義(CPROTDEF、NPROTDEF) (ページ 184)」を参照してください)。すべての輪郭要素は、G17、G18、または G19 で選択できる平面に含まれます。

3 次元

3 次元でのプロテクションゾーンでの拡張は、- ∞から+ ∞の間で制限できます。

- -∞から+∞まで
- -∞から上限まで
- 下限から+∞まで
- 下限から上限まで

座標系

プロテクションゾーンの定義は、チャネル、したがって基本座標系(BCS)のジオメトリ軸を基準としておこなわれます。

参照先

- 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。

注記

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。

旋回

プロテクションゾーンの向きは、輪郭が記述される平面の定義(横軸/縦軸)と輪郭に直交する軸(垂直軸)によって決定されます。

プロテクションゾーンの向きは、工具基準プロテクションゾーンでもワーク基準プロテクションゾーンでも同じにしてください。

機械/チャネル別保護領域

- 機械別保護領域
  機械別プロテクションゾーンのデータは、制御装置で 1 回定義されます。これらの保護領域は、すべてのチャネルにおいて有効にできます。
- チャネル別保護領域
  チャネル別プロテクションゾーンのデータは、チャネル内で定義されます。これらの保護領域は、このチャネルによってのみ有効にできます。

システム変数

パートプログラムで命令を使用してプロテクションゾーンを定義した場合、プロテクションゾーンデータはシステム変数に格納されます。システム変数は直接書き込むこともできるため、プロテクションゾーンの定義をシステム変数で直接おこなうこともできます。プロテクションゾーンの定義について、パートプログラムで命令を使用してプロテクションゾーンの輪郭の定義に同じ補足条件を適用します(「プロテクションゾーンの定義(CPROTDEF、NPROTDEF) (ページ 184)」を参照)。

プロテクションゾーンの定義では、次のシステム変数が対象となります。

システム変数	タイプ	意味	
$SN_PA_ACTIV_IMMED[<n>] $SC_PA_ACTIV_IMMED[<n>]	BOOL	起動タイプ プロテクションゾーンは、コントローラシステムの電源投入および軸の原点確立の直後に有効/無効になります。	
		FALSE	即時有効になりません。
		TRUE	即時有効になります。
$SN_PA_T_W[<n>] $SC_PA_T_W[<n>]	INT	プロテクションゾーンのタイプ	
		0	ワーク基準プロテクションゾーン
		1	予備
		2	予備
		3	工具関連プロテクションゾーン
$SN_PA_ORI[<n>] $SC_PA_ORI[<n>]	INT	プロテクションゾーンの向き、つまり次の平面における多角形の定義:	
		0	1 番目と 2 番目のジオメトリ軸
		1	3 番目と 1 番目のジオメトリ軸
		2	2 番目と 3 番目のジオメトリ軸
$SN_PA_LIM_3DIM[<n>] $SC_PA_LIM_3DIM[<n>]	INT	3 次元での制限のタイプ	
		0	制限なし
		1	マイナス方向の制限
		2	マイナス方向の制限
		3	正負方向の制限
$SN_PA_PLUS_LIM[<n>] $SC_PA_PLUS_LIM[<n>]	REAL	3 次元での正方向の制限値	
$SN_PA_MINUS_LIM[<n>] $SC_PA_MINUS_LIM[<n>]	REAL	3 次元での負方向の制限値	
$SN_PA_CONT_NUM[<n>] $SC_PA_CONT_NUM[<n>]	INT	有効な輪郭要素の数	
$SN_PA_CONT_TYP[<n>, <i>] $SC_PA_CONT_TYP[<n>, <i>]	INT	輪郭タイプ[<i>]、<i>番目の輪郭要素の輪郭タイプ(G1、G2、G3)	

システム変数	タイプ	意味
$SN_PA_CONT_ABS[<n>, <i>] $SC_PA_CONT_ABS[<n>, <i>]	REAL	輪郭[<i>]の終点、横座標値
$SN_PA_CONT_ORD[<n>, <i>] $SC_PA_CONT_ORD[<n>, <i>]	REAL	輪郭[<i>]の終点、縦座標値
$SN_PA_CENT_ABS[<n>, <i>] $SC_PA_CENT_ABS[<n>, <i>]	REAL	円弧輪郭[<i>]の中心点、アブソリュート横座標値
$SN_PA_CENT_ORD[<n>, <i>] $SC_PA_CENT_ORD[<n>, <i>]	REAL	円弧輪郭[i]の中心点、アブソリュート縦座標値
$SN_... は、NC 別と機械別プロテクションゾーンのシステム変数です。 $SC_... は、チャネル別プロテクションゾーンのシステム変数です。 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 = 第 1 プロテクションゾーン インデックス"<i>"は輪郭要素の数に対応します:0 = 第 1 輪郭要素 輪郭要素は昇順で定義してください。		

注記

プロテクションゾーン定義のシステム変数は REORG によって復元されません。

プロテクションゾーン定義のデータ

データ記憶

プロテクションゾーン定義は以下のファイルに保存されます。

ファイル	ブロック
_N_NCK_PRO	機械別プロテクションゾーンのデータブロック
_N_CHAN1_PRO	チャネル 1 のチャネル別プロテクションゾーンのデータブロック
_N_CHAN2_PRO	チャネル 2 のチャネル別プロテクションゾーンのデータブロック

データバックアップ

プロテクションゾーン定義は以下のファイルに保存されます。

ファイル	ブロック
_N_INITIAL_INI	プロテクションゾーンのすべてのデータブロック
_N_COMPLETE_PRO	プロテクションゾーンのすべてのデータブロック
_N_CHAN_PRO	チャネル別プロテクションゾーンのすべてのデータブロック

プロテクションゾーンの起動、起動待ち、および解除

起動状態

プロテクションゾーンの起動状態には以下の値があります。

- 有効化
- 起動待ち
- オプショナルストップ起動待ち
- 無効化

機械別のプロテクションゾーンであっても、起動状態は常にチャネル別です。

パートプログラムによる起動、起動待ち、および解除

プロテクションゾーンの起動状態は、命令を使用してパートプログラムでいつでも変更できます(「プロテクションゾーンの起動/解除(CPROT、NPROT) (ページ 188)」を参照してください)。

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。

後で PLC ユーザープログラム(以下を参照)から起動するプロテクションゾーンは、パートプログラムで起動待ちにしてください。

起動待ちプロテクションゾーンは、以下の NC/PLC インタフェース信号で表示されます。

- DB21、... DBX272.0 - 273.1 (機械基準プロテクションゾーン 1 ～ 10 が起動待ち) == 1
- DB21、... DBX274.0 - 275.1 (チャネル別プロテクションゾーン～ 10 が起動待ち)1 ==== 1

オプショナルストップ起動待ち

通知
プロテクションゾーン違反の可能性 オプショナルストップによる起動待ちプロテクションゾーンを適切なタイミングで起動しないと、起動ポイント後の減速距離が考慮されないため、プロテクションゾーンが適切なタイミングになる前に NC を停止できなくなる場合があります。

オプショナルストップによるプロテクションゾーンの起動待ちでは、移動動作がプロテクションゾーンに入る場合、移動動作はこの前に停止**されません**。停止は、プロテクションゾーンが動作中の場合にのみ行われます。この特性により、プロテクションゾーンが一時的にのみ必要な場合に、ユーザーが制御する加工を中断せずにおこなうことができます。

NC/PLC インタフェース信号を使用して起動

パートプログラムにより**起動待ち**になったプロテクションゾーンのみ、次の NC/PLC インタフェース信号により PLC ユーザープログラムで起動することができます。

- DB21、... DBX8.0 - 9.1 = 1 (機械基準プロテクションゾーン 1 ～ 10 の起動) = 1
- DB21、... DBX10.0 - 11.1 (チャネル別プロテクションゾーン 1 ～ 10 の起動) = 1

起動待ちのプロテクションゾーンの起動は、ジオメトリ軸の移動動作の前に行ってください。移動動作中に起動を行うと、これらのプロテクションゾーンは現在の移動動作には考慮されません。動作:

- アラーム「10704 プロテクションゾーン監視が保証されません」
- DB21、... DBX39.0 (プロテクションゾーン監視が保証されません) = 1

注記

起動待ちのプロテクションゾーンの起動は、ジオメトリ軸の移動動作の前に行ってください。

NC/PLC インタフェース信号を使用して解除

パートプログラムによって**起動待ち**にし、NC/PLC インタフェース信号によって起動したプロテクションゾーンは、NC/PLC インタフェース信号によって再度解除することができます。

- DB21、... DBX8.0 ～ DBX9.1 (機械基準プロテクションゾーン 1 ～ 10 の起動) = 0
- DB21、... DBX10.0 ～ DBX11.1 (チャネル別プロテクションゾーン 1 ～ 10 の起動) = 0

パートプログラムで直接**起動した**プロテクションゾーンは、PLC ユーザープログラムで解除することは**できません**。

座標変換の変更/ジオメトリ軸の入れ替えの自動無効化

初期設定で、座標変換の変更時、またはジオメトリ軸の入れ替えでは、有効なプロテクションゾーンが自動的に解除されます。それに対して、有効なプロテクションゾーンを有効なままにする必要がある場合は、ビット指定のマシンデータ MD10618 $MN_PROTAREA_GEOAX_CHANGE_MODE を相応に設定する必要があります(「マシンデータ (ページ 183)」を参照してください)。

特別なシステム状態の起動状態

計算ありのブロック検索

計算ありのブロック検索では、最後にプログラム指令されたプロテクションゾーンの起動状態が常に考慮に入れられます。

プログラムテスト

AUTOMATIC と MDI モードでは、起動されたプロテクションゾーンと起動待ちのプロテクションゾーンも「プログラムテスト」状態で監視されます。

NC RESET とエンドオブプログラム

プロテクションゾーンの起動状態は、NC リセットおよびプログラム終了後も保持されます。

プロテクションゾーン違反の表示

起動されたプロテクションゾーンの違反、または起動待ちのプロテクションゾーンが起動された場合に起こり得る違反が、以下の NC/PLC インタフェース信号によって表示されます。

- DB21、... DBX276.0 ～ 277.1 (機械基準プロテクションゾーン 1 ～ 10 の違反) == 1
- DB21、... DBX278.0 ～ 279.1 (チャネル別プロテクションゾーン 1 ～ 10 の違反) == 1

プロテクションゾーン違反のチェック

ファンクション CALCPOSI は、ジオメトリ軸がプロテクションゾーンを違反せずに指定の軌跡を移動できるかどうかを確認する場合に使用できます(「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) (ページ 193)」を参照)。

4.2 セットアップ

4.2.1 マシンデータ

必要メモリ

プロテクションゾーンに必要なメモリは、以下のマシンデータを使用してパラメータ設定されます。

- 永続性メモリ
  - MD18190 $MN_MM_NUM_PROTECT_AREA_NCK (使用可能な機械別プロテクションゾーンの数)
  - MD28200 $MC_MM_NUM_PROTECT_AREA_CHAN (使用可能なチャネル別プロテクションゾーンの数)
- ダイナミックメモリ
  - MD28210 $MC_MM_NUM_PROTECT_AREA_ACTIVE (チャネルで同時に有効にできるプロテクションゾーンの最大数)
  - MD28212 $MC_MM_NUM_PROTECT_AREA_CONTUR (1 プロテクションゾーンにつき定義できる輪郭要素の最大数)

座標変換の変更/ジオメトリ軸の入れ替えの動作

次のマシンデータでは、座標変換の変更時またはジオメトリ軸の入れ替えで、有効なプロテクションゾーンを維持するか、または解除するかを定義できます。

MD10618 $MN_PROTAREA_GEOAX_CHANGE_MODE

ビット	値	意味
0	0	座標変換の変更時に、有効なプロテクションゾーンが解除されます。
	1	座標変換の変更時に、有効なプロテクションゾーンは有効のままです。
1	0	ジオメトリ軸の変更時に、有効なプロテクションゾーンが解除されます。
	1	ジオメトリ軸の変更時に、有効なプロテクションゾーンは有効なままです。

4.3 プログラミング

4.3.1 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 188)」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

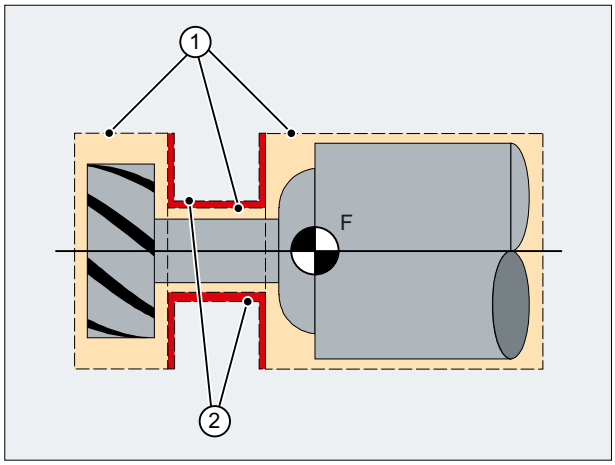
- 工具基準プロテクションゾーン
  工具基準プロテクションゾーンの座標値は、**工具ホルダレファレンス点 F** を基準とする絶対値として指定する必要があります。
- ワーク基準プロテクションゾーン
  ワーク基準プロテクションゾーンの座標値は、**基本座標系(BCS)**のゼロ点を基準とする絶対値として指定する必要があります。

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

4.3.2 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

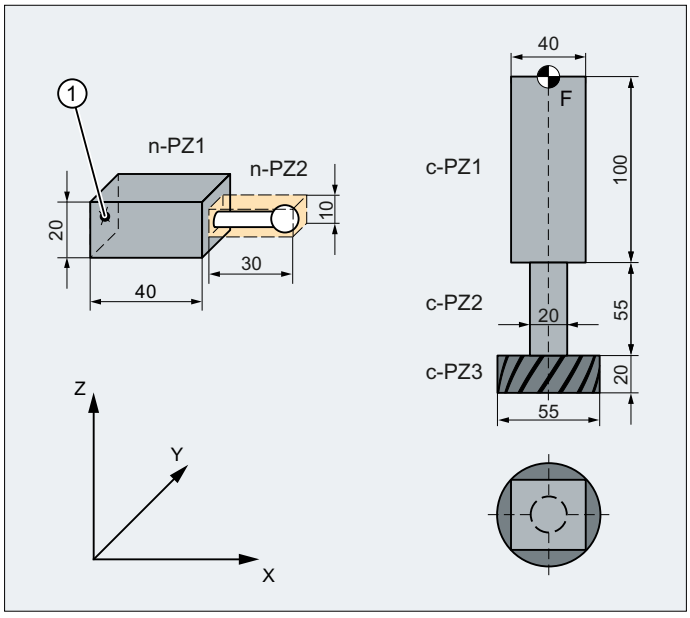
フライスカッターと計測プローブの衝突の発生の可能性を、フライス盤で監視します。機能が有効なときは、オフセットにより計測プローブの位置が定義されます。

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。

4.3.3 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	

<table>
<tr><td rowspan="8"><距離>:</td><td colspan="3">ベクトルの参照。</td></tr>
<tr><td colspan="3">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</td></tr>
<tr><td colspan="3">出力(<状態>で 100000 桁を設定している場合のみ)</td></tr>
<tr><td colspan="3"><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。</td></tr>
<tr><td colspan="3">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。</td></tr>
<tr><td colspan="3">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。</td></tr>
<tr><td>パラメータタイプ:</td><td colspan="2">入力/出力</td></tr>
<tr><td>データタイプ:</td><td colspan="2">VAR REAL [3]</td></tr>
<tr><td>値の範囲:</td><td colspan="2">-最大 REAL 値 ≤ x[<n>] ≤ +最大 REAL 値</td></tr>
</table>

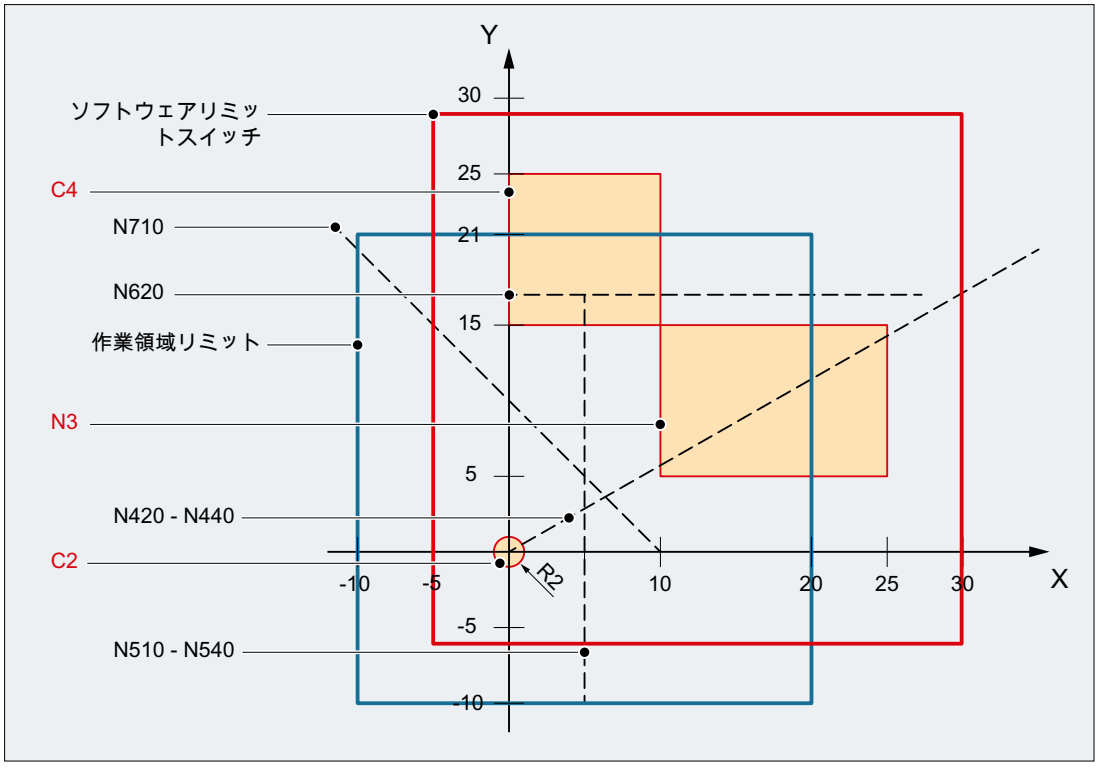
<リミット>:	長さ 5 の配列の参照 ● **<**リミット**>** [0 - 2]:リミットまでのジオメトリ軸の最小スペース – **<**リミット**>** [0]:1 番目のジオメトリ軸 – **<**リミット**>** [1]:2 番目のジオメトリ軸 – **<**リミット**>** [2]:3 番目のジオメトリ軸 最小スペースは、次のもので遵守されます。 – 作業領域リミット:制限なし – ソフトウェアリミットスイッチ:座標変換が有効でない場合や、5 軸座標変換など、直線機械軸へのジオメトリ軸の明確な割り当てが可能な座標変換が有効な場合。 ● **<**リミット**>** [3]: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
```

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。

4.4 特別な事例

4.4.1 プロテクションゾーンの一時移動許可

移動動作の開始時または移動動作中にプロテクションゾーン違反が発生した場合、特定の条件下で、一時的な移動などのためにプロテクションゾーンを移動を許可することができます。AUTOMATIC と MDI モード、ならびに JOG モードでは、プロテクションゾーンの一時移動許可をオペレータ操作によっておこないます。

以下も参照ください:

- AUTOMATIC と MDI モードでの動作 (ページ 207)
- JOG モードでの動作 (ページ 208)

注記

プロテクションゾーンの一時移動許可は、**ワーク**関連のプロテクションゾーン**のみ**で可能です。**工具**基準プロテクションゾーンは、パートプログラムで解除するか、「起動待ち」状態で NC/PLC インタフェースを介して解除してください。

次のイベントの後、プロテクションゾーンの一時移動許可が終了します。

- コントロールシステムの電源投入時
- AUTOMATIC または MDI モードブロック終点がプロテクションゾーン外
- JOG モード:移動動作終点がプロテクションゾーン外
- プロテクションゾーンが起動される

4.4.2 AUTOMATIC と MDI モードでの動作

AUTOMATIC と MDI モードでは、移動動作を起動中のプロテクションゾーンでおこなったい、プロテクションゾーンを通過させることはできません。

- 外部からの起動中のプロテクションゾーンへの移動動作は、プロテクションゾーンの外側にあるる最後のブロックの終点で停止します。
- 起動中のプロテクションゾーン内で始まる移動動作は、開始されません。

プロテクションゾーンの一時移動許可

移動動作がプロテクションゾーン違反のために AUTOMATIC または MDI モードで停止すると、アラームでオペレータに通知されます。オペレータが移動動作が続行可能と判定すると、プロテクションゾーンの通過が許可されます。

移動許可は一時的なものに過ぎず、NC スタートの起動により行われます。

DB21、... DBX7.1 (NC スタート) = 1

違反プロテクションゾーン毎にアラームが表示されます。NC スタート信号は、移動許可するプロテクションゾーンごとにオペレータが起動してください。

移動動作の停止という結果になったすべてのプロテクションゾーンの移動が許可されている場合に、移動動作が継続されます。

一時移動許可なしの移動動作の継続

移動動作が、プロテクションゾーン違反のためにアラームで停止しました。該当するプロテクションゾーンが NC/PLC インタフェースを介して「起動待ち」状態に設定されている場合、プロテクションゾーンの一時移動許可なしで、移動動作を NC スタートで続行することができます。

プロテクションゾーンの移動許可に対する保護の強化

プロテクションゾーンの移動許可の保護を、NC スタートのみによる保護よりも強化する場合、これは工作機械メーカまたはユーザーが PLC ユーザープログラムで行ってください。

4.4.3 JOG モードでの動作

複数のジオメトリ軸の同時移動

JOG モードでは、移動動作を複数のジオメトリ軸で同時におこなうことができます。関係するすべてのジオメトリ軸の移動範囲は、移動動作の開始時に、移動範囲制限(ソフトウェアリミットスイッチ、作業領域リミットなど)、および起動中のプロテクションゾーンに関して軸別に制限されます。ただし、すべての起動中のプロテクションゾーンの安全な監視は保証されません。以下がフィードバックとしてユーザーに提供されます。

- アラーム「10704 プロテクションゾーン監視が保証されません」
- DB31、... DBX39.0 (プロテクションゾーン監視が保証されません) = 1

移動動作の終了後、アラームは自動的に消去されます。

現在の位置が起動中または起動待ちのプロテクションゾーン内にある場合、以下の動作がトリガされます。

- アラームメッセージ 10702「手動モードでの NC プロテクションゾーンの違反」または 10703「手動モードでのチャネル別プロテクションゾーンの違反」が出力され、プロテクションゾーンおよび移動軸の違反が通知されます。
- 以降の移動動作が無効になります。
- 関連するプロテクションゾーンで、次の NC/PLC インタフェース信号が設定されます。
  DB21、... DBX276.0 ～ 277.1 (機械別プロテクションゾーン 1 ～ 10 の違反) == 1
  または
  DB21、... DBX278.0 ～ 279.1 (チャネル別プロテクションゾーン 1 ～ 10 の違反) == 1

続行するには、「プロテクションゾーンの一時移動許可」を参照してください。

例:

3 つのプロテクションゾーンが起動中で、2 つのジオメトリ軸が同時に移動:

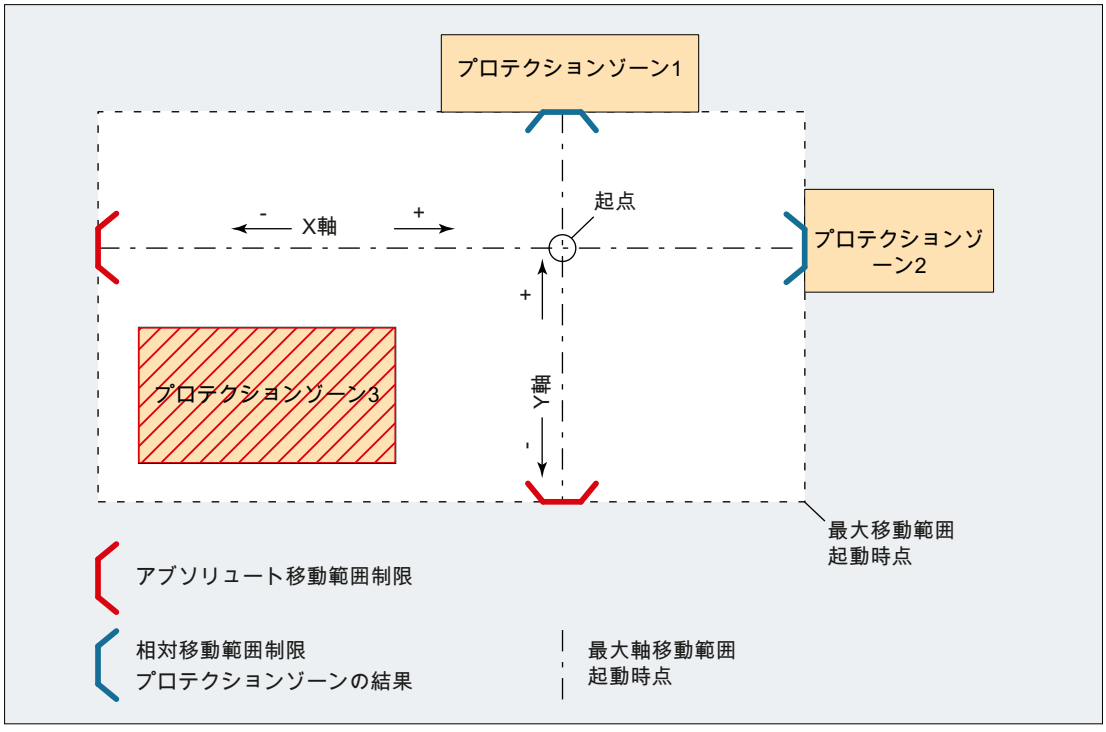


図 4-2　開始時のジオメトリ軸の移動範囲

軸 X と軸 Y の移動動作の開始時に、開始時からの軸別の移動範囲制限が決定されます。

- X 軸
  - 正の移動方向:プロテクションゾーン 2
  - 負の移動方向:アブソリュート移動範囲制限(たとえばソフトウェアリミットスイッチなど)
- Y 軸
  - 正の移動方向:プロテクションゾーン 1
  - 負の移動方向:アブソリュート移動範囲制限(たとえば作業領域リミットなど)

結果として得られる開始時の最大移動範囲には、プロテクションゾーン 3 が考慮されていません。その結果、プロテクションゾーン 3 の違反が発生する可能性があります。

注記

起動中と起動待ちのプロテクションゾーンは、手動運転モード JOG、INC、および DRF でも監視されます。

軸の移動動作の制限

プロテクションゾーンに到達したために軸の移動動作が制限された場合、アラーム 10706「手動モードでの NC プロテクションゾーンの到達」または 10707「手動モードでのチャネル別プロテクションゾーンの到達」が出力され、到達したプロテクションゾーンと移動軸が通知されます。JOG で 1 つの軸が移動しているとき、どのプロテクションゾーンにも違反しないことが保証されます(この応答は、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へアプローチする場合と同様です)。

アラームは次の場合にリセットされます。

- プロテクションゾーンに到達しない軸の移動時
- プロテクションゾーンが移動を許可された場合
- 制御装置の電源投入時

プロテクションゾーンのリミット位置にある軸がプロテクションゾーンに近づき始めると、アラーム 10706 または 10707 が出力され、動作は起動しません

プロテクションゾーンの一時移動許可

起動したプロテクションゾーン内またはそのリミット位置で移動動作が起動すると、アラーム 10702「手動モードでの NC プロテクションゾーンの違反」または 10703「手動モードでのチャネル別プロテクションゾーンの違反」が出力され、違反したプロテクションゾーンと移動軸が通知されます。動作は起動しません。移動動作は、該当するプロテクションゾーンが一時移動許可されている場合におこなうことができます。このためには、以下の操作をおこなってください。

- NC/PLC インタフェース信号で立ち上がりエッジを作成します。
  DB21、... DBX1.1 (プロテクションゾーンの移動許可)
- 同じ移動動作をもう一度開始します。

注記

一時移動許可されたプロテクションゾーンを移動しているときも、NC/PLC インタフェース信号「プロテクションゾーン違反」が設定されています。

DB21、... DBX276.0 ～ 277.1 (機械別プロテクションゾーン 1 ～ 10 の違反) == 1
または
DB21、... DBX278.0 ～ 279.1 (チャネル別プロテクションゾーン 1 ～ 10 の違反) == 1

移動を許可されたプロテクションゾーンに進入しない移動が開始されると、許可信号がリセットされます。

さらにプロテクションゾーンで移動動作がおこなわれると、追加アラーム 10702 または 10703 がプロテクションゾーン毎に表示されます。アラームで表示されたプロテクションゾーンは、NC/PLC インタフェース信号(DB21、... DBX1.1)でさらに立ち上がりエッジを作成すると移動許可されます。

運転モードの変更時の動作

プロテクションゾーンが JOG モードで一時的に移動許可され、AUTOMATIC または MDI モードに変更された後もそのまま保持されます。AUTOMATIC または MDI モードで設定された一時許可信号は、JOG モードに変更された後もそのまま保持されます。

移動許可のリセット

移動許可は、一時移動許可されたプロテクションゾーンが完全に終了しているジオメトリ軸の次の停止時に、NC/PLC インタフェースで内部的にリセットされます。

DB21、... DBX276.0 ～ 277.1 (機械別プロテクションゾーン 1 ～-10 の違反) == 0

または

DB21、... DBX278.0 ～ 279.1 (チャネル別プロテクションゾーン 1 ～-10 の違反) == 0

4.5 境界条件

プロテクションゾーン監視での制限事項

次の場合、プロテクションゾーンの監視は可能では**ありません**。

- 旋回軸の移動動作
- 正面座標変換(TRANSMIT)または円筒面座標変換(TRACYL)に対する**機械**別の**固定**プロテクションゾーン。
  例外:主軸の回転軸を中心として対称のプロテクションゾーン。この場合、DRF オフセットが有効であってはなりません。
- **工具**基準のプロテクションゾーンの相互監視

位置決め軸

位置決め軸の場合、プログラム指令されたブロック終点のみが監視されます。アラーム 10704「プロテクションゾーン監視が保証されません」が位置決め軸の移動時に出力されます。

軸入れ替え

軸入れ替え後のプロテクションゾーンの場合は、開始位置は中止チャネルでアプローチした最後の位置になります。引き継いだチャネルでの移動動作は考慮されません。このため、軸入れ替えをプロテクションゾーン違反のあった位置から行われないようにしてください。

ある軸が軸の入れ替えのためチャネル内で有効でない場合、そのチャネル内で最後にアプローチしようとした軸の位置を現在の位置と見なします。この軸がチャネル内でまだ移動したことがない場合は、0.0 がその位置と見なされます。

重畳移動の場合の動作

メインランに含まれる重畳移動は、起動中のプロテクションゾーンに関するブロック解析で考慮することはできません。

この結果、次の応答が得られます。

- アラーム「10704 プロテクションゾーン監視が保証されません」
- DB31、... DBX39.0 = 1(プロテクションゾーン監視が保証されません)

4.6 例

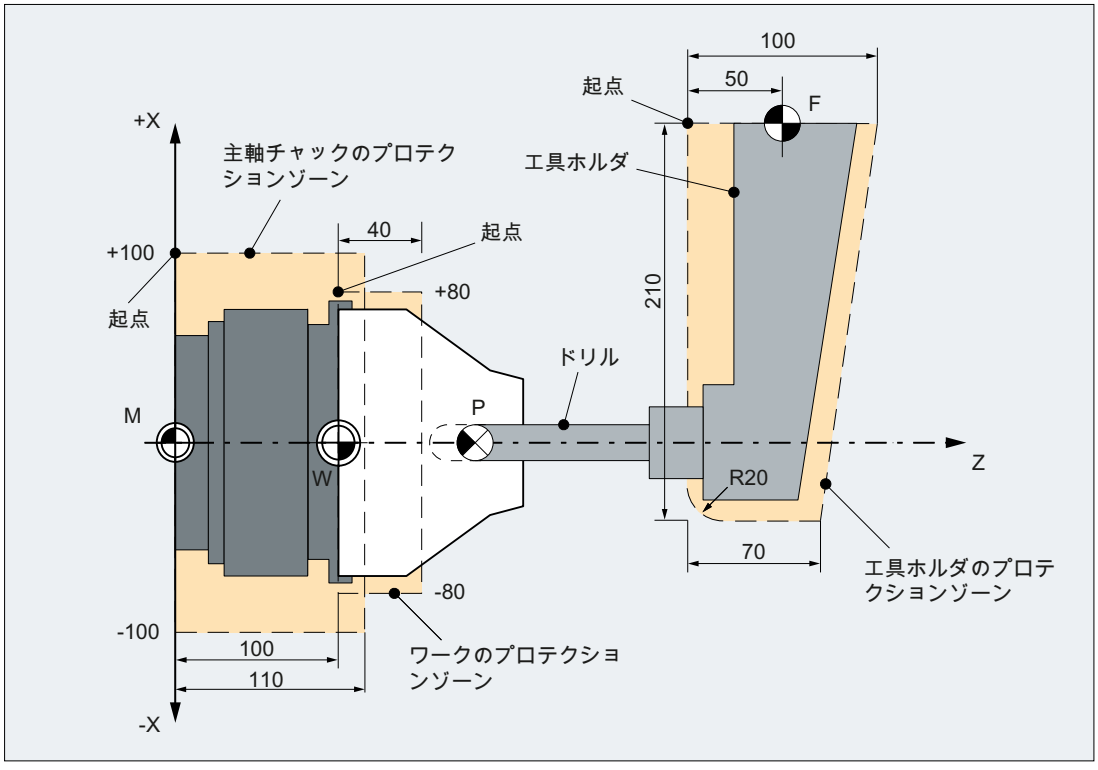
4.6.1 旋盤のプロテクションゾーン

旋盤に対して、次の内側プロテクションゾーンを定義するものとします。

- 主軸チャック用の機械別とワーク関連の 1 つのプロテクションゾーン(第 3 次元での制限なし)
- ワーク用のチャネル別の 1 つのプロテクションゾーン(第 3 次元での制限なし)
- 工具ホルダ用のチャネル別、工具関連の 1 つのプロテクションゾーン(第 3 次元での制限なし)

ワーク用プロテクションゾーンを定義するために、ワーク原点を機械原点に配置します。

プロテクションゾーンは有効になると、Z 軸のプラス方向に 100 mm シフトされます。



4.6.2 パートプログラムでのプロテクションゾーンの定義

プロテクションゾーン定義用のパートプログラムの抜粋:

プログラムコード	コメント
DEF INT AB	
G18	; 作業平面 ZX
NPROTDEF(1,FALSE,0,0,0)	; 定義の開始:主軸チャックのプロテクションゾーン
G01 X100 Z0	; 輪郭を記述します:輪郭に沿った移動動作
G01 X-100 Z0	; 輪郭を記述します:第 1 輪郭要素
G01 X-100 Z110	; 輪郭を記述します:第 2 輪郭要素
G01 X100 Z110	; 輪郭を記述します:第 3 輪郭要素
G01 X100 Z0	; 輪郭を記述します:第 4 輪郭要素
EXECUTE(AB)	; 定義の終わり:主軸チャックのプロテクションゾーン
CPROTDEF(1,FALSE,0,0,0)	; 定義の開始:ワークのプロテクションゾーン
G01 X80 Z0	; 輪郭を記述します:輪郭に沿った移動動作
G01 X-80 Z0	; 輪郭を記述します:第 1 輪郭要素
G01 X-80 Z40	; 輪郭を記述します:第 2 輪郭要素
G01 X80 Z40	; 輪郭を記述します:第 3 輪郭要素
G01 X80 Z0	; 輪郭を記述します:第 4 輪郭要素
EXECUTE(AB)	; 定義の終わり:ワークのプロテクションゾーン
CPROTDEF(2,TRUE,0,0,0)	; 定義の開始:工具ホルダのプロテクションゾーン
G01 X0 Z-50	; 輪郭を記述します:輪郭に沿った移動動作
G01 X-190 Z-50	; 輪郭を記述します:第 1 輪郭要素
G03 X-210 Z-30 I-20	; 輪郭を記述します:第 2 輪郭要素
G01 X-210 Z20	; 輪郭を記述します:第 3 輪郭要素
G01 X0 Z50	; 輪郭を記述します:第 4 輪郭要素
G01 X0 Z-50	; 輪郭を記述します:第 5 輪郭要素
EXECUTE(AB)	; 定義の終わり:工具ホルダのプロテクションゾーン

4.6.3 システム変数によるプロテクションゾーンの定義

主軸チャックの機械別プロテクションゾーン

システム変数	値	説明
$SN_PA_ACTIV_IMMED[0]	0	主軸チャックのプロテクションゾーンはすぐには有効にならない
$SN_PA_T_W[0]	0	主軸チャックのプロテクションゾーンはワーク関連
$SN_PA_ORI[0]	1	プロテクションゾーンの向き:1 = 3 番目と 1 番目のジオメトリ軸
$SN_PA_LIM_3DIM[0]	0	3 次元での制限のタイプ:0 = 制限なし

システム変数	値	説明
$SN_PA_PLUS_LIM[0]	0	3 次元での正方向の制限値
$SN_PA_MINUS_LIM[0]	0	3 次元での負方向の制限値
$SN_PA_CONT_NUM[0]	4	有効な輪郭要素の数
$SN_PA_CONT_TYP[0,0]	1	輪郭タイプ[<i>] :1 = 直線用 G1、 主軸チャックのプロテクションゾーン、輪郭要素 1
$SN_PA_CONT_TYP[0,1]	1	輪郭タイプ[<i>] :1 = 直線用 G1、 主軸チャックのプロテクションゾーン、輪郭要素 2
$SN_PA_CONT_TYP[0,2]	1	輪郭タイプ[<i>] :1 = 直線用 G1、 主軸チャックのプロテクションゾーン、輪郭要素 3
$SN_PA_CONT_TYP[0,3]	1	輪郭タイプ[<i>] :1 = 直線用 G1、 主軸チャックのプロテクションゾーン、輪郭要素 4
$SN_PA_CONT_TYP[0,4]	1	輪郭タイプ[<i>] :1 = 直線用 G1、 主軸チャックのプロテクションゾーン、輪郭要素 5
$SN_PA_CONT_TYP[0,5]	0	輪郭タイプ[<i>] :0 = 未定義、 主軸チャックのプロテクションーン、輪郭要素 6
$SN_PA_CONT_TYP[0,6]	0	輪郭タイプ[<i>] :0 = 未定義、 主軸チャックのプロテクションーン、輪郭要素 7
$SN_PA_CONT_TYP[0,7]	0	輪郭タイプ[<i>] :0 = 未定義、 主軸チャックのプロテクションーン、輪郭要素 8
$SN_PA_CONT_TYP[0,8]	0	輪郭タイプ[<i>] :0 = 未定義、 主軸チャックのプロテクションーン、輪郭要素 9
$SN_PA_CONT_TYP[0,9]	0	輪郭タイプ[<i>] :0 = 未定義、 主軸チャックのプロテクションーン、輪郭要素 10
$SN_PA_CONT_ORD[0,0]	-100	輪郭の終点[<i>]、主軸チャックの 縦座標値プロテクションゾーン、輪郭要素 1
$SN_PA_CONT_ORD[0,1]	-100	輪郭の終点[<i>]、主軸チャックの 縦座標値プロテクションゾーン、輪郭要素 2
$SN_PA_CONT_ORD[0,2]	100	輪郭の終点[<i>]、主軸チャックの 縦座標値プロテクションゾーン、輪郭要素 3
$SN_PA_CONT_ORD[0,3]	100	輪郭の終点[<i>]、主軸チャックの 縦座標値プロテクションゾーン、輪郭要素 4
$SN_PA_CONT_ORD[0,4]	0	輪郭の終点[<i>]、主軸チャックの 縦座標値プロテクションゾーン、輪郭要素 5

システム変数	値	説明
$SN_PA_CONT_ORD[0,5]	0	輪郭の終点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 6
$SN_PA_CONT_ORD[0,6]	0	輪郭の終点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 7
$SN_PA_CONT_ORD[0,7]	0	輪郭の終点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 8
$SN_PA_CONT_ORD[0,8]	0	輪郭の終点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 9
$SN_PA_CONT_ORD[0,9]	0	輪郭の終点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 10
$SN_PA_CONT_ABS[0,0]	0	輪郭の終点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 1
$SN_PA_CONT_ABS[0,1]	110	輪郭の終点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 2
$SN_PA_CONT_ABS[0,2]	110	輪郭の終点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 3
$SN_PA_CONT_ABS[0,3]	0	輪郭の終点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 4
$SN_PA_CONT_ABS[0,4]	0	輪郭の終点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 5
$SN_PA_CONT_ABS[0,5]	0	輪郭の終点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 6
$SN_PA_CONT_ABS[0,6]	0	輪郭の終点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 7
$SN_PA_CONT_ABS[0,7]	0	輪郭の終点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 8
$SN_PA_CONT_ABS[0,8]	0	輪郭の終点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 9
$SN_PA_CONT_ABS[0,9]	0	輪郭の終点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 10
$SN_PA_CENT_ORD[0,0]	0	円弧軌跡の中心点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 1
$SN_PA_CENT_ORD[0.1]	0	円弧軌跡の中心点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 2

システム変数	値	説明
$SN_PA_CENT_ORD[0,2]	0	円弧軌跡の中心点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 3
$SN_PA_CENT_ORD[0,3]	0	円弧軌跡の中心点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 4
$SN_PA_CENT_ORD[0,4]	0	円弧軌跡の中心点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 5
$SN_PA_CENT_ORD[0,5]	0	円弧軌跡の中心点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 6
$SN_PA_CENT_ORD[0,6]	0	円弧軌跡の中心点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 7
$SN_PA_CENT_ORD[0,7]	0	円弧軌跡の中心点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 8
$SN_PA_CENT_ORD[0,8]	0	円弧軌跡の中心点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 9
$SN_PA_CENT_ORD[0,9]	0	円弧軌跡の中心点[<i>]、主軸チャックの縦座標値プロテクションゾーン、輪郭要素 10
$SN_PA_CENT_ABS[0,0]	0	円弧軌跡の中心点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 1
$SN_PA_CENT_ABS[0.1]	0	円弧軌跡の中心点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 2
$SN_PA_CENT_ABS[0,2]	0	円弧軌跡の中心点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 3
$SN_PA_CENT_ABS[0,3]	0	円弧軌跡の中心点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 4
$SN_PA_CENT_ABS[0,4]	0	円弧軌跡の中心点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 5
$SN_PA_CENT_ABS[0,5]	0	円弧軌跡の中心点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 6
$SN_PA_CENT_ABS[0,6]	0	円弧軌跡の中心点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 7
$SN_PA_CENT_ABS[0,7]	0	円弧軌跡の中心点[<i>]、主軸チャックの横座標値プロテクションゾーン、輪郭要素 8

システム変数	値	説明
$SN_PA_CENT_ABS[0,8]	0	円弧軌跡の中心点[<i>]、主軸チャックの 横座標値プロテクションゾーン、輪郭要素 9
$SN_PA_CENT_ABS[0,9]	0	円弧軌跡の中心点[<i>]、主軸チャックの 横座標値プロテクションゾーン、輪郭要素 10

ワークのチャネル別プロテクションゾーン

システム変数	値	備考
$SC_PA_ACTIV_IMMED[0]	0	ワークのプロテクションゾーンはすぐには有効にならない
$SC_PA_TW[0]	0	ワークのプロテクションゾーンはワーク関連プロテクションゾーン
$SC_PA_ORI[0]	1	プロテクションゾーンの向き:1 = 3 番目と 1 番目のジオメトリ軸
$SC_PA_LIM_3DIM[0]	0	3 次元での制限のタイプ:0 = 制限なし
$SC_PA_PLUS_LIM[0]	0	3 次元での正方向の制限値
$SC_PA_MINUS_LIM[0]	0	3 次元での負方向の制限値
$SC_PA_CONT_NUM[0]	4	有効な輪郭要素の数
$SC_PA_CONT_TYP[0,0]	1	輪郭タイプ[<i>] :1 = 直線用 G1 、プロテクションゾーンのワーク、輪郭要素 1
$SC_PA_CONT_TYP[0,1]	1	輪郭タイプ[<i>] :1 = 直線用 G1 、プロテクションゾーンのワーク、輪郭要素 2
$SC_PA_CONT_TYP[0,2]	1	輪郭タイプ[<i>] :1 = 直線用 G1 、プロテクションゾーンのワーク、輪郭要素 3
$SC_PA_CONT_TYP[0,3]	1	輪郭タイプ[<i>] :1 = 直線用 G1 、プロテクションゾーンのワーク、輪郭要素 4
$SC_PA_CONT_TYP[0,4]	1	輪郭タイプ[<i>] :1 = 直線用 G1 、プロテクションゾーンのワーク、輪郭要素 5
$SC_PA_CONT_TYP[0,5]	0	輪郭タイプ[<i>] :0 = 未定義、 ワークのプロテクションゾーン、輪郭要素 6
$SC_PA_CONT_TYP[0,6]	0	輪郭タイプ[<i>] :0 = 未定義、 ワークのプロテクションゾーン、輪郭要素 7
$SC_PA_CONT_TYP[0,7]	0	輪郭タイプ[<i>] :0 = 未定義、 ワークのプロテクションゾーン、輪郭要素 8

システム変数	値	備考
$SC_PA_CONT_TYP[0,8]	0	輪郭タイプ[<i>] :0 = 未定義、 ワークのプロテクションゾーン、輪郭要素 9
$SC_PA_CONT_TYP[0,9]	0	輪郭タイプ[<i>] :0 = 未定義、 ワークのプロテクションゾーン、輪郭要素 10
$SC_PA_CONT_ORD[0,0]	-80	輪郭の終点[<i>]、ワークの 縦座標値プロテクションゾーン、輪郭要素 1
$SC_PA_CONT_ORD[0,1]	-80	輪郭の終点[<i>]、ワークの 縦座標値プロテクションゾーン、輪郭要素 2
$SC_PA_CONT_ORD[0,2]	80	輪郭の終点[<i>]、ワークの 縦座標値プロテクションゾーン、輪郭要素 3
$SC_PA_CONT_ORD[0,3]	80	輪郭の終点[<i>]、ワークの 縦座標値プロテクションゾーン、輪郭要素 4
$SC_PA_CONT_ORD[0,4]	0	輪郭の終点[<i>]、ワークの 縦座標値プロテクションゾーン、輪郭要素 5
$SC_PA_CONT_ORD[0,5]	0	輪郭の終点[<i>]、ワークの 縦座標値プロテクションゾーン、輪郭要素 6
$SC_PA_CONT_ORD[0,6]	0	輪郭の終点[<i>]、ワークの 縦座標値プロテクションゾーン、輪郭要素 7
$SC_PA_CONT_ORD[0,7]	0	輪郭の終点[<i>]、ワークの 縦座標値プロテクションゾーン、輪郭要素 8
$SC_PA_CONT_ORD[0,8]	0	輪郭の終点[<i>]、ワークの 縦座標値プロテクションゾーン、輪郭要素 9
$SC_PA_CONT_ORD[0,9]	0	輪郭の終点[<i>]、ワークの 縦座標値プロテクションゾーン、輪郭要素 10
$SC_PA_CONT_ABS[0,0]	0	輪郭の終点[<i>]、ワークの 横座標値プロテクションゾーン、輪郭要素 1
$SC_PA_CONT_ABS[0,1]	40	輪郭の終点[<i>]、ワークの 横座標値プロテクションゾーン、輪郭要素 2
$SC_PA_CONT_ABS[0,2]	40	輪郭の終点[<i>]、ワークの 横座標値プロテクションゾーン、輪郭要素 3
$SC_PA_CONT_ABS[0,3]	0	輪郭の終点[<i>]、ワークの 横座標値プロテクションゾーン、輪郭要素 4
$SC_PA_CONT_ABS[0,4]	0	輪郭の終点[<i>]、ワークの 横座標値プロテクションゾーン、輪郭要素 5

システム変数	値	備考
$SC_PA_CONT_ABS[0,5]	0	輪郭の終点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 6
$SC_PA_CONT_ABS[0,6]	0	輪郭の終点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 7
$SC_PA_CONT_ABS[0,7]	0	輪郭の終点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 8
$SC_PA_CONT_ABS[0,8]	0	輪郭の終点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 9
$SC_PA_CONT_ABS[0,9]	0	輪郭の終点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 10
$SC_PA_CENT_ORD[0,0]	0	円弧軌跡の中心点[<i>]、ワークの縦座標値プロテクションゾーン、輪郭要素 1
$SC_PA_CENT_ORD[0,1]	0	円弧軌跡の中心点[<i>]、ワークの縦座標値プロテクションゾーン、輪郭要素 2
$SC_PA_CENT_ORD[0,2]	0	円弧軌跡の中心点[<i>]、ワークの縦座標値プロテクションゾーン、輪郭要素 3
$SC_PA_CENT_ORD[0,3]	0	円弧軌跡の中心点[<i>]、ワークの縦座標値プロテクションゾーン、輪郭要素 4
$SC_PA_CENT_ORD[0,4]	0	円弧軌跡の中心点[<i>]、ワークの縦座標値プロテクションゾーン、輪郭要素 5
$SC_PA_CENT_ORD[0,5]	0	円弧軌跡の中心点[<i>]、ワークの縦座標値プロテクションゾーン、輪郭要素 6
$SC_PA_CENT_ORD[0,6]	0	円弧軌跡の中心点[<i>]、ワークの縦座標値プロテクションゾーン、輪郭要素 7
$SC_PA_CENT_ORD[0,7]	0	円弧軌跡の中心点[<i>]、ワークの縦座標値プロテクションゾーン、輪郭要素 8
$SC_PA_CENT_ORD[0,8]	0	円弧軌跡の中心点[<i>]、ワークの縦座標値プロテクションゾーン、輪郭要素 9
$SC_PA_CENT_ORD[0,9]	0	円弧軌跡の中心点[<i>]、工具ホルダの縦座標値プロテクションゾーン、輪郭要素 10
$SC_PA_CENT_ABS[0,0]	0	円弧軌跡の中心点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 1
$SC_PA_CENT_ABS[0,1]	0	円弧軌跡の中心点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 2

システム変数	値	備考
$SC_PA_CENT_ABS[0,2]	0	円弧軌跡の中心点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 3
$SC_PA_CENT_ABS[0,3]	0	円弧軌跡の中心点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 4
$SC_PA_CENT_ABS[0,4]	0	円弧軌跡の中心点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 5
$SC_PA_CENT_ABS[0,5]	0	円弧軌跡の中心点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 6
$SC_PA_CENT_ABS[0,6]	0	円弧軌跡の中心点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 7
$SC_PA_CENT_ABS[0,7]	0	円弧軌跡の中心点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 8
$SC_PA_CENT_ABS[0,8]	0	円弧軌跡の中心点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 9
$SC_PA_CENT_ABS[0,9]	0	円弧軌跡の中心点[<i>]、ワークの横座標値プロテクションゾーン、輪郭要素 10

工具ホルダのチャネル別プロテクションゾーン

システム変数	値	備考
$SC_PA_ACTIV_IMMED[1]	0	工具ホルダのプロテクションゾーンはすぐには有効にならない
$SC_PA_TW[1]	3	工具ホルダのプロテクションゾーンは工具関連
$SC_PA_ORI[1]	1	プロテクションゾーンの向き:1 = 3 番目と 1 番目のジオメトリ軸
$SC_PA_LIM_3DIM[1]	0	3 次元での制限のタイプ:0 = 制限なし
$SC_PA_PLUS_LIM[1]	0	3 次元での正方向の制限値
$SC_PA_MINUS_LIM[1]	0	3 次元での負方向の制限値
$SC_PA_CONT_NUM[1]	5	有効な輪郭要素の数
$SC_PA_CONT_TYP[1,0]	1	輪郭タイプ[<i>] :1 = 直線用 G1、プロテクションゾーンの工具ホルダ、輪郭要素 1
$SC_PA_CONT_TYP[1,1]	3	輪郭タイプ[<i>] :3 = 円弧要素用 G3、反時計方向、工具ホルダの保護領域、輪郭要素 2
$SC_PA_CONT_TYP[1,2]	1	輪郭タイプ[<i>] :1 = 直線用 G1、プロテクションゾーンの工具ホルダ、輪郭要素 3

システム変数	値	備考
$SC_PA_CONT_TYP[1,3]	1	輪郭タイプ[<i>] :1 = 直線用 G1 、プロテクションゾーンの工具ホルダ、輪郭要素 4
$SC_PA_CONT_TYP[1,4]	1	輪郭タイプ[<i>] :1 = 直線用 G1 、プロテクションゾーンの工具ホルダ、輪郭要素 5
$SC_PA_CONT_TYP[1,5]	0	輪郭タイプ[<i>] :0 = 未定義、 工具ホルダのプロテクションゾーン、輪郭要素 6
$SC_PA_CONT_TYP[1,6]	0	輪郭タイプ[<i>] :0 = 未定義、 工具ホルダのプロテクションゾーン、輪郭要素 7
$SC_PA_CONT_TYP[1,7]	0	輪郭タイプ[<i>] :0 = 未定義、 工具ホルダのプロテクションゾーン、輪郭要素 8
$SC_PA_CONT_TYP[1,8]	0	輪郭タイプ[<i>] :0 = 未定義、 工具ホルダのプロテクションゾーン、輪郭要素 9
$SC_PA_CONT_TYP[1,9]	0	輪郭タイプ[<i>] :0 = 未定義、 工具ホルダのプロテクションゾーン、輪郭要素 10
$SC_PA_CONT_ORD[1,0]	-190	輪郭の終点[<i>]、工具ホルダの 縦座標値プロテクションゾーン、輪郭要素 1
$SC_PA_CONT_ORD[1,1]	-210	輪郭の終点[<i>]、工具ホルダの 縦座標値プロテクションゾーン、輪郭要素 2
$SC_PA_CONT_ORD[1,2]	-210	輪郭の終点[<i>]、工具ホルダの 縦座標値プロテクションゾーン、輪郭要素 3
$SC_PA_CONT_ORD[1,3]	0	輪郭の終点[<i>]、工具ホルダの 縦座標値プロテクションゾーン、輪郭要素 4
$SC_PA_CONT_ORD[1,4]	0	輪郭の終点[<i>]、工具ホルダの 縦座標値プロテクションゾーン、輪郭要素 5
$SC_PA_CONT_ORD[1,5]	0	輪郭の終点[<i>]、工具ホルダの 縦座標値プロテクションゾーン、輪郭要素 6
$SC_PA_CONT_ORD[1,6]	0	輪郭の終点[<i>]、工具ホルダの 縦座標値プロテクションゾーン、輪郭要素 7
$SC_PA_CONT_ORD[1,7]	0	輪郭の終点[<i>]、工具ホルダの 縦座標値プロテクションゾーン、輪郭要素 8
$SC_PA_CONT_ORD[1,8]	0	輪郭の終点[<i>]、工具ホルダの 縦座標値プロテクションゾーン、輪郭要素 9
$SC_PA_CONT_ORD[1,9]	0	輪郭の終点[<i>]、工具ホルダの 縦座標値プロテクションゾーン、輪郭要素 10

システム変数	値	備考
$SC_PA_CONT_ABS[1,0]	-50	輪郭の終点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 1
$SC_PA_CONT_ABS[1,1]	-30	輪郭の終点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 2
$SC_PA_CONT_ABS[1,2]	20	輪郭の終点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 3
$SC_PA_CONT_ABS[1,3]	50	輪郭の終点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 4
$SC_PA_CONT_ABS[1,4]	-50	輪郭の終点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 5
$SC_PA_CONT_ABS[1,5]	0	輪郭の終点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 6
$SC_PA_CONT_ABS[1,6]	0	輪郭の終点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 7
$SC_PA_CONT_ABS[1,7]	0	輪郭の終点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 8
$SC_PA_CONT_ABS[1,8]	0	輪郭の終点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 9
$SC_PA_CONT_ABS[1,9]	0	輪郭の終点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 10
$SC_PA_CENT_ORD[1,0]	0	円弧軌跡の中心点[<i>]、工具ホルダの縦座標値プロテクションゾーン、輪郭要素 1
$SC_PA_CENT_ORD[1,1]	-190	円弧軌跡の中心点[<i>]、工具ホルダの縦座標値プロテクションゾーン、輪郭要素 2
$SC_PA_CENT_ORD[1,2]	0	円弧軌跡の中心点[<i>]、工具ホルダの縦座標値プロテクションゾーン、輪郭要素 3
$SC_PA_CENT_ORD[1,3]	0	円弧軌跡の中心点[<i>]、工具ホルダの縦座標値プロテクションゾーン、輪郭要素 4
$SC_PA_CENT_ORD[1,4]	0	円弧軌跡の中心点[<i>]、工具ホルダの縦座標値プロテクションゾーン、輪郭要素 5
$SC_PA_CENT_ORD[1,5]	0	円弧軌跡の中心点[<i>]、工具ホルダの縦座標値プロテクションゾーン、輪郭要素 6
$SC_PA_CENT_ORD[1,6]	0	円弧軌跡の中心点[<i>]、工具ホルダの縦座標値プロテクションゾーン、輪郭要素 7

システム変数	値	備考
$SC_PA_CENT_ORD[1,7]	0	円弧軌跡の中心点[<i>]、工具ホルダの縦座標値プロテクションゾーン、輪郭要素 8
$SC_PA_CENT_ORD[1,8]	0	円弧軌跡の中心点[<i>]、工具ホルダの縦座標値プロテクションゾーン、輪郭要素 9
$SC_PA_CENT_ORD[1,9]	0	円弧軌跡の中心点[<i>]、工具ホルダの縦座標値プロテクションゾーン、輪郭要素 10
$SC_PA_CENT_ABS[1,0]	0	円弧軌跡の中心点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 1
$SC_PA_CENT_ABS[1,1]	-30	円弧軌跡の中心点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 2
$SC_PA_CENT_ABS[1,2]	0	円弧軌跡の中心点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 3
$SC_PA_CENT_ABS[1,3]	0	円弧軌跡の中心点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 4
$SC_PA_CENT_ABS[1,4]	0	円弧軌跡の中心点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 5
$SC_PA_CENT_ABS[1,5]	0	円弧軌跡の中心点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 6
$SC_PA_CENT_ABS[1,6]	0	円弧軌跡の中心点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 7
$SC_PA_CENT_ABS[1,7]	0	円弧軌跡の中心点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 8
$SC_PA_CENT_ABS[1,8]	0	円弧軌跡の中心点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 9
$SC_PA_CENT_ABS[1,9]	0	円弧軌跡の中心点[<i>]、工具ホルダの横座標値プロテクションゾーン、輪郭要素 10

4.6.4 プロテクションゾーンの起動:

主軸チャックとワーク、工具ホルダ用のプロテクションゾーンを有効にするためのパートプログラムの抜粋:

プログラムコード	コメント
`NPROT(1,2,0,0,0)`	`; プロテクションゾーン:主軸チャック`
`CPROT(1,2,0,0,100)`	`; プロテクションゾーン:Z 軸において 100mm のオフセットを持つワーク`

プログラムコード	コメント
CPROT(2,2,0,0,0)	; プロテクションゾーン:工具ホルダ

4.7 データリスト

4.7.1 マシンデータ

4.7.1.1 NC 別マシンデータ

番号	識別子: $MN_	説明
10618	PROTAREA_GEOAX_CHANGE_MODE	座標変換の変更/ジオメトリ軸の入れ替えの動作
18190	MM_NUM_PROTECT_AREA_NCK	有効な機械別プロテクションゾーンの数

4.7.1.2 チャネルマシンデータ

番号	識別子: $MC_	説明
28200	MM_NUM_PROTECT_AREA_CHAN (SRAM)	有効なチャネル別プロテクションゾーンの数
28210	MM_NUM_PROTECT_AREA_ACTIVE	チャネルで同時に有効にできるプロテクションゾーンの最大数
28212	MM_NUM_PROTECT_AREA_CONTUR	チャネルで定義可能な輪郭要素の最大数

4.7.2 信号

4.7.2.1 チャネルへの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
プロテクションゾーン移動許可	DB21、... .DBX1.1	DB320x.DBX1.1
送り無効	DB21、... .DBX6.0	DB320x.DBX6.0
機械別プロテクションゾーン 1 ... 8 有効	DB21、... .DBX8.0 ... 7	DB320x.DBX8.0 ... 7

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
機械別プロテクションゾーン 9 有効	DB21、... .DBX9.0	DB320x.DBX9.0
機械別プロテクションゾーン 10 有効	DB21、... .DBX9.1	DB320x.DBX9.1
チャネル別プロテクションゾーン 1 ... 8 有効	DB21、... .DBX10.0 ... 7	DB320x.DBX10.0 ... 7
チャネル別プロテクションゾーン 9 有効	DB21、... .DBX11.0	DB320x.DBX11.0
チャネル別プロテクションゾーン 10 有効	DB21、... .DBX11.1	DB320x.DBX11.1

4.7.2.2 チャネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
機械別プロテクションゾーン 1 ... 8 起動待ち	DB21、... .DBX272.0 ... 7	DB330x.DBX8.0 ... 7
機械別プロテクションゾーン 9 起動待ち	DB21、... .DBX273.0	DB330x.DBX9.0
機械別プロテクションゾーン 10 起動待ち	DB21、... .DBX273.1	DB330x.DBX9.1
チャネル別プロテクションゾーン 1 ... 8 起動待ち	DB21、... .DBX274.0 ... 7	DB330x.DBX10.0 ... 7
チャネル別プロテクションゾーン 9 起動待ち	DB21、... .DBX275.0	DB330x.DBX11.0
チャネル別プロテクションゾーン 10 起動待ち	DB21、... .DBX275.1	DB330x.DBX11.1
機械別プロテクションゾーン 1 ... 8 起動待ち	DB21、... .DBX276.0 ... 7	DB330x.DBX12.0 ... 7
機械別プロテクションゾーン 9 違反	DB21、... .DBX277.0	DB330x.DBX13.0
機械別プロテクションゾーン 10 違反	DB21、... .DBX277.1	DB330x.DBX13.1
チャネル別プロテクションゾーン 1 ... 8 違反	DB21、... .DBX278.0 ... 7	DB330x.DBX14.0 ... 7
チャネル別プロテクションゾーン 9 違反	DB21、... .DBX279.0	DB330x.DBX15.0
チャネル別プロテクションゾーン 10 違反	DB21、... .DBX279.1	DB330x.DBX15.1

B1: 連続軌跡モード、イグザクトストップ、先読み: 5

5.1 概略説明

イグザクトストップまたはイグザクトストップモード

イグザクトストップの移動モードでは、移動動作に関係するすべての軸(モーダル移動モードの軸を除く)は、各ブロックの終点で停止状態になるまで減速されます。次のブロックへの遷移が行われるのは、移動動作に関係するすべての軸が、選択されたイグザクトストップ条件に応じて、プログラムされた目標位置に到達した場合のみです。

連続軌跡モード

連続軌跡モードでは、NC はプログラムされている軌跡速度を、できるだけ一定に保とうとします。特に、パートプログラムのブロック境界での軌跡軸の減速が回避されます。

先読み

先読みは、連続軌跡モードを最適化するための機能です。

高品質の表面仕上げを確保するには、ワークの滑らかで均一的な加工が必要です。そのため、加工中は軌跡速度変化をできるだけ回避すべきです。先読みを使用しない場合、NC では、可能な軌跡速度を決定するときに、現在の移動ブロックの直後の移動ブロックのみを考慮します。直後のブロックに短い軌跡しか含まれていない場合、必要に応じて次のブロックの終点で時間内に停止できるように、NC は軌跡速度を下げる(実行中のブロックで減速する)必要があります。

現在の移動ブロックの後に続く設定可能な数の移動ブロックを NC で「先読み」すると、特定の状況下では、はるかに速い軌跡速度が得られます。これは、より多くの移動ブロックとより多くの軌跡を使用して、NC で計算できるようになるためです。

これには、以下のメリットがあります:

- 平均してより速い軌跡速度での加工
- 減速および加速を回避することによる加工面品質の改善

スムージング軌跡速度

「軌跡速度のスムージング」は、極めて安定した一貫性のある軌跡速度を必要とする用途(金型製造での高速フライス加工など)専用の機能です。機械共振の高周波励起の原因となる減速および加速処理は、「軌跡速度のスムージング」機能で回避されます。

これには、以下のメリットがあります:

- 機械共振の励起を回避することによる、加工面品質の改善と加工時間の短縮
- 「不要な」加速処理、つまり、プログラム実行時間がそれほど改善されない加速処理を回避した、軌跡速度および切削速度の一定波形。

ダイナミック軌跡応答の補正

「軌跡速度のスムージング」に加えて、ダイナミック軌跡応答を最適化すると同時に、機械共振の高周波励起を回避できる「ダイナミック応答調整」という機能があります。この目的のために、軌跡速度の高周波の変化は、自動的に、マシンデータで設定されているダイナミック応答制限値パラメータより低い加々速度値や加速度値で実行されます。

したがって、軌跡速度の低周波の変化の場合は、最大のダイナミック応答制限値が適用されるのに対して、高周波の変化の場合は、自動ダイナミック応答調整のために抑えられたダイナミック応答制限値のみが適用されます。

軌跡補間のダイナミック応答モード

軌跡ダイナミック応答を最適化すると、用途ごとのダイナミック応答が設定されます。これは、さまざまな加工用途(タッピング、荒削り、スムージングなど)に対して事前設定されており、パートプログラムでそれぞれのダイナミック応答モードを呼び出して起動できます。

自由形式面モード

曲率または軸ねじれで変動が発生すると、軌跡速度が変化します。これにより、通常は、自由形式面のワークの加工中に不要な減速および加速が発生するため、ワーク表面の品質に悪影響を及ぼす場合があります。

自由形式面の加工には、以下の機能を使用できます。

- 「自由曲面モード:基本機能」の起動
  これにより、軌跡速度波形の定義で、曲率および軸ねじれでの変更に対する「感度が低下」します。
- 「自由曲面モード:拡張機能」用メモリ
  標準先読み機能のこの拡張を使用して、自由形式面の加工時の軌跡速度波形を計算します。

自由形式面モードの利点は、ワーク表面がより均質になり、機械の応力レベルが低下することです。

直線ブロックの圧縮

CAD/CAM システムでワークの設計が完了すると、その CAD/CAM システムは一般に、ワーク表面を作成するための、対応するパートプログラムのコンパイルも行います。そのために、大部分の CAD/CAM システムは直線ブロックを使用して、ワーク表面の曲線部も記述します。必要な輪郭精度を維持するには、一般に多くの補間点が必要です。この結果、通常非常に短い軌跡を持つ多くの直線ブロックが発生します。

「圧縮機能」は多項式ブロックを使用して、直線ブロックによって指定された輪郭の連続近似を実行します。この処理中、割り付け可能な数の直線ブロックが一つの多項式ブロックに置換されます。

長所:

- 移動ブロック数の削減
- 軌跡速度の向上
- 加工面品位の向上
- 連続したブロック遷移部

短いスプラインブロックの圧縮

スプラインは、第 2 レベルまたは第 3 レベルの多項式から成るカーブを定義します。スプライン補間によりコントロールシステムでは、指定された少数の輪郭セット補間点からスムーズなカーブ特性を生成できます。

直線補間と比較した場合のスプライン補間の利点は、次のとおりです。

- 輪郭の曲線部の記述に必要なパートプログラムブロック数の削減
- パートプログラムブロック間であっても、機械システムに対する応力を低減するソフトな特性。

直線補間と比較した場合のスプライン補間の欠点は、次のとおりです。

- スプライン曲線の場合、正確な曲線特性はなく、許容誤差範囲しか指定できません。スプライン曲線はこの範囲内に存在します。

直線補間と同様に、スプラインの処理によって短いブロックが作成されるため、スプラインブロックを補間できるように、軌跡速度を下げる必要がある場合があります。スプラインに長く、滑らかな曲線がある場合も同様です。「短いスプラインブロックの圧縮」機能を使用すると、これらのスプラインブロックを結合し、得られるブロック長が十分になり、軌跡速度が減少されないようにすることができます。

5.2 イグザクトストップモード

イグザクトストップまたはイグザクトストップモード

イグザクトストップ移動モードでは、移動動作に関与するすべての軌跡軸と付加軸が連続的には移動せず、各ブロックの終点で減速して停止状態になります。次のブロックへの遷移が行われるのは、移動動作に関係するすべての軸が、選択されたイグザクトストップ条件に応じて、プログラムされた目標位置に到達した場合のみです。

この結果、次の動作が得られます。

- プログラム実行時間は、連続軌跡モードと比較してかなり長くなります。これは、軸の減速と、関係するすべての機械軸について「イグザクトストップ」状態に達するまでの待機時間があるためです。
- イグザクトストップモードでは、加工中にワーク表面にアンダーカットが発生する可能性があります。

イグザクトストップ条件

次のイクザクトストップ条件を設定できます。

- 「汎用イグザクトストップ」
- 「精密イグザクトストップ」
- 「補間終了」

用途

イグザクトストップモードは、プログラムした輪郭を正確に実行する必要がある場合に常に使用してください。

起動

プログラムでは、モーダルまたはブロック別に、次の命令を使用してイグザクトストップ動作を指定できます。

命令	意味
G60	イグザクトストップ動作は、現在の**モーダル**ブロックから有効です。
G9	イグザクトストップ動作は、現在のブロックで有効です。

イグザクトストップ条件「汎用イグザクトストップ」および「精密イグザクトストップ」

マシンデータでパラメータ設定され、指令位置に対して現在の追従誤差が許容範囲以下である場合、イグザクトストップ条件「汎用」または「精密」が機械軸で満たされます。

- MD36000 $MA_STOP_LIMIT_COARSE (汎用イグザクトストップ)
- MD36010 $MA_STOP_LIMIT_FINE (精密イグザクトストップ)

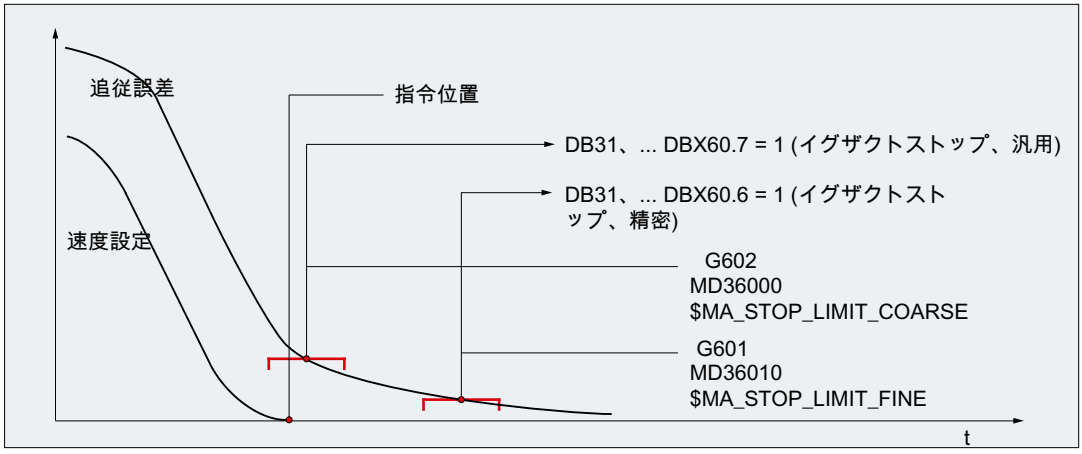


図 5-1　イグザクトストップ条件の許容範囲

注記

次の必要条件を満たすように、イグザクトストップ条件「汎用イグザクトストップ」および「精密イグザクトストップ」の許容範囲を割り当ててください。

「汎用イグザクトストップ」>「精密イグザクトストップ」

イグザクトストップ条件「補間終了」

イグザクトストップ条件「補間終了」では、移動動作に関連する軌跡軸およびノンモーダル付加軸がすべて、指令値の観点からブロック内でプログラムされた位置に到達するとすぐに、次のブロックへのブロックの変更が実行されます。つまり、補間器によりそのブロックが実行された直後です。

イグザクトストップ条件「補間終了」の場合、関連する機械軸の実際の位置および追従誤差は考慮されません。したがって、機械軸のダイナミック応答によっては、イグザクトストップ条件「汎用イグザクトストップ」および「精密イグザクトストップ」と比較して、ブロック変更時の輪郭の平滑量が比較的大きくなる可能性があります。

指令可能なイグザクトストップの条件の有効化

指令可能なイグザクトストップ条件は、次の命令を使用して有効化します。

命令	イグザクトストップ条件
G601	精密イグザクトストップ
G602	汎用イグザクトストップ
G603	補間終了

イグザクトストップ条件に対応したブロック変更

次の図は、選択されたイグザクトストップ条件に関するブロック変更のタイミングを示しています。

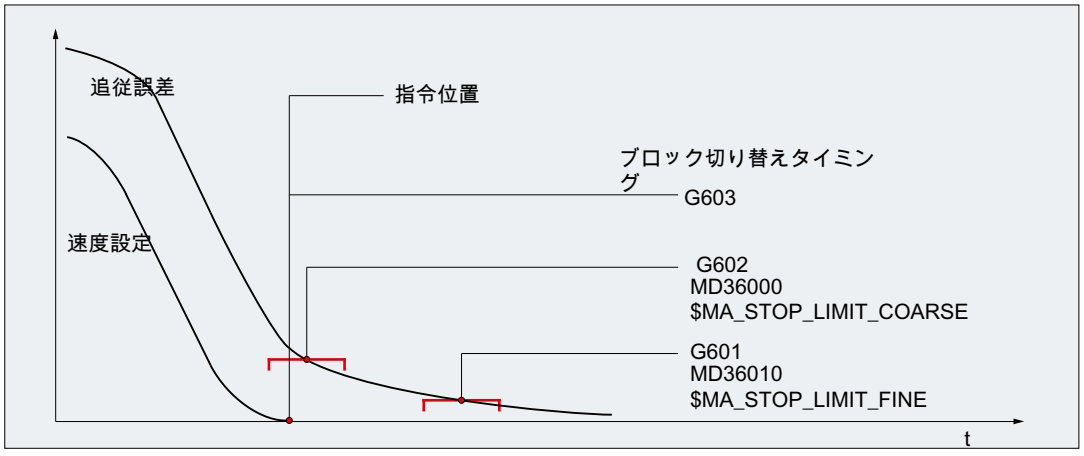


図 5-2　イグザクトストップ条件に対応したブロック変更

イグザクトストップ条件の加重係数

イグザクトストップ条件のパラメータセット依存の使用は、次の軸マシンデータを介して指定できます。

MD36012 $MA_STOP_LIMIT_FACTOR[<パラメータセット>] = <値>

この倍率は、次のマシンデータの値に適用されます。

- MD36000 $MA_STOP_LIMIT_COARSE
- MD36010 $MA_STOP_LIMIT_FINE
- MD36030 $MA_STANDSTILL_POS_TOL

用途例

- 新しいギヤボックス選択の実行後など、異なる質量比に対する位置決め応答の調整
- 荒削り、仕上げ削りなど、さまざまな加工状態に対応するインポジション時間の短縮

有効なイグザクトストップ条件のパラメータ設定可能な入力

1 番目の G グループの命令に対応するイグザクトストップ条件は、固定値として指定します。その結果として、プログラムされたイグザクトストップ条件は有効ではなくなります。

イグザクトストップ条件は、次の命令について**相互に無関係に**設定できます。

- 早送り G0
- 1 番目の G グループの他のすべての命令

このイグザクトストップ条件は、次の 10 進コードマシンデータを使用して、チャネルごとに設定します。

MD20550 $MC_EXACT_POS_MODE = <Z><E>

Z	E	有効なイグザクトストップ条件
0	0	プログラムされたイグザクトストップ条件
1	1	G601 (精密イグザクトストップ範囲)
2	2	G602 (汎用イグザクトストップ範囲)
3	3	G603 (補間終了)
E (1 の位):早送りのイグザクトストップ条件の設定。 Z (10 の位):1 番目の G グループの全命令すべてに対応するイグザクトストップ条件の設定。		

例

MD20550 $MC_EXACT_POS_MODE = 02

- <E> = 2:早送りを使用する場合、パートプログラム内のプログラムに関係なく、イグザクトストップ条件 G602 (汎用イグザクトストップ範囲)が常に有効になります。
- <Z> = 0:1 番目の G グループの他の全命令による移動時に、プログラムされたイグザクトストップ条件は有効です。

G0 ↔ G0 以外から連続軌跡モードへのブロック遷移のブロック変更特性

連続軌跡モードでは、早送りブロックと早送り以外のブロック(G0 ↔ G0 以外)間のブロック変更特性は、次のマシンデータを使用して設定できます。

MD20552 $MC_EXACT_POS_MODE_G0_TO_G1 = <値>

<値>	意味
0	ブロック遷移時の追加停止なし
1	ブロック遷移時に停止。 この特性は G601 に対応します(イグザクトストップウィンドウ、精密)。
2	ブロック遷移時に停止。 この特性は G602 に対応します(イグザクトストップウィンドウ、汎用)。
3	ブロック遷移時に停止。 この特性は、G603(補間終了)に対応します
4	ブロック遷移時に停止**なし**。 **G0→G0 以外**からの**ブロック変更**について連続軌跡モードでは、G0 ブロック内で後続の G0 以外のブロックの**送り速度補正**の現在値は、述語的に考慮されます。軸のダイナミックパフォーマンスおよび現在のブロックのパスの長さに応じて、次のブロックの正確なまたは最善適応の速度で、ブロック変更が実行されます。
5	ブロック遷移時に停止**なし**。 連続軌跡モードでは、G0→G0 以外および G0 以外→G0 からの**ブロック変更**について、**送り速度補正**の現在値(**G0→G0 以外**)、または次のブロックの**早送り軌跡**(**G0 以外→G0**)は、述語的に考慮されます。軸のダイナミックパフォーマンスおよび現在のブロックのパスの長さに応じて、次のブロックの正確なまたは最善適応の速度で、ブロック変更が実行されます。

5.3 連続軌跡モード

5.3.1 共通動作

連続軌跡モード

連続軌跡モードでは、軌跡速度はイグザクトストップ条件の実施を可能にするためのブロック切り替えに対する減速をおこない**ません**。 このモードの目的は、プログラムが次のブロックに移動するときに、軸速度ができるだけ一定のままであるように、ブロック切り替え点での軌跡軸の急激な減速を回避することです。 この目的を達成するために、連続軌跡モードが選択されたときに「先読み」機能も有効になります。

連続軌跡モードでは、プログラム指令輪郭の部分的な変更により、ブロック遷移の角が滑らかにされたり、その形状が少し変更されたりします。 プログラム指令輪郭に対する変更の範囲は、過負荷係数または丸み付けの条件を指定することにより制限できます。

連続軌跡モード:

- 輪郭の丸み付け。
- イグザクトストップ条件を満たすために必要な減速処理と加速処理が無くなるため、加工時間を短縮します。
- さらに速度変動がなくなり、切削条件が向上します。

連続軌跡モードは、次の場合は適していません。

- 輪郭が可能な限り速く移動する必要があるとき(たとえば早送りによる)
- 連続輪郭を得るために、輪郭精度に、プログラム指令輪郭から指定の許容範囲内の誤差があってもよい。

連続軌跡モードは、次の場合に適しません。

- 輪郭が正確に移動される必要があるとき
- 完全に一定の速度が必要なとき

自動イグザクトストップ

場合によっては、後続の動作を実行できるようにするために、連続軌跡モードでイグザクトストップを実施する必要があります。 このような状況では、軌跡速度はゼロまで減速します。

- 補助機能が移動動作の前に出力される場合、選択されたイグザクトストップ条件が満たされたときだけ、前のブロックを終了します。
- 補助機能が移動動作の後で出力される場合、それは、そのブロックの補間終了後に出力されます。
- 実行可能なブロック(位置決め軸の開始など)に軌跡軸の移動情報が含まれていない場合、前のブロックは選択されたイグザクトストップ条件に到達すると終了します。
- 位置決め軸がジオメトリ軸になるように宣言されたとき、前のブロックでジオメトリ軸をプログラム指令していると、前のブロックが補間終了で終了します。
- 直前に位置決め軸または主軸(付加軸の初期設定は位置決め軸です)としてプログラム指令された同期軸を指令した場合は、前のブロックは補間終了で終了します。
- 座標変換が変更された場合、前に処理されたブロックは、有効なイグザクトストップ条件により終了します。
- 次のブロックに加減速方法の切り替え BRISK/SOFT が含まれる場合は、ブロックが補間終了と同時に終了します(「B2:加速度 (ページ 317)」の章を参照してください)。
- 「バッファクリア」機能をプログラム指令した場合は、前のブロックが、選択したイグザクトストップ条件に到達したとき、終了します。

連続軌跡モードでの速度 = 0

自動イグザクトストップの応答に関係なく、以下の場合には軌跡移動がブロック終点でゼロ速度まで減速されます。

- 位置決め軸が命令 POS でプログラム指令され、その移動時間が軌跡軸より長い場合。 位置決め軸の「精密イグザクトストップ」に到達すると、ブロック切り替えがおこなわれます。
- 命令 SPOS でプログラム指令される主軸の位置決めにかかる時間が、軌跡軸の移動時間より長い場合。 位置決め主軸の「精密イグザクトストップ」に到達すると、ブロック切り替えが実行されます。

- 現在のブロックには、ジオメトリ軸用の移動命令があり、次のブロックには同期軸用の移動命令が含まれる場合。または逆に、現在のブロックに同期軸用の移動命令があり、次のブロックにジオメトリ軸用の移動命令が含まれる場合。
- 同期制御が必要な場合。

5.3.2 過負荷係数による速度低減

機能

この機能は、接線の変化が不連続なブロック遷移が 1 回の補間周期で移動できるまで、加速度制限に留意し、過負荷係数を考慮しながら、連続軌跡モードの軌跡速度を低下させます。

速度低下により、ブロック遷移の接線の変化が不連続な輪郭で、軸速度の不連続変化が発生します。 これらの速度の不連続変化は、連動する同期軸においてもおこなわれます。 速度の不連続変化は、軌跡速度がゼロに低下するのを防ぎます。 軸速度が軸加速度にしたがって、不連続変化により新しい指令値に到達できる速度まで低下した時に、この不連続変化がおこなわれます。 指令値の不連続変化の大きさは、過負荷係数を使用して制限できます。 不連続変化の大きさは各軸に依存するため、ブロック遷移中に有効な軌跡軸の最小不連続変化が、ブロック切り替え中に考慮されます。

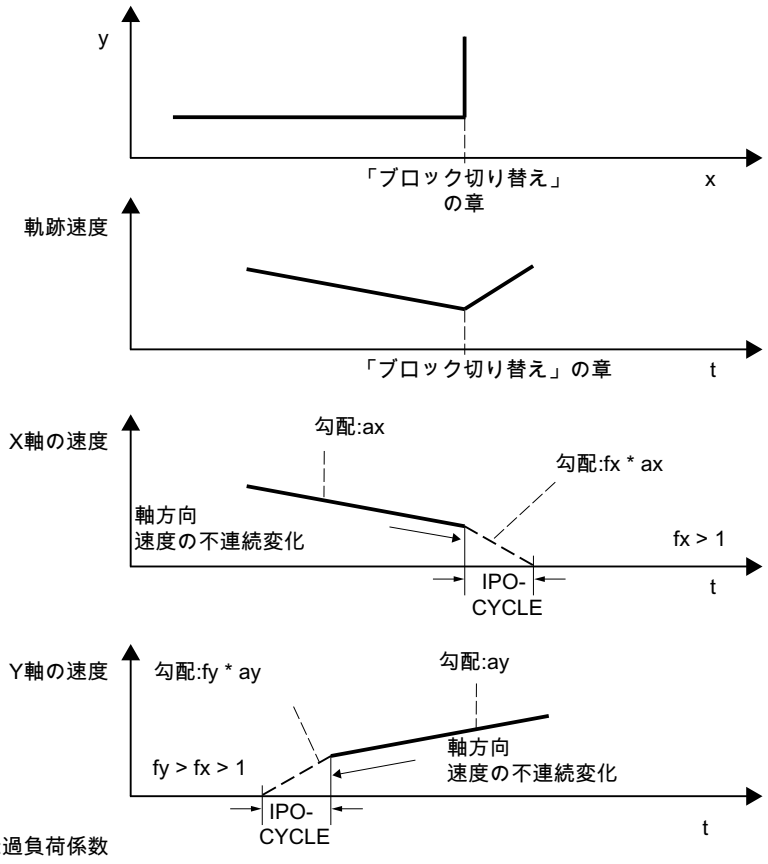


図 5-3　ブロック遷移での軸速度の変化

実際の接線の変化が連続的なブロック遷移では、許容可能な軸加速度を超えない場合、軌跡速度は低下しません。 これは、輪郭の非常に小さい曲がり(0.5 度など)は、すぐに通過するということです。

過負荷係数

過負荷係数は、ブロックの終点での機械軸速度のステップ変化を制限します。 速度不連続変化が軸の最大負荷を超えないことを保証するために、変化は軸の加速度から算出されます。

過負荷係数は、IPO-サイクルに対して機械軸(MD32300 $MA_MAX_AX_ACCEL)の加速度を超えることができる範囲を示します。

速度不連続変化は次の式で得られます。

速度不連続変化 = 軸加速度 * (過負荷係数-1) * 補間周期

過負荷係数は、マシンデータに保存されます。

MD32310 $MA_MAX_ACCEL_OVL_FACTOR (軸速度不連続変化の過負荷係数)

係数 1.0 は、有限速度での接線の変化が連続的な移行のみ移動できることを意味します。他のすべての移行では、速度は指令値の変更によりゼロまで低下します。 この動作は、「補間終了でのイグザクトストップ」機能に相当します。 これは、連続軌跡モードの場合には不適切なので、係数は 1.0 より大きく設定してください。

注記

スタートアップとインストールについて、角のあるブロック遷移の間に機械が振動しやすいと思われ、丸み付けを使用しない予定の場合は、係数を小さくするように注意してください。

以下のマシンデータを設定すると、ブロック遷移部は設定された過負荷係数に関係なく、G641 / G642 を使用して丸み付けされます。

MD20490 $MC_IGNORE_OVL_FACTOR_FOR_ADIS

起動と解除

過負荷係数に従って減速した連続軌跡モードは、あらゆる NC パートプログラムブロックでモーダル命令 G64 を使用して起動できます。

ブロック単位で機能するイグザクトストップを選択すると、丸み付けを中断できます (G9)。

連続軌跡モード G64 は、以下の選択により解除できます。

- モーダルイグザクトストップ G60
- 丸み付け G641、G642、G643、G644、または G645

自動的な連続軌跡モード

ブロック軌跡長が非常に短い(ゼロサイクルのブロックなど)ため、丸み付けのある連続軌跡モード G641 で丸み付けブロックを挿入できない場合は、モードが連続軌跡モード G64 に切り替わります。

5.3.3 丸み付け

機能

「丸み付け」機能では、非連続的な(角のある)ブロック遷移部で、プログラムされた輪郭(軌跡軸)に沿って、挿入ブロック(位置決めブロック)が追加されるため、得られる新しいブロック遷移部は連続的(接線の変化が連続的)になります。

同期軸

丸み付けでは、ジオメトリ軸だけでなく、すべての同期軸も考慮されます。 ただし、軌跡軸と同期軸の平行移動のために、両方の軸タイプについて連続的なブロック遷移部を同時に作成することはできません。 この場合、常に正確に移動する軌跡軸を優先するために、同期軸についてはほぼ連続的なブロック遷移部しか作成されません。

G64 の場合の丸み付け

丸み付けは、ブロック遷移部でダイナミック応答制限値を守るために、G64 の場合にブロック遷移部で許容速度を超える速度が要求されるときでも行われます(「過負荷係数による速度低減 (ページ 237)」の章の「過負荷係数」を参照)。

同期条件に対する影響

丸み付けを使用すると、間に丸み付けブロックが追加されるプログラム済みブロックが短縮されます。 プログラムされたブロック境界は消え、同期条件(動作と並列の補助機能出力、ブロック終点での停止など)の条件として使用できなくなります。

注記

「丸み付け」機能を使用するときは、挿入された丸み付けブロックの終点ではなく、丸み付け位置の前のブロック終点に同期条件を適用することを推奨します。 その場合、次のブロックは開始されず、ブロック終点での停止によって、次のブロックの輪郭をさらに手動で変更できます。

例外

以下の場合は、ブロック遷移部で、たとえば、N20 の後の N10 ブロック間で、丸み付けは**行われません**。つまり、丸み付けブロックは追加されません。

移動動作の自動的な停止の場合

考えられる原因

- N20 の移動動作の前に補助機能出力が有効になっている
- N20 に軌跡軸の移動動作が含まれていない
- N20 で、以前は位置決め軸であった軸が軌跡軸として初めて移動する
- N20 で、以前は軌跡軸であった軸が位置決め軸として初めて移動する
- ジオメトリ軸が N10 で移動するが、N20 で移動しない
- ジオメトリ軸が N20 で移動するが、N10 で移動しない
- N20 でのねじ切り G33 の起動
- BRISK と SOFT の切り替え
- 座標変換に関連する軸が軌跡移動に完全に割り当てられていない(揺動軸、位置決め軸など)。

丸み付けブロックの挿入により、パートプログラムの加工が不均衡的に遅くなる場合

考えられる原因

- プログラムまたはプログラムセクションが多数の非常に短い移動ブロックで構成されている(≈ 1 補間周期/移動ブロック。各移動ブロックには少なくとも 1 つの補間周期が必要であるため、挿入ブロックが挿入されると、加工時間はほぼ 2 倍になります)
- 減速なしの G64 (丸み付けなしの軌跡制御操作)がブロック変更に対して有効になっている
- パラメータ設定された過負荷係数(MD32310 $MA_MAX_ACCEL_OVL_FACTOR)により、軌跡速度を減少させる必要なく、プログラムされた輪郭を移動することが許可されている参照:MD20490 $MC_IGNORE_OVL_FACTOR_FOR_ADIS

軌跡パラメータによって丸み付けが禁止されている場合

考えられる原因

- G641 (軌跡条件に基づく丸み付けのある軌跡制御操作)が有効になっているが、早送りが有効(G0)でかつ ADISPOS == 0 (G0 の丸み付け距離)である
- G641 (軌跡条件に基づく丸み付けのある軌跡制御操作)が有効になっているが、早送りが有効**でなく**、かつ ADIS == 0 (軌跡機能 G1、G2、G3 ...の丸み付け距離)である
- G642 または G643 (定義した許容範囲を維持する丸み付けのある軌跡制御操作)が有効になっているが、すべての許容範囲== ゼロである

N10 または N20 に移動動作が含まれていない場合(ゼロブロック)。

通常、ゼロブロックは作成されません。例外

- シンクロナイズドアクションが有効
- プログラムジャンプ

同期条件に対する影響

間に丸み付け輪郭が追加されるプログラム済みブロックは、丸み付け中に短縮されます。 プログラムされたオリジナルのブロック境界は消え、同期条件(動作と並列の補助機能出力、ブロック終点での停止など)に使用できなくなります。

注記

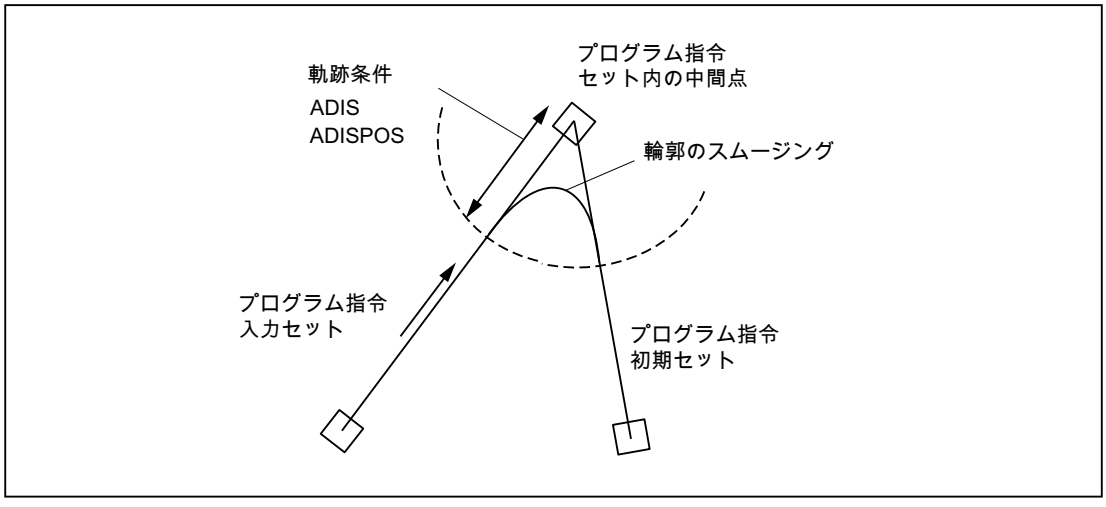
「丸み付け」機能を使用するときは、挿入された丸み付けブロックの終点ではなく、丸み付け位置の前のブロック終点に同期条件を適用することを推奨します。 その場合、次のブロックは開始されず、ブロック終点での停止によって、次のブロックの輪郭をさらに変更できます。

5.3.3.1 軌跡条件に応じた丸み付け(G641)

機能

軌跡条件に応じた丸み付けのある連続軌跡モードでは、丸み付けエリアのサイズが、軌跡条件 ADIS と ADISPOS に影響されます。

軌跡条件 ADIS と ADISPOS は、丸み付けブロックがブロックの前後で占有できる最大距離を表します。



注記

鋭角は曲率の大きい丸み付け曲線を生成するので、それに応じた速度の低下の原因になります。

注記

ADISPOS は ADIS と同様にプログラム指令されますが、特に早送りモード G0 での動作のために使用してください。

軌跡条件の範囲

- ADIS または ADISPOS をプログラム指令してください。初期設定が「ゼロ」の場合、G641 は G64 と同じように動作します。
- 関与するブロックの 1 つのみが早送り G0 の場合は、小さい方の丸み付け距離が適用されます。

- ADIS に非常に小さい値が使用される場合、コントローラは、全ての補間ブロックが、たとえ中間丸み付けブロックであっても、少なくとも 1 つの補間点を必ず含むように処理します。したがって、最高軌跡速度は ADIS /補間周期に制限されます。
- ADIS と ADISPOS に関係なく、丸み付けエリアはブロックの距離により制限されます。
  短い距離(距離 < 4* ADIS と距離 < 4 * ADISPOS)のブロックでは、丸み付け距離が縮小され、元のブロックの移動可能な部分が保持されます。残りの長さは軸軌跡により異なり、ブロックで移動される距離の約 60%になります。したがって、ADIS または ADISPOS は、移動される距離の 40%まで縮小されます。このアルゴリズムにより、輪郭の非常に小さい変更に対して、丸み付けブロックが挿入されることが防止されます。この場合、連続軌跡モード G64 への切り替えは、丸み付けブロックが再度挿入可能となるまで自動的におこなわれます。

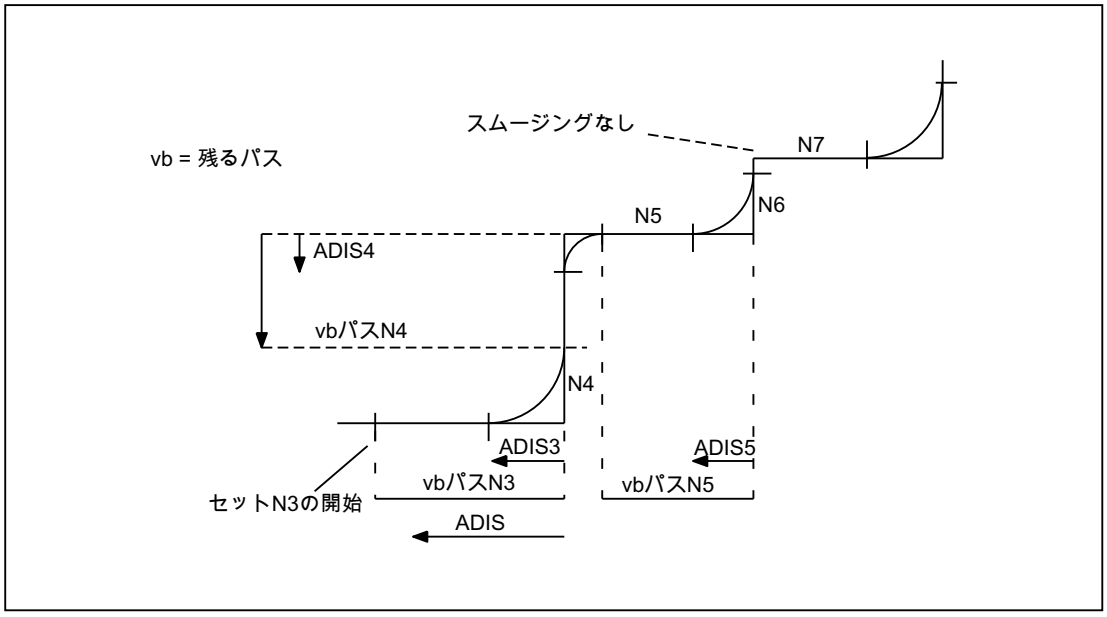


図 5-4　ADIS の制限による軌跡

起動と解除

軌跡条件に基づく丸み付けのある連続軌跡モードは、あらゆる NC パートプログラムブロックでモーダル命令 G641 を使用して起動できます。選択の前または選択時に、軌跡条件 ADIS/ADISPOS を指定してください。

ブロック単位で機能するイグザクトストップを選択すると、丸み付けを中断できます (G9)。

軌跡条件に基づく丸み付けのある連続軌跡モード(G641)は、以下を選択して解除できます。

- モーダルイグザクトストップ(G60)
- 連続軌跡モード G64、G642、G643、G644、または G645

プログラム例

プログラムコード	コメント
N1 G641 Y50 F10 ADIS = 0.5	; 軌跡条件に基づく丸み付けのある連続軌跡モード(丸み付け間隙:0.5 mm)
N2 X50	
N3 X50.7	
N4 Y50.7	
N5 Y51.4	
N6 Y51.0	
N7 X52.1	

5.3.3.2 定義した許容範囲に適合する丸み付け(G642/G643)

機能

定義した許容範囲に適合する丸み付けのある連続軌跡モードでは通常、最大許容軌跡偏差以下で丸み付けがおこなわれます。 これらの軸用の許容範囲の代わりに、最大軌跡誤差(輪郭の許容範囲)または工具の向きの最大角度誤差(向きの許容範囲)を設定できます。

起動

定義した許容範囲に適合する丸み付けのある連続軌跡モードは、あらゆる NC パートプログラムブロックでモーダル命令 G642 または G643 を使用して起動できます。

ブロック単位で機能するイグザクトストップを選択すると、丸み付けを中断できます(G9)。

定義した許容範囲に適合する丸み付けのある連続軌跡モード(G642/G643)は、以下を選択して解除できます。

- モーダルイグザクトストップ(G60)
- 連続軌跡モード G64、G641、G644、または G645

G642 - G643 の間の違い

これらの丸み付け動作に関しては、命令 G642 と G643 に以下の違いがあります。

G642	G643
G642 を使用すると、すべての軸の丸み付けをおこなう最短距離に基づいて、丸み付け軌跡が決定されます。 この値は、丸み付けブロックを生成するときに考慮されます。	G643 の場合は、各軸の丸み付け軌跡が異なる場合があります。 丸み付け移動は、軸別に、ブロック内部で処理されます(⇒ 個別の丸み付けブロックなし)。
G642 を使用すると、最小の許容範囲設定から丸み付けエリアが得られます。	大きく異なる輪郭許容範囲と工具の向きの許容範囲の指定は、G643 でのみ有効に動作することができます。

パラメータ設定

最大軌跡誤差

G642 / G643 により許容される最大軌跡誤差は、各軸に対して以下のマシンデータで設定されます。

MD33100 $MA_COMPRESS_POS_TOL

輪郭の許容範囲と向きの許容範囲

輪郭の許容範囲と向きの許容範囲は、チャネルセッティングデータに設定されます。

SD42465 $SC_SMOOTH_CONTUR_TOL (最大輪郭誤差)

SD42466 $SC_SMOOTH_ORI_TOL (工具の向きの最大角度偏差)

セッティングデータは NC プログラムで指令することができ、これにより各ブロック遷移に対して異なる指定をすることができます。

注記

セッティングデータ SD42466 $SC_SMOOTH_ORI_TOL は、有効な方向座標変換でのみ効果があります。

丸み付け動作

G642 と G643 による丸み付け動作は、以下のマシンデータで設定します。

MD20480 $MC_SMOOTHING_MODE (G64x による丸み付け動作)

一の位(**E**)は、G643 の動作を定義し、十の位(**Z**)は G642 の動作を定義します。

値 E または Z	意味
0	**すべての軸:** 最大許容軌跡誤差を維持することによる丸み付け: MD33100 $MA_COMPRESS_POS_TOL
1	**ジオメトリ軸:** 輪郭の許容範囲の維持による丸み付け SD42465 $SC_SMOOTH_CONTUR_TOL **残りの軸:** 最大許容軌跡誤差を維持することによる丸み付け: MD33100 $MA_COMPRESS_POS_TOL
2	**ジオメトリ軸:** 向きの許容範囲の維持による丸み付け SD42466 $SC_SMOOTH_ORI_TOL **残りの軸:** 最大許容軌跡誤差を維持することによる丸み付け: MD33100 $MA_COMPRESS_POS_TOL
3	**ジオメトリ軸:** 輪郭の許容範囲と向きの許容範囲の維持による丸み付け SD42465 $SC_SMOOTH_CONTUR_TOL SD42466 $SC_SMOOTH_ORI_TOL **残りの軸:** 最大許容軌跡誤差を維持することによる丸み付け: MD33100 $MA_COMPRESS_POS_TOL
4	**すべての軸:** ADIS または ADISPOS を使用してプログラム指令された丸み付けの長さを使用します(G641 の場合と同様)。 軸別の許容範囲や輪郭および向きの許容範囲の指定は、無視されます。

制限速度の波形

定義した許容範囲に適合する丸み付けの速度波形の使用は、MD20480 の百の桁で制御します。

値	意味	
< 100:	制限速度の波形は、関係する軸または軌跡について定義された加速度と加々速度の最大値に基づき、丸み付けエリア内で計算されます。 これは、丸み付けエリアでの軌跡速度の増大につながるので、関係する軸の加速につながることがあります。	
≥100:	制限速度の波形は、G641/G642 による丸み付けブロックに対しては計算されません。 代わりに、一定の速度制限が指定されます。 これにより、G641/G642 による丸み付け時に、関係する軸の丸み付けエリアへの加速した進入が防止されます。 ただし、場合によっては、この設定は丸み付けブロックの移動が遅くなりすぎる原因になる可能性があります。特に大きな丸み付けエリアでそうなります。	
	1xx:	G641 の速度波形なし
	2xx:	G642 の速度波形なし

注記

MD28530 $MC_MM_PATH_VELO_SEGMENTS (軌跡速度を制限するためのメモリ要素の数)

必要条件

有効な工具径補正と工具の向きによるプロテクションゾーンの制限

基本座標系の 3 つの基準面の 1 つに垂直ではない工具の向きに、工具径補正が適用されますが、プロテクションゾーンはその対応する平面にむかって回転しません。

G643 の場合は、以下を適用してください。

MD28530 $MC_MM_PATH_VELO_SEGMENTS > 0 (軌跡速度を制限するためのメモリ要素の数)

この条件が満たされた場合、すべての軸に対して以下を適用してください。

MD35240 $MC_ACCEL_TYPE_DRIVE = FALSE (軸に対する加速特性 DRIVE のオン/オフ)

5.3.3.3 使用可能な最大軸ダイナミック応答による丸み付け(G644)

機能

このタイプの丸み付けのある連続軌跡モードでは、軸のダイナミック応答の最大化が鍵となります。

注記

G644 による丸み付けは、以下の場合にのみ可能です。

- 関係のあるすべての軸が、監視されるブロックの両方で直線移動だけを含む。
- キネマティック座標変換が**無効**

関係する軸が多項式(プログラム指令多項式、スプライン有効、コンプレッサ有効)を含むか、またはキネマティック座標変換が有効な場合は、ブロック遷移部が G642 により丸み付けされます。

起動

使用可能な最大軸ダイナミック応答による丸み付けのある連続軌跡モードは、あらゆるNCパートプログラムブロックでモーダル命令 G644 を使用して起動できます。

ブロック単位で機能するイグザクトストップを選択すると、丸み付けを中断できます(G9)。

使用可能な最大軸ダイナミック応答による丸み付けのある連続軌跡モード(G644)は、以下を選択して解除できます。

- モーダルイグザクトストップ(G60)
- 連続軌跡モード G64、G641、G642、G643、または G645

パラメータ設定

G644 による丸み付け動作は、マシンデータで千と万の桁を使用して設定します。

MD20480 $MC_SMOOTHING_MODE (G64x による丸み付け動作)

値	意味
1,000 の位:	
0xxx:	G644 による丸み付けをおこなうと、次のマシンデータで指定した各軸の最大誤差が使用されます。 MD33100 $MA_COMPRESS_POS_TOL 軸の応答性により許容される場合は、指定した許容範囲は利用されません。
1xxx:	ADIS=...または ADISPOS=...をプログラム指令することにより、最大丸み付け軌跡を入力します(G641 と同様)。
2xxx:	下記のマシンデータを使用して、丸み付けエリアにおける各軸の最大許容周波数を入力します。 MD32440 $MA_LOOKAH_FREQUENCY (先読み用スムージング周波数) 丸み付けエリアは、丸み付け動作中に、指定された最大値を超える周波数が発生できないように定義されます。
3xxx:	コーナで速度の不連続変化のある軸は、コーナの回りを可能な最大ダイナミック応答(最高加速度と最高加々速度)で移動します。 SOFT: SOFT が有効な場合は、各軸の最高加速度**と**最高加々速度が守られます。 BRISK: BRISK が有効な場合は、各軸の最高加速度だけが守られ、最高加々速度は**守られません**。 この設定の場合、最大誤差も丸み付け距離も確認されません。 発生する誤差または丸み付け距離は、それぞれの軸の応答性の限界と現在の軌跡速度にだけによって決定されます。
4xxx:	0xxx の場合と同様に、次のマシンデータにより指定される各軸の最大誤差が使用されます。 MD33100 $MA_COMPRESS_POS_TOL 0xxx とは違って、可能なら、指定された許容範囲も利用されます。 したがって、この軸は、自身の可能な最大応答性には到達しません。
5xxx:	1xxx の場合と同様に、可能な最大の丸み付け軌跡が、それぞれ ADIS=...または ADISPOS=のプログラム指令により指定されます。 1xxx とは違い、可能なら、指定された丸み付け軌跡も利用されます。 したがって、関連する軸は、自身の可能な最大応答性には到達しません。

値	意味
10,000 の位	
0xxxx	丸み付けエリアの軸の速度波形は、BRISK の加々速度が無制限で、SOFT の加々速度は制限して決定されます。
1xxxx	丸み付けエリアの軸の速度波形は、BRISK や SOFT が動作中かどうかに関係なく、常に加々速度を制限して決定されます。

最大軸誤差(MD33100 $MA_COMPRESS_POS_TOL)または最大丸み付け距離(ADIS / ADISPOS)を指定したときに、関係のある軸の応答性により可能ならば、使用可能な丸み付け軌跡は通常使用されません。 このため、丸み付け軌跡の長さは、有効な軌跡送り速度に依存します。 軌跡速度を小さくした場合、プログラム指令輪郭による誤差は小さくなります。 ただし、これらの場合に指定された最大軸誤差または指定された丸み付け距離が、できれば利用されるように設定することができます。 この場合、誤差はプログラム指令された軌跡送り速度には無関係の、プログラム指令輪郭に依存します。

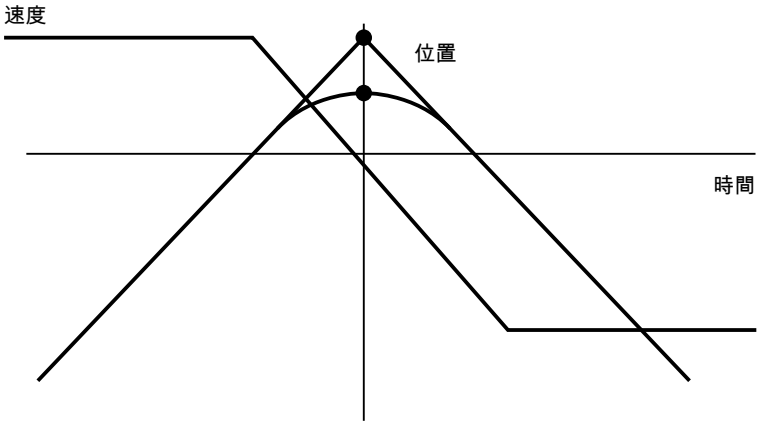
注記

前述のものとは別に、さらに、
それは、丸み付け距離が、元の関係するブロックの長さの半分を超えないことです。

加々速度制限

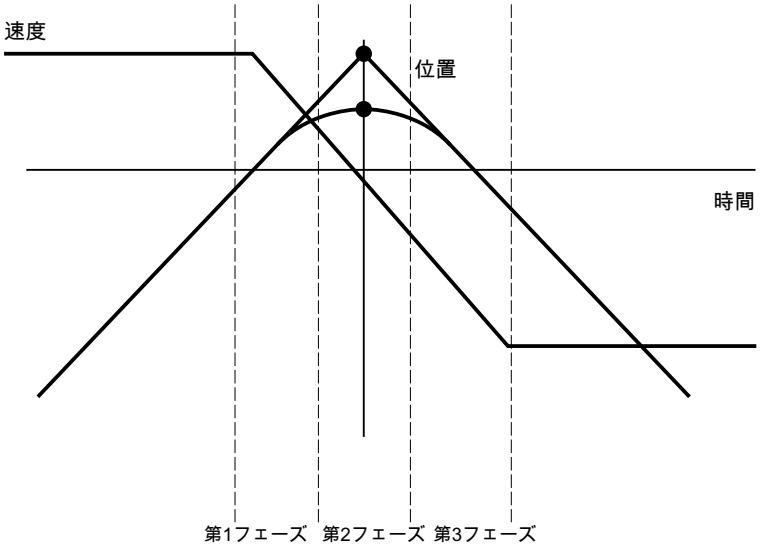
各軸での速度の不連続変化のスムージングとそれによる丸み付け軌跡の形状は、補間をおこなうときに加々速度制限の有無により異なります。

加々速度制限がない場合、丸み付けエリア全体で、各軸の加速度がその最大値に到達します。



加々速度制限がある場合、丸み付けエリア内で、各軸の加々速度が、その最大値に制限されます。 したがって、丸み付け動作は、一般に 3 つのフェーズで構成されます。

- **フェーズ 1**
  フェーズ 1 の間に、各軸は自身の最高加速度に増加します。 加々速度は一定であり、それぞれの軸で可能な最高加々速度に等しくなります。
- **フェーズ 2**
  フェーズ 2 の間は、最高許容加速度が適用されます。
- **フェーズ 3**
  最後のフェーズであるフェーズ 3 の間に、各軸の加速度が最高許容加々速度で小さくなってゼロに戻ります。



5.3.3.4 接線の変化が連続的なブロック遷移部の丸み付け(G645)

機能

丸み付けのある連続軌跡モードでは、オリジナルの輪郭の曲率が少なくとも 1 つの軸で不連続変化を示す場合は、丸み付けブロックも、接線の変化が連続的なブロック遷移部でのみ生成されます。

丸み付け移動はここで、関与するすべての軸の加速が滑らかなまま(不連続変化なし)であり、パラメータ設定した、オリジナルの輪郭からの最大誤差 (MD33120 $MA_PATH_TRANS_POS_TOL)を超えないように定義されます。

角度のある、接線の変化が不連続なブロック遷移の場合、丸み付け動作は G642 の場合と同様です(「定義した許容範囲に適合する丸み付け(G642/G643) (ページ 245)」の章を参照してください)。

起動と解除

接線の変化が連続的なブロック遷移部の丸み付けのある連続軌跡モードは、あらゆる NC パートプログラムブロックでモーダル命令 G645 を使用して起動できます。

ブロック単位で機能するイグザクトストップを選択すると、丸み付けを中断できます(G9)。

接線の変化が連続的なブロック遷移部の丸み付けのある連続軌跡モード(G645)は、以下を選択して解除できます。

- モーダルイグザクトストップ(G60)
- 連続軌跡モード G64、G641、G642、G643、または G644

G642 と G645 の比較

G642 による丸み付けをおこなうと、丸み付けがおこなわれるブロック遷移部は、コーナを形成するもののみです。つまり、少なくとも 1 つの軸の速度が不連続変化を示します。 ただし、ブロック遷移で接線の変化が連続的だが、曲率に不連続変化がある場合は、G642 で丸み付けブロックが挿入されません。 このブロック遷移が有限の速度で移動する場合は、ある程度の加速の不連続変化が軸に発生しますが、(加々速度一定が有効な場合は)これが、パラメータで設定した制限(MD32432 $MA_PATH_TRANS_JERK_LIM)を超えないようにしてください。 制限のレベルによっては、ブロック遷移の軌跡速度が大幅に低下する結果となる場合があります。 この制約は、G645 を使用すると回避できます。これは、丸み付け移動はここで、加速の不連続変化が発生しないように定義されているためです。

パラメータ設定

次のマシンデータは、G645 による丸み付け時の、各軸の最大許容軌跡誤差を指定します。

MD33120 $MA_PATH_TRANS_POS_TOL

この値は、可変加速による、接線の変化が連続的なブロック遷移部にのみ有効です。 角度のある、接線の変化が不連続なブロック遷移を丸み付けすると、MD33100 $MA_COMPRESS_POS_TOL による許容範囲が有効(G642 と同様)になります。

下記も参照

自由形式面モード:基本機能 (ページ 281)

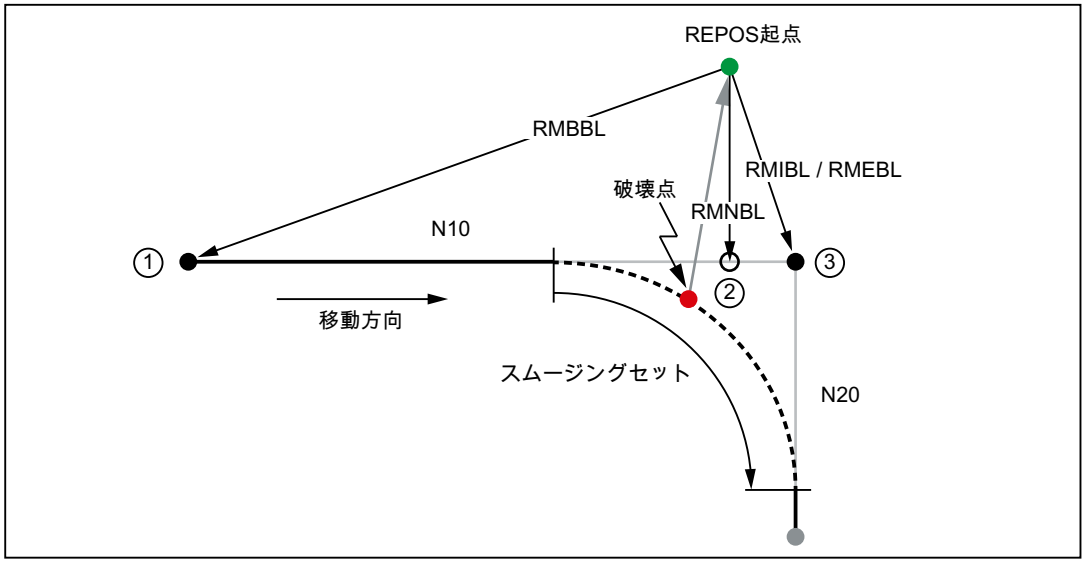
5.3.3.5 丸み付けと再位置決め(REPOS)

丸み付け輪郭の領域での加工が中断された場合、REPOS 操作を使用して丸み付け輪郭上に直接再位置決めすることは**できません**。 その場合、位置決めは、**プログラムされた**輪郭上でのみ行うことができます。

例

プログラム指令: 丸み付け G641 がプログラム指令された 2 つの移動ブロック N10 と N20 があります。

この丸み付け領域で移動動作が中断されます。 その後、軸が、たとえば、手動で REPOS 開始点に移動されます。 選択されている REPOS モードに応じて、輪郭上での再位置決めは点 ①、② または ③ で行われます。



RMBBL 中断された移動ブロックの始点への再位置決め
RMIBL 中断位置への再位置決め
RMEBL 中断された移動ブロックの終点への再位置決め
RMNBL 次の輪郭点への再位置決め
① N10 ブロックの始点
② 次の輪郭点の REPOS 開始点へ
③ N10 ブロックの終点/N20 ブロックの始点

5.3.4 先読み

5.3.4.1 標準機能

機能

先読みは、連続軌跡モード(G64、G64x)で動作する機能で、現在のブロック内およびそれ以降の複数の NC パートプログラムブロックの予測可能な速度制御を決定します。

注記

先読み機能は、軌跡軸にのみ使用でき、主軸と位置決め軸には**使用できません**。

パートプログラムに、軌跡の非常に短い連続するブロックが含まれる場合は、先読みを使用せずに到達する速度がブロック毎に 1 つのみであるため、加速制限を維持したまま、ブロック終点で軸を減速できます。これは、プログラム指令速度には決して到達しないということです。ただし、先読みを使用すると、複数のブロックにわたり、接線がほぼ連続的に変化するブロック遷移部による加速と減速区間を実装でき、その結果、短い距離でより高い送り速度が実現されます。

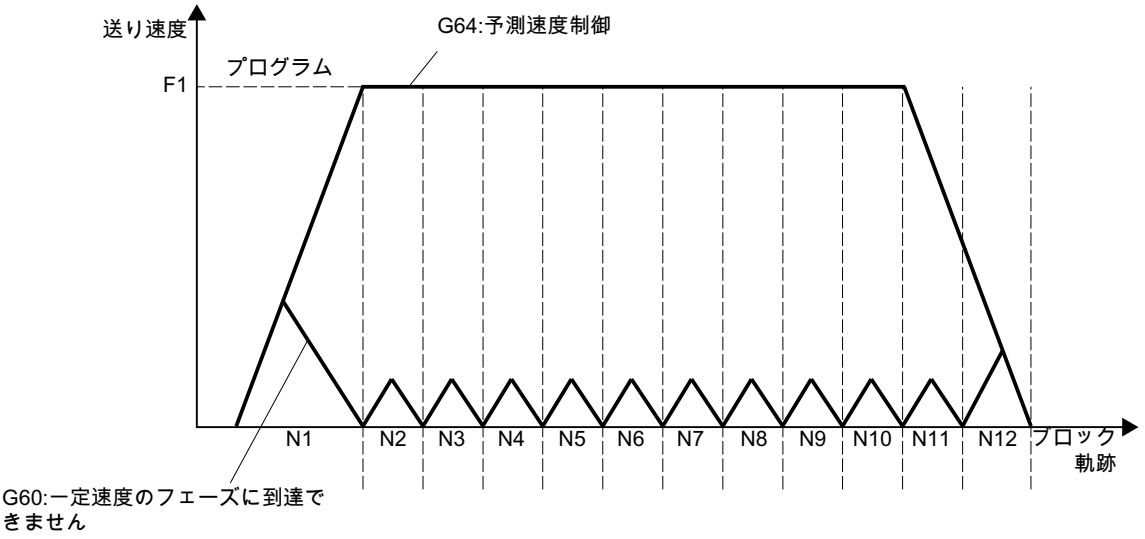


図 5-5　イグザクトストップ G60、または先読みを使用した連続軌跡モード G64 による短距離での速度制御

先読みを使用すると、加速度と速度制限の違反が防止されるような速度制限値までの減速が可能です。

先読みは、以下のように計画可能な速度制限を考慮します。

- ブロック終点でのイグザクトストップ
- ブロックでの速度制限
- ブロックでの加速制限
- ブロック遷移での速度制限
- ブロック遷移でのブロック切り替えによる同期

運転モード

先読みは、速度制限のブロック毎の分析を実行し、この情報に基づいて必要な減速カーブの波形を指定します。先読みは、ブロック長、減速能力、および許容可能な軌跡速度に対して自動的に補正されます。

安全上の理由のため、最後の準備済みブロックのブロック終点での速度は、最初はゼロであると仮定してください。これは、次のブロックが非常に小さいか、イグザクトストップのブロックである可能性があり、ブロック終点で軸がすでに停止している必要があるためです。

設定速度が高速で軌跡が非常に短い一連のブロックを使用すると、要求された設定速度を実現するために、先読み機能により現在計算された速度値に応じて、各ブロックで速度を増大させることができます。この後は、先読み機能により考慮される最後のブロックの終点での速度をゼロにできるように、低下させることができます。この結果、鋸歯状の速度波形(次の図を参照してください)が生じ、これは設定された速度を下げるか、または先読み機能により考慮されるブロックの数を増加させることにより回避できます。

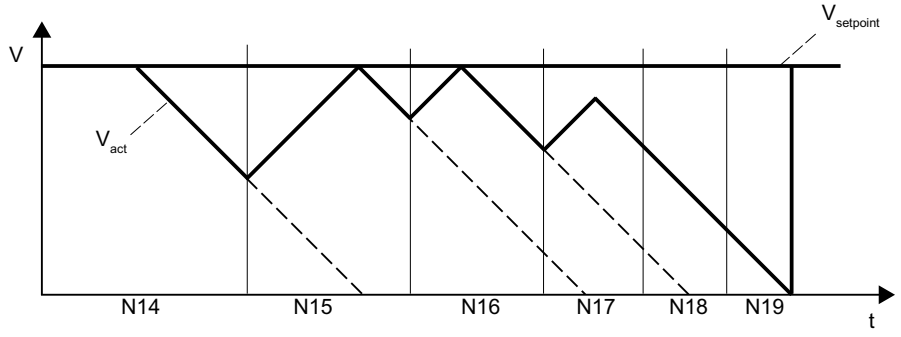


図 5-6　モーダル速度制御の例(先読み機能により考慮されるブロックの数 = 2)

起動と解除

先読みは、連続軌跡モード G64、G641、G642、G643、G644、または G645 を選択して起動します。

ノンモーダル単位で機能するイグザクトストップを選択すると、丸み付けを中断できます(G09)。

先読みは、モーダルイグザクトストップ(G60)を選択して解除します。

パラメータ設定

ブロックの数

連続軌跡モードで確実な軸移動を実現するには、複数のブロックで送り速度を補正する必要があります。先読み機能により考慮されるブロックの数は自動的に計算され、必要に応じてマシンデータにより制限されます。初期設定は 1 であり、これは、先読みでは速度制御のために次のブロックだけが考慮されるということです。

先読みは、短いブロックの場合に特に重要であるため(減速軌跡に関連して)、先読み減速には、必要なブロックの数が重要となります。これは、軌跡長が最高速度から停止状態に減速するために必要な減速軌跡と等しいと見なすのに十分です。

低い軸加速度($a = 1\ m/s^2$)と高い送り速度($v_{path} = 10\ m/min$)を持つ機械の場合は、以下の n 個の$_{先読み}$ブロックが、達成可能なブロックサイクルタイム TB = 10 ms を持つコントローラに割り当てられます。

$n_{先読み}$ = 減速軌跡/ブロック長 = $(v_{path}^2 / (2a)) / (v_{path} * TB) = 9$

これらの条件を考慮して、10 ブロックを超える送り速度は補正することをお勧めします。先読み機能の予測のために入力されたブロックの数により、先読みアルゴリズムとメモリの必要条件が変化することはありません。

プログラムでは、加工速度を最高速度より小さい値に設定することが多いため、必要なブロックより多いブロックが予測され、プロセッサが必要以上に過負荷になります。このため、必要なブロックの個数は、次の乗算で算出した速度で求められます。

- プログラム指令速度 * MD12100 $MN_OVR_FACTOR_LIMIT_BIN
  (**バイナリ**コードの送り速度オーバライドスイッチ使用時)
- プログラム指令速度 * MD12030 $MN_OVR_FACTOR_FEEDRATE[**30**]
  (**グレー**コードの送り速度オーバライドスイッチ使用時)

MD12100 の値または MD12030 の 31 番目のオーバライド値は、軌跡送り速度をオーバーシュートした場合に速度制御が提供するダイナミック応答予約を定義します。

注記

MD12030 の 31 番目のオーバライド値は、実際に使用する最大オーバライド係数に対応している必要があります。

注記

先読み機能により考慮されるブロックの数は、IPO バッファで可能な NC ブロックの数により制限されます。

速度波形

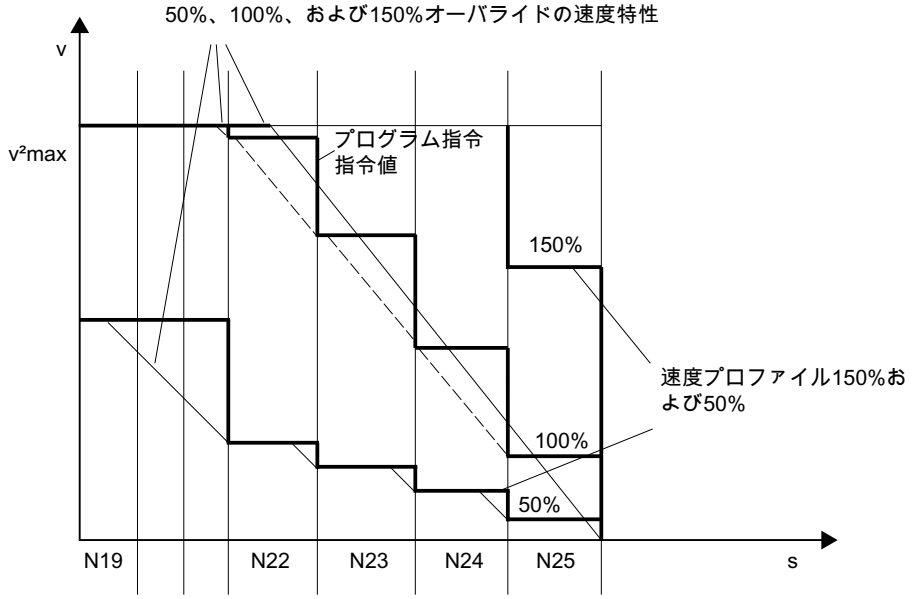
固定の計画可能な速度制限に加えて、先読みではプログラム指令速度も考慮できます。これにより、現在のブロックを越えて先読みを適用することにより、より低い速度を実現できるようになります。

- **次のブロックの速度の決定**
  1 つのありうる速度波形には、次のブロック速度の決定が含まれます。
  現在と次の NC ブロックの情報を使用して、速度波形が計算され、同様にそれから現在のオーバライドに必要な速度低下が算出されます。
  速度波形の計算された最大値は、最大軌跡速度により制限されます。
  この機能を使用すると、次のブロックでより小さい速度を実現できるようにオーバライドを考慮して、現在のブロックで減速を開始することができます。減速にかかる時間が現在のブロックの移動時間より長い場合、速度は次のブロックでさらに減速されます。速度制御は、いつも次のブロックに対して考慮されます。
  この機能は、以下のマシンデータで起動します。
  MD20400 $MC_LOOKAH_USE_VELO_NEXT_BLOCK

値	意味
TRUE	機能が動作中です
FALSE	機能が**動作していません**

- **オーバライド点の定義**
  たとえば、非常に大きいオーバライド値(200%など)を使用中、または周速一定制御 G96/G961 が動作中のために、次のブロックの速度の速度波形が不十分で、その結果として、次のブロックで速度をさらに下げる必要がある場合は、先読みにより、複数の NC ブロックにわたってプログラム指令速度を下げる方法が提供されます。
  オーバライド点を定義すると、先読みが各値に対して制限速度波形を計算します。現在のオーバライドに必要な速度低下は、これらの波形から算出されます。
  速度波形の計算された最大値は、最大軌跡速度により制限されます。
  上限には、以下のマシンデータに設定された最大値により到達される速度範囲を含めてください。
  MD12030 $MN_OVR_FACTOR_FEEDRATE [n] (軌跡送り速度のオーバライドスイッチの倍率)
  これは、下記のマシンデータの値を介しても到達できます。
  MD12100 $MN_OVR_FACTOR_LIMIT_BIN (バイナリコードのオーバライドスイッチの制限)
  このように、ブロック内でプログラム指令された速度低下は、回避することができます。
  100%のオーバライドで、ブロックの境界を越えて速度を下げる必要がある場合は、より低いオーバライド範囲にもオーバライド点を設定してください。
  チャネル毎に使用されるオーバライド点の数は、下記のマシンデータに指定されます。
  MD20430 $MC_LOOKAH_NUM_OVR_POINTS (先読みのためのオーバライドスイッチ点の数)
  関連した点は、下記のマシンデータに格納されます。
  MD20440 $MC_LOOKAH_OVR_POINTS (先読み用のオーバライドスイッチ点)
  例:
  制限速度特性の説明:
  – オーバライド= 50%、100%、または 150%
  – 先読みブロック数= 4
  – MD20430 $MC_LOOKAH_NUM_OVR_POINTS = 2
  – MD20440 $MC_LOOKAH_OVR_POINTS = 1.5, 0.5

– MD20400 $MC_LOOKAH_USE_VELO_NEXT_BLOCK = 1



両方の処理(次のブロックの速度の決定とオーバライド点の決定)を組み合わせて使用すると、速度波形を計算できるため、これを一般的にお勧めします。その理由は、これらの機能に対する事前設定のマシンデータではすでに、オーバライドに依存した速度制限の最大範囲が考慮されていることです。

注記

これらの処理のどちらも動作中ではない場合、現在のブロックでは常に指令値速度が適用されます。

注記

計画可能速度制限は、オーバライド別の速度制限を制限します。

ブロックサイクルに問題のある緩和係数

ブロックサイクル問題は、処理される NC ブロックの移動距離が非常に短いので、ブロック処理に十分な時間を提供するために、先読み機能が加工速度を低下させなければならない場合に発生します。この状況では、軌跡動作の連続的な減速と加速が発生する可能性があります。

このタイプの速度変動は、以下の緩和係数を指定して軽減できます。

MD20450 $MC_LOOKAH_RELIEVE_BLOCK_CYCLE (ブロックサイクルタイムの緩和係数)

必要条件

軸別の送り停止/軸停止

軸別の送り停止と軸別の軸無効は、先読みにより無視されます。

軸が補間される一方で、軸別の送り停止または送り無効により停止しなければならない場合、先読みは当該のブロックの前では軌跡移動を停止せず、そのブロック自身の**中で**減速させます。

この動作が望ましくない場合、**軸**別の送り停止を、PLC を介して**チャネル**別の送り停止に転送し、軌跡を直ちに停止することができます(「機能 (ページ 107)」の章も参照してください)。

5.3.4.2 自由形式面モード: 拡張機能

機能

「自由曲面モード:拡張機能」は、先読み標準機能を拡張したもので、これを使用して、自由曲面加工中の軌跡速度波形を計算します。(「自由形式面モード:基本機能 (ページ 281)」の章も参照してください)。

この機能は、連続軌跡モードを次のように最適化します。

- 加速波形と減速波形の均整
- 加々速度または加速制限が変動する場合でも均一な加速処理
- 指定したダイナミック応答制限で起動できる、または起動できない可能性の程度に関係しない、目標速度波形の均一な加速処理
- より低い指令値速度への先読み制動

均一性と、ダイナミック応答制限への適合により、部品で指令値速度波形が滑らかになり、均質な速度波形となることが保証されます。これにより、面品質の追従誤差の影響が最小となります。

長所

- ワーク面の均一性の向上
- 機械負荷の低下

用途

「自由曲面モード:拡張機能」は、主に自由曲面から成るワークの加工に使用します。

注記

標準の加工目的で使用すると、良い結果が得られないため、標準先読み機能は、これらの場合に使用してください。

起動

この機能は、以下の場合にのみ効果的です。

- AUTOMATIC の場合
- 「加々速度制限を設定した加速(SOFT)」モードの場合

パラメータ設定

作業メモリ

「自由形式面モード:拡張機能」のメモリは、以下のマシンデータで設定します。

MD28533 $MC_MM_LOOKAH_FFORM_UNITS = <値>

必要なメモリは、パートプログラム、ブロック長、軸ダイナミック応答のほか、有効なキネマティック座標変換によっても異なります。

自由曲面の加工のガイドラインとして、次の設定が適用されます。MD28533 = 18

注記

保管に関する追加の必要条件により、MD28533 は、自由曲面が加工されるチャネルにのみ設定してください。

IPO バッファの NC ブロック数

通常の場合は、「自由曲面モード:拡張機能」の使用時に、次のように、補間バッファ内の NC ブロック数の設定を大幅に増やすことをお勧めします。

MD28060 $MC_MM_IPO_BUFFER_SIZE > 100

ブロックメモリ容量が小さすぎる場合は、そのために、軌跡速度波形の均一性が低下する場合があります。

起動と解除

この機能は、以下を使用して、ダイナミック応答モードのたびに独立してオンとオフを切り替えることができます(「軌跡補間のダイナミック応答モード (ページ 279)」の章を参照してください)。

MD20443 $MC_LOOKAH_FFORM[<n>]= <値>

インデックス<n>	ダイナミック応答モード	<値>	自由曲面モード:拡張機能
0	標準ダイナミック応答の設定(DYNNORM)	0	オフ
		1	オン
1	位置決めモード、タッピング(DYNPOS)	0	オフ
		1	オン
2	荒削り(DYNROUGH)	0	オフ
		1	オン
3	仕上げ(DYNSEMIFIN)	0	オフ
		1	オン
4	滑らかな仕上げ(DYNFINISH)	0	オフ
		1	オン

「自由曲面モード:拡張機能」は通常、「自由曲面モード:基本機能」も動作中の場合にのみ、動作します。したがって、MD20443 $MC_LOOKAH_FFORM[n>]の設定内容は、MD20606 $MC_PREPDYN_SMOOTHING_ON[n>]の設定内容に対応している必要があります。

標準先読み機能は、「自由曲面モード:拡張機能」がオフになっているダイナミック応答モードで動作します。

プログラミング

一般的に、「自由曲面モード:拡張機能」は、パートプログラムでのダイナミック応答モードの変更の結果として有効になります。

例

以下のパラメータが設定されていると仮定します。

MD20443 $MC_LOOKAH_FFORM[0] = 0

MD20443 $MC_LOOKAH_FFORM[1] = 0

MD20443 $MC_LOOKAH_FFORM[2] = 1

MD20443 $MC_LOOKAH_FFORM[3] = 1

MD20443 $MC_LOOKAH_FFORM[4] = 1

パートプログラムでのダイナミック応答モードの変更:

プログラムコード	コメント
N10 DYNPOS	; DYNPOS ダイナミック応答モードを有効にします。 標準先読み機能が DYNPOS ダイナミック応答モードで有効です。
...	
N100 G17 G54 F10000	
N101 DYNFINISH	; DYNFINISH ダイナミック応答モードを有効にします。 「自由曲面モード:拡張機能」が DYNFINISH ダイナミック応答モードで動作中です。
N102 SOFT G642	
N103 X-0.274 Y149.679 Z100.000 G0	
N104 COMPCAD	
...	
N1009 Z4.994 G01	
N10010 X.520 Y149.679 Z5.000	
N10011 X10.841 Y149.679 Z5.000	
N10012 X11.635 Y149.679 Z5.010	
N10013 X12.032 Y149.679 Z5.031	
M30	

注記

標準先読み機能から「自由曲面モード:拡張機能」へ、またはその逆への切り替え時には、連続軌跡モードが補間器停止により中断されます。

必要条件

機能の自動切り替え

以下の機能を使用すると、自動的に標準先読み機能に切り替わります。

- ねじ切り/タッピング(G33、G34、G35、G331、G332、G63)
- 軸間連動機能軌跡
- パンチング、ニブリング
- 直交 PTP 移動

「自由曲面モード:拡張機能」は、その後、再度自動的に切り替わります。

G グループ 15 の命令の使用(送り速度のタイプ)

以下の送り速度のタイプを、「自由曲面モード:拡張機能」機能と組み合わせることは推奨されません。

- 毎回転送り速度(G95、G96、G97 など)
- インバースタイム送り(G93)

FLIN の使用

FLIN 送り速度波形は、「自由曲面:拡張機能」機能と組み合わせることは推奨されません。

送り速度オーバライドの影響

送り速度オーバライド(機械操作パネル、$AC_OVR などによる)で、標準先読み機能での移動時間を大幅に延長できます。

早送り移動(G0)との連携

自由曲面の加工中に散在する G0 ブロックは、先読み機能の切り替え(「自由曲面モード:拡張機能」から標準先読み機能へ、またはその逆)をおこないません。つまり、標準ダイナミック応答設定(DYNNORM)が G0 であっても、DYNNORM に事前設定された標準先読み機能(→ MD20443 $MC_LOOKAH_FFORM[0])は、自動的には有効になりません。現在動作中の先読み機能を保持すると、特に G0 と多項式ブロックが平滑化され、丸み付けにより接続されるため、より均質な速度波形を得られます。

5.4 ダイナミック補正

5.4.1 軌跡速度のスムージング

はじめに

速度制御機能は、指定された軸のダイナミック応答を利用します。 プログラム指令送り速度で処理できない場合、軌跡速度はパラメータ設定された軸の制限値と軌跡の制限値(速度、加速度、加々速度)になります。 これは、軌跡に関して減速と加速が繰り返される原因になる可能性があります。

高速の軌跡速度での加工処理中に短い加速が実行され、ほとんどすぐ後に減速が続く場合、加工時間の短縮はごくわずかです。 ただし、この種の加速には望ましくない影響があります。たとえば、機械共振をもたらす場合です。

金型加工のいくつかの用途、特に高速の切削の場合は、一定の軌跡速度を実現することが望まれます。 したがって、これらの場合、より滑らかな工具軌跡速度を優先して一時的な加速処理を犠牲にすることが適当である可能性があります。

機能

「軌跡速度のスムージング」機能が動作中の場合は、より滑らかな軌跡速度制御を得るために、最大許容生産性損失を決定するスムージング係数が有効になります。 この係数より、プログラム実行時間の短縮効果が低い加速処理は、おこなわれません。 関与する軸の設定可能な限界周波数を上回る周波数の加速度処理のみが考慮されます。

効果:

- 連続的で一時的な減速と加速処理により発生する可能性のある機械共振 (IPO サイクルより少ない範囲で)の回避。
- プログラムの実行時間を大きく短縮することのない加速により、常に変化する切削速度の回避。

注記

軌跡速度のスムージングは、輪郭誤差にはつながりません。
一定の軌跡速度での輪郭の曲率による軸速度の変化が発生し続け、この機能では削減されません。
新しい送り速度の入力による軌跡速度の変化も、変更されません。 これにより、サブプログラムのプログラマの責任はそのままになります。

必要条件

- 軌跡速度のスムージングは、SOFT と BRISK を含む複数のブロックにわたる先読みをおこなう連続軌跡モードでのみ有効です。 スムージングは、G0 で有効にすることは**できません**。
- コントローラの制御周期は、十分な数のブロックが事前処理で準備され、加速処理が解析可能になるように設定してください。

起動と解除

「軌跡速度のスムージング」機能は、以下のマシンデータで起動/解除します。

MD20460 $MC_LOOKAH_SMOOTH_FACTOR (先読み用スムージング係数)

値	意味
0.0	軌跡速度のスムージングが**動作していません**(初期設定)
> 0	軌跡速度のスムージングが動作中です

MD 設定の変更は、NEW CONF でのみ有効にできます。

パラメータ設定

スムージング係数

スムージング係数は、以下のチャネルマシンデータで設定します。

MD20460 $MC_LOOKAH_SMOOTH_FACTOR (先読み用スムージング係数)

このパーセンテージの値は、加速/減速のない処理ステップが、加速/減速実行時の対応するステップに対して、どれくらい長いかを定義します。

これは、最初のアプローチ動作を除くパートプログラム内のすべての加速がならされた場合の、「最悪の場合」の値です。 実際の拡張は常に、より小さくなり、どの加速によっても基準が満たされない場合は、0 になる可能性もあります。 加工時間の大きな増加なしで、50%と 100%の間の値を入力することもできます。

プログラム指令送り速度の考慮

軌跡速度のスムージングは、プログラム指令送り速度を考慮して、または考慮しないで実行できます。 これは、以下のマシンデータで選択します。

MD20462 $MC_LOOKAH_SMOOTH_WITH_FEED (プログラム指令送り速度による軌跡スムージング)

値	意味
0	プログラム指令送り速度は、**考慮されません**。
1	プログラム指令送り速度は考慮されます(初期設定)。

プログラム指令送り速度を考慮すると、指定したスムージング係数(MD20460 を参照してください)が、オーバライドが 100%のときに維持されやすくなります。

軸別の限界周波数

軸別の限界周波数は、以下のマシンデータで定義されます。

MD32440 $MA_LOOKAH_FREQUENCY (先読み用スムージング周波数)

高い周波数で実行する加速と減速処理は、以下のマシンデータのパラメータ設定に応じてならされます。そうでない場合は、応答性が低下します。

MD20460 $MC_LOOKAH_SMOOTH_FACTOR (先読み用スムージング係数)

MD20465 $MC_ADAPT_PATH_DYNAMIC (ダイナミック軌跡応答の補正)

MD20465 について詳しくは、「ダイナミック軌跡応答の補正 (ページ 269)」の章を参照してください。

注記

軸の機械システムで振動が発生しており、対応する周波数がわかっている場合は、この周波数より小さい値に MD32440 を設定してください。
必要な共振周波数は、内蔵された計測機能を使用して計算できます。

運転モード

MD32440 の最小値は、その軌跡に含まれる軸に基づき、f_{Path} として計算されます。 このスムージングについては、その動作の開始速度と終了速度間の時間が下記の式で与えられる時間内に入る加速処理だけが考慮されます。

$t = t_2 - t_1 = 2 / f_{path}$

これらの加速処理は、延長される処理時間が、スムージング係数(MD20460)を超えて指定した制限を超えない場合は省略されます。

例

以下のパラメータが設定されていると仮定します。

MD20460 $MC_LOOKAH_SMOOTH_FACTOR = 10%

MD32440 $MA_LOOKAH_FREQUENCY[AX1] = 20 Hz

MD32440 $MA_LOOKAH_FREQUENCY[AX2] = 20 Hz

MD32440 $MA_LOOKAH_FREQUENCY[AX3] = 10 Hz

軌跡には X = AX1、Y = AX2、Z = AX3 の 3 つの軸が含まれます。

これらの 3 つの軸の MD32440 の最小値は 10 Hz です。 これは、$t_2 \sim t_1 = 2/10$ Hz = 200 ms の期間内に完了する加速が検査されることを意味します。 時間 t_2 は、速度 v_1 で開始した加速処理の後に、速度 v_1 に再度到達するまでの所要時間です。 実行時間の増加も、この範囲内でのみ考慮されます。

時間 $t_2 - t_1$ が 200 ms を超えているか、追加のプログラム実行時間 $t_3 - t_2$ が $t_2 - t_1$ の 10% (= MD20460)を上回る場合は、以下の時間特性が適用されます。

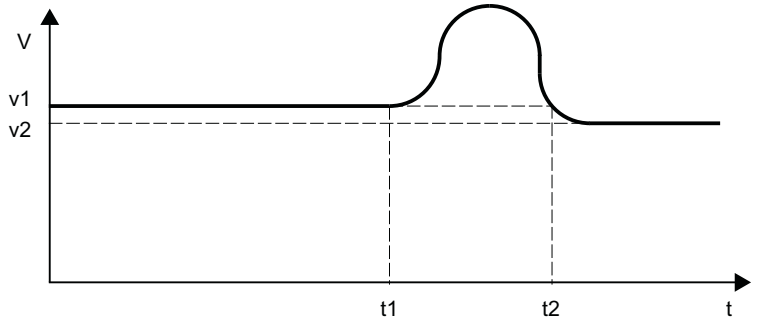


図 5-7　時間調整軌跡速度の特性(スムージングなし)

ただし、時間 $t_2 - t_1$ が 200 ms に満たないか、追加のプログラム実行時間 $t_3 - t_2$ が $t_2 - t_1$ の 10%以下である場合は、以下の時間特性が適用されます。

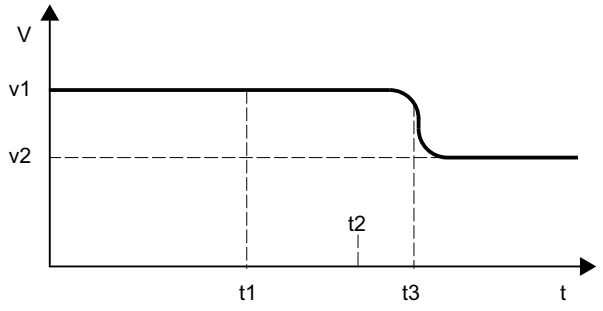


図 5-8　ならされた軌跡速度の特性

5.4.2　ダイナミック軌跡応答の補正

機能

加工中の非常にダイナミックな加速と減速処理は、機構部の共振発生の原因となり、結果としてワークの加工面品質を低下させる可能性があります。

加速と減速処理のダイナミック応答は、「ダイナミック軌跡応答の補正」機能を使用して、機械の状態に合わせて補正できます。

注記

「ダイナミック軌跡応答の補正」機能は、得られる軌跡のみに関係しており、軌跡に含まれる個々の軸の減速と加速処理には無関係です。 このため、一定の軌跡速度波形の場合でも、不連続な輪郭形状またはキネマティック座標変換によって、機械振動の励起につながるような、問題となる軸の減速と加速処理が生じる可能性があります。

有効性

「ダイナミック軌跡応答の補正」機能は、次の軌跡移動中にのみ有効です。

- 連続軌跡モード(G64, G64x)
  連続軌跡モードでは、ダイナミック応答調整の最適な効果は、有効な 100%オーバライドの場合に実現されます。 この値からの大きく外れた場合または軌跡軸が減速する原因になる機能(PLC への補助機能出力など)の場合、希望する作用がかなり小さくなります。
- イグザクトストップ(G60)

また、「ダイナミック軌跡応答の補正」機能は、次の軌跡移動中には**動作しません**。

- プログラム指令早送り(G0)
- オーバライド値の変更
- 移動中の停止要求(NC ストップ、NC リセットなど)
- 「速度に依存する軌跡加速度」機能(DRIVE)が動作中の場合

起動と解除

この機能は、以下のマシンデータで起動/解除します。

MD20465 $MC_ADAPT_PATH_DYNAMIC (ダイナミック軌跡応答の補正)

値	意味
= 1.0	ダイナミック補正が**動作していません**(初期設定)
> 1.0	ダイナミック補正が動作中です

起動すると、**「軌跡速度のスムージング」機能も常に、** 内部で連続軌跡モードで起動します(「軌跡速度のスムージング (ページ 265)」の章を参照してください)。

スムージング係数(MD20460 $MC_LOOKAH_SMOOTH_FACTOR)を 0%に設定(= 機能が解除されます; 初期設定!)した場合は、その代わりに 100%のスムージング係数を使用します。 0%以外のスムージング係数の場合は、設定した値が有効となります。

パラメータ設定

ダイナミック軌跡応答の補正係数

ダイナミック軌跡応答の補正係数を介して、軌跡速度の一時的な変更が、より小さいダイナミック応答制限値により実行されます。

補正係数は、チャネル別に設定します。

- 以下を使用した、加々速度制限なしの加速を伴う移動動作(**BRISK**)の場合:
  MD20465 $MC_ADAPT_PATH_DYNAMIC[**0**]
  →補正係数は、加速度に作用します。
- 以下を使用した、加々速度制限付きの加速を伴う移動動作(**SOFT**)の場合:
  MD20465 $MC_ADAPT_PATH_DYNAMIC[**1**]
  →補正係数は、加々速度に作用します。

軸別の限界周波数

ダイナミック応答制限は、特定の限界周波数より大きく、機械共振励起の原因になる機械振動を起こす減速と加速処理の間だけで有効にしてください。

ダイナミック応答制限が有効になる、この限界周波数は、以下のマシンデータで軸別に指定します。

MD32440 $MA_LOOKAH_FREQUENCY (先読み用スムージング周波数)

詳しくは、「軌跡速度のスムージング (ページ 265)」の章を参照してください。

運転モード

処理中、および軌跡上の関与するすべての軸を使用して、コントローラは周期的に、すべての限界周波数の最小値を、ダイナミック応答を補正する限界周波数(**f**)として決定し、関連する時間枠(t_{adapt})を以下の式で計算します。

$t_{adapt} = 1 / f$

関連する時間枠 t_{adapt} のサイズで、その他の動作が決まります。

1. 速度の変更に必要な時間が、t_{adapt} より小さいとき
   加速度は、1 より大きく、マシンデータに書き込まれた値以下である係数により低減されます。
   MD20465 ADAPT_PATH_DYNAMIC (ダイナミック軌跡の補正)
   加速度の縮小は、速度の変更にかかる時間を増加させます。
   以下の場合は別です。
   - 加速度は、t_{adapt} [s]に対して処理が継続するような MD20465 より小さい値により低減されます。 許容される低減は、十分に利用される必要はありません。
   - 加速時間は、MD20465 に書き込まれた値により低減されます。 加速度てい減にかかわらず、処理の継続は t_{adapt} より短くなります。 許容される低減は、十分に利用されました。
2. 速度の変更に必要な時間が、t_{adapt} より大きいとき
   ダイナミック応答調整は不要です。

例

以下は、「ダイナミック軌跡応答の補正」機能の、加々速度一定(BRISK)の加速による移動動作に対する効果を示す例です。

以下のパラメータが設定されていると仮定します。

MD20465 $MC_ADAPT_PATH_DYNAMIC[0] = 1.5

MD20460 $MC_LOOKAH_SMOOTH_FACTOR = 1.0

MD32440 $MA_LOOKAH_FREQUENCY[AX1] = 20 Hz　T_{AX1} = 1/20 Hz = 50 ms

MD32440 $MA_LOOKAH_FREQUENCY[AX2] = 10 Hz　T_{AX2} = 1/10 Hz = 100 ms

MD32440 $MA_LOOKAH_FREQUENCY[AX3] = 20 Hz　T_{AX3} = 1/20 Hz = 50 ms

注記

ダイナミック応答調整の効果を示すために、スムージング係数(MD20460)の値を「1」に設定しているため、「軌跡速度のスムージング」機能は実質的に解除されています。

軌跡には X = AX1、Y = AX2、Z = AX3 の 3 つの軸が含まれます。

軸 AX2 が関連する軌跡移動に対して、T_{AX2} より継続が短いすべての減速と加速処理が補正されます。

軸 AX1 と AX3 のいずれかまたは両方が軌跡移動に含まれる場合、$T_{AX1} = T_{AX3}$ より継続が短いすべての減速と加速処理が補正されます。

関連する時間枠は、以降の図で $t_{adapt...}$ とマークされています。

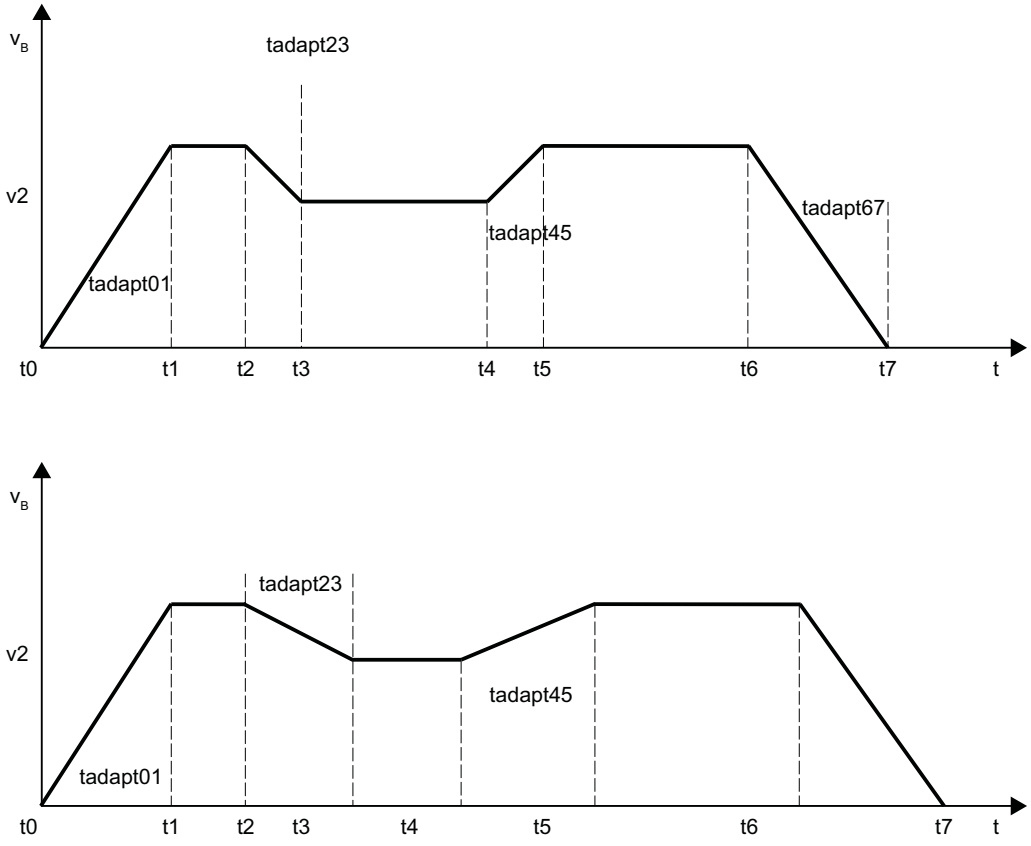


図 5-9　スムージングまたはダイナミック応答補正動作のない、時間的に最短の軌跡速度波形

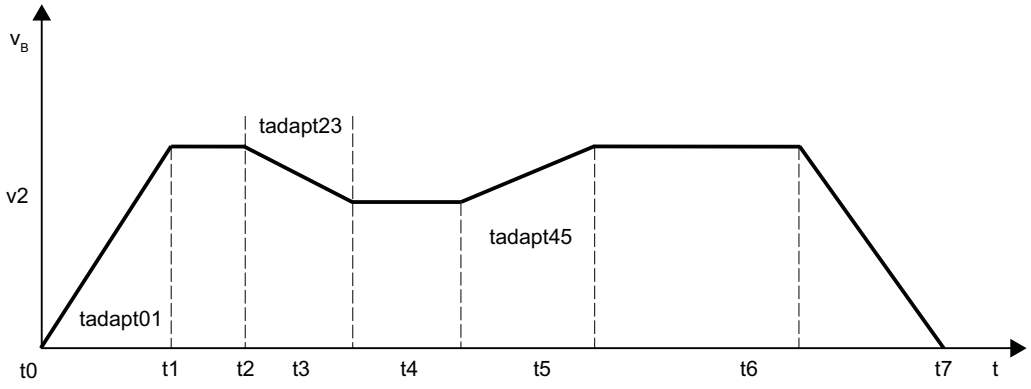


図 5-10　ダイナミック軌跡応答補正による軌跡速度波形

区間 t_0 ～ t_1 および t_2 ～ t_3: t_0 ～ t_1 の加速処理および t_2 ～ t_3 の減速処理は、補正される加速の結果、時間については $t_{adapt01}$ または $t_{adapt23}$ に拡張されます。

区間 t_4 ～ t_5: t_4 ～ t_5 の間の加速処理は、最大補正係数 1.5 により低減された加速度で実行されます。 しかし、加速処理は、時間 $t_{adapt45}$ より前に完了されます。

区間 t_6 ～ t_7: t_6 ～ t_7 の間の減速処理は、変更されずに、$t_{adapt67}$ より長く継続します。

5.4.3 ダイナミック応答制限値の決定

「ダイナミック軌跡応答の補正」機能の実装には、軌跡軸の固有周波数を決定して、パラメータを軸別の限界周波数に割り当てる(MD32440 $MA_LOOKAH_FREQUENCY)だけでなく、速度、加速、および加々速度に対してダイナミック応答限界を決定する必要があります。

手順

加々速度制限のある加速による軌跡軸の移動のためのダイナミック応答制限の決定(SOFT)について、以下に説明します。 この手順は、加々速度制限のない加速の事例に、同様に適用できます(BRISK)。

1. 以下を使用して「ダイナミック軌跡応答の補正」機能を解除します。
   MD20465 $MC_ADAPT_PATH_DYNAMIC[1] = 1
2. さまざまな移動速度での各軌跡軸の位置決め動作を観察します。 そうしているときに、希望する位置決め許容範囲が維持されるような加々速度を設定します。

   注記

   位置決め処理が開始される移動速度が大きくなると、一般に、設定できる加々速度も大きくなります。

3. もっとも余裕のある移動速度に対して決定される最大許容加々速度を使用します。
   MD32431 $MA_MAX_AX_JERK (最大加々速度)

4. 下記の式を使用して、すべての軌跡軸に対して、F_{APD} 係数を決定します。
   F_{APD} = (決定された最大加々速度) / (決定された最小加々速度)

   注記

   最小決定加々速度は、最も厳しい移動速度のときの加々速度に対する値です。

5. すべての軌跡軸を介して決定された最大の F_{APD} 係数を、軌跡のダイナミック応答の補正係数の値として入力します。
   MD20465 $MC_ADAPT_PATH_DYNAMIC[1] = F_{APD}

5.4.4 「軌跡速度のスムージング」および「軌跡のダイナミック応答の補正」機能の連携

以下の例は、連続軌跡モードでの「軌跡速度のスムージング」および「軌跡ダイナミック応答の補正」機能の連携を示しています。

例 1

加速モード:BRISK

軌跡には、X = AX1、Y = AX2、Z = AX3 の 3 つの軸が含まれます。

以下のパラメータが設定されていると仮定します。

MD20465 $MC_ADAPT_PATH_DYNAMIC[0] = 3
MD20460 $MC_LOOKAH_SMOOTH_FACTOR = 80.0
MD32440 $MA_LOOKAH_FREQUENCY[AX1] = 20 T_{AX1} = 1/20 Hz = 50 ms
MD32440 $MA_LOOKAH_FREQUENCY[AX2] = 20 T_{AX2} = 1/20 Hz = 50 ms
MD32440 $MA_LOOKAH_FREQUENCY[AX3] = 20 T_{AX3} = 1/20 Hz = 50 ms

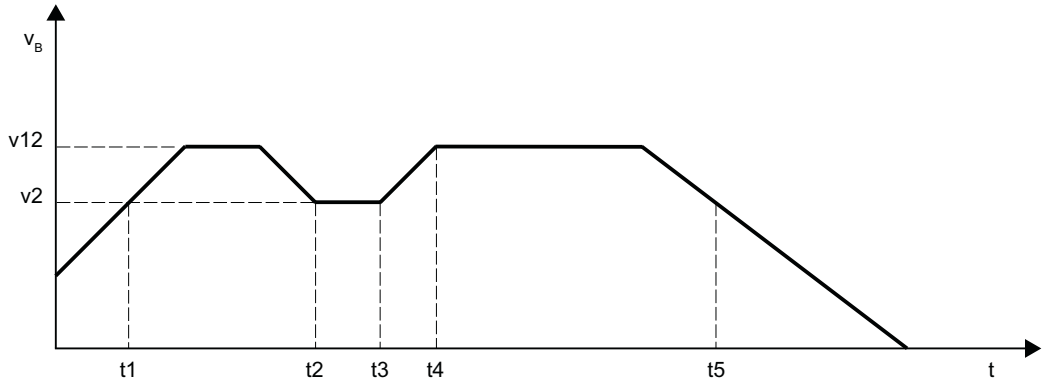


図 5-11 スムージングまたはダイナミック補正動作のない、時間的に最短の軌跡速度波形

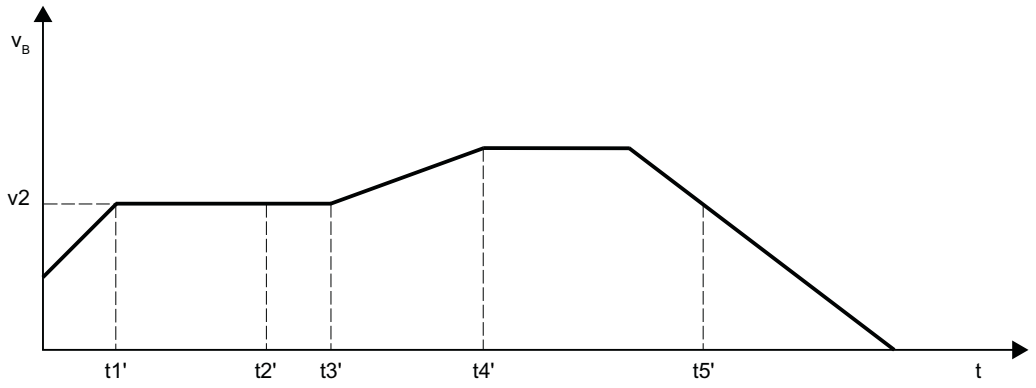


図 5-12　軌跡速度のスムージングによる軌跡速度波形、およびダイナミック軌跡応答の補正

軌跡速度へのスムージングの効果:

区間 t_1 ～ t_2:　t_1 および t2 の間の加速および減速処理は実行されません。これは、v_{12} まで加速処理を行わないことにより増加した加工時間が、スムージング係数の 80%が適用された場合に得られる時間より小さいためです。

区間 t_3 ～ t_5:　t_3 および t_5 の間の加速および制動波形は、この条件を満たさず、パラメータで設定したスムージング時間 T_{Axn} = 2/20 Hz = 100 ms より長い時間を要します。

ダイナミック応答調整の効果:

区間 t_3 ～ t_4:　t_3 および t_4 の間の加速処理は、MIN(T_{AXn}) = 1/20 Hz = 50 ms より短いため、補正係数 3 で低減された加速度で実行されます。

t_1 までの区間:　軌跡スムージングの後の残りの t_1 までの加速は、ダイナミック応答調整により t_1'までの期間に拡大されます。

注記

この例は、軌跡速度のスムージングで除去されない加速または減速処理は、その後にダイナミック軌跡応答を補正して最適化できることを示します。 このため、できれば両方の機能を常に有効にしてください。

例 2

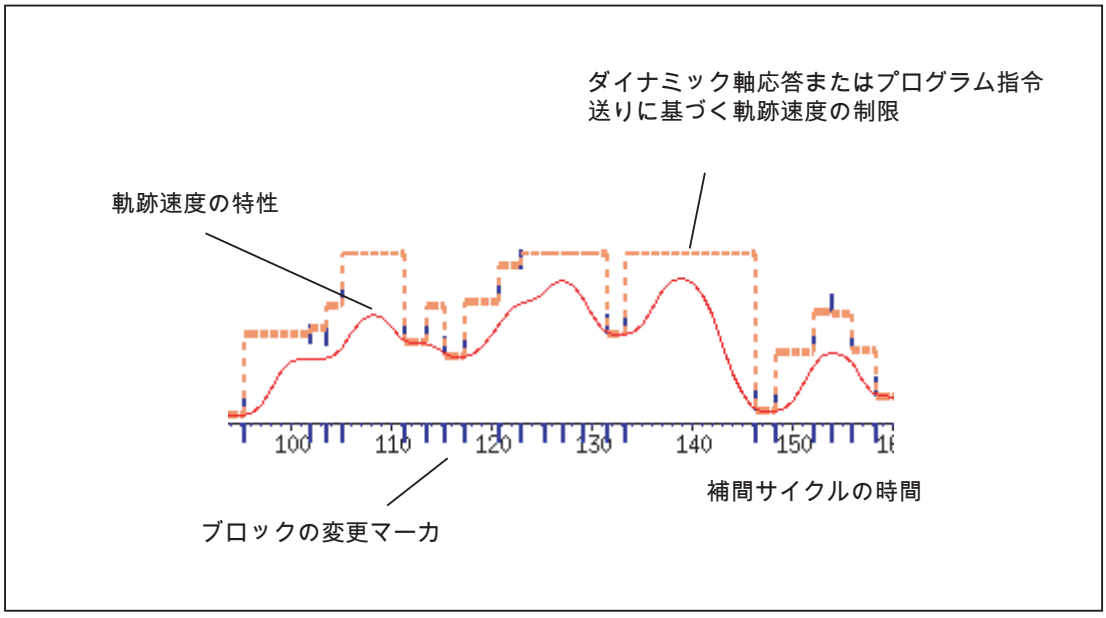
加速モード:SOFT

軌跡には X = AX1、Y = AX2、Z = AX3 の 3 つの軸が含まれます。

以下のパラメータが設定されていると仮定します。

MD20465 $MC_ADAPT_PATH_DYNAMIC[1] = 1
MD20460 $MC_LOOKAH_SMOOTH_FACTOR = 0.0
MD32440 $MA_LOOKAH_FREQUENCY[AX1] = 10　　T_{AX1} = 1/20 Hz = 100 ms
MD32440 $MA_LOOKAH_FREQUENCY[AX2] = 10　　T_{AX2} = 1/20 Hz = 100 ms
MD32440 $MA_LOOKAH_FREQUENCY[AX3] = 20　　T_{AX3} = 1/20 Hz = 50 ms

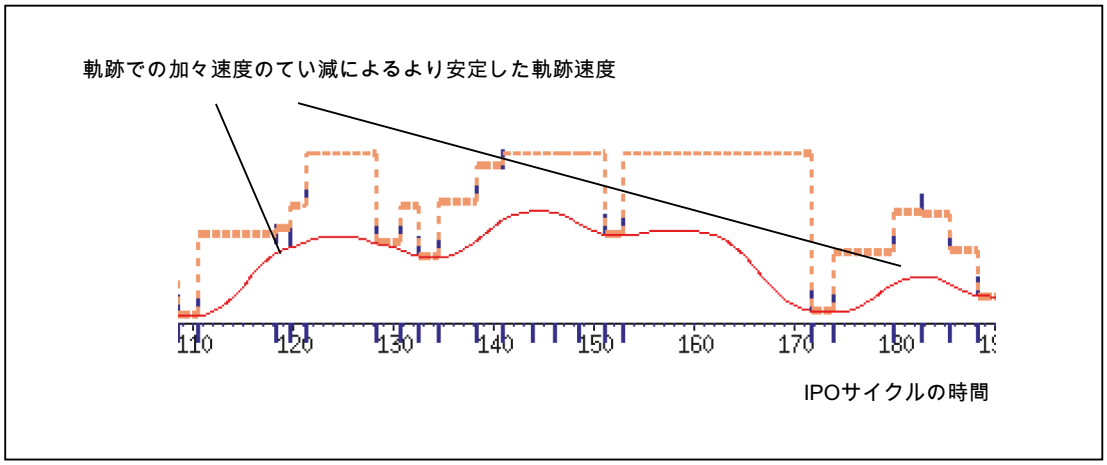
これにより、軌跡速度のスムージングもダイナミック軌跡応答も行わない、時間的に最適化された軌跡速度波形が得られます。



パラメータ設定は、以下のように変わります。

MD20465 $MC_ADAPT_PATH_DYNAMIC[1] = 4
MD20460 $MC_LOOKAH_SMOOTH_FACTOR = 1.0

この結果、ダイナミック軌跡応答の補正による、軌跡速度のスムージングが最小の(実質的には解除された)軌跡速度波形が得られます。

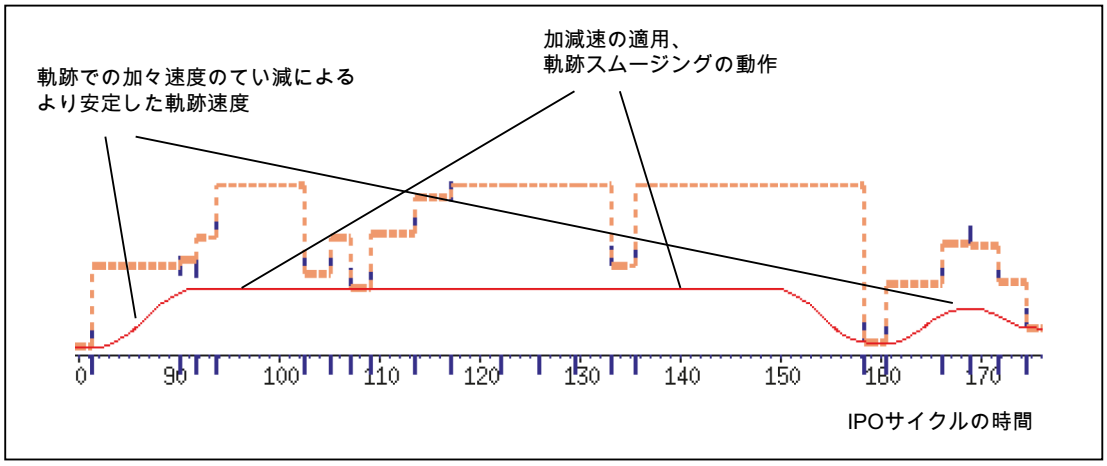


スムージング係数は、1%ではなく、0% (初期設定により)に設定されます。

MD20460 $MC_LOOKAH_SMOOTH_FACTOR = 0.0

100%のスムージング係数は、このパラメータ設定で有効になります。

これにより、軌跡速度のスムージングによる軌跡速度波形が得られ、ダイナミック軌跡応答の補正が行われます。



5.4.5 軌跡補間のダイナミック応答モード

機能

用途別のダイナミック応答の設定は、マシンデータに保存でき、パートプログラムで G グループ 59 の命令(軌跡補間のダイナミック応答モード)を使用して有効にすることができます。

命令	ダイナミック応答の設定を有効にする対象:
DYNNORM	標準ダイナミック応答設定
DYNPOS	位置決めモード、タッピング
DYNROUGH	荒削り
DYNSEMIFIN	仕上げ
DYNFINISH	滑らかな仕上げ

注記

G グループ 59 の命令(軌跡補間のダイナミック応答モード)で決定されるのは、**軌跡軸**のダイナミック応答のみです。この命令は、以下のものには影響しません。

- 位置決め軸
- PLC 軸
- コマンド軸
- 軸連結に基づく移動
- ハンドルによる重畳動作
- JOG 動作
- レファレンス点復帰(G74)
- 固定点アプローチ(G75)
- 早送り移動(G0)

標準ダイナミック応答の設定(DYNNORM)は常に、上記の軸移動に有効です。

用途

ダイナミック応答の設定を切り替えると、荒削りは時間について、中削りは面について、などのように最適化できます。

パラメータ設定

軸別ダイナミック応答設定:

- MD32300 $MA_MAX_AX_ACCEL[<n>] (軸加速度)
- MD32310 $MA_MAX_ACCEL_OVL_FACTOR[<n>] (速度における軸不連続変化の過負荷係数)
- MD32431 $MA_MAX_AX_JERK[<n>] (軌跡移動の最大軸加々速度)
- MD32432 $MA_PATH_TRANS_JERK_LIM[<n>] (連続軌跡モードのブロック遷移部での最大軸加々速度)
- MD32433 $MA_SOFT_ACCEL_FACTOR[<n>] (SOFT による加速制限の倍率)

チャネル別ダイナミック応答設定:

- MD20600 $MA_MAX_PATH_JERK[<n>] (軌跡関連の最大加々速度)
- MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL[<n>] (軌跡加速度への軌跡曲率の影響)
- MD20603 $MC_CURV_EFFECT_ON_PATH_JERK[<n>] (軌跡加々速度への軌跡曲率の影響)

ここで、インデックス**<n> =**

0 DYNNORM の場合

1 DYNPOS の場合

2 DYNROUGH の場合

3 DYNSEMIFIN の場合

4 DYNFINISH の場合

注記

インデックスを使用しないでマシンデータを**書き込む**と、当該のマシンデータのすべてのフィールド要素に同じ値が設定されます。

インデックスを使用しないでマシンデータを**読み取る**と、常にインデックス 0 のフィールドの値が提供されます。

G 命令の抑制

使用しない G グループ 59 の G 命令(軌跡補間のダイナミック応答モード)は、以下のマシンデータでマスクすることをお勧めします。

MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[<n>] (再設定された NC 命令のリスト)

マスクした G 命令を使用した場合は、アラームが表示されます。これにより、パラメータが設定されていないマシンデータが無効になります。

例

G 命令 DYNPOS と DYNSEMIFIN は、以下の設定で抑制できます。

- MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[0]="DYNPOS"
- MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[1]=" "
- MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[2]="DYNSEMIFIN"
- MD10712 $MN_ NC_USER_CODE_CONF_NAME_TAB[3]=" "

参照先

G グループ 59 の G 命令(軌跡補間のダイナミック応答モード)のプログラミングに関する詳細情報は、以下に記載されています。

参照先

『プログラミングマニュアル、上級編』、軌跡の移動動作

5.4.6 自由形式面モード:基本機能

はじめに

ツールと金型加工の用途では、ワーク面が、可能な限り均一であることが重要です。この条件は通常、ワーク面の精度より重要です。

ワーク面が均一でない原因には、以下が考えられます。

- ワークを製造するパートプログラムに、均一でない座標が含まれています。これは特に、曲率と軸ねじれのプロファイルに影響します。

  注記

  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。軸ねじれとは、曲率の変化(最初の派生物)です。

  ジオメトリが均一でないために、パートプログラムの処理中に機械のダイナミック応答制限に達してしまい、不要な減速と加速処理が発生します。これにより、軸の通過の有効性に応じて、さまざまな輪郭の誤差が生じます。

- 不要な減速と加速処理により、機械の振動が発生し、ワークに余分なマークが付けられる結果となります。

これらの原因を解消するためのさまざまなオプションが用意されています。

- CAD/CAM システムの使用
  CAD/CAM システムで生成したパートプログラムには、軌跡速度の不要な減速を防止する、非常に均一な曲率と軸ねじれプロファイルが含まれています。
- 最高軌跡速度を、曲率と軸ねじれプロファイルでの不要なジオメトリ変動が無効となるように指定します。

機能

「自由曲面モード:基本機能」を使用すると、軌跡速度制限を定義して、軸の加速と加々速度に関して機械の動的制限を超えることなく、曲率と軸ねじれでの小さなジオメトリ変動に影響されないようにすることができます。

これには下記の長所があります。

- 軌跡速度波形の均一性の向上
- ワーク面の均一性の向上
- 処理時間の短縮(機械のダイナミック応答で許容される場合)

用途

機能は、主に自由曲面から成るワークの処理に使用します。

必要条件

この機能は、次のように、必要なメモリ容量がメモリ構成中に予約されている場合にのみ起動します。

MD28610 $MC_MM_PREPDYN_BLOCKS = 10

入力した値で、軌跡速度の決定(速度解析)で考慮すべきブロック数が指示されます。
実用的な値は「10」です。

MD28610 の値が「0」の場合は、特定のブロック内の軸の移動のみが、そのブロックの軸の最高速度の決定時に考慮されます。軌跡の速度(値> 0)の決定時には隣接ブロックのジオメトリも考慮され、より均一な軌跡速度の波形が実現されます。

起動と解除

この機能は、以下を使用して、ダイナミック応答モードのたびに独立してオンとオフを切り替えることができます(「軌跡補間のダイナミック応答モード (ページ 279)」の章を参照してください)。

MD20606 $MC_PREPDYN_SMOOTHING_ON[<n>] = <値>

インデックス<n>	ダイナミック応答モード	<値>	自由曲面モード:基本機能
0	標準ダイナミック応答の設定(DYNNORM)	0	オフ
		1	オン
1	位置決めモード、タッピング(DYNPOS)	0	オフ
		1	オン
2	荒削り(DYNROUGH)	0	オフ
		1	オン
3	仕上げ(DYNSEMIFIN)	0	オフ
		1	オン
4	滑らかな仕上げ(DYNFINISH)	0	オフ
		1	オン

注記

保管に関する追加の必要条件により、この機能は、関連する処理チャネルでのみ起動してください。

パラメータ設定

輪郭サンプリング係数の変更

輪郭の曲線部の補間中に発生する割線アラームは、以下の係数に応じて異なります。

- 曲率
- 補間クロック周期(MD10071 $MN_IPO_CYCLE_TIME で表示)
- 関連する輪郭の移動速度

発生する割線アラームの最大値は、以下のマシンデータで各軸に対して定義します。

MD33100 $MA_COMPRESS_POS_TOL (圧縮による最大許容範囲)

設定した補間クロック周期が大きすぎると、曲率の大きい輪郭で最高軌跡速度が低下する可能性があります。これは、このような場合に、ワーク面も適切な精度で製造されるようにするために必要です。

輪郭サンプリング係数を変更すると、補間器で輪郭の曲線部をサンプリングする時間の間隔(輪郭サンプリング時間)に、補間クロック周期とは異なる値を設定できます。輪郭サンプリング時間を補間クロック周期より短くすると、曲率の大きい輪郭での軌跡速度の低下を防止できます。

輪郭サンプリング時間は、次のマシンデータを使用して設定します。

MD10682 $MN_CONTOUR_SAMPLING_FACTOR

有効な輪郭サンプリング時間は、以下のように計算します。

$T_s = f * T_1$

各記号の意味は次のとおりです。

T_s =有効な輪郭サンプリング時間
T_1 =補間クロック周期
f = 輪郭サンプリング係数(MD10682 の値)

初期設定の輪郭サンプリング係数は「1」です。つまり、輪郭サンプリング時間は補間クロック周期と同じです。

輪郭サンプリング係数には、「1」より大きい、または小さい値を指定できます。

「1」より小さい値を設定すると、輪郭サンプリング精度の監視が解除されます。

設定した最小輪郭サンプリング時間より短いサンプリング時間は、設定しないでください。

MD10680 $MN_MIN_CONTOUR_SAMPLING_TIME

注記

MD10680 は特に、すべてのコントローラモデルに対して設定されており、変更できません。

プログラミング

マシンデータ MD20606 $MC_PREPDYN_SMOOTHING_ON の設定内容に応じて、「自由曲面モード:基本機能」のオンとオフは、有効なダイナミック応答モードをパートプログラムで変更して切り替えることができます。

例:

パラメータ MD20606 $MC_PREPDYN_SMOOTHING_ON[2-4] = 1 と MD20606 $MC_PREPDYN_SMOOTHING_ON[0-1] = 0 を割り当てると、この機能を命令 DYNROUGH、DYNSEMIFIN、および DYNFINISH でオンに切り替えたり、命令 DYNNORM と DYNPOS でオフに切り替えたりできます。

下記も参照

接線の変化が連続的なブロック遷移部の丸み付け(G645) (ページ 252)

速度依存加々速度調整(軸別) (ページ 347)

自由形式面モード: 拡張機能 (ページ 261)

5.5 コンプレッサ機能

5.5.1 直線ブロック、円弧ブロック、および早送りブロックの圧縮

5.5.1.1 機能

複雑な輪郭を記述するために、CAD/CAM システムは、軌跡長が非常に短いブロックを含む大量の直線ブロックと円弧ブロックを生成します。最大許容軌跡速度は、多くの場合、ブロック変更持続時間によって制限されます。特定の軌跡速度以上では、新しい移動ブロックを先読みで十分に用意できなくなり、メインランに切り替わります。

直線ブロックの圧縮

コンプレッサ機能は、パラメータ設定可能な軌跡の精度を維持しながら、連続する直線ブロックを可能なかぎり長い経路を含む多項式ブロックに置換します。これには、以下のメリットがあります:

- 移動ブロック数の削減
- 軌跡速度の向上
- 加工面品質の向上
- 連続したブロック遷移部

有効かつ最も重要な特性/属性を備えたコンプレッサ機能を、次の表に示します。

コンプレッサ	機能	ブロック遷移部での連続性	使用に関する注記
COMPON	COMPON は、**10** 個までの連続する直線ブロック、タイプ「G01 X...Y...Z...F...」から多項式ブロックを作成します。	連続する速度	
COMPCURV	COMPON と同じ	連続する速度と加速度	

コンプレッサ	機能	ブロック遷移部での連続性	使用に関する注記
COMPCAD	COMPCAD は、**任意の数**の連続する直線ブロックから多項式ブロックを作成できます。	連続する速度と加速度	COMPCAD は、プロセッサとメモリを大量に使用します。そのため、CAD/CAM プログラムで対策を講じても加工面品質が改善されない場合のみ、COMCAD を使用してください。
COMPSURF	COMPCAD と同じ	連続する速度と加速度	COMPSURF を使用すると、COMPCAD を使用した場合より、さらに良い結果が得られます。ただし、COMPCAD と同様に、COMPSURF も多くの計算時間とメモリ領域を要します。 精密な仕上げプログラムでの傾斜、CAD/CAM プログラムでの低いデータ品質や不規則な点分布などの対策として COMPSURF を使用すると、ワーク面の品質を大幅に改善することができます。 さらに、COMPSURF で、複数のフライス削り軌跡で、方向に依存しない同じスムージングを容易に行ったり、無効にしたりできます。これにより、双方向のフライス工具による加工面品質が大幅に向上します。

移動指令以外の命令(補助機能の出力など)が、圧縮される移動ブロックの中または間でプログラム指令されている場合は、圧縮が中断されます。

プログラム指令位置までの計算された軌跡の最大許容誤差は、マシンデータを使用してすべてのコンプレッサ機能に対して指定できます(パラメータ設定 (ページ 288)を参照)。COMPON や COMPCURV とは異なり、COMPCAD と COMPSUR では、設定された許容範囲が、隣接する軌跡の異なった方向には使用されません。

円弧ブロックおよび早送りブロックの圧縮

直線ブロックの圧縮に加えて、すべてのコンプレッサ機能で、早送りブロック(G0 ブロック)を高速に圧縮することもできます。それに対して、円弧ブロックは COMPCAD コンプレッサ機能のみを使用して圧縮できます。

マシンデータで**数百の位置**を使用して、特定のチャネルに対して、円弧ブロック/早送りブロックの圧縮を設定します。

MD20482 $MC_COMPRESSOR_MODE = <値>

値	意味
0xx	円弧ブロックと G0 ブロックは圧縮**されません**。これは、初期 SW バージョンと互換性があります。
1xx	円弧ブロックは線形化され、COMPCAD によって圧縮されます。 長所: コンプレッサ機能は、より正確に動作し、したがって一般的により優れた表面が作成されます。 制限事項: コンプレッサ機能が、NC プログラムの不具合の影響を受けやすくなります。したがって、互換性を保つために、設定を 0xx のままにしておくことが必要な場合があります。
2xx	G0 ブロックが圧縮されます。別の許容範囲が有効になることがあります(「早送り移動の許容範囲 (ページ 303)」を参照してください)。 長所: したがって、許容範囲がより大きく設定され、G0 切り込み動作の圧縮により、より高速かつより流動的な実行が可能です。
3xx	前記の 2 つのオプションの組合わせ:円弧ブロックに加えて、G0 ブロックが圧縮されます。

起動と解除

G グループ 30 の G 命令を使用して、コンプレッサ機能 COMPON、COMPCURV、COMPCAD、および COMPSURF を起動/解除します(「プログラミング (ページ 291)」を参照してください)。

5.5.1.2 セットアップ

パラメータ設定

軸マシンデータ

番号	識別子$MA_	意味
MD33100	COMPRESS_POS_TOL	圧縮による軸別の最大許容軌跡誤差

チャネルマシンデータ

番号	識別子$MC_	意味
MD20170	COMPRESS_BLOCK_PATH_LIMIT	圧縮できる NC ブロックの最大移動距離
MD20171	SURF_BLOCK_PATH_LIMIT	COMPSURF による圧縮のための圧縮可能な NC ブロックの最大移動距離
MD20172	COMPRESS_VELO_TOL	圧縮のための送り速度からの最大許容誤差
MD20173	SURF_VELO_TOL	COMPSURF による圧縮のための送り速度からの最大許容誤差
MD20482	COMPRESSOR_MODE	コンプレッサの動作原理
MD20484	COMPRESSOR_PERFORMANCE	コンプレッサのパフォーマンス
MD20485	COMPRESS_SMOOTH_FACTOR	特定のダイナミック応答モードの COMPCAD による圧縮のスムージング係数
MD20486	COMPRESS_SPLINE_DEGREE	特定のダイナミック応答モードの COMPCAD による圧縮のスプライン度
MD20487	COMPRESS_SMOOTH_FACTOR_2	特定のダイナミック応答モードの COMPCAD による圧縮の回転軸のスムージング係数
MD28071	MM_NUM_SURF_LEVELS	COMPSURF 機能のサイズ決め(DRAM)
MD28072	MM_MAXNUM_SURF_GROUPS	軸グループに関する COMPSURF 機能のサイズ決め(DRAM)

チャネルセッティングデータ

番号	識別子$SC_	意味
SD42470	CRIT_SPLINE_ANGLE	COMPCAD による圧縮のコーナ角度制限
SD42471	MIN_CURV_RADIUS	COMPCAD による圧縮の曲率 の最小半径

番号	識別子$SC_	意味
SD42472	MIN_SURF_RADIUS	COMPSURF による圧縮の曲率 の最小半径
SD42473	ACTNUM_SURF_GROUPS	COMPSURF の軸グループの数
SD42475	COMPRESS_CONTUR_TOL	圧縮なしの最大許容輪郭誤差
SD42476	COMPRESS_ORI_TOL	圧縮の工具の向きの最大誤差 **注記:** 有効な方向座標変換の場合のみ。
SD42477	COMPRESS_ORI_ROT_TOL	圧縮の工具の回転の最大誤差 **注記:** 回転する工具を備えた 6 軸の機械のみ。

注記

コーナ制限角度とコンプレッサ機能 COMPCAD

セッティングデータ SD42470 $SC_CRIT_SPLINE_ANGLE によって設定されたコンプレッサ機能 COMPCAD のコーナ制限角度は、コーナ検出のための近似計測値としてのみ使用します。妥当性の評価により、コンプレッサが、より平たいブロック遷移部をコーナとして識別することも、より大きい角度を異常値として識別することもできます。

Advanced Surface/Top Surface によるツールと金型加工の推奨設定

コンプレッサ機能は、ツールと金型加工での自由曲面のフライス加工において非常に重要です。ライセンスが必要なオプション「Advanced Surface」または「Top Surface」の一部としてコンプレッサ機能が使用される場合は、設定推奨値に従ってください!

設定されたマシンデータおよびセッティングデータをチェックするために、SIOS ポータルで特別なテストプログラムが使用できます。

- Advanced Surface 用のテストプログラム (https://support.industry.siemens.com/cs/ww/en/view/78956392)
- Top Surface (https://support.industry.siemens.com/cs/ww/en/view/109738423)用のテストプログラム

5.5.1.3 プログラミング

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

プログラムコード	コメント
COMPOF	; コンプレッサ機能をオフ
...	

5.5.1.4 必要条件

方向座標変換 TRAORI

方向座標変換が有効な場合は、コンプレッサ機能も、工具の向きおよび工具の回転のための移動ブロックを圧縮できます。

参照先:
『機能マニュアル、応用機能』、「F2:複数軸の座標変換」の「向き」の「向きの圧縮」の章

計算ありのブロック検索

タイプ 2 または**タイプ 4**(...の**計算による**ブロック検索)のブロック検索のターゲットブロックが、コンプレッサ機能が有効なプログラムセクションにある場合は、再位置決め時にコンプレッサによって計算された軌跡上で位置にアプローチします。これらの位置は、パートプログラムで指令された軌跡上の位置と正確に一致するようにしてください。

圧縮によって置き換えられたパートプログラムブロックは、ブロック検索でターゲットブロックとは見なされません。アラーム 15370「検索ターゲットが見つからない」が出力されます。

5.5.2 短いスプラインブロックの圧縮

機能

スプラインブロックは、CAD/CAM システムを使用して、複雑な輪郭を描くために生成されます。長い軌跡長のスプラインブロック間に、非常に短い軌跡長のスプラインブロックが生成されます。これにより、軌跡速度が大幅に低下することになります。短いスプラインブロックを圧縮する機能により、最大限の軌跡長の新しいスプラインブロックが生成されます。

適用

システム	適用
SINUMERIK 840D sl	標準(基本仕様)
SINUMERIK 828D	オプション

セットアップ

起動

短いスプラインブロックの圧縮は、次のスプラインタイプに対して起動できます。

- BSPLINE
- BSPLINE/ORICURVE
- CSPLINE

チャネル別マシンデータで起動されます。

MD20488 $MC_SPLINE_MODE, ビット<n> = <値>

ビット	<値>	スプラインタイプ	短いスプラインブロックの圧縮
0	0	BSPLINE	無効
	1	BSPLINE	有効
1	0	BSPLINE/ORICURVE	無効
	1	BSPLINE/ORICURVE	有効
2	0	CSPLINE	無効
	1	CSPLINE	有効

必要条件

- 移動指令以外の命令(補助機能の出力など)が、圧縮される移動ブロックの中または間でプログラム指令されている場合は、スプラインブロックを結合できません。
- 連続して一つのプログラム区間に結合できる最大のブロック数は、ブロック解析中に使用できる、設定されたブロックメモリのサイズによって異なります。
  MD28070 $MC_MM_NUM_BLOCKS_IN_PREP (ブロック解析用のブロック数)

例

ブロックの移動の実行時に軌跡速度を上げるには、短いスプラインブロックの圧縮を BSPLINE 補間に対して起動します。

MD20488 $MC_SPLINE_MODE、ビット 0 = 1

プログラムコード	コメント
N10 G1 G64 X0 Y0 Z0 F1000	; 初期設定
N20 G91 F10000 **BSPLINE**	; 起動:B スプライン
N30 X0.001 Y0.001 Z0.001	; これ以降は、「短いスプラインブロックの結合」の章
N40 X0.001 Y0.001 Z0.001	
...	

5.6 輪郭/向きの許容範囲

5.6.1 セットアップ

5.6.1.1 パラメータの割り付け

マシンデータ

輪郭の許容誤差/向きの許容誤差

MD33100 $MA_COMPRESS_POS_TOL[<軸>] = <値> (圧縮による最大許容範囲)

軸マシンデータを使用して、各軸の最大許容輪郭誤差(輪郭の許容範囲)または工具の向きの角度誤差(向きの許容範囲)が設定されます。このマシンデータは、以下の機能で有効です。

- 丸み付け:G642、G643、G644、G645
- コンプレッサ:COMPON、COMPCURV、COMPCAD、COMPSURF
- 値が大きくなるほど、より多くの短いブロックを 1 つの長いブロックに圧縮できます。

このマシンデータは、スムージング機能 G641 には効果がありません。G641 では、ADIS または ADISPOS によりプログラム可能なブロック遷移に対する軌跡距離が有効です。

スムージングモード

MD20480 $MC_SMOOTHING_MODE (G64x による丸み付け動作)

コンプレッサモード

MD20482 $MC_COMPRESSOR_MODE (圧縮のモード)

G645 のスムージング

MD33120 $MA_PATH_TRANS_POS_TOL (G645 のスムージングの最大輪郭誤差)

接線の変化が連続的な丸み付けには効果がありますが、連続的に曲線を描くブロック遷移部(円-直線など)には効果がありません。

セッティングデータ

チャネル別の輪郭許容範囲

SD42465 $SC_SMOOTH_CONTUR_TOL (最大輪郭誤差)

チャネル別の向きの許容範囲

SD42466 $SC_SMOOTH_ORI_TOL (工具オリエンテーションの最大角度誤差)

OST によるスムージングでのチャネル別の向きの許容範囲

SD42676 $SC_ORI_SMOOTH_TOL (丸み付けでの向きによるスムージングの許容範囲)

ORISON を使用した旋回スムージングのチャネル別旋回許容範囲

SD42678 $SC_ORISON_TOL (向きのスムージングの許容範囲)

5.6.2 プログラミング

5.6.2.1 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...	

5.6.2.2 輪郭/旋回の許容範囲のプログラミング(CTOL、OTOL、ATOL)の追加情報

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。

5.7 早送り移動

5.7.1 機能

5.7.1.1 早送り

早送り時には、可能なかぎり高速に、速度プログラムされたツール移動が実行されます。

早送りの速度は、軸ごとに個別に定義します(「パラメータの割り付け (ページ 304)」を参照)。

用途

たとえば、早送り移動は以下のタスクに使用します:

- 工具の迅速な位置決め
- ワーク周囲の移動
- 工具交換位置へのアプローチ
- 工具の後退

注記

早送り移動は、ワーク加工には対応しません。

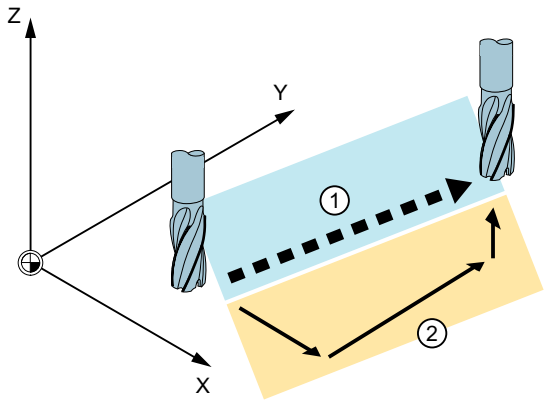
起動

早送りは、パートプログラムで G0 をプログラムすることで有効になります(「プログラミング (ページ 305)」を参照)。

5.7.1.2 早送り移動の軌跡軸の補間動作

直線/非直線補間

軌跡軸は、直線補間または非直線補間モードの早送り移動動作で移動できます。



① 直線補間による早送りパス

② 非直線補間による早送りの一軸移動

直線補間

機能:

- 軌跡軸はまとめて補間されます。
- G0 でプログラム指令された工具の移動は、可能な最高速度で実行されます(早送り)。
- 早送り速度は、各軸に対して個別に定義されます。
- 早送り移動動作が複数の軸で同時に実行される場合、早送り速度は、該当する軌跡区間移動のための必要時間が最大の軸によって決定されます。

直線補間は、次の場合にはいつでもおこなわれます。

- 次の位置決め軸の移動を**実行できない**、G0 による G 命令の組合わせ。
  G40、G41、G42、G96、G961、および MD20750 $MC_ALLOW_G0_IN_G96 == FALSE
- G0 と G64 の組合わせ
- 圧縮または座標変換が有効な場合。
- ポイントツーポイント(PTP)移動モードの場合。
- 輪郭ハンドルを選択している場合(FD=0)。
- ジオメトリ軸の座標回転フレームが有効な場合。
- ニブリングがジオメトリ軸に有効な場合。

非直線補間

機能:

- 各軌跡軸は、その他の軸とは無関係に、軸固有の早送り速度で、単独軸(位置決め軸)として補間されます。
- PLC とシンクロナイズドアクションによる、チャネル別残移動距離削除命令は、軌跡軸としてプログラム指令されたすべての位置決め軸に適用されます。

非直線補間では、軸別加々速度に関連して、次の 2 つの設定のいずれかが適用されます。

- 位置決め軸命令 BRISKA、SOFTA、DRIVEA
- マシンデータ:
  - MD32420 $MA_JOG_AND_POS_JERK_ENABLE
  - MD32430 $MA_JOG_AND_POS_MAX_JERK

残移動距離を参照するために、既存のシステム変数($AC_PATH、$AC_PLTBB と $AC_PLTEB)がサポートされています。

⚠ 注意
衝突の可能性
非直線補間の工具移動は直線補間の工具移動とは異なる場合があるので、パス移動の座標値を基準とするシンクロナイズドアクションが有効にならないことがあります。

補間タイプの選択

実際には早送り移動で有効になる補間のタイプは、マシンデータによってプリセットされます(「パラメータの割り付け (ページ 304)」を参照)。

初期設定とは無関係に、パートプログラムで適切な補間動作を設定できます(「早送り移動の直線補間のオン/オフを切り替えます(RTLION、RTLIOF) (ページ 306)」を参照)。

5.7.1.3 早送り移動の許容範囲

早送り移動の許容範囲は、G0 許容範囲係数を指定することにより、ワーク加工の許容範囲とは異なる設定が可能です。

長所

早送り移動の許容範囲が大きくなり(G0 許容範囲係数> 1)、G0 ブロックをより高速に移動できます。

必要条件

G0 許容範囲係数は、次の条件が満たされる場合にのみ有効です。

- 次の機能のいずれかが動作中である。
  - コンプレッサ機能(COMPON、COMPCURV、COMPCAD、または COMPSURF)
  - スムージング機能(G642 または G645)
  - 旋回スムージング OST
  - ORISON 旋回スムージング
  - 軌跡に関連する旋回のスムージング(ORIPATH)
- パートプログラム内の複数の(≥ 2)連続する G0 ブロック
  単一の G0 ブロックの場合は、G0 許容範囲係数が無効です。これは、G0 以外の移動から G0 移動へ(および G0 移動から G0 以外の移動)の遷移時には常に、「**小さい方の許容範囲**」が適用されるためです(ワーク加工許容範囲)。

G0 許容範囲係数の定義

早送り移動の許容範囲係数は、チャネル固有のマシンデータによってプリセットされます(「パラメータの割り付け (ページ 304)」を参照)。

プリセット許容範囲係数は、パートプログラムでプログラムすることによって一時的に調整できます(「早送り移動(STOLF)の許容範囲係数を調整します (ページ 308)」を参照)。

5.7.1.4 早送りオーバライド

機械操作パネルの早送りオーバライドスイッチを使って、オペレータはオンサイトのパーセント値により、早送り速度を減少させることができます。これは、即時に有効になります。

早送り補正が有効な場合、直線または非直線の補間により早送りモードで移動し、現在のチャネルに割り付けられたすべての軌跡軸に適用されます。

詳細については、「機械操作パネルでの送り速度オーバライド (ページ 1732)」を参照してください。

5.7.2 セットアップ

5.7.2.1 パラメータの割り付け

早送り速度

早送り速度は、以下のマシンデータによって軸ごとに定義される最大許容軸速度に対応します:

MD32000 $MA_MAX_AX_VELO (最大軸速度)が、連結が有効になった後にも適用されます。

早送り移動の補間動作

早送り移動の補間動作は、以下のマシンデータによって、チャネルごとにプリセットされます:

MD20730 $MC_G0_LINEAR_MODE = <値> (G0 による補間動作)

<値>	意味
0	早送りモード(G0)では、**非直線**補間が有効です。 軌跡軸は、位置決め軸として移動されます。
1	早送りモード(G0)では、**直線**補間が有効です。 軌跡軸はまとめて補間されます。

G0 許容範囲係数

早送り移動の許容範囲係数は、チャネルごとに以下のマシンデータによって設定されます:

MD20560 $MC_G0_TOLERANCE_FACTOR (G0 の許容範囲係数)

許容範囲係数には、1.0 より大きい、または小さい値を指定できます。この係数が 1.0(初期値)である場合は、非早送り移動と同じ許容範囲が早送り移動で有効になります。通常、許容範囲係数は 1.0 より大きい値に設定されます。

5.7.3 プログラミング

5.7.3.1 早送り(G0)の有効化

早送り速度での軌跡軸の移動は、G 命令 G0 により有効になります。

構文

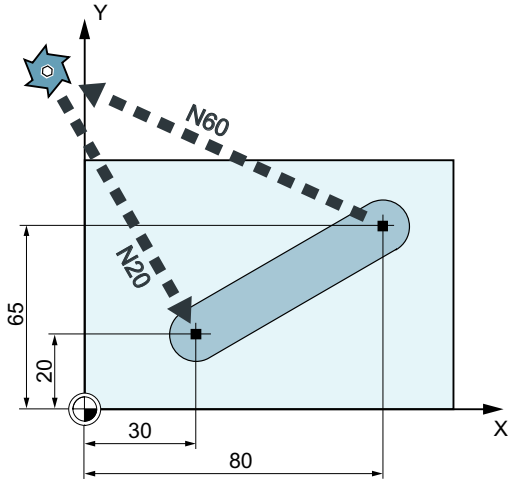
```
G0 X… Y… Z…
G0 RP=… AP=…
```

意味

G0:	早送り速度での軸の移動	
	有効:	モーダル
X...Y...Z...:	直交座標系での終点の指定	
RP=...AP=... :	極座標系での終点の指定	

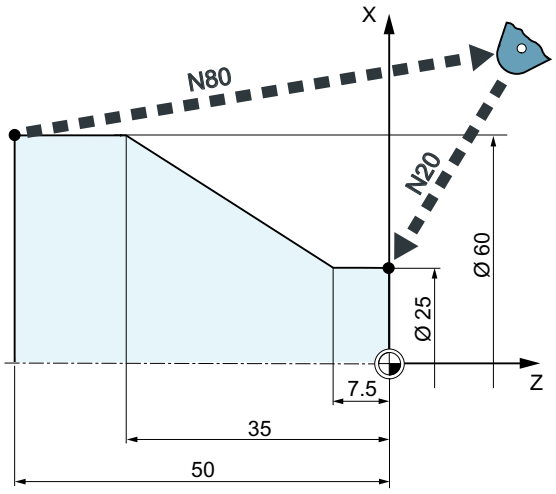
例

例 1:フライス加工



プログラムコード	コメント
N10 G90 S400 M3	; アブソリュート指令の入力、主軸は右回り
N20 G0 X30 Y20 Z2	; 開始位置へアプローチします。
N30 G1 Z-5 F1000	; 工具の切り込み。
N40 X80 Y65	; 直線上を移動
N50 G0 Z2	
N60 G0 X-20 Y100 Z100 M30	; 工具の後退、プログラム終了

例 2 :旋削



プログラムコード	コメント
N10 G90 S400 M3	; アブソリュート指令の入力、主軸は右回り
N20 G0 X25 Z5	; 開始位置へアプローチします
N30 G1 G94 Z0 F1000	; 工具の切り込み
N40 G95 Z-7.5 F0.2	
N50 X60 Z-35	; 直線上を移動
N60 Z-50	
N70 G0 X62	
N80 G0 X80 Z20 M30	; 工具の後退、プログラム終了

5.7.3.2 早送り移動の直線補間のオン/オフを切り替えます(RTLION、RTLIOF)

初期設定(MD20730 $MC_G0_LINEAR_MODE)とは無関係に、早送り移動の補間動作は、G グループ 55 の命令を使用してパートプログラムで設定することもできます。

構文

```
RTLIOF
```

```
...
RTLION
```

意味

RTLIOF:	直線補間をオフに切替えるための G 命令 ⇒ 早送りモード(G0)では、**非直線**補間が有効です。すべての軌跡軸は、相互に無関係に終点に到達します。	
	有効:	モーダル
RTLION:	直線補間をオンに切替えるための G 命令 ⇒ 早送りモード(G0)では、**直線**補間が有効です。すべての軌跡軸は、同時に終点に到達します。	
	有効:	モーダル

注記

RTLIOF の必要条件

RTLIOF **非直線**補間が確実に行われるよう、次の条件を満たしてください。

- 座標変換(TRAORI、TRANSMIT など)が有効でないこと。
- G60 が有効であること(ブロック終点で停止)。
- 圧縮(COMPOF)が有効でないこと。
- 工具半径補正(G40)が有効でないこと。
- 輪郭ハンドルを選択していないこと。
- ニブリングが有効でないこと。

これらの条件のいずれかを満たしていない場合、RTLION による直線補間が行われます。

例

プログラムコード	コメント
	; 初期設定は直線補間です:
	; MD20730 $MC_GO_LINEAR_MODE == TRUE
...	
N30 **RTLIOF**	; 直線補間をオフに切替えます。
N40 G0 X0 Y10	; G0 ブロックは、非直線補間を使用して移動します。
N50 G41 X20 Y20	; TRC 有効 □ G0 ブロックは、直線補間を使用して移動します。
N60 G40 X30 Y30	; TRC 無効 □ G0 ブロックは、非直線補間を使用して移動します。
N70 **RTLION**	; 直線補間をオンに切替えます。
...	

詳細情報

現在の補間動作の読み取り

現在の補間動作は、システム変数$AA_G0MODE で読み取ることができます。

5.7.3.3 早送り移動(STOLF)の許容範囲係数を調整します

マシンデータ(MD20560 $MC_G0_TOLERANCE_FACTOR)によりプリセットされる早送り移動(G0)の許容範囲係数は、パートプログラムで STOLF をプログラムすることで調整できます。この場合、マシンデータに指定した値は変更されません。パートプログラムのリセットまたは終了後、マシンデータで設定された許容係数は再度有効になります。

構文

```
STOLF=<値>
```

意味

STOLF:	早送り移動の許容範囲係数をプログラムするアドレス			
	用途:	● コンプレッサ機能(COMPON、COMPCURV、COMPCAD、および COMPSURF) ● スムージング機能(G642 と G645) ● 旋回スムージング OST ● ORISON 旋回スムージング ● 軌跡に関連する旋回のスムージング(ORIPATH)		
	先読み停止:	なし		
	効果:	モーダル		
	<Value>:	許容範囲係数 許容範囲係数には、1.0 より大きい、または小さい値を指定できます。この係数が 1.0(初期値)である場合は、非早送り移動と同じ許容範囲が早送り移動で有効になります。通常、許容範囲係数は 1.0 より大きい値に設定されます。		
		タイプ:	REAL	
		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。

例

プログラムコード	コメント
COMPCAD G645 G1 F10000	; コンプレッサ機能 COMPCAD
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...	
X...Y...Z...	
G0 X...Y...Z...	
G0 X...Y...Z...	; マシンデータ$MC_G0_TOLERANCE_FACTOR (=3 など) がここで適用されます。つまり、スムージング許容範囲は次の通りです。 $MC_G0_TOLERANCE_FACTOR * $MA_COMPRESS_POS_TOL
CTOL=0.02	
STOLF=4	

プログラムコード	コメント
G1 X...Y...Z...	; これ以降、0.02 mm の輪郭の許容範囲の適用が開始されます。
X...Y...Z...	
X...Y...Z...	
G0 X...Y...Z...	
X...Y...Z...	; ここから、G0 許容範囲係数 4 が適用されます、つまり輪郭許容範囲は 0.08 mm です。

他の情報

現在有効な許容範囲係数の読み取り

パートプログラムまたは現在の IPO ブロックで有効な早送り移動の許容範囲係数は、システム変数を使用して読み取ることができます。

- 次のシステム変数は、シンクロナイズドアクション、またはパートプログラムで先読み停止をおこなって読み取ります。

  $AC_STOLF　有効な G0 許容範囲係数
  現在のメインランブロックの処理時に有効であった G0 許容範囲係数。

- 次のシステム変数は、パートプログラムでは先読み停止をおこないません。

  $P_STOLF　プログラム指令の G0 許容範囲係数

動作中のパートプログラムで STOLF に値を指定していない場合は、これらの 2 つのシステム変数は、MD20560 $MC_G0_TOLERANCE_FACTOR で設定した値を示します。

ブロックで早送り(G0)が動作中でない場合は、これらのシステム変数が常に値 1 を示します。

5.8 リセット動作

MD20150

リセット(チャネルまたはモードグループのリセット)によって、チャネル別に設定された初期設定が、すべての G グループに対して有効になります。

MD20150 $MC_GCODE_RESET_VALUES (G グループの初期設定)

「連続軌跡モード、イグザクトストップ、先読み」に関連する G グループは以下のとおりです。

- グループ 10:イグザクトストップ-連続軌跡モード
- グループ 12:イグザクトストップのブロック切り替え条件
- グループ 21:加減速方法
- グループ 30:NC ブロック圧縮
- グループ 59:軌跡補間のダイナミック応答モード

初期状態の設定についての詳細は、「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 (ページ 569)」の章を参照してください。

5.9 必要条件

5.9.1 ブロック変更と位置決め軸

軌跡軸がパートプログラム内を連続軌跡モードで移動した場合、位置決め軸を移動すると、軌跡軸の応答とブロック変更の両方に同時に影響を及ぼす可能性があります。

位置決め軸の詳細な説明は、次の資料を参照してください。

参照先:

『機能マニュアル 上級機能』;「 位置決め軸(P2)」

5.9.2 ブロック変更の遅延

パートプログラムブロック内を移動するすべての軌跡軸と付加軸が特定のブロック遷移条件を満たしている場合でも、次のように、満たされていない他の条件および/または有効な機能があるために、ブロック変更が遅延する可能性があります。

例:

- PLC による補助機能の応答がない
- 後続のブロックが存在しない
- 「バッファを空にする」機能が有効である

働き

連続軌跡モードでブロック変更を実行できない場合、このパートプログラムブロック内でプログラムされているすべての軸(ブロックをまたいで移動する付加軸を除く)が停止します。 この場合、輪郭誤差は発生しません。

加工中に軌跡軸を停止すると、ワーク表面にアンダーカットが発生する可能性があります。

5.10 データリスト

5.10.1 マシンデータ

5.10.1.1 一般マシンデータ

番号	識別子: $MN_	説明
10110	PLC_CYCLE_TIME_AVERAGE	平均 PLC 応答時間
10680	MIN_CONTOUR_SAMPLING_TIME	最小輪郭サンプリング時間
10682	CONTOUR_SAMPLING_FACTOR	輪郭サンプリング係数
10712	NC_USER_CODE_CONF_NAME_TAB	再設定された NC 命令のリスト
12030	OVR_FACTOR_FEEDRATE	軌跡送り速度オーバライドスイッチの評価
12100	OVR_FACTOR_LIMIT_BIN	バイナリコードによるオーバライドスイッチの制限
18360	MM_EXT_PROG_BUFFER_SIZE	外部ソース(DRAM)からの実行用 FIFO バッファサイズ

5.10.1.2 チャネルマシンデータ

番号	識別子: $MC_	説明
20150	GCODE_RESET_VALUES	グループの初期設定
20170	COMPRESS_BLOCK_PATH_LIMIT	圧縮できる NC ブロックの最大移動距離
20171	SURF_BLOCK_PATH_LIMIT	COMPSURF による圧縮のための NC ブロックの最大移動距離
20172	COMPRESS_VELO_TOL	圧縮用の軌跡速度からの最大許容誤差

番号	識別子: $MC_	説明
20173	SURF_VELO_TOL	COMPSURF による圧縮のための軌跡速度からの最大許容誤差
20400	LOOKAH_USE_VELO_NEXT_BLOCK	後続ブロックの速度の先読み
20430	LOOKAH_NUM_OVR_POINTS	先読み用のオーバライドスイッチ位置の数
20440	LOOKAH_OVR_POINTS	先読み用のオーバライドスイッチ点
20443	LOOKAH_FFORM	拡張先読みの起動
20450	LOOKAH_RELIEVE_BLOCK_CYCLE	ブロックサイクルタイムの緩和係数
20455	LOOKAH_FUNCTION_MASK	特殊先読み機能
20460	LOOKAH_SMOOTH_FACTOR	先読みのスムージング係数
20462	LOOKAH_SMOOTH_WITH_FEED	プログラムされた送り速度でのスムージング
20465	ADAPT_PATH_DYNAMIC	軌跡ダイナミック応答の調整
20480	SMOOTHING_MODE	G64x による丸み付け動作
20482	COMPRESSOR_MODE	コンプレッサの動作原理
20484	COMPRESSOR_PERFORMANCE	コンプレッサのパフォーマンス
20485	COMPRESS_SMOOTH_FACTOR	特定のダイナミックモードの COMPCAD による圧縮のスムージング係数
20486	COMPRESS_SPLINE_DEGREE	特定のダイナミックモードの COMPCAD による圧縮のスプライン度
20487	COMPRESS_SMOOTH_FACTOR_2	特定のダイナミックモードの COMPCAD による圧縮の回転軸のスムージング係数
20488	SPLINE_MODE	スプライン補間の設定
20490	IGNORE_OVL_FACTOR_FOR_ADIS	過負荷係数に影響されない G641/G642
20550	EXACT_POS_MODE	G0/G1 のイグザクトストップ条件
20552	EXACT_POS_MODE_G0_TO_G1	連続軌跡モードでの早送り遷移に対するイグザクトストップ条件
20560	G0_TOLERANCE_FACTOR	G0 許容範囲係数
20600	MAX_PATH_JERK	軌跡関連の最大加々速度
20602	CURV_EFFECT_ON_PATH_ACCEL	軌跡ダイナミック応答への軌跡曲率の影響
20603	CURV_EFFECT_ON_PATH_JERK	軌跡加々速度への軌跡曲率の影響
20606	PREPDYN_SMOOTHING_ON	曲率スムージングの起動
20730	G0_LINEAR_MODE	G0 の補間動作

番号	識別子: $MC_	説明
28060	MM_IPO_BUFFER_SIZE	IPO バッファの NC ブロックの数(DRAM)
28070	MM_NUM_BLOCKS_IN_PREP	ブロック解析用の NC ブロックの数(DRAM)
28071	MM_NUM_SURF_LEVELS	COMPSURF 機能のサイズ決め(DRAM)
28072	MM_MAXNUM_SURF_GROUPS	軸グループに関する COMPSURF 機能のサイズ決め(DRAM)
28520	MM_MAX_AXISPOLY_PER_BLOCK	ブロックごとの軸多項式の最大数
28530	MM_PATH_VELO_SEGMENTS	ブロックでの軌跡速度を制限するための記憶要素の数
28533	MM_LOOKAH_FFORM_UNITS	拡張先読み用のメモリ
28540	MM_ARCLENGTH_SEGMENTS	ブロックごとの円弧長機能表示用の記憶要素の数
28610	MM_PREPDYN_BLOCKS	速度解析のブロック数

5.10.1.3 軸/主軸別のマシンデータ

番号	識別子: $MA_	説明
32000	MAX_AX_VELO	最大軸速度
32310	MAX_ACCEL_OVL_FACTOR	軸速度の不連続変化に対応した過負荷係数
32431	MAX_AX_JERK	軌跡上の移動時の最大軸加々速度
32432	PATH_TRANS_JERK_LIM	連続軌跡モードのブロック遷移部での最大軸加々速度
32433	SOFT_ACCEL_FACTOR	SOFT の加速度制限の倍率
32434	G00_ACCEL_FACTOR	G00 に対応した加速度制限の倍率
32435	G00_JERK_FACTOR	G00 に対応した軸加々速度制限の倍率
32440	LOOKAH_FREQUENCY	先読みのスムージング限界周波数
33100	COMPRESS_POS_TOL	補正の最大誤差
33120	PATH_TRANS_POS_TOL	G645 による丸み付け時の最大誤差
35240	ACCEL_TYPE_DRIVE	軸の DRIVE 加速特性のオン/オフ
36000	STOP_LIMIT_COARSE	汎用イグザクトストップ
36010	STOP_LIMIT_FINE	精密イグザクトストップ

番号	識別子: $MA_	説明
36012	STOP_LIMIT_FACTOR	汎用/精密イグザクトストップと停止状態監視
36020	POSITIONING_TIME	精密イグザクトストップ遅延時間

5.10.2 セッティングデータ

5.10.2.1 チャネルセッティングデータ

番号	識別子$SC_	説明
42465	SMOOTH_CONTUR_TOL	丸み付け中の最大輪郭誤差
42466	SMOOTH_ORI_TOL	丸み付け中の工具オリエンテーションの最大誤差
42470	CRIT_SPLINE_ANGLE	COMPCAD による圧縮のコーナ角度制限
42471	MIN_CURV_RADIUS	COMPCAD による圧縮の曲率 の最小半径
42472	MIN_SURF_RADIUS	COMPSURF による圧縮の曲率 の最小半径
42473	ACTNUM_SURF_GROUPS	COMPSURF の軸グループの数
42475	COMPRESS_CONTUR_TOL	圧縮なしの最大許容輪郭誤差
42476	COMPRESS_ORI_TOL	圧縮の工具の向きの最大誤差 **注記:** 有効な方向座標変換の場合のみ。
42477	COMPRESS_ORI_ROT_TOL	圧縮の工具の回転の最大誤差 **注記:** 回転する工具を備えた 6 軸の機械のみ。
42676	ORI_SMOOTH_TOL	スムージング時の旋回のスムージング許容範囲
42678	ORISON_TOL	旋回のスムージング許容範囲

5.10.3 信号

5.10.3.1 チャネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
全軸停止	DB21、... .DBX36.3	DB330x.DBX4.3

5.10.3.2 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
汎用イグザクトストップ位置に到達	DB31、... .DBX60.6	DB390x.DBX0.6
精密イグザクトストップ位置に到達	DB31、… .DBX60.7	DB390x.DBX0.7

B2:加速度

6

6.1 概略説明

6.1.1 一般情報

機能の範囲

この機能説明では、次の各機能について説明します。

- 加減速
- 加々速度
- 膝形加減速特性

加速度と加々速度

軸およびチャネル別のプログラム可能な最大値、パートプログラムとシンクロナイズドアクションでプログラムできる加減速方法、およびダイナミック応答調整と制限を使用すれば、効果的な加速度と加々速度を、該当する機械と加工状況に最適に適合させられます。

膝形加減速特性

膝形加減速特性とは、速度に高い依存性のあるトルク特性を持ったモータ(特にステッピングモータ) が機械軸に装備されている場合、モータの最適負荷率を確保するのに必要なレベルに加速度を設定し、同時にモータを過負荷から保護します。

6.1.2 特徴

加速度

軸別の機能:

- プログラム可能な最大加速度値
- パートプログラムの命令により選択可能な加減速方法:
  加々速度制限のない加速(BRISKA)
- パートプログラム命令(ACC)を使用した最大値の設定

- プログラムした早送り(G00)に固有の最大値
- 加々速度制限が有効な移動に固有な最大値
- 接線の変化が不連続なブロック遷移の過剰加速

チャネル別の機能:

- パートプログラムの命令による選択可能な加減速方法:
  加々速度制限のない加速(BRISK)
- 過度に急激な加減速を回避するためのプログラム可能な一定の移動時間
- 重畳移動に対応したプログラム可能な加速マージン
- 調整可能な加速度制限
- 特定のリアルタイムイベントに対応した調整可能な加速度
- 法線方向加速度に対応したプログラム可能な加速マージン

加々速度

軸別の機能:

- パートプログラムの命令による選択可能な加減速方法:
  加々速度制限が指定された加速(SOFTA)
- 単独軸補間に対応したプログラム可能な最大加々速度値
- 軌跡補間に対応したプログラム可能な最大加々速度値

チャネル別の機能:

- パートプログラムの命令による選択可能な加減速方法:
  加々速度制限が指定された加速(SOFT)
- 調整可能な加々速度制限
- 特定のリアルタイムイベントに対応した調整可能な軌跡加々速度
- プログラムした早送り(G00)に固有な最大値
- 一定した曲率を持たないブロック遷移に対する過剰加々速度

膝形加減速特性

膝形加減速特性は、次の特性データを使用してパラメータ設定されます。

- 最大速度 v_{max}
- 最大加速度 a_{max}

- クリープ速度 v_{red}
- クリープ加速度 a_{red}
- 加速度逓減の特性(一定、双曲線、直線)

6.2 機能

6.2.1 加々速度制限のない加減速(BRISK/BRISKA) (チャネル/軸別)

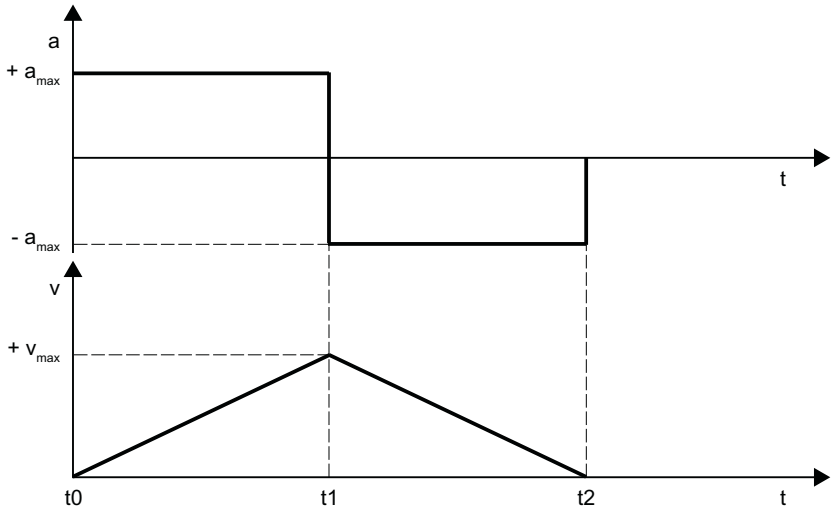
6.2.1.1 一般事項

一般事項

加々速度制限のない(加々速度=無限の)加減速では、加減速に最大値が直接適用されます。 加々速度制限のある加減速に関しては、次の点が異なります。

- **長所**
  速度と加速度に同じ最大値を使用した場合、処理時間が短くなります。
- **短所**
  機械の機構部にかかる負荷が増大し、高周波で制御が難しい機械的振動が発生する恐れがあります。

加減速方法



a_{max}: 最高加速度値

v_{max}: 最高速度値

t : 時間

図 6-1 階段状の加減速方法の速度と加速度の図

前述の図では、加減速方法の次の特徴が確認できます。

- 時間: t_0
  0 から$+a_{max}$ への急激な加速度
- 区間: t_0 - t_1
  $+a_{max}$ での一定の加速度、速度は直線的に増加します。
- 時間: t_1
  加速から減速への瞬間的な切り替えによる 2 * a_{max} の急激な加速度

  注記

  一定の速度期間を指定すれば、通常、この急激な加速度を避けることができます(「定速移動区間(チャネル別) (ページ 323)」の章を参照してください)。

- 区間: t1 - t2
  a_{max} での一定の加速度、速度は直線的に減速します。

6.2.1.2 パラメータ設定

軌跡移動の最高軸加速度

軌跡移動の最高軸加速度は、次のマシンデータを使用して各機械軸の特定の加工毎に指定できます。

MD32300 $MA_MAX_AX_ACCEL[<パラメータセットインデックス>]

<パラメータセットインデックス> = 0、1、2 ... (最大パラメータセット数 - 1)

加工別のパラメータセットについては、「軌跡補間のダイナミック応答モード (ページ 279)」の章を参照してください。

軌跡に関連している機械軸のパラメータ設定された最大値を超えないように、先読みの軌跡演算により軌跡パラメータが計算されます。

注記

特定の加工条件について、最大値を超える可能性があります(「加速度調整(ACC) (軸別) (ページ 325)」および「リアルタイムイベントの軌跡加速度(チャネル別) (ページ 329)」の章を参照)。

位置決め軸移動の最高軸加速度

位置決め軸の移動では、設定された位置決め軸のダイナミック応答モードに応じて、次の 2 つの最大値のいずれかが有効になります。

- MD32300 $MA_MAX_AX_ACCEL [0] (DYNNORM ダイナミック応答モード DYNNORM での軌跡移動に対する最大軸加速度)
- MD32300 $MA_MAX_AX_ACCEL [1] (DYNPOS ダイナミック応答モード DYNNORM での軌跡移動に対する最大軸加速度)

位置決め軸ダイナミック応答モードは、NC 別マシンデータで設定されます。

MD18960 $MN_POS_DYN_MODE = <モード>

<モード>	意味
0	有効な最大軸加速度:MD32300 $MA_MAX_AX_ACCEL[0]
1	有効な最大軸加速度:MD32300 $MA_MAX_AX_ACCEL[1]

JOG 移動の最高軸加速度

「JOG 移動の加速度と加々速度 (ページ 362)」の章を参照してください。

6.2.1.3 プログラミング

加々速度制限のない軌跡加減速(BRISK)

構文

```
BRISK
```

動作

BRISK パートプログラム命令は、軌跡加減速のために「加々速度制限のない」加減速方法を選択するのに使用されます。

G グループ:21

効果:モーダル

リセット応答

チャネル別の初期設定は、リセットにより有効になります。

MD20150 $MC_GCODE_RESET_VALUES[20]

必要条件

加工中にパートプログラム内で加減速方法(BRISK/SOFT)が変更されると、ブロック終点でイグザクトストップが実行されます。

加々速度制限のない単独軸加減速(BRISKA)

構文

BRISKA (*軸{,軸}*)

機能

BRISKA パートプログラム命令は、単独軸移動(JOG、JOG/INC、位置決め軸、揺動軸など)に「加々速度制限のない」加減速方法を選択するのに使用されます。

G グループ:-

有効性: モーダル

軸:

- 数値の範囲: チャネル軸の軸名称

軸別の初期設定

加々速度制限のない加減速は、単独軸移動の軸別初期設定として設定できます。

MD32420 $MA_JOG_AND_POS_JERK_ENABLE = FALSE

リセット応答

軸別の初期設定は、リセットにより有効になります。

MD32420 $MA_JOG_AND_POS_ENABLE

6.2.2 定速移動区間(チャネル別)

6.2.2.1 一般事項

一覧

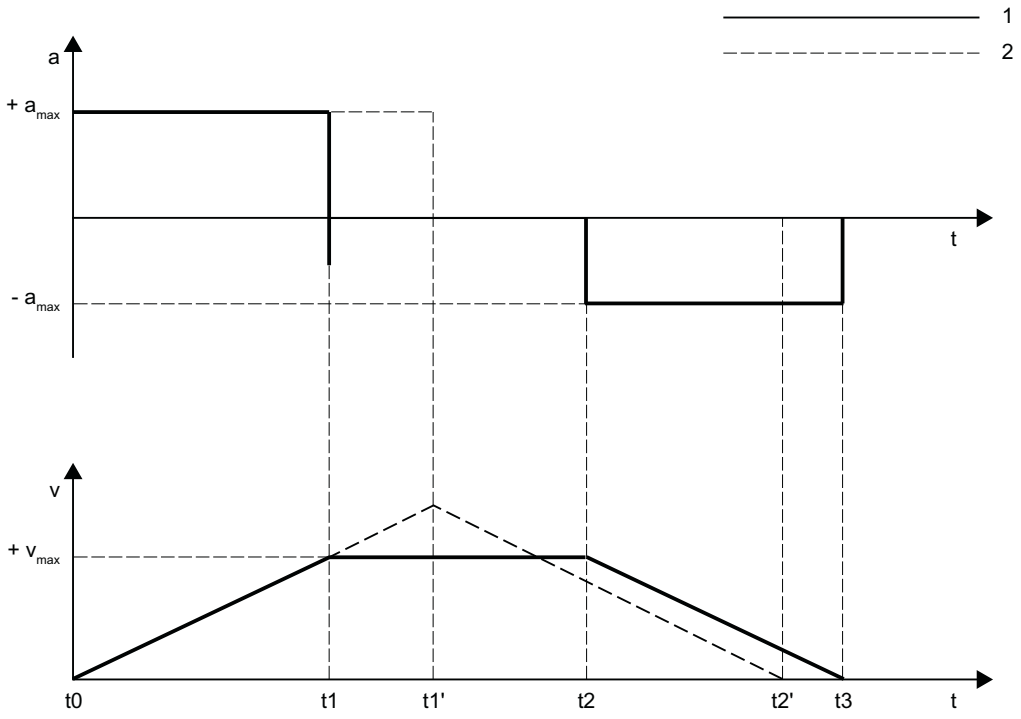
変化率制限のない加減速の場合は、加速から減速に切り替わるときに、2 * a_{max} の急激な加速が発生します。この急激な加速が発生しないようにするには、チャネル別の定速移動時間を設定します。一定の移動時間により、加速段階と減速段階の間の定速移動の時間が指定されます。

MD20500 $MC_CONST_VELO_MIN_TIME (定速移動の最短時間)

注記

定速移動時間は、以下の場合には機能しません。

- 次の機能が有効なとき:先読み
- 補間周期以下の移動時間の移動ブロック内



1: 定速移動時間の場合の曲線

2: 定速移動時間でない場合の曲線

a_{max}: 最高加速度値

v_{max}: 最高速度値

t : 時間

図 6-2 瞬時加速度の原理特性

上記の図は、定速移動時間の効果を示しています。

- 時点:t_1
  急激な加速度 1 * a_{max} が指定された加速段階の終わり
- 区間:t1 - t2
  加速度 0、パラメータ設定された定速移動時間の間の一定速度
- 時点:t_2
  急激な加速度 1 * a_{max} が指定された減速段階の開始

時点 t_0、t_1'、および t_2'は、定速移動時間でない場合の結果となる、関連付けられた曲線を示します。

6.2.2.2 パラメータ設定

定速移動区間は、次のマシンデータを使用して特定チャネル毎にパラメータ設定されます。

MD20500 $MC_CONST_VELO_MIN_TIME
定速移動の最短時間)

6.2.3 加速度調整(ACC) (軸別)

6.2.3.1 一般事項

機能

加速度別のマシンデータで設定された、現在有効な最大軸加速度は、ACC 命令を使用して、特定の軸に関して低下させることができます。この場合、 命令のプログラミング時に指定されるパーセンテージ係数の形式で低下させます。

たとえば、加工セグメントで最大許容軸加速度を低下させることで、大きな加工力の結果として、機械的な振動を防止できます。

有効性

ACC による加速度の低下は、AUTOMATIC および MDI 運転モードのすべての補間タイプ、およびドライラン送り速度に対して**有効**です。

加速度の低下は、次の場合は**無効です**。

- JOG モードの場合
- 機械機能 REF (レファレンス点復帰)の場合
- 検出された異常の結果として軸が急停止(セットポイント= 0)で停止した場合

6.2.3.2 プログラミング

構文

ACC[<軸>]=<低下係数>

ACC[SPI(<主軸番号>)]=<低下係数>

ACC(S<主軸番号>)=<低下係数>

意味

<ACC>:	現在の最大許容加速度を軸別に低下させるための、マシンデータからの命令	
	有効性:	モーダル
<軸>:	軌跡軸のチャネル軸名称	
	データタイプ:	AXIS
	値の範囲:	チャンネル軸名称
SPI(<主軸番号>):	SPI(...)機能は、主軸番号を、対応するチャネル軸名に変換します。	
S<主軸番号>:	チャネルの主軸の名前	

詳細情報

システム変数

チャネルで現在有効な加速度の低下(ACC を使用して設定)は、以下を使用して、アクシス毎に読み取ることができます。

$AA_ACC[<軸>]

リセット動作

ACC を使用して設定した加速度の低下は、チャネルリセット後、またはプログラムの終了後も保持できます。パラメータ設定は以下によっておこなわれます。

MD32320 $MA_DYN_LIMIT_RESET_MASK

6.2.4 加速マージン(チャネル別)

6.2.4.1 一般事項

一般事項

通常、事前解析では、軌跡加速度のためにパラメータ設定された機械軸の最大値が最大限使用されます。 たとえば、「輪郭からの高速リトラクト」機能のときの重畳移動に備えて、加速マージンを確保できるようにするために、変更可能な係数により軌跡加速度を減らすことができます。 たとえば、係数 0.2 を適用する場合、事前解析では、最大許容軌跡加速度の 80%だけを使用します。 20%は、重畳移動用加速マージンとしてとっておきます。

6.2.4.2 パラメータ設定

加速マージンのパラメータは、次のマシンデータによりチャネル毎に割り付けられます。
MD20610 $MC_ADD_MOVE_ACCEL_RESERVE
(重畳動作の加速マージン)

6.2.5 軌跡加速度制限(チャネル別)

6.2.5.1 一般事項

一般事項

該当する加工状況に柔軟に対応できるように、下記のセッティングデータを使用して事前解析中に計算される軌跡加速度をチャネル別に制限できます。

SD42500 $SC_SD_MAX_PATH_ACCEL (最大軌跡加速度)

セッティングデータで指定した値を考慮するのは、その値が、事前解析中に計算した軌跡加速度より小さい場合に限ります。

下記のセッティングデータを使用して、チャネル別に制限を有効にしてください。

SD42502 $SC_IS_SD_MAX_PATH_ACCEL = TRUE

6.2.5.2 パラメータ設定

次のセッティングデータを使用して、チャネル別のパラメータ設定が実行されます。

SD42500 $SC_SD_MAX_PATH_ACCEL (最高軌跡加速度)

SD42502 $SC_IS_SD_MAX_PATH_ACCEL (軌跡加速度制限の起動)

6.2.5.3 プログラミング

制限値

構文

```
$SC_SD_MAX_PATH_ACCEL = 制限値
```

動作

セッティングデータをプログラムすれば、状況に応じて軌跡加速度制限を調整できます。

制限値:

- 数値の範囲:≥ 0
- 単位:m/s^2

使用方法:

- パートプログラム
- 静的シンクロナイズドアクション

起動/解除

構文

```
$SC_IS_SD_MAX_PATH_ACCEL = 値
```

動作

このセッティングデータをプログラムすれば、軌跡加速度制限を起動/解除できます。

パラメータ:*値*

- 数値の範囲: TRUE、FALSE

使用方法:

- パートプログラム
- 静的シンクロナイズドアクション

6.2.6 リアルタイムイベントの軌跡加速度(チャネル別)

6.2.6.1 一般事項

一般事項

一方の加工を優先した加速度と、もう一方で、下記のリアルタイムイベントを組み合わせた時間優先の加速度が一切妥協しないように、

- NC ストップ/NC スタート
- 送り速度オーバライドの変更
- 「Safely Integrated」機能の使用時の「安全な速度低減」のための速度基準値の変更

上述のリアルタイムイベントの場合、チャネル別のシステム変数を使用して、軌跡加速度を設定できます。

$AC_PATHACC = *軌跡加速度*

リアルタイムイベント加速度が有効になるのは、上記リアルタイムイベントの 1 つに関する速度の変更中に限ります。

制限

指定された軌跡加速度が、その軌跡に関連する機械軸の性能を超えると、コントローラ内の軌跡加速度に制限が課せられて、得られる軸加速度(a_{res})が、パラメータに設定された軸の最大値(a_{max})の 2 倍未満に制限されます。

$a_{res} = 2 * a_{max,}$ ここで、a_{max} = MD32300 $MA_MAX_AX_ACCEL

注記

リアルタイムイベントの軌跡加速度は、法線方向加速度度に関係なく有効です。

有効性

有効	次のリアルタイムイベントに関して、AUTOMATIC と MDI の運転モードの場合に限り、リアルタイムイベント軌跡加速度が有効です。 ● NC ストップ/NC スタート ● オーバライドの変更 ● 「Safely Integrated」機能の使用時の「安全な速度低減」のための速度基準値の変更
無効	輪郭曲率、コーナー、キネマティック座標変換の制限などような、チャネルの事前解析中の軌跡演算に起因する軌跡速度の変更に対しては、リアルタイムイベントの軌跡加速度は無効です。 プログラム指令値が、該当する軌跡部分の事前解析中に計算された軌跡加速度より小さい場合、リアルタイムイベント軌跡加速度は無効です。

プログラミング

パートプログラムまたはシンクロナイズドアクション内でのシステム変数のプログラミングについては、「プログラミング (ページ 330)」の章を参照してください。

6.2.6.2 プログラミング

構文

```
$AC_PATHACC = 軌跡加速度
```

動作

リアルタイムイベント軌跡加速度は、チャネル別のシステム変数を使用して設定されます。

パラメータ:*軌跡加速度*

- 数値の範囲: 軌跡加速度≥ 0
- 単位:m/s²

解除: `$AC_PATHACC = 0`

使用方法:

- パートプログラム
- 静的シンクロナイズドアクション

リセット応答

リアルタイムイベント軌跡加速度は、リセットすると解除されます。

必要条件

パートプログラム内に$AC_PATHACC をプログラムすると、REORG による事前解析停止 (STOPRE)が自動的に起動されます。

6.2.7 早送り(G00)がプログラムされたのときの加速度(軸別)

6.2.7.1 一般事項

加工処理にかかわる機械軸の加速度は、機械の性能により公式に許容される加速度より低く設定しなくてはならないことがよくあります。それは、当該の特定処理に関連した必要条件がある場合です。

早送り(パートプログラム命令 G00)がプログラムされた機械軸の時間を優先した移動の場合、軸別の加速度に特定の最大値をプログラムできます。

JOG セットアップモード

この機能は、JOG セットアップモードの早送りオーバライドに関しては加速度に影響を及ぼしません。

6.2.7.2 パラメータ設定

早送り(G00)がプログラムされたときの軸別加速度の最大値は、次の軸マシンデータを使用してパラメータ設定されます。

MD32434 $MA_G00_ACCEL_FACTOR
(G00 のときの加速制限の倍率)

この設定を使用すれば、事前解析中の軌跡演算コンポーネントにより考慮される、早送り(G00)がプログラムされたときの軸別加速度の最大値を生成できます。

加速度[軸] =

MD32300 $MA_MAX_AX_ACCEL * MD32434 $MA_G00_ACCEL_FACTOR

6.2.8 加々速度制限が有効な加速度(SOFT/SOFTA) (軸別)

6.2.8.1 一般事項

機能

加々速度制限なしの加減速と比較すると、加々速度制限が指定された加減速では、同じ最大加速度値を使用しても、ある程度、時間を消費します。 この時間消費を補償するために、加々速度制限(SOFT/SOFTA)が有効になっている機械軸の移動のあいだ、軸別の加速度に特定の最大値をプログラムできます。

加々速度制限が有効になっている加速度の最大値は、軸別の最大値に関連して計算される係数を使用してパラメータ設定されます。 この設定を使用すれば、事前解析中の軌跡演算コンポーネントにより考慮される、加々速度制限が有効にされた軸別加速度の最大値を生成できます。

加速度[軸] =

MD32300 $MA_MAX_AX_ACCEL * MD32433 $MA_SOFT_ACCEL_FACTOR

6.2.8.2 パラメータ設定

加々速度制限(SOFT/SOFTA)が有効になった加速度の最大値は、次の軸マシンデータを使用してパラメータ設定されます。

MD32434 $MA_SOFT_ACCEL_FACTOR
(SOFT による加速度制限の倍率)

6.2.9 接線の変化が不連続なブロック遷移の過剰加速度(軸別)

6.2.9.1 一般事項

機能

接線の変化が不連続なブロック遷移(コーナー)の場合、パラメータ設定した軸応答性を確実に順守するために、制御装置でジオメトリ軸を大幅に減速することが必要になることがあります。 接線の変化が不連続なブロック遷移に関する減速を軽減/回避するために、より高いレベルの軸別加速度を有効にできます。

過剰加速度は、軸別の最大値に関連して計算される係数を使用してパラメータ設定されます。 この設定を使用すれば、事前解析中の軌跡演算コンポーネントにより考慮される、接線の変化が不連続なブロック遷移部の軸別加速度の最大値を生成できます。

加速度[軸] =

MD32300 $MA_MAX_AX_ACCEL * MD32310 $MA_MAX_ACCEL_OVL_FACTOR

6.2.9.2 パラメータ設定

接線の変化が不連続なブロック遷移の過剰加速度は、次の軸マシンデータを使用してパラメータ設定されます。

MD32310 $MA_MAX_ACCEL_OVL_FACTOR
(速度の不連続変化に対する過負荷係数)。

6.2.10 法線方向加速度に対応した加速マージン(チャネル別)

6.2.10.1 一般事項

概要

軌跡加速度(接線加速度)の他に、法線方向加速度も、輪郭の曲線部に影響を及ぼします。 軌跡パラメータの設定中にこのことを考慮しないと、輪郭曲線部での加速と減速中に、有効な軸加速度が少しの間、最大値の 2 倍に到達します。

有効な軸加速=

軌跡加速度+法線方向加速度=

2 * (MD32300 $MA_MAX_AX_ACCEL)

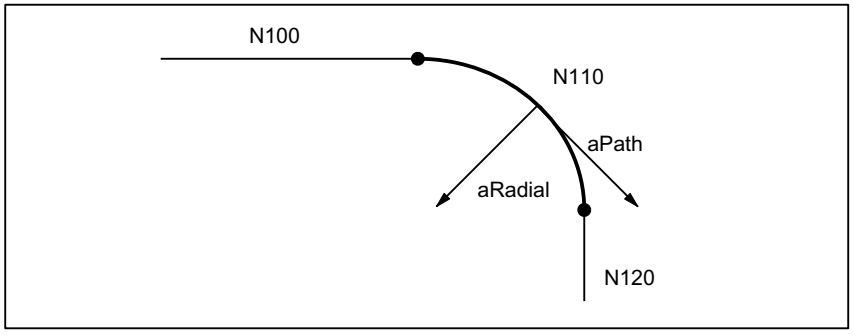


図 6-3 輪郭の曲線部での法線方向加速度と軌跡加速度

チャネルマシンデータ:
MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL
(ダイナミック軌跡応答への軌跡曲率の影響)
を使用すれば、法線方向加速度を考慮した軸別加速度の比率を設定できます。

たとえば、値 0.75 が使用される場合、法線方向加速度には軸別加速度の 75%を、軌跡加速度には軸別加速度の 25%を使用できます。

対応する最大値は、通常、次のように計算されます。

法線方向加速度=

MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL * MD32300 $MA_MAX_AX_ACCEL

軌跡加速度=

(1 - MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL) * MD32300 $MA_MAX_AX_ACCEL

例

次のマシンパラメータを使用します。

- すべてのジオメトリ軸に対して MD32300 $MA_MAX_AX_ACCEL:3 m/s
- 機械の機構上の制約に起因する軌跡半径 10 mm のときの最大軌跡速度: 5m/min

法線方向加速度は、次のように計算されます。

$$A_{Radial} = \frac{v^2_{パス}\ [m/min]}{r\ [mm] * 3.6\ [m/s^2]} = \frac{5^2}{10 * 3.6} = 0.694\ m/s^2$$

加速マージンは、次のように設定されます。

$$\text{MD20602 \$MC_CURV_EFFECT_ON_PATH_ACCEL} = \frac{a_{Radial}\ [m/s^2]}{\text{MD32300 \$MA_MAX_AX_ACCEL}\ [m/s^2]} = \frac{0.694}{3} \approx 0.23$$

直線動作

キネマティックトランスフォーメーションが有効でない直線動作(直線補間)の場合、上記加速マージンは無効です。

6.2.10.2 パラメータ設定

輪郭の曲線部の法線方向加速度に応じて加速マージンとして考慮される使用できる最高軸加速度の比率は、次のチャネルマシンデータを使用してパラメータ設定されます。

MD20602 $MC_CURV_EFFECT_ON_PATH_ACCEL
(ダイナミック軌跡応答への軌跡曲率の影響)

6.2.11 軌跡補間のときの加々速度制限(SOFT) (チャネル別)

6.2.11.1 一般事項

概要

この文書の残りの部分に記載されている機能が関連する限り、一定の加速度、つまり加々速度一定加減速(加々速度=有限値)が指定されている加速度が、想定される加減速方法になります。 加々速度制限が指定された加速の場合、直線補間は、0 から最大値までの加速に関して適用されます。

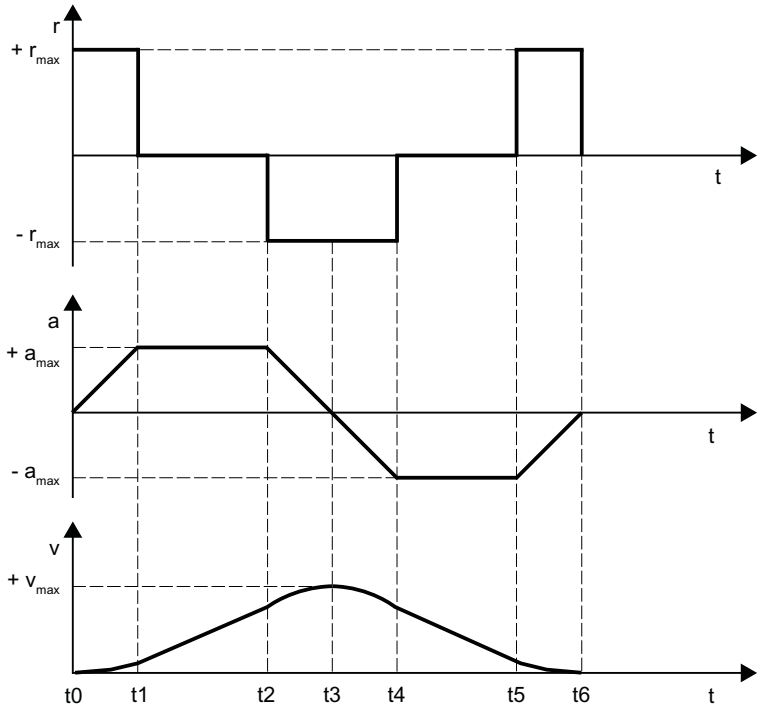
長所

機械の機構部にかかる負荷が最小になり、一定の過剰な加速度による高周波および制御が難しい機械的振動が発生する恐れが低くなります。

短所

同じ最大速度値と加速度値を使用する場合、ステップ状の加速波形と比べて加工時間が長くなります。

加速波形



r_{max}: 最大加々速度値

a_{max}: 最大加速度値

v_{max}: 最大速度値

t : 時間

図 6-4 加々速度制限加速波形が指定された加々速度、加速度および速度の図

前述の図では、加減速方法の次の特徴が確認できます。

- 区間: t_0 - t_1
  $+r_{max}$ が指定された一定の加々速度、加速度の直線的増加、速度の二次関数的増加
- 区間: t1 - t2
  $+a_{max}$ での一定の加速、速度は直線的に増加します。
- 区間: t_2 - t_3
  $-r_{max}$ が指定された一定の加々速度、加速度の直線的減少、最大値$+v_{max}$ に到達するまでの速度増加の二次関数的減少
- 区間: t_3 - t_4
  $-r_{max}$ が指定された一定の加々速度、減速加速度の直線的増加、速度の二次関数的減少

- 区間: t_4 - t_5
  $-a_{max}$ が指定された一定減速加速度、速度の直線的減少
- 区間: t_5 - t_6
  $+r_{max}$ が指定された一定の加々速度、減速加速度の直線的減少、ゼロ速度 v = 0 に到達するまでの速度減少の二次関数的減少

6.2.11.2 パラメータ設定

軌跡移動の最高加々速度値(軸別)

軌跡移動の最高軸加々速度は、次のマシンデータを使用して各機械軸の特定の加工毎に指定できます。

MD32431 $MA_MAX_AX_JERK[<パラメータセットインデックス>]

<パラメータセットインデックス> = 0, 1, 2 ... (最大パラメータセット数 - 1)の場合

加工別のパラメータセットについては、「軌跡補間のダイナミック応答モード (ページ 279)」の章を参照してください。

軌跡に関連している機械軸のパラメータ設定された最大値を超えないように、先読みの軌跡演算により軌跡パラメータが計算されます。

注記

特定の加工条件について、最大値を超える可能性があります(「リアルタイムイベントの軌跡加々速度(チャネル別) (ページ 343)」の章を参照してください)。

軌跡移動の最高加々速度値(チャネル別)

軸別設定に加えて、最高加々速度値も次のマシンデータを使用して、チャネル別軌跡パラメータとして指定することができます。

MD20600 $MC_MAX_PATH_JERK (軌跡関連の最高加々速度)

軸別とチャネル別の最高加々速度値が互いに干渉しないように、チャネル別の最大値を、最大軸値より大きい値に設定してください。

6.2.11.3 プログラミング

構文

SOFT

動作

SOFT パートプログラム命令を使用すれば、チャネル内のジオメトリ軸の移動操作に、加々速度制限が指定された加減速方法を選択できます。

G グループ: 21

効果: モーダル

リセット応答

チャネル別の初期設定は、リセットにより有効になります。

MD20150 $MC_GCODE_RESET_VALUES[20]

境界条件

加工中にパートプログラム内で加速モードが変更されると(BRISK ↔ SOFT)、連続軌跡モードのときでも、ブロック終点でイグザクトストップ付きの遷移の瞬間にブロックの変更が実行されます。

6.2.12 単独軸補間の加々速度制限(SOFTA) (軸別)

6.2.12.1 パラメータ設定

軸別の加々速度制限の初期設定

加々速度制限が指定された加減速は、軸別初期設定として設定できます。

MD32420 $MA_JOG_AND_POS_JERK_ENABLE== TRUE

位置決め軸移動の最高軸別加々速度

加々速度制限が有効な位置決め軸の移動時に、次に示すマシンデータのいずれかの値が最高軸別加々速度として有効になります。

- MD32430 $MA_JOG_AND_POS_MAX_JERK (最高軸別加々速度)
- MD32431 $MA_MAX_AX_JERK [0] (ダイナミック応答モード DYNNORM の軌跡移動の最大軸別加々速度)
- MD32431 $MA_MAX_AX_JERK [1] (ダイナミック応答モード DYNPOS での軌跡移動の最高軸別加々速度)

使用されるマシンデータは、設定された位置決め軸のダイナミック応答モードによって決定されます。

MD18960 $MN_POS_DYN_MODE = <モード>

<モード>	意味
	位置決め軸の移動の最高軸別加々速度として、以下が適用されます。
0	MD32430 $MA_JOG_AND_POS_MAX_JERK G75 が有効の場合(固定点アプローチ):MD32431 $MA_MAX_AX_JERK[**0**]
1	MD32431 $MA_MAX_AX_JERK[**1**]

JOG 移動の最高軸加々速度

「JOG 移動の加速度と加々速度 (ページ 362)」の章を参照してください。

6.2.12.2 プログラミング

構文

SOFTA (*Axis* {*Axis*})

機能

SOFTA パートプログラム命令は、単独軸移動(位置決め軸、揺動軸など)に対して加々速度一定加減速を選択するのに使用されます。

G グループ:-

有効性:モーダル

軸:

- 数値の範囲: チャネル軸の軸名称

軸別の初期設定

加々速度制限が指定された加減速は、単独軸移動の軸別初期設定として設定できます。

MD32420 $MA_JOG_AND_POS_JERK_ENABLE = TRUE

リセット応答

軸別の初期設定は、リセットにより有効になります。

MD32420 $MA_JOG_AND_POS_ENABLE

6.2.13 軌跡加々速度制限(チャネル別)

6.2.13.1 一般事項

概要

該当する加工状況に柔軟に対応できるように、下記のセッティングデータを使用して事前解析中に計算される軌跡加々速度をチャネル別に制限できます。

SD42510 $SC_SD_MAX_PATH_JERK (最大軌跡加々速度)

セッティングデータで指定した値をチャネルで考慮するのは、その値が、事前解析中に計算した軌跡加々速度より小さい場合に限ります。

下記のセッティングデータを使用して、チャネル別に制限を有効にしてください。

SD42512 $SC_IS_SD_MAX_PATH_JERK = TRUE

6.2.13.2 パラメータ設定

次のセッティングデータを使用して、チャネル別のパラメータ設定が実行されます。

SD42510 $SC_SD_MAX_PATH_JERK (最高軌跡加々速度)

SD42512 $SC_IS_SD_MAX_PATH_JERK
(軌跡加々速度制限の起動)

6.2.13.3 プログラミング

最大軌跡加々速度

構文

$SC_SD_MAX_PATH_JERK = *加々速度値*

動作

このセッティングデータをプログラムすれば、状況に応じて軌跡加々速度制限を調整できます。

加々速度値:

- 数値の範囲:≥ 0
- 単位:m/s^3

使用方法:

- パートプログラム
- 静的シンクロナイズドアクション

起動/解除

構文

$SC_IS_SD_MAX_PATH_JERK = *値*

動作

このセッティングデータをプログラムすれば、軌跡加々速度制限の起動/解除ができます。

パラメータ:*値*

- 数値の範囲: TRUE、FALSE

使用方法:

- パートプログラム
- 静的シンクロナイズドアクション

6.2.14 リアルタイムイベントの軌跡加々速度(チャネル別)

6.2.14.1 一般事項

概要

加工を優先した加々速度と、

- NC ストップ/NC スタート
- 送り速度オーバライドの変更
- 「Safely Integrated」機能の使用時の「安全な速度低減」のための速度基準値の変更

前述のリアルタイムイベントの場合、チャネル別のシステム変数を使用して、軌跡加々速度を設定できます。

$AC_PATHJERK = *軌跡加々速度*

リアルタイムイベントの軌跡加々速度が有効になるのは、上記リアルタイムイベントの 1 つに関する速度の変更中に限ります。

制限

加々速度はドライブと関連のある物理変数ではないため、設定される加々速度には制限が**課せられません**。

有効性

有効	次のリアルタイムイベントに関して、AUTOMATIC と MDI の運転モードの場合に限り、リアルタイムイベントの軌跡加々速度が有効です。 ● NC ストップ/NC スタート ● オーバライドの変更 ● 「Safely Integrated」機能の使用時の「安全な速度低減」のための速度基準値の変更
無効	輪郭曲率、コーナー、キネマティック座標変換の制限などような、チャネルの事前解析中の軌跡演算に起因する軌跡速度の変更に対しては、リアルタイムイベントの軌跡加々速度は無効です。 プログラム指令値が、該当する軌跡部分の事前解析中に計算された軌跡加々速度より小さい場合、リアルタイムイベント軌跡加々速度は無効です。

プログラミング

加速度に応じたリアルタイムイベントの加々速度を設定するために、システム変数を次のように設定できます。

$AC_PATHJERK = $AC_PATHACC/スムージング時間

- $AC_PATHACC: 軌跡加速度[m/s²]
  スムージング時間: 0.02 秒など、自由に選択可能

パートプログラムまたはシンクロナイズドアクション内でのシステム変数のプログラミングについては、「プログラミング (ページ 344)」の章を参照してください。

6.2.14.2 プログラミング

構文

$AC_PATHJERK = *軌跡加々速度*

機能

リアルタイムイベントの軌跡加々速度は、チャネル別のシステム変数を介して設定されます。

加々速度値:

- 数値の範囲: 軌跡加々速度 ≥ 0
- 単位: m/s^3

用途:

- パートプログラム
- 静的シンクロナイズドアクション

リセット応答

この機能はリセット時に解除されます。

境界条件

パートプログラム内に$AC_PATHJERK をプログラムすると、REORG による事前解析停止(STOPRE)が自動的に起動されます。

6.2.15 早送り(G00)がプログラムされたのときの加々速度(軸別)

6.2.15.1 一般事項

概要

加工処理にかかわる機械軸の加々速度は、機械の性能により公式に許容される加速度より低く設定しなくてはならないことがよくあります。それは、当該の特定処理に関連した必要条件がある場合です。

早送り(パートプログラム命令 G00)がプログラムされた機械軸の時間を優先した移動の場合、軸別の加々速度に特定の最大値をプログラムできます。

JOG セットアップモード

この機能は、JOG セットアップモードの早送りオーバライドに関しては加々速度に影響を及ぼしません。

6.2.15.2 パラメータ設定

早送り(G00)がプログラム指令されたときの軸別加々速度の最大値は、次の軸マシンデータを使用してパラメータ設定されます。

MD32434 $MA_G00_ACCEL_FACTOR
(G00 による加速度制限の倍率)

この設定を使用すれば、事前解析中の軌跡演算要素により考慮される、早送り(G00)がプログラム指令されたときの軸別加々速度の最大値を生成できます。

加々速度[軸] =
MD32431 $MA_MAX_AX_JERK * MD32435 $MA_G00_JERK_FACTOR

6.2.16 一定した曲率を持たないブロック遷移に対する過剰加々速度(軸別)

6.2.16.1 一般事項

概要

一定した曲率を持たないブロック遷移(直線> 円弧など)の場合、パラメータ設定した軸応答性に確実に準拠するために、制御装置でジオメトリ軸の移動を大幅に減速する必要があります。 一定した曲率を持たないブロック遷移に関する減速を軽減/回避するために、より高レベルの軸別加々速度を有効にできます。

過剰加々速度は、専用の軸別最大値を使用してパラメータ設定できます。

6.2.16.2 パラメータ設定

一定した曲率を持たないブロック遷移に対する過剰加々速度は、次の軸マシンデータを使用してパラメータ設定されます。

MD32432 $MA_PATH_TRANS_JERK_LIM
(一定した曲率を持たないブロック遷移に対する過剰加々速度)

6.2.17 速度依存加々速度調整(軸別)

機能

ダイナミック軌跡応答は、軌跡に関連する軸の速度、加速度、および加々速度に対してパラメータ設定された一定の軸最大値から得られます。

- MD32000 $MA_MAX_AX_VELO (最高軸速度)
- MD32300 $MA_MAX_AX_ACCEL (最高軸加速度)
- MD32431 $MA_MAX_AX_JERK (軌跡移動の最高軸加々速度)

自由曲面との接続時などのように**曲率が一定しない輪郭** (軸ねじれ)の場合、軌跡速度の変動、特に速度上限での変動は、主として軸加々速度が原因で発生します。 軌跡速度の変動は、面粗度に悪影響をもたらします。

軌跡速度に及ぼす軸加々速度の影響は、許容軸加々速度を速度に依存して上げることで、曲率が一定していない輪郭に対して小さくすることができます。 軌跡速度の変動は、適切なパラメータ設定によって防止することができます。

許容軸加々速度を速度依存で上げることは、最高許容軌跡加速度と軌跡加々速度には影響しません。 これらは、加々速度調整が適用されている場合でも、マシンデータでパラメータ設定されている一定の軸最大値から得られます。

直線移動の場合は曲率と軸ねじれの両方がゼロであるため、速度依存加々速度調整は直線移動には影響しません。

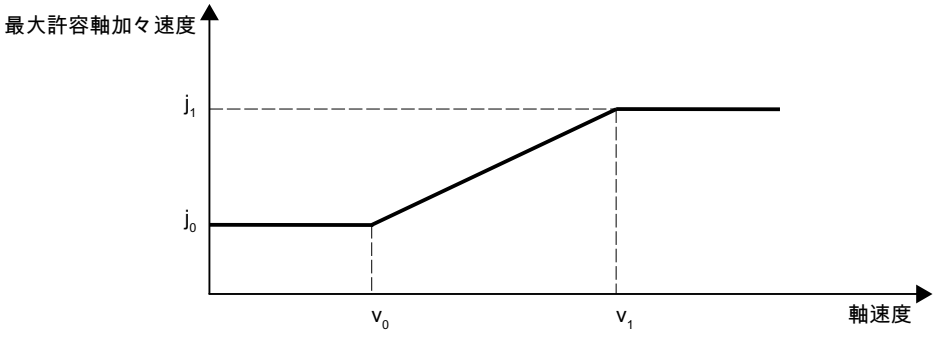
適用

「速度依存加々速度調整」機能は、機能「自由形式面モード:基本機能 (ページ 281)」とは無関係に使用可能です。

パラメータ設定

「速度依存加々速度調整」機能は、次のマシンデータを使用してパラメータ設定します。

- MD32437 $MA_AX_JERK_VEL0[<n>] = <しきい値$_{下限}$>
  加々速度調整の速度下限しきい値 速度依存加々速度調整は、この速度で有効になります。
  速度下限しきい値は、ダイナミック応答モード毎にインデックス n で個別に設定できます(「軌跡補間のダイナミック応答モード (ページ 279)」の章を参照してください)。
- MD32438 $MA_AX_JERK_VEL1[<n>] = <しきい値$_{上限}$>
  加々速度調整の速度上限しきい値 速度依存加々速度は、この速度で、 MD32439 $MA_MAX_AX_JERK_FACTOR でパラメータ設定された 最大値 j_{max} に達します。
  速度上限しきい値は、ダイナミック応答モード毎にインデックス n で個別に設定できます(「軌跡補間のダイナミック応答モード (ページ 279)」の章を参照してください)。
- MD32439 $MA_MAX_AX_JERK_FACTOR = <係数>
  速度上限しきい値 MD32438 $MA_AX_JERK_VEL1[<n>]に達する速度依存の最高加々速度 j_{max} のパラメータ設定のための係数
  j_{max} = (MD32431 $MA_MAX_AX_JERK) * (MD32439 $MA_MAX_AX_JERK_FACTOR)
  速度依存加々速度調整は、値 > 1.0 の場合に有効になります。
  速度依存加々速度調整は、値 = 1.0 の場合は無効です。



v_0: MD32437 $MA_AX_JERK_VEL0

v_1: MD32438 $MA_AX_JERK_VEL1

j_0: MD32431 $MA_MAX_AX_JERK

j_1: MD32439 $MA_MAX_AX_JERK_FACTOR * MD32431 $MA_MAX_AX_JERK

図 6-5　軸速度に応じた軸加々速度

注記

速度依存加々速度調整は、以下の場合にだけ有効になります。

MD32439 $MA_MAX_AX_JERK_FACTOR > 1.0

例

パラメータ設定の例:

- MD32437 $MA_AX_JERK_VEL0 = 3000 mm/min
- MD32438 $MA_AX_JERK_VEL1 = 6000 mm/min
- MD32439 $MA_MAX_AX_JERK_FACTOR[AX**1**] = 2.0
- MD32439 $MA_MAX_AX_JERK_FACTOR[AX**2**] = 3.0
- MD32439 $MA_MAX_AX_JERK_FACTOR[AX**3**] = 1.0

働き

- 速度依存加々速度調整は、1 番目と 2 番目の軸に対して有効になりますが、3 番目の軸に対しては有効になりません。
- パラメータ設定された加々速度は、軸速度が 0 ～ 3000 mm/min の範囲にある場合に有効です。
- 最高加々速度は、軸速度が 3000 mm/min ～ 6000 mm/min の範囲にある場合は、直線的に増加します。
- 1 番目の軸の最高許容加々速度は、軸速度が 6000 mm/min よりも大きい場合、係数 2 で増加します - 2 番目の軸の場合は係数 3 で増加します。
- パラメータ設定された値は、ダイナミック応答モード毎に適用されます。

6.2.18 加々速度フィルタ(軸別)

6.2.18.1 一般事項

概要

自由曲面のフライス加工など、特定用途の場合は、機械軸の位置指令値特性を滑らかにすると有益な場合があります。この場合、機械で発生する機械的振動が低減されるため、加工面品質を改善できます。

機械軸の位置指令値特性をならすため、補間器レベルで考慮されるチャネル別と軸別の加々速度制限にかかわりなく、位置コントローラレベルで加々速度フィルタを有効にできます。

加々速度フィルタの効果は、輪郭の精度に許容できない影響を及ぼさない範囲で、可能な限り強くなくてはいけません。また、フィルタのスムージング効果は可能な限り「バ

ランスのとれた」ものにします。つまり、同じ輪郭を前後に移動する場合、フィルタによってならされる輪郭は、両方向で可能な限り同じにします。

加々速度フィルタを機械条件に最大限に一致させることができるよう、いくつかのフィルタモードが使用可能です。

- 2 次フィルタ(PT2)
- 移動平均値の生成
- 帯域消去フィルタ

モード:2 次フィルタ

2 次フィルタは単純なローパスフィルタであるという事実により、「2 次フィルタ」モードは比較的小さいフィルタ時定数(約 10 ms)が関与する場合にのみ上記の要件に一致できます。これより大きい時定数と組み合わせて使用すると、許容できない輪郭のずれがすぐに現れます。フィルタの効果は比較的限定されます。

このフィルタモードは、非常に大きなフィルタ時定数が必要であり、輪郭精度には二次的な重要性しかない場合(位置決め軸など)に利点があります。

互換性の理由により、このフィルタモードが初期設定として設定されます。

モード:移動平均値の生成

輪郭のずれを極力小さく抑える必要がある場合は、「移動平均値生成」のフィルタモードを使用して、フィルタ時定数を 20 ～ 40 ms の範囲内に設定できます。スムージング効果はほぼ対称です。

計算されたサーボゲイン係数(K_V)の操作画面での表示は、フィルタに通常は適切な値より小さい値が表示されます。実際には、輪郭精度は、表示されたサーボゲイン係数(K_V)フィルタで提示されている値より高くなっています。

したがって、「2 次フィルタ」から「移動平均値生成」のフィルタモードに変更する場合、輪郭精度が向上しても、表示されるサーボゲイン係数(K_V)が(まったく同じフィルタ時定数で)低下することがあります。

モード:帯域消去フィルタ

帯域消去フィルタは、分子と分母の観点から見た 2 次フィルタです。

$$H(s) = \frac{\frac{s^2}{(2 * \pi * f_Z)^2} + \frac{2 * s * D_Z}{(2 * \pi * f_Z)}}{\frac{s^2}{(2 * \pi * f_N)^2} + \frac{2 * s * D_N}{2 * \pi * f_N}}$$

各記号の意味は次のとおりです。

f_Z: 分子の固有周波数

f_N: 分母の固有周波数

D_Z: 分子の減衰

D_N: 分母の減衰

振動に対応するフィルタ設定では、どのような場合にも、加々速度フィルタの「2 次フィルタ」(PT2)ローパスフィルタ(PT2)モードのように有用な結果が得られることは期待されないため、分母の減衰 D_N の設定オプションはありません。分母の減衰 D_N は、1 に固定されます。

帯域消去フィルタは、2 つの異なる方法でパラメータ設定できます。

- 実帯域消去フィルタ
- 高周波での振幅応答をさらに増加/低下させる帯域消去フィルタ

実帯域消去フィルタ

実帯域消去フィルタは、分子と分母でまったく同じ固有周波数が選択された場合に適用されます。

$f_Z = f_N = f_{block}$(ブロック周波数)

分子の減衰設定 = 0 が選択された場合、ブロック周波数は完全な減衰と同等です。この場合、3 dB の帯域幅は次のように計算されます。

$f_{3\ dB\ bandwidth} = 2 * f_{block}$

完全な減衰ではなく、k の因子による低減だけが必要な場合は、k に応じて分子の減衰を選択します。この場合、上記の 3 dB の帯域幅の計算式は適用されなくなります。

高周波での振幅応答をさらに増加/低下させる帯域消去フィルタ

この場合、分子と分母の固有周波数が異なる値に設定されます。分子の固有周波数によってブロック周波数が決定されます。

分子の固有周波数より低い/高い分母の固有周波数を選択すると、高周波での振幅応答を増加/低下させることができます。制御下のシステムでは、一般にそれ自体にローパス特性がある、つまり高周波ではいずれにしても振幅応答が低下するため、高周波での振幅応答の増加はほとんどの場合正当です。

必要条件

選択された分子の固有周波数が高すぎると、フィルタが無効になります。この場合、限界周波数 f_{Zmax} は位置制御周期に依存します。

$$f_{Zmax} = \frac{1}{2 * \pi * T_{Zmin}} = \frac{1}{2 * \pi * T_{位置制御周期}} \text{ (シャノンの定理)}$$

6.2.18.2 パラメータ設定

起動

加々速度フィルタは次のマシンデータを使用して有効なります。

MD32400 $MA_AX_JERK_ENABLE (軸の加々速度制限)

加々速度フィルタはすべての運転モードとすべての補間タイプで有効です。

フィルタモード

フィルタモードは、次のマシンデータを使用して選択されます。

MD32402 $MA_AX_JERK_MODE

値	フィルタモード
1	2 次フィルタ
2	移動平均値の生成
3	帯域消去フィルタ

時定数

軸加々速度フィルタの時定数 は、次のマシンデータで設定されます。

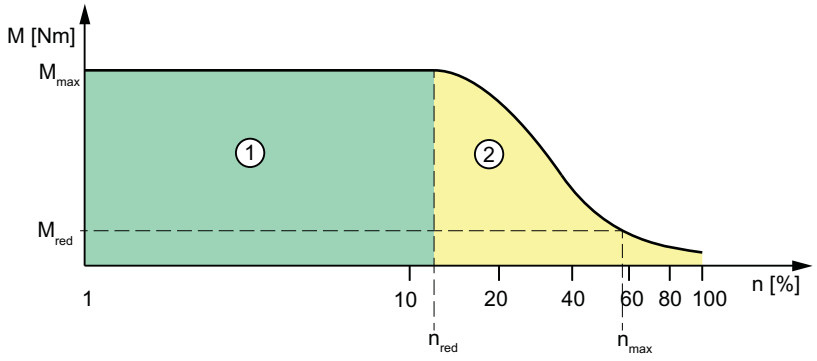
MD32410 $MA_AX_JERK_TIME

加々速度フィルタは、時定数が位置制御周期よりも大きい場合にのみ有効になります。

6.2.19 膝形加減速特性曲線

6.2.19.1 機能:モータ特性曲線の補正

いくつかのタイプのモータ、特にステッピングモータには、速度に大きく依存し、高い速度領域でトルクの急激な低下を示すトルク特性を持つものがあります。モータ特性の最適負荷率のために、関連する NC 軸の加速度を特定の速度以上でてい減する必要があります。



① 通常の範囲

② てい減される範囲

n_{red} トルクをてい減する最小移動速度

n_{max} 最高回転速度

M_{max} 最大トルク

M_{red} n_{max} でのトルク(加速度低下に対応)

特性曲線タイプ

モータのトルク特性曲線のコントローラ内部エミュレーションとして、てい減される範囲で以下の特性曲線タイプを選択できます(章「パラメータ設定 (ページ 357)」を参照):

- 一定トルク特性
- 双曲線トルク特性
- リニアトルク特性

6.2.19.2 機能:軌跡加速度に対する影響

軌跡加速度特性曲線は、軌跡に関連する軸の特性曲線のタイプに基づきます。特性曲線のタイプの異なる軸が一緒に補間される場合、軌跡加速度の加減速方法は、最も制限された低減タイプに基づいて決定されます。

次の優先順位が適用されます。ここで 1 = 最優先です。

1. 加速度低減:0 = 固定特性
2. 加速度低減:1 = 双曲線特性
3. 加速度低減:2 = 直線特性
4. 有効な加速度低減なし
   有効な加速度低減のない状況は、たとえば次のような場合に発生します。
   MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT = 1
   および/または
   MD35230 $MA_ACCEL_REDUCTION_FACTOR = 0

注記

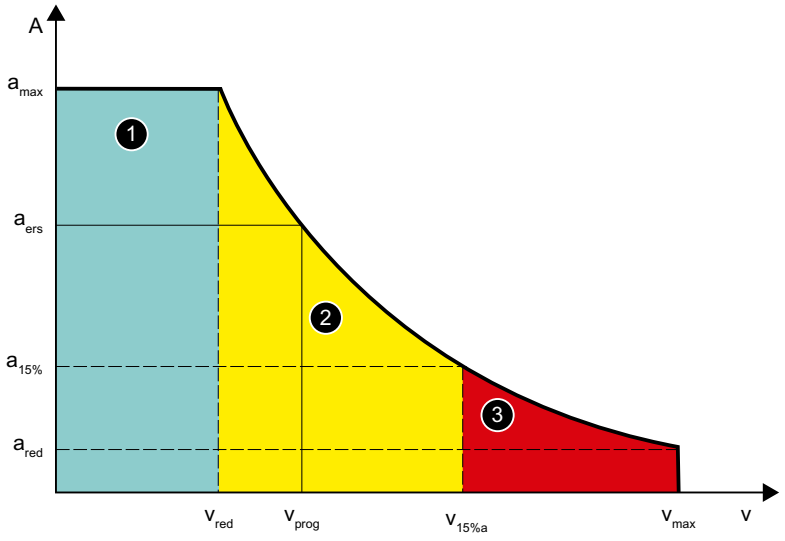
ステッピングモータと DC ドライブを備えた機械軸を一緒に補間できます。

6.2.19.3 機能:代替曲線

キネマティックトランスフォーメーションおよび加速度タイプ BRISK または SOFT が有効な場合などのように、プログラム指令された軌跡を、関連軸のマシンデータで設定された加速度曲線で移動できない場合は、ダイナミック制限値を削減して、代わりの曲線が生成されます。削減されるダイナミック制限値は、代替特性曲線によって最大速度と一定加速度の間で可能な最善の妥協が得られるように計算されます。

直線軌跡部分の代替特性曲線

この値の制限は、プログラムされた軌跡速度が、最大加速能力の 15 %が依然として使用可能な軌跡速度($v_{15\%a}$)より大きい場合に適用されます。その結果、加工状況にかかわりなく、最大加速能力/モータトルクの 15 %は常に使用可能になります。



① 通常の範囲

② 削減される範囲

③ ロック領域

a_{ers}: 代替特性曲線の一定加速

$a_{15\%}$ 最小一定加速

$a_{15\%} = 0.15 * (a_{max} - a_{red}) + a_{red}$

v_{ers} 代替特性曲線の速度

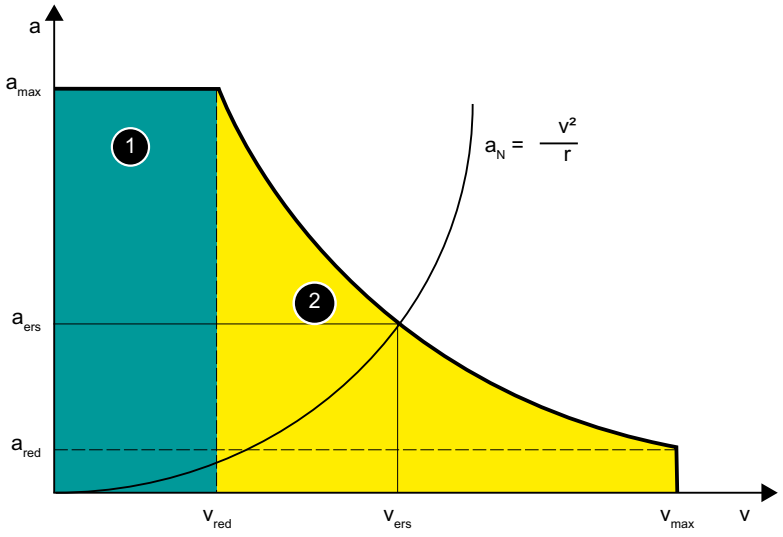
v_{prog} プログラムされた速度

$v_{15\%a}$ $a_{15\%}$時の速度

図 6-6 代替軌跡特性曲線:直線軌跡

曲線軌跡部の代替特性曲線

曲線軌跡部の場合、法線加速度と接線加速度は一緒に考慮します。速度に依存する軸の加速能力の最大 25 %のみが法線加速度に必要となるよう、軌跡速度が低減されます。加速能力の残り 75 %は、接線加速度、つまり軌跡上の減速/加速のために確保されます。



① 通常の範囲

② 削減される範囲

a_N 法線加速度

a_{ers}: 代替特性曲線の一定加速

v_{ers} 代替特性曲線の速度

R 軌跡半径

図 6-7 代替軌跡特性曲線:曲線軌跡

連続軌跡モードでのブロック遷移

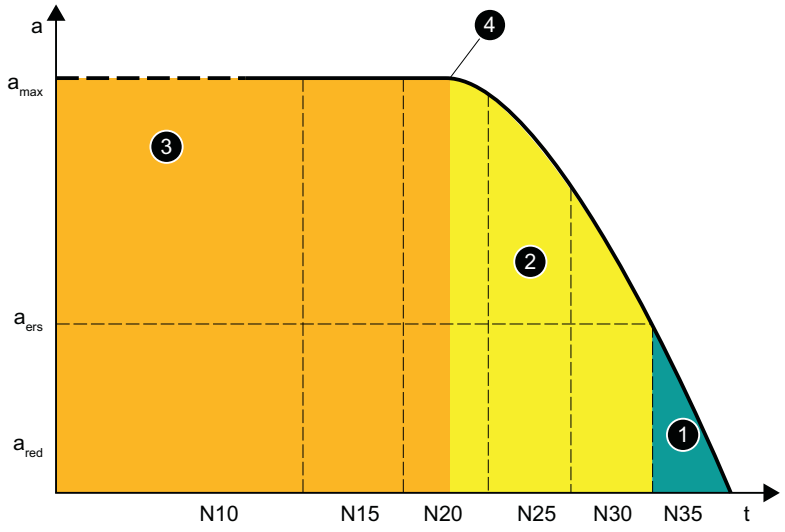
連続軌跡モードが有効な場合、接線の変化が不連続なブロック遷移は、プログラムされた軌跡速度が移動に使用される場合に、軸速度の不連続変化につながります。

その結果、ブロック遷移時にいずれの軸速度成分もクリープ速度 v_{red} を超えないように軌跡速度が制御されます。

連続軌跡モードおよび先読みによる減速カーブ

短い軌跡の連続した移動ブロックの場合は、加速または減速動作が複数のパートプログラムブロックにまたがることがあります。

このような状況では、パラメータ設定された速度依存加速度特性も「先読み」機能で考慮されます。



① 通常の範囲⇒ $a = a_{max}$

② 削減される範囲⇒ $a < a_{max}$

③ 定速移動範囲⇒ $a = 0\ m/s^2$

④ 減速適用点

v_{red}: クリープ速度

v_{max} 最大速度

NX ブロック番号 Nx の移動ブロック

図 6-8 先読みによる減速

6.2.19.4 パラメータ設定

軌跡軸

パラメータの割り付け

軸の移動動作の加速度をてい減する軸ごとの速度の下限は、以下のマシンデータによってパラメータ設定します:

MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT = <speed limit>

単独軸

起動

一軸の移動動作の加速度をてい減する軸ごとの速度の下限は、以下のマシンデータによって起動します:

MD35240 $MA_ACCEL_TYPE_DRIVE = TRUE

パラメータの割り付け

以下のマシンデータは、設定された速度制限を超過する軸ごとの加速度特性曲線のパラメータ設定に関連するものです:

- MD32000 $MA_MAX_AX_VELO (最高軸速度)
- MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT = <speed limit>
  パラメータ設定された速度から、加速度てい減が有効になります。
- MD35230 $MA_ACCEL_REDUCTION_FACTOR = <reduction factor>
  てい減速度を超えると、最大加速度と最小加速度間の補間が最高速度まで実行されます。最小加速度は、次のように計算されます:
  最小加速度 = 最大加速度 * (1 - てい減係数)
- MD35242 $MA_ACCEL_REDUCTION_TYPE (以下の段落を参照してください:「トルク特性曲線の選択」)
- MD32300 $MA_MAX_AX_ACCEL (最高軸加速度)

トルク特性曲線の選択

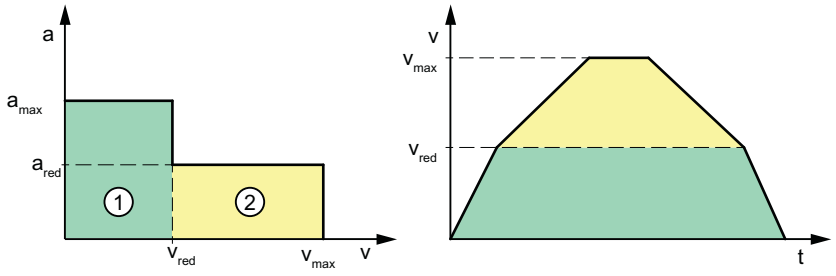
モータのトルク特性曲線のコントローラ内部エミュレーションとして、マシンデータによって、てい減される範囲で以下の特性曲線タイプを選択できます:

MD35242 $MA_ACCEL_REDUCTION_TYPE = <value>

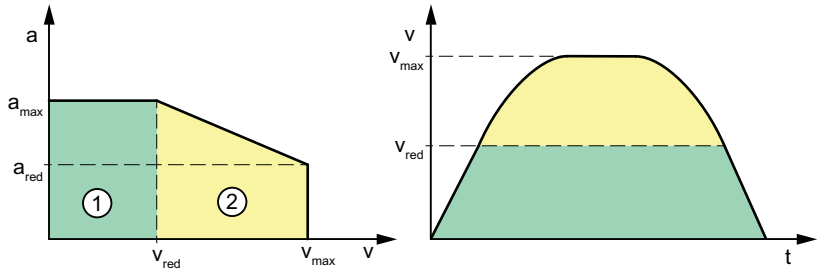
値	意味
0	一定トルク特性
1	双曲線トルク特性
2	リニアトルク特性

各特性曲線タイプでの主な速度および加減速特性曲線を次の図に示します。

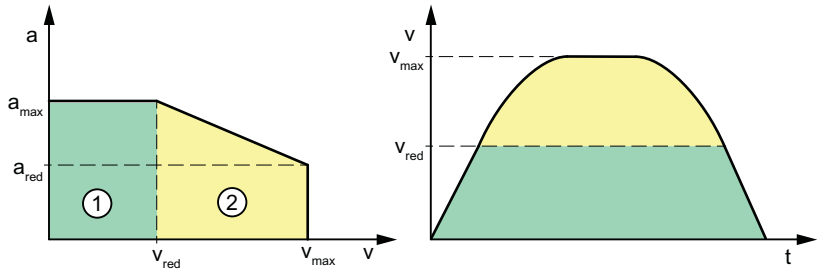
- 一定トルク特性(値 = 0)



- 双曲線トルク特性(値 = 1)



- リニアトルク特性(値 = 2)



特性パラメータ

この特性曲線パラメータは、以下のマシンデータから得られます:

① 通常の範囲

② てい減される範囲

v_{max} $MA_MAX_AX_VELO

v_{red} $MA_ACCEL_REDUCTION_SPEED_POINT * $MA_MAX_AX_VELO

a_{max} $MA_MAX_AX_ACCEL

a_{red} (1 - $MA_ACCEL_REDUCTION_FACTOR) * $MA_MAX_AX_ACCEL

6.2.19.5 プログラミングチャネル別の起動(DRIVE)

機能

軸の移動動作で加速度てい減が適用される軸ごとの速度の下限は、命令 DRIVE により有効になります。

構文

```
DRIVE
```

意味

DRIVE:	パラメータ設定された速度下限を超えた場合のチャンネル別の加速度てい減の有効化	
	G グループ:	21
	基本設定:	MD20150 $MC_GCODE_RESET_VALUES[20]
	有効性:	モーダル

必要条件

特性の切替

DRIVE のプログラミングにより加速度てい減が有効であり、SOFT または BRISK によって加速度方法が変更されている場合、削減された動的な制限値を含む代替特性で移動が実行されます。

DRIVE の再プログラミングにより、加速度のてい減は、パラメータ設定された速度制限を超えて再度有効にできます。

下記も参照

パラメータ設定 (ページ 357)

6.2.19.6 プログラミング軸別の起動(DRIVEA)

機能

一軸の移動動作の加速度てい減が適用される軸ごとの速度の下限が、事前定義された手順 DRIVEA()により、プログラムされた軸に対して有効になります。

構文

```
DRIVEA(<Axis_1>, <Axis_2>, ...)
```

意味

DRIVEA:	パラメータ設定された膝形加速度特性曲線を軸ごとに有効にします。	
	有効性:	モーダル
<軸_x>:	パラメータ設定された膝形加速度特性曲線を有効にする軸。	
	データタイプ:	AXIS
	数値の範囲:	チャネル軸名称

必要条件

膝形加減速特性曲線が軸に対してパラメータ設定されている場合は、これが移動操作の初期設定の加減速方法になります。

SOFTA または BRISKA パートプログラム命令を使用して、特定の軸に対して有効な加減速方法が変更されると、膝形加減速特性曲線の代わりに適切な代替特性曲線が使用されます。

DRIVEA をプログラミングして、特定の軸について膝形加減速特性曲線に戻すことができます。

6.2.19.7 境界条件

単独軸補間

単独軸補間(位置決め軸、振動軸、手動動作など)の場合に膝形加減速特性曲線を有効にすると、機械軸はモード DRIVEA のみで移動します。

次のパートプログラム命令による加減速方法の切り替えはできません。

- 瞬時加速度変更(BRISKA)
- 加々速度制限が指定された加速度(SOFTA)

パス補間

プログラムされた軌跡に関与する機械軸について、パートプログラム命令 DRIVE なしでパラメータ設定された膝形加減速特性曲線が有効な場合、応答性制限値の低減された代替特性曲線がその軌跡に対して特定されます。

キネマティックトランスフォーメーション

膝形加減速特性曲線は、有効なキネマティックトランスフォーメーションでは考慮されません。内部制御により、加々速度制限のない加速(BRISK) への切り替えが行われ、軌跡加速度に対して代替特性曲線が有効になります。

6.2.20 JOG 移動の加速度と加々速度

軸別加速度および加々速度の制限値は、JOG モードでも有効になります。

また、ジオメトリ軸と旋回軸の手動での移動に対して、加速度と加々速度をチャネル別に制限することも可能です。これにより、回転軸で全ての直交移動を生成するキネマティックの処理を設定できます。(ロボット)

6.2.20.1 パラメータ設定

軸別の加速度と加々速度の制限

最高軸別加速度

軸の手動移動の最高加速度は、次のマシンデータで定義します。

MD32300 $MA_MAX_AX_ACCEL [0] (ダイナミック応答モード DYNNORM での軌跡移動の最高軸別加速度)

注記

JOG モードでは、常にダイナミック応答モード DYNNORM のみが有効です。

「加々速度制限のない加減速(BRISK/BRISKA) (チャネル/軸別) (ページ 319)」の章も参照してください。

軸別の加々速度制限の初期設定

加々速度制限が指定された加減速は、JOG モードの軸別初期設定として指定できます。

MD32420 $MA_JOG_AND_POS_JERK_ENABLE== TRUE

最高軸別加々速度

加々速度制限が有効な JOG 移動の場合、次のマシンデータの値が最高軸別加々速度として有効になります。

MD32430 $MA_JOG_AND_POS_MAX_JERK (最高軸別加々速度)

「単独軸補間の加々速度制限(SOFTA) (軸別) (ページ 339)」の章も参照してください。

チャネル別の加速度と加々速度の制限

ジオメトリ軸を手動で移動する場合の最高加速度

ジオメトリ軸を手動で移動する場合の最高加速度は、次のマシンデータを使用してチャネル毎に指定することができます。

MD21166 $MC_JOG_ACCEL_GEO [<ジオメトリ軸>]

<ジオメトリ軸> = 0、1、2

注記

MD21166 $MC_JOG_ACCEL_GEO [<ジオメトリ軸>]の場合、MD32300 $MA_MAX_AX_ACCEL に対する直接の制限はありません。

注記

座標変換が有効の場合、MD32300 $MA_MAX_AX_ACCEL によって最高許容軸別加速度が決まります。

ジオメトリ軸の手動移動時の最大加々速度

SOFT 加減速モード(加々速度一定加減速)でジオメトリ軸を手動で移動する場合の最高加々速度は、次のマシンデータを使用してチャネル毎に指定することができます。

MD21168 $MC_JOG_JERK_GEO [<ジオメトリ軸>]

<ジオメトリ軸> = 0、1、2

注記

MD21168 $MC_JOG_JERK_GEO は、JOG モードの軸別加々速度制限が基本機械軸に対して有効になっている場合にのみ作用します。
MD32420 $MA_JOG_AND_POS_JERK_ENABLE [<軸>] == TRUE

MD21168 = 0 の場合、チャネル別の加々速度制限の代わりに、MD32430 $MA_JOG_AND_POS_MAX_JERK の軸別制限値が有効になります。

旋回軸の手動移動時の最大加々速度

次のマシンデータによって、各チャネルについて、手動で旋回軸を移動時の最高加々速度を指定できます。

MD21158 $MC_JOG_JERK_ORI [<旋回軸>]

MD21158 を有効にするには、次のマシンデータによって、旋回軸の手動移動のチャネル別加々速度制限を有効にしてください。

MD21159 $MC_JOG_JERK_ORI_ENABLE == TRUE

6.2.20.2 必要条件

有効な回転によってジオメトリ軸を手動で移動する場合の動作

SOFT 加減速モード(加々速度一定加減速)でジオメトリ軸を手動で移動する場合、MD32430 $MA_JOG_AND_POS_MAX_JERK の値は、有効な回転または有効な旋回工具ホルダでも使用されます。

パートプログラム命令 SOFTA/BRISKA/DRIVEA

パートプログラム命令 `SOFTA(<axis1>,<axis2>, ...)` は、JOG モードでも有効です。つまり、MD32430 $MA_JOG_AND_POS_MAX_JERK の最高軸別加々速度は、JOG モードでの移動時に、指定された軸に対して有効です(MD32420 $MA_JOG_AND_POS_JERK_ENABLE [<軸>] == TRUE を設定した場合とまったく同じです)。

注記

一方、SOFT パートプログラム命令は JOG モードでは無効になります。

SOFTA と同様に、パートプログラム命令 BRISKA と DRIVEA は JOG モードでも有効です。つまり、MD32420 $MA_JOG_AND_POS_JERK_ENABLE が関連する機械軸に対して「TRUE」に設定されている場合でも、加速度に加々速度制限はつきません。

注記

旋回軸の手動移動は、BRISKA/SOFTA/DRIVEA に影響されません。

下記も参照

加々速度制限のない加減速(BRISK/BRISKA) (チャネル/軸別) (ページ 319)

単独軸補間の加々速度制限(SOFTA) (軸別) (ページ 339)

6.3 例

6.3.1 加速度

6.3.1.1 軌跡速度特性

以下の例は、プログラム指令された移動動作とパートプログラムセグメントで開始された操作に基づく軌跡速度特性を示します。

パートプログラムの抜粋

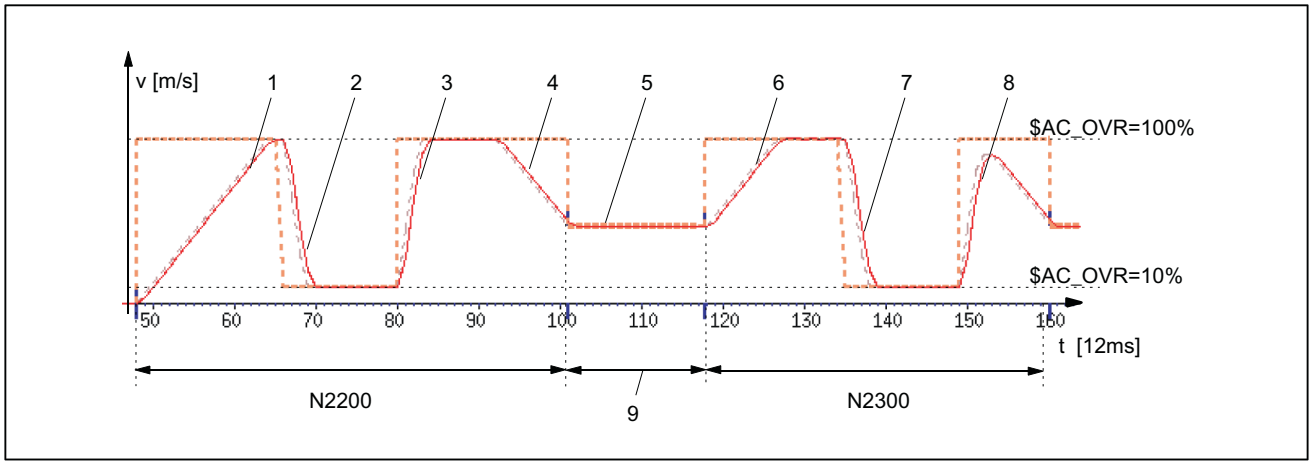
プログラムコード

```
; シンクロナイズドアクション:高速入力 1 ($A_IN[1])に依存する加速の切り替え:
N53 ID=1 WHEN $A_IN[1] == 1 DO $AC_PATHACC = 2.*$MA_MAX_AX_ACCEL[X]
; シンクロナイズドアクション:テストオーバーライドパターン(外部介入のシミュレーション):
N54 ID=2 WHENEVER ($AC_TIMEC > 16) DO $AC_OVR=10
N55 ID=3 WHENEVER ($AC_TIMEC > 30) DO $AC_OVR=100
;アプローチ
N1000 G0 X0 Y0 BRISK
N1100 TRANS Y=-50
N1200 AROT Z=30 G642
; 輪郭
N2100 X0 Y0
N2200 X = 70 G1 F10000 RNDM=10 ACC[X]=30 ACC[Y]=30
N2300 Y = 70
```

プログラムコード

```
N2400 X0
N2500 Y0
```

軌跡速度特性



加減速方法:BRISK

1: 加速度の初期設定 ACC に従って、軌跡速度(F10000)の 100%まで加速します。(N2200...)

2: リアルタイム加速度$AC_PATHACC に従って、オーバライド修正($AC_OVR)の結果として軌跡速度の 10%まで減速します。(N53/N54...)

3: リアルタイム加速度 $AC_PATHACC (N53/N55...)に従って、オーバライド修正($AC_OVR)の結果として軌跡速度の 100%まで加速します。

4: 加速度の初期設定 ACC に従って、中間スムージングブロックに対してブロックの最終速度まで減速します。(N2200...)

5: スムージングの結果としての速度制限(9 参照)

6: 加速どの初期設定 ACC に従って、軌跡速度($AC_OVR)の 100%まで加速します:(N2300...)

7: リアルタイム加速度$AC_PATHACC (N53/N54...)に応じた加速率で、オーバライド修正の結果として減速します。

8: リアルタイム加速度 $AC_PATHACC (N53/N55...)に従って、オーバライド修正($AC_OVR)の結果として軌跡速度の 100%まで加速します。

9: プログラム指令された丸み付け(RNDM) (N2200...)の結果として制御に挿入される中間ブロック

図 6-9 事前解析時に指定された軌跡加速度とリアルタイム加速度との切り替え

6.3.2 加々速度

6.3.2.1 軌跡速度特性

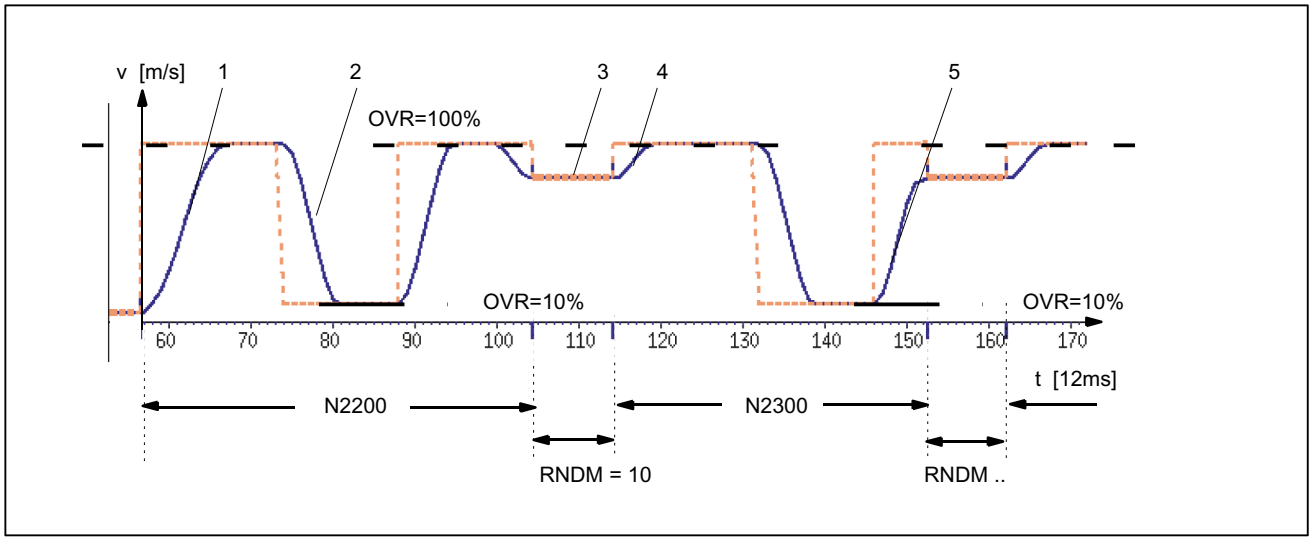
以下の例は、プログラム指令された移動動作とパートプログラムセグメントで開始された操作に基づく軌跡速度と加々速度の特性を示します。

パートプログラムの抜粋

プログラムコード

```
; 外部介入発生時の軌跡加速度と軌跡加々速度の設定:
N0100 $AC_PATHACC = 0.0
N0200 $AC_PATHJERK = 4.0 * ($MA_MAX_AX_JERK[X] + $MA_MAX_AX_JERK[Y]) / 2.0
; シンクロナイズドアクション:オーバーライドの変更 (外部介入のシミュレーション)
N0530 ID=1 WHENEVER ($AC_TIMEC > 16) DO $AC_OVR=10
N0540 ID=2 WHENEVER ($AC_TIMEC > 30) DO $AC_OVR=100
;アプローチ
N1000 G0 X0 Y0 SOFT
N1100 TRANS Y=-50
N1200 AROT Z=30 G642
; 輪郭
N2100 X0 Y0
N2200 X = 70 G1 F10000 RNDM=10
N2300 Y = 70
N2400 X0
N2500 Y0
```

軌跡速度と加々速度の特性



加減速方法:SOFT

1: $MA_MAX_AX_JERK[..]に応じた加々速度

2: $AC_PATHJERK に応じた加々速度

3: $MA_MAX_AX_JERK[..]に応じた加々速度(ブロックの最終速度へのアプローチ)

4: 円弧を使用した速度制限

5: $AC_PATHJERK に応じた加々速度

図 6-10 事前解析時に指定された軌跡加々速度と、$AC_PATHJERK との切り替え

6.3.3 加速度と加々速度

以下の例は、パートプログラム抜粋のプログラム指令された移動動作に基づく、X 軸の速度と加速度の特性を示します。さらに、速度と加速度に関連するマシンデータのどれが、輪郭のどのセクションを決定するかを示します。

パートプログラムの抜粋

プログラムコード	コメント
N90 F5000 SOFT G64	; 連続軌跡モード、加々速度制限された加速度
N100 G0 X0 Y0 Z0	; 早送り
N110 G1 X10	;直線
N120 G3 CR=5 X15 Y5	; 円弧、半径 5 mm、ブロック遷移:接線の変化が連続的
N130 G3 CR=10 X5 Y15	; 円弧、半径 10 mm、ブロック遷移:接線の変化が連続的
N140 G1 X-5 Y17.679	; 直線、15°のねじれ

輪郭

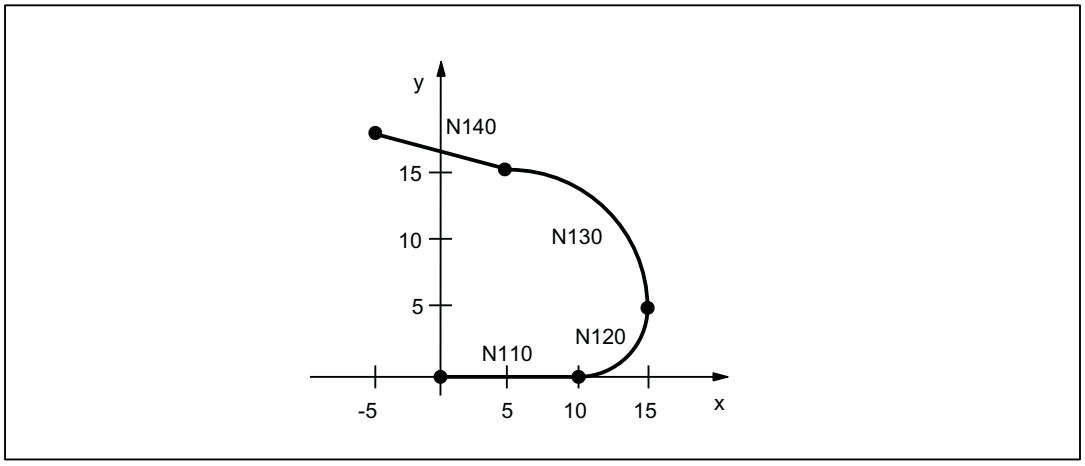


図 6-11　パートプログラム抜粋の輪郭

速度と加減速特性

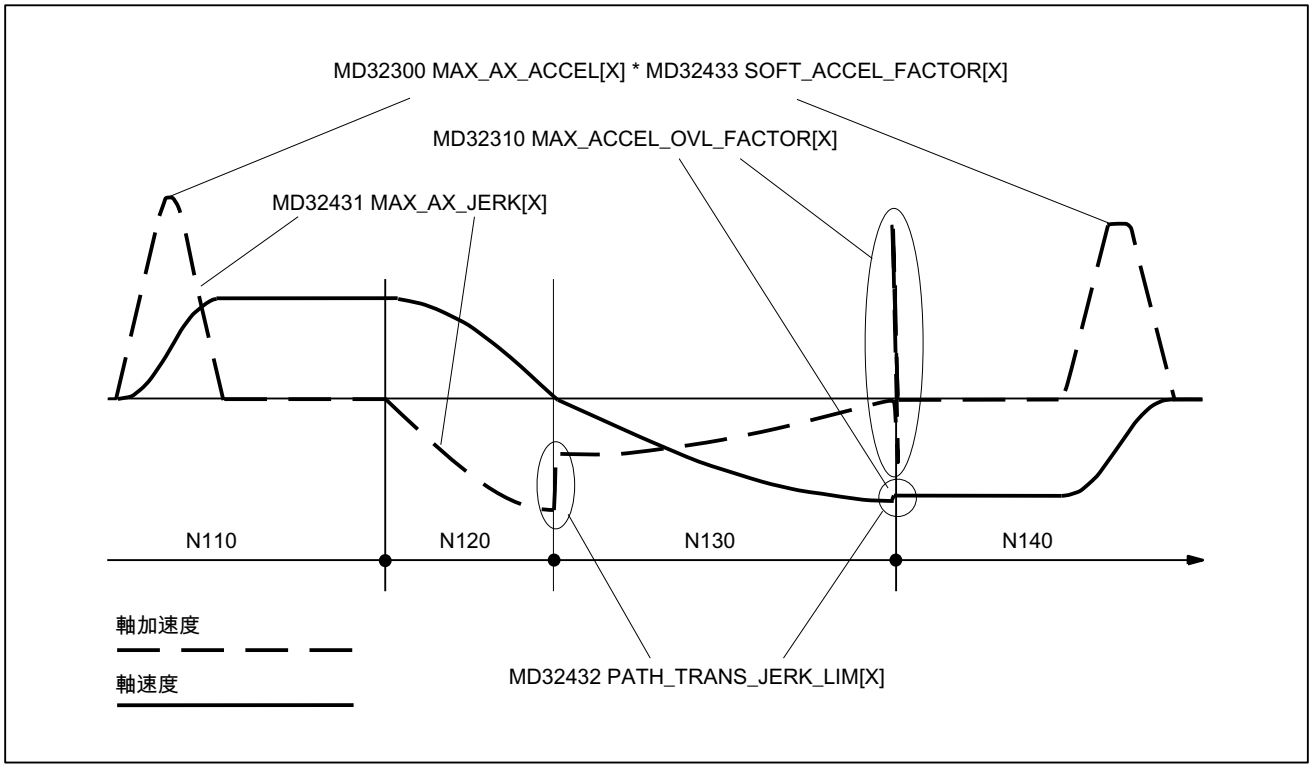


図 6-12　X 軸の速度と加速度の特性曲線

6.3.4 膝形加速特性曲線

6.3.4.1 起動

この例は、膝形加減速特性曲線を、マシンデータとパートプログラム抜粋に基づいて起動する方法を示しています。

マシンデータ

マシンデータ	値
MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT[**X**]	= 0.4
MD35230 $MA_ACCEL_REDUCTION_FACTOR[**X**]	= 0.85
MD35242 $MA_ACCEL_REDUCTION_TYPE[**X**]	= 2
MD35240 $MA_ACCEL_TYPE_DRIVE[**X**]	= TRUE
MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT[**Y**]	= 0.0
MD35230 $MA_ACCEL_REDUCTION_FACTOR[**Y**]	= 0.6
MD35242 $MA_ACCEL_REDUCTION_TYPE[**Y**]	= 1
MD35240 $MA_ACCEL_TYPE_DRIVE[**Y**]	= TRUE
MD35220 $MA_ACCEL_REDUCTION_SPEED_POINT[**Z**]	= 0.6
MD35230 $MA_ACCEL_REDUCTION_FACTOR[**Z**]	= 0.4
MD35242 $MA_ACCEL_REDUCTION_TYPE[**Z**]	= 0
MD35240 $MA_ACCEL_TYPE_DRIVE[**Z**]	= FALSE

チャネル別初期設定としての入力による起動:

MC_GCODE_RESET_VALUE[20] = 3 (DRIVE)

パートプログラムの抜粋

プログラムコード	コメント
N10 G1 X100 Y50 Z50 F700	; DRIVE による軌跡動作(X,Y, Z)
N15 Z20	; DRIVE による軌跡動作(Z)
N20 BRISK	; BRISK への切り替え
N25 G1 X120 Y70	; 代替特性曲線による軌跡動作(Y, Z)
N30 Z100	; BRISK による軌跡動作(Z)
N35 POS[X] = 200 FA[X] = 500	; DRIVEA による位置決め動作(X)
N40 BRISKA(Z)	; Z に対する BRISKA の起動
N40 POS[Z] = 50 FA[Z] = 200	; BRISKA による位置決め動作(Z)
N45 DRIVEA(Z)	; Z に対する DRIVEA の起動

プログラムコード	コメント
N50 POS[Z] = 100	; DRIVE による位置決め動作(Z)
N55 BRISKA(X)	; エラーメッセージが発生

6.4 データリスト

6.4.1 マシンデータ

6.4.1.1 NC 別マシンデータ

番号	識別子:$MN_	説明
18960	POS_DYN_MODE	位置決め軸のダイナミック応答のタイプ

6.4.1.2 チャネルマシンデータ

番号	識別子: $MC_	説明
20150	GCODE_RESET_VALUES	G グループの初期設定
20500	CONST_VELO_MIN_TIME	定速移動の最短時間
20600	MAX_PATH_JERK	軌跡関連の最高加々速度
20602	CURV_EFFECT_ON_PATH_ACCEL	軌跡ダイナミック応答への軌跡曲率の影響
20610	ADD_MOVE_ACCEL_RESERVE	重畳動作に対応した加速度予備
21158	JOG_JERK_ORI	JOG モードで移動する場合の旋回軸の最高加々速度
21159	JOG_JERK_ORI_ENABLE	JOG モードで移動する場合の旋回軸のチャネル別加々速度制限の初期設定
21166	JOG_ACCEL_GEO	JOG モードで移動する場合のジオメトリ軸の最高加速度
21168	JOG_JERK_GEO	JOG モードで移動する場合のジオメトリ軸の最高加々速度

6.4.1.3 軸/主軸マシンデータ

番号	識別子: $MA_	説明
32000	MAX_AX_VELO	最大軸速度
32300	MAX_AX_ACCEL	最高軸加速度
32310	MAX_ACCEL_OVL_FACTOR	速度不連続変化に対応した過負荷係数
32320	DYN_LIMIT_RESET_MASK	応答性制限のリセット応答
32400	AX_JERK_ENABLE	軸加々速度制限
32402	AX_JERK_MODE	軸加々速度制限のフィルタタイプ
32410	AX_JERK_TIME	軸加々速度フィルタの時定数
32420	JOG_AND_POS_JERK_ENABLE	軸の加々速度制限の初期設定
32430	JOG_AND_POS_MAX_JERK	単独軸移動の場合の軸加々速度
32431	MAX_AX_JERK	連続軌跡モードでのブロック変更時の最高軸加々速度
32432	PATH_TRANS_JERK_LIM	ブロック境界でのジオメトリ軸の最高軸加々速度
32433	SOFT_ACCEL_FACTOR	SOFT の加速度制限の倍率
32434	G00_ACCEL_FACTOR	G0 の加速度制限の倍率
32435	G00_JERK_FACTOR	G0 に対応した軸加々速度制限の倍率
32437	AX_JERK_VEL0	速度依存加々速度調整の 1 番目の速度検出値
32438	AX_JERK_VEL1	速度依存加々速度調整の 2 番目の速度検出値
32439	MAX_AX_JERK_FACTOR	設定値を超えた速度の最高加々速度を設定するための係数(速度依存加々速度調整)
35220	ACCEL_REDUCTION_SPEED_POINT	低減加速度の速度
35230	ACCEL_REDUCTION_FACTOR	加速度低減
35240	ACCEL_TYPE_DRIVE	軸の DRIVE 加速特性のオン/オフ
35242	ACCEL_REDUCTION_TYPE	加速度低減のタイプ

6.4.2 セッティングデータ

6.4.2.1 チャネルセッティングデータ

番号	識別子: $SC_	説明
42500	SD_MAX_PATH_ACCEL	最大軌跡加速度
42502	IS_SD_MAX_PATH_ACCEL	SD 42500 の使用: ON/OFF
42510	SD_MAX_PATH_JERK	最大軌跡関連加々速度
42512	IS_SD_MAX_PATH_JERK	SD 42510 の使用: ON/OFF

6.4.3 システム変数

識別子	説明
$AC_PATHACC	リアルタイムイベントの軌跡加速度
$AC_PATHJERK	リアルタイムイベントの軌跡加々速度

F1:突き当て点停止 7

7.1 概略説明

機能

「突き当て点停止」機能により、心押台やスリーブなどの移動する機械要素は移動できるため、他の機械要素に関して定義されたトルクまたは推力を、任意の期間にわたって適用することができます。

特性

- 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命令を使用してプログラム指令可能
  - 「突き当て点停止」機能の起動と解除
  - クランプトルクの設定
  - 監視範囲の設定
- セッティングデータを使用した手動介入オプション
  - 「突き当て点停止」機能の起動と解除
  - クランプトルクの設定
  - 監視範囲の設定
- マシンデータを使用した事前設定
  - クランプトルク
  - 監視範囲
- NC/PLC インターフェース信号による機能状態の識別
- NC/PLC インターフェース信号によるイネーブルオプションまたは確認オプション
- システム変数による、機能のレファレンスおよび実際の状態の読み取り
- 「突き当て点停止」は、軸と主軸に対して実行できます。
- 「突き当て点停止」は、複数の軸に対して同時に、他の軸の移動と同じようにおこなうことができます。
- マルチチャネルブロック検索を、必要なすべての追加データを計算しておこなうことができます。(SERUPRO)

- 「突き当て点停止」および「トルク低減」と組み合わせて、軸移動のシミュレーションが可能です。
- 「垂直」軸も FXS で突き当て点に移動できます。

7.2 詳細説明

7.2.1 プログラミング

機能

突き当て点停止

「突き当て点停止」機能は、FXS、FXST、および FXSW 命令で制御されます。

関連する軸の移動動作なしで、有効化をおこなうこともできます。トルクは直ちに制限されます。突き当て点は、軸が移動すると同時に監視されます。

注記

シンクロナイズドアクション

「突き当て点停止」機能は、シンクロナイズドアクションからでも制御できます。

参照先

『機能マニュアル』、「シンクロナイズドアクション」

トルク/推力を制限した移動

トルク/推力を制限した移動は、FOCON、FOCOF、および FOC 命令で制御できます(「トルク/推力を制限した移動(FOC) (ページ 397)」の章を参照してください)。

構文

```
FXS[<軸>]=<要求>
FXST[<軸>]=<クランプトルク>
FXSW[<軸>] = <範囲の幅>
```

意味

パラメータ	意味
FXS:	「突き当て点停止」機能、効率:モーダル
<要求>:	0 = 完全にオフ 1 = オン
FXST:	クランプトルクの設定
<クランプトルク>:	最大ドライブトルクに対する%で示したクランプトルクの割合 SINAMICS S120:p2003
FXSW:	監視範囲の設定
<範囲の幅>:	突き当て点を中心にした許容範囲の幅 単位:mm、in または°
<軸>:	チャネル軸の名称、タイプ:AXIS

トルク制限(FXST)への変更

トルク制限値は各ブロックで変更できます。変更は、ブロックでプログラム指令された移動動作の実行前に有効になります。加減速制限(ACC)に加えて、トルク制限も作動します。

カーブ形変更

マシンデータを使用して、トルク制限値が直線的に変更される時間の範囲を設定できます。

MD37012 $MA_FIXED_STOP_TORQUE_RAMP_TIME (新しいトルク制限に達するまでの時間)

監視範囲(FXSW)への変更

監視範囲は各ブロックで変更できます。変更は、ブロックでプログラム指令された移動動作の実行前に有効になります。

監視範囲を変更すると、範囲の幅が変更されるだけでなく、実際の軸位置に対する範囲のレファレンス点も変更されます。

連続軌跡モードが有効な場合(G64)の「突き当て点停止」

次のマシンデータを使用して、有効な連続軌跡モード(G64)中に(FXS)機能を選択することで、ブロック切り替え(G60)時にイクザクトストップが起動されないように設定することができます。

MD37060 $MA_FIXED_STOP_ACKN_MASK (突き当て点停止に対応した PLC 応答の監視)

ビット	値	意味
0	0	PLC による応答なしの移動動作の開始
	1	PLC による応答後の移動動作の開始

必要条件

- 突き当て点への移動動作は、軌跡単位またはブロック単位の位置決め軸移動、またはモーダル位置決め軸移動としてプログラム指令できます。
- 突き当て点停止は、複数の機械軸に対して同時に選択することができます。
- 突き当て点へと移動する機械軸の場合は、座標変換、連結、およびフレームの各機能を起動することはできません。
- 移動軌跡と機能の有効化を、パートプログラムの 1 つのブロックでプログラム指令してください。
- 「突き当て点停止」をシンクロナイズドアクションから有効化する場合は、移動軌跡と機能の有効化を別々のブロックでプログラム指令できます。

7.2.2 機能の処理

7.2.2.1 選択

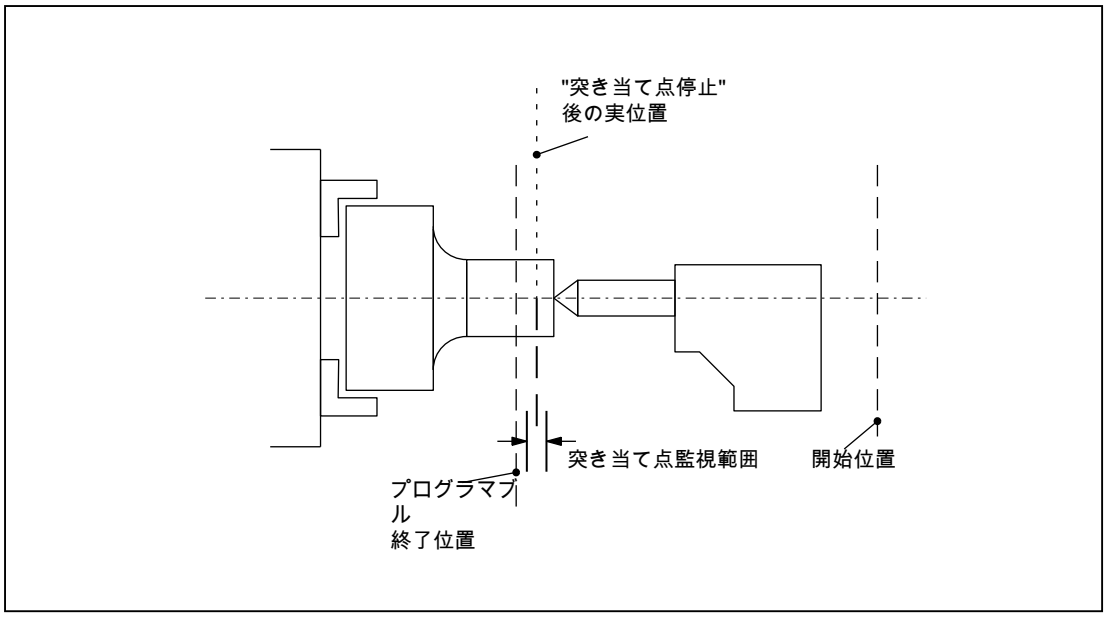


図 7-1　突き当て点停止の例

手順

NC は、機能「突き当て点停止」が選択されていることを、命令 FXS[x]=1 を使用して検出し、IS DB31、... DBX62.4 (「突き当て点停止の有効化」) を使用して PLC にこの機能が選択されていることを通知します。

これに関連して、マシンデータ

MD37060 $MA_FIXED_STOP_ACKN_MASK (突き当て点停止に対応した PLC 応答の監視)

が設定されている場合、システムは IS DB31、... DBX3.1 (「突き当て点停止の許可」)を使用して PLC の応答を待ちます。

次に、プログラム指令速度で、開始位置からプログラム指令された目標位置へのアプローチをおこないます。 突き当て点は、軸/主軸の開始位置と目標位置の間に位置する必要があります。 プログラム指令されたトルク制限は(FXST[<軸>]によって指定したクランプトルク)、ブロックが開始した時点から有効になっています。つまり、突き当て点へのアプローチは、トルクを下げておこなうこともできます。 この制限は、NC で加速度を自動的に下げることによって実施されます。

ブロックにトルクをプログラム指令していない場合、またはプログラムの開始時点からトルクをプログラム指令していない場合は、下記の軸マシンデータに入力した値が有効になります。

MD37010 $MA_FIXED_STOP_TORQUE_DEF (突き当て点用クランプトルクの初期設定)

が入力されます。

7.2.2.2 突き当て点に達しました。

突き当て点の検出

突き当て点の検出、または機械軸が突き当て点に達したことの認識は、以下のマシンデータを使用して設定できます。

MD37040 $MA_FIXED_STOP_BY_SENSOR = <値> (センサによる突き当て点検出)

<値>	意味
0	機械軸の輪郭誤差が、マシンデータで設定した値を超えたときに、突き当て点が到達しています。 MD37030 $MA_FIXED_STOP_THRESHOLD (突き当て点検出値)
1	外部センサが突き当て点に到達したことを検出し、以下の軸の NC/PLC インタフェース信号によってコントローラに通知します。 DB31、... DBX1.2 == 1 (突き当て点のセンサ)
2	<値> == 0 または<値> == 1 で指定された条件のいずれかが該当する場合は、突き当て点が到達しています。

無効な NC/PLC インタフェース信号

軸が「突き当て点に到達」状態である場合、以下の NC/PLC インタフェース信号は無効です。

- DB31、... DBX1.3 (軸/主軸無効)
- DB31、... DBX2.1 (コントローライネーブル)

突き当て点に達したときの動作

突き当て点に達すると、次の動作が実行されます。

- ドライブのトルクが、プログラム指令されたクランプトルク(FXST)まで増加する
- 残りの残移動距離が削除される
- 位置指令値が追従される
- NC/PLC インターフェース信号がリセットされる:DB31、... DBX62.5 = 1 (「突き当て点に到達」)
- ブロック変更の実行:
  - 突き当て点到達時にブロック変更を実行します。MD37060 $MA_FIXED_STOP_ACKN_MASK、ビット 1 = 0
    ブロック変更が、突き当て点への到達直後に実行されます。
  - ブロック変更が、PLC ユーザープログラムによる確認後にのみ実行されます。
    MD37060 $MA_FIXED_STOP_ACKN_MASK、ビット 1 = 1
    突き当て点への到達後、ブロックは、PLC ユーザープログラムにより確認されるまでは変更されません。
    DB31、... DBX1.1 == 1 (突き当て点到達の確認応答)

  ブロック変更後もクランプトルクを保持するために、NC はまだ、機械軸のセットポイントを出力します。
- 突き当て点の監視または監視範囲の有効化

監視範囲

突き当て点への移動ブロック内、またはプログラムの開始以降に、監視範囲用の特定の値が FXSW でプログラム指令されない場合は、次のマシンデータに設定 s された値が有効です。

MD37020 $MA_FIXED_STOP_WINDOW_DEF (突き当て点監視範囲の初期設定)

軸が、突き当て点を検出したときの位置から、範囲で指定された以上に外れると、アラーム 20093 「突き当て点監視から検出されました」が表示され、「突き当て点停止」機能が解除されます。

ユーザーは、軸が突き当て点から外れたときにだけアラームが発生するように監視範囲を選択しなければなりません。

通知
「突き当て点に到達」および突き当て点の破損時
「突き当て点に到達」状態が認識されるとすぐに、軸マシンデータのサーボゲイン係数(K_v) (MD32200)および突き当て点の検出値(MD37030)からの速度指令値がドライブに対して入力されます。この状態で突き当て点が破損すると、軸が、監視範囲制限に達するまで加速します。このとき達する速度は、指定されたマシンデータに設定された値に比例します。適切に高い値の場合は、ドライブが最大モータ速度まで加速します。

一覧

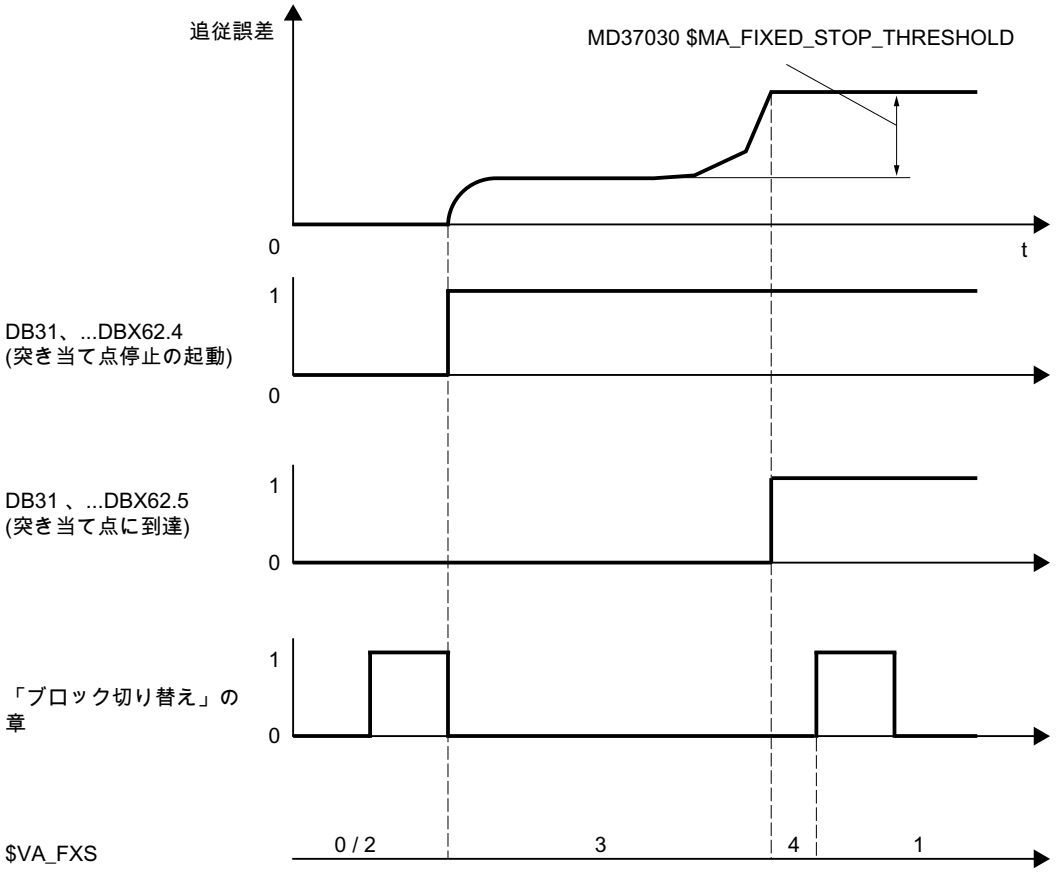


図 7-2　突き当て点に達しました。

7.2.2.3 突き当て点に未達

アラームのマスク

破損のさまざまな原因のアラームは、次のマシンデータを使用してマスクすることができます。

MD37050 $MA_FIXED_STOP_ALARM_MASK = <値>

値	説明：マスクされたアラーム
0	アラーム 20091 「突き当て点に未達」
2	アラーム 20091 「突き当て点に未達」 アラーム 20094 「突き当て点中止」
3	アラーム 20094 「突き当て点中止」

異常または破損の場合の動作

異常が発生した場合、または破損の場合は、以下の動作が実行されます。

- NC/PLC インタフェース信号がリセットされます。DB31、... DBX62.4 = 0 (突き当て点停止の起動)
- 以下のマシンデータの設定に応じて、システムは確認応答を待ちます。
  - MD37060 $MA_FIXED_STOP_ACKN_MASK
  - DB31、... DBX3.1 == 0 (突き当て点停止の許可)
- トルクリミットの取り消し
- ブロック変更の実行

一覧

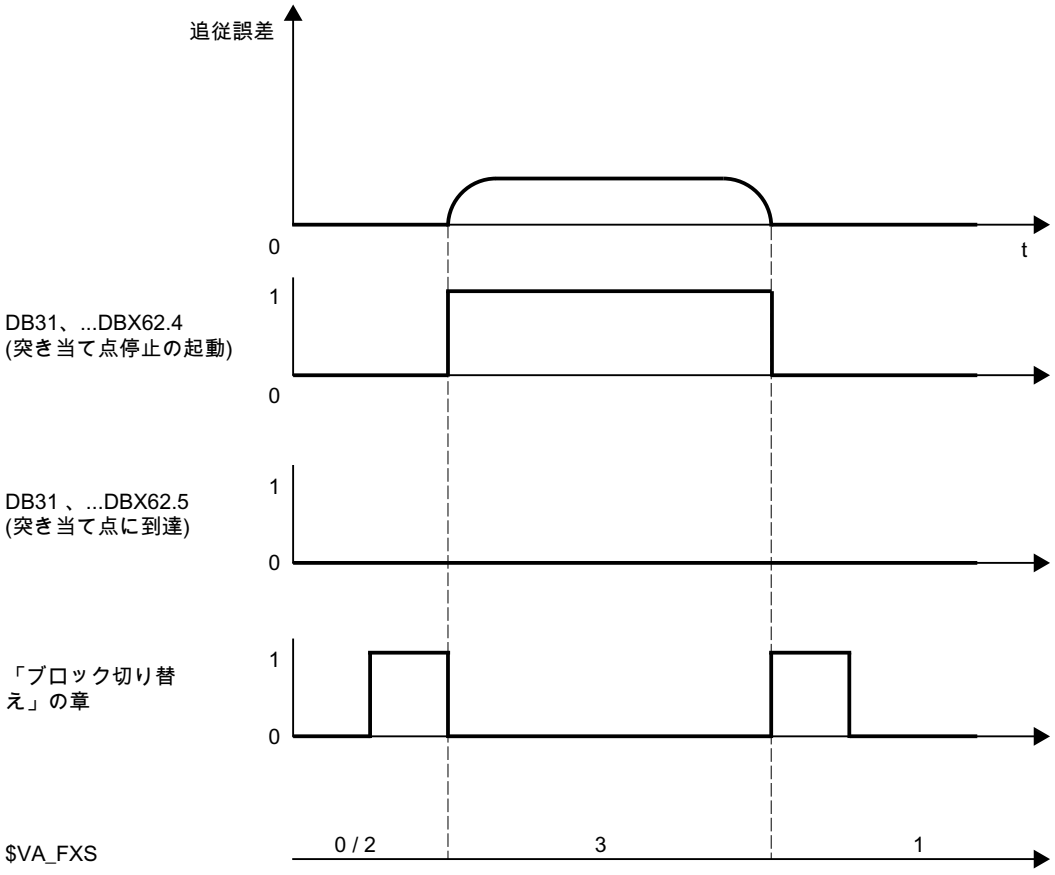


図 7-3　突き当て点に未達

7.2.2.4　解除

「突き当て点停止」機能が、NC プログラムのブロック内で命令 `FXS[<axis>] = 0` を使用して選択解除されます。

機能の選択解除時の動作

機能を選択解除すると、以下の動作が実行されます。

- 先読み停止が開始される(`STOPRE`)
- NC/PLC インターフェース信号がリセットされる
  - DB31、... DBX62.4 = 0 (突き当て点停止の起動)
  - DB31、... DBX62.5 = 0 (突き当て点に到達)

- 以下のマシンデータに応じて、システムは、PLC ユーザープログラムからの確認応答を待ちます。
  - MD37060 $MA_FIXED_STOP_ACKN_MASK
  - DB31、... DBX3.1 == 0 (突き当て点停止の許可)
  - DB31、... DBX1.1 == 0 (突き当て点到達への応答)
- フォローアップモードの終了
- 軸が閉ループ位置制御を再開する I

パルスイネーブル

パルスイネーブルまたはパルス禁止は以下によりキャンセルできます。

- ドライブ:端子 EP(パルス許可)を使用
- NC/PLC インタフェース信号:DB31、... DBX21.7 (「パルスイネーブル」)

突き当て点での動作は、以下のマシンデータを介して設定できます。

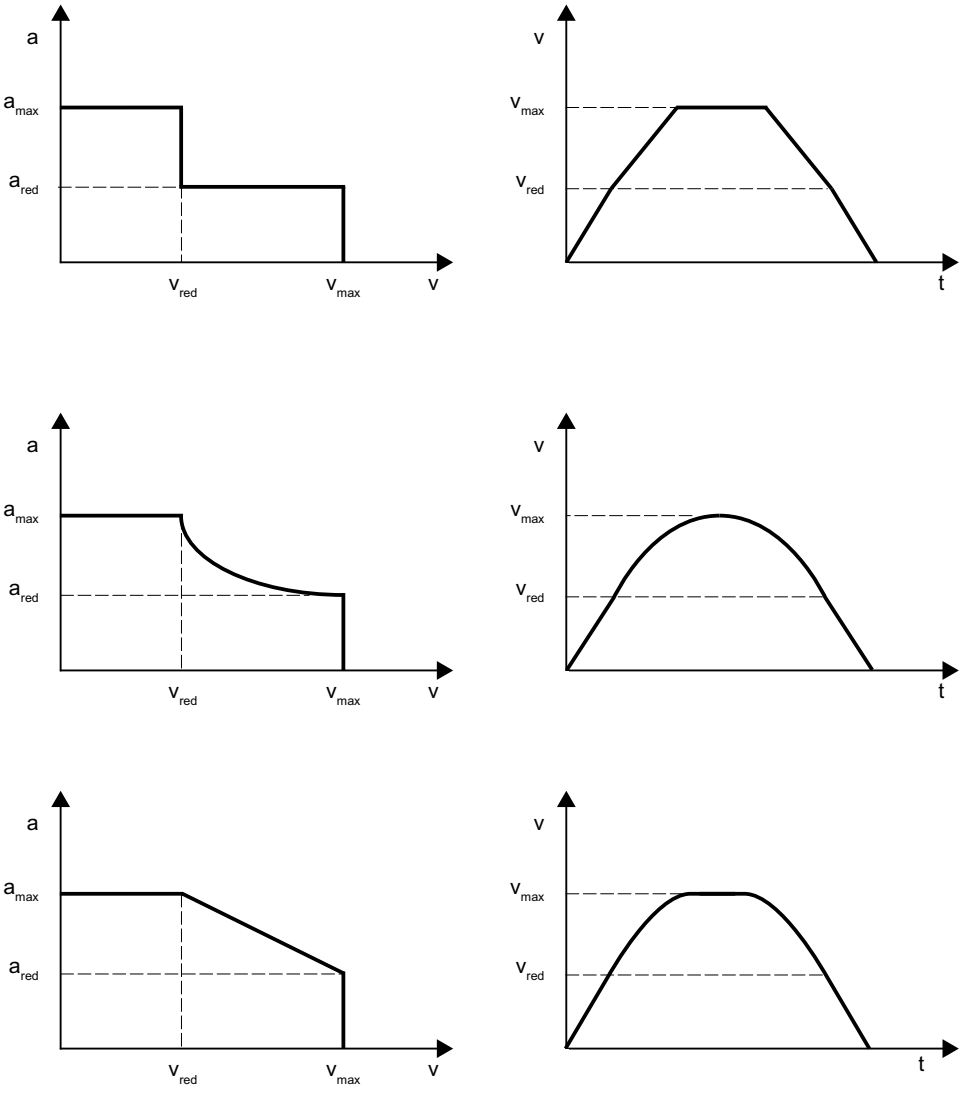
MD37002 $MA_FIXED_STOP_CONTROL、ビット 0 およびビット 1 (突き当て点への移動のシーケンス制御)

ビット	値	意味
0	突き当て点でのパルス**禁止**の動作	
	0	「突き当て点停止」が中止されます。
	1	「突き当て点停止」が中断されます。つまり、ドライブの通電が停止されます。
1	突き当て点でのパルス**イネーブル**の動作	
	0	トルクが即時適用されます。
	1	マシンデータに指定された時間にわたってトルクが直線的に加速します。 MD37012 $MA_FIXED_STOP_TORQUE_RAMP_TIME

特殊なケース:選択解除時のパルスイネーブルの削除

選択解除時に、機能の状態が「PLC 応答待ち」で、パルスイネーブルが削除された場合は、トルクリミットが 0 に削減されます。この状態では、パルスイネーブルを再度設定しても、トルクはドライブで確立されていない状態になります。選択解除が完了した後は、軸を正常に移動できます。

一覧



① 選択解除 FXS[<axis>]=0 の移動ブロック

図 7-4 突き当て点の解除

7.2.3 ブロック検索時の動作

機能

計算ありのブロック検索

- 軸を突き当て点で停止するようになっているプログラム区間内にターゲットブロックが存在する場合、軸が突き当て点にまだ到達していなければ、その軸は突き当て点にアプローチします。
- 軸を突き当て点で停止させないようになっているプログラム区間内にターゲットブロックが配置されている場合、軸がまだ突き当て点にあれば、その軸は突き当て点から離れます。
- 軸が「突き当て点に到達」状態にある場合、メッセージ 10208「[NC スタート]を押してプログラムを再開」が表示されます。「NC スタート」でプログラムを再開できます。
- ターゲットブロックの開始時には、クランプトルク FXST と監視範囲 FXSW に通常のプログラム処理の場合の値が設定されています。

計算なしのブロック検索

FXS、FXST および FXSW 命令はブロック検索時に無視されます。

FOCON/FOCOF の有効性

モーダル動作のトルク/推力制限の状態 FOCON/FOCOF は、ブロック検索時に維持され、アプローチブロックで有効になります。

FXS または FOC によるブロック検索

前回有効であった加工の状態と機能をすべて取得するために、ユーザーはターゲットブロックのプログラム領域で FXS または FOC を選択します。NC が、選択されたプログラムをプログラムテストモードで自動的に起動します。ターゲットブロックが見つかったら、NC はターゲットブロックの開始点で停止し、プログラムテストを内部的に再度解除し、そのブロック表示に停止状態「検索ターゲットが見つかりました」を表示します。

⚠ 注意

SERUPRO アプローチでは、実際には FXS 命令は考慮されません。

FXS ブロックのプログラム指令された終点へのアプローチが、**トルク制限なし**でシミュレーションされるだけです。

FXS がプログラムの開始点と検索ターゲットの間にある場合、この命令は NC によって実行されません。プログラム指令された終点までの動作がシミュレーションされるだけです。

ユーザーはパートプログラムに FXS のオンとオフを記述できます。必要に応じて、ユーザーは、この SERUPRO-ASUB で FXS を有効または無効にするために ASUP を起動できます。

システム変数

「突き当て点停止」のレファレンスと実際の状態は、以下のシステム変数を使用して読み取ることができます。

- $AA_FXS (レファレンス状態)
- $VA_FXS (実際の状態)

SERUPRO: $AA_FXS (レファレンス状態)

SERUPRO 時には、$AA_FXS の値は「突き当て点停止」機能の起動状態に対応して以下のようになります。

「突き当て点停止」機能の起動状態	システム変数$AA_FXS ==
「解除」	0 (軸が突き当て点にありません)
「起動」	3 (「突き当て点停止」の選択が有効です)

注記

SERUPRO 時は、システム変数$AA_FXS の値は 0 と 3 のみになります。その結果、$AA_FXS に基づき、SERUPRO を使用して、プログラム分岐の通常のプログラム実行と比較してプログラムシーケンスを変更できます。

SERUPRO: $VA_FXS (実際の状態)

SERUPRO 中、変数$VA_FXS は常に機械軸の実際の状態を示します。

例

SERUPRO ASUP でシステム変数$AA_FXS と$VA_FXS を使用して「突き当て点停止」機能の現在の状態を特定し、適切な動作を開始できます。

プログラムコード:FXS_SERUPRO_ASUP.MPF	コメント
N100 WHEN ($AA_FXS[X]==3) AND ($VA_FXS[X]==0) DO FXS[X]=1	; リファレンス==「選択が無効です」および ; 実際==「軸が突き当て点にありません」 ; => 「スイッチオン」

プログラムコード:FXS_SERUPRO_ASUP.MPF	コメント
N200 WHEN ($AA_FXS[X]==0) AND ($VA_FXS[X]==1) DO FXS[X]=0	; リファレンス==「軸が突き当て点にありません」および ; 実際==「突き当て点へのアプローチが成功しました」 ; => 「スイッチオン」
N1020 REPOSA	; すべての軸で輪郭に直線的に再アプローチ中

REPOS オフセットの表示

検索ターゲットが見つかると、「突き当て点停止」に関する実際の状態が、以下の NC/PLC インターフェース信号を介して軸毎に表示されます。

- DB31、... DBX62.4 (突き当て点停止の起動)
- DB31、... DBX62.5 (突き当て点に到達)

例:

機械が突き当て点にあり、FXS 解除後にターゲットブロックが使用できる場合、DB31、... DBX62.5 (突き当て点に到達) により、新しい目標位置が REPOS オフセットとして表示されます。

REPOS と FXS

REPOS を使用すると、FXS の動作が自動的に繰り返されます。以降ではこれを FXS-REPOS と呼びます。この処理は FXS_SERUPRO_ASUP.MPF プログラムと比較されます。各軸が考慮され、検索ターゲットの前に最後にプログラム指令されたトルクが適用されます。

ユーザーは SERUPRO ASUB 内で FXS を個別に処理できます。

その後、次のことが適用されます。

SERUPRO ASUB 内で実行されるすべての FXS 動作は、自動的に次の点を考慮します。

$AA_FXS[<軸>] = $VA_FXS[<軸>]

これにより、X 軸に対して FXS-REPOS が無効になります。

FXS-REPOS の無効化

FXS-REPOS は次の場合に無効になります。

- REPOSA を参照する FXS シンクロナイズドアクション
- SERUPRO_ASUP 内での$AA_FXS[X] = $VA_FXS[X]

注記

FXS 処理**がない** SERUPRO ASUP や SERUPRO ASUP がない場合は、自動的に FXS-REPOS となります。

⚠ 注意

FXS-REPOS に対して速すぎる速度

FXS-REPOS は、すべての軌跡軸を一緒に目標位置まで移動します。そのため、FXS 処理のある軸と処理のない軸は、ターゲットブロックで有効な G 命令と送り速度で一緒に移動します。結果として、早送り(G0)またはより早い速度で突き当て点にアプローチする場合があります。

FOC-REPOS

FOC-REPOS は、FXS-REPOS と同じように動作します。

プログラム事前処理中に変化するトルク特性は、FOC-REPOS では実行できません。

例

軸 X が位置 0 から 100 へと移動します。FOC が、20 ミリメートル毎に 10 ミリメートルに対してオンになります。得られるトルク特性はノンモーダルな FOC で生成されるため、FOC-REPOS によってトレースできません。軸 X は、ターゲットブロックの前の最後のプログラムに対応して、FOC を使用してまたは使用しないで FOC-REPOS によって移動されます。

FXS「突き当て点停止」のプログラム例については、「プログラムテスト (ページ 589)」の章を参照してください。

7.2.4 リセットと機能の中止の動作

NC リセット

機能がまだ「正常な突き当て点停止」状態に入っていない間は、突き当て点停止を NC リセットで中止できます。

既に突き当て点にアプローチ済みの場合でも、指定された停止トルクにまだ完全に到達していなければ、機能を NC リセットで中止できます。 軸の位置指令は、現在の実位置と原点同期されます。

機能は「正常な突き当て点停止」状態に入ると同時に、NC リセット後も有効のままとなります。

機能の中止

次の場合に機能は中止します。

- 非常停止

⚠ 注意

突き当て点停止の場合に機械の危険状況が発生する可能性

「非常停止」が実施またはリセットされた場合、突き当て点停止が有効な間は、機械の危険状況の発生を確実に防止する必要があります。

パルスイネーブルを設定およびキャンセルする場合の動作を例にあげます。
MD37002 $MA_FIXED_STOP_CONTROL、ビット 0 (停止時のパルス禁止の動作)

- ビット 0 = 0: 突き当て点停止が中止されます
- ビット 0 = 1:突き当て点停止が中断されました。つまり、ドライブに電源が供給されていません。
  再度パルス禁止をキャンセルすると、ドライブは再び停止トルクで前進します。

注記

NC とドライブは、「非常停止」中は電力を供給されません。つまり、PLC が応答する必要があります。

- 機能状態: 「突き当て点に未達」
- 機能状態: 「突き当て点中止」
- PLC ユーザープログラムにより中止
  DB31、... DBX62.4 = 0 (「突き当て点停止の有効化」)
- パルスイネーブルとマシンデータのパラメータ設定のキャンセル
  MD37002 $MA_FIXED_STOP_CONTROL、ビット 0 = 0 (上記を参照してください)

7.2.5 その他の機能に関する動作

残移動距離削除による測定

「突き当て点停止」 (FXS)は、 「残移動距離削除による測定」 (MEAS)機能を持つブロックでプログラム指令できません。 1 つの機能が 1 つの軌跡軸に作用し、その他の機能は 1 つの位置決め軸に作用するか、両方の機能が位置決め軸に作用する場合は除きます。

軌跡誤差監視

軸軌跡誤差監視機能は、「突き当て点停止」が有効になっている間は無効になります。

位置決め軸

位置決め軸 POSA での「突き当て点停止」の場合、位置決め軸が時間までに突き当て点に到達していない場合でも、ブロック切り替えが実行されます。

垂直軸

垂直軸に対する「突き当て点停止」機能は、アラームが有効になっている場合でも使用できます。

突き当て点停止時に垂直軸に対して機能別のアラームが発生した場合、NC/PLC インタフェース信号 DB11、DBX6.3 (モードグループ準備完了)はリセットされません。 つまり、対応するドライブは通電停止されません。

これは、垂直軸の電子重量補正に対応していて、次のマシンデータを使用して設定できます。

MD37052 $MA_FIXED_STOP_ALARM_REACTION

参照先:
垂直軸についての詳細は、以下を参照してください。

- 『SINAMICS S120 機能マニュアル』
- 『機能マニュアル、上級機能』; 「誤差補正(K3)」、
  :電子カウンタウェイトの章

7.2.6 セッティングデータ

機能別 FXS、FXST、および FXSW 命令によってプログラム指令値は、以下の直ちに有効になる軸セッティングデータに書き込まれます。

ファンクションのオン/オフ切り替え

SD43500 $SA_FIXED_STOP_SWITCH (突き当て点停止の選択/選択解除)

クランプトルク

SD43510 $SA_FIXED_STOP_TORQUE (クランプトルク)

注記

100%を超えるクランプトルク

SD43510 のクランプトルクの値が最大モータトルクの 100%を超えるのは、短時間にしてください。 また、最大モータトルクはドライブにより制限されます。 例えば、以下のドライブパラメータには制限作用があります。

- p1520/p1521 トルク上限/推力上限 / トルク下限/推力下限
- p1522/p1523 トルク上限/推力上限 / トルク下限/推力下限
- p1530/p1531 電力制限、監視 / 電力制限、回生
- p0640 電流リミット
- p0326 モータストール推力補正係数

ドライブパラメータと機能の詳細情報は、下記の資料を参照してください。

参照先

- 『SINAMICS S120/S150 リストマニュアル』
- 『SINAMICS S120 機能マニュアル』

監視範囲

SD43520 $SA_FIXED_STOP_WINDOW (監視範囲)

初期設定

セッティングデータの初期設定は、次のマシンデータを使用して設定されます。

- クランプトルク:
  MD37010 $MA_FIXED_STOP_TORQUE_DEF (初期設定のクランプトルク)
- 監視範囲:
  MD37020 $MA_FIXED_STOP_WINDOW_DEF (初期設定の監視範囲)

有効性

クランプトルクと監視範囲のセッティングデータは、直ちに有効になります。 このように、クランプ状態はオペレータまたは PLC ユーザープログラムによって、いつでも加工状況に合わせて調整することができます。

参照先

マシンデータとセッティングデータの詳細については、以下を参照してください。

『リストマニュアル』、「マシンデータの詳しい説明」

7.2.7 システム変数

リファレンス/実際の状態

「突き当て点停止」のレファレンスと実際の状態は、以下のシステム変数を使用して読み取ることができます。

- $AA_FXS = <値> (ステータス、「突き当て点停止」レファレンス状態)
- $VA_FXS = <値> (ステータス、「突き当て点停止」実際の状態)

<値>	説明
0	軸が突き当て点にありません。
1	突き当て点へのアプローチが成功しました。
2	突き当て点へのアプローチが失敗しました。
3	「突き当て点停止」の選択が有効です。
4	突き当て点が検出されました。
5	突き当て点停止の解除が有効です。

関連情報

突き当て点($VA_FXS == 2)への移動時にエラー が発生すると、以下のシステム変数に追加情報が表示されます。

- $VA_FXS_INFO = <値> (「突き当て点停止」時の追加情報)

<値>	説明
0	追加情報はありません。
1	アプローチ動作はプログラム指令されていません。
2	プログラム指令された終了位置に到達し、移動が停止しました。
3	リセット、DB21 ... DBX7.7 == 1 を使用した取り消し
4	監視範囲が終了しました。
5	トルク低減がドライブによって拒否されました。
6	PLC がイネーブル、DB31、... DBX3.1 == 0 (突き当て点停止の許可)を取り消しました。

$AA_FXS の用途例

ブロック切り替えを実行するには、障害発生時にアラームを発行しないでください。システム変数$VA_FXS_INFO を使用して原因が決定され、特定の動作が開始されます。

必要条件:MD37050 $MA_FIXED_STOP_ALARM_MASK = 0 (突き当て点停止時にアラームが開始されない)

プログラムコード	コメント
X300 Y500 F200 FXS[X1]=1 FXST[X1]=25 FXSW[X1]=5	突き当て点停止
IF $AA_FXS[X1] == 2 GOTOF FXS_ERROR	; IF 突き当て点==到達
R100=$AA_IM[X1]	; THEN (一般的な場合)
IF R100 ...	; 実際の位置の
...	; 現在位置
GOTOF PROG_END	
FXS_ERROR:	; ELSE (異常時)
CASE($VA_FXS_INFO[X1]) OF 0 GOTOF LABEL_0 OF 1 GOTOF LABEL_1 ...	; エラーの処理
LABEL_0: ...	
GOTOF CASE_END	
LABEL_1: ...	
GOTOF CASE_END	
...	
CASE_END:	
...	
PROG_END:M30	; ENDIF プログラム終了

7.2.8 アラーム

アラーム 20091 「突き当て点に未達」

突き当て点停止中に突き当て点に到達しない場合は、アラーム 20091 「突き当て点に未達」が表示され、ブロック切り替えが実行されます。

アラーム 20092「突き当て点停止がまだ有効です」

突き当て点に到達した後に、軸に対して移動要求または更新された機能選択があった場合、アラーム 20092「突き当て点停止がまだ有効です」が表示されます。

アラーム 20093「停止状態監視が異常を検出しました」

軸が突き当て点に到達した後で、その位置からセッティングデータで指定された値以上に離れた場合

SD43520 FIXED_STOP_WINDOW (突き当て点監視範囲)

アラーム 20093「停止状態監視が異常を検出しました」が表示され、当該軸の突き当て点停止が選択解除され、次のシステム変数が設定されます:
$AA_FXS[x] = 2

アラーム 20094「機能が中止されました」

パルスイネーブルがキャンセルされたためにクランプトルクが適用できなくなるか、NC/PLC インタフェースの要求された応答信号がリセットされた場合、突き当て点停止は中止されます。

- 必要な応答信号: MD37060 $MA_FIXED _STOP_ACKN_MASK、ビット 0 = 1
- 応答信号: DB31、... DBX3.1 == 0 (「突き当て点停止の許可」)

突き当て点アラームの許可

次のマシンデータを使用して、突き当て点アラーム

- アラーム 20091 「突き当て点に未達」
- アラーム 20094 「突き当て点中止」

を表示するかどうかを設定できます。

MD37050 $MA_FIXED_STOP_ALARM_MASK (突き当て点アラームの有効化)

突き当て点アラームに対して設定可能な機能動作

次のマシンデータを使用して、機能別アラームの発生時に機能を中止することを設定できます。

- アラーム 20090 「突き当て点停止を使用できません」
- アラーム 20091 「突き当て点に未達」
- アラーム 20092「突き当て点停止がまだ有効です」
- アラーム 20093「停止状態監視が異常を検出しました」
- アラーム 20094 「突き当て点停止が中止されました」

MD37052 $MA_FIXED_STOP_ALARM_REACTION (突き当て点アラームに対する応答)

新規プログラミング後のアラームマスク

突き当て点停止は、簡単な計測機能に使用できます。

たとえば、定義済の障害物まで移動して工具長を計測することによって、工具が破損していないか確認できます。そのためには、突き当て点アラームをマスクしてください。

その後、ワークをクランプする機能を「通常どおり」使用するとき、パートプログラム命令を使用してアラームを有効にできます。

7.2.9 トルク/推力を制限した移動(FOC)

機能

「トルク制限」機能を使用して、制限されたトルク/推力で移動している場合は、最大許容トルク/推力が、最大許容軸トルクの一定の割合までに制限されることがあります。制限値は、軸の移動時を含め、いつでも変更できます。変更は、距離、時間、または他の任意の変数に応じて、補間クロック周期で実行できます。「トルク制限」機能は、シンクロナイズドアクションにもプログラム指令できます。この機能は、常時動作中にすることも、ブロック単位で起動することもできます。

プログラミング

構文

```
FOCON[<軸>]
FOCOF[<軸>]
FOC[<軸>]
```

意味

パラメータ	意味
FOCON:	トルク/推力の制限を起動する
FOCOF:	トルク/推力の制限を解除する
FOC:	トルク/推力の制限をノンモーダル起動する
<軸>:	チャネル軸名称、タイプ:AXIS

例

プログラムコード	コメント
N10 FOCON[X]	; トルク制限のモーダル起動
N20 X100 Y200 FXST[X]=15	; 低減されたトルク(15%)でXが移動
N30 FXST[X]=75 X20	; トルクを75%に変更、
N40 FOCOF[X]	; トルク制限の解除

パラメータ設定

マシンデータ

- MD37010 $MA_FIXED_STOP_TORQUE_DEF (突き当て点用クランプトルクの初期設定)
  マシンデータで指定された値は、FXST を使用して明示的な値がプログラム指令されている限り、機能の起動後に有効になります。
- MD36042 $MA_FOC_STANDSTILL_DELAY_TIME (有効なトルク/推力の制限の遅延時間の停止状態監視)

常時起動(FOCON/FOCOF)

電源投入またはリセット後の「トルク制限」機能の常時起動は、以下のマシンデータを使用して設定できます。

MD37080 $MA_FOC_ACTIVATION_MODE、ビット 0 およびビット 1 (モーダル推力/トルク制限の初期設定)

ビット	値	意味
0	0	電源投入後に「トルク制限」機能が**無効**になります。
	1	電源投入後に「トルク制限」機能が有効になります。
1	0	リセット後に「トルク制限」機能が**無効**になります。
	1	リセット後に「トルク制限」機能が有効になります。

トルクまたは推力の制限を、起動する前に FXST = <値>を使用してプログラム指令した場合、起動以降は有効になります。

トルクまたは推力の制限を、起動する前に FXST = <値>を使用してプログラム指令しなかった場合、標準では、マシンデータで指定した値が有効になります。

MD37010 $MA_FIXED_STOP_TORQUE_DEF (突き当て点用クランプトルクの初期設定)

ブロック単位の起動(FOC)

実際のブロックの機能が、FOC 命令を使用して起動されます。

シンクロナイズドアクションを使用して機能を起動すると、現在のパートプログラムブロックの終了まで有効です。

FOC の前の FXS の優先度

「突き当て点停止」機能 FXS の起動には、「トルク制限」機能より高い優先度があります。「突き当て点停止」が「トルク制限」と同時に有効である場合は、「突き当て点停止」が実行されます。

「突き当て点停止」機能 FXS を解除すると、クランプがキャンセルされます。「トルク制限」機能は、同時に常時有効である場合は、有効のままです。

システム変数

「トルク制限」機能の状態

「トルク制限」機能の状態は、システム変数$AA_FOC を使用して読み取ることができます。

「突き当て点停止」機能 FXS の状態を変更しても、「トルク制限」機能の状態は変更されません。

$AA_FOC	
値	意味
0	FOC は無効です。
1	FOC はモーダルで有効です。
2	FOC はノンモーダルで有効です。

トルク制限の状態

トルク制限の実際の状態は、システム変数$VA_TORQUE_AT_LIMIT を使用して読み取ることができます。

$VA_TORQUE_AT_LIMIT	
値	意味
0	実効トルク < トルク制限値
1	実効トルク == トルク制限値

制限事項

「トルク制限」機能には、以下の制限事項があります。

- 加速度制限として記述されたトルク/推力制限の変更が考慮されるのは、**ブロック境界での移動動作時に限ります** (ACC 命令を参照してください)。
- FOC の場合に限る:有効なトルク制限に到達しているかどうか確認するために NC/PLC インタフェースから**監視することはできません**。
- 加速制限が適宜補正されない場合、移動動作中の追従誤差が増加します。
- 加速制限が適宜補正されない場合は、ブロック終了位置に到達するのが、MD36040 $MA_STANDSTILL_DELAY_TIME で
  指定したより遅くなる可能性があります。
  このためにマシンデータ
  MD36042 $MA_FOC_STANDSTILL_DELAY_TIME
  が導入され、この状態で監視されます。

リンク軸とコンテナ軸の可能な使用法

チャネル内を移動できる軸はすべて、つまりリンク軸とコンテナ軸も、突き当て点まで移動できます。

参照先:
『機能マニュアル、上級機能』、「NCU と操作パネルの n:m 接続(B3)」

コンテナ回転の場合、機械軸の状態は保持されます。つまり、クランプされた機械軸は停止したままです。

FOCON によりモーダルトルク制限が有効になっている場合、コンテナ回転後も、機械軸にこの制限が保持されます。

7.3 例

例 1:静的シンクロナイズドアクションによる突き当て点停止

静的シンクロナイズドアクションでは、R 変数($R1)を使用した要求に応じて突き当て点停止(FXS)を開始します。

プログラムコード	コメント
N10 IDS=1 WHENEVER	; 静的シンクロナイズドアクション 1
(($R1==1) AND	; R1==1 (Y の FXS が要求されました)および

プログラムコード	コメント
($AA_FXS[Y]==0)) DO	; 複数選択の回避
	; $AA_FXS[Y]==0 (軸が突き当て点にありません): =>
$R1=0 FXS[Y]=1	; $R1 をリセットし、Y の FXS を起動します
FXST[Y]=10	; 制限トルク:10%
FA[Y]=200	; 軸送り速度 Y:200
POS[Y]=150	; 位置決め動作 Y
N11 IDS=2 WHENEVER	; 静的シンクロナイズドアクション 2
($AA_FXS[Y]==4) DO	; $AA_FXS[Y]==4 (突き当て点が検出されました): =>
FXST[Y]=30	; 制限トルク:30%
N12 IDS=3 WHENEVER	; 静的シンクロナイズドアクション 3
($AA_FXS[Y]==1) DO	; $AA_FXS[Y]==1 (突き当て点に正常に到達しました): =>
FXST[Y]=$R0	; 制限トルク:R 変数$R0 からの値
N13 IDS=4 WHENEVER	; 静的シンクロナイズドアクション 4
(($R3==1) AND	; R3==1:Y の FXS の解除が要求されました
($AA_FXS[Y]==1)) DO	; $AA_FXS[Y]==1 (突き当て点に正常に到達しました): =>
FXS[Y]=0	; FXS を解除します
FA[Y]=1000 POS[Y]=0	; 位置決め動作 Y
N20 FXS[Y]=0 G0 G90 X0 Y0	; 初期設定の設定:FXS が解除され、 X および Y が初期位置にあります
N30 RELEASE(Y)	; シンクロナイズドアクション内での動作のために Y を解放します
...	
N60 GET(Y)	Y 軸を軌跡グループに戻します
...	

注記

FXS の場合の複数選択の回避

複数選択を回避するために、FXS を起動する前に、$AA_FXS==0 システム変数またはユーザー用フラグのいずれかを確認することを推奨します。前述のプログラム例 N10 を参照してください。

例 2 :ブロック単位のシンクロナイズドアクションによる突き当て点停止

突き当て点停止は、次のブロックの移動動作の特定位置から起動されます。

プログラムコード	コメント
N10 G0 G90 X0	; 開始位置
N20 WHEN $AA_IW[X]>17 DO FXS[X]=1	; シンクロナイズドアクション:X 軸の実位置> 17
	; => X 軸の FXS を起動します

プログラムコード	コメント
`N30 G1 F200 X100`	; X 軸の移動動作

注記

ブロック単位のシンクロナイズドアクションは、以下のメインプログラムブロックで処理されます。

7.4 データリスト

7.4.1 マシンデータ

7.4.1.1 軸/主軸マシンデータ

番号	識別子: $MA_	説明
36042	FOC_STANDSTILL_DELAY_TIME	有効なトルク/推力の制限のための遅延時間、停止状態監視
37000	FIXED_STOP_MODE	突き当て点停止モード
37002	FIXED_STOP_CONTROL	突き当て点停止のためのシーケンス監視
37010	FIXED_STOP_TORQUE_DEF	突き当て点クランプトルクの初期設定
37012	FIXED_STOP_TORQUE_RAMP_TIME	修正されたトルク制限に到達するまでの時間
37020	FIXED_STOP_WINDOW_DEF	突き当て点監視範囲の初期設定
37030	FIXED_STOP_THRESHOLD	突き当て点検出の検出値
37040	FIXED_STOP_BY_SENSOR	センサを使用した突き当て点検出
37050	FIXED_STOP_ALARM_MASK	突き当て点アラームの許可
37052	FIXED_STOP_ALARM_REACTION	突き当て点アラームへの応答
37060	FIXED_STOP_ACKN_MASK	突き当て点停止に対応した PLC 応答の監視
37070	FIXED_STOP_ANA_TORQUE	アナログドライブでの突き当て点アプローチのトルク制限
37080	FOC_ACTIVATION_MODE.	モーダルなトルク/推力制限の初期設定

7.4.2 セッティングデータ

7.4.2.1 軸/主軸セッティングデータ

番号	識別子: $SA_	説明
43500	FIXED_STOP_SWITCH	突き当て点停止の選択
43510	FIXED_STOP_WINDOW	突き当て点クランプトルク
43520	FIXED_STOP_TORQUE	突き当て点監視範囲

7.4.3 信号

7.4.3.1 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
突き当て点到達の応答	DB31、... .DBX1.1	DB380x.DBX1.1
突き当て点のセンサ	DB31、... .DBX1.2	DB380x.DBX1.2
軸/主軸無効	DB31、... .DBX1.3	DB380x.DBX1.3
コントローライネーブル	DB31、… .DBX2.1	DB380x.DBX2.1
突き当て点停止が有効	DB31、… .DBX3.1	DB380x.DBX3.1

7.4.3.2 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
突き当て点停止の起動	DB31、... .DBX62.4	DB390x.DBX2.4
突き当て点に到達	DB31、... .DBX62.5	DB390x.DBX2.5

G2: 速度、指令値/フィードバック回路、閉ループ制御 8

8.1 概略説明

各機能の説明では、次の各項目に関連して、機械軸をパラメータ設定する方法について説明します。

- フィードバック値/検出器
- 指令回路
- 操作精度
- 移動範囲
- 軸速度
- 制御パラメータ

8.2 速度、移動範囲、精度

8.2.1 速度

軌跡および軸、主軸の最高速度

軌跡および軸、主軸の最高速度は、 機械構成、ドライブのダイナミック応答およびフィードバック値の取得の限界周波数(エンコーダ限界周波数)により影響を受けます。

最高軸速度はマシンデータで定義します。

MD32000 $MA_MAX_AX_VELO (最高軸速度)

最高許容主軸速度は、マシンデータを使用して指定します。

MD35100 $MA_SPIND_VELO_LIMIT (最高主軸速度)

詳しくは、「S1:主軸 (ページ 1555)」の章を参照してください。

送り速度が早い(プログラムされた送り速度をオーバライドした結果)場合、最高軌跡速度は、V_{max} に制限されます。

この自動送り速度制限により、CAD システムにより生成された非常に短いブロックを持つプログラムにおいて複数のブロックにわたって速度が遅くなることがあります。

例:

補間周期= 12 ms

```
N10 G0 X0 Y0; [mm]
N20 G0 X100 Y100; [mm]
```

⇒ ブロックでプログラム指令された軌跡長= 141.42 mm

⇒ V_{max} = (141.42 mm/12 ms) * 0.9 = 10606.6 mm/s = 636.39 m/min

最小軌跡、軸速度

次の制限は、最小軌跡または軸速度に適用されます。

$$V_{分} \geq \frac{10^{-3}}{\text{計算分解能} \quad [\frac{\text{インクレメント}}{\text{mmまたは°}}] * \text{IPOサイクル}}$$

計算分解能は、次のマシンデータを使用して定義します。
MD10200 $MN_INT_INCR_PER_MM (直線位置用の計算分解能)

または
MD10210 $MN_INT_INCR_PER_DEG (回転位置用の計算分解能)

V_{min} に到達しない場合、移動は実行されません。

例:

MD10200 $MN_INT_INCR_PER_MM = 1000 [incr/mm] ;

補間周期= 12 ms;

⇒ $V_{min} = 10^{-3}/(1000\ ^{incr}/_{mm} \times 12\ ms) = 0.005\ ^{mm}/_{min}$;

送り速度の値の範囲は、選択した計算分解能によって決まります。

マシンデータの標準の割り当ての場合:

MD10200 $MN_INT_INCR_PER_MM
(直線位置用の計算分解能) (1000 incr./mm)

または

MD10210 $MN_INT_INCR_PER_DEG

(回転位置用の計算分解能) (1000 incr./deg)

以下の値の範囲を、指定した分解能でプログラム指令できます。

軌跡速度 F とジオメトリ軸の数値範囲	
メトリック単位系:	インチ単位系:
0.001 ≤ F ≤ 999,999.999 [mm/min、mm/rev、deg/min、deg/rev]	0.001 ≤ F ≤ 399,999.999 [in/min、in/rev]

位置決め軸の送り速度に対応した数値範囲:	
メトリック単位系:	インチ単位系:
0.001 ≤ FA ≤ 999,999.999 [mm/min、mm/rev、deg/min、deg/rev]	0.001 ≤ FA ≤ 399,999.999 [in/min、in/rev]

主軸速度 S に対応した数値範囲:	0.001 ≤ S ≤ 999,999.999 [rpm]

計算分解能を係数で増減すると、それに応じて数値範囲も変化します。

8.2.2 移動範囲

移動範囲の数値

移動範囲の数値は、選択した計算分解能によって異なります。

マシンデータの標準の割り当ての場合:

MD10200 $MN_INT_INCR_PER_MM

(直線位置用の計算分解能) (1000 incr./mm.)

または

MD10210 $MN_INT_INCR_PER_DEG

(回転位置用の計算分解能) (1000 incr./deg.)

以下の値の範囲を、指定した分解能でプログラム指令できます。

表 8-1　軸の移動範囲

	G71 [mm、deg]	G70 [inch、deg]
	範囲	範囲
直線軸 X、Y、Z など	∓ 999,999.999	∓ 399,999.999
回転軸 A、B、C など	∓ 999,999.999	∓ 999,999.999
補間パラメータ I、J、K	∓ 999,999.999	∓ 399,999.999

回転軸の処理単位は必ず度です。

計算分解能を 10 倍ずつ増加/減少すると、これに対応して数値範囲が変化します。

移動範囲は、ソフトウェアリミットスイッチと作業領域により制限できます(「A3:軸監視機能 (ページ 101)」の章を参照してください)。

直線軸と回転軸の移動範囲が大きい場合特記事項については、「R1:レファレンス点復帰 (ページ 1497)」の章を参照してください。

回転軸の移動範囲は、マシンデータを使用して制限できます。

参照先:

『機能マニュアル、上級機能』;「回転軸(R2)」

8.2.3　コントロールシステムの位置決め精度

フィードバック値の分解能と計算分解能

制御装置の位置決め精度は、フィードバック値の分解能(=エンコーダパルス/(mm または度))と計算分解能(=内部分割数/(mm または度))によって違ってきます。

これらの 2 つの値の粗い分解能により、制御の位置決め精度が決まります。

最小入力単位、補間および位置制御周期の設定は、この精度に影響を及ぼしません。

制御装置は MD32000 を使用して制限するだけでなく、 次の式に従って、状況に応じて最大軌跡速度を制限します。

$$\frac{\text{内部インクレメント/mm}}{\text{エンコーダパルス/mm}} = \frac{1}{\text{ENC_RESOL [n] * ENC_PULSE_MULT[n]}} * \frac{\text{DRIVE_ENC_RATIO_NUMERA [n]}}{\text{DRIVE_ENC_RATIO_DENOM [n]}} * \frac{\text{DRIVE_AX_RATIO_DENOM [n]}}{\text{DRIVE_AX_RATIO_NUMERA [n]}} * \text{LEADSCREW_PITCH} * \text{INT_INCR_PER_MM}$$

8.2.4 最小入力/表示単位、計算分解能

最小単位: 違い

最小単位、たとえば、直線位置と回転位置、速度、加速度および加々速度の最小単位は、次のように識別してください。

- 最小入力単位
  データは、操作パネルまたはパートプログラムを介して入力されます。
- 最小表示単位
  データは操作パネルを介して表示されます。
- 計算分解能
  操作パネルまたはパートプログラムを介して入力されたデータは内部的に表現されます。

最小入力と表示単位は、使用する指定された操作パネルで決定されます。それによって、次のマシンデータ

MD9004 $MM_DISPLAY_RESOLUTION (最小表示単位)

で位置値の最小表示単位を変更できます。

マシンデータ

MD9011 $MM_DISPLAY_RESOLUTION_INCH (インチ単位系の最小表示単位)

インチ設定で位置値の最小表示単位を設定するのに使用できます。

これにより、インチ設定で小数第 6 位まで表示することができます。

パートプログラムをプログラミングする場合は、プログラミングガイドに記載された最小入力単位が適用されます。

マシンデータを使用して、必要な計算分解能を定義します。

MD10200 $MN_INT_INCR_PER_MM (直線位置の計算分解能)

および

MD10210 $MN_INT_INCR_PER_ DEG (回転位置の計算分解能)

最小入力/表示単位とは無関係ですが、最低でも同じ分解能を持つ必要があります。

パートプログラム内の位置データや速度などの小数点以下の最大桁数、および工具オフセットやゼロオフセットなどの小数点以下の桁数は(同時に、可能な最大精度の小数点以下の桁数も)、計算分解能により定義されます。

回転位置と直線位置の精度は、計算分解能を使ってプログラム指令値の積を整数に丸めることにより、計算分解能に制限されます。

丸めを明確にするために、計算分解能には、10 のべき乗を使用してください。

丸めの例

計算分解能: 1000 incr./mm

プログラム指令軌跡 97.3786 mm

実効値:97.379 mm

$^1/_{10}$ µm 領域でのプログラミングの例:

機械の直線軸はすべて、値 0.1 ～ 1000 µm の範囲内でプログラム指令され移動するものとします。

⇒ 正確に 0.1 µm に位置決めするために、計算分解能≥ 10^4 incr./mm に設定する必要があります。

⇒ MD10200 $MN_INT_INCR_PER_MM = 10000 [incr./mm]:

⇒ 関係するパートプログラムの例:

プログラムコード	コメント
N20 G0 X 1.0000 Y 1.0000	; 軸が位置に移動します。 X=1.0000 mm、Y=1.0000 mm;
N25 G0 X 5.0002 Y 2.0003	; 軸が位置に移動します。 X=5.0002 mm、Y=2.0003 mm

8.2.5 マシンデータとセッティングデータの物理量の単位変換

入/出力単位

物理量を持つマシンデータとセッティングデータは、選択された単位系がメトリック系かインチ系かに応じて、以下に示す各入/出力単位で解釈されます。

物理単位	標準基本単位系の入/出力単位	
	メトリック	インチ
直線位置	1 mm	1 in
回転位置	1 度	1 度
直線速度	1 mm/min	1 in/min
回転速度	1 rpm	1 rpm
直線加速度	1 m/s^2	1 in/s^2
回転加速度	1 $rev./s^2$	1 $rev./s^2$
直線加々速度	1 m/s^3	1 in/s^3
回転加々速度	1 $rev./s^3$	1 $rev./s^3$
時間	1 秒	1 秒
位置コントローラのサーボゲイン	1/s	1/s
毎回転送り速度	1 mm/rev	1 in/rev
補正値直線位置	1 mm	1 in
補正値回転位置	1 度	1 度

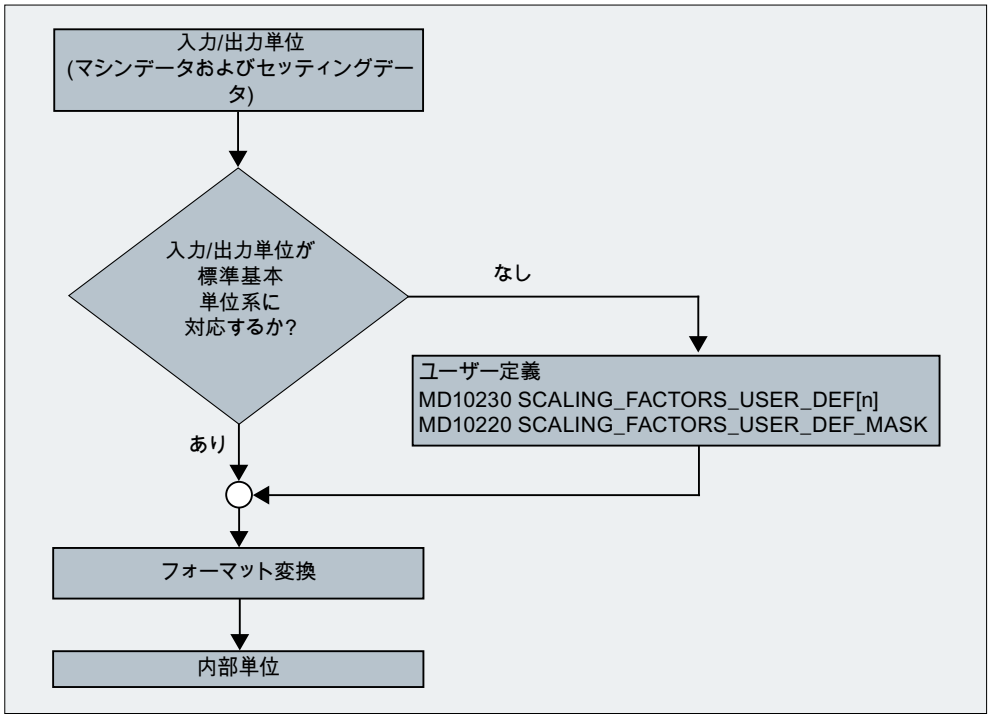
制御装置の内部では、選択された基本単位系にかかわらず、以下の単位が使用されます。

物理単位	単位
直線位置	1 mm
回転位置	1 度
直線速度	1 mm/s
回転速度	1 度/s
直線加速度	1 mm/s^2
回転加速度	1 $度/s^2$
直線加々速度	1 mm/s^3
回転加々速度	1 $度/s^3$

物理単位	単位
時間	1 秒
位置コントローラのサーボゲイン	1/s
毎回転送り速度	1 mm/度
補正値直線位置	1 mm
補正値回転位置	1 度

ユーザは、マシン/セッティングデータに対して別の入/出力単位を定義することができます。また、下記のマシンデータによって、新たに選択された入/出力単位と内部単位間の調整をおこなう必要があります。

- MD10220 $MN_SCALING_USER_DEF_MASK
- MD10230 $MN_SCALING_FACTORS_USER_DEF[n]



これは以下のように適用されます。

選択された入/出力単位 = (MD10230 $MN_SCALING_FACTORS_USER_DEF[n]) *内部単位

選択した I/O 単位は内部単位 1 mm、1 deg、1 s で表現され、マシンデータ MD10230 $MN_SCALING_FACTORS_USER_DEF[n]に入力します。

例 1:

直線速度のマシンデータ入/出力を、mm/min(初期状態)の代わりに、m/min にします。

(内部単位は mm/s です)

⇒ 直線速度の単位変換係数は、標準設定と異なる必要があります。この場合、マシンデータ

MD10220 $MN_SCALING_USER_DEF_MASK

には、ビット番号 2 を設定してください。

⇒ MD10220 $MN_SCALING_USER_DEF_MASK = 'H4'; (16 進数表示のビット番号 2)

⇒ 直線速度の単位変換係数は、標準設定と異なる必要があります。この場合、マシンデータ

MD10220 $MN_SCALING_USER_DEF_MASK

には、ビット番号 2 を設定してください。

⇒ MD10220 $MN_SCALING_USER_DEF_MASK = 'H4'; (16 進数表示のビット番号 2)

単位変換係数は、次の式を使用して計算されます。

$$\text{MD10230 SCALING_FACTORS_USER_DEF[n]} = \frac{\text{選択された入力/出力単位}}{\text{内部単位}}$$

$$\text{MD10230 SCALING_FACTORS_USER_DEF[n]} = \frac{1\ \frac{\text{M}}{\text{分}}}{1\ \frac{\text{mm}}{\text{s}}} = \frac{\frac{1000\ \text{mm}}{60\ \text{s}}}{1\ \frac{\text{mm}}{\text{s}}} = \frac{1000}{60} = 16{,}667;$$

⇒ MD10230 SCALING_FACTORS_USER_DEF[n] = 16,667

インデックス n は「物理量の単位変換係数」リストで「直線速度」を定義します。

例 2:

例 1 の変更に加えて、直線加速度のマシンデータ入/出力は、m/s²(初期状態)ではなく、ft/s² にする必要があります。
(内部単位は mm/s² です。)

⇒ MD10220 SCALING_USER_DEF_MASK = H14 ; (例1のビット番号4およびビット番号2の16進値)

⇒ MD10220 SCALING_FACTORS_USER_DEF[4] = $\frac{1\ \frac{ft}{s^2}}{1\ \frac{mm}{s^2}} = \frac{12*25{,}4\ \frac{mm}{s^2}}{1\ \frac{mm}{s^2}} = 304{,}8$;

⇒ MD10220 SCALING_FACTORS_USER_DEF[4] = 304,8

インデックス 4 は「物理量の単位変換係数」リストで「直線加速度」を定義します。

8.3 単位系、メトリック/インチ

8.3.1 機能

8.3.1.1 パラメータ設定およびプログラムされた単位系

SINUMERIK コントロールシステムは、メトリック単位系とインチ単位系で動作できます。

パラメータ設定された単位系(基本単位系)

単位系(基本単位系)の初期設定は、マシンデータ MD10240 $MN_SCALING_SYSTEM_IS_METRIC で設定します(「セットアップ (ページ 419)」を参照)。

長さに関連するすべてのデータは、基本単位系に応じて、メトリックまたはインチ単位で解釈されます。

メトリック単位系:	mm、mm/min、m/s2、m/s3、mm/rev.
インチ単位系:	inch、inch/min、inch/s2、inch/s3、inch/rev.

また基本単位系では、直線軸でプログラムされた F 値の解釈を定義します。

送り速度タイプ	メトリック単位系	インチ単位系
G94	mm/min	inch/min
G95	mm/rev	inch/rev.

プログラム指令単位系

パートプログラムで G グループ 13(単位系、インチ/メトリック)の命令を使用することで、メトリック単位系とインチ単位系を切り替えることができます(「プログラミング (ページ 423)」を参照)。

プログラムされた単位系と基本単位系は、いつも同じであっても、また異なっていてもかまいません。パートプログラムの特定の区間内で単位系を切り替えると、メトリック単位系でのワークでインチ単位のねじ山の処理などを行うことができます。

G グループのリセット設定は、マシンデータ MD20150 $MC_GCODE_RESET_VALUES で定義します(「セットアップ (ページ 419)」を参照)。

HMI 表示

長さに関連するデータは、ユーザーインタフェースで以下のように表示されます。

	表示	
	基本座標系の場合	プログラムされた単位系
マシンデータ	x	
機械座標系のデータ	x	
工具データ	x	
ゼロオフセット	x	
ワーク座標系のデータ		x

NC/PLC インタフェース

寸法情報(軌跡軸と位置決め軸用の送り速度など)を含む NC/PLC のインタフェース信号の場合は、設定した基本単位系で PLC によってデータ交信が実行されます。

外部パーツプログラムの読み取り

基本単位系とは異なる単位系でプログラムされたデータセット(ゼロオフセット、工具オフセットなど)を含むパートプログラムを外部メモリから読み込む場合は、まず、基本単位系を変更してください。

8.3.1.2 拡張単位系機能

SW5 以上で、単位系の機能が拡張され、単位系の切り替えが大幅に簡素化されています。

この機能の内容は次のとおりです。

- ユーザーインタフェースでの単位系の切り替え (ページ 416)
- 新しい G 命令 G700/G710 (「プログラミング (ページ 423)」を参照)
- 単位系の識別子 INCH/METRIC を含むデータバックアップ(「セットアップ (ページ 419)」を参照)
- 単位系の変更時の自動データ変換。たとえば、以下のデータが対象になります。
  - ゼロオフセット
  - 補正データ(EEC、QEC)
  - 工具補正
  - ...

互換性の理由から、拡張された単位系の機能はマシンデータを使用して起動する必要があります(「セットアップ (ページ 419)」を参照)。

8.3.1.3 ユーザーインタフェースでの単位系の切り替え

必要条件

拡張された単位系の機能が有効(MD10260 $MN_CONVERT_SCALING_SYSTEM = 1。「セットアップ (ページ 419)」を参照)

機能

HMI の[機械]操作エリアの該当するソフトキーを使用して、コントロールシステムの単位系を変更できます。

単位系は、以下の条件の下でのみ切り替わります。

- MD20110 $MC_RESET_MODE_MASK、ビット 0 がすべてのチャネルで設定されている。
- すべてのチャネルがリセット状態である。
- 軸が現在、JOG、DRF、PLC のいずれでも移動していない。
- 砥石周速度一定制御(GWPS)が有効ではない。

単位系の切り替えの間は、パートプログラム開始やモード変更などのアクションが無効になります。

単位系の実際の変更は、必要なマシンデータをすべて書き込み、続いて、それらのデータを有効にすることで実行されます。

以下のマシンデータは、すべての設定済みチャネルに対して自動的に一貫して切り替えられます。

- MD10240 $MN_SCALING_SYSTEM_IS_METRIC
- MD20150 $MN_GCODE_RESET_VALUES

位置をリセット

ユーザーインタフェースから単位系を切り替えると、G グループ 13 のリセット設定が自動的に現在の単位系に調整されます。

MD20150 $MC_GCODE_RESET_VALUES[12] = <現在の単位系>

NCU リンク

注記

複数の NCU が NCU リンクによってリンクされる場合、切り替えは、リンクされたすべての NCU で同じ影響があります。接続された NCU のいずれかで切り替えの必要条件が満たされない場合、どの NCU に対しても切り替えは行われません。NCU リンクが存在する場合に既存の NCU に対して複数の NCU 間の補間が実行されると想定されるため、同じ単位系を使用する場合のみ、正しい補間結果を得ることができます。

参照先:

『機能マニュアル、上級機能』;「B3:分散システム」

システムデータ

単位系が切り替わると、ユーザーには、 長さに関連する指定がすべて、新しい単位系に自動的に変換されたように見えます。

以下が対象になります。

- 位置
- 送り速度
- 加速度
- 加々速度
- 工具補正
- プログラマブル、設定可能オフセット、外部ゼロオフセット、DRF オフセット
- 補正値
- プロテクションゾーン
- マシンデータ
- JOG と手動パルス発生器の係数

切り替え後は、上記のデータをすべて物理的な量で使用できます。

固有の物理単位が定義されていないデータは、自動的には変換**されません**。

以下が対象になります。

- R 変数
- GUD(**G**lobal **U**ser **D**ata、グローバルユーザデータ)
- LUD(**L**ocal **U**ser **D**ata、ローカルユーザデータ)
- PUD(**P**rogram global **U**ser **D**ata、プログラムグローバルユーザーデータ)
- アナログ入力/出力
- FC21 によるデータ交信

現在の有効な単位系 MD10240 $MN_SCALING_SYSTEM_IS_METRIC を考慮するよう求められます。

NC/PLC インタフェース

現在の単位系の設定は、次の信号を使用して NC/PLC インタフェースで読み取ることができます。

DB10 DBX107.7 (インチ単位系)

制御装置の電源投入時以降の単位系の切り替え回数を次の信号バイトを使用して読み取ることができます。

DB10 DBB71 (カウンタの変更、インチ/メトリック単位系)

制御装置電源投入後の初期値:1

8.3.2 セットアップ

NC 別マシンデータ

基本システム

開ループ制御によって使用される物理変数に応じたスケーリング長の基本単位系。マシンデータの監視が事前定義されている場合に、データ入力/出力のサイズ。

MD10240 $MN_SCALING_SYSTEM_IS_METRIC

値	意味
0 (= FALSE)	長さに関連するデータは、**インチ**データとして解釈されます。
1 (= TRUE)	長さに関連するデータは、**メトリック**データとして解釈されます。

このマシンデータの変更後、システムは電源を再投入する必要があります。そうしない場合、物理単位を含む関連マシンデータが不適切にスケーリングされます。

次の手順を順守する必要があります。

- 手動入力を使用した MD の変更:
  - 1.システムの電源を投入します。
  - 2.物理単位を含むマシンデータ:値を入力します。
- MD はマシンデータを使用して変更します。
  - 1.システムの電源を投入します。
  - 2.新しい物理単位を考慮するために、マシンデータファイルをもう一度ロードします。

MD10240 の変更時に、アラーム 4070「正規化マシンデータが変更されました」が出力されます。

注記

コントロールシステムの単位系を切り替えるときは、長さに関連するすべてのデータを一貫して、完全に他の単位系に変換してください。

変換係数(NC 別)

メトリック単位系からインチ単位系に変換するための係数は、次のマシンデータで設定します。

MD10250 $MN_SCALING_VALUE_INCH (インチの変換係数)

初期値:25.4

メトリック以外の基本単位系(MD10240 $MN_SCALING_SYSTEM_IS_METRIC = 0)を選択すると、変換係数が有効になります。

次のデータは、変換係数により乗算されます。

- 入力/出力の長さ関連のデータ(マシンデータ、ゼロオフセットのアップロード時など)
- 直線軸でプログラムされた F 値
- G70/G71 によりプログラムされた単位系が基本単位系(MD10240 $MN_SCALING_SYSTEM_IS_METRIC)と異なる場合、プログラムされた軸のジオメトリ(位置、多項式係数、円弧の半径のプログラミングなど)。

注記

変換係数を変更して、コントロールシステムも、お客様固有の単位系に合わせて設定することができます。

拡張単位系機能

拡張された単位系の機能は、次の互換性マシンデータを使用して起動できます。

MD10260 $MN_CONVERT_SCALING_SYSTEM

値	意味
0 (= FALSE)	拡張された単位系の機能が**無効**です(初期設定; 以前のソフトウェアのリリースに対する互換性)。
1 (= TRUE)	拡張単位系機能が有効です

MD10240 の変更時に、アラーム 4070「正規化マシンデータが変更されました」が出力されます。

データバックアップ

コントロールシステムから読み取られ、単位系に影響されるデータを含む個々のデータセットには、読み取り時に、現在の単位系の設定を示す識別子が割り当てられます。

- MD10260 $MN_CONVERT_SCALING_SYSTEM
- MD10240 $MN_SCALING_SYSTEM_IS_METRIC

識別子には、どの単位系でデータが読み取られたかが記録されます。これにより、すべてのデータセットが確実に、現在設定されている単位系で制御装置に読み取られます。

識別子もパートプログラム内で評価されるため、上記のように操作ミスから「保護」することもできます。したがって、たとえばメトリックデータだけを含むパートプログラムがインチ系で実行されるなどの事態を防止することができます。

アーカイブおよびマシンデータセットには、以下の設定の場合は下位互換性があります。

MD11220 $MN_INI_FILE_MODE = 2

注記

インチ/メトリック識別子が生成されるのは、以下の互換マシンデータの場合のみです。

MD10260 $MN_CONVERT_SCALING_SYSTEM = TRUE

注記

マシンデータの丸め処理

丸め処理の問題を解消するために、インチ単位系で書き込むときは、長さ関連のマシンデータはすべて、最も近い 1 pm に丸められます(MD10240 $MN_SCALING_SYSTEM_IS_METRIC=0 と MD10260 $MN_CONVERT_SCALING_SYSTEM=1)。

これにより、インチ単位系でのデータバックアップの読み取り時に ASCII への変換により発生する精度の低下が、データがシステムに戻されるときに修正されます。

最小入力単位と計算分解能

入力と演算の最小単位は、ミリメートル毎のインクレメントの内部数として定義されます。

MD10200 $MN_INT_INCR_PER_MM

直線位置の入力の精度は、演算の最小単位までに制限されます。プログラム指令された位置の値と演算の最小単位の積は、その次の整数に丸められます。丸めをわかりやすく明確にするために、計算分解能には、10 のべき乗を使用してください。

例:

1 インチ= 25.4 mm ⇒ 0.0001 インチ= 0.00254 mm = 2.54 µm = 2540 nm

最後の 40 nm をプログラム指令して表示するには、入力と演算の最小単位として値 100,000 を設定してください。

このように両方の単位系で同じ設定をしたときだけ、精度を大きく損なうことなく単位系を変更できます。一度 MD10200 を上記の値に設定すれば、単位系を切り替えるたびに変更する必要はありません。

位置決めの単位系の表

割り出し軸テーブルの位置データとポジションスイッチの切り替え点用の単位系は、下記のマシンデータで設定されます。

MD10270 $MN_POS_TAB_SCALING_SYSTEM

参照先:
機能マニュアル、上級機能、章「N3:ソフトウェアカム、ポジションスイッチ信号」および「T1:軸割り出し」

ユーザー工具データ

カスタマイズされた工具の物理単位と工具の刃先データは、以下のマシンデータで設定できます。

- MD10290 $MN_CC_TDA_PARAM_UNIT
- MD10292 $MN_CC_TOA_PARAM_UNIT

注記

単位系が切り替わると、長さに関連する工具データがすべて、新しい単位系に変換されます。

チャネルマシンデータ

位置をリセット

次のマシンデータで、チャネルごとに、G グループ 13 のリセット設定(単位系、インチ/メトリック)を設定します。

MD20150 $MC_GCODE_RESET_VALUES[12] (G グループ 13 のリセット設定)

軸/主軸用のマシンデータ

変換係数(軸別)

G70/G71 とは無関係の純粋な位置決め軸を作成するために、次のマシンデータでメトリック単位系からインチ単位系への変換係数を設定できます。

MD31200 $MA_SCALING_FACTOR_G70_G71 (G70/G71 が動作中の場合の変換係数)

初期値:25.4

注記

この変換係数は、3 つのジオメトリ軸すべてで同じであることが必要です。

ジョグとハンドル係数

次のマシンデータは、2 つの各単位系の軸別インクレメント加重係数を含む 2 つの値から成ります。

MD31090 $MA_JOG_INCR_WEIGHT

MD10240 SCALING_SYSTEM_IS_METRIC の現在の設定に応じて、コントロールシステムでは自動的に適切な値が選択されます。

ユーザーはシステムのセットアップ時に 2 つのインクレメント係数を定義します。

例:1 番目の軸のインクレメント係数

- メトリックでは:
  MD31090 $MA_JOG_INCR_WEIGHT[0 ; AX1] = 0.001 mm
- インチでは:
  MD31090 $MA_JOG_INCR_WEIGHT[1 ; AX1] = 0.00254 mm ≙ 0.0001 インチ

この定義により、インチ/メトリックの切り替えのたびに MD31090 を変更しなくても済みます。

すべての内部位置が常に mm で表されるため、単位系が変更された場合に、ジョグによる相対移動中に残りの距離は蓄積されません。

真直度補正の単位系

真直度補正用の単位系は、以下のマシンデータを使用して設定します。

MD32711 $MA_CEC_SCALING_SYSTEM_METRIC

参照先:
機能マニュアル、上級機能、章「K3:補正:

8.3.3 プログラミング

8.3.3.1 単位系の切り替え(G70/G71/G700/G710)

パートプログラムで G グループ 13(インチ/メトリック単位系)の命令を使用することで、メトリック単位系とインチ単位系を切り替えることができます。

起動

命令 G700 および G710 が使用可能であるために、拡張された単位系の機能をオンにする必要があります(MD10260 $MN_CONVERT_SCALING_SYSTEM = 1)。

構文

```
G70
G71
G700
G710
```

意味

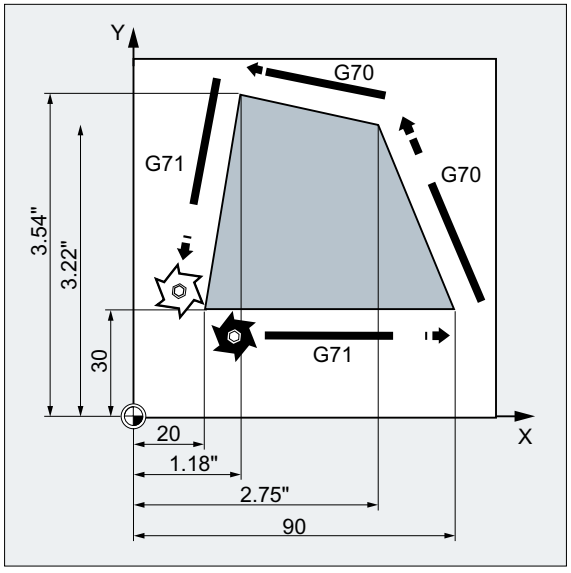
G70:	インチ単位系の機動 **インチ単位系**を使用すると、**インチ系の長さの単位で、ジオメトリデータ**の読み取りと書き込みができます。 **長さの単位の加工条件**(たとえば、送り速度、工具オフセット、設定可能ゼロオフセット、マシンデータ、システム変数など)は、**パラメータ設定された基本単位系**を使用して読み取ることができます。	
	G グループ:	13
	初期設定:	MD20150 $MC_GCODE_RESET_VALUES を使用して設定可能
	有効性:	モーダル
G71:	メトリック単位系の機動 **メトリック単位系**を使用すると、**メトリック系の長さの単位で、ジオメトリデータ**の読み取りと書き込みができます。 **長さの単位の加工条件**(たとえば、送り速度、工具オフセット、設定可能ゼロオフセット、マシンデータ、システム変数など)は、**パラメータ設定された基本単位系**を使用して読み取ることができます。	
	G グループ:	13
	初期設定:	MD20150 $MC_GCODE_RESET_VALUES を使用して設定可能
	有効性:	モーダル
G700:	インチ単位系の機動 すべてのジオメトリデータと加工データの長さの単位は、インチ単位系を使用して読み取りと書き込みが行われます。	
	G グループ:	13
	初期設定:	MD20150 $MC_GCODE_RESET_VALUES を使用して設定可能
	有効性:	モーダル

G710:	メトリック単位系の機動 すべてのジオメトリデータと加工データの長さの単位は、メトリック単位系を使用して読み取りと書き込みが行われます。	
	G グループ:	13
	初期設定:	MD20150 $MC_GCODE_RESET_VALUES を使用して設定可能
	有効性:	モーダル

通知
回転軸の軸別のデータ 回転軸の軸別のデータは、パラメータ設定された基本単位系を使用して読み取りと書き込みが行われます。

例

基本単位系はメトリックです(MD10240 $MN_SCALING_SYSTEM_IS_METRIC = 1)。ただし、ワーク図の寸法はインチで表されます。このことが、パートプログラムでインチ単位系を選択した理由です。インチ寸法の処理後、メトリック単位系が再度選択されます。



プログラムコード	**コメント**
N10 G0 G90 X20 Y30 Z2 S2000 M3 T1	; X=20 mm、Y=30 mm、Z=2 mm、F=早送り mm/min
N20 G1 Z-5 F500	; Z=-5 mm、F=500 mm/min

プログラムコード	コメント
N30 X90	; X=90 mm
N40 **G70** X2.75 Y3.22	; プログラム指令単位系:inch
	; X=2.75 インチ、Y=3.22 インチ、F=500 mm/min
N50 X1.18 Y3.54	; X=1.18 inch、Y=3.54 inch、F=500 mm/min
N60 **G71** X20 Y30	; プログラム指令単位系:メトリック
	; X=20 mm、Y=30 mm、F=500 mm/min
N70 G0 Z2	; Z=2 mm、F=早送り mm/min
N80 M30	; プログラム終了

他の情報

G70/G71 と G700/G710 の場合のデータの読み取りと書き込み

データ領域	G70/G71		G700/G710	
	読み取り	書き込み	読み取り	書き込み
表示、小数位(WCS)	P	P	P	P
表示、小数位(MCS)	G	G	G	G
送り速度	G	G	P	P
位置データ(X、Y、Z、...)	P	P	P	P
補間パラメータ I、J、K	P	P	P	P
円弧半径(CR)	P	P	P	P
極半径(RP)	P	P	P	P
ねじピッチ	P	P	P	P
プログラマブルフレーム	P	P	P	P
設定可能フレーム	G	G	P	P
基本フレーム	G	G	P	P
外部ゼロオフセット	G	G	P	P
軸プリセットオフセット	G	G	P	P
作業領域リミット(G25/G26)	G	G	P	P
プロテクションゾーン	P	P	P	P
工具補正	G	G	P	P
長さ関連のマシンデータ	G	G	P	P
長さ関連のセッティングデータ	G	G	P	P
長さ関連のシステム変数	G	G	P	P
GUDs	G	G	G	G

データ領域	G70/G71		G700/G710	
	読み取り	書き込み	読み取り	書き込み
LUD	G	G	G	G
PUD	G	G	G	G
R 変数	G	G	G	G
Siemens サイクル	P	P	P	P
ジョグ/ハンドル分割係数	G	G	G	G
P:読み取り/書き込みがプログラム指令単位系で実行されます。 G:読み取り/書き込みが、設定した基本単位系で実行されます。				

シンクロナイズドアクション

注記

シンクロナイズドアクションでの位置データの読み取り

単位系をシンクロナイズドアクションで明示的にプログラム指令していない(条件要素と動作要素のいずれかまたは両方)場合は、シンクロナイズドアクション内の**長さに関連する位置データ**を常に、パラメータで設定した**基本単位系**で読み取ります。

参照先:『機能マニュアル、シンクロナイズドアクション』

8.4 指令/フィードバック回路

8.4.1 一般情報

制御ループ

次の構造を持つ制御ループを、すべての閉ループ制御の軸/主軸について設定できます。

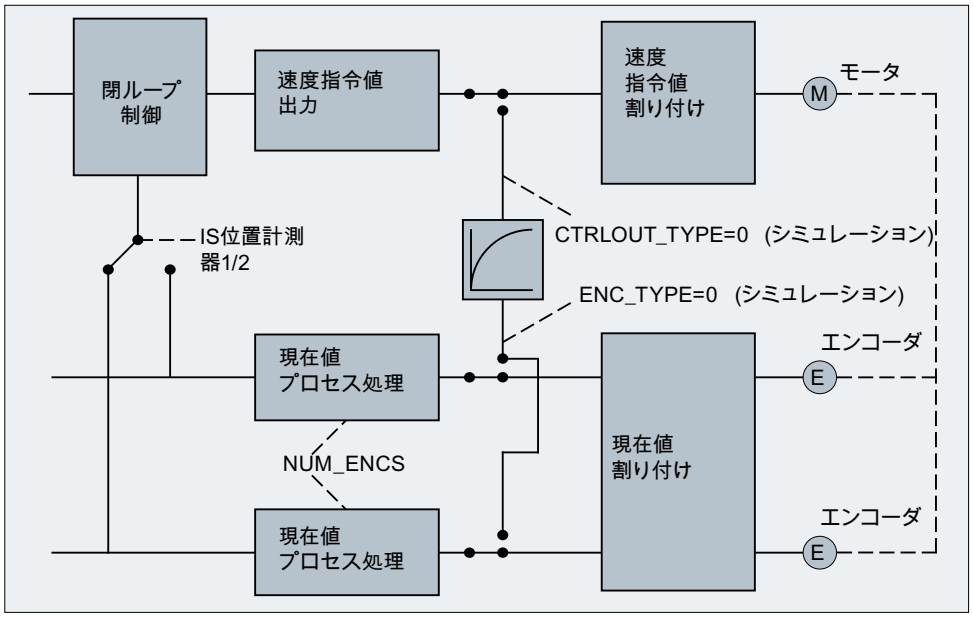


図 8-1 制御ループのブロック図

指令値出力

指令値メッセージを軸/主軸毎に出力することができます。駆動部への指令値出力は、SINUMERIK 840D sl から実現されます。

フィードバック値の取得

軸/主軸毎に、検出器を最大 2 つ接続できます。たとえば、高い精度が求められる加工処理用の直接検出器と、高速位置決め処理用の間接検出器です。

使用するエンコーダの数は、次のマシンデータに入力します。

MD30200 $MA_NUM_ENCS (エンコーダの数)

フィードバック値回路が 2 つある場合、フィードバック値は両方の回路で取得されます。

位置制御、アブソリュート値計算および表示には、有効な検出器が常に使用されます。PLC インタフェースによって両方の検出器が同時に有効になった場合は、コントローラによって位置決め検出器 1 が内部的に選択されます。

レファレンス点復帰は、選択された検出器によって実行されます。

すべての位置検出器を個別に原点確立してください。

エンコーダ監視機能の詳細については、「A3:軸監視機能 (ページ 101)」を参照してください。

フィードバック値取得用の補正機能については、以下を参照してください。

参照先:

『機能マニュアル、上級機能』;「誤差補正(K3)」

検出器の切り替え

次の NC/PLC インタフェース信号を介して 2 つの検出器を切り替えることができます。

DB31、... DBX1.5 (位置検出器 1)

DB31、... DBX1.6 (位置検出器 2)

詳細については、「A2: NC/PLC 間の各種共通インタフェース信号と各種機能 (ページ 45)」の章を参照してください。

検出器の切り替えは随時可能で、切り替えのために軸を停止させる必要はありません。ただし、切り替えができるのは、フィードバック値と 2 つの検出器とのずれが許容値を超えない場合に限られます。

該当する許容範囲は、次のマシンデータで入力します。

MD36500 $MA_ENC_CHANGE_TOL (位置フィードバック値の切り替えの最大許容範囲)

切り替え時、位置検出器 1 と 2 の現在のずれが直ちに処理されます。

監視機能

2 つの検出器のフィードバック値間の許容誤差は、次のマシンデータで入力します。

MD36510 $MA_ENC_DIFF_TOL

使用する 2 つの検出器の周期的な比較のために、この差分を超えてはなりません。超えた場合は、アラーム 25105 「エンコーダ位置が許容範囲を超えました」が生成されます。

軸が、少なくとも現在の制御検出器において、原点確立されていない場合、MD36510 = 0 であれば、関連する監視は無効です。軸の検出器が両方とも無効の場合も同様です。

フィードバック値取得のタイプ

使用するエンコーダのタイプは、次のマシンデータを使用して定義してください。

MD30240 $MA_ENC_TYPE (フィードバック値取得のタイプ(位置フィードバック値))

シミュレーション軸

テスト目的で軸の速度制御ループをシミュレションできます。

実際の軸と同様、シミュレーション軸は追従誤差を伴って「移動」します。

シミュレーション軸は、次の 2 つのマシンデータを「0」に設定することで定義します。

MD30130 $MA_CTRLOUT_TYPE[n] (指令値の出力値)

MD30240 $MA_ENC_TYPE[n] (フィードバック値取得のタイプ)

標準マシンデータが読み込まれるとすぐに、該当する軸がシミュレーション軸になります。

指令値とフィードバック値は、レファレンス点復帰によりレファレンス点の値に設定できます。

マシンデータ

MD30350 $MA_SIMU_AX_VDI_OUTPUT (シミュレーション軸用軸信号の出力)

を使用して、シミュレーション中に軸別インタフェース信号を PLC に出力するかどうかを定義できます。

フィードバック値更新

位置制御用に選択されたエンコーダに基づいて NC によりおこなわれるフィードバック値更新が、同じ軸に定義された別のエンコーダのフィードバック値に影響を及ぼさない場合には、次のマシンデータを使用して、このエンコーダを「独立」として宣言しなければなりません。

MD30242 $MA_ENC_IS_INDEPENDENT

フィードバック値更新には、次の処理が含まれます。

- モジュロ処理
- レファレンス点復帰

- 検出器の比較
- PRESET

8.4.2 指令値とエンコーダの割り当て

指令値のマーシャリング

以下のマシンデータは、機械軸の指令値の割り当てに関係します。

MD30100		$MA_CTRLOUT_SEGMENT_NR
指令値割り当て:バスセグメント		
システム	値	意味
840D sl	5	PROFIBUS DP/PROFINET (初期設定)

MD30110		$MA_CTRLOUT_MODULE_NR
指令値割り当て:ドライブ番号/モジュール番号		
システム	値	意味
840D sl	x	ドライブの論理 I/O アドレスは、MD13050 $MN_DRIVE_LOGIC_ADDRESS[n]からドライブ番号を介して割り当てられます。 ドライブ番号(x)は、MD13050 のインデックス(n)から生成されます。 x = n + 1 **注記** ドライブがシミュレートされている場合(MD30130 $MA_CTRLOUT_TYPE = 0)、マシンデータは意味を持ちません。

MD30120		$MA_CTRLOUT_NR
指令値割り当て:ドライブモジュール/モジュールでの指令値出力		
システム	値	意味
840D sl	1	PROFIdrive プロファイルを持つ PROFIBUS / PROFINET のモジュール式ドライブ(初期設定)

MD30130		$MA_CTRLOUT_TYPE
指令値の出力タイプ		
システム	値	意味
840D sl	0	シミュレーション(ドライブなしの運転)
	1	指令値出力が有効

エンコーダ割り当て

以下のマシンデータは、ドライブのエンコーダ情報(PROFIdrive メッセージで転送された)を機械軸のエンコーダ入力に割り当てるのに関係します。

MD30210		$MA_ENC_SEGMENT_NR[n]
フィードバック値の割り当て、バスセグメント		
システム	値	意味
840D sl	5	PROFIBUS DP/PROFINET

MD30220		$MA_ENC_MODULE_NR[n]
フィードバック値の割り当て:ドライブ番号/検出回路番号		
システム	値	意味
840D sl	x	ドライブの論理 I/O アドレスは、MD13050 $MN_DRIVE_LOGIC_ADDRESS[n]からドライブ番号を介して割り当てられます。 ドライブ番号(x)は、MD13050 のインデックス(n)から生成されます。 x = n + 1

MD30230		$MA_ENC_INPUT_NR[n]
フィードバック値の割り当て:ドライブモジュール/検出回路モジュールでの入力		
システム	値	意味
840D sl	x	PROFIdrive メッセージ内のエンコーダインタフェースの番号 例 PROFIdrive メッセージ 103 x = 1 → 1 番目のエンコーダインタフェース(G1_ZSW, G1_XIST1, G1_XIST2) x = 2 → 2 番目のエンコーダインタフェース(G2_ZSW, G2_XIST1, G2_XIST2) PROFIdrive メッセージ 118 x = 1 → 1 番目のエンコーダインタフェース(G2_ZSW, G2_XIST1, G2_XIST2) x = 2 → 2 番目のエンコーダインタフェース(G3_ZSW, G3_XIST1, G3_XIST2) 注： SINAMICS S120: - エンコーダ 1 (G1_...)の場合:モータエンコーダ - エンコーダ 2 (G2_...):直接検出器 - エンコーダ 3 (G3_...):その他の検出器

MD30240		$MA_ENC_TYPE[n]
フィードバック値取得のエンコーダタイプ(位置フィードバック値)		
システム	値	意味
840D sl	0	シミュレーション(エンコーダなしの運転)
	1	インクレメンタルエンコーダ
	4	絶対値エンコーダ
注記: PROFIdrive パラメータ p979 に対応します。		

MD30242		$MA_ENC_IS_INDEPENDENT[n, 軸]
エンコーダは独立している		
システム	値	意味
840D sl	0	エンコーダは独立していない
	1	エンコーダは独立している 位置制御用に選択されたエンコーダに対して行うフィードバック値更新が、同じ軸で定義された 2 番目のエンコーダのフィードバック値に影響しない場合、このエンコーダは「独立」として宣言する必要があります。 フィードバック値更新は、以下の通りです。 ● - モジュロ処理 ● - レファレンス点復帰 ● - 検出器の調整 ● - PRESET 例:1 つの軸、2 つのエンコーダで、2 番目のエンコーダが独立している MD30200 $MA_NUM_ENCS[AX1] = 2 MD30242 $MA_ENC_IS_INDEPENDENT[0, AX1] = 0 MD30242 $MA_ENC_IS_INDEPENDENT[1, AX1] = 1 選択、位置検出器 1/2:DB31.DBX1.5 / 1.6 エンコーダ 1 が閉ループ位置制御用に選択されている場合、エンコーダ 2 が独立しているために、フィードバック値更新はエンコーダ 1 でだけ行われます。 エンコーダ 2 が閉ループ位置制御用に選択されている場合、エンコーダ 1 は独立していないため、フィードバック値更新は両方のエンコーダで行われます。 つまり、マシンデータは機械軸の不使用エンコーダにだけ有効です。
	2	不使用エンコーダは独立していません。 エンコーダのフィードバック値は、有効なエンコーダによって変更されます。レファレンス点復帰の場合、MD35102 $MA_REFP_SYNC_ENCS = 1 と組み合わせて、不使用エンコーダを有効なエンコーダに合わせますがレファレンス復帰はしません。

MD30242		$MA_ENC_IS_INDEPENDENT[n, 軸]
エンコーダは独立している		
システム	値	意味
		指令モード MD34200 $MA_ENC_REFP_MODE = 3 (絶対番地化原点マーク)では、不使用エンコーダは原点マーク距離の通過後に、次の移動動作で自動的にレファレンス復帰します。これは、実際の運転モードの設定とは無関係に行われます。
	3	エンコーダは独立している モジュロ回転軸の場合、不使用エンコーダでモジュロフィードバック値更新も行われます。

注記

マシンデータインデックス[n]

エンコーダ割り当てのためのマシンデータインデックス[n]の意味は、次の通りです。

- n = 0:1 番目のエンコーダが機械軸に割り当てられています。
- n = 1:2 番目のエンコーダが機械軸に割り当てられています。

割り当ては、次のマシンデータを使用して行われます。

- MD30220 $MA_ENC_MODULE_NR[n]
- MD30230 $MA_ENC_INPUT_NR[n]

参照先

『試運転マニュアル、CNC 試運転:NC、PLC、ドライブ』
「NC とドライブ間の通信」の「ドライブ:軸の割り当て」の章

8.4.3 モータ/負荷側減速比の設定

ギヤタイプ

次のギヤタイプを使用して、機械的減速比を設定できます。

ギヤタイプ	適用タイミング	設定	据え付け場所
モータ/負荷側ギヤ	パラメータ設定	固定値	ギヤユニット
エンコーダ検出器ギヤ	電源投入	センサに依存	センサ側
負荷中間ギヤユニット	ウォーム再起動	負荷に依存	工具側

ギヤユニット/エンコーダの位置

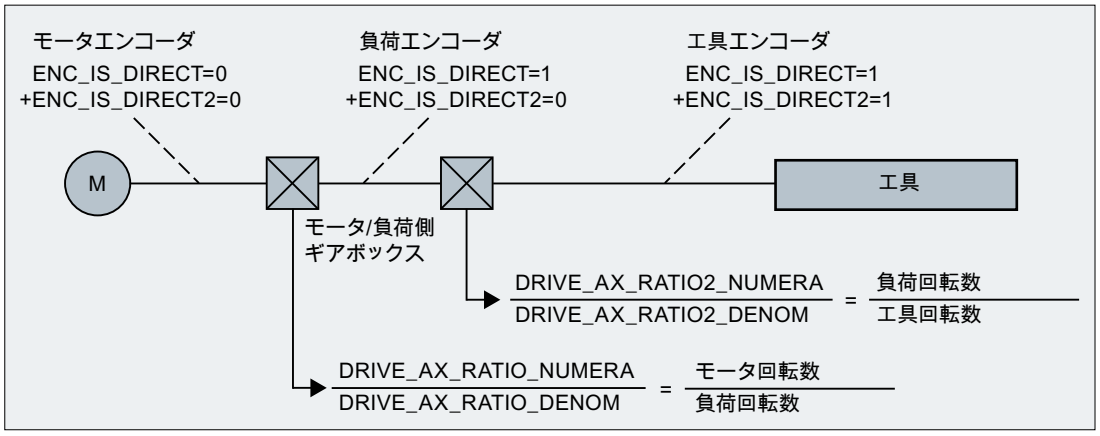


図 8-2 ギヤユニットタイプとエンコーダの位置

モータ/負荷側ギヤ

SINUMERIK がサポートするモータ/負荷側ギヤは、次のマシンデータを使用して設定します。

- MD31060 $MA_DRIVE_AX_RATIO_NUMERA (負荷側ギヤ分子)
- MD31050 $MA_DRIVE_AX_RATIO_DENOM (負荷側ギヤ分母)

伝達比は、両方のマシンデータの分子/分母の比率として計算されます。関連する伝達比に位置コントローラを同期させるために、対応するパラメータセットがコントロールシステムで既定値として自動的に使用されます。

ギヤ選択切替が自動的に実行されるとは限らないため、また、ギヤ選択切替を要求する方法は複数存在するため、必ずしも位置コントローラにパラメータセットが取り込まれるわけではありません。

注記

ギヤ選択切替のパラメータセットについて詳しくは、「S1:主軸 (ページ 1555)」の章を参照してください。

中間ギヤ

また、コントロールシステムは、構成可能な負荷中間ギヤもサポートしています。

- MD31066 $MA_DRIVE_AX_RATIO2_NUMERA (中間ギヤ分子)
- MD31064 $MA_DRIVE_AX_RATIO2_DENOM (中間ギヤ分母)

一般に駆動工具には「独自の」中間ギヤがあります。有効な中間ギヤとモータ/負荷側ギヤを増やすことで、このような可変機構を構成できます。

⚠ 注意

切り替えに対して異なるギヤ伝達比

モータ/負荷側ギヤとは異なり、中間ギヤ用のパラメータセットは存在しません。そのため、パートプログラムまたは PLC (NC/PLC インタフェース)に対して、時間的な同期をとって切り替えを制御する方法もありません。したがって、ギヤ切替中の部品加工は許可されません。当該の機械的切り替えに対応するマシンデータの変更の同期をとり、それを有効にする処理は、ユーザー側でおこなってください。移動中の切り替え時には、単位変換係数が不連続変化するため、補正を除外することは**できません**。このとき、最高加速度の超過は監視されません。

工具に直接接続されるエンコーダ

以下のマシンデータを設定して、中間ギヤの「工具側エンコーダ」に別の接続オプションを設定することができます。

MD31044 $MA_ENC_IS_DIRECT2

工具に直接接続されないエンコーダ

次の必要条件が、位置制御モードでの中間ギヤのギヤ切替に適用されます。変更されるギヤ比が、この場合のエンコーダ情報の再単位変換に取り込まれます。

この場合、位置決めモードでの軸/主軸に以下が適用されます。

- 滑らかなギヤ切替は、**停止状態でのみ可能です**。
  このために、減速比の変更に対して、ギヤ切替の前後における工具側位置が、同じに設定されます。ギヤ選択切替中に機械的位置は変化しない(またはほどんど変化しない)からです。
  推奨事項:
  21612「移動中のコントローライネーブルリセット」を避けるために、切り替えは「停止状態でのみ」実行してください。ギヤ切替の前または最中に軸または主軸を速度制御モードまたはフォローアップモードに切り替えることは可能で、応急的措置として使用できます。

境界条件

位置制御に使用するエンコーダが**工具に直接接続されている場合**、ギヤ選択切替は、モータ/負荷側ギヤのドライブと NC の間の速度インタフェースにおける物理量にのみ影響します。内部パラメータセットは変更されません。

レファレンス点と機械原点

⚠ 注意
機械原点の損失
制御装置は、機械原点の損失の原因となる状況をすべて検出できるわけではありません。したがって、このような場合に原点マーク同期の原点確立を明示的に開始するのは、セットアップエンジニアまたはユーザーの一般的な責任です。

ギヤ切替の場合、レファレンス点または機械位置基準がエンコーダスケーリングに及ぼす影響を明示することはできません。このような場合、「軸原点確立/同期済み」状態が制御装置によって部分的に取り消されます。

機械原点が失われた場合は、まず、失われた機械原点の調整または原点確立をおこなって復元してください。

下記も参照

R1: レファレンス点復帰 (ページ 1497)

8.4.4 速度指令値出力

送り軸の制御方向と移動方向

作業を始める前に送り軸の移動方向を決めてください。

制御方向

ドライブの速度コントローラと電流コントローラをセットアップし調整してから、位置制御をセットアップしてください。

移動方向

次のマシンデータ:

MD32100 $MA_AX_MOTION_DIR (移動方向),

を使用して、位置制御の制御方向に影響を与えることなく

軸の移動方向を反転させることができます。

速度指令値の調整

SINUMERIK 840D sl

速度指令値を比較する場合、軸の制御と監視のパラメータ設定のために、ドライブでどの速度指令値がどのモータ速度に対応するかということが NC に通知されます。 この比較は自動的に実行されます。

PROFIBUS DP ドライブでは、速度指令値の手動比較も可能です。

- **手動比較**
  マシンデータ:
  MD32250 $MA_RATED_OUTVAL
  で、ゼロ以外の値を入力します。

注記

速度調整と最高速度指令値

SINUMERIK 840D sl では、自動速度指令値比較が可能であるため、速度調節は不要です。

最高速度指令値

SINUMERIK 840D sl では、最高速度指令値はパーセント値として定義されます。 100% とは、PROFIdrive ドライブの最高速度指令値または最高速度を意味します(ドライブの工作機械メーカ別設定パラメータ。例: SINAMICS の場合は p1082)。

SINUMERIK 840D sl では、主軸速度の出力は NC において実行されます。

5 つのギヤ選択用のデータがコントローラに存在します。

これらのギヤは、各ギヤの最低速度と最高速度および、自動ギヤ選択切換の最低速度と最高速度によって定義されます。 新しいギヤ選択のセットは、新たにプログラム指令速度が現在のギヤ選択で運転できない場合にのみ出力されます。

次のマシンデータ:

MD36210 $MA_CTRLOUT_LIMIT[n] (最高速度指令値)

により、速度指令値は、パーセント単位で制限できます。

最大 200%までの値を指定できます。

速度が、この制限を超えると、アラームが出力されます。

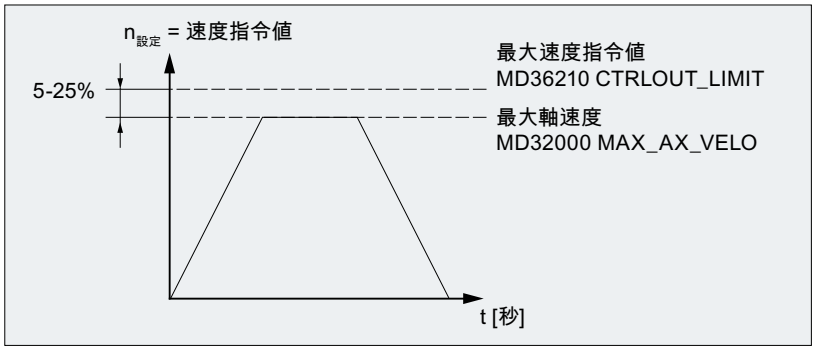


図 8-3　最高速度指令値

ただし、制御処理上の理由から、軸が最高速度(MD32000 $MA_MAX_AX_VELO)に達するのは、速度指令値の 100%ではなく、80%～ 95%であるべきです。

速度指令値範囲の約 80%で最高速度に達する軸の場合、次のマシンデータの初期値(80%)が使用可能です。

MD32000 $MA_MAX_AX_VELO (最高軸速度)

引継ぎが可能です。

8.4.5 フィードバック回路のマシンデータ

軸マシンデータ

フィードバック回路を設定するには、以下の軸マシンデータを設定してください。

エンコーダとパラメータセットとは無関係なマシンデータ: $MA_	意味
MD30200 NUM_ENCS	エンコーダの数
MD30300 IS_ROT_AX	回転軸/主軸
MD30310 ROT_IS_MODULO	回転軸/主軸のモジュロ変換
MD30320 DISPLAY_IS_MODULO	回転軸または主軸の 360 度モジュロ表示
MD30330 MODULO_RANGE	モジュロ範囲のサイズ
MD30340 MODULO_RANGE_START	モジュロ範囲の開始位置。
MD31030 $MA_LEADSCREW_PITCH	送りねじピッチ
MD31064 $MA_DRIVE_AX_RATIO2_DENOM	中間ギヤの分母
MD31066 $MA_DRIVE_AX_RATIO2_NUMERA	中間ギヤの分子
MD32000 $MA_MAX_AX_VELO	最大軸速度

エンコーダ別のマシンデータ: $MA_	意味
MD30210 ENC_SEGMENT_NR[n]	フィードバック値割り当て:バスセグメントの数
MD30220 ENC_MODULE_NR[n]	フィードバック値割り当て:ドライブ番号/検出回路番号
MD30230 ENC_INPUT_NR[n]	フィードバック値割り当て:ドライブモジュール/検出回路モジュールでの入力
MD30240 ENC_TYPE[n]	フィードバック値取得のエンコーダタイプ(位置フィードバック値)
MD30242 ENC_IS_INDEPENDENT[n]	エンコーダは独立している
MD30244 ENC_MEAS_TYPE[n]	エンコーダ計測タイプ
MD30250 ACT_POS_ABS[n]	内部エンコーダ位置

エンコーダ別のマシンデータ: $MA_	意味
MD30260 ABS_INC_RATIO[n]	絶対値エンコーダ:アブソリュート分解能とインクリメンタル分解能の間の比率
MD30270 ENC_ABS_BUFFERING[n]	絶対値エンコーダ:移動範囲の拡張
MD34090 $MA_REFP_MOVE_DIST_CORR[n]	基準点オフセット
MD34320 $MA_ENC_INVERS[n]	位置検出器が反対方向にある
N:エンコーダインデックス(n = 0, 1, ...)(1 番目のエンコーダ、2 番目のエンコーダ、...)	

注記

「マシンデータの機動」は、パートプログラムで命令 NEWCONF を使用するか、操作画面からソフトキーを押すかのいずれかで有効にすることができます。

8.4.6 フィードバック値の分解能

8.4.6.1 フィードバック値の分解能のマシンデータ

検出器(エンコーダ)の取り付け場所に応じて、以下の検出器タイプを考慮してください。

- 負荷側のエンコーダ:直接検出器(DM)
- モータ側のエンコーダ:間接検出器(IM)

フィードバック値の分解能の設定は、軸のタイプ(直線/回転軸)によって異なります。

コントロールシステムは、以下のマシンデータに基づいてフィードバック値分解能を計算します。

フィードバック値の分解能を計算するためのマシンデータ	直線軸	直線軸		回転軸	
	リニアスケール/直接検出器	間接検出器	直接検出器:機械/工具	間接検出器	直接検出器:機械/工具
MD30300 $MA_IS_ROT_AX	0	0	0	1	1
MD31000 $MA_ENC_IS_LINEAR[n]	1	0	0	0	0
MD31010 $MA_ENC_GRID_POINT_DIST[n]	間隔	-	-	-	-
MD34320 $MA_ENC_INVERS[n]	[1]	-	-	-	-
MD31020 $MA_ENC_RESOL[n]	-	パルス/rev	パルス/rev	パルス/rev	パルス/rev
MD31025 $MA_ENC_PULSE_MULT[n]	エンコーダの倍率				
MD31030 $MA_LEADSCREW_PITCH	-	mm/rev	mm/rev	-	-
MD31040 $MA_ENC_IS_DIRECT[n]	- / 1	0	1	0	1
MD31044 $MA_ENC_IS_DIRECT2[n]	- / 1	0	1	0	1
MD31050 $MA_DRIVE_AX_RATIO_DENOM[n]	-	負荷回転量	- / 1	負荷回転量	[2]
MD31060 $MA_DRIVE_AX_RATIO_NUMERA[n]	-	モータ回転量(送りギヤが存在する場合)	- / 1	モータ回転量	[2]
MD31070 $MA_DRIVE_ENC_RATIO_DENOM[n]	-	エンコーダ回転量	エンコーダ回転量	エンコーダ回転量	エンコーダ回転量
MD31080 $MA_DRIVE_ENC_RATIO_NUMERA[n]	-	モータ側のエンコーダ [3]	モータ回転量	モータ回転量	負荷回転量

- 関係なし

1) 絶対番地化検出器の場合
2) これらのマシンデータは、エンコーダの整合(軌跡評価)には必要ありませんが、指令値計算のために正確に入力しなければなりません。そうしないと、必要なサーボゲイン係数(K_V)が設定されません。負荷回転はマシンデータ MD31050 $MA_DRIVE_AX_RATIO_DENOM に入力し、モータ回転はマシンデータ MD31060 $MA_DRIVE_AX_RATIO_NUMERA に入力します。
3) モータ側のエンコーダは内蔵エンコーダであるため、検出ギヤユニットを備えていません。
   伝達比は常に 1:1 です。

フィードバック値の分解能のマシンデータ

フィードバック値の分解能は、機械の設計、ギヤボックスの使用可能性とそのギヤ比、直線軸の送りねじピッチ、および使用されるエンコーダの分解能から得られます。このために、コントロールシステムで以下のマシンデータを設定してください。

エンコーダとパラメータセットとは無関係なマシンデータ: $MA_	意味
MD30300 IS_ROT_AX	軸は回転軸/主軸です
MD31010 ENC_GRID_POINT_DIST	リニアスケールの原点マーク間の距離
MD31030 LEADSCREW_PITCH	送りねじピッチ
MD31064 DRIVE_AX_RATIO2_DENOM	付属ギヤボックスの分母
MD31066 DRIVE_AX_RATIO2_NUMERA	付属ギヤボックスの分子

エンコーダ別のマシンデータ: $MA_	意味
MD31000 ENC_IS_LINEAR[n]	検出器はリニアスケールです
MD31020 ENC_RESOL[n]	1 回転毎のエンコーダパルス数
MD31025 ENC_PULSE_MULT[n]	エンコーダの倍率(高分解能)
MD31040 ENC_IS_DIRECT[n]	直接または間接検出器
MD31044 ENC_IS_DIRECT2[n]	付属ギヤボックスに取り付けられたエンコーダ
MD31070 DRIVE_ENC_RATIO_DENOM[n]	検出器側ギヤボックス分母

エンコーダ別のマシンデータ: $MA_	意味
MD31080 DRIVE_ENC_RATIO_NUMERA[n]	検出器側ギヤボックス分子
N:エンコーダインデックス(n = 0, 1, ...)(1 番目のエンコーダ、2 番目のエンコーダ、など)	

パラメータセット関連のマシンデータ: $MA_	意味
MD31050 $MA_DRIVE_AX_RATIO_DENOM[m]	負荷側ギヤボックスの分母
MD31060 $MA_DRIVE_AX_RATIO_NUMERA[m]	負荷側ギヤボックスの分子
m:パラメータセットインデックス(m = 0, 1, ...)(1 番目のパラメータセット、2 番目のパラメータセット、など)	

計算分解能の設定

コントローラ内部の計算分解能とフィードバック値の分解能との比は、コントロールシステムによって算出された値を機械でどのくらい正確に実行できるかを示すものです。

計算分解能:直線軸

$$\frac{\text{計算分解能}}{\text{フィードバック値の分解能}} = \frac{\text{内部分割数/(mm)}}{\text{エンコーダパルス/(mm)}}$$

計算分解能:回転軸

$$\frac{\text{計算分解能}}{\text{フィードバック値の分解能}} = \frac{\text{内部分割数/(度)}}{\text{エンコーダパルス/(度)}}$$

マシンデータ

一般マシンデータ: $MN_	意味
MD10200 INT_INCR_PER_MM	直線位置の計算分解能
MD10210 INT_INCR_PER_DEG	回転位置の計算分解能

推奨設定

フィードバック値の分解能に関係する上記のコンポーネントと設定は、フィードバック値の分解能が設定された計算分解能より高くなるように選択してください。

$$\frac{\text{計算分解能}}{\text{フィードバック値の分解能}} \leq 1$$

8.4.6.2 例: リニアスケールを使用した直線軸

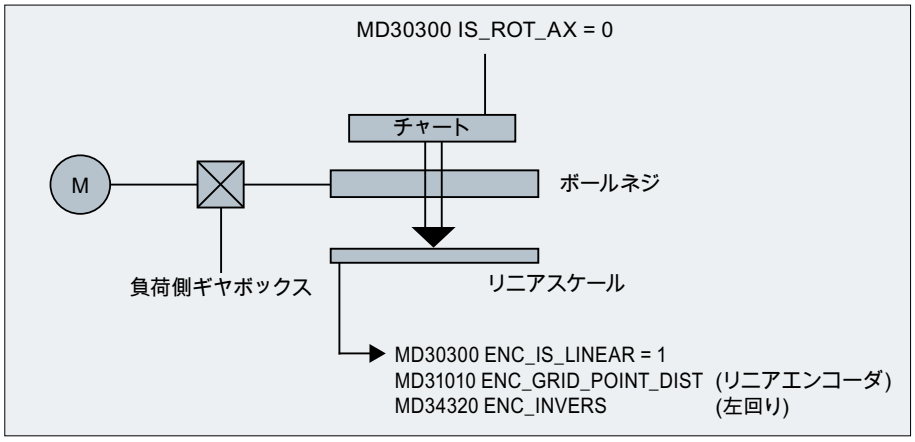


図 8-4 リニアスケールを使用した直線軸

1mm 毎の内部分割数とエンコーダパルスの比は、以下のようにして計算されます。

$$\frac{\text{内部インクレメント/mm}}{\text{エンコーダパルス/mm}} = \frac{\text{ENC_GRID_POINT_DIST [n]} * \text{INT_INCR_PER_MM}}{\text{ENC_PULSE_MULT[n]}}$$

8.4.6.3 例: モータ側にロータリエンコーダを使用した直線軸

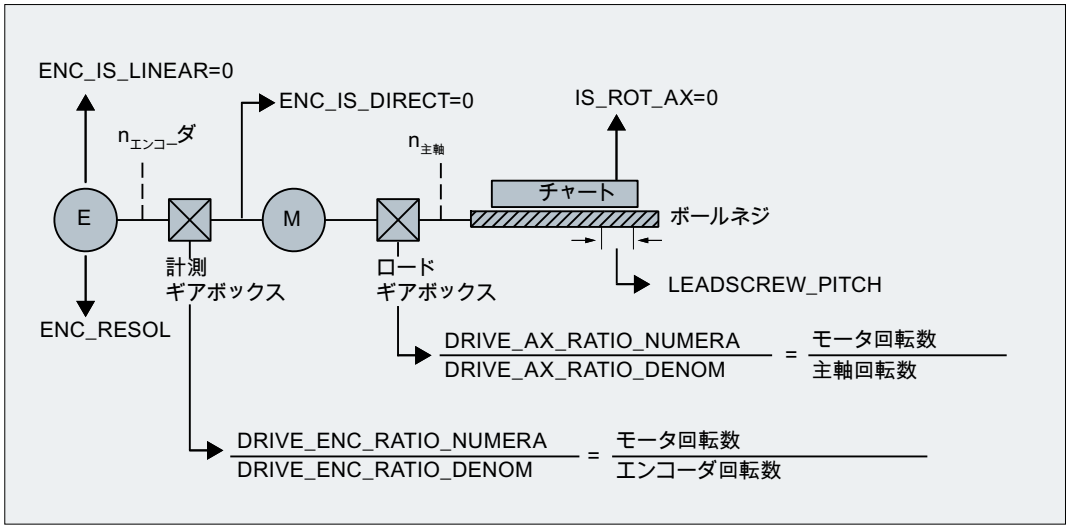


図 8-5　モータ側にロータリエンコーダを使用した直線軸

1mm 毎の内部分割数とエンコーダパルスの比は、以下のようにして計算されます。

$$\frac{\text{内部インクレメント/mm}}{\text{エンコーダパルス/mm}} = \frac{1}{\text{ENC_RESOL [n] * ENC_PULSE_MULT[n]}} * \frac{\text{DRIVE_ENC_RATIO_NUMERA [n]}}{\text{DRIVE_ENC_RATIO_DENOM [n]}} * \frac{\text{DRIVE_AX_RATIO_DENOM [n]}}{\text{DRIVE_AX_RATIO_NUMERA [n]}} * \text{LEADSCREW_PITCH} * \text{INT_INCR_PER_MM}$$

例

前提条件:

- モータ側のロータリエンコーダ: 2048 パルス/分解能
- 内部パルス倍率: 2048
- ギヤボックス、モータ/ボールねじ:5:1
- 送りねじピッチ: 10 mm/分解能
- 計算分解能: 1 mm 毎の分割数 10000

マシンデータ	値
MD30300 $MA_IS_ROT_AX	0
MD31000 $MA_ENC_IS_LINEAR[0]	0
MD31040 $MA_ENC_IS_DIRECT[0]	0
MD31020 $MA_ENC_RESOL[0]	2048
MD31025 $MA_ENC_PULSE_MULT	2048
MD31030 $MA_LEADSCREW_PITCH	10
MD31080 $MA_DRIVE_ENC_RATIO_NUMERA[0]	1
MD31070 $MA_DRIVE_ENC_RATIO_DENOM[0]	1
MD31060 $MA_DRIVE_AX_RATIO_NUMERA[0]	5

マシンデータ	値
MD31050 $MA_DRIVE_AX_RATIO_DENOM[0]	1
MD10210 $MN_INT_INCR_PER_DEG	10000

$$\rightarrow \frac{\text{内部インクレメント/mm}}{\text{エンコーダパルス/mm}} = \frac{1}{2048 * 2048} * \frac{1}{1} * \frac{1}{5} * 10\ \text{mm}$$

*10000インクレメント/mm = 0.004768

1 エンコーダパルスが 0.004768 内部分割数に対応します。つまり 209.731543 エンコーダパルスが 1 内部分割数に対応します。

8.4.6.4 例: 機械側にロータリエンコーダを使用した直線軸

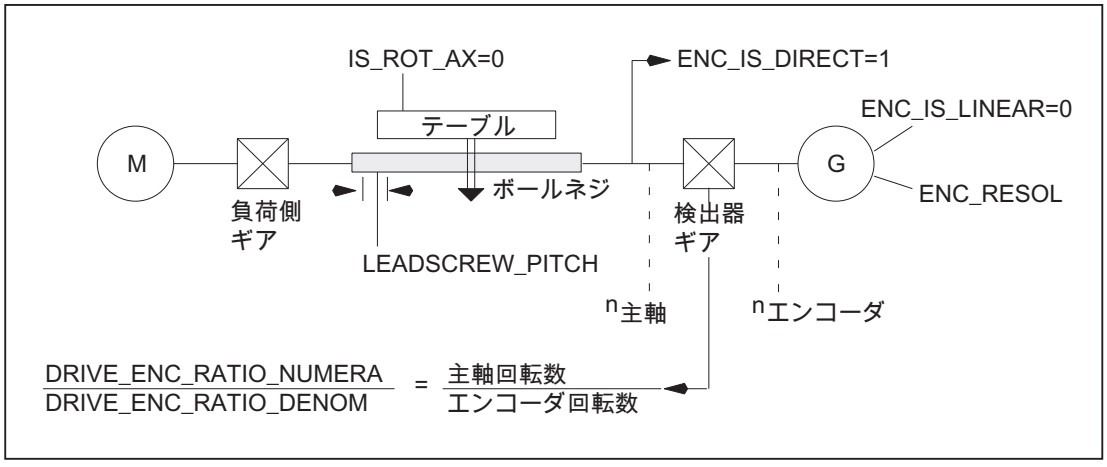


図 8-6 機械側にロータリエンコーダを使用した直線軸

1mm 毎の内部分割数とエンコーダパルスの比は、以下のようにして計算されます。

$$\frac{\text{内部インクレメント/mm}}{\text{エンコーダパルス/mm}} = \frac{1}{\text{ENC_RESOL [n]} * \text{ENC_PULSE_MULT[n]}} * \frac{\text{DRIVE_ENC_RATIO_NUMERA [n]}}{\text{DRIVE_ENC_RATIO_DENOM [n]}} * \text{LEADSCREW_PITCH} * \text{INT_INCR_PER_MM}$$

8.4.6.5 例: モータ側にロータリエンコーダを使用した回転軸

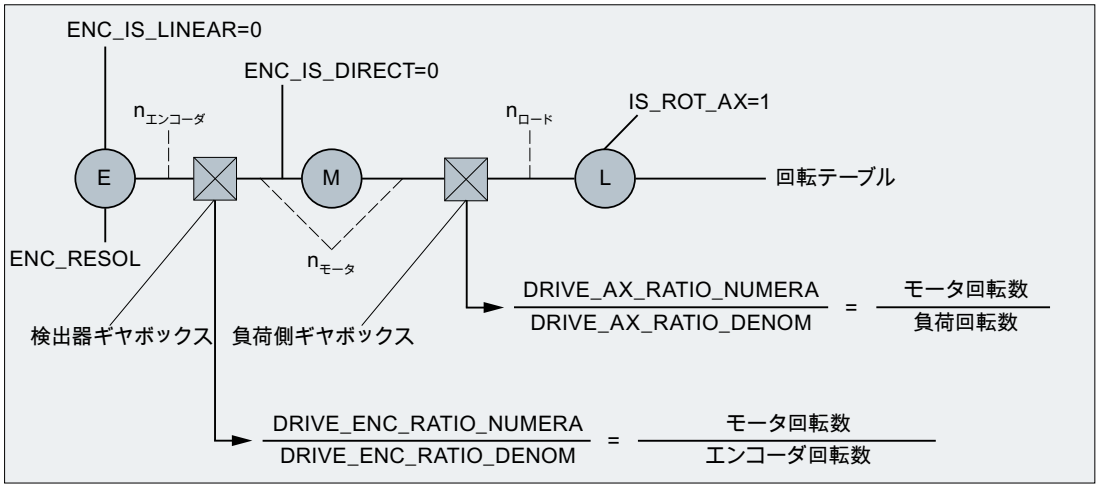


図 8-7　モータ側にロータリエンコーダを使用した回転軸

1°毎の内部分割数とエンコーダパルスの比は、以下のようにして計算されます。

$$\frac{\text{内部分割数/度}}{\text{エンコーダパルス/度}} = \frac{360°}{\text{ENC_RESOL [n] * ENC_PULSE_MULT[n]}} * \frac{\text{DRIVE_ENC_RATIO_NUMERA [n]}}{\text{DRIVE_ENC_RATIO_DENOM [n]}} * \frac{\text{DRIVE_AX_RATIO_DENOM [n]}}{\text{DRIVE_AX_RATIO_NUMERA [n]}} * \text{INT_INCR_PER_DEG}$$

例

前提条件:

- モータ側のロータリエンコーダ: 2048 パルス/分解能
- 内部パルス倍率: 2048
- ギアボックス、モータ/回転軸: 5:1
- 計算分解能: 1°毎の分割数 1000

マシンデータ	値
MD30300 $MA_IS_ROT_AX	1
MD31000 $MA_ENC_IS_LINEAR[0]	0
MD31040 $MA_ENC_IS_DIRECT[0]	0
MD31020 $MA_ENC_RESOL[0]	2048
MD31025 $MA_ENC_PULSE_MULT	2048
MD31080 $MA_DRIVE_ENC_RATIO_NUMERA[0]	1
MD31070 $MA_DRIVE_ENC_RATIO_DENOM[0]	1
MD31060 $MA_DRIVE_AX_RATIO_NUMERA[0]	5
MD31050 $MA_DRIVE_AX_RATIO_DENOM[0]	1
MD10210 $MN_INT_INCR_PER_DEG	1000

$$\Rightarrow \frac{\text{内部分割数/度}}{\text{エンコーダパルス/度}} = \frac{360\,^\circ}{2048 * 2048} * \frac{1}{1} * \frac{1}{5}$$

*1000 インクレメンタル/度 = 0,017166

1 エンコーダパルスが 0.017166 内部分割数に対応します。つまり 58.254689 エンコーダパルスが 1 内部分割数に対応します。

8.4.6.6 例: 機械側にロータリエンコーダを使用した回転軸

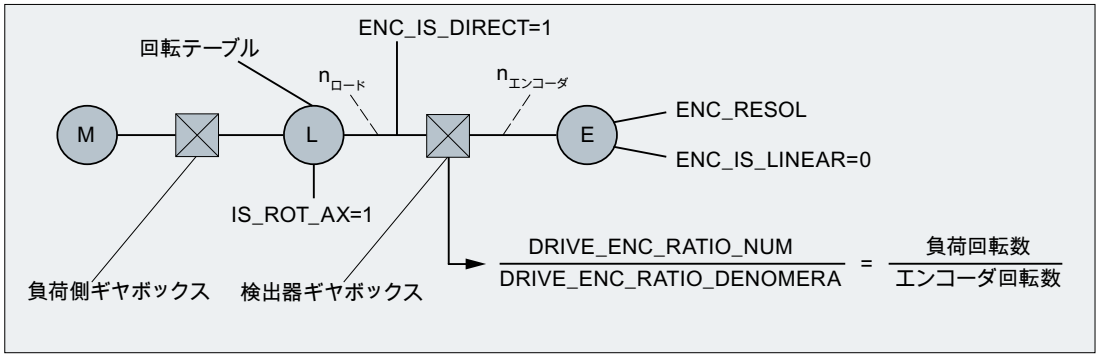


図 8-8 機械側にロータリエンコーダを使用した回転軸

1°毎の内部分割数とエンコーダパルスの比は、以下のようにして計算されます。

$$\frac{\text{内部分割数/度}}{\text{エンコーダパルス/度}} = \frac{360°}{\text{ENC_RESOL [n]} * \text{ENC_PULSE_MULT[n]}} * \frac{\text{DRIVE_ENC_RATIO_NUMERA [n]}}{\text{DRIVE_ENC_RATIO_DENOM [n]}} * \text{INT_INCR_PER_DEG}$$

8.4.6.7 例: エンコーダが工具に接続されている中間ギヤ

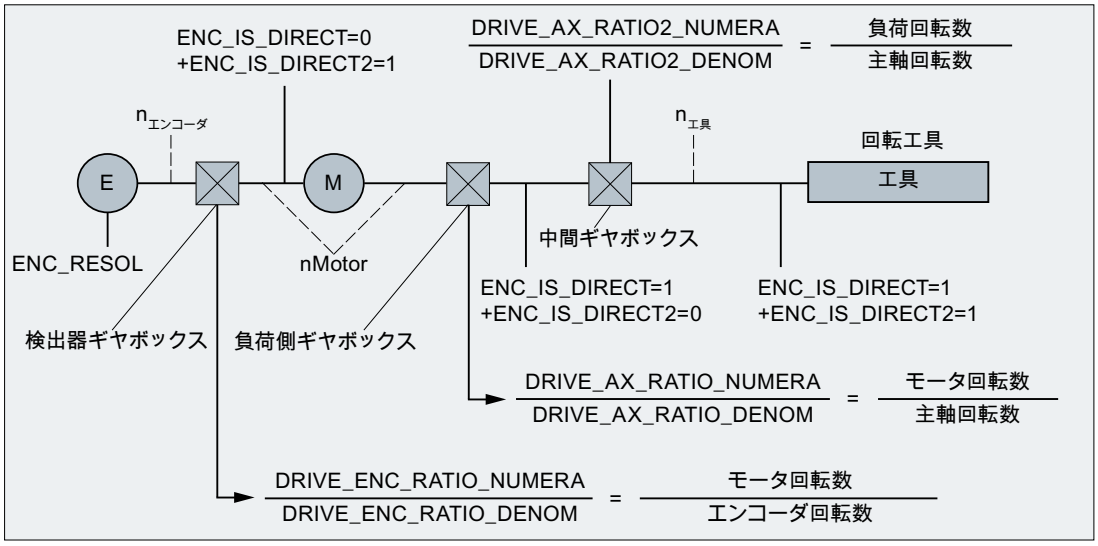


図 8-9 エンコーダが回転工具に直接接続されている中間ギヤ

1°毎の内部分割数とエンコーダパルスの比は、以下のようにして計算されます。

$$\frac{\text{内部分割数/度}}{\text{エンコーダパルス/度}} = \frac{360°}{\text{ENC_RESOL [n]} * \text{ENC_PULSE_MULT[n]}} * \frac{\text{DRIVE_ENC_RATIO_NUMERA [n]}}{\text{DRIVE_ENC_RATIO_DENOM [n]}} * \text{INT_INCR_PER_DEG}$$

8.5 閉ループ制御

8.5.1 一般情報

軸/主軸の位置制御

軸の閉ループ制御は、ドライブの電流と速度制御ループに加えて NC の上位レベルの位置制御ループで構成されます。

軸/主軸の位置制御の基本構造は、下図に示されています。

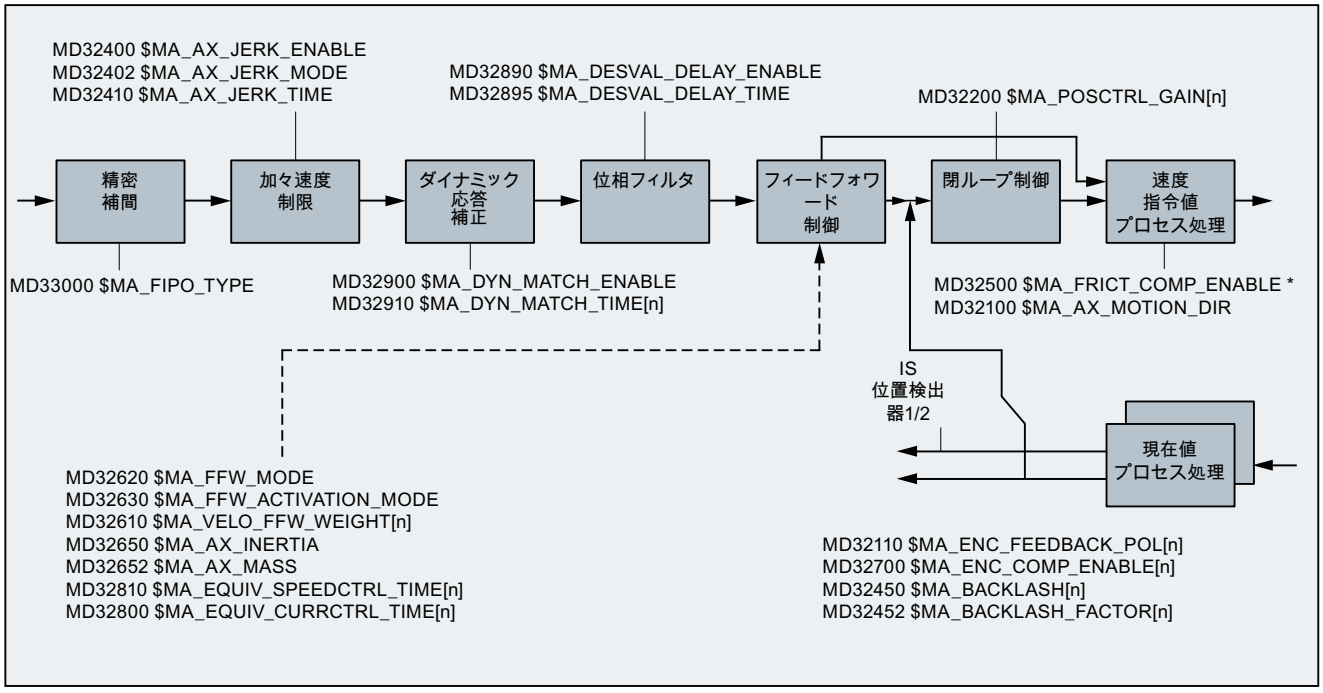


図 8-10　指令値処理と閉ループ制御の基本説明

加々速度制限については、「B2:加速度 (ページ 317)」の章を参照してください。

フィードフォワード制御、バックラッシ、摩擦補正およびピッチ誤差補正の説明については、以下を参照してください。

参照先

『機能マニュアル、上級機能』;「誤差補正(K3)」

ファインインタポレーション

ファインインタポレーションの補間器(FIPO)を使用して、速度指令値の階段効果を小さくして、輪郭精度をさらに上げることができます。3 つの異なるタイプのファインインタポレーションを設定できます。

MD33000 $MA_FIPO_TYPE = <FIPO モード>

<FIPO モード>	意味
1	IPO サイクル全体の平均値が生成される(スムージング)差動ファインインタポレーション
2	3 次ファインインタポレーション
3	最高の輪郭精度を得るためのプレコントロールでの使用に最適化された 3 次ファインインタポレーション

K_V 係数

連続軌跡モードで輪郭の誤差が生じないようにするために、サーボゲイン係数(K_v)を高くする必要があります。

MD32200 $MA_POSCTRL_GAIN[n]

ただし、サーボゲイン係数(K_v)が高すぎると、不安定性、オーバーシュートおよび場合によっては許容できないほどの機械への高い負荷が生じます。

最大許容可能なサーボゲイン係数(K_v)は、以下に依存します。

- ドライブの設計と応答性
  (立ち上がり時間、加速度および制動能力)
- 機械の品質
  (弾性、振動減衰)
- 有効な DSC の位置制御周期または速度制御周期

サーボゲイン係数(K_v)は、以下のように定義されます。

$$K_v = \frac{\text{速度}}{\text{追従誤差}} \; ; \; \frac{[\text{m/min}]}{[\text{mm}]}$$

ダイナミック応答調整

互いに補間し合うが、サーボゲイン係数(K_v)の異なる軸を、ダイナミック調整機能を使用して同一の追従誤差に設定することができます。これにより、動的に最も弱い軸に合わせてサーボゲイン係数(K_v)を小さくすることにより、制御性能を失うことなく、最適な輪郭確度を実現できます。

この機能は以下で起動します。

MD32900 $MA_DYN_MATCH_ENABLE = 1 (ダイナミック応答調整)

ダイナミック応答調整は、新規の等価時定数の入力によって実現されます。これは、動的に最も弱い軸と調整対象の軸の等価時定数の差分から計算されます。

MD32910 $MA_DYN_MATCH_TIME [n] = <等価時定数の差分>

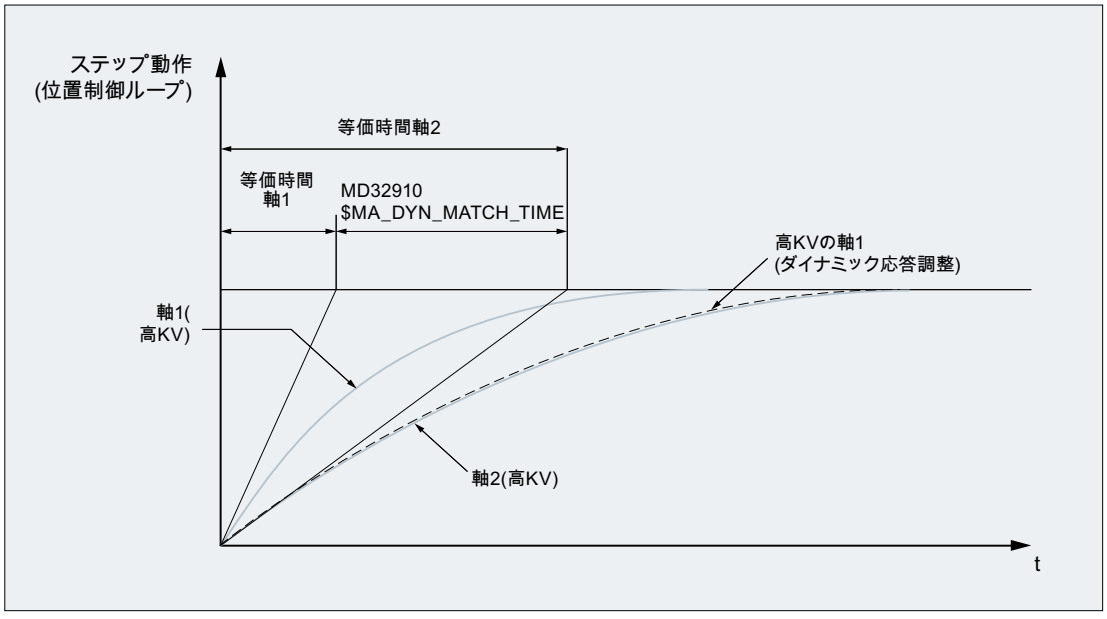


図 8-11　ダイナミック応答調整

速度のフィードフォワード制御なしの 3 つの軸のダイナミック応答調整の例

速度制御ループの等価時定数は次のようになります。	
軸 1:	30 ms
軸 2:	20 ms
軸 3:	24 ms

等価時定数 30 ms の場合、軸 1 が動的に最も弱い軸となります。

この結果、以下のような軸の新しい等価時定数が得られます。

軸 1:	MD32910 $MA_DYN_MATCH_TIME = 0 ms
軸 2:	MD32910 $MA_DYN_MATCH_TIME = 30 ms - 20 ms = 10 ms
軸 3:	MD32910 $MA_DYN_MATCH_TIME = 30 ms - 24 ms = 6 ms

軸の位置制御ループの等価時定数の近似値式は次のようになります。

軸の位置制御ループの等価時定数 T_{equiv} は、フィードフォワード制御のタイプに応じて近似値が計算されます。

- フィードフォワード制御なし

$$T_{Equiv} \approx \frac{1}{\text{MD32200 POSCTRL_GAIN [1/s]}}$$

- 速度フィードフォワード制御あり

$$T_{Equiv} \approx \text{MD32810 EQUIV_SPEEDCTRL_TIME}$$

- トルク/速度フィードフォワード制御の組み合わせ用

$$T_{Equiv} \approx \text{MD32800 EQUIV_CURRCTRL_TIME}$$

注記

ダイナミック応答調整をジオメトリ軸に対して実現する場合、他のすべてのジオメトリ軸を同一のダイナミック応答に設定してください。

参照先

『CNC 試運転マニュアル』:NC、PLC、ドライブ

8.5.2 位置コントローラのパラメータセット

1 機械軸毎に 6 つのパラメータセットを使用して、主軸のギヤ切替などの操作中に変更された機械特性に合わせて位置制御をすばやく補正したり、タッピング中などにダイナミック応答を別の軸に合わせて調整したりすることができます。

マシンデータ

パラメータセットは以下のマシンデータを比較します。

番号	識別子$MA_	意味
31050	DRIVE_AX_RATIO_DENOM	負荷側ギヤボックスの分母
31060	DRIVE_AX_RATIO_NUMERA	負荷側ギヤボックスの分子
32200	POSCTRL_GAIN	サーボゲイン係数(K_v)
32452	BACKLASH_FACTOR	バックラッシ補正
32610	VELO_FFW_WEIGHT	フィードフォワード制御係数
36012	STOP_LIMIT_FACTOR	汎用/精密イグザクトストップと停止状態
32800	EQUIV_CURRCTRL_TIME	等価時定数、フィードフォワード制御のための電流制御ループ
32810	EQUIV_SPEEDCTRL_TIME	等価時定数、フィードフォワード制御のための速度制御ループ
32910	DYN_MATCH_TIME	ダイナミック応答調整の時定数
36200	AX_VELO_LIMIT	速度監視の検出値

タッピング、ねじ切り

タッピングやねじ切りでは、軸のパラメータセットに関して以下が適用されます。

- タッピングやねじ切りに関係の**ない**機械軸の場合、パラメータセット 1 (インデックス = 0)が有効になります。他のパラメータセットは考慮する必要はありません。
- タッピングやねじ切りに関係のある機械軸の場合、主軸の現在のギヤ選択の場合と同じパラメータセット番号が有効になります。
  すべてのパラメータセットはギヤ選択に対応しているので、パラメータ設定する必要があります。

以下により、現在のパラメータセットがユーザーインタフェースに表示されます。

操作エリア[診断|サービス軸]

ギヤ選択切替時のパラメータセット

主軸の各ギヤ選択が個別のパラメータセットに割り当てられます。ギヤ選択は次の NC/PLC インタフェース信号を介して選択されます。

DB31、... DBX16.0 - 2 = <現在のギヤ選択>

現在のギヤ選択	DB31、... DBX16.0 - 2	パラメータセット	
1 番目のギヤ選択	000	2	(インデックス=1)
1 番目のギヤ選択	001	2	(インデックス=1)
2 番目のギヤ選択	010	3	(インデックス=2)
3 番目のギヤ選択	011	4	(インデックス=3)
4 番目のギヤ選択	100	5	(インデックス=4)
5 番目のギヤ選択	101 110 111	6	(インデックス=5)

主軸のギヤ選択について詳しくは、「S1:主軸 (ページ 1555)」の章を参照してください。

8.6 制御の調整

8.6.1 位置コントローラ、位置指令フィルタ:バランスフィルタ

機能

フィードフォワード制御が有効な場合、位置指令値はコントローラ自身に引き渡される前に、いわゆるバランスフィルタを通して送信されます。したがって、位置決め時にオーバーシュートすることなく、速度指令値を 100%に制御することができます。

起動

次の軸マシンデータを使用して、フィードフォワード制御タイプを選択し、フィルタも起動します。

MD32620 $MA_FFW_MODE (フィードフォワード制御モード)

値	意味
3	速度フィードフォワード
4	トルク/速度フィードフォワードの組み合わせ

パートプログラムによる有効化および無効化

パートプログラムで命令 FFWON および FFWOF を使用して、すべての軸に対するフィードフォワード制御を有効および無効にできます。

個々の軸のフィードフォワード制御が、FFWON/FFWOF の影響を受けるべきでない場合は、これらの軸について次のマシンデータの設定を変更してください。

MD32630 $MA_FFW_ACTIVATION_MODE (プログラムからのフィードフォワード制御の起動)

パラメータ設定

再セットアップ時の推奨設定

再セットアップの場合、または初期値がロードされた場合(セットアップスイッチのスイッチ位置 1 で POWER ON)、以下のマシンデータの初期値が適用されます。

- MD32620 $MA_FFW_MODE = 3
- MD32610 $MA_VELO_FFW_WEIGHT (速度フィードフォワード制御用のフィードフォワード制御係数) = 1

そして、速度フィードフォワード制御のバランス時間を、次のマシンデータで設定してください。

MD32810 $MA_EQUIV_SPEEDCTRL_TIME (フィードフォワード制御の速度制御ループ等価時定数)

速度制御ループの等価時定数の設定(MD32810)

パートプログラムを使用して軸を「AUTOMATIC」モードで移動できるようにすること、および目標位置への移動、すなわち、有効な検出器の位置フィードバック値をサーボトレースにより監視することを推奨します。

設定の初期値は、速度制御ループの時定数です。これは、速度制御ループの基準周波数特性から読み取ることができます。よく使用される、速度指令のスムージングをおこなう PI 制御 の場合は、ほぼ等しい時間をドライブマシンデータ p1414、p1415、p1416、p1421 から読み取ることができます。

この開始値(1.5 ms など)が、以下のように入力されます。

MD32810 $MA_EQUIV_SPEEDCTRL_TIME = 0.0015

軸が前後に移動し、オペレータは目標位置で位置フィードバック値の特性が大きく拡大されるのを監視します。

手動で微調整をおこなうには、以下の規則が適用されます。

モニタリング	計測	効果
オーバーシュート	MD32810 値の増加	位置コントローラは、よりゆっくりと位置コントローラは、ゆっくりと動作します。 ⇒ オーバーシュートの傾向は減少します。 問題:輪郭誤差(プログラムされた輪郭からの誤差)は曲線で増加します。
過度に緩やかなアプローチ	MD32810 値の削減	位置コントローラはより高速に動作します。 ⇒ 曲線の輪郭誤差は減少します。 問題:オーバーシュートの傾向は増加します。

注記

この効果は、位置コントローラゲイン(MD32200 $MA_POSCTRL_GAIN)を変更した場合と同様です。これは、ユーザーインターフェースの操作セクション[診断] > [軸サービス]で、計算した K_V 値を使用して確認できます。

MD32810 微調整

したがって、位置決め中のオーバーシュートが制限される設定になるように、MD32810にはできるだけ小さい値を割り当ててください。さらに、微調整中に必要となる初期値の変更はわずかで、通常 0.25 ms を加算または減算します。

たとえば、初期値が 1.5 ms の場合、手動で計算される最適値は通常 1.25 ms ～ 1.75 ms の範囲内になります。

直接検出器(負荷側エンコーダ)と高い柔軟性を備えた軸の場合、場合により数マイクロメータの小さいオーバーシュートを許容することもできます。これらは、ダイナミック応答調整(MD32910 $MA_DYN_MATCH_TIME)と加々速度(MD32410 $MA_AX_JERK_TIME)の位置指令値フィルタを使用して小さくすることができ、同時に軸の速度も低下させます。

補間グループ内の同一軸データ

補間グループ内のすべての軸では、以下のデータで**同一設定**をしてください。

- MD32200 $MA_POSCTRL_GAIN (MD32910 を使用して調整)
- MD32620 $MA_FFW_MODE
- MD32610 $MA_VELO_FFW_WEIGHT

- MD32810 $MA_EQUIV_SPEEDCTRL_TIME (または MD32800 $MA_EQUIV_CURRCTRL_TIME) (機械システムとドライブに応じて)
- MD32400 $MA_AX_JERK_ENABLE
- MD32402 $MA_AX_JERK_MODE
- MD32410 $MA_AX_JERK_TIME

確認するには、ユーザーインターフェースの操作セクション[診断] > [軸サービス]のサーボゲイン表示(K_V)を使用します。

補間グループ内の同一でない軸データ

上記データに同じ値を使用できない場合は、以下のマシンデータを使用して調整することができます。

MD32910 $MA_DYN_MATCH_TIME (ダイナミック応答調整の時定数)

これにより、同じサーボゲイン値(K_V)を表示できます。

異なるサーボゲイン表示値(K_V)は、通常は以下の場合を示します。

- ギヤ比は 1 つまたは複数の軸で一致しません。
- フィードフォワード制御セッティングデータが一致しません。

電流制御ループの等価時定数の設定(MD32800)

トルクフィードフォワード制御フィルタの設定は、次のマシンデータを使用して行います。

MD32620 $MA_FFW_MODE = 4

電流制御ループの時定数 MD32800 $MA_EQUIV_CURRCTRL_TIME の設定には、速度フィードフォワード制御の場合と同じ規則と推奨値が適用されます。

高剛性の機械への制限

経験では、この費用は非常に剛性の高い機械の場合のみ価値があり、適切な経験が必要です。機械の弾性は、トルクの入力によって、しばしば既存の振動が輪郭の精度のゲインを中和するほど強く励起されます。

この場合、代わりに「ダイナミックスティフネスコントロール」(DSC)機能を試してみてください。

MD32640 $MA_STIFFNESS_CONTROL_ENABLE = 1

注記

DSC の使用には以下の条件が適用されます。

- NC では、現在値の反転(MD32110 $MA_ENC_FEEDBACK_POL = -1)はパラメータ設定しないでください。ドライブの現在値の反転(SINAMICS パラメータ p0410)は使用できます。
- この機能は、補間関係(軌跡移動または連結)(MD32640 = 1)にあるすべての軸で起動する必要があります。
- いずれかの軸がシミュレーション軸(MD30130 $MA_CTRLOUT_TYPE = 0、MD30240 $MA_ENC_TYPE = 0)としてパラメータ設定されている場合、DSC は適用できません。この場合、補間関係(MD32640 = 0)にあるすべての軸で、この機能を解除する必要があります。

⚠ 注意

例外エラー条件

例外エラー条件は、p1192 が存在しない検出器に設定されており、DSC が有効な場合に発生する可能性があります!

追加検出器を無効にする必要があり、DSC が有効なサービス状況では、必ず p1192 も設定するようにします。

電源投入、リセット、REPOS などによるコントローラの動作

電源投入とリセット、さらに[マシンデータの有効化]を使用する場合も、フィードフォワード制御の設定データが再度読み込まれます(当該のマシンデータの適切な値を参照してください)。

モード変更、ブロック検索および REPOS は、フィードフォワード制御には影響が**ありません**。

8.6.2 位置コントローラ、位置指令フィルタ: 加々速度フィルタ

機能

型彫り平面のフライス加工のときなど、いくつかの用途では、加々速度フィルタを使用して位置指令曲線を滑らかにすることに利点がある場合があります。目的は、機械振動の励起を最小化することで加工面品質を最適にすることです。位置指令値のフィルタ効果は、輪郭の精度に許容できない影響を与えない範囲で、できる限り強くしてください。フィルタの 平滑動作もできる限り「対称」にしてください。すなわち、同じ輪郭を前進と後退の両方に移動した場合、フィルタにより丸み付けされる特性は両方の方向でできる限り同じになるようにしてください。

フィルタの効果は、実効サーボゲイン係数(K_V)によって監視することができ、これは[軸サービス]画面に表示されます。フィルタリングの効果により位置指令値がわずかに丸み付けされて、軌跡の精度が低下するので、フィルタ時間を増加させると、より小さいサーボゲイン係数(K_V)が表示されます。

注記

加々速度フィルタは、振幅設定ごとに従属的な位相差を生成します。位相フィルタ(「位置コントローラ、位置指令フィルタ: 位相フィルタ (ページ 464)」を参照)を付加的に使用することによってのみ、軸ダイナミック応答の使い易い設定が許可されます。

起動

軸別の加々速度フィルタ指令値の機能は、以下のマシンデータで起動してください。

MD32400 $MA_AX_JERK_ENABLE[<軸>] = "TRUE"

パラメータ設定

軸別の加々速度制限のフィルタタイプ

「フローティング平均」フィルタタイプは、通常、軸別の変化率制限に使用します。

MD32402 $MA_AX_JERK_MODE[<軸>] = **2**

注記

フィルタタイプ MD32402 $MA_AX_JERK_MODE = 2 が事前に有効になっていない場合は、一度「電源投入」を行ってください。それ以外の場合は、機械操作パネルの[マシンデータの有効化]または[リセット]をおこないます。

軸別の加々速度フィルタの時定数

軸別の加々速度フィルタ[s]の時定数は、以下のマシンデータで設定します。

MD32410 $MA_AX_JERK_TIME[<軸>]

微調整

加々速度フィルタの微調整は、以下のように実行します。

1. 軸の移動応答を解析します(サーボトレースによる位置決め処理などによって)。
2. MD32410 $MA_AX_JERK_TIME でフィルタ時間を修正します。
3. 修正された時間を機械操作パネルの[マシンデータの有効化]またはリセットを介して有効にします。

解除

加々速度フィルタの解除

1. 次のように、フィルタ計算を無視します。
   MD32400 $MA_AX_JERK_ENABLE = 0
2. インタロックを機械操作パネルの[マシンデータの有効化]またはリセットを介して有効にします。

境界条件

フィルタ時間

加々速度フィルタは、時定数(MD32410)が 1 位置制御周期よりも大きい場合にのみ有効になります。

フィルタ効果

- 計算されたサーボゲイン係数(K_V)の[軸サービス]画面での表示は、フィルタ効果に基づいた適切な値より小さい値が表示されます。
- 軌跡精度は、表示されるサーボゲイン(K_V)の期待より良くなります。
  したがって、MD32400 = 1 を MD32400 = 2 にリセットすると、表示されるサーボゲイン(K_V)は同じフィルタ時間を保持したままで小さくすることができ、それにもかかわらず軌跡精度は改善されます。

相互に補間する軸

- 同一に設定してください。
- これらの軸に対して一度最適値が決定されたら、最も長いフィルタ時間を補間グループ内のすべての軸に対する設定として使用してください

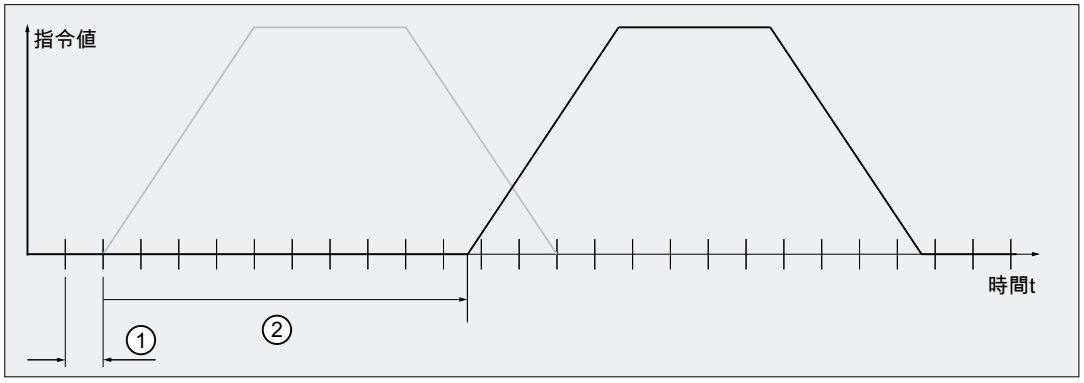
参照先

補間器レベルでの加々速度制限について詳しくは、「単独軸補間の加々速度制限(SOFTA) (軸別) (ページ 339)」と「軸/主軸マシンデータ (ページ 372)」の章を参照してください。

8.6.3 位置コントローラ、位置指令フィルタ: 位相フィルタ

機能

軸別の位相フィルタ指令値(むだ時間/遅延)で、指令値位相応答に影響を与えることのできる純粋な移相器が実装されます。軸別の加々速度フィルタ指令値(MD32402_ $MA_AX_JERK_MODE[<軸>] = 2、「位置コントローラ、位置指令フィルタ: 加々速度フィルタ (ページ 462)」の章を参照)と併用すると、振幅応答と位相応答を互いに独立して、数本の軸のうちで動的に最も弱いものに補正することができます。この軸が、プログラム指令された軌跡に沿って進行します。



① 位置制御周期クロック:2 ms

② 軸別の位相フィルタ指令値の時定数:19.2 ms

図 8-12 軸別の位相フィルタ指令値の効果:

パラメータ設定:時定数

軸別の位相フィルタ指令値の時定数は、以下の範囲内で設定できます。

(0 ～ 64) * 位置制御周期クロック(MD10061 $MN_POSCTRL_CYCLE_TIME)

時定数の値を、以下のマシンデータに秒[s]単位で入力してください。

MD32895 $MA_DESVAL_DELAY_TIME[<軸>]

例:位置制御周期クロック:2 ms ⇒ 設定可能な時定数:0.0 ～ 0.128 秒

注記

位相フィルタ指令値の時定数により、タッピング、後退、およびイグザクトストップ/ブロック変更などの軸の応答特性が遅延します。したがって、時定数は、できる限り低く設定することを推奨します。

最大値への制限

許容最大値(64 * 位置制御周期クロック)を超える時定数は、最大値までに内部的に制限されています。メッセージ/アラームは表示されません。

パラメータ設定:起動

軸別の位相フィルタ指令値の機能は、以下のマシンデータで起動してください。

MD32890 $MA_DESVAL_DELAY_ENABLE[<軸>] = "TRUE"

例

前提条件:位置制御周期クロック= 2 ms

1. MD32890 $MA_DESVAL_DELAY_ENABLE[<軸>] = "FALSE"
   MD32895 $MA_DESVAL_DELAY_TIME[<軸>] = <時定数>
   - 位相フィルタ指令値:無効
   - 時定数:無効
2. MD32890 $MA_DESVAL_DELAY_ENABLE[<軸>] = TRUE
   MD32895 $MA_DESVAL_DELAY_TIME[<軸>] = 0.002
   - 位相フィルタ指令値:有効
   - 時定数:2 ms ⇒ 指令値出力が位置制御周期クロックにより遅延します。
3. MD32890 $MA_DESVAL_DELAY_ENABLE[<軸>] = TRUE
   MD32895 $MA_DESVAL_DELAY_TIME[<軸>] = 0.256
   - 位相フィルタ指令値:有効
   - 設定した時定数:256 ms;
     最大許容時定数:64 * 2 ms = 128 ms
     ⇒ 内部的に有効な時定数:128 ms

必要条件

SINUMERIK Safety Integrated

位相フィルタ指令値により、軸別指令値の出力が遅延します(後退(停止 E の ESR)など)。ただし、位相フィルタ指令値は、停止プロセスには影響を与えません(SBH 起動時間など)。

8.6.4 位置コントローラ: 位置偏差の入力

必要条件

- この機能は、2 つのエンコーダのある軸についてのみ使用できます。
  MD30200 $MA_NUM_ENCS = 2
  一方のエンコーダは間接検出器としてパラメータ設定し、もう一方は直接検出器としてパラメータ設定してください。
  - 直接検出器:
    MD31040 $MA_ENC_IS_DIRECT[1]=1
    位置フィードバック値取得のためのエンコーダは、機械に直接接続されます(負荷エンコーダ)。
  - 間接検出器:
    MD31040 $MA_ENC_IS_DIRECT[0]=0
    位置フィードバック値取得のためのエンコーダは、モータ上に配置されています(モータエンコーダ)。
- 通信形式 136 または 138 は、ドライブと NC の両方で PROFIdrive 用の標準通信形式として設定してください(MD13060 $MN_DRIVE_TELEGRAM_TYPE)。

機能

有効な入力済み位置偏差の場合、軸の直接検出器と間接検出器の間で位置の差が特定され、加重係数設定に従って、位置制御周期でフィードフォワード制御用の追加電流指令値として適用されます。 得られる振動減衰により、軸の安定性および位置決め動作が改善されます。

用途

この機能は、振動する傾向が強い軸に対して使用されます。

有効性

この機能は、低い固有周波数(約 20 Hz まで)を持つ軸についてのみ機能します。

起動/パラメータ設定

この機能は、次の加重係数を指定することによって有効になります。

MD32950 $MA_POSCTRL_DAMPING (速度制御ループの減衰) = <値>

数値の範囲: -100% ... +100%

入力値「100%」の意味: 2 つの検出器の間で特定された位置の差が次の値に到達したときに、SINAMICS パラメータ p2003 に従った補助トルクが適用されます。

- リニアモータ:1 mm
- 回転モータを備えた直線軸:MD31030 $MA_LEADSCREW_PITCH (送りねじピッチ)
- 回転軸/主軸の場合: 360°

標準設定は 0 です。この場合、位置偏差の入力は無効になります。

注記

加重係数 MD32950 $MA_POSCTRL_DAMPING は、たとえば、ステップ応答に基づいて設定できます。

この制御が安定限界に近づいた場合(振動傾向が増大)は、パラメータが大きすぎるか、値の符号が正しくありません。

8.6.5 比例積分動作コントローラによる位置制御

機能

標準では、位置コントローラの核となるのはP コントローラです。特定のアプリケーション(電子ギヤなど)の場合に、積分要素に切り替えることができます。その結果、比例積分動作コントローラは、当該のマシンデータが適切に設定されたとき、有限の設定可能な時間の中で、指令値とフィードバック位置の間で誤差をゼロになるまで修正します。

⚠ 注意

PI 制御が有効な場合のフィードバック位置のオーバーシュート

この場合、この影響を許容できるかどうかを当該のアプリケーションに対して決定してください。制御技術の知識とサーボトレースを使用した測定結果は、この機能を使用するための絶対必要条件です。当該のマシンデータが正しく設定されていない場合、不安定性のために機械が破損するおそれがあります。

手順

1. まず、前のサブセクションで説明したツールを使用して、位置制御ループを比例動作コントローラとして最適化します。
2. 比例積分動作コントローラによる位置制御の品質の決定するために計測を行いながら、以下のマシンデータの許容範囲を増大させます。
   - MD36020 $MA_POSITIONING_TIME
   - MD36030 $MA_STANDSTILL_POS_TOL
   - MD36040 $MA_STANDSTILL_DELAY_TIME
   - MD36400 $MA_CONTOUR_TOL
3. 以下のマシンデータを設定することにより、位置制御ループを比例積分動作コントローラとして有効にします。
   MD32220 $MA_POSCTRL_INTEGR_ENABLE ; 設定値 1
   MD32210 $MA_POSCTRL_INTEGR_TIME ; 積分時間[秒]
   積分時間の効果:
   - $T_n \rightarrow 0$:
     制御偏差はすばやく減らされますが、制御ループは不安定になる可能性があります。
   - $T_n \rightarrow \infty$:
     積分要素の有効性は、ほぼ 0 です。コントローラの動作は、純粋な比例コントローラと同様です。

4. これらの両極端な例の間で、対象のアプリケーションに合った T_n の妥協値を探します。

   注記

   T_n の選択は、安定限界に近すぎないようにしてください。不安定な状態が発生すると、機械が破損するおそれがあるためです。

   したがって、T_n を 1 秒以上に設定することを推奨します。

   目標位置までおよび目標位置から移動する自動プログラムの移動の記録には、サーボトレースを使用します。

5. サーボトレースは以下ものを表示するように設定します。
   - 追従誤差
   - フィードバック速度
   - フィードバック位置
   - 指令位置
6. T_n の最適値が特定されたら、以下のマシンデータの許容値を必要な値に再設定します。
   - MD36020 $MA_POSITIONING_TIME
   - MD36030 $MA_STANDSTILL_POS_TOL
   - MD36040 $MA_STANDSTILL_DELAY_TIME
   - MD36400 $MA_CONTOUR_TOL

必要条件

DSC

積分器機能を使用する場合は、DSC (ダイナミックスティフネスコントロール)をオフにしてください。

例

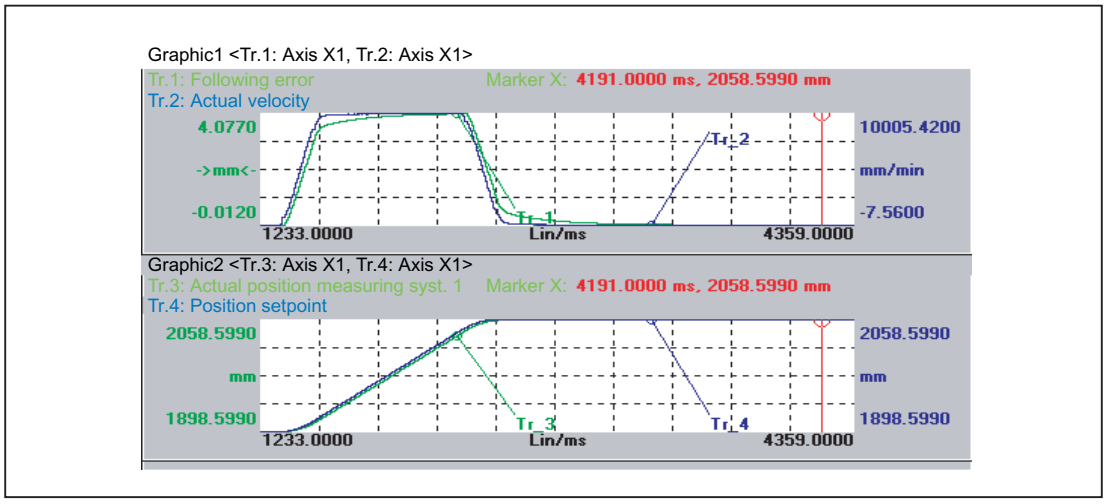
K_R および T_n の複数の反復プロセスの後の結果の設定

マシンデータの設定

- MD32220 $MA_POSCTRL_INTEGR_ENABLE = 1
- MD32210 $MA_POSCTRL_INTEGR_TIME = 0.003
- MD32200 $MA_POSCTRL_GAIN[1] = 5.0

パラメータセット選択 0

以下の量のそれぞれ - 追従誤差、フィードバック速度、フィードバック位置および位置指令値 - がサーボトレースにより記録されました。JOG モードで移動すると、以下の図に示された個々のデータの特性が描かれます。



8.7 データリスト

8.7.1 マシンデータ

8.7.1.1 表示マシンデータ

番号	識別子: $MM_	説明
9004	DISPLAY_RESOLUTION	最小表示単位
9010	SPIND_DISPLAY_RESOLUTION	主軸の最小表示単位
9011	DISPLAY_RESOLUTION_INCH	インチ単位系の最小表示単位

8.7.1.2 NC 別マシンデータ

番号	識別子: $MN_	説明
10000	AXCONF_MACHAX_NAME_TAB	機械軸名称
10050	SYSCLOCK_CYCLE_TIME	システム基本サイクル
10070	IPO_SYSCLOCK_TIME_RATIO	補間周期の係数
10060	POSCTRL_SYSCLOCK_TIME_RATIO	位置制御周期の係数
10200	INT_INCR_PER_MM	直線位置の計算分解能

番号	識別子: $MN_	説明
10210	INT_INCR_PER_DEG	回転位置の計算分解能
10220	SCALING_USER_DEF_MASK	単位変換係数の起動
10230	SCALING_FACTORS_USER_DEF	物理量の単位変換係数
10240	SCALING_SYSTEM_IS_METRIC	メトリック基本単位系
10250	SCALING_VALUE_INCH	インチ単位系に切り替えるための 変換係数
10260	CONVERT_SCALING_SYSTEM	基本単位系切り替え有効
10270	POS_TAB_SCALING_SYSTEM	位置テーブルの単位系
10290	CC_TDA_PARAM_UNIT	CC 用工具データの物理単位
10292	CC_TOA_PARAM_UNIT	CC 用工具刃先データの物理単位
13050	DRIVE_LOGIC_ADDRESS	論理ドライブアドレス
13060	DRIVE_TELEGRAM_TYPE	PROFIBUS DP の標準メッセージフレームタイプ
13070	DRIVE_FUNCTION_MASK	使用される DP 機能
13080	DRIVE_TYPE_DP	ドライブタイプ PROFIBUS DP

8.7.1.3 チャネルマシンデータ

番号	識別子: $MC_	説明
20150	GCODE_RESET_VALUES	G グループの初期設定

8.7.1.4 軸/主軸マシンデータ

番号	識別子: $MA_	説明
30110	CTRLOUT_MODULE_NR	指令値割り当て:ドライブ番号
30120	CTRLOUT_NR	指令値割り当て:ドライブモジュールでの指令値出力
30130	CTRLOUT_TYPE	指令値の出力タイプ
30200	NUM_ENCS	エンコーダの数
30220	ENC_MODULE_NR	フィードバック値割り当て:ドライブモジュール番号

番号	識別子: $MA_	説明
30230	ENC_INPUT_NR	フィードバック値割り当て:ドライブモジュールでの入力
30240	ENC_TYPE	フィードバック値取得のタイプ(位置フィードバック値)
30242	ENC_IS_INDEPENDENT	エンコーダは独立している
30300	IS_ROT_AX	回転軸
31000	ENC_IS_LINEAR	直接検出器(リニアスケール)
31010	ENC_GRID_POINT_DIST	リニアスケール上の原点マーク間の距離
31020	ENC_RESOL	1 回転当たりのエンコーダパルス数
31030	LEADSCREW_PITCH	送りねじピッチ
31040	ENC_IS_DIRECT	エンコーダが機械に直接接続されている
31044	ENC_IS_DIRECT2	中間ギヤのエンコーダ
31050	DRIVE_AX_RATIO_DENOM	負荷ギヤ分母
31060	DRIVE_AX_RATIO_NUMERA	負荷ギヤ分子
31064	DRIVE_AX_RATIO2_DENOM	中間ギヤの分母
31066	DRIVE_AX_RATIO2_NUMERA	中間ギヤの分子
31070	DRIVE_ENC_RATIO_DENOM	検出器ギヤ分母
31080	DRIVE_ENC_RATIO_NUMERA	検出器ギヤ分子
31090	JOG_INCR_WEIGHT	INC/ハンドルのパルスの重み
31200	SCALING_FACTOR_G70_G71	G70/G71 が有効な場合に値を変換するための係数
32000	MAX_AX_VELO	最大軸速度
32100	AX_MOTION_DIR	移動方向
32110	ENC_FEEDBACK_POL	現在値の符号(フィードバックの極性)
32200	POSCTRL_GAIN	サーボゲイン係数(K_v)
32210	POSCTRL_INTEGR_TIME	積分時間位置コントローラ
32220	POSCTRL_INTEGR_ENABLE	位置コントローラの積分要素の起動
32250	RATED_OUTVAL	定格出力電圧
32260	RATED_VELO	定格モータ速度
32450	バックラッシ	バックラッシ
32500	FRICT_COMP_ENABLE	摩擦補正有効

番号	識別子: $MA_	説明
32610	VELO_FFW_WEIGHT	速度フィードフォワード制御のフィードフォワード制御係数
32620	FFW_MODE	フィードフォワード制御モード
32630	FFW_ACTIVATION_MODE	プログラムからのフィードフォワード制御の起動
32650	AX_INERTIA	トルクフィードフォワード制御の慣性モーメント
32652	AX_MASS	トルクフィードフォワード補償の軸質量
32711	CEC_SCALING_SYSTEM_METRIC	真直度補正の単位系
32800	EQUIV_CURRCTRL_TIME	フィードフォワード制御のための電流制御ループ等価時定数
32810	EQUIV_SPEEDCTRL_TIME	フィードフォワード制御のための速度制御ループ等価時定数
32890	DESVAL_DELAY_ENABLE	軸別の位相フィルタ指令値
32895	DESVAL_DELAY_TIME	軸別の位相フィルタ指令値の時定数
32900	DYN_MATCH_ENABLE	ダイナミック応答調整
32910	DYN_MATCH_TIME [n]	ダイナミック応答調整の時定数
32930	POSCTRL_OUT_FILTER_ENABLE	位置コントローラ出力でのローパスフィルタの起動
32950	POSCTRL_DAMPING	速度制御ループの減衰
33000	FIPO_TYPE	ファインインタポレーションのタイプ
34320	ENC_INVERS[n]	位置検出器が逆方向
35100	SPIND_VELO_LIMIT	最高主軸速度
36200	AX_VELO_LIMIT [n]	速度監視の検出値
36210	CTRLOUT_LIMIT[n]	最高速度指令値
36400	AX_JERK_ENABLE	軸別の変化率制限
36410	AX_JERK_TIME	軸別の加々速度フィルタの時定数
36500	ENC_CHANGE_TOL	位置フィードバック値切り替えの最大許容範囲
36510	ENC_DIFF_TOL	検出器間許容誤差
36700	ENC_COMP_ENABLE[n]	補間補正

8.7.2 信号

8.7.2.1 NC からの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
輪郭の変更、インチ/メトリック単位系	DB10.DBB71	DB2700.DBB0015
インチ単位系	DB10.DBX107.7	DB2700.DBX1.7

8.7.2.2 NC への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
位置検出器 1	DB31、... .DBX1.5	DB38xx.DBX1.5
位置検出器 2	DB31、... .DBX1.6	DB38xx.DBX1.6

H2: PLC への補助機能出力

9

9.1 概略説明

9.1.1 機能

補助機能は、NC のシステムファンクションおよび PLC ユーザーファンクションの起動を可能にします。補助機能は、以下でプログラム指令できます。

- パートプログラムパートプログラム
- シンクロナイズドアクション
- ユーザーサイクル

シンクロナイズドアクションでの補助機能出力の使用についての詳細は、以下を参照してください。

参照先:
『機能マニュアル、シンクロナイズドアクション』

事前定義補助機能

事前定義補助機能は、システムファンクションを起動します。また、補助機能は、NC/PLC インタフェースに出力されます。

次の補助機能が予約されています。

タイプ	機能	例	意味
M	追加機能	M30	エンドオブプログラム
S	主軸機能	S100	主軸速度= 100 (rpm など)
T	工具番号	T3	工具番号 3
D、DL	工具オフセット	D1	工具刃先番号 1
F	送り速度	F1000	送り速度 1000 (mm/min など)

ユーザー定義補助機能

ユーザー定義の補助機能は、拡張された事前定義補助機能またはユーザー専用の補助機能です。

事前定義補助機能の拡張

事前定義補助機能の拡張は、「アドレス拡張」パラメータを参照します。アドレス拡張により、補助機能が適用される主軸の番号が定義されます。主軸機能 M3 (主軸右回転)は、チャネルのメイン主軸に対して予約されています。第 2 主軸がチャネルに割り当てられる場合、事前定義補助機能を拡張する、関連の対応ユーザー定義補助機能を定義してください。

タイプ	機能	例	意味
M	追加機能	M2 = 3	第 2 主軸:主軸右回転
S	主軸機能	S2=100	第 2 主軸:主軸速度= 100 (rpm など)
T	工具番号	T2=3	

ユーザー専用の補助機能

ユーザー専用の補助機能は、システムファンクションを起動しません。ユーザー専用の補助機能は、NC/PLC インタフェースにのみ出力されます。補助機能の機能は、PLC ユーザープログラムで工作機械メーカ/ユーザーが組み込んでください。

タイプ	機能	例	意味
H[1)]	補助機能	H2=5	ユーザー専用の機能

1) 推奨値

9.1.2 補助機能の定義

補助機能は、次のパラメータにより定義されます。

- **タイプ、アドレス拡張、値**
  この 3 つのパラメータは、NC/PLC インタフェースに出力されます。
- **出力動作**
  補助機能別の出力動作には、補助機能が NC/PLC インタフェースに出力される時間、および同じパートプログラムブロックにプログラム指令された移動動作に対する補助機能出力タイミングを定義します。
- **グループの割り当て**
  補助機能は、特定の補助機能グループに割り付けることができます。 出力動作は、補助機能グループ毎に個別に定義できます。 この機能が動作するのは、補助機能別の出力動作が定義されていない場合です。 グループのメンバ構成は、ブロック検索後の補助機能の出力にも影響を及ぼします。

NC/PLC インタフェースに出力される補助機能の詳細については、「P3: SINUMERIK 840D sl 用 PLC 基本プログラム (ページ 1043)」の章を参照してください。

9.1.3 補助機能の概要

M 機能

M (補助機能)					
アドレス拡張		値			
値の範囲	意味	数値の範囲 [10]	タイプ	意味	記述数 [8]
0 (省略時)	---	0 ... 99	INT	機能	5
値の範囲	意味	数値の範囲 [11]	タイプ	意味	記述数 [8]
1 ... 20	主軸番号	1 ... 99	INT	機能	5
値の範囲	意味	数値の範囲 [12]	タイプ	意味	記述数 [8]
0 ... 99	指定なし	100 ... 2147483647	INT	機能	5

[8] 概要の最後にある「脚注の意味」を参照してください。

[10] 数値の範囲 0 ... 99 では、アドレス拡張は 0 です。アドレス拡張子なしの場合、必須:M0、M1、M2、M17、M30

[11] M3、M4、M5、M19、M70:アドレス拡張は、主軸番号です(たとえば、主軸 2 では M2=5 ⇒ 主軸停止(M5))。
アドレス拡張がない場合、M 機能はメイン主軸で機能します。

[12] ユーザー専用の M 機能。

使用/用途

パートプログラムと同期した機械動作の制御

他の情報

- 次の M 機能には、予約された意味があります。M0、M1、M2、M17、M30、M3、M4、M5、M6、M19、M70、M40、M41、M42、M43、M44、M45。
- 有効性を確認するダイナミック NC/PLC インタフェース信号は、M 機能(M0 - M99)に割り当てられます。また、64 個の追加信号をユーザー M 機能に割り当てることができます(「P3: SINUMERIK 840D sl 用 PLC 基本プログラム (ページ 1043)」の章を参照してください)。
- サブプログラムでは、次のマシンデータを使用すれば、M17、M2、または M30 のパートプログラムの終わりに対して PLC に M 機能を出力するかどうか設定できます。
MD20800 $MC_SPF_END_TO_VDI (PLC へのサブプログラム終了)

- 事前定義 M 機能 M40 ～ M45 では、出力指定の制限付き再定義だけが可能です。
- 事前定義補助機能 M0、M1、M17、M30、M6、M4、M5 は再定義できません。
- M 機能用マシンデータには、次のものがあります。
  - MD10800 $MN_EXTERN_CHAN_SYNC_M_NO_MIN
  - MD10802 $MN_EXTERN_CHAN_SYNC_M_NO_MAX
  - MD10804 $MN_EXTERN_M_NO_SET_INT
  - MD10806 $MN_EXTERN_M_NO_DISABLE_INT
  - MD10814 $MN_EXTERN_M_NO_MAC_CYCLE
  - MD10815 $MN_EXTERN_M_NO_MAC_CYCLE_NAME
  - MD20094 $MC_SPIND_RIGID_TAPPING_M_NR
  - MD20095 $MC_EXTERN_RIGID_TAPPING_M_NR
  - MD20096 $MC_T_M_ADDRESS_EXT_IS_SPINO
  - MD22200 $MC_AUXFU_M_SYNC_TYPE
  - MD22530 $MC_TOCARR_CHANGE_M_CODE
  - MD22532 $MC_GEOAX_CHANGE_M_CODE
  - MD22534 $MC_TRAFO_CHANGE_M_CODE
  - MD22560 $MC_TOOL_CHANGE_M_CODE

S 機能

S (主軸機能)					
アドレス拡張 [10]		値			
値の範囲	意味	値の範囲	タイプ	意味	記述数 [8]
0 ... 20	主軸番号 [5]	0 ... ± 3.4028 exp38 [3]	REAL	主軸速度	3

[3] [5] [8] 概要の最後にある「脚注の意味」を参照してください。

[10] アドレス拡張が指定されていない場合、チャネルのメイン主軸がアドレス指定されます。

使用/用途

主軸速度。

他の情報

- S 機能は、初期設定として 3 番目の補助機能グループに割り当てられます。
- アドレス拡張がない場合、S 機能は、チャネルのメイン主軸を参照します。
- S 機能用マシンデータには、次のものがあります。
  MD22210 $MC_AUXFU_S_SYNC_TYPE (S 機能の出力タイミング)

H 機能

H 機能のこの機能は、PLC ユーザープログラムに組み込んでください。

H (補助機能) [10]					
アドレス拡張		値			
値の範囲	意味	値の範囲	タイプ	意味	記述数 [8]
0 ... 99	指定なし	- 2147483648 ... + 2147483647	INT	指定なし	3
		0 ... ± 3.4028 exp38 [2] [3] [4]	REAL		

[2] [3] [4] [8] 概要の最後にある「脚注の意味」を参照してください。

使用/用途

ユーザー専用の補助機能

他の情報

H 機能用マシンデータ:

- MD22110 $MC_AUXFU_H_TYPE_INT (H 補助機能のタイプは整数です)
- MD22230 $MC_AUXFU_H_SYNC_TYPE (H 機能の出力タイミング)

T 機能

工具名称は NC/PLC インタフェースに出力されません。

T (工具番号) [1] [5] [6]					
アドレス拡張		値			
値の範囲	意味	値の範囲	タイプ	意味	記述数 [8]
1 ... 12	主軸番号 (有効な工具管理機能と共に)	0 ... 32000 (有効な工具管理機能に応じたシンボル工具名称も)	INT	工具の選択	1

[1]、[5]、[6]、[8] 概要の最後にある「脚注の意味」を参照してください。

使用/用途

工具選択。

他の情報

- 必要に応じて工具番号またはロケーション番号を使用した工具の識別(「W1:工具補正 (ページ 1767)」の章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』
- T0 を選択すると、現在の工具が、新規工具に置き換わるのではなく、工具ホルダから取り外されます(初期設定)。
- T 機能用マシンデータ:
  MD22220 $MC_AUXFU_T_SYNC_TYPE (T 機能の出力タイミング)

D 機能

工具オフセットは、D0 を使用して解除します。初期設定は D1 です。

D (工具オフセット)					
アドレス拡張		値			
値の範囲	意味	値の範囲	タイプ	意味	記述数 [8)]
---	---	0 ... 9	INT	工具オフセットの選択	1

8) 概要の最後にある「脚注の意味」を参照してください。

使用/用途

工具オフセットの選択。

他の情報

- 初期設定:D1
- 工具交換後、次の設定を使用すれば、初期設定工具刃先をパラメータ設定できます。
  MD20270 $MC_CUTTING_EDGE_DEFAULT (プログラム指令されないときの工具刃先の基本設定)
- 工具オフセットの選択解除:D0
- D 機能用マシンデータ:
  MD22250 $MC_AUXFU_D_SYNC_TYPE (D 機能の出力タイミング)

DL 機能

DL により選択された追加工具オフセットは、有効な D 番号を参照します。

DL (追加工具オフセット)					
アドレス拡張		値			
値の範囲	意味	値の範囲	タイプ	意味	記述数 8)
---	---	0 ... 6	INT	サム工具オフセットの選択	1

8) 概要の最後にある「脚注の意味」を参照してください。

使用/用途

有効な工具オフセットに関する追加工具オフセットの選択

他の情報

- 初期設定:DL=0
- シンクロナイズドアクションを使用して DL 値を PLC に出力することはできません。
- 有効な DL 機能がないときの追加工具オフセットの初期設定
  MD20272 $MC_SUMCORR_DEFAULT (プログラム指令されないときの追加オフセットの初期設定)
- 追加工具オフセットの選択解除:DL=0
- DL 機能用マシンデータ:
  MD22252 $MC_AUXFU_DL_SYNC_TYPE (DL 機能の出力タイミング)

F 機能

F (送り速度)					
アドレス拡張		値			
値の範囲	意味	値の範囲	タイプ	意味	記述数 8)
---	---	0.001 ... 999 999.999	REAL	軌跡送り速度	6

8) 概要の最後にある「脚注の意味」を参照してください。

使用/用途

軌跡速度。

他の情報

F 機能用マシンデータ:

- MD22240 $MC_AUXFU_F_SYNC_TYPE (F 機能の出力タイミング)

FA 機能

FA (軸別送り速度)					
アドレス拡張		値			
値の範囲	意味	値の範囲	タイプ	意味	記述数 [8)]
1 - 31	軸番号	0.001 ... 999 999.999	REAL	軸送り速度	6

[8)] 概要の最後にある「脚注の意味」を参照してください。

使用/用途

軸別速度。

他の情報

F 機能用マシンデータ:

- MD22240 $MC_AUXFU_F_SYNC_TYPE (F 機能の出力タイミング)

脚注の意味

[1)] 工具管理機能が有効になっている場合、T 交換信号も T ワードもチャネル別 NC/PLC インタフェースに出力されません。

[2)] MD22110 $MC_AUXFU_H_TYPE_INT を使用すれば、ユーザーは、値のタイプを選択できます。

[3)] 操作パネル画面上の表示選択には制限があるため、表示される REAL タイプの値は、次の値に制限されます。

–999 999 999.9999 ～ 999 999 999.9999

NC は内部的に計算しますが、完全に正確なわけではありません。

[4)] 次のマシンデータを設定すると、REAL 値は丸められ、 PLC に出力されます。

MD22110 $MC_AUXFU_H_TYPE_INT = 1 (H 補助機能のタイプは整数です)

PLC ユーザープログラムは、マシンデータ設定に応じて伝送された値を解釈する必要があります。

5) 工具管理機能が有効になっている場合、アドレス拡張の意味をパラメータ設定できます。アドレス拡張= 0 は、値を、メイン主軸番号の値で置き換える必要があることを意味します。つまり、アドレス拡張をプログラム指令しないのと同じです。

ブロック検索中に収集された補助機能 M19「主軸の位置決め」は PLC に出力されません。

6) M6:アドレス拡張の数値範囲:

- 工具管理機能なし:0 ... 99

- 工具管理機能あり:0 ... 最大主軸番号

0:メイン主軸番号またはマスタ工具ホルダの値に置き換えられます。

7) 工具管理機能が有効になっている場合、プログラム指令されているアドレス拡張に関係なく、補助機能 M6「工具交換」は、パートプログラムブロックで 1 回だけプログラム指令できます。

8) 1 パートプログラムブロックあたりの最大補助機能数

9.2 事前定義補助機能

機能

すべての事前定義補助機能はシステムファンクションに割り当てられていて、変更することはできません。事前定義補助機能がパートプログラム/サイクルでプログラムされている場合、NC/PLC インタフェースを介して PLC に出力され、対応するシステムファンクションが NC で実行されます。

事前定義補助機能の定義

事前定義補助機能のパラメータはマシンデータに設定され、場合によっては変更が可能です。補助機能に割り当てられているすべてのマシンデータは、同じインデックス**<n>** を持ちます。

- MD22040 $MC_AUXFU_PREDEF_GROUP[<n>] (事前定義補助機能のグループ割り当て)
- MD22050 $MC_AUXFU_PREDEF_TYPE[<n>] (事前定義補助機能のタイプ)
- MD22060 $MC_AUXFU_PREDEF_EXTENSION[<n>] (事前定義補助機能のアドレス拡張)

- MD22070 $MC_AUXFU_PREDEF_VALUE[<n>] (事前定義補助機能の値)
- MD22080 $MC_AUXFU_PREDEF_SPEC[<n>] (事前定義補助機能の出力動作)

9.2.1 一覧: 事前定義補助機能

以下の表にリストされたパラメータの意味

パラメータ	意味
インデックス<n>	補助機能のパラメータのマシンデータのインデックス
タイプ	MD22050 $MC_AUXFU_PREDEF_TYPE[<n>]
アドレス拡張	MD22060 $MC_AUXFU_PREDEF_EXTENSION[<n>]
値	MD22070 $MC_AUXFU_PREDEF_VALUE[<n>]
グループ	MD22040 $MC_AUXFU_PREDEF_GROUP[<n>]

事前定義補助機能

一般的な補助機能、パート 1					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
ストップ	0	M	0	0	1
オプショナルストップ	1	M	0	1	1
サブプログラム終了	2	M	0	2	1
	3	M	0	17	1
	4	M	0	30	1
工具交換	5	M	(0)	6 [1)]	(1)

主軸用補助機能、主軸 1					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	6	M	1	3	(2)
主軸左回転	7	M	1	4	(2)
主軸停止	8	M	1	5	(2)
主軸の位置決め	9	M	1	19	(2)

主軸用補助機能、主軸 1					
システムファンクション	**インデックス<n>**	**タイプ**	**アドレス拡張**	**値**	**グループ**
軸モード	10	M	1	70 [2)]	(2)
自動ギヤ選択	11	M	1	40	(4)
ギヤ選択 1	12	M	1	41	(4)
ギヤ選択 2	13	M	1	42	(4)
ギヤ選択 3	14	M	1	43	(4)
ギヤ選択 4	15	M	1	44	(4)
ギヤ選択 5	16	M	1	45	(4)
主軸速度	17	S	1	-1	(3)

一般的な補助機能、パート 2					
システムファンクション	**インデックス<n>**	**タイプ**	**アドレス拡張**	**値**	**グループ**
送り速度	18	F	0	-1	(1)
刃先選択	19	D	0	-1	(1)
DL	20	L	0	-1	(1)
工具選択	21	T	(0)	-1	(1)
(関連の)プログラムストップ	22	M	0	-1 [3)]	1
(関連の)オプショナルストップ	23	M	0	-1 [4)]	1
サブプログラム終了	24	M	0	-1 [5)]	1
ニブリング	25	M	0	20 [6)]	(10)
ニブリング	26	M	0	23 [6)]	(10)
ニブリング	27	M	0	22 [6)]	(11)
ニブリング	28	M	0	25 [6)]	(11)
ニブリング	29	M	0	26 [6)]	(12)
ニブリング	30	M	0	122 [6)]	(11)
ニブリング	31	M	0	125 [6)]	(11)
ニブリング	32	M	0	27 [6)]	(12)

主軸用補助機能、主軸 2					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	33	M	2	3	(72)
主軸左回転	34	M	2	4	(72)
主軸停止	35	M	2	5	(72)
主軸の位置決め	36	M	2	19	(72)
軸モード	37	M	2	70 [2)]	(72)
自動ギヤ選択	38	M	2	40	(74)
ギヤ選択 1	39	M	2	41	(74)
ギヤ選択 2	40	M	2	42	(74)
ギヤ選択 3	41	M	2	43	(74)
ギヤ選択 4	42	M	2	44	(74)
ギヤ選択 5	43	M	2	45	(74)
主軸速度	44	S	2	-1	(73)

主軸用補助機能、主軸 3					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	45	M	3	3	(75)
主軸左回転	46	M	3	4	(75)
主軸停止	47	M	3	5	(75)
主軸の位置決め	48	M	3	19	(75)
軸モード	49	M	3	70 [2)]	(75)
自動ギヤ選択	50	M	3	40	(77)
ギヤ選択 1	51	M	3	41	(77)
ギヤ選択 2	52	M	3	42	(77)
ギヤ選択 3	53	M	3	43	(77)
ギヤ選択 4	54	M	3	44	(77)

主軸用補助機能、主軸 3					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
ギヤ選択 5	55	M	3	45	(77)
主軸速度	56	S	3	-1	(76)

主軸用補助機能、主軸 4					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	57	M	4	3	(78)
主軸左回転	58	M	4	4	(78)
主軸停止	59	M	4	5	(78)
主軸の位置決め	60	M	4	19	(78)
軸モード	61	M	4	70 [2)]	(78)
自動ギヤ選択	62	M	4	40	(80)
ギヤ選択 1	63	M	4	41	(80)
ギヤ選択 2	64	M	4	42	(80)
ギヤ選択 3	65	M	4	43	(80)
ギヤ選択 4	66	M	4	44	(80)
ギヤ選択 5	67	M	4	45	(80)
主軸速度	68	S	4	-1	(79)

主軸用補助機能、主軸 5					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	69	M	5	3	(81)
主軸左回転	70	M	5	4	(81)
主軸停止	71	M	5	5	(81)
主軸の位置決め	72	M	5	19	(81)
軸モード	73	M	5	70 [2)]	(81)
自動ギヤ選択	74	M	5	40	(83)
ギヤ選択 1	75	M	5	41	(83)

主軸用補助機能、主軸 5					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
ギヤ選択 2	76	M	5	42	(83)
ギヤ選択 3	77	M	5	43	(83)
ギヤ選択 4	78	M	5	44	(83)
ギヤ選択 5	79	M	5	45	(83)
主軸速度	80	S	5	-1	(82)

主軸用補助機能、主軸 6					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	81	M	6	3	(84)
主軸左回転	82	M	6	4	(84)
主軸停止	83	M	6	5	(84)
主軸の位置決め	84	M	6	19	(84)
軸モード	85	M	6	70 [2)]	(84)
自動ギヤ選択	86	M	6	40	(86)
ギヤ選択 1	87	M	6	41	(86)
ギヤ選択 2	88	M	6	42	(86)
ギヤ選択 3	89	M	6	43	(86)
ギヤ選択 4	90	M	6	44	(86)
ギヤ選択 5	91	M	6	45	(86)
主軸速度	92	S	6	-1	(85)

主軸用補助機能、主軸 7					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	93	M	7	3	(87)
主軸左回転	94	M	7	4	(87)
主軸停止	95	M	7	5	(87)
主軸の位置決め	96	M	7	19	(87)

主軸用補助機能、主軸 7					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
軸モード	97	M	7	70 2)	(87)
自動ギヤ選択	98	M	7	40	(89)
ギヤ選択 1	99	M	7	41	(89)
ギヤ選択 2	100	M	7	42	(89)
ギヤ選択 3	101	M	7	43	(89)
ギヤ選択 4	102	M	7	44	(89)
ギヤ選択 5	103	M	7	45	(89)
主軸速度	104	S	7	-1	(88)

主軸用補助機能、主軸 8					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	105	M	8	3	(90)
主軸左回転	106	M	8	4	(90)
主軸停止	107	M	8	5	(90)
主軸の位置決め	108	M	8	19	(90)
軸モード	109	M	8	70 2)	(90)
自動ギヤ選択	110	M	8	40	(92)
ギヤ選択 1	111	M	8	41	(92)
ギヤ選択 2	112	M	8	42	(92)
ギヤ選択 3	113	M	8	43	(92)
ギヤ選択 4	114	M	8	44	(92)
ギヤ選択 5	115	M	8	45	(92)
主軸速度	116	S	8	-1	(91)

主軸用補助機能、主軸 9					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	117	M	9	3	(93)
主軸左回転	118	M	9	4	(93)
主軸停止	119	M	9	5	(93)
主軸の位置決め	120	M	9	19	(93)
軸モード	121	M	9	70 [2)]	(93)
自動ギヤ選択	122	M	9	40	(95)
ギヤ選択 1	123	M	9	41	(95)
ギヤ選択 2	124	M	9	42	(95)
ギヤ選択 3	125	M	9	43	(95)
ギヤ選択 4	126	M	9	44	(95)
ギヤ選択 5	127	M	9	45	(95)
主軸速度	128	S	9	-1	(94)

主軸用補助機能、主軸 10					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	129	M	10	3	(96)
主軸左回転	130	M	10	4	(96)
主軸停止	131	M	10	5	(96)
主軸の位置決め	132	M	10	19	(96)
軸モード	133	M	10	70 [2)]	(96)
自動ギヤ選択	134	M	10	40	(98)
ギヤ選択 1	135	M	10	41	(98)
ギヤ選択 2	136	M	10	42	(98)
ギヤ選択 3	137	M	10	43	(98)
ギヤ選択 4	138	M	10	44	(98)

主軸用補助機能、主軸 10					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
ギヤ選択 5	139	M	10	45	(98)
主軸速度	140	S	10	-1	(97)

主軸用補助機能、主軸 11					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	141	M	11	3	(99)
主軸左回転	142	M	11	4	(99)
主軸停止	143	M	11	5	(99)
主軸の位置決め	144	M	11	19	(99)
軸モード	145	M	11	70 [2)]	(99)
自動ギヤ選択	146	M	11	40	(101)
ギヤ選択 1	147	M	11	41	(101)
ギヤ選択 2	148	M	11	42	(101)
ギヤ選択 3	149	M	11	43	(101)
ギヤ選択 4	150	M	11	44	(101)
ギヤ選択 5	151	M	11	45	(101)
主軸速度	152	S	11	-1	(100)

主軸用補助機能、主軸 12					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	153	M	11	3	(102)
主軸左回転	154	M	12	4	(102)
主軸停止	155	M	12	5	(102)
主軸の位置決め	156	M	12	19	(102)
軸モード	157	M	12	70 [2)]	(102)
自動ギヤ選択	158	M	12	40	(104)
ギヤ選択 1	159	M	12	41	(104)

主軸用補助機能、主軸 12					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
ギヤ選択 2	160	M	12	42	(104)
ギヤ選択 3	161	M	12	43	(104)
ギヤ選択 4	162	M	12	44	(104)
ギヤ選択 5	163	M	12	45	(104)
主軸速度	164	S	12	-1	(103)

主軸用補助機能、主軸 13					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	165	M	13	3	(105)
主軸左回転	166	M	13	4	(105)
主軸停止	167	M	13	5	(105)
主軸の位置決め	168	M	13	19	(105)
軸モード	169	M	13	70 [2)]	(105)
自動ギヤ選択	170	M	13	40	(107)
ギヤ選択 1	171	M	13	41	(107)
ギヤ選択 2	172	M	13	42	(107)
ギヤ選択 3	173	M	13	43	(107)
ギヤ選択 4	174	M	13	44	(107)
ギヤ選択 5	175	M	13	45	(107)
主軸速度	176	S	13	-1	(106)

主軸用補助機能、主軸 14					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	177	M	14	3	(108)
主軸左回転	178	M	14	4	(108)
主軸停止	179	M	14	5	(108)
主軸の位置決め	180	M	14	19	(108)

主軸用補助機能、主軸 14					
システムファンクション	**インデックス<n>**	**タイプ**	**アドレス拡張**	**値**	**グループ**
軸モード	181	M	14	70 [2)]	(108)
自動ギヤ選択	182	M	14	40	(110)
ギヤ選択 1	183	M	14	41	(110)
ギヤ選択 2	184	M	14	42	(110)
ギヤ選択 3	185	M	14	43	(110)
ギヤ選択 4	186	M	14	44	(110)
ギヤ選択 5	187	M	14	45	(110)
主軸速度	188	S	14	-1	(109)

主軸用補助機能、主軸 15					
システムファンクション	**インデックス<n>**	**タイプ**	**アドレス拡張**	**値**	**グループ**
主軸右回転	189	M	15	3	(111)
主軸左回転	190	M	15	4	(111)
主軸停止	191	M	15	5	(111)
主軸の位置決め	192	M	15	19	(111)
軸モード	193	M	15	70 [2)]	(111)
自動ギヤ選択	194	M	15	40	(113)
ギヤ選択 1	195	M	15	41	(113)
ギヤ選択 2	196	M	15	42	(113)
ギヤ選択 3	197	M	15	43	(113)
ギヤ選択 4	198	M	15	44	(113)
ギヤ選択 5	199	M	15	45	(113)
主軸速度	200	S	15	-1	(112)

主軸用補助機能、主軸 16					
システムファンクション	**インデックス<n>**	**タイプ**	**アドレス拡張**	**値**	**グループ**
主軸右回転	201	M	16	3	(114)
主軸左回転	202	M	16	4	(114)
主軸停止	203	M	16	5	(114)
主軸の位置決め	204	M	16	19	(114)
軸モード	205	M	16	70 [2)]	(114)
自動ギヤ選択	206	M	16	40	(116)
ギヤ選択 1	207	M	16	41	(116)
ギヤ選択 2	208	M	16	42	(116)
ギヤ選択 3	209	M	16	43	(116)
ギヤ選択 4	210	M	16	44	(116)
ギヤ選択 5	211	M	16	45	(116)
主軸速度	212	S	16	-1	(115)

主軸用補助機能、主軸 17					
システムファンクション	**インデックス<n>**	**タイプ**	**アドレス拡張**	**値**	**グループ**
主軸右回転	213	M	17	3	(117)
主軸左回転	214	M	17	4	(117)
主軸停止	215	M	17	5	(117)
主軸の位置決め	216	M	17	19	(117)
軸モード	217	M	17	70 [2)]	(117)
自動ギヤ選択	218	M	17	40	(119)
ギヤ選択 1	219	M	17	41	(119)
ギヤ選択 2	220	M	17	42	(119)
ギヤ選択 3	221	M	17	43	(119)
ギヤ選択 4	222	M	17	44	(119)

主軸用補助機能、主軸 17					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
ギヤ選択 5	223	M	17	45	(119)
主軸速度	224	S	17	-1	(118)

主軸用補助機能、主軸 18					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	225	M	18	3	(120)
主軸左回転	226	M	18	4	(120)
主軸停止	227	M	18	5	(120)
主軸の位置決め	228	M	18	19	(120)
軸モード	229	M	18	70 [2)]	(120)
自動ギヤ選択	230	M	18	40	(122)
ギヤ選択 1	231	M	18	41	(122)
ギヤ選択 2	232	M	18	42	(122)
ギヤ選択 3	233	M	18	43	(122)
ギヤ選択 4	234	M	18	44	(122)
ギヤ選択 5	235	M	18	45	(122)
主軸速度	236	S	18	-1	(121)

主軸用補助機能、主軸 19					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	237	M	19	3	(123)
主軸左回転	238	M	19	4	(123)
主軸停止	239	M	19	5	(123)
主軸の位置決め	240	M	19	19	(123)
軸モード	241	M	19	70 [2)]	(123)
自動ギヤ選択	242	M	19	40	(125)
ギヤ選択 1	243	M	19	41	(125)

主軸用補助機能、主軸 19					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
ギヤ選択 2	244	M	19	42	(125)
ギヤ選択 3	245	M	19	43	(125)
ギヤ選択 4	246	M	19	44	(125)
ギヤ選択 5	247	M	19	45	(125)
主軸速度	248	S	19	-1	(124)

主軸用補助機能、主軸 20					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
主軸右回転	249	M	20	3	(126)
主軸左回転	250	M	20	4	(126)
主軸停止	251	M	20	5	(126)
主軸の位置決め	252	M	20	19	(126)
軸モード	253	M	20	70 [2)]	(126)
自動ギヤ選択	254	M	20	40	(128)
ギヤ選択 1	255	M	20	41	(128)
ギヤ選択 2	256	M	20	42	(128)
ギヤ選択 3	257	M	20	43	(128)
ギヤ選択 4	258	M	20	44	(128)
ギヤ選択 5	259	M	20	45	(128)
主軸速度	260	S	20	-1	(127)

工具ホルダ用補助機能、T 補助機能					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
工具選択	261	T	1	-1	129
工具選択	262	T	2	-1	130
工具選択	263	T	3	-1	131
工具選択	264	T	4	-1	132

工具ホルダ用補助機能、T 補助機能					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
工具選択	265	T	5	-1	133
工具選択	266	T	6	-1	134
工具選択	267	T	7	-1	135
工具選択	268	T	8	-1	136
工具選択	269	T	9	-1	137
工具選択	270	T	10	-1	138
工具選択	271	T	11	-1	139
工具選択	272	T	12	-1	140
工具選択	273	T	13	-1	141
工具選択	274	T	14	-1	142
工具選択	275	T	15	-1	143
工具選択	276	T	16	-1	144
工具選択	277	T	17	-1	145
工具選択	278	T	18	-1	146
工具選択	279	T	19	-1	147
工具選択	280	T	20	-1	148

工具ホルダ用補助機能、M6 補助機能					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
工具交換	281	M	1	6 [1)]	149
工具交換	282	M	2	6 [1)]	150
工具交換	283	M	3	6 [1)]	151
工具交換	284	M	4	6 [1)]	152
工具交換	285	M	5	6 [1)]	153
工具交換	286	M	6	6 [1)]	154
工具交換	287	M	7	6 [1)]	155
工具交換	288	M	8	6 [1)]	156
工具交換	289	M	9	6 [1)]	157

工具ホルダ用補助機能、M6 補助機能					
システムファンクション	インデックス<n>	タイプ	アドレス拡張	値	グループ
工具交換	290	M	10	6 1)	158
工具交換	291	M	11	6 1)	159
工具交換	292	M	12	6 1)	160
工具交換	293	M	13	6 1)	161
工具交換	294	M	14	6 1)	162
工具交換	295	M	15	6 1)	163
工具交換	296	M	16	6 1)	164
工具交換	297	M	17	6 1)	165
工具交換	298	M	18	6 1)	166
工具交換	299	M	19	6 1)	167
工具交換	300	M	20	6 1)	168

記号解説:

( ) 値は変更可能です。

1) 値はマシンデータに応じて異なります。

MD22560 $MC_TOOL_CHANGE_M_MODE (工具交換用の M 機能)

2) 以下のマシンデータを使用すれば、異なる値をプリセットできます。

MD20095 $MC_EXTERN_RIGID_TAPPING_M_NR (制御軸モード(拡張モード)に切り替えるための M 機能)

MD20094 $MC_SPIND_RIGID_TAPPING_M_NR (制御軸モードに切り替えるためのM機能)

注記

値 70 は常に PLC に出力されます。

3) 値は、次のマシンデータを使用して設定されます。

MD22254 $MC_AUXFU_ASSOC_M0_VALUE (プログラムストップ用追加 M 機能)

4) 値は、次のマシンデータを使用して設定されます。

MD22256 $MC_AUXFU_ASSOC_M1_VALUE (オプショナルストップ用追加 M 機能)

5) 値は、次のマシンデータを使用して設定されます。
MD10714 $MN_M_NO_FCT_EOP (リセット後に有効な主軸用 M 機能)

6) 値は、次のマシンデータを使用して設定されます。
MD26008 $MC_NIBBLE_PUNCH_CODE (M 機能の定義)

9.2.2 概要:出力動作

以下の表にリストされたパラメータの意味

パラメータ	意味
インデックス<n>	補助機能のパラメータのマシンデータのインデックス
出力動作	MD22080 $MC_AUXFU_PREDEF_SPEC[<n>], ビット 0 ... 18 ビット 19 ... 31:予約済

事前定義補助機能の出力動作

システムファンクション	インデックス<n>																			
	出力動作、bit																			
18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
停止	0	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
オプショナルストップ	1	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
エンドオブプログラム	2	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
	3	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
	4	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
工具交換	5	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
主軸右回転	6	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)

システムファンクション	インデックス<n>																			
		出力動作、bit																		
主軸左回転	7	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
主軸停止	8	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
主軸位置決め	9	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
軸モード	10	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(0)	(1)	(0)	(0)	(0)	(0)	(1)
自動ギヤ選択	11	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
ギヤ選択 1	12	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
ギヤ選択 2	13	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
ギヤ選択 3	14	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
ギヤ選択 4	15	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
ギヤ選択 5	16	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	0	0	1	(0)	(0)	(0)	(0)	(1)
主軸速度	17	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(1)	(0)	(0)	(0)	0	(0)	(1)
送り速度	18	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(1)	(0)	0	(1)	0	(0)	(1)
刃先選択	19	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(0)	(1)	0	(0)	0	(0)	(1)
DL	20	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(0)	(1)	0	(0)	0	(0)	(1)
工具選択	21	0	0	0	(0)	0	0	0	(0)	(0)	(0)	0	(0)	(0)	(1)	0	(0)	0	(0)	(1)
(関連の)プログラムストップ	22	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)

システムファンクション	インデックス<n>																			
		出力動作、bit																		
(関連の)オプショナルストップ	23	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
エンドオブプログラム	24	0	0	0	(0)	0	0	0	0	(0)	0	0	1	0	0	0	0	0	(0)	(1)
ニブリング	25	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
ニブリング	26	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
ニブリング	27	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
ニブリング	28	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
ニブリング	29	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
ニブリング	30	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
ニブリング	31	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)
ニブリング	32	0	0	0	(0)	0	0	0	(0)	(0)	(0)	(0)	(0)	(1)	(0)	0	(0)	(0)	(0)	(1)

( ) 値は変更可能です。

ビットの有効桁

Bit	意味
0	OB1 サイクル後の応答「通常」
	通常応答が指定された補助機能は、OB1 サイクルの最初に NC/PLC インタフェースに出力されます。補助機能用の変更信号は、補助機能が有効であることを PLC ユーザープログラムに示します。 オーガニゼーションブロック OB1 を一度実行した直後に、補助機能に対する応答が行われます。これは、全ての PLC ユーザーサイクルに対応しています。 通常応答が指定された補助機能は、プログラムされているパートプログラムブロックと同期して出力されます。 補助機能の応答前に、パートプログラムブロックの実行、たとえば軌跡や位置決め軸の移動が完了している場合、PLC による応答後まで、ブロック変更が遅延されます。 連続軌跡モードでは、移動動作中、およびブロックの終わりに到達する前に補助機能が PLC によって出力される場合は、通常応答の補助機能との組み合わせにより一定の軌跡速度を維持できます。
1	OB40 での応答「高速」
	高速応答が指定された補助機能は、次の OB1 サイクルの前に NC/PLC インタフェースに出力されます。補助機能用の変更信号は、補助機能が有効であることを PLC ユーザープログラムに示します。 補助機能は、次の OB40 サイクルで PLC 基本プログラムにより直ちに応答されます。 補助機能の応答は、対応する PLC ユーザーファンクションが実行されたことの確認ではありません。 補助機能はまだ OB1 サイクルで実行されています。 このため、OB1 サイクルの実行完了後までは、PLC への次の補助機能は出力できません。 これが連続軌跡モードで問題になる(軌跡速度が低下する) のは、特に、複数の連続パートプログラムブロックで、高速応答が指定された補助機能が出力される場合です。 高速応答が指定された補助機能を使用しても、ブロックと同期して PLC ユーザープログラムが応答するとは限りません。 注記 「高速補助機能」の補助機能の出力動作のパラメータ設定が可能なのは、ユーザー定義補助機能と組み合わせている場合に限ります。
2	事前定義補助機能なし
	この設定では、事前定義補助機能はユーザー定義の補助機能と同様に扱われます。 すると、この補助機能は、対応するシステムファンクションを起動せずに、PLC への出力だけを実行します。 例: 移動前出力および通常の応答を行うユーザー定義の補助機能への、「主軸位置決め」補助機能(インデックス 9)の再設定。 MD22080 $MC_AUXFU_PREDEF_SPEC [9] = 'H25' (100101B)

Bit	意味
3	PLC への出力なし
	補助機能は PLC に出力されません。
4	PLC による応答後の主軸応答
	関連するシステムファンクションは、PLC による応答後にのみ実行されます。
5	移動前の出力
	補助機能は、パートプログラムブロックにプログラムされた移動動作(軌跡および/またはブロック関連の位置決め軸の移動)の前に PLC に出力されます。
6	移動中の出力
	パートプログラムブロックでプログラムされた移動動作(軌跡やブロック単位の位置決め軸動作)の完了後に、補助機能が出力されます。
7	ブロックエンドでの出力
	補助機能は、パートプログラムブロックにプログラムされた移動動作(軌跡および/またはブロック関連の位置決め軸の移動)後に PLC に出力されます。
8	ブロックサーチ後に出力されません、タイプ 1、2、4
	ブロックサーチ、タイプ 1、2、4: ブロックサーチ中に収集された補助機能は、出力されません。
9	プログラムテストによるブロックサーチ中の収集(タイプ 5、SERUPRO)
	プログラムテストによるブロックサーチの場合は、補助機能を以下の変数でグループ別に収集できます。 • $AC_AUXFU_M_VALUE[<n>] • $AC_AUXFU_M_EXT[<n>] • $AC_AUXFU_M_STATE[<n>]
10	プログラムテストによるブロックサーチ中の出力なし(タイプ 5、SERUPRO)
	プログラムテストによるブロックサーチの場合、補助機能は PLC に出力されません。
11	複数のチャネルにわたる補助機能(SERUPRO)
	プログラムテストによるブロックサーチ(SERUPRO)の場合は、ヘルプ機能を補助機能のグローバルリストで複数のチャネルにわたって収集できます。 注記 補助機能グループの場合、グループの最後の補助機能だけが常に収集されます。
12	シンクロナイズドアクションを介して実行される出力(読み取り専用)
	シンクロナイズドアクションを介して PLC に補助機能が出力された場合、ビットが設定されます。
13	自動補助機能(読み取り専用)
	補助機能が PLC に自動的に出力された場合、ビットが設定されます。

Bit	意味
14	有効な M01 (読み取り専用)
	有効な M01 の場合に、補助機能が PLC に自動的に出力された場合、ビットが設定されます。
15	位置決めテスト運転中の出力なし
	ランインテスト中に、補助機能は PLC に出力されません。
16	ニブリングオフ
17	ニブリングオン
18	ニブリング

注記

補助機能に出力動作が定義されていない場合、次の既定の出力動作が有効になります。

- ビット 0 = 1: 1 OB1 サイクルの出力時間
- ビット 7 = 1:ブロックエンドでの出力

9.2.3 パラメータ設定

9.2.3.1 グループ割り当て

ブロックサーチの場合の補助機能の取り扱いは、補助機能のグループ割り当てによって定義されます。 使用可能な 168 個の補助機能グループが、事前定義グループおよびユーザー定義グループに分割されます。

事前定義グループ:	1 ... 4	10 ... 12	72 ... 168
ユーザー定義グループ:	5 ... 9	13 ... 71	

標準では、それぞれの事前定義補助機能が補助機能グループに割り当てられます。 大部分の事前定義補助機能では、この割り当ては以下のマシンデータを使用して変更することができます。

MD22040 $MC_AUXFU_PREDEF_GROUP[<n>] (事前定義補助機能のグループ割り当て)

補助機能がどのグループにも割り当てられていない場合、値「0」をマシンデータに入力してください。

以下のインデックス<n>を持つ事前定義補助機能の場合、グループ割り当ては変更できません。 0, 1, 2, 3, 4, 22, 23, 24

注記

1. 補助機能グループとブロックサーチ

ブロックサーチの場合、1 番目の補助機能グループの補助機能だけが収集されますが、出力はされません。

9.2.3.2 タイプ、アドレス拡張、値

補助機能は、タイプ、アドレス拡張、および値の各パラメータでプログラム指令します(「補助機能のプログラミング (ページ 520)」の章を参照してください)。

タイプ

補助機能の種類は、たとえば、次のような「タイプ」を使用して定義されます。

"M"	補助機能
"S"	主軸機能
"F"	送り速度

この設定は、次のマシンデータを使用しておこないます。

MD22050 $MC_AUXFU_PREDEF_TYPE[<n>] (事前定義補助機能のタイプ)

注記

事前定義補助機能では、「タイプ」は変更できません。

アドレス拡張

補助機能の「アドレス拡張」は、同じタイプの異なる要素のアドレス指定用です。 事前定義補助機能の場合、「アドレス拡張」の値は、補助機能が適用される主軸番号です。

この設定は、次のマシンデータを使用しておこないます。

MD22060 $MC_AUXFU_PREDEF_EXTENSION[<n>] (事前定義補助機能のアドレス拡張)

補助機能のグループ化

チャネルの主軸すべての補助機能を同じ補助機能グループに割り当てるには、「アドレス拡張」パラメータに値「-1」を入力します。

例:

補助機能 M3 (マシンデータインデックス= 6)は、すべてのチャネルの主軸の 2 番目の補助機能グループに割り当てます。

```
MD22040 $MC_AUXFU_PREDEF_GROUP[ 6 ]      = 2
MD22050 $MC_AUXFU_PREDEF_TYPE[ 6 ]       = "M"
MD22060                                  = -1
$MC_AUXFU_PREDEF_EXTENSION[ 6 ]
MD22070 $MC_AUXFU_PREDEF_VALUE[ 6 ]      = 3
```

値

パラメータの「値」と「タイプ」は、補助機能の意味、つまりこの補助機能に基づいて起動されるシステムファンクションを定義します。

補助機能の「値」は、次のマシンデータで定義します。

MD22070 $MC_AUXFU_PREDEF_VALUE[<n>] (事前定義補助機能の値)

注記

事前定義補助機能では、「値」を変更できません。 一部の事前定義補助機能に対しては、その他のマシンデータで「値」を再設定できます(「関連補助機能 (ページ 515)」の章を参照してください)。

9.2.3.3 出力動作

パラメータ「出力動作」は、事前定義補助機能がいつ NC/PLC インタフェースに出力されるかと、PLC によっていつ応答されるかを定義します。

この設定は、次のマシンデータを使用して行います。

MD22080 $MC_AUXFU_PREDEF_SPEC[<n>] (事前定義補助機能の出力動作)

移動に関連する出力動作

移動前の出力

- 直前のパートプログラムの移動動作(軌跡やブロック単位の位置決め軸動作)はイグザクトストップで終了します。
- 現在のパートプログラムブロックの始めに、補助機能が出力されます。
- 実際のパートプログラムブロックの移動動作(軌跡および/または位置決め軸の動作)は、PLC による補助機能の応答後に初めて開始されます。
  - 1 OB1 サイクルの出力時間(通常応答): 1 OB1 サイクル後
  - 1 OB40 サイクルの出力時間(高速応答): 1 OB40 サイクル後

移動中の出力

- 移動動作(軌跡や位置決め軸動作)の最初に、補助機能が出力されます。
- ブロックの終わりまでにかかる時間が、PLC による補助機能の応答にかかる時間より長くなるように、現在のパートプログラムブロックの軌跡速度を減速します。
  - 1 OB1 サイクルの出力時間(通常応答): 1 OB1 サイクル
  - 1 OB40 サイクルの出力時間(高速応答): 1 OB40 サイクル

移動後の出力

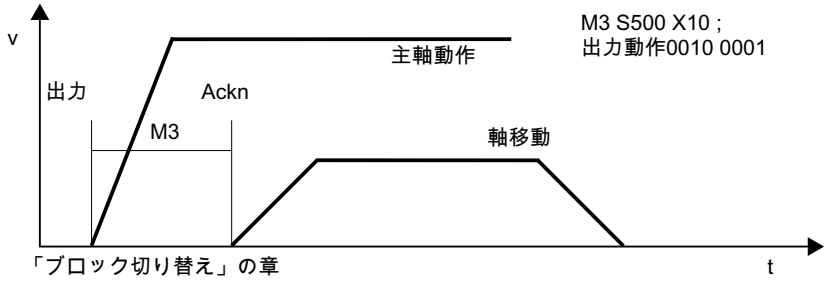
- 現在のパートプログラムの移動動作(軌跡やブロック単位の位置決め軸動作)はイグザクトストップで終了します。
- 移動動作の完了後に、補助機能が出力されます。
- PLC による補助機能の応答後に、ブロック変更が実行されます。
  - 1 OB1 サイクルの出力時間(通常応答): 1 OB1 サイクル後
  - 1 OB40 サイクルの出力時間(高速応答): 1 OB40 サイクル後

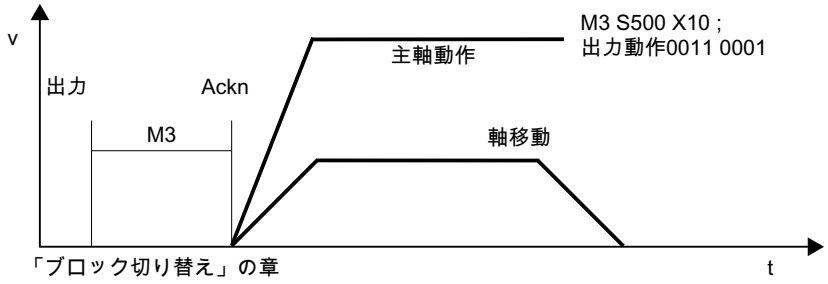
各種の出力動作の例

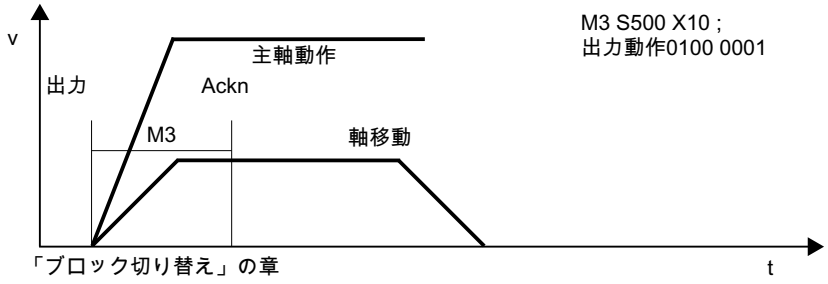
後述の各図に、次の事柄に関する各種動作を記載します。

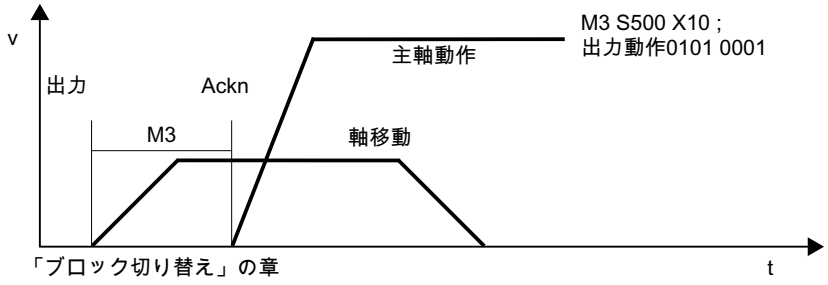
- 補助機能の出力と応答
- 主軸応答(速度変化)
- 移動動作(速度変化)

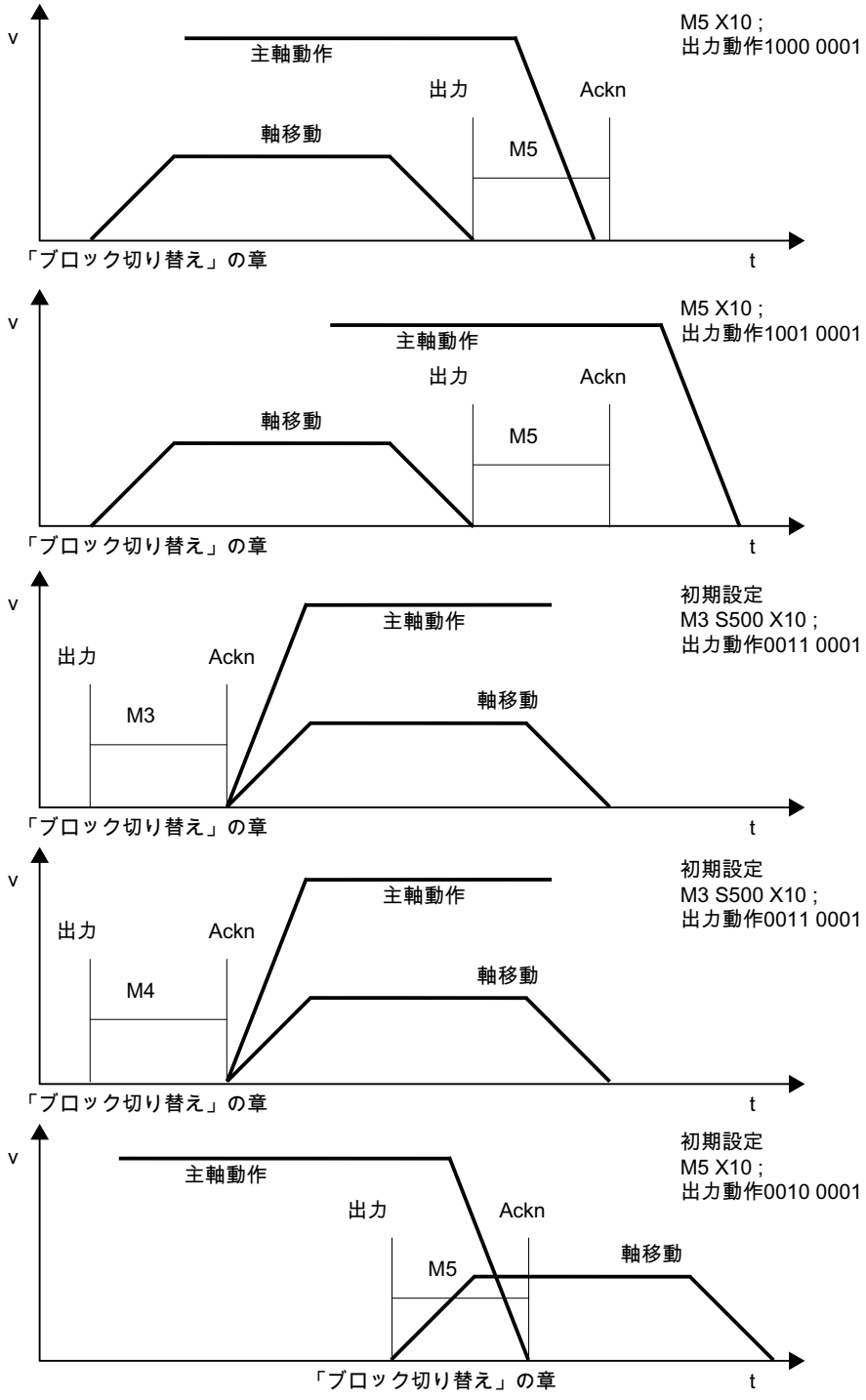
「出力動作」の図で指定されているバイナリ値は、パラメータに設定された出力動作(MD22080)を指しています。

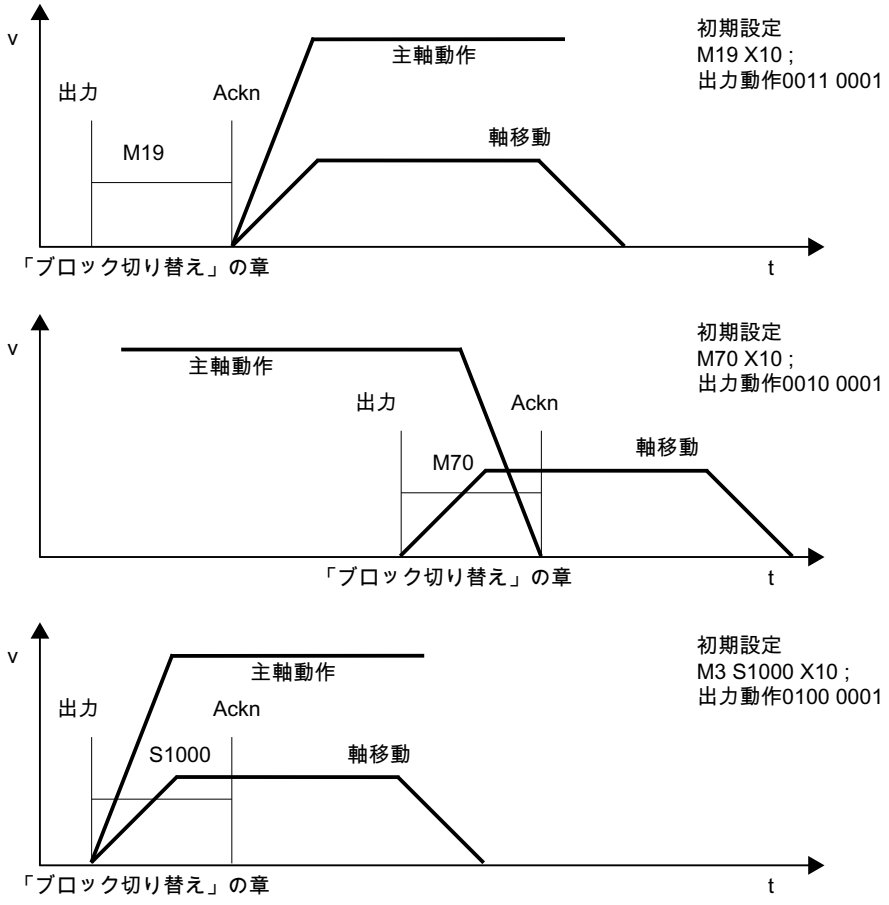

v
出力
Ackn
M3
主軸動作
軸移動
M3 S500 X10 ;
出力動作0010 0001
「ブロック切り替え」の章
t


v
出力
Ackn
M3
主軸動作
軸移動
M3 S500 X10 ;
出力動作0011 0001
「ブロック切り替え」の章
t


v
出力
Ackn
M3
主軸動作
軸移動
M3 S500 X10 ;
出力動作0100 0001
「ブロック切り替え」の章
t


v
出力
Ackn
M3
主軸動作
軸移動
M3 S500 X10 ;
出力動作0101 0001
「ブロック切り替え」の章
t


v
主軸動作
出力
Ackn
軸移動
M5
M5 X10 ;
出力動作1000 0001
「ブロック切り替え」の章
t
v
主軸動作
出力
Ackn
軸移動
M5
M5 X10 ;
出力動作1001 0001
「ブロック切り替え」の章
t
v
主軸動作
出力
Ackn
M3
軸移動
初期設定
M3 S500 X10 ;
出力動作0011 0001
「ブロック切り替え」の章
t
v
主軸動作
出力
Ackn
M4
軸移動
初期設定
M3 S500 X10 ;
出力動作0011 0001
「ブロック切り替え」の章
t
v
主軸動作
出力
Ackn
M5
軸移動
初期設定
M5 X10 ;
出力動作0010 0001
「ブロック切り替え」の章
t



9.3 ユーザー定義補助機能

ユーザー定義補助機能には、次の 2 種類の用途があります。

- 事前定義補助機能の拡張
- ユーザー固有の補助機能

事前定義補助機能の拡張

事前定義補助機能用のマシンデータのセットは 1 つのみであるため、事前定義補助機能を使用しても、アドレス指定が可能なチャネルの主軸は 1 つのみです。さらに主軸のアドレス指定を行うには、ユーザー定義補助機能のパラメータを設定して、事前定義補助機能を拡張してください。

事前定義補助機能の拡張は、「アドレス拡張」パラメータを参照します。補助機能が参照する主軸の番号は、「アドレス拡張」パラメータに入力します。

次の各システムファンクションの場合、対応する事前定義補助機能を拡張できます。

システムファンクション	タイプ		
		アドレス拡張 [1]	
			値
工具交換	M	1	6
主軸右回転	M	1	3
主軸左回転	M	1	4
主軸停止	M	1	5
主軸位置決め	M	1	19
軸モード	M	1	70
自動ギヤ選択	M	1	40
ギヤ選択 1	M	1	41
ギヤ選択 2	M	1	42
ギヤ選択 3	M	1	43
ギヤ選択 4	M	1	44
ギヤ選択 5	M	1	45
主軸速度	S	1	-1
工具選択	T	1	-1

[1] アドレス拡張= 1 は、マシンデータで事前定義された補助機能で使用される初期値です。

例:

チャネルの 2 番目および 3 番目の主軸に対応したシステムファンクション「主軸右回転」の事前定義補助機能の拡張

チャネルの 2 番目の主軸に対応した補助機能「主軸右回転」:

MD22010 $MC_AUXFU_ASSIGN_TYPE[n] = "M"

MD22020 $MC_AUXFU_ASSIGN_EXTENSION[n] = 2

MD22030 $MC_AUXFU_ ASSIGN_VALUE[n] = 3

チャネルの 3 番目の主軸に対応した補助機能「主軸右回転」:

MD22010 $MC_AUXFU_ ASSIGN_TYPE[m] = "M"

MD22020 $MC_AUXFU_ASSIGN_EXTENSION[m] = 3
MD22030 $MC_AUXFU_ ASSIGN_VALUE[m] = 3

ユーザー固有の補助機能

ユーザー固有の補助機能には次の特徴があります。

- ユーザー固有の補助機能は、ユーザーファンクションだけを起動します。
- ユーザー固有の補助機能によりシステムファンクションを起動することはできません。
- ユーザー固有の補助機能は、パラメータ設定された出力動作に応じて PLC に出力されます。
- ユーザー固有の補助機能の動作は、PLC ユーザープログラム内に工作機械メーカ/ユーザーにより実装されます。

9.3.1 パラメータ設定

9.3.1.1 ユーザー定義補助機能の最大数

チャネルごとのユーザー定義補助機能の最大数は、次のマシンデータを使用してパラメータ設定できます。

MD11100 $MN_AUXFU_MAXNUM_GROUP_ASSIGN (ユーザー定義補助機能の最大数)

9.3.1.2 グループの割り当て

ブロックサーチの場合の補助機能の取り扱いは、補助機能のグループ割り当てによって定義されます。 使用可能な 168 個の補助機能グループが、事前定義グループおよびユーザー定義グループに分割されます。

事前定義グループ:	1 ... 4	10 ... 12	72 ... 168
ユーザー定義グループ:	5 ... 9	13 ... 71	

標準として、すべてのユーザー定義補助機能が 1 番目の補助機能グループに割り当てられます。 割り当ては、以下のマシンデータを使用して変更することができます。

MD22000 $MC_AUXFU_ASSIGN_GROUP[<n>] (ユーザー定義補助機能のグループ割り当て)

補助機能がどのグループにも割り当てられていない場合、値「0」をマシンデータに入力してください。

注記

1. 補助機能グループとブロックサーチ

ブロックサーチの場合、1 番目の補助機能グループの補助機能だけが収集されますが、出力はされません。

9.3.1.3 タイプ、アドレス拡張、値

補助機能は、タイプ、アドレス拡張、および値の各パラメータでプログラム指令します(「補助機能のプログラミング (ページ 520)」の章を参照してください)。

タイプ

補助機能の名称は、「タイプ」を使用して定義されます。

ユーザー定義補助機能の種類には、次のものがあります。

タイプ	識別子	意味
"H"	補助機能	ユーザー専用の補助機能
"M"	特殊機能	事前定義補助機能の拡張
"S"	主軸機能	
"T"	工具番号	

この設定は、次のマシンデータを使用しておこないます。

MD22010 $MC_AUXFU_ASSIGN_TYPE[<n>] (ユーザー定義補助機能のタイプ)

アドレス拡張

MD22020 $MC_AUXFU_ASSIGN_EXTENSION[<n>] (ユーザー定義補助機能のアドレス拡張)

ユーザー専用の補助機能では、アドレス拡張の機能は定義されていません。 通常、同じ「値」を持つ補助機能を区別するのにアドレス拡張が使用されます。

補助機能のグループ化

同じタイプと値の補助機能をすべて同じ補助機能グループに割り当てる場合は、「アドレス拡張」パラメータに値「-1」を入力してください。

例:

値「=8」が指定されたユーザー専用の補助機能はすべて、10 番目の補助機能グループに割り当てられます。

MD22000 $MC_AUXFU_ASSIGN_GROUP[1]	= 10
MD22010 $MC_AUXFU_ ASSIGN_TYPE[1]	= "H"
MD22020 $MC_AUXFU_ ASSIGN_EXTENSION[1]	= -1
MD22030 $MC_AUXFU_ ASSIGN_VALUE[1]	= 8

値

MD22030 $MC_AUXFU_ASSIGN_VALUE[<n>] (ユーザー定義補助機能の値)

ユーザー専用の補助機能では、「値」パラメータの機能は定義されていません。 この値は、通常、対応する PLC ユーザーファンクションを起動するのに使用されます。

補助機能のグループ化

同じタイプとアドレス拡張のすべての補助機能を同じ補助機能グループに割り当てる場合、「値」パラメータに値「-1」を入力してください。

例:

アドレス拡張「=2」が指定されたユーザー専用の補助機能はすべて、11 番目の補助機能グループに割り当てられます。

MD22000 $MC_AUXFU_ASSIGN_GROUP[2]	= 11
MD22010 $MC_AUXFU_ ASSIGN_TYPE[2]	= "H"
MD22020 $MC_AUXFU_ ASSIGN_EXTENSION[2]	= 2
MD22030 $MC_AUXFU_ ASSIGN_VALUE[2]	= -1

9.3.1.4 出力動作

ユーザー定義補助機能の「出力動作」は、次のマシンデータを使用してパラメータ設定できます。

MD22035 $MC_AUXFU_ASSIGN_SPEC[<n>] (ユーザー定義補助機能の出力動作)

個々の出力パラメータの説明については、事前定義補助機能の「出力動作 (ページ 506)」セクションを参照してください。 そこに記載されている情報は、ユーザー定義補助機能の出力動作に同じように適用できます。

9.4 関連補助機能

機能

関連するヘルプ機能は、対応する事前定義されたヘルプ機能と同じ特性のユーザー定義されたヘルプ機能です。次の事前定義補助機能に対して、ユーザー定義補助機能を関連付けることができます。

- M0 (プログラムストップ)
- M1 (オプショナルストップ)

G グループ

関連するヘルプ機能は、対応する事前定義されたヘルプ機能の G グループに割り当てられます。

パラメータ設定

前述の事前定義補助機能の 1 つとユーザー定義補助機能の関連付けは、次のマシンデータで設定します。

- MD22254 $MC_AUXFU_ASSOC_M0_VALUE ("プログラムストップ"に関連する M 機能)
- MD22256 $MC_AUXFU_ASSOC_M1_VALUE ("オプショナルストップ"に関連する M 機能)

選択

"関連するヘルプ機能"(M-1)の選択は、SINUMERIK Operate ユーザーインターフェースの操作エリア[自動|プログラム制御]で、HMI/PLC インターフェース信号 DB21、... DBX24.4 を設定することで行います。

FB1 パラメータ `MMCToIf` の値に応じて、インターフェース信号が基本 PLC プログラムから NC/PLC インターフェース信号 DB21、... DBX30.5 に送信されます:

- "TRUE":伝送
- "FALSE":伝送なし

初期設定では、パラメータ値は"TRUE"です。

注記

ユーザーインターフェース SINUMERIK Operate による選択オプション

マシンデータでパラメータ設定されている場合、関連するヘルプ機能の選択肢は、[自動|プログラム制御]操作エリアにのみ表示されます。

用途

関連補助機能は、次の場所で使用できます。

- メインプログラム
- サブプログラム
- サイクル

注記

関連補助機能は、シンクロナイズドアクションでは使用**できません**。

NC/PLC インタフェース信号

関連ユーザー定義補助機能の場合は、対応する事前定義補助機能の場合と同じ信号が NC/PLC インタフェースに出力されます。実際にプログラムされている補助機能を区別するために、ユーザー定義補助機能の値(「値」パラメータ)が、補助機能の値として出力されます。つまり、PLC ユーザープログラムで、事前定義補助機能とユーザー定義補助機能を区別できます。

関連するヘルプ機能の NC/PLC インターフェース信号:

- DB21、... DBX24.4 (選択された関連するヘルプ機能)
- DB21、... DBX30.5 (関連するヘルプ機能の有効化)
- DB21、... DBX318.5 (関連するヘルプ機能が有効)

境界条件

次の境界条件に注意してください。

- 1 つのユーザー定義補助機能を複数回関連付けることはできません。
- 事前定義補助機能(M3、M4、M5 など)は関連付けることはできません。

例

1. ユーザー定義補助機能 M111 を M0 に関連付ける方法は次のとおりです。
   MD22254 $MC_AUXFU_ASSOC_M0_VALUE = 111
   これにより、ユーザー定義補助機能 M111 は、M0 と同じ機能になります。
2. ユーザー定義補助機能 M222 を M1 に関連付ける方法は次のとおりです。
   MD22256 $MC_AUXFU_ASSOC_M1_VALUE = 222
   これにより、ユーザー定義補助機能 M222 は、M1 と同じ機能になります。

9.5 タイプ別の出力動作

機能

パートプログラムブロックでプログラムされた移動動作に関連する補助機能の出力動作は、タイプ別に定義できます。

パラメータ設定

タイプ別の出力動作のパラメータ設定は、次のマシンデータを使用して割り当てます。

MD22200 $MC_AUXFU_**M**_SYNC_TYPE(M 機能の出力タイミング)

MD22210 $MC_AUXFU_**S**_SYNC_TYPE(S 機能の出力タイミング)

MD22220 $MC_AUXFU_**T**_SYNC_TYPE(T 機能の出力タイミング)

MD22230 $MC_AUXFU_**H**_SYNC_TYPE(H 機能の出力タイミング)

MD22240 $MC_AUXFU_**F**_SYNC_TYPE (F 機能の出力タイミング)

MD22250 $MC_AUXFU_**D**_SYNC_TYPE(D 機能の出力タイミング)

MD22252 $MC_AUXFU_**DL**_SYNC_TYPE (DL 機能の出力タイミング)

次の出力動作をパラメータ設定できます。

MD $MC_AUXFU_**xx**_SYNC_TYPE = <値>

値	出力動作
0	移動前の出力
1	移動中の出力
2	ブロック終点での出力
3	PLC への出力なし
4	MD22080 で定義した出力動作に従った出力

さまざまな出力動作の説明については、セクション「出力動作 (ページ 506)」を参照してください。

注記

補助機能の各タイプに対して設定できる出力動作については、「パラメータマニュアル」の「マシンデータ説明書」を参照してください。

例

移動動作のあるパートプログラムブロックにおける、さまざまな出力動作の補助機能の出力

パラメータに設定された出力動作は、以下のとおりです。

MD22200 $MC_AUXFU_M_SYNC_TYPE = 1 ⇒ M 機能:
移動**中の**出力

MD22220 $MC_AUXFU_T_SYNC_TYPE = 0 ⇒ T 機能:
移動**前の**出力

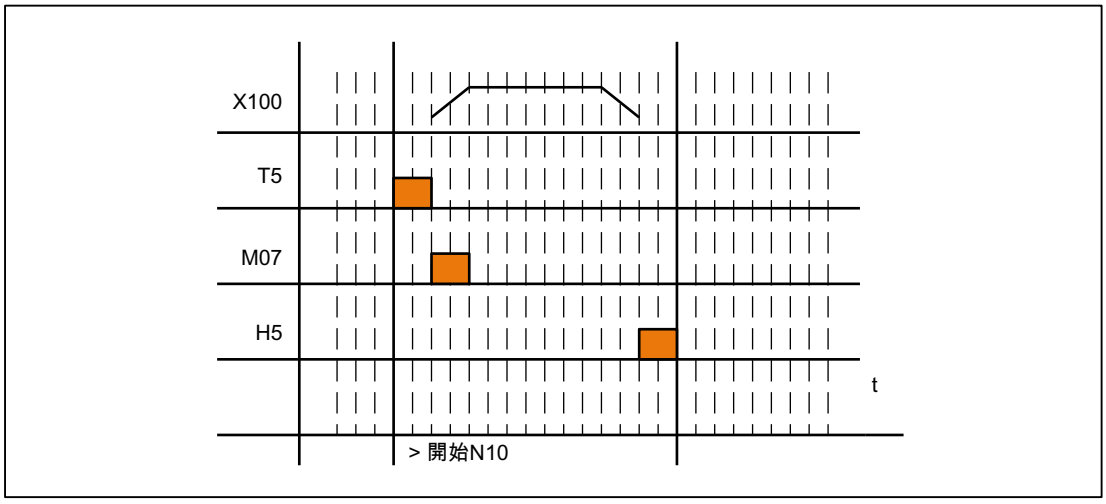
MD22230 $MC_AUXFU_H_SYNC_TYPE = 2 ⇒ H 機能:
ブロック終点での出力

パートプログラムブロック

プログラムコード

```
...
N10 G01 X100 M07 H5 T5
...
```

補助機能出力のタイミング図を以下に示します。



9.6 パラメータに設定された出力動作の優先順位

パラメータ設定された補助機能の出力動作に関連する優先順位は、次の条件に対して別々に考慮してください。

- 出力時間(通常/高速応答)
- 移動に関連する出力(移動前 / 中 / 後)

一般的な規則として、より優先順位の高い出力動作がパラメータ設定されていない場合は、より優先順位の低い、パラメータ設定された出力動作が有効になります。

出力時間

出力時間には次の優先順位が適用されます。

優先順位	出力動作	定義方法
最高	補助機能別	パートプログラム命令: QU(…) (「プログラム可能な出力時間 (ページ 522)」の章を参照してください)
↓	補助機能別	MD22035 $MC_AUXFU_ASSIGN_SYNC[<n>] MD22080 $MC_AUXFU_PREDEF_SYNC[<n>]
↓	グループ別	MD11110 $MC_AUXFU_GROUP_SPEC[<n>]
最低	定義なし	出力動作初期設定: 1 回の OB1 周期の出力時間

移動に関連する出力

移動に関連する出力には次のルールが適用されます。

優先順位	出力動作	定義方法
最高	補助機能別	MD22035 $MC_AUXFU_ASSIGN_SYNC[<n>] MD22080 $MC_AUXFU_PREDEF_SYNC[<n>]
↓	グループ別	MD11110 $MC_AUXFU_GROUP_SPEC[<n>]
↓	タイプ別	MD22200 $MC_AUXFU_M_SYNC_TYPE MD22210 $MC_AUXFU_S_SYNC_TYPE MD22220 $MC_AUXFU_T_SYNC_TYPE MD22230 $MC_AUXFU_H_SYNC_TYPE MD22240 $MC_AUXFU_F_SYNC_TYPE MD22250 $MC_AUXFU_D_SYNC_TYPE MD22252 $MC_AUXFU_DL_SYNC_TYPE
最低	定義なし	出力動作初期設定: ブロック終点での出力

注記

軌跡動作のないパートプログラムブロック

軌跡動作のないパートプログラムブロックでは(位置決め軸と主軸のあるブロックでも)、補助機能はブロック内で、すぐに出力されます。

9.7 補助機能のプログラミング

構文

補助機能は、パートプログラムブロックに次の構文でプログラムします。
<タイプ>[<アドレス拡張>=]<値>

注記

アドレス拡張をプログラムしない場合は、アドレス拡張= 0 が自動的に設定されます。
アドレス拡張= 0 の事前定義補助機能は常に、チャネルのメイン主軸を参照します。

シンボルアドレス指定

「アドレス拡張」および「値」パラメータの値は、シンボルで指定できます。この場合、アドレス拡張のシンボル名は括弧で囲んでください。

例:

第一主軸に対する、補助機能 M3(主軸右回転)のシンボルプログラミング

プログラムコード	コメント
DEF SPINDEL_NR=1	; チャネルの 1 番目の主軸
DEF DREHRICHTUNG=3	; 時計方向の回転
N100 M[SPINDEL_NR]=DREHRICHTUNG	; M1=3M1=3

注記

シンボル名を使用して補助機能をプログラムした場合、そのシンボル名は PLC への補助機能の出力時に転送されません。その代わりに、対応する数値が転送されます。

例

例 1:事前定義補助機能のプログラミング

プログラムコード	コメント
N10 M3	; チャネルのメイン主軸の「主軸右回転」。
N20 M0=3	; チャネルのメイン主軸の「主軸右回転」。
N30 M1=3	; チャネルの 1 番目の主軸の「主軸右回転」。
N40 M2=3	; チャネルの 2 番目の主軸の「主軸右回転」。

例 2:補助機能のプログラミング例と、それに対応する、PLC への出力値

プログラムコード	コメント
DEF Coolant=12	; PLC へ出力: - - -
DEF Lubricant=130	; PLC へ出力: - - -
H[coolant]=lubricant	; PLC へ出力:H12=130
H=coolant	; PLC へ出力:H0=12
H5	; PLC へ出力:H0=5
H=5.379	; PLC へ出力:H0=5.379
H17=3.5	; PLC へ出力:H17=3.5
H[coolant]=13.8	; PLC へ出力:H12=13.8
H='HFF13'	; PLC へ出力:H0=65299
H='B1110'	; PLC へ出力:H0=14
H5.3=21	; 異常

9.8 プログラム可能な出力時間

機能

「OB1 1 サイクルの出力時間(低速応答)」出力動作がパラメータ設定された、ユーザー固有の補助機能は、パートプログラム命令 QU (高速)を使用して、出力ごとに、高速応答の補助機能として定義できます。

構文

高速応答の補助機能は、次の構文でパートプログラムブロックに定義します。
<タイプ>[<アドレス拡張>]=QU(<値>)

例

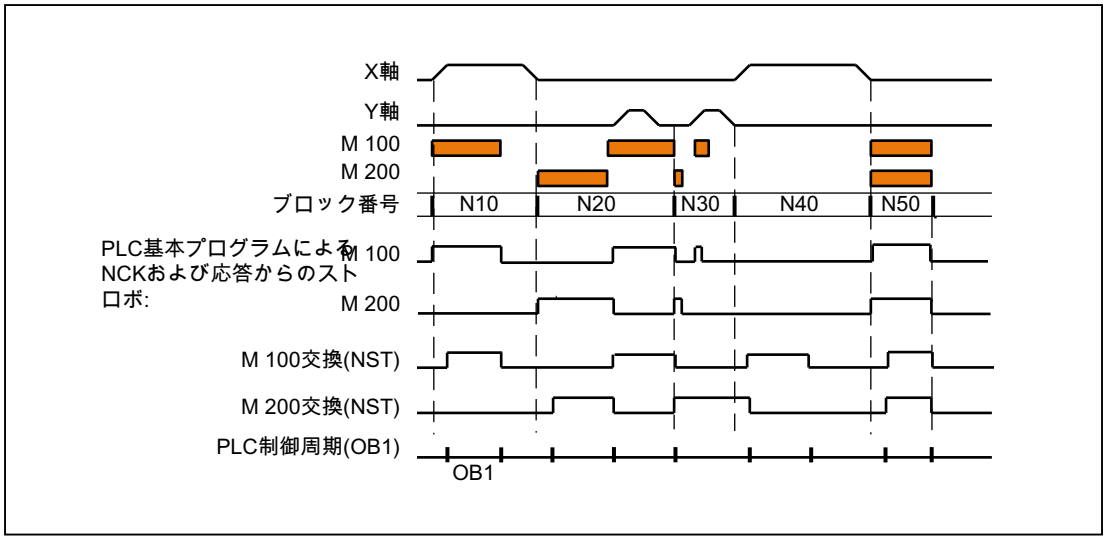
パートプログラム内の M100 および M200 の各補助機能出力のさまざまな動作。補助機能の出力動作は、次のようにパラメータ設定されています。

- M100
  - 1 OB1 サイクルの出力時間(低速応答)
  - 移動中の出力
- M200
  - 1 OB1 サイクルの出力時間(低速応答)
  - 移動前の出力

プログラムコード	コメント
N10 G94 G01 X50 M100	; M100 の出力:移動中
	; 確認応答:低速
N20 Y5 M100 M200	; M200 の出力:移動前
	; M100 の出力:移動中
	; 確認応答:低速
N30 Y0 M=QU(100) M=QU(200)	; M200 の出力:移動前
	; M100 の出力:移動中
	; 確認応答:高速
N40 X0	
N50 M100 M200	; M200 の出力:即時 1)
	; M100 の出力:即時 1)
	; 確認応答:低速
M17	

1) 移動動作がないときは、いつも補助機能は PLC に直ちに出力されます。

次の図に、パートプログラムのタイミング図を示します。パートプログラムブロック N20 および N30 の処理中の時間差に注目してください。



9.9 PLC への補助機能出力

機能

PLC への補助機能出力時には、次の信号と値が NC/PLC インタフェースに伝送されます。

- 変更信号
- 「アドレス拡張」パラメータ
- 「値」パラメータ

NC/PLC インタフェースのデータ領域

補助機能の変更信号と値は、NC/PLC インタフェース内の次のデータ領域にあります。

- 補助機能の変更信号は NC チャネルから伝送されます。
  DB21、... DBB58 ～ DBB67
- 伝送される M 機能と S 機能:
  DB21、... DBB68 ～ DBB112
- 伝送される T、D、および DL 機能:
  DB21、... DBB116 ～ DBB136

- 伝送される H と F 機能:
  DB21、... DBB140 ～ DBB190
- デコードされる M 信号(M0 ～ M99):
  DB21、... DBB194 ～ DBB206 (ダイナミック M 機能)

NC/PLC インタフェースへのアクセス手順については、「P3: SINUMERIK 840D sl 用 PLC 基本プログラム (ページ 1043)」の章を参照してください。

NC/PLC インタフェース内の上記のデータ領域に関する詳しい説明は、以下の資料にあります。

参照先:
『リストマニュアル、リスト、ブック 2』; 「PLC ユーザーインタフェース」、
: 「チャネル信号(DB 21 ～ DB 30)」の章

9.10 ブロック変更遅延のない補助機能

機能

補助機能のに対して、パラメータ設定またはプログラムされている出力動作が、

- 「1 OB40 サイクルの出力時間(高速応答)」
- 「移動前の出力」または「移動中の出力」

のいずれの場合でも、連続軌跡モード(短い移動軌跡かつ高速度)で速度の低下が発生する可能性があります。 これは、PLC による補助機能の確認応答があるまで、システムがブロック終点に対して待機する必要があるからです。 この速度低下を回避するために、それらの補助機能が確認されたかどうかにかかわらず、ブロック変更を実行できます。

パラメータ設定

高速補助機能によるブロック変更遅延マスクは、以下のマシンデータで設定します。

MD22100 $MC_AUXFU_QUICK_BLOCKCHANGE(高速補助機能によるブロック変更遅延)

値	意味
0	高速補助機能が PLC に出力される場合、PLC (OB40)による確認応答があるまでブロック変更が遅延されます。
1	高速補助機能が PLC に出力される場合、ブロック変更が遅延されません。

境界条件

ブロック変更遅延なしで出力される補助機能と、プログラムされたパートプログラムブロックとの同期が保証されなくなります。 最悪の場合、次のパートプログラムブロックに変更されてから、1 OB40 サイクル後に確認応答が行われ、1 OB1 サイクル後に補助機能が実行されます。

9.11 自動先読み停止を伴う M 機能

機能

補助機能と組み合わせた先読み停止の起動は、STOPRE パートプログラム命令でプログラムできます。M 機能のプログラミングで先読み停止を常に起動する場合は、次のマシンデータで、M 機能ごとにパラメータ設定できます。

MD10713 $MN_M_NO_FCT_STOPRE[<n>] (先読み停止を伴う M 機能)

例

ユーザー定義の M 機能 M88 は、先読み停止を起動するものとします。

パラメータ設定:

MD10713 $MN_M_NO_FCT_STOPRE [0] = 88

用途:

パートプログラム(抜粋)

プログラムコード	コメント
...	
N100 G0 X10 M88	; 移動動作と M88 による自動先読み停止。
N110 Y=R1	; N110 は、移動動作が完了して M 機能が確認された後にのみ解析されます。
...	

必要条件

サブプログラムでサブプログラムが M 機能を介して次のいずれかの方法で間接的に呼び出される場合、先読み停止は実行されません。

- MD10715 $MN_M_NO_FCT_CYCLE (サブプログラムに置換される M 機能)
- M98 (ISO 言語 T / ISO 言語 M)

9.12 オーバーストアへの応答

オーバストア

SINUMERIK 操作画面で、

- パートプログラムの NC START
- 中断したパートプログラムを再開するための NC START

の各機能を起動する前に、起動時に出力される補助機能を「オーバストア」機能によって変更できます。

これは、次のような用途に使用できます。

- ブロック検索後の補助機能の追加
- あるパートプログラム実行のための初期状態のリストア

オーバストアできる補助機能のタイプ

次のタイプの補助機能をオーバストアできます。

- M (補助機能)
- S (主軸速度)
- T (工具番号)
- H (補助機能)
- D (工具オフセット番号)
- DL (追加工具オフセット)
- F (送り速度)

有効期間

M3(主軸右回転)などのオーバストアされた補助機能は、同じ補助機能グループの別の補助機能、追加オーバストア、またはパートプログラムブロックでのプログラミングによって上書きされるまで有効です。

9.13 ブロック検索中の動作

9.13.1 タイプ 1、2、および 4 のブロック検索中の補助機能出力

出力動作

出力動作タイプ 1、2、および 4 のブロック検索の場合は、補助機能がグループ別に収集されます。 各補助機能グループの最後の補助機能が、実際の再エントリブロックの前の NC START の後に別のパートプログラムブロックで出力され、以下の出力動作をおこないます。

- 1 回の OB1 周期の出力時間(通常応答)
- 移動前の出力

出力制御

補助機能がブロック検索後に PLC に出力されるかどうかは、下記のマシンデータのビット 8 で設定できます。

- MD22080 $MC_AUXFU_PREDEF_SPEC[<n>]
  (事前定義補助機能の出力動作)
  この場合、**<n> =** システムファンクションのインデックス(0 ... 32)
- MD22035 $MC_AUXFU_ASSIGN_SPEC[<n>]
  (ユーザー定義補助機能の出力動作)
  この場合、**<n> =** 補助機能のインデックス(0 ... 254)
- MD11110 $MN_AUXFU_GROUP_SPEC[<n>]
  (グループ内の補助機能の出力動作)
  この場合、**<n> =** グループインデックス(0 ... 63)

ビット	値	意味
10	0	タイプ 1、2、および 4 のブロック検索中の出力
	1	タイプ 1、2、および 4 のブロック検索中の出力なし

この動作は表示には影響せず、変数$AC_AUXFU_STATE[<n>]、$AC_AUXFU_VALUE[<n>]、および$AC_AUXFU_EXT[<n>]のいずれにも影響を及ぼしません。

各補助機能は、PLC に出力されなくても、常にブロック検索後に収集されたとみなされます。

収集時には、ビット 8 が設定されていない補助機能も、ブロック検索後に出力されない補助機能で上書きされます。

ユーザーはブロック検索後に収集済みの補助機能を確認でき、特定の状況下ではサブプログラムまたはシンクロナイズドアクションによって再度それらを出力できます。

注記

以下の補助機能は収集されません。

- 補助機能グループに割り当てられていない補助機能。
- 1 番目の補助機能グループに割り当てられた補助機能。

補助機能のオーバストア

ブロック検索が完了した後、収集された補助機能が次の NC-START 時に出力されます。 追加の補助機能を出力する必要がある場合は、それらを「オーバストア」機能で追加できます(「オーバーストアへの応答 (ページ 526)」の章を参照してください)。

M19 動作(主軸位置決め)

ブロック検索後には、M19 を含むパートプログラムとターゲットブロックの間に主軸用の他の補助機能がプログラム指令されている場合でも、M19 でプログラム指令された最後の主軸位置決め命令が常に実行されます。 したがって、必要な主軸イネーブルは、PLC ユーザープログラムで移動指令のインタフェース信号から入手してください。

DB31、... DBX64.6/64.7 (マイナス/プラス移動指令)

この場合、主軸用補助機能 M3、M4、および M5 は、主軸位置決めの後まで PLC に出力されない可能性があるため、適切ではありません。

ブロック検索については、「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 (ページ 569)」の章を参照してください。

9.13.2 複数のグループへの補助機能の割り当て

機能

ユーザー定義補助機能は、グループ割り当て(MD22000 $MC_AUXFU_ASSIGN_GROUP)を使用して、複数のグループに割り当てることもできます。 ブロック検索時に、設定したすべてのグループに対して、これらの補助機能が収集されます。

注記

事前定義補助機能は、1 グループにのみ割り当てることができます。

例

DIN には、クーラント出力用の以下の M 命令が含まれます。

- M7:クーラント 2 のオン
- M8:クーラント 1 のオン
- M9: クーラント 1 と 2 のオフ

したがって、以下の場合は、両方のクーラントを同時にオンにできます。

- M7 と M8 が 2 つの別のグループ(グループ 5 と 6 など)で収集される
- 以下の例のように、M9 を 2 つのグループに割り当てる必要がある
  - グループ 5:M7、M9
  - グループ 6: M8、M9

パラメータ設定:

MD11100 $MN_AUXFU_MAXNUM_GROUP_ASSIGN = 4

MD22000 $MC_AUXFU_ASSIGN_GROUP[0] = 5

MD22000 $MC_AUXFU_ASSIGN_GROUP[1] = 5

MD22000 $MC_AUXFU_ASSIGN_GROUP[2] = 6

MD22000 $MC_AUXFU_ASSIGN_GROUP[3] = 6

MD22010 $MC_AUXFU_ASSIGN_TYPE [0] = M

MD22010 $MC_AUXFU_ASSIGN_TYPE [1] = M

MD22010 $MC_AUXFU_ASSIGN_TYPE [2] = M

MD22010 $MC_AUXFU_ASSIGN_TYPE [3] = M

MD22020 $MC_AUXFU_ASSIGN_EXTENSION[0] = 0

MD22020 $MC_AUXFU_ASSIGN_EXTENSION[1] = 0

MD22020 $MC_AUXFU_ASSIGN_EXTENSION[2] = 0

MD22020 $MC_AUXFU_ASSIGN_EXTENSION[3] = 0

MD22030 $MC_AUXFU_ASSIGN_VALUE[0] = 7

MD22030 $MC_AUXFU_ASSIGN_VALUE[1] = 9

MD22030 $MC_AUXFU_ASSIGN_VALUE[2] = 8

MD22030 $MC_AUXFU_ASSIGN_VALUE[3] = 9

MD22035 $MC_AUXFU_ASSIGN_SPEC [0] = 'H121'

MD22035 $MC_AUXFU_ASSIGN_SPEC [1] = 'H121'

MD22035 $MC_AUXFU_ASSIGN_SPEC [2] = 'H121'

MD22035 $MC_AUXFU_ASSIGN_SPEC [3] = 'H121'

パートプログラム(抜粋):

プログラムコード

```
...
N10 ... M8
N20 ... M9
N30 ... M7
...
```

ブロック検索時に、補助機能 M9 はタイプ 5 と 6 に収集されます。

収集した M 補助機能の確認:

5 番目のグループの M 機能: $AC_AUXFU_M_VALUE [4] = 7

6 番目のグループの M 機能: $AC_AUXFU_M_VALUE [5] = 9

9.13.3 有効な M 補助機能のタイムスタンプ

収集した補助機能をブロック検索後に出力するときは、収集したときの順番に注意してください。 このために、以下のシステム変数でグループ別に確認可能なタイムスタンプが、各グループに割り当てられます。

$AC_AUXFU_M_TICK[<n>] (有効な M 補助機能のタイムスタンプ)

9.13.4 出力順序の決定

機能

M 補助機能の出力順序の決定をプログラマが簡単に処理できるよう、以下の事前定義処理が用意されています。

```
AUXFUMSEQ(VAR INT _NUM_IN, VAR INT _M_IN[], VAR INT _EXT_IN[],
VAR INT _NUM_OUT, VAR INT _M_OUT[], VAR INT _EXT_OUT[])
```

入力パラメータ:

`VAR INT _NUM_IN:`	関連する M 命令グループの数
`VAR INT _M_IN[]:`	関連する M コードのフィールド
`VAR INT _EXT_IN[]:`	関連する M アドレス拡張のフィールド

出力パラメータ:

`VAR INT _NUM_OUT:`	決定した M コードの数
`VAR INT _M_OUT[]:`	決定した M コードのフィールド
`VAR INT _EXT_OUT[]:`	決定した M アドレス拡張のフィールド

この機能は、グループ別に収集した M 補助機能を、事前定義の M コードに対して出力する順序を決定します。 この順序は、収集時間$AC_AUXFU_M_TICK[<n>]によって決まります(「有効な M 補助機能のタイムスタンプ (ページ 530)」の章を参照してください)。

特定の M コードは、複数のグループに属する場合でも、1 度しか考慮されません。 関連する M 命令の数が 0 以下の場合は、収集した M コードがすべて出力されます。 関連する M 命令の数は 64 までに制限されています。

例

クーラント出力用の M 命令:

- M7:クーラント 2 のオン
- M8:クーラント 1 のオン
- M9: クーラント 1 と 2 のオフ

グループの割り当て:

- グループ 5:M7、M9
- グループ 6: M8、M9

パートプログラム(抜粋):

プログラムコード

```
...
N10 ... M8
N20 ... M9
N30 ... M7
...
```

ブロック検索時に、これらの補助機能がグループ別に収集されます。 補助機能グループの最後の補助機能は、次のように、ブロック検索に続いて PLC に出力されます。

- グループ 5: M7
- グループ 6: M9

これらが M7、M9 の順序で出力された場合は、クーラントは起動しません。 しかし、クーラント 2 は、プログラムの実行時に起動します。 したがって、M 補助機能の正しい出力順序は、事前定義された処理 AUXFUMSEQ(...) を含む ASUB で、次のように決定されます。

プログラムコード

```
DEF INT _I, _M_IN[3], _EXT_IN[3], _NUM_OUT, _M_OUT[2], _EXT_OUT[2]
_M_IN[0]=7 _EXT_IN[0]=0
_M_IN[1]=8 _EXT_IN[1]=0
_M_IN[2]=9 _EXT_IN[2]=0
AUXFUMSEQ(3,_M_IN,_EXT_IN,_NUM_OUT,_M_OUT,_EXT_OUT)
FOR _I = 0 TO _NUM_OUT-1
   M[_EXT_OUT[_I]]=_M_OUT[_I]
ENDFOR
```

9.13.5 主軸用補助機能の出力抑制

機能

たとえば工具交換とともに、アクションブロックでブロック検索時に収集された主軸用補助機能を出力するのではなく、工具交換後などまで出力を遅延させる必要がある場合があります。このために、ブロック検索後の主軸用補助機能の自動出力をマスクすることができます。後に、オーバストアまたは ASUP によって手動で出力を行うことができます。

パラメータ設定

ブロック検索後の主軸用補助機能の自動出力のマスクは、次のマシンデータで設定できます。

MD11450 $MN_SEARCH_RUN_MODE (ブロック検索後の動作)

ビット	値	意味
2	0	主軸用補助機能の出力はアクションブロックでおこなわれます。
	1	アクションブロックで補助機能の出力がマスクされます。

システム変数

主軸用補助機能は常に、上記のパラメータ設定とは関係なく、ブロック検索時に以下のシステム変数に保存されます。

システム変数	説明	
$P_SEARCH_S [<n>]	収集された主軸速度	
	値の範囲:	0 ... Smax
$P_SEARCH_SDIR [<n>]	収集された主軸回転方向	
	値の範囲:	3, 4, 5, -5, -19, 70
$P_SEARCH_SGEAR [<n>]	収集された主軸ギヤ選択 M 機能	
	値の範囲:	40 ... 45
$P_SEARCH_SPOS [<n>]	収集された主軸位置	
	値の範囲:	0 ... MD30330 $MA_MODULO_RANGE (モジュロ範囲のサイズ)
	または	
	収集された移動軌跡	
	値の範囲:	-100,000,000 ... 100,000,000
$P_SEARCH_SPOSMODE [<n>]	蓄積された位置アプローチモード	
	値の範囲:	0 ... 5

後で主軸用補助機能を出力する場合は、以下のように、ASUP などでシステム変数を読み取り、アクションブロック出力の後に出力できます。

DB21、... DBX32.6 = 1(最後のアクションブロック有効)

注記

システム変数$P_S、$P_DIR と$P_SGEAR の内容は、同期処理によりブロック検索後に失われることがあります。

ASUP、ブロック検索、およびアクションブロックについては、「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 (ページ 569)」の章を参照してください。

例

主軸用補助機能の出力がマスクされた輪郭のブロック検索と、アクションブロック出力後の ASUP 起動:

パラメータ設定:MD11450 $MN_SEARCH_RUN_MODE、ビット 2 = 1

N55 でのブロック検索の後、ASUP が起動します。

パートプログラム:

プログラムコード	コメント
N05 M3 S200	; 主軸 1:
N10 G4 F3	
N15 SPOS=111	; 主軸 1 は ASUP で 111 度に位置決めされます
N20 M2=4 S2=300	; 主軸 2:
N25 G4 F3	
N30 SPOS[2] = IC(77)	; 主軸 2 は、インクリメンタルに 77°移動します。
N55 X10 G0	; ターゲットブロック
N60 G4 F10	
N99 M30	

ASUB:

プログラムコード	コメント
PROC ASUP_SAVE	
MSG (「主軸機能の出力□)	
DEF INT SNR=1	
AUSG_SPI:	
M[SNR]=$P_SEARCH_SGEAR[SNR]	; ギヤ選択の出力
S[SNR]=$P_SEARCH_S[SNR]	; 出力速度 (M40 の場合、適切なギヤ選択が決定されます)。
M[SNR]=$P_SEARCH_SDIR[SNR]	; 回転方向、位置決めまたは軸モードの出力。
SNR=SNR+1	; 次の主軸。
REPEAT AUSG_SPI P=$P_NUM_SPINDLES-1	; すべての主軸に適用。
MSG("")	

プログラムコード	コメント
REPOSA	
RET	

例の説明

主軸の数が判明している場合、同じタイプの出力を 1 つのパートプログラムブロックに書き込んで、プログラムの実行時間を短縮できます。

ギヤ変更に関連して主軸の位置決めまたは軸モードへの切り替えがおこなわれるとアラームが発生する可能性があるため、$P_SEARCH_SDIR の出力は別のパートプログラムブロックでおこないます。

起動した ASUP が REPOSA で終了した場合、主軸 2 が 77°の位置に再位置決めされる間、主軸 1 は 111°の位置にとどまります。

別の応答が必要な場合は、ブロック検索のプログラム順序、たとえば「N05 M3 S...」と「N30 SPOS[2] = IC(...)」に特殊な処理が必要です。

ブロック検索が有効かどうかは、ASUP でシステム変数$P_SEARCH を介して確認できます。
$P_SEARCH==1 ; ブロック検索有効

速度制御運転後にインクリメンタルな位置決めをおこなう場合は、移動する軌跡が定義されますが、最終到達位置が位置決め中に初めて判明することがあります。たとえば、位置制御をオンにした場合の原点マーク通過時の位置合わせがこれに当たります。そのため、原点位置よりも後ろのプログラム指令距離は、REPOS 位置(ASUP での REPOSA)として反映されます。

必要条件

収集される S 値

パートプログラムでの S 値の意味は、下記の現在有効な送り速度の種類に依存します。

G93、G94、G95、G97、G971:	S 値は速度として解釈されます。
G96、G961:	S 値は定切削速度として解釈されます。

システム変数$P_SEARCH_S の出力前に送り動作が(工具交換などで)変更された場合は、不適切な送り速度の種類が使用されないように、パートプログラムにおけるターゲットブロックの元の設定を復元してください。

収集される回転方向

回転方向の出力について、ブロック検索の開始時にシステム変数$P_SEARCH_SDIR に初期値「-5」が割り当てられます。この値は、出力に影響しません。

これにより、回転方向、位置決めまたは軸モードとともに主軸がプログラム指令されていないプログラム区間全体にわたるブロック検索について、前回の主軸運転モードが保持されます。

M19、SPOS、および SPOSA のプログラミングは、M3、M4、M5、および M70 の代わりに、「M-19」(内部の M19)としてシステム変数$P_SEARCH_SDIR に収集されます。

「M-19」を出力する場合、位置決めデータが、システム変数$P_SEARCH_SPOS と $P_SEARCH_SPOSMODE から内部で読み取られます。どちらのシステム変数も、修正などのために書き込むことができます。

注記

上記の割り当て(M[<n>] = $P_SEARCH_SDIR[<n>]など)のため、「–5」と「19」という値は一般にユーザーに対して非表示のままとなり、ASUB でシステム変数の特殊な評価をおこなう場合にのみ参照する必要があります。

9.13.6 タイプ 5 のブロック検索(SERUPRO)による補助機能の出力

出力動作

タイプ 5 のブロック検索(SERUPRO)の場合は、補助機能をブロック検索時に PLC に出力したり、グループ別に以下のシステム変数に収集したりできます。

- $AC_AUXFU_PREDEF_INDEX[<n>] (事前定義補助機能のインデックス)
- $AC_AUXFU_TYPE[<n>] (補助機能のタイプ)
- $AC_AUXFU_STATE[<n>] (補助機能の出力状態)
- $AC_AUXFU_EXT[<n>] (補助機能のアドレス拡張)
- $AC_AUXFU_VALUE[<n>] (補助機能の値)

これらのシステム変数の説明については、「システム変数の確認 (ページ 552)」の章を参照してください。

出力制御

タイプ 5 のブロック検索(SERUPRO)時に補助機能を PLC に出力するかどうか、およびグループ別にシステム変数に収集するかどうかは、以下のマシンデータのビット 9 と 10 で設定できます。

- MD22080 $MC_AUXFU_PREDEF_SPEC[<n>]
  (事前定義補助機能の出力動作)
  この場合、<n> =システムファンクションのインデックス(0 ... 32)
- MD22035 $MC_AUXFU_ASSIGN_SPEC[<n>]
  (ユーザー定義補助機能の出力動作)
  この場合、<n> =補助機能のインデックス(0 ... 254)
- MD11110 $MN_AUXFU_GROUP_SPEC[<n>]
  (グループ内の補助機能の出力動作)
  この場合、<n> =グループインデックス(0 ... 63)

ビット	値	意味
9	0	タイプ 5 のブロック検索(SERUPRO)中の収集なし
	1	タイプ 5 のブロック検索(SERUPRO)中の収集
10	0	タイプ 5 のブロック検索(SERUPRO)中に出力
	1	タイプ 5 のブロック検索(SERUPRO)中の出力なし

出力カウンタ

ユーザーは、ブロック検索 ASUP のなかで、チャネルごとに、収集した補助機能を PLC に出力できます。複数チャネルでの連続出力をおこなう場合は、補助機能を出力するたびに、すべてのチャネルにわたり、以下の 3 つの出力カウンタが変更されます。

システム変数	意味		
$AC_AUXFU_TICK[<n>,<m>]	有効な補助機能の出力カウンタ		
	インデックス	意味	
	<n>	グループインデックス (0 … 63)	
	<m>	出力カウンタ(0 ～ 2)	
		値	意味
		0	出力順序カウンタ(IPO サイクル内のすべての出力)
		1	IPO サイクル内の一つの出力順序の中のパッケージカウンタ
		2	1 パッケージ内の補助機能カウンタ

説明

- 補助機能パッケージには、最大 10 個までの補助機能が含まれます。
- SERUPRO 時に、各チャネルの IPO サイクル毎に 2 個のパッケージを処理できます。これは、この周期でシンクロナイズドアクションを処理するためです。
- 1 回の IPO サイクル内で 20 パッケージ(チャネル毎に 2 パッケージ * 10 チャネル)までの出力順序を、すべてのチャネルにわたって処理できます。

この符号化方式により、同じ IPO サイクルで、いくつの補助機能パッケージが処理され、そのなかで、いくつの補助機能が処理されたかが示されます。

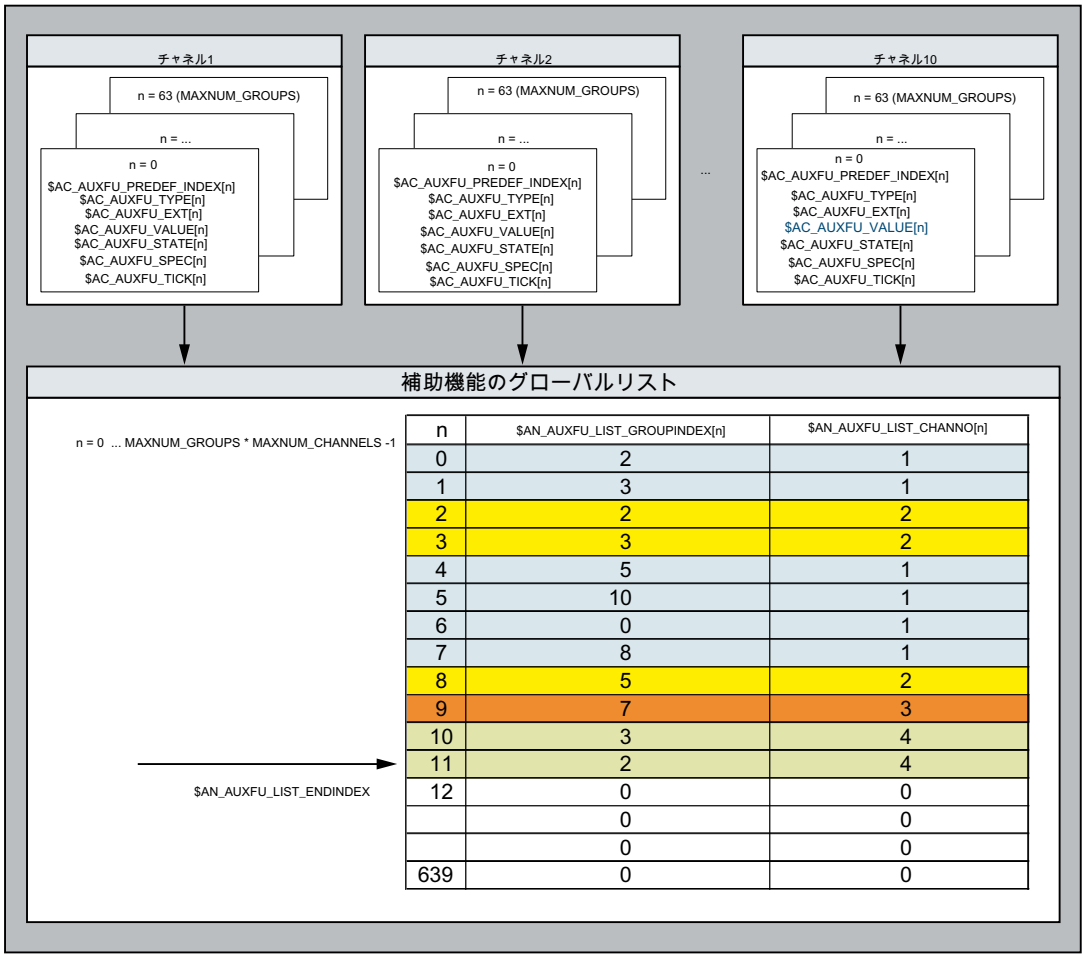
- 1 回の IPO サイクルで収集された補助機能で、順序カウンタが同じになります。
- 1 パッケージ(ブロックまたはシンクロナイズドアクション)で収集されたすべての補助機能で、パッケージカウンタが同じになります。

補助機能カウンタの総数は、補助機能が収集されるたびに加算されます。

補助機能のグローバルリスト

SERUPRO の終わりに、個々のチャネルでグループ別に収集された補助機能は、そのカウンタ状態に従って、チャネル番号とグループインデックスとともに、全チャネルにわたるリストに入力されます。

システム変数 1)	意味
$AC_AUXFU_TICK[<n>,<m>]	カウンタの値
$AN_AUXFU_LIST_CHANNO[<n>] 2)	チャネル番号
$AN_AUXFU_LIST_GROUPINDEX[<n>] 2)	グループインデックス
1) インデックス<n>の数値の範囲:0 ... MAXNUM_GROUPS * MAXNUM_CHANNELS - 1 2) システム変数は、読み取りと書き込みが可能です。	



n	$AN_AUXFU_LIST_GROUPINDEX[n]	$AN_AUXFU_LIST_CHANNO[n]
0	2	1
1	3	1
2	2	2
3	3	2
4	5	1
5	10	1
6	0	1
7	8	1
8	5	2
9	7	3
10	3	4
11	2	4
12	0	0
	0	0
	0	0
639	0	0

グローバルリストは、検索ターゲットが見つかった順序に基づいて作成されます。これは、SERUPRO 終了時の次の ASUP で補助機能を出力するときの、システムによる提案

として使用されます。補助機能を出力しない場合は、対応するグループインデックスが「0」に設定されます。

主軸補助機能に関する動作

ブロック検索開始後に、すべてのチャネルが補助機能をグループごとに、そのチャネル変数に収集します。収集した補助機能を使用して、SERUPRO ターゲットブロックで主軸状態を広範囲にわたって復元するために、主軸補助機能の各グループで最後に有効な補助機能が、ターゲットブロックで主軸状態を表す必要があります。主軸状態が変化する場合は、使用しなくなった補助機能が削除されるか、必要に応じて、自動補助機能が入力されます。

グローバル補助機能リスト中のすべての主軸補助機能は、ターゲットブロックで得られる主軸状態に対応する必要があります。これは、リストの出力時に補助機能を処理可能にするとともに、パートプログラムの続行を妨げるアラームや想定外の主軸状態が要求されないようにするためです。

これは、システムで設定されているあらゆる主軸の補助機能グループに関係します。この場合、主軸番号は、補助機能のアドレス拡張に対応します。

グループ a: M3、M4、M5、M19、M70

グループ b: M40、M41、M42、M43、M44、M45

グループ c: S

使用しなくなった補助機能の削除

以下の機能では、グループ a の補助機能が、当該の主軸に対して削除されます。

- COUPON、TRAILON、EGON などの一般的な連結の起動時のスレーブ主軸に対して

グループ a からの自動補助機能の起動

以下の機能では、グループ a の補助機能が、当該の主軸に対して自動的に起動されます。

- 主軸同期連結の解除時のスレーブ主軸
  - COUPOF は、連結の状況に応じて、メインランで M3、M4 と S、または M5 を起動します。
  - COUPOF(S<n>, S<m>, POS)および COUPOFS(S<n>, S<m>, POS, POS)は M3、M4 および S を起動します。
  - COUPOFS はメインランで M5 を起動します。
  - COUPOFS(S<n>, S<m>, POS)は、メインランで M19 を起動します。
    自動 M19 (ASUP の「SPOS[<アドレス拡張>] = IC(0)」)は、移動動作なしの位置決めモードを起動します。
- 軸として移動中、または座標変換を選択することによる軸モードへの遷移時に、主軸が軸として動作中に、M70 が起動されます。
- M5 は SPCOF 時に起動されます。

注記

「軸入れ替え」と「軸コンテナ回転」機能を使用する場合は、主軸のプログラミング用補助機能を常に、入れ替え/回転時の実際の(モータの)状態との互換性が保証されるように指定してください。ここでは、軸入れ替えと軸コンテナの仕組みは区別されます。

軸コンテナ回転の例:

軸コンテナには 4 つの主軸が含まれ、それぞれが別のチャネル(1 ～ 4)に割り当てられます。M3 S1000 は常にチャネル 1 でプログラム指令され、その後に軸コンテナ回転が実行されます。その他のチャネルでは、主軸のプログラミングをおこないません。3 番目の軸コンテナ回転と 4 番目の主軸プログラミング M3 に続いて、4 つの主軸がすべて、右回りに速度 1000 rpm で回転します。このとき、SERUPRO の終わりがこの範囲内にある場合は、チャネルの各 ASUP に、ローカルの主軸に対する M3 S1000 が含まれることになります。

ただし、入れ替え中は、収集した補助機能を割り当てることができるのは、主軸が現在存在するチャネルのみです。

複数のチャネルにわたる補助機能

タイプ 5 のブロック検索(SERUPRO)の場合は、補助機能を、グローバル補助機能リストで複数のチャネルにわたって収集できます。このグループから収集された最後の補助機能(最高度数の)のみがグローバルリストに入力されます。

次のマシンデータを使用して当該の設定が行われます。

- MD22080 $MC_AUXFU_PREDEF_SPEC[<n>]、ビット 11
  (事前定義補助機能の出力動作)
  この場合、<n> =システムファンクションのインデックス
- MD22035 $MC_AUXFU_ASSIGN_SPEC[<n>]、ビット 11
  (ユーザー定義補助機能の出力動作)
  この場合、<n> =補助機能のインデックス
- MD11110 $MN_AUXFU_GROUP_SPEC[<n>]、ビット 11
  (グループ内の補助機能の出力動作)
  この場合、<n> =グループインデックス

ビット	値	意味
11	0	チャネル別の収集
	1	複数チャネルにわたる収集

主軸用補助機能はブロック検索終了時には、主軸の状態に応じて、あらかじめフィルタされます。チャネルデータは、それに従って更新されます。グローバル補助機能リストは、SERUPRO 終了時の ASUP で順番に処理でき、ソートされた補助機能をチャネルと同期して出力できます。

収集した最後の補助機能の確認

グローバルリストの、収集した最後の補助機能のインデックスは、システム変数 $AN_AUXFU_LIST_ENDINDEX で確認できます。

9.13.7 SERUPRO 終了時の ASUB

機能

プログラムテストによるブロックサーチ(SERUPRO)の終了後、後続の処理を開始する前に、サーチ中に収集した補助機能を出力する必要があります。このために、ブロックサーチ中に、補助機能をグローバルリストに収集することができます。SERUPRO end ASUP が、このリストから対応するパートプログラムブロックをチャネル別に生成します。これにより、収集された補助機能をチャネル別と複数チャネルにわたっての両方で、正しい順番で出力することができます。完全版の SERUPRO end ASUP は、NC ソフトウェアのコンポーネントです。

ユーザー/工作機械メーカによる SERUPRO end ASUP の変更が可能です。以下に説明する機能は、補助機能のグローバルリストの処理と、同期化された補助機能の出力に必要なパートプログラムブロックの生成をサポートしています。

機能 AUXFUSYNC(...)

機能:

機能 AUXFUSYNC は、呼び出されるたびに、補助機能のグローバルリストから、完全なパートプログラムブロックを文字として生成します。 このパートプログラムブロックには、補助機能または補助機能出力を同期化するための命令(WAITM、G4 など)のどちらかが入っています。

この機能は先読み停止を起動します。

構文:

```
PROC AUXFUSYNC(VAR INT <数>, VAR INT <グループインデックス>[10], VAR
STRING[400] <ASSEMBLED>)
```

パラメータ:

<NUM>: パラメータ**<**ASSEMBLED**>**で提供されるパートプログラムブロックに関する情報か、パートプログラムブロックに含まれている補助機能の情報が入っています。

数値の範囲: -1、0、1 ... 10

値	意味
≥ 1	パートプログラムブロックに含まれている補助機能の数
0	補助機能なしのパートプログラムブロック。例:WAITM、G4
-1	終了識別子補助機能のグローバルリストが、実際のチャネルに対して完全に処理されました。

<グループインデックス>: パートプログラムブロックに含まれている補助機能グループのインデックスが入っています。インデックス = 補助機能グループの番号 - 1

<ASSEMBLED>: チャネル別 SERUPRO end ASUP の完全なパートプログラムブロックが、文字として含まれています。

詳細情報:

補助機能がシンクロナイズドアクションを介して収集された場合、2 つの NC ブロックが生成されます。一方の NC ブロックは、補助機能の出力に使用されます。もう一方は、

補助機能の出力のために、この NC ブロックをメインラインに転送するための実行可能 NC ブロックです。

1. シンクロナイズドアクションを介した補助機能の出力。例:`WHEN TRUE DO M100 M102`
2. 実行可能 NC ブロック。例:`G4 F0.001`

機能 AUXFUDEL(...)

機能:

機能 AUXFUDEL は、指定された補助機能を、呼び出し元チャネル別の補助機能のグローバルリストから削除します。削除は、対応するグループインデックス ...`GROUPINDEX[n]`を 0 に設定することで実現されます。

この機能は、`AUXFUSYNC` を呼び出す前に呼び出してください。

この機能は先読み停止を起動します。

構文:

```
PROC AUXFUDEL(CHAR <タイプ>, INT <拡張>, REAL <値>, INT <グループ>)
```

パラメータ:

<タイプ>:	削除する補助機能のタイプ
<拡張>:	削除する補助機能のアドレス拡張
<値>:	削除する補助機能の値
<グループ>:	補助機能グループの番号

機能 AUXFUDELG(...)

機能:

機能 AUXFUDELG は、指定された補助機能グループのすべての補助機能を、呼び出し元チャネル別の補助機能のグローバルリストから削除します。削除は、対応するグループインデックス ...`GROUPINDEX[n]`を 0 に設定することで実現されます。

この機能は、`AUXFUSYNC` を呼び出す前に呼び出してください。

この機能は先読み停止を起動します。

構文:

```
PROC AUXFUDELG(INT <グループ>)
```

パラメータ:

<グループ>: 補助機能グループの番号

マルチチャネルブロックサーチ

通知
マルチチャネルブロックサーチと AUXFUDEL / AUXFUDELG SERUPRO end ASUP でのマルチチャネルブロックサーチでは、AUXFUSYNC ファンクションを呼び出す前に、AUXFUDEL / AUXFUDELG を使用する補助機能が補助機能のグローバルリストから削除された場合、関連するチャネルを同期化する必要があります。同期化により、AUXFUSYNC を呼び出す前にすべての削除要求が処理され、一貫したリストが提供されます。

例

ユーザー固有の SERUPRO end ASUP の 2 つの設定例

例 1:AUXFUSYNC(...)を使用した補助機能の削除と補助機能出力の生成

プログラムコード	コメント
N10 DEF STRING[400] ASSEMBLED=""	
N20 DEF STRING[31] FILENAME="/_N_CST_DIR/_N_AUXFU_SPF"	
N30 DEF INT GROUPINDEX[10]	
N40 DEF INT NUM	
N60 DEF INT ERROR	
N90	
N140 AUXFUDEL("M",2,3,5)	; M2=3 (5 番目の補助機能グループ) 解除
N150	
N170 AUXFUDELG(6)	; 6 番目のグループの収集された補助機能を
	; 解除します。
N180	
N190 IF ISFILE(FILENAME)	
N210 DELETE(ERROR,FILENAME)	; FILENAME ファイルを削除します
N220 IF (ERROR<>0)	; エラー分析
N230 SETAL(61000+ERROR)	
N240 ENDIF	
N250 ENDIF	

プログラムコード	コメント
; CAUTION!	
; If, for a multi-channel block search, auxiliary functions with AUXFUDEL/AUXFUDELG	
; are deleted from the global list of auxiliary functions, before the loop to	
; generate the subprogram FILENAME with AUXFUSYNC, the channels must be synchronized.	
The synchronization ensures that all delete requests were processed	
; in all channels and a consistent list is available.	
; Example:WAITM(99,1,2,3)	
N270 LOOP	
N300 AUXFUSYNC(NUM,GROUPINDEX,ASSEMBLED)	; パートプログラムブロックを生成します
N310	
N320 IF (NUM==-1)	; チャネルのすべての補助機能が
	; 実行されました。
N340 GOTOF LABEL1	
N350 ENDIF	
N380 WRITE(ERROR,FILENAME,ASSEMBLED)	; ファイル FILENAME にパートプログラムブロックを書き込みます。
N390 IF (ERROR<>0)	; エラー分析
N400 SETAL(61000+ERROR)	
N410 ENDIF	
N430 ENDLOOP	
N440	
N450 LABEL1:	
N460	
N480 CALL FILENAME	; 生成されたサブプログラムを実行します。
N490	
N510 DELETE(ERROR,FILENAME)	; 実行後にファイルを再度削除します。
N520 IF (ERROR<>0)	
N530 SETAL(61000+ERROR)	
N540 ENDIF	
N550	
N560 M17	

例 2:AUXFUSYNC(...)を使用しない補助機能の削除と補助機能出力の生成

プログラムコード	コメント
N0610 DEF STRING[400] ASSEMBLED=""	
N0620 DEF STRING[31] FILENAME="/_N_CST_DIR/_N_AUXFU_SPF"	
N0630 DEF INT GROUPINDEX[10]	
N0640 DEF INT NUM	
N0650 DEF INT LAUF	
N0660 DEF INT ERROR	
N0670 DEF BOOL ISQUICK	
N0680 DEF BOOL ISSYNACT	

プログラムコード	コメント
N0690 DEF BOOL ISIMPL	
...	
N0760 AUXFUDEL("M",2,3,5)	; M2=3 (5 番目の補助機能グループ)解除
N0770	
N0790 AUXFUDELG(6)	; 6 番目のグループの収集された補助機能を
	; 解除します。
N0800	
N0810 IF ISFILE(FILENAME)	
N0830 DELETE(ERROR,FILENAME)	; ファイルはすでに存在し、削除する
	; 必要があります。
N0840 IF (ERROR<>0)	
N0850 SETAL(61000+ERROR)	
N0860 ENDIF	
N0870 ENDIF	
N0880	
; CAUTION!	
; If, for a multi-channel block search, auxiliary functions with AUXFUDEL/AUXFUDELG	
; are deleted from the global list of auxiliary functions, before the loop to	
; generate the subprogram FILENAME with AUXFUSYNC, the channels must be synchronized.	
The synchronization ensures that all delete requests were processed	
; in all channels and a consistent list is available.	
; Example:WAITM(99,1,2,3)	
N0890 LOOP	
N0920 AUXFUSYNC(NUM,GROUPINDEX,ASSEMBLED)	; グローバル補助機能リストから
	; 補助ファンクションブロックを
	; 生成する手順。
N0930	
N0940 IF (NUM==-1)	; チャネルのすべての補助機能が
	; 実行されました。
N0960 GOTOF LABEL1	
N0970 ENDIF	
N0980	
N1000 IF (NUM>0)	; 補助機能を出力する場合、
	; ブロックが生成されます。
N1010	
N1020 ASSEMBLED=""	
N1030	
N1050 FOR LAUF=0 TO NUM-1	; ブロックの収集された
	; 補助機能。
N1060	
N1080 IF GROUPINDEX[LAUF]<>0	; グローバルリストから解除された補助機能には、
	; グループインデックス 0 が含まれます。
N1090	
N1100 ISQUICK=$AC_AUXFU_SPEC[GROUPINDEX[LAUF]] BAND'H2'	

プログラムコード	コメント

```
N1110
N1120 ISSYNACT=$AC_AUXFU_SPEC[GROUPINDEX[LAUF]] BAND'H1000'
N1130
N1140 ISIMPL=$AC_AUXFU_SPEC[GROUPINDEX[LAUF]] BAND'H2000'
N1150
N1180 IF ISSYNACT                                    ; M 補助機能出力のブロックを
                                                     ; アセンブルします。
N1190           ASSEMBLED= ASSEMBLED << "WHEN TRUE DO "
N1200 ENDIF
N1210 ; 自動的に生成した M19 を SPOS[SPI(主軸番号>)] = IC(0)に割り当てます。
N1230 IF (ISIMPL AND ($AC_AUXFU_VALUE[GROUPINDEX[LAUF]==19))
N1240           ASSEMBLED= ASSEMBLED << "SPOS[SPI(" <<
                  $AC_AUXFU_EXT[GROUPINDEX[LAUF]] << ")=IC(0)"
N1260 ELSE
N1270           ASSEMBLED= ASSEMBLED << "M[" << $AC_AUXFU_EXT[GROUPINDEX[LAUF]] << "]="
N1280
N1290 IF ISQUICK
N1300             ASSEMBLED= ASSEMBLED << "QU("
N1310 ENDIF
N1320
N1330           ASSEMBLED= ASSEMBLED << $AC_AUXFU_VALUE[GROUPINDEX[LAUF]]
N1340
N1350 IF ISQUICK
N1360             ASSEMBLED= ASSEMBLED << ")"
N1370 ENDIF
N1380 ENDIF
N1400 ENDIF
N1420 ENDFOR
N1430
N1450 WRITE(ERROR,FILENAME,ASSEMBLED)                ; ファイルに補助機能ブロックを書き込みます。
N1460
N1470 IF ISSYNACT
N1480 ASSEMBLED="G4 F0.001"
N1490 WRITE(ERROR,FILENAME,ASSEMBLED)
N1500 ENDIF
N1510
N1520 ELSE
N1540 WRITE(ERROR,FILENAME,ASSEMBLED)                ; ファイルに補助機能ブロックを書き込みます。
N1550 ENDIF
N1560
N1570 ENDLOOP
N1580
N1590 LABEL1:
N1600
```

プログラムコード	コメント
N1620 CALL FILENAME	; 生成されたサブプログラムを実行します。
N1630	
N1650 DELETE(ERROR,FILENAME)	; 実行後にファイルを再度削除します。
N1660 IF (ERROR<>0)	
N1670 SETAL(61000+ERROR)	
N1680 ENDIF	
N1690	
N1700 M17	

9.14 自動出力補助機能

機能

自動出力補助機能とは、明確にプログラム指令がなく、他のシステムファンクション(座標変換選択、工具選択など)によって追加で出力される補助機能です。 これらの自動出力補助機能は、システムファンクションを呼び出しませんが、代わりに、その M コードに割り当てられている出力動作パラメータにしたがって収集され、PLC に出力されます。

パラメータ設定

自動出力される補助機能の M コードは、以下のマシンデータで定義します。

- MD22530 $MC_TOCARR_CHANGE_M_CODE (工具ホルダ変更時の M コード)
  このマシンデータの値は、工具ホルダを NC/PLC インタフェースで起動したときに出力される M コードの番号を示します。
  この値が正の場合は、常に M コードが、そのまま出力されます。
  値が負の場合は、工具ホルダの番号がマシンデータの値に追加され、この番号が出力されます。
- MD22532 $MC_GEOAX_CHANGE_M_CODE (ジオメトリ軸切り替え時の M コード)
  NC/PLC インタフェースによるジオメトリ軸の切り替え時に出力される M コードの番号。
- MD22534 $MC_TRAFO_CHANGE_M_CODE (座標変換変更時の M コード)
  NC/PLC インタフェースでのジオメトリ軸の座標変換切り替え時に出力される M コードの番号。

注記

出力される M コードの番号または MD22530/MD22532/MD22534 の値が 0 ～ 6、17、および 30 のいずれかである場合は、M コードは出力されません。この方法で生成された M コードが他の機能と競合するかどうかは監視されません。

出力動作

自動出力補助機能の場合は、マシンデータ MD22080 または MD22035 (事前定義またはユーザー定義の補助機能の出力動作)のビット 13 を設定します。

このビットは、システム変数$AC_AUXFU_SPEC[<n>]で確認できます。

自動出力補助機能 M19

NC/PLC インタフェースでの M19 の動作と SPOS または SPOSA の動作が均一となるよう、SPOS と SPOSA の場合には、補助機能 M19 を NC/PLC インタフェースに出力できます(「一般機能 (ページ 1562)」の章を参照してください)。

自動出力補助機能 M19 はブロック検索中に収集されます。

9.15 情報表示手段

補助機能に関する情報(出力状態など)は、以下の方法で確認できます。

- 操作画面のグループ別モーダル M 補助機能の表示
- パートプログラムおよびシンクロナイズドアクションでのシステム変数の確認

9.15.1 グループ別モーダル M 補助機能の表示

機能

M 補助機能の出力状態と確認応答状態は、グループ別に操作画面で表示できます。

必要条件

M 補助機能の機能に応じた確認応答と表示を使用するには、補助機能を PLC で、つまり、ユーザープログラム自体で管理してください。したがって、これらの補助機能の確認応答は、PLC プログラマの責任において、プログラムしてください。プログラマは、どのグループのどの補助機能の確認応答が必要かを把握している必要があります。

標準

PLC によって管理されていない M 機能は、NC から PLC に出力され、「伝送済み」としてマークされます。これらの補助機能については、機能の確認応答がありません。次のブロック検索後のスタートの後にどの補助機能が出力されるかをオペレータが把握できるよう、ブロック検索後に収集された M 補助機能がすべて表示されます。

PLC 動作

PLC 自体によって管理される補助機能グループの場合、**伝送**および**機能の完了**が有効になったとき、PLC ユーザープログラムによってこのグループのすべての補助機能が確認応答される必要があります。PLC プログラマは、これらのグループの補助機能をすべて認識している必要があります。

その他

グループ分けされた M 補助機能のみが表示されます。ブロック単位の表示も保持されます。最大 15 のグループを表示できますが、グループごとに、収集されるか PLC に出力された**グループ最後の M 機能のみ**が表示されます。M 機能は、その状態に応じて以下のようなさまざまな表示モードで表示されます。

状態	表示モード
補助機能が収集されています。	黄色のフォントの反転
補助機能が NC から PLC に出力されます	反転
補助機能が NC から PLC に転送済みで、伝送の確認応答が行われました。	灰色の背景に黒のフォント
補助機能が PLC によって管理されていて、PLC によって直接受け継がれました。	灰色の背景に黒のフォント
補助機能が PLC によって管理されていて、機能の確認応答が行われました。	灰色の背景に黒のフォント

表示の更新

この表示は、収集された補助機能が、PLC によって管理されていた補助機能より前、かつ NC によって管理されていた補助機能より前で、常に最初に出力されるように並べられます。収集された補助機能は、NC から PLC への出力が完了するまで収集済みとしてマークされます。PLC によって管理される補助機能は、別の補助機能に置き換わるまで保持されます。リセット時に、収集された補助機能と、NC によって管理される補助機能のみが削除されます。

9.15.2 システム変数の確認

機能

パートプログラムおよびシンクロナイズドアクションの中でシステム変数を使用して、グループ別に補助機能を確認できます。

$AC_AUXFU_... [<n>] = <値>

システム変数	意味	
$AC_AUXFU_PREDEF_INDEX[<n>]	<値>:	補助機能グループに対して収集した(ブロック検索)最後の補助機能、または出力される最後の事前定義補助機能のインデックス
		タイプ: INT
		指定したグループの補助機能がまだ出力されていない場合、または補助機能がユーザー定義の補助機能である場合は、変数の値が「-1」となります。
	<n>:	グループインデックス (0 … 63)
	注: 事前定義補助機能は、この変数により一義的に識別されます。	
$AC_AUXFU_TYPE[<n>]	<値>:	補助機能グループに対して収集した(ブロック検索)最後の補助機能、または出力される最後の補助機能のタイプ
		タイプ: CHAR
	<n>:	グループインデックス (0 … 63)
$AC_AUXFU_EXT[<n>] または M 機能用: $AC_AUXFU_M_EXT[<n>]	<値>:	補助機能グループに対して収集した(ブロック検索)最後の補助機能、または出力される最後の補助機能のアドレス拡張
		タイプ: INT
	<n>:	グループインデックス (0 … 63)
$AC_AUXFU_VALUE[<n>] または M 機能用: $AC_AUXFU_M_VALUE[<n>]	<値>:	補助機能グループに対して収集した(ブロック検索)最後の補助機能、または出力される最後の補助機能の値
		タイプ: REAL
	<n>:	グループインデックス (0 … 63)

システム変数	意味		
$AC_AUXFU_SPEC[<n>]	値:	補助機能グループに対して収集された(ブロック検索)最後の補助機能または出力される最後の補助機能の MD22080/ MD22035 (または QU のプログラミング)に従った、ビット符号による出力動作	
		タイプ:	INT
	<n>:	グループインデックス (0 … 63)	
	注: この変数を使用すると、補助機能を高速応答で出力するかどうかを決定できます。		
$AC_AUXFU_STATE[<n>] または M 機能用: $AC_AUXFU_M_STATE[<n>]	<値>:	補助機能グループに対して収集した(ブロック検索)最後の補助機能、または出力される最後の補助機能の出力状態	
		タイプ:	INT
		値の範囲:	0 ... 5
		0:	補助機能は存在しません
		1:	M 補助機能がブロック検索を介して収集されました
		2:	M 補助機能が PLC に出力されました。
		3:	M 補助機能が PLC に出力され、伝送確認応答が行われました
		4:	M 補助機能が PLC によって管理されていて、PLC によって受け継がれました。
		5:	M 補助機能が PLC によって管理されていて、機能の確認応答が行われました。
	<n>:	グループインデックス (0 … 63)	

例

第 1 グループのすべての M 補助機能が、出力された順序で格納されます。

```
id=1 every $AC_AUXFU_M_STATE[0]==2 do $AC_FIFO[0,0]=
$AC_AUXFU_M_VALUE[0]
```

参照先

システム変数の詳細については、以下を参照してください:

『リストマニュアル』、「システム変数」

9.16 必要条件

9.16.1 一般制約

主軸入れ替え

補助機能はチャネル別にパラメータ設定されるため、機能 「主軸入れ替え」が使用される場合、主軸が使用されるすべてのチャネルで、主軸用の補助機能を同じようにパラメータ設定してください。

工具管理

工具管理機能が有効な場合、次の制約が適用されます。

- T および M<k>機能は PLC に出力されません。
  注意:
  k は工具交換のための補助機能のパラメータ設定値です(初期設定: 6):
  MD22560 $MC_TOOL_CHANGE_M_CODE(工具交換のための補助機能)
- アドレス拡張がプログラムされていない場合、補助機能ではチャネルのメイン主軸またはマスタ工具ホルダが参照されます。
  メイン主軸の定義
  - MD20090 $MC_SPIND_DEF_MASTER_SPIND
  - パートプログラム命令: SETMS

  マスタ工具ホルダの定義
  - MD20124 $MC_TOOL_MANAGEMENT_TOOLHOLDER
  - パートプログラム命令: SETMTH

1 パートプログラムブロックあたりの補助機能の最大数

1 つのパートプログラムブロックで最大 10 の補助機能をプログラムできます。

DL (追加工具オフセット)

DL 機能には次の制約が適用されます。

- DL 機能は 1 つのパートプログラムブロックにつき 1 つだけプログラムできます。
- DL 機能がシンクロナイズドアクションで使用される場合、パラメータ 「値」は PLC に出力されません。

9.16.2 出力動作

ねじ切り

ねじ切 G33、G34 および G35 有効なときには、次の主軸用の補助機能に対して、その下の出力動作が常に有効です。

- M3(主軸右回転)
- M4(主軸左回転)

- 1 回の OB40 周期の出力時間(高速応答)
- 移動中の出力

主軸用の補助機能 M5(主軸停止)は、常にブロック終了時に出力されます。 M5 を含むパートプログラムブロックは、連続軌跡モードが有効なときでも、常にイグザクトストップで終了します。

シンクロナイズドアクション

シンクロナイズドアクションから出力された補助機能については、次のパラメータを除いて、パラメータ設定された出力動作が無視されます。

- ビット 0: 1 回の OB1 周期の出力時間(通常応答)
- ビット 1: 1 回の OB40 周期の出力時間(高速応答)

補助機能: M17 または M2/M30 (サブプログラム終了)

それ自身のパートプログラムブロック内

補助機能 M17、M2 または M30 のいずれかがパートプログラムブロック内の唯一の補助機能としてプログラム指令されていて、軸がまだ移動中の場合、その補助機能は軸の停止後まで PLC に出力されません。

パラメータ設定された出力動作のオーバライド

補助機能 M17 または M2/M30 のパラメータ設定された出力動作は、次のマシンデータで決定される出力動作によって置き換わります。

MD20800 $MC_SPF_END_TO_VDI, ビット 0(PLC へのサブプログラム終了/ストップ)

ビット	値	意味
0	0	補助機能 M17 または M2/M30(サブプログラム終了)は PLC に出力されません。 連続軌跡モードはサブプログラム終了時に中断されません。
	1	補助機能 M17 または M2/M30(サブプログラム終了)は PLC に出力されます。

補助機能: M1(オプショナルストップ)

パラメータ設定された出力動作のオーバライド

補助機能 M1 のパラメータ設定された出力動作は、次のマシンデータで定義された出力動作によって置き換わります。

MD20800 $MC_SPF_END_TO_VDI, ビット 1(PLC へのサブプログラム終了/ストップ)

ビット	値	意味
1	0	補助機能 M01(オプショナルストップ)は常に PLC に出力されます。 M01 は恒久的に第 1 の補助機能グループに割り当てられており、したがって常にブロック終了時に出力されるため、高速応答は無効です。
	1	補助機能 M01(オプショナルストップ)は、機能 「プログラムストップ」が HMI 操作画面で有効にされた場合にのみ PLC に出力されます。 高速応答の場合は、移動中に M1 が PLC に出力されます。 この機能が有効でない限り、連続軌跡モードは中断されません。

移動動作のないパートプログラムブロック

移動動作のないパートプログラムブロックでは、ブロック内のすべての補助機能が、パラメータ設定された出力動作にかかわりなく直ちに出力されます。

PLC ユーザープログラムに対する情報専用の主軸別補助機能出力

ブロック検索の終了時などのような特定のコントローラ状況では、収集された主軸用補助機能(M3、M4、M5、M19、M40...M45、M70 など)は PLC ユーザープログラムに対する情報のみの目的で NC/PLC インタフェースに出力されます。 コントローラは、収集された補助機能が**負の**アドレス拡張で入力されるパートプログラムブロック(アクションブロック)を生成します。 対応するシステムファンクションは実行**されません**。

例: M(-2) = 41 の場合、2 番目の主軸のギヤ選択切替が必要です。

9.17 例

9.17.1 事前定義補助機能の拡張

課題

チャネルの 2 番目の主軸に対する補助機能 M3、M4、および M5 のパラメータ設定

パラメータ設定 M3

必要条件

- マシンデータインデックス: 0 (1 番目のユーザー定義補助機能)
- 補助機能グループ: 5
- タイプと値: M3 (主軸右回転)
- アドレス拡張: チャネルの 2 番目の主軸に対応する値として 2
- 出力動作:
  - 1 OB1 サイクルの出力時間(通常応答)
  - 移動前の出力

パラメータ設定

```
MD22000 $MC_AUXFU_ASSIGN_GROUP[ 0 ]       = 5
MD22010 $MC_AUXFU_ ASSIGN_TYPE[ 0 ]       = "M"
MD22020 $MC_AUXFU_ ASSIGN_EXTENSION[ 0 ]= 2
MD22030 $MC_AUXFU_ ASSIGN_VALUE[ 0 ]      = 3
MD22035 $MC_AUXFU_ASSIGN_SPEC [ 0 ]       = 'H21'
```

パラメータ設定 M4

必要条件

- マシンデータインデックス: 1 (2 番目のユーザー定義補助機能)
- 補助機能グループ:5
- タイプと値: M4 (主軸左回転)
- アドレス拡張: チャネルの 2 番目の主軸に対応する値として 2
- 出力動作:
  - 1 OB1 サイクルの出力時間(通常応答)
  - 確認後の主軸応答
  - 移動中の出力

パラメータ設定

```
MD22000 $MC_AUXFU_ASSIGN_GROUP[ 1 ]      = 5
MD22010 $MC_AUXFU_ ASSIGN_TYPE[ 1 ]       = "M"
MD22020 $MC_AUXFU_ ASSIGN_EXTENSION[ 1 ]= 2
MD22030 $MC_AUXFU_ ASSIGN_VALUE[ 1 ]      = 4
MD22035 $MC_AUXFU_ASSIGN_SPEC [ 1 ]       = 'H51'
```

パラメータ設定 M5

必要条件

- マシンデータインデックス: 2 (3 番目のユーザー定義補助機能)
- 補助機能グループ:5
- タイプと値: M5 (主軸停止)
- アドレス拡張: チャネルの 2 番目の主軸に対応する値として 2
- 出力動作:
  - 1 OB1 サイクルの出力時間(通常応答)
  - 確認後の主軸応答
  - ブロックの終わりでの出力

パラメータ設定

```
MD22000 $MC_AUXFU_ASSIGN_GROUP[ 2 ]      = 5
MD22010 $MC_AUXFU_ ASSIGN_TYPE[ 2 ]       = "M"
```

MD22020 $MC_AUXFU_ ASSIGN_EXTENSION[2]= 2
MD22030 $MC_AUXFU_ ASSIGN_VALUE[2] = 5
MD22035 $MC_AUXFU_ASSIGN_SPEC [2] = 'H91'

9.17.2 補助機能の定義

タスク

次の構成の機械に対する、補助機能用マシンデータのパラメータ設定

主軸

- 主軸 1:メイン主軸
- 主軸 2:第 2 主軸

ギヤ選択

- 主軸 1:5 つのギヤ選択
- 主軸 2:ギヤ選択なし

冷却水オン/オフの切り替え機能

- 主軸 1
  - 「オン」= M50
  - 「オフ」= M51
- 主軸 2
  - 「オン」= M52
  - 「オフ」 = M53

必要条件

主軸 1(メイン主軸)

注記

初期設定の割り当て

- 主軸 1 (メイン主軸)の補助機能 M3、M4、M5、M70、および M1=3、M1=4、M1=5、M1=70 は、初期設定で、2 番目の補助機能グループに割り当てられます。
- 主軸 1(メイン主軸)の S 値および S1 値はすべて、標準で第 3 の補助機能グループに割り当てられます。

- 最後にプログラムされたギヤ選択は、ブロック検索の後に出力されます。この理由から、次の補助機能が 9 番目の補助機能グループに割り当てられます。
  - M40、M41、M42、M43、M44、M45
  - M1=40、M1=41、M1=42、M1=43、M1=44、M1=45
- 補助機能 M3、M4、M5、M70、および M1=3、M1=4、M1=5、M1=70 (2 番目の補助機能グループ)、および S と S1 の値(3 番目の補助機能グループ)の出力動作は、次のようになります。
  - 1 OB40 サイクルの出力時間(高速応答)
  - 移動前の出力
- ギヤ切り替えの補助機能 M40 ～ M45 および M1=40 ～ M1=45 9 (9 番目の補助機能グループ)の出力動作は次のようになります。
  - 1 OB1 サイクルの出力時間(通常応答)
  - 移動前の出力

主軸 2

- 方向反転の M 機能は、1 つのブロックで 1 つだけプログラムできます。最後にプログラムされた回転方向は、ブロック検索の後に出力されます。この理由から、次の補助機能が 10 番目の補助機能グループに割り当てられます。
  - M2=3、M2=4、M2=5、M2=70
- S2 値はすべて 11 番目の補助機能グループに割り当てられます。
- 補助機能 M2=3、M2=4、M2=5、M2=70 (10 番目の補助機能グループ)と S2 値(11 番目の補助機能グループ)の出力動作は次のようになります。
  - 1 OB40 サイクルの出力時間(高速応答)
  - 移動前の出力

冷却水

- 1 つのパートプログラムブロックで冷却水をオンおよびオフに切り替えることは許容されません。ブロック検索後、冷却水がオンまたはオフに切り替えられます。この理由から、次の補助機能が、たとえば 12 番目または 13 番目の補助機能グループに割り当てられます。
  - 12 番目の補助機能グループ:M50、M51
  - 13 番目の補助機能グループ:M52、M53
- 補助機能 M50、M51(12 番目の補助機能グループ)および M52、M53(13 番目の補助機能グループ)の出力動作は次のようになります。
  - 1 OB1 サイクルの出力時間(通常応答)
  - 移動前の出力

マシンデータのパラメータ設定

マシンデータは、パートプログラム内の各プログラミングによってパラメータ設定されます。

プログラムコード	コメント
$MN_AUXFU_MAXNUM_GROUP_ASSIGN = 21	; 1 チャネルあたりのユーザー定義補助機能の数
$MN_AUXFU_GROUP_SPEC[1]='H22'	; 2 番目の補助機能グループの出力動作
$MN_AUXFU_GROUP_SPEC[2]='H22'	; 3 番目の補助機能グループの出力動作
$MN_AUXFU_GROUP_SPEC[8]='H21'	; 9 番目の補助機能グループの出力動作
$MC_AUXFU_ASSIGN_TYPE[0]="M"	; 1 番目の補助機能の詳細:M40
$MC_AUXFU_ASSIGN_EXTENSION[0]=0	
$MC_AUXFU_ASSIGN_VALUE[0]=40	
$MC_AUXFU_ASSIGN_GROUP[0]=9	
	; ... (2 ～ 5 番目の補助機能についても同様)
$MC_AUXFU_ASSIGN_TYPE[5]="M"	; 6 番目の補助機能の詳細:M45
$MC_AUXFU_ASSIGN_EXTENSION[5]=0	
$MC_AUXFU_ASSIGN_VALUE[5]=45	
$MC_AUXFU_ASSIGN_GROUP[5]=9	
$MC_AUXFU_ASSIGN_TYPE[6]="M"	; 7 番目の補助機能の詳細:M1=40
$MC_AUXFU_ASSIGN_EXTENSION[6]=1	
$MC_AUXFU_ASSIGN_VALUE[6]=40	
$MC_AUXFU_ASSIGN_GROUP[6]=9	
	; ... (8 ～ 11 番目の補助機能についても同様)
$MC_AUXFU_ASSIGN_TYPE[11]="M"	; 12 番目の補助機能の詳細:M1=45
$MC_AUXFU_ASSIGN_EXTENSION[11]=1	
$MC_AUXFU_ASSIGN_VALUE[11]=45	
$MC_AUXFU_ASSIGN_GROUP[11]=9	
$MN_AUXFU_GROUP_SPEC[9] = 'H22'	; 10 番目の補助機能グループの出力動作
$MC_AUXFU_ASSIGN_TYPE[12]="M"	; 13 番目の補助機能の詳細:M2 = 3
$MC_AUXFU_ASSIGN_EXTENSION[12]=2	
$MC_AUXFU_ASSIGN_VALUE[12]=3	
$MC_AUXFU_ASSIGN_GROUP[12]=10	
$MC_AUXFU_ASSIGN_TYPE[13]="M"	; 14 番目の補助機能の詳細:M2 = 4
$MC_AUXFU_ASSIGN_EXTENSION[13]=2	
$MC_AUXFU_ASSIGN_VALUE[13]=4	
$MC_AUXFU_ASSIGN_GROUP[13]=10	
$MC_AUXFU_ASSIGN_TYPE[14]="M"	; 15 番目の補助機能の詳細:M2 = 5

プログラムコード	コメント
$MC_AUXFU_ASSIGN_EXTENSION[14]=2	
$MC_AUXFU_ASSIGN_VALUE[14]=5	
$MC_AUXFU_ASSIGN_GROUP[14]=10	
$MC_AUXFU_ASSIGN_TYPE[15]="M"	; 16 番目の補助機能の詳細:M2 = 70
$MC_AUXFU_ASSIGN_EXTENSION[15]=2	
$MC_AUXFU_ASSIGN_VALUE[15]=70	
$MC_AUXFU_ASSIGN_GROUP[15]=10	
$MN_AUXFU_GROUP_SPEC[10]='H22'	; 11 番目の補助機能グループの指定
$MC_AUXFU_ASSIGN_TYPE[16]="S"	; 17 番目の補助機能の詳細:S2=<すべての値>
$MC_AUXFU_ASSIGN_EXTENSION[16]=2	
$MC_AUXFU_ASSIGN_VALUE[16]=-1	
$MC_AUXFU_ASSIGN_GROUP[16]=11	
$MN_AUXFU_GROUP_SPEC[11]='H21'	; 12 番目の補助機能グループの指定
$MC_AUXFU_ASSIGN_TYPE[17]="M"	; 18 番目の補助機能の詳細:M50
$MC_AUXFU_ASSIGN_EXTENSION[17]=0	
$MC_AUXFU_ASSIGN_VALUE[17]=50	
$MC_AUXFU_ASSIGN_GROUP[17]=12	
$MC_AUXFU_ASSIGN_TYPE[18]="M"	; 19 番目の補助機能の詳細:M51
$MC_AUXFU_ASSIGN_EXTENSION[18]=0	
$MC_AUXFU_ASSIGN_VALUE[18]=51	
$MC_AUXFU_ASSIGN_GROUP[18]=12	
$MN_AUXFU_GROUP_SPEC[12]='H21'	; 13 番目の補助機能グループの指定
$MC_AUXFU_ASSIGN_TYPE[19]="M"	; 20 番目の補助機能の詳細:M52
$MC_AUXFU_ASSIGN_EXTENSION[19]=0	
$MC_AUXFU_ASSIGN_VALUE[19]=52	
$MC_AUXFU_ASSIGN_GROUP[19]=13	
$MC_AUXFU_ASSIGN_TYPE[20]="M"	; 21 番目の補助機能の詳細:M53
$MC_AUXFU_ASSIGN_EXTENSION[20]=0	
$MC_AUXFU_ASSIGN_VALUE[20]=53	
$MC_AUXFU_ASSIGN_GROUP[20]=13	

9.18 データリスト

9.18.1 マシンデータ

9.18.1.1 NC 別マシンデータ

番号	識別子: $MN_	説明
10713	M_NO_FCT_STOPRE	先読み停止を伴う M 機能
10714	M_NO_FCT_EOP	NC RESET 後の有効な主軸の M 機能
10715	M_NO_FCT_CYCLE	サブプログラムにより置換される M 機能
11100	AUXFU_MAXNUM_GROUP_ASSIGN	1 チャネルあたりのユーザー定義補助機能の最大数
11110	AUXFU_GROUP_SPEC	グループ別の出力動作
11450	SEARCH_RUN_MODE	ブロック検索後の動作

9.18.1.2 チャネルマシンデータ

番号	識別子: $MC_	説明
20110	RESET_MODE_MASK	パートプログラム起動後の制御初期設定の定義
20112	START_MODE_MASK	電源投入後および RESET 時またはパートプログラム終了時の制御初期設定の定義
20270	CUTTING_EDGE_DEFAULT	工具刃先のプログラムがないときの初期設定
20800	SPF_END_TO_VDI	PLC へのサブプログラム終了/停止
22000	AUXFU_ASSIGN_GROUP	ユーザー定義補助機能のグループ割り当て
22010	AUXFU_ASSIGN_TYPE	ユーザー定義補助機能のタイプ
22020	AUXFU_ASSIGN_EXTENSION	ユーザー定義補助機能のアドレス拡張
22030	AUXFU_ASSIGN_VALUE	ユーザー定義補助機能の値
22035	AUXFU_ASSIGN_SPEC	ユーザー定義補助機能の出力動作
22040	AUXFU_PREDEF_GROUP	事前定義補助機能のグループ割り当て
22050	AUXFU_PREDEF_TYPE	事前定義補助機能のタイプ

番号	識別子: $MC_	説明
22060	AUXFU_PREDEF_EXTENSION	事前定義補助機能のアドレス拡張
22070	AUXFU_PREDEF_VALUE	事前定義補助機能の値
22080	AUXFU_PREDEF_SPEC	事前定義補助機能の出力動作
22100	AUXFU_QUICK_BLOCKCHANGE	高速補助機能によるブロック変更の遅延
22110	AUXFU_H_TYPE_INT	H 補助機能のタイプ
22200	AUXFU_M_SYNC_TYPE	M 機能の出力タイミング
22210	AUXFU_S_SYNC_TYPE	S 機能の出力タイミング
22220	AUXFU_T_SYNC_TYPE	T 機能の出力タイミング
22230	AUXFU_H_SYNC_TYPE	H 機能の出力タイミング
22240	AUXFU_F_SYNC_TYPE	F 機能の出力タイミング
22250	AUXFU_D_SYNC_TYPE	D 機能の出力タイミング
22252	AUXFU_DL_SYNC_TYPE	DL 機能の出力タイミング
22254	AUXFU_ASSOC_M0_VALUE	プログラムストップの追加 M 機能
22256	AUXFU_ASSOC_M1_VALUE	オプショナルストップの追加 M 機能
22530	TOCARR_CHANGE_M_CODE	工具ホルダ変更用の M コード
22532	GEOAX_CHANGE_M_CODE	ジオメトリ軸切り替え用 M コード
22534	TRAFO_CHANGE_M_CODE	座標変換変更用の M コード
22560	TOOL_CHANGE_M_CODE	工具交換用の補助機能

9.18.2 信号

9.18.2.1 チャネルへの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
関連 M01 の起動	DB21、... .DBX30.5	DB320x.DBX14.5

9.18.2.2 チャネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
M 機能 1 ～ 5 変更	DB21、... .DBX58.0-4	DB250x.DBX4.0-4
M 機能 1 ～ 5 デコードなし	DB21、... .DBX59.0-4	-
S 機能 1 ～ 3 変更	DB21、... .DBX60.0-2	DB250x.DBX6.0
S 機能 1 ～ 3 高速	DB21、... .DBX60.4-6	-
T 機能 1 ～ 3 変更	DB21、... .DBX61.0-2	-
T 機能 1 ～ 3 高速	DB21、... .DBX61.4-6	-
D 機能 1 ～ 3 変更	DB21、... .DBX62.0-2	DB250x.DBX10.0
D 機能 1 ～ 3 高速	DB21、... .DBX62.4-6	-
DL 機能変更	DB21、... .DBX63.0	-
DL 機能高速	DB21、... .DBX63.4	-
H 機能 1 ～ 3 変更	DB21、... .DBX64.0-2	DB250x.DBX12.0-2
H 機能 1 ～ 3 高速	DB21、... .DBX64.4-6	-
F 機能 1 ～ 6 変更	DB21、... .DBX65.0-5	-
M 機能 1 ～ 5 高速	DB21、... .DBX66.0-4	-
F 機能 1 ～ 6 高速	DB21、... .DBX67.0-5	-
拡張アドレス、M 機能 1(16 ビット int)	DB21、... .DBB68-69	DB250x.DBB3004
M 機能 1 (DInt)	DB21、... .DBB70-73	DB250x.DBD3000
拡張アドレス、M 機能 2(16 ビット int)	DB21、... .DBB74-75	DB250x.DBB3012
M 機能 2 (DInt)	DB21、... .DBB76-79	DB250x.DBD3008
拡張アドレス、M 機能 3(16 ビット int)	DB21、... .DBB80-81	DB250x.DBB3020
M 機能 3 (DInt)	DB21、... .DBB82-85	DB250x.DBD3016
拡張アドレス、M 機能 4(16 ビット int)	DB21、... .DBB86-87	DB250x.DBB3028
M 機能 4 (DInt)	DB21、... .DBB88-91	DB250x.DBD3024
拡張アドレス、M 機能 5(16 ビット int)	DB21、... .DBB92-93	DB250x.DBB3036
M 機能 5 (DInt)	DB21、... .DBB94-97	DB250x.DBD3032
拡張アドレス、S 機能 1(16 ビット int)	DB21、... .DBB98-99	DB250x.DBB4004
S 機能 1(real)	DB21、... .DBB100-103	DB250x.DBD4000
拡張アドレス、S 機能 2(16 ビット int)	DB21、... .DBB104-105	DB250x.DBB4012
S 機能 2(real)	DB21、... .DBB106-109	DB250x.DBD4008

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
拡張アドレス、S 機能 3(16 ビット int)	DB21、... .DBB110-111	DB250x.DBB4020
S 機能 3(real)	DB21、... .DBB112-115	DB250x.DBD4016
拡張アドレス、T 機能 1(16 ビット int)	DB21、... .DBB116-117	-
T 機能 1(int)	DB21、... .DBB118-119	DB250x.DBD2000
拡張アドレス、T 機能 2(16 ビット int)	DB21、... .DBB120-121	-
T 機能 2(int)	DB21、... .DBB122-123	-
拡張アドレス、T 機能 3(16 ビット int)	DB21、... .DBB124-125	-
T 機能 3(int)	DB21、... .DBB126-127	-
拡張アドレス、D 機能 1(8 ビット int)	DB21、... .DBB128	-
D 機能 1(8 ビット int)	DB21、... .DBB129	DB250x.DBD5000
拡張アドレス、D 機能 2(8 ビット int)	DB21、... .DBB130	-
D 機能 2(8 ビット int)	DB21、... .DBB131	-
拡張アドレス、D 機能 3(8 ビット int)	DB21、... .DBB132	-
D 機能 3(8 ビット int)	DB21、... .DBB133	-
拡張アドレス、DL 機能(8 ビット int)	DB21、... .DBB134	-
DL 機能(real)	DB21、... .DBB136	-
拡張アドレス、H 機能 1(16 ビット int)	DB21、... .DBB140-141	DB250x.DBB6004
H 機能 1(real または DInt)	DB21、... .DBB142-145	DB250x.DBD6000
拡張アドレス、H 機能 2(16 ビット int)	DB21、... .DBB146-147	DB250x.DBB6012
H 機能 2(real または DInt)	DB21、... .DBB148-151	DB250x.DBD6008
拡張アドレス、H 機能 3(16 ビット int)	DB21、... .DBB152-153	DB250x.DBB6020
H 機能 3(real または DInt)	DB21、... .DBB154-157	DB250x.DBD6016
拡張アドレス、F 機能 1(16 ビット int)	DB21、... .DBB158-159	-
F 機能 1(real)	DB21、... .DBB160-163	-
拡張アドレス、F 機能 2(16 ビット int)	DB21、... .DBB164-165	-
F 機能 2(real)	DB21、... .DBB166-169	-
拡張アドレス、F 機能 3(16 ビット int)	DB21、... .DBB170-171	-
F 機能 3(real)	DB21、... .DBB172-175	-
拡張アドレス、F 機能 4(16 ビット int)	DB21、... .DBB176-177	-
F 機能 4(real)	DB21、... .DBB178-181	-

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
拡張アドレス、F 機能 5(16 ビット int)	DB21、... .DBB182-183	-
F 機能 5(real)	DB21、... .DBB184-187	-
拡張アドレス、F 機能 6(16 ビット int)	DB21、... .DBB188-189	-
F 機能 6(real)	DB21、... .DBB190-193	-
ダイナミック M 機能:M00 - M07	DB21、... .DBB194	DB250x.DBB1000
ダイナミック M 機能:M08 - M15	DB21、... .DBB195	DB250x.DBB1001
ダイナミック M 機能:M16 - M23	DB21、... .DBB196	DB250x.DBB1002
ダイナミック M 機能:M24 - M31	DB21、... .DBB197	DB250x.DBB1003
ダイナミック M 機能:M32 - M39	DB21、... .DBB198	DB250x.DBB1004
ダイナミック M 機能:M40 - M47	DB21、... .DBB199	DB250x.DBB1005
ダイナミック M 機能:M48 - M55	DB21、... .DBB200	DB250x.DBB1006
ダイナミック M 機能:M56 - M63	DB21、... .DBB201	DB250x.DBB1007
ダイナミック M 機能:M64 - M71	DB21、... .DBB202	DB250x.DBB1008
ダイナミック M 機能:M72 - M79	DB21、... .DBB203	DB250x.DBB1009
ダイナミック M 機能:M80 - M87	DB21、... .DBB204	DB250x.DBB1010
ダイナミック M 機能:M88 - M95	DB21、... .DBB205	DB250x.DBB1011
ダイナミック M 機能:M96 - M99	DB21、... .DBX206.0-3	DB250x.DBB1012.0-3
関連 M00/M01 有効	DB21、... .DBX318.5	DB330x.DBX4002.5

9.18.2.3 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
位置決め軸の F 機能(real)	DB31、... .DBB78-81	-

9.18.2.4 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
主軸の M 機能(int)	DB21、... .DBB86-87	DB370x.DBD0000
主軸の S 機能(real)	DB21、... .DBB88-91	DB370x.DBD0004

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

10

10.1 概略説明

チャネル

NC チャネルは、軸の手動移動とパートプログラムの自動運転の最小構成単位です。どの時点でも、チャネルは常に自動や MDI、ジョグなどの特定のモードになっています。チャネルは独立した NC とみなすことができます。

モードグループ

チャネルは常に一つのモードグループに属しています。一つのモードグループを複数のチャネルで構成することもできます。

モードグループは、そのモードグループの全チャネルが任意の時点で自動や MDI、ジョグなどの同一のモードであるという性質によって区別されます。これは、NC 内部のモード制御回路によって実施されます。

モードグループは独立したマルチチャネル NC とみなすことができます。

チャネルギャップ

チャネルの設定時には、同一シリーズの全ての機械で可能な限り一様な設定を作成するために、予約チャネルを設けることができます。この場合、実際に使用されるチャネルのみが有効になります。

プログラムテスト

新規パートプログラムの位置のテストまたは移動のために次のオプションが使用可能です。

- 指令値出力なしでのプログラムの実行
- シングルブロックモードでのプログラムの実行
- ドライラン送り速度でのプログラムの実行

- パートプログラムブロックのスキップ
- 計算ありまたはなしのブロックサーチ

ブロックサーチ

ブロックサーチ機能では、特定のプログラムポイントに移動するために、次のプログラムシミュレーションを実行できます。

- タイプ 1、輪郭での計算なし
- タイプ 2、輪郭での計算あり
- タイプ 4、ブロック終点での計算あり
- タイプ 5、履歴からのすべての必要なデータの計算ありの、選択したプログラムポイントの自動スタート
- ブロック検索後の ASUP の自動起動
- 多段ブロックサーチ
- 「プログラムテスト」モードでのチャネル間ブロックサーチ

プログラム運転

自動または MDI モードでのパートプログラムまたはパートプログラムブロックの実行は、プログラム運転と呼ばれています。実行時には、PLC インタフェース信号および命令によってプログラムシーケンスを制御できます。

チャネルごとに、チャネルマシンデータに対する初期設定を指定できます。これらの初期設定は、G グループや補助機能の出力などに影響を及ぼします。

パートプログラムは、関連するチャネルがリセット状態の場合にのみ選択できます。

さらに、それ以降のすべてのプログラム実行は、PLC プログラムと対応する命令によって処理されます。

- パートプログラムまたはパートプログラムブロックの起動
- パートプログラムの計算およびプログラム制御
- 命令、プログラムステータスおよびチャネルステータスのリセット
- オペレータアクションおよびプログラムアクションへの応答
- イベント起動のプログラム呼び出し

非同期サブプログラム(ASUP)、割り込みルーチン

割り込み入力により、割り込みルーチンまたは ASUP に含まれるより緊急性の高いイベントに応答できるよう、NC で現在のパートプログラムの実行を中断できます。

シングルブロック

シングルブロック機能を使用すると、ユーザーがパートプログラムを 1 ブロックずつ実行できます。

シングルブロック機能には 3 つの設定タイプがあります。

- SLB1: = IPO シングルブロック
- SLB2: = デコードシングルブロック
- SLB3: = サイクル内での停止

基本ブロック表示

既存のブロック表示とともに第 2 の基本ブロック表示を使用して、機械上で動作を行なうすべてのブロックを表示できます。

実際にアプローチされている目標位置は、絶対位置として表示されます。位置データは、ワーク座標系(WCS)または設定可能ゼロオフセットシステム(SZS)を基準とします。

外部ソースからのプログラム実行

複雑なワークを加工する場合、そのプログラムに対して十分なメモリが NC に存在しない場合があります。「外部ソースから実行」または「EES (外部記憶から実行)」機能を使用すると、外部メモリ(ハードディスクなど)からパートプログラムを呼び出して実行することができます。

電源投入後、リセット後などの動作

制御装置の

- 電源投入(POWER ON)
- リセット/パートプログラム終了
- パートプログラム起動

の後の応答は、マシンデータによる特定のシステム設定により、G 命令、工具長補正、座標変換、連結軸グループ、法線方向制御、プログラム可能主軸同期などの機能について変更できます。

M、T および D 機能によるサブプログラムの呼び出し

特定の用途では、M、T または D 機能と少数の NC 言語命令 SPOS、SPOSA の代わりに、サブプログラム呼び出しを使用した方が有利な場合があります。これは、工具交換プログラムの呼び出しなどに使用できます。

関連するマシンデータを使用して、M、T、または D 機能を含むサブプログラムを定義および制御できます。たとえば、ギヤ選択切替にこれを使用できます。

プログラム実行時間/ワークカウンタ

工作機械オペレータを支援するため、プログラム実行時間とワークカウンタに関する情報が提供されます。

このために、定義された機能は、**工具管理の機能と同じではなく**、主に工具管理機能の行われないシステムを対象としています。

10.2 モードグループ

モードグループ

モードグループには、NC がグループ化されて 1 つのサブシステムを形成している場合に、複数のチャネルが含まれます。原則として、NC 内で独立した「NC」となります

モードグループには基本的に、割り当てられたすべてのチャネルが、どの時点でも常に同じモード(自動、ジョグ、MDI)であるという特徴があります。

注記

この説明は、1 つのモードグループと 1 つのチャネルがあるという前提で継続します。
複数のチャネルが必要な機能(「軸の入れ替え」機能については、以下を参照してください。
参照先:
機能マニュアル、上級機能;「K5:チャネル間のプログラム調整」および「K10:チャネル間の軸入れ替え」を参照してください

パラメータ設定

チャネルは、以下のマシンデータでモードグループに割り当てます。

MD10010 $MN_ASSIGN_CHAN_TO_MODE_GROUP[<チャネルインデックス>] = モードグループ番号

同じモードグループ番号のすべてのチャネルは 1 つのモードグループになります。

注記

初期設定では、NC は常に、1 つのチャネルと 1 つのモードグループから構成されます。チャネルのない NC を設定することはできません。

モードグループ別の NC/PLC インターフェース信号

モードグループ別 NC/PLC インターフェース信号はデータブロック DB11 にあります。

モードグループ別 NC/PLC インターフェースは基本的に、以下のインターフェース信号から構成されます。

- 要求信号 PLC → NC
  - モードグループリセット
  - モードグループ停止、軸と主軸
  - モードグループ停止
  - モード切替禁止
  - モード:JOG、MDI、AUTOM.
  - シングルブロック:タイプ A、タイプ B
  - 運転機能 REF、REPOS、TEACH IN、INC x
- NC → PLC 状態信号
  - モードストローブ:JOG、MDI、AUTOMATIC
  - 運転機能ストローブ:REF、REPOS、TEACH IN
  - すべてのチャネルがリセット状態
  - モードグループリセット
  - モードグループ準備完了
  - アクティブモード:JOG、MDI、AUTOMATIC
  - 有効な運転機能:REF、REPOS、TEACH、INC x

有効な、および無効なチャネル、チャネルギャップ

有効なチャネル

モードグループ番号が 0 以外のチャネルは有効なチャネルです。

無効なチャネル

モードグループ番号が 0 のチャネルは無効なチャネルです。これは、制御装置で内部的にまったくメモリを占有しません。

チャネルギャップ

モードグループ番号が 0 のチャネルは、無効なチャネルというだけではありません。チャネルのシーケンスでの、いわゆるチャネルギャップでもあります。

チャネルギャップの利点は、類似した一連の機械で、ほぼ同じ設定データを保持できることです。特定の試運転では、該当する機械に必要なチャネルのみが有効になります。占有されていないメモリは、追加ユーザーメモリとして自由に使用できます。

表 10-1　例

マシンデータ	意味
MD10010 $MN_ASSIGN_CHAN_TO_MODE_GROUP[0] = 1	チャネル 1、モードグループ 1
MD10010 $MN_ASSIGN_CHAN_TO_MODE_GROUP[1] = 2	チャネル 2、モードグループ 2
MD10010 $MN_ASSIGN_CHAN_TO_MODE_GROUP[2] = 0	チャネル 3、使用されない
MD10010 $MN_ASSIGN_CHAN_TO_MODE_GROUP[3] = 1	チャネル 4、モードグループ 1
MD10010 $MN_ASSIGN_CHAN_TO_MODE_GROUP[4] = 2	チャネル 5、モードグループ 2

下記も参照

K2: 軸タイプ、座標系、フレーム (ページ 845)

S1:主軸 (ページ 1555)

10.2.1　モードグループ停止

機能

次の NC/PLC インタフェース信号を使用して、すべてのモードグループチャネルの軸または軸/主軸の移動動作を停止し、パートプログラムの実行を中断することができます。

DB11 DBX0.5 (モードグループ停止)

DB11 DBX0.6 (モードグループ停止、軸プラス主軸)

10.2.2 モードグループリセット

機能

モードグループリセットは、モードグループの下記の NC/PLC インタフェース信号を介して要求されます。

DB11 DBX0.7 = 1 (モードグループリセット)

働き

モードグループのチャネルに対する働きは、次のとおりです。

- パートプログラムの解析(先読み)が停止します。
- すべての軸と主軸が、輪郭を損なうことなく、その加速曲線に従ってゼロ速度まで減速します。
- PLC にまだ出力されていない補助機能が出力されなくなります。
- 先読みポインタが中断点に設定され、ブロックポインタが当該のパートプログラムの最初に設定されます。
- すべての初期設定(G 命令など)が、パラメータ設定された値に設定されます。
- 「チャネルリセット」条件を持つすべてのアラームが解除されます。

モードグループのチャネルがすべてリセット状態である場合の動作は、次のとおりです。

- 「モードグループリセット」解除条件を持つアラームがすべて解除されます。
- NC/PLC インタフェースによって、モードグループリセットの完了と、モードグループの運転準備完了が示されます。
  DB11 DBX6.7 (すべてのチャネルがリセット状態)
  DB11 DBX6.3 = 1 (モードグループ準備完了)

10.3 モードタイプおよびモードタイプ変更

単一モード

モードグループのすべてのチャネルは、常に同じモードになります。

- AUTOMATIC
- JOG
- MDI

個々のチャネルが異なるモードグループに割り当てられている場合、チャネル切り替えによって、対応するモードグループが有効になります。これにより、チャネル切り替えによってモード変更をおこなえるようになります。

モード

以下のモードを使用できます。

- **AUTOMATIC**
  パートプログラムの自動実行:
  - パートプログラムのテスト
  - モードグループのすべてのチャネルを同時に起動できます。
- **Automatic での JOG**
  AUTOMATIC での JOG は、使いやすさを目的とした AUTOMATIC の拡張したモードです。境界条件で許容される限り、AUTOMATIC モードを終了することなく JOG を実行できます。

- **JOG**
  機械操作パネルの移動キーまたは機械操作パネルに接続された手動パルス発生器を介した軸の手動移動
  - ASUB により、または静的シンクロナイズドアクションを介して実行される移動では、チャネル別信号とインタロックが考慮されます。
  - 連結が考慮されます。
  - モードグループのすべてのチャネルを有効にできます。
- **MDI**
  手動データ自動(ブロックがユーザーインタフェースを介して入力されます)
  - パートプログラムとパートプログラム区間の制限付き実行。
  - パートプログラムのテスト
  - 1 つのモードグループにつき最大 1 つのチャネルを有効にできます(TEACH IN にのみ適用)。
  - JOG、REPOS、TEACH IN などの下位運転機能では、軸を手動でのみ移動させることができます。

全モードに適用

モード間シンクロナイズドアクション

以下の機能のすべてのモードで IDS 毎にモーダルシンクロナイズドアクションをチャネルと並列に実行できます。

- コマンド軸機能
- 主軸機能
- テクノロジサイクル

選択

操作画面のソフトキーを使用して、必要なモードを選択できます。

この選択(AUTOMATIC、MDI、JOG)は、NC/PLC インタフェースで PLC に転送されますが、有効になりません:
DB11 DBX4.0, 0.1, 0.2 (ストローブモード)

起動と優先順位

モードグループのモードは、次の NC/PLC インタフェースを介して有効になります。

DB11 DBX0.0, 0.1, 0.2 (モード)

複数のモードを同時に選択している場合、優先順位は次のようになります。

優先順位	モード	モードグループ信号(NC → PLC)
1 番目の優先順位、高	JOG	DB11 DBX0.2
2 番目の優先順位、中	MDI	DB11 DBX0.1
3 番目の優先順位、低	AUTOMATIC	DB11 DBX0.0

出力

モードグループの現在のモードは、次の NC/PLC インタフェースを介して表示されます。

DB11 DBX6.0, 0.1, 0.2 (アクティブモード)

モードグループ信号(NC → PLC)	有効な運転モード
DB11 DBX6.2	JOG
DB11 DBX6.1	MDI
DB11 DBX6.0	AUTOMATIC

運転機能

運転機能はモード内で選択でき、それがモードグループ内でも適用されます。

- **JOG** モード内の運転機能
  - REF (レファレンス点復帰)
  - REPOS (再位置決め)
  - JOG 後退 (工具方向の後退動作)
- **MDI** モード内の運転機能
  - REF (レファレンス点復帰)
  - REPOS (再位置決め)
  - TEACHIN (軸位置のティーチング)

NC/PLC インタフェース信号

- DB11 DBX5.0, 0.1, 0.2 (運転機能ストローブ):必要条件
- DB11 DBX1.0, 0.1, 0.2 (運転機能):起動
- DB11 DBX7.0-2 (有効な運転機能):フィードバック信号

チャネル状態

- **チャネルリセット**
  機械が初期状態です。これは、工作機械メーカの PLC プログラムで定義されます(たとえば電源オン時やプログラムの終了時に)。
- **チャネル起動中**
  プログラムが起動されて、そのプログラムが実行中であるか、またはレファレンス点復帰を実行中です。
- **チャネル中断**
  実行中のプログラムまたはレファレンス点復帰が中断されています。

モード内の機能

モードは、ユーザー専用の機能によって補完されます。個々の機能は技術で、機械に依存せず、3 つのチャネル状態「チャネルリセット」、「チャネル起動中」、および「チャネル中断」のいずれかから開始または実行できます。

TEACH IN 運転機能の補足条件

TEACH IN は、次のような有効な軸グループのマスタ軸またはスレーブ軸に対しては使用できません。

- ガントリ軸グループまたはガントリ軸ペア
- マスタ軸とスレーブ軸の連結軸グループ

Automatic での JOG

AUTOMATIC モードでの JOG が使用できるのは、モードグループが「リセット」状態であり、かつ軸がジョグ対応である場合のみです。モードグループの「リセット」状態の意味は、次のとおりです。

- すべてのチャネルが「リセット」状態です
- すべてのプログラムがキャンセルされています
- DRF がどのチャネルでも有効ではありません

軸は、以下のどの状態でもない場合、**JOG 対応**です

- 同時位置決め軸としての PLC 軸 (PLC からの軸の要求)
- コマンド軸 (軸がシンクロナイズドアクションによってプログラム指令されており、移動がまだ完了していない)

- 回転主軸 (RESET にもかかわらず回転する主軸)
- 非同期揺動軸

注:「ジョグ対応」という特性は、「AUTOMATIC での JOG」機能とは独立しています。

起動

「AUTOMATIC での JOG」機能は、次のマシンデータによって起動できます。

MD10735 $MN_JOG_MODE_MASK

- 電源投入の前に、次のマシンデータを設定してください。
  MD10735 $MN_JOG_MODE_MASK、ビット 0 = 1
- ユーザーが AUTO に切り替えます(PLC ユーザーインタフェース DB11 DBX0.0 = 0→1 エッジ)。次に、NC のすべてのモードグループチャネルで、事前にチャネル状態「リセット」とプログラム状態「中止」になっていた場合、「AUTOMATIC での JOG」が有効になります。当該の軸は「ジョグ対応」にしてください。DRF を無効にしてください(まだ無効でない場合)。
- チャネル状態が「リセット」ではなく、かつプログラム状態が「中止」ではないモードグループチャネルすべてで、リセットが起動されるか、実行中のプログラムが M30/M2 によって終了されます。
- 当該の軸は自動的に「ジョグ対応」になります(たとえば、軸入れ替え:PLC → NC)。

注:ほとんどの用途で、移動される軸は「ジョグ対応」であり、AUTOMATIC への切り替えによって「AUTOMATIC での JOG」も有効になります。

特性

- +/–キーによって JOG 動作が生じ、モードグループが**内部的に** JOG に切り替わります(「内部 JOG」など)。
- ハンドルを動かすことで JOG 移動が生じ、モードグループが**内部的に** JOG に切り替わります(DRF が有効である場合を除く)。
- 実行中の JOG 移動は、インクレメントの終点位置に到達するまで(終点位置が設定されている場合)、または「残移動距離削除」によって移動が中止されるまで完了しません。
  このように、ストップキーを使用してインクレメントを停止し、更にスタートキーを使用してインクレメントを終点位置に移動させることができます。この間、NC は「内部 JOG」のままになります。部分インクレメントは可能ですが、ストップキーを使用して部分インクレメントを中断してはいけません。インクレメント中に移動キーを離すと中断する、というモードがあります。

- JOG 移動がない場合は、「AUTOMATIC での JOG」が「自動」と同様に動作します。具体的には、スタートキーによって、選択されたパートプログラムが起動し、適切な HMI ソフトキーによってブロック検索が開始します。
- JOG 移動が有効である場合、NC は内部的に JOG モードであり、ブロック検索要求は拒否され、スタートキーによってパートプログラムを起動することはできません。スタートキーによって残りのインクレメンタル送りが開始されるか、あるいは、まったく影響がありません。
- JOG モードでモードグループ軸が移動している間は、モードグループは内部的に JOG モードのままです。
  備考:この状態は、軸の JOG 移動から始まり、*別の*軸の JOG 移動の終了によって終わることができます。
- JOG 動作が有効である軸に対して、軸入れ替えはできません(軸のモードグループが変わる可能性があります)。NC は軸入れ替えの要求のすべてを無視します。
- PLC ユーザーインタフェースでは「Automatic」モードが示されます。
  - DB11 DBX6.0, 6.1, 6.2 = 1
  - DB11 DBX7.0, 7.1, 7.2 = 0
- 「AUTOMATIC での JOG」では、モードグループが「モードグループリセット」であるかどうかが NC/PLC インタフェースに表示されます。
  - DB11 DBX6.4 (モードグループリセット、モードグループ 1)
  - DB11 DBX26.4 (モードグループリセット、モードグループ 2)
  - DB11 DBX46.4 (モードグループリセット、モードグループ 3)
- 「AUTOMATIC での JOG」では、NC が「内部 JOG」に自動切り替えを行ったかどうかが NC/PLC インタフェースに出力されます。
  - DB11 DBX6.5 (NC 内部ジョブ有効、モードグループ 1)
  - DB11 DBX26.5 (NC 内部ジョブ有効、モードグループ 2)
  - DB11 DBX46.5 (NC 内部ジョブ有効、モードグループ 3)

必要条件

「AUTOMATIC での JOG」が内部的に JOG に切り替えることができるのは、モードグループが「モードグループリセット」状態である場合に限られます。つまり、停止されたプログラムの途中ですぐにジョグ移動をおこなうことはできません。この状況で、モードグループの*すべてのチャネル*で JOG キーまたはリセットキーを押すと、ジョグ移動が可能です。

AUTOMATIC を選択すると、INC キーが無効になります。希望のインクレメントを選択するには、INC キーを再度押してください。NC が「内部 JOG」に切り替わると、選択されたインクレメントが保持されます。

ジオメトリ軸または旋回軸のジョグを行おうとすると、NC が「内部 JOG」に切り替わり、動作が実行されます。この方法で複数の軸を物理的に移動させることができます。これらの軸はすべて「ジョグ対応」でなければなりません。

JOG 移動に続いて、NC により「内部 JOG」が再び解除され、AUTO モードが再び選択されます。内部モード変更は、動作が完了するまで遅延されます。これにより、たとえばハンドル使用時に、複数の不要な切り替え操作を回避することができます。PLC は、「内部 JOG 有効」PLC 信号だけを使用できます。

軸が有効でない場合でも、NC が「内部 JOG」に切り替わります。

下記も参照

R1:レファレンス点復帰 (ページ 1497)

10.3.1 各モードの監視機能およびインタロック

チャネルステータスによって決まる監視機能

運転モードにおける監視機能

各運転モードで、各種の監視機能が有効になります。 これらの監視機能は、特定の加工や機械とは一切関係ありません。

特定のモードでは、その動作状態に応じて、これらの監視機能の一部だけが有効になります。 どのモードおよび動作状態で、どの監視機能が有効になるかは、チャネルステータスによって決まります。

運転モードにおけるインタロック機能

異なる動作モードで、各種のインタロック機能を有効にできます。 これらのインタロック機能は、特定の加工または機械とは一切関係ありません。

インタロック機能の大半は、動作状態に応じてすべてのモードで有効にできます。

10.3.2 モード変更

はじめに

モード変更は、モードグループインタフェース(DB11)を介して要求され、有効になります。モードグループは、AUTOMATIC、JOG または MDI のいずれかのモードになります。つまり、同じモードグループの複数のチャネルが同時に別々のモードになることはできません。

どのモードへの移行が可能で、どのようにして移行を実行するかを、PLC プログラムで機械ごとに設定できます。

注記

信号「チャネルステータス起動中」の出力中は、モードは内部的に変更されません。ただし、問題のないモード変更については、すべてのチャネルが許容運転モードになっている必要があります。

考えられるモード変更

次の表では、1 つのチャネルについて考えられるモード変更を示しています。

	AUTOMATIC		JOG			MDI		
				AUTO	MDI	手動動作		AUTO
	リセット	中断中	リセット	中断中	中断中	リセット	中断中	有効	中断中
AUTOMATIC			X	X		X			
JOG	X	X				X	X		X
MDI	X	X	X		X				

可能なモード変更を「X」で示しています。

特別な事例

- **モード変更時の異常**
  モード変更要求がシステムによって拒否された場合、メッセージ「運転モードは NC ストップの後まで変更できません」が出力されます。このメッセージは、チャネルステータスを変更しなくてもクリアできます。
- **モード変更無効**
  モード変更は、インタフェース信号
  DB11、DBX0.4 (モード変更無効)を使用して禁止できます。

  これにより、モード変更要求がマスクされます。
  オペレータに対して、モード変更が無効であることを示すメッセージを準備してください。これに対して、システムによって発行される信号はありません。
- **MDI からジョグへのモード変更**
  MDI から JOG へのモード変更後にモードグループのすべてのチャネルがリセット状態になる場合、NC は JOG から AUTO に切り替わります。この状態では、パートプログラムの命令 START または INIT を実行できます。
  モード変更後にモードグループのチャネルがリセット状態でなくなる場合、パートプログラムの命令 START は拒否され、アラーム 16952 が発行されます。

10.4 チャネル

NC のチャネルで、ユーザーが定義したパートプログラムが実行されます。

以下の特性がチャネルの特徴です。

- 1 つのチャネルは常に、1 つのモードグループ (ページ 572)(BAG)に割り当てられます。
- どの時点でも、1 つのチャネルが実行できるのは、1 つのパートプログラムのみです。
- 機械軸、ジオメトリ軸、付加軸、および主軸が 1 つのチャネルに割り当てられます。これらの軸のみが、チャネルで実行されるパートプログラムを介して移動できます。
- チャネルは内部単位から構成されます。
  - 先読み(プログラムのデコーディングとブロック解析)
  - メインラン(軌跡と軸の補間)

- チャネルには、PLC とのインターフェースがあります。この NC/PLC インターフェースを介して PLC ユーザープログラムは、さまざまなチャネル別状態データを読み取り、チャネルへの要求を書き込むことができます。
- チャネルでは、チャネル別工具オフセット (ページ 1767)が有効になっています。
- チャネルでは、チャネル別座標系 (ページ 867)が有効になっています。
- 各チャネルには、初期設定で、一義的なチャネル名があります。定義されたチャネル名は、以下のマシンデータで変更できます。
  MD20000 $MC_CHAN_NAME = "<チャネル名>"

複数のチャネルを組み合わせて、1 つのモードグループにすることができます。モードグループのチャネルは、常に同じモードになります(自動、ジョグ、MDI)。

チャネル設定

次のマシンデータを使用して、独自のチャネル名称をチャネルに割り当てることができます。

MD20000 $MC_CHAN_NAME (チャネル名称)

その後、マシンデータによってさまざまな軸が、使用可能なチャネルに割り当てられます。1 つの軸/主軸に対して一度に一つのチャネルだけが指令値を出力できます。軸/主軸のフィードバック値は、同時に複数のチャネルによって読み取ることができます。軸/主軸は、関連するチャネルによって登録されていなければなりません。

次のチャネル別設定もマシンデータによっておこなえます。

- 次のマシンデータによる、G グループのプログラム初期設定
  MD20150 $MC_GCODE_RESET_VALUES (G グループのリセット設定)
- 組み合わせと出力タイミング間に関する補助機能グループ
- 機械軸とジオメトリ軸の座標変換条件
- パートプログラムの実行に関するその他の設定

チャネル割り当ての変更

チャネル設定のオンライン変更を、パートプログラムまたは PLC ユーザープログラムにプログラム指令できません。チャネル設定の変更は、マシンデータを使用しておこなってください。チャネル設定の変更は、再度電源を投入した後で初めて有効になります。

コンテナ軸とリンク軸

軸コンテナとは、軸のグループを 1 つのコンテナ内にまとめたものです。このような軸をコンテナ軸と呼びます。これには、コンテナスロットへのポインタ (関連するコンテナ内のリングバッファロケーション) をチャネル軸に割り当てる作業が含まれます。コンテナ内の軸の 1 つが一時的にこのスロットに存在します。

軸コンテナ内の各機械軸は常に、ただ **1 つの**チャネル軸に割り当てられていなければなりません。

リンク軸は、固定された 1 つのチャネルに割り当てることも、ローカル NCU またはその他の NCU の複数のチャネルに次々に(軸コンテナスイッチを使用して)割り当てることもできます。リンク軸が物理的に接続されていない NCU に属するチャネルの 1 つからみた場合、そのリンク軸はローカル軸ではありません。

リンク軸とチャネルとの割り当ては、次のように実施されます。

- マシンデータを使用した固定割り当ての場合:
  直接論理機械軸イメージでリンク軸を示すことができます。
- 可変割り当ての場合:
  軸コンテナスロットマシンデータでリンク軸を示すことができます。

リンク軸とコンテナ軸に関する詳細は、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』; 「NCU と操作パネルの n:m 接続(B3)」

インタフェース信号

1 番目のチャネルの信号は DB21 の NC/PLC インタフェースに置かれ、チャネル 2 の信号は DB22 に置かれます。1 つのチャネルまたは複数のチャネルを PLC または NC から監視と制御することができます。

チャネル別加工仕様

使用する加工は、チャネル毎に指定することができます。

MD27800 $MC_TECHNOLOGY_MODE

納品時、マシンデータは標準としてフライス加工に対して有効になっています。

PLC を使用する主軸機能

ファンクションブロック FC18 のほかに、主軸機能も実行中のパートプログラムと並列に、NC/PLC インタフェース信号を介して開始と停止することができます。

必要条件

- チャネル状態:「切断」または「リセット」
- プログラム状態:「中断」または「キャンセル」

インタフェース信号を介して PLC で制御できる機能は、次のとおりです。

- 停止(M5 に相当)
- 右回りの回転方向で開始(M3 に相当)
- 左回りの回転方向で開始(M4 に相当)
- ギヤ選択指定
- 位置決め(M19 に相当)

複数のチャネルの場合、PLC で開始された主軸は開始時に自身が割り当てられたチャネルで有効です。

特定の主軸のインターフェースの詳細については、「S1:主軸 (ページ 1555)」の章を参照してください。

PLC 制御単独軸の操作

軸をチャネルではなく、PLC から制御することもできます。このために、PLC は NC/PLC インタフェースを介して NC に軸を要求します。

DB31、... DBX28.7 = 1 (PLC 制御軸)

PLC で制御できる機能は、次のとおりです。

- 軸/主軸動作取り消し(残移動距離削除に相当)
- 軸/主軸の停止または中断
- 軸/主軸動作の再開(移動動作続行)
- 初期状態への軸/主軸のリセット

チャネル別の信号交信(PLC → NC)の詳細情報については、「P3: SINUMERIK 840D sl 用 PLC 基本プログラム (ページ 1043)」の章を参照してください。

独立した単独軸操作の正確な機能は、以下に記述されています。

参照先:
『機能マニュアル、上級機能』;「位置決め軸(P2)」

10.4.1 起動禁止、グローバルおよびチャネル別

機能

ShopMill/ShopTurn では、初期設定でプログラムは運転操作エリアでのみ起動できます。その他の領域(工具など)では、プログラムの起動はグローバルスタートロック(PI サービス)によって防止されます。

自動スタートロックシーケンスは、NC/PLC インタフェース信号を使用して解除できます。

NC/PLC インタフェース信号

スタートロックの解除

有効なスタートロックは、次のインタフェース信号により解除されます。

DB21、... DBX7.5 (スタートロックの解除)

マシンデータ

スタートロックの設定時に起動が要求された場合のアラームを設定するために、次のマシンデータを使用します。

MD11411 $MN_ENABLE_ALARM_MASK、ビット 6

OPI 変数

変数	説明
startLockState	グローバルスタートロックの状態
chanStartLockState	チャネル別のスタートロックの状態
startRejectCounter	グローバルスタートロックのスタートカウンタ
startLockCounter	チャネル別のスタートロックのスタートカウンタ

PI サービス

スタートロックの設定(_N_STRTLK)

スタートロックは、PI サービス_N_STRTLK を使用して、グローバルまたはチャネルごとに設定できます。

- グローバルスタートロック
  グローバルスタートロックが設定されている場合、NC プログラムの起動はすべてのチャネルでロックされます。すでに実行中の NC プログラムは影響を受けません。NC プログラムの次の起動のみがロックされます。
- チャネル別のスタートロック
  チャネル別のスタートロックでは、NC プログラムの起動は AUTOMATIC モードおよび「リセット」チャネル状態でのみロックされます。

グローバルまたはチャネル別のスタートロックのリセット(_N_STRTUL)

PI サービス_N_STRTUL により、スタートロックはグローバルまたはチャネル別にリセットできます。

10.5 プログラムテスト

新しいパートプログラムをテストするために、いつくかの機能が準備されています。 これらの機能の目的は、機械での危険を低減し、テスト段階での所要時間を短縮することです。 効果を高めるために、複数のプログラム機能を同時に有効にすることができます。

テストオプション

次に、以下のオプションについて説明します。

- 指令値出力のないプログラム実行
- シングルブロックモードでのプログラム実行
- ドライラン送り速度でのプログラム実行
- パートプログラムブロックのスキップ
- 計算ありまたはなしのブロックサーチ

10.5.1 指令値の出力のないプログラム実行

機能

「プログラムテスト」状態で、軸/主軸指令値を出力せずにパートプログラムが実行されます。

このオプションを使用すると、パートプログラムのプログラムされた軸位置と補助機能出力を確認できます。このプログラムシミュレーションは、拡張構文チェッカとしても使用できます。

選択

この機能は、操作画面の[プログラム制御]メニューで選択できます。

選択時に、次のインタフェース信号が設定されます。

DB21、... DBX25.7 (プログラムテスト選択済)

これによって機能が有効になるわけではありません。

起動

この機能は、次のインタフェース信号を介して有効になります。

DB21、... DBX1.7 (プログラムテスト起動)

表示

操作画面上の該当する欄が反転表示され、プログラムテスト有効のアンサーバックとして、次のインタフェース信号が PLC で設定されます。

DB21、... DBX33.7 (プログラムテスト有効)

プログラム起動と実行

プログラムテスト機能が有効の場合、パートプログラムをインタフェース信号を介して開始し、実行することができます(補助機能の出力、待機時間、G 命令の出力などを含めて)。

DB21、... DBX7.1 (NC スタート)

ソフトウェアリミットスイッチ、ワーキングエリアリミットなどの安全機能は、引き続き有効です。

通常のプログラム運転との唯一の違いは、すべての軸(主軸を含む)に対して内部**軸無効**が設定されるということです。機械軸は移動せず、フィードバック値は、出力されない指令値から内部的に生成されます。プログラムされた速度は変化しません。これは、操作画面に表示される位置および速度の情報が、通常のパートプログラム実行中に出力される情報とまったく同じであるということを意味します。位置制御は中断されません。したがって、この機能をオフにした場合、軸の原点確立は必要ありません。

注記

イグザクトストップ用の信号 DB31、... DBX60.6/60.7 (汎用/精密イグザクトストップ)は、機械の実際の状態を反映します。

これらの信号がプログラムテスト中に取り消されるのは、軸がその指令位置から押し出されたときだけです(プログラムテスト中、指令位置は一定です)。

信号 DB21、... DBX33.7 (プログラムテスト有効)に関し、PLC プログラムとパートプログラムの両方で、変数$P_ISTEST によって、テスト中のこれらの信号に対応する動作または分岐を実行できます。

注記

ドライラン送り速度

「軸動作のないプログラム実行」は、「ドライラン送り速度」機能を使用して起動することもできます。この機能を使用すると、送り速度が小さいパートプログラム部分をより短時間で処理できます。

注記

工具管理機能

軸無効であるため、工具マガジンの割り当てがプログラムテスト中に変更されることはありません。工具管理機能システムのデータのデータとマガジンの整合性が損なわれないように PLC アプリケーションを作成してください。ツールボックスディスクには、基本 PLC プログラムのサンプルが含まれています。

10.5.2 シングルブロックモードでのプログラムの実行

機能

「シングルブロックモードでのプログラムの実行」の場合、パートプログラムの実行はプログラムブロック毎に停止します。 工具径補正またはノーズ R 補正が選択されている場合、コントローラによって挿入されたすべての中間ブロックの後で処理が停止します。

プログラム状態は「プログラム状態停止」に切り替わります。

チャネル状態は動作中のままです。

NC スタートで、その次のプログラムブロックが処理されます。

用途

パートプログラムをブロック毎に実行して、個々の加工ステップをチェックすることができます。 実行したパートプログラムブロックが正しく機能していると判断したら、次のブロックを呼び出すことができます。

シングルブロックタイプ

次のシングルブロックタイプが識別されます。

- デコードシングルブロック
  このタイプのシングルブロックを使用すると、パートプログラムのすべてのブロック(移動動作のない単なる演算ブロックであっても)が「NC スタート」によって順次、処理されます。
- アクションシングルブロック(初期設定)
  このタイプのシングルブロックを使用すると、アクション(移動動作や補助機能出力など)を実行するブロックが一つずつ処理されます。
  解析中に追加生成されたブロック(たとえば、鋭角での工具径補正)もシングルブロックモードで処理されます。
  しかし、計算ブロックではアクションが発生しないため、処理が停止しません。

シングルブロックのタイプは、[プログラム制御]メニューの操作画面を介して選択します。

⚠ 注意

シングルブロックタイプシリーズの機能の特徴

一連の G33/G34/G35 ブロックでは、「ドライラン送り速度」が選択されている場合のみシングルブロックが有効です。

演算ブロックは、シングルステップモードでは処理されません(シングルデコーディングブロックが有効な場合のみ)。

SBL2 も、G33/G34/G35 では無効です。

選択

次の方法でシングルブロックモードを選択できます。

- 機械操作パネル(「シングルブロック」キー)
- 操作画面
  正確な手順については、以下を参照してください。
  参照先
  インストールされている HMI アプリケーションの『操作マニュアル』

起動

この機能は、PLC ユーザープログラムで次のインタフェース信号を介して有効になります。

DB21、... DBX0.4 (シングルブロックの起動)

表示

シングルブロックモードが有効であれば、操作画面上のステータス行中の関連する欄が反転表示されます。

シングルブロックモードでは、パートプログラムブロックが 1 つ処理されると、次のインタフェース信号が設定されます。

DB21、... DBX35.3 (プログラム状態中断)

シングルブロック停止を伴わない処理

シングルブロックモードが選択されていても、たとえば次のような特定のプログラム実行に対してシングルブロック停止を伴わない処理を設定できます。

- 内部 ASUB
- ユーザー ASUB
- 中間ブロック
- ブロック検索グループブロック(アクションブロック)
- 初期設定ブロック
- DISPLOF を持つサブプログラム
- 再解析できないブロック
- 再位置決めできないブロック

- 移動情報のない再位置決めブロック
- 工具アプローチブロック

この設定は、次のマシンデータを使用しておこないます。

MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (シングルブロック停止のマスク)

参照先
『リストマニュアル』、「マシンデータの詳しい説明」

10.5.3 ドライラン送り速度でのプログラムの実行

機能

「ドライブラン送り速度のプログラム処理」機能はたとえば、PLC ユーザープログラム、NC の外部信号やその他のチャネルにより、チャネルで選択された NC プログラムの協調をテストするために使用します。NC プログラムは、プログラムされた命令、機能、および移動速度に正確に対応して実行されます。テスト中のプログラム処理時間を短縮するために、ドライブラン送り速度を起動することにより移動操作を高速に実行できます。たとえば、G01、G02、G03、G33、G34、G35、G95 では、プログラムされた送り速度の代わりにテスト送り速度が適用されます。

ドライラン送り速度は、AUTOMATIC モードおよび MDAP モードでユーザーインタフェースにより選択し、自動操作の中断時またはブロック終点で起動できます。この起動は、NC/PLC インタフェース経由で PLC ユーザープログラムから実行できます。

送り速度に及ぼす影響について詳しくは、「V1:送り速度 (ページ 1693)」の章を参照してください。

通知
過度の切削速度による工具またはワークの破壊 「ドライラン送り速度によるプログラム処理」機能を有効にしてワークを加工した場合、許容範囲外の切削速度が得られることがあります。この高い速度によって、工具やワークが破壊される場合があります。

パラメータ設定

ドライラン送り速度

チャネルで有効なドライラン送り速度は以下を使用して設定します。

SD42100 $SC_DRY_RUN_FEED = <ドライラン送り速度>

注記

毎回転送り速度

ドライラン送り速度は、G95 によって、プログラムブロックにプログラム指令された毎回転送り速度も置き換えます。

ドライラン設定

「ドライラン送り速度によるプログラム処理」で有効な送り速度の選択条件は、以下を使用して設定します。

SD42101 $SC_DRY_RUN_FEED_MODE = <値>

<値>	意味
0	SD42100 の最大値およびプログラムされた送り速度は、ドライラン送り速度として有効です。
1	SD42100 の最小値およびプログラムされた送り速度は、ドライラン送り速度として有効です。
2	SD42100 は、ドライラン送り速度として有効です。
10	値 0 の場合。ねじ切り(G33、G34、G35)とタッピング(G331、G332、G63)を除く。この機能では、プログラムされた送り速度が適用されます。
11	値 1 の場合。ねじ切り(G33、G34、G35)とタッピング(G331、G332、G63)を除く。この機能では、プログラムされた送り速度が適用されます。
12	値 2 の場合。ねじ切り(G33、G34、G35)とタッピング(G331、G332、G63)を除く。この機能では、プログラムされた送り速度が適用されます。

起動

選択

この機能は、ユーザーインタフェースで操作エリア[機械|運転モード| AUTOMATIC または MDAP |プログラム制御|ドライラン送り速度]により選択します。

PLC ユーザープログラムでこの機能を起動することを要求するには、次の NC/PLC インタフェース信号を設定します。

DB21、... DBX24.6 = 1 (選択されたドライラン送り速度)

起動

NC でこの機能を起動することを要求するには、PLC ユーザープログラムで次の NC/PLC インタフェース信号を設定する必要があります。

DB21、... DBX0.6 = 1 (ドライラン送り速度の起動)

フィードバック信号

DRY は、NC で機能が有効である旨を機械オペレータに通知するフィードバック信号として、ユーザーインタフェースで「機械」操作エリアの状態に表示されます。

参照先

「ドライラン送り速度のプログラム処理」機能の使用に関する詳細については、以下を参照してください。

- 操作マニュアル、旋削
- 操作マニュアル、フライス加工

箇条書き:プログラム制御、ドライラン送り速度、DRY

10.5.4 パートプログラムブロックのスキップ

機能

新しいプログラムのテストや慣らしの際、プログラムの実行中に特定のパートプログラムブロックを無効にしたりスキップすることができれば便利です。 そのためには、スキップしたいブロックの最初にスラッシュを付けてください。

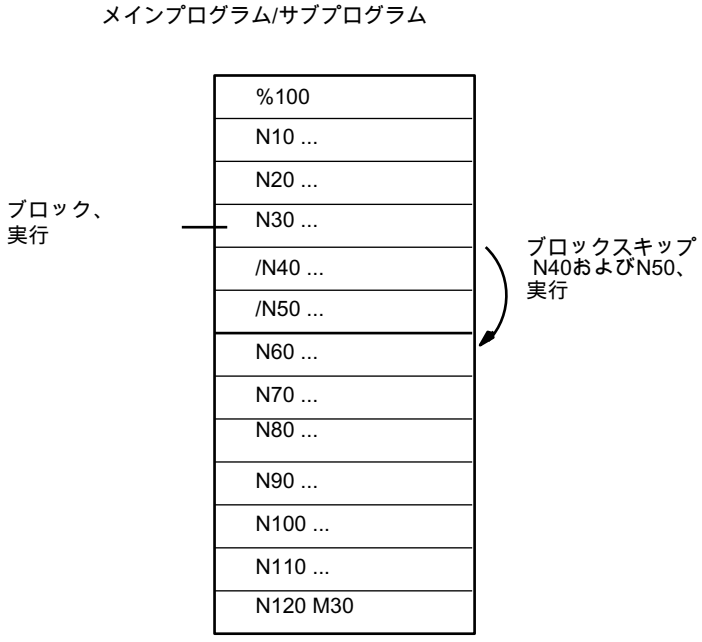


図 10-1 パートプログラムブロックのスキップ

選択

この機能は、[プログラム制御]メニューで操作画面を介して選択できます。

この選択時に、次のインタフェース信号が設定されます。

DB21、... DBX26.0 (ブロックスキップ選択中)

しかし、これによって機能が有効になるわけではありません。

起動

この機能は、次のインタフェース信号を介して有効になります。

DB21、... DBX2.0 (ブロックスキップの起動)

注記
ブロックサーチ中、「パートプログラムのスキップ」機能は有効なままです。

表示

「ブロックスキップ」機能が有効であれば、操作画面上の関連する欄が反転表示されます。

10.6 ワークシミュレーション

機能

実際のパートプログラムが工具シミュレーションで完全に計算され、その結果が操作画面にグラフで表示されます。.プログラミングの結果を、機械軸を移動せずに確認することができます。間違ってプログラム指令された加工ステップが早期に検出され、ワークが誤って加工されるのを防止できます。

シミュレーション NC

シミュレーションでは独自の NC インスタンス(シミュレーション NC)を使用します。そのため、シミュレーションを開始する前に、実際の NC をシミュレーション NC に合わせる必要があります。この調整により、すべての有効なマシンデータが NC から読み出され、シミュレーション NC に読み込まれます。NC とサイクルマシンデータは、有効なマシンデータに含まれています。

シミュレーションのコンパイルサイクル(840D sl のみ)

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 以降は、選択されたコンパイルサイクル(CC)だけがワークシミュレーション用にサポートされています。サポートされているコンパイルサイクルのマシンデータは、コントローラの電源投入後に調整されます。「シミュレーションの開始」による調整はおこなわれません。

注記

パートプログラムでは、CC 用の言語命令とサポートされていない CC のマシンデータは使用**できません**(「パートプログラム内の CC 命令」の節も参照してください)。
サポートされていない CC の特殊な移動(OEM 座標変換)は - 特定の状況下では - 正しく表示されません。

パートプログラム内の CC 命令

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:

プログラムコード	コメント
N1 G01 X200 F500	
IF (1==$P_SIM)	
N5 X300	; CC がシミュレーションに対して有効ではありません。
ELSE	
N5 X300 OMA1=10	
ENDIF	

10.7 ブロック検索、タイプ 1、2、および 4:

機能

ブロック検索を利用すると、ほぼすべてのパートプログラムブロックからパートプログラムの実行を開始できます。

この処理では、ブロック検索中に選択されたターゲットブロックまで NC がパートプログラム(移動動作のない)の内部処理を迅速に行います。そのとき、手動介入を最小限に抑えてターゲットブロックから自動パートプログラム実行を再開するために、通常のパ

ートプログラム実行中のターゲットブロックにおいて生じる(たとえば、軸位置、主軸速度、ロードされた工具、NC/PLC インタフェース信号、変数値に関する)状態と正確に同じ制御状態に達するためにあらゆる努力が払われます。

ブロック検索のタイプ

- **タイプ 1:計算なしのブロック検索**
  計算なしのブロック検索では、パートプログラムブロックを最も速く検索できます。いかなるタイプの計算も実行されません。ターゲットブロックにおける制御状態は、ブロック検索開始前の状態と変わりません。
- **タイプ 2:輪郭における計算ありのブロック検索**
  輪郭における計算ありのブロック検索では、どのような状況でも、プログラム指令輪郭にアプローチできます。NC スタートで、ターゲットブロックの開始位置、またはターゲットブロックの直前のブロックの終点にアプローチされます。これは終点まで移動します。処理は輪郭に一致します。
- **タイプ 4:ブロック終点における計算付ありロックサーチ**
  ブロック終点における計算ありのブロック検索では、どのような状況でも目標位置(工具交換位置など)にアプローチできます。ターゲットブロックで有効な補間タイプを使用して、ターゲットブロックの終点またはその次のプログラム指令位置にアプローチします。これは輪郭とは一致しません。
  ターゲットブロックでプログラム指令軸だけが移動します。必要であれば、自動パートプログラム実行を開始する前に、衝突のない初期状況を「JOG REPOS」モードにより機械上に手動で作り出さなければなりません。
- **タイプ 5:「プログラムテスト」モードにおける計算ありのブロック検索(SERUPRO)**
  SERUPRO (**se**arch **ru**n by **pro**gramtest (プログラムテストによる検索実行))とは、計算ありのチャネル間ブロック検索です。ここでは、選択されたパートプログラムはNC によって「プログラムテスト」モードで開始されます。ターゲットブロックに到達すると、プログラムテストは自動的に解除されます。このタイプのブロック検索では、ブロック検索がおこなわれているチャネルと、シンクロナイズドアクションおよび他の NC チャネルとの相互作用も可能です。
  「ブロック検索タイプ 5 (SERUPRO) (ページ 615)」の章を参照してください。

注記

ブロック検索についての詳細は、「ブロック検索中の動作 (ページ 527)」の章を参照してください。

後続の処理

ブロック検索の完了後、次のような処理がおこなわれることがあります。

- タイプ 1 ～タイプ 5:ASUP の自動起動
  最後のアクションブロックが起動されたとき、ユーザープログラムを ASUP として起動できます。
- タイプ 1 ～タイプ 4:多段ブロック検索
  別のターゲットを指定して次のブロック検索を「検索ターゲット発見」から開始できます。

10.7.1 機能の説明

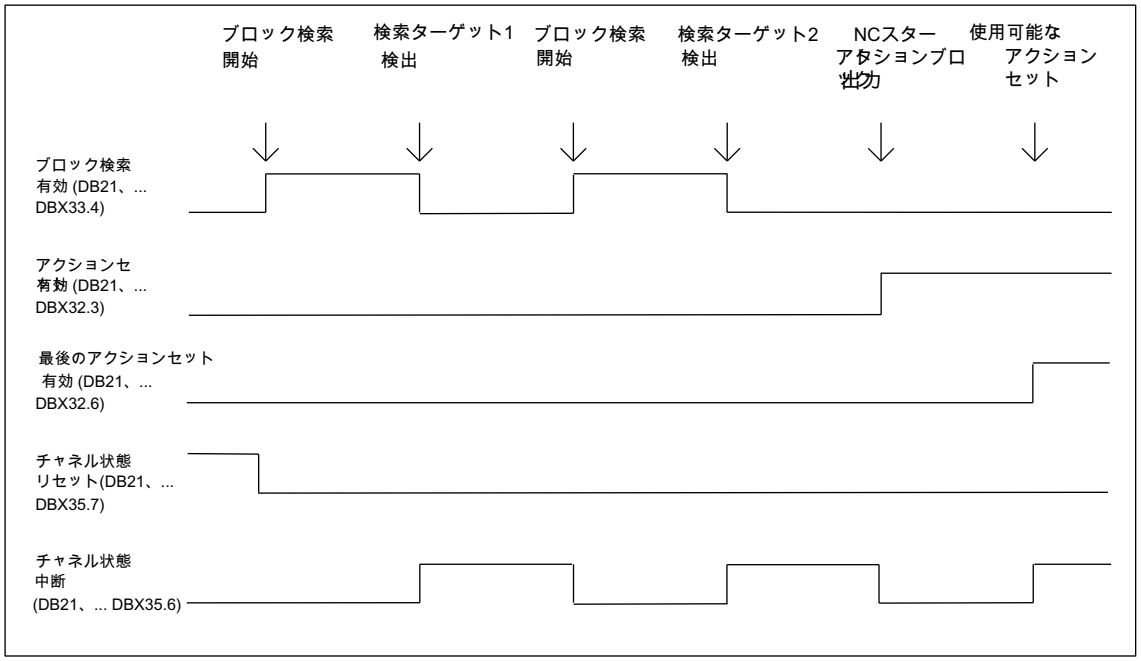
タイプ 2 またはタイプ 4 の基本シーケンス

1. User:操作画面を介したブロック検索**タイプ 2** または**タイプ 4** (...への**計算ありの**ブロック検索)の起動
2. 補助機能の収集によるターゲットブロックの検索
3. 「検索ターゲット検出」後の停止⇒ アラーム 10208「NC スタートによるプログラムの続行」の表示
4. User:NC スタートによるアクションブロックの実行⇒ DB21、... DBX7.1 = 1 (NC スタート)
5. アクションブロックの実行
6. 最後のアクションブロックが起動する⇒ ASUB としての/_N_CMA_DIR/_N_PROG_EVENT_SPF (初期設定)の自動起動
7. 最後の ASUB ブロック(REPOSA)が起動する⇒ DB21、... DBX32.6 = 1 (最後のアクションブロックが有効)
8. **オプション**：PLC ユーザープログラムを介したユーザー別必要条件の実行
9. アラーム 10208「NC スタートによるプログラムの続行」の表示
10. User:NC スタートによるプログラムの続行⇒ DB21、... DBX7.1 = 1 (NC スタート)

検索ターゲットが見つからない

検索ターゲットが見つからない場合は、アラーム 15370「ブロック検索時に検索ターゲットが見つからない」が表示され、ブロック検索がキャンセルされます。

タイミング図



アクションブロック

ブロック検索**タイプ 2** または**タイプ 4** (...への**計算ありの**ブロック検索)時に、工具(T、D)プログラミング、主軸(S)プログラミング、送り速度プログラミング、または M 機能の出力が収集されます。NC スタートを使用してアクションブロックを実行すると、収集された動作が PLC に出力されます。

注記

アクションブロックに NC スタートを使用すると、ブロック検索**タイプ 2** または**タイプ 4** (...への**計算ありの**ブロック検索) (S 値、M3/M4/M5/M19、SPOS)時に収集された主軸プログラミングが有効になります。

ユーザーは PLC ユーザープログラムで、工具が動作可能であること、または主軸プログラミングがリセットされているか、出力されていないことを確認しなければなりません。

DB31、... DBX2.2 = 1 (主軸リセット)

必要条件

ブロック検索タイプ 4 後の継続モード

軸の初期プログラミングをブロック検索タイプ 4 (ブロックの終点への計算ありのブロック検索)に対してインクリメンタルで実行する場合は、プログラムしたインクリメンタル

値を、検索ターゲットまで、または軸の現在のフィードバック値まで収集された位置の値に追加できます。以下を使用して設定します。

SD42444 $SC_TARGET_BLOCK_INCR_PROG

セッティングデータは NC スタートによって、アクションブロックを出力するために使用されます。

(Single BLock)

機能「シングルブロック」(DB21、... DBX0.4 == 1 (シングルブロック起動)が有効な間に、ブロック検索タイプ 2 またはタイプ 4(...への**計算ありの**ブロック検索)時に検索ターゲットの後でアクションブロックが見つかるたびに停止しないようにするには、この動作を以下のマシンデータで無効にすることができます。

MD10702 $MN_IGNORE_SINGLEBLOCK_MASK、ビット 3 = 1 (アクションブロックの信号ブロックを無視)

インターフェース 信号「開始ブロック有効」

インターフェース信号は、ブロック検索**タイプ 2 (輪郭への計算ありの**ブロック検索)時に**のみ**設定されます。

- DB21、... DBX32.4 = 1 (アプローチブロック有効)

ブロック検索**タイプ 4 (ブロックの終点への計算ありの**ブロック検索)では、アプローチブロックが生成されないため、インターフェース信号が設定されません(ターゲットブロックに等しいアプローチブロック)。

ターゲットブロックの補間のタイプ

ブロック検索**タイプ 4 (ブロックの終点への計算ありの**ブロック検索)では、ターゲットブロックで有効な補間タイプで開始動作がおこなわれます。直線補間(G0 または G1)以外の補間タイプでは、アラーム(円弧補間 G2 または G3 における円弧終点の異常など)で開始動作をキャンセルできます。

10.7.2 ブロックサーチの他の NCK 機能への関係

10.7.2.1 ブロックサーチ中およびブロックサーチ後の ASUB

ブロック検索タイプ 2 およびタイプ 4:チャネル軸の同期

ブロック検索タイプ 2 またはタイプ 4 (...への計算ありのブロック検索)後に ASUB を開始した場合は、すべてのチャネル軸の実位置が原点同期されます。

効果

以下のシステム変数が ASUB で読み取られる場合は、以下の値が含まれます。

- $P_EP:WCS でのチャネル軸の現在の実位置
- $AC_RETPOINT:WCS でのチャネル軸の収集済みブロック検索位置

ブロック検索タイプ 2:ASUB の完了

ブロック検索タイプ 2 (輪郭への計算ありのブロック検索)では、ASUP の完了によって、以下の REPOSA 命令(輪郭への再アプローチ、直線、すべてのチャネル軸)がプログラムされなければなりません。

効果

- すべてのチャネル軸が、ブロックサーチ中に収集されたサーチ位置に移動します。
- $P_EP == 「チャネル軸の収集されたブロック検索位置(WCS)」

ブロック検索タイプ 4:REPOS 動作

ブロック検索タイプ 4 (ブロックの終点への計算ありのブロック検索)では、以下の REPOS 命令による**開始**と**終了**で記述される期間に自動再位置決めがトリガされることはありません。

- **開始**:NC/PLC インターフェース信号 DB21、... DBX32.6 == 1(最後のアクションブロック有効)
- **終了**:NC スタートによるプログラム実行の継続

アプローチ動作の開始点は、NC スタート命令時のチャネル軸の現在位置です。終点は、パートプログラムでプログラムされた他の移動動作によって決まります。

ブロックサーチタイプ 4 では、NC によるアプローチ動作は生成されません。

動作:

- ASUB の終了後は、ASUB または手動(モード: JOG)でチャネル軸が位置決めされた実位置がシステム変数$P_EP(プログラム指令された終了位置)に含まれます JOG)。
  $P_EP ==「WCS でのチャネル軸の現在の実位置」

10.7.2.2 ブロックサーチ後の PLC 動作

すべてのアクションブロックが NC によって実行されており、PLC による操作(工具交換をおこなうための ASUB の開始など)またはオペレータの操作(復元など)が可能である場合は、以下のチャネル別 NC/PLC インターフェース信号が設定されます。

- DB21、... DB32.6 = 1(最後のアクションブロック有効)

アラーム出力時間のパラメータ設定

プログラム実行を継続するためにチャネルで NC スタートが必要であることをオペレータに知らせるために、アラーム 10208「NC スタートによるプログラムの続行」が表示されます。

アラームを表示するタイミングは、以下のマシンデータで設定できます。
MD11450 $MN_SEARCH_RUN_MODE、ビット 0 = <値>

<値>	意味
0	ブロックサーチ後に最後のアクションブロックが変更されると、次のことが発生します。 ● パートプログラムの実行が停止されます。 ● DB21、... DBB32.6 = 1 (最後のアクションブロック有効) ● アラーム 10208 を表示
1	ブロックサーチ後に最後のアクションブロックが変更されると、次のことが発生します。 ● パートプログラムの実行が停止されます。 ● DB21、... DBB32.6 = 1 (最後のアクションブロック有効) ● DB21、... DBX1.6 == 1 (PLC 動作終了)の場合にのみアラーム 10208 を表示

アラームとともに、次のインターフェース信号が設定されます。

- DB21、... DBX36.7 = 1 (加工の停止状態による NC アラームが未処置)
- DB21、... DBX36.6 = 1 (チャネル別 NC アラームが未処置)

10.7.2.3 ブロック検索後の主軸機能

コントロールシステムの応答と出力

ブロック検索中に収集された主軸別補助機能が自動的にアクションブロック内の PLC に出力されるか、後でユーザー別に出力されるかを、以下で設定できます。

MD11450 $MN_SEARCH_RUN_MODE、ビット 2 = <値>

<値>	意味
0	アクションブロックでブロック検索中に収集された主軸補助機能(M3、M4、M5、M19、M70)の出力。
1	ブロック検索中に収集された主軸別補助機能は、アクションブロックには出力されません。 後で、ASUB などに出力できます。収集された主軸別補助機能はシステム変数に格納されます。

システム変数

ブロック検索中に収集された主軸別補助機能は、以下のシステム変数に格納されます。

システム変数	説明
$P_SEARCH_S [<n>]	最後にプログラムされた主軸速度または切削速度
$P_SEARCH_SDIR [<n>]	最後にプログラムされた主軸方向
$P_SEARCH_SGEAR [<n>]	最後にプログラムされた M 機能のギヤ選択
$P_SEARCH_SMODE [<n>]	最後にプログラムされた主軸モード
$P_SEARCH_SPOS [<n>]	M19、SPOS、または SPOSA を使用して最後にプログラムされた主軸位置または移動軌跡
$P_SEARCH_SPOSMODE [<n>]	M19、SPOS、または SPOSA を使用して最後にプログラムされた位置アプローチモード
<n>:主軸番号	

主軸別の補助機能を後で出力する場合、たとえば、ASUB やアクションブロック出力後の出力においてシステム変数を読み取ることができます。

DB21、... DBX32.6 == 1 (最後のアクションブロック有効)

注記

システム変数$P_S、$P_DIR と$P_SGEAR の内容は、同期制御処理によりブロック検索後に失われることがあります。

ASUB、ブロック検索、およびアクションブロックについては、「主軸用補助機能の出力抑制 (ページ 532)」と「プログラムテスト (ページ 589)」の章を参照してください。

10.7.2.4 ブロック検索のためのシステム変数の読み取り

パートプログラムでは、以下のシステム変数を介して処理、メインラン、またはサーボ/ドライブの領域から値を読み取ることができます。

$P_...	プログラムされた値を含む、先読み関連のシステム変数
$A_...	現在値を含む、メインラン関連のシステム変数
$V_...	現在値を含む、サーボ/ドライブ関連のシステム変数

タイプ 2 およびタイプ 4 のブロック検索(...への**計算ありの**ブロック検索)中にはメインランにブロックが入力されないため、メインランおよびサーボ/ドライブに関連するシステム変数は、ブロック検索中には変更されません。必要な場合は、これらのシステム変数について、$P_SEARCH (ブロック検索が有効)でブロック検索が有効かどうかを確認して、NC プログラムで特別にブロック検索を処理してください。

先読み関連のシステム変数は、すべての検索タイプで正しい値を表示します。

10.7.3 ブロックサーチ後の ASUB の自動起動

パラメータ設定

機能の起動

ブロック検索後の ASUB 自動起動は、次のマシンデータ設定によって有効になります。

MD11450 $MN_SEARCH_RUN_MODE、ビット 1 = 1

起動するプログラム

初期設定では、ブロック検索後に最後のアクションブロックの変化により、プログラム **_N_PROG_EVENT_SPF** が、ディレクトリ_N_CMA_DIR から ASUB として起動されます。別のプログラムを起動する必要がある場合は、このユーザープログラムの名称を次のマシンデータに入力してください。

MD11620 $MN_PROG_EVENT_NAME

シングルブロック処理が設定されている場合の動作

次のチャネル別マシンデータを使用して、起動した ASUB を、シングルブロック処理が設定されていても中断なしで処理するかどうか、またはシングルブロック処理を起動するかどうかを設定できます。

MD20106 $MC_PROG_EVENT_IGN_SINGLEBLOCK、ビット 4 = <値>

<値>	意味
0	シングルブロック処理が有効です。
1	シングルブロック処理がマスクされます。

読み込み停止が設定されている場合の動作

チャネル別マシンデータを使用して 、読み込み停止(DB21、... DBX6.1 = 1)が設定されていても ASUB を中断なしで処理するかどうか、または読み込み停止を有効にするかどうかを設定します。

MD20107 $MC_PROG_EVENT_IGN_INHIBIT、ビット 4 = <値>

<値>	意味
0	読み込み停止が有効です。
1	読み込み停止がマスクされます。

注記

MD11620、MD20108、および MD20107 のパラメータ設定の詳細については、「パラメータ設定 (ページ 683)」の章を参照してください。

プログラミング

ASUB を開始したイベントは、システム変数$P_PROG_EVENT に格納されます。ブロック検索後に自動起動すると、$P_PROG_EVENT から値「5」が返されます。

処理

ブロック検索後の ASUB の自動起動の処理

1. User:操作画面を介したブロック検索**タイプ 2** または**タイプ 4** (...への**計算ありの**ブロック検索)の起動
2. 補助機能の収集によるターゲットブロックの検索
3. 「検索ターゲット検出」後の停止⇒ アラーム 10208「NC スタートによるプログラムの続行」の表示
4. User:NC スタートによるアクションブロックの実行⇒ DB21、... DBX7.1 = 1 (NC スタート)
5. アクションブロックの実行
6. 最後のアクションブロックが起動する⇒ ASUB としての/_N_CMA_DIR/_N_PROG_EVENT_SPF (初期設定)の自動起動

7. 最後の ASUB ブロック(REPOSA)が起動する⇒ DB21、... DBX32.6 = 1 (最後のアクションブロックが有効)
8. **オプション**：PLC ユーザープログラムを介したユーザー別必要条件の実行
9. アラーム 10208「NC スタートによるプログラムの続行」の表示

注記

MD11450 $MN_SEARCH_RUN_MODE、ビット 0 == 1 では、アラーム 10208 が、PLC ユーザープログラムにより有効にされた(DB21、... DBX1.6 = 1 (PLC 操作終了)後にのみ出力されます。

10. User:NC スタートによるプログラムの続行⇒ DB21、... DBX7.1 = 1 (NC スタート)

10.7.4 多段ブロックサーチ

機能

「多段ブロックサーチ」機能を使用して、「検索ターゲット発見」状態から別のブロックサーチを開始できます。多段処理は、到達した各検索ターゲットの後で何度でも続行することができ、次のブロックサーチ機能に適用されます。

- タイプ 1: 計算なしのブロックサーチ
- タイプ 2: 輪郭における計算ありのブロック検索
- タイプ 3: ブロック終点における計算ありのブロック検索

注記

検索ターゲットが発見された場合のみ、停止されたプログラム実行から別の「多段ブロックサーチ」を開始できます。

起動

「多段ブロックサーチ」は、既存のマシンデータ
MD11450 $MN_SEARCH_RUN_MODE で設定します

- ビット 3 = 0 (FALSE)では、多段ブロックサーチが有効です
  (つまり、複数の検索ターゲットを指定できます)。
- 互換性のため、ビット 3 = 1 (TRUE)により、多段ブロックサーチを無効にできます。
  多段ブロックサーチの初期設定は、ビット 3 = 0 です。

実行動作

検索ターゲット発見、ブロックサーチを再開

検索ターゲットに到達すると、プログラムの実行が停止し、検索ターゲットが現在のブロックとして表示されます。到達した各検索ターゲットの後で、新しいブロックサーチを何度でも繰り返すことができます。

検索ターゲットの指定の変更

ブロックサーチを開始する前に、検索ターゲット指定とブロックサーチ機能を変更できます。

例:多段ブロックサーチの実行シーケンス

- リセット
- 検索ターゲット 1 までのブロックサーチ
- 検索ターゲット 2 までのブロックサーチ → 「多段ブロックサーチ」
- アクションブロックを出力するための NC スタート → アラーム 10208
- NC スタート → プログラム実行の継続

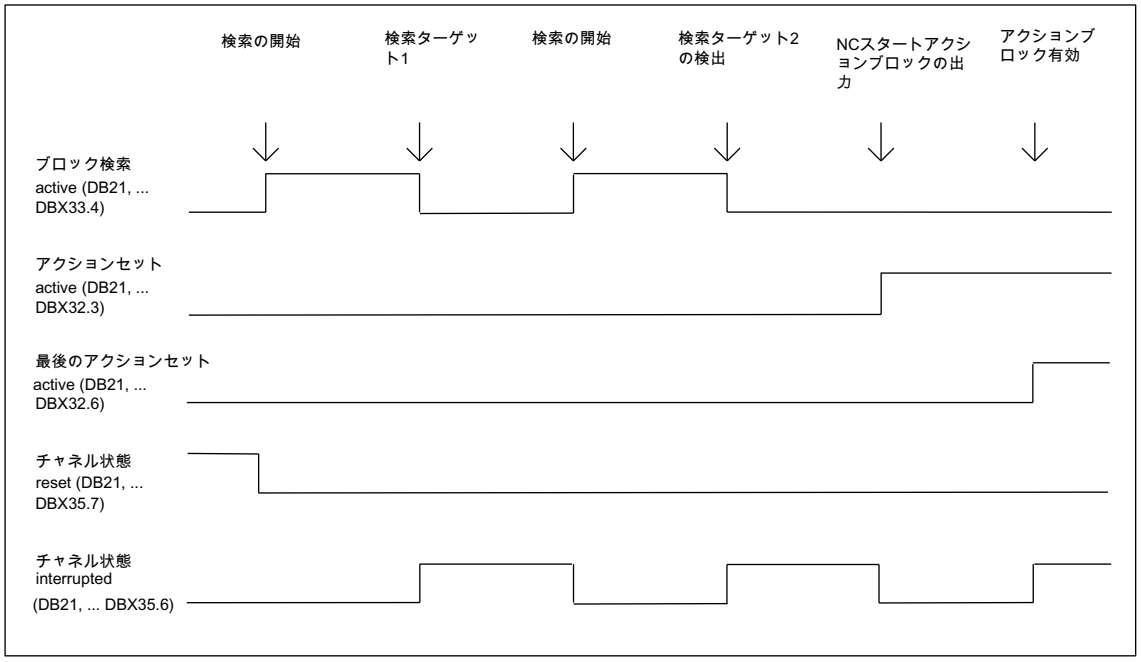


図 10-2　インタフェース信号の時間経過

10.7.5 計算ありのブロック検索の例

選択

次の例から、作業に適したブロック検索のタイプを選択してください。

タイプ 4: ブロック終点における計算ありのブロック検索

有効な工具管理機能を使用したブロック検索後の自動工具交換の例:

1. マシンデータの設定:
   MD11450 $MN_ SEARCH_RUN_MODE を 1 に設定します。
   MD11602 $MN_ASUP_START_MASK ビット 0 = 1 (停止状態からの ASUB 起動)
2. FB4 を介して PLC から ASUP「BLOCK_SEARCH_END」を選択します(「P3: SINUMERIK 840D sl 用 PLC 基本プログラム (ページ 1043)」の章も参照してください)。
3. パートプログラム「WORKPIECE_1」をロードし、選択します。
4. ブロック終点(ブロック番号 N220)までブロック検索します。
5. HMI が「検索ターゲット発見」を通知します。
6. アクションブロックを出力するための NC スタート
7. PLC 信号:
   DB21...DB32.6 (最後のアクションブロック有効)、
   PLC が FC9 を介して ASUB「BLOCK_SEARCH_END」を起動します(「P3: SINUMERIK 840D sl 用 PLC 基本プログラム (ページ 1043)」の章を参照)。
8. ASUP の終了後に(定義する M 機能 M90 などを使用して評価可能。ブロック N1110 の例を参照してください)、PLC が信号
   DB21、... DBX1.6 (PLC 動作完了)を設定します。
   また、NC/PLC インタフェース信号:
   DB21-DB30 DBB318 ビット 0 (ASUP が停止)
   をスキャンすることもできます。
   その結果、アラーム 10208 が表示されます。つまり、この時点で、オペレータによる他の動作が可能になります。

9. 手動オペレータ動作(JOG、JOG-REPOS、オーバストア)
10. NC スタートによりパートプログラムを続行します。

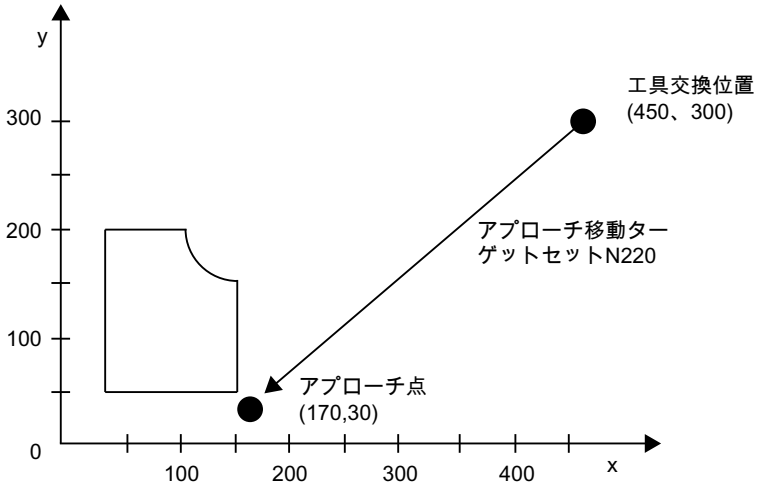


図 10-3　ブロック終点(ターゲットブロック N220)までブロック検索するためのアプローチ動作

注記

ターゲットブロック N220 での「輪郭のブロック検索」により、工具交換点(ターゲットブロックの開始点)へのアプローチ動作が生成されます。

タイプ 2: 輪郭における計算ありのブロック検索

有効な工具管理機能を使用したブロック検索後の自動工具交換の例:

1 ～ 3。　タイプ 4 のブロック検索例と同様
4.　輪郭の(ブロック番号 N260)のブロック検索
5 ～ 10。　タイプ 4 のブロック検索例と同様

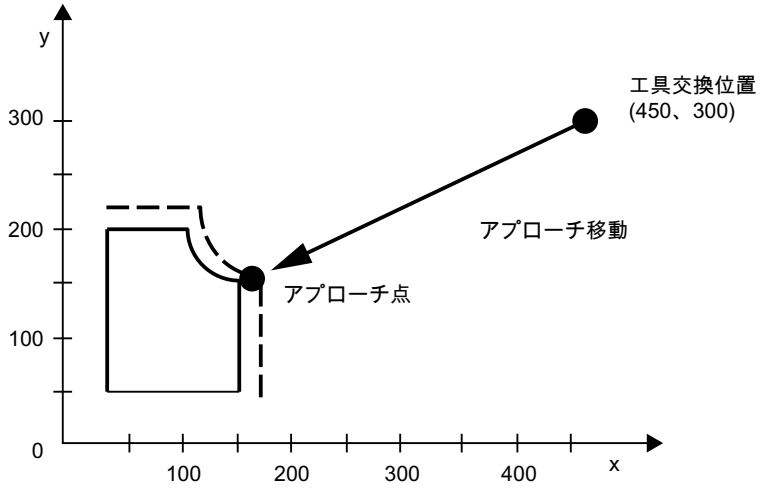


図 10-4　輪郭(ターゲットブロック N260)までブロック検索するためのアプローチ動作

注記

ターゲットブロック N260 の「ブロック終点までのブロック検索」の結果、アラーム 14040 (円弧終点異常)が出力されます。

タイプ 4 とタイプ 2 用のパートプログラム

PROC WORKPIECE_1

プログラムコード	コメント
; メインプログラム	
...	
;工具**"CUTTER_1"**を使用した輪郭部 1 の加工	
...	
N100 G0 G40 X200 Y200	; 工具径補正選択解除
N110 Z100 D0	; 工具長補正選択解除
;輪郭部 1 の終了	
;	
;工具**"CUTTER_2"**を使用した輪郭部 2 の加工	
N200 T="CUTTER_2"	; 工具の予備選択
N210 WZW	; 工具交換プログラムの呼び出し
N220 G0 X170 Y30 Z10 S3000 M3 D1	; 輪郭部 2 のアプローチブロック
N230 Z-5	; 切り込み
N240 G1 G64 G42 F500 X150 Y50	; 輪郭の開始点
N250 Y150	
N260 G2 J50 X100 Y200	
N270 G1 X50	
N280 Y50	

プログラムコード	コメント
N290 X150	
N300 G0 G40 G60 X170 Y30	; 工具径補正選択解除
N310 Z100 D0	; 工具長補正選択解除
輪郭部 2 の終了	
...	
M30	
PROC WZW	
;工具交換プログラム	
N500 DEF INT TNR_AKTIV	; 有効な T 番号用の変数
N510 DEF INT TNR_VORWAHL	; 事前選択された T 番号用の変数
N520 TNR_AKTIV = $TC_MPP6[9998,1]	; 有効な工具の T 番号の読み取り
N530 GETSELT(TNR_VORWAHL)	; 事前選択された工具の T 番号の読み取り
;	
;工具がまだ有効でない場合のみ工具交換を実行	
N540 IF TNR_AKTIV == TNR_VORWAHL GOTOF ENDE	
N550 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	; 工具交換点へのアプローチ
N560 M6	; 工具交換の実行
;	
END:M17	
PROC SUCHLAUF_ENDE SAVE	
ブロック検索後に工具交換プログラムを呼び出すための **ASUP**	
N1000 DEF INT TNR_AKTIV	; 有効な T 番号用の変数
N1010 DEF INT TNR_VORWAHL	; 事前選択された T 番号用の変数
N1020 DEF INT TNR_SUCHLAUF	; ブロック検索で決定された T 番号用の変数
N1030 TNR_AKTIV = $TC_MPP6[9998,1]	; 有効な工具の T 番号の読み取り
N1040 TNR_SUCHLAUF = $P_TOOLNO	; ブロック検索によって決定した T 番号の読み取り
N1050 GETSELT(TNR_VORWAHL)	; 事前選択された工具の T 番号の読み取り
N1060 IF TNR_AKTIV ==TNR_SUCHLAUF GOTOF ASUP_ENDE	
N1070 T = $TC_TP2[TNR_SUCHLAUF]	; 工具名称による T 選択
N1080 WZW	; 工具交換プログラムの呼び出し
N1090 IF TNR_VORWAHL == TNR_SUCHLAUF GOTOF ASUP_ENDE	
N1100 T = $TC_TP2[TNR_VORWAHL]	; 工具名称による T 事前選択のリストア
ASUP_ENDE:	
N1110 M90	; PLC へのフィードバック
N1120 REPOSA	; ASUP end

10.7.6 必要条件

10.7.6.1 コンプレッサ機能(COMPON、COMPCURV、COMPCAD)

- **タイプ 2** または**タイプ 4**(...の**計算による**ブロック検索)のブロック検索のターゲットブロックが、コンプレッサ機能(COMPON、COMPCURV、COMPCAD)が有効なプログラムセクションにある場合は、再位置決め時に**コンプレッサ**によって計算された軌跡上で位置にアプローチします。これらの位置は、パートプログラムで指令された軌跡上の位置と正確に一致するようにしてください。
- プログラムされたブロックが圧縮時にパートプログラムから削除された場合は、これらのブロックがブロック検索のターゲットブロックに見つかりません⇒アラーム 15370「検索ターゲットが見つからない」。

10.8 ブロック検索タイプ 5 (SERUPRO)

10.8.1 機能の説明

ブロック検索タイプ 5、「プログラムテスト」モードでの計算ありのブロック検索 (SERUPRO、「**Se**arch-**Ru**n by **Pro**gram test」)により、選択可能な中断点でチャネル間の計算ありのブロック検索が可能です。既存のプログラム協調命令を考慮して、中断チャネルでプログラムを続行するために必要なすべての状態データが SERUPRO 中に特定され、NC と PLC によりプログラムの続行を許可する状態に設定されます。

プログラムの実行を続行して再位置決めする前に、その後も必要となるすべての出力状態をユーザー専用の ASUP を介して自動的に生成できます。

チャネル

HMI との組み合わせで、SERUPRO が次のチャネル用に提供されています。

- 現在の SERUPRO チャネルのみ(1)
- SERUPRO チャネルと同じワーク名称を持つすべてのチャネル(2)
- SERUPRO チャネルと同じモードグループを持つすべてのチャネル(3)
- NCU のすべてのチャネル(4)

SERUPRO のチャネル範囲は、設定ファイル **maschine.ini** の[BlockSearch]のセクションを使用して選択されます。

セクション [BlockSearch]	**HMI 用のブロック検索を有効にし、ブロック検索設定を選択します。**
SeruproEnabled=1	;HMI で使用可能な SERUPRO ソフトキー。初期値は(1)です。
SeruproConfig=1	;上記のチャネルグループの番号(1)～(4)。初期値は(1)です。

SERUPRO によって開始される他のチャネルはすべて、「自動 SERUPRO」モードで動作します。ターゲットブロックが選択されているチャネルのみ、ブロック検索によって SERUPRO モードで開始できます。

起動

SERUPRO は、HMI を介して起動されます。SERUPRO は、[Prog.Test Contour]ソフトキーを使用して操作します。

SERUPRO では、REPOS を使用してターゲットブロックにアプローチします。

SERUPRO の動作順序

1. HMI の[Pog. test contour]ソフトキーを使用して検索ターゲットを操作します。
2. NC が、選択されたプログラムを「プログラムテスト」モードで自動的に起動します。
   - このモードでは、軸は移動しません。
   - 補助機能$A_OUT と直接 PLC IO が出力されます。
   - ターゲットブロックの補助機能は出力されません。
3. ターゲットブロックの最初で NC ストップは「プログラムテスト」を内部的に解除し、停止状態「待機: 検索ターゲットが見つかりました」を表示します。
4. ユーザー専用の ASUP「PROG_EVENT.SPF」が使用可能な場合は、これが自動的に開始されます。
5. 再位置決めは次の NC スタートでおこなわれます(REPOS)。
   REPOS 動作はシステム ASUP によって行われ、「可変 ASUP」機能を使用して拡張できます。

ブロック検索 SERUPRO に対する境界条件

SERUPRO 機能は、「AUTOMATIC」モードでのみ起動することができます。また、プログラム状態(チャネル状態 RESET)でのみ中止することができます。

通常モードで**のみ**、PLC が複数のチャネルを起動する場合、これは各チャネルで SERUPRO によってシミュレーションすることができます。

マシンデータ設定:
MD10708 $MN_SERUPRO_MASK、ビット 1 = 0、
:パートプログラム命令 START を使用すると、アラーム 16942「チャネル%1 起動プログラム命令アクション%2<ALNX>不可」
が出力されて、シミュレーションが中止されます。

マシンデータ:
MD10707 $MN_PROG_TEST_MASK
では、停止状態で停止することができ、SERUPRO 動作に影響しません。初期設定では、リセット状態でのみプログラムテストを無効にすることができます。

注記

プログラムテストが無効になった後、SERUPRO アプローチ動作と同じ制約を受ける REPOS 動作が開始されます。どのような悪影響も、ASUP を使用して防止できます。

SERUPRO の動作の制御

例として以下に挙げる機能では、SERUPRO の動作を NC 用に特別に設定できます。

- プログラムストップ (M0)
- プログラム協調命令 START
- グループ SERUPRO
- 複数のチャネルにわたる SERUPRO の終了
- オーバライド

MD10708 $MN_SERUPRO_MASK = <SERUPRO での動作>

SERUPRO のチャネル別初期設定

チャネル別初期設定は通常、パートプログラムの開始後に以下のマシンデータで指定されます。

MD20112 $MC_START_MODE_MASK= <初期設定>

SERUPRO に独自の初期設定を指定し、これを MD20112 の初期設定と置換することができます。

MD22620 $MN_START_MODE_MASK_PRT = <SERUPRO の初期設定>

SERUPRO の初期設定は、以下により明示的に設定してください。

MD22621 $MC_ENABLE_START_MODE_MASK_PRT = 1

NC/PLC インタフェース信号「プログラムテストを介したブロック検索有効」

プログラムテストを介したブロック検索は、NC/PLC インタフェース信号を使用して表示されます:
DB21、... DBX318.1 == 1

インタフェース信号は、ブロック検索の開始からターゲットブロックがメインラインに挿入されるまで設定されます。

SERUPRO 動作後のユーザー定義の ASUP の場合

注記

工作機械メーカが、上記の 7 項で説明したように SERUPRO 動作後に ASUP を起動すると決定した場合、以下の条件を守る必要があります。

6 項に対応した停止状態:

マシンデータ:
MD11602 $MN_ASUP_START_MASK
と
MD11604 $MN_ASUP_START_PRIO_LEVEL
を使用すると、NC が FC9 ブロックを介して ASUP を停止状態から自動的に起動できます。

REPOS ブロックの完了後にのみ可能な FC9 の応答:

REPOS ブロックも完了している場合にのみ、ASUP は「ASUP 完了」によって、FC9 ブロックから完了した旨の通知を受けることができます。

8 項の後における、割り当てられた REPOS 動作の解除:

ASUP の起動によって、割り当てられた REPOS 動作が解除されます。

したがって、REPOS 動作を保持するには、ASUP を REPOSA で終了しなければなりません。

不要な REPOS 動作の削除:

不要な REPOS 動作は、`M17` または `RET` で ASUP を完了することで削除されます。

ASUP の特別扱い:

原則として、REPOS で終了し停止状態から起動される ASUP は、特別な扱いを受けます。

ASUP が REPOS ブロックの前に自動的に停止し、そのことを以下によって示します。

DB21、... DBX318.0 (ASUB 停止)

ASUP の自動起動

パス:
/_N_CMA_DIR/_N_PROG_EVENT_SPF
における ASUP が、マシンデータ:
MD11450 $MN_SEARCH_RUN_MODE、ビット 1 = 1
によって SERUPRO アプローチで自動的に起動されます。この順序は次のとおりです。

1. SERUPRO 動作が完全におこなわれます。
2. ユーザーが「NC スタート」を押します。
3. ASUP が起動します。
4. REPOS パートプログラム命令の**前**に NC が自動的に停止し、
   メッセージ「NC スタートを押してプログラムを続行してください」が表示されます。
5. ユーザーが再度「NC スタート」を押します。
6. NC が再位置決め移動を実行し、ターゲットブロックでパートプログラムを続行します。

注記

MD11450 による ASUP 自動起動では、**「起動」**でプログラムを続行しなければなりません。
手順は、この点で他のブロック検索タイプと似ています。

10.8.2 輪郭への再位置決め(REPOS)

「輪郭への再位置決め」(REPOS)機能を使用すると、中断位置から中断された加工を再開できます。REPOS とは異なり、SERUPRO では、プログラムセクションの「再取得」や「繰り返し」が許可されています。このために、SERUPRO がターゲットブロックを発見すると、REPOS モードで選択された位置に輪郭が位置決めされ、加工が再開されます。

SERUPRO:REPOS 応答の設定

REPOS 動作、つまり、再位置決めブロックでの動作は、次のマシンデータを使用して設定します。
MD11470 $MN_REPOS_MODE_MASK = <REPOS モード>

<REPOS モード>		
ビット	値	意味
0	0	中断された遅延時間が繰り返されます
	1	中断された遅延時間が再開されます
1	-	予備
2	0	DB31、... DBX10.0 (REPOS 遅延)が再位置決め減速ブロックで考慮されません。
	1	DB31、... DBX10.0 (REPOS 遅延)が再位置決め減速ブロックで考慮されます。
3	0	SERUPRO:軌跡軸のみが再位置決めブロックで移動します
	1	SERUPRO:軌跡軸と位置決め軸が再位置決めブロックで同時に移動します
4	0	REPOS:軌跡軸のみが再位置決めブロックで移動します
	1	REPOS:軌跡軸と位置決め軸が再位置決めブロックで同時に移動します
5	0	中断中に、変更された送り速度および主軸速度は、中断位置の後の最初のパートプログラムブロックの後にしか機能しません
	1	中断中に変更された送り速度および主軸速度は、中断点以降に即時に有効になります。つまり、残るブロックですでに有効であり、優先度が与えられます。この動作は、各 REPOS 操作に関連しています。
6	0	SERUPRO:再位置決めブロックでは、中立軸と位置決め主軸が軌跡軸として移動します。
	1	SERUPRO:再位置決めブロックでは、中立軸と位置決め主軸がコマンド軸として移動します。 中立軸と位置決め主軸は、SERUPRO の後で再位置決めされます。 中立軸では、再位置決めできない場合に、REPOS の遅延を設定する必要があります。DB31、... DBX10.0 = 1 (REPOS 遅延)

<REPOS モード>		
ビット	値	意味
7	0	REPOS 遅延が有効では**ありません**。
	1	REPOS 遅延が有効です。 REPOS 遅延が有効な軸は(DB31、... DBX10.0 == 1)、ジオメトリ軸または旋回軸のいずれでもなく、再位置決め時に移動**しません**。

⚠ 注意

衝突の可能性

MD11470 $MN_REPOS_MODE_MASK、ビット 3 またはビット 4 = 1
使用者には、再位置決めブロックにおける軸の同時移動によって、機械で衝突が発生しないようにする責任があります。

制御された REPOS による再位置決め

REPOS モードは、次の NC/PLC インタフェースを介して軌跡軸について指定できます。

DB21、... DBX31.0 - 2 (REPOS モード)

REPOS モードは、NC プログラムでプログラムされ、アプローチ動作を定義します(「制御された REPOS による再位置決め (ページ 629)」の章を参照してください)。

個々の軸の REPOS 動作は、NC/PLC インタフェース信号を介して制御することもでき、次のマシンデータにより有効になります。
MD11470 $MN_REPOS_MODE_MASK BIT 2=1

軌跡軸に個別に作用することはできません。ジオメトリ軸でない他のすべての軸に対して、個々の軸の REPOS を一時的に停止して遅延させることもできます。NC/PLC インタフェース信号は、REPOS が移動したい個々のチャネル軸を後で再度有効にしたり、停止を継続したりするために使用できます。

⚠ 危険

衝突の可能性

信号 DB31, ... DBX2.2 (残移動距離削除)により、「個別軸の再位置決めの防止」機能の選択時に、次の危険が動作が発生します。

MD11470 $MN_REPOS_MODE_MASK ビット 2 == 1
中断後に軸がインクリメンタルでプログラム指令されているかぎり、NC は中断なしでアプローチする位置とは異なる位置に移動します。
次の例を参照してください。軸がインクリメンタルでプログラム指令されます。

例:回転軸 A がインクリメンタルでプログラム指令されます

回転軸 A は 4 番目の機械軸です。

- 軸 A は、REPOS 動作の前に 11°に位置決めされます。
  中断ブロック、つまり SERUPRO のターゲットブロックでは、回転軸 A は 27 °に移動する必要があります。
  以降の任意数のブロックで、回転軸 A は段階的に 5 度移動します。
  `N1010 POS[A]=IC(5) FA[A]=1000`
  インタフェース信号 DB34 DBX10.0 = 1 を設定する場合(REPOS 遅延)、回転軸 A は REPOS 動作で移動せず、`N1010` で 32 度に移動します。ユーザーは意識的に 11 度から 27 度までの移動に対応しなければならない可能性があります。

  ⚠ 危険

  干渉の可能性

  中断後、軸は段階的にプログラムされ、32 度ではなく 16 度に移動します。

- 軸の個別起動
  複数の軸による SERUPRO アプローチのための REPOS 動作が、以下により選択されます。
  MD11470 $MN_REPOS_MODE_MASK ビット 3 = 1
  NC は、**すべての**位置決め軸をプログラム指令された終点に移動し、軌跡軸をターゲットブロックに移動するブロックにより、SERUPRO アプローチを開始します。
  ユーザーが適切な送り速度有効を選択することにより、個々の軸が起動されます。そして、ターゲットブロックが実行されます。

- 再位置決めブロックでの位置決め軸の再位置決め
  位置決め軸は残りのブロックでは再位置決めされず、むしろ再位置決めブロックで再位置決めされ、その効果は SERUPRO アプローチでのプログラムテストを介したブロック検索に限定されません。
  MD11470 $MN_REPOS_MODE_MASK ビット 3 = 1:プログラムテストを介したブロック検索用(SERUPRO)
  MD11470 $MN_REPOS_MODE_MASK ビット 4 = 1:各 REPOS 用

注記

ビット 3 もビット 4 も設定されない場合、この段階の残りのブロックでは非軌跡軸が再位置決めされます。

REPOS オフセットによる軸の遅延アプローチ

DB21、... DBX31.4 (REPOS モード変更)の立ち上がりエッジで、軸別のインタフェース信号 DB31、... DBX10.0 (REPOS 遅延)が設定されている場合、この軸の REPOS オフセットは、次回にプログラムされたときにのみ移動します。

この軸が現在 REPOS オフセットの対象であるかどうかを$AA_REPOS_DELAY によりシンクロナイズドアクションを介して確認できます。

⚠ 注意

衝突の可能性

DB31、... DBX10.0 (REPOS 遅延)は、軌跡を形成する機械軸に対して作用しません。

DB31、... DBX76.4 (軌跡軸)を使用して、軸が軌跡軸であるかどうかを下記の信号で確認できます。

REPOS 信号の受け入れタイミング

DB21、... DBX31.4 (REPOS モードの変更)の立ち上がりエッジにより、次の REPOS 信号が NC に伝送されます。

- チャネル別:DB21、... DBX31.0 - 2 (REPOS モード)
- 軸別:DB31、... DBX10.0 (再位置決め遅延)

REPOS の信号レベルは、現在のメインランブロックを示します。以下の 2 つの異なるケースがあります。

1. メインランには、現在有効な REPOS 操作の **1 つの**再位置決めブロックが含まれます。有効な REPOS 動作はキャンセルされて再起動します。REPOS オフセットは、上記の REPOS 信号により影響を受けます。
2. メインランに含まれる現在有効な REPOS 操作の再位置決めブロックは**ありません**。現在のメインプログラムブロックに再アプローチする将来の各 REPOS 動作は、上記の REPOS 信号によって影響を受けます。

注記

実行中の ASUP で、DB21、... DBX31.4 (REPOS モード変更)は、REPOS ブロックが実行されているタイミングでこの信号が不注意に設定されないかぎり、最後の REPOS には作用しません。

最初のケースでは、信号は停止状態でのみ許可されます。

RESET への応答:

- NC がすでに PLC 信号を確認している場合
  DB21、... DBX31.4 == 1 (REPOS モード変更) **および**
  DB21、... DBX319.0 == 1 (REPOS モード変更確認)
  この状況では、チャネルリセットの発生時に、有効な REPOS モードがクリアされます。
  DB21、... DBX319.1 - 3 = 0 (REPOS モード有効)
- NC がまだ PLC 信号を確認していない場合
  DB21、... DBX31.4 == 1 (REPOS モード変更) **および**
  DB21、... DBX319.0 == 0 (REPOS モード変更確認)
  この状況では、チャネルリセットの発生時に、REPOS モード変更確認および有効な REPOS モードがクリアされます。
  DB21, ... DBX319.0 = 0 (REPOS モード変更確認)
  DB21、... DBX319.1 - 3 = 0 (REPOS モード有効)

NC/PLC インタフェース信号による SERUPRO アプローチの制御

SERUPRO アプローチは、DB21、... DBX31.4 (REPOS モード変更)と関連する信号により、下記の段階で使用することができます。

- 「検索ターゲット発見」から「SERUPRO ASUB の起動」まで
- 「SERUPO-ASUB が REPOS の前に自動的に停止」から「ターゲットブロックを実行」まで

SERUPRO ASUB が実行されている間、たとえば、REPOS 操作の前のプログラムセクションでは、インターフェース信号は SERUPRO 位置決めに影響を与えません。

NC/PLC インタフェース信号による REPOS 動作

NC/PLC インタフェース信号による REPOS の制御

REPOS オフセットは、次の NC/PLC インタフェース信号により影響を受ける場合があります。

- DB21、... DBX31.0 - 2 (REPOS モード)
- DB21、... DBX31.4 (REPOS モード変更)

- DB31、... DBX10.0 (再位置決め遅延)
- DB31、... DBX72.0 (REPOS 遅延)

REPOS 確認応答信号

以下の NC/PLC インターフェース信号は、PLC を介して REPOS 動作を制御する機能への **NC からの**応答に使用できます。

- DB21、... DBX319.0 (REPOS モード変更の確認)
- DB21、... DBX319.1 - 3 (REPOS モード有効)
- DB21、... DBX319.5 (REPOS 遅延の確認)
- DB31、... DBX70.0 (REPOS オフセット)
- DB31、... DBX70.1 (REPOS オフセット有効)
- DB31、... DBX70.2 (REPOS 遅延の確認)
- DB31、... DBX76.4 (軌跡軸)

その他の情報については、「インタフェースでの REPOS オフセット」を参照してください。

REPOS 確認応答動作

NC が REPOS モードの変更(DB21、... DBX31.4 == 1)を認識した場合、DB21、... DBX319.0 = 1 で PLC により確認されます。

注記

NC がインタフェース信号 DB21、... DBX319.0 (REPOS モード変更確認)によりインタフェース信号 DB21、... DBX31.4 (REPOS モード変更)をまだ確認していない場合、この状態のチャネルリセットによりプログラムがキャンセルされ、REPOS モードの制御に使用される REPOS が実行されません。

PLC から発行された REPOS モードは、以下のインタフェース信号を使用して NC によって確認されます。

- DB21、... DBX319.1 - 3 (REPOS モード有効)
- DB31、... DBX10.0 (再位置決め遅延)
- DB31、... DBX70.2 (REPOS 遅延の確認)

例

- タイミング ②:ブロック N20 の NC プログラムは、NC ストップにより停止します。すべての軸は、停止状態まで、パラメータ設定された減速カーブに沿って制動されます。
- タイミング ③:PLC ユーザープログラムで「REPOS モード」が設定された後、NC は「REPOS モード変更」の 0/1 エッジにより REPOS モードを確定します。
- タイミング ④:ASUP が開始されるまで、「REPOS モード変更確認」は指令されたままです。
- タイミング ⑤:REPOS 動作は ASUP で開始します。
- タイミング ⑥:ASUP の残りのブロックが再ロードされます。

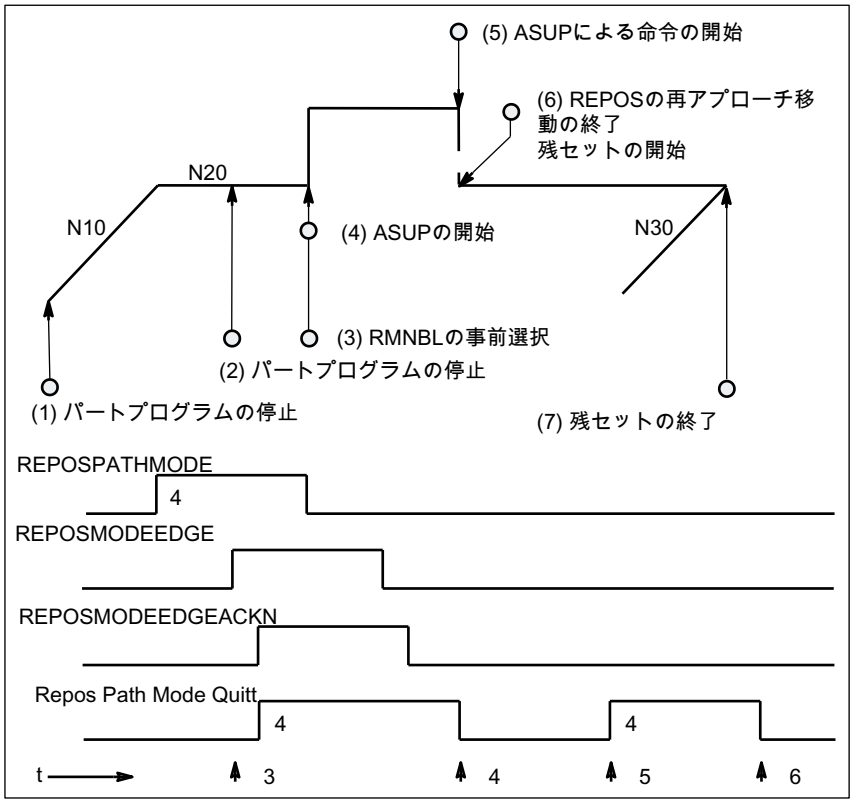


図 10-5　NC からの時間指定された応答信号によるパートプログラムの REPOS シーケンス

NC の応答の再設定

REPOSPATHMODE の段階がまだ有効です(→ 時間(2)で停止されたプログラムの残りのブロックはまだ完全に実行されません)。

ASUB の REPOS 再位置決め動作が実行されると直ちに、NC は再度「再位置決め軌跡モード応答」を設定します(→ 時間(5))。NC/PLC インタフェース信号を介して事前選択された REPOSPATHMODE がない場合、プログラム指令された REPOS モードが表示されます。

残りのブロックが起動されると、「再位置決め軌跡モード応答」がキャンセルされます(→ 時間(6))。
このブロックの後のパートプログラムブロック N30 → 時間(2)が再開されます。

インタフェース信号:
DB31、... DBX70.2 (REPOS 遅延確認)の定義も同様です。

DB31、... DBX70.1 (Repos オフセット有効) = 1、下記の場合:

DB21、... DBX319.1-319.3 (REPOS モード有効) = 4 (RMNBL).

有効な REPOS オフセット

SERUPRO 操作の最後に、次の軸/主軸 NC/PLC インタフェース信号(NC→PLC)を介して REPOS オフセットを読み取ることができます。
DB31、... DBX70.0 (REPOS オフセット)

この信号の関連する軸への効果は以下のとおりです。

値 0: 適用される REPOS オフセットはありません。
値 1: REPOS オフセットが適用されます。

有効範囲

インタフェース信号
DB31、... DBX70.0 (REPOS オフセット)
は、SERUPRO 操作の最後に供給されます。

REPOS オフセットは、SERUPRO ASUB の起動時または ASUB の自動起動時には無効になっています。

有効範囲内の REPOS オフセットの更新

SERUPRO の終点と SERUPRO の始点の間で、軸はモード変更して JOG モードで移動できます。

JOG モードでは、インタフェース信号
DB31、... DBX70.0 (REPOS オフセット)を値 0 に設定するために、軸を手動で REPOS オフセット軌跡上で移動します。

有効範囲内では、軸は FC18 を使用して移動することもできます。この FC18 によって、インターフェース信号 DB31
、... DBX70.0 (REPOS オフセット)は連続的に更新されます。

有効範囲の表示

REPOS オフセットの有効範囲は、以下のインタフェース信号で示されます。

DB31、... DBX70.1 (REPOS オフセット有効)

REPOS オフセット計算が有効なのか無効なのかが示されます。

値 0: この軸の REPOS オフセットが正しく計算されます。
値 1: REPOS がまだ存在しないので、この軸の REPOS オフセットは計算できません。たとえば、ASUB の最後であるか、有効な REPOS がありません。

軸入れ替え後の REPOS オフセット

グループ信号 DB21、... DBX319.5 (REPOS 遅延)は、有効な REPOS オフセットが実行されたかどうかの判断に使用できます。

値 0: このチャネルにより現在制御されているすべての軸に REPOS オフセットがないか、その REPOS オフセットが無効です。
値 1: その他。

主軸同期連結が同期された REPOS オフセット

SERUPRO により再位置決めしているとき、処理は中断時に継続されます。主軸同期連結がすでに同期されていた場合、追従主軸の REPOS オフセットと同期軌跡は存在しません。同期信号は設定されたままになります。

ブロック切り替えでの検索ターゲット発見

軸が軌跡グループの一員である場合、軸別の NC/PLC インタフェース信号 DB31、... DBX76.4 (軌跡軸)は 1 です。

この信号は、ブロック切り替え中に実行される現在のブロック状態を表します。この後の状態変更は無視されます。

SERUPRO 操作が「検索ターゲット発見」により終了される場合、DB31、... DBX76.4 (軌跡軸)はターゲットブロックを参照します。

10.8.2.1 制御された REPOS による再位置決め

中断されたプログラムの続行前にターゲットブロックの検索に SERUPRO が使用された場合、輪郭の再位置決めのために REPOS 動作が実行されます。REPOS モード「ターゲットブロック開始点の再位置決め」(RMBBL)は、初期設定で有効です。この REPOS モードは、次の NC/PLC インタフェースを介してユーザー別に定義できます。

DB21、... DBX31.0 - .2 (REPOS モード)

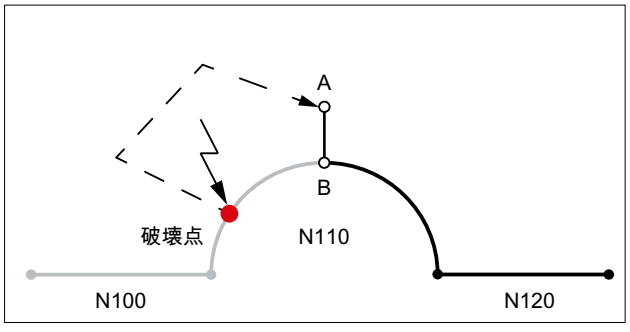
参照先:
このインタフェース信号の詳細については、『NC 変数およびインタフェース信号用パラメータマニュアル』を参照してください。

REPOS モード RMNBL 軌跡の次の点への再位置決め

RMNBL REPOS モードでは、位置決めは、REPOS 開始位置について次に最も近い輪郭点から行われます。

例

プログラムがブロック N110 内の任意の点で中断されます。その後、軸が位置(A)に、たとえば、手動で移動されました。SERUPRO がターゲットブロック N110 を発見すると、REPOS モード RMNBL による REPOS 操作が実行されます。REPOS 開始位置(A)について、点(B)は次に最も近い輪郭点です。点(B)に到達すると、REPOS 操作は完了します。中断されたプログラムのプログラムされた輪郭は、点(B)から再度移動されます。



NC/PLC インタフェースを介した REPOS モードの指定

REPOS モードは、次の NC/PLC インタフェース信号を介して指定できます。
DB21、... DBX31.0 - .2 (REPOS モード)

注記

RMNBL は一般の REPOS の拡張であり、SERUPRO に制限されません。

SERUPRO に対して、RMIBL と RMBBL の動作は同じです。

DB21、... DBX31.0 - .2 (REPOS モード)は、軌跡軸の移動動作にのみ影響を与えます。

他の軸の挙動は、インタフェース信号 DB31、... DBX10.0 (REPOSDELAY)を使用して個別に変更できます。この REPOS オフセットは直ちには適用されず、次にプログラムされたときのみ適用されます。

再位置決め点のプログラミングについて詳しくは、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、上級編; 軌跡動作、:「輪郭への再位置決め」の章

10.8.3 ブロック検索の加速

マシンデータの設定

SERUPRO 操作の実行速度には、以下のマシンデータを使用して影響を与えることができます。

MD22600 $MC_SERUPRO_SPEED_MODE = <値>

<値>	意味
0	ブロック検索速度/ドライラン送り速度でのプログラムテスト: ● 軸:MD22601 $MC_SERUPRO_SPEED_FACTOR *ドライラン送り速度 ● 主軸:MD22601 $MC_SERUPRO_SPEED_FACTOR *プログラムされた回転数 軸/主軸の動的な制限は**考慮されません**。
1	プログラムされた速度でのプログラムテスト: ● 軸:ドライラン送り速度 ● 主軸:プログラムされた回転数 軸/主軸の動的な制限が考慮されます。

<値>	意味
2	ドライラン送り速度でのプログラムテスト プログラムテストでは、プログラムされた速度/回転数で移動がおこなわれます。 軸/主軸の動的な制限が考慮されます。
3	ブロック検索速度でのプログラムテスト プログラムテストでは、以下の速度で移動がおこなわれます。 ● 軸:MD22601 $MC_SERUPRO_SPEED_FACTOR *プログラムされた送り速度 ● 主軸:MD22601 $MC_SERUPRO_SPEED_FACTOR *プログラムされた回転数 軸/主軸の動的な制限は**考慮されません**。 **注記** 毎回転送り速度が有効(G95 など)な場合は、プログラムされた送り速度が MD22601 $MC_SERUPRO_SPEED_FACTOR で積算されず、プログラムされた主軸速度でのみ積算されます。これにより、この場合も、MD22601 $MC_SERUPRO_SPEED_FACTOR により有効な軌跡速度が増加します。

必要条件

メイン軸

MD22600 $MC_SERUPRO_SPEED_MODE は、SERUPRO で、以下のメイン軸に機能します。

- PLC 軸
- コマンド軸
- 位置決め軸
- 揺動軸

シンクロナイズドアクション

通知
シンクロナイズドアクションの動作は SERUPRO では実行されない場合があります。 SERUPRO 時には、他の現在値(軸位置など)は内部的に生成され、SERUPRO による通常のプログラム実行時の値とは異なるため、現在値(軸位置など)をチェックするシンクロナイズドアクションの状態は、真(TRUE)であるとは認識されなくなります。したがって、シンクロナイズドアクションの動作部分は実行されません。

毎回転送り速度

ドライラン時の MD22601 $MC_SERUPRO_SPEED_FACTOR の効果:

- G95/G96/G961/G97/G971 から G94 への切り替え
- タッピングとねじ切り:通常のドライラン速度

フローティングチャックのないタッピング

- 「フローティングチャックのないタッピング」(G331/G332)を行うと、主軸が軌跡グループで位置制御されて補間されます。穴あけ深さ(直線軸)、ねじピッチ、および回転数(主軸)が定義されます。
  ドライラン時には、直線軸の速度が指定され、回転数が一定のままになり、ピッチが調整されます。
  SERUPRO の後は、主軸の位置が通常のモードからずれています。これは、SERUPRO では、主軸の回転数が少なくなるためです。

10.8.4 SERUPRO ASUB

SERUPRO ASUP の特別な注意点

SERUPRO ASUP の場合は、以下の点について特に注意してください。

- レファレンス点復帰:パートプログラム G74 によるレファレンス点復帰動作
- 工具管理機能:工具交換とマガジンデータ
- 主軸起動 SERUPRO ASUP の開始時

G74 レファレンス点復帰

命令 G74 (レファレンス点復帰)がプログラムスタートと検索ターゲットの間にプログラムされた場合、NC はこれを無視します。

SERUPRO アプローチでは、この G74 命令を考慮しません。

工具管理機能

工具管理機能が有効な場合、以下の設定が推奨されます。

MD20310 $MC_TOOL_MANAGEMENT_MASK、ビット 20 = 0

したがって、SERUPRO 操作中に生成された工具管理機能命令は、PLC に出力されません。

工具管理機能命令には、以下の効果があります。

- NC は命令に自動的に応答します。
- マガジンデータは変更されません。
- 工具データは変更されません。
  例外
  テストモード中に有効になった工具は、「有効な」状態とみなされます。このように、SERUPRO 運転後に間違った工具が主軸上に存在することがあります。
  対策
  実際に移動される SERUPRO ASUP を起動します。その起動より前に、正しい工具をロードする ASUP を起動できます。

SERUPRO 運転:機能:手順 2 ～ 6。
SERUPRO ASUP:機能:手順 7。

また、ASUP は T 選択を繰り返さなければならない場合があるため、マシンデータ設定 MD20310 $MC_TOOL_MANAGEMENT_MASK ビット 11 = 1 が必要です。

工具管理機能と補助主軸のあるシステムは、SERUPRO ではサポートされません。

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
N500 DEF INT TNR_AKTUELL	; 有効な T 番号用の変数
N510 DEF INT TNR_VORWAHL	; 事前選択された T 番号用の変数
	; 現在の工具の特定
N520 STOPRE	; プログラムテストで
N530 IF $P_ISTEST	; プログラムの中から
N540 TNR_AKTUELL = $P_TOOLNO	; 「現在の」工具を読み取ります。
N550 ELSE	; そうでない場合、主軸の工具を読み取ります。
N560 TNR_AKTUELL = $TC_MPP6[9998,1]	; 主軸の工具の T 番号の読み取り
N570 ENDIF	
N580 GETSELT(TNR_VORWAHL)	; メイン主軸の事前選択された工具の T 番号の読み取り
	; 工具がまだ最新でない場合のみ、工具交換を実行します
N590 IF TNR_AKTUELL <> TNR_VORWAHL	; 工具交換点へのアプローチ
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	
N610 M206	; 工具交換の実行
N620 ENDIF	

プログラムコード	コメント
N630 M17	

タイプ 5 のブロック検索の後で工具交換プログラムを呼び出すための ASUP

プログラムコード	コメント
PROC ASUPWZV2	
N1000 DEF INT TNR_SPINDEL	; 有効な T 番号用の変数
N1010 DEF INT TNR_VORWAHL	; 事前選択された T 番号用の変数
N1020 DEF INT TNR_SUCHLAUF	; ブロック検索で決定された T 番号用の変数
N1030 TNR_SPINDEL = $TC_MPP6[9998,1]	; 主軸の工具の T 番号の読み取り
N1040 TNR_SUCHLAUF = $P_TOOLNO	; ブロック検索によって決定した T 番号の読み取り。 すなわち、この工具が 現在の工具オフセットを決定します
N1050 GETSELT(TNR_VORWAHL)	; 事前選択された工具の T 番号の読み取り
N1060 IF TNR_SPINDEL ==TNR_SUCHLAUF GOTOF ASUP_ENDE1	
N1070 T = $TC_TP2[TNR_SUCHLAUF]	; 工具名称による T 選択
N1080 L6	; 工具交換プログラムの呼び出し
N1085 ASUP_ENDE1:	
N1090 IF TNR_VORWAHL == TNR_SUCHLAUF GOTOF ASUP_ENDE	
N1100 T = $TC_TP2[TNR_VORWAHL]	; 工具名称による T 事前選択のリストア
N1110 ASUP_ENDE:	
N1110 M90	; PLC へのフィードバック
N1120 REPOSA	; ASUP end

プログラム PROC L6 と PROC ASUPWZV2 の両方で、工具交換が M6 の代わりに M206 を使用してプログラムされます。
ASUP「ASUPWZV2」は、異なるシステム変数を使用して、プログラムの進捗を検出し ($P_TOOLNO)、現在の機械の状態($TC_MPP6[9998,1])を示します。

主軸起動

SERUPRO ASUP が起動されると、工具交換の後で SERUPRO ASUP が新しい工具をワークの正しい位置に移動しようとするので、主軸はプログラムで指定された速度に加速されません。

主軸起動は、SERUPRO ASUP により以下のように行われます。

- SERUPRO 運転が完全に終了します。
- 主軸を起動するには、ファンクションブロック FC 9 を介して SERUPRO ASUP を起動します。
- ASUP で M0 の後でスタートしても主軸状態は変更されません。
- SERUPRO ASUP は、REPOS パートプログラムブロックの前に自動的に停止します。
- START を押します。
- 主軸が ASUP で別にプログラムされなかった場合、主軸はターゲットブロックの状態に加速します。

注記

主軸の REPOS での変更:
速度制御モードと位置決めモードの変化は、SERUPRO アプローチと主軸機能の変更発生時に考慮してください。
主軸の運転モードの切り替えについて詳しくは、「モード (ページ 1556)」の章を参照してください。

10.8.5 自動 SERUPRO

自動 SERUPRO

チャネル別機能「自動 SERUPRO」により、関連した SERUPRO チャネルのプログラムで検索ターゲットを事前に定義する必要の**ない** SERUPRO 実行が可能です。

特別なチャネル「serurpoMasterChan」は、**各**「自動 SERUPRO」機能に対して定義できます。検索ターゲットは、このチャネルで定義できます。

「自動 SERUPRO」機能は、SERUPRO チャネル間ブロックサーチをサポートしています。

機能

「[自動 SERUPRO」操作は、検索ターゲットの検索には使用できません。検索ターゲットに到達しない場合、停止するチャネルはありません。ただし、特定の状況ではチャネルが一時的に停止されます。一般に、チャネルは別のチャネルを待機します。たとえばマーキング、連結または軸入れ替えの待機などです。

待機状態が発生するとき

この待機状態の間に、NC はチャネル「seruproMasterChan」が検索ターゲットに到達したかどうかをチェックします。検索ターゲットに達しない場合、待機状態は終了します。

検索ターゲットに到達した場合、
SERUPRO 操作はこのチャネルでも終了します。「seruproMasterChan」チャネルは、通常の SERUPRO モードで起動してください。

待機状態が発生しないとき

「自動 SERUPRO」は、パートプログラムの M30 により終了します。

チャネルは再度リセット状態になります。

SERUPRO のアプローチは行われません。

チャネルのグループの起動

チャネルのグループが「自動 SERUPRO」によってのみ起動された場合、すべてのチャネルは「RESET」により終了します。

例外:
チャネルは、まったく起動されたことがない協調チャネルを待機します。

チャネル間ブロックサーチは、以下のように実行できます。

- HMI を介して、共に動作しなければならないチャネル(チャネルグループ)を選択します。
- ユーザーは、検索ターゲットが明確に選択されるチャネルグループから特に重要なチャネルを選択します(ターゲットチャネル)。
- HMI が対象チャネルで SERUPRO を起動し、チャネルグループの残りのチャネルで「自動 SERUPRO」を起動します。

関連する**各**チャネルで「seruproActive」を解除した場合、操作は完了します。

「自動 SERUPRO」が別の NCU で受け入れるマスタチャネルはありません。

起動

「自動 SERUPRO」は、対象チャネル「seruproMasterChan」に対するタイプ 5 のブロックサーチスタートとして、HMI を介して起動されます。

対象チャネルから起動される従属チャネルに対して指定される検索ターゲットはありません。

10.8.6 「輪郭での加工継続」のプログラム区間のロック

プログラム指令された中断ポイント

製造や処理関連の理由により、プログラム中止時に特定のプログラム区間内で「輪郭での加工継続」が可能でない場合、このプログラム区間をブロック検索のターゲットブロック用にロックできます。

プログラム中止後に、「輪郭での加工継続」のためにロックされたプログラム区間内に中断点のブロック検索がある場合、コントローラはロックされた区間の開始直前に実行可能であったブロック(メインランブロック)をターゲットブロック(保持ブロック)として使用します。

プログラミング

構文

```
IPTRLOCK()
```

機能

「輪郭での加工継続」をロックするプログラム区間の開始点を特定します。次の実行可能なブロック(メインランブロック)で IPTRLOCK が有効になり、IPTRUNLOCK で解放されるまで、「中断点での加工継続」によるブロック検索のターゲットブロックとして使用されます。以降ではこのブロックを**保持ブロック**と呼びます。

有効性:モーダル

構文

```
IPTRUNLOCK()
```

「輪郭での加工継続」のためにロックされたプログラム区間の終点を特定します。次の実行可能なブロック(メインランブロック)で IPTRLOCK が有効になり、「中断点での加工継続」によって実行中のブロックがブロック検索のターゲットブロックとして再度使用されます。以降ではこのブロックを**解放ブロック**と呼びます。

有効性:モーダル

例

プログラムコード	**コメント**
...	
N010 IPTRLOCK()	; ロック領域:開始
N020 R1=R1+1	
N030 G4 F1	**; 保持ブロック**

プログラムコード	コメント
...	; ロック領域
N200 IPTRUNLOCK()	; ロック領域:終了
N220 R1=R1+1	
N230 G4 F1	**; 解放ブロック**
...	

必要条件

- IPTRLOCK はプログラム内(*.MPF、*.SPF)で最大でプログラムの最後(M30、M17、RET)まで機能します。プログラムの最後では IPTRUNLOCK が自動的に有効になります。
- プログラム内で IPTRLOCK を複数回プログラミングしても、累積効果はありません。プログラム内の IPTRUNLOCK の最初のプログラミングにより、またはプログラムの最後に達したときに、それより前のすべての IPTRLOCK 呼び出しが終了します。
- ロック領域内にサブプログラム呼び出しがある場合、「輪郭での加工継続」はこれと後続のすべてのサブプログラムレベルに対してもロックされます。また、ロックは、呼び出されたサブプログラム内では、明確にプログラム指令された IPTRUNLOCK でキャンセルすることはできません。

例:2 つのプログラムレベルでのロックされたプログラム区間の入れ子指令

PROG_1 での「輪郭での加工継続」ロックの起動により、「輪郭での加工継続」は PROG_2 と後続のすべてのプログラムレベルに対してもロックされます。

プログラムコード	コメント
PROC PROG_1	; プログラム 1
...	
N010 IPTRLOCK()	
N020 R1=R1+1	
N030 G4 F1	**; 保持ブロック**
...	; ロック領域:開始
N040 PROG_2	; ロック領域
...	; ロック領域:終了
N050 IPTRUNLOCK()	
N060 R2=R2+2	
N070 G4 F1	**; 解放ブロック**
...	

プログラムコード	コメント
PROC PROG_2	; プログラム 2
N210 IPTRLOCK()	; プログラム 1 により無効

プログラムコード	コメント
...	
N250 IPTRUNLOCK()	; プログラム 1 により無効
...	
N280 RET	; プログラム 1 により無効

例 3:IPTRLOCK の複数のプログラミング

プログラムコード	コメント
PROC PROG_1	; プログラム 1
...	
N010 IPTRLOCK()	
N020 R1=R1+1	
N030 G4 F1	**; 保持ブロック**
...	; ロック領域:開始
N150 IPTRLOCK()	; ロック領域
...	; ロック領域
N250 IPTRLOCK()	; ロック領域
...	; ロック領域:End
N360 IPTRUNLOCK()	
N370 R2=R2+2	
N380 G4 F1	**; 解放ブロック**
...	

システム変数

実行中のブロックの状態はシステム変数$P_IPTRLOCK を介して特定できます。

$P_IPTRLOCK	意味
FALSE	実行中のブロックは「輪郭での加工継続」用にロックされたプログラム区間内に**ありません**
TRUE	実行中のブロックは「輪郭での加工継続」用にロックされたプログラム区間内にあります

機能別の「輪郭での加工継続」自動ロック

各種の連結では、「輪郭での加工継続」ロックの起動/解除を、連結の起動/解除によってチャネル別に自動的に行うことができます。

MD22680 $MC_AUTO_IPTR_LOCK、ビット x

ビット	値	意味
0	電子ギヤ (EGON / EGOF)	
	1	「輪郭での加工継続」の自動ロックが有効です
	0	「輪郭での加工継続」の自動ロックが無効です
1	軸の軸間連動機能 (LEADON / LEADOF)	
	1	「輪郭での加工継続」の自動ロックが有効です
	0	「輪郭での加工継続」の自動ロックが無効です

このプログラム区間は、起動**前の**最後の実行可能なブロックにより開始し、解除により終了します。

自動中断ポインタは、シンクロナイズドアクションを介して起動または解除された連結に対しては有効ではありません。

例:軸の軸間連動機能を検索抑制として自動的に宣言

プログラムコード	コメント
N100 G0 X100	
N200 EGON(Y,"NOC",X,1,1)	; 検索抑制プログラムセクションの開始
N300 LEADON(A,B,1)	
...	
N400 EGOFS(Y)	
...	
N500 LEADOF(A, B)	; 検索抑制プログラムセクションの終了
N600 G0 X200	

検索抑制プログラム区間(N200 - N500)内のプログラム中止は、常に中断ポインタに N100 を提供します。

通知

機能のオーバーラップを原因とする望ましくない状態

「プログラム可能な中断ポインタ」とマシンデータによる「自動中断ポインタ」がオーバーラップしている場合、NC は可能な最大の検索抑制領域を選択します。

プログラムでは、実行時間のほとんどすべてに対する連結が必要な場合もあります。この場合、自動中断ポインタは常にプログラムの始点を指し、SERUPRO 機能が実際には役に立たなくなります。

10.8.7 POWER ON、モード変更および RESET 中の動作

SERUPRO は POWER ON の間は無効です。SERUPRO 中の操作モード変更は可能です。RESET は SERUPRO を中止します。内部で選択されたプログラムテストは再度解除されます。SERUPRO は他のブロックサーチタイプと組み合わせることはできません。

10.8.8 必要条件

10.8.8.1 ターゲットブロックの STOPRE

STOPRE ブロックは先行するブロックからすべてのモーダル設定を受け取るので、次の動作に関して前もって条件を適用することができます。

- 現在処理しているプログラム行をメインランと同期します。
- たとえば、SERUPRO のアプローチでこの REPOS 動作に影響を与えるために、SERUPRO のモーダル設定を入手します。

例:X 軸の指令値を指定することにより、Z 軸を位置決めします。

ブロック「G1 F100 Z=$AA_IM[X]」が中断されると、先行する STOPRE ブロックによりメインランとの同期が保証されます。したがって、Z 軸を同じ位置に移動するために、$AA_IM を介して X 軸の正しい指令値が読み取れます。

例:外部ゼロオフセットを読み取り、正しく計算します。

```
N10 G1 X1000 F100
N20 G1 X1000 F500
N30 G1 X1000 F1000
N40 G1 X1000 F5000
N50 SUPA G1 F100 X200              ; 外部ゼロオフセットを 200 へ移動
N60 G0 X1000
N70 ...
```

N50 の前の自動的な STOPRE により、NC は、現在のゼロオフセットを読み取って正確に計算することができます。
N50 検索ターゲットでの SERUPRO 運転では、再位置決めは、SERUPRO アプローチの自動的な STOPRE で行われ、速度は、N40 から F5000 で特定されます。

自動的な先読み停止

自動的に先読み停止が発行される状況:

1. 以下の変数のどれかのアクセス操作が発生するすべてのブロック:
   - $A...で始まるシステム変数のプログラミング
   - 属性 SYNR/SYNRW で始まるシステム変数のプログラミング
2. 以下の命令の場合:
   - パートプログラムの MEACALC、MEASURE 命令
   - SUPA (フレームとオンライン補正のマスク)のプログラミング
   - CTABDEF (カーブテーブル定義の開始)のプログラミング
   - パートプログラム WRITE/DELETE 命令(ファイルの書き込み/削除)
   - WRITE/DELETE 命令のシーケンスの最初の WRITE/DELETE 命令
   - パートプログラムの EXTCALL 命令
   - パートプログラムの GETSELT、GETEXET 命令
   - 工具および有効な FTOCON 工具の精密な補正の場合
3. 以下の命令処理の場合:
   - タイプ 1 ブロック検索(「計算なしのブロック検索」)の終了処理
   - 計算ありのタイプ 2 ブロック検索(「輪郭の終点でのブロック検索」)の終了処理

10.8.8.2 ターゲットブロックでの SPOS

主軸が M3/M4 でプログラム指令されており、ターゲットブロックで SPOS への切り替えが行われている場合は、SERUPRO 動作(「検索ターゲット検出」状態)の終了時に主軸が SPOS に切り替わります。

DB31、... DBX84.5 = TRUE (有効な主軸モード:位置決めモード)

10.8.8.3 突き当て点停止(FXS)

再位置決め(REPOS)の間、「突き当て点停止」機能(FXS)が自動的に繰り返されます。すべての軸が考慮されます。検索ターゲットの直前にプログラム指令されたトルクがトルクとして使用されます。

システム変数

「突き当て点停止」のシステム変数は、SERUPRO では次の意味を持ちます。

- $AA_FXS:プログラムシミュレーションの進行状況
- $VA_FXS:実際の機械の状態

2 つのシステム変数は、SERUPRO 機能**以外**では、常に同じ値です。

ASUB

ユーザー専用の ASUB を SERUPRO 用に起動できます。

参照先

SERUPRO ブロック検索についての詳細は、「詳細説明 (ページ 376)」の章を参照してください。

10.8.8.4 トルク/推力を制限した移動(FOC)

再位置決め(REPOS)の間、「トルク/推力を制限した移動」機能(FOC)が自動的に繰り返されます。 すべての軸が考慮されます。 検索ターゲットの直前にプログラム指令されたトルクがトルクとして使用されます。

システム変数

「トルク/推力を制限した移動」のシステム変数は、SERUPRO では次の意味を持ちます。

- $AA_FOC: プログラムシミュレーションの進行状況
- $VA_FOC: 実際の機械の状態

必要条件

変化するトルク特性は再位置決めの間は実行**できません**。

例

プログラムは軸 X を 0 から 100 に移動し、20 の移動毎のなかで 10 の移動の間「トルク/推力を制限した移動」(FOC)に切り替えます。 このトルク特性は通常はノンモーダルな FOC で生成されるため、再位置決め(REPOS)中はおこなわれません。 その代わり、前回のプログラミングに従って、制限されたトルク/推力を使用してまたは使用しないで軸 X を 0 から 100 に移動します。

参照先

SERUPRO ブロック検索についての詳細は、「詳細説明 (ページ 376)」の章を参照してください。

10.8.8.5 主軸同期

主軸同期はシミュレーション可能

マスタ主軸と任意の数のスレーブ主軸を使用する主軸同期運転を SERUPRO によりすべての既存のチャネルでシミュレーションできます。

主軸同期に関する詳細情報については、以下を参照してください。

参照先:

『機能マニュアル、上級機能』;「主軸同期(S3)」

10.8.8.6 連結とマスタスレーブ

指令値とフィードバック値の連結

SERUPRO 運転は、指令値とフィードバック値の連結をシミュレーションできるプログラムテストモードでのプログラムシミュレーションです。

EG シミュレーションの指定

EG のシミュレーションには、以下の定義が適用されます。

1. シミュレーションは、常に指令値を連結して実行されます。
2. SERUPRO に一部のマスタ軸しかない場合、シミュレーションはアラーム 16952「クリアリセット/起動なし」により中止されます。これは、チャネル間連結の場合に発生する可能性があります。
3. NC の観点から 1 つのエンコーダしかなく外部で移動される軸は、正しくシミュレーションできません。これらの軸を連結で統合しないでください。

⚠ 注意

誤ったシミュレーション

連結を正しくシミュレーションするには、その連結を事前にオフにしてください。

これは、マシンデータ MD10708 $MA_SERUPRO_MASK でおこなうことができます。

連結軸の指定

SERUPRO 運転では、連結軸が指令値の連結であると仮定してシミュレーションします。このように、SERUPRO アプローチの目標位置として使用される**すべての**軸に対して、終点が計算されます。この連結は「検索ターゲット発見」の場合でも有効です。現在の点から終点までの軌跡が、連結が有効な状態で SERUPRO アプローチに対して実行されます。

LEADON

軸間連動機能のシミュレーションには、以下の指定が適用されます。

1. シミュレーションは、常に指令値を連結して実行されます。
2. シミュレーションされた目標位置に到達するために、有効な連結とスレーブ軸の重畳動作を使用して、SERUPRO アプローチが実行されます。

連結により単独で移動されるスレーブ軸は、必ずしも目標位置に到達できません。
SERUPRO アプローチでは、シミュレーションされた位置にアプローチするために、重畳した直線動作がスレーブ軸に対して計算されます。

JOG による LEAD に対するシミュレーションされた目標位置への到達

「検索ターゲット発見」時点で連結はすでに有効であり、特に JOG 動作に対して有効です。目標位置に到達しない場合、SERUPRO アプローチは、有効な連結と重畳動作によるスレーブ軸の目標位置への移動に使用できます。

注記

軸連結の再位置決めについて詳しくは、「輪郭への再位置決め(REPOS) (ページ 619)」の章を参照してください。

マスタスレーブ

システム ASUB は、ブロック検索が終了した後に自動的に起動できます。この ASUB では、連結状態と関連する軸の位置を制御できます。必要な情報は以下のシステム変数で得られます。

システム変数	説明
$P_SEARCH_MASLD[<スレーブ軸>]	リンクが閉じているときのスレーブとマスタ軸の間の位置オフセット。
$AA_MASL_STAT[<スレーブ軸>]	マスタ/スレーブ連結の現在の状態

システム変数	説明
$P_SEARCH_MASLC[<スレーブ軸>]	状態:連結の状態がブロック検索中に変更されました
システム変数は、MASLON で連結をオンにしたときに削除されます。	

注記

ブロック検索を実行する場合、連結軸は同じチャネル内にあるようにしてください。

参照先

マスタ-スレーブ連結について詳しくは、以下の資料を参照してください。

『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

例

- システム ASUB
  - 軌跡と名前: /_N_CMA_DIR/PROGEVENT.SPF
  - マスタ軸:X
  - スレーブ軸:Y

プログラムコード	
PROG PROGEVENT	
N10 IF(($S_SEARCH_MASLC[Y]< >0) AND ($AA_MASL_STAT[Y]< >0))	
N20 MASLOF(Y)	
N30 SUPA Y = $AA_IM[X] - $P_SEARCH_MASLD[Y]	
N40 MASLON(Y)	
N50 ENDIF	
N60 REPOSA	
...	
RET	

- マシンデータ

  ASUB が確実に自動起動するように、以下のマシンデータを設定してください。
  - NC 別マシンデータ:
    - MD11604 $MN_ASUP_START_PRIO_LEVEL = 100
    - MD11450 $MN_SEARCH_RUN_MODE = 'H02'
  - ASUB を起動するチャネルまたは一般的なすべてのチャネルに対するチャネル別:
    - MD20105 $MC_PROG_EVENT_IGN_REFP_LOCK, bit<n> = TRUE
      N:すべての必要なイベント起動プログラム呼び出し(プログラムイベント)用
    - MD20115 $MC_IGNORE_REFP_LOCK_ASUP、ビット<n> = TRUE
      N:すべての**必要なユーザー**の割り込み用

通知
システムの割り込み MD20115 $MC_IGNORE_REFP_LOCK_ASUP、ビット 8 ～ 31 でシステムの割り込みが可能になります。 ビット /割り込み 9 で、移動動作を含む ASUB が開始されます。

軸連結

- 処理速度との加速、およびチャネルの異なるマスタ軸とスレーブ軸
  マスタ軸のスレーブ軸が、マスタ軸とは異なるチャネルに割り当てられている場合は、実行速度の設定と加速が無効になります(MD22601 (ページ 630)$MC_SERUPRO_SPEED_FACTOR)
- 連結動作
  連結動作機能(TRAILON)は SERUPRO でサポートされています。
  TRAILON と TRAILOF による連結動作にについて詳しくは、以下の資料を参照してください。
  参照先:
  – 機能マニュアル 応用機能; 軸の連結(M3)
  – 『プログラミングマニュアル、上級編』;「軸連結」の章
- ガントリ軸
  ガントリ軸機能は SERUPRO でサポートしています。
  ガントリ軸の機能について詳しくは、以下の資料を参照してください。
  参照先:
  『機能マニュアル、応用機能』、「G1:ガントリ軸」の章
- 法線方向制御
  個々の軸の法線方向制御機能は SERUPRO でサポートしています。
  法線方向制御の追加情報については、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』、「T3:法線方向制御」の章

10.8.8.7 軸機能

SERUPRO 条件

SERUPRO の特別な条件を、軸イネーブル、独立軸運転および軸入れ替えについて考慮してください。

軸イネーブル

閉ループ制御イネーブルが、機械で発行されない、または発行できない場合で、かつプログラムテストまたは SERUPRO が有効なときだけ動作しない場合、軸のインタフェース DB31、... DBX3.7 (「プログラムテスト軸/主軸イネーブル」)が、軸イネーブルを制御します。

インタフェース信号 PLC → NC
DB31、... DBX3.7 (プログラムテスト軸/主軸イネーブル)を介して、このイネーブルを発行することができます。プログラムテスト中または SERUPRO の間に実際のコントローライネーブルが欠落している場合、軸/主軸への影響は以下のとおりです。

- プログラムシミュレーションの実行により軸/主軸を移動しようとすると直ちに、メッセージ「軸イネーブルの待機中」または「主軸イネーブルの待機中」が表示され、シミュレーションが停止されます。
- シミュレーション動作中に、NC/PLC インタフェース信号 DB31、... DBX3.7 (プログラムテスト軸/主軸イネーブル)がキャンセルされた場合、アラーム 21612:「チャネル%1 軸%2 NC/PLC インタフェース信号「コントローライネーブル」の動作中のリセット」が発生します。

独立軸運転

独立軸運転は、SERUPRO でシミュレーション可能な PLC により制御される軸です。SERUPRO 運転の間に、通常の運転では、PLC は軸の制御を占有または放棄できます。必要に応じて、この軸は FC18 を使用して移動することもできます。PLC は、アプローチブロックの**前は**軸の制御を占有し、この軸の位置決めを担当します。これは、すべてのブロック検索タイプに対して有効です。

独立軸運転に関する詳細情報については、以下を参照してください。

参照先
『機能マニュアル、上級機能』；「位置決め軸(P2)」

軸入れ替え

問題:プログラムは軸を移動し、WAITP(X)によりターゲットブロックの前に制御を放棄します。したがって、X は REPOS の対象とならず、軸は SERUPRO アプローチで考慮されません。

マシンデータ MD11470 $MN_REPOS_MODE_MASK を介して、以下の動作を SERUPRO-REPOS に対して実現できます。

中立軸は、SERUPRO-REPOS では「コマンド軸」として移動されます。この軸は、最後に軌跡軸としてプログラムされた場合でも、軌跡に沿わずに補間されます。この状態では、速度は MD32060 $MA_POS_AX_VELO から得られます。SERUPRO アプローチの後、この軸は再度中立になります。

ただし再位置決めすることができない中立軸は、軸の NC/PLC インタフェース信号「REPOSDELAY」を設定してください。これにより、REPOS 動作が削除されます。

例:

SERUPRO の後、1 つの軸が意図的にテクノロジサイクルを介した シンクロナイズドアクションで移動されます。コマンド軸は、常にアプローチブロックで移動され、ターゲットブロックでは移動されません。ターゲットブロックは、すべてのコマンド軸が最後まで移動された場合のみ変更できます。

⚠ 注意

PLC 制御軸は再位置決めされません

ターゲットブロックの前に RELEASE(X)により有効になった軸は、再位置決めされません。

10.8.8.8 ギヤ選択切替

操作手順

ギヤ選択切替(GSC)では、新しいギヤを使用するために、NC からの物理的な移動が必要です。
SERUPRO 運転では、ギヤ選択切替の必要はなく、以下のように実行されます。

軸が揺動しなければならないか、事前に特定の位置にアプローチしなければならないので、NC により制御される場合のみ、一部のギヤを変更できます。

ギヤ選択切替は、MD35035 $MA_SPIND_FUNCTION_MASK でビット 0 ～ 2 を使用して、ドライラン、プログラムテストおよび SERUPRO に対して選択的にマスクできます。

ギヤ選択切替は、REPOS でおこなってください。これは、関係のある軸がターゲットブロックで「速度制御モード」になる場合でも機能します。その他の場合は、自動のギヤ選択切替はアラームにより拒否されます。たとえば、ギヤ選択切替とターゲットブロックの間で軸が座標変換または連結に関与していた場合です。

注記

ドライラン、プログラムテストおよび SERUPRO でのギヤ選択切替に関する詳細情報については、「S1:主軸 (ページ 1555)」の章を参照してください。

10.8.8.9 重畳動作

SERUPRO のみ

「重畳動作」を使用する場合、重畳動作はメインランで適宜補間されるので、プログラムテスト(SERUPRO)を介したブロックサーチだけが使用できます。 これは特に$AA_OFFに適用されます。

最大軸速度に代わる速度波形

プログラムテスト中は、メインランの中で「重畳動作」を補間することができる速度波形を使用してください。 したがって、最大軸速度で補間することはできません。

「ドライラン送り速度」モードの軸速度は
SD42100 $SC_DRY_RUN_FEED を使用して設定されます。

SERUPRO 操作の速度は、
MD22600 $MC_SERUPRO_SPEED_MODE を使用して選択されます。

10.8.8.10 NC/PLC インタフェース信号

REPOS オフセット有効

SERUPRO 中に軸の REPOS オフセットが生じた場合、SERUPRO 運転の最後に軸の NC/PLC インタフェースを介してこれが表示されます。

DB31、...DBX70.0 == 1 (REPOS オフセット有効)

REPOS オフセットの有効範囲

REPOS オフセットは、SERUPRO ASUB の開始時または加工を再開する NC スタートで無効になります。

DB31、... DBX70.1 == 1 (REPOS オフセット無効)

軸は、SERUPRO 運転の最後と加工を再開する NC スタートの間で、JOG モードで手動で移動するか、FC 18 を使用して PLC ユーザープログラムを介して移動できます REPOS オフセットが全て移動されると、インタフェース信号はリセットされます。

10.8.8.11 よりフレキシブルな初期設定の作成

初期設定/基本 SERUPRO 設定

マシンデータ MD20112 $MC_START_MODE_MASK は、G コード(特に平面選択と設定可能ゼロオフセット)、工具長補正、座標変換および軸連結に関する、パートプログラム起動用の制御の初期設定を定義します。SERUPRO 操作には MD22620 $MC_ENABLE_START_MODE_MASK_PRT を使った、通常のパートプログラムの起動とは異なる初期設定を選択するための特別なオプションが存在します。したがって新規設定はマシンデータに保存する必要があります。

MD22620 $MC_START_MODE_MASK_PRT

MD22620 のビットの意味は MD20112 $MC_START_MODE_MASK と同じです。

例:

SERUPRO 操作の開始時の主軸同期連結は、パートプログラムの起動では保持されます。

```
                                                    ; 主軸同期連結は設定されていません
$MC_START_MODE_MASK = 'H400'                        ; はオフになります
$MC_START_MODE_MASK_PRT = 'H00'                     ; 有効なままです
$MC_ENABLE_START_MODE_MASK_PRT = 'H01'              ; $MC_START_MODE_MASK_PRT は、
                                                    $MC_START_MODE_MASK の代わりに
                                                    SERUPRO で評価されます。
```

10.8.8.12 コンプレッサ機能(COMPON、COMPCURV、COMPCAD)

- **タイプ 2** または**タイプ 4**(...の**計算による**ブロック検索)のブロック検索のターゲットブロックが、コンプレッサ機能(COMPON、COMPCURV、COMPCAD)が有効なプログラムセクションにある場合は、再位置決め時に**コンプレッサ**によって計算された軌跡上で位置にアプローチします。これらの位置は、パートプログラムで指令された軌跡上の位置と正確に一致するようにしてください。
- プログラムされたブロックが圧縮時にパートプログラムから削除された場合は、これらのブロックがブロック検索のターゲットブロックに見つかりません⇒アラーム 15370「検索ターゲットが見つからない」。

10.8.9 システム変数

SERUPRO に関連するシステム変数の一覧

システム変数	意味
$AC_ASUP、ビット 20	ASUP の起動理由: $AC_ASUP、ビット 20 == 1 ⇒ システム ASUP が有効、理由:SERUPRO 検索目標に到達
$AC_SERUPRO	SERUPRO の状態 $AC_SERUPRO == 1 ⇒ SERUPRO が有効
$P_ISTEST	プログラムテストの状態 SERUPRO が有効 ⇒ $P_ISTEST == 1
$P_SEARCHL	直前に有効であったブロック検索タイプ: $P_SEARCHL == 5 SERUPRO の開始からプログラムのリセットまたはエンドオブプログラムまで
$AC_REPOS_PATH_MODE	SERUPRO の後に輪郭に再位置決めするための REPOS モード

10.9 プログラム運転

定義

プログラムモードは、自動または MDI 運転モードで NC プログラムまたは PC プログラムブロックが処理される場合に存在します。

NC/PLC インタフェース信号

プログラムモードは、モードグループ別およびチャネル別の NC/PLC インターフェース信号により PLC ユーザープログラムの影響を受ける場合があるか、または適切なフィードバックを PLC ユーザープログラムに返します。

NC/PLC インターフェース信号の機能別の概要については、以下の資料を参照してください。

参照先

- 『機能マニュアル、基本機能』、「Z1:NC/PLC インタフェース信号
- 『機能マニュアル、上級機能』;「Z2:NC/PLC インタフェース信号
- 『機能マニュアル、応用機能』;「Z3:NC/PLC インタフェース信号

すべての NC/PLC インターフェース信号の概要については、以下の資料を参照してください。

参照先

『リストマニュアル、NC 変数とインターフェース信号』、「インターフェース信号-概要」の章

10.9.1 初期設定

初期設定は、チャネルごとにチャネルマシンデータでプログラム指令できます。これらの初期設定は、G グループや補助機能の出力などに影響を及ぼします。

補助機能出力

補助機能の出力のタイミングは、マシンデータ AUXFU_x_SYNC_TYPE (MD22200、22210、22220、22230、22240、22250、22260)、(M、S、T、H、F、D、E 機能用の出力タイミング)を使用して予め定義できます。詳しくは、「H2: PLC への補助機能出力 (ページ 475)」の章を参照してください。

G グループ

初期プログラミング設定は、
MD20150 $MC_GCODE_RESET_VALUES (G グループのリセット状態)を使用して、使用可能な G グループのそれぞれについて指定できます。この初期設定は、プログラム起動またはリセットで自動的に有効になり、同じ G グループからの G 命令によって解除されるまで有効です。

G 命令の PLC インタフェースへの出力は、MD22510 $MC_GCODE_GROUPS_TO_PLC (ブロック切り替え/ RESET 後に NC-PLC インタフェースに出力される G 命令)で有効にできます。

参照先

G グループと、関連する G 命令のリストは、以下の資料にあります。

プログラミングマニュアル基本編

SINUMERIK ソリューションラインの NC 言語範囲の基本構成

SINUMERIK 840D sl の場合、マシンデータを介して NC 言語範囲の一定の基本構成を作成できます(構成できます)。NC 言語範囲のオプションと機能は、ユーザーのニーズに合わせて特別に調整(構成)されます。

10.9.1.1 マシンデータ

NC 言語範囲

有効でないオプションと機能を言語命令で使用する方法は、以下のマシンデータで設定できます。

MD10711 $MN_NC_LANGUAGE_CONFIGURATION = <値>

値	意味
0	すべての言語命令を使用できます。必要な機能が起動されるかどうかは、実行時にしか認識できません。
1	すべての言語命令を使用できます。 有効でないオプション用の言語命令は、プログラム解析時に認識されます⇒アラーム 12553「オプション/機能が無効です」。
2	有効なオプションの言語命令のみを使用できます。 有効でないオプション用の言語命令は、プログラム解析時に認識されます⇒アラーム 12550「名前が未定義またはオプション/機能が使用可能」。
3	すべての言語命令を使用できます。 起動されていない機能用の言語命令は、プログラム解析時に認識されます⇒アラーム 12553「オプション/機能が無効です」。 **例**: 1. 円筒補間用に設定されたオプション。 2. マシンデータ MD24100 $MC_TRAOF_TYPE_1 の円筒補間が**起動されていません**。 3. アラーム 12553 が、TRACYL 命令のプログラミング時にすでに発行されています。
4	使用できるのは、起動されている機能の言語命令のみです。 起動されていない機能の命令は認識されません⇒アラーム 12550「[名称が定義されていないか、オプション/機能が使用できません」。 **注** 関連づけられている言語命令がシーメンス言語で一般に使用できないのか、対応するシステムでのみ使用できないのかは、この場合は区別できません。

10.9.1.2 プログラミング

機能「STRINGIS(...)」では、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が確認されます。

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 グループの G 命令
- DIN または NC アドレス
- 機能
- プロシージャ
- キーワード
- 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

定義

```
INT STRINGIS(STRING <名称>)
```

構文

```
<戻り値> = STRINGIS(<名前>)
```

意味

STRINGIS():	戻り値を伴うテスト機能
<名称>:	テストする文字列

<戻り値>:	戻り値は、先頭の 3 桁 yxx にコード化されます。	
	000	指定された文字列は、現在の言語範囲の要素ではありません [1]。
	100	指定された文字列は、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	指定された文字列は、現在の言語範囲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)。 詳細情報は、次のように、先頭の 2 桁に含まれています。
		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	指定された文字列は NC アドレスです。ただし、その名称は、DIN アドレス、NC アドレス(xx==01)、マクロ名(xx==10)のいずれでもなく、また、G でも R でもありません [2]
	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前は、STRINGIS ではチェック**できません**。

4) マクロとして定義されたアドレス、たとえば G、H、M、L はマクロとして識別されます。

例

以下の例では、指定された NC 言語要素は、特に注記のない限り、原則として制御装置でプログラム指令できると想定しています。

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 NC 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")`
   以下の段落も参照してください。「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")`

必要条件

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。

`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。

- MD18800 $MN_MM_EXTERN_LANGUAGE (外部 NC 言語の適用)
- MD10880 $MN_ MM_EXTERN_CNC_SYSTEM (設定するコントロールシステム)

有効な「ISO モード」の場合は、`STRINGIS()` が、SINUMERIK G 命令に属するかどうかにかかわらず、指定された文字列を最初に確認します。

この文字列が SINUMERIK G 命令でない場合は、次に、ISO G 命令かどうかを確認します。

ISO モードのプログラム指令切り替え

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。

10.9.2 NC プログラムの選択と起動

NC/PLC インタフェース信号

選択

NC プログラムは、関連するチャネルがリセット状態の場合にのみ選択できます。

- DB21、... DBX35.7 == 1 (リセット)

開始

NC プログラムは、以下の 2 つの異なるイベントによって起動されます。

1. DB21、... DBX7.1 = 1 (NC スタート)、信号は、通常、機械操作パネルで「NC スタート」キーを押して開始します。
2. 別の有効なチャネルの NC プログラムでの START 命令。チャネルは、自動または MDI モードで、「リセット」または「中断」状態でなければなりません。
   - DB21、... DBX35.7 == 1 (リセット)
   - DB21、... DBX35.6 == 1 (中断)

初期条件

NC プログラムは、以下の初期条件を満たす場合にのみ起動できます。

- DB11 DBX4.4 == 1 (モードグループ準備完了)
- DB11 DBX0.7 == 0 (モードグループリセット)
- DB21、... DBX1.7 == 0 (プログラムテスト起動)
- DB21、... DBX7.0 == 0 (NC スタートロック)
- DB21、... DBX7.2 == 0 (ブロック境界での NC ストップ)
- DB21、... DBX7.3 == 0 (NC ストップ)
- DB21、... DBX7.4 == 0 (NC ストップ、軸および主軸)
- DB21、... DBX7.7 == 0 (リセット)
- DB10.DBX56.1 == 0 (非常停止)
- 未処置の軸も未処置の NC 別アラームも存在しない

命令の実行

パートプログラムまたはパートプログラムブロックが、自動的に実行され、次のインターフェース信号が設定されます。

- DB21、... DBX35.5 (チャネル状態有効)
- DB21、... DBX35.0 (プログラム状態実行中)

プログラムの最後まで到達するか、STOP 命令または RESET 命令によってチャネルが中断または中止されるまで、プログラムが処理されます。

参照先

このインタフェース信号の詳細については、『NC 変数およびインタフェース信号用パラメータマニュアル』を参照してください。

アラーム

特定の条件では、START 命令は効果がなく、次のアラームのうちの 1 つが発生します。

- 10200 [発生中のアラームにより、NC スタートは許可されません]
- 10202 [起動中の命令により、NC スタートは許可されません]
- 10203 [原点確立されていない軸については、NC スタートが許可されません]

参照先:

『診断マニュアル』、「アラーム」

10.9.3 プログラムの中断

NC/PLC インタフェース信号

必要条件

パートプログラムの中断は、チャネルと NC プログラムが有効な場合にのみ実行されます。

- DB21、... D35.5 == 1 (チャネル:有効)
- DB21、... D35.0 == 1 (プログラム:実行中)

プログラムの中断

プログラム処理は、以下のイベントで中断できます。

- DB21、... DBX7.2 == 1 (ブロック境界での NC ストップ)
- DB21、... DBX7.3 == 1 (NC ストップ)
- DB21、... DBX7.4 == 1 (NC ストップ、軸および主軸)
- DB21、... DBX2.0 == 1 (シングルブロック)
- 処理済み NC プログラムでプログラム指令された M00 または M01 命令

チャネルと NC プログラムが「中断」状態になります。

- DB21、... D35.6 == 1 (チャネルが中断)
- DB21、... D35.3 == 1 (プログラムが中断)

参照先

このインタフェース信号の詳細については、『NC 変数およびインタフェース信号用パラメータマニュアル』を参照してください。

処理

以下の動作は、プログラムの中断が検出されると実行されます。

- プログラム処理の中断
  - 以下のイベントの次のブロック境界:「ブロック境界での NC ストップ」、M00、M01、またはシングルブロック
  - 即時:その他のすべてのイベント
- チャネルの移動している軸が減速カーブで停止します。軸の減速は、複数のブロックにわたる場合があります。
- ブロック表示は、中断点の現在のブロックを表示します。
- 中断点までに出力されていなかった補助機能は、出力されなくなります。

中断状態で可能な動作

パートプログラムの中断中に、次のようなさまざまな機能をおこなうことができます。

- **オーバストア**
  参照先:
  『SINUMERIK Operate 用操作マニュアル』、「機械ワーク」の「オーバストア」の章
- **ブロック**検索
  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)」 の「ブロック検索」または「ブロック検索タイプ 5 SERUPRO」の章
- **輪郭への再位置決め(REPOS)**
  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、プログラムモード、リセット応答(K1)」の「ブロック検索タイプ 5 SERUPRO」の「REPOS」の「制御された REPOS による再位置決め」の章
- **指定方向への工具後退**
  参照先:
  – 『プログラミングマニュアル、上級編』; 「工具オフセット」の章
  – 『機能マニュアル、基本機能』; 「工具オフセット(W1)」 の「旋回工具ホルダ」の「
- **ASUB** (「非同期サブプログラム(ASUB) (ページ 700)」の章を参照)。
- **DRF 機能、ワーク原点のオフセット**
  参照先:
  『機能マニュアル、上級機能』、「JOG 運転と手動ハンドル運転(H1)」
- **中断された NC プログラムの再開**
  – 別のチャネルからの START 命令
    参照先:
    『プログラミングマニュアル、上級編』、「フレキシブルな NC プログラミング」 の「プログラム協調(INIT、START、WAITM、WAITMC、WAITE、SETM、CLEARM)」の章
  – NC/PLC インターフェース:
    DB21、... DBX7.1 = 1 (NC スタート)

10.9.4 チャネルリセット

機能

チャネルをリセットすると、NC プログラムが自動モードで処理されるか、プログラムブロックが MDI モードで処理されて終了します。

NC プログラムまたはプログラムブロックは、中断点からは再開できません。チャネルリセットの完了後は、チャネルのすべての軸と主軸が「イグザクトストップ」状態(例外:フォローアップモード)になります。

手順

NC リセットにより、以下の動作がチャネルで実行されます。

- プログラム解析が直ちに停止します。
- 軸と主軸が、その設定されている減速カーブによって停止状態まで減速します。
- この時点でまだ出力されていない、現在のブロックの補助機能はそれ以上、PLC に出力されません。
- ブロック表示が、選択した NC プログラムの先頭にリセットされます。
- チャネルのすべてのリセットアラームが画面から消去されます。

規則

- チャネルリセットが、あらゆるチャネル状態で実行されます。
- チャネルリセットは、他の命令では中止されません。

NC/PLC インタフェース信号

要求 :チャネルリセット

チャネルリセットは、以下の NC/PLC インターフェース信号で要求します。

- DB21、... DBX7.7 = 1 (リセット)

要求 :モードグループリセット

モードグループリセットにより、モードグループのすべてのチャネルでチャネルリセットが開始されます。

モードグループリセットは、以下の NC/PLC インターフェース信号で要求します。

- DB11、... DBX0.7 = 1 (モードグループリセット)

フィードバック:チャネルリセット完了

- DB21、… DBX35.7 == 1 (チャネル状態リセット)

フィードバック:モードグループリセット完了

- DB11、... DBX6.5 == 1 (モードグループリセット)

参照先

このインタフェース信号の詳細については、『NC 変数およびインタフェース信号用パラメータマニュアル』を参照してください。

10.9.5 プログラム状態

選択された NC プログラムの状態は、チャネルごとにインターフェースに出力されます。

すべてのプログラム状態は、自動および MDI モードで発生する可能性があります。他のすべてのモードまたは機械機能では、プログラム状態が中止または中断です。

NC/PLC インタフェース信号

以下のプログラムの状態が NC/PLC インタフェース(DB21、...)に表示されます。

- DB21、... DBX35.4 (「中止」)
- DB21、... .DBX35.3(「中断」)
- DB21、... .DBX35.2(「停止」)
- DB21、... DBX35.1 (「待機中」)
- DB21、... DBX35.0 (「実行中」)

このインタフェース信号の詳細については、『NC 変数およびインタフェース信号用パラメータマニュアル』を参照してください。

状態変更

プログラム状態は、命令と NC/PLC インタフェースおよびアラームの影響を受けます。以下の表は、「実行中」プログラム状態で開始した場合の、関連するイベントに基づく、その次の状態を示します。

プログラムの初期状態:「実行中」					
イベント	次のプログラム状態				
	中止	中断	停止	待機	運転中
DB21、... DBX7.7 (リセット)	x				
DB21、... DBX7.3 (NC ストップ)			x		
DB21、... DBX7.2 (ブロック境界での NC ストップ)			x		
DB21、... DBX7.4 (NC ストップ、軸および主軸)			x		
DB21、... DBX6.1 (読み込み禁止)					x
DB21、... DBX6.0 (送り速度停止)					x
DB21、... DBX12.3 / 16.3 / 20.3 (送り速度停止、ジオメトリ軸 1 / 2 / 3)					x
DB21、... DBB4; 送り速度オーバライド= 0%					x
DB31、... DBX4.3 (送り/主軸停止)					x
DB21、... DBX194.2/DBX197.6 (ブロック内の M02/M30)	x				
DB21、... DBX194.0/DBX194.1 (ブロック内の M00/M01)			x		
DB21、... DBX0.4 (シングルブロック)			x		
DB21、... DBX6.2 (残移動距離削除)					x
PLC に出力されたが、まだ応答されていない補助機能			x		
プログラム内の WAIT 命令				x	
「NOREADY」システム動作を伴うアラーム		x			

10.9.6 チャネル状態

すべての運転モードで、各チャネルのチャネル状態が NC/PLC インターフェース (DB21、...)に表示されます。

NC/PLC インタフェース信号

チャネル状態が、NC/PLC インターフェースで以下の信号によって示されます。

- DB21、... DBX35.7 (「リセット」)
- DB21、... .DBX35.6(「中断」)
- DB21、... .DBX35.5(「有効」)

このインタフェース信号の詳細については、『NC 変数およびインタフェース信号用パラメータマニュアル』を参照してください。

状態変更

チャネル状態は、命令と NC/PLC インターフェース信号の影響をうけます。以下の表は、「有効」チャネル状態で開始した場合の、関連するイベントに基づく、その次の状態を示します。

チャネルの初期状態:「有効」			
イベント	次のチャネル状態		
	「リセット」	「中断」	「有効」
DB21、... DBX7.7 (リセット)	X		
DB21、... DBX7.3 (NC ストップ)		x	
DB21、... DBX7.2 (ブロック境界での NC ストップ)		x	
DB21、... DBX7.4 (NC ストップ、軸および主軸)		x	
DB21、... DBX6.1 (読み込み禁止)			x
DB21、... DBX6.0 (送り速度停止)			x
DB21、… DBX12/16/20.3 (送り速度停止、ジオメトリ軸 1/2/3)			x
DB21、... DBB4; 送り速度オーバライド= 0%			x
DB31、... DBX4.3 (送り/主軸停止)			x

チャネルの初期状態:「有効」			
イベント	次のチャネル状態		
	「リセット」	「中断」	「有効」
DB21、... DBX194.2/DBX197.6 (ブロック内の M02/M30)	x		
DB21、... DBX194.0/DBX194.1 (ブロック内の M00/M01)		x	
DB21、... DBX0.4 (シングルブロック)		x	
DB21、... DBX6.2 (残移動距離削除)			x
PLC に出力されたが、まだ応答されていない補助機能			x
プログラム内の WAIT 命令			x

「有効」チャネル状態は、NC プログラムまたは NC プログラムブロックの実行中に、または JOG モードでの軸の移動時に実現します。

10.9.7 オペレータアクションおよびプログラムアクションへの応答

状態遷移

次の表では、特定のオペレータとプログラム動作後のチャネル状態とプログラム状態を示しています。

表の左側の部分は、チャネルおよびチャネルで選択したプログラムのさまざまな状態、および有効な運転モードの一覧です。

表の右側の部分は、運転/プログラム動作と次の状態の一覧です。

ステータス	チャネル状態			プログラム状態					運転モード			運転またはプログラム動作=> 次の状態
	R	U	A	R	U	S	W	A	A	M	J	
1		x						x	x			リセット⇒ 4
2		x						x		x		リセット⇒ 5
3		x						x			x	リセット⇒ 6
4	x			x					x			NC スタート⇒ 13; モード変更⇒ 5 または 6
5	x			x						x		NC スタート⇒ 14; モード変更⇒ 4 または 6

ステータス	チャネル状態			プログラム状態					運転モード			運転またはプログラム動作=> 次の状態
	R	U	A	R	U	S	W	A	A	M	J	
6	x			x							x	方向キー⇒ 15; モード変更⇒ 4 または 5
7		x		x						x		NC スタート⇒ 14
8		x		x							x	NC スタート⇒ 15
9		x			x				x			NC スタート⇒ 13; モード変更⇒ 10 または 11
10		x			x					x		NC スタート⇒ 16; モード変更⇒ 9 または 11
11		x			x						x	方向キー⇒ 17; モード変更⇒ 9 または 10
12		x				x			x			NC スタート⇒ 13; モード変更⇒ 10 または 11
13			x					x	x			NC スタート⇒ 12
14			x	x						x		NC ストップ⇒ 7; ブロックの終点の場合⇒ 5
15			x	x							x	NC ストップ⇒ 8; JOG の終点の場合⇒ 6
16			x		x					x		NC ストップ⇒ 10; ブロックの終点の場合⇒ 10
17			x		x						x	NC ストップ⇒ 11; JOG の終点の場合⇒ 11
18			x				x		x			リセット⇒ 4; 他のチャネルを待機⇒ 18

チャネル状態

R →中止

U →中断

A →実行中

プログラム状態

R →中止

U →中断

S →停止

W →待機中

A →実行中

モード

A →自動

M → MDI

J → JOG

10.9.8 プログラム実行のタイミングチャートの例

プログラムコード

```
N10 G01 G90 X100 M3 S1000 F1000 M170
N20 M0
```

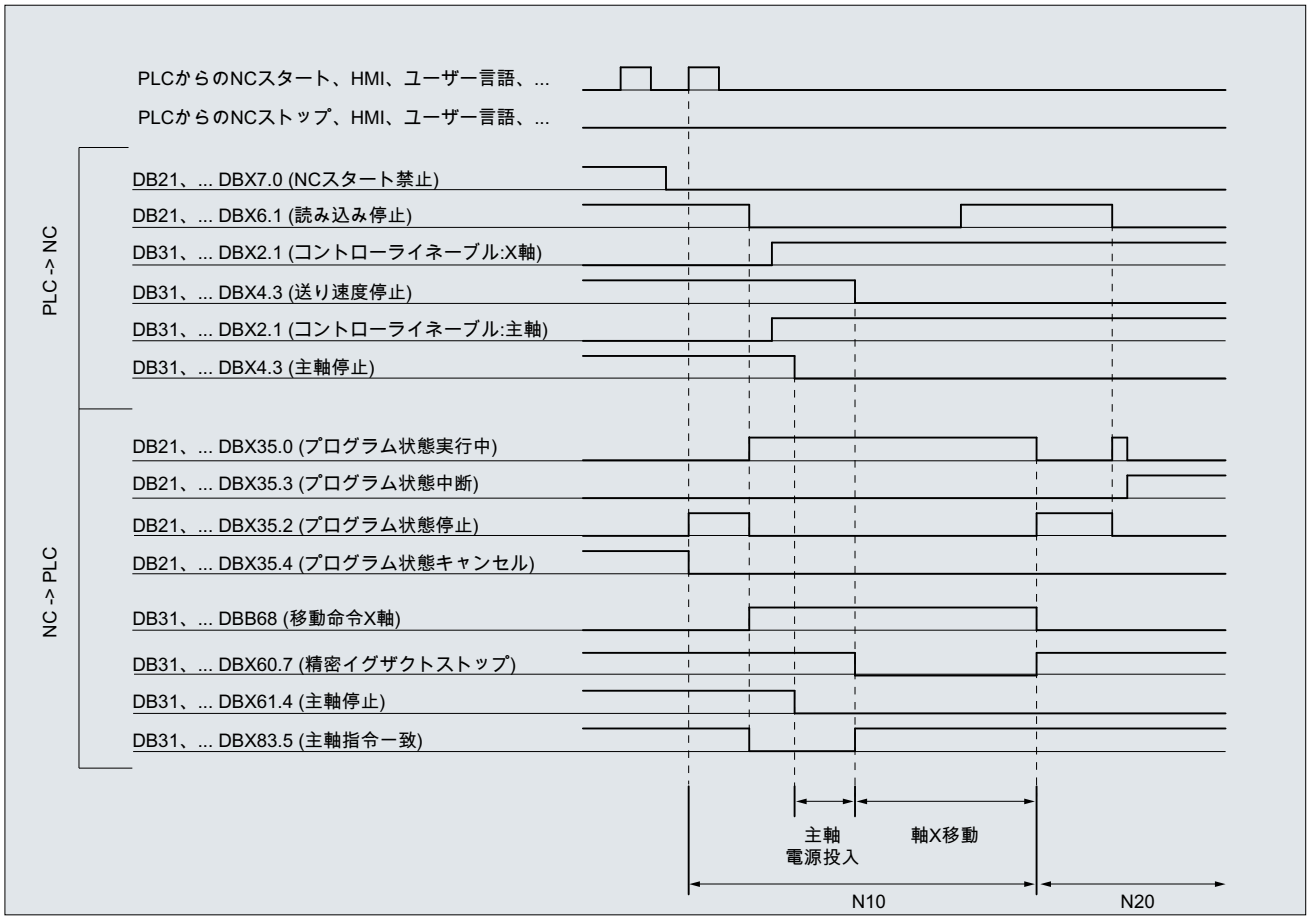


図 10-6　プログラムでの信号特性

10.9.9　プログラムジャンプ

10.9.9.1　プログラムの先頭への復帰ジャンプ(GOTOS)

機能

「プログラム先頭への戻りジャンプ」機能を使用すると、制御装置はパートプログラムからそのプログラムの先頭にジャンプして戻ることができます。その後、プログラムは再度処理されます。

プログラムの繰り返し処理も実行できる「ジャンプマーカへのプログラムジャンプ」機能と比較すると、「プログラム先頭への戻りジャンプ」機能には次の利点があります。

- プログラムの先頭でのジャンプマークのプログラミングは不要です。
- プログラムの再起動は、次の NC/PLC インタフェース信号を介して制御できます。
  DB21, ... DBX384.0 (プログラム分岐の制御)
- プログラム実行時間用のタイマは、プログラムの再起動時に「0」にリセットできます。
- ワークカウンタ用のタイマは、プログラムの再起動時に「1」ずつ加算できます。

用途例

この機能が使用されるのは、たとえば、バーローダ/チェンジャ付きの旋盤の場合など、後続のワークの加工が自動プログラム再起動を介して行われる場合です。

NC/PLC インタフェース信号

戻りジャンプが行われるのは、次の NC/PLC インタフェース信号が設定された場合だけです。

DB21、... DBX384.0 (プログラム分岐の制御) = **1**

パラメータ設定

プログラム実行時間

選択された NC プログラムの実行時間は、システム変数$AC_CYCLE_TIME に格納されます。新しいプログラムを起動すると、このシステム変数は自動的に「0」にリセットされます(「プログラム実行時間 (ページ 817)」の章を参照)。

「プログラム先頭への戻りジャンプ」を介してプログラムを再起動する場合にも、次のマシンデータを介して、システム変数$AC_CYCLE_TIME が「0」にリセットされるように設定できます。

MD27860 $MC_PROCESSTIMER_MODE.ビット 8 = <値> (プログラム実行時間計測を起動)

ビット	値	意味
8	0	「プログラム先頭への戻りジャンプ」機能によって$AC_CYCLE_TIME は「0」にリセット**されません**。
	1	「プログラム先頭への戻りジャンプ」機能によって$AC_CYCLE_TIME は「0」にリセットされます。

注記

ビット 8 の設定が有効になるようにするには、現在のプログラム実行時間の計測を有効にしてください(MD27860 ビット 1 = 1)。

ワークカウント

パートプログラムの末尾(M02/M30)に到達後に、有効なワークカウンタ ($AC_TOTAL_PARTS / $AC_ACTUAL_PARTS / $AC_SPECIAL_PARTS)が「1」ずつ加算されます(「ワークカウンタ (ページ 832)」の章を参照)。

「プログラム先頭への戻りジャンプ」機能を介してプログラムを再起動する場合にも、次のマシンデータを介して、有効なワークカウンタが加算されるように設定できます。

MD27880 $MC_PART_COUNTER.ビット<n> = <値> (ワークカウンタの起動)

ビット	値	意味「プログラム先頭への戻りジャンプ」機能を介してプログラムを再起動する場合、ワークカウンタは次のように処理されます。
7	0	$AC_**TOTAL**_PARTS は**加算されません**。
	1	$AC_**TOTAL**_PARTS は加算されます。
11	0	$AC_**ACT**_PARTS は**加算されません**。
	1	$AC_**ACT**_PARTS は加算されます。
15	0	$AC_**SPECIAL**_PARTS は**加算されません**。
	1	$AC_**SPECIAL**_PARTS は加算されます。

プログラミング

構文

GOTOS

意味

GOTOS:	現在のプログラムの先頭への復帰	
	先読み停止:	あり
	効果:	ノンモーダル

例:

プログラミング	コメント
N10 ...	; プログラムの開始
...	
状況	
N100 GOTOS	プログラム起動への復帰(N10)
ENDIF	
...	
RET	

10.9.10 プログラムセクションの繰り返し

10.9.10.1 プログラミング

プログラム区間の繰り返しでは、NC プログラム内でのプログラム区間の繰り返しが可能です。

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.REPEATB:単一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.REPEAT +ジャンプマーク: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.REPEAT +ジャンプマーク_1 +ジャンプマーク_2:2 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.REPEAT +ジャンプマーク+ ENDLABEL: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。 • REPEATB:繰り返すプログラム行 • REPEAT:繰り返すプログラム区間の先頭
	<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に開始されます。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に続行されます。 **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に**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、それらの命令が繰り返し毎に実行されます。
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。 **注記:** P=<n>に数字を指定しない場合は、プログラム区間が 1 回だけ繰り返されます。
<n>:	繰り返し回数
	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 命令の次のプログラム行からプログラムを再開します。

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	; プログラム区間の開始
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	; プログラム区間の開始
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y7=R10	
N50 X=-R10	
N60 Y=-R10	
N70 END:Z = 10	; プログラム区間の終了
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:	; プログラム区間 1 の開始
N30 X10	
N40 Y10	
N50 BEGIN2:	; プログラム区間 2 の開始
N60 X20	
N70 Y30	
N80 ENDLABEL:Z10	
N90 X0 Y0 Z0	
N100 Z-10	
N110 BEGIN3:X20	; プログラム区間 3 の開始
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 :フライス削り、さまざまなテクノロジによる穴あけ位置

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:	; プログラム区間 1 の開始; 穴あけ位置 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	
N70 ENDLABEL:	
N80 POS_2:	; プログラム区間 2 の開始; 穴あけ位置 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	
N120 ENDLABEL:	; プログラム区間 1 および 2 の終了
N130 DRILL()	; 穴あけサイクル
N140 THREAD(6)	; ねじ切りサイクル
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()	; 穴あけサイクル
N170 THREAD(8)	; ねじ切りサイクル
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間に、M17 または RET をプログラム指令した場合は、プログラム区間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。
- プログラム区間処理中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、この 2 つが重複し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。

10.9.11 イベント起動プログラム呼び出し(PROG_EVENT)

10.9.11.1 機能

機能

NC (ユーザー別 NC プログラム(PROG_EVENT プログラム))で選択されたイベントが発生した場合は、「イベント起動プログラム呼び出し」(PROG_EVENT)機能が起動します。

用途例

機能の初期設定、またはシステム変数またはユーザー変数の初期化。

イベント

起動イベントは、マシンデータ MD20108 $MC_PROG_EVENT_MASK を使用して選択します(「パラメータ設定 (ページ 683)」の章を参照してください)。

PROG_EVENT プログラム

PROG_EVENT プログラムの名前は、マシンデータ MD11620 $MN_PROG_EVENT_NAME で設定されます(「パラメータ設定 (ページ 683)」の章を参照してください)。

PROG_EVENT プログラムは、イベントが発生したチャネルで実行されます。

PROG_EVENT プログラムは最低優先順位で実行されるため、ユーザー ASUB で中断できます。

処理順

イベントによって起動された場合の処理順:プログラムの起動

初期状態:

チャネル: リセット状態
モード: 自動(オプション:オーバストア(有効)
MDI
TEACH IN

1. チャネル内のプログラム開始
2. 次の MD を参照した初期化処理
   - MD20112 $MC_START_MODE_MASK
3. サブプログラムとしての PROG_EVENT プログラムの自動呼び出し
4. メインプログラムのデータ部分の処理
5. メインプログラムのプログラム部分の処理

イベントによって起動された場合の処理順:プログラム終了

初期状態

チャネル: 有効
モード: 自動(オプション:オーバストア(有効)
MDI
TEACH IN

1. チャネルで、プログラムブロックの終了が、実行されたプログラム内で交換されます。
2. プログラム終了が実行され、以下のマシンデータが参照されます。
   - MD20110 $MC_RESET_MODE_MASK
   - MD20150 $MC_GCODE_RESET_VALUES
   - MD20152 $MC_GCODE_RESET_MODE

3. ASUB としての PROG_EVENT プログラムの自動呼び出し
4. チャネルでリセットが実行され、以下のマシンデータが参照されます。
   – MD20110 $MC_RESET_MODE_MASK
   – MD20150 $MC_GCODE_RESET_VALUES
   – MD20152 $MC_GCODE_RESET_MODE

イベントによって起動された場合のシーケンス:チャネルリセット

初期状態:

チャネル: 指定なし
モード: 指定なし

1. コントローラが次のマシンデータを参照してリセット処理を起動します。
   – MD20110 $MC_RESET_MODE_MASK
   – MD20150 $MC_GCODE_RESET_VALUES
   – MD20152 $MC_GCODE_RESET_MODE
2. ASUB としての PROG_EVENT プログラムの自動呼び出し
3. コントローラが次のマシンデータを参照してリセット処理を起動します。
   – MD20110 $MC_RESET_MODE_MASK
   – MD20150 $MC_GCODE_RESET_VALUES
   – MD20152 $MC_GCODE_RESET_MODE

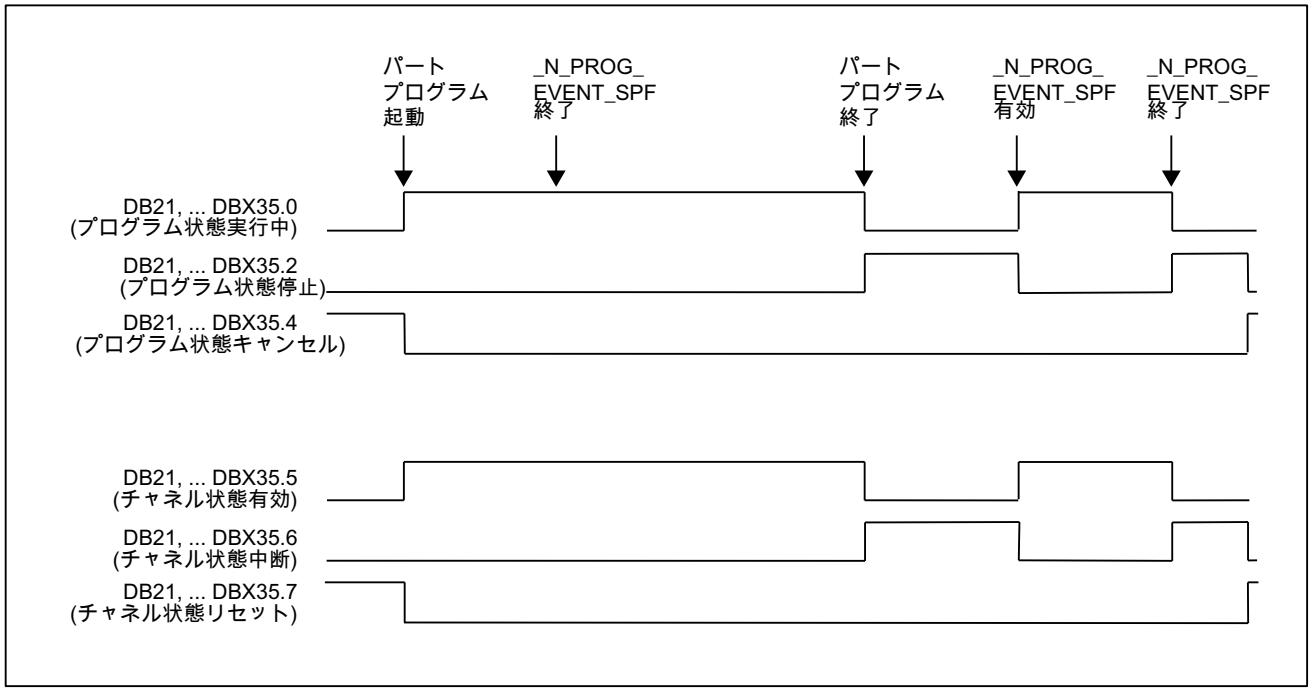
イベントによって起動された場合のシーケンス:NC 電源の投入

1. コントローラが電源投入後に次のマシンデータを参照してリセット処理を起動します。
   – MD20110 $MC_RESET_MODE_MASK
   – MD20150 $MC_GCODE_RESET_VALUES
   – MD20152 $MC_GCODE_RESET_MODE
2. ASUB としての PROG_EVENT プログラムの自動呼び出し
3. コントローラが次のマシンデータを参照してリセット処理を起動します。
   – MD20110 $MC_RESET_MODE_MASK
   – MD20150 $MC_GCODE_RESET_VALUES
   – MD20152 $MC_GCODE_RESET_MODE

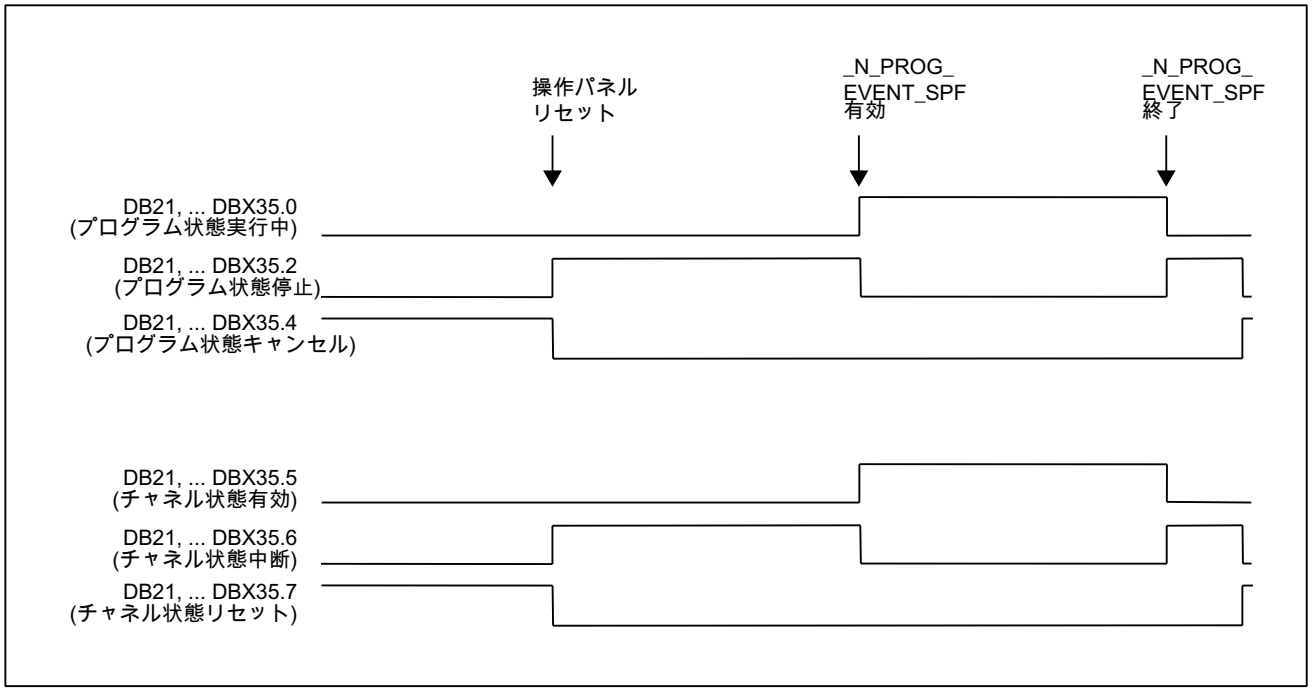
NC/PLC インターフェース信号、「プログラム状態」と「チャネル状態」の変更

以下の図は、イベント起動プログラム呼び出しシーケンス時の「プログラム状態」と「チャネル状態」のさまざまな NC/PLC インターフェース信号の変更を示しています。

プログラム起動とプログラム終了による起動時の信号シーケンス



チャネルリセットによる起動時の信号シーケンス



NC/PLC インタフェース信号:DB21、... DBX35.4 (プログラム状態中止)および DB21、... DBX35.7 (チャネル状態リセット)

- インターフェース信号は、PROG_EVENT プログラムが再び完了したときにのみ設定されます。
- インターフェース信号は、以下の場所には設定されません。
  - プログラム終了と PROG_EVENT プログラム開始の間
  - チャネルリセットと PROG_EVENT プログラム開始の間

NC/PLC インターフェース信号 DB21、... DBX376.0 ... 4 (起動イベント)

開始イベントは、インターフェース信号 DB21、... DBX376.0 ... 4 を使用して表示されます。

ビット	値	イベント
0	1	チャネル状態「リセット」からのプログラム起動
1	1	プログラムの終了
2	1	チャネルリセット
3	1	NC 電源の投入
4	1	1.ブロック検索後のプログラム起動(「ブロック検索後の ASUB の自動起動 (ページ 607)」を参照)

PROG_EVENT プログラムが完了したか、チャネルリセットによりキャンセルされた場合は、インターフェース信号が再びクリアされます。

インターフェース信号は、少なくとも PLC 周期の期間中は使用可能です。

10.9.11.2 パラメータ設定

イベント選択

PROG_EVENT プログラムを起動するイベントは、以下のマシンデータを使用してチャネル別に設定します。

MD20108 $MC_PROG_EVENT_MASK.ビット<n> = 1

ビット	<値>	意味イベント
0	1	パートプログラム起動
1	1	パートプログラムの終了
2	1	チャネルリセット
3	1	電源投入
5	1	電源投入時の安全プログラム

注記

MD20108 $MC_PROG_EVENT_MASK は、シミュレーション中は参照されません。

PROG_EVENT プログラム

The PROG_EVENT プログラム(初期設定: _N_PROG_EVENT_SPF)をロードし、起動してください。

初期設定

初期設定では、イベントの発生時にユーザープログラム_N_CMA_DIR/ **_N_PROG_EVENT_SPF** が実行されます。

The PROG_EVENT プログラムをロードし、起動してください。

ユーザー別の設定

イベントに対して、初期設定とは異なる PROG_EVENT プログラムを実行する場合は、以下のマシンデータに NC プログラム名を入力してください。

MD11620 $MN_PROG_EVENT_NAME = <プログラム名>

検索パス

PROG_EVENT プログラムは、以下のサイクルディレクトリのいずれかに存在しなければなりません。以下の検索パスが、設定されたイベントの発生時に渡されます。

1. /_N_CUS_DIR/ (ユーザーサイクル)
2. /_N_CMA_DIR/ (メーカーサイクル)
3. /_N_CST_DIR/ (標準サイクル)

MD11620 $MN_PROG_EVENT_NAME で指定された名前で見つかった最初のプログラムが実行されます。

注記

- 指定されたプログラム名は、サブプログラム名称の場合と同様に、構文的にチェックされます。つまり、最初の 2 文字は文字かアンダーバー(数字は不可)にしてください。プログラム名称の接頭語(_N_)と接尾語(_SPF)は、指定されていない場合、自動的に追加されます。
- サイクルについて有効にできるのと同じ保護メカニズム(書き込み、読み取りなどに対する保護レベル)が適用されます。

ユーザー ASUB の起動時の動作

チャネルリセット状態からのユーザー ASUB の起動時における「イベント起動プログラム呼び出し」機能の動作は、次のマシンデータを使用してチャネル別に設定できます。

MD20109 $MC_PROG_EVENT_MASK_PROPERTIES.ビット 0 = <値>

ビット	値	意味
0	0	PROG_EVENT プログラムが、MD20108 で設定されたイベントの 1 つが発生したときに起動されます。
	1	PROG_EVENT プログラムが、MD20108 で設定されたイベントの 1 つが発生したときに**起動されません**。

シングルブロック処理が設定されている場合の動作

設定されているシングルブロック処理の「イベント起動プログラム呼び出し」機能の動作は、以下のマシンデータでチャネル別に設定できます。

MD20106 $MC_PROG_EVENT_IGN_SINGLEBLOCK.ビット<n> = <値>

ビット	値	意味 PROG_EVENT プログラムでのイベントのシングルブロック処理
0	0	パートプログラム起動:**機能します**
	1	パートプログラム起動:**マスク**されます
1	0	パートプログラム終了:**機能します**
	1	パートプログラム終了:**マスク**されます

ビット	値	意味 PROG_EVENT プログラムでのイベントのシングルブロック処理
2	0	チャネルリセット:**機能します**
	1	チャネルリセット:**マスク**されます
3	0	電源投入:**機能します**
	1	電源投入:**マスク**されます

シングルブロック処理がマスクされている場合は、PROG_EVENT プログラムが中断なしで処理されます。

注記

- MD20106 $MC_PROG_EVENT_IGN_SINGLEBLOCK は、すべてのシングルブロック処理タイプに影響を与えます。
- PROG_EVENT プログラムのシングルブロック処理は、以下の設定で無効にすることができます。
  MD10702 $MN_IGNORE_SINGLEBLOCK_MASK、ビット 0 = 1 (シングルブロック停止のマスク)
  これで、MD20106 $MC_PROG_EVENT_IGN_SINGLEBLOCK の設定が機能しなくなります。

読み込み停止が設定されている場合の動作

読み込み停止が設定されている(DB21、... DBX6.1 = 1)場合の「イベント起動プログラム呼び出し」機能の動作を、下記のマシンデータで開始イベント毎にチャネル別に設定できます。

MD20107 $MC_PROG_EVENT_IGN_INHIBIT. ビット**<n> = <値>**

ビット	値	意味 PROG_EVENT プログラムでのイベントの読み込み停止
0	0	パートプログラム起動:**機能します** PROG_EVENT プログラムを RET 命令で終了した場合は、実行可能なブロックが**作成されません**。
	1	パートプログラム起動:**マスク**されます PROG_EVENT プログラムを RET 命令で終了した場合は、M17 と同様の実行可能なブロックが作成されます。
1	0	パートプログラム終了:**機能します**
	1	パートプログラム終了:**マスク**されます

ビット	値	意味 PROG_EVENT プログラムでのイベントの読み込み停止
2	0	チャネルリセット:**機能します**
	1	チャネルリセット:**マスク**されます
3	0	電源投入:**機能します**
	1	電源投入:**マスク**されます

プログラム状態とチャネル状態表示の更新マスク

プログラム状態とチャネル状態が HMI ユーザーインターフェースに表示されたときのちらつきをなくすには、通常は非常に短い PROG_EVENT プログラムの実行によって表示が更新されないようにすることができます。その場合、PROG_EVENT プログラムが起動する前のプログラム状態とチャネル状態が表示されたままになります。

MD20192 $MC_PROG_EVENT_IGN_PROG_STATE.ビット<n> = <値>

ビット	値	意味 PROG_EVENT プログラムの処理中の、イベントのプログラム状態とチャネル状態の表示の更新
1	0	パートプログラム終了:**マスクされません**
	1	パートプログラム終了:**マスクされます**
2	0	チャネルリセット:**マスクされません**
	1	チャネルリセット:**マスクされます**
3	0	電源投入:**マスクされません**
	1	電源投入:**マスクされます**

注記

システム変数$AC_STAT と$AC_PROG はこの機能の影響を受けません。つまり、実行中のイベント起動ユーザープログラム内では、$AC_STAT は「有効」に設定され、$AC_PROG は「実行中」に設定されます。

NC/PLC インタフェース信号 DB21、... DBX35.0-7 (「プログラム状態..」と「チャネル状態...」)も影響を受けません。

DB21、... DBX7.2 / 3 / 4 (NC ストップ...)の応答

「NC ストップ」、「ブロック境界での NC ストップ」、および「NC ストップ、軸および主軸」の場合の「イベント起動プログラム呼び出し」機能の応答を、以下のマシンデータで、パートプログラム終了、チャネルリセット、および電源投入の各イベントに対してチャネル別に設定できます。

MD20107 $MC_PROG_EVENT_IGN_STOP.ビット**<n> = <**値**>**

ビット	値	**意味 NC ストップの PROG_EVENT プログラムおよびイベント**
1	0	パートプログラム終了:**停止/マスクされます**
	1	パートプログラム終了:**完全に処理されます**
2	0	チャネルリセット:**停止/マスクされます**
	1	チャネルリセット:**完全に処理されます**
3	0	電源投入:**停止/マスクされます**
	1	電源投入:**完全に処理されます**

用途例

チャネルリセット時または電源投入時に NC ストップキーを押すことによってオペレータが開始したインターフェース信号 DB21、... DBX7.3 (NC ストップ)のエッジ変化は、PROG_EVENT プログラムの実行中は無視されます。そのため、機械での不要な停止動作は防止されます。

注記

PROG_EVENT プログラムでの DELAYFSTON/DELAYFSTOF のプログラミングを、MD20193 で設定された動作に置き換えることはできません。置き換えても、DELAYFSTON 命令の実行前の NC ストップにより中断される場合があります。

10.9.11.3 プログラミング

PROG_EVENT プログラム

プログラムの終了

パートプログラムの起動を介してユーザープログラムが起動される場合は、次のことを留意してください。

- ユーザープログラムは M17 または RET で終了しなければなりません。
- REPOS 命令による復帰ジャンプは許容されません。

ブロック表示

PROC 文で DISPLOF 属性を使用して、現在のブロック表示で表示をマスクできます。

PLC ユーザープログラムへの通知

PROG_EVENT プログラムにプログラム指令されたユーザー M 機能で、PROG_EVENT プログラムの処理状態などについて PLC ユーザープログラムに知らせることができます。

システム変数

起動イベントの確認

開始イベントは、以下のシステム変数を使用して PROG_EVENT プログラムで確認できます。

<値> = $P_PROG_EVENT (イベント起動プログラム呼び出し有効)

値	意味起動イベント
1	**パートプログラム起動**
2	**パートプログラムの終了**
3	**チャネルリセット**
4	**電源投入**
5	**ブロック検索**後の最後のアクションブロックの出力後(「ブロック検索後の ASUB の自動起動 (ページ 607)」を参照)

現在のチャネルの確認

PROG_EVENT プログラムが処理されるチャネルは、以下のシステム変数で特定できます。

<値> = $P_CHANNO (現在のチャネル番号の確認)

注記

PROG_EVENT プログラムは、開始イベントが発生したチャネルで処理されます。
電源投入は、すべてのチャネル内で同時に発生するイベントです。

10.9.11.4 境界条件

非常停止/アラーム

非常停止またはモードグループ/NC 別アラームがチャネルリセットに対して、または電源投入後に発生した場合は、非常停止または異常についてすべてのチャネルが確認応答した後にのみ、PROG_EVENT プログラムが処理されます。

注記

「電源投入」イベントは、すべてのチャネルで同時に発生します。

10.9.11.5 例

例 1:すべてのイベントについて PROG_EVENT プログラムを呼び出す

パラメータ設定

MD20108 $MC_PROG_EVENT_MASK = 'H0F'　次について_N_PROG_EVENT_SPF を呼び出し

- パートプログラム起動
- パートプログラムの終了
- チャネルリセット
- 電源投入

プログラミング

プログラムコード	コメント
PROC PROG_EVENT DISPLOF	
; パートプログラム起動の処理	
IF ($P_PROG_EVENT==1)	
MY_GUD_VAR=0	; GUD 変数の初期化
RET	

プログラムコード	コメント
ENDIF	
; パートプログラム終了とチャネルリセットの処理	
IF ($P_PROG_EVENT==2) OR ($P_PROG_EVENT==3)	
DRFOF	; DRF オフセットの解除
IF $MC_CHAN_NAME=="CHAN1"	
CANCEL(2)	; モーダルなシンクロナイズドアクション 2 のクリア
ENDIF	
RET	
ENDIF	
; 電源投入の処理	
IF ($P_PROG_EVENT==4)	
IF $MC_CHAN_NAME=="CHAN1"	
IDS=1 EVERY $A_INA[1]>5.0 DO $A_OUT[1]=1	
ENDIF	
RET	
ENDIF	
RET	

例 2: チャネルリセットについて PROG_EVENT プログラムを呼び出す

パラメータ設定

MD20108 $MC_PROG_EVENT_MASK = 'H04'　次について_N_PROG_EVENT_SPF を呼び出し
- 操作パネルリセット

プログラミング

プログラムコード	コメント
PROC PROG_EVENT DISPLOF	
N10 DRFOF	; DRF オフセットの解除
N20 M17	

例 3 :機能の初期化

パラメータ設定

マシンデータの割り当て、コミッショニングファイルからの抽出(_N_INITIAL_INI)

プログラムコード	コメント
...	

プログラムコード	コメント
CHANDATA(3)	; チャネル 3 用の初期化
$MC_PROG_EVENT_IGN_INHIBIT='H01F'	
$MC_PROG_EVENT_MASK='H04'	; イベント:チャネルリセット
...	

意味

PROG_EVENT プログラム、_N_CMA_DIR/_N_PROG_EVENT_SPF は、チャネルリセットのために 3 番目のチャネルで起動され、読み込み停止が有効か無効かにかかわらず、プログラムの終了まで処理されます。

10.9.12 停止イベントに及ぼす停止遅延領域の影響

10.9.12.1 機能

停止遅延領域

NC プログラムの、条件付きで中断可能な領域は、停止イベントに対する応答に影響を及ぼすことができます。このようなプログラム領域を停止遅延領域と呼びます。

停止遅延領域内では、停止することはできず、送り速度を変更することもできません。停止は、プログラム区間が完了するまでは有効になりません。

これには、以下のメリットがあります:

- プログラム区間では、速度を低下させずに処理されます。
- ユーザーが停止後に RESET でプログラムを中止した場合、中止したプログラムブロックは保護区間の後になります。このプログラムブロックは、その後のブロックサーチの検索ターゲットとして適しています。
- 以下のメイン軸は、停止遅延領域が処理中であるかぎり、停止しません。
  - コマンド軸
  - POSA で移動する位置決め軸

用途

例:ねじの加工。

定義

停止遅延領域の定義は、パートプログラム内で予約手順 DELAYFSTON および DELAYFSTOF によって行われます(「プログラミング (ページ 696)」を参照してください)。

停止事象

停止を引き起こすイベントの一覧:

イベント	応答
DB21、... DBX7.7 (リセット)または DB11 DBX20.7 (モードグループリセット)	即時
NC プログラム:M30	アラーム 16954
ユーザーの割り込みによる ASUB の起動: FC 9 または DB10 DBB1 (PLC のディジタル NC 入力の設定)	遅延
DB21、... DBX6.2/DB31、... DBX2.2 (残移動距離のクリア)チャネル/軸別	即時
DB21、... DBX6.3 (サブプログラム繰り返しの数のクリア)	遅延
DB21、... DBX6.4 (プログラムレベル中止)	遅延
シングルブロックによる停止 ● 停止遅延領域内: NC は、停止遅延領域外の最初のブロックの終点で停止します。 ● シングルブロックが停止遅延領域の前で有効です。 NC は、各ブロック境界で停止します。つまり、停止遅延領域でも停止します。 DB21、... DBX7.2 (ブロック境界での NC ストップ) **これにより、停止遅延領域が解除されます。**	停止遅延
DB11 DBX21.7 (シングルブロックタイプ A の起動)	遅延
DB11 DBX21.6 (シングルブロックタイプ B の起動)	遅延
DB21、... DBX7.4 (NC ストップ)および DB11 DBX0.6 (モードグループ停止、軸および主軸)	即時
DB21、... DBX7.3 (ブロック境界での NC ストップ)および DB11 DBX0.5 (モードグループ停止)	遅延
NC プログラム:オーバーストアバッファが空であるため停止	アラーム 16954
NC プログラム:プログラム指令された、または自動的な STOPRE	アラーム 16954
NC プログラム:M0 または M1	アラーム 16954
DB21、... DBX7.2 (ブロック境界での NC ストップ)	遅延
サブプログラムの最後で常に、停止遅延領域を解除する必要があります。	システム異常
NC プログラム:WAITM	アラーム 16954

イベント	応答
NC プログラム:WAITE	アラーム 16954
NC プログラム:パラメータ「S」のある INIT	アラーム 16954
NC プログラム:MMC(STRING, CHAR)	アラーム 16954
DB21、... DBB26 (ブロックスキップの起動/解除)	遅延
DB21、... DBB26 (ブロックスキップの解除)	遅延
DB21、... DBX0.6 (ドライランの起動)	遅延
DB21、... DBX0.6 (ドライランの解除)	遅延
DB21、... DBX6.1 (読み込み停止)	遅延
アラーム:アラーム設定 STOPATENDBYALARM	即時
アラーム:アラーム設定 STOPBYALARM	即時
内部:空の IPO バッファに対してアラーム出力後に停止	即時
内部:空の IPO バッファに対してアラーム出力後に停止	直ちに停止
NC プログラム:LFON に対するアラーム 16954	アラーム 16954
NC プログラム:WAITMC	アラーム 16954
NC プログラム:NEWCONF	アラーム 16954
NC プログラム:NEWCONF	アラーム 16954
OPI:PI "_N_SETUDT"	遅延
停止延長と退避	遅延
DB31、... DBX3.0 (外部ゼロオフセットの受け入れ)	停止遅延
OPI:PI "_N_FINDST" STOPRUN	アラーム 16955
NOREADY 応答をともなうアラーム	即時
DB21、... DBX7.2 == 1 (ブロック境界での NC ストップ) DB21、... DBX7.3 == 1 (NC ストップ) DB21、... DBX7.4 == 1 (NC ストップ、軸および主軸)	遅延
DB21、... DBX6.1 == 1 (読み込み停止)	遅延

イベント	応答
DB21、... DBX2.0 == 1 (シングルブロック)	遅延

応答

- **即時**
  停止遅延領域でも、直ちに停止します。「ハード停止イベント」として指定されます。
- **遅延**
  停止遅延領域を過ぎるまで(短時間でも)停止しません。これを「ソフト停止イベント」と呼びます。
- **アラーム 16954**
  停止遅延領域で不適切なプログラム命令が使用されたため、プログラムが中止されました。
- **アラーム 16955**
  停止遅延領域で不適切な動作が発生しましたが、プログラムが続行されます。
- **アラーム 16957**
  DELAYFSTON と DELAYFSTOF で囲まれたプログラム領域(停止遅延領域)を有効にできませんでした。その結果、停止はすべて直ちに実行され、遅延の対象となることはありません。この状況は、減速が停止遅延領域より前で始まり、停止遅延領域内で終わる場合に必ず生じます。同様に、停止遅延領域にオーバライド 0 で進んだ場合も、停止遅延領域を有効にすることはできません。(例:停止遅延領域より前に G4 を使用して、オーバーライドを 0 まで下げることができます。このとき、停止遅延領域内の次のブロックがオーバーライド 0 で始まり、上記のアラーム状況が発生します。)
  注
  MD11411 $MN_ENABLE_ALARM_MASK (警告の起動)ビット 7 によってこのアラームが有効になります。

注記

一部の NC イベントは、切り替え操作をおこなうために短期間だけ停止され、直ちに再開されます。これには、関連付けられた ASUB を起動するために NC プログラムを短期間停止する中断なども含まれます。このようなイベントも停止遅延領域で許容されますが、停止遅延領域の最後まで延期されるため、「ソフト停止イベント」とみなされます。

注記

「ハード」停止イベントと「停止遅延領域」が同時に発生すると、「停止遅延領域」全体が解除されます。このため、このプログラム区間でその他の停止処理があると、直ちに実行されます。新しいプログラム指令(新しい DELAYFSTON)を実行して、新しい停止遅延領域を開始してください。

10.9.12.2 パラメータ設定

マシンデータ

G331/G332 の停止動作

フローティングチャック(G331、G332)を使用しないタッピングでは、停止動作は以下のように設定できます。

MD11550 $MN_STOP_MODE_MASK

ビット	値	意味
0	0 (初期設定)	G331/G332 が有効であり、軌跡移動またはドウェル時間(G4)もプログラムされている場合、自動停止遅延範囲が設定されます。
	1	G331/G332 が有効である場合の停止は、次の状態で使用可能です。 • 連続軌跡モードが中断している(G64)。 • G33x ブロック間でドウェル時間(G4)がプログラムされている。 NC 命令 DELAYFSTON/DELAYFSTOF を使用して停止遅延範囲を定義してください。

10.9.12.3 プログラミング

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

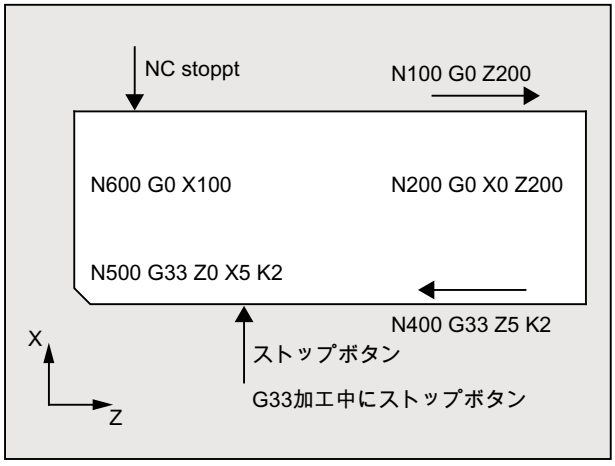
プログラミング例

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
...
```

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。
...	
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

10.9.12.4 必要条件

重複

2 つの停止遅延領域が重複しており、その 1 つが DELAYFSTON/DELAYFSTOF によるもので、もう 1 つが MD11550 $MN_STOP_MODE_MASK によるものである場合は、可能なかぎり最大の停止遅延領域が生成されます。

ドウェル時間(G4)

G4 は、停止遅延領域内で許容されます。その間に停止を引き起こすその他のパートプログラム命令(たとえば、WAITM)は使用できません。このような命令を使用すると、アラーム 16954 が出力されます。

オーバライド

停止遅延領域より**前**でオーバーライドが変更された場合は、オーバーライドが停止遅延領域**内**で有効になります。

停止遅延領域**内**でオーバーライドが変更された場合は、変更が停止遅延領域より**後**で有効になります。

注記

オーバライド = 0

停止遅延領域より**前**でオーバーライドを 0 に下げた場合は、停止遅延領域を有効にすることができません。

送り速度無効

DB21、... DBX6.0 送り無効は、停止遅延領域では効果がありません。プログラムは停止遅延領域の後までは停止しません。

シングルブロック

シングルブロックが停止遅延領域で有効になっている場合は、停止遅延領域外の最初のブロックの終点で NC が停止します。

シングルブロックが停止遅延領域より前ですでに選択されている場合は、各ブロック境界で NC が停止します。つまり、停止遅延領域内でも停止します。**これにより、停止遅延領域が解除されます。**

10.10 非同期サブプログラム(ASUB)

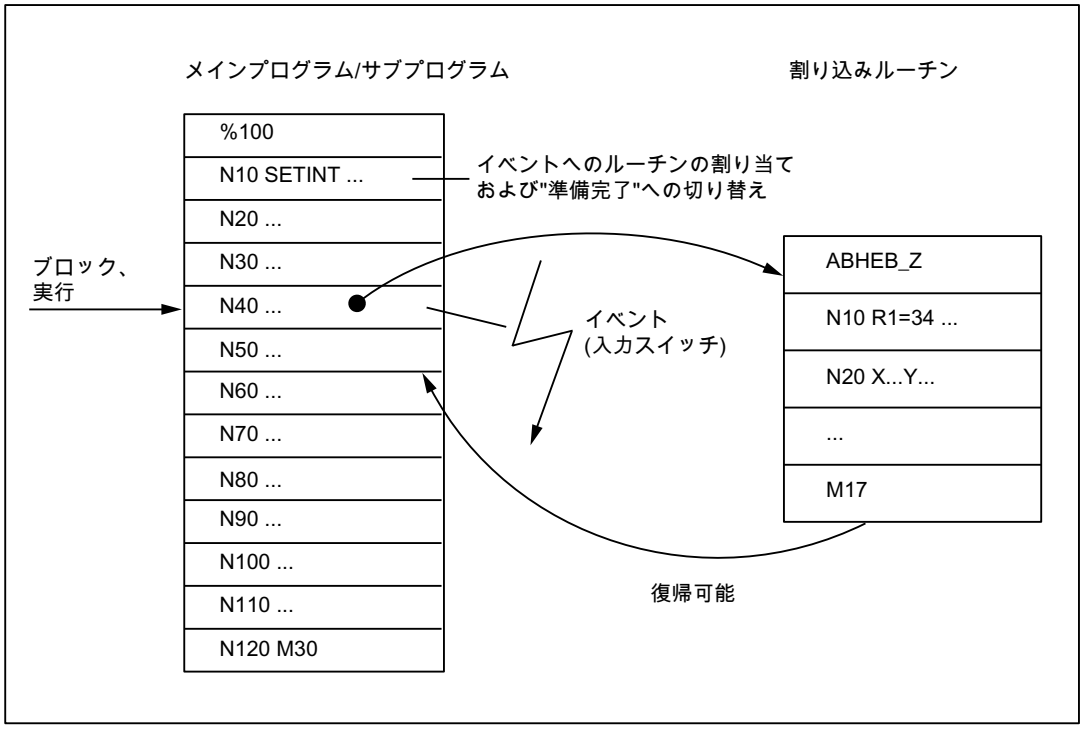
10.10.1 機能

注記

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

概要

非同期サブプログラム(ASUB)は、非同期イベント(割り込み入力信号、プロセス、および機械状態)への応答として NC チャネルで開始される NC プログラムです。ASUB の起動により、現在実行中の NC プログラムが中断します。NC プログラムは、ASUB 終了時に中断位置から続行できます。



チャネルで実行中の NC プログラムは、その全体、またはその一部の区間を、ASUB による中断から保護することができます。「プログラミング(SETINT、PRIO) (ページ 712)」「フレキシブルなプログラミング」の章を参照してください。

定義

NC プログラムから ASUB (割り込みルーチン)を作成するには、SETINT 命令(「プログラミング(SETINT、PRIO) (ページ 712)」の章を参照してください)または「ASUB」PI サービス(「PI サービス ASUB (ページ 1201)」の章を参照してください)で NC プログラムに割り込み信号を割り当ててください。

割り込み信号

- 合計 8 つの I/O 入力を割り込み信号として使用できます。
- I/O 入力信号には、PLC ユーザープログラムを使用して影響を与えることができます。
- 最初の 4 つの I/O 入力デバイスは、NCU モジュールの 4 つの高速入力です。信号状態は、DB10 データブロックで NC/PLC インターフェースを介して読み取ることができます。DB10 データブロックで NC/PLC インターフェースを介して入力信号をロックすることもできます。

高速 NC 入力(割り込み信号)の PLC 制御についての詳細は、「P3: SINUMERIK 840D sl 用 PLC 基本プログラム (ページ 1043)」の章を参照してください。

参照先:
機能マニュアル、上級機能; A4:デジタルとアナログ NC I/O

呼び出し

プログラムモード時

プログラムモード時、つまり、自動または MDI モードでは、常に ASUB を呼び出すことができます。

プログラムモード時以外

プログラムモードでないときは、 以下のモード、機械機能、および状態で ASUB を呼び出すことができます。

- JOG、JOG REF
- MDI TEACH IN、MDI TEACH IN REF、MDI TEACH IN JOG、MDI REF、MDI JOG
- 自動、プログラム状態「停止」、「準備完了」
- 軸状態「原点未確立」

JOG または JOG REF 時に ASUB を開始した場合は、現在の移動が中止されます。

起動

ASUB は、以下を介して起動します。

- 関連付けられた高速 NC 入力での 0/1 エッジで起動された、割り込み信号の 0/1 エッジ
- 「ファンクションコール ASUB」の呼び出し(「P3: SINUMERIK 840D sl 用 PLC 基本プログラム (ページ 1043)」の章を参照してください)
- 短絡を介した割り込みに設定された、シンクロナイズドアクションによる出力の設定(「例 (ページ 716)」の章を参照してください)
  参照先:
  『機能マニュアル、シンクロナイズドアクション』

出力

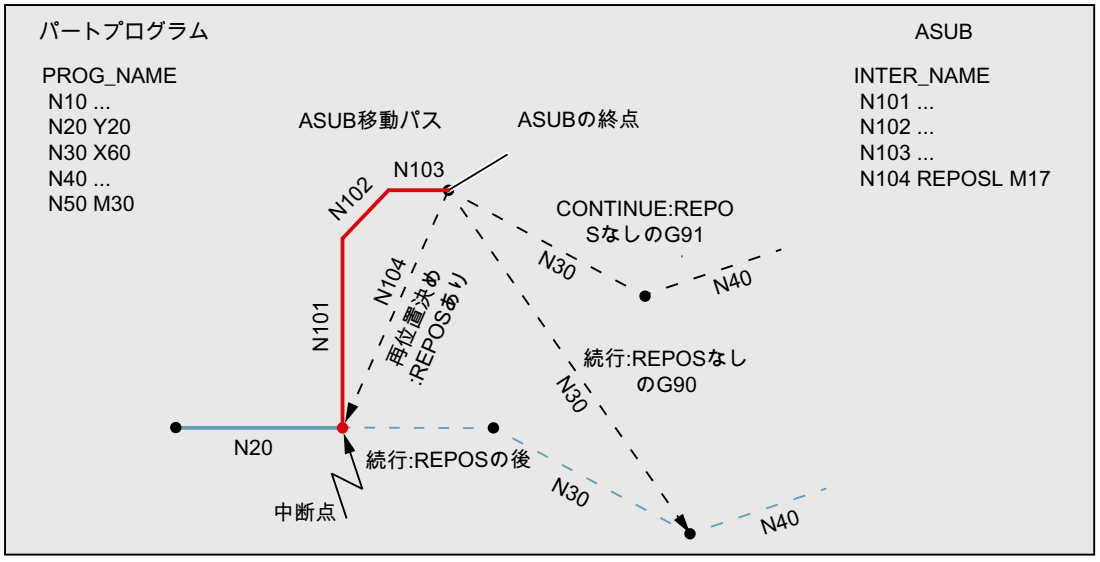
ASUB の起動は、以下の NC/PLC インターフェース信号でチャネル別に表示されます。

DB21、… DBX378.0 == 1 (ASUB が有効)

10.10.1.1 プログラムモードの ASUB の実行シーケンス

1. 軸の減速
   ASUB が起動されると、すべての機械軸が減速カーブ(MD32300 $MA_MAX_AX_ACCEL)に従って減速して停止し、その軸位置が保存されます。
2. 再解析
   軸の減速に加え、あらかじめ解析された計算ブロックが、中断ブロックまでさかのぼって計算されます。つまり、中断時にパートプログラムが事前に解析されていなければ、すべての変数、フレームおよび G 命令に、中断時に持っていたはずの値が割り当てられます。これらの値は記憶されるため、ASUB の終了後も使用可能です。
   次の場合は、例外的に再解析が不可能になります。
   - ねじ切りブロックのとき
   - 複雑な座標計算(たとえば、スプラインまたは径補正)をともなうとき

3. ASUB の処理
   ASUB は、再解析の完了後に自動的に起動します。
   ASUB は、通常のサブプログラム(ネストレベル、など)として処理されます。
4. ASUB の終了
   ASUB の終了コード(M02、M30、M17)が処理されると、初期設定では、中断ブロックに続くパートプログラムブロックにプログラム指令された終了位置まで軸が移動します。
   中断点への復帰位置決めが必要な場合は、たとえば、N104 REPOSL M17 など、ASUP の最後に REPOS 命令がプログラム指令されていなければなりません。



10.10.1.2 REPOSA による ASUB

一般的に、NC プログラムが NC ストップまたはアラームによって停止した場合、REPOSA による ASUP は FC9 を使用して PLC ユーザープログラムから開始され、次のシーケンスが得られます。

- ここでプログラムされた ASUP や移動操作が実行されます。
  - プログラム状態:「停止中」
  - DB21、... DBX318.0 (ASUB 停止) = 1
- 輪郭の再位置決めの前に新しい停止が実行されます(REPOS)。

- オペレータは、輪郭の再位置決めのために NC スタートを開始します(REPOS):
  - DB21、... DBX318.0 (ASUB 停止) = 0
  - 再アプローチの移動が実行されます。
- 再アプローチ動作が終了すると、FC9 確認信号「ASUB 完了」が設定され、中断された NC プログラムの軌跡が継続されます。

注記

NC/PLC インターフェース信号 DB21、... DBX318.0 (ASUB が停止)は、以下の場合にのみ設定します。
「中断」チャネル状態でのプログラムモードの割り込み。

注記

REPOS なしの ASUP では、タイミングの観点から、FC9 確認信号「ASUP 完了」と、NC/PLC インタフェース信号 DB21、... DBX318.0 = 0 (ASUP 停止)のリセットは一致します。

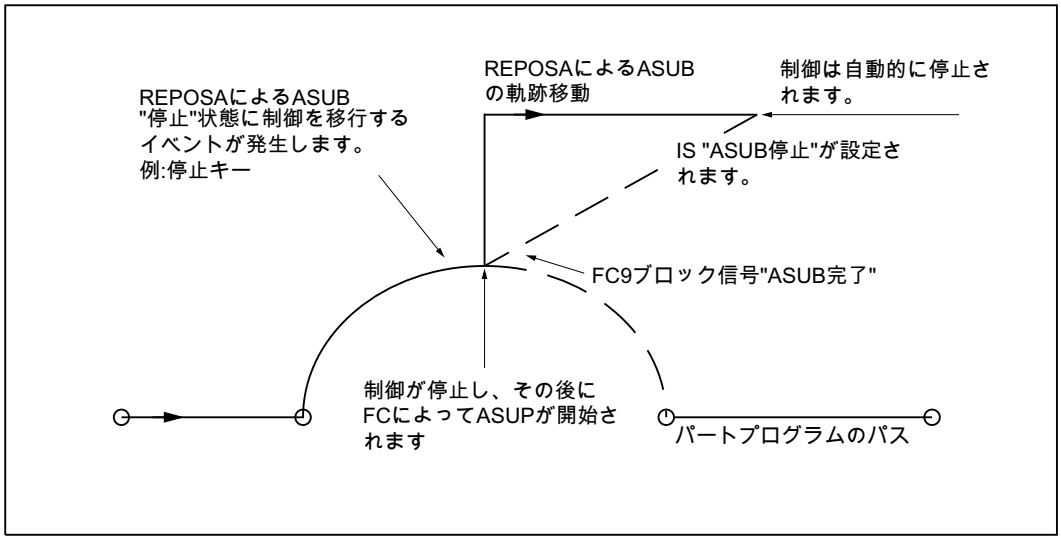


図 10-7　シーケンスの概略図:REPOSA による ASUB

10.10.1.3 NC 動作

起動した ASUB でのチャネル、モードグループ、または NC のさまざまな状態での各種の応答の一覧を、以下の表に示します。

ステータス	ASUB の起動	コントロールシステムの動作
プログラムが動作中	割り込み(PLC)	1. 高速リトラクトまたは軸停止 2. ASUB 中のプログラムの中断 3. ASUB に REPOS が含まれていれば、中断点へのアプローチ 4. パートプログラムの継続
リセット	割り込み(PLC)	ASUB は、メインプログラムと同様に実行されます。ASUB の最後にリセット(M30 を使用しない)が実行されます。その後のコントロールシステムの状態は、次のマシンデータによって決まります。 MD20110 $MC_RESET_MODE_MASK MD20112 $MC_START_MODE_MASK **参照先:** 機能マニュアル 基本機能; 軸、座標系、フレーム(K2)の章、「ワーク関連のフィードバック回路」
プログラムモード(自動または MDI) +チャネル停止	割り込み(PLC)	ASUB が実行されます。ASUB の最後に、再び停止状態になります。 ASUB に REPOS が含まれる場合: ● ASUB の処理がアプローチブロックの前で停止されます。 ● スタートキーでアプローチ動作を開始できます。
	スタートキー	ASUB が実行されると、中断されたプログラムの処理が再開されます。
手動モード +チャネル停止	割り込み(PLC)	コントロールシステムは、指定されたチャネルが「内部プログラム実行モード」状態である(外部的に明確ではない)とみなし、ASUB を起動します。選択された動作モードは有効なままです。ASUB の実行後(M17)、元の状態に戻ります。

ステータス	ASUB の起動	コントロールシステムの動作
JOG AUTO ティーチング AUTO ティーチングレファレンス点。	割り込み(PLC)	処理停止、次のマシンデータを使用： MD11602 $MN_ASUP_START_MASK MD11604 $MN_ASUP_START_PRIO_LEVEL 必要に応じて、「内部プログラム実行モード」への内部切り替え、ASUB の起動、ASUB 起動前の状態への復帰。 SETINT を使用して定義された LIFTFAST は、JOG モードでは有効になりません。
MDI JOG、 MDI ティーチング、 MDI ティーチングレファレンス点。	割り込み(PLC)	
手動モード +チャネル実行中	割り込み(PLC)	現在実行中の動作を停止します。残移動距離は削除されます。残りの動作は、「手動モード+チャネル停止」の場合と同じです。
ユーザーアラーム 65500 - 65999、 チャネルのリセット	割り込み(PLC)	以下のマシンデータを設定した場合は、「NC スタート禁止」アラーム応答にもかかわらず、ユーザー ASUB が処理されます。 MD20194 $MC_IGNORE_NONCSTART_ASUP
INITIAL.INI の処理	不可	信号「割り込み要求不可」が出力されます。
ブロック検索		
NC スタートによって解除できないアラーム		
デジタイジングが有効		
チャネルが異常状態		
65500 - 65999 以外のユーザーアラーム、 チャネルのリセット、 MD20194 が無効	不可	「アラームにより要求がキャンセルされました」信号が生成されます。

10.10.2 試運転:マシンデータ

10.10.2.1 NC 別:モードグループ別の NC/PLC インターフェース信号および操作モード切り替え

次のマシンデータは、DB11 モードグループ別 NC/PLC インタフェース信号、および操作モードの切り替えを実行するチャネルの有効性を指定します。

MD11600 $MN_BAG_MASK = <値>

値	意味
0	DB11 のモードグループ別 NC/PLC インタフェース信号が**有効**です。 内部操作モードの切り替えは、モードグループの**すべての**チャネルで行われます。
1	DB11 のモードグループ別 NC/PLC インタフェース信号が**無効**です。 内部の操作モード切り替えは、**ASUP** が起動しているチャネルで**のみ**実行されます。
2	DB11 のモードグループ別 NC/PLC インタフェース信号が**有効**です。 内部の操作モード切り替えは、**ASUP** が起動しているチャネルで**のみ**実行されます。
3	DB11 のモードグループ別 NC/PLC インタフェース信号が**無効**です。 内部の操作モード切り替えは、**ASUP** が起動しているチャネルで**のみ**実行されます。

注記

マルチチャネルシステム

マルチチャネルシステムで「JOG モードでの ASUP 割り込み中の手動移動」機能を使用可能にする場合(以下を参照)、MD11600 $MN_BAG_MASK を「2」または「3」に設定する必要があります。

下記も参照

プログラミング(SETINT、PRIO) (ページ 712)

10.10.2.2 NC 別:ASUP 起動イネーブル

次のマシンデータで、ASUP 起動で無視する停止理由を指定できます。

MD11602 $MN_ASUP_START_MASK, <ビット> = <値>

ビット	値	意味
0	0	停止キー M0 または M01 による停止が存在する場合、ASUP は起動**されません**。
	1	停止キー M0 または M01 による停止が存在する場合でも、ASUP は**起動**します。
2	0	ASUP の起動時、チャネルに読み込み停止が存在する場合に、チャネル別マシンデータが有効になります。 • MD20107 $MC_PROG_EVENT_IGN_INHIBIT • MD20116 $MC_IGNORE_INHIBIT_ASUP
	1	関連するチャネルに読み込み停止が存在する場合でも、NC のすべてのチャネルでユーザーまたはシステムの ASUP の起動が有効になります。 チャネル別マシンデータの設定が有効では**ありません**。 • MD20107 $MC_PROG_EVENT_IGN_INHIBIT • MD20116 $MC_IGNORE_INHIBIT_ASUP **注** 読み込み停止にかかわらず、NC のすべてのチャネルで起動イネーブルは有効です。これは、ユーザーだけでなく、システム ASUP についても該当します。したがって、NC 別イネーブルではなく**チャネル別**イネーブルを使用することを**強くお勧めします**。
3	0	ASUP が JOG モードで起動され、NC ストップによって中断された場合、この状態では軸を手動で移動することは**できません**。
	1	ASUP が JOG モードで起動され、NC ストップによって中断された場合、この状態では軸を手動で移動することができます。 NC スタートで ASUP を再開できます。この場合、REPOS 操作は自動的に実行されます。 **注** 制御装置で複数のチャネルがパラメータ設定されている場合、次のマシンデータも設定する必要があります。 MD11600 $MN_BAG_MASK、ビット 1 = 1

手動による起動イネーブル

ASUP がパラメータ設定された起動イネーブルにより**自動的に**起動されない場合でも、PLC ユーザープログラムから、または NC スタートの手動作動によって、NC/PLC インタフェース信号(DB21、... DBX7.1)を通じて ASUP を起動できます。

注記

どの場合でも「輪郭からの高速リトラクト」(LIFTFAST)の ASUB が起動されます。

10.10.2.3 NC 別:パラメータ設定した起動イネーブルの有効性

次のマシンデータにより、MD11602 $MN_ASUP_START_MASK の設定が有効になる最高優先順位から、限度となる ASUP 優先順位 (ページ 712)を設定できます。

MD11604 $MN_ASUP_START_PRIO_LEVEL = <ASUP 優先順位>

例

MD11604 $MN_ASUP_START_PRIO_LEVEL = 5

MD11602 $MN_ASUP_START_MASK の設定は、優先順位 1→5 の ASUP で有効です。

10.10.2.4 チャネル別:原点確立されていない軸にかかわらない起動イネーブル

次のマシンデータで、パラメータ設定された「レファレンス点なしで NC スタート無効」機能(MD20700 $MC_REFP_NC_START_LOCK)にかかわらず、関連する ASUP が起動する中断を設定できます。

MD20115 $MC_IGNORE_REFP_LOCK_ASUP、ビット (1 - <中断>) = TRUE

通知

システムの割り込み

MD20115 $MC_IGNORE_REFP_LOCK_ASUP、ビット 8 ～ 31 で、システム中断に割り当てられた ASUP が有効になります。

ビット /割り込み 9 で、移動動作を含む ASUB が開始されます。

NC 別の ASUP 起動イネーブル

MD11602 $MN_ASUP_START_MASK、ビット 2==TRUE である場合、パラメータ設定されたチャネル別「レファレンス点なしで NC スタート無効」機能(MD20700 $MC_REFP_NC_START_LOCK)にかかわらず、ASUP 起動イネーブルが NC のすべてのチャネルで設定されます。したがって、NC 別イネーブルではなく**チャネル別**イネーブルを使用することを**強くお勧めします**。

10.10.2.5 チャネル別:読み込み停止にかかわらない起動イネーブル

次のマシンデータでは、チャネル(DB21, ... DBX6.1)に存在する読み込み停止にかかわらず、関連する ASUP が起動する中断を設定できます。

MD20116 $MC_IGNORE_INHIBIT_ASUP、ビット (1 - <中断>) = TRUE

通知
システムの割り込み MD20116 $MC_IGNORE_INHIBIT_ASUP、ビット 8 ～ 31 で、システム中断に割り当てられた ASUP が有効になります。 ビット/割り込み 9 で、移動動作を含む ASUB が開始されます。 **NC 別の ASUP 起動イネーブル** MD11602 $MN_ASUP_START_MASK、ビット 2==TRUE である場合、NC のすべてのチャネルで MD20116 $MC_IGNORE_INHIBIT_ASUP のチャネル別設定が無視されます。したがって、NC 別イネーブルではなく**チャネル別**イネーブルを使用することを**強くお勧めします**。

10.10.2.6 チャネル別:シングルブロックにかかわらない連続実行

次のマシンデータで、チャネル(DB21、... DBX0.4)の有効なシングルブロック処理にかかわらず、関連する ASUP が**連続的に**、つまりブロック単位の中断なしで実行される中断を設定できます。

MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP、ビット (1 - <中断>) = TRUE

必要条件

MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP の設定は、シングルブロック SBL1(メインランシングルブロック)についてのみ有効です。

通知
システムの割り込み MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP、ビット 8 ～-31 で、システム中断に割り当てられた ASUP が有効になります。 ビット/割り込み 9 で、移動動作を含む ASUB が開始されます。 **NC 別の ASUP 起動イネーブル** MD10702 $MN_ASUP_START_MASK、ビット 1==TRUE である場合、NC のすべてのチャネルで MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP のチャネル別設定が無視されます。したがって、NC 別イネーブルではなく**チャネル別**イネーブルを使用することを**強くお勧めします**。

10.10.2.7 チャネル別:表示の更新

次のマシンデータで、非常に短い ASUP の実行で、ユーザーインタフェースのプログラムおよびチャネルの状態表示の点滅を避けるために、表示を更新しないように設定できます。

MD20191 $MC_IGN_PROG_STATE_ASUP、ビット(1 - <中断>) = TRUE

注記

NC/PLC インタフェース信号

表示の更新をマスクして ASUB を実行する場合は、以下の NC/PLC インターフェース信号を設定します。

DB21、… DBX378.1 = 1 (「サイレント」ASUP 有効)

システム変数と NC/PLC インタフェース信号

プログラム状態とチャネル状態のシステム変数と NC/PLC インターフェース信号は、ASUB の実行中に表示更新のマスクの影響は受けません。

- $AC_STAT (チャネル状態)
- $AC_PROG (プログラム状態)
- DB21、... DBX35.5 - 7 (チャネル状態)
- DB21、... DBX35.0 - 4 (プログラム状態)

10.10.3 プログラミングシステム変数

10.10.3.1 REPOS オプション($P_REPINF)

ASUB と組み合わせて、輪郭上の再位置決め点への一義的な復帰(REPOS)がないプログラム実行シーケンスを生成することができます。

システム変数を使用して、REPOS が可能かどうかを ASUB で確認することができます。

<値> = $P_REPINF

値	意味
0	次の理由で、REPOS による再位置決めができません。 • ASUB で呼び出されていない • ASUB がリセット状態で実行された • ASUB が JOG から実行された
1	ASUB で REPOS による再位置決めが可能です。

10.10.3.2 起動イベント($AC_ASUP)

$AC_ASUP システム変数によって、ASUP を起動するイベントに関する次の情報を確認できます。

- ASUP が起動した理由、たとえばビット 0:ユーザー割り込み「Blsync を使用した ASUP」
- ASUP が起動した方法、たとえばビット 0:NC/PLC インタフェース信号、ディジタル-アナログインタフェース
- 続行を可能にする方法、たとえばビット 0:REORG または RET を自由に選択可能

10.10.4 プログラミング(SETINT、PRIO)

割り当て:NC プログラムへの割り込み

SETI 命令で、NC プログラムを割り込みに割り当てます。これにより、NC プログラムが ASUB になります。

構文

```
SETINT(<n>) <NC プログラム>
```

意味

SETINT:	割り込み信号への NC プログラムの割り当て	
<n>:	割り込み信号番号	
	値の範囲:	0, 1, 2, ... 8
<NC プログラム>:	プログラム名称	

例

プログラムコード	コメント
N20 SETINT(3) LIFT_Z	; IF input 3 == 1
	; THEN start ASUP "LIFT_Z"

SETINT とともに、次の命令もプログラム指令できます。

- LIFTFAST
  割り込み信号が届くと、ASUP の起動前に「輪郭からの工具の高速リトラクト」が実行されます。高速リトラクトの動作方向は、プログラム命令 ALF によって指定されます。
- BLSYNC
  割り込み信号の受信時、現在のプログラムブロックが処理されます。その後にのみ、ASUB が起動されます。

注記

以下の場合、割り込み信号 ↔ パートプログラムの割り当てが解除されます。
- チャネルのリセット状態
- パートプログラム内の CLRINT 命令

優先順位

NC プログラムで SETINT によって複数の割り込みが起動された場合は、割り当てられた NC プログラムまたは ASUB に、異なる優先順位を割り当ててください。

構文
PRIO=<値>

意味

PRIO:	割り込みの優先順位を定義するためのキーワード
<値>:	優先度:1, 2, 3 ... 128.ここで、1 は、最高の優先順位です。

例

プログラムコード	コメント
N20 SETINT(3) PRIO=2 LIFT_Z	; IF input 3 == 1
	; THEN start ASUP "LIFT_Z"
N30 SETINT(2) PRIO=3 LIFT_X	; IF INPUT 2 == 1
	; THEN start ASUP "LIFT_X"

入力 2 および 3 が同時に切り替わった場合は、ASUB が、優先順位の値の順に実行されます。

1. "LIFT_Z"
2. "LIFT_Z"

追加の割り込み別命令

命令	意味
SAVE	ASUP が SAVE 命令を使用すると、割り込み先の NC プログラムで割り込み前に有効であった G 命令、フレーム、および座標変換が、ASUP の最後で再度有効になります。
DISABLE ENABLE	DISABLE ENABLE 命令ペアを設定して、ASUB の割り込みからプログラム区間を保護することができます。 SETINT による、NC プログラムへの割り込み信号の割り当ては保持されます。 そして、ENABLE の後の、割り込み信号の次の 0/1 エッジで ASUB が起動します。
CLRINT(<n>)	SETI で NC プログラムに割り当てられた割り込み信号 n の割り当てを削除します。

参照先

『プログラミングマニュアル、上級編』; :「フレキシブル NC プログラミング」の「割り込みルーチン(ASUB)」の章

下記も参照

プログラミング (ページ 719)

10.10.5 制限事項

モード間の ASUB 起動

チェックする設定

- MD11600 $MN_BAG_MASK
- MD11604 $MN_ASUP_START_PRIO_LEVEL
- 割り込み割り当ての優先順位

推奨設定

NC 別マシンデータ:

- MD11600 $MN_BAG_MASK = 'H3'

  注記

  この設定では、DB11 のモードグループ別 NC/PLC インタフェース信号は、ASUP が実行するチャネルに影響しなくなります。この動作が望ましくない場合は、設定 MD11600 $MN_BAG_MASK = 'H2"を使用することもできます(「試運転:マシンデータ (ページ 706)」を参照)。

  MD11602 $MN_ASUP_START_MASK = 'H5'

- MD11604 $MN_ASUP_START_PRIO_LEVEL = 7

ASUB を起動するチャネルまたは一般的なすべてのチャネルに対するチャネル別マシンデータ

- MD20105 $MC_PROG_EVENT_IGN_REFP_LOCK、ビット<n> = TRUE
  <n>:すべての必要なイベント起動プログラム呼び出し(プログラムイベント)用
- MD20115 $MC_IGNORE_REFP_LOCK_ASUP、ビット<n> = TRUE
  <n>:すべての**必要なユーザー**の割り込み用

通知
システムの割り込み MD20115 $MC_IGNORE_REFP_LOCK_ASUP、ビット 8 ～ 31 でシステムの割り込みが可能になります。 ビット /割り込み 9 で、移動動作を含む ASUB が開始されます。

10.10.6 例

シンクロナイズドアクションからの割り込みによる ASUB の起動

1. 有効なディジタル I/O バイト数を 2 に設定します。
   - MD10350 $MN_FASTIO_DIG_NUM_INPUTS = 2
   - MD10360 $MN_FASTIO_DIG_NUM_OUTPUTS = 2
2. 出力 9 から入力 9 への OR 演算で短絡を設定します。
   - 入力 1、入力バイト 2 = (出力 1、出力バイト 2) **OR** (ハードウェア入力信号 1、入力バイト 2):
     MD10361 $MN_FASTIO_DIG_SHORT_CIRCUIT[0] = 'H0102B102'
3. ハードウェア入力バイトを SETINT 割り込みプログラミングに割り当てます。
   - 入力バイト 2:
     MD21210 $MC_SETINT_ASSIGN_FASTIN = 2
4. 入力を ASUB トリガとして定義します。
   - 2 番目の入力バイトの入力 1、つまり、絶対入力 9、SYNCASUP プログラムの起動
     ```
     SETINT(1) PRIO=1 SYNCASUP
     ```
5. 出力を設定するシンクロナイズドアクションを定義します。
   - ID 1 のシンクロナイズドアクションでは、統一された軌跡パラメータの値が 0.5 以上になると、常に出力 9 が 1 に設定されます。
     ```
     IDS=1 EVERY $$AC_PATHN >= 0.5 DO $A_OUT[9]=1
     ```
     出力 9 から入力 9 への短絡により、割り込み 1 と「SYNC」NC プログラムが ASUB として起動されます。

10.11 RET および REPOS 用のユーザー固有 ASUB

10.11.1 機能

機能

制御ソフトウェアには、当社固有の ASUB が含まれており、NC プログラム終了(RET)機能と輪郭での再位置決め(REPOS)機能を実装しています。工作機械メーカは、システム ASUB の代わりにユーザー固有の ASUB を使用することができます。

⚠ 危険
プログラミング障害 機械メーカのみが、当社固有の ASUB(「ASUP.SYF」)の代わりに使用するユーザー固有の ASUB の正常な動作を保証する責任を負います。

10.11.2 パラメータ設定

インストール

ASUB の名前

ユーザー固有の ASUB は、以下の名前でなければなりません。

- **_N_ASUP_SPF**

ASUB のフォルダ

ユーザー固有の ASUP「_N_ASUP_SPF」は、以下の 2 つのフォルダのいずれかに格納してください。

- **_N_CMA_DIR** (メーカフォルダ)
- **_N_CUS_DIR** (ユーザーフォルダ)

起動および検索パス

以下のマシンデータのビット 0 および 1 で、ユーザー固有の ASUB「_N_ASUP_SPF」の起動により発生するイベントを指定します。

ビット 2 で、ユーザー固有の ASUB「_N_ASUP_SPF」を起動するための検索の開始を指定します。

MD11610 $MN_ASUP_EDITABLE, bit 0, 1, 2 = <値>

ビット	値	意味
0 および 1	0	ユーザー固有の ASUB は、プログラム終了(**RET**)または輪郭での再位置決め(**REPOS**)のどちらに対して起動することもできません。
	1	**ユーザー固有の** ASUB が **RET** に対して起動します。 **システム固有の** ASUB が **REPOS** に対して起動します。
	2	**システム固有の** ASUB が **RET** に対して起動します。 **ユーザー固有の** ASUB が **REPOS** に対して起動します。
	3	**ユーザー固有の** ASUB が、**RET** と **REPOS** に対して起動します。
2	0	ユーザー固有の ASUB が、ユーザーディレクトリ **_N_CUS_DIR** で最初に検索されます。
	1	ユーザー固有の ASUB が、メーカフォルダ **_N_CMA_DIR** で最初に検索されます。

保護レベルの定義

ユーザー固有の ASUB を、RET または REPOS (あるいはその両方)に対して使用する場合(MD11610 $MN_ASUP_EDITABLE ≠ 0)は、ユーザー固有のプログラム「_N_ASUP_SPF」に対して保護レベルを定義できます。この保護レベルとして、0 ～ 7 の値を設定できます。この設定は、次のマシンデータを使用しておこないます。

MD11612 $MN_ASUP_EDIT_PROTECTION_LEVEL <ユーザー固有の ASUB の保護レベル>

保護レベルについて詳しくは、以下を参照してください。

参照先

『試運転マニュアル』、「保護レベルの概念」

シングルブロック処理が設定されている場合の動作

以下のマシンデータで、シングルブロック処理が有効であっても、システム固有の ASUB とユーザー固有の ASUB「_N_ASUP_SPF」を、中断することなく処理することを指定します。

MD10702 $MN_IGNORE_SINGLEBLOCK_MASK、ビット 0 = <値>

ビット	値	意味
0	0	シングルブロックモードが有効な場合は、**すべての ASUB** で停止します。
	1	シングルブロックモードが有効な場合は、ASUB が、**中断されることなく**処理されます。

10.11.3 プログラミング

ASUB の起動の要因の特定

ASUB の起動の要因は、システム変数$AC_ASUP によりビット割り付けされて読み取ることができます。

継続

システム ASUB を使用する場合、動作の実行後に継続するための動作は ASUB 内部で固定値として指定されます。

- システム ASUB 1 → RET による継続(サブプログラムの復帰)
- システム ASUB 2 → REPOS による継続(再位置決め)

システム変数の記述で、システム ASUB に関する動作が「継続」時の要因毎に指定されています。

注記

ユーザー専用 ASUB に対して継続

ユーザー専用 ASUB に対して、システム ASUB の適切な継続を保持してください。

原因: 運転モードの変更($AC_ASUP、ビット 9 == 1)

運転モードの変更時は、継続は次のマシンデータに依存しています。

MD20114 $MC_MODESWITCH_MASK (モード変更による MDI の中断)

- ビット 0 == 0: システム ASUB 1 → RET による継続
- ビット 0 == 1: システム ASUB 2 → REPOS による継続

参照先

システム変数の詳細については、下記を参照してください。

『リストマニュアル』、「システム変数」

10.12 ユーザーアラーム用の ASUB 起動の実行

10.12.1 機能

説明

ASUP は、ユーザー、システム、またはイベント起動によって、さまざまな状況で起動できます。特定の状況では、「NC スタート禁止」アラーム応答のユーザーアラームで、ASUP が起動されなくなります。アラーム応答が未処置の場合は、リセットからのユーザー ASUP によってアラーム 16906 で確認されます。

チャネル別マシンデータを適切に設定すると、アラーム応答「NC スタート禁止」ユーザーアラームが有効な場合でも、ASUP を起動して実行できます。アラーム応答がオーバーライドされて ASUP が起動し、実行されます。

注記

「NC スタート禁止」のアラーム応答を含む NC アラームは、オーバーライドの影響を受けません。リセットからのユーザー ASUP は許可されず、アラーム 16906 とともに拒否されます。

オーバーライド可能なユーザーアラーム

オーバーライド可能なユーザーアラームの番号の範囲は、以下のように分類されます。

数値範囲	効果	解除
65000 - 65499	表示、NC スタート禁止	リセット
65500 - 65999	表示、NC スタート禁止(設定した MD20194 用の ASUP を除く)	リセット
66000 - 66999	表示、NC スタート禁止、事前にデコードされたブロックの実行後に移動が停止状態	リセット
67000 - 67999	ディスプレイ	キャンセル

数値範囲	効果	解除
68000 - 68999	表示、NC スタート禁止、補間器が即時停止	リセット
69000 - 69999	表示、NC スタート禁止、次のブロックの終点で停止	リセット

注記

以下のアラーム番号範囲については、「NC スタート禁止」アラーム応答をオーバーライド できますが、関連するアラームの他の応答によって停止条件が確定されます。停止条件は、この章で説明する機能ではオーバーライドできません。

- 66000 - 66999
- 68000 - 68999
- 69000 - 69999

10.12.2 起動

設定

各 ASUP チャネルを、以下のチャネル別マシンデータで個別に設定できます。

MD20194 $MC_IGNORE_NONCSTART_ASUP (特定のユーザーアラームのアラーム応答「NC スタート禁止」が未処置の場合でも ASUP 起動を許容)

MD 設定の変更は、NEWCONF パートプログラム命令による場合、またはソフトキーによるユーザーインターフェースからの場合にのみ機能します。

手順

ASUP 起動の通常のシーケンスは、以下のようになります。

- MD20194 $MC_IGNORE_NONCSTART_ASUP でマシンデータを適切に設定し、NEWCONF で起動します。
- パートプログラムを起動します。
  オーバーライド可能な番号範囲内のユーザーアラーム(アラーム 65500 など)が表示されます。これは、シンクロナイズドアクションから、またはパートプログラム命令によって発生します。

- アラームが表示されても、プログラム終了 M02、M30、または M17 までパートプログラムが処理されます。
  「リセット」チャネル状態が有効です。
- リセットから起動したユーザー ASUP が実行されます。

注記

実行中または停止したパートプログラムからの ASUP 起動も、「NC スタート禁止」アラーム応答が未処置の場合に許容されます。「NC スタート禁止」アラーム応答を含むユーザーアラームまたは NC アラームが関与するかどうかは関係ありません。

注記

ユーザー ASUP の処理中にパートプログラムが停止した場合、この ASUP は、たとえば、ASUP の M0 またはユーザー停止に対して、NC スタートキーで続行できなくなります。「NC スタート禁止」アラーム応答が、アラーム 16906 とともに拒否されます。以前に生成されたユーザーアラームは、リセットによってのみ確認応答が可能です。

10.12.3 例

10.12.3.1 リセットからのユーザー ASUB -例 1

MD20194 がアプリケーションで設定されていません。

メインプログラム

プログラムコード

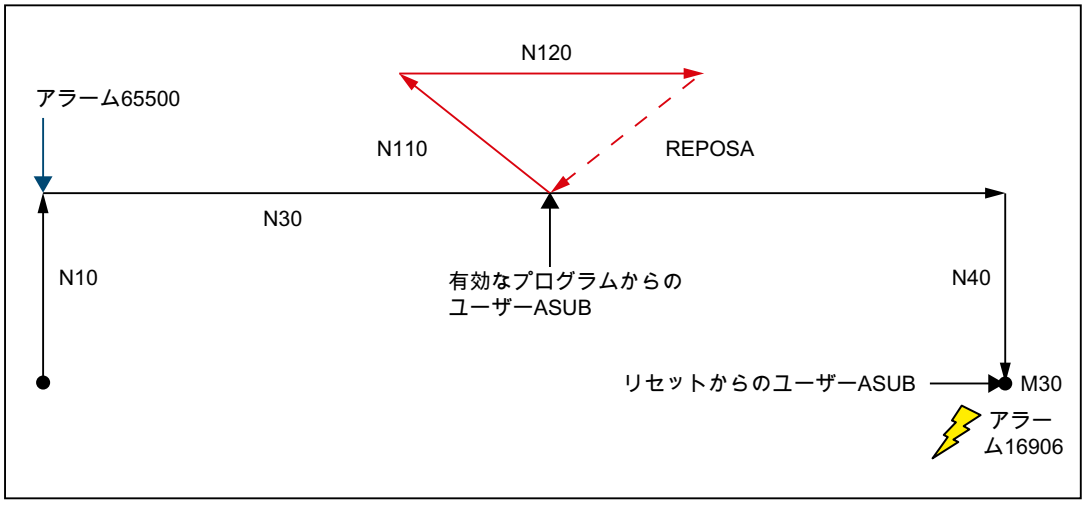
```
N10 G90 G0 Z10
N20 SETAL(65500)
N30 X100
N40 Z0
N50 M30
```

ユーザー ASUB

プログラムコード

```
N110 G91 G0 X-10 Z5
N120 X20
N130 REPOSA
```

手順



ブロック N10 が処理されます。「表示」と「NC スタート禁止」のアラーム応答を含むアラーム 65500 が表示されます。その結果、パートプログラムは停止しません。ブロック N30 がロードされ、処理されます。ユーザー ASUB をブロックの中間で使用した場合は、「NC スタート禁止」アラーム応答が未処置でも実行されます。REPOSA が、プログラムの中断で再開し、パートプログラムをプログラム終了 M30 まで処理します。ここでユーザー ASUB を使用すると、アラーム 16906 とともに拒否されます。NC 状態はリセットです。

10.12.3.2 リセットからのユーザー ASUB -例 2

アプリケーションが通常の場合であり、MD20194 が設定されています。

メインプログラム

プログラムコード

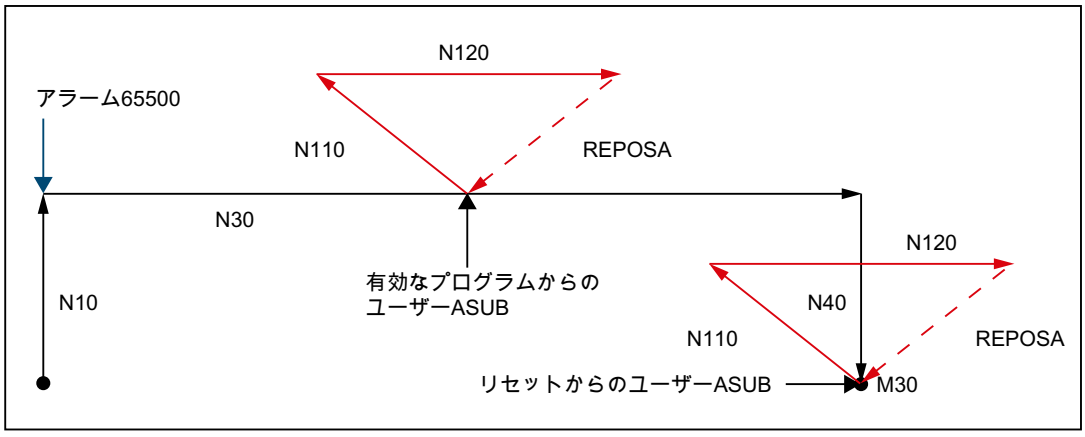
```
N4 $MC_IGNORE_NONCSTART_ASUP=1
N6 NEWCONF
N10 G90 G0 Z10
N20 SETAL(65500)
N30 X100
N40 Z0
N50 M30
```

ユーザー ASUB

プログラムコード

```
N110 G91 G0 X-10 Z5
N120 X20
N130 REPOSA
```

手順



マシンデータ MD20194 $MC_IGNORE_NONCSTART_ASUP が ASUB チャネル 1 に対して設定され、NEWCONF で起動されます。ブロック N10 が処理されます。「表示」と「NC スタート禁止」のアラーム応答を含むアラーム 65500 が表示されます。その結果、パートプログラムは停止しません。ブロック N30 がロードされ、処理されます。ユーザー ASUB をブロックの中間で使用した場合は、「NC スタート禁止」アラーム応答が未処置でも実行されます。REPOSA が、プログラムの中断で再開し、パートプログラムをプログラム終了 M30 まで処理します。ここでユーザー ASUB を使用した場合は、マシンデータ MD20194 が設定されているため、繰り返されます。

10.12.3.3 M0 を含むユーザー ASUB

MD20194 がアプリケーションで設定されています。

メインプログラム

プログラムコード

```
N4 $MC_IGNORE_NONCSTART_ASUP=1
N6 NEWCONF
N10 G90 G0 Z10
```

プログラムコード

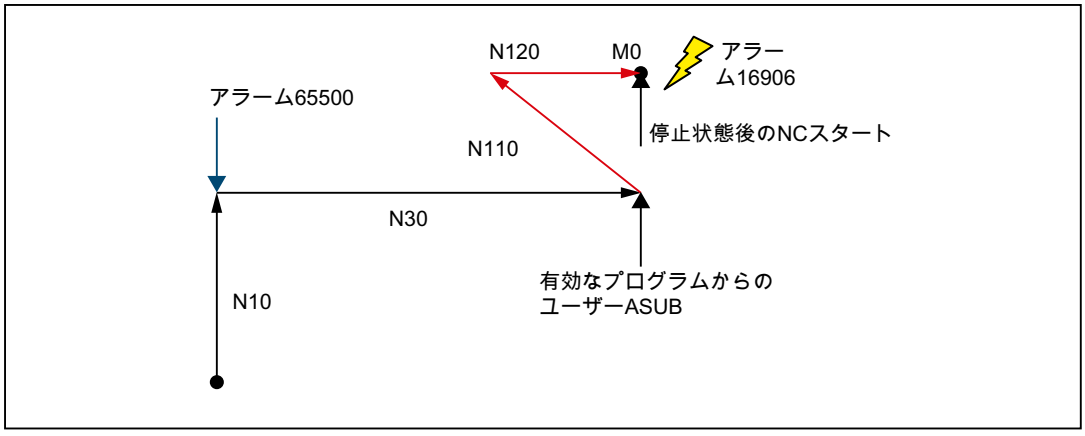
```
N20 SETAL(65500)
N30 X100
N40 Z0
N50 M30
```

ユーザー ASUB

プログラムコード

```
N110 G91 G0 X-10 Z5
N120 X10
N122 M0
N124 X10
N130 REPOSA
```

手順



マシンデータ MD20194 $MC_IGNORE_NONCSTART_ASUP が ASUB チャネル 1 に対して設定され、NEWCONF で起動されます。ブロック N10 が処理されます。「表示」と「NC スタート禁止」のアラーム応答を含むアラーム 65500 が表示されます。その結果、パートプログラムは停止しません。ブロック N30 がロードされ、処理されます。ユーザー ASUB をブロックの中間で使用した場合は、「NC スタート禁止」アラーム応答が未処置でも実行されます。M0 のために、パートプログラムがブロック N122 で停止します。NC スタートキーを押すと、ASUB が続行できなくなります。「NC スタート禁止」アラーム応答が、アラーム 16906 とともに拒否されます。以前に生成された 65500 アラームは、リセットによってのみ確認応答が可能です。

10.12.3.4 停止を含むユーザー ASUB

MD20194 がアプリケーションで設定されています。

メインプログラム

プログラムコード

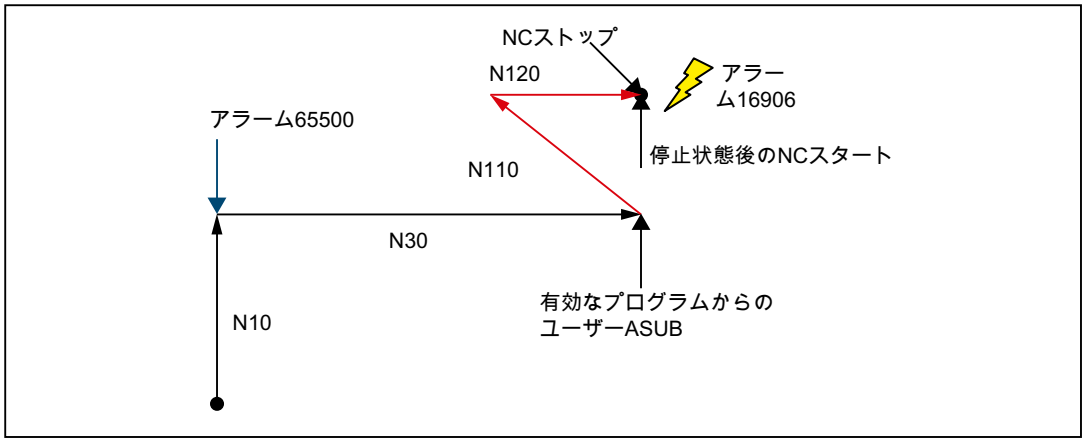
```
N4 $MC_IGNORE_NONCSTART_ASUP=1
N6 NEWCONF
N10 G90 G0 Z10
N20 SETAL(65500)
N30 X100
N40 Z0
N50 M30
```

ユーザー ASUB

プログラムコード

```
N110 G91 G0 X-10 Z5
N120 X20
N130 REPOSA
```

手順



マシンデータ MD20194 $MC_IGNORE_NONCSTART_ASUP が ASUB チャネル 1 に対して設定され、NEWCONF で起動されます。ブロック N10 が処理されます。「表示」と「NC スタート禁止」のアラーム応答を含むアラーム 65500 が表示されます。その結果、パートプログラムは停止しません。ブロック N30 がロードされ、処理されます。ユーザ

ー ASUB をブロックの中間で使用した場合は、「NC スタート禁止」アラーム応答が未処置でも実行されます。ブロックの中間で N120 のために NC ストップキーを押すと、ASUB が停止します。NC スタートキーを押すと、ASUB が続行できなくなります。「NC スタート禁止」アラーム応答が、アラーム 16906 とともに拒否されます。以前に生成された 65500 アラームは、リセットによってのみ確認応答が可能です。

10.12.3.5 停止からのユーザー ASUB

MD20194 がアプリケーションで設定されている場合と、設定されていない場合があります。

メインプログラム

プログラムコード

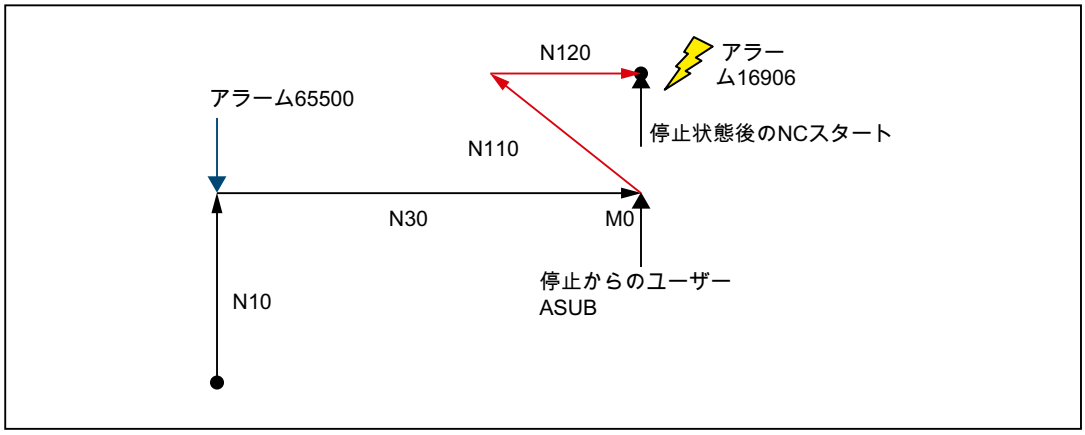
```
N10 G90 G0 Z10
N20 SETAL(65500)
N30 X50
N35 M0
N38 X100
N40 Z0
N50 M30
```

ユーザー ASUB

プログラムコード

```
N110 G91 G0 X-10 Z5
N120 X20
N130 REPOSA
```

手順



ブロック N10 が処理されます。「表示」と「NC スタート禁止」のアラーム応答を含むアラーム 65500 が表示されます。その結果、パートプログラムは停止しません。ブロック N30 がロードされ、処理されます。M0 のために、パートプログラムがブロック N35 で停止します。NC 状態は停止です。ここでユーザー ASUB を使用した場合は、NC 状態が停止、および「NC スタート禁止」アラーム応答が未処置でも実行されます。マシンデータ MD20194 $MC_IGNORE_NONCSTART_ASUP が設定されているかどうかは関係ありません。ブロック N120 の後は、REPOSA の前まで ASUB が停止状態のままです。再位置決めは、次の NC スタートでのみ可能です。ただし、「NC スタート禁止」アラーム応答のために、アラーム 16906 とともに NC スタートが拒否されます。以前に生成された 65500 アラームは、リセットによってのみ確認応答が可能です。

10.13 シングルブロック

「シングルブロック」機能では、3 つの異なる方法で NC プログラムをブロック単位で(ノンモーダルに)実行できます。

次のシングルブロックタイプが使用可能です。

- **「SBL1:汎用シングルブロック」**(各メインランブロック後に停止)
  各メインプログラムブロックが完全に実行された後、NC プログラムや処理が停止します。
- **「SBL2:計算ブロック」**(各デコード済みブロックの後で停止)
  各デコード済みブロックの後、NC プログラムや処理が停止します。
  複数のメインランブロックが、デコード済みブロックから生成される場合は、各メインランブロック後に停止します。

  注記

  ねじ切りブロック

  ねじ切りブロックでは、NC プログラムは停止せず、加工は停止しません。

- **「SBL3:精密シングルブロック」** (サイクル内でも、各デコード済みブロックの後で停止)
  ただし、SBL2 の場合と同様に、サイクル内でも各ブロックの後に NC プログラムと加工が停止します。

操作画面:シングルブロックタイプの選択

SBL1、SBL2、または SBL3 シングルブロックタイプを HMI ユーザーインタフェースから選択します。

操作エリアの[機械] **>** [プログラムへの影響] **>**メニュー:[プログラム制御]

10.13.1 パラメータ設定

インタフェース信号

起動

「シングルブロック」機能が、NC/PLC インターフェース信号を使用してチャネル別に起動します。

DB21、... DBX0.4 (シングルブロックの起動)

マシンデータ

シングルブロック加工の解除(MD10702、MD20106、MD20117)

- 次のマシンデータを使用すると、特定の加工状態およびプログラムタイプで、有効なシングルブロック機能にかかわらず、停止しないことを選択できます。
  MD10702 $MN_IGNORE_SINGLEBLOCK_MASK

注記

- ASUB またはサブプログラム内で SBLON/SBLOF をプログラミング (ページ 731)することによって、シングルブロック加工を明確に起動/解除できます。
- シングルブロックタイプ「SBL2:計算ブロック」では、このマシンデータは、属性 DISPLOF を含むシステム ASUB、ユーザー ASUB、およびサブプログラムに対してのみ機能します。

- シングルブロックに関するイベント起動プログラム呼び出し(プログラムイベント)の動作は、以下を使用して設定します。
  MD20106 $MC_PROG_EVENT_IGN_SINGLEBLOCK
- シングルブロックに関する割り込みプログラム(ASUB)の動作は、以下を使用して設定します。
  MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP
  シングルブロック中に ASUB が起動した場合は、ASUB の実行が完了します。シングルブロックは、ASUB が終了するまで、または、シングルブロックのマスクが有効でない最初のメインプログラムブロックまでは機能しません。ASUB から NC プログラムへの移行で軌跡速度が速すぎるために、 次のブロックで停止状態まで減速できない場合(有効な連続軌跡モード G64 など)は、以降の複数のブロックまで減速が延長される場合があります。

注記

ASUB で SBLON をプログラミング (ページ 731)することによって、この場合はシングルブロックの加工を再起動することは**できません**。

セッティングデータ

デバッグモード: シングルブロック「SBL2:計算ブロック」(SD42200)

ブロックの先行するデコードの結果として、ユーザーインタフェースでメインブロック関連の実ブロック表示と、表示変数値の間の参照が失われることがあります。これによって、特定の環境で妥当ではない変数値が表示されることがあります。

次のチャネル別セッティングデータにより、有効なシングルブロック「SBL2:計算ブロック」で、各ブロックに対して先読み停止を実行できます。これにより、パートプログ

ラムブロックの先読みがマスクされ、現在のブロック表示と変数値表示との関連性が維持されます。

SD42200 $SC_SINGLEBLOCK2_STOPRE (SBL2 のデバッグモードの起動)

注記

輪郭の偏差

移動ブロックを実行する場合、シングルブロックタイプ「**SBL2:計算ブロック**」(**デバッグモード**)で輪郭誤差が発生することがあります。

10.13.2 プログラミング

10.13.2.1 シングルブロック加工の無効化/有効化(SBLOF、SBLON)

NC プログラム全体のシングルブロック加工の無効化:

メインプログラムの第 1 行(PROC)でシングルブロック加工(SBLOF)の解除がプログラムされている場合、NC プログラムの終了時、または NC プログラムがキャンセルされるまで有効なままです。この場合、NC プログラムは、シングルブロックモード内では停止せずに実行されます。

サブプログラムの第 1 行(PROC)でシングルブロック加工(SBLOF)の解除がプログラムされている場合、NC プログラムの終了時、または NC プログラムがキャンセルされるまで有効なままです。プログラムされた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**

NC プログラム内でのシングルブロック加工の無効化

NC プログラム内のブロックでシングルブロック加工の解除(SBLOF)がプログラムされている場合、このブロック以降、シングルブロック加工の次にプログラムされた起動(SBLON)または有効なサブプログラムレベルの終了まで、シングルブロック加工が解除されます。

構文

```
SBLOF
SBLON
```

意味

SBLOF:	シングルブロック加工を解除するための事前定義された手順	
	単独ブロック指令:	はい、PROC ブロックで可能
	効果:	モーダル
SBLON:	シングルブロック加工を起動するための事前定義された手順	
	単独ブロック指令:	あり
	効果:	モーダル

10.13.2.2 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ブプログラムでマスクできます。DISPLOF が SBLOF と組み合わせてプログラムされている場合、サブプログラム内のシングルブロック停止について、サブプログラムの呼び出しが表示されます。

さまざまなシングルブロックタイプに関連する特別な問題

- 「SBL2:計算ブロック」および MD10702 $MN_IGNORE_SINGLEBLOCK_MASK、ビット 12==1:
  SBLON ブロック内で停止は行われません。
- 「SBL3:精密シングルブロック」: 命令 SBLOF がマスクされます

非同期サブプログラム(ASUB)のシングルブロックマスク

シングルブロック加工が有効な場合でも ASUB を停止なしで実行するには、ASUB の最初のプログラム行で PROC を SBLOF と組み合わせてプログラムする必要があります。

例

プログラムコード	コメント
N10 **PROC** ASUP1 **SBLOF** DISPLOF	
N20 IF $AC_ASUP=='H200'	
N30 RET	; モード変更では REPOS を実行しません。
N40 ELSE	
N50 REPOSA	; その他の場合はすべて REPOS を実行します。
N60 ENDIF	

10.13.2.3 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**	; シングルブロック起動
N70 M110	
N80 ...	

シングルブロックモードの起動時に、N20 ～ N60 がブロックのように処理されます。

例 2 :停止のないサブプログラム

メインプログラム

プログラムコード
...
N100 G1 X10 G90 F200
N120 X-4 Y6
N130 **CYCLE1**
N140 G1 X0
N150 M30

サブプログラム

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	

シングルブロック加工が有効な場合でも、このサイクルはすべて実行されます。

例 3 :シングルブロックマスクを含み、非表示の ASUB

シングルブロック加工が有効な場合でも、PLC ユーザープログラムによって起動した ASUB の処理を中断しないようにします。さらに、ASUB は表示しません。

プログラムコード

```
N100 PROC NV SBLOF DISPLOF          ; シングルブロックおよび表示マスク
N110    CASE $P_UIFRNUM OF
            0 GOTOF _G500
            1 GOTOF _G54
            2 GOTOF _G55
            3 GOTOF _G56
            4 GOTOF _G57
            DEFAULT GOTOF END
N120    _G54:G54 D=$P_TOOL T=$P_TOOLNO
N130    RET
N140    _G54:G55 D=$P_TOOL T=$P_TOOLNO
N150    RET
N160    _G56:G56 D=$P_TOOL T=$P_TOOLNO
N170    RET
N180    _G57:G57 D=$P_TOOL T=$P_TOOLNO
N190    RET
N200    END:D=$P_TOOL T=$P_TOOLNO
N210 RET
```

例 4 :サブプログラムでの個別の停止

前提条件:

- シングルブロック実行が有効です。
- MD10702 $MN_IGNORE_SINGLEBLOCK_MASK、ビット 12 = 1

メインプログラム

プログラムコード	コメント
N10 G0 X0	; シングルブロック停止
N20 X10	; シングルブロック停止
N30 **CYCLE**	; サイクルが生成した移動ブロックです。
N50 G90 X20	; シングルブロック停止
M30	

サブプログラム

プログラムコード	コメント
PROC CYCLE **SBLOF**	; シングルブロックマスク

```
プログラムコード                    コメント
N100    R0 = 1
N110    SBLON                       ; MD10702、ビット 12=1 によりシングル停止は発生しません
N120    X1                          ; シングルブロック停止
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

前提条件:シングルブロック実行が有効です。

```
プログラムコード                                    コメント
N10 X0 F1000                                        ; シングルブロック停止
N20 UP1(0)
      PROC UP1(INT _NR) SBLOF                       ; UP1 のシングルブロックの解除
      N100 X10
      N110 UP2(0)
            PROC UP2(INT _NR)
            N200 X20
            N210 SBLON                              ; シングルブロック起動
            N220 X22                                ; シングルブロック停止
            N230 UP3(0)
                  PROC UP3(INT _NR)
                  N300 SBLOF                        ; シングルブロックの停止
                  N305 X30
                  N310 SBLON                        ; シングルブロック起動
                  N320 X32                          ; シングルブロック停止
                  N330 SBLOF                        ; シングルブロックの停止
                  N340 X34
                  N350 M17                          ; シングルブロック停止(M17)
            N240 X24                                ; シングルブロック停止(N210)
            N250 M17                                ; シングルブロック停止(M17)
      N120 X12
      N130 M17                                      ; シングルブロック停止(M17)
N30 X0                                              ; シングルブロック停止
N40 M30                                             ; シングルブロック停止
```

10.13.3 モードグループ別シングルブロックタイプ A/B

モードグループ別シングルブロックでは、チャネル(制御チャネル)で、NC プログラムはシングルブロック内で実行されます。制御チャネルでは、シングルブロックは NC/PLC インタフェース信号(DB21 ... DBX0.4)を使用して起動する必要があります。

モードグループの他のチャネル(従属チャネル)では、特定の NC プログラムは、NC/PLC インタフェース信号(DB11 DBX1.6/7)を使用したモードグループに応じて選択したシングルブロックタイプ A または B に対応して、ブロック単位で実行されます。従属チャネルでは、NC/PLC インタフェース信号(DB21 ... DBX0.4)を設定することはできません。

シングルブロックタイプ A/B

- シングルブロックタイプ A:また、制御チャネルが停止すると、従属チャネルも即時に停止します(NC ストップに相当)。
- シングルブロックタイプ B:また、制御チャネルが停止すると、特定のブロック終点で従属チャネルも停止します(ブロック境界の NC ストップに相当)。

インタフェース信号

制御チャネル

- DB21、... DBX0.4 (シングルブロックの起動)

モードグループのすべてのチャネル

- DB21、... DBX7.1 (NC スタート)

モードグループ

- DB11 DBX1.6 (シングルブロック、タイプ B)
- DB11 DBX1.7 (シングルブロック、タイプ A)

シングルブロックタイプ A の実行手順

前提条件:すべてのモードグループチャネルが「リセット」状態または「中断」状態です。

1. PLC ユーザプログラム:制御チャネル、DB21 ... DBX0.4 = 1 でシングルブロックを選択します
2. PLC ユーザプログラム:モードグループ、DB11 DBX1.7 = 1 でシングルブロックタイプ A を選択します。
3. PLC ユーザプログラム:モードグループ起動のすべてのチャネル、DB21 ... DBX0.4 = 1
4. 制御チャネルはブロック終点で停止します。
5. すべての従属チャネルは、加工を即時に停止するための内部信号を受信します。
6. すべての従属チャネルが特定の減速段階の終点に到達すると、すべてのモードグループチャネルが「中断」状態になります。

シングルブロックタイプ B の実行手順

前提条件:すべてのモードグループチャネルが「リセット」状態または「中断」状態です。

1. PLC ユーザプログラム:制御チャネル、DB21 ... DBX0.4 = 1 でシングルブロックを選択します
2. PLC ユーザプログラム:モードグループ、DB11 DBX1.6 = 1 でシングルブロックタイプ B を選択します。
3. PLC ユーザプログラム:モードグループ起動のすべてのチャネル、DB21 ... DBX0.4 = 1
4. 制御チャネルはブロック終点で停止します。
5. すべての従属チャネルは、ブロック終点で加工を停止するための内部信号を受信します。
6. すべての従属チャネルが特定のブロック終点に到達すると、すべてのモードグループチャネルが「中断」状態になります。

10.13.4 必要条件

10.13.4.1 SBL2 シングルブロックタイプおよびブロック関連のシンクロナイズドアクション

シングルブロックタイプ「**SBL2:計算ブロック**」(ブロック単位のシンクロナイズドアクション)では、次のメインプログラムブロックの後でのみ、次の停止が実行されます。シンクロナイズドアクションと次のメインプログラムブロックの間にある初期ブロックでは、停止は実行されません。

10.13.4.2 プログラムされた停止(M0)、シングルブロック、およびシングルブロックタイプの切り替え

初期状態チャネル内で、NC プログラムはそこでプログラムされた **M0** によって停止し、またそのチャネル内で**シングルブロックは有効**です(DB21、... DBX0.4 == 1)。

この状態で、ユーザーインタフェースにより SBL1 または SBL3 と、SBL2 の間で、**シングルブロックタイプ**を**数回**切り替えた場合、アラーム 16922「最大ネストレベルの超過」が表示されます。

10.14 プログラム制御

オプション

1. 機能選択(操作画面または PLC を介した)
2. スキップレベルの起動
3. 補間バッファサイズの調整
4. 追加の基本ブロック表示を介したプログラム表示モード
5. 外部ソースからの実行(バッファサイズおよび数)
6. 外部サブプログラムからの実行

10.14.1 ユーザーインターフェースまたは PLC ユーザープログラムからの機能選択

手順

機能の選択は、SINUMERIK Operate ユーザーインターフェースの[自動|プログラム制御]操作エリアで、HMI/PLC インターフェースの対応する選択信号を設定することで行います。

FB1 パラメータ `MMCToIf` の値に応じて、HMI/PLC インターフェース信号の選択信号は、PLC 基本プログラムから NC/PLC インターフェースの対応する起動信号に送信されます:

- "TRUE":伝送
- "FALSE":伝送なし

初期設定では、パラメータ `MMCToIf` == "TRUE"の値。

ユーザー別の起動

NC/PLC インターフェースの対応する起動信号を設定することにより、PLC ユーザープログラムにより直接、ユーザー別に機能を起動できます。

FB1 パラメータ `MMCToIf` は、"FALSE"に設定する必要があります。それ以外の場合は、NC/PLC インターフェースは、HMI/PLC インターフェースの値により上書きされます。

予備充電の

起動時にいくつかの機能に対して、応答が返されます(以下の表を参照)。

NC/PLC インタフェース信号

表 10-2　プログラム制御:インタフェース信号

機能	選択(HMI → PLC)	起動(PLC → NC)	フィードバック(NC → PLC)
"ブロックスキップ" (SKP) / 0 - /7 "ブロックスキップ" (SKP) / 8 - /9	DB21、... DBX26.0 - 7 DB21、... DBX27.0 - 1	DB21、... DBX2.0 - 7 DB21、... DBX31.6 - 7	---
ドライラン送り速度(DRY)	DB21、... DBX24.6	DB21、... DBX0.6	DB21、... DBX318.6
"早送りオーバライド" (RG0)	DB21、... DBX25.3	DB21、... DBX6.6	---
シングルブロック(SBLx)	HMI のプログラム制御表示を介した SBL1、SBL2 または SBL3 の事前選択		
SBL1 "シングルブロックメインラン"	機械操作パネル(MCP) EB n + 5、ビット 2	DB21、... DBX0.4	---
SBL2 "デコードシングルブロック"			
SBL3 "サイクル内"			
プログラムストップ(M01)	DB21、... DBX24.5	DB21、... DBX0.5	DB21、... DBX32.5
関連するヘルプ機能(M-1)	DB21、... DBX24.4	DB21、... DBX30.5	DB21、... DBX318.5
ハンドルオフセット(DRF)	DB21、... DBX24.3	DB21、... DBX0.3	DB21、... DBX33.3
プログラムテスト(PRT)	DB21、... DBX25.7	DB21、... DBX1.7	DB21、... DBX33.7

参照先

- 『NC 変数とインタフェース信号』リストマニュアル
- 『操作マニュアル、HMI-Advanced 』、「 運転操作エリア」

10.14.2　スキップレベルの起動

機能

プログラムを実行する毎に実行しなくてもよいブロックをスキップすることができます。スキップされるブロックは、パートプログラム内でブロック番号の前に記号「/」を付けて示されます。

スキップレベルはパートプログラム内で「/0」～「/9」によって指定されます。

1 つのパートプログラムブロックで指定できるスキップレベルは 1 つのみです。

パラメータ設定

スキップレベルの数は、マシンデータを使用して定義します。

MD51029 $MM_MAX_SKP_LEVEL (NC プログラム内の最大スキップレベル数)

プログラミング

場合によってはプログラムのなかで実行しないブロック(プログラムテストブロックなど)は、以下の概略に従ってスキップできます。

プログラムコード	コメント
/N005	; ブロックスキップ、(DB21、... DBX2.0)第 1 スキップレベル
/0 N005	; ブロックスキップ、(DB21、... DBX2.0)第 1 スキップレベル
/1 N010	; ブロックスキップ、(DB21、... DBX2.1)第 2 スキップレベル
/2 N020	; ブロックスキップ、(DB21、... DBX2.2)第 3 スキップレベル
/3 N030	; ブロックスキップ、(DB21、... DBX2.3)第 4 スキップレベル
/4 N040	; ブロックスキップ、(DB21、... DBX2.4)第 5 スキップレベル
/5 N050	; ブロックスキップ、(DB21、... DBX2.5)第 6 スキップレベル
/6 N060	; ブロックスキップ、(DB21、... DBX2.6)第 7 スキップレベル
/7 N070	; ブロックスキップ、(DB21、... DBX2.7)第 8 スキップレベル
/8 N080	; ブロックスキップ、(DB21、... DBX31.6)第 9 スキップレベル
/9 N090	; ブロックスキップ、(DB21、... DBX31.7)第 10 スキップレベル

起動

10 個のスキップレベル「/0」～「/9」が、PLC 設定 PLC → NC インタフェース信号によって有効になります。

この機能は、[機械]操作エリアの[プログラム制御]メニューを介して、HMI から有効にされます。

- スキップレベル「/0」～「/7」の場合:
  インタフェース HMI → PLC DB21、... DBB26 (選択されたスキップブロック)
- スキップレベル「/8」～「/9」の場合:
  インタフェース HMI → PLC DB21、... DBX27.0 ～ DBX27.1

参照先:
『操作マニュアル』

注記
スキップレベルを変更できるのは、コントロールシステムが停止/リセット状態にある場合のみです。

10.14.3 補間バッファサイズの調整

MD28060

チャネル別の補間器によって、解析済みブロックがパートプログラムの運転中に補間バッファから実行されます。任意の時点で補間バッファ内に領域を必要とするブロックの最大数は、メモリ構成 MD28060 $MM_IPO_BUFFER_SIZE (IPO バッファ(DRAM)内の NC ブロックの数)によって定義されます。一部の用途では、解析から補間までの「間隔」を最小にするために、バッファの容量全体を使用しないことが有意義な場合があります。

SD42990; ＳＤ４２９９０

補間バッファ内のブロック数は、 セッティングデータ SD42990 $SC_MAX_BLOCKS_IN_IPOBUFFER (IPO バッファ内の最大ブロック数)により、2 ブロックを最小として、MD28060 $MC_MM_IPO_BUFFER_SIZE (IPO バッファ(DRAM)内の NC ブロック数)よりも小さい値に随時、制限されます。

セッティングデータ SD42990 $SC_MAX_BLOCKS_IN_IPOBUFFER の値:

値	効果
< 0	補間バッファ制限無効。 MD 28060 で設定された最大許容 IPO バッファ:MM_IPO_BUFFER_SIZE が有効になります。
or 1	2 ブロックの最小許容補間バッファが有効になります。
< MM_IPO_BUFFER_SIZE	指定された最大ブロック数を超えない IPO バッファが有効になります。
>= MM_IPO_BUFFER_SIZE	有効になる IPO バッファのブロック数を指定するのは、MD 28060MM_IPO_BUFFER_SIZE です。

注記

パートプログラムに SD42990 $SC_MAX_BLOCKS_IN_IPOBUFFER が設定されていると、SD のあるブロックがインタプリタによって解析されているときに、補間バッファの制限が即時有効になります。

つまり、IPO バッファの制限が、意図した制限より数ブロック早く有効になる可能性があります(MD 28070 $MC_MM_NUM_BLOCKS_IN_PREP も参照)。

早すぎる起動を回避し、IPO バッファの制限をブロックと同期して有効にするには、パートプログラムで SD を設定する前に STOPRE(先読み停止)をプログラムしてください。

適用

SD42990 $SC_MAX_BLOCK_IN_IPOBUFFER は、チャネル別のグローバルな有効性を持ち、パートプログラムでの変更も可能です。この修正値付きのパラメータは、プログラムの終了時に保持されます。定義されたイベントの発生時にこのセッティングデータを再度リセットするには、いわゆるイベント起動プログラムを作成してください。たとえば、このセッティングデータを RESET 時に常に事前定義値に設定できます。

用途

IPO バッファ制限は、パートプログラム内での現在の位置を他の目的のために読み取って処理する必要がある場合など、ブロック解析から補間までのブロック数を最小にする必要がある場合にいつでも使用できます。

例

```
N10 ...
N20 ...
..........
N100 $SC_MAX_BLOCKS_IN_IPOBUFFER = 5              ; IPO バッファの 5 NC ブロックへの制限
N110 ...
N120 ...
............
N200 $SC_MAX_BLOCKS_IN_IPOBUFFER = -1             ; IPO バッファ制限のキャンセル
N210 ...
............
```

10.14.4 追加の基本ブロック表示を介したプログラム表示モード

基本ブロック表示(ShopMill/ShopTurn 専用)

既存のブロック表示とともに第 2 のいわゆる基本ブロック表示を使用して、**機械上で動作する**すべてのブロックを表示できます。

基本ブロック表示の先読み

実際にアプローチした終点位置は、アブソリュート位置として表示されます。位置データは、ワーク座標系(WCS)または設定可能ゼロオフセットシステム(SZS)を基準とします。

表示バッファに格納される先読み表示ブロックの数は、関連する処理状態において、NC 先読みバッファ内にある、解析済みブロックの数に依存します。先読み停止が処理されると、表示ブロックの数がゼロに削減され、停止が応答された後に再度増加します。REORG イベント(モード変更、ASUP 起動など)の場合は、先読みのために格納された表示ブロックが削除され、後に再度先読みされます。

処理される値

基本ブロック表示で処理される値は、次の要素に一致します。

- 選択された工具
- 送り速度と主軸速度
- 実際にアプローチした位置の値
  例外:
  工具径補正が有効な場合は誤差が発生することがあります。
  モジュロ軸の場合、プログラム指令値が基本ブロック表示で表示されます。この値は、一回転の範囲外の場合もあります。

注記

一般的に、位置は WCS または SZS で表されます。
基本ブロック表示は、次のセッティングデータによって有効または無効にできます。
SD42750 $SC_ABSBLOCK_ENABLE

10.14.5 ShopMill/ShopTurn の基本ブロック表示

基本ブロック表示の設定

基本ブロック表示は、次のマシンデータを介して設定できます。

基本ブロック表示の NC マシンデータ	意味
MD28400 $MC_MM_ABSBLOCK	基本ブロック表示の起動
MD28402 $MC_MM_ABSBLOCK_BUFFER_CONF[2]	表示バッファのサイズ
マシンデータの表示	位置データの調整
MD9004 $MM_DISPLAY_RESOLUTION	メトリックサイズ表記
MD9011 $MM_DISPLAY_RESOLUTION_INCH	インチサイズ表記
MD9010 $MM_SPIND_DISPLAY_RESOLUTION	主軸最小表示単位に設定する座標系
MD9424 $MM_MA_COORDINATE_SYSTEM	WCS または SZS でのフィードバック値表示

この表示マシンデータは、NC マシンデータ
MD17200 $MN_GMMC_INFO_UNIT[0] (グローバル HMI 情報)から MD17200 $MN_GMMC_INFO_UNIT[3]にコピーされます。これにより、NC から、この表示マシンデータにアクセスできるようになります。

起動

基本ブロック表示は、電源オンによって、MD 28400 $MC_MM_ABSBLOCK(絶対値によるブロック表示の起動)で起動されます。MD28400 $MC_MM_ABSBLOCK が 1 に設定されていると、電源投入時にチャネル別の表示バッファ(FIFO)が作成されます。

表示バッファのサイズ(FIFO) = (MD28060 $MC_MM_IPO_BUFFER_SIZE (IPO バッファ内の NC ブロック数) + MD28070 $MC_MM_NUM_BLOCKS_IN_PREP (ブロック解析のためのブロック数)) に 128 バイトを積算。標準マシンデータ設定の場合、これは 6 KB のサイズを表します。

表示バッファのサイズの最適化:
128 と 512 の間の値を入力して、メモリ要件を最適化することができます。表示バッファで用意されている表示ブロックが、設定可能なアップロードバッファ経由で HMI に転送されます。

アップロードバッファの最大サイズは、(MD28402 $MC_MM_ABSBLOCK_BUFFER_CONF[0] + MD28402 $MC_MM_ABSBLOCK_BUFFER_CONF[1] + 1)に、MD28400 $MC_MM_ABSBLOCK で設定された
ブロック長を掛けて取得されます。

現在のブロックより**前**のブロックの数は、
MD28402 $MC_MM_ABSBLOCK_BUFFER_CONF[0]で設定され、現在のブロックより**後**のブロックの数は MD28402 $MC_MM_ABSBLOCK_BUFFER_CONF[1]で設定されます。

必要条件

MD28400 $MC_MM_ABSBLOCK で設定された表示ブロックの長さを超えると、適宜この表示ブロックが切り捨てられます。これを表すために、「...」という文字列がブロックエンドに付加されます。

先読みされたサイクル
(MD10700 $MN_PREPROCESSING_LEVEL > 1 (プログラム先読みレベル))の場合は、表示ブロックに軸位置**のみ**が含まれます。

基本ブロック表示の追加補足条件:

- 絶対値を持つモーダルシンクロナイズドアクションブロックは、
  考慮されません。
- 基本ブロック表示は、計算をともなうかどうかにかかわらず、ブロック検索中は無効です。
- 直交座標系では極座標プログラミングは表示されません。

半径/直径の値

基本ブロック表示および位置表示で示される直径値が、内部計算のための半径として必要となる場合があります。G グループ 29 に従った半径/直径単位の値は、次のオプションを使用して制御できます。

- G 命令 DIAMCYCOF(チャネル別の直径指定の拡張)この G 命令によって、
  サイクル実行時にチャネル別の直径指定が無効になります。この方法では、サイクルの計算を常に半径指定で実行できます。位置表示と基本ブロック表示は、DIAMCYCOF より前の直径指定の状態に従って継続されます。
  基本ブロック表示では、最後に表示された値が保持されます。
- G 命令 DIACYCOFA[AX](軸別の直径指定) この G 命令によって、
  サイクル実行時に軸別の直径指定が無効になります。この方法では、サイクルの計算を常に半径指定で実行できます。位置表示と基本ブロック表示では、DIAMCYCOF[AX] 表示より前の直径指定の状態に従って、これが継続されます。
  基本ブロック表示では、最後に表示された値が保持されます。
- MD27100 $MC_ABSBLOCK_FUNCTION_MASK (絶対値によるブロック表示のパラメータ設定)

Bit0 = 1　　基本ブロック表示では、径方向軸の指令値が常に直径値として表示されます。

有効なコンプレッサの動作

G グループ 30 が COMPOF ではない場合に、有効なコンプレッサに対して 2 つの表示ブロックが生成されます。SIMATIC

- 最初の表示ブロックには、有効なコンプレッサの G 命令が含まれます。
- 2 番目の表示ブロックには、欠落している表示ブロック用の文字として文字列「...」が含まれます。

例:

```
G0 X10 Y10 Z10          ; ブロックは、基本ブロック表示が可能な状態のままです
COMPCAD                 ; 最適な表面品質のためのコンプレッサ(CAD プログラム)
...                     ; 表示ブロックが欠落している印としての文字列
```

NC のパフォーマンスにおけるボトルネックを防止するため、基本ブロック表示は自動的に無効になります。表示ブロックが欠落している印として、文字列「...」の表示ブロ

ックが生成されます。
シングルブロックでは、すべての表示ブロックが常に生成されます。

10.14.6 DIN ブロックの構成

DIN ブロックの表示ブロックの構成

DIN ブロックの表示ブロックの基本構成

- ブロック番号/ラベル
- 最初の G グループの G 命令
  (最後の運転機能ブロックと比較して変更されている場合のみ)
- 軸位置
  (MD20070 $MC_AXCONF_MACHAX_USED(チャネルで有効な軸番号)に対応する順番)
- 追加のモーダル G 命令
  (最後の運転機能ブロックと比較して変更されている場合のみ)
- プログラム指令されたその他のアドレス

基本ブロック表示の表示ブロックは、次の規則に従って、プログラム指令されたパートプログラムブロックから直接入手されます。

- マクロは拡張されます。
- スキップ識別子とコメントは省略されます。
- ブロック番号とラベルは元のブロックから転送されますが、DISPLOF が有効な場合は省略されます。
- 小数点以下の桁数は、HMI を介して表示マシンデータ MD 9004、MD 9010 および MD 9011 で定義されます。

HMI 表示マシンデータ	NC マシンデータのアクセス
MD9004 $MM_DISPLAY_RESOLUTION	MD17200 $MN_GMMC_INFO_NO_UNIT[0]
MD9011 $MM_DISPLAY_RESOLUTION_INCH	MD17200 $MN_GMMC_INFO_NO_UNIT[1]

HMI 表示マシンデータ	NC マシンデータのアクセス
MD9010 $MM_SPIND_DISPLAY_RESOLUTION	MD17200 $MN_GMMC_INFO_NO_UNIT[2]
MD9424 $MM_MA_COORDINATE_SYSTEM	MD17200 $MN_GMMC_INFO_NO_UNIT[3]

- プログラム指令された軸位置は、MD9424 $MM_MA_COORDINATE_SYSTEM(フィードバック値表示の座標系)で指定された座標系(WCS / SZS)でのアブソリュート位置として表示されます。

注記

モジュロ軸についてはモジュロ補正が省略されます。つまり、一回転を超える位置が表示される可能性があります。したがって、基本ブロック表示は、値が常にモジュロ変換される位置表示とは異なることになります。

例

表示ブロック(元のブロック)と基本ブロック表示の比較:

- **プログラム指令位置**はアブソリュート値として表示されます。
  アドレス AP/RP は、そのプログラム指令値とともに表示されます。

元のブロック:	表示ブロック:
N10 G90 X10.123	N10 X10.123
N20 G91 X1	N20 X11.123

- **アドレスの割り当て** (非 DIN アドレス)は**<**アドレス**> = <**定数**>**の形式で表示されます。

元のブロック:	表示ブロック:
N110 R1 = -67.5 R2 = 7.5	
N130 Z = R1 RND = R2	N130 Z-67.5 RND = 7.5

- **アドレスインデックス**(アドレス拡張子)は、定数**<**アドレス**> [<**定数**>] = <**定数**>**として表示されます。

元のブロック:	表示ブロック:
N220 DEF AXIS AXIS_VAR = X	
N240 FA[AXIS_VAR] = R2	N240 FA[X] = 1000

- **アドレス拡張子のない DIN アドレス**は、
  <DIN アドレス**> <**定数**>**という形式で表示されます。

元のブロック：	表示ブロック：
N410 DEF REAL FEED = 1.5	
N420 F = FEED	N420 F1.5

以下は**H 機能**に適用されます。各プログラム指令値は、PLC への出力タイプにかかわりなく表示されます
(MD22110 $MC_AUXFU_H_TYPE_INT(H 補助機能の形式は整数))。

- **工具命令による工具選択**については、
  表示情報は T<値>または T=<文字列>の形式で生成されます。アドレス拡張子がプログラム指令されている場合、それも表示されます。
  複数の主軸が設定されているか、「マスタ工具ホルダによる工具交換」機能(MD20124 $MC_TOOL_MANAGEMENT_TOOLHOLDER(工具ホルダ番号))が有効な場合、T 番号は常にアドレス拡張子とともに出力されます。
  アドレス拡張子がプログラム指令されていない場合は、その代わりにメイン主軸またはマスタ工具ホルダの番号が使用されます(T<主軸番号/工具ホルダ>=)。
- S、M3、M4、M5、M19、M40 ～ M45 および M70(または MD 20094 $MC_SPIND_RIGID_TAPPING_M_NR(制御軸運転での切り替え用 M 機能))を介した**主軸指令**については、アドレス拡張子に関する次の規則が適用されます。
  アドレス拡張がプログラム指令されている場合、それも解決されます。
  複数の主軸が設定されている場合、アドレス拡張子も出力されます。
  アドレス拡張子がプログラム指令されていない場合、メイン主軸の番号が使用されます(S<主軸番号>=)。
- G[<グループ>] = <表現>という形式の**間接 G コードプログラミング**は、対応する G 命令に置き換わります。

元のブロック：	表示ブロック：
N510 R1=2	
N520 G[8]= R1	N520 G54

- 実行可能ブロックが生成されない**モーダル G コード**は、収集され、構文(DIN ブロック)によって許可されている場合は次の実行可能ブロックの表示ブロックとともに出力されます。これに該当しない場合(予約サブプログラム呼び出し TRANSMIT など)は、次の実行可能ブロックの前に、変更された G 命令を含む個別の表示ブロックが配置されます。

元のブロック：	表示ブロック：
N610 G64	G64
N620 TRANSMIT	N620 TRANSMIT

- アドレス **F** と **FA** が表示される**パートプログラム行**(MD22240 $MC_AUXFU_F_SYNC_TYPE = 3(F 機能の出力タイミング)を含む)については、一つの表示ブロックが常に生成されます。

元のブロック:	**表示ブロック:**
N630 F1000	N630 F1000
N640 X100	N640 X100

- **ブロック表示のために生成される表示ブロック**は、プログラム指令されたパートプログラムブロックから**直接**入手されます。輪郭の先読みの途中で中間ブロック(工具径補正 G41/G42、丸み付け/面取り RNDM、RND、CHF、CHR)が生成される場合、動作の基準となるパートプログラムブロックからの表示情報がそれらに割り当てられます。

元のブロック:	**表示ブロック:**
N710 Y157.5 G42	N710 Y157.5 G42
N720 Z-67.5 RND=7.5	N720 Z-67.5 RND=7.5

- **EXECTAB 命令**(輪郭要素の表の処理)では、EXECTAB によって生成されたブロックが表示ブロックに表示されます。

元のブロック:	**表示ブロック:**
N810 EXECTAB (KTAB[5])	N810 G01 X46.147 Z-25.38

- **EXECSTRING 命令**では、EXECSTRING を介して生成されたブロックが表示ブロックに表示されます。

元のブロック:
N910 DEF STRING[40] PROGSTRING = "N905 M3 S1000 G94 Z100 F1000 G55"
N920 EXECSTRING(PROGSTRING)

元のブロック:
N905 Z100 G55 G94 M3 S1000 F1000

10.14.7 外部からの実行

機能

「外部メモリからの実行」機能を使用して、メモリ不足により NC に直接保存できないプログラムを外部プログラムメモリから実行できます。

注記

保護されたサイクル(_CPF ファイル)は、外部プログラムメモリから実行することは**できません**。

外部プログラムメモリ

外部プログラムメモリは、以下のデータ記憶媒体にあります。

- 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 フラッシュメモリが原因の工具/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推奨します。サブプログラムの実行時に、接触異常、故障、トリガによる中止、または意図しないプラグ接続解除による USB フラッシュメモリへの通信割り込みが発生すると、直ちに加工が停止します。工具かワークまたは両方が損傷する恐れがあります。

用途

- **外部プログラムからの直接実行**
  基本的に、「外部からの実行」HMI モードで、インタフェースのディレクトリ構造からアクセス可能な任意のプログラムを選択および実行できます。
- **パートプログラムからの外部サブプログラムの実行**
  外部サブプログラムは、パートプログラム命令 EXTCALL を介して、呼び出しパス(オプション)とサブプログラム名称を指定して呼び出されます(→「外部サブプログラムの実行(EXTCALL) (ページ 753)」を参照)。

パラメータの割り付け

、「外部メモリからの実行」モードでプログラムを実行するために、ダイナミック NC メモリ内で再ロードメモリ(FIFO バッファ)を予約してください。

FIFO バッファのサイズ

FIFO バッファのサイズは、次のマシンデータで設定されます。

MD18360 $MN_MM_EXT_PROG_BUFFER_SIZE(外部からの処理用の FIFO バッファサイズ)

注記

ジャンプ命令を含むプログラム

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格納してください。

FIFO バッファの数

「外部メモリからの実行」モードで同時に実行される各プログラムに、それぞれ 1 つの FIFO バッファを準備してください。

FIFO バッファの数は、次のマシンデータで設定されます。

MD18362 $MN_MM_EXT_PROG_NUM(同時に実行可能な外部実行プログラムレベルの数)

リセット、電源投入時の動作

リセットおよび電源再投入によって外部プログラムの呼び出しが中止され、特定の FIFO バッファが消去されます。

同じプログラムメモリが使用可能なままで、EXTCALL 呼び出しの実行が MD9106 $MM_SERVER_EXTCALL_PROGRAMS で起動されている場合は、外部プログラムメモリから選択されたメインプログラムが、電源投入の後に自動的に再選択されます。

10.14.8 外部サブプログラムの実行(EXTCALL)

機能

複雑なワークを作成するための個別の加工ステップに、NC メモリに格納できないほど多量のメモリを必要とするプログラムシーケンスが含まれる場合があります。

この場合、EXTCALL パートプログラム命令を利用して、「外部メモリからの実行」モードでそのプログラムシーケンスをサブプログラムとして外部プログラムメモリから実行するというオプションがあります。

必要条件

外部サブプログラムからの実行には、次の必要条件が適用されます。

- サブプログラムを、操作画面のディレクトリ構造からアクセス可能にしてください。
- サブプログラムごとに、ダイナミック NC メモリ内で再ロードメモリ(FIFO バッファ)を予約してください。

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。

ロードバッファメモリのサイズは、以下で設定されます。

MD18360 MM_EXT_PROG_BUFFER_SIZE

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格納してください。

パラメータ設定

外部サブ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事前設定できます。

SD42700 $SC_EXT_PROG_PATH (EXTCALL 外部サブプログラム呼び出しのプログラムパス)

プログラミング時に指定されたサブプログラムパスまたは名称に加えて、呼び出すプログラムのパス全体が、ここから導出されます。

プログラミング

外部サブプログラムは、パートプログラム命令 EXTCALL によって呼び出されます。

構文: `EXTCALL("<パス/><プログラム名称>")`

パラメータ:

<パス>: 絶対パスまたは相対パスのデータ**(任意選択)**
タイプ: STRING

<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)**。
タイプ: STRING

注記

パスの指定: 省略名称

次のような省略名称でパスを指定できます。

- **LOCAL_DRIVE:** (ローカルドライブの場合)
- **CF_CARD:** (コンパクトフラッシュカードの場合)
- **USB:** (前面パネル接続の USB の場合)

CF_CARD: と **LOCAL_DRIVE:** は相互に代用可能です。

絶対パス名での EXTCALL 呼び出し

指定されたパスにサブプログラムが存在する場合、それが EXTCALL の呼び出しの後に実行されます。 存在しない場合は、プログラムの実行がキャンセルされます。

相対パス名付き/パス名なしでの EXTCALL 呼び出し

相対パス名付きまたはパス名なしで EXTCALL が呼び出された場合、使用可能なプログラムメモリが次のように検索されます。

- SD42700 $SC_EXT_PROG_PATH にパス名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生成されます。
  - SD42700 に設定されていたパス名
  - 分離記号としての「/」
  - EXTCALL でプログラムされたサブプログラムパスまたは名称
- 呼び出されたサブプログラムが事前設定パスで検出されない場合、次に、EXTCALL の呼び出しで指定されたデータがユーザメモリディレクトリ内で検索されます。
- サブプログラムが最初に検出されたとき、検索が終了します。 検索で該当するものがない場合、そのプログラムはキャンセルされます。

例

ローカルドライブからの実行

メインプログラム

プログラムコード

```
N010 PROC MAIN
N020 ...
N030 EXTCALL ("ROUGHING")
N040 ...
N050 M30
```

外部サブプログラム:

プログラムコード

```
N010 PROC ROUGHING
N020 G1 F1000
N030 X= ... Y= ... Z= ...
N040 ...
...
...
```

プログラムコード
`N999999 M17`

「_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。

その後にロードする「SCHRUPPEN.SPF」サブプログラムまたは「SCHRUPPEN.MPF」サブプログラムは、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。

サブプログラムパスは SD42700 で設定されています。

SD42700 $SC_EXT_PROG_PATH = "LOCAL_DRIVE:WKS.DIR/WST1.WPD"

注記

SD42700 にパスを指定していない場合は、この例のための `EXTCALL` 操作を次のようにプログラムしてください。

```
EXTCALL("LOCAL_DRIVE:WKS.DIR/WST1.WPD/SCHRUPPEN")
```

10.15 外部記憶からの実行(EES) (オプション)

10.15.1 機能

注記

この機能を使用するには、「拡張 CNC ユーザー記憶」または「外部記憶からの実行(EES)」ライセンスオプションが必要です。

機能

EES (外部記憶からの実行)機能を使用すると、NCK でプログラムが**外部プログラム記憶から直接**、実行されるようにすることができます。

外部メモリとしては、以下のドライブ装置が使用できます。

ドライブ	シンボル名称 1)	SINUMERIK 840D sl	SINUMERIK 828D
NC Extend (以前の"ローカルドライブ")	CF_CARD LOCAL_DRIVE SYS_DRIVE	ユーザーメモリ または PCU のローカルハードディスク	PPU のシステム CF カード
ユーザー CF	CF_CARD LOCAL_DRIVE	-	PPU のユーザー CF カード
ネットワークドライブ			
静的に管理された USB ドライブ			

1) SINUMERIK 840D sl では、シンボル名 LOCAL_DRIVE、CF_CARD および SYS_DRIVE がドライブ NC Extend に永続的に割り当てられます(⇒ NC Extend は、LOCAL_DRIVE、CF_CARD、および SYS_DRIVE によってアドレス指定できます)。
SINUMERIK 828D では、NC Extend に対するシンボル名 LOCAL_DRIVE、CF_CARD、および SYS_DRIVE の割り当てを設定できます。この方法では、たとえば USER CF ドライブが存在しない場合に、PPU のシステム CF カードもシンボル名 LOCAL_DRIVE および CF_CARD によってアドレス指定できます。

通知
USB フラッシュメモリが原因の工具/ワークの損傷 USB フラッシュメモリは、外部プログラムの実行時には推奨**できません**。プログラムの実行時に、接触異常、故障、トリガによる中止、または意図しないプラグ接続解除による USB フラッシュメモリへの通信割り込みが発生すると、制御不能な加工の停止が発生します。工具かワークまたは両方が損傷する恐れがあります。

必要条件

EES の使用時には、次の必要条件が該当します。

- 「拡張 CNC ユーザー記憶」または、「外部記憶からの実行(EES)」ライセンスオプションを設定する必要があります。
- 外部メモリとして制御で使用されるドライブは、論理ドライブとして設定する必要があります(「セットアップ (ページ 760)」を参照)。

運転モード

使用可能なオプションおよびドライブ設定に応じて、さまざまな EES 運転モードを使用可能です。有効な制御運転モードは、マシンデータ MD18045 $MN_EES_MODE_INFO を使用して表示できます。

MD18045	運転モード	オプション	外部メモリ	
			SINUMERIK 840D sl	SINUMERIK 828D
= 0	EES が無効	-	-	
= 1	ローカル EES が有効	6FC5800-0AP77-0YB0 CNC ユーザーメモリが拡張済み	NCU 上での EES の使用は、CF カードの拡張ユーザーメモリ (100 MB)に限定されています。 EES が PCU 上で使用されている場合、全空きメモリを NC Extend で使用できます。	EES の使用は、システム CF カードの拡張ユーザー記憶(100 MB)に限定されます。
			ネットワーク経由および USB 経由のいずれでも、EES を使用できません。	
		6FC5800-0AP12-0YB0 NCU の CF カード上に追加の HMI ユーザーメモリ	EES が NCU 上で使用されている場合、追加オプションにより、(MD9111 に応じて)ローカルユーザーメモリは 6 GB まで拡張できます。 PCU 上で EES を使用する場合、追加オプションは必要ありません。	-

MD18045	運転モード	オプション	外部メモリ	
			SINUMERIK 840D sl	SINUMERIK 828D
= 2	グローバル EES が有効	6FC5800-0AP75-0YB0 外部記憶からの実行 (EES)	EES は、使用可能なすべての外部記憶に使用できます。	EES は、使用可能なすべての外部記憶に使用できます。 **注:** ネットワークによる EES ではさらに、HMI ファンクション[ネットワークドライブ管理機能]を有効にする必要があります(MD19730 ビット 2 = 1)。
		6FC5800-0AP12-0YB0 NCU の CF カード上に追加の HMI ユーザーメモリ	EES が NCU 上で使用されている場合、追加オプションにより、(MD9111 に応じて)ローカルユーザーメモリは 6 GB まで拡張できます。 PCU 上で EES を使用する場合、追加オプションは必要ありません。	-
= 5	ローカル EES が有効 + グローバルパートプログラムメモリ	MD18045 = 1 と同様ですが、拡張ユーザーメモリでは、グローバルパートプログラムメモリ (ページ 762)が設定されます。		
= 6	グローバル EES が有効 + グローバルパートプログラムメモリ	MD18045 = 2 と同様ですが、拡張ユーザーメモリでは、グローバルパートプログラムメモリ (ページ 762)が設定されます。		

特性

EES 機能は、「外部からの実行 (ページ 751)」および「外部サブプログラムの実行 (EXTCALL) (ページ 753)」機能の代わりに使用できます。

EES 機能には、以下の利点があります。

- システム全体での標準プログラム処理
- 使用可能な命令に関する制限なし
  後方ジャンプや、再ロードメモリのサイズによるジャンプ命令のジャンプ距離など、「外部メモリからの実行」および「外部サブプログラムからの実行(EXTCALL)」の制限は、EES により排除されます。
- プログラムは、さまざまなプログラム記憶(NC、GDIR、外部ドライブ)の間を、大幅に容易に移動できます。
- パートプログラムのサイズとプログラム数は、実質的に制限はありません(外部データメモリの容量によってのみ制限されます)。
- サブプログラムの格納場所に影響されない、サブプログラム呼び出しの均一な構文 (EXTCALL 呼び出しが不要)。
- ネットワークドライブを、1 台以上のステーション(PCU/NCU)と併用できます。前提条件は、これらのステーションでドライブ設定が均一であることです。その結果、すべてのステーションでプログラムが均一に表示されます。
- すべてのステーションで外部プログラム記憶が均一に表示されるため、格納されているプログラムへの変更が、すべてのステーションに一貫して適用されます。

10.15.2 セットアップ

10.15.2.1 ドライブの設定

EES 機能を使用するには、コントロールシステムで使用するドライブを宣言してください。

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
- ユニバーサル/旋削加工/フライス加工操作マニュアル

新しいドライブ設定を有効にした後は、使用可能な各ドライブにプログラムを自由に配布できます。

注記

以前のドライブの一部は、新しく作成したドライブ設定では使用できなくなっている場合があります。このようなドライブ上のプログラムにはアクセスできなくなっています。

解決策：まず、これらのドライブに格納されたプログラムを、まだアクセス可能なドライブにコピーします。

注記

保護されたサイクル(_CPF ファイル)は、システム上の理由から、NC パートプログラムメモリからのみ実行できるため、外部プログラムメモリに格納して実行することは**できません**。

MPF.DIR、SPF.DIR、および WKS.DIR フォルダを含む以前の NC パートプログラムメモリは、EES の使用時に必ずしも必要ではありません。システムは、NC パートプログラムメモリを使用しないで設定することもできます。

通知

表示されないプログラムの実行

NC パートプログラムメモリは、ドライブ設定から加工されていても、システム自体では常に使用可能です。これは、特に、プログラムの実行時に、まだ存在しているプログラムが SPF.DIR から不用意に実行される可能性があることを意味します。

解決策：ドライブ設定に NC パートプログラムメモリを含めずにシステムが設定されている場合は、格納されているプログラムを手動で削除してください。

NC パートプログラムメモリを引き続き使用する場合は、システムから完全には削除せずに、必要に応じて、適切な保護レベルを割り当ててください。

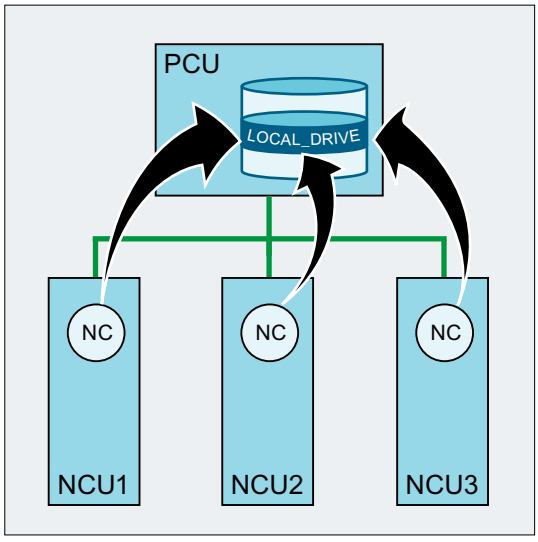
ドライブは、1 台以上のステーション(PCU/NCU)と共有できます。これらのステーションの標準ドライブ設定は、特定のステーションに依存しない標準プログラム表示を意味します。

注記

NCU/PPU の CF カードは、複数のステーションで使用することは**できません**。

例:

複数の NCU が PCU (LOCAL_DRIVE)のローカルハードディスク上でプログラムメモリを共用します。



注記

EES モードで別のステーションで外部プログラムメモリも使用する場合は、次の規則を遵守する必要があります。

- プログラムは、複数のステーションから同時に編集することは**できません**。
- 実行中のプログラムは、変更**できなくなります**。

SINUMERIK 840D sl のみ

外部 HMI で動作する場合は、ドライブを外部 HMI 上で設定してください。ドライブ設定(logdrive.ini)は、外部 HMI から、対応する NCU にロードしてください。ドライブ設定の対話画面では、ソフトキーを使用して転送することができます。

複数の NC が一緒に機能するシステムでは、すべての NC についてドライブ設定を同じにしてください。このためには、mmc.ini ファイルに記載されたすべての NCU に logdrive.ini ファイルを配布します。したがって、配布先の既存の設定は上書きされます。

10.15.2.2 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』、「プログラムの管理」>「ドライブのセットアップ」の章

GDIR として機能するドライブ上には、MPF.DIR、SPF.DIR、WKS.DIR の各ディレクトリが自動的に作成されます。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、EES 動作のための GDIR をセットアップしなければなりません。

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。したがって、パスの指定を含む NC プログラムを、パッシブファイルシステムから GDIR へと互換性を保って再配置することができます。

GDIR は、絶対パスを指定せずに呼び出されたサブプログラムの検索パスを拡張します。

NC パートプログラムメモリに代わる GDIR

NC パートプログラムメモリで、MPF.DIR、SPF.DIR、および WKS.DIR フォルダが完全に空(から)の場合は、GDIR が NC パートプログラムメモリに代わって機能します。以前の NC 検索パスが、GDIR によって 1:1 でエミュレートされます。

外部アーカイブ/データ記憶媒体でのメインプログラムの選択

サブプログラムの検索シーケンス:

1. 外部アーカイブ/データ記憶媒体の実際のフォルダ
2. GDIR メモリの SPF.DIR
3. CALLPATH を使用して参照されるドライブ
4. サイクル

NC パートプログラムメモリを拡張する GDIR

NC パートプログラムメモリで、MPF.DIR、SPF.DIR、および WKS.DIR フォルダが空(から)でない場合、サブプログラムの検索シーケンスは、メインプログラムがアーカイブされる場所(有効なフォルダ)に応じて異なります。

NC パートプログラムメモリ(MPF.DIR、または WKS.DIR の xxx.WPD)でのメインプログラムの選択

サブプログラムの検索シーケンス:

1. NCK パートプログラムメモリ内の実際のフォルダ
2. NCK パートプログラムメモリ内の SPF.DIR
3. CALLPATH を使用して参照されるドライブ
4. サイクル

EES 用に解放された外部アーカイブ/データ記憶媒体でのメインプログラムの選択

サブプログラムの検索シーケンス:

1. 外部アーカイブ/データ記憶媒体の実際のフォルダ
2. NCK パートプログラムメモリ内の SPF.DIR
3. GDIR の SPF.DIR
4. CALLPATH を使用して参照されるドライブ
5. サイクル

注記

検索シーケンスを定義するには、MD11625 $MN_FILE_ONLY_WITH_EXTENSION および MD11626 $MN_CYCLES_ONLY_IN_CYCDIR も参照してください。

注記

外部ドライブは、CALLPATH 命令を使用して参照することもできます。

10.15.2.3 EES のパートプログラムでのファイル処理の設定

システム全体で一義的なプログラム名

外部プログラムメモリを、EES モードのさまざまなステーションで一緒に使用する場合は、さまざまなステーションでファイル操作(WRITE、DELETE、...)を同時に実行すると、アクセスの競合が発生します。このようなアクセスの競合を回避するには、各ステーションで、システム全体で一義的な名前空間をセットアップすることを推奨します。

システム全体で一義的な名前空間

システム全体で一義的なファイル名の名前空間を実現するには、たとえば、NC の EES 別の名前(マシンデータで設定した)と、プログラムが実行されるチャネルのチャネル番号にファイル名を関連付けます。たとえば、プログラムを実行すると、以下のコードによりファイル名(MYFILE _NC1_1.SPF)が生成されます。この名前は、NC (NC1)の EES 別の名前とチャネル番号(チャネル 1)がプログラム名に付加された、システム全体で一義的な名前です。

```
$MN_EES_NC_NAME="NC1"
N10 DEF STRING[31] FILENAME
N20 FILENAME="MYFILE_" << $MN_EES_NC_NAME << "_" << $P_CHANNO <<
".SPF"
```

パラメータ設定

NC の EES 別の名前は、以下の NC 別マシンデータで設定します。

MD10125 $MN_EES_NC_NAME = <NC 名>

注記

システム全体で一義的な NC 名

アクセスの競合を回避するために、NC の EES 別の名前は、システム全体で一義的な名前にしてください。これに対する責任は、ユーザー/工作機械メーカにあります。

プログラムを呼び出すと、ファイル ID を持つファイルのみを検索

EES 動作時のサブプログラム呼び出しのプログラム検索を高速化するには、検索を、ファイル ID (SPF、MPF など)を**持つ**ファイルのみに制限することを推奨します。

MD11625 $MN_FILE_ONLY_WITH_EXTENSION = 1

注記

MD11625 は、外部サブプログラムを EXTCALL で処理する場合はプログラム検索に対して無効です。

参照先:
サブプログラム呼び出しの検索パスの概要については、『プログラミングマニュアル、上級編』を参照してください。

サイクルフォルダのインターフェースを備えたプログラムのみの検索

EES 動作時のサブプログラム呼び出しのプログラム検索を高速化するには、インターフェースの説明を(PROC 命令で)作成しており、インターフェースの説明がサイクルフォルダ(CUS、CMA、CST)の 1 つから生成されているサブプログラムの検索を、サイクルフォルダに制限することを推奨します。

MD11626 $MN_CYCLES_ONLY_IN_CYCDIR = 1

注記

MD11626 は、インターフェースが EXTERNAL 宣言を使用して作成されているサブプログラムには無効です。検索は、すべてのプログラムフォルダで実行されます。

通知
サイクルフォルダ以外ではサイクルの検索が失敗 現在のフォルダおよびグローバルサブプログラムフォルダのサイクルは、設定 MD11626 = 1 では見つからなくなっています。 解決策：サイクルを常にサイクルフォルダに格納します。

10.15.2.4 メモリ構成

パッシブファイルシステムでのエンドユーザープログラムメモリの削減

EES を有効にすると、パッシブファイルシステムでのエンドユーザープログラムメモリを削減できます。

MD18352 $MN_MM_U_FILE_MEM_SIZE (パートプログラム/サイクル/ファイルのエンドユーザーメモリ)

解放されたメモリは、工具データまたはメーカーサイクルなどに使用できます(MD18353 $MN_MM_M_FILE_MEM_SIZE)。

参照先:
メモリ設定の詳細については、『機能マニュアル、上級機能』を参照してください。

再ロードメモリの解放

EES 機能は、「外部メモリからの実行」および「外部サブプログラムの実行(EXTCALL)」機能の代わりに使用できます。

「外部サブプログラムの実行(EXTCALL)」機能ではなく、EES による EXTCALL 呼び出しでパートプログラムからサブプログラムを実行するには、EXTCALL 呼び出しを CALL 呼び出しに変更し、 必要に応じて、パスの指定を調整してください。

切り替えが完了した後は、「外部メモリからの実行」および「外部サブプログラムの実行(EXTCALL)」に必要な再ロードメモリ(FIFO バッファ)を解放できます。

MD18362 $MN_MM_EXT_PROG_NUM(同時に実行可能な外部実行プログラムレベルの数) = 0

10.15.3 必要条件

TEACH IN

EES モードでは、AUTOMATIC 運転モードで[ティーチング]ファンクションを使用することは**できません**。

10.16 Process Datashare - 外部の機器/ファイルへの出力

10.16.1 機能

「Process DataShare」機能を使用して、パートプログラムから外部機器/外部ファイルにデータを書き込むことができます。たとえば、生産データをログしたり、コントロールシステムの追加機器を制御することができます。

適用

この機能は以下のように使用できます。

- 実際の NC でのみ使用できます(SNC および VNC シミュレーションソフトウェアでは**使用不可**)。
- パートプログラムでのみ使用できます(シンクロナイズドアクションでは**使用不可**)。
- 使用可能な(設定済みの)すべての出力機器に対して、NC のすべての加工チャネルで並列に使用できます。

外部機器/ファイル

以下の外部機器/ファイルを使用できます。

- ローカル CF カード上のファイル
  ローカル CF カードは、シンボル識別子 LOCAL_DRIVE を使用して HMI から参照されるメモリです。SINUMERIK 840D sl の場合はローカルドライブ、SINUMERIK 828D の場合はユーザー CF カードとなります。

  注記

  SINUMERIK 840D sl の場合は、LOCAL_DRIVE 機器への出力に、「NCU の CF カードに xxx MB HMI ユーザーメモリの追加」オプションが必要です。SINUMERIK 828D の場合は、ユーザー CF カードを使用してください。オプションは不要です。

- ネットワークドライブ上のファイル
- V.24 インタフェース

  注記

  SINUMERIK 840D sl の場合は、V.24 インタフェースへの出力用に、NCU オプションモジュール RS232 インタフェースが必要です。SINUMERIK 828D 出力は内蔵の V.24 インタフェースで実現されます(必要条件:MD51233 $MNS_ENABLE_GSM_MODEM = 0).

開かれている外部機器の最大数

パートプログラム/チャネルに、複数の外部機器/ファイルを割り当てることもできます。すべての NC チャネル全体で、最大で 10 個の出力機器を同時に開くことができます。また、Siemens サイクル用に 2 つのエントリが予約されています。

最大 5 つのジョブを、出力機器で同時に有効にすることができます。

使用モード

出力機器を開くときに、それぞれの機器毎に、機器を 1 つのチャネルだけで使用するか複数のチャネルがその機器に出力できるか(「共有」モード)を指定することができます。

パートプログラム終了/チャネルリセットの動作

パートプログラムの終了とチャネルのリセットにより、チャネルで開かれていたすべての外部機器/ファイルが閉じられます。

制御装置へのデータ転送機能の使用

通知
データのセキュリティ Process Datashare 機能を使用して、データを外部機器から Ethernet-X130 インターフェース経由で制御装置に送信した場合は、制御装置上のデータが第三者によって改ざんされ、一貫性がなくなる可能性があります。この機能の使用時には、ネットワークを必ず、第三者のアクセスから保護してください。

10.16.2 セットアップ

使用する外部機器は、/oem/sinumerik/nck/extdev.ini または/user/sinumerik/nck/extdev.ini ファイルで設定されます。2 つのファイルを使用できる場合は、ユーザーエリアにあるファイルが優先されます。ファイルは、SYSTEMDATEN/CF カードの操作エリア COMMISSIONING で更新できます。

注記

LOCAL_DRIVE と CYC_DRIVE の使用時は、extdev.ini ファイルでの設定は必要はありません。この 2 つの機器は、対応するオプションが設定されているかユーザーコンパクトフラッシュカードが有効な場合は常に使用できます。

使用する外部機器は、extdev.ini ファイルの[ExternalDevices]セクションで定義/記載されます。シリアル機器(/dev/v24)と最高で 9 つまでのファイルまたはディレクトリ(/dev/ext/1…9)を機器として指定することができます。機器を指定する場合は、Linux 表記が使用されます。「;」で始まる行はコメントであり、オーバーリードされます。

/dev/v24 を除いて、機器はディレクトリパスとして(「/」を付けて終了する)またはファイルパスとして(つまり、完全修飾パスを付けてファイル名で終了する(「/」で終了しない))宣言できます。パートプログラムで使用する場合は、ファイル名(パス)もディレクトリパス付きの機器で指定してください。

/dev/v24 を除いて、機器は「サーバー」、「パス」、任意選択の「書き込みモード」の 3 つの項目をコンマで区切って定義されます。

ファイルまたはディレクトリに対して(これは、ディレクトリ内のすべてのファイルに適用します)、ファイルを開いた後に上書きするのか(「O」 = 上書き)、出力をファイルに付加するのか(「A」 = 付加)を指定することができます。初期値は「A」です。存在しないファイル/ディレクトリは、開いたときに新規に作成されます。

V.24 インタフェース機器の場合は、伝送速度、データビット、ストップビット、パリティ、プロトコル、および考えられる終了の設定だけがこのシーケンスで指定されます。

LOCAL_DRIVE に生成/保存されたファイルの場合、LOCAL_DRIVE_MAX_FILESIZE データを使用して、最大ファイルサイズをバイト単位で設定することができます - これは、すべてのファイルに対して標準として有効です。ファイルサイズは、付加モードで EXTOPEN 命令を実行するときにチェックされます。選択的に、書き込みモード(「O」 = 上書き、「A」 = 付加)を LOCAL_DRIVE_FILE_MODE データを使用して定義することができます。初期値は「A」です。

注記

extdev.ini 設定ファイルのコピーテンプレートは、/siemens/sinumerik/nck フォルダにあります。

注記

extdev.ini ファイルへの変更は、NC 再起動/起動後にしか有効になりません。

注記

USB 機器

SINUMERIK 828D の場合、「usb」 (パーティションデータなし)を前面パネルに差し込まれた USB 機器のターゲットとして定義することもできます。USB 機器は、シンボルの機器識別子"/dev/ext/x"を使用した、パートプログラムからの直接のアドレス指定しかおこなえません。

SINUMERIK 840D sl の場合、TCU の静的に接続されている USB インタフェースのみが USB 機器として設定できます。設定は、上記のように「サーバー」の指定として SERVER:/PATH タイプを使用しておこなわれます。SERVER は TCU 名であり、/PATH は USB インタフェースを指定しています。TCU の USB インタフェースは「dev0-0」、「dev0-1」、「dev1-0」を使用してアドレス指定されます。パスデータは常に「/Partition」で始まります。パーティションは 2 桁のパーティション番号またはパーティション名称で指定できます – 必要に応じて、以下のように必要な対象までファイルパスで拡張されます。

/dev/ext/8 = "TCU4:/dev0-0, /01/, A"

/dev/ext/8 = "TCU4:/dev0-0, /01/mydir.dir/"

/dev/ext/8 = "TCU4:/dev0-0, /myfirstpartition/Mydir.dir/myfile.txt, O"

例

[ExternalDevices]

; コメント行

; V24 の例

; /dev/v24 = "9600, 8, 1, none, rts [, etx]"

; ネットワークドライブの例

; /dev/ext/1 = "//[USERNAME[/DOMAIN][%PASSWORD]@]SERVER/SHARE/, /, A"

; /dev/ext/2 = "//[USERNAME[/DOMAIN][%PASSWORD]@]SERVER/SHARE, /myfile.txt, O"

; /dev/ext/3 = "//[USERNAME[/DOMAIN][%PASSWORD]@]SERVER/SHARE, /mydir/, A"

; /dev/ext/4 = "SERVER:/dev0-0, /01/, A"

; …

; SINUMERIK 828 のみ(USB)

; /dev/ext/9 = "usb, / [, O]"

; 初期設定:パーティション番号 = 1

; SIEMENS のみ

; /dev/cyc/1= "//[USERNAME[/DOMAIN][%PASSWORD]@]SERVER/SHARE, /mydir/, A"

; /dev/cyc/2= "//[USERNAME[/DOMAIN][%PASSWORD]@]SERVER/SHARE/mydir, /, A"

LOCAL_DRIVE_MAX_FILESIZE = 50000

LOCAL_DRIVE_FILE_MODE = "O"

EXTOPEN パラメータ<書き込みモード>の有効性

extdev.ini ファイルでの設定ならびに EXTOPEN 呼び出し時に、書き込みモードの指定によって、権限の干渉が発生する可能性があります。この場合は、EXTOPEN でエラーで通知されます。

extdev.ini の値	EXTOPEN パラメータの値		
	「OVR」	「APP」	-
"O"	O	エラー	O
"A"	エラー	A	A

extdev.ini の値	EXTOPEN パラメータの値		
	「OVR」	「APP」	-
-	O	A	A
	意味: O:「上書き」モードが有効です。 A:「付加」モードが有効です。 エラー:EXTOPEN 呼び出しがエラーで応答します。		

LOCAL_DRIVE:ファイル属性

LOCAL_DRIVE で EXTOPEN で作成されたファイルには、次のファイル属性が割り当てられます。

- 所有者: 「user」 読み取り/書き込み権限が設定されています
- グループ: 「operator」 読み取り/書き込み権限が設定されています

10.16.3 プログラミング

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。

<ExtDev>:	**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
	タイプ: STRING
	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。
	"LOCAL_DRIVE": ローカル CF カード(定義済み)
	"CYC_DRIVE": 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)
	"/dev/ext/1",... "/dev/ext/9": 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。
	"/dev/cyc/1", "/dev/cyc/2": 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。
	"/dev/v24": 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。
	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。
	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

	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
…
```

10.16.4 必要条件

連続軌跡モードに及ぼす影響

EXTOPEN、WRITE、EXTCLOSE 命令により先読み停止がおこなわれて、その結果、連続軌跡モードが中断されます。

ブロック検索中の動作

「計算ありのブロックサーチ」中は、WRITE による出力はおこなわれません。ただし、EXTOPEN と EXTCLOSE 命令が収集され、検索ターゲットに到達すると NC スタートで有効になります。したがって、それ以降の WRITE 命令の環境は通常のプログラム処理と同じになります。

(SERUPRO の)「プログラムテスト」モードでの計算ありのブロック検索の場合、EXTOPEN、WRITE、EXTCLOSE は通常のプログラム処理の場合とまったく同じように実行されます。

10.17 電源投入、RESET/パートプログラム終了およびパートプログラム起動のシステム設定

コンセプト

コントロールシステムの動作は、以下のイベント用のマシンデータを介して設定できます。

- 起動(電源投入)
- リセット/パートプログラム終了
- パートプログラム起動

設定するコントロールシステム動作の前の事象	設定に使用するマシンデータ
起動(電源投入) *)	MD20110 $MC_RESET_MODE_MASK MD20144 $MC_TRAFO_MODE_MASK MD20150 $MC_GCODE_RESET_VALUES
リセット/パートプログラム終了	MD20110 $MC_RESET_MODE_MASK MD20150 $MC_GCODE_RESET_VALUES MD20152 $MC_GCODE_RESET_MODE
パートプログラム起動	MD20112 $MC_START_MODE_MASK MD20110 $MC_RESET_MODE_MASK
*) 「POWER ON (ページ 991)」も参照してください。	

起動後のシステム設定

MD20110 $MC_RESET_MODE_MASK、ビット 0 = 0 または 1

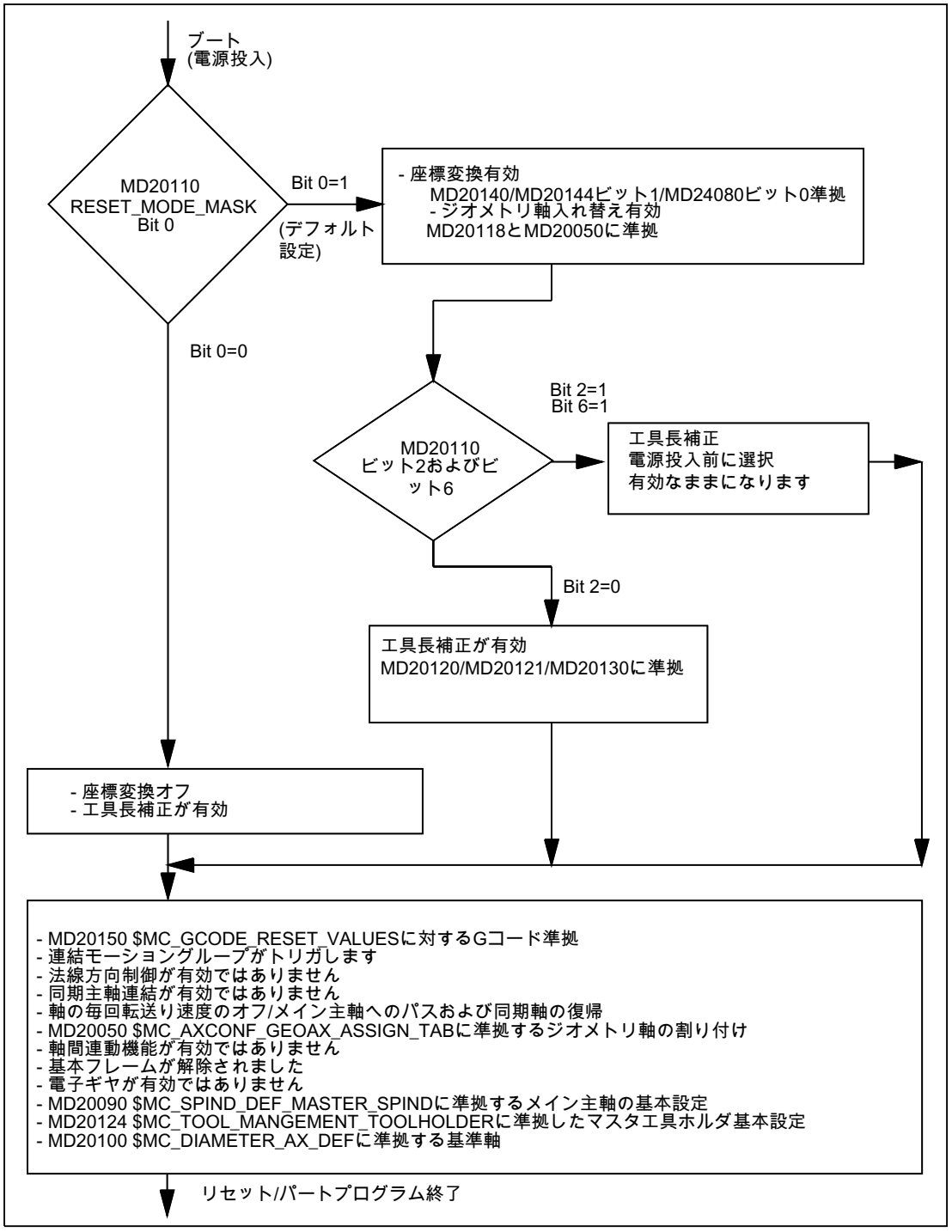


図 10-8　起動後のシステム設定(電源投入)

リセット/パートプログラム終了およびパートプログラム起動後のシステム設定

MD20110 $MC_RESET_MODE_MASK、ビット 0 = 0 または 1

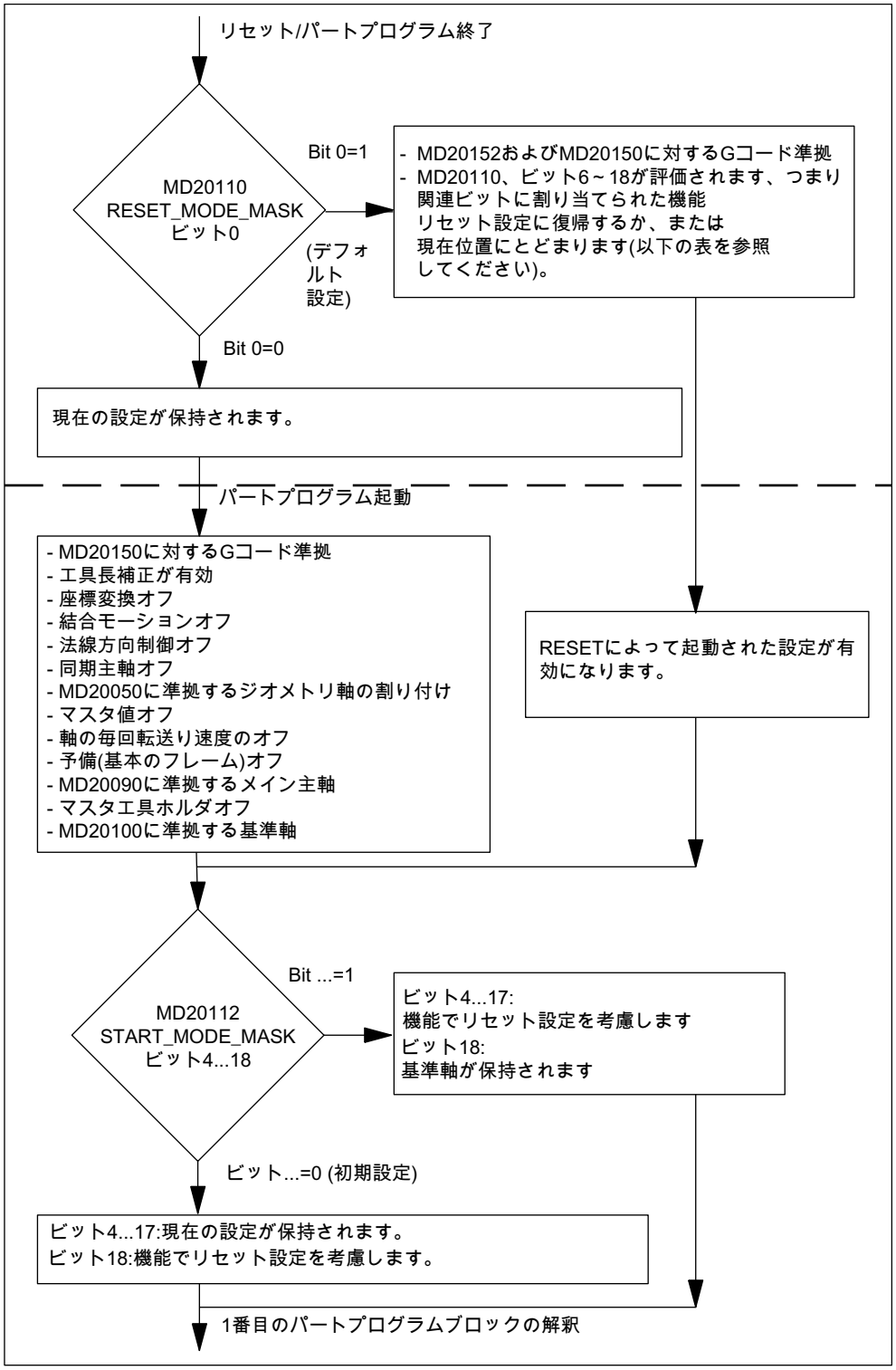


図 10-9　リセット/パートプログラム終了およびパートプログラム起動後のシステム設定

起動後およびリセット/パートプログラム終了後に有効にする G 命令

起動(電源投入)およびリセット/パートプログラム終了の後に各 G グループで有効にする G コードは、以下のマシンデータで設定します。

MD20150 $MC_GCODE_RESET_VALUES[**<G グループ>**] = **<**初期設定の G コード**>**

MD20152 $MC_GCODE_RESET_MODE[**<G グループ>**] = **<**値**>**

値	説明:G グループ毎
0	MD20150 $MC_GCODE_RESET_VALUES の初期設定 G 命令が有効です。
1	最後に有効であった/現在の G 命令が有効です。

起動後、リセット/パートプログラム終了後およびパートプログラム起動後の初期設定の制御

起動(電源投入)、リセット/パートプログラム終了、およびパートプログラム起動の後の初期制御設定は、以下のマシンデータで定義します。

- MD20110 $MC_RESET_MODE_MASK (**起動**および**リセット/パートプログラム終了**後の初期制御設定の定義)
- MD20112 $MC_START_MODE_MASK (**パートプログラム起動**後の初期制御設定の定義)

参照先

マシンデータの詳細説明

関連するマシンデータ

マシンデータ	意味
MD20120 $MC_TOOL_RESET_VALUE	起動、リセット/パートプログラム終了時の工具長補正
MD20121 $MC_TOOL_PRESEL_RESET_VALUE	リセット時の事前選択工具
MD20130 $MC_CUTTING_EDGE_RESET_VALUE	起動時の工具長補正刃先
MD20140 $MC_TRAFO_RESET_VALUE	起動座標変換データブロック
MD20144 $MC_TRAFO_MODE_MASK	キネマティック座標変換機能の選択

マシンデータ	意味
MD20150 $MC_GCODE_RESET_VALUES	G グループの初期設定
MD20152 $MC_GCODE_RESET_MODE	G グループのリセット応答
MD21330 $MC_COUPLE_RESET_MODE_1	連結取り消し動作
MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB	チャネル軸へのジオメトリ軸の割り当て
MD20118 $MC_GEOAX_CHANGE_RESET	ジオメトリ軸の自動変更が可能です

例

リセット時のリセット設定の起動

- MD20110、ビット 0 = 1
- MD20112 = 0

リセット/パートプログラム起動で座標変換を維持

- MD20110、ビット 0 = 1
- MD20110、ビット 7 = 1
- MD20112 = 0

リセット/パートプログラム起動後に工具長補正を保持

- MD20110、ビット 4 = 1
- MD20110、ビット 6 = 1
- MD20112 = 0

選択平面(ビット 4)と設定可能フレーム(ビット 5)がリセット後に有効のままであり、パートプログラム起動時にリセットされる:

- MD20110、ビット 4 = 1
- MD20110、ビット 5 = 1

- MD20112、ビット 4 = 1
- MD20112、ビット 5 = 1

注記

MD20110/MD20112、ビット 5 とビット 6

MD20110/MD20112 が、自動または MDI モードでのパートプログラム起動時に工具長補正またはフレームが有効になるようにパラメータ設定されている場合は、軸の最初のプログラミングでアブソリュート指定を使用してください(オフセットの移動のため)。

例外:MD42442/MD42440 の設定によっては、G91 のときのオフセット処理は無視されます。

10.17.1 方向座標変換による電源投入後の工具後退

機能

工具オリエンテーションでの加工運転を含むパートプログラムが停電やリセットによって中止された場合、コントローラの起動(電源投入)後に、前に有効であった座標変換を選択して、工具軸の方向でフレームを生成することができます。 工具は、JOG モードで、工具軸への後退移動により後退できます。

必要条件

有効な検出器が、座標変換に関与するすべての機械軸に対して機械原点を確立してください。「機械原点の自動復元 (ページ 1545)」の章を参照してください。

パラメータ設定

電源投入後に有効な座標変換が保持されるように以下のマシンデータを設定してください。

- MD20144 $MC_TRAFO_MODE_MASK、ビット 1 = 1
- MD20110 $MC_RESET_MODE_MASK、ビット 0 = 1
- MD20110 $MC_RESET_MODE_MASK、ビット 7 = 1

「電源投入、RESET/パートプログラム終了およびパートプログラム起動のシステム設定 (ページ 778)」の章も参照してください。

プログラミング

機械原点の待機 WAITENC

命令 WAITENC により、パラメータ設定した軸のすべての有効な検出器に対して機械原点が有効になるまで、システムはプログラム内でチャネル別に待機します。 上述の「必要条件」の節を参照してください。 軸のパラメータ設定は以下を介しておこないます。

MD34800 $MA_WAIT_ENC_VALID = 1

用途

起動時にイベント起動で呼び出されるユーザープログラム(…/_N_CMA_DIR/_N_PROG_EVENT_SPF) (必要条件: MD20108 ビット 3 = 1)では、システムは命令 WAITENC を使用して、有効な軸位置が使用可能になるまで待つ必要があります。

その後に、工具軸を X、Y、または Z 軸の方向に向けるフレームを NC 言語命令 TOROTX/TOROTY/TOROTZ を使用して生成できます。

例

インクレメンタルエンコーダによる方向座標変換と旋回軸

設定:	意味
MD10720 $MN_OPERATING_MODE_DEFAULT[0]=6	JOG モードでの起動
MD30240 $MA_ENC_TYPE [0, <軸>] = 1	インクレメンタル検出器
MD34210 $MA_ENC_REFP_STATE [0, <軸>] = 3	インクレメンタルエンコーダの軸位置の復元を可能にします。
MD20108 $MC_PROG_EVENT_MASK = 'H9'	起動時とパートプログラム起動時に、プログラム PROG_EVENT を使用してイベント起動を有効にします。
MD20152 $MC_GCODE_RESET_MODE[52] = 1	リセットにより TOFRAME を取得します。
MD20110 $MC_RESET_MODE_MASK = 'HC1'	リセットにより座標変換と工具オフセットを取得します。
MD20144 $MC_TRAFO_MODE_MASK = 'H02'	電源オフにより座標変換を取得します。

イベント起動ユーザープログラム (…/_N_CMA_DIR/_N_PROG_EVENT_SPF):

プログラムコード	コメント
; 起動時とパートプログラムの起動によるリセット時に、WCS を工具方向に位置合わせするフレームの起動を伴う例	
IF $P_PROG_EVENT == 4	; 起動
IF $P_TRAFO <> 0	; 座標変換が選択されています。

プログラムコード	コメント
WAITENC	; 旋回軸の有効な軸位置を待機します。
TOROTZ	; WCS の Z 軸を工具軸の方向に回転します。
ENDIF	
M17	
ENDIF	
IF $P_PROG_EVENT == 1	; パートプログラムの起動
TOROTOF	; 工具フレームをリセットします。
RET	
ENDIF	

WAITENC 命令は基本的に、以下のプログラムシーケンスに対応しています(AB キネマティックによる 5 軸機械の例):

プログラムコード	コメント
WHILE TRUE	; 検出器を待機します。
IF (($AA_ENC_ACTIVE[X]==TRUE) AND ($AA_ENC_ACTIVE[Y]==TRUE) AND ($AA_ENC_ACTIVE[Z]==TRUE) AND ($AA_ENC_ACTIVE[A]==TRUE) AND ($AA_ENC_ACTIVE[B]==TRUE)) GOTOF GET_LABEL	
ENDIF	
G4 F0.5	; 待機時間 0.5 秒
ENDWHILE	
:位置同期化	
GET_LABEL: GET(X,Y,Z,A,B,)	

加工継続

AUTOMATIC モード

AUTOMATIC モードでのプログラムの自動実行では、実位置が復元された有効な検出器のすべての機械軸を原点確立してください。

MDI モードとオーバストア

MDI モードでオーバストア機能の場合、軸を原点確立しないで、復元された位置で加工をおこなうこともできます。 これをおこなうには、以下により、特定のチャネルに対して、復元された位置での NC スタートを明確に有効にしてください。

MD20700 $MC_REFP_NC_START_LOCK = 2

必要条件

インクレメンタルエンコーダ付きで、現在位置を記憶していない軸

インクレメンタルエンコーダ付きで、現在位置記憶を**していない**軸は、最後の位置指令からずれないように、停電の場合には十分素早く締め付けられることを前提とします。

10.18 サブプログラムによる機能の置換

10.18.1 概要

機能

ユーザー専用の補助機能(M101 など)は、システムファクションを一切起動しません。 この補助機能は、NC/PLC インタフェースに出力されるだけです。補助機能の機能は、PLC ユーザープログラムで使用者/工作機械メーカが組み込んでください。 初期設定の NC/PLC インタフェースへの出力ではなく、ユーザー専用のサブプログラム呼び出しを設定する方法(置換サブプログラム)について説明します。

機能 M101 は、そのときはまだパートプログラムでプログラムされます。 ただし、パートプログラムの実行時に、代わりのサブプログラムが呼び出されます。 したがって、NC はサブプログラム呼び出しによって機能を置換します。 この結果、次の長所が得られます。

- 生産プロセスに合わせて調整するときに、既存のパートプログラム、テスト済みのパートプログラム、および実証済みのパートプログラムを変更せずにそのまま使用できます。 必要な変更は、その後、ユーザー専用のサブプログラムに移されます。
- NC 言語の対象機能全体を使用して、代わりのサブプログラム内で機能を実現できます。
- NC と PLC 間の通信オーバーヘッドが不要になります。

置換できる機能

次の機能がサブプログラムによって置換できます。

補助機能	
M	切り替え機能
T	工具選択
TCA	工具の状態とは無関係な工具の選択
D	工具オフセット
DL	追加工具オフセット

有効な主軸同期連結時の主軸関連機能:	
M40	自動ギヤ選択切替
M41 ～ M45	ギヤ選択切替 1 ～ 5
SPOS	主軸位置決め
SPOSA	主軸位置決め
M19	主軸位置決め

10.18.2 M、T/TCA、および D/DL 機能の置換

10.18.2.1 M 機能の置換

一般事項

M 機能の置換には、次の条件が適用されます。

- 1 つのブロック毎に、1 つの M 機能しか置換できません。
- M 機能が置換されるブロックには、次の要素を**入れないでください**。
  - M98
  - モーダルサブプログラム呼び出し
  - サブプログラムの復帰
  - パートプログラムの終了
- システムファンクションを実施する M 機能は、サブプログラムで置換しないでください(「置換できない M 機能」の章を参照してください)。

パラメータ設定

M 機能とサブプログラム

M 機能と置換サブプログラムは、次のマシンデータでパラメータ設定されます。

- MD10715 $MC_M_NO_FCT_CYCLE[**<インデックス>**] = **<**M 機能番号**>**
- MD10716 $MC_M_NO_FCT_CYCLE_NAME[**<インデックス>**] = "**<**サブプログラム名称**>**"

M 機能と対応する置換サブプログラムは、同一インデックスで連結されています。

例: M 機能 M101 はサブプログラム SUB_M101 で置換され、M 機能 M102 は SUB_M102 で置換されます。

MD10715 $MC_M_NO_FCT_CYCLE[**0**] = 101
MD10716 $MC_M_NO_FCT_CYCLE_NAME[**0**] = "SUB_M101"

MD10715 $MC_M_NO_FCT_CYCLE[**1**] = 102
MD10716 $MC_M_NO_FCT_CYCLE_NAME[**1**] = "SUB_M102"

情報伝送用のシステム変数

自由に選択できる M 機能の場合、置換済みの M 機能と置換サブプログラムでの評価のためのその他の機能(T、TCA、D、DL)に関する情報が、システム変数によって提供されます(「システム変数 (ページ 793)」の章を参照してください)。 システム変数に含まれるデータは、置換対象の M 機能がプログラム指令されるブロックを示しています。

M 機能は、置換対象の M 機能がパラメータ設定されているマシンデータ MD10715 $MC_M_NO_FCT_CYCLE[**<インデックス>**]のインデックスで選択されます。

MD10718 $MC_M_NO_FCT_CYCLE_PAR = **<インデックス>**

注記

システム変数を使用した情報の転送による M 機能の置換では、M 機能のアドレス拡張と機能値を定数値としてプログラム指令してください。

許容されるプログラミング:

- `M<機能値>`
- `M=<機能値>`
- `M[<アドレス拡張子>]=<機能値>`

不適切なプログラミング

- `M=<変数 1>`
- `M[<変数 2>]=<変数 1>`

プログラミング

M 機能の置換の規則

- 置換サブプログラムはブロックの終点で呼び出されます。
- 置換サブプログラム内では、M 機能は置換されません。
- ASUB では、ASUB が置換サブプログラム内で開始されている場合、M 機能も置換されます。

置換できない M 機能

以下の M 機能はシステムファンクションを事前定義補助機能として実施するため、サブプログラムで置換し**ない**でください。

- M0 ... M5
- M17、M30、
- M19
- M40 ... M45
- M98、M99 (MD18800 $MN_MM_EXTERN_LANGUAGE ≠ 0 の場合のみ)

マシンデータによってパラメータ設定されているユーザー専用の M 機能もまたシステムファンクションを起動するため、サブプログラムで置換しないでください。

マシンデータ	意味
MD10714 $MN_M_NO_FCT_EOP	RESET 後に有効な主軸の M 機能
MD10804 $MN_EXTERN_CHAN_M_NO_SET_INT	ASUB 起動用の M 機能(外部モード)
MD10806 $MN_EXTERN_CHAN_M_NO_DISABLE_INT	ASUB 解除用の M 機能(外部モード)
MD10814 $MN_EXTERN_M_NO_MAC_CYCLE	M 機能を介したマクロ呼び出し
MD20094 $MC_SPIND_RIGID_TAPPING_M_NR	制御軸モードへの切り替え用の M 機能
MD20095 $MC_EXTERN_RIGID_TAPPING_M_NR	制御軸モードへの切り替え用の M 機能(外部モード)
MD22254 $MC_AUXFU_ASSOC_M0_VALUE	プログラムストップの追加 M 機能
MD22256 $MC_AUXFU_ASSOC_M1_VALUE	オプショナルストップの追加 M 機能
MD26008 $MC_NIBBLE_PUNCH_CODE	M 機能の定義(ニブリング用)
MD26012 $MC_PUNCHNIB_ACTIVATION	パンチングとニブリング機能の起動

注記

例外

MD22560 $MC_TOOL_CHANGE_M_CODE (M 機能による工具交換)でパラメータ設定された M 機能は、サブプログラムで置換しないでください。

10.18.2.2 T/TCA と D/DL 機能の置換

必要条件

機能 T、TCA、D、および DL の置換では、以下の必要条件が適用されます。

- 有効な機能置換は、各ブロックで最大 1 つです。
- 機能が置換されるブロックに次の要素を含めることは**できません**。
  - M98
  - モーダルサブプログラム呼び出し
  - サブプログラムの復帰
  - パートプログラムの終了
- T = スロット番号によるマルチツール選択のためにアドレス MTL でマルチツールスロット番号がプログラムされている場合、T 置換は MTL アドレスも置換します。プログラムされた値は、$C_MTL_PROG および$C_MTL システム変数を使用して置換サブプログラムで確認できます。

パラメータ設定:置換サブプログラム

置換サブプログラムは、マシンデータで機能別に指定されます。

機能	マシンデータ
T	MD10717 $MN_T_NO_FCT_CYCLE_NAME
TCA	MD15710 $MN_TCA_CYCLE_NAME
D/DL	MD11717 $MN_D_NO_FCT_CYCLE_NAME

注記

T、TCA、および D/DL 機能の置換には、同じサブプログラムを使用してください。

パラメータ設定:D 機能または DL 機能と同時にプログラムされた T 機能に関する動作

D 機能または DL 機能と T 機能を 1 つのブロック内で同時にプログラム指令されている場合、D 番号と DL 番号のどちらかがパラメータとして置換サブプログラムに転送されるか、D 関数または DL 関数が置換プログラムの呼び出し前に実行されます。動作は次のマシンデータを使用して設定できます。

MD10719 $MN_T_NO_FCT_CYCLE_MODE (T 機能置換のパラメータ設定)

ビット	値	意味
0	0	D または DL 番号は、システム変数の形式でサブプログラム内で使用できます(初期状態)。
	1	D または DL 番号はブロック内で直接計算されます。 **注:** この機能は、M 機能によって工具交換が設定された場合にのみ有効です。 MD22550 $MC_TOOL_CHANGE_MODE = 1 そうでない場合は、D または DL 値が常に転送されます。

情報伝送用のシステム変数

置換サブプログラムには、ブロック内にプログラムされている機能に関連するすべての情報がシステム変数によって提供されます(「システム変数 (ページ 793)」の章を参照してください)。

システム変数に含まれるデータは、置換対象の機能がプログラム指令されたブロックを示しています。

パラメータ設定:置換サブプログラムの呼び出しタイミング

置換サブプログラムの呼び出しタイミングは、次のマシンデータを使用して設定します。

MD10719 $MN_T_NO_FCT_CYCLE_MODE、ビット 1 とビット 2

ビット 2	ビット 1	置換サブプログラムの呼び出しタイミング
0	0	ブロック終点で 置換サブプログラムが実行された後、置換操作がトリガされた行の次のプログラム行によって解析が再開されます。
0	1	ブロック開始時 置換サブプログラムが実行された後、置換サブプログラムの呼び出しの結果として得られたプログラム行が解析されます。T アドレスおよび D または DL アドレスと工具交換の M 機能は処理されなくなります。
1	-	ブロック開始時と終了時 置換プログラムが 2 回呼び出されます。

呼び出しタイミングのシステム変数

システム変数$P_SUB_STAT を使用して置換が有効かどうかを読み取り、有効であった場合は、置換サブプログラム – ブロックを基準にした – がいつ呼び出されたかを読み取ることができます。

値	意味
0	置換が無効
1	置換が有効。サブプログラム呼び出しは、ブロックの開始点でおこなわれます。
2	置換が有効。サブプログラム呼び出しは、ブロックの終点でおこなわれます。

例:T 機能の置換

パラメータ設定	意味
MD22550 $MC_TOOL_CHANGE_MODE = 0	T 機能による工具交換
MD10717 $MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	T 機能を置換するサブプログラムの名称
MD10719 $MN_T_NO_FCT_CYCLE_MODE = 0	呼び出しタイミングブロック終了

プログラミング	コメント
`N110 D1`	`; D1`
`N120 G90 G0 X100 Y100 Z50`	`; D1 は有効です。`

プログラミング	コメント
N130 D2 X110 Z0 T5	; D1 は有効なままです。T 機能が MY_T_CYCLE サブプログラム呼び出しによってブロックの終点で置換されます。D2 がシステム変数で MY_T_CYCLE を提供します。

T 機能の置換の詳細な例は、「工具交換時の M/T 機能の置換例 (ページ 795)」の章を参照してください。

10.18.2.3 システム変数

一般事項

置換サブプログラムには、ブロック内にプログラム指令されている機能(T または TCA、D または DL、M)に関連するすべての情報がシステム変数によって提供されます。

例外

以下の場合、D または DL 番号は転送されません。

- MD10719 $MN_T_NO_FCT_CYCLE_MODE、ビット 0 = 1
- MD22550 $MC_TOOL_CHANGE_MODE = 1

および

- D または DL が 1 つのブロックで T または M 機能と一緒にプログラム指令されている場合

⚠ 注意

値が機能しない

システム変数で置換サブプログラムに提供された値は、また有効になっていません。使用者/工作機械メーカが、置換サブプログラムで適切なプログラミングを使用してこれを解決してください。

システム変数

システム変数	意味
$C_M_PROG	M 機能がプログラム指令されている場合は TRUE
$C_M	$C_M_PROG == TRUE の場合、アドレス M の値が含まれます。 ここでは、2 つの事例に分ける必要があります。 • M 機能による工具交換の場合、サブプログラムがパラメータの転送で設定されていれば、$C_M で値が提供されます。 MD10715 MN_M_NO_FCT_CYCLE • アドレス T または D/DL、あるいはその両方に対してサブプログラムが 1 つだけ設定されており、プログラム内で工具交換の M 機能が置換するアドレスのどれかとともにプログラム指令されている場合、$C_M で値が提供されます。 MD22560 $MC_TOOL_CHANGE_M_CODE
$C_AUX_VALUE[0]	置換された M 機能の値
$C_ME	$C_M_PROG == TRUE の場合、M 機能のアドレス拡張の値が含まれます。
$C_AUX_EXT[0]	M 機能のアドレス拡張($C_ME と同じ)
$C_AUX_IS_QUICK[0]	PLC への高速出力を伴う M 機能をプログラム指令した場合は TRUE
$C_T_PROG	T 機能をプログラム指令した場合は TRUE
$C_T	$C_T_PROG == TRUE の場合、T 機能の値が含まれます。
$C_TE	以下の場合 • $C_T_PROG == TRUE • $C_TS_PROG == TRUE T 機能のアドレス拡張の値が含まれます。
$C_TS_PROG	T または TCA 置換用に工具名称がプログラムされている場合、TRUE
$C_TS	$C_TS_PROG == TRUE の場合、T または TCA 置換用にプログラムされた工具名称が含まれます
$C_TCA	TCA 置換が有効の場合、TRUE
$C_DUPLO_PROG	TCA 置換の予備工具番号がプログラム指令されている場合、TRUE

システム変数	意味
$C_DUPLO	$C_DUPLO_PROG == TRUE の場合、プログラム指令された予備工具番号が含まれます。
$C_THNO_PROG	TCA 置換の工具ホルダ番号/主軸番号がプログラム指令されている場合、TRUE
$C_THNO	$C_THNO_PROG == TRUE の場合、プログラム指令された工具ホルダ番号/主軸番号の値が含まれます。
$C_D_PROG	D 機能がプログラム指令されている場合は TRUE
$C_D	$C_D_PROG == TRUE の場合 D 機能の値が含まれます。
$C_DL_PROG	DL 機能をプログラム指令した場合は TRUE
$C_DL	$C_DL_PROG == TRUE の場合、DL 機能の値が含まれます。
$P_SUB_STAT	置換サブプログラムのブロック単位の呼び出しタイミング
$C_MTL_PROG	アドレス MTL がプログラムされていれば TRUE
$C_MTL	$C_MTL_PROG == TRUE の場合、アドレス MTL の値が含まれます

10.18.2.4 例: M 機能の置換

例 1

機能 M6 はサブプログラム「SUB_M6」の呼び出しによって置換されます。

工具交換に関連した情報は、システム変数を使用して転送してください。

パラメータ設定

マシンデータ	意味
MD10715 $MN_M_NO_FCT_CYCLE[2] = 6	M6 による工具交換
MD10716 $MN_M_NO_FCT_CYCLE_NAME[2] = "SUB_M6"	M6 の置換サブプログラム
MD10718 $MN_M_NO_FCT_CYCLE_PAR = 2	システム変数を使用した情報の転送

メインプログラム

プログラミング	コメント
PROC MAIN	
...	;
N10 T1 D1 M6	; M6 はサブプログラム「SUB_M6」で置換されます
	;
...	;
N90 M30	

サブプログラム「SUB_M6」

プログラミング	コメント
PROC SUB_M6	
N110 IF $C_T_PROG==TRUE	; アドレス T がプログラム指令されている場合
N120 T[$C_TE] = $C_T	; T 選択の実行
N130 ENDIF	; ENDIF
N140 M[$C_ME] = 6	; 工具交換の実行。
N150 IF $C_D_PROG==TRUE	; アドレス D がプログラム指令されている場合
N160 D=$C_D	; D 選択の実行
N170 ENDIF	; ENDIF
N190 M17	

例 2

T 機能によって新規工具の交換が準備されます。工具交換は機能 M6 によってのみ実現されます。T 機能は、サブプログラム「MY_T_CYCLE」の呼び出しによって置換されます。D/DL 番号がサブプログラムに転送されます。

パラメータ設定

パラメータ設定	意味
MD22550 $MC_TOOL_CHANGE_MODE = 1	T 機能で準備された工具交換
MD10717 $MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	置換サブプログラム
MD10719 $MN_T_NO_FCT_CYCLE_MODE = 0	D/DL 番号の転送

メインプログラム

プログラムコード	コメント
N210 D1	;
N220 G90 G0 X100 Y100 Z50	; D1 は有効です。

プログラムコード	コメント
N230 D2 X110 Z0 T5	; D1 は有効のまま、プログラム指令された D2 が
	; 変数としてサブプログラムに転送されます。
N240 M6	; 工具交換の実行

例 3

T 機能によって新規工具の交換が準備されます。工具交換は機能 M6 によってのみ実現されます。T 機能は、サブプログラム「MY_T_CYCLE」の呼び出しによって置換されます。D/DL 番号はサブプログラムに転送**されません**。

パラメータ設定

パラメータ設定	意味
MD22550 $MC_TOOL_CHANGE_MODE = 1	T 機能で準備された工具交換
MD10717 $MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	置換サブプログラム
MD10719 $MN_T_NO_FCT_CYCLE_MODE = 1	D/DL 番号の転送なし

メインプログラム

プログラムコード	コメント
N310 D1	
N320 G90 G0 X100 Y100 Z50	; D1 は有効です。
N330 D2 X110 Z0 T5	; D2 は有効で、変数として
	; 置換サブプログラムに転送されません。
N340 M6	; 工具交換の実行。

例 4

T と M6 機能は、サブプログラム「MY_T_CYCLE」の呼び出しによって置換されます。

M6 の置換時に、パラメータがサブプログラムに転送されます。

M6 が D または DL とともにブロックでプログラム指令されている場合、D/DL 番号の転送がパラメータ設定されていなければ、D または DL 番号もパラメータとしてサブプログラムに転送されます。

MD10719 $MN_T_NO_FCT_CYCLE_MODE = 1

パラメータ設定

設定	意味
MD22550 $MC_TOOL_CHANGE_MODE = 1	M 機能による工具交換
MD22560 $MC_TOOL_CHANGE_M_CODE = 6	工具交換の M コード
MD10715 $MC_M_NO_FCT_CYCLE[3] = 6	置換する M 機能
MD10716 $MC_M_NO_FCT_CYCLE_NAME[3] = "MY_T_CYCLE"	M 機能の置換サブプログラム
MD10717 $MN_T_NO_FCT_CYCLE_NAME = "MY_T_CYCLE"	T 機能の置換サブプログラム
MD10718 $MN_M_NO_FCT_CYCLE_PAR = 3	M6 の置換サブプログラムへのパラメータ転送
MD10719 $MN_T_NO_FCT_CYCLE_MODE = 1	D/DL 番号の転送なし

メインプログラム

プログラムコード	コメント
`N410 D1`	
`N420 G90 G0 X100 Y100 Z50`	`; D1 は有効です。`
`N330 D2 X110 Z0 T5 M6`	`; D1 は有効なまま、D2 と T5 が変数として M6 置換サブプログラムに転送されます。`

10.18.2.5 例: T 機能と D 機能の置換

T 機能と D 機能は、サブプログラム「D_T_SUB_PROG」の呼び出しによって置換されます。この例では、次の条件も TRUE にします。

- 工具交換はアドレス T を使用して実行
- サブプログラムはブロックの開始点で呼び出し
- 工具管理機能が動作中でない
- 軸はカップリングギヤ付きの割り出し軸です

パラメータ設定

マシンデータ	意味
MD11717 $MN_**D**_NO_FCT_CYCLE_NAME = "**D_T_SUB_PROG**"	D 機能の置換サブプログラム

MD10717 $MN_T_NO_FCT_CYCLE_NAME = "D_T_SUB_PROG"	M 機能の置換サブプログラム
MD10719 $MN_T_NO_FCT_CYCLE_MODE = 'H2'	ブロック開始時に呼び出し
MD22550 $MC_TOOL_CHANGE_MODE = 0	T 機能による工具交換

メインプログラム

プログラミング	コメント
PROC MAIN	
...	;
N10 G01 F1000 X10 T1=5 D1	; ブロックの開始点での ; 「D_T_SUB_PROG」の呼び出しによる T 機能と D 機能の置換
...	;
N90 M30	

サブプログラム「D_T_SUB_PROG」

プログラミング	コメント
N1000 PROC D_T_SUB_PROG DISPLOF SBLOF	
N4100 IF $C_T_PROG==TRUE	; アドレス T がプログラム指令されている場合
N4120 POS[B] = CAC($C_T)	; アプローチ割り出し位置
N4130 T[$C_TE] = $C_T	; 工具の選択 (T 選択)
N4140 ENDIF	; ENDIF
N4300 IF $C_T_PROG==TRUE	; アドレス D がプログラム指令されている場合
N4320 D=$C_D	; オフセットの選択 (D 選択)
N4330 ENDIF	; ENDIF
N4400 IF $C_DL_PROG==TRUE	; アドレス DL がプログラム指令されている場合
N4420 D=$C_DL	; 挿入オフセットの選択
N4430 ENDIF	; ENDIF
N9999 RET	

10.18.2.6 競合時の動作

競合する場合

1 つのブロックに複数の機能がプログラム指令されていて、これらの機能が異なったサブプログラムと置換される場合は競合が発生します。

- アドレス D と DL は以下のサブプログラムと置換されます:
  MD11717 $MN_FCT_CYCLE_NAME = "**D_SUB_PROG**"
- アドレス T は以下のサブプログラムと置換されます:
  MD10717 $MN_FCT_CYCLE_NAME = "**T_SUB_PROG**"
- M 機能 M6 は以下のサブプログラムと置換されます:
  MD10715 $MN_M_NO_FCT_CYCLE[0] = 6
  MD10716 $MN_M_NO_FCT_CYCLE_NAME[0] = "**M6_SUB_PROG**"
  MD10718 $MN_M_NO_FCT_CYCLE_PAR = 0
  MD22550 $MC_TOOL_CHANGE_MODE = 1
  MD22560 $MC_TOOL_CHANGE_M_CODE = 6

分解能

競合は以下の表に対応して解決されます:

次の機能は、1 行のプログラム行でプログラム指令されます。			呼び出されるサブプログラム:
D や DL	T または TCA	M6	
–	–	x	**M6**_SUB_PROG
–	x	–	**T**_SUB_PROG
–	x	x	**M6**_SUB_PROG
x	–	–	**D**_SUB_PROG
x	–	x	**M6**_SUB_PROG
x	x	–	**T**_SUB_PROG
x	x	x	**M6**_SUB_PROG

10.18.3 主軸機能の置換

10.18.3.1 一般情報

機能

連結が有効の場合、スレーブ主軸機能をマスタ主軸と置換できます。

- M40: 自動ギヤ選択切替
- M41 ... M45 プログラム指令されたギヤ選択切替
- SPOS、SPOSA、および M19:主軸位置決め

境界条件

- 主軸機能を置換するには、以下の条件を満たす必要があります。
  - プログラム指令された主軸が有効な連結のマスタ主軸であること。
  - マスタ主軸とスレーブ主軸が同じチャネル内にあること。 これが検出されるのは、マスタ主軸が、連結がおこなわれたチャネルに割り当てられている場合に限ります。 マスタ主軸が別のチャネルに変更されると、ギヤ選択切替またはこの主軸の位置決めを実行しても、置換サブプログラムが呼び出されません。
  - プログラム指令されたギヤ選択切替により、実際にギヤ選択切替がおこなわれること。. このために、プログラム指令されたギア切替と動作中のギア切替が異なっている必要があります。
- ブロック内で置換できる主軸機能は 1 つだけです。 置換を複数実行すると、プログラムの処理が中止されます。 置換される主軸機能は、その後、複数のブロックに分散されます。

パラメータ設定

主軸機能

サブプログラムを使用して置換する主軸機能は、次のマシンデータで選択されます。

MD30465 $MA_AXIS_LANG_SUB_MASK

ビット	意味	
0	自動(M40)と直接(M41 ～ M45)ギヤ選択切替	
	値	意味
	0	置換なし
	1	MD15700 と MD15702 で設定されたサブプログラムを使用した置換
1	SPOS / SPOSA / M19 による主軸位置決め	
	値	意味
	0	置換なし
	1	MD15700 と MD15702 で設定されたサブプログラムを使用した置換

サブプログラム:名称

置換サブプログラムの名称は、次のマシンデータで入力されます:

MD15700 $MN_LANG_SUB_NAME = "<サブプログラム名称>"

サブプログラム:パス

置換サブプログラムのパスは、次のマシンデータで設定されます:

MD15702 $MN_LANG_SUB_PATH = <値>

値	意味
0	メーカーサイクルフォルダ: /_N_CMA_DIR
1	ユーザーサイクルフォルダ: /_N_CUS_DIR
2	シーメンス サイクルフォルダ: /_N_CST_DIR

システム変数: 置換サブプログラムの呼び出しタイミング

置換サブプログラムの呼び出しタイミングは、システム変数$P_SUB_STAT を使用して読み取られます:

値	意味
0	置換が無効
1	置換が有効。サブプログラム呼び出しは、ブロックの開始点でおこなわれます。
2	置換が有効。サブプログラム呼び出しは、ブロックの終点でおこなわれます。

ブロックの処理

置換サブプログラムがブロックの開始点で呼び出された場合、置換サブプログラムの処理後に、呼び出しを開始したブロックが処理されます。 置換された命令はそれ以上処理されません。

置換サブプログラムがブロックの終了時に呼び出された場合、置換サブプログラムの呼び出しを開始したブロックは、置換される命令を使用せずに最初に処理されます。 その後で、置換サブプログラムが呼び出されます。

10.18.3.2 M40 ～ M45 (ギヤ選択切替)の置換

機能

連結が有効の場合、マスタ主軸のギア選択切替(M40、M41 ... M45)の命令は、ユーザー専用サブプログラムの呼び出しによって置換されます。

パラメータ設定

起動

- MD30465 $MA_AXIS_LANG_SUB_MASK、ビット 0 = 1

サブプログラムの呼び出しタイミング

- `M40`
  呼び出しタイミングを設定できません。 置換サブプログラムは常にブロックの開始点で呼び出されます。
- `M41` ... `M45`
  呼び出しタイミングは、PLC に対する補助機能の設定された出力動作によって異なります(下記の MD22080 を参照してください):
  – 移動**前**または移動**中**の出力: **ブロック開始点で**サブプログラム呼び出し
  – 移動**後**の出力 **ブロック終点で**サブプログラム呼び出し
  MD22080 $MC_AUXFU_PREDEF_SPEC[12 ... 16] (`M41` ... `M45` の出力動作)

ビット	値	意味
5	1	動作**前**の補助機能の出力
6	1	動作**中**の補助機能の出力
7	1	動作**後**の補助機能の出力

情報転送用のシステム変数

置換サブプログラムには、ブロック内にプログラム指令されている関数に関連するすべての情報がシステム変数によって提供されます(「システム変数 (ページ 805)」の章を参照してください)。 このデータは、置換する機能がプログラム指令されているブロックについてのみ述べています。

10.18.3.3 SPOS、SPOSA、M19 (主軸位置決め)の置換

機能

連結が有効の場合、マスタ主軸の位置決め命令(SPOS、SPOSA、または M19)はユーザー専用サブプログラム(置換サブプログラム)の呼び出しにより置換されます。

適用例

2 主軸機械でワークを同時に加工する場合、主軸は 1 以外の連結係数で連結されます。 工具の交換時に、主軸を同じ位置に移動させてください。 置換サブプログラムが連結を開き、主軸を工具交換位置に個別に位置決めした後、連結をもう一度閉じます。

パラメータ設定

起動

- MD30465 $MA_AXIS_LANG_SUB_MASK、ビット 1 = 1

置換サブプログラムの呼び出しタイミング

- SPOS、SPOSA
  呼び出しタイミングを設定できません。 置換サブプログラムは常にブロックの開始点で呼び出されます。
- M19
  呼び出しタイミングは、PLC に対する補助機能の設定された出力動作によって異なります(下記の MD22080 を参照してください):
  - 移動**前**または移動**中**の出力: **ブロック開始点で**サブプログラム呼び出し
  - 移動**後**の出力 **ブロック終点で**サブプログラム呼び出し

MD22080 $MC_AUXFU_PREDEF_SPEC[9]		
ビット	値	意味
5	1	動作**前**の補助機能の出力
6	1	動作**中**の補助機能の出力
7	1	動作**後**の補助機能の出力

情報転送用のシステム変数

置換サブプログラムには、ブロック内にプログラム指令されている関数に関連するすべての情報がシステム変数によって提供されます(「システム変数 (ページ 805)」の章を参照してください)。 このデータは、置換する機能がプログラム指令されているブロックについてのみ述べています。

10.18.3.4 システム変数

システム変数	意味
$P_SUB_AXFCT	M40、M41 ... M45 置換が有効の場合、TRUE
$P_SUB_GEAR	プログラム指令または計算されたギヤ選択 置換サブプログラム以外: メイン主軸のギヤ選択
$P_SUB_AUTOGEAR	置換操作を開始したブロックで M40 が有効の場合、TRUE 置換サブプログラム以外: インタープリタの実際の設定
$P_SUB_LA	置換操作を起動した、有効な連結のマスタ主軸の軸名称が含まれます。 **注** 変数が置換サブプログラム以外で使用された場合、プログラムの処理がアラームと共にキャンセルされます。
$P_SUB_CA	置換操作を起動した、有効な連結のスレーブ主軸の軸名称が含まれます。 **注** 変数が置換サブプログラム以外で呼び出された場合、プログラムの処理がアラームと共にキャンセルされます。
$P_SUB_AXFCT	MD30465 $MA_AXIS_LANG_SUB_MASK に対応した有効な置換タイプが含まれます。

システム変数	意味
$P_SUB_SPOS	SPOS 置換が有効の場合、TRUE
$P_SUB_SPOSA	SPOSA 置換が有効の場合、TRUE
$P_SUB_M19	M19 置換が有効の場合、TRUE
$P_SUB_SPOSIT	プログラム指令された主軸位置が含まれます。 **注** 変数が置換サブプログラム以外で呼び出された場合、プログラムの処理がアラームと共にキャンセルされます。
$P_SUB_SPOSMODE	プログラムされた主軸位置の位置アプローチモードが含まれます。

値	意味
0	位置アプローチモードの変更なし
1	AC
2	IC
3	DC
4	ACP
5	ACN
6	OC
7	PC

注: 変数が置換サブプログラム以外で呼び出された場合、プログラムの処理がアラームと共にキャンセルされます。

$P_SUB_STAT	置換サブプログラムのブロック単位の呼び出しタイミング

10.18.3.5 例:ギヤ選択切替

サブプログラムで、すべてのギヤ選択切替命令 M40、M41 ... M45 が置換されます。

パラメータ設定

マシンデータ	意味
MD15700 $MN_LANG_SUB_NAME = "LANG_SUB"	サブプログラム
MD15702 $MN_LANG_SUB_PATH = 0	工作機械メーカのフォルダ
MD22080 $MC_AUXFU_PREDEF_SPEC[12] = 'H21'	M41:移動前の出力

MD22080 $MC_AUXFU_PREDEF_SPEC[13] = 'H21' M42:移動前の出力
MD22080 $MC_AUXFU_PREDEF_SPEC[13] = 'H21' M43:移動前の出力
MD22080 $MC_AUXFU_PREDEF_SPEC[15] = 'H21' M44:移動前の出力
MD22080 $MC_AUXFU_PREDEF_SPEC[16] = 'H21' M45:移動前の出力
MD30465 $MA_AXIS_LANG_SUB_MASK[AX5] = 'H0001' ギヤ切替命令の置換

メインプログラム

プログラミング	コメント
PROC MAIN	
N110 COUPON(S2,S1)	; 主軸同期連結を閉じます
N120 G01 F100 X100 S5000 M3 M43	; M43 によるサブプログラム呼び出し
N130 M40	; 自動ギヤ選択切替オン
N140 M3 S1000	; S1000 によるサブプログラム呼び出し
	; その結果としての自動
	; ギヤ選択切替の起動
N9999 M30	

置換サブプログラム「LANG_SUB」、バージョン 1

主軸(S1: マスタ主軸、S2: スレーブ主軸)への直接的なアドレス指定により、簡略化と高速化のために最適化

プログラミング	コメント
N1000 PROC LANG_SUB DISPLOF SBLOF	
N1100 IF($P_SUB_AXFCT==1)	; ギヤ選択切替による置換
N1140 DELAYFSTON	; 停止遅延領域の始まり
N1150 COUPOF(S2,S1)	; 主軸同期連結を開く
N1160 ;マスタ主軸とスレーブ主軸の個別のギヤ選択切替	
N1170 M1=$P_SUB_GEAR M2=$P_SUB_GEAR	
N1180 DELAYFSTON	; 停止遅延領域の終わり
N1190 COUPON(S2,S1)	; 主軸同期連結を閉じます
N1200 ENDIF	
...	
N9999 RET	

置換サブプログラム「LANG_SUB」、バージョン 2

システム変数(マスタ主軸: $P_SUB_LA、スレーブ主軸: $P_SUB_CA)を使用した柔軟な間接アドレス指定。

```
プログラミング                                コメント
N1000 PROC LANG_SUB DISPLOF SBLOF
N1010 DEF AXIS _LA                    ; マスタ軸/マスタ主軸のビットメモリ
N1020 DEF AXIS _CA                    ; スレーブ軸/スレーブ主軸のビットメモリ
N1030 DEF INT _GEAR                   ; ギヤ選択のビットメモリ

N1100 IF($P_SUB_AXFCT==1)             ; ギヤ選択切替による置換
N1110 _GEAR=$P_SUB_GEAR               ; 有効化するギヤ選択
N1120 _LA=$P_SUB_LA                   ; マスタ主軸の軸名称
N1130 _CA=$P_SUB_CA                   ; スレーブ主軸の軸名称
N1140 DELAYFSTON                      ; 停止遅延領域の始まり
N1150 COUPOF(_CA,_LA)                 ; 主軸同期連結を開く
N1160    ;マスタ主軸とスレーブ主軸のギヤ選択切替
N1170 M[AXTOSPI(_LA)]=_GEAR M[AXTOSPI(_CA)]=_GEAR
N1180 DELAYFSTOF                      ; 停止遅延領域の終わり
N1190 COUPON(_CA,_LA)                 ; 主軸同期連結を閉じます
N1200 ENDIF
...
N9999 RET
```

10.18.3.6 例:主軸位置決め

サブプログラムでは、SPOS と SPOSA 命令の置換だけが明示的に実行されます。その他の置換は、基本的に同じ方法で補完してください。

パラメータ設定

マシンデータ	意味
MD30465 $MA_AXIS_LANG_SUB_MASK[AX5]='H0002'	位置決め命令の置換
MD22080 $MC_AUXFU_PREDEF_SPEC[9]='H0021'	移動前に PLC へ M19 を出力

セッティングデータ	意味
SD43240 $SA_M19_SPOS[AX5]=260	M19 の主軸位置 = 260
SD43250 $SA_M19_SPOSMODE[AX5]=4	M19 の位置アプローチモード:「正方向のアプローチ(ACP)」

メインプログラム

プログラミング	コメント
PROC MAIN	
...	
N210 COUPON(S2,S1)	; 主軸同期連結を起動します
N220 SPOS[1]=100	; SPOS によるマスタ主軸の位置決め
...	
N310 G01 F1000 X100 M19	; M19 によるマスタ主軸の位置決め

置換サブプログラム「LANG_SUB」、バージョン 1

主軸(S1: マスタ主軸、S2: スレーブ主軸)への直接的なアドレス指定により、簡略化と高速化のために最適化

プログラミング	コメント
N1000 PROC LANG_SUB DISPLOF SBLOF	
N2100 IF($P_SUB_AXFCT==2)	
N2110 ;有効な主軸同期連結のための SPOS/SPOSA/M19 の置換	
N2185 DELAYFSTON	; 停止遅延領域の始まり
N2190 COUPOF(S2,S1)	; 主軸同期連結を開く
N2200	; マスタ主軸とスレーブ主軸の位置決め
N2210 IF($P_SUB_SPOS==TRUE) OR ($P_SUB_SPOSA==TRUE)	
N2220 ;SPOS と SPOSA が SPOS にマッピングされます	
N2230 CASE $P_SUB_SPOSMODE OF \	
0 GOTOF LABEL1_DC \	
1 GOTOF LABEL1_IC \	
2 GOTOF LABEL1_AC \	
3 GOTOF LABEL1_DC \	
4 GOTOF LABEL1_ACP \	
5 GOTOF LABEL1_ACN \	
DEFAULT GOTOF LABEL_ERR	
LABEL1_DC:SPOS[1]=DC($P_SUB_SPOSIT) SPOS[2]=DC($P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	

```
プログラミング                                    コメント
LABEL1_IC:DELAYFSTOF
            SPOS[1]=IC($P_SUB_SPOSIT) SPOS[2]=IC($P_SUB_SPOSIT)
            DELAYFSTON
            GOTOF LABEL1_CONT
LABEL1_AC:SPOS[1]=AC($P_SUB_SPOSIT) SPOS[2]=AC($P_SUB_SPOSIT)
            GOTOF LABEL1_CONT
LABEL1_ACP:SPOS[1]=ACP($P_SUB_SPOSIT) SPOS[2]=ACP($P_SUB_SPOSIT)
            GOTOF LABEL1_CONT
LABEL1_ACN:SPOS[1]=ACN($P_SUB_SPOSIT) SPOS[2]=ACN($P_SUB_SPOSIT)
LABEL1_CONT:
N2250 ELSE                                  ; M19 による主軸の位置決め
N2270 M1=19 M2=19                           ; マスタ主軸とスレーブ主軸
N2280 ENDIF                                 ; SPOS、SPOSA の置換の終了
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    ;ここからさらに置換を処理します
...
N3300 ENDIF                                 ; 置換の終了
...
N9999 RET                                   ; 通常のプログラムの終了
LABEL_ERR:SETAL(61000)                      ; 異常が発生しました
```

置換サブプログラム「LANG_SUB」、バージョン 2

システム変数(マスタ主軸: $P_SUB_LA、スレーブ主軸: $P_SUB_CA)を使用した柔軟な間接アドレス指定。

```
プログラミング                                    コメント
N1000 PROC LANG_SUB DISPLOF SBLOF
N1010 DEF AXIS _LA                          ; マスタ軸/主軸
N1020 DEF AXIS _CA                          ; スレーブ軸/主軸
N1030 DEF INT _LSPI                         ; マスタ主軸番号(プログラム指令された
                                            ; 主軸)
N1040 DEF INT _CSPI                         ; スレーブ主軸番号
...
N2100 IF($P_SUB_AXFCT==2)
N2110 ;有効な主軸同期連結のための ; SPOS/SPOSA/M19 の置換
N2120 _LA=$P_SUB_LA                         ; マスタ主軸の軸名称
N2130 _CA=$P_SUB_CA                         ; スレーブ主軸の軸名称
N2140 _LSPI=AXTOSPI(_LA)                    ; マスタ主軸の番号
N2180 _CSPI=AXTOSPI(_LA)                    ; スレーブ主軸の番号
N2185 DELAYFSTON                            ; 停止遅延領域の始まり
```

プログラミング	コメント
N2190 COUPOF(_CA,_LA)	; 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 ;SPOS と SPOSA が SPOS にマッピングされます	
N2230 CASE $P_SUB_SPOSMODE OF	
0 GOTOF LABEL1_DC \	
1 GOTOF LABEL1_IC \	
2 GOTOF LABEL1_AC \	
3 GOTOF LABEL1_DC \	
4 GOTOF LABEL1_ACP \	
5 GOTOF LABEL1_ACN \	
DEFAULT GOTOF LABEL_ERR	
LABEL1_DC:SPOS[_LSPI]=DC($P_SUB_SPOSIT) SPOS[_CSPI]=DC($P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_IC:DELAYFSTOF	
SPOS[_LSPI]=IC($P_SUB_SPOSIT) SPOS[_CSPI]=IC($P_SUB_SPOSIT)	
DELAYFSTON	
GOTOF LABEL1_CONT	
LABEL1_AC:SPOS[_LSPI]=AC($P_SUB_SPOSIT) SPOS[_CSPI]=AC($P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_ACP:SPOS[_LSPI]=ACP($P_SUB_SPOSIT) POS[_CSPI]=ACP($P_SUB_SPOSIT)	
GOTOF LABEL1_CONT	
LABEL1_ACN:SPOS[_LSPI]=ACN($P_SUB_SPOSIT) POS[_CSPI]=ACN($P_SUB_SPOSIT)	
LABEL1_CONT:	
N2250 ELSE	; M19 による主軸位置決め
N2270 M[_LSPI]=19 M[_CSPI]=19	
N2280 ENDIF	
N2285 DELAYFSTOF	; 停止遅延領域の終わり
N2290 COUPON(_CA,_LA)	; 主軸同期連結を起動します
N2410 ELSE	
N2420 ;ここからさらに置換を処理します	
...	
N3300 ENDIF	
...	
N9999 RET	; 通常のプログラムの終了
LABEL_ERR:SETAL(61000)	; 異常が発生しました

10.18.4 サブプログラムのプロパティ

一般の規則

- 置換の実行時に呼び出されたサブプログラムには、命令 PROC と属性 SBLOF および DISPLOF が含むことができます。
- 置換は ISO 言語モードでもおこなわれます。ただし、置換サブプログラムは標準言語モード(シーメンス)でのみ処理されます。標準言語モードに自動的に切替えられます。置換サブプログラムからの復帰ジャンプで、元の言語モードが再選択されます。
- システム変数は、情報をサブプログラムに転送するためにだけ使用されます。パラメータの転送はおこなえません。
- シングルブロックと属性 SBLOF の動作は、次の設定によって異なります:
  MD10702 IGNORE_SINGLEBLOCK_MASK、ビット 14 (シングルブロック停止のマスク)

値	意味
0	置換サブプログラムは「通常の」サブプログラムと同じように動作します。 ● M17 による復帰ジャンプ:サブプログラムの最後で停止 **注** PLC での M 機能の出力は以下によって異なります: MD20800 $MC_SPF_END_TO_VDI、ビット 0 (PLC へのサブプログラム終了) - ビット 0 = 0:出力なし - ビット 0 = 1:M17 が PLC に出力されます。 ● RET による復帰ジャンプ:置換サブプログラムの最後で停止なし
1	置換サブプログラムが呼び出されたブロックでは、**停止は 1 回だけおこなわれます。**以下には関係なくおこなわれます。 ● サブプログラムが呼び出されたのが、ブロック開始時かブロック終了時か。 ● サブプログラムで他のサブプログラムが呼び出されているか。 ● サブプログラムが M17 または RET で終了しているか。 シングルブロック停止は、置換サブプログラムの終わりに M 機能の置換に対して実行されます。 T 機能と D/DL 機能の置換では、シングルブロック停止がいつ実行されるかはサブプログラムがいつ呼び出されるかによって決まります。 ● ブロック最初で呼び出し:ブロック終点でシングルブロック停止 ● ブロック終点で呼び出し:置換サブプログラムの最後でシングルブロック停止

- ブロック表示で属性 DISPLOF のサブプログラムを置換する場合は、プログラム行が実際のブロックとして表示され、それが呼び出されているサブプログラムとなります。
- 置換サブプログラムでは、各エリアまたは置換サブプログラム全体を、DELAYFSTON と DELAYFSTOF 命令を使用して NC 停止、読み込み禁止などの割り込みから保護することができます。
- 置換が再帰的におこなわれることはありません。つまり、置換サブプログラムを呼び出した機能は、置換サブプログラムで再度プログラム指令されても置換されません。

PLC への補助機能の出力

補助機能を置換する場合は、置換サブプログラムを呼び出してもその補助機能の PLC への出力が開始されません。補助機能が出力されるのは、置換サブプログラムで補助機能が再度プログラム指令された場合に限ります。

ブロック検索中の動作

またこの置換サブプログラムは、ブロック検索モード「計算ありのブロック検索」および「プログラムテストモードにおける計算ありのブロック検索」(SERUPRO)でも呼び出されます。すべての特殊機能は、システム変数$P_SEARCH, $AC_SERUPRO を使用して置換サブプログラムで実装する必要があります。

「計算ありのブロック検索」の収集動作に関しては、置換サブプログラムは「通常の」サブプログラムとまったく同じように動作します。

10.18.5 制限事項

- 以下では、機能の置換はおこなえません。
  - シンクロナイズドアクション
  - テクノロジサイクル
- 置換する機能がブロックの先頭に含まれているブロックの前に、ブロック毎のシンクロナイズドアクションを配置しないでください。 後述の「例: ノンモーダルシンクロナイズドアクション」の節を参照してください。

- 置換サブプログラムでは、個々の置換に必要な動作だけおこなうことができます。
- 置換プログラムがブロックの終点で呼び出されるブロックでは、以下に従ってください。
  - モーダルプログラム呼び出しを有効にしないでください。
  - サブプログラムの復帰ジャンプをプログラム指令しないでください。
  - プログラムの終了をプログラム指令しないでください。

通知
コントローラは、置換する機能が置換サブプログラムで実現されているかどうかを監視しません。

例: ノンモーダルシンクロナイズドアクション

MD30465 $MA_AXIS_LANG_SUB_MASK, ビット 0 = 1 (ギヤ選択切替)

プログラムコード

```
...
N1000 WHENEVER $AA_IM[X2] <= $AA_IM[X1] + 0.5 DO $AA_OVR[X1]=0
N1010 G1 X100 M43
...
```

ブロック N1010 で、機能 M43 によって置換サブプログラムの呼び出しが開始される場合、加工が中断されてアラームが出力されます。

10.19 NC 命令の名前変更/ロック

機能

機械メーカまたはエンドユーザーは、「NC 命令の名前変更/ロック」機能で、既存の NC 命令の名前を変更することができます。

用途

この機能は、以下の目的で使用することができます。

- パートプログラムの読み取りを容易にする
- NC 命令をロックする
- NC 機能をユーザー別に拡張する

パラメータ設定

NC 命令の名前変更/ロックは、以下のマシンデータでおこないます

MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[<n>] (再設定された NC コードのリスト)

偶数のインデックス[<n>]には、命令の元の名前が含まれます。以下の奇数のインデックスには、命令の新しい名前が含まれます。

注記

SINUMERIK 828D の場合は、偶数のインデックス(元の命令名)の入力があらかじめ定義されており、変更することはできません。奇数のインデックス(新しい命令名)のみを書き込むことができます。

新しい名前が空の文字列("")である場合は、命令に、新しい名前はありません。したがって、命令がロックされ、プログラム指令に使用できなくなります。

MD10712 への変更は、電源投入後に有効になります。

例

例 1 (840D sl):命令 G00 の名前を変更し、G01 を解除する

MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[0] = "G00"

MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[1] = "RAPIDTRAVERSE"

MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[2] = "G01"

MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[3] = ""

電源投入で起動した後、高速処理は、G00 命令でプログラム指令することはできなくなり、RAPIDTRAVERSE 命令でのみプログラム指令できます。G01 では、プログラム指令できなくなります。

例 2 (828D):SINUMERIK 828D の事前割り当て値の変更

ISO 系言語モードに適応するために、インデックス[0]～[7]が SINUMERIK 828D に事前に割り当てられており、G505 および G506 命令の名前が、G58 および G59 に変更されています。本来は、G58 および G59 がプログラマブルフレーム、G505 および G506 設定可能フレームでした。名前の変更により、G58 および G59 が設定可能フレームになっています。

この事前割り当てを望んでおらず、G58 と G59 をプログラマブルフレームのままにする場合は、以下のようにして、名前の変更を無効にしてください。

MD10712 のインデックス	命令名	
	予約	新規
[0]	「G58」	「G58」
[1]	**""**	**「G58」**
[2]	「G59」	「G59」
[3]	**""**	**「G59」**
[4]	「G505」	「G505」
[5]	**「G58」**	**「G505」**
[6]	「G506」	「G506」
[7]	**「G59」**	**「G506」**

例 3 (828D):_TCA 命令の名前を TC に変更する

SINUMERIK 828D の TC 命令の名前は、標準で_TCA に変更されます。TCA 機能にはサイクルがあります。このサイクルを使用しない場合は、_TCA 命令の名前を再び TC に変更してください。

MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[8] = "_TCA"

MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[9] = "TCA"

例 4 (828D):既存の命令をユーザー別サイクルに再転送する

既存の WAITM 命令を、ユーザー別機能が使用可能なサイクルに置き換えてください。

1. 既存の WAITM 命令の名前を MD10712 で_WAITM に変更します。
   MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[20] = "WAITM"
   MD10712 $MN_NC_USER_CODE_CONF_NAME_TAB[21] = "_WAITM"
2. WAITM という名前のパートプログラムを作成します。
   PROC WAITM のインターフェースは、NC 機能 WAITM のインターフェースと同じでなければなりません。
   これで、このパートプログラムが既存の WAITM NC 機能に置き換えられます。パートプログラムからのすべての WAITM 命令、および当社サイクルと Shopmill/Shopturn サイクルからの WAITM 命令は、ユーザー別 WAITM サブプログラムを呼び出します。この WAITM サブプログラムでは、ユーザー別機能を実行し、必要に応じて、名前を変更した _WAITM NC 機能を呼び出します。

プログラムコード	コメント
N5 PROC WAITM(<フラグ番号>,<チャネル番号>)	
...	; ユーザー別の機能
N100 _WAITM(<フラグ番号>,<チャネル番号>)	

プログラムコード	コメント
...	; ユーザー別の機能
M30	

10.20 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。

10.20.1 プログラム実行時間

10.20.1.1 機能

「プログラム実行時間」機能には、技術的な処理を監視するさまざまなタイマが搭載されています。この技術的な処理は、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内に読み込むことができます。2 つのタイプのタイマがあります。

1. 標準タイマ
   標準タイマは常に有効です。
2. 起動できるタイマ
   起動できるタイマを、マシンデータで起動してください。

標準タイマ

直近のコントローラ電源投入後の経過時間

システム変数	意味
$AN_SETUP_TIME	制御装置の最後の電源投入後の経過時間です。初期値は分単位です。 初期値でコントローラ電源投入毎に自動的に「0」にリセットされます。
$AN_POWERON_TIME	直近の通常のコントローラ電源投入(「ウォーム再始動」)後の経過時間(分単位)です。 通常のコントローラ電源投入毎に自動的に「0」にリセットされます。

プログラム実行時間

プログラム実行時間を計測するタイマは、自動モードでのみ使用できます。

システム変数(チャネル別)	説明
$AC_ACT_PROG_NET_TIME	現在のプログラムの実際の正味実行時間(単位: 秒) 正味実行時間とは、プログラムが停止した時間が差し引かれたものを指します。 AUTOMATIC 運転モードでパートプログラムがリセットチャネル状態から再起動された場合、$AC_ACT_PROG_NET_TIME が自動的に「0」にリセットされます。 その他プロパティ: • リセットボタンを押しても$AC_ACT_PROG_NET_TIME は「0」に戻らず、タイマが停止するだけです。 • ASUB の起動時に、$AC_ACT_PROG_NET_TIME が「0」に設定され、ASUB の実行時間もカウントされます。ASUB の終了時には、リセットボタンとまったく同じように動作します。タイマは保持されるだけで、「0」には設定されません。 • $AC_ACT_PROG_NET_TIME は、イベント起動プログラム(PROG_EVENT)の起動時にリセット**されません**。 プログラム実行時間は、起動、M30、またはサーチ PROG_EVENT に関係する場合にだけ、さらにカウントされます。 • GOTOS とオーバライド = 0%に対する $AC_ACT_PROG_NET_TIME の動作は、MD27850 を使用してパラメータ設定できます(「パラメータ設定」の章を参照してください)。 **ヒント:** $AC_PROG_NET_TIME_TRIGGER を使用すれば、$AC_ACT_PROG_NET_TIME を更に操作できます。

システム変数(チャネル別)	説明
$AC_OLD_PROG_NET_TIME	まったく正常に終了したプログラムの正味実行時間(単位: 秒) 「正常終了」とは、プログラムが RESET により一時中止されず、代わりに M30 により適切に終了したことを指します。 プログラムが新たに起動する場合、M30 に再度到達するまで、$AC_OLD_PROG_NET_TIME は操作されません。 その他プロパティ: • 、$AC_OLD_PROG_NET_TIME は、現在選択されているプログラムが編集されると、「0」に設定されます。 • $AC_OLD_PROG_NET_TIME は、ASUB またはイベント起動プログラム(PROG_EVENT)の終了時に変更されません。 ヒント: $AC_ACT_PROG_NET_TIME から$AC_OLD_PROG_NET_TIME への自動的なコピー処理が実行されるのは、$AC_PROG_NET_TIME_TRIGGER が書き込まれていない場合に限ります。
$AC_OLD_PROG_NET_TIME_COUNT	$AC_OLD_PROG_NET_TIME への変更 電源投入後、$AC_OLD_PROG_NET_TIME_COUNT は「0」になります。 コントローラが新たに$AC_OLD_PROG_NET_TIME を書き込んだとき、$AC_OLD_PROG_NET_TIME_COUNT は常にカウントアップします。 ユーザーが RESET により実行中のプログラムを終了したとき、$AC_OLD_PROG_NET_TIME と $AC_OLD_PROG_NET_TIME_COUNT は変更されません。 $AC_OLD_PROG_NET_TIME_COUNT を使用すれば、$AC_OLD_PROG_NET_TIME が書き込まれたかどうか判断できます。 例: 連続して実行される 2 つのプログラムの実行時間が同じで、両方とも正常終了した場合、ユーザーは、$AC_OLD_PROG_NET_TIME_COUNT 内の変更された値を使用して、このことを識別できます。

システム変数(チャネル別)	説明	
$AC_PROG_NET_TIME_TRIGGER	プログラム区間の選択計測のトリガ。 変数への値の書き込みによって機能が起動されます。	
	値	**機能**
	0	無効
	1	計測を終了します。 • $AC_OLD_PROG_NET_TIME = $AC_ACT_PROG_NET_TIME • $AC_ACT_PROG_NET_TIME = 0、その後、引き続き実行されます。
	2	計測を開始します。 • $AC_ACT_PROG_NET_TIME = 0 • $AC_OLD_PROG_NET_TIME は変更されません。
	3	計測を停止します。 • $AC_OLD_PROG_NET_TIME は変更されません。 • $AC_ACT_PROG_NET_TIME は変更されません。
	4	停止した計測を続行します。 • $AC_ACT_PROG_NET_TIME が引き続き実行されます。 • $AC_OLD_PROG_NET_TIME は変更されません。
電源を投入すると、システム変数がすべて 0 にリセットされます。		

注記

加工の残り時間

同じワークを順々に機械加工し、次のタイマ値を使用して、ワークの残り時間を決定できます。

- 前に生産されたワークの処理時間($AC_OLD_PROG_NET_TIME を参照してください)
- 現在の処理時間です($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$AC_OLD_PROG_NET_TIME と$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 を明記的に使用してください。

起動できるタイマ

プログラム実行時間

プログラム実行時間を計測するタイマは、自動モードでのみ使用できます。

システム変数(チャネル別)	説明
$AC_OPERATING_TIME	自動モードでの NC プログラムの合計実行時間(単位: 秒) 自動モードで、NC スタートからプログラム終了/リセットまでの全プログラムの実行時間が合計されます。 初期設定では、NC ストップとオーバライド= 0%のときにはカウントされません。オーバライドが 0%のときは、MD27860 で継続カウントを起動できます。 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。
$AC_CYCLE_TIME	選択された NC プログラムの実行時間(単位: 秒) 選択した NC プログラムで、NC スタートからプログラム終了/NC リセットまでの実行時間が測定されます。 初期設定では、NC ストップとオーバライド= 0%のときにはカウントされません。オーバライドが 0%のときは、MD27860 で継続カウントを起動できます。 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。MD27860 は、GOTOS によるプログラム始点へのジャンプがある場合、または ASUB と PROG_EVENT が起動する場合でも、この値を確実に削除するように設定できます。
$AC_CUTTING_TIME	処理時間(秒単位) (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 スタートとプログラム終了/NC リセットの間に計測します。MD27860 を使用すると、測定を工具の動作中にのみ実行するか、工具状態に無関係に実行するかを設定できます。 測定は、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。

10.20.1.2 セットアップ

起動と解除

起動可能なタイマを、マシンデータを使用してオン/オフにできます。

MD27860 $MC_PROCESSTIMER_MODE、ビット 0 - 2 = <値>

ビット	値	意味
0	0	$AC_OPERATING_TIME のタイマは無効
	1	$AC_OPERATING_TIME のタイマは有効
1	0	$AC_CYCLE_TIME のタイマは無効
	1	$AC_CYCLE_TIME のタイマは有効
2	0	$AC_CUTTING_TIME のタイマは無効
	1	$AC_CUTTING_TIME のタイマは有効

パラメータ設定

常に有効になっているタイマの動作

GOTOS とオーバライド = 0%に対して常に有効になっているタイマの動作は、マシンデータを使用して設定します。

MD27850 $MC_PROG_NET_TIMER_MODE

ビット	値	意味
0	0	GOTOS を使用してプログラム先頭にジャンプしても、 $AC_ACT_PROG_NET_TIME は「0」にリセットされません(初期設定)。
	1	GOTOS を使用してプログラム先頭にジャンプすると、 $AC_ACT_PROG_NET_TIME は「0」にリセットされ、値が $AC_OLD_PROG_NET_TIME に保存された後、プログラムカウンタ $AC_OLD_PROG_NET_TIME_COUNT がインクリメントされます。

ビット	値	意味
1	0	オーバライド = 0%の場合、$AC_ACT_PROG_NET_TIME は増分されません。つまり、プログラム実行時間はオーバライドが「0」に設定されていた時間を除いて測定されます(初期設定)。
	1	オーバライド = 0%の場合も、$AC_ACT_PROG_NET_TIME は増分されます。つまり、プログラム実行時間はオーバライドが「0」に設定されていた時間を含めて測定されます。

起動可能なタイマの動作

特定の機能(テストラン送り速度、プログラムテストなど)に対して起動可能なタイマの動作は、次のマシンデータを使用して設定します。

MD27860 $MC_PROCESSTIMER_MODE

ビット	値	意味
4	0	ドライラン送り速度が有効な時は測定しません。
	1	ドライラン送り速度が有効なときに測定します。
5	0	プログラムテスト中に測定しません。
	1	プログラムテスト中に測定します。
6	ビット 1 = 1($AC_CYCLE_TIME のタイマが有効)の場合のみ	
	0	ASUB と PROG_EVENT によるスタートの場合も、$AC_CYCLE_TIME が「0」にリセットされます。
	1	ASUB と PROG_EVENT によるスタートの場合に、$AC_CYCLE_TIME は「0」にリセットされません。
7	ビット 2 = 1($AC_CUTTING_TIME のタイマが有効)の場合のみ	
	0	有効な工具の場合に限り、$AC_CUTTING_TIME のタイマがカウントします。
	1	工具に関係なく、$AC_CUTTING_TIME のタイマがカウントします。
8	ビット 1 = 1($AC_CYCLE_TIME のタイマが有効)の場合のみ	
	0	GOTOS を使用してプログラム先頭にジャンプしても、$AC_CYCLE_TIME は「0」にリセットされません。
	1	GOTOS を使用してプログラム先頭にジャンプすると、$AC_CYCLE_TIME が「0」にリセットされます。

ビット	値	意味
9	ビット 0、1 = 1($AC_OPERATING_TIME と$AC_CYCLE_TIME のタイマが有効)の場合のみ	
	0	オーバライドが 0%のときは、プログラム実行時間のカウントが続行されません。
	1	オーバライドが 0%のとき、プログラム実行時間のカウントが続行されます。

10.20.1.3 必要条件

ブロックサーチ

ブロック検索を使用しても、プログラム実行時間は特定されません。

REPOS

REPOS の処理時間は、現在の処理時間($AC_ACT_PROG_NET_TIME)に加算されます。

10.20.1.4 例

例 1:MD27860 を使用した実行時間測定のパラメータ設定

有効な NC プログラムの実行時間測定の起動、および同時に、ドライラン送り速度が有効およびプログラムテストの場合には測定しません。

MD27860 $MC_PROCESSTIMER_MODE = 'H2'

工具使用時間の測定の起動、および、ドライラン送り速度が有効でプログラムテストの場合も測定します。

MD27860 $MC_PROCESSTIMER_MODE = 'H34'

プログラムテストでの測定を含む、合計実行時間と動作中の工具による処理時間の測定の起動:

MD27860 $MC_PROCESSTIMER_MODE = 'H25'

プログラムテストでの測定を含む、合計実行時間と加工時間(工具には無関係)の測定の起動:

MD27860 $MC_PROCESSTIMER_MODE = 'Ha5'

オーバライド 0%での測定を含むが、ドライラン送り中の測定は含まない、動作中の工具による処理時間の測定の起動:

MD27860 $MC_PROCESSTIMER_MODE = 'H22'

例 2 :「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 からの値は、

- M30 の後も保持されます。
- ループがひととおり実行されるたびに更新されます。

例 3 :「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
```

10.20.2 ワークカウンタ

10.20.2.1 機能

「ワークカウンタ」機能により、チャネル別システム変数の形式で値範囲 0 ～ 999,999,999 の各種カウンタを使用できます。システム変数への読み取りと書き込みアクセスが可能です。

以下のチャネル別マシンデータを使用して、カウンタの起動、カウンタリセットのタイミングとカウントのアルゴリズムを制御できます。

ワークカウントのシステム変数

システム変数	意味
$AC_REQUIRED_PARTS	加工ワーク数(ワーク数の指令値) このカウンタで定義したワーク数に達すると、実ワークカウント数($AC_ACTUAL_PARTS)を「0」にリセットします。 MD27880 を使用すると、表示アラーム 「チャネル%1 ワークターゲット=%2 に達しました」 の表示、およびチャネル別 NC/PLC インタフェース信号: DB21、DBX317.1 (ワークターゲットに到達) の生成を有効にできます。
$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、起動後に加工されたワークの総数を表します。この値は、初期値によるコントローラの起動時にのみ、自動的に「0」にリセットされます。
$AC_ACTUAL_PARTS	完成したワークの数(実際のワーク合計数) このカウンタは、起動後に加工されたワークの総数を記録します。このカウンタは、$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」にリセットされます。
$AC_SPECIAL_PARTS	ユーザーにより選択されたワークの数 このカウンタで、ユーザー専用のワークカウントを指定できます。指令ワーク数の値($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

初期値によるコントローラの起動時に、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。

10.20.2.2 セットアップ

起動

次のマシンデータを使用して、ワークカウンタを起動します。

MD27880 $MC_PART_COUNTER (ワークカウンタの起動)

ビット	値	意味
0	1	$AC_REQUIRED_PARTS が有効になります。
1	0	アラーム/信号の出力: $AC_ACTUAL_PARTS = $AC_REQUIRED_PARTS
	1	アラーム/信号の出力: $AC_SPECIAL_PARTS = $AC_REQUIRED_PARTS
4	1	$AC_TOTAL_PARTS が有効になります。
5	0	$AC_TOTAL_PARTS は、M02 / M30 により値が「1」だけ増加します。
	1	$AC_TOTAL_PARTS は、MD27882[0]で定義された M 命令により、値が「1」だけ増加します。
6	0	プログラムテスト/ブロック検索でも$AC_TOTAL_PARTS は有効になります。
7	1	GOTOS を使用してジャンプして戻ると、 $AC_TOTAL_PARTS は値が「1」だけ増加します。
8	1	$AC_ACTUAL_PARTS は有効になります。
9	0	$AC_ACTUAL_PARTS は、M02 / M30 により値が「1」だけ増加します。
	1	$AC_ACTUAL_PARTS は、MD27882[1]で定義された M 命令により、値が「1」だけ増加します。
10	0	プログラムテスト/ブロック検索でも$AC_ACTUAL_PARTS は有効になります。
11	1	GOTOS を使用してジャンプして戻ると、 $AC_ACTUAL_PARTS は値が「1」だけ増加します。
12	1	$AC_SPECIAL_PARTS が有効になります。

ビット	値	意味
13	0	$AC_SPECIAL_PARTS は、M02 / M30 により値が「1」だけ増加します。
	1	$AC_SPECIAL_PARTS は、MD27882[2]で定義された M 命令により、値が「1」だけ増加します。
14	0	プログラムテスト/ブロック検索でも$AC_SPECIAL_PARTS は有効になります。
15	1	GOTOS を使用してジャンプして戻ると、 $AC_SPECIAL_PARTS は値が「1」だけ増加します。

パラメータの割り付け

ユーザー定義 M 命令によるワークのカウント

対応するビットを MD27880 で設定している場合は、プログラム終了 M2/M30 ではなく、次のマシンデータにより設定したユーザー定義の M 命令でカウンタパルスを起動できます。

MD27882 $MC_PART_COUNTER_MCODE[<n>] (ユーザー定義の M 命令によるワークのカウント)

<n>	意味
0	MD27882[0]では、$AC_TOTAL_PARTS を増分する M 命令が定義されます。
1	MD27882[1]では、$AC_ACTUAL_PARTS を増分する M 命令が定義されます。
2	MD27882[2]では、$AC_SPECIAL_PARTS を増分する M 命令が定義されます。

ユーザー定義 M 命令が呼び出されると、各ワークカウンタは「1」だけ増加します。

ワークカウントの保護レベル

次のマシンデータを使用して、ユーザーインタフェースでワークカウントを有効/無効にする保護レベルを設定します。

MD51074 $MN_ACCESS_WRITE_WPC_COUNTER = <保護レベル>

10.20.2.3 必要条件

モード変更/NC リセット

各カウンタは、モード変更または NC RESET の影響を受けません。

$AC_REQUIRED_PARTS ≤ 0

$AC_REQUIRED_PARTS ≤ 0 と MD27880 $MC_PART_COUNTER ビット 0 = 1 の場合は、カウント処理、および MD27880 で設定した識別番号比較が、すべてのカウンタに対して**実行されません**。

10.20.2.4 例

ワークカウンタ$AC_REQUIRED_PARTS の起動:

- MD27880 $MC_PART_COUNTER = 'H3'

$AC_REQUIRED_PARTS が有効になります。

表示アラーム: $AC_REQUIRED_PARTS == $AC_SPECIAL_PARTS

ワークカウンタ$AC_TOTAL_PARTS の起動:

- MD27880 $MC_PART_COUNTER = 'H10'
- MD27882 $MC_PART_COUNTER_MCODE[0] = 80

$AC_TOTAL_PARTS が有効です。M02 毎に、カウンタの値が 1 ずつ増加します。

$MC_PART_COUNTER_MCODE[0]は無効です。

ワークカウンタ$AC_ACTUAL_PARTS の起動:

- MD27880 $MC_PART_COUNTER = 'H300'
- MD27882 $MC_PART_COUNTER_MCODE[1] = 17

$AC_TOTAL_PARTS が有効です。M17 毎に、カウンタの値が 1 ずつ増加します。

ワークカウンタ$AC_SPECIAL_PARTS の起動:

- MD27880 $MC_PART_COUNTER = 'H3000'
- MD27882 $MC_PART_COUNTER_MCODE[2] = 77

$AC_SPECIAL_PARTS が有効になります。

各 M77 により、次のことが行われます: $AC_SPECIAL_PARTS + 1

ワークカウンタ$AC_ACTUAL_PARTS の解除:

- MD27880 $MC_PART_COUNTER = 'H200'
- MD27882 $MC_PART_COUNTER_MCODE[1] = 50

$AC_ACTUAL_PARTS は無効になります。

すべてのカウンタの起動:

- MD27880 $MC_PART_COUNTER = 'H3313'
- MD27882 $MC_PART_COUNTER_MCODE[0] = 80
- MD27882 $MC_PART_COUNTER_MCODE[1] = 17
- MD27882 $MC_PART_COUNTER_MCODE[2] = 77

$AC_REQUIRED_PARTS が有効になります。

表示アラーム: $AC_REQUIRED_PARTS == $AC_SPECIAL_PARTS

$AC_TOTAL_PARTS が有効です。M02 毎に、カウンタの値が 1 ずつ増加します。

$MC_PART_COUNTER_MCODE[0]は無効です。

$AC_ACTUAL_PARTS が有効です。M17 毎に、カウンタの値が 1 ずつ増加します。

$AC_SPECIAL_PARTS が有効です。M77 毎に、カウンタの値が 1 ずつ増加します。

プログラムテスト/ブロック検索時にワークカウンタ$AC_ACTUAL_PARTS を処理しない:

- MD27880 $MC_PART_COUNTER = 'H700'
- MD27882 $MC_PART_COUNTER_MCODE[1] = 75

$AC_ACTUAL_PARTS が有効になります。プログラムテストと検索時を除き、M75 毎に、カウンタの値が「1」ずつ増加します。

MD27880 $MC_PART_COUNTER のビット 0 = 1 のときにカウントモードを取り消す:

- MD27882 $MC_PART_COUNTER_MCODE[0] = 41
- MD27882 $MC_PART_COUNTER_MCODE[1] = 42
- MD27882 $MC_PART_COUNTER_MCODE[2] = 43

プログラムコード	コメント
...	
N100 $AC_REQUIRED_PARTS=-10	; 値 <0:カウント設定。
N200 M41 M43	; カウントなし。
N300 M42	
...	
N500 $AC_REQUIRED_PARTS=52	; 値 > 0:MD27880 に応じたカウントを起動します。
N501 M43	; カウントします。
N502 M42 M41	; カウントします。
...	

10.21 データリスト

10.21.1 機能

「ワークカウンタ」機能により、チャネル別システム変数の形式で値範囲 0 ～ 999,999,999 の各種カウンタを使用できます。システム変数への読み取りと書き込みアクセスが可能です。

以下のチャネル別マシンデータを使用して、カウンタの起動、カウンタリセットのタイミングとカウントのアルゴリズムを制御できます。

ワークカウントのシステム変数

システム変数	意味
$AC_REQUIRED_PARTS	加工ワーク数(ワーク数の指令値) このカウンタで定義したワーク数に達すると、実ワークカウント数($AC_ACTUAL_PARTS)を「0」にリセットします。 MD27880 を使用すると、表示アラーム 「チャネル%1 ワークターゲット=%2 に達しました」 の表示、およびチャネル別 NC/PLC インタフェース信号: DB21、DBX317.1 (ワークターゲットに到達) の生成を有効にできます。
$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、起動後に加工されたワークの総数を表します。この値は、初期値によるコントローラの起動時にのみ、自動的に「0」にリセットされます。
$AC_ACTUAL_PARTS	完成したワークの数(実際のワーク合計数) このカウンタは、起動後に加工されたワークの総数を記録します。このカウンタは、$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」にリセットされます。
$AC_SPECIAL_PARTS	ユーザーにより選択されたワークの数 このカウンタで、ユーザー専用のワークカウントを指定できます。指令ワーク数の値($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

初期値によるコントローラの起動時に、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。

10.21.2 マシンデータ

10.21.2.1 一般マシンデータ

表示マシンデータ

番号	識別子: $MM_	説明
SINUMERIK Operate		
9421	MA_AXES_SHOW_GEO_FIRST	チャネルのジオメトリ軸を最初に表示します
9422	MA_PRESET_MODE	JOG における基本オフセット/ PRESET。
9423	MA_MAX_SKP_LEVEL	最大スキップレベル数

NC 別マシンデータ

番号	識別子: $MN_	説明
10010	ASSIGN_CHAN_TO_MODE_GROUP	モードグループ内で有効なチャネル
10125	EES_NC_NAME	EES モードで一義的な NC プログラム名を生成するための NCU 名
10280	PROG_FUNCTION_MASK	比較演算命令「>」および「<」
10700	PREPROCESSING_LEVEL	NC プログラム事前処理機能レベル
10702	IGNORE_SINGLEBLOCK_MASK	シングルブロック停止の防止
10707	PROG_TEST_MASK	プログラムテストモード
10708	SERUPRO_MASK	ブロック変更モード
10710	PROG_SD_RESET_SAVE_TAB	更新するセッティングデータ
10711	NC_LANGUAGE_CONFIGURATION	関連のオプションまたは機能が起動されていない言語の処理方法

番号	識別子: $MN_	説明
10713	M_NO_FCT_STOPRE	先読み停止をともなう M 機能
10715	M_NO_FCT_CYCLE	サブプログラムにより置換される M 機能
10716	M_NO_FCT_CYCLE_NAME	M 機能置換用のサブプログラム名称
10717	T_NO_FCT_CYCLE_NAME	T 機能置換に対応した工具交換サイクル名称
10718	M_NO_FCT_CYCLE_PAR	パラメータ付き M 機能置換
10719	T_NO_FCT_CYCLE_MODE	T 機能置換のパラメータ割り当て
11450	SEARCH_RUN_MODE	ブロックサーチパラメータ設定
11470	REPOS_MODE_MASK	再位置決め特性
11600	BAG_MASK	ASUB に対するモードグループ応答
11602	ASUP_START_MASK	ASUB の停止条件を無視
11604	ASUP_START_PRIO_LEVEL	優先度、$MN_ASUP_START_MASK が有効となる開始
11610	ASUP_EDITABLE	ユーザー用 ASUB プログラムの起動
11612	ASUP_EDIT_PROTECTION_LEVEL	ユーザー用 ASUB プログラムの保護レベル
11620	PROG_EVENT_NAME	PROG-EVENT のプログラム名称
11625	FILE_ONLY_WITH_EXTENSION	プログラムを呼び出すと、ファイル ID を持つファイルのみを検索
11626	CYCLES_ONLY_IN_CYCDIR	サイクルフォルダのインターフェースを備えたプログラムのみの検索
11717	D_NO_FCT_CYCLE_NAME	D 機能置換用のサブプログラム名称
15700	LANG_SUB_NAME	置換サブプログラムの名称
15702	LANG_SUB_PATH	置換サブプログラムの呼び出しパス
17200	GMMC_INFO_NO_UNIT	グローバル HMI 情報(物理単位なし)
17201	GMMC_INFO_NO_UNIT_STATUS	グローバル HMI 状態情報(物理単位なし)
18045	EES_MODE_INFO	EES 機能が動作するモード
18360	MM_EXT_PROG_BUFFER_SIZE	外部ソース(DRAM)からの実行用 FIFO バッファサイズ
18362	MM_EXT_PROG_NUM	外部から同時に処理できるプログラムレベルの数

10.21.2.2 チャネルマシンデータ

基本マシンデータ

番号	識別子: $MC_	説明
20000	CHAN_NAME	チャネル名称
20050	AXCONF_GEOAX_ASSIGN_TAB	ジオメトリ軸のチャネル軸への割り当て
20060	AXCONF_GEOAX_NAME_TAB	チャネル内の軸名称
20070	AXCONF_MACHAX_USED	チャネルで有効な機械軸番号
20080	AXCONF_CHANAX_NAME_TAB	チャネル内のチャネル軸名称[チャネル軸番号]: 0...7
20090	SPIND_DEF_MASTER_SPIND	チャネル内のメイン主軸の初期設定
20100	DIAMETER_AX_DEF	径方向軸機能が指定されたジオメトリ軸
20106	PROG_EVENT_IGN_SINGLEBLOCK	プログラムイベントがシングルブロックを無視します
20107	PROG_EVENT_IGN_INHIBIT	プログラムイベントが読み込み停止を無視します
20108	PROG_EVENT_MASK	イベント起動プログラム呼び出し
20109	PROG_EVENT_MASK_PROPERTIES	プログラムイベント特性
20114	MODESWITCH_MASK	REPOS の設定
20116	IGNORE_INHIBIT_ASUP	読み込み停止とは関係なく、ユーザー ASUP をすべて実行します
20117	IGNORE_SINGLEBLOCK_ASUP	シングルブロック処理とは関係なく、ユーザー ASUP をすべて処理します
20160	CUBIC_SPLINE_BLOCKS	C スプライン用ブロック数
20170	COMPRESS_BLOCK_PATH_LIMIT	圧縮のための NC ブロックの最大移動距離
20191	IGN_PROG_STATE_ASUP	操作パネル上に割り込みルーチンの実行を表示しません。
20192	PROG_EVENT_IGN_PROG_STATE	操作パネル上にプログラムイベントの実行を表示しません。
20193	PROG_EVENT_IGN_STOP	プログラムイベントが STOP キーを無視します

番号	識別子: $MC_	説明
20194	IGNORE_NONCSTART_ASUP	特定のユーザーアラームのアラーム応答「NC スタート禁止」が未処置でも ASUP 起動を許容します
20210	CUTCOM_CORNER_LIMIT	工具径補正の交点演算の最大角度
20220	CUTCOM_MAX_DISC	DISC の最大値
20230	CUTCOM_CURVE_INSERT_LIMIT	工具径補正の交点演算の最大角度
20240	CUTCOM_MAXNUM_CHECK_BLOCKS	工具径補正の予測輪郭演算のブロック
20250	CUTCOM_MAXNUM_DUMMY_BLOCKS	TRC に対する移動動作なしのブロックの最大数
20270	CUTTING_EDGE_DEFAULT	工具刃先をプログラム指令しないときの初期設定
20400	LOOKAH_USE_VELO_NEXT_BLOCK	プログラム指令された次のブロック速度の先読み
20430	LOOKAH_NUM_OVR_POINTS	先読みに対応したオーバライドポイント数
20440	LOOKAH_OVR_POINTS	先読みのオーバライドスイッチポイント
20500	CONST_VELO_MIN_TIME	定速移動の最短時間
20600	MAX_PATH_JERK	軌跡関連の最高加々速度
20610	ADD_MOVE_ACCEL_RESERVE	重畳動作に対応した加速度予備
20700	REFP_NC_START_LOCK	レファレンス点なしで NC スタート無効
20750	ALLOW_GO_IN_G96	G96、G961 の G0 ロジック
20800	SPF_END_TO_VDI	PLC へのサブプログラム終了
21000	CIRCLE_ERROR_CONST	円弧終点監視定数
21010	CIRCLE_ERROR_FACTOR	円弧終点監視係数
21100	ORIENTATION_IS_EULER	旋回プログラミングに対する角度定義
21110	X_AXIS_IN_OLD_X_Z_PLANE	自動フレーム定義用座標系
21200	LIFTFAST_DIST	輪郭からの高速リトラクトの移動距離
21210	SETINT_ASSIGN_FASTIN	NC プログラム割り込みに対応した外部 NC 入力バイトのハードウェア割り当て:
21202	LIFTFAST_WITH_MIRROR	ミラーリングのある高速リトラクト

ブロックサーチ

番号	識別子: $MC_	説明
20128	COLLECT_TOOL_CHANGE	ブロックサーチ後の PLC への工具交換命令
22600	SERUPRO_SPEED_MODE	ブロックサーチタイプ 5 の速度
22601	SERUPRO_SPEED_FACTOR	ブロックサーチタイプ 5 の速度係数
22621	ENABLE_START_MODE_MASK_PRT	SERUPRO 用 MD22620:START_MODE_MASK_ PRT、SERUPRO ブロック検索実行用に解放
22622	DISABLE_PLC_START	PLC によるパートプログラム起動の許可
22680	AUTO_IPTR_LOCK	割り込みポインタの解除

リセット応答

番号	識別子:$MC_	説明
20110	RESET_MODE_MASK	リセット/パートプログラム終了後の初期設定
20112	START_MODE_MASK	パートプログラム起動後の NC スタートの初期設定
20118	GEOAX_CHANGE_RESET	自動ジオメトリ軸変更可能
20120	TOOL_RESET_VALUE	電源投入(リセット/パートプログラム終了)時の工具長補正
20121	TOOL_PRESEL_RESET_VALUE	リセット時に事前選択される工具
20130	CUTTING_EDGE_RESET_VALUE	電源投入(リセット/パートプログラム終了)時の工具刃先長補正
20140	TRAFO_RESET_VALUE	電源投入(リセット/パートプログラム終了)時の座標変換データセット
20150	GCODE_RESET_VALUES	G グループの初期設定
20152	GCODE_RESET_MODE	G グループのリセット応答
20156	MAXNUM_GCODES_EXT	外部 G グループのリセット応答
22620	START_MODE_MASK_PRT	特定 NC スタート時の初期設定

補助機能の設定

番号	識別子: $MC_	説明
22000	AUXFU_ASSIGN_GROUP	補助機能グループ
22010	AUXFU_ASSIGN_TYPE	補助機能のタイプ
22020	AUXFU_ASSIGN_EXTENSION	補助機能拡張
22030	AUXFU_ASSIGN_VALUE	補助機能値
22200	AUXFU_M_SYNC_TYPE	M 機能の出力タイミング
22210	AUXFU_S_SYNC_TYPE	S 機能の出力タイミング
22220	AUXFU_T_SYNC_TYPE	T 機能の出力タイミング
22230	AUXFU_H_SYNC_TYPE	H 機能の出力タイミング
22240	AUXFU_F_SYNC_TYPE	F 機能の出力タイミング
22250	AUXFU_D_SYNC_TYPE	D 機能の出力タイミング
22400	S_VALUES_ACTIVE_AFTER_RESET	リセット後に S 機能有効
22410	F_VALUES_ACTIVE_AFTER_RESET	リセット後の F 機能有効
22510	GCODE_GROUPS_TO_PLC	ブロック変更/リセット時に NC/PLC インタフェースに出力される G 命令
22550	TOOL_CHANGE_MODE	M 機能の新しい工具オフセット
22560	TOOL_CHANGE_M_CODE	工具交換の M 機能

メモリ設定

番号	識別子:$MC_	説明
27900	REORG_LOG_LIMIT	有効なログファイル用 IPO バッファの比率
28000	MM_REORG_LOG_FILE_MEM	REORG のメモリサイズ(DRAM)
28010	MM_NUM_REORG_LUD_MODULES	REORG のローカルユーザー変数に対応したブロックの数
28020	MM_NUM_LUD_NAMES_TOTAL	ローカルユーザー変数の数(DRAM)
28040	MM_LUD_VALUES_MEM	ローカルユーザー変数のメモリサイズ(DRAM)
28050	MM_NUM_R_PARAM	チャネル別 R 変数の数(SRAM)
28060	MM_IPO_BUFFER_SIZE	IPO バッファ内の NC ブロックの数(DRAM)
28070	MM_NUM_BLOCKS_IN_PREP	ブロック解析のためのブロックの数(DRAM)
28080	MM_NUM_USER_FRAMES	設定可能フレームの数(SRAM)

番号	識別子:$MC_	説明
28090	MM_NUM_CC_BLOCK_ELEMENTS	コンパイルサイクル用ブロック要素の数 (DRAM)
28100	MM_NUM_CC_BLOCK_USER_MEM	コンパイルサイクル用ブロックメモリのサイズ (DRAM)
28400	MM_ABSBLOCK	絶対値による基本ブロックの起動
28402	MM_ABSBLOCK_BUFFER_CONF	アップロードバッファの寸法サイズ
28500	MM_PREP_TASK_STACK_SIZE	解析処理のスタックサイズ(DRAM)

プログラム実行時間およびワークカウンタ

番号	識別子: $MC_	説明
27860	PROCESSTIMER_MODE	実行時間測定の起動
27880	PART_COUNTER	ワークカウンタの起動
27882	PART_COUNTER_MCODE[ ]	M 命令を介したワークカウント

10.21.2.3 軸/主軸マシンデータ

番号	識別子: $MA_	説明
30465	AXIS_LANG_SUB_MASK	NC 言語命令の置換
30550	AXCONF_ASSIGN_MASTER_CHAN	軸入れ替え用のチャネルのリセット設定
30600	FIX_POINT_POS	G75 による軸の固定値点
33100	COMPRESS_POS_TOL	コンプレッサによる最大偏差

10.21.3 セッティングデータ

10.21.3.1 チャネルセッティングデータ

番号	識別子: $SC_	説明
42000	THREAD_START_ANGLE	ねじの開始角度
42010	THREAD_RAMP_DISP	ねじ切り時の軸の加速動作
42100	DRY_RUN_FEED	ドライラン送り速度
42200	SINGLEBLOCK2_STOPRE	SBL2 のデバッグモードの起動
42444	TARGET_BLOCK_INCR_PROG	計算ありのブロックサーチ後の継続モード
42700	EXT_PROG_PATH	外部サブプログラム呼び出し EXTCALL のプログラムパス
42750	ABSBLOCK_ENABLE	基本ブロック表示の起動
42990	MAX_BLOCKS_IN_IPOBUFFER	補間バッファ内のブロックの最大数

10.21.4 信号

10.21.4.1 NC への信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
非常停止	DB10.DBX56.1	DB2600.DBX0.1

10.21.4.2 モードグループの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
AUTOMATIC モード	DB11.DBX0.0	DB3000.DBX0.0
MDI モード	DB11.DBX0.1	DB3000.DBX0.1
JOG モード	DB11.DBX0.2	DB3000.DBX0.2
モード変更無効	DB11.DBX0.4	DB3000.DBX0.4
モードグループ停止	DB11.DBX0.5	DB3000.DBX0.5
モードグループ停止、軸と主軸	DB11.DBX0.6	DB3000.DBX0.6

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
モードグループリセット	DB11.DBX0.7	DB3000.DBX0.7
運転機能ティーチング	DB11.DBX1.0	DB3000.DBX1.0
運転機能 REPOS	DB11.DBX1.1	-
運転機能 REF	DB11.DBX1.2	DB3000.DBX1.2

10.21.4.3 NC への信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
AUTOMATIC モード選択	DB11.DBX4.0	-
MDI モード選択	DB11.DBX4.1	-
JOG モード選択	DB11.DBX4.2	-
運転機能ティーチング選択	DB11.DBX5.0	-
運転機能 REPOS 選択	DB11.DBX5.1	-
運転機能 REF 選択	DB11.DBX5.2	-
アクティブモード AUTOMATIC	DB11.DBX6.0	DB3100.DBX0.0
有効モード MDI	DB11.DBX6.1	DB3100.DBX0.1
アクティブモード JOG	DB11.DBX6.2	DB3100.DBX0.2
モードグループ準備完了	DB11.DBX6.3	DB3100.DBX0.3
モードグループリセット実行済み	DB11.DBX6.4	DB3100.DBX0.4
NC 内部 JOG 有効	DB11.DBX6.5	-
すべてのチャネルがリセット状態	DB11.DBX6.7	DB3100.DBX0.7
運転機能ティーチング有効	DB11.DBX7.0	DB3100.DBX1.0
運転機能 REPOS 有効	DB11.DBX7.1	-
運転機能 REF 有効	DB11.DBX7.2	DB3100.DBX1.2

10.21.4.4 チャネルへの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
DRF の起動	DB21、... .DBX0.3	DB320x.DBX0.3
シングルブロックの起動	DB21、... .DBX0.4	DB320x.DBX0.4

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
M01 起動	DB21、... .DBX0.5	DB320x.DBX0.5
ドライラン送り速度起動	DB21、... .DBX0.6	DB320x.DBX0.6
PLC 動作完了	DB21、... .DBX1.6	-
プログラムテスト起動	DB21、... .DBX1.7	DB320x.DBX1.7
ブロックスキップレベル: /0 ～/7	DB21、... .DBX2.0-7	DB320x.DBX2.0-7
読み込み停止	DB21、... .DBX6.1	DB320x.DBX6.1
プログラムレベル中止	DB21、... .DBX6.4	DB320x.DBX6.4
NC スタート禁止	DB21、... .DBX7.0	DB320x.DBX7.0
NC スタート	DB21、... .DBX7.1	DB320x.DBX7.1
ブロック境界での NC ストップ	DB21、... .DBX7.2	DB320x.DBX7.2
NC ストップ	DB21、... .DBX7.3	DB320x.DBX7.3
NC ストップ、軸と主軸	DB21、... .DBX7.4	DB320x.DBX7.4
リセット	DB21、... .DBX7.7	DB330x.DBX3.7
REPOSPATHMODE	DB21、... .DBX31.0-2	-
REPOSMODEEDGE	DB21、... .DBX31.4	-

10.21.4.5 チャネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
DRF 選択中	DB21、... .DBX24.3	DB170x.DBX0.3
NC に関連する M01 の選択	DB21、... .DBX24.4	-
M01 選択中	DB21、... .DBX24.5	DB170x.DBX0.5
ドライラン送り速度選択	DB21、... .DBX24.6	DB170x.DBX0.6
REPOSPATHMODE 0 - 2	DB21、... .DBX25.0-2	-
早送り速度オーバーライド選択	DB21、... .DBX25.3	DB170x.DBX1.3
REPOS MODE EDGE	DB21、... .DBX25.4	-
プログラムテスト選択中	DB21、... .DBX25.7	DB170x.DBX1.7
ブロックスキップ選択中: /0 ～/7	DB21、... .DBX26.0-7	DB170x.DBX2.0-7
ブロックスキップ選択中: /8	DB21、... .DBX27.0	DB170x.DBX3.0
ブロックスキップ選択中: /9	DB21、... .DBX27.1	DB170x.DBX3.1

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
REPOSPATHMODE 0 - 2	DB21、... .DBX31.0-2	-
REPOS MODE EDGE	DB21、... .DBX31.4	-
ブロックスキップ有効 /8	DB21、... .DBX31.6	DB320x.DBX15.6
ブロックスキップ有効 /9	DB21、... .DBX31.7	DB320x.DBX15.7
外部からの実行有効	DB21、... .DBX32.0	DB330x.DBX0.0
アクションブロック有効	DB21、... .DBX32.3	DB330x.DBX0.3
アプローチブロック有効	DB21、... .DBX32.4	DB330x.DBX0.4
M0/M1 有効	DB21、... .DBX32.5	DB330x.DBX0.5
最後のアクションブロック有効	DB21、... .DBX32.6	DB330x.DBX0.6
ブロック検索有効	DB21、... .DBX33.4	DB330x.DBX1.4
M02/M30 有効	DB21、... .DBX33.5	DB330x.DBX1.5
座標変換有効	DB21、... .DBX33.6	DB330x.DBX1.6
プログラムテスト有効	DB21、... .DBX33.7	DB330x.DBX1.7
プログラム状態:運転中	DB21、... .DBX35.0	DB330x.DBX3.0
プログラム状態:待機	DB21、... .DBX35.1	DB330x.DBX3.1
プログラム状態:停止中	DB21、... .DBX35.2	DB330x.DBX3.2
プログラム状態:中断中	DB21、... .DBX35.3	DB330x.DBX3.3
プログラム状態:中止	DB21、... .DBX35.4	DB330x.DBX3.4
チャネル状態:有効	DB21、... .DBX35.5	DB330x.DBX3.5
チャネル状態:中断中	DB21、... .DBX35.6	DB330x.DBX3.6
チャネル状態:リセット	DB21、... .DBX35.7	DB330x.DBX3.7
割り込み処理有効	DB21、... .DBX36.4	-
チャネル準備完了	DB21、... .DBX36.5	DB330x.DBX4.5
読み込み停止無視	DB21、... .DBX37.6	-
SBL によるブロックエンドでの停止を無視	DB21、... .DBX37.7	-
G グループ 1 ～ n (8 ビット整数)の有効な G 命令の数	DB21、... .DBB208-271	DB350x.DBB0-63
ワーク指令値に達しました	DB21、... .DBX317.1	DB330x.DBX4001.1
ASUB 停止	DB21、... .DBX318.0	DB330x.DBX4002.0
プログラムテストを介したブロック検索が有効	DB21、... .DBX318.1	-
REPOS MODE EDGEACKN	DB21、... .DBX319.0	-

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
再位置決めモード応答:0 - 2	DB21、... .DBX319.1-3	-
チャネルの再位置決め DEFERAL	DB21、... .DBX319.5	-
イベント起動プログラム呼び出しの場合の起動イベントの表示	DB21、... .DBX376.0-7	DB330x.DBB4004
ASUB 有効	DB21、... .DBX378.0	DB330x.DBB4006.0
表示更新がマスクされた ASUB が有効	DB21、... .DBX378.1	DB330x.DBB4006.1

10.21.4.6 NC への信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
REPOSDELAY	DB31、... .DBX10.0	-

10.21.4.7 軸/主軸からの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
REPOS オフセット	DB31、... .DBX70.0	-
REPOS オフセット有効	DB31、... .DBX70.1	-
REPOS 遅延確認	DB31、... .DBX70.2	-
REPOSDELAY	DB31、... .DBX72.0	-
軌跡軸	DB31、... .DBX76.4	DB390x.DBX1002.4

K2: 軸タイプ、座標系、フレーム 11

11.1 概略説明

11.1.1 軸

機械軸

機械軸は、工作機械上に実際に存在する軸です。

チャネル軸

すべてのジオメトリ軸とすべての付加軸がチャネル、したがってチャネル軸に割り当てられます。 ジオメトリ軸と付加軸は、常に「自身の」チャネルで移動します。

ジオメトリ軸

3 つのジオメトリ軸は、常に理論上の直交座標系、すなわち基本座標系(BCS)を作成します。

FRAME (オフセット、座標回転、スケーリング、ミラーリング)を使用することにより、ワーク座標系(WCS)のジオメトリ軸を BCS に投影することができます。

付加軸

ジオメトリ軸とは対照的に、付加軸間で幾何学的な関係は定義されません。

軌跡軸

軌跡軸は、互いに補間されます(チャネルのすべての軌跡軸には共通の軌跡補間器があります)。

1 つのチャネルのすべての軌跡軸には、同一の加速区間、定速移動区間および減速区間があります。

位置決め軸

位置決め軸は、別々に補間されます(位置決め軸毎に独自の軸補間器があります)。 位置決め軸毎に、独自の送り速度特性と加速特性があります。

同期軸

同期軸は軌跡軸とともに補間されます(1 つのチャネルのすべての軌跡軸と同期軸には共通の軌跡補間器があります)。

チャネルのすべての軌跡軸とすべての同期軸には、同一の加速区間、定速移動区間および減速区間があります。

軸設定

下のマシンデータは、ジオメトリ軸、付加軸、チャネル軸および機械軸、更に各軸タイプの名称の割り当てに使用されます。

MD20050 $MC_AXCONF_GEOAX_ASIGN_TAB (ジオメトリ軸のチャネル軸への割り当て)

MD20060 $MC_AXCONF_GEOAX_NAME_TAB (チャネル内でのジオメトリ軸の名称)

MD20070 $MC_AXCONF_MACHAX_USED (チャネルで有効な機械軸番号)

MD20080 $MC_AXCONF_CHANAX_NAME_TAB (チャネルのチャネル軸名称)

MD10000 $MN_AXCONF_MACHAX_NAME_TAB (機械軸名称)

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX (機械軸への主軸の割り当て)

置換可能なジオメトリ軸

「置換可能なジオメトリ軸」機能により、グループ中の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。

最初に同期付加軸としてチャネルに設定される軸は、プログラム命令に応じて、任意の選択されたジオメトリ軸を置き換えることができます。

リンク軸

リンク軸は、別の NCU に物理的に接続されていますが、現在の NCU から位置が制御される軸です。 リンク軸は、別の NCU のチャネルにダイナミックに割り当てることができます。 リンク軸は、特定の NCU の観点からするとローカル軸ではありません。

The **軸コンテナ**の概念は、NCU への割り当てのダイナミックな変更のために使用されます。 パートプログラムからの GET と RELEASE による軸入れ替えは、NCU の境界を越えてリンク軸に使用することはできません。

リンク軸の説明は、次を参照してください。
参照先:
『機能マニュアル、上級機能』；「NCU と操作パネルの n:m 接続(B3)」

軸コンテナ

軸コンテナはリングバッファデータ構造であり、ここでローカル軸とリンク軸のいずれか、または両方がチャネルに割り当てられます。 リングバッファの入力値は、**周期的にシフト**できます。

ローカル軸またはリンク軸への直接参照に加えて、論理機械軸イメージでのリンク軸設定により、軸コンテナへの参照も可能です。

このタイプの参照は、以下の項目から構成されます。

- 軸コンテナ番号
- スロット(対応するコンテナ内のリングバッファロケーション)

リングバッファロケーションの入力には、以下が含まれます。

- ローカル軸
  または
- リンク軸

軸コンテナの機能の説明は、次を参照してください。
参照先:
『機能マニュアル、上級機能』；「NCU と操作パネルの n:m 接続(B3)」

11.1.2 座標系

MCS

機械座標系(MCS)には、以下の特性があります。

- 機械軸により定義されます。
- 機械軸は、直交座標系を形成するように互いに垂直にすること、または、その他の方法で配置することができます。

- 機械軸の名前は定義することができます。
- 機械軸は直線軸または回転軸にすることができます。

BCS

基本座標系(BCS)には、以下の特性があります。

- ジオメトリ軸は直交するデカルト座標系を形成します。
- BCS は MCS のキネマティックトランスフォーメーションにより導出されます。

BZS

基本ゼロオフセットシステム(BZS)は、基本オフセットを使用した基本座標系です。

SZS

設定可能ゼロオフセットシステム(SZS)は、WCS の観点からプログラミング可能なフレームを使用するワーク座標系です。ワーク原点は、G54 から G599 の設定可能フレームにより定義されます。

WCS

ワーク座標系(WCS)には、以下の特性があります。

- ワーク座標系では、すべての軸の座標がプログラミングされます(パートプログラム)。
- ジオメトリ軸および付加軸で構成されています。
- ジオメトリ軸は常に直交するデカルト座標系を形成します。
- 付加軸は付加軸の間で幾何学的な関係のない座標系を形成します。
- ジオメトリ軸および付加軸の名前は定義することができます。
- ワーク座標系は、FRAMES (TRANS、ROT、SCALE、MIRROR)により、平行移動、回転、スケーリングおよびミラーリングすることができます。
  複数の平行移動動作、回転動作なども可能です。

外部ゼロオフセット

外部ゼロオフセットには、以下の特性があります。

- PLC で指定された時に、基本座標系とワーク座標系の間で事前定義された追加の原点オフセットが起動されます。
- オフセットの大きさは、関連する軸のそれぞれに対して以下により設定できます。
  - PLC
  - 操作パネル
  - パートプログラム
- 起動されたオフセットは、オフセット起動後で関連する軸の最初の動作ブロックが処理される時点で有効になります。オフセットは、プログラミングされた軌跡に重畳されます(補間なし)。
  外部ゼロオフセットが適用される速度は、以下のとおりです。
  プログラミングされた F 値 + 1/2 JOG 速度
  外部ゼロオフセットは、G0 ブロックの終点で移動されます。
- 起動されたオフセットは、RESET およびプログラム終了の後も維持されます。
- POWER ON の後、前回の有効オフセットはまだコントロールシステムに記憶されていますが、PLC により再度起動してください。

11.1.3 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一連の演算規則(アルゴリズム)です。

フレーム成分

フレームは、以下の成分から構成されます。

フレーム成分		プログラマブル
オフセット	荒削りオフセット	TRANS ATRANS(付加的な平行移動成分) CTRANS (複数の軸のゼロオフセット) G58 (軸のゼロオフセット)
	仕上げオフセット	CFINE G59 (軸のゼロオフセット)

フレーム成分	プログラマブル
座標回転	ROT/ROTS AROT/AROTS CROTS
スケーリング	SCALE ASCALE
ミラーリング	MIRROR AMIRROR

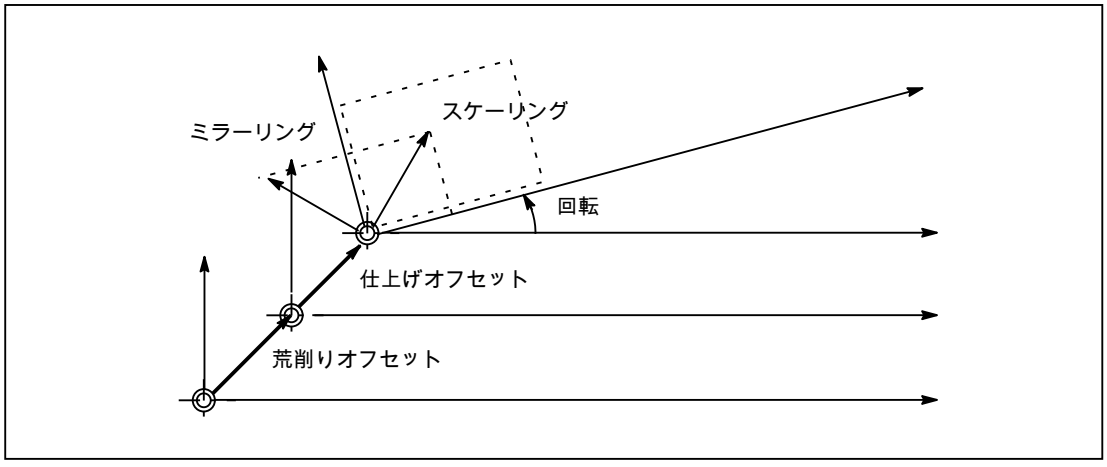


図 11-1　フレーム成分

荒削りオフセットと仕上げオフセット

チャネル軸と特にジオメトリ軸への機械軸の割り当てはすべてのチャネル内で異なることがあるため、チャネル軸の間にチャネル間の幾何学的な関係はありません。したがって、NCU グローバルフレームに対して可能なのは、オフセット、スケール、およびミラーリングのみです。回転をおこなうことはできません。

G58:軸別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)

平行移動オフセットのアブソリュート成分(荒削りオフセット)は、G58 で軸別にプログラム指令します。平行移動オフセットの追加成分(仕上げオフセット)は保持されます。

G59:追加の軸別プログラマブルゼロオフセット(仕上げオフセット)

平行移動オフセットの追加成分(仕上げオフセット)は、G59 で軸別にプログラム指令します。平行移動オフセットのアブソリュート成分(荒削りオフセット)は保持されます。

G59 は、仕上げオフセットが解放されているときにのみ使用できます。

- MD18600 $MN_MM_FRAME_FINE_TRANS = TRUE
- MD24000 $MC_FRAME_ADD_COMPONENTS = TRUE

座標回転

空間内の向きは、以下のようにフレーム回転を介して定義されます。

- ROT による回転は、すべてのジオメトリ軸に対して単独の回転を定義します。
- ROTS、AROTS、CROTS による立体角は、空間内の平面の向きを定義します。
- TOFRAME によるフレーム回転は、工具方向を向いている Z 軸を持ったフレームを定義します。

スケーリング

SCALE は、すべてのジオメトリ軸と付加軸のプログラマブルスケーリング係数のプログラム指令に使用されます。

新しいスケーリングが以前のスケーリング、座標回転、座標移動、またはミラーリングを基にする場合、ASCALE をプログラム指令してください。

ミラーリング

以下のマシンデータを使用して、ミラーリングの実行方法を設定します。

MD10610 $MN_MIRROR_REF_AX

結合

結合演算子「:」を使用して、フレームとフレーム成分を結合することができます。チャネルのすべての有効フレームの結合から、チャネル別の有効フレーム$P_ACTFRAME が得られます。

$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

必要条件

インクレメンタル指令 G91

G91 によるインクレメンタル指令は、原点オフセットが選択されると、インクレメンタルにプログラム指令された値にオフセット値が追加で移動されるように定義されます。

この動作は、次のセッティングデータの設定に依存します。

SD42440 $SC_FRAME_OFFSET_INCR_PROG (フレームのゼロオフセット)

値	意味
1	ゼロオフセットは、FRAME と軸のインクレメンタル指令に適用されます(= 初期設定)。
0	プログラム指令軌跡だけが移動されます。

適合性

フレームの書き込み、読み取り、起動をおこなう場合(チャネル協調の使用時など)は、複数のチャネル内で一貫した動作を達成するよう心がけてください。フレームのチャネル間をまたぐ起動はサポートされていません。

11.2 軸

11.2.1 概要

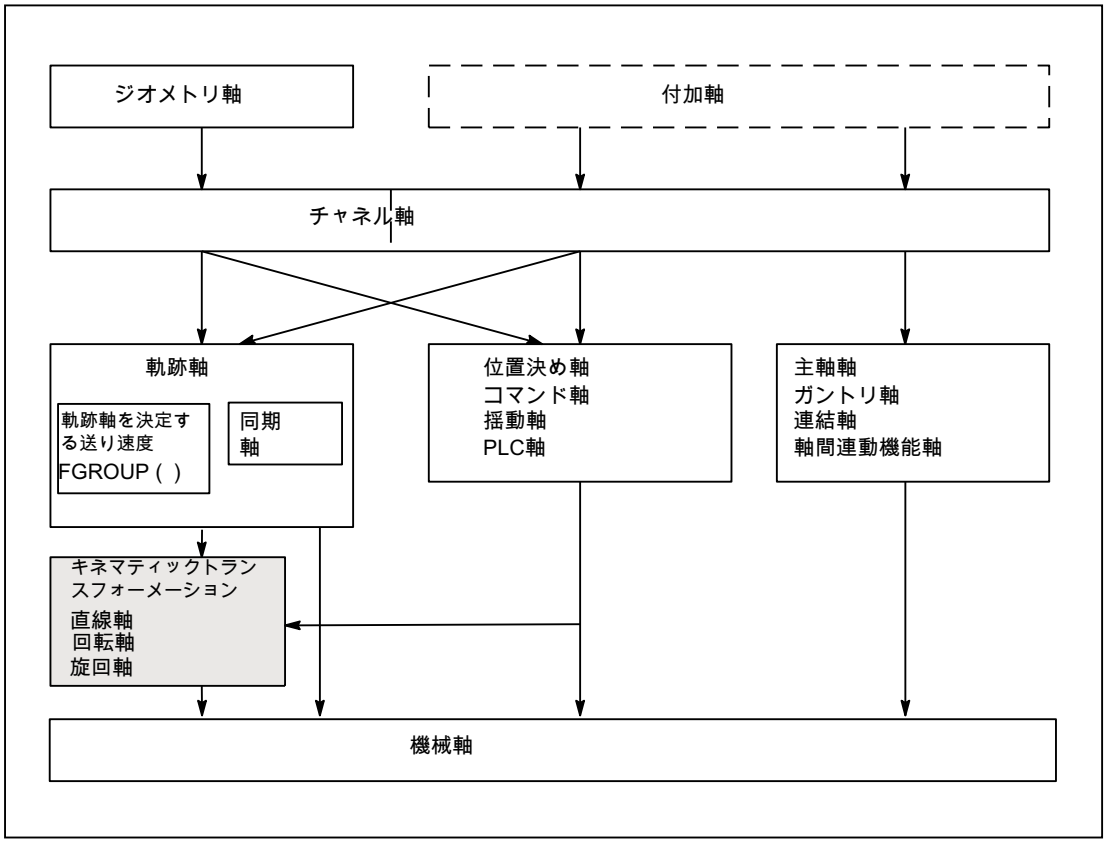


図 11-2 ジオメトリ軸、付加軸および機械軸の間の関係

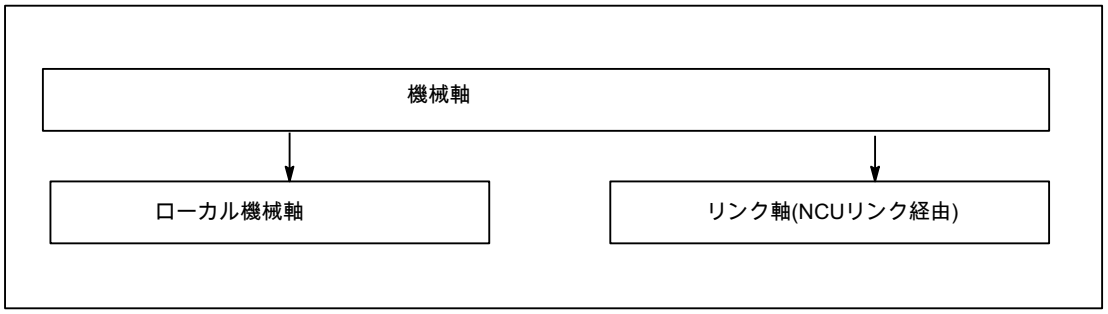


図 11-3 ローカルおよび外部機械軸(リンク軸)

11.2.2 機械軸

意味

機械軸は、工作機械上に実際に存在する軸です。

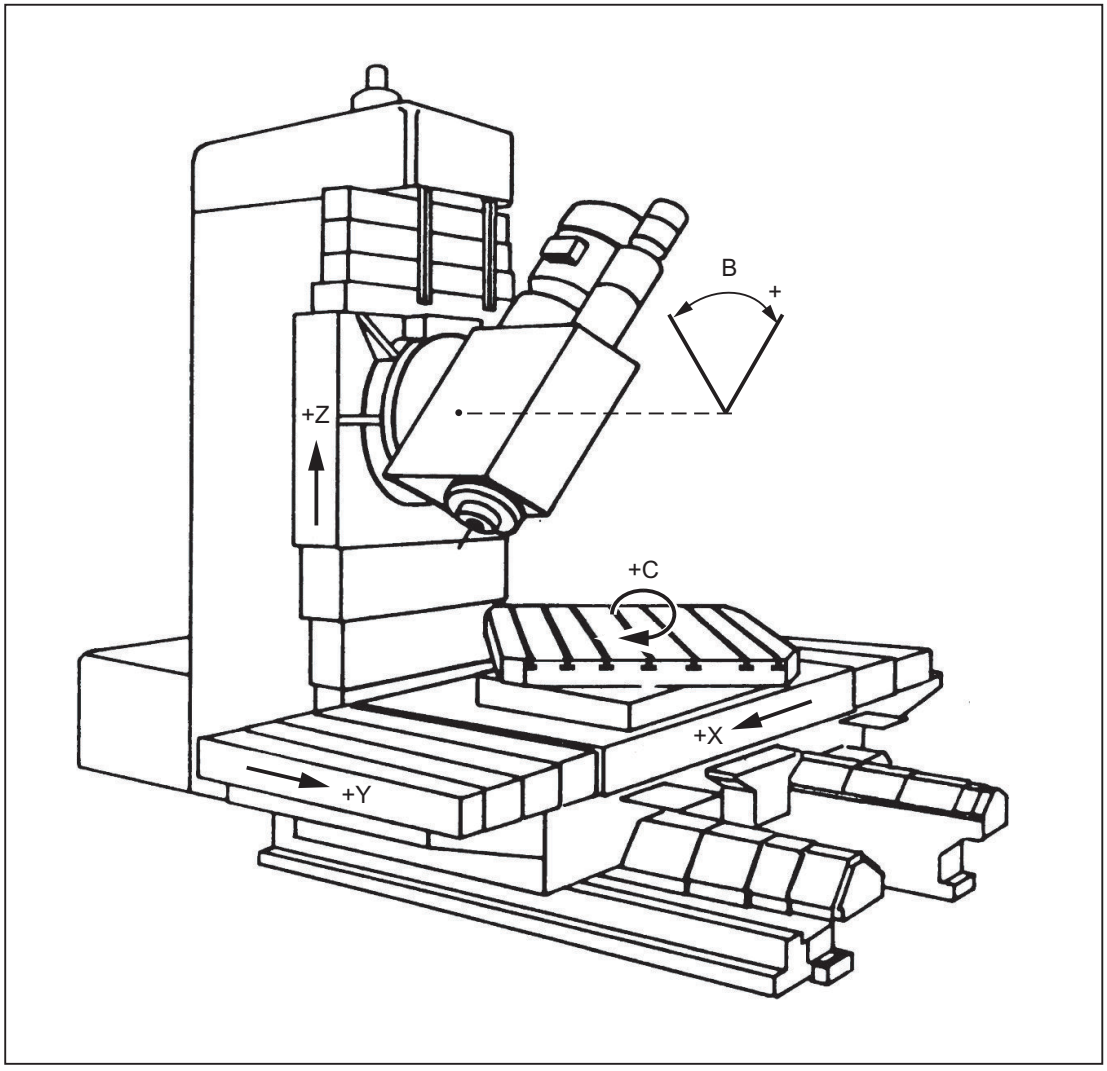


図 11-4　直交機械の機械軸 X、Y、Z、B、S

用途

以下は、機械軸にすることができます。

- ジオメトリ軸 X、Y、Z
- 旋回軸 A、B、C

- ローダ軸
- 工具タレット
- 工具マガジン用の軸
- 自動工具交換装置用の軸
- 主軸スリーブ
- パレット交換装置用の軸
- その他

11.2.3 チャネル軸

意味

各ジオメトリ軸および各付加軸は、チャネルに割り当てられます。 ジオメトリ軸および付加軸は、常に「自身の」チャネルで移動します。

11.2.4 ジオメトリ軸

意味

3 つのジオメトリ軸は、常に理論上の直交座標系を作成します。

FRAMES (ゼロオフセット、座標回転、スケーリング、ミラーリング)を使用することにより、ワーク座標系(WCS)のジオメトリ軸を BCS に投影することができます。

用途

ジオメトリ軸は、ワーク形状(輪郭)のプログラミングに使用されます。

平面選択 G17、G18 および G19 (DIN 66217)では、常に 3 つのジオメトリ軸を基準とします。 そこで 3 つのジオメトリ軸を X、Y および Z と命名すると便利です。

11.2.5 付加軸

内容

ジオメトリ軸とは対照的に、付加軸間で定義される幾何学的な関係はありません。

注記
ジオメトリ軸には、直交座標系の形で正確に定義された関係があります。

付加軸は、基本座標系(BCS)の一部です。 FRAMES (平行移動、スケーリング、ミラーリング)を使用すると、ワーク座標系の付加軸を基本座標系に投影できます。

用途

代表的な付加軸は、次のとおりです。

- 回転軸
- 工具マガジン軸
- 工具リボルバ軸
- ローダ軸

11.2.6 軌跡軸

意味

軌跡軸は、互いに補間されます(チャネルのすべての軌跡軸には共通の軌跡補間器があります)。

1 つのチャネルのすべての軌跡軸には、同一の加速区間、定速移動区間および減速区間があります。

アドレス F でプログラミングされた送り速度(軌跡送り速度)は、以下の例外を除き、ブロックにプログラミングされたすべての軌跡軸に適用されます。

- 命令 FGROUP により軌跡速度への制御なしとして定義された軸が、プログラミングされました。
- 命令 POS または POSA によりプログラミングされた軸には、個々の送り速度設定があります(軸補間器)。

用途

軌跡軸は、プログラミングされた輪郭によるワークの加工に使用されます。

11.2.7 位置決め軸

意味

位置決め軸は、別々に補間されます(位置決め軸毎に独自の軸補間器があります)。位置決め軸毎に、独自の送り速度特性と加速特性があります。 軌跡軸に加えて、位置決め軸をプログラム指令できます(同一ブロックの場合でも)。 軌跡軸補間(軌跡補間器)は、位置決め軸による影響を受けません。 軌跡軸と個々の位置決め軸は、必ずしも同時にブロックの終点に到達しません。

命令 POS と POSA により、位置決め軸がプログラム指令され、ブロック切り替え条件が定義されます。

- POS
  ブロック切り替えは、軌跡軸と位置決め軸がブロックの終点に到達したときに実行されます。
- POSA
  ブロック切り替えは、軌跡軸がブロックの終点に到達したときに実行されます。 位置決め軸は、ブロック境界を越えてブロックの終点まで移動を継続します。

同時位置決め軸は、以下の点で位置決め軸と異なります。

- PLC からブロックの終点を受け取るだけです
- いつでも起動できます(ブロック境界以外で)
- 現在のパートプログラムの実行に影響しません。

用途

代表的な位置決め軸は、次のとおりです。

- ワークを機械から取り出すためのローダ
- 工具マガジン/タレット

参照先

詳細については、「P3: SINUMERIK 840D sl 用 PLC 基本プログラム (ページ 1043)」と「S1:主軸 (ページ 1555)」の章を参照してください。

参照先:

- 『機能マニュアル、上級機能』；「位置決め軸(P2)」
- 『機能マニュアル、応用機能』；「ガントリ軸(G1)」
- 『機能マニュアル、応用機能』；「軸の連結と ESR (M3)」
- 『機能マニュアル、上級機能』；「揺動(P5)」
- 『機能マニュアル、シンクロナイズドアクション』

11.2.8 メイン軸

意味

メイン軸は、メインランにより補間される軸です。

この補間は、以下手段で開始できます。

- シンクロナイズドアクションから
  (ブロック単位、モーダル、または静的シンクロナイズドアクションを介したイベントによるコマンド軸として)
- PLC から PLC 基本プログラムの特定のファンクションブロックを介して
  (同時位置決め軸または PLC 軸と呼ばれます)
- セッティングデータを介してまたはパートプログラムから
  (非同期またはブロック同期揺動軸として)

チャネル制御

メイン軸により補間される軸は、以下のものに応答します。

- NC ストップ
- アラーム処理
- プログラム制御

- エンドオブプログラム
- RESET

注記

プログラムの終了時の応答は変化します。 軸の動作は必ずしも完了する必要がないので、プログラムの終了を越えて実行できます。

用途

メインランの特定の軸は、NC プログラムシーケンスによりトリガされるチャネル応答時に分離し、PLC から制御することができます。 これらの軸はメインランでも補間され、チャネルとプログラムシーケンスに対して独立して応答します。

PLC 制御軸は、NC により独立して制御できます。 これは、以下の動作と関係があります。

- 軸をキャンセルするためのシーケンス(残移動距離削除に相当)
- 軸の停止または中断
- 軸の継続(動作の継続シーケンス)
- 軸の初期状態へのリセット

11.2.9 同期軸

意味

同期軸は軌跡軸の一つですが、工具軌跡速度の計算には使用されません。これらは、軌跡軸とともに補間されます(1 つのチャネルのすべての軌跡軸と同期軸には共通の軌跡補間器があります)。

チャネルのすべての軌跡軸とすべての同期軸には、同一の加速区間、定速移動区間および減速区間があります。

アドレス F でプログラム指令送り速度(軌跡送り速度)は、ブロックのプログラム指令されたすべての軌跡軸に適用されますが、同期軸には適用されません。

同期軸は、プログラム指令軌跡を移動するために、軌跡軸の場合と同じ時間がかかります。

FGROUP 命令

命令 FGROUP は、軸が送り速度の定義された**軌跡軸**(軌跡速度の計算に使用)であるか**同期軸**(軌跡速度の計算に使用されない)であるかを指定します。

例

プログラムコード	コメント
N05 G00 G94 G90 M3 S1000 X0 Y0 Z0	;
N10 FGROUP(X,Y)	; 軸 X および Y は軌跡軸であり、 Z は同期軸です
N20 G01 X100 Y100 F1000	; 送り速度 1000 mm/min をプログラム。 軸 X の送り速度 = 707 mm/min。 軸 Y の送り速度 = 707 mm/min。
N30 FGROUP (X)	; 軸 X は軌跡軸であり、 軸 Y が同期軸です。
N20 X200 Y150	; プログラム:送り速度 1000 mm/min 軸 X の送り速度 = 1000 mm/min 軸 Y の送り速度は 500 mm/min に設定されます。移動する距離が半分だけだからです。

注記

FGROUP 命令には、チャネル軸名称を使用してください。
これは、次のマシンデータにより定義されます。
MD20080 $MC_AXCONF_CHANAX_NAME_TAB (チャネルのチャネル軸名称)

用途

ヘリカル補間の場合、FGROUP をプログラム指令して以下ことの決定できます。

- プログラム指令送り速度をこの軌跡に対して有効にすべきかどうか
  (3 つのプログラム指令軸はすべて軌跡軸です)。
- プログラム指令送り速度をこの円弧に対して有効にすべきかどうか
  (2 つの軸は軌跡軸であり、切り込み軸は同期軸です)。

11.2.10 軸設定

ジオメトリ軸、付加軸、チャネル軸、機械軸、およびドライブの割り当て

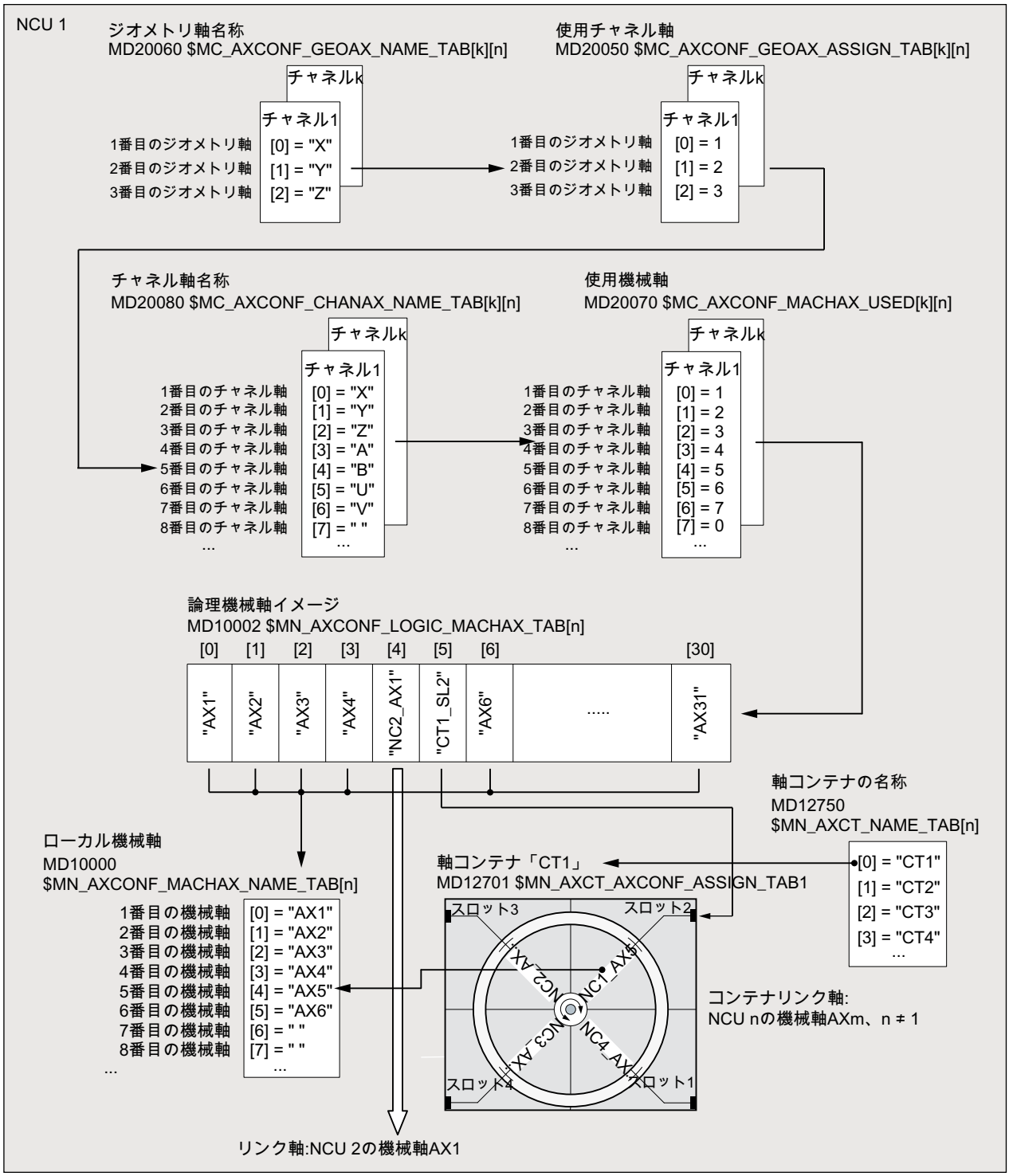


図 11-5　軸割り当て

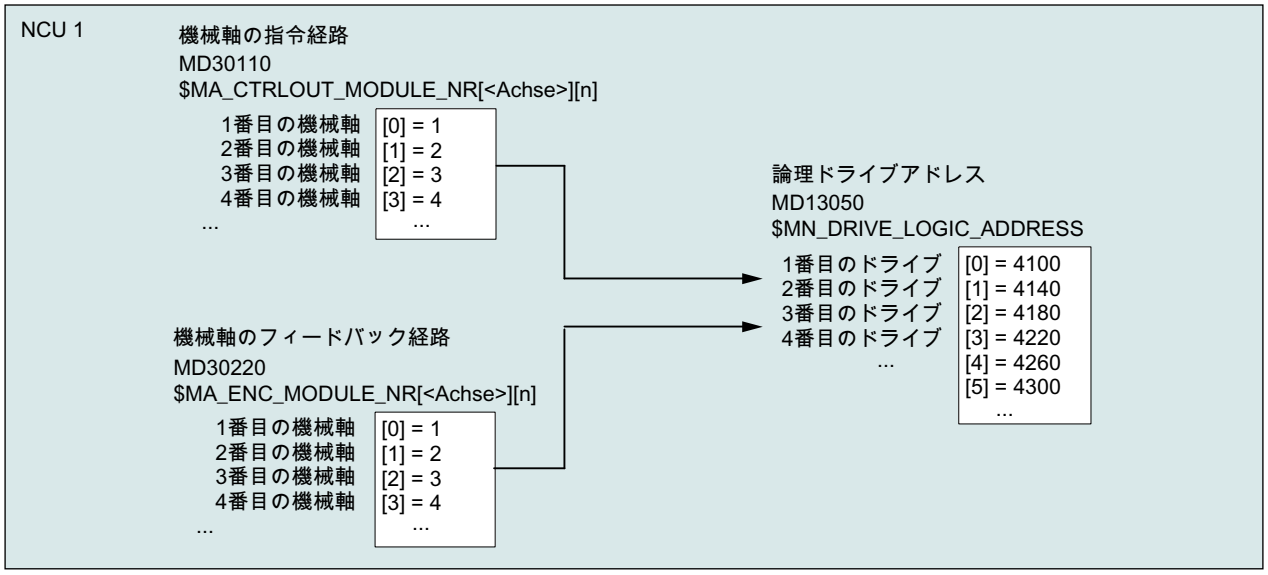


図 11-6　ドライブの割り当て

必要条件

- ユーザー定義軸名称の先頭のゼロは無視されます。
  MD10000 `$MN_AXCONF_MACHAX_NAME_TAB[0] = X01 は X1 に対応します。
- ジオメトリ軸は、ギャップなしで昇順でチャネル軸に割り当ててください。
- ジオメトリ軸以外のすべてのチャネル軸は、付加軸です。

チャネル軸ギャップ

通常、マシンデータ MD20080 $MC_AXCONF_CHANAX_NAME_TAB で定義されたチャネル軸ごとに、MD20070 $MC_AXCONF_MACHAX_USED により機械軸を割り当てる必要があります。

異なった数の機械軸を持つ機械の一括セットアップを簡素化するために、機械軸に割り当てられていないチャネル軸も定義できます。その結果、チャネル軸のナンバリングシーケンスに空きが生じることがあります。

すべてのチャネル軸ギャップを、意図的に有効にしてください。

MD11640 $MN_ENABLE_CHAN_AX_GAP = 1

これを有効にしない場合、マシンデータ MD20070 $MC_AXCONF_MACHAX_USED の値 0 で、機械軸へのさらなるチャネル軸の割り当てが停止します。

例

チャネル軸 B は機械軸に割り当てられていません。

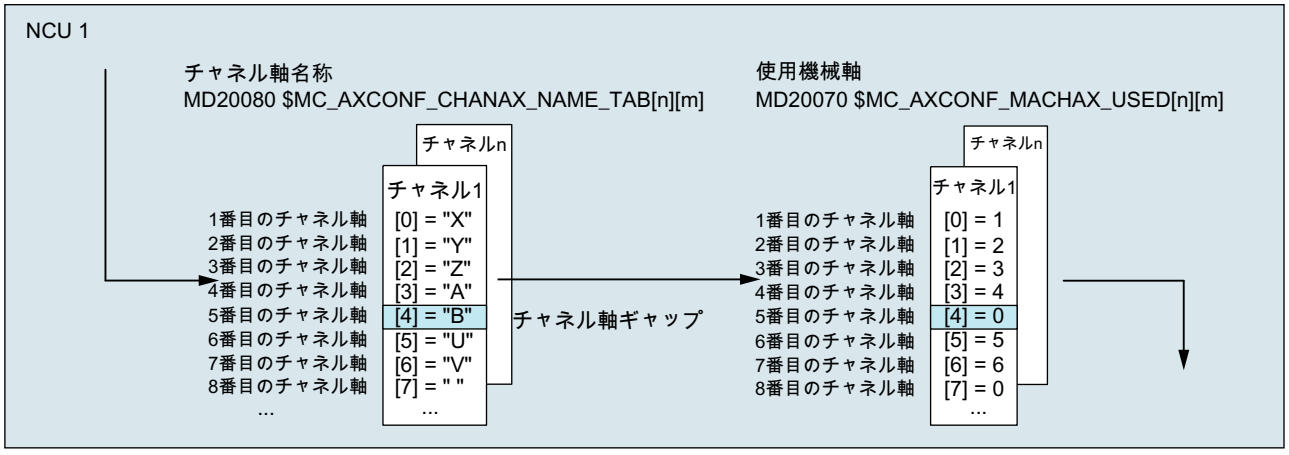


図 11-7　チャネル軸ギャップによる軸設定(抜粋)

必要条件

- 機械軸(チャネル軸ギャップ)が割り当てられていないチャネル軸は、チャネル軸の数とインデックスに関しては、対応する機械軸を持つ通常のチャネル軸とまったく同じに扱われます。
- 機械軸(チャネル軸ギャップ)が割り当てられていないチャネル軸がジオメトリ軸として定義されている場合、この軸はアラームなしで拒否されます。

11.2.11　リンク軸

意味

リンク軸とは、NCU から移動する、その NCU 上にはない機械軸です。 移動する NCU の論理機械軸イメージに対応したマシンデータには、ローカル機械軸の名称は入力されず、NCU と機械軸が物理的に接続されている NCU の機械軸名称が入力されます。

次に示す例では、NCU2 の機械軸 AX1 が NCU1 から移動します。

- NCU1: MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB[n] = NC2_AX1

必要条件

リンク軸を使用するための前提条件として、関係する NCU がリンク通信で接続されていること。 リンク軸とリンク通信機能についての詳細は、下記を参照してください。

参照先:
『機能マニュアル、上級機能』; 「NCU と操作パネルの n:m 接続(B3)」

11.3 原点とレファレンス点

11.3.1 作業空間のレファレンス点

原点とレファレンス点

機械のニュートラルポジションは、座標軸および機械の構造的な特性から得られます。座標系の原点は、機械の適切なレファレンス点をニュートラルポジションで定義することにより得られます。

座標系(MCS、BCS、BZS、SZS、WCS)の位置は、原点によって決定されます。

原点		レファレンス点	
	M = 機械原点		**R =** レファレンス点
	W = ワーク原点		**T =** 工具ホルダレファレンス点

機械原点 M

機械原点 M は、機械座標系 MCS を定義します。 他のすべてのレファレンス点は、機械原点を基準にします。

ワーク原点 W

ワーク原点 W は、機械原点 M に関連してワーク座標系を定義します。プログラミングされたパートプログラムブロックは、ワーク座標系 WCS で実行されます。

レファレンス点 R

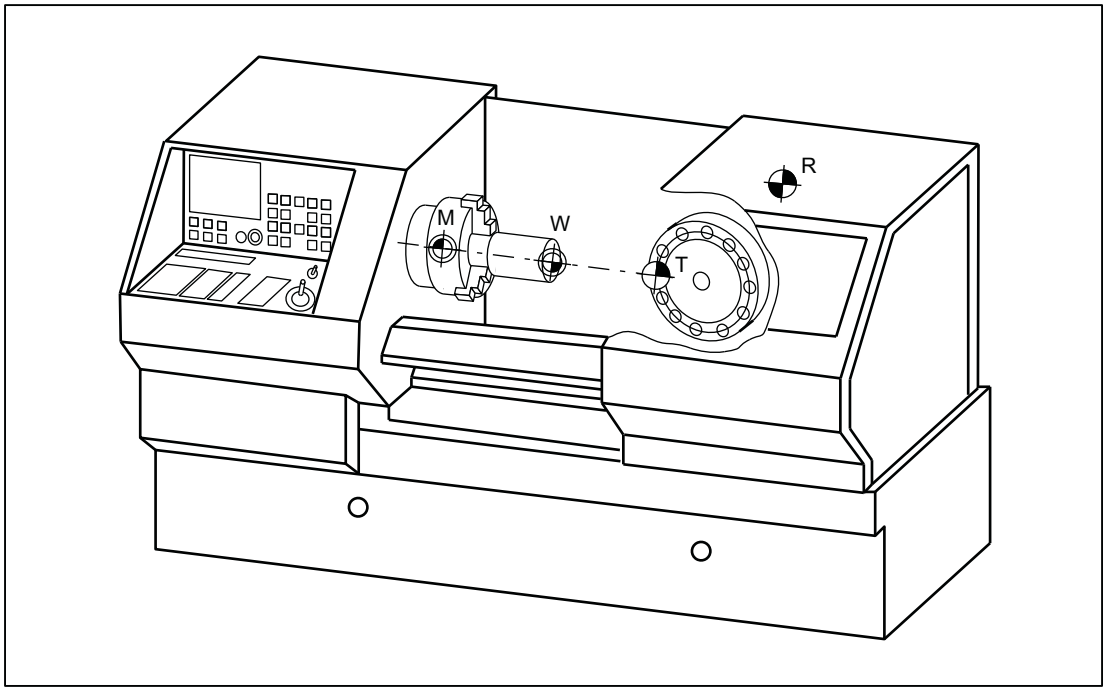
レファレンス点 R の位置は、カムスイッチにより定義されます。 レファレンス点 R は、位置検出器の位置合わせをします。

インクレメンタルエンコーダを使用する場合、コントロールシステムの電源をオンにするたびに、レファレンス点にアプローチしてください。 その場合のみ、コントロールシステムは検出器によって動作でき、すべての位置を座標系に投影できます。

工具ホルダレファレンス点 T

工具ホルダレファレンス点 T は、工具ホルダロケータにあります。 工具長を入力することにより、コントロールシステムは工具刃先(TCP 工具中心点)と工具ホルダレファレンス点の間の距離を計算します。

例: 旋盤の原点とレファレンス点



11.3.2 座標系とレファレンス点の位置

制御装置の POWER ON

インクリメンタル検出器の場合、制御装置がすべての位置データを座標系に投影できるように、制御装置が起動されるたびにレファレンス点にアプローチしてください。

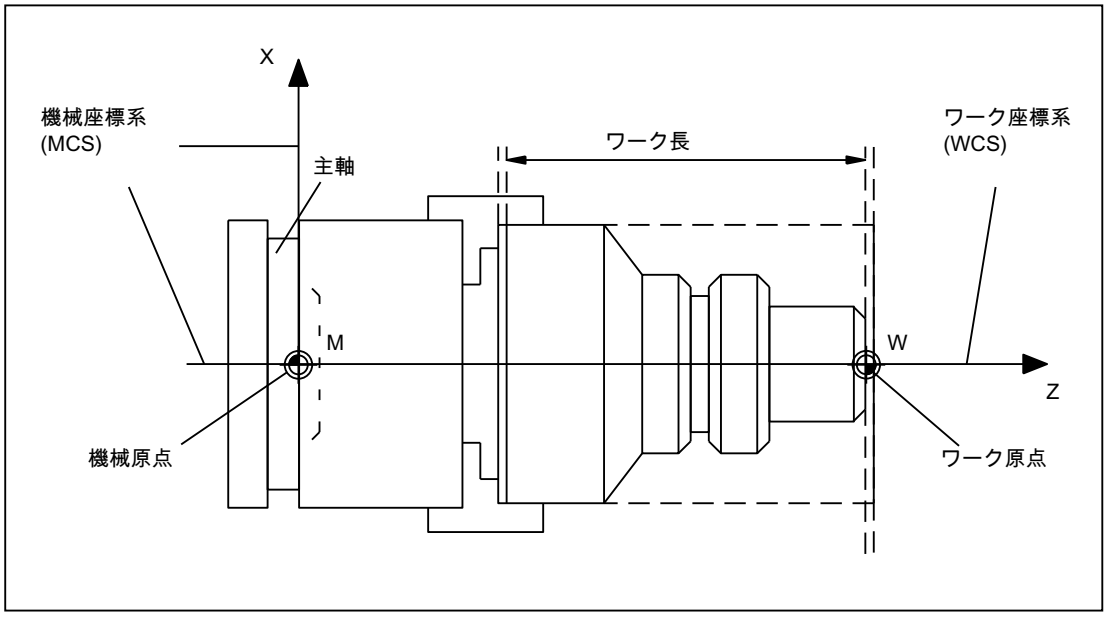


図 11-8　機械原点 M およびワーク原点 W による座標系の位置

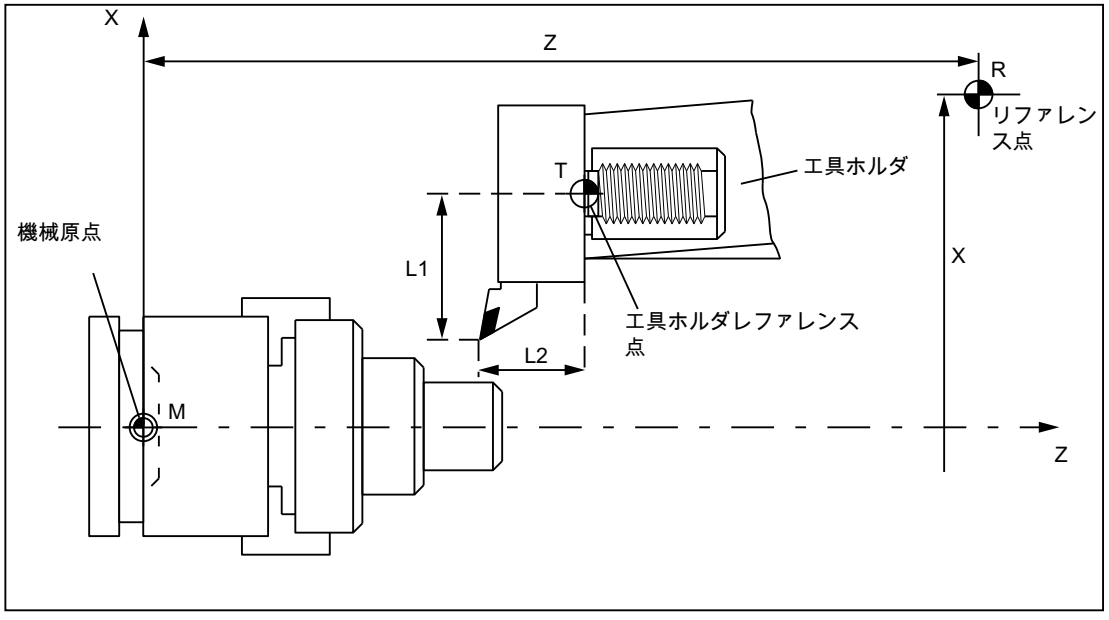


図 11-9　機械原点に対するレファレンス点の位置

11.4 座標系

11.4.1 概要

定義

DIN 66217 では、工作機械をプログラムする場合、直角の直交(デカルト)座標系を使用する必要があることを規定しています。座標軸の正の方向は、「右手の法則」を使用して特定できます。

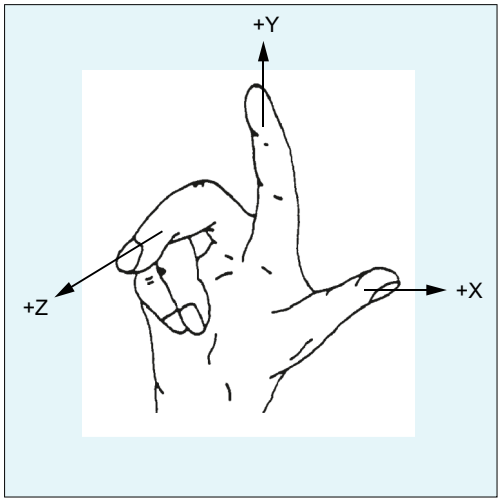


図 11-10　右手の法則

プログラミングでワークを参照する場合、座標系を使用します。プログラミングは、工具またはワークが移動しているかどうかにかかわらず実行できます。プログラミングしているとき、常に工具はワークの座標系に対して移動すると見なされ、ワークが静止状態であることを意図しています。

回転軸の正(右回り)の回転も「右手の法則」を使用して定義できます。右手の親指が座標軸(直線軸)の正方向を示す場合、各指は該当する回転軸の正の回転方向を示します。

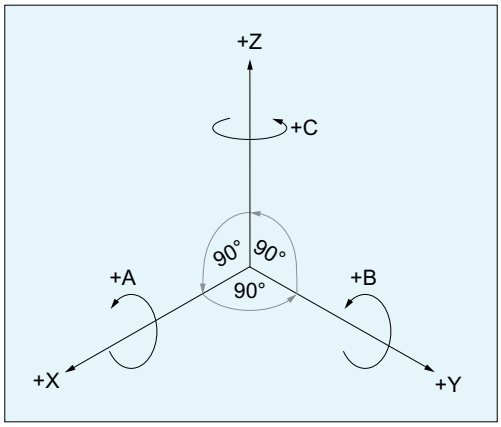


図 11-11　時計方向回転の直交のデカルト座標系

座標系

次の座標系を工作機械に定義します。

座標系	略語	備考
ワーク座標系(**W**orkpiece **C**oordinate **S**ystem)	WCS	ワークを加工するためのジオメトリ軸の移動動作のプログラミングは、WCS に書き込まれます。
設定可能ゼロオフセットシステム(**S**ettable **Z**ero **S**ystem)	SZS	フレームを使用した座標変換: WCS ⇒ SZS
基本ゼロオフセットシステム(**B**asic **Z**ero **S**ystem)	BZS	フレームを使用した座標変換: SZS ⇒ BZS:
基本座標系(**B**asic **C**oordinate **S**ystem)	BCS	フレームを使用した座標変換: BZS ⇒ BCS:
機械座標系(**M**achine **C**oordinate **S**ystem)	MCS	フレームとキネマティックトランスフォーメーションを使用した座標変換: BCS ⇒ MCS: WCS でプログラム指令されたジオメトリ軸の移動動作は、フレームとキネマティックトランスフォーメーションを使用して MCS の機械軸に割り当てます。キネマティックトランスフォーメーションは、MCS からの BCS の導出に使用されます。この方法で、たとえば、手動移動などの機能が MCS ではなく、BCS で有効になります。

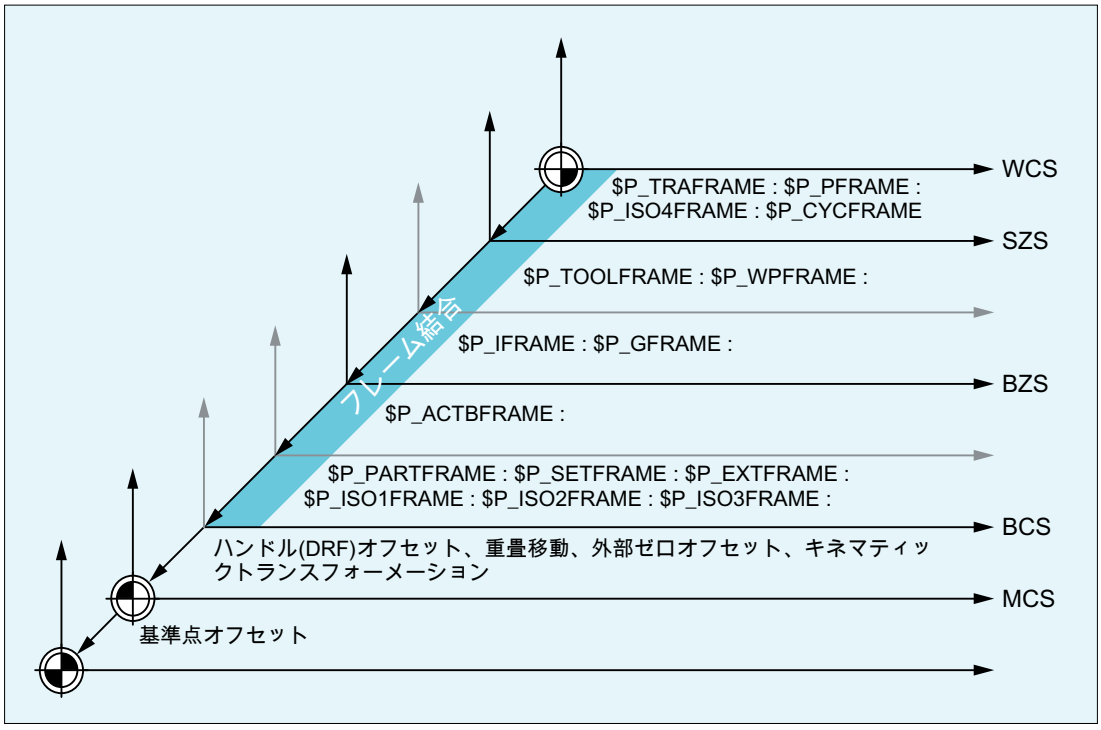


図 11-12　座標系、フレーム、およびキネマティックトランスフォーメーション

11.4.2 機械座標系(MCS)

機械座標系(MCS)

機械座標系(MCS)は、物理的に使用可能なすべての機械軸で構成されます。

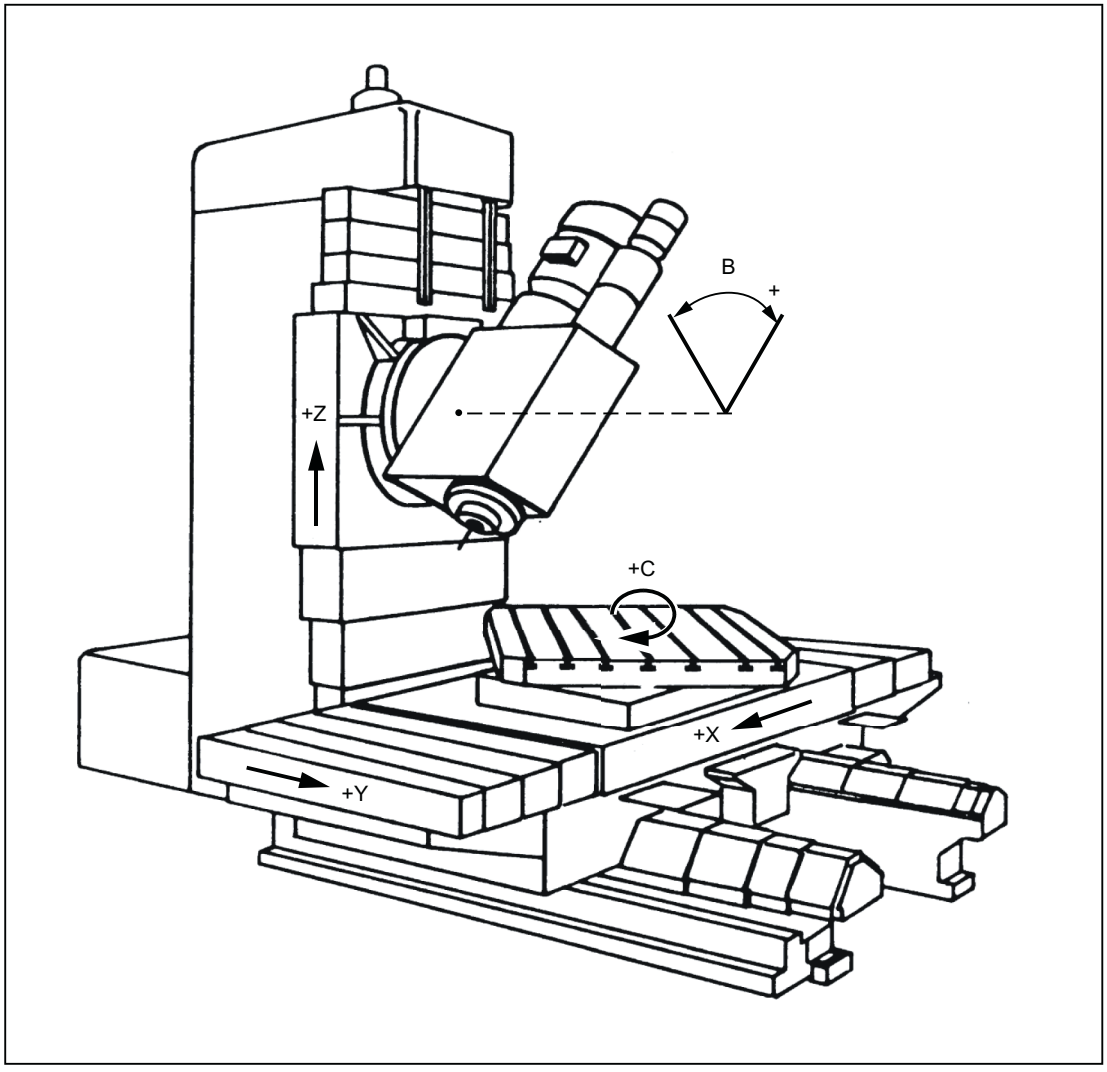


図 11-13　機械軸 X、Y、Z、B、C を有する MCS (5 軸フライス盤)

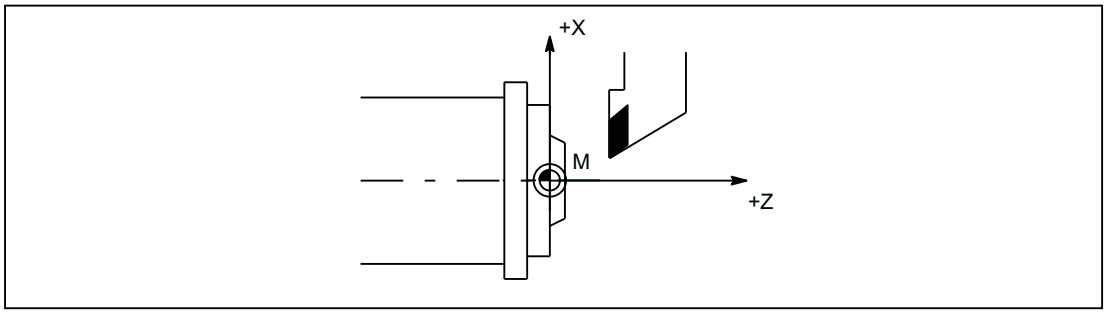


図 11-14　機械軸 X、Z を有する MCS (旋盤)

軸プリセットオフセット

機械座標系(機械原点)での制御のレファレンス点は、「プリセットオフセット (PRESETON)」機能を使用してリセットできます。

⚠ 注意

エンコーダ位置合わせの損失

プリセットオフセット後、当該の機械軸は「原点未確立」状態になります。 つまり、絶対値エンコーダを使用する場合は、エンコーダの位置合わせが失われるため、 再度位置合わせをおこなってください(レーザー干渉計などで)。 そのため、PRESETON を絶対値エンコーダと組み合わせて使用するのはお勧めしません。

注記

この機能は、レファレンス点が不要な機械軸に**のみ**使用することを推奨します。

元の機械座標系に復帰するには、たとえば G74(レファレンス点復帰)を使用して、機械軸の再原点確立をおこなってください。

機械軸はプリセットオフセットでは移動しません。

参照先

- プログラミングマニュアル、基本編
  「補助命令」の「レファレンス点復帰 (G74)」の章
- プログラミングマニュアル、上級編
  「座標変換(FRAMES)」の「プリセットオフセット (PRESETON)」の章

11.4.2.1 原点確立状態をリセットして現在値を設定(PRESETON)

機能

PRESETON()は、機械座標系(MCS)の 1 つ以上の軸に対して、新しい現在値を設定します。これは、軸 MCS のゼロオフセットに対応します。軸は移動しません。

PRESETON によって、同期による先読み停止が起動されます。実際の位置は、軸が停止状態になるまでは割り当てられません。

軸が PRESETON のチャネルに割り当てられていない場合、次の手順は以下のマシンデータでパラメータ設定された置換動作に応じて異なります。

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)

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 新しい現在値を PRESETON で機械座標系に設定すると、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: 原点確立状態をリセットして現在値を設定
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)での機械軸の新しい現在値
入力は、現在の有効な単位系(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

システム変数

$AC_PRESET

軸別システム変数$AC_PRESET は、機械軸の原点確立後に、現在オフセットされている MCS の原点からのベクトルを 、元の MCS_0 の原点に提供します。

$AC_PRESET<軸> = $AC_PRESET<軸> +「MCS 内の軸の現在の実位置」-「PRESETON 実位置」

ゼロオフセットは、以下のシステム変数で無効にすることができます。

PRESETON(<軸>, $VA_IM + $AC_PRESET[<軸>]) ;「MCS 内の軸の現在の実位置」+「オフセット」

例

プログラムコード

```
N10 G1 X10 F5000
N20 PRESETON(X, $AA_IM[X]+70) ; 現在値 = 10 + 70 = 80 =>
                              ; $AC_PRESET = $AC_PRESET - 70
```

必要条件

PRESETON を使用してはならない軸

- 軌跡軸の移動
- 位置決め軸の移動
- 主軸モードの移動指令軸
- 同時位置決め軸の移動(FC18)
- 座標変換に関連する軸

- 以下の Safety Integrated 機能のうち、1 つ以上が有効になっている軸:
  - セーフソフトウェアリミットスイッチ(SE):
    MD36901 $MA_SAFE_FUNCTION_ENABLE[<安全軸>]、ビット 1 = 1
  - セーフソフトウェアカム、セーフカムトラック(SN):
    MD36901 $MA_SAFE_FUNCTION_ENABLE[<安全軸>]、ビット 8 ... 15 = 1
- 揺動軸
- ハース軸
- ガントリグループの同期軸
- パートプログラム(G74)からのレファレンス点復帰が有効になっている軸
- 速度/トルクカップリング(マスタ-スレーブ)のスレーブ軸

ジオメトリ軸

- PRESETON は、その他のジオメトリ軸が同時にチャネルで移動していないときは、停止ジオメトリ軸で使用できます。
- PRESETON は、その他のジオメトリ軸が同時にチャネルで移動しているときでも、その軸が「中立軸」状態であるか、コマンド軸として移動している場合は、停止ジオメトリ軸で使用できます。

プログラムコード	コメント
N10 G0 X0 Y0	; X, Y:ジオメトリ軸
N15 RELEASE(Y)	; 中立軸
N20 PRESETON(Y,20)	; MCS = 20 での実位置 Y
N30 G0 X40	; ジオメトリ軸 X が移動
N40 M30	

例:別のジオメトリ軸(X)が、同時に"**中立軸**"状態に移動します。

プログラムコード	コメント
N10 G0 X0 Y0	; X, Y:ジオメトリ軸
N20 POS[X]=40 FA[X]=1000	; X コマンド軸
N30 DO PRESETON(Y,20)	; MCS = 20 での実位置 Y
N40 M30	

例:ジオメトリ軸(X)は、**コマンド軸**として同時に移動します。

PLC 制御軸

PRESETON は、PLC 制御軸で、その現在のタイプに従って使用できます。

主軸の状態

以下の表は、主軸で PRESETON をシンクロナイズドアクションで使用した場合の応答を示しています。

NC プログラムでの PRESETON			
主軸モード	**移動状態**	**NC プログラムへの割り当て**	**メイン軸**
開ループ速度制御モード	移動中	アラーム 22324	アラーム 22324
	停止	+	+
位置決めモード SPOS	移動中	-	+
	停止	+	+
ブロック境界を越えた位置決め SPOSA	移動中	アラーム 10610	-
軸モード	移動中	-	+
	停止	+	+
+:可 -:不可			

軸連結

- マスタ軸:PRESETON が引き起こすマスタ軸位置の急変では、スレーブ軸での移動は発生しません。カップリングは変更されません。
- スレーブ軸:スレーブ軸の重畳位置成分は PRESETON に影響されます。

ガントリグループ

ガントリグループのガイド軸で PRESETON を使用すると、ガントリグループのすべての同期軸でゼロオフセットも実行されます。

軸割り出し

PRESETON は割り出し軸で使用できます。

ソフトウェアのリミットスイッチ、動作範囲制限、プロテクションゾーン

PRESETON によるゼロオフセットの後に軸位置が、指定された制限の外側にある場合、軸の移動を試みるまではアラームが表示されません。

計算ありのブロック検索

PRESETON 命令は、ブロック検索時に収集され、NC スタートで実行されて NC プログラムを続行します。

位置に依存する NC/PLC インターフェース信号

位置に依存する NC/PLC インターフェース信号の状態は、新しい実位置に基づいて再決定されます。

例:固定点位置

- 設定された固定点位置 MD30600 $MA_FIX_POINT_POS[0...3] = <固定点位置 1...4>
- NC/PLC インタフェース信号 DB31、... DBX75.3 ... 5 (JOG モードで固定点へのアプローチ:到達)

軸が、イグザクトストップ許容範囲を伴う固定点位置にある場合は、関連する NC/PLC インターフェース信号が設定されます。NC/PLC インターフェース信号は、PRESETON により現在値が、固定点位置周辺のイグザクトストップ許容範囲の外側にある別の値に設定されると、リセットされます。

DRF オフセット

軸の DRF オフセットは、PRESETON によって削除されます。

重畳移動$AA_OFF

$AA_OFF によるシンクロナイズドアクションからの重畳移動は、PRESETON の影響を受けません。

オンライン工具補正 FTOC

FTOC によるシンクロナイズドアクションからの有効なオンライン工具補正は、PRESETON の後も有効なままです。

軸別の補正

軸別の補正は、PRESETON の後も有効なままです。

JOG モード

PRESETON は停止軸でのみ使用してください。

JOG モード、REF 機械機能

PRESETON は**使用しないでください**。

11.4.2.2 原点確立状態をリセットしない現在値の設定(PRESETONS)

機能

PRESETONS() は、機械座標系(MCS)の 1 つ以上の軸に対して、新しい現在値を設定します。これは、軸 MCS のゼロオフセットに対応します。軸は移動しません。

PRESETONS によって、同期による先読み停止が起動されます。実際の位置は、軸が停止状態になるまでは割り当てられません。

PRESETONS の軸がチャネルに割り当てられていない場合、次の手順は、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**。

必要条件

- **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、有効な検出器の以下の指令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)

セットアップ

軸マシンデータ

原点確立状態をリセットしない現在値の設定(PRESETONS)は、軸別に設定してください。

MD30455 $MA_MISC_FUNCTION_MASK、ビット 9 = 1

注記

PRESETON の解除

「原点確立状態をリセット**しない**現在値の設定 PRESETONS」機能を起動すると、「原点確立状態をリセット**する**現在値の設定 PRESETON」機能が解除されます。この 2 つのオプションを同時に使用することはできません。

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して現在値を設定
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
この値は、有効な単位系(インチ/メトリック)を示します。
有効な直径指定(DIAMON)が考慮されます
タイプ: REAL

システム変数

$AC_PRESET

軸別システム変数$AC_PRESET は、機械軸の原点確立後に、現在オフセットされている MCS の原点からのベクトルを 、元の MCS_0 の原点に提供します。

$AC_PRESET<軸> = $AC_PRESET<軸> +「MCS 内の軸の現在の実位置」-「PRESETONS 実位置」

ゼロオフセットは、以下のシステム変数で無効にすることができます。

PRESETONS(<軸>, $VA_IM + $AC_PRESET[<軸>]) ;「MCS 内の軸の現在の実位置」+「オフセット」

例

X 軸 MCS の 70 単位のゼロオフセット

X 軸(コマンド軸)のプログラム指令された終了位置は、PRESETONS で、新しい MCS に座標変換されます。

プログラムコード
```
N10 G1 X10 F5000
```

プログラムコード

```
N20 PRESETONS(X, $AA_IM[X]+70) ; 現在値 = 10 + 70 = 80 =>
                               ; $AC_PRESET = $AC_PRESET - 70
```

必要条件

PRESETON を使用してはならない軸

- 位置決め軸の移動
- 主軸モードの移動指令軸
- 同時位置決め軸の移動(FC18)
- 座標変換に関連する軸
- 軌跡軸の移動
- 揺動軸
- 以下の Safety Integrated 機能のうち、1 つ以上が有効になっている軸:
  - セーフソフトウェアリミットスイッチ(SE):
    MD36901 $MA_SAFE_FUNCTION_ENABLE[<安全軸>]、ビット 1 = 1
  - セーフソフトウェアカム、セーフカムトラック(SN):
    MD36901 $MA_SAFE_FUNCTION_ENABLE[<安全軸>]、ビット 8 ... 15 = 1
- ハース軸
- ガントリグループの同期軸
- パートプログラム(G74)からのレファレンス点復帰が有効になっている軸
- 速度/トルクカップリング(マスタ-スレーブ)のスレーブ軸

ジオメトリ軸

- PRESETONS は、その他のジオメトリ軸が同時にチャネルで移動していないときは、停止ジオメトリ軸で使用できます。
- PRESETONS は、その他のジオメトリ軸が同時にチャネルで移動しているときでも、その軸が「中立軸」状態であるか、コマンド軸として移動している場合は、停止ジオメトリ軸で使用できます。
  例:別のジオメトリ軸(X)が、同時に**"中立軸"**状態に移動します。

プログラムコード	コメント
N10 G0 X0 Y0	; X, Y:ジオメトリ軸
N15 RELEASE(Y) 1)	; 中立軸
N20 PRESETONS(Y,20)	; MCS = 20 での実位置 Y
N30 G0 X40	; ジオメトリ軸 X が移動
N40 M30	
1) **注記** シンクロナイズドアクションの動作部分で軸を解放しても、適切なタイミングで解放されるとは限りません。 N20 ID=1 WHEN 20.0 < $AA_IM[X] DO RELEASE(Y) PRESETONS(Y,20) ; 推奨されません。	

例:その他のジオメトリ軸(X)が**コマンド軸**と同時に移動している

プログラムコード	コメント
N10 G0 X0 Y0	; X, Y:ジオメトリ軸
N20 POS[X]=40 FA[X]=1000	; X コマンド軸
N30 PRESETONS(Y,20)	; MCS = 20 での実位置 Y
N40 M30	

PLC 制御軸

PRESETONS は、PLC 制御軸で、その現在のタイプに従って使用できます。

主軸の状態

以下の表は、主軸で PRESETONS をシンクロナイズドアクションで使用した場合の応答を示しています。

NC プログラムでの PRESETONS			
主軸モード	**移動状態**	**NC プログラムへの割り当て**	**メイン軸**
速度制御モード	移動中	アラーム 22324	アラーム 22324
	停止	+	+
位置決めモード SPOS	移動中	-	+
	停止	+	+
ブロック境界を越えた位置決め SPOSA	移動中	アラーム 10610	-
軸モード	移動中	-	+
	停止	+	+
+:可 -:不可			

軸連結

- マスタ軸:PRESETONS が引き起こすマスタ軸位置の急変では、スレーブ軸での移動は発生しません。カップリングは変更されません。
- スレーブ軸:スレーブ軸の重畳位置成分は PRESETONS に影響されます。

ガントリグループ

ガントリグループのガイド軸で PRESETONS を使用すると、ガントリグループのすべての同期軸でゼロオフセットも実行されます。

軸割り出し

PRESETONS は割り出し軸で使用できます。

ソフトウェアのリミットスイッチ、動作範囲制限、プロテクションゾーン

PRESETONS によるゼロオフセットの後に軸位置が、指定された制限の外側にある場合、軸の移動を試みるまではアラームが表示されません。

計算ありのブロック検索

PRESETONS 命令は、ブロック検索時に収集され、NC スタートで実行されて NC プログラムを続行します。

位置に依存する NC/PLC インターフェース信号

位置に依存する NC/PLC インターフェース信号の状態は、新しい実位置に基づいて再決定されます。

例:固定点位置

- 設定された固定点位置 MD30600 $MA_FIX_POINT_POS[0...3] = <固定点位置 1...4>
- NC/PLC インタフェース信号 DB31、... DBX75.3 ... 5 (JOG モードで固定点へのアプローチ:到達)

軸が、イグザクトストップ許容範囲を伴う固定点位置にある場合は、関連する NC/PLC インターフェース信号が設定されます。NC/PLC インターフェース信号は、PRESETONS により現在値が、固定点位置周辺のイグザクトストップ許容範囲の外側にある別の値に設定されると、リセットされます。

DRF オフセット

軸の DRF オフセットは、PRESETONS によって削除されます。

重畳移動$AA_OFF

$AA_OFF によるシンクロナイズドアクションからの重畳移動は、PRESETONS の影響を受けません。

オンライン工具補正 FTOC

FTOC によるシンクロナイズドアクションからの有効なオンライン工具補正は、PRESETONS の後も有効なままです。

軸別の補正

軸別の補正は、PRESETONS の後も有効なままです。

JOG モード

PRESETONS は停止軸でのみ使用してください。

JOG モード、REF 機械機能

PRESETONS は**使用しないでください**。

シンクロナイズドアクション

軸のシンクロナイズドアクションで、複数の PRESETONS 命令が含まれる場合、左から右の順で最後の命令のみが実際に実行されます。

例:

プログラムコード

```
N10 ID=1 WHEN TRUE DO PRESETONS(X,40) PRESETONS(X,39) PRESETONS(X,38)
```

プログラムコード

```
; 以下は、上記と同等です。
N10 ID=1 WHEN TRUE DO PRESETONS(X,38)
```

11.4.3 基本座標系(BCS)

基本座標系(BCS)

基本座標系(BCS)は、互いに垂直な 3 つの軸(ジオメトリ軸)と、幾何学的な相互関係を持たない他の付加軸で構成されます。

キネマティック座標変換のない工作機械の場合

BCS をキネマティック座標変換(たとえば、TRANSMIT /円筒補間、5 軸座標変換および3 つの機械軸まで)なしで MCS に投影できる場合、BCS と MCS は常に一致します。

そのような機械上では、機械軸とジオメトリ軸は同じ名称にできます。

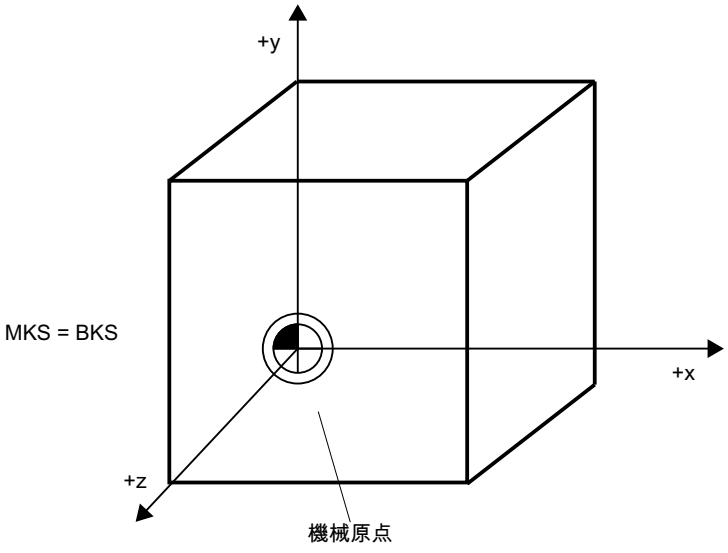


図 11-15 MCS=BCS (キネマティック座標変換なし)

キネマティック座標変換のある工作機械の場合

BCS がキネマティック座標変換により MCS に投影される場合、BCS と MCS は一致しません(たとえば、TRANSMIT /円筒補間、5 軸座標変換または 3 つ以上の軸)。

そのような機械上では、機械軸とジオメトリ軸は異なる名称にしなければなりません。

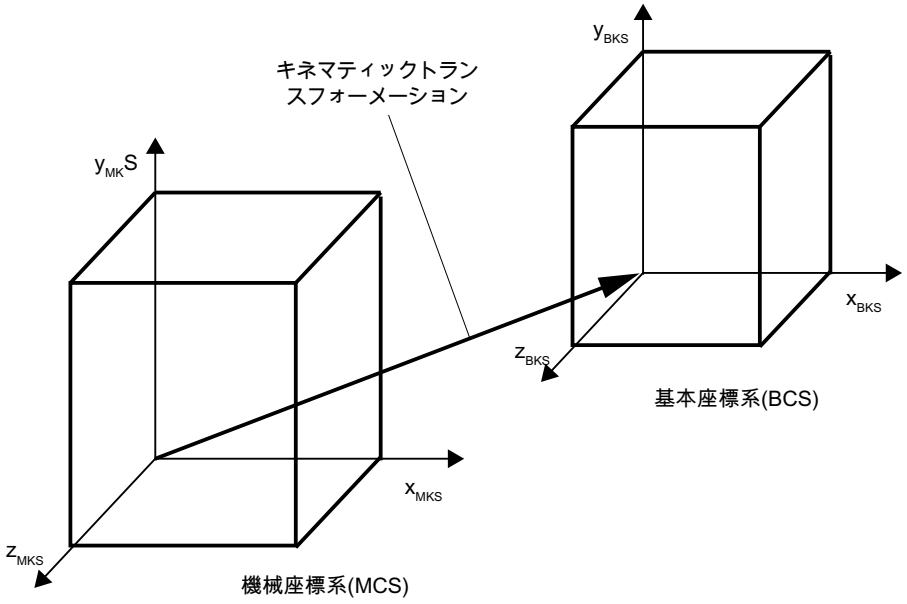


図 11-16　MCS と BCS の間のキネマティック座標変換

機械のキネマティックス

ワークは常に、2 次元または 3 次元の直交座標系(WCS)でプログラム指令されます。 しかし、そのようなワークは、互いに垂直でない回転軸または直線軸を装備した工作機械上でこれまで以上に頻繁にプログラム指令されています。 キネマティック座標変換は、ワーク座標系(直交)でプログラム指令された座標を実際の機械動作で表すために使用されます。

参照先:

『機能マニュアル、応用機能』；「3 ～ 5 軸座標変換(F2)」

『機能マニュアル 上級機能』；「キネマティック座標変換(M1)」

11.4.4 基本ゼロオフセットシステム(BZS)

基本ゼロオフセットシステム(BZS)

基本ゼロオフセットシステム(BZS)は、基本オフセットを使用した基本座標系です。

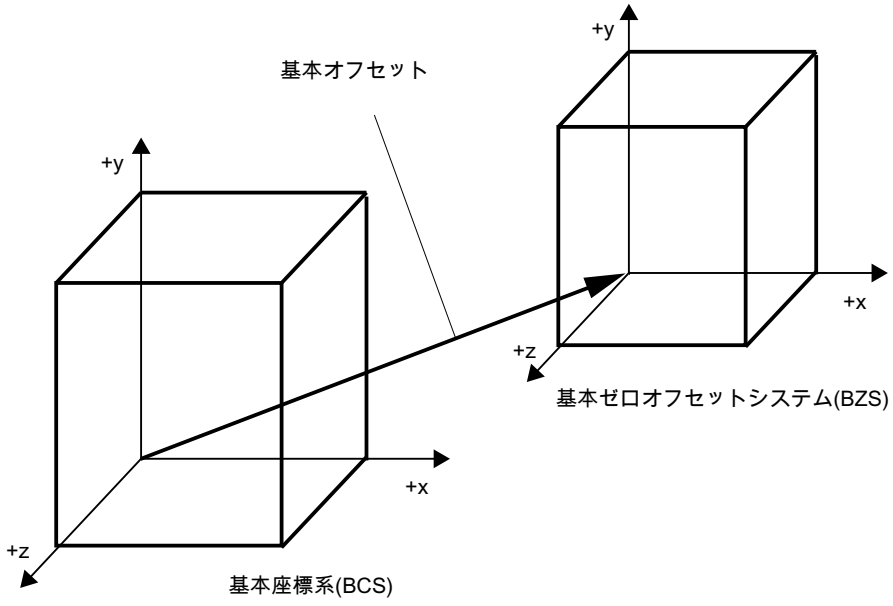


図 11-17　BCS と BZS の間の基本オフセット

基本オフセット

基本オフセットは、BCS と BZS の間の座標変換を記述します。たとえば、パレット原点の範囲を定義するために使用されます。

基本オフセットは、次の要素で構成されます。

- 外部ゼロオフセット
- DRF オフセットの章
- 重畳動作

- システムフレーム結合
- 基本フレーム結合

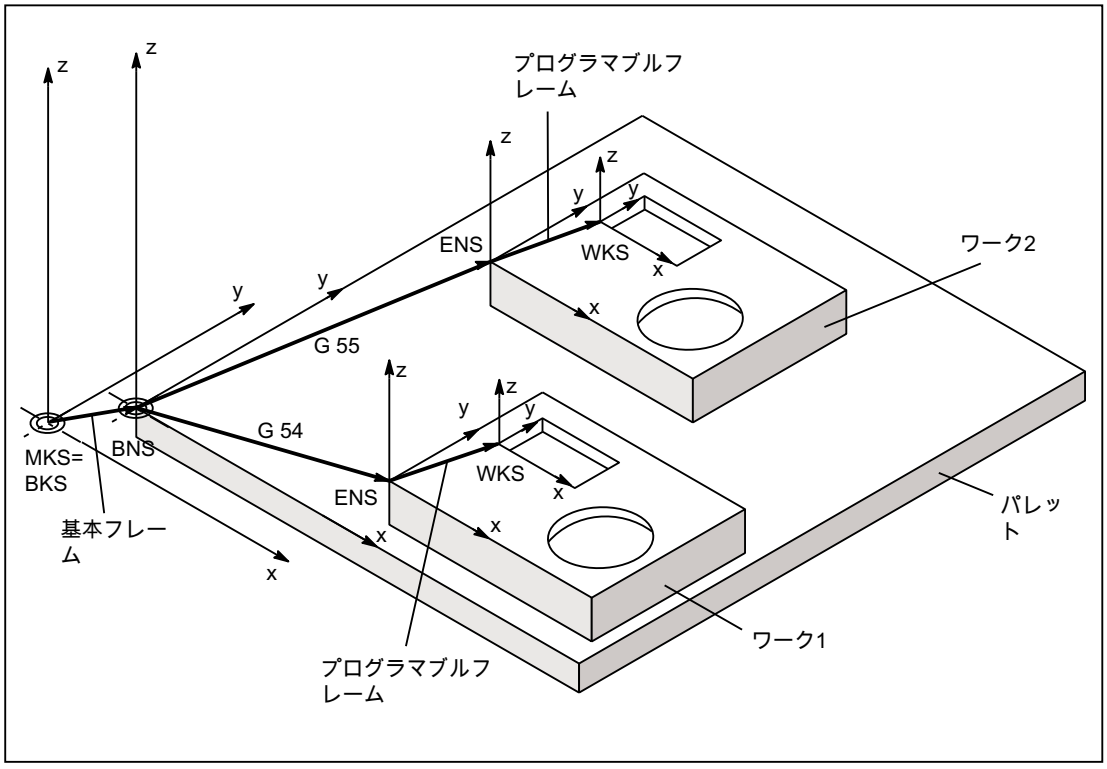


図 11-18　基本オフセットの使用例

これは以下のように適用されます。

- ユーザーはオペレータ操作、パートプログラム、および PLC から基本オフセットを変更できます。
- 基本オフセットをすぐに有効にしたい場合は、適切な G 命令を実行するために FC9 を使用して PLC を介して ASUP を起動できます。

注記

工作機械メーカへの推奨事項

メーカ独自の用途には第 3 基本オフセット以降を使用してください。

第 1 および第 2 基本オフセットは、PRESET および「外部ゼロオフセット」用に予約済です。

11.4.5 設定可能ゼロオフセットシステム(SZS)

設定可能ゼロオフセットシステム(SZS)

「設定可能ゼロオフセットシステム」(SZS)は、(WCS から見た場合に)プログラマブルフレームのあるワーク座標系 WCS です。ワーク原点は、設定可能な FRAMES G54 ～ G599 で定義します。

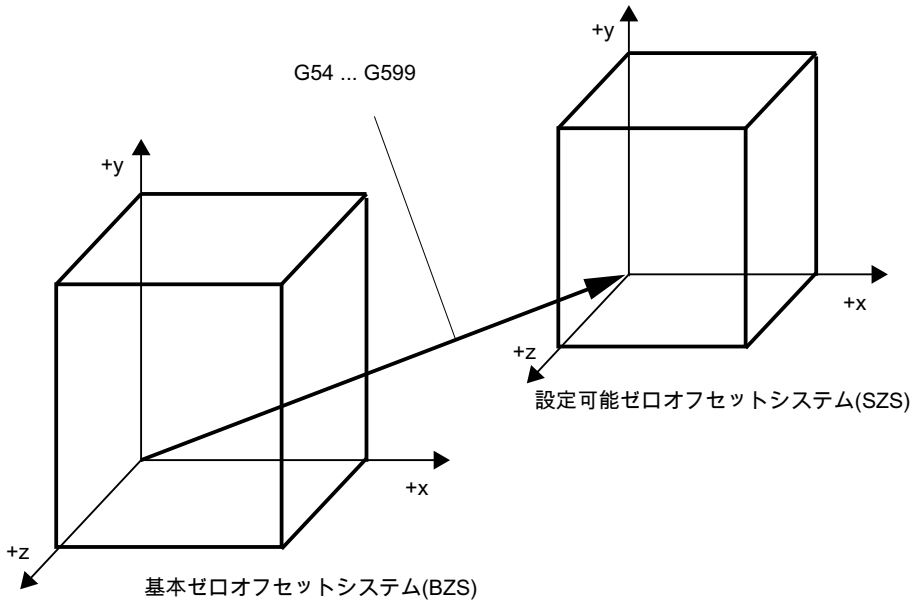


図 11-19 BZS と SZS の間の設定可能な FRAME G54 ... G599

プログラマブルオフセットは、「設定可能ゼロオフセットシステム」上で動作します。すべてのプログラマブルオフセットは、「設定可能ゼロオフセットシステム」を基準にします。

WCS または SZS における WCS の現在値の表示

機械座標系(MCS)または WCS 内の軸の現在値は、HMI 操作画面に表示できます。WCS での表示の場合、現在値は、SZS に対して表示することもできます。次のマシンデータを介して、対応するパラメータ設定を行います。

MD9424 $MM_MA_COORDINATE_SYSTEM (現在値表示用の座標系)

値	意味
0	WCS に対する現在値表示
1	SZS に対する現在値表示

注記

現在の座標系の表示

「SZS に対する現在値表示」が有効な場合も、WCS は、現在値表示に関する座標系として HMI 操作画面に表示されます。

例

WCS または SZS に対する現在値表示

コード(抜粋)	現在値表示: 軸 X (WCS)	現在値表示: 軸 X (SZS)
`N10 X100`	100	100
`N20 X0`	0	0
`N30 $P_PFRAME = CTRANS(X,10)`	0	10
`N40 X100`	100	110

11.4.6 ワーク座標系(WCS)

ワーク座標系(WCS)

ワーク座標系(WCS)はプログラミングの基礎となります。

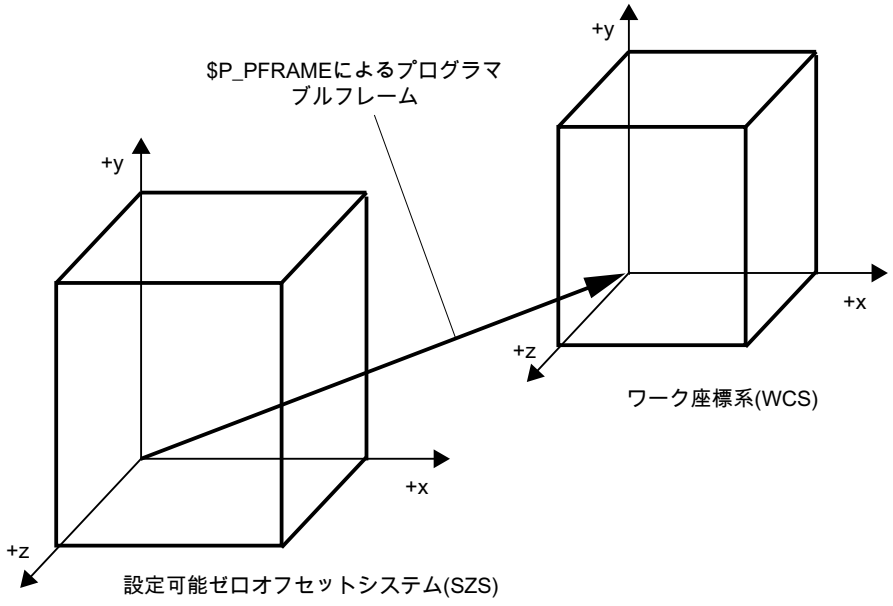
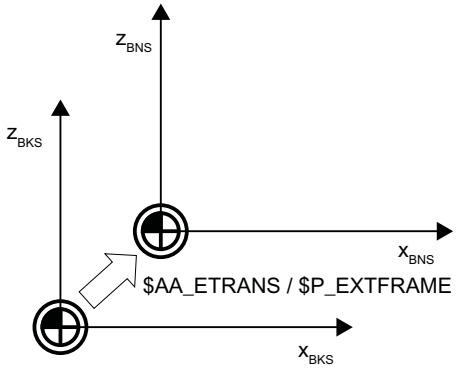


図 11-20 SZS と WCS の間のプログラム可能な FRAME

11.4.7 追加オフセット

11.4.7.1 外部ゼロオフセット

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 通りの方法で機能します。

1. システム変数$AA_ETRANS は、NC/PLC インターフェース信号による起動後にオフセット値として直接、機能します。
2. システム変数$AA_ETRANS の値は、NC/PLC インターフェース信号による起動後に、システムフレーム$P:EXTFRAME とデータ管理フレーム$P_EXTFR に取り込まれます。その後、動作中のフレーム一式$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、以下のマシンデータで選択した 2 つの処理が区別されます。

MD28082 $MC_MM_SYSTEM_FRAME_MASK、ビット 1 = <値>

<値>	意味
0	機能:PLC、HMI、または NC プログラムによる$AA_ETRANS[<軸>]の直接の書き込み 次の可能な移動ブロックで、$AA_ETRANS[<軸>]のゼロオフセットの移動を可能にします。DB31、...DBX3.0
1	機能:有効なシステムフレーム$P:EXTFRAME とデータ管理フレーム$P_EXTFR の起動 以下を使用して、$AA_ETRANS[<軸>]のゼロオフセットの移動を有効にします。DB31、...DBX3.0。チャネルで以下が実行されます。 ● チャネルのすべての移動動作を停止(コマンド軸と PLC 軸以外) ● 先読み停止とその後の再解析(STOPRE) ● 動作中のフレームの荒削りオフセット$P_EXTFRAME[<軸>] = $AA_ETRANS[<軸>] ● データ管理フレームの荒削りオフセット$P_EXTFR[<軸>] = $AA_ETRANS[<軸>] ● 動作中のフレーム一式$P_ACTFRAME の再計算 ● プログラム指令された軸でのオフセットの移動 ● 中断された移動動作または NC プログラムの再開

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一時的に記憶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

外部ゼロオフセットの起動:

DB31, ... DBX3.0 = 0 → 1 ⇒ $P_EXTFRAME[<軸>] = $P_EXTFR[<軸>] = $AA_ETRANS[<軸>]

抑制:外部ゼロオフセット

- SUPA 命令は、ブロックの処理中に外部ゼロオフセットをマスクします。
- G74 命令(レファレンス点復帰)と、それに相当する「レファレンス点復帰」モードでのオペレータ操作により、レファレンス点復帰の間、動作中の外部ゼロオフセットがマスクされます。
- G74、つまり「自動」または「MDI」モードでは、以前に動作中であった外部ゼロオフセットが再び、ブロック内の次の移動動作とともに自動的に起動されます
- レファレンス点復帰モードからのモード変更後、原点確立した軸の NC/PLC インターフェース信号を再起動のために設定してください。

11.4.7.2 DRF オフセット

DRF オフセットを使用すると、**ハンドルを介して**基本座標系内でジオメトリ軸と付加軸について追加インクリメンタル原点オフセットを調整できます。

システム変数

DRF オフセットは、以下の軸別システム変数から読み取ることができます。

$AC_DRF[<軸>]

参照先

機能マニュアル 上級機能; JOG 運転と手動ハンドル運転(H1):DRF オフセット

11.4.7.3 リセット応答

$P_EXTFRAME 有効フレームおよび$P_EXTFR データ管理フレームのリセット動作および電源投入動作は以下のマシンデータで設定できます。

マシンデータ

- チャネルで$P_EXTFRAME 外部ゼロオフセットに有効なシステムフレームのリセット動作は、以下のマシンデータで設定します。
- MD24006 $MC_CHSFRAME_RESET_MASK、ビット 1 = <値>

値	意味 チャネル/プログラムの最後のリセット後の、外部ゼロオフセットの有効なシステムフレームの状態:
0	無効
1	有効

- データ管理で$P_EXTFR 外部ゼロオフセットのチャネル別システムフレームの電源投入動作は、以下のマシンデータで設定します。
  MD24008 $MC_CHSFRAME_RESET_MASK、ビット 1 = <値>

値	意味 電源投入の外部ゼロオフセットの$P_EXTFR システムフレーム
0	解除されない
1	解除

11.4.8 軸別重畳($AA_OFF)

11.4.8.1 機能

たとえば、軸別システム変数$AA_OFF[ ]では、シンクロナイズドアクションで指定された軸のアブソリュート位置またはインクリメンタル距離を指定できます。これによって得られる移動操作は、軸のチャネルの移動動作と並列に実行されます。

$AA_OFF[<軸>] = <値>

11.4.8.2 セットアップ

マシンデータ

重畳モーションのパラメータ設定

軸マシンデータで、重畳動作の以下の設定を行います:

36750 $MA_AA_OFF_MODE、ビット<n> = <軸>

ビット	値	意味
0	0	**アブソリュート位置**としての値$AA_OFF の解釈
	1	**インクリメンタルパス**としての値$AA_OFF の解釈
1	0	重畳動作は、チャネルリセットで解除されます。
	1	重畳動作はチャネルリセット後も維持されます。
2	0	JOG モードで、重畳動作は後退**しません**。
	1	JOG モードで、重畳動作は後退します。
3	0	重畳動作が NC ストップのために中断します。
	1	重畳動作が NC ストップのために**中断しません**。

システム変数

軸重畳の統合パス

重畳動作の統合された値は、軸別システム変数から読み取ることができます。

<value> = $AA_OFF_VAL[<axis>]

11.4.8.3 プログラミング軸別の重畳の解除(CORROF)

CORROF 手順により、以下の軸別重畳が解除されます:

- ハンドル移動により設定された追加ゼロオフセット(DRF オフセット)
- $AA_OFF システム変数によってプログラムされた位置オフセット
- $AC_OFF_...システム変数によってプログラムされた工具オリエンテーションの重畳

先読み停止は、重畳値の解除によって開始され、解除された重畳動作の位置成分が基本座標系の位置に伝送されます。それによって、移動する軸はありません。

$AA_IM システム変数(軸の現在の **MCS** 指令)によって読み取ることができる位置データは、**機械**座標系で**変化しません**。

重畳動作の解除された成分が含まれるようになったために、$AA_IW システム変数(軸の現在の **WCS** 指令)によって読み取ることができる位置データは**ワーク**座標系では**変化**します。

注記

CORROF は、**NC プログラム**でプログラムできます。
CORROF は、**シンクロナイズドアクション**ではプログラム**しないでください**。

構文

```
CORROF(<Axis>,"<String>"[,<Axis>,"<String>"])
```

意味

CORROF:	軸の以下のオフセットおよび重畳の解除の手順: • DRF オフセット • 位置オフセット($AA_OFF) • 工具オリエンテーションの重畳($AC_OFF_...)	
	有効性:	モーダル
<軸>:	軸識別子(チャネル、ジオメトリ軸、または機械軸の識別子)	
	データタイプ:	AXIS
<String>:	重畳タイプを定義する文字列	
	データタイプ:	BOOL
	値	**意味**
	DRF	DRF オフセット
	AA_OFF	位置オフセット($AA_OFF)
	OFF_ORI	工具オリエンテーションの重畳($AC_OFF_...) **注** 工具オリエンテーションの重畳の解除は、旋回軸の軸別オフセットを解除することにより実行されます。**<Axis>**パラメータとして、**任意**のチャネル軸を指定できます。

例

例 1:DRF オフセットの軸別解除(1)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸で生成されています。チャネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
N10 CORROF(X,"DRF")	; ここでは CORROF は、DRFOF と同じ働きがあります。
...	

例 2 :DRF オフセットの軸別解除(2)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸と Y 軸で生成されています。チャネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
; X 軸の DRF オフセットのみが選択解除されます。Y 軸の DRF オフセットは保持されます	
; DRFOF では、両方のオフセットが解除されます。	
N10 CORROF(X,"DRF")	
...	

例 3 :$AA_OFF 位置オフセットの軸別解除

プログラムコード	コメント
; 位置オフセット== 10 が X 軸に対して補間されます。	
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	
...	
; X 軸の位置オフセットの選択解除: $AA_OFF[X]=0	
; X 軸は移動しません。	
; 位置オフセットを、X 軸の現在位置に加算します。	
N80 CORROF(X,"AA_OFF")	
...	

例 4 :DRF オフセットと$AA_OFF 位置オフセットの軸別選択解除 (1)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸で生成されます。チャネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
; 位置オフセット 10 が X 軸に対して補間されます。	
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	
...	
; X 軸の DRF オフセットと位置オフセットのみを選択解除します。	
; Y 軸の DRF オフセットは保持されます。	
N70 CORROF(X,"DRF",X,"AA_OFF")	

プログラムコード	コメント
...	

例 5 :DRF オフセットと$AA_OFF 位置オフセットの軸別選択解除 (2)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸と Y 軸で生成されます。チャネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
; 位置オフセット == 10 が X 軸に対して補間されます。	
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	
...	
; Y 軸の DRF オフセットと X 軸の位置オフセットのみを選択解除します。	
; X 軸の DRF オフセットは保持されます。	
N70 CORROF(Y,"DRF",X,"AA_OFF")	
...	

詳細情報

$AA_OFF_VAL

位置オフセットを$AA_OFF で選択解除すると、対応する軸のシステム変数 $AA_OFF_VAL (軸重畳の統合距離)はゼロになります。

JOG モードの$AA_OFF

JOG モードでも、$AA_OFF が変更されると、マシンデータ MD 36750 $MA_AA_OFF_MODE でこの機能を有効にしている場合は、位置オフセットは重畳移動として補間されます。

シンクロナイズドアクションの$AA_OFF

位置オフセットを CORROF(<軸>,"AA_OFF") で選択解除時に、$AA_OFF (DO $AA_OFF[<軸>]=<値>)を直ちにリセットするシンクロナイズドアクションが有効である場合、$AA_OFF は選択解除されますが、リセットされません。また、アラーム 21660 が表示されます。ただし、CORROF の後のブロックなど、後の方のブロックでシンクロナイズドアクションが有効になる場合は、$AA_OFF が設定されたままとなり、位置オフセットは補間されます。

自動チャネル軸入れ替え

別のチャネルで有効な軸が CORROF のプログラム指令対象である場合、その軸は、軸入れ替え時にそのチャネルに取り込まれ(条件:MD30552 $MA_AUTO_GET_TYPE > 0)、位置オフセットか DRF オフセットのいずれか、または両方が選択解除されます。

11.5 フレーム

11.5.1 フレームタイプ

フレームは、軸のオフセット(TRANS)、仕上げオフセット(FINE)、回転(ROT)、 ミラーリング(MIRROR)、スケーリング(SCALE)の値の入ったデータ構造体です。

フレーム値を使用してフレームを有効にする場合、フレームに含まれている軸の内部的な座標変換は定義済みのアルゴリズムを使用しておこなわれます。

軸別フレーム

軸別フレームには、軸のフレーム値が含まれています。

ジオメトリ軸 X の軸別フレームのデータ構造体の例:

軸	TRANS	FINE	ROT	MIRROR	SCALE
X	10.0	0.1	**0.0**	0	1

チャネル別フレーム

チャネル別フレームには、すべてのチャネル軸(ジオメトリ軸、付加軸、機械軸)のフレーム値が入っています。

回転(ROT)は、ジオメトリ軸の計算にのみ含まれています。

チャネル別フレームは、フレームが定義されているチャネルでのみ有効です。

チャネル別フレームのデータ構造体の例

- ジオメトリ軸:X、Y、Z
- 付加軸:A
- 機械軸:AX1

軸	TRANS	FINE	ROT	MIRROR	SCALE
X	10.0	0.1	**0.0**	0	1
y	0.0	0.0	**0.0**	1	1
Z	0.0	0.0	**45.0**	0	1
A	2.0	0.1	0.0	0	2
AX1	0.0	0.0	0.0	0	0

グローバルフレーム

グローバルフレームには、すべての機械軸のフレーム値が入っています。

グローバルフレームは、NC のすべてのチャネルで有効です。

グローバルフレームのデータ構造の例:

- 機械軸:AX1、... AX5

軸	TRANS	FINE	ROT	MIRROR	SCALE
AX1	10.0	0.1	-	0	1
AX2	0.0	0.0	-	1	1
AX3	0.0	0.0	-	0	1
AX4	2.0	0.1	-	0	2
AX5	0.0	0.0	-	1	1

11.5.2 フレームコンポーネント

11.5.2.1 座標移動

プログラミング

座標移動または荒削りオフセットのプログラミングは、次の命令でおこなうことができます。

- データ管理フレーム$P_UIFR の例
  - フレーム全体: $P_UIFR[<n>] = CTRANS(<K1>,<V1>[,<K2>,<V2>][,<K3>,<V3>])
    、Km = 座標 x、y、または z および Vm = オフセット m
  - フレーム成分: $P_UIFR[<n>,<k>,TR] = <V>
    、K = 座標 x、y、または z および V = オフセット
- プログラマブルフレームの例
  - TRANS <K1> <V1> [<K2> <V2>][<K3> <V3>]
    、Km = 座標 x、y、または z および Vm = オフセット m

プログラム例:

プログラムコード	備考
$P_UIFR[1] = CTRANS(x,10,y,10)	フレーム全体

プログラムコード	備考
$P_UIFR[1,X,TR] = 10	フレーム成分
TRANS x=10 y=10	プログラマブルフレーム

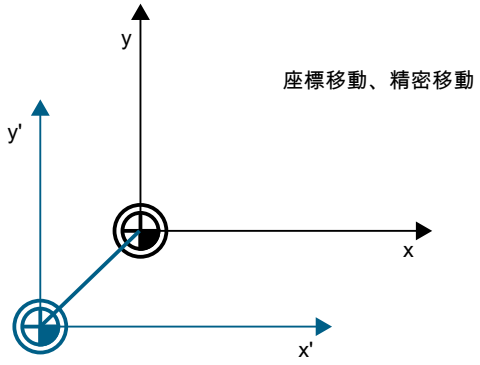


図 11-21　Z 方向のオフセット

11.5.2.2　仕上げオフセット

パラメータ設定

仕上げオフセットは、以下のマシンデータで有効になります。

MD18600 $MN_MM_FRAME_FINE_TRANS = <値>

値	意味
0	仕上げオフセットは、入力またはプログラム指令できません。
1	命令またはプログラムを介して設定可能フレーム、基本フレームおよびプログラマブルフレーム用の仕上げオフセットが可能です。

プログラミング

座標移動または荒削りオフセットのプログラミングは、次の命令でおこなうことができます。

- データ管理フレーム$P_UIFR の例
  - フレーム全体: $P_UIFR[<n>] = CFINE(<K1>,<V1>[,<K2>,<V2>][,<K3>,<V3>])
    、Km = 座標 x、y、または z および Vm = オフセット m
  - フレーム成分: $P_UIFR[<n>,<K>,FI] = <V>
    、K = 座標 x、y、または z および V = オフセット
- プログラマブルフレームの例
  - TRANS <K1> <V1> [<K2> <V2>][<K3> <V3>]
    、Km = 座標 x、y、または z および Vm = オフセット m

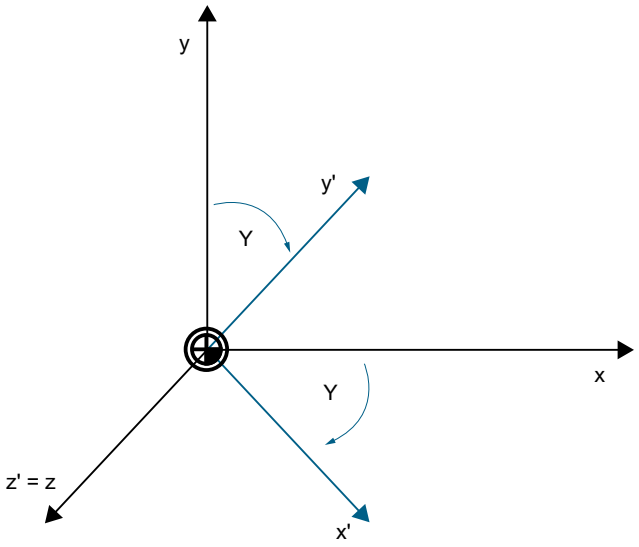
プログラミング例:

プログラムコード	備考
$P_UIFR[1] = CTRANS(x,10,y,10)	フレーム全体
$P_UIFR[1,X,TR] = 10	フレーム成分
TRANS x=10 y=10	プログラマブルフレーム

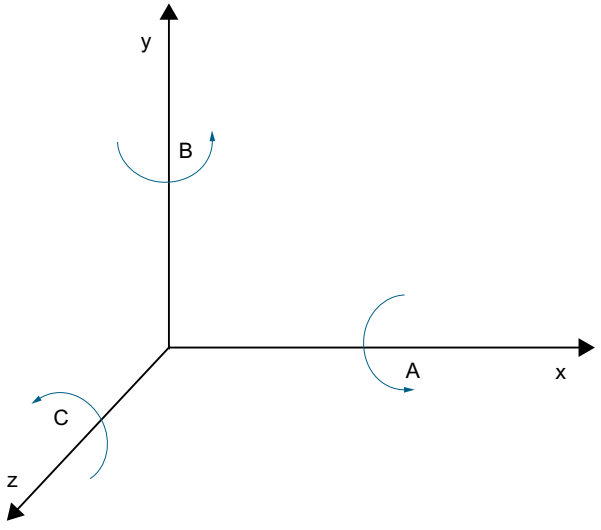
11.5.2.3 回転の概要(ジオメトリ軸のみ)

機能

座標軸を中心とする回転方向は、軸 X、Y、および Z の右手直交座標系で決定されます。各座標軸を正方向に向かって見た場合に、回転動作が右回りの方向が、正の回転方向です。 A、B、および C により、座標軸に平行な軸の回転が指定されます。



z を中心にした γ = -45°の回転後の座標系 x、y、および z'の新しい位置を、下の図に示します。



回転順序のパラメータ設定

次のマシンデータを使用して、複数の回転角度がプログラム指令された場合の回転の中心となる座標軸と回転順序を設定します。

MD10600 $MN_FRAME_ANGLE_INPUT_MODE = <値>

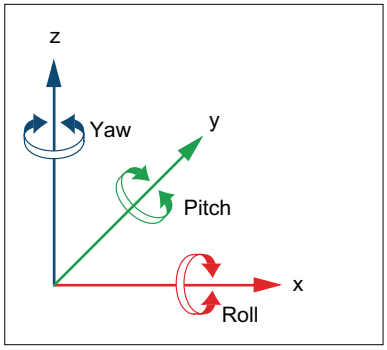
値	意味
1	zy'x"形式でのオイラー角(RPY 角)
2	zx'z"形式でのオイラー角

注記

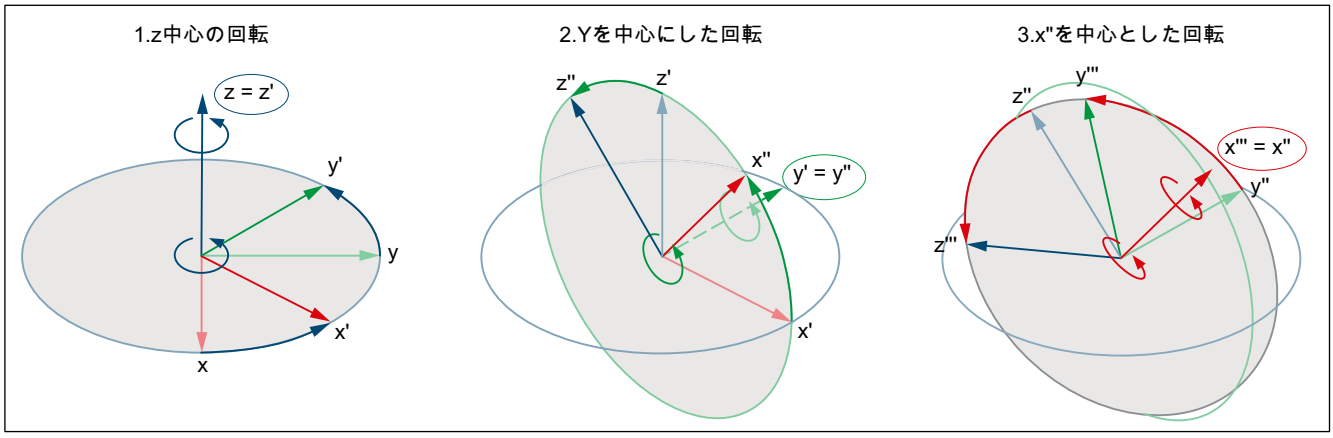
互換性の理由により、zx'z"形式のオイラー角が使用可能です。 ただし、zy'x"形式でのオイラー角(RPY 角)だけを使用することを強くお勧めします(「オイラー角による回転: ZY'X"形式(RPY 角) (ページ 902)」の章を参照してください)。

11.5.2.4 オイラー角による回転: ZY'X"形式(RPY 角)

ZY'X"形式のオイラー角は RPY 角とも呼ばれます。RPY は、ロール、ピッチ、およびヨーから算出されます:



ZY'X"表記規則では、回転は次の順序で実行されます。



値の範囲

RPY 角により、プログラム指令値は以下の値範囲内でのみ一義的に逆算できます。

-180	≤	**x**	≤	180
-90	<	**y**	<	90
-180	≤	**z**	≤	180

プログラミングすべての回転成分の書き込み

CROT、ROT、または AROT を使用して、フレームの回転成分をプログラム指令すると、すべての回転成分が常に書き込まれます。明確にプログラム指令されていないコンポーネントを回転すると、0 °の値が割り当てられます。

構文

```
<フレーム> = CROT([<1 番目の GAx>,<角度>,][<2 番目の GAx>,<角度>,][<3 番目の GAx>,<角度>])
ROT [<1 番目の GAx> <角度>] [<2.GAx> <角度>] [<3.GAx> <角度>]
AROT [<1 番目の GAx> <角度>] [<2.GAx> <角度>] [<3.GAx> <角度>]
```

意味

CROT:	アブソリュート回転
<フレーム>:	任意の有効なフレームまたはデータ管理フレーム
ROT:	アブソリュート回転 基準フレーム:プログラマブルフレーム$P_PFRAME、 レファレンス点:G54 ... G57、G505 ... G599 で設定された現在のワーク座標系の原点

AROT:	追加回転 基準フレーム:プログラマブルフレーム$P_PFRAME、 レファレンス点:G54 ... G57、G505 ... G599 で設定された現在のワーク座標系の原点
<n 番目の GAx>:	指定された角度で行われる回転の中心となる n 番目のジオメトリ軸の名称。ジオメトリ軸の回転の角度がプログラム指令されていない場合は、値 0°が自動的に設定されます。 回転軸に対するジオメトリ軸の割り当て

ジオメトリ軸	回転軸
1 番目のジオメトリ軸	x''
2 番目のジオメトリ軸	y'
3 番目のジオメトリ軸	z

<角度>:	角度指定(°単位)
[...]:	角括弧内のデータはオプションです。

プログラミング回転成分への書き込み

フレームの回転成分を明確にプログラムした場合、プログラムされた回転成分のみが書き込まれます。プログラムされていない回転成分は、変更されないままです。

構文
<フレーム>[<インデックス>,<GAx>,RT] = <角度>

意味

<フレーム>:	任意の有効なフレームまたはデータ管理フレーム
<インデックス>:	フレームの配列インデックス、例: $P_UIFR[0 ... n]
<GAx>:	指定された角度で行われる回転の中心となるジオメトリ軸の名称。
RT:	回転「RoTation」のキーワード
<角度>	角度指定(°単位)

回転成分の再読み込み

一般的には、フレームの回転成分をプログラム指令された成分として再読み込みすると、同じ値が取得されます。

プログラム指令	再読み取り時の値		
	x, RT	y, RT	z, RT
<フレーム> = CROT(X,45,Y,30,Z,-20)	45	30	-20

値範囲を超えた値

値範囲を超えてプログラム指令値は、範囲の限界値に割り当てられます。

プログラム指令	再読み取り時の値		
	x, RT	y, RT	z, RT
<フレーム> = CROT(X,190,Y,0,Z,-200)	-170	0	160

注記

フレームの回転成分を書き込む場合は、回転成分を再読み込みしたときに同じ値が得られるように、指定された値範囲を保持することを推奨します。

ジンバルロック

ジンバルロックは、回転成分を位置ベクトルから一義的に逆算できなくなる幾何学的な問題を意味します。ジンバルロックは、回転成分 y の回転位置= 90°の RPY 角で発生します。この場合、回転成分は、以下が適用されるように書き込み後にコントロールシステムで変換されます。

- 回転成分 z =回転成分 z -回転成分 x
- 回転成分 x = 0°
- 回転成分 y = 90°

プログラム指令	再読み取り時の値		
	x, RT	y, RT	z, RT
<フレーム> = CROT(X,30,Y,90,Z,40)	0	90	40 - 30 = 10

⚠ 注意

回転成分 z の異なる再読み込み値

すべてのフレームの書き込みまたは**データ管理フレーム**の**個々の回転成分**の書き込みと、**動作中のフレーム**の**個々の回転成分**の書き込み後の変換のタイミングが異なるために、回転成分 z に対して再読み込み値が異なる場合があります。

すべてのフレームを書き込む場合とフレーム成分を書き込む場合の違い

フレームの回転成分を書き込む場合、以下の 2 つのケースを区別してください。

1. すべてのフレームの書き込み: <フレーム> = CROT(X,a,Y,b,Z,c)
   すべてのフレームを書き込む場合は、書き込み時にすぐに変換されます。
2. X を中心にした回転など、個々の回転成分を書き込む: <フレーム>[0,X,RT]=<a>
   個々の回転成分を書き込む場合、変換はフレームの保存場所によって異なります。
   - データ管理フレーム
     データ管理フレームでは、これまでに書き込まれた回転成分に基づいて、フレームの起動時に変換がおこなわれます。そのため、データ管理フレームの変換に関しては、データ管理フレームは、個々の回転成分の書き込み後はすべてのフレームの書き込み時と同じように動作します。
   - 動作中のフレーム
     動作中のフレームの場合は、回転成分の書き込み時にすぐに変換がおこなわれます。

例

- すべてのフレームの書き込み
  変換は、**すべてのフレーム**が書き込まれた後に**各ブロック**で行われます。

プログラム指令	再読み取り時の値		
	x, RT	y, RT	z, RT
N10 <フレーム> = CROT(X,0,Y,90,Z,90)	0	90	90
N20 <フレーム> = CROT(X,90,Y,90)	0	90	**-90** 1)
N30 <フレーム> = CROT(X,90,Y,90,Z,90)	0	90	**0** 1)
1) **動作中のフレーム**の**個々の回転成分**の書き込みと異なる値			

- データ管理フレームの**個々の回転成分**の書き込み
  変換は、**データ管理フレームを起動する**と実行されます。この例では、N30 の後の任意のタイミングです。

プログラム指令	再読み取り時の値		
	x, RT	y, RT	z, RT
N10 <データ管理フレーム>[0,X,RT] = 0 N20 <データ管理フレーム>[0,Y,RT] = 90 N30 <データ管理フレーム>[0,Z,RT] = 90	0	90	90
N10 <データ管理フレーム>[0,X,RT] = 90 N20 <データ管理フレーム>[0,Y,RT] = 90 N30 <データ管理フレーム>[0,Z,RT] = 0	0	90	**-90** 1)
N10 <データ管理フレーム>[0,X,RT] = 90 N20 <データ管理フレーム>[0,Y,RT] = 90 N30 <データ管理フレーム>[0,Z,RT] = 90	0	90	**0** 1)
1) **動作中のフレーム**の**個々の回転成分**の書き込みと異なる値			

- 動作中のフレームの**個々の回転成分**の書き込み
  必要な**変換**は、**回転成分を書き込み次第直ちに**行われます。
  格納される有効なフレームの初期値:x = 0、y = 0、z = 0

プログラム指令	再読み取り時の値		
	x, RT	y, RT	z, RT
N10 <有効なフレーム>[0,X,RT] = 0 N20 <有効なフレーム>[0,Y,RT] = 90 N30 <有効なフレーム>[0,Y,RT] = 90	0	90	90
N10 <有効なフレーム>[0,X,RT] = 90 N20 <有効なフレーム>[0,Y,RT] = 90 N30 <有効なフレーム>[0,Y,RT] = 0	0	90	**0** 1)
N10 <有効なフレーム>[0,X,RT] = 90 N20 <有効なフレーム>[0,Y,RT] = 90 N30 <有効なフレーム>[0,Y,RT] = 90	0	90	**90** 1)
1) **すべてのフレーム**の書き込みまたは**データ管理フレーム**の**個々の回転成分**の書き込みと異なる値			

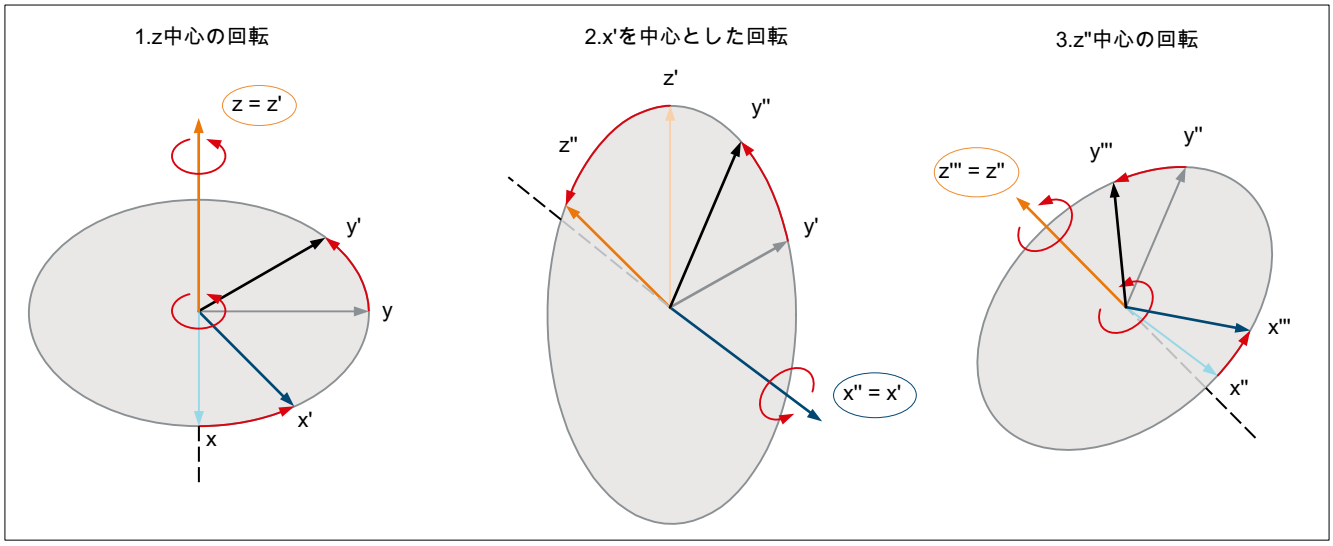
11.5.2.5 オイラー角による回転: ZX'Z"形式

オイラー角では、z、x'、z"の順序で回転されます。

注記

推奨される使用法

互換性の理由から、zx'z"形式のオイラー角を使用できますが、zy'x"形式のオイラー角(RPY角)のみを使用することを強く推奨します(「オイラー角による回転: ZY'X"形式(RPY 角) (ページ 902)」の章を参照してください)。



ジオメトリ軸への回転軸の割り当て

回転軸	チャネルのジオメトリ軸
z	3 番目のジオメトリ軸
x'	1 番目のジオメトリ軸
z"	3 番目のジオメトリ軸

値の範囲

オイラー角のデータは、以下の値範囲内でのみ一義的に逆算できます。

```
  0    <=   x   <    180
-180   <=   y   <=   180
-180   <=   z   <=   180
```

データが指定された値範囲を超える場合、特定の範囲限界値を基準としてモジュロ変換がおこなわれます。

注記

フレームの回転成分を書き込む場合は、回転成分を再読み込みしたときに同じ値が得られるように、指定された値範囲を保持することを推奨します。

11.5.2.6 任意の平面の回転

CRPL - 一定回転平面

事前に定義された機能「**C**onstant **R**otation **Pl**ane (一定回転平面)」を使用して、ジオメトリ軸の名称の指定なしで、任意の平面(G17、G18、G19)のフレームに回転をプログラムできます。 これにより、特定の機械座標のためにチャネル内に 2 つのジオメトリ軸しか存在しない場合でも、3 番目の平面で回転をプログラム指令できます。

構文

```
CRPL(<回転軸>,<回転角度>)
```

意味

CRPL: 任意の平面の回転

<回転軸>: 回転の中心となる軸

タイプ: INT

値	意味
0	有効な平面内での回転
1	Z を中心にした回転
2	Y を中心にした回転
3	X を中心にした回転

<回転角度>: 回転がおこなわれる角度(°)

タイプ: REAL

指定された角度の範囲を守ることを強くお勧めします。 制限値を守らない場合、一義的な計算がおこなえません。 制限値を超えた角度は拒否されます。

RPY 角:	X	-180 <= <回転角度> <= 180
	Y	-90 <= <回転角度> <= 90
	Z	-180 <= <回転角度> <= 180
ZX'Z"形式:	X	-180 <= <回転角度> <= 180
	Y	0 <= <回転角度> <= 180
	Z	-180 <= <回転角度> <= 180

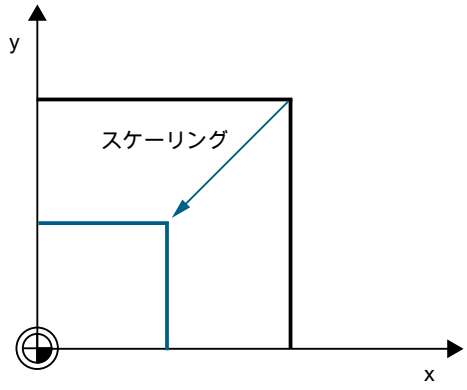
フレームとの結合

CRPL()は、フレームと既知のフレーム機能(CTRANS()、CROT()、CMIRROR()、CSCALE()、CFINE()など)と結合できます。

例:

```
$P_PFRAME = $P_PFRAME : CRPL(0,30.0)
$P_PFRAME = CTRANS(X,10) : CRPL(1,30.0)
$P_PFRAME = CROT(X,10) : CRPL(2,30.0)
$P_PFRAME = CRPL(3,30.0) : CMIRROR(Y)
```

11.5.2.7 スケーリング

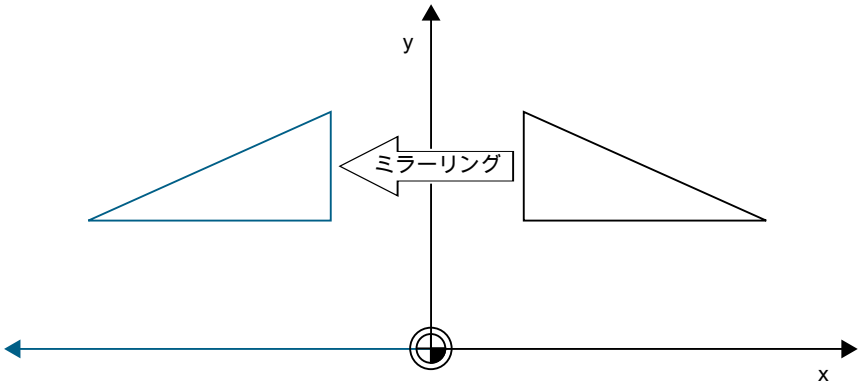


プログラミング

次のプログラム命令は、スケーリングのプログラムに使用されます。

```
$P_UIFR[1] = CSCALE(x,1,y,1)
SCALE x = 1y = 1
$P_UIFR[1,x,sc] = 1
```

11.5.2.8 ミラーリング



プログラミング

次のプログラム命令は、ミラーリングのプログラムに使用されます。

```
$P_UIFR[1] = CMIRROR(x,1,y,1)
MIRROR x = 1y = 1
```

```
$P_UIFR[1,x,mi] = 1
```

11.5.2.9 結合演算子

結合演算子(:)を使用して、フレームコンポーネントまたはすべてのフレームを 1 つの総合フレームに結合できます。).

11.5.2.10 プログラム可能な軸名称

ジオメトリ軸、チャネル軸および機械軸の名称は、フレーム命令で使用できます。 プログラムされた軸は、チャネル内のチャネル別フレームに認識されていなければなりません。

SPI

フレーム命令をプログラミングする場合は、軸名称の代わりに SPI(<spindle number>) 軸機能を使用できます。

SPI(<spindle number>) は主軸とチャネル軸の関係を表します。
→ MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[ ] (主軸の機械軸への割り当て)を参照してください。

次のフレーム命令は、SPI(spino) と共にプログラムできます。

CTRANS()

CFINE()

CMIRROR()

CSCALE()

主軸は、1 度に 1 つの回転軸にしか割り当てることができません。 したがって、CROT() 機能に使用できるのはジオメトリ軸だけであるため、CROT(..) を SPI() と共にプログラムすることはできません。

軸名称が SPI(..) を使用してパートプログラム内にプログラムされている場合でも、フレームの逆コンパイル時には、主軸に属する軸のチャネル軸名称または機械軸名称が常に出力されます。

たとえば、チャネル軸 A に主軸が割り当てられて、次のようにプログラミングされている場合、

```
N10 $P_UIFR[1] = CTRANS(SPI(1),33.33,X,1):CSCALE(SPI(1),
33.33):CMIRROR(SPI(1))
```

逆コンパイルすると、次のようになります。

```
$P_UIFR[1]=CTRANS(X,1,A,33.33):CSCALE(A,33.33):CMIRROR(A)
```

フレーム命令内で主軸と割り当てられる軸がプログラムされている場合、アラーム 16420「軸%が多重にプログラムされています」が出力されます。

例:

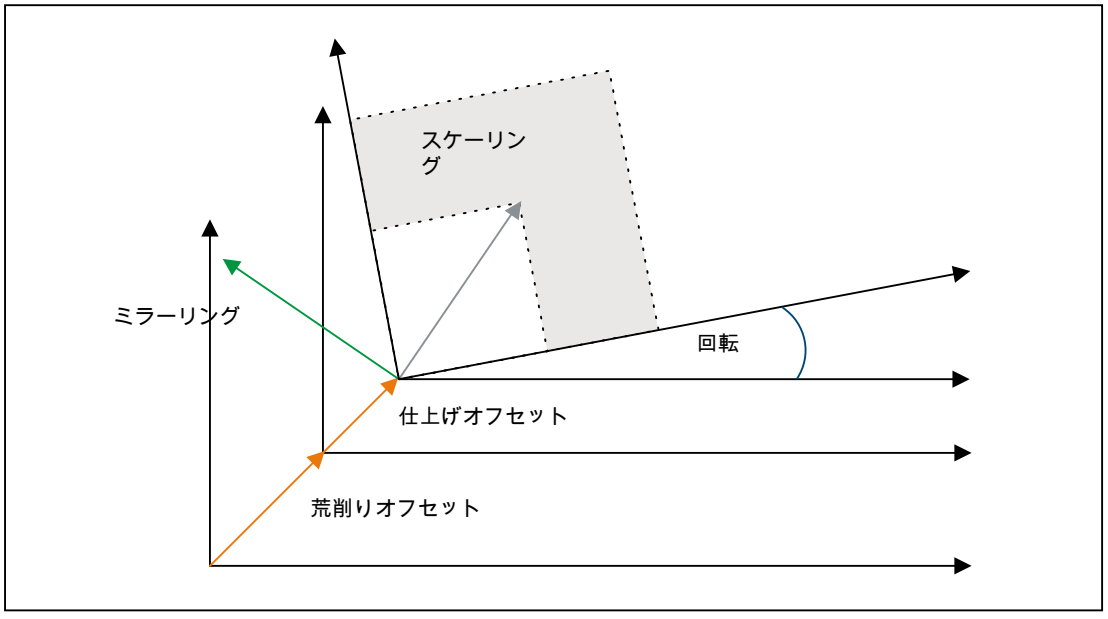
```
$P_UIFR[1] = CTRANS(SPI(1),33.33,X,1,A,44)
```

(主軸は軸 A に割り当てられています。)

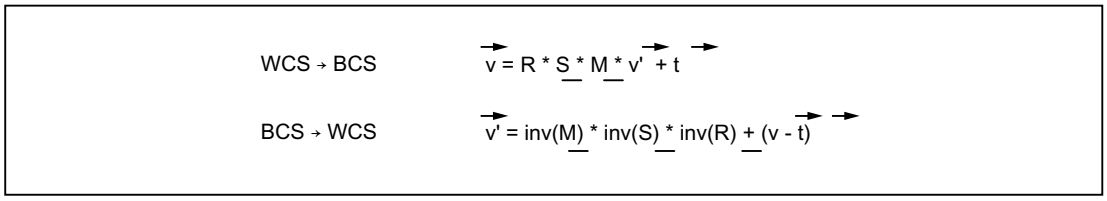
プログラミング例

```
$P_PFRAME[SPI(1),TR]=22.22
$P_PFRAME=CTRANS(X,軸の値,Y,軸の値,SPI(1),軸の値)
$P_PFRAME=CSCALE(X,倍率,Y,倍率,SPI(2),倍率)
$P_PFRAME = CMIRROR (S1, Y, Z)
$P_UBFR=CTRANS(A,10):CFINE(SPI(1),0.1)
```

11.5.2.11 座標変換



次の公式は、ジオメトリ軸に対する座標変換を計算するために使用されます。



V: BCS での位置ベクトル

V': WCS での位置ベクトル

11.5.3 データ管理フレームと有効フレーム

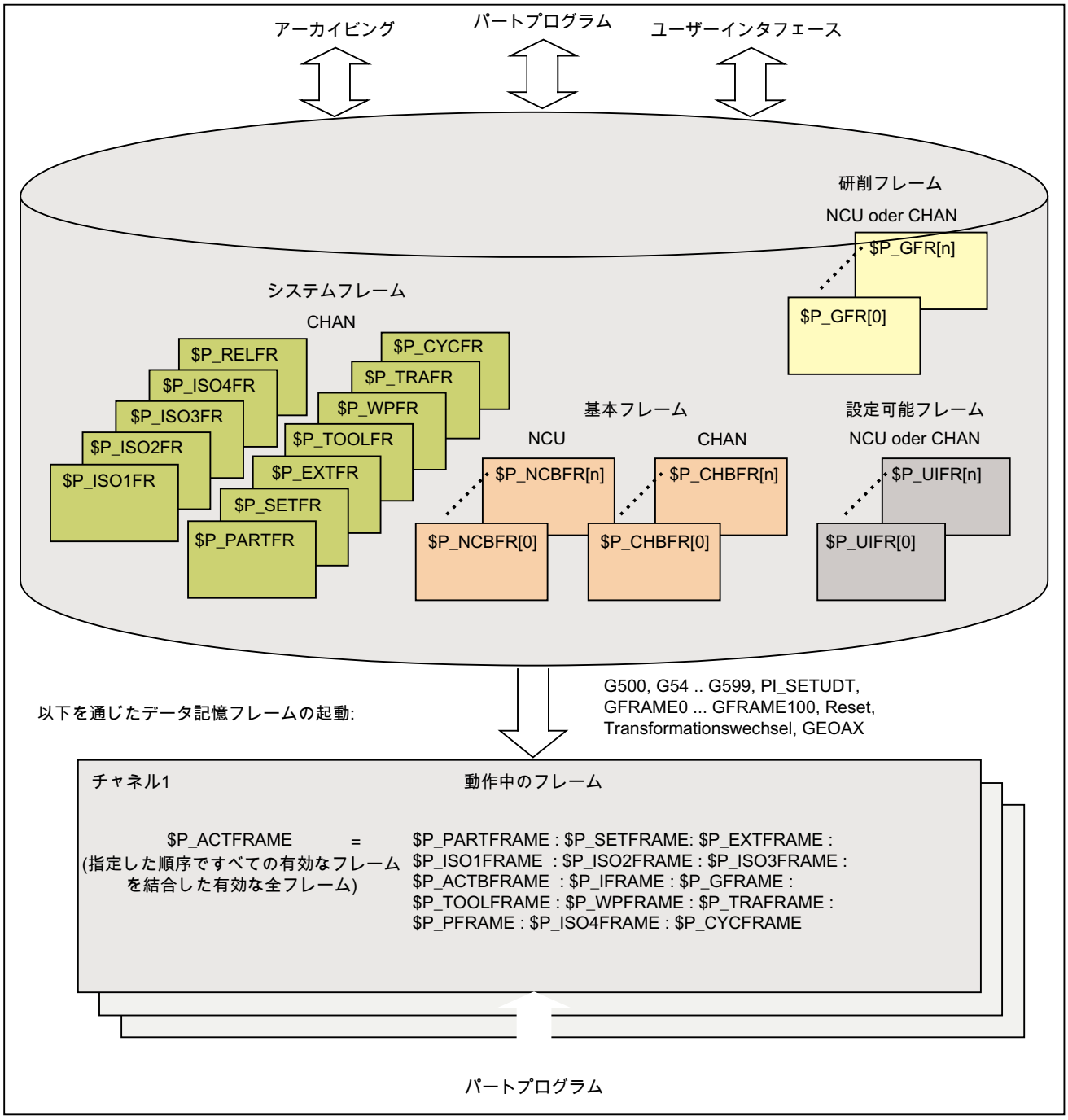
11.5.3.1 概要

フレームタイプ

次のフレームタイプが使用可能です。

- システムフレーム ($P_PARTFR、... 図を参照)
- 基本フレーム ($P_NCBFR[<n>], $P_CHBFR[<n>])
- 研削フレーム ($P_GFR[<n>])
- 設定可能フレーム($P_UIFR[<n>])
- プログラマブルフレーム$P_PFRAME

プログラマブルフレームを除くすべてのフレームタイプでは、チャネルで有効なフレームに加え、1 つ以上のフレームがデータ管理(データ管理フレーム)に存在します。プログラマブルフレームの場合は、チャネルで有効なフレームのみが存在します。



フレームの書き込み

データ管理フレームと有効なフレームは、パートプログラムから書き込むことができます。ユーザーインタフェースを介しては、データ管理フレームだけが書き込めます。

フレームのアーカイブ

データ管理フレームのみがアーカイブできます。

11.5.3.2 データ管理フレームの起動

以下の操作の結果、データ管理フレームが有効なフレームになります。

- 「設定可能フレーム」G グループ:G54 ... G57、G500、G505 ... G599
- 「検索フレーム」G グループ:GFRAME0 ... GFRAME100
- RESET および MD20110 $MC_RESET_MODE_MASK、ビット 14 == 1 (基本フレームの現在の設定が保持されます)
- 座標変換の切り替え
- ジオメトリ軸の割り当ての変更 GEOAX
- PI サービス「_N_SETUDT」により HMI から

HMI からの起動

PI サービス「_N_SETUDT」による HMI からのデータ管理フレームの起動は、現在のパートプログラムのホット再始動後にのみ、チャネル内で有効になります。

以下のマシンデータが設定されている場合、起動はリセット状態でも有効です。

MD9440 $MM_ACTIVATE_SEL_USER_DATA (設定された有効なオフセットが直ちに有効)

システムフレームの設定

システムフレームは、以下によって設定されます。

- パートプログラムでの対応するシステム機能のプログラミング
- SINUMERIK Operate でのオペレータコントロール

注記

データ管理のシステムフレームの変更

原則として、データ管理のシステムフレームは、サイクルプログラマが変更でき、G500、G54...G599 命令を使用して設定されます。ただし、このオプションは条件付きでのみ使用すべきです。

データ管理フレームの起動

データ管理フレームの起動を、以下のマシンデータで設定する場合の動作

MD24050 $MC_FRAME_SAA_MODE (データ管理フレームの保存と起動)

ビット	値	意味
0	0	@@@
	1	• データ管理フレームは、システム変数$P_CHBFRMASK, $P_NCBFRMASK and $P_CHSFRMASK のプログラム指令によって**のみ**起動されます。 • G500...G599 で、適切な設定可能フレームが起動します。
1	0	データ管理フレームが、TOROT、PAROT、外部ゼロオフセット、および座標変換などの機能によって自動的に記述されます。
	1	データ管理フレームが、TOROT、PAROT、外部ゼロオフセット、および座標変換などの機能で自動的には**記述されません**。

システム変数$P_CHSFRMASK によるシステムフレームの起動

データ管理のシステムフレームは、システム変数**$P_CHSFRMASK** を使用して設定できます。変数の値は、マシンデータ
MD28082 $MC_MM_SYSTEM_FRAME_MASK (データ管理のシステムフレーム)に従って、ビットでコード化されて指定されます。

チャネル内のデータ管理の対応するシステムフレームは、システム変数$P_CHSFRMASK のビットを 1 に設定して起動します。値が 0 の場合、チャネルで現在有効なシステムフレームは有効なままです。

リセット後のシステムフレームの設定

リセット後、以下のマシンデータでビットが設定されている、チャネル内のシステムフレームが設定されます。

MD24006 $MC_CHSFRAME_RESET_MASK (リセット後に有効なシステムフレーム)

TCARR、PAROT および TOROT、TOFRAME に対応したシステムフレームの設定

TCARR、PAROT および TOROT、TOFRAME 用のシステムフレームは、次のマシンデータの設定に従って設定されます。

MD20150 $MC_GCODE_RESET_VALUES (G グループの初期設定)

座標変換の選択/解除を使用するか、GEOAX 命令を使用したジオメトリ軸の切り替え時に、現在のトータルフレーム $P_ACTFRAME が削除されるか、新しいジオメトリ軸の座標を使用して再計算され、設定されます。システムフレームと他のすべてのフレームは、ジオメトリ軸に関連して再度調整されます。

11.5.3.3 NCU グローバルフレームとチャネル別フレーム

- 設定可能フレームと研削フレームは、NCU グローバル**フレームまたは**チャネル別フレームとしてのみ設定できます。
- 基本フレームは、NCU グローバル**フレームおよび**チャネル別フレームとして設定できます。
- NCU グローバルフレームは、NCU のすべてのチャネルで同様に機能します。
- NCU のすべてのチャネルは、NCU グローバルフレームを同様に読み取り、書き込むことができます。
- チャネル軸と特にジオメトリ軸への機械軸の割り当てはすべてのチャネル内で異なることがあるため、チャネル軸の間にチャネル間の幾何学的な関係はありません。したがって、NCU グローバルフレームに対して可能なのは、オフセット、スケール、およびミラーリングのみです。回転をおこなうことはできません。

注記

プログラム協調

NCU グローバルフレームへのチャネル別アクセスを協調させることは、ユーザーの責任となります。プログラム協調のための命令の使用を推奨します。

参照先

『プログラミングマニュアル、上級編』、「フレキシブルな NC プログラミング」>「プログラム協調(INIT、START、WAITM、WAITMC、WAITE、SETM、CLEARM)」の章

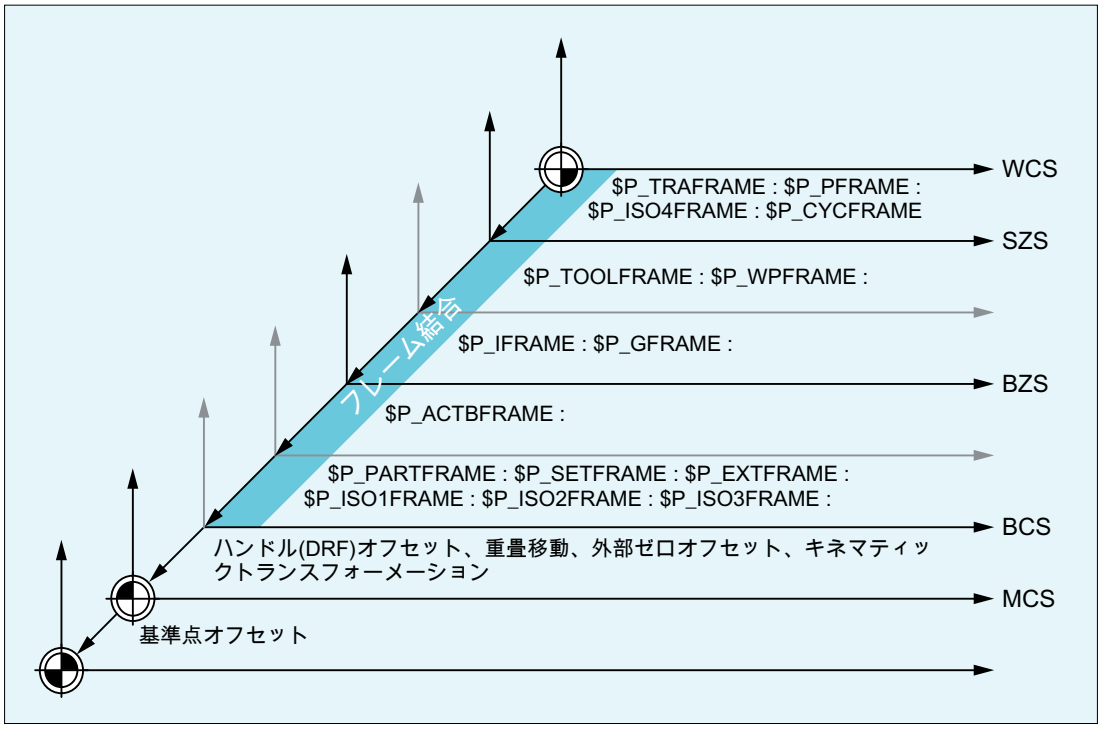
11.5.4 フレーム結合と座標系

11.5.4.1 概要

次の図では、現在の全フレームのフレーム結合を示しています。フレーム結合は、基本座標系(BCS)とワーク座標系(WCS)の間に配置されます。

設定可能ゼロオフセットシステム(SZS)は、プログラマブルフレームで変換した WCS に対応しています。基本ゼロオフセットシステム(BZS)には、現在の設定可能フレームも含

まれます。外部ゼロオフセット用のシステムフレームが存在するのは、そのシステムフレームが設定されている場合だけです。そうでない場合は、外部ゼロオフセットが、軸の重畳移動として移動します。



WCS　ワーク座標系(**W**orkpiece **C**oordinate **S**ystem)
:

SZS:　設定可能ゼロオフセットシステム(**S**ettable **Z**ero **S**ystem)

BZS:　基本ゼロオフセットシステム(**B**asic **Z**ero **S**ystem)

BCS:　基本座標系(**B**asic **C**oordinate **S**ystem)

MCS　機械座標系(**M**achine **C**oordinate **S**ystem)
:

フレーム全体

フレーム結合のすべての有効フレームの結合から 、在のフレーム一式$P_ACTFRAME が得られます。

```
$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
```

11.5.4.2 相対座標系

相対座標系は、表示されている有効な座標系内の指定のレファレンス点と相対的に位置する、軸の現在の指令値位置を表示します。相対座標系についてプログラミングをおこなうことはできません。システム変数を介して相対座標系内での軸位置のみを読み取ることができます。

この新しい表示座標系は、WCS と SZS 座標系と相対的に配置され、有効なシステムフレーム$P_RELFRAME を使用して WCS または SZS の軸位置を座標変換することにより生成されます。相対座標系は、直線的に移動できるだけでなく、回転や反転、縮小、拡大をおこなうこともできます。

軸指令値の位置表示は、WCS 内または SZS 内でおこないます。設定は、HMI マシンデータを介しておこないます。常に 1 つの表示座標系のみがチャネル内で有効になります。そのため、同じ割合で両方の相対座標系を生成する 1 つの相対フレームのみが提供されます。HMI は、設定に従って相対座標を表示します。

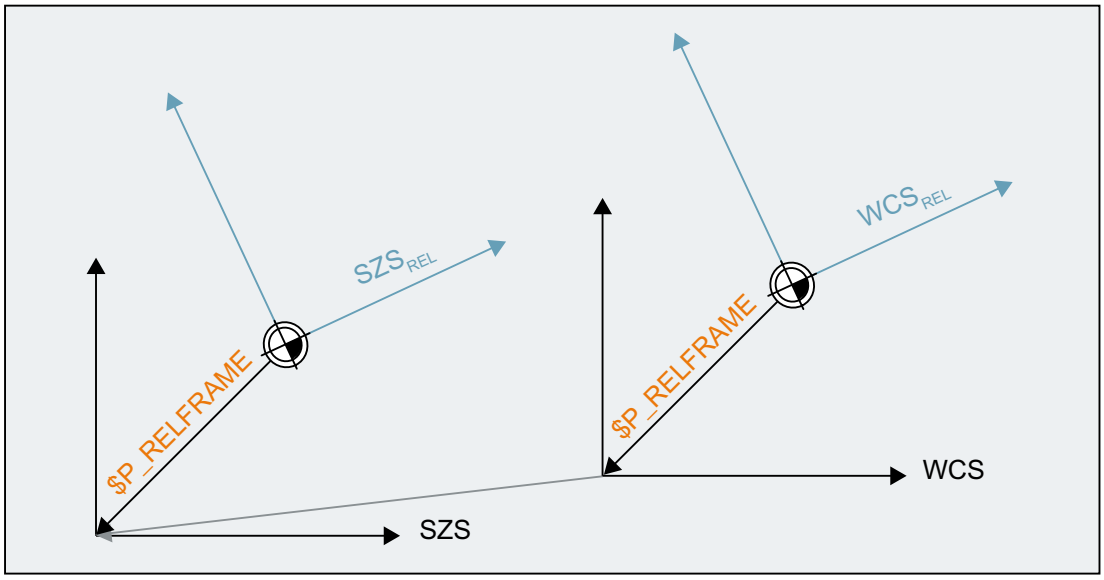


図 11-22　相対座標系

[相対座標系]ファンクションは、MD51036 $MNS_ENABLE_COORDINATE_REL=1 で起動します。

データ保守フレーム$P_RELFR は、OPI を介してパートプログラムに書き込むことができます。すべてのフレーム成分を変更できます。

有効なシステムフレーム$P_RELFRAME は、パートプログラムで書き込んで、OPI で読むことができます。

システムフレーム$P_RELFR の設定は、次のマシンデータを介しておこなわれます。

マシンデータ	ビット	意味
MD28082 $MC_MM_SYSTEM_FRAME_MASK	11	$P_RELFR の作成。これにより、相対座標系が存在する状態になります。
MD28083 $MC_MM_SYSTEM_DATAFRAME_MASK	11	データ保守フレーム$P_RELFR
MD24006 $MC_CHSFRAME_RESET_MASK.	11	リセット時に$P_RELFR が有効になります。
MD24007 $MC_CHSFRAME_RESET_CLEAR_MASK	11	リセット時に$P_RELFR が削除されます。
MD24008 $MC_CHSFRAME_POWERON_MASK	11	電源投入時に$P_RELFR が削除されます。

相対座標系 WCS_{Rel} 内での軸位置は、変数$AA_PCS_REL[ax]を介して読み取ることができます。この変数は、パートプログラム内、OPI およびシンクロナイズドアクションを介して読み取ることができます。

相対座標系 SZS_{Rel} 内での軸位置は、変数$AA_ACS_REL[ax]を介して読み取ることができます。この変数は、パートプログラム内、OPI およびシンクロナイズドアクションを介して読み取ることができます。

操作パネルを介した相対基準点の設定は、ワークと工具計測用の汎用命令インタフェースを介しておこないます。相対座標系用のシステムフレーム$P_RELFR は、次のように計算して起動されます。

- $AC_MEAS_TYPE = 14
- PI サービス _N_SETUDT(6, 7)

相対軸位置の設定例については、次を参照してください。

参照先:

『機能マニュアル、上級機能』;「計測(M5)」、
「ジオメトリ軸と付加軸の計測(計測タイプ 14、15)」の章

11.5.4.3 選択可能な SZS

サイクル内で、サイクル別ワーク座標系(WCS)で処理が実行されます。サイクル別 WCS は、サイクルに対してプログラム指令されたプログラマブルフレーム$P_PFRAME やサイクルフレーム$P_CYCFRAME によって変換された SZS から得られます。

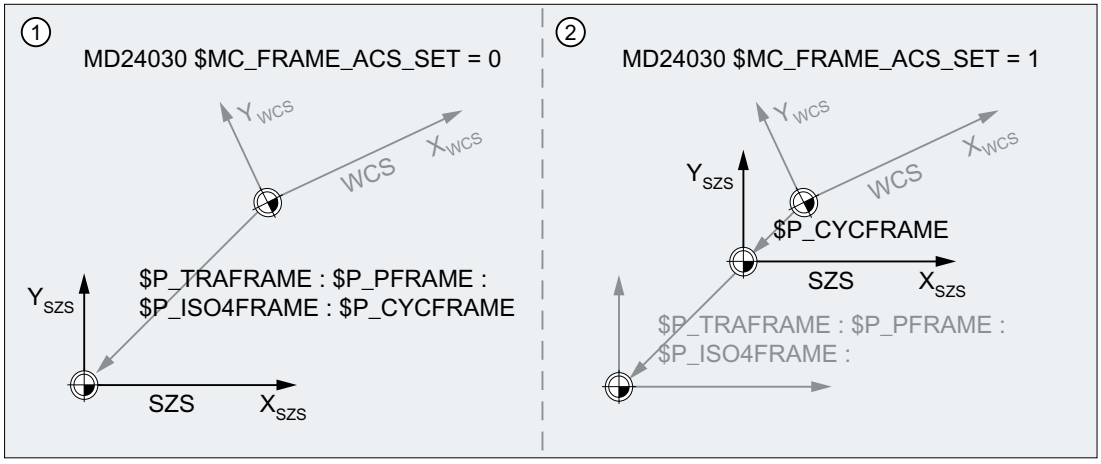
NC ストップなどにより、機械のオペレータによってサイクルが中断された場合は、サイクルの起動前に有効であった元の座標系(SZS)で移動が実行されます。

マシンデータ

サイクル別 WCS からの SZS の計算方法の仕様は、以下のマシンデータで設定します。

MD24030 $MC_FRAME_ACS_SET = <値>

<値>	意味
0	SZS = $P_TRAFRAME、$P_PFRAME、$P_ISO4FRAME、および $P_CYCFRAME で変換された WCS
1	SZS = $P_CYCFRAME のみで変換された WCS



① SZS = $P_TRAFRAME、$P_PFRAME、$P_ISO4FRAME、および $P_CYCFRAME で変換された WCS

② SZS = $P_CYCFRAME のみで変換された WCS

動作

SZS を再設定すると、以下に対して作用します。

- SZS 関連の現在値:現在値の表示、$AA_IEN などのシステム変数、など
- SZS でのジオメトリ軸の手動移動(JOG)

11.5.4.4 WCS または SZS でのジオメトリ軸の手動移動($AC_JOG_COORD)

これまで、ジオメトリ軸は、JOG モードでは WCS 内で手動移動されてきました。これに加えて、SZS 座標系内でこの手動移動を行うオプションも存在します。
$AC_JOG_COORD 変数により、WCS 内での手動移動と SZS 内での手動移動を切り替えることができます。これで、ユーザーは、SZS 内で移動するか、WCS 内で移動するかを選択できます。

JOG モードでの手動移動時にジオメトリ軸は、ワーク座標系(WCS)または設定可能ゼロオフセットシステム(SZS)で移動できます。

システム変数

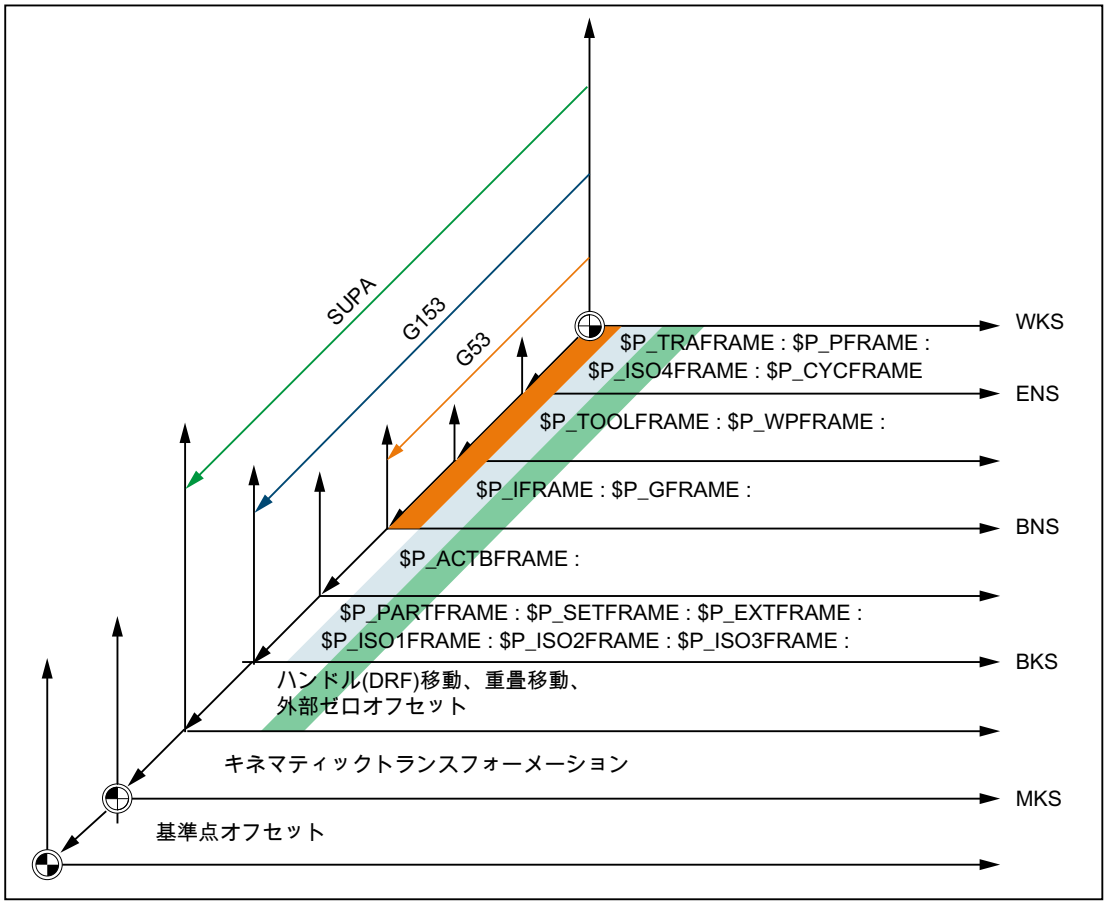
JOG モードでの手動移動時にジオメトリ軸は、ワーク座標系(WCS)または設定可能ゼロオフセットシステム(SZS)で移動できます。どちらを選択するかは、システム変数 $AC_JOG_COORD で指定します。

$AC_JOG_COORD = <値>

<値>	意味
0	ワーク座標系(WCS)
1	設定可能ゼロオフセットシステム(SZS)

11.5.4.5 フレームのマスク

フレームのマスクは、以下の図で説明する命令 G53、G135、および SUPA でチャネル別におこないます。フレームのマスクの起動は、位置表示(HMI)でのジャンプのほか、WCS、SZS、または BZS を基準とするシステム変数の位置データでのジャンプの結果として実行されます。この動作はマシンデータで設定できます。



マシンデータ

位置表示(HMI)の動作およびシステム変数の位置データは、以下のマシンデータで指定します。

MD24020 $MC_FRAME_SUPPRESS_MODE、ビット<n> = <値> (フレームマスク時の位置)

ビット	値	意味
0	0	フレームのマスクを**伴う**位置データ(OPI)[1]
	1	フレームのマスクを**伴わない**位置データ(OPI)[2]
1	0	フレームのマスクを**伴う**システム変数内の位置データ(OPI)[1]
	1	フレームのマスクを**伴わない**システム変数内の位置データ(OPI)[2]
1) 位置データのジャンプ 2) 位置データのジャンプ**なし**		

プログラミング

命令	意味
G53:	次のフレームのノンモーダルなマスク $P_TRAFRAME : $P_PFRAME : $P_ISO4FRAME : $P_CYCFRAME $P_IFRAME : $P_GFRAME : $P_TOOLFRAME : $P_WPFRAME :
G153:	G53 の場合のフレームのノンモーダルマスクおよび以下のフレーム $P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME $P_ISO1FRAME : $P_ISO2FRAME : $P_ISO3FRAME :
SUPA:	自動先読み停止と、G53 および G135 の場合のフレームのノンモーダルマスクおよび以下のフレーム ● ハンドルオフセット(DRF) ● 重畳移動 ● 外部ゼロオフセット
G500:	G500 フレームのモーダルな起動 G500 フレームは、通常はゼロフレームであるべきです。
DRFOF:	ハンドルオフセット(DRF)の解除(削除)

11.5.5 フレーム結合のフレーム

11.5.5.1 概要

以下のフレームが利用可能です。

- 設定可能フレーム(G500、G54 ... G57、G505 ... G599)
- 研削フレーム(GFRAME0 ... GFRAME100)
- 基本フレーム
- プログラマブルフレーム
- システムフレーム

11.5.5.2 設定可能フレーム($P_UIFR[<n>])

マシンデータ

チャネル別設定可能フレーム

チャネル別設定可能フレームの数は、次のマシンデータで設定します。

MD28080 $MC_MM_NUM_USER_FRAMES = <番号>

システムタグインデックス n = 0, 1, 2, ... <番号> - 1

NCU グローバル設定可能フレーム

NCU グローバル設定可能フレームの数は、次のマシンデータで設定します。

MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES = <番号>

システムタグインデックス n = 0, 1, 2, ... <番号> - 1

マシンデータに 0 より大きい値が含まれる場合は、**チャネル別**設定可能フレームが**ありません**。そのため、チャネル別設定可能フレームを設定するマシンデータは使用されません。

設定可能フレームの G グループのリセット設定

リセット設定、またはチャネルリセットまたは電源投入の後に設定可能フレーム別の **8 番目の G グループ**のどの G 命令が有効かは、以下で設定します。

MD20150 $MC_GCODE_RESET_VALUES[7] = <値>

値	G 命令
1	G500
2	G54
3	G55
4	G56
5	G57
6	G505
...	...
100	G599

研削別 G グループのリセット応答

設定可能フレーム別の **8 番目の G グループ**のリセット応答は、以下で設定します。

MD20152 $MC_GCODE_RESET_MODE[7] = <値>

値	意味 チャネルリセットまたはパートプログラム終了の後:
0	**MD20150** に従って設定可能フレーム別 G 命令が有効です。
1	現在有効な設定可能フレーム別 G 命令は有効なままです。

注記

MD20110 $MC_RESET_MODE_MASK

The MD20152 $MC_GCODE_RESET_MODE マシンデータは、以下にのみ使用します。
MD20110 $MC_RESET_MODE_MASK、ビット 0 == 1

システム変数

$P_UIFR[<n>] (データ管理の設定可能フレーム)

システム変数$P_UIFR[<n>]を使用して、データ管理の設定可能フレームの読み取りと書き込みを実行できます。データ管理の設定可能フレームの書き込み時に、新しい値は、すぐにはチャネルで有効になりません。ゼロオフセット G500、G54...G599 のプログラム指令によってのみ、チャネルで有効になります。

NCU グローバルフレームの場合は、データ管理の設定可能フレームを変更すると、G500、G54..G599 命令を実行する NCU の各チャネルで有効になります。

データバックアップ時には、データ管理の設定可能フレームも保存されます。

$P_IFRAME (有効な設定可能フレーム)

システム変数$P_IFRAME を使用して、チャネルで**有効な**設定可能フレームの読み取りと書き込みを実行できます。新しい値は、設定可能フレームの書き込み時に、すぐにチャネルで有効になります。

NCU グローバル設定可能フレームの場合、変更された有効なフレームは、新しい値がプログラムされたチャネルでのみ機能します。変更した NCU グローバル設定可能フレームを、NCU のすべてのチャネルで有効にする場合は、チャネルで有効な設定可能フレーム、**および**対応するデータ管理の設定可能フレームを一緒に書き込んでください。

$P_UIFR[<n>] = $P_IFRAME = <新しい値>

- $P_UIFR[<n>] (データ管理の設定可能フレーム)
- $P_IFRAME (チャネルで有効な設定可能フレーム)

変更した設定可能なフレームを別のチャネルで有効にするには、このチャネルで適切な命令(G54 など)によりフレームを起動する必要があります。

$P_UIFRNUM (有効な設定可能フレームの数)

システム変数$P_UIFRNUM を使用して、チャネルで有効なデータ管理の設定可能フレームのインデックス**<n>**を読み取ることができます。

チャネルで有効な設定可能フレーム$P_IFRAME == $P_UIFR[$P_UIFRNUM]

$P_UIFRNUM	$P_IFRAME == $P_UIFR[<n>]、ここで、n =
0	0
1	1
2	2
...	...
99	99

プログラミング

チャネル内で設定可能フレームを有効にする命令

G500、G54...G599 命令をプログラム指令すると、チャネル内の$P_UIFR[<n>]データ管理の設定可能フレーム、または$P_UIFR[<n>]データ管理の設定可能フレームと同じ設定の有効な$P_IFRAME 設定可能フレームが有効になります。

G<x> ⇒ $P_IFRAME = $P_UIFR[<n>]

命令	有効な設定可能フレーム$P_IFRAME
G500	$P_UIFR[0]
G54	$P_UIFR[1]
G55	$P_UIFR[2]
G56	$P_UIFR[3]
G57	$P_UIFR[4]
G505	$P_UIFR[5]
...	...
G599	$P_UIFR[99]

必要条件

HMI/PLC による設定可能フレームの書き込み

HMI または PLC ユーザープログラムからは、**データ管理**の設定可能フレームのみを書き込むことができます。

11.5.5.3 研削フレーム$P_GFR[<n>]

研削フレームは、特に研削の用途で使用可能な追加のゼロオフセットまたは「適合依存補正」です。設定可能フレーム (ページ 926)のゼロオフセットに追加します。

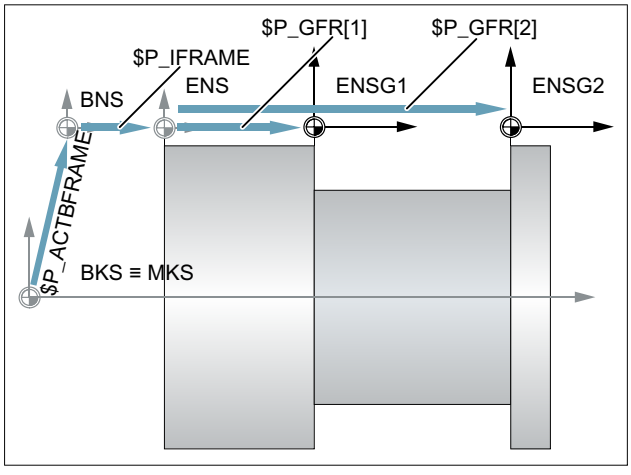


図 11-23 研削フレーム

基本ゼロオフセットシステム(BZS)に基づいて、設定可能ゼロオフセットシステム(SZS)が、チャネルで有効なフレームの結合によって得られます。

$P_IFRAME: **$P_GFRAME** : $P_TOOLFRAME :$P_WPFRAME

マシンデータ

チャネル別研削フレームの数

チャネル別研削フレームの数は、以下で設定します。

MD28079 $MC_MM_NUM_G_FRAMES = <番号>

ここで、<番号> = 0, 1, 2, ... 最大数

NCU グローバル研削フレームの数

NCU グローバル研削フレームの数は、以下で設定します。

MD18603 $MC_MM_NUM_GLOBAL_G_FRAMES = <番号>

ここで、<番号> = 0, 1, 2, ... 最大数

マシンデータに 0 より大きい値が含まれる場合は、**チャネル別**研削フレームが**ありません**。そのため、チャネル別研削フレームを設定するマシンデータは使用されません。

研削別 G グループのリセット設定(64)

リセット設定、またはチャネルリセットまたは電源投入の後に研削別の **64 番目の G 命令**のどの G 機能が有効かは、以下で設定します。

MD20150 $MC_GCODE_RESET_VALUES[63] = <値>

値	G 命令
1	GFRAME0 (ゼロフレーム)
2	GFRAME1
3	GFRAME2
...	...
101	GFRAME100

研削別 G グループのリセット応答

研削別の **64 番目の G ユーザーグループ**のリセット応答は、以下で設定します。

MD20152 $MC_GCODE_RESET_MODE[63] = <値>

値	意味 チャネルリセットまたはパートプログラム終了の後:
0	MD20150 に従って研削別 G 命令が有効です。
1	現在有効な研削別 G 命令は有効なままです。

注記

MD20110 $MC_RESET_MODE_MASK

The MD20152 $MC_GCODE_RESET_MODE マシンデータは、以下にのみ使用します。
MD20110 $MC_RESET_MODE_MASK、ビット 0 == 1

システム変数

$P_GFR[<n>] (データ管理の研削フレーム)

システム変数$P_GFR[<n>]を使用して、データ管理の研削フレームの読み取りと書き込みを実行できます。新しい値は、研削フレームの書き込み時に、すぐにはチャネルで有効になりません。適切な命令 GFRAME0...GFRAME100 のプログラム指令によってのみ、チャネルで有効になります。NCU グローバルフレームの場合、変更されたフレームは、GFRAME0 ... GFRAME100 命令を実行する NCU のチャネル内でのみ有効になります。

データバックアップ時には、データ管理の研削フレームも保存されます。

注記

表示(SINUMERIK Operate)

データ管理の研削フレームは、SINUMERIK Operate 操作画面の別のウィンドウに表示されます。

- 参照先
  操作マニュアル 研削; セクション「機械のセットアップ」>「ゼロオフセット」>「シート関連の仕上げオフセットの表示と編集」

解除(SINUMERIK Operate)

データ管理の研削フレームは個別に削除することができます。またはワークの変更の後などに SINUMERIK Operate 操作画面から完全に削除することもできます。

- 参照先
  操作マニュアル 研削; セクション「機械のセットアップ」>「ゼロオフセット」>「シート関連の仕上げオフセットの表示と編集」

$P_GFRAME (有効な研削フレーム)

チャネルで**有効な**研削フレームは、$P_GFRAME システム変数から読み取ったり、$P_GFRAME システム変数に書き込んだりできます。新しい値は、有効な検索フレームの書き込み時に、すぐにチャネルで有効になります。

NCU グローバル研削フレームの場合、変更したフレームは、新しいフレーム値をプログラム指令したチャネルでのみ機能します。

変更した NCU グローバル研削フレームを、1 つのチャネルから始めて NCU のすべてのチャネルで有効にする場合は、研削フレームで有効な$P_GFRAME、および$P_GFR[<n>]データ管理の研削フレームを一緒に書き込んでください。

$P_GFRAME = $P_GFR[<n>] = <新しい値>

$P_GFR[<n>]データ管理の研削フレームへの変更を別のチャネルで有効にするには、そのチャネルで、適切な命令 GFRAME<n>を使用して有効にしてください。

$P_GFRNUM (有効な研削フレームの数)

システム変数$P_GFRNUM を使用して、チャネルで有効なデータ管理の研削フレームのインデックス<n>を読み取ることができます。

チャネルで有効な研削フレーム$P_GFRAME == $P_GFR[$P_GFRNUM]

G 命令 GFRAME<n>で有効にした $P_GFRAME 研削フレーム	$P_GFRNUM
GFRAME0	0
GFRAME1	1
GFRAME2	2
...	...
GFRAME100	100

プログラミング

チャネル内で研削フレームを有効にする命令

GFRAME<n>命令をプログラミングすると、$P_GFR[<n>]データ管理の対応する研削フレームがチャネル内で有効になります。これにより、データ管理の$P_GFR[<n>]研削フレームと同一の有効な$P_GFRAME 研削フレームが設定されます。

GFRAME<n> ⇒ $P_GFRAME = $P_GFR[<n>]

命令	チャネル内で有効になった研削フレーム
GFRAME0	$P_GFR[0] (ヌルフレーム)
GFRAME1	$P_GFR[1]
...	...
GFRAME100	$P_GFR[100]

構文

GFRAME<n>

意味

GFRAME<n>:	データ管理の研削フレーム**<n>**の有効化	
	G グループ:	64
	基本設定:	MD20150 $MC_GCODE_RESET_VALUES[**63**]
	効果:	モーダル
<n>:	研削フレームの番号	
	値の範囲:	0, 1, 2, ... 100

必要条件

HMI/PLC による研削フレームの書き込み

HMI または PLC ユーザープログラムからは、**データ管理**の研削フレームのみを書き込むことができます。

11.5.5.4 チャネル別基本フレーム[<n>]

マシンデータ

チャネル別基本フレームの数

チャネル別基本フレームの数は、以下のマシンデータで設定します。

MD28081 $MC_MM_NUM_BASE_FRAMES = <番号>

システムタグインデックス n = 0, 1, 2, ... <番号> - 1

システム変数

$P_CHBFR[<n>] (データ管理のチャネル別基本フレーム)

システム変数$P_CHBFR[<n>]を使用して、データ管理のチャネル別基本フレームの読み取りと書き込みを実行できます。新しい値は、チャネル別基本フレームの書き込み時に、すぐにはチャネルで有効になりません。適切な命令 G500、G54...G599 のプログラム指令によってのみ、チャネルで有効になります。

データバックアップ時には、データ管理のチャネル別基本フレームも保存されます。

$P_CHBFRAME[<n>] (有効なチャネル別基本フレーム)

システム変数$P_CHBFRAME[<n>]を使用すると、**有効な**チャネル別基本フレームの読み取りと書き込みを実行できます。有効なチャネル別基本フレームの書き込み時には、有効な基本フレーム一式$P_ACTBFRAME の再計算によって、新しい値がすぐに有効になります。

互換性のためのシステム変数

$P_UBFR (最初のデータ管理のチャネル別基本フレーム)

このシステム変数は、$P_CHBFR[0]変数と重なりますが、互換性のために保持されています。

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。NCU グローバルフレームの場合、変更されたフレームは、G500、G54..G599 命令を実行する NCU のチャネル内でのみ有効になります。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。

$P_UBFR は$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、アラーム:「Frame:命令が許可されていません」が、読み取りまたは書き込みアクセス時に出力されます。

$P_BFRAME (最初の有効なチャネル別基本フレーム)

このシステム変数は、$P_CHBFRAME[0]変数と重なりますが、互換性のために保持されます。

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。NCU グローバル設定可能フレームの場合は、変更されたフレームは、そのフレームをプログラム指令したチャネルでのみ動作します。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、$P_UBFR と$P_BFRAME を同時に書き込んでください。その後、その他のチャネルは、G54 などを使用して、対応するフレームを起動しなければなりません。

$P_BFRAME は$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、アラーム:「Frame:命令が許可されていません」が、読み取りまたは書き込みアクセス時に出力されます。

必要条件

HMI/PLC による基本フレームの書き込み

HMI または PLC ユーザープログラムからは、**データ管理**の基本フレームのみを書き込むことができます。

11.5.5.5 NCU グローバル基本フレーム$P_NCBFR[<n>]

マシンデータ

NCU グローバル基本フレームの数

NCU グローバル基本フレームの数は、次のマシンデータで設定します。

MD18602 $MN_MM_NUM_GLOBAL_BASE_FRAMES = <番号>

システムタグインデックス n = 0, 1, 2, ... <番号> - 1

システム変数

$P_NCBFR[<n>] (データ管理の NCU グローバル基本フレーム)

システム変数$P_NCBFR[<n>]を使用して、データ管理の NCU グローバル基本フレームの読み取りと書き込みを実行できます。新しい値は、NCU グローバル基本フレームの書き込み時に、すぐにはチャネルで有効になりません。適切な命令 G500、G54...G599 のプログラム指令によってのみ、チャネルで有効になります。

データバックアップ時には、データ管理の NCU グローバル基本フレームも保存されます。

$P_NCBFRAME[<n>] (現在の NCU グローバル基本フレーム)

システム変数$P_NCBFRAME[<n>]を使用すると、**有効な** NCU グローバル基本フレームの読み取りと書き込みを実行できます。有効な NCU グローバル基本フレームの書き込み時には、有効な基本フレーム一式$P_ACTBFRAME の再計算によって、新しい値がすぐに有効になります。

変更した NCU グローバル基本フレームを、1 つのチャネルから始めて NCU のすべてのチャネルで有効にする場合は、チャネルで有効な NCU グローバル基本フレーム、およびデータ管理の NCU グローバルフレームを一緒に書き込んでください。

$P_NCBFR[<n>] = $P_NCBFRAME = <新しい値>

- $P_NCBFR[<n>] (データ管理の NCU グローバル基本フレーム)
- $P_NCBFRAME (チャネルで有効な NCU グローバル基本フレーム)

NCU グローバルフレームへの変更を別のチャネルで有効にするには、そのチャネルで、適切な命令 G500、G54..G599 を使用して有効にしてください。

プログラミング

データ管理のチャネル別設定可能フレーム$P_UIFR[<n>]は、適切な命令(G54 ... G57、G505 ... G599、および G500)によって、チャネルで有効な設定可能フレーム$P_IFRAME になります。

命令	NCU グローバル基本フレームおよびデータ管理のチャネル別基本フレームの有効化
G500	$P_CHBFR[0] : $P_NCBFR[0]
G54	$P_CHBFR[1] : $P_NCBFR[1]
G55	$P_CHBFR[2] : $P_NCBFR[2]
G56	$P_CHBFR[3] : $P_NCBFR[3]
G57	$P_CHBFR[4] : $P_NCBFR[4]
G505	$P_CHBFR[5] : $P_NCBFR[5]
...	...
G599	$P_CHBFR[99] : $P_NCBFR[99]

11.5.5.6 有効な基本フレーム一式$P_ACTBFRAME

機能

すべての有効な NCU グローバル基本フレーム およびチャネル別基本フレームを組み合わせると、基本フレーム一式$P_ACTBFRAME になります。

```
$P_ACTBFRAME = $P_NCBFRAME[0] : ... : $P_NCBFRAME[<n>] :
               $P_CHBFRAME[0] : ... : $P_CHBFRAME[<n>]
```

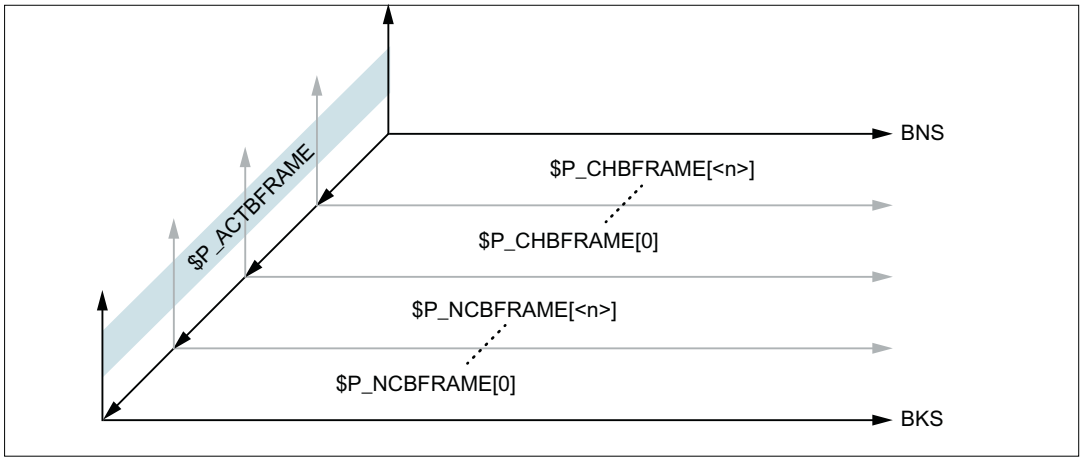


図 11-24　基本フレーム全体

マシンデータ

リセット動作

リセット(チャネルリセット、プログラム終了リセット、または電源投入)後に有効な基本フレームは、以下のマシンデータで設定します。

MD20110 $MC_RESET_MODE_MASK、ビット 0 = 1 およびビット 14 = 1

- ビット 1 = 0:初期設定⇒その他のビットの設定に対応するリセット応答
- ビット 14 = 0:基本フレームが、リセットにより完全に選択解除されます。
- ビット 14 = 1:リセットにより、マシンデータの設定がシステム変数に引き継がれ、そこで選択された基本フレームが有効になります。
  - $P_NCBFRMASK = MD10613 $MN_NCBFRAME_RESET_MASK
  - $P_CHBFRMASK = MD24002 $MC_CHBFRAME_RESET_MASK

例

マシンデータの設定	有効な基本フレーム
$P_NCBFRMASK = MD10613 $MN_NCBFRAME_RESET_MASK = 'H81'	$P_NCBFRAME[0] : $P_NCBFRAME[7]

プログラミング

基本フレームマスク

結合されて基本フレーム一式になっている基本フレームは、基本フレームマスク $P_NCBFRMASK および$P_CHBFRMASK を介して選択します。

適切な基本フレームは、基本フレーム**マスク**にビットを設定して選択します。

- $P_NCBFRMASK、ビット **0、1、2、... n** ⇒ $P_NCBFRAME[**0、1、2、... n**]
- $P_CHBFRMASK、ビット **0、1、2、... n** ⇒ $P_NCHFRAME[**0、1、2、... n**]

基本フレームマスク$P_NCBFRMASK および$P_CHBFRMASK は、NC プログラムでのみ読み取り/書き込みが可能です。基本フレームマスクは、OPI で読み取ることができます。

基本フレームマスクを書き込んだ後、有効な基本フレーム一式$P_ACTBFRAME およびすべてのフレーム$P_ACTFRAME が再計算されます。

例

プログラムコード	コメント
$P_NCBFRMASK = 'H81'	; 有効な NCU グローバル基本フレーム: $P_NCBFRAME[0] : $P_NCBFRAME[7]

11.5.5.7 プログラマブルフレーム$P_PFRAME

プログラマブルフレームは、有効なフレームとしてのみ使用できます。
このフレームはプログラマのために予約されています。

以下の場合、プログラマブルフレームは RESET のままとなります。

MD24010 $MC_PFRAME_RESET_MODE (プログラマブルフレームのリセットモード) = 1

この機能は、傾斜穴から後退する場合に、特に RESET 後に重要になります。

MIRROR

ジオメトリ軸のミラーリングは、これまで(SW-P4 まで)、マシンデータ:
MD10610 $MN_MIRROR_REF_AX
([ミラーリング用の基準軸])を使用して定義済の基準軸にのみ関連付けられました。

ユーザーの視点から見ると、この定義は理解しにくいものです。z 軸をミラーリングする場合、表示では、x 軸がミラーリングされ、y 軸が 180°まで回転されました。2 つの軸をミラーリングする場合はさらに複雑になり、どの軸がミラーリングされていて、どの軸がミラーリングされていないかを簡単に理解できなくなりました。

SW P5 以降では、軸のミラーリングを明確に表示するオプションがあります。ミラーリングは、基準軸のミラーリングとその他の軸の回転にマッピングされません。

この設定は以下を使用して設定することができます。

MD10610 $MN_MIRROR_REF_AX = 0

MIRROR と AMIRROR は、プログラマブルフレームのプログラミングを拡張するために使用されます。これまでは、座標軸の指定の値、たとえば、MIRROR X0 の値 0 は機能しませんでした。しかし、AMIRROR には切替機能があります。つまり、MIRROR X0 はミラーを起動し、次の AMIRROR X0 がそれを解除します。MIRROR には常にアブソリュート効果があり、AMIRROR には追加効果があります。

MD10612 $MN_MIRROR_TOGGLE = 0 (「ミラーリング切替」)
マシンデータ設定を使用して、プログラム指令値が使用されるように定義できます。AMIRROR X0 のように、値 0 は、軸のミラーリングを解除します。値が 0 でなければ、軸は、まだミラーリングされていない場合、ミラーリングされます。

成分毎のミラーリングの読み取りまたは書き込みは、マシンデータとは無関係です。

MD10612 $MN_MIRROR_TOGGLE

値= 0 の場合、軸はミラーリングされません。値= 1 の場合、軸は、すでにミラーリングされていたかどうかに関係なく、常にミラーリングされます。

```
$P_NCBFR[0,x,mi] = 1          ; x 軸は常にミラーリングされます。
$P_NCBFR[0,x,mi] = 0          ; x 軸のミラーリングはオフです。
```

軸別の変更 G58、G59 (840D sl のみ)

プログラマブルフレームの座標移動成分は、アブソリュート成分と、追加でプログラム指令されたすべての座標移動の合計の成分に分けられます。アブソリュート成分を変更するには、TRANS、CTRANS を使用するか、座標移動成分を書き込みます。これによって、追加成分は 0 に設定されます。G58 は、指定の軸のアブソリュート座標移動成分のみを変更します。追加でプログラム指令された座標移動の合計は保持されます。

G58 X...Y...Z...A... ...

G59 は、ATRANS により特定の軸について追加でプログラムされた座標移動の軸別の上書きに使用されます。

G59 X...Y...Z...A... ...

例

```
TRANS X10 Y10 Z10
ATRANS X5 Y5                  ; 総座標移動量 X15 Y15 Z10
G58 X20                       ; 総座標移動量 X25 Y15 Z10
G59 X10 Y10                   ; 総座標移動量 X30 Y20 Z10
```

G58 と G59 は、以下の場合にのみ使用できます。

MD24000 $MC_FRAME_ADD_COMPONENTS (G58/G59 のフレーム成分) == TRUE

次の表に、さまざまなプログラム命令がアブソリュートと追加の座標移動に与える影響を示します。

	荒削りオフセットまたはアブソリュートオフセット	仕上げまたは追加オフセット
TRANS X10	10	0
ATRANS X10	不変	旧仕上げ + 10
CTRANS(X,10)	10	0
CTRANS()	0	0
CFINE(X,10)	0	10
$P_PFRAME[X,TR] = 10	10	不変
$P_PFRAME[X,FI] = 10	不変	10
G58 X10	10	不変
G59 X10	不変	10

11.5.5.8 チャネル別システムフレーム

チャネル別システムフレームは、現在値の設定、接触計測、外部ゼロオフセット、および傾斜面加工などのシステムファンクションによってのみ書き込まれます。

マシンデータ

チャネル別システムフレームのパラメータ設定

メモリ領域に制限があるため、実際に使用されるシステムファンクションを持つチャネル別システムフレームのみを設定してください。

チャネルごとに、各システムフレームは約 1 KB の静的メモリと約 6 KB の動的メモリを占有します。

チャネル別システムフレームは、以下のマシンデータで設定します。

MD28082 $MC_MM_SYSTEM_FRAME_MASK、ビット<n>

ビット	値	使用可能なシステムフレーム:
0	1	$P_SETFR:現在値の設定および接触計測
1	1	$P_EXTFR:システムフレームを介した外部ゼロオフセット

ビット	値	使用可能なシステムフレーム:
2	1	$P_PARTFR:旋回工具ホルダを使用した TCARR および PAROT
3	1	$P_TOOLFR:TOROT および TOFRAME
4	1	$P_WPFR:ワークレファレンス点用のフレーム
5	1	$P_CYCFR:サイクル用のフレーム
6	1	$P_TRAFR:座標変換の選択および解除用のフレーム
7	1	$P_ISO1FRAME :G51.1 ミラーリング(ISO)用のフレーム
8	1	$P_ISO2FRAME :G68 二次元回転(ISO)用のフレーム
9	1	$P_ISO3FRAME :G68 三次元回転(ISO)用のフレーム
10	1	$P_ISO4FRAME:G51 スケーリング(ISO)用のフレーム
11	1	$P_RELFR:相対座標系用のフレーム

SZS (ACS)座標系のパラメータ設定

以下のマシンデータを使用して、SZS (ACS)座標系を形成するシステムフレームを指定します。

MD24030 $MC_FRAME_ACS_SET = <値>

<値>	意味 SZS (ACS)座標系を形成するシステムフレーム
0	$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME :$P_ISO1FRAME : $P_ISO2FRAME : $P_ISO3FRAME :$P_ACTBFRAME : $P_IFRAME : $P_GFRAME : $P_TOOLFRAME : **$P_WPFRAME**
1	$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**

システム変数

データ管理のチャネル別システムフレーム

データ管理のチャネル別システムフレームは、以下のフレーム変数を使用して、読み取りと書き込みが可能です。

システム変数	意味データ管理のシステムフレームの用途
$P_SETFR	現在値の設定および接触計測(**Set Fr**ame)
$P_EXTFR	外部ゼロオフセット(**Ext Fr**ame)
$P_PARTFR	旋回工具ホルダ用 TCARR および PAROT (**Part Fr**ame)
$P_TOOLFR	TOROT および TOFRAME (**Tool Fr**ame)
$P_WPFR	ワークのレファレンス点 (**WorkPiece Fr**ame)
$P_CYCFR	サイクル (**Cycle Fr**ame)
$P_TRAFRAME	座標変換(**Tra**nsformation **Fr**ame)
$P_ISO1FR	G51.1 ミラーリング(ISO)
$P_ISO2FR	G68 2DROT (ISO)
$P_ISO3FR	G68 3DROT (ISO)
$P_ISO4FR	G51 スケール(ISO)
$P_RELFR	相対座標系

注記

サイクルプログラミング

システムフレームのフレーム変数はサイクルプログラミング専用です。したがって、NC プログラムでは、ユーザーが直接、システムフレームを書き込むことはできず、TOROT、PAROT などのシステムファンクションを使用しなければなりません。

有効なチャネル別のシステムフレーム

有効なチャネル別システムフレームのシステム変数

システム変数	意味有効なシステムフレームの用途
$P_SETFRAME	現在値の設定および接触計測(**Set Fr**ame)
$P_EXTFRAME	外部ゼロオフセット(**Ext Fr**ame)
$P_PARTFRAME	旋回工具ホルダ用 TCARR および PAROT (**Part Fr**ame)
$P_TOOLFRAME	TOROT および TOFRAME (**Tool Fr**ame)
$P_WPFRAME	ワークのレファレンス点 (**WorkPiece Fr**ame)

システム変数	意味有効なシステムフレームの用途
$P_CYCFRAME	サイクル (**Cyc**le **Fr**ame)
$P_TRAFRAME	座標変換(**Tra**nsformation **Fr**ame)
$P_ISO1FRAME	G51.1 ミラーリング(ISO)
$P_ISO2FRAME	G68 2DROT (ISO)
$P_ISO3FRAME	G68 3DROT (ISO)
$P_ISO4FRAME	G51 スケール(ISO)
$P_RELFRAME	相対座標系

データ管理のチャネル別システムフレームを設定していない場合は、対応する有効なシステムフレームに「$P_<システムフレーム> ==ヌルフレーム」が適用されます。

有効なチャネル別 SZS (ACS)フレーム一式

システム変数$P_ACSFRAME を使用して、SZS (ACS)座標系を形成するシステムフレームを読み取ります。仕様は、前述の MD24030 $MC_FRAME_ACS_SET マシンデータで指定します。サブセクション「マシンデータ」>「SZS (ACS)座標系のパラメータ設定」を参照してください。

システム変数	意味有効なシステムフレームの用途
$P_ACSFRAME	MD24030 $MC_FRAME_ACS_SET での設定に従って、SZS (ACS) 座標系を形成するシステムフレーム

11.5.6 自動的なフレーム変更

11.5.6.1 ジオメトリ軸の切り替え

チャネルのジオメトリ軸となるチャネル軸は、座標変換を有効化/無効化および GEOAX() 命令で変更できます。

現在の$P_ACTFRAME フレーム一式の処理では、以下のマシンデータで 4 種類の設定が可能です。

MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = <値>

<値>	意味
0	解除 ジオメトリ軸が切り替わった場合、座標変換が選択／解除された場合、および GEOAX() のとき、現在のフレーム一式が削除されます。 変更したジオメトリ軸設定は、フレームが新たに起動されるまで使用されません。
1	回転の受信の直後の新しい演算 ジオメトリ軸が切り替わった場合、現在のフレーム一式が再度計算され、新しいジオメトリ軸のフレーム成分が有効になります。切り替えの前にプログラム指令されたジオメトリ軸の回転は、新規ジオメトリ軸に対しても有効なままです。 TRANSMIT、TRACYL、および TRAANG については、「座標変換の選択と選択解除:概要 (ページ 947)」の章を参照してください。
2	有効な回転がなかった場合にのみ新しい演算 ジオメトリ軸が切り替わった場合、現在のフレーム一式が再度計算され、新しいジオメトリ軸のフレーム成分が有効になります。切り替え前に、現在の基本フレーム、現在の設定可能フレーム、またはプログラマブルフレームで回転が有効になっている場合、切り替えが中止され、アラーム「フレーム:ジオメトリ軸の切り替えができません」が出力されます。 TRANSMIT、TRACYL、および TRAANG については、「座標変換の選択と選択解除:概要 (ページ 947)」の章を参照してください。
3	座標変換:リセット/ GEOAX():回転の受信の直後の新しい演算 ● 座標変換:座標変換を選択、解除すると、現在のフレームが削除されます。 ● GEOAX():GEOAX() を使用して、現在のフレーム一式が再計算され、新規ジオメトリ軸の座標移動、スケーリングおよびミラーリングが有効になります。切り替えの前にプログラム指令されたジオメトリ軸の回転は、新規ジオメトリ軸に対しても有効なままです。

ワークジオメトリは、ジオメトリ軸により形成される座標系により記述されます。各チャネル軸は各ジオメトリ軸に割り当てられ、各機械軸は各チャネル軸に割り当てられます。機械軸(システムフレーム、基本フレーム、設定可能フレーム、プログラムマブルフレーム)毎に軸フレームが 1 つ存在します。異なる機械軸がジオメトリ軸に割り当てられている場合は、機械軸が、それ自体の軸フレーム成分を提供します。その後、新しいジオメトリ軸(3 つまで)により生成される新規輪郭フレームにより、チャネル内の新規ジオメトリが生成されます。

ジオメトリ軸切り替え時に、現在の有効フレームが再度計算され、この結果としてフレーム一式が生成されます。データ管理フレームは、起動されるまで組み込まれません。

例:

ジオメトリ軸切り替えにより、チャネル軸 A がジオメトリ軸 X になります。この切り替えは、X 軸の 10 の平行移動コンポーネントをプログラマブルフレームに提供しなければなりません。現在の設定可能フレームは保持されます。

MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 1

プログラムコード	コメント
$P_UIFR[1] = CROT(x,10,y,20,z,30)	; ジオメトリ軸切り替え後にフレームは保持されます。
G54	; 設定可能フレームが有効になります。
TRANS A10	; A の軸オフセットも切り替わります。
GEOAX(1, a)	; A が X 軸になります。
	; $P_ACTFRAME = CROT(X,10,Y,20,Z, 30) :CTRANS(X10)

座標変換の変更で、複数のチャネル軸がジオメトリ軸になる可能性があります。

例:

チャネル軸 4、5 および 6 は、5 軸座標変換のジオメトリ軸になります。このため、座標変換前にジオメトリ軸がすべて交換されます。座標変換が起動されると、現在のフレームが変更されます。新規 WCS を計算する場合、ジオメトリ軸になるチャネル軸の軸フレーム成分が考慮されます。座標変換前にプログラム指令された回転は保持されます。座標変換が解除されると、旧 WCS が復元されます。座標変換の前後でジオメトリ軸が変更されず、フレームが座標変換前のまま変更されないのが、おそらく、最も一般的な用途です。

マシンデータ

プログラムコード

```
$MC_AXCONF_CHANAX_NAME_TAB[0] = "CAX"
$MC_AXCONF_CHANAX_NAME_TAB[1] = "CAY"
$MC_AXCONF_CHANAX_NAME_TAB[2] = "CAZ"
$MC_AXCONF_CHANAX_NAME_TAB[3] = "A"
$MC_AXCONF_CHANAX_NAME_TAB[4] = "B"
$MC_AXCONF_CHANAX_NAME_TAB[5] = "C"

$MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1
$MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 2
$MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3
```

プログラムコード

```
$MC_AXCONF_GEOAX_NAME_TAB[0] = "X"
$MC_AXCONF_GEOAX_NAME_TAB[1]="Y"
$MC_AXCONF_GEOAX_NAME_TAB[2] = "Z"

$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=4
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=5
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=6

$MC_TRAFO_AXES_IN_1[0] = 4
$MC_TRAFO_AXES_IN_1[1] = 5
$MC_TRAFO_AXES_IN_1[2] = 6
$MC_TRAFO_AXES_IN_1[3] = 1
$MC_TRAFO_AXES_IN_1[4] = 2
```

プログラム:

プログラムコード

```
$P_NCBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
$P_CHBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
$P_IFRAME = ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(z,45)
$P_PFRAME = ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(x,10,y,20,z,30)
```

プログラムコード	コメント
TRAORI	; 座標変換セット GEOAX(4,5,6)
	; $P_NCBFRAME[0] = ctrans(x,4,y,5,z,6,cax,1,cay,2,caz,3)
	; $P_ACTBFRAME =ctrans(x,8,y,10,z,12,cax,2,cay,4,caz,6)
	; $P_PFRAME = ctrans(x,4,y,5,z,6,cax,1,cay,2,caz,3):crot(x,10,y,20,z,30)
	; $P_IFRAME = ctrans(x,4,y,5,z,6,cax,1,cay,2,caz,3):crot(z,45)
TRAFOOF	; 座標変換セット GEOAX(1,2,3)の解除
	; $P_NCBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
	; $P_CHBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
	; $P_IFRAME = ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(z,45)
	; $P_PFRAME = ctrans(x,1,y,2,z,3,a,4,b,5,c,6):crot(x,10,y,20,z,30)

11.5.6.2 座標変換の選択と選択解除:概要

一般に、座標変換の選択および解除により、チャネル軸へのジオメトリ軸の割り当てが変更されます。回転軸が直線軸になる、または直線軸が回転軸になる座標変換を実行する際に、軸フレーム成分を幾何学的輪郭フレーム成分に一義的に割り当てることはできません。こうした非直線的な座標変換の場合、特殊な処理を使用して輪郭フレームを調整する必要があります。

MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 1 および 2 で設定されたモードは、上記の座標変換を考慮する方法で拡張されます。

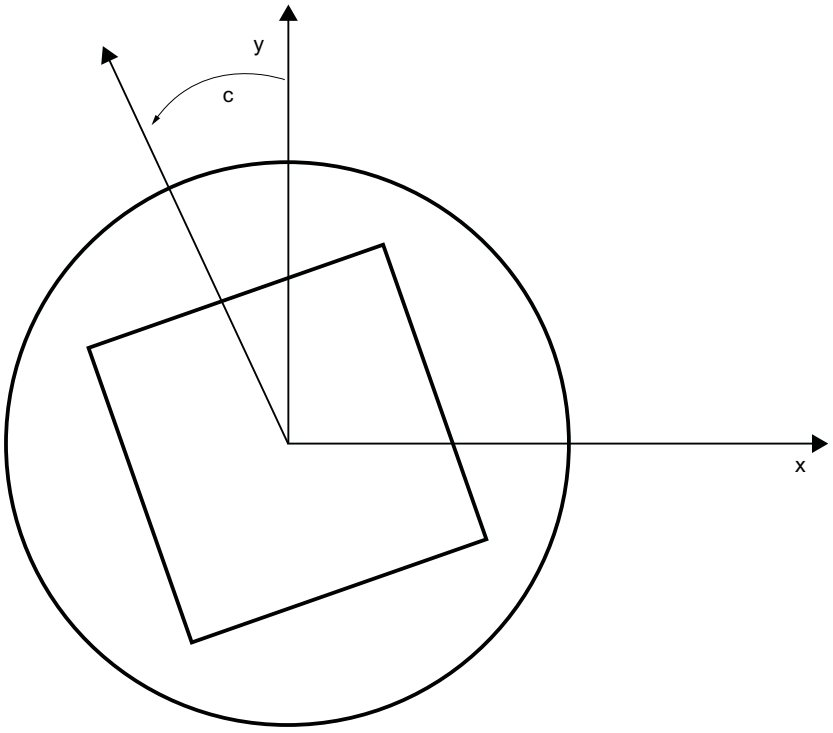
座標変換を選択すると、輪郭フレームが軸別フレームに接続されます。TRANSMIT、TRACYL および TRAANG の仮想ジオメトリ軸は、特殊な処理を受けます。

注記

仮想軸による座標変換

TRANSMIT または TRACYL を選択すると、実際の Y 軸のオフセット、スケーリング、ミラーリングは仮想 Y 軸に反映されません。TRAFOOF の場合、仮想 Y 軸のオフセット、スケーリング、ミラーリングは削除されます。

11.5.6.3 座標変換の選択と選択解除:TRANSMIT



Transmit の拡張

以下のマシンデータを使用して、TRANSMIT 回転軸の軸別フレーム一式(座標移動、ミラーリング、スケーリング)を座標変換で考慮することができます。

- MD24905 $MC_TRANSMIT_ROT_AX_FRAME_1 = 1
- MD24955 $MC_TRANSMIT_ROT_AX_FRAME_2 = 1

たとえば、フレーム結合内のフレームでワークの傾斜位置を補正することによって、回転軸オフセットを入力できます。一般に、このオフセットも、回転軸内のオフセットとして座標変換に組み込むことができます。上記図のような c 軸オフセットにより、対応する x 値と y 値が生成されます。

- MD24905 $MC_TRANSMIT_ROT_AX_FRAME_1 = 2
- MD24955 $MC_TRANSMIT_ROT_AX_FRAME_2 = 2

この設定では、SZS までの座標変換で、回転軸の軸別オフセットが考慮されます。SZS フレームに含まれている回転軸の軸別オフセットは、座標変換フレームに回転として入力されます。この設定が有効なのは、座標変換フレームが設定されている場合に限ります。

フレームの拡張:

以下に説明する拡張は、以下のマシンデータ設定の場合にのみ有効です。

- MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 1
- MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 2

座標変換 TRANSMIT の選択により、仮想ジオメトリ軸が生成され、この軸は、回転軸を介して連結され、輪郭フレームにだけ含まれていますが、軸別フレームには関係ありません。回転軸が回転すると、ジオメトリ値が生成されます。座標変換が選択されると、他のジオメトリ軸はすべて、その軸別コンポーネントを受け入れます。

コンポーネント:

- 平行移動
  TRANSMIT を選択すると、仮想軸の平行移動が削除されます。座標変換では、回転軸の平行移動を考慮できます。
- 回転
  座標変換前の回転を引き継ぎます。

- ミラーリング
  仮想軸のミラーリングが削除されます。座標変換では、回転軸のミラーリングを考慮できます。
- スケーリング
  仮想軸のスケーリングが削除されます。座標変換では、回転軸のスケーリングを考慮できます。

例:マシンデータ

```
; FRAME 設定

$MC_MM_SYSTEM_FRAME_MASK = 'H41'              ; TRAFRAME、SETFRAME
$MC_CHSFRAME_RESET_MASK = 'H41'               ; リセット後、フレームが有効になります。
$MC_CHSFRAME_POWERON_MASK = 'H41'             ; 電源投入時にフレームが削除されます。

$MN_FRAME_GEOAX_CHANGE_MODE = 1               ; ジオメトリ軸の切り替え後、フレームが計算されます。
$MC_RESET_MODE_MASK = 'H4041'                 ; リセット後、基本フレームは解除されません。
;$MC_RESET_MODE_MASK = 'H41'                  ; リセット後、基本フレームが解除されます。

;$MC_GCODE_RESET_VALUES[7] = 2                ; G54 が初期設定です。
$MC_GCODE_RESET_VALUES[7] = 1                 ; G500 が初期設定です。

$MN_MM_NUM_GLOBAL_USER_FRAMES = 0
$MN_MM_NUM_GLOBAL_BASE_FRAMES = 3

$MC_MM_NUM_USER_FRAMES = 10                   ; 5 ～ 100
$MC_MM_NUM_BASE_FRAMES = 3                    ; 0 ～ 8

$MN_NCBFRAME_RESET_MASK = 'HFF'
$MC_CHBFRAME_RESET_MASK = 'HFF'

$MN_MIRROR_REF_AX = 0                         ; ミラーリング時にスケーリングは行われません。
$MN_MIRROR_TOGGLE = 0
$MN_MM_FRAME_FINE_TRANS = 1                   ; 仕上げオフセット
$MC_FRAME_ADD_COMPONENTS = TRUE               ; G58、G59 が可能です。
```

```
; 最初の座標変換は TRANSMIT です。
$MC_TRAFO_TYPE_1=256

$MC_TRAFO_AXES_IN_1[0] = 1
$MC_TRAFO_AXES_IN_1[1] = 6
$MC_TRAFO_AXES_IN_1[2] = 3
```

```
$MC_TRAFO_AXES_IN_1[3] = 0
$MC_TRAFO_AXES_IN_1[4] = 0

$MA_ROT_IS_MODULO[AX6] = TRUE;

$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=1
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=6
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3

$MC_TRANSMIT_BASE_TOOL_1[0]=0.0
$MC_TRANSMIT_BASE_TOOL_1[1]=0.0
$MC_TRANSMIT_BASE_TOOL_1[2]=0.0

$MC_TRANSMIT_ROT_AX_OFFSET_1=0.0
$MC_TRANSMIT_ROT_SIGN_IS_PLUS_1=TRUE

$MC_TRANSMIT_ROT_AX_FRAME_1=1


; 2 番目の座標変換は TRANSMIT です。
$MC_TRAFO_TYPE_2=256

$MC_TRAFO_AXES_IN_2[0] = 1
$MC_TRAFO_AXES_IN_2[1] = 6
$MC_TRAFO_AXES_IN_2[2] = 2
$MC_TRAFO_AXES_IN_2[3] = 0
$MC_TRAFO_AXES_IN_2[4] = 0

$MC_TRAFO_GEOAX_ASSIGN_TAB_2[0] = 1
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[1] = 6
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[2] = 2

$MC_TRANSMIT_BASE_TOOL_2[0] = 4.0
$MC_TRANSMIT_BASE_TOOL_2[1] = 0.0
$MC_TRANSMIT_BASE_TOOL_2[2] = 0.0

$MC_TRANSMIT_ROT_AX_OFFSET_2=19.0
$MC_TRANSMIT_ROT_SIGN_IS_PLUS_2=TRUE

$MC_TRANSMIT_ROT_AX_FRAME_2=1
```

例:パートプログラム

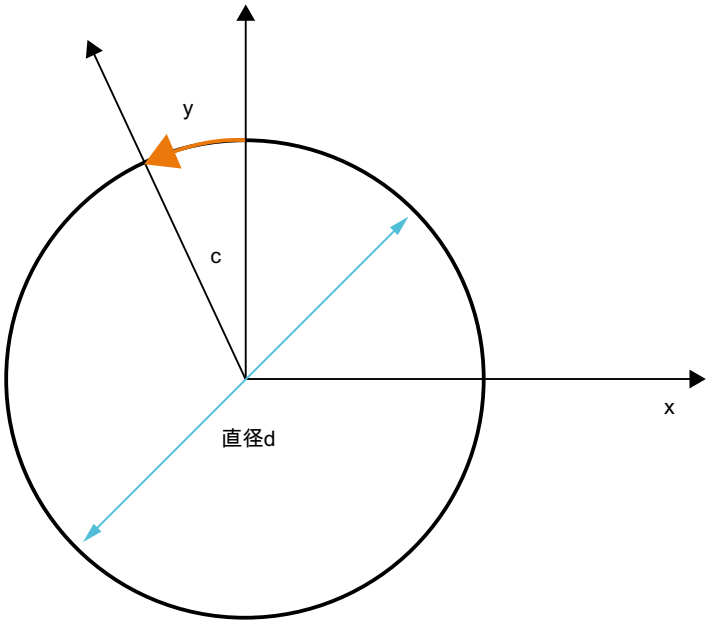
```
; フレーム設定
N820 $P_UIFR[1] = ctrans(x,1,y,2,z,3,c,4)
N830 $P_UIFR[1] = $P_UIFR[1] :crot(x,10,y,20,z,30)
N840 $P_UIFR[1] = $P_UIFR[1] :cmirror(x,c)
N850
N860 $P_CHBFR[0] = ctrans(x,10,y,20,z,30,c,15)
N870


; 工具の選択、クランプ補正、平面選択
N890 T2 D1 G54 G17 G90 F5000 G64 SOFT
N900


;開始位置へのアプローチ
N920 G0 X20 Z10
N930
N940 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,C,15)
N950 setal(61000)
N960 endif
N970 if $P_BFRAME <> $P_CHBFR[0]
N980 setal(61000)
N990 endif
N1000 if $P_IFRAME <>
CTRANS(X,1,Y,2,Z,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1010 setal(61000)
N1020 endif
N1030 if $P_IFRAME <> $P_UIFR[1]
N1040 setal(61000)
N1050 endif
N1060 if $P_ACTFRAME <>
CTRANS(X,11,Y,22,Z,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1070 setal(61000)
N1080 endif
N1090
N1100 TRANSMIT(2)
N1110
N1120 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,20,CAZ,30,C,15)
N1130 setal(61000)
N1140 endif
N1180 if $P_IFRAME <>
```

```
CTRANS(X,1,Y,0,Z,2,CAZ,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1190 setal(61000)
N1200 endif
N1240 if $P_ACTFRAME <>
CTRANS(X,11,Y,0,Z,22,CAZ,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1250 setal(61001)
N1260 endif
N1270
N1280
N1290 $P_UIFR[1,x,tr] = 11
N1300 $P_UIFR[1,y,tr] = 14
N1310
N1320 g54
N1330


;フレームの設定
N1350 ROT RPL=-45
N1360 ATRANS X-2 Y10
N1370


;四つ角の荒削り
N1390 G1 X10 Y-10 G41 OFFN=1; 削り代 1 mm
N1400 X-10
N1410 Y10
N1420 X10
N1430 Y-10
N1440


; 工具交換
N1460 G0 Z20 G40 OFFN=0
N1470 T3 D1 X15 Y-15
N1480 Z10 G41
N1490


; 四つ角の仕上げ
N1510 G1 X10 Y-10
N1520 X-10
N1530 Y10
N1540 X10
```

```
N1550 Y-10
N1560


; フレームの解除
 N2950 m30 N1580 Z20 G40
N1590 TRANS
N1600
N1610 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,20,CAZ,30,C,15)
N1620 setal(61000)
N1630 endif
N1640 if $P_BFRAME <> $P_CHBFR[0]
N1650 setal(61000)
N1660 endif
N1670 if $P_IFRAME <>
TRANS(X,11,Y,0,Z,2,CAZ,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1680 setal(61000)
N1690 endif
N1730 if $P_ACTFRAME <>
TRANS(X,21,Y,0,Z,22,CAZ,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1740 setal(61001)
N1750 endif
N1760
N1770 TRAFOOF
N1780
N1790 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,C,15)
N1800 setal(61000)
N1810 endif
N1820 if $P_BFRAME <> $P_CHBFR[0]
N1830 setal(61000)
N1840 endif
N1850 if $P_IFRAME <>
TRANS(X,11,Y,2,Z,3,C,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1860 setal(61000)
N1870 endif
N1880 if $P_IFRAME <> $P_UIFR[1]
N1890 setal(61000)
N1900 endif
N1910 if $P_ACTFRAME <>
TRANS(X,21,Y,22,Z,33,C,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1920 setal(61002)
N1930 endif
```

```
N1940
N2010 $P_UIFR[1] = ctrans()
N2011 $P_CHBFR[0] = ctrans()
N2020 $P_UIFR[1] = ctrans(x,1,y,2,z,3,c,0)
N2021 G54
N2021 G0 X20 Y0 Z10 C0
N2030 TRANSMIT(1)
N2040 TRANS x10 y20 z30
N2041 ATRANS y200
N2050 G0 X20 Y0 Z10
N2051 if $P_IFRAME <> CTRANS(X,1,Y,0,Z,3,CAY,2)
N2052 setal(61000)
N2053 endif
N2054 if $P_ACTFRAME <> CTRANS(X,11,Y,20,Z,33,CAY,2):CFINE(Y,200)
N2055 setal(61002)
N2056 endif
N2060 TRAFOOF
N2061 if $P_IFRAME <> $P_UIFR[1]
N2062 setal(61000)
N2063 endif
N2064 if $P_ACTFRAME <> CTRANS(X,11,Y,2,Z,33):CFINE(Y,0)
N2065 setal(61002)
N2066 endif
```

11.5.6.4 座標変換の選択と選択解除:TRACYL



TRACYL の拡張:

下記のマシンデータを使用して、座標変換で、TRACYL 回転軸の軸別フレーム一式(平行移動、仕上げオフセット、ミラーリング、スケーリング)を考慮することができます。

- MD24805 $MC_TRACYL_ROT_AX_FRAME_1 = 1
- MD24855 $MC_TRACYL_ROT_AX_FRAME_2 = 1

たとえば、フレーム結合内のフレームでワークの傾斜位置を補正することによって、回転軸オフセットを入力できます。一般に、このオフセットは、回転軸内のオフセットとして、または y オフセットとして座標変換に組み込むことができます。上記図のような c 軸オフセットにより、対応する x 値と y 値が生成されます。

- MD24805 $MC_TRACYL_ROT_AX_FRAME_1 = 2
- MD24855 $MC_TRACYL_ROT_AX_FRAME_2 = 2

この設定では、SZS までの座標変換で、回転軸の軸別オフセットが考慮されます。SZS フレームに含まれている回転軸の軸オフセットが、座標変換フレームに円筒面上のオフセットとして入力されます。この設定が有効なのは、座標変換フレームが設定されている場合に限ります。

フレームの拡張:

以下に説明する拡張は、以下のマシンデータ設定の場合にのみ有効です。

- MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 1
- MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 2

座標変換 TRACYL の選択により、仮想ジオメトリ軸が円筒面上に生成され、この軸は、回転軸を介して連結され、軸別フレームには関係なく、輪郭フレームでだけ考慮されます。仮想ジオメトリ軸のコンポーネントがすべて削除されます。座標変換が選択されると、他のジオメトリ軸はすべて、その軸別コンポーネントを受け入れます。

コンポーネント:

- 平行移動
  TRACYL を選択すると、仮想軸の平行移動が削除されます。座標変換では、回転軸の平行移動を考慮できます。
- 回転
  座標変換前の回転を引き継ぎます。

- ミラーリング
  仮想軸のミラーリングが削除されます。座標変換では、回転軸のミラーリングを考慮できます。
- スケーリング
  仮想軸のスケーリングが削除されます。座標変換では、回転軸のスケーリングを考慮できます。

例:マシンデータ

```
; フレーム設定

$MC_MM_SYSTEM_FRAME_MASK = 'H41'          ; TRAFRAME、SETFRAME
$MC_CHSFRAME_RESET_MASK = 'H41'           ; リセット後、フレームが有効になります。
$MC_CHSFRAME_POWERON_MASK = 'H41'         ; 電源投入時にフレームが削除されます。

$MN_FRAME_GEOAX_CHANGE_MODE = 1           ; ジオメトリ軸の切り替え後、フレームが計算されます。

$MC_RESET_MODE_MASK = 'H4041'             ; リセット後、基本フレームは解除されません。
;$MC_RESET_MODE_MASK = 'H41'              ; リセット後、基本フレームが解除されます。

;$MC_GCODE_RESET_VALUES[7] = 2            ; G54 が初期設定です。
$MC_GCODE_RESET_VALUES[7] = 1             ; G500 が初期設定です。

$MN_MM_NUM_GLOBAL_USER_FRAMES = 0
$MN_MM_NUM_GLOBAL_BASE_FRAMES = 3

$MC_MM_NUM_USER_FRAMES = 10               ; 5 ～ 100
$MC_MM_NUM_BASE_FRAMES = 3                ; 0 ～ 8

$MN_NCBFRAME_RESET_MASK = 'HFF'
$MC_CHBFRAME_RESET_MASK = 'HFF'

$MN_MIRROR_REF_AX = 0                     ; ミラーリング時にスケーリングは行われません。
$MN_MIRROR_TOGGLE = 0
$MN_MM_FRAME_FINE_TRANS = 1               ; 仕上げオフセット
$MC_FRAME_ADD_COMPONENTS = TRUE           ; G58、G59 が可能です。
```

```
; 3 番目の座標変換は、溝壁補正を使用する TRACYL です。

$MC_TRAFO_TYPE_3 = 513;  TRACYL

$MC_TRAFO_AXES_IN_3[0] = 1
```

```
$MC_TRAFO_AXES_IN_3[1] = 5
$MC_TRAFO_AXES_IN_3[2] = 3
$MC_TRAFO_AXES_IN_3[3] = 2

$MC_TRAFO_GEOAX_ASSIGN_TAB_3[0] = 1
$MC_TRAFO_GEOAX_ASSIGN_TAB_3[1] = 5
$MC_TRAFO_GEOAX_ASSIGN_TAB_3[2] = 3

$MC_TRACYL_BASE_TOOL_1[0] = 0.0
$MC_TRACYL_BASE_TOOL_1[1] = 0.0
$MC_TRACYL_BASE_TOOL_1[2] = 0.0

$MC_TRACYL_ROT_AX_OFFSET_1 = 0.0
$MC_TRACYL_ROT_SIGN_IS_PLUS_1 = TRUE

$MC_TRACYL_ROT_AX_FRAME_1 = 1
```

例:パートプログラム

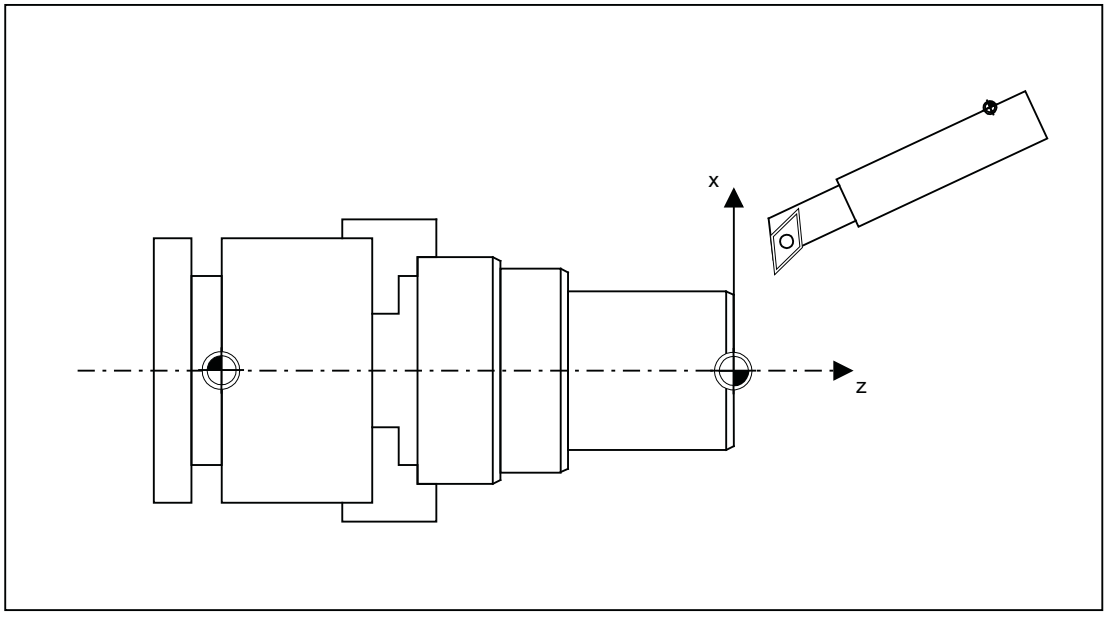
```
;溝側面オフセットのある単純移動テスト
N450 G603
N460


; フレーム設定
N500 $P_UIFR[1] = ctrans(x,1,y,2,z,3,b,4)
N510 $P_UIFR[1] = $P_UIFR[1] :CROT(x,10,y,20,z,30)
N520 $P_UIFR[1] = $P_UIFR[1] :CMIRROR(x,b)
N530
N540 $P_CHBFR[0] = ctrans(x,10,y,20,z,30,b,15)
N550
N560 G54
N570


; スムージングを選択した連続軌跡モード
N590 G0 x0 y0 z-10 b0 G90 F50000 T1 D1 G19 G641 ADIS=1 ADISPOS=5
N600
N610 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,B,15)
N620 setal(61000)
N630 endif
N640 if $P_BFRAME <> $P_CHBFR[0]
N650 setal(61000)
N660 endif
```

```
N670 if $P_IFRAME <>
TRANS(X,1,Y,2,Z,3,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N680 setal(61000)
N690 endif
N700 if $P_IFRAME <> $P_UIFR[1]
N710 setal(61000)
N720 endif
N730 if $P_ACTFRAME <>
TRANS(X,11,Y,22,Z,33,B,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N740 setal(61000)
N750 endif
N760


; 座標変換オン
N780 TRACYL(40.)
N790
N800 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,30,CAY,20,B,15)
N810 setal(61000)
N820 endif
N830 if $P_CHBFR[0] <> CTRANS(X,10,Y,0,Z,30,CAY,20,B,15)
N840 setal(61000)
N850 endif
N860 if $P_IFRAME <>
TRANS(X,1,Y,0,Z,3,CAY,2,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N870 setal(61000)
N880 endif
N890 if $P_UIFR[1] <>
TRANS(X,1,Y,0,Z,3,CAY,2,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N900 setal(61000)
N910 endif
N920 if $P_ACTFRAME <>
TRANS(X,11,Y,0,Z,33,CAY,22,B,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N930 setal(61001)
N940 endif
N950
N960 $P_UIFR[1,x,tr] = 11
N970 $P_UIFR[1,y,tr] = 14
N980
N990 g54
N1000
N1010 if $P_BFRAME <> CTRANS(X,10,Y,0,Z,30,CAY,20,B,15)
```

```
N1020 setal(61000)
N1030 endif
N1040 if $P_BFRAME <> $P_CHBFR[0]
N1050 setal(61000)
N1060 endif
N1070 if $P_IFRAME <>
TRANS(X,11,Y,0,Z,3,CAY,2,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N1080 setal(61000)
N1090 endif
N1100 if $P_IFRAME <> $P_UIFR[1]
N1110 setal(61000)
N1120 endif
N1130 if $P_ACTFRAME <>
TRANS(X,21,Y,0,Z,33,CAY,22,B,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N1140 setal(61001)
N1150 endif
N1160


; 座標変換オフ
N1180 TRAFOOF
N1190
N1200 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,B,15)
N1210 setal(61000)
N1220 endif
N1230 if $P_BFRAME <> $P_CHBFR[0]
N1240 setal(61000)
N1250 endif
N1260 if $P_IFRAME <>
TRANS(X,11,Y,2,Z,3,B,4):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N1270 setal(61000)
N1280 endif
N1290 if $P_IFRAME <> $P_UIFR[1]
N1300 setal(61000)
N1310 endif
N1320 if $P_ACTFRAME <>
TRANS(X,21,Y,22,Z,33,B,19):CROT(X,10,Y,20,Z,30):CMIRROR(X,B)
N1330 setal(61002)
N1340 endif
N1350
N1360 G00 x0 y0 z0 G90
N1370
```

```
N1380 m30
```

11.5.6.5 座標変換の選択と選択解除:TRAANG



フレームの拡張:

以下に説明する拡張は、以下のマシンデータ設定の場合にのみ有効です。

- MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 1
- MD10602 $MN_FRAME_GEOAX_CHANGE_MODE = 2

コンポーネント:

- 平行移動
  TRAANG を選択すると、仮想軸の平行移動が削除されます。
- 回転
  座標変換前の回転を引き継ぎます。
- ミラーリング
  仮想軸のミラーリングが引き継がれます。
- スケーリング
  仮想軸のスケーリングが引き継がれます。

例:マシンデータ

```
; フレーム設定
```

```
$MC_MM_SYSTEM_FRAME_MASK = 'H1'                ; SETFRAME
$MC_CHSFRAME_RESET_MASK = 'H41'                ; リセット後、フレームが有効になります。
$MC_CHSFRAME_POWERON_MASK = 'H41'              ; 「電源投入時」にフレームが削除されます。

$MN_FRAME_GEOAX_CHANGE_MODE = 1                ; ジオメトリ軸の切り替え後、フレームが計算されます。


$MC_RESET_MODE_MASK = 'H4041'                  ; リセット後、基本フレームは解除されません。
;$MC_RESET_MODE_MASK = 'H41'                   ; リセット後、基本フレームが解除されます。

;$MC_GCODE_RESET_VALUES[7] = 2                 ; G54 が初期設定です。
$MC_GCODE_RESET_VALUES[7] = 1                  ; G500 が初期設定です。

$MN_MM_NUM_GLOBAL_USER_FRAMES = 0
$MN_MM_NUM_GLOBAL_BASE_FRAMES = 3

$MC_MM_NUM_USER_FRAMES = 10                    ; 5 ～ 100
$MC_MM_NUM_BASE_FRAMES = 3                     ; 0 ～ 8

$MN_NCBFRAME_RESET_MASK = 'HFF'
$MC_CHBFRAME_RESET_MASK = 'HFF'

$MN_MIRROR_REF_AX = 0                          ; ミラーリング時にスケーリングは行われません。
$MN_MIRROR_TOGGLE = 0
$MN_MM_FRAME_FINE_TRANS = 1                    ; 仕上げオフセット
$MC_FRAME_ADD_COMPONENTS = TRUE                ; G58、G59 が可能です。
```

```
; 最初の座標変換は TRAANG です。

$MC_TRAFO_TYPE_1 = 1024

$MC_TRAFO_AXES_IN_1[0] = 4                     ; 傾斜軸
$MC_TRAFO_AXES_IN_1[1] = 3                     ; 軸が z に平行
$MC_TRAFO_AXES_IN_1[2] = 2
$MC_TRAFO_AXES_IN_1[3] = 0
$MC_TRAFO_AXES_IN_1[4] = 0

$MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]=4
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]=2
$MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]=3

$MC_TRAANG_ANGLE_1 = 85.
$MC_TRAANG_PARALLEL_VELO_RES_1 = 0.
$MC_TRAANG_PARALLEL_ACCEL_RES_1 = 0.
```

```
$MC_TRAANG_BASE_TOOL_1[0] = 0.0
$MC_TRAANG_BASE_TOOL_1[1] = 0.0
$MC_TRAANG_BASE_TOOL_1[2] = 0.0
```

; 2 番目の座標変換は TRAANG です。

```
$MC_TRAFO_TYPE_2 = 1024

$MC_TRAFO_AXES_IN_2[0] = 4
$MC_TRAFO_AXES_IN_2[1] = 3
$MC_TRAFO_AXES_IN_2[2] = 0
$MC_TRAFO_AXES_IN_2[3] = 0
$MC_TRAFO_AXES_IN_2[4] = 0

$MC_TRAFO_GEOAX_ASSIGN_TAB_2[0] = 4
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[1] = 0
$MC_TRAFO_GEOAX_ASSIGN_TAB_2[2] = 3

$MC_TRAANG_ANGLE_2 = -85.
$MC_TRAANG_PARALLEL_VELO_RES_2 = 0.2
$MC_TRAANG_PARALLEL_ACCEL_RES_2 = 0.2

$MC_TRAANG_BASE_TOOL_2[0] = 0.0
$MC_TRAANG_BASE_TOOL_2[1] = 0.0
$MC_TRAANG_BASE_TOOL_2[2] = 0.0
```

例:パートプログラム

```
; フレーム設定
N820 $P_UIFR[1] = ctrans(x,1,y,2,z,3,b,4,c,5)
N830 $P_UIFR[1] = $P_UIFR[1] :crot(x,10,y,20,z,30)
N840 $P_UIFR[1] = $P_UIFR[1] :cmirror(x,c)
N850
N860 $P_CHBFR[0] = ctrans(x,10,y,20,z,30,b,40,c,15)
N870


; 工具の選択、クランプ補正、平面選択
N890 T2 D1 G54 G17 G90 F5000 G64 SOFT
N900
```

```
; 開始位置へのアプローチ
N920 G0 X20 Z10
N930
N940 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,B,40,C,15)
N950 setal(61000)
N960 endif
N970 if $P_BFRAME <> $P_CHBFR[0]
N980 setal(61000)
N990 endif
N1000 if $P_IFRAME <>
TRANS(X,1,Y,2,Z,3,B,4,C,5):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1010 setal(61000)
N1020 endif
N1030 if $P_IFRAME <> $P_UIFR[1]
N1040 setal(61000)
N1050 endif
N1060 if $P_ACTFRAME <>
TRANS(X,11,Y,22,Z,33,B,44,C,20):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1070 setal(61000)
N1080 endif
N1090
N1100 TRAANG(,1)
N1110
N1120 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,CAX,10,B,40,C,15)
N1130 setal(61000)
N1140 endif
N1150 if $P_BFRAME <> $P_CHBFR[0]
N1160 setal(61000)
N1170 endif
N1180 if $P_IFRAME <>
CTRANS(X,1,Y,2,Z,3,CAX,1,B,4,C,5):CROT(X,10,Y,20,Z,30):CMIRROR(X,CAX,C)
N1190 setal(61000)
N1200 endif
N1210 if $P_IFRAME <> $P_UIFR[1]
N1220 setal(61000)
N1230 endif
N1240 if $P_ACTFRAME <>
TRANS(X,11,Y,22,Z,33,CAX,11,B,44,C,20):CROT(X,10,Y,20,Z,
30):CMIRROR(X,CAX,C)
N1250 setal(61001)
N1260 endif
```

```
N1270
N1280
N1290 $P_UIFR[1,x,tr] = 11
N1300 $P_UIFR[1,y,tr] = 14
N1310
N1320 g54
N1330


;フレームの設定
N1350 ROT RPL=-45
N1360 ATRANS X-2 Y10
N1370


;四つ角の荒削り
N1390 G1 X10 Y-10 G41 OFFN=1; 削り代 1 mm
N1400 X-10
N1410 Y10
N1420 X10
N1430 Y-10
N1440


; 工具交換
N1460 G0 Z20 G40 OFFN=0
N1470 T3 D1 X15 Y-15
N1480 Z10 G41
N1490


; 四つ角の仕上げ
N1510 G1 X10 Y-10
N1520 X-10
N1530 Y10
N1540 X10
N1550 Y-10
N1560


; フレームの解除
N1580 Z20 G40
N1590 TRANS
```

```
N1600
N1610 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,CAX,10,B,40,C,15)
N1620 setal(61000)
N1630 endif
N1640 if $P_BFRAME <> $P_CHBFR[0]
N1650 setal(61000)
N1660 endif
N1670 if $P_IFRAME <>
TRANS(X,11,Y,14,Z,3,CAX,1,B,4,C,5):CROT(X,10,Y,20,Z,30):CMIRROR(X,CAX,C)
N1680 setal(61000)
N1690 endif
N1700 if $P_IFRAME <> $P_UIFR[1]
N1710 setal(61000)
N1720 endif
N1730 if $P_ACTFRAME <>
TRANS(X,21,Y,34,Z,33,CAX,11,B,44,C,20):CROT(X,10,Y,20,Z,
30):CMIRROR(X,CAX,C)
N1740 setal(61001)
N1750 endif
N1760
N1770 TRAFOOF
N1780
N1790 if $P_BFRAME <> CTRANS(X,10,Y,20,Z,30,B,40,C,15)
N1800 setal(61000)
N1810 endif
N1820 if $P_BFRAME <> $P_CHBFR[0]
N1830 setal(61000)
N1840 endif
N1850 if $P_IFRAME <>
TRANS(X,1,Y,14,Z,3,B,4,C,5):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1860 setal(61000)
N1870 endif
N1880 if $P_IFRAME <> $P_UIFR[1]
N1890 setal(61000)
N1900 endif
N1910 if $P_ACTFRAME <>
TRANS(X,11,Y,34,Z,33,B,44,C,20):CROT(X,10,Y,20,Z,30):CMIRROR(X,C)
N1920 setal(61002)
N1930 endif
N1940
N1950 m30
```

11.5.6.6 有効なフレームの補正

ジオメトリ軸の座標は、プログラムの処理中に、または RESET により変化する場合があります。既存のジオメトリ軸の数は、0 ～ 3 のいずれかになります。存在しないジオメトリ軸では、有効なフレーム内のコンポーネント (たとえば、回転)を使用すると、これらの軸の座標のための有効なフレームが無効になる場合があります。これは、「チャネル%1 ブロック%2 の回転がプログラムされているジオメトリ軸は存在しません。」アラームで示されます。このアラームは、各フレームが適切に変更されるまで、表示されたままです。

有効なフレームの自動補正を、以下のマシンデータで起動することができます。

MD24040 $MC_FRAME_ADAPT_MODE、ビット<n> = <値>

ビット	<値>	意味
0	1	ジオメトリ軸を持たない座標軸を回転する有効なフレーム内の回転が、有効なフレームから削除されます。
1	1	有効なフレーム内のせん断角が直交します。
2	1	有効なフレーム内のジオメトリ軸すべての単位変換係数が値 1 に設定されます。

存在しないジオメトリ軸で座標軸移動を引き起こす可能性のある、有効なフレームのすべての回転は、以下のマシンデータで削除します。

MD24040 $MC_FRAME_ADAPT_MODE = 1

この処理中に、データ管理フレームは変更されません。データ管理フレームを起動するために、可能な回転のみが受け入れられます。

例:

y 軸が存在しない場合:

- MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[0] = 1
- MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[1] = 0
- MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB[2] = 3
- $P_UIFR[1] = crot(x,45,y,45,z,45)

プログラムコード	; コメント
N390 G54 G0 X10 Z10 F10000	
IF $P_IFRAME <> CROT(Y,45)	; Y を中心にした回転のみが転送されます。
SETAL(61000)	
ENDIF	

11.5.6.7 マッピングされたフレーム

概要

「マッピングされたフレーム」機能は、チャネル別データ管理フレームまたはグローバルデータ管理フレームでおこなわれる、チャネル間の軸別フレームの一貫した変更をサポートします。軸別マシンデータを使用して、どの軸の間でマッピングを実現するかが定義されます。

たとえば、フレームのマッピングが機械軸 AX1 と AX4 に対して有効で、軸 AX1 の軸別フレームがチャネル別データ管理フレーム(基本フレーム$P_CHBFR[x]など)で変更される場合(座標移動、精密座標移動、スケーリング、ミラーリング)、AX1 と AX4 のフレームデータは、チャネル軸としてパラメータ設定されているすべてのチャネル内のすべてのチャネル別データ管理フレーム(基本フレーム$P_CHBFR[x]など)に転送されます。

回転のために軸別フレームデータを変更する場合、フレームのマッピングは実現されません。

必要条件

フレームのマッピングでは、以下の必要条件を満たす必要があります。

- マッピングに使用するデータ管理フレームが設定されていること。
  MD28083 $MC_MM_SYSTEM_DATAFRAME_MASK (システムフレーム)
- **チャネル別**データ管理フレームがマッピングに対して明示的に有効になっていること。
  MD10616 $MN_MAPPED_FRAME_MASK (フレームのマッピングの有効化)

  注
  グローバルデータ管理フレームの場合、マッピングは常に実行されます。有効化は必要ありません。

パラメータ設定

マッピング関係のパラーメータ設定は、軸マシンデータで実現します。

MD32075 $MA_MAPPED_FRAME[<AXn>] = "AXm"

AXn、AXm:機械軸名称。n、m = 1、2、...機械軸の最大数

マッピング規則

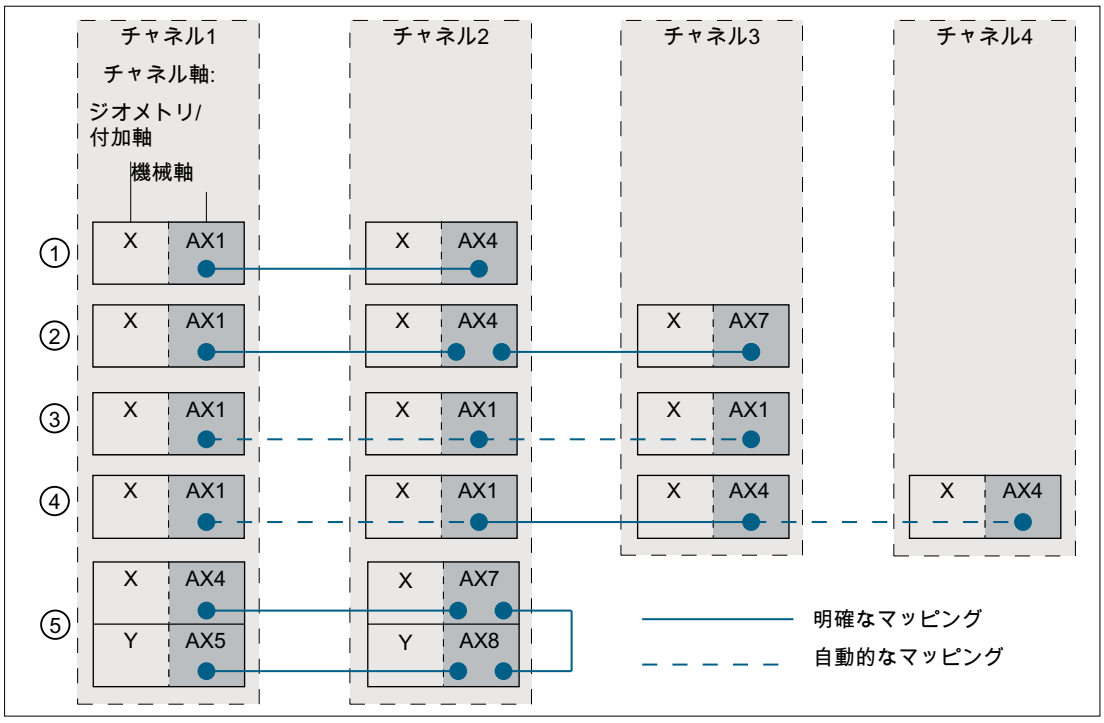
フレームのマッピングには以下の規則が適用されます。

- マッピングは双方向です。
  軸別フレームは、軸 AXn または AXm に対して書き込むことができます。フレームデータは常に、他の軸に反映され、引き継がれます。
- パラメータ設定されたすべてのマッピング関係は、常に評価されます。
  軸 AXn の軸別フレームを書き込むと、すべてのマッピング関係が評価され、直接的と間接的に関係するすべての軸に対してフレームデータが反映されます。
- マッピングはすべてのチャネル間でグローバルです。
  軸 AXn または AXm の軸別フレームをチャネル別フレームに対して書き込むと、AXn または AXm がチャネル軸としてパラメータ設定されているすべてのチャネルに対して、フレームデータが反映されます。
- ジオメトリ軸名称または付加軸名称を使用して軸別フレームを書き込む場合、マッピング関係は、ジオメトリ軸または付加軸に現在割り当てられている機械軸経由で評価されます。
- マッピングはフレーム別に行われます。
  軸別フレームを書き込むと、フレームデータは同じチャネル別データ管理フレームまたはグローバルデータ管理フレーム内でだけマップされます。

注記

データの適合性

ユーザー/工作機械メーカは、フレームの書き込み後、チャネル同期化の使用などによって、すべてのチャネルで一貫したフレームデータが使用されていることを確認してください。



説明	パラメータ設定: $MA_
① 簡単なマッピング関係 AX1(K1) ↔ AX4(K2)	MAPPED_FRAME[<AX1>] = "AX4"
② 結合されたマッピング関係 AX1(K1) ↔ AX4(K2) ↔ AX7(K3)	MAPPED_FRAME[<AX1>] = "AX4" MAPPED_FRAME[<AX4>] = "AX7"
③ AX1 をチャネル軸としたチャネル 1、2、3 の単独のマッピング関係 AX1(K1+K2+K3)	MAPPED_FRAME[<AX1>] = "AX1"
④ 2 つの軸(2 つのチャネル内のチャネル軸)の間のマッピング関係 AX1(K1+K2) ↔ AX4(K3+K4)	MAPPED_FRAME[<AX1>] = "AX4"
⑤ 複数のチャネル軸を同一のチャネルに書き込み可能な、結合された割り当ての関係 AX4(K1) ↔ AX7(K2) ↔ AX8(K2) ↔ AX5(K1)	MAPPED_FRAME[<AX4>] = "AX7" MAPPED_FRAME[<AX7>] = "AX8" MAPPED_FRAME[<AX8>] = "AX5"

図 11-25　割り当ての例

データ管理フレームの設定

データ管理フレームはパートプログラムで書き込むか、SINUMERIK Operate のユーザーインタフェースを使用して書き込むことができます。直接書き込まれたか、フレームのマッピングで書き込まれたデータ管理フレームを複数のチャネルで設定する場合は、以下に注意してください。

- パートプログラムで書き込んだ場合
  データ管理フレームは、チャネル毎に明確に設定してください(G500、G54 ... G599)
- ユーザーインターフェースを介して書き込んだ場合
  データ管理フレームは、新しいワークオフセットの入力などによって、ユーザーインターフェースを介して書き込まれています。関係するチャネルがどれも「チャネル起動中」状態でなければ、修正されたデータ管理フレームは関係するすべてのチャネルですぐに有効になります。関係するチャネルのどれかが「チャネル起動中」状態の場合、データ管理フレームはどのチャネルでも有効になりません。パート設定は、パートプログラムで、チャネル毎に意図的にプログラム指令してください(G500、G54 ... G599)。プログラムしない場合、次にチャネル状態が変更されたときに、データ管理フレームが「チャネルのリセット」後に有効になります。

例

以下のチャネルとチャネル軸が、コントローラでパラメータ設定されます。

- チャネル 1
  - Z:ジオメトリ軸
  - AX1:ジオメトリ軸 Z の機械軸
- チャネル 2
  - Z:ジオメトリ軸
  - AX4:ジオメトリ軸 Z の機械軸

Z 軸の原点は、両方のチャネルで常に同じです。

- マッピング関係: $MA_MAPPED_FRAME[AX1] = "AX4"

チャネル 1 および 2 のパートプログラム

チャネル 1	チャネル 2
...	...
N100 WAIT(10,1,2)	N200 WAIT(10,1,2)
N110 $P_UIFR[1] = CTRANS(Z, 10)	
N120 WAIT (20,1,2)	N220 WAIT (20,1,2)
N130 G54	N230 G54

チャネル 1	チャネル 2
N140 IF ($P_IFRAME[0, Z, TR] <> 10)	N230 IF ($P_IFRAME[0, Z, TR] <> 10)
N150 SETAL(61000)	N250 SETAL(61000)
N160 ENDIF	N260 ENDIF
...	...

説明:

N100 / N200 フレームデータの一貫した書き込みとマッピングのためのチャネル同期化

N110 設定可能なデータ管理フレーム$P_UIFR[1]の書き込み
Z 軸の原点を 10 mm に移動
軸別フレームデータのマッピング:
チャネル 1:Z ≙ AX1 ⇔ チャネル 2:Z ≙ AX4

N120 / N220 新しいフレームデータの一貫した設定のためのチャネル同期化

N130 / N230 新しいフレームデータの設定

N140 / N240 Z 軸の原点のチェック:10 mm

11.5.7 事前定義されたフレーム機能

11.5.7.1 逆フレーム

INVFRAME()機能は、適切な逆フレームをフレームから計算します。

機能説明

フレームとその逆フレーム間の結合では必ずゼロフレームが生成されます。
FRAME:INVFRAME(FRAME) ⇒ヌルフレーム

フレーム逆変換は座標変換を補助します。計測用フレームは通常 WCS で計算されます。この計算したフレームを別の座標系に変換したい場合、つまり計算したフレームを、フレーム結合内で必要なフレーム内に入力する必要がある場合は、以下の計算を使用してください。

新規フレーム一式は、旧フレーム一式と、計算したフレームの結合になります。

```
$P_ACTFRAME = $P_ACTFRAME: $AC_MEAS_FRAME
```

この結果、フレーム結合内の新規フレームは次のようになります。

- ターゲットフレームは$P_SETFRAME です。
  ```
  $P_SETFRAME = $P_ACTFRAME:
  $AC_MEAS_FRAME:INVFRAME($P_ACTFRAME): $P_SETFRAME
  ```
- ターゲットフレームは、n 番目のチャネル基本フレーム$P_CHBFRAME[<n>]です。
  ```
  k = $MN_MM_NUM_GLOBAL_BASE_FRAMES
  ```
  - n = 0 の場合は、TMP が、以下のように解決されます。
    ```
    TMP = $P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME :
    $P_NCBFRAME[<0...k>]
    ```
  - n ≠ 0 の場合は、TMP が、以下のように解決されます。
    ```
    TMP = $P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME :
    $P_NCBFRAME[<0..k>] : $P_CHBFRAME[<0...n> - 1]
    ```
  ```
  $P_CHBFRAME[<n>] = INVFRAME(TMP) : $P_ACTFRAME :
  $AC_MEAS_FRAME :INVFRAME($P_ACTFRAME) :TMP : $P_CHBFRAME[<n>]
  ```
- ターゲットフレームは$P_IFRAME です。
  ```
  TMP = $P_PARTFRAME: $P_SETFRAME: $P_EXTFRAME: $P_BFRAME
  $P_IFRAME = INVFRAME(TMP): $P_ACTFRAME:
  $AC_MEAS_FRAME:INVFRAME($P_ACTFRAME) :TMP : $P_IFRAME
  ```

例:

新規フレーム一式が、旧フレーム一式と計測フレームの結合になるように、たとえば計測用機能を使用して計算したフレームを、現在の SETFRAME に入力してください。フレーム逆変換を使用して、適宜 SETFRAME が計算されます。

プログラムコード	コメント
DEF INT RETVAL	
DEF FRAME TMP	
$TC_DP1[1,1] = 120	; タイプ
$TC_DP2[1,1]=20.	; 0
$TC_DP3[1,1]= 10.	; (z)長さ補正ベクトル
$TC_DP4[1,1]= 0.	; (y)
$TC_DP5[1,1]= 0.	; (x)
$TC_DP6[1,1]= 2.	; 半径
T1 D1	
g0 x0 y0 z0 f10000	
G54	
$P_CHBFRAME[0] = crot(z,45)	
$P_IFRAME[x,tr] = -sin(45)	
$P_IFRAME[y,tr] = -sin(45)	
$P_PFRAME[z,tr] = -45	

プログラムコード	コメント
$AC_MEAS_VALID = 0	; 4 つの計測ポイントによるコーナの計測
G1 X-1 Y-3	; 最初のアプローチ計測ポイント
$AC_MEAS_LATCH[0] = 1	; 最初の格納計測ポイント
G1 X5 Y-3	; 2.番目の計測点へアプローチ
$AC_MEAS_LATCH[1] = 1	; 2.格納計測ポイント
G1 X-4 Y4	; 3.番目の計測点へアプローチ
$AC_MEAS_LATCH[2] = 1	; 3.格納計測ポイント
G1 X-4 Y1	; 4.番目の計測点へアプローチ
$AC_MEAS_LATCH[3] = 1	; 4.格納計測ポイント
$AA_MEAS_SETPOINT[x] = 0	; コーナの位置指令値の設定
$AA_MEAS_SETPOINT[y] = 0	
$AA_MEAS_SETPOINT[z] = 0	
$AC_MEAS_CORNER_SETANGLE = 90	; 交差部分の指令角度の定義
$AC_MEAS_WP_SETANGLE = 30	
$AC_MEAS_ACT_PLANE = 0	; 計測平面は G17
$AC_MEAS_T_NUMBER = 1	; 工具選択
$AC_MEAS_D_NUMBER = 1	
$AC_MEAS_TYPE = 4	; コーナ 1 に関する計測タイプの設定
RETVAL = MEASURE()	; 計測処理の開始
IF RETVAL <> 0	
setal(61000 + RETVAL)	
ENDIF	
IF $AC_MEAS_WP_ANGLE <> 30	
setal(61000 + $AC_MEAS_WP_ANGLE)	
ENDIF	
IF $AC_MEAS_CORNER_ANGLE <> 90	
setal(61000 + $AC_MEAS_CORNER_ANGLE)	
ENDIF	
; 計測済みフレームを変換し、$P_SETFRAME に書き込んで、フレーム一式を作成します。	
; このフレーム一式が、旧フレーム一式からリンクされ、計測フレームが得られます。	
$P_SETFRAME = $P_ACTFRAME: $AC_MEAS_FRAME:INVFRAME($P_ACTFRAME): $P_SETFRAME	
$P_SETFR = $P_SETFRAME	; データ管理でシステムフレームを記述します
G1 X0 Y0	; コーナへのアプローチ
G1 X10	; 約 30 度回転した長方形を後退させます
Y10	
X0	
Y0	
M30	

11.5.7.2 フレーム結合内の追加フレーム

ワークの計測、あるいはパートプログラムまたはサイクルでの計算により、多くの場合、有効なフレーム一式に追加適用されるフレームが生成されます。これにより、たとえば、WCS およびプログラム指令された原点のずれや、場合によっては回転が発生します。そのため、計測済みフレームがフレーム変数内に存在し、まだ、フレーム結合の計算には含まれていません。

機能説明

ADDFRAME()機能は、一時フレームから、<STRING>パラメータで指定されたターゲットフレームを計算します。そのため、一次フレームを含む、有効な旧フレーム一式の結合からの新しい有効な$P_ACTFRAME フレーム一式から、以下が得られます。

ERG = ADDFRAME(TMPFRAME,"$P_SETFRAME") ⇒ $P_SETFRAME$_{new}$ = $P_SETFRAME$_{old}$ ADD TMPFRAME and $P_ACTFRAME$_{new}$ = $P_ACTFRAME$_{old}$:TMPFRAME

有効なフレームが、ターゲットフレームとして指定されている場合、新規フレーム一式は、**先読みの段階**で有効になります。

ターゲットフレームがデータ管理フレームの場合は、パートプログラムまたはサイクルにより明示的にチャネルで起動されたときにのみ有効になります。

機能により、ユーザーサイクル内など、ユーザー別の動作が可能な戻り値が返されます。

プログラミング

構文

INT ADDFRAME(<フレーム>,<文字列>)

意味

<フレーム>:	計算に追加で含まれる値を持つフレーム変数	
	タイプ	FRAME

<文字列>:	有効なフレームまたはデータ管理フレームの名前: ● 動作中のフレーム "$P_CYCFRAME"、"$P_ISO4FRAME"、"$P_PFRAME"、"$P_WPFRAME"、"$P_TOOLFRAME"、"$P_IFRAME"、"$P_GFRAME"、"$P_CHBFRAME[<n>]"、"$P_NCBFRAME[<n>]"、"$P_ISO1FRAME"、"$P_ISO2FRAME"、"$P_ISO3FRAME"、"$P_EXTFRAME"、"$P_SETFRAME"、"$P_PARTFRAME" ● データ管理フレーム "$P_CYCFR"、"$P_ISO4FR"、"$P_TRAFR"、"$P_WPFR"、"$P_TOOLFR"、"$P_UIFR[<n>]"、"$P_GFR"、"$P_CHBFR[<n>]"、"$P_NCBFR[<n>]"、"$P_ISO1FR"、"$P_ISO2FR"、"$P_ISO3FR"、"$P_EXTFR"、"$P_SETFR"、"$P_PARTFR"	
	タイプ	STRING
戻り値:	可能な戻り値: ● 0:OK ● 1:指定したターゲット(文字列)が間違っています ● 2:ターゲットフレームが設定されていません ● 3:フレームでの回転は許可されていません	
	タイプ	INT

11.5.8 動作

11.5.8.1 原点の設定、ワーク計測および工具計測

現在位置設定は、HMI オペレータ操作、または計測サイクルによって開始されます。計算したフレームを、システムフレーム SETFRAME に書き込むことができます。WCS で軸の位置指令は、現在値の設定時に変更できます。

接触計測という用語は、ワーク計測と工具計測の両方を意味します。エッジ、コーナー、または穴に関連するワークの位置を計測できます。ワークまたは穴の原点位置を特定するために、WCS 内の計測済み位置に位置指令値を追加できます。その結果のオフセットを、選択したフレームに入力できます。工具計測では、工具の長さまたは半径を、計測基準部品を使用して計測できます。

この計測は、オペレータ操作または計測サイクルによって開始できます。NC との通信が、予約システム変数によって行われます。この計算は、HMI オペレータ操作によって、または計測サイクルからパートプログラム命令によって PI サービスが起動された場合

に、NC で実行されます。工具と平面を、計算の基準として選択できます。計算されたフレームは、フレーム結果に入力されます。

11.5.8.2 軸別外部ゼロオフセット

マシンデータ

外部ゼロオフセットまたは$P_EXTFRAME システムフレームは、以下のマシンデータで起動します。

MD28082 $MC_MM_SYSTEM_FRAME_MASK、ビット 1 = TRUE

外部ゼロオフセットの大きさは、HMI ユーザーインターフェースおよび OPI 経由 PLC ユーザープログラムを介して手動で、または軸システム変数$AA_ETRANS[<軸>]を使ってパートプログラムでプログラム指令して指定できます。

起動

外部ゼロオフセットは、以下のインターフェース信号で起動します。
DB31、... DBX3.0 (外部ゼロオフセットの受け入れ)

動作

外部ゼロオフセットを起動すると、コマンド軸と PLC 軸を除く、すべての軸の移動動作が直ちに停止し、先読みが再解析されます。現在のシステムフレームの荒削りオフセットおよびデータ管理内のシステムフレームの荒削りオフセットは、軸システム変数$AA_ETRANS[<軸>]の値に設定されます。その後、まずこのオフセットを移動した後、割り込まれた動作が継続されます。

インクリメンタル指令での動作

インクリメンタル指令 G91 が有効で、マシンデータ:
MD42440 $MC_FRAME_OFFSET_INCR_PROG
が(フレーム内の原点オフセット) = 0 の場合は、このマシンデータがフレームで指定されていても、アプローチブロックにより、このマシンデータの逆の設定とは関係なく、システムフレームを介して外部ゼロオフセットの範囲内で、オフセットの移動が実行されます。

注記

外部ゼロオフセットは常にアブソリュートで動作します。

11.5.8.3 工具ホルダ

平行移動

タイプ「P」と「M」のキネマティックでは、工具ホルダの選択により追加フレームが有効になり(旋回工具ホルダのテーブルオフセット)、テーブルが回転した結果のゼロオフセットが考慮されます。ゼロオフセットはシステムフレーム$P_PARTFR に入力されます。この場合、このフレームの平行移動成分が上書きされます。他のフレーム成分は保持されます。

システムフレーム$P_PARTFR は、以下のマシンデータで有効にしてください。

MD28082 $MC_MM_SYSTEM_FRAME_MASK、ビット 2 = 1 (TCARR および PAROT 用システムフレーム)

注記

別の方法として、以下のマシンデータを使用してテーブルオフセットを入力してオフセットをパラメータ設定することもできます。

MD20184 $MC_TOCARR_BASE_FRAME_NUMBER = <基本フレームの数>

このオプションは、古いバージョンのソフトウェアとの互換性の理由でのみ使用されます。この処理は今後は使用**しない**ことを強く推奨します。

工具ホルダ変更の結果発生したフレームのオフセットは、TCARR=...を選択すると直ちに有効になります。一方、工具長の変更が直ちに有効になるのは、工具が有効になっている場合に限ります。

起動時にフレームの回転は実行されません。または、すでに動作中の回転は変更されません。T の(工具だけを回転できる)場合のように、計算に使用される回転軸の位置は、命令 TCOFR / TCOABS によって異なり、動作中のフレームの回転成分、または入力値 $TC_CARR から特定されます。フレームの起動により、ワーク座標系内の位置が適宜変化し、機械自体による補正移動はありません。

次の図に、その関係を記載します。

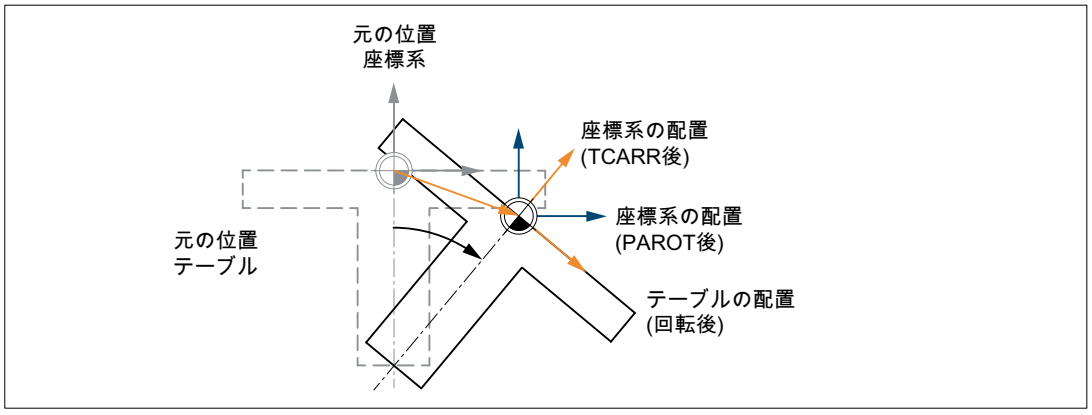


図 11-26　TCARR による回転テーブルの起動時のフレーム

タイプ M のキネマティック(工具とテーブルは各々 1 つの軸を中心に回転します)を使用すると、TCARR による工具ホルダの起動より、同時に、対応する変更が有効な工具長（工具が有効であれば）とゼロオフセット内に生成されます。

回転

機械加工処理によっては、旋回工具ホルダまたはテーブルの使用時のゼロオフセット(フレームと工具長のいずれとしてでも)だけでなく、回転も考慮する必要があります。ただし、旋回工具ホルダを起動しても、直ちに座標系が回転するわけではありません。

工具だけ回転できる場合、TOFRAME または TOROT を使用すれば、そのためにフレームを定義できます。

回転テーブル(キネマテックタイプ P と M)では、TCARR による起動も、同様に、座標系を直ちに回転させるわけではありません。つまり、座標系の原点が、機械に対してオフセットされても、ワークの原点に対して固定されたままであるかぎり、空間内で向きは変更されません。

座標系をワークに関して固定する必要がある場合、つまり、元の位置に対してオフセットするだけでなく、テーブルの回転に応じて回転させることも必要な場合、PAROT を使用すれば、回転工具を使用した状況と類似の方法で、こうした回転を起動できます。

PAROT を使用すると、有効なフレームでの平行移動、スケーリングおよびミラーリングが保持されます。ただし、回転成分は、テーブルに応じた旋回工具ホルダの回転成分により回転します。回転成分を含むプログラマブルフレーム全体は変更されません。

テーブルの回転を記述する回転成分は、システムフレーム$PARTFR または MD20184 $MC_TOCARR_BASE_FRAME_NUMBER で設定した基本フレームに入力されます。

$MC_MM_SYSTEM_FRAME_MASK、ビット 2 = <値>

値	意味
1	回転成分 → $PARTFR
0	回転成分 → MD20184 $MC_TOCARR_BASE_FRAME_NUMBER

テーブルオフセットの説明と同じように、新規システム用としては、ここに記載されている 2 番目の代案はお勧めできません。

パートフレームの回転成分は、PAROTOF を使用すれば、このフレームが検出されたのが基本フレーム内かシステムフレーム内かに関係なく、削除できます。
オフセットを生成しない工具ホルダが起動されるか、有効な旋回工具ホルダが TCARR=0 により選択解除されると、平行移動成分が削除されます。

2 つの回転軸によりテーブルまたは工具が回転する場合、PAROT または TOROT では、全ての向きの変更が考慮されます。複雑なキネマティックの場合は、回転軸に起因する対応成分だけが考慮されます。このため、たとえば、TOROT の使用時に、空間に固定された X-Y 面と平行に斜面が配置されるようにワークを回転させることができます。これにより、たとえばドリルで空ける穴がこの面に垂直にならない場所を加工する際に、この工具の回転を考慮してください。

例

機械上では、テーブルの回転軸は正の Y 方向を指します。テーブルは+45°だけ回転します。PAROT を使用すると、Y 軸を中心にした 45°の回転を同じように記述するフレームが定義されます。座標系は、実際の環境に対して回転するわけではなく(前述の図の「TCARR 後の座標系の位置」のマーク)、定義された座標系に対して-45°だけ回転します(PAROT 後の位置)。たとえば、この座標系が ROT Y-45 で定義され、工具ホルダが有効な TCOFR で選択されると、工具ホルダの回転軸として+45°の角度が特定されます。

旋回工具ホルダが有効になっていない場合、言語命令 PAROT は拒否されません。ただし、その後、こうした呼び出しによりフレーム変更はおこなわれません。

工具オリエンテーションの方向での加工

特に、向きを指定できる工具が装備された機械上では、軸の一つが工具の方向を指すフレームを起動せずに(たとえば、TOFRAME または TOROT を使用して)、移動を工具方向で実行すべきです(通常、穴あけのとき)。傾斜面を定義するフレームが有効なのに、割り出し工具ホルダ（カップリングギヤシステム）で工具の向きの自由な設定ができないため工具を正確に直角に設定出来ない場合にも、傾斜面加工を実行する際に、このことが問題になります。これらの場合、平面に垂直な実際に要求された動作とは違って、工具方向にドリルで穴をあける必要があります。そうしないと、ドリルがその縦軸の方向に誘導されません。

ステップ移動

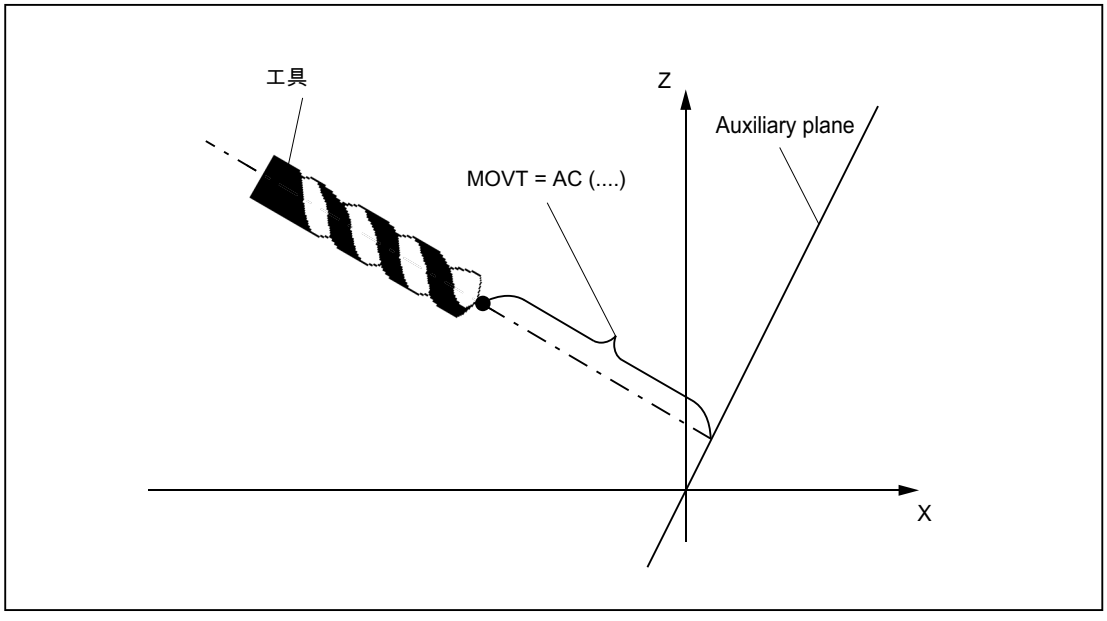
工具方向のステップ移動動作の終点は、MOVT = <値>または MOVT=IC (<値>) を使用してプログラムします。

正の移動方向は、工具刃先から工具アダプタに向かって定義されます。近軸加工(たとえば G91 Z ...など)に対応します。

アブソリュート移動

工具方向のアブソリュートステップ移動動作の終点は MOVT=AC (<値>) を使用してプログラムします。

この場合、現在の原点を貫く平面が定義され、その表面法線ベクトルは、工具の向きに平行になります。その後、MOVT で、この平面に対する位置が指定されます(図を参照してください)。基準面は、目標位置の計算にだけ使用されます。有効なフレームは、この内部計算の影響を受けません。



MOVT によるプログラミングは、向きを指定できる工具ホルダの有無とは関係ありません。動作の方向は、有効な平面によって決まります。
それは垂直軸の方向になります。つまり G17 では Z 方向、G18 では Y 方向、G19 では X 方向です。これは、旋回工具ホルダが有効でない場合や、旋回工具ホルダに旋回工具がないか、あっても初期設定の場合に適用されます。

方向座標変換(3、4、5 軸座標変換)が有効になっている場合、MOVT は同じように動作します。

MOVT が指定されたブロックで、工具の向きが同時に変更される(たとえば、回転軸の同時補間による有効な 5 軸座標変換の)場合、MOVT の動作方向では、ブロックの先頭の向きが重要になります。
有効な 5 軸変換を使用すると、工具中心点(TCP)の軌跡が、向きの変更の影響を受けません。つまり、この軌跡は直線のまま変更されず、その方向はブロックの先頭の工具の向きにより特定されます。

MOVT をプログラム指令する場合、直線補間またはスプライン補間を有効にしてください(G0、G1、ASPLINE、BSPLINE、CSPLINE)。そうしないと、アラームが発生します。スプライン補間が有効になっている場合、結果として生じる軌跡は、通常直線ではありません。これは、MOVT により計算される終点が、X、Y、Z により明確にプログラム指令されているかのように処理されるからです。

MOVT によりブロック内でジオメトリ軸をプログラムすることはできません。

立体角付きフレーム回転の定義

1 つのフレームが、複数の軸を中心にした回転を記述するように定義されている場合、これは、個々の回転の結合により実施されます。以降の回転は、新たな回転座標系で実現されます。

これはいずれも、1 ブロック内のプログラミングに対して、複数の連続するブロックでフレームを設定する場合に適用されます。

- 1 ブロック:`N10 ROT X...Y...Z...`
- 複数の連続するブロック:

```
N10 ROT  Y...
N20 AROT X...
N30 AROT Z...
```

立体角

多くの場合、傾斜面を定義するためにワーク外形図で立体角を指定します。立体角は、ワーク座標系(図を参照)の主平面(X-Y、Y-Z、Z-X 平面)と、傾斜面の交差線の角度です。

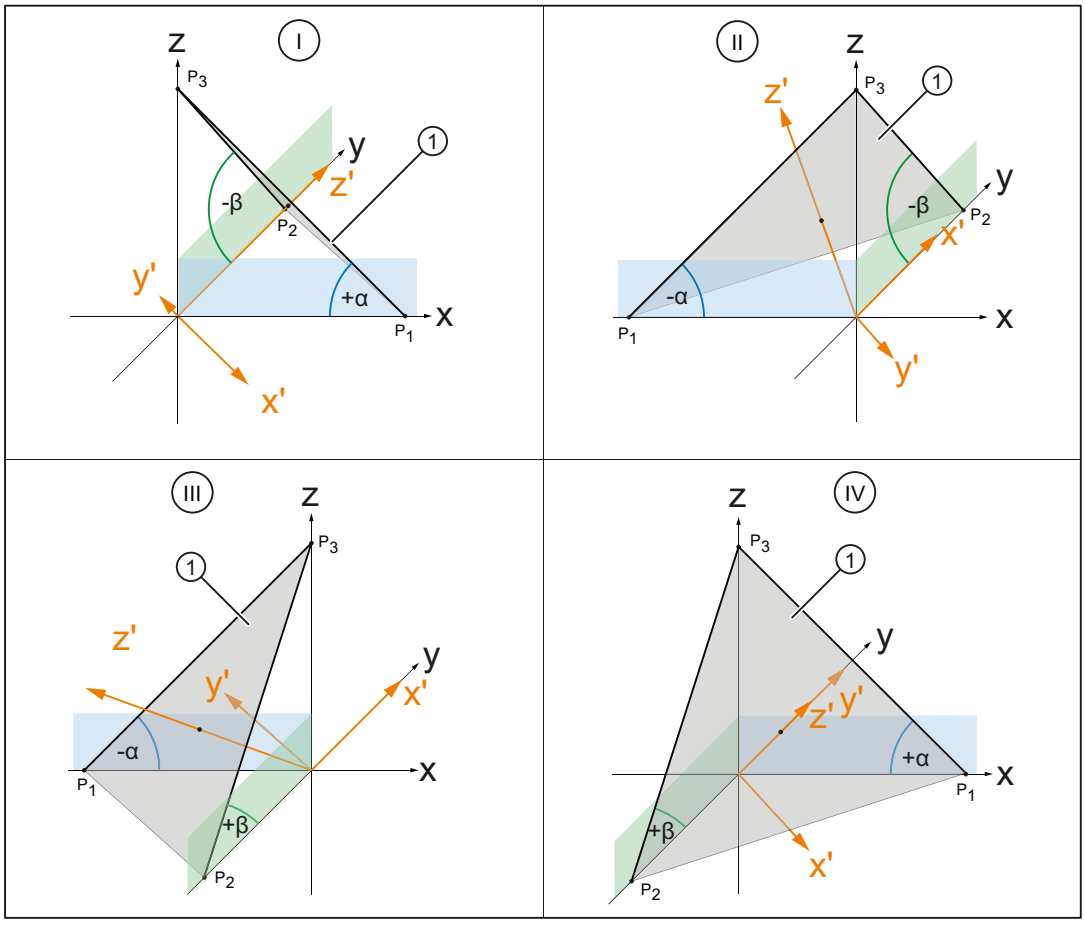
2 つの立体角を指定すれば、空間内の平面の向きは明確に定義されます。3 番目の立体角は、最初の 2 つの立体角から導出されます。

回転は ROTS、AROTS、および CROTS 命令を使用して、立体角として直接定義できます。

単一の立体角を指定できます。この場合、ROTS または AROTS により実行される回転は、ROT と AROT により実行される回転と同じです。

命令でプログラムする 2 軸により平面を定義します。プログラムされていない軸では、座標系の関連する 3 番目の軸を定義します。これは、2 つのプログラムされた軸について、第 1 軸と第 2 軸が一意に定義されることも意味します。この定義は、G17/G18/G19 の平面の定義に対応します。

その平面の一つの軸の軸アドレスによりプログラム指令された角度は、平面の別の軸を交線に移動するために回転するときに中心となる軸を定義します。その交線は、回転された平面が他の軸と 3 番目の軸によって囲まれた平面とともに作ります。この定義により、2 つのプログラム指令角度の一方が原点に向かう場合、定義した平面が、1 軸しかプログラム指令されない場合(たとえば、ROT または AROT を使用した場合)に作成される平面に、確実に入るようになります。



I, ..., IV　四分円 1 ～ 4

①　新しい G17 平面を指定する傾斜面

α、β　傾斜面の立体角

図 11-27　立体角を中心にした回転

図では、例として、四分円 I ～ IV の平面の立体角が示されています。この傾斜面は、ワーク座標系 WCS を回転した後、G17 平面の配置を定義します。立体角の符号は、座標系が関連する軸を中心として回転する方向を示します。

1. y を中心にした回転:
   y 軸を中心とした符号付き角度 α によるワーク座標系 WCS の回転 ⇒
   x'軸は、xz 平面と傾斜面の交差線に対して配置された平行線(共線)です
2. x'を中心にした回転:
   x'軸を中心とした符号付き角度 β による新しいワーク座標系 WCS の回転⇒
   - y'軸は、zy 平面と傾斜面の交差線に対して配置された平行線(共線)です
   - z'軸は、傾斜面に直交しています
   - G17'は、傾斜面と平行です

各四分円について、傾斜面に対してワーク座標系 WCS の G17 平面を配置するプログラミング:

- 四分円 I:ROTS X<+α> Y<-β>
- 四分円 II:ROTS X<-α> Y<-β>
- 四分円 III:ROTS X<-α> Y<+β>
- 四分円 IV:ROTS X<+α> Y<+β>

旋回

立体角の指定では、平面内の 2 次元座標系の向き(つまり、面法線ベクトルを中心にした回転角度)は定義されません。このため、回転した 1 番目の軸が、回転していない座標系の 1 番目と 3 番目の軸により形成される平面内に配置されるように、座標系の位置が特定されます。

つまり、

- X と Y のプログラム指令時、新しい X 軸は、オリジナルの Z-X 平面内に配置されます。
- Z と X のプログラム指令時、新しい Z 軸は、古い Y-Z 平面内に配置されます。
- Y と Z のプログラム指令時、新しい Y 軸は、オリジナルの X-Y 平面内に配置されます。

必要な座標系が、この基本設定に対応していない場合、AROT...を使用して回転を追加でおこなってください。

パラメータ設定:zy'x"形式(RPY 角)または zx'z"形式

プログラム指令されている立体角は、次のマシンデータに対応して、zy'x"形式(RPY 角)または zx'z"形式に基づいて、相当するオイラー角に変換されます。

MD10600 $MN_FRAME_ANGLE_INPUT_MODE

工具方向へのフレーム回転

旧バージョンのソフトウェアにも装備されていた言語命令 TOFRAME を使用すれば、Z 軸が工具方向を向いたフレームを定義できます。
その後、すでにプログラム指令されているフレームが、純粋な回転を記述するフレームにより上書きされます。直前に有効だったフレーム内に存在していたゼロオフセット、ミラーリングまたはスケーリングはどれも削除されます。
この応答は干渉することがあります。通常、ワーク内のレファレンス点を定義するゼロオフセットを保持することが特に役立ちます。

このため、言語命令 TOROT も使用されます。この命令は、プログラム指令したフレーム内の回転成分だけを上書きし、残りの成分を変更しません。TOROT で定義される回転は、TOFRAME で定義される回転と同じです。
TOROT は、TOFRAME と同じように、旋回工具ホルダの有無には関係ありません。この言語命令も、5 軸座標変換に特に役立ちます。

新しい言語命令 TOROT を使用すると、キネマティックタイプ毎に、有効な旋回工具ホルダとの整合性のあるプログラミングを確実に実行できます。

TOFRAME または TOROT は、Z 方向が工具方向に向いたフレームを定義します。この定義は、フライス加工に適切です。フライス加工では通常 G17 が有効です。ただし、特に旋削加工の場合、またはもっと一般的に言えば、G18 または G19 が有効な場合、X 軸または Y 軸を向くフレームを定義できることが望まれます。このために、G グループ 53 で以下の命令を使用できます。

- TOFRAMEX、TOFRAMEY、TOFRAMEZ
- TOROTX、TOROTY、TOROTZ

これらの命令により適切なフレームを定義できます。TOFRAME と TOFRAMEZ の機能、または TOROT と TOROTZ の機能は、互いに同じです。

TOROT または TOFRAME の結果のフレームは、個別のシステムフレーム($P_TOOLFR)に書き込むことができます。そうすると、プログラマブルフレームは変更されません。

- 必要条件:MD28082 $MC_MM_SYSTEM_FRAME_MASK、ビット 3 = 1

TOROT または TOFRAME などをプログラム指令する場合、システムフレームの使用に関係なく、応答は同じです。プログラマブルフレームが他のどこかで追加処理されると、違いが発生します。

注記

新しいシステムでは、G グループ 53 の命令より生成されるフレームには、予定したシステムフレームのみを使用することを推奨します。

例

TRANS は、TOROT の後にプログラム指令されます。パラメータが指定されていない TRANS を使用すると、プログラマブルフレームが削除されます。システムフレームを使用しない方法では、この結果、TOROT により生成されたプログラマブルフレームのフレーム成分も削除されます。ただし、TOROT 成分がシステムフレーム内にある場合、この成分は保持されます。

TOROT または TOFRAME などは、言語命令 TOROTOF により無効になります。TOROTOF を使用すると、システムフレーム$P_TOOLFR 全体が削除されます。命令 TOFRAME な

どで、ステムフレームではなく、プログラマブルフレーム(旧バージョン)が記述されている場合、TOROT は、回転成分だけを削除し、残りのフレーム成分は変更されません。

言語命令 TOFRAME または TOROT が起動される前にすでに回転フレームが有効になっている場合、通常、新たに定義したフレームが旧フレームからできるだけ逸脱しないことが要求されます。たとえば、カップリングギヤの回転軸が原因で工具の向きを自由に設定できないためにフレーム定義を若干変更する必要がある場合が、これに該当します。この言語命令は、新しいフレームの Z 方向を一義的に定義します。

パラメータ設定:TOFRAME、TOROT、PAROT のフレーム定義 (SD42980)

次のセッティングデータで、フレーム定義の実加工平面(G17: X 軸および Y 軸)のジオメトリ軸の方向は、TOFRAME、TOROT(TOROTY、TOROTX)、または PAROT を使用して定義します。

フレームの計算では、フレームの工具方向と垂直軸(G17:Z 軸)が平行で、加工平面に対して直交するように工具方向が定義されます。

最初、工具軸を中心とする回転は任意です。セッティングデータを使用して、新たに定義するフレームの逸脱が以前に有効なフレームに対して可能なかぎり小さくなるように、この自由回転を定義できます。

セッティングデータがゼロ以外である場合常に、新旧のフレームの工具方向が相互に一致すれば、有効なフレームは変わらないままです。

SD42980 $SC_TOFRAME_MODE

TCARR(工具ホルダの要求)と PAROT(ワーク上にワーク座標系を配置)

TCARR は、次のマシンデータで指定される基本フレームを使用します。
MD20184 $MC_TOCARR_BASE_FRAME_NUMBER

基本フレームをすでにすべて使用しているシステムとの競合を防止するために、TCARR と PAROT だけのシステムフレームを作成できます。

PAROT、TOROT および TOFRAME は、以前にプログラマブルフレームの回転成分を変更していました。この場合、PAROT または TOROT を個別に停止することはできません。RESET を実行すると、プログラマブルフレームが削除されます。つまり、運転モードを JOG に切り替えると、PAROT と TOROT の回転成分が使用できなくなります。ユーザーは、プログラマブルフレームに無制限にアクセスできる必要もあります。PAROT と TOROT で生成されたフレームは、アーカイブの実行とデータバックアップを使用した再ロードができる必要があります。

TCARR と PAROT のシステムフレームは、次のマシンデータで設定します。
MD28082 $MC_MM_SYSTEM_FRAME_MASK、ビット 2 = 1

これにより、次のマシンデータは使用されなくなります。
MD20184 $MC_TOCARR_BASE_FRAME_NUMBER

TCARR のシステムフレームが設定されると、TCARR と PAROT は、対応するシステムフレームを記述します。そうでない場合は、マシンデータ MD20184 で指定された基本フレームが記述されます。

タイプ P と M のキネマティックシステムを使用すると、TCARR は、旋回工具ホルダのテーブルオフセット(テーブルの回転の結果生じたゼロオフセット)を平行移動としてシステムフレームに入力します。PAROT は、ワーク基準の WCS が生じるようにシステムフレームを変換します。

システムフレームは保持されるように格納されるため、リセット後も保持されます。モードが変更された場合も、システムフレームは有効なままです。

表示のため、PAROT、TOROT、および TOFRAME の各命令は、別々の G グループに割り当てられます。

PAROTOF

PAROTOF は、PAROT の解除命令です。この命令は、PAROT 用のシステムフレーム内の回転を削除します。削除中、現在の$P_PARTFRAME とデータ管理フレーム$P_PARTFR 内の各回転は削除されます。その後、座標系内の位置が、TCARR に応じて再度作成されます。PAROTOF は、PAROT と同じ G グループ内にあるため、その G 命令表示で表示されます。

TOROT(フレームを回転して WCS の Z 軸を工具の向きと平行に配置)と TOFRAME (同左)

TOROT と TOFRAME のシステムフレームは、次のマシンデータで起動されます。
MD28082 $MC_MM_SYSTEM_FRAME_MASK、ビット 3 = 1

このシステムフレームは、フレーム結合内のプログラマブルフレームの前に配置されます。SZS 座標系は、プログラマブルフレームの前に配置されます。

TOROTOF

TOROTOF は、TOROT と TOFRAME の解除命令です。この命令は対応するシステムフレームを削除します。現在の$P_ TOOLFRAME とデータ管理フレーム$P_TOOLFR も削除されます。TOROTOF は、TOROT と TOFRAME と同じ G グループ内にあるため、その G 命令表示で表示されます。

例

解決済みキネマティックによる旋回工具ホルダの使用

```
N10 $TC_DP1[1,1]=120
N20 $TC_DP3[1,1]= 13                    ; 工具長 13 mm
; 工具ホルダ 1 の定義:
N30 $TC_CARR1[1] = 0                    ; 1 番目のオフセットベクトルの X 成分
N40 $TC_CARR2[1] = 0                    ; 1 番目のオフセットベクトルの Y 成分
N50 $TC_CARR3[1] = 0                    ; 1 番目のオフセットベクトルの Z 成分
N60 $TC_CARR4[1] = 0                    ; 2 番目のオフセットベクトルの X 成分
N70 $TC_CARR5[1] = 0                    ; 2 番目のオフセットベクトルの Y 成分
N80 $TC_CARR6[1] = -15                  ; 2 番目のオフセットベクトルの Z 成分
N90 $TC_CARR7[1] = 1                    ; 1 番目の軸の X 成分
N100 $TC_CARR8[1] = 0                   ; 1 番目の軸の Y 成分
N110 $TC_CARR9[1] = 0                   ; 1 番目の軸の Z 成分
N120 $TC_CARR10[1] = 0                  ; 2 番目の軸の X 成分
N130 $TC_CARR11[1] = 1                  ; 2 番目の軸の Y 成分
N140 $TC_CARR12[1] = 0                  ; 2 番目の軸の Z 成分
N150 $TC_CARR13[1] = 30                 ; 1 番目の軸の回転角度
N160 $TC_CARR14[1] = -30                ; 2 番目の軸の回転角度
N170 $TC_CARR15[1] = 0                  ; 3 番目のオフセットベクトルの X 成分
N180 $TC_CARR16[1] = 0                  ; 3 番目のオフセットベクトルの Y 成分
N190 $TC_CARR17[1] = 0                  ; 3 番目のオフセットベクトルの Z 成分
N200 $TC_CARR18[1] = 0                  ; 4 番目のオフセットベクトルの X 成分
N210 $TC_CARR19[1] = 0                  ; 4 番目のオフセットベクトルの Y 成分
N220 $TC_CARR20[1] = 15                 ; 4 番目のオフセットベクトルの Z 成分
N230 $TC_CARR21[1] = A                  ; 1 番目の軸の基準
N240 $TC_CARR22[1] = B                  ; 2 番目の軸の基準
N250 $TC_CARR23[1] = "M"                ; 工具ホルダのタイプ
N260 X0 Y0 Z0 A0 B45 F2000
N270 TCARR=1 X0 Y10 Z0 T1 TCOABS        ; 旋回工具ホルダの選択
N280 PAROT                              ; テーブルの回転
N290 TOROT                              ; 工具方向への z 軸の回転
N290 X0 Y0 Z0
N300 G18 MOVT=AC(20)                    ; G18 平面内での処理
N310 G17 X10 Y0 Z0                      ; G17 平面内での処理
N320 MOVT=-10
N330 PAROTOF                            ; テーブルの回転の解除
N340 TOROTOF                            ; WCS が工具と揃えられなくなります
N400 M30
```

11.5.9 SAVE 属性を持つサブプログラム(SAVE)

各種フレームに対して、サブプログラムに関する動作を SAVE 属性を使用して設定することができます。

設定可能フレーム G54 ～ G599

設定可能フレームの動作は、MD10617 $MN_FRAME_SAVE_MASK.**ビット 0** を使用して設定できます。

- BIT0 = 0
  サブプログラムを使用して、有効な設定可能フレームの値だけがシステム変数 $P_IFRAME によって変更され、G 命令は保持される場合、その変更内容はサブプログラムが終了しても有効なままです。
- BIT0 = 1
  サブプログラムが終了すると、サブプログラムの呼び出し前に有効だった設定可能フレーム、G 命令、および値が再び有効になります。

基本フレーム$P_CHBFR[ ]および$P_NCBFR[ ]

基本フレームの動作は、MD10617 $MN_FRAME_SAVE_MASK.**BIT1** を使用して設定できます。

- BIT1 = 0
  有効な基本フレームがサブプログラムによって変更される場合、その変更内容は、サブプログラムが終了しても有効なままです。
- BIT1 = 1
  サブプログラムが終了すると、サブプログラムの呼び出し前に有効であった基本フレームが再び有効になります。

プログラマブルフレーム

サブプログラムが終了すると、サブプログラムの呼び出し前に有効であったプログラマブルフレームが再び有効になります。

システムフレーム

システムフレームがサブプログラムによって変更される場合、その変更内容はサブプログラムが終了しても有効なままです。

11.5.10 データバックアップ

データブロック

フレームのデータバックアップは、以下のデータブロックで実行されます。

- チャネル別フレーム⇒ _N_CHAN<x>_UFR データブロック
- グローバルフレーム⇒ _N_NC_UFR データブロック

注記

- 以下のマシンデータは、保存済みシステムフレームのバックアップと復元の間では変更しないことを強く推奨します。変更すると、保存済みシステムフレームをロードできなくなる可能性があります。
  MD28082 $MC_MM_SYSTEM_FRAME_MASK (チャネルの計算に含まれるチャネル別システムフレーム設定)
- データバックアップは常に、マシンデータで最初に設定した軸座標ではなく、チャネルで現在有効なジオメトリ軸の割り当てに従って行われます。

システムフレームのデータバックアップ

データ管理で作成されたシステムフレームのみが、システムフレームのデータバックアップでは考慮されます。これらのシステムフレームは、制御装置のコミッショニング時に、以下のマシンデータで選択されています。

MD28083 $MC_MM_SYSTEM_DATAFRAME_MASK

データ管理で作成されていないフレームはバックアップされません。

NC グローバルフレームのデータバックアップ

NC グローバルフレームのデータバックアップは、以下のマシンデータ項目のいずれかで、1 つ以上の NC グローバルフレームがパラメータ設定されている場合にのみ実行されます。

- MD18601 $MN_MM_NUM_GLOBAL_USER_FRAMES
- MD18602 $MN_MM_NUM_GLOBAL_BASE_FRAMES
- MD18603 $MN_MM_NUM_GLOBAL_G_FRAMES

11.5.11 座標系の位置

座標系の現在の位置指令値は、次のシステム変数を介して読み取ることができます。フィードバック値は、PLC を介して WCS、SZS、BZS または MCS で表示できます。MCS/WCS には、フィードバック値表示用のソフトキーがあります。工作機械メーカは、機械のワーク座標系に対応する座標系を PLC で指定できます。HMI は、NC の対応する現在値を要求します。

$AA_IM[軸]

$AA_IM[軸]変数を使用すると、機械座標系の指令値を軸ごとに読み取ることができます。

$AA_IEN[軸]

変数$AA_IEN[軸]を使用すると、設定可能ゼロオフセットシステム(SZS)の指令値を軸ごとに読み取ることができます。

$AA_IBN[軸]

$AA_IBN[軸]変数を使用すると、基本ゼロオフセットシステム(BZS)の指令値を読み取ることができます。

$AA_IW[軸]

$AA_IW[軸]変数を使用すると、ワーク座標系(WCS)の指令値を読み取ることができます。

11.5.12 コントロールシステムの応答

11.5.12.1 POWER ON

フレーム	POWER ON 後のフレーム状態
$P_PFRAME プログラマブルフレーム	削除されます。
$P_IFRAME 設定可能フレーム	以下の設定に応じて保持されます。 • MD24080 $MC_USER_FRAME_POWERON_MASK、ビット 0 • MD20152 $MC_GCODE_RESET_MODE[7]
$P_GFRAME 研削フレーム	以下の設定に応じて保持されます。 • MD24080 $MC_USER_FRAME_POWERON_MASK、ビット 0 • MD20152 $MC_GCODE_RESET_MODE[63]

フレーム	POWER ON 後のフレーム状態
$P_ACTBFRAME 基本フレーム全体	以下の設定に応じて保持されます。 • MD20110 $MC_RESET_MODE_MASK、ビット 0 およびビット 14 基本フレームは、以下のマシンデータで削除できます。 • MD10615 $MN_NCBFRAME_POWERON_MASK • MD24004 $MC_CHBFRAME_POWERON_MASK
システムフレーム: $P_PARTFRAME、 $P_SETFRAME、 $P_ISO1FRAME、 $P_ISO2FRAME、 $P_ISO3FRAME、 $P_TOOLFRAME、 $P_WPFRAME、 $P_TRAFRAME、 $P_ISO4FRAME、 $P_RELFRAME、 $P_CACFRAME	保持されます。 個々のシステムフレームは、以下のマシンデータで削除できます。 • MD24008 $MC_CHSFRAME_POWERON_MASK システムフレームの削除は、データ管理において最優先で実行されます。
$P_EXTFRAME 外部ゼロオフセット	保持されますが、再び起動する必要があります。
DRF オフセット	削除されます。

11.5.12.2 モード変更

システムフレーム

システムフレームは保持され、有効なままです。

JOG モード

JOG モードでは、回転が有効である場合、現在のフレームのフレームコンポーネントがジオメトリ軸について考慮されるだけです。他の軸別フレームは考慮されません。

PLC 軸およびコマンド軸

次のマシンデータを使用して、PLC 軸およびコマンド軸に対する応答を設定できます。

MD32074 $MA_FRAME_OR_CORRPOS_NOTALLOWED (フレームまたは工具長補正は不許可)

11.5.12.3 チャネルリセット/パートプログラム終了

基本フレームのリセット応答

基本フレームのリセット応答は、次のマシンデータで設定します。

MD20110 $MC_RESET_MODE_MASK (チャネルリセット/パートプログラム終了後の初期制御設定の定義)

システムフレームのリセット応答

チャネルリセット/パートプログラム終了の後、システムフレームはデータ管理で保持されます。

次のマシンデータを使用して、個々のシステムフレームの起動を設定できます。

MD24006 $MC_CHSFRAME_RESET_MASK、ビット<n> = <値> (チャネルリセット/パートプログラム終了後の有効なシステムフレーム)

ビット	値	意味
0	1	チャネルリセット/パートプログラム終了の後に、現在位置と接触計測用のシステムフレームが有効です。
1	1	チャネルリセット/パートプログラム終了の後に、外部ゼロオフセット用のシステムフレームが有効です。
2	---	使用されません。
3	---	使用されません。
4	1	チャネルリセット/パートプログラム終了の後に、ワークレファレンス点用のシステムフレームが有効です。
5	1	チャネル/パートプログラム終了の後に、サイクル用のシステムフレームが有効です。
6	---	予約済。RESET 応答は MD20110 $MC_RESET_MODE_MASK によって決まります。
7	1	チャネルリセット/パートプログラム終了の後に、システムフレーム $P_ISO1FR が有効です。
8	1	チャネルリセット/パートプログラム終了の後に、システムフレーム $P_ISO2FR が有効です。
9	1	チャネルリセット/パートプログラム終了の後に、システムフレーム $P_ISO3FR が有効です。

ビット	値	意味
10	1	チャネルリセット/パートプログラム終了の後に、システムフレーム $P_ISO4FR が有効です。
11	1	チャネルリセット/パートプログラム終了の後に、システムフレーム $P_RELFR が有効です。
ビット<n> = 0 の場合は、関連するシステムフレームが有効では**ありません**。		

TCARR、PAROT、TOROT、および TOFRAME 用システムフレームのリセット応答

チャネルリセット/パートプログラム終了の後のシステムフレーム TCARR、PAROT、TOROT、および TOFRAME のリセット応答は、G 命令のリセット設定によって決まります。

この設定は次のマシンデータで行います。

- MD20110 $MC_RESET_MODE_MASK、ビット<n> = <値>

ビット	値	意味	
0	0	TCARR および PAROT 用の現在のシステムフレームが保持されます。	
0	1	**その他の関連するマシンデータ設定**	**効果**
		MD20152 $MC_GCODE_RESET_MODE[51] = **0** および MD20150 $MC_GCODE_RESET_VALUES[51] = **1**	PAROTOF
		MD20152 $MC_GCODE_RESET_MODE[51] = **0** および MD20150 $MC_GCODE_RESET_VALUES[51] = **2**	PAROT
		MD20152 $MC_GCODE_RESET_MODE[51] = **1**	TCARR と PAROT は保持されます。
		MD20152 $MC_GCODE_RESET_MODE[52] = **0** および MD20150 $MC_GCODE_RESET_VALUES[52] = **1**	TOROTOF
		MD20152 $MC_GCODE_RESET_MODE[52] = **0** および MD20150 $MC_GCODE_RESET_VALUES[52] = **2**	TOROT
		MD20152 $MC_GCODE_RESET_MODE[52] = **0** および MD20150 $MC_GCODE_RESET_VALUES[52] = **3**	TOFRAME
		MD20152 $MC_GCODE_RESET_MODE[52] = **1**	TOROT および TOFRAME は保持されます。
bit 0 == 1 のその他のビットの説明			

ビット	値	意味
14	0	結合された基本フレーム一式が削除されます。
	1	基本フレーム一式は、以下の追加マシンデータで得られます。
		MD24002 $MC_CHBFRAME_RESET_MASK、ビット<n> = 1 N:n 番目のチャネル別基本フレームが計算されて、結合された基本フレーム一式に含まれます。
		MD10613 $MC_CHBFRAME_RESET_MASK、ビット<n> = 1 N:n 番目の NCU グローバル基本フレームが計算されて、結合された基本フレーム一式に含まれます。

TCARR および PAROT は、同じフレームを表す 2 つの異なる機能です。PAROTOF を使用すると、TCARR の成分がチャネルリセット/パートプログラム終了時に起動しません。

- MD20152 $MC_GCODE_RESET_MODE[ ] (G グループの RESET 応答)
- MD20150 $MC_GCODE_RESET_VALUES (G グループのリセット設定)

チャネルリセット/パートプログラム終了の後のフレーム状態

フレーム	チャネルリセット/パートプログラム終了の後の状態
$P_PFRAME プログラマブルフレーム	削除されます。
$P_IFRAME 設定可能フレーム	以下の設定に応じて保持されます。 • MD20110 $MC_RESET_MODE_MASK • MD20152 $MC_GCODE_RESET_MODE[7]
$P_GFRAME 研削フレーム	以下の設定に応じて保持されます。 • MD20110 $MC_RESET_MODE_MASK • MD20152 $MC_GCODE_RESET_MODE[63]
$P_ACTBFRAME 基本フレーム一式	以下の設定に応じて保持されます。 • MD20110 $MC_RESET_MODE_MASK、ビット 0 およびビット 14 • MD10613 $MN_NCBFRAME_RESET_MASK • MD24002 $MC_CHBFRAME_RESET_MASK

フレーム	チャネルリセット/パートプログラム終了の後の状態
システムフレーム: $P_PARTFRAME、 $P_SETFRAME、 $P_ISO1FRAME、 $P_ISO2FRAME、 $P_ISO3FRAME、 $P_TOOLFRAME、 $P_WPFRAME、 $P_TRAFRAME、 $P_ISO4FRAME、 $P_RELFRAME、 $P_CACFRAME	以下の設定に応じて保持されます。 • MD24006 $MC_CHSFRAME_RESET_MASK • MD20150 $MC_GCODE_RESET_VALUES[<n>]
$P_EXTFRAME 外部ゼロオフセット	保持されます。
DRF オフセット	保持されます。

システムフレームの削除

以下のマシンデータを使用して、チャネルリセット/パートプログラム終了時にデータ管理のシステムフレームを削除できます。

MD24007 $MC_CHSFRAME_RESET_CLEAR_MASK、ビット<n> = <値>

ビット	値	意味
0	1	チャネルリセット/パートプログラム終了のために、現在位置と接触計測用のシステムフレームが削除されます。
1	1	チャネルリセット/パートプログラム終了のために、外部ゼロオフセット用のシステムフレームが削除されます。
2	---	予約済み(TCARR および PAROT については、MD20150 $MC_GCODE_RESET_VALUES[ ]を参照)
3	---	予約済み(TOROT および TOFRAME については、MD20150 $MC_GCODE_RESET_VALUES[ ]を参照)
4	1	チャネルリセット/パートプログラム終了のために、ワークレファレンス点用のシステムフレームが削除されます。
5	1	チャネル/パートプログラム終了のために、サイクル用のシステムフレームが削除されます。

ビット	値	意味
6	---	予約済。RESET 応答は MD20110 $MC_RESET_MODE_MASK によって決まります。
7	1	チャネルリセット/パートプログラム終了のために、システムフレーム$P_ISO1FR が削除されます。
8	1	チャネルリセット/パートプログラム終了のために、システムフレーム$P_ISO2FR が削除されます。
9	1	チャネルリセット/パートプログラム終了のために、システムフレーム$P_ISO3FR が削除されます。
10	1	チャネルリセット/パートプログラム終了のために、システムフレーム$P_ISO4FR が削除されます。
11	1	チャネルリセット/パートプログラム終了のために、システムフレーム$P_RELFR が削除されます。
ビット<n> = 0 の場合は、関連するシステムフレームが削除されません。		

11.5.12.4 パートプログラム起動

パートプログラム起動後のフレーム状態

フレーム	パートプログラム起動後の状態
$P_PFRAME プログラマブルフレーム	削除されます。
$P_IFRAME 設定可能フレーム	以下の設定に応じて保持されます。 MD20112 $MC_START_MODE_MASK
$P_GFRAME 研削フレーム	以下の設定に応じて保持されます。 MD20112 $MC_START_MODE_MASK
$P_ACTBFRAME 基本フレーム一式	保持されます。

フレーム	パートプログラム起動後の状態
システムフレーム: $P_PARTFRAME、 $P_SETFRAME、 $P_ISO1FRAME、 $P_ISO2FRAME、 $P_ISO3FRAME、 $P_TOOLFRAME、 $P_WPFRAME、 $P_TRAFRAME、 $P_ISO4FRAME、 $P_RELFRAME、 $P_CACFRAME	保持されます。
$P_EXTFRAME 外部ゼロオフセット	保持されます。
DRF オフセット	保持されます。

11.5.12.5 ブロックサーチ

計算ありのブロック検索

計算をともなうブロックサーチの実行中にデータ管理フレームも変更されます。

ブロックサーチのキャンセル

ブロック検索がチャネルリセットにより中止された場合は、以下のマシンデータを使用して、すべてのデータ管理フレームがブロック検索前の値にリセットされるよう設定できます。

MD28560 $MC_MM_SEARCH_RUN_RESTORE_MODE、ビット<n>

ビット	値	意味
0	1	データ管理のすべてのフレームが復元されます。

多段ブロックサーチの場合、フレームは、直前のブロックサーチの状態に設定されます。

SERUPRO

「SERUPRO」機能はサポートされません。

11.5.12.6 REPOS

フレームに対する特殊な処理はありません。 フレームが ASUB で変更される場合、フレームはプログラム内で保持されます。 変更が ASUB で有効であった場合、REPOS による再位置決め時に、変更されたフレームが含まれます。

11.6 ワーク基準のフィードバック回路

11.6.1 概要

定義

「ワーク関連のフィードバック値管理」という用語は、以下のことを可能にする一連の機能を指します。

- 電源投入後に、マシンデータで定義されたワーク座標系の使用。
  特徴:
  - オペレータの操作が不要です。
  - JOG モードおよび AUTOMATIC モードで有効です。
- プログラム終了後も、次のパートプログラムに備えて以下の設定を有効なままにする。
  - 有効な平面
  - 設定可能フレーム(G54-G57)
  - キネマティックトランスフォーメーション
  - 有効な工具オフセット
- ユーザーインタフェースを介してワーク座標系と機械座標系を切り替える。
- オペレータ操作によりワーク座標系を変更する(たとえば、設定可能フレームまたは工具オフセットの変更)

11.6.2 ワーク基準のフィードバック回路の使用

必要条件、初期設定

前の章で説明した設定は、システムに合わせた設定となっています。
HMI ソフトウェアの電源投入後の事前定義は、MCS です。

WCS への切り替え

ユーザーインタフェースを介して WCS に切り替えると、軸位置が WCS の原点を基準として表示されます。

MCS への切り替え

ユーザーインタフェースを介して MCS に切り替えると、軸位置が MCS の原点を基準として表示されます。

座標系の間の相互関係

下記の図では、機械座標系(MCS)とワーク座標系(WCS)の相互関係を示しています。

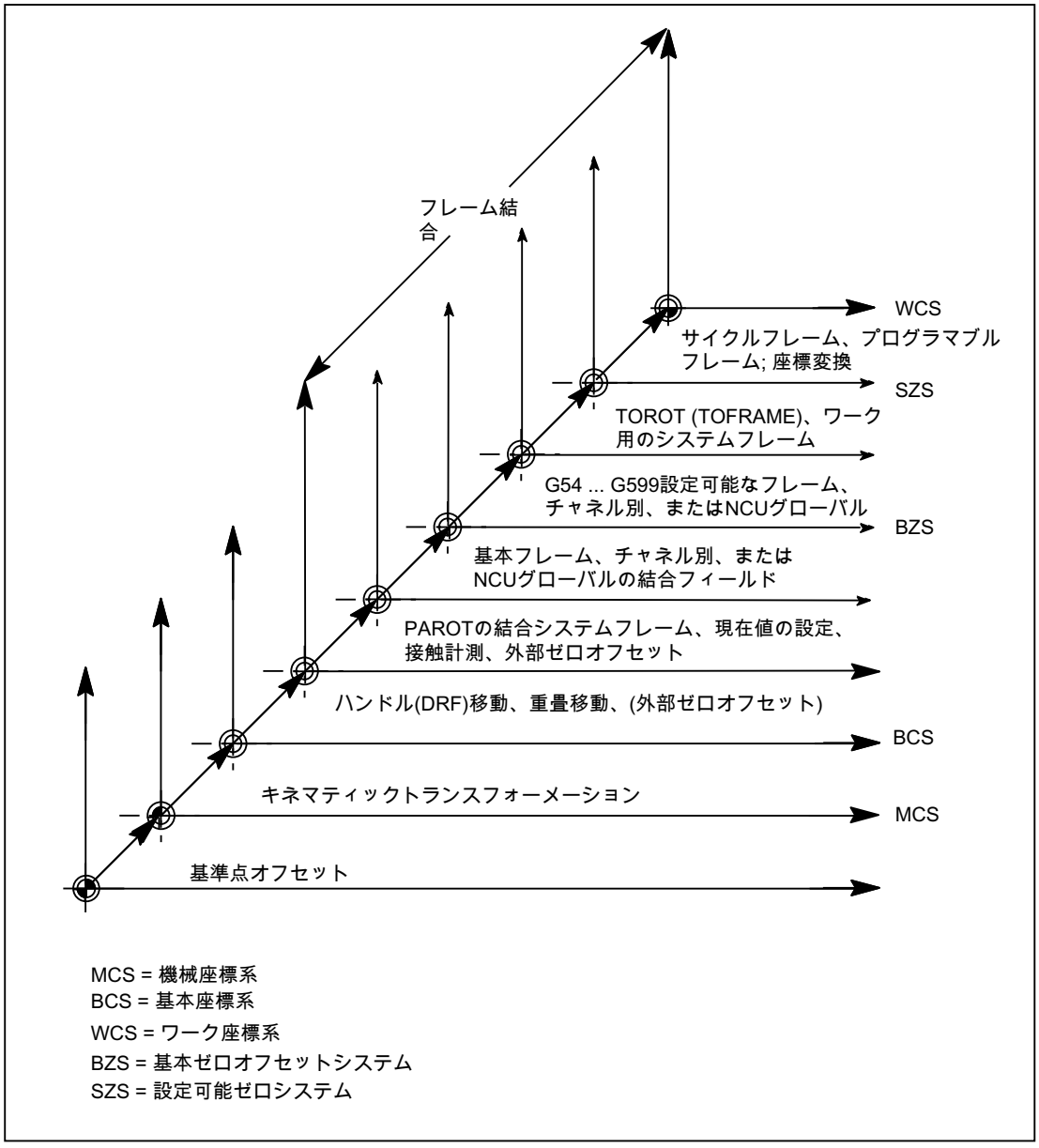


図 11-28　座標系の間の相互関係

詳細については、「H2: PLC への補助機能出力 (ページ 475)」と「W1:工具補正 (ページ 1767)」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』
- 『機能マニュアル、上級機能』; 「キネマティック座標変換(M1)」

- 『機能マニュアル、応用機能』; 「軸の連結と ESR (M3)」
  : 「連結動作」の章、:「軸間連動機能」の章
- 『機能マニュアル、応用機能』;「法線方向制御(T3)」

11.6.3 特殊な応答

オーバーストア

RESET 状態におけるオーバーストアでは

- フレーム(ゼロオフセット)
- 有効な平面
- 有効な座標変換
- 工具オフセット

は、すぐにチャネル内のすべての軸のフィードバック値表示に作用します。

操作パネルを介した入力

操作パネルの操作によって
[有効なフレーム](ゼロオフセット、[パラメータ]操作エリア)
および
[有効な工具長補正] ([パラメータ]操作エリア)
の値を変更すると、次のいずれかの動作によって、表示上でこれらの変更が反映されます。

- RESET キーを押す。
- 再選択:
  - パートプログラムによるゼロオフセット
  - パートプログラムによる工具オフセット
- リセット:
  - オーバーストアによるゼロオフセット
  - オーバーストアによる工具オフセット
- パートプログラム起動

MD9440

操作パネル用の HMI マシンデータ
MD9440 ACTIVATE_SEL_USER_DATA
が設定されている場合、入力した値が RESET 状態で直ちに有効になります。

パートプログラム実行停止状態で入力された値は、プログラム実行が継続されると有効になります。

フィードバック値の読み取り

フレーム(ゼロオフセット)または工具オフセットの起動後に$AA_IW のフィードバック値が WCS で読み取られる場合、たとえ軸がまだ、起動された変更により移動されていなくても、起動された変更は既にその読み取り結果に含まれます。

変数$AA_IEN[軸]を使用すると、設定可能な原点系(SZS)のフィードバック値を、それぞれの軸についてパートプログラムから読み取ることができます。

変数$AA_IBN[軸]を使用すると、 基本ゼロオフセットシステム(BZS)のフィードバック値を、それぞれの軸についてパートプログラムから読み取ることができます。

フィードバック値の表示

プログラム指令輪郭は、常に WCS で表示されます。

次のオフセットが MCS に追加されます。

- キネマティック座標変換
- DRF オフセット/外部ゼロオフセット
- 有効なフレーム
- 現在の工具の有効な工具オフセット

PLC による切り替え

フィードバック値は、PLC を介して WCS、SZS、BZS または MCS で表示できます。
PLC では、どの座標系が機械のワーク座標系に対応するかを定義できます。

MMC 電源投入時には MCS となるように、あらかじめ設定されています。
信号 DB19 DBB0.7「MCS/WCS 切り替え」を使用すると、PLC から WCS に切り替えることもできます。

PLC への伝送

マシンデータ
MD20110 / MD20112 ビット 1
に応じて、工具長補正選択時に、補助機能(D、T、M)が PLC に出力されます（または出力されません）。

注記

PLC によって WCS が選択されていても、該当するモードにおいてオペレータが WCS と MCS を切り替えることができます。
しかし、モードまたは操作エリア、あるいはその両方の変更時に、PLC によって選択された WCS が評価され、起動されます(「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 (ページ 569)」の章を参照してください)。

11.7 制限事項

注意すべき必要条件はありません。

11.8 例

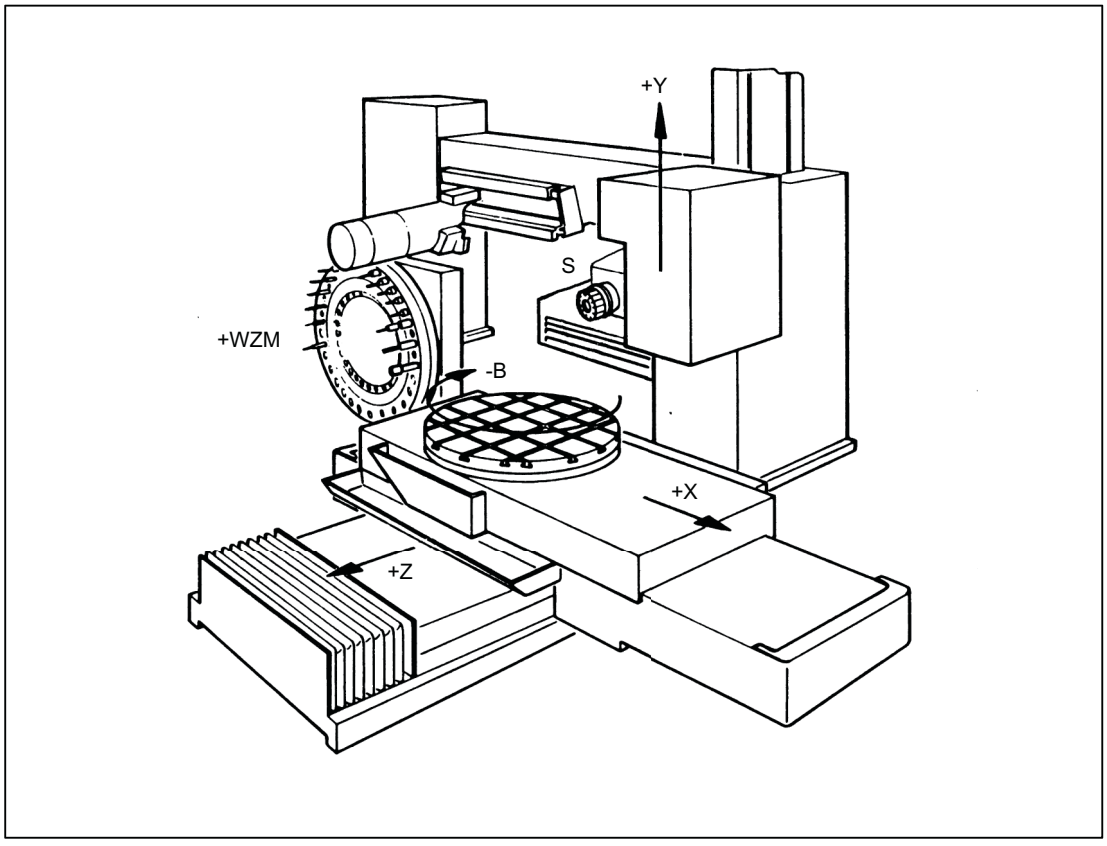
11.8.1 軸

回転テーブルを備えた 3 軸フライス盤の軸設定

1. 機械軸:X1	直線軸
2. 機械軸:Y1	直線軸
3. 機械軸:Z1	直線軸
4. 機械軸:B1	回転テーブル(多面加工のための回転用)
5. 機械軸:W1	工具マガジン(工具リボルバ)用の回転軸
6. 機械軸:C1	(主軸)

1. ジオメトリ軸:X	(1. 番目のチャネル)
2. ジオメトリ軸:Y	(1. 番目のチャネル)

3. ジオメトリ軸:Z (1. 番目のチャネル)
1. 付加軸:B (1. 番目のチャネル)
2. 付加軸:WZM (1. 番目のチャネル)
1. 主軸: S1/C (1. 番目のチャネル)



マシンデータのパラメータ設定

マシンデータ	値
MD10000 AXCONF_MACHAX_NAME_TAB[0]	= X1
MD10000 AXCONF_MACHAX_NAME_TAB[1]	= Y1
MD10000 AXCONF_MACHAX_NAME_TAB[2]	= Z1
MD10000 AXCONF_MACHAX_NAME_TAB[3]	= B1

マシンデータ	値
MD10000 AXCONF_MACHAX_NAME_TAB[4]	= W1
MD10000 AXCONF_MACHAX_NAME_TAB[5]	= C1
MD20050 AXCONF_GEOAX_ASSIGN_TAB[0]	= 1
MD20050 AXCONF_GEOAX_ASSIGN_TAB[1]	= 2
MD20050 AXCONF_GEOAX_ASSIGN_TAB[2]	= 3
MD20060 AXCONF_GEOAX_NAME_TAB[0]	=X
MD20060 AXCONF_GEOAX_NAME_TAB[1]	=Y
MD20060 AXCONF_GEOAX_NAME_TAB[2]	=Z
MD20070 AXCONF_MACHAX_USED[0]	= 1
MD20070 AXCONF_MACHAX_USED[1]	= 2
MD20070 AXCONF_MACHAX_USED[2]	= 3
MD20070 AXCONF_MACHAX_USED[3]	= 4
MD20070 AXCONF_MACHAX_USED[4]	= 5
MD20070 AXCONF_MACHAX_USED[5]	= 6
MD20080 AXCONF_CHANAX_NAME_TAB[0]	=X
MD20080 AXCONF_CHANAX_NAME_TAB[1]	=Y
MD20080 AXCONF_CHANAX_NAME_TAB[2]	=Z

マシンデータ	値
MD20080 AXCONF_CHANAX_NAME_TAB[3]	= B
MD20080 AXCONF_CHANAX_NAME_TAB[4]	= WZM
MD20080 AXCONF_CHANAX_NAME_TAB[5]	= S1
MD30300 IS_ROT_AX[3]	= 1
MD30300 IS_ROT_AX[4]	= 1
MD30300 IS_ROT_AX[5]	= 1
MD30310 ROT_IS_MODULO[3]	= 1
MD30310 ROT_IS_MODULO[4]	= 1
MD30310 ROT_IS_MODULO[5]	= 1
MD30320 DISPLAY_IS_MODULO[3]	= 1
MD30320 DISPLAY_IS_MODULO[4]	= 1
MD20090 SPIND_DEF_MASTER_SPIND	= 1
MD35000 SPIND_ASSIGN_TO_MACHAX[AX1]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX2]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX3]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX4]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX5]	= 0
MD35000 SPIND_ASSIGN_TO_MACHAX[AX6]	= 1

11.8.2 座標系

グローバル基本フレームの設定

2 つのチャネルを有する 1 つの NC が必要です。これは以下のように適用されます。

- この場合、両チャネルがグローバル基本フレームを書き込むことができます。
- グローバル基本フレームが再度起動すると、他のチャネルで変更が認識されます。
- 両チャネルがグローバル基本フレームを読み取ることができます。
- 両チャネルがグローバル基本フレームを個別に起動できます。

マシンデータ

マシンデータ	値
MD10000 $MN_AXCONF_MACHAX_NAME_TAB[0] MD10000 $MN_AXCONF_MACHAX_NAME_TAB[1] MD10000 $MN_AXCONF_MACHAX_NAME_TAB[2] MD10000 $MN_AXCONF_MACHAX_NAME_TAB[3] MD10000 $MN_AXCONF_MACHAX_NAME_TAB[4] MD10000 $MN_AXCONF_MACHAX_NAME_TAB[5]	= X1 = X2 = X3 = X4 = X5 = X6
MD18602 $MN_MM_NUM_GLOBAL_BASE_FRAMES	= 1
MD28081 $MC_MM_NUM_BASE_FRAMES	= 1

チャネル 1 のマシンデータ	値		チャネル 1 のマシンデータ	値
$MC_AXCONF_CHANAX_NAME_TAB[0] $MC_AXCONF_CHANAX_NAME_TAB[1] $MC_AXCONF_CHANAX_NAME_TAB[2]	= X = Y = Z		$MC_AXCONF_CHANAX_NAME_TAB[0] $MC_AXCONF_CHANAX_NAME_TAB[1] $MC_AXCONF_CHANAX_NAME_TAB[2]	= X = Y = Z
$MC_AXCONF_MACHAX_USED[0] $MC_AXCONF_MACHAX_USED[1] $MC_AXCONF_MACHAX_USED[2]	= 1 = 2 = 3		$MC_AXCONF_MACHAX_USED[0] $MC_AXCONF_MACHAX_USED[1] $MC_AXCONF_MACHAX_USED[2]	= 4 = 5 = 6
$MC_AXCONF_GEOAX_NAME_TAB[0] $MC_AXCONF_GEOAX_NAME_TAB[1] $MC_AXCONF_GEOAX_NAME_TAB[2]	= X = Y = Z		$MC_AXCONF_GEOAX_NAME_TAB[0] $MC_AXCONF_GEOAX_NAME_TAB[1] $MC_AXCONF_GEOAX_NAME_TAB[2]	= X = Y = Z
$MC_AXCONF_GEOAX_ASSIGN_TAB[0] $MC_AXCONF_GEOAX_ASSIGN_TAB[1] $MC_AXCONF_GEOAX_ASSIGN_TAB[2]	= 1 = 2 = 3		$MC_AXCONF_GEOAX_ASSIGN_TAB[0] $MC_AXCONF_GEOAX_ASSIGN_TAB[1] $MC_AXCONF_GEOAX_ASSIGN_TAB[2]	= 4 = 5 = 6

1 番目のチャネルのパートプログラム

コード (抜粋)	; コメント
. . .	
N100 $P_NCBFR[0] = CTRANS(x, 10)	; NC グローバル基本フレームの起動
. . .	
N130 $P_NCBFRAME[0] = CROT(X, 45)	; 回転 => の NC グローバル基本フレームの起動、 で回転が許可されないために 18310 アラーム、NC グローバルフレーム
. . .	

2 番目のチャネルのパートプログラム

コード (抜粋)	; コメント
. . .	
N100 $P_NCBFR[0] = CTRANS(x, 10)	; また、NC グローバル基本フレームは 2 番目のチャネルでも有効です
. . .	
N510 G500 X10	; 基本フレームを起動します
N520 $P_CHBFRAME[0] = CTRANS(x, 10)	; 2 番目のチャネルの現在のフレームは、オフセットにより起動します
. . .	

11.8.3 フレーム

例 1

ジオメトリ軸交換によってチャネル軸をジオメトリ軸に切り替える必要があります。

この交換の意図は、X 軸の 10 の平行移動コンポーネントをプログラマブルフレームに提供することです。

現在の設定可能フレームを保持する必要があります。

FRAME_GEOAX_CHANGE_MODE = 1

プログラムコード	コメント
$P_UIFR[1] = CROT(x,10,y,20,z,30)	; ジオメトリ軸交換後にフレームは保持されます。
G54	; 設定可能フレームが有効になります。
TRANS A10	; A の軸別オフセットも交換されます。
GEOAX(1, a)	; A が X 軸になります。
	; $P_ACTFRAME = CROT(X,10,Y,20,Z,30) :CTRANS(X10)

座標変換の変更と同時に、複数のチャネル軸がジオメトリ軸になる可能性があります。

例 2

チャネル軸 4、5 および 6 が 5 軸方向座標変換のジオメトリ軸になります。このため、座標変換前にジオメトリ軸がすべて交換されます。

座標変換が起動されると、現在のフレームが変更されます。

ジオメトリ軸になるチャネル軸の軸別フレームコンポーネントが新規 WCS の計算に使用されます。座標変換前にプログラム指令された回転は保持されます。座標変換が解除されると、旧 WCS が復元されます。

座標変換の前後でジオメトリ軸が変更されず、フレームが座標変換前のまま変更されないのが、おそらく、最も一般的な用途です。

マシンデータ:

$MN_FRAME_GEOAX_CHANGE_MODE = 1

$MC_AXCONF_CHANAX_NAME_TAB [0] = "CAX"
$MC_AXCONF_CHANAX_NAME_TAB [1] = "CAY"
$MC_AXCONF_CHANAX_NAME_TAB [2] = "CAZ"
$MC_AXCONF_CHANAX_NAME_TAB [3] = "A"
$MC_AXCONF_CHANAX_NAME_TAB [4] = "B"
$MC_AXCONF_CHANAX_NAME_TAB [5] = "C"

$MC_AXCONF_GEOAX_ASSIGN_TAB [0] = 1
$MC_AXCONF_GEOAX_ASSIGN_TAB [1] = 2
$MC_AXCONF_GEOAX_ASSIGN_TAB [2] = 3

$MC_AXCONF_GEOAX_NAME_TAB [0] = "X"
$MC_AXCONF_GEOAX_NAME_TAB [1] = "Y"
$MC_AXCONF_GEOAX_NAME_TAB [2] = "Z"

$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [0] = 4
$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [1] = 5
$MC_TRAFO_GEOAX_ASSIGN_TAB_1 [2] = 6

$MC_TRAFO_AXES_IN_1 [0] = 4
$MC_TRAFO_AXES_IN_1 [1] = 5
$MC_TRAFO_AXES_IN_1 [2] = 6
$MC_TRAFO_AXES_IN_1 [3] = 1
$MC_TRAFO_AXES_IN_1 [4] = 2

プログラム:

プログラムコード	コメント
`$P_NCBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)`	

プログラムコード	コメント
$P_CHBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)	
$P_IFRAME = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6) :CROT(Z,45)	
$P_PFRAME = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6) :CROT(x,10,y,20,z,30)	
TRAORI	; 座標変換セット GEOAX(4,5,6)
	; $P_NCBFRAME[0] = ctrans(x,4,y,5,z,6,cax,1,cay,2,caz,3)
	; $P_ACTBFRAME = CTRANS(X,8,Y,10,Z,12,CAX,2,CAY,4,CAZ,6)
	; $P_PFRAME = CTRANS(X,4,Y,5,Z,6,CAX,1,CAY,2,CAZ,3) :CROT(x,10,y,20,z,30)
	; $P_IFRAME = CTRANS(X,4,Y,5,Z,6,CAX,1,CAY,2,CAZ,3) :CROT(Z,45)
TRAFOOF	; 座標変換の解除、GEOAX (1,2,3)を設定
	; $P_NCBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
	; $P_CHBFRAME[0] = ctrans(x,1,y,2,z,3,a,4,b,5,c,6)
	; $P_IFRAME = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6) :CROT(Z,45)
	; $P_PFRAME = CTRANS(X,1,Y,2,Z,3,A,4,B,5,C,6) :CROT(x,10,y,20,z,30)

11.9 データリスト

11.9.1 マシンデータ

11.9.1.1 表示マシンデータ

番号	識別子: $MM_	説明
SINUMERIK Operate		
9242	MA_STAT_DISPLAY_BASE	移動するジョイント STAT を表示するための数値単位
9243	MA_TU_DISPLAY_BASE	回転軸位置 TU を表示するための数値単位
9244	MA_ORIAXES_EULER_ANGLE_NAME	オイラー角としての旋回軸の表示
9245	MA_PRESET_FRAMEIDX	接触計測および PRESET の値保存
9247	USER_CLASS_BASE_ZERO_OFF_PA	[パラメータ]操作エリアでの基本オフセットの適用
9248	USER_CLASS_BASE_ZERO_OFF_MA	[運転]操作エリアでの基本オフセットの適用
9424	MA_COORDINATE_SYSTEM	フィードバック値表示用の座標系

番号	識別子: $MM_	説明
9440	ACTIVE_SEL_USER_DATA	編集後、有効なデータ(フレーム)が直ちに使用可能になります。
9449	WRITE_TOA_LIMIT_MASK	刃先データおよび位置依存オフセットに対する MD9203 の適用可否
9450	MM_WRITE_TOA_FINE_LIMIT	摩耗(仕上げ)の制限値
9451	MM_WRITE_ZOA_FINE_LIMIT	オフセット(仕上げ)の制限値

11.9.1.2 NC 別マシンデータ

番号	識別子: $MN_	説明
10000	AXCONF_MACHAX_NAME_TAB	機械軸名称
10600	FRAME_ANGLE_INPUT_MODE	フレーム内の回転の順序
10602	FRAME_GEOAX_CHANGE_MODE	フレームおよびジオメトリ軸の切り替え
10610	MIRROR_REF_AX	ミラーリング用の基準軸
10612	MIRROR_TOGGLE	ミラーの切り替え
10613	NCBFRAME_RESET_MASK	リセット後の有効な NCU-グローバル基本フレーム
10615	NCBFRAME_POWERON_MASK	電源投入後のグローバル基本フレームのリセット
10617	FRAME_SAVE_MASK	SAVE サブプログラムに対応したフレームの動作
10650	IPO_PARAM_NAME_TAB	補間パラメータの名前
10660	INTERMEDIATE_POINT_NAME_TAB	G2/G3 の中間点座標の名前
11640	ENABLE_CHAN_AX_GAP	チャネル軸ギャップが可能
18600	MM_FRAME_FINE_TRANS	FRAME の精密オフセット(SRAM)
18601	MM_NUM_GLOBAL_USER_FRAMES	グローバルにあらかじめ定義されたユーザーフレーム数(SRAM)
18602	MM_NUM_GLOBAL_BASE_FRAMES	グローバル基本フレーム数(SRAM)

11.9.1.3 チャネルマシンデータ

番号	識別子: $MC_	説明
20050	AXCONF_GEOAX_ASSIGN_TAB	チャネル軸へのジオメトリ軸の割り当て
20060	AXCONF_GEOAX_NAME_TAB	チャネルのジオメトリ軸名称
20070	AXCONF_MACHAX_USED	チャネルで有効な機械軸番号
20080	AXCONF_CHANAX_NAME_TAB	チャネル内の軸名称
20110	RESET_MODE_MASK	リセット/パートプログラム終了後の基本制御設定の定義
20118	GEOAX_CHANGE_RESET	ジオメトリ軸の自動変更が可能
20126	TOOL_CARRIER_RESET_VALUE	リセット時に有効な工具ホルダ
20140	TRAFO_RESET_VALUE	電源投入(リセット/パートプログラム終了)時の座標変換セット
20150	GCODE_RESET_VALUES	G グループの初期設定
20152	GCODE_RESET_MODE	G グループのリセット応答
20184	TOCARR_BASE_FRAME_NUMBER	テーブルオフセットを取得するための基本フレームの番号
21015	INVOLUTE_RADIUS_DELTA	エボルベント(インボリュート曲線)の終点監視
22532	GEOAX_CHANGE_M_CODE	ジオメトリ軸切り替え用 M コード
22534	TRAFO_CHANGE_M_CODE	座標変換切り替え用 M コード
24000	FRAME_ADD_COMPONENTS	G58 と G59 用フレーム成分
24002	CHBFRAME_RESET_MASK	チャネル別の基本フレームの RESET 応答
24004	CHBFRAME_POWERON_MASK	電源投入後のチャネル別基本フレームのリセット
24006	CHSFRAME_RESET_MASK	リセット後に有効なシステムフレーム
24007	CHSFRAME_RESET_CLEAR_MASK	RESET 時のシステムフレームのクリア
24008	CHSFRAME_POWERON_MASK	電源投入後のシステムフレームのリセット
24010	PFRAME_RESET_MODE	プログラマブルフレームの RESET モード
24020	FRAME_SUPPRESS_MODE	フレームマスク用の設定
24030	FRAME_ACT_SET	SZS 座標系設定
24040	FRAME_ADAPT_MODE	有効なフレームの補正
24050	FRAME_SAA_MODE	データ管理フレームの保存と起動

番号	識別子: $MC_	説明
24805	TRACYL_ROT_AX_FRAME_1	回転軸オフセット TRACYL 1
24855	TRACYL_ROT_AX_FRAME_2	回転軸オフセット TRACYL 2
24905	TRANSMIT_ROT_AX_FRAME_1	回転軸オフセット TRANSMIT1
24955	TRANSMIT_ROT_AX_FRAME_2	回転軸オフセット TRANSMIT2
28080	MM_NUM_USER_FRAMES	設定可能フレームの数(SRAM)
28081	MM_NUM_BASE_FRAMES	基本フレームの数
28082	MM_SYSTEM_FRAME_FRAMES	システムフレーム(SRAM)
28560	MM_SEARCH_RUN_RESTORE_MODE	シミュレーション後のデータ復元

11.9.1.4 軸/主軸マシンデータ

番号	識別子: $MA_	説明
32074	FRAME_OR_CORRPOS_NOTALLOWED	FRAME または工具長オフセットが不許可
35000	SPIND_ASSIGN_TO_MACHAX	機械軸への主軸の割り当て

11.9.2 セッティングデータ

11.9.2.1 チャネルセッティングデータ

番号	識別子: $SC_	説明
42440	FRAME_OFFSET_INCR_PROG	フレーム内のワークオフセット
42980	TOFRAME_MODE	TOFRAME、TOROT、PAROT のフレーム定義

11.9.3 システム変数

識別子	説明
$AA_ETRANS[軸]	外部ゼロオフセット
$AA_IBN[軸]	基本原点座標系(BZS)のフィードバック値
$AA_IEN[軸]	設定可能な原点座標系(SZS)のフィードバック値
$AA_OFF[軸]	プログラム指令軸の重畳動作
$AC_DRF[軸]	軸のハンドルオーバライド
$AC_JOG_COORD	手動移動の座標系
$P_ACSFRAME	BCS と SZS 間の有効なフレーム
$P_ACTBFRAME	有効なすべての基本フレーム
$P_ACTFRAME	有効なすべてのフレーム
$P_BFRAME	最初の有効な基本フレーム。$P_CHBFR[0]に対応します。
$P_CHBFR[<n>]	データ管理フレーム:基本フレームは、G500、G54...G599 で起動できます。
$P_CHBFRAME[<n>]	有効な基本フレーム
$P_CHBFRMASK	チャネル内の有効な基本フレーム
$P_CHSFRMASK	システムフレームマスク
$P_CYCFR	データ管理フレーム:サイクル用のシステムフレーム
$P_CYCFRAME	サイクル用の有効なシステムフレーム
$P_EXTFR	データ管理フレーム:外部ゼロオフセットのシステムフレーム
$P_EXTFRAME	外部ワークオフセット用の有効なシステムフレーム
$P_IFRAME	有効な設定可能フレーム
$P_ISO1FR	データ管理フレーム:ISO G51.1 ミラーリング用のシステムフレーム
$P_ISO2FR	データ管理フレーム:ISO G68 2 次元回転用システムフレーム
$P_ISO3FR	データ管理フレーム:ISO G68 3 次元回転用システムフレーム
$P_ISO4FR	データ管理フレーム:ISO G51 スケール用のシステムフレーム
$P_ISO1FRAME	ISO G51.1 ミラーリング用の有効なシステムフレーム
$P_ISO2FRAME	ISO G68 2DROT 用の有効なシステムフレーム
$P_ISO3FRAME	ISO G68 3DROT 用の有効なシステムフレーム
$P_ISO4FRAME	ISO G51 スケール用の有効なシステムフレーム

識別子	説明
$P_NCBFR[n]	データ管理フレーム:データ管理の NCU グローバル基本フレームは、G500、G54...G599 で起動できます。
$P_NCBFRAME[n]	有効な NCU グローバル基本フレーム
$P_NCBFRMASK	グローバル基本フレームのマスク
$P_PARTFR	データ管理フレーム:TCARR と PAROT のシステムフレームのビットマスクです。
$P_PARTFRAME	旋回工具ホルダ付きの、TCARR および PAROT 用の有効なシステムフレーム
$P_PFRAME	有効なプログラマブルフレーム
$P_SETFR	データ管理フレーム:現在値設定用のシステムフレーム
$P_SETFRAME	フィードバック値の設定用の有効なシステムフレーム
$P_TOOLFR	データ管理フレーム:TOROT および TOFRAME 用のシステムフレーム
$P_TOOLFRAME	TOROT と TOFRAME 用の有効なシステムフレーム
$P_TRAFRAME	データ管理フレーム:座標変換用のシステムフレームのビットマスクです。
$P_TRAFRAME	座標変換用の有効なシステムフレーム
$P_UBFR	G500、G54...G599 の後で有効になる、チャネル内のデータ管理の 1 番目の基本フレーム$P_CHBFR[0]に対応しています。
$P_UIFR[n]	データ管理フレーム:設定可能フレームは、G500、G54...G599 で起動できます。
$P_UIFRNUM	有効な設定可能フレームの数
$P_WPFR	データ管理フレーム:ワーク用のシステムフレーム
$P_WPFRAME	サイクル用の有効なシステムフレーム

11.9.4 信号

11.9.4.1 チャネルからの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
T 機能変更	DB21、... .DBX61.0-.2	-
D 機能変更	DB21、... .DBX62.0-.2	-
T 機能 1	DB21、... .DBB118-119	DB250x.DBD2000
D 機能 2	DB21、... .DBB129	DB250x.DBD5000
有効な G 機能グループ 1 の番号 8(ビット整数)	DB21、... .DBB208	DB350x.DBB0
有効な G 機能グループ 2 の番号 8(ビット整数)	DB21、... .DBB209	DB350x.DBB1
...	...	…
有効な G 機能グループ 29 の番号 8(ビット整数)	DB21、... .DBB236	DB350x.DBB28

11.9.4.2 軸/主軸への信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
外部ゼロオフセットの受け入れ	DB31、... .DBX3.0	-

11.9.4.3 軸/主軸からの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
回転軸/主軸	DB31、... .DBX60.0	DB390x.DBX0.0

N2: 非常停止

12

12.1 概略説明

機能

このコントロールシステムでは、以下の機能を基に、工作機械メーカの非常停止機能実装をサポートします。

- 非常停止ボタンは、すべての SINUMERIK 機械操作パネル上で、オペレータがアクセスしやすい場所に設置されています。 非常停止ボタンの機能には、電気開閉接点の強制開放と機械式自己保持/ロック機構が含まれています。
- 非常停止要求は、PLC 上の NC/PLC インタフェースを介して NC に送信されます。
- 非常停止機能は、停止カテゴリ 0 または 1 (EN 60204)に従って機械を停止状態にしなければなりません。
- 非常停止では、PLC によって制御されるすべての運転機能を、工作機械メーカが設定できる、安全な状態に移行させることができます。
- 非常停止ボタンを解除しても、非常停止状態はキャンセルされず、機械が再起動されることもありません。
- 非常停止状態がキャンセルされた後、機械軸をレファレンス点復帰させたり主軸を原点同期させる必要はありません。 非常停止シーケンス中、機械軸のフィードバック位置が継続して監視されます。

12.2 関連規格

関連規格

非常停止機能は、以下の規格に準拠しなければなりません。

- EN ISO 12000-1
- EN ISO 12000-2
- EN 418
- EN 60204

非常停止

EN 418 によれば、非常停止に要求される機能は次のとおりです。

- 作業員に対して発生または存在する危険や、機械または加工材料に対する損害を未然に防いだり低減させること。
- 通常の停止機能では間に合わない場合に、作業員の 1 回の動作で起動されること。

危険

EN 418 によれば、危険は次の原因で発生します。

- 機能不良(機械の故障、材料の特性が加工に適していないこと、人為的過失など)
- 通常操作

EN ISO 12000-2 規格

非常停止に関する EC 機械指令の基本安全要求事項に従って、機械に非常停止装置を装備しなければなりません。

例外

次の場合は、非常停止装置を装備しなくてもかまいません。

- 停止時間を短縮できないため、または非常停止方法が危険を抑制するのに適していないため、非常停止装置によって危険を低減できない。
- 手持ち式の機械や手動で操作する機械である。

通知
工作機械メーカは、国内規格と国際規格に従うことが義務付けられています。 SINUMERIK コントローラは、以下に記す機能によって、工作機械メーカの非常停止機能実装をサポートします。 ただし、非常停止機能(非常停止のトリガ、シーケンスおよび応答)に関する責任はすべて工作機械メーカにあります。

12.3 非常停止制御部品

非常停止制御部品

非常停止制御部品は、 EN 418 に従って、 機械的に自動ラッチし、非常事態が発生したときにオペレータまたはその他の作業員が簡単に作動させられるように設計しなければなりません。

制御部品として、次のタイプがあります。

- マッシュルーム型押しボタンスイッチ
- ワイヤ/ケーブル、コード、ロッド
- 引っ張りグリップ
- 特殊な場合: 保護カバーのない足踏式スイッチ

非常停止ボタンと制御

非常停止ボタンが作動したこと、または非常停止ボタンから直接発信される信号を、PLC 入力としてコントローラ(PLC)に送らなければなりません。 PLC ユーザープログラムでは、この PLC 入力を次のインタフェース信号で NC に転送しなければなりません。

DB10 DBX56.1 (非常停止)

非常停止ボタンがリセットされたこと、または非常停止ボタンから直接発信される信号を、PLC 入力としてコントローラ(PLC)に送らなければなりません。 PLC ユーザープログラムでは、この PLC 入力を次のインタフェース信号で NC に転送しなければなりません。

DB10 DBX56.2 (非常停止応答)

接続条件

非常停止ボタンの接続に関する詳細については、次を参照してください。
参照先:
『オペレータコンポーネントマニュアル』

12.4 非常停止処理

非常停止制御部品が作動すると、非常停止装置は、危険を回避または低減するために可能な限り最善の方法で取り扱わなければなりません。

「可能な限り最善の方法」というのは、リスクアセスメントに基づいて、最適な遅延率と正しい停止カテゴリ(EN 60204 で定義)を指定できることを意味します。

NC における非常停止処理

EN 418 であらかじめ定義された、非常停止状態を取得するために実装される内部機能の処理は、コントロールシステムでは、次のとおりです。

1. パートプログラムの実行が中断されます。
   対応する軸別のパラメータ設定された時間で、すべての機械軸が減速されます。
   MD36610 $MA_AX_EMERGENCY_STOP_TIME (異常発生時の減速時間)
   これに関して実施可能な最大減速時間は、各ドライブの最大制動電流によって決まります。 最大制動電流は、指令値=0 (急制動)を設定することによって実施します。
2. 次のインタフェース信号がリセットされます。
   DB11 DBX6.3 (モードグループ準備完了)
3. 次のインタフェース信号が設定されます。
   DB10 DBX106.1 (非常停止有効)
4. アラーム 3000「非常停止」が表示されます。
5. パラメータ設定された遅延時間が経過すると、機械軸のサーボイネーブルがリセットされます。
   この遅延時間の設定は、次のマシンデータでプログラム指令されます。
   MD36620 $MA_SERVO_DISABLE_DELAY_TIME (コントローライネーブル遮断遅延)
   次の設定規則を順守しなければなりません。 MD36620 ≥ MD36610
6. すべての機械軸が、コントローラ内でフォローアップモードに切り替わります。
   この間、機械軸の位置制御状態ではありません。

機械における非常停止処理

機械における非常停止処理は、工作機械メーカが決めます。

NC における処理に関連して、次の点に注意が必要です。

- NC での処理は、次のインタフェース信号を使用して開始されます。
  DB10 DBX56.1 (非常停止)
  機械軸が停止状態になったら、EN 418 に従って動力供給を遮断しなければなりません。

  注記

  動力供給を遮断するのは工作機械メーカの責任です。

- PLC の I/O のデジタル/アナログ出力は、NC における非常停止処理の影響を受けません。
  非常停止時に、個々の出力により特定の状態または電圧レベルを得る必要がある場合、工作機械メーカは PLC ユーザープログラムで、これを実現しなければなりません。

- NCK の I/O システムの高速デジタル出力は、NC における非常停処理の影響を受けません。
  非常停止時に個々の出力が特定の状態になる必要がある場合、工作機械メーカは PLC ユーザープログラムで次のインタフェース信号を介して、この状態を NC に伝送しなければなりません。
  DB10 DBB4-7

  注記

  NC におけるシーケンスが上記のように実行すべきははない場合、 PLC ユーザープログラムで工作機械メーカが定義した非常停止状態に到達するまで、インタフェース信号 DB10 DBX56.1 (非常停止)を設定しないでください。
  このインタフェース信号が設定されておらず、他の未処置のアラームが存在しない限り、すべてのインタフェース信号が NC で有効です。 したがって、工作機械メーカが定義するいかなる非常停止状態(軸/主軸/チャネル別の非常停止状態を含む)をとることも可能です。

12.5 非常停止応答

非常停止制御部品は、EN 418 に従って非常停止制御部品を手動で操作することによってのみリセットできます。

非常停止制御部品をリセットするだけで再始動命令が起動されてはなりません。

作動中の非常停止制御部品がすべて手動で意図的にリセットされるまでは、機械の再始動が不可能でなくてはなりません。

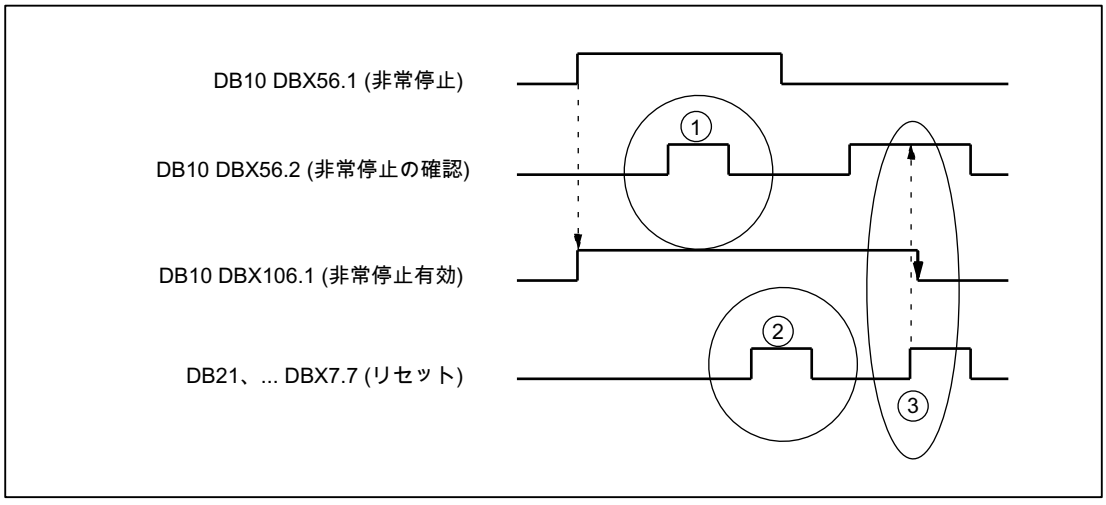
非常停止確認応答

インタフェース信号 DB10 DBX56.2 (非常停止応答)が設定され、次にインタフェース信号 DB11、... DBX0.7 (モードグループリセット)が設定された場合のみ非常停止状態がリセットされます。

したがって、インタフェース信号 DB10 DBX106.1(非常停止有効)がリセットされるまで、インタフェース信号 DB10 DBX56.2 (非常停止応答)とインタフェース信号 DB21、... DBX7.7 (リセット)を両方とも十分長い時間、設定するように注意してください。

注記

インタフェース信号 DB21、... DBX7.7 (リセット)のみ使用して非常停止状態をリセットすることはできません。



(1) DB10 DBX56.2 (非常停止応答)は無効です。

(2) DB21、... DBX7.7 (リセット)は無効です。

(3) DB10 DBX56.2 と DB21、... DBX7.7 によって DB10 DBX106.1(非常停止有効)がリセットされます。

図 12-1　非常停止状態のリセット

働き

非常停止状態をリセットには、次の働きがあります。

- コントローラ内で、すべての軸について
  - サーボイネーブルが設定されます。
  - フォローアップモードがキャンセルされます。
  - 位置制御が起動されます。
- 次のインタフェース信号が設定されます。
  DB31、... DBX60.5 (位置制御有効)
  DB11 DBX6.3 (モードグループ準備完了)
- 次のインタフェース信号がリセットされます。
  DB10 DBX106.1 (非常停止有効)
- アラーム 3000「非常停止」が解除されます。
- NC の全チャネルでパートプログラム処理が中断されます。

PLC および NC I/O

PLC ユーザープログラムで、PLC と NC の I/O を加工動作のための状態に戻さなければなりません。

POWER OFF/ON(電源オフ/オン)

コントローラの電源を一旦切って、再びオンにすること(POWER OFF/ON)でも、非常停止状態をリセットすることが可能です。

必要条件:

コントローラの電源投入時に、インタフェース信号 DB10 DBX56.1 (非常停止)を設定しないでください。

12.6 データリスト

12.6.1 マシンデータ

12.6.1.1 軸/主軸マシンデータ

番号	識別子: $MA_	説明
36610	AX_EMERGENCY_STOP_TIME	エラー状態における減速カーブの長さ
36620	SERVO_DISABLE_DELAY_TIME	サーボイネーブル遮断遅れ

12.6.2 信号

12.6.2.1 NC への信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
非常停止	DB10.DBX56.1	DB2600.DBX0.1
非常停止の応答	DB10.DBX56.2	DB2600.DBX0.2

12.6.2.2 NC からの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
非常停止有効	DB10.DBX106.1	DB2700.DBX0.1

12.6.2.3 モードグループへの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
モードグループリセット	DB11.DBX0.7	DB3000.DBX0.7

P1:径方向軸

13

13.1 機能

径方向軸

「旋削」加工のシステムでは、径方向軸は主軸の中心軸に垂直に移動する機械軸、言い換えれば、縦軸 Z を参照します。

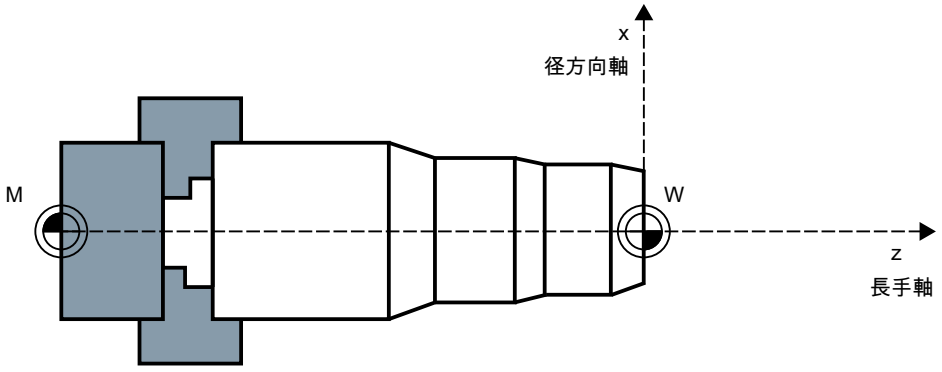


図 13-1　機械座標系での径方向軸の位置

径方向軸としてのジオメトリ軸

チャネルのどのジオメトリ軸でも、径方向軸として定義できます。

径方向軸としてのジオメトリ軸の場合は、以下の機能を、同時または別々に許可し、起動することができます。

- 直径でのプログラミングと表示
- 周速一定制御 G96/G961/G962 の基準軸

チャネルの複数の径方向軸

チャネルの複数の径方向軸の導入では、直径指定と G96/G961/G962 の基準軸の機能の結合の解除がおこなわれます。直径指定と G96/G961/G962 の基準軸は、異なる径方向軸に対して有効にできます(下の表を参照してください)。

	直径でのプログラミングと表示			**G96/G961/G962 の基準軸**	
許容される軸タイプ:	ジオメトリ軸		直線チャネル軸	ジオメトリ軸	
チャネルでの選択:	1 つ	3 のうちの m	n のうちの m	1 つ	3 のうちの 1 つ
特定の効果: **マシンデータ:**	チャネル MD20100	軸 MD30460		チャネル MD20100	
プログラミング	DIAM* チャネル別にモーダル G グループ 29			SCC[<軸>] チャネル別にモーダル G96/G961/G962 の基準軸	
軸入れ替え時の引き継ぎ	DIAM*A[<軸>] 軸別モーダル				
軸別のノンモーダル直径/半径指定:	DAC, DIC; RAC, RIC ブロックの軸別のみのプログラミング				
DIAM*:DIAMOF、DIAMON、DIAM90、DIAMCYCOF DIAM*A[<軸>]:DIAMOFA[<軸>]、DIAMONA[<軸>]、DIAM90A[<軸>]、DIACYCOFA[<軸>]、DIAMCHANA[<軸>] <軸>:ジオメトリ、チャネルの軸名称または機械軸名称					

注記

回転軸は、径方向軸として機能することはできません。

直径対応データ

DIAMON/DIAMONA[<軸>]

DIAMON/DIAMONA[<軸>] (「プログラミング (ページ 1037)」を参照)で直径指定を起動した後は、以下のデータが直径寸法を意味します。

- ワーク座標系での径方向軸の表示データ
  - 指令と実位置
  - 残移動距離
  - REPOS オフセット
- 「JOG」モード:
  - ステップ手動ハンドルによる移動量(INC) (有効な MD に応じて異なります)
- パートプログラムプログラミング
  - 指令モード(G90 / G91)に関係なく、終点
  - 次のパートプログラム命令 AC アブソリュートでプログラミングされる場合の、円弧軌跡プログラム指令(G2/G3)の補間パラメータ。
- ワーク座標系(WCS)を基準にして読み取られるフィードバック値:
  - $AA_MW[<径方向軸>]
    計測機能 MEAS (残移動距離を削除する計測)および MEAW (残移動距離を削除しない計測)のシステム変数
  - $P_EP[<径方向軸>]
  - $AA_IW[<径方向軸>]

DIAM90/DIAM90A[<軸>]

指令モード依存の直径指定を DIAM90/DIAM90A[<軸>]で起動した後は、以下のデータが、操作モード(G90/G91)に関係なく、常に直径に関連して表示されます。

- 現在値
- ワーク座標系(WCS)を基準にして読み取られるフィードバック値:
  - $AA_MW[<径方向軸>]
    計測機能 MEAS (残移動距離を削除する計測)および MEAW (残移動距離を削除しない計測)のシステム変数
  - $P_EP[<径方向軸>]
  - $AA_IW[<径方向軸>]

DIAMCYCOF/DIACYCOFA[<軸>]

DIAMCYCOF/DIACYCOFA[<軸>]によって、半径指定への切り替えがコントローラの内部で実行されます。DIAMCYCOF または DIACYCOFA[AX] の前に有効だった直径指定状態は、HMI に表示され続けます。

常時半径対応のデータ

径方向軸の場合、以下のデータは**必ず**半径に対して入力、プログラミングおよび表示されます。

- パートプログラムプログラミング
  - CIP による円弧軌跡プログラミングの補間パラメータ
- オフセット:
  - 工具オフセット
  - プログラマブルおよび設定可能フレーム
  - 外部ゼロオフセット
  - DRF およびプリセットオフセット
  - その他
- 作業領域リミット
- ソフトウェアリミットスイッチ
- 送り速度
- 機械座標系を基準にした表示データ
- 軸 、FSD および MSD のサービスイメージの表示データ

常に半径対応であるデータの拡張機能:

以下は、PLC 軸、FC18 による、または PLC により排他的に制御される軸に摘要されます。

- 半径での PLC 軸の寸法は、直径機能のある複数の径方向軸にも適用され、チャネル別または軸別の直径指定には関係ありません。
- JOG モード(Inc)では、PLC 軸はチャネル状態に依存します。直径指定が有効で MD20624 $MC_HANDWH_CHAN_STOP_COND ビット 15 = 0 の場合、指定された移動量の半分の軌跡だけを移動します。

13.2 パラメータ設定

径方向軸としてのジオメトリ軸の定義

1 つのジオメトリ軸の、チャネルの径方向軸としての定義は、次のマシンデータを使用しておこなわれます。

MD20100 $MC_DIAMETER_AX_DEF (径方向軸機能によるジオメトリ軸)

例

MD20100 $MC_DIAMETER_AX_DEF="X" ; ジオメトリ軸 X がチャネルの径方向軸です。

この軸に対して、直径指定と G96/G961/G962 による周速一定制御の割り当ての両方が可能です。

チャネルの複数の径方向軸

チャネルの追加の径方向軸の定義は、以下のマシンデータを使用して実行します。このチャネルでは、軸別の直径指定が可能になります。

MD30460 BASE_FUNCTION_MASK.Bit 2 = <Value>

ビット	番目の値	意味
2	0	軸別の直径指定は**おこなえません**。
	1	軸別の直径指定がおこなえます。

注記

設定 MD30460 ビット 2 = 1 は、直線軸に対してのみ有効です。

必要条件

チャネル内で経方向軸は、チャネル別(MD20100)および軸別(MD30460、ビット 2)の軸として同時に定義できます。チャネル別マシンデータには、より高い優先度があります。

MD20100 では、以下のチャネル別機能が起動時に経方向軸に割り当てられます。

以下の軸別の基本設定が NC の電源投入時に経方向軸に割り当てられます:"直径指定チャネル状態の移行"`DIAMCHANA[<Achse>]`

軸別直径指定のリリースにより(MD30460、ビット 2 = 1)、ユーザー側で以下の軸別操作を使用できます:

- DIAMONA (軸別モーダル直径指定:オン)
- DIAMOFA (軸別モーダル直径指定:オフ)
- DIAM90A (G90 と AC では軸別モーダル直径指定、G91 と IC では半径指定)
- DIACYCOFA (軸別モーダル直径指定:サイクル内はオフ)
- DIAMCHANA (直径指定チャネル状態の移行)

工具パラメータの寸法

以下のチャネル別マシンデータで、以下の工具パラメータは、チャネルで定義されたすべての経方向軸に対する直径値として有効にできます:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK.<Bit> = <value>

MD20360 $MC_TOOL_PARAMETER_DEF_MASK		
ビット	値	意味
1	1	直径としての径方向軸の工具長
2	1	直径としての摩耗または工具長および平面変更に対するアラーム
3 [1)]	0	ワークオフセット$P_EXTFRAME とフレーム 径方向軸に対して、フレームのワークオフセットは常に半径値として計算されます。
	1	直径としての経方向軸のフレームのゼロオフセット(WO) フレームには、ワークオフセットが半径値として内部に格納されます。直径指定から半径指定への変更、または逆の変更のときに、変換はおこなわれません。
4	1	直径としてのプリセット値
5 [1)]	0	外部ゼロオフセット(軸重畳) 径方向軸に対して、外部ゼロオフセットは常に半径値として計算されます。
	1	直径としての径方向軸の外部 WO 直径指定から半径指定への変更、または逆の変更のときに、変換はおこなわれません。
6	1	直径としての径方向軸のフィードバック値

MD20360 $MC_TOOL_PARAMETER_DEF_MASK		
ビット	値	意味
7	1	直径値としての径方向軸のフィードバック値の表示
8 1)	1	WCS の残りの軌跡を常に半径として表示
9 1)	ハンドル指定がパス指定(MD11346 $MN_HANDWH_TRUE_DISTANCE == 1)である設定対象のすべての経方向軸について、以下が適用されます:	
	0	チャネル別直径指定または軸別直系指定が当該軸に対して有効の場合、指定されたハンドル移動量の半分の軌跡を移動します。
	1	指定されたハンドル移動量の半分の軌跡を常に移動します。
10	1	有効な工具がない場合の向きを調整できる有効な工具ホルダの工具部分
11	1	直径としての$TC_DP6 の使用
12	1	工具直径の摩耗としての$TC_DP15 の使用
13 1)	1	円弧の周囲をジョグ移動する場合、円弧の中心点座標は常に半径の値です。SD42690 $SC_JOG_CIRCLE_CENTRE を参照してください。
14 1)	1	サイクルマスクの場合、径方向軸の絶対値は常に、半径を意味します。
15	1	直径としてのサイクルマスクのための径方向軸のインクレメンタル値
1) この機能は、MD20100 $MC_DIAMETER_AX_DEF および MD30460 $MA_BASE_FUNCTION_MASK、ビット 2 の設定に依存します		

手動ハンドル移動時の動作

直径指定が有効なときの径方向軸の手動ハンドル運転時の動作(MD11346 $MN_HANDWH_TRUE_DISTANCE == 1 または 3)は、以下のマシンデータで調整できます。

MD20624 $MC_HANDWH_CHAN_STOP_COND.Bit 15 = <value>

ビット	値	意味
15	0	指定された移動量の半分の軌跡だけ、移動します。
	1	指定された移動量をすべて移動します。

位置の値の直径での表示

径方向軸の位置の値は、以下の場合は、常に直径として表示されます。

MD27100 $MC_ABSBLOCK_FUNCTION_MASK、ビット 0 = 1

電源投入、リセット後のチャネル別の基本設定

電源投入またはチャネルリセット後、あるいは G グループ 29 (DIAMON、DIAM90、DIAMOF、DIAMCYCOF)のパートプログラム終了後のチャネル別基本設定は、マシンデータ MD20110 $MC_RESET_MODE_MASK によって定義されます。

MD20110 $MC_RESET_MODE_MASK、ビット 0 = 0:

- 電源投入:MD20150 $MC_GCODE_RESET_VALUE に準拠する基本設定
- チャネルリセットまたはパートプログラム終了:MD20150 $MC_GCODE_RESET_VALUES に準拠する基本設定

MD20110 $MC_RESET_MODE_MASK、ビット 0 = 1:

- 電源投入後:MD20150 $MC_GCODE_RESET_VALUE に準拠する基本設定
- チャネルリセットまたはパートプログラム終了の後:MD20152 $MC_GCODE_RESET_MODE に準拠する基本設定

イベント駆動型のプログラム呼び出し(ProgEvent)を介して、それぞれの希望する状態を設定できます。

電源投入後の基本設定が G96/G961/G962 である場合、経方向軸は MD20100 $MC_DIAMETER_AX_DEF により定義する必要があります。それ以外の場合は、アラーム 10870 が表示されます。

リセット中、パートプログラムの終了時、またはパートプログラムの開始時に、G96/G961/G962 の基準軸を保持するために、以下の設定を行う必要があります:

- チャネルリセットまたはパートプログラム終了:MD20110 $MC_RESET_MODE_MASK、ビット 18 = 1
- パートプログラム起動:MD20112 $MC_START_MODE_MASK、ビット 18 = 1

また、G96/G961/G962 の基準軸は、SCC[AX]による MD20100 での経方向軸のパラメータ設定なしで割り当てることもできます。この場合、周速一定制御が G96 によりすでに起動済みであることは許されません。追加情報については、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、基本編』、「送り速度制御と主軸動作」
「周速一定制御(G96/G961/G962、G97/G971/G972、G973、LIMS、SCC)」

13.3 プログラミング

径方向軸は、直径と半径の両方に関してプログラミングすることができます。一般に、これらは直径対応です。すなわち、対応する寸法情報を図面から直接パートプログラムに伝送できるように、2 倍の軌跡寸法でプログラミングされます。

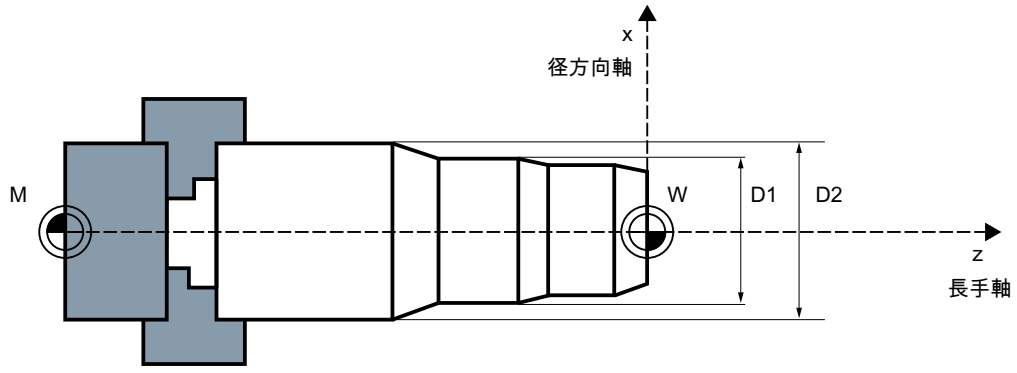


図 13-2　直径情報(D1, D2)による径方向軸

直径指定のオン/オフの切り替え

チャネル別の直径指定

直径指定の起動または解除は、G グループ 29 のモーダルで有効なパートプログラム命令を介して行われます。

- DIAMON:直径指定オン
- DIAMOF:直径指定オフ、すなわち、半径指定オン
- DIAM90:指令モードに応じた直径または半径指定:
  - アブソリュート指令 G90 については直径指定オン
  - インクレメンタル指令 G91 については半径指定オン
- DIAMCYCOF:G90 および G91 では半径指定オン、HMI に対しては、このグループで最後に有効であった G 命令が有効なまま

指令は、チャネルの径方向軸に対して排他的に行われます。

1 つのチャネルの複数の径方向軸に対する軸別の直径指定

注記

追加で指定される軸は、MD30460 $MA_BASE_FUNCTION_MASK、ビット 2 = 1 を介して起動してください。

指定される軸は、チャネルで認識された軸にしてください。ジオメトリ、チャネル、または機械軸が使用できます。

シンクロナイズドアクションでは、プログラミングできません。

パートプログラムブロックでは、以下の軸別のモーダル指令を複数回プログラム指令できます。

- DIAMONA[<軸>]:G90、G91、AC、および IC に対する直径指定オン
- DIAMOFA[<軸>]:直径指定オフ、すなわち、半径指定オン
- DIAM90A[<軸>]:指令モードに応じた直径または半径指定:
  - アブソリュート指令 G90 および AC については直径指定オン
  - インクレメンタル指令 G91 および IC については半径指定オン
- DIACYCOFA[軸]:G90 および G91 では半径指定オン、HMI に対しては、このグループで最後に有効であった G 命令が有効なまま
- DIAMCHANA[軸]:直径指定チャネル状態の引き継ぎ
- DIAMCHAN:MD304602、ビット 2=1 のすべての軸は、直径指定のチャネル状態を引き継ぎます。

軸別のモーダル命令には、チャネル設定より上の優先度があります。

13.4 必要条件

軸入れ替え

パートプログラムからの GET 要求により、軸の入れ替え中に RELEASE[<軸>]を使用して、追加の径方向軸の直径指定状態が新しいチャネルに反映されます。

シンクロナイズドアクションでの軸入れ替え

シンクロナイズドアクションの軸入れ替えの場合は、以下の場合に径方向軸が、新しいチャネルでの軸別直径指定の状態になります。

- MD304602、ビット 2=1 によって、径方向軸に軸別の直径指定が許可されている。
- チャネル解放で径方向軸が、チャネル別の直径指定に従属しない。

有効な寸法は、システム変数$AA_DIAM_STAT[<軸>]を介して確認できます。

軸コンテナ回転を介した軸入れ替え

軸コンテナを回転することにより、チャネル軸の割り当てを機械軸への割り当てに変更できます。ただし、現在の直径指定の状態は、回転後もチャネル軸に対して維持されます。これは、現在のチャネルおよび軸の状態にも適用されます。マシンデータが MD30460 $MA_BASE_FUNCTION_MASK により状態を「有効にしている」ときは、この軸コンテナのすべての軸に対して同じ状態であるためです。

13.5 例

例 1

X は、MD20100 $MC_DIAMETER_AX_DEF で定義された径方向軸です。

Y はジオメトリ軸で、U は付加軸です。これらの 2 本の軸は、ビット 2 = 1 の MD30460 $MA_BASE_FUNCTION_MASK で定義された、寸法が指定された追加径方向軸です。

DIAMON は、電源投入後は有効でなくなります。

プログラムコード	コメント
N10 G0 G90 X100 Y50	;直径指定が有効ではありません
N20 DIAMON	;X に対して有効なチャネル別の直径指定
N30 Y200 X200	;寸法:直径の X、半径の Y
N40 DIAMONA[Y]	;Y に対して有効な軸別のモーダル直径指定
N50 Y250 X300	;寸法:直径の X および Y
N60 DIAM90	;寸法:直径の X G90/AC、半径の G91/IC
N70 Y200	;Y:継続中、軸別のモーダル直径指定
N75 G91 Y20 U=DIC(40)	;寸法:直径の Y、直径の U ノンモーダル IC
N80 X50 Y100	;寸法:半径の X(G91)、直径の Y
N85 G90 X100 U200	;寸法:直径の X、半径の U
N90 DIAMCHANA[Y]	;Y はチャネル状態 DIAM90 を適用
N95 G91 X100 Y100	;寸法:半径の X および Y(G91)
N100 G90 X200 Y200	;寸法:直径の X および Y

例 2

直径指定の径方向軸は前述の例と同様です。

X および Y はチャネル 1 にあり、チャネル 2 でも登録されています。すなわち、軸入れ替えが許可されます。

プログラムコード	コメント
チャネル 1	
N10 G0 G90 X100 Y50	;直径指定が有効ではありません
N20 DIAMON	;X に対するチャネル別の直径指定
N30 Y200 X200	;寸法:直径の X、半径の Y
N40 DIAMONA[Y]	;Y 軸別のモーダルな直径指定
N50 Y250 X300	;寸法:直径の X および Y
N60 SETM(1)	;同期マーカ 1
N70 WAIT(1,2)	;チャネル 2 で同期マーカ 1 の待ち受け
チャネル 2	
...	
N50 DIAMOF	;チャネル 2 直径指定は無効
...	
N100 WAIT(1,1)	;チャネル 1 で同期マーカ 1 の待ち受け
N110 GETD(Y)	;直接 Y の軸入れ替え
N120 Y100	; Y チャネル別の直径指定、チャネル 2 で従属されます。 すなわち、半径での寸法です

13.6 データリスト

13.6.1 マシンデータ

13.6.1.1 チャネルマシンデータ

番号	識別子: $MC_	説明
20050	AXCONF_GEOAX_ASSIGN_TAB[n]	ジオメトリ軸のチャネル軸への割り当て
20060	AXCONF_GEOAX_NAME_TAB[n]	チャネル内の軸名称
20100	DIAMETER_AX_DEF	径方向軸機能が指定されたジオメトリ軸
20110	RESET_MODE_MASK	電源投入および RESET/パートプログラム終了の後の制御初期設定の定義
20112	START_MODE_MASK	NC スタートのための制御初期設定の定義
20150	GCODE_RESET_VALUES[n]	G グループの初期設定

番号	識別子: $MC_	説明
20152	GCODE_RESET_MODE[n]	リセット/パートプログラム終了時の G 命令の初期設定
20360	TOOL_PARAMETER_DEF_MASK	工具パラメータの定義
20624	HANDWH_CHAN_STOP_COND	手動ハンドル移動の特性の定義
27100	ABSBLOCK_FUNCTION_MASK	絶対値によるブロック表示のパラメータ設定

13.6.1.2 軸/主軸マシンデータ

番号	識別子: $MA_	説明
30460	BASE_FUNCTION_MASK	軸機能

P3: SINUMERIK 840D sl 用 PLC 基本プログラム 14

14.1 概略説明

一般事項

PLC 基本プログラムは、PLC ユーザープログラムと NC、HMI、および MCP 領域間の信号とデータの交換処理を分類します。信号とデータの場合は、次のグループが区別されます。

- サイクリック信号交信
- イベント起動信号交信
- メッセージ

サイクリック信号交信

周期的に交信される信号は、主としてビット配列で構成されます。

- これらには、PLC から NC に伝送される(開始または停止などの)**命令**、および NC から伝送される(プログラム実行中、中断などの)**ステータス情報**が含まれます。
- ビット領域は、以下のための信号に分類されます。
  - モードグループ
  - チャネル
  - 軸/主軸
  - 一般 NC 信号

周期的なデータ交信は、PLC 周期(OB1)の開始時に基本プログラムによりおこなわれます。これにより、たとえば、NC からの信号は 1 周期全体にわたって一定のままであることが保証されます。

イベント起動信号交信 NC → PLC

ワークプログラムの機能として実行する必要がある PLC ファンクションは、ワークプログラムで補助機能により呼び出されます。補助機能のあるブロックが実行された場合、補助機能のタイプは、NC がこの機能の実行を待つ必要がある(工具交換など) かどうか、またはこの機能をワーク加工処理とともに実行する(チェーンマガジンのあるフライス盤への工具のロードなど)かどうかを決定します。

NC の加工処理への影響を最小にするために、データ送受信はできるかぎり速く、ただしできるだけ高信頼性でなければなりません。したがって、データ送受信は、割り込み起動および確認応答駆動です。基本プログラムは、周期の開始時に、信号とデータを評価し、これを NC に通知し、データをアプリケーションインタフェースに伝送します。データがユーザー合意を必要としない場合、これは NC の処理には影響を及ぼしません。

イベント起動信号交信 PLC → NC

「イベント起動信号交信 PLC → NC」は、PLC が要求を NC へ伝送するとき(補助軸の移動など)はいつでも実行されます。この場合、データ送受信も確認応答駆動です。ユーザープログラムからおこなわれた場合、このタイプの信号交信は、ファンクションブロック(FB)またはファンクションコール(FC)を使用して起動されます。

関連する FB (ファンクションブロック)と FC (ファンクションコール)は、基本プログラムとともに提供されます。

メッセージ

ユーザーメッセージが、基本プログラムにより取得され、整理されます。メッセージ信号は、指定されたビット配列を使用して基本プログラムに伝送されます。この場合、これらの配列が使用され、メッセージイベントが発生すると、ALARM S/SQ 機能によって、PLC の割り込みバッファに入力されます。HMI (SINUMERIK Operate など)が使用中の場合、メッセージが HMI に転送されて表示されます。

PLC/HMI データ交信

このタイプのデータ交信では、HMI が主となり、バスシステムで「クライアント」と呼ばれます。HMI は、データをポーリングするか書き込みます。PLC は、これらの要求をオペレーティングシステムを介して周期的な制御ポイントで処理します。PLC 基本プログラムは、これらの交信には関係がありません。

注記

機械の機能の大部分は、PLC プログラムにより決定されます。RAM 内の各 PLC プログラムは、プログラミング装置を使用して編集できます。

ユーザーブロックのノウハウ保護

ユーザーブロック(OB、FB、および FC)に含まれているノウハウを保護するために、SIMATIC STEP 7 に含まれる SBP ツール(SIMATIC ブロックの保護)でブロックをコード化することができます。これらのブロックは、コード化のパスワードを指定しないと、開くことも、デバッグすることも、変更することもできなくなります。

符号化するときには、SIMATIC PLC-CPU と SINUMERIK PLC-CPU のどちらのオートメーションシステムの PLC-CPU でブロックを実行するかを指定してください。SIMATIC and/or SINUMERIK PLC-CPU.

CPU へのロードなどのブロックの処理は、符号化には影響されません。

必要条件

バージョン 5.5 SP3 での SIMATIC STEP 7

14.2 PLC CPU のキーデータ

PLC CPU のキーデータ

参照先:

SINUMERIK NCU に統合されている PLC CPU の主要データの概要は、以下に記載されています。

『NCU 7x0.3 PN マニュアル』、「技術仕様」の章

注記

内蔵ドライブの I/O アドレス

4096 以上の I/O アドレスは、NCU の内蔵ドライブ用に予約済みであり、それ以外に割り当てないでください。

PLC 基本プログラムの機能

有効範囲		
	軸/主軸	31
	チャネル	10
	モードグループ	10
機能		

	ステータス/制御信号	+
	M デコーダ(M00-99)	+
	G グループデコーダ	+
	補助機能の分配	+
	補助機能の伝送、割り込み起動	+
	リストに応じた M デコード	+
	PLC からの軸/主軸の移動	+
	ASUP インターフェース	+
	異常/操作メッセージ	+
	MCP と HHU 信号の伝送	+
	ハンドヘルドユニットの表示制御	+
	NC 変数と GUD の読み取り/書き込み	+
	PI サービス	+
	工具管理機能	+
	スター/デルタ切り替え	+
M 対 N		+
Safety Integrated		+
プログラム診断		+

14.3 PLC オペレーティングシステムのバージョン

PLC オペレーティングシステムのバージョンは、以下に表示されます。

- SINUMERIK Operate の操作画面: [操作エリアの切り替え|診断|バージョン] ⇒ バージョンデータ/システムソフトウェア NCU: [PLC|詳細]を選択 ⇒ バージョンデータ/システムソフトウェア NCU/PLC: PLC オペレーティングソフトウェアが、最初の行の「PLC 3xx…」に表示されます。
  注
  表示されるバージョンは SINUMERIK によって異なります。 基本 SIMATIC CPU との互換性はありません。
- SIMATIC STEP 7、HW Config: SINUMERIK ラックの PLC CPU のプロパティ: [プロパティ - CPU 3xx…|注文番号/ファームウェア]: xxxx / **Vx.y.z**
  注
  基本 SIMATIC CPU のバージョンが表示されます。

14.4 PLC モード選択

PLC モード選択は、NCU モジュールの前面にあります。以下の PLC 運転モードを PLC モード選択を使用して設定できます。

S [1)]	意味	備考
0	RUN-P	PLC プログラムを、パスワードを適用せずに変更できます。
1	RUN	プログラミング装置(PG)を使用して、読み取りアクセス操作のみが可能です。パスワードが設定されるまで、PLC プログラムを変更できません。
2	STOP	PLC プログラムの処理が停止され、すべての PLC 出力が代替値に設定されます。
3	MRES	PLC が停止状態に切り替わった後で、PLC のトータルリセット(初期設定データ)がおこなわれます。
1) PLC モード選択のスイッチ位置		

参照先

NCU モジュールの前面にある PLC モード選択の位置、ならびに NC と PLC のトータルリセットと関連したその使用についての詳細は、以下を参照してください。

『CNC 試運転マニュアル』:NC、PLC、ドライブ:

- 「スイッチオン/電源投入」の「電源投入のための操作部と表示部」の章
- 「スイッチオン/電源投入」の「NC と PLC のトータルリセット」の章
- 「一般的なヒント」の「個別の NC と PLC のトータルリセット」の章

14.5 予約リソース(タイマ、カウンタ、FC、FB、DB、I/O)

予約リソース(タイマ、カウンタ、FC、FB、DB、I/O)

基本プログラムのために、以下のコンポーネントが予約済みです。

- **タイマ**
  予約なし
- **カウンタ**
  予約なし

- **FC、FB、DB**
  FC0 ～ FC29 および FB0 ～ FB29 は、基本プログラム用に予約されています。FC および FB では、1000 と 1023 の間の範囲の番号も予約済みです。DB1 ～ DB61、DB71 ～ DB80 は、データブロック用に予約されています。さらに DB では、1000 ～ 1099 の範囲の番号も予約済みです。起動されていないチャネル、軸/主軸、および工具管理機能のデータブロックは、ユーザーが必要に応じて割り当てることができます。
- **I/O 範囲**
  PLC には、入力部および出力部のためにそれぞれ 16384 バイトの I/O アドレスがあります。4096 で始まるアドレス範囲は、内蔵ドライブにより予約/占有されます。ただし、モジュールの診断アドレスは、STEP 7 による指定に応じて、最上位のアドレス範囲に割り当てることができます。また、4080 と 4097 の間のアドレス範囲は、SIMATIC 300 ステーションのラック 0 で NC、CP、および HMI に割り当てることができます(NCU 7x0.3 の場合)。

14.6 PLC CPU のハードウェアコンフィグレーションのセットアップ

PLC CPU のセットアップの詳細については、以下を参照してください。

参照先

『CNC 試運転マニュアル』:NC、PLC、ドライブ:

- :「PG/PC の PLC への接続」の章
- :「PLC のセットアップ」の章
- :「基本編」の「PLC プログラム」の章
- :「一般的なヒント」の「個別の NC と PLC のトータルリセット」の章
- :「一般的なヒント」の「PG/PC のネットワークへの統合(NetPro)」の章

14.7 PLC プログラムのセットアップ

14.7.1 基本プログラムのインストール

基本プログラムのインストールの詳細については、以下を参照してください。

参照先

『CNC 試運転マニュアル』:NC、PLC、ドライブ』「PLC のセットアップ」の「PLC プログラムの作成」の章

注記

インストール/更新

SINUMERIK 840D sl のツールボックスをインストールする前に、SIMATIC STEP 7 をインストールしてください。

STEP 7 の更新後に、STEP 7 のハードウェア拡張をツールボックスから再度インストールすることをお勧めします。

目次

ハンドヘルドユニットと M デコード機能用の標準パラメータ設定、インタフェースシンボルおよびデータブロックテンプレートを含んだ OB ソースプログラムは、基本プログラムに含まれています。

14.7.2 基本プログラムの用途

新しい CPU プログラム(たとえば「Turnma1」)は、各据え付け(機械) 毎に STEP 7 によってプロジェクトにセットアップしてください。

備考

プロジェクトのカタログ構造とプロジェクトとユーザープログラムの作成手順については、関連する SIMATIC 取扱説明書に記載されています。

手順

SIMATIC マネージャを使って、[ファイル|開く|ライブラリ]により、基本プログラムブロックをコピーします。

以下のコンポーネントをライブラリからコピーしてください。

- ブロックコンテナから: FC、FB、DB、OB、SFC、SFB、UDT
- Source_files (ソースコンテナから): GPOB840D
- 場合により MDECLIST、HHU_DB、およびその他
- シンボルテーブル(シンボルコンテナから)

STEP 7 との互換性

基本プログラムと現在の STEP 7 バージョンの間に依存関係はありません。

14.7.3 バージョンコード

基本プログラム(Basic program)

基本プログラムのバージョンは、操作画面のバージョン画面に、コントロールシステムのタイプとともに表示されます。

コントロールシステムのタイプは、以下のようにコード化されます。

左揃えの 10 進数 DB17.DBD0 (バイト 0)	コントロールシステムのタイプ
03	SINUMERIK 840D sl (NCU 7x0)

ユーザープログラムのバージョン ID

ユーザー独自の PLC バージョンコードを HMI バージョン画面に表示することもできます。このために、任意のデータブロックに、最大 **54** 文字を含む STRING タイプのデータを定義してください。ただし、バージョンは解析されず、入力した文字列が受け入れられます。この文字列のパラメータ設定は、FB1 上のポインタによって実行されます。このため、データブロックは、シンボルを使用して定義してください。詳細については、FB1 (ページ 1162)ブロック概要を参照してください。

バージョン ID は、以下のフォーマットの文字列にすることができます。

- xx.yy
- xx.yy.zz
- ww.xx.yy.zz
- vv.ww.xx.yy.zzz
- x.y
- x.y.z
- w.x.y.z
- v.w.x.y.z

バージョン ID に加え、日付も文字列に入力できます。これは、適切なフォーマットで、バージョン画面の HMI に表示されます。ただし、日付は解析されず、入力した文字列が

受け入れられます。バージョン ID は、組み合わせることができます。次のフォーマットが現在、サポートされています。

- 00/00/0000
- 0000/00/00
- 00/00/00

例:

- 「テストプロジェクトバージョン 01.02.03 01/01/2015」
- 「1.2 2015/01/01 テストプロジェクト」
- 「01/01/15 バージョン 01.02 テストプロジェクト」

14.7.4 機械プログラム

工作機械メーカは、基本プログラムとともに提供されるライブラリルーチンを使用して機械プログラムを作成します。機械プログラムには、その機械についての論理演算およびシーケンスが含まれています。NC へのインタフェース信号もこのプログラムで制御されます。NC とのより複雑な通信機能(NC データの読み取り/書き込み、工具管理の応答など)は、基本プログラムの FC および FB のブロックを介して起動され、実行されます。

機械プログラムは、さまざまな STEP 7 作成言語(STL、LAD、FBD、S7-HIGRAPH、S7GRAPH、SCL など)で作成できます。機械プログラム全体を正しい順序で生成し、コンパイルしてください。

つまり、他のブロックにより呼び出されるブロックは、通常**これらのブロックの前に**コンパイルしてください。

プログラムを開発するため、他のブロックにより呼び出されるブロックをインタフェース(VAR_INPUT、VAR_OUTPUT、VAR_IN_OUT、VAR)で続けて修正する場合、呼び出しブロックおよびそれに関連したすべてのブロックを再度コンパイルしてください。この一般的な手順は、FB のインスタンスデータブロックにも同じように適用されます。これらの一連の操作を守らなかった場合、データが STEP 7 に再転送するときにタイムスタンプの矛盾が発生します。したがって、ブロックの再コンパイル適用性は保証されず、[ブロックのステータス]機能によって、無用な矛盾が発生する可能性もあります。さらに望ましいのは、ブロックがラダーダイアグラムまたはシングルステートメント(インクリメンタルモード)で作成されたときに、STEP 7 エディタによって ASCII-STL で生成することです。

14.7.5 データバックアップ

PLC-CPU では、シンボル名は保存しませんが、代わりにブロックパラメータ VAR_INPUT、VAR_OUTPUT、VAR_IN_OUT、VAR のデータタイプ記述およびグローバルデータブロックのデータタイプを保存します。

注記

この機械の関連したプロジェクトを使用しないで、合理的に再コンパイルすることはできません。 これが特に影響を与えるのは、たとえば、ブロックのファンクションステータスや、PLC-CPU プログラムで後で行われる必要な変更です。 したがって、この機械の PLC CPU 内に STEP 7 プロジェクトのバックアップコピーをとっておくことが必要です。 これはサービスのときに非常に役立ち、元のプロジェクトの復元で時間を無駄にするのを防ぎます。

STEP 7 プロジェクトが存在し、上で与えられた説明にしたがって作成されている場合、シンボルはこの機械の PLCCPU で処理できます。 将来のアップグレードに必要な場合に備えて、機械ソースプログラムを「.awl」ファイルとして格納することが望ましい場合もあります。

すべてのオーガニゼーションブロックおよびすべてのインスタンスデータブロックのソースプログラムを常に使用可能にしてください。

14.7.6 PLC 一括セットアップ、PLC アーカイブ

一度ブロックが PLC CPU にロードされると、機械上のデータをバックアップするために、HMI 操作画面によって一括アーカイブを生成できます。データの一貫性を確保するために、PLC が停止状態のとき、このバックアップはブロックロードの後すぐに作成してください。一括アーカイブではバイナリデータのみが保存されるため、SIMATIC プロジェクトのバックアップの代わりにはなりません。たとえば、シンボル情報は一括アーカイブに存在しません。さらに、CPU で生成された CPU DB (SFC 22 DB)や SDB は保存されません。

SINUMERIK アーカイブプログラムの選択

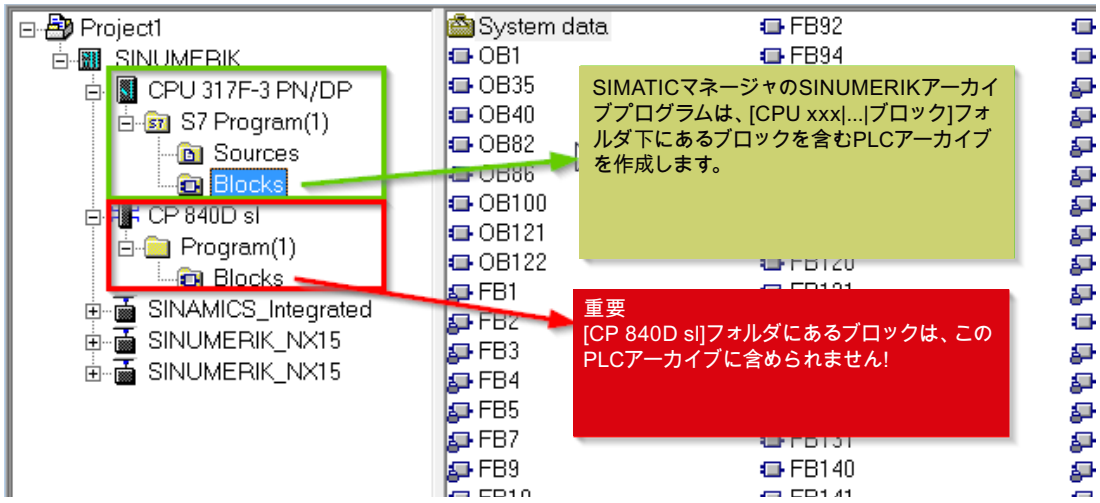
PLC 一括アーカイブは、SIMATIC プロジェクトから代替データとして直接生成できます。

1. SIMATIC マネージャで、[設定]ダイアログボックスを開きます。メニューバー[ツール|設定]を選択します。
2. [アーカイブ]タブを開きます。
3. [推奨されるアーカイブプログラム]ドロップダウンリストボックスで、SINUMERIK アーカイブプログラム「SINUMERIK (*.arc)」を選択します。

SINUMERIK アーカイブプログラムの起動

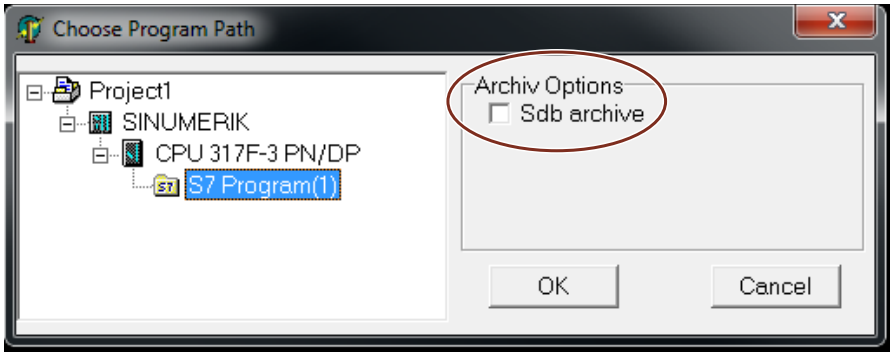
SINUMERIK アーカイブプログラムは、SIMATIC マネージャでメニュー命令[ファイル|アーカイブ]から起動できます。

プロジェクトを選択し、アーカイブ名を割り当てたら、PLC アーカイブが生成されます。プロジェクトに複数のプログラムパスが含まれている場合、PLC アーカイブを作成する S7 プログラムをダイアログボックスで選択できます。選択されたプログラムパスにある[CPU ...|...|ブロック]フォルダに含まれているすべてのブロックがアーカイブされます。[CP 840D sl]フォルダ内のブロックは、PLC アーカイブに含められません。



SFC22 (オンライン)で作成された CPU 内のデータブロックもアーカイブから除外されます。

「SDB アーカイブ」オプションをアーカイブプログラムに対して有効または無効にできます。



「SDB アーカイブ」が有効の場合、選択されたプログラムパスのシステムデータブロック(SDB)のみを含む PLC アーカイブが作成されます。

自動化

一括アーカイブの生成のプロセスは、自動化できます(STEP 7 の命令インタフェースに相当)。この一括アーカイブの生成では、命令インタフェースが拡張されます。

この拡張には、次の機能を使用できます。

ファンクション(ここでは VB スクリプトで表示)は、サーバがインスタンス化され、Magic が呼び出されるまで使用できません。

```
Const S7BlockContainer = 1138689, S7PlanContainer = 17829889

Const S7SourceContainer = 1122308

set S7 = CreateObject("Simatic.Simatic.1")

rem STEP 7 の命令インタフェースのインスタンス化

Set S7Ext = CreateObject("SimaticExt.S7ContainerExt")

Call S7Ext.Magic("")
```

ファンクション:

- `Function Magic(bstrVal As String) As Long`
- `Function MakeSerienIB (FileName As String, Option As Long, Container As S7Container) As Long`

説明

Function **Magic**(bstrVal As String) As Long

呼び出しにより、特定のファンクションにアクセスできます。ファンクションは、サーバのインスタンス化の後で一度呼び出してください。bstrVal の値は空にできます。これにより、Autoexec のパス名称と正しい STEP 7 のバージョンのチェックが開始されます。ファンクションは、戻りパラメータの 0 により有効になります。

戻りパラメータ(-1) = 間違った STEP 7 バージョン

戻りパラメータ(-2) = Autoexec.bat にエントリなし

Function **MakeSerienIB**(FileName As String, Option As Long, Container As S7Container) As Long

「オプション」パラメータ

0:	トータルリセットを伴う通常の一括セットアップファイル。
ビット 0 = 1:	トータルリセットのない一括セットアップファイル。プロジェクトにSDBが含まれる場合、このオプションは無効です。 その後、トータルリセットが必ず実行されます。
ビット 1 = 1:	PLC 再起動する一括セットアップファイル

戻り値:

0	= OK
-1	= 使用できないファンクション、事前に[Magic]ファンクションを呼び出します
-2	= ファイル名称を生成できません
-4	= コンテナパラメータが無効またはコンテナブロックが空です
-5	= 内部エラー(Windows によりメモリ要求が拒否されました)
-6	= 内部のエラー(STEP 7 プロジェクトに問題があります)
-7	= 一括セットアップファイルの生成時の書き込みエラー(ディスクがいっぱい、など)

スクリプトでの使用

プログラムコード

```
If S7Ext.Magic("") < 0 Then
    Wscript.Quit(1)
End If
    Set Proj1 = s7.Projects("new")
    set S7Prog = Nothing
    Set s7prog = Proj1.Programs.Item(1) '1 つのプログラムがある場合'
For Each cont In s7prog.Next
    If (Cont.ConcreteType = S7BlockContainer) Then
    ' ブロックコンテナのチェック
    Exit For
    End If
    Cont = Nothing
    Next
Error = S7Ext.MakeSerienIB("f:\dh\arc.dir\PLC.arc", 0, Cont)
'エラー分析
```

前述でプログラムされた For Each ... Next ブロックは、以下のように Delphi プログラミング言語でプログラムできます(C、C++プログラミング言語でのプログラミングと似ています)。

プログラムコード

```
var
    EnumVar:IEnumVariant;
    rgvar:OleVariant;
    fetched:Cardinal;

//For Each Next
EnumVar := (S7Prog.Next._NewEnum) as IEnumVariant;
While (EnumVar.Next(1,rgvar,fetched) = S_OK) Do Begin
    Cont := IS7Container(IDispatch(rgvar)); // ブロックコンテナ
    ソースのチェック
    If (Cont.ConcreteType = S7BlockContainer) Then Break;
    Cont := NIL;
End;
```

14.7.7 ソフトウェアのアップグレード

PLC ソフトウェアのアップグレード前に、定義済みの初期状態をアーカイブするには、トータルリセットをおこなってください。この場合、何よりもまず、すべてのユーザーデータ(プログラムとデータブロック)が削除されます。PLC トータルリセットの説明は、以下を参照してください。

参照先:
『CNC 試運転マニュアル』:「NC、PLC、ドライブ、一般的なヒント」、
:「PLC トータルリセット」の章

新規の SIMATIC S7 プロジェクトの生成

通常、新しい PLC 基本プログラムは新しい NCU ソフトウェアバージョンにリンクすべきです。そのために、基本プログラムブロックをユーザープロジェクト内にロードしてください。次のプログラムとデータブロックがユーザープロジェクトに既に存在している場合、それを PLC 基本プログラムのブロックで転送しないでください。OB1、OB40、OB82、OB86、OB100、FC12、および DB4 これらは、ユーザーによって変更されていると考えられるため、上書きしないでください。新しい基本プログラムはユーザープログラムにリンクしてください。次の手順を考慮してください。

1. PLC 基本プログラムをコピーする前に、すべてのユーザーブロックのテキストまたはソースファイルを生成します。
2. 新しい基本プログラムブロックを SIMATIC S7 プロジェクトにコピーします(詳細については、「基本プログラムの用途 (ページ 1049)」の章を参照してください)。
3. その後、すべてのユーザープログラム「*.awl」を正しい順序で再コンパイルしてください(「機械プログラム (ページ 1051)」も参照してください)。
4. この新規にコンパイルされた SIMATIC S7 プロジェクトを、STEP 7 で PLC にダウンロードします。

ただし、通常は S7 プロジェクトのオーガニゼーションブロック(OB)とインスタンスデータブロックを再コンパイルするだけで十分です。つまり、アップグレード前に、オーガニゼーションブロックとインスタンスデータブロックのソースのみを生成するだけでかまいません。

NC 変数

各 NC ソフトウェアバージョン(旧バージョンを含む)で、最新の NC VAR セレクタを使用できます。旧 NC ソフトウェアバージョンでも、最新のリストから変数を選択できます。DB120 (変数の初期設定 DB)のデータ内容は、ソフトウェア状態に依存しません。つまり、旧ソフトウェアバージョンで選択された変数は、ソフトウェアがアップグレードされても再選択する必要はありません。

14.7.8 I/O モジュール(FM、CP モジュール)

より複雑な I/O モジュール(FM、CP モジュール)には、一般に STEP 7 の追加パッケージが必要です。 この追加パッケージで、サポートブロック(FC/FB)が提供されます。 このブロックには、関連モジュールを操作するための特定の機能が含まれています。 これらの機能は、ユーザープログラムでパラメータ設定して呼び出すことができます。

同一の番号

処理ブロックと基本プログラムブロックの番号が同一の場合、基本プログラムのブロック番号をそのまま変更しないでください。 処理ブロックのブロック番号を、STEP 7 を使用して任意の番号に変更してください。

14.7.9 トラブルシューティング

このセクションでは、発生する可能性のある問題と、その原因および対策について説明します。ハードウェアを交換する際は、このセクションをよくお読みください。

異常、原因/概要および対策			
シリアル番号異常情報	**異常**	**原因/概要**	**異常を解除または回避するには**
1	MPI 経由での PLC への接続なし	MPI ケーブルが接続されていないか、故障しています。MPI カードに対して STEP 7 ソフトウェアが正しく設定されていない可能性もあります。	試験: STEP 7 エディタで接続「Direct_PLC」によってプログラミング装置とのリンクを作成します。 ここにはノードアドレス番号が表示されるはずです。 それらが表示されない場合は、MPI ケーブルが故障しているか、接続されていません。
2	PLC のトータルリセットを実行したにもかかわらず PLC にアクセスできません。	システムデータブロック SDB 0 が、変更された MPI アドレスでロードされています。 そのため、アドレスの二重割り当てによる MPI バスの競合が発生しました。	他のコンポーネントへの MPI ケーブルをすべて取り外します。 プログラミング装置とのリンク「Direct_PLC」を作成します。MPI アドレスを修正します。

異常、原因/概要および対策			
シリアル番号異常情報	異常	原因/概要	異常を解除または回避するには
3	PLC の 4 つの LED がすべて点滅しています(DI 障害)	PLC でシステムエラーが発生しました。 **対策:** PLC の診断バッファを読み取って、システムエラーを詳細に分析してください。 このバッファにアクセスするには、 (「PLC」スイッチを位置 2 に設定するなどの方法で) PLC を停止します。 次に、ハードウェアリセットを実行してください。 その後、診断バッファを STEP 7 で読み取ることができます。診断バッファの情報をホットライン/開発サービスに伝えてください。 ハードウェア RESET 後にトータルリセットを要求された場合は、トータルリセットを実行してください。 この場合、診断バッファは停止状態の PLC で読み取ることができます。	PLC プログラムが RESET または再ロードされた後、システムは通常運転に戻ることがあります。 その場合でも、診断バッファの内容を開発オフィスに送信してください。

14.8 PLC CPU の連結

14.8.1 一般情報

S7-300 オートメーションシステムの CPU は、SINUMERIK 840D sl の PLC として使用されます。 PLC-CPU は、サブモジュールとして NCU コンポーネントに組み込まれています。 PLC CPU のパフォーマンスデータの参照については、「PLC CPU のキーデータ (ページ 1045)」の章を参照してください。

14.8.2 PLC CPU の特性

SINUMERIK 840D sl に内蔵されている PLC は通常、対応する SIMATIC S7-300 PLC と同じ機能を持っています。

相違点については、「PLC CPU のキーデータ (ページ 1045)」の章の参照先を参照してください。

SIMATIC S7-300 PLC と比較するとメモリシステムが異なるため、特定の機能(メモリカード上にブロックを保存、メモリカードにプロジェクトを保存など)は使用できません。

注記

SINUMERIK に内蔵されている PLC と同様に、「PLC 停止」がプログラミング装置からオペレータ操作で実施された場合は、停電後に SIMATIC S7-300 PLC が復帰しても PLC は自動的に起動しません。 安全上の理由により、PLC は該当する診断エントリで停止状態のままになります。 プログラミング装置からのオペレータ操作「再起動の実行」、またはモード選択[停止|実行(ウォーム再起動)]によってのみ、PLC を起動できます。

14.8.3 内蔵 PLC とのインタフェース

物理インタフェース

SINUMERIK 840D sl により、NCU に統合された PLC は、デュアルポート RAM を介して NC と PLC の間で直接信号を交信するオプションを提供しています。

操作パネルとのデータ交信

操作パネルとのデータ交信 (TCU/OP など)は、Ethernet または PROFIBUS を介しておこなうことができます。Ethernet 経由で接続する場合、内蔵の通信プロセッサ(CP 840D sl)を使用して通信がおこなわれます。

機械操作パネル(MCP)とハンドヘルドユニット(HHU)とのデータ交信は、MPI、PROFIBUS または Ethernet を介しておこなうことができます。

プログラミング装置は、Ethernet または MPI (マルチポイントインタフェース)経由で PLC に直接接続することをお勧めします。

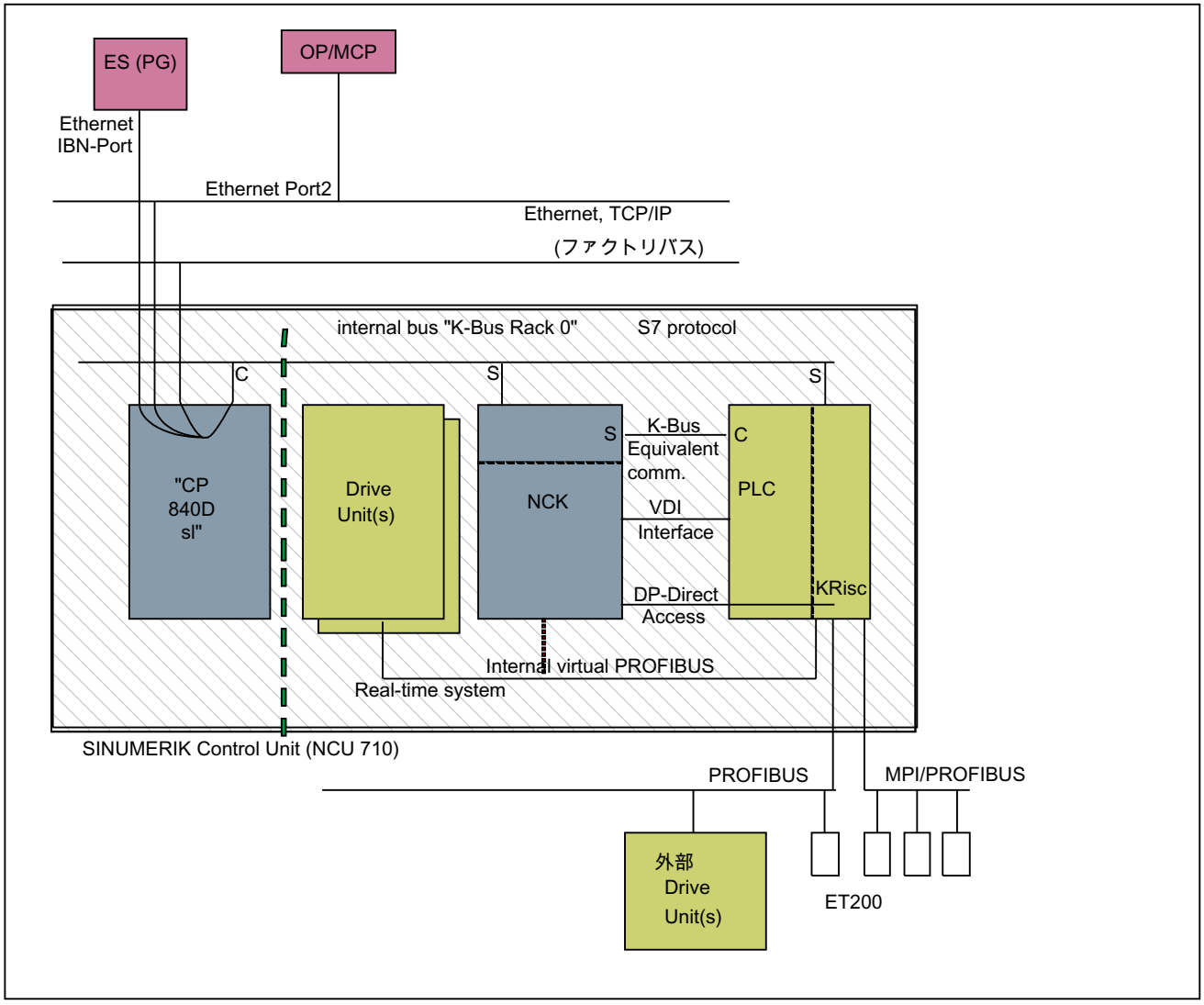


図 14-1　SINUMERIK 840D sl での NC-PLC 連結(内蔵 PLC)

インタフェース:NC/PLC

NC と PLC 間のデータ交信は、PLC 側の基本プログラムにより実行されます。NC によって NC/PLC インタフェースで保存された「プログラム実行中」などの**ステータス情報**は、(OB1)周期の開始時に基本プログラムによりデータブロックにコピーされ、その後ユーザープログラム(ユーザーインターフェース)でのアクセスが可能になります。ユーザーがインタフェースデータブロックに入力した NC の**制御信号**(NC スタートなど)も、内部 DPR に書き込まれ、周期の開始時に NC に伝送されます。

PLC に転送されたワーク/プログラム別**補助機能**は、まず、基本プログラム(アラーム駆動)で評価され、その後、OB1 の開始時にユーザーインターフェースに伝送されます。当該の NC ブロックに、NC 処理の中断(工具交換の M06 など)が必要な補助機能が含まれ

る場合は、最初に 1 回の PLC 周期の間、基本プログラムが NC ブロックの解析を停止します。その後、ユーザーが「読み込み停止」インタフェース信号を使用して、工具の変更が完了するまでブロックの実行を停止できます。一方、解析の中断を必要としない補助機能(冷却媒体オンの M08 など)だけが、関連する NC ブロックに含まれる場合は、この「高速」補助機能の転送が OB40 で直接確認されるため、解析は、PLC への伝送による影響をそれほど受けなくてすみます。

NC から伝送される **G 命令**の評価と設定もアラーム駆動であり、確認応答がありますが、それらはユーザーインターフェースに直接伝送されます。G 命令が PLC プログラム内の複数のポイントで評価される場合は、1 回の PLC 周期内で G 命令の情報の差異が発生する可能性があります。

PLC によって起動され、パラメータが割り当てられる **NC 動作**(同時軸の移動など)の場合、起動とパラメータ設定はインタフェースデータブロックではなく FC と FB を使用しておこなわれます。それらの動作に属する FC と FB は、基本プログラムとともに提供されます。必要とされる FC と FB は、ユーザーがロードし、工作機械メーカの PLC プログラム(機械プログラム)で呼び出してください。基本機能と拡張機能別にソートされた FC、FB およびデータブロックの概要は、「PLC プログラムの起動」の章を参照してください。

インタフェース:HMI/PLC

HMI/PLC データ交信は、内蔵の CP を介しておこなわれます。この場合、HMI は常にアクティブパートナ(クライアント)で、PLC は常にパッシブパートナ(サーバ)です。HMI によって伝送または要求されたデータは、PLC オペレーティングシステムによって HMI/PLC インタフェース領域での読み取りと書き込みがおこなわれます(タイミング:周期的な制御ポイント)。PLC アプリケーションから見ると、このデータは I/O 信号と全く同じです。

インタフェース:MCP/PLC または HHU/PLC (接続:Ethernet)

MCP/PLC および HHU (HT 2)/PLC データが内蔵 CP を介して交換されます。CP は MCP/HHU 信号を、PLC の内部 DPR (デュアルポート RAM)に伝送し、この DPR からフェッチします。PLC 側では、ユーザーインタフェースとの通信が基本プログラムによって処理されます。基本プログラムのパラメータ(FB1、DB7)により、オペランド領域(I/O 領域など)および先頭アドレスが定義されます。

インタフェース:MCP/PLC (接続:PROFIBUS)

MCP/PLC のデータ交信は、PLC の PROFIBUS を介しておこなわれます。MCP の I/O アドレスは、STEP 7 の HW 設定を介して、PLC のプロセスイメージ領域に配置されます。MCP*In、MCP*Out ポインタ変数は、同じアドレスに設定してください。選択した DP スレーブ番号を MCP*BusAdr に入力してください。

インタフェース:HHU/PLC (接続:MPI)

HHU/PLC データ交信は、PLC で MPI インタフェースによっておこなわれます。このために、「グローバルデータ(GD)を使用した通信」サービスが使用されます(『STEP 7 ユーザーマニュアル』も参照してください)。PLC オペレーティングシステムは、ユーザーインタフェースとの信号のやりとりを処理します。STEP 7「通信コンフィグレーション」設定ツールを使用して、GD パラメータおよびオペランド領域(I/O 領域など)とその開始アドレスの両方を定義します。

14.8.4 PLC の診断バッファ

PLC の診断バッファ(STEP 7 を使用して読み取り可能)によって、PLC オペレーティングシステムに関する診断情報が入力されます。

14.9 インタフェース構造

インタフェース DB

NC と PLC の間で多数の信号が交換されるため、インタフェース DB 内でのマッピングが必要です。これらは、PLC プログラムから見ればグローバル DB です。システム起動時に、基本プログラムによって現在の NC マシンデータ(チャネル数、軸数など)からこれらのデータブロックが作成されます。この方法の利点は、現在の機械設定に必要な最小量の PLC RAM が使用されることです。

14.9.1 PLC/NCK インタフェース

一般事項

PLC/NC インタフェースは、一方ではデータインタフェースを、他方では機能インタフェースを構成します。データインターフェースには、状態と制御信号、補助機能およびG命令が含まれる一方、PLC から NC へのジョブの転送に機能インターフェースが使用されます。

データインタフェース

データインタフェースは次のグループに分割されます。

- NC 別信号
- モードグループ別信号
- チャネル信号
- 軸/主軸/ドライブ信号

機能インタフェース

機能インタフェースは FB と FC によって形成されます。次の図に、PLC と NC 間のインタフェースの一般的な構造を示します。

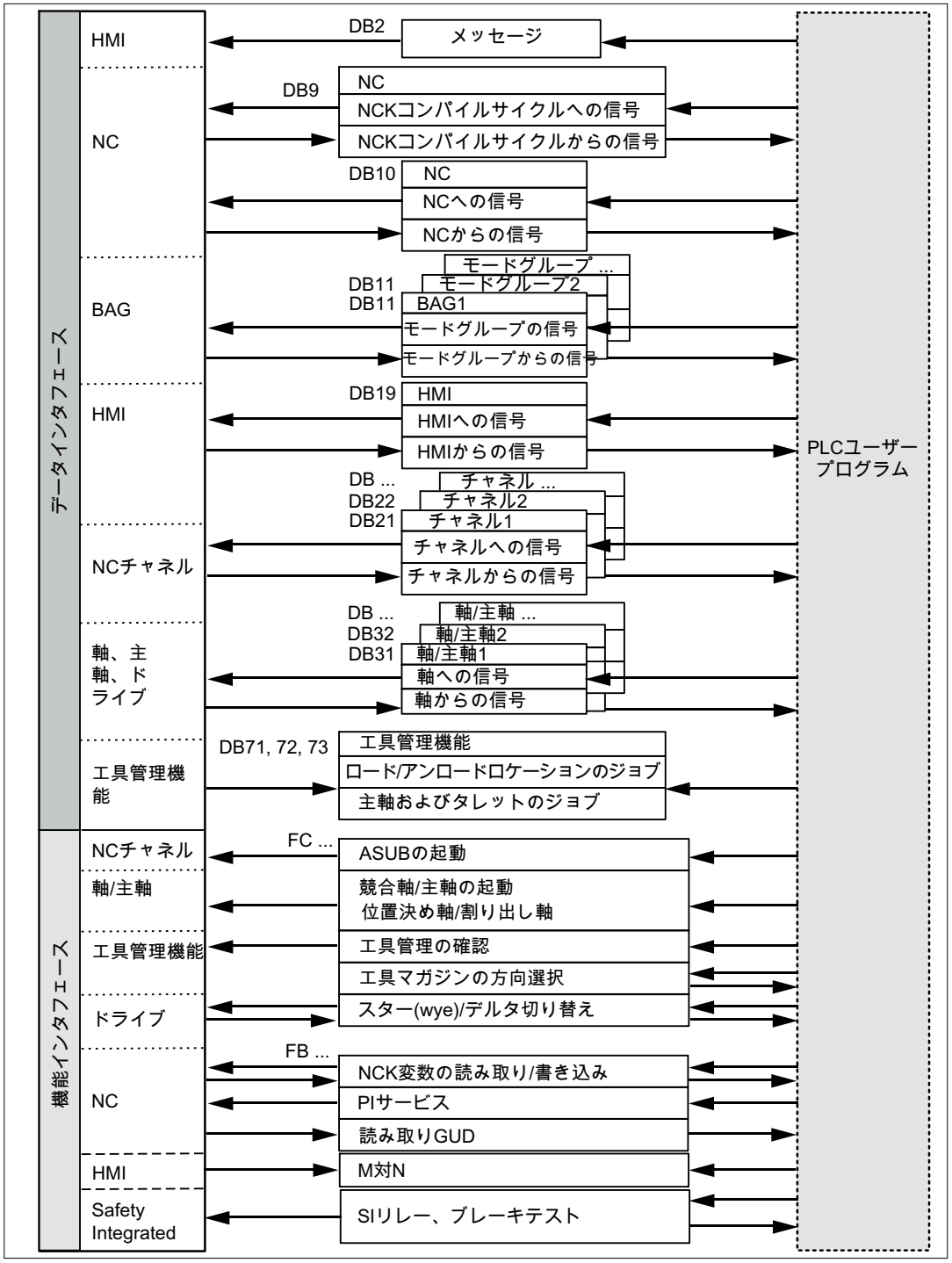


図 14-2　PLC/NC ユーザーインタフェース

コンパイルサイクル信号

PLC と NC の間で交信される標準信号に加え、コンパイルサイクル用のインタフェースデータブロック(DB9)も、必要に応じて生成されます。それに関連付けられた、コンパイルサイクルに依存する信号は、OB1 の開始時に周期的に伝送されます。基本プログラムによって、最下位のアドレスで伝送が開始され、最上位のアドレスへと進みます。信号がまず PLC から NC に伝送され、次に NC から PLC に伝送されます。ユーザーは必要に応じて NC と PLC を同期させてください(セマフォ技術の使用など)。信号伝送は NC と PLC の間で同期しません。これは、たとえばアクティブな NC データ伝送が PLC によって中断される場合があることを意味します。つまり、データの一貫性が必ずしも常に確保されないということです。

PLC/NC 信号

PLC から NC への信号グループには、次の信号が含まれます。

- NC の高速デジタル I/O 信号を変更するための信号
- キースイッチと非常停止信号

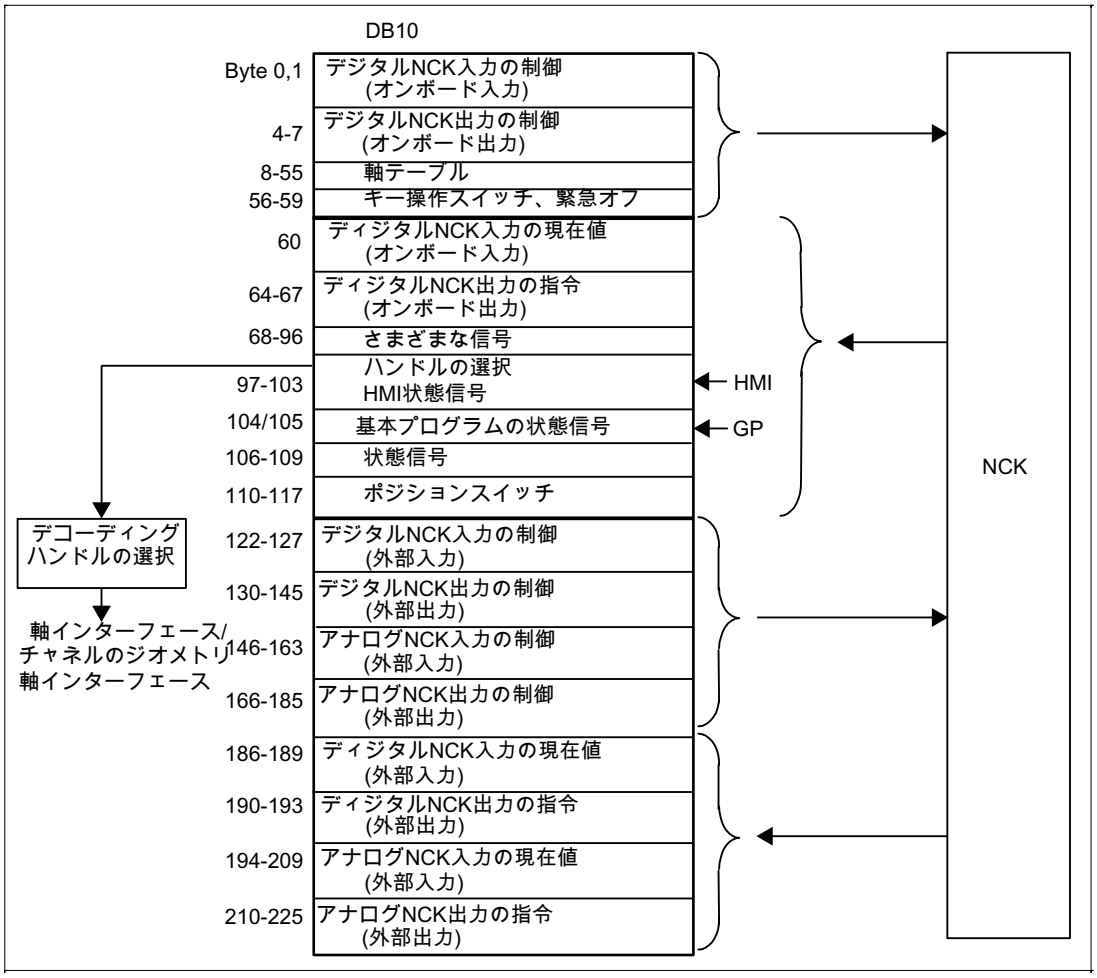


図 14-3　PLC/NC インタフェース

NC/PLC 信号

NC から PLC への信号グループには、次の信号が含まれます。

- NC のデジタルとアナログ I/O 信号の実際値
- NC の準備完了と状態信号

このグループには、HMI ハンドル選択信号とチャネル状態信号も格納されます。

ハンドル選択の信号は、基本プログラムによって解析され、機械/軸別のインタフェースに入力されます。

NC のデジタル/アナログの入力/出力

NC のデジタルおよびアナログ入出力に関しては、次の点に注意してください。

入力:

- NC のすべての入力信号または入力値も PLC に伝送されます。
- NC パートプログラムへの信号の伝送は、PLC によってマスクできます。その代わりに、PLC で信号または値を指定できます。
- PLC では、NC 側にこのチャネルのハードウェアが存在しなくても、NC に信号または値を伝送できます。

出力:

- 出力されるすべての信号または値も PLC に伝送されます。
- NC では、NC 側にこのチャネルのハードウェアが存在しなくても、PLC に信号または値を伝送できます。
- NC によって伝送される値は、PLC で上書きできます。
- PLC からの信号と値は、NC I/O を介して直接出力することもできます。

注記

デジタルとアナログの NC I/O を実行する際は、以下の文書に記載されている情報を考慮してください。
参照先:
『機能マニュアル、上級機能』、「デジタルおよびアナログ NC I/O (A4)」

PLC/モードグループ信号

機械操作パネルまたは HMI によって設定された運転モード信号は、NC の運転モードグループに伝送されます。これは、すべての NC チャネルに適用されます。複数のモードグループを NC で任意に定義できます。

モードグループによって、その現在の状態が PLC に報告されます。

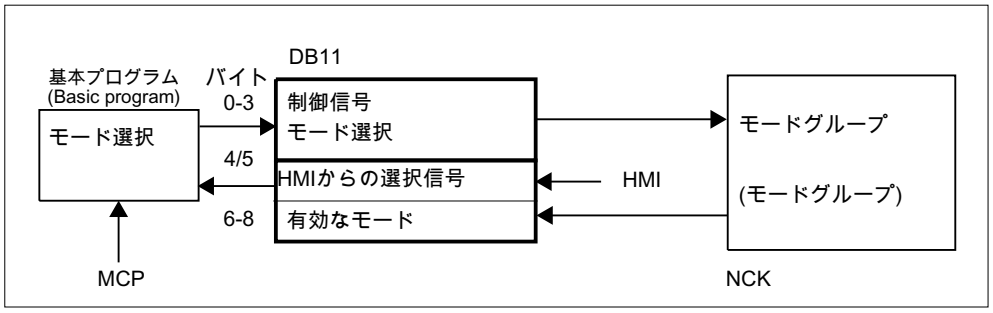


図 14-4　PLC/モードグループインターフェース

PLC/NC 信号

インタフェース上で、次の信号グループを考慮に入れてください。

- 制御/状態信号
- 補助命令/G 命令
- 工具管理機能信号
- NC 機能

制御/状態信号は、OB1 の開始時に周期的に伝送されます。HMI によってチャネル別のインタフェースに入力される信号も(HMI 信号は PLC オペレーティングシステムによって入力される) 、MCP ではなく HMI 操作パネルで定義されていれば、この時点で伝送されます。

補助命令と G 命令は、2 通りの方法でインターフェースデータブロックに入力されます。最初に、変更信号によって入力されます。

- **M 信号** M00 ～ M99 (拡張アドレス 0 とともに NC から伝送される)もデコードされ、関連インタフェースのビットが 1 周期の間、設定されます。
- **G 命令**の場合は、マシンデータを介して選択されたグループのみがインターフェースデータブロックに入力されます。
- **S 値**も、関連する M 信号(M03、M04、M05)とともに主軸別のインタフェースに入力されます。軸別の送り速度も、該当する軸別のインタフェースに入力されます。

NC で**工具管理(マガジン管理)**機能が起動されている場合、主軸またはリボルバの割り当てとロード/アンロードポイントは、別のインターフェース DB (DB71 ～ 73)に入力されます。

NC 機能の起動とパラメータ割り当ては、PLC ファンクションコールによっておこなわれます。
次のファンクションコールが使用可能です。

- 直線軸または回転軸の位置決め
- 割り出し軸の位置決め
- 準備された非同期サブプログラム(ASUB)の起動
- NC 変数の読み取り/書き込み
- マガジンと工具移動の更新

上記のファンクションの一部は、それぞれの機能取扱説明書で説明されています。

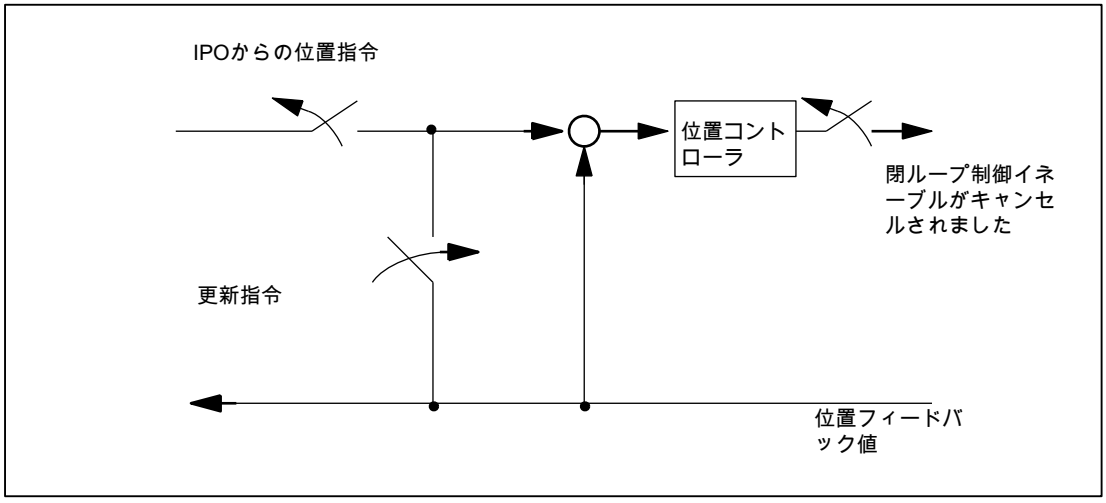


図 14-5　PLC/NC チャネルインタフェース

PLC/軸、主軸、ドライブ信号

軸別と主軸別の信号は、次のグループに分割されます。

- 共有軸/主軸信号
- 軸信号
- 主軸信号
- ドライブ信号

これらの信号は、以下の値を例外として、OB1 の開始時に周期的に伝送されます。

例外:

- 軸別の F 値
- M 値
- S 値

軸別の F 値は、NC プログラム処理中に PLC に伝送される場合、基本プログラムの M、S、F 配信処理を介して入力されます。

M と S 値も、どちらかまたは両方の値の処理が必要な場合、基本プログラムの M、S、F 配信処理を介して入力されます。

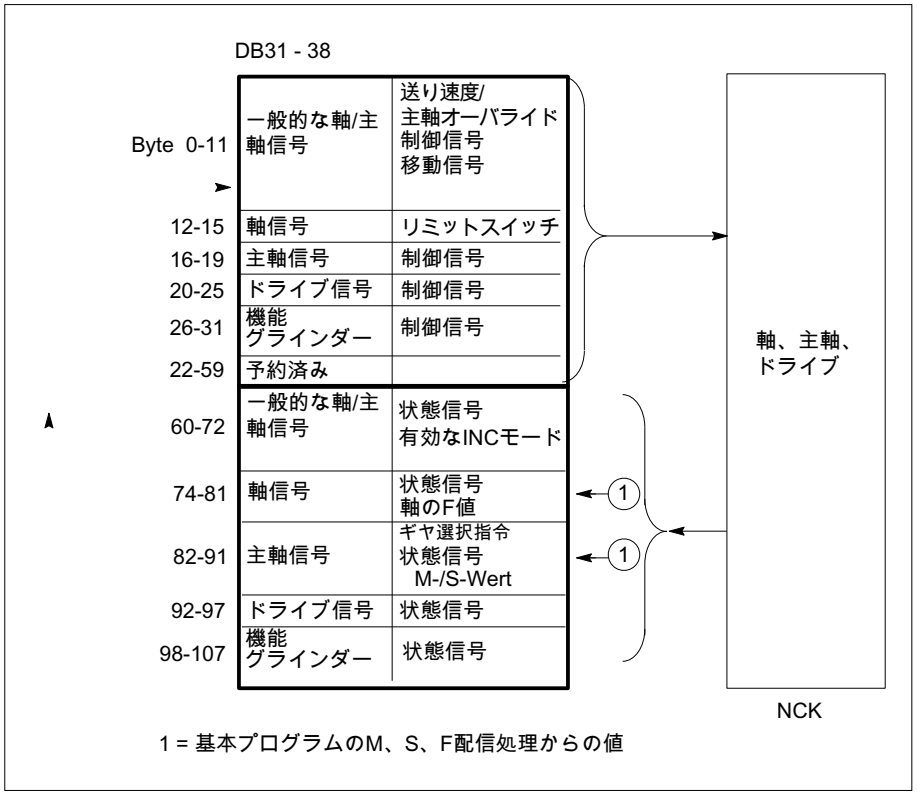


図 14-6　PLC と軸/主軸/ドライブとのインタフェース

14.9.2 PLC/HMI インタフェース

一般事項

PLC/HMI インタフェースに対して、次の機能グループを考慮してください。

- 制御信号
- 機械操作
- PLC メッセージ
- PLC 状態表示

制御信号

一部の制御信号は、HMI で考慮に入れる必要がある、機械操作パネルなどを介した信号入力です。この信号のグループには、MCS または WCS での現在値表示、キーロックなどが含まれます。これらは、別のインターフェースデータブロック(DB19)を介して HMI で交換されます。

機械操作

機械での対応動作をもたらすオペレータ入力は、すべて PLC によって監視されます。通常、オペレータ操作は機械操作パネル(MCP)でおこなわれます。ただし、モード選択など、一部のオペレータ操作を HMI 上でおこなうこともできます。

PLC オペレーティングシステムは、HMI によって送信された操作信号をインタフェースデータブロックに直接入力します。これらの操作信号は標準として、同等のオペレータアクションが使用可能な場合、それらを HMI **または** MCP でおこなうことができるように基本プログラムによって転送されます。必要に応じて、ユーザーは、FB1 パラメータ「MMCToIF」を使用して HMI 経由で操作をオフにすることができます。

PLC メッセージ

通知機能は、AS300 のオペレーティングシステムに統合されているシステム診断機能に基づいています。これらの機能には、以下の特徴があります。

- PLC オペレーティングシステムによって、すべての重要なシステムス状態と状態遷移が**診断状態リスト**に入力されます。通信イベントと I/O モジュール診断データ(診断機能を備えたモジュールの場合)も入力されます。
- システム停止をもたらす診断イベントも、タイムスタンプ付きで、イベントの発生順に**診断バッファ**(リングバッファ)に入力されます。

- 診断バッファに入力されたイベントは、ヒューマンマシンインタフェースシステム(OPまたはHMI)の準備完了信号が発行されていれば(メッセージサービス) 、バスシステムを介して、それらに自動的に伝送されます。「準備完了ノードへの伝送」は、PLCオペレーティングシステムの機能です。メッセージの受信と解釈は、HMI ソフトウェアによって実行されます。
- PLC ユーザープログラムでは、SFC(システムファンクションコール)を使用して診断バッファまたは ALARM S/ALARM SQ バッファにメッセージを入力できます。
- イベントはアラームバッファに入力されます。
  関連するメッセージテキストは、OP または HMI に格納してください。

メッセージの取得のための FC(FC10)は、基本プログラムとともに準備されます。この FC によってイベントが記録され、信号グループに分割され、アラームバッファを介して HMI にレポートされます。

図「PLC イベントの取得と通知処理」にメッセージの取得の構造を示します。次の特徴があります。

- NC/PLC インタフェースに関連するイベントのビット領域は、単一のデータブロック(DB2)内でユーザーメッセージのビットフィールドと共に結合されます。
- ビット領域は FC10 によって複数のレベルで評価されます。
  - **評価 1: グループ信号**の取得
    1 つ以上のビットが「1」に設定されている場合、信号グループ毎にグループ信号が生成されます。この信号は一般に、(診断機能を備えたモジュールの) NC/PLC インタフェースのディセーブル信号にリンクされます。グループ信号は周期内で完全に取得されます。
  - **評価 2: アラームメッセージ**の取得
    グループ内のどの信号で、その信号が「0」から「1」に変更された場合にアラームメッセージが生成されるかを定義する、固定された指定が存在します。
  - **評価 3: 操作信号**の取得
    グループ内のどの信号によって操作メッセージが生成されるかを定義する、固定された指定が存在します。
- ユーザービット領域(ユーザー領域)の範囲は、それぞれ 8 バイトの 10 の領域に初期設定されますが、FB1 の基本プログラムパラメータを介して、工作機械メーカの要望に合わせて領域数を調整することもできます。

確認応答の仕様

異常と操作メッセージ用に次の確認応答手順が実装されています。

操作メッセージは、正常な動作状態をユーザー向けの情報として表示するためのものです。したがって、このタイプのメッセージには認応答信号が必要ありません。診断状態

リストに入力と出力メッセージのエントリが作成されます。HMI によって、「operating message arrived」と「operating message gone」という識別子を使用して、既存の操作メッセージの最新ログが保守されます。

アラームメッセージは、機械上の異常状態を表示するために使用されるもので、通常は機械の停止につながります。複数の異常が間断なく発生する場合は、トラブルシューティングのためにそれらの発生順序を区別できるようにすることが重要です。これは、一方では異常が診断バッファに入力される順序によって、他方ではすべてのエントリに割り当てられるタイムスタンプによって示されます。

異常の原因が消滅しても、関連するアラームメッセージはユーザーが(MCP 上のキーを押すなどして)それに確認応答した場合にしか削除されません。この信号に応答して、レポートされたどのエラーが消去されたかが「メッセージの取得」FC によって調べられ、それらが「Alarm gone」のエントリとともに診断バッファに入力されます。これによって、保留中のアラームメッセージの最新ログを HMI で維持できるようになります。未処置のメッセージについて、(受信された問い合わせと異なり)異常が発生した時間の表示は維持されます。

STEP 7

SIMATIC Manager で、メニュー項目[目標システム|CPU メッセージ]を介して、ツールを起動できます。このツールを使用して、アラームとメッセージを番号順に表示できます。そのためには、[アラーム]タブを有効にし、画面上半分にある[A]にチェックマークを付けます。

ユーザープログラム

ユーザー PLC プログラムでは、基本プログラムブロック FC10 を、周期的なプログラム区間内で適切なパラメータ設定とともに呼び出し、DB2 でビット領域を設定またはリセットするだけでかまいません。必要なそれ以上の対策は、基本プログラムと HMI によって実施されます。

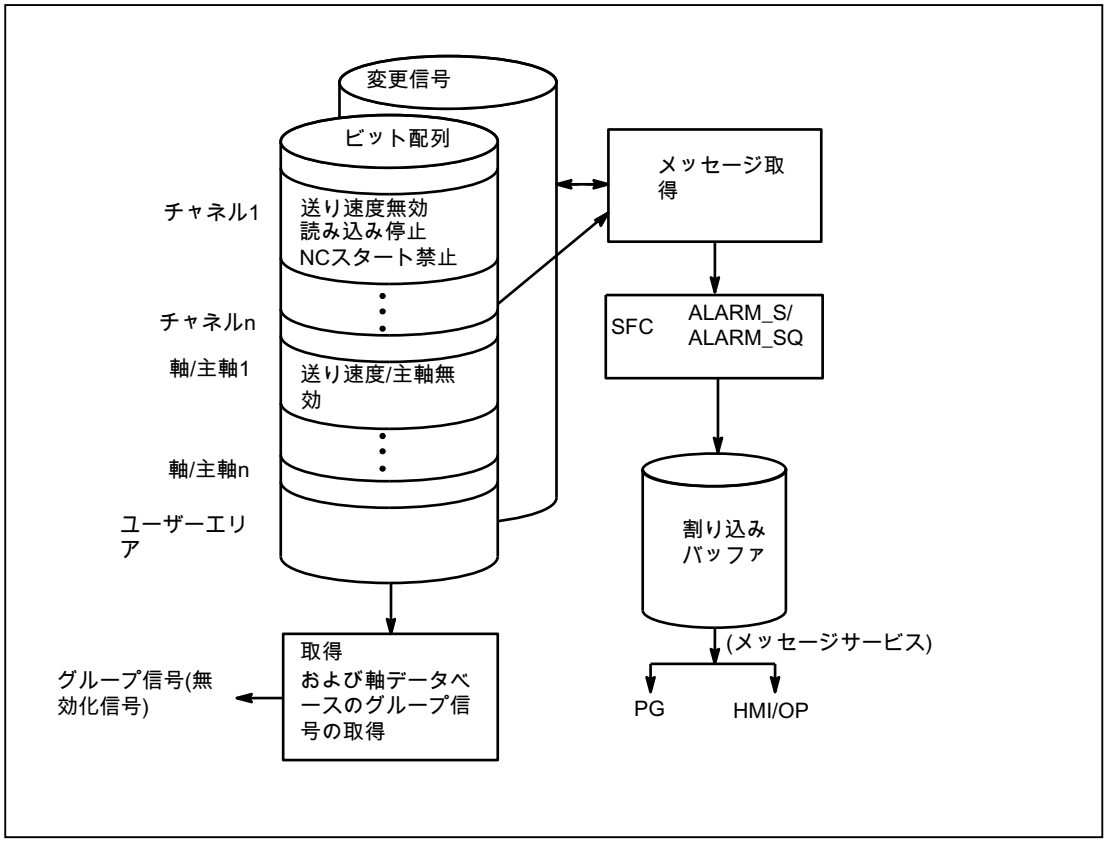


図 14-7　PLC イベントの取得と通知処理

ブロック FC10 を介した PLC アラームの拡張

FB1 パラメータ「ExtendAlMsg」により、PLC アラームメカニズムが選択されます。

「ExtendAlMsg:= FALSE」の場合は、DB2 を伴う FC10 の以前のプロセスがビットフィールドデータブロックとして有効です。チャネル数と軸数に関する既知の制限が適用されます。

一方、「ExtendAlMsg:= TRUE」の場合は、FC10 の拡張が有効になります。DB2 と DB3 は、以前と同様に作成されます。ユーザーは、DB2 のビットを設定またはリセットしてください。メッセージとアラームによるパラメータ設定と、ユーザーアラームの 2 番目の桁の数値のパラメータ設定は、DB5 に含まれます。

拡張の内容は次のとおりです。

- 10 個のチャネル、31 本の軸、64 個のユーザー領域(ユーザー領域の数は、FB1 パラメータ「MsgUser」に入力してください)のサポート。
- 送り停止、読み込み停止などの領域がメッセージなしで使用可能になります。この領域からの情報は、グループ信号としての関連メッセージビットとともに、FC10 パラメータ「ToUserIF」に応じて、DB21、DB31 のインターフェースに設定されます。そのため、以前の面倒な信号処理が省略されます。
- アラーム/メッセージには、ユーザー領域 0 のアラーム番号に加え、16 ビット整数値(アラームテキスト中の%Z パラメータ)も追加されます。ユーザーは、アラームビットの設定と並行して、配列変数 ZInfo0 の DB2 に 16 ビット整数値を書き込んでください。ユーザー領域 0 の各ビットで整数値が使用可能です。基本プログラムの UDT1002 を参照してください。
- ユーザーメッセージは、メッセージ番号の 2 番目の桁で、0 ～ 9 の範囲の数値で設定できます。2 番目の桁の表示値は、配列変数 UserDek2No の DB5 にユーザーが書き込んでください。ユーザー領域毎に番号を定義できます。基本プログラムの DB5 を参照してください。
  2 番目の桁には、値 0 が初期設定されます。

UDT1002 の DB2 の構造を認識できます(基本プログラム)。新規アラームファンクションの場合、UDT1002 をシンボルを使用して DB2 に割り当ててください。

DB2 の開始時に、アラーム生成のない信号のビット配列があります。この後に、ユーザー領域 0 に関する追加情報用に、サイズ 64 の整数値の配列が続きます。

その後ろに、やはりアラーム/メッセージがトリガされる領域が続きます(リストマニュアルを参照してください)。これらの領域は、10 のチャネル、31 の軸に拡張されます。

新規アラームでのユーザープログラムの単純実装

基本プログラムのソースコンテナには、ファイル「udt2_for_Convert.awl」が含まれます。このファイルには、UDT1002 からの次の構造要素が含まれます。

- 1 ... 8 の配列としての ChanA
- 1 ... 18 の配列としての AxisA
- 0 ... 63 の配列としての UserA

この UDT2 は、LAD/FBD/STL エディタを介してコンパイルされます。UDT2 をシンボルテーブルで DB2 に割り当ててください。

DB 2 での割り当てのあるコンポーネントに対してソースを生成してください。または、すべてのブロックに対して、そのままソースを作成することもできます。UDT1002 をシ

ンボルテーブルで DB2 に割り当ててください。その後、ソースを再コンパイルしてください。

これで、すべてのアラーム割り当てが DB2 内の新しいデータ領域に割り当てられ、FB1 のパラメータ「ExtendAlMsg」のみを TRUE に設定する必要があります。

電源投入 RESET 後のアラームの挙動は以前と同じです。

14.9.3 PLC/MCP/HHU インタフェース

一般事項

機械操作パネル(MCP)とハンドヘルドユニット(HHU)にはさまざまな接続オプションがあります。これは、一部には MCP と HHU の歴史によるものです。ここでの説明では、主に Ethernet コンポーネントの接続に重点を置きます。

SINUMERIK 840 D sl では、機械操作パネル(MCP)とハンドヘルドユニット(HHU)は Ethernet バスを介して接続され、TCU もこのバスによって NCU にリンクされます。その利点は、操作ユニットの接続に 1 つのバスケーブルが必要なだけだということです。

SINUMERIK 840D sl のネットワーク形態

840 D では、機械操作パネルとハンドヘルドユニットが CP 840D sl の Ethernet バスに接続されます(後述の図を参照してください)。カスタマイズされた操作パネル用にそれ以上のキーとディスプレイの接続が必要な場合は、追加キーボードインタフェース(操作ユニットのない機械操作パネル)を使用できます。キーボードインタフェース毎に、64 の押ボタン、スイッチ等と、64 の表示部品をリボンケーブル経由で接続できます。

MCP から送信された信号は、内蔵 Ethernet CP-840D sl によって PLC の DPR(デュアルポート RAM)にコピーされます。PLC の基本プログラムによって、FB1 で設定されている入力イメージに入力信号が入力されます。NC 関連信号は通常、基本プログラムによって NC/PLC インタフェースに配信されます。必要に応じて、ユーザーがそれらの信号を変更できます。

PLC から MCP(ディスプレイ)への信号は、反対方向に伝送されます。

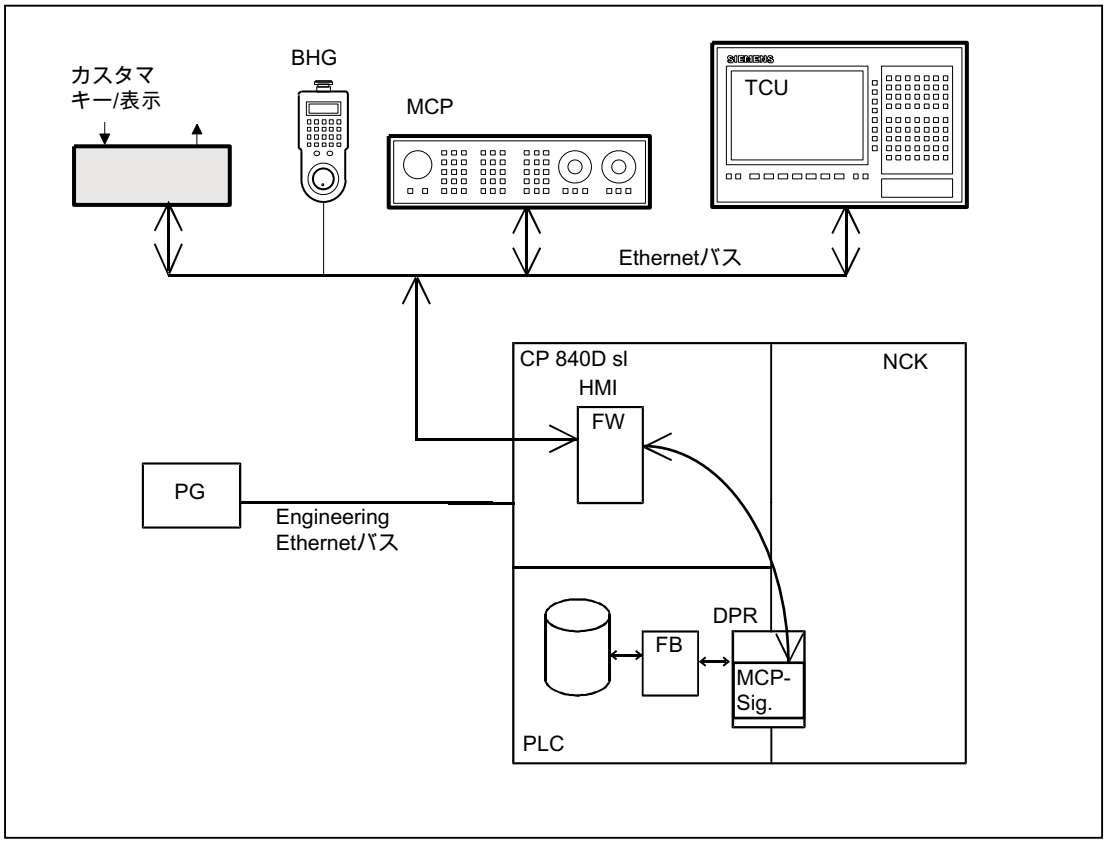


図 14-8　840D sl での機械操作パネルの接続

バスアドレス

Ethernet コンポーネントでは、MAC と IP アドレスまたは論理名称が、通信に関する決定要因となります。コントロールシステムのシステムプログラムによって、論理名称が MAC または IP アドレスに変換されます。PLC では、論理名称の数値部分が通信に使用されます。この数値部分は、ユーザーにより、パラメータ「MCPxBusAdr」を使用して FB1 に指定されます。

MCP または HHU の論理名称は常に「DIP」で始まります。これに、MCP コンポーネントのスイッチ位置に対応する番号(DIP192、DIP17 など)が続きます。

PLC の MCP インタフェース

機械操作パネルからの信号は、初期設定で I/O インタフェースを介して PLC 領域に送信されます。NC 別の信号と機械別の信号を区別してください。NC 別のキー信号は、初期設定で FC19 (または、MCP のタイプに応じて FC24、FC25、FC26)によって、当該のモードグループ、NC、軸および主軸別のインタフェースに配信されます。MCP インタフェースに送信される関連状態信号には、その逆が当てはまります。そのために、FC19 または上記の他のブロックを**ユーザープログラム**で呼び出してください。

広範な機械機能のトリガに使用できるカスタマイズされたキーは、ユーザープログラムで直接使用してください。ユーザープログラムによって、状態信号も LED の出力領域に送信されます。

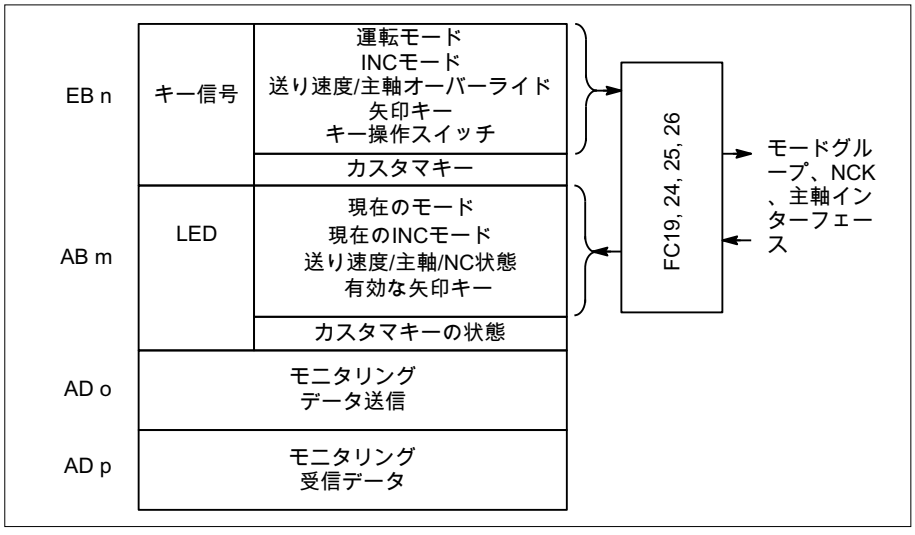


図 14-9　機械操作パネルへのインタフェースと機械操作パネルからのインタフェース

14.10　基本プログラムの構造とファンクション

一般事項

PLC プログラムはモジュール構造です。オーガニゼーションブロック(OB)が、オペレーティングシステムと基本プログラムとユーザープログラムとの間のインタフェースを形成します。

- 起動と同期(OB100)による再起動(ウォーム再起動)
- サイクリック運転(OB1)

- プロセスアラーム(OB40)
- 非同期エラー:診断アラーム(OB82)、モジュール故障(OB86)

基本プログラムとユーザープログラムのファンクションブロックの呼び出しは、ユーザーがオーガニゼーションブロック(OB)でプログラム指令してください。

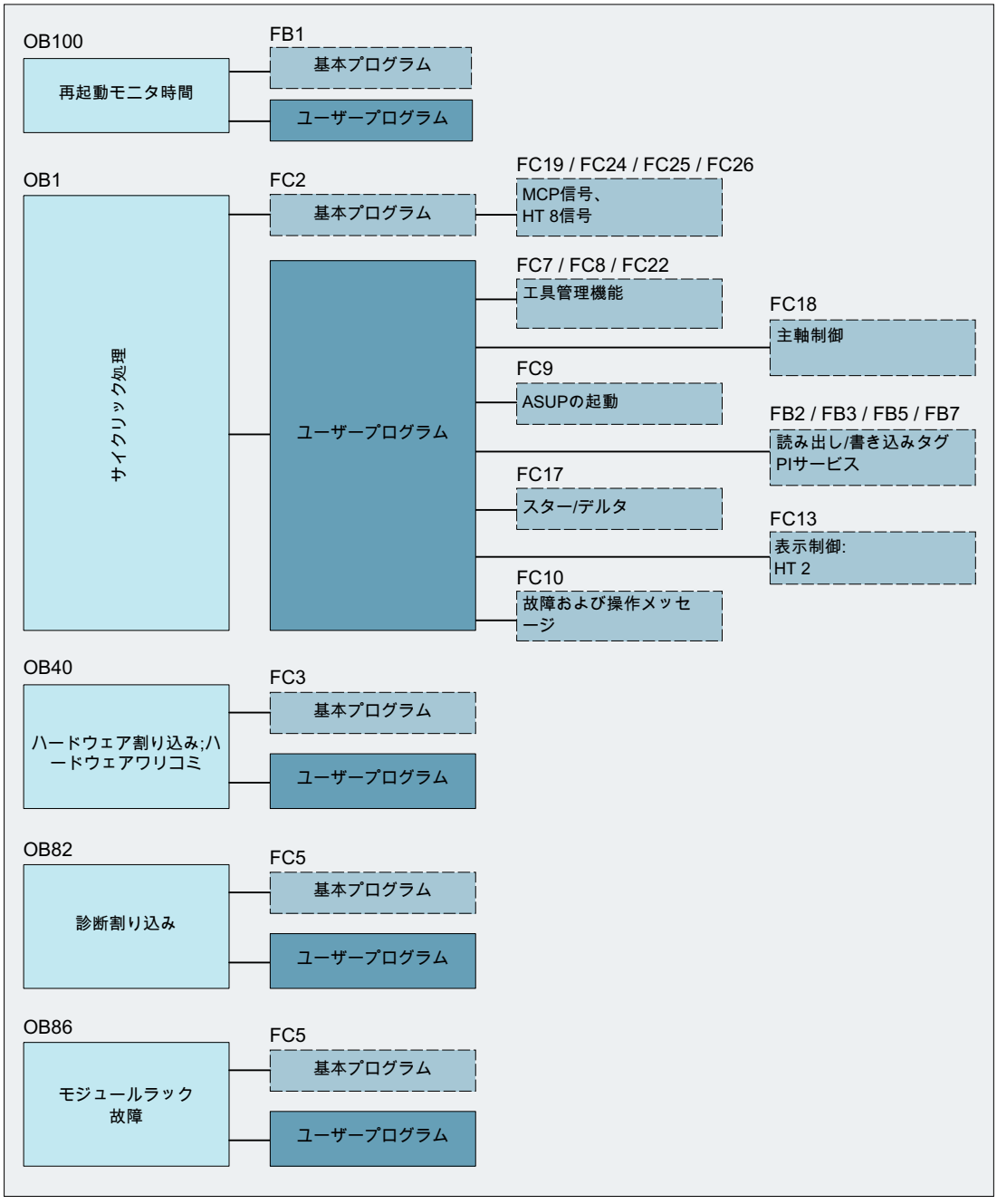


図 14-10　基本プログラムの構造(原則)

14.10.1 NCK PLC の起動および同期

基本プログラムのロード

基本プログラムは、PLC が停止状態のとき S7 ツールでロードしてください。これによって、基本プログラムのすべてのブロックが、次回呼び出されたとき正常に起動されます。そうでないと、PLC で不安定な状態(すべての PLC LED の点滅など)が発生します。

起動

NC と PLC の同期は起動時に行われます。システムおよびユーザーデータブロックの整合性が確認され、最も重要な基本プログラムパラメータの妥当性が検証されます。異常が発生した場合、基本プログラムによってアラーム(HMI で表示可能)が生成され、PLC が停止状態に切り替わります。

ウォーム再起動は準備されていません。つまり、システム初期化後、オペレーティングシステムによってオーガニゼーションブロック OB100 が実行され、周期的な実行が常に OB1 の先頭で開始されます。

同期制御

PLC は、電源投入時に HMI、NC、および CP と同期されます。

ライフサイン

正常な初期起動と最初の OB1 周期(初期設定周期)完了の後、PLC と NC によってライフサイン信号が継続的に交信されます。NC からのライフサイン信号が提示されない場合は、PLC/NC インタフェースが初期化され、DB10 の信号「NC CPU 準備完了」が FALSE に設定されます。

14.10.2 サイクリック運転(OB1)

一般事項

NC/PLC インタフェースは、完全にサイクリックモードで処理されます。時系列の視点から見れば、基本プログラムはユーザープログラムに先行して実行されます。基本プログラムの実行時間を最小化するため、制御/状態信号のみが周期的に伝送されます。補助命令と G 命令の伝送は、要求によってのみ実行されます。

次の機能は基本プログラムのサイクリックセクションでおこなわれます。

- 制御/状態信号の伝送
- 補助機能の配信
- M デコーディング(M00 ～ M99)、
- M、S、F 配信
- NC を介した MCP 信号の転送
- ユーザー異常と使用可能メッセージの取得と選別

制御/状態信号

制御と状態信号の共通の特徴は、それらがビット領域であるということです。基本プログラムでは、それらが OB1 の開始時に更新されます。

これらの信号は、次のグループに分割できます。

- 一般信号
- モードグループ別の信号(モードなど)
- チャネル信号(プログラムと送り速度変更など)
- 軸と主軸信号(送り無効など)

補助命令と G 命令

補助命令と G 命令には、次の特徴があります。

- PLC への伝送はブロック同期型です(パートプログラムブロックを参照します)。
- 転送は確認応答により起動されます。
- 確認応答時間が、確認応答の必要な補助機能を含む NC ブロックの実行時間に直接影響を及ぼします。

数値の範囲を次の表に示します。

機能	構造		値の範囲		データタイプ	
	1 番目の値	2 番目の値	1 番目の値	2 番目の値	1 番目の値	2 番目の値
G 命令		G 命令		255[1)]		バイト
M ワード	M グループ	M ワード	99	99.999.999	ワード	ダブルワード

機能	構造		値の範囲		データタイプ	
S ワード	主軸番号	S ワード	6	浮動小数点 2)	ワード	ダブルワード
T ワード	マガジン番号	T ワード	99	65535	ワード	ワード
ダブルワード	-	ダブルワード	99	255	バイト	バイト
H ワード	H グループ	H ワード	99	浮動小数点	ワード	ダブルワード
F ワード	軸番号	F ワード	18	浮動小数点	ワード	ダブルワード

1) 相対番号、G グループ毎に伝送

2) 対応する STEP 7 形式(24 ビットの仮数、8 ビットの指数)

NC によって送信される M、S、T、H、D および F 値は、付随する変更信号とともに、補助命令/G 命令を介して **CHANNEL DB** インタフェースに出力されます(『リストマニュアル』を参照してください)。補助機能の 2 つの値が、該当するデータワードに伝送されます。付随する変更信号が 1 回の PLC 周期の間、1 に起動されます。変更信号がリセットされると、確認応答が NC に渡されます。高速補助機能の確認応答は、基本プログラムが補助機能を検出すると実行されます。

補助命令と G 命令の配信に加え、選択された信号が、以下の説明のとおりに処理されます。

M デコーダ

M 機能を使用して、切り替え命令と固定小数点値を伝送できます。標準 M 機能(範囲 M00 ～ M99)に対して、デコードされたダイナミック信号が **CHANNEL DB** インターフェースに出力されます(信号持続時間 = 1 制御周期)。

G グループデコーダ

NC によって送信される G 命令の場合、関連するグループがデコードされ、現在の G 番号が CHANNEL DB の対応するインタフェースバイトに入力されます。つまり、有効なすべての G 命令がチャネル DB に入力されます。入力された G 命令は、NC プログラムが終了または中止された後も保持されます。

注記

システム起動時に、すべての G グループバイトが値「0」で初期化されます。

M、S、F 配信処理

M、S、F 配信処理は、軸送りに関する主軸別の M ワード M(1...6)=[3,4,5]、S ワードと F ワードを該当する**主軸と軸データブロック**に入力するために使用されます。配信の基準は拡張アドレスであり、M ワード、S ワードおよび軸 F ワードについてはこれが PLC に渡されます。

MCP 信号伝送

バス接続に応じて、MCP 信号が PLC に直接、または基本プログラムを使用し内部プロシージャを介して間接的に、パラメータ設定された I/O 領域に伝送されます。

ユーザーメッセージ

ユーザー異常メッセージと使用可能メッセージの取得と処理は、基本プログラムで FC によっておこなわれます。

14.10.3 時間割り込み処理(OB35)

ユーザーは、時間割り込み処理のために **OB35** をプログラム指令してください。OB 35 の初期設定の時間基準設定は 100 ms です。STEP7 の「HW Config」ツールを使用して、別の時間基準を選択できます。ただし、PLC CPU 停止を回避するため、OB35 の時間設定を 3 ms 以上にしてください。停止は、HMI の電源投入時に HMI システムステータスリストが読み取られることによって発生します。この読み取り処理によって、優先クラス制御が約 2 ms 間ブロックされます。時間基準がより低い値に設定された OB35 では、正しく処理されなくなります。

14.10.4 プロセス割り込み処理(OB 40)

プロセス割り込み **OB40** (割り込み)は、たとえば、適切に設定された I/O によって、または特定の NC 機能によって起動できます。別の割り込み要因のため、PLC ユーザープログラムでは、まず割り込みの原因を OB 40 で解析してください。割り込みの原因は、OB40 のローカルデータに含まれます。

参照先:
SIMATIC STEP 7 説明書または STEP 7 のオンラインヘルプ

14.10.5 診断アラーム、モジュール故障処理(OB82、OB86)

概要

診断アラームまたは I/O モジュールの故障が発生した場合には、基本プログラムブロック OB82 または OB86 が呼び出されます。基本プログラムブロック FC5「診断アラームとモジュール故障」 (ページ 1267) が、これらのブロックから呼び出されます。

バス診断

PROFIBUS 接続 MPI/DP、DP1、または PROFINET 接続 PN の DP スレーブモジュールの状態は、特定のバスシステムの「スレーブ OK」グループ信号を使用して基本プログラムによってユーザープログラムに通知されます。

- DB10.DBX92.0 == 1 (MPI/DP バス:スレーブ OK)
- DB10.DBX92.1 == 1 (DP1 バス:スレーブ OK)
- DB10.DBX92.2 == 1 (PN バス:スレーブ OK)

このグループ信号は、それぞれのバスシステムの LED 状態(システムステータスリスト SZL 0x174)から導出されます。

アラーム出力

特定のバスシステムでスレーブの異常または故障が検出された場合、次のアラームが表示されます。

- アラーム 400551 (MPI/DP バス)
- アラーム 400552 (DP1 バス)
- アラーム 400553 (PN バス)

このアラームは、異常が取り除かれると再度自動的に削除されます。

アラーム出力 400551、400552、400553 のマスク

次のいずれかの信号を設定することによって、アラームは特定のバスシステムに抑制されます。

- DB10.DBX92.4 = 1 (抑制アラーム 400551)
- DB10.DBX92.5 = 1 (抑制アラーム 400552)
- DB10.DBX92.6 = 1 (抑制アラーム 400553)

アラームは、信号が設定されると直ちに抑制されます。故障が発生する前、または故障が既に有効であるとき。

信号がリセットされると、故障時に対応するアラームが表示されます。

この信号は、制御システムの電源投入時にリセットされます。

注記

この信号を設定することで、すべてのバスラインに対する故障監視が解除されます。

14.10.6 NCK 故障に対する応答

概要

PLC は、サイクリック運転中に連続的にライフサインを確認することにより NC の有効性を監視します。NC が反応しなくなった場合は、NC/PLC インタフェースが初期化され、NC からの信号領域内の **NC-CPU 準備完了**の信号**(DB 10.DBX 104.7)**がリセットされます。さらに、NC と PLC の間で送受信される信号が初期状態に設定されます。

PLC 自体は引き続き有効であるため、機械機能の制御を継続できます。ただし、機械を安全な状態に置くことは、依然としてユーザープログラムの責任です。

NC → PLC 信号

NC から PLC に送信される信号は、次のグループに分割されます。

- NC、チャネル、軸および主軸からの状態信号
- 補助機能の変更信号

- 補助機能値
- G 命令値

状態信号:

NC、チャネル、軸および主軸からの状態信号がリセットされます。

補助機能変更信号:

補助機能変更信号もリセットされます。

補助機能値:

NC によって起動された最後の機能をトレースできるように、補助機能値が保持されます。

G 命令値:

G 命令値はリセットされます(つまり、それぞれが値 0 で初期化されます)。

PLC → NC 信号

PLC から NC に送信される信号は、制御信号と、FC によって NC に伝送される制御信号と処理に分割されます。

制御信号:

PLC から NC への制御信号が凍結されます。PLC 基本プログラムによる周期的な更新が一時停止します。

PLC から NC へのジョブ:

NC にジョブを渡すために使用される FC および FB は、PLC ユーザープログラムで処理する必要はなくなります。これが不正なアンサーバック信号の生成をもたらす可能性があるからです。コントロールシステムの電源投入時には、**NC-CPU 準備完了**信号が設定されるまで、ユーザープログラムでジョブ(NC データの読み取りなど)を起動しないでください。

14.10.7 ユーザープログラムから呼び出される基本プログラムのファンクション

一般事項

OB1、OB40、および OB100 の開始時に呼び出される基本プログラムのモジュールに加え、ユーザープログラムの適切なポイントでパラメータ付きで呼び出され、供給される必要のあるファンクションも提供されます。

これらのファンクションを使用して、たとえば次のジョブを PLC から NC に渡すことができます。

- 同時軸の移動(FC18)
- 非同期サブプログラムの起動(ASUB) (FC9)
- NC プログラムの選択(FB4)
- 主軸の制御(FC18)
- 読み取り変数と書き込み変数(FB2、FB3)

注記

PLC 基本プログラムのファンクションコールのチェックと診断

トリガによって(たとえば、Req、Start パラメータなどによって)制御され、出力パラメータとして(たとえば、Done、NDR、Error パラメータなどによって)実行確認応答を提供する、PLC 基本プログラムのファンクションコール(FB または FC)のチェックと診断を簡素化するには、以下の手順を実施します。

ファンクションコールの起動を生成する、他の信号でコンパイルされる変数を設定してください。起動条件は、パラメータ Done、NDR および Error の状態に応じてのみリセットできます。

この制御メカニズムは、ファンクションコールの前または後ろに配置できます。このメカニズムを呼び出しの後ろに配置する場合は、出力変数をローカル変数として定義できます(利点:グローバル変数、マーカ、データ変数の削減、およびデータ変数全体の時間関連の利点)。

トリガパラメータはグローバル変数(マーカ、データ変数)にしてください。

まだ有効になっているジョブは、ユーザープログラムから OB100 でリセットしてください(Req、Start、
パラメータなどを TRUE ⇒ FALSE)。POWER OFF/ON によって、ジョブが引き続き有効な状態となる場合があります。

同時軸

同時軸を他と区別する特徴は、次のとおりです。

- 同時軸は NC マシンデータを介して定義する必要があります。
- JOG キーによって、PLC からも、NC からも移動できます。
- PLC からの開始は、NC 運転モード MDI および AUTOMATIC で FC を介して可能です。
- この移動開始は NC ブロック境界に依存しません。

位置決め軸、割り出し軸および主軸に対してファンクションコールが使用可能です(FC18)。

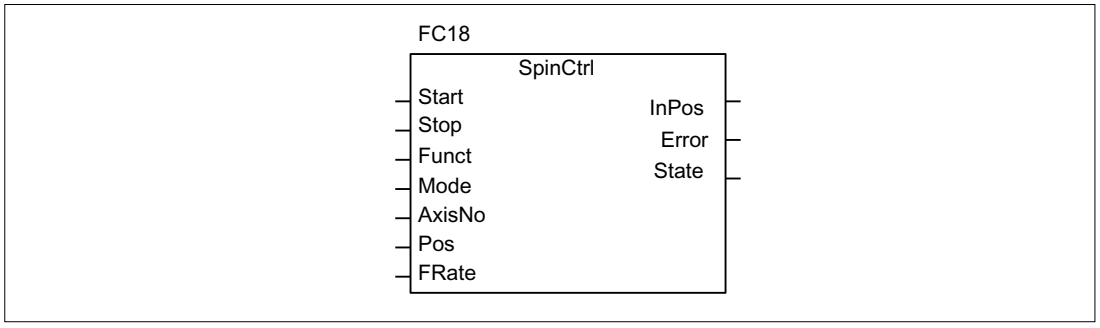
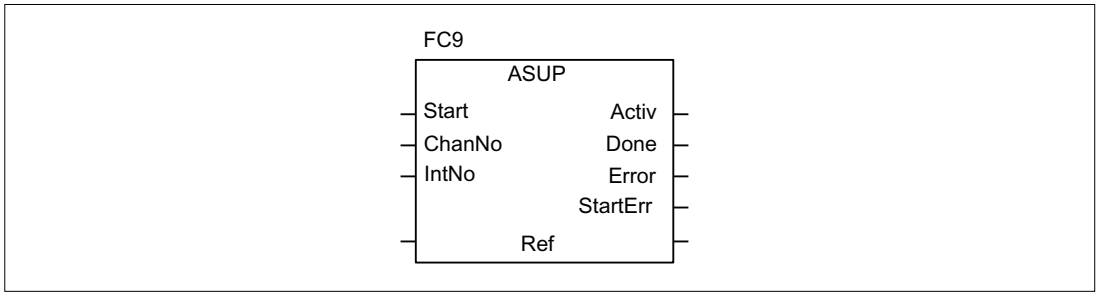


図 14-11　FC18 入力/出力パラメータ

非同期サブプログラム(ASUB)

ASUP を使用すると、NC で任意の機能を起動することができます。非同期サブプログラムを PLC から起動するには、そのプログラムを NC プログラムまたは FB4 PI サービス(ASUB)によって、使用可能にし、準備してください。

この方法で準備されていれば、PLC からいつでも起動できます。パラメータ設定された FC9 のチャネルのいずれかで実行されている NC プログラムは、非同期サブプログラムによって中断されます。ASUB は、起動パラメータを 1 に設定して、ユーザープログラムから FC9 を呼び出すことで起動します。



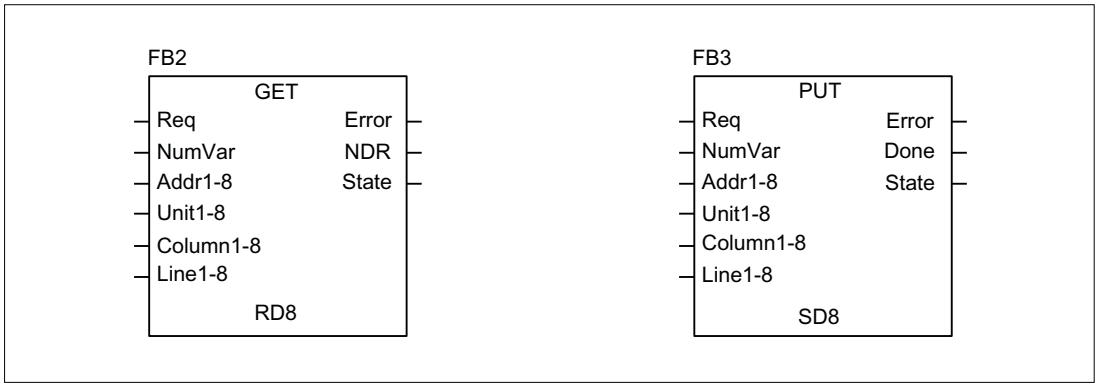
注記

非同期サブプログラムが NC プログラムまたは FB4 (ASUB)によって準備されていない場合(割り込み番号が割り当てられていない場合など)は、起動異常が出力されます(StartErr = TRUE)。

NC 変数の読み取り/書き込み

NC 変数は FB GET で読み取ることができ、FB PUT で NC 変数に値を入力できます。NC 変数は入力 Addr1 から Addr8 までの識別子を介してアドレス指定されます。これらの識別子(シンボル)はアドレスデータを指し、そのアドレスデータはグローバル DB に格納される必要があります。この DB の生成を可能にするため、付属の表から必須変数を選択できる PC ソフトウェア(NC Var セレクタ)が基本プログラムとともに提供されています。選択された変数は、まず第 2 のプロジェクト関連リストに収集されます。命令 **Generate DB** によって「*.AWL」ファイルが作成されます。このファイルを、関与する機械のプログラムファイルにリンクし、機械プログラムと一緒にコンパイルしてください。

1 つの読み取りまたは書き込みジョブで、1 ～ 8 個の値の読み取りまたは書き込みができます。必要に応じて、値が変換されます(NC 浮動小数点値(64 ビット)の PLC 形式(24 ビットの仮数と 8 ビットの指数から成る 32 ビット)への変換や、その逆など)。64 ビットから 32 ビット REAL に変換すると、精度の低下が発生します。32 ビット REAL 数値の最大精度は、約 10 の 7 乗です。



AG_SEND/AG_RECV ファンクション

AG_SEND/AG_RECV ファンクションは、STEP 7 の S7-300 CPU のライブラリ「SIMATIC_NET_CP」のファンクションに対応します。一般に、これらのファンクションには、オンラインヘルプが有効です。
AG_SEND/AG_RECV ファンクションは、内蔵される「CP 840D sl」を介した別のステーションとのデータ交信に使用できます。ファンクションの説明は、「ブロックの説明 (ページ 1162)」の章を参照してください。

注記

BSEND、USEND など、CP343-1 を処理するその他の通信ブロックは、SINUMERIK 840D sl ではサポートされません。

14.10.8 インタフェース DB によるユーザープログラムのシンボルプログラミング

概要

注記

840D のツールボックスに付属する CD の基本プログラムライブラリには、ファイル NST_UDTB.AWL および TM_UDTB.AWL が含まれます。

これら 2 つのファイルからコンパイルされた UDT ブロックが、基本プログラムの CPU プログラムに格納されます。

UDT は、たとえば CPU で生成されるデータブロックに割り当てることができる、ユーザーによって定義されるデータタイプです。

これらの UDT ブロックで、ほとんどのインタフェース信号のシンボル名が定義されます。

UDT 番号 2、10、11、19、21、31、71、72、73、77、1002、1071、1072、1073 が使用されます。

割り当ては次のように行われています。

UDT 割り当て		
UDT 番号	**インタフェース DB への割り当て**	**意味**
UDT2	DB2	アラーム/メッセージ
UDT10	DB10	NCK 信号
UDT11	DB11	モードグループ信号
UDT19	DB19	HMI 信号
UDT21	DB21 ～ DB30	チャネル信号
UDT31	DB31 ～ 61	軸/主軸信号
UDT71	DB71	工具管理機能:ロード/アンロード位置
UDT72	DB72	工具管理機能:主軸での変更
UDT73	DB73	工具管理機能:リボルバでの変更
UDT77	DB77	標準 SDB 210 による MCP および HHU 信号
UDT1002	DB2	拡張アラーム/メッセージ(FB1 パラメータ「ExtendAlMsg:=TRUE」)

UDT 割り当て		
UDT 番号	**インタフェース DB への割り当て**	**意味**
UDT 1071	DB 1071	工具管理機能:ロード/アンロードポイント(マルチツール)
UDT 1072	DB 1072	工具管理機能:主軸での変更(マルチツール)
UDT 1073	DB 1073	工具管理機能:タレットでの変更(マルチツール)

インタフェース信号をシンボルを使用してプログラムするには、最初にシンボルエディタを使用してインタフェースデータブロックをシンボルを使用して割り当ててください。

たとえば、シンボル「AxisX」は、シンボルファイルのデータタイプ UDT31 でオペランド DB31 に割り当てられます。

この入力の後、このインタフェースに対して STEP 7 プログラムをシンボルでプログラムできます。

注記

前述のインタフェース DB を使用する旧ソフトウェアバージョンで生成されたプログラムも、シンボルプログラムに変換できます。そして、データアクセス(たとえば"U DB31.DBX60.0"など) (主軸/回転軸)の完全修飾命令が以前に作成したプログラムで必要になります。この命令は、エディタ「AxisX.E_SpKA」でシンボルが起動されたとき変換されます。

説明

インタフェース信号の短縮シンボル名が、2 つの STL ファイル NST_UDTB.AWL および TM_UDTB.AWL で定義されます。

インタフェース信号の名前への参照を作成するため、各信号の後ろのコメントに名前が含まれます。

それらの名前は英語に基づきます。コメントは英語です。

シンボル名、命令および絶対アドレスは、UDT ブロックが開かれたとき STEP 7 エディタ命令によって表示できます。

注記

未使用のビットおよびバイトが、「f56_3」の指定などによってリストされます。

- "56":関連データブロックのバイトアドレス
- "3":このバイトのビット番号

14.10.9 リストに応じた M デコード

機能説明

拡張されたアドレスを持つ 256 個までの M 機能を、「リストに応じた M デコード」機能を使用して基本プログラムからデコードできます。この機能は、FB1 パラメータ「ListMDecGrp」(デコード用 M グループの数)を使用して起動します。拡張アドレスと、信号リストに含まれる信号を持つ M 機能の割り当ては、デコーディングリストで定義されます。この目的のために、信号のグループ化もおこなわれます。

デコーディングリスト(DB75)

デコーディングリストのソースファイル(MDECLIST.AWL)は基本プログラムに付属しています。STL ソースがコンパイルされると、データブロック DB75 が作成されます。機能が起動する前に、デコーディングリスト(DB75)とそれに続く再起動を PLC に伝送してください。

M 機能は、デコーディングリストに含まれている場合はデコードされます。M 機能のデコード時には、対応する信号が、特定のグループに応じて信号リストに設定されます。信号リストでの信号の設定時に、インターフェース信号「読み込み停止」が基本プログラムによって、 NC の関連チャネルで設定されます。このインターフェース信号は、ユーザーが、信号リストのこのチャネルからの信号出力をすべてリセットするとすぐに(つまり、確認された直後に)、チャネルに再度リセットされます。

信号リスト(DB76)

データブロック DB76 での機能の起動時に、基本プログラムは信号リストを作成します。それ以降、リストに従ってデコードされる各 M 信号に対して、1 つの信号が、対応するグループの信号リスト(DB76)に設定されます。同時に、その M 機能が出力されたチャネルで「読み込み停止」が設定されます。このインターフェース信号は、ユーザーが、信号リストのこのチャネルからの信号出力をすべてリセットするとすぐに(つまり、確認された直後に)、チャネルに再度リセットされます。

高速補助機能

デコーディングリストに含まれる M 機能が「高速補助機能」として出力される場合は、NC の対応するチャネルに読み込み停止が設定されません。

リストに従った M デコーディングの構造を次の図に示します。

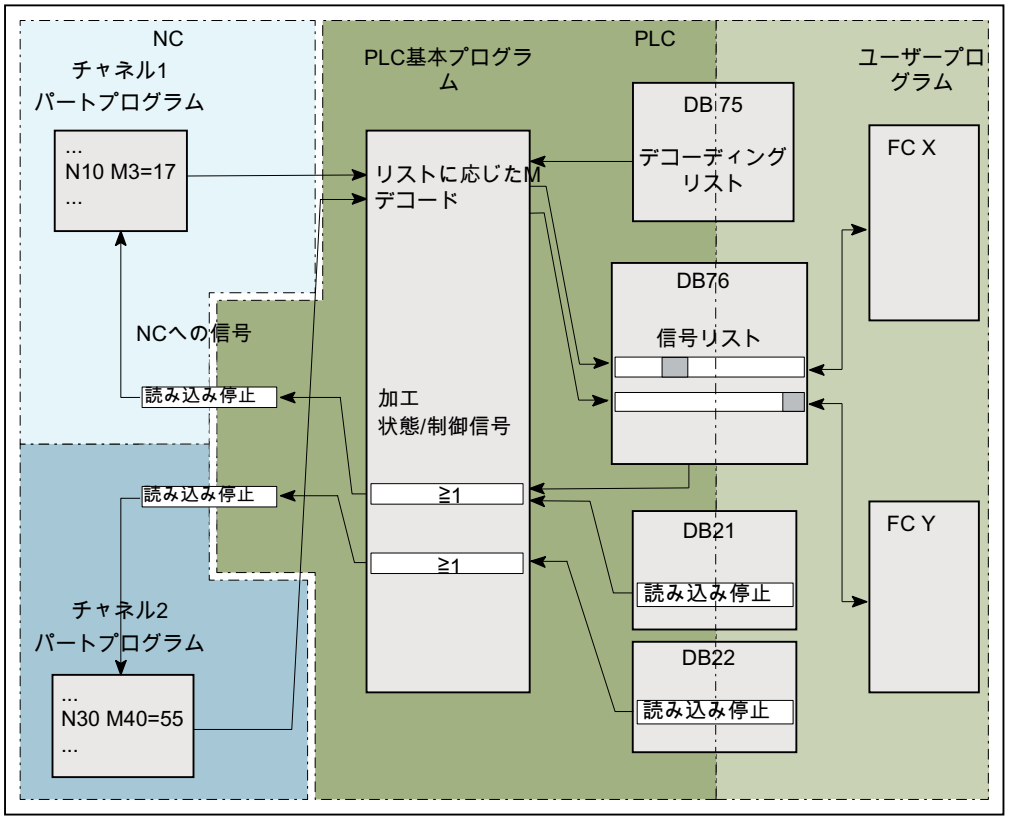


図 14-12　リストに応じた M デコード

起動

M デコーディングは、FB1 パラメータ「**ListMDecGrp**」を使用して起動します。

評価やデコードがおこなわれる M グループの数は、該当するパラメータを使用して指定します。パラメータ値 = 1 ... 16 の場合は、機能が有効です。

- 基本プログラム、OB100、FB1 パラメータ **ListMDecGrp** = <M グループの数>
  (「FB1:RUN_UP - 基本プログラム、起動セクション (ページ 1162)」も参照してください)。

デコーディングリスト(DB75)の特性と構造

デコーディングリスト(DB75)の**特性**:

- チャネルに依存しないデコーディングリストは 1 つだけです。
- デコーディングリストには、最大 16 のグループを含めることができます。
- グループには、最大 16 個の信号が含まれます。
- デコーディングリストに、デコードされる M 機能のすべてのグループのエントリが含まれるようにしてください。
- 拡張アドレスのある M 機能と、信号リストで設定されるビット間の割り当ては、関連するグループの最初と最後の M 機能を使用してデコーディングリストで指定されます。
  - 最初の M 機能:パラメータ:「MFirstAdr」≙ 信号またはビット 0
  - 最後の M 機能:パラメータ:「MLastAdr」≙最初の M 機能の最大の信号またはビット 15 との差に依存

デコーディングリスト(DB75)の**構造**:

デコーディングリストのエントリは、3 つのパラメータで構成され、各パラメータは 1 つのグループに割り当てられます。

グループ	拡張 M アドレス	グループの最初の M アドレス	グループの最後の M アドレス
1	MSigGrp[1].MExtAdr	MSigGrp[1].MFirstAdr	MSigGrp[1].MLastAdr
2	MSigGrp[2].MExtAdr	MSigGrp[2].MFirstAdr	MSigGrp[2].MLastAdr
...	...	...	...
16	MSigGrp[16].MExtAdr	MSigGrp[16].MFirstAdr	MSigGrp[16].MLastAdr

信号のタイプと値の範囲:

信号	タイプ	値の範囲	意味
MExtAdr	INT	0 ... 99	拡張 M アドレス
MFirstAdr	DINT	0 ～ 99.999.999	グループ内の最初の M アドレス
MLastAdr	DINT	0 ～ 99.999.999	グループ内の最後の M アドレス

信号リスト(DB76)の特性

信号リスト(DB76)には、以下の特性があります。

- チャネルに依存しない信号リストは 1 つだけです。
- 信号リストには、各 M グループに対して最大 16 個の信号を含めることができます。

例

M 機能の 3 つのグループが、以下のようにデコードされます。

- グループ 1:M2 = 1 ～ M2 = 5
- グループ 2:M3 = 12 ～ M3 = 23
- グループ 3:M40 = 55

デコーディングリストと信号リストの構造

グループ	デコーディングリスト(DB75)			信号リスト(DB76)
	拡張 M アドレス	グループの最初の M アドレス	グループの最後の M アドレス	
1	2	1	5	DB76.DBX0.0 ... DBX0.4
2	3	12	23	DB76.DBX2.0 ... DBX3.3
3	40	55	55	DB76.DBX4.0

プログラムコード

```
DATA_BLOCK DB 75
TITLE =
VERSION :0.0
   STRUCT
      MSigGrp :ARRAY [1 ..16 ] OF STRUCT
         MExtAdr :INT;
         MFirstAdr :DINT;
         MLastAdr :DINT;
      END_STRUCT;
BEGIN
   MSigGrp[1].MExtAdr   :=    2; 最初のグループの拡張 M アドレス
   MSigGrp[1].MFirstAdr :=  L#1; グループの最初の M アドレス
   MSigGrp[1].MFirstAdr :=  L#5; グループの最後の M アドレス
   MSigGrp[2].MExtAdr   :=    3; 2 番目のグループの拡張 M アドレス
```

プログラムコード

```
   MSigGrp[2].MFirstAdr :=  L#12; グループの最初の M アドレス
   MSigGrp[2].MFirstAdr :=  L#23; グループの最後の M アドレス
   MSigGrp[3].MExtAdr   :=    40; 3 番目のグループの拡張 M アドレス
   MSigGrp[3].MFirstAdr :=  L#55; グループの最初の M アドレス
   MSigGrp[3].MFirstAdr :=  L#55; グループの最後の M アドレス
END_DATA_BLOCK
```

OB100 の FB1 の構造

ファンクションを起動するには、デコードする M グループの数のパラメータ「ListMDecGrp」を入力します。

```
Call FB 1, DB 7(
   ...
   ListMDecGrp := 3;      //3 つのグループの M デコーディング
   ...
);
```

説明

OB100 に入力し、デコーディングリスト(DB75)を PLC に伝送した後、再起動を実行してください。再起動すると、基本プログラムにより信号リスト(DB76)が作成されます。

そして、NC プログラムが、最初のチャネルなどで起動します。拡張 M 機能が、この(M3=17)に含まれます。M 機能(M3 ≙グループ 2)のデコード時に、関連する信号(DBW1.5)が信号リスト(DB76)に設定され、最初のチャネルにインターフェース信号「読み込み停止」が設定されます。NC プログラムの実行が停止します。さらに、「拡張アドレス M 機能」と「M 機能番号」が、最初のチャネルのチャネル DB に表示されます。

信号リスト(DB76)でユーザーが、このチャネルからの信号出力をすべてリセットし、それらを確認した後は、最初のチャネルの「読み込み停止」信号がリセットされます。

14.10.10 PLC マシンデータ

一般事項

PLC 用のマシンデータを NC に設定するというオプションがあります。これで、これらのマシンデータを、PLC (OB100)の電源投入時に処理できます。これによって、ユーザーオプション、機械拡張レベル、機械構成などを実装できます。

このデータを読み取るためのインターフェースは DB20 にあります。ただし、DB20 は、ユーザーマシンデータが使用された(つまり、GP パラメータ「UDInt」、「UDHex」、およ

び「UDReal」の合計がゼロより大きい)場合、電源投入時に基本プログラムによってのみ作成されます。

データエリアのサイズ

個別の領域のサイズと、したがってそれに応じた DB20 の全長は、次の PLC マシンデータによって設定されます。

- MD14504 $MN_MAXNUM_USER_DATA_INT
- MD14506 $MN_MAXNUM_USER_DATA_HEX
- MD14508 $MN_MAXNUM_USER_DATA_FLOAT

ユーザー関連の PLC 基本プログラムパラメータ

このマシンデータ設定は、以下の PLC 基本プログラムパラメータによってユーザーが利用できます:

- "UDInt"
- "UDHex"
- "UDReal"

データ記憶

データは、以下の手順で PLC 基本プログラムによって DB20 にシームレスに格納されます:

1. INT 値
2. HEX 値(ビット配列)
3. FLOAT 値

INT と FLOAT の値は S7 フォーマットで保存されます。

16 進数値は、入力された順序で DB20 に格納されます(ビット配列として使用)。

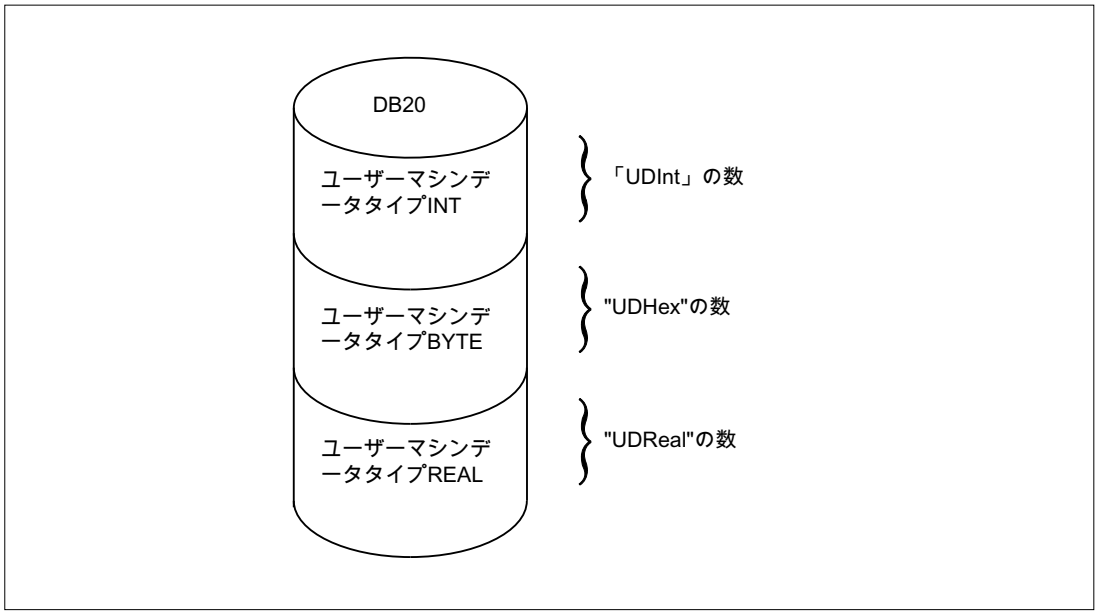


図 14-13　DB20 のユーザーデータ

注記

使用される PLC マシンデータの数を後で増やす場合は、その前に DB20 を削除してください。そのような拡張の使用によって既存のユーザープログラムが影響を受けないように、可能な限り UDT での構造定義などによるシンボル形式で DB20 のデータにアクセスしてください。

例

この例のプロジェクトでは、4 つの INT 値、ビット情報に 2 つの HEX 値、および 1 つの FLOAT 値が必要です。

マシンデータ:

MD14510 $MN_USER_DATA_INT[0] = 123

MD14510 $MN_USER_DATA_INT[1] = 456

MD14510 $MN_USER_DATA_INT[2] = 789

MD14510 $MN_USER_DATA_INT[3] = 1011

...

MD14512 $MN_USER_DATA_HEX[0] 12

=

MD14512 $MN_USER_DATA_HEX[1] AC
=

...

MD14514 123.456
$MN_USER_DATA_FLOAT[0] =

PLC 基本プログラムパラメータ
(OB100):

```
CALL FB1, DB7(
          MCPNum :=          1,
          MCP1In :=          P#E0.0,
          MCP1Out :=         P#A0.0,
          MCP1StatSend :=    P#A8.0,
          MCP1StatRec :=     P#A12.0,
          MCP1BusAdr :=      6,
          MCP1Timeout :=     S5T#700MS,
          MCP1Cycl :=        S5T#200MS,
          NCCyclTimeout :=   S5T#200MS,
          NCRunupTimeout :=  S5T#50S;
```

PLC 基本プログラムパラメータ(ランタ
イムにスキャン):

```
          l gp_par.UDInt;     //=4,
          l gp_par.UDHex;     //=2,
          l gp_par.UDReal;    //=1 )
```

PLC の電源投入時に、DB20 が長さ 28 バイトで生成されました。

DB20	
アドレス	データ
0.0	123
2.0	456
4.0	789
6.0	1011
8.0	b#16#12

DB20	
アドレス	データ
9.0	b#16#AC
10.0	1.234560e+02

使用されるマシンデータの構造は、UDT で指定されます。

```
TYPE UDT20
   STRUCT
      UDInt :    ARRAY [0 ..3] OF INT;
      UDHex0 :   ARRAY [0 ..15]OF BOOL;
      UDReal :   ARRAY [0 ..0] OF REAL;      //フィールドとしての説明、
                                             後の拡張に対応
   END_STRUCT;
END_TYPE
```

注記

ARRAY OF BOOL は常に偶数アドレスに送信されます。そのため、通常は UDT 定義で0 ～ 15 の配列範囲を選択するか、すべてのブール変数を個別に指定してください。

この例では最初は REAL 値のみが使用されますが、変数用にフィールド(1 つの要素を持つ)が作成されています。これによって将来、シンボルアドレスの変更なしで簡単に拡張できます。

シンボルアクセス

シンボル形式でのデータアクセスを可能にするエントリがシンボル表に作成されます。

シンボル	オペランド	データタイプ
UData	DB20	UDT20

ユーザープログラムでのアクセス操作(シンボル形式での読み取りアクセスのみがリストに含まれます)

```
...
        L      "UData".UDInt[0];
        L      "UData".UDInt[1];
        L      "UData".UDInt[2];
        L      "UData".UDInt[3];
```

```
            U       "UData".UDHex0[0];
            U       "UData".UDHex0[1];
            U       "UData".UDHex0[2];
            U       "UData".UDHex0[3];
            U       "UData".UDHex0[4];
            U       "UData".UDHex0[5];
            U       "UData".UDHex0[6];
            U       "UData".UDHex0[7];
            ...     ...
            U       "UData".UDHex0[15];

            L       "UData".UDReal[0];
...
```

14.10.11 機械操作パネル、ハンドヘルドユニット、ダイレクトキーの構成

一般事項

最大で 2 つの機械操作パネルと 1 つのハンドヘルドユニットを同時に操作できます。機械操作パネル(MCP)とハンドヘルドユニット(HHU)にはさまざまな接続オプション(Ethernet/PROFINET、PROFIBUS)があります。2 個の MCP を異なるバスシステムに接続できます(複合操作は、Ethernet と PROFIBUS 上でのみ可能です)。そのためには、FC1 パラメータ「MCPBusType」を使用します。このパラメータでは、右側の桁(1 の位)が第 1 MCP に関与し、左側の桁(10 の位)が第 2 MCP に関与します。

コンポーネントのパラメータ設定は常に、OB100 の基本プログラムブロック FC1 を呼び出して実行します。FC1 は、そのパラメータを関連するインスタンスデータブロック(DB7、シンボル表記の「LBP_ConfigData」)に保存します。それぞれの機械操作パネルとハンドヘルドユニットに対して、別々のパラメータセットが提供されます。ユーザーの入出力アドレスをこれらのパラメータセットにおいて定義してください。これらの入力アドレスと出力アドレスは、FC19、FC24、FC25、FC26、および FC13 でも使用します。さらに、状態情報、PROFIBUS または Ethernet/PROFINET のアドレスも定義してください。タイムアウトと周期的強制再トリガの初期時間設定は、変更すべきではありません。MCP および HHU コンポーネントについて詳しくは、オペレータコンポーネントマニュアルを参照してください。

起動

各コンポーネントは、機械操作パネルの数(「MCPNum」パラメータ)を介してまたは、ハンドヘルドユニットの場合、「BHG」パラメータを介しても有効になります。MCP と HHU の接続設定は、FC1 パラメータ「MCPMPI」、「MCPBusType」または「BHG」、「BHGMPI」に入力します。

ハンドヘルドユニット(HT 2)

ハンドヘルドユニットでは、アドレス指定は、GD パラメータセットのパラメータを介しておこないます。これは、パラメータ名称の互換性のために必要でした。

設定

基本的に、MCP/HHU と PLC 間のデータには、さまざまな通信メカニズムがあります。こうしたメカニズムは、MCP と HHU のバス接続によって決まります。一つの場合 (Ethernet) には、データは「CP 840D sl」を介して伝送されます。

このパラメータ設定は、FC1 の MCP/HHU パラメータを介してすべて実行されます。

他の場合には、伝送は、PROFIBUS コンフィグレーションにより PLC オペレーティングシステムを介して行われます。

パラメータ設定は、HW-Config で STEP 7 を介しておこないます。基本プログラムが MCP/HHU のこうしたデータと故障監視にアクセスできるようにするには、FC1 パラメータ内で設定されているアドレスを基本プログラムで使用できるようにしてください。

さまざまな接続メカニズムの概要を次に示します。複合操作も構成できます。

タイムアウトによって異常が検出されると、PLC CPU のアラームバッファ内に入力されます(アラーム 400260 から 400262 まで)。この場合、MCP またはハンドヘルドユニット(MCP1In/MCP2In または BHGIn)からの入力信号は 0 にリセットされます。PLC と MCP/HHU の再原点同期が可能な場合は、通信が自動的に再開され、エラーメッセージが GP によって削除されます。

注記

下の表の略語「(n.r.)」は「関連なし」を意味しています。

Ethernet 接続(MCPBusType = 5)

それ以上の構成設定がおこなわれない場合、通信は、CP 840D sl を介して PLC BP から直接おこなわれます。以下のリストに示す FC1 パラメータが、パラメータ設定に使用されます。

コンポーネントの論理名称の数値部は、「MCP1 BusAdr」、「MCP2 BusAdr」または「BHGRecGDNo」に入力してください(ノードのバスアドレスに対応します)。論理名称は、MCP または端子箱のスイッチを介して定義します。

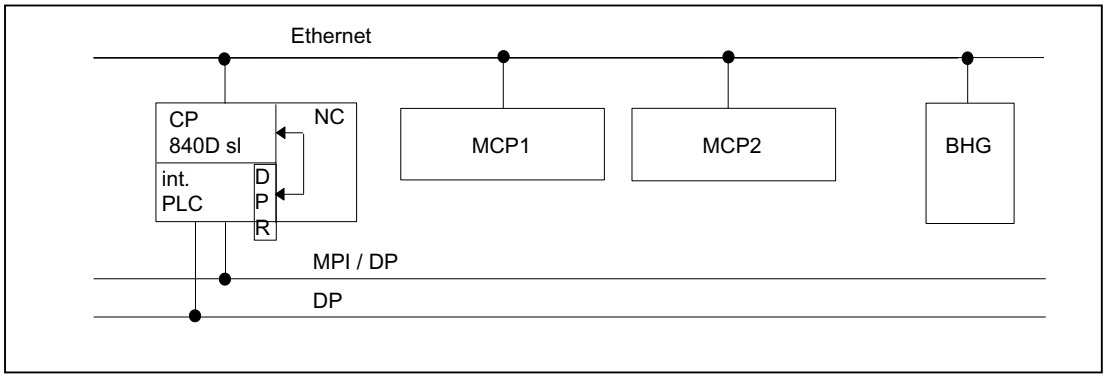


図 14-14　Ethernet 接続

関連パラメータ(FB1)		
MCP		**HHU**
MCPNum = 1 または 2 (MCP の数)		HHU = 5 (CP 840D sl を介して)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend	MCP2StatSend	BHGStatSend
MCP1StatRec (n.r.)	MCP2StatRec (n.r.)	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen (n.r.)
MCP1Timeout (n.r.)	MCP2Timeout (n.r.)	BHGOutLen (n.r.)
MCP1Cycl (n.r.)	MCP2Cycl (n.r.)	BHGTimeout (n.r.)
MCPMPI = FALSE		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
MCP1NotSend	MCP2NotSend	BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPBusType = b#16#55 (CP 840D sl を介して)		BHGSendGDNo (n.r.)
		BHGSendGBZNo (n.r.)

関連パラメータ(FB1)	
MCP	**HHU**
MCPSDB210= FALSE	BHGSendObjNo (n.r.)
MCPCopyDB77 = FALSE	**BHGMPI = FALSE**
	BHGStop
	BHG NotSend

タイムアウト用の PLC アラームバッファにも、エラー入力がおこなわれます。その結果、以下のエラーメッセージが HMI に出力されます。

- 400260:MCP 1 故障
  または
- 400261:MCP 2 故障
- 400262:HHU 故障

MCP/HHU と PLC との間で、まだデータ交信がおこなわれていなくても、コールド再始動後に、MCP または HHU の故障は即時に検出されます。
電源投入後、すべてのコンポーネントが「準備完了」信号を出すとすぐに、監視機能が有効になります。

例:ダイレクトキーを持つ OP

Ethernet バスにある OP のダイレクトキーは、PLC に伝送すべきです。以前は、ダイレクトキーは、PROFIBUS を介して、あるいは、OP と MCP 間の特殊なケーブル接続を介して PLC に伝送されてきました。

Ethernet を介してダイレクトキーを接続する場合、これは、「OP 08T」などに関連しており、データ伝送を起動するためのパラメータ設定が基本プログラム内でおこなわれます。関連するパラメータは、FC1 のインスタンス DB 内にあります(OpKeyNum から OpKeyBusType まで。データ表を参照してください)。このパラメータは、開始 OB100 において、ユーザーが FC1 呼び出しでパラメータを接続して指定します。バスアドレスと Op1/2KeyStop は、FC1 インスタンス DB DB7 を書き込むことによって、サイクリック運転で変更することもできます。

ダイレクトキーのユーザーデータの伝送は、Ethernet MCP の場合と同じように実行されます。データ伝送は、DB7 パラメータ「Op1/2KeyStop」を書き込むことによって、停止と再開をおこなうこともできます。停止状態のとき、ダイレクトキーモジュールのアドレス(TCU インデックスまたは MCP アドレス)を変更することもできます。

停止信号をリセットした後、新しいアドレスへの接続が確立されます。

それぞれのダイレクトキーインタフェースの状態は、次のインタフェース信号で読み取ることができます。

DB10.DBX104.3 (OP1Key 準備完了)

または

DB10.DBX104.4 (OP2Key 準備完了)

アドレスダイレクトキー

Op1/2KeyBusAdr パラメータの場合、通常は TCU インデックスを使用してください。これは、ダイレクトキーについて、Ethernet MCP への特殊なケーブル接続を**おこなわない** OP (OP 08T、OP 12T など)に影響を与えます。

ダイレクトキーを持つ OP が特殊なケーブル接続をおこない、これらが Ethernet-MCP に接続される場合は、Op1/2KeyBusAdr パラメータについて、MCP のアドレス(MCP の DIP スイッチ設定)を使用してください。ダイレクトキーのデータストリーム(2 バイト)のみがダイレクトキーインタフェースを介して伝送されます。

アラームダイレクトキー

タイムアウト用の PLC アラームバッファにも、エラー入力がおこなわれます。その結果、以下のエラーメッセージが HMI に出力されます。

- 400274:ダイレクトキー 1 で故障が発生しました
  または
- 400275:ダイレクトキー 2 で故障が発生しました

コントロールユニットのダイレクトキーの切り替え

ユーザーが、0xFF で Op1/2KeyBusAdr を接続し、開始ブロック OB100 で= TRUE を停止します。M 対 N インターフェースのダイレクトキーアドレスは、M 対 N ブロック FB9 を介してパラメータ「Op1KeyBusAdr」に接続します。

関連パラメータ(FC1)		
ダイレクトキー		**例: ダイレクトキー OP 08T**
OpKeyNum = 1 または 2 (ダイレクトキーを持 OP の数)		
Op1KeyIn	Op2KeyIn	
Op1KeyOut	Op2KeyOut	
OpKey1BusAdr	Op2KeyBusAdr	アドレス:TCU インデックス:
Op1KeyStop	Op2KeyStop	
Op1KeyNotSend	Op2KeyNotSend	

関連パラメータ(FC1)	
ダイレクトキー	例: ダイレクトキー OP 08T
OpKeyBusType = b#16#55 (CP 840D sl を介して)	

MCP の識別

DB7 の識別インターフェースを介して、Ethernet コンポーネントのタイプ(MCP、HT 2、HT 8、またはダイレクトキー)を、サイクリック運転中に入出力時の関連パラメータとともに確認することができます。

- 入力時の関連パラメータは次のとおりです。
  「IdentMcpBusAdr」、「IdentMcpProfilNo」、「IdentMcpBusType」、「IdentMcpStrobe」
- 出力時の関連パラメータは次のとおりです。
  「IdentMcpType」、「IdentMcpLengthIn」、「IdentMcpLengthOut」

パラメータ「IdentMcpBusAdr」にある DIP デバイスアドレスまたは TCU インデックスは、ストローブ信号の設定とともにユーザープログラムによって起動されます。

入力パラメータ「IdentMcpProfilNo」は、通常、値 0 に設定されます。このパラメータは、ダイレクトキーを識別する場合にのみ、値 1 に設定します。パラメータ「IdentMcpBusType」は現在、ユーザープログラムにとって意味を持たないため、その初期値のままにしてください。

基本プログラムによるストローブ信号のリセット後に、有効な出力情報をユーザーが使用できるようになります。基本プログラムによるストローブ信号のリセットは、複数の PLC 周期の間(最大 2 秒)続くことがあります。

出力パラメータは、指定された機器のデータ領域のサイズをユーザーに示します。さらに、ここで端子箱に HT 2 または HT 8 のどちらが接続されるか、あるいは、機器が接続されないかを定義できます。この情報を使用して、MCP チャネルまたは HHU チャネルを起動できます。サイクリック運転では、ユーザープログラムによって DB7 のシンボル名称(LBP_ConfigData)を介してシンボル表記でパラメータを書き込み、読み取ることができます。

関連パラメータ(FB1)		
MCP 機器の識別		入力パラメータ、例:OP 08T
入力	出力	ダイレクトキーの値

関連パラメータ(FB1)		
MCP 機器の識別		**入力パラメータ、例:OP 08T**
IdentMcpBusAdr	IdentMcpType	IdentMcpBusAdr = TCU インデックス
IdentMcpBusProfilNo	IdentMcpLengthIn	IdentMcpBusProfilNo = 値 1
IdentMcpBusType	IdentMcpLengthOut	IdentMcpBusType = 初期値
IdentMcpStrobe		
IdentMcpBusProfilNo		**値**
MCP、HHU、HT 8、HT 2		B#16#0
例: ダイレクトキー OP 08T、OP 12T		B#16#1
IdentMcpType (Mcp タイプ)		
接続されている機器なし		0
MCP 483C IE (コンパクト)		B#16#80
MCP 483C IE		B#16#81
MCP 310		B#16#82
MCP OEM		B#16#83
MCP DMG		B#16#84
HT 8		B#16#85
TCU_DT (ダイレクトキー)		B#16#86
MCP_MPP		B#16#87
HT 2		B#16#88
OP 08T (ダイレクトキー)		B#16#89

DP ポートでの PROFIBUS 接続(MCPBusType = 3)

MCP の PROFIBUS 接続の場合は、このコンポーネントを STEP 7 のハードウェアコンフィグレーション設定で考慮してください。MCP は、PLC (**MPI/DP ではなく**)の標準 DP バスに接続します。アドレスは、入出力マッピングエリアに保存してください。こうした開始アドレスは、FC1 のポインタパラメータにも保存してください。以下のリストに示す FC1 パラメータが、その他のパラメータ設定に使用されます。

HHU の PROFIBUS タイプはありません。そのため、この図では、HHU について Ethernet 接続を示します。PROFIBUS スレーブアドレスは、パラメータ「MCP1BusAdr」と「MCP2BusAdr」に保存してください。構成された診断アドレス(たとえば、P#A8190.0)へのポインタを「MCPxStatRec」に入力してください。

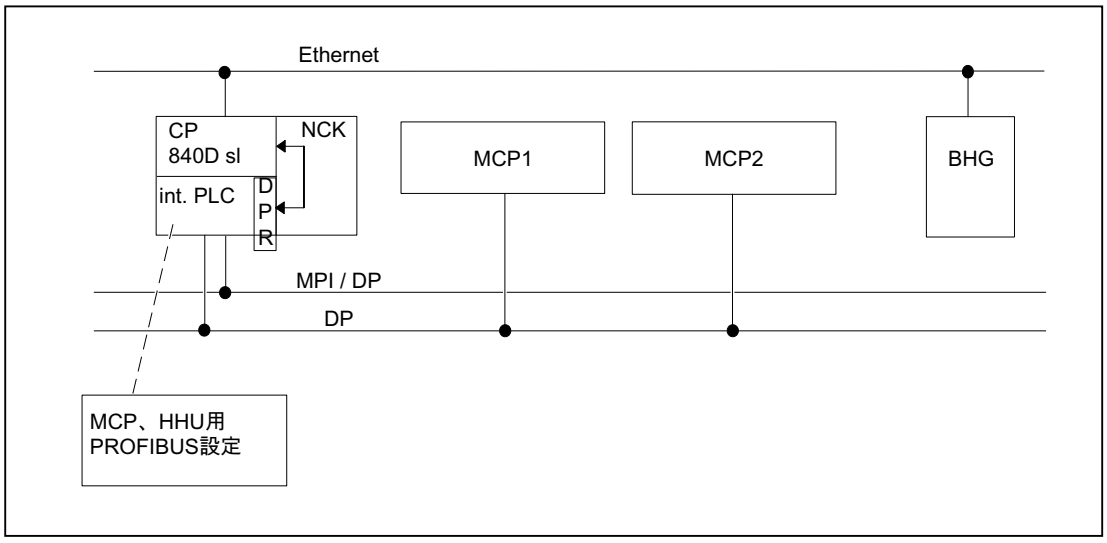


図 14-15　PROFIBUS 接続

関連パラメータ(FB1)		
MCP		**HHU**
MCPNum = 1 または 2 (MCP の数)		BHG = 5 (CP 840D sl を介して)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend (n.r.)	MCP2StatSend (n.r.)	BHGStatSend
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOutLen
MCP1Cycl (n.r.)	MCP2Cycl (n.r.)	BHGTimeout (n.r.)
MCPMPI = FALSE		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
MCPBusType = b#16#33		BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPSDB210= FALSE		BHGSendGDNo (n.r.)

関連パラメータ(FB1)		
MCP		**HHU**
MCPCopyDB77 = FALSE		BHGSendGBZNo (n.r.)
		BHGSendObjNo (n.r.)
		BHGMPI = FALSE
		BHGStop

MCP の故障が発生すると、通常、PLC が STOP 状態に切り替わります。これが望ましくない場合は、OB 82、OB 86 を使用して停止を回避できます。基本プログラムには、標準で、OB82 呼び出しと OB86 呼び出しがあります。FC5 は、これらの OB 内で呼び出されます。この FC5 が、故障したスレーブが MCP であるかどうかをチェックします。そうである場合、PLC 停止は実施されません。「MCPxStop」:= TRUE を設定すると、基本プログラムが SFC12 を介して、スレーブとしての MCP を解除します。MCP の故障またはエラーの後に PLC が停止状態に切り替わらない場合は、基本プログラムを介してアラームメッセージが生成されます。ステーションが回復すると、この割り込みは削除されます。

MPI/DP ポートでの PROFIBUS 接続(MCPBusType = 4)

MCP の PROFIBUS 接続をおこなう場合、このコンポーネントは STEP 7 ハードウェアコンフィグレーションで考慮してください。MCP は、PLC の MPI/DP バスで接続します。

アドレスは、入出力マッピングエリアに保存してください。こうした開始アドレスは、FC1 のポインタパラメータにも保存してください。以下のリストに示す FC1 パラメータが、その他のパラメータ設定に使用されます。HHU の PROFIBUS タイプはありません。そのため、この図では、HHU について Ethernet 接続を示します。PROFIBUS スレーブアドレスは、パラメータ MCP1BusAdr および MCP2BusAdr に保存する必要があります。MCPxStatRec の構成された診断アドレス(P#A8190.0)にポインタを入力します。

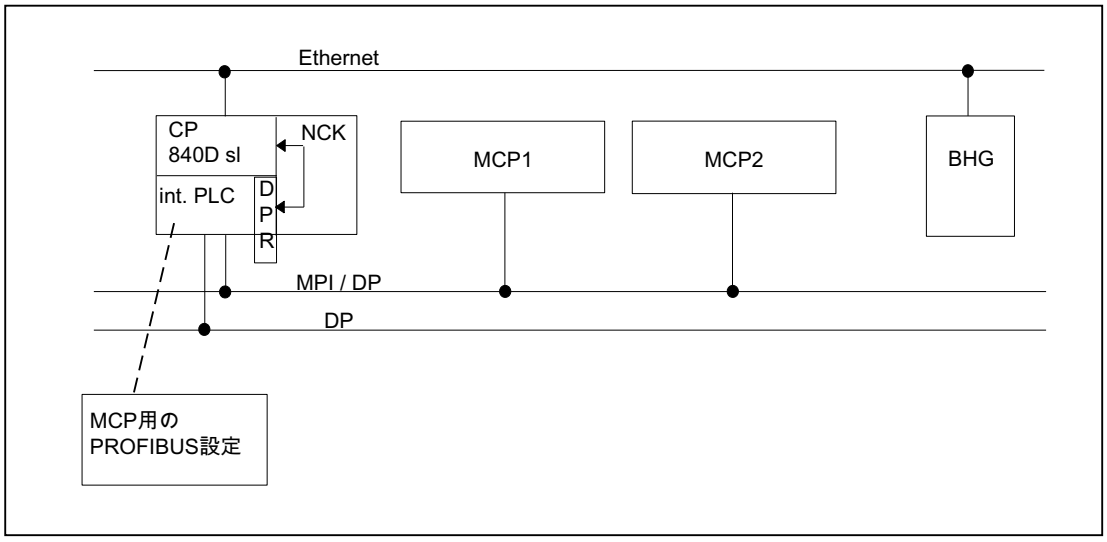


図 14-16　MPI/DP ポートでの PROFIBUS 接続

関連パラメータ(FB1)		
MCP		**HHU**
MCPNum = 1 または 2 (MCP の数)		HHU = 5 (CP 840D sl を介して)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend (n.r.)	MCP2StatSend (n.r.)	BHGStatSend
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOutLen
MCP1Cycl (n.r.)	MCP2Cycl (n.r.)	BHGTimeout (n.r.)
MCPMPI = FALSE		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
MCPBusType = b#16#44		BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPSDB210= FALSE		BHGSendGDNo (n.r.)
MCPCopyDB77 = FALSE		BHGSendGBZNo (n.r.)
		BHGSendObjNo (n.r.)
		BHGMPI = FALSE
		BHGStop

MCP の故障が発生すると、通常、PLC が STOP 状態に切り替わります。これが望ましくない場合は、OB82 と OB86 を使用して PLC 停止を回避できます。基本プログラムには、標準で、OB82 呼び出しと OB86 呼び出しがあります。FC5 は、これらの OB 内で呼び出されます。この FC5 が、故障したスレーブが MCP であるかどうかをチェックします。そうである場合、PLC 停止は実施されません。MCPxStop:= TRUE を設定すると、基本プログラムが SFC12 を介して、スレーブとしての MCP を解除します。MCP の故障またはエラーの後に PLC が停止状態に切り替わらない場合は、基本プログラムを介してアラームメッセージが生成されます。ステーションが回復すると、このアラームは削除されます。

PROFINET 接続(MCPBusType = 6)

MCP の PROFIBUS 接続の場合は、このコンポーネントを STEP 7 のハードウェアコンフィグレーション設定でパラメータ設定してください。MCP は、CPU の PROFINET モジュールと連結されます。

MCP をハードウェコンフィグレーションでパラメータ設定する時に、アドレスを入力と出力マッピングエリアに入れてください。こうした開始アドレスは、FC1 のポインタパラメータ(MCPxIn と MCPxOut) にも保存してください。これをおこなう理由は、このパラメータを介して信号が MCP と基本プログラム間でやりとりされるからです。MCP は、パラメータ MCPxIn を使用しても監視できます。このため、パラメータ MCPxBusAdr はこの MCP バリアントとは関連していません。

構成された診断アドレス(たとえば、P#A8190.0)へのポインタを MCPxStatRec に入力してください。

PROFINET MCP には独自のタイプがあり、それをパラメータ MCPBusType に適用してください。

以下のリストに示す FC1 パラメータが、その他のパラメータ設定に使用されます。HHU の PROFIBUS タイプはありません。HHU の Ethernet ポートを下の図に示します。

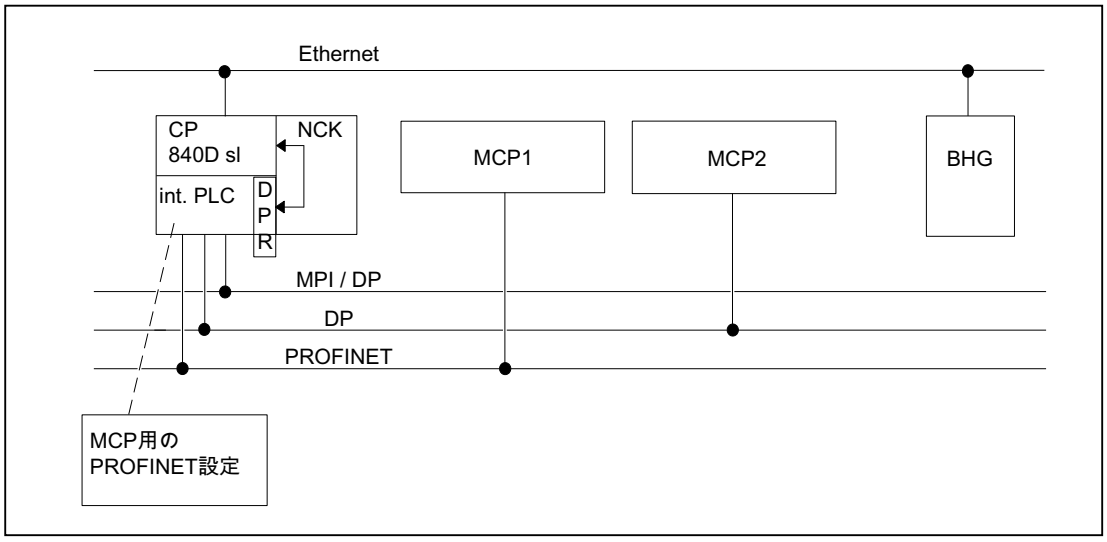


図 14-17　PROFINET 接続

関連パラメータ(FB1)		
MCP		**HHU**
MCPNum = 1 または 2 (MCP の数)		HHU = 5 (CP 840D sl を介して)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend (n.r.)	MCP2StatSend (n.r.)	BHGStatSend
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1BusAdr (n.r.)	MCP2BusAdr (n.r.)	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOutLen
MCP1Cycl	MCP2Cycl	BHGTimeout (n.r.)
MCPMPI = FALSE		BHGCycl (n.r.)
MCP1Stop	MCP2Stop	BHGRecGDNo
MCPBusType = b#16#36 (例として図のように) (6 = MCP1 用の PROFINET) (3 = MCP2 用の PROFIBUS)		BHGRecGBZNo (n.r.)
		BHGRecObjNo (n.r.)
MCPSDB210= FALSE		BHGSendGDNo (n.r.)
MCPCopyDB77 = FALSE		BHGSendGBZNo (n.r.)

関連パラメータ(FB1)		
MCP		HHU
		BHGSendObjNo (n.r.)
		BHGMPI = FALSE
		BHGStop

MCP の故障が発生すると、通常、PLC が STOP 状態に切り替わります。これが望ましくない場合は、OB82 と OB86 を使用して PLC 停止を回避できます。基本プログラムには、標準で、OB82 呼び出しと OB86 呼び出しがあります。FC5 は、これらの OB 内で呼び出されます。この FC5 が、故障したスレーブが MCP であるかどうかをチェックします。そうである場合、PLC 停止は実施されません。MCPxIn の故障の監視時に、パラメータ MCPxIn の入力アドレスは 意味を持ちます。

MCPxStop:= TRUE を設定すると、基本プログラムが SFC12 を介して、スレーブとしての MCP を解除します。MCP の故障またはエラーの後に PLC が停止状態に切り替わらない場合は、基本プログラムを介してアラームメッセージが生成されます。ステーションが回復すると、このアラームは削除されます。

14.10.12 機械操作パネル、ハンドヘルドユニットの切り替え

オペレータコンポーネント(MCP または HHU)の切り替え/解除を標準でサポートしているのは、Ethernet タイプのみです。

PROFIBUS タイプ

PROFIBUS タイプを使用すると、この機能が制限され、ユーザーによる追加の作業が必要になります。

MCP の PROFIBUS タイプを使用すると、MCP1、MCP2 または HHU に対して指定されている DB77 のデータ領域を FC1 上の MCP ポインタに使用できます。MCP スレーブのバスアドレスは、監視の基準として使用されるため、`MCPxBusAdr` に正しく設定してください。ユーザープログラムのコピールーチンで、HW Config で構成されている I/O 領域から DB77 に、有効な MCP の信号がコピーされなければなりません。これにより、PROFIBUS 上のいくつかの MCP を信号を介して切り替えることができます。ある MCP から別の MCP への切り替えフェーズの場合は、`MCPxStop` パラメータを TRUE に設定してください。

制御信号

パラメータ MCP1Stop、MCP2Stop、および HGStop は、個々のコンポーネントとの通信を停止します(パラメータ設定 = 1)。通信のこうした停止または起動は、現在の周期で適用できます。ただし、値の変更は、別の FC1 呼び出しを使用するのではなく、パラメータのシンボル表記を通して実行してください。

例:第 1 機械操作パネルからの伝送の停止

```
SET;
S LBP_ConfigData.MCP1Stop;
```

パラメータ MCP1Stop、MCP2Stop、HGStop を設定すると、アラーム 400260 ～ 400262 もマスクまたは削除されます。

バスアドレスの切り替え

オペレータコンポーネント(MCP または HHU)への既存の通信接続をキャンセルし、通信アドレスの異なる別のコンポーネント(MCP または HHU)への新しい通信接続を確立する場合は、以下の手順を実行します。

1. 接続を切断するオペレータコンポーネントの通信を停止します。パラメータ MCP1Stop、MCP2Stop、または HGStop = 1
2. 以下に該当する場合は、通信が停止します。DB10、DBX104.0、.1 または.2 == 0
3. バスアドレスを変更します。
   - MCP:FC1 パラメータ MCP1BusAdr または MCP2BusAdr = <新しいオペレータコンポーネントのバスアドレス>
   - HHU (Ethernet タイプ):FC1 パラメータ BHGRecGDNo = <新しいオペレータコンポーネントのバスアドレス>
4. 通信を有効にします(ポイント 3 と同じ PLC 周期で可能):パラメータ MCP1Stop、MCP2Stop、または HGStop = 0
5. 以下に該当する場合は、新しいコンポーネントとの通信が有効になります。DB10、DBX104.0、.1 または.2 == 1

Ethernet MCP の LED の点滅のオフ

電源投入の後、MCP では通常、電源投入が完了したことと、接続の確立を待機していることが、LED の点滅で示されます。LED の点滅は、以下に示す手順でオフにすることができます。現在、この動作は、MCP に記憶させることは**できません**。

必要条件

V02.02.04 以降の MCP ファームウェア

点滅をオフにする設定

MCP との通信開始前に、MCPxStop に送信状態を設定します。通信開始前とは、DB7 パラメータ MCPxStop = FALSE の設定前の電源投入(OB100)時またはサイクリック運転(OB1)時です。

送信状態の設定:FC1 パラメータ MCPxStatSend、ビット 30 = 0 およびビット 31 = 1

現在の状態のフィードバックはありません。

例

OB100 からの抽出: (MCP1 の例に基づいています)

```
CALL "RUN_UP" , "LBP_ConfigData"
   ...
   MCP1StatSend   := P#A 8.0
   ...
   // MCP の点滅を停止します
   SET
   R   A 11.6
   S   A 11.7
   ...
```

14.11 Safety Integrated 用の SPL

SPL は、基本プログラムの機能というよりユーザー機能です。基本プログラムは、データブロック(DB18)を安全 SPL 信号に対して使用可能にし、NC 内での SPL プログラムデータの適合性を確保するためにデータ比較を実行します。

参照先:
『/FBSI/機能説明書、Safety Integrated』

14.12 割り付け一覧

14.12.1 割り付け: NCK/PLC インタフェース

SINUMERK 840D sl 用の NC/PLC インタフェースの値については、次を参照してください。

参照先:
『データ一覧表 sl 第 2 巻』

14.12.2 割り付け:FB/FC

番号	意味
FB15	基本プログラム(Basic program)
FB1、FC2、FC3、FC5	基本プログラム(Basic program)
FC0 ... 29	当社用に予約済
FB0 ... 29	当社用に予約済
FC30 ... 999 [1)]	ユーザー割り当て用に空き
FB30 ... 999 [1)]	ユーザー割り当て用に空き
FC1000 ... 1023	当社用に予約済
FB1000 ... 1023	当社用に予約済
FC1024 ... 上限値	ユーザー割り当て用に空き
FB1024 ... 上限値	ユーザー割り当て用に空き

1) ブロック番号(FB/FC)の実際の上限は 選択された NCU に配置されている PLC CPU に依存します。

注記

FC、FB の値については、「PLC 基本プログラムの必要メモリ (ページ 1142)」を参照してください。

14.12.3 割り付け:DB

注記

データブロックは、NC マシンデータ設定により要求される分、準備されるだけです。

データブロックの概要			
DB 番号	名称	名称	パッケージ
1		当社用に予約済	BP
2 ... 5	PLC-MELD	PLC メッセージ	BP
6 ... 8		基本プログラム	

データブロックの概要			
DB 番号	名称	名称	パッケージ
9	NC コンパイル	NC コンパイルサイクル用インタフェース	BP
10	NC INTERFACE	セントラル NC インタフェース	BP
11	BAG1	モードグループインタフェース	BP
12		コンピュータリンクおよび伝送システムインタフェース	
13 ... 14		基本プログラム用に予約済	
15		基本プログラム	
16		PI サービス定義	
17		バージョン識別子	
18		基本プログラム用に予約済	
19		HMI インタフェース	
20		PLC マシンデータ	
21 ... 30	CHANNEL 1 ... n	NC チャネルのインタフェース	BP
31 ... 61	AXIS 1 ... m	軸/主軸のインタフェース またはユーザーのために解放	BP
62 ... 70		ユーザーのために解放	
71 ... 74		工具管理機能	BP
75 ... 76		M グループデコーディング	
77		MCP 信号用の DB	
78 ... 80		当社用に予約済	
81 ... 999 [1)]		下記を参照:ShopMill、ManualTurn	
1000 ... 1099		当社用に予約済	
1100 ... 上限		ユーザーのために解放	

[1)] ブロック番号(DB)の実際の上限は、選択された NCU に配置されている PLC CPU に依存します。起動されていないチャネル、軸/主軸、および工具管理のファンクションのデータブロックは使用可能です。

注記

起動されていないチャネル、軸/主軸および工具管理機能のデータブロックは、自由に使用することができます。

14.12.4 割り付け:タイマ

タイマ番号	内容
T 0 ... T 512 [1]	ユーザー領域

[1] タイマ番号(DB)の実際の上限は、選択された NCU に配置されている PLC CPU に依存します。

14.13 HMI 用の PLC 機能(DB19)

14.13.1 チャネルの選択

機能

機械の開始画面などの HMI に表示されるチャネルは、HMI/PLC インターフェースを介して PLC ユーザープログラムから選択できます。

必要条件

複数のチャネルが NC でパラメータ設定されます。

ジョブおよび確認のインターフェース

DB19	意味	
DBX32.0 - .5	PLC → HMI	ファンクション番号:1 =チャネル選択
DBX32.6	PLC → HMI	ファンクション要求
DBX32.7	HMI → PLC	状態:1 =「ファンクション実行中」

DB19	意味	
DBB33	PLC → HMI	チャネル番号:1、2、3、...チャネルの最大数 次のチャネル:FF_H
DBB36	HMI → PLC	エラー ID: ● 0:異常なし ● 1:無効なファンクション番号(DBX32.0 - .5) ● 2:無効なパラメータ(DBB33 - DBB35) ● 3:HMI 内部変数の書き込み時のエラー ● 10:チャネルが存在しない(DBB33)

機能の処理

PLC → HMI

PLC ユーザープログラムは、以下の実行シーケンスを維持しなければなりません。

1. インターフェースが新しいジョブのために空いているかどうかをチェックします。
   - DB19.DBX32.6 == 0 (ファンクション要求)
   - DB19.DBX32.7 == 0 (状態)
2. インターフェースが空いている場合は、ジョブデータを入力します。これで、ファンクション要求が設定されます。
   - DB19.DBB33 = <チャネル番号>
   - DB19.DBX32.0 - .5 = 1 (ファンクション番号)
   - DB19.DBX32.6 = 1 (ファンクション要求)

HMI → PLC

パラメータ設定に**エラーがない**場合は、HMI が以下のように動作します。

1. チャネル選択のファンクション要求が HMI で認識されると、状態が「ファンクション実行中」に設定され、ファンクション要求がリセットされます。
   - DB19.DBX32.7 = 1 (状態)
   - DB19.DBX32.6 = 0 (ファンクション要求)
2. チャネル選択が実行されると、状態が再度リセットされ、エラー ID として値 0 が設定されます。
   - DB19.DBX32.7 = 0 (状態)
   - DB19.DBX36 = 0 (エラー ID)

パラメータ設定に**エラーがある**場合は、HMI が以下のように動作します。

- ファンクション要求がリセットされ、適切なエラー ID が設定されます。
  - DB19.DBX32.6 = 0 (ファンクション要求)
  - DB19.DBX36 = <エラー ID>

14.13.2 プログラムの選択

機能

事前選択されたプログラム/ワークを、NC で加工するために、PLC/HMI インターフェースを介して選択できます。

この事前選択は、ファイルにプログラム/ワークを入力することによって実装されます。これは、PLC プログラムリスト(*.ppl)と呼ばれています。

必要条件

HMI がタスクを処理できるよう、以下のマシンデータを設定してください。

MD9106 $MM_SERVE_EXTCALL_PROGRAMS

分野別の PLC プログラムリストを有効にするために、適切なマシンデータと、少なくとも保護レベルのパスワードを設定してください。

- **ユーザー**領域
  - MD51041 $MN_ENABLE_PROGLIST_USER = 1
  - 保護レベルのパスワード:3 (ユーザー)
  - プログラムリスト: /user/sinumerik/hmi/plc/programlist/plc_proglist_user.ppl
- **工作機械メーカ(OEM)**領域
  - MD51043 $MN_ENABLE_PROGLIST_MANUFACT = 1
  - 保護レベルのパスワード:1 (工作機械メーカ)
  - プログラムリスト: /oem/sinumerik/hmi/plc/programlist/ plc_proglist_manufacturer.ppl

プログラムリストの構造

プログラムリストはテキストファイルです。各行には、以下の情報が含まれます。

<プログラム番号> <プログラムパス><プログラム名> [CH=<チャネル番号>]

- プログラム番号
  分野に応じてプログラムリストで使用できるプログラム番号。
  - user:1 - 100
  - 工作機械メーカ(OEM):201 - 255
- プログラムパス
  プログラムパスは、完全な絶対パスで指定してください。
  プログラムパスの指定については、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
- チャネル番号
  チャネル番号「CH=<チャネル番号>」の指定はオプションです。これが必要なのは、NC に複数のチャネルがある場合のみです。

以下の抜粋の例は、ユーザープログラムリストの構造を示しています。

プログラムリスト:plc_proglist_user.ppl

```
1 //DEV2:/MPFDir/PROG_01.MPF CH=1
2 //DEV2:/MPFDir/PROG_01.MPF CH=2
```

プログラムリストへの入力の生成

プログラムリスト(*.ppl)への入力は、ファイルで直接編集するか、ユーザーインターフェースの画面形式に入力することができます。

- **ユーザー**領域のユーザーインターフェースで
  操作エリア[プログラムマネージャ|ETC キー([>])|プログラムリスト]
- **工作機械メーカ**領域のユーザーインターフェースで
  操作エリア[コミッショニング|システムデータ|ETC キー([>])|プログラムリスト]

プログラムの選択:ジョブインターフェース

注記

PLC は、最後のジョブに対して HMI が確認応答している場合にのみ、新しいジョブを要求できます。DB19.DBB26 == 0

プログラムリスト

DB19.DBB16 = <プログラムリストの番号>

番号	プログラムリスト
129	/user/sinumerik/hmi/plc/programlist/**plc_proglist_user.ppl**
131	/oem/sinumerik/hmi/plc/programlist/**plc_proglist_manufacturer.ppl**

プログラム番号

プログラム番号は、選択したプログラムリストに含まれるプログラムを表します。

DB19.DBB17 = <プログラム番号>

- ユーザー領域:1 - 100
- OEM 領域:201 - 255

プログラムの選択の要求

DB19.DBX13.7 = 1

プログラムの選択:確認のインターフェース

ジョブの確認

- DB19.DBX26.7 == 1 (選択が識別済み)
- DB19.DBX26.3 == 1 (プログラムが選択済み)
- DB19.DBX26.2 == 1 (プログラム選択時のエラー、エラー ID DB19.DBB27 を参照)
- DB19.DBX26.1 == 1 (ジョブが完了)

エラーの検出

DB19.DBB27 == <エラー ID>

エラーの検出	
値	**意味**
0	異常なし
1	無効なプログラムリスト番号(DB19.DBB16)
3	ユーザーが指定したプログラムリスト plc_proglist_main.ppl が見つからない (DB19.DBB16 ≠ 129、131 の場合のみ)
4	無効なプログラム番号(DB19.DBB17)
5	選択されたワーク内のジョブリストを開けませんでした。

エラーの検出	
6	ジョブリスト内の異常(ジョブリストインタプリタが異常を返します)
7	ジョブリストインタプリタが空のジョブリストを返します

プログラムの選択:ジョブの処理

プログラムを選択するためのジョブは、以下のように実行します。

1. 確認バイトのチェック:DB19.DBB26 == 0
   確認バイトが 0 ではない場合は、最後のジョブが、まだ完了していません。
2. プログラムリストの指定:DB19.DBB16
3. プログラム番号の指定:DB19.DBB17
4. プログラムを選択するための要求の設定:DB19.DBX13.7 = 1
5. 確認インターフェースとエラーインターフェースの評価:DB19.DBB26 および DBB27
   以下が該当する限り、HMI 側では、要求がまだ完了されていません。DB19.DBX26.3 == 1 (有効)
   以下の 2 つの信号のいずれかが設定されている場合は、要求が完了されています。
   - DB19.DBX26.1 == 1 (OK)
   - DB19.DBX26.2 == 1 (エラー)
6. 要求を完了するには、プログラムの選択要求をリセットしてください。DB19.DBX13.7 = 0
7. 確認バイトのリセットにより HMI が新しい要求を受け入れられる状態になったことを示す HMI 信号。DB19.DBB26 == 0

14.13.3 キーロックの有効化

操作パネルのキーボードと、HMI に直接接続されたキーボードは、以下のインターフェース信号を使用して無効にすることができます。

- 1.HMI:DB19.DBX0.2 = <値>
- 2.HMI:DB19.DBX50.2 = <値>

値	意味
0	キーロックが無効
1	キーロックが有効

14.13.4 操作エリア番号

有効な操作エリアの数は、通常、以下で表示されます。DB19.DBB21

HMI モニタが有効な場合は、有効な操作エリアの数が DB19.DBB21 に表示されなくなりますが、代わりに、ユーザー別に設定された、HMI モニタ (ページ 1138)の領域に表示されます。

操作エリア	番号
運転	201
パラメータ	205
プログラム	203
プログラムマネージャ	202
アラーム診断	204
セットアップ	206

14.13.5 画面番号

現在の画面番号は、通常、以下で表示されます。DB19.DBW24

HMI モニタが有効な場合は、現在の画面番号が DB19.DBW24 に表示されなくなりますが、代わりに、ユーザー別に設定された、HMI モニタ (ページ 1138)の領域に表示されます。

画面番号の範囲

以下の画面番号の範囲が使用可能です。

- JOG、Manual Machine (ページ 1127)
- 基準点アプローチ (ページ 1133)
- MDI (ページ 1134)
- AUTOMATIC (ページ 1134)
- パラメータ操作エリア (ページ 1135)
- プログラム操作エリア (ページ 1137)
- プログラムマネージャの操作エリア (ページ 1137)
- 診断操作エリア (ページ 1138)

14.13.5.1 画面番号:JOG、Manual Machine

JOG モード

図	番号
旋削加工	
引き継ぎが可能な、すべての画面のサイクルスタート画面	81
フライス盤用途	
引き継ぎが可能な、すべての画面のサイクルスタート画面	3
旋削/フライス削り	
開始画面	19
T、S、M	2
Set WO	21
位置決め	4
正面削り	18
切削	80
すべてのユーザー画面のサイクルスタート画面	91
一般設定	1
マルチチャネル機能の設定	106
干渉の回避の設定	107
計測ログの設定	108
旋回	60
すべての G 命令	100
実際のズーム値(MCS/WCS)	101
ねじの原点同期	102
後退	103
ハンドル	104
操作の同期	105
旋削加工:ワーク原点	
ワーク原点(メインメニュー)	30
ユーザー画面	31
ユーザー画面	34
ユーザー画面	35

図	番号
ユーザー画面	36
ユーザー画面	37
ユーザー画面	38
ユーザー画面	40
測定エッジ Z	5
旋削加工:ワーク、計測	
工具計測(メインメニュー)	50
手動 X またはユーザー画面	51
手動 Y	71
手動 Z またはユーザー画面	52
ズームまたはユーザー画面	53
ユーザー画面	54
ユーザー画面	55
プローブ校正 X またはユーザー画面	56
プローブ校正 Z またはユーザー画面	57
Z での自動長さ	58
Y での自動長さ	73
X での自動長さ	59
フライス加工:ワーク原点	
ワーク原点(メインメニュー)	30
エッジ X を計測	5
エッジ X を計測	22
測定エッジ Z	23
ユーザー画面	7
エッジ揃えまたはユーザー画面	31
距離 2 エッジまたはユーザー画面	32
直角コーナ	33
任意のコーナまたはユーザー画面	8
長方形ポケット	34
1 個の穴またはユーザー画面	9
2 個の穴	35

図	番号
3 個の穴	36
4 個の穴	37
長方形凸形状	38
1 個の円形スピゴットまたはユーザー画面	10
2 個の丸凸形状	39
3 個の丸凸形状	40
4 個の丸凸形状	41
レベルを設定	42
プローブ長さの校正またはユーザー画面	11
プローブ半径の校正	12
フライス加工:ワーク、計測	
工具計測(メインメニュー)	50
長さを測定、手動(フライス工具で)、 または X で長さを測定、手動(旋削工具で)、 またはユーザー画面	16
Y で長さを測定、手動(旋削工具で)	74
Z で長さを測定、手動(旋削工具で)	24
直径を測定、手動、またはユーザー画面	17
長さを測定、自動(フライス工具で)、 または X で長さを測定、自動(旋削工具で)、 またはユーザー画面	13
Y で長さを測定、自動(旋削工具で)	75
Z で長さを測定、自動(旋削工具で)	25
直径を測定、自動、またはユーザー画面	14
ユーザー画面	51
プローブ校正またはユーザー画面	15
固定点の校正またはユーザー画面	52
RunMyScreens (JobShopIntegration を設定した場合のみ)	
最初の水平ソフトキーのユーザー画面	96
2 番目の水平ソフトキーのユーザー画面	98
3 番目の水平ソフトキーのユーザー画面	99
4 番目の水平ソフトキーのユーザー画面	94

図	番号
5 番目の水平ソフトキーのユーザー画面	95
6 番目の水平ソフトキーのユーザー画面	92
7 番目の水平ソフトキーのユーザー画面	97
8 番目の水平ソフトキーのユーザー画面	90
9 番目の水平ソフトキーのユーザー画面	83
10 番目の水平ソフトキーのユーザー画面	82
11 番目の水平ソフトキーのユーザー画面	93
12 番目の水平ソフトキーのユーザー画面	84
13 番目の水平ソフトキーのユーザー画面	85
14 番目の水平ソフトキーのユーザー画面	86
15 番目の水平ソフトキーのユーザー画面	87
16 番目の水平ソフトキーのユーザー画面	88

JOG モード、Manual Machine

DB19.DBB24	
図	画面番号
旋削/フライス削り	
テーパ旋削	61
角度フライス削り	62
ストップ	63
直線	1300
すべての軸が直線	1330
X アルファが直線	1340
Z アルファが直線	1350
円弧	1360
穴あけ	1400
センタ穴あけ	1410
穴あけ、ねじ切りの中央揃え	1420
穴あけ、センタリング	1433
穴あけ、穴あけ	1434
穴あけ、リーマ仕上	1435

DB19.DBB24	
図	画面番号
穴あけ、ボーリング	1436
穴あけ、深穴ドリル	1440
穴あけ、深穴ドリル 2	1441
穴あけ、タッピング	1453
ねじ切り穴あけ	1455
第	1473
位置行	1474
位置グリッド	1477
位置フレーム	1478
円弧位置決め	1475
円弧位置決め	1479
障害物	1476
旋削	1500
旋削、切り粉除去 1	1513
旋削、切り粉除去 2	1514
旋削、切り粉除去 3	1515
旋削、溝 1	1523
旋削、溝 2	1524
旋削、溝 3	1525
旋削、アンダーカット形状 E	1533
旋削、アンダーカット形状 F	1534
旋削、アンダーカットねじ DIN	1535
旋削、アンダーカットねじ	1536
旋削、ねじ、縦	1543
旋削、ねじ、テーパ	1544
旋削、ねじ、面削り	1545
旋削、ねじ、連続	1546
旋削、遮断	1550
フライス削り	1600
フライス削り、正面削り	1610

DB19.DBB24	
図	画面番号
フライス削り、長方形ポケット	1613
フライス削り、円形ポケット	1614
フライス削り、長方形の凸形状	1623
フライス削り、円弧の凸形状	1624
フライス削り、直線溝	1633
フライス削り、円周溝	1634
フライス削り、オープン溝	1635
フライス削り、多角形	1640
フライス削り、ねじ切り	1454
フライス削り、彫り込み	1670
輪郭旋削	1200
輪郭旋削、新規輪郭/最終輪郭	1210
輪郭旋削、輪郭に沿った旋削	1220
輪郭旋削、輪郭の溝削り	1230
輪郭旋削、輪郭のプランジ旋削	1240
輪郭切削	1100
輪郭のフライス削り、新規輪郭/最終輪郭	1110
輪郭のフライス削り、軌跡のフライス削り	1120
輪郭フライス削り、センタリング	1130
輪郭フライス削り、下穴あけ	1140
輪郭フライス削り、輪郭ポケット	1150
旋削加工:シミュレーション	
側面図	1740
正面図	1750
3D 表示	1760
2 画面表示	1770
片側断面図	1780
旋削加工:同時描画	
側面図	1741
正面図	1751

DB19.DBB24	
図	画面番号
3D 表示	1761
2 画面表示	1771
機械スペース	1791
片側断面図	1781
フライス加工:シミュレーション	
平面図	1742
3D 表示	1760
正面図	1744
背面図	1746
左から	1748
右から	1752
片側断面図	1780
旋削図	1782
フライス加工:同時描画	
平面図	1743
3D 表示	1761
正面図	1745
背面図	1747
左から	1749
右から	1753
機械スペース	1791
片側断面図	1781
旋削図	1783

14.13.5.2 画面番号:基準点アプローチ

図	番号
実際のズーム値 MCS/WCS	101

14.13.5.3 画面番号:MDI

図	番号
MDI	20
すべての G 命令	100
実際のズーム値 MCS/WCS	101
ハンドル	104
操作の同期	105
プログラム制御	210
設定	250

14.13.5.4 画面番号:AUTOMATIC

図	番号
自動	200
オーバーストア	202
プログラム制御	210
ブロックサーチ	220
一般設定	250
マルチチャネル機能の設定	106
干渉の回避の設定	107
すべての G 命令	100
実際のズーム値 MCS/WCS	101
ハンドル	104
操作の同期	105
旋削加工:同時描画	
側面図	243
正面図	244
3D 表示	245
2 画面表示	246
機械スペース	247
片側断面図	253

図	番号
フライス加工:同時描画	
平面図	242
3D 表示	244
正面図	248
背面図	249
左から	251
右から	252
機械スペース	247
片側断面図	253
旋削図	254

14.13.5.5 画面番号:パラメータ操作エリア

図	番号
工具リスト	600
工具摩耗	610
ユーザー工具リスト	620
マガジン	630
ゼロオフセット	
ゼロオフセット、有効	642
ゼロオフセット、概要	643
ゼロオフセット、基本	644
ゼロオフセット、G54 - G509	645
ゼロオフセットの詳細、有効、概要、基本、または G54 - G509	647
ユーザー変数	
R 変数	650
グローバル GUD 1 (SGUD)	660
グローバル GUD 2 (MGUD)	661
グローバル GUD 3 (UGUD)	662
グローバル GUD4	663
グローバル GUD5	664

図	番号
グローバル GUD6	665
グローバル GUD7	666
グローバル GUD8	667
グローバル GUD9	668
チャネル GUD 1 (SGUD)	690
チャネル GUD 2 (MGUD)	691
チャネル GUD 3 (UGUD)	692
チャネル GUD 4	693
チャネル GUD 5	694
チャネル GUD 6	695
チャネル GUD 7	696
チャネル GUD 8	697
チャネル GUD 9	698
ローカル LUD	681
ローカル LUD/PUD	684
セッティングデータ	
作業領域リミット	671
主軸データ	670
主軸チャックデータ	672
Ctrl-Energy	
Ctrl-Energy、メインメニュー	6170
Ctrl-Energy、分析	6171
Ctrl-Energy、プロファイル	6172
Ctrl-Energy、グラフィック分析	6176
Ctrl-Energy, 長期的計測分析	6177
Ctrl-Energy、詳細分析	6179
Ctrl-Energy、計測比較	6178

14.13.5.6 画面番号:プログラム操作エリア

図	番号
旋削加工:シミュレーション	
側面図	413
正面図	414
3D 表示	415
2 画面表示	416
片側断面図	423
フライス加工:シミュレーション	
平面図	412
3D 表示	414
正面図	418
背面図	419
左から	421
右から	422
片側断面図	423
旋削図	424

14.13.5.7 画面番号:プログラムマネージャ操作エリア

図	番号
NC ディレクトリ	300
ローカルドライブ	325
USB/ドライブ 1 構成	330
ドライブ 2 構成	340
ドライブ 3 構成	350
ドライブ 4 構成	360
ドライブ 5 構成	383
ドライブ 6 構成	384
ドライブ 7 構成	385
ドライブ 8 構成	386

14.13.5.8 画面番号:アラーム診断操作エリア

図	番号
アラームリスト	500
メッセージ	501
アラームログ	502
NC/PLC 変数	503

14.13.6 HMI モニタ

機能

HMI モニタは、自由に選択可能なデータブロックの 8 バイトのデータ領域です。ここで、HMI が PLC ユーザープログラムに、以下のデータを提供できます。

- 操作エリア番号 (ページ 1125)
- 画面番号 (ページ 1126)

パラメータ設定

データ領域は、以下の表示マシンデータを使用して設定されます。

MD9032 $MM_HMI_MONITOR =「文字列」

ここで、「文字列」=「DB<DB 番号>.DBB<バイトアドレス>」

注記

偶数のバイトアドレス

データ領域は必ず、偶数のバイトアドレスで始まります。

データ領域の構造

バイト	意味
EB n + 0	有効な SINUMERIK 操作エリア
EB n + 1	予備
EB n + 2	現在の画面番号
EB n + 3	
EB n + 4	予備

バイト	意味
...	...
EB n + 7	予備

必要条件

HMI モニタが有効なときは、以下の PLC/HMI インターフェース信号が処理されなくなります。

- DB19.DBB10 (PLC ハードキー)
- DB19.DBB21 (有効な SINUMERIK 操作エリア)
- DB19.DBW24 (現在の画面番号)

例

前提条件

- 現在の操作エリア「機械」、番号:201
- 実際の画面:「自動」開始画面、番号:200
- PLC データ領域:DB60.DBB10

パラメータ設定

- MD9032 $MM_HMI_MONITOR =「DB60.DBB10」

データ領域の値

- DB60.DBB10:201
- DB60.DBW12:200

14.14 内蔵 PROFIBUS 上のドライブコンポーネントの PLC 機能

14.14.1 一覧

以下に説明する機能を使用して、内蔵 PROFIBUS 上のドライブコンポーネントでの入力および出力データを、**ハードウェア PLC** の PLC ユーザープログラムから一貫して周期的に読み取り、書き込むことができます。次の境界条件を遵守してください。

- 入力/出力データの読み取り/書き込みには、システムファンクション **SFC14/SFC15** を使用してください。
- 入力/出力データの読み取り/書き込みは、常にスロットの長さ**全体**にわたります。
- この機能がサポートしているのは、**等間隔でない**、周期的なデータ転送のみです。
- 出力スロットは、まだ、NC 側(ドライブ装置の出力スロットなど)で占有されていない場合があります。

14.14.2 セットアップの実行

必要条件

NC 側で機能が動作を開始する前に、以下の必要条件を満たしていなければなりません。

- NCU の内蔵 PROFIBUS 上のドライブコンポーネントが、SIMATIC STEP 7、HW Config を使用して完全に設定されています。
- 設定が PLC にロードされています。

NC マシンデータ

周期的に伝送されるスロットの先頭アドレスを、以下のマシンデータに入力します。

- 入力スロット:
  MD10520 $MN_PLCINTERN_LOGIC_ADDRESS_IN[<インデックス>] = <スロットアドレス>
- 出力スロット:
  MD10525 $MN_PLCINTERN_LOGIC_ADDRESS_OUT[<インデックス>] = <スロットアドレス>

<インデックス>:0, 1, 2, ...(スロットの最大数) – 1

<スロットアドレス>:HW Config でパラメータ設定するスロットアドレス

注記

データの最大量

発信および着信として周期的に伝送されるすべてのデータの合計が、現在、**2,048** バイトを超えないようにしてください。

14.14.3 例

スロットアドレスの特定

DP スレーブ「SINAMICS_Integrated」を、HW Config のステーションウィンドウの内蔵 PROFIBUS「PROFIBUS Integrated:DP master system (3)」で選択した後に、その PROFIdrive メッセージフレームと関連スロットアドレスが詳細ビューに表示されます。

- メッセージフレーム 136:ドライブ
- メッセージフレーム 391:コントロールユニット
- メッセージフレーム **370**:切り込み

NC 機械でのパラメータ設定に必要な切り込みスロットアドレスは、次のとおりです。

- 入力スロット:スロット 31、アドレス **6514**
- 出力スロット:スロット 32、アドレス **6514**

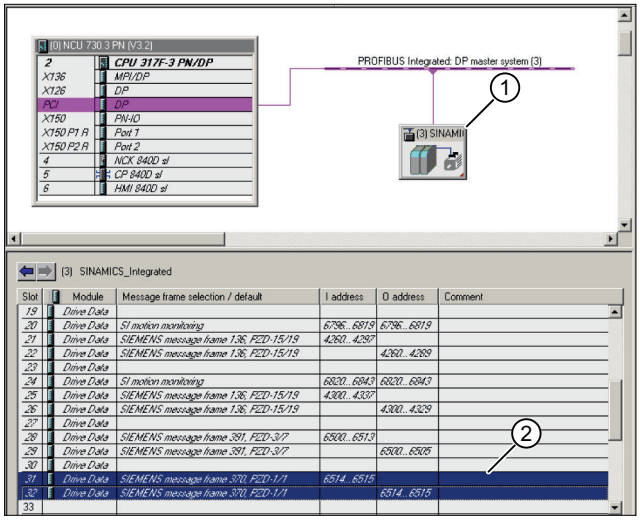


図 14-18 切り込み PROFIBUS メッセージフレーム

NC マシンデータの設定

- 入力スロット:
  MD10520 $MN_PLCINTERN_LOGIC_ADDRESS_IN[0] = 6514
- 出力スロット:
  MD10525 $MN_PLCINTERN_LOGIC_ADDRESS_OUT[0] = 6514

FB390「ALM_Control」を使用した ALM の制御

SINUMERIK ハードウェア PLC を、内部 PROFIBUS 経由で SINAMICS S120 の CU320 に接続します。ALM を DRIVE-CLiQ 経由で CU に接続します。

SINAMICS S120 では、制御とステータスメッセージフレームを各モジュール(CU、ALM、モータモジュールなど)に対して定義できます。これを ALM に対して実行した場合は、ALM のオン/オフを PLC ユーザープログラムから実行できます。

SIMATIC S7 ブロック FB390「ALM_Control」は ALM の状態をチェックし、ユーザーがオン/オフを実行できるようにします。

ブロックの概要とサンプルプロジェクトは、以下のリンク先「Industry Online Support」でダウンロードできます。

http://support.automation.siemens.com/WW/view/de/49515414

14.15 PLC 基本プログラムの必要メモリ

基本プログラムは、基本機能とオプション機能で構成されます。**基本機能**には、NC とPLC の間の周期的信号交信が含まれます。**オプション**には、必要に応じて使用できる FC などがあります。

次の表に、基本機能とオプションの必要メモリを示します。引用したデータはあくまでも目安となる値で、実際の値は最新のソフトウェアバージョンによって異なります。

SINUMERIK 840D sl のブロックの必要メモリ			
ブロック タイプ、番号	機能	備考	ブロックサイズ(バイト)
	作業メモリ		

基本プログラムにある基本機能			
FB1、FB15		ロードすること/ コンパクトフラッシュカードカード上で	52182
FC2、3、5、12		ロードすること	470
DB4、5、7、8		ロードすること	1006
DB2、3、17		BP によって生成	632
OB1、40、100、 82、86		ロードすること	398
		合計	55698

PLC/NC、PLC/HMI インタフェース			
DB10	PLC/NC 信号	ロードすること	262
DB11	PLC/モードグループ信号	BP によって生成	56
DB19	PLC/HMI 信号	BP によって生成	434
DB21 ～ 30	PLC/チャネル信号	NCMD に応じて BP によって生成:DB 毎に	416
DB31 ～ 61	PLC /軸または主軸信号	NCMD に応じて BP によって生成:DB 毎に	148

基本プログラムオプション			
	機械操作パネル		
FC19	MCP 信号の伝送、M タイプ	MCP の M タイプを取り付けるときにロードすること	92
FC25	MCP 信号の伝送、T タイプ	MCP の T タイプを取り付けるときにロードすること	92
FC24	MCP 信号の伝送、スリムタイプ	MCP のスリムタイプを取り付けるときにロードすること	100
FC26	MCP 信号の伝送、HT8 タイプ	HT8 用にロードすること	68
	ハンドヘルドユニット		
FC13	表示／制御 HHU	ハンドヘルドユニット用にロード可能	144

基本プログラムオプション			
	異常/操作メッセージ		
FC10	異常/操作メッセージ取得	異常/操作メッセージの使用時にロード	66
	ASUB		
FC9	ASUB の起動	PLC ASUB の使用時にロード	128

基本プログラムオプション			
	スター/デルタ切り替え		
FC17	MSD のスター/デルタ切り替え	スター/デルタ切り替え用にロード	114
	主軸制御		
FC18	主軸制御	PLC からの主軸制御用にロード	132
	PLC/NC 通信		
FB2	NC 変数の読み取り	NC 変数の読み取り用にロード	76
DBn	NC 変数の読み取り	FB2 呼び出し毎に 1 つのインスタンス DB:	270
FB3	NC 変数の書き込み	NC 変数の書き込み用にロード	76
DBm	NC 変数の書き込み	FB3 呼び出し毎に 1 つのインスタンス DB:	270
FB4	PI サービス	PI サービス用にロード	76
DBo	PI サービス	FB4 呼び出し毎に 1 つのインスタンス DB:	130
DB16	PI サービス記述	PI サービス用にロード	618
FB5	GUD 変数の読み取り	PI サービス用にロード	76
DBp	GUD 変数の読み取り	FB5 呼び出し毎に 1 つのインスタンス DB:	166
DB15	一般通信	通信用のグローバルデータブロック	146
FB7	PI サービス 2	PI サービス用にロード	76
DBo	PI サービス 2	FB4 呼び出し毎に 1 つのインスタンス DB:すべて	144
FC21	伝送	デュアルポート RAM でロード、...	164

基本プログラムオプション			
	M 対 N		
FB9	M 対 N の切り替え	M 対 N でロード	58
	Safety Integrated		
FB10	安全リレー	セーフティオプションでロード	74
FB11	ブレーキテスト	セーフティオプションでロード	76
DB18	セーフティデータ	安全用の DB	308
	工具管理機能		
FC7	伝送機能 タレット	工具管理機能オプション用にロード	84
FC8	伝送機能	工具管理機能オプション用にロード	132
FC22	方向選択	方向選択が必要な場合にロード	138
DB71	ロードポイント	NC MD に応じて BP によって生成	40+30*B
DB72	主軸	NC MD に応じて BP によって生成	40+48*Sp
DB73	リボルバ	NC MD に応じて BP によって生成	40+44*R
DB74	基本機能	NC MD に応じて BP によって生成	100+(B+Sp+R)*22
	コンパイルサイクル		
DB9	インタフェース PLC コンパイルサイクル	NC オプションに応じて BP によって生成	436

例:

上記の表に示す必要メモリに基づいて、2 つの構成サンプルについて必要メモリが決定されています(次の表を参照してください)。

ブロック タイプ、番号	機能	備考	ブロックサイズ(バイト)
	作業メモリ		

最小構成(1 主軸、2 軸および T MCP)			
上記を参照	基本プログラム、ベース		54688
	インタフェース DB		1612
	MCP		92
		合計	56392

ブロック タイプ、番号	**機能**	**備考**	**ブロックサイズ(バイト)**
	作業メモリ		

最大構成(2 チャネル、4 主軸、4 軸、T MCP)			
上記を参照	基本プログラム、ベース		54688
上記を参照	インタフェース DB		2768
上記を参照	MCP		92
上記を参照	異常/操作メッセージ		66
上記を参照	ASUB	1 つの ASUB 起動	128
上記を参照	同時軸	2 つのタレット用	132
上記を参照	PLC/NC 通信	1 つの読み取り変数と 1 つの書き込み変数	838
上記を参照	工具管理機能	1 つのロードポイントにつき 2 つのタレット	674
上記を参照	コンパイルサイクル		436
		合計	59822

14.16 必要条件および NC VAR セレクタ

14.16.1 必要条件

14.16.1.1 プログラミングとパラメータ設定ツール

ハードウェア

SINUMERIK 840D sl で使用される PLC の場合、プログラミング装置または PC で次の装備が必要になります。

	最低	推奨
プロセッサ	Pentium	Pentium
RAM (MB)	256	512 以上
ハードディスク、 空き容量(MB)	500	> 500
インターフェース	MPI、Ethernet (ケーブルを含む) メモリカード	
グラフィック	SVGA (1024*768)	
マウス	必要	
オペレーティングシステム	Windows 2000 /XP Professional、STEP 7 バージョン 5.3 SP2 以上	

STEP 7 パッケージがまだプログラミング装置に提供されていない場合、前述の必要条件を満たす装置に、**STEP 7** の必要なバージョンをインストールできます。

このパッケージを使用すると、次の機能を実行できます。

- プログラミング
  - STL 用エディタおよびコンパイラ(言語の範囲全体、SFB/SFC 呼び出しを含む)、LAD、FBD
  - 割り当てリストの作成と編集(シンボルエディタ)
  - データブロックエディタ
  - オン／オフラインでのブロックの入出力
  - オン／オフラインでの変更と追加の挿入
  - プログラミング装置から PLC へまたはその逆のブロックの伝送
- パラメータ設定
  - CPU と I/O 装置をパラメータ設定するためのパラメータ設定ツール **HW Config**
  - CPU 通信パラメータ設定用の **NetPro** パラメータ設定ツール
  - ハードウェアとソフトウェアのバージョン、メモリ容量、I/O 拡張/割り当てなどのシステムデータの出力
- テストと診断(オンライン)
  - 変数状態/固定(I/O、フラグ、データブロック内容など)
  - 個々のブロックの状態
  - システム状態の表示(ISTACK、BSTACK、システム状態リスト)
  - システムメッセージの表示
  - PLC 停止の起動/再起動/PG からのトータルリセットの実施
  - PLC の圧縮
- 文書作成
  - 個々のブロックまたはすべてのブロックのプリントアウト
  - シンボル名称の割り当て(データブロックにある変数についても)
  - 各ブロック内のコメントの入出力
  - テストと診断表示のプリントアウト
  - ハードコピー機能
  - クロスリファレンスリスト
  - プログラム一覧
  - I/O/M/T/Z/D の割り当て計画

- ユーティリティプログラムのアーカイブ
  - 個々のブロックの出力状態の割り当て
  - ブロックの比較
  - 再配線
  - STEP 5 → STEP 7 コンバータ
- オプションパッケージ
  - S7–HIGRAPH、S7–GRAPH、SCL におけるプログラミング。
    これらのパッケージは、当社販売部にご注文ください。
  - 構成モジュール用の追加パッケージ(たとえば、CP3425 → NCM パッケージ)

注記

使用できる機能の詳細は、SIMATIC カタログと STEP 7 の取扱説明書を参照してください。

14.16.1.2 必要な SIMATIC 取扱説明書

参照先:

- システム説明 SIMATIC S7
- S7-300 命令リスト
- STEP 7 でのプログラミング
- ユーザーマニュアル STEP 7
- プログラミングマニュアル STEP 7; ユーザープログラムの設計
- リファレンスマニュアル STEP 7; 命令リスト AWL
- リファレンスマニュアル STEP 7; ラダー図 KOP
- リファレンスマニュアル STEP 7; 標準およびシステムファンクション
- マニュアル STEP 7; STEP 5 プログラムの変換
- STEP 7 の全体のインデックス
- マニュアル CPU 317-2DP

14.16.1.3 関連 SINUMERIK 文書

参照先:

- 『試運転マニュアル IBN CNC』:NC、PLC、ドライブ
- 『オペレータコンポーネントとネットワーク設定マニュアル』

- 機能マニュアル、基本機能
- 機能マニュアル、上級機能
- 『機能マニュアル、応用機能』
- データ一覧表 sl (第 1 巻)
- データ一覧表 sl (第 2 巻)

14.16.2 NC VAR セレクタ

14.16.2.1 概要

一般事項

PC アプリケーションの「NC VAR セレクタ」は、必要な NC 変数のアドレスを取得し、PLC プログラム(FB2/FB3)内でアクセスするために、それらを処理します。これによりプログラマは、NC 変数の全範囲から NC 変数を選択し、この変数の選択を保存し、STEP 7 コンパイラ用コードジェネレータを使用してこれらを編集することができます。最後に、これらを機械 CPU プログラムに ASCII ファイル(*.AWL)として保存できます。図「NC VAR セレクタ」にこの処理を示します。

NC-VAR セレクタによって作成されたファイルを保存するために、Windows Explorer を介して任意のカタログ名称でカタログを実装してください。NC-VAR セレクタの選択されたデータ(data.VAR と data.AWL ファイル)は、このカタログに保存してください。その後、STL ファイルが転送され、メニューオプション[コード|STEP 7 プロジェクト内]でコンパイルされます。その後、「data.AWL」(STL データ)ファイルを、STEP 7 マネージャで[挿入|外部メモリ]を使用して STEP 7 マシンプロジェクトに挿入してください。このために、マネージャでソースのコンテナを選択してください。この操作により、このファイルはプロジェクト構造内に保存されます。ファイルが伝送されたら、これらの AWL (STL)ファイルを STEP 7 でコンパイルしてください。

注記

各 NC ソフトウェアバージョン(旧バージョンを含む)で、最新の NC VAR セレクタを使用できます。旧 NC ソフトウェアバージョンでも、最新のリスト一式から変数を選択できます。DB120 (変数の初期設定 DB)のデータ内容は、ソフトウェア状態に依存しません。つまり、旧ソフトウェアバージョンで選択された変数は、ソフトウェアがアップグレードされても再選択する必要はありません。

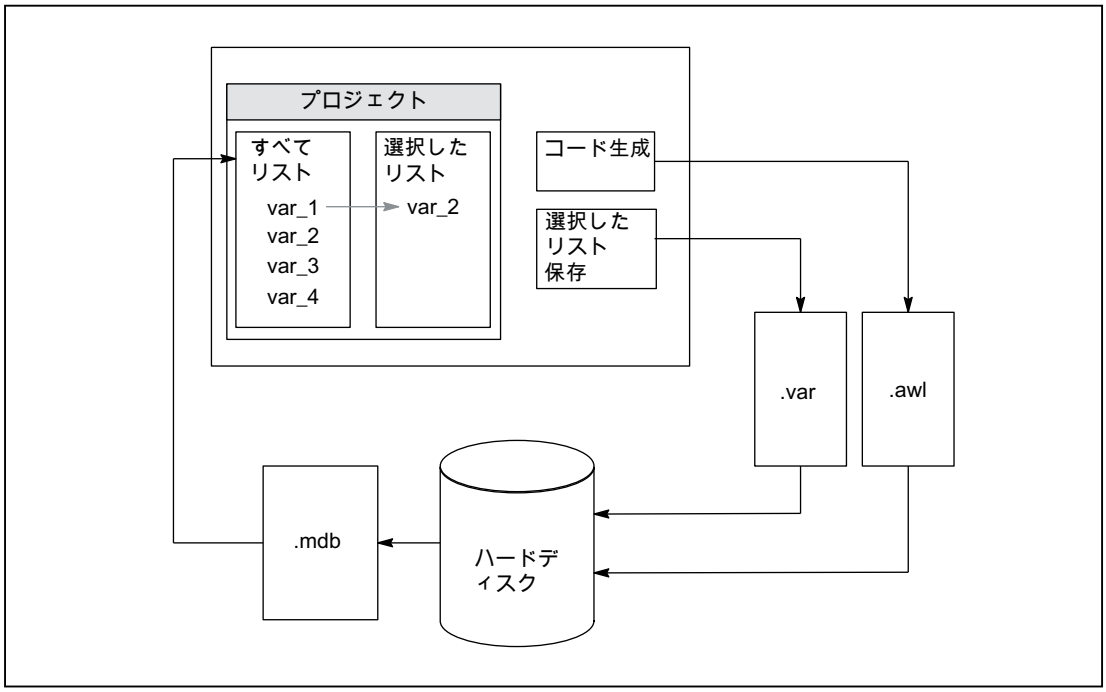


図 14-19　NC VAR セレクタ

「NC VAR セレクタ」アプリケーションが起動した後、NC タイプの変数のリストを選択して(ハードディスク→ファイル Ncv.mdb)、このリストに含まれているすべての変数をウィンドウに表示してください。

以下の ncv*.mdb 変数リストが使用可能です。

変数	一覧
マシンデータとセッティングデータを含む NC 変数:	ncv_NcData.mdb
ドライブのパラメータ:	ncv_SinamicsServo.mdb

ユーザーは 2 番目のリスト(別のウィンドウ)に変数を伝送することもできます。その後、後で選択された方の変数を、ASCII ファイルに保存するか、または STEP 7 ソースファイル(.awl) として編集し、保存することができます。

プログラマは、STEP 7 コンパイラを使用して PLC データブロックを生成した後で、STEP 7 ファイルを使用して、基本プログラムのファンクションブロック「PUT」と「GET」を介して、NC 変数を読み書きできます。

選択変数のリストも ASCII ファイルとして保存されます(ファイル拡張子.var)。

「NC VAR セレクタ」ツールとともに提供されている変数リストは、現在の NC ソフトウェアバージョンに適合しています。このリストには、ユーザーによって定義された変数

(GUD 変数)は含まれていません。これらの変数は、基本プログラムのファンクションブロック FB5 によって処理されます。

注記

「NC VAR セレクタ」の最新バージョンは、これまでの NC ソフトウェアバージョンをすべて処理できます。したがって、「NC VAR セレクタ」の異なるバージョンを一緒にインストールする必要はありません。

システムの特徴、必要条件

PC アプリケーション「NC VAR セレクタ」では、Windows 2000 以上のオペレーティングシステムが必要です。

変数への名前の割り当ての説明は、以下の資料にあります。

参照先:

『/リスト sl (ブック 1)』、「変数」の章

または、変数ヘルプファイル内(NC VAR セレクタに組み込まれています)。

14.16.2.2 機能説明

概要

次の図は、STEP 7 の環境における NC VAR セレクタの使用方法を示しています。

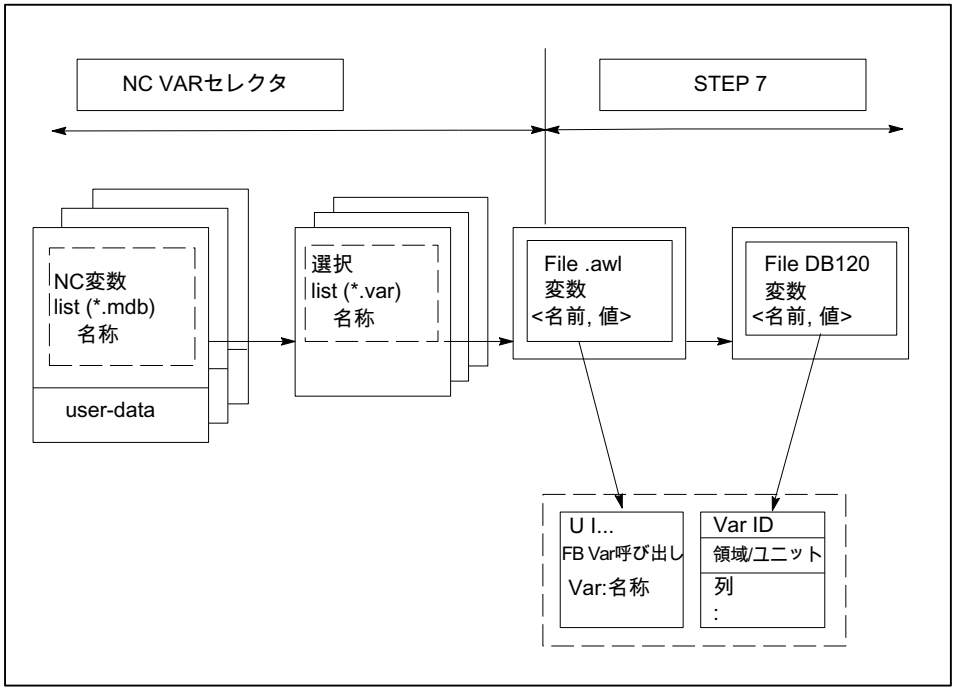


図 14-20 STEP 7 の環境における NC VAR セレクタの用途

NC VAR セレクタは、変数のリストから選択変数のリストを生成してから、STEP 7 コンパイラでコンパイルできる**.awl** ファイルを生成するのに使用されます。

- *.awl ファイルには、NC 変数の名称と別名称と、それらのアドレスパラメータに関する情報が含まれています。このファイルから生成されたデータブロックには、アドレスパラメータ(パラメータ毎に 10 バイト)のみが含まれます。
- 生成したデータブロックは、STEP 7 仕様に従って、機械別ファイル記憶装置に常に保存してください。
- シンボルを使用して、NC アドレスに対して GET/PUT (FB2/3)ブロックを確実にパラメータ設定できるようにするために、生成されたデータブロックの、自由に割り当て可能なシンボル名称を STEP 7 シンボルテーブルに含めてください。

基本画面/基本メニュー

NC VAR セレクタを選択(起動)すると、すべての入力選択(上部のメニューバー)が表示されている基本画面が画面に現れます。表示される他のすべてのウィンドウは汎用ウィンドウ内に配置されます。

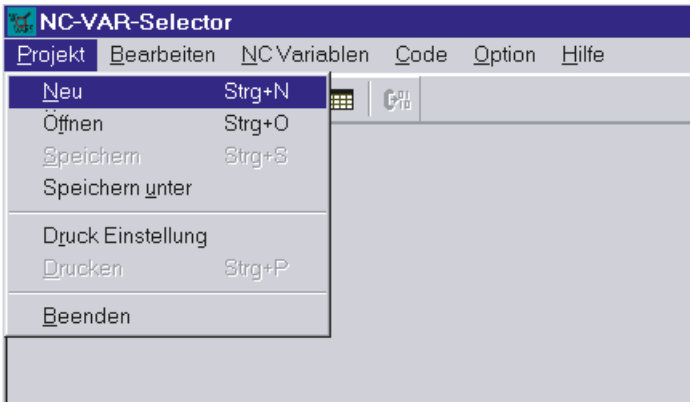


図 14-21　基本メニューが表示されている基本画面

[プロジェクト]メニュー項目

プロジェクトファイル(選択変数のファイル)に関するオペレータ操作は、すべてこのメニュー項目の下でおこないます。

アプリケーションの終了

アプリケーションを終了するには、[プロジェクト]メニュー項目の下にある[終了]を選択します。

新規プロジェクトの作成

新規プロジェクト(選択変数用の新しいファイル)の設定は、[プロジェクト]メニュー項目の下でおこないます。

[新規]を選択すると、選択変数用のウィンドウが表示されます。プロンプトの後に、NC 変数リストのファイル選択が表示されます(NC 変数リストがまだ開いていない場合にのみ適用されます)。

図 14-22　新規プロジェクト用選択変数のウィンドウ

選択変数がウィンドウに表示されます。

既存のプロジェクトを開く

既存のプロジェクト(変数がすでに選択されている)を開くには、[プロジェクト]メニュー項目の下にある[開く]を選択してください。ファイル選択ウィンドウが表示されて、「.var」拡張子が付いた適切なプロジェクトを選択できます。

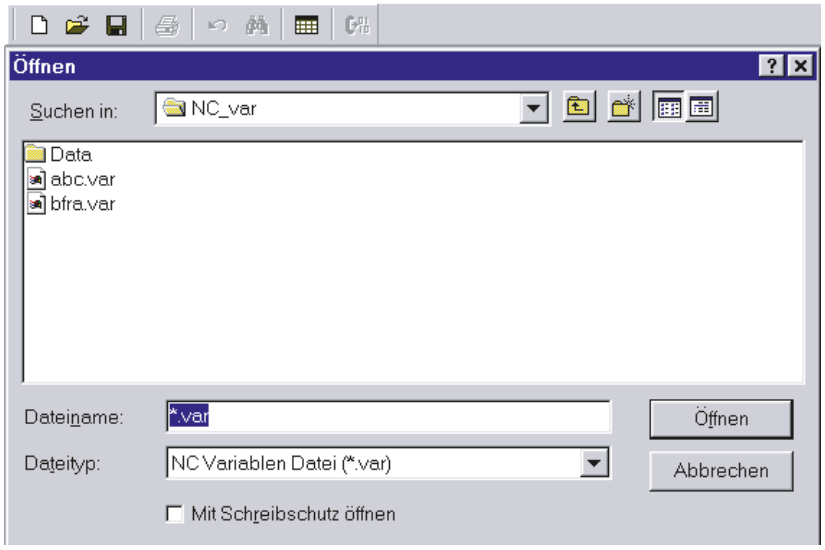


図 14-23　既存のプロジェクトの選択ウィンドウ

プロジェクトを選択した後で、新しい変数の追加が必要な場合は、NC 変数の全リストを選択してください。ユーザーがプロジェクトから変数を削除するだけであれば、全リストを呼び出す必要はありません。

プロジェクトの保存

変数リストは、[プロジェクト|保存]または[名前を付けて保存....]メニュー項目を使用して保存します。

[保存]を選択すると、すでに指定されているパスの下に変数リストが保存されます。プロジェクトパスが未定の場合は、[名前を付けて保存....]と同じ方法で保存します。

[名前を付けて保存...]を選択すると、保存するプロジェクト用のパスを指定できるウィンドウが表示されます。

プロジェクトの印刷

プロジェクトファイルを印刷するには、[プロジェクト]メニュー項目の下にある[印刷]命令を選択します。ページ毎の行数は、[印刷設定]メニュー項目の下で選択します。初期設定は 77 行です。

[編集]メニュー項目

次のオペレータ操作は、このメニュー項目で直接実行できる操作の例です。

- 変数の伝送
- 変数の削除
- 別名称の変更
- 変数の検索

これらの操作は、[編集]の下で再度取り消すこともできます。

操作のアンドゥ

プロジェクトファイルの作成に関するオペレータ操作(変数の伝送、変数の削除、別名称の変更)のアンドゥは、このメニューでおこなうことができます。

[NC 変数]メニュー項目

すべての変数の基本リストは、NC Var セレクタのパス Data\Swxy (xy はソフトウェアバージョン番号を表します。例:SW 5.3:=xy=53)に保存されます。このリストは、NC 変数リストとして選択できます。SINUMERIK 840D sl の場合、基本リストはパス Data \Swxy_sl 内に存在します。

NC 変数リストの選択

[NC 変数リスト]、[選択]メニュー項目を介して、NC バージョンのすべての NC 変数のリストを選択と表示できるようになりました。

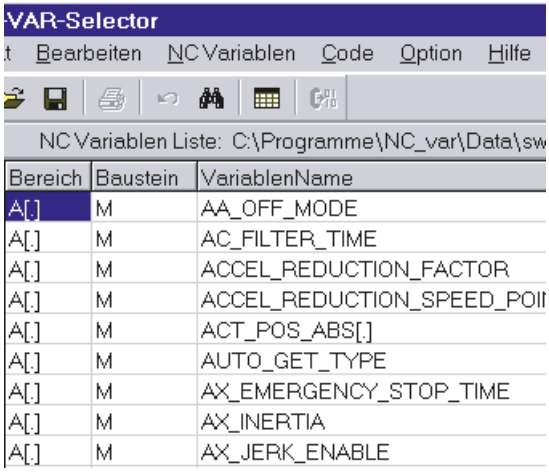


図 14-24　選択された全リストのウィンドウ

フィールド変数(たとえば、軸領域、T 領域データなど)は、括弧([.])を使用して表示されます。追加情報は、ここで指定してください。変数がプロジェクトリストに伝送されると、必要な追加情報が要求されます。

サブセットの表示

表のフィールドをどれでもダブルクリックすると(変数のフィールド以外)、フィルタ基準をプリセットできるウィンドウが表示されます。

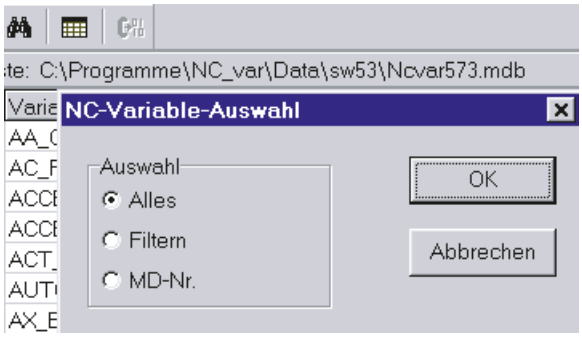


図 14-25　変数リスト表示のフィルタ基準のウィンドウ

次の 3 つのオプションがあります。

- すべてのデータを表示
- 領域、ブロック、名称を入力(組み合わせも含む)
- MD/SE データ番号を表示

次のワイルドカードも使用できます。

*	必要に応じて検索基準の範囲を広げる

検索基準の例

名称検索基準:CHAN* 検索結果: CHAN_NAME
chanAlarm
chanStatus
channelName
chanAssignment

- 変数の選択
  変数を選択するには、マウスを 1 回クリックします。変数を選択変数のウィンドウに伝送するには、ダブルクリックします。この操作も、[編集]メニュー項目の下でアンドゥできます。

別名称

提供される変数名称は、32 文字までの長さがありえます。生成される変数をデータブロック内で明確に識別できるようにするために、選択した名称にいくつかの ASCII 文字が追加されます。しかし、STEP 7 コンパイラは、一義的な STEP 7 変数として 24 個の ASCII 文字しか認識しません。変数名称を最後の 8 文字の位置だけで差別化することはできないため、長すぎる名称の場合は**別名称**が使用されます。そのため、変数の選択時に、使用する STEP 7 名称の長さがチェックされます。名称が 24 文字よりも長い場合、ユーザーは、別名称として使用する追加の名称を入力しなければなりません。

この場合、別名称は必ず一義的にしてください。

入力された別名称は、[オプション]メニューで、ユーザーによっていつでも有効にすることができます。
別名称は、変数が伝送されるたびに入力できます。

S7 変数名称のフィールドをダブルクリックすることによって、後で別名称を編集することもできます。この操作も、[編集]メニュー項目の下でアンドゥできます。

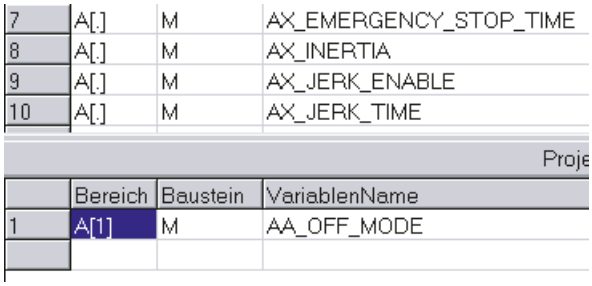


図 14-26　全リストと選択変数の画面

スクロール

すべての変数をウィンドウに表示できない場合、スクロールバーが表示されます。残りの変数は、スクロールする(ページを上下する)と現れます。

多次元構造における変数

多次元構造から変数を選択した場合は、領域番号だけでなく、列または行番号またはその両方も入力して、変数のアドレス指定ができるようにしてください。要求されている番号については、NC 変数の文書を参照してください。

参照先:
『データ一覧表 sl 第 1 巻』；「変数」

ブロック番号または行または列インデックスとしてゼロ(0) を入力すると、S7 PLC にある変数をこれらのデータに対するポインタとして使用できます。「PUT」と「GET」ファンクションを介してこれらのデータの読み取りまたは書き込みをおこなうときに、オプションのパラメータである「UnitX」、「ColumnX」、「LineX」に必要な情報を入力してください。

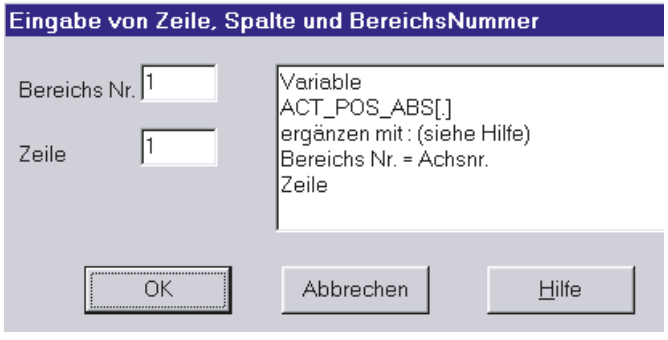


図 14-27　行、列およびブロック番号の入力フィールド

変数の削除

選択変数のウィンドウで変数を削除するには、適切な変数を選択し(マウスを 1 回クリックします)、「削除」キーを押します。ダブルクリック機能では、削除操作は実行されません。複数の変数を削除用に選択することができます(「検索条件の例」の「変数の選択」の章を参照してください)。

この操作も、[編集]メニュー項目の下でアンドゥできます。

注記

変数を削除すると、変数に対するポインタ構造の絶対アドレスが変更されます。したがって、変数選択を変更するときは、**変更する前に**すべてのユーザブロックから 1 つ以上のテキストファイルを必ず**生成する**必要があります。これは、FB「GET」または FB「PUT」における変数の割り当てが、再コンパイル後も正しいままであることを確認する唯一の方法です。

選択リストの保存

変数を選択したら、プロジェクト名称を付けて保存することができます。ファイルは、プロジェクト毎に保存されます。

保存するファイル用のウィンドウが表示されます。このウィンドウで、ファイルのプロジェクトパスと名称を選択してください。

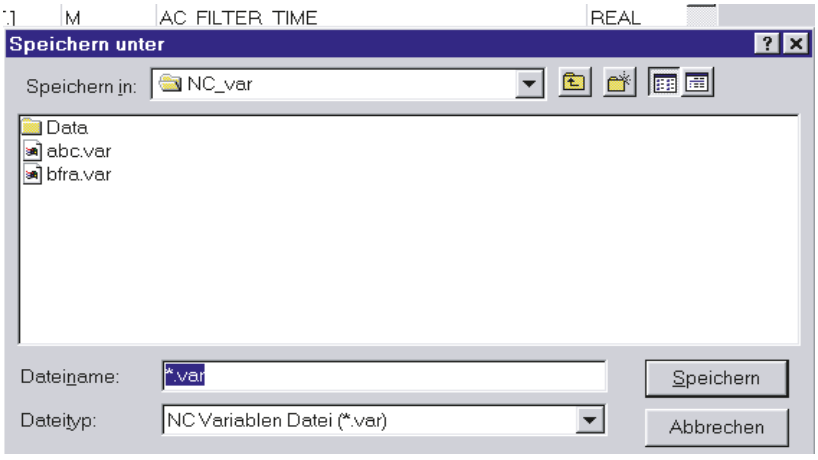


図 14-28　保存するファイルのプロジェクトパスと名称のウィンドウ

コード生成

このメニュー項目には、次の 3 つの選択オプションがあります。

1. 設定(生成されるデータブロック番号の入力)と他の設定
2. 生成(データブロックを作成)
3. STEP 7 プロジェクト内へ(STEP 7 プロジェクトにデータブロックを伝送)

設定

このメニュー項目の下で、コードが作成される DB 番号とこの DB 番号のシンボルを入力します。

[単位系]タブの下で、PLC 内でのユニットシステム変数をどのような単位系で計算するか決定するための選択をおこないます。

[生成]タブの下で、関連するターゲットシステムについてプロジェクト作成を定義します。

生成

このメニュー項目の下で、拡張子「.awl」の選択変数リストの STEP 7 ファイルを設定します。

[選択]をクリックすると、次のファイルが生成されます。

STEP 7 コンパイラ用の入力として使用できる**.awl** ファイル。

ウィンドウが表示されます。ファイルを保存するには、このウィンドウで、生成される.awl ファイルのパスと名称を指定してください。

STEP 7 プロジェクト内へ

生成された STL ファイルは、選択可能な SIMATIC プロジェクト(プログラムパス)に伝送され、コンパイルされます。さらに、シンボルも伝送できます。この機能は、STEP 7 バージョン 5.1 以降で使用できます。この処理には、STEP 7 の呼び出しのために、より長い時間がかかります。新しい STL ファイルを伝送する前に、STL ファイルのファイルウィンドウが LAD/FBD/STL エディタで閉じられます。

[オプション]メニュー項目

[オプション]メニュー項目の下で、次を選択できます。

- 現在の言語
- 別名入力モード(常に 24 文字以下)

[ヘルプ]メニュー項目

次の情報を表示するには、対応するサブメニュー項目を選択します。

- 操作マニュアル
- 変数の説明

著作権とバージョン番号も表示できます。

14.16.2.3 セットアップ、インストール

Windows のアプリケーションである「NC Var セレクタ」は、パッケージに付属の SETUP プログラムを使用してインストールします。

14.17 ブロックの説明

14.17.1 FB1:RUN_UP - 基本プログラム、起動セクション

機能

NC と PLC の同期は起動時におこなわれます。マシンデータ内で定義された NC 設定に基づいて、NC/PLC ユーザーインターフェース用のデータブロックが作成され、基本プログラムのパラメータの妥当性が検証されます。異常が発生した場合は、FB1 が異常識別子を診断バッファに渡し、PLC を停止状態に切り替えます。

「再起動」スタートアップモード

内蔵 PLC は、スタートアップモード「再起動」のみをサポートしています。基本単位系を初期化した後は、オーガニゼーションブロック OB100「再起動」が常に最初に実行され、その後に OB1「サイクリックモード」が続きます。

入力パラメータ

基本プログラムのパラメータ設定では、ファンクションブロック FB1 で該当する各パラメータのみを、ユーザー別の値とともに書き込んでください。FB1 のインスタンスデータブロック DB7 にあるプリセット値を割り当てる必要はありません。ファンクションブロック FB1 は、オーガニゼーションブロック OB100 でのみ呼び出してください。

出力パラメータ

ファンクションブロック FB1 の出力パラメータも、基本プログラムの周期的な部分から読み取ることができます。このために、以下の 2 つのオプションが使用可能です。

1. FB1 のインスタンスデータブロック DB7 へのシンボル形式での直接アクセス。
   例:「L gp_par.MaxChan」(「gp_par」は、DB7 のシンボル名称)
2. ビットメモリが、FB1 のパラメータ設定時に出力パラメータに割り当てられます。そして、ビットメモリが基本プログラムに読み取られ、出力パラメータの値が決まります。
   例:「MaxChan」:= MW 20

注記

MCP および BHG への FB1 パラメータの割り当てについては、「機械操作パネル、ハンドヘルドユニット、ダイレクトキーの構成 (ページ 1103) ］を参照してください。

ファンクションの宣言

```
FUNCTION_BLOCK FB1
VAR_INPUT
   MCPNum:            INT := 1;            // 0:MCP なし
                                           // 1:1 つの MCP (初期設定)
                                           //2:2 つの MCP
   MCP1In:            POINTER;             //MCP1 の入力信号用スタートアドレス
   MCP1Out:           POINTER;             //MCP1 の出力信号用スタートアドレス
   MCP1StatSend:      POINTER;             // MCP1 の送信用状態 DW
   MCP1StatRec:       POINTER;             // MCP1 の受信用状態 DW
   MCP1BusAdr:        INT := 6;            //初期設定
   MCP1Timeout:       S5TIME:= S5T#700MS;
   MCP1Cycl:          S5TIME:= S5T#200MS;
   MCP2In:            POINTER;             //MCP2 の入力信号用スタートアドレス
   MCP2Out:           POINTER;             //MCP2 の出力信号用スタートアドレス
   MCP2StatSend:      POINTER;             // MCP2 の送信用状態 DW
   MCP2StatRec:       POINTER;             // MCP2 の受信用状態 DW
   MCP2BusAdr:        INT;
   MCP2Timeout:       S5TIME:= S5T#700MS;
   MCP2Cycl:          S5TIME:= S5T#200MS;
```

```
MCPMPI:            BOOL := FALSE;
MCP1Stop:          BOOL := FALSE;
MCP2Stop:          BOOL := FALSE;
MCP1NotSend:       BOOL := FALSE;
MCP2NotSend:       BOOL := FALSE;
MCPSDB210:         BOOL := FALSE;
MCPCopyDB77:       BOOL := FALSE;
MCPBusType:        BYTE=B#16#0;
HHU:               INT := 0;           //ハンドヘルドユニットインタフェース
                                       // 0:HHU なし
                                       // 1:HHU (MPI)
                                       // 2:HHU (OPI)
BHGIn:             POINTER;            //ハンドヘルドユニットの送信データ
BHGOut:            POINTER;            //ハンドヘルドユニットの受信データ
BHGStatSend:       POINTER;            //HHU の送信用状態 DW
BHGStatRec:        POINTER;            //HHU の受信用状態 DW
BHGInLen:          BYTE:= B#16#6;      //6 バイト入力
BHGOutLen:         BYTE:= B#16#14;     //20 バイト出力
BHGTimeout:        S5TIME:= S5T#700MS;
BHGCycl:           S5TIME:= S5T#100MS;
BHGRecGDNo:        INT := 2;
BHGRecGBZNo:       INT := 2;
BHGRecObjNo:       INT := 1;
BHGSendGDNo:       INT := 2;
BHGSendGBZNo:      INT := 1;
BHGSendObjNo:      INT := 1;
BHGMPI:            BOOL := FALSE;
BHGStop:           BOOL := FALSE;
BHGNotSend:        BOOL := FALSE;
NCCyclTimeout:     S5TIME:= S5T#200MS;
NCRunupTimeout:    S5TIME:= S5T#50S;
ListMDecGrp:       INT := 0;
NCKomm:            BOOL := FALSE;
MMCToIF:           BOOL:= TRUE;
HWheelMMC:         BOOL:= TRUE;        //HMI を介してのハンドル選択
ExtendAlMsg :      BOOL;
MCP_IF_TCS:        BOOL;
ExtendChanAxMsg:   BOOL;
MsgUser:           INT := 10;          //DB 2 のユーザー領域数
UserIR:            BOOL := FALSE;      // OB40 のユーザープログラム、
                                       //ローカルデータ拡張に注意
IRAuxfuT:          BOOL := FALSE;      // OB40 の T 機能の評価
```

```
   IRAuxfuH:          BOOL := FALSE;       // OB40 の H 機能の評価
   IRAuxfuE:          BOOL := FALSE;       // OB40 の DL 機能の評価
   UserVersion:       POINTER;             //ユーザーインターフェースのバージョン画
                                           面に表示される文字列変数へのポインタ
   OpKeyNum :         INT;
   Op1KeyIn           POINTER;
   Op1KeyOut :        POINTER;
   Op1KeyBusAdr :     INT;
   Op2KeyIn :         POINTER;
   Op2KeyOut :        POINTER;
   Op2KeyBusAdr :     INT;
   Op1KeyStop :       BOOL;
   Op2KeyStop :       BOOL;
   Op1KeyNotSend :    BOOL;
   Op2KeyNotSend :    BOOL;
   OpKeyBusType :     BYTE;
   IdentMcpBusAdr :   INT;
   IdentMcpProfilNo :BYTE;
   IdentMcpBusType :  BYTE;
   IdentMcpStrobe :   BOOL;
END_VAR

VAR_OUTPUT
   MaxBAG:            INT;
   MaxChan:           INT;
   MaxAxis:           INT;
   ActivChan:         ARRAY [1..10] OF
                      BOOL;
   ActivAxis:         ARRAY[1..31] OF
                      BOOL;
   UDInt :            INT;
   UDHex:             INT;
   UDReal :           INT;
   IdentMcpType :     BYTE;
   IdentMcpLengthIn :BYTE;
   IdentMcpLengthOut  BYTE;
   :
END_VAR
```

仮パラメータの説明

信号	タイプ	タイプ	値の範囲	意味
MCPNum:	I	INT	0, 1, 2	有効な MCP の数 0:使用可能な MCP なし
MCP1In: MCP2In:	I	POINTER	E0.0 ... E120.0 または M0.0 ... M248.0 または DBn DBX0.0 ... DBXm.0	関連する機械操作パネルの入力信号用スタートアドレス
MCP1Out: MCP2Out:	I	POINTER	A0.0 ... A120.0 または M0.0 ... M248.0 または DBn DBX0.0 ... DBXm.0	関連する機械操作パネルの出力信号用スタートアドレス
MCP1StatSend: MCP2StatSend:	I	POINTER	A0.0 ... A124.0 または M0.0 ... M252.0 または DBn DBX0.0 ... DBXm.0	Ethernet MCP の場合のみ 点滅の停止(「機械操作パネル、ハンドヘルドユニットの切り替え (ページ 1115)」の章を参照)
MCP1StatRec: MCP2StatRec:	I	POINTER	A0.0 ... A124.0 または M0.0 ... M252.0 または DBn DBX0.0 ... DBXm.0	現在、割り当てはありません。
MCP1BusAdr:	I	INT	1, 2, 3 ... 126	DP スレーブ:PROFIBUS アドレス
MCP2BusAdr:			192, 193, 194 ..223	Ethernet MCP:DIP 設定
MCP1Timeout: MCP2Timeout:	I	S5time	推奨値:700 ms	機械操作パネル用周期的ライフサインモニタ
MCP1Cycl: MCP2Cycl:	I	S5time	推奨値:200 ms	PROFIBUS にのみ関連
MCPMPI:	I	BOOL	FALSE	互換性のために使用可能

信号	タイプ	タイプ	値の範囲	意味
MCP1Stop: MCP2Stop:	I	BOOL	0 (FALSE)、1 (TRUE)	0:機械操作パネル信号の伝送を開始 1:機械操作パネル信号の伝送を停止
				DP スレーブ:スレーブ無効
MCP1NotSend : MCP2NotSend:	I	BOOL	0 (FALSE)、1 (TRUE)	0:送受信操作有効 1:機械操作パネル信号のみを受信
MCPSDB210:	I	BOOL	false	互換性のために使用可能
MCPCopyDB77:	I	BOOL	false	互換性のために使用可能
MCPBusType:	I	BYTE	3, 4, 5, 6	b#16#yx: ● バスタイプ MCP1:下位ニブル(x) ● バスタイプ MCP2:上位ニブル(y) 3:PROFIBUS 4:MPI/DP ポートの PROFIBUS 5:Ethernet 6:PROFINET 混合モードが可能(「機械操作パネル、ハンドヘルドユニット、ダイレクトキーの構成 (ページ 1103)」の章を参照)
HHU:	I	INT	0, 5	ハンドヘルドユニットインタフェース 0:HHU なし 5:HHU (Ethernet)
BHGIn:	I	POINTER	E0.0 ... E124.0 または M0.0 ... M252.0 または DBn DBX0.0 ... DBXm.0	スタートアドレス。HHU から PLC がデータを受信
BHGOut:	I	POINTER	A0.0 ... A124.0 または M0.0 ... M252.0 または DBn DBX0.0 ... DBXm.0	スタートアドレス。HHU へ PLC がデータを送信

信号	タイプ	タイプ	値の範囲	意味
BHGStatSend:	I	POINTER	A0.0 ... A124.0 または M0.0 ... M252.0 または DBn DBX0.0 ... DBXm.0	互換性のために使用可能
BHGStatRec:	I	POINTER	A0.0 ... A124.0 または M0.0 ... M252.0 または DBn DBX0.0 ... DBXm.0	互換性のために使用可能
BHGInLen:	I	BYTE	HHU 初期設定: B#16#6 (6 バイト)	互換性のために使用可能
BHGOutLen:	I	BYTE	HHU 初期設定: B#16#14 (20 バイト)	互換性のために使用可能
BHGTimeout:	I	S5time	推奨値:700 ms	互換性のために使用可能
BHGCycl:	I	S5time	推奨値:100 ms	互換性のために使用可能
BHGRecGDNo:	I	INT	HHU 初期設定:2	Ethernet DIP スイッチ
BHGRecGBZNo:	I	INT	HHU 初期設定:2	互換性のために使用可能
BHGRecObjNo:	I	INT	HHU 初期設定:1	互換性のために使用可能
BHGSendGDNo:	I	INT	HHU 初期設定:2	互換性のために使用可能
BHGSendGBZNo:	I	INT	HHU 初期設定:1	互換性のために使用可能
BHGSendObjNo:	I	INT	HHU 初期設定:1	互換性のために使用可能
BHGMPI:	I	BOOL	0 (FALSE)	互換性のために使用可能
BHGStop:	I	BOOL	0 (FALSE)、1 (TRUE)	0:ハンドヘルドユニット信号の伝送を開始 1:ハンドヘルドユニット信号の伝送を停止
BHGNotSend:	I	BOOL	0 (FALSE)、1 (TRUE)	0:送受信操作有効 1:ハンドヘルドユニット信号のみを受信
NCCyclTimeout:	I	S5time	推奨値:200 ms	周期的ライフサインモニタ、NC
NCRunupTimeout:	I	S5time	推奨値:50s	電源投入監視 NC

信号	タイプ	タイプ	値の範囲	意味
ListMDecGrp:	I	INT	0, 1, 2 ... 16	拡張 M グループデコーディングの起動 0:無効 1 ～ 16:M グループの数
NCKomm:	I	BOOL	0 (FALSE)、1 (TRUE)	PLC NC 通信サービス FB2、3、4、5、7 1:有効
MMCToIF:	I	BOOL	0 (FALSE)、1 (TRUE)	インターフェースへの HMI 信号の伝送(モード、プログラム制御など) 1:有効
HWheelMMC:	I	BOOL	0 (FALSE)、1 (TRUE)	0:HMI を介してのハンドル選択 1:ユーザープログラムを介してのハンドル選択
ExtendAlMsg :	I	BOOL	0 (FALSE)、1 (TRUE)	拡張の起動、FC10 のエラーメッセージおよび操作メッセージ(「PLC/HMI インタフェース (ページ 1073)」の章の「ブロック FC10 を介した PLC アラームの拡張」の章を参照)
MCP_IF_TCS	I	BOOL	0 (FALSE)、1 (TRUE)	FC19:評価 R11 キー 0:評価なし 1:R11 キーは、工具オリエンテーションで手動運転として機能します(「FC19:MCP_IFM - MCP 信号のインターフェースへの伝送 (ページ 1314)」の章の「直交座標ジョグ移動」の章を参照)
ExtendChanAxMsg	I	BOOL	0 (FALSE)、1 (TRUE)	FC10 のエラーメッセージおよび操作メッセージのすべての領域の起動(「FC10:AL_MSG - 異常メッセージおよび操作メッセージ (ページ 1287)」の章を参照)
MsgUser:	I	INT	0, 1, 2 ... 64	ユーザー領域の数(DB2) –パラメータ「ExtendAlMsg」によって異なります

信号	タイプ	タイプ	値の範囲	意味
UserIR:	I	BOOL	0 (FALSE)、1 (TRUE)	ユーザーからの信号の処理に必要なローカルデータ拡張 OB40
IRAuxfuT:	I	BOOL	0 (FALSE)、1 (TRUE)	OB40 の T 機能の評価
IRAuxfuH:	I	BOOL	0 (FALSE)、1 (TRUE)	OB40 の H 機能の評価
IRAuxfuE:	I	BOOL	0 (FALSE)、1 (TRUE)	OB40 の DL 機能の評価
UserVersion:	I	POINTER	P#DBn.DBXx.0	ユーザーインターフェースのバージョン画面に表示される文字列変数へのポインタ 文字列変数には、以下の表記(最大41文字)が含まれます。 「**<名前> <**バージョン **xx.xx.xx> <**日付 **yy/mm/dd>**」 例:「テストバージョン 07.06.02 13/06/04」
OpKeyNum :	I	INT	0, 1, 2	有効なダイレクトキーモジュールの数 0:使用可能な Ethernet ダイレクトキーなし
Op1KeyIn: Op2KeyIn:	I	POINTER	P#Ex.0 または P#Mx.0 または P#DBn.DBXx.0	影響を受けるダイレクトキーモジュールの入力信号用スタートアドレス
Op1KeyOut : Op2KeyOut :	I	POINTER	P#Ax.0 または P#Mx.0 または P#DBn.DBXx.0	影響を受けるダイレクトキーモジュールの出力信号用スタートアドレス
Op1KeyBusAdr : Op2KeyBusAdr :	I	INT	1, 2, 3 ... 191	Ethernet を介したダイレクトキー:TCU インデックス:
Op1KeyStop : Op2KeyStop :	I	BOOL	0 (FALSE)、1 (TRUE)	0:ダイレクトキー信号の伝送を開始 1:ダイレクトキー信号の伝送を停止

信号	タイプ	タイプ	値の範囲	意味
Op1KeyNotSend : Op2KeyNotSend :	I	BOOL	0 (FALSE)、1 (TRUE)	0:送受信操作有効 1:ダイレクトキー信号のみを受信
OpKeyBusType :	I	BYTE	b#16#55	Ethernet
IdentMcpBusAdr :	I	INT	1, 2, 3 ... 254	IE 機器のみ
IdentMcpProfilNo :	I	BYTE	0, 1	機器のプロファイル 0:すべての機器 1:ダイレクトキーのみ
IdentMcpBusType :	I	BYTE	b#16#5	IE 機器のみ
IdentMcpStrobe :	I	BOOL	0 (FALSE)、1 (TRUE)	1:問い合わせを起動
MaxBAG:	O	INT	1, 2, 3 ... 10	モードグループの数
MaxChan:	O	INT	1, 2, 3 ... 10	チャネル数
MaxAxis:	O	INT	1, 2, 3 ... 31	軸数
ActivChan:	O	ARRAY[1...10] OF BOOL	0 (FALSE)、1 (TRUE)	有効なチャネルのビット文字列
ActivAxis:	O	ARRAY[1..31] OF BOOL	0 (FALSE)、1 (TRUE)	有効な軸のビット文字列
UDInt :	O	INT	0,	DB20 の整数マシンデータの数
UDHex:	O	INT	---	DB20 の 16 進数マシンデータの数
UDReal :	O	INT	---	DB20 の REAL マシンデータの数
IdentMcpType :	O	BYTE	0、B#16#80、B#16#81、B#16#82 ... B#16#89	MCP タイプ
IdentMcpLengthIn :	O	BYTE	---	MCP 入力データの長さ(MCP→PLC)
IdentMcpLengthOut :	O	BYTE	---	MCP 出力データの長さ(PLC→MCP)

機械操作パネル(MCP)またはハンドヘルドユニット(HHU)で通信中に異常が派生した場合は、以下のアラームが HMI に表示され、入力信号(MCP1In、MCP2In、または BHGIn)が値 0 に設定されます。

異常時:MCP/HHU

- アラーム 400260:「MCP 1 故障」
- アラーム 400261:「MCP 2 故障」
- アラーム 400262:「HHU 故障」

PLC と MCP/HHU の間で再原点同期が可能な場合は、通信が再開され、HMI 上のエラーメッセージが基本プログラムによって削除されます。また、プロセス値が入力信号(MCP1In、MCP2In、または BHGIn)に再度伝送されます。

例:OB100 の FB1 呼び出し:

```
ORGANIZATION_BLOCK OB100
VAR_TEMP
    OB100_EV_CLASS :              BYTE;
    OB100_STRTUP :                BYTE;
    OB100_PRIORITY :              BYTE;
    OB100_OB_NUMBR :              BYTE;
    OB100_RESERVED_1 :            BYTE;
    OB100_RESERVED_2 :            BYTE;
    OB100_STOP :                  WORD;
    OB100_RESERVED_3 :            WORD;
    OB100_RESERVED_4 :            WORD;
    OB100_DATE_TIME :             DATE_AND_TIME;
END_VAR
BEGIN
    CALL FB1, DB7(                // FB1 呼び出し、インスタンス DB:DB7
          MCPNum :=               1,
          MCP1In :=               P#E0.0,
          MCP1Out :=              P#A0.0,
          MCP1StatSend :=         P#A8.0,
          MCP1StatRec :=          P#A12.0,
          MCP1BusAdr :=           6,
          MCP1Timeout :=          S5T#700MS,
          MCP1Cycl :=             S5T#200MS,
          NC-CyclTimeout :=       S5T#200MS,
          NC-RunupTimeout :=      S5T#50S);
//INSERT USER PROGRAM HERE
END_ORGANIZATION_BLOCK
```

14.17.2 FB2:GET - NC 変数の読み取り

機能

FB2「GET」ファンクションブロックを使用して、NC 領域から変数が読み取られます。

NC 変数を参照するために、まず、NC 変数が「NC VAR セレクタ」ツールを使用して選択され、次に、データブロック内に STL ソースとして生成されます。その後、S7 シンボルテーブル内のこのデータブロックに名前を割り当ててください。FB2 の呼び出し時に変数アドレスが、以下の形式で伝送されます。パラメータ「Addr1」～「Addr8」=「<DB 名>.<S7 名>」

NC 変数の読み取り要求

立ち上がりエッジによる FB2 の呼び出し、変更、パラメータ「Req」= 0→1

NC 変数の S7 名:パラメータ「Addr1」～「Addr8」=「NCVAR.<S7 名>」

変数値を書き込むためのポインタ:パラメータ「RD1」～「RD8」=「P#<アドレス>」

読み取り要求の完了

読み取り要求が正常に終了:パラメータ「Done」== 1.

読み取り要求が異常とともに終了:パラメータ「Error」== 1、異常の原因はパラメータ「State」

前提条件

- OB100 による NC/PLC 通信の解放、FB1 パラメータ「NCKomm」= 1
- データブロック DB120 (データインターフェース)の場合は、S7 シンボルエディタを使用して、シンボル (初期設定:NCVAR)を S7 プロジェクトの S7 シンボルテーブルに割り当ててください。このシンボルを介して、NC 変数を FB2 パラメータ「Addr<x>」で指定します(例:"ADDR1": = "NCVAR".<NC 変数>"

一般条件

- FB2 には、マルチインスタンス機能があります。
- FB2 呼び出し毎に、ユーザー領域から個別のインスタンス DB を割り当ててください。
- **チャネル別**変数が読み取られると、FB2 を呼び出した場合は、ただ **1 つ**のチャネルからの変数のみが、「Addr1」～「Addr8」を介してアドレス設定されます。

- **ドライブ別**変数が読み取られると、FB2 を呼び出した場合は、ただ **1 つ**の SERVO ドライブオブジェクトからの変数のみが、「Addr1」～「Addr8」を介してアドレス設定されます。サーボドライブオブジェクトは、NC の機械軸に割り当ててください。行インデックスは、論理ドライブ番号に対応しています。
- 読み取りジョブでは、同じ領域、チャネル、またはドライブオブジェクトからの変数のみを読み取ることができます。

注記

異常時
さまざまなチャネルまたはドライブオブジェクトからの変数の読み取り時に、またはチャネルとドライブオブジェクトから同時に、エラーメッセージが出力されます。

- 「Error」 == TRUE
- 「State」 == W#16#02

可変アドレス指定

NC 変数の中には、NC VAR セレクタで「Area no.」や「Line」または「Column」を選択することが必要なものがあります。基本タイプを選択できます。つまり、「Area no.」、「Line」、「Column」には、「0」が事前に割り当てられます。NC VAR セレクタで指定される「Area no.」、「Line」、および「Column」の値が FB で「0」になっているかどうかがチェックされます。NC-VAR セレクタ値==「0」の場合は、対応する FB2 パラメータ値が使用されます。これをおこなうには、**<x>** = 1 ～ 8 の FB2 パラメータ「Unit**<x>**」、「Column**<x>**」、および「Line**<x>**」を、FB2 の呼び出し前に書き込んでください。

表 14-1　　一致するパラメータ

FB2 パラメータ	**NC VAR セレクタ**
単位	Area no.
列	列
行	行

1 つのグループ内の変数を 1 つのジョブに結合することができます。

グループ	領域				
1	C[1]	N	B	O	T
2	C[2]	N	B	O	T

グループ	領域				
3	V[.]	H[.]	---	---	---
例で示したグループ 1 とグループ 2 の場合と同じ規則が、チャネル 3 ～ 10 に適用されます。					

注記

「文字列」タイプの複数の変数を同時に読み取るときに、使用可能な変数の数は 8 未満となります。

ファンクションの宣言

```
FUNCTION_BLOCK FB2
VAR_INPUT
```

```
    Req :                BOOL;
    NumVar :             INT;
    Addr1 :              ANY ;
    Unit1 :              BYTE;
    Column1 :            WORD;
    Line1 :              WORD;
    Addr2 :              ANY ;
    Unit2 :              BYTE;
    Column2 :            WORD;
    Line2 :              WORD;
    Addr3 :              ANY ;
    Unit3 :              BYTE;
    Column3 :            WORD;
    Line3 :              WORD;
    Addr4 :              ANY ;
    Unit4 :              BYTE;
    Column4 :            WORD;
    Line4 :              WORD;
    Addr5 :              ANY ;
    Unit5 :              BYTE;
    Column5 :            WORD;
    Line5 :              WORD;
    Addr6 :              ANY ;
    Unit6 :              BYTE;
    Column6 :            WORD;
    Line6 :              WORD;
    Addr7 :              ANY ;
    Unit7 :              BYTE;
    Column7 :            WORD;
    Line7 :              WORD;
    Addr8 :              ANY ;
    Unit8 :              BYTE;
    Column8 :            WORD;
    Line8 :              WORD;
END_VAR
VAR_OUTPUT
    Error :              BOOL;
    NDR :                BOOL;
    State :              WORD;
END_VAR
```

```
VAR_IN_OUT
    RD1 :                ANY ;
    RD2 :                ANY ;
    RD3 :                ANY ;
    RD4 :                ANY ;
    RD5 :                ANY ;
    RD6 :                ANY ;
    RD7 :                ANY ;
    RD8 :                ANY ;
END_VAR
```

仮パラメータの説明

パラメータ	タイプ	タイプ	値の範囲	説明
Req :	I	BOOL	---	信号の立ち上がりエッジを使用してジョブが起動します
NumVar :	I	INT	1 ... 8	読み取る変数の数:「Addr1」～「Addr8」
Addr1 ～ Addr8:	I	ANY	「DBName」.<VarName>	NC Var セレクタからの変数識別子
Unit1 ～ Unit8:	I	BYTE	---	領域アドレス、可変アドレス指定のオプション
Column1 ～ Column8:	I	WORD	---	列アドレス、可変アドレス指定のオプション
Line1 ～ Line8:	I	WORD	---	行アドレス、可変アドレス指定のオプション
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	1:ジョブの否定応答、またはジョブの実行不可
NDR :	O	BOOL	0 (FALSE)、1 (TRUE)	1:ジョブは正常終了しました。データを使用できます
State :	O	WORD	---	段落「エラー ID」を参照
RD1 ～ RD8:	I/O	ANY	P#Mm.n BYTE x... P#DBnr.dbxm.n BYTE x	読み取りデータの対象領域

異常識別子

NC 変数

State		説明	注
上位バイト 1)	下位バイト		
1 - 8	1	アクセス異常	---
0	2	ジョブ内の異常	ジョブ内の変数の間違ったコンパイル
0	3	否定応答、ジョブ実行不可	内部エラー、考えられる対応: ● ジョブデータのチェック ● NC リセット
1 - 8	4	空きローカルユーザーメモリが不足	読み取り変数のデータタイプが、「RD1」～「RD8」で指定された値を超えている
0	5	フォーマット変換異常	Double タイプの変数の変換時の異常。変数が S7 REAL 領域内にありません
0	6	FIFO が一杯	待ち行列が一杯になったため、ジョブを繰り返してください
0	7	オプション設定なし	FB1 パラメータ「NCKomm」が未設定
1 - 8	8	間違った対象領域(RD)	「RD1」～「RD8」をローカルデータにすることはできません
0	9	転送が占有されています	ジョブを繰り返してください
1 - 8	10	可変アドレス指定の異常	「Unit」または「Column」/「Line」に値 0 が含まれています
0	11	変数のアドレスが無効	「Addr」 (または変数名称)、「Area」、「Unit」をチェック
0	12	NumVar == 0	パラメータ「NumVar」をチェック
1 - 8	13	間違った ANY データ参照	要求された NcVar データがパラメータ設定されていない
1) 上位バイト> 0 ⇒異常が発生した変数の番号			

ドライブ別変数

ドライブ別変数(DB1200.DBX3000.1 == 1)の読み取り/書き込み中に異常が発生した場合は、アクセス結果に異常番号が表示されます。この番号は、PROFIdrive プロファイルで定義された異常番号に基づいています。

状態	説明
X	<PROFIdrive プロファイルの異常番号> + 20_H または 36_D

アクセス結果の意味の特定:

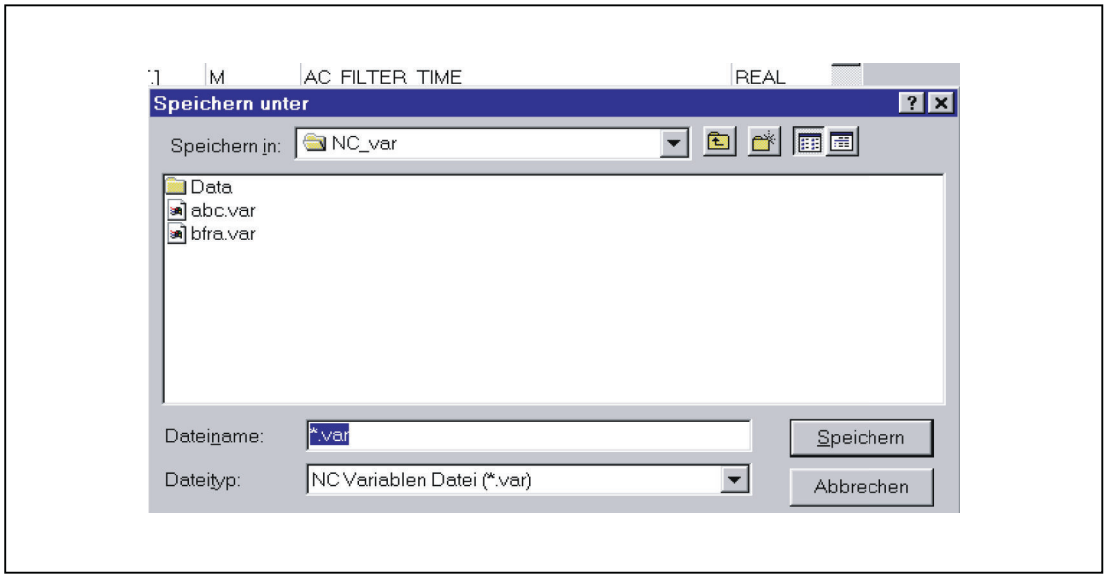
1. PROFIdrive プロファイルの異常番号の計算
   <PROFIdrive プロファイルの異常番号> =アクセスの結果- 20_H または 36_D
2. PROFIdrive プロファイルの異常番号の意味の特定
   PROFIdrive プロファイルの異常番号の説明は、以下の資料にあります。
   参照先
   『SINAMICS S120 ドライブの機能、機能マニュアル』、「通信」の章の「PROFIdrive による通信」の「非周期通信」の「要求と応答の構造」の「パラメータ応答の異常値」の段落

設定手順

NC 変数の読み取り実施手順:

- NC VAR セレクタを使用して変数を選択
- *.VAR ファイルに、選択した変数を保存
- STEP 7 *.STL ソースファイルを生成
- 関連するアドレスデータを持つ DB を生成
- ユーザープログラムでシンボルを使用してアドレスパラメータにアクセスできるように、シンボルテーブルに、生成した DB のシンボルを入力
- FB2 のパラメータ設定

タイミングチャート



① User:要求の設定、Req = 0→1

② FB2 が正常に終了、NDR = 1

User:要求のリセット、IF NDR == 1 THEN Req = 0

③ User:IF NDR == 1 THEN、要求のリセット:1 → 0

④ FB2 がジョブ確認のリセット、NDR = 0

⑤ User:IF NDR == 0 AND Error == 0 THEN、要求のリセット、Req = 1 → 0 **許容されません**

⑥ FB2 が異常終了、Error = 1

User:要求のリセット、IF NDR == 1 OR Error == 1 THEN Req = 0、さらにエラー処理の可能性

呼び出し例

チャネル 1 からの 3 つのチャネルマシンデータの読み取ると、そのアドレス指定が DB120 に格納されます。

データの指定

データは、NC VAR セレクタで選択し、DB120.**VAR** ファイルに格納します。このファイルから DB120.**AWL** ファイルを作成します。

S7 (ALIAS)名を選択します。

チャネル名称を変数名に使用し、STEP 7 シンボルには使用できない文字「[」および「]」を削除するために、新しい S7 名を選択します。

領域	ブロック	名称	タイプ	番号	バイト	S7 の名称
C[1]	M	MD20070 $MC_AXCONF_MACHAX_USED[1]	CHAR	20070	1	C1AxConfMachAxUsed1
C[1]	M	MD20070 $MC_AXCONF_MACHAX_USED[2]	CHAR	20070	1	C1AxConfMachAxUsed2
C[1]	M	MD20090 $MC_SPIND_DEF_MASTER_SPIND	INT	20090	1	C1SpindDefMasterSpind

S7 シンボルテーブル

「NCVAR」を、データブロック DB120 のシンボル名称として S7 シンボルテーブルに入力します。

シンボル	オペランド	データタイプ
NCVAR	DB120	DB120

ファイル DB120.AWL をコンパイルして PLC に伝送してください。

インスタンス DB110 による FB2 のパラメータ設定

```
DATA_BLOCK DB110          // FB2 のインスタンスとして、未割り当てユーザー DB
FB2
BEGIN
END_DATA_BLOCK
Function FC "VariablenCall" :VOID
```

```
U      I 7.7;                    //未割り当て機械操作パネルキー
S      M 100.0;                  //要求の起動
U      M 100.1;                  //NDR 完了メッセージ
R      M 100.0;                  //ジョブの終了
U      I 7.6;                    //Manual error acknowledgment
U      M 102.0;                  //異常未処置
R      M 100.0;                  //ジョブの終了
CALL FB2, DB110(
   Req :           M 100.0,
   NumVar :        3,            //3 つの変数の読み取り
   Addr1 :         "NCVAR".C1AxConfMachAxUsed1,
   Addr2 :         "NCVAR".C1AxConfMachAxUsed2,
   Addr3 :         "NCVAR".C1SpindDefMasterSpind,
   Error :         M102.0,
   NDR :           M100.1,
   State :         MW104,
   RD1 :           P#DB99.DBX0.0 BYTE 1,
   RD2 :           P#DB99.DBX1.0 BYTE 1,
   RD3 :           P#M110.0 INT 1);
```

例:可変アドレス指定

チャネル 1 の 2 つの R パラメータを読み取ると、そのアドレス指定が基本タイプとして DB120 に設定されます。R パラメータ番号は、パラメータ「<Linex>」を使用してパラメータ設定されます。

```
DATA_BLOCK DB120
VERSION :0.0
STRUCT
   C1_RP_rpa0_0:
   STRUCT
   SYNTAX_ID :               BYTE := B#16#82;
   area_and_unit :           BYTE := B#16#41;
   column :                  WORD := W#16#1;
   line :                    WORD := W#16#0;
   block type :              BYTE := B#16#15;
   NO.OF LINES :             BYTE := B#16#1;
   type :                    BYTE := B#16#F;
   length :                  BYTE := B#16#8;
   END_STRUCT;
END_STRUCT;
BEGIN
```

```
END_DATA_BLOCK
CALL FB2, DB110(
   Req :=                                M 0.0,
   NumVar :=                             2,
   Addr1 :=                              "NCVAR".C1_RP_rpa0_0,
   Line1 :=                              W#16#1,
   Addr2 :=                              "NCVAR".C1_RP_rpa0_0,
   Line2 :=                              W#16#2,
   Error :=                              M 1.0,
   NDR :=                                M 1.1,
   State :=                              MW 2,
   RD1 :=                                P#M 4.0 REAL 1,
   RD2 :=                                P#M 24.0 REAL 1);
```

データタイプの割り当て

表 14-2　　データタイプの割り当て

NC 内部または BTSS データタイプ	S7 データタイプ
DOUBLE	REAL
DOUBLE	REAL2
REAL	REAL
LONG	DINT
INTEGER	DINT
UINT_32	DWORD
int_16	INT
UINT_16	WORD
UNSIGNED	WORD
CHAR	CHAR または BYTE
STRING	STRING
BOOL	BOOL
DATETIME	DATE_AND_TIME

例

たとえば、タイプ DOUBLE の NC 変数を、フォーマットを調整せずに読み取ることができるようにするには、REAL2 タイプの任意の ANY ポインタを宛先エリア「RDx」に指定してください(例:P#M100.0 REAL2)。「double」タイプの変数の読み取り時に基本プロ

グラムが REAL2 をターゲットタイプと認識した場合は、このデータが、64 ビットの浮動小数点数として PLC データ領域に伝送されます。

14.17.3 FB3:PUT - NC 変数の書き込み

機能

FB3「PUT」ファンクションブロックを使用して、NC 領域から変数が書き込まれます。

NC 変数を参照するために、まず、NC 変数が「NC VAR セレクタ」ツールを使用して選択され、次に、データブロック内に STL ソースとして生成されます。その後、S7 シンボルテーブル内のこのデータブロックに名前を割り当ててください。FB3 の呼び出し時に変数アドレスが、以下の形式で伝送されます。パラメータ「Addr1」～「Addr8」=「<DB 名>」.<S7 名>

NC 変数の書き込み要求

立ち上がりエッジによる FB3 の呼び出し、変更、パラメータ「Req」= 0→1

NC 変数の S7 名:パラメータ「Addr1」～「Addr8」=「NCVAR.<S7 名>」

変数値を書き込むためのポインタ:パラメータ「RD1」～「RD8」=「P#<アドレス>」

書き込み要求の完了

書き込み要求が正常に終了:パラメータ「Done」== 1.

書き込み要求が異常とともに終了:パラメータ「Error」== 1、異常の原因はパラメータ「State」

前提条件

- OB100 による NC/PLC 通信の解放、FB1 パラメータ「NCKomm」= 1
- データブロック DB120 (データインターフェース)の場合は、S7 シンボルエディタを使用して、シンボル (初期設定:NCVAR)を S7 プロジェクトの S7 シンボルテーブルに割り当ててください。このシンボルを介して、NC 変数を FB3 パラメータ「Addr<x>」で指定します(例:「ADDR1: =「NCVAR」.<NC 変数>」

一般条件

- FB3 には、マルチインスタンス機能があります。
- FB3 呼び出し毎に、ユーザー領域から個別のインスタンス DB を割り当ててください。

- パスワードを使用せずマシンデータと GUD を定義するために、アクセスするデータの保護レベルを最下位レベルに再定義してください。
  参照先:
  – 『試運転マニュアル』; :「保護レベル」の章
  – 『プログラミングマニュアル　上級編』; :「ユーザーデータの保護レベルの定義」の章
- **チャネル別**変数が読み取られると、FB2 を呼び出した場合は、ただ **1 つ**のチャネルからの変数のみが、「Addr1」～「Addr8」を介してアドレス設定されます。
- **ドライブ別**変数が読み取られると、FB2 を呼び出した場合は、ただ **1 つ**の SERVO ドライブオブジェクトからの変数のみが、「Addr1」～「Addr8」を介してアドレス設定されます。サーボドライブオブジェクトは、NC の機械軸に割り当ててください。行インデックスは、論理ドライブ番号に対応しています。
- 書き込みジョブでは、同じ領域、チャネル、またはドライブオブジェクトからの変数のみを書き込むことができます。

注記

異常時
さまざまなチャネルまたはドライブオブジェクトからの変数の書き込み時に、またはチャネルとドライブオブジェクトから同時に、エラーメッセージが出力されます。

- 「Error」 == TRUE
- 「State」 == W#16#02

可変アドレス指定

NC 変数の中には、NC VAR セレクタで「Area no.」や「Line」または「Column」を選択することが必要なものがあります。基本タイプを選択できます。つまり、「Area no.」、「Line」、「Column」には、「0」が事前に割り当てられます。NC VAR セレクタで指定される「Area no.」、「Line」、および「Column」の値が FB で「0」になっているかどうかがチェックされます。NC-VAR セレクタ値==「0」の場合は、対応する FB3 パラメータ値が使用されます。これをおこなうには、**<x>** = 1 ～-8 の FB3 パラメータ「Unit**<x>**」、「Column**<x>**」、および「Line**<x>**」を、FB3 の呼び出し前に書き込んでください。

FB3 パラメータ	NC VAR セレクタ
Unit<x>	Area no.
Column<x>	列
Column<x>	行

1 つのグループ内の NC 変数を 1 つのジョブに結合することができます。

グループ	領域				
1	C[1]	N	B	O	T
2	C[2]	N	B	O	T
3	V[.]	H[.]	---	---	---
例で示したグループ 1 とグループ 2 の場合と同じ規則が、チャネル 3 ～ 10 に適用されます。					

注記

「文字列」タイプの複数の変数を同時に書き込むときに、使用可能な変数の数は 8 未満となります。

ファンクションの宣言

```
FUNCTION_BLOCK FB3
VAR_INPUT
```

```
    Req :               BOOL;
    NumVar :            INT;
    Addr1 :             ANY ;
    Unit1 :             BYTE;
    Column1 :           WORD;
    Line1 :             WORD;
    Addr2 :             ANY ;
    Unit2 :             BYTE;
    Column2 :           WORD;
    Line2 :             WORD;
    Addr3 :             ANY ;
    Unit3 :             BYTE;
    Column3 :           WORD;
    Line3 :             WORD;
    Addr4 :             ANY ;
    Unit4 :             BYTE;
    Column4 :           WORD;
    Line4 :             WORD;
    Addr5 :             ANY ;
    Unit5 :             BYTE;
    Column5 :           WORD;
    Line5 :             WORD;
    Addr6 :             ANY ;
    Unit6 :             BYTE;
    Column6 :           WORD;
    Line6 :             WORD;
    Addr7 :             ANY ;
    Unit7 :             BYTE;
    Column7 :           WORD;
    Line7 :             WORD;
    Addr8 :             ANY ;
    Unit8 :             BYTE;
    Column8 :           WORD;
    Line8 :             WORD;
END_VAR
VAR_OUTPUT
    Error :             BOOL;
    Done :              BOOL;
    State :             WORD;
END_VAR
VAR_IN_OUT
```

```
    SD1 :                ANY ;
    SD2 :                ANY ;
    SD3 :                ANY ;
    SD4 :                ANY ;
    SD5 :                ANY ;
    SD6 :                ANY ;
    SD7 :                ANY ;
    SD8 :                ANY ;
END_VAR
```

仮パラメータの説明

信号	タイプ	タイプ	値の範囲	意味
Req :	I	BOOL	-	信号の立ち上がりエッジを使用してジョブが起動します
NumVar :	I	INT	1 ... 8	書き込む変数の数:「Addr1」～「Addr8」
Addr1 ～ Addr8:	I	ANY	「DBName」.<VarName>	NC Var セレクタからの変数識別子
Unit1 ～ Unit8:	I	BYTE	-	領域アドレス、可変アドレス指定のオプション
Column1 ～ Column8:	I	WORD	-	列アドレス、可変アドレス指定のオプション
Line1 ～ Line8:	I	WORD	-	行アドレス、可変アドレス指定のオプション
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	ジョブの否定応答、またはジョブの実行不可
Done :	O	BOOL	0 (FALSE)、1 (TRUE)	ジョブが正常終了
State :	O	WORD	-	段落「エラー ID」を参照
SD1 ～ SD8	I/O	ANY	P#Mm.n BYTE x... P#DBnr.dbxm.n BYTE x	書き込むデータ

異常識別子

NC 変数

State		意味	注
上位バイト 1)	下位バイト		
1 - 8	1	アクセス異常	---
0	2	ジョブ内の異常	ジョブ内の変数の間違ったコンパイル
0	3	否定応答、ジョブ実行不可	内部異常、 ● ジョブのチェック ● NC リセット
1 - 8	4	データ領域またはデータタイプ不一致、または空の文字列です	「SD1」～「SD8」のデータのチェック
0	6	FIFO が一杯	待ち行列が一杯になったため、ジョブを繰り返してください
0	7	オプション設定なし	FB1 パラメータ「NCKomm」が未設定
1 - 8	8	間違った対象領域(SD)	「SD1」～「SD8」はローカルデータになれません
0	9	転送が占有されています	ジョブを繰り返してください
1 - 8	10	可変アドレス指定の異常	「Unit」または「Column」/「Line」に値 0 が含まれています
0	11	可変アドレスが無効です。または、読み取り専用変数です	「Addr」 (または変数名称)、「Area」、「Unit」をチェック
0	12	NumVar == 0	パラメータ「NumVar」をチェック
1 - 8	13	間違った ANY データ参照	要求された NcVar データがパラメータ設定されていない
1 - 8	15	ユーザーデータが長すぎます	対策:ジョブ毎に書き込む変数の数を減らすか、使用する文字列変数を短くします
1) 上位バイト> 0 ⇒異常が発生した変数の番号			

ドライブ別変数

ドライブ別変数(DB1200.DBX3000.1 == 1)の読み取り/書き込み中に異常が発生した場合は、アクセス結果に異常番号が表示されます。この番号は、PROFIdrive プロファイルで定義された異常番号に基づいています。

状態	意味
X	<PROFIdrive プロファイルの異常番号> + 20_H または 36_D

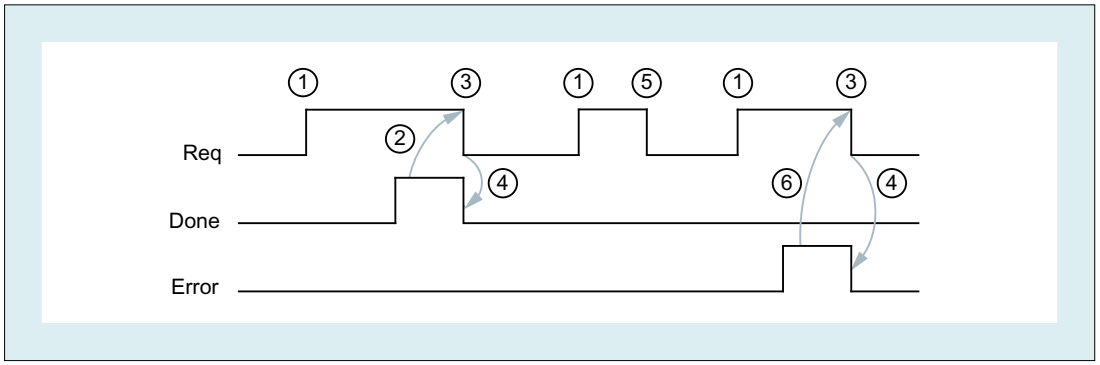
アクセス結果の意味の特定:

1. PROFIdrive プロファイルの異常番号の計算
   <PROFIdrive プロファイルの異常番号> =アクセスの結果- 20_H または 36_D
2. PROFIdrive プロファイルの異常番号の意味の特定
   PROFIdrive プロファイルの異常番号の説明は、以下の資料にあります。
   参照先
   『SINAMICS S120 ドライブの機能、機能マニュアル』、「通信」の章の「PROFIdrive による通信」の「非周期通信」の「要求と応答の構造」の「パラメータ応答の異常値」の段落

設定手順

NC 変数を書き込むには、NC 変数の読み取りの場合と同じ設定手順が必要になります。読み取る NC 変数または書き込む NC 変数すべてのアドレスデータを DB に格納することが適切です。

タイミングチャート



① User:要求の設定、Req = 0→1

② FB3 が正常に終了、Done = 1
User:要求のリセット、IF Done == 1 THEN Req = 0

③ User:IF Done == 1 THEN、要求のリセット:1 → 0

④ FB3 がジョブ確認をリセット、Done = 0

⑤ User:IF Done == 0 AND Error == 0 THEN、要求のリセット、Req = 1 → 0 **許容されません**

⑥ FB3 が異常終了、Error = 1
User:要求のリセット、IF Done == 1 OR Error == 1 THEN Req = 0、さらにエラー処理の可能性

呼び出し例

チャネル 1 の 3 つのチャネルマシンデータ項目の書き込み:

NC VAR セレクタでの 3 つのデータ項目の選択、およびファイル DB120.VAR への格納
ブロック名称を名前に使用し、STEP 7 シンボルには使用できない文字「[ ]」を削除するために、新しい S7 名を選択します。

領域	ブロック	名称	タイプ	バイト	S7 の名称
C[1]	RP	rpa[5]	DOUBLE	4	rpa_5C1RP
C[1]	RP	rpa[11]	DOUBLE	4	rpa_11C1RP
C[1]	RP	rpa[14)	DOUBLE	4	rpa_14C1RP

S7 SYMBOL エディタによる DB120 の NCVAR の入力

シンボル	オペランド	データタイプ
NCVAR	DB120	DB120

ファイル DB120.AWL をコンパイルして PLC に伝送してください。

インスタンス DB111 による FB3 の呼び出しとパラメータ設定

```
DATA_BLOCK DB111          // FB3 のインスタンスとして、未割り当てユーザー DB
FB3
BEGIN
Function FC "VariablenCall" :VOID
END_DATA_BLOCK
   U      I 7.7;          //未割り当て機械操作パネルキー
   S      M 100.0;        //要求の起動
   U      M 100.1;        //Done 完了メッセージ
   R      M 100.0;        //ジョブの終了
   U      I 7.6;          //Manual error acknowledgment
   U      M 102.0;        //異常未処置
   R      M 100.0;        //ジョブの終了
   CALL FB3, DB111(
      Req :              M 100.0,
      NumVar :           3,                        //3 つの変数の書き込み
      Addr1 :            "NCVAR".rpa_5C1RP,
      Addr2 :            "NCVAR".rpa_11C1RP,
      Addr3 :            "NCVAR".rpa_14C1RP,
      Error :            M102.0,
      Done :             M100.1,
      State :            MW104,
      SD1 :              P#DB99.DBX0.0 REAL 1,
      SD2 :              P#DB99.DBX4.0 REAL 1,
      SD3 :              P#M110.0 REAL 1);
```

例:可変アドレス指定

チャネル 1 の 2 つの R パラメータを書き込むと、そのアドレス指定が基本タイプとして DB120 に設定されます。R パラメータ番号は、パラメータ LineX を使用してパラメータ設定されます。

```
DATA_BLOCK DB120
VERSION :0.0
```

```
STRUCT
   C1_RP_rpa0_0:
   STRUCT
   SYNTAX_ID :              BYTE := B#16#82;
   area_and_unit :          BYTE := B#16#41;
   column :                 WORD := W#16#1;
   line :                   WORD := W#16#0;
   block type :             BYTE := B#16#15;
   NO.OF LINES :            BYTE := B#16#1;
   type :                   BYTE := B#16#F;
   length :                 BYTE := B#16#8;
   END_STRUCT;
END_STRUCT;
BEGIN
END_DATA_BLOCK
CALL FB3, DB122(
   Req :=                   M 10.0,
   NumVar :=                2,
   Addr1 :=                 "NCVAR".C1_RP_rpa0_0,
   Line1 :=                 W#16#1,
   Addr2 :=                 "NCVAR".C1_RP_rpa0_0,
   Line3 :=                 W#16#2
   Error :=                 M 11.0,
   Done :=                  M 11.1,
   State :=                 MW 12,
   SD1 :=                   P#M 4.0 REAL 1,
   SD2 :=                   P#M 24.0 REAL 1);
```

データタイプの割り当て

「FB2:GET - NC 変数の読み取り (ページ 1173)」の章の「データタイプの割り当て」にある表を参照してください

14.17.4 FB4:PI_SERV - PI サービス要求

機能

ファンクションブロック FB4「PI_SERV」を使用して PI サービスを開始します。

使用可能な PI サービスについては、その特定のパラメータとともに、以下の章に説明があります。使用可能な PI サービスの概要は、「使用可能な PI サービスのリスト (ページ 1199)」にあります 。

注記

多数の「WVar」パラメータがあるため、ファンクションブロック FB4 ではなく、FB7 を使用することを推奨します。「FB7:PI_SERV2 - PI サービス要求 (ページ 1242)」の章を参照してください。

PI サービスの開始

PI サービス開始の要求:立ち上がりエッジによる FB4 の呼び出し、変更、パラメータ「Req」= 0→1

PI サービスの完了

ジョブまたは PI サービスが正常に終了:パラメータ「Done」== 1.

ジョブまたは PI サービスが異常とともに終了:パラメータ「Error」== 1、異常の原因はパラメータ「State」

前提条件

- OB100 による NC/PLC 通信の解放、FB1 パラメータ「NCKomm」= 1
- データブロック DB16 (PI サービスのデータインターフェース)の場合は、S7 シンボルエディタを使用して、シンボル (初期設定:PI)を S7 プロジェクトの S7 シンボルテーブルに割り当ててください。これで、要求された PI サービスが、このシンボルを使用して FB4 パラメータ「PIService」で指定されます(例:「PIService:= "PI".<PI サービス>」)。

一般条件

- FB4 呼び出し毎に、ユーザー領域から個別のインスタンス DB を割り当ててください。
- PI サービス開始(「Req」= 1 での FB4 呼び出し)は、PLC 基本プログラム(OB1)のサイクリックセクションでのみ許容されます。
  PI サービスが開始されない(「Req」= 0 の FB4 呼び出し)場合は、PLC 基本プログラム(OB100)の起動部分にもパラメータを書き込むことができます。その後、PI サービスは、「Req」= 1 の FB4 を呼び出すことで、PLC 基本プログラム(OB1)のサイクリックセクションにすでに書き込まれているパラメータを使用して開始することができます。
- PI サービスの実行は、通常、数回の PLC 周期に及びます。

ファンクションの宣言

```
FUNCTION_BLOCK FB4
VAR_INPUT
w   Req :               BOOL;
    PIService :         ANY ;
    Unit :              INT;
    Addr1 :             ANY ;
    Addr2 :             ANY ;
    Addr3 :             ANY ;
    Addr4 :             ANY ;
    WVar1 :             WORD;
    WVar2 :             WORD;
    WVar3 :             WORD;
    WVar4 :             WORD;
    WVar5 :             WORD;
    WVar6 :             WORD;
    WVar7 :             WORD;
    WVar8 :             WORD;
    WVar9 :             WORD;
    WVar10 :            WORD;
END_VAR
VAR_OUTPUT
    Error :             BOOL;
    Done :              BOOL;
    State :             WORD;
END_VAR
```

仮パラメータの説明

信号	タイプ	タイプ	値の範囲	説明
Req :	I	BOOL	0 (FALSE)、1 (TRUE)	ジョブの要求
PIService :	I	ANY	「<DBName>」.<PI サービス>	要求された PI サービス ● <DBName>:DB16 のシンボル名称、初期設定:"PI" ● <PI サービス>: 使用可能な PI サービスのリスト (ページ 1199)
単位:	I	INT	1, 2, 3 ...	領域番号
Addr1 ～ Addr4:	I	ANY	「DBName」.<VarName>	文字列の参照 選択した PI サービスにより異なる内容
WVar1 ～ WVar10:	I	WORD	-32768 ... 32767 8000_H - $7FFF_H$	INTEGER 変数または WORD 変数 選択した PI サービスにより異なる内容
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	エラー状態 1:ジョブの否定応答、またはジョブの実行不可
Done :	O	BOOL	0 (FALSE)、1 (TRUE)	ジョブの状態 1:ジョブが正常終了
State :	O	WORD	[1)]	エラーの検出 「Error」 == 1 の場合にのみ該当

信号		タイプ	タイプ	値の範囲	説明
1)	状態	説明			注
	3	否定応答、ジョブ実行不可			内部エラー、NC リセットによる可能な対策
	6	FIFO が一杯			命令の繰り返し
	7	オプション設定なし			FB1 パラメータ「NCKomm」が未設定
	9	転送が占有されています			命令の繰り返し
	13	Addr1..Adddr4:参照先が無効			欠けている文字列を指定
	14	不明な PI サービス			「PIService」パラメータに、不明な PI サービスが指定されています。→ 表記/スペルを確認してください。
	15	Addr1..Adddr4:文字列が長すぎます			文字列の長さをチェック

呼び出し例

機能:チャネル 1 内のプログラム選択(メインプログラムとワークプログラム)

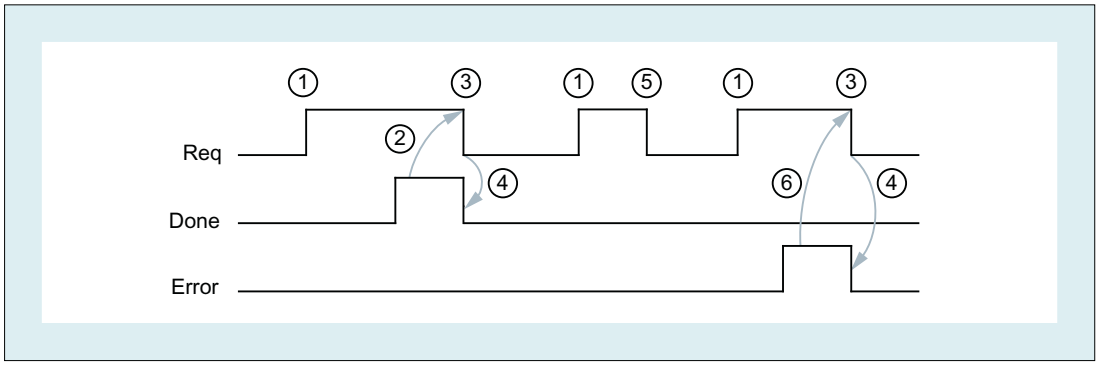
S7 SYMBOL エディタを使用した、DB16 に対する PI サービスおよび DB124 に対する STR の入力

パラメータ設定		
シンボル	オペランド	データタイプ
PI	DB16	DB16
STR	DB124	DB124

```
DATA_BLOCK DB126          // FB4 のインスタンス、未割り当てユーザー DB
FB4
BEGIN
END_DATA_BLOCK
// ---------------------------------------------------------------------------
DATA_BLOCK DB124
   STRUCT
```

```
      PName:        string[32]:= '_N_TEST_MPF   //メインプログラム
                    ';
      Path:         string[32]:= '/             //メインプログラムパス
                    _N_MPF_DIR/';
      PName_WST:    string[32]:= '_N_ABC_MPF';  //ワークプログラム
      Path_WST:     string[32]:= '/_N_WCS_DIR/  //ワークプログラムパス
                    _N_ZYL_WPD';
   END_STRUCT
BEGIN
END_DATA_BLOCK
// ------------------------------------------------------------------------
Function FC "PICall" :VOID
   U   I 7.7;        //未割り当て機械操作パネルキー
   S   M 0.0;        //要求の起動
   U   M 1.1;        //Done 完了メッセージ
   R   M 0.0;        //ジョブの終了
   U   I 7.6;        //Manual error acknowledgment
   U   M 1.0;        //異常未処置
   R   M 0.0;        //ジョブの終了

   CALL FB4, DB126 (
      Req :=        M0.0,
      PIService :=  "PI".SELECT,              // PI サービス:SELECT
      Unit:=        1,                        //チャネル 1
      Addr1:=       "STR".Path,               //メインプログラム:パス
      Addr2:=       "STR".PName,              //メインプログラム:プログラム
   //  Addr1:=       "STR".Path_WST,           //ワーク:パス
   //  Addr2:=       "STR".PName_WST,          //ワーク:プログラム
      Error :=      M1.0,
      Done :=       M1.1,
      State :=      MW2
   );
```

フローチャート



① User:要求の設定、Req = 0→1

② FB4:PI サービスが正常に終了、Done = 1
User:要求のリセット、IF Done == 1 THEN Req = 0

③ User:IF Done == 1 THEN、要求のリセット:1 → 0

④ FB4:ジョブ確認のリセット、Done = 0

⑤ User:IF Done == 0 AND Error == 0 THEN、要求のリセット、Req = 1 → 0 **許容されません**

⑥ FB4:PI サービスが異常とともに終了、Error = 1
User:要求のリセット、IF Done == 1 OR Error == 1 THEN Req = 0、さらにエラー処理の可能性

14.17.4.1 使用可能な PI サービスのリスト

一般 PI サービス

PI サービス	機能
ASUB (ページ 1201)	割り込みの割り当て
CANCEL (ページ 1202)	キャンセルの実行
CONFIG (ページ 1203)	タグ付きマシンデータの再設定
DIGION (ページ 1203)	ディジタル化オン
DIGIOF (ページ 1203)	ディジタル化オフ
FINDBL (ページ 1204)	ブロック検索の起動
LOGIN (ページ 1204)	パスワードの起動
LOGOUT (ページ 1205)	パスワードのリセット
NCRES (ページ 1205)	NC-RESET の起動

PI サービス	機能
SELECT (ページ 1205)	1 つのチャネルに対する処理用プログラムの選択
SETUDT (ページ 1206)	現在のユーザーデータを有効に設定
SETUFR (ページ 1207)	ユーザーフレームの起動
RETRAC (ページ 1207)	工具方向への工具の後退

工具管理機能の PI サービス

PI サービス	機能
CRCEDN (ページ 1208)	T 番号が指定された工具刃先の作成
CREACE (ページ 1209)	次に高い/次の未使用の D 番号を持つ工具刃先の作成
CREATO (ページ 1209)	T 番号が指定された工具の作成
DELECE (ページ 1210)	工具刃先の削除
DELETO (ページ 1210)	工具の削除
MMCSEM (ページ 1211)	各種 PI サービスのセマフォ
TMCRTO (ページ 1212)	名称、予備工具番号が指定された工具の作成
TMFDPL (ページ 1214)	ロードのための空きロケーションの検索
TMFPBP (ページ 1215)	空きロケーションの検索
TMGETT (ページ 1217)	予備工具番号で指定された工具名称の T 番号
TMMVTL (ページ 1218)	工具のロード／アンロード用マガジンロケーションの準備
TMPOSM (ページ 1220)	マガジンロケーションまたは工具の位置決め
TMPCIT (ページ 1221)	ワークカウンタ用のインクリメント値の設定
TMRASS (ページ 1222)	有効状態のリセット
TRESMO (ページ 1222)	監視値のリセット
TSEARC (ページ 1223)	検索画面を使用した高度な検索
TMCRMT (ページ 1227)	マルチ工具の作成
TMDLMT (ページ 1228)	マルチ工具の削除
POSMT (ページ 1229)	マルチ工具の位置決め
FDPLMT (ページ 1230)	マルチ工具内の空きロケーションを検索/チェック

14.17.4.2 PI サービス ASUB

機能:割り込みの割り当て

割り込み番号は、指定されたチャネルにおいて、制御装置に存在するパートプログラムに割り当てられ、パスおよびプログラム名で識別されます。

PI サービス ASUB には、プログラム命令 SETINT (または、パス名を指定せずに PI サービスが呼び出された場合は CLRINT)と同じ効果があります。

SETINT (または CLRINT)とは異なり、PI サービス ASUB はプログラム終了(M30 またはチャネルリセット)後も有効なままになります。PI サービス ASUB によって起動された割り当ては、ウォーム再起動後までクリアされません。

プログラム管理、パス、およびファイル名の詳細については、以下の文書を参照してください:

参照先
『プログラミングマニュアル、上級編』、「ファイルとプログラムの管理」の「プログラムメモリ」の章

仮パラメータの説明

信号	タイプ	値の範囲	意味
PIService :	ANY	"PI".ASUP	割り込みの割り当て
単位:	INT	1, 2, 3, ... 10	チャネル番号
Addr1:	STRING		パス名、 例: "/_N_MPF_DIR/"
Addr2	STRING		プログラム名、 例: "_N_TST_FC9ASUP_MPF"
WVar1:	WORD	1, 2, 3, ... 8	割り込み番号
WVar2:	WORD	1, 2, 3, ... 8	優先順位
WVar3:	WORD	0, 1	高速リトラクト [1] 輪郭からの高速リトラクト

信号	タイプ	値の範囲	意味
WVar4:	WORD	0, 1	BLSYNC [2] 割り込みルーチンの処理は、次のブロック変更でのみ開始します。

[1] **参照先**
『プログラミングマニュアル、上級編』、「フレキシブルな NC プログラミング」の「割り込みルーチン(ASUB)」の「輪郭からの高速リトラクト(SETINT、**LIFTFAST**、ALF)」の章

[2] **参照先**
『プログラミングマニュアル、上級編』、「フレキシブルな NC プログラミング」の「割り込みルーチン(ASUB)」の「割り込みルーチンの割り当てと起動(SETINT、PRIO、**BLSYNC**)」の章

注記

ASUB PI サービスは、指定されたチャネルがリセット状態の場合にのみ実行できます。FB7 により準備された ASUB は、FC9 により引き続き開始できます。

参照先
『プログラミングマニュアル、上級編』、「フレキシブル NC プログラミング」の「割り込みルーチン(ASUB)」の章

14.17.4.3 PI サービス CANCEL

機能:キャンセルの実行

操作画面(操作パネル)の[アラームのキャンセル]ボタンに相当する「キャンセル」機能を起動します。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".CANCEL	キャンセル

14.17.4.4 PI サービス CONFIG

機能:再構成

再構成命令を使用すれば、ほとんど同時に、オペレータまたは PLC が連続して入力したマシンデータが有効になります。

この命令を起動できるのは、コントローラが RESET 状態になっているか、プログラムが中断されている場合(ブロック境界での NC ストップ停止)に限ります。この条件が満たされない(状態= 3)場合は、FB4 異常アンサーバック信号が出力されます。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	PI.CONFIG	再構成
単位:	INT	1	
WVar1:	INT	1	分類

14.17.4.5 PI サービス DIGION

機能:ディジタル化オン

パラメータ設定したチャネル内でのディジタル化の選択

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".DIGION	ディジタル化オン
単位:	INT	1, 2, 3 ... 10	チャネル

14.17.4.6 PI サービス DIGIOF

機能:ディジタル化オフ

パラメータ設定したチャネル内でのディジタル化の解除

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".DIGIOF	ディジタル化オフ
単位:	INT	1, 2, 3 ... 10	チャネル

14.17.4.7 PI サービス FINDBL

機能:ブロック検索の起動

チャネルがブロックサーチモードに切り替えられた後、対応する応答が伝送されます。その後、NC により直ちにブロック検索が実行されます。ブロック検索ポインタはこの

時点ですでに NC 内に設定されている必要があります。NC RESET を使用すれば、いつでもブロック検索を中断できます。ブロック検索が正常に完了すると、再度、通常の処理モードが自動的に有効になります。その後、到達した検索ターゲットから NC スタートが有効になります。

衝突のないアプローチ軌跡を保証するのは、オペレータの責任です。

仮パラメータの説明

信号	タイプ	値の範囲	意味
PIService :	ANY	"PI".FINDBL	ブロックサーチ
単位:	INT	1 ～ 10	チャネル
WVar1:	WORD	1, 2, 3	事前解析モード 1:計算なし 2:計算あり 3:メインブロックを考慮

14.17.4.8 PI サービス LOGIN

機能:パスワードの作成

パラメータ設定したパスワードを NC に伝送します。パスワードは、通常 8 文字で構成されます。より短いパスワードでは、8 文字に達するまで文字列を空白で補うことができます

例

パスワード:STRING[8] := 'SUNRISE';

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".LOGIN	パスワードの作成
単位:	INT	1	NC
Addr1:	STRING	8 文字	パスワード

14.17.4.9 PI サービス LOGOUT

機能:パスワードのリセット

NC に最後に伝送されたパスワードがリセットされます。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".LOGOUT	パスワードのリセット
単位:	INT	1	NC

14.17.4.10 PI サービス NCRES

機能:NC-RESET の起動

NC-RESET を実行します。パラメータ「Unit」と「WVar1」は常に 0 に設定してください。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".NCRES	NC-RESET の起動
単位:	INT	0	-
WVar1:	WORD	0	-

14.17.4.11 PI サービス SELECT

機能:チャネルの処理の選択

1 つのチャネルの実行のために、NC に格納されたプログラムを選択します。プログラムは実行可能であることが必要です。パスとプログラム名は、完全に指定する必要があります。

詳細については、以下の資料を参照してください。

参照先

『プログラミングマニュアル　上級編』; :「ファイルとプログラムの管理」の「プログラムメモリ」の章

考えられるブロックタイプ

ブロックタイプ	
ワークディレクトリ	WPD
メインプログラム	MPF
サブプログラム	SPF
サイクル	CYC

ブロックタイプ	
非同期サブプログラム	ASP
バイナリファイル	BIN

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".SELECT	プログラムの選択
単位:	INT	1, 2, 3 ... 10	チャネル
Addr1:	STRING	-	パス名称
Addr2:	STRING	-	プログラム名称

14.17.4.12 PI サービス SETUDT

機能:現在のユーザーデータを有効に設定します

工具オフセット、基本フレームおよび設定可能フレームなどの現在のユーザーデータが、次の NC ブロックで有効に設定されます(停止状態の場合のみ)。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".SETUDT	ユーザーデータを有効にします
単位:	INT	1, 2, 3 ... 10	チャネル
WVar1:	WORD	1, 2, 3, 4, 5	ユーザーデータタイプ 1:有効な工具オフセット 2:有効な基本フレーム 3:有効な設定可能フレーム 4:有効なグローバル基本フレーム 5:有効なグローバル設定可能フレーム
WVar2:	WORD	0	予備
Wvar3:	WORD	0	予備

14.17.4.13 PI サービス SETUFR

機能:ユーザーフレームの起動

ユーザーフレームが NC にロードされます。FB3「変数の書き込み」で、必要なフレーム値をすべて、最初に NC に伝送しておいてください。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".SETUFR	ユーザーフレームの起動
単位:	INT	1, 2, 3 ... 10	チャネル

14.17.4.14 PI サービス RETRAC

機能:JOG 後退を選択

JOG 後退モードを選択します。後退軸、つまり、後退を実行するジオメトリ軸は、NC によって自動的に、または明示的に指定して特定できます。

このモードは、リセットで終了されるまで、有効なままになります。

注記

RETRAC PI サービスは、JOG モードの「リセット」状態でのみ起動できます。

自動特定

自動で特定するには、現在選択している作業平面に垂直な(直交の)ジオメトリ軸を後退軸として選択します。

- G17:後退軸⇒ 3 番目のジオメトリ軸(Z)
- G18:後退軸⇒ 2 番目のジオメトリ軸(Y)
- G19:後退軸 ⇒ 1 番目のジオメトリ軸(X)

注記

OPI 変数 retractState

有効な後退軸は、OPI 変数 retractState.Bit 2/3 を介して読み取ることができます。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".RETRAC	JOG 後退モードを選択
単位:	INT	1, 2, 3 ... 10	チャネル
WVar1:	WORD	0、1、2、3	後退軸 0:NC による後退軸の自動特定。 1:後退軸は WCS の 1 番目のジオメトリ軸 2:後退軸は WCS の 2 番目のジオメトリ軸 3:後退軸は WCS の 3 番目のジオメトリ軸
WVar2:	WORD	0	予約済み値は 0 でプリセットしてください。

14.17.4.15 PI サービス CRCEDN

機能:新しい刃先を作成します。

PI サービス内の"T number"パラメータで、既存の工具の T 番号が入力されている場合、この既存の工具に対して刃先が作成されます。この場合、パラメータ"D number" (作成する刃先の番号)の数値の範囲は 1 ～ 9 です。

パラメータとして正の T 番号が指定されたときに、入力される T 番号の工具が存在していない場合、PI サービスが中止されます。

T 番号として値 0 が入力されている場合(絶対 D 番号のモデル)、D 番号の値の範囲は 1 ～ 31999 です。新しい刃先が、指定した D 番号で準備されます。

指定した刃先がすでに存在している場合、どちらの場合も PI サービスが中止されます。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".CRCEDN	新しい刃先の作成
単位:	INT	1 … 10	TOA

信号	タイプ	値の範囲	説明
WVar1:	INT		刃先の作成が必要な工具の T 番号 00000 と設定すると、この刃先は、どの特定工具も参照する必要がありません(絶対 D 番号)。
WVar2:	INT	1 ... 9	工具刃先の刃先番号
		1 ... 31999	

14.17.4.16 PI サービス CREACE

機能:工具刃先の作成

TO および TS(もしあれば)での、伝送された T 番号を持つ工具に対応した、次に高い/次の未割り当ての D 番号を持つ刃先の作成 TUE ブロックがあれば、OEM 刃先データに対応した刃先が TUE ブロックに同時に準備されます。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".CREACE	工具刃先の作成
単位:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		T 番号

14.17.4.17 PI サービス CREATO

機能:工具作成

T 番号が指定された工具の作成工具は工具ディレクトリ領域(TV)内に存在しているとおりに入力されます。TO ブロックで、工具オフセットに、最初の「刃先」D1 (内容はゼロ)が作成されます。TUE ブロックがある場合には、このブロックで OEM「刃先」データに対しても D1 (内容がゼロ)が作成されます。TU ブロックが存在している場合、これには工具に設定されたデータが含まれます。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".CREATO	工具作成
単位:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		T 番号

14.17.4.18 PI サービス DELECE

機能:工具刃先の削除

PI サービス内の"T number"パラメータで、既存の工具の T 番号が入力されている場合、この既存の工具に対して刃先が削除されます。この場合、パラメータ"D number" (作成する刃先の番号)の数値の範囲は 00001 ～ 00009 です。パラメータとして正の T 番号が指定されたときに、入力される T 番号の工具が存在していない場合、PI サービスが中止されます。T 番号として値 00000 が入力されている場合(絶対 D 番号のモデル)、D 番号の値の範囲は 00001 ～ 31999 です。指定した刃先が存在しないと、どちらの場合でも PI サービスが中止されます。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".DELECE	刃先の削除
単位:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		刃先の作成が必要な工具の T 番号 00000 と設定すると、この刃先は、どの特定工具も参照する必要がありません(絶対 D 番号)。
WVar2:	INT	1 - 9	削除する必要がある刃先の D 番号
		1 - 31999	

14.17.4.19 PI サービス DELETO

機能:工具の削除

伝送された T 番号に割り当てられた工具をすべての刃先と共に削除します。(TO、場合によっては TU、TUE および TG (タイプ 4xx)、TD と TS ブロックで)

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".DELETO	工具の削除
単位:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		T 番号

14.17.4.20 PI サービス MMCSEM

機能

PI サービスは、PLC を通じた HMI からのテスト、設定、チャネル別セマフォのリセットに使用されます。重要なデータ領域を保護するために、チャネル毎に 10 個のセマフォを使用できます。

ファンクション(PI サービス)がセマフォ 1 ～ 6 に割り当てられます。セマフォ 7 ～ 10 は自由に使用できます。

FB4 戻り値

- 「Done」== TRUE
  セマフォが設定されており、重要な機能を呼び出すことができます。
- 「Error」== TRUE AND「State」== 3
  セマフォはすでに設定されており、現時点では重要な機能を呼び出すことはできません。

自由なセマフォのシーケンスの図式

```
Testing and setting the semaphore
IF semaphore == FREE
THEN
   Writing/reading of critical data
   Resetting the semaphore
ELSE // Semaphore is blocked
...
ENDIF
```

通知
Resetting the semaphore セマフォ、および後続のデータの読み取りまたは書き込みを設定することにより重要なデータ領域をブロックした後、セマフォをリセットすることにより重要なデータ領域を再度有効にする必要があります。それ以外の場合、以降のブロックは可能になりません。

ブロックされたセマフォのシーケンスの図式

```
Testing and setting the semaphore
IF semaphore == FREE
THEN
   ...
ELSE // Semaphore is blocked
   Set bit memory for "Function could not be executed, repeat
necessary"
ENDIF
```

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	PI.MMCSEM	セマフォの編集
単位:	INT	1, 2, 3 ... 10	チャネル
WVar1:	INT	1)	セマフォの PI サービス別番号
WVar2:	WORD	0, 1	ジョブタイプ 0:セマフォのリセット 1:セマフォのテストと設定

1)	番号	PI サービス
	1	TMCRTO (工具の作成)
	2	TMFDPL (ロードのための空ロケーションの検索)
	3	TMMVTL (工具のロード、アンロードのためのマガジンロケーションの準備)
	4	TMFPBP (位置の検索)
	5	TMGETT (工具番号の検索)
	6	TSEARC (工具の検索)
	7 ... 10	自由に使用可能

14.17.4.21 PI サービス TMCRTO

機能:工具作成

名前、予備工具番号の指定による工具作成(例: $TC_TP1[y] =予備工具番号、または $TC_TP2[y] =<工具名>)。または、y = <T 番号>のように、T 番号を任意に使用。

工具は、TV 領域(工具ディレクトリ)内で使用可能として入力されます。

1 番目の刃先"D1" (内容はゼロ)が TO ブロックで補正用に作成されます。

1 番目の刃先 D1 (内容はゼロ)が TS ブロックで、データの監視用に作成されます。

使用可能な場合は、TUE ブロックで刃先 D1 が OEM 刃先データ用に作成されます。

TD ブロックには、NC により指定されるか、割り当てられる T 番号に対する識別子、予備工具番号、および刃先の番号(=1)が含まれます。

TU ブロックが存在している場合、これには工具のデータブロックが含まれます。

PI サービスの実行後、**TnumWZV** 下の TV ブロックで、作成された工具の T 番号を使用できます。

注記

この PI サービスの前後で、パラメータが「WVar1」の PI サービス MMCSEM を、TMCRTO のファンクション番号 1 で呼び出してください。「PI サービス MMCSEM (ページ 1211)」の章を参照してください。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".TMCRTO	工具作成
単位:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT	>0 → T 番号が指定されています。 -1 → NC により T 番号が割り当てられます	T 番号
WVar2:	INT		予備工具番号
Addr1:	STRING	最大 32 文字	工具名称

14.17.4.22 PI サービス TMFDPL

機能

パラメータ設定に応じて、ロード用の空きロケーションを検索します。

- LocationNumber_To = -1 および MagazineNumber_To = -1
  T 番号で指定された工具の空きローケーションを指定領域(=チャネル)のすべてのマガジンで検索します。PI サービスの実行後、検索中に検出されたマガジン番号とロケーション番号は、チャネルの設定ブロックに記載されます(コンポーネント **magCMCmdPar1** (マガジン番号)と **magCMCmdPar2** (ロケーション番号))。
  LocationNumber_ID と MagazineNumber_ID を検索条件として設定することも、設定しないこと(=-1)も可能です。PI サービスは検索結果に応じて、肯定または否定の確認応答をおこないます。
- LocationNumber_To = -1 および MagazineNumber_To = マガジン番号
  指定したマガジンで、T 番号を指定した工具の空きローケーションが検索されます。ロケーション番号(参照先)および MagazineNumber_Ref には、検索条件または-1 を割り当てることができます。PI サービスは検索結果に応じて、肯定または否定の確認応答をおこないます。
- LocationNumber_To = ロケーション番号および MagazineNumber_To = マガジン番号
  指定したロケーションをチェックし、工具をロードできることを確認します。
  LocationNumber_ID と MagazineNumber_ID を検索条件として設定することも、-1 を設定することも可能です。PI サービスは検索結果に応じて、肯定応答または否定応答します。

パラメータ「WVar1」および「WVar2」は、ソースに配置します。

ロード: ソースが内部ロードマガジンの場合は、パラメータが目標に配置されます(実マガジン)。

アンロード: ソースは必ず実マガジンです。

注記

この PI サービスの前後で、パラメータが「WVar1」の PI サービス MMCSEM を、TMFDPL のファンクション番号 2 で呼び出してください。
章「PI サービス MMCSEM (ページ 1211)」を参照してください。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".TMFDPL	ロードのための空きローケーション
単位:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		T 番号
WVar2:	INT		LocationNumber_To (ターゲット)
WVar3:	INT		MagazineNumber_To (ターゲット)
WVar4:	INT		LocationNumber_Ref (参照先)
WVar5:	INT		MagazineNumber_Ref (参照先)

14.17.4.23 PI サービス TMFPBP

機能:空きローケーションの検索

PI サービスは、指定されたマガジンで、工具サイズやロケーションタイプなど、指定された基準を満たす空きロケーションを検索します。

検索が正常に完了すると、以下の OPI 変数から検索結果を読み取ることができます。

- magCMCmdPar1 (マガジン番号)
- magCMCmdPar2 (ロケーション番号)

注記

PI サービスは、FB7 でのみ要求できます。「FB7:PI_SERV2 - PI サービス要求 (ページ 1242)」の章を参照してください。

注記

この PI サービスの前後で、パラメータが「WVar1」の PI サービス MMCSEM を、TMFPBP のファンクション番号 4 で呼び出してください。「PI サービス MMCSEM (ページ 1211)」の章を参照してください。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	PI.TMFPBP	空きロケーションの検索
単位:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT		検索を**開始する**マガジンの**マガジン番号**

信号	タイプ	値の範囲	説明
WVar2:	INT		「WVar1」のマガジン内のロケーションの**ロケーション番号**
WVar3:	INT		検索を終了するマガジンの**マガジン番号**
WVar4:	INT		「WVar3」のマガジン内のロケーションの**ロケーション番号**
WVar5:	INT		マガジン番号参照先
WVar6:	INT		ロケーション番号参照先
WVar7:	INT	0, 1, 2 ... 7	左に必要な半ロケーションの番号
WVar8:	INT	0, 1, 2 ... 7	右に必要な半ロケーションの番号
WVar9:	INT	0, 1, 2 ... 7	上方に必要な半ロケーションの番号
WVar10:	INT	0, 1, 2 ... 7	下方に必要な半ロケーションの番号
WVar11:	INT		必要なロケーションタイプの番号
WVar12:	INT	0, 1, 2, 3, 4	必要な検索方向を指定 0:$TC_MAMP2 に設定された検索方法 1:進む 2:戻る 3:対称

例:検索範囲の設定

から	ロケーション	まで	ロケーション	説明
WVar1	**WVar2**	**WVar3**	**WVar4**	
#M1	#P1	#M1	#P1	マガジン#M1 のロケーション#P1 だけがチェックされます
#M1	#P1	#M2	#P2	マガジン#M1、ロケーション#P1 から開始され、マガジン #M2、ロケーション#P2 までのロケーションが検索されます。
#M1	-1	#M1	-1	マガジン#M1 内のロケーションがすべて検索され、他は検索されません。
#M1	-1	-1	-1	マガジン#M1 から開始されるロケーションがすべて検索されます。
#M1	#P1	-1	-1	マガジン#M1 とロケーション#P1 から開始されるすべてのロケーションが検索されます。

から	ロケーション	まで	ロケーション	説明
WVar1	**WVar2**	**WVar3**	**WVar4**	
#M1	#P1	#M1	-1	マガジン#M1 とロケーション#P1 から開始されるマガジン#M1 内のロケーションが検索されます。
#M1	#P1	#M2	-1	マガジン#M1、ロケーション#P1 から開始され、マガジン#M2 までのロケーションが検索されます。
#M1	-1	#M2	#P2	マガジン#M1 から開始され、マガジン#M2、ロケーション#P2 までのロケーションが検索されます。
#M1	-1	#M2	-1	マガジン#M1 から開始され、マガジン#M2(含む)までのロケーションが検索されます。
-1	-1	-1	-1	マガジンロケーションがすべて検索されます。

14.17.4.24 PI サービス TMGETT

機能:予備工具番号で指定された工具名称の T 番号の特定

PI サービスを使用して、工具名と予備工具番号で工具の T 番号を特定します。

結果は、TF ブロックの BTSS 変数に書き込まれます(パラメータ設定、TMGETT、TSEARC からの戻りパラメータ)。

- resultNrOfTools
  - resultNrOfTools == 0:指定された工具を検出しませんでした
  - resultNrOfTools == 1:指定された工具を検出しました
- resultToolNr:「resultNrOfTools」== 1 で指定された工具の T 番号

注記

この PI サービスの前後で、パラメータが「WVar1」の PI サービス MMCSEM を、TMGETT のファンクション番号 5 で呼び出してください。「PI サービス MMCSEM (ページ 1211)」の章を参照してください。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	PI.TMGETT	T 番号の特定
単位:	INT	1, 2, 3 ... 10	TOA

信号	タイプ	値の範囲	説明
Addr1:	STRING	最大 32 文字	工具名称
WVar1:	INT		予備工具番号

14.17.4.25 PI サービス TMMVTL

機能:工具のロード／アンロード用マガジンロケーションの準備

PI サービスを使用して工具をロード、アンロード、および再配置します。

1. ロードとアンロード:ロードポイント ↔ マガジン
2. ロードとアンロード:ロードポイント↔バッファロケーション(主軸など)
3. マガジン内の再配置
4. 異なるマガジン間の再配置
5. マガジンとバッファロケーション間の再配置
6. バッファロケーション内の再配置

1、3、4、および 5 の場合:ブロック TM (マガジンデータ:一般データ)の以下の BTSS 変数が書き込まれます。

- magCmd (area no. = TO ユニット、line =マガジン番号)
- magCmdState←「応答」

2 および 6 の場合:ブロック TMC (マガジンデータ:設定データ)の以下の BTSS 変数が書き込まれます。

- magCBCmd (area no. = TO ユニット)
- magCBCmdState←「応答」

ロード

- "WVar2" LocationNumber_From、"WVar3" MagazineNumber_From
  マガジンの工具ロケーションが、ロード用のロードステーション/ロードポイントに移動し、工具がロードされます。
- "WVar4" LocationNumber_To== -1
  最初に、工具のための空きロケーションをマガジン内で検索します。そして、マガジンの空きロケーションが、ロード用のロードステーション/ロードポイントに移動し、工具がロードされます。
  PI サービスが実行された後、検出された空きロケーションの番号が「magCMCmdPar2」(BTSS ブロック TM)、チャネルの**実際の**マガジンに配置されます。
- "WVar4" LocationNumber_To == -2
  工具が、マガジンの現在の工具ロケーション内に配置されます。PI サービスが実行された後、ロケーション番号が「magCMCmdPar2」(BTSS ブロック TM)、チャネルの**実際の**マガジンに配置されます。

アンロード

- "WVar4" LocationNumber_To、"WVar5" MagazineNumber_To
- マガジンの工具ロケーションが、ロード用のロードステーションに移動し、工具がアンロードされます。

BTSS ブロック TP (マガジンデータ:ロケーションデータ)で、取り外された工具のマガジンロケーションが、空きとして指定されます。

工具のアドレス指定

工具は、T 番号を使用するか、ロケーション番号とマガジン番号を使用してアドレス指定できます。値-1 は、未使用のパラメータに割り当てられます。

注記

この PI サービスの前後で、パラメータが「WVar1」の PI サービス MMCSEM を、TMMVTL のファンクション番号 3 で呼び出してください。「PI サービス MMCSEM (ページ 1211)」の章を参照してください。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	PI.TMMVTL	工具のロード、アンロード用のマガジンロケーションの準備
単位:	INT	1, 2, 3 ... 10	TOA

信号	タイプ	値の範囲	説明
WVar1:	INT		T 番号
WVar2:	INT		LocationNumber_From (ソース)
WVar3:	INT		MagazineNumber_From (ソース)
WVar4:	INT		LocationNumber_To (ターゲット)
WVar5:	INT		MagazineNumber_To (ターゲット)

14.17.4.26 PI サービス TMPOSM

機能:パラメータ設定に応じて、マガジンロケーションまたは工具を位置決めします。

この PI サービスにより、マガジンを移動して指定の位置(たとえばロード位置)にマガジンロケーションを位置決めできます。マガジンロケーションは、直接指定することもできますし、該当ロケーションにある工具によって指定することもできます。

移動先(ロードポイントなど)は、パラメータで指定します。

- "WVar5" LocationNumber_Ref
- "WVar6" MagazineNumber_Ref

位置決めするマガジンロケーションは、それぞれの仕様に応じて、以下のパラメータで指定します。

- "WVar1"工具の T 番号
  以下のパラメータは、ここでは無意味です。
  - "Addr1"工具名 = ""
  - "WVar2"予備工具番号 = -1
  - "WVar3" LocationNumber_From = -1
  - "WVar4" MagazineNumber_From = -1
- "Addr1"工具名、"WVar2"予備工具番号
  以下のパラメータは、ここでは無意味です。
  - "WVar1"工具の T 番号 = -1
  - "WVar3" LocationNumber_From = -1
  - "WVar4" MagazineNumber_From = -1
- "WVar3" LocationNumber_From、"WVar4" MagazineNumber_From
  以下のパラメータは、ここでは無意味です。
  - "Addr1"工具名 = ""
  - "WVar1"工具の T 番号 = -1
  - "WVar2"予備工具番号 = -1

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	PI.TMPOSM	マガジンロケーションまたは工具の位置決め
単位:	INT	1, 2, 3 ... 10	TOA
Addr1:	STRING	最大 32 文字	工具名称
WVar1:	INT		工具の T 番号
WVar2:	INT		予備工具番号
WVar3:	INT		LocationNumber_From (ソース)
WVar4:	INT		MagazineNumber_From (ソース)
WVar5:	INT		LocationNumber_Ref
WVar6:	INT		MagazineNumber_Ref

14.17.4.27 PI サービス TMPCIT

機能:ワークカウンタ用のインクリメント値の設定

主軸工具のワークカウンタのインクリメント

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	PI.TMPCIT	ワークカウンタ用のインクリメント値の設定
単位:	INT	1, 2, 3 ... 10	TOA
WVar1:	WORD	0、1、2、... max。	主軸番号。チャネル内のバッファマガジンの「主軸」位置タイプが指定されたロケーションデータ内のタイプインデックスに対応します。 0:主軸
WVar2:	WORD	0 ... 最大値	インクリメント値。主軸の回転量を示します。回転後、ワークカウンタがインクリメントされます。

14.17.4.28 PI サービス TMRASS

機能:工具状態「有効」のリセット

PI サービスにより、工具状態が「有効」または「ブロック」の工具すべての状態が「無効」に設定されます。

以下のイベントは、工具状態をリセットするタイミングとして適しています。

- NC/PLC インターフェース信号「工具の無効化が無効」の立ち下がりエッジ
- プログラムの終了
- チャネルのリセット

この PI サービスは PLC 向けです。これは、PLC が無効化された工具が最終的に使用されなくなるのはいつであるかを認識しているためです。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	PI.TMRASS	有効状態のリセット
単位:	INT	1, 2, 3 ... 10	TO 領域

14.17.4.29 PI サービス TRESMO

機能:監視値のリセット

この PI サービスは、指定工具の指定刃先の監視値を、その指令値(初期値)にリセットします。
これは、監視が有効な工具に対してのみおこなわれます。

RESETMON 命令も参照してください。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".TRESMO	監視値のリセット
単位:	INT	1, 2, 3 ... 10	TO 領域
WVar1:	WORD	-最大～最大	工具番号 0:すべての工具の処理 > 0:指定された工具の処理 < 0:すべての姉妹工具の処理

信号	タイプ	値の範囲	説明
WVar2:	WORD	0、1、2、... max。	D 番号 < 0:指定された刃先の監視のリセット 0:すべての刃先の監視のリセット
Wvar3:	WORD		リセットされる監視モード(すべての組合わせが可能):
			ビット 0 = 1:工具寿命の監視
			ビット 1 = 1:数量の監視
			ビット 2 = 1:摩耗の監視
			ビット 3 = 1:サムオフセットの監視
			0:有効な工具監視機能($TC_TP9)がすべてリセットされます。

14.17.4.30 PI サービス TSEARC

機能:パラメータ設定に応じて、検索画面を使用した複合検索をおこないます。

PI サービスを使用して、1 つ以上のマガジンの特定のロケーションから特定のロケーションまでの検索範囲内で、特定の特性を持つ工具を検索します。

注記

有効な工具管理機能

PI サービスを使用できるのは、工具管理機能が起動されている場合に限ります。

指定オプション

- 検索方向
- 指定された特性を持つ次の工具の検索
- 指定された特性を持つすべての工具の検索

結果

結果として、検出された工具の内部 T 番号を含むリストが作成されます。

論理演算

フィルタ特性の場合、リンクオプションとして使用できるのは、1 つの AND リンクのみです。OR リンクは、ユーザーが PI サービスを複数回呼び出し、その後に個々の結果を評価して実現してください。

工具特性のパラメータ設定

検索する工具の特性は、BTSS ブロック TF (パラメータ設定、TMGETT や TSEARC からの戻りパラメータ)で、以下の変数を介して設定します。

- "parMask<X>" パラメータ設定マスク
- "parData<X>" 比較値

With <X> = TAD、TAO、TAS、TD

結果リスト

PI サービスが異常なく終了した後、検索結果が、以下の変数の BTSS ブロック TF (パラメータ設定、TMGETT や TSEARC の戻りパラメータ)に配置されます。

- "resultCuttingEdgeNrUsed"、最後の数量カウント以降に使用された刃先の D 番号
- "resultNrOfCutEdgesUsed"、最後の数量カウント以降の刃先の数
- "resultNrOfTools"、検出された工具の数
- "resultToolNr"、検出された工具の T 番号
- "resultToolNrUsed"、最後の数量カウント以降に使用された刃先の T 番号

工具が検出されなかった場合、検出された工具の数は 0 です("resultNrOfTools" == 0)

検索範囲の指定

開始マガジン番号 **WVar1**	開始ロケーション番号 **WVar2**	終了マガジン番号 **WVar3**	終了ロケーション番号 **WVar4**	説明 以下のマガジンロケーションが検索されます。
#M<a>	#L<b>	#M<x>	#L<y>	国:マガジン#M<a>、ロケーション #L<b> ～:マガジン#M<x>、ロケーション#L<y>
#M<a>	-1	#M<a>	-1	国:マガジン#M<a>、最初のロケーション ～:マガジン#M<a>、最後のロケーション
#M<a>	-1	-1	-1	国:マガジン#M<a>、最初のロケーション ～:最後のマガジン、最後のロケーション

開始マガジン番号 WVar1	開始ロケーション番号 WVar2	終了マガジン番号 WVar3	終了ロケーション番号 WVar4	説明 以下のマガジンロケーションが検索されます。
#M<a>	#L<b>	-1	-1	国:マガジン#M<a>、ロケーション#L<b> ～:最後のマガジン、最後のロケーション
#M<a>	#L<b>	#M<a>	-1	国:マガジン#M<a>、ロケーション#L<b> ～:マガジン#M<a>、最後のロケーション
#M<a>	#L<b>	#M<x>	-1	国:マガジン#M<a>、ロケーション#L<b> ～:マガジン#M<b>、最後のロケーション
#M<a>	-1	#M<x>	#L<y>	国:マガジン#M<a>、最初のロケーション ～:マガジン#M<b>、ロケーション#L<y>
#M<a>	-1	#M<x>	-1	国:マガジン#M<a>、最初のロケーション ～:マガジン#M<b>、最後のロケーション
-1	-1	-1	-1	国:最初のマガジン、最初のロケーション ～:最後のマガジン、最後のロケーション

対称検索

マガジンロケーションを基準とする対称検索は、以下の条件が満たされている場合にのみ実行できます:

- 検索範囲に含まれるのは、1 つのマガジンのみ:「WVar1」(開始:マガジン番号) == 「WVar3」(終了:マガジン番号)
- 基準ロケーション、つまり対称検索を実行するマガジンロケーションの指定:「WVar5」(基準マガジンの数)および「WVar6」(基準ロケーションの数)

- 基準ロケーションに対して、検索されるマガジンの複数割り当てを TPM ブロックで設定しておいてください。
- 「WVar7」(検索方向) = 3

基準ロケーションは、バッファロケーション、つまりバッファマガジンまたは内部ロードマガジン(たとえば交換ステーション、グリッパー、ロードステーションなど)からの工具ロケーションです。対称検索は、基準ロケーションの前のマガジンロケーションを基準にして実行されます。

マガジンロケーションが、検索範囲外にある基準ロケーションの上流である場合は、PI サービスが、適切なロケーションが検出されなかったかのように動作します。

注記

この PI サービスの前後で、この PI サービスの関連パラメータ WVar1 とともに MMCSEM PI サービスを呼び出してください。「PI サービス MMCSEM (ページ 1211)」の章を参照してください。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".TSEARC	検索画面を使用した高度な検索
単位:	INT	1, 2, 3 ... 10	TOA
WVar1:	INT	-1, 1, …	国:マガジン番号
WVar2:	INT	-1, 1, …	国:ロケーション番号
WVar3:	INT	-1, 1, …	～:マガジン番号
WVar4:	INT	-1, 1, …	～:ロケーション番号
WVar5:	INT	-1, 1, …	基準マガジンの数 (対称検索にのみ該当: 検索方向==3)
WVar6:	INT	-1, 1, …	基準ロケーションの数 (対称検索にのみ該当: 検索方向==3)

信号	タイプ	値の範囲	説明
WVar7:	INT	1, 2, 3	検索方向: 1:検索領域の最初のロケーションから前方へ 2:検索領域の最後の位置から後方へ 3:「WVar5」(基準マガジンの数)と「WVar6」(基準ロケーションの数)で指定されたロケーションの**上流**に配置された実際のマガジンロケーションと対称的に
WVar8:	INT	0、1、2、3	工具の特性: 0:すべての工具、刃先別 1:最初の工具、刃先別 2:すべての工具、すべての刃先を介して 3:最初の工具、すべての刃先を介して

14.17.4.31 PI サービス TMCRMT

機能:マルチ工具の作成

PI サービスを使用して、識別子、オプションで指定できるマルチ工具番号、工具ロケーションの数、および距離コードのタイプが定義された新しいマルチ工具を作成します。

注記

この PI サービスの前後で、パラメータが「WVar1」=1 (TMCRTO)の PI サービス MMCSEM を呼び出してください。「PI サービス MMCSEM (ページ 1211)」の章を参照してください。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	PI.TMCRMT	マルチ工具の作成
単位:	INT	1, 2, 3 ... 10	TOA
Addr1:	STRING		マルチ工具識別子(最大 32 文字)
WVar1:	INT	0	予備

信号	タイプ	値の範囲	説明
WVar2:	INT	-1, 1, 2, ... 32000	マルチ工具番号 -1:NC によるマルチ工具番号の自動割り当て 1, 2, 3 ... 32000:マルチ工具番号、 **注記**:マルチ工具番号は、一義的でなければなりません。
WVar3:	INT	2、3、4、... MAX	工具ロケーションの数 MAX = MD17504 $MN_MAX_TOOLS_PER_MULTITOOL で設定された数
WVar4:	INT	1, 2, 3	距離の割り付けのタイプ

14.17.4.32 PI サービス TMDLMT

機能:マルチ工具の削除

PI サービスを使用して、マルチ工具が保存されているすべてのデータブロックでマルチツールを削除します。マルチ工具に装備された工具は、以降に装備またはロードされませんが、削除もできない場合は引き続き定義されます。

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	PI.TMDLMT	マルチ工具の削除
単位:	INT	1, 2, 3 ... 10	TOA
Addr1:	STRING		マルチ工具識別子(最大 32 文字)
WVar1:	INT	0	予約済み、常時 0

信号	タイプ	値の範囲	説明
WVar2:	INT	-1, 1, 2, ... 32000	マルチ工具番号 -1:「Addr1」で名称が指定されているマルチ工具を削除します。 1 ,2, 3 ... 32000:指定されたマルチ工具番号のマルチ工具を削除します。 **注記:**パラメータ「Addr1」は使用されません。
WVar3:	INT	0, 1	マルチ工具に含まれる工具: 0:削除しない。 1:削除

14.17.4.33 PI サービス POSMT

機能:マルチ工具の位置決め

PI サービスを使用して、マルチ工具を、プログラム指令されたロケーションに位置決めするか、マルチ工具のロケーションのいずれかに配置されている、プログラム指令された工具に位置決めします。工具自体は、T 番号を使用して、または名称と予備工具番号で指定することができます。マルチ工具は、工具ホルダ(たとえば主軸など)にある場合で、その工具ホルダを基準とする工具オフセットが有効でない場合にのみ位置決めできます。

位置の指定

位置の指定は、3 タイプのうちのいずれかとして指定できます:

No.	Addr1	WVar1	WVar2	WVar3	WVar4
1	空の文字列	工具ホルダ番号	工具番号	-1	
2	工具名称	工具ホルダ番号	-1	予備工具番号	
3	空の文字列	工具ホルダ番号	-1	-1	マルチツールのロケーション番号

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	"PI".POSMT	マルチ工具の位置決め
単位:	INT	1, 2, 3 ... 10	TOA
Addr1:	STRING		マルチ工具内に位置決めされる工具の工具名(最大 32 文字)
WVar1:	INT	1, 2, 3, ... 999	工具ホルダ番号
WVar2:	INT	-1, 1, 2, 3 ... 32000	マルチ工具内に位置決めされる工具の工具番号(T 番号) 1, 2, 3, ... 32000:工具番号 -1:"Addr1" (工具名)および"WVar3" (予備工具番号)が使用されます
WVar3:	INT	-1, 1, 2, 3 ... 32000	マルチ工具内に位置決めされる工具の予備工具番号 1, 2, 3, ... 32000:予備工具番号 -1:"WVar2" (T 番号)が使用されます
WVar4:	INT	1, 2, 3 ... 999	システムが位置決めされる工具ホルダロケーションのマルチ工具ロケーション番号

14.17.4.34 PI サービス FDPLMT

機能:マルチ工具内の空きロケーションの検索またはチェック

PI サービスを使用してマルチ工具内の空き工具ロケーションを検索し、指定された工具を収納するか、または、指定された工具を収納するためにマルチ工具内で指定された工具ロケーションが空いているかどうかをチェックします。T 番号、または識別子と予備工具番号を使用して工具を指定できます。

注記

この PI サービスの前後で、パラメータが「WVar1」=x (FDPLMT)の PI サービス MMCSEM を呼び出してください。「PI サービス MMCSEM (ページ 1211)」の章を参照してください。

位置の指定

マルチ工具内で位置決めする工具は、次の 3 タイプのうちのいずれかとして指定できます:

- 「Addr1」= <空の文字列>、「WVar1」= <工具番号>、および「WVar2」= <予備工具番号>
- 「Addr1」= <工具名>、「WVar1」= -1、および「WVar2」= <予備工具番号>
- 「Addr1」= <空の文字列>、「WVar1」= <工具番号>、および「WVar2」= -1

位置の指定は、以下を介して割り当てることもできます。

- "WVar4" = <マルチツールのロケーション番号>

仮パラメータの説明

信号	タイプ	値の範囲	説明
PIService :	ANY	PI.FDPLMT	マルチ工具内の空き工具ロケーションを検索/チェック
単位:	INT	1, 2, 3 ... 10	TOA
Addr1:	STRING		マルチ工具内に位置決めされる工具の工具名(最大 32 文字)
WVar1:	INT	-1, 1, 2, 3, ... 32000	マルチ工具内に位置決めされる工具の工具番号(T 番号) -1:"Addr1" (工具名)および"WVar2" (予備工具番号)が使用されます。 1, 2, 3 ... 32000:工具番号
WVar2:	INT	-1, 1, 2, 3, ... 32000	マルチ工具内に位置決めされる工具の予備工具番号 1, 2, 3, ... 32000:予備工具番号 -1:"WVar1" (T 番号)が使用されます

信号	タイプ	値の範囲	説明
WVar3:	INT	-1, 1, 2, 3, ... 32000	マルチツールの番号 1, 2, 3, ... 32000:マルチツールの番号 -1:すべてのマルチ工具での空きロケーションの検索、または、工具を収納するために指定された工具ロケーションが空いているマルチ工具があるかどうかを調べるためのすべてのマルチ工具のチェック。
WVar4:	INT	-1, 1, 2, 3, ... 999	システムが位置決めされる工具ロケーションのマルチ工具ロケーション番号 > 0:マルチツールのロケーション番号 -1:マルチ工具内でのあらゆる空の工具ロケーションの検索

14.17.5 FB5:GETGUD - GUD 変数の読み取り

機能

ファンクションブロック FB5「GETGUD」を使用して、NC またはチャネル領域のグローバルユーザーデータ(GUD)を読み取ります。

NC 変数の読み取り要求

立ち上がりエッジによる FB5 の呼び出し、変更、パラメータ「Req」= 0→1

パラメータ「Addr」:GUD 変数の名前へのポインタ。シンボルを使用して「<データブロック>」.<変数名>

パラメータ「Area」、「Unit」、「Index1」、および「Index2」:変数のアドレス指定の追加情報

パラメータ「CnvtToken」が有効である場合、ユーザーは、読み取る GUD 変数のトークン(変数ポインタ)を受信します。これを使用して、パラメータ「Addr1」...「Addr8」=「<トークン>」の FB2 および FB3 を介して、GUD 変数の読み取りと書き込みをおこなうことができます。トークンを使用したアドレス指定は、GUD 変数の**書き込み**に必須で

す。GUD 変数配列のアドレス指定のためにトークンを使用している場合、FB2/FB3 のパラメータ"Line1" ..."Line8" = "<配列インデックス>"と値も提供する必要があります。

DOUBLE タイプの GUD 変数を、フォーマットを調整せずに読み取るには、REAL2 タイプの ANY ポインタを対称領域で指定してください。たとえば、P#M100.0 REAL 2 などです。これで、DOUBLE タイプの GUD 変数の値が、PLC データ領域で 64 ビットの浮動小数点数として使用されます。

読み取り要求の完了

読み取り要求が正常に終了:パラメータ「Done」== 1.

読み取り要求が異常とともに終了:パラメータ「Error」== 1、異常の原因はパラメータ「State」

前提条件

- OB100 による NC/PLC 通信の解放、FB1 パラメータ「NCKomm」= 1
- GUD 変数名の文字列を含むデータブロックでは、S7 プロジェクトの S シンボルテーブルでシンボルを割り当てるために S7 シンボルエディタを使用する必要があります(以下の呼び出し例 1 を参照:DB_GUDVAR)。データブロックでは、GUD 変数名を含めるために適切な長さの文字列を作成する必要があります(以下の呼び出し例 1 を参照:"DB_GUDVAR".GUDVar1)。次に、このシンボルは FB5 の"Addr"パラメータに渡されます(たとえば、Addr := "DB_GUDVAR".GUDVar1 など)。FB2/FB3 の以下の呼び出しで GUD 変数のトークンが必要な場合、DB (たとえば、同じ DB)で 10 バイトのトークン構造を作成する必要があります(以下の呼び出し例を参照:"DB_GUDVAR".GUDVar1Token)。この構造は、FB5 の VarToken パラメータで指定します(以下の呼び出し例 1 を参照:FB5、VarToken := "DB_GUDVAR".GUDVar1Token。次にこのトークンは、FB2/FB3 の呼び出し時にパラメータ"Addr1" … "Addr8"で指定します(以下の呼び出し例 1 を参照:FB3, Addr1 := "GUDVar1Token".

一般条件

- FB5 には、マルチインスタンス機能があります。
- FB5 呼び出し毎に、ユーザー領域から個別のインスタンス DB を割り当ててください。
- GUD 変数の読み取り(Req = 1 での FB5 呼び出し)は、PLC 基本プログラム(OB1)のサイクリックセクションでのみ許容されます。ジョブが開始されない(「Req」= 0 のFB5 呼び出し)場合は、PLC 基本プログラム(OB100)の起動部分にもパラメータを書き込むことができます。その後、ジョブは、Req = 1 の FB5 を呼び出すことで、PLC 基本プログラム(OB1)のサイクリックセクションにすでに書き込まれているパラメータを使用して実行することができます。

- GUD 変数の名前に使用できるのは、大文字のみです。
- GUD 変数の読み取りは、通常、数回の PLC 周期に及びます。

注記

異常ケース
別のチャネルから変数を読み取る場合、次のフィードバックメッセージが出力されます:

- 「Error」 == TRUE
- 「State」 == W#16#02

注記

DOUBLE タイプの変数を NC から、フォーマットを調整せずに読み取るには、REAL2 タイプの ANY ポインタを対称領域で指定してください(例:(P#M100.0 REAL 2 など)。DOUBLE タイプの変数の読み取り時に基本プログラムが REAL 2 をターゲットタイプと認識した場合は、このデータが、64 ビットの浮動小数点数として PLC データ領域に適用されます。

ファンクションの宣言

```
FUNCTION_BLOCK FB5          //Server name
   KNOW_HOW_PROTECT
   VERSION :3.0
VAR_INPUT
   Req :          BOOL;
   Addr:          ANY ;      //変数名
   Area           BYTE;      //Area:NCK = 0, channel = 2
   Unit :         BYTE;
   Index1:        INT;       //配列インデックス 1
   Index2:        INT;       //配列インデックス 2
   CnvtToken:     BOOL;      //Conversion into 10-byte token
   VarToken       ANY ;      //Struct with 10 bytes for the variable token
END_VAR

VAR_OUTPUT
   Error :        BOOL;
   Done :         BOOL;
   State :        WORD;
END_VAR
```

```
VAR_IN_OUT
   RD:            ANY ;
END_VAR

BEGIN
END_FUNCTION_BLOCK
```

仮パラメータの説明

表記	種別	タイプ	数値の範囲	説明
Req :	I	BOOL		信号の立ち上がりエッジによりジョブが起動
Addr:	I	ANY	「DBName」.<VarName>	STRING タイプの変数の変数名
Area:	I	BYTE	0, 2	範囲: 0:NC 2:チャネル
単位:	I	BYTE	1, 2, 3 ... 10	Area == NC:Unit:=1 Area ==チャネル:チャネル番号
Index1:	I	INT	0、1、2、... <最大配列インデックス>	配列インデックス 1 配列インデックスのない変数の場合:Index1 = 0
Index2:	I	INT	0、1、2、... <最大配列インデックス>	配列インデックス 2 2 番目の配列インデックスのない変数の場合:Index2 = 0
CnvtToken:	I	BOOL	0 (FALSE)、1 (TRUE)	10 バイトの変数トークンの生成の起動
VarToken:	I	ANY	「DBName」.<VarName>	10 バイトトークンのアドレス(例を参照)
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	ジョブの否定応答、またはジョブの実行不可
Done :	O	BOOL	0 (FALSE)、1 (TRUE)	ジョブが正常終了

表記	種別	タイプ	数値の範囲	説明
State :	O	WORD	---	段落「エラー ID」を参照
RD:	I/O	ANY	P#Mm.n BYTE x... P#DBnr.dbxm.n BYTE x	読み取りデータの対象領域

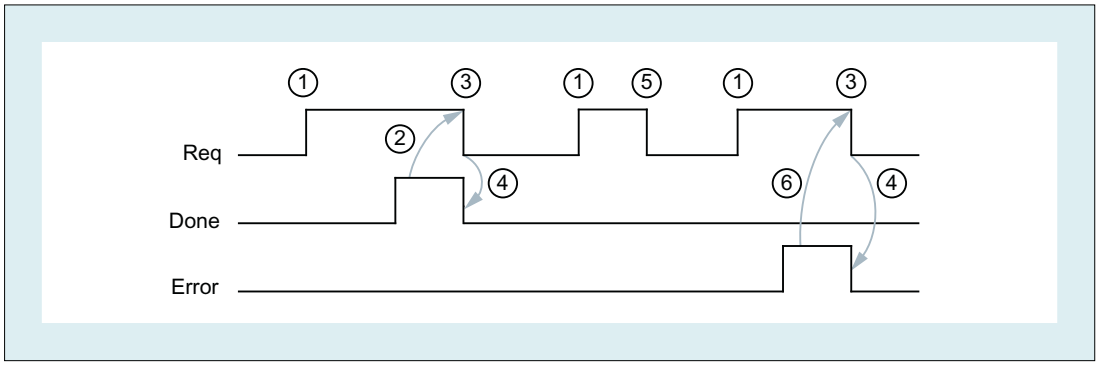
異常識別子

State		説明	注
WORD H [1]	下位 WORD		
0	1	アクセス異常	
0	2	ジョブ内の異常	ジョブ内の変数の間違ったコンパイル
0	3	否定応答、ジョブ実行不可	内部異常、 NC RESET を試行
0	4	データ領域またはデータタイプが不一致	RD で読み取られる データを確認
1	4	空きローカルユーザーメモリが不足	読み取り変数が RD の指定より長い
0	6	FIFO が一杯	待ち行列が一杯になったため、 ジョブを繰り返してください
0	7	オプションが未設定	BP パラメータ「NCKomm」が未設定
0	8	間違った対象領域(RD)	RD をローカルデータにすることはできません
0	9	転送が占有されています	ジョブを繰り返してください
0	10	アドレス指定の異常	Unit に値 0 が含まれます
0	11	変数のアドレスが無効	Addr (または変数名)、Area、Unit をチェック
1 - 8	13	間違った ANY データ参照	必要な文字列/NcVar データが、パラメータ設定されていません
0	15	32 文字を超える文字列	GUD 変数名が長すぎます
1) 上位バイト> 0 ⇒異常が発生した変数の番号			

設定手順

GUD 変数を読み取れるようにするために、その名前は文字列変数で保存してください。「Add」パラメータを FB GETGUD に対してシンボルを使用して割り当てることができるように、この文字列変数のあるデータブロックをシンボルテーブルに定義してください。変数ポインタを受け取るために、構造変数をオプションで PLC の任意のデータ領域に定義できます(次の例の指定を参照してください)。

タイミングチャート



① User:要求の設定、Req = 0→1

② FB5 が正常に終了、Done = 1
User:要求のリセット、IF Done == 1 THEN Req = 0

③ User:IF Done == 1 THEN、要求のリセット:1 → 0

④ FB5 がジョブ確認をリセット、Done = 0

⑤ User:IF Done == 0 AND Error == 0 THEN、要求のリセット、Req = 1 → 0 **許容されません**

⑥ FB5 が異常終了、Error = 1
User:要求のリセット、IF Done == 1 OR Error == 1 THEN Req = 0、さらにエラー処理の可能性

呼び出し例 1

チャネル 1 からの GUD 変数の読み取り:

- 名前「GUDVAR1」
- タイプ:INTEGER

10 バイト変数ポインタへの変換。「FB2:GET - NC 変数の読み取り (ページ 1173)」の章の「データタイプの割り当て」にある表を参照してください

GUD 変数の読み取り:インスタンス DB111 の FB5

```
// Data block for GUD variable
DATA_BLOCK DB_GUDVAR    //シンボルテーブルでの割り当て

STRUCT
   GUDVar1 :STRING[32] := 'GUDVAR1';                //Name is defined by
                                                    user
   GUDVar1Token :
   STRUCT
      SYNTAX_ID :BYTE;
      area_and_unit :BYTE;
      column :WORD;
      line :WORD;
      block type :BYTE;
      NO.OF LINES :BYTE;
      type :BYTE;
      length :BYTE;
   END_STRUCT;
END_STRUCT;

BEGIN
END_DATA_BLOCK

// FB5 のインスタンスとして、未割り当てユーザー DB
DATA_BLOCK DB111

        FB5
BEGIN
END_DATA_BLOCK

// FB3 のインスタンスとして、未割り当てユーザー DB
DATE_BLOCK DB112

        FB3
BEGIN
END_DATA_BLOCK

//変数ポインタへの変換を伴う、チャネル 1 からのチャネル別 GUD 変数の読み取り
Function FC "VariablenCall" :VOID
U      I 7.7;           //未割り当て機械操作パネルキー
S      M 100.0;         //要求の起動
```

```
U     M 100.1;          //Done 完了メッセージ
R     M 100.0;          //ジョブの終了
U     I 7.6;            //Manual error acknowledgment
U     M 102.0;          //異常未処置
R     M 100.0;          //ジョブの終了
CALL FB5, DB111(
           Req         := M 100.0,             //Starting edge for reading
           Addr        := "DB_GUDVAR".GUDVar1, // GUD の名称
           Area        := B#16#2,              //Channel variable
           Unit        := B#16#1,              //チャネル 1
           Index1      := 0,                   //配列インデックス 1 なし
           Index2      := 0,                   //配列インデックス 2 なし
           CnvtToken   := TRUE,                //要求:10 バイトトークンへの変換
           VarToken    := "DB_GUDVAR".GUDVar1Token, //トークンのアドレス
           Error       := M 102.0,
           Done        := M 100.1,
           State       := MW 104
           RD          := P#DB99.DBX0.0 DINT 1 // free memory area
);
```

GUD 変数の書き込み:インスタンス DB112 の FB3

FB5 からの GUD 変数トークン、パラメータ:「VarToken」(FB3 による書き込みの場合)、パラメータ「Addr1」

```
CALL FB3, DB112(
           Req         := M 200.0,
           NumVar      := 1,                     // 1 つの GUD 変数
           Addr1       :=                        // トークン
                       "DB_GUDVAR".GUDVar1Token,
           Error       := M 102.0,
           Done        := M 100.1,
           State       := MW 104
           SD1         := P#DB99.DBX0.0 DINT 1
);
```

呼び出し例 2

チャネル 1 からの GUD 変数の読み取り:

- 名前「GUD_STRING」
- タイプ:STRING[32]

10 バイト変数ポインタへの変換。

GUD 変数の読み取り:インスタンス DB111 の FB5

```
// Data block for GUD variable
DATA_BLOCK DB_GUDVAR    //シンボルテーブルでの割り当て

STRUCT
    GUDVarS :STRING[32] := 'GUD_STRING';     //ユーザーが定義した名前
    GUDVarSToken :
    STRUCT
       SYNTAX_ID :BYTE;
       area_and_unit :BYTE;
       column :WORD;
       line :WORD;
       block type :BYTE;
       NO.OF LINES :BYTE;
       type :BYTE;
       length :BYTE;
    END_STRUCT;

    string_of_GUD :STRING[30];      // must at least be so long as
                                    // the definition of 'GUD_STRING'!
    new_name :STRING[30] := 'GUD_123';
END_STRUCT;

BEGIN
END_DATA_BLOCK

// FB5 のインスタンスとして、未割り当てユーザー DB
DATA_BLOCK DB111

        FB5
BEGIN
END_DATA_BLOCK

// FB3 のインスタンスとして、未割り当てユーザー DB
DATE_BLOCK DB112

        FB3
BEGIN
END_DATA_BLOCK
```

```
//変数ポインタへの変換を伴う、チャネル 1 からのチャネル別 GUD 変数の読み取り
Function FC "VariablenCall" :VOID
U      I 7.7;            //未割り当て機械操作パネルキー
S      M 100.0;          //要求の起動
U      M 100.1;          //Done 完了メッセージ
R      M 100.0;          //ジョブの終了
U      I 7.6;            //Manual error acknowledgment
U      M 102.0;          //異常未処置
R      M 100.0;          //ジョブの終了
CALL FB5, DB111(
            Req          := M 100.0,     //Starting edge for reading
            Addr         :=  DB_GUDVAR.GUDVarS,
            Area         := B#16#2,      //Channel variable
            Unit         := B#16#1,      //チャネル 1
            Index1       := 0,           //配列インデックスなし
            Index2       := 0,           //配列インデックスなし
            CnvtToken    := TRUE,        //要求:10 バイトトークンへの変換
            VarToken     := "DB_GUDVAR".GUDVarSToken, //トークンのアドレス
            Error        := M 102.0,
            Done         := M 100.1,
            State        := MW 104
            RD           := "DB_GUDVAR".string_of_GUD
);
```

GUD 変数の書き込み:インスタンス DB112 の FB3

FB5 からの GUD 変数トークン、パラメータ:「VarToken」(FB3 による書き込みの場合)、パラメータ「Addr1」

```
CALL FB3, DB112(
            Req          := M 200.0,
            NumVar       := 1,                  // 1 つの GUD 変数
            Addr1        := "DB_GUDVAR".GUDVarSToken, //トークン
            Error        := M 102.0,
            Done         := M 100.1
            State        := MW 104
            SD1          := "DB_GUDVAR".new_name
);
```

データタイプの割り当て

「FB2:GET - NC 変数の読み取り (ページ 1173)」の章の「データタイプの割り当て」にある表を参照してください

14.17.6 FB7:PI_SERV2 - PI サービス要求

機能

高い数値の「WVar」パラメータ(「WVar11」～「WVar16」)を除き、ファンクションブロック FB7 には、ファンクションブロック FB4 と同じ機能があります。ファンクションブロック FB4 の代わりにファンクションブロック FB7 の使用を推奨します。

詳細については、「FB4:PI_SERV - PI サービス要求 (ページ 1193)」の章を参照してください。

ファンクションの宣言

```
FUNCTION_BLOCK FB7
Var_INPUT
    Req :           BOOL;
    PIService :     ANY ;
    単位:           INT;
    Addr1:          ANY ;
    Addr2:          ANY ;
    Addr3:          ANY ;
    Addr4:          ANY ;
    WVar1:          WORD;
    WVar2:          WORD;
    WVar3:          WORD;
    WVar4:          WORD;
    WVar5:          WORD;
    WVar6:          WORD;
    WVar7:          WORD;
    WVar8:          WORD;
    WVar9:          WORD;
    WVar10:         WORD;
    WVar11:         WORD;
    WVar12:         WORD;
    WVar13:         WORD;
```

```
    WVar14:         WORD;
    WVar15:         WORD;
    WVar16:         WORD;
END_VAR
VAR_OUTPUT
    Error :         BOOL;
    Done :          BOOL;
    State :         WORD;
END_VAR
```

14.17.7 FB9:MtoN - 操作パネルの切り替え

機能

ファンクションブロック FB9 "MzuN"は、1 つ以上の NCU 制御モジュールにバスシステム経由で接続するオペレータコンポーネント(MCP/OP)の切り替えに使用します。

個々の運転ユニット(操作パネル)と NCU (PLC)の間のインターフェースは、データブロック DB19 内の M :N インターフェースです。FB9 は、これらのインターフェースの信号を使用します。

初期化、ライフサイン関し、および異常ルーチンのほかに、以下の基本機能もコントロールユニット切り替えのためのブロックにより実行されます。

機能の一覧表:	
基本機能	**説明**
HMI 待機中	HMI は NCU とのオンラインを要求しています
HMI 接続中	HMI は NCU に接続しています
HMI 切断中	HMI は NCU から切り離されています
強制切断	HMI で NCU との接続を切断しなくてはなりません
操作対象のサーバモードへの切り替え	ある NCU からその他の NCU への操作対象の変更
アクティブ/パッシブ操作モード:	オペレータ制御と監視/監視のみ
MCP 切り替え	オプションとして、MCP を HMI により切り替えることができます

注記

このブロックは、ユーザープログラムにより呼び出してください。このために、任意の番号のインスタンス DB を準備してください。この呼び出しはマルチインスタンス対応です。

いくつかの重要機能の概略説明

アクティブ/パッシブ操作モード:

オンライン HMI は、2 つの異なるモードで操作できます。

アクティブモード:	オペレータは、制御と監視ができます
パッシブモード:	オペレータは監視できます(HMI ヘッダのみ)

NCU への切り替えの後、これにより最初にオンライン NCU の PLC でアクティブ操作モードが要求されます。2 つのコントロールユニットがオンラインで同時に NCU に接続された場合、2 つのうち 1 つは必ずアクティブモードであり、もう 1 つはパッシブモードです。オペレータは、パッシブ HMI でボタンを押してアクティブモードを要求できます。

MCP 切り替え

オプションとして、HMI に割り当てられた MCP は同時に切り替えることができます。これを実行するには、MCP アドレスを HMI の NETNAMES.INI 設定ファイルの「mstt_adress」パラメータに入力し、「MCPEnable」を TRUE に設定してください。NCU で有効な MCP は一度に 1 つだけであるように、パッシブ HMI の MCP は無効にされます。

起動条件

NCU の再起動で、OB100 の FB1 の呼び出し時に、以前に選択した MCP が起動することを防ぐために、入力パラメータ"MCP1BusAdr"を**= 255** (第 1 MCP のアドレス)、"MCP1Stop"を**= TRUE** (第 1 MCP のスイッチオフ)に設定する必要があります。

認証規格

1 つの MCP が別の MCP に切り替えられたとき、すべての送り有効または軸イネーブルは維持されます。

注記

切り替えのときに作動したキーは、(後で起動される HMI により)新しい MCP が起動されるまで動作したままになります。送り速度および主軸のオーバライド設定も有効のままです。作動したキーを無効にするには、DB10.DBX104.01(MCP 1 準備完了)の立ち下がりエッジで、機械制御信号の入力イメージを非作動信号レベルに切り替えてください。オーバーライド設定は変更しないでください。キーを無効にするための手段は、PLC ユーザープログラムで実行してください (例「オーバライド切り替え」を参照)。

この呼び出しを実行できるのは、OB1 サイクリックプログラムの場合に限ります。

ファンクションの宣言

```
FUNCTION_BLOCK FB9
VAR_INPUT
    [Quit]          BOOL;           //割り込みの応答
    OPMixedMode:    BOOL := FALSE;  //M 対 N が有効でない OP による混合操作//無効
    ActivEnable:    BOOL:= TRUE;    //サポートされていません
    MCPEnable :     BOOL:= TRUE;    //MCP 切り替えの起動
END_VAR
VAR_OUTPUT
    Alarm1 :        BOOL;           //アラーム:HMI バスアドレス、バスタイプの異常
    Alarm2 :        BOOL;           //アラーム:HMI 1 オフラインの確認なし
    Alarm3 :        BOOL;           //アラーム:HMI 1 がオフラインになっていない
    Alarm4 :        BOOL;           //アラーム:HMI 2 オフラインの確認なし
    Alarm5 :        BOOL;           //アラーム:HMI 2 がオフラインになっていない
    Alarm6 :        BOOL;           //アラーム:待機中の HMI がオンラインになっていない
    Report :        BOOL;           //メッセージ:サインオブライフモニタ
    ErrorMMC :      BOOL;           //異常検出 HMI
END_VAR
```

仮パラメータの説明

信号	種別	タイプ	値の範囲	説明
[Quit]	I	BOOL	0 (FALSE)、1 (TRUE)	割り込みの応答
OPMixedMode:	I	BOOL	0 (FALSE)、1 (TRUE)	M 対 N 機能のない OP に対して無効にされる混合操作
ActivEnable:	I	BOOL	0 (FALSE)、1 (TRUE)	機能はサポートされていません。操作パネルの切り替え。DB19 で MMCx_SHIFT_LOCK を使用したインターロック
MCPEnable :	I	BOOL	0 (FALSE)、1 (TRUE)	MCP 切り替えの有効化 1:MCP は操作パネルによって切り替えられます。 0:MCP は操作パネルによって切り替えられません。これを使用して一つの MCP を常に接続できます。DB19 の MMCx_MSTT_SHIFT_LOCK も参照してください。
Alarm1 :	O	BOOL	0 (FALSE)、1 (TRUE)	アラーム:HMI バスアドレス、バスタイプの異常
Alarm2 :	O	BOOL	0 (FALSE)、1 (TRUE)	アラーム:HMI 1 オフラインの確認なし
Alarm3 :	O	BOOL	0 (FALSE)、1 (TRUE)	アラーム:HMI 1 がオフラインになっていない
Alarm4 :	O	BOOL	0 (FALSE)、1 (TRUE)	アラーム:HMI 2 オフラインの確認なし
Alarm5 :	O	BOOL	0 (FALSE)、1 (TRUE)	アラーム:HMI 2 がオフラインになっていない
Alarm6 :	O	BOOL	0 (FALSE)、1 (TRUE)	アラーム:待機中の HMI がオンラインになっていない
Report :	O	BOOL	0 (FALSE)、1 (TRUE)	Message:ライフサイン監視 HMI
ErrorMMC :	O	BOOL	0 (FALSE)、1 (TRUE)	//異常検出 HMI

FB9 の呼び出し例

```
CALL FB9, DB109(
Ackn            := Error_ack,       //MCP リセットなど
OPMixedMode     := FALSE,
ActivEnable     := TRUE,
MCPEnable       := TRUE);           // MCP 切り替え有効
```

注記

MCP の切り替えを有効にするために入力パラメータ"MCPEnable"を TRUE に設定する必要があります。これらのパラメータの初期値はこの方法で設定するため、ファンクションの呼び出し時に特に割り当てる必要はありません。

アラーム、異常

出力パラメータ「Alarm1」～「Alarm6」と「Report」は PLC に情報として存在し、アラーム 410900 - 410906 の表示により HMI に表示される M 対 N 異常の場合に出力されます。

HMI 機能の実行が失敗した場合(さらに適切なエラーメッセージを表示できない場合)、状態パラメータ「ErrorMMC」は「logical 1」に設定されます(起動異常、接続が行われないときなど)。

FB の呼び出しの例(OB100 での呼び出し)

```
CALL "RUN_UP", "gp_par" (
    MCPNum              := 1,
    MCP1In              := P#I 0.0,
    MCP1Out             := P#Q 0.0,
    MCP1StatSend        := P#Q 8.0,
    MCP1StatRec         := P#Q 12.0,
    MCP1BusAdr          := 255,             // 第 1 MCP のアドレス
    MCP1Timeout         := S5T#700MS,
    MCP1Cycl            := S5T#200MS,
    MCP1Stop            := TRUE,            // MCP スイッチオフ
    NCCyclTimeout       := S5T#200MS,
    NCRunupTimeout      := S5T#50S);
```

例:オーバーライドの切り替え

```
//使用される補助フラグ M100.0、M100.1、M100.2、M100.3
//MCP1Ready の立ち上がりエッジでは、オーバライドと
//起動方法をチェックしてください
// MCP ブロックの開始
//この例は、送り速度のオーバーライドに適用されます。
//主軸オーバライドに対しては、インタフェースと入力バイトを変更してください。
U     DB10.DBX104.0       //MCP1Ready
EN    M   100.0;          //エッジメモリビット 1
JCN smth1;
S     M   100.2;          //補助メモリビット 1 の設定
R     M   100.3;          //補助フラグ 2 のリセット


//オーバライドの保存
      L DB21.DBB4;        //送り速度オーバライドインタフェース
      T EB28;             //保持バッファ(自由入力
                          //またはフラグバイト)

wei1:
U     M   100.2;          //切り替え実行
O     DB10.DBX104.0       //MCP1Ready
JCN smth2;
U     DB10.DBX104.0       //MCP1Ready
FP    M   100.1;          //エッジメモリビット 2
JC smth2;
U     M   100.2;          //切り替え実行
R     M   100.2;          //補助フラグ 1 のリセット
JC smth2;
U     M   100.3;          //比較が実行された
SPB MCP;                  //MCP プログラムの呼び出し
//保存されたオーバライドを切り替えられる MCP のインタフェースに転送
//オーバライド値が一致するまで
      L EB28;             //保持バッファオープン
      DB21.DBB4:          //オーバライドインタフェースの転送
      L EB3;              //送り速度用オーバライド入力バイト
      <>i;                //一致しているか
JC smth2;                 //いいえ、ジャンプ
S     M   100.3;          //はい、補助フラグ 2 の設定
//オーバライド値が一致したときは、MCP プログラムを再度呼び出し
MCP:CALL "MCP_IFM"(       FC19:
      BAGNo         := B#16#1,
```

```
      ChanNo       := B#16#1,
      SpindleIFNo  := B#16#0,
      FeedHold     := M 101.0,
      SpindleHold  := M 101.1);
wei2:NOP           0:
```

14.17.8 FB10:安全リレー(SI リレー)

機能

PLC における「Safety Integrated」のための SPL ファンクションブロック FB10「安全リレー」は、同じ名称の NC 機能に相当します。標準の SPL「安全リレー」ブロックは、安全プログラマブルロジックを持つ非常停止機能の実装をサポートするように設計されています。ただし、これを使用して、防護ドアの制御などの他の類似した安全機能を実装することもできます。

このファンクションには、3 つの入力パラメータ(「In1」、「In2」、「In3」)が含まれます。これらのパラメータの 1 つを値 0 に切り替えるとき、出力「Out0」は遅延なしで解除され、出力「Out1」、「Out2」、および「Out3」は、パラメータ設定されたタイマ値(パラメータ「TimeValue1」、「TimeValue2」、「TimeValue3」)を介して解除されます。入力「In1」～「In3」が値 1 になり、確認入力「Ack1」、「Ack2」のうちの 1 つで立ち上がりエッジ変化が検出された場合、出力は遅延なしに再度起動されます。

呼び出しビットメモリ

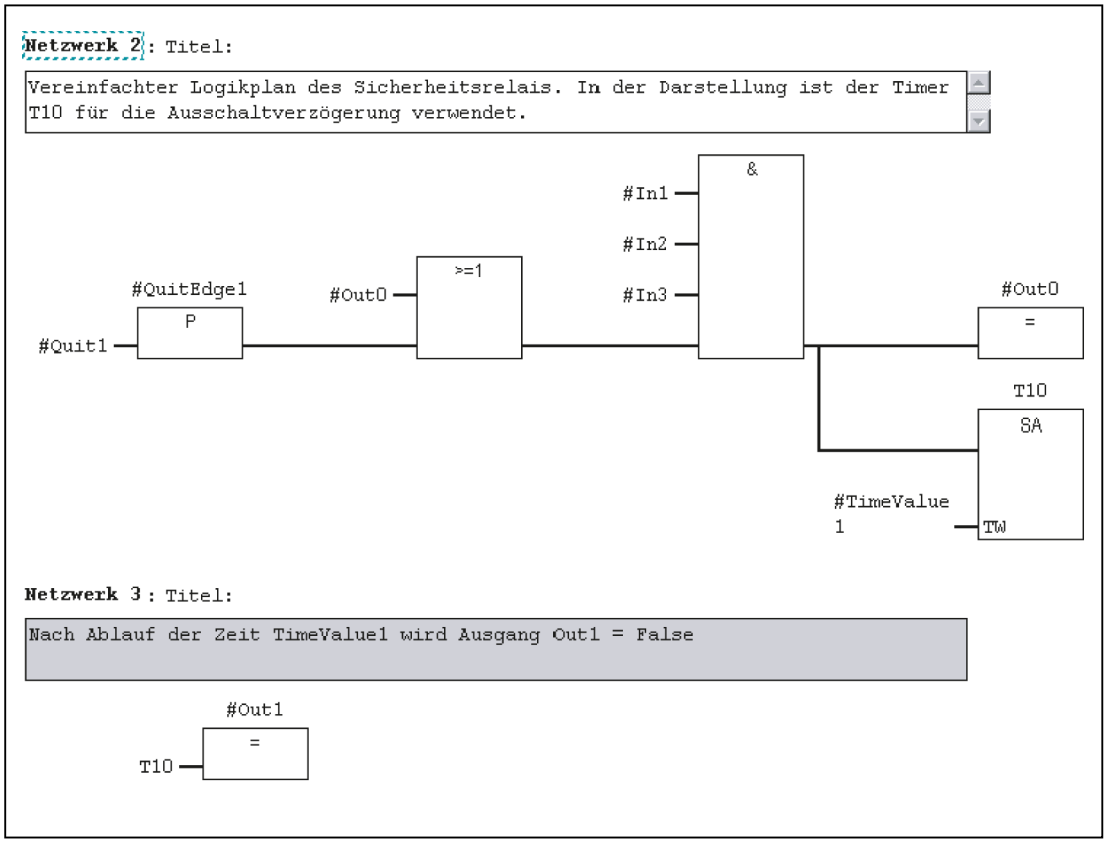
起動の後で出力を初期設定(値 = 0)にするには、パラメータ「FirstRun」を以下のように設定してください。パラメータ「FirstRun」は、コントローラ起動後の最初の実行時に保持データ(メモリビット、データブロックのビット)を介して値 TRUE に切り替えてください。このデータは、OB100 などに事前に設定することができます。パラメータ「FirstRun」は、FB10 の初回の実行時に FALSE にリセットされます。自身のインスタンスを持つ各呼び出し用のパラメータ「FirstRun」には、別々のデータを使用してください。

一般条件

- FB10 には、マルチインスタンス機能があります。
- SPL プログラムの起動後は、PLC 基本プログラム(OB1)のサイクリックセクションで SI リレー毎に 1 回、FB10 が呼び出されます。
- すべての FB10 呼び出しに、ユーザー領域から個別のインスタンス DB を割り当ててください。

CSF の簡易ブロック図

以下の図は、1 つの確認入力「Ack1」と 1 つの遅延解除出力「Out1」を示しています。「Ack2」用回路とその他の遅延出力は同じです。この機能図には、パラメータ「FirstRun」もありません。動作モードは、上に記述されています。



ファンクションの宣言

```
FUNCTION_BLOCK FB10
VAR_INPUT
    In1 :           BOOL;
    In2 :           BOOL;
    In3 :           BOOL;
    Ackn1 :         BOOL;
    Ackn2 :         BOOL;
    TimeValue1 :    TIME;
    TimeValue2 :    TIME;
```

```
    TimeValue3 :    TIME;
END_VAR
VAR_OUTPUT
    Out0            BOOL;
    Out1            BOOL;
    Out2 :          BOOL;
    Out3 :          BOOL;
END_VAR
VAR_INOUT
    FirstRun:       BOOL;
END_VAR
```

仮パラメータの説明

パラメータ	種別	タイプ	値の範囲	説明
In1:	I	BOOL	0 (FALSE)、1 (TRUE)	入力 1
In2:	I	BOOL	0 (FALSE)、1 (TRUE)	入力 2
In3:	I	BOOL	0 (FALSE)、1 (TRUE)	入力 3
Ackn1 :	I	BOOL	0 (FALSE)、1 (TRUE)	確認入力 1
Ackn2 :	I	BOOL	0 (FALSE)、1 (TRUE)	確認入力 2
TimeValue1 :	I	S5TIME		遅延 OFF の時間 1
TimeValue2 :	I	S5TIME		遅延 OFF の時間 2
TimeValue3 :	I	S5TIME		遅延 OFF の時間 3
Out0	O	BOOL	0 (FALSE)、1 (TRUE)	出力、即時(遅延なし)
Out1	O	BOOL	0 (FALSE)、1 (TRUE)	出力、TimeValue1 の分だけ遅延
Out2 :	O	BOOL	0 (FALSE)、1 (TRUE)	出力、TimeValue2 の分だけ遅延
Out3 :	O	BOOL	0 (FALSE)、1 (TRUE)	出力、TimeValue3 の分だけ遅延
FirstRun:	I/O	BOOL	0 (FALSE)、1 (TRUE)	呼び出しビットメモリ

14.17.9 FB11:ブレーキテスト

機能

ブレーキ動作チェックは、保持ブレーキにより、軸が制御できない状態で移動されるのを防止するため、すべての軸に対して使用してください。主な用途は「垂直軸」です。

機械メーカは、PLC ユーザープログラムを使用して、適切な時間に定期的(8 時間毎など)にブレーキを閉じ、軸の重量に加えてトルク/推力をドライブで発生させることができます。正常な動作では、ブレーキが、必要な制動トルク/制動力を発生させることができます。この間、軸は、ほとんど移動しません。

異常が発生すると、位置フィードバック値は、パラメータ設定可能な監視ウィンドウを外れます。位置コントローラは、軸がたるまないようにします。ブレーキ機械システムのファンクションテストにより、否定的な確認応答がおこなわれます。

参照先

「Safety Integrated」ファンクション の仕組みの中での NC とドライブのパラメータ設定の詳細な説明は、以下の資料にあります。

Safety Integrated のファンクションマニュアル

ブレーキテストの開始

ブレーキテストは、必ず軸が停止状態のときに開始してください。ブレーキテスト期間全体に対して、パラメータ設定された軸の使用可能信号を有効に設定してください(コントローラディセーブル、送り有効など)。さらに、テスト期間全体に対して、ユーザープログラムにより、軸/主軸 DB31、 ... .DBX28.7 (PLC 制御軸)の信号をステータス 1 に設定します。

NC/PLC インターフェース DB31、... .DBX28.7 (PLC 制御軸)の起動前に、軸は「中立軸」に切り替えられ(例:DB31、... .DBX8.0 ～ 8.3 (NC 軸のチャネルへの割り当て))、チャネル 0 に設定されます。また、DB31、... .DBX8.4 (このバイトの変更時の起動信号)も設定されます。

応答メッセージ:

- 現在の状態については、DB31、... DBB68 で確認できます。
- ブロックが起動される前に、信号 DB31、... .DBX63.1 (PLC は軸を制御します)を介して、NC を待機状態にしてください。ドライブがトルク/推力を発生させなければならない方向は、PLC により「移動動作」の形式で指定されます(FC18 を介してなど)。

このブレーキにより必要なトルク/推力を形成できない場合、軸はこの動作の移動先に衝突のリスクなしで到達できるようにしてください。

注記

FC18 に関する注記

ユーザープログラムの後の部分で FC18 が同じ軸に対して呼び出される場合は、これらの呼び出しを互いにインターロックしてください。たとえば、FC18 パラメータのためのインターロックされた共通のデータを持つ、このファンクションの共通の呼び出しを介して、これを実行できます。2 番目の選択肢は、無効な FC18 がプログラムにより処理されない状態で、FC18 を繰り返し呼び出すことです。複数使用のインタロックを提供してください。

ブレーキテストの構造

ステップ		予想されるフィードバック	監視時間
1	ブレーキテストの開始	DBX71.0 = 1	TV_BTactiv
2	ブレーキ拘束	Bclosed = 1	TV_Bclose
3	移動指令の出力	DBX64.6 または DBX64.7	TV_FeedCommand
4	テスト移動指令の発行	DBX62.5 = 1	TV_FXSreached
5	保持時間待ち	DBX62.5 = 1	TV_FXShold
6	ブレーキテストの解除/ブレーキの開放	DBX71.0 = 0	TV_BTactiv
7	出力テスト良好	---	---

一般条件

- FB2 には、マルチインスタンス機能があります。
- すべての FB11 呼び出しに、ユーザー領域から個別のインスタンス DB を割り当ててください。

ファンクションの宣言

```
FUNCTION_BLOCK FB11
VAR_INPUT
    開始                BOOL;
    [Quit]              BOOL;
```

```
    Bclosed           BOOL;
    Axis :            INT;
    TimerNo :         TIMER ;
    TV_BTactiv        S5TIME ;
    TV_Bclose         S5TIM;
    TV_FeedCommand    S5TIME ;
    TV_FXSreached     S5TIME ;
    TV_FXShold        S5TIME ;
END_VAR
VAR_OUTPUT
    CloseBrake    BOOL;
    MoveAxis :    BOOL;
    Done :        BOOL;
    Error :       BOOL;
    State : :     BYTE;
END_VAR
```

仮パラメータの説明

信号	タイプ	タイプ	説明
開始	I	BOOL	ブレーキテストの開始
Ackn	I	BOOL	故障リセット
Bclosed	I	BOOL	ブレーキの閉鎖が起動されたかどうかのアンサーバック入力(単一チャネル - PLC)
軸	I	INT	**テストされる軸の軸番号**
TimerNo	I	TIMER	ユーザープログラムのタイマ
TV_BTactiv	I	S5TIME	監視時間 → ブレーキテスト有効、軸信号 DBX71.0 の確認
TV_Bclose	I	S5TIME	監視時間 → ブレーキの動作、出力 CloseBrake が設定された後の入力信号 Bclosed のチェック。
TV_FeedCommand	I	S5TIME	監視時間 → 移動指令発行、MoveAxis が設定された後の移動指令のチェック
TV_FXSreched	I	S5TIME	監視時間 → 突き当て点に到達

信号	タイプ	タイプ	説明
TV_FXShold	I	S5TIME	監視時間 → ブレーキのテスト
CloseBrake	O	BOOL	要求、ブレーキ動作
MoveAxis	O	BOOL	要求、移動動作の開始
Done	O	BOOL	テストが正常に終了
異常	O	BOOL	異常が発生しました
状態	O	BYTE	段落「エラー ID」を参照

故障 ID

状態	説明
0	エラーなし
1	開始条件を満たしていません。たとえば、軸が閉ループ制御でないか、ブレーキが閉じているか、軸がディセーブルです。
2	ブレーキテストの選択での「ブレーキテスト有効」信号に NC アンサーバックなし
3	入力信号 Bclosed による「ブレーキ適用」アンサーバックなし
4	移動指令出力なし(軸の動作が開始されていないなど)
5	突き当て点に到達しません → 軸のリセットが開始されました。
6	移動禁止/アプローチが遅すぎ→突き当て点に到達できません。TV FXSreached 監視タイムアウト
7	ブレーキがまったく保持していません(終点の位置に到達しました)/アプローチ速度が速すぎます
8	保持時間中にブレーキが開放されています
9	ブレーキテストを解除するときの異常
10	内部エラー
11	ユーザープログラムで「PLC 制御軸」信号が有効ではありません

FB11 の呼び出しの例:

```
UN    M        111.1;   //FB から Z 軸のブレーキ動作の要求
=     O        85.0;    //ブレーキ制御、Z 軸
開             Axis3";  //ブレーキテスト、Z 軸
```

```
        O     I       73.0;    //ブレーキテスト起動、Z 軸
        O     M       110.7;   //ブレーキテスト実行
        FP    M       110.0;
        UN    M       111.4;   //異常が発生しました
        S     M       110.7;   //ブレーキテスト実行
        S     M       110.6;   //次のステップ
        JCN   m001
        L     DBB     68;
        AW    W#16#F;
        T     MB      115;     //フラグチャネルステータス
        L     B#16#10
        T     DBB     8;       //中立軸の要求

m001:   U     DBX     68.6;    //アンサーバック信号、軸が中立です
        U     M       110.6;
        FP    M       110.1;
        R     M       110.6;
        S     M       110.5;   //次のステップ
        S     DBX     28.7;    //PLC 制御軸の要求

        U     DBX     63.1;    //アンサーバック信号、PLC により制御される軸
        U     M       110.5;
        FP    M       110.2;
        R     M       110.5;
        S     M       111.0;   //FB のブレーキテストの開始

        CALL FB11, DB211 (//ブレーキテストブロック
              開始              :=M   111.0,   //ブレーキテストの開始
              Ackn              := I  3.7,     //RESET キーによる異常の確認
              Bclosed           := I  54.0,    //ブレーキ動作の応答メッセージ
                                               //制御
              軸                := 3,          //テストされる軸の軸番号
                                               //Z 軸
              TimerNo           := T  110,     //タイマ番号
              TV_BTactiv        := S5T#200MS,  //監視時間:
                                               //ブレーキテスト有効 DBX71.0
              TV_Bclose         := S5T#1S,     //監視時間:
                                               //ブレーキ動作
              TV_FeedCommand    := S5T#1S,     //監視時間:
                                               //移動指令の出力
              TV_FXSreache      := S5T#1S,     //監視時間:
                                               //突き当て点に到達
              TV_FXShold        := S5T#2S,     //監視時間:
                                               //ブレーキテスト時間
```

```
      CloseBrake        :=M   111.1,   //ブレーキ動作要求
      MoveAxis          :=M   111.2,   //移動動作の要求//起動
      Done              :=M   111.3,   //テストは正常終了
      異常              :=M   111.4,   //異常が発生しました
      状態              := MB 112);    //異常状態

   開            "Axis3"; //ブレーキテスト、Z 軸

   U     M       111.2;   //Moveaxis
   FP    M       111.5;   //FC18 の起動
   S     M       111.7;   //FC18 の起動

   O     M       111.3;   //テストは正常終了
   O     M       111.4;   //異常が発生しました
   FP    M       110.3;
   R     DBX     28.7;    //PLC 制御軸の要求

   UN    DBX     63.1;    //アンサーバック信号、PLC により制御される軸
   U     M       111.0;   //FB のブレーキテストの開始
   U     M       110.7;   //ブレーキテスト実行
   FP    M       110.4;
   R     M       111.0;   //FB のブレーキテストの開始
   R     M       110.7;   //ブレーキテスト実行

//選択の開始
   JCN   m002:
   L     MB      115;     //以前のチャネル状態
   OW    W#16#10;
   T     DBB     8;       //チャネル軸の要求
m002: NOP   0;
//選択の終了

   CALL "SpinCtrl" (//Z 軸の移動
      開始      :=M      111.2,     //移動動作の開始
      ストップ  := FALSE,
      Funct     := B#16#5,         //モード:軸モード
      モード    := B#16#1,         //手順:インクレメンタル
      AxisNo    := 3,              //移動される軸の軸番号
                                   //軸 Z 軸
```

```
      位置      := -5.000000e+000, //移動距離:-5 mm
      FRate     := 1.000000e+003,  //送り速度:1000 mm/min
      InPos     :=M      113.0,    //位置決め完了
      異常      :=M      113.1,    //異常が発生しました
      状態      := MB    114);     //異常状態

開             "Axis3"; //ブレーキテスト、Z 軸
U     M        113.0;   //位置決め完了
O     M        113.1;   //異常が発生しました
FP    M        113.2;
R     M        111.7;   //FC18 の起動
```

14.17.10 FB29:信号レコーダおよびデータトリガ診断

機能

信号レコーダ

FB29「診断」を使用すると、PLC ユーザープログラムでさまざまな診断処理を実行できます。診断処理を実行すると、信号状態および信号変更がログに記録されます。この診断処理では、機能番号 1 がパラメータ「Func」に割り当てられます。最大 8 つの信号(パラメータ「Signal_1」～「Signal_8」)が、いずれかの信号が変化するごとにリングバッファに記録されます。パラメータ「Var1」(BYTE 値)と「Var2」および「Var3」(INTEGER 値)の最新情報もリングバッファに保存されます。

それまでに処理された OB1 周期の数も追加情報としてリングバッファに保存されます。この情報により、信号および値を OB1 周期グリッドで図式的に評価できます。

呼び出し規則

OB1 周期での最初の FB29 呼び出し:パラメータ「NewCycle」= 1

同じ OB1 周期内の他のすべての FB29 呼び出し:パラメータ「NewCycle」= 0

リングバッファ

リングバッファは、ユーザーが定義しなければならず、ソースコードで指定したものと同じ ARRAY(配列)構造でなければなりません。この配列の要素数はいくつでもかまいません。推奨される要素数は 250 です。「ClearBuf」パラメータを使用して、リングバッファをクリアし、「BufAddr」ポインタを先頭に設定します。FB29 に関連するインスタンス DB は、ユーザー領域の DB です。この DB は、パラメータ「BufDB」を使用して診断 FB に伝送してください。

データトリガ

データトリガ機能の目的は、許容されるメモリセルにおける特定の値(またはビット)によるトリガを可能にすることです。トリガされるセルは、診断ブロックでパラメータ「TestVal」が比較される前に、ビットマスク(パラメータ「AndMask」)によって「丸め」られます。

注記

このファンクション用のソースコードは、「Diagnose.awl」という名前の基本プログラムライブラリのソースコンテナにあります。インスタンス DB およびリングバッファ DB も、このソースブロックで定義されています。ファンクション呼び出しについても、このファンクションで記述しています。DB 番号および呼び出しを変更してください。

ファンクションの宣言

```
FUNCTION_BLOCK FB29
VAR_INPUT
Func              :INT;
    Signal_1      :BOOL;
    Signal_2      :BOOL;
    Signal_3      :BOOL;
    Signal_4      :BOOL;
    Signal_5      :BOOL;
    Signal_6      :BOOL;
    Signal_7      :BOOL;
    Signal_8      :BOOL;
    NewCycle      :BOOL;
    Var1          :BYTE;
    Var2          :INT;
    Var3          :INT;
    BufDB         :INT;
    ClearBuf      :BOOL;
    DataAdr       :POINTER;
    TestVal       :WORD;
    AndMask       :WORD;
END_VAR
```

```
VAR_OUTPUT
    TestIsTrue     :BOOL;
END_VAR
VAR_IN_OUT
    BufAddr        :INT;
END_VAR
```

リングバッファの構造

```
TITLE =
                                        // FB29 のリングバッファ DB
VERSION :1.0

STRUCT
   フィールド:ARRAY [0 ..249 ] OF     //この構造体のサイズは任意
   STRUCT

   Cycle :INT;                         //バッファでの前の記憶までのデルタサイクル
   Signal_1 :BOOL;                     // FB29 と同じ信号名
   Signal_2 :BOOL;
   Signal_3 :BOOL;
   Signal_4 :BOOL;
   Signal_5 :BOOL;
   Signal_6 :BOOL;
   Signal_7 :BOOL;
   Signal_8 :BOOL;
   Var1 :BYTE;
   Var2 :WORD;
   Var3 :WORD;
   END_STRUCT;
END_STRUCT;
BEGIN
END_DATA_BLOCK
```

仮パラメータの説明

信号	タイプ	タイプ	値の範囲	説明
Func:	I	INT	0, 1, 2	機能 0:停止 1:信号レコーダ 2:データトリガ
機能 1 のパラメータ				
Signal_1 ... Signal_8:	I	BOOL	0 (FALSE)、1 (TRUE)	変化を監視されるビット信号
NewCycle	I	BOOL	0 (FALSE)、1 (TRUE)	前述の段落「機能」を参照
Var1 :	I	BYTE		追加の値
Var2 :	I	INT		追加の値
Var3 :	I	INT		追加の値
BufDB	I	INT		リングバッファ DB 番号
ClearBuf	I	BOOL	0 (FALSE)、1 (TRUE)	リングバッファ DB の削除およびポインタ BufAddr のリセット
BufAddr	I/O	INT		読み取りデータの対象領域
機能 2 のパラメータ				
DataAdr	I	POINTER		テスト対象ワードを指すポインタ
TestVal	I	Word		比較値
AndMask	I	Word		前述の段落「機能」を参照
TestIsTrue	O	BOOL	0 (FALSE)、1 (TRUE)	比較結果

設定手順

- 診断ブロックの機能を選択します。
- 信号レコーダまたはデータトリガとしての記録に適したデータを定義します。
- ユーザープログラム内で、診断 FB を呼び出すのに適したポイント(1 つまたは複数)を見つけます。

- リングバッファ用のデータブロックを作成します(呼び出し例を参照)。
- ユーザープログラムでパラメータ付きで診断 FB を呼び出します。

機能 1 では、パラメータ「ClearBuf」を使用してリングバッファをクリアすることをお勧めします。機能 1 による記録段階が完了したら、「データビューでデータブロックを開く」機能を使用して STEP7 でリングバッファ DB を読み取ります。これで、リングバッファ DB の内容を分析できます。

呼び出し例

```
FUNCTION FC99:VOID
TITLE =
VERSION :0.0

BEGIN
NETWORK
TITLE = NETWORK

CALL FB29, DB80(
Func                := 1,
        Signal_1    :=M         100.0,
        Signal_2    :=M         100.1,
        Signal_3    :=M         100.2,
        Signal_4    :=M         100.3,
        Signal_5    :=M         10.4,
        Signal_6    :=M         100.5,
        Signal_7    :=M         100.6,
        Signal_8    :=M         100.7,
        NewCycle    := TRUE,
        Var1        := MB       100,
        BufDB       := 81,
        ClearBuf    :=M         50.0);
END_FUNCTION
```

14.17.11 FC2:GP_HP - 基本プログラム、サイクリックセクション

機能

NC/PLC インターフェースは、基本プログラム によってサイクリックモード(OB1)で処理されます。ランタイムを最小限に保つために、制御信号と状態信号のみが周期的に伝送されます。補助命令と G 命令の伝送は、NC からの要求によってのみ処理されます。

さらに、ハンドル選択のデータ、モードおよび他の運転信号が操作パネル(HMI)から NCK/PLC インターフェースに伝送されるため、必要に応じて、MCP または HMI からの選択がモードによりサポートされます。

NC/PLC インターフェースへの HMI 信号の伝送は、FB1 (DB7)でパラメータ「MMCToIF」の値を「FALSE」に設定して無効にすることができます。

ハンドル選択信号

必要条件:FB1、パラメータ「HWheelMMC == TRUE」

HMI からのハンドル選択信号は、デコードされ、それぞれのハンドルのそれぞれの機械軸またはジオメトリ軸で有効になります。

宣言

```
FUNCTION FC2:VOID
//パラメータなし
```

呼び出し例

タイミングに関しては、この基本プログラムはユーザープログラム**より前に**実行しなければなりません。したがって、OB1 で最初に呼び出されます。

次の例には、OB1 の標準的な宣言と基本プログラムの呼び出し(FC2)、MCP 信号の伝送(FC19)、および異常/操作メッセージの取得(FC10)が含まれています。

```
ORGANIZATION_BLOCK OB1
VAR_TEMP
   OB1_EV_CLASS :    BYTE;
   OB1_SCAN_1 :      BYTE;
   OB1_PRIORITY :    BYTE;
   OB1_OB_NUMBR :    BYTE;
   OB1_RESERVED_1 :  BYTE;
   OB1_RESERVED_2 :  BYTE;
```

```
   OB1_PREV_CYCLE :  INT;
   OB1_MIN_CYCLE :   INT;
   OB1_MAX_CYCLE :   INT;
   OB1_DATE_TIME :   DATE_AND_TIME;
END_VAR
BEGIN
CALL FC2 (                  //最初の FC としての基本プログラムの呼び出し
//INSERT USER PROGRAM HERE
CALL FC19 (                 //インタフェースへの MCP 信号
BAGNo :=                    B#16#1,    //モードグループ番号 1
ChanNo :=                   B#16#1,    //チャネル番号 1
SpindleIFNo :=              B#16#4,    //主軸インタフェース番号 = 4
                                       // (関連する機械軸の数)
FeedHold :=                 m22.0,     //送り停止信号
                                       //モーダル
SpindleHold :=  db2.dbx151.0);         //主軸停止モーダル
                                       //メッセージデータブロックの
CALL FC10 (                            //異常メッセージおよび操作メッセージ
               ToUserIF := TRUE,       //DB2 から伝送される信号
                                       //インタフェースへ
               Ackn :=     I6.1);      //異常メッセージの確認応答
                                       // I6.1 経由
END_ORGANIZATION_BLOCK
```

14.17.12 FC3:GP_PRAL - 基本プログラム、割り込み起動セクション

機能

NC から PLC へのブロック同期伝送(補助命令および G 命令)は、基本プログラムのアラーム起動セクションで処理されます。**補助機能**は、通常の補助機能と高速の補助機能に分けられます。

NC ブロックの高速補助機能はバッファリングされ、NC に対する伝送の通知が行われます。これらの機能は、次回の OB1 周期の開始時にアプリケーションインターフェースに伝送されます。

次々と続けてプログラムされた高速補助機能は、ユーザープログラムで失われることはありません。これは、基本プログラムに含まれるメカニズムによって保証されています。

NC に対する通常の補助機能の通知は、1 サイクル全体の時間経過後にのみ行われます。これにより、アプリケーションが NC に対して読み込み停止を発行できるようになります。

G 命令は直ちに解析され、アプリケーションインターフェースに引き渡されます。

NC プロセスアラーム

NC によって割り込みが起動されると(IPO サイクルごとに可能)、OB40 のローカルデータ(「GP_IRFromNCK」)のビットが基本プログラムによって設定されます。ただし、FB1 パラメータ「UserIR」が TRUE の場合のみです。このデータは、他のイベント(I/O によるプロセスアラーム)では設定されません。この情報を利用することで、必要な処理を開始するために、ユーザープログラム内の関連割り込みルーチンに分岐できます。

機械用のユーザープログラムの高速ジョブ起動処理を実装するために、次の NC 機能を PLC ユーザープログラムの割り込み処理ルーチン(OB40 プログラムセクション)で使用できます。

- 選択された**補助機能**
- 工具管理オプション用の**工具交換機能**
- PLC を介して起動された位置決め軸、割り出し軸、および主軸の**位置決め完了**

上記の機能は、機械での応答を開始するために OB40 のユーザープログラムによって評価できる場合と、評価が必要な場合があります。たとえば、旋盤で T 命令がプログラムされている場合、リボルバ切り替えシステムを起動することができます。

プロセスアラーム(遅延時間や割り込み可否など)のプログラミングの詳細については、該当する SIMATIC 取扱説明書を参照してください。

補助機能

一般に、高速補助機能または確認応答補助機能は、それらがどのように割り当てられているかに関係なく、割り込み制御によって、または割り込み制御なしで処理されます。

ユーザープログラムによって割り込み起動方式でのみ処理しなければならない補助機能(T、H、DL)を定義するために、FB1 の基本プログラムパラメータを設定することができます。

割り込みを介して割り当てられない機能は、以前のバージョンと同様に周期的基本プログラムでのみ使用できます。このような機能の変更信号は、PLC サイクルで使用できます。

割り込み制御を使用して補助機能グループ(T、H、DL)が選択された場合でも、選択された機能について、ユーザープログラムによって処理できる割り込みは 1 つのみです。

1 ビットが、ユーザープログラムのローカルデータ「GP_AuxFunction」の特定のチャネルに設定されます(「GP_AuxFunction[1]」が設定された場合は、最初のチャネルに対して補助機能を使用できます)。

関連のチャネル DB で、変更信号および機能値を使用できます。この割り込み起動機能の要求信号は、全 OB1 周期が少なくとも 1 回が実行された後(最大で約 2 つの OB1 周期)、周期的基本プログラムセクションでゼロにリセットされます。

工具交換

工具管理オプションを使用すると、リボルバに対する工具交換命令と主軸の工具交換が割り込みによってサポートされます。このために、OB40 のローカルデータビット「GP_TM」を設定します。このようにして PLC ユーザープログラムで、工具交換機能用の工具管理 DB (DB72 または DB73)をチェックし、工具交換処理を開始できます。

位置決め完了

OB40 のローカルデータ「GP_InPosition」は、ビット構造で、機械軸別に構成されています(各ビットが 1 つの軸/主軸に対応します。たとえば、GP_InPosition[5]は第 5 軸に対応します)。

機能が軸または主軸に対して FC18 (主軸制御、位置決め軸、割り出し軸) によって起動されている場合は、当該の「GP_InPosition」ビットを使用して、上に挙げた FC の「InPos」信号の即時評価を実装することができます。この機能は、割り出し軸のクランプの起動を即座に行なうためなどに使用できます。

宣言

```
FUNCTION FC3:VOID
//パラメータなし
```

呼び出し例

タイミングに関しては、この基本プログラムは他のアラーム起動ユーザープログラム**より前に**実行しなければなりません。したがって、OB40 で最初に呼び出されます。

次の例には、OB40 の標準的な宣言と基本プログラムの呼び出しが含まれています。

```
ORGANIZATION_BLOCK OB40
VAR_TEMP
   OB40_EV_CLASS :      BYTE;
   OB40_STRT_INF :      BYTE;
   OB40_PRIORITY :      BYTE;
   OB40_OB_NUMBR :      BYTE;
   OB40_RESERVED_1 :    BYTE;
```

```
    OB40_MDL_ID :        BYTE;
    OB40_MDL_ADDR :      INT;
    OB40_POINT_ADDR :    DWORD;
    OB40_DATE_TIME :     DATE_AND_TIME;

//基本プログラムへの割り当て
GP_IRFromNCK :BOOL;                        //NC による割り込み
GP_TM :BOOL;                               //工具管理
GP_InPosition :ARRAY [1..3] OF BOOL;       //軸対応、位置決め用、
                                           //割り出し軸用、主軸用
GP_AuxFunction :ARRAY [1..10] OF BOOL;     //補助機能用チャネル対応
GP_FMBlock :ARRAY [1..10] OF BOOL;         //現在は未使用
//この位置から先に、ローカルユーザーデータをさらに定義可能
END_VAR
BEGIN
    CALL FC3 (
    //INSERT USER PROGRAM HERE
END_ORGANIZATION_BLOCK
```

14.17.13 FC5:GP_DIAG - 基本プログラム、診断アラーム、およびモジュール故障

機能

ブロック FC5「GP_DIAG」を使用して、組み立ての中断と故障を記録します。

PLC 停止は、パラメータ「PlcStop」でトリガできます。PLC 停止は、着信イベントに対してのみトリガされます。この場合、FB1 で設定されている PROFIBUS (DP1)に接続している MCP は除外されます。

宣言

```
FUNCTION FC5:VOID
   VAR_INPUT
      PlcStop:BOOL:= TRUE;
   END_VAR
```

呼び出し例

基本プログラムは、ユーザープログラムの処理後に実行してください。これが推奨される理由は、FC5 によって PLC 停止がトリガされる場合があることです。

例には、OB82 および OB86 の標準宣言と FC5 の呼び出しが含まれています。

```
ORGANIZATION_BLOCK OB82
VAR_TEMP
     OB82_EV_CLASS :BYTE;
     OB82_FLT_ID :BYTE;
     OB82_PRIORITY :BYTE;
     OB82_OB_NUMBR :BYTE;
     OB82_RESERVED_1 :BYTE;
     OB82_IO_FLAG :BYTE;
     OB82_MDL_ADDR :INT;
     OB82_MDL_DEFECT :BOOL;
     OB82_INT_FAULT :BOOL;
     OB82_EXT_FAULT :BOOL;
     OB82_PNT_INFO :BOOL;
     OB82_EXT_VOLTAGE :BOOL;
     OB82_FLD_CONNCTR :BOOL;
     OB82_NO_CONFIG :BOOL;
     OB82_CONFIG_ERR :BOOL;
     OB82_MDL_TYPE :BYTE;
     OB82_SUB_NDL_ERR :BOOL;
     OB82_COMM_FAULT :BOOL;
     OB82_MDL_STOP :BOOL;
     OB82_WTCH_DOG_FLT :BOOL;
     OB82_INT_PS_FLT :BOOL;
     OB82_PRIM_BATT_FLT :BOOL;
     OB82_BCKUP_BATT_FLT :BOOL;
     OB82_RESERVED_2 :BOOL;
     OB82_RACK_FLT :BOOL;
     OB82_PROC_FLT :BOOL;
     OB82_EPROM_FLT :BOOL;
     OB82_RAM_FLT :BOOL;
     OB82_ADU_FLT :BOOL;
     OB82_FUSE_FLT :BOOL;
     OB82_HW_INTR_FLT :BOOL;
     OB82_RESERVED_3 :BOOL;
     OB82_DATE_TIME :DATE_AND_TIME;
END_VAR
     BEGIN
        CALL FC5 (
           (PlcStop := FALSE) ;
END_ORGANIZATION_BLOCK
```

```
ORGANIZATION_BLOCK OB86
VAR_TEMP
     OB86_EV_CLASS :BYTE;
     OB86_FLT_ID :BYTE;
     OB86_PRIORITY :BYTE;
     OB86_OB_NUMBR :BYTE;
     OB86_RESERVED_1 :BYTE;
     OB86_RESERVED_2 :BYTE;
     OB86_MDL_ADDR :WORD;
     OB86_RACKS_FLTD :ARRAY [0 ..31]OF BOOL;
     OB86_DATE_TIME :DATE_AND_TIME;
END_VAR
     BEGIN
        CALL FC5 (
           (PlcStop := TRUE) ;
END_ORGANIZATION_BLOCK
```

14.17.14 FC6:TM_TRANS2 - 工具管理およびマルチツール用伝送ブロック

機能

ブロック FC6「TM_TRANS2」は、工具の位置変更、状態変更、マルチツールに使用されます。

FC6 ブロックは、FC8 ブロックと同じ機能に加えて、**マルチツール**機能を備えています。

マルチツール機能は、FC6 概要にのみ含まれています。

FC8 の機能の説明は、「FC8:TM_TRANS - 工具管理機能用伝送ブロック (ページ 1275)」にあります。

ファンクションの宣言

```
FUNCTION FC6:VOID
VAR_INPUT
   Start:           BOOL;
   TaskIdent:       BYTE;
   TaskIdentNo:     BYTE;
   NewToolMag:      INT;
```

```
   NewToolLoc:       INT;
   OldToolMag:       INT;
   OldToolLoc:       INT;
   Status:           INT;
   MtoolPlaceNum:    INT;
END_VAR
VAR_OUTPUT
   Ready:            BOOL;
   Error:            INT;
END_VAR
BEGIN
END_FUNCTION
```

仮パラメータの説明

信号	種別	タイプ	数値の範囲	説明
Start:	I	BOOL	0 (FALSE)、1 (TRUE)	ブロック概要 FC8 を参照してください
TaskIdent:	I	BYTE		ブロック概要 FC8 を参照してください
TaskIdentNo:	I	BYTE		ブロック概要 FC8 を参照してください
NewToolMag:	I	INT		ブロック概要 FC8 を参照してください
NewToolLoc:	I	INT		ブロック概要 FC8 を参照してください
OldToolMag:	I	INT		ブロック概要 FC8 を参照してください
OldToolLoc:	I	INT		ブロック概要 FC8 を参照してください
状態:	I	INT		ブロック概要 FC8 を参照してください
MtoolPlaceNum:	**I**	**INT**		**マルチツールのロケーション番号**
Ready:	O	BOOL	0 (FALSE)、1 (TRUE)	ブロック概要 FC8 を参照してください
エラー:	O	INT		ブロック概要 FC8 を参照してください

14.17.15 FC7:TM_REV - リボルバの工具交換伝送ブロック

機能

リボルバの交換後に、ユーザーはブロック FC7「TM_REV」を呼び出します。このために、DB73 のインターフェース番号に対応するリボルバ番号をパラメータ「ChgdRevNo」で指定してください。このブロックを呼び出すと、パラメータ「Ready」= TRUE が返された後で、FC7 のデータブロック DB73.DBW0 にある関連「インターフェース有効」ビットがリセットされます。

ジョブが正しく実行された場合

ジョブが正しく実行された場合は、「Ready」== 1 になります。その後、ユーザーは、パラメータ「Start」= 0 を設定してください。または、それ以上、FC7 を呼び出さないでください。

ジョブが異常とともに実行された場合

ジョブが異常とともに実行された場合は、パラメータ「Ready」== 0 およびパラメータ「Error」== 1 になります。次の PLC 周期でジョブを繰り返してください。パラメータ「Start」には、後続のジョブ「Start」の立ち上がりエッジが必要でないため、「Start」= 1 のままです。これは、ジョブがまだ完了してないためです。後述の「呼び出し例」および「タイミングチャート」を参照してください。

一般条件

- ブロック FC7 をパラメータ「Start」= 1 で起動できるのは、この伝送に関連するインターフェース(DB73.DBW0)の起動信号が工具管理機能によって提供されている場合のみです。
- チャネルリセットなど、伝送のキャンセルは実行できません。
- パラメータ「Ready」== 1 または「Error」== 1 となるまではパラメータ「Start」= 1

参照先

- 工具管理機能についての詳細は、『機能マニュアル 、ツールマネージャ』を参照してください。
- 工具管理用 PI サービスについては、以下を参照してください。
  - FB4:PI_SERV - PI サービス要求 (ページ 1193)
  - FC8:TM_TRANS - 工具管理機能用伝送ブロック (ページ 1275)
  - FC22:TM_DIR - 工具管理用の方向選択 (ページ 1332)

手動によるリボルバ切り替え

リボルバを手動操作で回転する場合、工具交換とオフセット選択のどちらもこの操作には関係しません。最初の手順は、工具を工具ホルダから取り外し、リボルバ内のそのロケーションに戻すことです。非同期転送は、FC8 により実行する必要があります(代替:FC6:関連するパラメータ設定は、以下のとおりです。

```
TaskIdent = 4
TaskIdentNo = チャネル番号
NewToolMag = リボルバのマガジン番号
NewToolLoc = 元の工具ロケーション
OldToolMag = バッファロケーション(主軸)のマガジン番号 = 9998
OldToolLoc = 主軸のバッファロケーション番号
Status = 1
```

これでリボルバが工具を配置する任意の位置に回転されたら、この工具を起動してください。これを簡単におこなうには、パートプログラムで新しい T プログラミングを使用します。ただし、これをおこなう場合は、PLC ユーザープログラムから切り替わるリボルバの最後などで、ASUB を起動してください。現在のリボルバ位置を ASUB に伝送してください。このようにして、このロケーションにある工具が ASUB で特定されて選択されます(ツールボックスの JobShop 例を参照してください)。

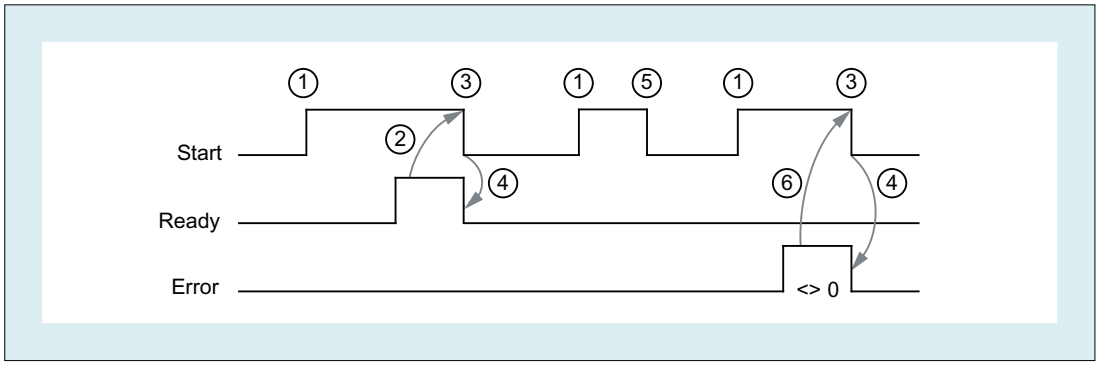
ファンクションの宣言

```
FUNCTION FC7:       VOID
//名称: TM_REV
VAR_INPUT
    Start:          BOOL;
    ChgdRevNo:      BYTE;
END_VAR
VAR_OUTPUT
    準備完了        BOOL;
    Error :         INT;
END_VAR
BEGIN
END_FUNCTION
```

仮パラメータの説明

表記	種別	タイプ	値の範囲	説明
Start:	I	BOOL	0 (FALSE)、1 (TRUE)	1 = 伝送開始
ChgdRevNo:	I	BYTE	1, 2, 3, ...	リボルバインタフェースの番号
Ready:	O	BOOL	0 (FALSE)、1 (TRUE)	1 = 伝送完了
エラー:	O	INT	0, 1 , 2, 3	異常アンサーバック 0:異常が発生していない 1:リボルバが存在しない 2:パラメータ「ChgdRevNo」のリボルバ番号が不正 3:不正なジョブ(選択されたリボルバの「インタフェース有効」信号 =「FALSE」)

タイミングチャート



① User:要求の設定、Start = 0→1

② FB4:PI サービスが正常に終了、Ready = 1
User:要求のリセット、IF Ready == 1 THEN Start = 0

③ User:IF Ready == 1 THEN、要求のリセット:1 → 0

④ FB4:ジョブ確認のリセット、Ready = 0

⑤ User:IF Ready == 0 AND Error == 0 THEN、要求のリセット、Start = 1 → 0 **許容されません**

⑥ FB4:PI サービスが異常とともに終了、Error = 1
User:要求のリセット、IF Ready == 1 OR Error == 1 THEN Start = 0、さらにエラー処理の可能性

呼び出し例

```
CALL     //工具管理:リボルバ用ブロックの伝送
FC7 (
    Start :=          m 20.5,          // Start :=「1」 => 伝送開始
    ChgdRevNo :=      DB61.DBB1:
    Ready :=          m 20.6,
    Error :=          DB61.DBW12
};
u       M 20.6;                        //準備完了のポーリング
R       M 20.5;                        //リセット開始
spb     m001;                          //すべて OK の場合はジャンプ
L       db61.DBW 12);                  //異常情報
OW      W#16#0;                        //異常解析
```

```
SPN;ＳＰ error :                    // <> 0 の場合はトラブルシューティングにジャンプ
N
m001:                               //別のプログラムの起動

error :

R       M 20.5;                     //エラーが発生した場合はリセット開始
```

14.17.16 FC8:TM_TRANS - 工具管理機能用伝送ブロック

機能

工具の位置または伝送処理の状態が変わる際に、このブロック FC TM-TRANS を呼び出します。工具管理機能のインターフェースにおいて、パラメータ「TaskIdent」を使用してブロック FC8 の以下の伝送ジョブを指定します。

- ロード/アンロード位置
- 主軸交換位置
- 伝送識別子としてのリボルバ変更位置
- 非同期伝送
- ロケーション確保を伴う非同期伝送

インタフェース番号は、パラメータ「TaskIdentNo」で示されます。

ロードポイント 5 の例:

パラメータ「TaskIdent」:= 1 と「TaskIdentNo」:= 5

さらに、**現在の**工具位置と状態データ(後述のパラメータ「Status」の状態一覧を参照してください)も、この伝送ファンクションで伝送されます。

注記

FC8 は、古い工具の現在位置を NC に通知します。
NC は、位置が変更するまで古い工具と新しい工具がどこに配置されているかを認識します。

いわゆる「古い工具」がない場合の伝送では(たとえば、ロード時)、値 0 がパラメータ「OldToolMag」と「OldToolLoc」に割り当てられます。

この伝送に対応するインターフェース(ワード 0 の DB71、DB72、DB73)の起動信号が工具管理機能によって提供されている場合、ブロック FC TM_TRANS はパラメータ「Start」=「TRUE」の設定でのみ開始できます。

このジョブが正しく実行されると、出力パラメータ「Ready」の値が TRUE になります。

この場合、パラメータ「Start」を FALSE に設定するか、またはこのブロックを二度と呼び出さしないようにしてください。

パラメータ「Ready」が FALSE の場合、パラメータ「Error」の異常コードを解析してください(FC8 の呼び出し例とタイミングチャートを参照してください)。

異常コード= 0 の場合は、このジョブを次の PLC 周期で繰り返してください(例:「Start」の設定は「TRUE」のまま)。これは、伝送処理がまだ完了していないことを意味しています。

100 未満の値を「Status」パラメータに割り当てると、ワード 0 のデータブロック DB71、DB72、または DB73 にある関連インターフェースが解除されます(処理完了)。インターフェースの適切なビット FC8 によって 0 に設定されます。

パラメータ「Start」には、以降のジョブのための信号変化は不要です。つまり、「Ready = TRUE」を受信すると直ちに新しいパラメータを「Start = TRUE」で割り当てることができます。

非同期伝送

工具位置の変更が PLC から工具管理機能に自動的に通知されるように(たとえば、命令がアクティブである時の電源故障、または PLC による独立した位置変更)、FC8 が「TaskIdent」= 4 または 5 で呼び出されます。この呼び出しでは、工具管理機能によるインターフェースの起動は不要です。

パラメータ「TaskIdent」= 5 の場合は、工具管理機能によって、位置変更に加えてロケーション確保がおこなわれます。ただし、ロケーションが確保されるのは、工具が実際のマガジンからバッファロケーションに移された場合のみです。

関連の NC チャネルをパラメータ「TaskIdentNo」で指定してください。

前の工具のロケーションは、パラメータ「OldToolMag」および「OldToolLoc」で指定します。工具の現在のロケーションは、パラメータ「NewToolMag」および「NewToolLoc」で指定します。「Status」= 1 を指定してください。

「Status」= 5 の場合は、指定された工具が、ロケーション「OldToolMag」と「OldToolLoc」に残ります。このロケーションはバッファでなければなりません(たとえば、主軸)。実際のマガジンとロケーションは、パラメータ「NewToolMag」と「NewToolLoc」で指定してください。そのロケーションはバッファ位置にします。特定のマガジンロケーションの位置を工具管理機能に通知する必要がある場合は、必ずこの手順に従ってください。この手順は、検索方法の調整に使用します。

必要条件

- チャネルリセットなど、伝送のキャンセルは実行できません。
- パラメータ「Ready」== 1 または「Error」== 1 となるまではパラメータ「Start」= 1

参照先

- 工具管理機能についての詳細は、以下を参照してください。
  『機能マニュアル、工具管理機能』
- 工具管理用 PI サービス
  - FB4:PI_SERV - PI サービス要求 (ページ 1193)
  - FC7:TM_REV - リボルバの工具交換伝送ブロック (ページ 1271)
  - FC22:TM_DIR - 工具管理用の方向選択 (ページ 1332)

ファンクションの宣言

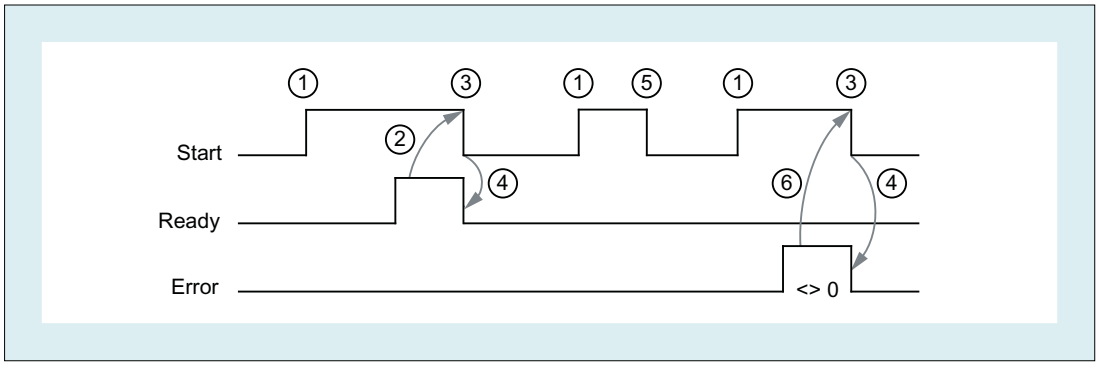
```
FUNCTION FC8:VOID
//NAME: TM_TRANS
VAR_INPUT
    Start:              BOOL;
    TaskIdent:          BYTE;
    TaskIdentNo:        BYTE;
    NewToolMag:         INT;
    NewToolLoc:         INT;
    OldToolMag:         INT;
    OldToolLoc:         INT;
    Status:             INT;
END_VAR
VAR_OUTPUT
    Ready               BOOL;
    Error :             INT;
END_VAR
BEGIN
END_FUNCTION
```

仮パラメータの説明

信号	種別	タイプ	値の範囲	意味
Start:	I	BOOL	0 (FALSE)、1 (TRUE)	1:伝送開始
TaskIdent:	I	BYTE	1, 2, 3, 4, 5	インタフェースまたはジョブ指定 1:ロード/アンロード位置 2:主軸変更位置 3:リボルバ変更位置 4:非同期伝送 5:ロケーション確保を伴う非同期伝送
TaskIdentNo:	I	BYTE	1, 2, 3 ... 10	関連インターフェース番号またはチャネル番号。
NewToolMag:	I	INT	-1, 0, 1, 2 ...	置き換えられる工具の現在のマガジン番号
				-1:工具がその位置にとどまります。 NewToolLoc = 任意の値 TaskIdent = 2 の場合のみ使用できます。
NewToolLoc:	I	INT	0、1、2 ... 最大ロケーション番号	新しい工具の現在のロケーション番号
OldToolMag:	I	INT	-1, 0 ...	置き換えられる工具の現在のマガジン番号 -1:工具がその位置にとどまります。"OldToolLoc" = <任意の値>。 「TaskIdent」= 2 の場合のみ使用できます。
OldToolLoc:	I	INT	最大ロケーション番号	置き換えられる工具の現在のロケーション番号
状態:	I	INT	1, 2, 3 ... 7, 103, 104, 105	伝送処理に関する状態情報

信号	種別	タイプ	値の範囲	意味
Ready:	O	BOOL	0 (FALSE)、1 (TRUE)	1:伝送完了
エラー:	O	INT	0 ... 65535	異常アンサーバック 0:異常が発生していない 1:「TaskIdent」不明 2:「TaskIdentNo」不明 3:許容されないジョブ、(選択されたリボルバの「インターフェース有効」信号==0) その他の値:番号は、この伝送によって生じる、NC での工具管理機能の異常メッセージに対応しています。

タイミングチャート



① 立ち上がりエッジによるファンクションの起動

② 肯定応答:工具管理機能が伝送済み

③ 確認応答受信後の機能起動のリセット

④ FC を使用した信号変更

⑤ この信号変化は受け入れられません。新しい工具位置を NC の工具管理機能に伝えなければならないため、通常はジョブを終了しなければなりません。

⑥ 否定応答:異常が発生、出力パラメータ「Error」に異常コード

ステータス

ステータス	説明
1	**工具管理ジョブが完了しています。** FC8 ブロックのパラメータ「NewToolMag」、「NewToolLoc」、「OldToolMag」、および「OldToolLoc」を、関連する工具の実際の位置に設定してください。交換準備の場合を除き、これらは通常、関連する工具管理インターフェースで指定されている工具ターゲット位置です(「正式パラメータの説明」も参照してください)。 1. **ロード/アンロード/再ロード**の場合、工具は、要求されたターゲットアドレスに到達しています。DB 71.DBX (n+0).3「ロードポイントの位置決め」のインタフェースのビットが有効である場合、この設定(Status = 1)をファンクション終了に使用することはできません。正しく終了するには、Status = 5 の設定を使用してください。 2.「**交換準備**」の場合は、新しい工具を使用できます。たとえば、工具をバッファ(グリッパ)に位置決めできます。場合によっては、新しい工具をバッファに配置した後、古い工具のターゲット(マガジン、ロケーション) が工具交換位置に移動されています。ただし、古い工具は依然として主軸に残っています。工具交換の準備は、これで完了します。この確認応答の後、「交換」命令を受け取ることができます。パラメータ「NewToolMag」、「NewToolLoc」、「OldToolMag」、および「OldToolLoc」で指定される位置は、現在の工具位置に対応しています。 3.「**交換**」の場合(主軸またはリボルバ)、インターフェースでアドレス指定された工具は、要求されたターゲットアドレスに到達しています。 工具交換処理は、これで完了します。
2	**「新しい」工具を使用可能にすることができません。** この状態は、「工具交換」命令と組み合わせる場合のみ使用できます。この状態が適用されると、提示された工具の変更を PLC でおこなうことはできません。提示された(新しい)工具が、NC の工具管理機能によって無効になります。新しい命令が、予備工具を伴う工具管理機能によって出力されます。パラメータ「NewToolMag」、「NewToolLoc」、「OldToolMag」、および「OldToolLoc」で指定される位置は、元の工具位置に対応しています。
3	**異常が発生しています。** 工具位置を変更できません。この間におこなわれた工具のマガジン位置の変更はすべて、FC8 伝送ブロックを介して状態= 105 などで、事前に通知しておいてください。通知しておいた場合にのみ、工具位置が工具管理機能によって考慮されます。

ステータス	説明
4	**「古い」工具は、パラメータ「OldToolMag」と「OldToolLoc」で指定されたマガジン位置に位置決めすることが望まれます。** この状態は、工具交換(主軸への変更)の準備と組み合わせる場合のみ使用できます。「古い」工具用に指定されたマガジンロケーションは、空いていなければなりません。この状態が NC の工具管理に伝えられると、新しい準備命令が生成され(Status_4 = 最終確認)、DB72 に出力されます。この場合、要求された「古い」工具のマガジン位置が考慮されています。 新しい工具検索は明示的には実行されず、元の準備命令の「NewToolMag」と「NewToolLoc」の位置が使用されます。ただし、考慮されるのは、この位置が空いている場合だけです。パラメータ「NewToolMag」と「NewToolLoc」は無視されます。
5	**処理が完了しています。** 「新しい」工具は、パラメータ「NewToolMag」と「NewToolLoc」で指定された位置にあります。この場合、指定された工具は実際にはこの位置にはなく、まだ同じマガジンロケーションにあります。ただし、このマガジンロケーションが、これらのパラメータで設定された位置(たとえば、工具交換位置)に移動しています。この状態は、リボルバ、チェーンタイプマガジン、およびディスクマガジンについてのみ使用できます。この状態では、工具管理機能でマガジンの現在位置を調整することができ、以降の命令での検索方法を改善できます。この状態は、ロード/アンロード/再ロード処理および工具交換の準備と組み合わせる場合のみ使用できます。パラメータ「OldToolMag」と「OldToolLoc」は、バッファのデータを設定しなければなりません。 ● **ロード、再ロード**: ロードまたは再ロードでは、工具用のロケーションが NC ですでに確保されています。機械オペレータは、工具をターゲットロケーションに挿入しなければなりません。重要:コントロールシステムの電源を再投入すると、ロケーションの確保が取り消されます。 ● **工具交換準備**: まだ実行されていない工具の動きは、工具が変更されるまでおこなわれません。 ● **ロードポイントへの位置決め:** DB 71.DBX (n+0).3「ロードポイントの位置決め」のインターフェースのビットが有効である場合は、状態 5 (状態 1 ではなく)のみをファンクション終了に使用します。

ステータス	説明
6	**工具管理ジョブが完了しています。** Status = 6 は、Status = 1 と同じ機能ですが、追加でソースロケーションの確保が実行されます。この状態は、再ロード時にのみ使用できます。ターゲットロケーションがバッファマガジンにある場合、命令が終了し、工具のソースロケーションが確保されます。
7	**「工具準備」命令の繰り返しを開始します。** この状態は、「工具交換」命令と組み合わせる場合のみ使用できます。この状態は、「新しい」工具の位置が変更された(たとえば、「新しい」工具の非同期命令を介して)場合に使用します。FC8 で「Ready = 1」と設定された後、同じ工具で「交換準備」命令が自動的に繰り返されます。自動繰り返しのために、新しい工具の検索が実行されます。パラメータ「NewToolMag」、「NewToolLoc」、「OldToolMag」、および「OldToolLoc」で指定される位置は、元の工具位置に対応しています。
103	**「新しい」工具を挿入できます。** この状態は、PLC で新しい工具を拒否できる場合に工具交換準備でのみ使用できます(たとえば、MD20310 $MC_TOOL_MANAGEMENT_MASK、ビット 4=1 の時、PLC から変更パラメータを再度要求する場合など)。工具位置は変わっていません。この状態は、NC において不要な停止を伴わないで処理を続行しなければならない場合に必要です。
104	**「新しい」工具は、パラメータ「NewToolMag」と「NewToolLoc」で指定された位置にあります。** この状態は、工具がまだ同じマガジンロケーションにある場合にのみ使用できます。「古い」工具は、パラメータ「OldToolMag」と「OldToolLoc」で指定された位置(バッファ)にあります。ただし、この場合、新しい工具は実際にはこの位置にはなく、まだ同じマガジンロケーションにあります。ただし、このマガジンロケーションが、これらのパラメータで設定された位置(たとえば、工具交換位置)に移動しています。この状態は、「工具交換準備」段階でリボルバ、チェーンタイプマガジン、およびディスクマガジンと組み合わせる場合のみ使用できます。この状態では、工具管理機能でマガジンの現在位置を調整することができ、以降の命令での検索方法を改善できます。
105	**関係するすべての工具が、指定したバッファロケーションに到達しています。** 処理が完了していない場合の標準的なケース。 工具が、指定された工具位置にあります(パラメータ「NewToolMag」、「NewToolLoc」、「OldToolMag」、および「OldToolLoc」)。

状態定義

応答状態に関する一般的規則では、状態情報が 1 ～ 7 の場合に命令が終了します。FC8 が状態情報を受け取ると、FC8 で指定されたインターフェースの「インターフェース有効ビット」が「0」にリセットされ(インターフェースリスト DB 71 ～ DB 73 も参照)、処理が完了します。状態情報が 103 ～ 105 の場合は、動作が異なります。FC8 が状態情報を受け取ったとき、このインターフェースの「インターフェース有効ビット」が「1」のままです。PLC のユーザープログラムによって、さらなる処理が必要です(たとえば、マガジン位置決めの続行)。この状態情報は通常、処理がまだ進行中に、一方または両方の工具の位置変更を伝送するために使用します。

呼び出し例

```
CALL FC8(           //工具管理機能伝送ブロック
   Start :          m 20.5,         // Start :「1」  => 伝送開始
   TaskIdent :      DB61.DBB0,
   TaskIdentNo :    DB61.DBB1:
   NewToolMag :     DB61.DBW2,      //新しい工具の現在位置
   NewToolLoc :     DB61.DBW4,
   OldToolMag :     DB61.DBW6,      //古い工具の現在位置
   OldToolLoc :     DB61.DBW8,
   Status :         DB61.DBW10,     //状態
   Ready :          m 20.6,
   Error :          DB61.DBW12,
u m 20.6;                           //準備完了のポーリング
r m 20.5;                           //リセット開始
spb m001;                           //すべて OK の場合はジャンプ
DB61.dbw12,                         //異常情報
ow w#16#0;                          //異常解析
JC error;                           // トラブルシューティングへジャンプ

m001:                               //通常の分岐

error :                             //トラブルシューティング
r m 20.5:                           //リセット開始
```

14.17.17 FC9:ASUB - 非同期サブプログラムの起動

機能

FC9「ASUB」を使用して、NC で任意の機能を起動することができます。PLC から ASUB を起動する前に、NC プログラムまたは FB4 (PI サービス ASUB)によって、ASUB の選択とパラメータ設定をおこなってください。この場合は、チャネルと割り込み番号が FC9 のパラメータと一致していなければなりません。

この方法で準備されていれば、PLC からいつでも起動できます。該当するチャネル上で実行されている NC プログラムは、非同期サブプログラムによって中断されます。

同じチャネルで一度に起動できる ASUB は 1 つのみです。複数の ASUB を 1 つの PLC 周期で起動した場合は、この順序で ASUB が NC で起動されます。

ASUB が終了する(「Done」== 1)か、異常が発生した(「Error」== 1)場合、ユーザーはパラメータ「Start」= 0 を設定してください。

処理ジョブの場合は、各 FC9 にグローバルユーザー領域のパラメータ「Ref」がそれぞれ必要です。これは内部でのみ使用するパラメータであるため、変更しないでください。パラメータ「Ref」は、最初の OB1 周期で値 0 に初期化されます。このため、すべての FC を**必ず**呼び出してください。別の方法として、起動時にパラメータ「Ref」を値 0 で初期化することができます。このオプションでは、条件付き呼び出しが可能になります。条件付き呼び出しには、FC9 の起動中に、立ち下がりエッジ変更がパラメータ「Done」(1→0)で発生するまでパラメータ「Start」= 1 が必要です。

一般条件

- ファンクションブロック FB4 は、ブロック FC9 が起動する前に終了させてください。
- ブロック FC9 は、DB10、DBX56.1 == 1 (非常停止)の場合は起動できません。
- ブロック FC9 は、ASUB が起動されるチャネルでチャネルリセットが有効な場合は起動できません。

ファンクションの宣言

```
FUNCTION FC9:VOID
//NAME: ASUB
VAR_INPUT
    Start:       BOOL;
    ChanNo:      INT;
    IntNo:       INT;
```

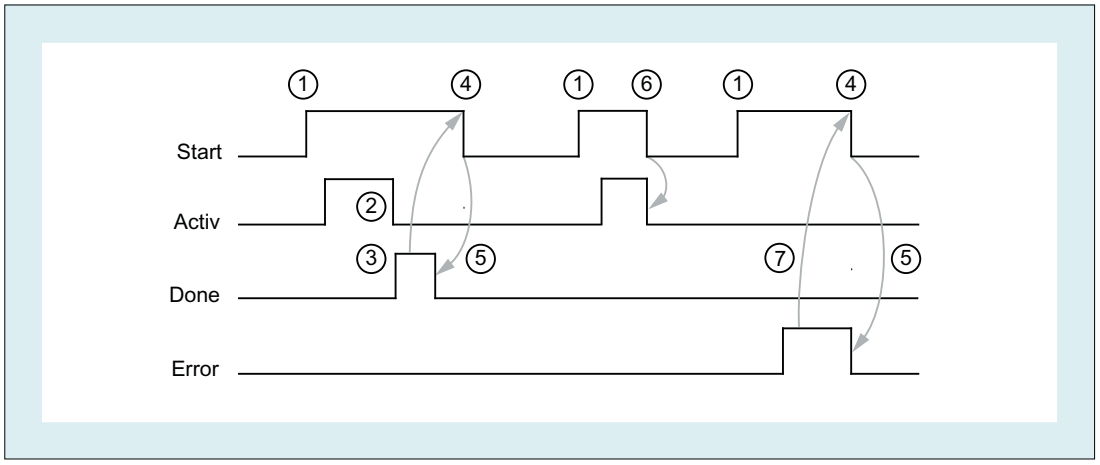
```
END_VAR
VAR_OUTPUT
    Active:      BOOL;
    Done :       BOOL;
    Error :      BOOL;
    StartErr:    BOOL;
END_VAR
VAR_IN_OUT
    Ref:         WORD;
END_VAR
```

仮パラメータの説明

次の表に、ASUB ファンクションのすべての仮パラメータを示します。

信号	種別	タイプ	値の範囲	説明
Start:	I	BOOL	0 (FALSE)、1 (TRUE)	信号の立ち上がりエッジを使用してジョブが起動します
ChanNo:	I	INT	1, 2, 3 ... 10	チャネル番号
IntNo:	I	INT	1, 2, 3 ... 8	割り込み番号
有効:	O	BOOL	0 (FALSE)、1 (TRUE)	1:有効
Done :	O	BOOL	0 (FALSE)、1 (TRUE)	1:ASUB 完了
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	1:割り込みオフ
StartErr:	O	BOOL	0 (FALSE)、1 (TRUE)	1:未割り当てまたは削除された割り込み番号
Ref:	I/O	WORD	グローバル変数(MW, DBW,..)	FC9 毎に 1 ワード(内部使用)

タイミングチャート



(1) ファンクションの起動

(2) ASUB 有効

(3) 肯定応答:ASUB 完了

(4) 確認応答受信後の機能起動のリセット

(5) FC を使用した信号変更

(6) 許容されません。確認応答を受信する前に機能の起動がリセットされている場合は、出力信号が更新されると必ず、起動された機能の処理手順が影響を受けます。

(7) 否定応答:異常が発生しました

呼び出し例

```
CALL FC9(                          //非同期サブプログラムの起動
                                   //チャネル 1 割り込み番号 1 において
     Start :=      I 45.7,
     ChanNo :=     1,
     IntNo :=      1,
     Active :=     M 204.0,
     Done :=       M204.1,
     Error :=      M 204.4,
     StartErr :=   M 204.5,
     Ref :=        MW 200);
```

14.17.18 FC10:AL_MSG - 異常メッセージおよび操作メッセージ

機能

ブロック FC10「AL_MSG」の場合は、DB2 に入力された信号が評価され、着信または発信の異常メッセージおよび操作メッセージがユーザーインターフェースに表示されます。

異常メッセージの場合でも操作メッセージの場合でも、入力信号(立ち上がりエッジ)は即座に表示されます。

出力信号(立ち下がりエッジ)は、操作メッセージのついてのみ、直ちに削除されます。異常メッセージが表示された場合は、未処置でなくなったメッセージのみがパラメータ「Quit」とともに削除されます。つまり、ユーザーに確認応答されるまでは、信号が未処置ではなくなっている場合でも、異常がユーザーインターフェースに表示されたままです。

パラメータ「ToUserIF」を使用すると、送り、補間、および NC スタートの各無効信号および送り停止信号用のグループ信号を、存在する軸/主軸/チャネルインタフェースに伝送することができます。これらのグループ信号は、アラーム応答には関係なく、DB2 の状態情報から直接ユーザーインターフェースに伝送されます。

1. パラメータ「ToUserIF」= 0 の場合は、ユーザーインターフェースに信号が伝送されません。この場合、これらの信号が確実にインタフェース内で影響を受けるように、ユーザー PLC プログラムで対処しなければなりません。FB1 パラメータ「ExtendChanAxMsg」を使用し、NCK マシンデータを設定して、使用可能なメッセージ範囲が制限されないようにします。
2. パラメータ「ToUserIF」= 1 の場合は、上記の信号がすべて、それぞれの場合のグループ信号としてユーザーインターフェースに送られます。そのため、ユーザー PLC プログラムは、メッセージまたはアラーム出力とともに DB2 を介してのみ、これらの信号に影響を及ぼすことができます。対応する情報はユーザーインタフェースで上書きされます。

上記の 2.で説明した応答の代わりに、FC10 の呼び出し後に無効信号状態または停止信号状態でインターフェース信号に影響を及ぼして、メッセージを出力せずに無効信号および保持信号に影響を及ぼすこともできます。

次のプログラムは、この方法を示しています。

```
CALL FC10(
   ToUserIF :=     TRUE,
   Ackn :=         I 6.1);

u m 50.0;                          //チャネル 1 の送り無効
```

```
to DB 21;
s dbx 6.0;                          //遮断条件の設定
                                    // FC AL_MSG でおこなわれたリセット
                                    //(M 50.0 が信号「0」を出力する場合)。
```

FB1 パラメータ「ExtendAlMsg」

FB1 パラメータ「ExtendAlMsg」= TRUE の場合は、新しい DB2 構造が有効になります(「PLC/HMI インタフェース (ページ 1073)」を参照してください)。起動するために、ロック信号と停止信号のビット領域が、10 個のチャネル、31 本の軸、および最大 64 のユーザー領域(ユーザー領域の数は FB1 パラメータ「MsgUser」に入力します)で使用可能です。関連する機能は、DB2 に信号を設定/リセットするだけで自動的に入手できます。

データブロック DB2 の異常メッセージと操作メッセージは、ユーザー別に提供してください。

FB1 パラメータ「ExtendChanAxMsg」

このパラメータの起動により、チャネル番号または軸番号とは無関係に、アラームとメッセージが取得されます。ユーザーは、すべての DB2 領域を使用できます。グループ信号は、ユーザーインターフェースに伝送できません。パラメータは、FC10 パラメータ「ToUserIF」が無効になっている場合にのみ評価されます。

HMI での表示

DB2 では、メッセージが HMI にも表示されるように、複数の OB1 周期にわたって「1」信号が出力されなければなりません。

同時に保留できるアラームおよびメッセージの数には上限があります。この上限は、PLC CPU によって決まります。PLC 317-2DP では、同時に保留できるメッセージの上限は 60 です。

参照先:
『NC 変数およびインターフェース信号用パラメータマニュアル』、「PLC ユーザーインターフェース」の章の「PLC アラーム/メッセージ」

ファンクションの宣言

```
FUNCTION FC10:      VOID
    //NAME:         AL_MSG
VAR_INPUT
    ToUserIF :      BOOL;
```

```
    [Quit]          BOOL;
END_VAR
END_FUNCTION
```

仮パラメータの説明

信号	タイプ	タイプ	値の範囲	意味
ToUserIF :	I	BOOL	0 (FALSE)、1 (TRUE)	1:各サイクルでのユーザーインターフェースへの信号の伝送
[Quit]	I	BOOL	0 (FALSE)、1 (TRUE)	1:異常メッセージの確認応答

呼び出し例

```
CALL FC10(                  //異常メッセージおよび操作メッセージ
   ToUserIF :=   TRUE,      // DB2 からの信号をインターフェースに伝送
   Ackn :=       E6.1       // I6.1 経由の異常メッセージの確認応答
);
```

14.17.19 FC12:AUXFU - 補助機能のユーザー用呼び出しインターフェース

機能

入力パラメータで伝送されたチャネルに新しい補助機能が含まれる場合、ブロック FC12「AUXFU」は、基本プログラム内でイベント起動型で呼び出されます。PLC ユーザーは、チャネル DB の周期的ポーリングを避けて補助機能を処理するように、プログラム命令で FC AUXFU を拡張できます。このメカニズムにより、ジョブ起動型で補助機能を処理することができます。FC AUXFU は、基本プログラム内でコンパイルされた空のブロックとして提供されます。この場合、基本プログラムはパラメータ「Chan」にチャネル番号を提供します。PLC ユーザーは、どのチャネルで新しい補助機能が使用可能であるかがわかります。そのチャネルにおいて、新しい補助機能を補助機能変更信号によって決定することができます。

ファンクションの宣言

```
FUNCTION FC12:VOID              //補助機能のイベント制御
VAR_INPUT
    Chan:     BYTE;
END_VAR
BEGIN
    BE;
END_FUNCTION
```

仮パラメータの説明

信号	タイプ	タイプ	値の範囲	説明
Chan:	I	BYTE	0, 1, 2 ... 9	チャネルのインデックス=チャネル番号-1

例

```
FUNCTION FC12:VOID          //補助機能のイベント制御
VAR_INPUT
    Chan:       BYTE;       //パラメータは基本プログラムによって提供
END_VAR
VAR_TEMP
    ChanDB:     INT;
END_VAR

BEGIN
L Chan;                     //チャネルインデックス
+ 21;                       //チャネル DB オフセット
T ChanDB;                   //チャネル DB 番号を保存
TO DB[ChanDB];              //チャネル DB を間接的にオープン
//ここで補助機能の変更信号をスキャン、など
    BE;
END_FUNCTION
```

14.17.20 FC13:BHGDisp - ハンドヘルドユニット用表示制御

機能

ブロック FC13「BHGDisp」は、ハンドヘルドユニット(HHU または HT 2)の表示制御をおこないます。画面に表示される情報は、文字列変数に保存してください。文字列へのポインタは、パラメータ「ChrArray」で指定します。これをおこなうには、この文字列用のデータブロックの作成時に、32 文字(HHU)または 64 文字(HT 2)で固定されたテキストの割り当てが必要です。

16 文字が、ジョブ毎に HHU に送られます。「ChrArray」での各行に対する文字の割り当ては、一義的です。行 1 について文字列データ ChrArray の文字 1 ～ 16 が伝送され、行 2 について文字 17 ～ 32 が伝送されます。また、HT 2 の場合は、文字 33 ～ 48 が行 3 に、文字 49 ～ 64 が行 4 に表示されます。1 つのジョブは、複数の OB1 周期にわたります。

ディスプレイ

ブロック FC13 は、ハンドヘルドユニットの操作に必要な最小長の「ChrArray」が存在するかどうかをチェックします。文字列変数に存在する文字数が、表示されるべき文字数より少ない場合は、行にスペースが入力されます。1 つ以上の PLC 周期で複数の変数が、表示出力されずに文字列に入力される場合は、表示出力をパラメータ「Row」= 0 でマスクすることができます。行への文字の伝送は、複数の OB1 周期にわたります。複数の行が「同時に」更新される(パラメータ「Row」> 1)場合は、1 行あたり 16 文字が、連続して行で更新されます。

変数部分

文字列内の変数成分は、オプションの再割り当て機能を使用してパラメータ「Convert」= 1 で挿入できます。表示される変数は、パラメータ「Addr」を介して参照します。変数のフォーマットは、パラメータ「DataType」で記述します。変数のバイトの数は、フォーマット説明とリンクしています。文字列内で右揃えされたアドレスを、パラメータ「StringAddr」によって指定します。書き込まれた文字の数は、パラメータ表にあります。

高分解能表示

たとえば、より高い分解能で軸の値を表示する場合は、以下の条件を守る必要があります。

- 変数は、以前と同様に、FB2 または FB5 で読み取ります。64 ビット浮動小数点数として出力する条件としての anypointer BYTE 8 の代わりに、REAL 2 を使用します (例:(P#M100.0 REAL 2 など)。
- HHU/HT 2 で 64 ビット浮動小数点数を指定するときに、指定された固定フォーマットではなく、小数点の前後に自由に配置できる 14 か所までの出力フォーマットを選択できます。

HHU 出力信号

バイト 1 は HHU の出力信号によって使用され、文字の指定はブロックによって使用されます。これらは、PLC ユーザープログラムによって書き込むことはできません。

関連 FB1 パラメータ

ハンドヘルドユニット HHU

OB100 では、ハンドヘルドユニットの入力データと出力データに FB1 パラメータを設定してください。

- パラメータ「BHGIn」は、ハンドヘルドユニットからの PLC の入力データ(PLC が受信したデータ)に対応しています。
- パラメータ「BHGOut」は、ハンドヘルドユニットへの PLC の出力データ(PLC によって伝送されたデータ)に対応しています。

この 2 つのポインタは、関連データ領域の開始点に設定してください(SDB 210 においても MPI リンクで同様にパラメータ設定されます)。

FB1 パラメータ「HHU」= 2 を、HHU の操作用に設定してください。

HT 2 ハンドヘルドターミナル

HT 2 を使用する場合は、FB1 パラメータ「HHU」= 5 を設定してください。上記の「ハンドヘルドユニット HHU」に従って、入力データと出力データのパラメータを設定してください。

HT2 の接続モジュールの DIP 固定スイッチ(ロータリスイッチ)の S2 で設定されている値を、パラメータ「BHGRecGDNo」および「BHGRecGBZNo」に割り当ててください。

ファンクションの宣言

```
DATA_BLOCK "strdat"
   STRUCT
     disp:      STRING [32]:= 'character_line1 character_line2';
   END_STRUCT;
BEGIN
END_DATA_BLOCK

FUNCTION FC13:VOID
   VAR INPUT
     Row :        BYTE;
     ChrArray :   STRING ;
     Convert :    BOOL;
     Addr:        POINTER;
     DataType :   BYTE;
     StringAddr : INT;
     Digits :     BYTE;
END VAR
VAR OUTPUT
     Error :      BOOL;
END VAR
```

仮パラメータの説明

信号	種別	タイプ	数値の範囲	説明
Row :	I	BYTE	0, 1, 2, ... 8 B#16#F	行「binary」の評価の表示 0:表示出力なし 1:行 1 2:行 2 3:行 1 と行 2 を変更 4:行 3 5:行 1 と行 3 を変更 8:行 4 4 行すべての B#16#F 自動変更
ChrArray :	I	STRING	「DBName」.<VarName>	文字列へのポインタとしての内容の表示 string[32]:HT 2 以外 string[64]:HT 2
Convert :	I	BOOL	0 (FALSE)、1 (TRUE)	数値変換の起動
Addr:	I	Pointer		変換される変数へのポインタ

信号	種別	タイプ	数値の範囲	説明
DataType :	I	BYTE	1, 2, 3 ... 8, B#16#13, B#16#30	変数のデータタイプ 1:BOOL、1 文字 2:BYTE、3 文字 3:CHAR、1 文字 4:WORD、5 文字 5:INT、6 文字 6:DWORD、7 文字 7:DINT、8 文字 8:REAL、9 文字 (7 桁と 1 個の記号と小数点。小数点の後の桁については、「桁パラメータ」を参照してください) B#16#13:文字列、最大 32/64 文字、「Addr」は STRING へのポインタでなければなりません。 B#16#30:REAL64、 (12 文字:10 桁と 1 個の記号と小数点。小数点の後の桁については、「桁パラメータ」を参照してください)
StringAddr :	I	INT	1 ... 32 / 64	「ChrArray」の変数内にある右揃えアドレス
Digits :	I	BYTE	1, 2, 3 ... 9	小数点の後の桁数: 1 ... 4:DataType REAL 1 ... 9:DataType REAL64
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	異常 1:異常が発生しました

値の範囲

データタイプの値の範囲	
データタイプ	記述可能な数値範囲
BOOL	0, 1
BYTE	0 ... 255
WORD	0 ... 65535
INT	- 32768 ... 32767
DWORD	0 ... 9999999
DINT	-9999999 ... 9999999
REAL (Digits := 1)	-999999.9 ... 999999.9
REAL (Digits := 2)	-99999.99 ... 99999.99
REAL (Digits := 3)	-9999.999 ... 9999.999
...	...
REAL (Digits := 9)	-0.9999999 ... 0.9999999

呼び出し例

// strdat という名前の DB を簡単なテーブルで宣言し、データ要素 disp を String[32] (HT 2 では
// String[64])として宣言し、完全に文字列とともに割り当てます。

```
CALL FC13(
  Row :=         MB 26,
  ChrArray :=    "strdat".disp,
  Convert :=     M 90.1,
  Addr :=        P#M 20.0,      //変換される番号
  DataType :=    MB 28,         //変数のデータタイプ
  StringAddr :=  MW 30,
  Digits :=      B#16#3,        //小数第 3 位
  Error :=       M 90.2);
```

14.17.21 FC17:YDelta - スター/デルタ切り替え

機能

ブロック FC17 を使用して、ディジタル主軸ドライブのスター/デルタ切り替えをおこないます。切り替えは、どちらの方向(スター>デルタまたはデルタ>スター)にも実行できます。

必要条件

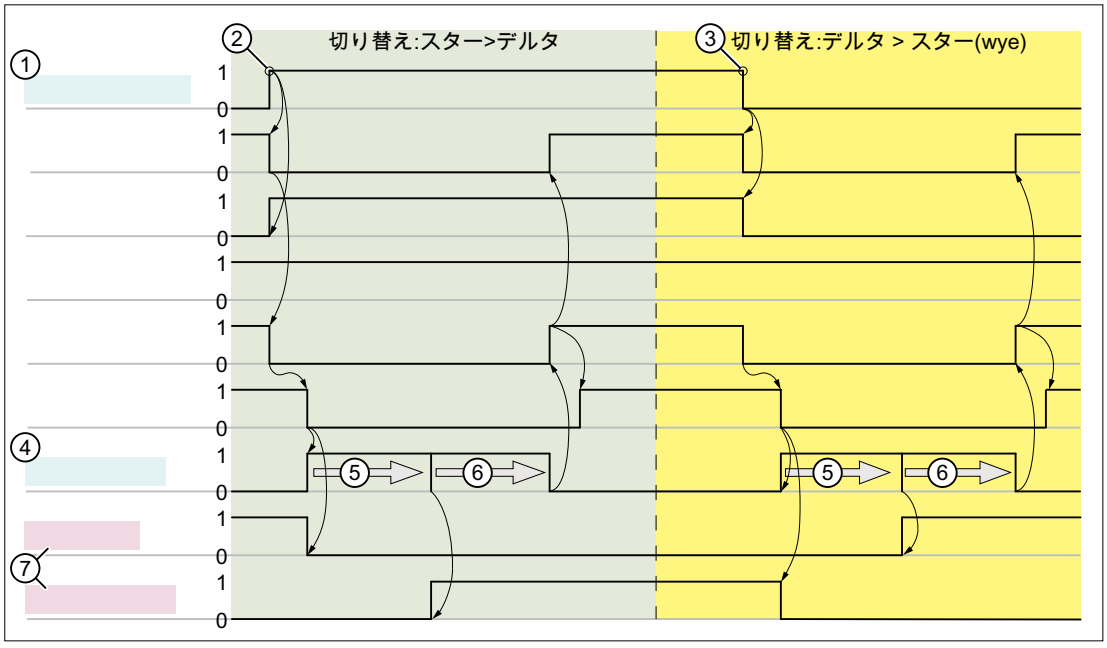
必要条件は、絶縁された 2 個のコンタクタです。コンタクタは、FC17 出力で設定された周辺出力信号、「Y」または「デルタ」を介して制御されます。

内部シーケンス

FC17 制御信号の切り替え後のスターとデルタの切り替え時の内部シーケンスの説明は、以下のとおりです。「YDelta」が表示されます。

1. DB31、... DBX21.5 = 0 (フィードバック信号「モータ選択」のリセット)
   DB31、... .DBX21.x = 1 (インターフェースパラメータ設定 DB31、... .DBX130.0 - 4 (ページ 84)に対応する要求「2 番目のモータデータ設定」の設定)
2. DB31、... DBX93.7 == 0 (フィードバック信号「パルスイネーブル中」がリセット済み)⇒
   - FC17 タイマの起動
   - FC17:「Y」= 0 (スターコンタクタの出力のリセット)
3. FC17 タイマの経過後(FC17:「TimeVal」) ⇒
   - FC17:「Y」= 1 (デルタコンタクタの出力の設定)
4. FC17 タイマの再経過後(FC17:「TimeVal」) ⇒
   - パルスが内部的に再度設定されます
   - DB31、... DBX93.7 == 1 (フィードバック信号「パルスイネーブル中」が設定済み)⇒
   - DB31、... DBX21.5 = 0 (フィードバック信号「モータ選択」が設定済み)

信号フロー



① FC17 入力:スター/デルタ切り替えの信号

② - 切り替え:スター>デルタ

③ - 切り替え:デルタ>スター

④ FC17 入力:パラメータ設定可能な切り替え時間「TimeVal」

⑤ - 出力信号の制御までの待機時間:「Y」または「デルタ」

⑥ - パルスイネーブルまでの待機時間

⑦ FC17 出力:コンタクタ制御の信号

参照先

モータの速度調整の詳しい説明については、以下を参照してください。

- 『機能マニュアル、基本機能』、「主軸(S1)」、「設定可能なギヤ調整」の章
- 『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)」

注記

ドライブパラメータ

スター/デルタ切り替えの場合は、以下のドライブパラメータを考慮してください。

- p833 (データセット切り替え設定)
  - ビット 0 = 1 (**アプリケーション**を介したコンタクタ切り替え)
  - ビット 1 = 0 (**ドライブ**によるパルスキャンセル)
- p826 (モータ切り替えモータ番号)
- p827 (モータ切り替えステータスワードビット番号)

一般条件

- ブロック FC17 を必ず、主軸毎に個別に呼び出してください。
- ドライブパルスはスター/デルタ切り替え時に解除されます。PLC に対するフィードバック信号は以下を使用して実現されます。
  - DB31、... DBX93.7 == 0 (パルスイネーブル中)
  - DB31、... DBX61.7 == 0 (電流コントローラ有効)
  - DB31、... DBX61.6 == 0 (速度コントローラ有効)
- 主軸が、M70 や SPOS などの軸モードにある場合は、スター/デルタ切り替えが実行されません。
- 閉ループ位置制御主軸(DB31、... .DBX61.5 == 1 (位置制御有効))の場合は、主軸の移動中にスター/デルタ切り替えを実行することはできません。
  異常が発生した場合は、アラーム 25050「軌跡誤差監視」が表示され、スター/デルタ切り替えは実行されません。
- FC17 を使用してスター/デルタ切り替えが開始されると、たとえば、運転中にスター/デルタコンタクタが切り替わるまで待機することによって、ユーザーがこの切り替えを遅延させることはできません。このような遅延は、PLC ユーザープログラムでユーザーが組み込んでください。

ファンクションの宣言

```
VAR_INPUT
    YDelta:         BOOL;
    SpindleIFNo:    INT;
    TimeVal:        S5TIME ;
    TimerNo :       INT;
```

```
END_VAR
VAR_OUTPUT
    Y:              BOOL;
    Delta:          BOOL;
END_VAR
VAR_IN_OUT
    Ref:            WORD;
END_VAR
```

仮パラメータの説明

表記	種別	タイプ	数値の範囲	説明	
YDelta:	I	BOOL	0 (FALSE)、1 (TRUE)	スター/デルタ切り替えの入力信号 切り替えは、信号の変更によって開始されます。 • 0 (FALSE)スター結線 • 1 (TRUE)デルタ結線	
SpindleIFNo:	I	INT	1 ...	主軸インターフェースの数 (関連する機械軸の数)	
TimeVal:	I	S5time	0 ... 50	切り替え時間	
				設定済み	内部的に有効
				0	100
				< 50	50
TimerNo :	I	INT	10 ...	使用中のタイマの数	
Y:	O	BOOL	0 (FALSE)、1 (TRUE)	スターコンタクタ制御のための周辺出力	
Delta:	O	BOOL	0 (FALSE)、1 (TRUE)	デルタコンタクタ制御のための周辺出力	
Ref:	I/O	WORD		状態情報用インスタンス、内部使用	

呼び出し例

```
CALL FC17 (
    YDelta :=          I 45.7,          //入力 45.7 からのスター/デルタ切り替えの受け入れ
```

```
    SpindleIFNo :=     4,              //主軸インターフェース番号:4
                                       // (関連する機械軸の数)
    TimeVal :=         S5T#150ms,      //切り替え時間:150 ms
    TimerNo :=         10,             //タイマ:10
    Y :=               O 52.3,         //スターコンタクタの制御:出力 52.3
    Delta :=           O 52.4,         //デルタコンタクタの制御:出力 52.4
    Ref :=             MW50            //ビットメモリワード 50
};
```

14.17.22 FC18:SpinCtrl - 主軸制御

機能

ブロック FC18 "SpinCtrl"を使用して、PLC から主軸と軸を制御できます。このブロックは、次の機能をサポートしています。

- 主軸の位置決め
- 主軸回転
- 主軸揺動
- 割り出し軸の移動
- 位置決め軸の移動

それぞれの機能は、対応する開始信号(起動、停止)の立ち上がりエッジによって引き起こされます。この信号は、InPos =「1」または Error =「1」によって、機能が肯定応答または否定応答されるまで、論理的な「1」のままにしてください。出力パラメータは、関連トリガ信号がリセットされ、機能が完了したときに削除されます。

PLC を介して軸または主軸を制御できるようにするには、PLC についてその軸または主軸を起動してください。これをおこなうには、たとえば、「Start」または「Stop」パラメータを有効にしてブロックを呼び出します。こうすると、ブロックが NC から主軸/軸の制御を要求します。

NC は、関連する軸別インターフェース DB31、... DBX68.4 ～ 7 の主軸/軸の状態を通知します。軸/主軸を PLC 制御のもとで運転すると、有効状態の移動命令を、関連する軸別インターフェースを介して評価できます。

完了する(「InPos」が TRUE、「Start」が 0 に変更される)と、軸/主軸のチェック機能が、ブロック FC18 を使用して中立状態に切り替えられます。

または、PLC ユーザープログラムが、FC18 を呼び出す前に PLC の制御を要求することもできます。

このファンクションを続けて何回か呼び出すと、FC における切り替え処理が省かれるため、主軸/軸からのより良い応答が得られます。

PLC ユーザープログラムを通しての起動は、バイト 8 にある対応する主軸インタフェース内で実行されます。

確認を返した後、主軸は、NC プログラムによって再度プログラム指令できます。

参照先

- 『機能マニュアル、基本機能』;「主軸(S1)」
- 『機能マニュアル、上級機能』;「位置決め軸(P2)」
- 機能マニュアル 上級機能; 割り出し軸(T1)

⚠ 警告

軸/主軸の変更された応答動作

複数のブロックの呼び出し(FC18)を、PLC ユーザープログラムにおいて同じ軸/主軸用にプログラム指令した場合、そのファンクションをユーザープログラムにおいて条件付き呼び出しによってインターロックしてください。出力パラメータ「Active」または「InPos」の信号状態が 1 から 0 に変わるまで、起動したブロックの条件付き呼び出し(パラメータ起動または停止 = TRUE)を周期的に呼び出してください。

注記

呼び出しの注記

「InPos」信号、あるいは、異常の場合は「Error」信号の、1 から 0 へのエッジ変化があるまでは、FC 18 を繰り返し呼び出してください。"InPos"/"異常"信号が値 0 を提示した場合にのみ、この主軸/軸に対するさらなる"起動"または"停止"が可能です(システムは、次の"起動"または"停止"により少なくとも 1 PLC サイクル待機する必要があります)。これは、軸のインタフェースにあるデータバイト 8 での割り当てが変更されたときにも当てはまります。

中止

ファンクションの中止は、パラメータ「起動」または「停止」を使用してではなく、軸のインタフェース信号を使用してのみおこなうことができます(たとえば、残移動距離削除)。軸のインタフェースは、評価する必要がある軸の状態信号も返します(たとえば、イグザクトストップ、移動指令)。

同時性

FC18 ブロックによって複数の軸を同時に、あるいは、遅延を発生させて、移動させることができます。上限は、NC の軸の最大数でのみ制限されます。

軸無効

軸停止(DB31、... .DBX1.4 == 1)が設定されている場合、FC18 で制御される軸は移動しません。シミュレートされたフィードバック値のみが生成されます。NC による軸停止のための軸の移動時と同じ動作が生成されます。

機能

機能 1:主軸の位置決め

パラメータ	意味
Start:	0→1:機能の開始
Funct:	1:「主軸位置決め」のファンクション番号
モード:	位置決めモード 1、2、3、4 (以下の段落「仮パラメータの説明」を参照してください)
AxisNo:	機械軸の番号
Pos:	位置
FRate:	FRate ≠ 0:位置決め速度 = 0:MD35300 $MA_SPIND_POSCTRL_VELO に対応する速度
InPos:	1:「精密イグザクトストップ」で位置決め完了

パラメータ	意味
Error :	1:位置決めの異常
State :	異常コード

機能 2:主軸回転

パラメータ	意味
Start:	0→1:機能の開始
Stop:	0→1:機能の停止
Funct:	2:「主軸回転」のファンクション番号
モード:	モード= 5:回転方向 M4 モード≠ 5:回転方向 M3
AxisNo:	機械軸の番号
FRate:	主軸速度
InPos:	1:指令速度が出力されます。DB31、... DBX83.5 (主軸指令一致)も参照してください
Error :	1:位置決めの異常
State :	異常コード

機能 3:主軸揺動

パラメータ	意味
Start:	0→1:機能の開始
Stop:	0→1:機能の停止
Funct:	3:「主軸振動」のファンクション番号
AxisNo:	機械軸の番号
Pos:	ギヤ選択指令
InPos:	1:指令速度が出力されます。DB31、... DBX83.5 (主軸指令一致)も参照してください
Error :	1:位置決めの異常
State :	異常コード

パラメータ設定された揺動速度:MD35400 $MA_SAFE_DES_VELO_LIMIT

パラメータ「Pos」の機能は、MD35010 $MA_GEAR_STEP_CHANGE_ENABLE = <値>の設定に応じて異なります。

<値>	位置	機能
0	0, 1, 2, ... 5	揺動
1	0	ギヤ選択切替 M40 での揺動
	1	ギヤ選択切替 M41 での揺動
	2	ギヤ選択切替 M42 での揺動
	3	ギヤ選択切替 M43 での揺動
	4	ギヤ選択切替 M44 での揺動
	5	ギヤ選択切替 M45 での揺動

機能 4:割り出し軸の移動

注記

モジュロ変換は、パートプログラム内での POS[AX] = CIC (値)を介した割り出し位置へのアプローチに相当します。

パラメータ	意味
Start:	0→1:機能の開始
Funct:	4:「割り出し軸」のファンクション番号
モード:	位置決めモード 0、1、2、3、4
AxisNo:	機械軸の番号
Pos:	割り出し位置
FRate:	FRate ≠ 0:位置決め速度 = 0:MD32060 $MA_POS_AX_VEL に対応する速度
InPos:	1:「精密イグザクトストップ」で位置決め完了
Error :	1:位置決めの異常
State :	異常コード

機能 5、6、7、8:軸位置決め

パラメータ	意味
Start:	0→1:機能の開始
Funct:	5, 6, 7, 8:「位置決め軸」のファンクション番号

パラメータ	意味
モード:	位置決めモード 0、1、2、3、4
AxisNo:	機械軸の番号
Pos:	位置
FRate:	FRate ≠ 0:位置決め速度 = 0:MD32060 $MA_POS_AX_VELO に対応する速度
InPos:	1:「精密イグザクトストップ」で位置決め完了
Error :	1:位置決めの異常
State :	異常コード

機能 9:自動ギヤ選択切替を使用した主軸回転

パラメータ	意味
Start:	0→1:機能の開始
Stop:	0→1:機能の停止
Funct:	9:「ギヤ選択切替を使用した主軸回転」のファンクション番号
モード:	モード= 5:回転方向 M4 モード≠ 5:回転方向 M3
AxisNo:	機械軸の番号
FRate:	主軸速度
InPos:	1:指令速度を出力
Error :	1:位置決めの異常
State :	異常コード

機能 10、11:周速一定制御を使用した主軸回転

「定切削速度」機能(G96)を NC で有効にしてください。

パラメータ	意味
Start:	0→1:機能の開始
Stop:	0→1:機能の停止
Funct:	10:「定切削速度(m/min)」のファンクション番号 11:「定切削速度(フィート/min)」のファンクション番号
モード:	モード= 5:回転方向 M4 モード≠ 5:回転方向 M3

パラメータ	意味
AxisNo:	機械軸の番号
FRate:	切削速度
InPos:	1:指令速度を出力
Error :	1:位置決めの異常
State :	異常コード

ファンクションの宣言

```
FUNCTION FC18:VOID          //SpinCtrl
VAR_INPUT
    Start:                  BOOL;
    Stop:                   BOOL;
    Funct:                  BYTE;
    モード:                 BYTE;
    AxisNo:                 INT;
    Pos:                    REAL;
    FRate:                  REAL;
END_VAR
VAR_OUTPUT
    InPos:                  BOOL;
    Error :                 BOOL;
    State :                 BYTE;
END_VAR
```

仮パラメータの説明

信号	種別	タイプ	値の範囲	意味
Start:	I	BOOL	0 (FALSE)、1 (TRUE)	0→1:機能の開始
Stop:	I	BOOL	0 (FALSE)、1 (TRUE)	0→1:機能の停止

信号	種別	タイプ	値の範囲	意味
Funct:	I	BYTE	1, 2, 3, ... 11	1:主軸の位置決め 2:主軸回転 3:主軸揺動 4:軸割り出し 5:軸位置決め、メトリック 6:軸位置決め、インチ 7:ハンドルオーバライドでの PosAxis、メトリック 8:ハンドルオーバライドでの PosAxis、インチ 9:自動ギヤ選択切替を使用した主軸回転 10:周速一定制御(m/min)を使用した主軸回転 11:周速一定制御(feet/min)を使用した主軸回転
モード:	I	BYTE	0, 1, 2, ... 5	0:アブソリュート位置への位置決め 1:インクレメンタル位置決め 2:最短軌跡に沿った位置決め 3:絶対位置決め、正の移動方向 4:絶対位置決め、負の移動方向 5:M4 に応じた回転方向
AxisNo:	I	INT	1, 2, 3, ... 31	移動する軸/主軸の番号
Pos:	I	REAL	∓0.1469368 I -38 ～ ∓ 0.1701412 I +39	回転軸:度 割り出し軸:割り出し位置 直線軸:mm または inch
FRate:	I	REAL	∓0.1469368 I -38 ～ ∓ 0.1701412 I +39	回転軸と主軸: [rev/min] 直線軸: [m/min]または[ft/min]
InPos:	O	BOOL	0 (FALSE)、1 (TRUE)	1:位置決め完了または機能実行済み

信号	種別	タイプ	値の範囲	意味
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	1:異常
State :	O	BYTE	0, 1, 2, ... 255	エラーの検出

異常識別子

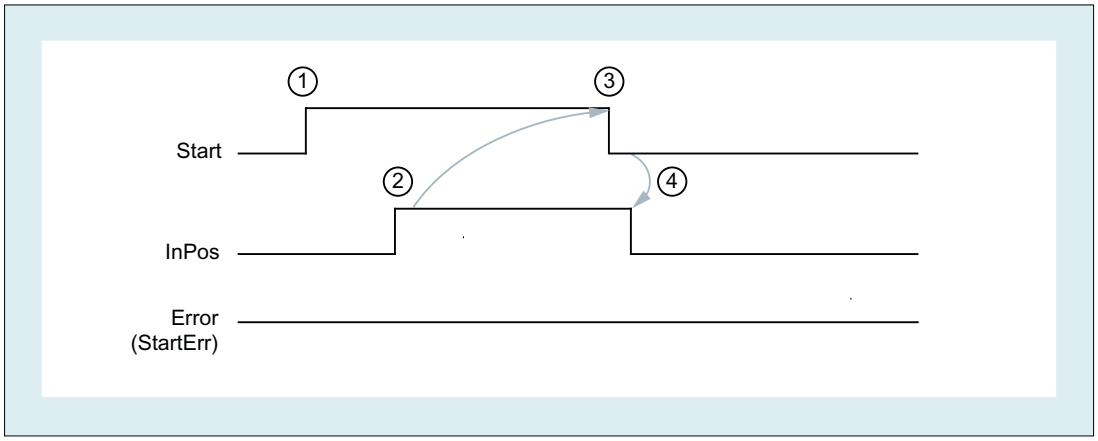
異常が有効なのは、「Error」 == TRUE

異常の原因は、パラメータ「State」に表示されます。

State	意味
PLC 側の問題の原因:	
1	軸/主軸の複数の機能が同時に起動された。
20	機能が位置決め完了せずに起動された。
30	軸/主軸が移動中に NC に転送された。
40	軸が NC プログラムによってプログラム指令された、NC 内部異常
50	固定割り当て PLC 軸。移動(ジョグ)または原点確立
60	固定割り当て PLC 軸。チャネル状態により起動できない
NC 側の問題の原因	
100	軸/主軸について誤った位置がプログラムされている(アラーム 16830 に対応)
101	プログラム指令速度が速すぎる
102	周速一定制御の値の範囲が正しくない(アラーム 14840 に対応)
104	スレーブ主軸:不正なプログラミング(アラーム 22030 に対応)
105	使用可能な検出器がない(アラーム 16770 に対応)
106	軸の位置決めが有効なままである(アラーム 22052 に対応)
107	原点マークが見つからない(アラーム 22051 に対応)
108	速度制御から位置制御への遷移がない(アラーム 22050 に対応)
109	原点マークが見つからない(アラーム 22051 に対応)
110	負の速度
111	指令速度==ゼロ
112	無効なギヤ選択
115	プログラム指令位置に到達していない
117	G96/G961 が NC 内で無効

State	意味
118	G96/G961 が NC 内で**まだ有効**
120	軸が割り出し軸ではない(アラーム 20072 に対応)
121	割り出し位置の異常(アラーム 17510 に対応)
125	DC (最短距離)は不可(アラーム 16800 に対応)
126	アブソリュート値のマイナスは不可(アラーム 16820 に対応)
127	アブソリュート値のプラスは不可(アラーム 16810 に対応)
128	直径指定に使用可能な径方向軸なし(アラーム 16510 に対応)
130	ソフトウェアリミットスイッチ、プラス(アラーム 20070 に対応)
131	ソフトウェアリミットスイッチ、マイナス(アラーム 20070 に対応)
132	作業領域リミット、プラス(アラーム 20071 に対応)
133	作業領域リミット、マイナス(アラーム 20071 に対応)
134	フレームは割り出し軸に使えない
135	「カップリングジョイント」による軸割り出しが有効(アラーム 17501 に対応)
136	「カップリングジョイント」による軸割り出しが有効で、軸の原点確立が未完(アラーム 17503 に対応)
137	座標変換された主軸/軸に対する主軸運転は不可(アラーム 22290 に対応)
138	軸:座標系別の作業領域プラス違反(アラーム 20082 に対応)
139	軸:座標系別の作業領域マイナス違反(アラーム 20082 に対応)
システム異常	
200	アラーム 450007 に対応
アラーム番号:**参照先:**『診断マニュアル』	

信号シーケンス:一般的な場合



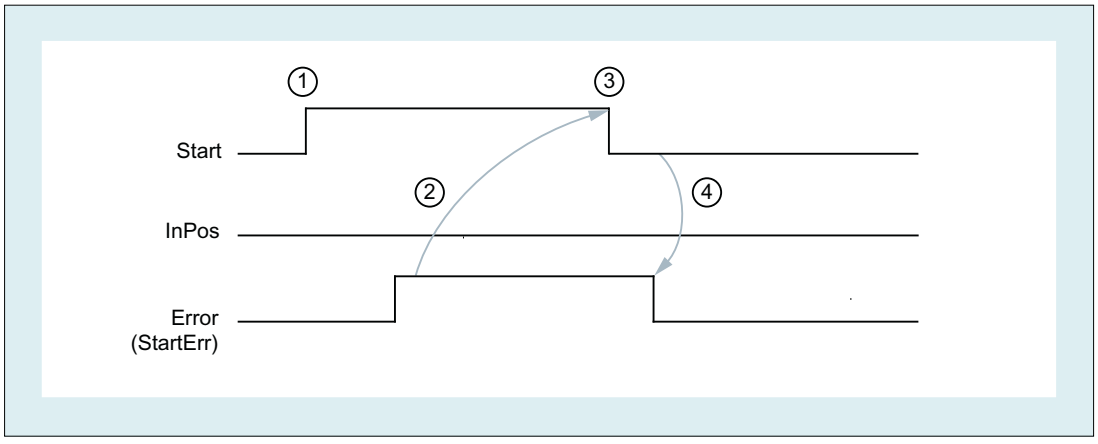
① PLC ユーザプログラム:信号立ち上がりエッジを使用した機能の開始:0 → 1

② NC:肯定応答、機能実行済み/位置決め完了

③ PLC ユーザプログラム:肯定応答検出後のリセット

④ FC18:肯定応答のリセット

信号シーケンス:異常時



① PLC ユーザプログラム:信号立ち上がりエッジを使用した機能の開始:0 → 1

② NC:否定応答、異常が発生しました

③ PLC ユーザプログラム:否定応答検出後のリセット

④ FC18:否定応答のリセット

呼び出し例

例 1:主軸の位置決め

```
//肯定応答で Start をリセット:
U M112.0;                    //InPos
R     M    100.0;            //Start
//否定応答、異常確認後(ステータス:MB114)、T12 で Start をリセット
U M113.0;                    //異常
U     E      6.4;            //T12 キー
R     M    100.0;            //Start
//T13 で起動
U     E      6.3;            //T13 キー
UN M 112.0;                  //InPos または Error = 0 のときにのみ再起動
UN M 113.0;
S M 100.0;

CALL FC18(
    Start :=  M100.0,
    Stop :=  FALSE,
    Funct :=  B#16#1,        //主軸の位置決め
    Mode :=  B#16#2,         //最短距離
    AxisNo :=  5,
    Pos :=  MD104,
    FRate :=  MD108,
    InPos :=  M112.0,
    Error   :=M113.0,
    State :=  MB114);
```

例 2 :主軸回転の起動:

```
CALL FC18(
    Start :=  M100.0,
    Stop :=  FALSE,
    Funct :=  B#16#2,        //主軸回転
    Mode :=  B#16#5,         //M4 に応じた回転方向
    AxisNo :=  5,
    Pos :=  0.0,
    FRate :=  MD108,
    InPos :=  M112.0,
    Error   :=M113.0,
    State :=  MB114);
```

例 3 :主軸揺動の起動

```
CALL FC18(
```

```
    Start :=  M100.0,
    Stop :=  FALSE,
    Funct   :=B#16#3,        //主軸揺動
    Mode :=  B#16#0,
    AxisNo :=  5,
    Pos :=  0.0,
    FRate :=  MD108,
    InPos :=  M112.0,
    Error   :=M113.0,
    State :=  MB114);
```

例 4 :割り出し軸の移動

```
CALL FC18(
    Start :=  M100.0,
    Stop :=  FALSE,          //不使用
    Funct :=  B#16#4,        //割り出し軸の移動
    Mode :=  B#16#0,         //絶対位置決め
    AxisNo :=  4,
    Pos :=  MD104,           //REAL での初期設定:1.0;2.0;..
    FRate :=  MD108,
    InPos :=  M112.0,
    Error   :=M113.0,
    State :=  MB114);
```

例 5 :軸位置決め

```
CALL FC18(
    Start :=  M100.0,
    Stop :=  FALSE,          //不使用
    Funct :=  B#16#5,        //軸位置決め
    Mode :=  B#16#1,         //インクレメンタル位置決め
    AxisNo :=  6,
    Pos :=  MD104,
    FRate :=  MD108,
    InPos :=  M112.0,
    Error   :=M113.0,
    State :=  MB114);
```

14.17.23 FC19:MCP_IFM - MCP 信号のインターフェースへの伝送

機能

ブロック FC19 "MCP_IFM" (M バージョン、たとえば MCP 483 など)は、機械操作パネルから NC/PLC インタフェースにデータを転送するために使用します。

- モード
- 軸選択
- WCS/MCS 切り替え
- 移動キー
- オーバライド
- キー操作スイッチ

次の仕様は、有効なモードまたは選択された座標系に応じて、**送り速度オーバライド、軸移動キー**および **INC キー**に適用されます。

- **送り速度オーバライド**
  - 送り速度オーバライドは、選択されたチャネルのインタフェース、および軸のインタフェースに伝送されます。
  - HMI 信号「早送り用送り速度オーバーライド有効」が設定されている場合、「早送りオーバーライド」(DBB5)インタフェースバイトに加えて、送り速度オーバーライド信号が NC チャネルに伝送されます。(例外:スイッチ設定「ゼロ」)。「早送りオーバライド有効」もこの HMI 信号で設定します。
- **運転機能 INC と軸移動キー**
  - MCS が選択されると、信号は、選択された機械軸のインタフェースに伝送されます。
  - WCS が選択されると、信号は、パラメータ設定されたチャネルのジオメトリ軸インタフェースに伝送されます。
  - MCS と WCS の間で座標系が切り替わると、有効な軸は通常解除されます。

機械操作パネルの LED は、フィードバックでの選択に対応しています。

送り速度と主軸の Start/Stop は、インタフェースに伝送されるのではなく、「FeedHold」または「SpindleHold」信号としてモーダルに出力されます。ユーザーは、これらの信号を、送りまたは主軸停止を引き起こす他の信号に接続することができます(たとえば、次のファンクションの適切な入力信号を使って、FC10:AL_MSG)。同時に、関連 LED が点灯します。

機械操作パネルが故障すると、ここから出力される信号はゼロにプリセットされます。これは、「FeedHold」と「SpindleHold」出力信号にも当てはまります。

FC19、または FC24、FC25、FC26 も、1 回の PLC 周期で複数回呼び出すことができます。この場合、この周期における最初の呼び出しが LED 表示を駆動します。さらに、パラメータ設定されたブロックのすべての動作は、最初の呼び出しで実行されます。以降の呼び出しでは、削減されたチャネルとモードグループインタフェースの処理のレベルだけがおこなわれます。ジオメトリ軸には、この周期における最初のブロック呼び出しにおいてのみ、方向データが提供されます。

シングルブロック処理の選択/解除は、この周期における最初の呼び出しでのみおこなうことができます。

B#16#10 によってパラメータ「ModeGroupNo」が増加している場合は、2 つ目の機械操作パネルを処理することができます。パラメータ設定時に、モードグループ番号は下位ニブルに含まれます。

「BAGNo」= 0 または B#16#10⇒モードグループ信号が処理されません。

「ChanNo」= 0⇒チャネル信号の処理をおこないません。

INC 選択は、モードグループインタフェースに伝送されます。この指定の起動は、電源投入後すぐにこのブロックによって、DB10.DBX57.0 (モードグループ領域での INC 入力有効)を介して行われます。

さらに、FC19 ブロックにより、2 つの機械操作パネルも平行して処理することができます。この場合、OB1 周期での 2 つ目の機械操作パネルに対するブロックの呼び出しは、1 つ目の機械操作パネルの呼び出しの後にしてください。2 つの機械操作パネルのサポートは、 機械操作パネルブロックに存在しますが、制限されています。2 つの機械操作パネルに同等に割り当てられた複数の軸に対する軸選択の相互ロックはサポートされていません。

直交手動移動

機械操作パネルの R11 方向キー(WCS/MCS の左隣にある)で、「工具オリエンテーションの手動運転」機能が使用可能になります。このために、DB7 で FB1 入力パラメータ「MCP_IF_TCS」を介して起動してください。

「MCP_IF_TCS」= TRUE の場合は、R11 キーで「工具オリエンテーションの手動運転」に切り替わります。これにより、Z キー(R3)が固定的に FC19 で選択されます。方向キーは、関連するチャネルの 3 番目のジオメトリ軸に作用します。

フレキシブルな軸設定

機械軸番号について軸選択または方向キーの割り当てをフレキシブルにおこなうことができます。

特に 2 つのチャネルと 2 つのモードグループを使用するアプリケーションについて、並列に実行される 2 つの MCP を使用できるように、MCP ブロックによってより適切なサポートが提供されるようになりました。軸番号は、関連 MCP の軸テーブル内で、MCP ブロックのパラメータ設定済モードグループ番号でも指定されることに注意してください。

このフレキシビリティを提供するために、軸番号のテーブルが DB10 に保存されます。

1 番目の機械操作パネルでは、テーブルがバイト 8 から始まり(シンボル名称:MCP1AxisTbl[1..22])、**2 番目の**機械操作パネルでは、バイト 32 から始まります(シンボル名称:MCP2AxisTbl[1..22])。ここでは、機械軸番号をバイト単位で入力してください。

軸テーブルに値 0 を入力することができます。不正な軸番号を検出するためのチェックはおこなわれません。つまり、誤った入力をおこなうと、PLC が停止することがあります。

FC19 については、**軸選択の最大可能数**も制限できます。この上限は、第 1 機械操作パネルについては、DB10.DBW30 に設定します(シンボル名称:MCP1MaxAxis)。または、第 2 機械操作パネルについては、DB10.DBW54 に設定します(シンボル名称:MCP2MaxAxis)。

初期設定は 0 であり、構成された軸の最大数に対応します。軸番号と制限も自動的に補正できます。後で、新しい軸を FC19 のために選択してください。関連する方向キーを使用して軸を移動している間は、軸番号を切り替えることはできません。
互換モードは、両方の MCP について軸番号 **1 ～ 9** で設定されています。また、構成済の軸数に制限されます。

例

特殊アプリケーションにより、FC19 で 9 軸以上の制御をおこないます。次の実施手順をおすすめします。

- MCP 上に空きキーを予約します。
- フリップフロップとしてこのキーを解析します。
- 立ち上がりエッジおよび立ち下がりエッジとしてフリップフロップ出力を解析します。
- 立ち上がりエッジで、軸テーブル(DB10)に 1 セットの軸番号を書き込み、このキーを介して LED をオンにします。
- 立ち下がりエッジで、軸テーブル(DB10)に別のセットの軸番号を書き込み、このキーを介して LED をオフにします。

ファンクションの宣言

```
FUNCTION FC19:VOID                  //シンボル名称:MCP_IFM

   VAR_INPUT
      BAGNo :           BYTE;
      ChanNo:           BYTE;
      SpindleIFNo:      BYTE;
   END_VAR

   VAR_OUTPUT
      FeedHold :        BOOL;
      SpindleHold :     BOOL;
   END_VAR

BEGIN
END_FUNCTION
```

仮パラメータの説明

信号	種別	タイプ	値の範囲	意味
BAGNo :	I	BYTE	B#16#00 ～ B#16#0A	**第 1 MCP**:モード信号の伝送先となるモードグループの番号
			B#16#10 - B#16#1A	**第 2 MCP**:モード信号の伝送先となるモードグループの番号
ChanNo:	I	BYTE	B#16#00 ～ B#16#0A	チャネル信号の伝送先となるチャネルの番号
SpindleIFNo:	I	BYTE	0 ～ 31 (B#16#1F)	主軸データの伝送先となる軸/主軸の番号(関連する機械軸の番号)
FeedHold :	O	BOOL	0 (FALSE)、1 (TRUE)	MCP からの送り停止、モーダル
SpindleHold :	O	BOOL	0 (FALSE)、1 (TRUE)	MCP からの主軸停止、モーダル

ユーザーインタフェースへの MCP 選択信号

表 14-3　キー操作スイッチ

ソース: MCP スイッチ	ターゲット: インタフェース DB
位置 0	DB10.DBX56.4
位置 1	DB10.DBX56.5
位置 2	DB10.DBX56.6
位置 3	DB10.DBX56.7

表 14-4　運転モードと運転機能

ソース: MCP キー	ターゲット: インタフェース DB (パラメータ BAGNo) モードグループ 1 用の表示
AUTOMATIC	DB11.DBX0.0
MDI	DB11.DBX0.1
JOG	DB11.DBX0.2
REPOS	DB11.DBX1.1
REF	DB11.DBX1.2
TEACH IN	DB11.DBX1.0
INC 1 ... 10 000, INC Var.	DB11.DBX2.0 ～ 2.5

表 14-5　方向キー、早送りオーバライド

ソース: MCP キー	ターゲット: インタフェース DB (パラメータ ChanNo)
方向キー+	DB21、... .DBX12.7
方向キー-	DB21、... .DBX12.6
早送りオーバライド	DB21、... .DBX12.5
方向キー+	DB21、... .DBX16.7
方向キー-	DB21、... .DBX16.6
早送りオーバライド	DB21、... .DBX16.5
方向キー+	DB21、... .DBX20.7

ソース: MCP キー	ターゲット: インタフェース DB (パラメータ ChanNo)
方向キー-	DB21、... .DBX20.6
早送りオーバライド	DB21、... .DBX20.5

ソース: MCP キー	ターゲット: インタフェース DB (すべての軸 DB)
方向キー+	DB31, ...DBX4.7
方向キー-	DB31, ...DBX4.6
早送りオーバライド	DB31, ...DBX4.5

伝送は、選択した軸に対応します。関連インタフェースビットは、選択されなかった軸についてオフされます。

表 14-6　　オーバライド

ソース: MCP スイッチ	ターゲット: インタフェース DB (パラメータ ChanNo)
送り速度オーバライド	DB21、... .DBB4

ソース: MCP スイッチ	ターゲット: インタフェース DB (すべての軸 DB)
送り速度オーバライド	DB31、... .DBB0 (選択された軸番号)。第 1 MCP の送り速度オーバーライドがすべての軸に適用される。
主軸オーバライド	DB31, ...DBB19 (パラメータ SpindleIFNo)

表 14-7　　チャネル信号

ソース: MCP キー	ターゲット: インタフェース DB (パラメータ ChanNo)
NC スタート	DB21、... .DBX7.1
NC ストップ	DB21、... .DBX7.3
リセット	DB21、... .DBX7.7
シングルブロック	DB21、... .DBX0.4

表 14-8　送り、主軸信号

ソース: **MCP キー**	ターゲット: **FC 出力パラメータ**
送り停止 送り有効	パラメータ:メモリとリンクされた 「FeedHold」、LED が制御される。
主軸停止 主軸有効	パラメータ:メモリとリンクされた 「SpindleHold」、LED が制御される。

表 14-9　直交手動移動

出典 : **MCP キー**	ターゲット: **インタフェース DB (パラメータ ChanNo)**
R11 方向キー-	DB21、... .DBB392
方向キー+	DB21、... .DBX20.7
方向キー-	DB21、... .DBX20.6

表示制御用のユーザーインタフェースからのアンサーバック信号

表 14-10　運転モードと運転機能

ターゲット: **MCP - LED**	ソース: **インタフェース DB (パラメータ BAGNo)** **モードグループ 1 用の表示**
AUTOMATIC	DB11.DBX6.0
MDI	DB11.DBX6.1
JOG	DB11.DBX6.2
REPOS	DB11.DBX7.1
REF	DB11.DBX7.2
TEACH IN	DB11.DBX7.0

ターゲット: **MCP - LED**	ソース: **インタフェース DB (パラメータ BAGNo)** **モードグループ 1 用の表示**
INC 1 ... 10 000, INC Var.	DB11.DBX8.0 ～ 8.5

表 14-11　チャネル信号

ターゲット: MCP - LED	ソース: インタフェース DB (パラメータ ChanNo)
NC スタート	DB21、... .DBX35.0
NC ストップ	DB21、... .DBX35.2 または DB21、... .DBX35.3
シングルブロック	DB21、... .DBX0.4

注記

方向キーの LED は、方向キーを操作することによって制御されます。
軸選択と WCS/MCS の LED は、関連するキーを操作することによって制御されます。

呼び出し例

```
CALL FC19 (     //インタフェースへの機械操作パネル M 型信号
   BAGNo :=          B#16#1,           //モードグループ番号 1
   ChanNo :=         B#16#1,           //チャネル番号 1
   SpindleIFNo :=    B#16#4,           //主軸インターフェース番号= 4
   FeedHold :=       m22.0,            //送り停止信号モーダル
   SpindleHold :=    db2.dbx151.0);    //主軸停止モーダル
                                       //メッセージ DB 内
```

このパラメータ設定により、最初のモードグループ、最初のチャネルおよびすべての軸に信号が送信されます。また、主軸オーバーライドは、4 番目の軸/主軸インターフェースに伝送されます。フィードホールド信号はビットメモリ 22.0 に渡され、主軸停止信号はデータブロック DB2、データビット 151.0 に渡されます。

軸選択の再接続

適切な軸または主軸に対する軸選択キーを柔軟に割り当てるために、FC19 の**変更または再プログラムは必要ありません。**軸テーブル DB10.DBB8 にそのまま軸番号を入力し、必要に応じて以下に従います。軸番号を軸表 DB10.DBB8 に入力し、必要に応じてこれに従うだけで済みます。

例

主軸は、4 番目の機械軸として定義し、軸キー 9 を使用して選択してください。

対策:
第 4 軸用の DB10 バイト(8+(9-1))に値 4 を入力してください。

```
CALL FC19 (            //インタフェースへの信号
   BAGNo :=            B#16#1,      //モードグループ番号 1
   ChanNo :=           B#16#1,      //チャネル番号 1
   SpindleIFNo :=      B#16#4,      //主軸インターフェース番号= 4

   FeedHold :=         m30.0,       //送り停止信号モーダル
   SpindleHold :=      m30.1);      //主軸停止モーダル
```

14.17.24 FC21:転送 - データ交信 NC/PLC

14.17.24.1 機能

ブロック FC21 を使用して、PLC と NC の間でデータを交換します。FC21 が呼び出されると、次の PLC 基本プログラム周期が開始されるまで待機せず、データが直ちに伝送されます。

データ転送は、パラメータ「Enable」=1 でブロック FC21 を呼び出すことによって起動します。

機能

このブロックは、次の機能を提供します。

- シンクロナイズドアクション信号:PLC→NC チャネル
- シンクロナイズドアクション信号:NC チャネル→PLC
- PLC-NC 間での高速データ交換(NC の読み取り機能)
- PLC-NC 間での高速データ交換(NC の書き込み機能)
- NC チャネルへの更新信号
- 軸への更新信号(ユーザーインターフェースのデータバイト 2)
- 軸への更新信号(ユーザーインターフェースのデータバイト 4)

14.17.24.2 ファンクションの宣言

ファンクションの宣言

```
VAR_INPUT
          Enable :BOOL;
          Funct:BYTE;
          S7Var :ANY ;
          IVar1 :INT;
          IVar2 :INT;
END_VAR

VAR_OUTPUT
          Error :BOOL;
          ErrCode :INT;
END_VAR
```

14.17.24.3 仮パラメータの説明

仮パラメータの説明

表記	種別	タイプ	数値の範囲	説明	
Enable :	I	BOOL	0 (FALSE)、1 (TRUE)	1:	データの転送
Funct:	I	BYTE	1, 2, 3, ... 7	1:	チャネルへのシンクロナイズドアクション
				2:	チャネルからのシンクロナイズドアクション
				3:	データの読み取り
				4:	データの書き込み
				5:	チャネルへの制御信号
				6:	軸への制御信号
				7:	軸への制御信号
S7Var :	I	ANY	S7 データ記憶領域	「Funct」による	
IVAR1:	I	INT	---	「Funct」による	
IVAR2:	I	INT	---	「Funct」による	

表記	種別	タイプ	数値の範囲	説明	
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	1:	異常が有効
ErrCode :	O	INT	---	「Funct」による	

14.17.24.4 機能 1、2:チャネルへ/チャネルからのシンクロナイズドアクション通知

シンクロナイズドアクションは、PLC により解除または起動できます。

データ領域は、DB21、... .DBB 300...307(チャネルへ)と、DB21、... .DBB 308 ...315(チャネルから)のユーザーインタフェースに配置されています。このファンクションでは、パラメータ「S7Var」は解析されませんが、実際のパラメータを割り当ててください(呼び出し例を参照してください)。FC21 が処理されると直ちに、NC へ/からデータが伝送されます。

表記	種別	タイプ	数値の範囲	説明	
Enable :	I	BOOL	0 (FALSE)、1 (TRUE)	1:	データの転送
Funct:	I	BYTE	1, 2	1: 2:	チャネルへ チャネルから
S7Var :	I	ANY	S7 データ記憶領域	使用しない	
IVAR1:	I	INT	1、2、... 最大チャネル数	チャネル番号	
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	1:	異常が有効
ErrCode :	O	INT	1, 10	1: 10 : 	「Funct」は無効 チャネル番号が無効

呼び出し例:

```
FUNCTION FC100:VOID
VAR_TEMP
       myAny:ANY ;
END_VAR

BEGIN
NETWORK

//Deactivate synchronized actions with ID3, ID10 and ID31 in NC channel 1 :
SYAK:   AUF   DB21;
```

```
        SET;
        S     DBX300. //ID3
              2;
        S     DBX301. //ID10
              1;
        S     DBX303. //ID31
              6;
        L     B#16#1;
        T     MB11;
        SPA   TRAN;

//Synchronized actions from NCK channel 1:
SYVK:   L B#16#2;
        T MB11;
TRAN:   CALL FC21 (
             Enable   := M 10.0,      //if TRUE, FC 21 active
             Funct    := MB 11,
             S7Var    := #myAny,      //Not used
             IVAR1    := 1,           //Channel no.
             IVAR2    := 0,
             Error    := M 10.1,
             ErrCode  := MW 12);

END_FUNCTION
```

14.17.24.5 機能 3、4:PLC-NC 間での高速データ交換

概要

NC と PLC 間のデータの高速交信を可能にするインタフェースとして、個別の内部データ領域が提供されます。インタフェースには、4096 バイトが含まれます。PLC アクセス操作(読み取り/書き込み)は、FC21 を介しておこないます。インタフェースの内部構造は、ユーザーによって単独で定義され、NC および PLC 側で同じ定義を正確に行う必要があります。

このデータは命令$A_DBB[x]、$A_DBW[x]、$A_DBD[x]、$A_DBR[x]を使用して NC プログラムによりアクセスできます(参照:『リストマニュアル』、「システム変数」)。

データフィールドの具体的なアドレスは、パラメータ「IVAR1」のバイトオフセット(0 ～ 4095)により指定されます。この場合、配置はデータ形式に応じて選択してください。つまり、DWORD は 4 バイト境界で、WORD は 2 バイト境界で始まります。バイトは、データフィールド内の任意のオフセットに配置できます。個別ビットはアクセスできま

せん。FC21 により、バイトアクセスに変換されます。データタイプ情報とデータ量は ANY パラメータから取得され、S7Var を介して伝送されます。

追加プログラミングに関連する方法を採らない場合、NCU および PLC の双方から、1 バイトおよび 2 バイトのアクセス操作についてのみデータの完全性が確保されます。2 バイトの一貫性については、データタイプ WORD または INT の場合にかぎり当てはまり、データタイプ BYTE の場合には当てはまりません。

データタイプがこれより長い場合、または一貫性のある伝送が必要な配列の伝送の場合は、セマフォバイトをパラメータ「IVAR2」で使用してください。FC21 は、このパラメータを使用して、ブロックの妥当性または一貫性を判断します。この処理を NC で、つまり、NC プログラムでセマフォバイトを書き込むか削除することで、補助してください。セマフォバイトは、ユーザーデータと同じデータフィールドに格納されます。

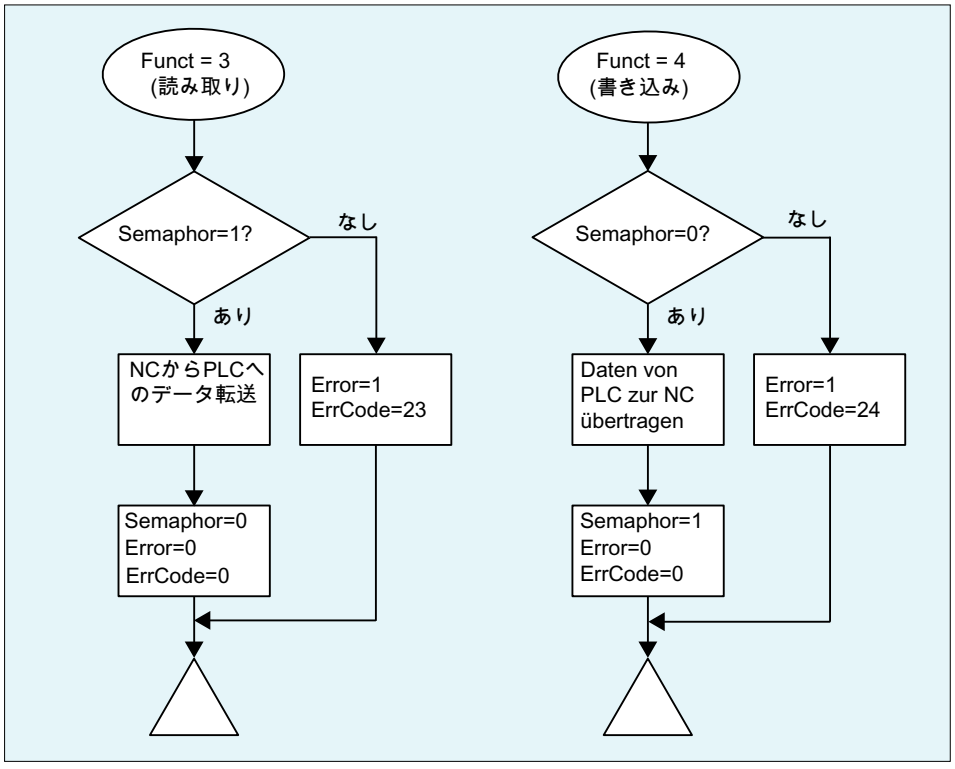
セマフォバイトは、「IVAR2」で、0 ～ 4095 の値で識別されます。

PLC は、ユーザーデータを伝送するのと同じ呼び出しで FC21 を介してセマフォバイトを読み取って書き込みます。PLC プログラマは、インタフェースでセマフォ変数のみを定義する必要があります。NC プログラムを介した NC からのアクセスの場合、セマフォメカニズムは、下記フローチャートに応じて個々の命令を使用してプログラムしてください。この手順は、変数の読み取りと書き込みの場合で異なります。

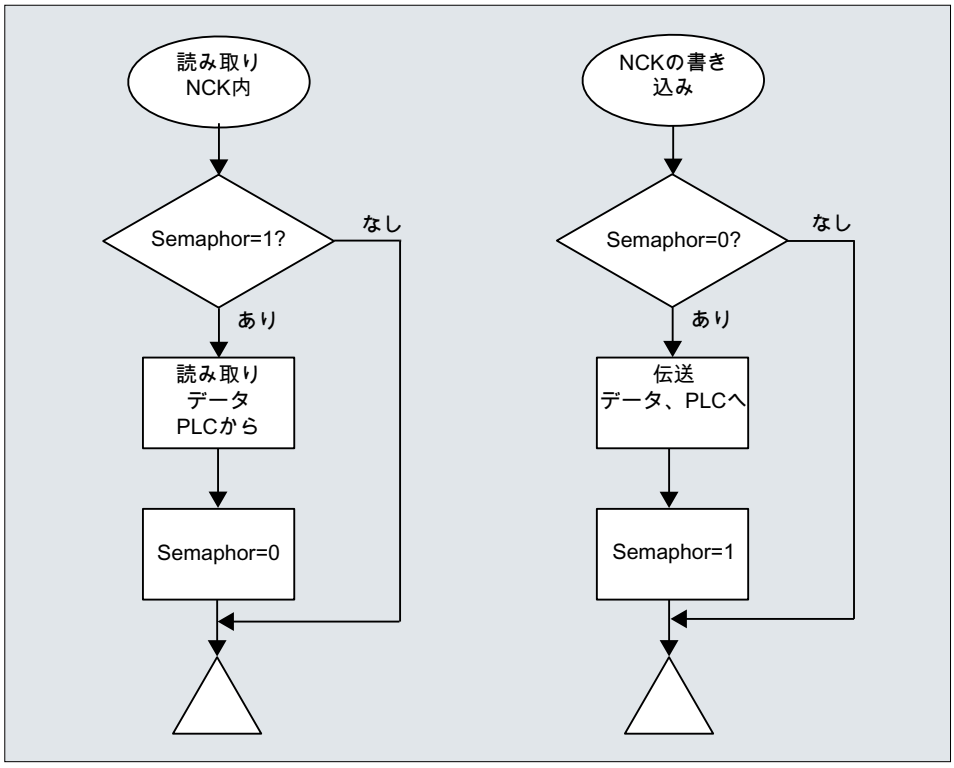
このセマフォ技術で直接サポートできるのは、個々の変数または配列(フィールド)だけです。転送構造は、個別ジョブに分割する必要があります。この場合ユーザーは、適切なセマフォメカニズムをプログラムすることにより、この構造のデータ完全性を確保する必要があります。

「IVAR2」= -1 が設定されている場合は、データがセマフォなしで伝送されます。

PLC でのセマフォを使用したデータ交換(FC21 の図)



NC の基本構造:



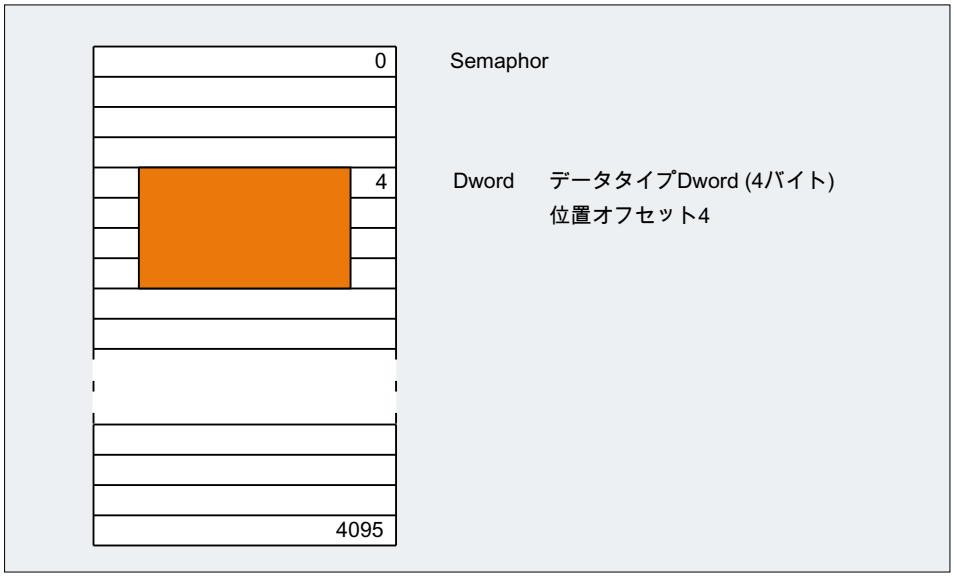
変数値の範囲

表記	種別	タイプ	数値の範囲	説明	
Enable :	I	BOOL	0 (FALSE)、1 (TRUE)	1:	データの転送
Funct:	I	BYTE	3, 4	3:	データの読み取り
				4:	データの書き込み
S7Var :	I	ANY	S7 データ領域、ローカルデータを除く	送信元/送信先データ記憶領域	
IVAR1:	I	INT	0 ... 4095	位置オフセット	
IVAR2:	I	INT	-1 ... 4095	セマフォバイト セマフォなしの転送: -1	
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	1:	異常が有効

表記	種別	タイプ	数値の範囲	説明	
ErrCode :	O	INT	20, 21, 22, 23, 24, 25	20:	配置異常
				21:	不正な位置オフセット
				22:	不正なセマフォバイト
				23:	読み取る新規データがありません
				24:	データを書き込めません
				25:	S7Var で設定されたローカルデータ

例 1:位置オフセット 4 からバイト 0 でセマフォを使用して DWORD を読み取り、メモリ内にダブルワード 100 を保存する

- データタイプ Dword (4 バイト)
- 位置オフセット 4



PLC プログラミング

```
CALL   FC21:
         Enable     := M 10.0,              // TRUE の場合は FC21 が有効
         Funct      := B#16#3,              //Read data
         S7Var      := P#M 100.0 DWORD 1,
         IVAR1      := 4,
```

```
        IVAR2      := 0,
        Error      := M 10.1,
        ErrCode    := MW12);
UN     M10.1;                 //Enable while 1, until value is read
R      M10.0;
```

シンクロナイズドアクションによる NC のプログラム

- PLC へのデータの書き込み - バイト 0 がセマフォとして機能:
  ID=1 WHENEVER $A_DBB[0] == 0 DO $A_DBR[4] = $AA_IM[X] $A_DBB[0] = 1
- PLC からのデータの読み込み - バイト 1 がセマフォとして機能:
  ID=2 WHENEVER $A_DBB[1] == 1 DO $R1 = $A_DBR[12] $A_DBB[1] = 0

例 2:セマフォなしで位置オフセット 8 から WORD を読み取り、メモリにワード 104 を保存する

```
CALL   FC21:
        Enable     := M 10.0,              // TRUE の場合は FC21 が有効
        Funct      := B#16#3,              //Read data
        S7Var      :=P#M 104.0 WORD 1,
        IVAR1      := 8,
        IVAR2      := -1,
        Error      := M 10.1,
        ErrCode    :=MW12);
```

14.17.24.6 機能 5:チャネルへの制御信号更新

この機能の目的は、周期的なデータ伝送の合間に高速で重要な制御信号を伝送することです。ユーザーインタフェース DB21、...のデータバイト 6 と 7 は NC に伝送されます。このチャネルは、パラメータ「IVAR1」で指定されます。この結果、たとえば、送り無効、読み込み停止を PLC サイクルの外で伝送できます。

表記	種別	タイプ	数値の範囲	説明	
Enable :	I	BOOL	0 (FALSE)、1 (TRUE)	1:	データの転送
Funct:	I	BYTE	5	5:	チャネルへの制御信号
S7Var :	I	ANY	S7 データ記憶領域	使用しない	
IVAR1:	I	INT	1.最大チャネル	チャネル番号	

表記	種別	タイプ	数値の範囲	説明	
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	1:	異常が有効
ErrCode :	O	INT	1, 10	1:	「Funct」は無効
				10:	チャネル番号は無効

14.17.24.7 機能 6:軸への制御信号更新

機能 6 の目的は、周期的なデータ伝送の合間に高速で重要な制御信号を伝送することです。アプリケーションインタフェース DB31、...の**データバイト 2** は、NC に伝送されます。この伝送は、起動された軸すべてに対して実行されます。この結果、たとえば、コントローライネーブルを PLC サイクルの外で伝送できます。

表記	種別	タイプ	数値の範囲	説明	
Enable :	I	BOOL	0 (FALSE)、1 (TRUE)	1:	データの転送
Funct:	I	BYTE	6	6:	軸への制御信号
S7Var :	I	ANY	S7 データ記憶領域	使用しない	
IVAR1:	I	INT	0		
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	1:	異常が有効
ErrCode :	O	INT	1	1:	「Funct」は無効

14.17.24.8 機能 7:軸への制御信号更新

機能 7 の目的は、周期的なデータ伝送の合間に高速で重要な制御信号を伝送することです。アプリケーションインタフェース DB31、...の**データバイト 4** は、NC に伝送されます。この伝送は、起動された軸すべてに対して実行されます。この結果、たとえば、送り停止を PLC サイクルの外で伝送できます。

表記	種別	タイプ	数値の範囲	説明	
Enable :	I	BOOL	0 (FALSE)、1 (TRUE)		
Funct:	I	BYTE	7	7:	軸への制御信号

表記	種別	タイプ	数値の範囲	説明	
S7Var :	I	ANY	S7 データ記憶領域	使用しない	
IVAR1:	I	INT	0		
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	1:	異常が有効
ErrCode :	O	INT	1	1:	「Funct」は無効

14.17.25 FC22:TM_DIR - 工具管理用の方向選択

通知
使用/用途 ブロック FC22「TM_DIR」を使用できるのは、工具管理と組み合わせた場合に限ります。

機能

工具マガジンやリボルバ(割り出し軸)など、ロケーション番号を参照してください。ブロック FC22「TM_DIR」は、位置決めの移動の最短経路と方向を、目標位置と現在の位置に基づいて提供します。

出力部

- 入力 FC22:「Start」= 1⇒出力が周期的に更新されます。
- 入力 FC22:「Start」= 0⇒出力が周期的に未定義になります。

特殊な位置決め

特殊な位置決め(入力 FC22:「Offset」> 0)によって方向を選択する場合は、新しい目標位置が、目標位置、特殊な位置決めのオフセット、およびマガジンロケーション番号から計算されます。

New_target position = (目標位置- (特殊な位置-1))負の MODULO number_of_magazine locations

新しい目標位置は、ユーザーが要求する目標位置が特殊位置のロケーション番号に対応するように、マガジンを位置決めする必要があるロケーション番号に対応します。

方向の最適化は、特殊位置決めを使用する場合も使用しない場合も有効です。

呼び出し

マガジンごとに、適切なパラメータ設定を使用して、このブロックを 1 回呼び出してください。

参照先

- 工具管理機能のその他の PI サービス:
  - FB4:PI_SERV - PI サービス要求 (ページ 1193)
  - FC7:TM_REV - リボルバの工具交換伝送ブロック (ページ 1271)
  - FC8:TM_TRANS - 工具管理機能用伝送ブロック (ページ 1275)
- 機能マニュアル 工具管理機能

ファンクションの宣言

```
FUNCTION FC22:VOID
//NAME:            TM_DIR
VAR_INPUT
    MagNo:         INT;
    ReqPos:        INT;
    ActPos:        INT;
    オフセット:    BYTE;
    Start:         BOOL;
END_VAR

VAR_OUTPUT
    Cw:            BOOL;
    Ccw:           BOOL;
    InPos:         BOOL;
    Diff:          INT;
    Error :        BOOL;
END_VAR
BEGIN
END_FUNCTION
```

仮パラメータの説明

信号	タイプ	タイプ	値の範囲	説明
MagNo:	I	INT	1, 2, 3, ...	マガジン番号
ReqPos:	I	INT	1, 2, 3, ...	目標位置(マガジンロケーション番号)
ActPos:	I	INT	1, 2, 3, ...	現在位置(マガジンロケーション番号)
オフセット:	I	BYTE	0, 1, 2, ...	特殊位置決めのオフセット
Start:	I	BOOL	0 (FALSE)、1 (TRUE)	1 =計算開始
Cw:	O	BOOL	0 (FALSE)、1 (TRUE)	1 = マガジンを時計方向に移動
Ccw:	O	BOOL	0 (FALSE)、1 (TRUE)	1 =マガジンを反時計回りに移動
InPos:	O	BOOL	0 (FALSE)、1 (TRUE)	1 = インポジション
Diff:	O	INT	0, 1, 2, ...	差分軌跡の絶対値 (最短軌跡)
エラー:	O	BOOL	0 (FALSE)、1 (TRUE)	1 =異常

呼び出し例

```
CALL FC22 (                    //工具管理方向選択
                               //入力
   MagNo :=      2,            //マガジン番号
   ReqPos :=     mw 20,        //目標位置
   ActPos :=     mw 22,        //現在の位置
   Offset :=     b#16#0,       //特殊位置決めのオフセット
   Start :=      m 30.4,       //開始トリガ
                               //出力
   Cw :=         m 30.0,       //マガジンを時計回りに移動
   Ccw :=        m 30.1,       //マガジンを反時計回りに移動
   InPos :=      m 30.2,       //マガジンインポジション
```

```
Diff :=       mw 32,       //差分軌跡
Error :=      m30.3        //異常が発生しました
);
```

14.17.26 FC24:MCP_IFM2 - インターフェースへの MCP 信号の伝送

機能

ブロック FC24 "MCP_IFM2" (M タイプ、たとえば MCP 310 など)は、機械操作パネルから NC/PLC インタフェースにデータを転送するために使用します。

- モード
- 軸選択
- WCS/MCS 切り替え
- 移動キー
- オーバライドまたはオーバライドの転写信号
- キー操作スイッチ位置

次の仕様は、有効なモードまたは選択された座標系に応じて、**送り速度オーバライド、軸移動キー**および **INC キー**に適用されます。

- **送り速度オーバライド**
  - 送り速度オーバライドは、選択されたチャネルのインタフェース、および軸のインタフェースに伝送されます。
  - HMI 信号「早送り用送り速度オーバライド有効」が設定されている場合、「早送りオーバライド」(DBB 5)インタフェースバイトに加えて、送り速度オーバライド信号が NC チャネルに伝送されます。(例外:スイッチ設定「ゼロ」)。「早送りオーバライド有効」もこの HMI 信号で設定します。
- **INC と軸移動キーに対する運転機能**
  - MCS が選択されると、信号は、選択された機械軸のインタフェースに伝送されます。
  - WCS が選択されると、信号は、パラメータ設定されたチャネルのジオメトリ軸インタフェースに伝送されます。
  - MCS と WCS の間で座標系が切り替わると、有効な軸は通常解除されます。

機械操作パネルの対応する LED は、関連する選択の応答から得られます。

送り速度と主軸の Start/Stop は、インタフェースに伝送されるのではなく、「FeedHold」または「SpindleHold」信号としてモーダルに出力されます。ユーザーは、これらの信号

を、送りまたは主軸停止を引き起こす他の信号に接続することができます(たとえば、次のファンクションの適切な入力信号を使って、FC10:AL_MSG)。同時に、関連 LED が点灯します。

主軸方向(+、-)を直接切り替えることもできませんが、出力パラメータ「SpindleDir」として使用できます。この結果、たとえば、FC 18 をパラメータ設定できます。主軸イネーブル信号は、パラメータ「SpindleHold」でも切り替えられます。(軸)方向キーを使用して主軸を移動できるように主軸を軸として事前選択するのも、主軸を直接移動できる方法の 1 つです。

機械操作パネルが故障すると、ここから出力される信号はゼロにプリセットされます。これは、「FeedHold」と「SpindleHold」出力信号にも当てはまります。

FC24、または FC19、FC25、FC26 も、1 回の PLC 周期で複数回呼び出すことができます。この場合、この周期における最初の呼び出しが LED 表示を駆動します。さらに、パラメータ設定されたブロックのすべての動作は、最初の呼び出しで実行されます。以降の呼び出しでは、削減されたチャネルとモードグループインタフェースの処理のレベルだけがおこなわれます。ジオメトリ軸には、この周期における最初のブロック呼び出しにおいてのみ、方向データが提供されます。

シングルブロック処理の選択/解除は、この周期における最初の呼び出しでのみおこなうことができます。

B#16#10 によってパラメータ「ModeGroupNo」が増加している場合は、2 つ目の機械操作パネルを処理することができます。パラメータ設定時に、HHU 番号は下位ニブル(下位 4 ビット)に含まれます。

「BAGNo」= 0 または B#16#10⇒モードグループ信号が処理されません。

「ChanNo」= 0⇒チャネル信号の処理をおこないません。

INC 選択は、モードグループインタフェースに伝送されます。この指定の起動は、電源投入後すぐにこのブロックによって、DB10.DBX57.0 (モードグループ領域での INC 入力有効)を介して行われます。

さらに、このブロックにより、2 つの機械操作パネルを平行して処理することができます。この場合、OB1 周期の第 2 機械操作パネルに対するブロック呼び出しは、第 1 MCP の呼び出しの後に設定してください。2 つの MCP のサポートは、操作パネルブロック内である程度まで提供されます(標準では、2 つの MCP 上での割り当てが同じである軸選択の相互インタロックに対するサポートは提供されていません)。

キー操作スイッチ位置

ソフトウェアバージョン 4.5 SP2 以降では、FC24 におけるキー操作スイッチ信号もユーザーインターフェースに伝送されます(DBX56.5 ～ 7)。この伝送は、キースイッチが MCP に取り付けられているかどうかに関係なく行われます。

注記

詳しくは、「FC19:MCP_IFM - MCP 信号のインターフェースへの伝送 (ページ 1314)」を参照してください。

フレキシブルな軸設定

機械軸番号について軸選択または方向キーの割り当てをフレキシブルにおこなうことができます。

特に 2 つのチャネルと 2 つのモードグループを使用するアプリケーションについて、並列に実行される 2 つの MCP を使用できるように、MCP ブロックによってより適切なサポートが提供されるようになりました。軸番号は、関連 MCP の軸テーブル内で、MCP ブロックのパラメータ設定済モードグループ番号でも指定されることに注意してください。

このフレキシビリティを提供するために、軸番号のテーブルが DB10 に保存されます。**1 番目の**機械操作パネル(MCP)では、テーブルはバイト 8 から始まり(シンボル名称:MCP1AxisTbl[1..22])、**2 番目の**機械操作パネル(MCP)では、バイト 32 から始まります(シンボル名:MCP2AxisTbl[1..22])。ここでは、機械軸番号をバイト単位で入力してください。軸テーブルに値 0 を入力することができます。不正な軸番号を検出するためのチェックはおこなわれません。つまり、誤った入力をおこなうと、PLC が停止することがあります。

FC24 については、**軸選択の最大可能数**も制限できます。
この上限は、第 1 機械操作パネルについては、DB10.DBW30 に設定します (シンボル名:MCP1MaxAxis)。または、第 2 機械操作パネルについては、DB10.DBW54 に設定します(シンボル名:MCP2MaxAxis)。

初期設定は 0 であり、構成された軸の最大数に対応します。軸番号と制限も自動的に補正できます。後で、新しい軸を FC24 のために選択してください。関連する方向キーを使用して軸を移動している間は、軸番号を切り替えることはできません。互換モードは、両方の MCP について軸番号 **1 ～ 6** でプリセットされます。また、構成済の軸数に制限されます。

ファンクションの宣言

```
FUNCTION FC24:VOID
//NAME:                 MCP_IFM2

VAR_INPUT
    BAGNo :             BYTE;
    ChanNo:             BYTE;
    SpindleIFNo:        BYTE;
END_VAR

VAR_OUTPUT
    FeedHold :          BOOL;
    SpindleHold :       BOOL;
    SpindleDir:         BOOL;
END_VAR

BEGIN
END_FUNCTION
```

仮パラメータの説明

信号	種別	タイプ	値の範囲	意味
BAGNo :	I	BYTE	B#16#00 ～ B#16#0A	**第 1 MCP**:モード信号の伝送先となるモードグループの番号
			B#16#10 - B#16#1A	**第 2 MCP**:モード信号の伝送先となるモードグループの番号
ChanNo:	I	BYTE	B#16#00 ～ B#16#0A	チャネル信号の伝送先となるチャネルの番号
SpindleIFNo :	I	BYTE	0 ～ 31 (B#16#1F)	主軸データの伝送先となる軸/主軸の番号(関連する機械軸の番号)
FeedHold :	O	BOOL	0 (FALSE)、1 (TRUE)	MCP からの送り停止、モーダル
SpindleHold :	O	BOOL	0 (FALSE)、1 (TRUE)	MCP からの主軸停止、モーダル

信号	種別	タイプ	値の範囲	意味	
SpindleDir:	O	BOOL	0 (FALSE)、1 (TRUE)	主軸の回転方向	
				0:	回転方向+ (左)
				1:	回転方向- (右)

呼び出し例

```
CALL FC24 (                          //スリム機械操作パネル M タイプ
                                     //インタフェースへの信号
    BAGNo :=       B#16#1,           //モードグループ番号 1
    ChanNo :=      B#16#1,           //チャネル番号 1
    SpindleIFNo :=B#16#4,            //主軸インターフェース番号= 4
    FeedHold :=    m22.0,            //送り停止信号モーダル
    SpindleHold :=db2.dbx151.0,      //メッセージデータブロックでの主軸停止モーダル
    SpindleDir:=   m22.1);           //主軸戻り方向
```

このパラメータ設定により、最初のモードグループ、最初のチャネルおよびすべての軸に信号が送信されます。また、主軸オーバーライドは、4 番目の軸/主軸インタフェースに伝送されます。フィードホールド信号はビットメモリ 22.0 に渡され、主軸停止信号はデータブロック DB2、データビット 151.0 に渡されます。パラメータ「SpindleDir」により提供される主軸方向フィードバック信号は、追加の FC18 呼び出しの方向入力として使用できます。

14.17.27 FC25:MCP_ IFT - インターフェースへの MCP/OP 信号の伝送

機能

ブロック FC25 "MCP_IFT" (T タイプ、たとえば MCP 483 など)は、機械操作パネルから NC/PLC インタフェースにデータを転送するために使用します。

- モード
- 4 つの軸の方向キー
- WCS/MCS 切り替え
- オーバライド
- キー操作スイッチ

次の仕様は、有効なモードまたは選択された座標系に応じて、**送り速度オーバライド、軸移動キー**および **INC キー**に適用されます。

- **送り速度オーバライド**
  - 送り速度オーバライドは、選択されたチャネルのインタフェース、および軸のインタフェースに伝送されます。
  - HMI 信号「早送り用送り速度オーバライド有効」が設定されている場合、「早送りオーバライド」(DBB 5)インタフェースバイトに加えて、送り速度オーバライド信号が NC チャネルに伝送されます。(例外:スイッチ設定「ゼロ」)。「早送りオーバライド有効」もこの HMI 信号で設定します。
- **INC と軸移動キーに対する運転機能**
  - MCS が選択されると、信号は、選択された機械軸のインタフェースに伝送されます。
  - WCS が選択されると、信号は、パラメータ設定されたチャネルのジオメトリ軸インタフェースに伝送されます。

機械操作パネルの対応する LED は、関連する選択の応答から得られます。

送り速度と主軸の Start/Stop は、インタフェースに伝送されるのではなく、「FeedHold」または「SpindleHold」信号としてモーダルに出力されます。ユーザーは、これらの信号を、送りまたは主軸停止を引き起こす他の信号に接続することができます(たとえば、次のファンクションの適切な入力信号を使って、FC10:AL_MSG)。同時に、関連 LED が点灯します。

機械操作パネルが故障すると、ここから出力される信号はゼロにプリセットされます。これは、「FeedHold」と「SpindleHold」出力信号にも当てはまります。

FC25、または FC19、FC24、FC26 も、1 回の PLC 周期で複数回呼び出すことができます。この場合、この周期における最初の呼び出しが LED 表示を駆動します。さらに、パラメータ設定されたブロックのすべての動作は、最初の呼び出しで実行されます。以降の呼び出しでは、チャネルとモードグループインタフェースの処理の削減レベルだけがおこなわれます。ジオメトリ軸には、この周期における最初のブロック呼び出しにおいてのみ、方向データが提供されます。

シングルブロック処理の選択/解除は、この周期における最初の呼び出しでのみおこなうことができます。

B#16#10 によってパラメータ「ModeGroupNo」が増加している場合は、2 つ目の機械操作パネルを処理することができます。パラメータ設定時に、HHU 番号は下位ニブル(下位 4 ビット)に含まれます。

「BAGNo」= 0 または B#16#10⇒モードグループ信号が処理されません。

「ChanNo」= 0⇒チャネル信号の処理をおこないません。

フレキシブルな軸設定

機械軸番号について軸選択または方向キーの割り当てをフレキシブルにおこなうことができます。

特に 2 つのチャネルと 2 つのモードグループを使用するアプリケーションについて、同時に操作される 2 つの MCP を使用できるように、MCP ブロックによってサポートが提供されるようになりました。OB1 周期の第 2 機械操作パネルに対するブロック呼び出しは、第 1 MCP の呼び出しの後に設定してください。軸番号は、関連 MCP の軸テーブル内で、MCP ブロックのパラメータ設定済モードグループ番号でも指定されることに注意してください。

このフレキシビリティを提供するために、軸番号のテーブルが DB10 に保存されます。**1 番目の**機械操作パネル(MCP)では、テーブルはバイト 8 から始まり(シンボル名称:MCP1AxisTbl[1..22])、**2 番目の**機械操作パネル(MCP)では、バイト 32 から始まります(シンボル名:MCP2AxisTbl[1..22])。ここでは、機械軸番号をバイト単位で入力してください。軸テーブルに値 0 を入力することができます。不正な軸番号を検出するためのチェックはおこなわれません。つまり、誤った入力をおこなうと、PLC が停止することがあります。

FC25 で可能な軸数の制限は、軸テーブル内の 0 値を使用して実行されます。軸番号と制限も自動的に補正できます。方向キーを使用した軸の手動移動中に、軸番号を切り替えることはできません。互換モードでは、両 MCP で軸番号が **1 ～ 4** にプリセットされ、軸の設定数に制限されます。

注記

詳しくは、「FC19:MCP_IFM - MCP 信号のインターフェースへの伝送 (ページ 1314)」を参照してください。

ファンクションの宣言

```
FUNCTION FC25:VOID
//NAME:               MCP_IFT

VAR_INPUT
    BAGNo :           BYTE;
    ChanNo:           BYTE;
    SpindleIFNo:      BYTE;
END_VAR

VAR_OUTPUT
```

```
    FeedHold :        BOOL;
    SpindleHold :     BOOL;
END_VAR

BEGIN
END_FUNCTION
```

仮パラメータの説明

信号	種別	タイプ	値の範囲	意味
BAGNo :	I	BYTE	B#16#00 ~ B#16#0A	**第 1 MCP**:モード信号の伝送先となるモードグループの番号
			B#16#10 - B#16#1A	**第 2 MCP**:モード信号の伝送先となるモードグループの番号
ChanNo:	I	BYTE	B#16#00 ～ B#16#0A	チャネル信号の伝送先となるチャネルの番号
SpindleIFNo:	I	BYTE	B#16#00 ～ B#16#1F	主軸データの伝送先となる軸/主軸の番号(関連する機械軸の番号)
FeedHold :	O	BOOL	0 (FALSE)、1 (TRUE)	MCP からの送り停止、モーダル
SpindleHold :	O	BOOL	0 (FALSE)、1 (TRUE)	MCP からの主軸停止、モーダル

呼び出し例

このパラメータ割り当て例により、最初のモードグループ、最初のチャネルおよびすべての軸に信号が送信されます。主軸オーバライドは、4 番目の軸/主軸に伝送されます。送り停止はビットメモリ 22.0 に、主軸停止はデータブロック DB2、DBX151.0 に送信されます。

```
CALL FC25 (                          //機械操作パネル T 型
                                     //インタフェースへの信号
  BAGNo :=         B#16#1,           //モードグループ番号 1
  ChanNo :=        B#16#1,           //チャネル番号 1
  SpindleIFNo :=   B#16#4,           //主軸インターフェース番号= 4
  FeedHold :=      m22.0,            //送り停止信号モーダル
  SpindleHold :=   db2.dbx151.0); //メッセージデータブロックでの主軸停止モーダル
```

14.17.28 FC26:HPU_MCP - インターフェースへの HT 8 信号の伝送

ファンクションの宣言

```
FUNCTION FC26:VOID
//NAME:          HPU_MCP

VAR_INPUT
    BAGNo :      BYTE;
    ChanNo:      BYTE;
END_VAR

BEGIN
END_FUNCTION
```

仮パラメータの説明

信号	種別	タイプ	値の範囲	意味
BAGNo :	I	BYTE	B#16#00 ～ B#16#0A	第 1 MCP:モード信号の伝送先となるモードグループの番号
			B#16#10 - B#16#1A	第 2 MCP:モード信号の伝送先となるモードグループの番号
ChanNo:	I	BYTE	B#16#00 ～ B#16#0A	チャネル信号の伝送先となるチャネルの番号

呼び出し例

NC の最初の MCP、最初のモードグループ、および最初のチャネルの FC26 を呼び出します。

```
CALL FC26 (                          //HT 8 の機械操作パネル
           BAGNo :=    B#16#01,      //第 1MCP、第 1 モードグループ
           ChanNo :=   B#16#01);     //チャネル 1
```

NC の 2 番目の MCP、2 番目のモードグループ、および 3 番目のチャネルの FC26 を呼び出します。

```
CALL FC26 (                          //HT 8 の機械操作パネル
           BAGNo :=    B#16#12,      //第 2MCP、第 2 モードグループ
           ChanNo :=   B#16#03 );    //チャネル 3
```

機能の一般的な説明

ファンクション FC26「HPU_MC」(携帯ユニットの HT 8 機械操作パネル信号)は、次の機能の HT 8 用信号を、ファンクションブロック FB1 でパラメータ設定された入力/出力データ領域(パラメータ:MCPxIn と MCPxOut)と NC/PLC インタフェース間で伝送します。

- モード
- 運転機能 INC
- 座標系 WCS または MCS
- 軸移動キー
- 軸選択
- 送り速度オーバライド
- 早送りオーバライド
- キースイッチ情報

注記

HT 8 または HMI、およびその両方によるモード切り替え

ファンクション FC2「GP_HP 基本プログラム、サイクリックセクション」は、HT 8 と HMI の MCP の代替選択を実行できるように、モード切り替えの信号を伝送します。パラメータ「MMCToIF」= FALSE により、ファンクションブロック FB1 で、NC/PLC インターフェースへの HMI 信号の伝送を無効にすることもできます。

有効な軸

HT 8 を使用すれば、最大 6 つの軸を同時にアドレス指定できます。軸の選択は、PLC ユーザープログラムで、ユーザー/工作機械メーカがおこなってください。

フレキシブルな軸設定

ファンクション FC26 を使用すると、移動キーまたは軸選択への機械軸の割り当てを柔軟に実行できます。DB10 では、このために次の 2 つのテーブルを使用できます。

- 機械軸テーブル、第 1MCP:DB10.DBB8 ～ DBB13 (機械軸番号のテーブル)
  シンボル名称:MCP1AxisTbl[1..22]
- 機械軸テーブル、第 2MCP:DB10.DBB32 ～ DBB37 (機械軸番号のテーブル)
  シンボル名称:MCP2AxisTbl[1..22]

このテーブルでは、有効な機械軸の軸番号 n (ここで、n = 1、2、...)はバイト毎に入力してください。値 0 は、未使用テーブル位置に入力してください。

テーブル長さは、FC26 に次のように指定できます。

- 第 1 MCP:DB10.DBB30 (機械軸テーブルの上限)
- 第 2 MCP:DB10.DBB54 (機械軸テーブルの上限)

たとえば、値 4 は、FC26 が最初の 4 つのテーブルエントリまたは機械軸だけを考慮することを意味します。FC26 の最大値は 6 です。値 0、または 6 より大きい値の場合は、自動的に最大値になります。

注記

次の制限に注意してください。

- 機械軸番号が許容されるかどうかのチェックは実行されません。機械軸番号が無効な場合、PLC が停止することがあります。
- 機械軸番号は自動的に変更できます。移動キーを使用して既に機械軸が移動している場合、このテーブルに書き込むことはできません。

有効な座標系に応じた移動キー信号の伝送

6 軸に対応した移動キー信号は、HT 8 入力データ内に次のように配置されています。

- EB n + 2、ビット 0 ～ビット 5 (正の移動方向)
- EB n + 3、ビット 0 ～ビット 5 (負の移動方向)

座標系の切り替えは、次の入力信号により実行されます。

- EB n + 0、ビット 0 (MCS/WCS)

入力信号が、エッジトリガフラグを使用して FC26 で評価されます。有効な座標系は、次の出力信号により示されます。

- AB n + 0、ビット 0 (MCS/WCKS)、0 = MCS、1 = WCS

MCS が有効な場合、軸 1 ～ 6 の移動キー信号は、機械軸テーブル(DB10.DBB8 ～ DBB13 または DBB32 ～ DBB37)で指定された軸の軸別インタフェース(DB31、... .DBX4.6 および DBX4.7 (移動キー+/-))で伝送されます。

WCS が有効な場合、機械軸テーブルの軸 1 ～ 3 はジオメトリ軸であると想定されます。このため、移動キー信号は

- 1 ～ 3 の各軸では(EB n + 2 / 3、ビット 0 ～ビット 2)、パラメータ「ChanNo」で指定されたチャネルの DB21、... .DBB12 + (n * 4)、(ここで、n = 0、1、2)、ビット 6 とビット 7 (移動キー+/-)内のジオメトリ軸のインターフェースで伝送されます。
  チャネルのジオメトリ軸 1、2 および 3 への軸 1、2 および 3 の移動キー信号の割り当ては、固定なので変更できません。
- o4 ～ 6 の各軸では(EB n + 2 / 3, ビット 3 - ビット 5)機械軸テーブル(DB10.DBB11 ～ DBB13 または DBB35 ～ DBB37)に入力された軸 4 ～ 6 の軸別インタフェース (DB31、 ... .DBX4.6 および DBX4.7 (起動キー+/-))に伝送されます。

WCS での機械軸の移動なし

WCS が有効な場合(AB n + 0、ビット 0 = 1)、機械軸の移動をロックできます。このため、PLC ユーザープログラムで、次の出力信号を設定してください。

- AB n + 3、ビット 7 = 1 (WCS の場合:機械軸なし)
  FC26 の必要条件、機械軸の移動キー信号を伝送しないこと。機械軸テーブルの軸 1 ～ 3 の移動キー信号は、指定チャネルのジオメトリ軸 1 ～ 3 に伝送されます。機械軸テーブルの軸 4 ～ 6 の移動キー信号は伝送されません。
- AB n + 2、ビット 6 (軸 7 ～ n 選択済み)
  FC26 の必要条件は、移動キー信号を伝送していないことです。これは、機械軸テーブルの軸 1 ～ 6 が切り替えられるためです。したがって、軸 1 ～ 3 はジオメトリ軸ではなく、機械軸です。

送り速度オーバライド

HT 8 オーバーライドスイッチの値は、プログラム指令チャネル(パラメータ:「ChanNo」)のチャネル別インターフェース DB21、... .DBB4(送り速度オーバライド)およびテーブル DB10.DBB8 ～ DBB13 (機械軸番号)でプログラム指令軸の軸別インタフェース DB31、... .DBB0 (送り速度オーバライド)における送り速度オーバライドとして伝送されます。

早送りオーバライド

プログラム指令チャネル(パラメータ:"ChanNo")信号 DB21, ...DBX25.3 = 1 (早送り用の送り速度オーバライド)が設定されている場合、DB21, ...DBB5 (早送りオーバライド)および信号 DB21, ...DBX6.6 = 1 (早送りオーバライド有効)でチャネル別インタフェースに対する早送りオーバライドも設定されているために、HT 8 オーバーライド切り替え値が送信されます。

運転機能 INC

運転機能 INC の HT 8 信号の伝送は、有効な座標系が MCS か WCS により違ってきます。

- 有効な座標系:MCS
  選択した運転機能 INC は、6 軸すべてに対して、DB10.DBB8 ～ DBB13 (機械軸番号)のテーブルにプログラム指令された軸の DB31, ...DBX5.0 ～ DBX5.5 (運転機能)の軸別インタフェースに伝送されます。
- 有効な座標系:WCS
  軸 1 ～ 3 の場合は、運転機能 INC の信号は、プログラム指令されたチャネル(パラメータ:「ChanNo」)の DB21、... .DBX13.0 ～ DBX13.5 (運転機能)内のチャネル別インタフェースで伝送されます。「ChanNo」)
  軸 4 ～ 6 の場合は、運転機能 INC の信号は、DB10.DBB11 ～ DBB13(機械軸番号)のテーブルにプログラム指令された軸の DB31, ...DBX5.0 ～ DBX5.5 (運転機能)の軸別インタフェースに伝送されます。

運転機能 INC の選択信号は、モードグループ別インターフェース DB11.DBB 2 + (n * 20)、ビット 0 ～ビット 5 (ここで、n = 0、1、2、...)で伝送されます。FC26 は、DB10.DBX57.0 (モードグループ領域で INC 入力有効)の起動後に INC 運転機能のモードグループインタフェースの起動について NC に通知します。

ハンドルの選択

ハンドル選択信号は、HMI により解析され、機械軸またはジオメトリ軸の対応する NC/PLC インタフェース信号に伝送されます。

- ジオメトリ軸:DB21、... DBB 12 + (n * 4)、ビット 0 ～ビット 2 (ここで、n = 0、1、2)
- 機械軸:DB31、... .DBX4.0 ～ DBX4.2

必要条件:FB1 パラメータ:「HWheelMMC」= TRUE

1 回の PLC 周期での複数呼び出し

FC26 は、1 回の PLC 周期で複数回呼び出すことができます。PLC 周期での最初の呼び出しにより:

- パラメータ設定されたブロックの動作がすべて実行されます
- LED 信号が、出力領域に書き込まれます
- WCS が選択されている場合、ジオメトリ軸の移動キー信号が書き込まれます
- 個々のブロックの選択と解除用の信号が処理されます

より頻繁に FC26 を呼び出すと、チャネルとモードグループインターフェースの処理速度が低下します。

2 つの MCP の処理

PLC プログラム(オーガニゼーションブロック OB1)のサイクリックシーケンスで、2 つの MCP にファンクション FC26 を 2 回呼び出す場合は、最初の MCP の呼び出し後に、2 番目の MCP の呼び出しを実行してください。

注記

2 つの MCP から軸を移動できる場合、相互インタロックは、ユーザー(工作機械メーカ)が実行します。

HT 8 の MCP の故障

HT 8 の MCP が故障すると、入力信号がすべて値 0 に設定されます。

14.17.28.1 HT8 の NC/PLC インタフェース信号の一覧

運転モードと運転機能

出典：MCP	保存先:プログラムされたモードグループ(パラメータ BAGNo) モードグループに対する表示
AUTOMATIC	DB11.DBX0.0
MDI	DB11.DBX0.1
JOG	DB11.DBX0.2
REPOS	DB11.DBX1.1

出典：MCP	保存先:プログラムされたモードグループ(パラメータ BAGNo) モードグループに対する表示
REF	DB11.DBX1.2
TEACH IN	DB11.DBX1.0
INC 1 ... 10 000, INC Var.	DB11.DBX2.0 - DBX 2.5

移動キーと早送りオーバライド

出典：MCP	保存先:プログラムされたチャネル(パラメータ:ChanNo)
移動キー+	DB21、... .DBX12.7
移動キー-	DB21、... .DBX12.6
早送りオーバーライド	DB21、... .DBX12.5
移動キー+	DB21、... .DBX16.7
移動キー-	DB21、... .DBX16.6
早送りオーバーライド	DB21、... .DBX16.5
移動キー+	DB21、... .DBX20.7
移動キー-	DB21、... .DBX20.6
早送りオーバーライド	DB21、... .DBX20.5

出典：MCP	保存先:DB 10、DBB 8 ～ 13 (第 1MCP)または DBB32 ～ 37 (第 2MCP)内のテーブルに対応するプログラムされた軸
移動キー+	DB31, ...DBX4.7
移動キー-	DB31, ...DBX4.6
早送りオーバーライド	DB31, ...DBX4.5

オーバライド

出典：MCP	保存先:プログラムされたチャネル(パラメータ:ChanNo)
送り速度オーバライド	DB21、... .DBB4

出典：MCP	保存先:DB 10、DBB 8 ～ 13 (第 1MCP)または DBB32 ～ 37 (第 2MCP)内のテーブルに対応するプログラムされた軸
送り速度オーバライド	DB31、... .DBB0

チャネル信号

出典：MCP	保存先:プログラムされたチャネル(パラメータ:ChanNo)
NC スタート	DB21、... .DBX7.1
NC ストップ	DB21、... .DBX7.3
RESET	DB21、... .DBX7.7
シングルブロック	DB21、... .DBX0.4

14.17.28.2 HT8 の NC/PLC インタフェース信号の一覧

運転モードと運転機能

保存先:MCP	出典：インターフェース DB (パラメータ BAGNo) モードグループに対する表示
AUTOMATIC	DB11.DBX6.0
MDI	DB11.DBX6.1
JOG	DB11.DBX6.2
REPOS	DB11.DBX7.1
REF	DB11.DBX7.2
TEACH IN	DB11.DBX7.0

14.17.29 FC1005:AG_SEND - Ethernet CP へのデータの伝送

機能

AG_SEND ファンクションブロックは、Ethernet CP への設定済み接続を介して伝送用のデータを伝送します。AG_RECV ファンクションブロックとの併用により、内蔵「CP

840D sl」を介して、別のステーションとのデータ交換を確立できます。このステーションは、STEP 7、「NetPro」で設定してください。

基本プログラムでは、この機能をファンクションブロック FC1005 として使用できます。これは、「SIMATIC_NET_CP」ライブラリのファンクションブロック FC5 と、ほぼ同じです。

TCP、UDP、および ISO-on-TCP の各プロトコルがサポートされています。

注記

SINUMERIK CP および UDP または ISO-on-TCP プロトコルでは、伝送可能なデータ量が減少

SINUMERIK CP の場合、ファンクションブロック FC1005 および UDP または ISO-on-TCP プロトコル使用時に伝送可能なデータは **240** バイトのみです。

仮パラメータの説明

信号	種別	タイプ	数値の範囲	説明
実際	I	BOOL	0 (FALSE)、1 (TRUE)	ジョブの開始 1)
ID:	I	INT	1, 2, 3 ...16	接続 ID
LADDR	I	WORD	-	モジュール開始アドレス 2)
送信:	I	ANY	-	アドレスと長さを指定します。データ領域のアドレスは、以下のいずれかを指します。 ● ビットメモリアドレス領域 ● データブロック領域
LEN	I	INT	TCP1, 2, ... 8192 UDP1, 2, ... **240** ISO-on-TCP:1, 2. ... **240**	伝送されるバイト数
Done :	O	BOOL	0 (FALSE)、1 (TRUE)	0:ジョブを実行中 1:ジョブが実行済み
ERROR:	O	BOOL	0 (FALSE)、1 (TRUE)	0:エラーなし 1:異常が有効

信号	種別	タイプ	数値の範囲	説明
Status:	O	WORD	-	ステータス表示
1) パラメータ ACT は、以下が該当するまで、TRUE にしてください。applies:(DONE == 1) OR (ERROR == 1) 2) SINUMERIK 840D sl の場合:パラメータ LADDR := **W#16#8110**				

取扱説明書

モジュールの詳しい説明は、以下の資料にあります。

- SINUMERIK ユーザーインターフェース:オンラインヘルプ
- 『SIMATIC プログラミングマニュアル』:SIMATIC NET S7-CP のプログラムブロック章:「Industrial Ethernet 用プログラムブロック」>「オープン通信サービス用プログラムブロック(SEND/RECEIVE インターフェース)」>「AG_SEND/AG_LSEND/AG_SSEND」

14.17.30 FC1006:AG_RECV - Ethernet CP からのデータの受信

機能

ファンクションブロック AG_RECV は、設定された接続を介して Ethernet CP から伝送されたデータを受信します。AG_SEND ファンクションブロックとの併用により、内蔵「CP 840D sl」を介して、別のステーションとのデータ交換を確立できます。このステーションは、STEP 7、「NetPro」で設定してください。

基本プログラムでは、この機能をファンクションブロック FC1006 として使用できます。これは、「SIMATIC_NET_CP」ライブラリのファンクションブロック FC6 と、ほぼ同じです。

TCP、UDP、および ISO-on-TCP の各プロトコルがサポートされています。

注記

SINUMERIK CP および UDP または ISO-on-TCP プロトコルでは、伝送可能なデータ量が減少

SINUMERIK CP の場合、ファンクションブロック FC1006 および UDP または ISO-on-TCP プロトコル使用時に伝送可能なデータは **240** バイトのみです。

仮パラメータ

信号	タイプ	タイプ	数値の範囲	説明
ID:	I	INT	1, 2 ...16	接続 ID
LADDR	I	WORD	-	モジュール開始アドレス 1)
RECV	I	ANY	-	アドレスと長さを指定します。データ領域のアドレスは、以下のいずれかを指します。 ● ビットメモリアドレス領域 ● データブロック領域
NDR :	O	BOOL	0 (FALSE)、1 (TRUE)	0:ジョブを実行中 1:新規データの受信
ERROR:	O	BOOL	0 (FALSE)、1 (TRUE)	0:エラーなし 1:異常が有効
Status:	O	WORD	-	ステータス表示
LEN	O	INT	TCP1, 2, ... 8192 UDP1, 2, ... **240** ISO-on-TCP:1, 2. ... **240**	伝送されるバイト数
1) SINUMERIK 840D sl の場合:パラメータ LADDR := **W#16#8110**				

取扱説明書

モジュールの詳しい説明は、以下の資料にあります。

- SINUMERIK ユーザーインターフェース:オンラインヘルプ
- 『SIMATIC プログラミングマニュアル』:SIMATIC NET S7-CP のプログラムブロック章:「Industrial Ethernet 用プログラムブロック」>「オープン通信サービス用プログラムブロック(SEND/RECEIVE インターフェース)」>「AG_RECV/AG_LRECV/AG_SRECV」

14.18 信号/データの説明

14.18.1 NCK/PLC、HMI/PLC、MCP/PLC の各インタフェース信号

参照先

すべてのインターフェース信号の概要については、以下の資料を参照してください。

『NC 変数とインタフェース信号』リストマニュアル

基本 PLC プログラムで評価され、選別された形でユーザーインターフェースに伝送される NC インターフェース信号については、以降の各セクションを参照してください。

14.18.2 デコードされた M 信号

パートプログラム、ASUB またはシンクロナイズドアクションでプログラムされた M 機能は、チャネル別に以下のように NC から PLC に伝送されます。

- チャネル 1 からの M 機能:DB21
- チャネル 2 からの M 機能:DB22
- ...

DB21、...	意味
DBX194.0 - .7	M 機能 M0 ... M7
DBX195.0 - .7	M 機能 M8 ... M15
DBX196.0 - .7	M 機能 M16 ... M23
DBX197.0 - .7	M 機能 M24 ... M31
DBX198.0 - .7	M 機能 M32 ... M39
DBX199.0 - .7	M 機能 M40 ... M47
DBX200.0 - .7	M 機能 M48 ... M55

DB21、...	意味
DBX201.0 - .7	M 機能 M56 ... M63
DBX202.0 - .7	M 機能 M64 ... M71
DBX203.0 - .7	M 機能 M72 ... M79
DBX204.0 - .7	M 機能 M80 ... M87
DBX205.0 - .7	M 機能 M88 ... M95
DBX206.0 - .3	M 機能 M96 ... M99

関連する M 機能の信号は、NC/PLC インターフェースの 1 回以上の PLC 周期で未処置のままです。

注記

以下の主軸別 M 機能は、NC/PLC インターフェースに伝送されません。M3、M4、M5、および M70。

M02/M30 パートプログラム終了

M02/M30 M 機能の出力では、パートプログラムが完了したことが確実には通知されません。M02/M30 M 機能の出力は、この場合のパートプログラムの実際の終了とは無関係な非同期サブプログラム(ASUB)またはシンクロナイズドアクションから発生する可能性があります。チャネルでパートプログラムの終了を明確に検出するには、次のインターフェース信号を使用してください。

DB21、... .DBX35.7 (チャネル状態:リセット)

14.18.3 G 命令

パートプログラム、ASUB、またはシンクロナイズドアクションでプログラム指令された G 命令は、以下のように、NC から PLC にチャネル別に伝送されます。

- チャネル 1 の G 命令:DB21
- チャネル 2 の G 命令:DB22
- ...

DB21、...	意味
DBB208	G グループ 1 の有効な G 命令
DBB209	G グループ 2 の有効な G 命令
DBB210	G グループ 3 の有効な G 命令
...	...
DBB271	G グループ 64 の有効な G 命令

関連する G 命令は、NC/PLC インターフェースの 1 回以上の PLC 周期で未処置のままです。

DB21、...内のアドレス

DB21、...内の G グループの DBBx アドレスは、以下のように計算されます。

バイト番号= 208 +「G グループ数」- 1

電源投入後の初期設定

電源投入後に、値 0、つまり、有効な未定義の G グループが、すべての G グループについて伝送されます。

パートプログラムの終了または中止

パートプログラムが終了または中止した後は、最後の有効な G 命令が保持されます。

NC スタート

NC-START の後は、以下のマシンデータに指定された 8 つの G グループの値が、以下のマシンデータを介して設定される初期設定およびパートプログラムでプログラム指令された値に応じて、上書きされます。
MD22510 $MC_GCODE_GROUPS_TO_PLC

参照先

すべての G 命令の概要は、以下の資料にあります。

プログラミングマニュアル 基本編、「テーブル」の章>「G 命令」

14.18.4 DB 2 のメッセージ信号

DB2 を使用すると、個々の信号のメッセージが操作パネルに出力されます。信号は、事前定義されたグループに分割されます。メッセージが表示、消去または確認応答されると、メッセージ番号列に入力された番号が HMI に伝送されます。各メッセージ番号に対して、テキストを HMI に格納できます。

参照先:

- 『リストマニュアル、NC 変数とインターフェース信号』、「インターフェース信号-概要」の章>「PLC アラーム/メッセージ」
- 『試運転マニュアル』、「アラームおよびメッセージテキスト」の章

注記

ユーザー領域の数は、FB1 を使用してパラメータ設定できます。
設定を修正した後(FB1:MsgUser)は、DB2/DB3 を削除してください。

14.19 STEP 7 のプログラミングに関する注意事項

以下の章の手順は、STEP 7 の複雑なプロセスおよびファンクションのプログラミングを簡素化するためのものです。

- データのコピー (ページ 1358)
- データタイプ ANY および POINTER (ページ 1358)
- マルチインスタンス DB (ページ 1364)
- 文字列 (ページ 1365)
- データブロック構造のオフセットアドレスの決定 (ページ 1366)
- FB 呼び出し (ページ 1367)

参照先

データタイプ ANY および POINTER の構造に関する一般事項については、以下を参照してください。

『SIMATIC STEP 7 マニュアル』、「ユーザープログラムの設計」>「CPU のレジスタとデータの保存」

14.19.1 データのコピー

コピーの種類

1 つの DB から別の DB にデータを高速コピーするために推奨されることは、

- **大量のデータの場合は**システムファンクション SFC BLKMOV または SFC FILL を使用するということです。その場合、高速コピーが行われるからです。
- 以降のルーチンは、**少量のデータ**用です。SFC に ANY パラメータを与えることにより、消費する時間が追加されるからです。

例

```
                                    // DB xx.[AR1]がコピー元です
                                    // DI yy.[AR2]がコピー先です
AUF         DB 100;                 //コピー元 DB
LAR1        P#20.0;                 //データバイト 20 のコピー元開始アドレス
AUF         DI 101;                 //コピー先 DB
LAR2        P#50.0;                 //データバイト 50 のコピー先開始アドレス
                                    //AR1、AR2、DB、DI は事前にロード済みです
L           4;                      //8 バイトの伝送
M001:
L           DBW [AR1,P#0.0];        //ワードに応じたコピー
T           DIW [AR2,P#0.0];
+AR1        P#2.0;
+AR2        P#2.0;
TAK;
LOOP        M001;
```

14.19.2 ANY および POINTER

以下のプログラミング例は、プログラミングの仕組みを示しています。 これらは、FC または FB 内でのデータタイプ「POINTER」または「ANY」による入力/出力および遷移変数(VAR_INPUT、VAR_OUTPUT、VAR_IN_OUT)へのアクセス方法を示します。 このアクセス操作は、プログラミングの部分的なシンボル方法を使用できる方法で記述されます。

14.19.2.1 FC での POINTER および ANY の使用

機能

FC99 には、POINTER または ANY として定義される入力パラメータがあります。

この例は、それを介して POINTER または ANY のサブコンポーネントにアクセスできる本体プログラムを示しています。この場合、POINTER または ANY によりパラメータ設定される DB が開かれ、アドレスオフセットは領域間ポインタとしてアドレスレジスタ AR1 に格納されます。したがって、POINTER、ANY を介してアドレス指定される変数のデータ要素(一般に構造および配列)にアクセスできます。

このアクセス操作は、この例の関連するプログラムシーケンスの最後に記述されています。データタイプ ANY を使用すると、データタイプおよび関係のある要素の数に基づいて変数にアクセスするときに、チェックを実行したり分岐したりすることもできます。

例

```
FUNCTION FC99:VOID
VAR_INPUT
   Row :BYTE;
   Convert :BOOL;                    //数値変換を起動
   Addr:POINTER;                     //変数の指定
   Addr1 :ANY ;
END_VAR
VAR_TEMP
   dbchr :WORD;
   番号:WORD;
   type :BYTE;
END_VAR
BEGIN
NETWORK
TITLE =
                                     //ポインタ
   L          P##Addr;
   LAR1 ;                            //ポインタの取り出し
   L          W [AR1,P#0.0];         //DB 番号の取り出し
   T          #dbchr;
   L          D [AR1,P#2.0];         //ポインタのオフセット部分
LAR1 ;
AUF DB [#dbchr];                     //変数の DB のオープン
L B [AR1,P#40.0];                    //ポインタを使用したバイト値の取り出し
```

```
                                        //アドレスオフセット 40 による
                                        //ANY
   L            P##Addr1;
   LAR1 ;                               //ANY の取り出し
   L            B [AR1,P#1.0];          //タイプの取り出し
   T            #typ;
   L            W [AR1,P#2.0];          //量の取り出し
   T            #Amount;
   L            W [AR1,P#4.0];          //DB 番号の取り出し
   T            #dbchr;
   L            D [AR1,P#6.0];          //ポインタのオフセット部分
LAR1 ;
   AUF          DB [#dbchr];            //変数の DB のオープン
   L            B [AR1,P#0.0];          //ANY を使用したバイト値の取り出し
```

14.19.2.2 FB での POINTER および ANY の使用

機能

FB99 には、POINTER または ANY として定義される入力パラメータがあります。

この例は、それを介して POINTER または ANY のサブコンポーネントにアクセスできる本体プログラムを示しています。この場合、POINTER または ANY によりパラメータ設定される DB が開かれ、アドレスオフセットは領域間ポインタとしてアドレスレジスタ AR1 に格納されます。したがって、POINTER、ANY を介してアドレス指定される変数のデータ要素(一般に構造および配列)にアクセスできます。

このアクセス操作は、この例の関連するプログラムシーケンスの最後に記述されています。データタイプ ANY を使用すると、データタイプおよび関係のある要素の数に基づいて変数にアクセスするときに、チェックを実行したり分岐したりすることもできます。

例

```
FUNCTIONBLOCK FB99
VAR_INPUT
   Row :BYTE;
   Convert :BOOL;                       //数値変換を起動
   Addr:POINTER;                        //変数の指定
   Addr1 :ANY ;
END_VAR
VAR_TEMP
```

```
   dbchr :WORD;
   番号:WORD;
   type :BYTE;
END_VAR
BEGIN
NETWORK
TITLE =
                                          //ポインタ
   L              P##Addr;
   LAR1 ;                                 //インスタンス DB からのポインタの取り出し
   L              DIW [AR1, P#0.0];       //DB 番号の取り出し
   T              #dbchr;
   L              DID [AR1,P#2.0];        //ポインタのオフセット部分
   LAR1 ;
   AUF            DB [#dbchr];            //変数の DB のオープン
   L              B [AR1,P#40.0];         //ポインタを使用したバイト値の取り出し
                                          //アドレスオフセット 40 による
                                          //ANY
   L              P##Addr1;
   LAR1 ;                                 //インスタンス DB からの ANY の取り出し
   L              DIB [AR1,P#1.0];        //タイプの取り出し
   T              #typ;
   L              DIW [AR1,P#2.0];        //量の取り出し
   T              #Amount;
   L              DIW [AR1,P#4.0];        //DB 番号の取り出し
   T              #dbchr;
   L              DID [AR1,P#6.0];        //ポインタのオフセット部分
   LAR1 ;
   AUF            DB [#dbchr];            //変数の DB のオープン
   L              B [AR1,P#0.0];          //ANY を使用したバイト値の取り出し
```

14.19.2.3 FC または FB への伝送用 POINTER または ANY 変数

POINTER または ANY 変数

STEP 7 のバージョン 1 以降では、VAR_TEMP にポインタまたは ANY を定義できます。

次の 2 つの例では、ANY の提供方法を示しています。

例 1:選択リストを介した別の FB (FC)への ANY パラメータの伝送

複数の ANY パラメータが、FB (FC)に定義されます。別の FB (FC)に伝送するために、選択リストから特定の ANY パラメータを選択してください。これは、VAR_TEMP のANY によってのみ行うことができます。Addr1 ～ Addr4 を選択するために、パラメータ「WhichAny」に 1 ～ 4 を設定できます。

注記

アドレスレジスタ AR2 は、このブロックで使用されます。ただし、このアドレスレジスタ AR2 は、マルチインスタンス DB でも使用されます。このため、この FB はマルチインスタンス DB として宣言**しないで**ください。

```
FUNCTIONBLOCK FB100
CODE_VERSION1                          //STEP 7 バージョン 2 からの起動
                                       //マルチインスタンス DB を無効にするため
VAR_INPUT
WhichAny :INT;
   Addr1 :ANY ;                        //事前定義された順序の遵守
   Addr2 :ANY ;
   Addr3 :ANY ;
   Addr4 :ANY ;
END_VAR
VAR_TEMP
   dbchr :WORD;
   番号:WORD;
   type :BYTE;
   Temp_addr :ANY ;
END_VAR
BEGIN
NETWORK
TITLE =
L        WhichAny;
DEC 1;
L        P#10.0;                       //ANY ごとに 10 バイト
*I;
LAR2;
L        P##Addr1;
+AR2;                                  //ANY 開始アドレスの追加
L        P##Temp_addr;
LAR1 ;                                 //VAR_TEMP からのポインタの取り出し
L        DID [AR2,P#0.0];              //VAR_TEM へのポインタ値の伝送
```

```
T        LD [AR1,P#0.0];
L        DID [AR2,P#4.0];
T        LD [AR1,P#4.0];
L        DIW [AR2,P#8.0];
T        LW [AR1,P#8.0];

CALL FB101, DB100
         (ANYPAR := #Temp_addr);   //ANYPAR はデータタイプ ANY
```

例 2 :別の FB (FC)への事前に作成された ANY パラメータの伝送

すでにコンパイルされた ANY パラメータが、別の FB (FC)に伝送されます。これは、VAR_TEMP に格納された ANY によってのみ行うことができます。

```
FUNCTIONBLOCK FB100
VAR_INPUT
   DBNumber:INT;
   DBOffset :INT;
   データタイプ:INT;
   番号:INT;
END_VAR
VAR_TEMP
   dbchr :WORD;
   Temp_addr :ANY ;
END_VAR
BEGIN
NETWORK
TITLE =
L        P##Temp_addr;
LAR1 ;                             //VAR_TEMP からのポインタの取り出し
L        B#16#10;                  //ANY 識別子
T        LB [AR1,P#0.0];
L        Data type;
T        LB [AR1,P#1.0];
L        Amount;
T        LW [AR1,P#2.0];
L        DBNumber;
T        LW [AR1,P#4.0];
L        DBOffset;
SLD 3;                             //オフセットはビットオフセット
T        LD [AR1,P#6.0];
```

```
CALL FB101, DB100
         (ANYPAR := #Temp_addr);   //ANYPAR はデータタイプ ANY
```

14.19.3 マルチインスタンス DB

機能

STEP 7 のバージョン 2 以降では、マルチインスタンスが可能な、すなわちマルチインスタンス DB を持つ FB を提供できます。マルチインスタンス DB の主な特性は、データモジュールを FB のさまざまなインスタンスに使用できることです(STEP 7 取扱説明書を参照)。DB の数量構造は、これによって最適化できます。

マルチインスタンス DB は、FB の実行時間とコードサイズを増大させるので、実際に使用されるときのみ有効にしてください。

注記

ポインタおよびアドレスレジスタを使用するマルチインスタンスが可能な FB で、複雑なプログラムを実行する場合、プログラマが一定の規則を守ることが重要です。

マルチインスタンス DB を使用すると、変数の開始アドレス(VAR_INPUT、VAR_OUTPUT、VAR_IN_OUT、VAR)は、DI データブロックレジスタおよびアドレスレジスタ AR2 とともに伝送されます。マルチインスタンスが可能な FB で変数にアクセスすると、コンパイラはアドレスレジスタ AR2 を介して単独でアクセス操作を制御します。ただし、複雑なプログラムセクションで同じ FB のアドレスレジスタも処理しなければならない場合(データのコピーなど)、レジスタが変更される前に AR2 の前の内容を保存してください。インスタンス変数(VAR_INPUT、VAR_OUTPUT、VAR_IN_OUT、VAR)にアクセスする前に、AR2 の内容を元の状態に戻してください。インスタンスのAR2 レジスタは、通常ローカル変数(VAR_TEMP)に保存してください。

命令「ポインタをインスタンス変数にロード」は、インスタンスデータの開始からのポインタ値を返します。ポインタを介してこの変数にアクセスできるようにするには、AR2 に格納されたオフセットを追加してください。

例

```
FUNCTION_BLOCK FB99
VAR_INPUT
   varin:INT;
END_VAR
var
```

```
   variable1:ARRAY[[0..9]
   of INT;
   variable2:INT;
END_VAR
BEGIN
L           P##variable1;     //ARRAY の開始時のポインタ
                              //値 8500  0010 はアキュムレータにあり
                              //領域間ポインタは AR2 ポインタにあります。
                              //1 つが領域をまたいで機能する場合、
                              //これらの 2 つのポインタを追加するとき、
                              //領域は無効化にしてください。
AD          DW#16#00FF_FFFF, //領域のスキップ
LAR1                          //AR1 へのロード
TAR2;
+AR1 AR2;                     //追加される AR2 インスタンスオフセット
                              //これで変数 1 の配列に間接的に
                              //AR1 を介してアクセスできます。
L           DIW [AR1, P#0.0];//最初の要素へのアクセスなど
END_FUNCTION_BLOCK
```

14.19.4 文字列

STRING データタイプは、基本プログラムの特定のサービスに必要です。 このため、文字列構造に関するいくつかの追加情報とパラメータ割り当てに対する一般的な処理手順が次に示されています。

文字列の構造

STRING タイプのデータは、一般にデータブロックに格納(定義)されます。 文字列の定義には、次の 2 つの方法があります。

1. 単にデータタイプ STRING を変数に割り当てます。 STEP7 コンパイラにより、自動的に 254 文字の長さが生成されます。
2. データタイプ STRING は、角かっこ内の文字列長とともに変数に割り当てられます([32] など)。 この方法を使用すると、STEP7 コンパイラにより、入力に対応する文字列長が生成されます。

STRING データタイプの変数に対しては、常に定義により事前に記述されたバイトより 2 バイト多く格納されます。 STEP 7 コンパイラにより、最初のバイトには可能な最大文字数が格納されます。 2 番目のバイトには、実際に使用される文字数が含まれます。通常、割り当てられる文字列の有効な長さが、STEP 7 コンパイラにより格納されます。文字列(文字ごとに 1 バイト)は、3 番目のバイトから先に格納されます。

文字列パラメータは、通常 POINTER または ANY によって、基本プログラムのブロックに割り当てられます。 一般にこのような割り当ては、シンボルプログラミング方法を使用して行ってください。 パラメータ設定された文字列を含むデータブロックは、シンボルリストに格納してください。 基本プログラムブロックへの割り当ては、シンボルデータブロック名とその後に続くピリオドおよび文字列変数のシンボル名によって行われます。

14.19.5 データブロック構造のオフセットアドレスの決定

機能

頻繁に発生するもう 1 つの処理は、構造化された DB 内のオフセットアドレスのシンボル決定です。たとえば、ARRAY または STRUCTURE は、DB 内のどこかに格納されます。開始アドレスによりアドレスレジスタをシンボルを使用してロードした後は、アドレスレジスタを介して ARRAY または STRUCTURE の個々の要素にアクセスすることになります。アドレスレジスタをシンボルを使用してロードする 1 つの方法は、入力パラメータがポインタである FC を使用することです。その場合、ARRAY または STRUCTURE のアドレスは、プログラムでこの FC の入力パラメータにシンボルを使用して割り当てられます。FC のプログラムコードにより、入力パラメータからオフセットアドレスが決定され、アドレスレジスタ(AR1)のオフセットアドレスが呼び出しファンクションに渡されます。このようにシンボルを使用したアドレス指定は、間接アクセスでも可能です。

例

```
FUNCTION FC99:VOID
VAR_INPUT
    Addr:POINTER;              //変数の指定
END_VAR
BEGIN
NETWORK
TITLE =
L         P##Addr;
LAR1 ;                         //Addr からのポインタの取り出し
L         D [AR1,P#2.0];       //変数のポインタのオフセット部分
LAR1 ;
END_FUNCTION
```

14.19.6 FB 呼び出し

機能

実行速度を最適化するには、ブロック FB2、3、4、5、および 7 など、すべてのファンクションブロック呼び出しを多くの内部パラメータと共に呼び出すと、役に立ちます。これらは、関連するインスタンスパラメータと共に、起動時に基本プログラムにより提供されます。起動(OB100)時には、パラメータの事前割り当てを行ってください。これらは、これ以降、サイクリックセクション(OB1)で変更できません。これらの固定パラメータ値は、インスタンス DB にすでに書き込まれているため、周期的な呼び出しではパラメータ設定されません。

例:インスタンス DB110 による FB2 のパラメータ設定

以下の例は、OB100 および OB1 コンポーネントの効果的な配分の実装方法を示しています。

まず、サイクリックプログラムの通常の呼び出しを示します。

```
CALL FB2, DB110(
     Req :      M 100.0,
     NumVar :   2,                    //2 つの変数の読み取り
     Addr1 :    NCVAR.C1_RP_rpa0_0
     Line1 :    W#16#1
     Addr2 :    NCVAR.C1_RP_rpa0_0
     Line2 .    W#16#2
     Error :    M1.0,
     NDR :      M1.1,
     State :    MW 2,
     RD1 :      P#M 4.0 REAL 1,
     RD2 :      P#M 24.0 REAL 1,
```

ここからは、プログラム呼び出しの修正されたバージョンです。
ここでは、OB100 での呼び出しが表示されます。

```
CALL FB2, DB110(
     Req :      FALSE,
     NumVar :   2,                    //2 つの変数の読み取り
```

```
Addr1 :=    NCVAR.C1_RP_rpa0_0
Line1 :     W#16#1
Addr2 :=    NCVAR.C1_RP_rpa0_0
Line2 .     W#16#2
RD1 :=      P#M 4.0 REAL 1,
RD2 :=      P#M 24.0 REAL 1,
```

ここでは、OB1 に残っている呼び出しが表示されます。

```
CALL FB2, DB110(
     Req :      M0.0,
     Error :    M1.0,
     NDR :      M1.1,
     State :    MW 2,
```

注記

この方法により、OB1 でより短いサイクルタイムが実現されます。OB1 周期ごとにインスタンス DB に内部パラメータ値をコピーする必要がないためです。

変更による節約:

インスタンス DB に関する 3 つの整数値および 4 つの ANY パラメータを周期的にコピーする動作。これは、インスタンスデータブロックでの 3 回の伝送により定数を 3 回ロードすると発生します。各 ANY 伝送の場合、その後の伝送により定数がデータブロックに 4 回ロードされます。

14.20 データリスト

14.20.1 マシンデータ

14.20.1.1 表示マシンデータ

番号	識別子:$MM_	説明
9032	HMI_MONITOR	HMI モニタ情報用 PLC データの特定

14.20.1.2 NC 別マシンデータ

番号	識別子: $MN_	説明
10100	PLC_CYCLIC_TIMEOUT	PLC 周期監視時間
14504	MAXNUM_USER_DATA_INT	ユーザーデータの数(INT)
14506	MAXNUM_USER_DATA_HEX	ユーザーデータの数(HEX)
14508	MAXNUM_USER_DATA_FLOAT	ユーザーデータの数(FLOAT)
14510	USER_DATA_INT	ユーザーデータ(INT)
14512	USER_DATA_HEX	ユーザーデータ(HEX)
14514	USER_DATA_FLOAT[n]	ユーザーデータ(FLOAT)

注記

整数/16 進数形式のマシンデータは、NC では DWORD として演算されます。浮動小数点形式のマシンデータは、NC では FLOAT (8 バイト IEEE)として管理されます。これらは NC/PLC インターフェースにのみ格納され、PLC 起動時でも DB20 から PLC ユーザープログラムにより読み取ることができます。

14.20.1.3 チャネルマシンデータ

番号	識別子: $MC_	説明
28150	MM_NUM_VDIVAR_ELEMENTS	PLC 変数を書き込むための要素の数

14.20.2 信号

14.20.2.1 操作パネルからの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
有効な SINUMERIK 操作エリア	DB19.DB21	DB1900.DBB1
現在の画面番号	DB19.DBW24	DB1900.DBW4

P4 : SINUMERIK 828D 用 PLC 15

15.1 概要

15.1.1 PLC ファームウェア

SINUMERIK 828D の PLC は、SIMATIC S7-200 命令セットに基づく内蔵 PLC です。

PLC ユーザープログラムは基本的に、Windows PC を使用して「PLC プログラミングツール」でプログラムされます。 また、制御装置の操作画面を介して PLC を診断および編集することができます。 そのために、「ラダーアドオンツール」を制御装置で使用できます。

以下の特記事項に注意してください。

- PLC ユーザープログラムは、すべてがラダーロジック(LAD)でプログラムされています。
- S7-200 のプログラミング言語のサブセットがサポートされています。
- CPU へのロード時に、実行用コードに加えて、プロジェクトデータ全体(シンボルとコメントを含む)が制御装置にロードされます。 つまり、制御装置には常に、現在実行されている PLC ユーザープログラムに一致するプロジェクトが存在します。
- CPU からのロード時に、プロジェクトデータ全体(シンボルとコメントを含む)が PLC プログラミングツールにロードされ、このツールを使用して処理/編集することができます。
- タイプに応じて、データと処理情報を管理してください。 データにアクセスするときはいつも、宣言されたデータタイプを一貫して使用してください。

15.1.2 PLC ユーザーインタフェース

ユーザーインタフェースは、PLC と NC および HMI の間ですべての信号とデータの交信も行う PLC ファームウェアによって設定されます。

ユーザーインタフェースは以下の部分で構成されます。

- 周期的な交信をおこなうデータインタフェース(「データインタフェース (ページ 1382)」を参照してください)
- 機能またはタスク関連のデータ交信をおこなう機能インタフェース(「機能インタフェース (ページ 1406)」を参照してください)

このようなインタフェースの構造化されたデータ(保持と非保持)は、ファームウェアによるデータブロックへの割り当てによってユーザーが使用可能になります。NC (NC、工具管理、NC チャネル、軸、主軸、 …)と HMI が、PLC ユーザープログラムの「通信相手」です。

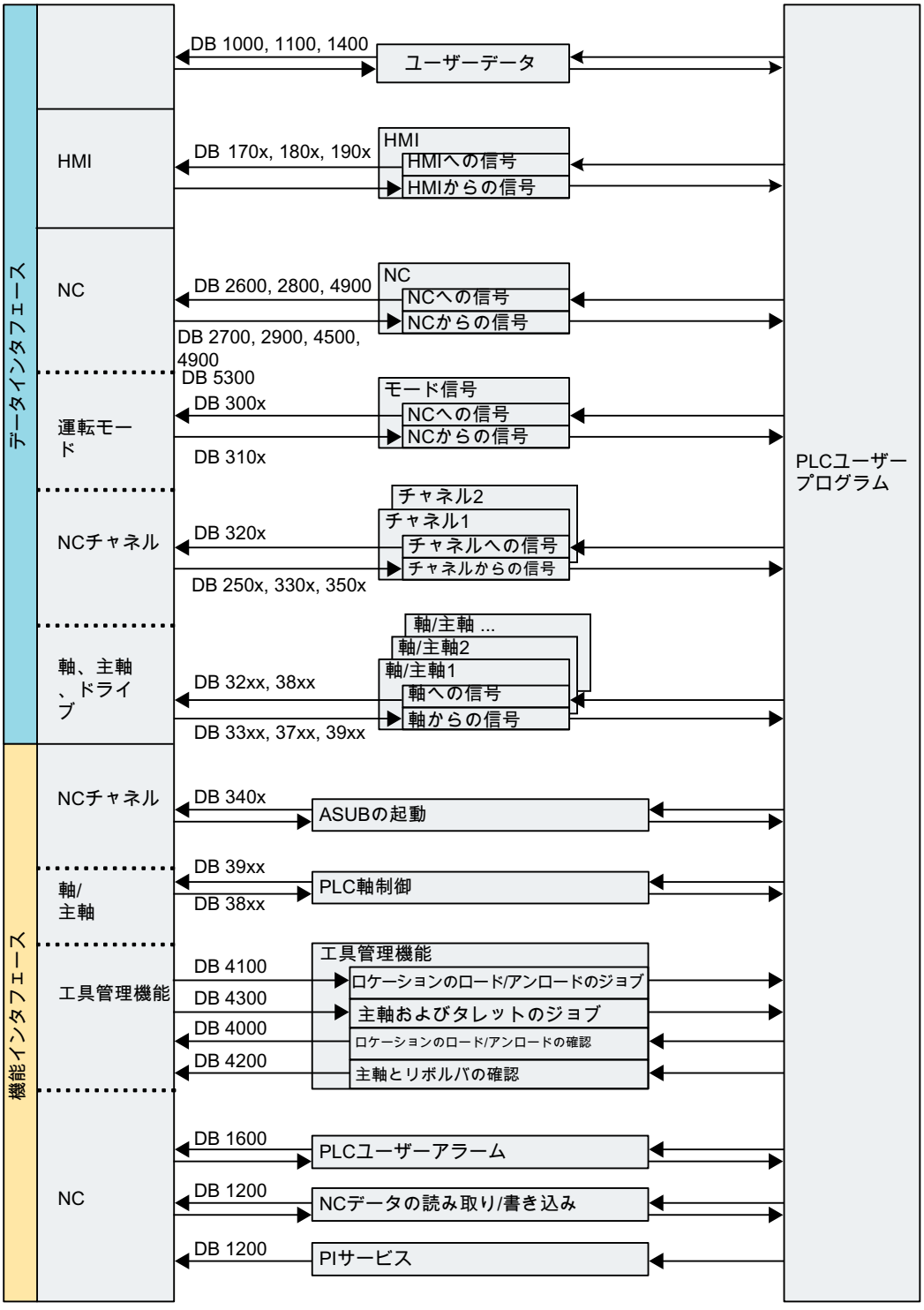


図 15-1　PLC 828D のユーザーインタフェースの概要

15.1.2.1 周期的に交信されるデータ

データは、PLC と NC の間だけでなく、PLC と HMI の間でも交信されます。

PLC へのデータは、ユーザープログラムのサイクル開始時にファームウェアにより提供されます。これにより、たとえば、NC からの信号は 1 周期全体にわたって一定のままであることが保証されます。

ファームウェアは、ユーザープログラムのサイクル終了時にデータを PLC から NC または HMI に伝送します。

PLC ↔ NC インタフェース

サイクリックデータには、状態信号(「プログラム実行中」、「プログラム中断中」)、制御信号(起動、停止)、補助命令、G 命令が含まれます。

データは、以下のグループの信号で構成されます。

- モード
- チャネル
- 軸/主軸
- 一般 NC 信号

インタフェース PLC ↔ HMI

これは、以下のグループの信号です。

- リストを介したプログラムの選択
- メッセンジャ制御命令
- HMI との間の一般信号
- 保守スケジューラとの間の信号
- 操作パネル(保持領域)からの信号
- HMI(保持領域)との間の一般選択/状態信号

15.1.2.2 アラームとメッセージ

DB1600 のユーザーインタフェースにより、HMI に故障および操作メッセージを表示するオプションが提供されます。

ファームウェアが入力された信号を確認し、この信号を表示先の HMI にアラームおよびメッセージの発生／解除として送信します。HMI が障害テキストを管理します。

注記

最大 8 個の PLC アラームが HMI に表示されます。

15.1.2.3 保持データ

保持データ用に、ユーザーデータブロック DB9000 - DB9063 とデータ領域 DB1400.DBW0 - DBW127 があります。 そこに、電源切断/投入後も有効のままでなければならないすべてのデータを保存することができます。 保持データは不揮発性メモリに保存されますが、データのバックアップ用ではありません。

15.1.2.4 非保持データ

非保持データ(ビットメモリ、タイマ、カウンタなど)は、制御装置が起動するたびに消去されます。

15.1.2.5 PLC マシンデータ

PLC マシンデータは、NC マシンデータ領域にあります。電源投入時に、このデータは PLC ファームウェアによって PLC ユーザーインタフェースの DB4500 に転送され、そこで PLC ユーザープログラムによって使用されます。

参照先

『SINUMERIK 828D パラメータマニュアル』

15.1.3 PLC 主要データ

内蔵 PLC には、1 回の固定 PLC 周期ですべて実行される、24000 個の PLC 命令が可能なプログラムメモリがあります。

追加で使用できる INT0 割り込みプログラムでは、最大 500 個の命令を実行することができます。これはサーボ同期で実行され、可能な限り最高速度でプロセスイベントに応

答するのを可能にします。このため、割り込み対応の PLC I/O モジュールは必要ありません。

データ		数	特記事項
メインプログラム(MAIN)		1	
サブプログラム(SBRx)		256	
割り込み		2	
時間起動型の割り込み		1	サーボ同期割り込みプログラム
アラーム/メッセージ		1248	最大 8 個のアラームが表示されます。
ビットメモリ		4096	非保持。
カウンタ		64	非保持。
タイマ、 内訳:		128	非保持。
	10 ms 単位	112	
	100 ms 単位	16	
それぞれが最大 512 バイトのユーザーデータブロック		64	アドレス範囲 DB9000 ～ DB9063
NC ↔ PLC データ伝送			固定のパラメータ設定可能インターフェース経由。

15.1.4 PLC I/O、高速オンボード入力/出力

高速なオンボード入出力の特性とその応答時間については、「高速オンボード入出力部 (ページ 1377)」の章を参照してください。

I/O モジュール、機械操作パネル、オンボード入力/出力の PLC への割り当てについての情報は、以下を参照してください。

参照先:

『マニュアル SINUMERIK 828D　PPU』

15.1.5 PLC ツールボックス

15.1.5.1 スター/デルタ切り替え

スター/デルタ切り替え用に、以下のブロックが PLC ツールボックスで提供されます。

- StarDelta

注記

このブロックを使用して、スター/デルタ切り替えをおこなうことができます - SMI が SINAMICS S120 に接続された 1PH8 主軸モータの場合も同様です。

15.2 高速オンボード入出力部

ディジタルの入力信号および出力信号は、コントロールシステムが特に高速で処理しなければならないため、PPU モジュールには複数のインターフェースがあり、これを使用して、信号を直接接続できます。

- コネクタ X242: 4 個の入力信号、4 個の出力信号
- コネクタ X252: 4 個の入力信号、2 個の出力信号

インターフェース	I/O 信号		アドレス指定
X242	入力:	DIN1 - DIN4	I256.0 ... I256.3
	出力部:	DOUT1 - DOUT4	Q256.0 ... Q256.3
X252	入力:	DIN9 - DIN12	I256.4 ... I256.7
	出力部:	DOUT9 - DOUT10	Q256.4 ... Q256.5

応答時間

この結果、位置制御周期、実行場所、使用されるディジタル入出力に対応して以下の応答時間が発生します。

位置制御周期		1.5 ms	3 ms
	以下による処理:	応答時間 1)	
	• サイクリック運転のサブプログラム SBRx (MAIN、OB1) • ディジタル入/出力部:**I/O モジュール PP 72/48**	12.5 ms	14 ms
	• 割り込みプログラム INT0 (サーボ同期) • ディジタル入/出力部:PPU の**オンボード入/出力部**	3 ms	4.5 ms
1) 入力端子での信号 → PLC プログラムでの処理 → 出力端子での信号			

読み取り/書き込みアクセス操作

入/出力信号は、モジュール入/出力部で直接、読み取られ、書き込まれます。可能な最速のアクセス(サーボ同期)にするには、割り込みプログラム **INT0** で直接操作命令を使用することをお勧めします。

命令	シンボル
ダイレクト NO 接点	-| I |-
ダイレクト NC 接点	-| /I |-
ビットの値の直接セット	-(SI)-
ビットの値の直接リセット	-(RI)-
ビットの値の直接割り当て	-(I)-

パラメータ設定

マシンデータを使用して、NC または PLC にのみ入/出力信号を割り当てることができます。

- MD10366 $MN_HW_ASSIGN_DIG_FASTIN[<n>]
- MD10368 $MN_HW_ASSIGN_DIG_FASTIN[<n>]

ここで、<n>は、入/出力バイトのアドレス指定用インデックス(0 = 1. バイト、1 = 2. バイト、...)

I/O 信号		割り付け	
		NC	PLC
入力:	DIN1 - DIN4	MD10366[0] = 00 01 **01** 01_H	MD10366[0] = 00 01 00 01_H
出力部:	DOUT1 - DOUT4	MD10368[0] = 00 01 **01** 01_H	MD10368[0] = 00 01 00 01_H
入力:	DIN9 - DIN12	MD10366[1] = 00 01 01 01_H	MD10366[1] = 00 01 00 01_H
出力部:	DOUT9 - DOUT10	MD10368[1] = 00 01 01 01_H	MD10368[1] = 00 01 00 01_H

参照先

インターフェースの詳細な仕様は、以下の資料にあります。

SINUMERIK 828D PPU マニュアル

15.3 ラダー図、ラダーエディタおよびラダーアドオンツール

15.3.1 一覧

ラダー図

ラダー図では、PLC ユーザープログラムでの接点とリレーの論理演算がラダーロジック(LAD)として表示されます。

ラダーエディタ

ラダーエディタでは、PLC プロジェクトのブロックおよびネットワークを編集できます。PLC タイプでサポートされる操作はすべて編集に使用可能です。

参照先

ラダーエディタの機能および操作の詳細説明については、『SINUMERIK 操作説明書』を参照してください。

ラダーアドオンツール

ラダーアドオンツールでは、**ラダーエディタが無効になっているときに割り込みルーチン INT100 および INT101 の制限付き編集**が可能です。これは、たとえば、サービスに入出力を再割り当てするときや、エンドユーザーの実際の条件に機械を適合させるときの最終セットアップ作業に役立ちます。

制限事項

ラダーアドオンツールで編集するときは、以下の制限事項が適用されます。

- 空きネットワークのみが編集可能です。既に命令が含まれているネットワークのみが削除できます。
- ネットワークごとに、単純な 1 行を編集できます。
- ネットワークごとに、最大 3 列まで作成することができます。

列	命令	
列 1	• NO 接点 • NC 接点	-| |- -|/|-
列 2 (オプション)	NOT 立ち上がりエッジ 立ち下りエッジ 代入 セット リセット	-|NOT|- -|P|- -|N|- -( ) -(S) -(R)
列 3 (2 番目の列で代入、セット、またはリセット命令が指定されていない場合にのみ使用可能)	代入 セット リセット	-( ) -(S) -(R)

注記

論理式 AND (直列接点)および論理式 OR (並列接点)は使用できません。

ビットの組み合わせは、1 つ以上の論理演算と、出力/ビットメモリへの代入で構成されています。

矢印キーを使用してカーソルをさらに左に移動すると、代入または論理演算のタイプを選択できます。代入の右側にさらに論理演算を置くことはできません。ネットワークは常に、代入で終了してください。

15.3.2 パラメータ設定

アドレスの表示

NC 別マシンデータを使用して、SIMATIC S7 300 表記法に従ってラダー図でのアドレスの表示を設定できます。

MD51230 $MN_ENABLE_LADDER_DB_ADDRESSES = <値>

<値>	意味
0	SIMATIC S7 200 表記法でのアドレス表示(例: Vxxxxx)
1(初期設定)	SIMATIC S7 300 表記法でのアドレス表示(例: DBxx.DBBxxxx)

編集の有効化/ロック

PLC プロジェクトの編集は、以下のマシンデータで有効または無効にできます。

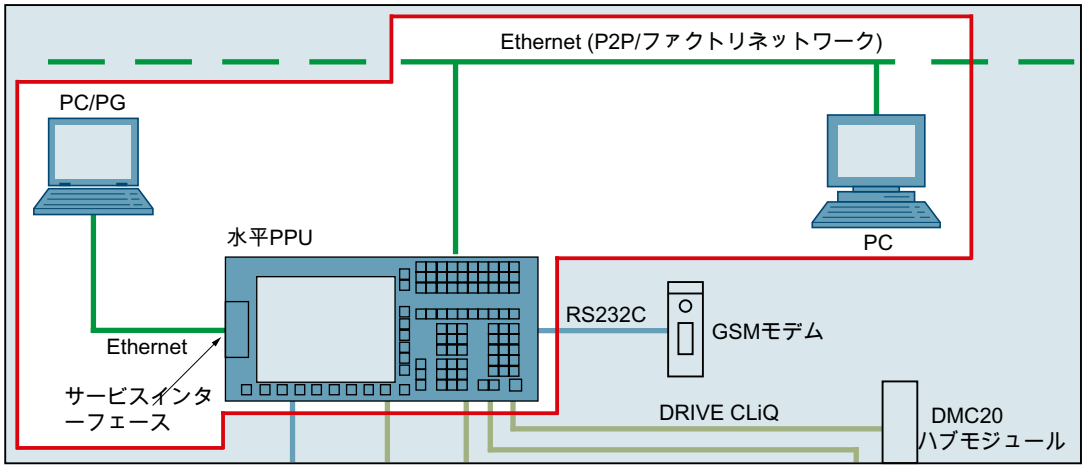
MD51231 $MN_ENABLE_LADDER_EDITOR (INT100/101 に対して PLC ラダーアドオンツールを有効化)

MD51232 $MN_ENABLE_LADDER_EDITOR_ADV (PLC プロジェクト全体に対して PLC ラダーエディタを有効化)

デフォルト設定では、両方の機能が有効です。

15.4 PLC プログラミングツール

「PLC プログラミングツール」は、PLC ユーザープログラムを記述するための使いやすいツールです。これは Windows プログラムなので、Windows PC にインストールしてください。制御装置にオンラインでアクセスするには、Industrial Ethernet ポート、X130 (工場内ネットワーク)または X127 (サービスインターフェース)経由で PC を制御装置に接続してください。



既存のプロジェクトを指定せずに「PLC プログラミングツール」を呼び出すと、新しいプロジェクトが、「Project1」という既定の名前で作成されます。このプロジェクトを使って、すぐに PLC ユーザープログラムを生成でき、任意の名前で保存して、制御装置にロードすることができます。既存のプロジェクトは、通常の Windows の方法で開くことができます。

「PLC プログラミングツール」のオンラインヘルプは、Windows と同様に、「F1」ファンクションキーで呼び出すこともできます。

参照先

- 『SINUMERIK 828D 試運転マニュアル、旋削加工とフライス加工』、「納入と必要条件の範囲」の章>「コントローラによる通信」>「コントローラを使用してプログラミングツール経由で通信する方法」
- PLC プログラミングツールのオンラインヘルプ

15.5 データインタフェース

データは周期的に、PLC と NC の間、ならびに PLC と HMI の間で交換されます。これは特に、HMI から受信され、NC に送信されるデータを有効にするには、ユーザープログラムでそれを制御する必要があることを意味します。

PLC へのデータは、ユーザープログラムサイクルの開始時にファームウェアにより提供されます。これにより、たとえば、NC からの信号は 1 周期全体にわたって一定のままであることが保証されます。

PLC からのデータは、ユーザープログラムのサイクル終了時にファームウェアにより NC または HMI に伝送されます。

このインタフェースのすべてのデータは、SINUMERIK 828D のマニュアル、PPU に記載されています。

15.5.1 PLC-NC インタフェース

このサイクリックデータには、状態信号(「プログラム実行中」、「プログラム中断中」)、制御信号(起動、停止)、補助命令、および G 命令が含まれます。

データは、以下のグループの信号で構成されます。

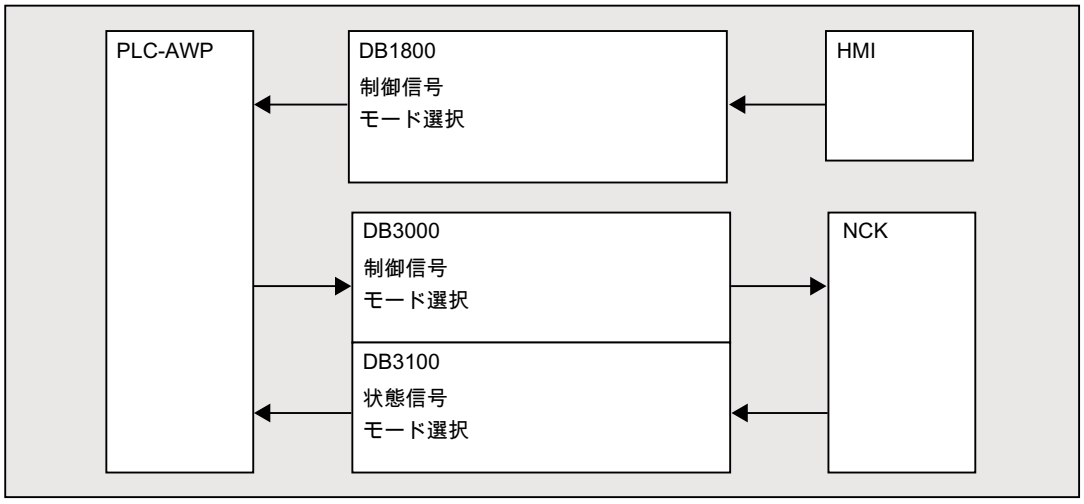
- モード信号
- NC チャネル信号
- 軸および主軸信号
- 一般 NC 信号
- PLC-NC 高速データ交換

15.5.1.1 モード信号

DB3000、3100

機械操作パネルまたは HMI で指定されたモード信号が、NC に伝送されます。

そこで、現在の状態が、NC から PLC に通知されます。

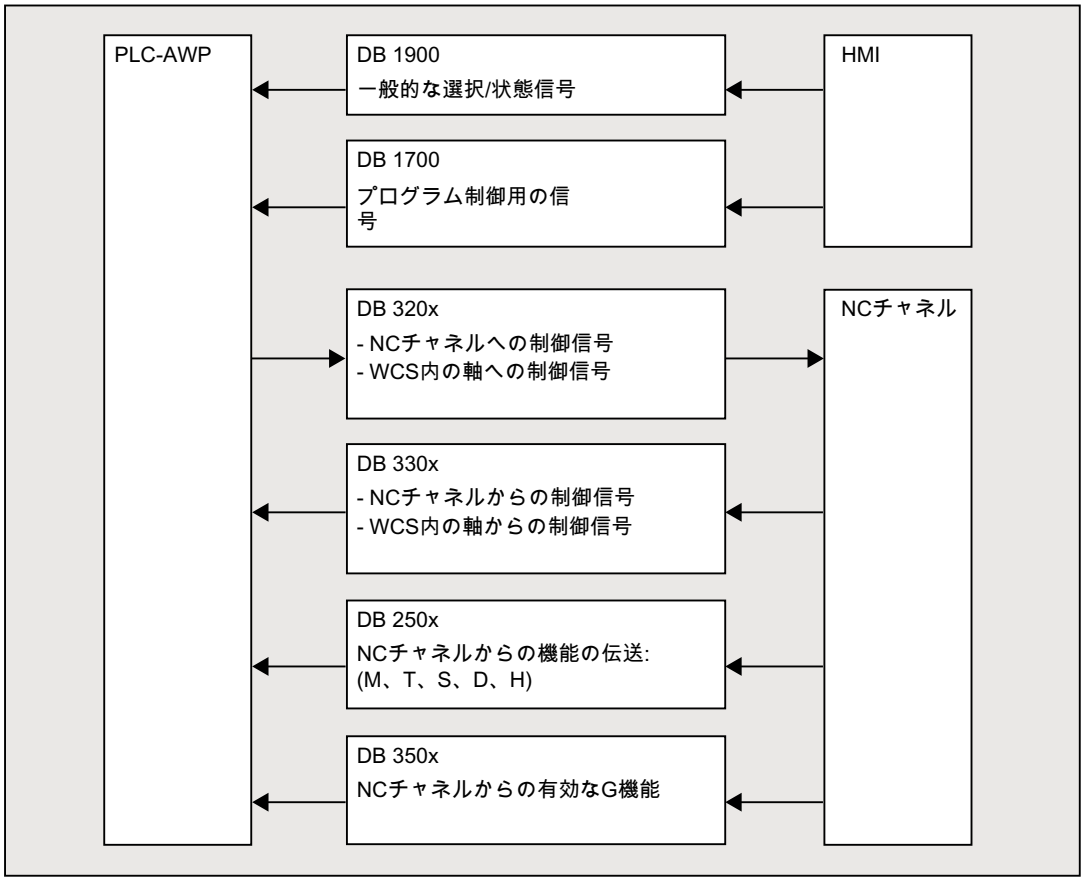

PLC-AWP
DB1800
制御信号
モード選択
HMI
DB3000
制御信号
モード選択
NCK
DB3100
状態信号
モード選択

15.5.1.2 NC チャネル信号

DB250x、320x、330x、350x

信号の構造は次のとおりです。

- 通常のサイクリック伝送による制御/状態信号については、「モード信号 (ページ 1383)」を参照してください。
- 補助命令と G 命令
  これは、2 つの方法でインタフェースデータブロックに入力されます。
  最初に、変更信号によって入力されます。
  更に、M 信号 M0 ～ M99 がデコードされ、関連するインタフェースビットが 1 サイクルの間、設定されます。
  G 命令の場合は、マシンデータで選択されたグループだけがインタフェースデータブロックに入力されます。
  S 値も、関連する M 信号(M03、M04、M05)とともに軸別のインタフェースに入力されます。軸別の送り速度も、該当する軸別のインタフェースに入力されます。



15.5.1.3 軸および主軸信号

DB370x、380x、390x

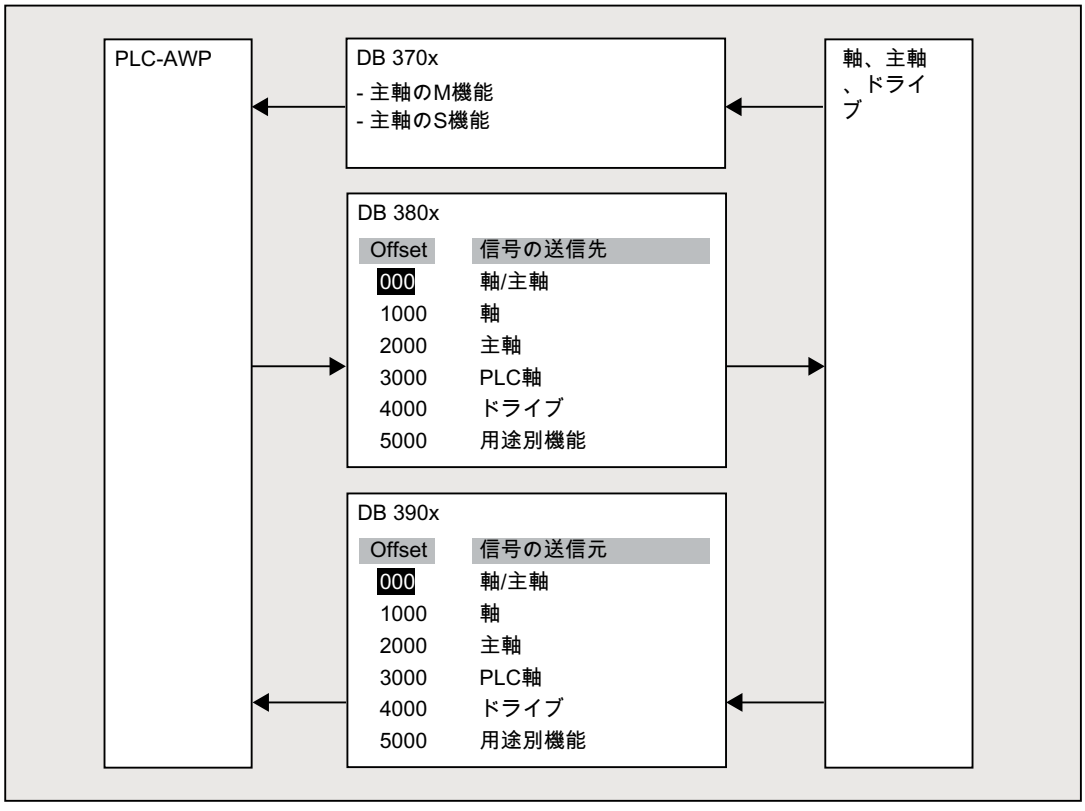
軸別および主軸別の信号は、次のグループに分割されます。

- 共有軸/主軸信号
- 軸信号
- 主軸信号
- ドライブ信号

信号は、以下の例外を除いて周期的に伝送されます。例外は、軸の F 値、M および S 値です。

軸の F 値は、NC プログラム運転中に PLC に伝送される場合、M、S、F 配信処理を介して入力されます。

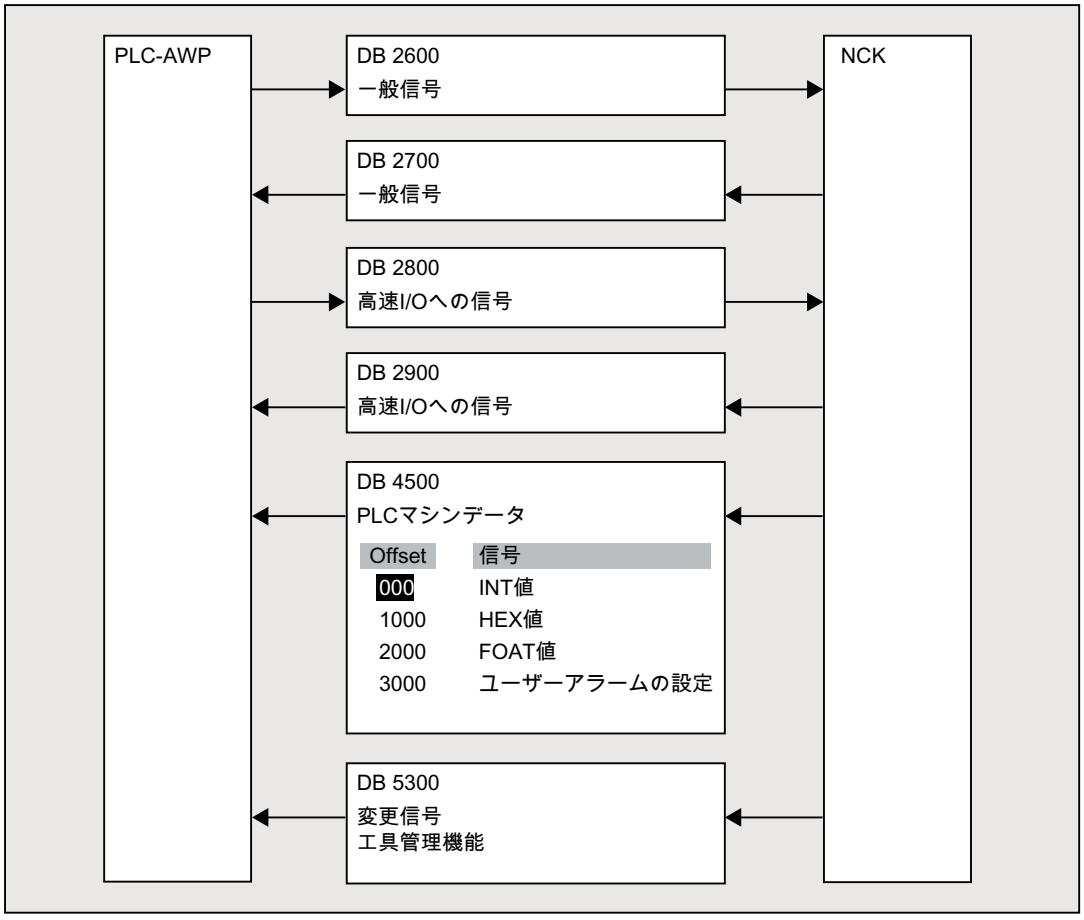
M 値と S 値のどちらかまたは両方の処理が必要な場合は、どちらも、M、S、F 配信処理を介して入力されます。



15.5.1.4 一般 NC 信号

DB2600、2700、2800、2900、4500、5300

- NC のディジタル/アナログ入出力への指令値
- NC のディジタル/アナログ入出力からの現在値
- キースイッチおよび非常停止信号
- NC の準備完了と状態信号



15.5.1.5 PLC-NC 高速データ交換

DB4900

PLC と NC 間の高速情報交換には、1024 バイトのサイズのデータブロック DB4900 が使用されます。

領域(構造体)の割り当ては、NC パートプログラムと PLC ユーザープログラムの間で一致するように調整しなければなりません。

このデータは、命令$A_DBB[x]、$A_DBW[x]、$A_DBD[x]、および$A_DBR[x]; 0 ≤ x ≤ 1023 を使用して、NC パートプログラムからアクセスできます(『パラメータマニュアル　システム変数一覧表』を参照)。

この場合、データの配置はデータ形式に応じて選択してください。つまり、DWORD は 4 バイト境界で、WORD は 2 バイト境界で始まります。バイトは、データフィールド内の任意のオフセットに配置できます。

バイト、ワード、ダブルワードアクセスの場合は、データの一貫性が保証されます。複数のデータを伝送する場合、ブロックの適切さや一貫性を検出するのに使用されるセマフォを使用して、ユーザー側で一貫性を確保してください。

15.5.2 PLC-HMI 間インタフェース

DB1700、1800、1900

この信号は、セクション「PLC-NC インタフェース (ページ 1383)」の図ですでに指定されています。

「データインタフェース (ページ 1382)」に記載する内容をもう一度参照してください。

HMI から受信されたデータと NC に送信されるデータは、自動的に NC インタフェース領域に入力されません。このような信号およびデータは、ユーザープログラムによって制御する必要があります。

これには、次の信号が含まれます。

- リストを介したプログラムの選択
- メッセンジャ制御命令
- HMI との間の一般信号
- 保守スケジューラとの間の信号
- 操作パネル(保持領域)からの信号
- HMI(保持領域)との間の一般選択/状態信号

15.5.2.1 プログラムの選択

機能

事前選択されたプログラム/ワークを、NC で加工するために、PLC/HMI インターフェースを介して選択できます。

事前設定は、PLC プログラムリスト(*.ppl)と呼ばれているファイルにプログラム/ワークを入力することによって行います。

必要条件

HMI がタスクを処理できるよう、以下のマシンデータを設定してください。

MD9106 $MM_SERVE_EXTCALL_PROGRAMS

分野別の PLC プログラムリストを有効にするために、適切なマシンデータと、少なくとも保護レベルのパスワードを設定してください。

- **ユーザー**領域
  - MD51041 $MN_ENABLE_PROGLIST_USER = 1
  - 保護レベルのパスワード:3 (ユーザー)
  - プログラムリスト: /user/sinumerik/hmi/plc/programlist/plc_proglist_user.ppl
- **工作機械メーカ(OEM)**領域
  - MD51043 $MN_ENABLE_PROGLIST_MANUFACT = 1
  - 保護レベルのパスワード:1 (工作機械メーカ)
  - プログラムリスト: /oem/sinumerik/hmi/plc/programlist/plc_proglist_manufacturer.ppl

プログラムリストの構造

プログラムリストはテキストファイルです。各行には、以下の情報が含まれます。

<プログラム番号> <プログラムパス><プログラム名> [CH=<チャネル番号>]

- プログラム番号
  分野に応じてプログラムリストで使用できるプログラム番号。
  - user:1 - 100
  - 個別(oem_i):101 - 200
  - 工作機械メーカ(OEM):201 - 255
- プログラムパス
  プログラムパスは、完全な絶対パスで指定してください。
  プログラムパスの指定については、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
- チャネル番号
  チャネル番号「CH=<チャネル番号>」の指定はオプションです。これが必要なのは、NC に複数のチャネルがある場合のみです。

以下の抜粋の例は、ユーザープログラムリストの構造を示しています。

プログラムリスト:plc_proglist_user.ppl

```
1 //DEV2:/MPFDir/PROG_01.MPF CH=1
2 //DEV2:/MPFDir/PROG_01.MPF CH=2
```

プログラムリストへの入力の生成

プログラムリスト(*.ppl)への入力は、ファイルで直接編集するか、ユーザーインターフェースの画面形式に入力することができます。

- **ユーザー**領域のユーザーインターフェースで
  操作エリア[プログラムマネージャ|ETC キー([>])|プログラムリスト]
- **工作機械メーカ**領域のユーザーインターフェースで
  操作エリア[コミッショニング|システムデータ|ETC キー([>])|プログラムリスト]

プログラムの選択:ジョブインターフェース

注記

PLC は、最後のジョブに対して HMI が確認応答している場合にのみ、新しいジョブを要求できます。DB1700.DBB2000 == 0

プログラムリスト

DB1700.DBB1001 = <プログラムリストの番号>

番号	プログラムリスト
129	/user/sinumerik/hmi/plc/programlist/**plc_proglist_user.ppl**
130	/oem_i/sinumerik/hmi/plc/programlist/**plc_proglist_individual.ppl**
131	/oem/sinumerik/hmi/plc/programlist/**plc_proglist_manufacturer.ppl**

プログラム番号

プログラム番号は、選択したプログラムリストに含まれるプログラムを表します。

DB1700.DBB1002 = <プログラム番号>

- ユーザー領域:1 - 100
- 個別領域:101 - 200
- OEM 領域:201 - 255

プログラムの選択の要求

DB1700.DBX1000.7 = 1

プログラムの選択:確認のインターフェース

ジョブの確認

- DB1700.DBX2000.7 == 1 (選択が識別済み)
- DB1700.DBX2000.3 == 1 (プログラムが選択済み)
- DB1700.DBX2000.2 == 1 (プログラムの選択時のエラー、異常 ID DB1700.DBB2001 を参照)
- DB1700.DBX2000.1 == 1 (ジョブの完了)

エラーの検出

DB1700.DBB2001 == <異常 ID>

エラーの検出	
値	**意味**
0	異常なし。
1	無効なプログラムリスト番号(DB1700.DBB1001)。

エラーの検出	
3	ユーザーが指定したプログラムリスト plc_proglist_main.ppl が見つからない (DB1700.DBB1001 ≠ 129、130、131 の場合のみ)
4	無効なプログラム番号(DB1700.DBB1002)。
5	選択されたワーク内のジョブリストを開けませんでした。
6	ジョブリスト内の異常(ジョブリストのインタープリタがアラームを返します)。
7	ジョブリストのインタープリタが、空のジョブリストを返します。

プログラムの選択:ジョブの処理

プログラムを選択するためのジョブは、以下のように実行します。

1. 確認バイトのチェック:DB1700.DBB2000 == 0
   確認バイトが 0 ではない場合は、最後のジョブが、まだ完了していません。
2. プログラムリストの指定:DB1700.DBB1001
3. プログラム番号の指定:DB1700.DBB1002
4. プログラムを選択するための要求の設定:DB1700.DBX1000.7 = 1
5. 確認インターフェースとエラーインターフェースの評価:DB1700.DBB2000 および DBB2001
   以下が該当する限り、HMI 側では、要求がまだ完了されていません。DB1700.DBX2000.3 == 1 (有効)
   以下の 2 つの信号のいずれかが設定されている場合は、要求が完了されています。
   - DB1700.DBX2000.1 == 1 == 1 (ジョブの完了)
   - DB1700.DBX2000.2 == 1 (エラー)
6. 要求を完了するには、プログラムの選択要求をリセットしてください。DB1700.DBX1000.7 = 0
7. 確認バイトのリセットにより HMI が新しい要求を受け入れられる状態になったことを示す HMI 信号。DB1700.DBB2000 == 0

15.5.2.2 操作エリア番号

有効な操作エリアの数は、以下に表示されます。DB1900.DBB1

操作エリア番号

操作エリア	番号
運転	201
パラメータ	205

操作エリア	番号
プログラム	203
プログラムマネージャ	202
アラーム診断	204
セットアップ	206

15.5.2.3 画面番号

現在の画面番号は、以下に表示されます。DB1900.DBW4

画面番号の範囲

以下の画面番号の範囲が使用可能です。

- JOG、Manual Machine (ページ 1393)
- 基準点アプローチ (ページ 1400)
- MDI (ページ 1400)
- AUTOMATIC (ページ 1401)
- パラメータ操作エリア (ページ 1402)
- プログラム操作エリア (ページ 1403)
- プログラムマネージャの操作エリア (ページ 1404)
- 診断操作エリア (ページ 1404)

画面番号:JOG、Manual Machine

JOG モード

図	番号
旋削加工	
引き継ぎが可能な、すべての画面のサイクルスタート画面	81
フライス盤用途	
引き継ぎが可能な、すべての画面のサイクルスタート画面	3
旋削/フライス削り	
開始画面	19

図	番号
T、S、M	2
Set WO	21
位置決め	4
正面削り	18
切削	80
すべてのユーザー画面のサイクルスタート画面	91
一般設定	1
マルチチャネル機能の設定	106
干渉の回避の設定	107
計測ログの設定	108
旋回	60
すべての G 命令	100
実際のズーム値(MCS/WCS)	101
ねじの原点同期	102
後退	103
ハンドル	104
操作の同期	105
旋削加工:ワーク原点	
ワーク原点(メインメニュー)	30
ユーザー画面	31
ユーザー画面	34
ユーザー画面	35
ユーザー画面	36
ユーザー画面	37
ユーザー画面	38
ユーザー画面	40
測定エッジ Z	5
旋削加工:ワーク、計測	
工具計測(メインメニュー)	50
手動 X またはユーザー画面	51
手動 Y	71

図	番号
手動 Z またはユーザー画面	52
ズームまたはユーザー画面	53
ユーザー画面	54
ユーザー画面	55
プローブ校正 X またはユーザー画面	56
プローブ校正 Z またはユーザー画面	57
Z での自動長さ	58
Y での自動長さ	73
X での自動長さ	59
フライス加工:ワーク原点	
ワーク原点(メインメニュー)	30
エッジ X を計測	5
エッジ X を計測	22
測定エッジ Z	23
ユーザー画面	7
エッジ揃えまたはユーザー画面	31
距離 2 エッジまたはユーザー画面	32
直角コーナ	33
任意のコーナまたはユーザー画面	8
長方形ポケット	34
1 個の穴またはユーザー画面	9
2 個の穴	35
3 個の穴	36
4 個の穴	37
長方形凸形状	38
1 個の円形スピゴットまたはユーザー画面	10
2 個の丸凸形状	39
3 個の丸凸形状	40
4 個の丸凸形状	41
レベルを設定	42
プローブ長さの校正またはユーザー画面	11

図	番号
プローブ半径の校正	12
フライス加工:ワーク、計測	
工具計測(メインメニュー)	50
長さを測定、手動(フライス工具で)、 または X で長さを測定、手動(旋削工具で)、 またはユーザー画面	16
Y で長さを測定、手動(旋削工具で)	74
Z で長さを測定、手動(旋削工具で)	24
直径を測定、手動、またはユーザー画面	17
長さを測定、自動(フライス工具で)、 または X で長さを測定、自動(旋削工具で)、 またはユーザー画面	13
Y で長さを測定、自動(旋削工具で)	75
Z で長さを測定、自動(旋削工具で)	25
直径を測定、自動、またはユーザー画面	14
ユーザー画面	51
プローブ校正またはユーザー画面	15
固定点の校正またはユーザー画面	52
RunMyScreens (JobShopIntegration を設定した場合のみ)	
最初の水平ソフトキーのユーザー画面	96
2 番目の水平ソフトキーのユーザー画面	98
3 番目の水平ソフトキーのユーザー画面	99
4 番目の水平ソフトキーのユーザー画面	94
5 番目の水平ソフトキーのユーザー画面	95
6 番目の水平ソフトキーのユーザー画面	92
7 番目の水平ソフトキーのユーザー画面	97
8 番目の水平ソフトキーのユーザー画面	90
9 番目の水平ソフトキーのユーザー画面	83
10 番目の水平ソフトキーのユーザー画面	82
11 番目の水平ソフトキーのユーザー画面	93
12 番目の水平ソフトキーのユーザー画面	84
13 番目の水平ソフトキーのユーザー画面	85

図	番号
14 番目の水平ソフトキーのユーザー画面	86
15 番目の水平ソフトキーのユーザー画面	87
16 番目の水平ソフトキーのユーザー画面	88

JOG モード、Manual Machine

DB19.DBB24	
図	画面番号
旋削/フライス削り	
テーパ旋削	61
角度フライス削り	62
ストップ	63
直線	1300
すべての軸が直線	1330
X アルファが直線	1340
Z アルファが直線	1350
円弧	1360
穴あけ	1400
センタ穴あけ	1410
穴あけ、ねじ切りの中央揃え	1420
穴あけ、センタリング	1433
穴あけ、穴あけ	1434
穴あけ、リーマ仕上	1435
穴あけ、ボーリング	1436
穴あけ、深穴ドリル	1440
穴あけ、深穴ドリル 2	1441
穴あけ、タッピング	1453
ねじ切り穴あけ	1455
第	1473
位置行	1474
位置グリッド	1477
位置フレーム	1478

DB19.DBB24	
図	画面番号
円弧位置決め	1475
円弧位置決め	1479
障害物	1476
旋削	1500
旋削、切り粉除去 1	1513
旋削、切り粉除去 2	1514
旋削、切り粉除去 3	1515
旋削、溝 1	1523
旋削、溝 2	1524
旋削、溝 3	1525
旋削、アンダーカット形状 E	1533
旋削、アンダーカット形状 F	1534
旋削、アンダーカットねじ DIN	1535
旋削、アンダーカットねじ	1536
旋削、ねじ、縦	1543
旋削、ねじ、テーパ	1544
旋削、ねじ、面削り	1545
旋削、ねじ、連続	1546
旋削、遮断	1550
フライス削り	1600
フライス削り、正面削り	1610
フライス削り、長方形ポケット	1613
フライス削り、円形ポケット	1614
フライス削り、長方形の凸形状	1623
フライス削り、円弧の凸形状	1624
フライス削り、直線溝	1633
フライス削り、円周溝	1634
フライス削り、オープン溝	1635
フライス削り、多角形	1640
フライス削り、ねじ切り	1454

DB19.DBB24	
図	画面番号
フライス削り、彫り込み	1670
輪郭旋削	1200
輪郭旋削、新規輪郭/最終輪郭	1210
輪郭旋削、輪郭に沿った旋削	1220
輪郭旋削、輪郭の溝削り	1230
輪郭旋削、輪郭のプランジ旋削	1240
輪郭切削	1100
輪郭のフライス削り、新規輪郭/最終輪郭	1110
輪郭のフライス削り、軌跡のフライス削り	1120
輪郭フライス削り、センタリング	1130
輪郭フライス削り、下穴あけ	1140
輪郭フライス削り、輪郭ポケット	1150
旋削加工:シミュレーション	
側面図	1740
正面図	1750
3D 表示	1760
2 画面表示	1770
片側断面図	1780
旋削加工:同時描画	
側面図	1741
正面図	1751
3D 表示	1761
2 画面表示	1771
機械スペース	1791
片側断面図	1781
フライス加工:シミュレーション	
平面図	1742
3D 表示	1760
正面図	1744
背面図	1746

DB19.DBB24	
図	画面番号
左から	1748
右から	1752
片側断面図	1780
旋削図	1782
フライス加工:同時描画	
平面図	1743
3D 表示	1761
正面図	1745
背面図	1747
左から	1749
右から	1753
機械スペース	1791
片側断面図	1781
旋削図	1783

画面番号:基準点アプローチ

図	番号
実際のズーム値 MCS/WCS	101

画面番号:MDI

図	番号
MDI	20
すべての G 命令	100
実際のズーム値 MCS/WCS	101
ハンドル	104
操作の同期	105
プログラム制御	210
設定	250

画面番号:AUTOMATIC

図	番号
自動	200
オーバーストア	202
プログラム制御	210
ブロックサーチ	220
一般設定	250
マルチチャネル機能の設定	106
干渉の回避の設定	107
すべての G 命令	100
実際のズーム値 MCS/WCS	101
ハンドル	104
操作の同期	105
旋削加工:同時描画	
側面図	243
正面図	244
3D 表示	245
2 画面表示	246
機械スペース	247
片側断面図	253
フライス加工:同時描画	
平面図	242
3D 表示	244
正面図	248
背面図	249
左から	251
右から	252
機械スペース	247
片側断面図	253
旋削図	254

画面番号:パラメータ操作エリア

図	番号
工具リスト	600
工具摩耗	610
ユーザー工具リスト	620
マガジン	630
ゼロオフセット	
ゼロオフセット、有効	642
ゼロオフセット、概要	643
ゼロオフセット、基本	644
ゼロオフセット、G54 - G509	645
ゼロオフセットの詳細、有効、概要、基本、または G54 - G509	647
ユーザー変数	
R 変数	650
グローバル GUD 1 (SGUD)	660
グローバル GUD 2 (MGUD)	661
グローバル GUD 3 (UGUD)	662
グローバル GUD4	663
グローバル GUD5	664
グローバル GUD6	665
グローバル GUD7	666
グローバル GUD8	667
グローバル GUD9	668
チャネル GUD 1 (SGUD)	690
チャネル GUD 2 (MGUD)	691
チャネル GUD 3 (UGUD)	692
チャネル GUD 4	693
チャネル GUD 5	694
チャネル GUD 6	695
チャネル GUD 7	696
チャネル GUD 8	697
チャネル GUD 9	698

図	番号
ローカル LUD	681
ローカル LUD/PUD	684
セッティングデータ	
作業領域リミット	671
主軸データ	670
主軸チャックデータ	672
Ctrl-Energy	
Ctrl-Energy、メインメニュー	6170
Ctrl-Energy、分析	6171
Ctrl-Energy、プロファイル	6172
Ctrl-Energy、グラフィック分析	6176
Ctrl-Energy, 長期的計測分析	6177
Ctrl-Energy、詳細分析	6179
Ctrl-Energy、計測比較	6178

画面番号:プログラム操作エリア

図	番号
旋削加工:シミュレーション	
側面図	413
正面図	414
3D 表示	415
2 画面表示	416
片側断面図	423
フライス加工:シミュレーション	
平面図	412
3D 表示	414
正面図	418
背面図	419
左から	421
右から	422

図	番号
片側断面図	423
旋削図	424

画面番号:プログラムマネージャ操作エリア

図	番号
NC ディレクトリ	300
ローカルドライブ	325
USB/ドライブ 1 構成	330
ドライブ 2 構成	340
ドライブ 3 構成	350
ドライブ 4 構成	360
ドライブ 5 構成	383
ドライブ 6 構成	384
ドライブ 7 構成	385
ドライブ 8 構成	386

画面番号:アラーム診断操作エリア

図	番号
アラームリスト	500
メッセージ	501
アラームログ	502
NC/PLC 変数	503

15.5.2.4 HMI モニタ

機能

HMI モニタは、自由に選択可能なデータブロックの 8 バイトのデータ領域です。ここで、HMI が PLC ユーザープログラムに、以下のデータを提供できます。

- 操作エリア番号 (ページ 1392)
- 画面番号 (ページ 1393)

パラメータ設定

データ領域は、以下の表示マシンデータを使用して設定されます。

MD9032 $MM_HMI_MONITOR =「文字列」

ここで、「文字列」=「DB<DB 番号>.DBB<バイトアドレス>」

注記

偶数のバイトアドレス

データ領域は必ず、偶数のバイトアドレスで始まります。

データ領域の構造

バイト	意味
EB n + 0	有効な SINUMERIK 操作エリア
EB n + 1	予備
EB n + 2	現在の画面番号
EB n + 3	
EB n + 4	予備
...	...
EB n + 7	予備

必要条件

HMI モニタが有効なときは、以下の PLC/HMI インターフェース信号が処理されなくなります。

- DB1900.DBB5003 (PLC ハードキー)
- DB1900.DBB0001 (有効な SINUMERIK 操作エリア)
- DB1900.DBW0004 (現在の画面番号)

15.6 機能インタフェース

15.6.1 NC 変数の読み取り/書き込み

15.6.1.1 ユーザーインタフェース

PLC ユーザープログラムは、NC/PLC インタフェース「Read/write NC variable」を介して、最高で 8 個の NC 変数を同時に読み取りまたは書き込みできます。

ジョブの一環として以下の手順をおこなってください(読み取り/書き込み)。

1. ジョブの指定 (ページ 1407)
2. ジョブの管理: ジョブの開始 (ページ 1409)
3. ジョブの管理: ジョブの終了の待機 (ページ 1409)
4. ジョブの管理: ジョブの完了 (ページ 1410)
5. ジョブの評価 (ページ 1412)

ジョブのフロー図:「ジョブの管理:フローチャート (ページ 1411)」を参照してください。

15.6.1.2 ジョブの指定

変数別ジョブインタフェース

ジョブで処理される各変数を、**変数別ジョブインタフェース**でそのパラメータにより指定してください。全般の識別子については、インタフェースからアクセスできる変数毎に、後で詳しく説明します。

DB120x [1]	**NC データの読み取り/書き込み(PLC → NC)**							
バイト	**ビット 7**	**ビット 6**	**ビット 5**	**ビット 4**	**ビット 3**	**ビット 2**	**ビット 1**	**ビット 0**
DBB1000	変数インデックス							
DBB1001	領域番号							
DBW1002	行インデックス、NC 変数 x							
DBW1004	列インデックス、NC 変数 x							
DBW1006	---							
DBD1008	書き込:NC 変数 x へのデータ(変数のデータタイプ:1…4 バイト)[2]							
DBD1012	---							
DBD1016	書き込:NC 変数 x へのデータ(REAL) [3]							
DBD1020	書き込:NC 変数 x へのデータ(DWORD/DINT) [3]							
DBW1024	書き込:NC 変数 x へのデータ(WORD/INT) [3]							
DBB1026	書き込:NC 変数 x へのデータ(BYTE) [3]							
DBB1027	---	---	---	---	---	---	---	書き込: NC 変数 x へのデータ [3]

1) DB120**x**、x = 0 ... 7 が変数 1 ... 8 に相当

2) 「NC 変数の読み取り/書き込み」ユーザーインタフェースの事前定義変数の場合のみ

3) DB9910 NC_DATA からの変数の場合のみ

注記

チャネル別変数

チャネル別変数を読み取り/書き込みする場合、ちょうど **1 つの**チャネルの変数のみ、ジョブでアドレス指定できます。

ドライブ別変数

ドライブ別変数を読み取り/書き込みする場合、ちょうど **1 つの** SERVO ドライブオブジェクトの変数のみ、ジョブでアドレス指定できます。SERVO ドライブオブジェクトはNC の機械軸に割り当ててください。行インデックスは論理的なドライブ番号に対応します。

異常時

異常の場合に、異なるドライブオブジェクトから、またはチャネルとドライブオブジェクトから同時に変数を読み取り/書き込みすると、次のエラーメッセージが出力されます。

DB1200.DBX3000.1 == 1 (異常が発生)

例:「ロケーションタイプ」の変数を 4 番目の変数として読み取り

```
DB1203.DBB1000:7
DB1203.DBB1001: -
DB1203.DBW1002: <ロケーション番号>
DB1203.DBW1004: <マガジン番号>
DB1203.DBW1006: -
DB1203.DBD1008: -
```

例:変数を 4 番目の変数として書き込み

データ項目を NC に書き込みには、その値をダブルワード DBD1008 に入力してください。

```
DB1203.DBB1000: <変数インデックス>
DB1203.DBB1001: <領域番号>
DB1203.DBW1002: <列インデックス>
DB1203.DBW1004: <行インデックス>
DB1203.DBW1006: -
DB1203.DBD1008: <値>
```

15.6.1.3 ジョブの管理: ジョブの開始

以下のデータは、ユーザーが**グローバルジョブインタフェース**に書き込んでください。

DB120x [1]	**NC データの読み取り/書き込み(PLC → NC)**							
バイト	**ビット 7**	**ビット 6**	**ビット 5**	**ビット 4**	**ビット 3**	**ビット 2**	**ビット 1**	**ビット 0**
DBB0							ジョブタイプ	ジョブ開始
DBB1	ジョブ内で処理する変数の数							
1) DB120**x**、x = 0 ... 7 は変数 1 ... 8 に相当								

ジョブタイプ

- 変数の**読み取り** DB1200.DBX0.1 = **0**
- 変数の**書き込み** DB1200.DBX0.1 = **1**

ジョブを開始します。

指定された数の変数を使用してジョブを開始するよう、開始信号を設定してください。

DB1200.DBX0.0 = **1**

注記

前のジョブが完了していないと、新しいジョブを開始できません。「ジョブの管理: ジョブの終了の待機 (ページ 1409)」の章を参照してください。

ジョブの実行には複数の PLC 周期が必要なことがあり、負荷率によって異なってきます。そのため、このファンクションの時間は定義できません。

15.6.1.4 ジョブの管理: ジョブの終了の待機

ジョブの終了は常に NC により、ジョブ**全体**に対して**グローバル結果インターフェース**で通知が返されます。この信号は、PLC ユーザーだけが読み取ることができます。

DB120x [1]	**NC データの読み取り/書き込み(NC → PLC)**							
バイト	**ビット 7**	**ビット 6**	**ビット 5**	**ビット 4**	**ビット 3**	**ビット 2**	**ビット 1**	**ビット 0**
DBB2000							ジョブ内の異常	ジョブの完了
1) DB120**x**、x = 0 ... 7 は変数 1 ... 8 に相当								

ジョブの状態

- エラーなしのジョブの終了
  DB1200.DBX2000.0 == **1** と DB1200.DBX2000.1 == **0**
- エラーありのジョブの終了
  DB1200.DBX2000.0 == **1** と DB1200.DBX2000.1 == **1**

考えられる異常の原因

- 変数の数(DB1200.DBB1)が有効な範囲外
- 変数インデックス(DB1200.DBB1000)が有効な範囲外
- 別々のサーボドライブオブジェクトの NC データを同時に読み取り/書き込み

15.6.1.5 ジョブの管理: ジョブの完了

必要条件

ジョブを完了するには、ジョブの終了後にジョブの開始信号を PLC ユーザープログラムからリセットしてください。

DB1200.DBX0.0 = **0**

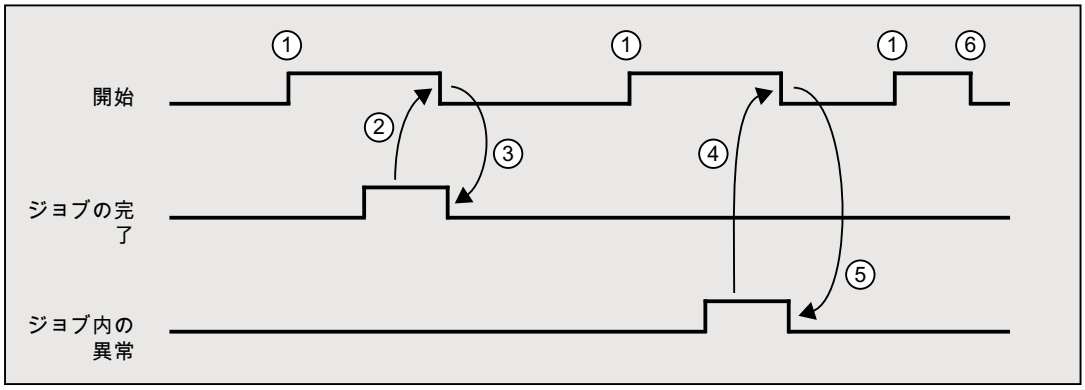
フィードバック信号

フィードバックとして、NC は状態信号をリセットします。

- DB1200.DBX2000.0 == **0**
- DB1200.DBX2000.1 == **0**

これで、ジョブが完了しました。

15.6.1.6 ジョブの管理:フローチャート



① ジョブの開始:
DB1200.DBX0.0 (開始) = 1

② ジョブの終了待ち:
DB1200.DBX2000.0 (ジョブの完了) == **1** と
DB1200.DBX2000.1 (ジョブのエラー) == **0**
⇒ ジョブのリセット要求
DB1200.DBX0.0 = 0 (開始)

③ DB1200.DBX0 0 == 0 (開始)では、ジョブが PLC 基本プログラムにより完了します。
DB1200.DBX2000.0 (ジョブの完了) = 0

④ ジョブの終了待ち:
DB1200.DBX2000.0 (ジョブの完了) == **0** と
DB1200.DBX2000.1 (ジョブのエラー) == **1**
⇒ エラー処理の実行
⇒ ジョブのリセット要求
DB1200.DBX0.0 (開始) = 0

⑤ DB1200.DBX0.0 == 0 (開始)では、ジョブが PLC 基本プログラムにより完了します。
DB1200.DBX2000.1 (ジョブのエラー) = 0

⑥ PLC 基本プログラムによりジョブの終了が通知される前に DB1200.DBX0.0 (開始)がリセットされた場合、ジョブは その他のフィードバックなしで実行されます。

15.6.1.7 ジョブの評価

変数別結果インタフェースは、ジョブで処理された変数毎に評価してください。

DB120x[1]	NC サービス(NC → PLC)							
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB3000							異常発生	変数有効
DBB3001	アクセス結果(後述の「アクセス結果」を参照)							
DBW3002	---							
DBD3004	読み取り:NC 変数 x からのデータ(変数のデータタイプ:1…4 バイト)[2]							
DBD3008	---							
DBD3012	---							
DBD3016	読み取り:NC 変数 x からのデータ(REAL)[3]							
DBD3020	読み取り:NC 変数 x からのデータ(DWORD / DINT)[3]							
DBW3024	読み取り:NCK 変数 x からのデータ(WORD / INT)[3]							
DBB3026	読み取り:NCK 変数 x からのデータ(BYTE)[3]							
DBB3027	---	---	---	---	---	---	---	読み取り: NC 変数 x からのデータ [3]

1) DB120x、x = 0 ... 7 が変数 1 ... 8 に相当

2) 「NC 変数の読み取り/書き込み」ユーザーインタフェースの事前定義変数の場合のみ

3) DB9910 NC_DATA からの変数の場合のみ

注記

チャネル別変数

チャネル別変数を読み取り/書き込みする場合、ちょうど **1 つの**チャネルの変数のみ、ジョブでアドレス指定できます。

ドライブ別変数

ドライブ別変数を読み取り/書き込みする場合、ちょうど **1 つの** SERVO ドライブオブジェクトの変数のみ、ジョブでアドレス指定できます。SERVO ドライブオブジェクトはNC の機械軸に割り当ててください。行インデックスは論理的なドライブ番号に対応します。

異常時

異常の場合に、異なるドライブオブジェクトから、またはチャネルとドライブオブジェクトから同時に変数を読み取り/書き込みすると、次のエラーメッセージが出力されます。

DB1200.DBX3000.1 == 1 (異常が発生)

アクセス結果

NC 変数

DBB3001	
規格値	**意味**
0	異常なし
3	オブジェクトアクセスが許可されていません
5	無効なアドレス
10	オブジェクトが存在しません

ドライブ別変数

ドライブ別変数(DB1200.DBX3000.1 == 1)の読み取り/書き込み中に異常が発生した場合は、アクセス結果に異常番号が表示されます。この番号は、PROFIdrive プロファイルで定義された異常番号に基づいています。

DBB3001	
値	**意味**
X	<PROFIdrive プロファイルの異常番号> + 20_H または 36_D

アクセス結果の意味の特定:

1. PROFIdrive プロファイルの異常番号の計算
   <PROFIdrive プロファイルの異常番号> =アクセスの結果- 20_H または 36_D
2. PROFIdrive プロファイルの異常番号の意味の特定
   PROFIdrive プロファイルの異常番号の説明は、以下の資料にあります。
   参照先
   『機能マニュアル、SINAMICS S120 ドライブの機能』、「通信」の章の「PROFIdrive による通信」の「非周期通信」の「要求と応答の構造」の「パラメータ応答の異常値」の段落

例:ジョブの状態

エラーなしのジョブ

- DB1200.DBX3000.0 == 1 (有効な変数) と
- DB1200.DBX3000.1 == 0 (エラーの発生なし)

結果:

- DB1200.DBB3001 == 0 (アクセス結果:「異常なし」)
- DB1200.DBD3004 == <読み取り値>

異常ありのジョブ

- DB1200.DBX3000.0 == 0 (無効な変数) と
- DB1200.DBX3000.1 == 1 (異常が発生)

結果:

- DB1200.DBB3001:考えられる異常の原因については、前述の「アクセス結果」を参照してください。

15.6.1.8 操作可能な変数

以下の変数が使用可能です。

変数	意味
cuttEdgeParam (ページ 1415)	ある工具に対する D 番号による補正値パラメータと刃先リスト
numCuttEdgeParams (ページ 1416)	刃先の P 要素の数
linShift (ページ 1416)	設定可能な原点オフセットの座標移動
numMachAxes (ページ 1417)	最大既存チャネル番号

変数	意味
RPA (ページ 1418)	R 変数
actLineNumber (ページ 1418)	現在の NC ブロックの行番号
$TC_MPPx (ページ 1419)	マガジンロケーションデータ
r0078[1] (ページ 1420)	トルク実績値、トルク生成
r0079[1] (ページ 1421)	速度コントローラの出力でのトルク指令値
r0081 (ページ 1421)	トルク負荷率(パーセント)
r0082[1] (ページ 1422)	有効電力実績値表示
TEMP_COMP_x (ページ 1422)	熱変位補正データ

変数「cuttEdgeParam」

ある工具に対する D 番号による補正値パラメータと刃先リスト

個々のパラメータの意味は、当該の工具のタイプによって異なります。現在、25 個のパラメータが工具刃先ごとに予約されています(ただし、値と一緒にロードされているのはその一部だけです)。将来の拡張のための柔軟性を残しておけるよう、25 個のパラメータの固定値を計算に使用するのではなく、変数値'numCuttEdgeParams' (変数インデックス 2)を使用することをお勧めします。

工具パラメータの詳しい説明については、「W1:工具補正 (ページ 1767)」の章を参照してください。

アドレス	値/意味
DB120x.DBB1000	1
DB120x.DBB1001	-
DB120x.DBW1002	(刃先番号 - 1) * numCuttEdgeParams + ParameterNr (WORD)
DB120x.DBW1004	T 番号(1...32000) (WORD)
DB120x.DBD1008	**書き込み**:NC 変数 x へのデータ(データタイプ:REAL)
DB120x.DBW3004	**読み取り**:NC 変数 x へのデータ(データタイプ:REAL)

変数「numCuttEdgeParams」

刃先の P 要素の数

アドレス	値/意味
DB120x.DBB1000	2
DB120x.DBB1001	-
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	読み取り:NCK 変数 x からのデータ(データタイプ:WORD)

変数「linShift」

設定可能なワークオフセット(チャネル別の設定可能なフレーム)の座標移動

MD18601 MM_NUM_GLOBAL_USER_FRAMES > 0 の場合にのみ、変数が存在します。

次のフレームインデックスが使用可能です。

インデックス	意味
0	ACTFRAME = 現在得られるワークオフセット
1	IFRAME = 現在の設定可能ワークオフセット
2	PFRAME = 現在のプログラマブルゼロオフセット
3	EXTFRAME = 現在の外部ゼロオフセット
4	TOTFRAME = 現在の合計ゼロオフセット = ACTFRAME と EXTFRAME の合計
5	ACTBFRAME = 現在の合計基本フレーム
6	SETFRAME = 現在の 1 番目のシステムフレーム(PRESET、接触計測)
7	EXTFRAME = 現在の 2 番目のシステムフレーム(PRESET、接触計測)
8	PARTFRAME = 現在の 3 番目のシステムフレーム(旋回可能な工具ホルダ付きの TCARR および PAROT)
9	TOOLFRAME = 現在の 4 番目のシステムフレーム(TOROT および TOFRAME)

インデックス	意味
10	MEASFRAME = ワークおよび工具計測用の結果フレーム
11	WPFRAME = 現在の 5 番目のシステムフレーム(ワークレファレンス点)
12	CYCFRAME = 現在の 6 番目のシステムフレーム(サイクル)

アドレス	値/意味
DB120x.DBB1000	3
DB120x.DBB1001	-
DB120x.DBW1002	フレームインデックス* numMachAxes (ページ 1417) +軸番号
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	**読み取り**:NC 変数 x へのデータ(データタイプ:REAL)

変数「numMachAxes」

最大既存チャネル番号

チャネル軸**ギャップ**がない場合は、変数の値が、チャネルで使用可能な軸の数にもなります。

アドレス	値/意味
DB120x.DBB1000	4
DB120x.DBB1001	-
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	**読み取り**:NC 変数 x へのデータ(データタイプ:WORD)

変数「rpa」

R 変数

アドレス	値/意味
DB120x.DBB1000	5
DB120x.DBB1001	-
DB120x.DBW1002	R 番号 + 1
DB120x.DBW1004	-
DB120x.DBD1008	書き込み:NC 変数 x へのデータ(データタイプ:REAL)
DB120x.DBW3004	読み取り:NC 変数 x へのデータ(データタイプ:REAL)

変数「actLineNumber」

現在の NC ブロックの行番号

No.	意味
≥ 1	現在の NC ブロックの行番号
0	プログラムが起動していないため、使用可能な行番号はありません。
-1	使用可能な行番号がありません。異常
-2	使用可能な行番号がありません。DISPLOF によるブロック表示のマスクが有効

アドレス	値/意味
DB120x.DBB1000	6
DB120x.DBB1001	-
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	読み取り:NC 変数 x へのデータ(データタイプ:INT)

工具管理機能:マガジンロケーションデータ

ロケーションタイプ($TC_MPP2)

アドレス	値/意味		
DB120x.DBB1000	7		
DB120x.DBB1001	-		
DB120x.DBW1002	ロケーション番号(1 … 31999)		
DB120x.DBW1004	マガジン番号(1 … 9999)		
DB120x.DBD1008	-		
DB120x.DBW3004	読み取り:NC 変数の値(データタイプ:WORD)		
		値	意味
		> 0	実ロケーションのロケーションタイプ
		0	「match all」(バッファ)
		9999	未定義(実ロケーションなし)

ロケーション状態($TC_MPP4)

アドレス	値/意味
DB120x.DBB1000	8
DB120x.DBB1001	-
DB120x.DBW1002	ロケーション番号(1 … 31999)
DB120x.DBW1004	マガジン番号(1 … 9999)
DB120x.DBD1008	-

アドレス	値/意味		
DB120x.DBW3004	読み取り:NC 変数の値(データタイプ:WORD)		
		値	意味
		1	無効
		2	空き(<>占有)
		4	バッファ内の工具用に予約
		8	ロードする工具用に予約
		16	左半分のロケーションで占有
		32	右半分のロケーションで占有
		64	上半分のロケーションで占有
		128	下半分のロケーションで占有

このロケーションの工具の T 番号($TC_MPP6)

アドレス	値/意味
DB120x.DBB1000	9
DB120x.DBB1001	-
DB120x.DBW1002	ロケーション番号(1 … 31999)
DB120x.DBW1004	マガジン番号(1 … 9999)
DB120x.DBD1008	-
DB120x.DBW3004	読み取り:このロケーションの工具の T 番号(データタイプ:WORD)

変数 r0078[1]

- ドライブオブジェクト:SERVO、SERVO_AC、SERVO_I_AC
- CO:トルクを発生している電流の現在値[A_{rms}]
- インデックス[1]:表示フィルタあり (表示フィルタ設定 = p0045)

アドレス	値/意味
DB120x.DBB1000	10
DB120x.DBB1001	ドライブモジュール番号

アドレス	値/意味
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	読み取り:NC 変数 x へのデータ(データタイプ:REAL)

変数 r0079[1]

- ドライブオブジェクト:SERVO、SERVO_AC、SERVO_I_AC
- CO:速度コントローラ出力でのトルク指令値(クロックサイクル補間前) [Nm]
- インデックス[1]:表示フィルタあり (表示フィルタ設定 = p0045)

アドレス	値/意味
DB120x.DBB1000	11
DB120x.DBB1001	ドライブモジュール番号
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	読み取り:NC 変数 x へのデータ(データタイプ:REAL)

変数 r0081

- ドライブオブジェクト:SERVO、SERVO_AC、SERVO_I_AC
- CO:トルク負荷率(パーセント)

アドレス	値/意味
DB120x.DBB1000	12
DB120x.DBB1001	ドライブモジュール番号

アドレス	値/意味
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	読み取り:NC 変数 x へのデータ(データタイプ:REAL)

変数 r0082[1]

- ドライブオブジェクト:SERVO、SERVO_AC、SERVO_I_AC
- CO:有効電力の現在値[kW]
- インデックス[1]:表示フィルタあり (表示フィルタ設定 = p0045)

アドレス	値/意味
DB120x.DBB1000	13
DB120x.DBB1001	ドライブモジュール番号
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	-
DB120x.DBW3004	読み取り:NC 変数 x へのデータ(データタイプ:REAL)

熱変位補正

変数「TEMP_COMP_ABS_VALUE」(SD43900)

位置に依存しない熱変位補正値

アドレス	値/意味
DB120x.DBB1000	14
DB120x.DBB1001	軸の番号(1、2、...)

アドレス	値/意味
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	書き込み:NC 変数 x へのデータ(データタイプ:REAL)
DB120x.DBW3004	読み取り:NC 変数 x へのデータ(データタイプ:REAL)

変数「TEMP_COMP_SLOPE」(SD43910)

位置に依存する熱変位補正の傾き

アドレス	値/意味
DB120x.DBB1000	15
DB120x.DBB1001	軸の番号(1、2、...)
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	**書き込み**:NC 変数 x へのデータ(データタイプ:REAL)
DB120x.DBW3004	**読み取り**:NC 変数 x へのデータ(データタイプ:REAL)

変数「TEMP_COMP_REF_POSITION」(SD43920)

位置に依存した熱変位補正の基準位置

アドレス	値/意味
DB120x.DBB1000	16
DB120x.DBB1001	軸の番号(1、2、...)
DB120x.DBW1002	-
DB120x.DBW1004	-
DB120x.DBD1008	**書き込み**:NC 変数 x へのデータ(データタイプ:REAL)
DB120x.DBW3004	**読み取り**:NC 変数 x へのデータ(データタイプ:REAL)

変数「TOOL_TEMP_COMP」(SD42960[...])

工具に対応する熱変位補正

アドレス	値/意味
DB120x.DBB1000	17
DB120x.DBB1001	-
DB120x.DBW1002	インデックス + 1 (1、2、3)
DB120x.DBW1004	-
DB120x.DBD1008	**書き込み**:NC 変数 x へのデータ(データタイプ:REAL)
DB120x.DBW3004	**読み取り**:NC 変数 x へのデータ(データタイプ:REAL)

15.6.1.9 選択された NC 変数の指定

選択された NC 変数は、DB9910 データブロック(選択された NC 変数)を介して指定します。このデータブロックの長さは、変数リストで選択された NC 変数の数に依存します。変数リストには、選択された NC 変数を最大 42 個含めることができます。DB9910 データブロックには、NC 変数ごとに変数のアドレス指定用のデータとデータタイプ変換用のデータが含まれています。

DB9910	選択された NC 変数、読み取り (PLC → NC)							
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB0	変数インデックス**変数 1**							
DBB1	構文 ID							
DBB2	区分							
DBB3	単位							
DBW4	列インデックス							
DBW6	行インデックス							
DBB8	ブロック							
DBB9	行数							
DBB10	タイプ							
DBB11	長さ							

DB9910	選択された NC 変数、読み取り (PLC → NC)
....	.....
DBB492	変数インデックス**変数 42**
DBB493	構文 ID
DBB494	区分
DBB495	単位
DBW496	列インデックス
DBW498	行インデックス
DBB500	ブロック
DBB501	行数
DBB502	タイプ
DBB503	長さ

変数インデックス

変数インデックスは、NC 変数の名前を表します。変数インデックスは、開始値 100 とリスト内の NC 変数のオフセット(0 ～ 41)で構成されます。

変数インデックスは、DB120x RW_NCDx ユーザーインタフェースで DBB1000 A_VarIdx として入力されます。

コメントには、スペースを分離記号として使用して、次のデータレコードが含まれています。

- 区分
- ブロック
- VariablenName
- VarType
- 列
- VarAnzByte

拡張ユーザーインタフェース

データブロックでのデータテストとは、変数の値をデータブロック内の同じタイプのアドレスに書き込むことができることを意味します。たとえば、REAL 値は、(たとえば、MOV_R で) REAL アドレスにのみ書き込むことができます。「NC 変数の読み取り/書き込み」ユーザーインタフェースには現在、書き込まれる値について DWORD タイプのアドレス(DBD1008)が 1 つしか含まれていません。つまり、REAL 値は、一時変数、フラグまたはアキュムレータを介してしか書き込むことができません。同じことは読み取り(DBD3004)にも当てはまります。そのため、「NC 変数の読み取り/書き込み」ユーザーインタフェースが拡張されます。読み取りおよび書き込みについて、次のタイプごとに 1 つのアドレスが追加されます。REAL、DWORD/DINT、WORD/INT、BYTE および BOOL (DBD1016 … DBB1027 または DBD3016 … DBB3027)。これらの新しいアドレスは、PLC ファームウェアによって、NC 変数エディタで選択され、コンパイル時に DB9910 NC_DATA に入力された変数(変数インデックス≥ 100)にのみ使用されます。「NC 変数の読み取り/書き込み」ユーザーインタフェースで現在定義されている NC 変数は、引き続き古いアドレス(DBD1008 または DBD3004)を使用します。

15.6.2 プログラムインスタンスサービス(PI サービス)

15.6.2.1 ジョブの指定

PI サービスは、**ジョブインターフェース**(オフセット 4000 からの DB1200)を介して指定します。

DB1200			**PI サービス[r/w]**					
			PLC → NC インタフェース					
バイト	**ビット 7**	**ビット 6**	**ビット 5**	**ビット 4**	**ビット 3**	**ビット 2**	**ビット 1**	**ビット 0**
DBB 4000	-	-	-	-	-	-	-	開始 [1]
DBB 4001	PI インデックス [2]							
DBB 4002	-							
DBB 4003	-							
DBW 4004	PI パラメータ 1 [3]							
DBW 4006	PI パラメータ 2							
DBW 4008	PI パラメータ 3							

DB1200	PI サービス[r/w]
DBW 4010	PI パラメータ 4
DBW 4012	PI パラメータ 5
DBW 4014	PI パラメータ 6
DBW 4016	PI パラメータ 7
DBW 4018	PI パラメータ 8
DBW 4020	PI パラメータ 9
DBW 4022	PI パラメータ 10
1) DB1200.DBX4000.1、開始:DBX4000.1 = 1 ⇒PI サービスの開始; DBX4000.1 = 0 ⇒ PI サービス完了 2) DB1200.DBB4001、PI インデックス:特定の PI サービスを指定 3) DB1200.DBW4004 ...、PI パラメータ:PI 固有のパラメータ	

PI サービスの概要:

- PI サービス ASUB (ページ 1428)
- PI サービス LOGOUT (ページ 1430)
- PI サービス DATA_SAVE (ページ 1431)
- PI サービス TMMVTL (ページ 1431)

15.6.2.2 ジョブのフィードバック

PLC は、開始された PI サービスが正常に終了したかどうかに関するフィードバックを、**結果インターフェース** (オフセット 5000 からの DB1200)で提供します。

ジョブの終了が、以下の信号を使用して通知されます。

- DB1200.DBX5000.0
- DB1200.DBX5000.1

この信号は PLC によって書き込まれるため、ユーザーだけが読み取ることができます。

ジョブは、ユーザーによる「開始」信号のリセット(DB1200.DBX4000.1)の直後に完了しています。その後、状態信号 DB1200.DBX5000.0 および.1 が 0 に設定されます。

DB1200			PI サービス[r]					
			NCK → PLC インタフェース					
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB 5000	-	-	-	-	-	-	ジョブの異常 2)	ジョブが完了 1)
DBB 5001	-							
DBB 5002	-							
1) DB1200.DBX5000.0、ジョブの状態:DBX5000.0 == 1⇒ジョブが完了 2) DB1200.DBX5000.1、異常状態:DBX5000.1 == 0 ⇒異常なし; DBX5000.1 == 1 ⇒異常								

考えられる異常の原因

- 設定された PI サービスのインデックス(DB1200.DBB4001)が有効範囲外
- パラメータの異常

15.6.2.3 PI サービス ASUB

割り込み

注記

チャネルと割り込みの関係

最大 **1** つのチャネルを備えたコントロールシステムでは、**2** つの割り込みを、最大 **2** つのチャネルを備えたコントロールシステムでは、**4** つの割り込みを使用できます。

すべての割り込み番号には、PLC からの「ASUB」PI サービスで割り込みプログラム(ASUB)を割り当てることができます。そして、関連する割り込みが発行されると、NC で割り込みプログラムが実行されます。関連する割り込みプログラムは、NC の工作機械メーカディレクトリ(_N_CMA_DIR)に、以下のプログラム名称で存在する必要があります。

割り込み	プログラム名称
1	PLCASUP1_SPF
2	PLCASUP2_SPF

割り込み	プログラム名称
3	PLCASUP3_SPF
4	PLCASUP4_SPF

割り込みプログラムが用意されていない場合は、作成してください。そして、NC で NC リセット(電源投入)を実行してください。

PI サービス「ASUP」は、制御装置の電源投入後の割り込みの割り当てのたびに 1 回、実行してください。割り込みプログラムへの割り込みの割り当ては、制御装置が再度電源投入されるまで保持されます。

ジョブの指定

PI サービス ASUB		
アドレス	意味	有効な値
DB1200.DBW4001	PI インデックス 1)	1, 2, 13, 14
DB1200.DBW4004	パラメータ 1:高速退避 2)	0 (FALSE)、1 (TRUE)
DB1200.DBW4006	パラメータ 2:BLSYNC 3)	0 (FALSE)、1 (TRUE)
DB1200.DBW4008	パラメータ 3:チャネルインデックス 4)	0, 1
DB1200.DBW4010	パラメータ 4:割り込み優先度 5)	0, 1, 2, 3, 4

1)

- PI インデックス= 1:割り込み 1⇒_N_CMA_DIR/PLCASUP1_SPF、初期設定として設定した割り込み優先度:1
- PI インデックス= 2:割り込み 2⇒_N_CMA_DIR/PLCASUP2_SPF、初期設定として設定した割り込み優先度:2
- PI インデックス= 13:割り込み 3⇒_N_CMA_DIR/PLCASUP3_SPF、初期設定として設定した割り込み優先度:3
- PI インデックス= 14:割り込み 4⇒_N_CMA_DIR/PLCASUP4_SPF、初期設定として設定した割り込み優先度:4

2) 割り込みが開始された後、LIFTFAST によって高速リトラクトが発生します(高速リトラクト)。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。

参照先

『機能マニュアル、応用機能』、「R3:拡張された停止状態および後退」の章の「制御装置で管理された ESR」の「後退」

3BLSYNC は、割り込みが開始された後、まず現在のブロックが実行され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。

PI サービス ASUB		
アドレス	意味	有効な値
4) 0→チャネル 1、1→チャネル 2		
5) 初期設定の割り込み優先度を使用する場合は、パラメータを値 0 に設定してください。		

マシンデータ

- 最低の割り込み優先度
  以下のマシンデータでは、NC の最低の割り込み優先度を定義します。この優先度に関連する割り込みが処理されます。マシンデータで指定された優先度より低い優先度の割り込みは、コントロールシステムで処理されません。
  MD11604 $MN_ASUP_START_PRIO_LEVEL

必要条件

チャネル状態

PI サービス「ASUB」は、このサービスが要求されているチャネルが「リセット」状態の場合にのみ実行できます。

ProgEvent「電源投入」

イベント起動プログラム呼び出し(ProgEvent)に対して「電源投入」を開始イベントとして設定(MD20108 $MC_PROG_EVENT_MASK)した場合、PI サービス「ASUB」を開始できるのは、ProgEvent プログラムの完了(PROG_EVENT_SPF または MD11620 $MN_PROG_EVENT_NAME = <user_prog_event_SPF>)後のみです。

下記も参照

ジョブの指定 (ページ 1426)
ジョブのフィードバック (ページ 1427)

15.6.2.4 PI サービス LOGOUT

機能

NC に最後に伝送されたパスワードがリセットされます。

ジョブの指定

PI サービス LOGOUT		
アドレス	説明	有効な値
DB1200.DBW4001	PI インデックス	3 (リセットパスワード)

15.6.2.5 PI サービス DATA_SAVE

機能

システムのコンパクトフラッシュカードに現在の NC の状態を保存します。

注記

電源投入

次回のコントローラの電源投入時に、「起動メニュー」で「保存されたユーザーデータを再ロード」することにより、保存された状態を NC にロードできます。

ジョブの指定

PI サービス DATA_SAVE		
アドレス	説明	有効な値
DB1200.DBW4001	PI インデックス	4

参照先

『SINUMERIK 828D 試運転マニュアル』、「CNC 試運転」の章の「納入と必要条件の範囲」の「制御装置の起動」

15.6.2.6 PI サービス TMMVTL

機能

PI サービス TMMVTL を使用して、工具を再配置するジョブを PLC から要求することができます。その結果、PI サービス、ツールマネージャがターゲットマガジンで空きロケーション検索を実行して、PI サービスで指定された工具(工具番号またはソースロケーション番号/ソースマガジン番号)を検索します。そして、PLC がジョブを受信して、DB41xx.DBB0 を介して工具を再配置します。

ジョブの指定

アドレス	説明	有効な値
DB1200.DBW4001	PI インデックス	5
DB1200.DBW4004	パラメータ 1:工具番号 1)	-1, 1 … 31999
DB1200.DBW4006	パラメータ 2:ソースロケーション番号 1)	-1, 1 … 31999
DB1200.DBW4008	パラメータ 3:ソースマガジン番号 1)	-1, 1 … 9999
DB1200.DBW4010	パラメータ 4:ターゲットロケーション番号 2)	-1, 1 … 31999
DB1200.DBW4012	パラメータ 5:ターゲットマガジン番号 3)	-1, 1 … 9999

1) 工具は、以下を使用して指定することもできます。

- 工具番号(T 番号)
- ソースロケーション番号およびソースマガジン番号

他のバージョンの未使用のパラメータには、-1 を入力してください。

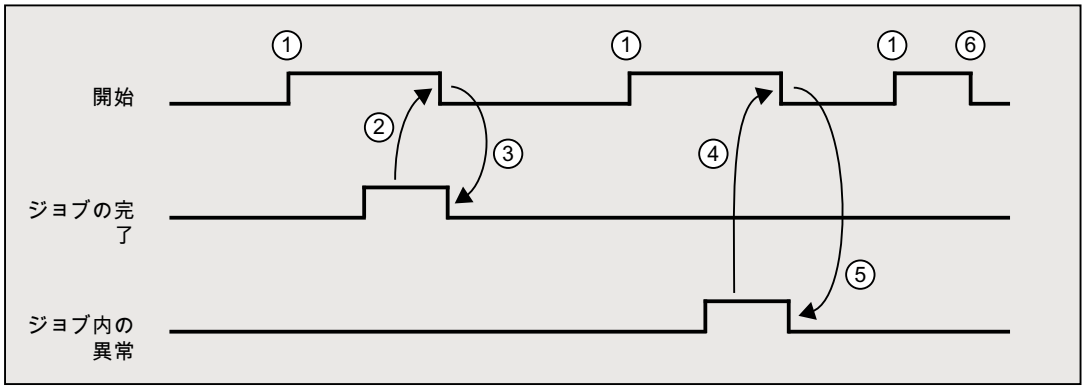
2) ターゲットロケーション番号= -1 の場合は、選択されている検索方法に従って、マガジン全体で工具の空きロケーションが検索されます。ターゲットロケーションが指定されている場合、指定されているターゲットロケーション番号のロケーションが特定の工具に対して空いていて適切であるかどうかがチェックされます。

3) ターゲットマガジン番号= -1 の場合は、$TC_MDP2 から取得した割り当てに対応して、工具がバッファ内で検索されます。

用途例

- 工具を返すのにバッファを使用する場合(例えば、Toolboy および/またはシフタ)、非同期の戻り移動時にマガジン内での明示的な空きロケーション検索が必要です。この場合、PLC は、元のロケーションに注意する必要はなく、PI サービス TMMVTL が適切なロケーションを検索します。
- 工具は、背面マガジンから前面マガジンに移動します。

15.6.2.7 PI サービス: タイミングチャート



① ユーザーが信号「起動」を設定し、ジョブの実行が開始されます。

② PLC ファームウェアが「ジョブの完了」通知後に、ユーザーが信号「起動」をリセットします。

③ 信号「起動」のリセットによって、PLC ファームウェアが信号「ジョブの完了」をリセットします。

④ PLC ファームウェアが「ジョブ内の異常」通知後に、ユーザーが信号「起動」をリセットします。

⑤ 信号「起動」のリセットによって、PLC ファームウェアが信号「ジョブ内の異常」をリセットします。

⑥ 信号「ジョブの完了」または「ジョブ内の異常」のどちらかが受信される前に、ユーザーが誤って信号「起動」をリセットした場合、このジョブの結果信号は更新されません。 ただし、ジョブは実行されます。

15.6.3 PLC ユーザーアラーム

15.6.3.1 ユーザーインタフェース

注記

これ以降では、ユーザー「アラーム」という名称が使用されていますが、これは、関連する**解除条件**の入力時に**メッセージ**または**アラーム**が関係しているかどうかのみを定義しています(「ユーザーアラームの設定 (ページ 1437)」を参照してください)。

DB1600 のユーザーインターフェースにより、異常メッセージと操作メッセージを HMI に表示するオプションが提供されます。

これには、以下の成果物が含まれます。

- ユーザーアラーム 700000 ～ 700247 と拡張ユーザーアラーム 701000 ～ 701999 の有効化。

  注記

  拡張 PLC ユーザーアラーム 701000 ～ 701999 を使用するには、以下の必要条件を満たさなければなりません。
  - 互換性モードが無効になっている。
  - データブロック DB9913 が PLC プロジェクトに含まれている(つまり、PLC プログラミングツールの[ライブラリ] > [特殊データブロック]で DB9913 が選択されており、コピー&ペースト/ダブルクリックで PLC プロジェクトに伝送されている)。

- ユーザーアラーム 700000 ～ 700247 および 701000 ～ 701247 に追加の数値パラメータを指定できる。
- ユーザーアラームの無効化と確認。
- システム応答の評価がユーザーアラームによって開始される。

ファームウェアが入力された信号を確認し、この信号を表示先の HMI にアラームおよびメッセージの発生／解除として送信します。HMI が異常テキストを管理します。

15.6.3.2 ユーザーアラームの起動インタフェース

各ユーザーアラームは、割り当てられている起動ビットを使用して有効にします。これらのビットは、**起動インターフェース**で設定されます。

新規のユーザーアラームは、特定のビットの 0/1 エッジで有効にされます。

アラーム 700000 ～ 700247 の起動インターフェース

DB1600	**アラームの有効化[r/w] (PLC→HMI)**							
バイト	**ビット 7**	**ビット 6**	**ビット 5**	**ビット 4**	**ビット 3**	**ビット 2**	**ビット 1**	**ビット 0**
	有効にするアラーム番号							
DBB0	700007	700006	700005	700004	700003	700002	700001	700000
	有効にするアラーム番号							
DBB1	700015	700014	700013	700012	700011	700010	700009	700008
	有効にするアラーム番号							

DB1600	アラームの有効化[r/w] (PLC→HMI)							
DBB2	700023	700022	700021	700020	700019	700018	700017	700016
	有効にするアラーム番号							
DBB3	700031	700030	700029	700028	700027	700026	700025	700024
	有効にするアラーム番号							
DBB4	700039	700038	700037	700036	700035	700034	700033	700032
	有効にするアラーム番号							
DBB5	700047	700046	700045	700044	700043	700042	700041	700040
…	….							
	有効にするアラーム番号							
DBB30	700247	700246	700245	700244	700243	700242	700241	700240

アラーム 701000 ～ 701999 の起動インターフェース

DB1600	アラームの有効化[r/w] (PLC→HMI)							
Byte	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
	有効にするアラーム番号							
DBB4000	701007	701006	701005	701004	701003	701002	701001	701000
	有効にするアラーム番号							
DBB4001	701015	701014	701013	701012	701011	701010	701009	701008
	有効にするアラーム番号							
DBB4002	701023	701022	701021	701020	701019	701018	701017	701016
	有効にするアラーム番号							
DBB4003	701031	701030	701029	701028	701027	701026	701025	701024
	有効にするアラーム番号							
DBB4004	701039	701038	701037	701036	701035	701034	701033	701032
	有効にするアラーム番号							
DBB4005	701047	701046	701045	701044	701043	701042	701041	701040
…	….							
	有効にするアラーム番号							
DBB4124	701999	701998	701997	701996	701005	701994	701993	701992

15.6.3.3 ユーザーアラームの変数インタフェース

ユーザーアラーム 700000 ～ 700247 および 701000 ～ 701247 には、パラメータとして変数を指定できます。それぞれについて、ダブルワードが**変数インターフェース**で予約されています。したがって、有効なオフセットは 4 の倍数になります。

アラーム 700000 ～ 700247 の変数インターフェース

DB1600	アラーム[r32/w32]の変数(PLC→HMI)							
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBD1000	アラーム 700000 用変数							
DBD1004	アラーム 700001 用変数							
DBD1008	アラーム 700002 用変数							
	...							
DBD1980	アラーム 700245 用変数							
DBD1984	アラーム 700246 用変数							
DBD1988	アラーム 700247 用変数							

アラーム 701000 ～ 701247 の変数インターフェース

DB1600	アラーム[r32/w32]の変数(PLC→HMI)							
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBD5000	アラーム 701000 用変数							
DBD5004	アラーム 701001 用変数							
DBD5008	アラーム 701002 用変数							
	...							
DBD5980	アラーム 701245 用変数							
DBD5984	アラーム 701246 用変数							
DBD5988	アラーム 701247 用変数							

15.6.3.4 ユーザーアラームの設定

アラーム毎に以下の属性を指定することができます。

- アラーム応答:異常の発生時にコントロールシステムがどのように応答するか
- 解除条件:アラームの再解除または確認応答をおこなうために何をするべきか。解除条件は、アラームタイプと優先順位を同時に定義します。
- チャネル割り当て: アラームが割り当てられるチャネル。

ユーザーアラーム 700000 ～ 700247

ユーザーアラーム 700000 ～ 700247 は、マシンデータを介して設定します。

アラーム応答と解除条件

アラーム応答と解除条件は、コード化されたビットとしてマシンデータで指定します。

MD14516 $MN_USER_DATA_PLC_ALARM [x] = <アラーム応答と解除条件>

ここで、x =ユーザーアラーム番号- 700000、数値の範囲:0 ≤ x ≤ 247

<アラーム応答と解除条件>	
ビット	意味
アラーム応答	
0	NC スタート禁止
1	読み込み停止
2	すべての軸に対して送り無効
3	非常停止
4	PLC STOP
5	アラームログ
解除条件	
6	DB1600 DBX3000.0 による中断
7	電源投入
ビット x = 1:機能を起動します ビット x = 0:機能の選択を解除します	

チャンネルの割り付け

チャネルは、コード化されたビットとしてマシンデータで割り当てます。

MD14518 $MN_USER_DATA_PLC_ALARM_ASSIGN [x] = <チャネル割り当て>

ここで、x =ユーザーアラーム番号- 700000、数値の範囲:0 ≤ x ≤ 247

<チャネル割り当て>	
ビット	意味
0	有効範囲 NC チャネル 1
1	有効範囲 NC チャネル 2
ビット x = 1:チャネルが選択済み ビット x = 0:チャネルが選択解除済み	

ユーザーアラーム 701000 ～ 701999

ユーザーアラーム 701000 ～ 701999 は、データブロック DB9913(ALARM_INI)で設定します。各アラームには、2 バイトの設定データが必要です。1 バイトは、アラーム応答と解除条件の選択用、1 バイトはチャネル割り当て用です。この 2 バイトは、1 語としてグループ化されます。

DB9913	ユーザーアラーム 701000 ～ 701999 [r]の設定
DBW0	アラーム 701000
DBW2	アラーム 701001
DBW4	アラーム 701002
	...
DBW1998	アラーム 701999

ビット	意味
アラーム応答	
0	NC スタート禁止
1	読み込み停止
2	すべての軸に対して送り無効
3	非常停止
4	PLC 停止
5	アラームログ
解除条件	
6	DB1600 DBX3000.0 による中断

ビット	意味
7	電源投入
チャンネルの割り付け	
8	NC チャネル 1 の PLC ユーザーアラーム
9	NC チャネル 2 の PLC ユーザーアラーム
10	予備
11	予備
12	予備
13	予備
14	予備
15	予備

解除条件と優先度

ユーザーアラームの解除条件(したがって、タイプも)と優先度は、 ビット 6 および 7 を使用して設定します。

ビット 7	ビット 6	解除条件:	タイプ	優先順位
0	0	起動ビットのリセット	メッセージ	低
0	1	DB1600.DBX3000.0 での確認(「ユーザーアラームの応答インタフェース (ページ 1440)」を参照してください)	アラーム	中
1	0	電源投入	アラーム	高
1	1	予約済み(内部的な評価:ビット 7 = 1、ビット 6 = 0)	-	-

メッセージの表示

前述の表のユーザーアラームのいずれかでアラーム応答が有効でない(マシンデータのビット 0 ～ 4 = 0)場合、その定義は、システムに影響することなく、「表示メッセージ」と呼ばれるものが関係するということです。これは特に、対応するマシンデータの解除条件(マシンデータのビット 6 および 7)も評価されないことを示します。

15.6.3.5 有効なアラーム応答と解除条件のエクスポート

現在の**有効なアラーム応答** (つまり、実際の応答)、**有効な解除条件**、および**有効なチャネル割り当て**を、インターフェースからグローバルにエクスポートすることができます。

DB1600	有効なアラーム応答[r]							
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB2000	電源投入	DB1600 DBX3000.0 による中断		PLC 停止	非常停止	チャネル 1		
						すべての軸に対して送り速度が無効	読み込み停止	NC スタート禁止
DBB2001						チャネル 2		
						すべての軸に対して送り無効	読み込み停止	NC スタート禁止

最低でも 1 つの有効なアラームに対して、対応する応答または対応する解除条件が設定されている場合、1 つのビットが設定されます。この応答/解除条件が設定されている未処置のアラームがすべてなくなると、このビットはクリアされます。

15.6.3.6 ユーザーアラームの応答インタフェース

ユーザーアラームに応答するための必要条件は、対応する起動ビットがリセットされていることです。

- 解除条件が{0,0}のメッセージは、自動的に表示から消えます。
- 解除条件が{0,1}のアラームは、応答ビット *Ack* によって解除されます。
- 解除条件{1,0}のアラームは、応答ビットが設定されていても影響を受けず、電源投入によってのみ解除されます。

DB1600	アラームの応答[r/w]							
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB3000								応答
DBB3001								

DB1600	アラームの応答[r/w]
DBB3002	
DBB3003	

15.6.3.7 HMI へのインタフェース

PLC は、HMI に表示するための 8 つのメッセージまたはアラームを伝送できます。このメッセージまたはアラームは、発生順に表示されます。

さらにメッセージ/アラームが発生すると、最初の 7 つは HMI に残され、最後のメッセージまたはアラームが以下の規則に従って、新たに発生したものと置き換えられます。

- システムメッセージ/アラームを、ユーザーメッセージ/アラームと置き換えます。
- 優先度の高いメッセージ/アラームを、優先度の低いものと置き換えます。

最初の 7 つのメッセージ/アラームには問題の原因が定義されていて、その後に続くメッセージ/アラームは単に 2 次的なものに過ぎない可能性が高いため、最初の 7 つが表示に残されます。ただし、7 つのメッセージ/アラームのうちの 1 つまたは複数が応答された結果、クリアされると、受信された対応する数のアラーム/メッセージが HMI に移されます。

15.6.4 PLC 軸制御

15.6.4.1 一般情報

PLC は、ユーザーインターフェースのデータブロックを介して軸/主軸を制御できます。軸/主軸は DB 番号によって指定されます。

- **DB380x** PLC → NC インタフェース(軸/主軸へ)
- **DB390x** NC → PLC インタフェース(軸/主軸から)

ここで、軸インデックス x:0 ≤ x ≤ 最大軸インデックス、軸インデックス=軸番号- 1

次の制御がサポートされています。

- 軸位置決め
- 主軸の位置決め
- 主軸回転
- 主軸揺動
- 軸割り出し

参照先

- 『機能マニュアル、上級機能』;「位置決め軸(P2)と割り出し軸(T1)」
- 『機能マニュアル、基本機能』;「主軸(S1)」

必要条件

制御される軸が、PLC に割り当てられていなければなりません(PLC 軸)。軸は、ユーザーインタフェース「*軸入れ替え*」(DB3800.DBB8/DB3900.DBB8)を使用して、NC と PLC 間で交換できます。

機能の開始

各機能は、対応する「起動」信号の立ち上がりエッジで有効になります。この信号は、機能が肯定的または否定的に応答されるまで(*位置決め完了*= 「1」または*異常* = 「1」の使用などによって)、論理「1」のままでなければなりません。信号「軸位置決めが有効」 = 「1」は、機能が有効であり、出力信号が有効であることを示しています。

中止

開始信号のリセットによって機能を**中断**することは**できません**。他のインタフェース信号(軸別信号*残移動距離削除/主軸リセット*、DB380x DBX2.2 の使用)によってのみ、中断することができます。

軸インタフェースは、評価する必要がある軸の状態信号(*イグザクトストップ*、*移動指令*、→ DB390x など)を返します。

PLC 軸制御が呼び出された時(移動指令が出されている時)に、NC プログラムによって軸/主軸が移動している場合、機能は、この移動動作が終了しないと開始されません。この場合、異常コードは出力されません。

軸無効

軸無効が設定されている場合、PLC 軸制御で制御されている軸は移動しません。シミュレートされたフィードバック値のみが生成されます。(NC プログラミングと同様の動作)。

15.6.4.2 ユーザーインタフェース: PLC 軸としての NC 軸の準備

軸/主軸への要求信号(抜粋)

まず、軸/主軸を PLC から要求してください。

DB380x	軸/主軸への信号(PLC→NC) [r/w]							
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB8	PLC 軸/主軸を要求			このバイトが変更された場合の起動信号			NC 軸/主軸チャネルの割り当て	
							B	A

要求信号の変更(DB380x.DBX8.7 または 8.0) は、起動信号の 0→1 エッジ (DB380x.DBX8.4)を介して NC に通知してください。PLC 周期の後に、起動信号を再度リセットしてください。

軸/主軸からの状態信号(抜粋)

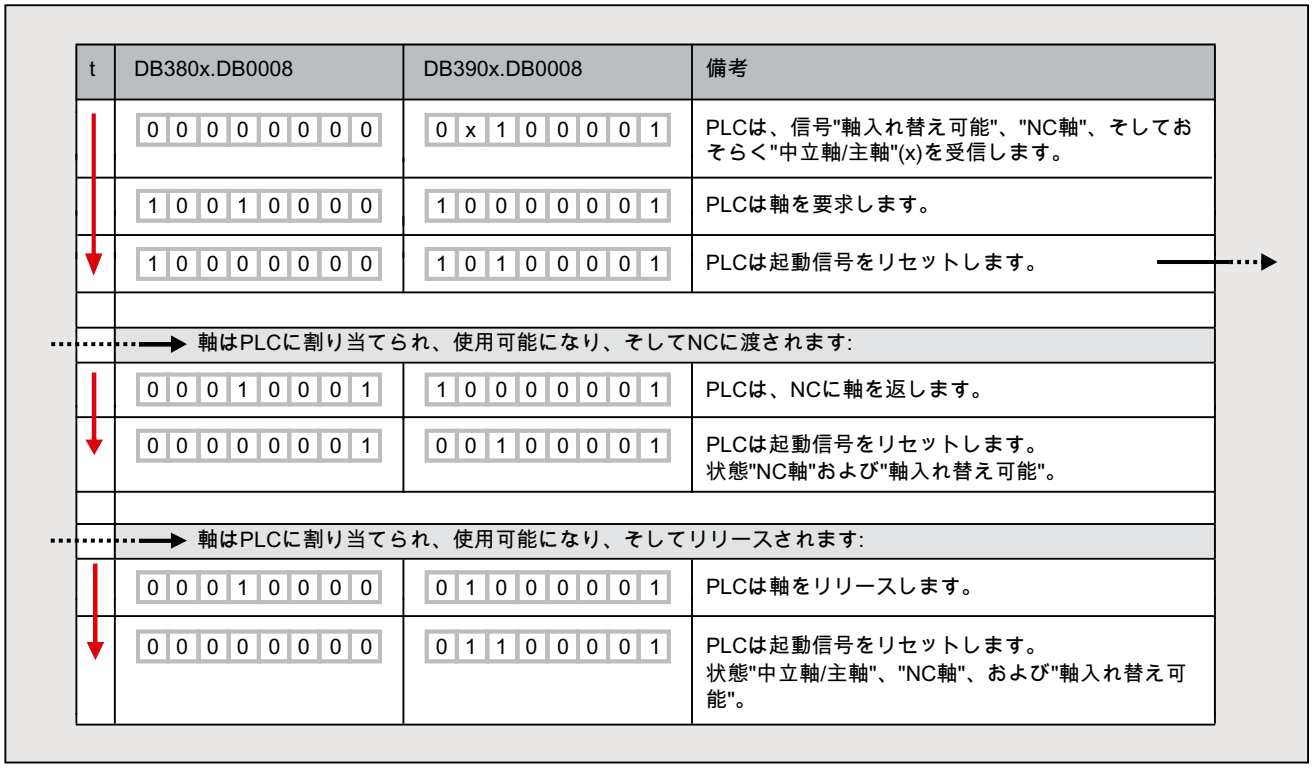
DB390x	軸/主軸からの信号(NC → PLC) [r]							
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB8	PLC 軸/主軸	中立軸/主軸	軸入れ替えが可能	PLC から要求された新しいタイプ			チャネルでの NC 軸/主軸の現在の割り当て	
							B	A

注記

シミュレーション

インターフェース信号を有効にするには、マシンデータ MD30350 $MA_SIMU_AX_VDI_OUTPUT を、シミュレーション時に必要な各軸に設定してください。

PLC 軸の要求と引き渡し



t	DB380x.DB0008	DB390x.DB0008	備考
	0 0 0 0 0 0 0 0	0 x 1 0 0 0 0 1	PLCは、信号"軸入れ替え可能"、"NC軸"、そしておそらく"中立軸/主軸"(x)を受信します。
	1 0 0 1 0 0 0 0	1 0 0 0 0 0 0 1	PLCは軸を要求します。
	1 0 0 0 0 0 0 0	1 0 1 0 0 0 0 1	PLCは起動信号をリセットします。
	軸はPLCに割り当てられ、使用可能になり、そしてNCに渡されます:		
	0 0 0 1 0 0 0 1	1 0 0 0 0 0 0 1	PLCは、NCに軸を返します。
	0 0 0 0 0 0 0 1	0 0 1 0 0 0 0 1	PLCは起動信号をリセットします。 状態"NC軸"および"軸入れ替え可能"。
	軸はPLCに割り当てられ、使用可能になり、そしてリリースされます:		
	0 0 0 1 0 0 0 0	0 1 0 0 0 0 0 1	PLCは軸をリリースします。
	0 0 0 0 0 0 0 0	0 1 1 0 0 0 0 1	PLCは起動信号をリセットします。 状態"中立軸/主軸"、"NC軸"、および"軸入れ替え可能"。

15.6.4.3 ユーザーインタフェース:機能

下の 2 つの表に、使用可能なインタフェース信号の概要を示します。その後、信号の詳しい説明と、どの信号がどの機能に関連しているかを説明します。

PLC 軸への信号

DB380x	PLC 軸への信号(PLC → NC) [r/w]							
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB3000	軸位置決めの起動	主軸位置決めの起動	主軸回転の起動	主軸揺動の起動	-	-	-	-
DBB3001	-	-	主軸回転の停止	主軸揺動の停止	-	-	-	-

DB380x	PLC 軸への信号(PLC → NC) [r/w]							
DBB3002	自動ギヤ選択	周速一定制御	M4 に応じた回転方向	-	手動パルス発生器による移動量重畳	移動寸法、インチ(メトリックではない)	距離条件、最短距離(DC)	距離条件、インクリメンタル(IC)
DBB3003	割り出し位置	-	-	-	-	-	距離条件、絶対値正方向(ACP)	距離条件、絶対値負方向(ACP)
DBD3004	位置決め(REAL、軸割り出しのとき:DINT)							
DBD3008	送り速度(REAL)。< 0 の場合、マシンデータ POS_AX_VELO の値が使用されます。							

距離条件と回転定義の方向のビットは、特定の位置決めまたは移動モードを定義します。いずれかのビットだけを設定してください。

意味	設定する距離条件
絶対位置決め	どのモードビットも設定しない
インクレメンタル位置決め	DBB3002.0 = 1
最短距離位置決め	DBB3002.1 = 1
絶対位置決め、正の移動方向	DBB3003.1 = 1
絶対位置決め、負の移動方向	DBB3003.0 = 1
M4 に応じた回転方向	DBB3002.5 = 1

残りのビットは、特定の機能を指定して開始するのに使用します。これらの機能ビットを、位置および速度と共に機能ごとに詳しく説明します。

PLC 軸からの信号

DB390x	PLC 軸からの信号(NC → PLC) [r]							
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB3000	軸位置決め有効	位置決め完了		-	-	-	移動中の異常	軸の起動不可
DBB3001	-	-	-	-	-	-	-	-
DBB3002	-	-	-	-	-	-		-
DBB3003	異常番号							

下記の機能を使用するには、以下の要件を満たしている必要があります。

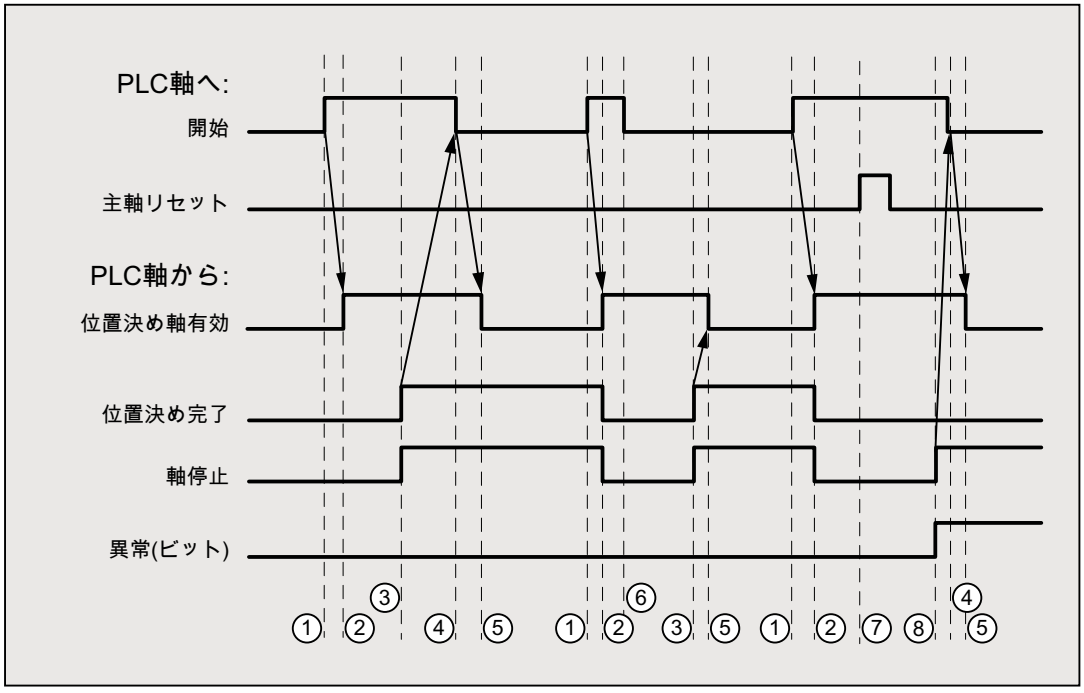
- 軸または主軸が PLC に正しく割り当てられていること。
- コントローライネーブルとパルスイネーブルが設定されていること。
- 制御信号をすべて設定した後で、DB380x.DBB3000 で開始信号が 1 つのみ設定されていること。

15.6.4.4 主軸位置決め

DB380x	PLC → NC 制御信号	備考
DBX3002.0	ステップ移動	4 つの信号のうち、同時に設定できるのは 1 つだけです。 すべての信号 0:絶対位置決め
DBX3002.1	最短軌跡に沿った移動	
DBX3003.0	アブソリュート、負の方向	
DBX3003.1	アブソリュート、正の方向	
DBD3004	位置指令/距離指令	DBX3002.0 == 1 の場合の距離指令
DBD3008	送り速度	0:MD35300 $MA_SPIND_POSCTRL_VELO からの値による移動
DBX3000.6	起動	**注記**信号をリセットしても停止しません。
DBX2.2	残移動距離削除、主軸リセット	中断信号。機能を終了します。

DB390x	NC → PLC 状態信号	備考
DBX3000.7	位置決め軸有効	1: オーバライド = 0 の場合または起動時に指令位置決め完了の場合
DBX3000.6	位置決め完了	1:「精密イグザクトストップ」で指令位置決め完了
DBX3000.0	主軸の起動不可	
DBX3000.1	移動中の異常	1:移動時の異常、DBB3003 で異常番号を確認してください。

DB390x	NC → PLC 状態信号	備考
DBB3003	異常番号	
DBX1.4	軸/主軸停止	1:n < n_{min} の場合



① *起動*の立ち上がりエッジで、ユーザーによって起動された機能

② *軸位置決め有効* = メッセージは、機能が有効で出力信号が有効であり、*位置決め完了*と *軸停止*が取り消し可能であることを示します。パス指定 = 0 の場合、信号はキャンセルされません。

③ 位置に到達すると、信号(*位置決め完了*)で通知され、*主軸停止*が設定されます。

④ ユーザーは*起動*を取り消します。

⑤ それから*位置決め軸有効*信号がリセットされます。

⑥ ユーザーは、*位置決め軸有効*信号を受け取るとすぐに*起動*信号をリセットします。

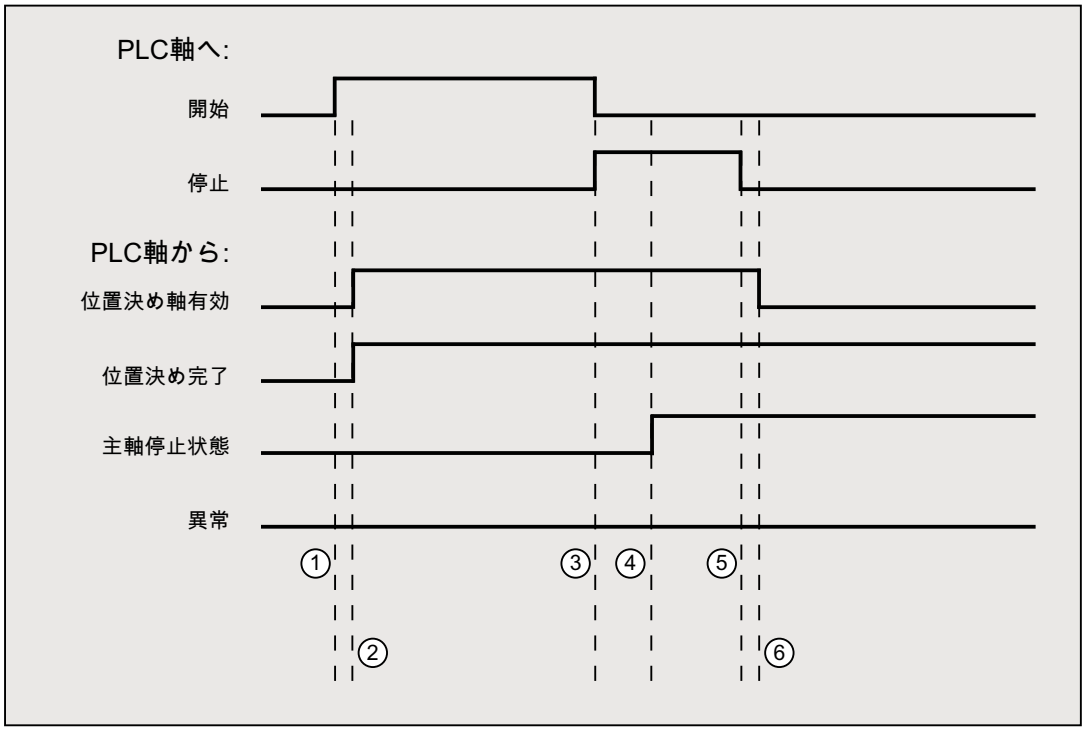
⑦ 位置決めは、*主軸のリセット*を設定すると中止されます。この信号は、最低でも1回の PLC 周期の間、存在している必要があります。

⑧ 主軸が停止状態(*主軸停止*)になり、*異常*信号が設定されます。(この場合、異常番号 115 が出力されます。)

15.6.4.5 主軸回転

DB380x	PLC → NC 制御信号	備考
DBX3002.0	ステップ移動	-
DBX3002.1	最短軌跡に沿った移動	-
DBX3002.5	M4 に応じた回転方向	1:M4 で指定される回転の方向 0:M3 で指定される回転の方向
DBX3003.0	絶対値、負の方向	-
DBX3003.1	絶対値、正の方向	-
DBD3008	送り速度	主軸速度
DBX3000.5	主軸回転の起動	-
DBX3001.5	主軸回転の停止	-

DB390x	NC → PLC 状態信号	備考
DBX3000.7	位置決め軸有効	1:起動または停止== 1 の場合
DBX3000.6	位置決め完了	1:機能が異常なく開始されました。
DBX3000.1	異常	1:移動時の異常。DBB3003 の異常番号を評価してください。
DBB3003	異常番号	-
DBX1.4	軸/主軸停止	-



① *起動*の立ち上がりエッジで、ユーザーによって起動された機能

② メッセージ*位置決め軸有効*および*位置決め完了*が返されます。この場合、*位置決め完了*は無意味です。

③ ユーザーが、*起動*のリセットと*停止*の設定によって主軸の回転を停止します。

④ 主軸が停止し、*主軸停止*信号が設定されます。

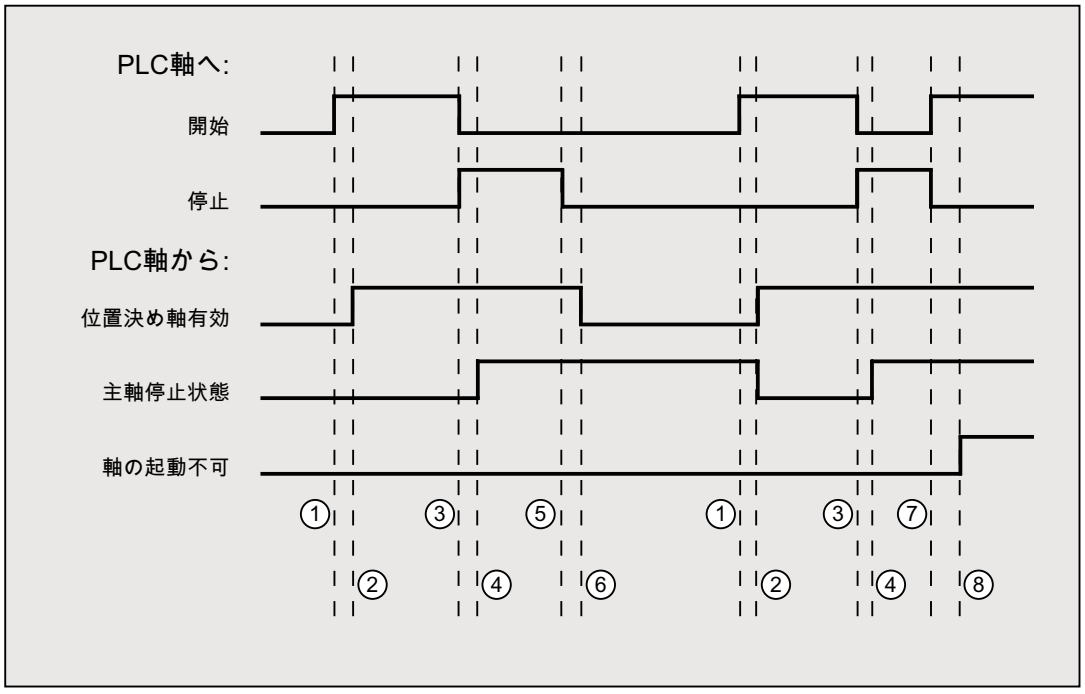
⑤ ユーザーが*停止*をリセットします。

⑥ *停止*をリセットすると、*軸位置決め有効*がリセットされます。

15.6.4.6 主軸揺動

DB380x	PLC → NC 制御信号	備考
DBX3002.0	インクレメンタル	どのビットも設定することはできません。
DBX3002.1	最短距離	
DBX3002.5	M4 に応じた回転方向	
DBX3003.0	絶対値、負の方向	
DBX3003.1	絶対値、正の方向	
DBD3004	ギヤ選択指令	MD35010 $MA_GEAR_STEP_CHANGE_ENABLE = 0 0 - 5:揺動 MD35010 $MA_GEAR_STEP_CHANGE_ENABLE = 1 0:揺動 1:ギヤ選択切替 M41 での揺動 2:ギヤ選択切替 M42 での揺動 3:ギヤ選択切替 M43 での揺動 4:ギヤ選択切替 M44 での揺動 5:ギヤ選択切替 M45 での揺動
DBD3008	送り速度	揺動の場合、無効です。揺動速度は、マシンデータ MD35400 $MA_SPIND_OSCILL_DES_VELO から得られます。
DBX3000.5	主軸揺動の起動	停止の直後に開始することはできません。まず、停止をリセットしてください(両方が 0)。
DBX3001.5	主軸揺動の停止	

DB390x	NC → PLC 状態信号	備考
DBX3000.7	位置決め軸有効	1:起動または停止== 1 の場合
DBX3000.6	位置決め完了	1:開始後
DBX3000.1	異常	1:移動時の異常。DBB3003 の異常番号を評価してください。
DBX3000.0	軸の起動不可	1:開始時の異常。DBB3003 の異常番号を評価してください。
DBB3003	異常番号	
DBX1.4	軸/主軸停止	



① *起動*の立ち上がりエッジで、ユーザーによって起動された機能
注:これは、*位置決め軸有効*信号がリセットされた場合にのみ可能です。

② メッセージ*位置決め軸有効*および*位置決め完了*が返されます。この場合、*位置決め完了*は無意味なので、表示していません。

③ ユーザーが、*起動*のリセットと*停止*の設定によって主軸の揺動を停止します。

④ 主軸が停止し、*主軸停止*信号が設定されます。

⑤ ユーザーが*停止*をリセットします。

⑥ *停止*をリセットすると、*軸位置決め有効*がリセットされます。

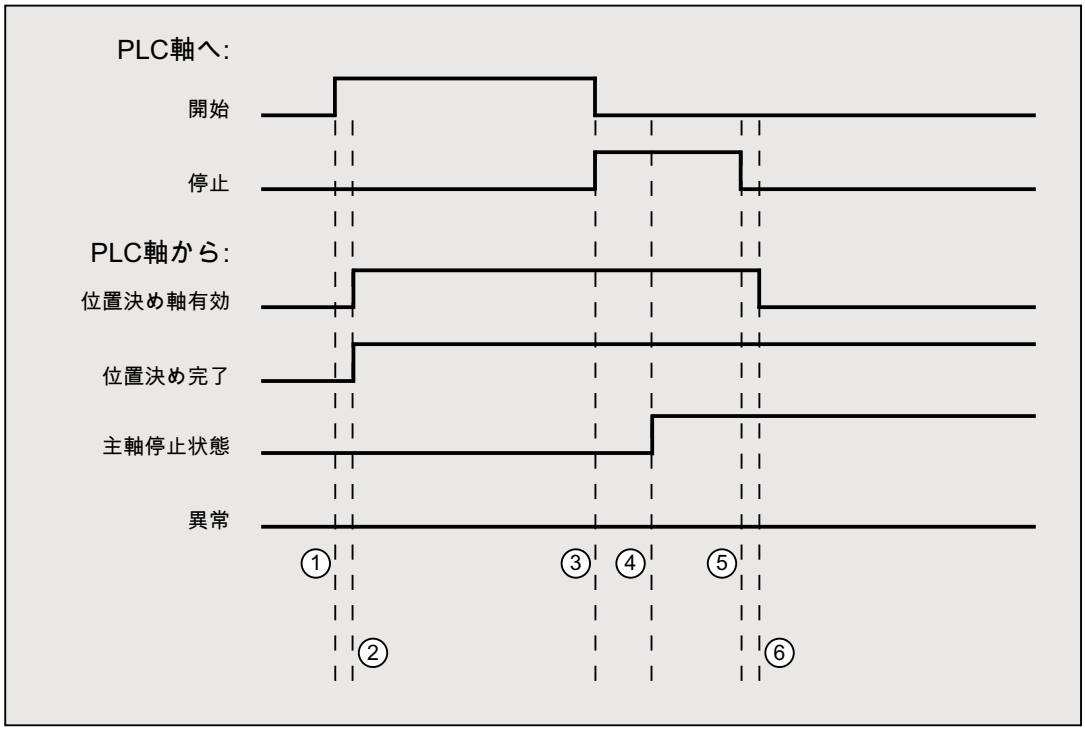
⑦ *停止*がユーザープログラムでリセットされ、*開始*が誤って同じ PLC サイクルで再度、設定されています。つまり、*軸位置決め有効*がリセットされず、

⑧ *軸の起動不可*信号が設定されます(異常番号 106)。

15.6.4.7 軸割り出し

DB380x	PLC → NC 制御信号	備考
DBX3002.0	ステップ移動	信号のうち、同時に設定できるのは 1 つだけです。 すべての信号 0:絶対位置決め
DBX3002.1	最短軌跡に沿った移動	
DBX3002.5	M4 に応じた回転方向	
DBX3003.0	絶対値、負の方向	
DBX3003.1	絶対値、正の方向	
DBX3003.7	割り出し位置	軸割り出し ON
DBD3004	位置指令/距離指令	DBX3002.0 == 1 の場合の距離指令
DBD3008	送り速度	0:MD32060 $MA_POS_AX_VELO からの値による移動
DBX3000.7	軸位置決めの起動	**注記**信号をリセットしても停止しません。
DBX2.2	残移動距離削除、主軸リセット	中断信号。機能を終了します。

DB390x	NC → PLC 状態信号	備考
DBX3000.7	位置決め軸有効	オーバーライド = 0 または位置指令値に到達した場合も 1
DBX3000.6	位置決め完了	1:「精密イグザクトストップ」で位置指令値に到達
DBX3000.1	異常	1:移動時の異常。DBB3003 の異常番号を評価してください。
DBB3003	異常番号	-
DBX1.4	軸/主軸停止	-



① *起動*の立ち上がりエッジで、ユーザーによって起動された機能
注:これは、*位置決め軸有効*信号がリセットされた場合にのみ可能です。

② メッセージ*位置決め軸有効*および*位置決め完了*が返されます。この場合、*位置決め完了*は無意味です。

③ ユーザーが、*起動*のリセットと*停止*の設定によって主軸の揺動を停止します。

④ 主軸が停止し、*主軸停止*信号が設定されます。

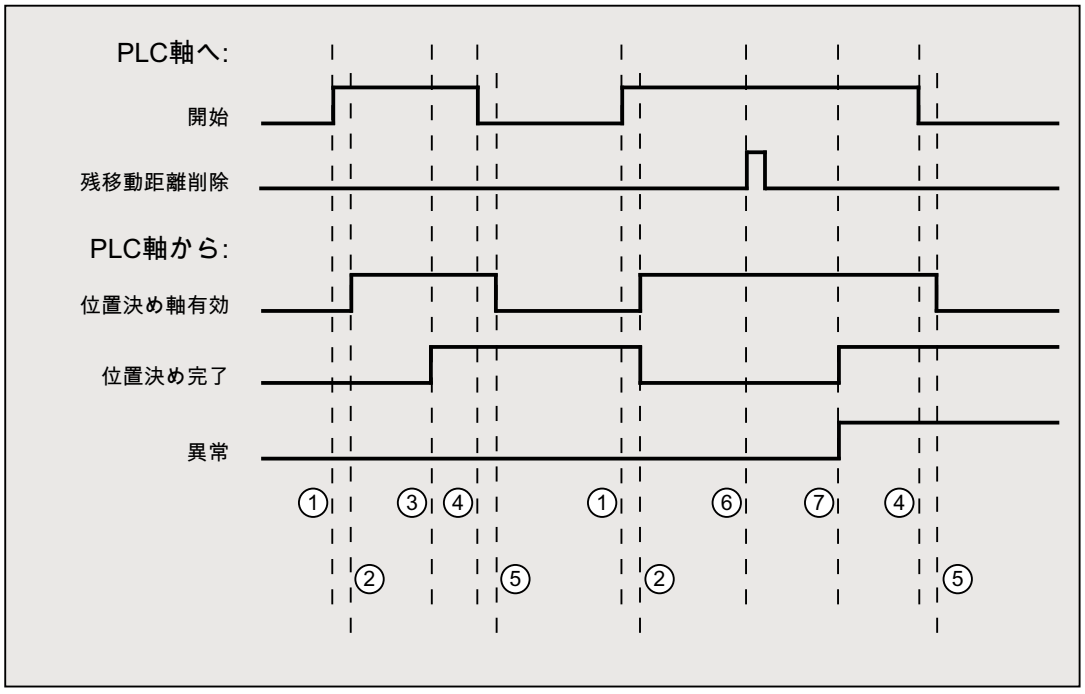
⑤ ユーザーが*停止*をリセットします。

⑥ *停止*をリセットすると、*軸位置決め有効*がリセットされます。

15.6.4.8 軸位置決め、メトリック

DB380x	PLC → NC 制御信号	備考
DBX3002.0	ステップ移動	信号のうち、同時に設定できるのは 1 つだけです。 すべての信号 0:絶対位置決め
DBX3002.1	最短軌跡に沿った移動	
DBX3002.5	M4 に応じた回転方向	
DBX3003.0	絶対値、負の方向	
DBX3003.1	絶対値、正の方向	
DBD3002.2	移動寸法、インチ	0:移動寸法、メトリック
DBX3002.3	手動パルス発生器による移動量重畳	0:オーバライド OFF
DBD3004	位置指令/距離指令	DBX3002.0 == 1 の場合の距離指令
DBD3008	送り速度	0:MD32060 $MA_POS_AX_VELO からの値による移動
DBX3000.7	軸位置決めの起動	**注記**信号をリセットしても停止しません。
DBX2.2	残移動距離削除、主軸リセット	中断信号。機能を終了します。

DB390x	NC → PLC 状態信号	備考
DBX3000.7	位置決め軸有効	オーバーライド = 0 または位置指令値に到達した場合も 1
DBX3000.6	位置決め完了	1:軸が位置指令値に到達しました。
DBX3000.1	異常	1:移動時の異常。DBB3003 の異常番号を評価してください。
DBB3003	異常番号	-



① *起動*の立ち上がりエッジを使用した機能開始の起動

② *軸位置決め有効* = 1 は、機能がアクティブで出力信号が有効なことを示し、*位置決め完了*および *軸停止*は必要に応じてクリアされます。

③ 肯定応答*位置決め完了* = 1 と*位置決め軸有効* = 1

④ 確認応答受信後の機能起動のリセット

⑤ 機能による信号の変更

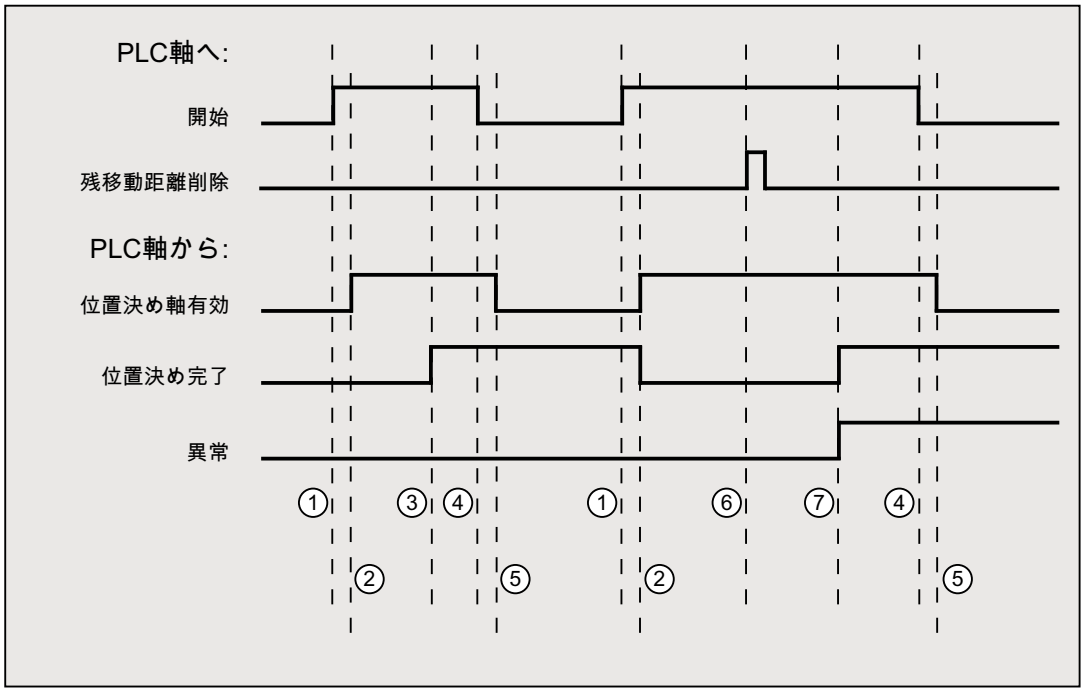
⑥ 位置決めが残移動距離削除によって中止されます。信号持続時間は、最小で 1 PLC 周期です。

⑦ 信号*位置決め完了*と*異常*がセットされ、*異常番号*を読み取ることができます(この場合は 30)。

15.6.4.9 軸位置決め、インチ

DB380x	PLC → NC 制御信号	備考
DBX3002.0	ステップ移動	信号のうち、同時に設定できるのは 1 つだけです。 すべての信号 0:絶対位置決め
DBX3002.1	最短軌跡に沿った移動	
DBX3002.5	M4 に応じた回転方向	
DBX3003.0	絶対値、負の方向	
DBX3003.1	絶対値、正の方向	
DBX3002.2	移動寸法、インチ	0:移動寸法、メトリック
DBX3002.3	手動パルス発生器による移動量重畳	0:オーバライド OFF
DBD3004	位置指令/距離指令	DBX3002.0 == 1 の場合の距離指令
DBD3008	送り速度	0:MD32060 $MA_POS_AX_VELO からの値による移動
DBX3000.7	軸位置決めの起動	**注記**信号をリセットしても停止しません。
DBX2.2	残移動距離削除、主軸リセット	中断信号。機能を終了します。

DB390x	NC → PLC 状態信号	備考
DBX3000.7	位置決め軸有効	オーバーライド = 0 または位置指令値に到達した場合も 1
DBX3000.6	位置決め完了	1:軸が位置指令値に到達しました。
DBX3000.1	異常	1:移動時の異常。DBB3003 の異常番号を評価してください。
DBB3003	異常番号	-



① *起動*の立ち上がりエッジを使用した機能開始の起動

② *軸位置決め有効* = 1 は、機能がアクティブで出力信号が有効なことを示し、*位置決め完了*および *軸停止*は必要に応じてクリアされます。

③ 肯定応答*位置決め完了* = 1 と*位置決め軸有効* = 1

④ 確認応答受信後の機能起動のリセット

⑤ 機能による信号の変更

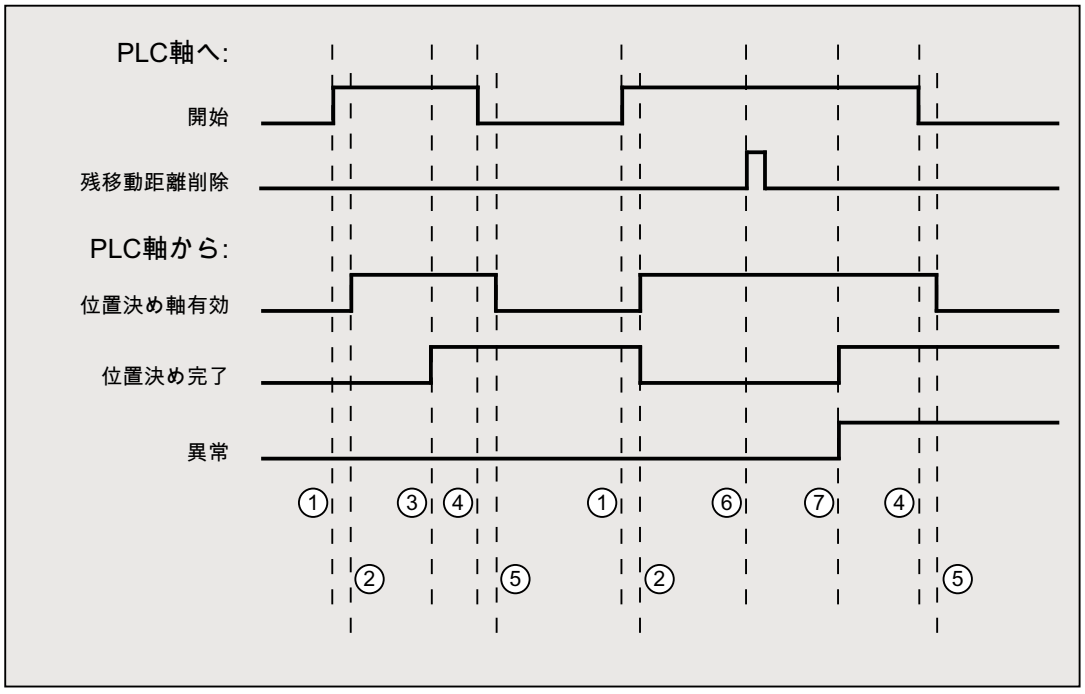
⑥ 位置決めが残移動距離削除によって中止されます。信号持続時間は、最小で 1 PLC 周期です。

⑦ 信号*位置決め完了*と*異常*がセットされ、*異常番号*を読み取ることができます(この場合は 30)。

15.6.4.10 手動パルス発生器の移動量重畳による軸位置決め、メトリック

DB380x	PLC → NC 制御信号	備考
DBX3002.0	ステップ移動	信号のうち、同時に設定できるのは 1 つだけです。 すべての信号 0:絶対位置決め
DBX3002.1	最短軌跡に沿った移動	
DBX3002.5	M4 に応じた回転方向	
DBX3003.0	絶対値、負の方向	
DBX3003.1	絶対値、正の方向	
DBX3002.2	移動寸法、インチ	0:移動寸法、メトリック
DBX3002.3	手動パルス発生器による移動量重畳	0:オーバライド OFF
DBD3004	位置指令/距離指令	DBX3002.0 == 1 の場合の距離指令
DBD3008	送り速度	0:MD32060 $MA_POS_AX_VELO からの値による移動
DBX3000.7	軸位置決めの起動	**注記**信号をリセットしても停止しません。
DBX2.2	残移動距離削除、主軸リセット	中断信号。機能を終了します。

DB390x	NC → PLC 状態信号	備考
DBX3000.7	位置決め軸有効	オーバーライド = 0 または位置指令値に到達した場合も 1
DBX3000.6	位置決め完了	1:軸が位置指令値に到達しました。
DBX3000.1	異常	1:移動時の異常。DBB3003 の異常番号を評価してください。
DBB3003	異常番号	-



① *起動*の立ち上がりエッジを使用した機能開始の起動

② *軸位置決め有効* = 1 は、機能がアクティブで出力信号が有効なことを示し、*位置決め完了*および *軸停止*は必要に応じてクリアされます。

③ 肯定応答*位置決め完了* = 1 と*位置決め軸有効* = 1

④ 確認応答受信後の機能起動のリセット

⑤ 機能による信号の変更

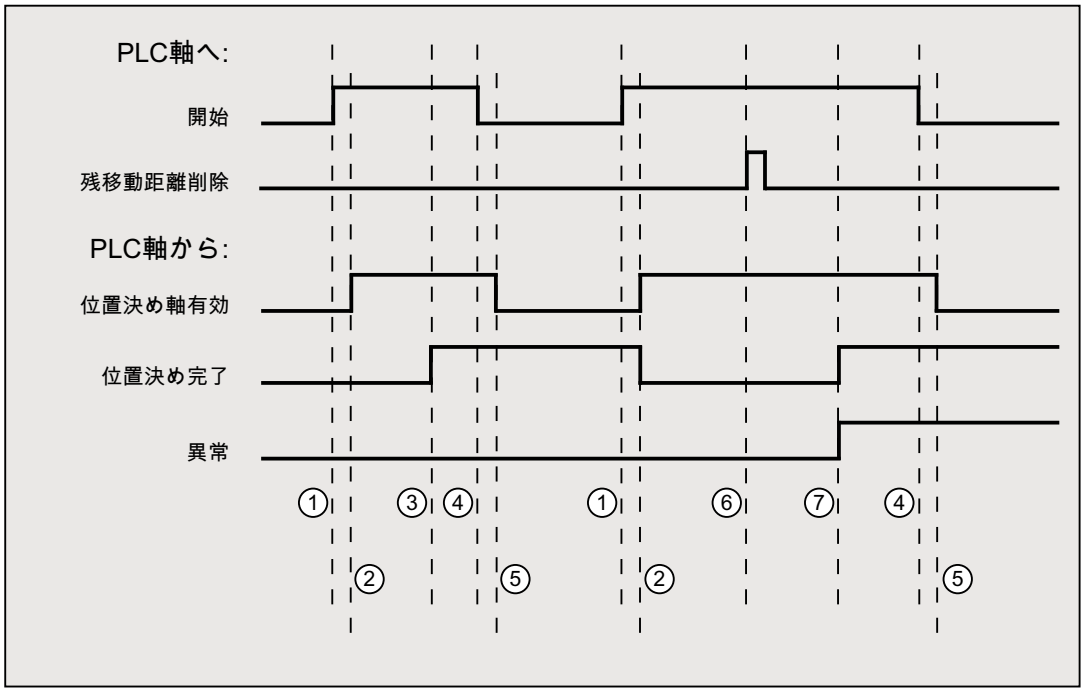
⑥ 位置決めが残移動距離削除によって中止されます。信号持続時間は、最小で 1 PLC 周期です。

⑦ 信号*位置決め完了*と*異常*がセットされ、*異常番号*を読み取ることができます(この場合は 30)。

15.6.4.11 手動パルス発生器の移動量重畳による軸位置決め、インチ

DB380x	PLC → NC 制御信号	備考
DBX3002.0	ステップ移動	信号のうち、同時に設定できるのは 1 つだけです。 すべての信号 0:絶対位置決め
DBX3002.1	最短軌跡に沿った移動	
DBX3002.5	M4 に応じた回転方向	
DBX3003.0	絶対値、負の方向	
DBX3003.1	絶対値、正の方向	
DBX3002.2	移動寸法、インチ	0:移動寸法、メトリック
DBX3002.3	手動パルス発生器による移動量重畳	0:オーバライド OFF
DBD3004	位置指令/距離指令	DBX3002.0 == 1 の場合の距離指令
DBD3008	送り速度	0:MD32060 $MA_POS_AX_VELO からの値による移動
DBX3000.7	軸位置決めの起動	**注記**信号をリセットしても停止しません。
DBX2.2	残移動距離削除、主軸リセット	中断信号。機能を終了します。

DB390x	NC → PLC 状態信号	備考
DBX3000.7	位置決め軸有効	オーバーライド = 0 または位置指令値に到達した場合も 1
DBX3000.6	位置決め完了	1:軸が位置指令値に到達しました。
DBX3000.1	異常	1:移動時の異常。DBB3003 の異常番号を評価してください。
DBB3003	異常番号	-



① *起動*の立ち上がりエッジを使用した機能開始の起動

② *軸位置決め有効* = 1 は、機能がアクティブで出力信号が有効なことを示し、*位置決め完了*および *軸停止*は必要に応じてクリアされます。

③ 肯定応答*位置決め完了* = 1 と*位置決め軸有効* = 1

④ 確認応答受信後の機能起動のリセット

⑤ 機能による信号の変更

⑥ 位置決めが残移動距離削除によって中止されます。信号持続時間は、最小で 1 PLC 周期です。

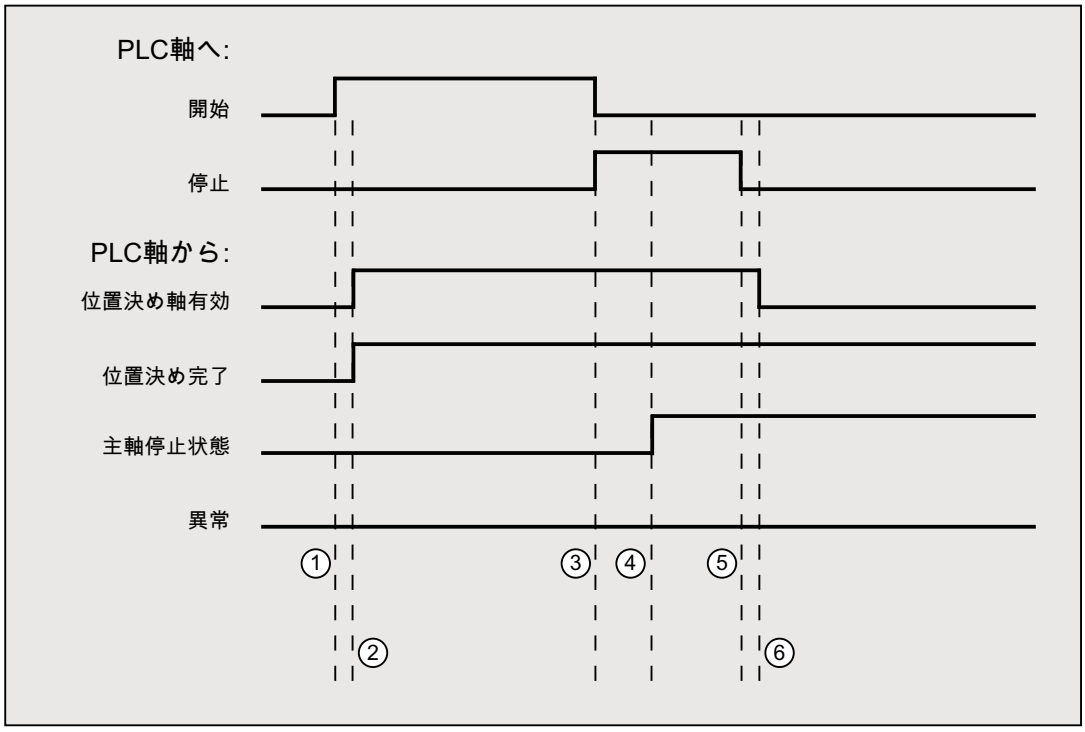
⑦ 信号*位置決め完了*と*異常*がセットされ、*異常番号*を読み取ることができます(この場合は 30)。

15.6.4.12 自動ギヤ選択切替を使用した主軸回転

DB380x	PLC → NC 制御信号	備考
DBX3002.0	ステップ移動	-
DBX3002.1	最短軌跡に沿った移動	-
DBX3002.5	M4 に応じた回転方向	1:M4 で指定される回転の方向 0:M3 で指定される回転の方向

DB380x	PLC → NC 制御信号	備考
DBX3003.0	絶対値、負の方向	-
DBX3003.1	絶対値、正の方向	-
DBX3002.7	自動ギヤ選択切替	1:自動ギヤ選択切替 ON
DBD3008	送り速度	主軸速度
DBX3000.5	主軸回転の起動	-
DBX3001.5	主軸回転の停止	-

DB390x	NC → PLC 状態信号	備考
DBX3000.7	位置決め軸有効	1:起動または停止 == 1 の場合
DBX3000.6	位置決め完了	1:指令速度を出力
DBX3000.1	異常	1:移動時の異常。DBB3003 の異常番号を評価してください。
DBB3003	異常番号	-
DBX1.4	軸/主軸停止	-



① *起動*の立ち上がりエッジで、ユーザーによって起動された機能

② メッセージ*位置決め軸有効*および*位置決め完了*が返されます。この場合、*位置決め完了*は無意味です。

③ ユーザーが、*起動*のリセットと*停止*の設定によって主軸の回転を停止します。

④ 主軸が停止し、*主軸停止*信号が設定されます。

⑤ ユーザーが*停止*をリセットします。

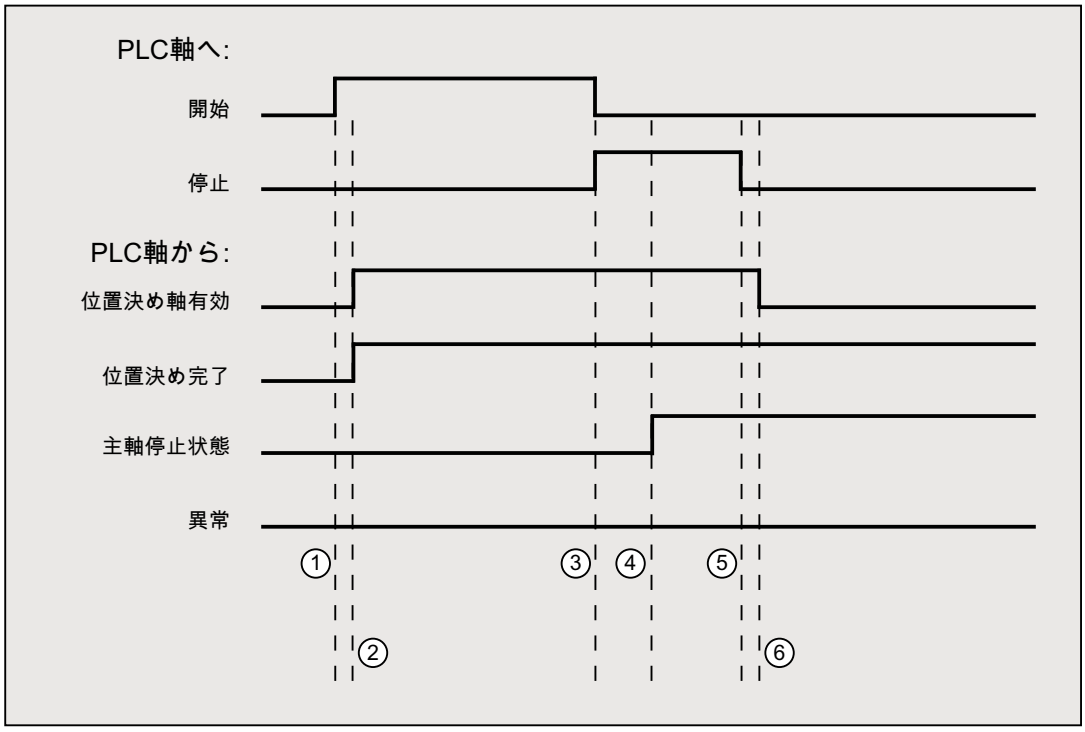
⑥ *停止*をリセットすると、*軸位置決め有効*がリセットされます。

15.6.4.13 周速一定制御[m/min]を使用した主軸回転

DB380x	PLC → NC 制御信号	備考
DBX3002.0	ステップ移動	-
DBX3002.1	最短軌跡に沿った移動	-
DBX3002.5	M4 に応じた回転方向	1:M4 で指定される回転の方向 0:M3 で指定される回転の方向
DBX3003.0	絶対値、負の方向	-

DB380x	PLC → NC 制御信号	備考
DBX3003.1	絶対値、正の方向	-
DBX3002.2	移動寸法、インチ	0:移動寸法、メトリック
DBX3002.6	周速切削速度	1:一定の切削速度がオン
DBD3008	送り速度	主軸速度
DBX3000.5	主軸回転の起動	-
DBX3001.5	主軸回転の停止	-

DB390x	NC → PLC 状態信号	備考
DBX3000.7	位置決め軸有効	1:起動または停止== 1 の場合
DBX3000.6	位置決め完了	1:指令速度を出力
DBX3000.1	異常	1:移動時の異常。DBB3003 の異常番号を評価してください。
DBB3003	異常番号	-
DBX1.4	軸/主軸停止	-



① *起動*の立ち上がりエッジで、ユーザーによって起動された機能

② メッセージ*位置決め軸有効*および*位置決め完了*が返されます。この場合、*位置決め完了*は無意味です。

③ ユーザーが、*起動*のリセットと*停止*の設定によって主軸の回転を停止します。

④ 主軸が停止し、*主軸停止*信号が設定されます。

⑤ ユーザーが*停止*をリセットします。

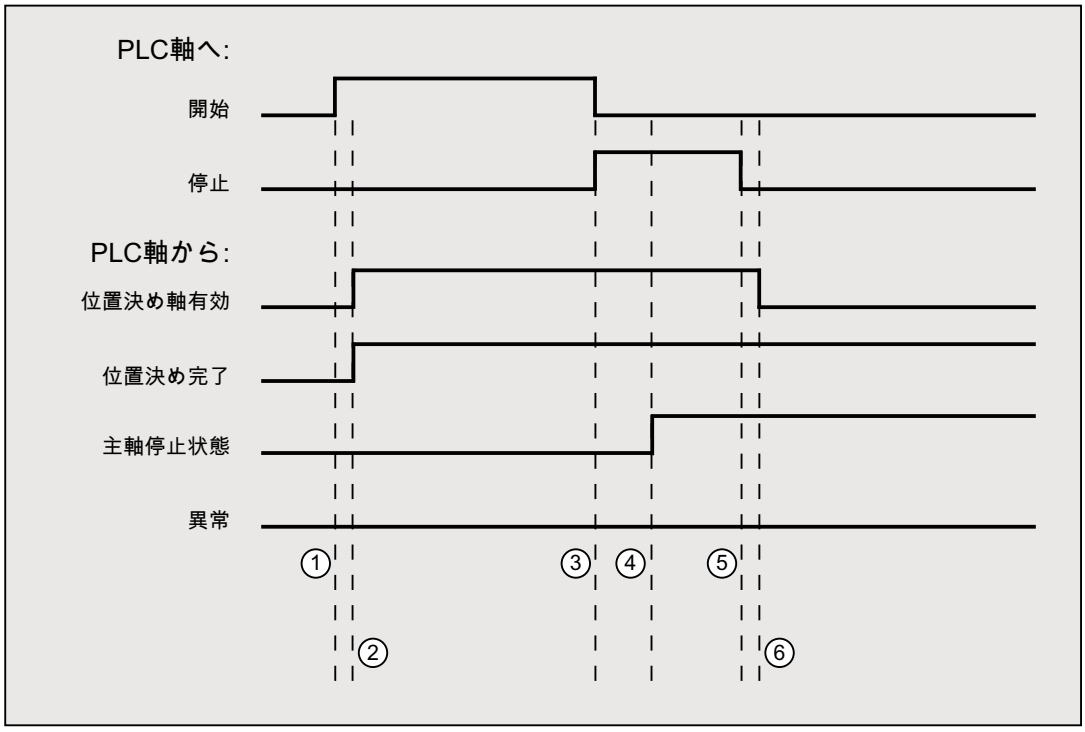
⑥ *停止*をリセットすると、*軸位置決め有効*がリセットされます。

15.6.4.14 周速一定制御[ft/min]を使用した主軸回転

DB380x	PLC → NC 制御信号	備考
DBX3002.0	ステップ移動	-
DBX3002.1	最短軌跡に沿った移動	-
DBX3002.5	M4 に応じた回転方向	1:M4 で指定される回転の方向 0:M3 で指定される回転の方向
DBX3003.0	絶対値、負の方向	-

DB380x	PLC → NC 制御信号	備考
DBX3003.1	絶対値、正の方向	-
DBX3002.2	移動寸法、インチ	1:移動寸法、インチ
DBX3002.6	周速切削速度	1:一定の切削速度がオン
DBD3008	送り速度	主軸速度
DBX3000.5	主軸回転の起動	-
DBX3001.5	主軸回転の停止	-

DB390x	NC → PLC 状態信号	備考
DBX3000.7	位置決め軸有効	1:起動または停止== 1 の場合
DBX3000.6	位置決め完了	1:指令速度を出力
DBX3000.1	異常	1:移動時の異常。DBB3003 の異常番号を評価してください。
DBB3003	異常番号	-
DBX1.4	軸/主軸停止	-



① *起動*の立ち上がりエッジで、ユーザーによって起動された機能

② メッセージ*位置決め軸有効*および*位置決め完了*が返されます。この場合、*位置決め完了*は無意味です。

③ ユーザーが、*起動*のリセットと*停止*の設定によって主軸の回転を停止します。

④ 主軸が停止し、*主軸停止*信号が設定されます。

⑤ ユーザーが*停止*をリセットします。

⑥ *停止*をリセットすると、*軸位置決め有効*がリセットされます。

15.6.4.15 エラーメッセージ

機能を実行できなかった場合、異常に応じて次の信号が設定されます。

- DB390x .DBX3000.0 == 1 (軸の起動不可)
- DB390x.DBX3000.1 == 1 (移動中の異常)

正確な異常の原因は、次のように示されます。

- DB390x.DBB3003 (異常番号)

異常番号		意味
10 進数	16 進数	
1	01	軸/主軸の複数の機能が同時に起動された。
20	14	機能が位置決め完了せずに起動された。
30	1E	軸/主軸が移動中に NC に転送された。
40	28	軸が NC プログラムによってプログラム指令された、NC 内部異常
50	32	固定された PLC 軸:移動(ジョグ)または原点確立
60	3C	固定された PLC 軸:チャネル状態により起動できない
100	64	軸/主軸について誤った位置がプログラムされている(アラーム番号 1) 16830)
101	65	プログラム指令速度が速すぎる
102	66	周速一定制御の値の範囲が正しくない(アラーム番号 1) 14840)
104	68	スレーブ主軸:不適切なプログラミング(アラーム番号 1) 22030)
105	69	使用可能な検出器がない(アラーム番号 1) 16770)
106	6A	軸の位置決め処理がまだ有効(アラーム番号 1) 22052)
107	6B	原点マークが見つからない(アラーム番号 1) 22051)
108	6C	速度制御から位置制御への遷移がない(アラーム番号 1) 22050)
109	6D	原点マークが見つからない(アラーム番号 1) 22051)
110	6E	負の速度
111	6F	速度指令値がゼロ
112	70	無効なギヤ選択
115	73	プログラム指令位置に到達していない
117	75	G96/G961 が NC 内で無効
118	76	G96/G961 が NC 内でまだ有効
120	78	軸が割り出し軸ではない(アラーム番号 1) 20072)
121	79	割り出し位置の異常(アラーム番号 1) 17510)
125	7D	DC (最短距離)は不可(アラーム番号 1) 16800)
126	7E	絶対値のマイナスは不可(アラーム番号 1) 16820)
127	7F	絶対値のプラスは不可(アラーム番号 1) 16810)
128	80	直径指定に使用可能な径方向軸なし(アラーム番号 1) 16510)
130	82	ソフトウェアリミットスイッチ、プラス(アラーム番号 1) 20070)
131	83	ソフトウェアリミットスイッチ、マイナス(アラーム番号 1) 20070)

異常番号		意味
10 進数	16 進数	
132	84	作業領域リミット、プラス(アラーム番号 1) 20071)
133	85	作業領域リミット、マイナス(アラーム番号 1) 20071)
134	85	フレームは割り出し軸に使えない
135	87	「カップリングジョイント」による軸割り出しが有効(アラーム番号 1) 17501)
136	88	「カップリングジョイント」による軸割り出しが有効で、軸の原点確立が未完(アラーム番号 1) 17503)
137	89	座標変換された主軸/軸に対する主軸運転は不可(アラーム番号 1) 22290)
138	8A	対応する有効な座標系別作業領域リミット、プラスが軸に対して違反(アラーム番号 1)20082)
139	8B	対応する有効な座標系別作業領域リミット、マイナスが軸に対して違反(アラーム番号 1)20082)
200	C8	システムアラーム番号 1) 450007
1) 詳細なアラームの説明については、以下を参照してください。『アラーム診断マニュアル; SINUMERIK 828D、SINAMICS S120』		

15.6.5 ASUB の起動

15.6.5.1 ジョブの開始

DB340x	ASUB:ジョブ[r/w] (PLC → NC)							
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB0	-	-	-	-	-	-	-	INT1 の開始
DBB1	-	-	-	-	-	-	-	INT2 の開始
DBB2	-	-	-	-	-	-	-	INT3 の開始
DBB3	-	-	-	-	-	-	-	INT4 の開始

割り込み

「INT1 の開始」:

- DBX0.0 = 1:INT1 に割り当てられた割り込みプログラム(ASUB)の起動を要求します。
- DBX0.0 = 0:結果インターフェース DB340x.DBB1000.0 - 3 での確認後に ASUB 要求をリセットします。

「INT2 の開始」:

- DBX1.0 = 1:INT2 に割り当てられた割り込みプログラム(ASUB)の起動を要求します。
- DBX1.0 = 0:結果インターフェース DB340x.DBB1001.0 - 3 での確認後に ASUB 要求をリセットします。

「INT3 の開始」:

- DBX2.0 = 1:INT3 に割り当てられた割り込みプログラム(ASUB)の起動を要求します。
- DBX2.0 = 0:結果インターフェース DB340x.DBB1002.0 - 3 での確認後に ASUB 要求をリセットします。

「INT4 の開始」:

- DBX3.0 = 1:INT4 に割り当てられた割り込みプログラム(ASUB)の起動を要求します。
- DBX3.0 = 0:結果インターフェース DB340x.DBB1003.0 - 3 での確認後に ASUB 要求をリセットします。

15.6.5.2 ジョブ結果

DB340x			**ASUB:結果 [r] (NC → PLC)**					
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB1000	INT1 を開始する信号(DB340x.DBX0.0)							
					ASUB 実行不可 [1]	割り込み番号の割り当てなし [2]	ASUB 実行中	ASUB 完了 [3]
DBB1001	INT2 を開始する信号(DB340x.DBX1.0)							
					ASUB 実行不可 [1]	割り込み番号の割り当てなし [2]	ASUB 実行中	ASUB 完了 [3]

DB340x					**ASUB:結果 [r] (NC → PLC)**			
DBB1002	INT3 を開始する信号(DB340x.DBX2.0)							
					ASUB 実行不可 1)	割り込み番号の割り当てなし 2)	ASUB 実行中	ASUB 完了 3)
DBB1003	INT4 を開始する信号(DB340x.DBX3.0)							
					ASUB 実行不可 1)	割り込み番号の割り当てなし 2)	ASUB 実行中	ASUB 完了 3)

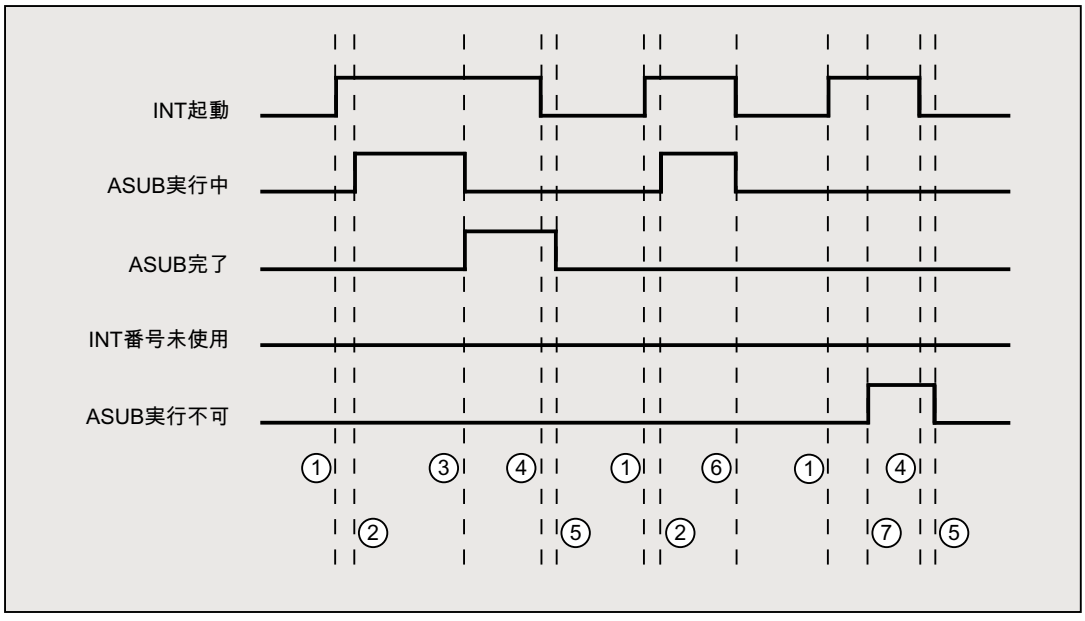
1)否定応答:例: 非常停止またはチャネルリセット要求の場合など。

2)否定応答:番号がまだ割り当てられていません。対策:PI サービス「ASUB」を実行してください。

3)肯定応答:ASUB が正常に終了しました。開始信号をリセット:DB340x.:DBXn.0 = 0。ここで、n = 0、1、2、3

15.6.5.3 信号フロー

信号フロー



① *起動*の立ち上がりエッジで、ユーザーによって起動された機能

② *ASUB 実行中*が返されます。

③ 応答 *ASUB 完了*によって正常に実行されたことが示され、*ASUB 実行中*が取り消されます。

④ 応答を受信すると、ユーザーにより機能を開始するための信号がリセットされます。

⑤ ファームウェアによる信号の変更

⑥ 不可。許容されません。確認応答を受信する前に機能の起動がリセットされている場合は、出力信号が更新されると必ず、起動された機能の処理手順が影響を受けます。

⑦ *ASUB 実行不可*:否定応答、異常が発生しました。

図 15-2 例:信号フロー

15.6.6 HMI でのチャネル選択

機能

機械の開始画面などの HMI に表示されるチャネルは、HMI/PLC インターフェースを介して PLC ユーザープログラムから選択できます。

必要条件

複数のチャネルが NC でパラメータ設定されます。

ジョブおよび確認のインターフェース

DB1900			チャネル選択[r/w] (PLC↔HMI)					
バイト	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0
DBB5011	異常 ID (以下のサブセクションを参照) (HMI → PLC)							
...	...							
DBB5021	ステータス (HMI → PLC)	必要条件 (PLC↔HMI)	ファンクション番号 (PLC→HMI)					
DBB5022	チャネル番号(以下のサブセクションを参照) (PLC→HMI)							

チャネル番号

- チャネル番号:1、2、... 最大チャネル数
- 次のチャネル:FF_H

異常 ID

- 0:エラーなし
- 1:無効なファンクション番号(DBX32.0 - .5)
- 2:無効なパラメータ(DBB33 - DBB35)
- 3:HMI 内部変数への書き込み時の異常
- 10:チャネルが存在しない(DBB33)

機能の処理

PLC → HMI

PLC ユーザープログラムは、以下の実行シーケンスを維持しなければなりません。

1. インターフェースが新しいジョブのために空いているかどうかをチェックします。
   - DB1900.DBX**5021**.6 == 0 (ファンクション要求)
   - DB1900.DBX**5021**.7 == 0 (状態)
2. インターフェースが空いている場合は、ジョブデータを入力します。これで、ファンクション要求が設定されます。
   - DB1900.DBB**5022** = <チャネル番号>
   - DB1900.DBX**5021**.0 - .5 = 1 (ファンクション番号:チャネル選択)
   - DB1900.DBX**5021**.6 = 1 (ファンクション要求)

HMI → PLC

パラメータ設定に**エラーがない**場合は、HMI が以下のように動作します。

1. チャネル選択のファンクション要求が HMI で認識されると、状態が「ファンクション実行中」に設定され、ファンクション要求がリセットされます。
   - DB1900.DBX**5021**.7 = 1 (状態)
   - DB1900.DBX**5021**.6 = 0 (ファンクション要求)
2. チャネル選択が実行されると、状態が再度リセットされ、エラー ID として値 0 が設定されます。
   - DB1900.DBX**5021**.7 = 0 (状態)
   - DB1900.DBX**5011** = 0 (異常 ID)

パラメータ設定に**エラーがある**場合は、HMI が以下のように動作します。

- ファンクション要求がリセットされ、適切なエラー ID が設定されます。
  - DB1900.DBX**5021**.6 = 0 (ファンクション要求)
  - DB1900.DBX**5021**.7 = 0 (状態)
  - DB1900.DBX**5011** = <異常 ID>

15.7 CNC ロック機能(オプション)

15.7.1 機能

注記

「CNC ロック機能」は、ライセンスオプションです(記事番号:6FC5800-0AP76-0YB0)。
CNC ロック機能を使用するには、該当するライセンスを当社から購入してください。試用ライセンスでロック機能を使用することは**できません**。

通知
ライセンス証明書
CNC ロック機能を作成した会社(機械メーカまたは代理店)は、このオプションのライセンス証明書(CoL)を保管しておいてください。
PIN を忘れた場合は、この証明書を使用して当社に問い合わせることができます。証明書(CoL)の所有者は、機械のロックを解除してもらうことができます。

機械メーカは、SINUMERIK Access MyMachine (AMM) Integrate アプリケーションで作成した「CNC ロック機能」と暗号化ファイルを使用して、コントロールシステムでロック期日を有効にすることができます。これにより、機械の使用をロック期日までに制限することができます。コントロールシステムの NC スタート機能は、ロック期日を過ぎるとロックされます。

CNC ロック機能は、追加の暗号化ファイルにより延長または無効化することができます。同意した義務をエンドユーザーが遂行している場合は、機械メーカが、このファイルをエンドユーザーに送信します。

15.7.2 必要条件

「CNC ロック機能」を使用するには、以下の必要条件を満たしてください。

- 「CNC ロック機能」オプションを設定する。
- CPU タイプ「828D Step 2 x.yy」の PLC プロジェクトを使用している。
  互換性モードが無効になっている。
- SINUMERIK Integrate Access MyMachine/ P2P (PC)アプリケーションをインストールしている。

15.7.3 制限事項

CNC ロック機能は、期限付き使用でのビジネスモデルをサポートしています。これにより、設定した期間を超える不正な使用から機能が保護されます。ただし、CNC に直接アクセスすると、この機能の使用が可能になります。CNC ロック機能には、不正な使用を防ぐ絶対的な保護は用意されていません。機械の不正な使用は、CNC の自動モードをロックすると防止されます。実行中の自動プログラムは中断できないため、これによってランタイムが長くなり、ロック期日を過ぎる可能性があります。SINUMERIK コントロールシステムの他のすべての機能は使用可能なままです。

CNC ロック機能を有効にするには、機械メーカのサポートが必要です。したがって、CNC ロック機能の使用時には、以下の注記と補足条件を遵守してください。

注記

CNC ロック機能は、関連する SINUMERIK コントロールシステムへの、PLC プロジェクトの接続に基づいています。SINUMERIK コントロールシステムは、Panel Processing Unit (PPU)、CF カード、およびシステムソフトウェアの組合わせで構成されています。

必要条件

- 不正に使用しようとしたり、矛盾が生じたりすると、CNC ロック機能により機械が停止状態になる場合があります。
- CNC ロック機能を使用するには、機械メーカまたは代理店による、お客様のサイトでの追加のサービス作業が必要な場合があります。
- PLC プロジェクトは、お客様に、保存済み OEM PIN なしで提供しないでください。「無料」の PLC プロジェクトを使用すると、CNC ロック機能の不正使用が可能になります。
- 関連する工作機械の PLC プロジェクトを再実装すると、CNC ロック機能の不正使用が可能になります。
- SINUMERIK の PLC のプログラム組織単位保護を使用してください。PLC プログラミングツールでの起動が可能です。PLC プロジェクトのコードは、秘密にしておいてください。
- 保護を強化するには、各機械に対して固有の OEM PIN を付与してください。
- ロック機能を始めてセットアップ(「最初の CNC ロック機能の作成 (ページ 1478)」の章を参照)する前に、セットアップエンジニアは、日付と時間を SINUMERIK 制御装置に正しく設定しなければなりません。過去の日付を設定すると、機械の運転時間が、実際の日付との差に対応して延長されます。

- CNC ロック機能は、SINUMERIK 828D のリアルタイムクロックに基づいて構築されています。SINUMERIK 828D は、保守が不要な設計であるため、時刻が不明になるおそれがあります。CNC ロック機能は、時刻の電子化数値整合性チェックをおこないます。このチェックは、リアルタイムクロックの電力損失により、正しく動作しない場合があります。この場合は、電源が失われていた時間が無視されます。
- CNC ロック機能を作成した会社は、関連するライセンス証明書(CoL)を保持しています。この証明書は、SIEMENS に対する、許可を受けた機能作成者であることの証拠となるため、自動的に、許可を受けた機械ユーザーである証拠にもなります。
- ソフトウェアの誤動作により、コントロールシステムの想定外のロックが発生する場合があります。

15.7.4 不正使用からの保護

現在想定されている CNC の使用と誤用の一部として、CNC ロック機能は、設定された期間内のみの使用を許可する役割を果たします。使用可能な保護メカニズムで、CNC ロック機能の許可されていない不正使用は防止できますが、保護メカニズムを回避できるリスクは、わずかに残っています。CNC ロック機能は、インストールされている PPU、CF カード、および PLC プロジェクトの組合わせを定期的にチェックします。ロック機能は、3 つのコンポーネントすべての交換時には適用されません。不正使用から保護するために、必ず、「制限事項 (ページ 1477)」の章に記載された二次的な条件を慎重に遵守してください。

注記

CNC ロック機能は、暗号化技術を使用します。最初に市場に導入されたとき、使用されていた技術は最先端のテクノロジでした。時間がたつにつれ、この技術をバイパス(不正使用)できる可能性は増加しています。

15.7.5 最初の CNC ロック機能の作成

最初に CNC ロック機能を作成する場合は、コントロールシステムハードウェア、Panel Processing Unit (PPU)を、CF カード上のシステムソフトウェアと共に、機械に属する PLC プロジェクトに組み合わせます。最初のロック期日も同時に設定します。

ロック機能を作成するには、ハードウェアに適合する暗号化 lockset ファイル(lockset.clc)を生成する必要があります。このファイルは、SINUMERIK Integrate Access MyMachine (AMM)アプリケーションで作成します。

lockset ファイルの生成

lockset ファイルを生成するには、以下のデータが必要です。

- CF カードのシリアル番号
- コントロールシステム(Panel Processing Unit、PPU)のシリアル番号
- OEM PIN
- ロック期日

注記

CF カードとコントロールシステム(PPU)の**シリアル番号**は、SINUMERIK Operate ユーザーインターフェースの[診断|バージョン|ハードウェア| NCU/PLC | ncu1]操作エリアで確認できます。

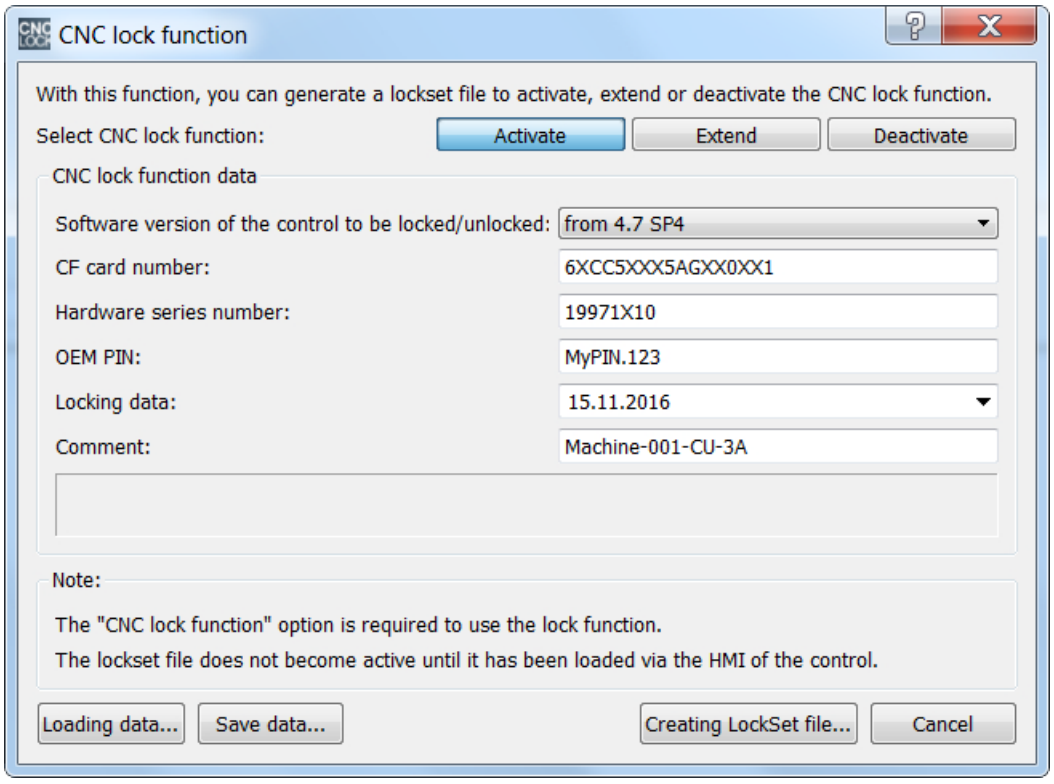
- **CF カード:**[CF カード| SerialNo.]領域
- **PPU**[SINUMERIK 828D PPU | SerialNo.]領域

注記

OEM PIN によって、不正使用からの CNC ロック機能の保護が強化されます。

OEM PIN は、PLC ユーザープログラムでの CNC ロック機能の起動時に、システムによって保存されます。OEM PIN は、ユーザーが PLC ユーザープログラムで表示、変更、削除することはできません。

このデータは、Access MyMachine で[CNC ロック機能]ダイアログウィンドウに入力します(メインメニューから[工具| CNC ロック機能]を呼び出します)。**[起動]**ボタンを選択する必要があります:



必要なすべてのデータを入力した後、[lockset ファイルの作成]を使用して起動する lockset ファイルを生成する必要があります。

lockset ファイルの伝送

lockset ファイルは、Ethernet 接続で直接、または USB フラッシュメモリなどの記憶媒体でコントロールシステムに伝送してください。

このファイルは、/System-CF-Card/User/sinumerik/data/license にあります。

lockset ファイルの読み取り

lockset ファイルをコントロールシステムにインポートする前に、機械メーカは、コントロールシステムに正しい時刻を設定してください。これは、CNC ロック機能を起動する時点の時刻が、監視の開始値として保存されるためです。これで、インポートをユーザーインターフェース、[コミッショニング|ライセンス|ライセンスキーのインポート]操作エリアから開始できます。

インポートには、アクセスレベルは必要ありません。

lockset ファイルのインポート時に異常が発生しない場合は、CNC ロック機能がコントロールシステムで有効になっています。

注記

lockset ファイルのインポート時に異常が発生する場合は、異常別アラームが発行されます。CNC ロック機能の状態は未変更のままです。

注記

機械メーカが、機械のコミッショニングと CNC ロック機能の起動の後に、コントロールシステムのすべてのコンポーネントに関する**コミッショニングアーカイブ**一式を作成することを推奨します。これにより、CNC ロック機能のデータの完全性が確保されます。必要に応じて、このコミッショニングアーカイブを使用して、コントロールシステムを再コミッショニングできます。この場合、CNC ロック機能を再起動するためにサービス作業を依頼する必要はありません。

ロック期日のチェック

CNC ロック機能の起動は、以下のユーザーインタフェースに表示される、lockset ファイルに入力されたロック期日で確認できます。

SINUMERIK Operate ユーザーインターフェース:[コミッショニング|ライセンス]操作エリア

警告時間の設定

警告時間は、ロック期日になる前の時間範囲です。この時間範囲では、アラーム 8063 が毎日 1 回、表示されます。アラームは、ロック期日が迫っており、NC スタートがコントロールシステム用にロックされることを示します。警告時間は、次のマシンデータを使用して設定します。

MD17300 $MN_CNC_LOCK_WARNING_TIME

重要な注記

CNC ロック機能が正確かつ確実に機能するように、以下の情報に留意してください。

注記

CNC ロック機能の起動前に、コントロールシステムで**時刻**を正しく設定してください。

注記

PLC ユーザープログラムには、常に **POU パスワード保護**を使用してください。これにより、ユーザーが機械固有のノウハウをコピーし、自分の PLC ユーザープログラムで使用して、この PLC ユーザープログラムを、CNC ロック機能の PLC キーを含む機械メーカ製 PLC ユーザープログラムの代わりに使用することを防止できます。

注記

エンドユーザーの**コミッショニングアーカイブ**は、CNC ロック機能の**起動後**にのみエクスポートできます。

注記

CNC ロック機能の**解除後**(「CNC ロック機能の解除 (ページ 1484)」を参照)には、**新しいコミッショニングアーカイブ**をエクスポートして、元のアーカイブに置き換えてください。これにより、不注意による CNC ロック機能の再起動を確実に防止できます。ただし、アラームが発行され、NC スタートが無効になります。これを取り消すには、CNC ロック機能を解除するための lockset ファイルを再インポートしてください。

注記

CNC ロック機能の正確かつ確実な動作に対して責任を負うのは、機械メーカのみです。

15.7.6 CNC ロック機能の拡張

CNC ロック機能を拡張するために、機械メーカは、Access MyMachine (AMM)を使用して、CNC ロック機能の**新しいロック期日**を設定した新しい lockset ファイル(lockset.clc)を作成してください。

lockset ファイルの生成

lockset ファイルを生成するには、以下のデータが必要です。

- CF カードのシリアル番号
- コントロールシステム(Panel Processing Unit、PPU)のシリアル番号
- OEM PIN
- **新しい ロック期日**

注記

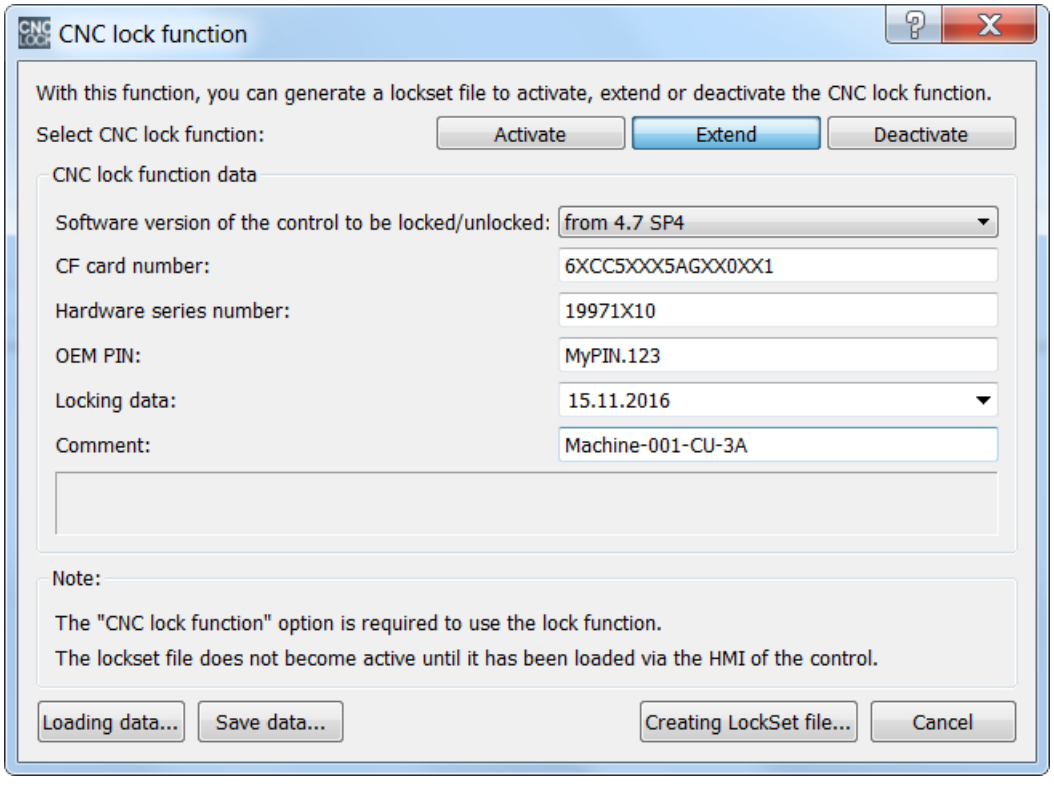
CF カードとコントロールシステム(Panel Processing unit、PPU)のシリアル番号、および OEM PIN は、CNC ロック機能の初回起動時に使用した値と一致するようにしてください。

注記

CF カードとコントロールシステム(PPU)の**シリアル番号**は、SINUMERIK Operate ユーザーインターフェースの[診断|バージョン|ハードウェア| NCU/PLC | ncu1]操作エリアで確認できます。

- **CF カード:**[CF カード| SerialNo.]領域
- **PPU**[SINUMERIK 828D PPU | SerialNo.]領域

このデータは、Access MyMachine で[CNC ロック機能]ダイアログウィンドウに入力します(メインメニューから[工具| CNC ロック機能]を呼び出します)。**[拡張]**ボタンを選択する必要があります:



必要なすべてのデータを入力した後、[lockset ファイルの作成]を使用して拡張する lockset ファイルを生成する必要があります。

lockset ファイルの伝送

新しい lockset ファイルをコントロールシステムに伝送してください。機械メーカは、Ethernet 接続で直接、コントロールシステムに伝送できます。

このファイルは、/System-CF-Card/User/sinumerik/data/license にあります。

または、機械メーカが、新しい lockset ファイルをエンドユーザーに送信し、エンドユーザーがコントロールシステムに伝送します。

lockset ファイルの読み取り

lockset ファイルのインポートは、ユーザーインタフェース、[コミッショニング|ライセンス|ライセンスキーのインポート]操作エリアから開始できます。

インポートには、アクセスレベルは必要ありません。

lockset ファイルのインポート時に異常が発生しない場合は、新しいロック期日を設定した CNC ロック機能がコントロールシステムで有効になっています。

注記

lockset ファイルのインポート時に異常が発生する場合は、異常別アラームが発行されます。CNC ロック機能の状態は未変更のままです。

変更したロック期日のチェック

ロック期日がコントロールシステムで正しく変更されたかどうかは、以下のユーザーインターフェースでチェックできます。

SINUMERIK Operate ユーザーインターフェース:[コミッショニング|ライセンス]操作エリア

15.7.7 CNC ロック機能の解除

CNC ロック機能を解除するために、機械メーカは、Access MyMachine (AMM)を使用して、**ロック期日を設定しない** lockset ファイル(lockset.clc)を作成してください。

lockset ファイルの生成

lockset ファイルを生成するには、以下のデータが必要です。

- CF カードのシリアル番号
- コントロールシステム(Panel Processing Unit、PPU)のシリアル番号
- OEM PIN

注記

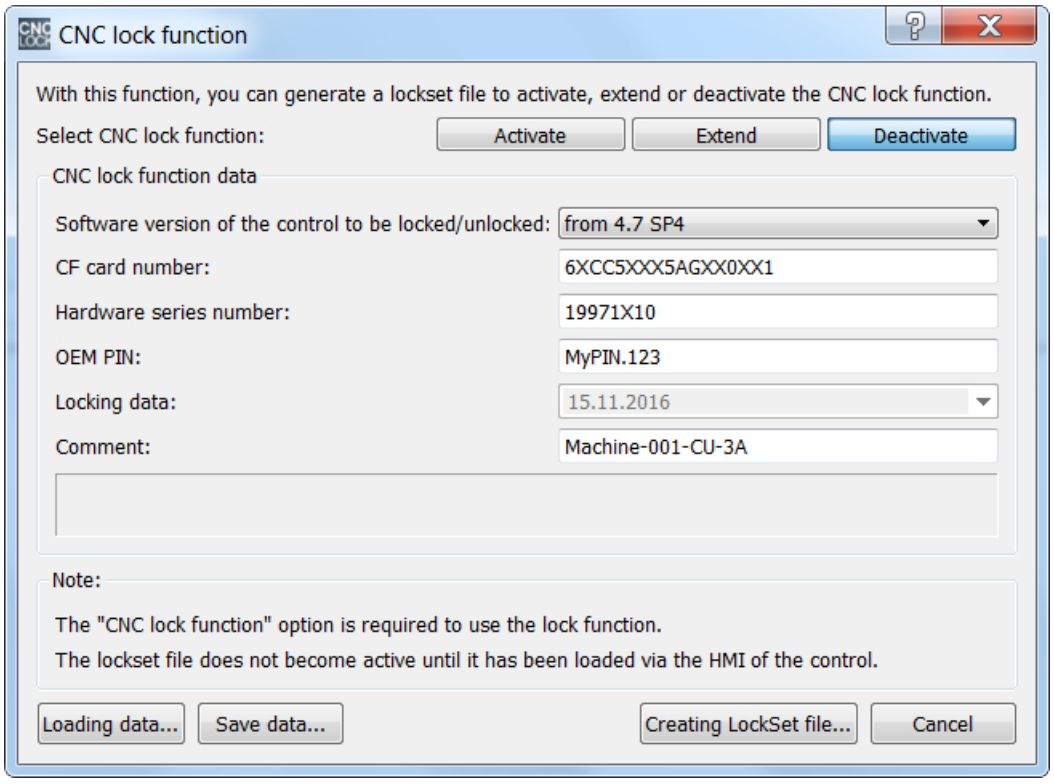
CF カードとコントロールシステム(Panel Processing unit、PPU)のシリアル番号、および OEM PIN は、CNC ロック機能の初回起動時に使用した値と一致するようにしてください。

注記

CF カードとコントロールシステム(PPU)の**シリアル番号**は、SINUMERIK Operate ユーザーインターフェースの[診断|バージョン|ハードウェア| NCU/PLC | ncu1]操作エリアで確認できます。

- **CF カード:**[CF カード| SerialNo.]領域
- **PPU**[SINUMERIK 828D PPU | SerialNo.]領域

このデータは、Access MyMachine で[CNC ロック機能]ダイアログウィンドウに入力します(メインメニューから[工具| CNC ロック機能]を呼び出します)。**[解除]**ボタンを選択する必要があります:



必要なすべてのデータを入力した後、[lockset ファイルの作成]を使用して解除する lockset ファイルを生成する必要があります。

lockset ファイルの伝送

新しい lockset ファイルをコントロールシステムに伝送してください。機械メーカは、Ethernet 接続で直接、コントロールシステムに伝送できます。

このファイルは、/System-CF-Card/User/sinumerik/data/license にあります。

または、機械メーカが、新しい lockset ファイルをエンドユーザーに送信し、エンドユーザーがコントロールシステムに伝送します。

lockset ファイルの読み取り

lockset ファイルのインポートは、ユーザーインタフェース、[コミッショニング|ライセンス|ライセンスキーのインポート]操作エリアから開始できます。

インポートには、アクセスレベルは必要ありません。

lockset ファイルのインポート時に異常が発生しない場合は、CNC ロック機能が解除されています。

注記

lockset ファイルのインポート時に異常が発生する場合は、異常別アラームが発行されます。CNC ロック機能の状態は未変更のままです。

注記

エンドユーザーが、ロック期日を**無効にした後**に、コントロールシステムのすべてのコンポーネントに関する**コミッショニングアーカイブ**一式を作成することを推奨します。必要に応じて、CNC ロック機能を再起動することなく、このコミッショニングアーカイブを使用してコントロールシステムを再コミッショニングすることができます。

確認

ロック期日がコントロールシステムで正しく解除されたかどうかは、以下のユーザーインターフェースにロック期日が表示されなくなることでチェックできます。

SINUMERIK Operate ユーザーインターフェース:[コミッショニング|ライセンス]操作エリア

15.7.8 故障したコントロールシステムハードウェア(PPU)の交換

故障したコントロールシステムハードウェア(PPU)の交換時には、新しいコントロールシステムハードウェアで使用できるように、システム CF カードを機械から取り外さないでください。

新しいハードウェアのコントロールシステムの初回の電源投入後に、前のコントロールシステムハードウェアで作成した既存のコミッショニングアーカイブをインポートしてください。コントロールシステムの次回の電源投入後に、アラーム 8062:「CNC ロック機能:機能を実行できませんでした:原因 2」(ハードウェアが交換されました)が表示され、NC スタートが無効になります。アラームの原因は、新しいコントロールシステムハードウェアの新しいシリアル番号です。

新しい lockset ファイルの要求

コントロールシステムを再度ロック解除するために、エンドユーザーは、コントロールシステムに該当する新しい lockset ファイル(lockset.clc)を、CNC ロック機能を作成した会社(機械メーカまたは代理店)に要求してください。

lockset ファイルの生成

lockset ファイルを生成するには、以下のデータが必要です。

- 機械に残っている CF カードのシリアル番号
- **新しい**コントロールシステム(PPU)のシリアル番号
- 最初に割り当てた OEM PIN
- 最後のロック期日または新しいロック期日

注記

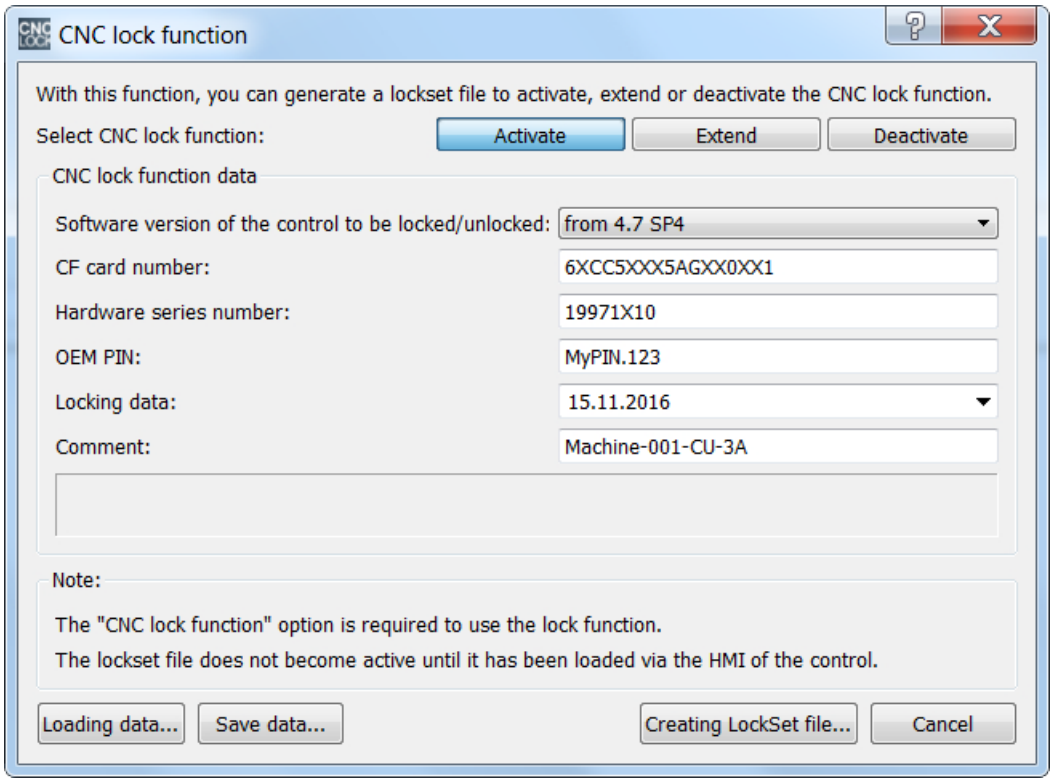
故障したコントロールシステムハードウェア(PPU)を交換する場合は、最初に割り当てた **OEM PIN** を、CNC ロック機能を作成した会社に知らせてください。

注記

CF カードとコントロールシステム(PPU)の**シリアル番号**は、SINUMERIK Operate ユーザーインターフェースの[診断|バージョン|ハードウェア| NCU/PLC | ncu1]操作エリアで確認できます。

- **CF カード:**[CF カード| SerialNo.]領域
- **PPU**[SINUMERIK 828D PPU | SerialNo.]領域

このデータは、Access MyMachine で[CNC ロック機能]ダイアログウィンドウに入力します(メインメニューから[工具| CNC ロック機能]を呼び出します)。**[起動]**ボタンを選択する必要があります:



必要なすべてのデータを入力した後、[lockset ファイルの作成]を使用してロック解除する lockset ファイルを生成する必要があります。

lockset ファイルの伝送

新しい lockset ファイルをコントロールシステムに伝送してください。機械メーカは、Ethernet 接続で直接、コントロールシステムに伝送できます。

このファイルは、/System-CF-Card/User/sinumerik/data/license にあります。

または、機械メーカが、新しい lockset ファイルをエンドユーザーに送信し、エンドユーザーがコントロールシステムに伝送します。

lockset ファイルの読み取り

注記

lockset ファイルを**インポートする前**に、新しいコントロールシステムハードウェアの**時刻**を正しく設定してください。これは、CNC ロック機能の起動時点の時刻が、監視の開始値として保存されるためです。CNC ロック機能を作成した会社は、サービス員が交換と時刻設定を行うことによって、操作を保護することができます。

lockset ファイルのインポートは、ユーザーインタフェース、[コミッショニング|ライセンス|ライセンスキーのインポート]操作エリアから開始できます。

インポートには、アクセスレベルは必要ありません。

lockset ファイルのインポート時に異常が発生しない場合は、CNC ロック機能が新しいコントロールシステムで有効になっています。

15.7.9 故障した CF カードの交換

故障したシステム CF カードは、取り外して SIEMENS に送付してください。その受領後に SIEMENS は、新しい CF カードを返送します。

新しい CF カードのコントロールシステムの初回の電源投入後に、前のコントロールシステムハードウェアで作成した既存のコミッショニングアーカイブをインポートしてください。コントロールシステムの次回の電源投入後に、アラーム 8062:「CNC ロック機能:機能を実行できませんでした:原因 1」(CF カードが交換されました)が表示され、NC スタートがロックされます。アラームの原因は、新しい CF カードの新しいシリアル番号です。

新しい lockset ファイルの要求

コントロールシステムを再度ロック解除するために、エンドユーザーは、コントロールシステムに該当する新しい lockset ファイル(lockset.clc)を、CNC ロック機能を作成した会社(機械メーカまたは代理店)に要求してください。

lockset ファイルの生成

lockset ファイルを生成するには、以下のデータが必要です。

- 新しい **CF カード**のシリアル番号
- コントロールシステム(PPU)のシリアル番号
- 最初に割り当てた OEM PIN
- CNC ロック機能の状態(最後のロック期日、新しいロック期日、ロック期日なし)に応じたロック期日

注記

故障した CF カードを交換する場合は、最初に割り当てた **OEM PIN** を、CNC ロック機能を作成した会社に知らせてください。

注記

CF カードとコントロールシステム(PPU)の**シリアル番号**は、SINUMERIK Operate ユーザーインターフェースの[診断|バージョン|ハードウェア| NCU/PLC | ncu1]操作エリアで確認できます。

- **CF カード:**[CF カード| SerialNo.]領域
- **PPU**[SINUMERIK 828D PPU | SerialNo.]領域

このデータは、Access MyMachine で[CNC ロック機能]ダイアログウィンドウに入力します(メインメニューから[工具| CNC ロック機能]を呼び出します)。**[拡張]**ボタンまたは**[解除]**ボタンを選択する必要があります:

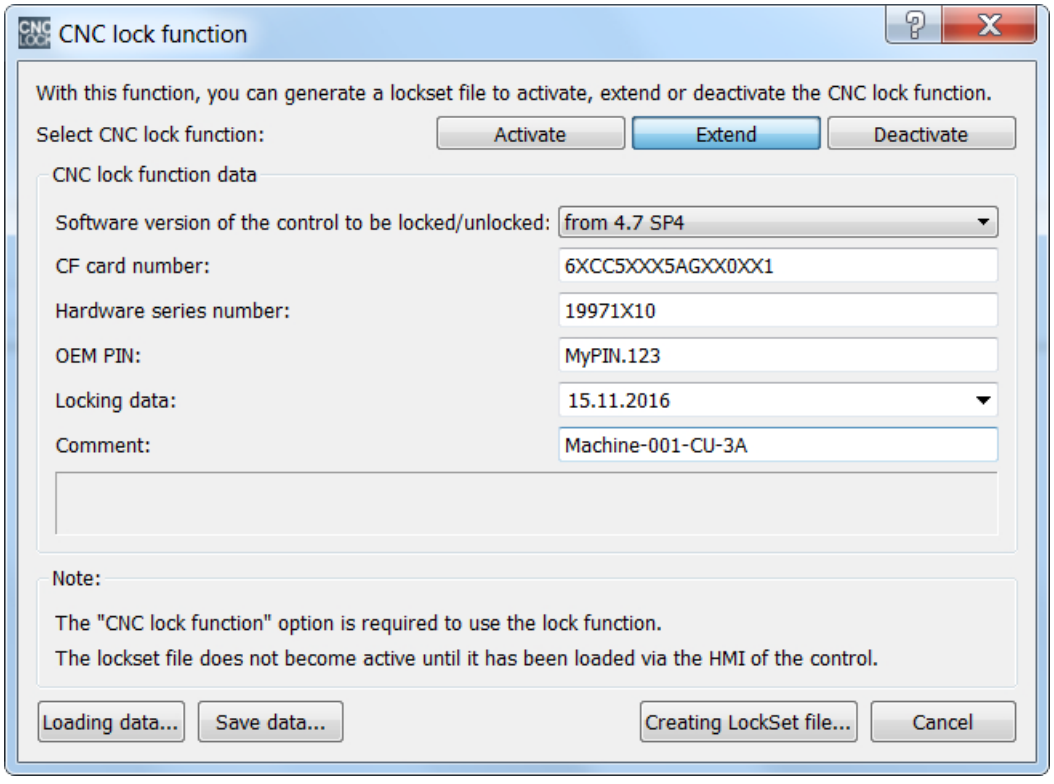


図 15-3　CNC ロック機能の更新

必要なすべてのデータを入力した後、[lockset ファイルの作成]を使用してロック解除する lockset ファイルを生成する必要があります。

lockset ファイルの伝送

新しい lockset ファイルをコントロールシステムに伝送してください。機械メーカは、Ethernet 接続で直接、コントロールシステムに伝送できます。

このファイルは、/System-CF-Card/User/sinumerik/data/license にあります。

または、機械メーカが、新しい lockset ファイルをエンドユーザーに送信し、エンドユーザーがコントロールシステムに伝送します。

lockset ファイルの読み取り

注記

lockset ファイルを**インポートする前**に、新しいコントロールシステムハードウェアの**時刻**を正しく設定してください。これは、CNC ロック機能の起動時点の時刻が、監視の開始値として保存されるためです。CNC ロック機能を作成した会社は、サービス員が交換と時刻設定を行うことによって、操作を保護することができます。

lockset ファイルのインポートは、ユーザーインタフェース、[コミッショニング|ライセンス|ライセンスキーのインポート]操作エリアから開始できます。

インポートには、アクセスレベルは必要ありません。

lockset ファイルのインポート時に異常が発生しない場合は、CNC ロック機能が新しいコントロールシステムで有効になっています。

15.7.10 OEM PIN を忘れた場合

CNC ロック機能を作成した会社(機械メーカまたは代理店)が、初回作成時に割り当てた OEM PIN を忘れると、関連するコントロールシステムの有効な lockset ファイルを作成できなくなります。

機械のロック解除

CNC ロック機能を作成した会社が機械を操作できるようにするには、技術者が SIEMENS ホットラインに連絡し、以下の情報を提供しなければなりません。

- 「CNC ロック機能」オプションのライセンス証明書(CoL)
- CF カードのシリアル番号
- コントロールシステム(PPU)のシリアル番号
- CNC ソフトウェアのソフトウェアバージョン

注記

CF カードとコントロールシステム(PPU)の**シリアル番号**は、SINUMERIK Operate ユーザーインターフェースの[診断|バージョン|ハードウェア| NCU/PLC | ncu1]操作エリアで確認できます。

- CF カード:[CF カード| SerialNo.]領域
- PPU[SINUMERIK 828D PPU | SerialNo.]領域

CNC ソフトウェアの**ソフトウェアバージョンは、** SINUMERIK Operate ユーザーインターフェースの[診断|バージョン|実際のバージョン]操作エリアで確認できます。

オリジナルのライセンス証明書(CoL)を SIEMENS に送付してください。SIEMENS によるロック解除は有料となります。

CNC ロック機能を作成した会社は、機械をロック解除するための lockset ファイルをホットラインから受け取ります。ただし、ロック解除されるのは、コントロールシステム(PPU)のハードウェアのみです。PLC プロジェクトはロック解除されません。このため、機械に適切な元の PLC プロジェクトが存在していなければなりません。

機械のロック解除は、CNC ロック機能を作成した会社のサービス技術者と、機械のエンドユーザー(操作する会社)がオンサイトでおこなってください。

この手順の詳細情報は、SIEMENS ホットラインから入手できます。

15.7.11 その他の情報

プロジェクトファイル

注記

機械メーカは、シリアル番号と OEM PIN の割り当てに関する文書を提供する義務があります。

Access MyMachine を使用して、以下のデータを含む、暗号化されていないプロジェクトファイル(「**U**ser-**C**NC-**L**ock-**S**et (ユーザー CNC ロックセット)」ファイルを意味する「ucls」) を作成できます。

- CF カードのシリアル番号
- コントロールシステム(PPU)のシリアル番号
- OEM PIN

- 作成日時
- ロック期日

この機能は、[CNC ロック機能]ダイアログウィンドウの[データの保存...]ボタンを使用して呼び出します。

[データのロード...]ボタンをクリックすると、プロジェクトファイルに格納された、暗号化されていないデータが再インポートされます。

日付または時刻の設定の誤り

起動した CNC ロック機能で日付または時刻について決定された設定が正しくない場合、次の操作が開始されます。

- 次のアラーム 8065 が出力されます:「CNC ロック機能:日付/時刻を正しく設定してください。」
- NC スタートのブロック

ロックを取り消すには、コントローラをオフにする前に不正な設定を修正する必要があります。

通知
恒久的なロック コントローラをオフにする前に、不正な日付または時刻の設定を修正する必要があります。そうしない場合は、次のアラーム 8064 により恒久的なロックが発生する恐れがあります:「CNC ロック機能:ロック期日が過ぎています。NC は起動できません。」 解決策:コントローラをオフにする**前**に、日付/時刻を正しく設定してください。

CNC ロック機能を起動したまま、将来の日付を設定すると、次のアラームが出力されます:

アラーム 8066「CNC ロック機能:日付を変更すると、残りランタイムが減少します」。

コントローラシステムのスイッチをオフにしないかぎり、日付は引き続き修正できます。

通知
耐用年数の減少 コントロールシステムをオフにした後に、設定された将来の日付が実際の日付と見なされ、再設定できなくなります。この場合、耐用年数がロック期日までに減少します。 解決策：コントローラをオフにする**前**に、日付を正しく設定してください。

詳細情報

参照:

- SINUMERIK Integrate Access MyMachine/P2P (PC)のオンラインヘルプ
- PLC プログラミングツールのオンラインヘルプ

R1:レファレンス点復帰 16

16.1 概略説明

機能

機械軸を原点確立すると、機械軸の座標系が機械の座標系と同期されます。機械軸は機械原点に移動され、機械軸の現在位置が原点に設定されます。

機械構成の結果、機械原点に直接アプローチできない場合は、レファレンス点が機械軸の移動範囲内で定義され、これを使用して機械軸が同期されます。機械原点を基準としたレファレンス点の位置を認識してください。原点確立時に、実際の機械軸位置が、この値に設定されます。

検出器と原点確立の方法

原点確立時に、以下の検出器と原点確立タイプで機械軸を同期することができます。

- 測定システム
  - 1 つの以上の原点マークがあるインクリメンタルロータリー検出器
  - インクリメンタルリニア検出器
  - 絶対番地化原点マークのあるロータリー検出器(ハイデンハイン殿製)
  - 絶対番地化原点マークのあるリニア検出器(ハイデンハイン殿製)
  - 絶対値ロータリー検出器
  - 絶対値リニア検出器
- レファレンス点復帰動作の方法
  - 近接スイッチと 1 エッジおよび 2 エッジ検出のあるインクリメンタル検出器による原点確立
  - 原点復帰カムを近接スイッチと置き換えるインクリメンタル検出器による原点確立
  - 主軸の用途のためにアプローチ速度を設定された近接スイッチのあるインクリメンタル検出器による原点確立
  - 2 または 4 原点マークの通過による絶対番地化原点マークのある検出器による原点確立
  - 検出器調整を使用する不使用検出器の原点確立
  - フォローアップモードでの原点確立
  - ドライブのカムスイッチによる原点確立

開始

機械軸の原点確立は、手動で、またはパートプログラムから開始できます。

- 手動:運転モード JOG および MDI、運転機能 REF
- パートプログラム:G74 指令

16.2 軸別原点確立

レファレンス点復帰を使用して軸別に原点確立する場合は、原点確立する機械軸ごとに、個別に操作を開始してください。

運転機能およびモードの選択

機械軸のレファレンス点復帰を開始する前に、まず関連するモードグループを JOG または MDI モードにしてください。

DB11、... DBX0.2 (JOG モード有効)

DB11、... DBX0.1 (MDI モード有効)

次に運転機能 REF (レファレンス点復帰)を選択してください。

DB11、... DBX1.2 (REF 運転機能)

レファレンス点復帰の開始

軸別のレファレンス点復帰では、各機械軸は個別に起動してください。

レファレンス点復帰は、軸別の移動キーにより開始されます。

DB31、... DBX4.6 (マイナス移動キー)

DB31、... DBX4.7 (プラス移動キー)

有効な方向

誤った操作を回避するために、許可方向をパラメータ設定してください。有効な方向は、レファレンス点復帰を開始する移動キーを指定します。

MD34010 $MA_REFP_CAM_DIR_IS_MINUS = <値>

<値>	意味
0	プラス方向でレファレンス点復帰
1	マイナス方向のレファレンス点復帰

ジョグモード

次のマシンデータ要素は、方向キーが一度押されただけでレファレンス点復帰が完結するか、または安全上の理由のためにオペレータが方向キーを押したままにしておく必要がある (ジョグ動作) かどうかを指定するために使用できます。

MD11300 $MN_JOG_INC_MODE_LEVELTRIGGRD (ジョグモードの INC および REF)

機械オペレータが方向キーを解除した場合、機械軸はゼロ速度に減速されます。レファレンス点復帰は中止されません。レファレンス点復帰は、次に方向キーが押されると継続します。

原点確立状態

機械軸の原点確立状態は、レファレンス点復帰が開始するとリセットされます。

DB31、... DBX60.4 (原点確立済み/原点同期済み 1)

DB31、... DBX60.5 (原点確立済み/原点同期済み 2)

DB21、... DBX36.2 (レファレンス点復帰が必要なすべての軸が原点確立済み)

絶対番地化検出器

絶対番地化検出器では、レファレンス点復帰を任意の移動キーにより開始できます。

手順

機械オペレータや工作機械メーカは(PLC ユーザープログラムを介して)、機械軸を適切な順序で確実に原点確立する義務があります。

- 機械オペレータ
  機械オペレータは機械軸を指令された順序で起動してください。
- 工作機械メーカ
  工作機械メーカの PLC ユーザープログラムで、機械軸が適切な順序でのみ起動するようにします。

複数の機械軸の同時レファレンス点復帰

制御装置に応じて、複数の機械軸を同時に原点確立できます。

レファレンス点復帰の完了

機械軸のレファレンス点復帰が正常終了したことは、原点確立状態の設定によって確認できます。

DB31、... DBX60.4 (原点確立済み/原点同期済み 1)

DB31、... DBX60.5 (原点確立済み/原点同期済み 2)

レファレンス点復帰のキャンセル

軸別のレファレンス点復帰では、機械軸は機械軸のマスタチャネルとして割り当てられたチャネルで移動します。

MD30550 $MA_ AXCONF_ASSIGN_MASTER_CHAN

レファレンス点復帰を中止するためには、モードグループリセットまたは機械軸のマスタチャネルのチャネルリセットのいずれかを起動してください。

DB11、... DBX0.7 (モードグループリセット)

DB21、... DBX7.7 (チャネルリセット)

動作がキャンセルされたときにレファレンス点復帰を正常終了していないすべての機械軸は、状態が「原点確立未完」のままになりす。

DB31、... DBX60.4 (原点確立済み/原点同期済み 1)

DB31、... DBX60.5 (原点確立済み/原点同期済み 2)

16.3 チャネル別の原点確立

チャネル別の原点確立では、レファレンス点復帰が開始されると、チャネルのすべての機械軸がパラメータ設定された方法で原点確立されます。

運転機能およびモードの選択

機械軸のレファレンス点復帰を開始する前に、関連するモードグループを JOG またはMDI 操作モードに切り替えてください。

DB11、... DBX0.2 (JOG モード有効)

DB11、... DBX0.1 (MDI モード有効)

次に運転機能 REF (レファレンス点復帰)を選択してください。

DB11、... DBX1.2 (REF 運転機能)

軸の順序のパラメータ設定

次のマシンデータは、チャネルの機械軸が原点確立される順序を指定するために使用されます。

MD34110 $MA_REFP_CYCLE_NR = <番号>

<番号>	意味
-1	機械軸は、チャネルで NC スタートのために原点確立される必要はありません。
0	機械軸は、チャネル別のレファレンス点復帰には関与しません。
1 - 15	チャネル別のレファレンス点復帰での順番。

機械軸は、番号順で原点確立されます。

同じ番号の機械軸は、同時に原点確立されます。

複数の機械軸の同時レファレンス点復帰

制御装置に応じて、複数の機械軸を同時に原点確立できます。

レファレンス点復帰の開始

チャネル別のレファレンス点復帰は、下記の信号により開始されます。

DB21、... DBX1.0 (レファレンス点復帰動作起動)

チャネル別のレファレンス点復帰の状態は、下記の信号を使用してチャネルごとに示されます。

DB21、... DBX33.0 (レファレンス点復帰動作有効)

原点確立状態

機械軸の原点確立状態は、レファレンス点復帰が開始するとリセットされます。

DB31、... DBX60.4 (原点確立済み/原点同期済み 1)

DB31、... DBX60.5 (原点確立済み/原点同期済み 2)

レファレンス点復帰の完了

関与するすべての機械軸に対してチャネル別のレファレンス点復帰が正常終了するとすぐに、以下を使用してこれが通知されます。

DB21、... DBX36.2 (レファレンス点復帰が必要なすべての軸が原点確立済み)

レファレンス点復帰のキャンセル

チャネル別のレファレンス点復帰では、機械軸はその軸がチャネル軸として現在割り当てられているチャネルで移動します。

レファレンス点復帰を中止するためには、モードグループリセットと対応するチャネルのチャネルリセットのいずれかを起動してください。

DB11、... DBX0.7 (モードグループリセット)

DB21、... DBX7.7 (チャネルリセット)

動作がキャンセルされたときにレファレンス点復帰を正常終了していないすべての機械軸は、状態が「原点確立未完」のままになりす。

DB31、... DBX60.4 (原点確立済み/原点同期済み 1)

DB31、... DBX60.5 (原点確立済み/原点同期済み 2)

16.4 パートプログラムによるレファレンス点復帰(G74)

機能

命令 G74 を使用して、初回、またはそれ以降でも、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機械軸を原点確立できます。

たとえば以下の後に、原点確立を繰り返してください。

- 現在値オフセット PRESETON
  参照先:『プログラミングマニュアル 上級編』、「座標変換(FRAMES)」の章の「PRESETON によるプリセットオフセット 」
- 機械軸のパーキング (ページ 145)
- 位置検出器のエンコーダ限界周波数の超過

プログラミング

構文

```
G74 <機械軸> {<機械軸>}
```

意味

G74:	機械軸の原点確立の命令	
	単独ブロック指令	あり
	有効性:	ノンモーダル
<機械軸>:	機械軸の名前は、(MD10000 $MN_AXCONF_MACHAX_NAME_TAB)で指定してください。 機械軸は、パートプログラムまたはシンクロナイズドアクションが実行されるチャネルのチャネル軸にしてください。	

リセット動作

モードグループリセットまたはチャネルリセットすると、プログラミングされたすべての機械軸に対してレファレンス点復帰が中止されます。

- DB11、... DBX0.7 (モードグループリセット)
- DB21、... DBX7.7 (チャネルリセット)

動作がキャンセルされたときにレファレンス点復帰を正常終了していないすべての機械軸は、状態が「原点確立未完」のままになりす。

- DB31、... DBX60.4 (原点確立済み/原点同期済み 1) == 0
- DB31、... DBX60.5 (原点確立済み/原点同期済み 2) == 0

16.5 インクリメンタルエンコーダによる原点確立

16.5.1 ハードウェア信号

機械構成と使用しているインクレメンタル検出器の特性に合わせて、異なるハードウェア信号を接続してください。

レファレンス点スイッチ

- 接続
  レファレンス点スイッチ信号は、PLC I/O モジュールのディジタル入力部または NCU X142 インタフェースの高速入力部に接続できます。
- NC/PLC インタフェース信号
  レファレンス点スイッチ信号は、PLC ユーザープログラムから軸の NC/PLC インタフェースに伝送してください。
  DB31、... DBX12.7 (レファレンス点復帰の減速)

原点マーク選択

軸または主軸のレファレンス点復帰中に、検出器の複数の原点マークが検出される場合は(モータとエンコーダの間の計測ギアなど)、追加の近接スイッチ信号により特定の原点マークを選択してください。

- 接続
  近接スイッチは、NCU X122 または X132 インタフェースの高速ディジタル入力部に接続してください。
- 起動
  近接スイッチ信号を評価するために、近接スイッチを接続するディジタル入力部を軸/主軸のインバータパラメータ p0493 で選択してください。

等価ゼロマーク

使用する検出器が原点マーク信号を出力しない場合は、近接スイッチ信号により相当する原点マークを作成できます。

- 接続
  近接スイッチは、NCU X122 または X132 インタフェースの高速ディジタル入力部に接続してください。
- 起動
  近接スイッチ信号を評価するために、近接スイッチを接続するディジタル入力部を軸/主軸のインバータパラメータ p0494 または p0495 で選択してください。

概要

信号	接続:デジタル入力の経由方法	設定
レファレンス点スイッチ	外部 PLC I/O モジュールまたは NCU:X142	PLC ユーザープログラム:DB31、...DBX12.7
原点マーク選択	NCU:X122 または X132	インバータパラメータ:p0493
外部原点マークまたは相当する原点マーク	NCU:X122 または X132	インバータパラメータ:p0494 または p0495

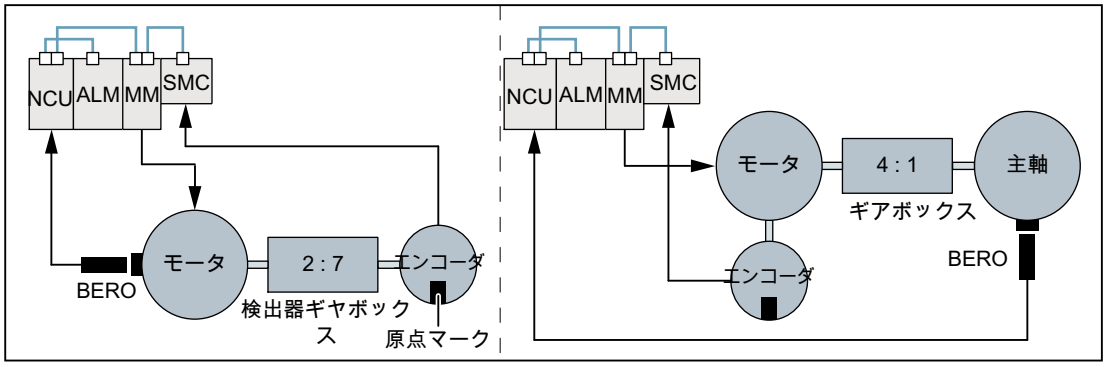
参照先

- NCU インタフェース:SINUMERIK 840D sl マニュアル、NCU7x0.3 PN、「接続」の「ディジタル I/O」の章
- インバータパラメータ:『SINAMICS S120/S150 リストマニュアル』

16.5.2 原点マーク選択

機能

インクリメンタル検出器の原点確立は、機械軸の移動範囲全体に関連したエンコーダ原点マークの一義的な位置に基づいています。機械固有の条件により、機械軸の移動範囲で複数のエンコーダ原点マークが検出された場合(たとえば、下の図を参照してください)、レファレンス点を明確に特定するために、近接スイッチを機械に組み込んでください。次に、近接スイッチ信号とエンコーダの原点マークの組合わせからレファレンス点の位置が導出されます。



BER 近接スイッチの商標
O

図 16-1 モータとエンコーダ間の検出器ギヤまたはモータと主軸間の減速ギヤ

パラメータ設定

NC:原点復帰動作モード

「インクリメンタル検出期、ロータリー検出器、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」をレファレンス点復帰モードとしてパラメータ設定してください。

MD34200 $MA_ENC_REFP_MODE[<軸>] = 1

ドライブ:原点マークの選択

近接スイッチの接続先となる NCU インタフェースのディジタル入力は、パラメータ p0493 で設定してください。

注記

原点マークの選択

近接スイッチ信号の処理は、ドライブでのみ行われます。接続とパラメータ設定については、「ハードウェア信号 (ページ 1503)」の章を参照してください。

16.5.3 タイミング図

インクリメンタル検出器によるレファレンス点復帰は、次の 3 つのフェーズに分割できます。

- フェーズ 1: "フェーズ 1:レファレンス点スイッチへの移動 (ページ 1508)"
- フェーズ 2: "フェーズ 2:原点マークとの同期 (ページ 1511)"
- フェーズ 3: "フェーズ 3: レファレンス点への移動 (ページ 1516)"

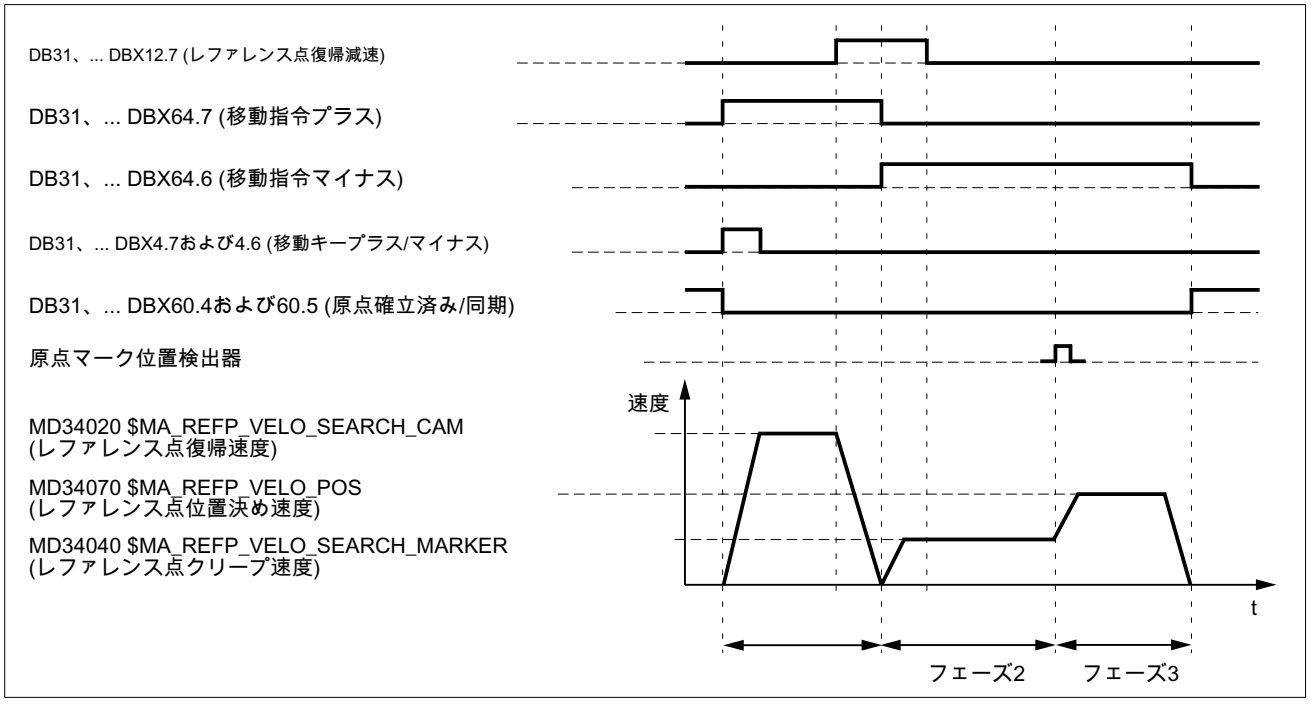


図 16-2　インクリメンタル検出器によるレファレンス点復帰の際のタイミング図(例)

16.5.4 フェーズ 1:レファレンス点スイッチへの移動

フェーズ 1:グラフ表示

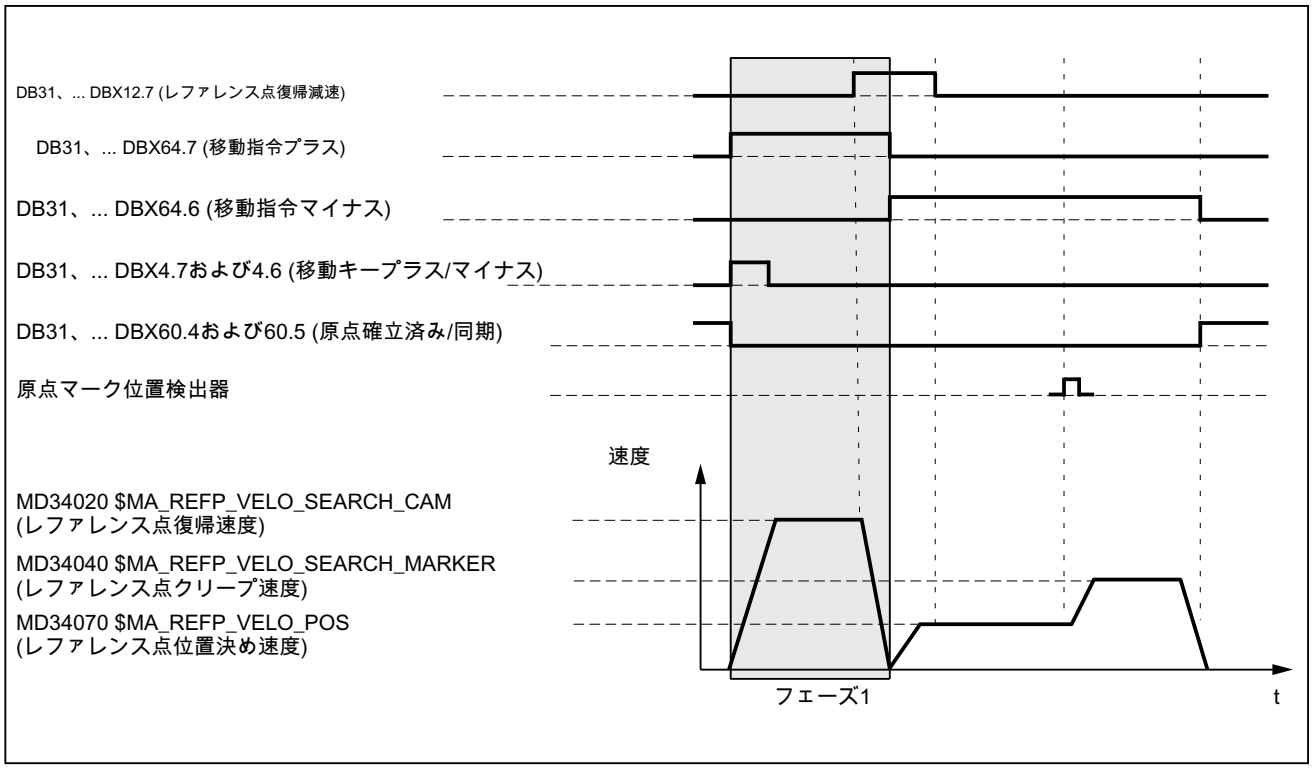


図 16-3 フェーズ 1:レファレンス点スイッチへの移動

フェーズ 1:開始

レファレンス点復帰を開始するには、「軸別原点確立 (ページ 1498)」と「軸別原点確立 (ページ 1498)」の章を参照してください。

フェーズ 1:処理

フェーズ 1 では、レファレンス点スイッチに関連した機械軸の位置により、次の 3 つのケースを区別します。

1. 機械軸が、レファレンス点スイッチの前にあるとき
2. 機械軸が、レファレンス点スイッチ上にあるとき
3. 機械軸に、レファレンス点スイッチがないとき

ケース 1:機械軸が、レファレンス点スイッチの前にあるとき

レファレンス点復帰の開始後、機械軸はパラメータ設定された方向に、パラメータ設定されたレファレンス点復帰速度で移動を開始します。

- MD34010 $MA_REFP_CAM_DIR_IS_MINUS (マイナス方向のレファレンス点復帰)
- MD34020 $MA_REFP_VELO_SEARCH_CAM (レファレンス点復帰速度)

レファレンス点スイッチの到達を、PLC ユーザープログラムでディジタル入力の確認により検出し、次のインタフェース信号により NC に通知してください。

DB31、... DBX12.7 = 1 (レファレンス点復帰減速)

NC/PLC インタフェース信号を検出すると、機械軸はゼロ速度まで減速します。これにより、最低でも距離 s_{min} を移動します。これにより、フェーズ 2 で機械軸は、パラメータ設定されたレファレンス点クリープ速度でレファレンス点スイッチから離れます。

$$s_{min} = \frac{(\text{MD34040 \$MA_REFP_VELO_SEARCH_MARKER})2}{2 * \text{MD32300 \$MA_MAX_AX_ACCEL}}$$

フェーズ 1 は、これで完了です。レファレンス点復帰は、フェーズ 2 により継続されます。

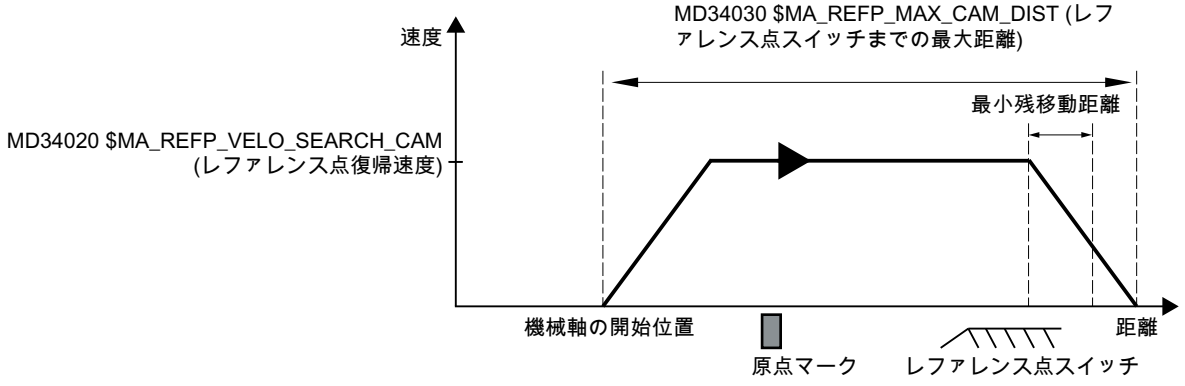


図 16-4　減速のための最小距離

ケース 2:機械軸が、レファレンス点スイッチ上にあるとき

機械軸は、開始位置のままです。
フェーズ 1 は、これで完了です。レファレンス点復帰は、フェーズ 2 により継続されます。

ケース 3:機械軸に、レファレンス点スイッチがないとき

レファレンス点スイッチのない機械軸は、開始位置のままです。

これらには、以下の例があります。

- 移動範囲全体に沿って 1 つのみ原点マークがある機械軸
- 1 回転に 1 つの原点マークだけがある回転軸

レファレンス点スイッチのない機械軸の場合、以下のマシンデータにゼロを入力してください。

MD34000 $MA_REFP_CAM_IS_ACTIVE = 0 (レファレンス点スイッチのある軸)

フェーズ 1 は、これで完了です。レファレンス点復帰は、フェーズ 2 により継続されます。

フェーズ 1:特徴

- 送り速度オーバライドが有効です。
- 送り停止(チャネル別と軸別)が有効です。
- NC ストップと NC スタートが有効です。
- 下記でパラメータ設定された最大距離内でレファレンス点スイッチに到達しない場合、機械軸は停止します。
  MD34030 $MA_REFP_MAX_CAM_DIST (レファレンス点スイッチまでの最大距離)

下記も参照

チャネル別の原点確立 (ページ 1500)

16.5.5 フェーズ 2:原点マークとの同期

フェーズ 2:グラフ表示

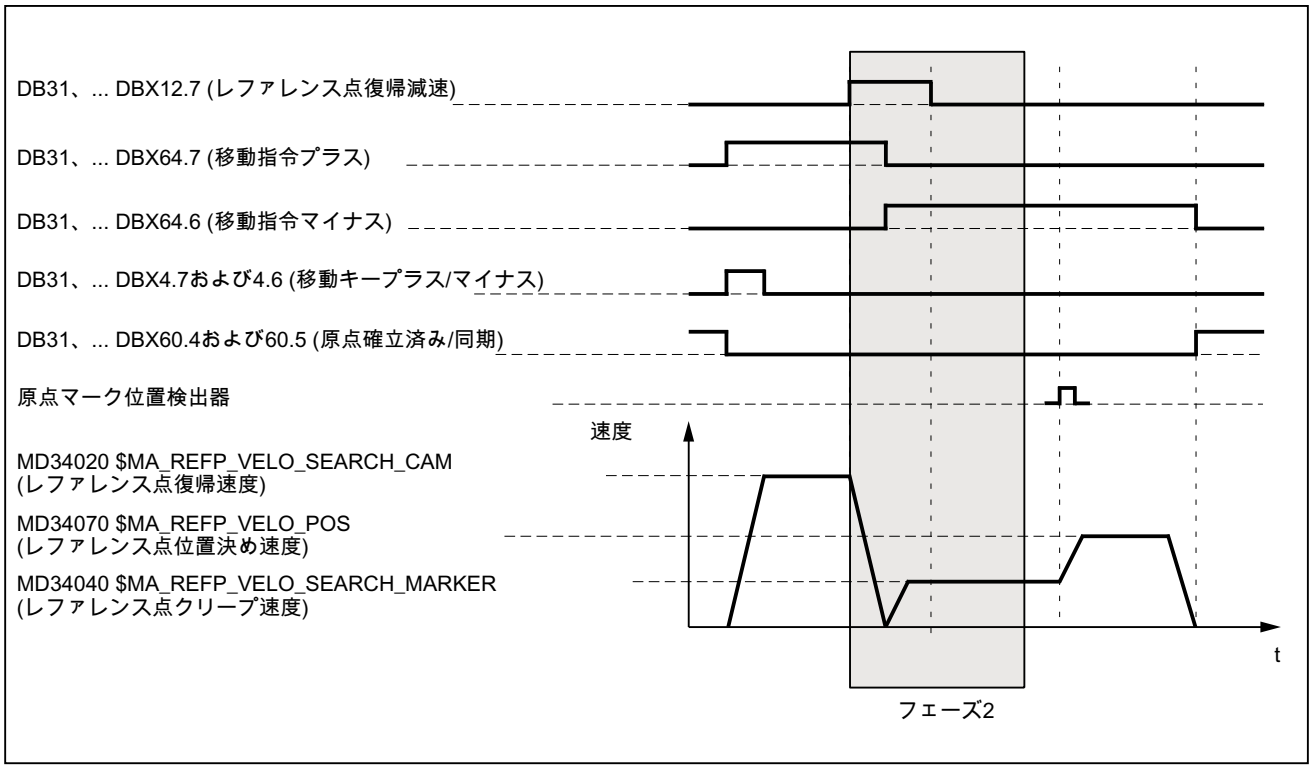


図 16-5　フェーズ 2:原点マークとの同期

フェーズ 2:開始

フェーズ 1 がアラームなしで完了すると、フェーズ 2 が自動的に開始します。

初期状態

機械軸は、レファレンス点スイッチの上にあります。

原点マーク検索方向:

原点マーク検索の方向は、次のマシンデータでの設定によって決まります。

- MD34010 $MA_REFP_CAM_DIR_IS_MINUS (マイナス方向のレファレンス点復帰)
- MD34050 $MA_REFP_SEARCH_MARKER_REVERSE (レファレンス点スイッチでの方向の反転)

フェーズ 2:処理

フェーズ 2 での同期制御は、レファレンス点スイッチの立ち上がりエッジまたは立下りエッジを使用しておこなうことができます。パラメータ設定は以下によっておこなわれます。

MD34050 $MA_REFP_SEARCH_MARKER_REVERSE[<軸>] = <値>

値	意味
0	レファレンス点スイッチ信号の立ち下がりエッジによる同期
1	レファレンス点スイッチ信号の立ち上がりエッジによる同期

注記

レファレンス点スイッチのアプローチでの機械軸の実際の速度が、パラメータ設定された許容範囲制限値内でフェーズ 2 の目標速度に到達していない場合、フェーズ 1 が再開されます。これはたとえば、レファレンス点復帰が開始するときに、機械軸がレファレンス点スイッチに位置している場合です。

MD35150 $MA_SPIND_DES_VELO_TOL (主軸速度許容範囲)

ケース 1:レファレンス点スイッチ信号の立ち下がりエッジによる同期

レファレンス点スイッチ信号の立ち下がりエッジによる同期のとき、機械軸はパラメータ設定されたレファレンス点クリープ速度まで、パラメータ設定されたレファレンス点復帰方向(フェーズ 1 の移動方向)と逆の方向に加速します。

レファレンス点スイッチから離れた後、機械軸は次のエンコーダ原点マークを待ちます。
DB31、... DBX12.7 == 0

エンコーダ原点マークが検出されるとすぐに、フェーズ 2 は終了になります。機械軸は一定の速度を維持し、レファレンス点復帰はフェーズ 3 により継続されます。

- MD34040 $MA_REFP_VELO_SEARCH_MARKER (レファレンス点クリープ速度)

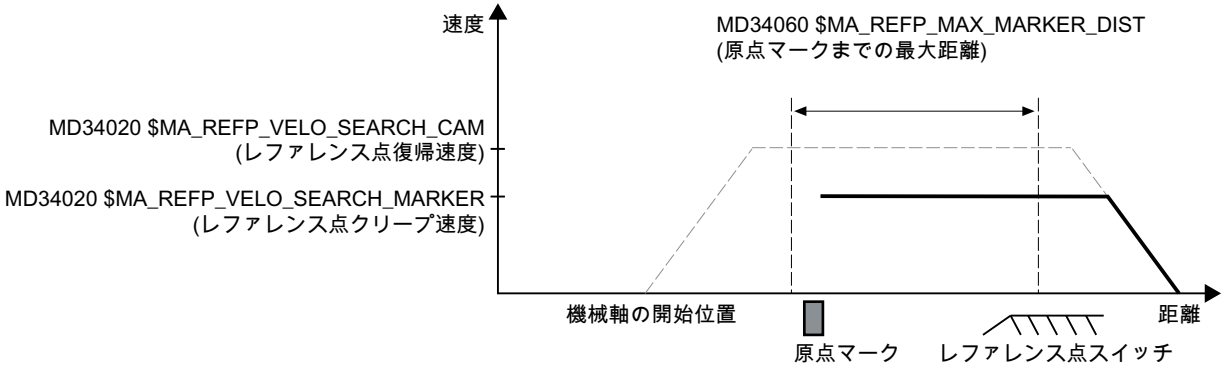


図 16-6　レファレンス点スイッチ信号の立ち下がりエッジによる同期

ケース 2:レファレンス点スイッチ信号の立ち上がりエッジによる同期

レファレンス点スイッチ信号の立ち上がりエッジによる同期のとき、機械軸はパラメータ設定されたレファレンス点復帰速度まで、パラメータ設定されたレファレンス点復帰方向(フェーズ 1 の移動方向)と逆の方向に加速します。

- MD34020 $MA_REFP_VELO_SEARCH_CAM (レファレンス点復帰速度)
- MD34010 $MA_REFP_CAM_DIR_IS_MINUS (マイナス方向のレファレンス点復帰)

レファレンス点スイッチから離れた後、機械軸は減速して停止します。
DB31、... DBX12.7 == 0

それから機械軸は、パラメータ設定されたレファレンス点クリープ速度で、レファレンス点スイッチに戻る方向へ移動します。

MD34040 $MA_REFP_VELO_SEARCH_MARKER (レファレンス点クリープ速度)

レファレンス点スイッチに到達後(DB31、... DBX12.7 = 1)、機械軸は次のエンコーダ原点マークを待ちます。

エンコーダ原点マークが検出されるとすぐに、フェーズ 2 は終了になります。機械軸は一定の速度を維持し、レファレンス点復帰はフェーズ 3 により継続されます。

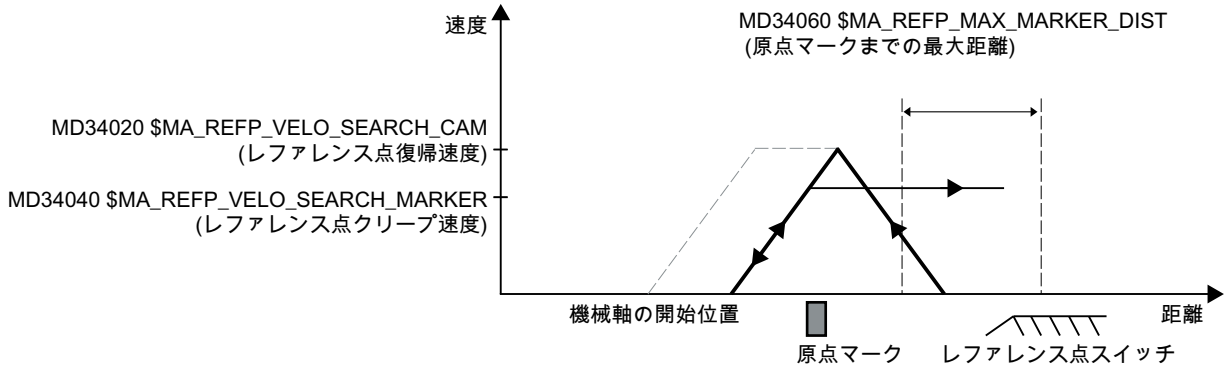


図 16-7　レファレンス点スイッチ信号の立ち上がりエッジによる同期

ソフトレファレンス点スイッチシフト

ソフトレファレンス点スイッチシフトを使用して、温度によって生じるレファレンス点スイッチの膨張を補正し、常に同じエンコーダ原点マークに同期されるようにします。

MD34092 $MA_ REFP_CAM_SHIFT (等間隔原点マーク付きインクリメンタル検出器のソフトレファレンス点スイッチシフト)

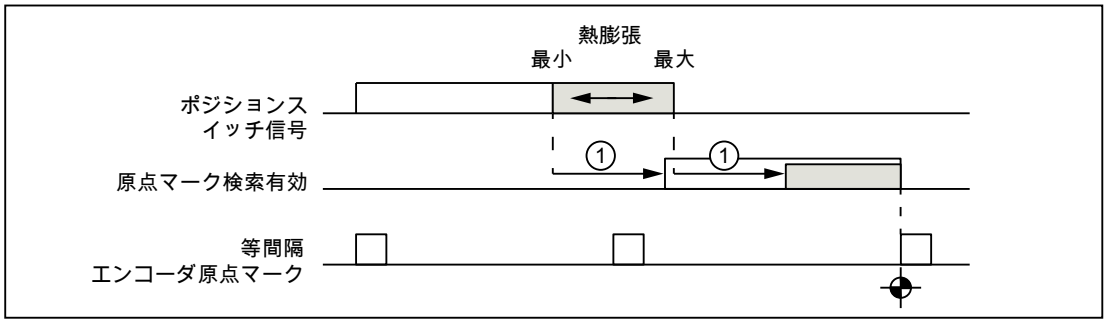
ソフトレファレンス点スイッチシフトでは、次のエンコーダ原点マークへの同期は、レファレンス点スイッチ信号のエッジの検出後すぐにではなく、パラメータ設定されたオフセット距離を移動した後でのみおこなわれます。

レファレンス点スイッチ信号のエッジの検出から補間クロック周期で移動する距離の特定により、有効なシフト距離は s_{shift} になります。

s_{shift_min} = MD34092 $MA_ REFP_CAM_SHIFT

s_{shift_max} = MD34092 $MA_ REFP_CAM_SHIFT +
MD34040 $MA_REFP_VELO_SEARCH_MARKER *補間クロック周期

ソフトレファレンス点スイッチシフトは、原点マーク検索の方向に作用します。



① レファレンス点スイッチシフト

図 16-8 ソフトレファレンス点スイッチシフト

必要条件

ソフトレファレンス点スイッチシフトは、レファレンス点スイッチのある機械軸にのみ有効です。

MD34000 $MA_REFP_CAM_IS_ACTIVE == 1

レファレンス点スイッチ調整

等間隔原点マーク付きエンコーダ

等間隔で原点マークを提供するエンコーダのレファレンス点スイッチは、レファレンス点復帰の間に正しい原点マークが常に検出されるように、正確な調整を確実におこなってください。

ダイナミック応答

以下の係数は、レファレンス点スイッチが機械的に到着してから、PLC ユーザープログラムから NC に伝送されるレファレンス点スイッチ信号が検出されるまで、ダイナミック応答に影響を及ぼします。

- レファレンス点スイッチスイッチの精度の切り替え
- レファレンス点スイッチスイッチの遅延(NC 接点)

- PLC 入力の遅延
- PLC 制御周期
- NC/PLC インタフェースを更新するための制御周期
- 補間クロック周期
- 位置制御周期

設定に関する注意事項

- レファレンス点スイッチ
  2 つの原点マークの間の真ん中にレファレンス点スイッチの信号エッジを調整することが、最も実用的な方法であることが分かっています。
- ソフトレファレンス点スイッチシフト

  ⚠ 警告

  衝突の可能性

  レファレンス点スイッチ調整が誤ったり、不正確である場合、間違った原点マークが使用されます。それで、コントローラにより間違った機械原点が計算されます。結果として、機械軸は間違った位置にアプローチします。ソフトウェアリミットスイッチとプロテクションゾーン、および作業領域リミットが間違った位置で動作するので、機械を保護することはできません。距離の違いは、2 つの原点マークの間で機械軸が移動する距離の+/-です。

  ソフトレファレンス点スイッチシフトのパラメータ設定に必要な情報は、読み取り専用マシンデータに記載されています。
  MD34093 $MA_REFP_CAM_MARKER_DIST (レファレンス点スイッチ/原点マークの間の距離)
  表示される値は、レファレンス点スイッチを起点に基準マークを検出するまでの距離に等しくなります。この値が小さすぎる場合、温度による影響またはレファレンス点スイッチ信号の動作時間の変動により、レファレンス点の決定が一定しない危険があります。

フェーズ 2:特徴

- 送り速度オーバライドは、**無効**です。
  送り速度オーバライド = 100%で移動が内部的に実行されます。
  0%の送り速度オーバライドが指定されると、中止されます。
- 送り停止(チャネル別おと軸別)が有効です。

- NC ストップと NC スタートは有効では**ありません**。
- フェーズ 2 で、機械軸がパラメータ設定された原点マーク(エンコーダ原点マーク)距離内で到達しない場合、機械軸は停止します。
  MD34060 $MA_REFP_MAX_ MARKER_DIST(原点マークまでの最大距離)

16.5.6 フェーズ 3: レファレンス点への移動

フェーズ 3:グラフ表示

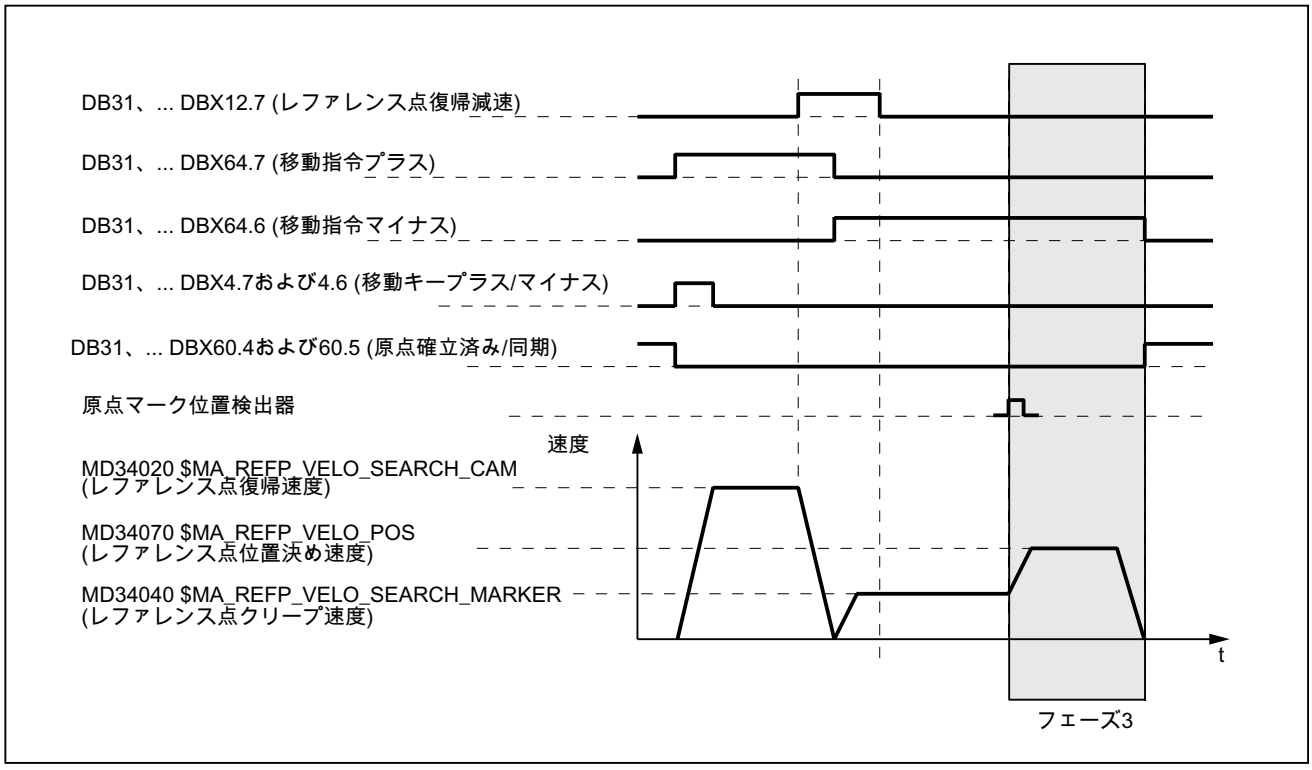


図 16-9 フェーズ 3:レファレンス点への移動

フェーズ 3:起動

フェーズ 2 の終了時に、機械軸がレファレンス点クリープ速度で移動します。したがって、フェーズ 2 がアラームなしで正常に完了した直後に、フェーズ 3 が中断なしで開始されます。

初期状態

エンコーダ原点マークが検出されています。

フェーズ 3:処理

機械軸が、フェーズ 2 で検出されたエンコーダ原点マークからレファレンス点まで、割り当てられたレファレンス点位置決め速度:
MD34070 $MA_REFP_VELO_POS(レファレンス点位置決め速度)
で移動します。

移動する距離 s_{ref} が、レファレンス点距離とレファレンス点オフセットの合計から計算されます。

MD34080 $MA_REFP_MOVE_DIST(レファレンス点距離)

MD34090 $MA_REFP_MOVE_DIST_CORR(レファレンス点オフセット)

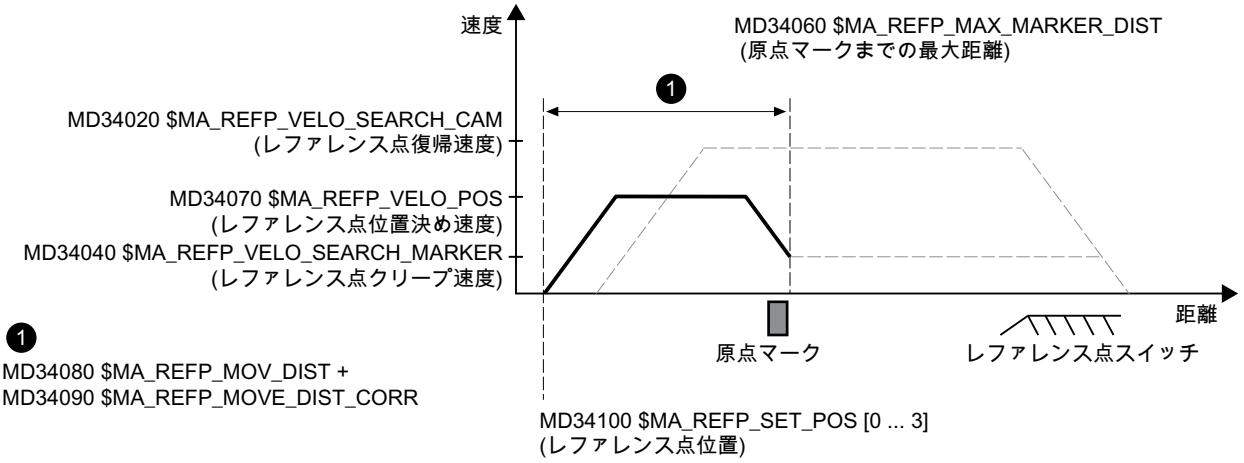


図 16-10　レファレンス点の位置

レファレンス点に達すると機械軸が停止し、フィードバック値システムが、NC/PLC インタフェースによって指定されたレファレンス点位置と同期します。

MD34100 $MA_ REFP_SET_POS[<n>] (レファレンス点位置)

レファレンス点位置の選択は、NC/PLC インタフェースを介しておこなわれます。

DB31、... DBX2.4 ... 7 (レファレンス点位置 1 ... 4)

フィードバック値システムは、レファレンス点スイッチがフェーズ 1 に達した時点 (DB31、... DBX12.7 == 1)で選択されたレファレンス点位置に同期されます。

ここで機械軸の原点が確立されます。インタフェース信号は、有効な検出器に対応して、PLC ユーザープログラムに対するフィードバックとして設定されます。

DB31、... DBX60.4/5(原点確立済み/原点同期済み 1/2) = 1

フェーズ 3 の特徴

- 送り速度オーバライドが有効です。
- 送り停止(チャネル別と軸別)が有効です。
- NC ストップと NC スタートが有効です。

フェーズ 3 の特記事項

以下の場合には、機械軸は原点マークの検出後に最初に停止し、それからレファレンス点に戻ります。

- レファレンス点の位置決め速度により、レファレンス点距離とレファレンス点オフセットの合計は、必要な減速距離より小さくなります。
  MD34080 + MD34090 < 「MD34070 による必要な減速距離」
- レファレンス点は、現在の移動方向と反対側の、レファレンス点スイッチの「後ろ側」にあります。

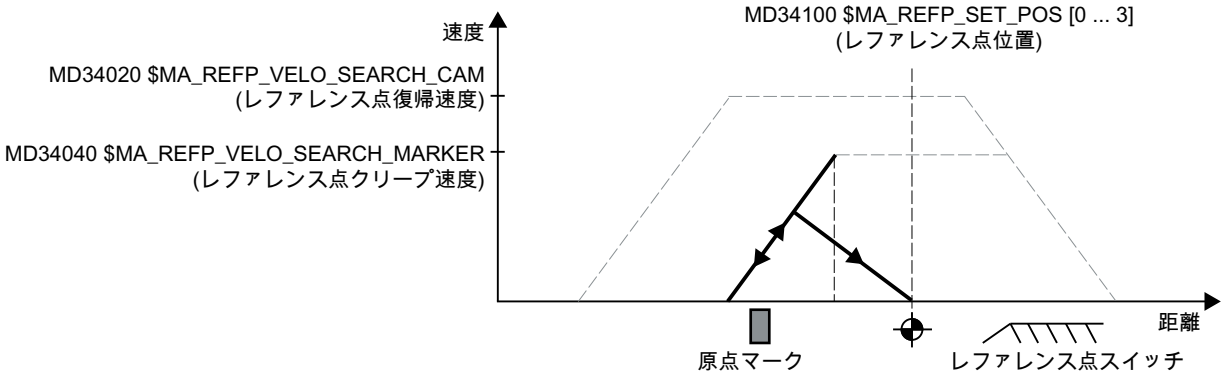


図 16-11　レファレンス点距離とレファレンス点オフセットの合計が減速距離未満

16.6　絶対番地化原点マークによるレファレンス点復帰動作

16.6.1　概要

絶対番地化原点マーク

絶対番地化原点マークのある検出器は、2 つの並列スケールトラックで構成されます。

- インクレメンタルグリッド
- 原点マークトラック

連続した任意の 2 つの原点マーク間の距離が定義されます。 そのため、連続した 2 つの原点マークを通過したとき、機械軸の絶対位置を特定できます。 たとえば、原点マーク間の距離が約 10 mm の場合、機械軸を原点確立をするには約 20 mm の移動距離のみが必要です。

原点確立は、正または負方向の任意の軸位置から実行できます(例外: 移動範囲の終点)。

16.6.2 基本パラメータ設定

リニア検出器

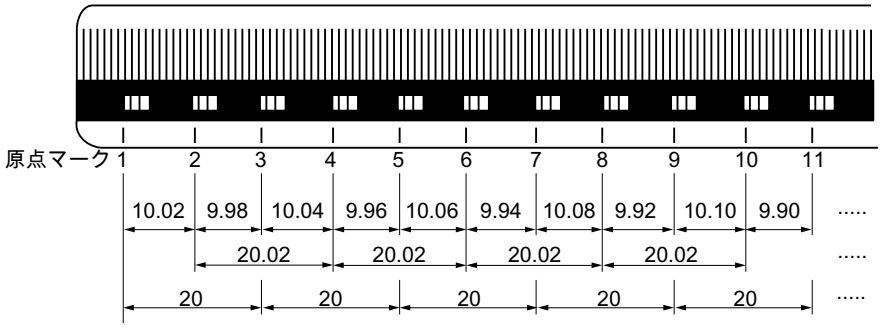


図 16-12 絶対番地化原点(マーク)、グリッド間隔付きのガラス測定スケール:20 mm

リニア検出器をパラメータ設定するには、次のデータを設定してください。

- 機械原点とリニア検出器の最初の原点マーク位置とのアブソリュートオフセット:
  MD34090 $MA_REFP_MOVE_DIST_CORR(レファレンス点/アブソリュートオフセット)
  下記も参照のこと:アブソリュートオフセットの特定
- 機械のシステム座標系に対する位置検出器の向き(同方向または逆方向) :
  MD34320 $MA_ENC_INVERS(機械システムに対して逆方向の位置検出器)

ロータリー検出器

ロータリー検出器にも、リニア検出器と同じことが当てはまります(上記参照)。

アブソリュートオフセットの特定

機械原点と機械軸の最初の原点マーク位置とのアブソリュートオフセットを決定するには、次の手順をお勧めします。

1. アブソリュートオフセットに値 0 を入力します。
   MD34090 $MA_REFP_MOVE_DIST_CORR = 0
2. レファレンス点復帰を実行します。
   注:レファレンス点復帰は、機械原点に対する機械軸の正確な相対位置を、たとえばレーザ干渉計などで、簡単に特定できる機械内の位置で実行してください。
3. ユーザーインタフェース、たとえば SINUMERIK Integrate から、機械座標系 MCS で表示された機械軸の実位置を読み取ります。
4. 機械原点を基準とする機械軸の実位置を計測します。
5. アブソリュートオフセットを計算して、MD34090 に入力します。
   アブソリュートオフセットは、検出器の向きに応じて、機械座標系に対して次のように計算されます。

検出器の向き	アブソリュートオフセット
同方向	(測定された位置) **+** (表示されている現在位置)
反転方向	(測定された位置) **-** (表示されている現在位置)

⚠ 警告

レファレンス点の誤差

アブソリュートオフセットが特定され、MD34090 に入力された後、その機械軸のレファレンス点復帰を再度実行してください。

レファレンス点復帰動作の方法

絶対番地化原点マークによるレファレンス点復帰動作は、次の 2 つの方法のどちらかで実行できます。

- 連続した **2 つの**原点マークの評価:
  MD34200 $MA_ENC_REFP_MODE = 3
  長所:
  – 短い移動距離
- 連続した **4 つの**原点マークの評価:
  MD34200 $MA_ENC_REFP_MODE = 8
  長所:
  – NC による妥当性チェックが可能
  – 結果としてレファレンス点復帰動作の信頼性が向上

16.6.3 タイミング図

タイミング図

絶対番地化原点マークによるレファレンス点復帰動作は、次の 2 つのフェーズに分割できます。

- フェーズ 1: 同期をともなう原点マークの通過
- フェーズ 2: 固定終点への移動

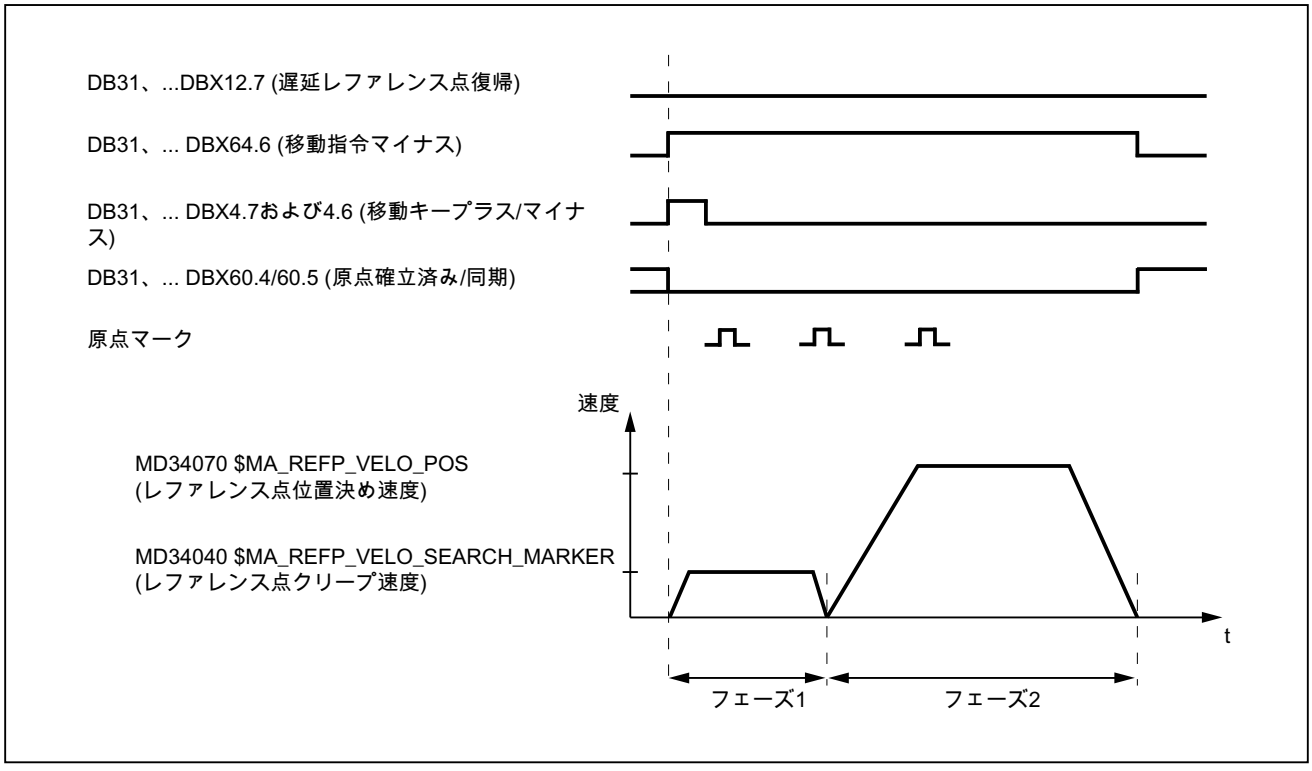


図 16-13 絶対番地化原点マーク

16.6.4 フェーズ 1: 同期をともなう原点マークの通過

フェーズ 1:開始

レファレンス点復帰を開始するには、「軸別原点確立 (ページ 1498)」と「チャネル別の原点確立 (ページ 1500)」の章を参照してください。

レファレンス点スイッチ

絶対番地化原点マークのある検出器では、実際のレファレンス点復帰動作にはレファレンス点スイッチが必要ありません。 ただし、機能上の理由により、機械軸の移動範囲終点より前のパートプログラム(G74)によるレファレンス点復帰とチャネル別のレファレンス点復帰には、レファレンス点スイッチが必要です。

フェーズ 1:処理

レファレンス点スイッチとの接触なしの処理

レファレンス点復帰プロセスが開始されると、機械軸がパラメータ設定で設定されたレファレンス点停止速度まで加速されます。

MD34040 $MA_REFP_VELO_SEARCH_MARKER (レファレンス点クリープ速度)

パラメータ設定された原点マーク数を過ぎると、機械軸が再度停止し、機械軸のフィードバック値系が、NC によって計算されたアブソリュート位置と同期します。

レファレンス点スイッチから開始した場合の処理

レファレンス点移動の開始時に機械軸がレファレンス点スイッチにある場合、機械軸はパラメータ設定されたレファレンス点クリープ速度まで、パラメータ設定されたレファレンス点復帰方向と逆の方向に加速します。

MD34040 $MA_REFP_VELO_SEARCH_MARKER (レファレンス点クリープ速度)

MD34010 $MA_CAM_DIR_IS_MINUS(マイナス方向のレファレンス点復帰)

これによって、機械軸がパラメータ設定された数の原点マークを通過する前に、移動範囲の終端に達しないことが保障されます。

パラメータ設定された原点マーク数を過ぎると、機械軸が再度停止し、機械軸のフィードバック値系が、NC によって計算されたアブソリュート位置と同期します。

レファレンス点復帰動作中にレファレンス点スイッチと接触する場合の処理

レファレンス点復帰プロセスが開始されると、機械軸がパラメータ設定で設定されたレファレンス点停止速度まで加速されます。

MD34040 $MA_REFP_VELO_SEARCH_MARKER (レファレンス点クリープ速度)

機械軸がパラメータ設定された数の原点マークを移動する前に、レファレンス点スイッチに接触します。 この場合、機械軸が反転し、逆方向で原点マーク検出が再開されます。

パラメータ設定された原点マーク数を過ぎると、機械軸が再度停止し、機械軸のフィードバック値系が、NC によって計算されたアブソリュート位置と同期します。

原点マーク距離の妥当性チェック

レファレンス点移動中に後続の 2 つの原点マークに対して、パラメータ設定された原点マーク距離の 2 倍より大きい距離が NC によって検出されると、異常が検地されます。

MD34300 $MA_ENC_REFP_MARKER_DIST(原点マーク距離)

この場合、機械軸がパラメータ設定されたレファレンス点クリープ速度(MD34040)の半分の速度で逆方向に移動し、原点マークの検出が再開されます。

不正な原点マーク距離が再度検出されると、機械軸が停止し、レファレンス点移動が中止されます(アラーム 20003「検出器での異常」)。

中止条件

パラメータ設定された距離内で、パラメータ設定された原点マーク数が検出されないと、機械軸が停止し、レファレンス点復帰が中止されます。

MD34060 $MA_REFP_MAX_ MARKER_DIST(原点マークまでの最大距離)

フェーズ 1 の機能

フェーズ 1 が正常に完了すると、機械軸のフィードバック値系が原点に同期します。

16.6.5 フェーズ 2:目標点への移動

フェーズ 2:開始

フェーズ 1 がアラームなしで完了すると、フェーズ 2 が自動的に開始します。

初期状態:

- 機械軸は、パラメータ設定された原点マーク数のうち、最後の原点マークの直後に位置決めされます。
- 機械軸のフィードバック値系が同期します。

フェーズ 2:処理

フェーズ 2 では、定義された目標位置(レファレンス点)への機械軸の移動によって、機械軸のレファレンス点復帰が完了します。この動作は、レファレンス点復帰を短縮するためにマスクできます。

MD34330 $MA_STOP_AT_ABS_MARKER = <値>

値	意味
0	目標位置への移動
1	目標位置への移動なし

目標位置への移動(通常の場合)

機械軸がパラメータ設定されたレファレンス点位置決め速度まで加速し、パラメータ設定された目標点(レファレンス点)に移動します。

MD34070 $MA_REFP_VELO_POS (レファレンス点位置決め速度)

MD34100 $MA_REFP_SET_POS(レファレンス点位置)

機械軸の原点が確立されます。これを識別するため、現在有効な検出器のインタフェース信号が NC によって下記のように設定されます。

DB31、... DBX60.4/60.5(原点確立済み/原点同期済み 1/2) = 1

目標位置への移動なし

ここで機械軸の原点が確立されます。これを識別するため、現在有効な検出器のインタフェース信号が NC によって下記のように設定されます。

DB31、... DBX60.4/60.5(原点確立済み/原点同期済み 1/2) = 1

フェーズ 2 の特性

フェーズ 2 では、機械軸に対してレファレンス点スイッチがパラメータ設定されているかどうかに応じて、異なる特性が表われます。

レファレンス点スイッチのない機械軸

MD34000 $MA_REFP_CAM_IS_ACTIVE (レファレンス点スイッチのある軸) = 0

特性:

- 送り速度オーバライドが有効です。
- 送り停止(チャネル別と軸用)が有効です。
- NC ストップと NC スタートが有効です。

レファレンス点スイッチのある機械軸

MD34000 $MA_REFP_CAM_IS_ACTIVE (レファレンス点スイッチのある軸) = 1

特性:

- 送り速度オーバライドは、**無効**です。
  送り速度オーバライド = 100%で移動が内部的に実行されます。
  0%の送り速度オーバライドが指定されると、中止されます。
- 送り停止(チャネル別と軸用)が有効です。
- NC ストップと NC スタートが**有効ではありません**。
- レファレンス点スイッチの終了後に、パラメータ設定された距離内でパラメータ設定された原点マーク数が検出されないと、機械軸が停止します。
  MD34060 $MA_REFP_MAX_ MARKER_DIST(原点マークまでの最大距離)

ロータリー検出器の特記事項

絶対番地化ロータリー検出器では、アブソリュート位置は 1 回転内でのみ一義的に決定できます。エンコーダの機械的な取り付け方法によっては、ハードウェアでのアブソリュート位置のオーバートラベルが必ずしも回転軸の移動範囲と一致しません。

モジュロ回転軸の特記事項

モジュロ回転軸では、パラメータ設定されたモジュロ範囲でレファレンス点位置が割り当てられます。

MD30330 $MA_MODULO_RANGE(モジュロ範囲のサイズ)

MD30340 $MA_MODULO_RANGE_START (モジュロ範囲の開始位置)

注記

レファレンス点位置は、軸機能「レファレンス点位置の決定、モジュロ範囲内の絶対番地化エンコーダ」の場合でも、パラメータを設定された(仮想)モジュロ範囲に割り当てられます。

MD30455 $MA_MISC_FUNCTION_MASK(軸機能)、ビット 1 = 1

16.7 フィードバック値の位置合わせによる原点確立

16.7.1 フィードバック値の原点確立検出器への位置合わせ

機能

フィードバック値の原点確立検出器への位置合わせが実行されると、機械軸の検出器が正常に原点確立された後に生成される実際の絶対位置が、機械軸の他のすべての検出器に自動的に伝送され、機械軸が原点確立済みとして指定されます。

DB31、... DBB60.4 / 60.5(原点確立済み/原点同期済み 1/2) = 1

長所

機械軸が、直接原点確立される検出器から、フィードバック値の位置合わせによって原点確立される検出器に切り替えられた場合、実際の位置の一致によりフィードバック値の急激な変動が防止されるため、連続したサーボ制御(サーボイネーブル有効)が確保されます。

注記

検出器別のエンコーダ精密情報の特定によって位置決め精度を改善するため、事前にフィードバック値の位置合わせによって原点確立された検出器の切り替え後に、再び直接原点確立することをお勧めします。

適用

原点確立検出器へのフィードバック値位置合わせの適用タイミングは、次のマシンデータを介して機械別に実行されます。

MD34102 $MA_REFP_SYNC_ENCS = 1

16.7.2 絶対番地化原点マークのある検出器へのフィードバック値の位置合わせ

機能

検出器別のエンコーダ精密情報の特定によって位置決め精度を改善するため、事前にフィードバック値の位置合わせによって原点確立された検出器の切り替え後に、再び直接原点確立することをお勧めします。

絶対番地化原点マークのあるエンコーダを不使用検出器として使用する場合、次の条件が満たされれば原点確立を回避できます。

1. 有効な検出器:絶対値エンコーダを備えた間接検出器(モータ検出器)など
2. 不使用検出器絶対番地化原点マークのある直接検出器
3. 検出器の切り替え以前に、間接検出器のある機械軸が原点確立に必要な原点マーク数を通過する移動動作。これによって、パッシブ直接検出器が自動的に原点確立されます。

パラメータ設定

個々の検出器を原点確立するために必要な個別のマシンデータに加え、次のマシンデータを設定してください。

- フィードバック値位置合わせの有効化:
  MD34102 $MA_REFP_SYNC_ENCS = 1
- 絶対番地化原点マークのある直接検出器:
  - MD34200 $MA_ENC_REFP_MODE[*検出器*] = 3
    絶対番地化原点マーク
  - MD30242 $MA_ENC_IS_INDEPENDENT[*検出器*] = 2
    フィードバック値の位置合わせ時に、不使用直接検出器が有効間接検出器の実際の位置に対して割り出されますが、原点確立済みとして特定されません。パラメータ設定された数の原点マークを通過した後、不使用直接検出器が自動的に原点確立されます。原点確立はすべての運転モードで実行されます。

手順

1. 初期状態両方の検出器が原点確立されていません。
   DB31、... DBX60.4 = 0 (原点確立済み/原点同期済み 1)
   DB31、... DBX60.5 = 0 (原点確立済み/原点同期済み 2)
2. 検出器のタイプに応じて間接検出器を原点確立します。
   DB31、... DBX60.4 = 1 (原点確立済み/原点同期済み 1)
   DB31、... DBX60.5 = 0 (原点確立済み/原点同期済み 2)
3. 機械軸に、パラメータ設定された数の原点マークを通過させます。
   これによって直接検出器が自動的に原点確立されます。
   DB31、... DBX60.4 = 1 (原点確立済み/原点同期済み 1)
   DB31、... DBX60.5 = 1 (原点確立済み/原点同期済み 2)

16.8 フォローアップモードでの原点確立

機能

インクリメンタル検出器と、絶対番地化原点マークのある検出器は、機械軸がフォローアップモードの場合にも原点確立できます。このための必要条件は、使用される検出器に対応して、レファレンス点復帰を適切にパラメータ設定していることです(「インクリメンタルエンコーダによる原点確立 (ページ 1503)」と「絶対番地化原点マークによるレファレンス点復帰動作 (ページ 1518)」の章を参照してください)。

フォローアップモードでの原点確立の場合は、機械軸が、NC ではなく外部の移動動作によって、エンコーダ原点マークまで、またはパラメータ設定された絶対番地化原点マークの数まで、移動します。エンコーダ原点マーク、またはパラメータ設定された絶対番地化原点マークの数が検出されると、検出器の原点が確立されます。

注記

原点確立結果の再現性

NC 主導のレファレンス点復帰では、割り当てられた移動速度を原点確立操作時に順守することによって、原点確立結果の再現性が確保されます。フォローアップモードでの原点確立時には、原点確立結果の再現性を得る責任は工作機械メーカ/ユーザーにあります。

一義的な原点マーク

インクリメンタル検出器の原点確立は、機械軸の移動範囲全体に対するエンコーダ原点マークの明示的位置に基づきます。

フォローアップモードでの原点確立時にはレファレンス点スイッチ信号が NC によって使用されないため、フォローアップモードでの原点確立におけるレファレンス点の一義的な識別は、次の条件でのみおこなわれます。

- 機械軸の移動範囲内に 1 つのみのエンコーダ原点マーク
- 絶対番地化原点マーク付き直線検出器
- モジュロ回転軸(1 回転内のアブソリュート位置)

複数の原点マーク信号が発生した場合の原点マーク選択

エンコーダと負荷の間の減速ギアなど、機械固有の要因により、機械軸の移動範囲で複数のエンコーダ原点マークが検出された場合、レファレンス点を明確に特定するために、近接スイッチを機械に組み込み、NCU インタフェースのディジタル入力部を介して接続してください。

注記

近接スイッチの信号:原点マーク選択

近接スイッチ信号の処理は、ドライブでのみ行われます。接続とパラメータ設定については、「ハードウェア信号 (ページ 1503)」の章を参照してください。

設定

「フォローアップモードでの原点確立」機能は、次のマシンデータによって有効になります。

MD34104 $MA_REFP_PERMITTED_IN_FOLLOWUP = TRUE

原点確立操作の開始

レファレンス点復帰の開始時に機械軸がフォローアップモードの場合(DB31、... DBX61.3 == TRUE)、検出器はフォローアップモードで原点確立されます。

レファレンス点復帰の開始時に機械軸がフォローアップモードで動作していない場合、NC 制御下のレファレンス点復帰での「正常」が実行されます。

フォローアップモードでの原点確立は、次のモードで開始できます。

- JOG-REF:移動キー
- AUTOMATIC:パートプログラムの命令 G74

原点確立操作の手順(JOG-REF モード)

1. 機械軸のフォローアップモードを有効にします。
   DB31、... DBX1.4 (フォローアップモード) = 1
   DB31、... DBX2.1 (コントローライネーブル) = 0
2. フォローアップモードの起動を待ちます。
   DB31、... DBX61.3 (フォローアップ有効) == 1
3. JOG モードへの切り替え、REF 機械機能

4. エンコーダ原点マーク、またはパラメータ設定された絶対番地化原点マークの数を通過する機械軸の外部動作機械軸が移動した直後に原点確立動作が NC 内部で開始されます。以下の NC/PLC インターフェース信号がフィードバックとしてリセットされます。
   DB31、... DBX61.4 (軸/主軸停止) == 0
5. エンコーダ原点マーク、または絶対番地化原点マークの数が正常に検出された後、検出器が原点確立されます。以下の NC/PLC インターフェース信号がフィードバックとして設定されます。
   DB31、... DBX60.4/60.5 (原点確立済み/原点同期済み 1/2) == 1

原点確立操作の中止

動作中の原点確立操作を次の方法で中止できます。

- フォローアップモードの解除
- NC リセット

検出器が既に原点確立されている場合の動作

既に原点確立されている検出器は、パートプログラム命令 G74 を使用して、AUTOMATIC モードのみで再度原点確立できます。

原点確立操作の手順(AUTOMATIC モード)

1. AUTOMATIC モードに切り替えます。
2. パートプログラムを起動します。
3. 機械軸のフォローアップモードを有効にします。
   DB31、... DBX1.4 (フォローアップモード) = 1
   DB31、... DBX2.1 (コントローライネーブル) = 0
4. フォローアップモードの起動を待ちます。
   DB31、... DBX61.3 (フォローアップ有効) == 1
5. パートプログラム命令 G74 が処理された直後に、原点確立動作が NC 内部で開始されます。
6. エンコーダ原点マーク、またはパラメータ設定された絶対番地化原点マークの数を通過する機械軸の外部動作
7. エンコーダ原点マーク、または絶対番地化原点マークの数が正常に検出された後、検出器が原点確立されます。以下の NC/PLC インターフェース信号がフィードバックとして設定されます。
   DB31、... DBX60.4/.60.5 (原点確立済み/原点同期済み 1/2) == 1
8. 原点確立操作が正常に完了した後、ブロック切り替えが実行されます。

原点確立操作の中止

動作中の原点確立操作を次の方法で中止できます。

- フォローアップモードの解除
- NC リセット

検出器が既に原点確立されている場合の動作

すでに原点確立した検出器を再原点確立できます。

16.9 絶対値エンコーダの原点確立

16.9.1 位置合わせに関する情報

絶対値エンコーダを備えた機械軸

絶対値エンコーダを備えた機械軸の長所は、一度位置合わせ手順を実行した後、インクリメンタル検出器による必要なレファレンス点復帰(制御の構築、機械軸の「パーキング」の解除など)をスキップでき、機械軸のフィードバック値系を、特定されたアブソリュート位置へすぐに原点同期できることです。

位置合わせ

絶対値エンコーダの位置合わせには、エンコーダのフィードバック値を機械原点と 1 回照合した後、フィードバック値を有効に設定する操作が含まれます。

絶対値エンコーダの現在の位置合わせ状態は、そのエンコーダが接続されている機械軸の次の軸マシンデータに表示されます。

MD34210 $MA_ENC_REFP_STATE(絶対値エンコーダの状態)

規格値	意味
0	エンコーダ位置合わせ未完
1	エンコーダ位置合わせ有効
2	エンコーダ位置合わせ済み

位置合わせ方法

次の位置合わせ方法がサポートされます。

- レファレンス点オフセットの入力による位置合わせ
- レファレンス点位置の入力による位置合わせ

- プローブによる自動位置合わせ
- 近接スイッチによる位置合わせ

再位置合わせ

次のイベントの後、絶対値エンコーダの再位置合わせが必要です。

- 負荷側と絶対値エンコーダ間のギヤ切替
- 絶対値エンコーダの取り外し/取り付け
- 絶対値エンコーダ付きのモータの取り外し/取り付け
- スタティック NC メモリでのデータ損失

- バッテリ低下
- フィードバック値の設定(PRESETON)

⚠ 警告

データバックアップ

機械 A のマシンデータのバックアップ時に、機械軸のエンコーダ状態(MD34210)もバックアップされます。

一括セットアップのときや、保守をおこなった後などで、このデータレコードを同じタイプの機械 B に読み込むと、原点確立された機械軸が NC によって自動的に位置合わせ済み/原点確立済みとみなされます。このような場合の再位置合わせは、工作機械メーカ/ユーザーの責任において必ず実行してください。

次のマシンデータに関する説明も参照してください。

MD30250 $MA_ACT_POS_ABS (電源遮断時の絶対値エンコーダ位置)

注記

コントローラでは、次のイベントの発生時にのみ、必要な絶対値エンコーダの再位置合わせを検出できます。

- ギヤ比の変更によるギヤ切替
- 原点マーク監視のアドレス指定
- 絶対値エンコーダの変更後の新規エンコーダシリアル番号

コントローラは絶対値エンコーダの状態を「0」に設定します。

MD34210 $MA_ENC_REFP_STATE = 0 (エンコーダの位置合わせなし)

次のアラームが表示されます。
アラーム 25022「軸<軸名称>エンコーダ<番号>警告 0」
原点マーク監視が応答した場合、次のアラームも表示されます。
アラーム 25020「軸<軸名称>有効なエンコーダの原点マーク監視」

その他の場合は(たとえば、PRESETON など)、ユーザーが**手動で**状態を「0」に設定して絶対値エンコーダの不一致を示し、再位置あわせを実行してください。

16.9.2 レファレンス点オフセットの入力による位置合わせ

動作

レファレンス点オフセットの入力による位置合わせ時には、操作画面に表示される位置と機械での実位置との差が特定され、レファレンス点オフセットとして NC に通知されます。

手順

1. 次のような方法による、機械原点に対する機械軸の位置の特定:
   – 位置測定(レーザ干渉計など)
   – 既知の位置(突き当て点など)への機械軸の移動
2. 操作画面に表示されている機械軸のフィードバック位置の読み取り
3. レファレンス点オフセット(項 1 および 2 で特定した実位置間の差)の計算と、マシンデータへの入力:
   MD34090 $MA_REFP_MOVE_DIST_CORR(レファレンス点オフセット)
4. 絶対値エンコーダを位置合わせ済みとして設定:
   MD34210 $MA_ENC_REFP_STATE = 2

   注記

   エンコーダの位置合わせは、次回エンコーダが起動するまで(コントローラの電源投入時など)有効になりません。

5. POWER ON リセットの開始
6. 操作画面に表示される機械軸の位置の照合

   注記

   バックラッシ補正

   絶対値エンコーダを備えた検出器に対してバックラッシ補正がパラメータ設定されている場合は、次の規則を順守してください。

   位置合わせの機械位置へ機械軸が移動している間は、バックラッシが許可されません。

   レファレンス点オフセットの恒久的な有効化

   入力したレファレンス点オフセット(MD34090)は、最初の POWER ON - リセット後にのみ恒久的に有効になります。 絶対値エンコーダの位置合わせ後に暫定的な POWER ON - リセットなしで機械軸が移動した場合、マシンデータに入力されたレファレンス点オフセットを、たとえば内部オーバランオフセットの一部として上書きできます。

   実際の位置のチェック

   絶対値エンコーダの位置合わせ後、次回コントローラに電源を投入したときに(POWER ON)、機械軸の実位置を検証することをお勧めします。

16.9.3 レファレンス点位置の入力による位置合わせ

機能

レファレンス点位置の入力による位置合わせ時には、機械原点に対する機械軸の絶対位置を次のような方法で特定できます。

- 位置測定(レーザ干渉計など)
- 既知の位置(突き当て点など)への機械軸の移動

この特定された位置データがレファレンス点位置として NC に通知されます。その後、NC によってエンコーダの絶対値とレファレンス点位置の差からレファレンス点オフセットが計算されます。

処理

1. 指令モードが「レファレンス点位置を想定」に設定されているかどうかをチェックします。
   MD34200 $MA_ENC_REFP_MODE == 0
   そうでない場合は、値 0 を機械に入力し、電源投入リセットをトリガしてください。
2. (レーザ干渉計などで)測定する位置または既知の位置(突き当て点など)まで機械軸を JOG モードで移動します。

   注記

   機械軸は、移動キーで原点確立に対して有効になっている方向のみに移動できます。

   MD34010 $MA_REFP_CAM_DIR_IS_MINUS (マイナス方向のレファレンス点復帰)

   動力伝達でのバックラッシによる不正な位置を回避するため、既知の位置には低速で移動してください。
3. 機械原点に対する機械軸の位置を、レファレンス点位置としてマシンデータに入力します。
   MD34100 $MA_REFP_SET_POS = *位置*
4. エンコーダの位置合わせを有効にします。
   MD34210 $MA_ENC_REFP_STATE = 1
5. 電源投入リセットをトリガして、入力されたマシンデータ値を受け入れます。
6. JOG-REF モードに切り替えます。

7. ステップ 2 で原点確立に使用した移動キーを操作します。
   移動キーを作動させたときに、機械軸は移動しません。
   入力されたレファレンス点位置と、絶対値エンコーダによって提供されたレファレンス点位置から、レファレンス点オフセットが NC によって計算されます。その結果がマシンデータに入力されます。
   MD34090 $MA_REFP_MOVE_DIST_CORR (レファレンス点オフセット)
   絶対値エンコーダの状態が「エンコーダ位置合わせ済み」に設定されます。
   MD34210 $MA_ENC_REFP_STATE = 2
   機械軸のフィードバック値系が同期します。
   ここで機械軸の原点が確立されます。NC によって、現在有効な検出器に基づく当該のインターフェース信号が通知のために設定されます。
   DB31、... DBB60.4 / 60.5 (原点確立済み/原点同期済み 1 / 2) = 1
8. POWER ON リセットの実施

注記

レファレンス点オフセットの恒久的な有効化

入力されたレファレンス点オフセット(MD34090)は、電源投入リセット後にのみ、恒久的に有効になります。

絶対値エンコーダの位置合わせ後に暫定的な電源投入リセットなしで機械軸が移動した場合は、マシンデータに入力されたレファレンス点オフセットを、内部オーバラン補正などで上書きできます。

実位置のチェック

絶対値エンコーダの位置合わせ後、次回コントローラに電源を投入したときに(POWER ON)、機械軸の実位置を検証することをお勧めします。

16.9.4 プローブによる自動校正

動作

プローブによる自動校正では、パートプログラムから、機械軸によって機械内の既知の位置に移動します。 位置データはレファレンス点位置として NC に格納されます。 プローブのスイッチが入ったときにその位置に達し、その後エンコーダの値とレファレンス点位置の差からレファレンス点オフセットが NC によって計算されます。

注記

自動位置合わせ用のパートプログラム

プローブを使用する自動位置合わせ用のパートプログラムは、機械に固有の要求事項に合わせて工作機械メーカ/ユーザーが作成してください。

衝突の防止

原点確立中の機械軸に対してフィードバック値関連の監視が有効になっていないため、機械オペレータは機械軸の移動中に機械で衝突が発生しないように特に注意してください。

パートプログラム

プローブによる絶対値エンコーダの自動位置合わせ用のパートプログラムでは、各軸に対して次の各項目がこの順序で実行される必要があります。

1. プローブの応答から検出される機械軸の位置合わせ位置に移動します。
   この位置には同じ方向から複数回アプローチする必要がありますが、取得される測定値を可能な限り正確にするために、移動ごとに速度を徐々に低減してください。 計測値はシステム変数$AA_IM に格納されます。
2. レファレンス点オフセットの計算と書き込み:

```
MD34090 $MA_REFP_MOVE_DIST_CORR = MD34100 $MA_REFP_SET_POS – $AA_IM
```

3. 絶対値エンコーダの状態を「エンコーダ位置合わせ済み」に設定します。
   MD34210 $MA_ENC_REFP_STATE = 2

手順

プローブによる自動位置合わせを行うには、次の手順を実行します。

1. 原点確立されない機械軸に対しても、パートプログラムの起動を有効にします。
   MD20700 $MC_REFP_NC_START_LOCK = 0
2. プローブのスイッチが入るときの機械原点に対する機械軸の位置を、すべての関連する機械軸のレファレンス点位置として入力します。
   MD34100 $MA_REFP_SET_POS = レファレンス点位置

3. 入力したマシンデータ値を受け入れるための NCK-リセットを起動します。
4. パートプログラムを起動します。
5. パートプログラムが完了した後、原点確立されない機械軸に対してパートプログラムスタートが禁止されます。
   MD20700 $MC_REFP_NC_START_LOCK = 1
6. パートプログラムによって書き込まれたレファレンス点オフセットが恒久的に有効になるように、POWER ON - リセットを開始します。
   MD34090 $MA_REFP_MOVE_DIST_CORR(レファレンス点オフセット)

注記

レファレンス点オフセットの恒久的な有効化

入力されたレファレンス点オフセット(MD34090)は、POWER ON - リセット後にのみ恒久的に有効になります。

絶対値エンコーダの位置合わせ後に暫定的な POWER ON - リセットなしで機械軸が移動した場合、マシンデータに入力されたレファレンス点オフセットを、たとえば内部オーバランオフセットの一部として上書きできます。

実位置のチェック

絶対値エンコーダの位置合わせ後、次回コントローラに電源を投入したときに(POWER ON)、機械軸の実位置を検証することをお勧めします。

16.9.5 BERO による位置合わせ

機能

近接スイッチを使用した位置合わせでは、定義された機械位置へのレファレンス点復帰が、インクリメンタル検出器の場合と同様におこなわれます。この場合、絶対値エンコーダにないエンコーダ原点マークが近接スイッチによって置換されます。レファレンス点復帰が正常に完了した後、エンコーダのアブソリュート値とパラメータ設定されたレファレンス点位置の差からレファレンス点オフセットが NC によって自動的に計算されます。

パラメータ設定

NC:原点復帰動作モード

原点復帰モードは、「インクリメンタル、回転、またはリニア検出器の原点復帰:エンコーダトラックでゼロパルス」に設定してください。

MD34200 $MA_ENC_REFP_MODE[<軸>] = 1

NC:レファレンス点位置

レファレンス点位置は以下によってパラメータ設定されます。

MD34100 $MA_REFP_SET_POS[<軸>] = <レファレンス点位置>

ドライブ:等価原点マーク

近接スイッチの接続先となる NCU インタフェースのディジタル入力は、パラメータ p0494 または p0495 で設定してください。

実行

レファレンス点復帰は、JOG-REF モードで手動で開始するか、パートプログラム(G74)から AUTOMATIC または MDI モードで開始できます。

レファレンス点復帰が正常に終了した後で、絶対値エンコーダが位置合わせされ、機械軸のフィードバック値系が原点同期済されます。

PLC ユーザープログラムへのフィードバックとして、NC は有効な検出器に対応して機械軸の NC/PLC インタフェース信号を設定します。

DB31、... DBB60.4 / 60.5 (原点確立済み/原点同期済み 1 / 2) = 1

注記

絶対値エンコーダの位置合わせ後に近接スイッチが取り外された場合は、レファレンス点復帰モードを「絶対値エンコーダによる原点確立」に再度パラメータ設定してください。

MD34200 $MA_ENC_REFP_MODE[<軸>] = 0

16.9.6 絶対値エンコーダによるレファレンス点復帰

パラメータの割り付け

移動動作許可

位置合わせ済みの絶対値エンコーダを有効な検出器として備えた機械軸で、レファレンス点移動が有効にされた場合(JOG-REF モードで手動、またはパートプログラム命令 G74 から自動)、機械軸はパラメータ設定された許可移動動作に応じて移動します。

MD34330 $MA_REFP_STOP_AT_ABS_MARKER = <value>

値	意味
0	移動が **有効**です。 レファレンス点復帰が開始されると、機械軸がレファレンス点位置まで移動します。レファレンス点位置に到達すると、レファレンス点復帰が完了します。
1	移動が有効 **ではありません**。 レファレンス点移動の起動後、機械軸が移動せず、レファレンス点移動が即時完了します。

16.9.7 等価原点マークによるロータリー絶対値エンコーダでのレファレンス点復帰

機能

原点パルス(セクション「インクリメンタルエンコーダによる原点確立 (ページ 1503)」参照)経由でのレファレンス点復帰が、絶対値エンコーダと共に使用することができるように、絶対値エンコーダのハードウェアには提供されない原点パルスがシミュレーションされます。このため、コントローラは、常に同じ回転位置でエンコーダ 1 回転につき等価ゼロマークの信号を生成します。

インクリメンタルエンコーダによる原点確立と比較した場合の違い

置換原点マークのある絶対値エンコーダを、インクリメンタルエンコーダと完全に同等なものとみなさないでください。絶対値エンコーダのすべての特性が保持されます。インクリメンタルエンコーダと絶対値エンコーダの異なる特性を次の表にリストします。

表 16-1 インクリメンタルエンコーダと絶対値エンコーダの特性

特性	インクリメンタルエンコーダ	絶対値エンコーダ
エンコーダタイプ	MD30240 $MA_ENC_TYPE =	
	1	4
内部エンコーダ位置	MD30250 $MA_ACT_POS_ABS =	
	MD34210 ≥ 1 の場合にのみ値を更新	MD30270 = 0 の場合にのみ値を更新
移動範囲の拡張	MD30270 $MA_ENC_ABS_BUFFERING =	
	動作なし	= 0(初期設定):有効

特性	インクリメンタルエンコーダ	絶対値エンコーダ
レファレンス点オフセット	MD34090 $MA_REFP_MOVE_DIST_CORR =	
	値の入力が可能	コントローラを介してのみ値を更新
サポートされるレファレンス点復帰タイプ	MD34200 $MA_ENC_REFP_MODE =	
	1, 3, 4, 8	0, 1
位置合わせ状態	MD34210 $MA_ENC_REFP_STATE = 0, 1, 2	
	動作中のシャットダウン時の自動エンコーダ不一致	位置の不連続変化のあるパラメータ設定の変更時、またはシリアル番号変更時の自動エンコーダ不一致
アブソリュート位置のモジュロ範囲	MD34220 $MA_ENC_ABS_TURNS_MODULO =	
	0	1 - 4096
エンコーダシリアル番号	MD34230 $MA_ENC_SERIAL_NUMBER =	
	0	エンコーダを変更するたびに、PLC からこの値を更新してください。そうでないと、位置合わせが失われアラームが発生します。
一括セットアップファイルの伝送	制限事項なし。	エンコーダ特性(MD30250、MD30270、MD34090、MD34210、MD34220、MD34230)により制限事項あり。
適用タイミング	0 秒	数秒
原点マーク	エンコーダ 1 回転につき 1 つ	なし
ゼロマーク監視	ハードウェア	ソフトウェア
現在位置が記憶されない場合の POWER ON 後の位置	0.0	MD34220 内での最終位置
	MD34210 = 0	MD30270 = 1
現在位置が記憶される場合の POWER ON 後の位置	電源しゃ断前の最終停止位置	POWER OFF 時の小さな移動を含む最終位置
	MD34210 = 1	MD30270 = 0
POWER ON 後の原点確立	位置合わせ状態に依存	

必要条件

この機能はロータリー絶対値エンコーダのみで使用できます。

- MD31000 $MA_ENC_IS_LINEAR == 0
- MD30240 $MA_ENC_TYPE == 4

パラメータ設定

等価原点マークによるレファレンス点復帰:

MD34200 $MA_ENC_REFP_MODE = 1

必要条件

- レファレンス点オフセット(MD34090 $MA_REFP_MOVE_DIST_CORR)が設定されない可能性があります。
  この MD は、絶対値エンコーダに関連して、機械原点と絶対値エンコーダ原点の間のオフセットが記述されるため、意味が異なります。
- 負荷側の原点マーク検出速度(MD34040 $MA_REFP_VELO_SEARCH_MARKER)が、エンコーダのアブソリュートトレースの限界周波数
  (MD36302 $MA_ENC_FREQ_LIMIT_LOW)を超えないようにしてください。
  速度が高すぎると、アブソリュート値情報の読み取りができなくなるため、再現性のある原点マークが生成されません。
- 原点マークが、設定されたパス(MD34060 $MA_REFP_MAX_MARKER_DIST)内に見つからない場合は、アラームが出力されます。
- 絶対値エンコーダで、前回位置のほかに原点確立済み状態も POWER OFF 後まで保持される場合は、次の MD を設定してください。
  MD34210 $MA_ENC_REFP_STATE = 2

データバックアップと標準セットアップ

絶対値エンコーダの一部の特性によって、コミッショニングアーカイブの他の機械への伝送が制限されます。コミッショニングアーカイブをコントローラにロードした後で、次のマシンデータをチェックして修正してください。

- MD30250 $MA_ACT_POS_ABS(内部エンコーダ位置)
- MD30270 $MA_ENC_ABS_BUFFERING(移動範囲拡張)
- MD34090 $MA_REFP_MOVE_DIST_CORR(アブソリュートオフセット)
- MD34210 $MA_ENC_REFP_STATE(位置合わせ状態)

- MD34220 $MA_ENC_ABS_TURNS_MODULO (モジュロ範囲)
- MD34230 $MA_ENC_SERIAL_NUMBER(エンコーダシリアル番号)

16.9.8 検出器の起動

機械軸の検出器は、次の場合に有効になります。

- 制御装置の POWER ON(POWER ON)
- 内部信号(「パーキング」の解除)を介した検出器の適用
  DB31、... DBB1.5 / 1.6(位置検出器 1/2)
  DB31、... DBB2.1(サーボイネーブル)
- 割り当てられたエンコーダ限界周波数への違反(主軸):
  MD36300 $MA_ENC_FREQ_LIMIT

検出器が有効にされると、NC によって機械軸のフィードバック値系が現在の絶対値と同期します。同期中は、軸では移動が無効になりますが、主軸ではなりません。

エンコーダ限界周波数のパラメータ設定(主軸)

Heidenhain 製の EQN 1325 絶対値エンコーダは、インクリメンタルトラックと絶対トラックを備えています。

インクリメンタルトラックのエンコーダ限界周波数を超える速度で主軸が駆動される場合は、十分低い絶対トラックの周波数をエンコーダ限界周波数としてパラメータ設定してください。

MD36300 $MA_ENC_FREQ_LIMIT

そうでないと、検出器が有効になったとき、パラメータ設定されたエンコーダ限界周波数に達しないため、不正な絶対位置が読み取られます。これによって、機械軸のフィードバック値系に位置オフセットが発生します。

エンコーダ限界周波数の特定

パラメータ設定するエンコーダ限界周波数は、次の 2 つの制限速度のうち低い方から導出されます。

- エンコーダ
  制限速度またはエンコーダ限界周波数は、エンコーダのデータシートに記載されています(制限速度 = 2000 [rpm]など)。
- NC
  NC 内部の評価プロセスの理由により、NC で異常なく絶対値を計算可能な最大制限速度は、1 補間クロック周期あたりエンコーダ 4 回転です。
  たとえば、補間クロック周期が 12 ms の場合、制限速度 = 4 / 12 ms = 20,000 rpm となります。
  制限速度に対応する限界周波数は、次のマシンデータとして計算されます。

$$MD36300 = \frac{4 * MD31020}{MD10050 * MD10070}$$

MD31020 $MA_ENC_RESOL (1 回転あたりのエンコーダパルス)
MD10050 $MN_SYSCLOCK_CYCLE_TIME (基本システムクロックサイクル)
MD10070 $MN_IPO_SYSCLOCK_TIME_RATIO (補間周期の係数)

注記

主軸に関連する位置制御切り替え速度は、主軸の絶対値エンコーダのエンコーダ限界周波数に従って設定されます。
MD35300 $MA_SPIND_POSCTRL_VELO(位置制御切り替え速度)
MD36300 $MA_ENC_FREQ_LIMIT(エンコーダ限界周波数)

16.9.9 サポートされない原点確立方法

次の原点確立方法は、絶対値エンコーダとともに使用される場合はサポートされません。

- エンコーダ原点マークによる原点確立/位置合わせ
- 絶対番地化原点マーク
- 2 つのエッジ評価をともなう近接スイッチ

16.10 機械原点の自動復元

機械軸の移動時に定義済みの機械原点確立がない場合、座標変換や工具フレームなどの位置に依存する機能を実行できません。さまざまな機械条件において、コントローラの電源投入後や「機械軸のパーキング (ページ 145)」の終了後などに、エンコーダ起動によってすぐにこれらの機能を使用して軸を移動できるようにしてください。ただし、機械軸を原点確立のために再度移動することはできません。

絶対値エンコーダ

調整済み絶対値エンコーダ付きの検出器の場合、機械原点はエンコーダ値を読み取るときに何もしなくてもすぐに復元されます。

インクリメンタルエンコーダ

インクレメンタル検出器では、「自動原点確立」または「実位置の復元」により、軸を移動しなくても機械原点を復元できます。

必要条件

⚠ 警告

実際の機械軸位置のオフセットを原因とする、位置検出器の誤った同期

機械軸の検出器をオフにしている間は、軸を機械的に**移動しない**でください。移動すると、機械軸の最後に記憶された実位置と実際の実位置の間にオフセットが生じます。このオフセットのために検出器の同期制御が不正確となり、人身事故や機械の損傷につながります。

工作機械メーカは、実位置が変化しないよう、機械に保持ブレーキを搭載するなどの対策をおこない、それをユーザーに認知させてください。これに対する責任は工作機械メーカ/ユーザーにあります。

停止状態にあるときに軸の移動を機械的に防止できない場合は、絶対値エンコーダを使用するか、電源投入後にレファレンス点復帰によって軸を再原点確立するかのいずれかをおこなってください。

注記

SMExx センサモジュール

コントローラの再起動後の自動原点確立または最後に記憶された位置への実位置の復元は、SM**E**xx (外部に取り付け)センサモジュールとの組み合わせでのみ可能です。SM**C**xx (制御盤)または SM**I**xx (内蔵)センサモジュールを使用する場合、コントローラの再起動(電源投入)後に実位置を復元**できません**。機械軸の検出器を再度原点確立してください。

16.10.1 自動原点確立

機能

コントローラの電源投入後、自動原点確立の間に、機械軸の有効な検出器の実位置が最後に記憶された位置に設定され、エンコーダ状態として「原点確立」されます。これにより、コントローラの起動後に直接 AUTOMATIC と MDI モードでプログラムを起動することができます。

必要条件

- コントローラを電源投入したときに有効な検出器がすでに原点確立されているように、電源オフする前に一度原点確立しておいてください。
- コントローラを電源オフした時点で、機械軸を「精密イグザクトストップ」(DB31、... DBX60.7 == 1)で停止状態にしてください。

注記

電源オフしたときに機械軸が「精密イグザクトストップ」で停止状態になって**いない**場合は、電源投入したときに実位置が「0」で初期化されます。エンコーダ状態として「原点未確立」が表示されます。

パラメータ設定

自動原点確立は、エンコーダ状態を「自動原点確立が有効ですが、エンコーダは原点確立されていません」に設定することによって有効になります。

MD34210 $MA_ENC_REFP_STATE[<エンコーダ>] = **1**

検出器の原点確立後、次回エンコーダを起動したときに自動原点確立が実行されることを示すエンコーダ状態が表示されます。

MD34210 $MA_ENC_REFP_STATE[<エンコーダ>] == **2**

NC/PLC インタフェース信号

自動原点確立後、有効な検出器に対してエンコーダ状態「原点確立済み」が表示されます。

DB31、... DBX60.4/.5 == **1** (原点確立済み/原点同期済み 1/2)

必要条件

MD34210 $MA_ENC_REFP_STATE[<エンコーダ>] == 1 によるエンコーダ起動

エンコーダ起動時にエンコーダ状態が「1」である場合、「自動原点確立」が有効であることを意味します。ただし、検出器はまだ原点確立されていないか、または機械軸が「精密イグザクトストップ」状態で停止しているときにオフにされませんでした。機械軸または有効な検出器に対して以下が設定されます。

- 実位置 = 0
- 有効な検出器、エンコーダ状態 = 「原点未確立」:
  DB31、... DBX60.4 / .5 = **0** (原点確立済み/原点同期済み 1/2)

参照先

機能マニュアル、基本機能、「R1 原点確立」、「インクリメンタルエンコーダによる原点確立 (ページ 1503)」の章

16.10.2 実位置の復元

機能

実位置を最後に記憶した位置に復元後、有効な検出器のエンコーダ状態は「**復元済み**」に設定されます。軸は手動でのみ移動できます。

AUTOMATIC モード

AUTOMATIC モードでプログラムを自動実行するために NC スタートを有効にするには、機械軸の検出器を再原点確立してください。

MDI モードとオーバストア

MDI モードでオーバストア機能の場合、軸を原点確立しないで、復元された位置で加工をおこなうこともできます。これをおこなうには、以下により、特定のチャネルに対して、復元された位置での NC スタートを明確に有効にしてください。

MD20700 $MC_REFP_NC_START_LOCK = 2

必要条件

コントローラを電源投入したときに有効な検出器がすでに原点確立されているように、電源オフする前に一度、原点確立しておいてください。

パラメータ設定

リリース:実位置の復元

実位置の復元は、エンコーダ状態を「オフにする前に最後に記憶された軸位置が復元され、自動原点確立されません」に設定することによっておこなわれます。

MD34210 $MA_ENC_REFP_STATE[<エンコーダ>] = **3**

リリース:「MDI」と「オーバストア」モードでの NC スタート

「位置復元済み」状態で「MDI」と「オーバストア」モードでパートプログラムまたはパートプログラムブロックを実行するために NC スタートを有効化するには、以下を使用します。

MD34110 $MA_REFP_CYCLE_NR **≠ -1** (チャネル別原点確立のための軸処理)

MD20700 $MC_REFP_NC_START_LOCK = **2** (レファレンス点なしの NC スタートロック)

NC/PLC インタフェース信号

復元された実位置は、レファレンス点復帰後の実位置に相当するものとはみなされません。そのため、「位置復元済み」で「原点確立済み/原点同期済み」でない状態は、機械軸の検出器では表示されません。

復元される実位置:

- DB31、... DBX60.4/.5 = **0** (原点確立済み/原点同期済み 1/2)
- DB31、... DBX71.4/.5 = **1** (**位置復元済み**、エンコーダ 1/2)

検出器原点確立済み:

- DB31, ... DBX60.4/.5 = 0 → **1** (**原点確立済み/原点同期済み** 1/2)
- DB31、... DBX71.4/.5 = 1 → **0** (位置復元済み、エンコーダ 1/2)

注記

移動範囲制限の監視(ソフトウェアリミットスイッチ、作業領域リミットなど)は、「位置復元済み」状態ではすでに有効になっています。

必要条件

主軸

エンコーダ限界周波数を超えた場合、主軸は「原点未確立/原点未同期」状態にリセットされます。

- DB31、... DBX60.4/.5 = 1 → **0** (原点確立済み/原点同期済み 1/2)
- DB31、... DBX71.4/.5 = 1 → **0** (位置復元済み、エンコーダ 1/2)

参照先

機能マニュアル、基本機能、「R1 原点確立」、「インクリメンタルエンコーダによる原点確立 (ページ 1503)」の章

16.11 必要条件

16.11.1 広範な移動範囲

移動範囲が 4096 のエンコーダ回転数より大きい直線軸、ロータリー絶対値エンコーダ EQN 1325、およびパラメータ設定された絶対値エンコーダ範囲
MD34220 $MA_ENC_ABS_TURNS_MODULO = 4096

最大許容移動範囲はインクリメンタルエンコーダの最大許容移動範囲に対応します。

絶対値エンコーダを備えた、無限に回転する回転軸では:

- 任意の数の整数伝達比率を使用できます。
- 絶対値エンコーダを備えた、無限に回転する回転軸を、モジュロ回転軸(移動範囲 0 ～ 360°)としてパラメータ設定することをお勧めします。
  MD30310 $MA_ROT_IS_MODULO = 1
  そうでない場合は、検出器が有効になったとき、回転軸がアブソリュート原点に到達するために長い移動距離が必要になることがあります。

絶対値エンコーダを備えた機械軸:

検出器の再起動後にコントローラが現在の実位置を正確に決定できるよう、検出器がオフのときの機械軸の移動範囲は、絶対値エンコーダの範囲の半分未満としてください。

MD34220 $MA_ENC_ABS_TURNS_MODULO

エンコーダ位置の一意性に関する注記

注記

リニア絶対値エンコーダ

Heidenhain 製 LC181 などのリニア位置エンコーダのアブソリュート値は、使用可能なスケール長に対して常に一義的です。

ロータリー絶対値エンコーダ

ロータリー絶対値エンコーダのアブソリュート値は、特定の最大エンコーダ回転数の範囲内のみで一義的です。

たとえば、Heidenhain 製 EQN 1325 ロータリー絶対値エンコーダでは、0 ～ 4,096 のエンコーダ回転数の範囲で一義的なアブソリュート値が提供されます。

これは、エンコーダの接続方法に応じて、次のような結果をもたらします。

- 負荷にエンコーダが接続された回転軸: 負荷の 4096 回転
- モータにエンコーダが接続された回転軸: モータの 4096 回転
- モータにエンコーダが接続された直線軸: モータの 4096 回転

例:
EQN 1325 ロータリー絶対値エンコーダが直線軸のモータに搭載されています。 有効送りねじピッチが 10 mm の場合、これによって-20.48 から+20.48 m の移動範囲内でアブソリュート値が一義的になります。

16.12 データリスト

16.12.1 マシンデータ

16.12.1.1 NC 別マシンデータ

番号	識別子: $MN_	説明
11300	JOG_INC_MODE_LEVELTRIGGRD	JOG/連続モードの INC/REF

16.12.1.2 チャネルマシンデータ

番号	識別子: $MC_	説明
20700	REFP_NC_START_LOCK	レファレンス点なしで NC スタート無効

16.12.1.3 軸/主軸マシンデータ

番号	識別子: $MA_	説明
30200	NUM_ENCS	エンコーダの数
30240	ENC_TYP	エンコーダタイプのフィードバック値
30242	ENC_IS_INDEPENDENT	エンコーダは独立している
30250	ACT_POS_ABS	解除時の絶対値エンコーダの位置
30270	ENC_ABS_BUFFERING	絶対値エンコーダ:移動範囲の拡張
30300	IS_ROT_AX	回転軸/主軸
30310	ROT_IS_MODULO	回転軸/主軸のモジュロ変換
30330	MODULO_RANGE	モジュロ範囲のサイズ
30340	MODULO_RANGE_START	モジュロ範囲の開始位置
30355	MISC_FUNCTION_MASK	軸機能
31122	BERO_DELAY_TIME_PLUS	プラス方向の BERO 遅延時間
31123	BERO_DELAY_TIME_MINUS	マイナス方向の BERO 遅延時間
34000	REFP_CAM_IS_ACTIVE	レファレンス点スイッチ付きの軸
34010	REFP_CAM_DIR_IS_MINUS	マイナス方向のレファレンス点復帰
34020	REFP_VELO_SEARCH_CAM	リファレンス点復帰速度
34030	REFP_MAX_CAM_DIST	レファレンス点スイッチまでの最大距離
34040	REFP_VELO_SEARCH_MARKER	レファレンス点クリープ速度
34050	REFP_SEARCH_MARKER_REVERSE	レファレンス点スイッチで反転方向
34060	REFP_MAX_MARKER_DIST	原点マークへの最大距離; 絶対番地化スケールによる 2 つの原点マークに対する最大距離
34070	REFP_VELO_POS	レファレンス点位置決め速度
34080	REFP_MOVE_DIST	絶対番地化システムのレファレンス点距離/終点
34090	REFP_MOVE_DIST_CORR	レファレンス点オフセット/アブソリュートオフセット、絶対番地
34092	REFP_CAM_SHIFT	制御回路レファレンス点スイッチは、等間隔の原点マーク付きインクリメンタル検出器によって移動します
34093	REFP_CAM_MARKER_DIST	レファレンス点スイッチ/原点マークの距離

番号	識別子: $MA_	説明
34100	REFP_SET_POS	レファレンス点位置
34102	REFP_SYNC_ENCS	フィードバック値の原点確立検出器への位置合わせ
34104	REFP_PERMITTED_IN_FOLLOWUP	フォローアップモードでの原点確立の有効化
34110	REFP_CYCLE_NR	チャンネル固有参照のための軸シーケンス
34120	REFP_BERO_LOW_ACTIVE	BERO カムの極性変更
34200	ENC_REFP_MODE	原点復帰動作モード
34210	ENC_REFP_STATE	絶対値エンコーダの状態
34220	ENC_ABS_TURNS_MODULO	ロータリーエンコーダの絶対値エンコーダ範囲
34230	ENC_SERIAL_NUMBER	エンコーダシリアル番号
34232	EVERY_ENC_SERIAL_NUMBER	エンコーダシリアル番号の範囲
34300	ENC_REFP_MARKER_DIST	絶対番地化エンコーダの基本原点マークの距離
34310	ENC_MARKER_INC	絶対番地化スケールでの 2 つの原点マーク間の距離
34320	ENC_INVERS	機械システムに対して反転する直線検出器:
34330	REFP_STOP_AT_ABS_MARKER	終点のない、絶対番地化リニア検出器
35150	SPIND_DES_VELO_TOL	主軸速度許容値
36302	ENC_FREQ_LIMIT_LOW	エンコーダ限界周波数の再原点確立
36310	ENC_ZERO_MONITORING	ゼロマーク監視

16.12.2 信号

16.12.2.1 モードグループへの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
モードグループリセット	DB11.DBX0.7	DB3000.DBX0.7
運転機能 REF	DB11.DBX1.2	DB3000.DBX1.2

16.12.2.2 モードグループからの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
有効な運転機能 REF	DB11.DBX5.2	DB3100.DBX1.2

16.12.2.3 チャネルへの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
レファレンス点復帰動作起動	DB21、... .DBX1.0	DB320x.DBX1.0
OEM チャネル信号(HMI → PLC) REF	DB21、... .DBX28.7	-

16.12.2.4 チャネルからの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
レファレンス点復帰動作有効	DB21、... .DBX33.0	DB330x.DBX1.0
リセット	DB21、... .DBX35.7	DB330x.DBX3.7
全軸原点確立済み	DB21、... .DBX36.2	DB330x.DBX4.2

16.12.2.5 軸/主軸への信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
フォローアップモード(要求)	DB31、... .DBX1.4	DB380x.DBX1.4
位置検出器 1/位置検出器 2	DB31、... .DBX1.5-6	DB380x.DBX1.5-6
レファレンス点位置 1 ～ 4	DB31、... .DBX2.4-7	DB380x.DBX2.4-7
マイナス/プラス移動キー	DB31、... .DBX4.6-7	DB380x.DBX4.6-7
レファレンス点復帰の減速	DB31、... .DBX12.7	DB380x.DBX1000.7

16.12.2.6 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
原点確立済み、原点同期済み 1/2	DB31、... .DBX60.4-5	DB390x.DBX0.4-5
フォローアップ有効	DB31、... .DBX61.3	DB390x.DBX1.3
移動指令マイナス/プラス	DB31、... .DBX64.6-7	DB390x.DBX4.6-7
復元済み 1/2	DB31、... .DBX71.4-5	DB390x.DBX11.4-5

S1:主軸 17

17.1 概略説明

主軸の基本的な機能は、加工を容易にするために工具またはワークに回転動作を与えることです。

これを実現するために、機械のタイプに応じて、主軸は後述の機能をサポートする必要があります。

- 主軸の回転方向の入力(M3、M4)
- 主軸速度の入力(S、SVC)
- オリエンテーションを行わない主軸停止(M5)
- オリエンテーションを行う主軸停止/主軸位置決め
  (SPOS、M19、および SPOSA)
- ギヤ切替(M40 ～ M45)
- 主軸/軸の機能(主軸が回転軸に、回転軸が主軸になる)
- ねじ切り(G33、G34、G35)
- フローティングチャックを使用しないタッピング (G331、G332)
- フローティングチャックを使用したタッピング (G63)
- 毎回転送り(G95)
- 周速一定制御(G96、G961、G97、G971)
- プログラマブル主軸速度制限(G25、G26、LIMS=)
- 主軸または主軸モータ上の位置エンコーダ取り付け
- 主軸の最低速度および最高速度の監視
  、主軸のエンコーダ限界周波数および終点の監視
- 位置制御のオン/オフ切り替え(SPCON、SPCOF、M70)
- 以下による主軸機能のプログラミング:
  - パートプログラム
  - シンクロナイズドアクション
  - FC18 を使用した PLC、または単純な主軸起動用の特定主軸インタフェース

17.2 モード

17.2.1 概要

主軸は際限なく回転する軸で、閉ループ位置制御モードまたは閉ループ速度制御モードのいずれかで動作できます。

運転モード

- 制御運転(閉ループ速度制御)
- 揺動運転(閉ループ速度制御)
- 位置決め運転(閉ループ位置制御 I)
- 同期モード
  参照先:
  『機能マニュアル、上級機能』、「主軸同期(S3)」
- フローティングチャックを使用しないタッピング(G331、G332)
  参照先:
  プログラミング説明書、基本編、章:[位置命令|タッピング|補正チェックなしのタッピング(G331 / G332)]
- 軸モード(閉ループ位置制御 I)

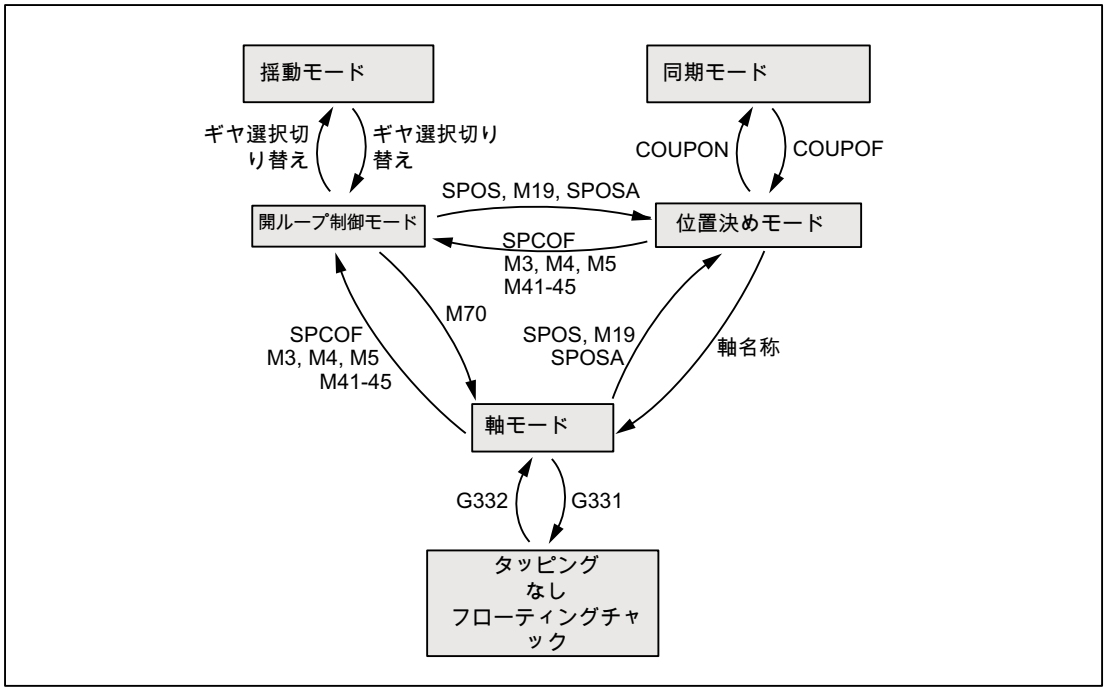
注記

切り替え、軸/主軸モード

主軸と軸の運転に同一モータが使用されている場合、主軸を軸モードと主軸モードで切り替えることができます。

17.2.2 モード変更

主軸/軸操作の切り替えは、以下のように行うことができます。



- コントロールモード → 揺動モード
  新しい S 値と組み合わせた自動ギヤ選択(M40)を使用して、または M41 ～ M45 によって、新しいギヤ選択を指定している場合は、主軸が揺動モードに変わります。 主軸が揺動モードに変わるのは、新しいギヤ選択が現在の実際のギヤ選択とは異なる場合のみです。

- 揺動モード → コントロールモード
  新しいギヤがかみ合うと、インタフェース信号
  DB31、... DBX84.6 (揺動モード)
  がリセットされ、インタフェース信号
  DB31、... DBX16.3 (ギヤ切替済)
  を使用してコントロールモードに変更されます。
  最後にプログラムした主軸速度(S 値)が再度、起動します。

- コントロールモード → 位置決めモード
  オリエンテーションを行う主軸の回転(M3 または M4)を停止するか、停止状態(M5)から再度オリエンテーションを行うには、SPOS、M19、または SPOSA を使用して位置決めモードに切り替えます。

- 位置決めモード → コントロールモード
  主軸のオリエンテーションを終了する必要がある場合は、M3、M4、または M5 を使用してコントロールモードに切り替えます。 最後にプログラムした主軸速度(S 値)が再度、起動します。
- 位置決めモード → 揺動モード
  主軸のオリエンテーションを終了する必要がある場合は、M41 ～ M45 を使用して揺動モードに変更することができます。 ギヤ切替が完了すると、最後にプログラムされた主軸速度(S 値)および M5 (コントロールモード)が再び起動します。
- 位置決めモード → 軸モード
  主軸がオリエンテーションにより停止した場合、割り当てられた軸名を使用して、軸モードへの変更がプログラムされます。 ギヤ選択は保持されます。
- コントロールモード → 軸モード
  コントロールモードから軸モードへの切り替えは、M70 をプログラムして実行することもできます。 この場合、回転中の主軸が M5 と同じ方法で減速し、位置制御が動作中となり、原点パラメータセットが選択されます。
- 軸モード → コントロールモード
  軸モードを終了するには、M3、M4 または M5 を使用してコントロールモードに変更します。 最後にプログラムした主軸速度(S 値)が再度、起動します。
- 軸モード → 揺動モード
  軸モードを終了するには、M41 ～ M45 を使用して揺動モードに変更します(プログラムしたギヤ選択が現在のギヤ選択と異なる場合のみ)。 ギヤ切替が完了すると、最後にプログラムされた主軸速度(S 値)および M5 (コントロールモード)が再び起動します。

17.2.3 コントロールモード

開ループ制御モードを使用するタイミング

主軸は、以下の機能によって開ループ制御モードになります。

- 一定主軸速度:
  - S...M3/M4/M5 および G93、G94、G95、G97、G971
  - S...M3/M4/M5 および G33、G34、G35
  - S...M3/M4/M5 および G63
- 周速一定制御:
  - G96/G961 S...M3/M4/M5

主軸を原点同期済みにさせる必要はありません。

必要条件

以下については、M3/M4/M5 に対する主軸位置フィードバック値エンコーダが必須です。

- 毎回転送り(G95)
- 周速一定制御(G96、G961、G97、G971)
- ねじ切り(G33、G34、G35)
- フローティングチャックを使用しないタッピング(G331、G332)
- 位置制御起動(SPCON、M70)

以下については、M3/M4/M5 に対する主軸位置フィードバック値エンコーダは必要**ありません**。

- インバースタイム送りコード(G93)
- 毎分送り速度(mm/min または in/min)(G94)
- フローティングチャックを使用したタッピング(G63)

速度制御モード

速度制御モードが特に適しているのは、一定主軸速度が必要だが、主軸の位置は重要ではない(たとえば、一定のフライス加工速度でワーク表面を均一に仕上げる)場合です。

- 速度制御モードは、M3、M4、M5 または SPCOF を使用してパートプログラムで起動します。
- 次の NC/PLC インタフェース信号が設定されます。
  DB31、... DBX84.7 (開ループ制御モード)
- NC/PLC IS:
  DB31、... DBX61.5 (位置コントローラ有効)
  は、位置制御を使用しない場合、リセットされます。
- 速度制御モードでの加速は、ギヤ選択とは無関係に、以下のマシンデータで定義します。
  MD35200 $MA_GEAR_STEP_SPEEDCTRL_ACCEL
  できれば、値に物理的状況が反映されるようにしてください。

位置制御モード

位置制御が特に適しているのは、長期間にわたって主軸位置を追従する必要がある場合、または主軸同期の指令値連結を有効にしなければならない場合です。

- 位置制御モードは、パートプログラムで以下を使用してオンにします。SPCON(<主軸番号>)
- 次の NC/PLC インタフェース信号が設定されます。
  DB31、... DBX61.5 (位置コントローラ有効)
- 位置制御モードでの加速は、ギヤ選択とは無関係に、以下のマシンデータで定義します。
  MD35210 $MA_GEAR_STEP_POSCTRL_ACCEL

独立主軸リセット

リセット後またはプログラム終了時(M2、M30)の主軸動作は、以下のマシンデータで設定します。

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET (個々の主軸リセット)

値	意味
0	主軸がリセットされるか、プログラムが終了すると、直ちに主軸が減速して、有効な加速度で停止します。最後にプログラム指令された主軸速度と回転方向が解除されます。
1	リセット時またはプログラムの終了時には、最後にプログラム指令された主軸速度(S 値)と最後にプログラム指令された主軸回転方向(M3、M4、M5)が保持されます。主軸は減速しません。

リセットまたはプログラム終了の前に周速一定制御(G96、G961)が有効である場合は、現在の主軸速度(100%の主軸オーバライドに対する)が、最後にプログラム指令された主軸速度として内部的に使用されます。

主軸は、以下の NC/PLC インタフェース信号でのみ停止できます。

DB31、... DBX2.2 (残移動距離削除/主軸リセット)

回転方向は、主軸の急停止の原因となるアラームが発生すると削除されます。最後にプログラム指令された主軸速度(S 値)は保持されます。アラームの原因を取り除いたら、主軸を再起動してください。

G96 と G961 での主軸動作と主軸の実速度通知

DB31、... DBX61.4 (軸/主軸停止)

主軸が「静止」していると見なされる速度は、以下のマシンデータで設定します。

MD36060 $MA_STANDSTILL_VELO_TOL

停止状態で、以下の NC/PLC インタフェース信号が確実に存在するような値にしてください。

DB31、...DBX61.4 (軸/主軸停止)

DB31,...、... DBX61.4 (軸/主軸停止状態)が通知され、主軸に対して閉ループ位置制御が動作中でない場合は、操作画面でゼロという実速度が表示され、システム変数$AA_S[<n>]でゼロが読み取られます。

周速一定制御(G96、G961)での主軸動作

- 加工開始時(G0 から Gx への遷移)と、NC ストップ、G60(イグザクトストップ、モーダル)および G9 (イグザクトストップ、ノンモーダル)の後に、システムは、実速度が速度指令値許容範囲に達するまで待機し、その後に軸を起動します。
  DB31、... DBX83.5 (nact = nset)
- 大幅な速度の変更が指定されている場合でも、NC/PLC IS:
  DB31、... DBX83.5 (nact = nset)
  および
  DB31、... DBX83.1 (指令速度制限中)は
  定義された値に設定されます(径方向軸が位置 0 方向に移動します)。
- 速度が最低速度を下回るか
  NC/PLC IS:
  DB31、... DBX61.4 (軸/主軸停止)
  が検出されると、NC/PLC IS:
  DB31、... DBX83.5 (nact = nset)
  がリセットされます(たとえば、非常時の機械設計用)。
- 起動している軌跡運転(G64、丸み付け)は中断されません。

また、主軸動作は、以下のマシンデータに影響されます。

MD35500 $MA_SPIND_ON_SPEED_AT_IPO_START (主軸指令一致で送り速度が有効)

ギヤ選択切替の終了時の主軸動作

- NNC/PLC IS:
  DB31、... DBX16.3 (ギヤ切替済み)
  によって、新しいギヤ選択
  (NC/PLC IS DB31、... DBX16.0-16.2 (現在のギヤ選択 A ～ C))
  が適用され、揺動モードが終了したことが NC に通知されます。
  この場合、NC/PLC IS:
  DB31、... DBX18.5 (揺動モード)
  がまだ設定されているかどうかは重要ではありません。
  現在のギヤ選択は、設定されたギヤ選択に対応していなければなりません。
  通知された現在のギヤ選択は、パラメータセットの選択に関連しています。
- ギヤ選択切替(GSW)が PLC (DB31、... DBX16.3)を介して応答されると、主軸は速度制御モード(DB31、... DBX84.7 = 1)になります。
  回転方向(M3、M4、M5 または FC18:「主軸回転開始」)または主軸速度(S 値)をギヤ選択切替の前にプログラム指令している場合は、ギヤ選択切替の後で、最後の速度と回転方向が再び有効になります。

17.2.4 揺動モード

揺動モードは、ギヤ選択切替中に主軸に対して有効になります。

動作モードの詳細説明は、項目「揺動モードでのギヤ選択切替 (ページ 1625)」にあります。

17.2.5 位置決めモード

17.2.5.1 一般機能

位置決めモードを使用するタイミング

主軸位置決めモードでは、定義された位置で主軸が停止され、位置制御が有効になります(解除されるまで有効なままです)。

以下の機能の場合は、主軸が位置決めモードになります。

- SPOS[<n>]=...
- SPOS[<n>]=ACP(...)

- SPOS[<n>]=ACN(...)
- SPOS[<n>]=AC(...)
- SPOS[<n>]=IC(...)
- SPOS[<n>]=DC(...)
- SPOSA[<n>]=ACP(...)
- SPOSA[<n>]=ACN(...)
- SPOSA[<n>]=AC(...)
- SPOSA[<n>]=IC(...)
- SPOSA[<n>]=DC(...)、SPOSA[<n>]=...と同じ
- M19 または M[<n>]=19

<n> =主軸番号のアドレス拡張[<n>]は、主軸に適用できません。

SPOS[<n>]=AC(...)

アブソリュート位置への主軸位置決め(0 ～ 359.999°)。位置決め方向は、現在の主軸回転(主軸が回転している)方向または残移動距離によって決まります。

SPOS[<n>]=IC(...)

最後にプログラム指令位置に対しての相対位置への主軸位置決め(+/- 999999.99°)。位置決め方向は、移動する距離の符号によって定義されます。

SPOS[<n>]=DC(...)

最短距離を通る、アブソリュート位置への主軸位置決め(0 ～ 359.999°)。
位置決め方向は、現在の主軸回転方向によって決まる(主軸回転中)か、あるいは制御装置によって自動的に決められます(主軸停止状態)。

SPOS[<n>]=...

SPOS [<n>]=DC(...)と同じ機能処理です。

SPOS[<n>]=ACP(...)

正の方向から位置にアプローチします。

負の回転方向から位置決めする場合、速度がゼロまで減速された後、逆方向に加速されて、正のアプローチが実行されます。

SPOS[<n>]=ACN(...)

負の方向から位置にアプローチします。

正の回転方向から位置決めする場合、速度がゼロまで減速された後、逆方向に加速されて、負のアプローチが実行されます。

M19 (DIN 66025)

M19 を使用して、主軸を位置決めすることができます。ここで、位置と位置アプローチモードが次のセッティングデータから読み取られます。

SD43240 $SA_M19_SPOS[<n>] (M19 による主軸位置決め用の主軸位置)

SD43250 $SA_M19_SPOSMODE[<n>] (M19 による主軸位置決め用の主軸位置)

M19 の位置決めオプションは以下と同じです。

SPOS = <アプローチモード> <位置/軌跡>

M19 は、M3、M4、M5、および M70 に代わる機能として、NC/PLC インタフェースに補助機能として出力されます。M19 ブロックは、位置決め動作の間、補間器内で有効なままです(SPOS と同様)。

M19 をマクロ(`DEFINE M19 AS SPOS = 0`)として、またはサブプログラムとして使用するパートプログラムは、引き続き実行可能です。以前のコントローラとの互換性を確保するために、下記の例で示したように、M19 の内部処理(NC による主軸の位置決め)を無効にすることができます。

```
MD22000 $MC_AUXFU_ASSIGN_GROUP[0] = 4        ; 補助機能グループ:4
MD22010 $MC_AUXFU_ASSIGN_TYPE[0] = "M"       ; 補助機能のタイプ:"M"
MD22020 $MC_AUXFU_ASSIGN_EXTENSION[0] = 0    ; 補助機能の拡張:0
MD22030 $MC_AUXFU_ASSIGN_VALUE[0] = 19       ; 補助機能値:19
```

自動生成補助機能 M19

NC/PLC インタフェースでの M19 の動作と SPOS または SPOSA の動作が均一となるよう、SPOS と SPOSA の場合には、補助機能 M19 を NC/PLC インタフェースに出力できます。

以下の 2 つの方法で、この機能を起動できます。

- 次のマシンデータを使用して、チャネルのすべての主軸をチャネル別に起動。
  MD20850 $MC_SPOS_TO_VDI (SPOS/SPOSA による PLC への M19 の出力)

ビット	値	意味
0	0	MD35035 $MA_SPIND_FUNCTION_MASK では、ビット 19 も「0」に設定され、SPOS と SPOSA で補助機能 M19 は生成されません。そのため、補助機能の応答時間が削除されます。
	1	補助機能 M19 が生成され、パートプログラムでの SPOS と SPOSA のプログラミング時に PLC に出力されます。アドレス拡張は主軸番号に対応します。

- 次のマシンデータによる、主軸別と複数チャネルにわたる起動
  MD35035 $MA_SPIND_FUNCTION_MASK (主軸機能)

ビット	値	意味
19	0	MD20850 $MC_SPOS_TO_VDI では、ビット 0 も「0」に設定され、SPOS と SPOSA で補助機能 M19 は生成されません。そのため、補助機能の応答時間が削除されます。
	1	自動補助機能 M19 が生成され、SPOS と SPOSA のプログラミング時に PLC に出力されます。アドレス拡張は主軸番号に対応します。

注記

複数のチャネルで主軸を使用するときは、MD35035 による起動を推奨します(軸/主軸交換)。

いずれかのマシンデータの設定が 1 の場合は、補助機能 M19 が自動生成されます。

起動後に、SPOS/SPOSA ブロックの最小所要時間が、PLC による補助機能の出力および確認応答のための時間まで増加します。

自動生成の補助機能出力 M19 の特性は、「高速」と「移動中の出力」です。これらの特性は固定設定値で、補助機能別マシンデータ(MD..._$M..._**AUXFU**_...)での M19 の設定には影響されません。

FC 18 による主軸位置決め命令の場合、自動的に生成される補助機能 M19 は**ありません**。

位置決めの終了

位置決めは、以下の命令を使用してプログラム指令できます。

FINEA[S<n>]:	「精密イグザクトストップ」(DB31、... DBX60.7)到達で動作終了
COARSEA[S<n>]:	「汎用イグザクトストップ」(DB31、... DBX60.6)到達で動作終了
IPOENDA[S<n>]:	「IPO 停止」到達で移動終了

また、単独軸補間を行う IPOBRKA を使用して、ブロック変更の移動終了条件を減速カーブ(100 ～ 0%)に設定できます。

参照先
『機能マニュアル、上級機能』；「位置決め軸(P2)」

ブロック切り替え

現在のブロックでプログラム指令されているすべての主軸または軸の動作終了条件と、軌跡補間のブロック切り替え条件が満たされると、プログラムが次のブロックに進みます。これは、パートプログラムとテクノロジサイクルブロックの両方に当てはまります。

SPOS、M19、および SPOSA には同じ機能がありますが、ブロック切り替えの動作が以下のように異なります。

- SPOS と M19
  ブロックでプログラム指令されているすべての機能がブロック終了条件(すべての補助機能が PLC によって応答され、すべての軸が終点に到達する、など)を満たし、主軸がその位置決め動作を完了した場合に、ブロック切り替えが実行されます。
- SPOSA
  現在のブロックでプログラム指令されているすべての機能(主軸を除く)がブロック終了条件を満たすと、プログラムが次のブロックに進みます。SPOSA がブロック内の唯一の入力値である場合、ブロック切り替えが即座におこなわれます。主軸位置決め動作を、複数のブロックにわたってプログラム指令できます(WAITS を参照)。

協調

動作順序の協調は、以下を使用して実行できます。

- PLC
- マシンデータ設定
- パートプログラムでのプログラミング

PLC

NC/PLC インタフェース信号:
DB31、... DBX83.5 (主軸指令一致)
がない場合、主軸が特定の位置に到達するのを待つために、チャネル別 NC/PLC インタフェース信号:
DB21、... DBX 6.1 (読み込み禁止)
を設定することができます。

マシンデータ設定

MD35500 $MA_SPIND_ON_SPEED_AT_IPO_START = 1
という設定を使用して、主軸が回転して、事前に選択された速度に達した場合のみ、許容範囲
MD35150 $MA_SPIND_DES_VELO_TOL
を考慮に入れた軌跡補間を行うことができます。

設定
MD35500 $MA_SPIND_ON_SPEED_AT_IPO_START = 2
を使用すると、最後の G0 で加工が開始される前に、移動中の軌跡軸が停止します。

以下の状況では、加工が継続します。

- 次の移動指令時
- 主軸速度に達した場合
- MD35510 $MA_SPIND_STOPPED_AT_IPO_START = 1
  (主軸停止状態で軌跡速度有効)の場合

パートプログラムでのプログラミング

パートプログラム内での協調動作には、次のような利点があります。

- ワークの加工開始などのために、プログラムのどのポイントで主軸の速度を上げる必要があるかを、パートプログラムの作成者が決めることができます。
- これにより、不要な遅延が回避されます。

パートプログラム中での協調には、以下の WAITS 命令のプログラミングを使用します。

WAITS:	主軸(マスタ主軸)
WAITS[<n>]:	番号<n>の主軸
WAITS[<n>,<m>,...]:	主軸の最大数までの複数の主軸

⚠ 注意

協調異常

パートプログラムの作成者は、WAITS に関して、以下の保守条件のいずれかが満たされるようにする必要があります。

- 位置決め完了
- 主軸停止状態
- プログラム指令速度までの主軸加速

複数のチャネルで 1 つの主軸を使用する場合、パートプログラムの作成者は、新しい速度または回転方向に向かって別のチャネルの主軸がすでに加速または減速し始めている早期の段階で、WAITS が起動するようにしてください。

コントローラは、以下の時点まで、後続のブロックの実行を待機します。

- SPOSA でプログラムした位置に達する。
- M5 による主軸停止状態に達する
  DB31、... DBX61.4 (主軸停止状態)
  許容範囲を考慮:
  MD36060 $MA_STANDSTILL_VELO_TOL
  最初の信号が検出されると、WAITS が終了し、次のブロックが読み込まれます。
- M3/M4 (速度制御モード)では、速度指令一致は以下のとおりです。
  DB31、... DBX83.5 (主軸指令一致)
  許容範囲を考慮:
  MD35150 $MA_SPIND_DES_VELO_TOL
  最初の信号が検出されると、WAITS が終了し、次のブロックが読み込まれます。
  この WAITS 機能は、プログラム指令チャネルで適用されます。
  WAITS を使用すると、このチャネルが認識しているすべての主軸を待機することができます。ただし、別のチャネルで主軸が起動されている場合もあります。

特別な場合

- **主軸速度の許容範囲:**
  マシンデータの設定が次のような場合、
  MD35150 $MA_SPIND_DES_VELO_TOL = 0
  NC/PLC インタフェース信号
  DB31、... DBX83.5 (主軸指令一致)
  は常に 1 に設定されます。
  WAITS は、速度または方向(M3/M4)の変更後に主軸が指令値側の目標に達すると同時に終了します。
- **許可信号の欠落:**
  WAITS 機能が速度制御モードで「主軸指令一致」信号を待機し、許可信号(軸送り有効、コントローラ、パルスイネーブルなど)が欠落しているために主軸の停止も回転もおこなわれない場合、許可信号を再び受信して、「主軸指令一致」信号が有効になるまでブロックは終了しません。
- **NC ストップ/モードグループ停止に対する動作:**
  NC ストップ/モードグループ停止が WAITS 中に実施されると、NC スタート後に上記のすべての条件で待機運転が再開されます。

注記

複数のチャネルにわたって主軸を使用する場合は特に、プログラミング時に、1 つのチャネルでの WAITS の起動が早すぎること(つまり、別のチャネルの主軸がまだ「古い」速度で回転中の場合)がないようにしてください。
起動が早すぎると、「主軸指令一致」信号が有効になり、WAITS が直ちに停止します。これを防止するために、WAITS の前に WAITM を設定することを強く推奨します。

送り速度

位置決め速度は、次のマシンデータで設定します。

MD35300 $MA_SPIND_POSCTRL_VELO(位置制御切り替え速度)

設定した位置決め速度は、プログラミングまたはシンクロナイズドアクションで変更できます。
`FA[S<n>]=<値>`

各記号 <n>: 主軸番号
の意味
は次の
とおり
です。
<値>: 位置決め速度(°/min)

FA[S<n>]=0 の場合は、設定した速度が有効になります。

加速度

加速度は、次のマシンデータで設定します。

MD35210 $MA_GEAR_STEP_POSCTRL_ACCEL (位置制御モードでの加速度)

MD35200 $MA_GEAR_STEP_SPEEDCTRL_ACCEL (速度制御モードでの加速度)

位置決め中の設定済みダイナミック応答は、プログラミングまたはシンクロナイズドアクションで変更できます。
ACC[S<n>]=<値>

各記号 <n>: 主軸番号
の意味
は次の
とおり
です。
<値>: 設定した加速度のパーセンテージ値による加速度

ACC[S<n>]=0 の場合は、設定した加速度が有効になります。

位置決め処理の中止

位置決め動作は、以下により中止します。

- 以下の NC/PLC インタフェース信号:
  DB31、... DBX2.2 (残移動距離削除/主軸リセット)
- 各リセット(操作パネルによるリセットなど)
- NC ストップによる

中止動作は、次のマシンデータとは無関係です。

MD35040 $MA_SPIND_ACTIVE_AFTER_RESET (個々の主軸リセット)

特記事項

主軸オーバライドスイッチは有効です。

17.2.5.2 回転からの位置決め

初期状態

位置決めを(プログラム中の SPOS、M19 または SPOSA 命令で)開始すると、主軸を速度制御モードまたは位置制御モードにすることができます。

以下の各事例は、区別してください。

ケース 1:	主軸が速度制御モード、エンコーダ限界周波数超過
ケース 2:	主軸が速度制御モード、エンコーダ限界周波超過なし
ケース 3:	主軸が位置制御モード
ケース 4:	主軸速度<位置制御起動速度

処理

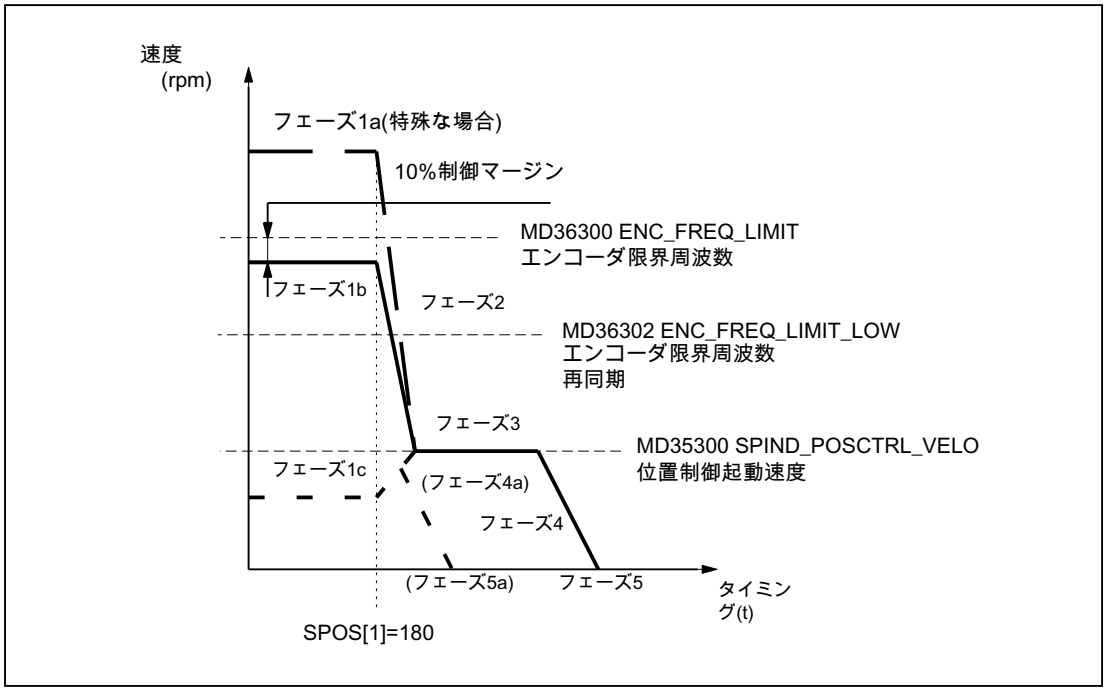


図 17-1 回転からの位置決め

注記

エンコーダの再原点同期のためのエンコーダ限界周波数の設定による速度(MD36302 $MA_ENC_FREQ_LOW)は、位置制御開始速度(MD35300 $MA_SPIND_POSCTRL_VELO)より大きくしてください。

フェーズ 1

フェーズ 1a からの位置決め:

主軸がエンコーダ限界周波数より大きい速度で回転しています。主軸が原点同期されていません。

フェーズ 1b からの位置決め:

主軸がエンコーダ限界周波数より小さい速度で回転しています。主軸が原点同期済みです。

注記

位置制御が有効である場合、速度は、主軸の最高速度またはエンコーダ限界周波数の 90% までしか上がりません(10%の制御確保が必要です)。

フェーズ 1c からの位置決め:

主軸はプログラムした主軸速度で回転します。これは、以下により設定した位置制御起動速度を下回る速度です。

MD35300 $MA_SPIND_POSCTRL_VELO

主軸が原点同期済みです。

フェーズ 2

主軸速度 > 位置制御起動速度

SPOS、M19、または SPOSA 命令を起動すると、以下のように設定した加速度で、主軸が位置制御起動速度へと減速し始めます。

MD35200 $MA_GEAR_STEP_SPEEDCTL_ACCEL

エンコーダ限界周波数のしきい値を超えると、主軸が原点同期されます。

主軸速度<位置制御起動速度

SPOS、M19、または SPOSA をプログラムして、主軸を位置制御モードに切り替えます(このモードになっていない場合)。

位置制御モードの設定済み加速度が起動します。

MD35210 $MA_GEAR_STEP_ POSCTRL_ACCEL

目標点への移動軌跡が計算されます。

最適な時間で主軸がプログラムされた終点に移動します。これは、可能な最高速度(最大MD35300 $MA_SPIND_POSCTRL_VELO)で終点に移動することを意味しています。 適切な二次的条件に応じて、フェーズ 2 - 3 - 4 - 5 または 4a - 5a が実行されます。

フェーズ 3

主軸速度 > 位置制御起動速度

設定した位置制御起動速度(MD35300 $MA_SPIND_POSCTRL_VELO)に達すると、以下のようになります。

- 位置制御が起動します(まだ起動していない場合)。
- 残移動距離(目標点までの)が計算されます。
- 位置制御モードの設定済み加速度に切り替わります(または、この加速度が保持されます)。
  MD35210 $MA_GEAR_STEP_ POSCTRL_ACCEL

主軸速度<位置制御起動速度

終点に到達するために、主軸は、設定した位置制御起動速度(MD35300 $MA_SPIND_POSCTRL_VELO)まで加速されています。 これを超えることはありません。

減速開始点計算により、位置制御モードの設定済み加速度(MD35210 $MA_GEAR_STEP_POSCTRL_ACCEL)で正確に、プログラムした主軸位置に移動できるタイミングが特定されます。

フェーズ 4

主軸速度 > 位置制御起動速度

主軸が、マシンデータ:
MD35210 $MA_GEAR_STEP_POSCTRL_ACCEL
で計算した「減速点」から目標位置まで減速します。

主軸速度<位置制御起動速度

フェーズ 3 の減速開始点計算で特定されたタイミングで、位置制御モードの設定済み加速度(MD35210 $MA_GEAR_STEP_POSCTRL_ACCEL)により、主軸が停止状態まで減速します。

フェーズ 4a:

SPOS 命令を起動すると、終点の近くでは、主軸が、設定した位置制御起動速度(MD35300 $MA_SPIND_POSCTRL_VELO)まで加速できなくなります。

主軸は、位置制御モードの設定済み加速度(MD35210 $MA_GEAR_STEP_POSCTRL_ACCEL)で停止状態まで減速します。

フェーズ 5

主軸速度 > 位置制御起動速度

位置制御が有効な状態を維持し、主軸がプログラムされた位置に保たれます。

注記

主軸位置フィードバック値エンコーダの最大エンコーダ限界周波数が制御装置によって監視されます(超過可能)。位置制御モードでは、指令値速度がエンコーダ制限速度の 90% まで下がります。

次の NC/PLC インタフェース信号が設定されます。

DB31、... DBX83.1 (プログラムした速度が大きすぎる)

指令値速度の減少後にも「MS 限界周波数超過」が検知される場合は、アラームが出力されます。

主軸速度<位置制御起動速度(フェーズ 5、5a)

主軸が停止状態であり、位置に到達しています。位置制御が有効であり、主軸がプログラムされた位置で停止します。

主軸の実位置とプログラムした位置(主軸指令値位置)の間の距離が、設定した精密および汎用イグザクトストップ制限を下回る場合は、以下の NC/PLC インタフェース信号が設定されます。

DB31、... DBX60.7(汎用イグザクトストップ位置決め完了)

DB31、... DBX60.7(精密イグザクトストップ位置決め完了)

イグザクトストップ範囲は、次のマシンデータで定義します。

MD36010 $MA_STOP_LIMIT_FINE (精密イグザクトストップ)

MD36000 $MA_STOP_LIMIT_COARSE (汎用イグザクトストップ)

注記

位置決め条件終了に達して、信号が送られると、位置決め処理が完了したと見なされます。

この条件は「精密イグザクトストップ」です。このことは、パートプログラムからの SPOS、M19 または SPOSA と、FC 18 を使用した PLC によるシンクロナイズドアクションおよび主軸位置決めに適用されます。

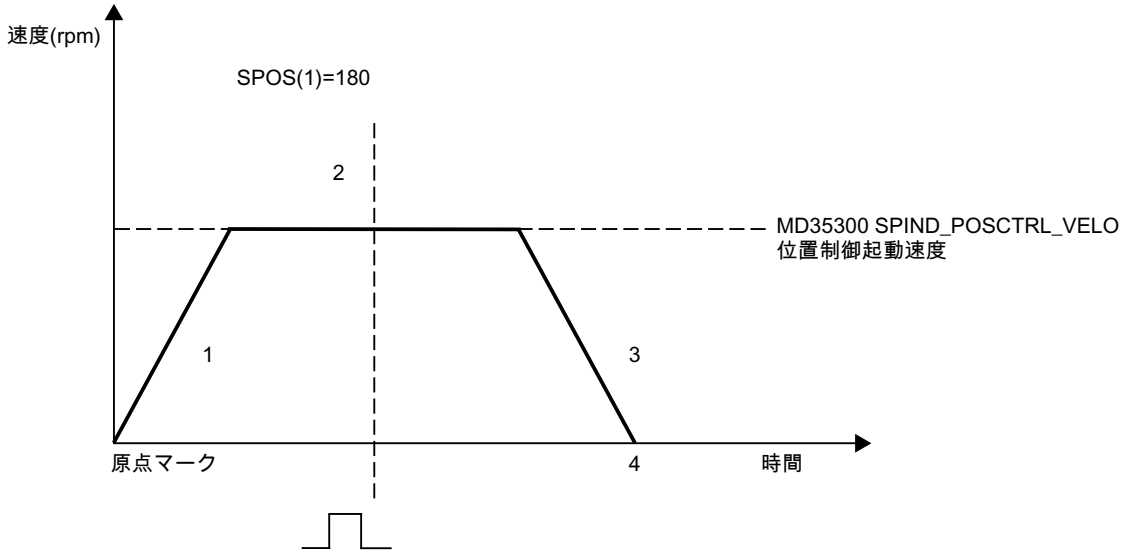
17.2.5.3 停止状態からの位置決め

処理

停止状態からの位置決めに関して、次の 2 つのケースを区別します。

- ケース 1:主軸が原点同期されていません。
  このケースでは、制御および運転の切り替え後、またはギヤ選択切替(たとえば、工具交換のため)後に主軸が位置決めされます。
  MD31040 $MA_ENC_IS_DIRECT = 0
- ケース 2:主軸が原点同期済みです。
  このケースでは、制御および運転を切り替えた後の最初の位置決めアクションの前に、M3 または M4 を使用した最低 1 回の回転によって主軸が回転し、M5 (原点マークによる同期)により停止します。

ケース1:主軸が同期されていません



ケース1:同期主軸

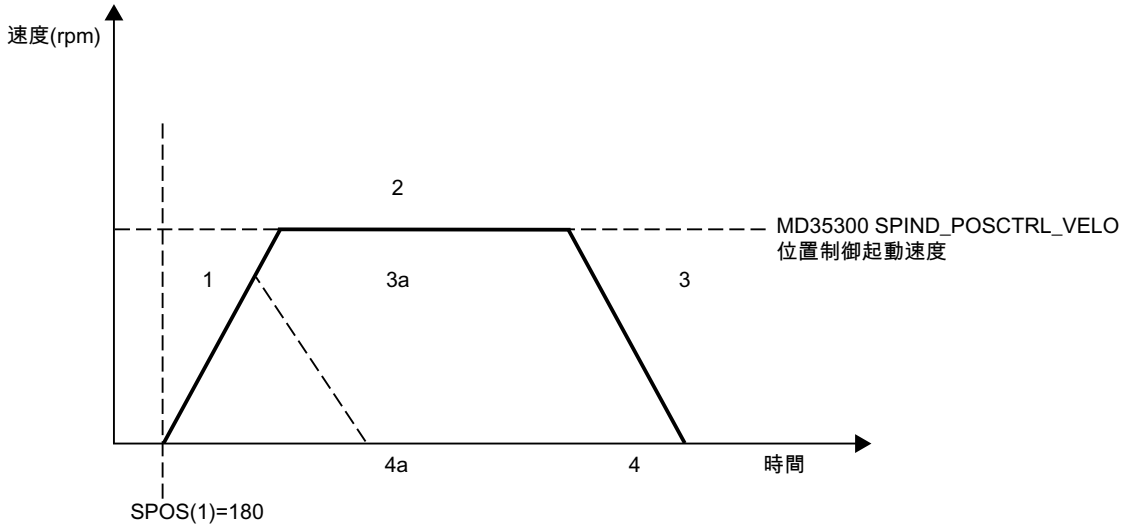


図 17-2　停止状態の主軸の位置決め

フェーズ 1

ケース 1:主軸が原点同期済みではない

SPOS、M19 または SPOSA のプログラミングにより、主軸は次のマシンデータからの加速度で加速されます。

MD35200 $MA_GEAR_STEP_SPEEDCTRL_ACCEL(速度制御モードでの加速度)

この回転方向は、次のマシンデータで定義されます。

MD35350 $MA_SPIND_POSITIONING_DIR (停止状態になるまでの位置決め中の回転方向)

例外:

位置決めに ACN、ACP、IC を使用すると、プログラムした移動方向が有効になります。

主軸が、主軸位置フィードバック値エンコーダの次の原点マークで原点同期され、位置制御モードに切り替わります。

移動軌跡内に原点マークが検出されるかどうかが監視されます(IC を除く)。

MD34060 $MA_REFP_MAX_MARKER_DIST (原点マークまでの最大距離)

マシンデータ:
MD35300 $MA_SPIND_POSCTRL_VELO (位置決め速度)
で定義した速度に到達したが、主軸がまだ原点同期されていない場合は、主軸が位置決め起動速度で回転し続けます(それ以上は加速されません)。

ケース 2:主軸が原点同期済み

SPOS、M19 または SPOSA によって主軸が位置制御モードに切り替わります。

次のマシンデータによる加速度が有効になります。

MD35210 $MA_GEAR_STEP_POSCTRL_ACCEL(位置制御モードでの加速度)

回転方向は、プログラムされた動作(ACP、ACN、IC、DC)または未処理の残移動距離で定義されます。

MD35300 $MA_SPIND_POSCTRL_VELO (位置制御起動速度)
に入力された速度を超過することはありません。

終了位置までの移動軌跡が計算されます。

最適な時間で、主軸がプログラムされた終点に移動します。これは、可能な最高速度(最大 MD35300 $MA_SPIND_POSCTRL_VELO)で終点に移動することを意味しています。適切な二次的条件に応じて、フェーズ 1 - 2 - 3 - 4 または 1- 3a - 4a が実行されます。

フェーズ 2

ケース 1:主軸が原点同期済みではない

主軸が原点同期済みになると、位置制御が起動します。

減速開始点計算によって、プログラムされた主軸位置に正確に、定義された加速度で移動できるタイミングが特定されるまで、主軸は、マシンデータ:

MD35300 $MA_SPIND_POSCTRL_VELO
で設定した最高速度で回転します。

ケース 2:主軸が原点同期済み

終点に到達するために、マシンデータ:
MD35300 $MA_SPIND_POSCTRL_VELO
で定義した速度まで、主軸が加速します。
これを超えることはありません。

減速開始点計算により、マシンデータ:
MD35210 $MA_GEAR_STEP_POSCTRL_ACCEL
で定義されている加速で正確に、プログラムされた主軸位置に移動できるタイミングが特定されます。

フェーズ 1 の減速開始点計算で特定された時点で、主軸は、次のマシンデータで定義されている加速度から停止状態になるまで減速されます。

MD35210 $MA_GEAR_STEP_ POSCTRL_ACCEL

フェーズ 3

フェーズ 2 の減速開始点計算で特定された時点で、主軸は、次のマシンデータで定義されている加速度から停止状態になるまで減速されます。

MD35210 $MA_GEAR_STEP_ POSCTRL_ACCEL

フェーズ 3a:

SPOS 命令が有効になると、終点に接近して、主軸がマシンデータ:
MD35300 $MA_SPIND_POSCTRL_VELO
の値まで加速できないようになります。

主軸が、次のマシンデータで定義されている加速度により停止状態になるまで減速されます。

MD35210 $MA_GEAR_STEP_ POSCTRL_ACCEL

フェーズ 4、4a

主軸が停止状態であり、位置に到達しています。位置制御が有効であり、主軸がプログラムされた位置で停止します。

主軸現在位置とプログラムされた位置(主軸指令値位置)の距離が精密および汎用イグザクトストップ範囲の設定より短い場合は、NC/PLC インタフェース信号:
DB31、... DBX60.6 (汎用イグザクトストップ位置に到達)

および
DB31、... DBX60.7 (精密イグザクトストップ位置に到達)
が設定されます。

これは、次のマシンデータで定義されます。

MD36010 $MA_STOP_LIMIT_FINE

MD36000 $MA_STOP_LIMIT_COARSE

フェーズ 3:

フェーズ 2 の減速開始点計算で特定された時点で、主軸は、次のマシンデータで定義されている加速度から停止状態になるまで減速されます。

MD35210 $MA_GEAR_STEP_ POSCTRL_ACCEL

フェーズ 4:

主軸が停止状態であり、位置に到達しています。位置制御が有効であり、主軸がプログラムされた位置で停止します。

主軸現在位置とプログラムされた位置(主軸指令値位置)の距離が精密および汎用イグザクトストップ範囲の設定より短い場合は、NC/PLC インタフェース信号:
DB31、... DBX60.6 (汎用イグザクトストップ位置に到達)
および
DB31、... DBX60.7 (精密イグザクトストップ位置に到達)
が設定されます。

これは、次のマシンデータで定義されます。

MD36010 $MA_STOP_LIMIT_FINE

MD36000 $MA_STOP_LIMIT_COARSE

17.2.5.4 工具交換用の「主軸インポジション」信号

機能

特にフライス盤の場合の工具交換の移動手順は、主軸の位置決めと、その後の(最適運転の場合は、それと同時の)軌跡軸による工具交換位置へのアプローチで構成されます。この場合、絶対的な必要条件は、工具交換位置にアプローチする前に主軸位置に到着することです。

工具交換サイクルが機械のオペレータによって(たとえば NC ストップ、NC ストップ軸および主軸、モードグループ停止などを使用して)中断された場合、主軸が誤った位置にある工具交換装置に移動することを完全に阻止しなければなりません。

このため、主軸の位置決めでは、最後にプログラム指令された主軸位置に「精密イグザクトストップ」で到達すると、次のような位置をチェックするための NC/PLC インタフェース信号が出力されます。

DB31、... DBX85.5 (主軸インポジション)

注記

この信号は、「主軸位置決め」機能の場合にだけ出力されます。

以下が対象になります。

- パートプログラムでの SPOS、SPOSA および M19
- シンクロナイズドアクションでの SPOS と M19
- FC18 を使った主軸位置決め
- PLC インタフェース(DB31、... DBX30.4)を使用した主軸位置決め

信号の設定

信号 DB31、... DBX85.5 (主軸インポジション)の出力の必要条件は、次のとおりです。

- 主軸の原点確立状態が下記の状態であること。
  DB31、... DBX60.4/5(原点確立済み/原点同期済み 1/2) = 1

  注記

  主軸の位置決め時に、原点マークが自動的に検索されます。このため、異常のない処理を実現するために、原点確立済みの信号は、常に位置決め移動の終了時に使用可能にななります。

- 「精密イグザクトストップ」に到達していること。
  DB31、... DBX60.7 (精密イグザクトストップ) = 1
  最後にプログラム指令された主軸の位置に、指令値側で到達していなければなりません。

信号の解除

信号 DB31、... DBX60.7(精密イグザクトストップ)が取り消されると、信号 DB31、... DBX85.5 (主軸インポジション)も常にリセットされます。

その他の特性

- 位置決め後に、すでに主軸がプログラム指令された位置にあれば、NC/PLC インタフェース信号 DB31、... DBX85.5 (主軸インポジション)は設定されたままになります。
- 位置決め後に(「主軸インポジション」が出力された後に)、主軸がたとえば JOG モードなどで移動すると、NC/PLC インタフェース信号 DB31、... DBX85.5 (主軸インポジション)は解除されます。
  このモードで主軸が元の位置に戻されると、NC/PLC インタフェース信号 DB31、... DBX85.5 (主軸インポジション)が再度、設定されます。最後の位置の選択が維持されます。

17.2.6 軸モード

17.2.6.1 一般機能

機能

たとえば旋盤での端面加工などの特定の加工処理に対して、M3、M4、M5 による速度制御下で主軸のみの移動や、SPOS、M19 または SPOSA での位置決めが十分でない場合、主軸を位置制御の軸モードに切り替えて回転軸として移動することができます。

回転軸機能の例:

- 軸名称によるプログラミング
- ゼロオフセット(G54、G55、TRANS など)
- G90、G91、IC、AC、DC、ACP、ACN
- キネマティック座標変換(TRANSMIT など)
- 軌跡補間
- 位置決め軸として移動

参照先:
機能マニュアル、上級機能; 「回転軸(R2)」の章

必要条件

- 主軸モードと軸モードには同一の主軸モータが使用されます。
- 主軸モードと軸モードに同じ位置検出器または別の位置検出器を使用できます。

- 位置フィードバック値エンコーダは、軸モードの必須要件です。
- 軸モードの使用のために、主軸を G74 などで原点確立してください。
  例：

プログラムコード	コメント
M70	; 主軸を軸モードに切り替え
G74 C1=0 Z100	; 基準軸
G0 C180 X50	; 位置制御されている軸の移動

設定可能な M 機能

主軸を軸モードに切り替えるために使用する M 機能は、次のマシンデータを使用してチャネル別に設定できます。

MD20094 $MC_SPIND_RIGID_TAPPING_M_NR

注記

コントローラは、プログラム処理から、軸モードへの遷移を自動的に検出します("軸モードへの自動遷移 (ページ 1585)"を参照してください)。そのため、主軸の軸モードへの切り替え用に設定されている M 機能を、パートプログラムで明示的にプログラム指令する必要はありません。ただし、パートプログラムの可読性を上げるなどのために、引き続き M 機能をプログラム指令することができます。

特記事項

- 送り速度オーバライドスイッチは有効です。
- 初期設定では、NC/PLC インタフェース信号で軸モードは終了しません。
  DB21、... DBX7.7 (リセット)
- NC/PLC インタフェース信号: DB31、... DBX60.0 == 0 (主軸/回転軸)の場合、
  DB31、 ... DBB16 ～ DBB19 と DBB82 ～ DBB91
  は重要ではありません。
- 軸モードはすべてのギヤ選択で有効にすることができます。
  位置フィードバック値エンコーダがモータ(間接検出器)に取り付けられている場合、位置決めと軌跡精度がギヤ選択ごとに異なります。

- 軸モードが有効な場合、ギヤ選択を変更できません。このために、主軸はまず、M41 ... M45 または M5、SPCOF による制御モードに切り替える必要があります。
- 軸モードでは、1 番目のパラメータセットが有効です(マシンデータインデックス = 0)。
  参照先
  機能マニュアル、基本機能;「速度、指令値/フィードバック系、位置ループ制御 (G2)」、「閉ループ制御」、「位置コントローラのパラメータセット」の章

ダイナミック応答

軸モードでは、軸のダイナミック制限が適用されます。たとえば、

- MD32000 $MA_MAX_AX_VELO[<軸>] (最高軸速度)
- MD32300 $MA_MAX_AX_ACCEL[<軸>] (最高軸加減速)
- MD32431 $MA_MAX_AX_JERK[<軸>] (軌跡移動の最高軸加々速度)

フィードフォワード制御

軸に有効なフィードフォワード制御モードが保持されます。

「ダイナミックフィードフォワード制御」機能の詳細は、以下の資料を参照してください。

参照先
機能マニュアル、上級機能; 「補正(K3」「ダイナミックフィードフォワード制御(追従誤差補正)」の章

例:アナログ駆動部の分解能の切り替え

軸モードへの切り替え

プログラミング	コメント
SPOS=...	
M5	; コントローライネーブルオフ (PLC から) → PLC 上の出力
M70	; スイッチ駆動部 (M70 のアカウント上の PLC から) コントローライネーブルオン (PLC から)
C=...	; NC は軸パラメータセットで移動

主軸モードへの切り替え

プログラミング	コメント
C=...	

プログラミング	コメント
M71	; → PLC への出力 閉ループ制御イネーブルオフ (PLC から) ; スイッチ駆動部 (PLC から) NC の内部で主軸パラメータセット (1-5) に切り替え、コントローライネーブルオン (PLC から)
M3/4/5 または SPOS=...	; NC は主軸パラメータセットで運転

主軸モードへの変更

適切な有効なギヤ選択に対して、パラメータセット 1 ～ 5 が選択されます。

下記が該当する場合、フローティングチャックによるタッピングを除いて、フィードフォワード制御が有効です。

MD32620 $MA_FFW_MODE (フィードフォワード制御モード) ≠ 0

パラメータセット	軸モード	主軸モード
1	有効	-
2	-	有効
3	-	有効
4	-	有効
5	-	有効
6	-	有効
主軸モード:ギヤ選択に対応したパラメータセット		

17.2.6.2 軸モードへの自動遷移

機能

制御装置は、軸モードへの遷移をプログラム処理から自動的に検出し、必要な M70 処理を制御装置内で生成します。状況により、どの手順を実行するかが決まります。ほとんどの場合、以下のことが行われます。

1. 主軸の停止
2. 位置制御の切り替え、フィードフォワード制御の処理、およびパラメータブロックの切り替え
3. ブロック解析の位置同期制御(必要に応じて、内部の先読み停止処理)

この機能は常に動作中です。したがって、基本的には、パートプログラム中に M70 を明示的にプログラムする必要はありません。

手順

軸モードへの自動遷移処理(パートプログラムで M70 をプログラムしていない場合):

- 速度制御モード(M3、M4、M5、SPCOF、...)から軸モードへの遷移:
  制御装置が内部で遷移を検出し、ブロックの入力側に中間ブロックを挿入して軸モードを要求します。作成されたブロックには M70 機能が含まれます。このブロックの実行時間は、プログラムした M70 ブロックの実行に必要な時間と、ほぼ同じです。補助機能 M70 の自動生成および PLC への出力を省略した場合(MD35035 を参照してください)に、主軸が静止(ブレーキ時間なし)しているために短時間で切り替わると、これらの時間が異なる可能性があります。
- 位置決めモード(M19、SPOS、SPOSA)から軸モードへの遷移:
  遷移は、中間ブロックを生成することなく、直ちに実行されます。自動生成され、これに合わせて設定(MD35035 を参照してください)した補助機能 M70 が PLC に出力され、主軸が軸モードであるブロックが読み込まれます。

PLC への補助機能の出力

PLC には、補助機能の出力によって、軸モードへの自動遷移を通知できます。

起動と解除

次のマシンデータを使用して、この機能の起動/解除を実行します。

MD35035 $MA_SPIND_FUNCTION_MASK (主軸機能)

ビット	値	意味
20	0	M70 機能が制御装置内で生成される場合、補助機能は PLC に出力されません。
	1	M70 機能が制御装置内で生成される場合、補助機能 M70 が生成され、PLC に出力されます。アドレス拡張は主軸番号に対応します。

注記

パートプログラム中にプログラムした補助機能 M70 は常に、PLC に出力されます。

特性

自動生成の補助機能出力 M70 の特性は、「高速」および「移動中の出力」です。これらの特性は固定設定値で、補助機能別マシンデータ(MD..._$M..._**AUXFU**_...)での M70 の設定には影響されません。

M70 は、軸モードへの遷移時に 1 度だけ生成されます。補助機能 M70 はそれ以上、生成されず、主軸が軸として動作する隣接ブロックにも出力されません。SPOS、M3、M4、M5、SPCOF などにより軸モードが終了し、軸モードへの遷移が更新されるまで、M70 は自動生成および再出力されません。

必要条件

シンクロナイズドアクション

シンクロナイズドアクションで主軸を軸としてプログラムするときは、アプリケーションが、軸モードへの遷移の条件を継続して満たすようにすることが重要です。

主軸が速度制御モードの場合は、軸としてプログラムする前に命令 M70 または SPOS をプログラムしてください。そうでない場合は、軸のプログラミング時にアラーム信号が発生します。

FC 18

シンクロナイズドアクションの場合と同様に、軸モードへの遷移も、位置決め準備命令などを使用して、FC 18 のアプリケーション側で実行してください。実行しない場合は、FC 18 呼び出しに対して、FC 18 ステータスワードで異常ビットが返されます。

FC 18 でのプログラミングによる軸モードへの遷移の場合、補助機能 M70 は**自動生成されません**。

例

例 1:

パートプログラム:回転主軸から軸モードへの遷移

設定:MD35035 $MA_SPIND_FUNCTION_MASK、ビット 20 = 1

プログラムコード	コメント
N05 M3 S1000	
N10 ...	
N15 POS[C]=77	:N15 を読み込む前に、M70 中間ブロックが生成されて、そこで主軸が停止し、M70 が PLC に出力されます。
…	

例 2:

パートプログラム:位置決めモードから軸モードへの遷移

設定:MD35035 $MA_SPIND_FUNCTION_MASK、ビット 20 = 1

プログラムコード	コメント
N05 SPOS=0	
N10 ...	
N15 C77	; 自動 M70 の PLC への出力、中間ブロックなし。
…	

例 3 :

シンクロナイズドアクション:主軸位置決めモードから軸モードへの遷移

設定:MD35035 $MA_SPIND_FUNCTION_MASK、ビット 20 = 1

プログラムコード	コメント
WHEN COND1==TRUE DO SPOS=180	
WHEN COND2==TRUE DO POS[C]=270	; PLC への自動 M70 の出力。

例 4:

シンクロナイズドアクション:速度制御モードから軸モードへの M70 による遷移

設定:MD35035 $MA_SPIND_FUNCTION_MASK、ビット 20 = 1

プログラムコード	コメント
WHEN COND11==TRUE DO M3 S1000	
WHEN COND12==TRUE DO M70	; PLC への M70 の出力。
WHEN COND13==TRUE DO POS[C]=270	; 軸モードがすでに存在するため、自動 M70 は生成されません。

例 5 :

シンクロナイズドアクション:速度制御モードから軸モードへの無効な遷移

設定:MD35035 $MA_SPIND_FUNCTION_MASK、ビット 20 = 1

プログラムコード	コメント
WHEN COND21==TRUE DO M3 S1000	
WHEN COND22==TRUE DO POS[C]=270	; アラーム 20141 です!

17.2.7 初期主軸状態

主軸の初期設定

次のマシンデータを使用して、初期設定として主軸モードを指定します。

MD35020 $MA_SPIND_DEFAULT_MODE

値	主軸の初期設定
0	速度制御モード、位置制御無効
1	速度制御モード、位置制御有効
2	位置決めモード
3	軸モード

主軸の初期設定が有効になるタイミング

主軸の初期設定が有効になるタイミングは、次のマシンデータで設定します。

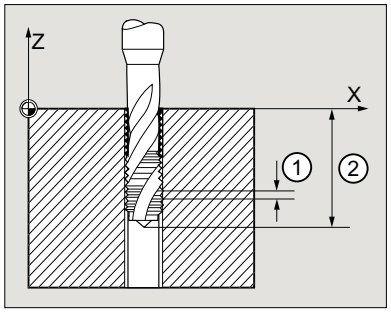
MD35030 $MA_SPIND_DEFAULT_ACT_MASK

値	適用時期
0	電源投入
1	電源投入とプログラム開始
2	電源投入とリセット(M2 / M30)

17.2.8 フローティングチャックのないタッピング

17.2.8.1 機能

フローティングチャックのないタッピングでは、直線軸および主軸の移動動作は補間され、閉ループ位置制御が実行されます。この場合、**位置検出器**付き**位置制御主軸**が必要です。



① ねじピッチ

② ねじ深さ

ねじ回転方向

ねじの回転方向(右ねじまたは左ねじ)は、ピッチの符号によって定義されます。

- 正のピッチ → 右ねじ(M3 に対応する回転方向)
- 負のピッチ → 左ねじ(M4 に対応する回転方向)

主軸開始位置の定義

フローティングチャックなしでタッピングするときは、次の場合で定義された主軸開始位置(SPOS)をプログラムする必要があります。

- 複数の加工動作を使用したねじの製造
- 定義されたねじ開始位置を指定するねじ

```
SPOS = <開始位置>
G331 ...
```

17.2.8.2 プログラミング

フローティングチャックのないタッピングでは、G331 命令および G332 命令を使用して次の移動動作を実行できます。

- G331:ねじ終点までのタッピング方向でのタッピング
- G332:主軸回転方向の自動反転を伴うタッピングブロック G331 への後退動作

構文

```
G331 <軸> <ねじピッチ>
G331 <軸> <ねじピッチ> S...
G332 <軸> <ねじピッチ>
```

意味

G331:	タッピング タップ加工される穴は、軸の移動動作(穴あけ深さ)およびねじピッチによって定義されます。	
	効果:	モーダル
G332:	タッピング時の後退動作 後退動作では、タッピング時(G331)と同じピッチであることが必要です。主軸の回転方向は、自動的に逆になります。	
	効果:	モーダル
<軸>:	ねじ終点でのジオメトリ軸(X、Y、または Z)の移動距離/位置(たとえば、Z50 など)	
<ねじピッチ>:	ねじピッチ I (X)、J (Y)、または K (Z): ● 正のピッチ:右ねじ(例: K1.25) ● 負のピッチ:左ねじ(例: K-1.25)	
	値の範囲:	±0.001 ～±2000.00 mm/回転
S...:	主軸速度 主軸速度を指定しない場合、最後に有効だった主軸速度が使用されます。	

注記

2 番目のギヤ選択データブロック

主軸速度とモータトルクを効果的に適用させて、より短時間で加速するために、その他に、2 つの設定可能なしきい値(最大速度と最小速度)の切り替えのための 2 番目のギヤ選択データブロックを、軸マシンデータに設定しておくことができます。このデータは、1 番目のギヤ選択データブロックとは異なるデータで、1 番目の速度切り替えのしきい値には依存しません。工作機械メーカの指定に従う必要があります。

参照先:
機能マニュアル 基本機能; 主軸(S1)、"設定可能なギヤ調整"

例

- 例:G331/G332 のタッピング (ページ 1592)
- 例:現在のギヤ選択でプログラム指令穴あけ速度を出力 (ページ 1593)
- 例:2 番目のギヤ選択データブロックの用途 (ページ 1593)
- 例:速度はプログラムされておらず、ギヤボックス選択が監視されています (ページ 1594)
- 例:ギヤボックスの選択を変更できません、ギヤボックスの選択は監視されています (ページ 1594)
- 例:SPOS を使用しないプログラミング (ページ 1595)

17.2.8.3 例:G331/G332 のタッピング

プログラムコード	コメント
N10 SPOS[n]=0	; 主軸:位置制御モード
	; 開始位置 0 度
N20 G0 X0 Y0 Z2	; 軸:開始位置にアプローチ
N30 G331 Z-50 K-4 S200	; Z のタッピング、
	; ピッチ K-4 が負 =>
	; 主軸の回転方向:CCW 回転、
	; 主軸速度 200 rpm
N40 G332 Z3 K-4	; Z 方向の後退動作、
	; ピッチ K-4 が負 => (左回り)
	; 回転方向の自動逆転 =>
	; 右回りの主軸回転方向
N50 G1 F1000 X100 Y100 Z100 S300 M3	; 主軸運転中の主軸

17.2.8.4 例:現在のギヤ選択でプログラム指令穴あけ速度を出力

プログラムコード	コメント
N05 M40 S500	; プログラムされた主軸速度:500 rpm =>
	; ギヤボックス選択 1 (20 ～ 1028 rpm)
...	
N55 SPOS=0	; 主軸の位置決め
N60 G331 Z-10 K5 S800	; タッピング
	; 主軸速度 800 rpm => ギヤボックス選択 1

プログラム指令主軸速度 S500 に対応するギヤ選択は M40 で、1 番目のギヤ選択データブロックに基づいて特定されます。プログラム指令穴あけ速度 S800 は、現在のギヤ選択で出力され、必要に応じて、ギヤ選択の最大速度に制限されます。SPOS 命令後のギヤ選択の自動変更はできません。ギヤ選択の自動変更をおこなうためには、主軸を速度制御モードにしてください。

注記

主軸速度 800 rpm でギヤボックス選択 2 を選択する場合は、最大速度と最小速度の切り替えのしきい値を、2 番目のギヤ選択データブロックの当該のマシンデータに設定してください(以下の例を参照してください)。

17.2.8.5 例:2 番目のギヤ選択データブロックの用途

2 番目のギヤ選択データブロックにある最大速度と最小速度の切り替えのしきい値は、動作中のメイン主軸の S 値のプログラミング時に、G331/G332 に対して使用されます。自動 M40 ギヤ選択変更を有効にしてください。上記の方法で特定されたギヤ選択は、動作中のギヤ選択と比較されます。両者が異なることが検出された場合、ギヤボックスの選択が変更されます。

プログラムコード	コメント
N05 M40 S500	; プログラムされた主軸速度:500 rpm
...	
N50 G331 S800	; メイン主軸:ギヤ選択 2 を選択します
N55 SPOS=0	; 主軸の位置決め
N60 G331 Z-10 K5	; タッピング
	; 2 番目のギヤボックス選択データブロック 2 からの主軸の加減速

17.2.8.6 例:速度はプログラムされておらず、ギヤボックス選択が監視されています

2 番目のギヤ選択データブロックの使用時に速度を G331 でプログラム指令していない場合は、最後のプログラム指令速度を使用して、ねじが加工されます。ギヤ選択は変更されません。ただし、この場合は監視がおこなわれ、最後のプログラム指令速度が、動作中のギヤ選択の設定速度レンジ(最大速度と最小速度のしきい値で定義されています)内であることが確認されます。それ以外の場合は、アラーム 16748 が出力されます。

プログラムコード	コメント
N05 M40 S800	; プログラムされた主軸速度:800 rpm
...	
N55 SPOS=0	; 主軸の位置決め
N60 G331 Z-10 K5	; タッピング
	; 主軸速度の監視、800 rpm
	; ギヤ選択 1 が有効です
	; ギヤボックス選択 2 は有効であることが必要です => アラーム 16748

17.2.8.7 例:ギヤボックスの選択を変更できません、ギヤボックスの選択は監視されています

2 番目のギヤ選択データブロックの使用時に、G331 ブロックではジオメトリに加えて主軸速度をプログラム指令していますが、速度が、動作中のギヤ選択の設定速度レンジ(最大速度と最小速度のしきい値で定義されています)内でない場合でも、ギヤ選択を変更できません。これは、主軸と切り込み軸(複数軸の場合もあります)の軌跡移動が維持されないためです。

上記の例のように、速度とギヤ選択が G331 ブロックで監視され、アラーム 16748 が、必要に応じて発生します。

プログラムコード	コメント
N05 M40 S500	; プログラムされた主軸速度:500 rpm =>
	; ギヤボックス選択 1
...	
N55 SPOS=0	; 主軸の位置決め
N60 G331 Z-10 K5 S800	; タッピング
	; ギヤボックスの選択は変更できません、
	; 主軸速度の監視、800 rpm
	; ギヤボックス選択データセット 1:ギヤボックス選択 2
	; は有効であることが必要です => アラーム 16748

17.2.8.8 例:SPOS を使用しないプログラミング

プログラムコード	コメント
N05 M40 S500	; プログラムされた主軸速度:500 rpm =>
	; ギヤボックス選択 1 (20 ～ 1028 rpm)
...	
N50 G331 S800	; メイン主軸:ギヤ選択 2 を選択します
N60 G331 Z-10 K5	; タッピング
	; 2 番目のギヤボックス選択データブロック 2 からの主軸の加減速

ギヤ選択が変更された場合など、以前に処理されたパートプログラム区間により特定される現在位置から、主軸のねじ補間が開始されます。したがって、ねじの再加工ができない場合があります。

注記

複数の主軸による加工の場合は、ドリル主軸もメイン主軸であることが必要です。
SETMS (主軸番号>) をプログラム指令して、ドリル主軸をメイン主軸として設定できます。

17.2.8.9 特記事項:NC プログラムの NC/PLC インタフェース信号による回転方向の反転

通常、NC/PLC インタフェース信号による主軸の回転方向の反転要求"M3/M4 反転" (DB31、... DBX17.6)は、NC プログラムの起動前に PLC ユーザープログラムによって実現されます。開始時にのみ、NC は自動的にインタフェース信号の状態を評価し、必要であれば主軸の回転方向を変更します。

このプログラムが実行中である場合、主軸の回転方向は NC/PLC インタフェース信号 "M3/M4 反転" (DB31、... DBX17.6)の結果として、再度変更する必要があります。その後、NC プログラム内で、記載する基本手順に従うことでのみ、これを実現できます。

N110 SPCOF または M5	主軸を停止し、閉ループ速度制御モードに切り替えます
N111 Mxx または Hxx	PLC ユーザープログラムで次の操作を開始するためにユーザー別にヘルプ機能を表示します。 ● チャネルでの読み込み禁止の設定:DB21、... DBX6.1 = 1 ● 主軸のインタフェース信号"M3/M4 反転"の反転:DB31、...DBX17.6 ● チャネルでの読み込み禁止のリセット:DB21、... DBX6.1 = 0
N112 STOPRE	先読み停止

`N113 SPOS=IC(0.001)` 閉ループ位置制御モードで主軸を段階的に移動します。

閉ループ位置制御モードで主軸を段階的に移動することによって、NC はインタフェース信号 DB31、... DBX17.6 (M3/M4 反転)を再度評価します。

このブロック以降、主軸の移動時に、インタフェース信号に基づいて方向の反転が有効になります。

`N114 SPOS=0` **オプション**:定義された出力位置が技術的に必要な場合に(たとえば 0°など)、主軸を位置決めします。

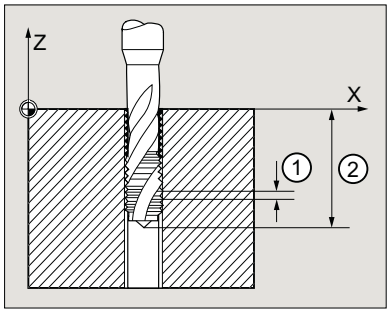
例

プログラムコード	コメント
`N010 G0 X0 Z0`	`; 軸の最初の位置`
`N020 G1 G94 F1000 Z10`	`; 初期位置に Z 軸を移動します`
`N030 M3 S100`	`; 閉ループ速度制御モードで`
	`; プログラムの起動時に決定した主軸の回転方向に 100 rpm により`
	`; 主軸を移動します`
`...`	`; 任意の加工`
`N040 M5`	`; 主軸を停止し、閉ループ速度制御モードに切り替えます`
`N050 H10`	`; 補助機能出力:H0=10`
	`; PLC ユーザープログラム内で`
	`; - 読み込み禁止の設定:DB21、... DBX6.1 = 1`
	`;  - 回転の方向:DB31、...DBX17.6`
	`; - 読み込み禁止のリセット:DB21、... DBX6.1 = 0`
`N060 STOPRE`	`; 先読み停止`
`N070 SPOS=IC(0.001)`	`; 閉ループ位置制御モードで主軸を段階的に移動します。`
`N080 SPOS=0`	`; 主軸の位置決め`
`N090 G331 Z-10 K1 S500`	`; フローティングチャックなしのタッピングです`
`N100 G332 Z2 K1`	`; フローティングチャック後退移動のないタッピングです`
`N110 G1 G94 F1000 Z10`	`; Z 軸の初期位置に後退します`
`N120 SPCOF`	`; 閉ループ速度制御モードに主軸を切り替えます`
`N130 H10`	`; 上記を参照してください`
`N140 STOPRE`	`; 先読み停止`
`N150 SPOS=IC(.001)`	`; 閉ループ位置制御モードで主軸を段階的に移動します。`
`N160 SPOS=0`	`; 主軸の位置決め`
`N170 G331 Z-10 K1 S500`	`; フローティングチャックなしのタッピングです`
`N180 G332 Z2 K1`	`; フローティングチャック後退移動のないタッピングです`
`N190 G1 G94 F1000 Z10`	`; 初期位置に Z 軸を移動します`
`...`	

17.2.9 フローティングチャックを使用したタッピング

17.2.9.1 機能

フローティングチャック(G63)によるタッピング時に、主軸は直線軸で補間されず、閉ループ速度制御モードで動作します。フローティングチャック(G63)によるタッピング時に、主軸は直線軸で補間されず、閉ループ速度制御モードで動作します。主軸と直線軸が相互に補間されないために、主軸速度の変動を補正するために、たとえば長さのフローティングチャックでタッピングドリルが必要です。



後退移動

後退動作も G63 によりプログラムしますが、主軸回転方向は反対になります。

軸および主軸のオーバライド値

"フローティングチャックを使用したタッピング"ファンクションを選択した場合、主軸および軸のオーバライド値として 100 %が有効になります。

送り速度

プログラムされた直線軸の送り速度 F は、主軸速度の製品およびタッピングドリルのねじピッチから取得されます。

F [mm/min] = 主軸速度 S [rpm] *ねじピッチ [mm/U]

17.2.9.2 プログラミング

フローティングチャックによるタッピングでは、G63 命令を使用して次の移動動作を実行できます。

- G63:ねじ終点までのタッピング方向でのタッピング
- G63:プログラムされた主軸反転回転方向の後退動作

注記

G63 ブロックの後、最後に有効だった補間タイプ G0、G1、G2 が有効になります。

構文

G63 <軸> <回転方向> <速度> <送り速度>

意味

G63:	フローティングチャックによるタッピング	
	効果:	ノンモーダル
<軸>:	ねじ終点でのジオメトリ軸(X、Y、または Z)の移動距離/位置(たとえば、Z50 など)	
<回転方向>:	主軸の回転方向: • M3:右回り、右ネジ • M4:左回り、左ねじ	
<速度>:	タッピング時の最大許容主軸速度、例: S100 など	
<送り速度>:	タッピング軸 F の送り速度、F = 主軸速度 * ねじピッチ	

例

M5 ねじのタッピング:

- 標準に準拠した主軸ピッチ:0.8 mm/rev
- 主軸速度 S:200 rpm
- 送り速度 F = 200 rpm * 0.8 mm/rev = 160 mm/min

プログラムコード	コメント
N10 G1 X0 Y0 Z2 F1000 S200 M3	; 起点にアプローチします。
	; 右回りの主軸回転方向、200 rpm
N20 G63 Z-50 F160	; フローティングチャックによるタッピング
	; 穴あけ深さ:絶対値 Z=50mm
	; 送り速度:160 mm/min
N30 G63 Z3 M4	; 後退移動:絶対値 Z=3mm
	; 回転方向の逆転
	; 左回りの主軸回転方向、200 rpm

17.3 原点確立/原点同期

原点同期の必要性

電源投入時の主軸の正確な位置を検出するために、コントローラと主軸の位置検出器の原点を同期する必要があります。

原点同期済み主軸を使用した場合のみ、以下の機能が使用可能です。

- ねじ切り
- フローティングチャックのないタッピング
- 軸指令

主軸の原点同期の詳細については、「R1:レファレンス点復帰 (ページ 1497)」の章を参照してください。

原点確立の必要性

電源投入時の正確な機械原点を検出するために、コントローラを回転軸の位置検出器と原点同期させてください。この処理を原点確立といいます。軸を原点確立するのに必要な一連の処理をレファレンス点復帰といいます。

原点確立された軸のみ、機械上のプログラム指令位置に正確にアプローチできます。

回転軸の原点確立の詳細については、「R1:レファレンス点復帰 (ページ 1497)」の章を参照してください。

位置検出器の設置場所

位置検出器は、次のように設置できます。

- 原点マークエンコーダとしての主軸上の近接スイッチと組み合わせて、モータ上に直接設置
- 原点マークエンコーダとしての主軸上の近接スイッチと組み合わせて、検出器ギヤボックスを使用してモータ上に設置
- 主軸上に直接設置
- 原点マークエンコーダとしての主軸上の近接スイッチと組み合わせて、検出器ギヤボックスを介して主軸上に設置(比率が 1:1 以外の場合に限る)

2 個の位置検出器がある場合は、両方を同じ場所に設置することも、別々の位置に設置することもできます。

原点同期手順

主軸がオンになると、次のように原点同期できます。

- 主軸は、主軸速度(S 値)と主軸回転(M3 または M4)で起動し、位置検出器の次の原点マーク、または近接スイッチからの次の信号と原点同期されます。
- SPOS、M19 または SPOSA を使用して、主軸を停止状態から位置決めしなければなりません。主軸は、位置検出器の次の原点マークまたは次の近接スイッチ信号を使用して原点同期されます。その後、プログラム指令位置に位置決めされます。
- 主軸は、SPOS、M19 または SPOSA を使用して移動(M3 または M4 の後)から原点同期することができます。
  動作は次のとおりです。
  - SPOS=<位置>、SPOS=DC(<位置>)、および SPOS=AC(<位置>)では、移動方向が保持され、その位置にアプローチします。
  - SPOS = ACN(<位置>)または SPOS = ACP(<位置>)では、常に正または負の移動方向で、その位置にアプローチします。.必要に応じて、位置決めの前に動作方向が反転されます。
- 速度制御モードのとき JOG モードの方向キーによる原点マークの通過

注記

この手順は、パートプログラム、FC 18、またはシンクロナイズドアクションのいずれから開始しても、違いはありません。

注記

主軸の原点同期中は、選択した検出器に応じて、4 つの考えられるすべてのレファレンス点位置が有効です。検出器オフセットにも同じ効果があります。

次のマシンデータを確認してください。

- MD34080 $MA_REFP_MOVE_DIST
  (絶対番地化システムのレファレンス点距離/終点)
- MD34090 $MA_REFP_MOVE_DIST_CORR
  (レファレンス点オフセット/アブソリュートオフセット、絶対番地)
- MD34100 $MA_REFP_SET_POS
  (レファレンス点位置、無意味な絶対番地化システムによる)

SPOS=IC(...)と 360°未満の距離による、原点同期していない主軸が位置決めされた場合は、原点マークを通っておらず、主軸位置が原点マークと同期されていない可能性があります。これは、以下の場合に発生することがあります。

- 電源投入後
- 以下の軸 NC/PLC インタフェース信号の設定:
  DB31、... DBX17.5 (位置決め中の主軸の再同期 2)
  DB31、... DBX17.4 (位置決め中の主軸の再同期 1)

非接触近接スイッチを使用して原点同期を行う場合の特記事項

近接スイッチ使用時の信号遅延によって引き起こされる位置の異常は、信号実行時間補正値を入力することで NC において内部的に訂正することができます。

信号実行時間補正は、次のマシンデータを使用して設定します。

- MD31122 $MA_BERO_DELAY_TIME_PLUS
  (近接スイッチの遅延時間、正の動作方向)
- MD31123 $MA_BERO_DELAY_TIME_MINUS
  (近接スイッチの遅延時間、負の動作方向)

この効果は、選択した指令モードに応じて異なります。

MD34200 $MA_ENC_REFP_MODE = <値>

<値>	意味
2	特定の開始速度/速度を入力せずに、位置が同期します。
7	位置を原点同期する開始速度として、次のマシンデータの速度を使用します。 MD34040 $MA_REFP_VELO_SEARCH_MARKER (速度の低下) 原点マークは自動的には検索されません。以下のように、NC/PLC インタフェース信号の 0/1 エッジで明示的に要求してください。 DB31, ... DBX16.4 / 5 (主軸の再原点同期、検出器 1/2)

注記

一般に、システム配信時に信号の実行時間および信号の実行時間の補正がプリセットされるので、変更は必要ありません。

原点確立手順

コントローラに電源を投入した直後に主軸を軸モードでプログラム指令が必要な場合、軸の原点が確立されていなければなりません。

コントローラをオンにすると、主軸の原点を確立できます(条件は、1 回転毎に 1 つの原点マーク)。

原点確立手順についての詳細は、「R1:レファレンス点復帰 (ページ 1497)」の章を参照してください。

主軸に使用した位置検出器を回転軸にも使用した場合は、主軸が原点同期されると同時に、回転軸が原点確立されます(「原点同期手順」の章を参照してください)。

位置検出器、主軸

主軸モードと軸モードで単一のモータが使用される場合、主軸を主軸モードから軸モード(回転軸)に切り替えることができます。

主軸(主軸モードと軸モード)に 1 つまたは 2 つの位置検出器を取り付けることができます。検出器を 2 つ取り付けると、一方の検出器を主軸に、他方の検出器を回転軸にそれぞれ割り当てるか、あるいは両方の検出器を主軸に割り当てることができます。2 つの検出器はどちらもコントローラによって更新されますが、有効になるのは 1 つのみです。

動作中の位置検出器は、NC/PLC インタフェース信号:

DB31、... DBX1.5/6 (位置検出器 1/2)で選択します

有効な位置検出器は、以下の機能で必要とされます。

- 主軸の位置制御(SPCON)
- 主軸の位置決め(SPOS、M19 および SPOSA)
- ねじ切り(G33、G34、G35)
- フローティングチャックを使用しないタッピング(G331、G332)
- 毎回転送り(G95)
- 周速一定制御(G96、G961、G97、G971)
- 主軸実速度表示
- 軸モード
- 主軸同期指令値連結

主軸の位置検出器の再原点同期

以下の場合、主軸位置検出器を再原点同期しなければなりません。

- 位置エンコーダがモータ上にあり、近接スイッチが主軸上に取り付けられており、ギヤ選択が切り替えられる場合。新しいギヤ選択で主軸が回転すると、同期が内部的に起動されます(「原点同期手順」を参照してください)。
- 機械に垂直主軸と水平主軸の選択スイッチがある場合。2 種類の位置エンコーダが使用されますが(1 つは垂直主軸用、もう 1 つは水平主軸用)、コントローラでは 1 つのフィードバック値入力しか使用されません。システムが垂直主軸から水平主軸に切り替わると、主軸を再原点同期しなければなりません。
  この同期は、次の NC/PLC インタフェース信号で開始します。
  DB31、... DBX16.4 (主軸の再原点同期 1)
  または
  DB31、... DBX16.5 (主軸の再原点同期 2)
  主軸は開ループ制御モードでなければなりません。

ウォーム再起動後の位置の復元

インクリメンタル位置検出器を備えた主軸の場合は、座標変換などの位置に依存する機能を復元できるよう、電源切断と電源投入後にフィードバック値を記憶して、オフになる前に最後に記憶された位置を復元することができます(「機械原点の自動復元 (ページ 1545)」の章を参照してください)。アプリケーションの例として、工具オリエンテーションを使った加工時のウォーム再起動後の工具後退があります(「方向座標変換による電源投入後の工具後退 (ページ 783)」を参照してください)。

次の NC/PLC インタフェース信号で、位置の復元後の位置検出器の状態が表されます。

DB31、... DBX71.4 (「復元済み 1」)、位置検出器 1 の場合

DB31、... DBX71.5 (「復元済み 2」)、位置検出器 2 の場合

工具が JOG モードで後退した後は、位置が復元された軸の原点が確立されます。その結果、信号 DB31、.... DBX71.4/5 (「復元済み 1/2」)が削除され、信号
DB31、.... DBX60.4/5 (「原点確立済み/原点同期済み 1/2」)が設定されます。

注記

マシンデータ MD20700 $MC_REFP_NC_START_LOCK が値「2」に設定されている場合、「復元された」軸位置でも NC スタートが可能です(MDI モードまたはオーバストア時)。

17.4 設定可能なギヤ調整

17.4.1 主軸のギヤ選択とギヤ選択切替

ギヤ選択が必要な理由

低速主軸で高いトルクを生成するためにモータの回転速度を段階的に下げる、または高速を維持するために回転速度を段階的に上げるには、ギヤ選択を主軸上で使用します。

ギヤ選択の数

主軸毎に 5 つのギヤ選択を設定できます。

使用するギヤ選択の数は、マシンデータで定義します。

MD35090 $MA_NUM_GEAR_STEPS

ギヤ選択のパラメータ設定

次のマシンデータを使用して、ギヤ選択 1 ～ 5 をパラメータ設定できます。

マシンデータ	意味
MD35012 $MA_GEAR_STEP_CHANGE_POSITION[<n>]	ギヤ選択切替位置
MD35110 $MA_GEAR_STEP_MAX_VELO[<n>]	自動ギヤ選択切替の最高速度
MD35120 $MA_GEAR_STEP_MIN_VELO[<n>]	自動ギヤ選択切替の最低速度
MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>]	ギヤ選択の最高速度
MD35135 $MA_GEAR_STEP_PC_MAX_VELO_LIMIT[<n>]	位置制御でのギヤ選択の最高速度
MD35140 $MA_GEAR_STEP_MIN_VELO_LIMIT[<n>]	ギヤ選択の最低速度
MD35200 $MA_GEAR_STEP_SPEEDCTRL_ACCEL[<n>]	速度制御モードでの加速度
MD35210 $MA_GEAR_STEP_POSCTRL_ACCEL[<n>]	位置制御モードでの加速度
MD35300 $MA_SPIND_POSCTRL_VELO[<n>]	位置制御起動速度

マシンデータ	意味
MD35310 $MA_SPIND_POSIT_DELAY_TIME[<n>]	位置決め遅延時間
MD35550 $MA_DRILL_VELO_LIMIT[<n>]	フローティングチャックを使用しないタッピングの最高速度

ギヤ選択切替のタイプ

ギヤ選択切替のタイプは、次のマシンデータで設定されます。

MD35010 $MA_GEAR_STEP_CHANGE_ENABLE

ビット	値	意味
0	0	主軸モータが主軸に直接取り付けられている(1:1)か、または固定の伝達比によって取り付けられています(初期設定)。 第 1 ギヤ選択のマシンデータが有効です。
	1	最大 5 つのギヤ選択を持つ主軸モータ。 ギヤ選択切替は、以下の状態でおこなわれます。 ● 揺動モードで ● 不確定の変更位置で
1	0	ビット 0 = 0 の場合と同じです。
	1	ビット 0 = 1 の場合と同じですが、ギヤ選択切替は、設定された主軸位置で実行されます。 変更位置は、次のマシンデータで設定されます。 MD35012 $MA_GEAR_STEP_CHANGE_POSITION この位置は、ギヤ選択切替の前に現在のギヤ選択でアプローチします。 ビット 1 が設定されていると、ビット 0 は無視されます。
3	1	NC と PLC の間のギヤ選択切替の交信がシミュレートされます。
5	1	G331/G332 によるタッピング中に、第 2 ギヤ選択データが使用されます(後述の項「第 2 ギヤ選択データセット」を参照してください)。タッピング中に使用されるメイン主軸に対してこのビットを設定してください。

ギヤ選択切替の要件

原則として、ギヤ選択切替は、要求されたギヤ選択が有効なギヤ選択とは異なる場合のみおこなわれます。

ギヤ選択切替中のパラメータセットの選択

以下の設定の場合、サーボパラメータセットもギヤ選択とともに切り替えられます。

MD35590 $MA_PARAMSET_CHANGE_ENABLE = 0 または 1

詳細については、「ギヤ選択切替中のパラメータセット選択 (ページ 1619)」の章を参照してください。

ギヤ選択切替の要求

ギヤ選択切替は次のようにして要求できます。

- パートプログラムで以下の命令を使用して
  - `M40 S...`
    プログラム指令速度 `S...`への自動ギヤ選択切替
  - `M41 ... M45`
    ギヤ選択 1 ... 5 を直接選択
  - `M70`
    MD35014 $MA_GEAR_STEP_USED_IN_AXISMODE = 1 ... 5 の場合
    (「M70 の設定可能なギヤ選択 (ページ 1639)」を参照してください)
  - `G331 S...`
    MD35010 $MA_GEAR_STEP_CHANGE_ENABLE、ビット 5 = 1 の場合
- シンクロナイズドアクションで以下の命令を使用して
  - `DO M40 S...`
    プログラム指令速度 `S...`への自動ギヤ選択切替
  - `DO M41...M45`
    ギヤ選択 1 ... 5 を直接選択
  - `DO M70`
    MD35014 $MA_GEAR_STEP_USED_IN_AXISMODE = 1 ... 5 の場合

- PLC で FC18 ファンクションブロックを使用して
- リセット状態で、下記の NC/PLC インタフェースの記述を通じて
  DB31、... DBX16.0-16.2 (実際のギヤ選択 A ～ C)
  機械的に有効なギヤ選択は、特に POWER ON 後に NC に通知することができます。

注記

主軸モータが主軸に直接取り付けられている場合(1:1)、または固定の伝達比によって取り付けられている場合(MD35010 = 0)、M40 および M41...M45 補助機能はこの主軸と無関係です。

ギヤ選択切替

最高主軸速度の指定による、2 つのギヤ選択間のギヤ選択切替の例を以下に示します。

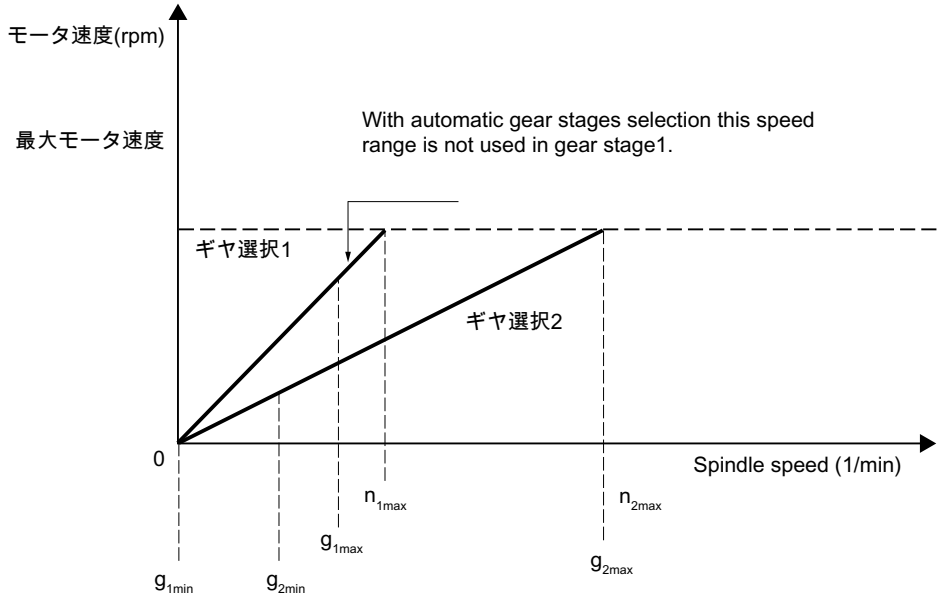


図 17-3　2 つのギヤ選択間の選択によるギヤ選択切替

ギヤ選択切替の処理手順

新しいギヤ選択が事前選択されると、次の処理が実行されます。

1. 切替処理
   次の 2 つの NC/PLC インタフェース信号が設定されます。
   DB31、... DBX82.0-82.2 (ギヤ選択指令 A ～ C)
   DB31、... DBX82.3 (ギヤ選択の切り替え)
   NC/PLC IS:
   DB31、... DBX18.5 (揺動速度)
   が設定された時点から、主軸は、揺動用の加速度により、または速度制御/位置制御用の加速度により停止状態になるまで減速されます。
   遅くとも主軸が停止状態
   DB31、... DBX61.4 (軸/主軸停止)
   に達するまでに、下記の NC/PLC IS を使用して、揺動を有効にすることができます
   DB31、... DBX18.5 (揺動速度)。

   ほとんどの場合、新しいギヤ選択は、揺動なしで使用することもできます。
   新しいギヤが入ると、以下の NC/PLC インタフェース信号が PLC プログラムによって設定されます。
   DB31、... DBX16.0-16.2 (実際のギヤ選択 A ～ C)
   DB31、... DBX16.3 (ギヤ切替済)
2. ギヤ選択切替の終了
   下記の NC/PLC インタフェース信号が設定されている場合、ギヤ選択切替は完了したとみなされます(主軸運転タイプ「揺動モード」選択解除)。
   DB31、... DBX16.3 (ギヤ切替済)
   モータが停止状態のとき、新しい実際のギヤ選択がサーボパラメータセットと補間パラメータセットに変更されます。
   NC/PLC インタフェース信号
   DB31、... DBX16.3 (ギヤ切替済)
   を使用して、新しいギヤ選択が有効であり、揺動モードを完了できることを、NC に通知できます。
   NC/PLC インタフェース信号:
   DB31、... DBX82.3 (ギヤ切替)は、
   NC によってリセットされます。
   これによって、PLC プログラムが NC/PLC IS:
   DB31、... DBX16.3 (ギヤ切替済み)をリセットします。

   この場合、NC/PLC IS:
   DB31、... DBX18.5 (揺動モード)
   がまだ設定されているかどうかは重要ではありません。
   表示される実際のギヤ選択はパラメータ選択と関係しており、
   指令されたギヤ選択に対応している必要があります。
   そうでない場合、アラーム 22010 :
   MD11410 $MN_SUPPRESS_ALARM_MASK、ビット 3 = 0
   が出力されます。
   NC/PLC IS:DB31、... DBX16.3 (ギヤ切替済)による PLC
   経由のギヤ選択切替の確認応答に続いて、
   主軸が速度制御モードになります
   (DB31、... DBX84.7)。
   PLC と NC 間での信号交信について詳しくは、「A2: NC/PLC 間の各種共通インタフェース信号と各種機能 (ページ 45)」の章を参照してください。

第 2 ギヤ選択データセット

自動ギヤ選択切替 M40 は、第 2 の設定可能なギヤ選択データセットで拡張することができます。

第 2 ギヤ選択データセットを、フローティングチャックを使用しないタッピング(G331、G332)について使用した場合**のみ**、主軸速度とモータトルクを効果的に調整できます。

次のようにメイン主軸用のビットを設定することで、適用が実行されます。

MD35010 $MA_GEAR_STEP_CHANGE_ENABLE、ビット 5 = 1

第 2 のギヤ選択データセットで使用されるギヤ選択の数は、次のマシンデータによって定義されます。

MD35092 $MA_NUM_GEAR_STEPS2

以下の場合、第 2 ギヤ選択ブロックデータセットは無効になります。

MD35092 $MA_NUM_GEAR_STEPS2 = 0 (初期設定)

M40 が有効の場合、ギヤ選択は第 1 ギヤ選択データセットを通じておこなわれます。

注記

第 2 データセットのギヤ選択の数が、第 1 データセットと異なる場合があります。M40 にプログラム指令速度に適切なギヤ選択が見つからない場合、ギヤ選択は実行されません(例外については、「M40: 設定された切り替えレベルを超えた速度の自動ギヤ選択 (ページ 1676)」を参照してください)。
フローティングチャック G331/G332 を使用しないねじ切りでの代表的なプログラム手順の詳細については、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 基本編; 動作命令

第 2 ギヤ選択データセットのギヤ選択 1 ～ 5 を、次のマシンデータを使用してパラメータ設定できます。

マシンデータ	意味
MD35112 $MA_GEAR_STEP_MAX_VELO2[n]	自動ギヤ選択切替の最高速度
MD35122 $MA_GEAR_STEP_MIN_VELO2[n]	自動ギヤ選択切替の最低速度
MD35212 $MA_GEAR_STEP_POSCTRL_ACCEL2[n]	位置制御モードでの加速度

注記

機械的係数に関するサーボパラメータセットの数は変わりません。また、主軸に対して5つの機械的ギヤ選択と、軸運転に対して 1 つの機械的ギヤ選択を設定することができます。

以下のマシンデータを使用して、さまざまな切り替え検出値とは無関係に、各ギヤ選択について 1 回のみ速度制限を設定できます。

マシンデータ	意味
MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[n]	ギヤ選択の最高速度
MD35140 $MA_GEAR_STEP_MIN_VELO_LIMIT[n]	ギヤ選択の最低速度

フローティングチャックを使用しないタッピング(G331、G332)では、さらに速度をモータの直線加減速範囲に制限できます。このために、モータの直線特性範囲の最高速度を、ギヤ選択に応じて次のマシンデータで指定します。

MD35550 $MA_DRILL_VELO_LIMIT[n]

パートプログラムでのギヤ選択の指定

M40 実行による自動選択

ギヤ選択は、コントローラによって自動的に選択されます。この場合、プログラムされた主軸速度(S...)が可能なギヤ選択がチェックされます。これによって、現在の実ギヤ選択と一致しないギヤ選択が行われた場合、次の NC/PLC インタフェース信号が設定されます。

DB31、... DBX82.3 (ギヤ選択の切り替え)

DB31、... DBX82.0-82.2 (ギヤ選択指令 A ～ C)

適切なギヤ選択の決定の際、新しい速度が有効なギヤ選択の許容速度範囲内にない場合にだけ、ギヤ選択切替が要求されます。

速度が現在のギヤ選択の最高速度に制限されるか、または現在のギヤ選択の最低速度まで引き上げられ、適切な NC/PLC インタフェース信号が設定されます。

DB31、... DBX83.1 (速度指令値制限)

DB31, ... DBX83.2 (速度指令値増加)

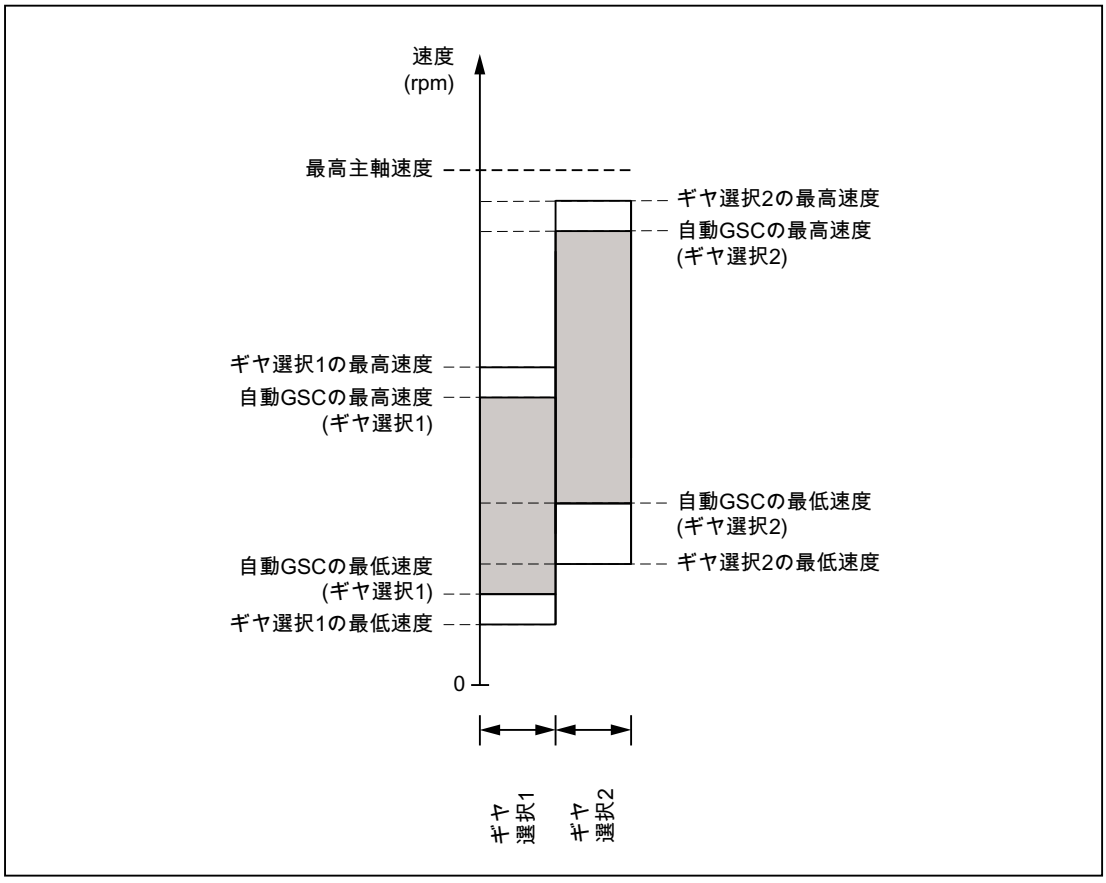


図 17-4　自動ギヤ選択切替(M40)のための、速度範囲が重なる 2 つのギヤ選択の例

注記

M40 では、S ワードによる自動ギヤ選択切替のために、主軸が開ループ制御モードでなければなりません。そうでない場合、ギヤ選択切替は拒否され、次のアラームが設定されます。

アラーム 22000「ギヤ選択切替はできません」

注記

自動ギヤ選択切替のための選択では、有効な減速ギヤは考慮されません。

M41 ～ M45 を使用したギヤ選択の直接指定

パートプログラムで M41 ～ M45 を使用して、ギヤ選択を直接指定できます。

M41 ～ M45 を使用して、現在の(実際の)ギヤ選択とは異なるギヤ選択を指定すると、次の NC/PLC インタフェース信号が設定されます。

DB31、... DBX82.3 (ギヤ選択の切り替え)

DB31、... DBX82.0-82.2 (ギヤ選択指令 A ～ C)

プログラム指令された主軸速度(S...)は、この直接指定されたギヤ選択を基準にします。

- 主軸速度がプログラム指令されていて、直接指定されているギヤ選択の最高速度(MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT)を超えている場合、速度はこの最高速度まで下げられ、次の NC/PLC インタフェース信号が設定されます。
  DB31、... DBX83.1 (速度指令値制限)
- 主軸速度がプログラム指令されていて、直接指定されているギヤ選択の最低速度(MD35140 $MA_GEAR_STEP_MIN_VELO_LIMIT)を下回っている場合、速度はこの最低速度まで引き上げられ、次の NC/PLC インタフェース信号が設定されます。
  DB31, ... DBX83.2 (速度指令値増加)

ブロック切り替え

パートプログラムでギヤ選択切替をプログラム指令すると、ギヤ選択切替は PLC によって中止されるまで有効のままです。

これは、次の NC/PLC インタフェース信号が設定されているのと同じような動作になります。

DB21、... DBX6.1 (読み込み停止)

FC18 を使用した PLC によるギヤ選択の指定

ギヤ選択切替は、パートプログラムの実行中、リセット状態、またはすべての運転モードで、ファンクションブロック FC18 を使用しておこなうこともできます。

回転の速度と方向が FC18 で指定されている場合、速度に適したギヤを選択するよう NC に要求することができます。これは、M40 による自動ギヤ選択切替に対応しています。

以下の場合、ギヤ選択は切り替えられません。

- 主軸が FC18 によって位置決めされている。
- 主軸が軸モードで移動している。

FC18 ファンクションブロックについて詳しくは、「P3: SINUMERIK 840D sl 用 PLC 基本プログラム (ページ 1043)」の章を参照してください。

シンクロナイズドアクションでのギヤ選択の指定

シンクロナイズドアクションによって以下のようにギヤ選択切替を要求できます。

- `DO M40 S...`
  プログラム指令速度 `S...`への自動ギヤ選択切替
- `DO M41...M45`
  ギヤ選択 1 ... 5 を直接選択
- `DO M70`
  MD35014 $MA_GEAR_STEP_USED_IN_AXISMODE = 1 ... 5 の場合
  (「M70 の設定可能なギヤ選択 (ページ 1639)」を参照してください)

以下の場合、ギヤ選択は切り替えられ**ません**。

- 主軸がシンクロナイズドアクションによって位置決めされている。
- 主軸が軸モードで移動している。

注記

詳細については、「パートプログラムでのギヤ選択の指定」の章を参照してください。

例外:

ブロック切り替えは、シンクロナイズドアクションでのギヤ選択の指定によって影響されません。

ギヤ選択の手動での指定

実行中のパートプログラムの外側で、NC または機械からの要求なしでギヤ選択を変更することもできます。これは、たとえば、ギヤ選択が手動で切り替えられる場合などです。

適切なパラメータセットを選択するには、現在のギヤ選択を NC に通知しなければなりません。このためには、コントローラまたはパートプログラムをリセット状態にしてください。

必要条件

ギヤ選択の NC への伝送は、
NC/PLC IS:
DB31、... DBX16.0-16.2 (実際のギヤ選択 A ～ C)が変更されると開始されます。

運転中は、これらの 3 つのビットを常に設定していなければなりません。

伝送が成功すると、NC/PLC IS:
DB31、... DBX82.0-82.2 (ギヤ選択 A ～ C の設定)
によって PLC に対する確認応答がおこなわれます。

NC/PLC IS:
DB31、... DBX16.3 (ギヤ切替済み)
を設定しないでください。

DB31、... DBX16.0-16.2 を使用して PLC による新しいギヤ選択が指定されたとき位置制御が有効である場合、この切り替え処理の間、位置制御がオフになります。

ギヤ選択切替中の NC ストップ

以下の場合、NC/PLC IS:
DB21、... DBX7.4 (NC ストップ)
によって主軸を停止することはできません。

- 主軸がギヤ選択切替用の揺動モードになっていない。
- NC/PLC IS:
  DB31、... DBX16.3 (ギヤ切替済み)
  が設定されていない。

注記

中止の方法:

DB31、... DBX2.2 (残移動距離削除/主軸リセット)
または
DB31、... DBX16.3 (ギヤ切替済)
下記の実際のギヤ選択からの対応する確認応答付き
DB31、... DBX16.0-16.2 (実際のギヤ選択)

ギヤ選択切替後の主軸動作

ギヤ選択が変更された後の主軸の挙動は、以下の初期条件によって決まります。

- 主軸が、ギヤ選択切替の前に停止状態であった(M5、FC18:「回転主軸の停止」)、あるいは位置決めモードまたは軸モードであった場合、ギヤ選択切替の完了後に M5 (主軸停止)が有効になります。
- 回転の方向
  (M3、M4、FC18:「主軸回転の開始」)がプログラムされていた場合、ギヤ選択切替後に、その回転の最後の速度と方向が再度、有効になります。新しいギヤ選択では、主軸が、最後のプログラム指令された主軸速度(S...)まで加速されます。
- ギヤ選択切替(SPCON)の前に位置制御が有効であった場合、ギヤ選択切替の後に位置制御が再び有効になります。
  パートプログラムの次のブロックを実行できます。

特記事項

ギヤ選択切替では、以下の点を遵守してください。

- ギヤ選択切替は、
  NC/PLC インタフェース信号:
  DB31、... DBX20.1 (V/f モードへの起動切り替え)の選択では終了しません。

  指令値 0 が出力されます。
  ギヤ選択切替の確認応答は、通常どおり
  下記の NC/PLC インタフェース信号を介しておこなわれます。
  DB31、... DBX16.3 (ギヤ切替済)
- PLC によってギヤ選択切替を完了する前に、PLC によって「ランプ関数発生器急停止」信号をリセットしてください。
- ギヤ選択切替の処理手順は、アラーム出力なしで NC リセット中に終了します。
  NC/PLC IS:
  DB31、... DBX16.0-16.2 (実際のギヤ選択 A ～ C)
  を使用して出力されるギヤ選択が NC によって適用されます。

FC17 によるスター/デルタ切り替え

ディジタル主軸ドライブは、主軸が運転中であっても、FC17 を使用してスターとデルタ間の切り替えが可能です。この自動切り替えは、FC17 で定義されている論理回路によって制御されます。この回路により、関連する主軸の切り替え時間を設定できます。

FC17 ファンクションブロックについて詳しくは、「P3: SINUMERIK 840D sl 用 PLC 基本プログラム (ページ 1043)」の章を参照してください。

17.4.2 主軸ギヤ選択 0

技術的背景

主軸の負荷側ギヤの切替が可能な機械では、モータと負荷(ワーク/工具)との間のギヤ接続が切り離されることがあります。 このような状態は、たとえば、ギヤ選択切替の実行中に RESET または非常停止を押した場合や、機械の据え付け中に最初のセットアップを行う場合などに発生します。 制御装置は、ギヤ接続が切り離されている状態を認識し、次のギヤ選択切替要求を無条件に実行しなければなりません。

機能

ギヤが外れた場合、バイナリコード値「0」 (≙ ギヤ選択 0)が、インタフェース信号のビット DB31、... DBX16.0-2 (実際のギヤ選択 A ～ C)を使用して、PLC から NC に伝送されます。

DB31、... DBX16.0-2 = 0

この値は、ギヤ接続が切り離されている状態を識別するために、制御装置で使用されます。

ギヤ選択切替時の影響

パートプログラムでのギヤ選択切替

PLC から通知された実際のギヤ選択が、パートプログラムの起動時に NC によって読み取られます。 この時点で、実際のギヤ選択に対して値「0」が読み取られた場合、次のギヤ選択切替が実行され、ギヤ選択切替の処理が PLC によって行われます。 「0」より大きい値が読み取られた場合は、プログラムですでに、要求されたギヤ選択と現在のギア選択の比較が行われています。 両方のギヤ選択が同じ場合、ギヤ選択は切り替えられず、軌跡移動がプログラムされていた場合も、その動作は中断されません。

シンクロナイズドアクションでのギヤ選択切替、FC18 および DBB30

PLC から通知された実際のギヤ選択は、ギヤ選択の切替時に NC によって常に検査されます。 値「0」が NC によって読み取られた場合、ギヤ選択は常に切り替えられます。 「0」より大きい値が読み取られた場合、要求されたギヤ選択と現在のギヤ選択が比較されます。 2 つの値が等しくない場合のみ、ギヤ選択は PLC によって切り替えられ、NC/PLC インタフェース信号 DB31、... DBX82.3 (ギヤ選択の切替)が出力されます。

境界条件

- **DB31、... DBX16.0-2 = 0 の出力**
  ギヤが外れた場合、PLC は NC/PLC インタフェース DB31、... DBX16.0-2 (実際のギヤ選択 A ～ C)にギヤ選択 0 を入力しなければなりません。
- **ギヤ選択切替の有効化**
  ギヤ選択 0 に到達した後のギヤ選択切替の必要条件は、マシンデータを介したギヤ選択切替の全般的な有効化です。
  MD35010 $MA_GEAR_STEP_CHANGE_ENABLE (ギヤ選択切替へのパラメータ割り当て)
  MD35090 $MA_NUM_GEAR_STEPS (設定したギヤ選択の数)
  MD35092 $MA_NUM_GEAR_STEPS2 (第 2 ギヤ選択データセット: 作成されているギヤ選択の数)、MD35010 $MA_GEAR_STEP_CHANGE_ENABLE、ビット 5 = 1 (フローティングチャックなしのタッピング)の場合
- **PLC ユーザープログラム/POWER ON ASUB**
  PLC ユーザープログラムまたは POWER ON ASUB では、主軸移動前にギヤが外された場合(ギヤ選択 0)は、ギヤ選択切替要求がプログラムされていることを確認してください。 これは、たとえば、ASUB で `M41` を使用して実現できます。 JOG または軸運転などでの主軸移動では、ギヤ選択切替そのものは行われません。

例

電源投入後に第 1 ギヤ選択に切り替えるための手順の例

1. 電源投入。
2. PLC ユーザープログラムが、機械側で「ギヤが外れている」状態を特定します。
3. PLC が「ギヤが外れている」状態を、以下を設定して NC に伝送します。
   DB31、... DBX16.0-2 = 0
4. パートプログラム起動または POWER ON ASUB。
5. N05 (パートプログラム、下記を参照)が実行されます。
   ギヤがギヤ選択 1 に切り替えられます。
   NC から
   – 次の NC/PLC インタフェース信号が設定されます。
     DB31、... DBX82.3 (ギヤ選択の切り替え)
   – ギヤ選択 1 の指令が PLC に通知されます。
     DB31、... DBX82.0 = 1
     DB31、... DBX82.1 = 0
     DB31、... DBX82.2 = 0

6. 機械側のギヤ選択切替、確認応答
   ギヤ選択が切り替えられると、PLC から
   – 次の NC/PLC インタフェース信号が設定されます。
     DB31、... DBX16.3 (ギヤ切替済)
   – 現在のギヤ選択 1 が NC に通知されます。
     DB31、... DBX16.0 = 1
     DB31、... DBX16.1 = 0
     DB31、... DBX16.2 = 0
7. N80 が実行されます。
   パートプログラムでギヤ選択切替の回数を最適化するため、ギヤ選択は切り替えられません。

パートプログラム:

プログラムコード	コメント
N05 M41	; Select 1st gear stage
...	
N80 M41	; No gear stage change, if the 1st gear stage is selected.

主軸 1 (AX5)のデータを、次のように設定します。

MD35010 $MA_GEAR_STEP_CHANGE_ENABLE[AX5] = 1 (ギヤ選択切替有効)

17.4.3 主軸ギヤ選択の特定

主軸の実際のギヤ選択を、システム変数を使用して読み取ることができます。

- 操作画面での表示、シンクロナイズドアクション、またはパートプログラムでの先読み停止処理の場合は、以下のシステム変数を使用します。

$VC_SGEAR[<n>]　現在選択されている主軸ギヤ選択

$VC_SGEAR は、PLC から通知された実際のギヤ選択を読み取ります。

値の範囲:　0 ... 5

$AC_SGEAR[<n>] 現在の主軸ギヤ選択

$AC_SGEAR は、メインランのギヤ選択指令を読み取ります。

値の範囲: 1 ... 5

主軸のデータセットは、このギヤ選択に応じて有効になります。

注記

ブロック検索の場合は、検索中はギヤ選択切替が行われないので、実際のギヤ選択($VC_SGEAR[<n>])がギヤ選択指令($AC_SGEAR[<n>])と異なることがあります。そのため、$VC_SGEAR[<n>]と$AC_SGEAR[<n>]を使用して、ブロック検索後にギヤ選択切替を行うかどうかを問い合わせることができます。

- 次のシステム変数は、パートプログラムでは先読み停止をおこないません。

$P_SGEAR[<n>] ギヤ選択指令

$P_SGEAR は、パートプログラムでプログラム指令されたギヤ選択(M41 ... M45)、または M40 の場合は選択されたギヤ、M70 の場合は設定されたギヤ選択を読み取ります。

$P_SEARCH_SGEAR[<n>] ブロック検索ギヤ毎の M 機能

$P_SEARCH_SGEAR には、ブロック検索で収集された、最後にプログラム指令されたギヤ選択 M 機能が入っています。

17.4.4 ギヤ選択切替中のパラメータセット選択

サーボパラメータセット

サーボパラメータセット 1 ～ 6 を使用して、主軸のギヤ切替中に変更された機械特性に位置コントローラを合わせます。

ギヤ選択切替時のパラメータセットの選択

ギヤ選択パラメータセット(補間パラメータ)と次のマシンデータの設定によっては、サーボパラメータセットもギヤ選択切替の間に変更されます。

MD35590 $MA_PARAMSET_CHANGE_ENABLE (パラメータセット変更可能)

値	意味
0	システム内パラメータセット選択 特定のサポートパラメータセットが割り当てられます。 これは以下のように適用されます。 ● 軸と主軸が軸モードである場合、原則として最初のパラメータセットが有効です。 例外: G33、G34、G35、G331、および G332 の場合、関連する軸に対して、次の番号のパラメータセットが有効になります。 メイン主軸ギヤ選択 + 1 (パラメータセット番号 2 ... 6 に相当) ● 主軸モードの主軸の場合、パラメータセットはギヤ選択に合わせて設定されます。
1	システム内パラメータセット選択の他に、以下の方法による「外部」パラメータセット選択というオプションもあります。 ● PLC (DB31、... DBX 9.0 - 9.2)による ● 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の SCPARA のプログラミングによる ただし、システム内パラメータ選択に優先順位があります。 注:値 1 は軸にのみ関連します。
2	サーボパラメータセットが、PLC (DB31、... DBX 9.0 - 9.2)によって、あるい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における SCPARA のプログラミングによって**のみ**指定されます(軸と主軸に対して)。 第 1 パラメータセットが電源投入後に選択されます。

主軸モード

MD35590 $MA_PARAMSET_CHANGE_ENABLE = 0 または 1

パラメータセットは、ギヤ選択 + 1 に従って選択されます。

現在のギヤ選択は、以下に入っています。

DB31、... DBX16.0-16.2 (実際のギヤ選択 A ～ C)

有効なパラメータセットは、次のインタフェース信号で出力されます。

DB31、... DBX69.0-69.2 (制御パラメータセット A ～ C)

5 つのギヤ選択毎に、以下の割り当てを持つパラメータセットが 1 つ、NC によって提供されます。

データセット	NC/PLC インタフェース DBX 69.2 / 69.1 / 69.0	パラメータセット番号	パラメータセットインデックス[n]
軸モード	最後に有効だったギヤ選択	1	0
ギヤ選択 1	001	2	1
ギヤ選択 2	010	3	2
ギヤ選択 3	011	4	3
ギヤ選択 4	100	5	4
ギヤ選択 5	101 110 111	6	5

軸モードの主軸

主軸が軸モードの場合は、パラメータセットインデックス「0」がサーボで選択されます (MD35590 $MA_PARAMSET_CHANGE_ENABLE に注意してください)。

ギヤ選択切替動作は、以下のマシンデータの設定により異なります。

MD35014 $MA_GEAR_STEP_USED_IN_AXISMODE (M70 での軸モードのギヤ選択)

軸モードにギヤ選択を設定していない(MD35014 = 0)場合は、M70 で自動ギヤ選択切替がおこなわれません(初期設定)。最後のギヤ選択は内部的に保存され、次の主軸プログラミング時に、関連するパラメータセットとともに再度有効になります。

ただし、軸モードにギヤ選択を指定している(MD35014 = 1 ～ 5)場合は、M70 の実行時に、ギヤ 1 ～ 5 へのギヤ選択切替がおこなわれます。軸モードから主軸モードへの切り替え時には、M70 で読み込まれたギヤ選択は有効なままです。M70 の前に主軸モードで有効になるギヤ選択は、自動的には再度読み込まれません。

「M70 の設定可能なギヤ選択 (ページ 1639)」も参照してください。

負荷側ギヤボックス伝達比

ギヤ選択毎に軸モードで、正または負の**負荷側ギヤボックス係数**を設定することができます。

この設定は、次のマシンデータを介して分子と分母で別々におこなわれます。

MD31050 $MA_DRIVE_AX_RATIO_DENOM[n] (分母負荷側ギヤボックス)

MD31060 $MA_DRIVE_AX_RATIO_DENOM[n] (分子負荷側ギヤボックス)

正と負の負荷側ギヤボックス係数で、同じ設定範囲です。

値「0」を入力することはできません。

注記

間接エンコーダが使用され、負荷側ギヤボックス伝達比が変化する場合、関連する検出器について、原点が失われ、NC/PLC インタフェース信号
DB31、... DBX60.4/60.5 (原点確立済み/原点同期済み 1 または 2)
がリセットされます。

参照先

制御とサーボパラメータセットの詳細については、以下を参照してください。

- 『機能マニュアル 基本機能』;「速度、位置制御ループ(G2)」
- プログラミングマニュアル、上級編; :「プログラム可能なサーボパラメータセット」の章

17.4.5 中間ギヤ

用途と機能

設定済の中間ギヤを使用して、さまざまな回転工具に適応できます。工具側にある中間ギヤは、モータ/負荷側ギヤボックスに相乗的影響を与えます。

中間ギヤは、次のマシンデータを介して設定します。

MD31066 $MA_DRIVE_AX_RATIO2_NUMERA (中間ギヤ分子)

MD31064 $MA_DRIVE_AX_RATIO2_DENOM (中間ギヤ分母)

中間ギヤ用の工具上のエンコーダは、
マシンデータ:
MD31044 $MA_ENC_IS_DIRECT2 (中間ギヤ上のエンコーダ)
で設定します。

このマシンデータで変更されたパラメータ設定は、SinuCOM-NC 試運転ソフトウェアまたは操作パネル(HMI)のソフトキーのいずれかによって、「マシンデータの起動」ファン

クションとともに起動できます。一方、既存のモータ/負荷側ギヤボックスは、POWER ON 後に有効になります。

工具交換

中間ギヤを工具と同時に交換した場合、ユーザーは、中間ギヤのマシンデータを介して分子と分母の伝達比も再設定しなければなりません。

例:

⚠ 注意

工学的な異常

必要に応じてマシンデータに変更を加えてから「マシンデータの起動」ファンクションを有効にするために、適切な期間内に停止することは、引き続きユーザーの処理です。

取り付けた工具の伝達比が 2:1 である場合、NEWCONF 命令を使用してパートプログラム内で適切な中間ギヤが設定され、即座に有効にされます。

プログラムコード

```
N05 $MA_DRIVE_AX_RATIO2-NUMERA[AX5] = 2
M10 $MA_DRIVE_AX_RATIO2-DENOM[AX5] = 1
N15 NEWCONF
```

切替

新しい伝達比への切り替えは、「マシンデータの起動」ファンクションによって直ちに実行されます。技術的な観点から見ると、関連する機械的な切替処理には、ある程度時間がかかります。これは、機械的には、回転工具をともなう別の中間ギヤが取り付けられるためです。

注記

停止状態での切替には衝撃が発生しません。したがって、ユーザーには適切な予防措置を取る責任があります。

動作中に切替が行われるアプリケーションや、平滑化された、つまり、滑らかな速度遷移を必要とするアプリケーションは、既存の指令速度フィルタを使用して処理できます。

制御工学との依存関係についての詳細は、「G2: 速度、指令値/フィードバック回路、閉ループ制御 (ページ 405)」の章を参照してください。

17.4.6 応答のないギヤ選択切替

モード変更

応答されていないギヤ選択切替は、運転モードの変更(たとえば、JOG への切り替え)によって中断することはできません。

切り替えは、マシンデータ:
MD10192 $MN_GEAR_CHANGE_WAIT_TIME
に入力されている最大期間だけ遅延されます。

ギヤ選択切替がこの期間内に応答されなかった場合、NC はアラームを出力します。

その他のイベント

再解析を開始するイベントも、ギヤ選択切替が完了するまで待機します。

マシンデータ:
MD10192 $MN_GEAR_CHANGE_WAIT_TIME
に入力されている時間によって、ギヤ選択切替を実行するまでにコントローラが待機する時間が決定されます。
ギヤ選択切替が完了せずに、この時間が経過した場合、NC はアラームを出力します。

次のイベントは同様の動作をおこないます。

- ユーザー ASUB
- モード変更
- 残移動距離削除
- 軸入れ替え
- PI ユーザーデータの起動
- PI サービスマシンデータの設定
- ブロックスキップの切り替え、ドライランの切り替え
- モードでの編集
- 補正ブロックアラーム
- オーバストア
- G33、G34、G35 を使用した高速リトラクト
- サブプログラムレベル中止、サブプログラム中止

POWER ON 後の動作

機械上の有効なギヤ選択は、NC の POWER ON 後とリセット状態で PLC によって指定できます。

NC は適切なパラメータセットを選択し、
NC/PLC インタフェース信号
DB31、... DBX82.0 ～ 82.2 (ギヤ選択設定 A ～ C)
を確認し PLC へ返します。

17.4.7 揺動モードでのギヤ選択切替

新しいギヤ選択が M40 (自動ギヤ選択)または M41 ～ M45 を使用して定義された場合、主軸は揺動モードに入ります。

揺動時には、主軸モータの回転方向が短い間隔で継続的に反転します。このモータの揺動動作により、たとえば新しいギヤ選択がかみ合いやすくなります。ほとんどの場合、新しいギヤ選択は、揺動なしで使用することもできます。

次の揺動タイプが可能です。

- NC による揺動制御
- PLC による揺動
- FC 18 による揺動
  参照先:
  『機能マニュアル 基本機能』;「PLC 基本プログラム(P3)」

試運転:マシンデータ

- MD35010 $MA_GEAR_STEP_CHANGE_ENABLE (ギヤ選択切替へのパラメータ割り当て)
- MD35410 $MA_SPIND_OSCILL_ACCEL (揺動時の加速度)
- MD35430 $MA_SPIND_OSCILL_START_DIR (揺動時の開始方向)
- MD35400 $MA_SPIND_OSCILL_DES_VELO (揺動速度)
- MD35450 $MA_SPIND_OSCILL_TIME_CCW (M4 方向用の揺動時間)
- MD35440 $MA_SPIND_OSCILL_TIME_CW (M3 方向用の揺動時間)

試運転:NC/PLC インタフェース信号

- DB31、... DBX16.0 - 2 (現在のギア選択)
- DB31、... DBX16.3 (ギヤ選択が変更されました)
- DB31、... DBX18.4 (PLC による揺動制御)
- DB31、... DBX18.5 (揺動イネーブル)
- DB31、... DBX18.6 (回転方向指令、右回り)
- DB31、... DBX18.7 (回転方向指令、左回り)
- DB31、... DBX61.4 (主軸停止)
- DB31、... DBX82.0 - 2 (ギヤ選択指令)
- DB31、... DBX82.3 (ギヤ選択の切り替え)
- DB31、... DBX83.5 (主軸指令一致)
- DB31、... DBX84.6 (有効な主軸モード:揺動モード)

シーケンスの説明:ギヤ選択切替

パラメータ設定:MD35010 $MA_GEAR_STEP_CHANGE_ENABLE、ビット 0 = 1

シーケンス:

- 主軸の減速。
  減速動作は M5 動作に相当します。
- NC/PLC インタフェース信号 NC → PLC:
  - DB31、... DBX84.6 == 1 (揺動モード)
  - DB31、... DBX82.3 == 1 (ギヤ選択切替)
  - DB31、... DBX82.0 - 2 == <ギヤ選択の設定>
  - DB31、... DBX61.5 == 0 (位置コントローラ有効)
- 負荷側ギヤを「引き離す」ことができるようになります。
- 揺動は現在、NC または PLC から実行できます(以下を参照してください)。
- ギヤ選択が切り替えられ、新しいギヤ選択が有効になったことを通知する PLC から へ NC へのフィードバック信号の後、プログラムの実行は続行します。
- DB31、... DBX16.3 (ギヤ選択が変更されました)

- DB31、... DBX16.0 - 2 (現在のギア選択)
- 間接エンコーダ(モータエンコーダ)付きの検出器では、原点確立状態が削除されます: DB31, ... DBX60.4 / 5 = 0

NC による揺動制御

必要条件

- DB31、... DBX18.5 = 1 (揺動イネーブル)

状態フィードバック信号

- DB31、... DBX84.6 == 1 (揺動モード)
- DB31、... DBX82.3 == 1 (ギヤ選択切替)
- DB31、... DBX82.0 - 2 == <ギヤ選択の設定>
- DB31、... DBX61.5 == 0 (位置コントローラ有効)

負荷側ギヤを「引き離す」ことができるようになります。

揺動の開始

PLC ユーザープログラムは、揺動イネーブル信号を設定できるようになりました。

- DB31、... DBX18.5 = 1 (揺動イネーブル)

モータは、回転方向に依存する揺動時間、開始方向に揺動速度まで揺動加速度で加速します。

- MD35410 $MA_SPIND_OSCILL_ACCEL (揺動時の加速度)
- MD35430 $MA_SPIND_OSCILL_START_DIR (揺動時の開始方向)
- MD35400 $MA_SPIND_OSCILL_DES_VELO (揺動速度)
- MD35450 $MA_SPIND_OSCILL_TIME_CCW (M4 方向用の揺動時間)
- MD35440 $MA_SPIND_OSCILL_TIME_CW (M3 方向用の揺動時間)

回転方向の反転

方向依存の揺動時間の経過後、モータは揺動減速で制動され、残る揺動時間、揺動速度まで逆回転方向に加速されます。

PLC による揺動制御

必要条件

- DB31、... DBX18.4 = 1 (PLC による揺動制御)
- DB31、... DBX18.5 = 1 (揺動イネーブル)

状態フィードバック信号

- DB31、... DBX84.6 == 1 (揺動モード)
- DB31、... DBX82.3 == 1 (ギヤ選択切替)
- DB31、... DBX82.0 - 2 == <ギヤ選択の設定>
- DB31、... DBX61.5 == 0 (位置コントローラ有効)

負荷側ギヤを「引き離す」ことができるようになります。

揺動の開始

PLC ユーザープログラムは、揺動イネーブル信号を設定できるようになりました。

- DB31、... DBX18.7 = 1 (回転方向指令、左回り)または DB31、... DBX18.6 = 1 (回転方向指令、右回り)

モータは、指定した回転方向指令に従って、揺動速度まで揺動加速度で加速します。

- MD35410 $MA_SPIND_OSCILL_ACCEL (揺動時の加速度)
- MD35400 $MA_SPIND_OSCILL_DES_VELO (揺動速度)

回転方向の反転

揺動時間および回転方向の切り替えは、PLC ユーザープログラムで実現する必要があります。

揺動モードの終了

揺動モードの終了時に、主軸は開ループ制御モードに戻り、SPCON または SPCOF によって定義されているモードに自動的に切り替わります。

すべてのギヤ別の制限値(最低/最高速度など)は、実際のギヤ選択のパラメータ設定された値に対応します。

ブロック切り替え

主軸を揺動モードに切り替えた場合(DB31、... DBX82.3 = 1 (ギヤ選択切り替え))、有効な NC プログラムの処理が停止します。

ギヤ選択が切り替え済みである場合(DB31、... DBX16.3 == 1 (ギヤ選択切り替え済み))、NC プログラムの処理が続行します。

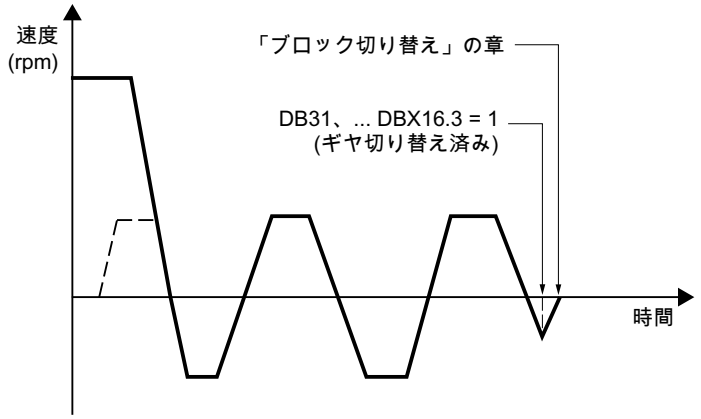
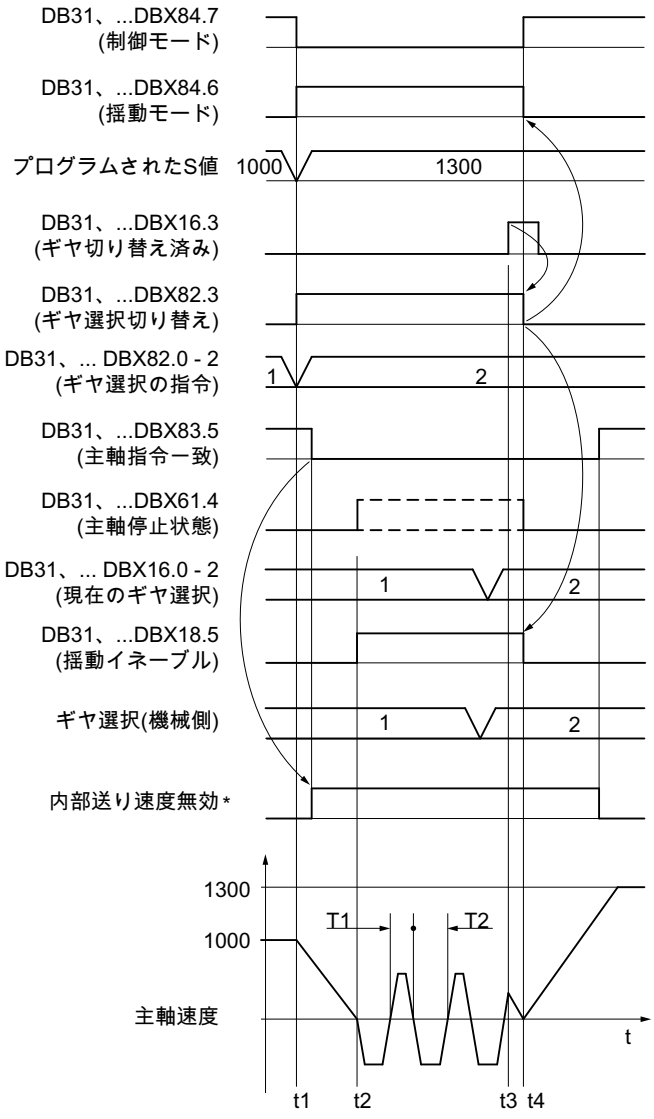


図 17-5　揺動モードに続くブロック切り替え

信号フローの例



t_1: NC プログラムで新たにプログラムした **S 値 S1300** により、NC はギヤ選択切り替え(第 1 → 第 2 ギア選択)の必要性を識別し、PLC からイネーブルを要求します。

- NC → PLC:DB31、... DBX82.3 = 1 (ギヤ選択切り替え)

NC はプログラム処理をロックします (内部送り速度の無効化*)

t_2: NC は、**主軸速度 == 0** を PLC に通知します。

- DB31,... DBX61.4 = 1 (主軸停止状態)

PLC ユーザープログラムは、揺動モードを有効にします。

- PLC → NC:DB31、... DBX18.5 = 1 (揺動イネーブル)

  注

  揺動イネーブルは、遅くとも時間 t_2 に設定する必要があります。

NC は揺動を開始します。

t_3: 機械のギヤ選択が切り替えられました。

PLC ユーザープログラムは、NC への切り替えを通知し、揺動モードイネーブルを削除します。

- PLC → NC:DB31、... DBX16.0 - 2 = 2 (現在のギア選択)
- PLC → NC:DB31、... DBX16.3 = 1 (ギヤ選択が変更されました)
- PLC → NC:DB31、... DBX18.5 = 0 (揺動イネーブル)

t_4: NC は、揺動モードから制御モードに主軸モードを切り替えることで、PLC および揺動に対する要求をリセットすることにより、ギヤ選択の切り替えを終了します。

- NC → PLC:DB31、... DBX82.3 = 0 (ギヤ選択切り替え)
- DB31、... DBX84.6 = 0 (有効な主軸モード:揺動モード)
- DB31、... DBX84.7 = 1 (有効な主軸モード:制御モード)

NC は、次のブロックの加工を有効にし、新しい S 値 S1300 まで主軸を加速します。

*: 次の場合は、内部送り無効が設定されます。

- NC プログラムの S 値でプログラミングした結果としてギヤ選択が切り替え、**および**
- **加工**ブロックが有効(早送り G0 ではない)

注

内部送り無効は、シンクロナイズドアクションからのギヤ選択切替時も、PLC での FC 18 の指定によるのギヤ選択切替時も、設定**されません**。

図 17-6　主軸停止でのギヤ選択切替

必要条件

- 主軸を制動するために NC/PLC インタフェース信号を設定する必要はありません。
  DB31、... DBX4.3 (主軸停止)
  ギヤ選択切替が要求されると、主軸はコントロールシステムで停止状態になります。
- NC/PLC インタフェース信号を設定することにより、ギヤ選択切り替えは常に終了する必要があります。
  DB31、... DBX16.3 (ギヤ選択が変更されました)
- NC/PLC インタフェース信号のリセット時に、NC で揺動動作が停止します。
  DB31、... DBX18.5 (揺動イネーブル)
  ただし、主軸は「揺動モード」モードのままです。
- ギヤ選択の切り替えおよび間接検出器により、主軸の同期が失われます(MD31050 $MA_ENC_IS_DIRECT = 0 モータエンコーダ)。
  ギヤ選択切り替え ⇒ DB31、... DBX60.4 / 5 = 0 (原点確立済み/原点同期済み)
  ギヤ選択切り替え後、次に原点マークが渡されるときに、主軸が自動的に再同期されます。

17.4.8 固定位置でのギヤ選択切替

用途と利点

工作機械では、標準化された主軸ドライブの使用が増えていますが、これは、第一にギヤ選択切替に関する技術的な無駄時間を節約するためであり、第二に標準化されたコンポーネントを使用するコスト面の利点を得るためです。

固定位置でのギヤ選択切替機能は、NC とは異なる方法で有効にする必要がある負荷側ギヤボックスの「方向付けられたギヤ選択切替」をサポートしています。この場合、ギヤ選択切替は、定義された主軸位置でのみおこなうことができます。したがって、従来の負荷側ギヤボックスで必要とされる揺動動作は不要になります。

固定位置でのギヤ選択切替の処理

固定位置でのギヤ選択切替

マシンデータ設定:
MD35010 $MA_GEAR_STEP_CHANGE_ENABLE = 2
では次の処理を実行します。

- 停止状態、または移動状態からのマシンデータ:
  MD35012 $MA_GEAR_STEP_CHANGE_POSITION で設定した位置への主軸の位置決め。
  ギヤ選択切替が移動状態からおこなわれた場合、現在の回転方向が維持されます。主軸は、位置決め動作中に位置決めモードに入ります。
  NC/PLC IS:
  DB31、... DBX84.5 (位置決めモード)
  が出力されます。
  有効なレファレンス点がない場合:
  DB31、... DBX60.4/5 = 0
  または NC/PLC IS:
  DB31、... DBX17.4/5 (位置決めでの再同期 MS 1/2)
  が設定されている場合、位置決め動作は、原点マークが見つかるまでの時間だけ伸ばされます。

- マシンデータ:
  MD35012$MA_GEAR_STEP_CHANGE_POSITION
  で設定されたギヤ選択切替位置に到達後、機械はマシンデータ:
  MD35310 $MA_SPIND_POSIT_DELAY_TIME
  で設定された時間だけ待機してから揺動モードに切り替わり、
  既知のギヤ選択切替の交信が開始されます。

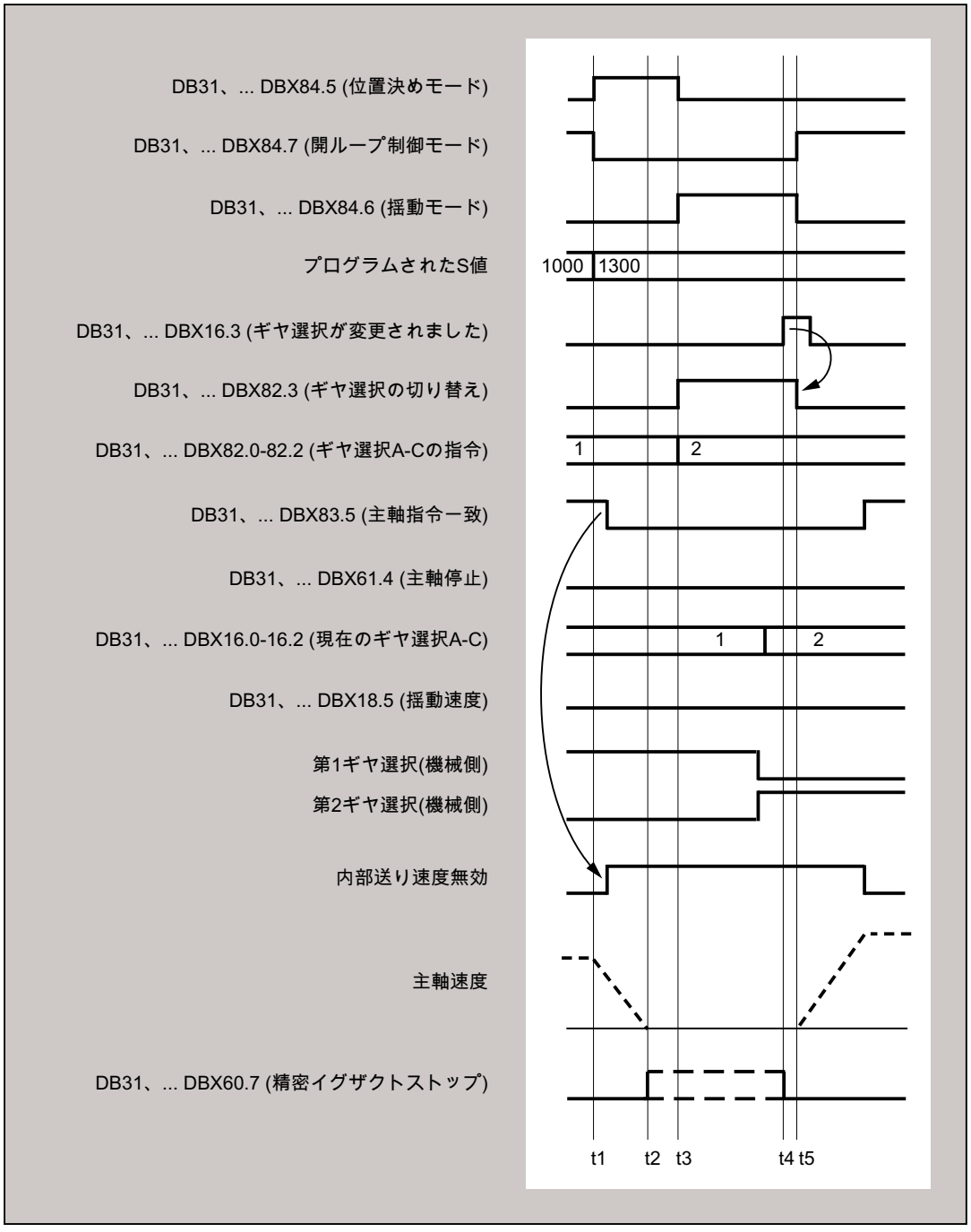
- NC/PLC インタフェース信号の出力:
  DB31、... DBX84.6 (揺動モード)
  DB31、... DBX82.3 (ギヤ切替)
  DB31、... DBX82.0-82.2 (ギヤ選択指令 A ～ C)

- 間接エンコーダ(モータエンコーダ)を使用した有効な検出器が使用されるときに、位置制御は解除されません。
  MD31040 $MA_ENC_IS_DIRECT = 0
  もし、直接エンコーダ（負荷エンコーダ）を使用した検出システムが有効な場合、位置制御は、無効になります。
  DB31、... DBX61.5 = 0
  これは、負荷への動力伝達が中断されて、閉ループ位置制御ができなくなるためです。

- 位置制御運転が実行不可能な場合は、
  「コントローライネーブル」:
  DB31、... DBX2.1 = 0
  をリセットしてそれをキャンセルすることができます。
- 機械でのギヤ選択の機械的な切替
  ドライブから揺動動作が要求されることはありません。
  NC/PLC IS:
  DB31、... DBX18.5 (揺動有効)
  と
  DB31、... DBX18.4 (PLC からの揺動)
  は、設定**しない**でください。
  原則として、揺動動作はこの時点ではまだ可能です。
- NC/PLC IS:
  DB31、... DBX16.0-16.2 (実際のギヤ選択 A ～ C)
  の PLC による書き込み
- 信号:
  DB31、... DBX16.3 (ギヤ選択切替済)後に、
  有効になる前回の移動がある場合は、それが継続されます。
  間接エンコーダ(モータエンコーダ)の場合、原点確立状態はクリアされます。
  DB31、... DBX60.4/5 = 0
  主軸は速度制御モードにあり、NC/PLC IS:
  DB31、... DBX84.7 (開ループ制御モード)
  が出力されます。

固定位置での GSC

固定位置でのギヤ選択切替の典型的なタイミング図は下図のとおりです。



t_1: S1300 のプログラミングにより、NC が新しいギヤ選択(第 2 ギヤ選択)を検出し、IS DB31、... DBX84.5 (位置決めモード)を設定し、次のパートプログラムブロックの処理をブロックします(= 内部送り無効*)。

t_2: 主軸が静止し、イグザクトストップの信号が発行されます。

t_3: ギヤ選択切替 - 待機時間

t_4: 新しいギヤ選択がかみ合います。PLC ユーザーが新しい(実際の)ギヤ選択を NC に送信し、IS DB31、... DBX16.3 (ギア切替済み)を設定します。

t_5: NC は、IS DB31、... DBX82.3 (ギヤ切替)を解除し、次のパートプログラムブロック処理を許可して、新しい S 値(S1300)へ主軸を加速します。

* : 次の場合は、内部送り無効が設定されます。

- 主軸ギヤ選択切替がパートプログラムでプログラム指令されている**および**
- 処理ブロックが動作中である(つまり、G0 が動作中でない)

内部送り無効は、シンクロナイズドアクションからのギヤ選択切替時も、PLC での FC 18 の指定によるのギヤ選択切替時も、設定**されません**。

図 17-7 主軸停止でのギヤ選択切替

ギヤ選択切替位置 MD35012

ギヤ選択切替位置は、マシンデータ:
MD35012 $MA_GEAR_STEP_CHANGE_POSITION
でギヤ選択毎に定義されます。

ギヤ選択切替待機時間 MD35310

位置決め動作の後、機械はヤ切替要求:
DB31、... DBX84.6 (揺動モード)
DB31、... DBX82.3 (ギア切替)
および
DB31、... DBX82.0-82.2 (ギヤ選択指令 A ～ C)
が出力されるまで、マシンデータ:
MD35310 $MA_SPIND_POSIT_DELAY_TIME
で設定された時間の間、待機します。

位置決め特性/位置

位置決めは常に最短距離(DC に対応)でアプローチします。

使用可能なレファレンス点がなく、主軸が停止している場合
(たとえば、電源投入後など)、移動方向は、下記のマシンデータによって決定されます。
MD35350 $MA_SPIND_POSITIONING_DIR

ギヤ選択切替位置の調整が必要な場合、マシンデータを書き込み続いて「マシンデータの起動」を実行することで実現できます。
MD 値の変更は、パートプログラムまたは HMI によって実行できます。

システムがプリセットされた位置に到達できない場合、アラーム 22020 が出力され、ギヤを損傷しないために、NC と PLC の間のギヤ選択切替の交信はおこなわれません。このアラームは重大なため、パートプログラムは続行できず、あらゆる状況下で原因を排除する必要があります。経験的には、位置決めの中止は通常、MD 設定が間違っているか PLC 信号に互換性がないことが原因です。

速度

位置決め速度は、ギヤ選択に応じて設定された次のマシンデータの値が使用されます。

MD35300 $MA_SPIND_POSCTRL_VELO

NC/PLC インタフェース信号「主軸オーバーライド」/「送り速度オーバーライド」、
DB31、... DBX17.0=0:DB31、... DBB19
および
DB31、... DBX17.0=1:DB31、... DBB0
は、通常どおり位置決めに対して有効です。
位置決め速度は、プログラム命令 OVRA[Sn]に比例して変更できます。

注記

OVRA[Sn]はモーダルで有効です。ギヤ選択切替後、加工に適した値を再設定する必要があります。

パートプログラム命令 FA[Sn]では、ギヤ選択切り替え中の位置決め速度は変更されません。

加減速

加速度は、ギヤ選択に応じて設定された次のマシンデータによって決定されます。

MD35200 $MA_GEAR_STEP_SPEEDCTRL_ACCEL

および

MD35210 $MA_GEAR_STEP_POSCTRL_ACCEL

加速度は、ACC[Sn]のプログラム指令に比例して変更できます。

注記

ACC[Sn]はモーダルで有効です。ギヤ選択切替後、加工に適した値を再設定する必要があります。

速度に依存する加減速

「膝形加減速特性」は、SPOS または FC18 での位置決めの場合と同様に有効です。

加々速度

現在、加減速の変化を制限することはできません。

位置決めの終了

位置決め動作の終了(DB31、... DBX84.5)
と揺動モードの開始(DB31、... DBX84.6)の遷移は、
"精密イグザクトストップ"への到達 (DB31、... DB60.7)とマシンデータ:
MD3510 $MA_SPIND_POSIT_DELAY_TIME
で設定された時間で定義されます。

遷移条件の決定は、第一にギヤ選択切替時間に影響を与え、第二にプリセットされたギヤ選択切替位置へのアプローチの精度に影響を与えます。

ブロック切り替え

PLC (DB31、 ... DBX16.3)によってギヤ選択が切り替えられるまで、ブロック切り替えは停止され、加工ブロックは開始されません。

ギヤ選択切替の終了

ギヤ選択切替が終了すると、主軸が開ループ制御モードに戻って、SPCON または SPCOF で定義された閉ループ制御モードに自動的に変わります。

すべてのギヤ固有の制限値(ギヤ選択の最高/最低速度など)は、実際のギヤ選択のアンサーバック値に対応します。

必要条件

- 主軸には 1 つ以上のエンコーダが必要です。
- 位置制御運転を実行可能にし、かつ有効にしてください。
- 一般に SPOS 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から、あるいは、FC18:「主軸位置決めの開始」を介して、異常なく実行できるようにする必要があります。

すべての必要条件を満たすことができなければ、説明した機能を正常に使用することはできません。

起動

固定位置でのギヤ選択切替の機能は、
設定:
MD35010 $MA_GEAR_STEP_CHANGE_ENABLE = 2
によって有効になります。

17.4.9 M70 の設定可能なギヤ選択

技術的バックグラウンド

機械によっては、軸モード時に主軸が特定のギヤ選択を使用する必要があります。

考えられる理由:

- 軸モードのサーボパラメータセットにある、ギヤ選択に適合する調整(K_v、フィードフォワード制御、フィルタ)は、1 つのみです(インデックス 0)。このパラメータセット用のマシンデータの設定は変更しないでください。
- 他の場合と異なり、バックラッシ補正のほとんどない機械的ギヤ比は 1 つのみです。主軸は、このギヤ選択のその他の軸と共に、軌跡移動または座標変換(TRANSMIT など)に従うだけです。

機能

この機能が起動した場合は、軸モードへの遷移中に、事前定義のギヤ選択が自動的に読み込まれます。

ギヤ選択切替は M70 処理に統合され、主軸の減速後、およびインデックス 0 (MD35590 $MA_PARAMSET_CHANGE_ENABLE に注意してください)のサーボパラメータセットの読み込み前に行われます

ギヤ選択切替時の NC と PLC 間での対話は、プログラムしたギヤ選択切替(M41 ... M45)と同様の方法で実行されるのが普通です。

必要条件

軸モードへの遷移中にギヤ選択切替を行うには、次のマシンデータで、ギヤ選択切替を全般的に有効にする必要があります。

MD35010 $MA_GEAR_STEP_CHANGE_ENABLE (ギヤ選択切替へのパラメータ割り当て)。

MD35090 $MA_NUM_GEAR_STEPS (設定したギヤ選択の数)

起動と解除

この機能は、以下のマシンデータで起動/解除します。

MD35014 $MA_GEAR_STEP_USED_IN_AXISMODE (M70 での軸モードのギヤ選択)

値	意味
0	M70 では自動ギヤ選択切替が行われません。現在のギヤ選択が保持されます(初期設定)。
1 ... 5	ギヤ選択 1 ... 5 へのギヤ選択切替が、M70 の処理中に行われます。

必要条件

固定位置でのギヤ選択切替(MD35010 $MA_GEAR_STEP_CHANGE_ENABLE = 2)

「固定位置でのギヤ選択切替」機能がサポートされています。この機能は、M70 の手順を、主軸の位置決めの所要時間によって拡張します。位置には、現在のギヤ選択で移動します。

M70 のプログラミングによらない軸モードへの遷移

コントロールシステムはプログラム処理から自動的に、軸モードへの遷移を検出し(「軸モードへの自動遷移 (ページ 1585)」を参照してください)、コントロールシステム内で、ギヤ選択切替を含む必要な M70 処理を生成します。

FC 18 による軸モードへの遷移

自動ギヤ選択切替は、FC 18 (「軸起動」)による軸モードへの遷移ではサポートされません。これを行うには、軸モードに切り替える前に、適切なギヤ選択を PLC アプリケーションで使用する必要があります。ギヤ選択切替は、FC 18 (「ギヤ選択切替の起動」)で行うこともできます。

軸モードから主軸モードへの切り替え

軸モードから主軸モードへの切り替え時には、M70 で読み込まれたギヤ選択は有効なままです。M70 の前に主軸モードで有効になるギヤ選択は、自動的には再度読み込まれません。サーボパラメータセットは、ギヤ選択
(MD35590 $MA_PARAMSET_CHANGE_ENABLE < 2 による)に合わせて、パラメータセット 1 (インデックス 0)からパラメータセット 2 ... 6 (インデックス 1 ... 5)に変更されます。

例

軸モードへの主軸遷移の場合は、ギヤ選択 4 を読み込んでください。

設定:MD35014 $MA_GEAR_STEP_USED_IN_AXISMODE[<主軸名称>] = 4

プログラムコード	コメント
N05 M3 S1000	
N10 G1 X100 F1000	
N15 M70	; ギヤ選択 4 を読み込みます。
N20 POS[C]=77	
N25 ...	

注記

MD35014 は、「マシンデータ起動」ファンクションによって変更できます。したがって、読み込んだギヤ選択を、必要に応じて、軸モードへの遷移前にパートプログラムで切り替えることができます。

17.4.10 ドライラン、プログラムテスト、および SERUPRO に対するギヤ選択切替のマスク

機能

テスト送り速度(ドライラン)、プログラムテスト、および SERUPRO の場合は通常、ギヤ選択切替は必要ありません。そのため、これらの機能に対してギヤ選択切替をマスクすることができます。対応する設定は、次のマシンデータのビット 0 ... 2 によって行われます。

MD35035 $MA_SPIND_FUNCTION_MASK

ドライラン送り速度(ドライラン)	
ビット 0 = 0	ドライラン機能が有効の場合 - パートプログラムブロックでは - ギヤ選択を M40、M41 ～ M45、または FC18 によるプログラミングとシンクロナイズドアクションで切り替えることができます。
ビット 0 = 1	ドライラン機能が有効の場合 - パートプログラムブロックでは - ギヤ選択切替を M40、M41 ～ M45、または FC18 によるプログラミングとシンクロナイズドアクションでマスクすることができます。
プログラムテストおよび SERUPRO	
ビット 1 = 0	プログラムテスト/SERUPRO 機能が有効の場合 - パートプログラムブロックでは - ギヤ選択を M40、M41 ～ M45、または FC18 によるプログラミングとシンクロナイズドアクションで切り替えることができます。
ビット 1 = 1	プログラムテスト/SERUPRO 機能が有効の場合 - パートプログラムブロックでは - ギヤ選択切替を M40、M41 ～ M45、または FC18 によるプログラミングとシンクロナイズドアクションでマスクすることができます。
ドライラン、プログラムテストおよび SERUPRO	
ビット 2 = 0	プログラムされたギヤ選択のギヤ選択切替は、ドライラン、プログラムテストおよび SERUPRO の機能の解除後の REPOS 時には、続けて実行**されません**。
ビット 2 = 1	プログラムされたギヤ選択のギヤ選択切替は、可能な場合、ドライランおよび SERUPRO の機能の選択後に実行されます。

手順

ギヤ選択切替がマスクされた場合、必要に応じて、補間器が、プログラムされた主軸速度を有効なギヤ選択の許容速度範囲に制限します。

この制限の結果として発生する NC/PLC インタフェース信号 DB31、... DBX83.2 (指令速度増加)と DB31、... DBX83.1 (指令速度制限中)がマスクされます。

PLC プログラムによる監視は、ドライラン中およびドライラン送り速度では不要です。

ギヤ選択切替がマスクされている場合、新しいギヤ選択指令(DB31、...DBX82.0-82.2)は PLC に出力されません。

ギヤ選択切替要求 DB31、... DBX82.3 (ギヤ切替)もマスクされます。

これにより、ギヤ選択切替情報が PLC プログラムによって処理される必要がなくなります。

最後に有効だったギヤ選択の特定

システム変数$P_GEAR は、パートプログラムでプログラムされたギヤ選択を返します(このギヤ選択は PLC に出力されていない場合があります)。

システム変数$AC_SGEAR を使用して、パートプログラム、シンクロナイズドアクション、ユーザーインタフェースから、最後に有効だったギヤ選択を読み取ることができます。

解除後の動作

ドライラン機能は、実行中のパートプログラム内で解除できます。この機能が解除されると、パートプログラムによって要求された正しいギヤ選択が識別および選択されなければなりません。

正しいギヤ選択が有効にされるまで、パートプログラムの残りが異常なしで実行されることは保証できません。主軸が速度制御モードにある場合、解除時に起動されるシステム REPOS で必要なギヤ選択切替が実行されます。PLC により、完全なギヤ選択切替の対話が行われ、プログラムされた最後のギヤ選択が有効になります。

REPOS では、パートプログラムでプログラムされたギヤ選択と、NC/PLC インタフェース経由で提供された実際のギヤ選択が一致しない場合、ギヤ選択切替は行われません。

同じことが SERUPRO 機能にも当てはまります。

ブロックサーチ機能 SERUPRO についての詳細は、以下を参照してください。
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、プログラムモード、リセット応答(K1)」

必要条件

ギヤ選択切替がマスクされた場合、出力された主軸速度は、現在のギヤ選択によって指定された速度範囲内で移動します。

この後の REPOS によるギヤ選択切替の起動には、次の制限事項が適用されます。

- 解除ブロックまたはターゲットブロック内の主軸がコマンド主軸(シンクロナイズドアクション)または PLC 主軸(FC18)である場合、ギヤ選択切替はそれ以降有効になりません。
- 主軸が位置決めモードまたは軸モードにあるか、リンクが有効であるために、ギヤ選択を有効にできなかった場合は、アラーム 22011「チャネル%1 ブロック%3 主軸 2% プログラムされたギヤ選択への変更不可」が通知されます。

例

ドライランでのギヤ選択切替

```
; 出力ステータスについて 1 番目のギヤ選択有効
N00 M3 S1000 M41                    ; 1 番目のギヤ選択を有効にします。
M0                                  ; パートプログラムを停止します

; PI service:Activate dry run feedrate (DryRun)
                                    ; (設定)
N10 M42                             ; 2 番目の GS が要求される。ギヤ選択切替は実行されない
N11 G0 X0 Y0 Z0                     ; 軸位置決め
N12 M0                              ; パートプログラムを停止します

; PI service:Deactivate dry run feedrate (DryRun)
                                    ; REORG および REPOS が実行されます
                                    ; 次に 2 番目のギヤ選択へのギヤ選択切替が実行されます
N20 G1 Z100 F1000
...
N99 M30                             ; パートプログラムの終了
```

17.5 設定可能な主軸機能の追加の調整

次の主軸ファンクションは、マシンデータを使用して起動できます。

MD35035 $MA_SPIND_FUNCTION_MASK, <ビット> = <値>

<ビット>	<値>	意味
0 ... 2	1	適用タイミング:テスト送り速度(ドライラン)、プログラムテスト、および SERUPRO に対するギヤ選択切替の動作 「ドライラン、プログラムテスト、および SERUPRO に対するギヤ選択切替のマスク (ページ 1642)」を参照してください。
4	1	NC プログラムでプログラムされた主軸速度 S...は、SD43200 $SA_SPIND_S の後に取得されます。これはまた、FC18 およびシンクロナイズドアクションによる PLC ユーザープログラムでの速度指令値信号にも該当します。 「機能 (ページ 1657)」を参照してください。
5	0	ジョグモードで主軸を移動する場合、現在有効な速度指令値が適用されます。 SD41200 $SN_JOGSPIND_SET_VELO を参照してください。
	1	ジョグモードで主軸を移動する場合、SD43200 $SA_SPIND_S でパラメータ設定された速度が有効になります。
8	1	NC プログラムでプログラムされた切削速度 S...は、SD43202 $SA_SPIND_CONSTCUT_S の後に取得されます。 これはまた、FC18 による PLC ユーザープログラムでの切削速度設定信号にも該当します。 「機能 (ページ 1657)」を参照してください。
10	0	SD 43206 $SA_SPIND_SPEED_TYPE は、NC プログラムまたはチャネル設定によって変更されません。
	1	主軸がメイン主軸である場合、送り速度タイプ(15 番目の G グループの有効値)が SD43206 $SA_SPIND_SPEED_TYPE の後に取得されます。 主軸がメイン主軸でない場合、SD43206 $SA_SPIND_SPEED_TYPE は変更されません。 「機能 (ページ 1657)」を参照してください。
12	原点マーク検出に対する主軸オーバライドの有効性、M19、SPOS、SPOSA=0:	
	0	主軸オーバライドは有効では**ありません**。
	1	主軸オーバライドは有効です。

<ビット>	<値>	意味
19	自動補助機能出力 M19 に関する応答:	
	0	および MD20850 $MC_SPOS_TO_VDI == 0: SPOS および SPOSA では、補助機能 M19 は PLC に出力されません。
	1	SPOS および SPOSA では、補助機能 M19 は PLC に出力されます。 アドレス拡張は主軸番号に対応します。 「位置決めモード (ページ 1562)」を参照してください
20	自動補助機能出力 M70 に関する応答:	
	0	自動補助機能 M70 は PLC に出力されません。 注:プログラムされた補助機能 M70 は、常に PLC に出力されます。
	1	主軸から軸運転に移行する場合、補助機能 M70 は PLC に出力されます。 アドレス拡張は主軸番号に対応します。 「軸モードへの自動遷移 (ページ 1585)」を参照してください。
22	フローティングチェック(G331、G332)のないタッピングに対する主軸回転方向を反転するために、軸別 NC/PLC インタフェース信号 DB31、... DBX17.6 (M3/M4 反転)を解除します。	
	0	インタフェース信号 DB31, ... DBX17 が有効です。 注 ● 標準ケース:NC プログラムの起動時にインタフェース信号の値が決定されます。インタフェース信号が以降に変更された場合、NC プログラムでは適用されません。 ● 特記事項:NC プログラムの実行中にインタフェース信号に対する変更を考慮する必要がある場合、アプリケーションエンジニアリングの一環としてこれを実装する必要があります。「特記事項:NC プログラムの NC/PLC インタフェース信号による回転方向の反転 (ページ 1595)」の章を参照してください。
	1	インタフェース信号 DB31、... DBX17 が有効ではありません。

17.6 主軸再割付け

機能

チャネル主軸について、「切り替え可能な主軸」ファンクションでは、異なるチャネルで使用できる一般的な NC プログラムを書き込むことができます。NC プログラムで使用される論理主軸番号は、主軸再割り当てにより物理主軸番号に変換されます。これはつまり、NC プログラムでプログラムされた同じ主軸番号に対して、異なる物理主軸(機械軸)を異なるチャネルで移動できることを意味します。

「軸交換」によってロードまたはアンロードされる物理的主軸は、パートプログラム内で明記する必要がなくなりました。

マシンデータ

物理主軸番号

各主軸は、設定可能な一意の番号(物理主軸番号)を使用して、機械軸に割り当てる必要があります。

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[<機械軸>] = <物理主軸番号>

主軸再割り当ての起動

主軸再割り当ては、次のマシンデータを使用してチャネルごとに起動します。

MD20092 $MC_SPIND_ASSIGN_TAB_ENABLE = TRUE

セッティングデータ

主軸再割り付け

チャネル別に主軸再割り当てのパラメータを設定するために、各論理主軸番号またはプログラムされたアドレス拡張が物理主軸番号(機械軸)に割り当てられます。このアドレス拡張によって、主軸(チャネル主軸)を NC プログラムでアドレス指定します。

SD42800 $SC_SPIND_ASSIGN_TAB[<論理主軸番号>] = <物理主軸番号>

他の情報

- 現在のメイン主軸の論理主軸番号は、SD42800 $SC_SPIND_ASSIGN_TAB[0]から読み取ることができます。このデータに書き込むことはできません。このデータは、SETMS のプログラミング時に更新されます。
  使用されない主軸には、SD42800 内で値 0 が割り当てられます。
- 補助機能が出力される場合、物理主軸番号がアドレス拡張として出力されます。

- NC プログラムおよびシンクロナイズドアクションで主軸をプログラムする場合、主軸再割り当てが有効になります。
- ファンクションブロック FC18 を使用して NC/PLC インタフェースで主軸を入力する場合は、物理主軸が直接アドレス指定されるので、主軸再割り当ては有効に**なりません**。
- 主軸変換に対応するシステム変数は、$P_S、$P_SDIR、$P_SMODE、$P_GWPS、$AC_SDIR、§AC_SMODE、$AC_MSNUM、$AA_S です。

必要条件

- 切り替え可能なチャネル主軸は、軸交換機能の代替には**なりません**。
- 主軸変換は、NC プログラム、PLC ユーザープログラム、または PI サービスによってセッティングデータ(SD42800 $SC_SPIND_ASSIGN_TAB)に書き込むことによって変更できます。この変更は即時有効になります。
- 主軸再割り当てでは、チャネル(MD20070 $MC_AXCONF_MACHAX_USED)に割り当てられる物理主軸のみを使用できます。
  別のチャネルで現在有効である主軸再割り当てで物理主軸が指定されている場合、MD30552 $MA_AUTO_GET_TYPE の設定に応じて、チャネルに対して物理主軸が要求されるか、またはアラーム 16105「割り当て済みの主軸が存在しません」が表示されます。
- SD42800 $SC_SPIND_ASSIGN_TAB[<n>]が PLC または HMI から指定される場合、テーブルが変更されたチャネルがリセット状態になっているか、変更する主軸が実行中のパートプログラムで使用されていない状態でなければなりません。
  同期動作は、STOPRE プリプロセッサ停止によっておこなうことができます。
- 論理主軸から複数の物理主軸への変換はロックされません。しかし、論理的主軸が操作画面に表示されたときに、変換テーブルに対応する不明瞭さが現れます。

例

割り当て、主軸番号と機械軸

```
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX [AX4] = 1
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX [AX5] = 2
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX [AX6] = 3
MD35000 $MA_SPIND_ASSIGN_TO_MACHAX [AX7] = 5
```

チャネルへの機械軸の適用

```
MD20070 $MC_AXCONF_MACHAX_USED[0] = 4
MD20070 $MC_AXCONF_MACHAX_USED[1] = 5
MD20070 $MC_AXCONF_MACHAX_USED[2] = 6
MD20070 $MC_AXCONF_MACHAX_USED[3] = 7
```

メイン主軸の指定

```
MD20090 $MC_SPIND_DEF_MASTER_SPIND = 1
```

主軸再割り付け

```
MD20092                              ; 主軸再割り付けの機動
$MC_SPIND_ASSIGN_TAB_ENABLE=1
SD42800 $SC_SPIND_ASSIGN_TAB[0]      ; 設定されたメイン主軸
= 1
SD42800 $SC_SPIND_ASSIGN_TAB[1]      ; テーブルの初期設定
= 1
SD42800 $SC_SPIND_ASSIGN_TAB[2]
= 2
SD42800 $SC_SPIND_ASSIGN_TAB[3]
= 3
SD42800 $SC_SPIND_ASSIGN_TAB[4]      ; 論理主軸は割り当てられていません
= 0
; アドレス拡張は 1 です、出力:M1=3 S1=1000
; 主軸 1(物理メイン主軸)が回転します
M3 S1000
...
; 論理主軸 1 から物理主軸 5 への割り当て
SD42800 $SC_SPIND_ASSIGN_TAB[1]
= 5

; 論理主軸 2 から物理主軸 3 への割り当て
SD42800 $SC_SPIND_ASSIGN_TAB[2]
= 3
; 重要:これで、物理主軸 3 は 2 回割り当てられています。
論理主軸 2 と 3 をプログラムするときに、物理主軸 3 が
常にアドレス指定されます。
; 基本運転画面では、両方の主軸が回転します。

SETMS(2)                             ; SD42800 $SC_SPIND_ASSIGN_TAB[0] = 2
; メイン主軸 = アドレス拡張 = 2、変換された主軸番号 M3 = 5 が出力されます
; 番号「3」で設定された物理主軸が停止します。
M5
```

```
GET (S4)                    ; アラーム 16105:論理主軸 4
                            ; 変換することはできません
RELEASE (S1)                ; チャネル主軸 1 = 物理主軸 5
                            ; が解放されます。
```

17.7 プログラミング

17.7.1 パートプログラムによるプログラミング

プログラミング命令

命令	説明	
SETMS:	メイン主軸は、以下のマシンデータで指定した主軸です。 MD20090 $MC_SPIND_DEF_MASTER_SPIND (チャネルのメイン主軸の初期設定)	
SETMS(<n>):	番号<n>の主軸はメイン主軸です (初期設定: MD20090 $MC_SPIND_DEF_MASTER_SPIND と異なる場合があります)。 メイン主軸は、次の機能に対して定義してください。	
	• G95:	毎回転送り速度
	• G96 S.../G961 S...:	周速一定制御(m/min または ft/min)
	• G97/G971:	G96/G961 をキャンセルし、前の主軸速度を維持する
	• G63:	フローティングチャックを使用したタッピング
	• G33/G34/G35:	ねじ切り
	• G331/G332:	フローティングチャックのないタッピング
	• G4 S...:	主軸回転でのドウェル時間
	• 主軸番号を入力せずに、M3、M4、M5、S、SVC、SPOS、M19、SPOSA、M40、M41 ～ M45、および WAITS をプログラム指令	
	現在のメイン主軸の設定は、リセット/パートプログラム終了とパートプログラム起動で保持できます。 この設定は、以下のマシンデータを使用しておこないます。 • MD20110 $MC_RESET_MODE_MASK • MD20112 $MC_START_MODE_MASK	
M3: M<n>=3:	メイン主軸の右回りの主軸回転 主軸番号<n>の右回りの主軸回転	

命令	説明
M4: M<n>=4:	メイン主軸の左回りの主軸回転 主軸番号**<n>**の左回りの主軸回転
M5: M<n>=5:	旋回のない、メイン主軸の主軸停止 主軸番号**<n>**の旋回のない主軸停止
S...: S<n>=...:	メイン主軸の主軸速度(rpm) 主軸番号**<n>**の主軸速度(rpm)
SVC=...: SVC[<n>]=...:	メイン主軸の切削速度(m/min または ft/min) 主軸**<n>**の切削速度(m/min または ft/min)
SPOS=...: SPOS[<n>]=...:	メイン主軸の主軸位置決め 主軸番号**<n>**の主軸位置決め ブロック切り替えは、主軸がインポジションになったときにのみおこなわれます。
SPOSA=...: SPOSA[<n>]=...:	メイン主軸の主軸位置決め 主軸番号**<n>**の主軸位置決め ブロック切り替えは即座に実行されます。 その他のパートプログラム処理に関係なく、主軸がその位置に到達するまで主軸位置決めは継続されます。
SPOS=DC(...): SPOS[<n>]=DC(...): SPOSA=DC(...): SPOSA[<n>]=DC(...):	移動中に位置決めする場合は動作方向が保持されて、この位置にアプローチします。 停止状態からの位置決めの場合、この位置には最短距離でアプローチします。
SPOS=ACN(...): SPOS[<n>]=ACN(...): SPOSA=ACN(...): SPOSA[<n>]=ACN(...):	この位置には常に負の動作方向でアプローチします。 必要に応じて、位置決めの前に動作方向が反転されます。
SPOS=ACP(...): SPOS[<n>]=ACP(...): SPOSA=ACP(...): SPOSA[<n>]=ACP(...):	この位置には常に正の動作方向でアプローチします。 必要に応じて、位置決めの前に動作方向が反転されます。

命令	説明
SPOS=IC(...): SPOS[<n>]=IC(...): SPOSA=IC(...): SPOSA[<n>]=IC(...):	移動距離が指定されます。 移動方向は、移動距離の前の符号から決定されます。 主軸が動作中である場合、移動方向は、プログラム指令方向に移動できるように必要に応じて反転されます。 使用可能なレファレンス点がないか、または、インタフェース信号を介して新しいレファレンス点が要求されているならば、移動中に原点マークを横切ったときに、主軸は原点マークと自動的に原点同期します。
M19: M[<n>]=19:	SD43240 の位置へのメイン主軸の位置決め SD43240 の位置への主軸番号**<n>**の位置決め ブロック切り替えは、主軸がインポジションになったときにのみおこなわれます。
M70: M<n>=70:	主軸を停止状態にし、位置制御を起動して、パラメータセット 0 を選択し、軸モードを起動 メイン主軸 主軸番号**<n>**
SPCON: SPCON(<n>): SPCON(<n>,<m>):	主軸位置制御オン メイン主軸 主軸番号**<n>** 主軸番号**<n>**と**<m>**
PCOF: SPCOF(<n>): SPCOF(<n>,<m>):	主軸位置制御オフ、速度制御モードを起動 メイン主軸 主軸番号**<n>** 主軸番号**<n>**と**<m>**
FPRAON(S<n>):	メイン主軸による、主軸**<n>**の毎回転送り速度オン
FPRAON(S<n>,S<m>):	主軸**<m>**による、主軸**<n>**の毎回転送り速度オン 毎回転送り速度値は、FA[S<m>]で指定してください。
FPRAOF(S<n>):	主軸**<n>**の毎回転送り速度オフ
C30 G90 G1 F3600	回転軸 C(軸モードの主軸)が、3600°/min = 10 rpm の速度で位置 30°に移動
G25 S...: G25 S<n>:	プログラム可能な最低主軸速度制限 メイン主軸 主軸番号**<n>**
G26 S...: G26 S<n>:	プログラム可能な最高主軸速度制限 メイン主軸 主軸番号**<n>**

命令	説明
	G96、G961、G97 によるプログラム可能な主軸最高速度制限
LIMS=...:	メイン主軸
LIMS[<n>]=...:	主軸番号**<n>**
VELOLIM[<主軸>]=...:	設定されたギヤ選択に依存したプログラム可能な最高速度制限 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名称で選択します。 ● 主軸運転モードの場合は主軸名称 S<n> ● 軸運転の場合は「C」などの軸名称 補正値は、以下を基準にします。 ● 軸運転での主軸(MD30455、ビット 6 = 0 の場合) 設定した最高軸速度(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 の場合) 有効なギヤユニット選択の最高速度 (MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>]) VELOIM のプログラミングの詳しい説明については、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 上級編』
WAITS:	メイン主軸の協調命令 後続のブロックは、前の NC ブロックで SPOSA を使用してプログラム指令された主軸が、その位置に(精密イグザクトストップで)達するまで処理されません。 M5 の後の WAITS: 主軸が停止するまで待機します。 M3/M4 の後の WAITS: 主軸がその指令速度に達するまで待機します。
WAITS(<n>):	主軸番号**<n>**の同期制御命令
WAITS(<n>,<m>):	主軸番号**<n>**と**<m>**の同期制御命令
FA[S<n>]:	主軸**<n>**の位置決め速度(軸送り速度)のプログラミング[°/min] FA[S<n>]=0 の場合は、設定した値が再度有効になります。 MD35300 $MA_SPIND_POSCTRL_VELO
OVRA[S<n>]:	主軸**<n>**の軸オーバライド値のプログラミング[%]
ACC[S<n>]:	[%]で示される主軸**<n>**の軸加減速能力のプログラミング

命令	説明
SPI(<n>):	SPI(<n>)では、主軸番号がマシンデータ MD35000 $MA_SPIND_ASSIGN_TO_MACHAX[ ] に従って、データタイプ AXIS に変換されます。 SPI は、主軸を使用して軸機能をプログラム指令する場合に使用します。 SPI を使用して、以下の命令を実行できます。 ● フレーム命令 – CTRANS() – CFINE() – CMIRROR() – CSCALE() ● 以下の主軸の速度値と加速度値: – FA[SPI(<n>)] – ACC[SPI(<n>)] – OVRA[SPI(<n>)] ● システム変数: – $P_PFRAME[SPI(<n>),TR]=<値> – $P_PFRAME= CTRANS(X,<軸の値>,Y,<軸の値>,SPI(<n>),<軸の値>) – $P_PFRAME= CSCALE(X,<比率>,Y,<比率>,SPI(<n>),<比率>) – $P_PFRAME=CMIRROR(S<n>,Y,Z) – $P_UBFR=CTRANS(A,10) : CFINE (19,0.1) SPI のプログラミングの詳しい説明については、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 上級編』
M40: M<n>=40:	メイン主軸の自動ギヤ選択 主軸番号**<n>**の自動ギヤ選択切替
M41 ～ M45: M<n>=41 ～ M<n>=45:	メイン主軸に対してギヤ選択 1 ～ 5 を選択 主軸番号**<n>**に対してギヤ選択 1 ～ 5 を選択

注記

主軸がチャネル内で設定されている場合は、M 機能の M3、M4、M5、および M70 が DB21、... DBB194 および DBB202 には出力されません。 これらの M 機能は、DB21、... DBB68 以降および関連する軸 DB、DB31、... DBB86 以降において拡張 M 機能として提供されます。

参照先

主軸のプログラミングについての詳しい説明は、以下に記載されています。

- 『プログラミングマニュアル、基本編』

17.7.2 シンクロナイズドアクションを介したプログラミング

M 機能の M40 ～ M45 は、シンクロナイズドアクションにもプログラムできます。

注:

- パートプログラム中で M40 ～ M45 をプログラムしても、シンクロナイズドアクションの自動ギヤ選択切替の現在の状態には影響しません。また、その逆も同様です。
- M40 で S 値をプログラムする場合、自動ギヤ選択切替は、シンクロナイズドアクションとパートプログラムに対して別々に有効になります。
- M40 は、電源投入後に無効になります。
  シンクロナイズドアクションから S 値が指定された場合、ギヤ選択は調整されません。
- シンクロナイズドアクションを使用してプログラムした M40 命令は、シンクロナイズドアクションに対して常に動作中のままとなり(モーダル)、リセット時にクリアされません。
- M41 ～ M45 は、パートプログラム中のプログラミングに従って、1 ～ 5 番目のギヤ選択を選択します。
  この機能を実行するには、軸入れ替えが必要です。
  ギヤ選択切替が実行されると、主軸状態は中立になります(M3、M4、M5 をプログラムした場合も動作は同じです)。

参照先

主軸のプログラミング、およびシンクロナイズドアクションからの主軸移動の詳しい説明については、以下を参照してください。

- 『プログラミングマニュアル、上級編』
- 『機能マニュアル、シンクロナイズドアクション』

17.7.3 FC18 による PLC を介した主軸制御のプログラミング - 840D sl のみ

PLC が FC18 を使用して回転方向および速度を指定すると、NC は、その速度と一致するギヤ選択を決定および選択できます。これは、パートプログラムを介してプログラミングする場合の M40 機能と同等です。

ギヤ選択を有効にするには、FC18 が PLC ユーザープログラム内で呼び出されたときに、適切な起動コードを設定してください。

参照先

PLC による FC18 を使用した主軸制御のプログラミングについての詳しい説明は、以下に記載されています。

- 『機能マニュアル、基本機能』;「PLC 基本プログラム(P3)」

17.7.4 NC/PLC インタフェース信号を使用してプログラミング

17.7.4.1 機能

注記

この機能は、SINUMERIK Operate の使用時にだけ有効です。

次のジョブ/タスクは、ジョブインタフェース DB31、… DBX30.0 - 4 を使用して、PLC ユーザープログラムから主軸に発行できます。

- 主軸停止
- 主軸起動、時計回り
- 主軸起動、左回り
- ギヤ選択指定
- 主軸起動位置決め

ジョブ ID

NC 側では主軸ジョブは、当該のインタフェース信号(0 → 1)の立ち上がり**エッジの変更**結果として識別されます。

必要条件

主軸ジョブの時点では、チャネル、プログラム、および主軸の状態について次の必要条件が満たされることが必要です。

- チャネル状態:
  - DB21、... DBX35.6 == 1 (チャネル状態「中断」)**または**
  - DB21、… DBX35.7 == 1 (チャネル状態「リセット」)
- プログラム状態:
  - DB21、... DBX35.3 = 1 (プログラム状態「中断」)**または**
  - DB21、... DBX35.4 = 1 (プログラム状態「中断」)
- 主軸状態
  - (「チャネル軸」**または**「中立軸」) **および NO**
  - (FC18 **または**シンクロナイズドアクションを使用する位置決め動作)

注記

エラーの場合の応答

異常が発生した場合、つまりジョブ信号が無効の状態で設定された場合、対応するジョブは**フィードバック信号なしで**無視されます。

チャンネルの割り付け

主軸ジョブは、該当する特定のチャネルで処理されます。このチャネルは、ジョブの時点で主軸に割り当てられます。

主軸に割り当てられるチャネルは、次の NC/PLC インタフェース信号を使用して決定できます。

DB31、... DBX68.0 - 3 (NC 軸/主軸のチャネル割り当て)

モード変更

主軸ジョブのコンテキスト内での定義は、モード変更を越えて維持されます。

プログラムの起動

主軸に割り当てられているチャネルで NC プログラムが起動した場合、実主軸の定義は維持されます。主軸定義は、NC プログラム命令またはシンクロナイズドアクションを使用して変更できます。

下記も参照

M40: 設定された切り替えレベルを超えた速度の自動ギヤ選択 (ページ 1676)

G2: 速度、指令値/フィードバック回路、閉ループ制御 (ページ 405)

17.7.4.2 試運転:マシンデータ

セッティングデータでの自動的な主軸定義の確定

PLC (FC18)によって、次のマシンデータ設定を使用して、NC プログラム、シンクロナイズドアクション、および主軸制御から主軸定義が当該のセッティングデータに伝送されます:
MD35035 $MA_SPIND_FUNCTION_MASK, bit x =<値>

ビット	値	意味
4	1	**速度指令値**は、NC プログラム、シンクロナイズドアクション、または FC18 によって入力され、次の主軸別セッティングデータに伝送されます。 **SD43200** $SA_SPIND_S[<主軸>] **注** プログラムされた速度値ではない S 値は伝送されません。たとえば、次の場合です。 ● 周速一定制御(G96、G961)での S 値 ● 回転関連ドウェル時間(G4)の S 値
	0	速度指令値はセッティングデータに伝送**されません**。
8	1	「周速一定制御」ファンクションが有効である場合(G96、G961、G962)、**切削速度**は NC プログラム、シンクロナイズドアクション、または FC18 によって次の主軸別セッティングデータに指令として入力されます。 **SD43202** $SA_SPIND_CONSTCUT_S[<主軸>] **注** プログラムされた切削速度値ではない S 値は伝送されません。たとえば、次の場合です。 ● 回転関連ドウェル時間(G4)の S 値
	0	切削速度指令値はセッティングデータに伝送**されません**。

ビット	値	意味
10	1	**メイン主軸**では、**送り速度タイプ**が次のセッティングデータに伝送されます。 **SD43206** $SA_SPIND_SPEED_TYPE[<主軸>]
	0	メイン主軸の送り速度タイプ定義はセッティングデータに伝送**されません**。

セッティングデータに主軸定義が伝送され、主軸ジョブを実行するために主軸が移動します。

17.7.4.3 試運転:NC/PLC インタフェース信号

ジョブインタフェース

- DB31、… DBX30.0 (主軸停止)
- DB31、… DBX30.1 (主軸起動、右回り)
- DB31、… DBX30.2 (主軸起動、左回り)
- DB31、... DBX30.3 (ギヤ選択)
- DB31、… DBX30.4 (主軸位置決め起動)

当該の状態信号

- DB21、... DBX35.3 (プログラム状態「中断」)
- DB21、... DBX35.4 (プログラム状態「キャンセル」)
- DB21、... DBX35.7 (チャネル状態「リセット」)
- DB21、... DBX35.6 (チャネル状態「中断」)
- DB31、... DBX68.0 - 3 (NC 軸/主軸のチャネル割り当て)
- DB31、... DBX 84.0 (周速一定制御)

参照先:

NC/PLC インタフェース信号の詳細については、『NC 変数およびインタフェース信号用パラメータマニュアル』を参照してください。

17.7.4.4 速度指令値 (SD43200)

主軸ジョブを使用して主軸を起動する場合、セッティングデータに入力された主軸別の値は主軸速度として有効になります。

SD43200 $SA_SPIND_S[<主軸>] (主軸が PLC ジョブインタフェース経由で起動する場合の速度)

新しい速度値の書き込み

次の状態では、新しい速度値がセッティングデータに書き込まれます。

- 速度指令値は、NC プログラム、シンクロナイズドアクション、または FC18 によって入力された
  補足条件:
  - MD35035 $MA_SPIND_FUNCTION_MASK (ページ 1659)、ビット 4 == 1
  - DB31、... DBX84.0 == 0 (周速一定制御)
- NC プログラムまたはシンクロナイズドアクションへのセッティングデータの書き込み
- HMI (OPI)によるセッティングデータの書き込み@@@

必要条件

ギヤ選択切替

指令速度が実ギヤ選択の速度レンジ外である場合、ギヤ選択切替は開始されません。

例外、「M40: 設定された切り替えレベルを超えた速度の自動ギヤ選択 (ページ 1676)」の章を参照してください。

17.7.4.5 周速一定制御の入力(SD43202)

主軸ジョブを使用してメイン主軸を起動する場合、および「周速一定制御」主軸速度タイプが有効である場合、セッティングデータに入力された主軸別値が周速一定制御として有効になります。

SD43202 $SA_SPIND_CONSTCUT_S[<主軸>] (主軸が PLC ジョブインタフェース経由で起動する場合の切削速度)

新しい切削速度値の書き込み

次の状態で、メイン主軸の新しい周速一定制御が主軸別のセッティングデータに書き込まれます。

- 周速一定制御は、NC プログラム、シンクロナイズドアクション、または FC18 によって入力された
  補足条件:
  - MD35035 $MA_SPIND_FUNCTION_MASK (ページ 1659)、ビット 8 == 1
  - DB31、... DBX84.0 == 1 (周速一定制御が有効)
- NC プログラムまたはシンクロナイズドアクションへのセッティングデータの書き込み
- HMI (OPI)によるセッティングデータの書き込み@@@

単位系

セッティングデータを書き込む場合、次の二次条件に対応して値が解釈されます。

- NC プログラムまたはシンクロナイズドアクションによる書き込み:
  - G700 が有効:feet/min
  - G710 が有効:m/min
  - G70、G71 が有効:MD10240 $MN_SCALING_SYSTEM_IS_METRIC の設定に依存
- SINUMERIK Operate による書き込み:
  - MD10240 $MN_SCALING_SYSTEM_IS_METRIC の設定に依存
- FC18 による書き込み:
  - ファンクション番号 10: 「定切削速度(m/min)」
  - ファンクション番号 11: 「定切削速度(feet/min)」

システムまたは OPI 変数による読み取り

NC プログラム、シンクロナイズドアクション、または HMI では、周速一定制御に対して現在入力されている指令は次のシステムデータを使用して読み取ることができます。

システム変数	OPI 変数	意味
$P_CONSTCUT_S	---	最後にプログラムされた切削速度
$AC_CONSTCUT_S	acConstCutS	現在の周速一定制御

17.7.4.6 メイン主軸の主軸速度タイプの入力(SD43206)

パートプログラム、FC18、またはシンクロナイズドアクションからの**メイン主軸の**主軸速度タイプの定義は、すべての決まったソースから次のセッティングデータに書き込まれます。

SD43206 $SA_SPIND_SPEED_TYPE (PLC インタフェースを介した主軸起動の主軸速度タイプ)

数値の範囲と機能は、15 番目のグループ(送り速度タイプ)に対応しています。

許容値は G 値:93、94、95、96、961、97、および 971 です。

DB31、… DBX30.1/2 (主軸起動、右回り/左回り)に対し、設定に応じて、SD43200 $SA_SPIND_S の速度または SD43202 $SA_SPIND_CONSTCUT_S の切削速度のどちらかが有効になります。

93、94、95、97、および 971:	メイン主軸は、SD43200 の速度で起動します。
96 と 961:	メイン主軸の速度は、指定された切削速度(SD43202)と径方向軸の半径から入手されます。

17.7.5 外部プログラミング(PLC、HMI)

SD43300 および SD42600

毎回転送り速度の挙動は、チャネルセッティングデータ SD42600 $SC_JOG_FEED_PER_REV_SOURCE (JOG モードでの毎回転送り速度制御)を使用して、JOG 操作モードで、軸セッティングデータ: SD43300 $SA_ASSIGN_FEED_PER_REV_SOURCE (主軸の毎回転送り速度)を介して外部的に選択できます。

セッティングデータを介して次の設定を行うことができます。

>0:	毎回転送り速度が得られる回転軸/主軸の機械軸番号。
-1:	毎回転送り速度は、軸/主軸がいずれの場合にも有効なチャネルのメイン主軸から得られます
0:	機能は選択解除されます。

FPRAON (S2)

メイン主軸から得られる、主軸 S2 の毎回転送り速度 ON。

FPRAON (S2, A)

軸 A から得られる、主軸 S2 の毎回転送り速度 ON。
毎回転送り速度値は、`FA[Sn]`で指定してください。

FPRAOF (S2)

主軸 S2 の毎回転送り速度 OFF。

SPI(n)

`SPI(Sn)`の代わりに `SPI(n)`をプログラムすることも可能です。

17.8 主軸監視

17.8.1 許容速度範囲

パラメータ設定またはプログラム指令された速度制限値と有効な主軸機能(G94、G95、G96、G961、G97、G971、G33、G34、G35、G331、G332 など)によって、主軸の許容速度レンジが決まります。

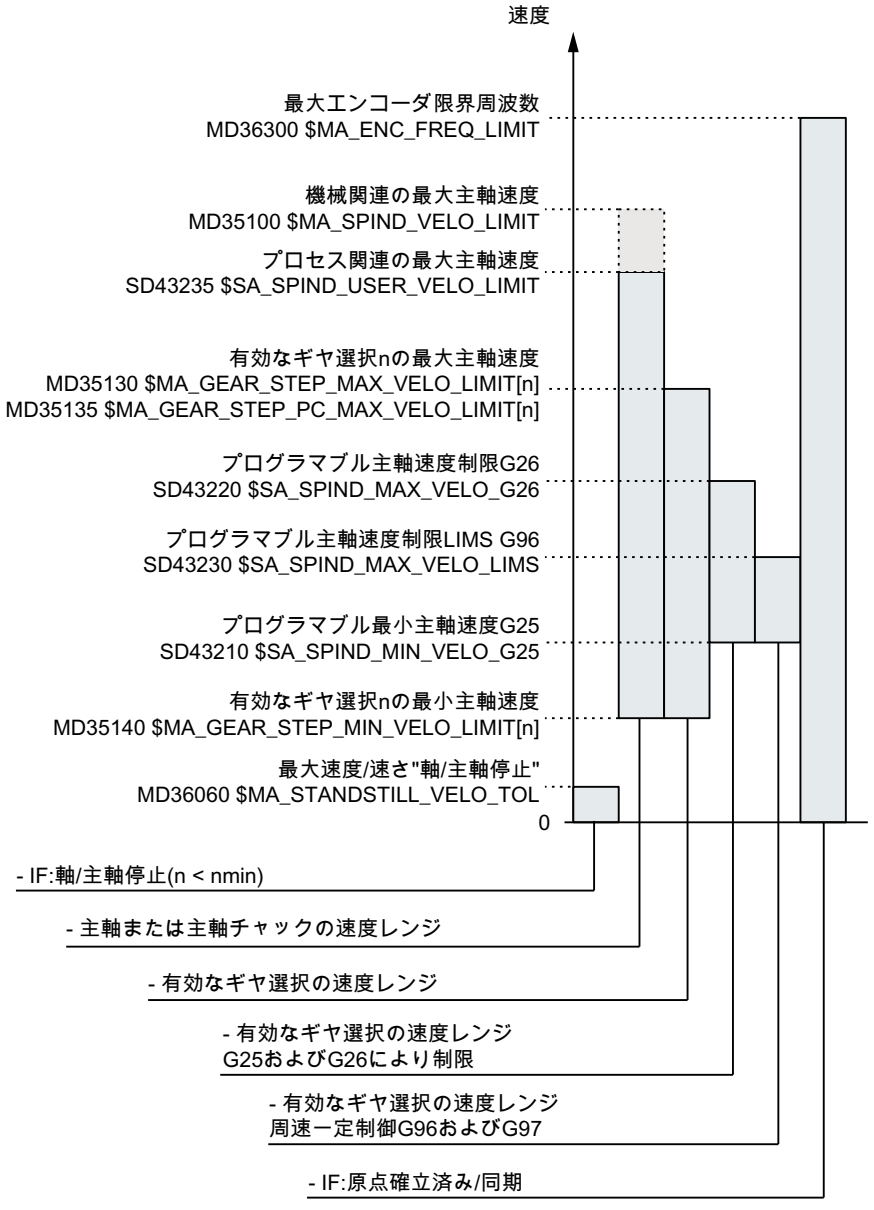


図 17-8　主軸監視機能/速度の範囲

17.8.2 軸/主軸停止

工具交換、機械の扉を開ける、軌跡送り速度有効などの機能は、主軸が停止状態の場合にのみ機械で行うことができます。

機能

指令値が生成されなくなり、主軸の実速度が「軸/主軸停止」用に設定された検出値を下回ると、「軸/主軸停止」状態に到達します。

MD36060 $MA_STANDSTILL_VELO_TOL (「軸/主軸停止」用の最高速度)

主軸が停止状態になると、次の NC/PLC インタフェース信号が設定されます。

DB31、... DBX61.4 (軸/主軸停止)

有効性

主軸停止の監視は、すべての主軸モードおよび軸モードで有効です。

軌跡速度の停止

主軸が開ループ制御モード(M5)で停止するとき、次のマシンデータが設定されていると軌跡速度が無効になります。

MD35510 $MA_SPIND_STOPPED_AT_IPO_START (主軸停止の送り速度有効)

主軸が停止状態になると、軌跡速度が再び有効になります。

17.8.3 主軸指令一致

機能

「主軸指令一致」監視は、次のことをチェックします。

- プログラム指令された主軸速度に到達したかどうか
- 主軸が停止状態にあるかどうか:
  DB31、... DBX61.4 (軸/主軸停止) = 1
- 主軸がまだ加速または減速区間にあるかどうか

主軸モードの開ループ制御モードでは、指令速度が実速度と比較されます。実速度が、MD(後述を参照)に入力できる主軸許容範囲以上に離れていると、

- 次の軸 NC/PLC インタフェース信号が「0」に設定されます。
  DB31、... DBX83.5 (主軸指令一致) = 0
- 次の加工ブロックは有効になりません
  (MD35500 $MA_SPIND_ON_SPEED_AT_IPO_START の設定に対応。「軸/主軸停止 (ページ 1666)」を参照してください)。

主軸の指令速度

主軸の指令速度は、主軸オーバライドと有効な制限値を考慮して、プログラム指令速度から算出されます。

プログラム指令速度が制限されるか加速された場合、DB31、... DBX83.1 (指令速度制限中)または DB31、... DBX83.2 (指令速度増加)を使用して、これが示されます(「ギヤ選択の最高/最低速度 (ページ 1668)」も参照してください)。指令速度の許容範囲への到達は、妨げられ**ません**。

指令速度の許容範囲

指令速度の許容範囲は、主軸速度許容範囲係数で定義されます。

MD35150 $MA_SPIND_DES_VELO_TOL

例:

MD35150 $MA_SPIND_DES_VELO_TOL = 0.1

⇒ 主軸の実速度は、指令速度から±10%の誤差が許容されます。

主軸の実速度が許容範囲内にある場合、次の NC/PLC インタフェース信号が「1」に設定されます。

DB31、... DBX83.5 (主軸指令一致) = 1

特記事項:

主軸速度の許容範囲が「0」に設定されている場合、DB31、... DBX83.5 (主軸指令一致)は常に「1」に設定され、軌跡制御はおこなわれません。

速度変更

軌跡制御は、移動ブロックの起動時にのみ、速度変更がプログラムされている場合にかぎり実行されます。たとえば過負荷などにより速度許容範囲から逸脱した場合、軌跡移動は自動的に停止状態になりません。

17.8.4 ギヤ選択の最高/最低速度

最低速度

主軸のギヤ選択の最低速度は、次のマシンデータで設定されます。

MD35140 $MA_GEAR_STEP_MIN_VELO_LIMIT[<n>]

オーバライドを考慮して生成された指令速度は、最低速度を下回りません。

最低速度を下回る S 値がプログラム指令されている場合、指令速度は最低速度まで引き上げられ、次の NC/PLC インタフェース信号が設定されます。

DB31、... DBX83.2 (速度指令値増加)

ギヤ選択最低速度は、速度モードでのみ有効で、次によってのみ下回ることができます。

- 主軸オーバライド 0%
- M5
- S0
- DB31、... DBX4.3 (主軸停止)
- DB31、... DBX2.1 (コントローライネーブル取り消し)
- DB21、... DBX7.7 (リセット)
- DB31、... DBX2.2 (残移動距離削除/主軸リセット)
- DB31、... DBX18.5 (揺動速度)
- DB21、... DBX7.4 (NC ストップ軸と主軸)
- DB31、... DBX1.3 (軸/主軸無効)
- DB31、... DBX16.7 (S 値削除)

最高速度

主軸のギヤ選択の最高速度は、次のマシンデータで設定されます。

MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>]

オーバライドを考慮して生成された指令速度は、この速度に制限されます。

速度が制限されている場合、次の NC/PLC インタフェース信号が設定されます。

DB31、... DBX83.1 (速度指令値制限中)

17.8.5 主軸速度制限の診断

システム変数

有効な/制限的な主軸パラメータは、以下のシステム変数で読み取ることができます。システム変数は、主軸番号でインデックスされていなければならず、速度制御モードおよび主軸位置決めモードだけに関連する値を返します。

システム変数	意味
$AC_SMAXVELO[<n>]	最高主軸速度制限[RPM]。
$AC_SMAXVELO_INFO[<n>]	速度制限データ要素の ID。
$AC_SMINVELO[<n>]	最高主軸速度制限[RPM]。
$AC_SMINVELO_INFO[<n>]	速度増加データ要素の ID。
$AC_SMAXACC[<n>]	最高主軸加速度制限[rev/s^2]。
$AC_SMAXACC_INFO[<n>]	加速度制限データ要素の ID。
$AC_SPIND_STATE[<n>]	速度制御モードの主軸状態。
<n>:0、1、2、...最大主軸番号(0:チャネルの現在のメイン主軸)	

参照先

システム変数の詳細については、以下を参照してください。

『リストマニュアル』、「システム変数」

NC/PLC インタフェース信号

主軸速度の制限または増加は、以下の NC/PLC インターフェース信号によって通知されます。

- DB31、... DBX83.1 (速度指令値制限)
- DB31, ... DBX83.2 (速度指令値増加)

境界条件

主軸モード

システム変数によって提供される値は、次のように主軸モードに依存します。

- 速度制御モード
  すべてのシステム変数は現在の値を提供します。
- 位置決めモード:
  システム変数$AC_SMAXVELO、$AC_SMAXACC および$AC_SPIND_STATE は有効な値を提供します。システム変数$AC_SMINVELO および$AC_SMINVELO_INFO は、速度制御モードへの切り替え時に有効になるデータを提供します。
- 軸モード(例: 主軸が座標変換 TRANSMIT、TRACYL、...で使用されているか、付加軸として軌跡移動に追従しする場合)
  システム変数$AC_SPIND_STATE は、軸モードでも使用できます。動的データについて軸モードでは次の異なるシステム変数が使用可能です。
  $AA_VMAXM、$AA_VMAXB および$AA_VLFCT。

SERUPRO ブロック検索

SERUPRO ブロック検索に対して、以下の制御応答が得られます。

- システム変数$AC_SMAXVELO / $AC_SMAXACC は表示可能な最高速度/加減速を提供します。
- $AC_SMAXVELO_INFO および$AC_SMAXACC_INFO は、VALUE「0」を提供します(制限は無効)。
- $AC_SMINVELO および$AC_SMINVELO_INFO は、通常のパートプログラム処理の場合と同様にデータを提供します。
- $AC_SPIND_STATE は、SERUPRO について設定される状態を返します。

例

主軸 1 の有効な/制限的な主軸パラメータが、R 変数に定期的に書き込まれ、HMI の[パラメータ|ユーザー変数]操作エリアに R 変数として表示されます

プログラムコード

```
N05 IDS=1 WHENEVER TRUE DO $R10=$AC_SMAXVELO[1]
N10 IDS=2 WHENEVER TRUE DO $R11=$AC_SMAXVELO_INFO[1]
N15 IDS=3 WHENEVER TRUE DO $R12=$AC_SMINVELO[1]
N20 IDS=4 WHENEVER TRUE DO $R13=$AC_SMINVELO_INFO[1]
N25 IDS=5 WHENEVER TRUE DO $R14=$AC_SPIND_STATE[1]
```

下記も参照

主軸指令一致 (ページ 1666)

17.8.6 最高主軸速度

最高主軸速度:パラメータ設定可能な、機械関連の制限値

機械関連の最大主軸速度(主軸チャックまたは工具を保護するためなどの)は、以下のマシンデータを使用してパラメータ設定します。

MD35100 $MA_SPIND_VELO_LIMIT (最高主軸速度)

主軸速度は、NC によりフィードバック値で監視されます。つまり、現在のギヤ選択が考慮されます。

注記

工作機械メーカ

マシンデータ MD35100 での最高主軸速度の変更は、主軸が停止している場合にのみおこなってください。これは特に、NC リセット後に有効になる主軸に対して重要です (MD35040 SPIND_ACTIVE_AFTER_RESET も参照してください)。これに従わない場合は、アラーム 22100 が出力されます。

参照先

- 工作機械メーカの取扱説明書:『リストマニュアル』、「マシンデータの詳しい説明」
- 『ユーザーマニュアル』『診断マニュアル』

違反に対する動作

主軸の現在速度が、パラメータ設定された最高主軸速度を、許容範囲値より大きく超えている場合は、

「現在の主軸速度」> MD35100 $MA_SPIND_VELO_LIMIT (最大主軸速度)
+ MD35150 $MA_SPIND_DES_VELO_TOL (主軸速度の許容範囲値)

以下の動作が NC 側で発生します。

- DB31、... DBX83.0 = 1 (速度制限超過)
- アラーム 22100「チャック速度超過」が表示されます。
- チャネルのすべての軸と主軸が停止します。

注記

回転軸/主軸

主軸が、一時的に回転軸としても動作している場合は、アラーム 22100「チャック速度超過」が、以下の場合に表示されます。

「回転軸の現在速度」> MD35100 $MA_SPIND_VELO_LIMIT (最大主軸速度)
+ MD35150 $MA_SPIND_DES_VELO_TOL (主軸速度の許容範囲値)

解決策：回転動作時に最大主軸速度を最大回転軸回転数に合わせます。

MD35100 $MA_SPIND_VELO_LIMIT = MD32000 $MA_MAX_AX_VELO (回転軸の最高速度)

変更を有効にするためにリセットをトリガしてください。

最高主軸速度:設定可能なプロセス関連の制限値

プロセス関連の最大主軸速度は、以下のセッティングデータで設定します。

SD43235 $SA_SPIND_USER_VELO_LIMIT (最高主軸速度)

変更するには、以下の方法でセッティングデータを記述してください。

- NC プログラムでのプログラミング
- 機械のオペレータによるユーザーインターフェースからの手動の記述

変更は直ちに有効になります。

速度指令値の制限

適切な場合は、コントローラが、プログラム指令された主軸速度の指令値を、セッティングデータで指定された値に制限します。その後、制限は、以下のシステム変数を介して表示されます。

$AC_SMAXVELO_INFO[<主軸番号>] == 21

17.8.7 最大エンコーダ限界周波数

> ⚠ 注意
>
> **限界値違反**
>
> 実際の主軸位置エンコーダの最大エンコーダ周波数制限は、コントローラにより監視されます(この制限は超過することがあります)。工作機械メーカで、主軸モータ、ギヤボックス、検出器ギヤボックス、エンコーダおよびマシンデータの設定をすることで、確実に、実際の主軸位置エンコーダの最大エンコーダ周波数制限を超えないようにしてください。

最大エンコーダ制限周波数の超過

主軸速度が、最大エンコーダ限界周波数(エンコーダの機械的な最高速度制限を超えてはいけない)を超える速度(プログラム指令された大きい方の S 値)に到達すると、同期が失われます。主軸は引き続き回転しますが、機能が制限されます。

次の各機能を使用すると、有効な検出器がエンコーダ限界周波数未満で再度動作するまで、主軸速度が低下します。

- ねじ切り(G33、G34、G35)
- フローティングチャックを使用しないタッピング(G331、G332)
- 毎回転送り(G95)
- 周速一定制御(G96、G961、G97、G971)
- SPCON (位置制御主軸動作)

エンコーダ限界周波数を超えると、その検出器の
NC/PLC IS:
DB31、... DBX60.4 (原点確立済み/原点同期済み 1)
または
DB31、... DBX60.5 (原点確立済み/原点同期済み 2)
がリセットされ、NC/PLC IS:
DB31、... DBX60.2 (エンコーダ限界周波数超過 1)
または
DB31、... DBX60.3 (エンコーダ限界周波数超過 2)
が設定されます。

主軸が軸モードになっている場合、最大エンコーダ限界周波数を超えてはいけません。最高速度(MD32000 $MA_MAX_AX_VELO)は、最大エンコーダ限界周波数未満にしてください。さもなければ、アラーム 21610 が出力され、軸が停止状態になります。

最大エンコーダ限界周波数の未達

最大エンコーダ限界周波数を超えた後、速度が最大エンコーダ限界周波数を下回ると(プログラム指令された小さい方の S 値、主軸オーバライドスイッチの変更など)、主軸は自動的に、次の原点マークまたは次の近接スイッチ信号で原点同期します。同期を失い、現在はその最大エンコーダ限界周波数には達していない有効な位置検出器では必ず、新しい原点同期が実行されます。

特記事項

次の各機能が有効な場合、最大エンコーダ周波数を超えることはできません。

- 主軸位置決めモード、軸モード
- ねじ切り(G33、G34、G35)
- フローティングチャックなしのタッピング G331, G332 (G63 には適用されません)
- 毎回転送り(G95)
- 周速一定制御(G96、G961、G97、G971)
- SPCON

17.8.8 終点監視

終点監視

位置決め(主軸が位置決めモードである)実行中、システムは、(実際の位置に関する)主軸からプログラム済み主軸指令値(終点)までの距離を監視します。

この監視が作動するために、マシンデータ:
MD36000 $MA_STOP_LIMIT_COARSE (汎用イグザクトストップ範囲)
および
MD36010 $MA_STOP_LIMIT_FINE (精密イグザクトストップ範囲)
の 2 つの制限値を、主軸位置指令値から開始されるインクリメンタル距離として定義できます。

この 2 つの制限値とは関係なく、主軸の位置決めは常に、接続した主軸測定エンコーダ、バックラッシ、伝達比などで定義される範囲で正確です。

パラメータ設定によって異なるイグザクトストップ範囲

パラメータ設定によって異なるさまざまなイグザクトストップ範囲を設定できます。
この結果、軸モードおよび主軸位置決めで各種レベルの精度に合わせて動作できます。
イグザクトストップ範囲は、主軸位置決めのギヤ選択ごとに個別に設定できます。

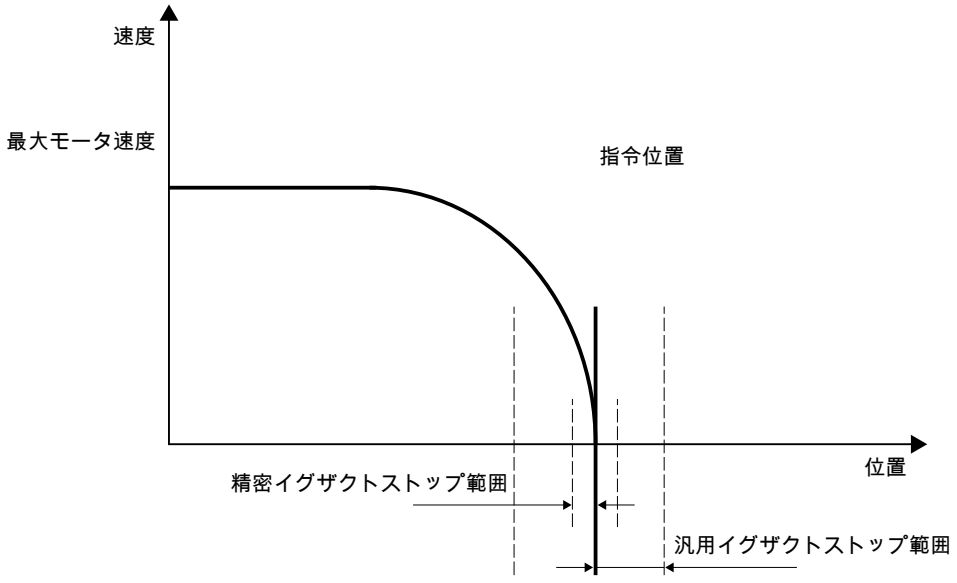


図 17-9　主軸のイグザクトストップ範囲

DB31、... DBX60.7 および DB31、... DBX60.6 (汎用/精密イグザクトストップ位置に到達)

マシンデータ:
MD36000 $MA_STOP_LIMIT_COARSE (汎用イグザクトストップ範囲)
と
MD36010 $MA_STOP_LIMIT_FINE (精密イグザクトストップ範囲)
によって定義されている 2 つの制限値は、NC/PLC IS:
DB31、... DBX60.7 (汎用イグザクトストップ位置に到達)
と
DB31、... DBX60.6 (精密イグザクトストップ位置に到達)を使用して PLC に出力されます。

SPOS および M19 のブロック変更

SPOS または M19 を使用して主軸の位置決めをすると、
NC/PLC IS:
DB31、... DBX60.6 (精密イグザクトストップ位置に到達)による終点監視に応じてブロックが変更されます。

ブロックにプログラムされている他の機能がすべて、その終了基準に到達したことが必要になります(たとえば、補助機能はすべて PLC により応答されます)。

SPOSA では、ブロックの変更は、終点の監視に左右されません。

17.8.9 M40: 設定された切り替えレベルを超えた速度の自動ギヤ選択

機能

M40 が有効の時は、プログラムされた主軸速度 S...が設定された切り替えレベルを超えた場合も、自動ギヤ選択切替が行われます。

これには、次の三つの場合があります。

- **プログラムされた速度が高すぎる場合**
  プログラムされた速度が、数値の一番大きなギヤ選択に設定された最高速度を超えています。
  S... > MD35110 $MA_GEAR_STEP_MAX_VELO[<n>]
  この場合、**最上位のギヤ**が選択されます(MD35090 $MA_NUM_GEAR_STEPS に応じて)。
- **プログラムされた速度が低すぎる場合**
  プログラムされた主軸速度が、第 1 ギヤ選択に設定された最低速度を下回っています。
  S... < MD35120 $MA_GEAR_STEP_MIN_VELO[1]
  この場合、**第 1 ギヤ**が選択されます。
- **プログラムされた速度 = 0 の場合**
  プログラムされた速度が 0 の場合(S0)、動作は第 1 ギヤ選択の最低速度 MD35120 $MA_GEAR_STEP_MIN_VELO[1]の設定によって決まります。
  - MD35120 $MA_GEAR_STEP_MIN_VELO[1] **= 0** が設定されている場合、S0 をプログラムすると、**第 1 ギヤ**が選択されます。
  - MD35120 $MA_GEAR_STEP_MIN_VELO[1] **> 0** が設定されている場合、S0 **をプログラムすると、ギヤ選択切替**は行われず、前回のギヤ選択がそのまま有効になります。つまり、S0 (M5 ではなく)を使用して、ギヤ選択切替を起動せずに主軸を停止することが可能です。

有効性

自動ギヤ選択(M40)での最上位ギヤまたは第 1 ギヤの選択は、パートプログラムやシンクロナイズドアクションにより、または PLC の FC18 からの入力で、主軸速度 S...をプログラムしている場合に有効です。

G331 を使ったタッピング用のパートプログラムによる速度のプログラミングでは、ギヤ選択のための第 2 データセットに対しても動作がサポートされます(必要条件:MD35010 $MA_GEAR_STEP_CHANGE_ENABLE、ビット 5 = 1)。

必要条件

ギヤ選択切替の有効化

この機能の必要条件は、ギヤ選択切替がマシンデータを介して全般的に有効になっていることです。

MD35010 $MA_GEAR_STEP_CHANGE_ENABLE (ギヤ選択切替へのパラメータ割り当て)。

MD35090 $MA_NUM_GEAR_STEPS (設定したギヤ選択の数)

例

自動ギヤ選択切替 M40 は、NC リセット後の初期設定です。

パートプログラム:

プログラムコード	コメント
...	
N15 S3500 M3	; S3500 が、第 2 ギヤ選択の MD35110 を上回っています。The 2nd gear stage is selected.
...	
N50 S0 M3	; 主軸が停止しています。S0 はギヤ選択切替を要求しません(特別処理、S0)。
...	
N95 S5 M3	; S5 が、第 1 ギヤ選択の MD35120 を下回っています。The 1st gear stage is selected.
...	

主軸 1 (AX5)のデータ設定:

```
MD35010 $MA_GEAR_STEP_CHANGE_ENABLE[AX5] = 1  ; ギヤ選択切替有効
MD35090 $MA_NUM_GEAR_STEPS[AX5] = 2           ; 既存のギヤ選択の数
MD35110 $MA_GEAR_STEP_MAX_VELO[1,AX5] = 500   ; ギヤ選択 1 の上側の切り替えレベル
MD35120 $MA_GEAR_STEP_MIN_VELO[1,AX5] = 10    ; ギヤ選択 1 の下側の切り替えレベル
```

MD35110 $MA_GEAR_STEP_MAX_VELO[2,AX5] = 2000 ; ギヤ選択 2 の上側の切り替えレベル

MD35120 $MA_GEAR_STEP_MIN_VELO[2,AX5] = 500 ; ギヤ選択 2 の下側の切り替えレベル

17.9 SMI 24 付き主軸(ワイス主軸)

17.9.1 一般事項

コントローラ内の主軸のセンサデータを処理できるようにするために、まずセンサを I/O モジュールに接続して、フィールドバス(PROFIBUS DP または PROFINET I/O)を介して PLC に伝送してください。

SMI 24 (ワイス主軸)付き主軸の場合は、センサデータは DRIVE-CLiQ を使用してドライブに伝送され、そこでドライブパラメータで使用可能となります。 周期的なドライブメッセージ 139 を使用すると、ドライブからのセンサデータがコントローラに伝送されます。 センサデータはコントローラ内で、以下のシステムデータで使用可能となります。

- システム変数
- OPI 変数
- NC/PLC インタフェース信号

必要条件

- 主軸が DRIVE-CLiQ を使用して、センサモジュール SMI 24 によりドライブに接続されていること。.
- ドライブメッセージ 139 が主軸用に設定されていること。

注記

ドライブメッセージ 139

原則として、センサモジュール SMI 24 付きの主軸は別のドライブメッセージでも操作できます。 ただし、センサデータはドライブメッセージ 139 を使用してのみコントローラに伝送されます。

17.9.2 センサデータ

主軸モータ内のセンサ

主軸センサは、クランプデバイスに関する情報とモータシャフトの回転位置に関する情報を提供します。

- アナログセンサ S1:クランプ済み状態
  ドローバーの位置に応じて、電圧値は 0 ～ 10 V の範囲になります。
- ディジタルセンサ S4:ピストン終端位置
  - 0 = ピストンがインポジションでない
  - 1 = ピストンがインポジション、つまり、ピストンは自由に移動できます。
- ディジタルセンサ S5:モータシャフトの回転位置
  - 0 = モータシャフトが割り出しされていない
  - 1 = モータシャフトがインポジション(必要条件:主軸が停止状態)

注記

センサモジュール SMI 24 付きの主軸と軸コンテナ

センサモジュール SMI 24 とセンサデータをコントローラに伝送するためのドライブメッセージ 139 を備えた主軸は、NCU リンクを介して複数の NCU に分散される軸を持つ軸コンテナに入れないでください。

センサデータの伝送

センサデータは、プロセスデータ 11 ～ 14 と同様に、周期的なドライブメッセージ 139 でセンサモジュール SMI 24 から制御装置に伝送されます。ドライブメッセージ 139 は、2 番目のエンコーダのデータの代わりにセンサデータが伝送されるドライブメッセージ136 に基づいています。ドライブメッセージ 139 の詳しい説明は、以下を参照してください。

参照先

『SINAMICS S120/S150 リストマニュアル』;「ファンクションダイアグラム」、「PROFIdrive」の章

システムデータ:センサデータ

センサデータは、以下のシステムデータを介してコントローラに読み込まれます。

意味	システム変数 $VA_	NC/PLC インタフェース DB31、...	OPI 変数	ドライブパラメータ
センサ設定	MOT_SENSOR_CONF[<軸>]	DBB132、DBB133	vaMotSensorConf	r5000
クランプ済み状態 [1]	MOT_CLAMPING_STATE[<軸>]	DBW134	vaMotClampingState	r5001
アナログセンサ S1 の測定値 [1] [2]	MOT_SENSOR_ANA[<軸>]	DBW136	vaMotSensorAna	r5002
ディジタルセンサの状態	MOT_SENSOR_DIGI[<軸>]	DBB138、DBB139	vaMotSensorDigi	r5003
<軸>:機械軸名称:AX1 ... AXn または主軸名称:S1 ... Sm				

[1] 「クランプ済み状態 (ページ 1681)」の章を参照してください

[2] センサ S1:0 ～ 10 V

アナログフィードバック値:10000 インクレメント、最小単位 1 mV

例:

SIMATIC S7 入力モジュール 27648 インクレメント、最小単位 0.36 mV

SMI 24 付き主軸に切り替える場合の調整係数:(10000 インクレメント * 1 mV) / (27648 インクレメント * 0.36 mV) = 1.00469393

システムデータの詳細

NC/PLC インタフェース信号	『NC 変数とインタフェース信号』リストマニュアル
システム変数:	『リストマニュアル』、「システム変数」
OPI 変数:	『リストマニュアル 2』、「変数」
インバータパラメータ:	『SINAMICS S120/S150 リストマニュアル』

17.9.3 クランプ済み状態

クランプデバイスの位置に応じて、センサ S1 は 0 V ～ 10 V のアナログ電圧値を提供します。電圧値は、ユーザー側のクランプ済み状態を評価するために、システムデータで使用できます。

注記

以下に説明するクランプ済み状態の状態値を生成するためのセンサ S1 の評価と主軸速度の制限は、以下の状態値がドライブパラメータ r5000 で表示されている場合にだけ実現されます。:

- r5000.0 == 1:センサが使用可能
- r5000.1 == 1:センサ S1 (クランプ済み状態)が使用可能
- r5000.10 == 1:状態値が生成され、速度制限 p5043 が有効

「センサデータ (ページ 1679)」の章の「システムデータ:センサデータ」の節も参照してください。

状態値

評価を簡素化するために、システムデータ内のクランプ済み状態も状態値 0 ～ 11 として使用できます。

電圧範囲は特定のクランプ済み状態に対応しています。電圧範囲は、ドライブパラメータ p5041[0...5]を使用して設定できます。

ドライブパラメータ p5040 を使用して、電圧範囲に対して電圧許容範囲を設定することもできます。

注記

電圧範囲±電圧許容範囲はオーバラップしてはなりません。

速度制限

状態値 3 ～ 10 のクランプ済み状態の場合、制限値をドライブパラメータ p5043[0...6]を使用して指定できます。 速度制限値をドライブパラメータ p5043[0...6]を使用して指定できます。その他のクランプ済み状態(状態値 1、2、および 11)では、制限値 0 [rpm]が固定値として適用されます。

各クランプ済み状態で、コントローラが主軸速度を適用される制限値に制限します。

注記

速度制限値の変更

ライブパラメータ p5043[0...6]による速度制限値の変更は、以下の後にのみコントローラ内で有効になります(主軸速度の制限値が新しい速度制限値に変わります)。

- 電源投入リセット後またはコントローラの電源切断/電源投入後
- 主軸の「パーキング」状態解除後(「機械軸のパーキング (ページ 145)」の章を参照してください)

関連:状態値、電圧レンジ、および速度制限

状態値 1)	クランプ済み状態	電圧範囲 2)		速度制限
		上限値	下限値	
0	センサ S1 が使用できないか、ステータス値が有効になっていません。	---	---	---
1	状態初期化実行中	---	---	3)
2	信号による解放済み	---	p5041[0] + p5040	3)
3	解放済み	p5041[0]	p5041[1]	p5043[0]
4	工具によるクランプ中	---	---	p5043[1]
5	工具による解放中	---	---	p5043[2]
6	工具なしの解放中	---	---	p5043[3]
7	工具によるクランプ済み、かつ、S4 = 0	p5041[2]	p5041[3]	p5043[4]
8	工具によるクランプ済み、かつ、S4 = 1			p5043[4]
9	工具なしのクランプ中	---	---	p5043[5]
10	工具なしのクランプ済み	p5041[4]	p5041[5]	p5043[6]
11	信号によるクランプ済み(エラー状態)	p5041[5]～p5040	---	3)

1) ステータス値は、以下のシステムデータを使用してコントローラに読み込まれます。
   - システム変数: $VA_MOT_CLAMPING_STATE[<軸>]
   - NC/PLC インタフェース DB31、...DBW134
   - OPI 変数: vaMotClampingState
   - ドライブパラメータ r5001
2) p5041[0...5]:電圧検出値、p5040:電圧検出値許容範囲
3) 速度制限は固定値として設定されます:0 [rpm]

17.9.4 追加ドライブパラメータ

P5042: 移行時間

以下の時間を、クランプ済み状態の識別のためにドライブパラメータ p5042 で設定することができます。

- p5042[0]: 「工具によるクランプ済み」の場合の安定化時間
  状態がコントローラに通知される前に、主軸モータは最低でも設定された安定化時間のあいだ、クランプ済み状態「工具によるクランプ済み」でなければなりません。
- p5042[1]: クランプに要する最大時間
  「解放済み」状態から「工具によるクランプ済み」状態または「工具なしのクランプ済み」状態への移行には、最大でも設定された時間を要します。

r5044: クランプサイクルの速度制限

クランプ済み状態「工具なしのクランプ済み」で有効になる p5043[6]の速度制限は、ドライブパラメータ r5044 で表示されます。

値 65535 は、速度制限が有効でないことを示しています。

17.10 必要条件

17.10.1 制御パラメータの変更

位置制御モードにない主軸の場合、マシンデータの変更は主軸が NEWCONF 命令でしていない場合も有効になります。

制御パラメータに変更を加えた場合、新しい値が有効になると速度指令値が不連続変化することがあります。 制御パラメータには以下のようなものがあります。

- MD32200 $MA_POSCTRL_GAIN (サーボゲイン係数)
- MD32210 $MA_POSCTRL_INTEGR_TIME (位置コントローラ積分時間)
- MD32410 $MA_AX_JERK_TIME (軸のジャークフィルタの時定数)

17.11 例

17.11.1 自動ギヤ選択(M40)

例

新規ブロックサーチ変数の内容を説明するために、
自動ギヤ選択(M40)を以下のように仮定します。

S0 ～ 500	1. 第 1 ギヤ選択
S501 ～ 1000	2. 第 2 ギヤ選択
S1001 ～ 2000	3. 第 3 ギヤ選択

システム変数の内容:

$P_SEARCH_S	; 収集された S 値
$P_SEARCH_DIR	; 収集された回転方向
$P_SEARCH_GEAR	; 収集されたギヤ選択

収集	S 値:	回転方向:	ギヤ選択:
	; 0/最終速度	-5	40/最終 GS
N05 G94 M40 M3 S1000	; 1000	3	40
N10 G96 S222	: 222	3	40
N20 G97	; f (PlanAxPosPCS)*	3	40
N30 S1500	; 1500	3	40
N40 SPOS=0**	; 1500	-19	40

```
N50 M19**                ; 1500      -19     40
N60 G94 G331 Z10 S300    ; 300       -19     40
N70 M42                  ; 300       -19     42
N80 M4                   ; 300       4       42
N90 M70                  ; 300       70      42
N100 M3 M40              ; 300       3       40
N999 M30
```

* f (PlanAxPosPCS): この速度は、ワーク座標系内の径方向軸の現在の位置に応じて異なります。

** ($P_SEARCH_SPOS および$P_SEARCH_SPOSMODE がプログラムされます)

17.12 データリスト

17.12.1 マシンデータ

17.12.1.1 NC 別マシンデータ

番号	識別子: $MN_	説明
10192	GEAR_CHANGE_WAIT_TIME	再解析中にギヤ選択切替の確認応答を待機する時間
10714	M_NO_FCT_EOP	リセット後に有効な主軸の M 機能
12060	OVR_SPIND_IS_GRAY_CODE	主軸オーバライドグレーコード
12070	OVR_FACTOR_SPIND_SPEED	主軸オーバライドスイッチの評価
12080	OVR_REFERENCE_IS_PROG_FEED	オーバライド基準速度
12082	OVR_REFERENCE_IS_MIN_FEED	軌跡オーバライドの基準を定義します
12090	OVR_FUNCTION_MASK	オーバライド指定の選択

17.12.1.2 チャネルマシンデータ

番号	識別子: $MC_	説明
20090	SPIND_DEF_MASTER_SPIND	チャネルのメイン主軸の初期設定
20092	SPIND_ASSIGN_TAB_ENABLE	主軸再割付けの有効/無効
20850	SPOS_TO_VDI	SPOS/SPOSA による PLC への M19 の出力
22400	S_VALUES_ACTIVE_AFTER_RESET	リセット後に S 機能有効

17.12.1.3 軸/主軸マシンデータ

番号	識別子: $MA_	説明
30300	IS_ROT_AX	回転軸
30310	ROT_IS_MODULO	モジュロ変換
31044	ENC_IS_DIRECT2	中間ギヤのエンコーダ
31050	DRIVE_AX_RATIO_DENOM	負荷側ギヤ分母
31060	DRIVE_AX_RATIO_NUMERA	負荷側ギヤ分子
31064	DRIVE_AX_RATIO2_DENOM	中間ギヤの分母
31066	DRIVE_AX_RATIO2_NUMERA	中間ギヤの分子
31070	DRIVE_ENC_RATIO_DENOM	検出器ギヤ分母
31080	DRIVE_ENC_RATIO_NUMERA	検出器ギヤ分子
31122	BERO_DELAY_TIME_PLUS	プラス方向の BERO 遅延時間
31123	BERO_DELAY_TIME_MINUS	マイナス方向の BERO 遅延時間
32200	POSCTRL_GAIN	K_V 係数
32800	EQUIV_CURRCTRL_TIME	フィードフォワード制御の電流制御等価時定数
32810	EQUIV_SPEEDCTRL_TIME	フィードフォワード制御の速度制御ループ等価時定数
32910	DYN_MATCH_TIME	ダイナミック応答調整の時定数
34040	REFP_VELO_SEARCH_MARKER	レファレンス点クリープ速度
34060	REFP_MAX_MARKER_DIST	原点マーク距離の監視
34080	REFP_MOVE_DIST	絶対番地化システムのレファレンス点距離/終点

番号	識別子: $MA_	説明
34090	REFP_MOVE_DIST_CORR	レファレンス点オフセット/アブソリュートオフセット、絶対番地
34100	REFP_SET_POS	レファレンス点位置
34200	ENC_REFP_MODE	原点復帰動作モード
35000	SPIND_ASSIGN_TO_MACHAX	機械軸への主軸の割り当て
35010	GEAR_STEP_CHANGE_ENABLE	ギヤ選択切替のタイプ
35012	GEAR_STEP_CHANGE_POSITION	ギヤ選択切替位置
35014	GEAR_STEP_USED_IN_AXISMODE	M70 による軸モードのギヤ選択
35020	SPIND_DEFAULT_MODE	初期主軸モード
35030	SPIND_DEFAULT_ACT_MASK	初期主軸モードの機動
35035	SPIND_FUNCTION_MASK	主軸別機能の設定
35040	SPIND_ACTIVE_AFTER_RESET	リセット後に主軸有効
35090	NUM_GEAR_STEPS	指令されたギヤ選択の数
35092	NUM_GEAR_STEPS2	第 2 ギヤ選択データセット:指令されたギヤ選択の数
35100	SPIND_VELO_LIMIT	最高主軸速度
35110	GEAR_STEP_MAX_VELO[n]	自動ギヤ選択切替の最高速度
35112	GEAR_STEP_MAX_VELO2[n]	第 2 ギヤ選択データセット:自動ギヤ選択切替の最高速度
35120	GEAR_STEP_MIN_VELO[n]	自動ギヤ選択切替の最低速度
35122	GEAR_STEP_MIN_VELO2[n]	第 2 ギヤ選択データセット:自動ギヤ選択切替の最低速度
35130	GEAR_STEP_MAX_VELO_LIMIT[n]	ギヤ選択の最高速度
35135	GEAR_STEP_PC_MAX_VELO_LIMIT[n]	位置制御でのギヤ選択の最高速度
35140	GEAR_STEP_MIN_VELO_LIMIT[n]	ギヤ選択の最低速度
35150	SPIND_DES_VELO_TOL	主軸速度許容値
35160	SPIND_EXTERN_VELO_LIMIT	PLC からの主軸速度制限
35200	GEAR_STEP_SPEEDCTRL_ACCEL[n]	速度制御モードでの加速度
35210	GEAR_STEP_POSCTRL_ACCEL[n]	位置制御モードでの加速度
35212	GEAR_STEP_POSCTRL_ACCEL2[n]	第 2 ギヤ選択データセット:位置制御モードでの加速度

番号	識別子: $MA_	説明
35220	ACCEL_REDUCTION_SPEED_POINT	加速度てい減の速度境界
35230	ACCEL_REDUCTION_FACTOR	加速度てい減
35300	SPIND_POSCTRL_VELO	位置制御起動速度
35350	SPIND_POSITIONING_DIR	原点同期されていない主軸の位置決め回転方向
35400	SPIND_OSCILL_DES_VELO	揺動速度
35410	SPIND_OSCILL_ACCEL	振動加速度
35430	SPIND_OSCILL_START_DIR	振動開始方向
35440	SPIND_OSCILL_TIME_CW	M3 方向の揺動時間
35450	SPIND_OSCILL_TIME_CCW	M4 方向の揺動時間
35500	SPIND_ON_SPEED_AT_IPO_START	主軸指令一致による送り有効
35510	SPIND_STOPPED_AT_IPO_START	主軸停止での送り有効
35550	DRILL_VELO_LIMIT[n]	タッピングの最大速度
35590	PARAMSET_CHANGE_ENABLE	PLC からのパラメータセット指定可能
36060	STANDSTILL_VELO_TOL	「軸/主軸停止」検出速度
36200	AX_VELO_LIMIT	速度監視の検出値

17.12.2 セッティングデータ

17.12.2.1 チャネルセッティングデータ

番号	識別子: $SC_	説明
42600	JOG_FEED_PER_REF_SOURCE	JOG モードの毎回転送り制御
42800	SPIND_ASSIGN_TAB	主軸再割付け
42900	MIRROR_TOOL_LENGTH	反転工具長補正
42910	MIRROR_TOOL_WEAR	工具長補正の摩耗値反転
42920	WEAR_SIGN_CUTPOS	加工平面の摩耗値反転
42930	WEAR_SIGN	すべての摩耗値の反転記号
42940	TOOL_LENGTH_CONST	加工平面を変更するときに工具長コンポーネントの割り当てを保持(G17 ～ G19)

17.12.2.2 軸/主軸別セッティングデータ

番号	識別子: $SA_	説明
43200	SPIND_S	PLC インタフェースを介した主軸起動の速度
43202	SPIND_CONSTCUT_S	PLC インタフェースを介した主軸起動の切削速度
43206	SPIND_SPEED_TYPE	PLC インタフェースを介した主軸起動用
43210	SPIND_MIN_VELO_G25	プログラマブル 主軸速度制限 G25
43220	SPIND_MAX_VELO_G26	プログラマブル 主軸速度制限 G26
43230	SPIND_MAX_VELO_LIMS	プログラマブル主軸速度制限 G96/G961
43235	SPIND_USER_VELO_LIMIT	最高主軸速度
43240	M19_SPOS	M19 による主軸位置決め用の主軸位置
43250	M19_SPOSMODE	M19 による主軸位置決め用の主軸位置決めアプローチモード
43300	ASSIGN_FEED_PER_REF_SOURCE	位置決め軸/主軸の毎回転送り

17.12.3 信号

17.12.3.1 軸/主軸への信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
送り速度オーバライド A ～ H	DB31、... .DBX0.0-7	DB380x.DBX0.0-7
軸/主軸無効	DB31、... .DBX1.3	DB380x.DBX1.3
フォローアップモード	DB31、... .DBX1.4	DB380x.DBX1.4
位置検出器 1	DB31、... .DBX1.5	DB380x.DBX1.5
位置検出器 2	DB31、... .DBX1.6	DB380x.DBX1.6
オーバライド有効	DB31、... .DBX1.7	DB380x.DBX1.7
コントローライネーブル	DB31、... .DBX2.1	DB380x.DBX2.1
主軸リセット/残移動距離削除	DB31、... .DBX2.2	DB380x.DBX2.2
速度/主軸速度制限	DB31、... .DBX3.6	DB380x.DBX3.6
プログラムテスト軸/主軸有効	DB31、... .DBX3.7	DB380x.DBX3.7

信号名	SINUMERIK 840D sl	SINUMERIK 828D
実際のギヤ選択 A ～ C	DB31、... .DBX16.0-2	DB380x.DBX2000.0-2
ギヤ切り替え済み	DB31、... .DBX16.3	DB380x.DBX2000.3
主軸 1 の再原点同期	DB31、... .DBX16.4	DB380x.DBX2000.4
主軸 2 の再原点同期	DB31、... .DBX16.5	DB380x.DBX2000.5
ギヤ切替による回転数監視なし	DB31、... .DBX16.6	DB380x.DBX2000.6
S 値削除	DB31、... .DBX16.7	DB380x.DBX2000.7
主軸の送り速度オーバライド有効	DB31、... .DBX17.0	DB380x.DBX2001.0
位置決め中の主軸の再原点同期 1	DB31、... .DBX17.4	DB380x.DBX2001.4
位置決め中の主軸の再原点同期 2	DB31、... .DBX17.5	DB380x.DBX2001.5
M3/M4 反転	DB31、... .DBX17.6	DB380x.DBX2001.6
PLC による揺動制御	DB31、... .DBX18.4	DB380x.DBX2002.4
揺動有効(揺動速度)	DB31、... .DBX18.5	DB380x.DBX2002.5
揺動回転方向右回り(回転方向を右回りに設定)	DB31、... .DBX18.6	DB380x.DBX2002.6
揺動回転方向左回り(回転方向を左回りに設定)	DB31、... .DBX18.7	DB380x.DBX2002.7
主軸オーバライド A ～ H	DB31、... .DBX19.0-7	DB380x.DBX2003.0-7

17.12.3.2 軸/主軸からの信号

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
回転軸/主軸	DB31、... .DBX60.0	DB390x.DBX0.0
エンコーダ限界周波数超過 1	DB31、... .DBX60.2	DB390x.DBX0.2
エンコーダ限界周波数超過 2	DB31、... .DBX60.3	-
原点確立済み/原点同期済み 1	DB31、... .DBX60.4	DB390x.DBX0.4
原点確立済み/原点同期済み 2	DB31、... .DBX60.5	DB390x.DBX0.5
汎用イグザクトストップで位置決め完了	DB31、... .DBX60.6	DB390x.DBX0.6
精密イグザクトストップで位置決め完了	DB31、... .DBX60.7	DB390x.DBX0.7
軸/主軸停止($n < n_{min}$)	DB31、... .DBX61.4	DB390x.DBX1.4
位置コントローラ有効	DB31、... .DBX61.5	DB390x.DBX1.5
速度コントローラ有効	DB31、... .DBX61.6	DB390x.DBX1.6
電流コントローラ有効	DB31、... .DBX61.7	DB390x.DBX1.7

信号名称	SINUMERIK 840D sl	SINUMERIK 828D
復元済み 1	DB31、... .DBX71.4	DB390x.DBX11.4
復元済み 2	DB31、... .DBX71.5	DB390x.DBX11.5
ギヤ選択指令 A ～ C	DB31、... .DBX82.0-2	DB390x.DBX2000.0-2
ギヤ切替	DB31、... .DBX82.3	DB390x.DBX2000.3
速度制限超過	DB31、... .DBX83.0	DB390x.DBX2001.0
指令速度制限中	DB31、... .DBX83.1	DB390x.DBX2001.1
指令速度増加	DB31、... .DBX83.2	DB390x.DBX2001.2
主軸指令一致	DB31、... .DBX83.5	DB390x.DBX2001.5
右回りの実回転方向	DB31、... .DBX83.7	DB390x.DBX2001.7
リジッドタッピング有効	DB31、... .DBX84.3	DB390x.DBX2002.3
有効な主軸モード同期モード	DB31、... .DBX84.4	DB390x.DBX2002.4
有効な主軸位置決めモード	DB31、... .DBX84.5	DB390x.DBX2002.5
有効な主軸モード揺動モード	DB31、... .DBX84.6	DB390x.DBX2002.6
有効な主軸開ループ制御モード	DB31、... .DBX84.7	DB390x.DBX2002.7
主軸がインポジションに実際に到達	DB31、... .DBX85.5	DB390x.DBX2003.5
主軸の M 機能	DB31、... .DBB86-87	DB370x.DBD0000
主軸の S 機能	DB31、... .DBB88-91	DB370x.DBD0004
センサが使用可能	DB31、... .DBX132.0	DB390x.DBX7000.0
センサ S1 が使用可能(クランプ済み状態)	DB31、... .DBX132.1	DB390x.DBX7000.1
センサ S4 が使用可能(ピストン終端位置)	DB31、... .DBX132.4	DB390x.DBX7000.4
センサ S5 が使用可能(モータシャフトの回転位置)	DB31、... .DBX132.5	DB390x.DBX7000.5
状態値が生成され、速度制限 p5043 が有効です。	DB31、... .DBX133.2	DB390x.DBX7001.2
クランプ済み状態	DB31、... .DBW134	DB390x.DBW7002
アナログ値:クランプ済み状態	DB31、... .DBW136	DB390x.DBW7004
センサ S4 (ピストン終端位置)	DB31、... .DBX138.4	DB390x.DBX7006.4
センサ S5 (モータシャフトの回転位置)	DB31、... .DBX138.5	DB390x.DBX7006.5

V1:送り速度

18

18.1 概略説明

送り速度のタイプ

(G 命令に応じて)輪郭または工具中心点軌跡上の工具オフセットが考慮されていても、送り速度により加工速度(軸速度または軌跡速度)が決定され、すべてのタイプの補間で遵守されます。

次の各タイプの送り速度を使用すれば、さまざまな加工用途に最適な補正を実行できます(旋削加工、フライス加工、穴あけなど)。

- 早送り速度(G0)
- インバースタイム送り(G93)
- 毎分送り(G94)
- 毎回転送り(G95)
- 定切削速度(G96、G961)
- 定速度(G97、G971)
- ねじ切りの送り速度(G33、G34、G35)
- フローティングチャックなしのタッピングの送り速度(G331、G332)
- フローティングチャック付きタッピングの送り速度(G63)
- 面取り/丸み付け FRC、FRCM の送り速度
- ノンモーダル送り速度 FB

送り速度の軸割り付け

送り速度を軸に可変的に割り当てれば、各種技術的な必要条件に合わせて調整できます。

次の方式があります。

- 作業平面と切り込み軸に個別の送り速度
- 軌跡送り速度の可変軸割り付け
- 位置決め軸の送り速度

送り速度の制御

加工中またはテスト目的のために、プログラム指令送り速度を変更すれば、変更した加工条件に応じて調整できます。

- 機械操作パネルを使用して
- 操作パネルを使用して
- PLC を使用して
- プログラム命令毎に

18.2 軌跡送り速度 F

軌跡送り速度 F

軌跡送り速度は、関連軸内の速度成分の幾何学的な合計を表します。このため、軌跡送り速度は、補間軸の個々の動作から生成されます。

初期設定では、プログラム指令されたジオメトリ軸の軸速度が使用されます。FGROUP 命令を使用すれば、軌跡送り速度の計算に他のジオメトリ軸や同期軸を組み込むことができます。

工具オフセットが考慮されていても、軌跡送り速度 F により加工速度が決定され、すべてのタイプの補間で遵守されます。アドレス F のプログラム指令値は、送り速度の新しい F 値または新しいタイプの送り速度をプログラム指令するまで、プログラム中で指定されたままです。

軌跡送り速度 F の有効範囲

参照先: 機能マニュアル G2:「速度、指令/フィードバック回路、閉ループ制御」、「速度 (ページ 405)」の章

PLC インタフェースでの F 値の章

現在の軌跡送り速度の F 値は必ず、補助機能(DB21、... DBB158 ～ 193)のチャネル別 PLC インタフェースに入力されます。

関連するインターフェース信号(信号変更、F 値)については、機能説明書「H2: PLC への補助機能出力 (ページ 475)」を参照してください。

挿入円による送り速度

参照先:
『プログラミングマニュアル、基本編』

内径と外径の各軌跡部分の送り速度

円弧ブロックまたは同じ方向の曲率を持つスプラインブロックと工具径補正が有効になっている(G41/G42)場合、内径または外径の軌跡部分に応じて、プログラム指令送り速度は、中心点軌跡上または輪郭上で動作できます。

このために、次の G 命令のグループが設けられています。

- CFTCP
  プログラム指令送り速度は、中心点軌跡上で動作します。
- CFC
  プログラム指令送り速度は輪郭上で動作します。
- CFCIN
  プログラム指令送り速度は、くぼんだスプラインが指定された輪郭上でのみ動作します。

参照先:
プログラミングマニュアル、基本編

最高工具軌跡速度

関係する直線軸または回転軸が最高速度(MD32000 $MA_MAX_AX_VELO)になると、最高軌跡速度が生成されます。つまり、最高速度が最も低い軸により、最高軌跡速度が決定します。この最高軌跡速度を超えることはできません。

G0 がプログラム指令されている場合、MD32000 $MA_MAX_AX_VELO 制限の結果生じた軌跡速度で移動します。

軌跡軸の制限速度

また、FL[<軸>]命令を使用すると、軌跡軸(ジオメトリ軸と同期軸)の制限速度をプログラム指令できます。

この結果、作業平面と切り込み軸に個別に送り速度をプログラム指令できます。つまり、送り速度は、軌跡関連の補間と切り込み軸に対して指定されます。選択した加工平面に垂直な軸が、切り込み軸として指定されます。切り込み軸毎の送り速度をプログラム指令すれば、軸速度を制限できるため、軌跡速度も制限できます。フレームを使用した座

標回転を組み込んではいけません。つまり、切り込み軸は、標準座標系の軸にしてください。この機能を使用すれば、カッタの正面の切削能力が、カッタの側面にわたる切削能力より低い事実を補正できます。

プログラミング例:

プログラムコード	コメント
...G94...	;送り速度タイプの選択(mm/min)
X30 Y20 F200	;軌跡送り速度= 200 mm/s
FL[Z]=50 Z-30	; Z 軸の最高送り速度:50 mm/s

低分解能エンコーダ

低分解能エンコーダを使用する場合、平滑化されたフィードバック値を使用すると、より連続的なパスまたは軸動作を達成できます。時定数が大きくなればなるほど、フィードバック値の平滑化が向上しますが、移動超過が長くなります。

MD34990 $MA_ENC_ACTVAL_SMOOTH_TIME[<軸>] (現在値の平滑時定数)

平滑化された現在値は以下に使用されます。

- ねじ切り(G33、G34、G35)
- 毎回転送り速度(G95、G96、G97、FPRAON)
- 回転数、実位置、速度の表示

18.2.1 送り速度タイプ G93、G94、G95

有効性

送り速度タイプ G93、G94、G95 は、自動モードで、グループ 1 の G 命令(G0 を除く)に対して有効です。

G94 または G95 は、JOG モードでの移動に使用できます。

参照先:
『機能マニュアル、上級機能』;「手動での移動と手動ハンドルでの移動(H1)」

インバースタイム送り(G93)

ブロックの移動で、送り速度より時間をプログラム指令する方が簡単な場合、インバースタイム送りを使用します。

インバースタイム送りは、次の式から計算されます。

```
F = v / s
```

記号の意味
- F: インバースタイム送り(rpm 単位)
- v: 必要な軌跡速度(単位: mm/min または inch/min)
- s: 距離(単位: mm/inch)

プログラミング例

プログラムコード	コメント
N10 G1 G93 X100 Y200 F2	; プログラムした距離を 0.5 分で移動します。
...	

注記

G41/G42 が動作中の場合は、G93 を使用しないでください。ブロック長がブロックごとに大きく異なる場合は、G93 のブロックごとに、F 値を新たにプログラム指令してください。

毎分送り (G94)

毎分送りは、直線軸または回転軸に関連する次の各単位でプログラム指令されます。

- 標準 mm 単位系では[mm/min、°/min]
- 標準 inch 単位系では[inch/min、°/min]

毎回転送り (G95)

毎回転送りは、メイン主軸に関連する次の単位でプログラム指令されます。

- 標準 mm 単位系では[mm/rev]
- 標準 inch 単位系では[inch/rev]
- 回転軸では[°/rev]

軌跡速度は、次の式に従って、主軸の実速度から計算されます。

```
V = n * F
```

記号の意味
- V: 軌跡速度(mm/min または inch/min 単位)
- n: メイン主軸の速度(rpm 単位)
- F: プログラム指令された毎回転送り速度(mm/rev または inch/rev)

注記

システムが G93、G94 および G95 の各送り速度タイプ間で切り替えを実行すると、プログラム指令 F 値が削除されます。

1 刃当り送り速度

主にフライス加工運転では、毎回転送り速度 F...の代わりに、より一般的に使用されている 1 刃当り送り速度 FZ...をプログラム指令できます

コントロールシステムは、有効な工具補正データ記録に関連する$TC_DPNT (1 回転あたりの刃数)工具パラメータを使用して、プログラムした 1 刃あたり送り速度から、各移動ブロックの有効な毎回転送り速度を計算します。

```
F = FZ * $TC_DPNT
```

使用する機能
- F: 毎回転送り速度(mm/rev または inch/rev)
- FZ: 1 刃当り送り速度(mm/刃または inch/刃単位)
- $TC_DPNT: 工具パラメータ:刃数/rev

例:5 刃のフライスカッター($TC_DPNE = 5)

プログラムコード	コメント
N10 G0 X100 Y50	
N20 G1 G95 FZ=0.02	; 1 刃当り送り速度 0.02 mm/tooth
N30 T3 D1	; 工具を装着して、工具オフセットデータブロックを有効にします。
M40 M3 S200	; 主軸速度 200 rpm
N50 X20	; FZ = 0.02 mm/tooth によるフライス
	; 有効毎回転送り速度:
	; F = 0.02 mm/tooth * 5 teeth/rev = 0.1 mm/rev
	; または:
	; F = 0.1 mm/rev * 200 rpm = 20 mm/min
...	

セッティングデータ

JOG モードでの毎回転送り速度

JOG モードで軸が現在割り当てられているチャネルのメイン主軸を基準とした毎回転送り速度の観点から、軸の特性は NC 別のセッティングデータの設定に依存します。

SD41100 $SN_JOG_REV_IS_ACTIVE、ビット**<x>**

ビット	値	意味
0	0	この特性は、軸のファンクションに応じて、以降のセッティングデータに依存します。 ● **軸/主軸** SD43300 $SA_ASSIGN_FEED_PER_REV_SOURCE ● **回転が有効なフレーム**に対する**ジオメトリ軸** SD42600 $SC_JOG_FEED_PER_REV_SOURCE ● **旋回軸**:SD42600 $SC_JOG_FEED_PER_REV_SOURCE
	1	この軸は、毎回転送り速度で移動します。 **必要条件** 以下が該当する場合、毎回転送り速度なしで移動が実行されます。 ● **軸/主軸** メイン主軸が停止し、**また** SD43300 $SA_ASSIGN_FEED_PER_REV_SOURCE == -3 ● **回転が有効なフレーム**に対する**ジオメトリ軸**メイン主軸が停止し、**また** SD42600 $SC_JOG_FEED_PER_REV_SOURCE == -3 ● **旋回軸**:メイン主軸が停止し、また SD42600 $SC_JOG_FEED_PER_REV_SOURCE == -3
1	0	**早送り**で、移動は毎回転送り速度**を使用して**実行されます。
	1	**早送り**で、移動は毎回転送り速度**を使用しないで**実行されます。
2 1)	0	**ハンドルジョグ**で、移動は毎回転送り速度**を使用して**実行されます。
	1	**ハンドルジョグ**で、移動は毎回転送り速度**を使用しないで**実行されます。
3 1)	0	**DRF ハンドル移動**で、移動は毎回転送り速度**を使用して**実行されます。
	1	**DRF ハンドル移動**で、移動は毎回転送り速度**を使用しないで**実行されます。
1) ビット 1 の設定に加えて、このファンクションが有効です。		

NC/PLC インタフェース信号

- 毎回転送り速度が有効です (**チャネル別**)
  このインタフェース信号は、チャネルで軸が毎回転送り速度を使用して移動することを示します:
  – AUTOMATIC モード:軌跡軸または同期軸
  – JOG モード:ジオメトリ軸
  DB31、... DBX62.2 == 1 (毎回転送り速度が有効)
- 毎回転送り速度が有効です (**軸別**)
  このインタフェース信号は、軸が毎回転送り速度を使用して移動することを示します。
  DB31、... DBX33.2 == 1 (毎回転送り速度が有効)

18.2.2 G96、G961、G962、G97、G971 送り速度のタイプ

周速一定制御(G96、G961)

旋盤で周速一定制御を使用すれば、ワークの作業直径に関係なく、切削条件を一定に保つことができます。この結果、最適な切削能力の範囲内で工具を操作できるため、その寿命が長持ちします。

G96、G961 の選択:

G96、G961 をプログラム指令すると、対応する S 値が、径方向軸端での周速(単位: m/min または ft/min)として解釈されます。加工中にワークの直径が減少すると、定切削速度に到達するまで速度が増加します。

G96、G961 をパートプログラム中で初めて選択するときは、定切削速度を mm/min または ft/min 単位で入力してください。

G96 を使用すると、コントロールシステムは自動的に毎回転送りに切り替わります(G95 と同じです)。つまり、プログラム指令送り速度 F は、mm/rev または in/rev で解釈されます。

G961 をプログラム指令すると自動的に、(G94 と同じように)毎分送りが選択されます。プログラム指令送り速度 F は、mm/min または in/min で解釈されます。

主軸速度の決定

プログラム指令された切削速度(S_{G96} または S_{G961} のどちらか)と径方向軸(半径)の現在の直交座標位置を元に、コントロールシステムは次の式を使用して TCP での主軸速度を算出します。

$$n = \frac{S_{Speed}}{2 * \pi * r}$$

n: 主軸速度

S_{Speed}: プログラム指令された切削速度

π 円周率

r: 半径(距離、回転の中心から TCP まで)

半径の決定時には、以下のように仮定されます。

- WCS の径方向軸位置 0 は、回転の中心を表しています。
- 位置オフセット(オンライン工具オフセット、外部ゼロオフセット、$AA_OFF、DRF オフセット、およびコンパイルサイクル)と連結による位置要素(たとえば、TRAIL のスレーブ軸)は、半径の決定時には考慮されません。

主軸速度の計算ではフレーム(たとえば、SCALE、TRANS、ROT 等のプログラマブルフレーム)が考慮されます。TCP の有効径に変更があった場合には、このフレームが速度変化をもたらすことがあります。

1 つのチャネル内の複数の径方向軸に対応した直径指定と基準軸:

複数の径方向軸を同時または個別に有効にでき、下記のものを起動できます。

- 操作画面での直径指定と表示
- 周速一定制御 G96、G961、G962 に対応した SCC[<軸>]による指定基準軸の割り当て

詳細については『機能マニュアル』の「P1:径方向軸 (ページ 1029)」を参照してください。

例

S_{G96} = 230 m/min

- ここで、r = 0.2 m→n = 183.12 rpm
- ここで、r = 0.1 m→n = 366.24 rpm

⇒ ワークの直径が小さくなればなるほど、速度が増加します。

G96、G961 または G962 の場合、ジオメトリ軸を径方向軸として定義してください。

その位置が、メイン主軸の速度に影響を及ぼす径方向軸は、チャネルマシンデータを使用して定義されます。

MD20100 $MC_DIAMETER_AX_DEF (径方向軸機能によるジオメトリ軸)

G96、G961 または G962 機能では、機械原点と径方向軸のワーク原点が、主軸の回転の中心となっている必要があります。

定速度(G97、G971)

G97、G971 は、「周速一定制御機能」(G96、G961)を解除し、最後に計算した主軸速度を保存します。G97 では、送り速度は、(G95 と同じように)毎回転送りとして解釈されます。G971 をプログラム指令すると、(G94 と同じように)毎分送りが選択されます。送り速度 F は、mm/min または in/min で解釈されます。

G97、G971 が有効になっている場合、S 値をプログラム指令すれば、主軸速度を新たに定義できます。この場合、G96、G961 でプログラム指令された切削速度は変更されません。

G97、G971 を使用すれば、加工のない径方向軸に沿った動作(たとえば、切削工具)での速度変化の発生を回避することができます。

注記

G96、G961 が有効なのはワークの加工中だけです(G1、G2、G3 スプライン補間など、送り速度 F が有効な場合)。

G96、G961 と G0 が有効なブロックの主軸速度の動作は、チャネルマシンデータで定義できます。

MD20750 ALLOW_G0_IN_G96 (G96、G961 の G0 論理)

周速一定制御 G96、G961 を選択すると、ギヤ選択切替を実行できません。

主軸速度オーバライドスイッチは、計算した主軸速度に基づいて動作します。

径方向軸の DRF オフセットは、主軸速度指令値計算に影響を及ぼしません。

(G0 後の)加工開始時、および NC ストップ、G60、G09、...の後には、軌跡開始は「nAct= nSet」を待ちます。

「nAct = nSet」と「指令速度制限中」の各インタフェース信号は、内部速度設定によって変更されません。

速度が最低速度を下回るか、信号「軸/主軸停止」が検出されると、「nAct =nSet」がリセットされます。

起動している軌跡運転(G64、丸み付け)は中断されません。

G96、G961 による主軸速度制限

「周速一定制御」機能に対して最高主軸速度を指定することができます。

- 以下のセッティングデータ:
  SD43230 $SA_SPIND_MAX_VELO_LIMS (G96/G961 の主軸速度制限)
- パートプログラムで(メイン主軸の場合)、プログラミング命令 LIMS を使用して

一番最後に変更された値(LIMS または SD)が有効になります。

LIMS は、G96、G961、G97 により有効になり、1 つのブロック内のパートプログラムで最大 4 つの速度制限まで指定できます。特定の瞬間に有効なメイン主軸の主軸番号 <Sn> = 1、2、3 または 4 は、パートプログラム命令 LM[<Sn>]でプログラムできます。

メインランにこのブロックがロードされると、プログラム指令値はすべて、セッティングデータ SD43230 $SA_SPIND_MAX_VELO_LIMS に伝送されます。

マシンデータ:
MD10710 PROG_SD_RESET_SAVE_TAB[n] (更新するセッティングデータ)に応じて、LIMS で設定された速度制限は、制御装置の電源を切った後も格納されたままです。
G96、G961、G97 を再度有効にすると、この主軸速度制限も有効になります。

G26 またはセッティングデータ:
SD43220 $SA_SPIND_MAX_VELO_G26 (最高主軸速度)
で定義された最高許容主軸速度を超えることはできません。

速度制限(G26 または SD43220 $SA_SPIND_MAX_VELO_G26)の超過を発生させるプログラミングの誤りがあった場合、以下のインタフェース信号が設定されます。

DB31、... DBX83.1 (プログラム指令速度が大きすぎる)

大きな直径の部品で滑らかな回転を確保するために、主軸速度が最低レベルを下回らないようしてください。
この速度は、セッティングデータ(
SD43210 $SA_SPIND_MIN_VELO_G25 (最小主軸速度)
)によって設定できます。また、ギヤ選択に応じて、マシンデータ(
MD35140 $MA_GEAR_STEP_MIN_VELO_LIMIT (ギヤ選択の最小速度)
)により設定できます。

G25 により、パートプログラムで最低主軸速度を変更できます。速度制限(G25 または SD43210 $SA_SPIND_MIN_VELO_G25)の未達を発生させるプログラミングの誤りがあった場合、以下のインタフェース信号が設定されます。

DB31、... DBX83.2 (速度指令値が低すぎます)

主軸速度制限の詳細については、以下を参照してください。『機能マニュアル』の S1: 「主軸」、「主軸監視 (ページ 1665)」。

注記

パートプログラムで G25/G26/LIMS により変更された速度制限値はセッティングデータに設定されるため、プログラム終了後もそのまま保存されます。

ただし、G25/G26/LIMS で変更された速度制限値がプログラム終了後に適用されなくなる場合、以下の定義を工作機械メーカの GUD ブロックに挿入してください。

```
REDEF $SA_SPIND_MIN_VELO_G25 PRLOC
REDEF $SA_SPIND_MAX_VELO_G26 PRLOC
REDEF $SA_SPIND_MAX_VELO_LIMS PRLOC
```

G96、G961 でのメイン主軸の切替

G96、G961 が有効になっているときにメイン主軸を切り替えると、切り替えられる前のメイン主軸の速度が保持されます。これは、G96 から G97 への遷移に相当します。
SETMS により新たに定義されたメイン主軸は、この方法で生成された「周速一定制御」機能を実行します。

アラーム

周速一定制御 G96、G961、G962

- F 値がプログラム指令されていない場合、アラーム 10860 「送り速度がプログラム指令されていません」が出力されます。G0 ブロックでは、このアラームは発生しません。
- 負の軌跡速度をプログラム指令すると、アラーム 14800「ゼロ以下の軌跡速度がプログラム指令されています」が出力されます。
- G96、G961 または G962 が有効になっていて、マシンデータ:
  MD20100 $MC_DIAMETER_AX_DEF (径方向軸機能を持つジオメトリ軸)で径方向軸が定義されていない場合、
  アラーム 10870「径方向軸が未定義」が出力されます。
- G96、G961 が有効になっていて、LIMS プログラム命令により負の最高主軸速度がプログラム指令されている場合、アラーム 14820「負の最高主軸速度が G96、G961 に対してプログラム指令されています」が出力されます。
- 最初に G96、G961 が選択されるときに定切削速度がプログラム指令されていないと、アラーム 10900 「周速一定制御に S 値がプログラム指令されていません」が出力されます。

18.2.3 ねじ切りの送り速度(G33、G34、G35、G335、G336)

18.2.3.1 G33 による送り速度

G33

機能 G33 を使用すれば、一定ピッチを持つねじ山を加工できます。

回転数 S、送り速度 F、ねじピッチ

毎回転送り速度[mm/rev]は、G33 のねじ山に使用されます。 回転送り速度は、ねじのピッチ[mm/rev]をプログラム指令することで定義されます。

ねじの長さに対応する軸の速度は、プログラム指令指令された主軸速度 S とねじピッチにより計算されます。

送り速度 F [mm/min] =回転数 S [rev/min] * ピッチ[mm/rev]

加速カーブの終了時、主軸のフィードバック値(メイン主軸での SPCON にる主軸指令値)と軸指令値間の位置連結が確立されます。 この時点で、(原点マークオフセットを含む)主軸の原点マークに対する軸の位置は、ねじ山開始位置(原点マークに SF を加えたもの)を横切ったときに、ブロックの先頭で軸が瞬間的に加速したかのようになります。 軸の追従誤差に対して補正が実行されます。

最低主軸速度

低速度での滑らかな回転を確保するために、主軸速度は最低レベル未満に設定できません。

この速度は以下によって設定できます。

- セッティングデータを使用
  SD43210 $SA_SPIND_MIN_VELO_G25 (最低主軸速度)
- ギヤ選択毎に、以下のマシンデータを使用:
  MD35140 $MA_GEAR_STEP_MIN_VELO_LIMIT (ギヤ選択切替の最低速度)

G25 により、パートプログラムで最低主軸速度を変更できます。

NC ストップ、シングルブロック

NC ストップおよび(ブロック境界でも)シングルブロックが有効になるのは、ねじ山の連続が完了した後のみです。 連続するすべての G33 ブロックとその後に続く最初の非 G33 ブロックが、1 つのブロックとして移動します。

破損のない早期中止

ねじ切りは、終点に到達する前に破損せずに中止できます。 後退動作を有効にすれば、この運転を実行できます。

ROT フレームによるねじ切り

ROT フレームおよび G33、G34、G35 を使用すると、回転によりねじの長さが変更され、これによりピッチが変更される場合、アラーム 10607「フレームによるねじ切りを実行できません」が発生します。 ねじ切り軸を中心とした回転が実行できます。

アラーム 10607「 フレームによるねじ切りを実行できません」は、ROT 命令がアプリケーションで意図的に使用されている場合、マシンデータ MD11410 $MN_SUPPRESS_ALARM_MASK のビット 12 を設定してマスクすることができます。

他のフレームはすべて、アラームなしで NC によって許容されています。 SCALE のピッチ変化動作に配慮してください。

18.2.3.2 G34 と G35 による単調増加/単調減少の可変ねじピッチ

機能

ねじピッチの増加(G34)は、ピッチ値の数値的な増加を定義します。 ピッチが大きくなると、ワーク上のねじ山間の距離が大きくなります。 そのため、ねじ切り軸の速度は仮定された一定の主軸速度で増加します。

ねじピッチの減少(G35)には、この逆が適用されます。

ねじピッチの変化に対して、以下の定義がおこなわれます。

- G34: ねじピッチの増加は、増加変化に対応します。
- G35: ねじピッチの減少は、減少変化に対応します。

G34 と G35 機能はどちらも G33 の機能を暗示し、ねじのピッチ変化絶対値を F でプログラム指令する選択肢を提供します。ねじの開始と終了ピッチがわかっている場合、ねじピッチ変化は以下の数式を使用して決定できます。

$$F = \frac{|k^2_e - k^2_a|}{2 * l_G}$$

次に意味を示します。

F: プログラム指令されるねじピッチ変化[mm/rev²]

k_e: 軸目標点座標のねじピッチ、ねじ切り軸[mm/rev]

k_a: (I、J または K でプログラム指令された)初期ねじピッチ[mm/rev]

l_G: ねじ長[mm]

必要なピッチ増加または減少に応じて、絶対値 F を G34 または G35 に適用してください。

ねじ長 l_G、ピッチ変化 F と初期ピッチ k_a が分かっている場合、ブロック k_e の終点でのピッチ増加は、次の式を変更すれば、次のように特定できます。

- G34 (ピッチの増加):

$$k_e = \sqrt{k^2_a + F * 2 * l_G}$$

- G35 (ピッチの減少):

$$k_e = \sqrt{k^2_a - F * 2 * l_G}$$

注記

この式を使用した結果、負のルート表現が求められた場合、ねじ山を加工できません。この場合、NC は、アラーム 10605 またはアラーム 22275 を通知します。

用途

G34 と G35 機能を使用すれば、自己剪断ねじ山を生成できます。

例

減少ねじピッチ G35 が指定されたねじ切り G33

プログラムコード	コメント
`N1608 M3 S10`	`;` 主軸速度
`N1609 G0 G64 Z40 X216`	`;` アプローチの開始点

プログラムコード	コメント
N1610 G33 Z0 K100 SF=R14	; 一定のピッチ 100 mm/rev が指定されたねじ山
N1611 G35 Z-220 K100 F17.045455	; ねじピッチ減少 17.045455 mm/rev2
	; ブロック終点での 50 mm/rev のねじピッチ
N1612 G33 Z-240 K50	; 衝撃のないねじブロック移動
N1613 G0 X218	
N1614 G0 Z40	
N1616 M17	

ブロック解析中の監視

G34 が有効になっている場合に、ピッチ変化によりねじ切り軸が過負荷になるか、G35 が有効になっている場合に、ピッチ変化の結果、軸が停止状態になる場合、ブロック解析中に前もって、こうしたピッチ変化がすべて検出されます。 アラーム 10604「ねじピッチ増加が大きすぎます」または 10605「ねじピッチ減少が大きすぎます」が通知されます。

ねじ切り中、一部の実践用途では、主軸速度の修正が必要になります。 この場合、オペレータは、ねじ切り軸の許容速度を基礎として修正を実施します。

これをおこなうために、アラーム 10604 と 10605 の出力を以下のようにマスクすることができます。

MD11410 $MN_SUPPRESS_ALARM_MASK ビット 10 = 1

これで、ブロック解析が正常に続行されます。

実行中に監視

ねじ山が加工される場合(補間)、次の各状況が周期的に監視されます。

- ねじ切り軸の最高速度を超えていないかどうか
- G35 により軸停止状態に到達していないかどうか

監視機能が応答すると、以下のアラームが通知されます。

- アラーム 22269「ねじ切り軸の最高速度に達しました」または
- アラーム 22275「ねじ切り軸のゼロ速度に達しました」

18.2.3.3 G33、G34、および G35 の軸の加速度特性

チャネル別のセッティングデータによって G33、G34、または G35 によりねじ切り/タッピングを設定できる場合のフィード軸の加速度特性:

SD42010 $SC_THREAD_RAMP_DISP[<n>] = <値>

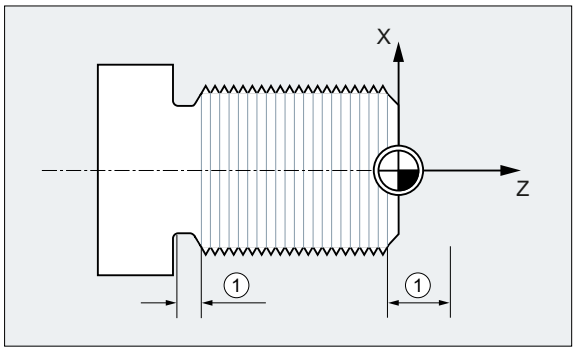
<n>	意味	<値>	意味
0	ねじ切り始めの加速度特性:	< 0 (初期設定: -1)	BRISK/SOFT でプログラムされたねじ切り始めのねじ軸の加速度
		= 0	ねじ切り始めのねじ軸の加速度は急激です(ステップファンクション)(≙ BRISK)。
		> 0	ねじ切り始めの最大距離を指定します。 また、ねじ切り始めの最大距離は、パートプログラムのアドレス DITS で指定することもできます(「G33、G34、および G35(DITS、DITE)でプログラムされた切り始めと切り上げの距離 (ページ 1711)」を参照してください)。ブロックをメインランに挿入するときに、プログラムされた切り始め距離がセッティングデータに伝送されます。 **注** 距離が短すぎる場合、加速時に軸が過負荷になる場合があります。

<n>	意味	<値>	意味
1	ねじ切り上げ時の加速度特性:	< 0 (初期設定: -1)	BRISK/SOFT でプログラムされたねじ切り上げ時のねじ軸の減速
		= 0	ねじ切り上げ時のねじ軸の制動は急激です(ステップファンクション)(≙ BRISK)。
		> 0	ねじ切り上げ時の最大距離を指定します。 また、ねじ切り上げ時の最大距離は、パートプログラムのアドレス DITS で指定することもできます(「G33、G34、および G35(DITS、DITE)でプログラムされた切り始めと切り上げの距離 (ページ 1711)」を参照してください)。ブロックをメインランに挿入するときに、プログラムされた切り上げ時の距離がセッティングデータに伝送されます。 **注記** 距離が短すぎると、軸の加速過負荷が生じることがあります。
2	ねじ結合内のねじ遷移の加速度特性	= -1 (初期設定)	ダイナミック応答の制限を維持するために 2 つのねじブロック間でコーナの丸み付けが可能です。 (たとえば、軸速度ステップ/ジャンプ (MD32310 $MA_MAX_ACCEL_OVL_FACTOR) の過負荷係数の有効性の結果として)機械のダイナミックパフォーマンスでハード遷移が許容される場合、スムージングを使用した丸み付けは実行されません。
		= 0	2 つのねじブロック間のコーナは、丸み付けなしで正確に実行されます。軸は制御ループ命令に従います。
		> 0	予備

18.2.3.4 G33、G34、および G35(DITS、DITE)でプログラムされた切り始めと切り上げの距離

ねじの切り始め距離と切り上げ距離は、パートプログラムで DITS と DITE アドレスにより指定できます。

ねじ切り軸は、指定された距離で加速、減速されます。



① 加工方向に対応した切り始め軌跡と切り上げ軌跡

短い切り始め軌跡

ねじ切り始めの部分で、工具開始カーブの余地がほとんどありません。
このため、DITS を使用して、これをさらに短く指定してください。

短い切り上げ軌跡

ねじの切り上げの段差により、工具の減速カーブのためのスペースがなく、ワークと工具刃先が衝突する可能性を招きます。DITE を使用すると、減速カーブをより短く指定できます。ただし、機械システムの慣性のせいで、まだ衝突が発生することがあります。

対策:より短いねじをプログラム指令して、主軸速度を減速します。

注記

DITE は、ねじ山の最後で、丸み付け間隙として動作します。これにより、軸の移動がなめらかに変化します。

作用

プログラム指令した切り始めと切り上げ軌跡は、軌跡上の加速率だけを増加します。2 つの軌跡の一方の設定が、有効な加速でねじ切り軸が必要とする設定より大きい場合、ねじ切り軸は加速されるか、最高加速度に達すると減速されます。

構文

```
DITS=<値> DITE=<値>
```

意味

DITS:	ねじの切り始め軌跡の定義
DITE:	ねじの切り上げ軌跡の定義
<値>:	DITS と DITE では、位置ではなく、軌跡のみをプログラム指令します。 プログラム指令された切り始め/切り上げ距離は、現在の寸法設定に応じて処理されます(インチ、メトリック)。

例

プログラムコード	コメント
...	
N40 G90 G0 Z100 X10 SOFT M3 S500	
N50 G33 Z50 K5 SF=180 DITS=1 DITE=3	; Z=53 でスムージングを開始します。
N60 G0 X20	

詳細情報

SD42010 $SC_THREAD_RAMP_DISP

DITS や DITE を含むブロックがメインランに挿入された場合、プログラムされた切り始め/切り上げ時の距離がセッティングデータ SD42010 $SC_THREAD_RAMP_DISP に伝送されます。

- SD42010 $SC_THREAD_RAMP_DISP[**0**] = DITS のプログラム指令値
- SD42010 $SC_THREAD_RAMP_DISP[**1**] = DITE のプログラム指令値

最初のねじブロックの前または最初のねじブロック内で切り始め/切り上げ軌跡がプログラム指令されていない場合、セッティンデータの現在の値が使用されます。

チャネル/モードグループ/プログラム終了リセット後の動作

DITS や DITE によって上書きされた SD 42010 値は、チャネル/モードグループ/プログラム終了リセットの後でも有効なままです。

ウォームスタート後の動作

ウォームスタートの場合、セッティングデータは、DITS や DITE による上書きの前に有効だった値にリセットされます(標準動作)。

ただし、ウォーム再起動の後、DITS および DITE によりプログラムされた値が依然として有効である場合、セッティングデータ SD42010 $SC_THREAD_RAMP_DISP をマシンデータ MD10710 $MN_PROG_SD_RESET_SAVE_TAB に含める必要があります。

MD10710 $MN_PROG_SD_RESET_SAVE_TAB[<n>] = 42010

切り始め/切り上げ時の距離が非常に短い場合の特性

切り始め軌跡と切り上げ軌跡の一方または両方が非常に短い場合は、ねじ軸の加減速が設定値より大きくなります。これにより、軸に加減速の過負荷が生じます。

そして、ねじの切り始めに対してアラーム 22280 ("Programmed run-in path too short"(プログラム指令切り始め軌跡が短かすぎます」))が発生します (MD11411 $MN_ENABLE_ALARM_MASK で当該の設定がされている場合)。このアラームは、情報を示すのみであり、パートプログラムの実行には影響を与えません。

下記も参照

G33、G34、および G35 の軸の加速度特性 (ページ 1708)

18.2.3.5 ねじ切り時の高速リトラクト

機能

「ねじ切り時の高速リトラクト」機能を使用すると、次のような状況でも修復が困難な損傷を引き起こさずに、ねじ切りを中断できます。

- NC ストップ (NC/PLC インタフェース信号)
- 自動的に NC ストップを起動するアラーム
- 高速入力の切り替え
  参照先
  プログラミングマニュアル、上級編; 「輪郭からの高速リトラクト」の章

後退動作は以下でプログラム指令できます。

- 後退距離と後退方向(インクレメント)
- 後退位置(アブソリュート)

注記

タッピング

「高速リトラクト」機能は、タッピング(G331/G332)では使用**できません**。

プログラミング

構文

高速リトラクトの設定、後退距離と後退方向による後退動作
G33 ... LFON DILF=<値> LFTXT/LFWP ALF=<値>

高速リトラクトの設定、後退位置による後退動作

POLF[<軸名称>]=<値> LFPOS
POLFMASK/POLFMLIN(<軸 1 名称>,<軸 2 名称>, ...)
G33 ... LFON

ねじ切りの高速リトラクトを解除
LFOF

意味

LFON: ねじ切りの高速リトラクトを設定(G33)

LFOF: ねじ切りの高速リトラクトを解除(G33)

DILF=: 後退軌跡の長さを定義

マシンデータ(MD21200 $MC_LIFTFAST_DIST)の設定値は、DILF をプログラム指令すると、パートプログラムで変更できます。

注:
マシンデータ設定値は常に、NC RESET の後に有効になります。

LFTXT LFWP: 後退方向が、G 命令 LFTXT および LFWP により、ALF と組み合わせて制御されます。

- LFTXT: 後退移動が実行される平面は、軌跡タンジェントと工具方向(初期設定)から計算されます。
- LFWP: 後退移動が実行される平面は、現在有効な作業平面です。

ALF=: この方向は、後退移動の平面で ALF により、ステップ値を使用してプログラム指令します。

LFTXT の場合は、ALF=1 で、工具方向の後退が定義されます。

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

参照先:

ALF のプログラミングオプションの説明は、プログラミングマニュアル 上級編の「輪郭からの高速リトラクトの移動方向」にも記載されています。

LFPOS: POLFMASK または POLFMLIN を使用して宣言された軸を、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。

POLFMASK: 個別にアブソリュート位置へ後退する軸(<軸 1 名称>,<軸 1 名称>, ...)の解放

POLFMLIN: アブソリュート位置へ直線補間で後退する軸の開放

注:

すべての関連軸のダイナミック応答によっては、戻し位置に達する前に直線補間を確立できない場合があります。

POLF[]: インデックスでジオメトリ軸または機械軸のアブソリュート後退位置を定義します。

有効性: モーダル

=<値>: ジオメトリ軸の場合は、割り当てた値はワーク座標系の位置として解釈されます。機械軸の場合は、割り当てた値は機械座標系の位置として解釈されます。

割り当てた値は、インクリメンタル指令としてプログラム指令することもできます。

=IC<値>

<軸名称>: ジオメトリ軸または機械軸の名称

注記

LFON または LFOF は常にプログラム指令できますが、その評価は、ねじ切り(G33)中のみおこなわれます。

注記

POLFMASK/POLFMLIN を伴う POLF は、ねじ切り以外の用途にも使用されます。

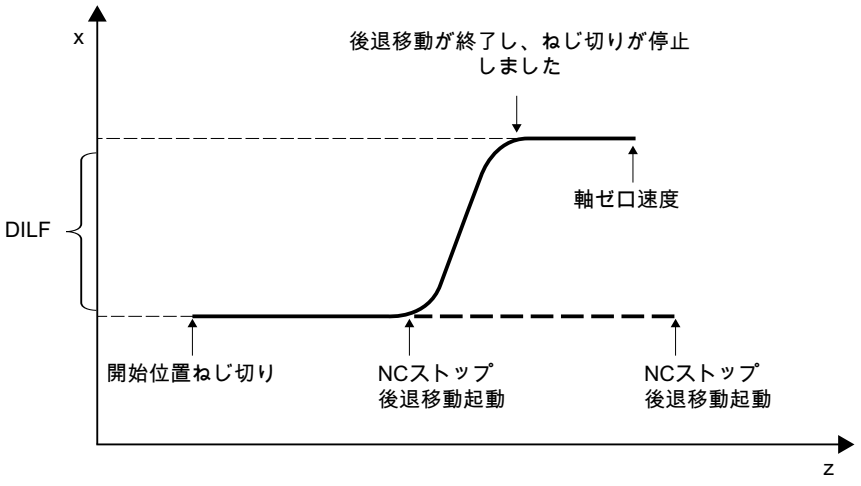


図 18-1　後退動作による G33 の中断

後退動作のダイナミック応答

後退動作は最大の軸ダイナミック応答で実行されます。

- MD32000 $MA_MAX_AX_VELO[<軸>] (速度)
- MD32300 $MA_MAX_AX_ACCEL[<軸>] (加速度)
- MD32431 $MA_MAX_AX_JERK[<軸>] (加々速度)

例

```
プログラムコード                          コメント
N55 M3 S500 G90 G18                      ; 有効な加工平面を設定します。
...
N65 MSG ("thread cutting")
MM_THREAD:
N67 $AC_LIFTFAST=0                       ; ねじの開始前にリセットします。
N68 G0 Z5
```

```
N69 X10
N70 G33 Z30 K5 LFON DILF=10 LFWP      ; ねじ切りの高速リトラクトを有効化します。
ALF=7
                                      ; 後退軌跡= 10 mm
                                      ; イニシャル点 Z/X (G18 により)
                                      ; 後退方向-X (ALF=3、後退方向 +X)
N71 G33 Z55 X15
N72 G1                                ; ねじ切りを選択解除します。
N69 IF $AC_LIFTFAST GOTOB MM_THREAD   ; ねじ切りが中断されている場合。
N90 MSG ("")
...
N70 M30
N55 M3 S500 G90 G0 X0 Z0
...
N87 MSG ("tapping")
N88 LFOF                              ; タッピング前に高速リトラクトを解除します。
N89 CYCLE...                          ; G33 によるタッピングサイクル。
N90 MSG ("")
...
N99 M30
```

電源投入とリセットでの動作

電源投入とリセット後、以下の設定が有効になります。

- 後退動作(LFON /LFOF)と後退方向(LFTXT/LFWP)の初期設定:MD20150 $MC_GCODE_RESET_VALUES
- 後退距離:MD21200 $MC_LIFTFAST_DIST

18.2.3.6 凸型ねじ(G335、G336)

機能

G 命令 G335 と G336 を使用して、凸型ねじ(=円筒形状とは異なる)を旋削できます。これは、自重により機械でたるんでいる極めて大きいコンポーネントの加工に使用されます。このような場合、近軸のねじでは、ねじがコンポーネントの中央で小さくなりすぎます。これを凸型ねじで補正できます。

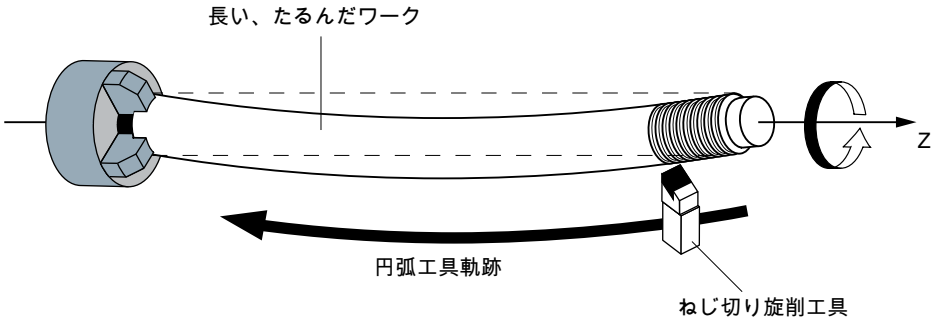


図 18-2 凸型ねじの旋削

プログラミング

凸型ねじの旋削は、G335 または G336 でプログラム指令します。

G335:	**右回り**の円弧工具軌跡での凸型ねじの旋削
G336:	**左回り**の円弧工具軌跡での凸型ねじの旋削

プログラミングは、最初はストレートねじと同様に、パラメータ I、J、および K によって軸のブロック終点とピッチを指定します。

円弧も指定します。G2/G3 と同様に、これは、中心点、半径、開口角度または中間点の指定によってプログラム指令できます。中心点の指定によって凸型ねじをプログラム指令する場合は、以下のことを考慮してください。I、J および K はねじ切りのピッチに使用されるため、中心点の指定での円弧パラメータは、IR=...、JR=...および KR=... を使用してプログラム指令してください。

IR=...:	円弧中心点の X 方向の直交座標
JR=...:	円弧中心点の Y 方向の直交座標
KR=...:	円弧中心点の Z 方向の直交座標

注記

IR、JR、および KR は、凸型ねじ用の補間パラメータの名称の初期設定で、マシンデータ(MD10651 $MN_IPO_PARAM_THREAD_NAME_TAB)で設定できます。

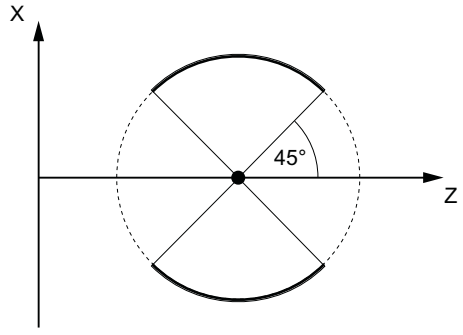
オプションで、起点オフセット SF も指定できます。

構文

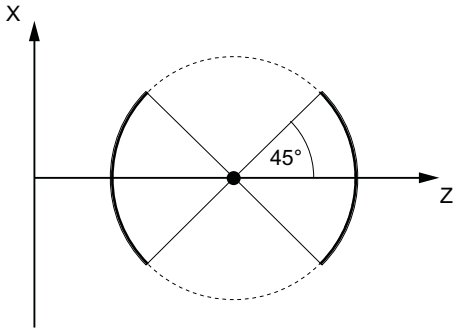
凸型ねじのプログラミングの構文は、一般的には以下のような形式になります。
G335/G336 <軸目標点座標> <ピッチ> <円弧> [<起点オフセット>]

使用可能な円弧領域

G335/G336 でプログラム指令する円弧は、指定されたねじのメイン軸(I、J または K)が、円弧全体に対する円弧でメイン軸を共有する領域にあるようにしてください。



許容される **Z** 軸の領域(K でプログラム指令されるピッチ)



許容される **X** 軸の領域(I でプログラム指令されるピッチ)

次の図で示すようなねじのメイン軸の変更は許容され**ません**。

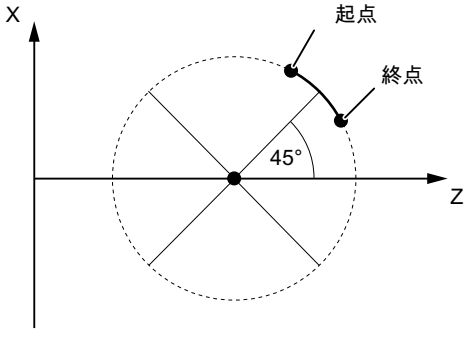


図 18-3　凸型ねじ:許容されない領域

境界条件

フレーム

G335 と G336 は動作中のフレームでも使用できます。ただし、許容される円弧領域が基本座標系(BCS)で維持されることを確認してください。

応答

G335/G336 応答:

- 電源投入/電源切断
- モード変更
- NCK/モードグループ/チャネル/パートプログラム終了リセット
- ブロック検索/REPOS/ASUB
- アラーム/非常停止/誤動作

上記に対する G335/G336 応答は、G33/G34/G35 の動作に対応します。
固有の制限事項はありません。

例

例 1:終点と中心点の指定による右回りの凸型ねじ

プログラムコード	コメント
N5 G0 G18 X50 Z50	; 起点にアプローチします。
N10 **G335 Z100** K=3.5 **KR=25 IR=-20** SF=90	; 右回り方向に凸型ねじを旋削します。

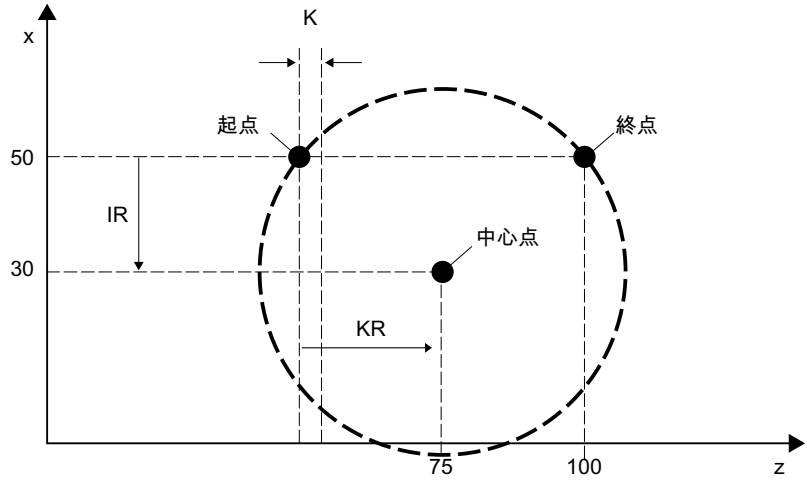


図 18-4　終点と中心点の指定による右回りの凸型ねじ

例 2 :終点と中心点の指定による左回りの凸型ねじ

プログラムコード	コメント
N5 G0 G18 X50 Z50	; 起点にアプローチします。
N10 **G336 Z100** K=3.5 **KR=25 IR=20** SF=90	; 凸型ねじを左回り方向に旋削します。

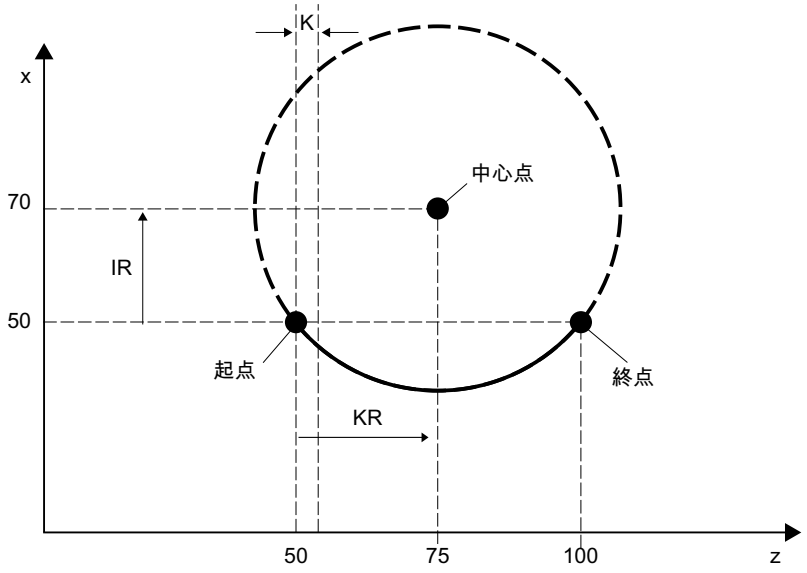


図 18-5　終点と中心点の指定による左回りの凸型ねじ

例 3 :終点と半径の指定による右回りの凸型ねじ

プログラムコード
N5 G0 G18 X50 Z50
N10 **G335 Z100** K=3.5 **CR=32** SF=90

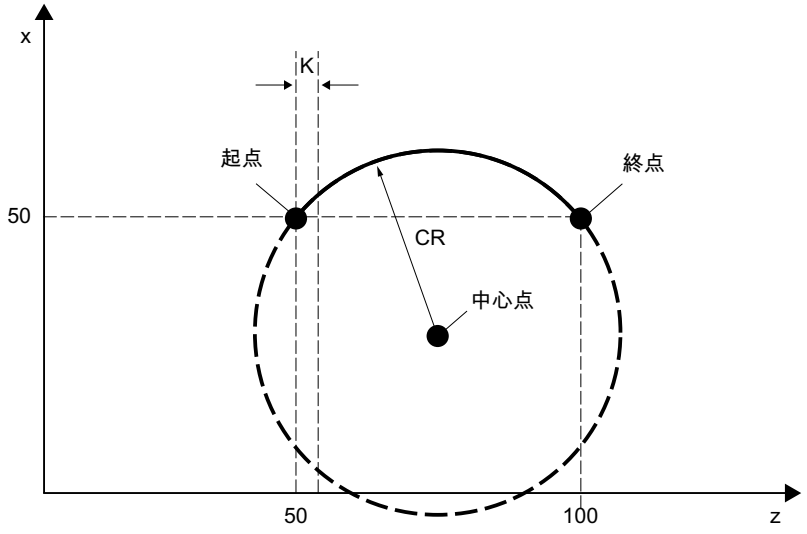


図 18-6　終点と半径の指定による右回りの凸型ねじ

例 4 :終点と開口角度の指定による右回りの凸型ねじ

プログラムコード

```
N5 G0 G18 X50 Z50
N10 G335 Z100 K=3.5 AR=102.75 SF=90
```

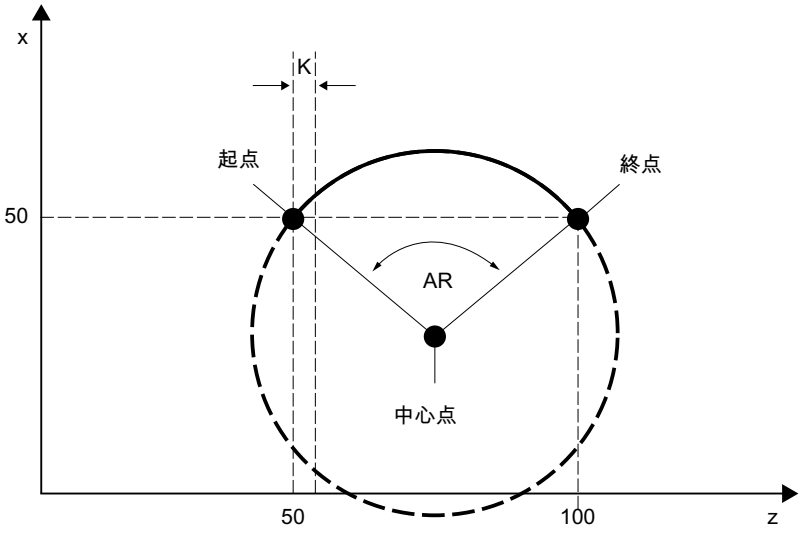


図 18-7　終点と開口角度の指定による右回りの凸型ねじ

例 5 :中心点と開口角度の指定による右回りの凸型ねじ

プログラムコード

```
N5 G0 G18 X50 Z50
N10 G335 K=3.5 KR=25 IR=-20 AR=102.75 SF=90
```

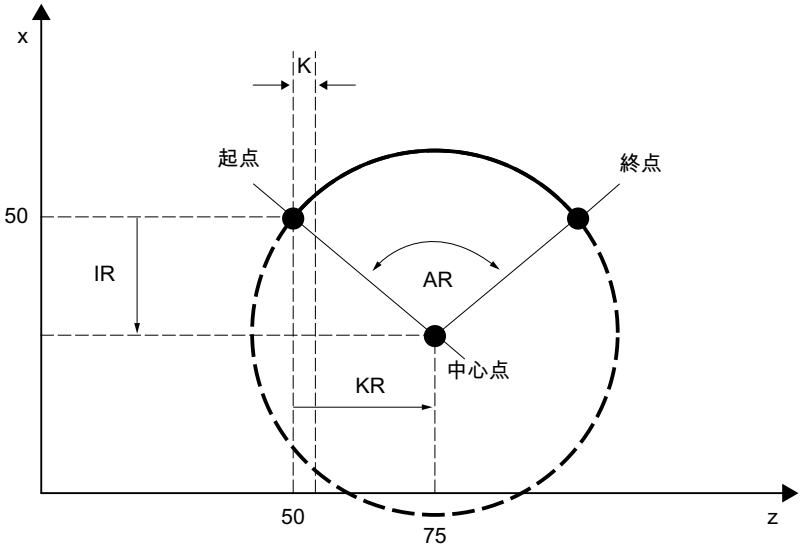


図 18-8　中心点と開口角度の指定による右回りの凸型ねじ

例 6 :終点と中間点の指定による右回りの凸型ねじ

プログラムコード
N5 G0 G18 X50 Z50
N10 **G335 Z100** K=3.5 **I1=60 K1=64**

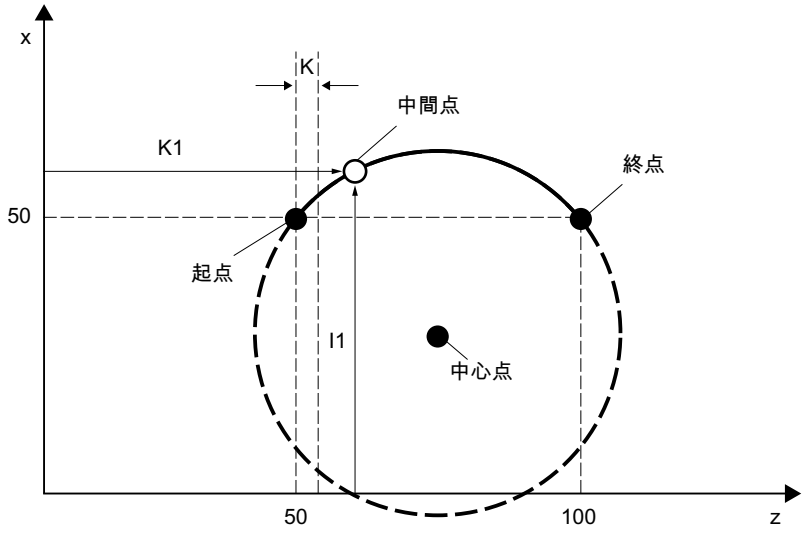


図 18-9　終点と中間点の指定による右回りの凸型ねじ

18.2.4　フローティングチャックなしのタッピングの送り速度(G331、G332)

機能

ねじは、ファンクション G331(タッピング)および G332(タッピング後退)によるリジッドタッピングが可能です。

軌跡送り速度

G331/G332 により、タッピングに関連する軸の軌跡送り速度 F は、有効な主軸速度 S およびプログラムされたピッチから導出されます。

F [mm/min] = S [rpm] * ねじピッチ [mm/U]

必要条件

リジッドタッピングの必要条件は、位置検出器により**主軸が位置制御**されていることです。

マシンデータ

- 停止イベントの防止
  マシンデータで、G331/G332 が有効である場合の停止応答を定義します。
  MD11550 $MN_STOP_MODE_MASK, Bit <x> = <値>

ビット	値	意味
0	0	G331/G332 が有効である場合、軌跡移動またはドウェル時間(G4)中に停止は実行**されません**。
	1	G331/G332 が有効である場合、たとえば、G60 または G4 によって、軌跡制御モードの中断中に停止が実行されます。

G331/G332 が有効である場合、移動動作の停止を常に防止したい場合は、この領域で命令 DELAYFSTON および DELAYFSTOF により停止遅延領域を宣言する必要があります。

参照先:
『機能マニュアル、基本機能』、「停止イベントに及ぼす停止遅延領域の影響 (ページ 692)」の章

注記

(Single BLock)

シングルブロックが停止遅延領域で有効になっている場合は、停止遅延領域外の最初のブロックの終点で NC が停止します。シングルブロックが停止遅延領域より前ですでに選択されている場合は、各ブロック境界で NC が停止します。つまり、停止遅延領域内でも停止します。これにより、停止遅延領域が解除されます。

オーバライドの変更

停止遅延領域より**前**でオーバーライドが変更された場合は、オーバーライドが停止遅延領域**内**で有効になります。

停止遅延領域**内**でオーバーライドが変更された場合は、変更が停止遅延領域より**後**で有効になります。

- オーバライド
  次でリジッドタッピングを設定中に有効になるオーバライド。
  MD12090 $MN_OVR_FUNCTION_MASK, Bit <x> = <値>

ビット	値	意味
0	0	G331/G332 に対して**主軸オーバライド**が有効です 次のマシンデータの設定に応じて、プログラムされた主軸速度または設定された主軸速度制限のいずれかにオーバライドは対応します。 MD12080 $MN_OVR_REFERENCE_IS_PROG_FEED
	1	G331/G332 に対して**軌跡オーバライド**が有効です 次のマシンデータの設定に応じて、プログラムされた軌跡送り速度または設定された軌跡送り速度制限のいずれかにオーバライドは対応します。 MD12082 $MN_OVR_REFERENCE_IS_MIN_FEED **注** これは、プログラムされた軌跡送り速度よりも制限値が小さい場合にのみ適用されます。

必要条件

その他のオーバライド

リジッドタッピング中に次のオーバライドは無効です。

- プログラミング可能な軌跡送り速度オーバライド OVR
- 早送りオーバライド

18.2.5 フローティングチャック付きタッピングの送り速度(G63)

機能

G63 は、フローティングチャックのあるタップを使用してねじ山をタッピングするためのサブファンクションです。 エンコーダ(位置エンコーダ)は、必要ありません。

回転数 S、送り速度 F、ねじピッチ

G63 で、回転数 S を主軸に対して、送り速度 F を切り込み軸(ねじ長手軸)に対してプログラム指令してください。

送り速度 F は、回転数 S とねじピッチに基づいて、プログラマが計算してください。

送り速度 F [mm/min] =回転数 S [rev/min] * ピッチ[mm/rev]

参照先

G63 についての詳細は、『プログラミングマニュアル、基本編』を参照してください。

18.2.6 FGROUP と FGREF

プログラミング

工具またはワーク、あるいはその両方が回転軸によって移動する加工処理(回転チューブのレーザ加工など)では、普通の方法で有効な加工送り速度を、F 値を介した軌跡送り速度としてプログラム指令できなければなりません。

このためには、関連する回転軸それぞれについて有効半径(基準半径)を指定しなければなりません。 この指定は、下記のモーダル NC アドレスをプログラム指令することで可能です。
FGREF[<回転軸>]=<基準半径>

基準半径の単位は、G70/G71/G700/G710 の設定によって異なります。

軌跡送り速度の計算に軸を含めるには、関係する軸をすべて FGROUP 命令で指定しておかなければなりません。

FGREF 指令がされていない動作との互換性を確保するため、システムの電源投入時とリセット時に評価 1 deg = 1 mm が有効になります。

これは、下記の基準半径に対応しています。

FGREF = 360 mm/(2π) = 57.296 mm

この初期設定は、有効な基本単位系(MD10240 $MN_SCALING_SYSTEM_IS_METRIC)、および現在有効な G70/G71/G700/G710 の設定には依存しません。

FGROUP の回転軸用送り速度の重みづけの特記事項

プログラムコード

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

ブロック N110 でプログラム指令された F 値は、回転軸送り速度(deg/min)として解釈されます。一方、ブロック N120 の送り速度の重みづけは、現在のインチ/メトリック設定に応じて 100 in/min または 100 mm/min のいずれかになります。

通知
異なる単位系 そのブロック内でプログラム指令されているのが回転軸のみであっても、FGREF 係数は有効です。 この場合、deg/min 単位の通常の F 値の解釈は、基準半径が FGREF の初期設定に対応する場合のみ適用されます。 • G71/G710 の場合: FGREF[A]=57.296 • G70/G700 の場合: FGREF[A]=57.296/25.4

例

次の例は、軌跡と軌跡送り速度に対する FGROUP の働きを示すものです。 変数$AC_TIME には、秒単位のブロック開始時間が含まれます。 これは、シンクロナイズドアクションでのみ使用できます。

```
プログラムコード                    コメント
N100 G0 X0 A0
N110 FGROUP(X,A)
N120 G91 G1 G710 F100          ; 送り速度= 100 mm/min または 100 度/min
N130 DO $R1=$AC_TIME
N140 X10                       ; 送り速度= 100 mm/min、軌跡= 10 mm、R1 =約 6 秒
N150 DO $R2=$AC_TIME
N160 X10 A10                   ; 送り速度= 100 mm/min、軌跡= 14.14 mm、R2 =約 8 秒
N170 DO $R3=$AC_TIME
N180 A10                       ; 送り速度= 100 度/min、軌跡= 10 度、R3 =約 6 秒
N190 DO $R4=$AC_TIME
N200 X0.001 A10                ; 送り速度= 100 mm/min、軌跡= 10 mm、R4 =約 6 秒
N210 G700 F100                 ; 送り速度= 2540 mm/min または 100 度/min
N220 DO $R5=$AC_TIME
N230 X10                       ; 送り速度= 2540 mm/min、軌跡 = 254 mm、R5 =約 6 秒
N240 DO $R6=$AC_TIME
N250 X10 A10                   ; 送り速度= 2540 mm/min、軌跡= 254.2 mm、R6 =約 6 秒
N260 DO $R7=$AC_TIME
N270 A10                       ; 送り速度= 100 度/min、軌跡= 10 度、R7 =約 6 秒
N280 DO $R8=$AC_TIME
N290 X0.001 A10                ; 送り速度= 2540 mm/min、軌跡= 10 mm、R8 =約 0.288 秒
N300 FGREF[A]=360/(2*$PI)      ; 有効半径を使用して 1 度= 1 インチを設定します
N310 DO $R9=$AC_TIME
N320 X0.001 A10                ; 送り速度= 2540 mm/min、軌跡= 254 mm、R9 =約 6 秒
N330 M30
```

診断

基準半径の読み取り

回転軸の基準半径の値は、以下のシステム変数を使用して読み取ることができます。

- 操作画面での表示、シンクロナイズドアクション、またはパートプログラムでの先読み停止処理の場合は、以下のシステム変数を使用します。

  $AA_FGREF[<軸>]　現在のメインラン値

- 次のシステム変数は、パートプログラムでは先読み停止をおこないません。

  $PA_FGREF[<軸>]　プログラム指令値

値がまったくプログラム指令されていない場合は、既定の 360 mm/(2π) = 57.296 mm (1° あたり 1 mm に対応します)が両方のシステム変数で読み込まれます。

直線軸の場合は、両方のシステム変数の値は常に 1 mm です。

速度に影響する軌跡軸の読み込み

軌跡補間に関係する軸を、次のように、システム変数を使用して読み込むことができます。

- 操作画面での表示、シンクロナイズドアクション、またはパートプログラムでの先読み停止処理の場合は、以下のシステム変数を使用します。

  $AA_FGROUP[<軸>]　指定軸が、初期設定または FGROUP 指令により、現在のメインラン処理の軌跡速度に影響する場合は、値「1」を返します。 それ以外の場合は、値「0」を返します。

  $AC_FGROUP_MASK　FGROUP でプログラム指令され、軌跡速度に影響するチャネル軸のビットキーを返します。

- 次のシステム変数は、パートプログラムでは先読み停止をおこないません。

  $PA_FGROUP[<軸>]　指定軸が、初期設定または FGROUP 指令により軌跡速度に影響する場合は、値「1」を返します。 それ以外の場合は、値「0」を返します。

  $P_FGROUP_MASK　FGROUP でプログラム指令され、軌跡速度に影響するチャネル軸のビットキーを返します。

18.3 位置決め軸の送り速度(FA)

機能

位置決め軸の速度は、軸別送り速度 FA を使用してプログラム指令されます。

FA はモーダルです。

送り速度は、常に G94 です。

注記

最高軸速度(MD32000 $MA_MAX_AX_VELO)を越えていません。

プログラミング

パートプログラムブロック毎に、5 つまでの軸別送り速度をプログラム指令できます。

構文:

```
FA[<位置決め軸>] = <送り速度値>
```

<位置決め軸>: チャネル軸の名称
(MD20080 $MC_AXCONF_CHANAX_NAME_TAB)

<送り速度値>: 送り速度

数値の範囲: 0.001… 999 999.999 mm/min、°/min
または
0.001…39 999.9999 inch/min

初期設定

軸送り速度 FA がプログラム指令されていない場合、軸の初期設定が適用されます。

MD32060 $MA_POS_AX_VELO (位置決め軸速度の初期設定)

PLC への出力

送り速度値を以下のように PLC に出力することができます。

- 以下を介して、チャネル別 NC/PLC インタフェースに:
  DB21、...DBB158 ～ DBB193
- 以下を介して、軸別 NC/PLC インタフェースに:
  DB31、...DBB78 ～ DBB81

出力タイミングは、以下のマシンデータにより指定されます。

MD22240 $MC_AUXFU_F_SYNC_TYPE (F 機能の出力タイミング)

連続軌跡モードで F 機能を NC/PLC インタフェースに出力することにより速度が低下するため、初期設定で出力がマスクされます(MD22240 = 3) 。

詳細については『機能マニュアル』の「H2: PLC への補助機能出力 (ページ 475)」を参照してください。

リセット応答

エンドオブプログラムまたは NC リセット後の動作は、以下のマシンデータで指定されます。

MD22410 $MC_F_VALUES_ACTIVE_AFTER_RESET (リセット後も F 機能が有効)

値	意味
0	初期値が NC リセット後も有効です。
1	最後にプログラム指令された FA 値が NC リセット後も有効です。

18.4 送り速度の制御

18.4.1 送り速度無効および送り速度/主軸停止

機能

「送り無効」または、「送り/主軸停止」は、プログラムされた輪郭が維持されるねじ切り G33 の場合を除いて、減速特性に従って軸を停止状態にします。

チャネル別送り速度無効

チャネルのすべての軸(ジオメトリと付加軸)は、チャネル別の NC/PLC インタフェース信号を使用して、すべてのモードで停止します。

DB21、... DBX6.0 (送り速度無効化)

JOG モードでのジオメトリ軸のチャネル別「送り停止」

チャネルの特定のジオメトリ軸の移動動作は、チャネル別の NC/PLC インタフェース信号を使用して停止します。

- DB21、... DBX12/3 (送り速度停止、ジオメトリ軸 1)
- DB21、... DBX16/3 (送り速度停止、ジオメトリ軸 2)
- DB21、... DBX20/3 (送り速度停止、ジオメトリ軸 3)

インタフェース信号は、**JOG** モードでのみ有効です。

機械軸の軸別「送り速度停止/主軸停止」

機械軸の移動動作は、軸別の NC/PLC インタフェース信号を使用して停止します。

DB31、... DBX4.3 (送り停止/主軸停止)

AUTOMATIC モードおよび MDI モード

AUTOMATIC モードおよび MDI モードでは次のことが該当します。

- **軌跡軸**に対して「送り停止」が行われた場合、現在のブロックで移動されるすべての軸とその軸グループに関与するすべての軸が停止します。
- **位置決め軸**に対して「送り停止」が行われた場合、この軸だけが停止します。

ねじ切り

ねじ切り	有効性
G33、G34、G35	有効 **注** 輪郭の偏差
G331、G332	有効
G63	軸:有効 主軸:**無効**

機械軸に対する軸別の軸/主軸無効

機械軸の移動動作は、軸別の NC/PLC インタフェース信号を使用して停止します。

DB31、... DBX1.3 (軸/主軸無効)

18.4.2 機械操作パネルでの送り速度オーバライド

機能

「機械操作パネルによる送り速度オーバライド」を使用して、オペレータは機械の軌跡送り速度を直接働くパーセント値として、その場で増加または減少できます。これを実現するには、プログラム指令送り速度を NC/PLC インタフェースで提供されるオーバライド値で乗算します。

送り速度は、位置決め軸に対して軸毎に変更できます。

「主軸オーバライド」を使用して、主軸速度と切削速度(G96、G961)を変更できます。

送り速度の変更により、軸の加速度と速度制限値を維持できます。軌跡に沿った輪郭誤差は発生しません。

送り速度オーバライドは、軌跡と位置決め軸に対して別々に変更できます。

オーバライドは、プログラム指令値または制限値(G26、主軸速度の LIMS など)に影響を与えます。

チャネル別送り速度と早送りのオーバライド

送り速度と早送りオーバライドでは、専用の許可信号と補正/オフセット係数は NC/PLC インタフェースで提供されます。

DB21、... DBX6.7 (送り速度オーバライド有効)

DB21、... DBB4 (送り速度オーバライド)

DB21、... DBX6.6 (早送りオーバライド有効)

DB21、... DBB5 (早送りオーバライド)

オーバライド係数は、バイナリフォーマットまたはグレーコードフォーマットのいずれかで PLC から指定できます。このフォーマットは、次のマシンデータによって NC に通知されます。

MD12020 $MN_OVR_FEED_IS_GRAY_CODE (軌跡送り速度オーバライドスイッチがグレーコードされています)

MD12040 $MN_OVR_RAPID_IS_GRAY_CODE (早送りオーバライドスイッチがグレーコードされています)

以下の固定の割り当てが、バイナリコードに適用されます。

バイナリコード	10 進	オーバライド係数
00000000	0	0.00 ≙ 0%
00000001	1	0.01 ≙ 1%
00000010	2	0.02 ≙ 2%
00000011	3	0.03 ≙ 3%
00000100	4	0.04 ≙ 4%
...	...	...
01100100	100	1.00 ≙ 100%
...	...	...
11001000	200	2.00 ≙ 200%

グレーコードの場合、スイッチ位置に対応するオーバライド係数を以下のマシンデータで入力してください。

MD12030 $MN_OVR_FACTOR_FEEDRATE [<n>] (軌跡送り速度のオーバライドスイッチの倍率)

MD12050 $MN_OVR_FACTOR_RAPID_TRA [<n>] (早送りオーバライドスイッチの倍率)

有効な送り速度オーバライドは、現在のチャネルに割り当てられたすべての軌跡軸について機能します。有効な早送りオーバライドは、早送りで移動するすべての軸と現在のチャネルに割り当てられたすべての軸について有効です。

早送りオーバーライドスイッチは使用できません。

専用の早送りオーバライドスイッチがない場合、早送りオーバライドと送り速度オーバライドを切り替えることができます。有効にするオーバライドは、PLC または操作パネ

ルを介して選択できます。早送りオーバライドが有効の場合、送り速度オーバライド値は 100%に制限されます。

- 早送りオーバライドを操作パネルから有効にした場合、PLC 基本プログラム:
  - 早送り用の送り速度オーバライドの選択を、早送りオーバライド用の起動信号で伝送します。
    DB21、... DBX6.6 = DB21、... DBX25.3
  - 送り速度オーバライド値を早送り値で伝送します。
    DB21、... DBB5 = DB21、... DBB4
- 早送りオーバライドを PLC から選択した場合、PLC ユーザープログラム:
  - 早送りオーバライド用起動信号を設定します。
    DB21、... DBX6.6 = 1
  - 送り速度オーバライド値を早送りオーバライド値で伝送します。
    DB21、... DBB5 = DB21、... DBB4

チャネル別の送り速度と早送りオーバライドの有効性:	
• G33、G34、G35 が有効の場合:	無効
• G63 が有効の場合:	無効
• G331、G332 が有効の場合:	無効

軌跡送り速度オーバライドの基準速度

「機械操作パネルからの軌跡送り速度オーバライド」の基準速度を、標準送り速度(=プログラム指令送り速度)に別の設定することができます。

MD12082 $MN_OVR_REFERENCE_IS_MIN_FEED

軸別送り速度オーバーライド

位置決め軸ごとに、送り速度オーバライド係数の 1 つの許可信号と 1 つのバイトが NC/PLC インタフェースで使用できます。

DB31、... DBX1.7 (オーバライド有効)

DB31、... DBB0 (送り速度オーバライド)

オーバライド係数は、バイナリフォーマットまたはグレーコードフォーマットのいずれかで PLC から指定できます。このフォーマットは、次のマシンデータによって NC に通知されます。

MD12000 $MN_OVR_AX_IS_GRAY_CODE (軸送り速度オーバライドスイッチがグレーコードされています)

以下の固定の割り当てが、バイナリコードに適用されます。

バイナリコード	10 進数	オーバライド係数
00000000	0	0.00 ≙ 0%
00000001	1	0.01 ≙ 1%
00000010	2	0.02 ≙ 2%
00000011	3	0.03 ≙ 3%
00000100	4	0.04 ≙ 4%
...	...	...
01100100	100	1.00 ≙ 100%
...	...	...
11001000	200	2.00 ≙ 200%

グレーコードの場合、スイッチ位置に対応するオーバライド係数を以下のマシンデータで入力してください。

MD12010 $MN_OVR_ FACTOR_AX_ SPEED [<n>] (軸送り速度オーバライドスイッチの倍率)

軸別送り速度オーバライドの有効性:	
• G33、G34、G35 が有効の場合:	**無効**
• G63 が有効の場合:	**有効**ではない (オーバライドは NC では固定値として 100%に設定されます)
• G331、G332 が有効の場合:	**有効**ではない (オーバライドは NC では固定値として 100%に設定されます)

主軸オーバライド

主軸ごとに、主軸オーバライド係数の 1 つの許可信号と 1 つのバイトが NC/PLC インタフェースで使用できます。

DB31、... DBX1.7 (オーバライド有効)

DB31, ... DBB19 (主軸オーバライド)

オーバライド係数は、バイナリフォーマットまたはグレーコードフォーマットのいずれかで PLC から指定できます。このフォーマットは、次のマシンデータによって NC に通知されます。

MD12060 $MN_OVR_SPIND_IS_GRAY_CODE (主軸オーバライドスイッチがグレーコードされています)

以下の固定の割り当てが、バイナリコードに適用されます。

バイナリコード	10 進	オーバライド係数
00000000	0	0.00 ≙ 0%
00000001	1	0.01 ≙ 1%
00000010	2	0.02 ≙ 2%
00000011	3	0.03 ≙ 3%
00000100	4	0.04 ≙ 4%
...	...	...
01100100	100	1.00 ≙ 100%
...	...	...
11001000	200	2.00 ≙ 200%

グレーコードの場合、スイッチ位置に対応するオーバライド係数を以下のマシンデータで入力してください。

MD12070 $MN_OVR_FACTOR_SPIND_SPEED [<n>] (主軸オーバライドスイッチの倍率)

「主軸オーバライド」の有効性	
• G33、G34、G35 が有効の場合:	有効
• G63 が有効の場合:	**無効**
• G331、G332 が有効の場合:	有効

主軸オーバライドの基準

主軸オーバライドは、マシンデータまたはセッティングデータで制限された速度またはプログラム指令速度を基準にすることができます。設定は以下でおこなわれます。

MD12080 $MN_OVR_REFERENCE_IS_PROG_FEED (オーバライド基準速度)

オーバライド係数の制限

バイナリコード化されたオーバライド係数の場合、軌跡送り速度、軸送り速度、および主軸速度の最大許容オーバライドを制限することができます。

MD12100 $MN_OVR_FACTOR_LIMIT_BIN (バイナリコードのオーバライドスイッチの制限)

オーバライド有効

有効化されているオーバライドの場合、機械操作パネルから入力された指定されたオーバライド値は、すべての運転モードと運転機能で直ちに有効になります。

オーバライド無効

オーバライドが有効になっていない場合、オーバライド係数 100%が内部的に有効になります。NC/PLC インタフェースのオーバライド係数は評価されません。

例外は、バイナリインタフェースに対するゼロ設定、およびグレーコードされたインタフェースに対する最初のスイッチ設定です。これらの場合、NC/PLC インタフェースに入力されたオーバライド係数が評価されます。バイナリインタフェースの場合、オーバライド係数は常に 0%です。グレーコード化されたインタフェースの場合、1 番目のスイッチ位置値のマシンデータで入力された値がオーバライド値として出力されます。値「0」を割り当てる必要があります。

18.4.3 プログラム可能な送り速度オーバライド

機能

「プログラム可能な送り速度オーバライド」は、パートプログラムによって、軌跡軸と位置決め軸の速度レベルの変更に使用できます。

プログラミング

構文	意味
OVR=<値>	軌跡送り速度 F の送り速度変更
OVRA[<軸>=<値>	位置決め送り速度 FA の送り速度変更

プログラム可能な範囲は、0 から 200%の間です。

初期設定:100%

有効性

NC/PLC インタフェース信号 DB21、... DBB6 (早送りと送り速度オーバライド有効)および DB31、... DBX1.7 (軸別のオーバライド有効)は、プログラム可能な送り速度オーバライドを基準に**しません**。これらの信号が無効になると、プログラム可能な送り速度オーバライドは有効のままになります。

有効なオーバライドは、「プログラム可能な送り速度オーバライド」と「機械操作パネルでの送り速度オーバライド (ページ 1732)」から計算されます。

「プログラム可能な送り速度オーバライド」の初期設定は 100%です。

初期設定は以下の場合に有効になります。

- 送り速度オーバライドがプログラム指令されていない場合、または
- マシンデータ:
  MD22410 $MC_F_VALUES_ACTIVE_AFTER_RESET (リセット後も F 機能が有効)
  が設定されていない場合は、リセット後。

注記

OVR は、G33、G34、G35 では有効になりません。

18.4.4 ドライラン送り速度

機能

ドライラン送り速度は、たとえばプログラムまたはプログラム区間が軌跡送り速度を上昇させて実行できるようにするために、ワークを加工しないでパートプログラムをテストするときに使用されます。

適用

ドライラン送り速度は、自動モードで選択して、PLC または操作パネルから有効にできます。

操作パネルから選択されたとき、インタフェース信号
DB21、... DBX24.6 (ドライラン送り速度選択済み)
が設定され、PLC 基本プログラムから下記のインタフェース信号へ伝送されます。
DB21、... DBX0.6 (ドライラン送り速度起動)

PLC で選択する場合、インタフェース信号 DB21、... DBX0.6 ドライラン送り速度起動) は PLC ユーザープログラムから設定してください。

有効性

「ドライラン送り速度起動」インタフェース信号が設定されている間は、プログラム指令送り速度の代わりに、SD42100 DRY_RUN_FEED で設定された送り速度値が、SD42101 $SC_DRY_RUN_FEED_MODE で指定した方法で有効になります。 (パラメータ設定を参照)

ドライラン送り速度は常に、毎分送り(G94)として解釈されます。

パラメータ設定

ドライラン送り速度の起動

起動のタイミングは、次のマシンデータの設定によって異なります。

MD10704 $MN_DRYRUN_MASK (ドライラン送り速度の起動)

値	意味
0	ドライラン送り速度は、ブロック終点でのみオン/オフできます(初期設定)。
1	また、ドライラン送り速度はプログラムの処理中も起動できます(パートプログラムブロックで)。 **重要:** 処理中の起動により、コントローラで内部的な再解析動作が実施され、それによって軸が少しの間停止します。これは、加工されているワークの面仕上げに影響を与える可能性があります。
2	ドライラン送り速度は、軸を停止させずにいつでも起動/解除することができます。この機能は、「後で」プログラム実行するブロックでのみ有効になります。

ドライラン送り速度の変更

ドライランの送り速度は、セッティングデータで入力します。

SD42100 $SC_DRY_RUN_FEED (ドライラン送り速度)

セッティングデータは、操作パネルを使用して「パラメータ」操作エリアで変更できます。

選択が NC によって受け付けられた場合、以下の NC/PLC インタフェース信号が設定されます。

DB21、... DBX318.6 (ドライラン送り速度有効)

以下の場合、有効なドライラン送り速度を示すために、操作パネルの状態バーに「DRY」が表示されます。

- 選択がブロックの終了時にプログラムストップ中におこなわれた場合、または
- マシンデータ MD10704 $MN_DRYRUN_MASK がプログラムの実行中に「1」に設定された場合。

ドラインラン送り速度の動作モード

SD42100 で入力されたドライラン送り速度の動作モードは、以下のセッティングデータで設定できます。

SD42101 $SC_DRY_RUN_FEED_MODE

値	意味
0	プログラム指令送り速度が SD42100 のドライラン送り速度と比較され、2 つの送り速度のうちの速いほうで移動がおこなわれます(初期設定)。
1	プログラム指令送り速度が SD42100 のドライラン送り速度と比較され、2 つの送り速度のうちの遅いほうで移動がおこなわれます。
2	SD42100 で入力されたドライラン送り速度は、プログラム指令速度とは無関係に直ちに有効になります。
3 - 9	予備
10	設定値 0 と同じ。ねじ切り(G33、G34、G35)とタッピング(G331、G332、G63)を除く。これらの機能は、プログラム指令どおりに実行されます。
11	設定値 1 と同じ。ねじ切り(G33、G34、G35)とタッピング(G331、G332、G63)を除く。これらの機能は、プログラム指令どおりに実行されます。
12	設定値 2 と同じ。ねじ切り(G33、G34、G35)とタッピング(G331、G332、G63)を除く。これらの機能は、プログラム指令どおりに実行されます。

18.4.5 1 ブロックでの複数送り速度値

機能

「1 ブロック複数送り速度値」機能は、外部のディジタル入力やアナログ入力に応じて、NC ブロックの 6 つの異なる送り速度値、ドウェル時間または後退同期動作を有効にするために使用できます。

スパークアウト時間または後退距離の入力が有効にされると、軌跡軸または特定の一軸の残移動距離が削除され、ドウェル時間または後退が開始されます。

後退は IPO サイクル内で開始されます。

信号

機能の入力信号は、1 入力バイトにまとめられています。固定の割り当てが、バイト内で適用されます。

表 18-1　「1 ブロック複数送り速度」機能の入力バイト

	ビット							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
入力番号	I7	I6	I5	I4	I3	I2	I1	I0
送り速度アドレス	F7	F6	F5	F4	F3	F2	ST	SR

I7 ～ I2:送り速度 F7 ～ F2 の適用

E1:ドウェル時間 ST/STA (秒)の適用

I0:後退動作 SR/SRA の適用

信号の優先順位

信号は、I0 を先頭に昇順でスキャンされます。したがって、後退動作(SR)の優先順位は最高であり、送り速度 F7 の優先順位は最低になります。

SR と ST は、F2 - F7 により有効にされた送り速度動作を終了します。
SR は、ST すなわちすべての動作を終了します。

最高優先順位の信号により、現在の送り速度が決定します。

個々の最高優先順位入力(F2 - F7)の欠落に対する動作は、次のマシンデータで定義できます。

MD21230 $MC_MULTFEED_STORE_MASK (「1 ブロック複数送り速度値」機能の保存動作)

ビット	値	意味
2 ... 7	0	個々の最高優先順位入力の欠落により、関連する送り速度は保持されません(初期設定)。
	1	ビット 2 を 7 に設定すると、個々の最高優先順位入力信号で選択された関連する送り速度(F2 ～ F7)は、入力信号の欠落があり、より低い優先順位入力が有効になる場合にも保持されます。

ブロック終了の条件は、以下の場合に満たされます。

- プログラム指令された終点の位置に到達した場合
- 後退動作(SR)が終了した場合
- ドウェル時間(ST)が経過した場合

ハードウェア割り当て

「1 ブロック複数送り速度値」機能の入力バイトは、最高 2 つのディジタル入力バイトまたは NC I/O のコンパレータ入力バイトに割り当てることができます。

MD21220 $MC_MULTFEED_ASSIGN_FASTIN (「1 ブロック複数送り速度値」に対する NC I/O の入力バイトの割り当て)、ビット 0 ... 15

入力ビットは反転することもできます。

MD21220 $MC_MULTFEED_ASSIGN_FASTIN、ビット 16 ... 31

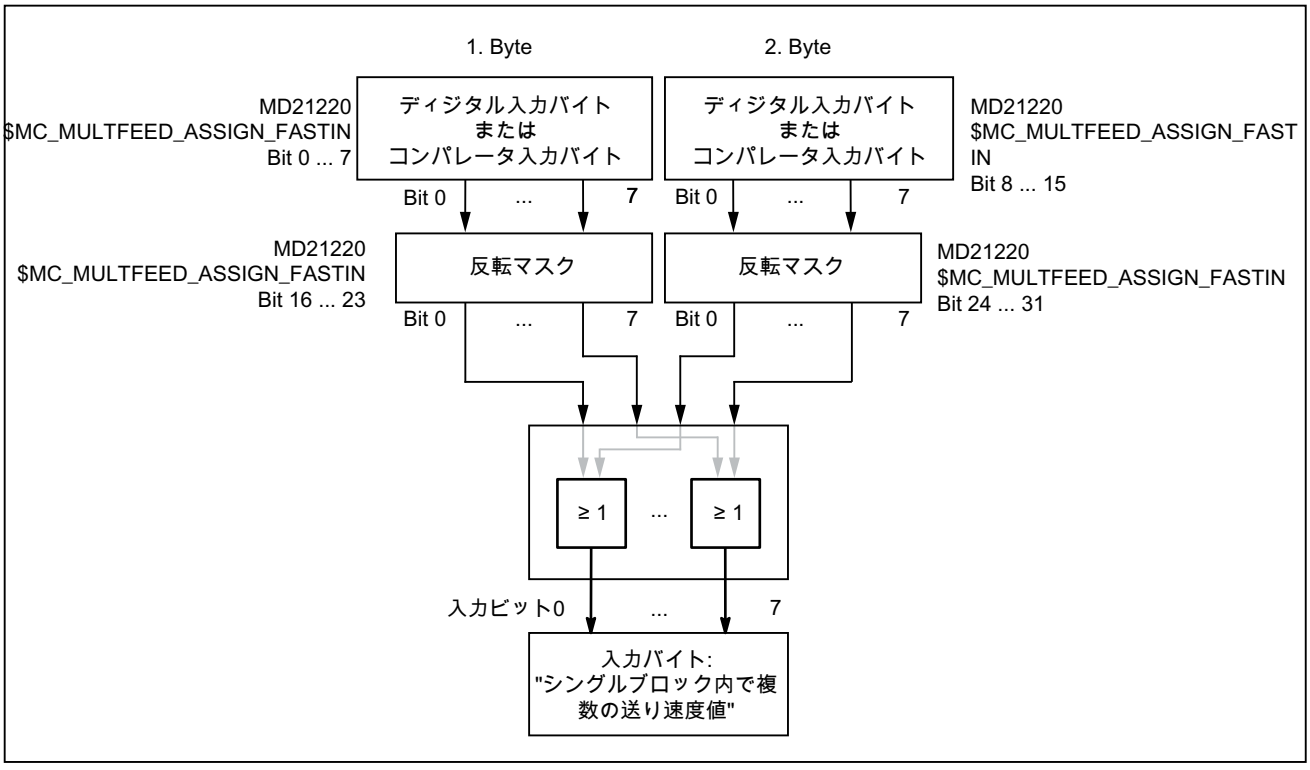


図 18-10 「1 ブロック複数送り速度値」機能の信号割り当て

ディジタル入力の割り当てとコンパレータのパラメータ設定については、以下に記載されています。

参照先:

機能マニュアル、上級機能; A4: ディジタルとアナログ NC I/O

プログラミング

軌跡移動

軌跡送り速度は、アドレス F でプログラム指令され、入力信号が処理されるまで有効です。この値は、モーダルで作用します。

F2=...～ F7=...は、軌跡送り速度に加えて、ブロックでさらに送り速度を 6 つまでプログラム指令するために使用できます。数字の拡張は、変更されたときに送り速度を有効にする入力のビット番号を示しています。

例:

F7=1000 7 は入力ビット 7 に対応します

プログラム指令値は、ノンモーダルで作用します。F でプログラム指令された軌跡送り速度は、次のブロックに適用されます。

ドウェル時間(スパークアウト時間)と後退距離は、ブロックの別のアドレスでプログラム指令されます。

ST=...	ドウェル時間(研削のためのスパークアウト時間)
SR=...	後退距離

これらのアドレスは、ノンモーダルで適用されます。

軸動作

軸送り速度は、アドレス FA でプログラム指令され、入力信号が処理されるまで有効のままです。これらはモーダルで作用します。

FMA[2,<軸>]=...～ FMA[7,<軸>]=... は、ブロックで軸毎にさらに送り速度を 6 つまでプログラム指令するために使用できます。

角かっこ内の最初の値は、変更されたときに送り速度を有効にする入力のビット番号を示しています。2 番目の値は、送り速度が適用される軸を示しています。

例:

```
FMA[3,Y]=1000          ; Y 軸の軸の送り速度、入力ビット 3 に対応
```

FMA のプログラム指令値は、ノンモーダルで作用します。FA のプログラム指令送り速度は、次のブロックに適用されます。

ドウェル時間(スパークアウト時間)と後退距離は、単独軸に対しても定義できます。

STA[<軸>]=...	軸のドウェル時間(スパークアウト時間)
SRA[<軸>]=...	軸の後退軌跡

角かっこ内の拡張は、スパークアウト時間と後退距離が適用される軸を示しています。

例:

```
STA[X]=2.5             ; X 軸のスパークアウト時間は 2.5 秒です。
SRA[X]=3.5             ; X 軸の後退距離は 3.5 (単位 mm など)です。
```

これらのアドレスは、ノンモーダルで適用されます。

注記

後退距離

後退距離の単位は、現在有効な単位系を基準とします(ミリメートルまたはインチ)。

常に逆のストロークが、現在の動作に対して反対の方向で発生します。SR/SRA は、常に逆のストロークの値をプログラム指令します。符号はプログラム指令しません。

注記

POSA の代わりに POS を使用

外部入力を考慮して軸の送り速度、スパークアウト時間(ドウェル時間)、または戻り軌跡をプログラム指令する場合は、このブロックでは、この軸を POSA 軸(複数ブロックにわたる位置決め軸)としてプログラム指令しないでください。

注記

状態確認

さまざまな軸の同期命令に対する入力状態をポーリングすることもできます。

注記

先読み

1 ブロック複数送り速度に対して、先読みも有効です。これにより、現在の送り速度を先読み値によって制限できます。

用途

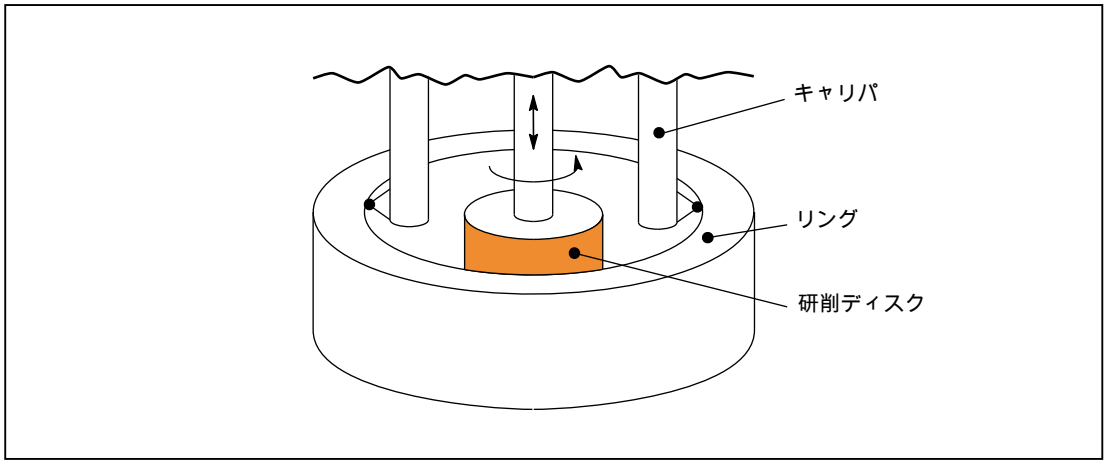
「1 ブロック複数送り速度値」機能は、主として研削加工に使用されますが、これに限定はされません。

たとえば、代表的な用途は次のとおりです。

- アナログまたはディジタルキャリパ
  外部入力がアナログかディジタルかに応じて、さまざまな送り速度値、ドウェル時間と後退距離を有効にできます。制限値は、セッティングデータを介して定義されます。
- 近接スイッチを介した早送り速度から加工送り速度への切り替え

例

実際の直径がキャリパを使用して決定され、条件に応じて、荒削り、仕上げまたは精密仕上げに必要な送り速度が有効になる、ボールベアリングの内部研削です。キャリパの位置により、終点の位置も提供されます。したがって、ブロック終点の基準は、切り込み軸のプログラム指令軸の位置によってだけでなく、キャリパによっても決定されます。




```
G00 X40 F=1000                                          ; 初期設定
G01 X48 F=20 F7=5 F3=2.5 F2=0.5 ST=1.5 SR=0.5
                                                  後退
                                              スパークアウト時間
                                          精密仕上げの送り速度
                                      仕上げ送り速度
                                  荒削りの送り速度
                              空間での研削送り速度
```

SINUMERIK 828D の機能の制限

MD21220=	1H	1 番目のバイト(X242 の 4 ビットおよびインタフェース経由の 4 ビット)
	2H	2 番目のバイト(X252 の 4 ビットおよびインタフェース経由の 4 ビット)
	102H	インタフェース経由で 8 ビットのみ

SINUMERIK 828D では、ハードウェア入力端子が限られているために、バイト全体をアドレス指定することはできません。したがってこれは、以下のことを意味します。

- ハードウェアで 4 種類の送り速度のみをプログラムできます(X242 で 4 ビット、または PPU の X252 で 4 ビット)。
- 残る 4 ビットは、インタフェース経由で実装されます。
- ハードウェア入力ではなく、インタフェースのみが使用されます。したがって、送り速度は、1 回の IPO サイクルで実現しなくなります。

入力またはインタフェースに対する送り速度アドレスの割り当て表:

送り速度アドレス	1 バイト	$A_IN[x]	インタフェース	2 バイト	$A_IN[x]	インタフェース
SR	0	1	DB2800.DBX1.0	0	9	DB2800.DBX1001.0
ST	1	2	DB2800.DBX1.1	1	10	DB2800.DBX1001.1
F2	2	3	DB2800.DBX1.2	2	11	DB2800.DBX1001.2
F3	3	4	DB2800.DBX1.3	3	12	DB2800.DBX1001.3
F4	4	5	DB2800.DBX1.4	4	13	DB2800.DBX1001.4
F5	5	6	DB2800.DBX1.5	5	14	DB2800.DBX1001.5
F6	6	7	DB2800.DBX1.6	6	15	DB2800.DBX1001.6
F7	7	8	DB2800.DBX1.7	7	16	DB2800.DBX1001.7

例:

G90 G0 X600

POS[X]=700 FA[X]=1000

POS[X]=700 FA[**3**,X]=2000 **->** (if $A_IN[**4**]=**1** **->** この場合、送り速度値は 2000 であり、それ以外の場合は値 1000 が送り速度に適用されます。)

試運転:

ハードウェア端子経由の 4 つの送り速度アドレスとインタフェース経由の 4 つの送り速度アドレス

- -X242 = ケーブル IN1 ～ IN4 (PIN 3 ～ 6) -> SR、ST、F2、F3
- インタフェース:DB2800.DBX1.4 ～ 1.7 = IN5 ～ IN8 -> F4、F5、F6、F7
- MD21220 =1H

18.4.6 固定送り速度値

機能

「固定送り速度」機能を使用して、プログラム指令送り速度または設定された JOG 速度の代わりに、マシンデータで定義した固定送り速度(最大 4 個)を起動することができます。

この機能は、AUTOMATIC と JOG モードで使用できます。

自動モードでの特性

輪郭は、プログラム指令送り速度を使用する代わりに、有効にされた固定送り速度で移動します。

JOG モードでの動作

軸が、設定された JOG 速度/JOG 早送り速度の代わりに、固定送り速度で移動します。移動方向は、インタフェース信号を介して指定されます。

パラメータ設定

固定送り速度の設定は、次のようにおこなわれます。

- 直線軸の場合、以下のマシンデータを使用:
  MD12202 $MN_PERMANENT_FEED[<n>]
- 回転軸の場合、以下のマシンデータを使用:
  MD12204 $MN_PERMANENT_ROT_AX_FEED[<n>]

この場合、<n> = 0、1、2、3 (固定送り速度 1、2、3、4 に対応)

注記

固定送り速度は、常に毎分送り速度値です。 毎分送り速度への切り替えは、毎回転送り速度の場合でも内部で実行されます。

起動

固定送り速度は、NC/PLC インタフェース信号で起動されます。

- 軌跡/ジオメトリ軸の AUTOMATIC モードでは、チャネル別インタフェース信号を使用。
  DB21、... DBX29.0 (固定送り速度 1 を有効化)
  DB21、... DBX29.1 (固定送り速度 2 を有効化)
  DB21、... DBX29.2 (固定送り速度 3 を有効化)
  DB21、... DBX29.3 (固定送り速度 4 を有効化)
- 機械軸の JOG モードでは、軸別インタフェース信号を使用。
  DB31、... DBX3.2 (固定送り速度 1 を有効化)
  DB31、... DBX3.3 (固定送り速度 2 を有効化)
  DB31、... DBX3.4 (固定送り速度 3 を有効化)
  DB31、... DBX3.5 (固定送り速度 4 を有効化)

必要条件

有効性

機能「固定送り速度値」は以下に対して有効では**ありません**。

- 主軸
- 位置決め軸
- タッピング時

オーバライド = 0

オーバライド = 0 の移動動作は、以下のマシンデータのセッティングによって異なります。

MD12200 $MN_RUN_OVERRIDE_0

DRF オフセット

DRF オフセットは、選択された固定送り速度に対して有効にできません。

18.4.7 プログラム可能な送り速度特性

機能

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 次パターンは直接プログラム指令することも、スプライン補間としてプログラム指令することもできます。

プログラミング

次の送り速度パターンをプログラム指令できます。

- FNORM
  DIN 66025 に従った動作(初期設定)。
  ブロックでプログラム指令された F 値は、ブロックの軌跡全体にわたって適用され、これ以降、固定モーダル値とみなされます。
- FLIN
  ブロックでプログラム指令された F 値は、ブロック始点にある現在値からブロック終点までの軌跡にわたって直線的に移動し、これ以降、モーダル値とみなされます。
- FCUB
  ノンモーダルにプログラム指令された F 値 (ブロック終点に対する)をスプラインで接続します。このスプラインは、直前または次の送り速度設定に応じて接線方向で開始して終了します。 1 つのブロック内で F アドレスが欠落している場合、最後にプログラムム指令された F 値が使用されます。
- FPO
  F アドレス[構文: `F=FPO(...,...,...)`]は、現在値から、現在値がプログラム指令されたブロック終了まで、多項式を使用して送り速度の特性を指定します。それ以降、終了値はモーダルとして処理されます。

パラメータ設定

FLIN と FCUB を圧縮 COMPON と組み合わせて使用した場合、軌跡送り速度の許容範囲を定義することができます。

MD20172 $MC_COMPRESS_VELO_TOL (圧縮を使用した軌跡送り速度の最大許容誤差)

必要条件

FLIN/FCUB

FLIN または FCUB でプログラム指令された軌跡速度パターンは、G95 の毎回転送り速度ならびに G96/G961 と G97/G971 の周速一定制御と一緒に有効にはなりません。

参照先

プログラム可能な送り速度特性の詳細については、『プログラミングマニュアル、上級編』を参照してください。

18.4.8 面取り/丸み付け FRC、FRCM の送り速度

平面から面取り/丸み付けへの移行中、加工条件が大きく変化することがあります。 このため、面取り/丸み付け輪郭要素では、必要な面品質を実現するために、固有の最適化された送り速度値が必要になります。

機能

面取り/丸み付けの送り速度は、NC アドレスでプログラム指令できます。

プログラミング

構文:

```
... FRC/FRCM=<値>
```

意味

FRC: 面取り/丸み付けの**ノンモーダル**送り速度

FRCM: 面取り/丸み付けの**モーダル**送り速度

<値>: 送り速度は、有効な送り速度のタイプに従って解釈されます。

- G94、G961、G971: mm/inch、inch/min、または °/min 単位の送り速度
- G95、G96、G97: 毎回転送り速度(mm/rev または inch/rev)

注記

FRC は、面取り/丸み付けをブロックにプログラム指令している場合、または RNDM が有効な場合にのみ効果があります。

FRC は、実行中のブロックの F または FRCM 値に上書きします。

FRC のプログラム指令送り速度は、ゼロより大きい値にしてください。

FRCM=0 を設定すると、F でプログラム指令された面取り/丸み付けの送り速度が動作します。

パラメータ設定

前または後のブロックへの面取り/丸み付けの割り当て

送り速度タイプ(G94、G95、G96、G961 ...)および内部フォーマットへの変換は、F と FRC/FRCM のブロック内で矛盾しないようにしてください。 このような状況では、以下のマシンデータを考慮してください。

MD20201 $MC_CHFRND_MODE_MASK (面取り/丸み付け動作)

ビット	値	意味
0	0	面取り/丸み付けのテクノロジ(送り速度、送り速度タイプ、M 命令など)は、後のブロックで決定されます(初期設定)。
	1	面取り/丸み付けのテクノロジは、前のブロックで決定されます(推奨設定)。

最大空きブロック数

有効な面取り/丸み付け時の、移動情報を持つ 2 つのブロック間で許容される、補正平面に移動情報のないブロック(空きブロック)の最大数は制約されています。 最大数は次のマシンデータで指定されます。

MD20200 $MC_CHFRND_MAXNUM_DUMMY_BLOCKS (面取り/丸み付け用の空きブロック)

必要条件

FLIN/FCUB

送り速度補間 FLIN と FCUB は、面取り/丸み付けには指令**できません**。

G0

面取りが G0 で移動する場合、FRC/FRCM は有効ではありません。 プログラミングは、F 値に従ってエラーなしでおこなうことができます。

G94 ↔ G95 の変更

FRCM をプログラム指令した場合は、FRCM 値を、G94 ↔ G95 などの変更時に F と同様に再プログラム指令する必要があります。F のみが再プログラム指令され、変更前の送り速度タイプ FRCM > 0 の場合は、エラーメッセージが出力されます。

例

例 1: MD20201 ビット 0 = 0; 後のブロックの送り速度を使用(初期設定)

プログラムコード	コメント
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; F=100 mm/min による面取り N20-N30
N30 Y10 CHF=4	; FRC=200 mm/min による面取り N30-N40
N40 X20 CHF=3 FRC=200	; FRCM=50 mm/min による面取り N40-N60
N50 RNDM=2 FRCM=50	
N60 Y20	; FRCM=50 mm/min によるモーダル丸み付け N60-N70
N70 X30	; FRCM=50 mm/min によるモーダル丸み付け N70-N80
N80 Y30 CHF=3 FRC=100	; FRC=100 mm/min による面取り N80-N90
N90 X40	; F=100 mm/min によるモーダル丸み付け N90-N100 (FRCM の選択解除)
N100 Y40 FRCM=0	; G95 FRC=1 mm/ rev によるモーダル丸み付け N100-N120
N110 S1000 M3	
N120 X50 G95 F3 FRC=1	
...	
M02	

例 2: MD20201 ビット 0 = 1; 前のブロックの送り速度を使用(推奨設定)

プログラムコード	コメント
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; F=100 mm/min による面取り N20-N30
N30 Y10 CHF=4 FRC=120	; FRC=120 mm/min による面取り N30-N40
N40 X20 CHF=3 FRC=200	; FRC=200 mm/min による面取り N40-N60
N50 RNDM=2 FRCM=50	
N60 Y20	; FRCM=50 mm/min によるモーダル丸み付け N60-N70
N70 X30	; FRCM=50 mm/min によるモーダル丸み付け N70-N80
N80 Y30 CHF=3 FRC=100	; FRC=100 mm/min による面取り N80-N90
N90 X40	; FRCM=50 mm/min によるモーダル丸み付け N90-N100
N100 Y40 FRCM=0	; F=100 mm/min によるモーダル丸み付け N100-N120
N110 S1000 M3	
N120 X50 CHF=4 G95 F3 FRC=1	; G95 FRC=1 mm/ rev による面取り N120-N130
N130 Y50	; F=3 mm/rev によるモーダル丸み付け N130-N140
N140 X60	
...	
M02	

18.4.9 ノンモーダル送り速度 FB

機能

「ノンモーダル送り速度」機能を使用して、1 つのパートプログラムブロックに用途別送り速度を定義できます。このブロックの後は、以前のモーダル軌跡送り速度が再度有効になります。

プログラミング

構文:
```
...FB=<値>
```

意味

FB: 実行中のブロックの用途別送り速度

<値>: 送り速度は、有効な送り速度のタイプに従って解釈されます。

- G94、G961、G971: mm/inch、inch/min、または °/min 単位の送り速度
- G95、G96、G97: 毎回転送り速度(mm/rev または inch/rev)

注記

FB のプログラム指令送り速度は、ゼロより大きい値にしてください。

ブロックに移動動作をプログラム指令していない場合(計算ブロックなど)は、FB は無効となります。

プログラム指令される面取り/丸み付けの明確な送り速度がない場合、FB の値は、このブロックのどの面取り/丸み付け輪郭要素にも適用されます。

FB と、FD (送り速度オーバライドによる手動パルス発生器の移動)または F (モーダル軌跡送り速度)は、同時にプログラミングできません。

18.4.10 単独軸のダイナミック応答に対するの影響

単独軸

単独軸はパートプログラム、シンクロナイズドアクション、または PLC によりプログラム指令できます。

- パートプログラム:
  `POS[<軸>]=...`
  `POSA[<軸>]=...`
  `SPOS[<軸>]=...`
  `SPOSA[<軸>]=...`
  `OS[<軸>]=...`
  `OSCILL[<軸>]=...`
- シンクロナイズドアクション:
  `EVERY ... DO`
  `POS[<軸>]=...`
  `SPOS[<主軸>]=...`
  `MOV[<軸>]=...`
- PLC:
  FC18

ダイナミック応答

軸のダイナミック応答は、以下に影響されます。

- MD32060 $MA_POS_AX_VELO (位置決め軸速度)
  有効な位置決め軸速度は以下で変更できます。
  - パートプログラム/シンクロナイズドアクション軸送り速度 FA または送り速度オーバライド率 OVRA
  - PLC:FRate の指定または軸オーバライドの上書き
- MD32300 $MA_MAX_AX_ACCEL (最高軸加速度)
  有効な最高軸加速度は以下で変更できます。
  - パートプログラムにより間接的に:後続の「マシンデータの起動」でマシンデータを書き込みます
  - パートプログラムにより直接:加減速オーバライド率 ACC
  - シンクロナイズドアクションにより間接的に:マシンデータを書き込み、「マシンデータの起動」ファンクションを起動するために ASUP をトリガします
  - シンクロナイズドアクションにより直接:加減速オーバライド率 ACC (PLC によって事前設定できません)

  PLC により、シンクロナイズドアクションと同じ選択肢が適用されます。

- パートプログラム命令:BRISKA、SOFTA、DRIVEA、JERKA
  シンクロナイズドアクションではプログラムできません(ASUP を使用した間接のみ)。
  PLC により事前設定できません(ASUP を使用した間接のみ)。
- 有効なサーボパラメータセット
  有効なパラメータセットを、以下によって変更できます。
  – パートプログラム/シンクロナイズドアクション SCPARA
  – PLC:DB31、… DBX9.0-2 (制御パラメータセット)
  サーボパラメータセットの詳細は、「位置コントローラのパラメータセット (ページ 455)」を参照してください。

注記

ダイナミック応答の変更

パートプログラムでおこなわれたダイナミック応答の変更は、コマンド軸や PLC 軸の動作に影響を与えません。シンクロナイズドアクションでおこなわれたダイナミック応答の変更は、パートプログラムでプログラム指令された移動動作に影響を与えません。

フィードフォワード制御

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を、FFWON/FFWOF を使用してパートプログラムで直接プログラム指令することができます。シンクロナイズドアクションと PLC では、プログラミングは ASUP を介して間接的にのみおこなえます。

加減速オーバライド率(ACC)

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は、マシンデータ:
MD32300 $MA_MAX_AX_ACCEL (最高軸加速度)
で指定された加速度を ACC 命令を使用して 0%～ 200%の範囲で変更することができます。

構文:
```
ACC[<軸>]=<値>
```

意味

ACC:	加減速オーバライド率のプログラミングのキーワード
<軸>:	チャネル軸または主軸の名称
<値>:	MD32300 に対応したパーセント単位の加速度の変化 値の範囲: 0 ... 200

実際の軸の加速度値は、システム変数$AA_ACC を使用して読み取ることができます。これは、以下により決定されます。

$AA_ACC[<軸>] = (MD32300 $MA_MAX_AX_ACCEL[<軸>]) * ACC[<軸>] / 100

チャネルのリセットまたはパートプログラム M30 の終了のために、MD32320 $MA_DYN_LIMIT_RESET_MASK を使用して、ACC でプログラム指令された値の初期設定を指定できます。

注記

ACC によりプログラムされた加速度オーバライドは、システム変数$AA_ACC を使用して読み取ることができます。ただし、$AA_ACC は、パートプログラムでは、シンクロナイズドアクションで読み取られる場合と異なったタイミングで読み取られます。

加減速オーバライドがシンクロナイズドアクションで ACC をプログラム指令して変更されていない場合、システム変数$AA_ACC には、パートプログラムの ACC を使用したプログラム指令値だけが含まれます。 逆の状況についても同様です。

加減速オーバライド率とメイン軸

システム変数$AA_ACC がパートプログラムで読み取られたか、シンクロナイズドアクションで読み取られたかに応じて、ACC でプログラム指令された加減速オーバライドの値が、NC 軸または メイン軸(コマンド軸、PLC 軸、非同期揺動軸など)に対して出力されます。

正しい結果を得るには、システム変数$AA_ACC を常に同じ場所(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、加減速オーバライドが ACC でプログラム指令された場所から読み取ってください。

例:

パートプログラムでの ACC の書き込み:

```
N80 G01 POS[X]=100 FA[X]=1000 ACC[X]=90 IPOENDA[X]
```

シンクロナイズドアクションでの ACC の書き込み:

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

パートプログラムでの ACC の書き込みと $AA_ACC の読み取り:

```
ACC[X]=50                                          ; 書き込み
RO=$AA_ACC[X]                                      ; 読み取り

IF (RO <> $MA_MAX_AX_ACCEL[X] * 0.5)               ; チェック
```

パートプログラムでの ACC の書き込みと$AA_ACC の読み取り:

```
   SETAL(61000)
ENDIF
```

シンクロナイズドアクションでの ACC の書き込みと$AA_ACC の読み取り:

```
WHEN TRUE DO ACC[X]=25 R0=$AA_ACC[X]                ; 読み取りと書き込み
G4 F1

IF (RO <> $MA_ MAX_AX_ACCEL[X] * 0.25)              ; チェック
   SETAL(61001)
ENDIF
```

単独軸の動作終了の条件

軌跡補間(G601、 G602、G603)のブロック切り替え条件と同様に、個々の軸の移動動作に対する動作終了の条件をパートプログラム/シンクロナイズドアクションでプログラム指令できます。

プログラム命令	動作終了条件
FINEA[<軸>]	「精密イグザクトストップ」
COARSEA[<軸>]	「汎用イグザクトストップ」
IPOENDA[<軸>]	「補間器停止」(IPO 停止)

最も最近のプログラム指令値が、エンドオブプログラムまたは NC リセット後に保管されます。

有効な動作終了条件を、軸別システム変数$AA_MOTEND を使用して読み取ることができます。

注記

システム変数$AA_MOTEND は、パートプログラムで読み取られるか、シンクロナイズドアクションで読み取られるかに対応して、NC 軸またはメイン軸に対する値が含まれます。

例:

パートプログラム:

```
N80 G01 POS[X]=100 FA[X]=1000 ACC[X]=90 COARSEA[X]
```

シンクロナイズドアクション:

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

参照先:

ブロック切り替えおよびFINEA、COARSEA、およびIPOENDAの動作終了の条件についての詳細は、次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)」、:「ブロック切り替え」の章

プログラム可能なサーボパラメータセット(SCPARA)

パートプログラム/シンクロナイズドアクションで、サーボパラメータセットをSCPARAを使用して指定することができます。

構文

```
SCPARA[<軸>] = <パラメータセット番号>
```

意味

SCPARA:	指定されたサーボパラメータの起動のキーワード
<軸>:	チャネル軸の名称
<パラメータセット番号>:	起動するサーボパラメータセットの番号

注記

SCPARAを使用して指定されたパラメータセットの起動を、PLCユーザープログラムからマスクすることができます。

DB31、... DBX9.3 1 (SCPARAによるパラメータセット指定要求無効)

この場合、メッセージは**表示されません**。

有効なパラメータセットの番号を、システム変数$AA_SCPARを使用して読み取ることができます。

必要条件

さまざまな動作終了の条件

さまざまな動作終了の条件が、パートプログラムブロックをすばやく完了するか、ゆっくり完了するかに影響を与えます。これは、テクノロジサイクルとPLCユーザー部分に対して副作用がある可能性があります。

パラメータセットの変更

パートプログラムまたはシンクロナイズドアクション内部と PLC の両方でサーボパラメータセットを変更する場合は、PLC ユーザープログラムを拡張してください。

電源投入

POWER ON 後は、以下の初期設定が実行されます。

- すべての単独軸補間に対する加減速オーバライド率 100%
- すべての単独軸補間に対する動作終了の条件 FINEA
- サーボパラメータセット:1

モード変更

運転モードを AUTOMATIC から JOG に変更したとき、プログラム指令されたダイナミック応答の変更は、有効のままです。

リセット

リセットの場合、最後のプログラム指令値がパートプログラムの指定のために保持されます。メインラン補間の設定は変わりません。

ブロック検索

軸に対してプログラム指令された動作終了の最後の条件が収集され、アクションブロックに出力されます。ブロック検索実行で処理された、プログラム指令された動作終了の条件による最後のブロックが、すべての軸に対するすべてのプログラム指令された動作終了の条件のキャリヤとして機能します。

18.5 必要条件

単位系

有効な送り速度の単位は、設定されている単位系と入力されている軸タイプによって異なります。

MD10240 $MN_SCALING_SYSTEM_IS_METRIC (コントローラの基本単位系メトリック/インチ)

MD30300 $MA_IS_ROT_AX (回転軸または直線軸)

送り速度タイプの初期設定

送り速度タイプの初期設定は、マシンデータで指定します。

MD20150 $MC_GCODE_RESET_VALUES (G グループの初期設定)

初期設定は G94 です。

送り速度タイプの初期設定は、パートプログラムの起動時にのみ表示されます。

リセット後の有効性

最後にプログラム指令された F、FA、OVR、OVRA 値がリセット後も有効かどうかは、次のマシンデータの設定によります。

MD22410 $MC_F_VALUES_ACTIVE_AFTER_RESET (リセット後も F 機能が有効)

主軸位置決め

G95、G96、G961、G97、G971、G33、G34、G35 が有効な場合、主軸の位置決めの後に得られる軌跡送り速度 = 0 であるため、主軸の位置決めはおこなわないでください。プログラム指令された軸位置に到達していない場合、ブロックを完了できません。

18.6 データリスト

18.6.1 マシンデータ

18.6.1.1 NC 別マシンデータ

番号	識別子: $MN_	説明
10651	IPO_PARAM_THREAD_NAME_TAB	凸型ねじの補間パラメータの名前
10704	DRYRUN_MASK	ドライラン送り速度の適用
10710	PROG_SD_RESET_SAVE_TAB	更新するセッティングデータ
11410	SUPPRESS_ALARM_MASK	特定のアラームを抑制するためのマスク
11550	STOP_MODE_MASK	停止動作を定義
12000	OVR_AX_IS_GRAY_CODE	軸送り速度オーバライドスイッチ、グレーコード
12010	OVR_FACTOR_AX_SPEED	軸送り速度オーバライドスイッチの評価
12020	OVR_FEED_IS_GRAY_CODE	軌跡送り速度オーバライドスイッチ、グレーコード

番号	識別子: $MN_	説明
12030	OVR_FACTOR_FEEDRATE	軌跡送り速度オーバライドスイッチの評価
12040	OVR_RAPID_IS_GRAY_CODE	早送りオーバライドスイッチ、グレーコード
12050	OVR_FACTOR_RAPID_TRA	早送りオーバライドスイッチの評価
12060	OVR_SPIND_IS_GRAY_CODE	主軸オーバライドスイッチ、グレーコード
12070	OVR_FACTOR_SPIND_SPEED	主軸オーバライドスイッチの評価
12080	OVR_REFERENCE_IS_PROG_FEED	オーバライド基準速度
12082	OVR_REFERENCE_IS_MIN_FEED	軌跡オーバライドの基準を定義します
12090	OVR_FUNCTION_MASK	オーバライド指定の選択
12100	OVR_FACTOR_LIMIT_BIN	バイナリコードによるオーバライドスイッチの制限
12200	RUN_OVERRIDE_0	オーバライド 0 による移動
12202	PERMANENT_FEED	直線軸の固定送り速度
12204	PERMANENT_ROT_AX_FEED	回転軸の固定送り速度

18.6.1.2 チャネルマシンデータ

番号	識別子: $MC_	説明
20100	DIAMETER_AX_DEF	径方向軸動作のあるジオメトリ軸
20150	GCODE_RESET_VALUES	G グループの初期設定
20172	COMPRESS_VELO_TOL	圧縮のための軌跡速度からの最大許容偏差
20200	CHFRND_MAXNUM_DUMMY_BLOCKS	面取り/半径のあるダミーブロック
20201	CHFRND_MODE_MASK	面取り/丸み付けの動作
20750	ALLOW_GO_IN_G96	G96、G961 の G0 ロジック
21200	LIFTFAST_DIST	輪郭からの高速リトラクトの移動距離
21220	MULTFEED_ASSIGN_FASTIN	「1 ブロック複数の送り速度値」に対する NC I/O の入力バイトの割り当て
21230	MULTFEED_STORE_MASK	「1 ブロック複数送り速度値」機能に対する保持動作
22240	AUXFU_F_SYNC_TYPE	F 機能の出力タイミング
22410	F_VALUES_ACTIVE_AFTER_RESET	リセット後に F 機能有効

18.6.1.3 軸/主軸マシンデータ

番号	識別子: $MA_	説明
30300	IS_ROT_AX	回転軸/主軸
32000	MAX_AX_VELO	最大軸速度
32060	POS_AX_VELO	位置決め軸速度の初期設定
32300	MAX_AX_ACCEL	軸加減速度
32320	DYN_LIMIT_RESET_MASK	応答性制限のリセット応答
34990	ENC_ACTIVAL_SMOOTH_TIME	フィードバック値の平滑時定数
35100	SPIND_VELO_LIMIT	最高主軸速度
35130	GEAR_STEP_MAX_VELO_LIMIT	ギヤ選択の最高速度
35140	GEAR_STEP_MIN_VELO_LIMIT	ギヤ選択の最低速度
35160	SPIND_EXTERN_VELO_LIMIT	PLC による主軸速度制限

18.6.2 セッティングデータ

18.6.2.1 チャネルセッティングデータ

番号	識別子: $SC_	説明
42000	THREAD_START_ANGLE	ねじの開始角度
42010	THREAD_RAMP_DISP	ねじ切り時の軸の加速動作
42100	DRY_RUN_FEED	ドライラン送り速度
42101	DRY_RUN_FEED_MODE	ドライラン送り速度モード
42110	DEFAULT_FEED	軌跡速度の初期値
42600	JOG_FEED_PER_REV_SOURCE	JOG モードでの毎回転送り速度制御
43300	ASSIGN_FEED_PER_RES_SOURCE	位置決め軸/主軸の毎回転送り速度

18.6.2.2 軸/主軸セッティングデータ

番号	識別子: $SA_	説明
43210	SPIND_MIN_VELO_G25	プログラム指令された主軸速度制限 G25
43220	SPIND_MAX_VELO_G26	プログラム指令された主軸速度制限 G26
43230	SPIND_MAX_VELO_LIMS	G96 による主軸速度制限

18.6.3 信号

18.6.3.1 チャネルへの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
ドライラン送り速度起動	DB21、... .DBX0.6	DB320x.DBX0.6
送り速度オーバライド	DB21、... .DBB4	DB320x.DBB4
早送りオーバライド	DB21、... .DBB5	DB320x.DBB5
送り無効	DB21、... .DBX6.0	DB320x.DBX6.0
早送りオーバライド有効	DB21、... .DBX6.6	DB320x.DBX6.6
送り速度オーバライド有効	DB21、... .DBX6.7	DB320x.DBX6.7
送り停止、ジオメトリ軸 1	DB21、... .DBX12.3	DB320x.DBX1000.3
送り停止、ジオメトリ軸 2	DB21、... .DBX16.3	DB320x.DBX1004.3
送り停止、ジオメトリ軸 3	DB21、... .DBX20.3	DB320x.DBX1008.3

18.6.3.2 チャネルからの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
ドライラン送り速度選択	DB21、... .DBX24.6	DB170x.DBX0.6
早送り速度オーバーライド選択	DB21、... .DBX25.3	DB170x.DBX1.3
軌跡/ジオメトリ軸の固定送り速度 1 の有効化	DB21、... .DBX29.0	DB320x.DBX13.0
軌跡/ジオメトリ軸の固定送り速度 2 の有効化	DB21、... .DBX29.1	DB320x.DBX13.1
軌跡/ジオメトリ軸の固定送り速度 3 の有効化	DB21、... .DBX29.2	DB320x.DBX13.2

信号名	SINUMERIK 840D sl	SINUMERIK 828D
軌跡/ジオメトリ軸の固定送り速度 4 の有効化	DB21、... .DBX29.3	DB320x.DBX13.3
ドライラン送り速度有効	DB21、... .DBX318.6	DB330x.DBX4002.6

18.6.3.3 軸/主軸への信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
送り速度オーバライド/主軸オーバライド	DB31、... .DBB0	DB380x.DBB0
オーバライド有効	DB31、... .DBX1.7	DB380x.DBX1.7
機械軸の固定送り速度 1 の有効化	DB31、... .DBX3.2	DB380x.DBX3.2
機械軸の固定送り速度 2 の有効化	DB31、... .DBX3.3	DB380x.DBX3.3
機械軸の固定送り速度 3 の有効化	DB31、... .DBX3.4	DB380x.DBX3.4
機械軸の固定送り速度 4 の有効化	DB31、... .DBX3.5	DB380x.DBX3.5
送り停止/主軸停止	DB31、... .DBX4.3	DB380x.DBX4.3

18.6.3.4 軸/主軸からの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
毎回転送り速度有効	DB31、... .DBX62.2	DB390x.DBX2.2
位置決め軸用 F 機能	DB31、... .DBB81	-
プログラムされた速度が速すぎる	DB31、... .DBX83.1	DB390x.DBX2001.1

W1:工具補正

19

19.1 概略説明

工具補正データの計算

SINUMERIK 840D sl コントローラを使用して、以下の工具補正データを計算できます。

- 長さ補正
- 径補正
- フレキシブルな工具補正メモリへの工具データの保存
  - 0 ～ 32000 の T 番号を使用する工具識別
  - 最大 9 つの刃先を持つ工具の定義
  - 最大 25 の工具パラメータによって記述される刃先
- 工具選択方法: 即時または選択可能な M 機能を介して
- 工具径補正:
  - 選択と解除方法の設定: 法線または輪郭関連
  - すべての補間タイプで有効な補正:
    直線
    円弧
    ヘリカル
    スプライン
    多項式
    インボリュート曲線
  - 外側コーナでの補正の選択方法:
    円弧/楕円の挿入(G450)または等距離での交点(G451)
  - 輪郭に対する G450/G451 機能のパラメータによる調整
  - G450 と DISC パラメータを使用する外側コーナでの任意の移動
  - 補正平面で選択可能な軸動作のないダミーブロックの数
  - 選択可能な衝突監視:
    以下の場合は、予測により経路干渉が検出されます。
    - 軌跡が工具径より短い
    - 内側コーナの幅が工具直径より短い
  - 工具径補正を一定に保つ
  - 多項式用の交点処理

旋回工具ホルダ

この機能を使用すると、工具長補正もできる、斜面の加工が可能です。ただし、工具ホルダのキネマテック (NC 軸なし)を使用して静的な工具の向きの調整が可能な場合に限ります。 この場合、さらに複雑な 5 軸座標変換は不要です。

参照先:
『機能マニュアル、応用機能』;「多軸座標変換(F2)」

工具データと工具ホルダデータを適切に選択すると、コントローラのキネマテックを記述できるため、工具長補正を考慮に入れることができます。 コントローラが、現在のフレームから直接、記述データを取得できます。

注記

工具と工具補正についての詳細情報および、工具補正(TLC と TRC)の一般的または個別のプログラミング機能のすべての技術的な説明については、次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』

フラット/固有 D 番号構造

管理機能によって固有の D 番号を使用して、補正を選択できます。

工具補正の特殊処理

次のセッティングデータを使用して、工具の長さと摩耗の符号の取り扱いを制御できます。

SD42900 $SC_MIRROR_TOOL_LENGTH (ミラーリング時の工具長符号変更)

SD42960 $SC_TOOL_TEMP_COMP (工具に関する熱変異補正)

同じことが、セッティングデータを使用した、ジオメトリ軸のミラーリングや加工平面の変更の際の摩耗成分の応答に当てはまります。

参照先:
『プログラミングマニュアル、基本編』、「工具オフセット」

G461/G462

工具径補正の起動と解除によって、特定の状況で内側コーナの確実な加工ができるように、G461 命令と G462 命令が導入されており、アプローチ/後退方法が工具径補正向けに拡張されています。

- G461
  最新の TRC ブロックとその前のブロックの間の交点がみつからない場合、コントローラは、その中心点が未補正ブロックの終点と一致し、半径が工具径に等しい円を使用して、このブロックのオフセット曲線を拡張することで、交点を計算します。
- G462
  最新の TRC ブロックとその前のブロックの間の交点がみつからない場合、コントローラは、工具径補正を使用して、最後のブロックの終点で直線を挿入することで、交点を計算します(ブロックはその終点の接線によって拡張されます)。

G40 から G41/42 への変更

G40 から G41/G42 への変更およびその逆の変更は、関連するコントロールポイントを持つ工具の工具変更としては処理されません(旋削工具と研削工具)。

工具補正関連機能

工具データの現在の状態に関連する以下の処理を可能にする機能を SW 7.1 で使用できます。

- 保存
- 削除
- 読み取り
- 変更

一部の機能は以前、計測サイクルに実装されていました。 しかし現在は、一般的に使用できます。

別の機能を使用して、横軸、縦軸、垂直軸に対する有効な工具の工具長の割り当てに関する情報を特定できます。

19.2 工具

19.2.1 一般情報

工具選択

プログラム内で工具を選択するには、T 機能を使用します。

T 機能の実行時に新しい工具が直ちにロードされるかどうかは下記のマシンデータの設定によって決まります。

MD22550 $MC_TOOL_CHANGE_MODE (M 機能による新しい工具補正)

工具の直接交換

MD22550 $MC_TOOL_CHANGE_MODE = 0 (M 機能による新しい工具補正)

T 機能によって新規工具がすぐに交換されます。

この設定は、主に工具リボルバ付きの旋盤に使用します。

M06 による工具交換

MD22550 $MC_TOOL_CHANGE_MODE = 1 (M 機能による新しい工具補正)

T 機能によって新規工具の交換が準備されます。

この設定は、加工処理を中断することなく新しい工具を工具交換位置に配置するために、主に工具マガジン付きのフライス盤に使用されます。

下記のマシンデータに入力された M 機能を使用して、古い工具が主軸から取り外され、新しい工具が主軸にロードされます。

- MD22560 $MC_TOOL_CHANGE_M_CODE(工具交換用 M 機能)

この工具交換は、DIN 66025 に準じて、M 機能 M06 を使用してプログラム指令しなければなりません。

次の工具は、下記のマシンデータによって事前に選択されます。

- MD20121 $MC_TOOL_PRESEL_RESET_VALUE (リセット時の事前選択工具)

この場合、RESET と電源投入の際に、次のマシンデータに従って工具長補正値を考慮してください。

- MD20110 $MC_RESET_MODE_MASK (RESET/パートプログラム終了後のシステム初期設定の定義)

T 機能または工具番号の値の範囲

T 機能または工具番号は、以下の整数番号にすることができます。

- 最小値：T0 (工具なし)
- 最大値：T32000 (番号 32000 の工具)まで

工具刃先 D

工具 Tx は、さまざまな工具刃先 D1 ... Dn に割り当てることができます。

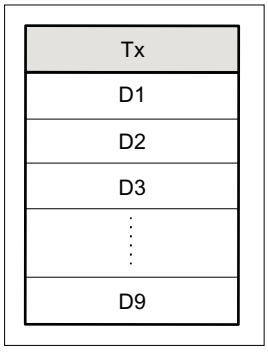


図 19-1　9 つの刃先(D1 ～ D9)を持つ工具 Tx の例

1 工具あたりの最大工具刃先数

各工具に定義できる工具刃先(D 番号)の最大数は、以下のマシンデータを使用して指定します。

- MD18106 $MN_MM_MAX_CUTTING_EDGE_PERTOOL

D 機能

加工に必要な工具刃先 1 ... n の選択は、パートプログラムなどに D1 ... Dn で適切な D 機能をプログラム指令して実行します。選択した工具刃先は常に、現在有効な工具を参照します。有効な工具のない工具刃先(実際の T 番号は T0 ⇒工具が選択されていない)は無効です。

有効な工具 Tx の工具オフセットは、D0 を使用して選択解除します。

工具交換時の刃先の選択

新しい T 番号を使用して、新しい工具を選択し、この工具をロードする場合は、以下のオプションを使用して刃先を選択できます。

- 刃先番号 Dx をプログラム指令する。
- 下記のマシンデータを使用して、刃先番号を定義する。
  MD20270 $MC_CUTTING_EDGE_DEFAULT =<値> (プログラム指令されないときの工具刃先の基本設定)

値	意味
= 0	M06 による刃先の自動選択なし
≠ 0	M06 により選択される刃先番号
= -1	M06 により、古い工具の刃先番号が維持され、新しい工具に対しても選択される。

工具オフセットの起動

D1 ～ Dn によって、有効な工具の刃先に対する工具補正が起動されます。工具長補正と工具径補正は異なるタイミングで有効にできます。

- **工具長補正** (TLC)は、TLC が作用すべき、軸の最初の移動動作時におこなわれます。この移動動作は、直線補間(G0、G1、POS、POSA)または多項式補間(POLY)でなければなりません。POS/POSA 軸が有効なジオメトリ軸の 1 つである場合、工具長補正は、TLC が作用することになっている最初の軸動作で適用されます。
- **工具径補正** (TRC)は、G41/G42 が有効な平面(G17、G18、または G19)でプログラム指令されるときに有効になります。
  工具半径補正が G41/G42 で起動されている NC ブロックでは、G0 (高速処理)または G1 (直線補間)を有効にし、有効な作業平面の 1 本以上のジオメトリ軸をプログラム指令してください。
  電源投入時に指定されているジオメトリ軸が 1 本のみの場合は、2 番目のジオメトリ軸の最後の位置が自動的に補完され、両方の軸で移動します。
  これらの軸がジオメトリ軸でない場合は、GEOAX プログラミングを使用してジオメトリ軸に変換してください。
  G41/G42 を使用した工具径補正の選択は、G0 (早送り)または G1 (直線補間)を持つプログラムブロックでのみ許容されます。

19.2.2 補正メモリ構造

工具オフセットメモリサイズ

各チャネルは、専用の工具オフセットメモリ(TO ユニット)を持つことができます。

下記のマシンデータを使用して、関連するチャネルについてどの工具オフセットメモリが存在するかを設定します。

MD28085 $MC_MM_LINK_TOA_UNIT (チャネルへの TO ユニットの割り当て)

下記のマシンデータを使用して、NC によって管理されるすべての工具の工具刃先の最大数を設定します。

MD18100 $MN_MM_NUM_CUTTING_EDGES_IN_TOA (NC での工具刃先数)

工具

ツールオフセットメモリは、T1 ～ T32000 の番号が付けられた工具で構成されます。

各工具は、TOA ファイルを介して、または個別に「新しい工具」ソフトキーを使用してセットアップできます。不要なオフセット値には、値ゼロを割り当ててください。これは、オフセットメモリが作成されるときの初期設定です。オフセットメモリの個々の値(工具パラメータ)は、システム変数を使用してプログラムから読み取り/書き込むことができます。

注記

工具(T1 ～ T32000)を工具オフセットメモリに昇順または連続して保存しなくてもかまいません。また、最初の工具に番号 T1 を割り当てる必要もありません。

工具刃先

各工具は、最大 9 つの刃先(D1 ～ D9)を持つことができます。最初の刃先(D1)は、新しい工具が工具オフセットメモリにロードされるときに自動的にセットアップされます。他の刃先(8 つまで)は、「新しい刃先」ソフトキーを使用して連続的に隣接してセットアップされます。この方法で、各番号の工具刃先を各工具に割り当てることができます。

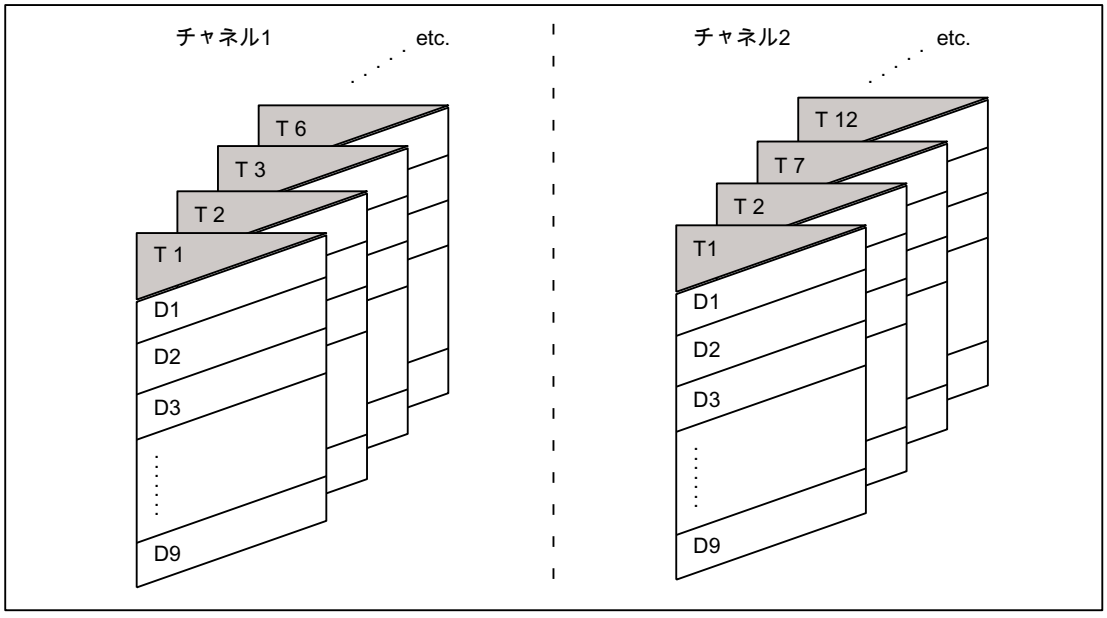
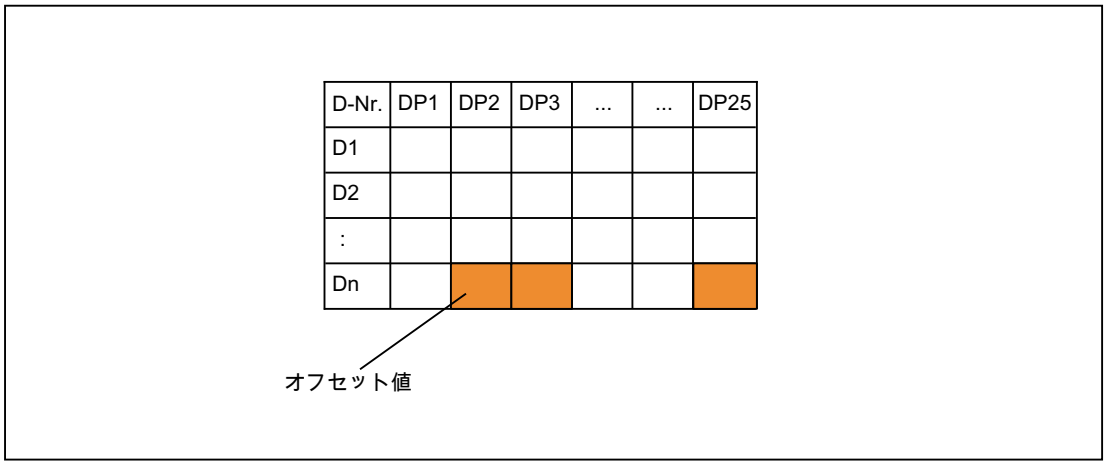


図 19-2　2 チャネルでの工具オフセットメモリ構造の例

19.2.3 工具補正の計算

D 番号

工具補正を計算するには **D 番号**があれば十分です(マシンデータによって設定可能)。



プログラミング

上記の補正ブロックは、NC 内で計算されます。

パートプログラムでの呼び出し:

```
...
Dn
```

19.2.4 NC アドレス T および M のアドレス拡張

MD20096;MD２００９６

下記のマシンデータを使用すると、工具管理が有効**でない**場合でも T および M のアドレス拡張を主軸番号として解釈すべきかどうかを指定できます。

MD20096 $MC_T_M_ADDRESS_EXT_IS_SPINO (アドレス拡張としての主軸番号)

「工具管理」機能が有効な場合と同じ規則が、D 番号と T 番号の間の関係に適用されます。

D 番号での影響

D 番号によってオフセットデータセットが決められます。

アドレス拡張とともに D アドレスをプログラムすることはできません。

D アドレスの適用は常に、現在有効な工具を対象とします。

下記のマシンデータが設定されている場合、プログラムされた D アドレスは、メイン主軸に関連する有効な工具を対象とします(工具管理機能の場合と同じ)。

MD20096 $MC_T_M_ADDRESS_EXT_IS_SPINO = TRUE (アドレス拡張としての主軸番号)

T 番号での影響

「工具管理」機能が有効である場合、メイン主軸(またはマスタ工具ホルダ)に関してプログラムされた値は、プログラムされた/有効な T 番号として表示されます。

工具管理が有効でない場合、プログラムされた**すべての** T 値は、プログラムされたアドレス拡張とは無関係に、プログラムされた/有効な値として表示されます。

次のマシンデータが設定されている場合、メイン主軸に関してプログラムされた T 値のみ、プログラムされた/有効な値として表示されます。

MD20096 $MC_T_M_ADDRESS_EXT_IS_SPINO = TRUE (アドレス拡張としての主軸番号)

例

次の例は、MD20096 の働きを示しています。

2 つの主軸を使用します。主軸 1 がメイン主軸です。M6 が工具交換の合図として定義されました。

T1 = 5	
M1 = 6	
T2 = 50	
M2 = 6	
D4	

- 工具管理が有効である場合、D4 は工具「5」を対象とします。
  T2=50 で、軌跡補正に影響しない第 2 主軸の工具が定義されます。軌跡は、メイン主軸用にプログラムされた工具によってのみ決められます。
- 工具管理が有効でなく、次のマシンデータが設定されている場合、D4 は工具「50」に関係します。
  MD20096 $MC_T_M_ADDRESS_EXT_IS_SPINO = **FALSE** (T および M 工具交換時のアドレス拡張の意味)
  T および M のどちらのアドレス拡張も NC で使用されません。
  各工具交換命令によって新しい軌跡補正が定義されます。
- 工具管理が有効でなく、次のマシンデータが設定されている場合、D4 は工具「5」を基準とします(工具管理が有効な場合と同じ)。
  MD20096 $MC_T_M_ADDRESS_EXT_IS_SPINO = **TRUE**
  アドレス拡張 1 (T1= ..., M1= ...)は、メイン主軸を指定します。

注記

以前は、工具管理が有効でない場合、各工具交換命令(T または M でプログラムされた)を使用すると、工具オフセットが軌跡内で再計算されていました。アドレス拡張は、この処理ではそれ以上定義されません。拡張の意味は、ユーザーが PLC ユーザープログラム内で定義します。

19.2.5 D 番号の任意割り当て

「相対」D 番号

NC では、D 番号を、工具補正データセットの「相対」D 番号として管理できます。対応する D 番号が各 T 番号に割り当てられます。

機能

D 番号割り当て時の機能の拡張

- 許容される D 番号の最大数は、下記のマシンデータによって定義されます。
  MD18105 $MN_MM_MAX_CUTTING_EDGE_NO (D 番号の最大値(DRAM))
  既存のアプリケーションとの互換性を維持するために、初期値は 9 となっています。
- **各工具の**刃先数(またはオフセットデータセット)は、次のマシンデータによって定義できます。
  MD18106 $MN_MM_MAX_CUTTING_EDGE_PERTOOL (工具別 D 番号の最大数 (DRAM))
  これにより、監視目的で、各工具について刃先の数が実際の刃先の数に合わせて設定されるようにカスタマイズできます。
- また、NC では、D 番号の名称を変更できるため、どのような D 番号でも刃先に割り当てることができます。

注記

相対 D 番号の割り当てに加え、D 番号を、T 番号を参照することなく「フラット」または「絶対」D 番号(1 ～ 32000)として割り当てることもできます。(「フラット D 番号構造」機能内で)

刃先番号 CE

D 番号の名称を変更すると、これらの刃先について定義された番号を記載した工具カタログの情報が失われます。このため、名称の変更の後で、カタログのどの刃先が参照されているかを特定することができません。

この情報は段取り替え手順で必要とされるため、**刃先番号 CE** が各刃先について導入されました。この番号は、D 番号の名称が変更されても保存されたままになります。

D 番号は、パートプログラムでの刃先補正を識別します。この**補正番号 D** は、**刃先番号 CE** (工具カタログに記載される番号)とは別に管理されます。どのような番号でも使用できます。この番号は、パートプログラム内とディスプレイ上の補正を識別するのに使用されます。

CE 番号は、段取り替え中の実際の物理的な刃先を識別します。工具交換中の補正選択時に、刃先番号 CE は NC によって評価されません(OPI を介してのみ使用可能)。

刃先番号 CE は、システム変数**$TC_DPCE[t,d]**によって定義されます。

- **t** は内部 T 番号を表します。
- **d** は D 番号を表します。

衝突がないかどうか、書き込みアクセスが監視されます。つまり、工具のすべての刃先番号が異なっていなければなりません。変数$TC_DPCE は、刃先パラメータデータセット$TC_DP1 ～$TC_DP25 の要素です。

最大刃先番号(MD18105)が工具あたりの最大刃先数(MD18106)より大きい場合だけ、$TC_DPCE をパラメータ設定することが役立ちます。

この場合、初期設定の刃先番号は、刃先の分類番号と同じです。工具の補正は番号 1 から作成され、工具あたりの最大刃先数(MD18106)まで 1 ずつ加算されます。

以下の場合、刃先番号 CE は D 番号と等しくなります(これまでの挙動との互換性あり)。

MD18105 ≤ MD18106

読み取り動作は CE=D を返します。書き込み動作は無視され、アラームメッセージは表示されません。

注記

有効な工具補正の補正値$TC_DP1 to $TC_DP25 は、システム変数$P_AD[n] (ここで、n=1 ～ 25)で読み取ることができます。有効な補正の CE 刃先番号は、n=26 と共に返されます。

命令

最大刃先番号が、各工具の刃先の最大番号**より大きい**場合は、以下の表の命令を使用できます。

- (MD18105 $MN_MM_MAX_CUTTING_EDGE_NO) > (MD18106 $MN_MM_MAX_CUTTING_EDGE_PERTOOL)

命令	意味
CHKDNO	使用可能な D 番号が一義的であるかどうか確認します。 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。
GETDNO	工具の刃先の D 番号を特定します。 入力パラメータに一致する D 番号が存在しない場合、d=0 です。 D 番号が無効な場合、32000 より大きい値が返されます。
SETDNO	工具 T の刃先 CE の D 番号を設定または変更します。 指定されたパラメータに一致するデータブロックが存在しない場合、値 FALSE が返されます。構文異常があると、アラームが出力されます。D 番号を 0 に設定することはできません。

命令	意味
GETACTTD	絶対 D 番号に関連付けられた T 番号を特定します。 一義的かどうかはチェックされません。TO ユニット内の複数の D 番号が同じである場合、検索で見つかった最初の工具の T 番号が返されます。 この命令は、「フラット」D 番号で使用するには適していません。その場合、常に値 1 が返されるからです(データベースに T 番号がない)。
DZERO	TO ユニットのすべての D 番号に、無効の印を付けます。 この命令は、段取り替え中のサポートに使用します。 この命令によって印が付けられた補正データセットは、CHKDNO 言語命令によって検証されません。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。

注記

最大刃先番号が、各工具の刃先の最大番号**より小さい**場合、この表の命令は無効です。この初期設定は、互換性を保つためにシステムに用意されています。

個々の命令の詳細については、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、基本編』

起動

一義的な D 番号、つまり定義された言語命令を使用して作業するには、工具に対する D 番号を自由に命名できなければなりません。

このためには、以下の条件を満たさなければなりません。

- MD18105 > MD18106
- 'フラット D 番号'機能が起動されていない。
  MD18102 $MN_MM_TYPE_OF_CUTTING_EDGE (D 番号指令のタイプ(SRAM)).

例

MD18105 $MN_MM_MAX_CUTTING_EDGE_NO = 1 (D 番号の最大値)

これにより、工具あたり少なくとも 1 つの補正を定義できます(D 番号= 1 で)。

注記

「フラット D 番号」が有効な場合、D 補正を 1 つだけ TO ユニットで定義できます。

MD18105 $MN_MM_MAX_CUTTING_EDGE_NO = 9999

工具に一義的な D 番号を割り当てることができます。

例:T 番号(工具)への D (刃先)の割り当てと、一義性をチェックするオプションのリスト:

- 割り当て:
  - T 番号 1⇒D 番号 1、2、3 の割り当て
  - T 番号 2⇒D 番号 10、20、30、40、50 の割り当て
  - T 番号 3⇒D 番号 100、200、30 の割り当て(**エラー**:300 の代わりに 30 が入力された)
- 一義性のチェック:
  TO ユニット内で定義されたすべての工具の D 番号は一義的でなければなりません。チェックは、CHKDNO 命令を使用しておこないます。パラメータを指定しない場合は、すべての工具のすべての D 番号が相互にチェックされます。この場合は、工具T2 および T3 で D30 を使用できるため、戻り値== FALSE,となります。

命令	テスト	テスト結果
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。	戻り値== FALSE、D=30 が重複しています。
CHKDNO (2, 3, 30)	工具 2 および 3 の D 番号	戻り値== FALSE、D=30 が重複しています。
CHKDNO (1, 3)	工具 1 および 3 のすべての D 番号	戻り値== TRUE、ただし、3 番目の工具の D=30 と 2 番目の工具の D=30 の間に競合があります。

注記

1 工具あたりの工具刃先

n 個の刃先を持つ工具を使用する場合は、マシンデータの値を **n** に設定してください。これで、許可されない刃先が工具に対して定義されなくなります。

MD18106 $MN_MM_MAX_CUTTING_EDGE_PERTOOL = **n**

プログラミング例

D 番号の名称変更

刃先 CE = 3 の D 番号の名称を 2 ～ 17 に変更します。以下の指定を適用します。

- $TC_DPx[<工具 Tn>, <刃先 Dm>]
- 内部 T 番号 Tn = 1
- D 番号 Dm = 2
- 下記により 1 つの刃先を持つ工具

プログラムコード	コメント
$TC_DP2[1, 2] = 120	工具長= 120
$TC_DP3[1, 2] = 5.5	工具径= 5.5
$TC_DPCE[1, 2] = 3	刃先番号 CE = 3

- MD18105 $MN_MM_MAX_CUTTING_EDGE_NO = 20 (D 番号の最大値)

パートプログラム内で、この補正は標準で T1, ...D2 を使用してプログラム指令されます。

下記により、刃先 3 の現在の D 番号を変数(DNoOld)に割り当て、新しい D 番号向けの変数 DNoNew を定義します。

```
プログラムコード
def int DNoOld, DNoNew = 17
DNoOld = GETDNO( 1, 3 )
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。

この補正が、T1, ...D17 を使用してパートプログラムでプログラム指令され、次のようにアドレス指定されます。

```
プログラムコード                    コメント
$TC_DP2[ 1, 17 ] = 120
$TC_DP3[ 1, 17 ] = 5.5
$TC_DPCE[ 1, 17 ] = 3               ; 刃先番号 CE
```

注記

工具でさらに刃先を定義している場合(たとえば、$TC_DPCE[1, 2] = 1 ; = CE)、刃先 1 の D 番号 2 は、刃先 3 の D 番号と同じ名前を持つことができません。つまり SETDNO(1, 1, 17)は、戻り値として状態=FALSE を返します。

DZERO - D 番号の解除

この命令によって、TO ユニット内の工具の D 番号がすべて無効になります。有効な D 番号を NC で再び使用できるようになるまで、補正を起動することはできません。SETDNO 命令を使用して、D 番号を再度割り当ててください。

以下の工具を定義しなければなりません(すべて刃先番号 1 で)。

T1, D1	刃先 CE=1 の D 番号
T2, D10	刃先 CE=1 の D 番号
T3, D100	刃先 CE=1 の D 番号

次に、以下の命令をプログラム指令します。

プログラムコード
DZERO

この時点でいずれかの補正が起動される、たとえば、T3 D100 を指令すると、D100 は現在定義されていないため、アラームが出力されます。

D 番号を以下のように再定義します。

プログラムコード	コメント
SETDNO(1, 1, 100)	; T=1、(新しい) D 番号 100 が刃先 1 に割り当てられます。
SETDNO(2, 1, 10)	; T=2、(新しい) D 番号 10 が刃先 1 に割り当てられます。
SETDNO(3, 1, 1)	; T=3、(新しい) D 番号 1 が刃先 1 に割り当てられます。

注記

停電が発生すると、DZERO 命令によって、NC を D 番号に関して無定義状態のままにできます。このような場合は、電源が復帰したときに DZERO 命令を繰り返してください。

段取り替えプログラムの基本操作

必要な工具と刃先を確実に使用可能にしたいものとします。NC の工具保持マガジンは、どのような状態でもかまいません。新しい加工運転用のパートプログラム内の D 番号は

通常、実際の刃先の D 番号とは一致しません。次のような段取り替えプログラムが可能です。

プログラムコード	コメント
DZERO	; TO ユニットのすべての D 番号に、無効の印を付けます。
....	;工具とその刃先番号をチェックするための、マガジンロケーション全体への 1 つまたは複数のループ。
	;必要な刃先番号 CE (GETDNO)を持つ、まだ有効な($TC_TP8)工具が見つかると、新しい D 番号が刃先に割り当てられます(SETDNO)。
....	;ロードおよびアンロード操作がおこなわれます。
	;工具状態「アンロードされる」と「ロードされる」で作業できます。
CHKDNO	;ロード/アンロードと D 番号変更処理が完了しました。
	;個々の工具か D 番号または、その両方がチェックされ、戻り値に従って重複が自動的に処理されます。

19.2.6 工具交換中の異常発生時の補正ブロック

MD22550

工具準備がパートプログラムでプログラムされて、NC で異常が検出された(たとえば、プログラムされた T 番号のデータセットが NC に存在しない)場合、後で加工を再開するために、異常状況を診断して適切な対策を講じることができます。

工具交換は、次のマシンデータに応じて個別にプログラムすることができます。

MD22550 $MC_TOOL_CHANGE_MODE (M 機能による新しい工具オフセット)

MD22550 $MC_TOOL_CHANGE_MODE = 0

```
T= "T 番号"          ; 1 つの NC ブロック内での工具準備および工具交換
                     ; つまり、T がプログラムされると、新しい D 補正が
                     ; NC で有効。次のマシンデータを参照してください。
                     ; MD20270 $MC_CUTTING_EDGE_DEFAULT (プログラム指令されないときの工具刃先の基本設定)
```

MD22550 $MC_TOOL_CHANGE_MODE = 1

```
T= "T 番号"          ; 工具準備
M06                  ; 工具交換
                     ; (工具交換 M コードの番号を設定できます)
                     ; つまり、M06 がプログラムされると、新しい D 補正が
                     ; NC で有効。
                     ; マシンデータ MD20270 $MC_CUTTING_EDGE_DEFAULT を参照
```

工具管理が有効でない場合、次のような問題が検出されることがあります。

- D 補正データセットの欠落
- パートプログラムの異常

注記

「マガジンに工具がない」という問題を検出することはできません。なぜならば、NC が工具オフセットのマガジン情報にアクセスしていないからです。

D 補正データセットの欠落

無効な D 値を含むブロックでプログラムの実行が中断されます(マシンデータ MD22550 の値とは関係なく)。オペレータは、プログラムを修正するか、欠落したデータセットを再ロードしなければなりません。

D 番号、または T 番号は、フラット D 番号機能で必要です。これらのパラメータは、アラームが出力されたときに渡されます。

パートプログラムの異常

異常が発生した場合の介入オプションは、工具交換のプログラム方法(次のマシンデータによって定義)によって決まります。

MD22550 $MC_TOOL_CHANGE_MODE (M 機能による新しい工具オフセット)

T プログラミングを使用した工具交換(MD22550 = 0)

この場合、NC で利用可能な「補正ブロック」機能が使用されます。NC プログラムは、プログラムされた T 値で異常が検出された NC ブロックで停止します。「補正ブロック」は、プログラムの再開時に再び実行されます。

オペレータは、次のことを実行できます。

- パートプログラムを修正する。
- 欠落した刃先補正データを HMI から再ロードする。
- 「オーバストア」を使用して、欠落した刃先補正データを NC に含める。

オペレータの介入後、START キーを押すと、異常が発生したブロックが再び実行されます。異常が修正された場合、プログラムが再開されます。異常が修正されない場合は、アラームが再度出力されます。

T および M06 プログラミングを使用した工具交換(MD22550 = 1)

この場合、工具準備(T プログラミング)の NC ブロックで異常が検出されますが、この異常は最初無視されます。工具交換要求(通常 M06)を NC プログラムで実行するまで、実行は続行します。この時点でプログラムを停止します。

プログラムされた T アドレスでは、M06 命令の前に何行でもプログラム行を含むことができます。また、2 つの命令が異なる(サブ)プログラムに出現してもかまいません。したがって、一般に、補正ブロックによりすでに実行されているブロックを修正することはできません。

オペレータには、 MD22550 = 0 の場合と同じ介入オプションがあります。

欠落したデータの再ロードが可能です。ただし、この場合、「オーバストア」を使用して T をプログラムしなければなりません。

プログラム異常が発生した場合、異常のある行を修正できません(Txx)。プログラムが停止しアラームを出力した行のみ編集できます(次のマシンデータ設定の場合のみ)。

MD22562 $MC_TOOL_CHANGE_ERROR_MODE ビット 0 = 1 (工具交換時の異常に対する動作)

この結果、次の処理が行われます。

```
Txx                ; 異常!xx を含むデータセットが存在しません
                   ; 状態に注目、xx に注目
                   ; プログラムを続行します。
....
M06                ; ビットメモリ「xx 欠落」を検出 → アラーム出力
                   ; プログラムを停止します。
                   ; ブロックを修正します (たとえば、Tyy M06 を使用して開始)。
                   ; ブロック Tyy M06 が解析され OK になります。
                   ; 実行は継続します。
```

プログラムポイントの実行が更新されたことで、次のようになります。

```
Txx                ; 異常!xx を含むデータセットが存在しません。
                   ; 状態に注目、xx に注目
                   ; プログラムを続行します。
....
Tyy M06            ; ビットメモリ「xx 欠落」を検出 → それ以外の応答をせず、拒否します。
                   ; Tyy M06 が正しいため → プログラムは停止しません (正しい)。
```

必要に応じて、T 呼び出しの起点をプログラム終了後に修正できます。機械の工具交換ロジックでこれを処理できない場合、プログラムを強制終了し、異常を修正してください。

データセットが 1 つのみ欠落している場合、それを NC に伝送し、Txx が「オーバストア」でプログラムされるとプログラムが再開されます。

「欠落した D 番号」の場合と同様に、「欠落した T 番号」に必要なパラメータ(T 番号)が、当該のアラーム(17191)を介してアクセスできます。

注記

プログラムを修正できるようにするには、異常のある Txx ブロックで直ちにプログラムを停止します。

次のマシンデータ設定の場合、プログラムテストモードも停止します。

MD22562 $MC_TOOL_CHANGE_ERROR_MODE ビット 0 = 1 (工具交換時の異常に対する動作)。

19.2.7 工具パラメータの働きの定義

MD20360

直径指定に関連する、径方向軸に対する工具パラメータの影響は、次のマシンデータを使用して的確に制御できます。

MD20360 $MC_TOOL_PARAMETER_DEF_MASK (工具パラメータの定義)

このマシンデータで詳細を指定します。

半径による DRF ハンドル移動

DRF ハンドル移動の間は、次のように、径方向軸を指定された移動量の半分の距離だけ動かすことができます。

次のマシンデータを使用して、ハンドルによる距離を指定してください。

MD11346 $MN_HANDWHEEL_TRUE_DISTANCE = 1 (ハンドル軌跡または速度の指定)

次のマシンデータを使用して、径方向軸での DRF オフセットを直径オフセットとして定義してください。

MD20360 $MC_TOOL_PARAMETER_DEF_MASK ビット 9 = 1 (工具パラメータの定義)

軸の DRF 補正を解除(DRFOF)すると、既存の工具補正(工具方向でのハンドルオーバライド)も削除されます。

注記

ハンドルによる重畳移動の詳細については、以下を参照してください。

参照先:

『機能マニュアル、上級機能』;「手動での移動と手動ハンドルでの移動(H1)」

『プログラミングマニュアル、基本編』

(『プログラミングマニュアル』には、DRF オフセットを軸別に解除するための技術的な全てにプログラムオプションが説明されています。.)

19.3 フラット D 番号構造

19.3.1 一般情報

旋盤の場合は、フラット D 番号構造()を D 番号経由でのみ使用して、簡単に工具管理を実行できます。

これにより、以下の境界条件が適用されます。

- **有効でない**一般的な工具管理の場合にのみ使用可能
- 予備工具を定義できない
- マガジンを定義できない
- 研削工具を定義できない

起動

有効な D 番号管理のタイプは、以下を使用してパラメータ設定します。

MD18102 $MN_MM_TYPE_OF_CUTTING_EDGE = <値>

値	意味
0	フラット D 番号管理が無効
1	絶対**直接 D プログラミング**を使用したフラット D 番号構造が有効

19.3.2 新しい D 番号(補正ブロック)の作成

プログラミング

システム変数$TC_DP1 ～$TC_DP25 を使用して、工具オフセットをプログラムできます。この内容の意味は、前述と同じです。

表記の変更:T 番号はまだ指定されていません。

- 「フラット D 番号」機能が**有効**な場合
  $TC_DPx[d] = 値、x = パラメータ番号、d = D 番号
  つまり、「フラット D 番号」機能が有効な場合、この構文を持つデータのみ NC にロードできます。
- 「フラット D 番号」機能が**無効**な場合
  $TC_DPx[t][d] = 値、t = T 番号、d = D 番号

工具ごとに D 番号を 1 つのみ割り当てることができます。つまり、各 D 番号は、厳密に 1 つの補正データブロックを表します。

存在しない D 番号が初めて作成されると、新しいデータブロックが NC メモリに保存されます。

次のマシンデータを使用して、D 番号またはオフセットデータブロックの最大数(最大 600)を設定できます。

MD18100 $MN_MM_NUM_CUTTING_EDGES_IN_TOA (TO エリアでの工具オフセット)

データバックアップ

データバックアップは、同じフォーマットです。つまり、「フラット D 番号」機能を使用して作成されたバックアップファイルは、この機能を起動していない制御装置の NC にロードすることはできません。

同じことが、反対の転送にも当てはまります。

D 番号範囲

1 - 99 999 999

19.3.3 D 番号のプログラミング

MD18102 = 1

MD18102 $MN_MM_TYPE_OF_CUTTING_EDGE = 1 の場合は、D 補正が、特定の工具を参照せずに有効になります。

D0 の内容は、以前の「NC での有効な補正の無効化」と同じです。

D のアドレス拡張

D のアドレスを拡張することはできません。同一時点で工具軌跡について有効な補正データブロックは 1 つのみ可能です。

プログラミング

パートプログラムでのプログラミングは以前と同様に実行されます。プログラムされるD番号の数値の範囲のみ拡張されます。

例 1:

マシンデータのパラメータ設定		意味
MD22550 $MC_TOOL_CHANGE_MODE	= 0	T のプログラミングによる工具交換。
MD18102 $MN_MM_TYPE_OF_CUTTING_EDGE	= 1	特定の工具を参照しない D 番号のプログラミング。
MD20270 $MC_CUTTING_EDGE_DEFAULT	= -1	工具を交換した場合も D 補正は変わりません。

プログラムコード	コメント
...	
D92	
X0	; D92 の補正による移動。
T17	; PLC への出力 T=17
X1	; D92 の補正による移動。
D16	
X2	; D16 の補正による移動。
D32000	
X3	; D32000 の補正による移動。
T29000	; PLC への出力 T=29000
X4	; D32000 の補正による移動。

プログラムコード	コメント
D1	
X5	; D1 の補正による移動。
...	

例 2:

MD22550 = 0

プログラムコード	コメント
T1	
T2	
T3	
D777	; 待機なし、D777 が起動される、表示では T3=プログラムされ有効な工具、D777=プログラムされ有効な補正

注記

工具交換、および実際の工具への D 補正の割り当てを行うのは、NC プログラムおよび PLC プログラム(該当する場合)です。

パートプログラムによる D 番号の削除

- フラット D 番号**あり**。
  $TC_DP1[d] = 0
  TO ユニット内に番号 D のある補正データブロックは削除されます。
  次に、別の D 番号を定義するためにメモリが解放されます。
- フラット D 番号**なし**。
  $TC_DP1[t][d] = 0
  工具 t の刃先 d が削除されます。
- $TC_DP1[0]=0
  TO ユニットのすべての D 補正が削除されます。

有効な補正データブロック(D 番号)を削除することはできません。これは、削除する前に D0 のプログラミングが必要となる場合があるということです。

工具マシンデータ

以下のマシンデータを使用して、NC での工具および刃先(D 番号)の動作に影響を与えることができます。

マシンデータ	意味
MD20270 $MC_CUTTING_EDGE_DEFAULT	工具交換後の標準工具刃先
MD20130 $MC_CUTTING_EDGE_RESET_VALUE	電源投入(リセット/パートプログラム終了)時の工具刃先長補正
MD20120 $MC_TOOL_RESET_VALUE	電源投入(リセット/パートプログラム終了)中の工具長補正
MD20121 $MC_TOOL_PRESEL_RESET_VALUE	リセット時に事前選択される工具
MD22550 $MC_TOOL_CHANGE_MODE	T 機能ではなく、M 機能による工具交換
MD22560 $MC_TOOL_CHANGE_M_CODE	工具交換の M 機能
MD20110 $MC_RESET_MODE_MASK	リセット/パートプログラム終了後の基本制御設定の定義
MD20112 $MC_START_MODE_MASK	NC スタート時の初期制御設定の定義

19.3.4 T 番号のプログラミング

「フラット D 番号構造」機能が有効な場合、NC アドレス T が継続して使用されます。つまり、プログラムされた T 番号および有効な T 番号が表示されます。 ただし、NC では、プログラムされた T 番号とは関係なく D 番号が決められます。

NC では、チャネルごとに 1 つのメイン主軸が検出されます(主軸番号を通じて、マシンデータを使って設定可)。 補正および M6 命令(工具交換)は、メイン主軸に関してのみ計算されます。

アドレス拡張 T は、主軸番号として解釈されます(たとえば、T2 = 1 の場合、工具 1 が主軸 2 で選択されます)。主軸 2 がメイン主軸である場合のみ、工具交換が検出されます。

19.3.5 M6 のプログラミング

MD22550 と MD22560

NC では、チャネルごとに 1 つのメイン主軸が検出されます(主軸番号を通じて、マシンデータを使って設定可)。補正および M6 命令(工具交換)は、メイン主軸に関してのみ計算されます。

M 機能を使用して工具交換命令を実行するかどうかを、次のマシンデータを介して定義します。

MD22550 $MC_TOOL_CHANGE_MODE (M 機能による新しい工具オフセット)

T は、工具準備命令として使用されます。

工具交換用の M 機能の名前は、次のマシンデータを使用して定義します。

MD22560 $MC_TOOL_CHANGE_M_CODE(工具交換用 M 機能)

初期設定は M6 です。M6 のアドレス拡張は、主軸番号として識別されます。

例

主軸 1 および主軸 2 という 2 つの主軸が定義されています。以下の動作が適用されます。

```
MD20090 = 2        ; 主軸番号 2 がメイン主軸です。
M6                 ; 求められる工具交換では、命令は暗にメイン主軸を参照します。
M1 = 6             ; 主軸番号 2 がメイン主軸であるため、工具交換はありません。
M2 = 6             ; 主軸番号 2 がメイン主軸であるため、工具交換が実行されます。
```

19.3.6 プログラムテスト

MD20110

次のマシンデータを使用して、有効な工具および工具補正を下記の表に従って処理するように定義できます。

MD20110 $MC_RESET_MODE_MASK, ビット 3 (RESET/パートプログラム終了後のシステム初期設定の定義)

値		意味
ビット 3	= 1	テストモードで終わる最後のテストプログラムから。
	= 0	プログラムテストの起動前の最後に終わるプログラムから。

必要条件

次のマシンデータを使用して、ビット 0 および 6 を設定しなければなりません。

MD20110 $MC_RESET_MODE_MASK, ビット 3 (RESET/パートプログラム終了後のシステム初期設定の定義)

19.3.7 工具管理機能または「フラット D 番号構造」

特徴

工具管理機能

アクティブな NC の工具管理機能は、次のような前提の下で機能します。

1. 工具がマガジンの中で管理されます。
2. 刃先が監視されます。リミットに達すると、工具が無効になります。
3. 予備工具:工具は、その名称に基づいてのみ選択されるようにプログラムされます。NC は特定の工具を選択します。

注記

SINUMERIK 828D では、この機能はオプションとしてのみ使用可能です。

NC に管理される特定の工具が定義されている場合のみ工具管理機能を使用することができます。

フラット D 番号の構成

フラット D 番号構造を使用すると、NC (PLC)の外で工具管理機能が実行されます。T 番号への参照は行われません。

工具管理機能またはフラット D 番号構造

工具管理機能(NC)とフラット D 番号構造(PLC)を同時に使用しても意味はありません。**時間** 因子が関係する場合のみ工具管理機能を使用する意義があります。これは、管理が NC により実行される場合のみ保証されます。ただし、これは「フラット D 番号」には当てはまりません。

注記

工具管理機能およびフラット D 番号を同時を有効にすると、工具管理機能がアクティブになります。

19.4 工具刃先

19.4.1 一般情報

工具刃先

以下のデータを使用して、工具刃先が一義的に記述されます。

- 工具タイプ(エンドミル、ドリル、その他)
- ジオメトリの記述
- テクノロジの記述

工具パラメータ

ジオメトリの記述、テクノロジの記述、および工具タイプが、各工具刃先の工具パラメータに割り当てられます。

工具タイプに関連して、以下の工具パラメータが用意されています。

工具パラメータ	意味	備考
1	工具タイプ	
2	刃先位置	輪郭工具での 2 次元 TRC 付きの旋削工具またはフライス/研削工具
ジオメトリ - 工具長		

工具パラメータ	意味	備考
3	長さ 1	
4	長さ 2	
5	長さ 3	
ジオメトリ - 工具形状		
6	半径 1	2 次元 TRC 付き 旋削:刃先の丸み付け半径 3 次元正面削り:シャンク半径 メタルソー:直径
7	半径 2	3 次元正面削り:コーナ半径 メタルソー:溝幅
8	長さ 4	メタルソー:突起
9	長さ 5	3 次元正面削り:(球状工具の)上部テーパ刃エンドミル直径
10	角度 1	旋削:ホルダ角度 輪郭工具での 2 次元 TRC:最小角度
11	角度 2	旋削:切削方向 輪郭工具での 2 次元 TRC:最大角度 3 次元正面削り:(テーパ工具の)側線と工具長手軸間の角度
摩耗 - 工具長		
12	磨耗長さ 1	
13	磨耗長さ 2	
14	磨耗長さ 3	
摩耗 - 工具形状		
15	摩耗半径 1	
16	摩耗半径 2	
17	磨耗長さ 4	
18	磨耗長さ 5	
19	摩耗角度 1	

工具パラメータ	意味	備考
20	摩耗角度 2	
工具ベース寸法/アダプタ寸法		
21	基本長さ 1	
22	基本長さ 2	
23	基本長さ 3	
用途		
24	アンダーカット角度	旋削工具のみ
25	予約済み*	

*「予約済み」とは、この工具パラメータが使用されておらず、拡張用に予約されていることを意味します。

3 次元正面削り

工具パラメータ(1 ～ 23)を使用することで、フライス盤のタイプ 111、120、121、130、155、156、および 157 が、3 次元正面フライス加工で特別に取り扱われます。

参照先

さまざまな工具タイプの詳細については、以下を参照してください。

- 総合機能説明書 基本機能; 工具オフセット(W1)、「工具タイプ(工具パラメータ)」の章
- 『プログラミングマニュアル、基本編』、「工具補正」 > 「工具タイプリスト」の章
- 機能マニュアル、応用機能; 3 次元工具径補正(W5)

19.4.2 工具パラメータ 1:工具タイプ

概要

工具タイプ(3 桁の数字)で該当する工具を定義します。 この工具タイプを選択することで、更に、ジオメトリ、摩耗、工具ベース寸法などの要素を事前に指定できます。

条件

以下の条件が、「工具タイプ」パラメータに適用されます。

- 各工具刃先について工具タイプを指定しなければなりません。
- 指定された値のみ、工具タイプに使用できます。
- 工具タイプ「0 (ゼロ)」は、有効な工具が定義されていないことを意味します。

工具タイプおよび工具パラメータ

さまざまな工具タイプと最も重要な工具パラメータを下記の表に挙げています。 特定の工具タイプに使用できる工具パラメータは「x」で示しています。

Table of most important tool parameters (only tool type is necessary)																		
1	Tool parameter $TC_DP	2	3	4	5	6	7	8/9	12/13/14	15/16	21/22/23	24	TPG 3	TPG 4/5	TPG 6	TPG 7	TPG 8
工具タイプ	Billing by NCK	刃先位置	ジオメトリ – 長さ1	– 長さ2	– 長さ3	ジオメトリ - 半径1	スロット幅 / – 半径2	突起/ - 長さ4/5	磨耗 - 長さ1/2/3	摩耗 - 半径1/2	調整/工具ベース寸法 - 長さ1/2/3	アンダーカット角度	– 最小研削砥石半径	– 最小/現在の研削砥石幅	– 最大速度	– 最大円周速度	– 傾斜研削砥石の角度
フライス工具およびタッピング工具(その他すべて)																		
100	CLDATA*1に準拠するフライス工具		x	x	x	x			x	x	x						
110	ボールエンドミル(テーパ)		x	x	x	x			x	x	x						
120	エンドミル		x	x	x	x			x	x	x						
121	ラジアスエンドミル		x	x	x	x	x		x	x	x						
130	アングルヘッドカッタ		x	x	x	x			x	x	x						
131	丸コーナ角度付きヘッドフライス		x	x	x	x	x		x	x	x						
140	正面フライス		x	x	x	x			x	x	x						
145	ねじ切り工具		x	x	x	x			x	x	x						
150	サイドカッタ		x	x	x	x			x	x	x						
155	テーパエンドミル		x	x	x	x			x	x	x						
200	ドリル		x	x	x	x			x		x						
205	ドリル		x	x	x	x			x		x						
210	ボーリングバー		x	x	x	x			x		x						
220	センタ穴ドリル		x	x	x	x			x		x						
230	皿取りドリル		x	x	x	x			x		x						
231	座ぐりフライス		x	x	x	x			x		x						
240	標準ねじタップ		x	x	x	x			x		x						
241	精密ねじタップ		x	x	x	x			x		x						
242	ウィットねじタップ		x	x	x	x			x		x						
250	リーマ		x	x	x	x			x		x						
研削工具および旋削工具(400～599)																		
400	平面研削砥石	x	x	x	x	x			x	x	x					x	x
401	平面研削砥石(監視機能あり)	x	x	x	x	x			x	x	x		x	x	x	x	x
403	401と同一、ただしGWPS*2に対応する工具ベース寸法な	x	x	x	x	x			x	x	x		x	x	x		x
410	正面研削砥石	x	x	x	x	x			x	x	x					x	
411	監視機能付き正面研削砥石	x	x	x	x	x			x	x	x		x	x	x	x	
413	411と同一、ただしGWPS*2に対応する工具ベース寸法な	x	x	x	x	x			x	x	x		x	x	x		
490	ドレッサ	x	x	x	x	x			x	x	x						
500	荒削り工具	x	x	x	x	x			x	x	x	x					
510	仕上げ工具	x	x	x	x	x			x	x	x	x					
520	プランジカッタ	x	x	x	x	x			x	x	x	x					
530	突っ切りバイト	x	x	x	x	x			x	x	x	x					
540	ねじ切りバイト	x	x	x	x	x			x	x	x	x					
550	ボタン工具/断面鋼(TOOLMAN)	x	x	x	x	x			x	x	x	x					
特殊工具(700)																	
700	メタルソー		x	x	x	x	x	x	x	x	x						
CLDATA*1 "cutter Location Data" — DIN 66215に準拠する工具ロケーションデータ、 BI 1SUG*2 — 砥石周速制御																		

注記

旋削工具グループでは、工具タイプは意味を持ちません。
特に研削工具(400 ～ 499)では、表に挙げていない番号も使用できます。

工具オフセットデータ

工具オフセットデータ(TOA データ)は、システム変数に保存されます。

溝フライス工具タイプ(タイプ 700)の例

	ジオメトリ	摩耗	ベース	単位
長さ補正				
長さ 1	$TC_DP3	$TC_DP12	$TC_DP21	mm
長さ 2	$TC_DP4	$TC_DP13	$TC_DP22	mm
長さ 3	$TC_DP5	$TC_DP14	$TC_DP23	mm
のこ刃補正				
直径 d	$TC_DP6	$TC_DP15		mm
スロット幅 b	$TC_DP7	$TC_DP16		mm
突起 k	$TC_DP8	$TC_DP17		mm

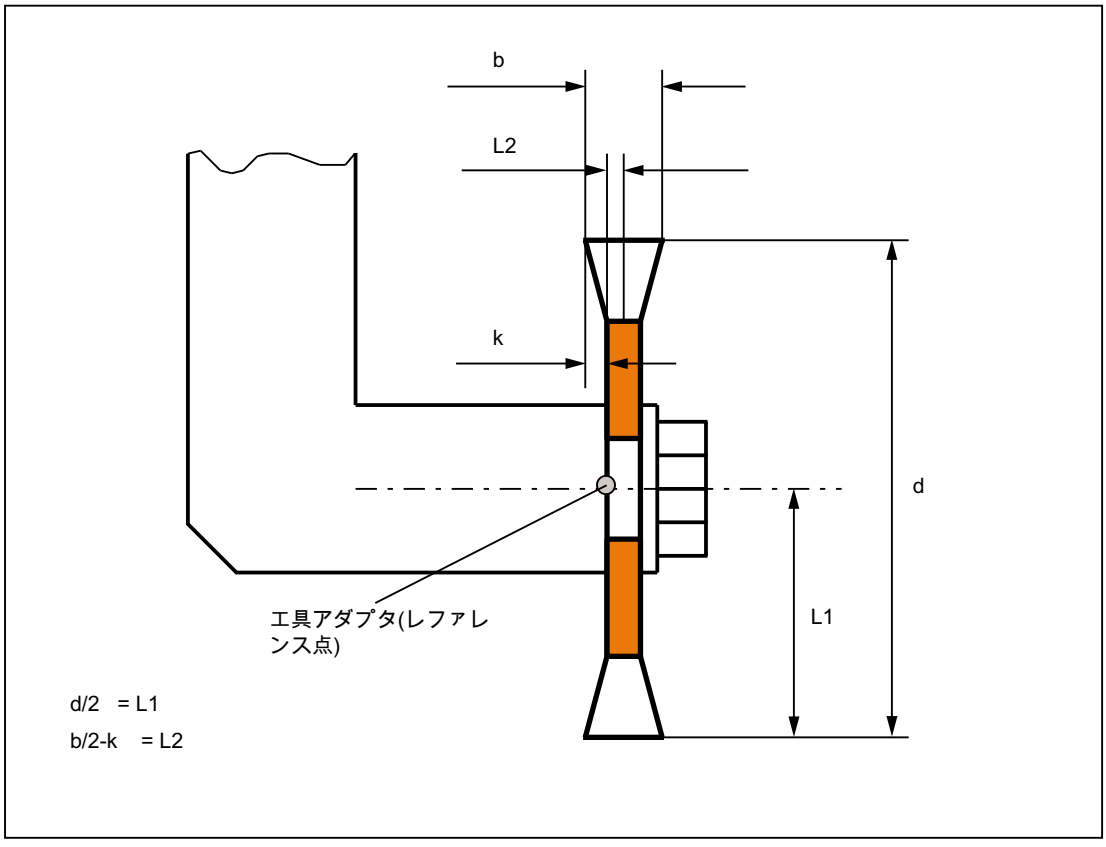


図 19-3　溝フライスのジオメトリ(角度フライスに類似)

次のように、のこ刃の幅が工具径補正(G40 ～ G42)で考慮されます。

命令	意味
G40	のこ刃補正なし
G41	のこ刃補正左
G42	のこ刃補正右

19.4.3　工具パラメータ 2:刃先位置

概要

刃先の中心点 S に対する工具先端 P の位置が、刃先位置として工具パラメータ 2 に入力されます。

刃先位置は刃先径(工具パラメータ 8)とともに、旋削工具(工具タイプ 5xx)の工具径補正を計算するのに必須です。

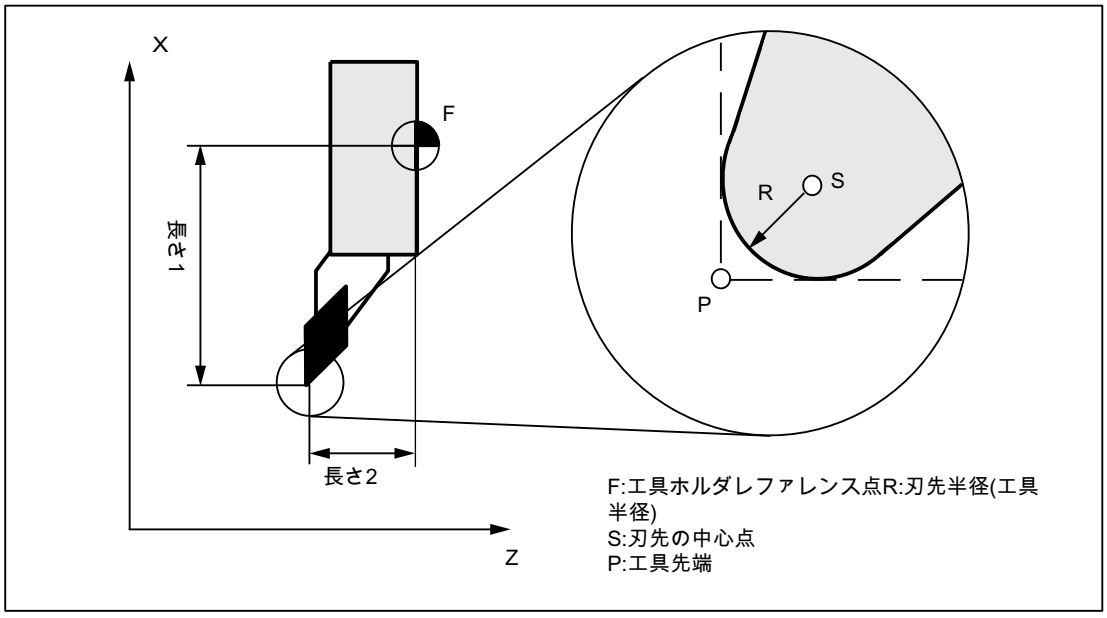


図 19-4　旋削工具の各種寸法: 旋削工具

刃先位置パラメータの値

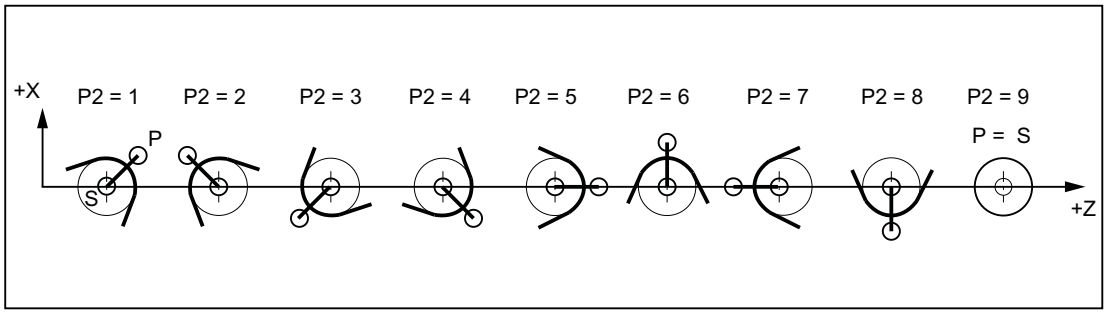


図 19-5　工具パラメータ 2 (P2): ターニングセンタの背面での加工

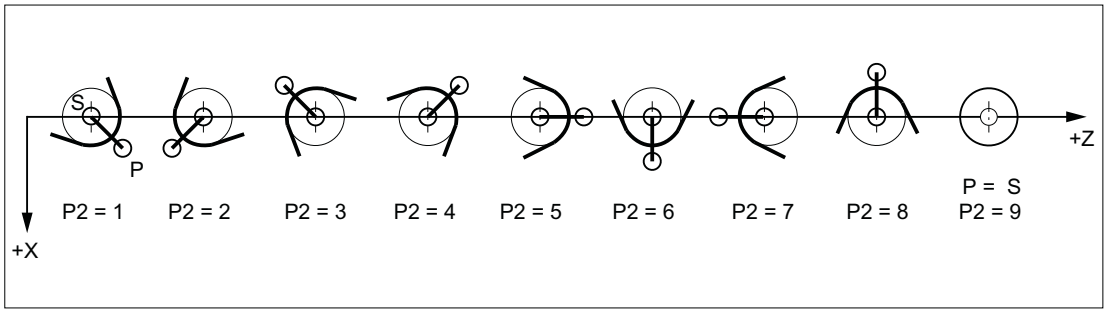


図 19-6　工具パラメータ 2 (P2): ターニングセンタの前面での加工

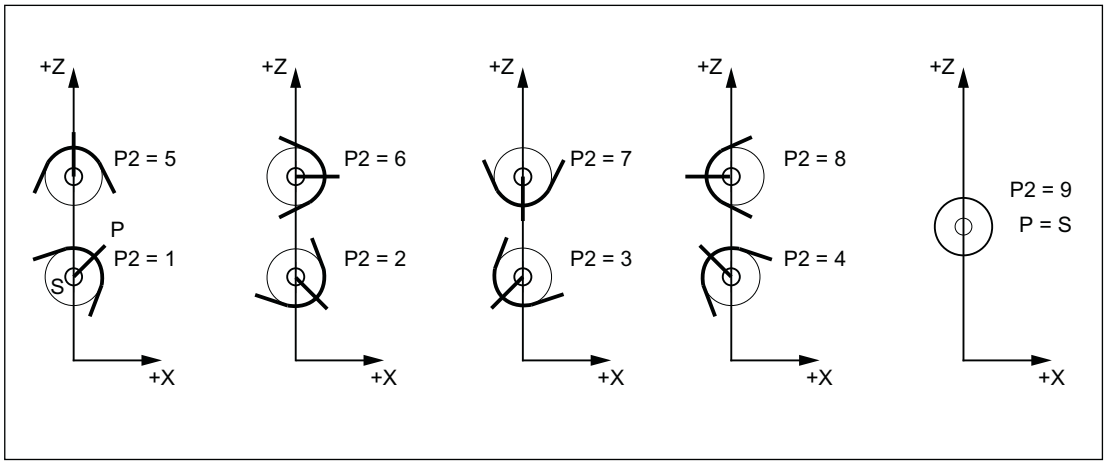


図 19-7　工具パラメータ 2 (P2): 立て旋盤での刃先位置

特別な注意点

- 基準点として点 P の代わりに刃先の中心点 S を使用して工具長補正を計算する場合、刃先位置として識別子 9 を入力しなければなりません。
- 識別子 0 (ゼロ)は、刃先位置として使用できません。

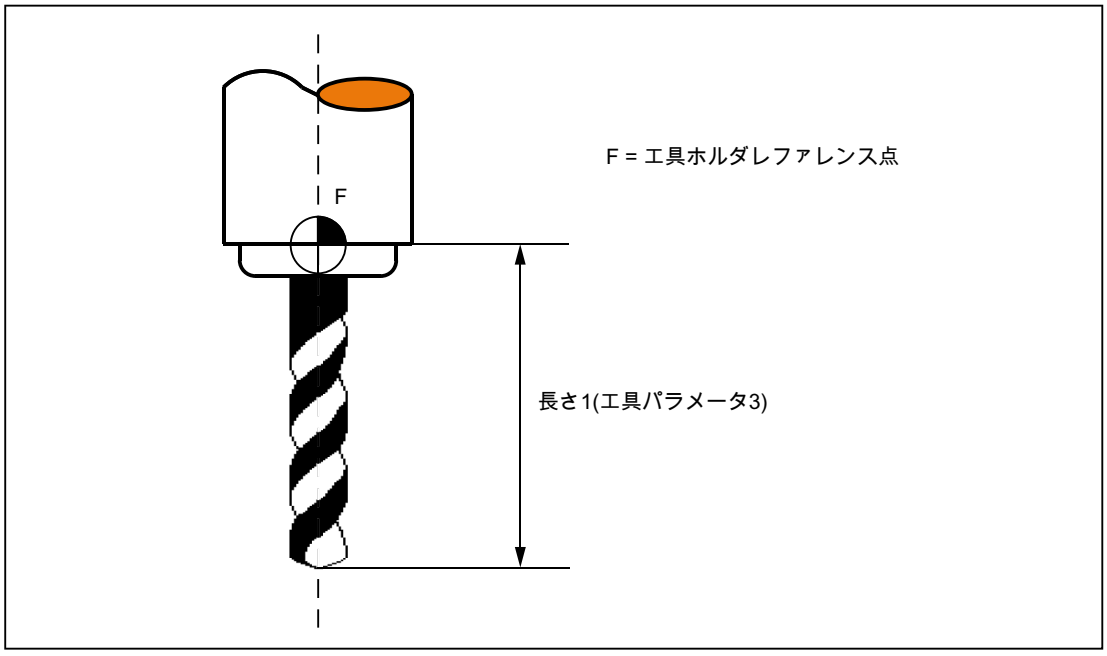
19.4.4　工具パラメータ 3 ～ 5: ジオメトリ - 工具長

説明

ジオメトリ工具長補正には、工具の長さが必要です。 それらは工具長 1 ～ 3 として工具パラメータ 3 ～ 5 に入力されます。工具タイプごとに、最低でも次の長さ指定を入力してください。

工具タイプ	必要な工具長
工具タイプ 12x、140、145、150:	工具長 1
工具タイプ 13x:	工具長 1 ～ 3(平面 G17 ～ G19 に依存)
工具タイプ 2xx:	工具長 1
工具タイプ 5xx:	工具長 1 ～ 3

工具長(工具パラメータ 3)のツイストドリル(工具タイプ 200)の例



注記

3 つの工具パラメータ 3 ～ 5(工具長 1 ～ 3)はすべて、工具タイプに関係なく、常に 3 つのジオメトリ軸で計算されます。

1 つの工具タイプについて、最低限として必要な数より多くの工具長が工具パラメータ3 ～ 5 に入力された場合、それらの余分な工具長はアラームなしでジオメトリ軸内で解決されます。

特別な注意点

工具の有効サイズは、ジオメトリ工具長補正(工具パラメータ 3 ～ 5)と摩耗工具長補正(工具パラメータ 12 ～ 14)が合計されて初めて定義されます。 ジオメトリ軸での合計工具長補正を計算するために、ベース寸法/アダプタ寸法工具長補正も加算されます。

参照先

工具パラメータ 3 ～ 5(工具長 1 ～ 3)への工具寸法(工具長)の入力と、3 つのジオメトリ軸におけるそれらの計算方法の詳細は、「操作説明書」を参照してください。

19.4.5 工具パラメータ 6 ～ 11: ジオメトリ - 工具形状

意味

工具の形状は、工具パラメータ 6 ～ 11 を使用して定義されます。このデータはジオメトリ工具径補正に必要です。

ほとんどの場合、工具パラメータ 6(工具径 1)のみが使用されます。

工具パラメータ	意味	使用法	
6	工具径 1	工具半径は、次の工具タイプで指定する必要があります。	
		1xx	フライス工具
		5xx	旋削工具 工具パラメータ 2:刃先位置 (ページ 1801)旋削工具でも指定する必要があります。
		注 工具タイプ 2xx:穴あけ工具には工具半径は指定する必要はありません。	
7	工具径 2	使用しない	
8	工具長 4	使用しない	
9	工具長 5	使用しない	
10	工具角度 1	使用しない	
11	工具角度 2	使用しない	

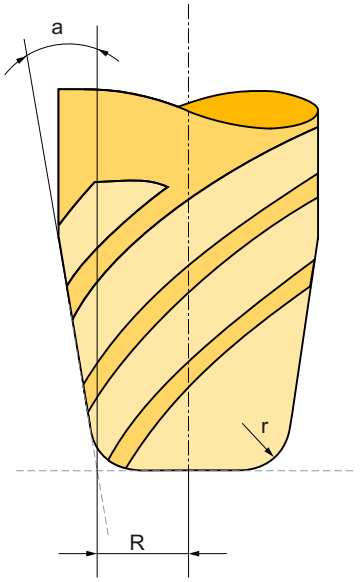
輪郭工具での 2 次元 TRC

複数の工具刃先のある輪郭工具の定義では、最小と最大制限角度も入力できます。どちらの制限角度も、それぞれ刃先中心点から刃先基準点へのベクトルに関連し、反時計回りで数えられます。

工具角度 1	1 工具刃先毎の最小制限角度
工具角度 2	1 工具刃先毎の最大制限角度

3 次元正面削り

3 次元正面削りの工具記述に関連する工具パラメータは、使用される工具タイプに依存します。したがって、たとえば、ボールエンドミルの場合は工具パラメータ 6 のみが関連し、コーナ半径ありのテーパ刃エンドミルの場合はさらに工具パラメータ 7 および 11 が関連します。



R　工具パラメータ 6:シャンク半径

R　工具パラメータ 7:コーナ半径

a　工具パラメータ 11:(テーパ工具の)側線と工具長手軸間の角度

図 19-8　テーパ刃ラジアスエンドミルの例を使用した 3 次元正面削りの工具記述

参照先

工具パラメータ 6 ～ 11 への工具形状(工具径補正での半径)の入力と、3 つのジオメトリ軸におけるジオメトリ工具径補正によるそれらの計算方法の詳細は、次を参照してください。

- 『プログラミングマニュアル、基本編』、「工具補正」「2.5 D 工具補正」の章
- 『機能マニュアル、応用機能』、「W5:3 次元工具径補正」の章

3 次元正面フライス加工については、以下の資料を参照してください。

- 『機能マニュアル、応用機能』、「W5:3 次元工具径補正」の章

19.4.6 工具パラメータ 12 ～ 14: 摩耗 - 工具長

説明

工具のサイズの定義にはジオメトリ工具長補正(工具パラメータ 3 ～ 5)が使用されますが、摩耗工具長補正を使用して、有効な工具サイズの変化を修正できます。

次の原因により、有効な工具寸法が変動することがあります。

- 工具測定装置の工具取付器具と工作機械の工具取付器具の相違
- 耐用年数内に加工によって発生する工具の摩耗
- 仕上げ代の定義

有効な工具サイズ

ジオメトリ工具長補正(工具パラメータ 3 ～ 5)と摩耗工具長補正(工具パラメータ 12 ～14)が合計されて(ジオメトリ工具長 1 が摩耗工具長 1 に加算されるなど)、有効な工具のサイズになります。

19.4.7 工具パラメータ 15 ～ 20: 摩耗 - 工具形状

説明

工具の形状の定義にはジオメトリ工具径補正(工具パラメータ 6 ～ 11)が使用されますが、摩耗工具径補正を使用して、有効な工具形状の変化を修正できます。

次の原因により、有効な工具寸法が変動することがあります。

- 耐用年数内に加工によって発生する工具の摩耗
- 仕上げ代の定義

有効な工具形状

ジオメトリ工具径補正(工具パラメータ 6 ～ 11)と摩耗工具径補正(工具パラメータ 15 ～ 20)が合計されて(ジオメトリ工具径 1 が摩耗工具径 1 に加算されるなど)、有効な工具の形状になります。

19.4.8 工具パラメータ 21 ～ 23: 工具ベース寸法/アダプタ寸法

説明

工具アダプタの基準点(工具サイズ)が工具ホルダの基準点と異なる場合は、工具ベース寸法/アダプタ寸法を使用できます。

これは、次の場合に該当します。

- 工具と工具アダプタが別個に測定されるが、1 つのユニットとして機械に取り付けられている(工具サイズとアダプタサイズが刃先に別々に入力される)場合
- 工具が別の位置(垂直および水平主軸など)にある第 2 の工具取付器具で使用される場合
- 工具タレットの工具取付器具が別の位置にある場合

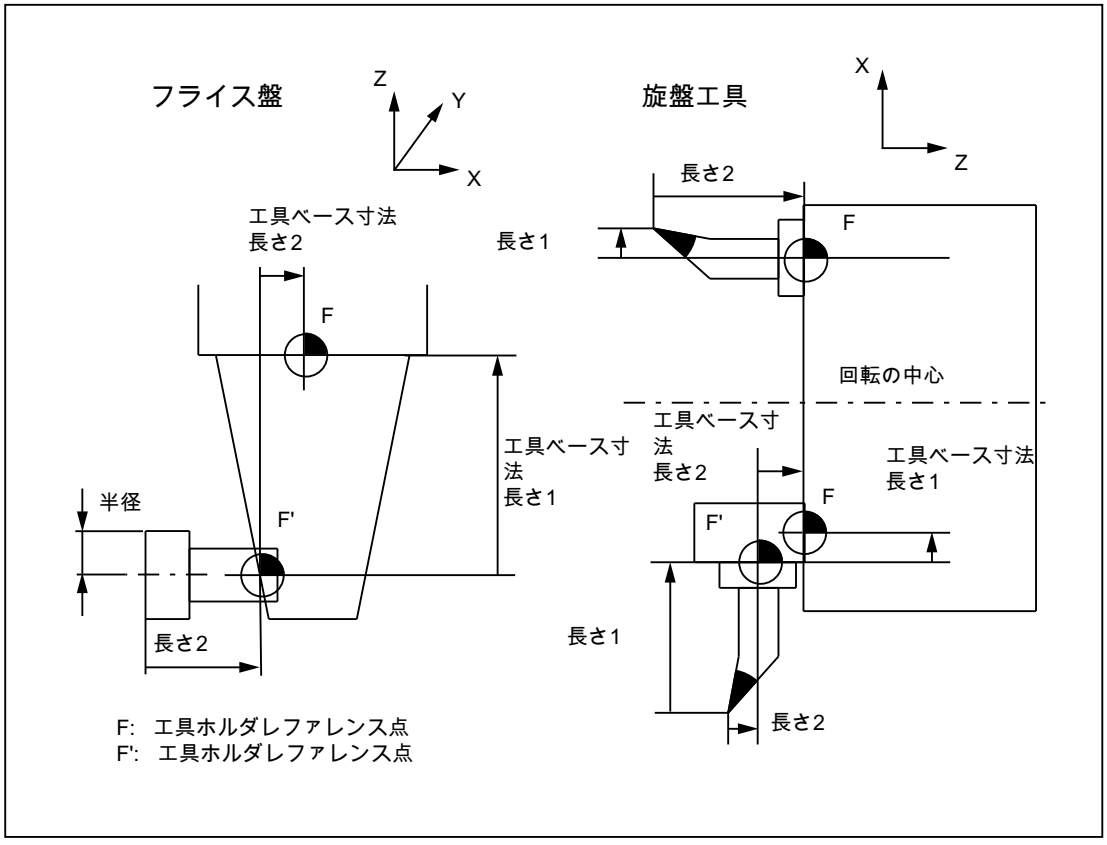


図 19-9 ベース寸法/アダプタ寸法 TLC の適用例

工具基本長 1 ～ 3(工具パラメータ 21 ～ 23)

3 つのジオメトリ軸(3 次元)で工具ホルダ基準点 F と工具アダプタ基準点 F'の相違を修正するには、工具タイプにかかわりなく、3 つの基本長をすべて有効にします。 つまり、工具長補正(長さ 1)が行われたツイストドリル(工具タイプ 200)でも、3 軸で工具ベース寸法/アダプタ寸法が使用できます。

参照先

ベース寸法/アダプタ寸法工具長補正の詳細は、次の資料を参照してください。

- 『プログラミング説明書、基本編』

19.4.9 工具パラメータ 24: 逃げ角

意味

レリーフカットのある移動動作が発生する特定の旋削サイクルでは、経路干渉の可能性について有効な工具の工具逃げ角が監視されます。

値の範囲

この角度(符号なしで 0 ～ 90°)は、工具逃げ角としてパラメータ 24 に入力されます。

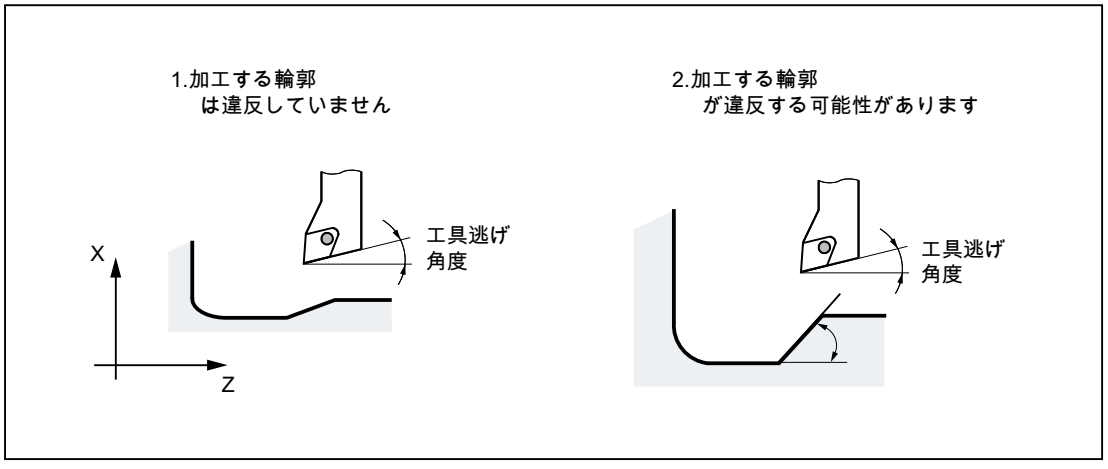


図 19-10 レリーフカット時の旋削工具の工具逃げ角

加工タイプ、縦または径方向

工具逃げ角は、加工のタイプ(縦方向または端面)に応じて異なる方法で入力されます。工具が縦方向加工と端面加工の両方に使用される場合、それぞれ工具逃げ角に 2 つの刃先を入力してください。

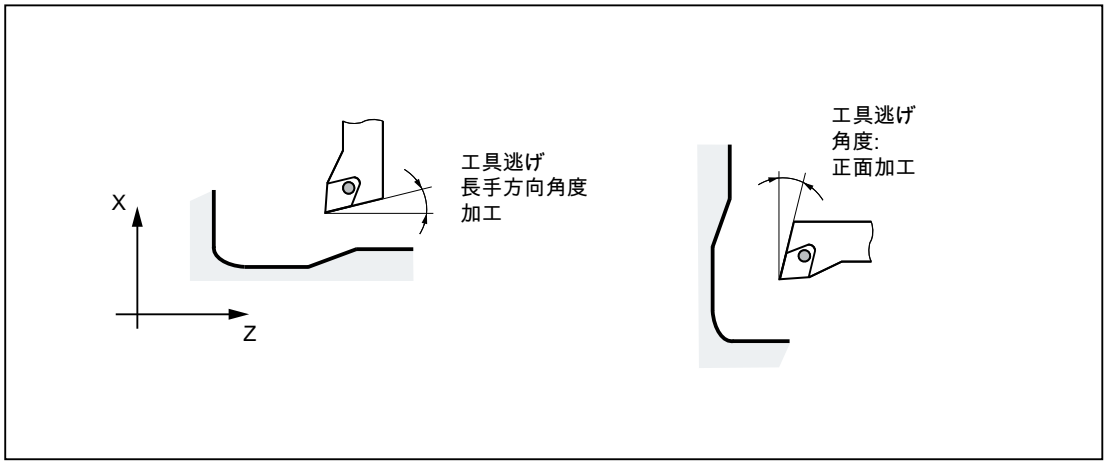


図 19-11　縦方向加工と端面加工の工具逃げ角

注記

0 の工具逃げ角(工具パラメータ 24)を入力すると、旋削サイクル内でレリーフカットが監視されません。

参照先

工具逃げ角の詳細は、次を参照してください。

- 『プログラミング説明書 サイクル編』

19.4.10 関連する工具刃先位置を持つ工具

関連する刃先位置を持つ工具については、次の規則に従ってください。

- ブロック開始時とブロック終了時の工具刃先中心点を結ぶ直線を使用して、アプローチブロックおよび後退ブロックとの交点が計算されます。 工具刃先基準点と工具刃先中心点の差異がこの移動に重ね合わせられます。
  KONT によるアプローチか後退、またはその両方について、この移動がアプローチまたは後退動作の直線サブブロックで重ね合わせられます。 したがって、関連する刃先位置がある工具とそれがない工具の形状条件は同じです。
- 円弧ブロックおよび 4 次以上の分母を持つ有理多項式を含む移動ブロックでは、工具径補正が有効な工具の変更は、工具刃先中心点と工具刃先基準点の距離が変更される場合には許可されません。 他のタイプの補間では、座標変換が有効な場合に工具を変更できるようになりました(TRANSMIT など)。
- 工具の向きが可変である工具径補正では、工具刃先基準点から工具刃先中心点への変換を単純なゼロオフセットによって実行できなくなりました。 したがって、関連する刃先位置を持つ工具は、3 次元外周削りには使用できません(アラームが出力されます)。

注記

これは、正面フライス加工には該当しません。正面フライス加工では、どちらにしても、関連する刃先位置のない定義済みの工具タイプのみが許可されるからです (明確に承認されていないタイプの工具は、指定された半径のボールエンドミルとして扱われます。 刃先位置のパラメータは無視されます)。

19.5 2 次元工具径補正(2D-WRK)

19.5.1 一般情報

注記

工具径補正(TRC)については、次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』
工具径補正(TRC)とその特色全体の技術的説明は、プログラミングガイドのみに記載されています。

TRC を使う理由

パートプログラムにプログラム指令されるワークの輪郭(形状)は、加工で使用される工具に依存しないことが望まれます。 そのためには、工具長と工具径の値を現在のオフセットメモリから抽出する必要があります。 工具径補正を使用して、現在の工具径からプログラム指令輪郭への等間隔の軌跡を計算できます。

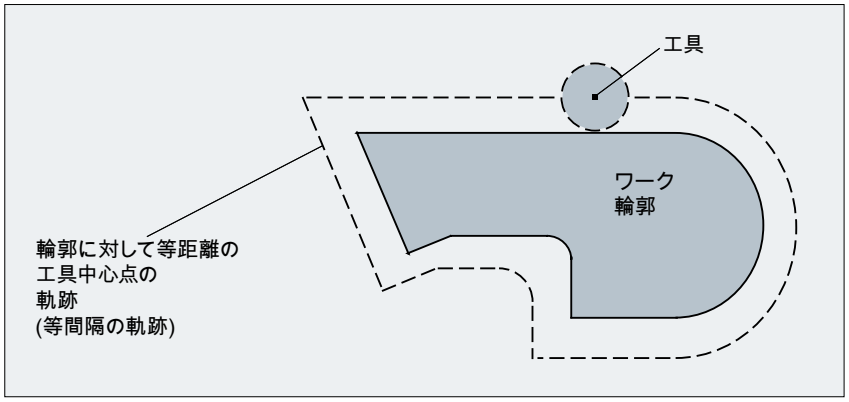


図 19-12　等間隔の軌跡によるワーク輪郭(形状)

平面上の TRC

次のタイプの補間では、現在の平面上(G17 ～ G19)で TRC が有効です。

● 直線補間	...	G0、G1
● 円弧補間	...	G2、G3、CIP
● ヘリカル補間	...	G2、G3
● スプライン補間	...	ASPLINE、BSPLINE、CSPLINE
● 多項式補間	...	POLY

19.5.2 TRC の選択(G41/G42)

補正の方向

プログラムされた輪郭に対して等間隔の軌跡が TRC によって計算されます。 補正は、移動の方向に向かって、プログラムされた輪郭の左側または右側で実行できます。

命令	意味
G41	移動の方向に向かって輪郭の左側での TRC
G42	移動の方向に向かって輪郭の右側での TRC
G40	TRC の解除

中間ブロック

一般に、TRC が有効な場合、選択平面内のジオメトリ軸の位置を含むプログラムブロックのみがプログラムされます。 ただし、ダミーブロックも TRC が有効な状態でプログラムできます。 ダミーブロックは、選択平面内のジオメトリ軸の位置を含まないプログラムブロックです。

- 切り込み軸の位置
- 補助機能
- その他

ダミーブロックの最大数は、次のマシンデータで定義できます。

MD20250 $MC_CUTCOM_MAXNUM_DUMMY_BLOCKS(TRC で移動動作のないダミーブロックの最大数)

特別な注意点

- TRC は G0(早送り)または G1(直線補間)を含むプログラムブロックのみで選択できます。
- 工具径補正の選択を含むブロック以前で、工具をロードし(T 機能)、工具刃先(工具補正)(D1 ～ D9)を起動してください。
- 工具径補正は、D0 の工具刃先/工具補正と一緒には選択されません。
- 工具径補正が選択されたとき、平面上でジオメトリ軸が 1 つだけプログラムされている場合、第 2 の軸が平面上に自動的に追加されます(前回のプログラム位置)。

- 工具径補正の選択を含むブロックで、選択平面にジオメトリ軸がプログラムされていないと、選択が実行されません。
- 工具径補正の選択の次のブロックで工具径補正が解除されると(G40)、選択が実行されません。
- 工具径補正が選択された場合、アプローチ動作は NORM/KONT 命令によって決定されます。

19.5.3 アプローチおよび後退動作(NORM/KONT/KONTC/KONTT)

NORM および KONT

NORM および KONT 命令を使用して、アプローチ動作(G41/42 による工具径補正の選択)と後退動作(G40 による工具径補正の解除)を制御できます。

命令	意味
NORM	開始点/終点の通常設定(初期設定)
KONT	開始点/終点で輪郭に沿って移動
KONTC	一定曲率によるアプローチ/後退
KONTT	一定接線によるアプローチ/後退

特記事項

- KONT は、工具の移動開始位置が輪郭の背後にある場合にのみ NORM と異なります。

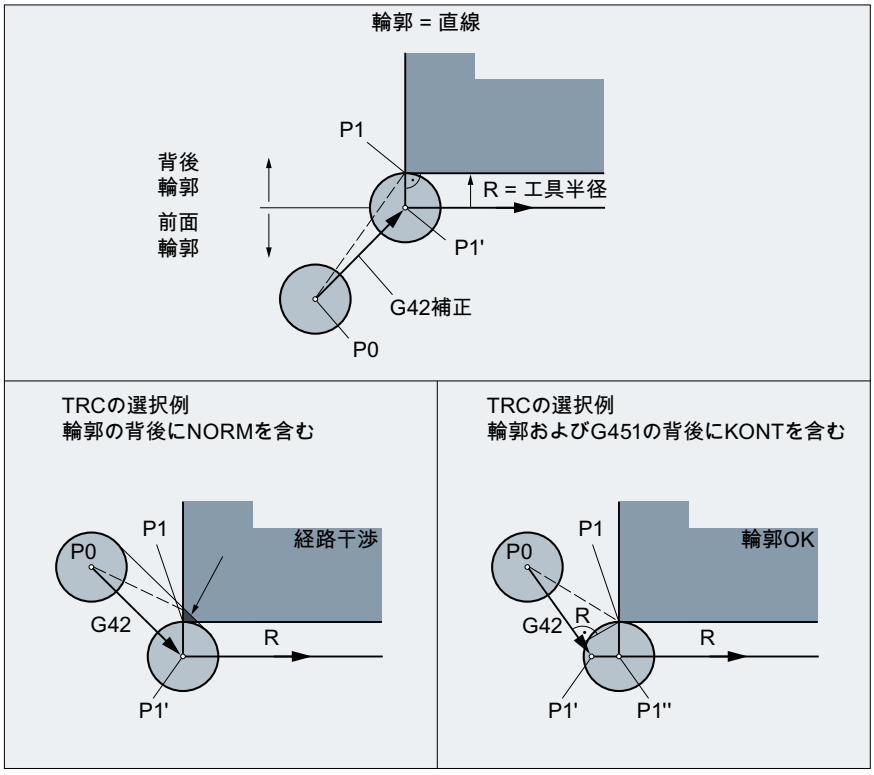


図 19-13　輪郭の前面および背後での KONT または NORM による TRC 選択の例

- KONT と G450/G451(外周コーナでのコーナ挙動)は、一緒に作用し、TRC によるアプローチおよび後退動作を決定します。
- 工具径補正が解除された場合、後退動作は NORM/KONT 命令によって決定されます。

必要条件

アプローチおよび後退ブロックは、次の 2 つの種別では多項式になります。したがって、これらは多項式補間をサポートする制御装置のみで使用できます。

- KONTT
  KONTT では、輪郭へのアプローチおよび輪郭からの後退は一定接線で行われます。ブロック遷移時の曲率は、通常は一定ではありません。
- KONTC
  KONTC では、遷移時に接線だけでなく曲率も一定であるため、起動/解除時に**加速度の不連続**がなくなります。

KONTC には KONTT の特性が含まれますが、KONTC で要求される一定曲率によって望ましくない輪郭が生成される場合があるため、一定接線タイプの KONTT だけが自由に使用できます。

軸

3 つの軸すべてで連続性の条件が守られます。したがって、アプローチ/後退の補正平面に対して垂直な同時軌跡要素をプログラムできます。

KONTT/KONTC を含む元のアプローチおよび後退ブロックでは、**直線ブロック**のみが許可されます。プログラムされたこれらの直線ブロックは、制御装置内で対応する多項式曲線に置換されます。

例外

KONTT および KONTC は、工具径補正の 3 次元タイプ (CUT3DC、CUT3DCC、CUT3DF) では使用できません。

これらをプログラム指令すると、コントローラは、エラーメッセージを表示せずに、内部で NORM に切り替えます。

KONTC の例

一定曲率でのアプローチおよび後退の典型的な用途を後述の 2 つの図で示します。

円弧中心点から開始して、一周円へアプローチします。ブロック終点のアプローチ円の方向と曲率半径は、その次の円弧の値と同じです。切り込みは、アプローチ/後退ブロックの両方で同時に、Z 方向へ行われます。

関連する NC プログラム部分は次のとおりです。

```
$TC_DP1[1,1] = 121                                  フライス工具
$TC_DP6[1,1]=10                                     半径 10 mm
N10 G1 X0 Y0 Z60 G64 T1 D1 F10000
N20 G41 KONTC X70 Y0 Z0
N30 G2 I-70                                         一周円
N40 G40 G1 X0 Y0 Z60
N50 M30
```

意味:

この例では、半径 70 mm の一周円が X/Y 平面上で加工されます。工具の半径は 10 mm であるため、結果として工具中心点の軌跡によって半径 60 mm の円が描かれます。開始

点/終点は X0 Y0 Z60 にあるため、Z 方向の移動は補正平面上のアプローチ/後退動作と同時に行われます。

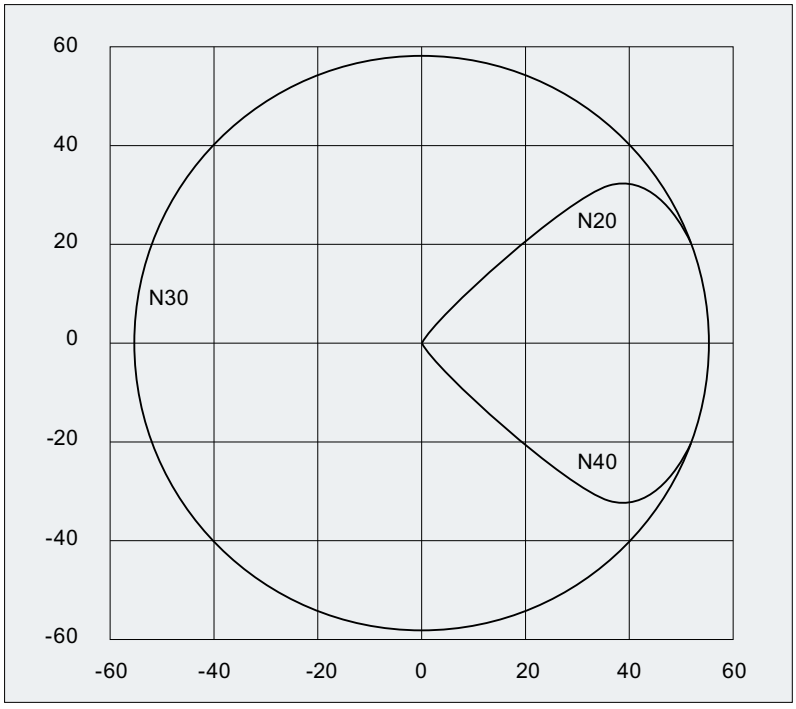


図 19-14　一周円の内側加工における一定曲率でのアプローチおよび後退:X-Y 平面での投影

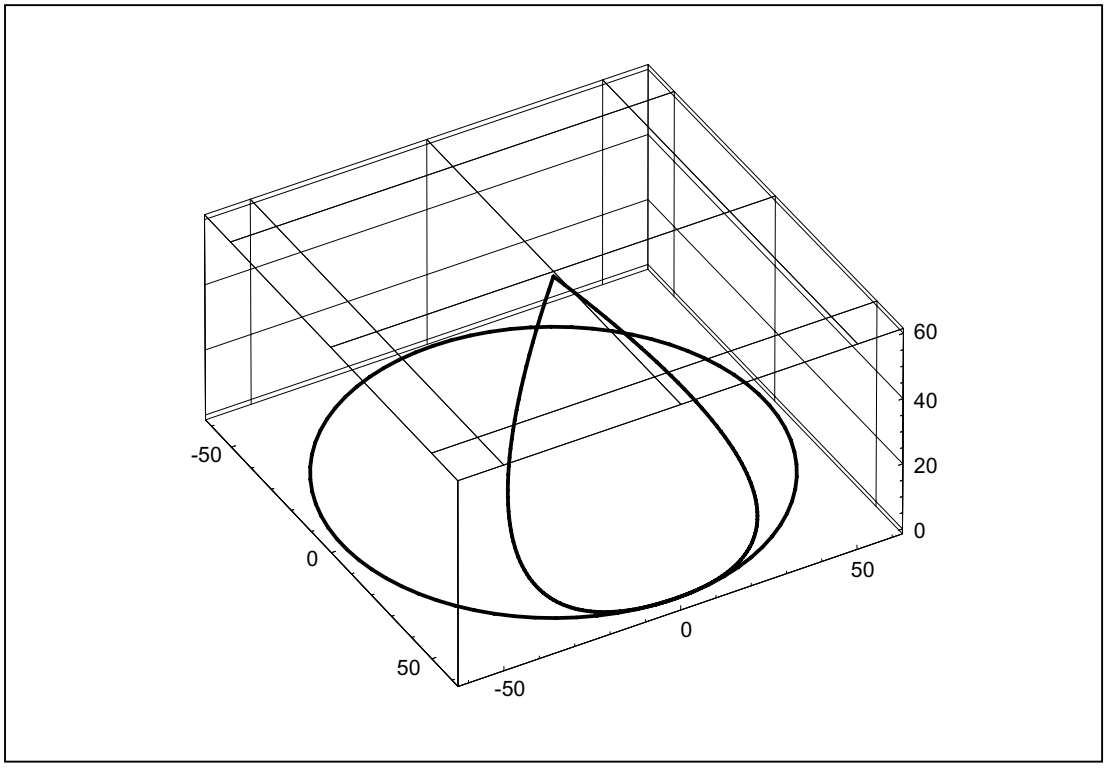


図 19-15　一周円の内側加工における一定曲率でのアプローチおよび後退:3 次元表示

KONTT と KONTC の比較

以下の図は、KONTT と KONTC 間のアプローチ/後退動作の違いを示します。X0 Y-40 の中心点を中心とする半径 20 mm の円は、外側半径 20 mm の工具で補正されます。したがって、工具の中心点は、半径 40 mm の円弧軌跡に沿って移動します。アプローチブロックの終点は X40 Y30 にあります。円弧ブロックと後退ブロック間の遷移は原点で行われます。KONTC に関連した曲率はその後も継続するため、後退ブロックは最初に、負の Y 成分で移動を実行します。これは、多くの場合、望ましくない状況です。この動作は、KONTT 後退ブロックでは発生しません。ただし、このブロックの場合は、ブロック遷移のとき加減速の不連続変化が発生します。

KONTT または KONTC ブロックが後退ブロックではなくアプローチブロックの場合、輪郭はまったく同じになりますが、単に反対方向に加工されます。つまり、**アプローチ動作と後退動作は対称形です。**

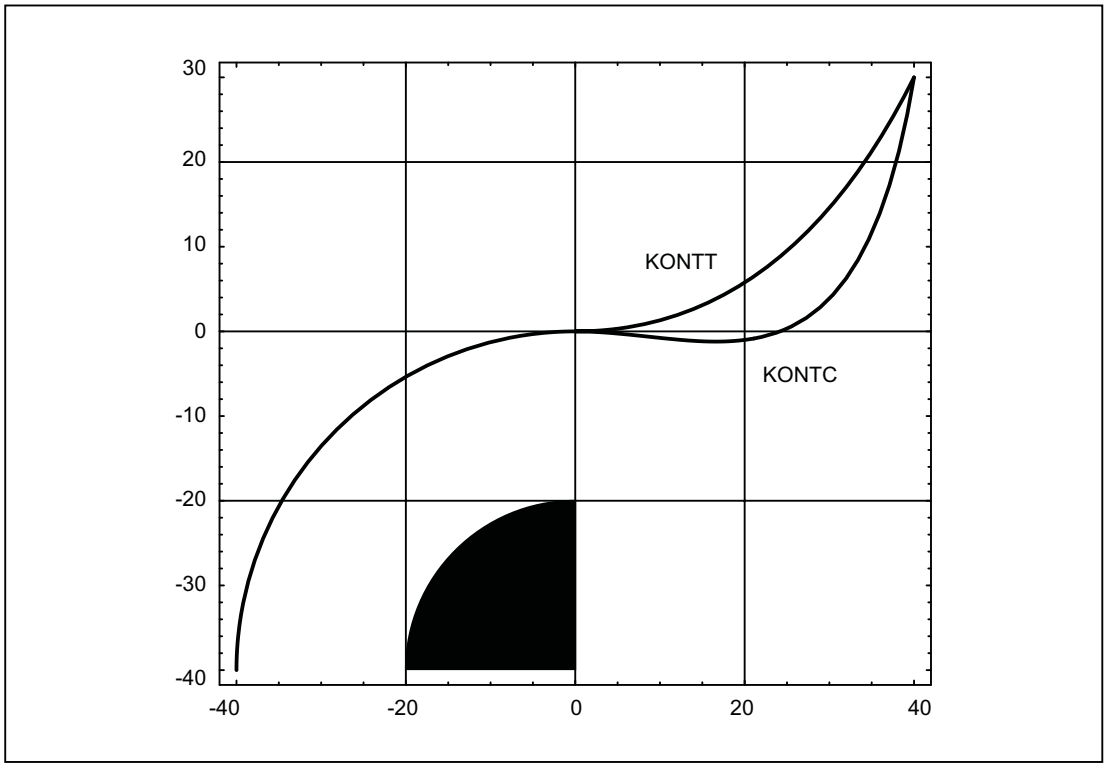


図 19-16　KONTT と KONTC の違い

注記

この図は、KONTC の後退/アプローチでは、輪郭の四分円につながる、たとえば X20 Y-20 までの直線よりも X0 Y0 の方へ侵入することを示しています。

19.5.4 滑らかなアプローチと後退

19.5.4.1 機能

説明

SAR (Smooth Approach and Retraction)機能は、輪郭の開始点の位置にかかわりなく、開始点への接線アプローチを実現するために使用されます。

さまざまな追加パラメータを使用して、アプローチ動作を変化させ、特殊なニーズに適合させることができます。

「滑らかなアプローチ」と「滑らかな後退」の 2 つの機能は、ほぼ対称形です。 したがって以降のセクションでは、アプローチの詳細な説明に限定します。後退に影響を及ぼす相違については、特に言及します。

動作区分

次の位置により、滑らかな後退とアプローチでは、最大 4 つの動作区分が存在します。

- 移動の開始点 P_0
- 中間点 P_1、P_2 および P_3
- 終点 P_4

点 P_0、P_3 および P_4 は常に定義されます。 中間点 P_1 および P_2 は、定義されたパラメータと幾何学上の条件に従って省略できます。

後退時には、これらの点を逆方向に通過します。つまり、P_4 から始まり、P_0 で終わります。

19.5.4.2 パラメータ

「滑らかなアプローチと後退」機能の動作は、最大 9 つのパラメータによって決定されます。

アプローチおよび後退経路を定義するためのノンモーダル G コード

この G 命令は省略できません。

- G147:直線でのアプローチ
- G148:直線での後退
- G247:四分円でのアプローチ
- G248:四分円での後退
- G347:半円でのアプローチ
- G348:半円での後退

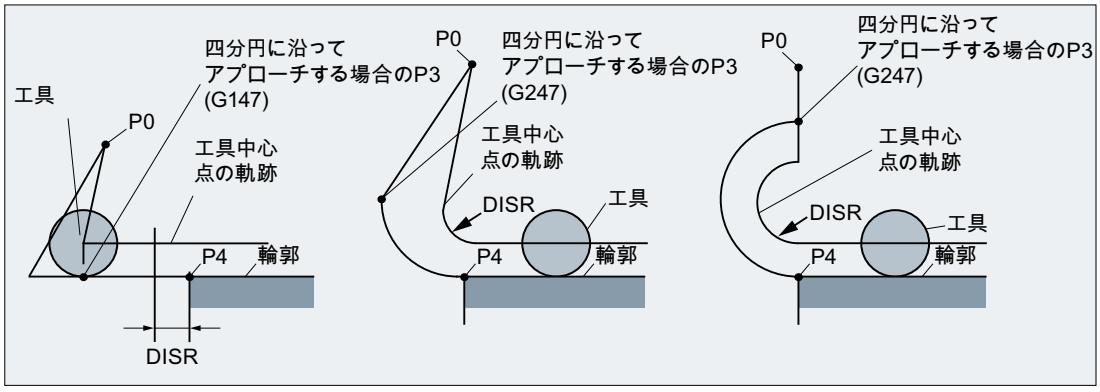


図 19-17 G147 ～ G347 および DISR に依存する、工具径補正の同時起動をともなうアプローチ動作

アプローチおよび後退経路を定義するためのモーダル G 命令

この G 命令は、アプローチ経路が四分円または半円の場合にのみ該当します。アプローチおよび後退方向は次のように決定できます。

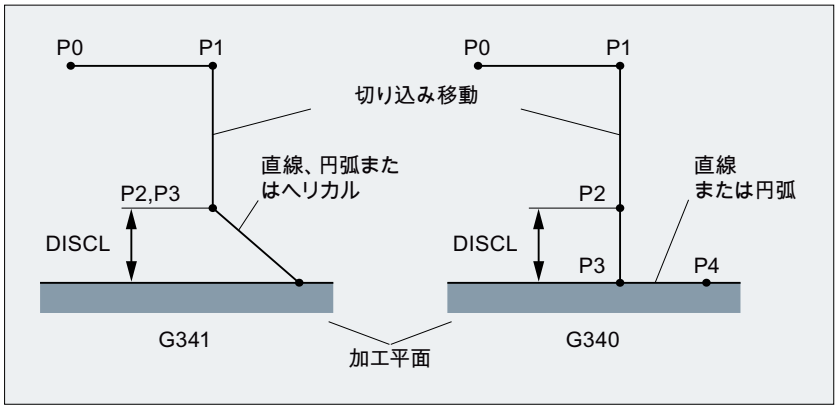
- G140:
  有効な工具径補正を使用したアプローチおよび後退方向の定義(G140 は初期設定値です)。
  工具径が正の場合:
  - G41 が有効 → 左からアプローチ
  - G42 が有効 → 右からアプローチ
    工具径補正が有効(G40)でない場合、応答は G143 と同じになります。この場合、アラームは出力されません。有効な工具の半径が 0 の場合、どちら側からアプローチおよび後退するかは、工具半径が正の場合と同様に決定されます。
- G141:
  輪郭に左からアプローチ、または左へ後退

- G142:
  輪郭に右からアプローチ、または右に後退
- G143:
  アプローチ方向を自動的に決定。つまり、次のブロックの開始点(P_4)での接線に対して開始点が位置する側から輪郭にアプローチします。

注記

これに応じて、**後退**時には前のブロックの終点での接線が使用されます。後退時に終点がプログラムに明示されていない場合、つまり終点が自動的に決定される場合は、アプローチ側と終点の位置に相互依存関係が存在するため、後退時に G143 を使用できません。ここで G143 がプログラムされている場合、アラームが出力されます。G140 が有効なとき、無効な工具径補正の結果として G143 への自動切り替えが発生する場合にも、同じことが当てはまります。

開始点から終点までの移動の個別ブロックへの分割を定義するモーダル G 命令(G340、G341)



G34 P_0 から P_4 までのアプローチ特性を図に示します。

0: G247 または G347 が有効であり(四分円または半円)、 開始点 P_3 が終点 P_4 によって定義される加工面から離れている場合は、円ではなくヘリカルが挿入されます。点 P_2 は定義されないか、または P_3 と一致します。

円弧平面またはヘリカル軸は、SAR ブロックで有効な平面(G17 - G19)によって決定されます。つまり、次のブロックで、開始接線自体ではなく接線の投影を使用して円が定義されます。

点 P_0 から点 P_3 への移動は、SAR ブロックの前に有効だった速度で、2 本の直線に沿って行われます。

G34 P_0 から P_4 までのアプローチ特性を図に示します。

1: P_3 と P_4 は加工平面内にあるため、G247 または G347 ではヘリカルではなく常に円弧が挿入されます。

図 19-18　G340/G341 に依存するアプローチ動作の処理

注記

有効な平面 G17 - G19 の位置が関連するすべての場合(円弧平面、ヘリカル軸、有効な平面に垂直な送り動作)に、有効な回転フレームが含まれます。

DISR

DISR は直線アプローチ線の長さ、またはアプローチ円弧の半径を指定します。

直線での後退/アプローチ

直線に沿ったアプローチ/後退時には、DISR によって、刃先から輪郭の開始点までの距離が指定されます。つまり、TRC が有効なときの直線の長さは、工具半径とプログラムされた DISR の値の合計として計算されます。

下記の場合はアラームが表示されます。

- DISR が負であり、値が工具径より大きい場合(算出されるアプローチ線の長さが 0 以下)

円弧による後退/アプローチ

円弧によるアプローチ/後退の
DISR では、常に工具中心点軌跡の半径が指定されます。工具径補正が起動されると、内部で円弧が生成されます。この円弧の半径は、この場合にも工具中心軌跡がプログラムされた半径から得られる寸法を持ちます。

次の場合には、円弧でのアプローチおよび後退時にアラームが出力されます。

- 内部で生成される円の半径が 0 または負の場合
- DISR がプログラムされていない場合
- 半径の値が 0 以下の場合

DISCL

DISCL では、点 P2 から加工面までの距離が指定されます。

点 P_2 の位置が円弧平面に対して垂直な軸のアブソリュート指令によって指定される場合は、この値を DISCL = AC(....) という形でプログラムしてください。

DISCL がプログラムされていない場合、点 P_1、P_2 および P_3 は G340 では同じであり、アプローチの経路は P_1 から P_4 に向かって描かれます。

DISCL によって定義される点が、P_1 と P_3 の間にあることがシステムによって確認されます。つまり、加工平面に対して垂直な要素のあるすべての移動(切り込み方向移動、P_3 から P_4 へのアプローチ動作)で、この要素の符号は同じでなくてはなりません。方向の反転はできません。この条件に違反すると、アラームが出力されます。

方向の反転が検出された場合、次のマシンデータによって定義された許容値は認められます。

MD20204 $MC_WAB_CLEARANCE_TOLERANCE(SAR 時の方向反転)

ただし、P_2 が P_1 と P_3 によって定義される範囲外にあり、誤差がこの許容値以下の場合、P_2 は P_1 か P_3、またはその両方によって定義される平面内にあるとみなされます。

例:

点 P_1 の位置 Z=20 から、G17 でのアプローチが行われます。P_3 によって定義される SAR 平面は Z=0 にあります。したがって、DISCL によって定義される点は、これら 2 つの点の間になくてはなりません。MD20204=0.010。P_2 が 20.000 ～ 20.010 または 0 ～-0.010 の範囲内にある場合、値 20.0 または 0.0 がプログラムされているものとみなされます。P_2 の Z 位置が 20.010 より大きいか、-0.010 より小さい場合、アラームが出力されます。

加工平面に対する開始点 P_0 および終点 P_4 の相対位置に応じて、切り込み軸方向の移動が負(通常アプローチとして)または正(通常後退として)の方向に実行されます。つまり、G17 では、終点 P_4 の Z 要素が開始点 P_0 の Z 要素より大きいことが受け入れられます。

終点 P_4 (または、後退の場合の P_0)の通常プログラミング: X...Y...Z...

アプローチについて可能な終点 P_4 のプログラミング方法

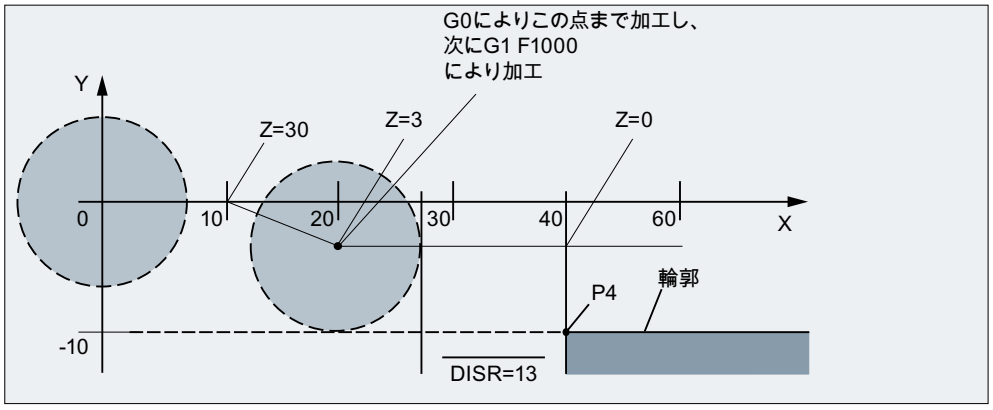
終点 P_4 は、SAR ブロック自体にプログラムできます。

P_4 は次の移動ブロックの終点によって決定されます。

ジオメトリ軸の移動なしで、その他のブロック(ダミーブロック)を SAR ブロックと次の移動ブロックの間に挿入できます。

加工平面で 1 つ以上のジオメトリ軸(G17 では X または Y)がプログラムされている場合、終点は SAR ブロック自体にプログラムされているとみなされます。加工平面に対して垂直な軸(G17 では Z)の位置のみが SAR ブロックでプログラムされている場合、この要素は SAR ブロックから取得されますが、平面内の位置は次のブロックから取得されます。この場合、加工平面に対して垂直な軸が次のブロックでもプログラムされていると、アラームが出力されます。

例:



プログラムコード	コメント
$TC_DP1[1,1] = 120	;Milling tool T1/D1
$TC_DP6[1,1] =7	; 半径 7 mm の工具
N10 G90 G0 X0 Y0 Z30 D1 T1	
N20 X10	
N30 G41 G147 DISCL=3 DISR=13 Z=0 F1000	
N40 G1 X40 Y-10	
N50 G1 X50	
...	
...	

N30/N40 は次のように置換できます。

```
N30 G41 G147 DISCL=3 DISR=13 X40 Y-10 Z0 F1000
```

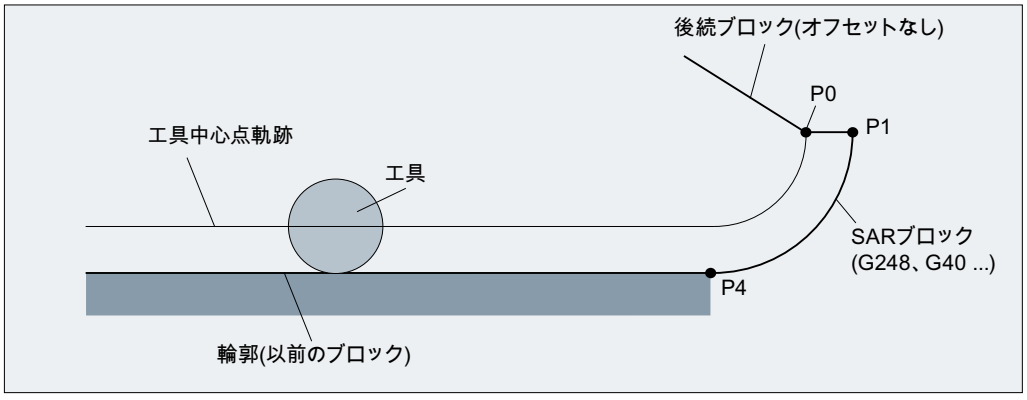
または

```
N30 G41 G147 DISCL=3 DISR=13 F1000
N40 G1 X40 Y-10 Z0
```

後退について可能な終点 P_0 のプログラム方法

終点は、プログラムされている軸の数にかかわりなく、常に SAR ブロックから取得されます。次の状態を区別します。

1. SAR ブロックでジオメトリ軸がプログラムされていない場合。
   この場合は、輪郭は点 P_2(または、P_1 と P_2 が一致する場合は P_1)で終了します。加工平面を記述する軸内の位置は、後退輪郭によって決定されます(直線または円弧の終点)。これに対して垂直な軸要素が DISCL によって定義されます。この場合、DISCL = 0 であれば、移動がすべて平面内で行われます。
2. 加工平面に対して垂直な軸のみが SAR ブロックでプログラムされている場合。
   この場合、輪郭は点 P_1 で終了します。他の 2 つの軸の位置は、1 と同じ方法で決定されます。



TRC の同時解除をともなう SAR での後退
SAR 後退ブロックが工具径補正の解除にも使用される場合、1 と 2 のケースでは、後退輪郭の終点で工具径補正が解除されるときに移動が生成されないように、P_1 から P_0 までの追加軌跡が挿入されます。つまり、この点は、補正された輪郭上の位置ではなく、工具中心点を表します。

3. 加工平面の 1 つ以上の軸がプログラムされている場合。
   加工平面の 2 つ目の軸を、前のブロックにおけるその軸の最終位置からモーダルに決定できます。加工平面に対して垂直な軸がプログラムされているかどうかに応じて、この軸の位置が 1 または 2 の説明のとおりに生成されます。この方法で生成される位置によって、終点 P_0 が定義されます。
   プログラムされた点 P_0 によって、全輪郭の終点での工具中心点の位置が既に直接定義されているため、工具径補正を解除するための特別な対策は不要です。
   SAR 輪郭の開始点と終点(P_0 と P_4)が、アプローチおよび後退時に一致してもかまいません。

前のブロック(通常は G0)の速度

点 P_0 から点 P_2 へのすべての移動、つまり加工平面に対して平行な移動と、安全間隔までの切り込み軸方向の動作の一部は、この速度で行われます。

FAD による送り速度のプログラミング

FAD のプログラム方法	
G340	P_2 または P_3 から P_4 への送り速度
G341	P_2 から P_3 への、加工平面に対して垂直な切り込み方向軸移動の送り速度

FAD がプログラムされていない場合、速度を定義する F 命令が SAR ブロックで定義されないと、前のブロックからモーダルに有効になった速度で輪郭のこの部分を移動します。

プログラムされる動作

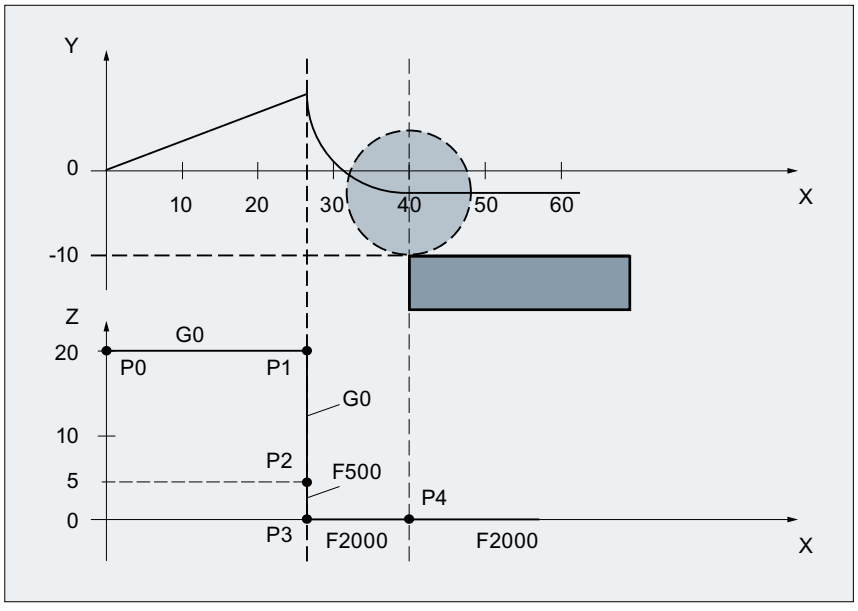
FAD=0 または負 → アラーム出力

FAD=... → プログラムされた値は、グループ 15 の有効な G 命令(送りタイプ G93、G94 など)に従って動作します。

FAD=PM(...) → プログラムされた値は、グループ 15 の有効な G 命令にかかわりなく、毎分送り(G94 など)として解釈されます。

FAD=PR(...) → プログラムされた値は、グループ 15 の有効な G 命令にかかわりなく、毎回転送り(G95 など)として解釈されます。

例:



プログラムコード	コメント
$TC_DP1[1,1] = 120	;Milling tool T1/D1
$TC_DP6[1,1] =7	; 半径 7 mm の工具
N10 G90 G0 X0 Y0 Z20 D1 T1	
N20 G41 G341 G247 DISCL=AC(5) DISR=13FAD 500 X40 Y-10 Z=0 F2000	
N30 X50	
N40 X60	
...	

送り速度 F のプログラミング

この送り速度値は、点 P_3(または、FAD がプログラムされていない場合は点 P_2)から有効です。SAR ブロックで F 命令がプログラムされていない場合、前のブロックの速度が有効です。FAD によって定義される速度は、後続のブロックには使用されません。

19.5.4.3 速度

アプローチ時の速度

次の両方のアプローチ図では、SAR ブロックの次のブロックで新しい速度がプログラムされていないと仮定しています。 そうでない場合は、点 P_4 の後で新しい速度が有効になります。

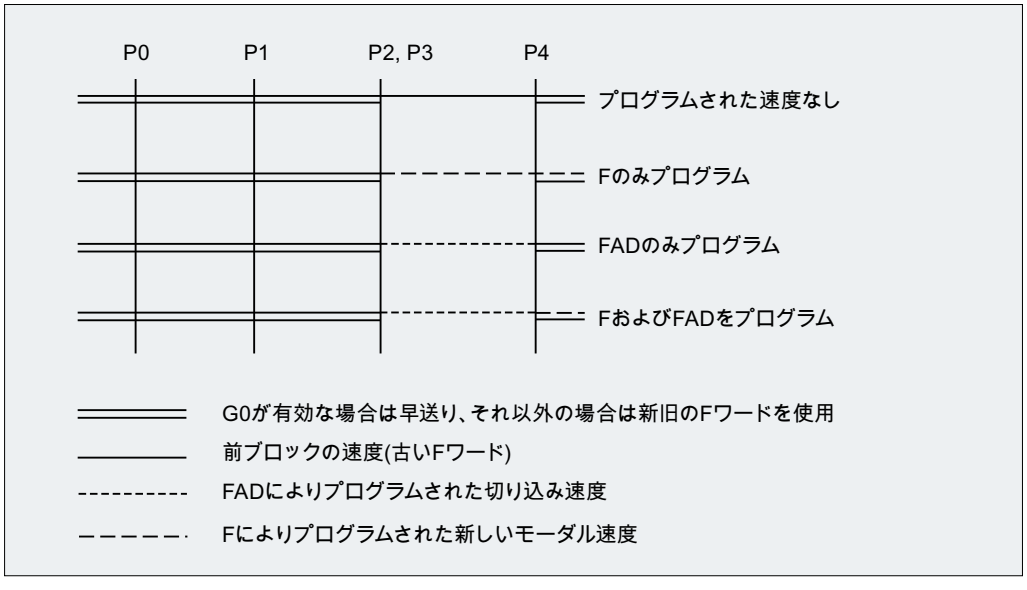


図 19-19　G340 でのアプローチ時の SAR サブブロックにおける速度

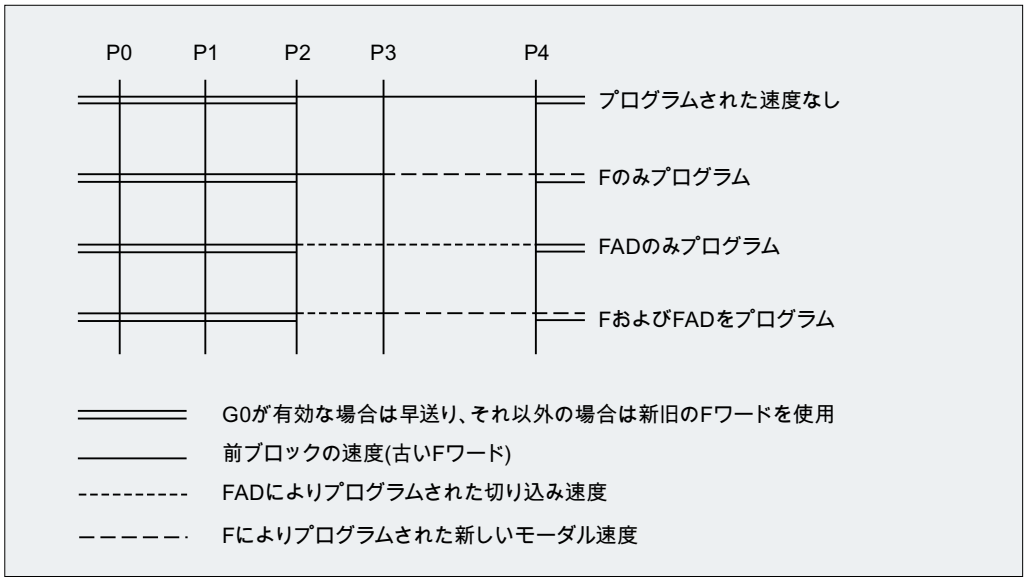


図 19-20　G341 でのアプローチ時の SAR サブブロックにおける速度

後退時の速度

後退時には、前のブロックからモーダルで有効な送り速度と、SAR ブロックでプログラムされた送り速度値の役割が入れ替わります。つまり、後退輪郭自体(直線、円弧、ヘリカル)を前のブロックからモーダルで有効な送り速度で移動し、F ワードでプログラムされた新しい速度が点 P_2 から P_0 までに適用されます。

平らな後退が有効であり、FAD がプログラムされている場合は、P_3 から P_2 への軌跡は FAD で移動し、そうでない場合は旧速度で移動します。 P_4 から P_2 への軌跡には、前のブロックでプログラムされた最後の F 命令が常に適用されます。 これらのブロックでは、G0 による影響はありません。

P_2 から P_0 への移動は、SAR ブロックでプログラムされた F 命令によって行われるか、F 命令がプログラムされていない場合は、前のブロックからのモーダル F 命令によって行われます。 これは、G0 が無効であることを条件として適用されます。

ブロック内での P_2 から P_0 までの後退時に早送りを使用する必要がある場合は、SAR ブロックより前または SAR ブロック自体で G0 を有効にしてください。 SAR ブロック自体に追加の F 命令がプログラムされている場合、それは無効です。 ただし、後続のブロックでは、それがモーダルで有効なままとなります。

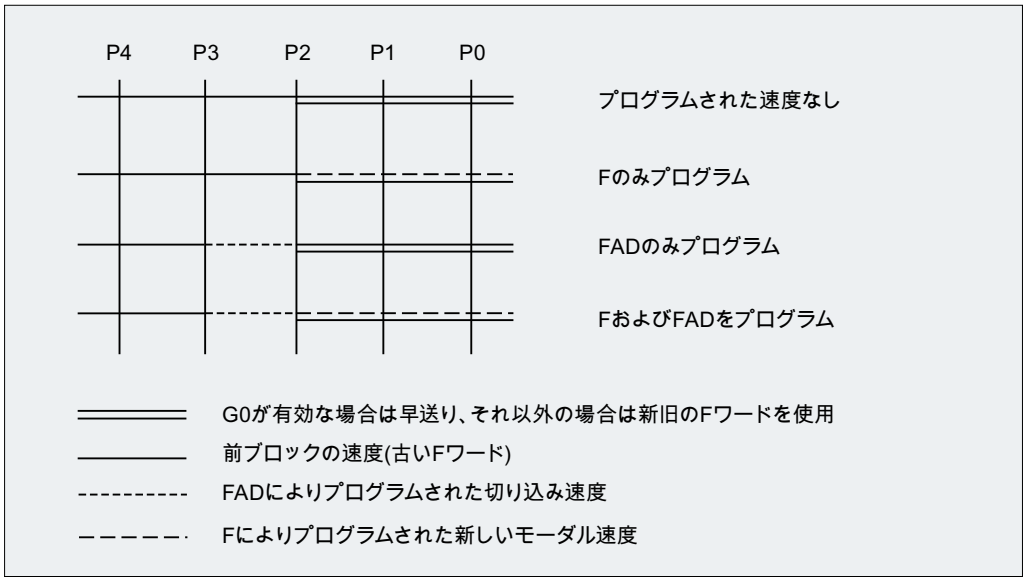


図 19-21　後退時の SAR サブブロックにおける速度

19.5.4.4　システム変数

点 P_3 および P_4 は、アプローチ時に WCS でシステム変数として読み取ることができます。

$P_APR:	WCS での P_3(開始点)の読み取り	
$P_AEP:	WCS での P_4(輪郭開始点)の読み取り	
$P_APDV	=1	$P_APR と$P_AEP の内容が有効な場合、つまりプログラムされた最新の SAR アプローチブロックに属する位置データがこれらに含まれる場合
	=0	旧 SAR アプローチブロックの位置が読み取られます。

SAR ブロックと読み取り操作の間に WCS が変更されても、位置データには影響を及ぼしません。

19.5.4.5 必要条件

必要条件

- SAR ブロックで任意の追加 NC 命令(補助機能出力、同期軸移動、位置決め軸移動など)をプログラム指令できます。
  これらは、アプローチ時には最初のサブブロック、後退時には最後のサブブロックで実行されます。
- 終点 P_4 が SAR ブロックではなく後続の移動ブロックから取得される場合は、SAR 輪郭自体(直線、四分円または半円)は、このブロックで移動します。
  この場合、元の SAR ブロックの最終サブブロックには、ジオメトリ軸の移動情報は含まれません。 ただし、このブロックでそれ以上の動作(単独軸など)の実行が必要な場合があるため、このサブブロックは常に出力されます。
- 常に 2 つ以上のブロックを考慮に入れてください。
  – SAR ブロック自体
  – アプローチまたは後退方向を定義するブロック
  これら 2 つのブロックの間に、他のブロックをプログラム指令できます。
  プログラム指令できるダミーブロックの数は、下記のマシンデータで制限されます。
  MD20202 $MC_WAB_MAXNUM_DUMMY_BLOCKS(SAR での移動動作をともなわないブロックの最大数)
- アプローチブロックで工具径補正が同時に起動される場合、SAR 輪郭の最初の直線ブロックが、起動のおこなわれるブロックとなります。
  SAR 機能によって生成される輪郭全体は、工具径補正によって、明確にプログラム指令された場合(衝突検出、交点の計算、アプローチ動作 NORM/KONT)と同様に処理されます。
- 切り込み軸の動作方向と、円弧平面またはヘリカル軸の位置は、対応する有効なフレームによって回転された有効な平面(G17 ～ G19)によってのみ定義されます。
- アプローチ時には、SAR ブロックと、接線の方向を定義する次のブロックとの間に、先読み停止を挿入しないでください。
  先読み停止は、明確にプログラム指令されたか、コントローラによって自動的に挿入されたかにかかわりなく、この場合アラームの原因となります。

REPOS での動作

SAR サイクルが中断されて再位置めされた場合、そのサイクルは RMIBL で中断の地点から再開します。 RMEBL では、接続点は最後の SAR ブロックの終点です。RMBBL では、最初の SAR ブロックの開始点です。

RMIBL が DISPR とともにプログラム指令されている場合(中断点より前の距離 DISPR での再アプローチ)、再アプローチ点は中断サブブロックより前の SAR サイクルのサブブロックにすることができます。

19.5.4.6 例

例 1

次の条件が当てはまります。

- ブロック N20 で滑らかなアプローチを起動します。
- X=40(終点); Y=0; Z=0
- 四分円でおこなわれるアプローチ動作(G247)
- アプローチ方向がプログラム指令されておらず、G140 が有効です。つまり、TRC が有効であり(G42)、補正値が正(10)であるため、右から輪郭にアプローチします。
- 工具中心軌跡の半径がプログラム指令値 DISR=10 に等しくなるように、内部で生成されるアプローチ円(SAR 輪郭)の半径は 20 になります。
- G341 により、アプローチ動作が平面内の円でおこなわれるため、開始点は(20, -20, 0)となります。
- 点 P2 は DISCL=5 によって位置(20, -20, 5)にあり、点 P1 は Z30 によって N10 内の(20, -20, 30)にあります。

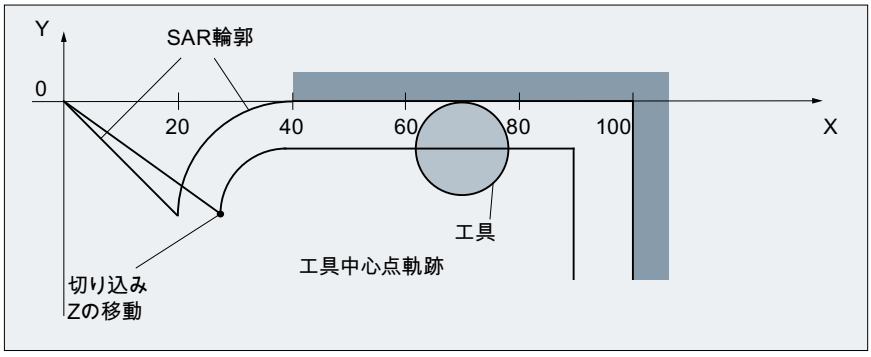


図 19-22　輪郭の例 1

パートプログラム:

プログラムコード	コメント
$TC_DP1[1,1] = 120	;工具定義 T1/D1
$TC_DP6[1,1]=10	; 半径

プログラムコード	コメント
N10 G0 X0 Y0 Z30	
N20 G247 G341 G42 NORM D1 T1 Z0 FAD=1000 F=2000 DISCL=5 DISR=10	
N30 X40	
N40 X100	
N50 Y-30	
...	

例 2

アプローチについて、次の条件が当てはまります。

- ブロック N20 で滑らかなアプローチを起動します。
- 四分円でおこなわれるアプローチ動作(G247)
- アプローチ方向がプログラム指令されておらず、G140 が有効です。つまり、TRC が有効であるため(G41)、左から輪郭にアプローチします。
- 輪郭オフセット OFFN=5(N10)
- 現在の工具半径=10 であるため、TRC の有効補正半径=15 です。したがって SAR 輪郭の半径は 25 に等しく、その結果工具中心軌跡の半径は DISR=10 に等しくなります。
- N20 では Z 位置のみがプログラム指令されているため、円弧の終点は N30 から取得されます。
- 切り込み移動
  - Z20 から Z7(DISCL=AC(7))までは早送りで
  - その後、Z0 へは FAD=200 で
  - X-Y 平面内のアプローチ円と後続ブロックは、F1500 で
    (後続のブロックでこの速度を有効にするため、N30 で有効 G コード G0 を G1 で上書きしてください。そうでない場合は、輪郭が引き続き G0 で加工されます)

後退について、次の条件が当てはまります。

- ブロック N60 で滑らかな後退を起動します。
- 四分円(G248)とヘリカル(G340)でおこなわれる後退動作
- FAD は G340 に関連しないためプログラム指令されていません。
- DISCL=6 であるため、開始点で Z=2、終点で Z=8

- DISR=5 の場合、SAR 輪郭の半径=20、工具中心点軌跡の半径=5
- 円弧ブロックの後、後退動作が Z8 から Z20 まで続き、X-Y 平面に対して平行に X70 Y0 の終点まで移動がおこなわれます。

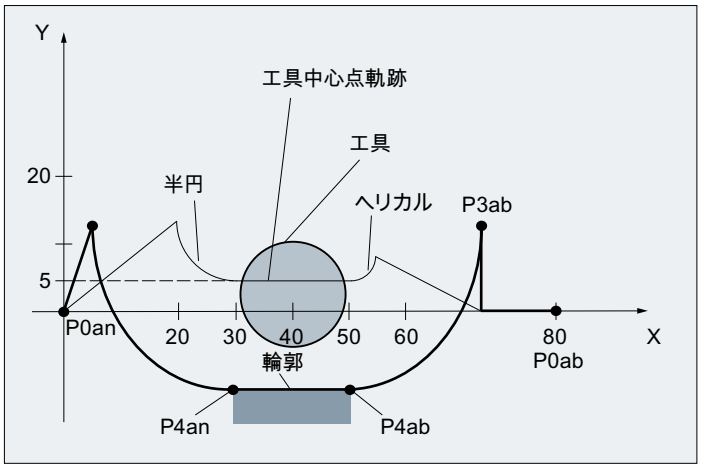


図 19-23　輪郭の例 2

パートプログラム:

プログラムコード	コメント
$TC_DP1[1,1] = 120	;工具定義 T1/D1
$TC_DP6[1,1]=10	; 半径
N10 G0 X0 Y0 Z20 G64 D1 T1 OFFN = 5	; (P0app)
N20 G41 G247 G341 Z0 DISCL = AC(7) DISR = 10 F1500 FAD=200	; (P3app)
N30 G1 X30 Y-10	; (P4app)
N40 X40 Z2	
N50 X50	; (P4ret)
N60 G248 G340 X70 Y0 Z20 DISCL = 6 DISR = 5 G40 F10000	; (P3ret)
N70 X80 Y0	; (P0ret)
N80 M 30	

注記

この方法で生成される輪郭は、工具径補正にによって変更されます。工具径補正は SAR アプローチブロックで起動され、SAR 後退ブロックで解除されます。

工具径補正では、工具径(10)と輪郭オフセット(5)の合計である 15 という有効半径が考慮されます。したがって、その結果生成される工具中心軌跡の半径は、アプローチブロックでは 10、後退ブロックでは 5 となります。

19.5.5 TRC の解除(G40)

G40 命令

TRC は G40 命令によって解除されます。

特別な注意点

- TRC は、G0(早送り)または G1(直線補間)を含むプログラムブロックのみで解除できます。
- 工具径補正が有効な場合に D0 がプログラムされていると、補正が解除されず、異常メッセージ 10750 が出力されます。
- 工具径補正の解除を含むブロックでジオメトリ軸がプログラムされている場合、その軸が選択平面上になくても、補正が解除されます。

19.5.6 外側コーナでの補正

G450/G451

G 命令 G450/G451 を使用して、外側コーナでの不連続ブロック遷移の応答を制御できます。

命令	意味
G450	遷移円による不連続ブロック遷移
G451	等間隔の軌跡の交点による不連続ブロック遷移

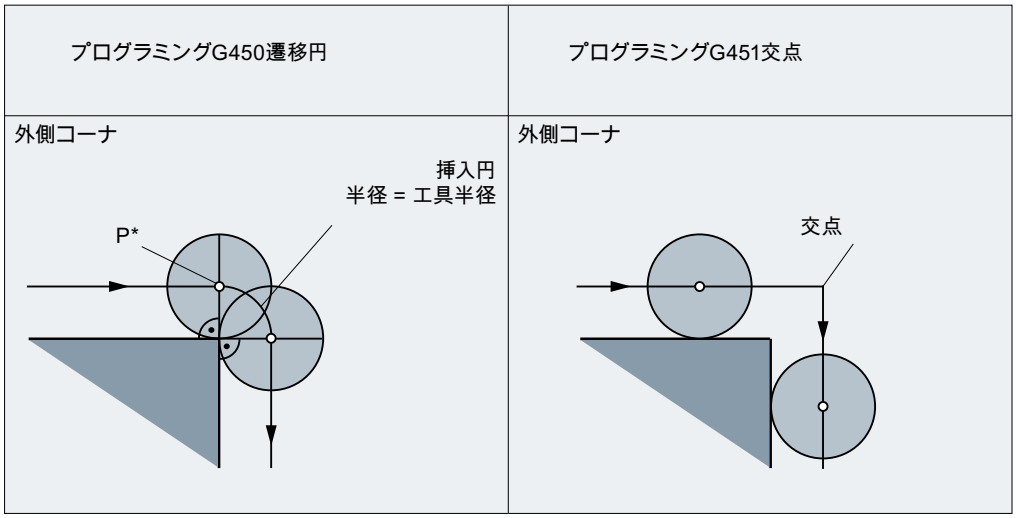


図 19-24　G450 と G451 での 90°の外側コーナの例

G450(遷移円)

外側コーナで G 命令 G450 が有効な場合、工具の中心点は、工具径に沿った円弧上を移動します。この円弧は、前の軌跡部分(プログラムブロック)の終点における通常の位置(軌跡接線に対して直角)から始まり、新しい軌跡部分(プログラムブロック)の開始点における通常の位置で終わります。

外側コーナが非常に平らな場合、G450(遷移円)と G451(交点)による応答が次第に同等になります(「非常に平らな外側コーナ」を参照してください)。

鋭い外側コーナが望ましい場合は、工具を輪郭から引き離してください(「DISC」の章を参照してください)。

DISC

G450 遷移円では、刃先が外側コーナ(プログラム指令位置)で停止するように、遷移円を通る工具中心点の軌跡が制御されるため、鋭い外側輪郭コーナは生成されません。G450 を使用して鋭い外側コーナを加工する場合は、DISC 命令を使用してオーバーシュートをプログラム指令できます。これによって遷移円が円錐曲線となり、工具刃先が外側コーナから離れます。

DISC 命令の値の範囲は、増分値 1 で 0 ～ 100 です。

値	意味
DISC = 0	オーバーシュート無効、遷移円有効
DISC = 100	交点(G451)と同様の応答を理論的に生成する大きさのオーバーシュート

DISC のプログラム可能な最大値は、次のマシンデータを介して設定できます。

MD20220 $MC_CUTCOM_MAX_DISC(DISC の最大値)

一般に、DISC では 50 より大きい値はお勧めできません。

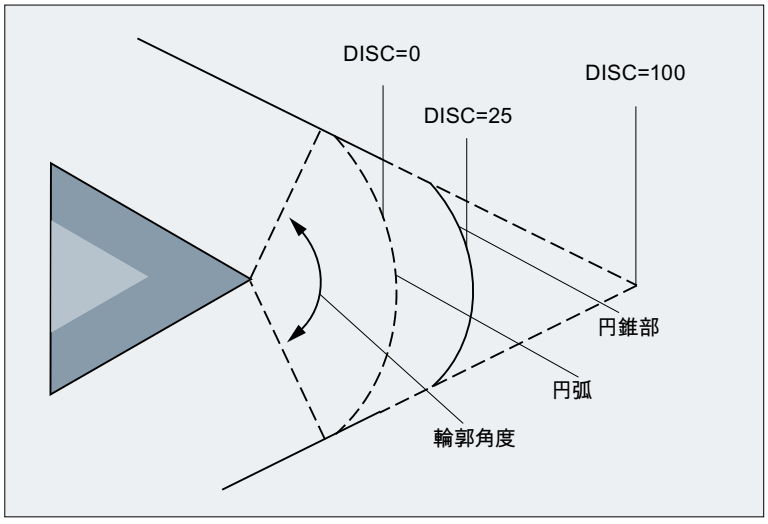


図 19-25　例:DISC = 25 によるオーバーシュート

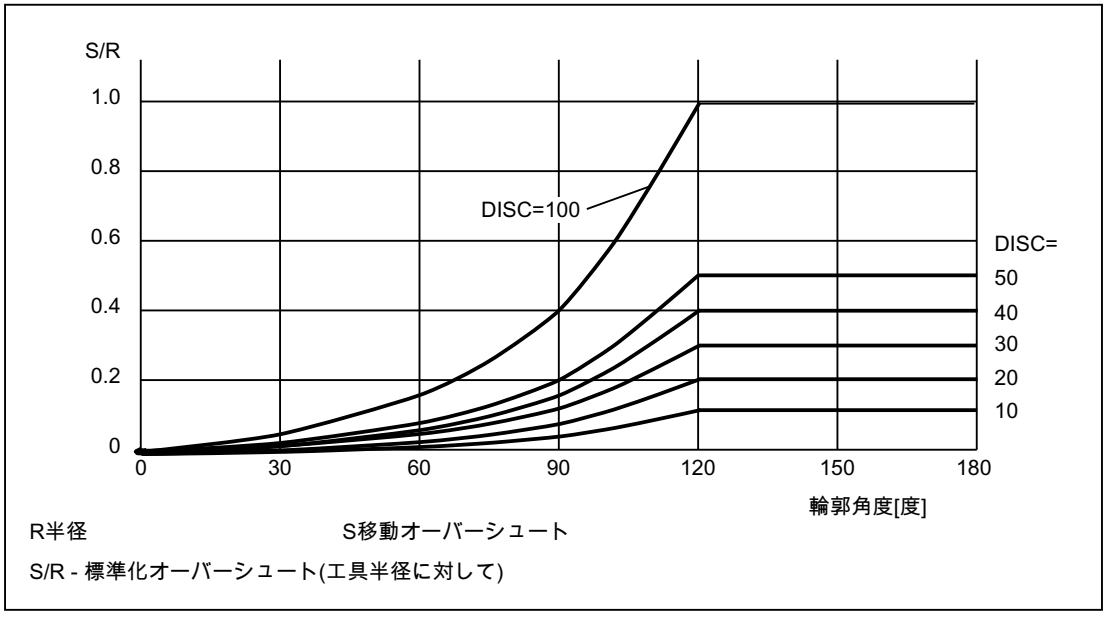


図 19-26　輪郭角度に応じた DISC によるオーバーシュート

G451(交点)

G 命令 G451 が有効な場合は、プログラム指令輪郭に対して工具径の距離(工具の中心点軌跡) にある軌跡線(直線、円弧、またはヘリカルのみ)から生成される位置(交点)にアプローチします。スプラインと多項式が拡張されることはありません。

非常に鋭い外側コーナ

外側コーナが非常に鋭い場合、G451 によって過剰に無駄な軌跡が生成されることがあります。したがって、外側コーナが非常に鋭い場合は、システムが G451(交点)から G450(遷移円、該当する場合は DISC を含む)に自動的に切り替わります。

この自動切り替え(交点→遷移円)の角度(輪郭角度)は、次のマシンデータで指定できます。

MD20210 $MC_CUTCOM_CORNER_LIMIT(工具径補正での補正ブロックの最大角度)

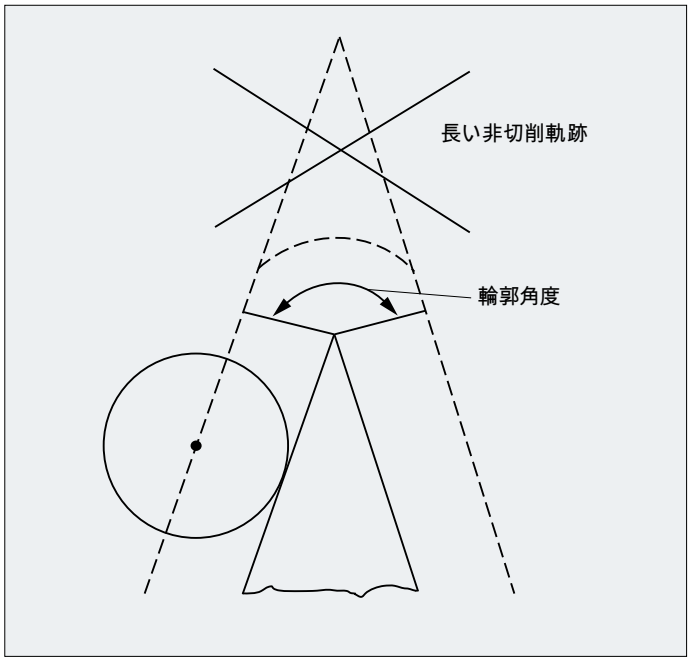


図 19-27　遷移円への自動切り替えの例

非常に平らな外側コーナ

外側コーナが非常に平らな場合、G450(遷移円)と G451(交点)による応答が次第に同等になります。この場合、遷移円の挿入はお勧めできなくなります。5 軸同時加工でこれらの外側コーナに遷移円を挿入することが許容されない 1 つの理由は、それによって輪郭モード(G64)での速度に制限が課されるからです。したがって、外側コーナが非常に平らな場合は、システムが G450(遷移円、該当する場合は DISC を含む)から G451(交点)に自動的に切り替わります。

この自動切り替え(遷移円→交点)の角度(輪郭角度)は、次のマシンデータで指定できます。

MD20230 $MC_CUTCOM_CURVE_INSERT_LIMIT(工具径補正での交点演算の最大角度)

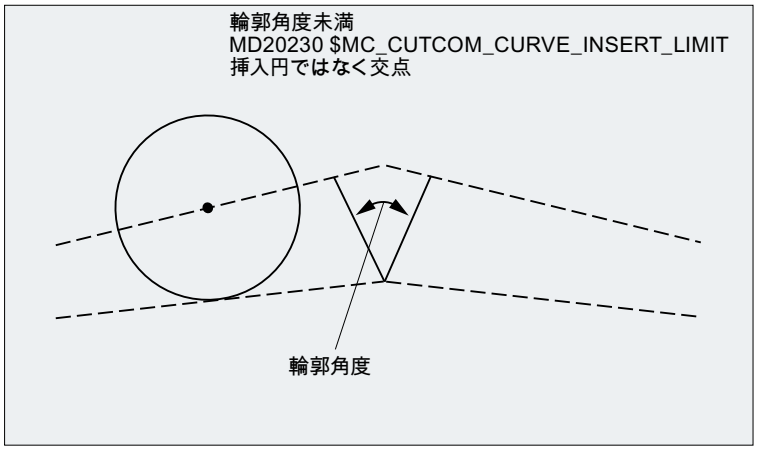
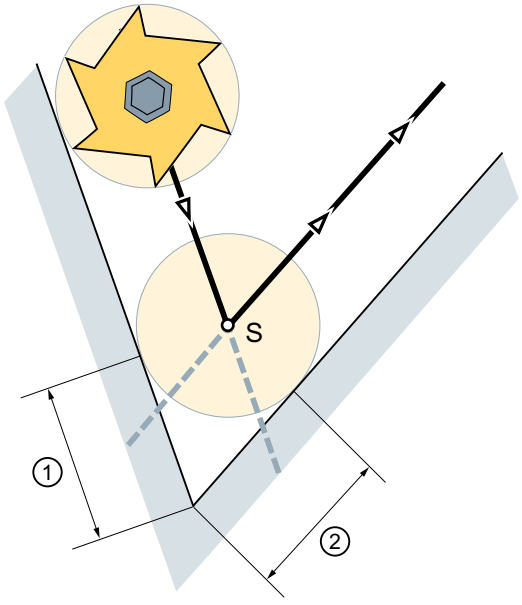


図 19-28　交点への自動切り替えの例

19.5.7　補正と内側コーナ

交点

連続した 2 つのブロックが内側コーナを形成する場合、2 つの等間隔の軌跡が交差する点の検出しようとします。交点が検出されると、プログラムされた輪郭がその交点まで短縮されます。



S:　交点

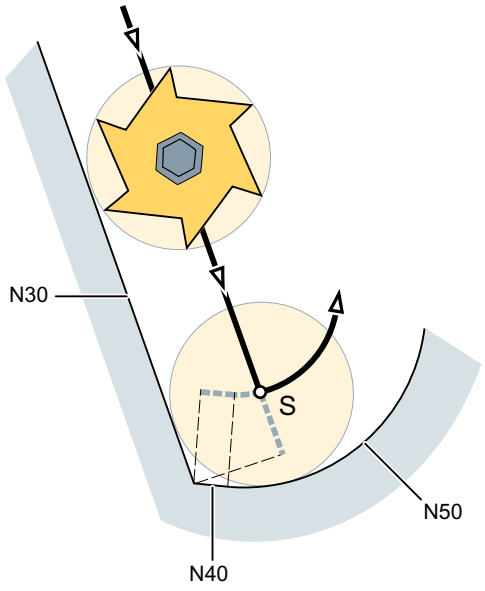
①　最初のブロックは終点で短縮

②　第 2 のブロックは先頭で短縮

図 19-29　プログラムされた輪郭が短縮される

交点なし

内側コーナの場合、2 つの連続したブロック間で交点が検出されない可能性があります。この場合、コントローラによって次のブロックが自動的にチェックされ、このブロックの等間隔の軌跡に対する交点を検出しようとします。



S:　交点

図 19-30　予測輪郭演算

次のブロックの自動チェック(予測輪郭演算)は、下記のマシンデータによって定義されたブロック数に達するまで、常に行われます。

MD20240 $MC_CUTCOM_MAXNUM_CHECK_BLOCKS (TRC のための予測輪郭演算のブロック)

先読み対象として定義されたブロック数内で交点が検出されないと、プログラムの実行が中断され、アラームが出力されます。

複数の交点

内側コーナでは、予測輪郭演算によって、連続した複数のブロックで等間隔の軌跡の交点が複数検出される可能性もあります。この場合、常に最後の交点が有効な交点として使用されます。前の交点はアプローチされていません。

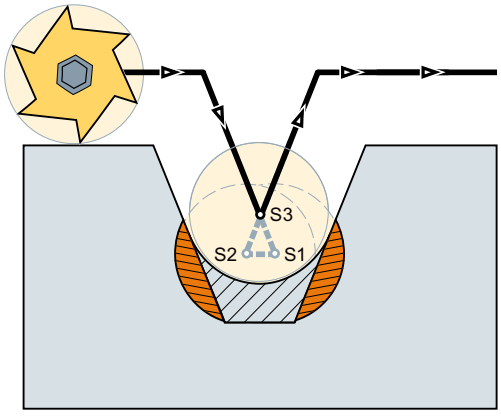


図 19-31　この例では、できるだけ経路干渉を起こさないようにポケットが加工されます。

詳しくは、「衝突監視(「ボトルネック検出」) (ページ 1841)」章も参照してください。

特記事項

次のブロックとの交点が複数検出された場合、次のブロックの開始点に最も近い交点が適用されます。

19.5.8　衝突監視(「ボトルネック検出」)

19.5.8.1　機能

工具径補正が有効な場合、衝突監視(「ボトルネック検出」)では、予測輪郭演算によって、**連続しない**ブロックの等間隔の軌跡が交差するかどうかがチェックされます。この先読み機能を使用すると、発生する可能性のある衝突が事前に検出されるため、コントローラが能動的にそれを回避できます。

先読みされるブロックの最大数は、チャネルマシンデータによって設定できます(「パラメータ設定 (ページ 1842)」を参照)。

交点が検出された場合、補正平面においてこれらのブロック間にプログラムされた動作は実行されません。アラーム 10763 が表示されます。

「補正平面でのブロックの軌跡成分がゼロになります」

NC プログラムは中断されません。ボトルネックはバイパスされます。省略されたブロックに含まれる他のすべての動作および実行可能命令(M 命令、位置決め軸の移動など)は、それらが NC プログラムでプログラムされた順序で、最後に検出された交点の位置で実行されます。

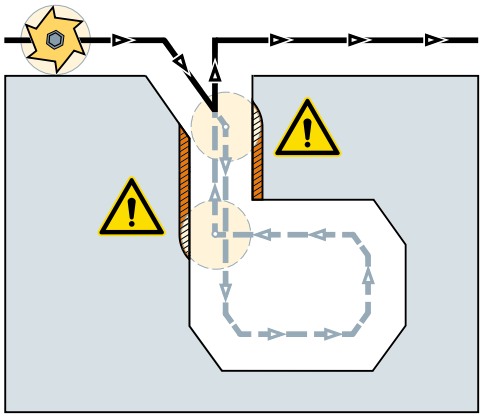


図 19-32　衝突監視が有効な間、ボトルネックをバイパス(8 つのブロックを先読み)

起動と解除

この機能の起動/解除は、NC プログラムで、G グループ 23 の命令を使用して行います。

「プログラミング (ページ 1843)」を参照してください。

19.5.8.2　パラメータ設定

予測輪郭演算のブロックの最大数

予測チェック対象のブロックの最大数は、以下で設定します。

MD20240 $MC_CUTCOM_MAXNUM_CHECK_BLOCKS (TRC のための予測輪郭演算のブロック)

アラーム 10763 の抑制

アラーム 10763 (ページ 1841)は、以下の設定によって抑制できます。

MD11410 $MN_SUPPRES_ALARM_MASK (特殊なアラームの出力をサポートするマスク)、ビット 1 = 1

19.5.8.3 プログラミング

有効な TRC による衝突検出(「ボトルネック検出」)は、G グループ 23 のコマンドによって、NC プログラムで起動または解除されます。

構文

```
G41/G42 CDON
...
CDOF/CDOF2
```

意味

CDON:	衝突検出(「ボトルネック検出」)の起動 CDON は、ブロックの設定可能な(MD20240 (ページ 1842))数から **非隣接** ブロックの工具経路が交差するかどうかを確認します。この先読み機能を使用すると、発生する可能性のある衝突が事前に検出されるため、コントローラが能動的にそれを回避できます。
CDOF:	衝突検出(「ボトルネック検出」)の解除 CDOF では、実行中のブロックの共通の交点に対する検索が(内側コーナの) **先行の** 移動ブロックで実行され、必要に応じて、さらに先行のブロックに対しても実行されます。交点が検出されると、それ以上のブロックは調べられません。外側コーナでは、交点は常に 2 つの連続したブロック間で検出されます。 **注:** CDOF を使用すると、必要な情報が NC プログラムに存在しない場合などに発生するボトルネックの検出ミスを回避できます。
CDOF2:	3 次元外周加工用の衝突検出の無効化 工具オフセット方向は、CDOF2 の隣接ブロック部分により特定されます。CDOF2 は 3 次元外周削りにのみ有効で、他のすべてのタイプの加工(3 次元正面削りなど)に対しては、CDOF と同じ意味です。

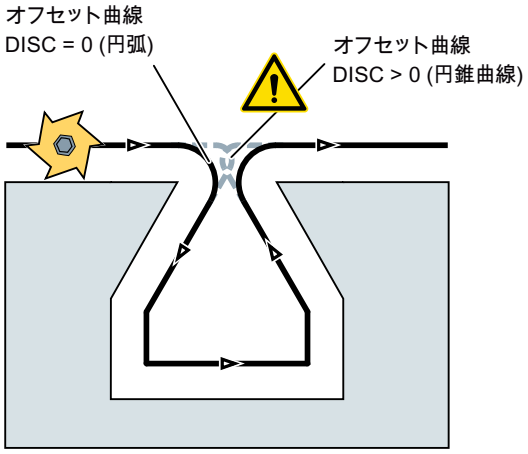
19.5.8.4 必要条件

プログラムテスト

プログラムが停止しないようにするには、プログラムテストのときは、常に使用工具の中で最も大きい半径の工具を使用します。

外側コーナでのオーバハングに起因する「ボトルネック検出」

連続しないブロックの交点がチェックされる場合、調べられるのは、プログラムされた元の輪郭ではなく、それに対応する計算された等間隔の軌跡です。そのため、外側コーナで「ボトルネック」が誤って検出される可能性があります。その理由は、DISC > 0 の場合は、計算された工具軌跡がプログラムされた元の輪郭に対して等距離でないためです。



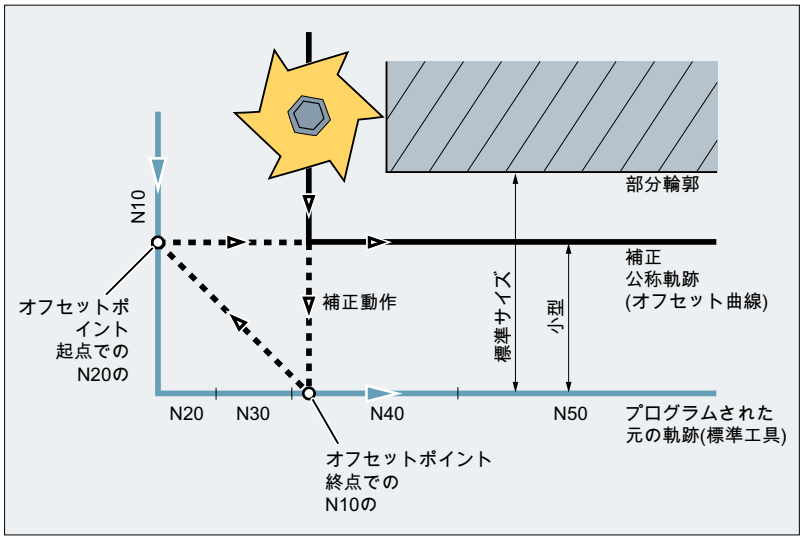
19.5.8.5 例

一例を使用した衝突検出の働き

NC プログラムで、標準工具の中心点軌跡を記述します。以下の図では、ジオメトリ関係を図示するために、輪郭が実際以上に大きく表示されているため、実際に使用する工具の輪郭は小さくなります。

例では、制御は 3 つのブロックの概要のみです。

MD20240 $MC_CUTCOM_MAXNUM_CHECK_BLOCKS = 3



交点は、N10 と N40 の 2 つのブロックのオフセット曲線間にのみ存在するため、N20 と N30 の 2 つのブロックは省略されます。例では、N40 ブロックでは、コントローラは、N10 を完全に処理する必要があるかどうかを認識していません。したがって、省略できるブロックは 1 つのみです。

CDOF2 が動作中の場合は、図に示す補正動作が実行されます。また、この動作は停止しません。この事例では、CDOF または CDON が動作中の場合は、結果的にアラームが発生します。

19.5.9 スロット形状認識(オプション) - 840D sl のみ

工具半径補正が有効で、衝突監視 (ページ 1841)がオンである場合、工具直径よりも小さな幅のプログラムされたスロットが経路干渉の可能性があるものとして検出され、加工が省略されます。ただし、特定の技術依存用途の事例(たとえば、レーザー切断機付きの製造たわみ線)では、工具(レーザービーム)の半径がプログラムされたスロット幅よりも若干大きい場合に、スロットを加工する必要があります。ユーザーにこのオプションを提示するために、工具半径補正が「スロット形状認識」ファンクションを含めるように拡張されています。

注記

「スロット形状認識」ファンクションは、ライセンスを必要とする SINUMERIK 840D sl のオプションです。

注文番号:6FC5800-0AS18-0YB0

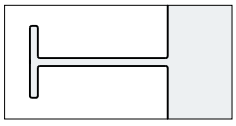
機能

「スロット形状認識」ファンクションが有効な場合、形状が予約されたスロットが自動的に識別され、加工性が確保されます。

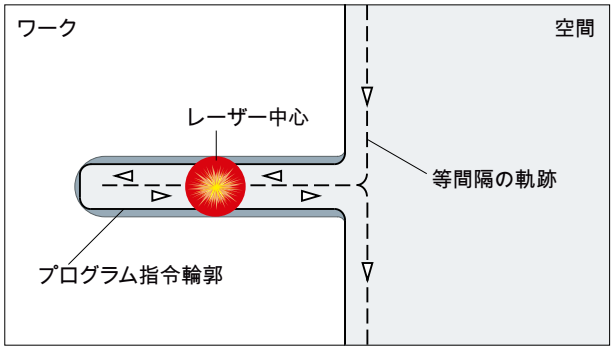
以下の項目が検出されます。

- I-slot

- T-slot



スロット幅が工具直径以上ある場合、スロットがプログラムどおりに移動します。スロット幅が工具直径より小さい場合は、プログラムされたスロットの中心に沿って工具が移動するように、等間隔の軌跡が計算されます。



下記の条件を満たしてください。

- 新しい工具(R2)の半径は、元の工具(R1)の半径と係数 2 の乗算を上回ることはできません。
  R2 < 2 * R1
- スロットは、奇数のブロックを持つ必要があります。
- セントラルブロックは直線を構成する必要があります。
- スロットの起動時の丸み付けの半径またはスロット前のブロックの長さは、R2 - R1 の相違未満とすることはできません。

起動

ファンクション「スロット形状認識」を起動するには、次のチャネル別セッティングデータを「1」に設定する必要があります。

SD42977 $SC_SLOT_FORM_RECOGN = 1

パラメータ設定

予測機能認識のブロック数

次のチャネル別マシンデータにゼロより大きい値が設定されている場合にのみ、スロット形状認識は有効です。

MD28620 $MC_MM_NUM_FEATURE_BLOCKS > 0

MD28620 により、スロットの形状を検出するために同時に考慮されるブロックの最小数を定義します。

推奨値:15 (これは、直線間の円弧を含む T スロットの輪郭要素数に対応します)。

例

次のパートプログラムセクションでは、レーザービームにより切断されるたわみ線をプログラムしています。

プログラムコード	コメント
N1 G01 G90 G71 F10000	
N2 $TC_DP1[1,1] = 120	
N3 $TC_DP3[1,1] = 0.0	
N4 $TC_DP6[1,1] = **0.2**	; 工具半径 = 0.2 mm
N6 G91 **G41** G01 T=1 D=1	; TRC を起動します。
N7 X100.00	
N8 Y-24.500	
N9 G02 X-0.200 Y-0.200 I-0.200	; たわみ線輪郭の起動時に丸み付けします。
N10 G01 X-49.700	
N11 G03 X-0.100 Y-0.100 J-0.100	
N12 G01 Y-0.100	
N13 G03 X0.100 Y-0.100 I0.100	
N14 G01 X49.700	
N15 G02 X0.200 Y-0.200 J-0.200	; たわみ線輪郭の終了時に丸み付けします。
N16 G01 Y-24.500	

スロット形状認識が無効である場合(SD42977 $SC_SLOT_FORM_RECOGN = 0)、スロット幅が 0.3 mm であっても工具半径が 0.2 mm であるために、工具半径の補正で使用工具(レーザービーム)に対して許容されない輪郭として、プログラムされたたわみ線を検出し、省略します。その一方で、スロット形状認識が有効である場合(SD42977 $SC_SLOT_FORM_RECOGN = 1)、工具半径補正は I スロットとしてプログラムされたたわみ線を検出し、レーザービームはプログラムされたスロットの中心に沿ったたわみ線を切断します。

この例では、スロットが 7 つの輪郭要素(4 つの円弧と 3 つの直線)から構成されるために、次の最小値を MD28620 $MC_MM_NUM_FEATURE_BLOCK で設定する必要があります。

MD28620 $MC_MM_NUM_FEATURE_BLOCK ≥ 7

19.5.10 可変補正値のあるブロック

必要条件

すべての補間タイプ(円弧およびスプライン補間を含む)で可変補正値を使用できます。

符号(したがって補正の方向)も変更できます。

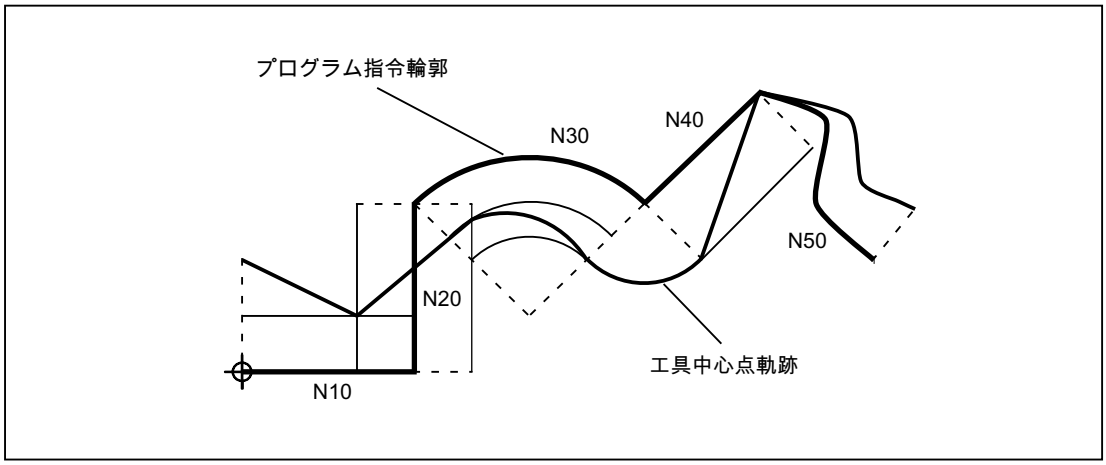


図 19-33　可変補正値による工具径補正

交点の演算

可変補正値のあるブロックでの交点が計算される場合、オフセット曲線(工具軌跡)の交点は常に、補正値が一定であるという前提に基づいて計算されます。

可変補正値のあるブロックが、移動方向において、調べられる 2 つのブロックのうち最初のブロックである場合、ブロック終了時の補正値が計算に使用されます。その他の場合は、ブロック開始時の補正値が使用されます。

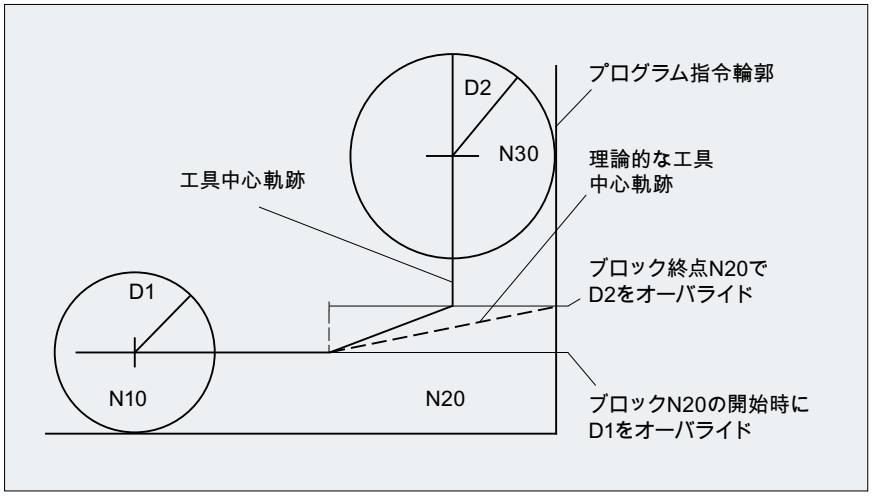


図 19-34　可変補正値による交点演算

制限事項

円弧の内側での加工時に、補正半径がプログラムされた円弧半径より大きくなると、加工は受け付けられず、次のアラームが出力されます。

アラーム 10758「可変補正値での曲率半径過小」

閉じた輪郭の確実な実施

2 つの円の半径をわずかに大きくすると、閉じた輪郭を確実に実施するために第 3 のブロックが必要になることがあります。 閉じた輪郭に存在する可能性のある 2 つの交点を表す 2 つの隣接ブロックが、補正が原因でスキップされる場合が、これに該当します。

閉じた輪郭の確実性は、第 2 の交点ではなく最初の交点を選択することで実現されます。

SD42496 $SC_CUTCOM_CLSD_CONT ≠ 0(閉じた輪郭での TRC の応答)

この場合は、ブロックがスキップされても、常に第 2 の交点に到達します。 第 3 のブロックは不要です。

19.5.11 アラーム応答

解析中のアラーム

解析中に工具径補正アラームが出力されると、メインランの加工が、到達した次のブロックの終点で停止します。つまり、通常は、現在補間されているブロックの終点で停止します(先読みが有効な場合は、軸が静止した後で)。

先読み停止アラームと有効な工具径補正

一般に、工具径補正では、ブロックの終点を特定するために、以降の移動ブロックが 1 つ以上必要です(ボトルネックの場合は 2 つ以上)。 このようなブロックの読み停止はできないため、最終ブロックのオフセット点まで移動が継続されます。 したがって、先読み停止後の最初のブロックで始点のオフセット点がアプローチされます。

取得された輪郭が、先読み停止なしで得られる輪郭から大幅にそれることがあります。特に輪郭干渉の可能性があります。 このため、次のセッティングデータが導入されました。

SD42480 $MC_STOP_CUTCOM_STOPRE (工具径補正および先読み停止に対するアラーム応答)

その値に応じて、工具径補正の応答が以前の状態と比較して変化しないままであるか、工具径補正が有効なときの先読み停止に対してアラームが出力されてプログラムが停止します。

ユーザーはこのアラームに応答し、NC スタートを使用して NC プログラムを続行するか、RESET を使用して NC プログラムを中止することができます。

19.5.12 多項式用の交点処理

機能

工具径補正が有効な 2 つの曲線が外側コーナを形成する場合、18 番目のグループの G コード(工具補正時のコーナ動作: G450/G451)に応じて、かつ関連する曲線のタイプ(直線、円弧、多項式)に関係なく、以下のようになります。

- 円錐曲線が挿入されて、コーナをバイパスするか、
  または
- 関係する曲線が延長されて、交点を形成します。

起動された G451 により交点が見つけられないか、2 本の曲線で形成される角度が鋭すぎる場合、挿入モードへの切り替えは自動で行われます。

次のマシンデータを使用して、多項式の交点処理を実行できます。

MD20256 $MC_CUTCOM_INTERS_POLY_ENABLE (多項式に対して可能な交点処理)

注記

このマシンデータを無効に設定すると、ブロック(非常に短い可能性がある)が常に実装されます(遷移がほぼ接線方向であっても)。これらの短いブロックでは常に、G64 処理中に不必要な速度低下が生じます。

19.5.13 G461/G462 アプローチ/後退方法の拡張

機能

ある特定の幾何学的条件では、工具径補正の起動/解除のために、以前の処理に比べ、拡張されたアプローチ/後退方法が必要です(下図参照)。

注記

次の例は、工具径補正解除のときの事例のみを説明します。アプローチに対する動作は事実上同じです。

例

```
G42 D1 T1                          ; 工具半径 20 mm
...
G1 X110 Y0
```

```
N10 X0
N20 Y10
N30 G40 X50 Y50
```

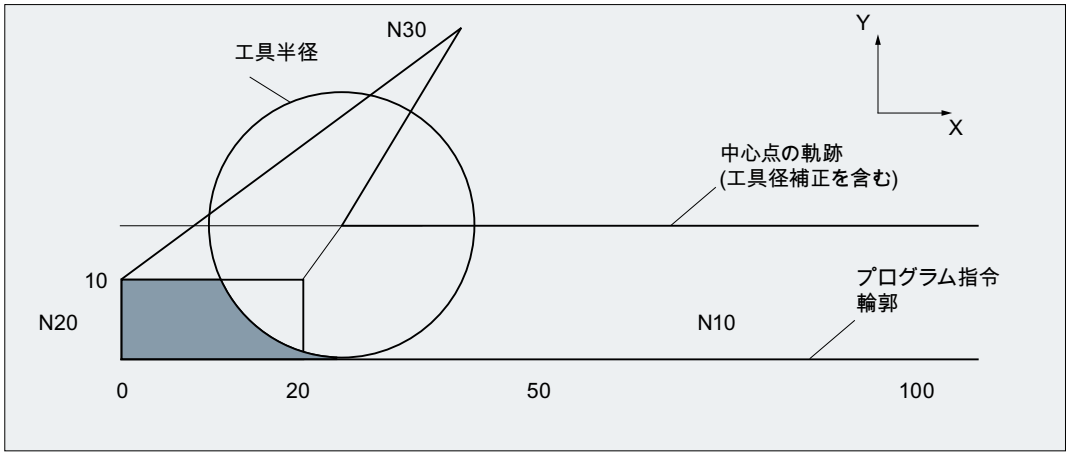


図 19-35　G460 を使用した後退動作

工具径補正が有効な最終ブロック(N20)は非常に短いため、現在の工具径について、オフセット曲線と直前のブロック(または前のブロック)の間に交点が存在しません。したがって、次のブロックと直前のブロックのオフセット曲線間(この例では、N10 と N30 の間)で交点を探します。後退ブロックに使用される曲線は実際のオフセット曲線ではなく、ブロック N20 の終点のオフセット点から N30 のプログラムされた終点までの直線です。見つかった交点にアプローチします。この図の色付き部分は、使用される工具で加工可能であっても加工されません。

G460

G460 を使用したアプローチ/後退方法は、これまでと同じです。

G461

最後の TRC ブロックと先行するブロックの間に交点ができない場合は、このブロックのオフセット曲線は、中心点が補正なしのブロックの終点にあり、半径が工具半径と同じである円弧により拡張されます。

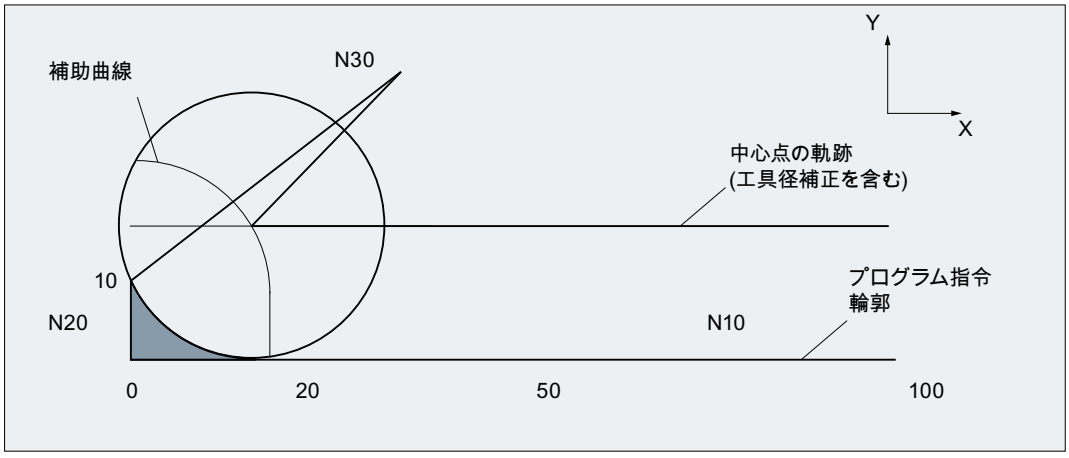


図 19-36　G461 を使用した後退動作

制御装置は、先行するブロックのいずれかを使用して、この円弧を切削しようとします。CDOF が有効である場合、交点が見つかると検出が終了します。つまり、さらに前のブロックとの交点があるかどうかはチェックされません。

CDON が有効である場合、最初の交点が見つかった後でも、さらなる交点の検出が継続されます。

このようにして見つかった交点が、先行ブロックの新しい終点であり、解除ブロックの始点です。挿入された円弧は、交点の計算専用としてのみ使用され、移動が生じることはありません。

注記

交点が見つからないと、次のアラームが出力されます。

アラーム「10751 衝突の危険」

G462

G462 (初期設定)を使用した後退では、最終工具径補正ブロックと先行ブロックの間で交点が可能でない場合、工具径補正を持つ最終ブロックの終点に直線が挿入されます(ブロックがその終点の接線によって拡張されます)。

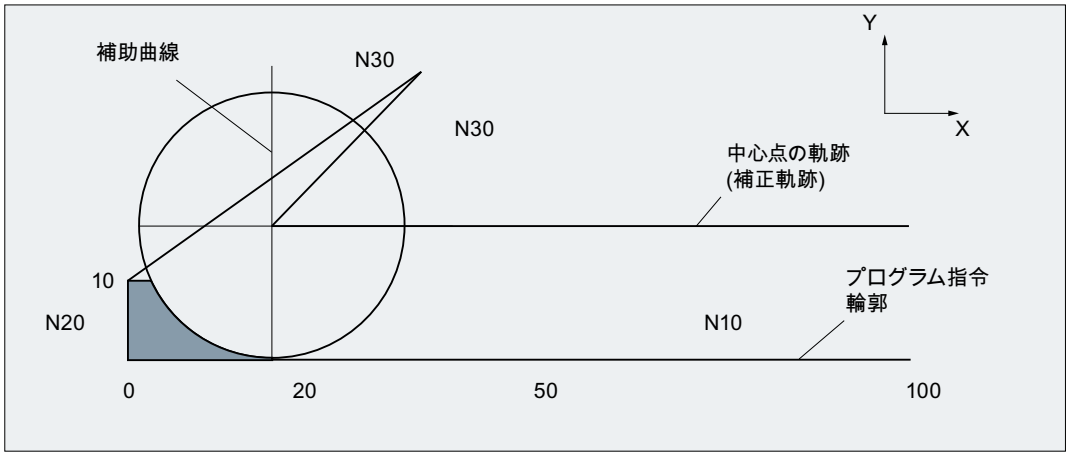


図 19-37　G462 を使用した後退動作

交点の検出は、G461 の手順と同じです。

G462 を使用した場合、プログラム例の中で N10 および N20 によって生成されるコーナは、使用される工具で事実上可能な範囲の最後までは加工されません。しかし、プログラム例の N20 の左で、パーツの輪郭(プログラムされた輪郭とは異なる)への侵入が、y 値が 10 mm より大きくても許容されない場合、この動作が必要です。

KONT が有効である(始点または終点で輪郭を回り込む)場合、終点が輪郭の手前にあるか背後にあるかによって動作が異なります。

終点が輪郭の前の場合

終点が輪郭の手前にある場合、後退動作は NORM の場合と同じです。この特徴は、G451 を使用した最終輪郭ブロックが直線または円弧によって拡張される場合でも変わりません。したがって、輪郭の終点付近の経路干渉を避けるために、さらに回り込みの移動方法を設定する必要はありません。

終点が輪郭の後ろの場合

終点が輪郭の背後にある場合、G450/G451 に応じて、常に円または直線が挿入されます。この場合、G460-G462 は何も影響しません。

この状況で、最終移動ブロックと直前のブロックとの交点がない場合、挿入された輪郭要素との交点、または回り込み円の終点からプログラムされた終点までの直線部分との交点が生じることがあります。

挿入された輪郭要素が円弧(G450)で、直前のブロックと交差する場合、これは、NORM および G461 で生成される交点と同じです。ただし通常は、円弧の残りの部分は、まだ移動する必要があります。後退ブロックの直線部分については、交点演算は不要です。

2 番目のケースでは(挿入された輪郭要素と先行ブロックの間に交点が見つからない場合)、後退直線と先行ブロックの間の交点にアプローチされます。

このため、G461 または G462 が有効である場合、G460 と異なる動作は、NORM が有効である場合、または KONT を使用した動作が幾何学的な条件により NORM と同じである場合のみ生じます。

注記

アプローチ動作は後退動作とは対称に動作します。

アプローチ/後退動作は、アプローチ/後退ブロックの G 命令の状態で特定されます。したがって、アプローチ動作は、後退動作とは無関係に設定できます。

例:

アプローチ中に G461 を使用するプログラム

```
N10 $TC_DP1[1,1]=120                    ; フライス工具のタイプ
N20 $TC_DP6[1,1]=10                     ; 半径
N30 X0 Y0 F10000 T1 D1
N40 Y20
N50 G42 X50 Y5 G461
N60 Y0 F600
N70 X30
N80 X20 Y-5
N90 X0 Y0 G40
N100 M30
```

19.6 工具径補正の継続

意味

「工具径補正の抑制」機能を使用して、多くのブロックの工具径補正をマスクします。これに対し、前のブロックの工具径補正で設定された、プログラム指令工具中心軌跡と実際に移動した工具中心軌跡との差は、補正として保持されます。

反転点での直線フライス加工時に複数の移動ブロックが必要であるものの、工具径補正(バイパス方法)で作成した輪郭は不要である場合は、この方式を使用すると、役に立つ場合があります。

起動

「工具径補正の維持」機能は G 命令 CUTCONON CUTter compensation CONstant ON)によって起動され、G 命令 CUTCONOF (CUTter compensation CONstant OFF)によって解除されます。

CUTCONON および CUTCONOF は、モーダル G グループを形成します。

初期設定は CUTCONOF です。

この機能は、工具径補正のタイプ(2.5/次元、3 次元正面フライス、3 次元外周加工)とは関係なく使用できます。

一般的な場合

工具径補正は、通常は補正マスクの前に有効で、補正マスクを再度解除するときも、まだ有効です。

CUTCONON の前の最後の移動ブロックでは、ブロック終点のオフセット点へアプローチします。以後のすべてのブロックは、補正マスクが有効であるため、補正なしで移動します。ただし、最後のオフセットブロックの終点からオフセット点までのベクトルによってオフセットされます。これらのブロックでは、任意のタイプの補間(直線、円弧、多項式)を使用できます。

補正マスクの解除ブロック、つまり、CUTCONOF を含むブロックは、通常どおりに補正されます。開始点のオフセット点で起動します。1 つの直線ブロックが、前のブロック(つまり、動作中の CUTCONON を含む、最後のプログラム指令移動ブロック)の終点とこの点の間に挿入されます。

円弧平面が補正平面に対して垂直である円弧ブロック(垂直円弧)は、CUTCONON でプログラムされたかのように扱われます。補正マスクのこの自動起動は、補正平面に移動動作を含み、このような円弧でない最初の移動ブロックで自動的にキャンセルされます。この意味で、垂直円弧が発生する可能性があるのは、外周削りのときのみです。

例:

N10	;工具 d1 の定義
N20 $TC_DP1[1,1] = 110	; タイプ
N30 $TC_DP6[1,1]=	; 半径

N40	
N50 X0 Y0 Z0 G1 G17 T1 D1 F10000	
N60	
N70 X20 G42 NORM	
N80 X30	
N90 Y20	
N100 X10 CUTCONON	;補正マスクの起動
N110 Y30 KONT	;輪郭マスクの解除時に、 必要に応じてバイパス円弧を挿入します。
N120 X-10 CUTCONOF	
N130 Y20 NORM	; TRC 解除時にはバイパス円弧がありません。
N140 X0 Y0 G40	
N150 M30	

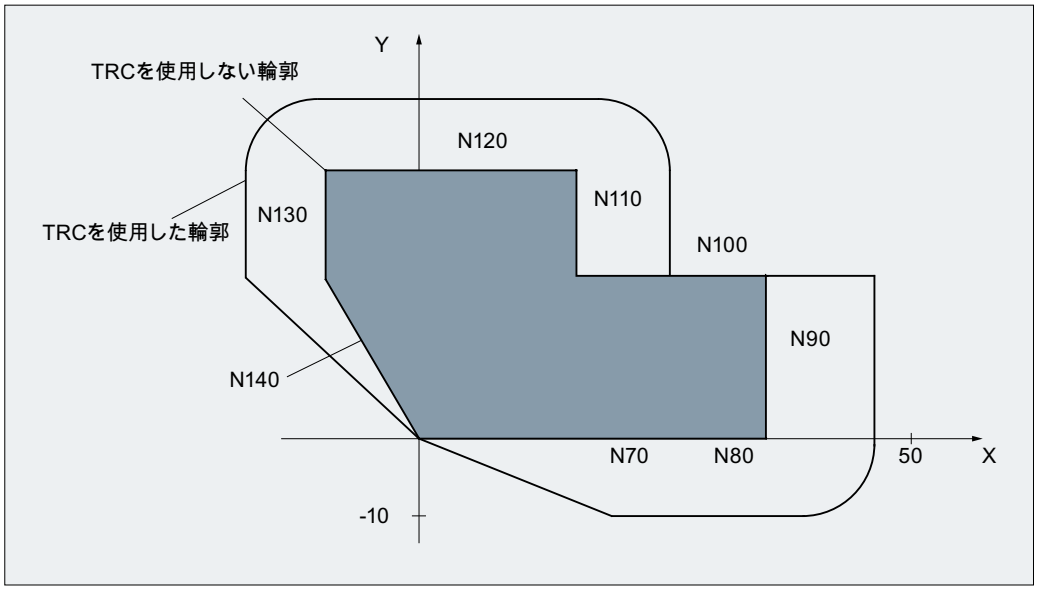


図 19-38　輪郭マスクのためのサンプルプログラム

特別な場合

- 工具径補正が有効でない(G40)場合、CUTCONON は動作しません。アラームは発生しません。ただし、G 命令は有効なままです。
  これは、G41 または G42 を使用して後のブロックで工具径補正が有効になる場合に重要になりす。
- CUTCONON が有効な状態で、7 番目の G グループの G 命令(工具径補正: G40/G41/G42)を変更することができます。G40 への変更は、直ちに有効になります。
  前ブロックの移動に使用されたオフセット分を移動します。
- CUTCONON または CUTCONOF が、有効な補正平面での移動を含まないブロックにプログラムされている場合、そのような移動動作を持つ次のブロックまで起動が遅延されます。
- CUTCONON が、アクティブな工具径補正でプログラムされ、プログラムの終了前にキャンセルされない場合、その移動ブロックは、最後の有効なオフセットによって移動します。
  同じことが、プログラムの最終移動ブロックにおける G41 または G42 の再プログラミングに当てはまります。
- 工具径補正が G41 または G42 によって起動されたとき、CUTCONON が既に有効である場合、CUTCONOF を持つ次の移動ブロックまで補正の起動が遅延されます。

- CUTCONOF により輪郭に再位置決めするとき、17 番目の G グループ(工具補正時のアプローチ/後退動作: NORM/KONT)が確認されます。つまり、KONT で必要であればバイパス円が挿入されます。バイパス円は、G41 または G42 を使用した工具径補正の起動と同じ条件で挿入されます。
- 工具径補正がマスクされるブロックの数は下記のマシンデータで制限されます。
  MD20252 $MC_CUTCOM_MAXNUM_SUPPR_BLOCKS (補正マスクの最大ブロック数)
  この数を超えると、加工が中止され、異常メッセージが発行されます。
  この制限が必要な理由は、CUTCONON の前の最終ブロックでの内部ブロック処理を再位置決め時に再開しなければならないからです。
- 工具径補正が既に有効である場合、G41 または G42 を再プログラムした後の応答は、補正マスクと似ています。
  ただし、次のような差異があります。
  - 直線ブロックのみ許容されます。
  - G41 または G42 を含む単一移動ブロックは、次のブロック内の始点のオフセット点で終了するように修正されます。したがって、中間ブロックを挿入する必要はありません。同じことが、それぞれが G41 または G42 を含む一連の移動ブロックにおける最終ブロックに当てはまります。
  - 17 番目のグループの G 命令(工具補正時のアプローチ/後退動作: NORM/KONT)とは関係なく、輪郭は常に NORM を使用して再アプローチされます。
- 連続した移動ブロックで G41 G42 が複数回プログラムされる場合、最終ブロックを除く全ブロックが CUTCONON の場合と同様に加工されます。
- 輪郭マスクのタイプは、一連の連続した移動ブロックにおける最初の移動ブロックでのみ確認されます。
  CUTCONON と同時に、G41 または G42 が、最初のブロックでプログラムされる場合、輪郭マスクの解除に対する応答は、CUTCONON によって決定します。
  この場合、G41 から G42 への変更またはその逆の変更が、再起動時に補正方向(輪郭の左または右)を変更する手段として有効です。
  輪郭マスクが有効であっても、補正方向の変更(G41/G42)を後のブロックでプログラムすることもできます。
- 衝突検出およびボトルネック検出は、輪郭抑制が有効であるブロックすべてについて無効になります。

19.7 旋回工具ホルダ

19.7.1 一般情報

はじめに

ある種の工作機械については、工具の向きを変更することができます(たとえば**段取り替えにより**)。しかし、機械の運転中、設定されている向きは**固定**で、移動中に変更することはできません。このため、キネマティックオリエンテーション変換(3 軸、4 軸または5 軸座標変換 TRAORI)は、このような機械で不要なだけでなく、意味がありません。ただし、難解な数学上の問題でユーザーが煩わされないように、向きの変更によって生じる工具長成分の変化を考慮する必要があります。これらの計算は、コントローラがおこないます。

適用

SINUMERIK 828D の場合、「旋回工具ホルダ」機能はフライス加工タイプでのみ有効です。

必要なデータ

旋回工具ホルダについてコントローラが工具補正を考慮しなければならない場合、以下のデータが必要です。

- 工具データ(ジオメトリ、摩耗など)
- 工具ホルダデータ(旋回工具ホルダのジオメトリに関するデータ)

工具ホルダ選択

コントローラで定義される工具ホルダを「旋回工具ホルダ」機能用に指定しなければなりません。指定には、次の NC プログラム命令を使用します。

```
TCARR = m
```

m:工具ホルダ番号

工具ホルダには、そのジオメトリを示す工具ホルダデータブロックが割り当てられます。

工具ホルダとそのブロックを起動すると、すぐに、つまり次の移動ブロックから有効になります。

工具/工具ホルダの割り当て

以前に有効だった工具が新しい工具ホルダに割り当てられます。

制御装置の側から見て、工具ホルダ番号 m と工具番号 T を自由に組み合わせることができます。ただし、実際の用途では、加工または構造上の理由で一部の組み合わせを除外することができます。コントローラは、組み合わせに矛盾があるかどうかをチェックしません。

工具ホルダのキネマテックの記述

旋回工具ホルダのキネマテックは、合計 33 のパラメータセットを使用して記述します。

ユーザはデータブロックのデータを編集できです。

旋回工具ホルダ

例:工具の向きを調整するための 2 つの軸を備えるカルダン工具ホルダ

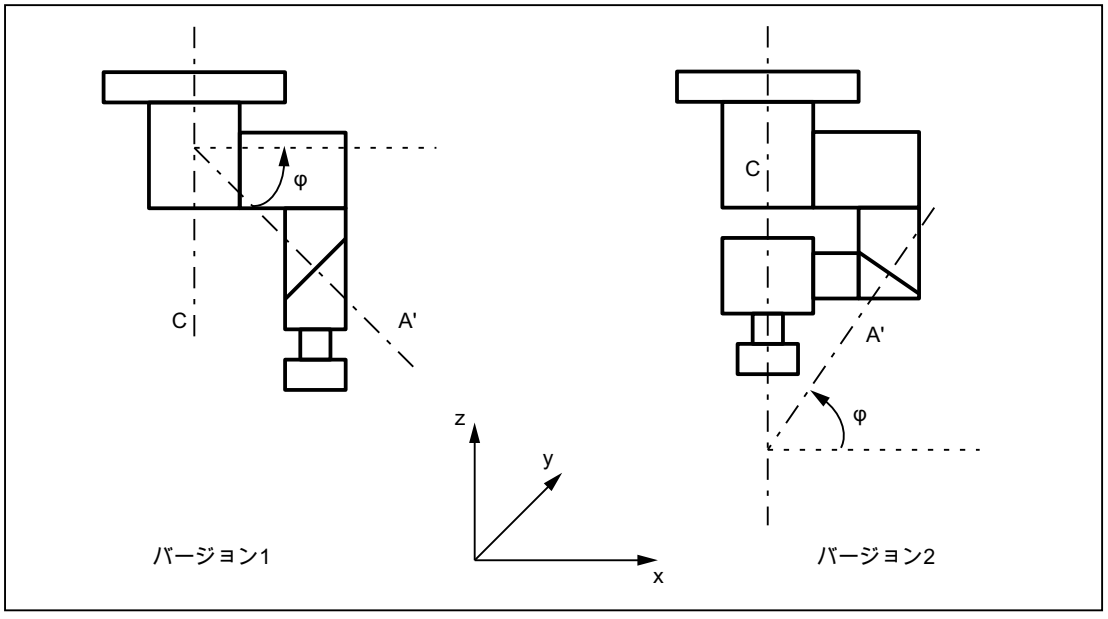


図 19-39　2 つの軸を備えるカルダン工具ホルダ

工具ホルダデータブロックの処理

2 つの方法を使用できます。

- パートプログラムから工具ホルダデータブロックへの入力の明記
- フレームからの特定の値(角度)の自動受け入れ
  このための要求事項は、工具ホルダの選択時に TCOFR (工具ホルダの旋回フレーム)も指定することです。
  工具長の計算に使用される工具の向きは、工具ホルダが変更される時点で有効なフレームから再び決められます。

Z 方向での向き

G 命令 TOFRAME を使用して、このフレームの Z 方向が現在の工具の向きと同じであるようにフレームを定義します。

工具ホルダが無効か、または向きの変化しない工具ホルダが有効である場合、新しいフレームの Z 方向は以下のようになります。

- G17 では前の Z 方向と同じ。
- G18 では前の Y 方向と同じ。
- G19 では前の X 方向と同じ

有効なフレームに対する TCOABS

次の命令を使用して、アブソリュートの工具ホルダの向きを設定します。

TCOABS (アブソリュートの工具ホルダの向き)

工具長補正のために考慮される向きは、有効なフレームの向きとは**無関係**です。

TCOABS または TCOFR のうち、一方の命令のみ有効です。

フレーム変更

工具の選択に続いてフレームを変更できます。フレームを変更しても、工具長補正成分には影響ありません。

工具ホルダデータの角度:

工具ホルダデータに格納された、プログラムされた回転角度は、フレームによって定義される回転角度の影響を受けません。TCOFR から TCOABS への変更時、工具ホルダデータのオリジナルの(プログラムされた)回転角度が再び有効になります。

工具補正タイプ

CUT2D または CUT3DFS が有効な場合は、現在の工具の向きが TRC (工具径補正)で考慮されます。

他のすべての工具補正タイプ

これらはすべて、G グループ 22 の補正タイプです(CUT3DC と CUT3DF を除く)。動作は、補正で使用される平面に関して、同じです。これは、有効なフレームからの工具の向きとは無関係に決定されます。

CUT2DF と CUT3DFF では、工具径補正に使用される補正平面は、現在の工具の向きとは**無関係に**フレームから決定されます。選択中の平面(G17/G18/G19)が考慮されます。

CUT3DC と CUT3DF

外周加工用の 3 次元工具補正

5 軸座標変換が有効である正面フライス用の 3 次元工具補正は、「旋回工具ホルダ」機能の影響を受けません。

向きの情報は、有効なキネマテック 5 軸座標変換によって決定します。

工具ホルダの向きの制限

定義された工具ホルダのキネマテックで実現できない向きがフレームによって指定された場合、アラームが出力されます。

以下のキネマテックでは、いかなる向きも実現できません。

- キネマテイックの定義に必要な 2 つの回転軸が互いに垂直でなく、工具方向を定義する工具軸が 2 番目の回転軸に対して垂直でない場合
  または
- 定義されている軸が 2 軸未満である場合

非旋回工具ホルダ

内部で使用される工具の向きは、基本的な工具の向きと選択平面(G17 - G19)にのみ依存します。

あいまいさ

2 つの軸では、フレームによって定義される特定の工具の向きは一般に、**2 つの**異なる回転角度ペアによって設定できます。コントローラは、回転角度がプログラム指令された回転角度になるべく近くなるような設定を 2 つのうちから選択します。

工具ホルダデータへの角度の保存

実際にあいまいさが生じる場合多くでは、フレームから期待される近似角度を工具ホルダデータに保存する必要があります。

パラメータセット

旋回工具ホルダのパラメータセット一式は、33 の値で構成されます。

以下のシステム変数を使用できます。

- $TC_CARR1 ～$TC_CARR33
- さらに、$TC_CARR34 ～$TC_CARR65 を仕上げオフセット用に使用できます。

個々のパラメータの内容は、次のように区別されます。

機械のキネマテイックス:

$TC_CARR1 ～$TC_CARR20 と$TC_CARR23

$TC_CARR18 ～$TC_CARR20 では、複合旋回機構を備える(工具とワークの両方を回転できる)機械を記述するのに必要な、追加ベクトル l_4 を定義します。

$TC_CARR21 と$TC_CARR22 には、必要に応じて旋回工具ホルダの向きを決めるのにその位置を使用できる、回転軸のチャネル軸名称が含まれます。

キネマテックタイプ:

文字 T、P または M を持つ$TC_CARR23

以下の 3 つのオプションをキネマテックタイプに使用できます(大文字でも小文字でもかまいません)。

T: 工具(**T**ool)のみ回転できます(基本値)。

P: ワーク(**P**art)のみ回転できます。

M: 工具とワークの両方を回転できます(**M**ixed mode)。

上記以外の文字では、旋回工具ホルダを有効にする際に、次のアラームが出力されます。

アラーム「14153 チャネル%1 ブロック%2 不明の工具ホルダタイプ: %3」

回転軸パラメータ:

$TC_CARR24 ～$TC_CARR33

$TC_CARR24 ～$TC_CARR33 のシステム変数を使用して、オフセット、角度補正、カップリングギヤシステムおよび軸制限を定義できます。

注記

これらのシステム変数は、工具管理機能が有効であってもなくても使用できます。

チェーン/データブロックの要素とプリセット

工具ホルダデータブロックの値$TC_CARR1 ～$TC_CARR20 と$TC_CARR24 ～$TC_CARR33 は NC 言語フォーマットのタイプ REAL です。

1 番目の回転軸(v_1)と 2 番目の回転軸(v_2)の軸名称の値$TC_CARR21 と$TC_CARR22 は NC 言語フォーマットのタイプ AXIS です。これらはすべてゼロにプリセットされます。

値$TC_CARR23 は、大文字「T」で初期化されます(工具のみ回転可能)。

$TC_CARRn[m]

$TC_CARR[0]= 0 には、特別な意味があります。

旋回工具ホルダ用のシステム変数

$TC_CARRn[m]

n: パラメータ 1...33

m: 旋回工具ホルダの番号、1 ～マシンデータ設定値:
MD18088 $MN_MM_NUM_TOOL_CARRIER (定義可能な工具ホルダの最大数)

説明	NC 変数	言語フォーマット	初期設定
オフセットベクトル l_1 の x 成分	$TC_CARR1	REAL	0
オフセットベクトル l_1 の y 成分	$TC_CARR2	REAL	0
オフセットベクトル l_1 の z 成分	$TC_CARR3	REAL	0
オフセットベクトル l_2 の x 成分	$TC_CARR4	REAL	0
オフセットベクトル l_2 の y 成分	$TC_CARR5	REAL	0
オフセットベクトル l_2 の z 成分	$TC_CARR6	REAL	0
回転軸 v_1 の x 成分	$TC_CARR7	REAL	0

説明	NC 変数	言語 フォーマット	初期設定
回転軸 v_1 の y 成分	$TC_CARR8	REAL	0
回転軸 v_1 の z 成分	$TC_CARR9	REAL	0
回転軸 v_2 の x 成分	$TC_CARR10	REAL	0
回転軸 v_2 の y 成分	$TC_CARR11	REAL	0
回転軸 v_2 の z 成分	$TC_CARR12	REAL	0
回転角度 α1 (°)	$TC_CARR13	REAL	0
回転角度 α_2 (°)	$TC_CARR14	REAL	0
オフセットベクトル l_3 の x 成分	$TC_CARR15	REAL	0
オフセットベクトル l_3 の y 成分	$TC_CARR16	REAL	0
オフセットベクトル l_3 の z 成分	$TC_CARR17	REAL	0
オフセットベクトル l_4 の x 成分	$TC_CARR18	REAL	0
オフセットベクトル l_4 の y 成分	$TC_CARR19	REAL	0
オフセットベクトル l_4 の z 成分	$TC_CARR20	REAL	0
回転軸 v_1 の軸名称	$TC_CARR21	AXIS	0
回転軸 v_2 の軸名称	$TC_CARR22	AXIS	0
キネマテックタイプ	$TC_CARR23	CHAR	T
回転軸 v_1 のオフセット	$TC_CARR24	REAL	0
回転軸 v_2 のオフセット	$TC_CARR25	REAL	0
回転軸 v_1 の角度オフセット(カップリングギヤ)	$TC_CARR26	REAL	0
回転軸 v_2 の角度オフセット(カップリングギヤ)	$TC_CARR27	REAL	0
回転軸 v_1 の角度移動単位(カップリングギヤ)	$TC_CARR28	REAL	0
回転軸 v_2 の角度移動単位(カップリングギヤ)	$TC_CARR29	REAL	0
回転軸 v_1 の最小位置(ソフトウェアリミット)	$TC_CARR30	REAL	0
回転軸 v_2 の最小位置(ソフトウェアリミット)	$TC_CARR31	REAL	0

説明	NC 変数	言語 フォーマット	初期設定
回転軸 v_1 の最大位置(ソフトウェアリミット)	$TC_CARR32	REAL	0
回転軸 v_2 の最大位置(ソフトウェアリミット)	$TC_CARR33	REAL	0

ユーザー用と仕上げオフセット用のシステム変数

- $TC_CARR34 ～$TC_CARR40
  ユーザーで自由に使用できるパラメータが含まれます。
- $TC_CARR41 ～$TC_CARR65
  基本的なパラメータの値に追加できる仕上げオフセットパラメータが含まれます。基本的なパラメータに割り当てられる仕上げオフセット値は、値 40 をパラメータ番号に加算することで得られます。
- $TC_CARR47 ～$TC_CARR54 と$TC_CARR61 ～$TC_CARR63
  未定義のシステム変数であり、読み取り/書き込みのアクセスをしようとするとアラームを出力します。

説明	NC 変数	言語 フォーマット	初期設定
工具ホルダ名称	$TC_CARR34	String[32]	""
軸名称 1	$TC_CARR35 *)	String[32]	""
軸名称 2	$TC_CARR36 *)	String[32]	""
識別子	$TC_CARR37 *)	INT	0
位置成分 X	$TC_CARR38 *)	REAL	0
位置成分 Y	$TC_CARR39 *)	REAL	0
位置成分 Z	$TC_CARR40 *)	REAL	0
オフセットベクトル l_1 の x 成分の仕上げオフセット	$TC_CARR41	REAL	0
オフセットベクトル l_1 の y 成分の仕上げオフセット	$TC_CARR42	REAL	0
オフセットベクトル l_1 の z 成分の仕上げオフセット	$TC_CARR43	REAL	0

説明	NC 変数	言語 フォーマット	初期設定
オフセットベクトル l_2 の x 成分の仕上げオフセット	$TC_CARR44	REAL	0
オフセットベクトル l_2 の y 成分の仕上げオフセット	$TC_CARR45	REAL	0
オフセットベクトル l_2 の z 成分の仕上げオフセット	$TC_CARR46	REAL	0
オフセットベクトル l_3 の x 成分の仕上げオフセット	$TC_CARR55	REAL	0
オフセットベクトル l_3 の y 成分の仕上げオフセット	$TC_CARR56	REAL	0
オフセットベクトル l_3 の z 成分の仕上げオフセット	$TC_CARR57	REAL	0
オフセットベクトル l_4 の x 成分の仕上げオフセット	$TC_CARR58	REAL	0
オフセットベクトル l_4 の y 成分の仕上げオフセット	$TC_CARR59	REAL	0
オフセットベクトル l_4 の z 成分の仕上げオフセット	$TC_CARR60	REAL	0
回転軸 v_1 の仕上げオフセットのオフセット	$TC_CARR64	REAL	0
回転軸 v_2 の仕上げオフセットのオフセット	$TC_CARR65	REAL	0
注:			
*)	システム変数$TC_CARR35 ～$TC_CARR40 は、ShopMill および ShopTurn だけでなく、計測サイクルでも使用されます。		

19.7.2 キネマテックの伝達と機械構成

キネマティックチェーンの表現

キネマティックチェーンという概念を使用して、レファレンス点と工具先端の間の運動の伝達を記述します。

キネマティックチェーンには、工具ホルダデータブロックに必要なデータを規程された方法ですべて記述します。特定のキネマテックに関する具体的なケースを記述するには、チェーンの関連成分に実際のベクトル、長さおよび角度を割り当ててください。チェーンは、最大配置を表します。より単純なケースでは、個々の成分がゼロであってもかまいません(たとえば、1 つの回転軸を持つ、または回転軸を持たないキネマテック)。

機械は、工具またはワークテーブル(あるいはその両方)を回転させる軸を持たなくてもかまいません。ハンドルまたは再設定によって手動で向きを設定する場合でも、この機能を使用できます。

機械構成は、以下のパラメータを使用して記述されます。

- 2 つの回転軸(v_1 と v_2)。それぞれが 1 つの回転角度(α_1 または α_2)を持ちます。回転角度は、回転ベクトルの方向から見て時計回りにプラスにカウントされます。
- 関連する機械寸法(軸距離、機械または工具レファレンス点までの距離)についての、最大 4 つのオフセットベクトル(l_1 ～ l_4)。

ゼロベクトル

ベクトル v_1 と v_2 はゼロであってもかまいません。この場合、回転軸の方向が定義されないため、関連する回転角度(プログラム指令に明記または有効なフレームから計算)もゼロでなければなりません。この条件が満たされない場合、工具ホルダが起動されるとアラームが出力されます。

2 つ未満の回転軸

回転軸を定義しないというオプションは、記述される工具ホルダが 1 つの面でしか工具を回転させない場合に有用です。したがって、意味のある最小データブロックは、工具ホルダデータ内の 0 でない唯一つの入力を含むことができます。つまり、1 つのフレームから回転角度 α1 または α2 が決められる場合、軸に平行な回転軸を記述するための、v_1 または v_2 の成分のいずれか一方の値を含むことができます。

その他の特別な事例

2 つのベクトル v_1 と v_2 は、同一直線上にあってもかまいません。しかし、向きの自由度は失われます。つまり、この種のキネマテックは、回転軸が 1 つのみ定義される場合と同じです。すべての可能な向きは、1 つの円錐面上にあります。円錐面は、工具の向き t および v_1 または v_2 が同一直線上になる場合、直線に変形します。したがって、このような特殊な場合は向きの変更は不可能です。円錐面は、工具の向き t および v_1 または v_2 が互いに垂直である場合、円の平面に変形します(つまり、すべての向きが 1 つの面で可能です)。

2 つのベクトル v_1 と v_2 がゼロであってもかまいません。この場合、向きの変更は不可能です。このような特殊な場合では、ゼロ以外の任意の長さ l_1 と l_2 は追加の工具長補正の役割を果たし、個々の軸の成分は、平面(G17 ～ G19)の変更による影響を受けません。

キネマテックデータの拡張

- 回転軸位置を介して工具ホルダ設定を定義するための、既存の機械軸への直接アクセスの手段
- 旋回ワークによるキネマテック、および旋回工具と旋回ワークによるキネマテックの拡張。
- 回転軸位置用のグリッドで不連続な値だけを許容する手段(カップリングギヤシステム)。

これらの拡張は、以前のソフトウェアバージョンと互換性があり、$TC_CARR18 ～ $TC_CARR23 のキネマテックデータブロックを含みます。

旋回工具を備える機械

旋回工具を備える機械に関しては、以前のソフトウェアバージョンから、キネマテックの定義に変更はありません。新たに導入されたベクトル l_4 に特別な意味はありません。l_4 がゼロ以外の値を含む場合、無視されます。

「旋回工具ホルダ」という用語は、テーブルが単独でまたは工具に加えて旋回できるような、新しいキネマテックタイプについては適切ではありません。しかし、互換性を確保するために使用されているのです。

旋回工具を備える機械(一般的なケース)を記述するのに使用されるキネマティックチェーンを下図に示しています。

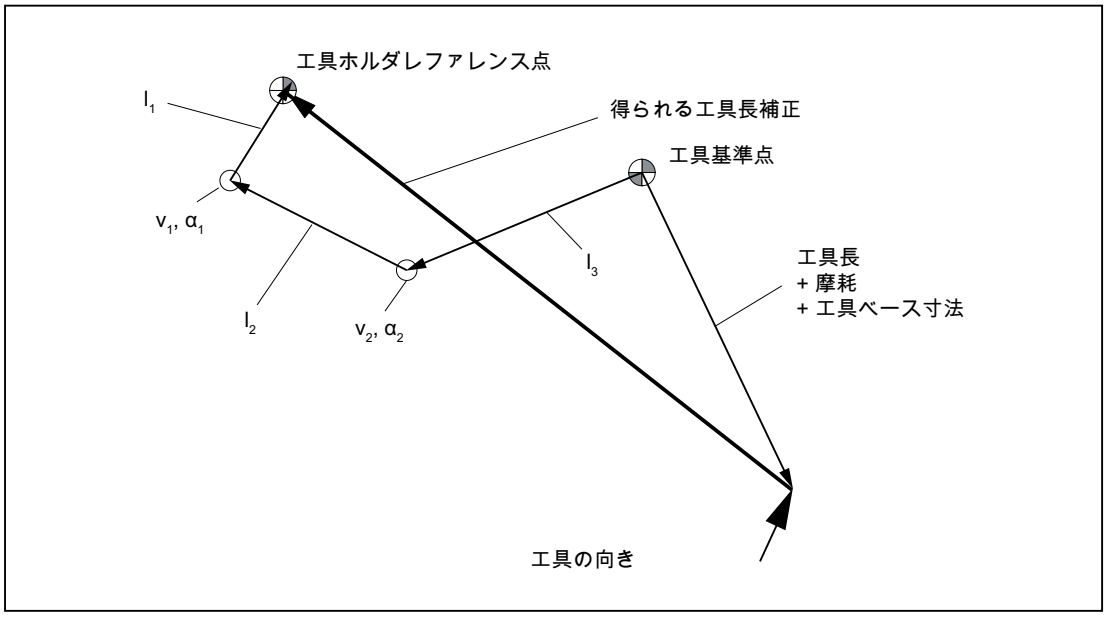


図 19-40　旋回機構付きの工具を記述するキネマティックチェーン

旋回ヘッドでのオフセットを記述するベクトルは、工具先端から工具ホルダのレファレンス点への方向において正です。

次のキネマテックタイプが、旋回工具を備える機械について定義されます。

文字 T を使用する$TC_CARR23

旋回ワークを備える機械

旋回ワークを備える機械では、ベクトル l_1 は意味を持ちません。このベクトルがゼロ以外の値を含む場合、無視されます。

旋回ワークを備える機械のキネマティックチェーンを下図に示しています。

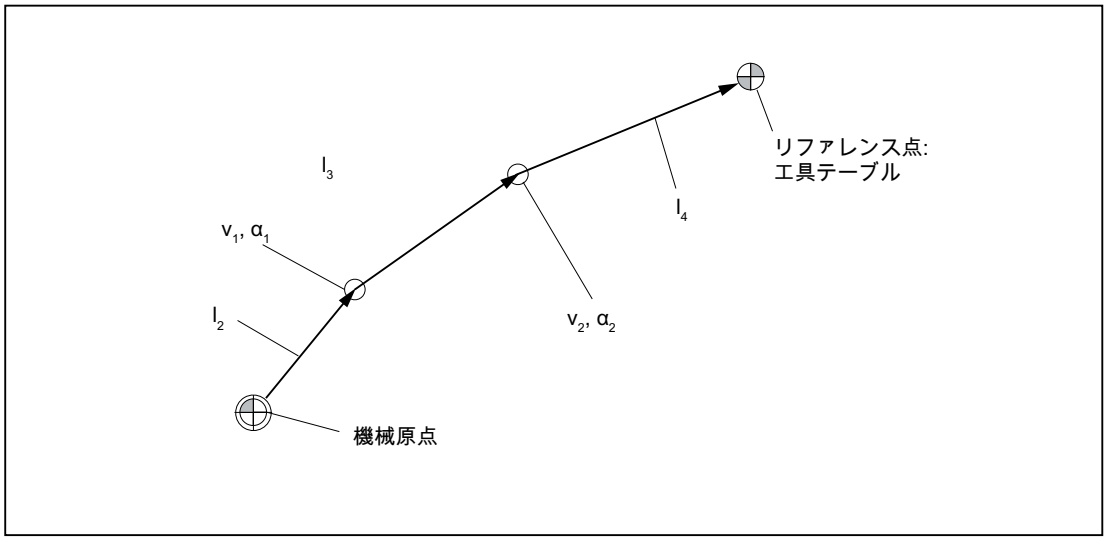


図 19-41　回転テーブルを記述するキネマティックチェーン

回転テーブル x でのオフセットを記述するベクトルは、機械レファレンス点からテーブルへの方向において正です。

次のキネマテックタイプが、旋回ワークを備える機械について定義されます。

文字 P を使用する$TC_CARR23

注記

旋回ワークを備える機械では、このタイプは通常、選択された機械レファレンス点とテーブルの基準点が同一である場合に有用です。この方法でレファレンス点を選択することには、初期状態(つまり、回転軸が回転していない状態)におけるワークの位置ゼロが、回転テーブルが作動しても変わらないという利点があります。この場合、(開いた)キネマティックチェーン(図を参照してください)が閉じます。

したがって、このような特殊なケースでは、次の公式が適用されます。$l_2 = - (l_3 + l_4)$

複合旋回機構を備える機械

複合旋回機構(工具とワークの両方が回転)を備える機械では、各成分を 1 つの軸だけによって回転させることができます。

旋回工具のキネマテックは最初の回転軸(v_1)と 2 つのベクトル l_1 および l_2 によって記述され、回転テーブルの運動は 2 番目の回転軸(v_2)と 2 つのベクトル l_3 および l_4 によって記述されます。旋回工具と旋回ワークを備える機械の 2 つのキネマティックチェーン成分を下図に示しています。

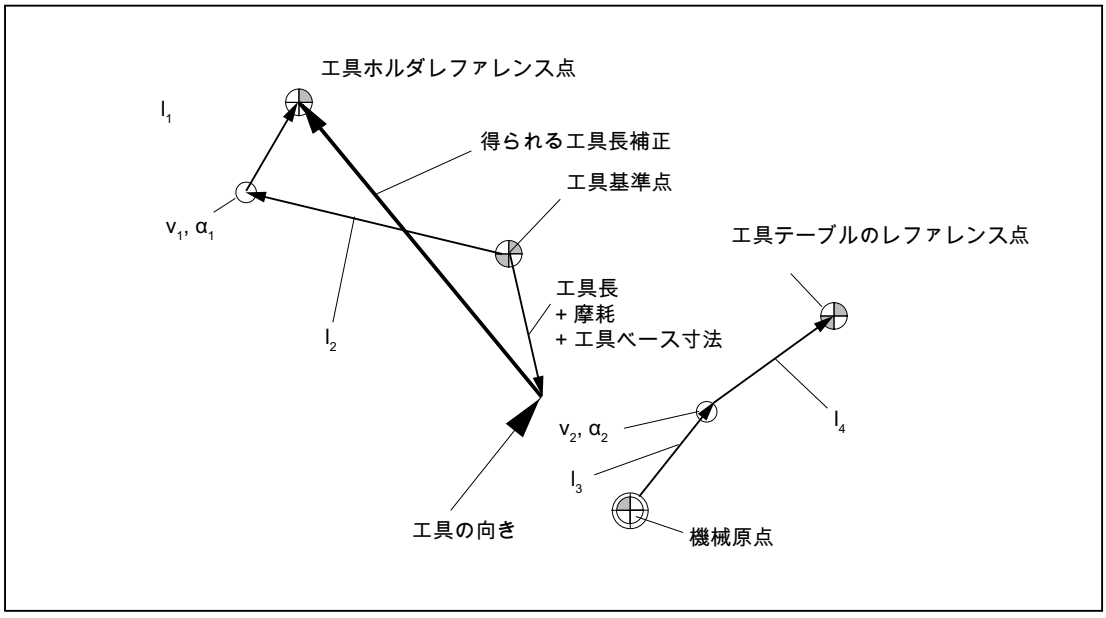


図 19-42　複合旋回機構によるキネマティックチェーン

次のキネマテックタイプが、旋回工具と旋回ワークを備える機械について定義されます。

文字 M を使用した$TC_CARR23(複合旋回機構)

注記

複合旋回機構備える機械では、このタイプは通常、テーブルのみ回転できる機械と同様、機械レファレンス点とテーブルのレファレンス点が同一である場合に有用です。この場合、テーブルを記述する(開いた)チェーン成分(図を参照してください)が閉じます。
このような特殊なケースでは、次の公式が適用されます。$l_3 = - l_4$

旋回工具タイプ T と M

旋回工具を備える機械のキネマティック(タイプ T と M) では、工具またはヘッドコンポーネント(テーブルコンポーネントとは対照的に)を記述する、旋回工具ホルダコンポーネントが、有効な工具とともに新しい全体的工具の役割を果たします。

仕上げオフセット

オフセットベクトル l_1 ～ l_4 と、回転軸 v_1 および v_2 のオフセットを、基本値と仕上げオフセットの合計として表すことができます。基本値に割り当てられる仕上げオフセットパラメータは、**基本値のインデックスに値 40 を加算すること**によって得られます。

例:

パラメータ$TC_CARR5 は、仕上げオフセット$TC_CARR45 に割り当てられます。

注記

仕上げオフセットに使用できるシステム変数$TC_CARR41 ～$TC_CARR65 の意味については、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、上級編』;「工具オフセット」

起動

次の設定によって、仕上げオフセット値が基本値に加算されます。

SD42974 $SC_TOCARR_FINE_CORRECTION = 1 (仕上げオフセット TCARR オン/オフ)

必要条件

許容可能な仕上げオフセットの合計は、制限されます。

許容可能な最大値は、次の表のようにマシンデータを使用して定義します。

値:	マシンデータ:
● ベクトル l_1 ～ l_4 の成分:	MD20188 $MC_TOCARR_FINE_LIM_LIN
● 2 つの回転軸 v_1 と v_2 のオフセット:	MD20190 $MC_TOCARR_FINE_LIM_ROT

以下の場合のみ、不正な仕上げオフセット値が検出されます。

- このような値を含む旋回工具ホルダが有効になる場合
  および
- 次のセッティングデータが設定されるのと同時に
  SD42974 $SC_TOCARR_FINE_CORRECTION

回転の記述

回転を記述するデータブロックは、1 つのベクトル v_1/v_2 で構成され、初期状態における回転軸の回転方向と角度 α1/α2 が記述されます。回転角度は、回転ベクトルの方向から見て時計回りにプラスにカウントされます。

2 つの工具ホルダ角度 α1 と α2 は、現在選択されている有効な平面(G17 ～ G19)とは無関係に、フレームを使用して決定されます。

初期状態(両方の角度 α1 と α2 がゼロ)における工具の向きは、次のとおりです(初期設定の場合と同様)。

- G17:Z に平行。
- G18:Y に平行。
- G19:Z に平行

工具ホルダへのデータの割り当て

旋回工具ホルダを備える機械の例

キネマテックタイプ T の旋回工具ホルダを備える機械について、図示したフライスヘッドで以下の設定が得られます。

オフセットベクトル l_1 の成分=	(-200, 0, 0)
オフセットベクトル l_2 の成分=	(0, 0, 0)
オフセットベクトル l_3 の成分=	(-100, 0, 0)
回転軸 v_1 の成分=	(1, 0, 0)
回転軸 v_2 の成分=	(-1, 0, 1)
工具レファレンス点の工具ベース寸法	(0, 0, 250)

注記

工具ベース寸法の工具レファレンス点は、機械におけるレファレンス点によって定義されます。

作業領域のレファレンス点についての詳細は、「K2: 軸タイプ、座標系、フレーム (ページ 845)」の章を参照してください。

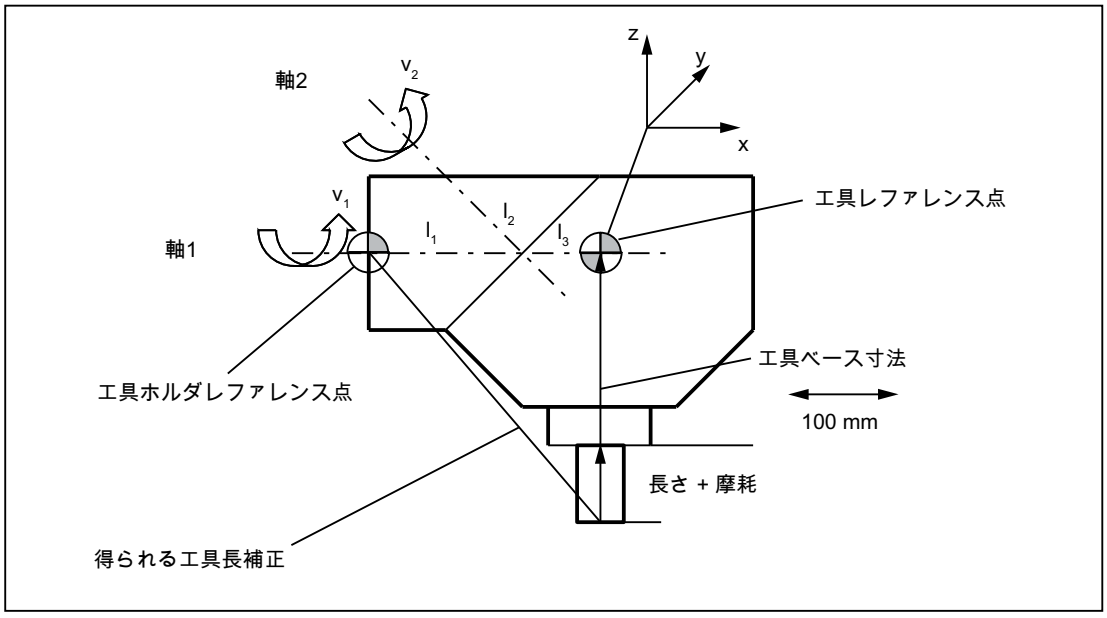


図 19-43　工具ホルダデータの割り当て

次のような適切な仮定の下で、データブロックの値の割り当てをおこなっています。

- 2 つの回転軸が、ある点で交差する。
  したがって、l_2 の成分はすべてゼロです。
- 最初の回転軸は x/z 面にあり、2 番目の回転軸は x 軸に平行です。
  これらの条件によって、v_1 と v_2 の方向が定義されます(長さは、ゼロでない限り関係ありません)。
- 工具ホルダのレファレンス点は、2 つの回転軸の交点から見て x 軸のマイナス方向 200 mm のところにある。
  この条件によって、l_1 が定義されます。

関連データブロック値の指定

旋回工具ホルダを備える機械上にある工具ホルダについて、以下の関連データブロック値が指定されます。

説明	NC 変数	値
オフセットベクトル l_1 の x 成分	$TC_CARR1	-200
オフセットベクトル l_1 の y 成分	$TC_CARR2	0
オフセットベクトル l_1 の z 成分	$TC_CARR3	0
オフセットベクトル l_2 の x 成分	$TC_CARR4	0
オフセットベクトル l_2 の y 成分	$TC_CARR5	0

説明	NC 変数	値
オフセットベクトル l_2 の z 成分	$TC_CARR6	0
回転軸 v_1 の x 成分	$TC_CARR7	1
回転軸 v_1 の y 成分	$TC_CARR8	0
回転軸 v_1 の z 成分	$TC_CARR9	0
回転軸 v_2 の x 成分	$TC_CARR10	-1
回転軸 v_2 の y 成分	$TC_CARR11	0
回転軸 v_2 の z 成分	$TC_CARR12	1
回転角度 α1 (°)	$TC_CARR13	0
回転角度 α2 (°)	$TC_CARR14	0
オフセットベクトル l_3 の x 成分	$TC_CARR15	-100
オフセットベクトル l_3 の y 成分	$TC_CARR16	0
オフセットベクトル l_3 の z 成分	$TC_CARR17	0

説明

この例で選ばれている工具ホルダのキネマテックでは、2 つの回転軸が 45°の角度を成しており、その向きは、どの値もとることができないことを意味しています。具体的には、この例では、X 成分が負である向きの記述ができません。

工具ベース寸法の x 成分:	0
工具ベース寸法の y 成分:	0
工具ベース寸法の z 成分:	250

注記

必要なデータを工具ホルダのジオメトリから明確に決めることはできません。つまり、ユーザーは、どのデータを保存するかを、ある程度自由に決めることができます。したがって、この例では、2 番目の軸までの工具ベース寸法について 1 つの z 成分のみ指定することが可能です。この場合、l_2 はゼロではなく、2 番目の軸上のこの点と最初の軸上の追加点との距離の成分を含みます。最初の軸上の点も自由に選択できます。どの点を選択するかに応じて、レファレンス点(これも自由に選択できる)に到達するように l_1 を選択しなければなりません。

一般的に:軸の回転によって変更されないベクトル成分は、回転の「前」と「後」で任意のベクトルに分散できます。

19.7.3 キネマティック結合による工具ホルダ

キネマティック結合による工具ホルダのモデリング

キネマティック結合により定義された構造の機械で、工具ホルダとして動作する機械パーツもキネマティック結合を使用して記述できます。キネマティック結合と工具ホルダで同じジオメトリデータを合わせて管理できます。

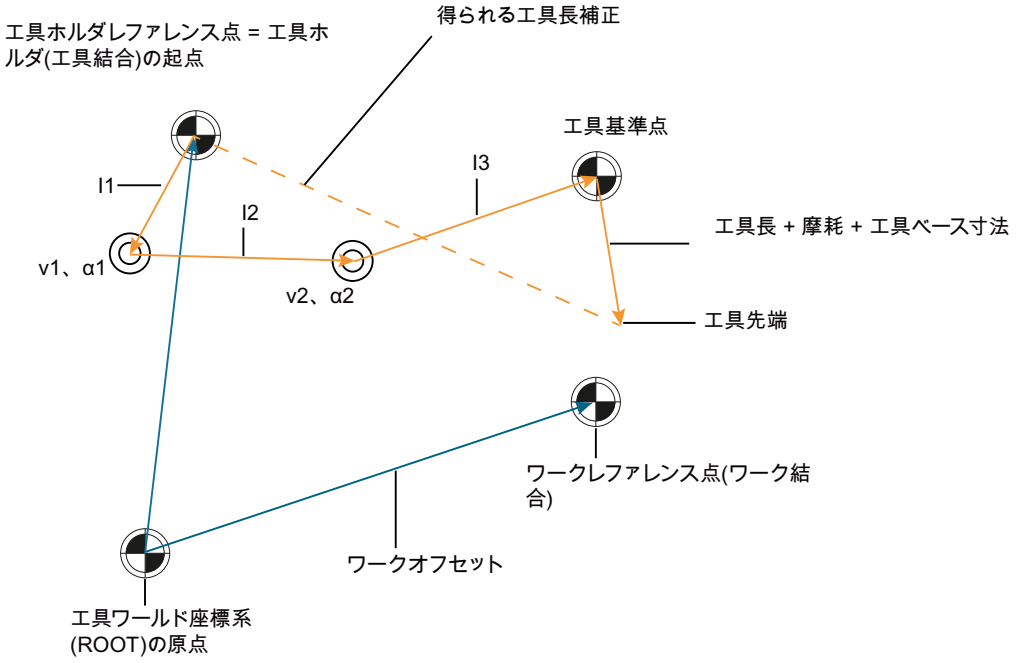
機械のキネマティックを定義するには、2 つのキネマティック結合が必要です。

- 機械原点(世界座標系の原点)からワークレファレンス点(ワーク構成部分)までの 1 つの結合点。
- 機械原点から工具ホルダレファレンス点(工具構成部分)までの他の結合点。
- キネマティック結合については、機能マニュアル「応用機能」の「K7:キネマティックチェーン」を参照してください。

$TC_CARR23 によってパラメータ設定された 3 つの異なるキネマティックタイプが工具ホルダに対応します。

以下のグラフィックは、旋回工具(タイプ T)、旋回ワーク(タイプ P)、および解決済みキネマティック(タイプ M)を含む機械のキネマティックの例を示しています。解決済みキネマティック(タイプ M)は、両タイプの組み合わせです。

旋回工具の例

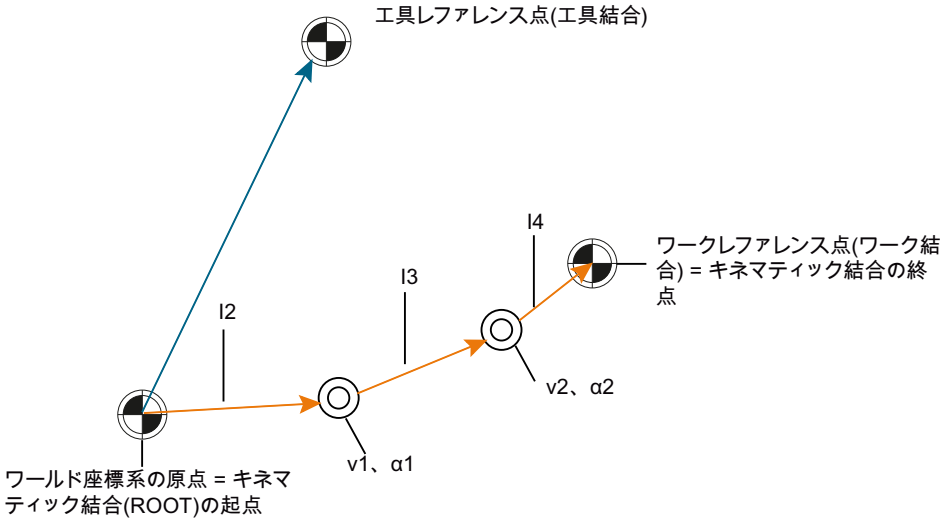


オレンジ: 工具ホルダのキネマティック結合

青: 機械のキネマティック結合

図 19-44 旋回工具を含む工具ホルダ

旋回ワークの例

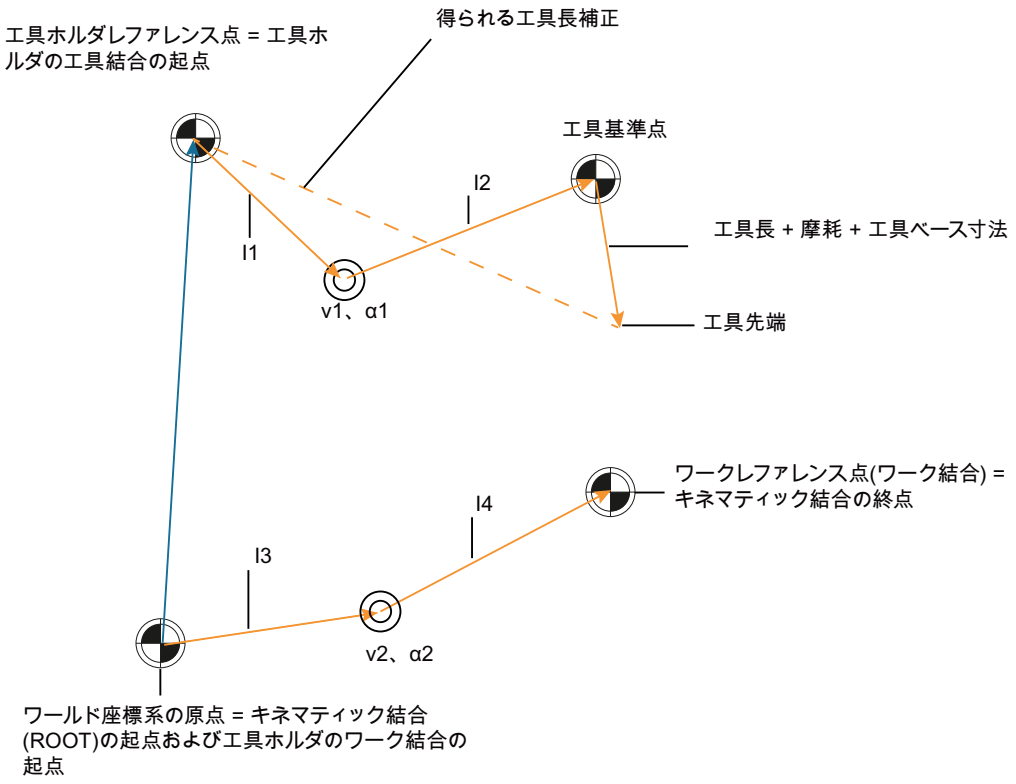


オレンジ: 工具ホルダのキネマティック結合

青: 機械のキネマティック結合

図 19-45 旋回ワークを含む工具ホルダ

混合キネマティックの例



l_1 および l_2　工具ホルダの工具結合のオフセット

v_1　工具結合の回転軸

l_3 および l_4　工具ホルダのワーク結合のオフセット

v_2　工具結合の回転軸

図 19-46　混合キネマティック

工具ホルダのキネマティックは、機械のキネマティックにより定義されます。

工具ホルダの定義:

- $TC_CARR_KIN_CNTRL による工具ホルダの特性の定義:
  - 工具ホルダのキネマティック結合、工具結合、およびテーブル結合により工具ホルダを接続します。
- 工具ホルダのキネマティック結合の起点は、システム変数 $TC_CARR_KIN_TOOL_START および$TC_CARR_KIN_PART_START (結合エレメント名)によって設定します。
- $TC_CARR21 (ページ 1860)および$TC_CARR22 による工具ホルダの回転軸の定義
- $TC_CARR23 (T、P、または M)による工具ホルダタイプの定義

工具ホルダのキネマティックの定義:

工具ホルダのキネマティック結合のワーク構成部分は、以下のように構造化できます(例):

キネマティック結合(ジオメトリック軸 X、Y、および Z)の直線軸を定義します。

X、Y、および Z 方向でタイプ"OFFSET" (=l_2)のキネマティックエレメントを定義します。$TC_CARR4 - $TC_CARR6 に対応します。

X、Y、および Z 方向でタイプ"AXIS_ROT" (=v_1)のキネマティックエレメントを定義します。$TC_CARR7 - $TC_CARR9 に対応します。

X、Y、および Z 方向でタイプ"OFFSET" (=l_3)のエレメントを定義します。$TC_CARR15 - $TC_CARR17 に対応します。

X、Y、および Z 方向でタイプ"AXIS_ROT" (=v_2)のエレメントを定義します。$TC_CARR10 - $TC_CARR12 に対応します。

X、Y、および Z 方向でタイプ"OFFSET" (=l_4)のエレメントを定義します。$TC_CARR18 - $TC_CARR20 に対応します。

必要に応じて、軸 v1 および v2 の回転軸オフセットを定義します。このオフセットは、"AXIS_ROT"タイプ結合エレメントに割り当てられたシステム変数$NK_A_OFF に入力します。$TC_CARR24 – TC_CARR25 に対応します。

少なくとも 1 つの回転軸を定義する必要があります。

注記

回転軸オフセットの符号

$TC_CARR24 または$TC_CARR25 の回転軸に対する回転軸オフセットの正の値は、負の値としてキネマティック結合に入力します。

工具ホルダの選択:

工具ホルダは、パートプログラムの NC プログラム命令により選択します:

```
TCARR = <name>
```

注記

キネマティック結合の開始ポイント

工具ホルダ(ワークまたは工具)のキネマティック結合の起点は、TC_CARR_KIN_PART_START または TC_CARR_KIN_TOOL_START によって定義されていない場合、ROOT エレメント(ワールド座標系)は結合の起点として自動的に定義されます。

$TC_CARR_KIN_CNTRL

$TC_CARR_KIN_CNTRL は、キネマティック結合エレメントまたは手動工具ホルダデータ($TC_CARRxx)から工具ホルダのジオメトリデータを読み取るかどうかを指定します。システム変数は、ビット指定です。

構文	説明	
$TC_CARR_KIN_CNTRL[n]	工具ホルダをパラメータ設定するためにキネマティック結合で定義された機械モデルのジオメトリデータの伝送を制御します。Bit0=1 の場合、$TC_CARRxx からのデータは無視されますが、データはキネマティック結合から取得されます。さらに、キネマティック結合の接続方法を設定できます。	
	データタイプ:INT	
n	工具ホルダのインデックス。工具ホルダの最大数はマシンデータ MM_NUM_TOOL_CARRIER で定義できます。	
ビット 0=1	工具ホルダは、キネマティック結合エレメントからパラメータ設定できます。仕上げオフセットを含む工具ホルダの以下のデータは、キネマティック結合エレメントからのジオメトリデータにより置換されます:	
	手動工具ホルダ	キネマティックチェーン
	$TC_CARR1 – TC_CARR3 (オフセットベクトル l_1)	オフセットベクトル l_1 の X、Y、および Z 成分
	$TC_CARR4 – TC_CARR6 (オフセットベクトル l_2)	オフセットベクトル l_2 の X、Y、および Z 成分
	$TC_CARR7 – TC_CARR9 (回転軸方向 v_1)	回転軸 v_1 の X、Y、および Z 成分
	$TC_CARR10 – TC_CARR12 (回転軸方向 v_2)	回転軸 v2 の X、Y、および Z 成分
	$TC_CARR15 – TC_CARR17 (オフセットベクトル l_3)。	オフセットベクトル l3 の X、Y、および Z 成分
	$TC_CARR18 – TC_CARR20 (オフセットベクトル l_4)	オフセットベクトル l4 の X、Y、および Z 成分
	$TC_CARR24 – TC_CARR25 (回転軸オフセット)	
ビット 0=0	工具ホルダパラメータには、$TC_CARRxx のデータが割り当てられます。	

構文	説明
ビット 1=1	**工具**結合の終点がワールド座標系の原点と一致するように、オフセットベクトルl_1の成分が変更されます(結合が接続されます)。
ビット 2=1	**ワーク**結合の終点がワールド座標系の原点と一致するように、オフセットベクトル l_4 の成分が変更されます(結合が接続されます)。

$TC_CARR_KIN_TOOL_START および$TC_CARR_KIN_PART_START

システム変数$TC_CARR_KIN_TOOL_START および$TC_CARR_KIN_PART_START は、工具ホルダの工具またはワークの構成部分をパラメータ設定するために使用するキネマティック結合の起点を定義します。システム変数が空の場合、各キネマティック結合の起点がキネマティック記述のルートエレメントです。

工作機械の全変換キネマティックの静的モデリングに対してではなく、"工具拡張"として元の機能に工具ホルダを設置する場合に、ルートエレメントで始まらないキネマティック結合の構成部分のみをパラメータ設定するオプションは特に意味を持ち、実際に必要になります。

構文	説明
$TC_CARR_KIN_TOOL_START[n] = <name> $TC_CARR_KIN_PART_START[n] = <name>	工具またはワークの結合の開始エレメントの名前を定義します。 データタイプ:STRING
n	工具ホルダのインデックス。工具ホルダの最大数はマシンデータ MM_NUM_TOOL_CARRIER で定義できます。

$TC_CARR_KIN_TOOL_END および$TC_CARR_KIN_PART_END

システム変数$TC_CARR_KIN_TOOL_END および$TC_CARR_KIN_PART_END は、工具ホルダの工具またはワークの構成部分をパラメータ設定するために使用するキネマティック結合の終点を定義します。少なくとも、この 2 つの名前のいずれかが空文字列ではないことが必要です。

構文	説明
$TC_CARR_KIN_TOOL_END[n] = <name> $TC_CARR_KIN_PART_END[n] = <name>	工具またはワークの結合の終了エレメントの名前を定義します。 データタイプ:STRING
n	工具ホルダのインデックス。工具ホルダの最大数はマシンデータ MM_NUM_TOOL_CARRIER で定義できます。

必要条件とメモ

- 工具ホルダ(TC_CARRxx)のシステムデータがキネマティック結合エレメントからのデータで置換される場合、システム変数$TC_CARRxx の成分は変更できません。
- キネマティック結合により定義された工具ホルダおよび変換は、同時に有効にできます。工具ホルダがパラメータ設定された方法(システムデータ TC_CARR…から直接設定されたか、キネマティック結合から設定されたか)は重要ではありません。
- 工具ホルダをパラメータ設定するために使用するキネマティック結合を工具ホルダ設定に割り当て可能であることが必要です。この条件が満たされない場合、アラーム 14149 がトリガします。

キネマティック結合からの工具ホルダの測定。

工具ホルダのキネマティックおよび以降のオフセットを測定する場合、有効なオフセットベクトルを変更する必要があります。このようなオフセットベクトルは基本的に、パラメータ設定を提供するキネマティック結合の任意数の部分ベクトルから形成可能であるため、補正する結合エレメントは自動的には特定できません。

構文	説明
$TC_CARR_CORR_ELEM[n, m] = <string>	工具ホルダデータが読み取られるキネマティック結合の次のエレメントの名前。このエレメントは、インデックス m のオフセットベクトルの補正エレメントとして使用されます。 データタイプ:STRING

構文	説明
n	工具ホルダのインデックス。工具ホルダの最大数はマシンデータ MM_NUM_TOOL_CARRIER で定義できます。
M	補正ベクトルのインデックス(数値の範囲 0... 3)

この結合エレメントは、システム変数$TC_CARR_CORR_ELEM[m, n]によって一意に特定できます。ここで、m は工具ホルダデータセットのインデックス、n は補正ベクトルのインデックス(n=0...3])です。

TCARR のベクトルの長さと方向の読み出し。

現在有効なオフセットベクトル、回転軸ベクトル、および回転軸オフセットは、システム変数$PC_TCARR_OFFSET、$PC_TCARR_AX_VECT、および $PC_TCARR_AX_OFFSET によって読み出すことができます。オフセットベクトルの詳細については、「一般情報 (ページ 1860)」を参照してください。

構文	説明
$PC_TCARR_OFFSET[m,n] = <value>	有効な工具ホルダのオフセット l_m の n 番目のベクトル成分の現在値。有効な工具ホルダに対して、最大 4 オフセットベクトル(l1 ～ l4)が定義されます。このシステム変数により、ベクトル成分を読み出すことができます。
	データタイプ:DOUBLE
M	有効な工具ホルダのオフセットベクトル[m] (m = 1...4)。
n	フィールドインデックス m で選択されたベクトルのベクトル成分 n (0 ≤ n ≤ 2)。

構文	説明
$PC_TCARR_AX_VECT[m,n] = <value>	有効な工具ホルダの回転軸 V_m の n 番目のベクトル成分の現在値。有効な工具ホルダに対して最大 2 つの回転軸(v1 ～ v2)が定義されます。このシステム変数により、ベクトル成分を読み出すことができます。
	データタイプ:DOUBLE
M	回転軸方向 v1、v2。ここで、m = 有効な工具ホルダ 1...2。
n	m で選択されたベクトルのベクトル成分 n (0 ≤ n ≤ 3)。

構文	説明
$PC_TCARR_AX_OFFSET[m] = <value>	回転軸 1 および 2 の回転オフセットの現在値。有効な工具ホルダに対して最大の 2 つの回転軸が定義されます。初期状態の回転軸の位置は、このシステム変数により読み出すことができます。
	データタイプ:DOUBLE
M	回転オフセット(m = 有効な工具ホルダ 1...2)。

2 つを超える回転軸を備える工具ホルダ

2 つを超える回転軸で定義された工具ホルダでは、3 番目の軸は無視されます。2 つの回転軸間の一定の角度(膝形)を記述します。この例では軸 v_2 は考慮されません。その代わりに、オフセット l_2 が形成されます。

関連する回転軸は、結合が行われる手順で、システム変数 $TC_CARR_KIN_ROTAX_NAME により定義されます。最大 2 つの回転軸を定義できます。

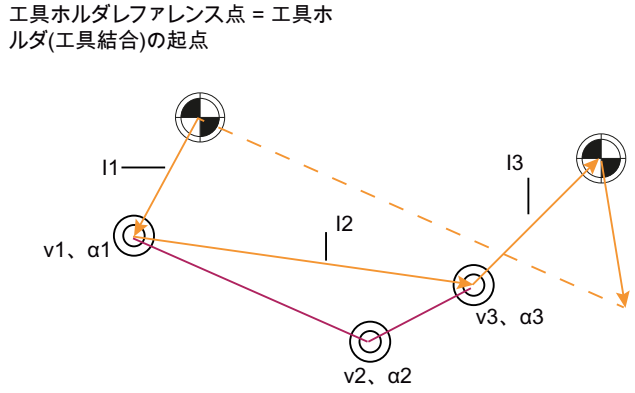


図 19-47　3 つの回転軸を備える旋回工具ホルダ。

5 軸変換に使用される回転軸は旋回軸ではありません。

回転軸の定義(2 つを超える回転軸)

工具ホルダが 2 つを超える回転軸で構成される場合、システム変数 $TC_CARR_KIN_ROTAX_NAME を使用して、関連する軸によって定義します。この件については、結合エレメントの名前は、システム変数に入力されます。キネマティック結合で定義された順序で、最大 2 つの回転軸が定義されます。このエントリは、インデックス"0"から開始する必要があります。

構文	説明
$TC_CARR_KIN_ROTAX_NAME[n, m] = <String>	回転軸として定義するキネマティック結合内のエレメントの名前。
	データタイプ:STRING
n	工具ホルダのインデックス。工具ホルダの最大数はマシンデータ MM_NUM_TOOL_CARRIER で定義できます。
M	回転軸のインデックス(数値範囲 0... 1)

例

キネマティック結合による工具ホルダのパートプログラムについては、「例:キネマティック結合による方向付け機能を備えた工具キャリア (ページ 1910)」を参照してください。

19.7.4 3 + 2 軸による傾斜面加工

機能の説明

3 + 2 軸による傾斜面加工では、旋回工具ホルダという概念の拡張が記述され、この概念が回転テーブルを備える機械に適用されます。それは、工具とテーブルの向きを同時に変更できる機械です。

「3 + 2 軸による傾斜面加工」機能は、主要な平面 X/Y (G17)、Z/X (G18)、および Y/Z (G19)を基準にした回転をともなう面加工に使用します。

工具またはワーク、あるいはその両方を回転させることで、ワークに対する工具の向きを調整できます。

ソフトウェアが、工具長、レバーアームおよび回転軸の角度から生じる、必要な補正動作を自動的に計算します。 常に前提とされるのは、必要な向きが最初に設定され、傾斜面でのポケット加工などの加工中に変更されないことです。

さらに、傾斜面加工に必要な以下の 3 つの機能について説明します。

- 有効なフレームとは無関係の工具の向きにおける**位置指令**
- 立体角を指定することによる、**フレーム回転**の定義
- 残りのフレーム成分を維持している間の、プログラム指令フレームにおける**工具方向の回転成分**の定義

5 軸座標変換との区別

要求機能が、ワークに関する向きの変更の際に(補間中でも)、TCP(工具中心点)が変化しないことを条件とする場合、5 軸ソフトウェアが必要です。

5 軸座標変換の詳細については、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』;「3 ～ 5 軸座標変換(F2)」

旋回工具ホルダの指定

旋回工具ホルダは、合計 33 個の REAL 値を使用して工具補正メモリのデータブロックによって記述される一般的な 5 軸のキネマティックシーケンスで表されます。 向きを設定するための 2 つの回転軸を持つ工具ホルダ(たとえば、フライスヘッド)では、これらの値のうち 31 個が定数です。

現在のソフトウェアバージョンでは、工具補正メモリのデータブロックが合計 47 個の REAL 値によって記述されます。 向きを設定するための 2 つの回転軸を持つ工具ホルダでは、これらの値のうち 45 個が定数です。

残りの 2 つの値は可変であり、向きを指定するのに使用します。 定数値はオフセットおよび方向と回転軸の設定オプションを記述し、変数値は回転軸の角度を記述します。

19.7.5 回転作業テーブルを備える機械

システム変数

これまで、$TC_CARR13 と$TC_CARR14 に保存された角度が、TCOABS による有効な工具長の計算に使用されていました。$TC_CARR21 と$TC_CARR22 が回転軸を参照しない場合、このことが今でも当てはまります。$TC_CARR21 または$TC_CARR22 にチャネルの回転軸への参照が含まれる場合、$TC_CARR13 または$TC_CARR14 の入力値ではなく、現在のブロックの開始時の関連軸の軸位置が角度として使用されます。

組み合わせ運転モードが可能です。つまり、一方の軸に関してシステム変数$TC_CARR13 または$TC_CARR14 の入力値から、他方の軸に関してチャネル軸の位置から角度を決めることができます。

これにより、旋回工具ホルダの設定に使用される軸が NC 内で既知であるような機械が、これらの軸の位置に直接アクセスできるようになります。その結果、特に、軸位置を読み取る際の潜在的な先読み停止がなくなります。これに対し以前は、たとえば、システム変数$AA_IM[軸]を読み取ってその結果を$TC_CARR13/14 に書き込む必要がありました。特にこれは、軸位置の読み取り時に、自動先読み停止を解除します。

MD20180

次のマシンデータ設定の場合、回転軸位置は、プログラム指令値または計算値とともに使用されます。

MD20180 $MC_TOCARR_ROT_ANGLE_INCR[i] = 0 (旋回工具ホルダの回転軸グリッド)

しかし、このマシンデータがゼロ以外の場合、使用される位置は、適切な整数値 n について次の式から得られる最も近いグリッド点です。

φ = $MC_TOCARR_ROT_ANGLE_OFFSET[i] + n * $MC_TOCARR_ROT_ANGLE_INCR[i]

この機能が必要なのは、回転軸を割り出す必要があるため、自由に定義される位置を想定できない場合です(たとえば、カップリングギヤシステム)。システム変数$P_TCANG[i]は近似値を提供し、システム変数$P_TCDIFF[i]は正確な値と近似値の差を提供します。

フレームの向き TCOFR

TCOFR (有効なフレームによって定義される向きによる角度の特定)を使用して、有効なフレーム回転から角度を特定した後のグリッドが決まります。機械のキネマティックが原因で、要求される向きを実現できない場合、アラームが出力されて加工が中止されます。このことは、目標の向きが達成可能な向きに非常に近い場合にも当てはまります。特に、このような状況でのアラームを角度の近似によって回避することはできません。

TCARR フレームオフセット

工具ホルダ変更の結果発生したフレームのオフセットは、TCARR=...を選択すると直ちに有効になります。一方、工具長の変更が直ちに有効になるのは、工具が有効になっている場合に限ります。

TCOFR/TCOABS フレーム回転

起動時にフレームの回転は実行されません。すでに有効になっている回転は変更されません。T の(工具だけを回転できる)場合のように、計算に使用される回転軸の位置は、G 命令 TCOFR /TCOABS によって異なり、動作中のフレームの回転成分、または入力値 $TC_CARRn から特定されます。

フレームの起動により、ワーク座標系内の位置が適宜変化し、機械自体による補正移動はありません。次の図に、その関係を記載します。

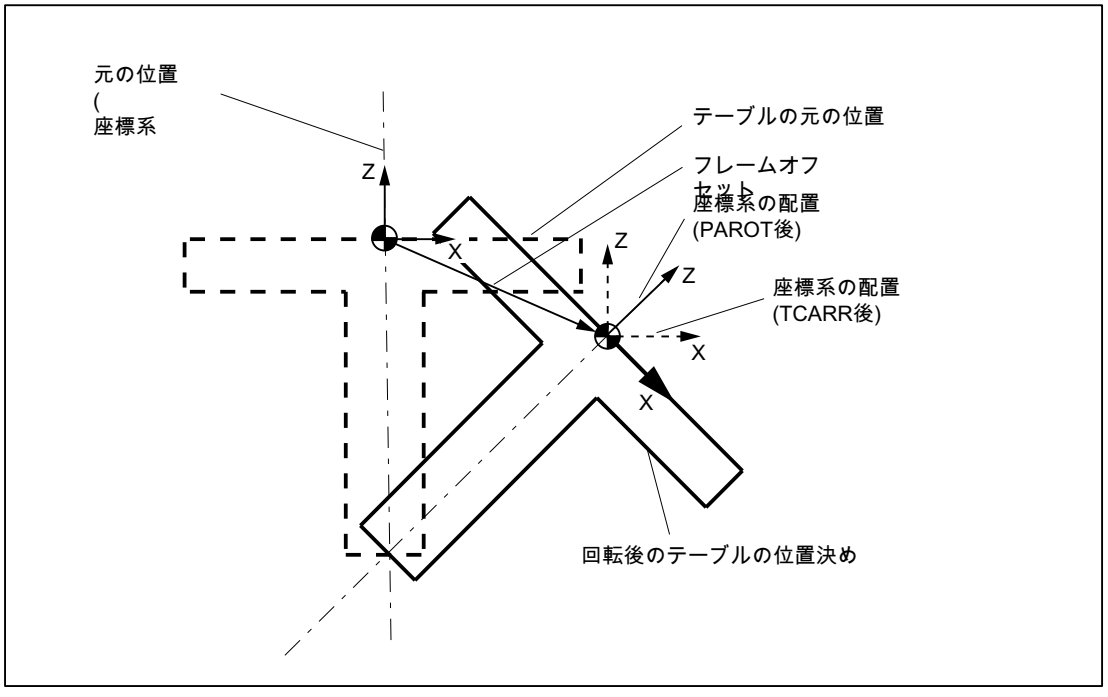


図 19-48　TCARR を使用した回転テーブルの作動時のゼロオフセット

例

図の機械で、回転テーブルの回転軸は正の Y 方向に向いています。テーブルは+45°だけ回転します。PAROT を使用すると、Y 軸を中心にした 45°の回転を同じように記述するフレームが定義されます。座標系は、実際の環境(「TCARR 後の座標系の位置」で示している)に対して回転するのではなく、定義された座標系(PAROT 後の位置)に対して-45°だけ回転します。たとえば、この座標系が ROT Y-45 によって定義され、工具ホルダが有効な TCOFR によって選択されると、工具ホルダの回転軸として+45°の角度が特定されます。

回転テーブル

回転テーブル(キネマテックタイプ P と M)の場合、同様に、TCARR を使用した起動により、直ちに座標系が回転されるわけではありません(図を参照してください)。つまり、座標系の原点が機械に対してオフセットされ、同時にワークの原点に対しては固定されたままであっても、空間における向きは変わりません。

キネマテックタイプ P と M の起動

キネマテックタイプ P と M では、工具ホルダを選択すると、追加フレーム(旋回工具ホルダのテーブルオフセット)が有効になり、原点オフセットがテーブルの回転の結果として考慮されます。

ゼロオフセットを専用のシステムフレーム$P_PARTFR に書き込むことができます。書き込むには、次のマシンデータのビット 2 を設定しなければなりません。

MD28082 $MC_MM_SYSTEM_FRAME_MASK (システムフレーム(SRAM))

この場合、次のマシンデータによって識別される基本フレームがゼロオフセットに必要でなくなります。

MD20184 $MC_TOCARR_BASE_FRAME_NUMBER (テーブルオフセットを取り込むための基本フレーム数

キネマテックタイプ M の起動

タイプ M のキネマティック(工具とテーブルは各々 1 つの軸を中心に回転します)を使用すると、TCARR による工具ホルダの起動より、同時に、対応する変更が有効な工具長(工具が有効であれば)とゼロオフセット内に生成されます。

回転

機械加工処理によっては、旋回工具ホルダまたはテーブルの使用時のゼロオフセット(フレームと工具長のいずれとしてでも)だけでなく、回転も考慮する必要があります。ただし、旋回工具ホルダを起動しても、直ちに座標系が回転するわけではありません。

TOROT

工具のみ回転できる場合、Z 軸が工具の方向を指すフレームを TOFRAME または TOROT を使用して定義できます。

PAROT

座標系をワークに関して固定する必要がある場合、つまり、元の位置に対するオフセットするだけでなく、テーブルの回転に応じて回転させることも必要な場合、PAROT を使用すれば、旋回工具を使用した状況と類似の方法で、こうした回転を起動できます。

PAROT を使用すると、有効なフレームでの平行移動、スケーリングとミラーリングが保持されますが、回転成分は、テーブルに対応する旋回工具ホルダの回転成分だけで回転します。

PAROT と TOROT では、テーブルまたは工具の向きが 2 つの回転軸によって調整される場合に向きの全体的な変化が考慮されます。複雑なキネマティックの場合は、回転軸に起因する対応成分だけが考慮されます。このため、たとえば、TOROT の使用時に、空間に固定された XY 面と平行に斜面が配置されるようにワークを回転させることができます。これにより、たとえばドリルで空ける穴がこの面に垂直にならない場所を加工する際に、この工具の回転を考慮してください。

旋回工具ホルダが有効になっていない場合、言語命令 PAROT は拒否されません。このため、プログラム指令フレームが変更されることはありません。

注記

チャネル別システムフレームに関した TCARR 機能と TOROT 機能、ならびに PAROT 機能についての詳細は、「K2: 軸タイプ、座標系、フレーム (ページ 845)」の章を参照してください。

19.7.6 旋回工具ホルダの使用手順

工具ホルダの作成

使用可能な工具ホルダデータセット数の設定

制御システムで使用可能な工具ホルダデータセット数は、マシンデータにより定義されます:

MD18088 $MN_MM_NUM_TOOL_CARRIER (定義可能な工具ホルダの最大数)

この数字は、以下の規則に従って設定する必要があります:

MD18088 $MN_NUM_TOOL_CARRIER =	
	<1 チャネルに必要な工具ホルダの最大数**> ***
	<パラメータ設定されたチャネル数**>**

この場合、マシンデータでパラメータ設定された工具ホルダの数は、パラメータ設定されたチャネルと TO ユニット間で均等に分散されます。

<チャネルまたは TO ユニットごとに使用可能な工具ホルダ数**> =**	
	MD18088 $MN_NUM_TOOL_CARRIER /
	<パラメータ設定されたチャネル数**>**

注記

マシンデータ MD28085 によるマシンデータ$MC_MM_LINK_TOA_UNIT を使ったチャネルへの TO ユニットの定義および割り付けの詳細については、以下を参照してください:

参照先:
『機能マニュアル 上級機能』、「メモリ構成(S7)」

工具ホルダデータのゼロ設定:

命令$TC_CARR1[0] = 0 を使用して、すべてのデータセットのすべての値をゼロにすることができます。

NC 命令 DELTC または PI サービス _N_DELTCAR を使用して、個々の工具ホルダデータセットを個別に削除できます。

工具ホルダのデータへのアクセス:

- パートプログラム
  - Write: `$TC_CARR<n>[<m>] = <value>`
    値**<value>**は、工具ホルダ**<m>**のパラメータ**<n>**に書き込まれます。
  - Read: `<value> = $TC_CARR<n>[<m>]`
    工具ホルダ**<m>**のパラメータ**<n>**が読み取られます。参照された工具ホルダが定義されていない場合は、アラームが表示されます。
- OPI インタフェース
  旋回工具ホルダのパラメータは、システム変数**$P_TCANG[<n>]**により読み取ることができます。

データバックアップ

上記のシステム変数は、一般的な NC データバックアップの一部として保存されます。

工具ホルダの選択

番号**<m>**の工具ホルダは、`TCARR = <m>` NC プログラム命令(`TCARR`**T**ool **C**arr**i**er)により選択されます。

`TCARR = 0` では、有効な工具ホルダが解除されます。

新しい工具または新しい工具ホルダ

新しい工具が起動されると、あたかも有効な工具ホルダに取り付けられているかのように処理されます。

プログラムされている場合、新しい工具ホルダが直ちに起動します。工具の交換または有効な工具の再プログラムは必要ありません。工具ホルダ(番号)と工具(番号)は互いに独立しており、任意の組み合わせで使用することができます。

グループ 42 の G 命令からの工具ホルダ

アブソリュートの工具の向き `TCOABS` (**T**ool **C**arrier **O**rientation **ABS**olute: アブソリュートの工具ホルダの向き):

工具の向きは、対応する値がシステム変数**$TC_CARR13** または**$TC_CARR14** に入力され、G 命令 `TCOABS` が G グループ 42 で起動されている場合に、明確に決めることができます。

フレームの工具の向き `TCOFR` (**T**ool **C**arrier **O**rientation **FR**ame: 工具ホルダ旋回フレーム):

工具ホルダの選択時に次のいずれかの G 命令が G グループ 42 で有効である場合は、工具の選択時に有効なフレームの現在の向きから工具の向きを自動的に決めることもできます。

- TCOFR または TCOFRZ
  工具が Z 方向を指すように、旋回工具ホルダが設定されます。
- TCOFRX
  工具が X 方向を指すように、旋回工具ホルダが設定されます。
- TCOFRY
  工具が Y 方向を指すように、旋回工具ホルダが設定されます。

TCOFR の効果として、傾斜面での加工中、あたかも工具が傾斜面に垂直になっているかのように工具補正が自動的に考慮されます。

注記

工具の向きは、フレームの向きに厳密に拘束されるわけではありません。フレームが有効であり、G 命令 TCOABS が有効である場合、工具を選択できますが、工具の向きは、有効なフレームの向きとは関係ありません。

工具選択に続いて、フレームを変更できます。フレームを変更しても、工具長補正の成分は影響を受けません。この場合、工具が加工平面に対して垂直であるかどうかは不明です。したがって、望ましい工具の向きが傾斜面で維持されていることを最初に確認してください。

TCOFR などが有効である場合、工具長計算で使用される工具の向きは常に、工具ホルダが変更されるたびに有効なフレームから決定されます。

グループ 53 の G 命令からの工具ホルダ

グループ 53 の G 命令(TOFRAME、TOROT など)を使用して、フレーム内の軸方向(Z、Y または X)が現在の工具の向きに等しくなるようにフレームを定義することができます。

TOFRAME の呼び出し時に有効である、グループ 6 の G 命令(G17 ～ G19)を使用して、工具の向きを決定することができます。

工具ホルダが有効でない場合、または工具ホルダが有効であるにもかかわらず工具の向きが変更されない場合、新しいフレームの Z 方向は次のようになります。

- G17 では前の Z 方向と同じ。
- G18 では前の Y 方向と同じ。
- G19 では前の X 方向と同じ

回転する工具ホルダでは、適宜これらの方向が変わります。同じことが、新しい X 方向と Y 方向に当てはまります。

TOFRAME または TOROT の代わりに、TOFRAMEX、TOFRAMEY、TOROTX、または TOROTY のいずれかの G 命令を使用できます。各軸の意味が適宜入れ替わります。

グループ変更

グループ 42 の G 命令(TCOABS、TCOFR など)を変更すると、工具長成分が再計算されます。

工具ホルダデータに保存されている(プログラム指令された)回転角度は影響を受けません。その結果、工具ホルダデータに元から保存されている角度が、TCOFR から TCOABS への変更の際に再び有効になります。

回転角度(α_1 または α_2)の読み取り:

システム変数$P_TCANG[n] (n = 1 または n = 2)を使用して、向きの計算に現在使用されている角度を読み取ることができます。

特定の向きについて 2 つの解(つまり、2 番目の有効な角度ペア)が考えられる場合、$P_TCANG[3]または$P_TCANG[4]を使用してそれらの値にアクセスできます。
$P_TCSOL を使用して、有効な解の数(0 ～ 2)を読み取ることができます。

CUT2D または CUT3DFS による工具径補正:

G グループ 22 (工具補正タイプ)の CUT2D または CUT3DFS のいずれかが有効である場合は、現在の工具の向きが工具径補正に含まれます。

回転しない工具ホルダの場合、G 命令グループ 6 (G17 ～ G19)の有効平面によってのみ決まるため、動作は以前と同じです。

他のすべての工具補正タイプ:

他のすべての工具補正タイプの動作は以前と同じです。

特に CUT2DF と CUT3DFF では、工具径補正に使用される補正平面は、現在の工具の向きとは関係なく、有効なフレームから決定されます。有効な平面(G17 ～ G19)が考慮されるため、動作は以前と同じです。

グループ 22 の残りの 2 つの G 命令 CUT3DC と CUT3DF は、工具ホルダ機能の影響を受けません。なぜならば、このような場合の工具の向き調整情報は、有効なキネマティック座標変換によって使用可能になるからです。

2 つの回転軸

2 つの回転軸では、一般的に 2 つの解が存在します。フレームから決定される向きの角度が指定角度になるべく近くなるように、コントローラ自体がこれら 2 つの解のペアを選択します。

次の 2 つの方法で角度を指定できます。

1. $TC_CARR21 または$TC_CARR22 に回転軸への参照が含まれる場合、工具ホルダが起動されるブロックの先頭でのこの軸の位置を使用して、角度を指定します。
2. $TC_CARR21 または$TC_CARR22 に回転軸への参照が含まれない場合、$TC_CARR13 または$TC_CARR14 の値が使用されます。

例

コントローラがまず、一方の軸について 10°という角度を計算します。指定された角度は 750°です。次に、720°(= 2 * 360°)を初期角度に加算して、730°という最終角度が得られます。

回転軸オフセット

システム変数$TC_CARR24 と$TC_CARR25 を使用して、回転軸オフセットを指定できます。これらのパラメータのいずれかの値がゼロ以外である場合、関連する回転軸は初期状態は、そのパラメータによって指定される位置になります(したがって、位置ゼロではありません)。この場合、角度指定はすべて、この値によって移動される座標系を参照します。

加工平面(G17 ～ G19)が変更されると、有効な工具の工具長成分のみ入れ替わります。工具ホルダの成分は入れ替わりません。次に、得られた工具長ベクトルが現在の工具ホルダに従って回転され、必要に応じて、工具ホルダに属するオフセットによって変更されます。

2 つの工具ホルダ角度 α_1 と α_2 は、現在選択されている有効な平面(G17 ～ G19)とは無関係に、フレームを使用して決定されます。

制限値

旋回工具ホルダの記述に使用されるシステム変数セット($TC_CARR30 ～$TC_CARR33)で、各回転軸について制限角度(ソフトウェアリミット)を指定できます。これらの制限値は、最小値と最大値が両方ともゼロの場合は使用されません。

2 つの制限値の少なくとも 1 つがゼロでない場合、以前に計算された解が許容範囲内であるかどうかがチェックされます。許容範囲内でない場合、無効な軸に 360°の倍数を加算するか、無効な軸から 360°の倍数を減算することで、有効な設定を得るための最初の

試みがなされます。これが成功せず、2 つの異なる解が存在する場合、最初の解が破棄され、2 番目の解が使用されます。2 番目の解は、軸制限に関して最初の解と同様に処理されます。

最初の解が破棄され、代わりに 2 番目の解が使用される場合、$P_TCANG[1/2]と$P_TCANG[3/4]の内容が入れ替わるため、実際に使用される解も$P_TCANG[1/2]に保存されます。

軸角度が計算されるのではなく指定される場合でも、軸制限が監視されます。これが当てはまるのは、旋回工具ホルダが起動されるときに TCOABS が有効である場合です。

19.7.7 プログラミング

工具ホルダの選択

工具ホルダは、次のプログラムを使用して選択されます。ここで、**m** は工具ホルダの番号です。

```
TCARR = m
```

工具ホルダデータブロックへのアクセス

パートプログラムから次のアクセスが可能です。

次のプログラムにより、工具ホルダ **m** に対するパラメータ **n** の現在の値が新しい「値」で**書き換え**られます。

```
$TC_CARRn[m] = 値
```

次の設定により、工具ホルダ m のパラメータを**読み取る**ことができます。ただし、工具ホルダデータセットが既に定義されている場合に限ります。

```
値 = $TC_CARRn[m]
```
(値は REAL 変数でなければなりません)

工具ホルダデータセット番号は、次のマシンデータで定義される範囲内になければなりません。

MD18088 $MN_MM_NUM_TOOL_CARRIER (定義可能な工具ホルダデータセットの総数)

この工具ホルダデータセット数は、有効なチャネル数で割って、一つのチャネルに定義できます。

例外:

この基準から外れる設定は、次のマシンデータを使用して選択します。

MD28085 $MC_MM_LINK_TOA_UNIT (チャネルへの TO ユニットの割り当て)

全工具ホルダデータブロックのキャンセル

次の命令を使用して、パートプログラム内で、すべての工具ホルダデータセットのすべての値を削除できます。

$TC_CARR1[0] = 0

ユーザーが設定しない場合、値は 0 にプリセットされます。

適用

工具ホルダは、**工具ホルダ**と**工具**が両方とも起動されている場合に有効になります。工具ホルダのみ選択しても、機能しません。工具ホルダ選択による動作は、G 命令 TCOABS/TCOFR (工具ホルダのモーダル G コードグループ)に左右されます。

工具ホルダが有効なときに TCOABS/TCOFR グループの G 命令を変更すると、工具長コンポーネントが再計算されます。TCOABS では、α1 および α2 の両方の回転角度について工具ホルダデータに保存された値を使用して、工具の向きが決定します。

TCOFR では、2 つの角度が現在のフレームから決定します。ただし、工具ホルダデータに保存された値は変更されません。これらは、回転角度を 1 つのフレームから計算する際に生じる曖昧さを解決するためにも使用されます。ここで、プログラムされた角度からの差が最小の角度が、考えられる多くの角度の中から選択されます。

注記

工具ホルダキネマティクスを含む工具補正のプログラミングとシステム変数の詳細については、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 上級編』

19.7.8 向きのための必要条件とシステム応答

全ての向き

特定のキネマティックを表す所定のデータセットに対して、考えられる空間のすべての向きは、以下の条件が満たされる場合のみ、表現できます。

- 回転軸を表す 2 つのベクトル v_1 と v_2 も定義されていること(すなわち、両方のベクトルがゼロ以外であること)。
- 2 つのベクトル v_1 と v_2 が、互いに垂直であること。
- 工具の向きは、2 番目の回転軸に対して垂直であること。

定義されない向き

これらの条件が満たされず、有効なフレームにより実現できない向きが TCOFR により要求された場合、アラームが出力されます。

ベクトル/回転角度の依存

回転軸の方向を表すベクトル v_1 または v_2 をゼロに設定した場合、関連する回転角度 α1 または α2 もゼロに設定してください。 そうしないと、アラームが発生します。 工具ホルダが起動されるまで、すなわち工具ホルダが変更されるまでは、アラームは出力されません。

向き調整と組み合わせた工具の精密補正

工具の精密補正と工具ホルダは、組み合わせることが**できません**。 工具ホルダが有効なときの工具の精密補正の起動、逆に工具の精密補正が有効なときの工具ホルダの起動では、アラームが発生します。

自動工具ホルダ選択、RESET

RESET またはプログラムの開始時には、以下のマシンデータを介して工具ホルダを自動的に選択できます。

MD20126 $MC_TOOL_CARRIER_RESET_VALUE (RESET 時に有効な工具ホルダ)

これは、以下のマシンデータを介した工具の制御された選択と同様に処理されます。

MD20120 $MC_TOOL_RESET_VALUE (工具長補正起動(RESET/パートプログラム終了))

RESET またはプログラム起動時の動作は、工具選択の場合と同様に、以下のマシンデータの同じビット 6 を介して、制御されます。

MD20110 $MC_RESET_MODE_MASK (RESET/パートプログラム終了後のコントローラ初期設定の定義)

または

MD20112 $MC_START_MODE_MASK (NC-START 時の初期コントロールシステム設定の定義)

詳しくは、「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 (ページ 569)」の章を参照してください。

SW 6.3 以上

リセット前の最後の選択で TCOABS が有効だった場合、動作は前のバージョンと比較して変わりません。 他の G コードが有効であれば、旋回工具ホルダは、直前のリセットの前に有効だったフレームにより起動されます。 修正された工具ホルダデータ ($TC_CARR...)も考慮されます。 これらのデータが変更されない場合、工具ホルダはリセット前とまったく同じ状態で起動されます。 リセット前の工具ホルダ選択の後に工具ホルダデータが変更された場合、最後のフレームに対応する選択は必ずしも可能ではありません。 この場合、この時点で有効な G コード(グループ 42)値と有効なフレームに応じて、旋回工具ホルダが選択されます。

MD22530 PLC への補助機能出力

工具ホルダが選択されるときに、そのコード番号が工具ホルダ番号から得られる、特定のコードまたは M コードを出力することは、自由に、 以下のマシンデータにより設定できます。

MD22530 $MC_TOCARR_CHANGE_M_CODE (工具ホルダ変更時の M コード)

詳しくは、「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 (ページ 569)」の章を参照してください。

工具ホルダキネマティク

工具ホルダキネマティクについて、以下の必要条件を満たしてください。

- 初期状態、つまり α_1 と α_2 の両方の角度がゼロのときの工具の向きも、標準設定と同じで、下記のようになります。
  - G17 では Z に平行
  - G18 では Y に平行
  - G19 では Z に平行
- 軸制限に関して許容される位置には、移動できるようにしてください。
- 自由に向きを設定できるようにするために、2 つの回転軸は互いに垂直にしてください。
  両方の軸によりテーブルが回転する機械の場合、工具の向きも最初の回転軸に垂直にしてください。
  組み合わせキネマテックを備えた機械の場合、工具の向きは、工具を回転する軸、すなわち最初の回転軸に垂直にしてください。

以下は、**フレームに**指定される向きに適用されます。

- フレームに指定された向きは、定義された工具ホルダキネマテックにより実現できるようにしてください。さもないと、アラームが出力されます。
  この状況は、キネマテックを定義するために必要な 2 つの回転軸が互いに垂直でない場合に発生する可能性があります。
  これは、2 つ未満の回転軸が定義されていて、なおかつ以下のケースの場合に適用されます。
  - **旋回工具を持つキネマテックタイプ T** で、工具方向を定義する工具軸が **2 番目の**軸に垂直でない場合。
  - **回転ワークを持つキネマテックタイプ M と P** で、工具方向を定義する工具軸が**最初の**軸に垂直でない場合です。
- 特定の位置に到達するために工具の向きが定義されたフレームが必要な回転軸は、1 つの回転軸の場合にのみ明白に決定されます。 2 つの回転軸では、一般的に 2 つの解が存在します。
- あいまいさが発生する可能性があるすべての場合において、フレームから予測される近似角度が工具データに格納されることと、回転軸が予測される位置の周辺にあることは、特に重要です。

ASUB、REPOS による応答

工具ホルダは、非同期サブプログラム(ASUB)で変更できます。 中断されたプログラムが REPOS により再開される場合、新しい工具ホルダのアプローチ動作が考慮され、プログラムはこの動作とともに継続します。 この処理は、ASUB の工具交換に似ています。

詳しくは、「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 (ページ 569)」の章を参照してください。

19.7.9 例

19.7.9.1 例: 旋回工具ホルダ

必要条件

次の例では、Y 軸を中心とする回転によって完全に記述される工具ホルダが使用されています。したがって、回転軸を定義するために 1 つの値を入力するだけで十分です(ブロック N20)。

ブロック N50 ～ N70 では、半径 5 mm、長さ 20 mm のエンドミルが記述されます。

ブロック N90 では、Y 軸を中心とする 37 度の回転が定義されます。

ブロック N120 では工具径補正が起動され、Y 軸を中心とする 37 度の回転による後続ブロックでの補正を記述するためのすべての設定が行われます。

```
N10                                          ; 工具ホルダ 1 の定義
N20 $TC_CARR8[1] = 1                         ; Y 方向における最初の回転軸のコンポーネント
N30
N40                                          ; 工具オフセットメモリ T1/D1 の定義
N50 $TC_DP1[1,1] = 120                       ; エンドミル
N60 $TC_DP3[1,1] = 20                        ; 長さ 1
N70 $TC_DP6[1,1] = 5                         ; 半径
N80
N90 ROT Y37                                  ; Y 軸を中心とする 37 度の回転
N100
N110 X0 Y0 Z0 F10000
N120 G42 CUT2DF TCOFR TCARR = 1 T1 D1 X10
N130 X40
N140 Y40
N150 X0
N160 Y0
N170 M30
```

19.7.9.2 回転テーブルのある旋回工具ホルダの例

MOVT 命令の使用

MOVT 命令の使用に関して、B 軸の回転の場合に工具が Y 軸を中心として回転する 5 軸機械でプログラムが実行されるとみなされます。

```
N10 TRAORI()
N20 X0 X0 Z0 B45 F2000                    ; 工具向きの設定
N30 MOVT=-10                              ; 工具方向に 10 mm の送り動作
                                          ; (Y-Z 平面で 45 度未満)
N40 MOVT=AC(20)                           ; 工具方向における後退、距離は
                                          ; 原点から 20 mm
```

回転テーブルのある機械

回転テーブルのある旋回工具ホルダの使用に対するすべての定義

```
N10 $TC_DP1[1,1] =120
N20 $TC_DP3[1,1] = 13                     ; 工具長 13 mm

; 工具ホルダ 1 の定義:

N30 $TC_CARR1[1] = 0                      ; 1 番目のオフセットベクトルの X コンポーネント
N40 $TC_CARR2[1] = 0                      ; 1 番目のオフセットベクトルの Y コンポーネント
N50 $TC_CARR3[1] = 0                      ; 1 番目のオフセットベクトルの Z コンポーネント

N60 $TC_CARR4[1] = 0                      ; 2 番目のオフセットベクトルの X コンポーネント
N70 $TC_CARR5[1] = 0                      ; 2 番目のオフセットベクトルの Y コンポーネント
N80 $TC_CARR6[1] = -15                    ; 2 番目のオフセットベクトルの Z コンポーネント

N90 $TC_CARR7[1] = 1                      ; 1 番目の軸の X 成分
N100 $TC_CARR8[1] = 0                     ; 1 番目の軸の Y 成分
N110 $TC_CARR9[1] = 0                     ; 1 番目の軸の Z 成分

N120 $TC_CARR10[1] = 0                    ; 2 番目の軸の X 成分
N130 $TC_CARR11[1] = 1                    ; 2 番目の軸の Y 成分
N140 $TC_CARR12[1] = 0                    ; 2 番目の軸の Z 成分

N150 $TC_CARR13[1] = 30                   ; 1 番目の軸の回転角度
N160 $TC_CARR14[1] =-30                   ; 2 番目の軸の回転角度

N170 $TC_CARR15[1] = 0                    ; 3 番目のオフセットベクトルの X コンポーネント
N180 $TC_CARR16[1] = 0                    ; 3 番目のオフセットベクトルの Y コンポーネント
```

```
N190 $TC_CARR17[1] = 0                    ; 3 番目のオフセットベクトルの Z コンポーネント

N200 $TC_CARR18[1] = 0                    ; 4 番目のオフセットベクトルの X コンポーネント
N210 $TC_CARR19[1] = 0                    ; 4 番目のオフセットベクトルの Y コンポーネント
N220 $TC_CARR20[1] = 15                   ; 4 番目のオフセットベクトルの Z コンポーネント

N230 $TC_CARR21[1] = A                    ; 1 番目の軸の基準
N240 $TC_CARR22[1] = B                    ; 2 番目の軸の基準
N250 $TC_CARR23[1] = "P"                  ; 工具ホルダのタイプ

N260 X0 Y0 Z0 A0 B45 F2000
N270 TCARR=1 X0 Y10 Z0 T1 TCOABS
N280 PAROT
N290 X0 Y0 Z0
N300 G18 MOVT=AC(20)
N310 G17 X10 Y0 Z0
N320 MOVT=-10
N330 PAROTOF
N340 TCOFR
N350 X10 Y10 Z-13 A0 B0
N360 ROTS X-45 Y45
N370 X20 Y0 Z0 D0
N380 Y20
N390 X0 Y0 Z20
N400 M30
```

旋回工具ホルダの定義がすべて示されています。コンポーネントはいずれにしても 0 にプリセットされているため、値 0 を含むコンポーネントを実際に指定する必要はありません。

この工具ホルダは N270 で起動されます。

$TC_CARR21 および *$TC_CARR22* で機械軸 A および B が参照され、TCOABS が有効なため、*$TC_CARR13* および *$TC_CARR14* の値は無視されます。つまり、軸位置 A0 B45 が回転に使用されます。

B 軸を中心とする 4 番目のオフセットベクトル(Z 方向に長さ 15 mm)の回転によって、X10.607 [= 15 * sin(45)]および Z-4.393 [= -15 * (1. - cos(45))]だけ原点のオフセットが発生します。この原点オフセットは、位置 X10.607 Y10.000 Z8.607 にアプローチするように、自動的に記述される基本またはシステムフレームで考慮に入れられます。Z 方向では工具選択によって 13 mm の追加オフセットがもたらされます。Y コンポーネントはテーブル回転による影響を受けません。

N280 では、旋回工具ホルダのテーブルの回転に応じた回転が定義されます。したがって新しい X 方向は、第 4 象限の二分線の方向を指し、新しい Z 軸は第 1 象限の二分線の方向を指します。

N290 で原点にアプローチします。つまり、原点の位置は回転によって変更されないため、機械位置 X10.607 Y0 Z-4.393 にアプローチします。

N300 では Y 方向に位置 Y33.000 まで移動します。G18 が有効であり、Y コンポーネントは有効フレームによって影響を受けないからです。X および Z 位置は変更されません。

N310 では位置 X17.678 Y0 Z1.536 にアプローチします。

N320 では MOVT 命令の結果として、Z 位置のみが値-8.464 に変更されます。テーブルのみ回転できるため、有効なフレームの Z 方向が 45 度回転しても、工具の向きは機械の Z 方向に平行のままです。

N330 では基本またはシステムフレームが削除されます。したがって、N280 でのフレーム定義が元に戻されます。

N340 では、TCOFR によって、旋回工具ホルダを有効なフレームに従って位置合わせするよう指定されます。N330 では PAROTOF 命令によって回転が有効でなくなっているため、初期設定が適用されます。フレームオフセットは 0 になります。

したがって、N350 では位置 X10 X10 Z0(= Z-13 + 工具長)にアプローチします。重要:回転軸 A および B の両方を同時にプログラムすることで、旋回工具ホルダの実際の位置が、N340 で使用される位置に一致させられます。ただし、3 つの直線軸がアプローチする位置は、この位置に依存します。

N360 では、立体角を使用して、XZ 平面および YZ 平面での交差線がそれぞれ X 軸または Y 軸と+45 度または-45 度の角度を成す平面が定義されます。したがって、そのように定義される平面の位置は、次のようになります。平面の法線が立体対角線方向を指します。

N370 では、新しい座標系内の位置 X20 Y0 Z0 に移動します。同時に D0 で工具が解除されるため、Z 方向の追加オフセットはなくなります。新しい X 軸は旧 X-Z 平面上にあるため、このブロックでは機械位置 X14.142 Y0 Z-14.142 に到達します。

N380 では、回転した座標系の Y 軸上を移動するだけです。これによって 3 つの機械軸すべてが移動します。機械位置は X5.977 Y16.330 Z-22.307 です。

N390 では、新しい Z 軸上の点にアプローチします。これは、機械軸に対して立体対角線上にあります。したがって、3 つの軸がすべて位置 11.547 に到達します。

19.7.9.3 位置別およびワーク別の補正値の計算

アダプタを備えた工具

次のプログラム例では、アダプタと旋回工具ホルダを備えた工具が定義されています。概要を単純化するため、サムオフセットおよび挿入オフセットと、工具自体の場合はアダプタについて、長さ L1 のみが 0 と異なります。旋回工具ホルダのオフセットベクトルは、すべて 0 です。

プログラムコード	コメント
N10 $TC_TP2[1]="MillingTool"	; 識別子の名前
N20 $TC_TP7[1]=9	; 位置タイプ
N30 $TC_TP8[1]=2	; 状態:有効および未ブロック
; D corr.D=1	
N40 $TC_DP1[1,1]=120	; 工具タイプ - フライス
N50 $TC_DP3[1,1]=	;長さオフセットベクトル
N60 $TC_DP12[1,1]=	; 摩耗
N70 $TC_SCP13[1,1]=0.1	; サムオフセット DL=1
N80 $TC_ECP13[1,1]=0.01	; 挿入オフセット DL=1
N90 $TC_ADPTT[1]=5	; アダプタ座標変換
N100 $TC_ADPT1[1]=0.001	; アダプタ寸法
; マガジンデータ	
N110 $TC_MAP1[1]=3	; マガジンタイプ:タレット
N120 $TC_MAP2[1]="Turret"	; マガジンの名前
N130 $TC_MAP3[1]=17	; マガジンの状態
N140 $TC_MAP6[1]=1	; 寸法 - 行
N150 $TC_MAP7[1]=2	; 寸法 - 列 -> 2 つの位置
N160 $TC_MPP1[1,1]=1	; 位置タイプ
N170 $TC_MPP2[1,1]=9	; 位置タイプ
N180 $TC_MPP4[1,1]=2	; ロケーションの状態
N190 $TC_MPP7[1,1]=1	; アダプタの配置
N200 $TC_MPP6[1,1]=1	; T 番号「MillingTool」
N210 $TC_MAP1[9999]= 7	; マガジンタイプ:バッファ
N220 $TC_MAP2[9999]="Buffer"	; マガジンの名前
N230 $TC_MAP3[9999]=17	; マガジンの状態
N240 $TC_MAP6[9999]=1	; 寸法 - 行
N250 $TC_MAP7[9999]=1	; 寸法 - 列 -> 1 つの位置

プログラムコード	コメント
N260 $TC_MPP1[9999,1]=2	; 位置タイプ
N270 $TC_MPP2[9999,1]=9	; 位置タイプ
N280 $TC_MPP4[9999,1]=2	; ロケーションの状態
N290 $TC_MPP5[9999,1]=1	; 主軸 No. 1
N300 $TC_MDP2[1,1]=0	; 主軸からマガジンまでの距離 1
; 工具ホルダ 1 の定義	
N310 $TC_CARR10[1]=1	; X 方向における 2 番目の回転軸のコンポーネント
N320 $TC_CARR14[1]=45	; 2 番目の軸の回転角度
N330 $TC_CARR23[1]="T"	; 工具モード
N340 STOPRE	
N350 $SC_WEAR_TRANSFORM='B101'	
N360 T0 D0 DL=0	
N370 ROT X30	
N380 G90 G1 G17 F10000 X0 Y0 Z0	
N390 T="MillingTool" X0 Y0 Z0 TOWSTD	; X 0.000 Y11.110 Z 0.001
N400 T="MillingTool" X0 Y0 Z0 TOWMCS	; X 0.000 Y10.100 Z 1.011
N410 T="MillingTool" X0 Y0 Z0 TOWWCS	; X 0.000 Y 9.595 Z 0.876
N420 TCARR=1 X0 Y0 Z0	; X 0.000 Y 6.636 Z 8.017
N430 G18 X0 Y0 Z0	; X10.100 Y-0.504 Z 0.876
N440 M30	

上の例に関する説明

ブロック N390 から、さまざまな方法を使用して位置 X0 Y0 Z0 にアプローチします。到達する機械位置は、各ブロックのコメントに示されています。プログラムについて、それらの位置への到達方法を説明します。

N390:アダプタ座標変換 5(ブロック N90)によって、長さ L1 が長さ L2 に変換されます。実際のアダプタ寸法のみが、この変換の対象外です。Y 値(G17 での L2)は、工具長(10)、工具磨耗(1)、サムオフセット(0.1)および挿入オフセット(0.01)から算出されます。アダプタ寸法(0.001)は Z(L1)にあります。

N400:ブロック N350 では、ビット 0 および 2 が次のセッティングデータで有効化されます。

SD42935 $SC_WEAR_TRANSFORM (工具成分の座標変換)

つまり、ブロック N400 の TOWMCS のために、工具の磨耗と挿入オフセットはアダプタ座標変換の対象になりません。これら 2 つの補正の合計は 1.01 です。したがって、ブロ

ック N390 と比較して、Z 位置はその値だけ大きく、Y 位置はその値だけ小さくなります。

TOWWCS は N410 で有効です。したがって、工具磨耗と挿入オフセットの合計は、有効なワーク座標系内で作用します。ブロック N370 では、X 軸を中心に 30 度の回転が起動されます。したがって、Z 方向での元の補正値 1.01 により、0.875(= 1.01 * cos(30))の新しい Z 成分と、-0.505(= 1.01 * sin(30))の新しい Y 成分が生成されます。ブロック N390 で生成された工具長、サムオフセット、およびアダプタ寸法の合計にこれが加算されると、プログラムコメントで指定された寸法が算出されます。

また、旋回工具ホルダがブロック N420 で起動されます。これによって、X 軸を中心とする 45 度の回転が実行されます(N310 ～ N330 を参照)。工具ホルダのオフセットベクトルはすべて 0 であるため、追加のゼロオフセットはありません。旋回工具ホルダが工具長、サムオフセットおよびアダプタ寸法の合計に影響を及ぼします。その結果生成されるベクトル成分は、X0 Y7.141 Z7.142 です。これに、ブロック N410 と同様、WCS で評価された工具磨耗と挿入オフセットが加算されます。

N430 で G18 が起動されます。工具長合計、サムオフセットおよびアダプタ寸法の成分が、それに応じて交換されます。旋回工具ホルダは、この新しいベクトル(X 軸を中心とする 45 度の回転)にも引き続き影響を及ぼします。その結果生成されるベクトル成分は、X10.100 Y0.0071 Z0.0071 です。工具摩耗および挿入オフセットから生成されたベクトル(X0 Y-0.505 Z0.875)は、平面の変更の影響を受けません。2 つのベクトルの合計によって、 N430 のコメントで指定された寸法が生成されます。

19.7.9.4 例:キネマティック結合による方向付け機能を備えた工具キャリア

混合キネマティックによる工具ホルダの例(旋回工具およびテーブル)

この例は、キネマティック結合によって定義された工具ホルダを示しています。結合は、2 つの別個の結合(工具とテーブル)の終点で自動的に接続します。

プログラムコード	コメント
	; 定義工具ホルダ 1
N1070 _MIXED_1_S:	
N1080 IF (DELOBJ("KIN_CHAIN_ELEM") <> 0)	
N1080 IF (DELOBJ("KIN_CHAIN_ELEM") <> 0)	
N1090 SETAL(62000)	
N1100 ENDIF	
N1110 _KIE_CNT = 0	
N1120 _CARR_CNT = 1	
N1130 TCARR=0	; 現在の工具ホルダを無効にします。
N1140 $TC_CARR1[0] = 0	; すべての工具ホルダデータを削除します。

プログラムコード	コメント
N1150 $TC_CARR_KIN_CNTRL[_CARR_CNT]=7	; キネマティック結合による工具ホルダ
	; + 工具と構成部分の結合を接続します。
N1160 $TC_CARR_KIN_TOOL_START[_CARR_CNT]="Y_AXIS"	; 工具結合の開始エレメント。
N1170 $TC_CARR_KIN_TOOL_END[_CARR_CNT]="B_OFFSET_CORR"	; 工具結合の終了エレメント。
N1180 $TC_CARR_KIN_PART_START[_CARR_CNT]="X_AXIS"	; 構成部分結合の開始エレメント。
N1190 $TC_CARR_KIN_PART_END[_CARR_CNT]="END_PART_CHAIN"	; 構成部分結合の終了エレメント。
N1200 $TC_CARR21[_CARR_CNT]=B	
N1210 $TC_CARR22[_CARR_CNT]=C	
N1220 $TC_CARR23[_CARR_CNT]="M"	
N1230 $NK_NAME[_KIE_CNT] = "ROOT"	
N1240 $NK_PARALLEL[_KIE_CNT]="X_AXIS"	
N1250 $NK_TYPE[_KIE_CNT] = "OFFSET"	
N1260 $NK_NEXT[_KIE_CNT] = "Y_AXIS"	
N1270 _KIE_CNT=_KIE_CNT+1	
	;***** 工具結合の開始 *****
N1280 $NK_NAME[_KIE_CNT] = "Y_AXIS"	
N1290 $NK_TYPE[_KIE_CNT] = "AXIS_LIN"	
N1300 $NK_OFF_DIR[_KIE_CNT,1] = 1	
N1310 $NK_AXIS[_KIE_CNT] = "Y1"	
N1320 $NK_NEXT[_KIE_CNT] = "Z_AXIS"	
N1330 _KIE_CNT=_KIE_CNT+1	
N1340 $NK_NAME[_KIE_CNT] = "Z_AXIS"	
N1350 $NK_TYPE[_KIE_CNT] = "AXIS_LIN"	
N1360 $NK_OFF_DIR[_KIE_CNT,2] = 1	
N1370 $NK_AXIS[_KIE_CNT] = "Z1"	
N1380 $NK_NEXT[_KIE_CNT] = "CLOSE_HEAD"	
N1390 _KIE_CNT=_KIE_CNT+1	
N1400 $NK_NAME[_KIE_CNT] = "CLOSE_HEAD"	
N1410 $NK_TYPE[_KIE_CNT] = "OFFSET"	
N1420 $NK_OFF_DIR[_KIE_CNT,0] = 0	
N1430 $NK_OFF_DIR[_KIE_CNT,1] = 0	
N1440 $NK_OFF_DIR[_KIE_CNT,2] = 0	
N1450 $NK_NEXT[_KIE_CNT] = "B_AXIS"	
N1460 _KIE_CNT=_KIE_CNT+1	
N1470 $NK_NAME[_KIE_CNT] = "B_AXIS"	
N1480 $NK_TYPE[_KIE_CNT] = "AXIS_ROT"	
N1490 $NK_OFF_DIR[_KIE_CNT,1] = 1	
N1500 $NK_OFF_DIR[_KIE_CNT,2] = 1	
N1520 $NK_A_OFF[_KIE_CNT] = -27	
N1540 $NK_AXIS[_KIE_CNT] = "B1"	
N1550 $NK_NEXT[_KIE_CNT] = "B_OFFSET"	
N1560 _KIE_CNT=_KIE_CNT+1	
N1570 $NK_NAME[_KIE_CNT] = "B_OFFSET"	
N1580 $NK_TYPE[_KIE_CNT] = "OFFSET"	

プログラムコード	コメント
N1590 $NK_OFF_DIR[_KIE_CNT,2] = -30.6	
N1600 $NK_NEXT[_KIE_CNT] = "B_OFFSET_CORR"	
N1610 _KIE_CNT=_KIE_CNT+1	
N1620 $NK_NAME[_KIE_CNT] = "B_OFFSET_CORR"	
N1630 $NK_TYPE[_KIE_CNT] = "OFFSET"	
N1640 $NK_NEXT[_KIE_CNT] = ""	
N1650 _KIE_CNT=_KIE_CNT+1	
	;***** 工具結合の終了 *****
	;***** 構成部分結合の開始 *****
N1660 $NK_NAME[_KIE_CNT] = "X_AXIS"	
N1670 $NK_TYPE[_KIE_CNT] = "AXIS_LIN"	
N1680 $NK_OFF_DIR[_KIE_CNT,0] = 1	
N1690 $NK_AXIS[_KIE_CNT] = "X1"	
N1700 $NK_NEXT[_KIE_CNT] = "C_OFFSET"	
N1710 _KIE_CNT=_KIE_CNT+1	
N1720 $NK_NAME[_KIE_CNT] = "C_OFFSET"	
N1730 $NK_TYPE[_KIE_CNT] = "OFFSET"	
N1740 $NK_OFF_DIR[_KIE_CNT,0] = 300	
N1750 $NK_OFF_DIR[_KIE_CNT,1] = 150	
N1760 $NK_NEXT[_KIE_CNT] = "C_OFFSET_CORR"	
N1770 _KIE_CNT=_KIE_CNT+1N1780 $NK_NAME[_KIE_CNT] = "C_OFFSET_CORR"	
N1790 $NK_TYPE[_KIE_CNT] = "OFFSET"	
N1800 $NK_OFF_DIR[_KIE_CNT,0] = 0	
N1810 $NK_OFF_DIR[_KIE_CNT,1] = 0	
N1820 $NK_OFF_DIR[_KIE_CNT,2] = 0	
N1830 $NK_NEXT[_KIE_CNT] = "C_AXIS"	
N1840 _KIE_CNT=_KIE_CNT+1	
N1850 $NK_NAME[_KIE_CNT] = "C_AXIS"	
N1860 $NK_TYPE[_KIE_CNT] = "AXIS_ROT"	
N1870 $NK_OFF_DIR[_KIE_CNT,2] = -1	
N1890 $NK_A_OFF[_KIE_CNT] = -58	
N1910 $NK_AXIS[_KIE_CNT] = "C1"	
N1920 $NK_NEXT[_KIE_CNT] = "CLOSE_PART"	
N1930 _KIE_CNT=_KIE_CNT+1	
N1940 $NK_NAME[_KIE_CNT] = "CLOSE_PART"	
N1950 $NK_TYPE[_KIE_CNT] = "OFFSET"	
N1960 $NK_OFF_DIR[_KIE_CNT,0] = 0	
N1970 $NK_OFF_DIR[_KIE_CNT,1] = 0	
N1980 $NK_OFF_DIR[_KIE_CNT,2] = 0	
N1990 $NK_NEXT[_KIE_CNT] = "END_PART_CHAIN"	
N2000 _KIE_CNT=_KIE_CNT+1	
N2010 $NK_NAME[_KIE_CNT] = "END_PART_CHAIN"	
N2020 $NK_TYPE[_KIE_CNT] = "OFFSET"	

プログラムコード	コメント
	;***** 構成部分結合の終了 *****

19.8 旋回工具のオフセットデータの変更

19.8.1 はじめに

機能

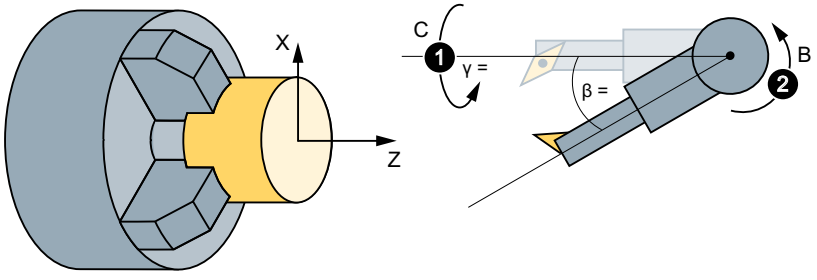
"旋回工具のオフセットデータの変更"機能を使用すると、工具(主に旋削工具だが、穴あけ工具およびフライス工具も)が回転するときに、加工されているワークに関して取得される変更された幾何学的関係を考慮できます。

現在の工具回転は常に、現在有効な旋回工具ホルダ(章「旋回工具ホルダ (ページ 1860)」を参照)またはキネマティック結合により定義されたオリエンテーション変換(機能マニュアルの応用機能の章「キネマティック結合による変換の定義」を参照)により決定されます。

旋回工具ホルダの回転の角度は、通常(ただし必ずしもそうではない)有効なフレームから TCOFR 命令により定義されます。この方法は、工具を回転する実際のキネマティクと無関係に工具の向きを定義するために使用できます。これは 2 つの角度を利用する場合とまったく同様です。

有効なキネマティックトランスフォーメーションでは、必要な角度は機能 ORISOLH により決定できます(「オリエンテーションの計算(ORISOLH) (ページ 1927)」を参照)。

機械と無関係な 2 つの向きの角度 β (ベータ)と γ (ガンマ)を使用して、工具の回転を定義します。β は垂直軸(通常 G18 の B 軸)まわりの回転角度であり、γ は垂直軸(通常 G18 の C 軸)まわりの回転角度です。回転はまず γ の周囲で実行され、最後に β の周囲で実行されます。すなわち、γ 軸は β 軸により回転されます:



① 横座標を中心とした回転

② 垂直軸を中心とした回転

図 19-49　工具の回転

セットアップ

この機能は、マシンデータとセッティングデータを使用してセットアップします。

「セットアップ (ページ 1923)」の章を参照してください。

起動

旋回工具のオフセットデータの変更は、言語命令 CUTMOD (旋回工具ホルダと組み合わせた場合)または CUTMODK (キネマティック結合により定義されたオリエンテーション変換の場合)により NC プログラムで有効になります。

「旋回工具のオフセットデータの変更の起動(CUTMOD、CUTMODK) (ページ 1941)」の章を参照してください。

結果

機能および工具回転が有効な場合、変更されたデータはシステム変数および OPI 変数で使用可能になります。

「旋回工具のオフセットデータの変更の起動(CUTMOD、CUTMODK) (ページ 1941)」の章を参照してください。

19.8.2　旋削工具の旋回

19.8.2.1　旋削工具の刃先位置、切削方向、および角度

旋削工具

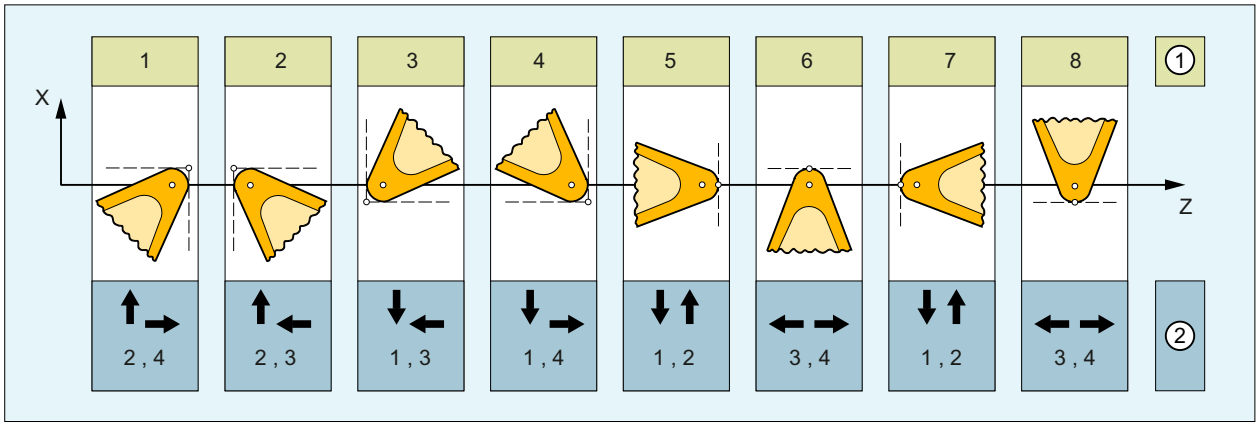
以降、旋削工具とは、工具タイプ($TC_DP1)の値が 500 ～ 599 の範囲にある工具を意味します。研削工具(工具タイプ 400 ～ 499)も、旋削工具と同等に扱われます。

工具は、下記の場合、工具タイプとは無関係に旋削工具として処理されます。

SD42950 $SC_TOOL_LENGTH_TYPE = 2

刃先位置と切削方向

旋削工具は、主要な刃先と補助的な刃先により区切られます。刃先位置は、座標軸を基準にした主要な刃先と補助的な刃先の位置により定義されます。刃先位置ごとに、2 つの異なる切削方向を割り当てることができます。



① 刃先位置(1-8)

② 刃先位置への切削方向(1 - 4)の割り当て

図 19-50 刃先位置と切削方向

刃先位置

刃先位置 1 - 4 では、主要な刃先と補助的な刃先は同じ象限にあります。

5 - 8 の刃先位置では、主要な刃先と補助的な刃先は、隣接する象限にあり、また 2 つの刃先間に座標軸があります。

刃先位置は、工具パラメータ$TC_DP2 に格納されます。

切削方向

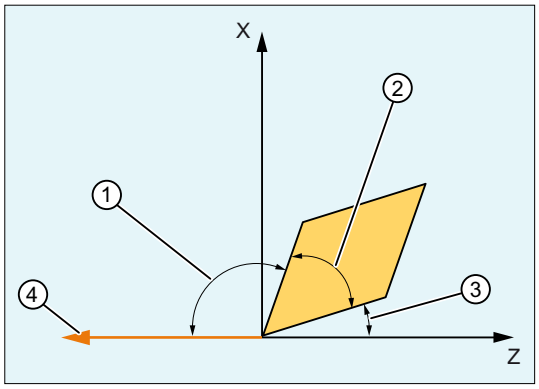
切削方向 1 - 4 は、座標軸の正方向または負方向を記述します:

- 1:縦座標-
- 2:縦座標+
- 3:横座標-
- 4:横座標+

この切削方向は、工具パラメータ$TC_DP11 に格納されます。

ホルダ角度と逃げ角

以下の図は、刃先位置 3 の旋回工具に対するホルダ角度と逃げ角度を示します。加工平面は G18 (Z/X)です。切削方向は、3 (負の Z 方向または横軸方向)です。



① ホルダ角度

② くさび角度またはプレート角度 = 180° - ホルダ角度 - 逃げ角度

③ 逃げ角

④ 切削方向

図 19-51 旋削工具の角度と切削方向

切削方向

切削方向が、ホルダ角度の基準の方向を示します。逃げ角は、切削方向とは逆の方向と隣接した刃先の間で測定される角度です(正)。

ホルダ角度と逃げ角は、工具パラメータ$TC_DP10 または$TC_DP24 に格納されます。

注記

切削方向と工具角度は、刃先位置 1 ～ 8 の場合のみ関係します。

19.8.2.2 旋削工具の回転時の修正

工具の向きの説明

フライス工具とは異なり、旋削工具は自身の回転に関して対称ではありません。つまり、工具の向きを表すには、通常 3 次の自由度または 3 つの回転軸が必要です。したがって、具体的なキネマティクは、希望する向きに設定できる範囲においては、機械と無関係です。必要に応じて、3 番目の自由度を工具座標系の回転により代用できます。

注記

旋回工具ホルダにより作成される 1 つの部分と、座標系の回転を介して実現できる 2 番目の部分に向きを分割することは、アプリケーションに委ねられます。これに関して、コントローラは何も追加機能を提供しません。

キネマティックトランスフォーメーションにより工具のオリエンテーションが設定されている場合、ユーザーはオリエンテーションを計算するために機能 ORISOLH を使用できます(章「オリエンテーションの計算(ORISOLH) (ページ 1927)」を参照)。

工具差し込みと加工平面間の角度

旋削工具が、加工平面に対して(すなわち、加工平面の軸まわりで、通常 C 軸まわりで)180 度の倍数でない角度だけ回転する場合、加工平面の(円形)工具刃先の形状は楕円になります。このような回転を考慮して生じる円形からのずれは、あまり重要ではないので無視できると仮定します(傾斜角 5°未満)。すなわち、コントローラでは常に工具の向きを無視し、刃先を円形と見なします。

これは、有効な平面に関して、コントローラでは最初の位置からのずれの設定として、180 度の回転のみを認めるということも意味します。この制限は、刃先の形状に対してのみ有効です。工具長は、任意の空間の回転において常に正確に考慮されます。

加工平面の軸まわりに 180 度だけ回転することは、工具を同じ位置で使用する場合、回転していない工具の使用に対して、主軸回転方向を反転しなければならないことを意味します。

工具差し込みと加工平面間の角度の最大許容値は、セッティングデータ SD42998 $SC_CUTMOD_PLANE_TOL で指定できます(章「パラメータ割り当て (ページ 1923)」を参照)。

刃先位置、切削方向、および工具角度

工具が平面から± 90°(許容範囲は約 1°)回転される場合、刃先位置と切削方向は刃先基準点と同様に修正されません(以下を参照してください)。刃先の設定が選択平面で定義されないためです。

変換が有効な場合、有効な全フレームが常に、ワークに対する現在の工具回転の計算に含まれます。旋回工具ホルダが有効な場合、マシンデータ MD20360 $MC_TOOL_PARAMETER_DEF_MASK (ページ 1923) でビット 21 が設定されているときにのみ、フレームが計算に含まれます。この場合、このフレーム成分が全フレーム(システムフレーム PAROT)にすでに含まれていることを想定する必要があるために、工具ホルダのどのテーブル成分も個別には扱われなくなります。

オリエンテーション変換が有効な場合、刃先がすでに有効な加工平面にないときは、変更後の刃先データの計算のために、マシンデータ MD20360 $MC_TOOL_PARAMETER_DEF_MASK のビット 17 を使用して、刃先を加工平面に投影するか(ビット 17=1)、または刃先を機械平面に旋回できます(ビット 17=0)。

注記

MD20360、ビット 17

このビットを設定した場合ユーザーは、加工時の幾何学的条件が計算時に存在する条件と一致するように、追加フレームを使用するか、またはオリエンテーションを変更する必要があります。

マシンデータ MD20360 $MC_TOOL_PARAMETER_DEF_MASK のビット 20 を値"0"に設定した場合、ホルダ角度および切削方向の刃先パラメータ($TC_DP10 または $TC_DP24)が値 0 の場合に、変更後の刃先位置および変更後の切削方向を計算するために、0 以外の標準値を使用できます。これは下記のように適用されます。

- 0°のホルダ角度は、刃先位置 1 ～ 4 で 112.5°の角度と同じ動作になります。
- 0°のホルダ角度は、刃先位置 5 ～ 8 で 67.5°の角度と同じ動作になります。
- 0°の逃げ角は、刃先位置 1 ～ 4 で 22.5°の角度と同じ動作になります。
- 0°の逃げ角は、刃先位置 5 ～ 8 で 67.5°の角度と同じ動作になります。

ホルダ角度と逃げ角度は、変更後の刃先位置と変更後の切削方向の計算を可能にするために、上記で指定された標準値に一時的に割り当てることのみが可能です。つまり、システム変数$TC_DP10 または$TC_DP24 を読み取ると、依然として値"0"が返されます。

工具が平面で回転する場合(加工平面に垂直な軸まわりの回転または G18 の Y 軸まわりの回転)、刃先位置は逃げ角とホルダ角度の結果として生じる角度から決定されます。これらの 2 つの角度が工具に対して指定されていない(すなわち、$TC_DP10 と$TC_DP24 が両方ともゼロ)場合、新しい刃先位置は回転角度のみから決定されます。ここでの特記

事項は、刃先位置が 90 度単位でのみ変化することです。すなわち、刃先位置は数値の範囲 1 ～ 4 または 5 ～ 8 のいずれかの初期状態と無関係のままです。新しい刃先位置は、ホルダ角度と逃げ角に指定された値が許容されない場合(負の値、得られる工具差し込み角度が負または 90 度超)、回転角度によってのみ決定されます。許容されない角度を設定すると、アラームが出力されます(MD20125 $MC_CUTMOD_ERR (ページ 1923)を参照)。これらの場合のすべてにおいて、逃げ角とホルダ角度は修正されません。

回転に応じて、逃げ角が 90 度未満のままになるように切削方向が修正されます。元の切削方向と元の刃先位置が一致しない場合、工具の回転時に切削方向が修正されません。この場合、アラームが出力されます(MD20125 $MC_CUTMOD_ERR を参照)。

旋回工具ホルダまたは有効なキネマティックトランスフォーメーションから決定される平面内の回転角は、OPI 変数 pTCutMod または ptCut-ModS、およびシステム変数 $P_CUTMOD_ANG または$AC_CUTMOD_ANG によって提供されます。この角度は、45 度または 90 度の倍数への最終的なまるめをしない元の角度です。

特別な事例

旋削工具に関して、刃先位置、切削方向、逃げ角およびホルダ角度が正当な値を持ち、平面の適切な回転により、すべての刃先位置(1 ～ 8)が得られるとき、刃先(主要な刃先または補助的な刃先)の 1 つが座標軸から離れる量が最小入力単位の半分(小数点以下 3 桁の入力指定の場合は 0.0005 度)未満である場合、刃先位置 1 ～ 4 は刃先位置 5 ～ 8 より優先されます。

以下は、回転が 90 度単位でのみ可能な、その他のすべての場合に適用されます(有効な刃先パラメータのないフライス工具または旋削工具)。回転角度の量が 45°+ 0.5 入力単位(小数点以下 3 桁の入力指定の場合 45.0005°に対応)より小さい場合、刃先位置と切削方向は変更されません。すなわち、これらの場合は 45°より小さい回転として処理されます。180°からのずれの量が 45°+ 0.5 入力単位未満である回転は、135°～ 225°の範囲の回転として処理されます。

刃先基準点

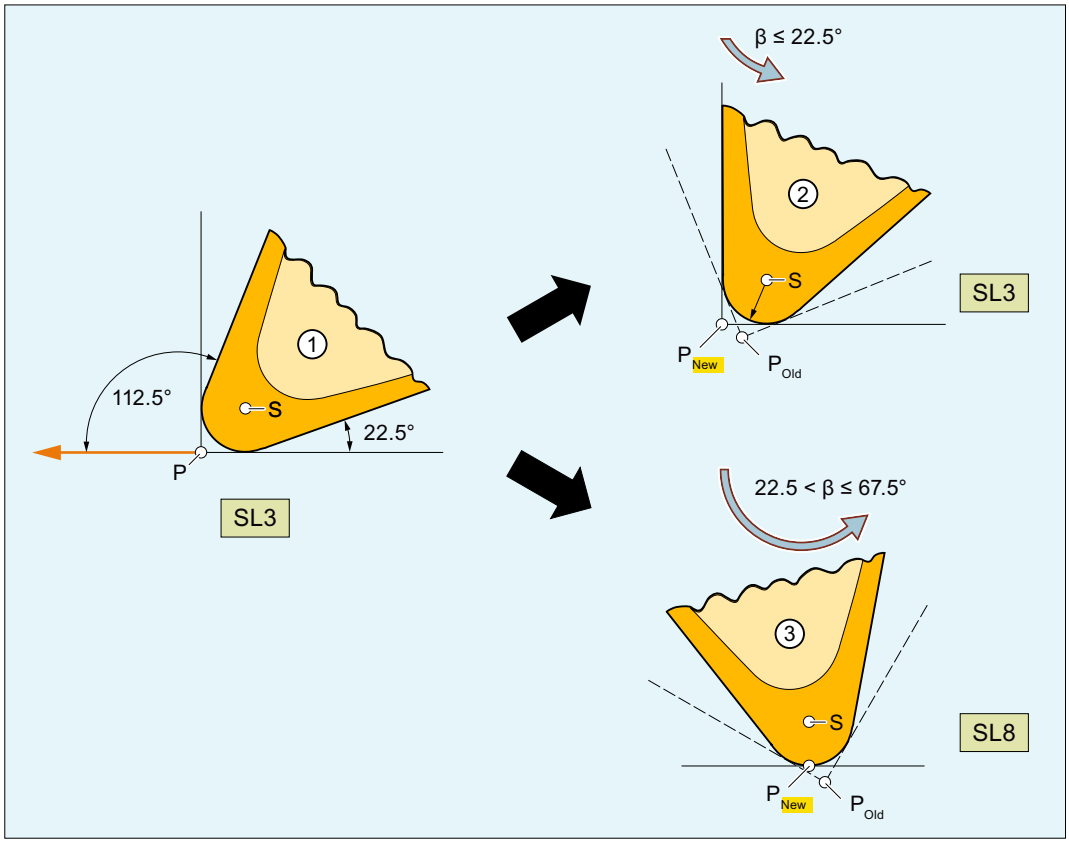
刃先中心点と刃先基準点が、旋削工具に対して定義されます。これらの 2 つの点のお互いの相対位置は、刃先位置により定義されます。

刃先位置 1 ～ 4 の 2 つの点の距離は、刃先半径の√2 倍に等しくなります。刃先位置 5 ～ 8 の 2 つの点の距離は、刃先半径に等しくなります。最初のケースでは、刃先中心点に対する刃先基準点は加工平面の座標軸がなす角を二分する方向にあります。一方、2 番目のケースでは、座標軸方向にあります。

工具を加工平面に垂直な軸まわりの任意の角度により回転させるとき、刃先基準点が工具に対して固定位置がある場合は、それも回転します。このとき、ほとんどの場合、上記の条件(軸上の位置または軸の二分線)は満たされません。これは、望ましくありませ

ん。代わりに、刃先基準点は、刃先基準点と刃先中心点の間の距離ベクトルが上記の 8 つの方向の 1 つになるように、常に修正されるべきです。このために、刃先位置を必要に応じて修正してください。

この関係は、下図に例として示されています。



S: 刃先の中心点

P: 刃先基準点

SL: 刃先位置

① 刃先位置 3、逃げ角度 22.5°、およびホルダ角度 112.5°の工具

② 工具の回転が 22.5°以内の場合、刃先位置は維持され、工具を基準とした刃先基準点の位置は、両方の点の相対位置が加工平面で維持されるように補正されます。

③ 工具がより大きく回転した場合(67.5°まで)、刃先位置は値 8 に変わります。

図 19-52 工具の回転に対する刃先基準点と刃先位置

注記

刃先基準点は工具長ベクトルにより定義されるため、刃先基準点を変更すると、有効な工具長が変更されます。

工具の回転

工具先端および工具アダプタの回転は、工具パラメータ$TC_DPROT によって記述されます。このパラメータ(角度)は、対称的に回転しない工具(たとえば旋削工具、ボーリングバーなど)でのみ有用です。回転は、工具の法線ベクトルと工具のオリエンテーションベクトルのベクトル積によって定義される方向を中心として行われます。有効な工具では、このベクトルはシステム変数$P_TOOLBIN により読み取ることができます。標準の旋削工具では、このベクトルは G18 の Z 軸を参照します。つまり、角度 γ の回転軸でもある軸を中心として回転します。

19.8.3 フライス工具と穴あけ工具の回転

19.8.3.1 フライス工具およびタッピング工具の刃先位置

フライス工具と穴あけ工具

以降、フライス工具および穴あけ工具とは、工具タイプ($TC_DP1)の値が 100 ～ 299 の範囲にある工具を意味します。

工具は、下記の設定では、工具タイプと無関係に、フライス工具およびタッピング工具として処理されます。

SD42950 $SC_TOOL_LENGTH_TYPE = 1

刃先位置

相応に修正される刃先位置は、上記で定義されたフライス工具およびタッピング工具でも導入されます(章「フライス工具およびタッピング工具の回転時の修正 (ページ 1922)」を参照)。

注記

フライス工具またはタッピング工具ではない工具、あるいは上記の定義に準拠する旋削工具ではない工具で宣言された刃先位置は、評価されません。

フライス工具およびタッピング工具の刃先位置は、旋削工具の場合と同様に工具パラメータ$TC_DP2 に格納されます。旋削工具の刃先位置の定義に基づき、このパラメータは値 5 ～ 8 と見なすことができます。ここでは、刃先位置は工具の向き(回転軸の方向)を指定します。

刃先位置	工具の回転軸の方向
5	横座標+
6	縦座標+
7	横座標-
8	縦座標-

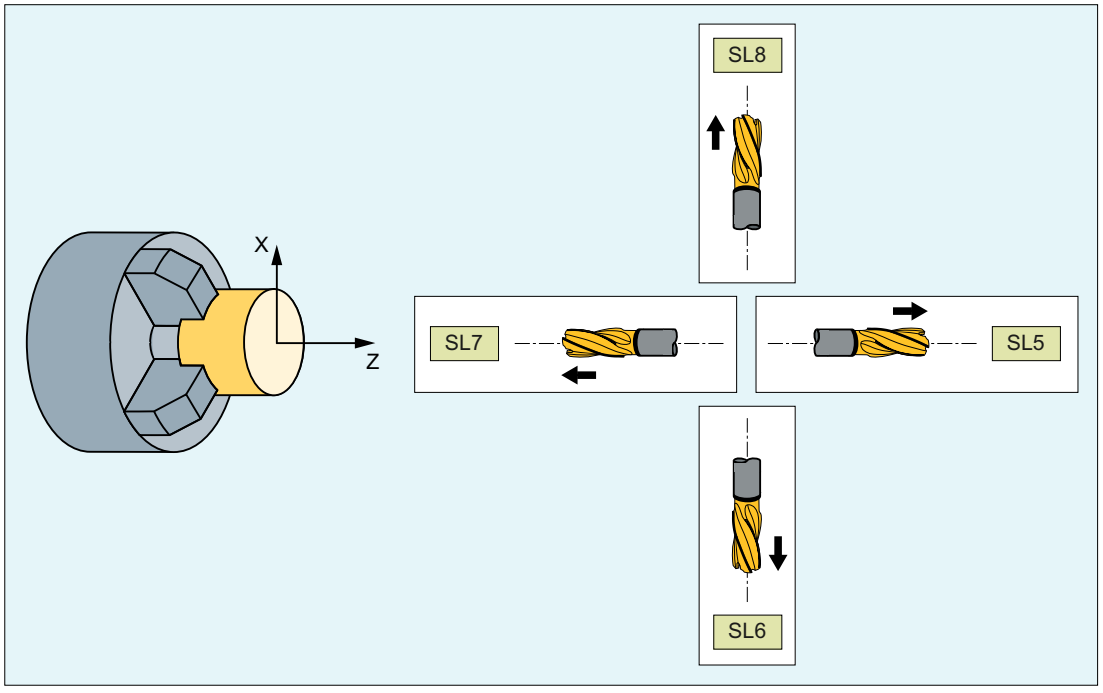


図 19-53　フライス工具の刃先位置 5 ～ 8

19.8.3.2　フライス工具およびタッピング工具の回転時の修正

刃先位置は、フライス工具およびタッピング工具の回転時に適切に再計算されます。切削方向と工具角度(逃げ角またはホルダ角度)は、フライス工具およびタッピング工具に対して定義されていないので、刃先位置の変更は回転によってのみ算出されます。したがって、フライス工具およびタッピング工具に対して、ゼロ設定に対する回転量が 45 度より大きい場合、刃先位置は必ず変更されます。

19.8.4 セットアップ

19.8.4.1 パラメータ割り当て

エラーの場合の応答

(パートプログラム内の明示的な呼び出し、または工具の選択による)機能"旋回工具のオフセットデータの変更"の起動時に、さまざまな故障状態が発生する場合があります。異常番号が 100 未満の異常では(『診断マニュアル』を参照)、異常によりアラームの出力をトリガするかどうか、またアラームの出力をトリガする場合は、そのようなアラームのみを表示するか、またはプログラムストップもトリガするかを定義できます。この設定は、次のマシンデータで行います。

MD20125 $MC_CUTMOD_ERR

マシンデータの 2 つのビットが、個々の故障状態に割り当てられます。

故障ステータス	ビット	意味
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。	0	アラーム出力
	1	プログラムストップ
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。	2	アラーム出力
	3	プログラムストップ
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。	4	アラーム出力
	5	プログラムストップ
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。	6	アラーム出力
	7	プログラムストップ
有効な工具の工具差し込み角度が許容されない値(0°未満または 90°超)です。	8	アラーム出力
	9	プログラムストップ
有効な工具の刃先位置 - ホルダ角度の組み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°、刃先位置 5 ～ 8 のホルダ角度 ≥ 90°でなければならない)。	10	アラーム出力
	11	プログラムストップ
工具差し込みが加工平面になく、工具差し込みと加工平面間の角度 γ がセッティングデータ SD42998 $SC_CUTMOD_PLANE_TOL により指定された上限を超過しています。	12	アラーム出力
	13	プログラムストップ

故障ステータス	ビット	意味
工具差し込みが加工平面にありません。角度 α の大きさが 1°を超えています。角度 α は座標軸を中心とした回転角であり、角度 β の回転軸および角度 γ の回転軸の両方に直交しています(G18 の場合は X 軸)。	14	アラーム出力
	15	プログラムストップ

注記

対応する 1 番目のビット(アラーム出力)が設定されている場合にのみ、2 番目のビット(プログラムストップ)は有効です。

注記

異常番号 100 以上の異常では常に、アラームが出力され、プログラムストップが発生します。

CUTMOD の初期値

NC 命令 CUTMOD (ページ 1927) によりプログラムできる機能は、ウォーム再起動後に以下の値により自動的に初期化されます。

MD20127 $MC_CUTMOD_INIT = <CUTMOD の初期化値>

初期値:0

MD20127 = -2 の場合、以下のマシンデータの値が初期化値になります:

MD20126 $MC_TOOL_CARRIER_RESET_VALUE (リセットで有効な工具ホルダ)

機能に固有の工具設定

以下のマシンデータは、工具に関連するさまざまな機能を制御できるビット文字列です:

MD20360 $MC_TOOL_PARAMETER_DEF_MASK

以下のビットは、機能"旋回工具のオフセットデータの変更"に関連します。

ビット	意味	
17	ビット 17 を使用して、旋削工具および研削工具のオフセットデータの変更時に、オフセット値を計算するために刃先平面を加工平面に投影するか、または回転するかを設定します。	
	= 0	刃先平面は加工平面に投影されます。
	= 1	刃先平面は加工平面まで回転されます。
	注記: ビット 17 は、この機能がキネマティックオリエンテーション変換とともに有効な場合にのみ評価されます。	
18	ビット 18 は、旋削工具および研削工具のオフセットデータを変更するために平面をどのように定義するかを特定します。	
	= 0	G グループ 6 (平面選択 G17-G19)の G 命令による平面の定義よりも、セッティングデータ SD42940 $SC_TOOL_LENGTH_CONST による平面の定義が優先されます。
	= 1	有効な平面(G17 - G19)は常に使用されます。
20	工具パラメータ$TC_DP10 (ホルダ角度)および/または$TC_DP24 (逃げ角)に値「0」が含まれる場合、変更後の刃先位置および変更後の切削方向を計算するために「0」とは異なる初期値を使用できます。	
	= 0	以下の初期値を使用します。 ● 刃先位置 1-4 のホルダ角 112.5° ● 刃先位置 5-8 のホルダ角 67.5° ● 刃先位置 1-4 の逃げ角 22.5° ● 刃先位置 5-8 の逃げ角 67.5°
	= 1	この場合アラームが出力されます。
	注記: ビット 20 は、古いソフトウェアバージョンとの互換性を確立するために使用します。	

ビット	意味	
21	ビット 21 を設定すると、旋回工具ホルダが有効な場合に、オフセットデータの変更で有効な全フレームが考慮されます。	
	= 0	工具ホルダのテーブル成分の回転が考慮されます。フレームは無視されます。
	= 1	工具ホルダのテーブル構成部分の代わりに、有効な全フレームが使用されます。また、全フレームに工具ホルダのテーブル構成部分を含めることができます。

CUTMOD または CUTMODK の工具先端平面および加工平面の差異

機能 CUTMOD または CUTMODK (ページ 1941)の呼び出し時に加工平面(標準 G18)から工具差し込みを遠ざけることができる最大許容角度(2 つ標準位置 0°または 180°のいずれかからの角度 γ の最大許容偏差角)は、以下のセッティングデータで設定します。

SD42998 $SC_CUTMOD_PLANE_TOL

例:

SD42998 = 5.0 ⇒ 工具差し込みは、加工平面から 5°を超えて旋回させることはできません。

注記

SD42998 = 0

SD42998 $SC_CUTMOD_PLANE_TOL を"0"に設定した場合、最大 89°の変化が許容されます!

ORISOLH の工具先端平面と加工平面の差

機能 ORISOLH (ページ 1941)の呼び出し時に工具差し込みを加工平面から遠ざけることができる最大許容角度は、以下のセッティングデータで設定します。

SD42999 $SC_ORISOLH_INCLINE_TOL

19.8.5 プログラミング

19.8.5.1 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。

19.8.5.2 オリエンテーションの計算(ORISOLH):詳細情報

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。

19.8.5.3 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。

19.8.5.4 旋回工具のオフセットデータの変更の起動(CUTMOD、CUTMODK)詳細情報

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。
$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。

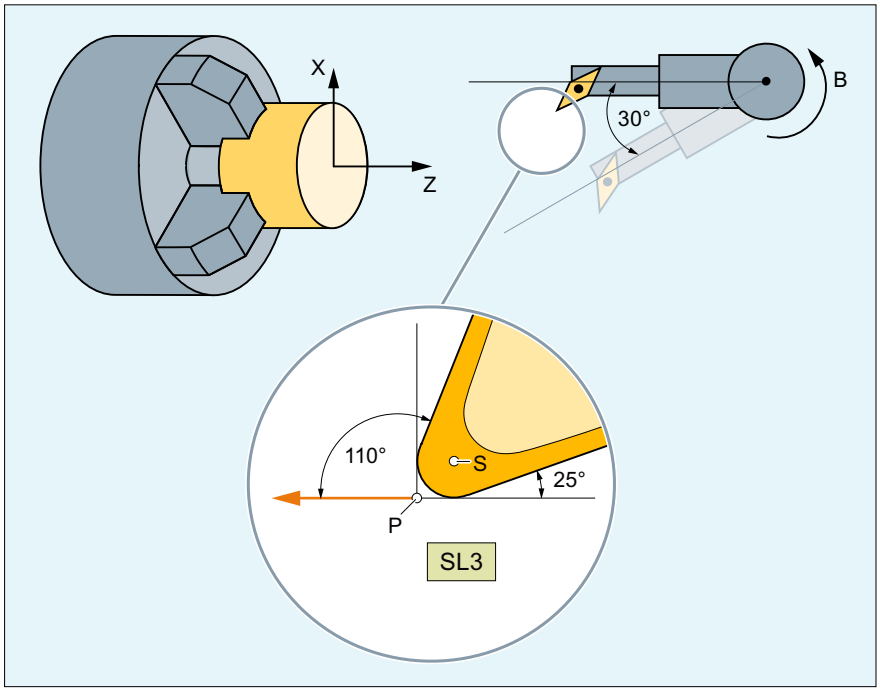
ただし、セッティングデータ SD42940 $SC_TOOL_LENGTH_CONST (平面選択時の工具長成分の変更)がゼロ以外の有効値(正または負の 17、18、または 19)の場合は、その内容で、当該の数量を使用する平面を定義します。

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。

19.8.6 例



S: 刃先の中心点

P: 刃先基準点

SL: 刃先位置

図 19-54 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"	

プログラムコード	コメント
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。

19.9 インクリメンタルでプログラムされた補正値

19.9.1 G91 拡張

必要条件

G91 によるインクリメンタル指令は、工具補正が選択されたときに、インクリメンタル指令値に補正値が追加で移動されるように定義されます。

用途

接触計測などの用途のためには、インクリメンタル座標でプログラム指令軌跡だけを移動することが必要です。 起動された工具補正は移動しません。

処理

インクリメンタル指令での工具補正の選択

- 工具先端でワークに接触します。
- 実際の位置を、工具補正分だけ減算した後、基本フレームに保存します(フィードバック値の設定)。
- 原点からインクリメンタルで移動します。

起動

変更した工具長を FRAME と軸のインクリメンタル指令で移動するかどうか、または、プログラム指令軌跡だけをセッティングデータで移動するかどうかを設定できます。

SD42442 $SC_TOOL_OFFSET_INCR_PROG (工具長補正)

ゼロオフセット/フレーム G91

ゼロオフセットを標準で値 = 1 として FRAME と軸のインクリメンタル指令で移動するかどうか、または、プログラム指令軌跡だけを値 = 0 としてセッティングデータで移動するかどうかを設定できます。

SD42440 $SC_FRAME_OFFSET_INCR_PROG (フレームの原点オフセット)

詳しくは、「K2: 軸タイプ、座標系、フレーム (ページ 845)」の章を参照してください。

補足条件

オフセットがプログラムの終了と RESET の後もそのまま有効になるように動作が設定された場合
MD20110 $MC_RESET_MODE_MASK、ビット 6=1 (リセット/TP 終了後のコントローラの初期設定の指定)
更にインクリメンタル軌跡が最初のプログラムブロックでプログラム指令された場合、補正は必ずプログラム指令軌跡に追加で移動します。

注記
この設定を使用する場合、パートプログラムは必ずアブソリュート指令により開始してください。

19.9.2 工具方向(MOVT)への移動

一般的な用途

旋回工具ホルダが装備された機械上では、軸の一つが工具の方向を指すフレームを起動せずに(たとえば、TOFRAME または TOROT を使用して)、移動を工具方向で実行すべきです(通常、穴あけのとき)。

傾斜面加工動作を実行する際に、傾斜平面を定義するフレームが有効なのに、割り出し工具ホルダ（カップリングギヤシステム）で工具方向の設定が制限されるため工具を正確に直角に設定出来ない機械でも同じです。

これらの場合、平面に垂直な実際に要求された動作とは違って、工具方向にドリルで穴をあける必要があります。そうしないと、ドリルが、その縦軸の方向に誘導されず、とりわけドリルの破損につながることが問題となるからです。

MOVT

こうした動作の終点は、MOVT= ...によりプログラムされます。 プログラムした値は、標準として工具方向にインクリメンタルで作用します。 正の方向は、工具刃先から工具ホルダに向かって定義されます。 このため、MOVT の内容は、通常、切り込み動作(ドリルによる穴あけ時)では負に、後退動作では正になります。 これは、通常の軸に平行な加工を使用した状況、たとえば G91Z ...を使用する場合に相当します。

MOVT=AC(...)形式で動作がプログラムされると、MOVT はアブソリュートで機能します。 この場合、現在の原点を貫く平面が定義され、その表面法線ベクトルは、工具の向きに平行になります。 その後、MOVT で、この平面に対する位置が指定されます。

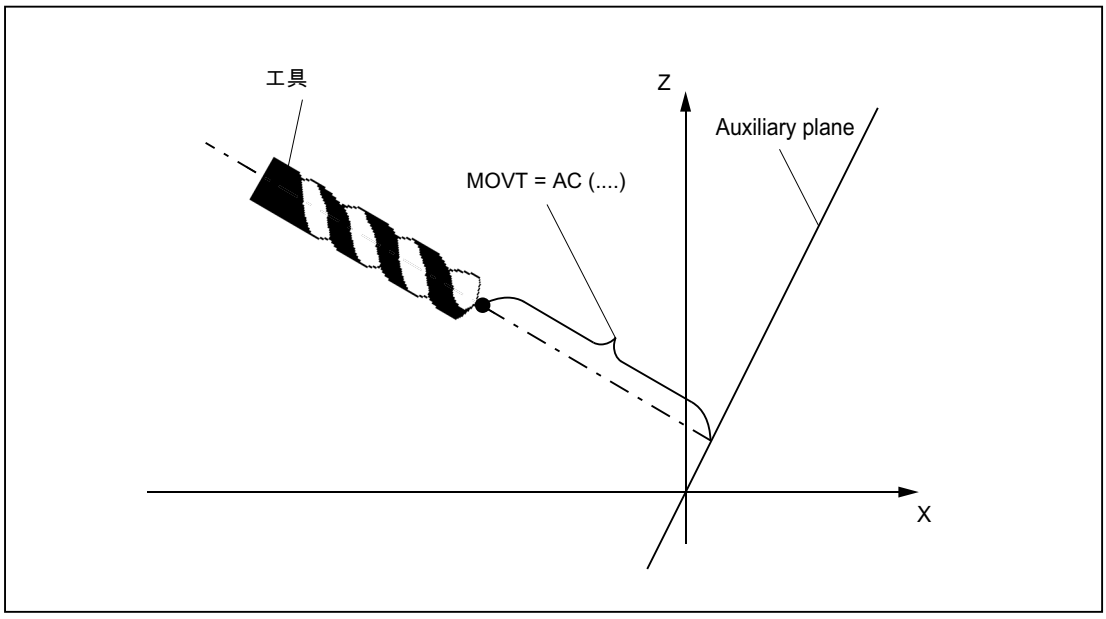


図 19-55　工具方向の動作のアブソリュート指令のための位置の定義

補助平面への参照は、終点位置を計算する役割だけを果たします。 有効なフレームは、この内部計算の影響を受けません。

MOVT がインクリメンタルで機能することを明確に示したい場合には、MOVT= ...の代わりに、MOVT=IC(...)を書き込むこともできます。 この 2 つの形式に機能的違いはありません。

必要条件

以下の必要条件が、MOVT を使用するプログラミングに適用されます。

- これは、旋回工具ホルダの存在とは関係ありません。 動作の方向は、有効な平面によって決まります。 それは垂直軸の方向になります。つまり G17 では Z 方向、G18 では Y 方向、G19 では X 方向です。 これは、旋回工具ホルダが有効でない場合も、旋回工具ホルダがあって、その基本設定で旋回工具のある場合も、ない場合も、これが当てはまります。
- 方向変換(345 軸座標変換)が有効になっている場合、MOVT は同じように動作します。
- MOVT が指定されたブロックで、工具の向きが同時に変更される(たとえば、回転軸の同時補間による有効な 5 軸変換の)場合、MOVT の動作の方向では、ブロックの先頭の向きが重要になります。 工具先端(TCP - 工具中心点)の軌跡は、向きの変更の影響を受けません。

- 直線補間またはスプライン補間(G0、G1、ASPLINE、BSPLINE、CSPLINE)は、有効にしてください。 そうしないと、アラームが発生します。 スプライン補間が有効になっている場合、結果として生じる軌跡は、通常直線ではありません。これは、MOVTにより計算される終点が、X、Y、Z により明確にプログラムされているかのように処理されるからです。
- MOVT を指定したブロックには、ジオメトリ軸をプログラミングしないでください(アラーム 14157)。

19.10 ジオメトリ軸への工具長成分の割り当て

19.10.1 工具タイプおよび作業平面に応じた割り当て

工具パラメータ「長さ 1 ... 3」の値は、システム変数$TC_DP3 ... $TC_DP5 に格納されます(「工具刃先 (ページ 1795)」の章を参照)。ジオメトリ軸への割り当てひいては、結果として得られる工具長成分の切削方向は、工具タイプ($TC_DP1)および有効な加工平面(G17/G18/G19)に依存します。

表 19-1 旋削/研削工具($TC_DP1 = 400 … 599)

作業平面	長さ 1	長さ 2	長さ 3
G17 (X/Y)	Y	X	Z
G18 (Z/X)	X	Z	Y
G19 (Y/Z)	Z	Y	X

表 19-2 フライス工具/特殊工具($TC_DP1 <> 400 … 599)

作業平面	長さ 1	長さ 2	長さ 3
G17 (X/Y)	Z	Y	X
G18 (Z/X)	Y	X	Z
G19 (Y/Z)	X	Z	Y

19.10.2 平面変更時の割り当て

次のセッティングデータが 0 以外に設定された場合、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、加工平面の変更時には変化**しません**。

SD42940 $SC_TOOL_LENGTH_CONST <> 0

ジオメトリ軸への工具長成分の割り当ては、以下の表に示すように、このセッティングデータの 1 および 10 の位から導出されます。

表 19-3　旋削/研削工具($TC_DP1 = 400 … 599)

SD42940	ジオメトリ軸への工具長成分の割り当て		
	Length L1	Length L2	Length L3
= x17	Y	X	Z
= x18	X	Z	Y
= x19	Z	Y	X
= -x17	X	Y	Z
= -x18	Z	X	Y
= -x19	Y	Z	X

表 19-4　フライス/特殊工具($TC_DP1 <> 400 … 599)

SD42940	ジオメトリ軸への工具長成分の割り当て		
	Length L1	Length L2	Length L3
= x17	Z	Y	X
= x18	Y	X	Z
= x19	X	Z	Y
= -x17	Z	X	Y
= -x18	Y	Z	X
= -x19	X	Y	Z

記載された 6 つの値のいずれにも等しくない、0 以外の各値は、値「17」(フライス/特殊工具の場合)または「18」(旋削/研削工具の場合)として評価されます。

注記

工具オリエンテーションの成分の割り付け

工具オリエンテーションの成分の割り付けは、SD42940 $SC_TOOL_LENGTH_CONST の影響を受けません。

必要に応じて、以下のセッティングデータを調整する必要があります:

- SD42954 $SC_TOOL_ORI_CONST_M
- SD42956 $SC_TOOL_ORI_CONST_T

「平面変更の場合の工具オリエンテーション (ページ 1958)」の章を参照してください。

19.10.3 工具タイプに依存しない割り当て

次のセッティングデータを使用して、実際の工具タイプ($TC_DP1)に依存しない、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を定義できます。

SD42950 $SC_TOOL_LENGTH_TYPE = <値>

<値>	工具長成分の割り当て
0 (デフォルト値)	システム変数$TC_DP1[...]の指定に従って 以下の工具タイプが区別されます。 • $TC_DP1[...] == 400 ... 599 ⇒ 旋削/研削工具 • $TC_DP1[...] <> 400 ... 599 ⇒ フライス/特殊工具
1	常にフライス/特殊工具として
2	常に旋削/研削工具として

<値>	工具長成分の割り当て	
3	セッティングデータ SD42942 $SC_TOOL_LENGTH_CONST_T を有効にします。 この設定により、フライス/特殊工具と旋削/研削工具に対して別々に、加工平面変更時に有効になる工具長成分の割り当て (ページ 1954)を定義することができます。	
	SD42940 $SC_TOOL_LENGTH_CONST	フライス/特殊工具の場合の割り当て
	SD42942 $SC_TOOL_LENGTH_CONST_T	旋削/研削工具の場合の割り当て
	例: SD42940 $SC_TOOL_LENGTH_CONST = 17 SD42942 $SC_TOOL_LENGTH_CONST_T = 18 フライス/特殊工具の工具長補正は平面 G17 で有効になります。旋削/研削工具の工具長補正は平面 G18 で有効になります。	
> 3	SD42950 = 0 の場合と同様。	

19.11 近軸工具オリエンテーション

19.11.1 基本工具オリエンテーション

基本工具オリエンテーションは、有効な加工平面から得られます。

作業平面	工具オリエンテーション(初期設定)
G17 (X/Y)	Z に平行
G18 (Z/X)	Y に平行
G19 (Y/Z)	X に平行

19.11.2 平面変更の場合の工具オリエンテーション

以下のセッティングデータの設定により、加工平面変更時に工具オリエンテーションがどのように変更されるかを特定します。

- SD42954 $SC_TOOL_ORI_CONST_M (フライス/特殊工具の場合)
- SD42956 $SC_TOOL_ORI_CONST_T (旋削/研削工具の場合)

標準の動作

デフォルト設定(SD42954 および SD42956 = 0)では、以下のように、平面変更によって工具オリエンテーションが変更されます。

平面変更	工具オリエンテーションの変更
G17 → G18	1. Z 座標を中心とした-90°の回転 2. X 座標を中心とした-90°の回転
G18 → G19	
G19 → G17	
G18 → G17	1. X 座標を中心とした 90°の回転 2. Z 座標を中心とした 90°の回転
G19 → G18	
G17 → G19	

平面変更の場合に一定の工具オリエンテーション

SD42954 または SD42956 が 0 に等しくない場合は、右手直交工具座標系が定義されます。これは、加工平面(G17-G19)が変更された場合でも変更されません。オリエンテーション座標系は、オリエンテーションベクトルと、これに直行するオリエンテーション法線ベクトルによって特定されます。基本の向きは、セッティングデータの 1 および 10 の位によって定義されます。

表 19-5 フライス/特殊工具($TC_DP1 <> 400 … 599)

SD42954	工具オリエンテーションの座標系	
	配向ベクトル	オリエンテーション法線ベクトル
= xx17	(0, 0, 1)	(0, 1, 0)
= xx18	(0, 1, 0)	(1, 0, 0)
= xx19	(1, 0, 0)	(0, 0, 1)

表 19-6　旋削/研削工具($TC_DP1 = 400 … 599)

SD42956	工具オリエンテーションの座標系	
	配向ベクトル	オリエンテーション法線ベクトル
= xx17	(0, 0, 1)	(0, 1, 0)
= xx18	(0, 1, 0)	(1, 0, 0)
= xx19	(1, 0, 0)	(0, 0, 1)

記載された 3 つの値のいずれにも等しくない、0 以外の各値は、値「17」(フライス/特殊工具の場合)または「18」(旋削/研削工具の場合)として評価されます。

参照先

SD42954 および SD42956 の詳細説明については、『リストマニュアル、システム変数』を参照してください。

注記

向きが刃先データ($TC_DPV…)によって定義される工具の場合、通常、SD42954 または SD42956 は無視されます(「パラメータ設定可能な基本工具のオリエンテーション (ページ 1959)」の章を参照)。

19.12 パラメータ設定可能な基本工具のオリエンテーション

19.12.1 機能

注記

工具 T および刃先 D

以降では構文[...]は、基本工具のオリエンテーションのシステム変数に対する[<t> ,<d>]を表します。この場合、<t>は工具 T=<t>の番号、<d>は工具刃先 D=<d>の番号を示します。

"パラメータ設定可能な基本工具オリエンテーション"機能により、以下のシステム変数とともに各工具刃先に個別の初期オリエンテーションを割り当てることができます。

- 事前定義されたオリエンテーションベクトルの選択
  - $TC_DPV[...]:オリエンテーションベクトル(値 1 ... 6)
- オリエンテーションベクトルの定義
  - $TC_DPV3[...]:オリエンテーションベクトルの L1 成分
  - $TC_DPV4[...]:オリエンテーションベクトルの L2 成分
  - $TC_DPV5[...]:オリエンテーションベクトルの L3 成分
- オリエンテーションベクトルの法線ベクトルの定義
  - $TC_DPVN3[...]:法線ベクトルの L1 成分
  - $TC_DPVN4[...]:法線ベクトルの L2 成分
  - $TC_DPVN5[...]:法線ベクトルの L3 成分

システム変数$TC_DPV[...]が 0 に等しい場合、3 つのシステム変数$TC_DPV3 - 5[...]が基本工具のオリエンテーションの方向ベクトルを定義します。さらに、3 つのシステム変数$TC_DPVN3 - 5[...]では、オリエンテーションベクトルに垂直な平面での工具のオリエンテーションを定義できます。オリエンテーションベクトルの大きさは重要ではありません。

19.12.2 セットアップ

19.12.2.1 起動

"パラメータ設定可能な基本工具のオリエンテーション"は、以下のマシンデータによって起動されます:

MD18114 $MN_MM_ENABLE_TOOL_ORIENTATION = <値>

<値>	意味
0	機能"パラメータ設定可能な基本工具のオリエンテーション"は有効ではありません。
1	システム変数の起動: ● 機能選択: $TC_DPV[...] システム変数$TC_DPV[...] = 1、2、... 6 により、事前定義された 6 つの基本工具オリエンテーションのいずれかを、工具 T=<t>の各工具刃先 D=<d>に割り当てることができます。

<値>	意味
2	システム変数の起動: ● 機能選択: $TC_DPV[...] ● オリエンテーションベクトル: $TC_DPV3[...]、$TC_DPV4[...]、および $TC_DPV5[...] システム変数$TC_DPV[...] = 0、そして$TC_DPV3[...]、$TC_DPV4[...]、および$TC_DPV5[...]により、工具 T=<t>の各工具刃先 D=<d>にオリエンテーションベクトルを割り当てることができます。
3	システム変数の起動: ● 機能選択: $TC_DPV[...] ● オリエンテーションベクトル: $TC_DPV3[...]、$TC_DPV4[...]、および $TC_DPV5[...] ● オリエンテーションベクトルの法線ベクトル: $TC_DPVN3[...]、$TC_DPVN4[...]、および$TC_DPVN5[...] 設定$TC_DPV[...] = 0 により、システム変数$TC_DPV3[...]、$TC_DPV4[...]、および$TC_DPV5[...]を使用して、工具 T=<t>の各工具刃先 D=<d>にオリエンテーションベクトルを割り当てることができます。さらに、システム変数 $TC_DPVN3[...]、$TC_DPVN4[...]、および$TC_DPVN5[...]によりオリエンテーションベクトルの法線ベクトルを定義できます。

19.12.2.2 パラメータ設定

システム変数$TC_DPVx[...]の割り当て

システム変数$TC_DP1[...]の指定に従って以下の工具タイプが区別されます:

- $TC_DP1[...] == 400 ... 599 ⇒ 旋削/研削工具
- $TC_DP1[...] <> 400 ... 599 ⇒ フライス/特殊工具

$TC_DPVx[...]でプログラムされた基本工具オリエンテーションは、工具タイプに応じて座標軸に割り当てられます。

工具タイプは、(工具長成分に関する「工具タイプに依存しない割り当て (ページ 1956)」の章で説明されているように)セッティングデータ SD42950 $SC_TOOL_LENGTH_TYPE の設定によって切り替えることができます。

セッティングデータ SD42954 $SC_TOOL_ORI_CONST_M および SD42956 $SC_TOOL_ORI_CONST_T (ページ 1958)で他のオリエンテーションが定義されない場

合は、標準のケース(フライス/特殊工具の場合は G17、旋削/研削工具の場合は G18)がシステム変数$TC_DPVx[...]の割り当てに適用されます。

工具タイプ	TC_DPV3[…]	TC_DPV4[…]	TC_DPV5[…]
旋削/研削工具	X	Z	Y
フライス/特殊工具	Z	Y	X

注記

SD42954/SD42956

システム変数$TC_DPVx[...]の割り当てを変更できるのは、SD42954 $SC_TOOL_ORI_CONST_M および SD42956 $SC_TOOL_ORI_CONST_T で、1000 の位が「1」に等しい場合のみです。

セッティングデータ SD42950 $SC_TOOL_LENGTH_TYPE の値に関係なく、SD42954 は、パラメータ$TC_DP1[...]がフライス工具を定義する工具に対してのみ有効です。同様に、SD42956 は、パラメータ$TC_DP1[...]が旋削/研削工具を定義する工具に対してのみ有効です。

19.12.3 プログラミング

システム変数$TC_DPV3 - 5[...]または$TC_DPV[...]を使用して、工具刃先ごとに別々の基本の向きを割り当てることができます。

オプションの設定

基本的には、以下の設定オプションが利用できます。

- **$TC_DPV[...] == 0 および$TC_DPV3 - 5[...] == 0**
  基本工具オリエンテーションのベクトルは、有効な加工平面から得られます。
  - G17:Z 座標
  - G18:Y 座標
  - G19:X 座標

  「近軸工具オリエンテーション (ページ 1957)」も参照してください。
- **$TC_DPV[...] == 0 および$TC_DPV3 - 5[...] <> 0**
  基本工具オリエンテーションのベクトルは、$TC_DPV3 - 5[...]によって示されます。
  - `$TC_DPV3[...] = <L1 方向の値>`
  - `$TC_DPV4[...] = <L2 方向の値>`
  - `$TC_DPV5[...] = <L3 方向の値>`

  例:
  基本工具オリエンテーションは、L1-L3 平面の二分線の方向、すなわち、フライス工具および有効な平面 G17 の場合に、ZX 平面の二分線の方向を指します。

```
$TC_DPV [1,1] = 0
$TC_DPV3[1,1] = 1.0
$TC_DPV4[1,1] = 0.0
$TC_DPV5[1,1] = 1.0
```

- **$TC_DPV[...] == 1、2 ... 6**
  基本工具オリエンテーションのベクトルは、$TC_DPV[...]によって示されます。
  以下の表に、どの基本工具オリエンテーションが事前定義され、かつ$TC_DPV[...]によって選択できるかを示します。

事前定義されたオリエンテーションベクトルの選択

```
$TC_DPV[...] = <値>
```

表 19-7　旋削/研削工具($TC_DP1 = 400 … 599)

<値>	意味		
0	$TC_DPV3[...]	$TC_DPV5[...]	$TC_DPV4[...]
1	0	1	0
2	1	0	0
3	0	0	1

<値>	意味		
4	0	-1	0
5	-1	0	0
6	0	0	-1

表 19-8　フライス/特殊工具($TC_DP1 <> 400 … 599)

<値>	意味		
0	$TC_DPV5[...]	$TC_DPV4[...]	$TC_DPV3[...]
1	0	0	1
2	0	1	0
3	1	0	0
4	0	0	-1
5	0	-1	0
6	-1	0	0

例

旋削/研削工具:

```
$TC_DPV[...] = 2   対応      $TC_DPV3[...] = 1
                             $TC_DPV4[...] = 0
                             $TC_DPV5[...] = 0
```

フライス/特殊工具

```
$TC_DPV[...] = 3   対応      $TC_DPV3[...] = 0
                             $TC_DPV4[...] = 0
                             $TC_DPV5[...] = 1
```

19.12.4　例

基本工具の向きが X-Z 平面の二分線であるフライス工具が、長さ L1=10 で定義されます。

例 1

プログラムコード	コメント
$SC_TOOL_LENGTH_TYPE=2	; 旋削/研削工具として座標軸に割り当て。
$SC_TOOL_ORI_CONST_M=1019	; 作業平面 G19
N10 $TC_DP1[1,1]=120	; 工具タイプ:フライス工具
N20 $TC_DP3[1,1]=10	; 長補正ベクトル:L1=10
N30 $TC_DPV[1,1]=0	; $TC_DPV3-5 による基本工具オリエンテーション
N40 $TC_DPV3[1,1]=1	; Z 座標
N50 $TC_DPV4[1,1]=0	; Y 座標
N60 $TC_DPV5[1,1]=1	; X 座標
N70 TRAFOON(...)	; 座標変換を起動。
N80 G17 F1000 X0 Y0 Z0 T1 D1	; 工具およびアプローチ原点位置を起動します。
	; ブロック N80 での平面選択 G18 は、長さ補正にのみ関連します。
...	

例 2

プログラムコード	コメント
N10 $TC_DP1[1,1]=120	; 工具タイプ:フライス工具
N20 $TC_DP3[1,1]=10	; 長補正ベクトル:L1=10
N30 $TC_DPV[1,1]=0	; $TC_DPV3-5 による基本工具オリエンテーション
N40 $TC_DPV3[1,1]=1	; Z 座標
N50 $TC_DPV4[1,1]=0	; Y 座標
N60 $TC_DPV5[1,1]=1	; X 座標
N70 TRAFOON(...)	; 座標変換を起動。
N80 G17 F1000 X0 Y0 Z0 T1 D1	; 工具およびアプローチ原点位置を起動します。
	; 工具長 L1=10 により =>
	; 機械の位置:X=0、Y=0、Z=10
N90 MOVT=10	; 工具方向のインクリメンタル移動動作。
	; 得られる軸位置:X=7.071 Y=0 Z=17.071
N100 M30	

19.13 工具補正の特殊処理

19.13.1 関連するセッティングデータ

SD42900 ～ 42960

セッティングデータ SD42900 ～ SD42940 は、工具補正に関して以下の設定をおこなうために使用できます。

- 工具長の符号
- 磨耗の符号
- ジオメトリ軸のミラーリング時の磨耗成分の動作
- セッティングデータを介した加工平面変更時の磨耗成分の動作
- 実際の工具タイプに無関係な工具長成分の割り当て
- 有効な工具長を制御するための磨耗成分の適切な座標系への座標変換

注記

以下の記述で、磨耗には次の成分の合計値が含まれます。

- 摩耗値: $TC_DP12 ～$TC_DP20
- サムオフセット、以下から構成されます。
  - 摩耗値: $SCPX3 ～$SCPX11
  - セットアップ値: $ECPX3 ～$ECPX11

サムオフセットと工具オフセットに関する詳細情報については、以下を参照してください。

参照先:

『機能マニュアル、工具管理機能』

『プログラミングマニュアル、基本編』;「工具オフセット」

必要なセッティングデータ

- SD42900 $SC_MIRROR_TOOL_LENGTH (工具長成分と工具ベース寸法成分のミラーリング) (ページ 1967)
- SD42910 $SC_MIRROR_TOOL_WEAR (工具長成分の磨耗値のミラーリング) (ページ 1967)
- SD42920 $SC_WEAR_SIGN_CUTPOS (磨耗成分の符号評価) (ページ 1968)

- SD42930 $SC_WEAR_SIGN (磨耗寸法の符号の反転) (ページ 1968)
- SD42935 $SC_WEAR_TRANSFORM (磨耗値の座標変換) (ページ 1989)
- SD42940 $SC_TOOL_LENGTH_CONST (工具長成分のジオメトリ軸への割り当て) (ページ 1954)
- 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) (ページ 1956)
- SD42960 $SC_TOOL_TEMP_COMP (工具長オフセット) (ページ 1970)

19.13.2 工具長のミラーリング

起動

工具長ミラーリングは、以下のセッティングデータを介して起動されます。

SD42900 $SC_MIRROR_TOOL_LENGTH <> 0 (TRUE) (ミラーリング時の工具長符号変更)

機能

以下のコンポーネントは、符号を反転することによりミラーリングされます。

- 工具長: $TC_DP3、$TC_DP4、$TC_DP5
- 工具ベース寸法: $TC_DP21、$TC_DP22、$TC_DP23

ミラーリングは、関連した軸がミラーリングされるすべての工具ベース寸法に対して実行されます。 磨耗値は、ミラーリング**されません**。

磨耗値のミラーリング

磨耗値をミラーリングするには、以下のセッティングデータを設定してください。

SD42910 $SC_MIRROR_TOOL_WEAR <> 0 (ミラーリング時の工具磨耗符号変更)

符号を反転すると、関連した軸がミラーリングされる工具長コンポーネントの磨耗値がミラーリングされます。

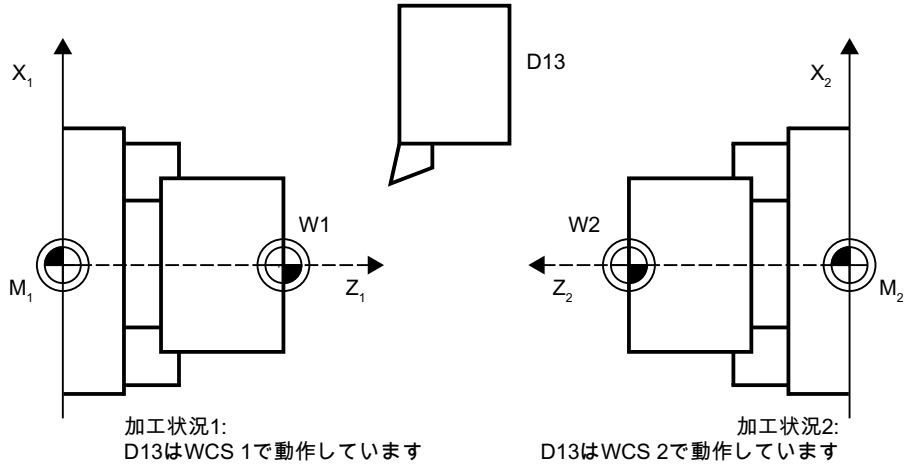


図 19-56　適用例: 二主軸旋盤

19.13.3　摩耗長のミラーリング

起動

磨耗長は、以下により起動されます。

SD42920 $SC_WEAR_SIGN_CUTPOS <> 0 (TRUE) (刃先位置のある工具の磨耗符号)

機能

刃先位置	長さ 1	長さ 2
1	---	---
2	---	反転
3	反転	反転
4	反転	---
5	---	---
6	---	---
7	---	反転
8	反転	---
9	---	---

対応する刃先位置のない工具タイプの場合、磨耗長はミラーリングされません。

注記

1 つ以上のコンポーネントでのミラーリング(符号反転)は、下記の機能の同時起動によりそれ自身キャンセルできます。

工具長ミラーリング(SD42900 <> 0)

および:

磨耗長ミラーリング(SD42920 <> 0)

SD42930 $SC_WEAR_SIGN

ゼロ以外のセッティングデータ:

すべての磨耗寸法の符号を反転します。 これは、工具長と工具半径、ラウンディング半径などの他の変数の両方に影響を与えます。

正の磨耗寸法を入力すると、工具を「より短く」かつ「より薄く」します。

修正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修正すると、次に工具刃先が選択されるまで、工具コンポーネントは再計算されません。 工具がすでに有効であり、この工具のデータを再計算する必要がある場合は、この工具を再度選択してください。

例:

```
N10 $SC_WEAR_SIGN = 0              ; 磨耗値の符号反転なし
N20 $TC_DP1[1,1] = 120             ; エンドミル
N30 $TC_DP6[1,1] = 100             ; 工具半径 100 mm
N40 $TC_DP15[1,1] = 1              ; 工具半径の磨耗寸法 1 mm、結果として生じる工具半径 101
                                     mm

N100 T1 D1 G41 X150 Y20
....
N150 G40 X300N10
....
N200 $SC_WEAR_SIGN = 1             ; すべての磨耗値の符号反転。99 mm の新しい半径が、新規選
                                     択 (D1) 時に起動されます。 D1 がない場合、半径は 101 mm
                                     のままになります。

N300 D1 G41 X350 Y-20
N310 ....
```

結果として生じる工具長が軸のミラーリング状態の変更により修正される場合に、同じことが当てはまります。 ミラーリング命令の後は、修正された工具長コンポーネントを起動するために、工具を再度選択してください。

19.13.4 向きを考慮した WCS の工具長

工具または作業平面の変更

工具に対して表示される値は、WCS での拡張に対応します。 傾斜したクランプ位置で工具ホルダを使用する場合、使用される座標変換が工具ホルダをサポートしていることを確認してください。 これが該当しない場合、間違った工具寸法が表示されます。 G17 から G18 または G19 に作業平面を変更するときは、これらの作業平面に対して座標変換も使用できることを確認してください。 座標変換が G17 加工に対してのみ使用できる場合、平面の変更後は Z 方向の工具に対して寸法が表示され続けます。

座標変換が解除されると、作業平面に従って、x、y または z 方向で基本工具が表示されます。 プログラミングされた工具ホルダが考慮されます。 これらの工具寸法は、座標変換なしで移動するときは変更されません。

19.13.5 工具方向の工具長オフセット

リアルタイムの熱変異補正

移動工具のある 5 軸の機械では、加工ヘッドで温度変動が生じる可能性があります。これらは膨張によるずれの直接的な原因になる可能性があり、長さの膨張の形で工具主軸に伝わります。たとえば、5 軸ヘッドの典型的なケースは、主軸の長手軸方向での熱膨張です。

熱変異補正値を機械軸ではなく工具に割り当てることにより工具の向きを調整する場合でも、この熱膨張を補正することができます。このように、長さ膨張のずれは工具の向きの変更時でも補正できます。

現在の工具の向きにより方向が決定される方向座標変換を使用して、リアルタイムに動作を重畳させ、同時に回転させることができます。同時に、補正値は工具座標系で連続的に調整されます。

熱変位補正は、補正対象の軸が実際に原点確立済みの場合にだけ有効です。

起動

工具方向での熱変位補正は、以下のマシンデータを **0 以外の**値に設定して起動します。

MD20390 $MC_TOOL_TEMP_COMP_ON (工具長の熱変異補正の起動)

また、対応するチャネル軸毎に、次のマシンデータでビット 2 を設定してください。

MD32750 $MA_TEMP_COMP_TYPE [<軸インデックス>] (熱変異補正タイプ)

座標変換切り替えのジオメトリ軸置換の結果として、続けて 4 つ以上のチャネル軸を一時的にジオメトリ軸に割り当てることがある場合、これは 4 つ以上の軸になることがあります。このビットを特定のチャネル軸に対して設定しない場合、この補正値を軸に適用できません。これは、他の軸へ影響しません。この場合、アラームは出力されません。

適用範囲

工具方向での熱変異補正は、以下のタイプの総合 5 軸座標変換でのみ作用します。

- 座標変換タイプ 24
  2 つの軸が工具を回転します
- 座標変換タイプ 56
  1 つの軸が工具を回転させ、他方の軸が熱変異補正のなしでワークを回転させます。

以下のタイプの総合 5 軸座標変換:

- 座標変換タイプ 40
  工具の向きは、回転ワークにより一定です。つまり、機械の回転軸の動作は熱変異補正方向に影響を与えません。

工具方向の熱変異補正は方向座標変換(総合 5 軸座標変換ではない)と組み合わせても作用します。

- 座標変換タイプ 64 ～ 69
  回転する直線軸

注記

熱変異補正は、他のすべてのタイプの座標変換とともに起動できます。これは、工具の向きの変更の影響を受けません。軸は、熱変異補正による方向座標変換が無効であるかのように移動します。

制限値

補正値は、以下のマシンデータによる最大値に制限されます。

MD20392 $MC_TOOL_TEMP_COMP_LIMIT[0...2] (工具長の最大熱変異補正)

制限値の初期設定は 1 mm です。この制限より高い熱変異補正値が指定された場合、これはアラームなしで制限されます。

SD42960

3 つの熱変異補正値は、ともに補正ベクトルを形成し、以下のセッティングデータに含まれます。

SD42960 $SC_TOOL_TEMP_COMP[0...2] (工具に関する熱変異補正)

セッティングデータは、たとえば PLC からまたはシンクロナイズドアクションを使用してユーザーが定義します。したがって、補正値は他の補正のためにも使用できます。

方向座標変換が解除されるとき、または初期状態では、3 つの補正値すべてが 3 つのジオメトリ軸の方向に適用されます(通常の順序 X、Y、Z で)。ジオメトリ軸への補正の割り当ては、工具タイプ(旋削工具、フライス加工または研削工具)および選択された加工平面 G17 ～ G19 とは無関係です。セッティングデータ値への変更は、直ちに有効になります。

旋回工具ホルダ

旋回工具ホルダが有効な場合、熱変異補正ベクトルは向きの変更と同時に回転します。これは、有効な方向座標変換とは無関係に適用されます。

旋回工具ホルダが、総合 5 軸座標変換または直線軸の回転による座標変換と組み合わせて有効である場合は、熱変異補正ベクトルは両方の回転に従います。

注記

直線軸の回転による座標変換により工具ベクトル(長さ)の変化が考慮される一方、旋回工具ホルダにより影響を受ける可能性のある、その向きの変化は**無視され**ます。

熱変異補正値は、直ちに適用される向きの変更に従います。これは特に、方向座標変換が起動または解除されるときに適用されます。

同じことが、ジオメトリ軸とチャネル軸の間で割り当てが変更されるときに当てはまります。たとえば、座標変換の変更の後でジオメトリ軸でなくなるときに、軸の熱変異補正値がゼロまで減らされます(補間的に)。です。反対に、変更がジオメトリ軸状態に移行する軸の熱変異補正値は、直ちに適用されます。

例

工具方向の熱変異補正

工具を C と B 軸を中心にして回転できる、旋回工具のある 5 軸の機械の例です。

初期状態では、工具は Z 軸に平行です。B 軸が 90°回転される場合、工具は X 方向を指します。

したがって、座標変換が有効な場合、以下のセッティングデータの熱変異補正値は、機械の X 軸の方向にも有効です。

SD42960 $SC_TOOL_TEMP_COMP[2] (工具に関する熱変異補正)

この状態で座標変換が解除された場合、工具の向きは定義により再度 Z 軸に平行になるので、実際の向きとは異なります。X 軸方向の温度オフセットはゼロまで減らされ、同時に Z 方向に再度適用されます。

旋回工具のある 5 軸の機械の例(座標変換タイプ 24)です。関連するマシンデータが以下に示されています。

- 1 番目の回転軸は、Z を中心に回転する C 軸です。
- 2 番目の回転軸は、Y を中心に回転する B 軸です。

下表に、基本的なマシンデータが示されています。

マシンデータ	値	備考
MD20390 $MC_TOOL_TEMP_COMP_ON	= TRUE	熱変異補正有効
MD32750 $MA_TEMP_COMP_TYPE[AX1]	= 4	工具方向の補正
MD32750 $MA_TEMP_COMP_TYPE[AX2]	= 4	工具方向の補正
MD32750 $MA_TEMP_COMP_TYPE[AX3]	= 4	工具方向の補正
		座標変換タイプ 24 の割り当て
MD24100 $MC_TRAFO_TYPE_1	= 24	1 番目のチャネルの変換タイプ 24
MD24110 $MC_TRAFO_AXES_IN_1[0]	= 1	座標変換の最初の軸
MD24110 $MC_TRAFO_AXES_IN_1[1]	= 2	座標変換の 2 番目の軸
MD24110 $MC_TRAFO_AXES_IN_1[2]	= 3	座標変換の 3 番目の軸
MD24110 $MC_TRAFO_AXES_IN_1[3]	= 5	座標変換の 5 番目の軸
MD24110 $MC_TRAFO_AXES_IN_1[4]	= 4	座標変換の 4 番目の軸

マシンデータ	値	備考
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[0]	= 1	チャネル軸 1 のジオメトリ軸
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[1]	= 2	チャネル軸 2 のジオメトリ軸
MD24120 $MC_TRAFO_GEOAX_ASSIGN_TAB_1[2]	= 3	チャネル軸 3 のジオメトリ軸
MD24570 $MC_TRAFO5_AXIS1_1[0]	= 0.0	
MD24570 $MC_TRAFO5_AXIS1_1[1]	= 0.0	方向
MD24570 $MC_TRAFO5_AXIS1_1[2]	= 1.0	1 番目の回転軸は Z に平行です
MD24572 $MC_TRAFO5_AXIS1_2[0]	= 0.0	方向
MD24572 $MC_TRAFO5_AXIS1_2[1]	= 1.0	2 番目の回転軸は Y に平行です
MD24572 $MC_TRAFO5_AXIS1_2[2]	= 0.0	
MD25574 $MC_TRAFO5_BASE_ORIENT_1[0]	= 0.0	
MD25574 $MC_TRAFO5_BASE_ORIENT_1[1]	= 0.0	基本工具の向き
MD25574 $MC_TRAFO5_BASE_ORIENT_1[2]	= 1.0	Z 方向

NC プログラムでの熱変異補正値

軸 X と Z に割り当てられる補正値はゼロではなく、工具長に関して熱変異補正に適用されます。各ケースで到達する機械軸位置は、プログラム行のコメントとして指定されます。

プログラムコード	コメント
`$SC_TOOL_TEMP_COMP[0] = -0.3`	`; X 方向の補正値`
`$SC_TOOL_TEMP_COMP[1] = 0.0`	
`$SC_TOOL_TEMP_COMP[2] = -1.0`	`; Z 方向の補正値`
	`; 機械軸の位置指令値`
`N10 G74 X0 Y0 Z0 A0 B0`	`; X Y Z`
`N20 X20 Y20 Z20 F10000`	`; 20.30 20.00 21.00`
`N30 traori()`	`; 20.30 20.00 21.00`

プログラムコード	コメント
N40 X10 Y10 Z10 B90	; 11.00 10.00 9.70
N50 trafoof	; 10.30 10.00 11.00
N60 X0 Y0 Z0 B0 C0	; 0.30 0.00 1.00
N70 M30	

ブロック N40 を除いて、工具は基本の向きの方向を指しているので、熱変異補正は常に元の方向で作用します。これは、特にブロック N50 に当てはまります。B 軸はまだ 90° にあるため、工具は実際にまだ X 軸の方向を指しています。ただし、座標変換はすでに解除されたため、適用される向きは再度 Z 軸に平行になりす。

マシンデータ	値	備考
MD20390 $MC_TOOL_TEMP_COMP_ON	= TRUE	熱変異補正有効
MD32750 $MA_TEMP_COMP_TYPE[AX1]	= 4	工具方向の補正
MD32750 $MA_TEMP_COMP_TYPE[AX2]	= 4	工具方向の補正
MD32750 $MA_TEMP_COMP_TYPE[AX3]	= 4	工具方向の補正

関連参照先

「熱変異補正」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、上級機能』;「誤差補正(K3)」

「総合 5 軸座標変換」の情報については、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』;「3 ～ 5 軸座標変換(F2)」

19.13.6 旋回工具ホルダの特徴

セッティングデータ SD42900 ～ D42950

セッティングデータ SD42900 ～ SD42950 は、有効な旋回工具ホルダの成分には影響を与えません。旋回工具ホルダによる計算では、常に合計して得られた長さ(工具長 + 摩

耗 + 工具ベース寸法)のある工具が考慮されます。長さの合計の計算では、セッティングデータによるすべての修正が考慮されます。

注記

旋回工具ホルダを使用する場合、一般的に、すべての工具はミラーリングされない基本単位系に対して定義されます。それらが、ミラーリングされる加工にのみ使用される場合でも定義されます。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。

機械タイプの物理特性により、さまざまな向きで固定されている工具の回転が妨げられる場合も、旋回工具ホルダを使用することが実用的です。工具のサイズ決めは、基本の向きで一様に行うことができます。この場合、加工に関連する寸法は実際の工具ホルダの回転に応じて計算されます。

19.14 サムオフセットとセットアップオフセット

19.14.1 一般情報

サムオフセット

サムオフセットは、加工中に**プログラム可能な処理補正**として扱うことができ、(摩耗などの)すべての誤差量で構成されます。誤差量とは、指定された寸法からワークが逸脱する原因になるものです。

サムオフセットは、汎用タイプの摩耗です。サムオフセットは、刃先データの一部です。サムオフセットの各変数は、刃先のジオメトリデータに該当します。

サムオフセットの補正データは、**DL 番号**に対応します(**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)。

これとは対照的に、D 番号の摩耗値は、刃先の物理的摩耗を記述します。つまり、特殊な状況では、サムオフセットが刃先の摩耗と一致することがあります。

サムオフセットは、汎用的な使用を意図しています。つまり、工具管理機能が有効または無効な場合、またはフラット D 番号機能を使用する場合を対象とします。

マシンデータにより、サムオフセットは次の 2 つに分類されます。

- 仕上げサムオフセット
- 荒削りサムオフセット(セットアップオフセット)

セットアップオフセット

セットアップオフセットは、加工をする前に、セットアップエンジニアにより入力される補正を指します。これらの値は個別に NC に格納されます。オペレータは、これ以降、HMI を介して「精密サムオフセット」にのみアクセスできます。

NC では、「精密サムオフセット」と「荒削りサムオフセット」が内部的に加算されます。この値は、以降、サムオフセットと呼ばれます。

注記

次のマシンデータ設定を使用すると、この機能が有効になります。

MD18080 $MN_MM_TOOL_MANAGEMENT_MASK, Bit 8=1 (工具管理機能の段階的メモリ確保)

キネマティックトランスフォーメーション(たとえば、5 軸座標変換)が有効になっている場合、まず、各種摩耗コンポーネントを考慮した後、工具長が計算されます。その後、座標変換で合計工具長が使用されます。このため、旋回工具ホルダの場合と異なり、グループ 56 の G 命令に関係なく、摩耗値は必ず座標変換に組み込まれます。

19.14.2 機能説明

サムオフセット

D 番号ごとに、サムオフセット(DL 番号)を複数定義できます。たとえばこの方法で、**ワークの位置依存**のオフセット値を決定し、刃先に割り当てることができます。サムオフセットには、摩耗オフセットと同じ効果があります。つまり、サムオフセットは、D 番号のオフセット値に追加されます。このデータは常に D 番号に割り当てられます。

設定

マシンデータで、次の設定を定義できます。

- サムオフセットの起動
- NC で作成される DL データセットの最大数を定義

- D 番号に割り当てられる DL 番号の最大数を定義
- データのバックアップ中に(仕上げ/荒削り)サムオフセットを保存するかどうかを定義
- 次の場合、起動するサムオフセットを定義
  - 刃先補正を新たに起動する場合
  - 操作パネル RESET が実行される場合
  - 操作パネル START が実行される場合
  - プログラムの終わりに到達した場合

名前は、工具および刃先に対応するマシンデータの論理に適応します。

システム変数および対応する OPI サービスを使用すれば、「セットアップオフセット」と「精密サムオフセット」の読み書きができます。

注記

工具管理機能が有効になっている場合、マシンデータ項目を使用すれば、プログラムした工具交換時に起動された工具のサムオフセットをそのまま変更しないか、ゼロに設定するかを定義できます。

補正変数$TC_DPx の要約

次の一般システム変数は、刃先を記述するために以前に定義されました。

$TC_DP1	工具タイプ
$TC_DP2	刃先位置

形状と摩耗の変数

工具形状補正は、システム変数$TC_DP3 ～$TC_DP11 に割り当てられます。システム変数$TC_DP12 ～$TC_DP20 を使用すれば、これらのパラメータごとに摩耗を指定できます。

ジオメトリ	摩耗	補正
長補正		
$TC_DP3	$TC_DP12	長さ 1
$TC_DP4	$TC_DP13	長さ 2
$TC_DP5	$TC_DP14	長さ 3
径補正		

ジオメトリ	摩耗	補正
$TC_DP6	$TC_DP15	半径 1
$TC_DP7	$TC_DP16	半径 2
その他の補正		
$TC_DP8	$TC_DP17	長さ 4
$TC_DP9	$TC_DP18	長さ 5
$TC_DP10	$TC_DP19	角度 1
$TC_DP11	$TC_DP20	角度 2

工具ベース寸法またはアダプタ寸法

$TC_DP21	アダプタ長さ 1
$TC_DP22	アダプタ長さ 2
$TC_DP23	アダプタ長さ 3

加工

システム変数	逃げ角
$TC_DP24	ManualTurn、工具タイプ 5xx では、ここに逃げ角が格納されます。旋削工具の標準サイクルと同じ意味です。 ShopMill、工具タイプ 2xx では、ここにドリルのチップ角度が格納されます。 旋削工具、工具タイプ 5xx では、標準サイクルに使用されます。これは、これらの工具の 2 次刃先の角度です。
$TC_DP25	ManualTurn では、ここに切削速度の値が格納されます。 ShopMill では、ここに、工具タイプ 1xx および 2xx の各種状態に対応するビットコード値が格納されます。

サムオフセットとセットアップオフセットの変数($TC_SCPxy、$TC_ECPxy)

変数の番号付けは、システム変数$TC_DP3 ～$TC_DP11 の番号付けに適応します。

各変数の効果は、(工具ジオメトリに追加される)摩耗と同様です。刃先変数ごとに、最大 6 つまでのサム/セットアップ変数を定義できます。

工具形状パラメータ (補正が追加される)	サム/セットアップパラメータ	工具摩耗パラメータ
長補正		
$TC_DP3	長さ 1 $TC_SCP**1**3, $TC_SCP**2**3,$TC_SCP**3**3, $TC_SCP**4**3,$TC_SCP**5**3,$TC_SCP**6**3 $TC_ECP**1**3, $TC_ECP**2**3,$TC_ECP**3**3, $TC_ECP**4**3,$TC_ECP**5**3,$TC_ECP**6**3 太字の各番号 1、2、... 6 は、列 1 で指定された変数の DL =1 ～ 6 でプログラムできる最大6つの(位置に関係、など)補正の各変数を指定します。	$TC_DP12
$TC_DP4	長さ 2 $TC_SCP**1**4, $TC_SCP**2**4,$TC_SCP**3**4, $TC_SCP**4**4,$TC_SCP**5**4,$TC_SCP**6**4 $TC_ECP**1**4, $TC_ECP**2**4,$TC_ECP**3**4, $TC_ECP**4**4,$TC_ECP**5**4,$TC_ECP**6**4	$TC_DP13
$TC_DP5	長さ 3 その他 ...	$TC_DP14
径補正		
$TC_DP6	半径	$TC_DP15
$TC_DP7	コーナ半径	$TC_DP16
その他の補正		
$TC_DP8	長さ 4	$TC_DP17
$TC_DP9	長さ 5	$TC_DP18

工具形状パラメータ(補正が追加される)	サム/セットアップパラメータ	工具摩耗パラメータ
$TC_DP10	角度 1 ...その他	$TC_DP19
$TC_DP11	角度 2 $TC_SCP**2**1, $TC_SCP**3**1,$TC_SCP**4**1, $TC_SCP**5**1,$TC_SCP**6**1,$TC_SCP**7**1 $TC_ECP**2**1, $TC_ECP**3**1,$TC_ECP**4**1, $TC_ECP**5**1,$TC_ECP**6**1,$TC_ECP**7**1 太字の各番号 2、3、... 7 は、列 1 で指定された変数の DL =1 ～ 6 でプログラムできる最大6つの(位置に関係、など)補正の各変数を指定します。	$TC_DP20

必要条件

刃先の最大 DL データセット数、および NC 内のサムオフセット数の合計数は、マシンデータで定義されます。初期値はゼロです。つまり、サムオフセットはプログラムできません。

「監視機能」を有効にして、摩耗と「サムオフセット」について工具を監視します。

サム/セットアップデータセットの追加には、追加の保持メモリを使用します。変数ごとに 8 バイト必要です。

サムオフセットデータセットでは、次のバイト数が必要になります。8 バイト* 9 変数= 72 バイト

セットアップデータセットでは、等しい量のメモリが必要になります。内部管理データにも、所定のバイト数が必要です。

19.14.3 起動

機能

この機能は、次のマシンデータを使用して起動してください。

MD18108 $MN_MM_NUM_SUMCORR (TO 領域内のサムオフセット)

OPI インタフェースを介して定義された$TC_ECPx と$TC_SCPx の各システム変数、およびセットアップオフセットとサムオフセット (「仕上げ」)は、パートプログラムで起動できます。

そのために、言語命令 DL="番号"をプログラムします。

D 番号を新たに起動する場合、DL 番号を新たにプログラムするか、次のマシンデータを使用して定義した DL 番号が有効になります

MD20272 $MC_SUMCORR_DEFAULT (プログラム指令されないときのサムオフセットの初期設定)

DL プログラミング

サムオフセットは必ず、次の命令を使用して、有効な D 番号に関連してプログラムされます。

```
DL = "n"
```

サムオフセット「n」は、有効な D 番号の摩耗に追加されます。

注記

「セットアップオフセット」**および**「サムオフセット(仕上げ)」を使用する場合、両方の補正が組み合わされた後、工具摩耗に追加されます。

次の命令を使用すると、サムオフセットが解除されます。

```
DL=0
```

注記

DL0 は使用できません。補正が解除される(D0 および T0)と、サムオフセットも無効になります。

存在しないサムオフセットをプログラムすると、存在していない D 補正をプログラムするのと同様、アラームが発生します。

従って、定義した摩耗だけが、補正の一部のままになります(システム変数$TC_DP12 ～ $TC_DP20 で定義された)。

D 補正が有効な場合にサムオフセットをプログラムするのは(解除にも適用されます)、D 命令をプログラムするのと同じ影響を軌跡に及ぼします。このため、たとえば有効な径補正は、その隣接ブロックへの関係を失います。

設定

MD18112 $MN_MM_KIND_OF_SUMCORR、ビット 4=0:(TO 領域内のサムオフセットのプロパティ)初期設定:

DL 番号ごとに存在するサムオフセットは **1** セットだけです。

通常、サムオフセットを指します。

これは、$TC_SCPx で表されるデータを意味します。

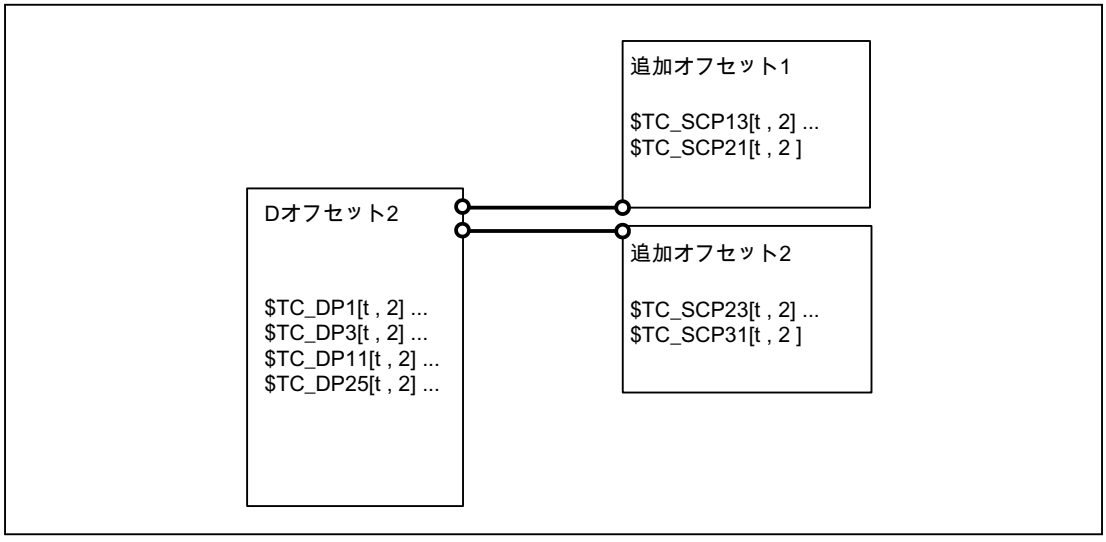


図 19-57　MD18112 $MN_MM_KIND_OF_SUMCORR、ビット 4 = 0

工具、T = t が有効です。図内のデータを使用して、次のプログラムが実行されます。

D2	;	刃先補正、つまり$TC_DP3 ～$TC_DP11 + 摩耗($TC_DP12 ～ $TC_DP20) + アダプタ寸法
...		
DL=1	;	サムオフセット 1 つまり$TC_SCP13 ～$TC_SCP21 は、直前の D2 補正に追加されます。
...		
DL=2	;	サムオフセット 2 つまり$TC_SCP23 ～$TC_SCP31 は、サムオフセット 1 に代わり、D2 補正に追加されます。
...		
DL=0	;	サムオフセットの解除、 D2 のデータだけが有効なままになります。

MD18112 $MN_MM_KIND_OF_SUMCORR、ビット 4=1:セットアップオフセットを使用できます

サムオフセットは、($TC_SCPx で表される)「サムオフセット(仕上げ)」 と($TC_ECPx で表される)セットアップオフセットで構成されるようになりました。このため、1 つの DL 番号に、2 つのデータセットがあります。対応するコンポーネントを加算して ($TC_ECPx + $TC_SCPx)、サムオフセットが計算されます。

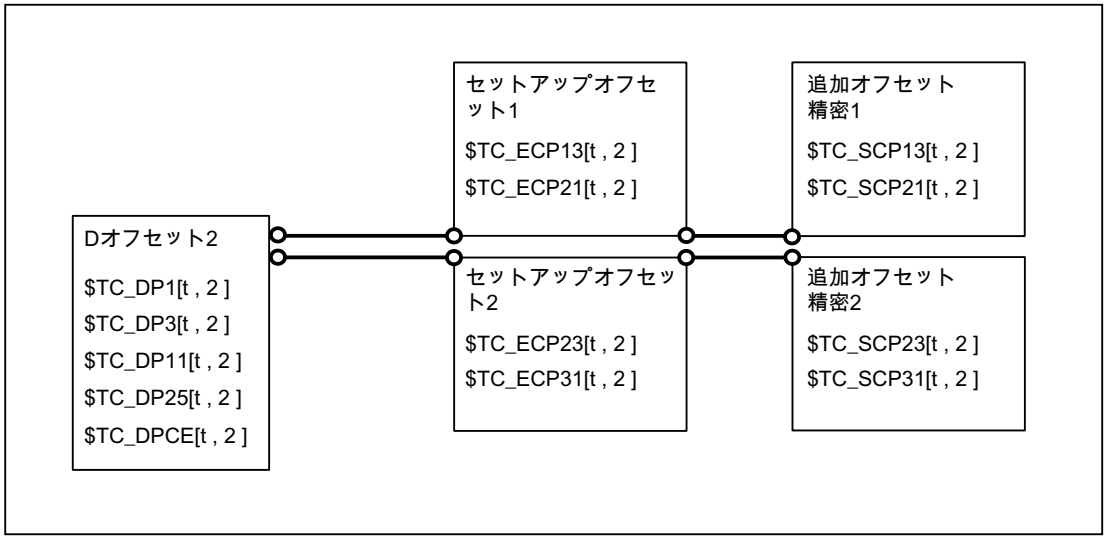


図 19-58 MD18112 $MN_MM_KIND_OF_SUMCORR、ビット 4 = 1 「セットアップオフセット」 + 「サムオフセット(仕上げ)」

工具、T = t が有効です。図内のデータを使用して、次のプログラムが実行されます。

D2	;	刃先補正、つまり$TC_DP3 ～$TC_DP11 + 摩耗($TC_DP12 ～ $TC_DP20) + アダプタ寸法
...		
DL=1	;	サムオフセット 1 つまり$TC_ECP13 + $TC_SCP13 ～ $TC_ECP21 + $TC_SCP21 は、直前の D2 補正に追加されます。
...		
DL=2	;	サムオフセット 2 つまり$TC_ECP23 + $TC_SCP23 ～ $TC_ECP31 + $TC_SCP31 は、サムオフセット 1 に代わり、D2 補正に追加されます。
...		
DL=0	;	サムオフセットの解除。D2 のデータだけが有効なままになります。

パートプログラムでの読み書き

サムオフセット変数の個々のセットは、システム変数$TC_SCP の番号範囲に応じて区別されます。

個々の変数の意味は、ジオメトリ変数$TC_DP3 ～$TC_DP11 と同様です。基本機能では、長さ 1、長さ 2、長さ 3 だけが有効です(刃先の最初のサムオフセットでは$TC_SCP13 ～$TC_SCP15 の各変数)。

R5 = $TC_SCP13[t, d]	;	R 変数の値を、刃先(d)におけるサムオフセット 1 の最初のコンポーネントの値に設定します。 工具(t)において。
R6 = $TC_SCP21[t, d]	;	R 変数の値を、工具(t)上の刃先(d)におけるサムオフセット 1 の最後のコンポーネントの値に設定します。
R50 = $TC_SCP23[t, d]	;	R 変数の値を、工具(t)上の刃先(d)におけるサムオフセット 2 の最初のコンポーネントの値に設定します。
$TC_SCP43[t, d] = 1.234	;	工具(t)上の刃先(d)におけるサムオフセット 4 の最初のコンポーネントの値を、1.234 に設定します。

上記命令はセットアップオフセットにも適用されます(そのオプションを使用して NCK が設定されている場合)。つまり、

R5 = $TC_ECP13[t, d]	;	R 変数の値を、工具(t)上の刃先(d)におけるセットアップオフセット 1 の最初のコンポーネントの値に設定します。
R6 = $TC_ECP21[t, d]	;	R 変数の値を、工具(t)上の刃先(d)におけるセットアップオフセット 1 の最後のコンポーネントの値に設定します。
その他		

セットアップオフセットと一緒に使用する場合、$TC_SCPx システム変数を使用して「サムオフセット(仕上げ)」を書き込みます。

サムオフセットの新規作成

補正データセット(x)が存在しない場合、その変数(y)の 1 つの最初の書き込み操作により作成されます。

$TC_SCPxy[t, d] = r.r	;	サムオフセット x の変数 y に値「r.r.」が割り当てられます。x の他の変数には、値ゼロが設定されます。

セットアップオフセットと一緒に使用する場合、$TC_SCPx システム変数を使用して「サムオフセット(仕上げ)」を書き込みます。

注記

セットアップオフセットを使用する場合、[t, d]にデータセットがまだ存在していなければ、「サムオフセット(仕上げ)」に対してデータセットを作成したとき、セットアップオフセットのデータセットが作成されます。

セットアップオフセットの新規作成

補正データセット(x)が存在しない場合、その変数(y)の 1 つの最初の書き込み操作により作成されます。

$TC_ECPxy[t, d] = r.r	;	セットアップオフセット x の変数 y に値「r.r.」が割り当てられます。x の他の変数には、値ゼロが設定されます。

注記

セットアップオフセットを使用する場合、[t, d]にデータセットがまだ存在していなければ、セットアップオフセットのデータセットを作成すると、「サムオフセット(仕上げ)」のデータセットが作成されます。

DELDL - サムオフセットの削除

サムオフセットは、通常、ワークの特定の位置の、特定の時間の、ある刃先で加工する場合だけ関係します。NC 言語命令 DELDL を使用すれば、メモリを解放するために、刃先からサムオフセットを削除できます。

status = DELDL(t, d)	;	工具 t 上の刃先 d に対するサムオフセットをすべて削除します。
	;	t、d は任意選択パラメータです。

d を指定していない場合、工具 t のすべての刃先のすべてのサムオフセットが削除されます。

d と t を指定していない場合、TO ユニットのすべての工具の刃先に対応するすべてのサムオフセットが削除されます(命令がプログラムされているチャネルで)。

セットアップオフセットを使用する場合、DELDL 命令を使用すると、指定の刃先のセットアップオフセットと「サムオフセット(仕上げ)」が両方とも削除されます。

注記

データセットに使用されるメモリは、削除の後に解放されます。
削除したサムオフセットは、続いて、起動したり、プログラムすることはできません。

有効な工具のサムオフセットとセットアップオフセットは削除できません (D 補正または工具データの削除と同様)。

「status」戻り値は、削除命令の結果を示します。

0:	削除は成功しました
-1:	削除に失敗しました(1 つの刃先)または削除の一部が失敗しました(複数の刃先)

データバックアップ

通常の工具データをバックアップ時に、データが格納されます(D 番号データセットのコンポーネントとして)。

重大問題の発生時に現在の状態を復元できるように、サムオフセットを保存しておくことをお勧めします。マシンデータを設定すれば、データのバックアップからサムオフセ

ットを除外できます(各設定は、「セットアップオフセット」と「サムオフセット(仕上げ)」で個別に実行できます)。

注記

サムオフセットは、ブロックサーチおよびREPOSに関するD補正と同じように動作します。リセットおよび電源投入に関する挙動は、マシンデータにより定義できます。

次のマシンデータの設定が、電源オン後に、最後に有効であった工具補正番号(D)が起動されることを示している場合、最後に有効であったDL番号は有効ではなくなります。

MD20110 $MC_RESET_MODE_MASK (リセット/パートプログラム終了後のコントローラの初期設定の定義)

19.14.4 例

例 1

工具交換中、次のマシンデータを使用して、補正もサムオフセットも有効にならないと定義します。

- MD20270 $MC_CUTTING_EDGE_DEFAULT=0 (プログラムされないときの工具刃先の初期設定)
- MD20272 $MC_SUMCORR_DEFAULT=0 (プログラムされないときの初期設定サムオフセット)

```
T5 M06              ; 工具番号 5 がロードされます- 補正は無効です。
D1 DL=3             ; 補正 D1 + D1 のサムオフセット 3 が起動されます。
X10
DL=2                ; 補正 D1 + D1 のサムオフセット 2 が起動されます。
X20
DL=0                ; サムオフセットの解除、これで補正 D1 だけが有効になります。
D2                  ; 補正 D2 が起動されます-補正内にサムオフセットは含まれていません。
X1
DL=1                ; 補正 D2 + サムオフセット 1 が起動されます。
X2
D0                  ; 補正の解除
X3
DL=2                ; 影響なし-D0 の DL2 はゼロです(T0 D2 のプログラムと同じです)。
```

例 2

工具の交換中、次のマシンデータを使用して、オフセット D2 とサムオフセット DL=1 が起動されるように定義します。

MD20270 $MC_CUTTING_EDGE_DEFAULT=2 (プログラムされないときの工具刃先の初期設定)

MD20272 $MC_SUMCORR_DEFAULT=1 (プログラムされないときの初期設定サムオフセット)

```
T5 M06              ; 工具番号 5 がロードされます- D2 + DL=1 が有効になります (= マシンデータの値)
D1 DL=3             ; 補正 D1 + D1 のサムオフセット 3 が起動されます。
X10
DL=2                ; 補正 D1 + サムオフセット 2 が起動されます。
X20
DL=0                ; サムオフセットの解除、これで補正 D1 だけが有効になります。
D2                  ; 補正 D2 が起動されます-サムオフセット DL=1 が起動されます。
X1
DL=2                ; 補正 D2 + サムオフセット 2 が起動されます。
D1                  ; 補正 D1 + サムオフセット 1 が起動されます。
```

19.14.5 工具長定義の改良

19.14.5.1 位置別およびワーク別の補正値の計算

有効な工具長の構成

キネマティック座標変換を起動しない工具補正の場合、有効な工具長は、最大で次の 8 つのベクトルで構成されます。

- 工具長(ジオメトリ) ($TC_DP3 - $TC_DP5)
- 摩耗 ($TC_DP12 - $TC_DP14)
- 工具ベース寸法(注を参照してください) ($TC_DP21 - $TC_DP23)
- アダプタ寸法(注を参照してください) ($TC_ADPT1 - $TC_ADPT3)
- サムオフセット(仕上げ) ($TC_SCPx3 - $TC_SCPx5)
- 荒削りサムオフセット、つまりセットアップオフセット ($TC_ECPx3 - $TC_ECPx5)
- 旋回工具ホルダのオフセットベクトル l_1 ($TC_CARR1 - $TC_CARR3)

- 旋回工具ホルダのオフセットベクトル l_2 ($TC_CARR4 - $TC_CARR6)
- 旋回工具ホルダのオフセットベクトル l_3 ($TC_CARR15 - $TC_CARR17)

注記

工具ベース寸法とアダプタ寸法はどちらか一方だけを適用できます。

個々のベクトルの動作のタイプ

個々のベクトルまたはベクトルグループの動作のタイプは、次の別の量に応じて異なります。

影響する量	動作原理
G 命令	有効な加工平面
工具タイプ	フライス工具または旋削/研削工具
マシンデータ	工具管理機能有効/無効、旋回工具ホルダあり/なし
セッティングデータ	ミラーリングの場合または平面を変更する場合の工具長成分の挙動
旋回工具ホルダ	旋回工具ホルダの値を設定
アダプタ座標変換	座標変換された工具補正値

各ジオメトリ軸成分への分配

3 つのジオメトリ軸成分に、必要なベクトルの部分的な合計である 3 つのベクトル成分を分配する方法は、次の量によって特定されます。

影響する量	依存関係
有効な処理平面: G17 X/Y 方向 G18 Z/X 方向 G19 Y/Z 方向	送り込み面: Z Y X
工具タイプ: フライス工具、穴あけ工具、研削工具、旋削工具	「工具パラメータ 1:工具タイプ (ページ 1797)」の章の、「必要な工具パラメータの最小数」の表を参照してください。

影響する量	依存関係
SD42900 $SC_MIRROR_TOOL_LENGTH SD42910 $SC_MIRROR_TOOL_WEAR SD42920 $SC_WEAR_SIGN_CUTPOS SD42930 $SC_WEAR_SIGN SD42940 $SC_TOOL_LENGTH_CONST SD42950 $SC_TOOL_LENGTH_TYPE	「工具補正の特殊処理 (ページ 1966)」の章と「セッティングデータ (ページ 2036)」の章を参照してください。
アダプタ座標変換	参照先: 『機能マニュアル、工具管理機能』

結果として生じる工具の向きは必ず、この 3 つの軸方向 X、Y、または Z のどれかに平行になったままであり、もっぱら有効な加工平面 G17-G19 によって決まります。これは、まだ工具に向きを割り当てることがまだできないからです。

工具の向きの連続変化

旋回工具ホルダを使用すれば、オフセットベクトル l_1 ～ l_3 を使用してオフセットまたは長さ修正を追加実行するだけでなく、工具の向きを連続して変化させることもできます。

詳しくは、「旋回工具ホルダ (ページ 1860)」の章を参照してください。

軽微なオペレータ補正

ただし、通常の加工モードの最中に、軽微な補正も変更する必要があります。

この理由として、たとえば以下があげられます。

- 工具の摩耗
- クランプ誤差
- 機械の温度感受性

これらの補正は、次のように定義されます。

定義	摩耗成分
摩耗	$TC_DP12 ～$TC_DP14,
サムオフセット(仕上げ)	$TC_SCPx3 ～$TC_SCPx5,
荒削りサムオフセット、つまりセットアップオフセット	$TC_ECPx3 ～$TC_ECPx5

特に、工具長の計算に影響を及ぼす補正を、測定に使用する座標に入力してください。

TOWSTD、TOWMCS、および TOWWCS の 3 つの値を含む G コードグループ 56 と次のセッティングデータを使用すれば、これらのワーク固有の補正をもっと簡単に実現できます。

SD42935 $SC_WEAR_TRANSFORM (工具成分の座標変換)

SD42935

摩耗成分:

- 摩耗($TC_DP12 ～$TC_DP14)
- セットアップオフセット、つまりサムオフセット(荒削り)($TC_ECPx3 ～$TC_ECPx5)
- サムオフセット(仕上げ)($TC_SCPx3 ～$TC_SCPx5)

が、座標変換:

- アダプタ座標変換
- 旋回工具ホルダ

において、座標変換されるか否かは、次のセッティングデータを使用して定義できます。

SD42935 $SC_WEAR_TRANSFORM (磨耗値の座標変換)

初期状態のセッティングデータを使用すると、摩耗値がすべて変換されます。

このセッティングデータは、次のように考慮されます。

- 機械座標系内の摩耗値
  パートプログラム命令:TOWMCS
- ワーク座標系内の摩耗値
  パートプログラム命令:TOWWCS

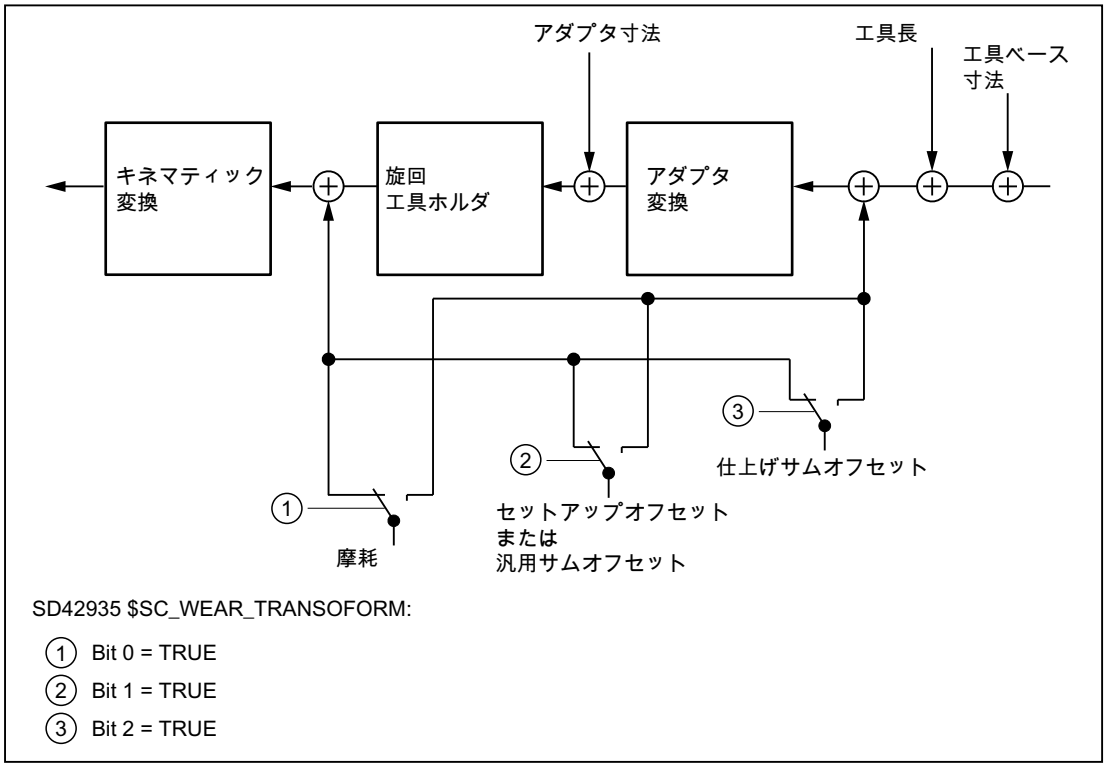


図 19-59　SD42935 に応じた摩耗データの座標変換

プログラミング

G コードグループ 56 を使用すれば、次の値を定義できます。

構文	補正
TOWSTD	工具長補正の初期設定値
TOWMCS	機械座標系(MCS)内の摩耗値
TOWWCS	ワーク座標系(WCS)内の摩耗値
TOWBCS	基本座標系(BCS)の摩耗値

構文	補正
TOWTCS	工具ホルダでの工具座標系(TCS: Tool Coordinate System)内の摩耗値 (工具ホルダレファレンス点 T)
TOWKCS	工具ヘッドのキネマティック座標変換(KCS)の場合の工具座標系内の摩耗値

工具長のオフセットに対応する座標系

TOWMCS、TOWWCS、TOWBCS、TOWTCS、および TOWKCS の各 G 命令を使用すれば、たとえば、5 種類の座標系内の摩耗工具長成分を測定できます。

1. 機械座標系	MCS
1. 基本座標系	BCS
1. ワーク座標系	WCS
1. キネマティック座標変換の工具座標系	KCS
1. 工具座標系	TCS

事前定義された GETTCOR (ページ 2005)機能を使用して、これらの座標系の 1 つで、計算した工具長または工具長成分を記述し、呼び出すことができます。

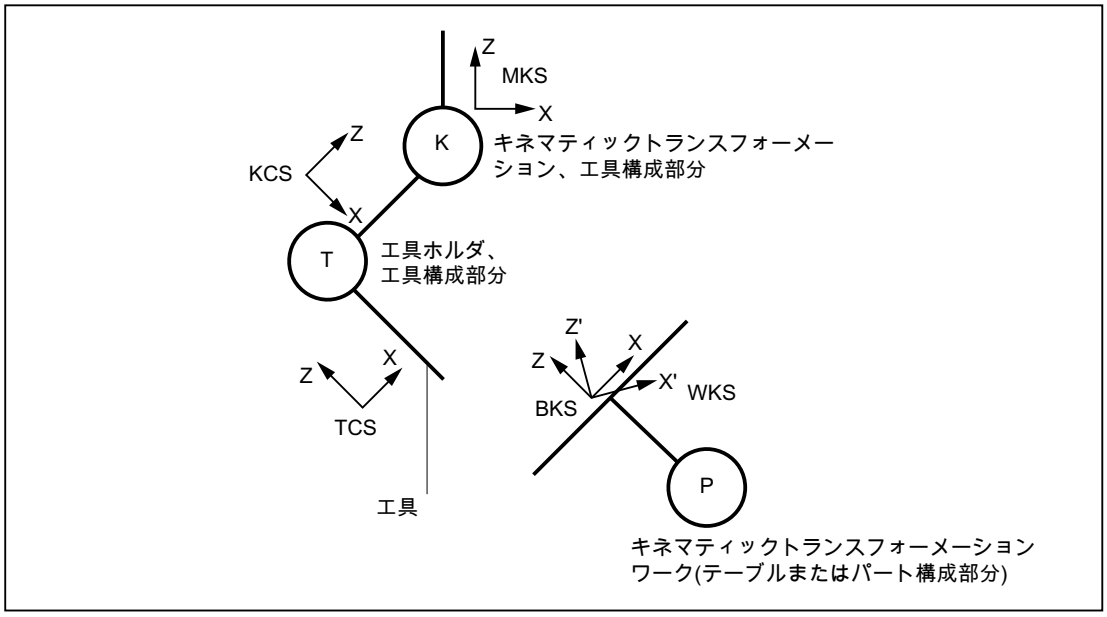


図 19-60　工具長の評価に対応した座標系

19.14.5.2 個々の摩耗値の動作

TOWSTD

初期設定(初期設定動作):

- 摩耗値は、他の工具長成分に加算されます。
  その後の計算で、結果として生じた合計工具長が使用されます。

旋回工具ホルダが有効な場合

- 摩耗値は、当該の回転の影響を受けます。

TOWMCS

MCS (機械座標系)内の摩耗データ

旋回工具ホルダによる回転が有効な場合

- 工具ホルダは、結果として生じた工具長のベクトルのみを回転させます。摩耗は無視されます。
  その後、この方法で回転した工具長ベクトルと摩耗が加算されます。この摩耗は、回転の影響を受けません。

旋回工具ホルダが有効になって**いない**か、それによる回転が発生しない場合、TOWMCS と TOWSTD は同じです。

直線的座標変換

MCS で工具長を一義的に定義できるのは、BCS からの直線的な座標変換により MCS が生成される場合に限ります。

これは、以下のいずれかの条件下にある場合にも当てはまります。

- キネマティック座標変換が無効
- 方向座標変換(3 軸、4 軸および 5 軸座標変換)が有効

TOWWCS

WCS (ワーク座標系)内の摩耗値:

- 旋回工具ホルダが有効になっている場合、TOWMCS に関して工具ベクトルが計算され、摩耗は考慮されません。
- 摩耗データは、ワーク座標系で解釈されます。

ワーク座標系内の摩耗ベクトルは、機械座標系に変換され、工具ベクトルに加算されます。

TOWBCS

BCS (基本座標系)内の摩耗値

- 旋回工具ホルダが有効になっている場合、TOWMCS に関して工具ベクトルが計算され、摩耗は考慮されません。
- 摩耗データは、ワーク座標系で解釈されます。

基本座標系内の摩耗ベクトルは、ワーク座標系に変換され、工具ベクトルに加算されます。

非直線座標変換

直線的ではない座標変換が有効になっている場合、たとえば、TRANSMIT のとき、および必要な座標系として MCS が指定されている場合、MCS ではなく、BCS が自動的に使用されます。

方向付け機能を備えた工具キャリア

旋回工具ホルダのテーブル成分を使用できる場合、キネマティック座標変換のテーブル(またはワーク)成分と異なり、このテーブル成分は座標系に直接適用されません。こうした成分で記述された回転は、基本フレームまたはシステムフレームで表されるため、WCS から BCS への移行に組み込まれます。

キネマティック座標変換

キネマティック座標変換のテーブル(またはワーク)成分は、BCS から MCS への移行により記述されます。

TOWTCS

TCS (工具座標系)内の摩耗値

- 旋回工具ホルダが有効になっている場合、TOWMCS に関して工具ベクトルが計算され、摩耗は考慮されません。
- 摩耗データは、工具座標系で解釈されます。

TCS (工具座標系)内の摩耗ベクトルは、キネマティック座標変換(KCS)の工具座標系により機械座標系に変換された後、工具ベクトルに加算されます。

TOWKCS

キネマティック座標変換の摩耗値指定は、関連の TCS (工具座標系)で解釈されます

摩耗ベクトルは、キネマティック座標変換の工具座標系により機械座標系に変換された後、工具ベクトルに追加されます。

工具が有効な場合の G 命令変更

TOWSTD、TOWMCS、TOWWCS、TOWBCS、TOWTCS、および TOWKCS のグループ内で G 命令を変更しても、既に有効になっている工具は影響を受けません。この変更は、次の工具を選択して初めて有効になります。

工具を選択するのと同じブロックでプログラム指令される場合、このグループの G 命令も有効になります。

個々の摩耗成分の評価

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、次の影響を受けます。

- 有効な平面
- アダプタ座標変換
- 次の表に記載されている 5 つのセッティングデータ

セッティングデータ	摩耗成分		
SD42910 $SC_MIRROR_TOOL_WEAR	TOWSTD	TOWMCS	TOWWCS
SD42920 $SC_WEAR_SIGN_CUTPOS	X	X	—
SD42930 $SC_WEAR_SIGN	X	—	—
SD42940 $SC_TOOL_LENGTH_CONST	X	X	X
SD42950 $SC_TOOL_LENGTH_TYPE	X	X	X

注記

アダプタ座標変換または旋回工具ホルダによる有効な回転の影響を受ける摩耗成分は、非座標変換摩耗成分と呼ばれます。

特記事項

TOWMCS または TOWWCS が有効になっている場合、次のセッティングデータは、非座標変換摩耗成分に影響を及ぼしません。

SD42920 $SC_WEAR_SIGN_CUTPOS (刃先位置のある工具の摩耗の符号)

TOWWCS の場合、次のセッティングデータも、非座標変換摩耗成分に影響を及ぼしません。

SD42910 $SC_MIRROR_TOOL_WEAR (ミラーリング時の工具摩耗の符号変更)

この場合、有効なミラーリングはすでにフレーム内に含まれています。摩耗成分を評価するために、このフレームが参照されます。

平面の変更時、非座標変換摩耗成分とジオメトリ軸間の割り当ては保持されます。つまり、これらは、他の長さ成分と同様に交換されません。各成分の割り当ては、工具選択の有効平面により異なります。

例

長さ 1 に割り当てられた摩耗値$TC_DP12 だけがゼロに等しくない場合にフライス工具を使用するとします。

G17 が有効になっている場合、この長さは、Z 軸方向で有効です。

TOWMCS または TOWWCS が有効になっており、次のセッティングデータでビット 1 が設定されている場合、工具選択後の平面交換直後にも、この量は常に Z 方向に実行されます。

SD42935 $SC_WEAR_TRANSFORM (工具成分の座標変換)

たとえば、工具選択時に G18 が有効になっている場合、代わりに、この成分は必ず Y 方向で有効になります。

19.15 工具環境の使用

機能の概要

- 工具環境の保存(TOOLENV) (ページ 1999)
- 工具環境の削除(DELTOOLENV) (ページ 2002)
- T 番号、D 番号、および DL 番号の読み出し (GETTENV) (ページ 2003)
- 工具長、工具長成分、またはその両方の読み取り(GETTCOR) (ページ 2005)
- 工具成分の変更(SETTCOR) (ページ 2013)

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 2004)

19.15.1 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。

19.15.2 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。		

19.15.3 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。

19.15.4 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」			

19.15.5 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。

19.15.6 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。

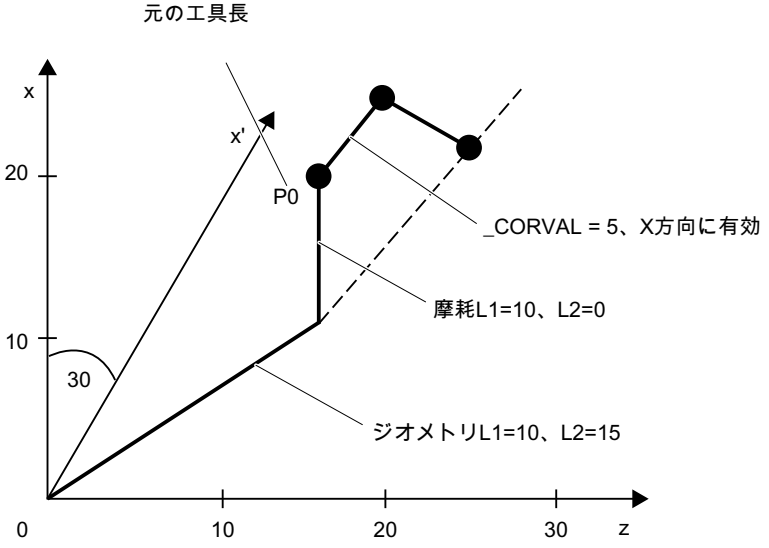
$TC_DP3[1,1]:21.830 ; ジオメトリ L1

$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

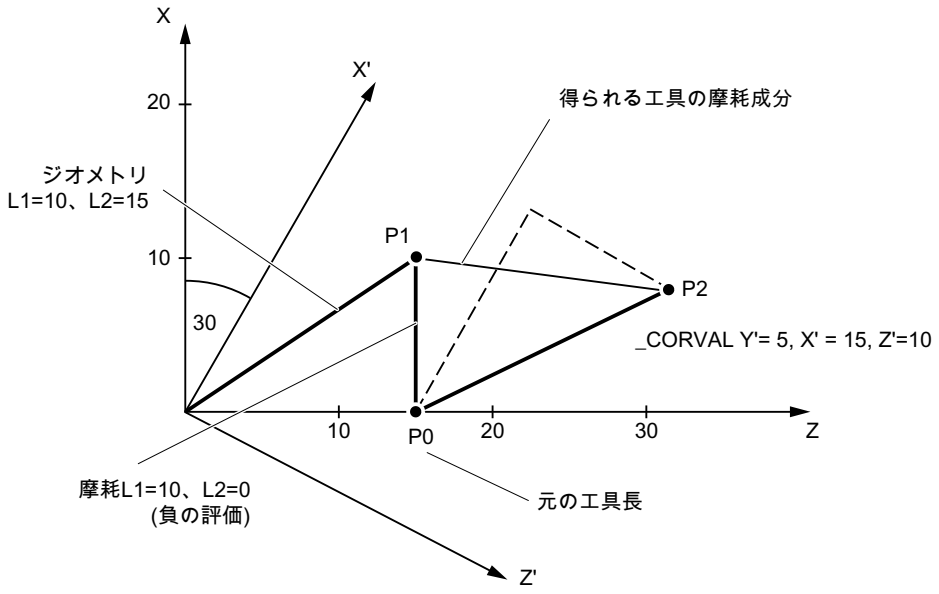
プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 $TC_DP1[1,1]=500	; 旋削工具
N30 $TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 $TC_DP4[1,1]=15.0	; ジオメトリ L2
N50 $TC_DP12[1,1]=10.0	; 磨耗 L1
N60 $TC_DP13[1,1]=0.0	; 磨耗 L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=**SETTCOR(_CORVAL,"GW",3,3)**	
N110 T1 D1 X0 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。

19.16 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				
1	<AxInd>:	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		
		データタイプ:		INT[3]
		値:	0	既存の割り当てなし(軸はありません)
			1 ... 3 または -1 ... -3	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。
		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。		
2	<Matrix>:	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。		
		データタイプ:		REAL
		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。		

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。

$$<\text{Matrix}> = \begin{pmatrix} 0 & \sin 60° & \cos 60° \\ 0 & \cos 60° & -\sin 60° \\ 1 & 0 & 0 \end{pmatrix}$$

19.17 必要条件

19.17.1 フラット D 番号構造

ブロック検索

ブロック検索については、以下を使用して、PLC に補助機能を出力するタイミングをパラメータ設定できます。

- MD22080 $MC_AUXFU_ASSIGN_SPEC
- MD22035 $MC_AUXFU_ASSIGN_SPEC
- MD11110 $MN_AUXFU_GROUP_SPEC

19.17.2 SD42935 の拡張

磨耗値の座標変換

命令 TOWMCS (機械座標系の摩耗値)および TOWWCS (ワーク座標系の摩耗値)と組み合わせて、どの摩耗成分を変換するか、変換しないかは、以下を使用して定義できます。

SD42935 $SC_WEAR_TRANSFORM (磨耗値の座標変換)

19.17.3 接触計測

接触計測

「接触計測」機能をリセット状態で使用する場合、初期設定に関して以下のことに注意してください。

- 磨耗コンポーネントは、G グループ TOWSTD、TOWMCS および TOWWCS の初期設定に応じて評価されます。
- 正確な計算を保障するために初期設定以外の値が必要な場合、接触計測は STOP 状態でのみ実行できます。

19.18 データリスト

19.18.1 マシンデータ

19.18.1.1 NC 別マシンデータ

番号	識別子: $MN_	説明
18082	MM_NUM_TOOL	NC で管理できる工具の数(SRAM)
18088	MM_NUM_TOOL_CARRIER	定義された工具ホルダの最大数
18094	MM_NUM_CC_TDA_PARAM	工具データの数(SRAM)
18096	MM_NUM_CC_TOA_PARAM	データの数、コンパイルサイクルの工具刃先当たり(SRAM)
18100	MM_NUM_CUTTING_EDGES_IN_TOA	TOA 領域内の工具オフセット(SRAM)
18102	MM_TYPE_OF_CUTTING_EDGE	D 番号プログラミングのタイプ(SRAM)

番号	識別子: $MN_	説明
18105	MM_MAX_CUTTING_EDGE_NO	D 番号の最大値
18106	MM_MAX_CUTTING_EDGE_PERTOOL	工具毎の D 番号の最大数
18108	MM_NUM_SUMCORR	NC でのサムオフセットの総数
18110	MM_MAX_SUMCORR_PER_CUTTEDGE	刃先当たりのサムオフセットの数
18112	MM_KIND_OF_SUMCORR	TO 領域内のサムオフセットの特性(SRAM)
18114	MM_ENABLE_TOOL_ORIENT	刃先への向きの割り当て
18116	MM_NUM_TOOL_ENV	TOA 領域内の工具環境(SRAM)

19.18.1.2 チャネルマシンデータ

番号	識別子: $MC_	説明
20096	T_M_ADDRESS_EXT_IS_SPINO	T、M 工具交換時のアドレス拡張の意味
20110	RESET_MODE_MASK	RESET/パートプログラム終了後の初期制御設定の定義
20120	TOOL_RESET_VALUE	電源投入(RESET/パートプログラム終了)時の工具長補正
20121	TOOL_PRESEL_RESET_VALUE	リセット時の事前選択工具
20125	CUTMOD_ERR	機能"旋回工具のオフセットデータの変更"のエラー処理
20126	TOOL_CARRIER_RESET_VALUE	リセット時に工具ホルダが機能します
20127	CUTMOD_INIT	CUTMOD の初期値
20130	CUTTING_EDGE_RESET_VALUE	電源投入(RESET/パートプログラム終了)時の工具刃先長補正
20132	SUMCORR_RESET_VALUE	RESET 時に有効な追加オフセット
20140	TRAFO_RESET_VALUE	電源投入(RESET/パートプログラム終了)時の座標変換データ記録
20180	TOCARR_ROT_ANGLE_INCR[i]	旋回工具ホルダの回転軸移動単位
20182	TOCARR_ROT_ANGLE_OFFSET[i]	旋回工具ホルダの回転軸オフセット
20184	TOCARR_BASE_FRAME_NUMBER	テーブルオフセットを取得するための基本フレームの番号
20188	TOCARR_FINE_LIM_LIN	TCARR 直線仕上げオフセット制限
20190	TOCARR_FINE_LIM_ROT	TCARR 回転仕上げオフセット制限

番号	識別子: $MC_	説明
20202	WAB_MAXNUM_DUMMY_BLOCKS	SAR での移動動作のないブロックの最大数
20204	WAB_CLEARANCE_TOLERANCE	SAR に対する方向の反転
20210	CUTCOM_CORNER_LIMIT	工具径補正による交点演算に対する最大角度
20220	CUTCOM_MAX_DISC	DISC の最大値
20230	CUTCOM_CURVE_INSERT_LIMIT	TRC での交点演算の最大値
20240	CUTCOM_MAXNUM_CHECK_BLOCKS	工具径補正による予測輪郭演算に対応したブロック
20250	CUTCOM_MAXNUM_DUMMY_BLOCKS	TRC に対する移動動作なしのブロックの最大数
20252	CUTCOM_MAXNUM_SUPPR_BLOCKS	補正をマスクするブロックの最大数
20256	CUTCOM_INTERS_POLY_ENABLE	多項式で可能な交点処理
20270	CUTTING_EDGE_DEFAULT	工具刃先のプログラムがないときの初期設定
20272	SUMCORR_DEFAULT	サムオフセットのプログラムがないときの初期設定
20360	TOOL_PARAMETER_DEF_MASK	工具パラメータの定義
20390	TOOL_TEMP_COMP_ON	工具長の熱変位補正の起動
20392	TOOL_TEMP_COMP_LIMIT	工具長の最大熱変位補正
20610	ADD_MOVE_ACCEL_RESERVE	重畳動作に対応した加速度余裕
21080	CUTCOM_PARALLEL_ORI_LIMIT	3D 工具径補正の最小角度 (軌跡接線と工具方向)
22530	TOCARR_CHANGE_M_CODE	工具ホルダ変更用の M コード
22550	TOOL_CHANGE_MODE	M 機能による新しい工具補正
22560	TOOL_CHANGE_M_CODE	工具交換の M 機能
22562	TOOL_CHANGE_ERROR_MODE	工具交換時にエラーが発生した場合の応答
24558	TRAFO5_JOINT_OFFSET_PART_1	テーブルのキネマティックオフセットのベクトル、座標変換 1
24658	TRAFO5_JOINT_OFFSET_PART_2	テーブルのキネマティックオフセットのベクトル、座標変換 2
28085	MM_LINK_TOA_UNIT	チャネルへの TO ユニットの割り当て(SRAM)

19.18.1.3 軸/主軸マシンデータ

番号	識別子: $MA_	説明
32750	TEMP_COMP_TYPE	熱変位補正タイプ

19.18.2 セッティングデータ

19.18.2.1 チャネルセッティングデータ

番号	識別子$SC_	説明
42442	TOOL_OFFSET_INCR_PROG	工具長補正
42470	CRIT_SPLINE_ANGLE	コーナ制限角度、コンプレッサ用
42480	STOP_CUTCOM_STOPRE	工具径補正と先読み停止のアラーム応答
42494	CUTCOM_ACT_DEACT_CTRL	工具径補正のためのアプローチと後退
42496	CUTCOM_CLSDT_CONT	閉じた輪郭のための工具径補正の動作
42900	MIRROR_TOOL_LENGTH	ミラーリング時の工具長符号変更
42910	MIRROR_TOOL_WEAR	ミラーリング時の工具磨耗符号変更
42920	WEAR_SIGN_CUTPOS	刃先位置のある工具の磨耗符号
42930	WEAR_SIGN	磨耗の符号
42935	WEAR_TRANSFORM	工具コンポーネントの変換
42940	TOOL_LENGTH_CONST	平面の変更に対する工具長コンポーネントの変更
42950	TOOL_LENGTH_TYPE	工具タイプとは無関係な工具長補正の割り当て
42960	TOOL_TEMP_COMP	工具に対する熱変位補正値
42974	TCARR_FINE_CORRECTION	仕上げオフセット TCARR オン/オフ
42984	CUTDIRMOD	$P_AD[2]または$P_AD[11]の変更
42998	CUTMOD_PLANE_TOL	CUTMOD または CUTMODK の工具先端平面および加工平面の差異
42999	ORISOLH_INCLINE_TOL	ORISOLH の工具先端平面と加工平面間の差異

19.18.3 信号

19.18.3.1 チャネルからの信号

信号名	SINUMERIK 840D sl	SINUMERIK 828D
T 機能 1 の変更	DB21、... .DBX61.0	-
D 機能 1 の変更	DB21、... .DBX62.0	-
T 機能 1	DB21、... .DBB116-119	DB250x.DBD2000
D 機能 1	DB21、... .DBB128-129	DB250x.DBD5000
グループ 7 の有効な G 命令	DB21、... .DBB214	DB350x.DBB6
グループ 16 の有効な G 命令	DB21、... .DBB223	DB350x.DBB15
グループ 17 の有効な G 命令	DB21、... .DBB224	DB350x.DBB16
グループ 18 の有効な G 命令	DB21、... .DBB225	DB350x.DBB17
グループ 23 の有効な G 命令	DB21、... .DBB230	DB350x.DBB22

Z1:NC/PLC インタフェース信号

20

エディション 05/2017 以降、NC/PLC インタフェース信号の詳細については、『NC 変数およびインタフェース信号用パラメータマニュアル』を参照してください。

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 マニュアルの一覧

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

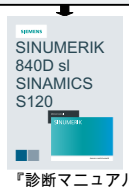
システムマニュアル
Ctrl-Energy

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

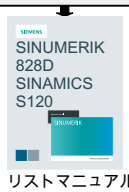
メーカ/サービスマニュアル

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

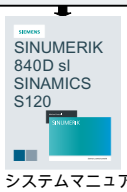
製品マニュアル
試運転マニュアル
サービスマニュアル

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。

構文

$SC_IS_SD_MAX_PATH_JERK = *値*

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。

動作

このセッティングデータをプログラムすれば、軌跡加々速度制限の起動/解除ができます。

パラメータ:*値*

- 数値の範囲: TRUE、FALSE

使用方法:

- パートプログラム
- 静的シンクロナイズドアクション

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

$

$AA_ACC, 1757
$AA_ATOL, 298
$AA_FGREF, 1728
$AA_FGROUP, 1728
$AA_G0MODE, 308
$AA_MOTEND, 1758
$AA_S, 1561
$AA_SCPAR, 1759
$AA_VLFCT, 1670
$AA_VMAXB, 1670
$AA_VMAXM, 1670
$AC_ACT_PROG_NET_TIME, 818
$AC_ACTUAL_PARTS, 827, 832
$AC_ASUP, 719
$AC_AUXFU_EXT, 553
$AC_AUXFU_M_EXT, 553
$AC_AUXFU_M_STATE, 554
$AC_AUXFU_M_TICK, 530
$AC_AUXFU_M_VALUE, 553
$AC_AUXFU_PREDEF_INDEX, 536, 553
$AC_AUXFU_SPEC, 550, 554
$AC_AUXFU_STATE, 554
$AC_AUXFU_TYPE, 553
$AC_AUXFU_VALUE, 553
$AC_CTOL, 298
$AC_CUT_INV, 1946
$AC_CUTMOD, 1946
$AC_CUTMOD_ANG, 1919, 1946
$AC_CUTMODK, 1946
$AC_CUTTING_TIME, 822
$AC_CYCLE_TIME, 822
$AC_DELAYFST, 698
$AC_FGROUP_MASK, 1728
$AC_OLD_PROG_NET_TIME, 819
$AC_OLD_PROG_NET_TIME_COUNT, 819
$AC_OPERATING_TIME, 822
$AC_OTOL, 298
$AC_PATHACC, 344
$AC_PATHACC; $AC__PATHACC, 330
$AC_PATHJERK, 343, 344, 345
$AC_PROG_NET_TIME_TRIGGER, 820
$AC_REQUIRED_PARTS, 827, 832
$AC_SGEAR, 1619, 1643
$AC_SMAXACC, 1669
$AC_SMAXACC_INFO, 1669
$AC_SMAXVELO, 1669
$AC_SMAXVELO_INFO, 1669
$AC_SMINVELO, 1669
$AC_SMINVELO_INFO, 1669
$AC_SPECIAL_PARTS, 827, 832
$AC_SPIND_STATE, 1669
$AC_STOLF, 310
$AC_TOTAL_PARTS, 827, 832
$AN_AUXFU_LIST_ENDINDEX, 542
$AN_POWERON_TIME, 817
$AN_SETUP_TIME, 817
$C_AUX_EXT, 794
$C_AUX_IS_QUICK, 794
$C_AUX_VALUE, 794
$C_D, 795
$C_D_PROG, 795
$C_DL, 795
$C_DL_PROG, 795
$C_DUPLO, 795
$C_DUPLO_PROG, 794
$C_M, 794
$C_M_PROG, 794
$C_ME, 794
$C_MTL, 790, 795
$C_MTL_PROG, 790, 795
$C_T, 794
$C_T_PROG, 794
$C_TCA, 794
$C_TE, 794
$C_THNO, 795
$C_THNO_PROG, 795
$C_TS, 794
$C_TS_PROG, 794
$NT_ROT_AX_NAME, 1941
$P_AD, 1945
$P_CHANNO; $P__CHANNO, 690
$P_CTOL, 299
$P_CUT_INV, 1946
$P_CUTMOD, 1946
$P_CUTMOD_ANG, 1919, 1946
$P_CUTMOD_ERR, 1947
$P_CUTMODK, 1946
$P_DELAYFST, 698
$P_FGROUP_MASK, 1728
$P_GEAR, 1643
$P_GFRNUM, 932
$P_IFRAME, 928
$P_ISTEST, 591
$P_ORI_DIFF, 1935
$P_ORI_POS, 1934

$P_ORI_SOL, 1936
$P_ORI_STAT, 1939
$P_OTOL, 299
$P_PROG_EVENT; $P_PROG_EVENT, 689
$P_REPINF, 712
$P_SEARCH_S, 533, 606
$P_SEARCH_SDIR, 533, 606
$P_SEARCH_SGEAR, 533, 606, 1619
$P_SEARCH_SMODE, 606
$P_SEARCH_SPOS, 533, 606
$P_SEARCH_SPOSMODE, 533, 606
$P_SGEAR, 1619
$P_SIM, 599
$P_STOLF, 310
$P_SUB_AUTOGEAR, 805
$P_SUB_AXFCT, 805
$P_SUB_CA, 805
$P_SUB_GEAR, 805
$P_SUB_LA, 805
$P_SUB_M19, 806
$P_SUB_SPOS, 806
$P_SUB_SPOSA, 806
$P_SUB_SPOSIT, 806
$P_SUB_SPOSMODE, 806
$P_SUB_STAT, 792, 795
$P_TOOLENV, 2005
$P_TOOLENVN, 2004
$P_TOOLR:, 1921
$P_UIFR, 927
$P_UIFRNUM, 928
$P_WORKAREA_CS_COORD_SYSTEM, 140
$PA_ATOL, 299
$PA_FGREF, 1728
$PA_FGROUP, 1728
$SC_IS_SD_MAX_PATH_ACCEL; $SC_IS_SD_MAX_PATH_ACCEL, 328
$SC_IS_SD_MAX_PATH_JERK; $SC_IS_SD_MAX_PATH_JERK, 342
$SC_PA_CENT_ABS, 179
$SC_PA_CENT_ORD, 179
$SC_PA_CONT_ABS, 179
$SC_PA_CONT_NUM, 178
$SC_PA_CONT_ORD, 179
$SC_PA_CONT_TYP, 178
$SC_PA_LIM_3DIM, 178
$SC_PA_MINUS_LIM, 178
$SC_PA_ORI, 178
$SC_PA_PLUS_LIM, 178
$SC_PA_T_W, 178
$SC_SD_MAX_PATH_ACCEL; $SC_SD_MAX_PATH_ACCEL, 327
$SC_SD_MAX_PATH_JERK; $SC_SD_MAX_PATH_JERK, 342
$SN_PA_ACTIV_IMMED, 178, 192
$SN_PA_CENT_ABS, 179
$SN_PA_CENT_ORD, 179
$SN_PA_CONT_ABS, 179
$SN_PA_CONT_NUM, 178
$SN_PA_CONT_ORD, 179
$SN_PA_CONT_TYP, 178
$SN_PA_LIM_3DIM, 178
$SN_PA_MINUS_LIM, 178
$SN_PA_ORI, 178
$SN_PA_PLUS_LIM, 178
$SN_PA_T_W, 178
$TC_DP1, 1914
$TC_DP10, 1916
$TC_DP11, 1915
$TC_DP2, 1915, 1921
$TC_DP24, 1916
$TC_DPCE[t,d], 1778
$TC_DPNT, 1698
$TC_DPROT, 1921
$VA_ABSOLUTE_ENC_DELTA_INIT, 129
$VA_ENC_ZERO_MON_ERR_CNT, 126, 129
$VA_TORQUE_AT_LIMIT, 399
$VC_SGEAR, 1618

(

(Single BLock)
SBL1; SBL1, 729
SBL2; SBL2, 729
SBL3; SBL3, 729

_

_N_STRTLK, 589

1

1 刃当り送り速度, 1698

3

3 次元正面削り, 1806

A

ACC, 1654, 1756
Access MyMachine (AMM), 1476

ACN, 1652
ACP, 1652
ADDFRAME, 974
AG_SEND、AG_RECV, 1091
ALF, 713, 1713
ANY;ＡＮＹ, 1358
ASUB
SERUPRO の終了, 542
ユーザーアラーム用, 720
起動, 702
再解析, 702
内部, 717
優先順位, 713
ATOL, 295
ATRANS, 849
AUXFUDEL, 544
AUXFUDELG, 544
AUXFUMSEQ, 531
AUXFUSYNC, 543

B

BLSYNC, 713
BRISK;ＢＲＩＳＫ, 322
BRISKA, 322

C

CDOF, 1843
CDOF2, 1843
CDON, 1843
CFC, 1695
CFCIN, 1695
CFINE, 849
CFTCP, 1695
CLRINT, 714
CNC ロック機能, 1476
COMPCAD, 287, 291
COMPCURV, 286, 291
COMPOF, 291
COMPON, 286, 291
COMPSURF, 287, 291
CORROF, 893
CPROT, 188
CPROTDEF, 184
CTOL, 295
CTRANS, 849
CUTMOD, 1941
CUTMODK, 1941

D

D/DL 機能の置換, 790
DB 2 のメッセージ信号;ＤＢ ２ノメッセージシンゴウ, 1356
DB 31、...
DBX83.1, 1574
DB 31、...;ＤＢ ３１、...
DBX60.6;ＤＢＸ６０．６, 1675
DBX60.7;ＤＢＸ６０．７, 1675
DB10
DBB4-7, 1025
DBB71;ＤＢＢ７１, 418
DBX103.0, 48
DBX104.3, 1107
DBX104.4, 1107
DBX104.7, 47, 1087
DBX106.1, 1024, 1026, 1027
DBX107.7, 418
DBX108.3, 47
DBX108.5, 48
DBX108.6, 48
DBX108.7, 48
DBX109.0, 48
DBX109.6, 48
DBX109.7, 48
DBX56.1, 61, 1023, 1025, 1027
DBX56.2, 1023, 1026
DBX56.4, 82
DBX56.5, 82
DBX56.6, 82
DBX56.7, 82
DBX92.0, 1086
DBX92.1, 1086
DBX92.2, 1086
DBX92.4, 1087
DBX92.5, 1087
DBX92.6, 1087
DB11
DBX0.0, 578, 580
DBX0.1, 578, 1498, 1501
DBX0.2, 578, 1498, 1500
DBX0.4, 584
DBX0.5, 575
DBX0.6, 575
DBX0.7, 575, 664, 665, 1026, 1500, 1502, 1503
DBX1.0 - DBX1.2, 578
DBX1.2, 1498, 1501
DBX26.4, 581
DBX26.5, 581
DBX4.0 ～ DBX4.2, 577

DBX46.4, 581
DBX46.5, 581
DBX5.0 - DBX5.2, 578
DBX6.0, 578, 581
DBX6.1, 578
DBX6.2, 578
DBX6.3, 575, 1024, 1027
DBX6.4, 581
DBX6.5, 581
DBX6.7, 575
DBX7.0, 581
DBX7.0 - DBX7.2, 578
DB1600, 1374
DB19
DBB13, 53
DBB16, 53
DBB17, 53
DBB26, 53
DBB27, 53
DBB33, 1121
DBB36, 1121
DBX 0.3, 52
DBX 0.4, 52
DBX0.0, 51
DBX0.1, 51, 52
DBX0.2, 52
DBX0.7, 53
DBX20.3, 52
DBX20.4, 52
DBX32.0 - .5, 1120
DBX32.6, 1120
DBX32.7, 1120
DB1900
DBB5011, 1474
DBB5021, 1474
DBB5022, 1474
DBX5021.0 ～.5, 1475
DBX5021.6, 1475
DBX5021.7, 1475
DB21
DBX24.4;ＤＢＸ２４．４, 516
DBX378.1, 711
DB21、...
D35.0, 661
D35.5, 661, 662
DBB116 ～ DBB136, 523
DBB140 ～ DBB190, 524
DBB194, 1655
DBB194 ～ DBB206, 524
DBB202, 1655
DBB376, 683
DBB4, 1732
DBB5, 1732
DBB58 ～ DBB67, 523
DBB68 ～ DBB112, 523
DBB68 以降, 1655
DBX0.4, 593
DBX0.6, 596, 1738
DBX1.0, 1501
DBX1.1, 211
DBX1.6, 601, 605
DBX1.7, 590
DBX10.0 ～ DBX11.1, 181
DBX12.3, 1731
DBX16.3, 1731
DBX2.0, 598, 662, 694, 695
DBX2.0 - 7, 740
DBX20.3, 1731
DBX24.6, 595, 1738
DBX25.7, 590
DBX26.0, 597
DBX272.0 ～ DBX273.1, 181
DBX274.0 ～ DBX275.1, 181
DBX276.0 ～ DBX277.1, 182, 209
DBX278.0 ～ DBX279.1, 182, 209
DBX29.0, 1749
DBX29.1, 1749
DBX29.2, 1749
DBX29.3, 1749
DBX30.5, 516
DBX31.0 - 2, 623
DBX31.0 ～ DBX31.2, 624
DBX31.0-31.2, 630
DBX31.4, 623, 624, 625
DBX31.6 - 7, 740
DBX317.1, 827, 832
DBX318.0, 703, 704
DBX318.5, 516
DBX318.6, 1740
DBX319.0, 625
DBX319.1 ～ DBX319.3, 624, 625, 627
DBX319.5, 625, 628
DBX32.3, 601
DBX32.4, 601, 603
DBX32.6, 534, 601
DBX33.0, 1501
DBX33.4, 601
DBX33.7, 590
DBX35.0, 661, 665
DBX35.0 - DBX35.4, 711
DBX35.1, 665
DBX35.2, 665
DBX35.3, 593, 665
DBX35.4, 665

DBX35.5, 661, 667
DBX35.5 - DBX35.7, 711
DBX35.6, 667
DBX35.7, 665, 667
DBX36.2;ＤＢＸ３６．２, 1499, 1502
DBX36.6, 49
DBX36.7, 49, 605
DBX384.0, 671
DBX39.0, 181
DBX39.1, 49
DBX4.3, 1731
DBX6.0, 1731
DBX6.1, 1567, 1612
DBX6.6, 1732
DBX6.7, 1732
DBX7.1, 590, 663
DBX7.2, 662, 694
DBX7.3, 662, 694
DBX7.4, 662, 694, 1614, 1668
DBX7.5, 588
DBX7.7, 664, 1026, 1500, 1502, 1503
DBX8.0 ～ DBX9.1, 181
DB21、... DB32.6, 605
DB21、... DBX0.4, 603, 710
DB21、... DBX6.1, 710
DB21、...;ＤＢ２１、...
DBX6.2;ＤＢＸ６．２, 54
DB21、...DBB4, 113, 114
DB21、...DBX36.6, 605
DB21、…
DBX378.0, 702
DB31 、...DBX62.5, 381
DB31、...
DBB0, 1734
DBB1.5;ＤＢＢ１．５, 1543
DBB1.6;ＤＢＢ１．６, 1543
DBB19, 1735
DBB2.1;ＤＢＢ２．１, 1543
DBB60.4, 1536, 1539
DBB60.5, 1536, 1539
DBB68 以降, 1655
DBX1.0, 54, 65
DBX1.1, 385
DBX1.2, 380
DBX1.3, 54, 380, 1668
DBX1.4, 55, 58, 62, 117, 1529, 1530
DBX1.5, 59, 123, 145, 429, 1602
DBX1.6, 59, 123, 145, 429, 1602
DBX1.7, 1734, 1735
DBX10.0, 621, 623, 625, 630
DBX102.5, 145, 149
DBX102.6, 145, 149
DBX12.0, 130
DBX12.1, 130
DBX12.2, 132
DBX12.3, 132
DBX12.7, 1509, 1512
DBX16.0 - 2, 1627
DBX16.0 ～ DBX16.2, 1562, 1608, 1613, 1614, 1615, 1620, 1634
DBX16.3, 1557, 1562, 1608, 1614, 1615, 1626, 1629, 1632, 1634, 1638
DBX16.4, 1603
DBX16.5, 1603
DBX16.7, 1668
DBX17.0, 1637
DBX17.4 ～ DBX17.5, 1633
DBX18.4, 1634
DBX18.5, 1562, 1608, 1634, 1668
DBX2.1, 55, 58, 60, 61, 145, 147, 380, 1529, 1530, 1634
DBX2.2, 63, 621, 1560, 1570, 1614, 1668
DBX2.3, 117
DBX2.4 ～ DBX2.7, 1517
DBX20.1, 1615
DBX21.0-4, 66, 68, 167
DBX21.5, 67, 167
DBX21.6, 67, 167
DBX21.7, 61, 67, 167, 385
DBX28.7, 587
DBX3.1, 379, 383, 385
DBX3.2, 1749
DBX3.3, 1749
DBX3.4, 1749
DBX3.5, 1749
DBX3.7, 649
DBX30.0, 1660
DBX30.1, 1660
DBX30.2, 1660
DBX30.3, 1660
DBX30.4, 1660
DBX39.0, 208, 213
DBX4.3, 1632, 1668
DBX4.6, 1498
DBX4.7, 1498
DBX60.2, 124, 1673
DBX60.3, 124, 1673
DBX60.4, 59, 145, 1499, 1500, 1501, 1502, 1503, 1517, 1524, 1527, 1530, 1673
DBX60.4 ～ DBX60.5, 1622, 1627, 1632, 1633, 1634
DBX60.5, 59, 145, 1027, 1499, 1500, 1501, 1502, 1503, 1517, 1524, 1527, 1530, 1673
DBX60.6, 59, 106, 117, 1566, 1574, 1580

DBX60.7, 59, 106, 117, 1574, 1580, 1638
DBX61.0, 54, 65
DBX61.3, 55, 65, 1529, 1530
DBX61.4, 64, 65, 1530, 1561, 1568, 1608, 1666
DBX61.5, 61, 62, 65, 1560, 1626, 1627, 1628, 1633
DBX61.6, 62, 66
DBX61.7, 66
DBX62.4, 379, 383, 384
DBX62.5, 384, 389
DBX64.6, 106, 528
DBX64.7, 106, 528
DBX69.0, 66, 92
DBX69.0 - DBX69.2, 1621
DBX69.1, 66, 92
DBX69.2, 66, 92
DBX7.7, 1668
DBX70.0, 625, 627, 628
DBX70.1, 625, 627, 628
DBX70.2, 625, 627
DBX71.4, 1603
DBX71.5, 1603
DBX72.0, 625
DBX76.0, 66
DBX76.4, 625, 628
DBX82.0 - DBX82.2, 1608, 1610, 1612, 1613, 1625, 1626, 1627, 1628, 1633, 1636
DBX82.3, 1608, 1610, 1611, 1626, 1627, 1628, 1633, 1636
DBX83.1, 1561, 1610, 1612, 1667, 1669
DBX83.2, 1610, 1612, 1667, 1668
DBX83.5, 1561, 1567, 1568, 1569, 1667
DBX84.5, 1633, 1638
DBX84.6, 1557, 1626, 1627, 1628, 1633, 1636, 1638
DBX84.7, 1559, 1608, 1634
DBX85.5, 1581
DBX9.0, 64, 66, 91
DBX9.1, 64, 66, 91
DBX9.2, 64, 66, 91
DBX9.3, 65
DBX92.1, 68
DBX93.5, 48, 61, 68
DBX93.6, 67, 68
DBX93.7, 68, 69
DBX94.0, 69
DBX94.1, 69
DBX94.2, 70
DBX94.3, 70
DBX94.4, 70
DBX94.5, 70
DBX94.6, 70
DB31、... DBX1.4, 117
DB31、...;ＤＢ３１、...
  DBB60.4;ＤＢＢ６０．４, 1526
  DBB60.5;ＤＢＢ６０．５, 1526
DB31、...DBX1.1, 381
DB31、...DBX102.3, 108, 109, 110
DB31、...DBX2.3, 107, 109
DB31、...DBX61.3, 110
DB31、...DBX64.6, 110, 112, 114
DB31、…
  DBX9.3, 1759
DB31;ＤＢ３１
  DBX84.0, 1661, 1662
DB4500, 1375
DB9000 - DB9063, 1375
DC, 1652
DELAYFSTOF, 696
DELAYFSTON, 696
DELTOOLENV, 2002
DIACYCOFA, 1038
DIAM90, 1037
DIAM90/DIAM90A[AX], 1031
DIAM90A, 1038
DIAMCHAN, 1038
DIAMCHANA, 1038
DIAMCYCOF, 1037
DIAMCYCOF/DIACYCOFA[AX], 1032
DIAMOF, 1037
DIAMOFA, 1038
DIAMON, 1037
DIAMON/DIAMONA[AX], 1031
DIAMONA, 1038
DILF, 1713
DISABLE, 714
DISC, 1835
DITE, 1711
DITS, 1711
DL 機能, 481
DRIVE, 360
DRIVEA, 361
DYNFINISH, 279
DYNNORM, 279
DYNPOS, 279
DYNROUGH, 279
DYNSEMIFIN, 279
D 機能, 480, 1771
D 番号
  任意割り当て..., 1777
D 番号構造;Dバンゴウコウゾウ
  - フラット(工具管理なし);ー フラット（コウグカンリナシ）, 1788

E

EES, 756
ENABLE, 714
ENDLABEL の間にあるパートプログラム区間, 673
Ethernet
　接続, 1105
EXTCLOSE, 772
EXTOPEN, 772

F

FA, 1654, 1729
FA 機能, 482
FB, 1754
FB1 RUN_UP(基本プログラム、起動セクション), 1162
FB10 安全リレー, 1249
FB11 ブレーキテスト;ＦＢ１１ブレーキテスト, 1252
FB2 GET (NC 変数の読み取り), 1173
FB29 信号レコーダおよびデータトリガ診断;ＦＢ２９シンゴウレコーダオヨビデータトリガシンダン, 1258
FB3 PUT (NC 変数の書き込み), 1184
FB4 PI_SERV (PI サービス), 1193
　使用可能な PI サービス, 1199, 1200
FB5 GETGUD (GUD 変数の読み取り), 1232
FB7 PI_SERV2 (PI サービス), 1242
FB9 MzuN (操作パネルの切り替え), 1243
FB の ANY;ＦＢノＡＮＹ, 1360
FB の POINTER;ＦＢノＰＯＩＮＴＥＲ, 1360
FC10 AL_MSG, 1287
FC1005 AG_SEND, 1350
FC1006 AG_RECV, 1352
FC12 AUXFU;ＦＣ１２　ＡＵＸＦＵ, 1289
FC13 BHGDisp, 1291
FC17 YDelta, 1297
FC17 によるスター/デルタ切り替え, 1615
FC18 SpinCtrl, 1301
FC19 / MCP_IFM, 1314
FC2 GP_HP (基本プログラム、サイクリックセクション), 1263
FC21 伝送, 1322
FC22 TM_DIR;ＦＣ２２　ＴＭ__ＤＩＲ, 1332
FC24 MCP_IFM2, 1335
FC25 / MCP_IFT, 1339
FC26 HPU_MCP, 1343
FC3 GP_PRAL (基本プログラム、割り込み起動セクション), 1264
FC5 GP_DIAG (基本プログラム、診断), 1267
FC6 TM_TRANS2, 1269
FC7 TM_REV, 1271
FC8 TM_TRANS, 1275
FC9 ASUB;ＦＣ９　ＡＳＵＢ, 1284
FCUB, 1750
FC の ANY;ＦＣノＡＮＹ, 1359
FC の POINTER;ＦＣノＰＯＩＮＴＥＲ, 1359
FGROUP, 860
FGROUP;ＦＧＲＯＵＰ, 856
FIFO バッファ, 752
FL, 1695
FLIN, 1750
FNORM, 1750
FOC, 376
FOCOF, 376
FOCON, 376
FPO, 1750
FPRAOF, 1653
FPRAON, 1653
FRAME;ＦＲＡＭＥ, 855
FRC, 1751
FRCM, 1751
FXS, 376
FXS-REPOS, 390
FXST, 376
FXSW, 376
FZ, 1698
F 機能, 481

G

G0 許容範囲係数, 303
G0 許容範囲係数, 308
G25, 137, 1653
G26, 137, 1653
G33, 1705
G331, 1723
G332, 1723
G335, 1718
G336, 1718
G40;Ｇ４０, 1813
G41;Ｇ４１, 1813
G42;Ｇ４２, 1813
G450/G451, 1834
G451, 1837
G460, 1852
G461, 1769, 1853
G462, 1769, 1854
G58, 849
G59, 849
G60, 231

G601, 232
G602, 232
G603, 232
G63, 1597, 1725
G64, 239
G642, 245
G643, 245
G644, 249
G645, 253
G70, 423
G700, 423
G71, 423
G710, 423
G74, 1503
G9, 231
G91 拡張, 1951
G91 拡張;Ｇ９１カクチョウ
　原点オフセット;ゲンテンオフセット, 1952
G93, 1696
G94, 1697
G95, 1697
G96, 1700
G96、G961 でのマスタとスレーブの切替, 1704
G96、G961 による主軸速度制限, 1703
G961, 1700
G97, 1702
G971, 1702
GETTCOR, 2005
GETTENV, 2003
GFRAME0 ... GFRAME100, 932
G グループ, 654

H

HHH"; "UUU, 109
HMI モニタ, 1138, 1405
H 機能, 479

I

IC, 1653
IR, 1718

J

JOG による LEAD に対するシミュレーションされた目標位置への到達, 645
JR, 1718

K

KONT, 1814
KONTC, 1814
KONTT, 1814
KR, 1718

L

LENTOAX, 2029
LFOF, 1713
LFON, 1713
LFPOS, 1713
LFTXT, 1713
LFWP, 1713
LIFTFAST, 713
LIMS, 1654
lockset ファイル
　CNC ロック機能を解除するには, 1484
　CNC ロック機能を拡張するには, 1482
　CNC ロック機能を起動するには, 1478
　コントロールシステムをロック解除するには, 1487, 1490

M

M1, 557
M17, 556
M19, 1564, 1653
M2, 556
M3, 1651
M30, 556
M4, 1652
M40, 1655, 1676
M41, 1655
M42, 1655
M43, 1655
M44, 1655
M45, 1655
M5, 1652
M70, 1653
MCP の識別, 1108
Md < Mdx, 70
MD10000, 846
MD10010, 572
MD10050, 1544
MD10070, 1544
MD10125, 765
MD10131, 71
MD10192, 1624

MD10200, 406, 407, 408, 410, 421
MD10210, 406, 407, 408, 410, 448, 450
MD10220, 412
MD10230, 412
MD10240, 414, 419, 425, 1760
MD10250, 420
MD10260, 420, 423
MD10270, 422
MD10290, 422
MD10292, 422
MD10366, 1378
MD10368, 1378
MD10600, 902, 984
MD10602, 945, 947, 955, 960
MD10610, 851, 938
MD10612, 939
MD10615;ＭＤ１０６１５, 992
MD10618, 184
MD10651, 1719
MD10680, 285
MD10682, 284
MD10700, 745
MD10702, 594, 603, 710, 730
MD10704, 1739
MD10707, 617
MD10708, 617
MD10710, 138, 1703, 1713
MD10712, 281, 815
MD10713, 525
MD10714, 499, 789
MD10715, 787
MD10716, 787
MD10719, 791
MD10735, 580
MD10804, 789
MD10806, 789
MD10814, 789
MD1103, 393
MD1104, 393
MD1105, 393
MD11100, 512
MD11110, 519, 2032
MD11220, 421
MD11300, 1499
MD11346, 1787
MD11410, 1608, 1706, 1842
MD11411;ＭＤ１１４１１, 588
MD11450, 533, 605, 606, 607, 609, 619, 647
MD11470, 620, 621, 649
MD11550, 696, 1724
MD11600, 707
MD11602, 706, 708, 709, 710
MD11604, 647, 706, 709
MD11610, 718
MD11620, 684
MD11625, 765
MD11626, 765
MD12000, 1734
MD12010, 1735
MD12020, 1732
MD12030, 257, 259, 1733
MD12040, 1733
MD12050, 1733
MD12060, 1736
MD12070, 1736
MD12080, 1736
MD12082, 1734
MD12090, 1724
MD12100, 257, 259, 1737
MD12200, 1749
MD12202, 1748
MD12204, 1748
MD1230/1231, 393
MD14504, 1099
MD14506, 1099
MD14508, 1099
MD14516, 1437
MD14518, 1437
MD15700, 802
MD15702, 802
MD17200;ＭＤ１７２００, 744
MD17300, 1481
MD18080;ＭＤ１８０８０, 1977
MD18088, 1865, 1894, 1899
MD18100, 1773, 1789
MD18102, 1780, 1788, 1790
MD18104;ＭＤ１８１０４, 2001
MD18105, 1778, 1779, 1781, 1782
MD18106, 1778, 1779
MD18108;ＭＤ１８１０８, 1981
MD18112;ＭＤ１８１１２, 1983
MD18114, 1960
MD18116, 2001
MD18150, 74
MD18190, 183
MD18360, 752
MD18362, 752
MD18600, 899
MD18602, 926, 930, 933, 935
MD18960, 321, 340
MD20000, 585
MD20050, 782, 846, 966
MD20060, 846
MD20070, 846

MD20080, 846, 860
MD20090, 1651
MD20090;MD 2 0 0 9 0, 555
MD20092, 1647
MD20094, 498, 789
MD20095, 498, 789
MD20096;MD 2 0 0 9 6, 1775, 1776, 1777
MD20100, 1033, 1701, 1704
MD20105, 647, 715
MD20106, 608, 685, 730
MD20107, 608, 686
MD20108, 684
MD20109, 685
MD20110, 417, 705, 991, 992, 993, 995, 1005, 1036, 1771, 1792, 1902, 1952, 1988
MD20110;MD 2 0 1 1 0, 1794
MD20112, 617, 705, 781, 998, 1005, 1036, 1792, 1902
MD20115, 647, 709, 715
MD20116, 710
MD20117, 710, 730
MD20118, 782
MD20120, 781, 1792, 1901
MD20121, 781, 1770, 1792
MD20124;MD 2 0 1 2 4, 555
MD20125, 1923
MD20125;MD 2 0 1 2 5, 1919
MD20126, 1901, 1924
MD20127, 1924
MD20130, 781, 1792
MD20140, 781
MD20144, 783
MD20150, 138, 310, 339, 360, 422, 585, 654, 782, 917, 996, 997, 1036, 1761
MD20150;MD 2 0 1 5 0, 322
MD20152, 782, 996
MD20170, 289
MD20171, 289
MD20172, 289, 1750
MD20173, 289
MD20180, 1890
MD20184, 986, 987, 1892
MD20188, 1874
MD20190, 1874
MD20191, 711
MD20192, 687
MD20193, 688
MD20194, 706, 721
MD20200, 1752
MD20201, 1752
MD20202, 1830
MD20204, 1823
MD20210, 1837
MD20220, 1836
MD20230, 1838
MD20240, 1840, 1842
MD20250;MD 2 0 2 5 0, 1813
MD20252;MD 2 0 2 5 2, 1859
MD20256;MD 2 0 2 5 6, 1851
MD20270, 480, 1772, 1790, 1792
MD20270;MD 2 0 2 7 0, 1988, 1989
MD20272, 481, 1982
MD20272;MD 2 0 2 7 2, 1988, 1989
MD20310, 632
MD20360, 1034, 1787, 1918, 1924, 2011
MD20390, 1971, 1973, 1975
MD20392, 1971
MD20400, 258
MD20430, 259
MD20440, 259
MD20443, 263
MD20450, 260
MD20460, 267, 268
MD20462, 267
MD20465, 271, 272
MD20480, 246, 250, 294
MD20482, 289, 295
MD20484, 289
MD20485, 289
MD20486, 289
MD20487, 289
MD20488, 293
MD20490, 239
MD20500, 323, 325
MD20550, 233
MD20552, 234
MD20560, 304
MD20600, 338
MD20602, 335
MD20602;MD 2 0 6 0 2, 334, 335
MD20606, 283
MD20610, 327
MD20624, 1035
MD20700, 709
MD20700;MD 2 0 7 0 0, 1537, 1538
MD20730, 304
MD20750, 302, 1702
MD20800, 477, 557
MD20850, 1565
MD21015, 72
MD21016, 73
MD21020, 135
MD21158, 364
MD21159, 364

MD21166, 363
MD21168, 363
MD21220, 1742
MD21230, 1742
MD21330, 782
MD22000, 512, 1564
MD22010, 513, 1564
MD22020, 513, 1564
MD22030, 514, 1564
MD22035, 514, 2032
MD22040, 504, 513
MD22050, 505
MD22060, 505
MD22070, 506
MD22080, 506, 803, 2032
MD22100, 524
MD22110, 479, 482
MD22200, 517
MD22210, 478, 517
MD22220, 480, 517
MD22230, 479, 517
MD22240, 482, 517, 1730
MD22250, 480, 517
MD22252, 481, 517
MD22254, 498, 515, 789
MD22256, 498, 515, 789
MD22410, 1730, 1738, 1761
MD22510, 654, 1356
MD22530, 550, 1902
MD22532, 550
MD22534, 550
MD22550, 791, 1770, 1784, 1785, 1790, 1792, 1793
MD22560, 498, 790, 1770, 1792, 1793
MD22560;MD22560, 555
MD22562, 1786, 1787
MD22600, 630
MD22620;MD22620, 652
MD24004;MD24004, 992
MD24006, 892, 917, 993
MD24007;MD24007, 997
MD24008, 892, 992
MD24010, 938
MD24020, 925
MD24040;MD24040, 966
MD24100, 1973
MD24110, 1973
MD24120, 1974
MD24558;MD24558, 2013
MD24570, 1974
MD24572, 1974
MD24658;MD24658, 2013
MD24805, 955
MD24855, 955
MD24905, 948
MD24955, 948
MD25574, 1974
MD26008, 499, 789
MD26012, 789
MD27100, 746, 1036
MD27800, 586
MD27850, 823
MD27860, 671, 823, 824
MD27880, 672, 828
MD27882, 829
MD28060, 741, 744
MD28070, 293
MD28071, 289
MD28072, 289
MD28082, 917, 940, 976, 977, 987, 1892
MD28085, 1773, 1894, 1900
MD28150, 77
MD28200, 183
MD28210, 183
MD28212, 183
MD28400;MD28400, 744
MD28402, 745
MD28530, 74, 248
MD28533, 262
MD28560;MD28560, 999
MD28600, 139
MD28610, 283
MD28620, 1847
MD30100, 431
MD30110, 431
MD30120, 431
MD30130, 430, 432
MD30200, 428, 441
MD30210, 432, 441
MD30220, 432, 441
MD30230, 433, 441
MD30240, 430, 433, 441
MD30242, 430, 434, 435, 441, 1527
MD30244, 441
MD30250, 441, 1533
MD30260, 167, 442
MD30270, 167, 442
MD30300, 441, 443, 447, 450, 1760
MD30310, 132, 441
MD30320, 441
MD30330, 441, 1525
MD30340, 441, 1525
MD30350, 430, 1443
MD30455, 1525, 1654
MD30460, 1033, 1038

MD30550, 1500
MD31000, 443, 447, 450
MD31010, 443
MD31020, 167, 443, 447, 450, 1544
MD31025, 167, 447
MD31030, 168, 441, 443, 447
MD31040, 443, 447, 450, 466, 1576, 1633
MD31044, 437, 443, 1622
MD31046, 149
MD31050, 91, 168, 436, 443, 445, 448, 450, 1622, 1632
MD31060, 91, 168, 436, 443, 445, 447, 450, 1622
MD31064, 437, 441, 1622
MD31066, 437, 441, 1622
MD31070, 168, 443, 447, 450
MD31080, 168, 443, 447, 450
MD31090, 423
MD31122, 1601
MD31123, 1601
MD31200;MD31200, 422
MD31700, 166
MD31710, 166
MD31720, 166
MD31730, 166
MD32000, 102, 304, 405, 440, 441, 1673, 1695
MD32060, 649, 1755
MD32074, 992
MD32100, 439
MD32200, 64, 91, 102, 105, 453
MD32210, 468
MD32220, 468
MD32250, 168, 439
MD32260, 168
MD32300, 102, 110, 321, 362, 1755, 1756
MD32300;MD32300, 332, 333, 334, 335
MD32310, 238, 333
MD32310;MD32310, 333
MD32320, 1757
MD32400, 352, 462
MD32402, 352, 462
MD32410, 352, 463
MD32420, 302, 323, 339, 341, 363
MD32430, 302, 340, 363
MD32431, 338, 346
MD32432, 346
MD32433;MD32433, 332
MD32434, 332, 346
MD32434;MD32434, 331, 332
MD32435, 346
MD32439, 348
MD32440, 267, 271
MD32610, 102
MD32620, 457
MD32630, 458
MD32640, 461
MD32711, 423
MD32750, 1971, 1973, 1975
MD32800, 92, 102
MD32810, 92, 102, 168, 458
MD32890, 464, 465
MD32900, 454
MD32910, 92, 454, 460
MD32950, 467
MD33050, 66
MD33100, 246, 284, 288, 294
MD33120, 253, 295
MD34000, 1514, 1524, 1525
MD34010, 1499, 1509, 1511, 1513, 1522, 1535
MD34020, 1509, 1513
MD34040, 1511, 1513, 1514, 1522, 1523, 1601
MD34050, 1511
MD34060, 1516, 1523, 1525, 1578
MD34070, 1517, 1524
MD34080, 1517
MD34090, 442, 1517, 1519, 1520, 1536
MD34090;MD34090, 1534, 1537, 1538
MD34092, 1514
MD34093, 1515
MD34100, 1517, 1524, 1535
MD34100;MD34100, 1537
MD34102;MD34102, 1526, 1527
MD34104, 1529
MD34110, 1501
MD34200, 1520, 1527, 1529, 1535, 1539, 1601
MD34210, 1531, 1533, 1535, 1536
MD34210;MD34210, 1534, 1537
MD34300, 1523
MD34320, 442, 443, 1519
MD34330, 1540
MD34990, 1696
MD35000, 846, 912, 1647, 1655
MD35010, 1605, 1609, 1626, 1633, 1639
MD35012, 1604, 1633, 1636
MD35014, 1640
MD35020, 1589
MD35030, 1589
MD35035, 1565, 1642, 1645
MD35040, 1560
MD35090, 1604
MD35092, 1609
MD3510, 1638
MD35100, 405, 1671
MD35110, 1604, 1676
MD35112, 1609

MD35120, 1604, 1676
MD35122, 1609
MD35130, 92, 1604, 1610, 1612, 1669
MD35135, 1604
MD35140, 1604, 1610, 1612, 1668, 1703, 1705
MD35150, 1512, 1567, 1569, 1667, 1671
MD35200, 1559, 1570, 1572, 1577, 1604, 1637
MD35210, 1560, 1570, 1573, 1578, 1579, 1604, 1637
MD35212, 1609
MD35220, 354, 358
MD35230, 354, 358
MD35240, 248, 358
MD35242, 358
MD35300, 1544, 1569, 1572, 1578, 1579, 1604, 1637, 1654
MD35310, 1605, 1633, 1636
MD35350, 1578, 1636
MD35400, 1304, 1625, 1627, 1628
MD35410, 1625, 1627, 1628
MD35430, 1625, 1627
MD35500, 1561, 1567
MD35510, 1567, 1666
MD35550, 1605, 1610
MD35590, 64, 91, 1606, 1620
MD36000, 231, 1575, 1580
MD36000;ＭＤ３６０００, 1674, 1675
MD36010, 105, 231, 1574, 1580
MD36010;ＭＤ３６０１０, 1674, 1675
MD36012, 107, 233
MD36020, 105
MD36030, 106, 107
MD36040, 106, 400
MD36042, 400
MD36050, 107, 108, 117
MD36051, 108
MD36052, 109, 110, 112, 113, 116
MD36060, 64, 65, 1561, 1568, 1666
MD36100, 131
MD36110, 131
MD36120, 131
MD36130, 131
MD36200, 119
MD36210, 118, 440
MD36220, 119
MD36300, 123, 1543, 1544
MD36302, 1572
MD36310, 121, 125
MD36312, 126
MD36400, 102
MD36500, 429
MD36510, 429
MD36600, 131
MD36610, 62, 103, 106, 107, 110, 119, 120, 124, 126, 1024
MD36620, 62, 1024
MD37002, 385
MD37010, 380
MD37012, 385
MD37020, 381, 393
MD37030, 380
MD37040, 380
MD37060, 379, 383, 385
MD37080, 398
MD51029, 740
MD51041, 1122, 1389
MD51043, 1122, 1389
MD51074, 829
MD51230, 1381
MD51231, 1381
MD51232, 1381
MD9004, 409, 747
MD9006, 52
MD9010, 747
MD9011, 409, 747
MD9106, 1122, 1389
MD9424, 748, 887
MD9440, 1004
MTL, 790
M 機能の置換, 787

N

nact, 70
NC
　スタート, 660
　言語範囲, 655
　故障, 1087
　-変数の読み取り/書き込み, 1091
NC CPU 準備完了, 47
NC VAR セレクタ, 1150
　入力選択, 1154
NC VAR セレクタ;ＮＣ ＶＡＲセレクタ
　セットップ、インストール;セットップ、インストール, 1162
NC/PLC インタフェース, 585
NCK PLC の起動と同期, 1082
NC アラーム未処置, 48
NC バッテリアラーム, 48
NC 準備完了, 48
NORM, 1814
NPROT, 188
NPROTDEF, 184

O

ORISOLH, 1927
OTOL, 295
OVR, 1737
OVRA, 1654, 1737

P

PCOF, 1653
PLC
- 変数の読み取り/書き込み, 74
HMI モニタ, 1138, 1405
キーロック, 1125
バージョン, 1045
プログラムリスト, 1120, 1122, 1389, 1474
基本プログラム機能, 1045
-軸, 857
PLC を使用する主軸機能, 586
POINTER;ＰＯＩＮＴＥＲ, 1358
POLF, 1713
POLFMASK, 1713
POLFMLIN, 1713
POS, 857
POS;ＰＯＳ, 856
POSA, 857
POSA;ＰＯＳＡ, 856
PRESETON, 872, 1502
PRESETONS, 876
Process DataShare, 767, 772
PROFIBUS
接続, 1109
PROFIBUS 診断, 1086
PROFIBUS 接続, 1110
PROFINET
接続, 1113

R

REPEAT, 673
REPEATB, 673
REPOS オフセット, 389
RESET
命令, 664

S

S..., 1652
SAVE, 714
SBL2 のデバッグモード, 730
SBLOF, 731
SBLON, 731
SCC, 1036
SCPARA, 1759
SD41100, 1699
SD42010, 1709, 1712
SD42100, 595, 1739
SD42101, 1740
SD42200, 731
SD42440, 852, 1951
SD42442, 1951
SD42444, 603
SD42465, 246, 295
SD42466, 246, 295
SD42470, 289
SD42471, 289
SD42472, 290
SD42473, 290
SD42475, 290
SD42476, 290
SD42477, 290
SD42480;ＳＤ４２４８０, 1850
SD42496;ＳＤ４２４９６, 1850
SD42500, 327
SD42500;ＳＤ４２５００, 327
SD42502, 327
SD42502;ＳＤ４２５０２, 327
SD42510, 341
SD42510;ＳＤ４２５１０, 341
SD42512, 341
SD42512;ＳＤ４２５１２, 341
SD42600, 1699
SD42600;ＳＤ４２６００, 1663
SD42676, 295
SD42678, 295
SD42700, 754
SD42750, 743
SD42800, 1647
SD42900, 1768, 1991
SD42900;ＳＤ４２９００, 1967
SD42910, 1991, 1997, 1998
SD42910;ＳＤ４２９１０, 1967
SD42920, 1991, 1997, 1998
SD42920;ＳＤ４２９２０, 1968
SD42930, 1991, 1997
SD42930;ＳＤ４２９３０, 1969
SD42935, 1909, 1992, 1998, 2033
SD42940, 1955, 1991, 1997
SD42950, 1914, 1956, 1991, 1997
SD42954, 1958
SD42956, 1958

SD42960, 1768, 1972, 1973
SD42974, 1874
SD42977, 1847
SD42984, 1945
SD42990;ＳＤ４２９９０, 741
SD42998, 1917, 1926
SD42999, 1926
SD43200, 1659, 1661
SD43202, 1659, 1661
SD43206, 1660, 1663
SD43210, 1703, 1705
SD43220, 1703
SD43230, 1703
SD43235, 1672
SD43240, 1564, 1653
SD43250, 1564
SD43300, 1699
SD43300;ＳＤ４３３００, 1663
SD43400, 137
SD43410, 137
SD43420, 137
SD43430, 137
SD43500, 393
SD43510, 393
SD43520, 393
SERUPRO
    -end ASUP, 542
    プログラム可能な中断ポインタ, 637
    自動中断ポインタ, 639
SERUPRO ASUB, 388
    特記事項, 632
seruproMasterChan, 635
SERUPRO アプローチ
    PLC からの制御, 624
SETINT, 701, 712
SETMS, 1651
SETTCOR, 2013
SOFT, 339
SOFTA, 340
SPCOF, 1653
SPCON, 1653
SPI, 1655
SPOS, 1562, 1652
SPOSA, 1652
STOLF, 308
STRINGIS, 656
SVC, 1652
S 機能, 478

T

TCARR, 1891
TCOABS, 1891
TCOFR, 1891
TCP 工具中心点, 865
TEACH IN, 577
TOA
    -ユニット, 1773
TOA;ＴＯＡ
    - データ;ー データ, 1800
TOOLENV, 1999
TOWBCS, 1996
TOWKCS, 1997
TOWMCS, 1995
TOWSTD, 1995
TOWTCS, 1996
TOWWCS, 1995
TRANS, 849
TRANSMIT, 883
T 機能, 479, 1770
T 機能の置換, 790

U

UDT 割り当て, 1092
UTD ブロック;ＵＴＤブロック, 1092

V

VELOLIM, 1654

W

WAITENC, 784
WAITS, 1654
WALCS0, 141
WALIMOF, 138
WALIMON, 138
WCS / MCS での現在位置, 53

X

XE * MERGEFORMAT,

ア

アクションシングルブロック, 592
アクションブロック, 602
アクセスセキュリティ;アクセスセキュリティ, 78
アクセス権;アクセスケン, 78
アクセス保護の特徴;アクセスホゴノトクチョウ, 80
アダプタ寸法;アダプタスンポウ, 1808

イ

イグザクトストップ, 230
　自動, 236
イグザクトストップ条件, 231, 232
イベント起動プログラムシーケンス, 679
インクリメンタルでプログラム指令された補正値, 1951
インターフェース
　MCP/PLC と HHU/PLC, 1063
　PLC/HMI, 1063
インタフェース
　PLC/HMI, 1073
　PLC/HMI メッセージ, 1073
　PLC/MCP, 1078
　PLC/NCK, 1064
インバースタイム送り(G93), 1696

エ

エンコーダデータセットの切替え, 165
エンコーダ監視機能, 123
　エンコーダ周波数, 123

オ

オフセット番号, 1778
オペレーティングソフトウェア準備完了, 47

カ

カウンタパルス, 830

キ

キーロック, 52, 1125
キー操作スイッチ, 82
キネマティック座標変換, 883
キネマテックタイプ, 1864
キャンセルアラーム, 52
ギヤ選択, 1604
　M70 の, 1639
　PLC による指定, 1612
　手動入力, 1613
ギヤ選択の指定, 1606
ギヤ選択切替
　自動, 1676

コ

コントローライネーブル, 60
コントロールシステムの動作
　パートプログラム起動時, 778
　パートプログラム終了時, 778
　リセット時:, 778
　起動中, 778

サ

サーボゲイン係数(Kv), 453
サイクリック運転, 1082
サイクリック信号交信, 46

シ

ジオメトリ軸, 883
ジオメトリ軸;ジオメトリジク, 855, 856
シミュレーション, 598
シミュレーション軸, 430
ジャンプマーク
　プログラム区間の繰り返し, 673
ジョグ
　モードタイプ AUTOMATIC での, 579
シングルブロック
　プログラム運転モード, 591
　マスク, 731
シンボルプログラミング;シンボルプログラミング, 1092

ス

スキップレベル, 739
スター/デルタ切り替え, 1297
スプライン, 229
スムージング
　軌跡速度, 266

セ

ゼロオフセット
　外部ゼロオフセット, 889
ゼロベクトル, 1869

ソ

ソフトウェアリミットスイッチ, 131

タ

ターゲットブロックでの特別な注意点
STOPRE ブロック, 641
ダイナミック応答
補正, 269
ダイナミック応答調整, 454
ダイレクトキー
Ethernet バスにある OP, 1106
アドレス, 1107
アラーム, 1107
タッピング
フローティングチャックあり, 1597

チ

チャネル
基本設定, 654
状態, 667
状態表示, 667
設定, 585
電流, 689
特性, 584
チャネル軸;チャネルジク, 855
チャネル状態
チャネルリセット, 579
チャネル起動中, 579
チャネル中断, 579
チャネル別 NC アラームが未処置, 49

テ

データインタフェース, 1065
データチャネル、高速, 74
データ交信
操作パネルと, 1061
デコードシングルブロック, 592

ト

ドライブがサイクリック運転, 48
ドライブテスト移動許可, 54
ドライブテスト移動要求, 65
ドライブ準備完了, 48
ドライラン送り速度, 595, 1738

ね

ねじ
切り取り;キリトリ, 1713
ねじ切り G33, 1705

ハ

パーキング, 145, 1502
ハードウェアリミットスイッチ, 130
ハードウェア割り込み;ハードウェアワリコミ, 1086
パートプログラム
選択, 660
特定のパートプログラムブロックのスキップ, 597
パス
送り速度 F, 1694
速度、最高, 1695
パスワード, 80, 81
パスワードの初期設定, 81
パッケージ
-カウンタ, 538
パラメータセット
軸の, 456
旋回工具ホルダ, 1864

フ

ファームウェア, 1371
ファインインタポレーション, 453
フィードバック値の取得, 428
フィードバック値の処理, 439
フィードバック値管理;フィードバックチカンリワーク関連;ワークカンレン, 1000
フィードバック値更新, 430
フォローアップ有効, 65
フラット D 番号の構成, 1788
フレーム回転, 851
工具方向, 984, 986
立体角付き, 981
フレーム変更, 1862
プログラミング装置
ハードウェアの必要条件, 1147
プログラム
PLC による選択, 1122, 1389
ステータス, 665
プログラムテスト, 589
実行時間, 817
状態表示, 665
動作, 668

プログラムストップを伴う NC アラーム, 49
プログラム運転, 653
プログラム区間
　繰り返し, 673
プログラム制御、インタフェース信号, 739
プログラム表示モード, 743
ブロック
　非表示, 739
ブロックサーチ SERUPRO;ブロックサーチＳＥＲＵＰＲＯ
　重畳動作;チョウジョウドウサ, 651
ブロック検索
　プログラムテストモードにおける計算あり、SERUPRO (タイプ 5), 600
　ブロック終点における計算あり(タイプ 4), 600
　計算なしの(タイプ 1), 600
　多段, 601
　輪郭における計算あり(タイプ 2), 600
ブロック検索 SERUPRO
　NC/PLC インタフェース信号による REPOS の制御, 624
　REPOS 応答;ＲＥＰＯＳオウトウ, 625
　REPOS 応答の設定, 620
　ギヤ選択切替, 650
　位置決め軸の再位置決め, 622
　基本設定, 652
　軌跡軸, 623
　指令値とフィードバック値の連結, 644
　軸機能の条件, 648
　軸入れ替え後の REPOS オフセット, 628
　主軸同期連結のある REPOS オフセット, 628
　動作順序, 616
　有効範囲内の REPOS オフセット, 627
ブロック単位の制限(FOC), 398
プロテクションゾーン, 175, 184
　制限事項, 212

ヘ

ヘリカル補間, 860

ホ

ホルダ角度, 1916

マ

マルチインスタンス DB, 1364
マルチ工具, 1227

ミ

ミラーリング;ミラーリング　フレーム;フレーム, 911

メ

メインラン(Main run), 584
　-軸, 1757
メイン軸, 858

モ

モータ/負荷側ギヤ, 436
モータ/負荷側減速比の設定, 436
モーダル起動(FOCON/FOCOF), 398
モード
　AUTOMATIC, 576
　AUTOMATIC での JOG, 576
　JOG, 577
　MDI, 577
　モードグループの, 576
　モード間シンクロナイズドアクション, 577
　変更, 576, 583
　優先順位, 577
モードグループ, 572, 1083
モード変更
　AUTOMATIC、JOG、MDI の各モードへ/からの, 583

ユ

ユーザー定義の ASUP
　SERUPRO 動作後の, 618

ラ

ラダーアドオンツール, 1380
ラダーエディタ, 1379
ラダー図, 1379
ラベル, 673

リ

リコールアラーム, 52
リストに応じた M デコード, 1094
リセット
　動作, 778

リミットスイッチ監視, 130
リモート診断, 48

レ

レファレンス点, 864
レファレンス点 R, 864
レファレンス点復帰
　インクリメンタル検出器による, 1507
　ロータリー絶対値エンコーダ内, 1540
レファレンス点復帰動作の方法, 1497

ロ

ローダ軸;ローダジク, 856
ロック期日, 1476

ワ

ワーク
　-カウンタ, 827, 832
　-シミュレーション, 598
ワーク原点 W, 864
ワーク座標系(WCS), 848
ワーク座標系(WCS);ワークザヒョウケイ（ＷＣＳ）, 889

位

位相フィルタ, 464
位置コントローラ有効, 65
位置決め軸, 407, 857
位置決め精度, 408
位置検出器, 59
位置制御ループ, 452

移

移動動作許可, 1539
移動範囲, 407

一

一致確認, 830
一定曲率, 1815
一定接線, 1815

運

運転モード;ウンテンモード
　インタロック;インタロック, 582
　監視機能;カンシキノウ, 582
運転状態, 579

加

加々速度
　-増加、速度依存加々速度, 347
加々速度制限, 251
加工停止を伴う NC アラームが未処置, 49

過

過負荷係数, 238

画

画面消去, 51
画面表示, 51

解

解析, 584

回

回転の記述, 1874
回転軸;カイテンジク, 856
回転軸パラメータ, 1864
回転成分, 978

外

外部サブプログラムの実行, 752
外部ゼロオフセット, 889
外部プログラムメモリ, 751

角

角度
　ホルダ, 1916
　取り付けスペース, 1916

割

割り込み
　lock, 714
　ルーチン, 700
　信号, 701
割り込みルーチン
　終了, 703
割線エラー, 284

滑

滑らかなアプローチと後退;ナメラカナアプローチトコウタイ
　意味;イミ, 1819
　動作区分;ドウサクブン, 1819

関

関連規格, 1021

丸

丸み付け, 240

基

基準軸
　G96 / G961 / G962 の, 1029
基本システム, 414
基本ブロック表示
　設定, 744
　有効化, 744
基本座標系(BCS), 848, 883
基本表示;キホンヒョウジ
　表示バッファのサイズ, 744

機

機械のキネマティックス, 1864
機械原点, 1497
機械原点 M, 864
機械座標系(MCS), 847, 870
機械軸;キカイジク, 854
機能インタフェース, 1066

気

気温アラーム, 48

起

起動
　機械操作パネル、ハンドヘルドユニットから, 1104
起動完了, 70

軌

軌跡軸;キセキジク, 856
軌跡条件, 242
軌跡送り速度, 407

許

許容範囲
　G0 による, 303
　-主軸速度の係数, 1667

曲

曲率, 281

径

径方向軸, 1029
　寸法, 1037

計

計算ありのブロック検索
　収集された主軸機能, 606
計算分解能, 409

検

検出器, 429

原

原点, 864
原点マーク診断, 125

現

現在値の同期, 62

限

限界値
速度、軌跡軸の, 1695

個

個々の摩耗成分の評価, 1997

固

固定送り速度, 1748

後

後退
ねじ切りの方向, 1714

向

向き
許容範囲, 294

工

工具, 1770
2 次元の工具径補正(TRC), 1812
DISC, 1835
M06 による工具交換, 1770
T 機能, 1771
管理, 572
-後退, 783
交換, 1770
工具刃先, 1771
選択, 1770
工具(Tool)
パラメータ, 1795
形状, 1805
刃先, 1795
工具/工具ホルダの割り当て, 1861
工具;コウグ
- 長さ;ー ナガサ, 1803
-オフセットデータ;ーオフセットデータ, 1800
-サイズ、有効;ーサイズ、ユウコウ, 1807
-タイプ;ータイプ, 1797
-形状、有効;ーケイジョウ、ユウコウ, 1807
工具ベース寸法/アダプタ寸法;コウグベーススンポウ／アダプタスンポウ, 1808
-摩耗;ーマモウ, 1807
工具オフセット;コウグオフセット
NC でのオフセット;ＮＣデノオフセット, 1775
工具キャリア、方向付け機能装備
新規作成, 1894
工具に直接接続されるエンコーダ, 437
工具ベース寸法;コウグベーススンポウ, 1808
工具ホルダ、旋回, 1860
プログラミング, 1899
リセット、プログラム開始、REPOS 時のシステム応答, 1901
傾斜面加工, 1888
作業テーブルの旋回, 1890
必要条件, 1901
工具ホルダデータへの角度の保存, 1863
工具ホルダの向きの制限, 1863
工具ホルダレファレンス点 T, 865
工具ホルダ選択, 1860
工具マガジン軸;コウグマガジンジク, 856
工具リボルバ軸;コウグリボルバジク, 856
工具径補正
継続;ケイゾク, 1855
工具径補正、2 次元, 1811
アプローチおよび後退動作, 1814
ジオメトリ, 1805
外側コーナでの補正, 1834
交点 G451, 1837
遷移円, 1835
工具径補正、2 次元;コウグケイホセイ、２ジゲン
可変補正値;カヘンホセイチ, 1848
解除;カイジョ, 1834
滑らかなアプローチと後退;ナメラカナアプローチトコウタイ, 1819
選択;センタク, 1813
変更されたアラーム応答;ヘンコウサレタアラームオウトウ, 1850
工具径補正;コウグケイホセイ
摩耗;マモウ, 1807
工具交換
D 機能, 1771
オフセットメモリ, 1773
工具交換時の刃先の選択, 1772
工具長補正
ワーク別の計算, 1989
工具長補正(Tool length compensation)
工具別の計算、例, 1908

工具長補正;コウグチョウホセイ
　ジオメトリ;ジオメトリ, 1803
　摩耗;マモウ, 1807
工具点
　方向, 1921
工具補正
　タイプ, 1863
工具方向での加工;コウグホウコウデノカコウ, 1952

溝

溝フライス;ミゾフライス, 1800

荒

荒削りオフセット, 849

高

高速データチャネル, 74

座

座標系およびレファレンス点の位置;ザヒョウケイオヨビレファレンステンノイチ, 866

最

最小入力単位, 409
最小表示単位, 409

作

作業領域リミット, 134
　BCS 内, 136
　WCS/SZS 内, 139
作業領域リミットグループ, 139

残

残り時間
　加工の, 820
残移動量削除;ザンイドウリョウサクジョ, 54

仕

仕上げオフセット, 849

指

指令回路, 428
指令値出力, 428
指令値出力のないプログラム実行, 590

自

自動 SERUPRO, 635
自動イグザクトストップ, 236
自動的な先読み停止, 642
自動的な連続軌跡モード, 239
自由曲面, 282
　モード, 228, 282

軸

軸/主軸停止, 65
軸ねじれ, 281
軸監視機能
　フィードバック速度, 119
　速度指令値, 118
　追従誤差, 102
　停止状態, 106
　必要条件, 168
軸設定, 861
軸無効, 54, 1732

実

実行時間
　プログラム, 817

主

主軸
　-オーバライド, 1732, 1735
　-ギヤ選択 0, 1616
　-指令速度, 1667
　-速度、最大, 1668
　-速度、最低, 1668
　速度制限, 1669
主軸速度, 407
主軸無効, 54, 1732

周

周速一定制御, 1700

出

出力
-カウンタ, 538
外部機器/ファイルへの, 772
処理, 538
補助機能の動作, 506

潤

潤滑パルス, 66

処

処理時間, 822

衝

衝突監視, 1841
衝突検出, 1843

信

信号
NCK/PLC, 1068
PLC/NCK, 1068
PLC/NCK チャネル, 1070
PLC/モードグループ, 1069
PLC/軸、主軸, 1071
アラーム信号, 48
コンパイルサイクル, 1067
チャネル別(DB21、...), 46
軸/主軸別(DB31、...), 47
信号ひずみ, 101
信号交信
イベント起動, 1043
サイクリック, 1043

刃

刃先
位置, 1915
-基準点, 1919
中心点, 1919
刃先位置;ハサキイチ, 1801
関連する;カンレンスル, 1811
刃先番号, 1778

垂

垂直軸, 392

制

制御パラメータセットの切り替え, 90
制御方向, 439

切

切削方向, 1915

先

先読み, 255
選択と解除, 257
先読み停止アラーム;サキヨミテイシアラーム, 1850

旋

旋回ワークを備える機械, 1871
旋回工具ホルダ, 1768, 1860
キネマティックチェーン, 1868
有効な工具長の計算, 1890
例, 1904
旋回工具を備える機械, 1870

線

線形信号ひずみ, 101

早

早送り
-オーバライド, 1732
補間タイプ, 301

送

送り速度
1 刃当り, 1698
インバースタイム(G93), 1696
-オーバライド, 1732
タイプ, 1693
タイプ(G93、G94、G95), 1696
ドライラン送り速度, 1738

軌跡送り速度 F, 1694
送り速度/主軸停止, 1730
送り速度無効, 1730
毎回転送り(G95), 1697
毎分送り(G94), 1697
面取り/丸み付けの, 1751
送り速度オーバライド
プログラム可能な, 1737
送り速度無効
チャネル別, 1731

測

測定システム, 1497

速

速度, 405
フィードフォワード制御, 457
速度コントローラ有効, 66
速度指令値の調整, 439
速度指令値出力, 439
速度制御ループ, 452

多

多項式、交点処理, 1851
多段ブロック検索, 609

単

単位系, 414, 423
単独軸のダイナミック応答, 1755

置

置換サブプログラム, 786
置換原点マーク, 1540

中

中間ギヤ, 437, 1622

直

直径指定
チャネル別, 1037
軸別, 1038

停

停止イベント, 693
停止遅延領域, 692

定

定速度, 1702

適

適用された位置の差, 466

電

電流コントローラ有効, 66

度

度数, 538

逃

逃げ角, 1916
逃げ角;ニゲカク, 1809

動

動作終了条件
単独軸の場合, 1758

同

同期軸, 859
同時軸, 1089

独

独立一軸運転, 649

凸

凸型ねじ, 1718

突

突き当て点停止, 375
　ブロック検索, 387
　位置決め軸, 392
　解除, 384
　監視範囲, 381
　機能の中止, 391
　軌跡誤差監視, 392
　選択, 379
　突き当て点に達しました。, 380

半

半径対応データ, 1032

非

非常停止
　インタフェース, 1023
　応答, 1026
　処理, 1024
非常停止制御部品, 1022
非線形信号ひずみ, 101

必

必要メモリ
　PLC 基本プログラムの, 1142

表

表示ブロック、構成(DIN), 747

付

付加軸;フカジク, 856

負

負のアドレス拡張, 558

複

複合旋回機構を備える機械, 1872
複数の径方向軸
　軸入れ替え, 1038

物

物理量, 411

文

文字列;モジレツ, 1365

閉

閉ループ制御, 452

保

保護できるデータエリア;ホゴデキルデータエリア, 83
保護レベル
　ユーザー ASUB 用, 718
保護レベル;ホゴレベル, 78
　設定可能;セッテイカノウ, 83

補

補間
　G0 による, 301
　直線, 301
　非直線, 302
補間軸のグループ, 62
補間終了, 232
補助機能
　アドレス拡張, 505
　-カウンタ, 538
　タイプ, 505
　ユーザー固有, 512
　ユーザ定義の, 475
　関連, 515
　事前定義, 475, 483
　出力動作, 506
　値, 506
　定義, 476
補助機能のグループ化, 506
補助機能出力, 654
補正係数
　ダイナミック軌跡応答, 271

方

方向付け機能を備えた工具キャリア
　新規作成, 1894

毎

毎回転送り(G95), 1697
毎分送り(G94), 1697

未

未応答のギヤ選択切替, 1624

予

予測輪郭演算, 1840

輪

輪郭
　サンプリング係数, 284
　サンプリング時間, 284
　許容範囲, 294
輪郭誤差, 101

連

連続軌跡モード, 235
　自動, 239