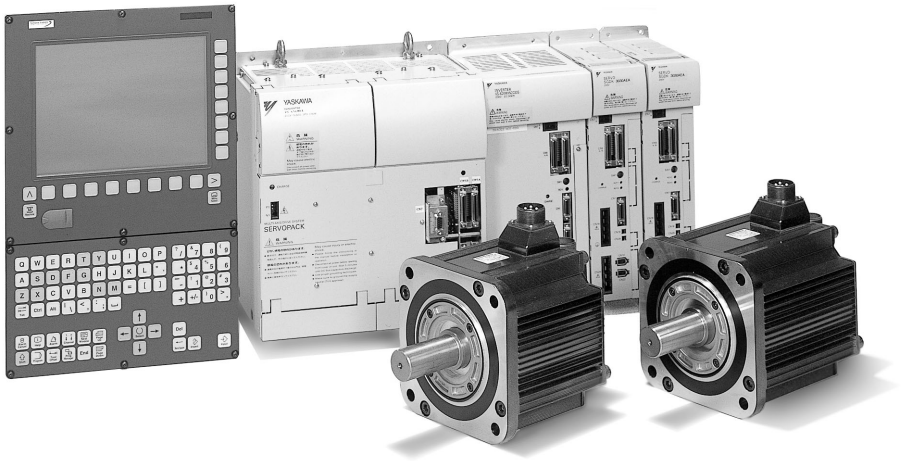
Yaskawa Siemens
numerical controls

Yaskawa Siemens CNC シリーズ

結合説明書・機能編（暫定版）
基本編（Part2）



資料番号　DE0400309-2

安川シーメンス NC 株式会社はシーメンス株式会社に統合の後，2010 年 8 月より
シーメンス・ジャパン株式会社へ社名を変更いたしました。本書に記載の「安川
シーメンス NC 株式会社」などの社名に類する名称は「シーメンス・ジャパン株式
会社」へ読み替えをお願いします。

本マニュアルは Yaskawa Siemens 840DI，Yaskawa Siemens 830DI 両モデル用に
作成されています。本文中の記述では両モデルの機能差は区別されておりませんの
で，それぞれのモデルにどの機能が標準装備されているか，どの機能がオプション
で装備可能かについては別途，機能一覧表をご参照ください。また，本文中に
840DI と言った表現が出て来ますが，830DI も意味していることがあるとご理解
ください。

1.1 各種インタフェース信号 (A2) - 1-1
1.1.1 概略説明 - - - - - - - - - - - - - - - - - - -

1.6.2 詳細説明 ---------- 1-250
1.6.3 補足条件 ---------- 1-277
1.6.4 データの説明 (MD，SD) ---------- 1-278
1.6.5 信号の説明 ---------- 1-278
1.6.6 例 ---------- 1-278
1.6.7 データフィールド，リスト ---------- 1-279
1.7 突き当て停止 (F1) ---------- 1-281
1.7.1 概略説明 ---------- 1-281
1.7.2 詳細説明 ---------- 1-281
1.7.3 補足条件 ---------- 1-293
1.7.4 データの説明 (MD，SD) ---------- 1-293
1.7.5 信号の説明 ---------- 1-299
1.7.6 例例例例 ---------- 1-302
1.7.7 データフィールド，リスト ---------- 1-302
1.8 位置指令単位と検出単位 (G2) ---------- 1-305
1.8.1 概略説明 ---------- 1-305
1.8.2 詳細説明 ---------- 1-306
1.8.3 補足条件 ---------- 1-344
1.8.4 データの説明 (MD，SD) ---------- 1-344
1.8.5 信号の説明 ---------- 1-369
1.8.6 例 ---------- 1-369
1.8.7 データフィールド，リスト ---------- 1-369
1.9 補助機能出力 (H2) ---------- 1-373
1.9.1 概略説明 ---------- 1-373
1.9.2 詳細説明 ---------- 1-375
1.9.3 補足条件 ---------- 1-391
1.9.4 データの説明 (MD，SD) ---------- 1-392
1.9.5 信号の説明 ---------- 1-400
1.9.6 例 ---------- 1-406
1.9.7 データフィールド，リスト ---------- 1-411
1.10 運転モードの種類とプログラム運転モード (K1) ------ 1-417
1.10.1 概略説明 ---------- 1-417
1.10.2 詳細説明 ---------- 1-418
1.10.3 補足条件 ---------- 1-466
1.10.4 データの説明 (MD，SD) ---------- 1-466
1.10.5 信号の説明 ---------- 1-488
1.10.6 例 ---------- 1-511
1.10.7 データフィールド，リスト ---------- 1-515
1.11 軸構成及び座標系（K2） ---------- 1-525
1.11.1 概略説明 ---------- 1-525
1.11.2 詳細説明 ---------- 1-532
1.11.3 補足条件 ---------- 1-585

1.11.4 データの説明 (MD，SD)- 1-586
1.11.5 信号の説明 - 1-608
1.11.6 例 - 1-610
1.11.7 データフィールド，リスト- 1-613
1.12 通信機能 (K4)- 1-617
1.12.1 概略説明- 1-617
1.12.2 詳細説明- 1-618
1.12.3 補足条件- 1-623
1.12.4 データの説明 (MD, SD) - 1-624
1.12.5 信号の説明 - 1-629
1.12.6 例- 1-631
1.12.7 データフィールド，リスト- 1-634
1.13 非常停止入力 (N2) - 1-635
1.13.1 概略説明- 1-635
1.13.2 詳細説明- 1-636
1.13.3 補足条件- 1-641
1.13.4 データの説明 (MD, SD) - 1-641
1.13.5 信号の説明 - 1-641
1.13.6 例- 1-643
1.13.7 データフィールド，リスト- 1-643
1.14 直径指定／半径指定 (P1)- 1-645
1.14.1 概略説明- 1-645
1.14.2 詳細説明- 1-646
1.14.3 補足条件- 1-649
1.14.4 データの説明 (MD, SD) - 1-649
1.14.5 信号の説明 - 1-650
1.14.6 例- 1-650
1.14.7 データフィールド，リスト- 1-650
1.15 基本的 PLC プログラム (P3) - 1-651
1.15.1 概略説明- 1-651
1.15.2 詳細説明- 1-653
1.15.3 補足条件 および NC VAR セレクタ - 1-703
1.15.4 ブロック説明- 1-719
1.15.5 信号／データの説明- 1-836
1.15.6 データフィールド，リスト- 1-842
1.15.7 STEP7 でのプログラミングにおける役立つポイント - - - - - - - - - 1-848
1.15.8 マシンデータ，アラーム- 1-856
1.16 原点復帰 (R1)- 1-859
1.16.1 概略説明- 1-859
1.16.2 詳細説明- 1-860
1.16.3 補足条件- 1-885
1.16.4 データの説明 (MD, SD) - 1-889

1.1.5 信号の説明 ---------- 1-908
1.1.6 例---------- 1-912
1.1.7 データフィールド、リスト---------- 1-912
1.17 主軸機能 (S1)---------- 1-915
1.17.1 概略説明---------- 1-915
1.17.2 詳細説明---------- 1-916
1.17.3 補足条件---------- 1-954
1.17.4 データの説明 (MD，SD)---------- 1-955
1.17.5 信号の説明 ---------- 1-974
1.17.6 例---------- 1-988
1.17.7 データフィールド，リスト---------- 1-988
1.18 送り機能 (V1)---------- 1-993
1.18.1 概略説明---------- 1-993
1.18.2 詳細説明---------- 1-995
1.18.3 補足条件---------- 1-1031
1.18.4 データの説明 (MD，SD)---------- 1-1031
1.18.5 信号の説明 ---------- 1-1043
1.18.6 例---------- 1-1057
1.18.7 データフィールド，リスト---------- 1-1057
1.19 工具補正 (W1) ---------- 1-1061
1.19.1 概要 ---------- 1-1061
1.19.2 詳細説明---------- 1-1064
1.19.3 補足条件---------- 1-1163
1.19.4 データの説明 (MD，SD)---------- 1-1164
1.19.5 信号の説明 ---------- 1-1182
1.19.6 例---------- 1-1182
1.19.7 データフィールド，リスト---------- 1-1183

1.12 通信機能 (K4)

1.12.1 概略説明

通信方法

制御装置には，以下の通信機器が提供されます（インストールされている MMC モジュールによって異なります）：

MMC 100 (コネクタ X 6)，MMC 101 ～ 103 (コネクタ X 6，X7)
　にはシリアルインタフェース (RS 232)

MMC 101 ～ 103 (コネクタ X 8)
　にはパラレルインタフェース（セントロニクス）

MMC 101 ～ 103 にはディスク装置用の接続

ISA アダプタ（オプション）
　にはネットワークカード（AT タイプ）用の接続

シリアル

データのインポート，エクスポートをする際，PCIN などのデータ転送プログラムを使って，プログラミング装置または PC とデータのやりとりするために，シリアルインタフェースが標準装備されています。

　例 : マシンデータのパートプログラムをインポートする。

パラレル

パラレルインタフェースは，主としてプリンタへの接続装置に使用されます。
データのバックアップ用に，ストリーマも接続できます（OEM_ アプリケーションなど)。

ディスクドライブ

MMC101-103 では，3.5 インチ用のディスク装置へ接続することができるので，たとえば，マシンと生産計画との間でデータのやりとりが容易に行えます。

ネットワークカード

ISA 拡張ボックスには，Ethernet 用などのネットワークカードを取り付けることができます。こうしたカードを使用すると，たとえば，統括コンピュータから制御装置へ直接プログラムを読み込むすることができます。ただし，このようなカードを使用する場合には，MMC の OEM 設定処理が必要となります。

カードの機能性はシステムの不可欠の要素ではありません。

　（注）OEM_ ユーザに限られます。

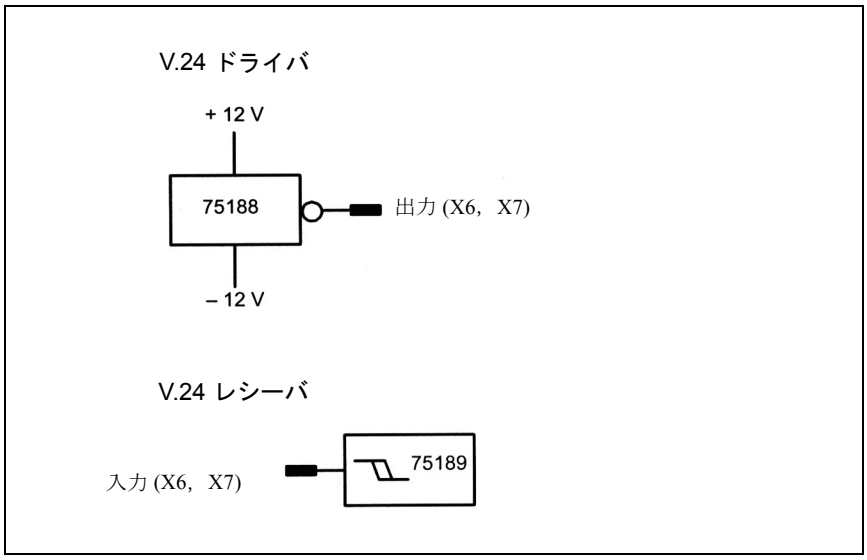
1.12.2 詳細説明

■ シリアルインタフェース (RS232)

シリアルインタフェースは，MMC モジュールの以下のコネクタで使用できます。

・MMC 100	シリアルインタフェース (RS 232)	コネクタ X 6
・MMC 101 ～ 103	第 1 シリアルインタフェース (RS 232)	コネクタ X 6
	第 2 シリアルインタフェース (RS 232)	コネクタ X 7
	パラレルインタフェース（セントロニクス）	コネクタ X 8

接続



最大長さ

RS 232 のリード線の接続可能な長さは最大 30 m です。

■ MMC 100

MMC 100 のインタフェース X6 は，次の 3 つのパラメータセットを介してセットできます。

- ユーザ
- プリンタ
- PG/PC

これらのパラメータセットは，SERVICES-DATA-RS232-SET とオペレータインタフェースで入力します。
プリセット値は，各端子に合わせてセットすることができます。

パラメータ変更に必要なオペレータ入力は，以下で説明されています。

参照： YS840DI ユーザーズマニュアル・操作編

ファイルのタイプ

MMC 100 のシリアルインタフェースは，以下のファイルタイプを転送するのに使用されます。

略語／拡張子	ファイルタイプ	略語の意味
BIN	バイナリファイル	Binary Files（バイナリファイル）
BOT	ドライブ ブートファイル	BOOT Files（ブートファイル）
COM	コメントファイル	Comment File（コメントファイル）
CPA	コンパイラ設定データ	Compiler Projecting Data （コンパイラプロジェクトデータ）
DIR	ディレクトリ	Directory（ディレクトリ）
GIA	ギア補間データ	Gear Interpolation Data（ギア補間データ）
GUD	ユーザデータ（グローバル）	Global User Data（グローバルユーザデータ）
IKA	補間補償	Interpolative Compensation（補間補償）
INI	初期化データ	Initializing Data（初期化データ）
LUD	ユーザデータ（ローカル）	Local User Data（ローカルユーザデータ）
MPF	パートプログラム	Main Program File （メインプログラムファイル）
OPT	オプション	Options（オプション）
RPA	値割り当てのある R パラメータ	R-Parameter Active （R パラメータアクティブ）
SEA	値割り当てのあるアドレス	Setting Data Active（設定データアクティブ）
SPF	サブプログラム	Sub Program File（サブプログラムファイル）
SYF	システムファイル	System Files（システムファイル）
TEA	NC マシンデータ	Testing Data Active （テストデータアクティブ）
UFR	ゼロオフセット	User Frame（ユーザフレーム）
WPD	ワークピースディレクトリ	Work Piece Directory （ワークピースディレクトリ）

パラメータ設定

以下のパラメータは変更可能です。

デバイスタイプ :	RTS/CTS または XON/XOFF
ボーレート :	300, 600, 1200, 2400, 4800, 9600
パリティ :	なし，偶数，奇数
データビット :	7 または 8
ストップビット :	1 または 2
	XON，XOFF 用の文字
	テキストエンド用文字
特殊ファンクション	XON で開始
	LF でプログラム先頭
	CR LF でブロックエンド
	転送終了文字で停止
	DSR 信号の判定
	リーダおよびトレーラ
	テープフォーマット
	時間監視

16 進数の入力には，大文字，小文字の両方が使用できますが，画面には常に小文字で表示されます。(1A，1a のどちらを入力しても画面上には 1a と表示されます)

これらのパラメータはマシンデータとして内部的に保存されます。パラメータ画面フォームを介して，またマシンデータ領域でも変更できます。
パラメータ／マシンデータの説明については，1.12.4「データの説明」を参照してください。

■ MMC 101 ～ 103

MMC101 ～ 103 のインタフェース X6 および X7 は，次の 2 つのパラメータセットを介してセットできます。

- RS 232 C (V. 24)
- PG

アドレス

この 2 つのインタフェースは，以下のアドレスへ恒久的に接続されます。

- COM1: 3F8 ～ 3FF
- COM2: 2F8 ～ 2FF

ファイルタイプ

MMC 101 ～ 103 のシリアルインタフェース X6 および X7 を介して，すべてのファイルがインポート，エクスポートできます。

インポート，エクスポートに使用するファイル選択用オペレータ入力については，以下に説明があります。

参照： YS840DI ユーザーズマニュアル・操作編

パラメータ設定

以下に示すパラメータセットは，SERVICES-INTERFACES でオペレータインタフェースを介して変更できます。

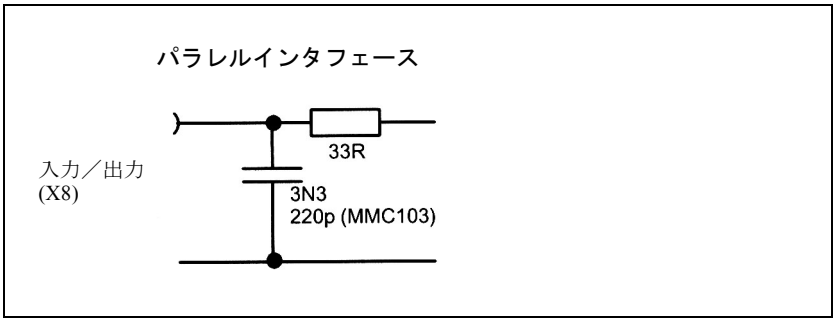
- インタフェース割り当て (COM1，COM2)
- プロトコル (Xon, Xoff, RTS/CTS)
- パリティ
- ストップビット
- データビットの数
- ボーレート
- アーカイブフォーマット（バイナリフォーマット (PC フォーマット)，LF のみのテープフォーマット，CR+LF 付きのテープフォーマット）
- リーダ／トレーラ
- 特殊転送文字
- 時間切れ

■ パラレルインタフェース（セントロニクス）

パラレルインタフェースは，MMC 101 および 102/103 にのみ提供されます。これは，双方向セントロニクスインタフェースで，主にプリンタインタフェースとして使用されます。

・MMC 101/102/103　　パラレルインタフェース
(セントロニクス)　　コネクタ X 8

接続



最大長さ

インタフェース接続用リード線（プリンタ用など）の長さは最大 2.5 m です。

パラメータ設定

パラレルインタフェースのパラメータ設定はできません。

アドレス

パラレルインタフェースは，アドレス 3BC ～ 3BE へ恒久的に接続されます。

■ ディスク装置

ディスク装置インタフェースは，MMC 101 ～ 103 の コネクタ X9 に接続できます (ディスクドライブ用の電源はありません)。このインタフェースは，3.5 インチディスク装置の接続専用に提供されます。

最大長さ

インタフェースに接続するリード線の長さは最大 0.5 m です。リード線は，ディスクドライブの一部として提供されます。

パラメータ設定

インタフェースのパラメータ設定はできません。

1.12.3 補足条件

パラメータ

シリアルインタフェースを介してデータを転送する場合，互いの通信相手に同じパラメータ設定がされていることが大切です。特にボーレートおよびストップビットの数が適切に設定されているか確認してください。

ケーブル

オーダ番号 6FX2 002-1AA01-1BF0 のケーブルは，PC または PG への接続用です。シリアル入力のあるプリンタを接続する場合は，モデムエリミネータコネクタの付いたケーブル，または標準のシリアルプリンタケーブルを使用する必要があります。

1.12.4 データの説明 (MD, SD)

MMC 100 標準装置のシリアルインタフェース X6 に使用する 3 つのパラメータセットは，以下に示す 3 つのマシンデータグループに保存されます。

~ マシンデータ	パラメータブロック	略語
9300・9308	ユーザ	USER
9310・9318	プリンタ	PRINTER
9320・9328	PG/PC	PG_PC

■ XON 文字

9300 9310 9320 MD 番号	V24_USER_XON V24_PRINTER_XON V24_PG_PC_XON XON character		
初期設定 : 11	最小入力リミット : 00 (01 から使用)	最大入力リミット : FF (1F までで使用)	
変更後直ちに有効	保護レベル : 2	単位 : 16 進数	
データタイプ : BYTE	適用開始 SW バージョン : 1.1		
意味 :	XON 文字 : この文字で転送を開始します。XON/OFF デバイスタイプでのみ有効です。 特殊ファンクション "Start with XON"（XON で開始）が起動すると，接続デバイスから送られてくる XON 文字を受け取ってから，プログラムが開始します。 標準では，デバイス制御文字（DEVICE CONTROL 1 (X-ON) または DC 1 が 11H。 これが初期値です。 入力 : XON の下の「パラメータ」画面に数字を入力（16 進）		

■ XOFF 文字

9301 9311 9321 MD 番号	V24_USER_XOFF V24_PRINTER_XOFF V24_PG_PC_XOF XOFF 文字		
初期設定 : 13	最小入力リミット : 00 (01 から使用することを推奨)	最大入力リミット : FF (1F までで使用することを推奨)	
変更後直ちに有効	保護レベル : 2	単位 : 16 進数	
データタイプ : BYTE	適用開始 SW バージョン : 1.1		
意味 :	XOFF 文字 : この文字で転送を停止します。XON/OFF デバイスタイプでのみ有効です。 標準では，デバイス制御文字（DEVICE CONTROL 3 (X-OFF) または DC 3 が 13H。 これが初期値です。 入力 : XOFF の下の「パラメータ」画面に数字を入力（16 進）		

■ 転送終了文字

9302 9312 9322 MD 番号	V24_USER_EOF V24_PRINTER_EOF V24_PG_PC_EOF **転送終了文字**		
初期設定 : 1A	最小入力リミット : 00 (01 から使用することを推奨)	最大入力リミット : FF (1F までで使用することを推奨)	
変更後直ちに有効	保護レベル : 2	単位 : 16 進数	
データタイプ : BYTE	適用開始 SW バージョン : 1.1		
意味 :	この文字で転送を終了します。初期設定は 1A: テキストファイルのファイルエンドを示す DOS 文字入力 : "End of transmission"（転送終了）の下の「パラメータ」画面に入力。 この文字は，"Stop with end of transmission character"（転送終了文字で停止）に X マークがある場合に有効となります。		

■ 特殊ビット

9303 9313 9323 MD 番号	V24_USER_CONTROLS V24_PRINTER_CONTROLS V24_PG_PC_CONTROLS **特殊ビット**		
初期設定 : 01001100 01001100 10000000	最小入力リミット : -	最大入力リミット : -	
変更後直ちに有効	保護レベル : 2	単位 : 空白または X マーク	
データタイプ : BYTE	適用開始 SW バージョン : 1.1		
意味 :	「パラメータ」画面で起動できる特殊ファンクションは，これらの特殊ビットに保存されます。このビットがセットされると，特殊ファンクションはアクティブになります。以下に示す割り当てで，マシンデータにビットとして保存されます。 Bit 0　XON で開始 Bit 1　LF でプログラム開始 Bit 2　CR LF でブロック終了 Bit 3　転送終了文字で停止 Bit 4　DSR 信号の判定 Bit 5　リーダおよびトレーラ Bit 6　テープフォーマット Bit 7　時間監視 入力 : "Special functions"（特殊ファンクション）下の「パラメータ」画面で起動 特殊ビットの説明を以下に示します。		

ビット 0

XON で開始

有効 : データの中に XON として定義された文字があると，転送が開始されます。デバイスタイプ XON/XOFF がセットされている場合にのみあてはまります。

無効 : XON 文字に関係なく開始します。

ビット 1

LF でプログラム開始

今のところ使用できません。

ビット 2

CR LF でブロック終了

有効 : CR 文字 (キャリッジリターン，16 進 OD) が，テープフォーマット出力に挿入されます。

テープフォーマット入力では，CR 文字は消去されます。

無効 : 文字は挿入されません。

ビット 3

転送終了文字で停止

有効 : 転送終了文字が判定されます。

無効 : 転送終了文字は判定されません (バイナリデータの転送に必要)。

ビット 4

DSR 信号の判定

有効 : DSR 信号（X6 または X7 コネクタの 6 番ピン（MMC 101 ～ 103 場合のみ）) が存在しないと，転送が中断します。

無効 : DSR 信号は無効。

ビット 5

リーダおよびトレーラ

有効 : 入力中リーダは無視。出力中は 120x0（16 進）が出力（データの前後にフィード)。

無効 : リーダ，トレーラとも読込まれます

出力には，リーダ 0（16 進）はなし。

ビット 6

テープフォーマット

有効 : DIN 66025 にしたがってプログラムが読み込まれ，% ファイル名 , %MPFxxx または %SPFxxx により開始します。

無効 : アーカイブが YS840DI アーカイブフォーマットに読込まれます。

ビット 7

時間監視

有効 : 転送障害があると，10 秒後に転送が異常終了します。時間監視は，文字の転送ごとにリセットされるタイマにより制御されます。

無効 : 転送は異常終了しません。

■ デバイスタイプ

9304 9314 9324 MD 番号	V24_USER_RTS V24_PRINTER_RTS V24_PG_PC_RTS デバイスタイプ		
初期設定 : 1 (XON/XOFF)	最小入力リミット : 0	最大入力リミット : 1	
変更後直ちに有効	保護レベル : 2	単位 : ビット	
データタイプ : BYTE	適用開始 SW バージョン : 1.1		
意味 :	2 つのデバイスタイプ（XON/OFF および RTS/CTS）に対して転送制御サポートがあります。 1: XON/XOFF 制御文字 XON (DC1，DEVICE CONTROL 1) および XOFF (DC3) が，転送制御のタイプの 1 つです。I/O 装置のバッファがいっぱいになると直ちに XOFF が転送され，データが受信可能になると直ちに XON が転送されます。 0: RTS/CTS RTS (送信要求) 信号は，データ転送装置の転送を制御します。有効 : 送信データ，受動。転送データのセットが完了するまでは，転送モードを解除してはなりません。 CTS (送信許可) 信号は，RTS に対する確認応答信号で，データ転送装置が送信レディ状態であることを知らせます。 入力 : "Device type"（デバイスタイプ）下の「パラメータ」画面で選択。		

■ ボーレート

9305 9315 9325 MD 番号	V24_USER_BAUD V24_PRINTER_BAUD V24_PG_PC_BAUD ボーレート		
初期設定 : 5 9600 baud	最小入力リミット : 0	最大入力リミット : 5	
変更後直ちに有効	保護レベル : 2	単位 : 16 進数	
データタイプ : BYTE	適用開始 SW バージョン :		
意味 :	ボーで表した変調レートで，データ転送レートの単位です。入力 : "Baud rate"（ボーレート）下の「パラメータ」画面で選択。 0: 300 baud 1: 600 baud 2: 1200 baud 3: 2400 baud 4: 4800 baud 5: 9600 baud		

■ データビット

9306 9316 9326 MD 番号	V24_USER_DATABITS V24_PRINTER_DATABITS V24_PG_PC_DATABITS データビット	
初期設定 : 1 8 データビット	最小入力リミット : 0	最大入力リミット : 1
変更後直ちに有効	保護レベル : 2	単位 : 16 進数
データタイプ :	適用開始 SW バージョン :	
意味 :	非同期転送用のデータビット数。 入力 : "Data bits"（データビット）下の「パラメータ」画面で選択。 0: 7 データビット 1: 8 データビット	

■ パリティ

9307 9317 9327 MD 番号	V24_USER_PARITY V24_PRINTER_PARITY V24_PG_PC_PARITY パリティ	
初期設定 : 0 パリティなし。	最小入力リミット : 0	最大入力リミット : 2
変更後直ちに有効	保護レベル : 2	単位 : 16 進数
データタイプ : BYTE	適用開始 SW バージョン :	
意味 :	パリティビットでエラーを検知します。コード文字にパリティビットの桁が追加され，1 にセットされた桁の数を奇数（奇数パリティの場合）または偶数（偶数パリティの場合）にします。 入力 : "Parity"（パリティ）下の「パラメータ」画面で選択。 0: パリティなし 1: 偶数パリティ 2: 奇数パリティ	

■ ストップビット

9308 9318 9328 MD 番号	V24_USER_STOPBIT V24_PRINTER_STOPBIT V24_PG_PC_STOPBIT ストップビット	
初期設定 : 0 1 ストップビット	最小入力リミット : 0	最大入力リミット : 1
変更後直ちに有効	保護レベル : 2	単位 : 16 進数
データタイプ : BYTE	適用開始 SW バージョン : 1.1	
意味 :	非同期転送用のストップビット数。 入力 : "Stop bits"（ストップビット）下の「パラメータ」画面で選択。 0: 1 ストップビット 1: 2 ストップビット	

1.12.5 信号の説明

ハードウェアインタフェース

インタフェースのハードウェア設計については，以下の説明を参照してください。

参照： /BH/ Operator Components Manual

■ シリアルインタフェース

割当て

シリアルインタフェース X 6 および X 7(X7 は MMC 101 ～ 103 のみ) のピン割当ては，一般の PC 機器管理基準に準拠しています。RS 232 C (V. 24) が，以下に ID コードとともに示される各リード線の役割を定義します。

接続	名称	意味	接続タイプ
1	DCD	キャリア検知	入力
2	RxD	データ受信	入力
3	TxD	データ転送	出力
4	DTR	データ端末レディ	出力
5	GND	接地	接地
6	DSR	データセットレディ	入力
7	RTS	送信要求	出力
8	CTS	送信許可	入力
9	RI	リングインジケータ	入力

レベル

シリアルインタフェースのレベルは，V.28 規格で管理されています。± 12V で動作する標準 IC 75188 および 75189 はドライバとしてもレシーバとしても使用されます。

制御回路

MMC 100 の制御回路は，インタフェースコントローラ 16450 タイプと互換性があります。

MMC 101 ～ 103 の制御スイッチは，インタフェースコントローラ 16450 および 16550 と互換性があります。

■ パラレルインタフェース

割当て

パラレルインタフェース X 8(MMC 101 ～ 103 のみ) のピン割当ては，一般の PC 機器管理基準，D-sub 25 ピンメスコネクタに準拠しています。コネクタの割当てを下表に示します。

接続	名称	意味	接続タイプ
1	STROBE	キャリア検知	出力
2	DATA 0	プリンタデータビット 0	双方向
3	DATA 1	プリンタデータビット 1	双方向
4	DATA 2	プリンタデータビット 2	双方向
5	DATA 3	プリンタデータビット 3	双方向
6	DATA 4	プリンタデータビット 4	双方向
7	DATA 5	プリンタデータビット 5	双方向
8	DATA 6	プリンタデータビット 6	双方向
9	DATA 7	プリンタデータビット 7	双方向
10	ACKNLG	プリンタからの確認応答	入力
11	BUSY	プリンタからのビジー信号	入力
12	PE	紙なし	入力
13	SLCT	プリンタからの選択信号	入力
14	ADF	オートフィードプリンタ	出力
15	ERROR	プリンタからのエラー信号	入力
16	INIT	プリンタ初期化	出力
17	SLCTIN	プリンタへの選択信号	出力
18	GND	接地	接地
19	GND	接地	接地
20	GND	接地	接地
21	GND	接地	接地
22	GND	接地	接地
23	GND	接地	接地
24	GND	接地	接地
25	GND	接地	接地

1.12.6 例

転送処理に先立ち，お互いの通信相手に同じパラメータ設定をする必要があります。たとえば下の例では，左側のパラメータが，MMC 100 の SERVICES/DATA/V.24_SET で操作パネルの選択フィールドへ入力されます。右側のパラメータは，チェックボックスに X を入力すると起動します。X の付いたファンクションが起動します。

■ シリアルプリンタ用のパラメータ

シリアルインタフェースのあるプリンタが，適切なケーブル（CTS ケーブルチェック）で接続されます。

デバイスタイプ	RTS-CTS		XON で開始
ボーレート	9600		LF でプログラム開始
ストップビット	1	X	CR LF でブロック終了
パリティ	なし		転送終了文字で停止
データビット	8		DSR 信号の判定
XON			リーダおよびトレーラ
XOFF		X	テープフォーマット
転送終了	0c (フォームフィード)		時間切れ

■ PG/PC を使用したアーカイブ用パラメータ

デバイスタイプ	RTS-CTS		XON で開始
ボーレート	9600		LF でプログラム開始
ストップビット	1		CR LF でブロック終了
パリティ	なし		転送終了文字で停止
データビット	8	X	DSR 信号の判定
XON	00		リーダおよびトレーラ
XOFF	00		テープフォーマット
転送終了	00	X	時間切れ

この設定により，ファイルは保管用に圧縮され，YS840DI フォーマットに読込まれます。

（MSD，FDD ファイルの）トランスペアレントな転送では，"Stop with end of transmission character"（転送終了文字で停止）を選択してはいけません（書込み時，PG は自動的に停止しません）。

ASCII データを使用した他の設定も可能ですが，プログラマ側の設定と一致していることが必要です。これには，6FX 2002-1AA01- ケーブルを使用します。

■ DIN プログラム用のパラメータ

デバイスタイプ	RTS-CTS		XON で開始
ボーレート	9600		LF でプログラム開始
ストップビット	1	X	CR LF でブロック終了
パリティ	なし	X	転送終了文字で停止
データビット	8	X	DSR 信号の判定
XON	11		リーダおよびトレーラ
XOFF	13	X	テープフォーマット
転送終了	1a		時間切れ

この設定では，プログラムは DIN に準拠して読込むことも，システム 3/8 から読込む（% で開始）こともできます。

■ テープ入力および出力

テープリーダおよびテープパンチャでは，リーダおよびトレーラに X のマークを付けなければなりません。テープリーダが CTS を介して制御される場合は，"Start with XON"（XON で開始）も合わせてマークしなければなりません。

INPUT START/STOP を押すことによって，テープを挿入するためにテープリーダを止めることができます。

デバイスタイプ	RTS-CTS		XON で開始
ボーレート	9600		LF でプログラム開始
ストップビット	2	X	CR LF でブロック終了
パリティ	なし	X	転送終了文字で停止
データビット	8		DSR 信号を判定
XON	00	X	リーダおよびトレーラ
XOFF	00	X	テープフォーマット
転送終了	00	X	時間切れ

■ マシンデータのインポート

マシンデータファイル INITIAL.INI で，マシンの初期設定を行います。

V.24 (RS 232) 規格のインタフェースでは，入力する前に，"Path from workpiece/archive"（ワーク／アーカイブからのパス）フィールドに X を付けておく必要があります。

これは，アーカイブフォーマット，テープフォーマットのデータにあてはまります。

NC リセットが実行された後，データが起動します。

■ バイナリデータのインポート（FDD，MSD）

デバイスタイプ	RTS-CTS	XON で開始
ボーレート	9600	LF でプログラム開始
ストップビット	1	CR LF でブロック終了
パリティ	なし	転送終了文字で停止
データビット	8	DSR 信号の判定
XON		リーダおよびトレーラ
XOFF		テープフォーマット
転送終了	00	時間切れ

1.12.7 データフィールド，リスト

■ マシンデータ

番号	識別子	名称	備考
操作パネルデータ ($MM_ ...)　ユーザパラメータ (USER)			
9300	V24_USER_XON	XON 文字	
9301	V24_USER_XOFF	XOFF 文字	
9302	V24_USER_EOF	転送終了文字	
9303	V24_USER_CONTROLS	特殊ビット	
9304	V24_USER_RTS	デバイスタイプ	
9305	V24_USER_BAUD	ボーレート	
9306	V24_USER_DATABITS	データビット	
9307	V24_USER_PARITY	パリティ	
9308	V24_USER_STOPBIT	ストップビット	

番号	識別子	名称	備考
操作パネルデータ ($MM_ ...)　プリンタパラメータ (PRINTER)			
9310	V24_PRINTER_XON	XON 文字	
9311	V24_PRINTER_XOFF XOFF 文字		
9312	V24_PRINTER_EOF	転送終了文字	
9313	V24_PRINTER_CONTROLS	特殊ビット	
9314	V24_PRINTER_RTS	デバイスタイプ	
9315	V24_PRINTER_BAUD	ボーレート	
9316	V24_PRINTER_DATABITS	データビット	
9317	V24_PRINTER_PARITY	パリティ	
9318	V24_PRINTER_STOPBIT	ストップビット	

番号	識別子	名称	備考
操作パネルデータ ($MM_ ...)　PG/PC パラメータ (PG_PC)			
9320	V24_PG_PC_XON	XON 文字	
9321	V24_PG_PC_XOFF	XOFF 文字	
9322	V24_PG_PC_EOF	転送終了文字	
9323	V24_PG_PC_CONTROLS	特殊ビット	
9324	V24_PG_PC_RTS	デバイスタイプ	
9325	V24_PG_PC_BAUD	ボーレート	
9326	V24_PG_PC_DATABITS	データビット	
9327	V24_PG_PC_PARITY	パリティ	
9328	V24_PG_PC_STOPBIT	ストップビット	

1.13 非常停止入力 (N2)

1.13.1 概略説明

EN 292-2 規格

EN 292-2 の 6.1.1 項の EMERGENCY STOP（非常停止）で規定されている，EC Machinery Directive（EC 機械規則）の基本安全要求事項に従い，マシンには非常停止設備がついていなければなりません。

例外

次の場合は非常停止設備が要求されません：

機械に非常停止設備があっても危険が回避できない場合。これは，停止時間がどうしても減少できない場合や，その非常停止方法が危険を抑制するのに適切でない場合などです。

ハンドヘルドあるいはポータブルマシンの場合。

制御システム中の非常停止

本制御システムは，以下に示す機能を提供することによって，マシン製造業者が非常停止ファンクションを実現できるようにしています：

非常停止ボタンは，マシン制御パネル上の手に届きやすいところに位置し，黄色の背景色で目立たせる。

非常停止ボタンは，色が赤で，確実に開き，自動メカラッチ（メカロック）式である。

PLC 入力に応じて NC 内で非常停止シーケンスが起動する。

NC で非常停止手順が取られると，全ての軸とスピンドルはできるだけ早く減速する。

PLC により制御された全てのマシンファンクションは安全状態になる。この安全状態は，マシン製造業者が設定することができる。

非常停止状態は，非常停止ボタンを元に戻しただけではキャンセルされない。

制御デバイスをリセットしても，マシンがかってに再起動することはない。

非常停止状態がキャンセルされた後，軸を基準化したりスピンドルを同期化する必要はない（位置が自動的に補正される）。

1.13.2 詳細説明



重要

マシン製造業者は，国内あるいは国際規格をよく読んでおくことを義務づけられています（次項の注記を参照）。YS840DI は，マシン製造業者が本ファンクション説明に従って非常停止ファンクションを実施できるようになっています。非常停止ファンクション（非常停止のトリガ，手順，および確認）に関する責任はすべてマシン製造業者にあります。

（注）

非常停止ファンクションに関しては，次の規格を参照してください：

・EN 292 パート 1

・EN 292 パート 2

・EN 418

・EN 60204 パート 1:1992 10.7 項

VDE 0113 パート 1 は移行期間中のみ有効で，その後は EN 60204 に置き換えられます。

非常停止ファンクション

EN 418: 非常停止の目的は下記のとおりです：

- 人身にかかわる危険や，マシンや加工材料への損傷を未然に防いだり，既にあるそのような危険を少なくしたりすること。
- 通常の停止ファンクションでは間に合わないとき，人間の一回の動作でトリガされること。

EN 418 によれば，危険は次のことから起こります：

- 機能不良（マシンの故障，加工できない材料の加工，人為的な過失など）
- 通常運転

■ 非常停止アクチュエータ

EN 418 規格

非常停止の操作部品（アクチュエータ）は，オペレータあるいは非常停止を使う必要のある者が簡単に動作できるよう設計されている必要があると，EN418 が定めています。

次にアクチュエータのタイプとして考えられるものを示します：

- きのこ形の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

全ての非常停止スイッチは，機械的にセルフロックするタイプで，すぐに手が届くところについていなければなりません。

非常停止押しボタン

YS840DI のマシン制御パネル (MCP) には，確実に開いて動作するきのこ形の押しボタンスイッチがついています。この非常停止アクチュエータは非常停止押しボタンと呼ばれます。

接続条件

非常停止押しボタンへの接続に関する詳細は，ハードウェア構成ガイドを参照ください。

参照： /BH/ Operator Components Manual

NC の非常停止

非常停止押しボタンの動作によりボタンから直接出る信号は，PLC 入力として，制御システム (PLC) に送られなければなりません。PLC ユーザプログラムでは，この PLC 入力は，NC 内の IS "EMERGENCY STOP" (DB10 DBX56.1) に送られなければなりません。

非常停止ボタンを元に戻した（リセットした）こと，あるいはそれによって直接発生する信号が，PLC 入力信号の形でプログラマブルコントローラ (PLC) に伝達される必要があります。PLC ユーザプログラムでは，この PLC 入力を，NC 内の IS "Acknowledge EMERGENCY STOP" (DB10 DBX56.2) に送らなければなりません。

■ 非常停止シーケンス

EN 418 規格

非常停止スイッチの操作部の動作によって起こる「非常停止」は，危険を回避あるいは減少するために可能な限りベストの方法で，自動的に実行されることが必要です。

「可能な限りベストの方法」というのは，予測される危険に従って，最小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 内のシーケンス

下記に示すのは，非常停止制御システムに (EN 418 で）あらかじめ定義された，非常停止状態になるために実行される内部ファンクションのシーケンスです：

1. パートプラグラムの実行が中断される。全ての軸とスピンドルが MD 36610: AX_EMERGENCY_STOP_TIME で定義された制動ランプに沿って停止する。急勾配の制動ランプ（AX_EMERGENCY_TIME を小さく設定）によって，最大制動電流（セットポイント速度入力 = 0) で急減速が行われる。
2. IS "Mode group Ready" (DB11, ... DBX6.3) がリセットされる。
3. IS "EMERGENCY STOP active" (DB10 DBX106.1) がセットされる。
4. アラーム 3000 がセットされる。
5. MD 36620: SERVO_DISABLE_DELAY_TIME （カットアウト遅延サーボイネーブル）で特定の軸／スピンドルに設定された遅延時間が経過すると，コントローライネーブル（FM-NC: コントローライネーブルリレー）がディスエーブルされる。SERVO_DISABLE_DELAY_TIME での設定は，少なくとも AX_EMERGENCY_STOP_TIME での設定と同じであることに注意する。
6. 全ての軸とスピンドルは，内部でフォローアップモードに切り替わる（位置制御はアクティブにならない)。

マシンのシーケンス

マシンでの非常停止ファンクションのシーケンスは，マシン製造業者が決定します。NC での手順に関連して，次の点に注意が必要です：

- NC での運転シーケンスは，IS "EMERGENCY STOP" (DB10 DBX 56.1) から開始される。軸とスピンドルが停止したら，EN 418 に従って電源を遮断しなければならない。

重要

電源を遮断するのはマシン製造業者の責任です。

- PLC の I/O （デジタルとアナログの I/O）は，NC の手順に影響される。非常停止の場合，個別の出力が，特定の状態あるいは電圧レベルに達する必要があれば，マシン製造業者は PLC ユーザプログラムにこの目的を果たすファンクションを入れておかなければならない。
- NCK I/O （高速デジタル I/O）は，NC での手順に影響されない。非常停止の場合，個別の出力が，ある特定の状態になっておく必要があれば，マシン製造業者は PLC ユーザプログラムで，この状態を IS "DB10 DBB4 to 7" 経由で NC に転送しておかなければならない。

重要

非常停止のときに，あらかじめ決定されたシーケンスで NC の内部ファンクションが実行されない方がよい場合は，PLC ユーザプログラムで，マシン製造業者によって定義された非常停止状態になる点までは，IS "EMERGENCY STOP" (DB10 DBX56.1) をセットしてはいけません。

「非常停止」インタフェース信号がセットされておらず，他のアラームが存在しない限り，全てのインタフェース信号は NC 内で有効です。したがって，製造業者によって定義されたいかなる非常停止状態（軸／スピンドル別とチャンネル別も含む）も，実行可能となります。

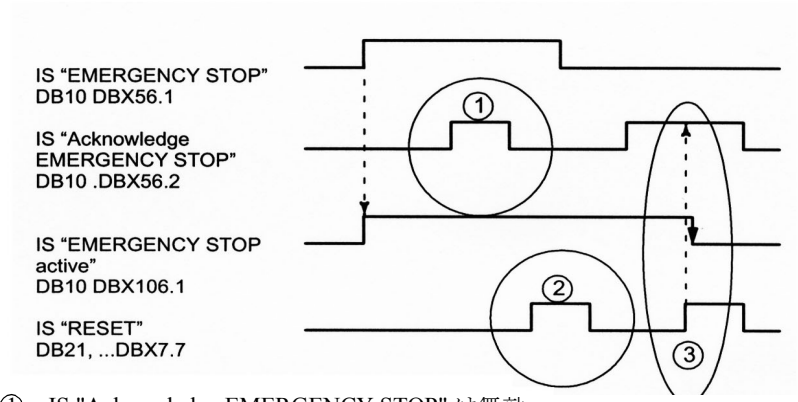
■ 非常停止の確認

EN 418 規格

非常停止スイッチは，操作部を手動で動作することによってのみリセットできます。非常停止のリセットにより，再始動コマンドをスタートさせてはいけません。全ての動作された非常停止操作部が，個別に意図的にリセットされるまでは，マシンが始動する可能性があってはいけません。

非常停止の確認

IS "Acknowledge EMERGENCY STOP" (DB10 DBX56.2) がセットされ，次に IS "Mode group reset" (DB11, ... DBX0.7) がセットされるまでは，非常停止状態はリセットされません。注意すべき事は，IS "EMERGENCY STOP active" (DB10 DBX106.1) をリセットするために，IS "Acknowledge EMERGENCY STOP" と IS "Reset" を（両方とも）十分長く設定しておく必要があることです（図 1.108 参照）。



① IS "Acknowledge EMERGENCY STOP" は無効
② IS "Reset" は無効
③ IS "Acknowledge EMERGENCY STOP" と "RESET" が "EMERGENCY STOP active" をリセットする

図 1.108 非常停止のリセット

非常停止をリセットすると次のことが起こります：

- サーボイネーブルがイネーブルになる。
- 全ての軸についてフォローアップモードがキャンセルされ，位置制御モードが再開される。
- IS "Position control active" がセットされる。
- IS "Mode group ready" がセットされる。
- "EMERGENCY STOP active" インタフェース信号がリセットされる。
- アラーム 3000 がキャンセルされる。
- パートプログラム処理が全チャンネルについて中断される。

PLC + NCK I/O

マシンが運転できるように，PLC ユーザプログラムが，PLC と NCK の I/O を正常状態に戻す必要があります。

リセット

非常停止状態は，IS "Reset" (DB21, ...DBX7.7) だけではリセットできません（上図を参照)。

電源オフ／オン

"EMERGENCY STOP" (DB10 DBX56.1) がセットされていない場合，電源を一旦切り，再起動させると，非常停止状態がリセットされます。

1.13.3 補足条件

いかなる場合にも，次の規格に準拠していなければなりません :

- EN 292
- EN 418
- EN 60204

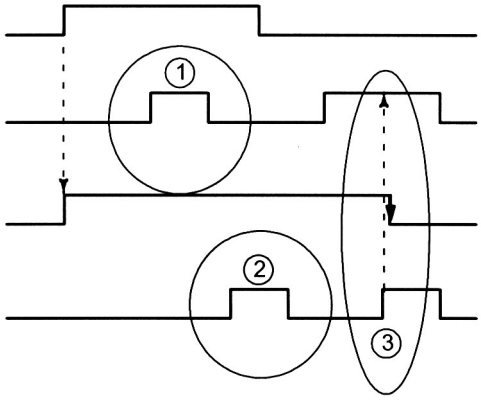
1.13.4 データの説明 (MD, SD)

なし

1.13.5 信号の説明

■ 一般信号

DB10 DBX56.1 **データブロック**	**非常停止** **NC への信号 (PLC → NC)**	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	NC が非常停止状態に切り替わり，非常停止手順が NC で始まる (1.13.2 ■「非常停止シーケンス」参照)。	
信号ステータス 0 あるいは信号遷移 1 → 0	• NC は非常停止状態 • 非常停止状態は（まだ）有効であるが，"acknowledge EMERGENCY STOP" と "mode group reset" のインタフェース信号でリセットすることができる。	
関連性	IS "Acknowledge EMERGENCY STOP" (DB10 DBX56.2) IS "EMERGENCY STOP active" (DB10 DBX106.1)	

DB10 DBX56.2 **データブロック**	**非常停止の確認** **NC への信号 (PLC → NC)**	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 あるいは信号遷移 0 → 1	非常停止状態は，"Acknowledge EMERGENCY STOP" (DB10 DBX56.2) に続いて IS "Mode group reset" (DB11, ... DBX0.7) がセットされた場合にのみリセットされます。この点で注意すべき事は，IS "EMERGENCY STOP active" (DB10 DBX106.1) をリセットするために，IS "Acknowledge EMERGENCY STOP" と IS "Reset" を（両方とも）十分長く設定しておく必要があることです。 非常停止をリセットすると，次の効果があります : • サーボイネーブルがイネーブルになる。 • 全ての軸についてフォローアップモードがキャンセルされ，位置制御モードが再開される。 • IS "Position control active" がセットされる。 • IS "Mode group ready" がセットされる。 • "EMERGENCY STOP active" インタフェース信号がリセットされる。 • アラーム 3000 がキャンセルされる。 • パートプログラム処理が全チャンネルについて中断される。	
	非常停止 インタフェース信号 非常停止確認 インタフェース信号 非常停止アクティブ インタフェース信号 リセット インタフェース信号  ① IS "Acknowledge EMERGENCY STOP" は無効 ② IS "Reset" は無効 ③ IS "Acknowledge EMERGENCY STOP" と "RESET" が "EMERGENCY STOP active" をリセットする	
例外，エラー	非常停止状態は，IS "Reset" (DB21, ... DBX7.7) ではリセットできない。	
関連性	IS "EMERGENCY STOP" (DB10 DBX56.1) IS "EMERGENCY STOP active" (DB10 DBX106.1) IS "Mode group reset" (DB10 DBX0.7)	

DB10 DBX106.1 **データブロック**	**非常停止アクティブ** **NC への信号 (PLC → NC)**	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン：1.1
信号ステータス 1 あるいは信号遷移 0 → 1	NC は非常停止状態	
関連性	IS "EMERGENCY STOP" (DB10 DBX56.1) IS "Acknowledge EMERGENCY STOP" (DB10 DBX56.2)	

1.13.6 例

なし

1.13.7 データフィールド，リスト

■ インタフェース信号

DB 番号	ビット，バイト	名称	参照
備考			
10	56.1	非常停止	
10	56.2	非常停止確認	
10	106.1	非常停止アクティブ	
モードグループ別			
11, ...	0.7	モードグループリセット	1.10 (K1)

■ マシンデータ

番号	識別子	名称	参照
軸別 ($MA_ ...)			
36610	AX_EMERGENCY_STOP_TIME	エラー状態用の制動ランプ長	1.2 (A3)
36620	SERVO_DISABLE_DELAY_TIME	シャットダウン遅延制御イネーブル	1.1 (A2)

■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

1.14 直径指定／半径指定 (P1)

1.14.1 概略説明

マシンデータを介してジオメトリ軸を横軸と定義することができます。横軸用の測定値は，半径フォーマットあるいは直径フォーマットでプログラムされます。プログラムされた直径は，G コマンドの DIAMON ／ DIAMOF によって起動／停止されます。

直径プログラミングを選択すると，

ワーク座標系 (WCS) の指令位置／実際値画面に直径関連データが表示されます。

すべてのオフセットが，半径フォーマットで入力され，プログラムされ，表示されます。

プログラムされたエンド位置が内部半径値に変換されます。

G2/G3 プログラミング用の絶対補間パラメータ（たとえば，I，J，K）が内部半径値に変換されます。

タッチトリガプローブを使用することによって WCS で生成された測定値が，直径フォーマットで保存されます。

システム変数を使用することにより，指令位置および実際値を WCS で直径フォーマットで読取ることができます。

半径プログラミングを選択すると，上記のデータは常に半径フォーマットで入力され，プログラムされ，内部保存され，読取られ，表示されます。

1.14.2 詳細説明

■ 横軸の定義

一般事項

"Longitudinal axis"（縦軸）は，通常 "turning"（旋盤加工）テクノロジでは Z 軸を表すのに使用される用語です。"transverse axis"（横軸）という用語は X 軸を表すのに使用されます。横軸用の測定値は通常直径フォーマットで指定されます（他の軸のパス寸法を 2 倍にします)。G コマンドを使用することにより，直径プログラミングと半径プログラミングの切換えを行うことができます。

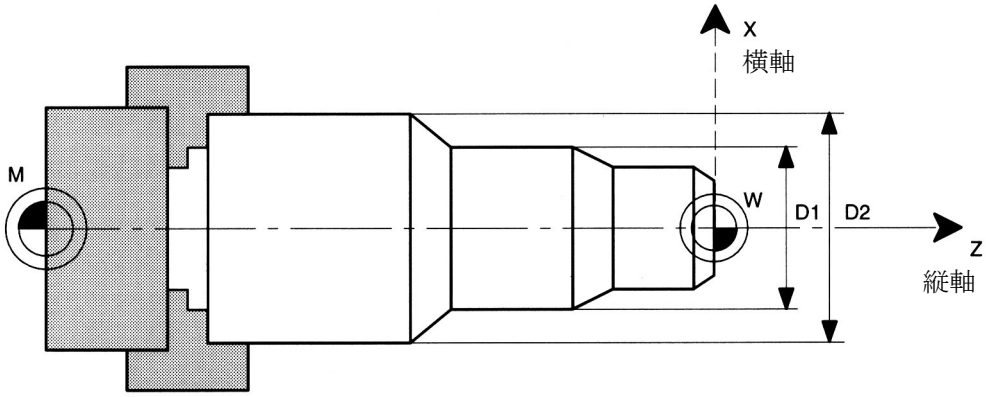


図 1.109　直径指定を使用した横軸

ジオメトリ軸

横軸

チャンネル別 MD 20100: DIAMETER_AX_DEF （横軸ファンクションを使用したジオメトリ軸）を使用して，ジオメトリ軸を向合った軸と定義することができます。1 つのチャンネルにつき 1 つの横軸を定義できます。

G96/G97 を使用した横軸

G96 （定切削速度）および G97 （定速）の場合に，その位置がマスタスピンドル速度を決定するジオメトリ軸は，横軸と定義されなければなりません。そうでない場合は，アラーム 10870 "No transverse axis defined"（横軸が定義されていない）が起動されます。

■ 直径プログラミング

起動と停止

直径あるいは半径を基準にして，横軸をプログラムすることができます。チャンネルの横軸用の直径プログラミングを起動または停止させるのに，プログラムコマンド "DIAMON" および "DIAMOF" が使用できます。

DIAMON および DIAMOF は G グループ 29 に属していて，モーダル的に有効です。初期設定は MD 20150: GCODE_RESET_VALUES [28]（G グループの初期設定）で定義されます。横軸と定義されていない軸について直径プログラミングを起動するために DIAMON が使用された場合，アラーム 16510 "No transverse axis defined"（横軸が定義されていない）が出力されます。

JOG モード

DIAMON が有効な場合，マシンファンクション INC （相対寸法）および JOG モードでハンドル移動用に入力されたインクリメントが，関連する横軸についての直径値として解釈され，移動します。

指令位置／実際値画面

横軸について "DIAMON" ファンクションが起動された場合，その位置，移動距離，および Repos オフセットが直径関連データとして表示されます（ワーク座標系 (WCS)）。

この情報は，常に半径関連データとしてマシン座標系 (MCS) で表示されます。

軸，FDD，および MSD 用のサービス画面で使用されるフォーマットは常に半径です。

オフセット

すべてのオフセット（たとえば，ツールオフセット，プログラム可能フレーム，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）は常に半径値として入力され，プログラムされ，表示されます（これらのオフセットが横軸で有効であって，コマンド DIAMON が有効であっても）。

作業エリア制限，ソフトウェアリミットスイッチ，フィード値

これらのデータは常に半径値として入力され，プログラムされ，表示されます。

位置決め軸

マシンデータによって横軸と定義されているジオメトリ軸が，コマンド POS [U] あるいは POSA [U] によってパートプログラム中で位置決め軸として操作される場合，この軸について有効な直径プログラミングは，再び選択解除されるまで操作可能なままです。

コマンド POSA [U] が有効なときは，位置がまだ到達されていなくても，プログラムは次の NC ブロックに進みます。

■ 直径値から内部半径値への変換

直径値⇒半径値

直径プログラミングが有効なときに，横軸について次の量が，内部半径値に変換されます（つまり，プログラムされた値が半分になります)。

G90/G91 を使用した絶対寸法であるか相対寸法であるかにかかわりなく，プログラムされたエンド位置

G2/G3 プログラミング用の絶対補間パラメータ（たとえば I，J，K)。絶対補間パラメータは WCS の座標原点を基準化します。

参照： YS840DI ユーザーズマニュアル・プログラミング編
(相対的にプログラムされた補間パラメータは変換されません)。

■ 内部半径値から直径値への変換

半径値⇒直径値

直径プログラミングが有効なときに，下記の機能と関連して，横軸の内部半径値が直径値に変換され（たとえば，半径値が 2 倍され)，保存されます。

ファンクション "MEAS" および "MEASW" を使用した WCS での測定値

システム変数 $P_EP[x] あるいは $AA_IW[x] を使用して WCS で読取られた指令位置および実際値

(MCS で測定あるいは読取りを行う場合は，計算された値が半径値として保存されます)。

1.14.3 補足条件

ファンクションの可用性

"Transverse axes"（横軸）ファンクションは SW2 以降で利用できます。

1.14.4 データの説明 (MD, SD)

20100 MD 番号	DIAMETER_AX_DEF 横軸ファンクションを使用したジオメトリ軸	
初期設定 : -	最小入力リミット : -	最大入力リミット : 16 文字
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : STRING	適用開始 SW バージョン : 1.1	
意味 :	1 つのチャンネルにつき 1 つの横軸を定義できる。 チャンネル別 MD 20050: AXCONF_GEOAX_ASSIGN_TAB[n] あるいは MD 24120: TRAFO_AX_GEOAX_ASSIGN_TAB_1[n]（SW 4 以降）および MD 20060: AXCONF_GEOAX_NAME_TAB[n] によって指定された有効なジオメトリ軸の軸識別子が指定されなければならない。空白文字が入力された場合，あるいは軸識別子が、ジオメトリ軸と定義されていない軸について指定された場合に，"diameter programming"（直径プログラミング）ファンクション (DIAMON) を起動すると，アラーム 16510 "No transverse axis exists"（横軸が存在しない）が出力され，G96/G97 をプログラムすると，アラーム 10870 "No transverse axis defined"（横軸が定義されていない）が出力される。	
関連性	MD 20050: AXCONF_GEOAX_ASSIGN_TAB[n] （ジオメトリ軸とチャンネル軸間の割当て） MD 20060: AXCONF_GEOAX_NAME_TAB[n] （チャンネル中のジオメトリ軸名） MD 24120: TRAFO_AX_GEOAX_ASSIGN_TAB_1[n] （変換 1 用のチャンネル軸に対するジオメトリ軸の割当て）	

1.14.5 信号の説明

なし

1.14.6 例

なし

1.14.7 データフィールド，リスト

■ マシンデータ

番号	識別子	名称	リファレンス
チャンネル別 ($machine code_ ...)			
20050	AXCONF_geometry_user interface_TAB[n]	チャンネル軸に対するジオメトリ軸の割当て	1.11 (K2)
20060	AXCONF_geometry_NAME_TAB[n]	チャンネル中のジオメトリ軸名	1.11 (K2)
20100	DIAMETER_AX_DEF	横軸ファンクションを使用したジオメトリ軸	
20150	GCODE_RESET_VALUES[n]	G グループのリセット	1.10 (K1)

■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

1.15 基本的 PLC プログラム (P3)

1.15.1 概略説明

一般事項

基本プログラムは，PLC ユーザプログラムと NCK（数値制御カーネル）間，MMC（マンマシンコミュニケーション）と MCP（マシン制御パネル）領域間の，信号およびデータの交換を管理します。信号およびデータは，以下のグループに分けられます。

- 周期的信号交換
- イベント駆動信号交換
- メッセージ

周期的信号交換

周期的に交換される信号は，基本的にはビットフィールドから構成されています。

- これらは，PLC から NCK へ転送されるコマンド（スタートまたはストップ，など），および NCK からのステータス情報です（プログラムが実行中または中断，など）。
- ビットフィールドは以下の信号グループに編成されます。
  モードグループ
  チャンネル
  軸／スピンドル
  一般 NCK 信号

データの周期的交換は，PLC 周期（OB1）のスタート時に基本プログラムによって実行されます。これにより NCK の信号が，周期中は一定となります。

イベント制御信号交換：NCK から PLC

PLC 機能は，ワークプログラムの機能として実行されるもので，ワークプログラムにある補助機能によって起動します。補助機能でブロックの実行をスタートした場合，NCK は機能の実行（ツール交換中，など）まで待たなければならないのかどうか，あるいはワークの機械加工と平行して機能が実行されるのかどうか（例えば，鎖型の工具入れのあるフライス盤にツールを準備するため）は，補助機能のタイプによって決まります。

データ転送は，NC マシニングのプロセスに及ぼす影響を最小限にするためにも，できる限り速く，より確実に行わなければなりません。したがって，アラームおよび確認応答によってデータ転送を制御します。基本プログラムは信号およびデータを評価し，これを NCK に確認応答し，周期のスタート時に，アプリケーションインターフェースにデータを転送します。ユーザの確認応答を必要としないデータの場合，NC マシニングに影響を及ぼすことはありません。

イベント制御信号交換：PLC から NCK

「イベント駆動信号交換：PLC から NCK」は，PLC が NCK にジョブを送信するときはいつも行われます（補助軸用移動コマンド，など)。この場合，データ転送も確認応答によって制御されます。ユーザプログラムから実行した場合，このタイプの信号交換は，機能ブロック（FB）あるいは機能コール（FC）を使用してトリガされます。

関連 FB（機能ブロック）および FC（機能コール）は，基本プログラムに添付してあります。

メッセージ

基本プログラムによって，ユーザメッセージを収集して条件付けします。定義されたビットフィールドを使用して，メッセージ信号を基本プログラムへ転送します。メッセージイベントが発生すると，信号はここで評価され PLC 診断バッファへ入力されます。運転パネルがある場合（MMC 100，など)，メッセージは OP に転送され，ここで表示されます。



警告

マシンの機能は，PLC プログラムによってほぼ確定します。作業メモリ内にあるすべての PLC プログラムは，プログラミング装置で編集することができます。

1.15.2 詳細説明

■ PLC-CPU のキーデータ

以下は，PLC のパフォーマンスデータと，さまざまな制御バージョンに対する基本プログラムの範囲の一覧表です。

制御のタイプ 特徴		YS840DI 統合 PLC CPU315-2DP マスター／スレーブ 6ES7 315-2AF01-0AB0
1. キー CPU データ		
ユーザおよび基本プログラム用メモリ		64, 96 128, 160, 192, 224, 256, 288 K バイト (オプションによる)
メモリサブモジュール		なし
ビットメモリ		2048/4096 PLC のオペレーティングシステムがバージョンコード 03. 10. 13 以降
タイマ		128
カウンタ		64
クロックメモリ		8
プログラム／データブロック OB FB FC DB		 1, 10, 20, 35, 40, 80 ~ 82, 85 ~ 87 , 100, 120 ~ 122 0 ~ 255 0 ~ 255 1 ~ 399
最大データブロック長		16 K バイト
FC, FB の最大ブロック長		24 K バイト
入力／出力（アドレス指定容量） - デジタル - アナログ		 1024/1024 64
入力／出力（アドレス指定容量） - デジタル - アナログ	Rack 0 は NC に統合されている。Rack 1 から 3 は I/O 装置で使用。	I/O 装置のオプション設定により。 I/O 32 バイトから PI/PO 272 バイトから
実行時間 - ビット命令 (I/O) - ワード命令		 0.3 ms/kA 1-4 ms/kA
DIAG (アラーム S,SQ)		あり
Profibus		マスター／スレーブ
プログラマブル PBK ブロックコミュニケーション		あり
SFC 14, 15 を介して標準スレーブへ一貫しているデータ		26
2. I/O 拡張		
I/O モジュール		24
Profibus L2DP モジュール		あり
インタフェース (MPI)		1

バージョンディスプレイにあるバージョンコードは，以下の情報を表しています。

AUT 1 PLC バージョン　ファームウェア転送のインクリメント　内部インクリメント

MMC のバージョンディスプレイ

NCK ソフトウェアにあるバージョンディスプレイを最後までスクロールすると，現在インストールされている PLC，および PLC オペレーティングシステムの関連バージョンの詳細を見ることができます。

例えば，

S7 PLC_315-2DP システム

03.10.08

インストールされている PLC モジュールのモジュール ID は，次の欄にあります。

現在使用されている PLC モジュール ID は，以下のとおりです。

1108	PLC 315-2DP （MLFB 6ES7 315-2AF01-0AB0）
1200	PLC 315-2DP (近日発売)

		YS840DI
インタフェース (OPI)		1
3. 基本プログラムの設定可能な機能		
	適用範囲：	
	軸／スピンドル	31
	チャンネル	10
	モードグループ	10
	機能 :	
	ステータス／制御信号	*
	M デーコーダ (M00-99)	*
	G グループデコーダ	*
	補助機能ディストリビュータ	*
	アラーム駆動補助機能転送	*
	軸／スピンドルの PLC からの移動	*
	非同期サブプログラムインタフェース	*
	エラー／運転メッセージ	*
	MCP および NCK を介したハンドヘルドユニット信号	*
	NC 変数の読取り／書込み	*
	PI サービス	*
	ツール管理	*
	star/delta スイッチオーバー	*

PLC 機能については，カタログ ST 60.1 もあわせて参照してください。

■ PLC-CPU のハードウェア構成のスタートアップ

一般事項

ご使用になる PLC CPU の，関連 I/O を含めたハードウェア構成は，STEP 7 を使用して定義してください。最も簡単な方法は，STEP 7 を介して，ハードウェアコンポーネントの定義に必要な情報を読取ることです。

以下は，STEP 7 バージョン 2.1 の場合の，操作手順とその仕組みです。

1. 新規プロジェクトの作成（File New Project）
2. ハードウェア SIMATIC 300 station を挿入
3. SIMATIC 300 station 1 をマウスで選択
4. 右クリックでオブジェクトを開き，HW Config をスタート
5. 通信相手のシステムを PG にロード，ハードウェア装置の定義に必要な情報を中央システムから読取る
6. 分散 I/O は拡張

I/O モジュール用アドレスは，必要であれば変えることができます（ただし，PLC 315-2DP などの特定された PLC CPU に限ります）。

選択可能なので，すべてのハードウェア構成をご自分で入力することができます（STEP 7 の取扱説明書の該当部分もあわせてお読みください。）

以下の注記に示す内容にしたがってください。

YS840DI コンポーネントのハードウェア構成は，RACK 300 に入力して行うことができます。この設定を行うには，ツールボックスディスクにある基本プログラムをインストールしてください。

（注）YS840DI では，SIMATIC Rack 0 は NC に統合されます。この Rack にインストールされているコンポーネントは以下のとおりです。
スロット２の統合 PLC (PLC 314 もしくは PLC 315-2DP)
スロット３の IM 360

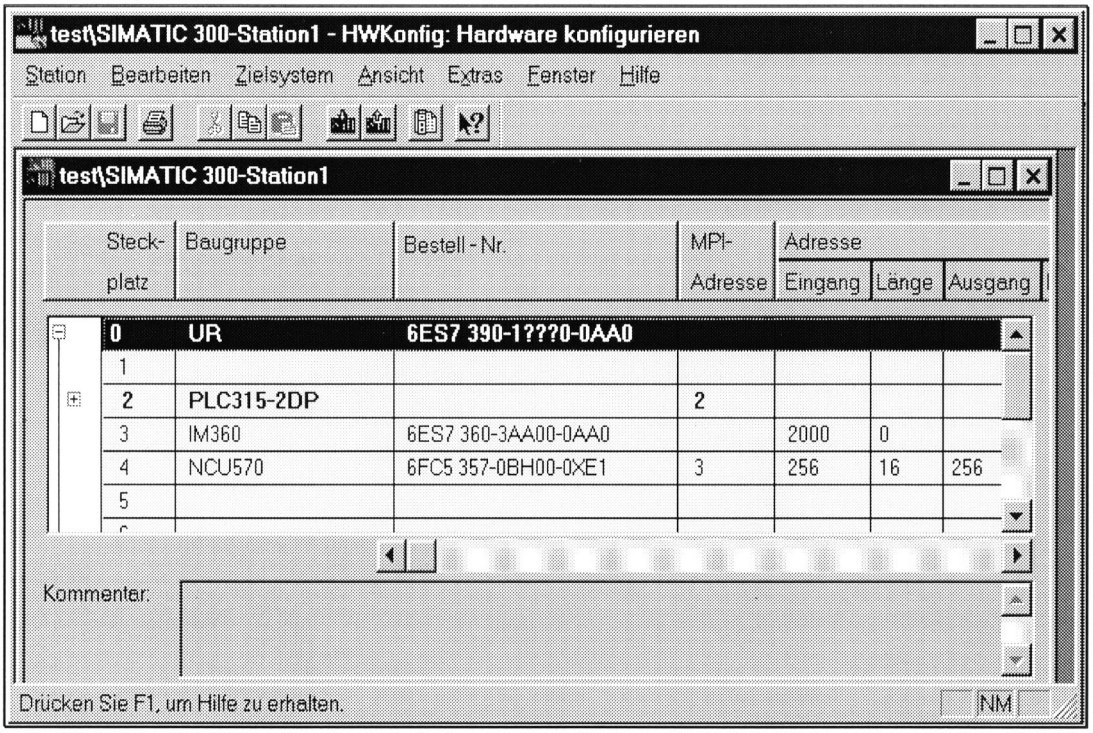


図 1.110 Rack 0 のハードウェア構成

マシン制御パネル，ハンドヘルドユニット

グローバルデータ構成

マシン制御パネルあるいはハンドヘルドユニットの構成が必要な場合（デフォルト以外の設定）は，別の SIMATIC 300 station を，マシンプロジェクトにあるそれぞれのオペレータコンポーネントに挿入してください。どのようなタイプの CPU でも，ハードウェア構成（HW config.）を使用して，この station の Rack 0 にあるスロット 2 に挿入してください。オペレータコンポーネントの希望する MPI アドレスを，MP I アドレスとして設定してください。MPI ネットワーク（1）は，SIMATIC manager においてマークすることができます。メニュー項目 Extras を介してグローバルデータがアクティブとなります。残りの手順の詳細については，Installation and Start-Up Guide を参照してください。

マシン制御パネルおよびハンドヘルドユニットの構成に関する詳細情報については，本書の 1.15.2「■マシン制御パネル，ハンドルヘルドユニットの構成可能性」を参照してください。

■ PLC プログラムのスタートアップ

■ 基本プログラムのインストール

NCU コンポーネントを初期スタートアップする前に，NCU および PLC を完全にリセットする必要があります。リセットを行うには，スイッチ S3 を位置 1 に，スイッチ S4 を位置 3 にあわせます。それから再びスイッチを入れます。この動作により，PLC は "hard reset request" （ハードリセット要求）となります。これにより，PLC および NC のメモリは初期化されます。

インストール

INSTALL.BAT (INSTALL1.BAT, INSTALL4.BAT) （ダブルクリック）により，ソフトウェア バージョン 3.7 および バージョン 4.2 のインストールを実行します。このプログラムは，適切な STEP 7 バージョンの基本プログラムおよび追加ファイルをインストールします。自動的にインストールを行うため，ハードウェアカタログにある TYP, GSD および Meta ファイルは，拡大してアップデートされます。YASNAC システムのハードウェアコンポーネントは，STEP 7 のもとでのハードウェア構成にも有効です。以下に示すように，ファイルを展開する必要はありません。

STEP 7 バージョン 2.x 以降では，library として添付されています。

一般事項

基本プログラムの SIMATIC project または SIMATIC library には，標準パラメータ表記，ハンドヘルドユニット用インタフェース記号および DB テンプレート，M デコーディング機能のある OB のソースプログラムが入っています。

基本プログラムの前に，STEP 7 をインストールしてください。

■ 基本プログラムのアプリケーション

新しい CPU プログラム（Turnma 1，など）を，それぞれのインストール (マシン) 用に STEP 7 ソフトウェアを使用して，プロジェクトにセットアップしてください。

注

プロジェクトのカタログ構造，およびプロジェクトとユーザプログラムの作成手順については，SIMATIC 取扱説明書の該当部分をお読みください。

■ バージョンコード

基本プログラム

制御タイプを含む基本プログラムバージョンは，ソフトウェアバージョンが 4（NCK, PLC）以降の MMC のバージョンディスプレイに出力します。

基本プログラムが初期のバージョンの場合，PLC バージョンは，ダブルワード DBD 0. データ上のデータブロック DB 17 に保存されます。データフォーマットとして 16 進数をセットしてください。バージョン 3.2 の場合どのように表示されるかを以下に示します。

DB17.DBD0: 0332_0100

プログラムのバージョンは，10 進法で 3 桁目と 4 桁目 (ボールド文字) で表されています。10 進数の左から 1 桁目と 2 桁目は，基本プログラムの制御タイプを表しています。その他の 10 進数は開発コードです。

制御タイプは以下のようにコード化されています。

ユーザプログラム

ユーザー自身のプログラムのバージョンコードを MMC に表示することができます。

この表示をするには，最大 54 文字の STRING タイプのデータを任意のデータブロックに定義してください。データには，ユーザーの選択したテキストを収めることができます。この文字列のためのパラメータ割当ては，FB 1 にあるポインタを介して行います。パラメータ化するには，データブロックのシンボリック定義をする必要があります。詳しくは，FB 1 の説明書を参照してください。

■ マシンプログラム

マシン製造業者は，基本プログラムに添付されているライブラリルーチンを使用して，マシンプログラムを作成しています。このマシンプログラムには，マシンの論理演算およびシーケンスがあります。NC へのインタフェース信号も，このプログラムで制御しています。NC とのより複雑なコミュニケーション機能（NC データの読取り／書込み，ツール管理確認応答，など）は，基本プログラムの FC および FB を介して，起動して実行されます。STL, ladder diagram, FBD, HIGRAPH, S7GRAPH, SCL などの異なった言語でも，マシンプログラムを生成することができます。完全なマシンプログラムは，正しいシーケンスで生成およびコンパイルしてください。つまり，他のブロックから呼び出されたブロックは，通常呼出しをかけたブロックの前にコンパイルしなければなりません。プログラム形成時に，他のブロックから呼び出されたブロックを，引き続きインタフェース内で変換した場合（VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR），呼出したブロックおよびこれに関連するすべてのブロックは，再度コンパイルしなければなりません。この通常の手順は，FB のインスタンスデータブロックの場合にも同様に適用されます。この操作手順に従わない場合，データを STEP 7 に再度翻訳する際，タイムスタンプのコンフリクトが発生します。したがって，場合によっては，例えば，"Block status" 機能などによって問題が発生して，ブロックの再翻訳ができないことがあります。ブロックの作成をラダーブロック図あるいは単一ステートメント（インクリメントモード）で行った場合，STEP 7 エディタを使用して ASCII-STL でブロックを作成することをおすすめします。

■ データのセーブ

PLC CPU は，シンボリック名を保存していませんが，ブロックパラメータのデータタイプ記述（VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR）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ています。したがって，このマシンの関連プロジェクトなしでは，ブロックは有意義な再翻訳を行うことができません（"Block status" 機能に対して，あるいは PLC CPU プログラムの引続き要求される変換，など）。このため，マシンの PLC-CPU にある STEP 7 プロジェクトのバックアップコピーを保存しておく必要があります。これにより時間を節約し，問題発生を抑え，非常に有益な効果が得られます。すでに STEP 7 プロジェクトが存在して，前述の手順ですでに作成済みの場合は，このマシンの PLC-CPU で記号を処理することができます。今後，マシンソースプログラムのアップグレードが必要な場合のため，STL ファイルとして保存することをおすすめします。

すべての構成ブロック，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は , 常に使用可能でなければなりません。

■ ソフトウェアのアップグレード

ソフトウェアのアップグレード

いかなるソフトウェアをアップグレードする場合でも，必ず事前に PLC のメモリをクリアしてください。PLC 総リセットを使用して，この初期クリア状態となります。この種のリセットで，現在あるすべてのブロックは削除されます。

一般に，新しいバージョンの NC ソフトウェアをインストールする場合，基本プログラムも新しいものを組み込む必要があります。このために，基本プログラムブロックを，ユーザプロジェクトにロードしてください。FC 12 および DB 4 がすでにユーザプロジェクトに含まれている場合は，ロードしないでください。これらは，ユーザによって変更されたブロックです。新しい基本プログラムを，ユーザプログラムとリンクしてください。これは以下の手順で行ってください。

1. 基本プログラムをコピーする前に，すべてのユーザブロックのテキストあるいはソースファイルを生成してください。
2. 新しい基本プログラムブロックをこのマシンプロジェクトにコピーしてください（基本プログラムのアプリケーション」の項を参照してください）。
3. すべてのユーザプログラム *.awl を，必ず正しい手順で再翻訳してください（「マシンプログラム」の項を参照してください）。

   新しくコンパイルしたマシンプログラムは，STEP 7 を使用して PLC CPU にロードしてください。

しかしながら，マシンプログラムの構成ブロック（OB）およびインスタンスデータブロックを再度コンパイルすれば，通常では十分です。つまり，構成ブロックおよびインスタンスデータブロックのソースの生成のみ行う必要があります（アップグレードの前に）。

総リセット

PLC の総リセットの方法については，Installation and Start-Up Guide （インストールおよびスタートアップガイド）を参照してください。ただし，総リセットでは，診断バッファの内容，および MPI バスにあるノードアドレスも削除できません。

総リセットの別の方法を以下に示します。通常の総リセットのプロセスでは実行されない場合に，この方法で行ってください。

手順は以下のとおりです。

番号	動作	結果
1	コントロールシステムのスイッチはオフ	
2	PLC スイッチを 3 (MRES) にセットして，コントロールをオンに再び切換える，またはハードウェアのリセットを実行。	"PS" とラベル表示された LED がゆっくりと点滅
3	PLC スタートアップスイッチを位置 2 (STOP) にセットして，位置 3 (MRES) へ戻す	"PS" とラベル表示された LED の点滅が速くなる
4	PLC スタートアップスイッチを位置 2 または 0 にセットする	

NC 変数

最新の NC VAR セレクタは，NC ソフトウェアのどのバージョンにも使用できます（初期のバージョンを含む）。NC ソフトウェアの初期のバージョンでも，最新のリストから変数を選択することができます。DB 120 のデータ内容（変数用のデフォルト DB）は，ソフトウェアのバージョンの新旧に依存しません。すなわち，古いバージョンのソフトウェアで選択した変数を，ソフトウェアをアップグレードした時に，再選択する必要はありません。

■ I/O モジュール (FM, CP モジュール)

通常 STEP 7 のスペシャルパッケージは，より複雑な I/O モジュールのために必要となります。これらのスペシャルパッケージのいくつかは，STEP 7 ライブラリにサポートブロック（FC, FB）を保存しています。このブロックには，ユーザプログラムによってパラメータ化され，呼出される関連モジュールを操作するための機能があります。多くの場合，CP および FM ハンドリングブロックの FC 番号は，YS840DI システム用基本プログラムの番号の範囲内にあります。

これらのコンフリクトはどのように解くことができるのでしょうか。

基本プログラムのブロック番号は，変更しないでください。ハンドリングブロックのブロック番号は，STEP 7 を使用して好きな数字を新しく割当てることができます。新しい FC 番号のついた新規ブロックは，機能によって要求されているパラメータ割当てを行い，ユーザプログラムに呼出します。

■ エラー消去

ここでは，予想されるエラーの原因と処置について説明します。ハードウェアを交換する前には，よくお読みください。

表 1.29 エラーの原因／説明および処置

シリアル番号エラーコード	エラー	原因／説明	処置
1	MPI を介して PLC への接続が確立しない	MPI のケーブルが差込まれていないか，ケーブルの不良。	テスト：接続 "Diret_PLC" を使用して，STEP 7 エディタ内でプログラマがリンクを作成する。ノードアドレスの番号がここに表示される。表示されない場合は，MPI ケーブルの不良，あるいはケーブルが差込まれていない。
		MPI カード用 STEP 7 ソフトウェアが正しく構成されていない。	STEP 7 の Windows 95 バージョンでは，ハードウェアは制御パネルでチェックする。PC ／プログラマインタフェース構成も同じようにチェックする。
2	PLC を総リセットしても PLC にアクセスできない	システムデータブロック SDB 0 が変更された MPl アドレスでロードされた。このため，アドレスの二重割当てにより MPl バスのコンフリクトが生じている。	すべての MPI ケーブルを他のコンポーネントから離す。プログラマが "Direct_PLC" リンクを作成する。MPI アドレスを訂正する。
3	PLC の LED が 4 個ともすべて点滅する（DI 故障）	PLC にシステムエラーが生じている。 対策：システムエラーの詳細を分析するために，PLC 上の診断バッファを読む。このバッファにアクセスするには，PLC を「停止」すること（例えば，スイッチ S4 を位置２ にセットする）。続いてハードウェアをリセットする。この結果 STEP 7 によって診断バッファが読み出される。情報が，診断バッファから Hotline/ Development Service へ中継される。ハードウェアのリセット後に要求しても，総リセットは続行される。そして，PLC が「停止」状態で診断バッファを読むことができる。	PLC プログラムのリセットあるいは再ロードの後，システムは再び正常に走る。この場合でも，診断バッファの内容は Development Office へ送ること。

■ PLC-CPU から YS840DI へのリンク

一般事項

AS 300 系は，PLC としてすべてのシステムに使用されます。それぞれの NCU 型の本質的な違いは，リンクの方法にあります。YS840DI では，PLC 314 CPU（ユーザメモリの最大容量が 128 K バイト）または PLC 315-2 DP（ユーザメモリの最大容量が 288 K バイト）は，サブモジュールとして NC 装置に統合されています。

PLC/CPU 315-2 DP は，Profibus (L2DP) 上の分散 I/O をサポートします。個々の PLC CPU 用の関連パフォーマンスデータについては，前述の表または AUT 1 カタログの該当個所を参照してください。

■ 統合 PLC のある YS840DI 上のインタフェース

物理的インタフェース

YS840DI システムは統合 PLC があるため，デュアルポート RAM（図 1.111）を介して，NCK と PLC の間で直接信号の交換をすることができます。

OP および MCP とのデータ交換

通常データは，YS840DI の操作パネル（OP），マシン制御パネル（MCP）およびハンドヘルドユニット（HHU）との間を，データ移送の役割をする COM モジュールを通して，操作パネルインタフェース（OpI(1)）を介して交換されます。

YS840DI の場合では，上記のすべての装置はマルチポイントインタフェース（MPI）上で運転することができます。

プログラミング装置は，MPI （マルチポイントインタフェース）を介して，PLC に直接接続します。

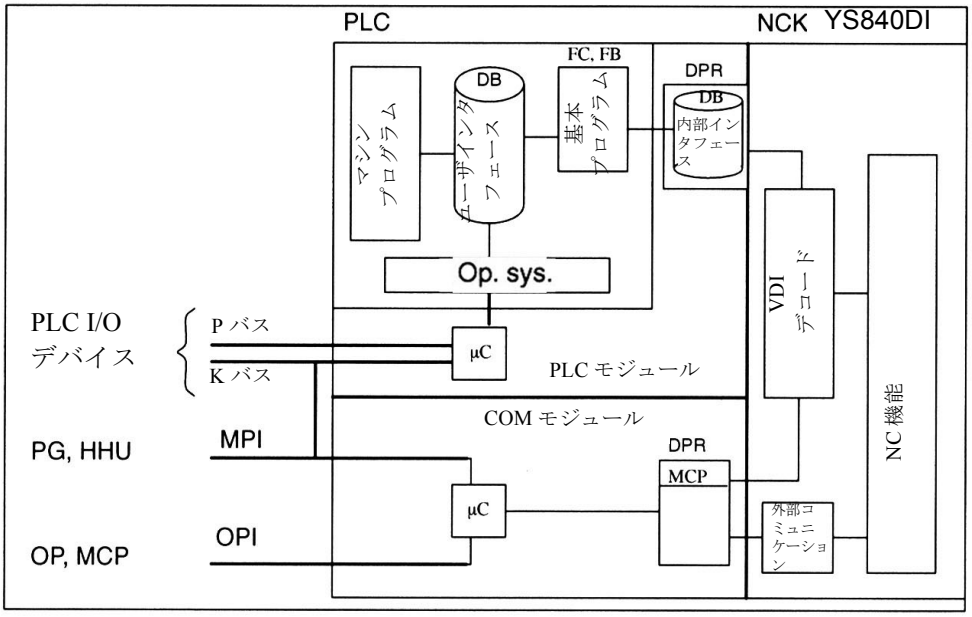


図 1.111 YS840DI （統合 PLC）の NCK/PLC カップリング

(1) OPI = Operator panel interface 運転パネルインタフェース

NCK/PLC インタフェース

NCK/PLC データ交換（図 1.111 参照）は，PLC の基本プログラムによって管理されています。ステータス情報（例えば，内部 DPR (1) に保存されている "Program running"）は，周期の初めに（OB 1），基本プログラムによってデータブロックへコピーされます。これは，ユーザがアクセスすることができます（ユーザインタフェース）。また，ユーザはインタフェースブロックにある NC 制御信号（NC スタート，など）を入力することもできます。これらの信号は周期のスタート時に NC へ転送されます。

補助機能は，ワークプログラムによって PLC へ転送されます。基本プログラムはまずこれら（アラーム駆動）を評価して，OB1 スタート時にユーザインタフェースへ転送します。関連 NC ブロックに，NC マシニングプロセス（ツールチェンジ用 M06 など）の中断を必要とする補助機能がある場合，基本プログラムは，NC のブロックの実行を PLC 周期 1 回分停止します。ユーザは，「読取りディスエーブル」インタフェース信号を使用して，ツールチェンジが完了するまでブロックの実行を停止します。一方，関連 NC ブロックに，NC マシニングプロセス（冷却オン用 M08，など）の中断を必要とする補助機能がない場合，これらの「緊急」補助機能は直接 OB 40 に転送されるので，ブロックの実行が PLC への転送によって受ける影響は極わずかなものとなります。

NCK から転送される G 機能の評価およびイネーブルもアラーム駆動ですが，ユーザインタフェースに直接転送されます。G 機能が PLC プログラムの中のいくつかのポイントにおいて評価された場合，PLC の 1 周期の中で G 機能の情報に差異が生じます。

PLC からトリガして，パラメータによって割当てられた NC 動作の場合（移動同時軸，など），インタフェースデータブロックを使用するのではなく，FC および FB を使用してトリガしてパラメータ割当てを行います。この動作に属する FC および FB は，基本プログラムとともに供給されています。FC は，ユーザでロードして，マシン製造業者の PLC プログラム（マシンプログラム）に呼び出します。

OP/PLC インタフェース

データは，OP/NC シリアルバス，COM モジュールおよび C バスを介して，OP と PL の間で交換されます。COM モジュールは，１つのバスセグメントから別のセグメントへ，完全な状態のままでデータを転送します。ボーレートの変換は極わずかです。OP は常にアクティブパートナ（クライアント）となり，PLC は常に受動パートナ（サーバ）となります。OP によって転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（タイミングは，周期制御ポイント）によって，OP/PLC インタフェースエリアで読取りと書込みが行われます。PLC アプリケーションの点からすると，データは I/O 信号と同じものです。

MCP/PLC インタフェース
HHU/PLC インタフェース

データは，MCP シリアルバス，HHU/NC，COM モジュールおよび NCK を介して，MCP/PLC と HHU/PLC の間で交換されます。NCK は，MCP/HHU 信号を転送して，内部 NC/NCK デュアルポート RAM からこれらの信号をフェッチします。PLC 側では，基本プログラムはユーザインタフェースとのコミュニケーションを操作します。機能コールのパラメータでオペランドエリア（I/O など）およびスタートアドレスを定義します。

(1) DPR = デュアルポート RAM

■ PLC 上の診断バッファ

一般事項

PLC 上の診断バッファは，PLC オペレーティングシステムに関する診断情報を収めていて，STEP 7 を使用して読み出すことができます。データは，FC 10 を介して，基本プログラムおよび アラーム／メッセージ機能によって，診断バッファに入力されます。基本プログラムによって生成され，診断バッファに保存されているアラームおよびメッセージのインタープリタについての簡単なガイドを，下記に示します。これらの アラーム／メッセージについての説明テキストは，診断バッファに入力されています。

アラーム／メッセージ

イベント ID は，診断バッファにおいて，インタープリタする最初の項目です。基本プログラムは，イベント ID の最初の 10 進数 2 桁に，以下の英数字の組み合わせを生成します。

A1 :　　アラーム設定

A0 :　　アラーム削除

B1 :　　メッセージ設定

B0 :　　メッセージ削除

イベント ID にある隣合う，2 つの 10 進数は，アラーム／メッセージ番号の左側の 10 進数 2 桁に対応しています。このイベント ID の 10 進数 2 桁の番号は，表示された 16 進数の値から 10 進数に変換してください。info 1/2/3 のデータフィールドも，評価を行ってください。この場合，info 1 および info 3 の下部の 10 進数 2 桁のみ，メッセージ番号についての有効な情報を表しています。Info 1/2/3 にあるすべての番号は，16 進数で，メッセージ番号がコンパイルされる前に，10 進数に変換してください。中央の 10 進数は info 1 から引出されており，右側 2 つの十進数は info 3 から引出されます。

（例えば，イベント ID = B133, info 1/2/3 = 2900 0 9　は，番号 510009 でメッセージの設定を表しています）。

メッセージ番号の意味は，マシン製造業者によって規定されています。16 進の値が 28, 29 のイベント ID の右側 2 つの 10 進数のみ，基本プログラムによって生成されたエラーを示しています。診断バッファに保存されたメッセージは，関連メッセージテキストによって MMC 上から読出すことができます。

■ インタフェースの構造

インタフェースデータブロック

NC 上にある PLC ユーザインタフェースは，データ量以外はそれぞれ同じです。インタフェースデータブロックでは，非常に多くの信号があるのでマッピングが必要になります。これらは，PLC プログラムの点からすると，グローバルデータブロックになります。システムのスタートアップ時に，基本プログラムは，現在のマシンデータ（チャンネル，軸の数，など）から，これらのデータブロックを作成します。この方法の有利な点は，現在のマシン構成に必要な PLC RAM の容量が最少量ですむことです。

■ PLC/NCK インタフェース

一般事項

PLC/NCK インタフェースは，一方ではデータインタフェース，もう一方では機能インタフェースを構成しています。機能インタフェースを，PLC から NCK へジョブを転送するために使う一方，データインタフェースには，ステータス信号，制御信号，補助機能および G 機能があります。

データインタフェース

データインタフェースは，さらに下記のグループに分けられます。

- NCK 別信号
- モードグループ別信号
- チャンネル別信号
- 軸／スピンドル別信号

機能インタフェース

機能インタフェースは，FB および FC から構成されています。図 1.112 は，PLC と NCK の間にあるインタフェースの構成の概略です。

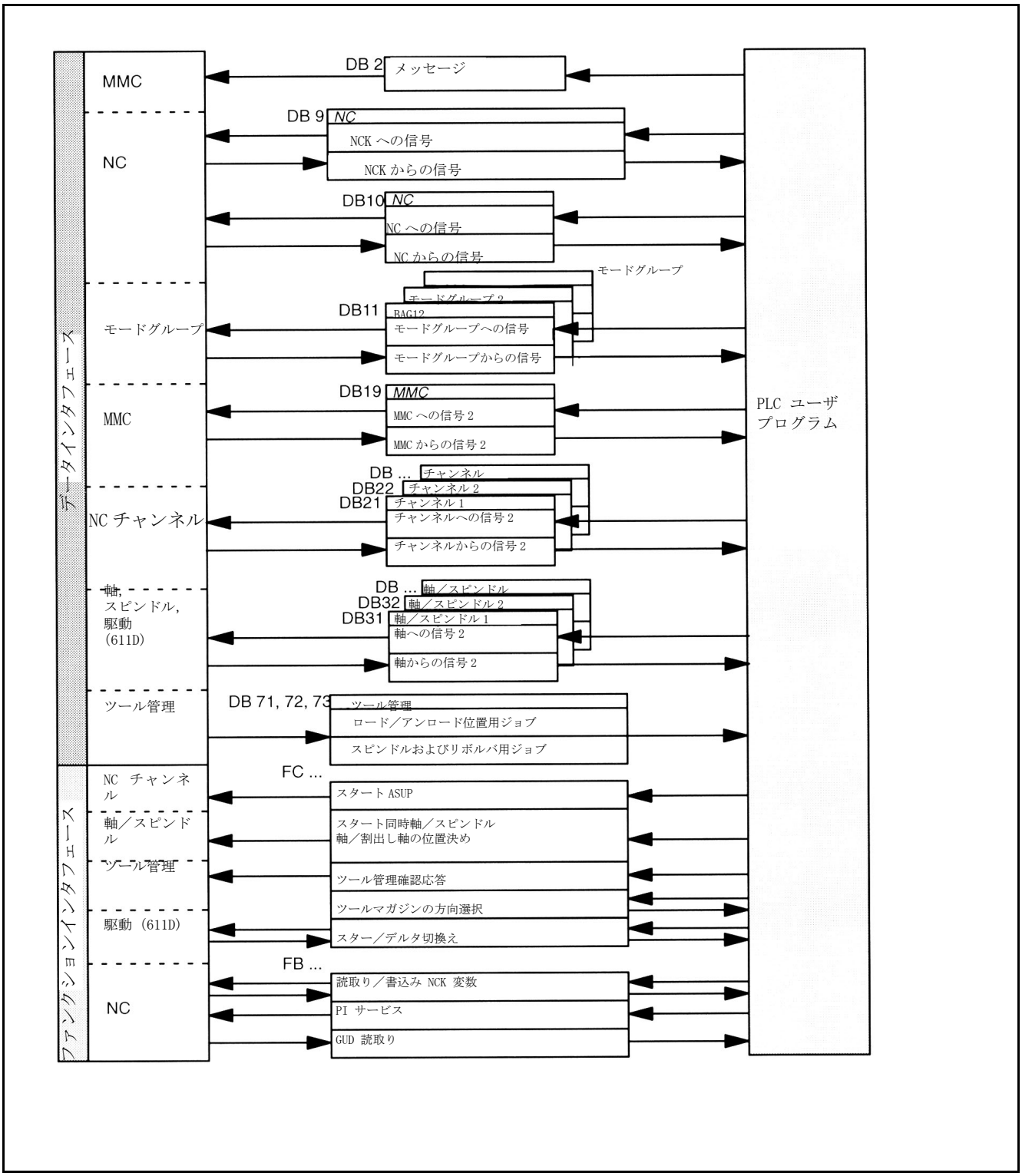


図 1.112 PLC/NCK インタフェース

コンパイル周期信号

PLC と NCK の間で交換される標準信号に加えて，コンパイル周期用インタフェース DB が，必要な時に生成されます（DB 9)。コンパイル周期に依存する信号は，OB1 スタート時に周期的に転送されます。

PLC/NC 信号

PLC から NC への信号グループは，以下のものがあります（図 1.113）

- NC の高速デジタル I/O 信号を変更する信号
- キースイッチおよび非常停止信号

NC/PLC 信号

NC から PLC への信号グループは，以下のものがあります。

- NC のデジタルおよびアナログ I/O 信号の実際値
- NC のレディおよびステータス信号

また，このグループにおける出力は，MMC のステータス信号および MMC からのハンドル選択信号です。

ハンドル選択のための信号は，基本プログラムでデコードして，マシン／軸別インタフェースに入力します。

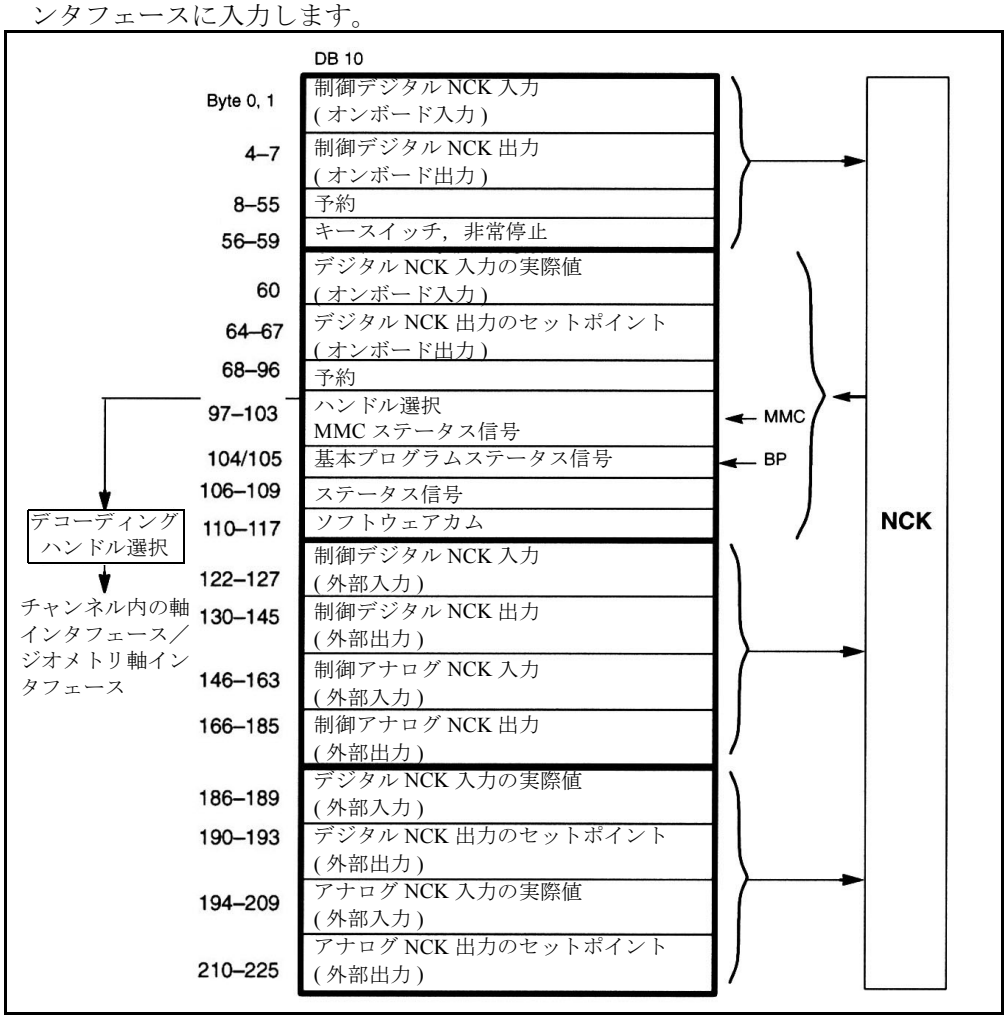


図 1.113 PLC/NC インタフェース

NCK のデジタル／アナログ入出力

NCK のディジタル／アナログ入出力に関して下記の事項に留意してください。

入力:

- NCK のすべての入力信号または入力値は，PLC へ転送されます。
- NC パートプログラムへの信号の転送は，PLC によって低減することができます。かわりに，信号または値を PLC が指定します。
- PLC は，NCK 側にそのチャンネル用ハードウェアがない場合でも，NCK へ信号または値を転送することができます。

出力:

- すべての出力信号または出力値は，PLC へ転送されます。
- NCK は，NCK 側にチャンネル用ハードウェアがない場合でも，PLC に信号または値を転送することができます。
- NCK によって転送された値は，PLC によって上書きすることができます。
- PLC からの信号および値は，NCK I/O 装置を介して直接出力することができます。

（注）デジタルおよびアナログで NCK 入出力を実行する場合は，下記の参照の内容にしたがってください。

参照： 2.1 NCK ダイレクト入出力信号（A4）

PLC ／モードグループ信号

マシン制御パネルまたは MMC から入力された運転モード信号は，運転モードグループ（モードグループ）へ転送されます。モード信号は，FM-NC, 810D および 840D 上のすべての NC チャンネルに有効となります。

モードグループは，現在のステータスを PLC に伝えます。

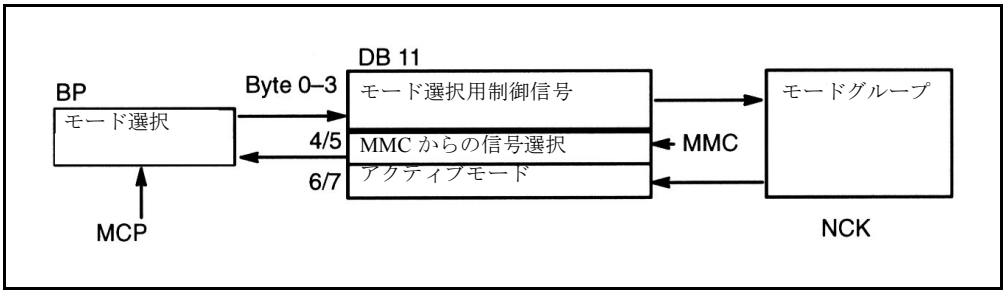


図 1.114 PLC ／モードグループ信号 (モードグループ 1，など)

PLC/NCK チャンネル信号

インタフェースには，3 つの信号グループがあります (図 1.114 参照)。

- 制御／ステータス信号
- 補助／ G 機能
- ツール管理信号
- NCK 機能

制御／ステータス機能は，OB1 のスタート時に，周期的に転送されます。信号を，マシン制御パネル上ではなく，NC 操作パネル上で定義した場合，MMC からチャンネル別のインタフェースに入力された信号 (1) も，この時転送されます（MMC 信号の転送は，PLC オペレーティングシステムによって実行されます)。

補助機能および G 機能の，インタフェースデータブロックへの入力は 2 通りあります。まず，それらはチェンジ信号によって入力されます。

- M 信号　M00 - M99 (2)（拡張アドレス 0 で NCK から転送される）はデコードされて，関連インタフェースビットが 1 周期の持続時間をセットします。
- G 機能の場合，グループは付加的にデコードされて，関連グループでアクティブとなっている G 機能が，インタフェースデータブロックに入力されます。
- S 値は，スピンドル別インタフェースへ，関連する M 信号（M03, M04, M05）とともに付加的に入力されます。軸別フィードレートもまた，適切な軸別インタフェースに入力されます。

ツール管理機能を NCK で起動する時，スピンドルまたはリボルバの割当ておよびロード／アンロードポイントは，別々のインタフェース DB に入力します（DB 71 - 73)。

NCK 機能のトリガおよびパラメータの割当ては，PLC 機能コールを使って実行します。以下の機能コールが有効となります。

- 直線軸または回転軸位置
- 割出し軸位置
- 調整済み非同期サブプログラムの起動（ASUP）
- NC 変数の読取り／書込み
- マガジン割当てのアップデート

上記機能の部分的説明については，それぞれの機能の取り扱い説明書または本書の 1.15.4「ブロック説明」を参照してください。

(1) MMC 信号は，PLC オペレーティングシステムで入力します。

(2) M 信号 M0 から M99 は，拡張アドレス 0 を使って NCK で転送してください。

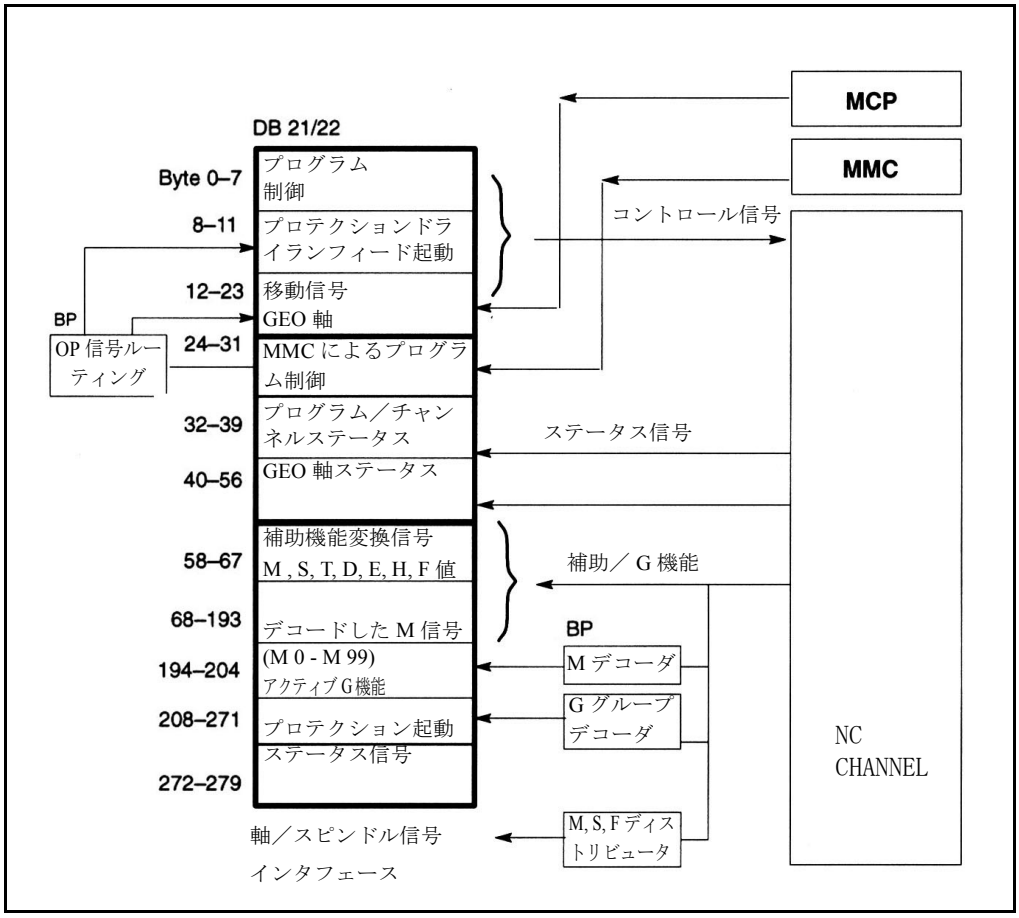


図 1.115 PLC/NC チャンネルインタフェース

PLC ／軸，スピンドル，駆動信号

軸別およびスピンドル別信号（次のページにある図 1.116）は，以下のようなグループに分けられます。

- 共有軸／スピンドル信号
- 軸信号
- スピンドル信号
- 駆動信号

信号は，OB 1 のスタート時に周期的に転送されます。以下はその例外です。

MMC の INC モード，軸方向 F 値，M/S 値

軸方向 F 値は，NC マシニングプロセスの間で PLC へ転送される場合，基本プログラムの M, S, F ディストリビュータを介して入力されます。

M および S 値は，対応する M 値（M03, M04, M05）とともに S 値が処理を要求する場合，基本プログラムの M, S, F ディストリビュータを介して入力されます。

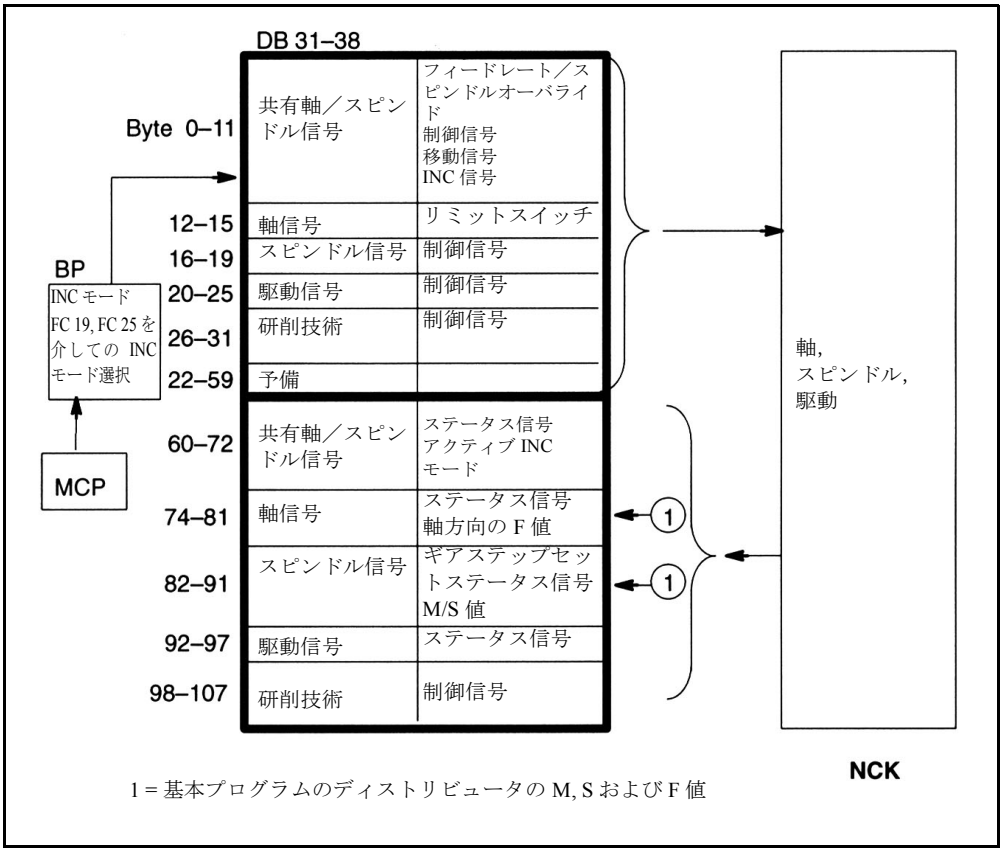


図 1.116 PLC と軸／スピンドル／駆動間のインタフェース

■ PLC/MMC インタフェース

一般事項

PLC/MMC インタフェースには，以下の機能グループが必要です。

- 制御信号
- マシン運転
- PLC メッセージ
- PLC ステータスディスプレイ

制御信号

マシン制御パネルを介して入力する信号，および MMC によって考慮しなければならない信号が，場合によってあります。この信号グループは，"Display actual values in MCS or WCS" （MCS または WCS におけるディスプレイ実際値），キーディスエーブル，などです。これらは，別々のインタフェース DB (DB 19) を介して MMC で交換します。

マシン操作

マシンを直接動作させるすべてのオペレータ入力は，PLC によってチェックされます。オペレータ動作は，通常マシン制御パネル上で行います。ただし，いくつかのオペレータ動作は，MMC 上で行うことができます (モード選択，INC モード選択，など)。

PLC オペレーティングシステムは，MMC から直接インタフェースデータブロックへ到達する運転信号を入力します（図 1.113, 1.114, 1.115, 1.116）。基本的には，運転信号をデコードする基本プログラムにより，マシン制御パネル上（オペレータアクションが有効），あるいは MMC 上で，オペレータ動作を行うことができます。必要であれば，ユーザが，FB1 のパラメータ "MMC ToIF" で，MMC を介してオペレータ動作をオフに切り換えることができます。

PLC メッセージ

シグナリング機能は，AS 300 オペレーティングシステムに統合されたシステム診断機能に基づいています。これらは，以下に示す特性があります。

- PLC オペレーティングシステムは，すべての重要なシステムステータスおよびステータス遷移を，診断ステータスリストに入力します。コミュニケーションイベントおよび I/O モジュール診断データ（診断機能のあるモジュール用）も入力します。
- システム停止を発生させる診断イベントは，その発生時間順に診断バッファ（循環バッファ）へ，タイムスタンプとともに入力します。
- 診断バッファに入力されたイベントは，いったんレディ信号（メッセージサービス）が発生すると，COM モジュールを通過し OPI，または，MPI を介して，マン―マシンインタフェース（OS または MMC）へ自動的に転送されます。ノードレディへの転送は，PLC オペレーティングシステムの機能です。MMC ソフトウェアによって，メッセージの受取りおよびインタープリタを実行します。
- SFC （システム機能コール）は，メッセージをユーザプログラムから診断バッファに入力する場合にも使用します。
- イベントは，コード化されて診断バッファに入力されます。メッセージテキストは，OP または MMC 上に保存してください。

メッセージ収集用 FC (FC 10) は，基本プログラムとともに準備されます。この FB はイベントを記録して，それらを信号グループにさらに分割し，診断バッファを介して，MMC へ報告します。

メッセージ収集の構造の概要を図 1.117 に示しています。その特徴を以下に示します。

- VDI インタフェースに関係するイベント用ビットフィールドは，ユーザメッセージ用ビットフィールドとともに 1 つのデータブロック（DB2）に統合されています。
- ビットフィールドは，FC 10 によっていくつかのレベルに評価されます。

  - 評価 1: グループ信号の収集

    少なくとも 1 つのビット信号が「1」にセットされた時，それぞれの信号のグループに対して，グループ信号が発生します。この信号は，通常 VDI インタフェース（診断機能のあるモジュール上）のディスエーブル信号にリンクされます。グループ信号は周期の中で完全に収集されます。

    - 評価 2: エラーメッセージの収集。グループの信号が，「0」から「1」へ変更することでエラーメッセージを発生するよう，あらかじめ定義されています。

  - 評価 3: 運転メッセージの収集

    グループの信号が運転メッセージを生成することが，あらかじめ定義されます。

- ユーザビットフィールドの範囲（ユーザエリア）は，8 バイトごとの 10 エリアを標準として定義します。ただし，FB 1 にある基本プログラムパラメータを介して，マシンメーカーの要望に沿うように調整することもできます。

確認応答コンセプト

下記の確認応答手順は，エラーおよび運転メッセージのために履行されます。

運転メッセージは，ユーザ用インフォメーションとして，通常運転状態を表示するためのものです。したがって確認応答信号は，このタイプのメッセージには必要ではありません。入ってくるメッセージおよび出ていくメッセージのために，診断バッファの入力が行われます。MMC には，識別子 "operating message arrived"（運転メッセージ到着）および "operating message departed" （運転メッセージ出発）を使用した，このまま運転メッセージの最新情報を載せているログがあります。

エラーメッセージは，通常，マシン停止を発生させるマシンのエラー状態の表示をするために使用されます。非常に速く連続していくつかのエラーが発生する場合，故障を診断修理するために，どのような順でエラーが起こったのかを識別できることが重要です。これは，診断バッファの入力シーケンス及び，入力された時点でのタイムスタンプによって示されます。

エラーの原因が解消されると，ユーザがメッセージを確認応答した場合のみ（マシン制御パネル上のキーを押す，など）エラーメッセージが削除されます。この信号に応答して，FC "message acquisition" （メッセージ収集）は，報告されたエラーが解消されたことを検査し，"error departed" （エラー離脱）を，診断バッファへ入力します。これにより MMC は，現在あるエラーメッセージの最新の情報を載せているログを保持できることになります。時刻は，エラー発生の時間を示したままなので，保留されているメッセージが有効になります（受け取った質問とは対照的です)。

ユーザプログラム

ユーザ PLC プログラムは，周期プログラムセクションにおいて，適切なパラメータ設定を行って基本プログラムブロック FC 10 を呼び出し（1.15.4「ブロック説明」を参照)，DB2 にあるビットフィールドをセットあるいはリセットする必要があるだけです。それ以上の必要な処置はすべて，基本プログラムおよび MMC により実行されます。

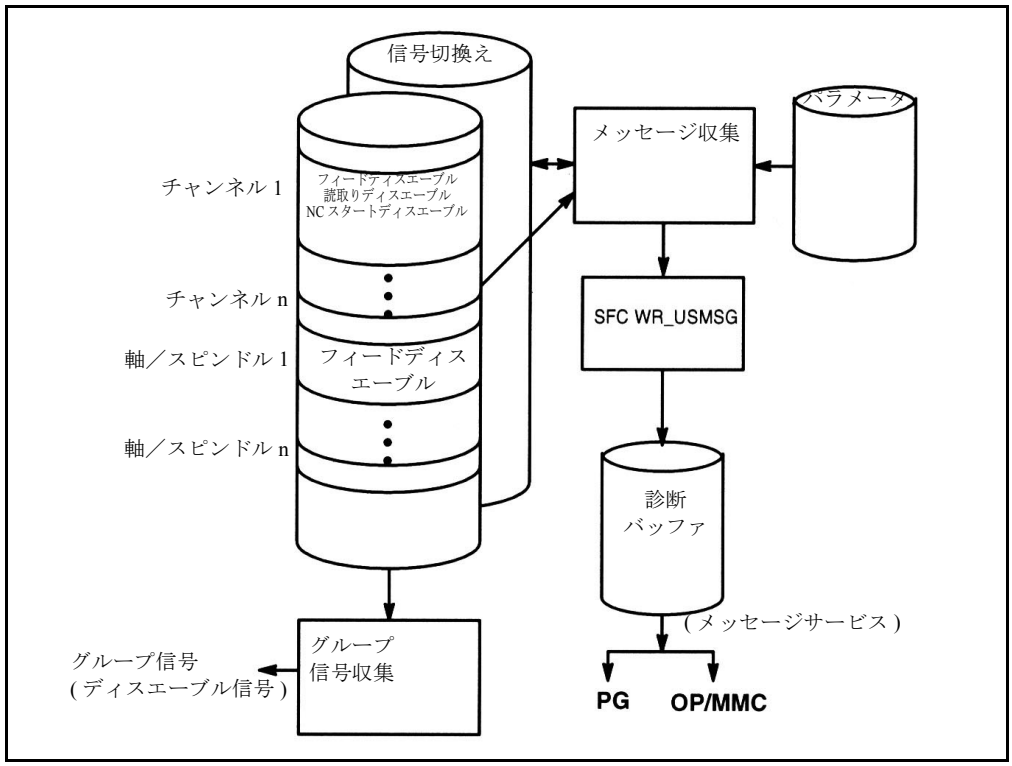


図 1.117 PLC イベント の収集およびシグナリング

PLC の MCP インタフェース

マシン制御パネルからの信号は，I/O エリアを介して標準で転送されます（図 1.118）。NC とマシン別信号の間で識別されます。NC 別キー信号は通常，FC 19 によって，さまざまなモードグループ，NCK，軸およびスピンドル別インタフェースへ配信されます。この逆がステータス信号には当てはまり，これはマシン制御パネルインタフェースへ転送されます。

（注）FC 19 は，PLC ユーザプログラムに呼び出してください。

カスタマイズされたキーは，マシン機能の広範囲をトリガするために使用することができますが，これは，ユーザプログラムによって直接評価してください。ユーザプログラムは，ステータス信号を LED 用に出力エリアへ転送します。

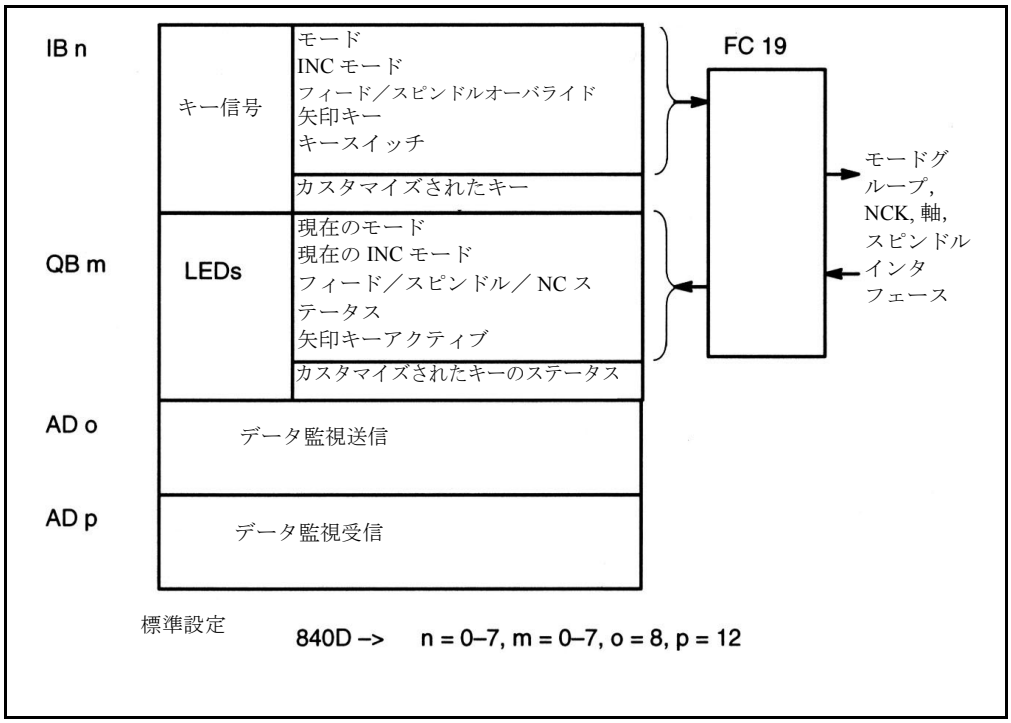


図 1.118 マシン制御パネルへのインタフェースの行き来

■ 基本プログラムの構造および機能

一般事項

プログラムはモジュラー式設計です。すなわち，NCK 機能にしたがって構成されています（図 1.119 参照）。オペレーティングシステムにおいて，下記の実行レベルの間で識別が行われます。

- スタートアップおよび同期（OB 100）
- 周期モード（OB 1）
- プロセス割込み取扱い（OB 40）

基本プログラムのそれぞれのセクション（図 1.119 で説明）は，OB 1, OB40, OB100 で，ユーザが呼び出してください。

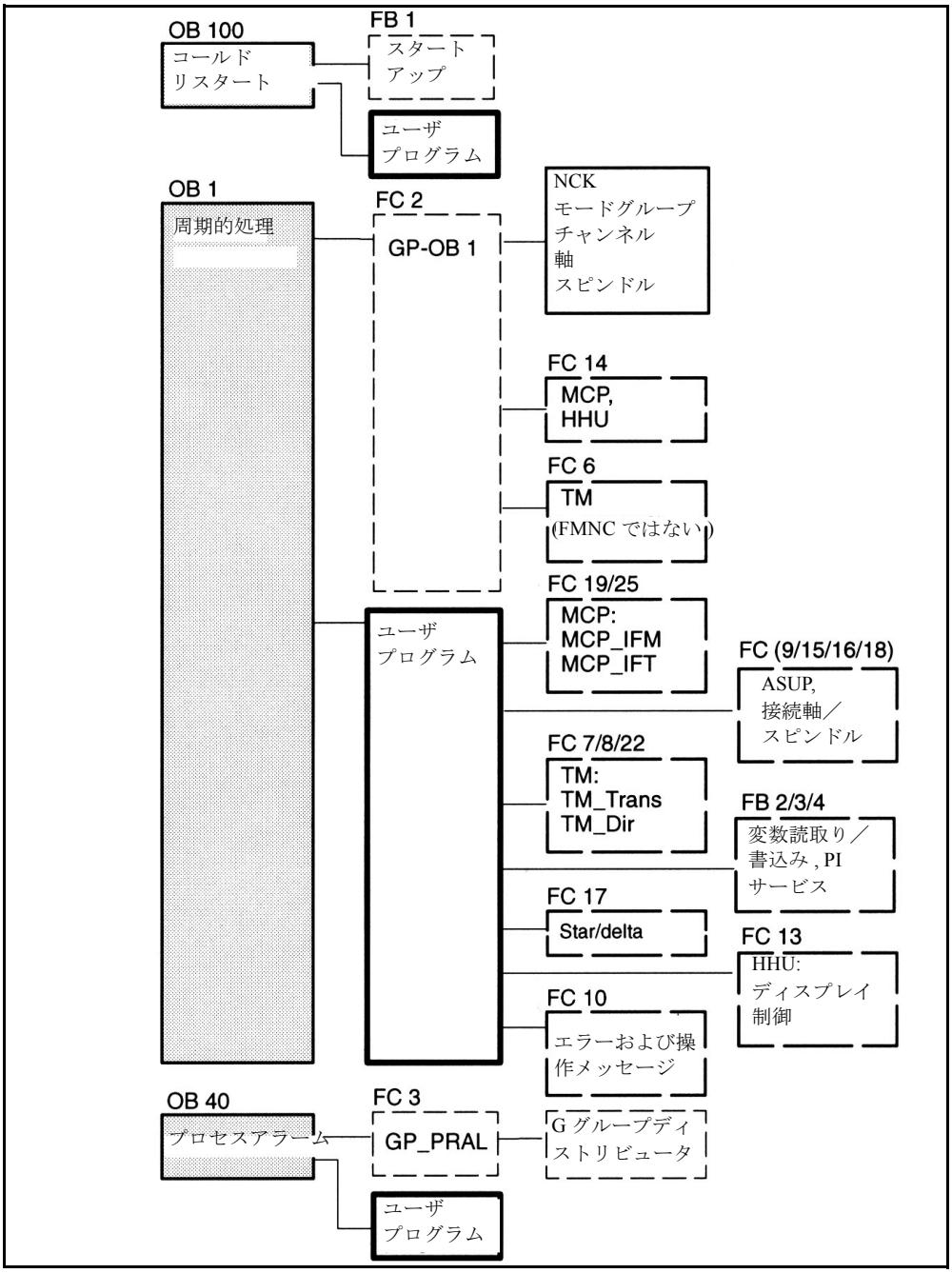


図 1.119 PLC プログラムの構造

■ NCK-PLC のスタートアップおよび同期

スタートアップ

NCK と PLC の同期は，スタートアップ時に行われます。システムおよびユーザデータブロックは，完全性を調べ，最も重要な基本プログラムパラメータが正しいかどうかを検証します。エラーの場合は，基本プログラムはエラー識別子を診断バッファへ出力して，PLC を STOP に切換えます。

ウォームスタートは備わっていません。すなわち，システム初期化に続いて，オペレーティングシステムは構成ブロック OB 100 を走らせ，OB 1 のスタート時は常に周期的実行を開始します。

同期

電源を投入している間は，PLC は MMC および NCK と同期しています。

ハートビート

正しくスタートアップして，最初の OB 1 周期（基本設定周期）が完了するのに続いて，PLC および NCK は継続的に 「ハートビート」文字をやり取りします。NCK からのハートビート信号が到着しなければ，PLC/NCK インタフェースは無効となって，DB 10 にある "NCK-CPU-ready" 信号がゼロにセットされます（1.15.2「■ NC 障害への応答」を参照)。

■ 周期運転

一般事項

NCK/PLC インタフェースを完全に実行することだけが，周期モードで行われます。時系列の観点からすると，基本プログラムはユーザプログラムの前に走ります。基本プログラムの実行時間を最短にするために，制御／ステータス信号のみが周期的に転送されます。つまり，補助機能および G 機能は，要求がある場合のみ転送されます。

以下の機能は，基本プログラムの周期的部分において実行します。

- 制御／ステータス信号の転送
- 補助機能および G 機能の分散
- M デコーディング (M00 - 99)
- M, S, F 分散
- NCK （840D のみ）を介したマシン制御パネル信号の転送
- ユーザエラーおよび運転メッセージの収集および調整

制御／ステータス信号

制御およびステータス信号の共有する特徴は，どちらもビットフィールドであることです。基本プログラムは OB 1 のスタート時にこれらをアップデートします。

信号は以下のグループにさらに分けることができます。

- 一般信号
- 運転モードなどのモードグループ別信号
- プログラムおよびフィード変更などのチャンネル別信号
- フィードディスエーブルなどの軸およびスピンドル別信号

補助機能および G 機能

補助機能および G 機能は，以下の特徴があります。

- PLC への転送は，ブロック同期です（パートプログラムブロックを参照)。
- 転送は確認応答制御です。
- 確認応答を必要としている補助機能のある NC ブロックの実行時間に，確認応答時間は，ただちに影響を及ぼします。

以下の表は数値の範囲を表しています。

機能	構成		数値の範囲		データタイプ	
	第 1 の値	第 2 の値	第 1 の値	第 2 の値	第 1 の値	第 2 の値
G 機能		G 機能		0 ~ 2551		バイト
M ワード	M グループ	M ワード	0 ~ 99	0 ~ 99.999.999	ワード	D ワード
S ワード	スピンドル番号	S ワード	0 ~ 6	浮動小数点 2	ワード	D　ワード
T ワード	マガジン番号	T ワード	0 ~ 99	0 ~ 65535	ワード	ワード
D ワード	-	D ワード	0 ~ 99	0 ~ 255	バイト	バイト
H ワード	H グループ	H ワード	0 ~ 99	浮動小数点	ワード	D ワード
F ワード	軸番号	F ワード	0 ~ 18	浮動小数点	ワード	D ワード

M, S, T, H, D, F 値は，NCK から補助／ G 機能ディストリビュータへ伝えられ，ここから，付属の切換え信号で CHANNEL DB インタフェースへ出力されます（"Lists of SINUMERIK 840D, 810D, FM-NC" （SINUMERIK 840D, 810D, FM-NC のリスト）の取扱い説明書を参照してください)。機能値および拡張アドレスは，適切なデータワードに転送されます。付属の変換信号は，1 つの PLC 周期に対して 1 つがアクティブとなります。変換信号がリセットされると，確認応答が NCK へ伝えられます。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知して即時に，基本プログラムによって行われます。

補助機能および G 機能の分散に加えて，以下の項に述べられているように選択信号を処理してください。

M デコーダ

M 機能は，切換えコマンドと固定ポイント値の両方を伝えるのに使うことができます。デコードされた動的信号は，標準 M 機能（M00 から M99）用に CHANNEL DB インタフェースへ出力されます。信号の長さ＝ 1 周期時間

1) それぞれの G グループへ伝えられる関連番号

2) STEP 7 のフォーマットで（24 ビット仮数，8 ビット指数）

G グループデコーダ

G 機能が NCK から送られる場合，グループが決められて，それぞれの 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。すなわち，すべてのアクティブな G 機能は，Channel DBs　に入力されます。NC プログラムが終了あるいは中途で停止した後でも，入力された G 機能は保持されます。

（注）システムスタートアップ時に，すべての G グループバイトは，
"0" 値で初期化します。

M, S, F ディストリビュータ

M, S, F ディストリビュータは，軸別 M ワード　M(1...6)=[3,4,5], S ワード，および軸方向フィード用 F ワードを，適切なスピンドルおよび軸データブロックに入力するために使用します。ディストリビューションの基準は拡張アドレスで，これは M ワード，S ワード および軸方向 F ワード用に PLC に伝えられます。

MCP 信号転送

840 D では，MCP 信号はシリアルバス（MPI）を介して，NCK へ転送され，そこから PLC に転送されます。基本プログラムからの機能コールは，基本プログラムパラメータによって指定された入出力のインタフェースへ，この信号を転送します。マシン制御パネルにある LED の制御用ステータス信号は，反対方向に送られます。

ユーザメッセージ

このシステムについては，すでに 1.15.2「■基本プログラムのアプリケーション」で説明しています。ユーザエラーおよび運転メッセージの収集と処理は，基本プログラムにある FC によって実行します。

■ タイムアラーム処理 (OB 35)

一般事項

時刻アラーム処理のための OB 35 は，ユーザがプログラムしてください。OB 35 のデフォルト時間基本設定は，100 ms です。その他の時間基本選定は，STEP 7 のアプリケーション "S7 Configuration" （S7 構成）を使用してください。ただし，OB 35 による時間基本設定で，約 15 ms 未満の場合は，追加の測定を必ず行って設定してください。これを行わないと，PLC の CPU が停止することになります。
MMC の実行中に，MMC システムステータスのリストを読取って停止します。この読取りプロセスは，約 8 から 12 ms に対して，優先順位段階制御を阻止します。かなり低い値の時間基本設定を OB 35 で行うと，正確な処理は実行されません。ただし，OB 35 で少量の時間基本設定をしなければならない時，少なくとも OB 80 プログラミングのプログラムコマンド "BE" で，停止を防ぐことができます。

■ プロセスアラーム処理 (OB40)

一般事項

プロセスアラーム OB 40 (割込み) は，適切な構成の I/O あるいは，ある一定の NC 機能などによって引き起こされます（1.15.2「■ FC3 : GP-PRAL 基本プログラム，アラーム駆動セクション」の FC の説明も参照してください)。割込みの原因がさまざまであるため，PLC ユーザプログラムは，まず OB 40 にある割込みの原因をインタープリタします。割込みの原因は，OB 40 のローカルデータにあります (SIMATIC STEP 7 の説明あるいは STEP 7 のオンラインヘルプを参照してください)。

■ NC 障害への応答

一般事項

周期運転の間，PLC は，ハートビート文字を照会することによって，NC の稼動可能性を継続して監視します。NCK が反応しなくなった場合，NCK/PLC インタフェースは無効となり，NC グループ（DB 10. DBX 104.7）からの信号にある IS NCK-CPU ready がリセットされます。NCK から PLC に送られる信号は，初期設定により初期化されます（以下に説明します)。
PLC そのものはアクティブのままなので，マシン機能の制御は引続き行うことができます。

NCK から PLC への信号

NCK から PLC へ転送される信号は，以下のグループに分けられます。

- NCK, チャンネル，軸およびスピンドルからのステータス信号
- 補助機能の変更信号
- 補助機能の値
- G 機能の値

ステータス信号：
NCK, チャンネル，軸およびスピンドルからのステータス信号はリセットされます。

補助機能の変換信号：

補助機能の変換信号もリセットされます。

補助機能の値：

補助機能の値は維持されるので，NCK によってトリガされた最後の機能がどれなのかを調べることができます。

G 機能の値：

G 機能の値はリセットされます（すなわち，0 値で初期化されます)。

PLC から NCK への信号

PLC から NCK へ送られる信号は，制御信号と，FC によって NCK に転送されるジョブとに分けられます。

制御信号

PLC から NCK への制御信号は，フリーズします。基本プログラムによる周期的アップデートは中断されます。

PLC から **NCK** へのジョブ

NCK へジョブを送るのに使用される FC および FB は，間違ったチェックバック信号を導く可能性があるので，PLC ユーザプログラムで処理しないでください。制御の実行中に，NCK-CPU ready 信号がセットされるまで，ユーザプログラムの中でジョブ（NCK データ読取りなど）をアクティブにしないでください。

■ ユーザプログラムから呼び出す基本プログラムの機能

一般事項

OB 1, 40 および 100 のスタート時に呼び出す基本プログラムのモジュールに加えて，基本プログラムの機能も，ユーザプログラムのふさわしいポイントに呼び出して，パラメータとともに提供します。これらの機能は，例えば，下記のジョブを PLC から NCK へ送る時に使用してください。

- 移動同時軸（FC 15, FC 16）
- 非同期サブプログラムスタート（ASUPs）(FC 9)
- NC プログラムおよび NC ブロック選択（FB 4）
- スピンドルの制御（FC 18）
- 読取り／書込み変数 (FB 2, FB 3).

（注）以下の注記は，機能コール（基本プログラムの FC, FB）の点検および診断の際に役立ててください。これらは，トリガ信号（パラメータ Req, Start を介する，などの）によって制御されている FC および FB であり，またこれらは，出力パラメータ（パラメータ Done, NDR, Error を介する , などの）として確認応答を実行する FC および FB です。機能コールをトリガする他の信号のコンパイルされた変数をセットしなければなりません。スタート状態は，パラメータ Done, NDR および Error ステータスの機能としてのみリセットされます。この制御メカニズムは，機能コールの前，あるいは後ろに位置付けされます。メカニズムがコールの後に置かれた場合，出力変数はローカル変数として定義することができます（この利点は，グローバル変数，マーカー，データ変数の削減といった，データ変数上の時間関連の優位点です)。トリガパラメータ

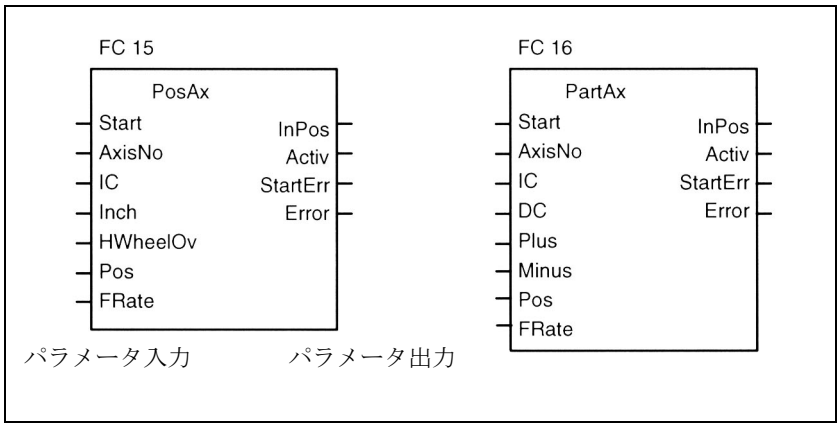
は，グローバル変数にしてください（マーカー，データ変数，など）。
ユーザプログラムによって起動したジョブ（パラメータ Req, Start,...:=TRUE）は，OB 100 で命名したパラメータ用に，ゼロにリセットしてください。電源オフ／オンで，ジョブがまだアクティブである状態にできます。

同時軸

以下は同時軸の主な特徴です。

- PLC，NC どちらからでも移動できます。
- PLC 上の機能コールから，あらゆる運転モードでスタートできます。
- NC ブロック境界に依存しないスタートです。

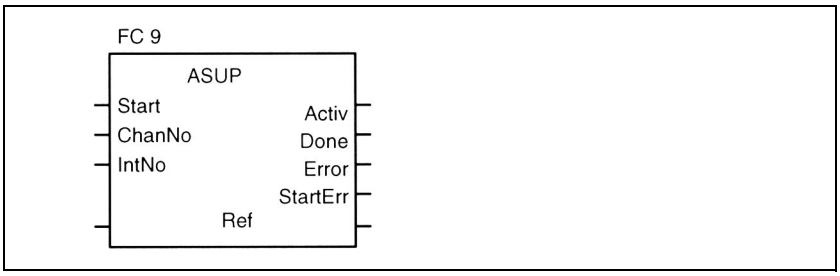
機能コールは，位置決め（FC 15）および割出し軸（FC 16）に有効です。



ASUP

非同期サブプログラム (ASUP) は，NC において選択された機能を起動するために使用します。PLC から非同期サブプログラムをスタートさせる前に，このプログラムが有効で，NC プログラムあるいは FB 4 PI サービス（ASUP）によって準備したものであることを確認してください。ASUP は，パートプログラムを走らせて，MDA あるいは Automatic モードでのみスタートすることができます。

この方法で準備すると，いつでも PLC からスタートすることができます。このチャンネル上を走っている 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。非同期サブプログラムは FC 9 でスタートします。



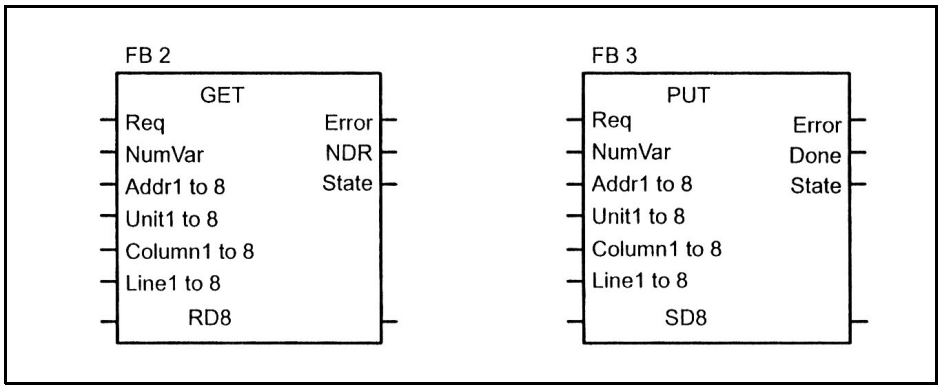
（注）非同期サブプログラムが，NC プログラムあるいは FB 4(ASUP) によって準備されなかった場合（例えば，割込み番号がひとつも割当てられていない場合），スタートエラーが出力されます。

NC 変数　読取り／書込み

FB PUT で NCK 変数に値を入力するのに対して，FB GET で NCK の変数を読取ることができます。NCK 変数を，入力 Addr 1 から Addr 8 にある識別子を介してアドレス指定します。識別子 (記号) は，グローバル DB に保存されているアドレスデータを指します。この DB を生成するには，PC ソフトウェアにある基本プログラムの中の表から，必要な変数を選択することができます。選択された変数は，最初にプロジェクト関連のリストにすぐ集められます。コマンド Generate DB で *.STL ファイルを作成します。そのマシン用に，*.STL ファイルをプログラムファイルにリンクして，マシンプログラムとともにコンパイルしてください（@セクション 3.2 も参照してください)。

読取りあるいは書込みジョブで，1 から 8 の値を，読取りあるいは書込むことができます。必要な場合は，その値を変換してください（例えば，NCK 浮動小数点値（64 ビット）を，24 ビット 仮数部および 8 ビット 指数の合計 32 ビットの PLC フォーマットに変換します。この逆の場合も同様です)。64 ビットから 32 ビット REAL への変換をすると，正確さが損なわれます。32 ビット REAL 番号の最大精密度は，7 のパワーに対しておよそ 10 です。

FB 2 および FB 3 については，本書の 1.15.4「ブロック説明」を参照してください。



■ インタフェース DB を用いたユーザプログラムのシンボリックプログラミング

一般事項

（注）基本プログラム バージョン 3.2 以降では，ファイル NST_UDT.STL および TM_UDT.AWL は，システムに添付されている基本プログラムディスケットに保存されています。

これらの 2 つのファイルからコンパイルされた UDT ブロックは，基本プログラムの CPU プログラムに保存されています。

UDT は，ユーザによって定義することのできるタイプのデータです。例えば，このデータを，CPU において生成されたデータブロックに割当てることができます。

ほとんどすべてのインタフェース信号のシンボリック名は，これらの UDT ブロックにおいて定義します。

UDT 番号 には，2, 10, 11, 19, 21, 31, 71, 72, 73 を使用します。

割当ては，下記のとおりです。

表 1.30 UDT 割当て

UDT 番号	インタフェース DB への割当て	意味
UDT 2	DB 2	アラーム／メッセージ
UDT 10	DB 10	NCK 信号
UDT 11	DB 11	モードグループ信号
UDT 19	DB 19	MMC 信号
UDT 21	DB 21 から DB 30	チャンネル信号

表 1.31 UDT 割当て

UDT 番号	インタフェース DB への割当て	意味
UDT 31	DB 31 から DB 61	軸／スピンドル信号
UDT 71	DB 71	ツール管理：ロード／アンロード位置
UDT 72	DB 72	ツール管理／スピンドルに切換え
UDT73	DB 73	ツール管理：リボルバに切換え

インタフェース信号を記号としてプログラムするために，インタフェースのデータブロックは，まず記号エディタでシンボリックに割当てる必要があります。

例えば，記号 "AxisX" は，記号ファイルにあるデータタイプ UDT31 で，オペランド DB31 に割当ててください。

この入力の後，STEP 7 プログラムは，このインタフェース用記号でプログラムすることができます。

（注）上記のインタフェース DB に用いるプログラムが，初期のバージョンのソフトウェアで作成されたものでも，「記号」プログラムに変換することができます。
ただし，これを行うには，初期のプログラムにデータアクセスするために，十分に的確な命令が必要となります（例えば，"U DB31.DBX 60.0" このコマンドは，シンボリック機能がエディタ内で起動した時，"AxisX.E_SpKA" に変換されます）。

説明

インタフェース信号の省略されたシンボル名は，NST_UDT.AWL と TM_UDT.AWL の，2 つの STL ファイルに定義されます。

インタフェース信号の名前の参照を作成するためには，それぞれの信号の後，コメントの中に名前を入れてください。

UDT ブロックを開いた時に，STEP 7 エディタコマンドを使って，シンボル名，コマンドおよびすべてのアドレスを見ることができます。

（注）使用しないビットおよびバイトが示されます。例えば，
"f56_3" と指定されます。
"56" は，関連データブロックの関連バイトアドレスを指定します。
"3" この数字は，このバイトにあるビット番号に対応します。

UDT インタフェースの英語版は，NST_UDTB.AWL および TM_UDTB.AWL で，SW 4.4 以降に使用できます。また，英語版のみ今後も新版が開発されます。以前使用していたドイツ語の記号を，英語に変換する時に必要な手順を次に示します。ソースなしで作業している場合，NST_UDTB.AWL および TM_UDTB.AWL ブロックをコンパイルするだけで，新しい記号がすぐに現れます。

ソースを使用している場合，ソースをまず最初に，以前の UDT(NST_UDT.AWL および TM_UDT.AWL) でコンパイルしてください。つぎに，NST_UDTB.AWL および TM_UDTB.AWL でコンパイルしてください。以上を終えてから，以前のソースのセットアップにしたがって，逆にソースの状態にコンパイルを開始してください。この結果，インタフェース記号は，永久に英語に変換されます。

■ リストによる M デコーディング

機能説明

リストによる M デコーディング機能が，FB1 "ListMDecGrp" の BP パラメータを介して起動した時，拡張アドレスのある 256 までの M 機能が，基本プログラムでデコードできます。

拡張アドレスのある M 機能と，信号リストにセットされたビットとの間の割当ては，デコーディングリストに定義します。このために，信号をグループ分けします。

信号リストには，それぞれデコードされた信号として，16 ビットの 16 グループがあります。

1 つのデコーディングリストと 1 つの信号リストがあるのみです。つまり，クロスチャンネル機能となっています。

M 機能をデコードします。いったんデコーディングリストに入力されると，信号リストにある関連ビットがセットされます。

ビットが信号リストにセットされると，関連 NC チャンネルにある読取りディスエーブルが，基本プログラムによって同時にセットされます。

ユーザがすべてのビット出力をこのチャンネルによってリセットして，このことを確認応答すれば，チャンネルの読取りディスエーブルはリセットされます。

高速補助機能としてリストにデコードされた M 機能の出力は，読取りディスエーブルにはなりません。

下記の図 1.120 は，リストによる M デコーディング機能の構造です。

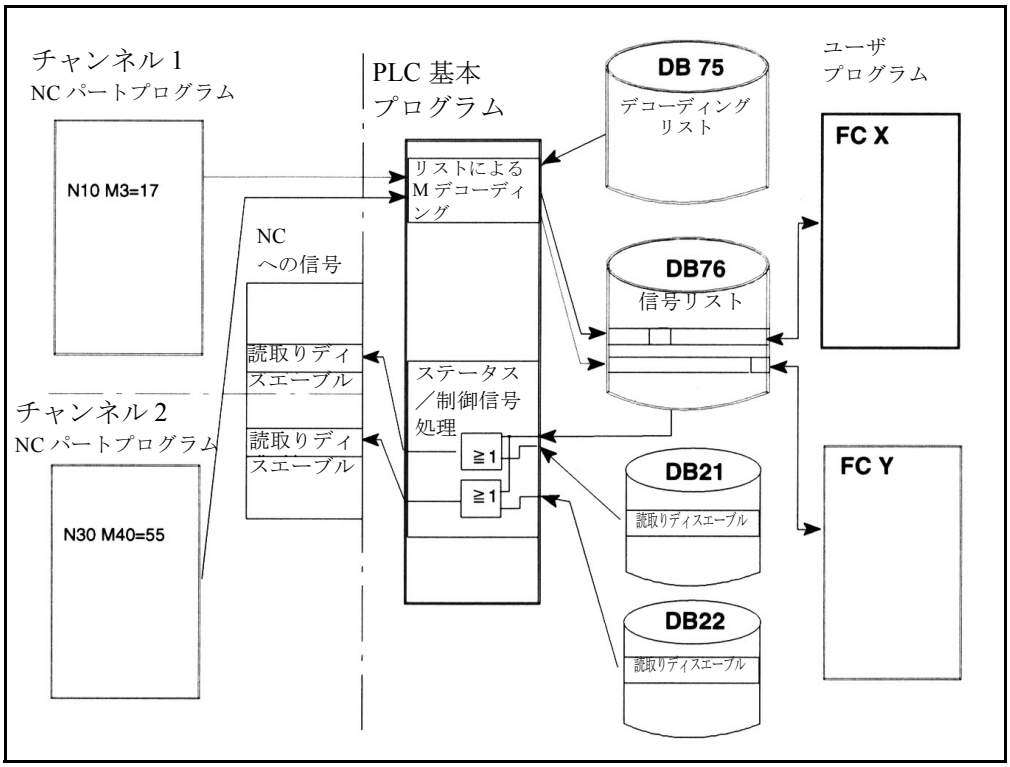


図 1.120 リストによる M デコーディング

デコーディングリストの構造

デコードされる M 機能のすべてのグループは，デコーディングリスト DB 75 にエントリしてください。

最大で 16 グループ作成できます。

デコード信号のリストにあるそれぞれのグループに 16 ビットが使用できます。

拡張アドレスのある M 機能と，信号リストにセットされたビットとの間の割当ては，デコーディングリストの最初と最後の M 機能を介して指定します。

最初の M 機能（"MfirstAdr"）から最後の M 機能（"MlastAdr"）までが，それぞれのグループのビット 0 から最大でビット 15 までに対応して，ビットアドレスが生成されます。

デコーディングリストにあるそれぞれのエントリは，3 個のパラメータから構成され，それぞれのグループに割当てられます。

デコーディングリストのソースファイル（MDECLIST:AWL）は，基本プログラムに添付しています。

表 1.32 グループの割当て

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

表 1.33 信号のタイプと数値の範囲

信号	タイプ	数値の範囲	メモ
MExtAdr	Int	0 から 99	拡張 M アドレス
MFirstAdr	DInt	0 から 99.999.999	グループの最初の M アドレス
MLastAdr	Dint	0 から 99.999.999	グループの最後の M アドレス

例

```
DATA_BLOCK DB 75
TITLE =
VERSION : 0.0
    STRUCT
    MSigGrp : ARRAY [1 .. 16 ] OF STRUCT
        MExtAdr :       INT ;
        MFirstAdr :     DINT ;
        MLastAdr :      DINT ;
        END_STRUCT ;
        END_STRUCT ;
BEGIN
    MSigGrp[1].MExtAdr :=           0;
    MSigGrp[1].MFirstAdr :=         L#0;
    MSigGrp[1].MLastAdr :=          L#0;
    MSigGrp[2].MExtAdr :=           0;
    MSigGrp[2].MFirstAdr :=         L#0;
    MSigGrp[2].MLastAdr :=          L#0;
    MSigGrp[15].MExtAdr :=          0;
    MSigGrp[15].MFirstAdr :=        L#0;
    MSigGrp[15].MLastAdr :=         L#0;
    MSigGrp[16].MExtAdr :=          0;
    MSigGrp[16].MFirstAdr :=        L#0;
    MSigGrp[16].MLastAdr :=         L#0;
END_DATA_BLOCK
```

信号リスト

機能の起動時，データブロック DB 76 をセットします。

リストでデコードされた M 信号用に，DB 76 の適切なグループにビットをセットします。

同時に読取りディスエーブルを，M 機能が出力されたチャンネル内にセットします（上記の通り）。

機能の起動

評価されるグループの数は，基本プログラムパラメータ ListMGDecGrp で指定します（FB 1 の説明をお読みください）。

デコーディング機能は，その値が 1 から 16 までの場合に起動します。

デコーディングを起動する前に，デコーディングリスト DB 75 を再スタートに続いて PLC に転送してください。

例

下記の表 1.34 は，プログラミングのためのパラメータ例です。

表 1.34　パラメータ例

グループ	デコーディングリスト (DB 75)			信号リスト
	拡張 M アドレス	最初の M アドレス	グループの最後の M アドレス	DB 76
1	2	1	5	DBX0.0 to DBX0.4
2	3	12	23	DBX2.0 to DBX3.3
3	40	55	55	DBX4.0

```
DATA_BLOCK DB 75
TITLE =
VERSION : 0.0
    STRUCT
    MSigGrp : ARRAY [1 .. 16 ] OF STRUCT
            MExtAdr :           INT ;
            MFirstAdr :         DINT ;
            MLastAdr :          DINT ;
        END_STRUCT ;
    END_STRUCT ;
BEGIN
    MSigGrp[1].MExtAdr :=       2;
    MSigGrp[1].MFirstAdr :=     L#1;
    MSigGrp[1].MLastAdr :=      L#5;
    MSigGrp[2].MExtAdr :=       3;
    MSigGrp[2].MFirstAdr :=     L#12;
    MSigGrp[2].MLastAdr :=      L#23;
    MSigGrp[3].MExtAdr :=       40;
    MSigGrp[3].MFirstAdr :=     L#55;
    MSigGrp[3].MLastAdr :=      L#55;
END_DATA_BLOCK
```

■ PLC マシンデータ

一般事項

ユーザ用に，NC 内に PLC 別マシンデータを保存できるオプションがあります。PLC の実行中 (OB 100) にユーザはこれらのデータを処理することができます。この機能を使って，例えば，ユーザオプション，マシン拡張，マシン設定などが実行できます。

データを読取るためのインタフェースは，DB20 にあります。ただし，DB20 は，ユーザマシンデータを使用している時のみ，電源投入中に基本プログラムによってセットアップされます。すなわち，BP パラメータ UDInt, UDHex, UDReal の合計は，ゼロよりも大きくなります。個々のエリアのサイズと，DB20 のトータルの長さも，このように PLC マシンデータ（通常のマシンデータ：MAXNUM_USER_DATA_INT, MAXNUM_USER_DATA_HEX, MAXNUM_USER_DATA_FLOAT）によってセットされ，BP においてユーザ用に表示されます。

パラメータ：UDInt, UDHex, UDReal。BP は下記の順で，DB20 においてデータフラッシュを保存します。

整数 MD, 16 進数フィールド MD, リアル MD.

整数および実際値は，S7 フォーマットで DB20 に保存されます。

16 進数データは，入力順に DB20 に保存されます（ビットフィールドとして使用）。

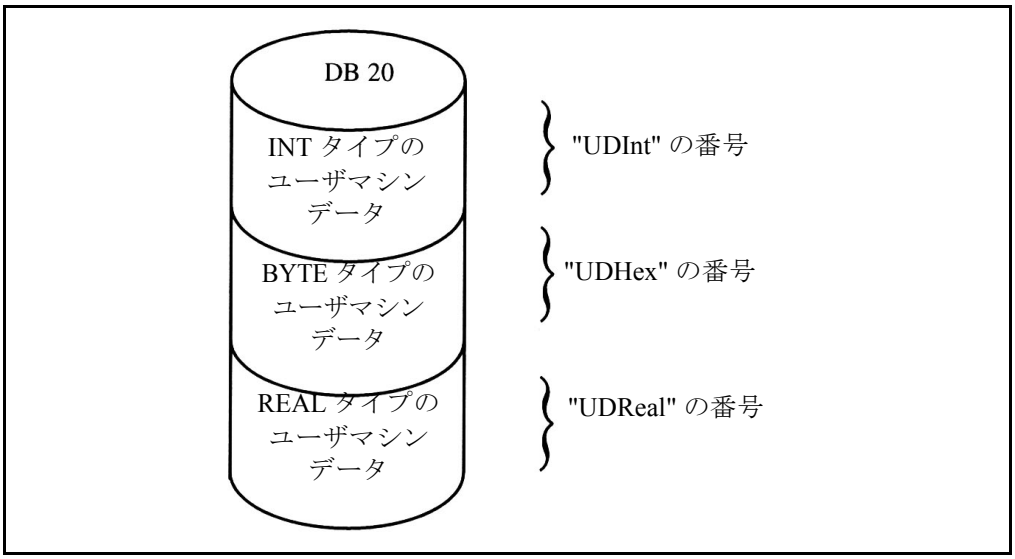


図 1.121 DB 20

（注）使用する PLC マシンデータの数が後で増える場合は，DB20 をあらかじめ削除してください。このような拡張をして使用した場合に，現在のユーザプログラムに影響を及ぼすことのないよう，DB20 にあるデータは，できる限り記号形式でアクセスしてください。例えば，UDT における構造体定義を使用するなどしてください。

表 1.35 アラーム例

400120	PLC の DB20 を削除して再スタート
説明	DB の長さが要求された DB の長さと違う
反応	アラームディスプレイおよび PLC 停止
処置	リセットにしたがって DB 20 を削除
続行	コールドリセット後

例

整数値は 4，ビット情報の 16 進数フィールドは 2，および実際値 1，の場合のプロジェクトを例にします。

マシンデータ：

51900 USER_DATA_INT[0] 123

51900 USER_DATA_INT[1] 456

51900 USER_DATA_INT[2] 789

51900 USER_DATA_INT[3] 1011

...

51902 USER_DATA_HEX[0] 12

51902 USER_DATA_HEX[1] AC

...

51904 USER_DATA_FLOAT[0] 123.456

BP パラメータ (OB100):

```
CALL fb 1, db 7(
        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;
```

BP パラメータ (実行中に走査):

```
        l gp_par.UDInt;     //=4,
        l gp_par.UDHex;     //=2,
        l gp_par.UDReal;    //=1 )
```

PLC の電源投入中は，DB20 は，28 バイトの長さでセットアップしています。

表 1.36 DB 20

アドレス	書込まれた値
0.0	123
2.0	456
4.0	789
6.0	1011
8.0	b#16#12
9.0	b#16#AC
10.0	1.234560e+02

使用するマシンデータの構造は，UDT において指定されます。

```
TYPE UDT 20
   STRUCT
    UDInt :        ARRAY [0 .. 3 ] OF INT ;
    UDHex0 :       ARRAY [0 .. 15 ] OF BOOL ;
    UDReal :       ARRAY [0 .. 0 ] OF REAL ; // 以降のエクステンション用のフィー
                                                ルドとしての説明
   END_STRUCT ;
END_TYPE
```

（注）BOOL 配列は，常に同じ番号のアドレスに送られます。なぜならば，0 から 15 の配列範囲は，通常 UDT 定義あるいは個々に指定されたすべての Boolean 変数中で選択されるべきものだからです。

REAL 値のみ，例において最初に使用していますが，変数用にフィールド（1 つの要素のある）が生成されます。こうすることにより，記号アドレスを変換することなく，今後容易に拡張ができるようになります。

記号リストへ入力すると，記号形式でデータアクセスできます。

アラーム

記号	オペランド	データタイプ
UData	DB 20	UDT 20

ユーザプログラムにおけるアクセスオペレーション（記号読取りのアクセスオペレーションのみ）

```
...
    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];
...
```

■ マシン制御パネル，ハンドヘルドユニットの構成可能性

一般情報

NC に統合されるコミュニケーションシステムは，NC とデータの交換をするために，最大で 2 個のマシン制御パネルと 1 個のハンドヘルドユニットが使用できます。

コンポーネントは，OB 100 において基本プログラムブロック FB 1 の呼出しによって，パラメータ化されます。FB 1 は，そのパラメータを関連インスタンス DB（DB 7，シンボリック名 "gp_par" ）に保存します。それぞれのマシン制御パネルとハンドヘルドユニットに対して，別々にパラメータがセットされます。ユーザ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。これらの入出力アドレスは，FC 19, FC 24, FC 25, FC 26, FC 13 でも使用します。ステータス情報に対するアドレス，MPI あるいは OPI （GD パラメータセットは，MPI アドレスではなくハンドヘルドユニット用にセットしてください）も定義してください（セクション 4 の FB 1 の説明を参照してください)。タイムアウトおよび周期的強制再トリガの時間設定は，初期設定のままになります。

起動

それぞれのコンポーネントは，マシン制御パネルの数（パラメータ MCPNum）あるいは，ハンドヘルドユニットの場合，パラメータ BHG:=2 (BHG:=1 は，SDB 210 とともに MPI インタフェースを介したリンクを示しています) のどちらかを介して起動します。コンポーネントが，OPI か MPI のどちらにリンクされるのかは，パラメータ MCPMPI と BHGMPI によって確定します。

ハンドヘルドユニット

ハンドヘルドユニットは，GD パラメータセットを使用して MPI または OPI にアドレス指定します。これらのパラメータ値は，ハンドヘルドユニットの設定にしたがって割当ててください。ただし，ハンドヘルドユニット上のパラメータ名は，基本プログラムにあるパラメータ名とは反対になっています。ハンドヘルドユニットにある Send タイプのすべてのパラメータは，基本プログラムにある Rec タイプとして（Rec タイプは，Send として）定義してください。

制御信号

パラメータ MCP1Stop, MCP2Stop, BHGStop は，個々のコンポーネントのコミュニケーションを停止するために使用します（パラメータ設定＝1）。コミュニケーションの停止または起動は，現在の周期において適用することができます。ただし，値を変換するには，他の FB 1 コールを使用するのではなく，パラメータの記号表記を通して実行してください。

第 1 のマシン制御パネルでのデータ交換を停止する例

```
SET;
S               gp_par.MCP1Stop;
```

MPI と OPI アドレス間での切換え

SW バージョン 4 以降では，マシン制御パネル (MCP) あるいはハンドヘルドユニット (HHU) との既成の接続を，切離すことができます。すでに別の MPI あるいは OPI アドレスでバスに接続している，他の MCP あるいは HHU コンポーネントが起動します。アドレスの切換えは，以下の手順にしたがってください。

1. パラメータ MCP1Stop, MCP2Stop, BHGStop = 1 を介して切離されるコンポーネントのコミュニケーションを停止します。
2. PLC 周期がバスアドレスまたは GD パラメータで変化した後，この装置を新しいコンポーネントにセットします。
3. 別の PLC 周期の後，新しいコンポーネントのコミュニケーションは，パラメータ MCP1Stop，MCP2Stop，BHGStop = 0 を使用して，再び起動することができます。

サブセクション "Control signals" （制御信号）で述べているとおり，すべてのパラメータはデータタイプにしたがってプログラムしてください。

構成

MCP/HHU と PLC との間でのデータ転送は，基本的に 2 つのコミュニケーションメカニズムがあります。1 つ目のメカニズムでは，コムモジュールを介してデータを移送します。メカニズムは，FB 1 の MCP/HHU パラメータを介して完全にパラメータで表されます。

NC タイプの機能として使用するさまざまなインタフェースオプションの概要を以下に記します。それぞれの場合，データ転送メソッドに関連する FB1 のパラメータセットおよび有効なステータス情報を指定します。

タイムアウトモニタによってエラーが検知されると，PLC-CPU の診断バッファ内で対応する入力が行われます（エラー 400260 から 400262 まで）。この場合，MCP あるいはハンドヘルドユニット（MCP1In/MCP2In または BHGIn）からの入力信号が 0 で初期化されます。PLC と MCP/HHU が再び同期できれば，コミュニケーションは自動的に再開して，エラーメッセージは BP によってリセットされます。

OPI/MPI カップリング

コミュニケーションは，NCK および COMM モジュールなどを介して，PLC BP からスタートします。つまり，MPI を介したリンクでも同じことです。オペレーションは，FB1 にある適切なパラメータを介して，パラメータで表されます。

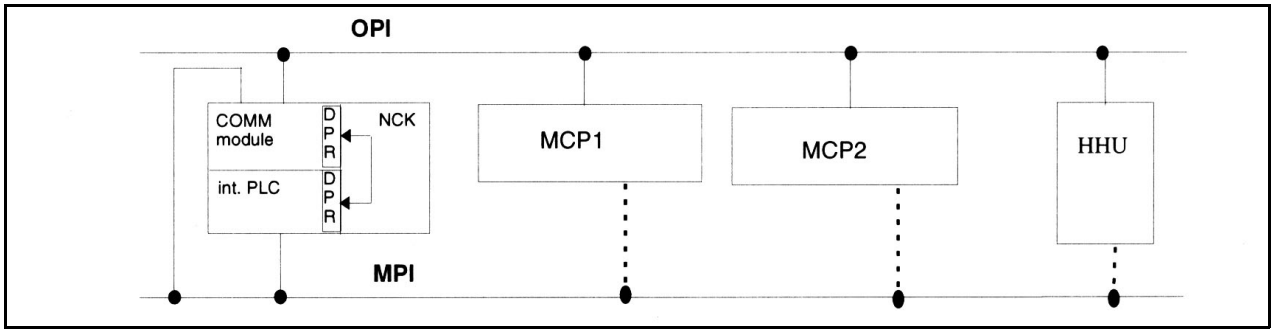


図 1.122 OPI/MPI カップリング

表 1.37 適切なパラメータ (FB1)

MCP		HHU
MCPNum=1 または 2 (MCP の数)		BHG=2 (Comm モジュールを介した転送)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend	MCP2StatSend	BHGStatSend
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOutLen
MCP1Cycl	MCP2Cycl	BHGTimeout
MCPMPI = FALSE (OPI), TRUE (MPI)		BHGCycl
MCP1Stop	MCP2Stop	BHGRecGDNo
		BHGRecGBZNo
		BHGRecObjNo
		BHGSendGDNo
		BHGSendGBZNo
		BHGSendObjNo
		BHGMPI = FALSE (OPI),TRUE (MPI)
		BHGStop

表 1.38 ステータス情報

適応機種	ビット番号	説明
MCP1StatSend MCP2StatSend BHGStatSend	4	GD パケットの構文エラー : パラメータセットのエラー (FB1)
MCP1StatSend MCP2StatSendB HGStatSend	27	送信者 : タイムアウト
MCP1StatRec MCP2StatRec BHGStatRec	10	受信者 : タイムアウト

タイムアウトモニタ (ビット 10 および 27) 用に，PLC 診断バッファでエラー入力が行われると，MMC に以下のエラーメッセージが現れます。

- 400260: MCP1 障害または
- 400261: MCP2 障害
- 400262: HHU 障害

MCP/HHU と MCP との間で，まだデータの交換が行われていなくても，コールド再スタート後に，MCP または HHU 障害（SW バージョン 4 以降）は，即時に検知されます。電源投入後，すべてのコンポーネントが "Ready" 信号を出すとすぐに，モニタリング機能が起動します。

HHU 用 MPI カップリング (新規開発用ではありません)

PLC オペレーティングシステムおよびパラメータ表現による HHU 用コミュニケーションは，SDB210 を介しています。

MCP 用コミュニケーションは，下記のとおり，NCK および COMM モジュールを介して PLC BP から制御されています。

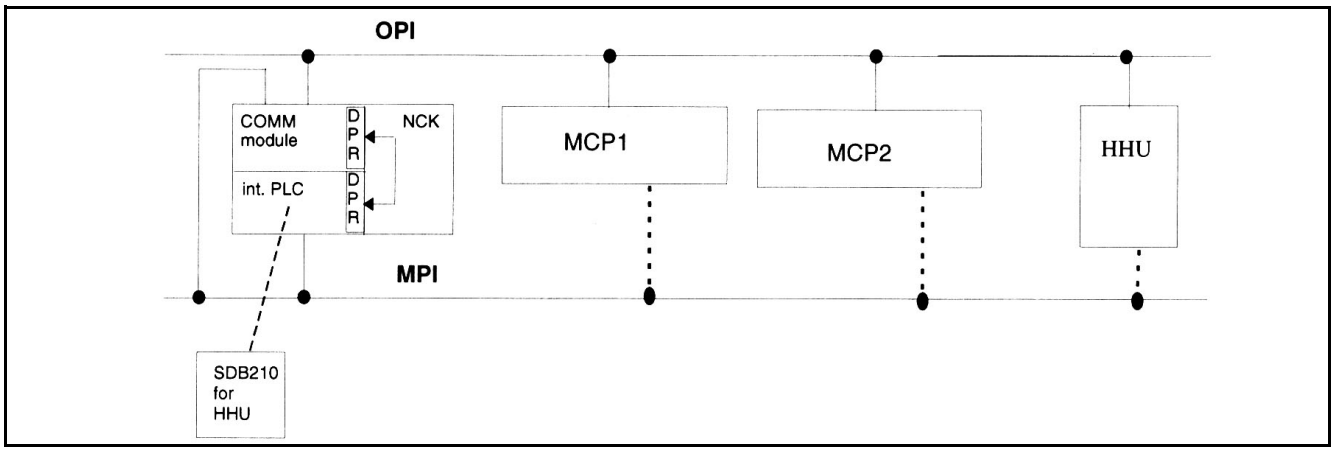


図 1.123 HHU 用 MPI カップリング

適切なパラメータ (FB1):

PLC と HHU の間でのコミュニケーションは，構成に続いて，SDB210 (グローバルデータ) をロードして実行します。基本プログラムを HHU データにアクセスして，HHU 障害モニタリングを実行するには，グローバルデータを介してセットされたアドレスを FB1 パラメータ内で宣言してください。

表 1.39 適切なパラメータ (FB1)

MCP		HHU (SDB210 を介してパラメータで表す)
インタフェースは，FB 1 の適切なパラメータを介してパラメータで表される（表 1.36 参照)		BHG = 1 (SDB210 を介した転送)
		BHGIn (パラメータ SDB210 で表した場合)
		BHGOut (パラメータ SDB210 で表した場合)
		BHGStatRec (パラメータ SDB210 で表した場合)
		BHGTimeout (パラメーター SDB210 で表した場合)

表 1.40 ステータス情報 (MCP1 および MCP2，表 1.40 参照)

適応機種	ビット番号	説明
BHGStatRec	10	受信者 : タイムアウト

タイムアウトモニタがあるので，PLC 診断バッファでエラー入力が生成されると，MMC に以下のエラーメッセージが現れます。

- 400262: HHU 障害

HHU 障害は，データ交換においてすでに発生している場合のみ検知されます。HHU との最初のデータ交換で，モニタリング機能が起動します。

MPI カップリング (SW バージョン 4 以降)

コミュニケーションは，SDB210 を必要としない MPI を介したリンクでも，NCK および COMM モジュールなどを介して，PLC BP からスタートします。オペレーションは，FB1 にある適切なパラメータを介して，パラメータで表されます。

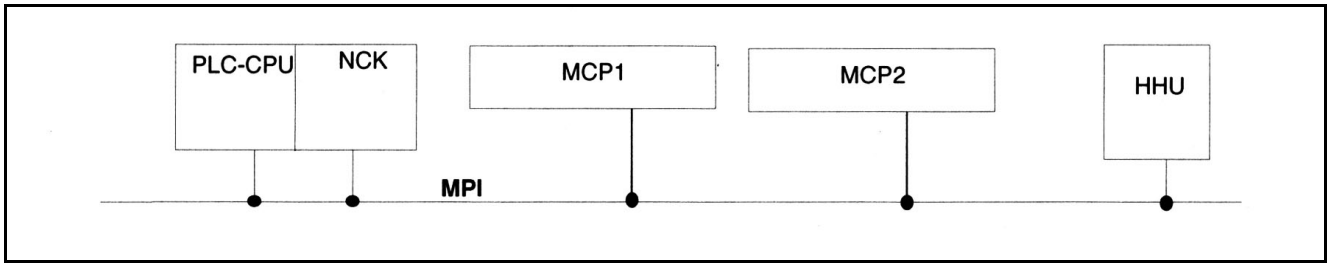


図 1.124 MPI カップリング

表 1.41 適切なパラメータ (FB1)

MCP		HHU
MCPNum=1 または 2 (MCP の数)		BHG=2 (Comm モジュールを介した転送)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatSend	MCP2StatSend	BHGStatSend

表 1.42 適切なパラメータ（FB1)

MCP		HHU
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1BusAdr	MCP2BusAdr	BHGInLen
MCP1Timeout	MCP2Timeout	BHGOutLen
MCP1Cycl	MCP2Cycl	BHGTimeout
MCPMPI = TRUE (MPI)		BHGCycl
MCP1Stop	MCP2Stop	BHGRecGDNo
		BHGRecGBZNo
		BHGRecObjNo
		BHGSendGDNo
		BHGSendGBZNo
		BHGSendObjNo
		BHGMPI = TRUE (MPI)
		BHGStop

表 1.43 ステータス情報

適応機種	ビット番号	説明
MCP1StatSendMCP2Stat Send BHGStatSend	4	GD パケットの構文エラー : パラメータセットのエラー (FB1)
MCP1StatSendMCP2Stat SendB HGStatSend	27	送信者 : タイムアウト
MCP1StatRec MCP2StatRec BHGStatRec	10	受信者 : タイムアウト

タイムアウトモニタ (ビット 10 および 27) があるので，PLC 診断バッファでエラー入力が生成されると，MMC に以下のエラーメッセージが現れます。

- 400260: MCP1 障害または
- 400261: MCP2 障害
- 400262: HHU 障害

MCP/HHU と MCP との間で，まだデータの交換が行われていなくても，コールド再スタート後に，MCP または HHU 障害（SW バージョン 4 以降）は，即時に検知されます。電源投入後，すべてのコンポーネントが "Ready" 信号を出すとすぐに，モニタリング機能が起動します。

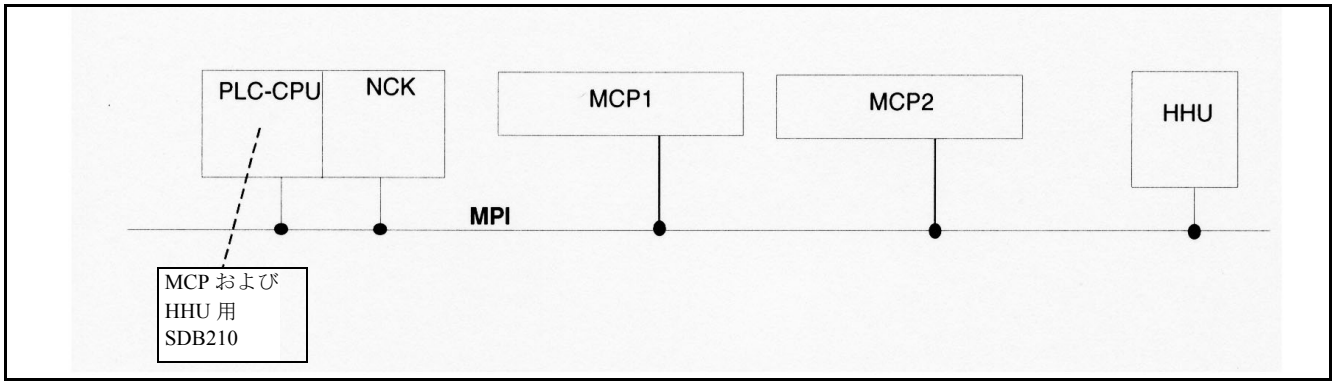


図 1.125 MPI カップリングおよび 810D

適切なパラメータ (FB1):

PLC と HHU の間でのコミュニケーションは，構成に続いて，SDB210 (グローバルデータ) をロードして実行します。基本プログラムを これらのデータにアクセスして，HHU 障害モニタリングを実行するには，グローバルデータを介してセットされたアドレスを FB1 パラメータ内で宣言してください。

表 1.44 適切なパラメータ (FB1),
(グローバルデータ SDB210 における，パラメータで表したすべての入力)

MCP		HHU
MCPNum=1 または 2 (MCP の数)		BHG=1 (MPI)
MCP1In	MCP2In	BHGIn
MCP1Out	MCP2Out	BHGOut
MCP1StatRec	MCP2StatRec	BHGStatRec
MCP1Timeout	MCP2Timeout	BHGTimeout

表 1.45 ステータス情報 (MCP1 および MCP2，@表 2-11 参照)

適応機種	ビット番号	説明
MCP1StatRec MCP2StatRec BHGStatRec	10	受信者 : タイムアウト

加えて，タイムアウトモニタがあるので，PLC 診断バッファでエラー入力が生成されると，MMC に以下のエラーメッセージが現れます。

- 400260: MCP1 障害または
- 400261: MCP2 障害
- 400262: HHU 障害

MCP/HHU 障害は，データ交換において，すでにその装置で発生している場合のみ検知されます。MCP/HHU との最初のデータ交換で，モニタリング機能が起動します。

■ 安全統合 SPL

安全な統合のための機能説明を参照してください。

■ プログラミング言語　C (S7/C7 にある C)

使用

このプログラミング言語の使用は，（PLC プロジェクト管理に関する協議後の）販売部との契約によってのみ許可されるものです

一般事項

S7/C7 用ソフトウェアパッケージ C によって，統合 C インタフェースのある CPU に，C ベースの機能ブロックを作成することができます。この統合 C インタフェースは，統合 PLC 314 またはソフトウェアバージョン 4.1 以降の PLC 315-2DP のある システム で使用できます（ソフトウェアバージョン 2.1 の システム)。S7/C7 用プログラミング言語 C は，STL, CAD, CSF およびその他の高級プログラミング言語とともに使用するミクストランゲージに適しています。

S7/C7 用ソフトウェアパッケージ C のインストールおよび操作方法に関する基本的指示は，取扱い説明書 "C for S7/C7, Programming Blocks" （S7/C7 用 C，プログラミングブロック）を参照してください。取扱い説明書で説明している CPU に比べて，統合 CPU PLC 314 および PLC 315-2DP についてはいくらかの違いがあり，これについては以下で説明しています。

S7/C7 用プログラムパッケージ C は，メニュー項目 C Project, Properties （C プロジェクト，プロパティ）で，Linker（リンカー）および Locator（ロケータ）レジスタに設定してください。

リンカーレジスタのプロパティ

正しい API アドレスを，リンカースイッチフィールドに入力してください。API アドレスは，各々の CPU にある API 機能のジャンプテーブルを指しています。アドレスは，下記のフォームで入力してください。

PLC314

- api_base(00c70000h) (PLC 314 用)

PLC315-2DP

- api_base(00470000h) （PLC 315-2DP 用）

C 機能ブロック用アドレスエリア

固定されたアドレス範囲は，C 機能ブロック用に確保されます。C 機能ブロック用には使われないメモリは，STEP 7 ブロックに有効です。これらは，データブロック，ユーザ構成の機能ブロックおよび，特に基本 PLC プログラムに属するブロックになります。

	PLC 314	PLC 315-2DP
初期アドレス	0001A058H	H00040058
最終アドレス	000C0000H *)	00180000H *)
最大サイズ	664 K バイト	1280 K バイト

*) 最大メモリ構成を適用

C 機能ブロック用ユーザメモリ

ユーザメモリは，STEP 7 ブロックまたは C 機能ブロックに均等に割当てられます。C 機能ブロック用に確保される標準メモリエリアは，292 K バイト です。ただし，146 K バイト までは，マシン構成による基本 PLC プログラムに充てられるようにしてください。ユーザメモリは，146 K バイト までのメモリブロック毎に，C 機能ブロック用に拡張することができます。インストールした CPU により，以下のように最大可能拡張が表わされます。

- 最大 2*146 K バイト （PLC 314 用）
- 最大 7*146 K バイト (PLC 315-2DP 用)

C プログラミング言語用の 146 K バイト のブロックは，32 KB メモリ拡張オプションに相当します。

1.15.3 補足条件 および NC VAR セレクタ

■ 補足条件

■ プログラミングとパラメータ化ツール

ハードウェア

PLC をインストールした YS840DI には，下記の装備のプログラミング装置または PC が必要になります。

	最小	推奨
プロセッサ	80486	ペンティアム
RAM (MB)	8 または 16	32 以上
ハードディスク，容量フリー (MB)	80	120
インタフェース	ケーブルのある MPI，メモリカード	
グラフィック	VGA または TIGA	SVGA
マウス	あり	
オペレーティングシステム	Windows_3.1 MS-DOS_5.0 または STEP7 バージョン 2 からの Windows 95	

S7-300 用 STEP 7 パッケージがプログラミング装置に添付されていない場合，上記の必要条件を満たす装置であれば，パッケージを入手してインストールすることができます。

このパッケージは，以下の機能を実行することができます（実行できる機能についてのさらに詳しい情報は，SIMATIC カタログおよび STEP 7 取扱い説明書を参照してください)。

プログラミング

- STL (SFB-/SFC コールのあるコンプリート言語セット), LAD, CSF 用エディタおよびコンパイラ
- 割当てリストの作成および編集（記号エディタ）
- データブロックエディタ
- ON/OFF ラインのブロックの入出力
- ON/OFF ラインの変更および追加の挿入
- プログラミング装置から PLC へのブロックの転送および, PLC からプログラミング装置へのブロックの転送

パラメータ化

- CPU および I/O 装置をパラメータ化するためのパラメータ化ツール HW Config
- 構成ツールの Communication Configuration （コミュニケーション構成）あるいは, CPU コミュニケーションパラメータの設定
- ハードウェアおよびソフトウェアのバージョン, メモリ容量, I/O 拡張／割当てなどのシステムデータの出力

テストおよび診断 (ONLINE)

- 変数ステータス／強制（I/O, フラグ, データブロック内容など）
- それぞれのブロックのステータス
- システムステータスの表示 (ISTACK, BSTACK, システムステータスリスト)
- システムメッセージの表示
- プログラミング装置からの PLC STOP/ 完全なリセット／すべてのリセットのトリガ
- PLC の圧縮

取扱い説明書

- 個々のブロックまたはすべてのブロックについてのプリントアウト
- シンボリック名の割当て (データブロックにある変数についても)
- それぞれのブロックでのコメントの入出力
- テストおよび診断表示のプリントアウト
- ハードコピー機能
- クロス参照リスト
- プログラム概要
- 割当てプラン I/O/M/T/C/B/P/D

ユーティリティルーチンの達成

- 個々のブロックの出力ステータスの割当て
- 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

-HIGRAPH, GRAPH7, SCL, C (SW バージョン 4.1 以降) におけるプログラミング。
これらのパッケージは，SIMATIC Sales and Marketing Offices（SIMATIC 販売部）にご注文ください。

- モジュール構成用の追加のパッケージ (例えば，CP3425 -> NCM パッケージ)

■ 必要な SIMATIC 取扱い説明書

参照 :

System description of SIMATIC S 7
List of S7-300 operations, CPU 314, CPU 315-2DP
Programming with STEP 7
STEP 7 User's Guide
STEP 7 Programming Guide: Design of user programs
STEP 7 Reference Manual: Statement List (STL)
STEP 7 Reference Manual: Ladder Diagram (LAD)
STEP 7 Reference Manual: System and Standard Functions
STEP 7 Manual: Conversion of STEP 5 Programs
STEP 7 General Index
Manual CPU 314, CPU 315-2DP

■ 関連 SINUMERIK 文書

参照 : /IAF/, Installation and Start-Up Guide FM-NC,
Section "PLC interface",
/IAD/, Installation and Start-Up Guide 840 D, 611 D,
Section "PLC interface",
/IAG/, Installation and Start-Up Guide 810 D, 611 D,
Section "PLC interface",
/BH/, Operator Components Manual (HW)
840 D/ FM-NC/ 810 D
/FB/, Descriptions of Functions FM-NC/ 840 D, 810 D
/LIS/, Lists 840 D/ FM-NC/ 810 D
/FBP/, PLC C Programming

■ NC VAR セレクタ

■ 概要

一般事項

VAR セレクタは，バージョン 1.x のプロジェクト構造に一致するように設計されました。このため，STEP 7 バージョン 2.x 以降のシステムがインストールされているシステムには，ユーザご自身で以下の手順を行ってください。

拡張子 xxxxxxxx.S7D のついているカタログおよびサブカタログのオプション版は，NC-VAR セレクタバージョン 4.01.01 以降のファイルマネージャあるいはエクスプローラにセットしてください。VAR セレクタの選択されたデータ（data.VAR および data.AWL(STL)）は，このカタログに保存してください。"Data.AWL" ファイル用に，STEP 7 マネージャ（バージョン 2 以降）を介して，バージョン 2 のマシンプロジェクトへの "Insert"（挿入）, "External source" （外部ソース）入力を実行してください。このためには，ソースのコンテナはマネージャから選択してください。このファイルは，バージョン 2 (以降) のプロジェクト構造体に，保存および記録されます。ファイルが転送されたら，これらの AWL (STL) ファイルを STEP 7 バージョン 2 (以降) で翻訳してください。

PC アプリケーションの NC VAR セレクタは，必要な NC 変数のアドレスをフェッチして，PLC プログラムにアクセスするため，それらを条件づけします。これによりユーザは，全 NC 変数の範囲から，NC 変数を選択して，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，マシン CPU プログラムに ASCII ファイル（*.AWL）として保存します。図 1.126 にこのプロセスを説明しています。

（注）最新の NC VAR セレクタは，NC ソフトウェアのどのバージョンにも使用することができます（初期のバージョンでも）。初期の 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することができます。DB 120 (変数用デフォルト DB) のデータ内容は，ソフトウェアのバージョンに左右されません。すなわち，古いソフトウェアバージョンにある選択された変数は，ソフトウェアをアップグレードした場合でも，再選択しないでください。

警告

Windows 95 で VAR セレクタをインストールする場合，適切な Windows ディレクトリにセレクタをインストールする前に，現存の "SESETUP.*" および "SIBATCH.*" ファイルを削除してください。これらのファイルは，VAR セレクタによって，ツールボックスから自動的に削除されます (バージョン 3.5 以降)。

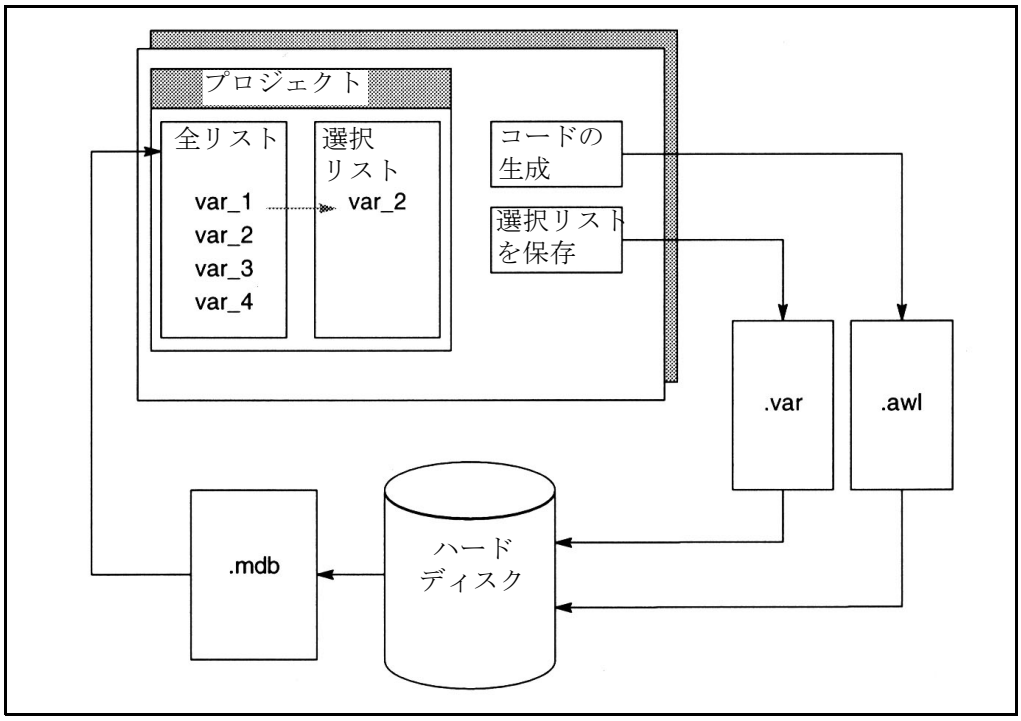


図 1.126 NC VAR セレクタ

NC VAR セレクタアプリケーションをスタートした後，NC 型の変数の全リストが選択され（ハードディスク File. mdb），このリストにあるすべての変数がウィンドウに表示されます。ユーザは，変数を 2 番目のリスト（別のウィンドウ）に転送することができます。後で選択されたほうの変数を，ASCII ファイルで保存するか，または STEP 7 ソースファイル (.awl) として編集して保存することができます。

STEP 7 コンパイラを使用して PLC データブロックを生成すれば，ユーザは，STEP 7 ファイルを使用して，基本プログラム機能ブロック "PUT" および "GET" を介して，NCK 変数を読取ったり，書込んだりすることができます。

選択変数のリストも，ASCII ファイルとして保存することができます (File Extension. Var)。

NC VAR セレクタツールとともに提供されている全リストは，現在の NC ソフトウェアバージョンに相応に合うようになっています。このリストには，ユーザによって定義された変数（GUD 変数）はまったく含まれていません。これらの変数は，機能ブロック FB 5 による基本プログラムで処理します（本説明の 1.15.4「ブロック説明」を参照）。

（注）NC VAR セレクタの最新のバージョンは，いままでの NC ソフトウェアバージョンすべてを処理することができます。したがって，NC VAR セレクタの異なったバージョンを平行してインストールする必要はありません。

システムの特徴と，一般的なコンディション

PC アプリケーション NC VAR セレクタは，Windows 3.1, Windows 3.11 または，Windows 95 の環境において使用することができます。変数の名前の割当てについては，下記の参照を参照してください。

参照： パラメーター一覧

■ 機能の説明

概要

図 1.127 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。

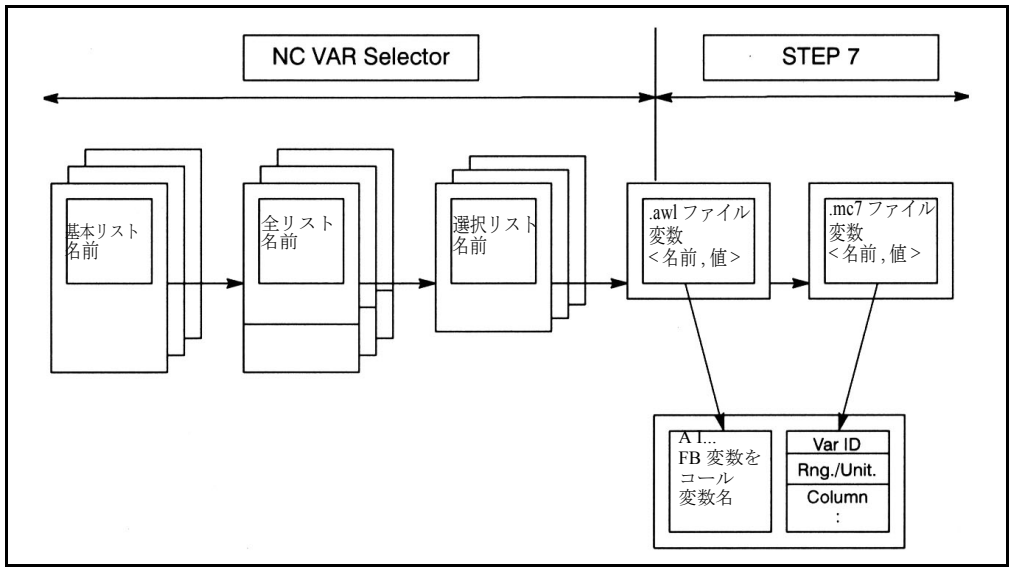


図 1.127 STEP 7 の環境における NC VAR セレクタのアプリケーション

NC VAR セレクタは，選択変数のリストを変数の全リストから生成して，STEP 7 コンパイラで処理することのできる STEP 7 ソースフォーマットの .awl ファイルを生成するのに使われます。

ファイルストレージに関する注記

NC 変数を管理するために，特に頻繁に使う場合，多くのファイルが必要になります。したがって，最初のプロジェクトをセットアップする時には，ふさわしいファイルストレージ構造を選ぶことが大切です。この点に関しては，以下に述べている一般的なコンディションについて注意してください。

- 基本リストは，すべての NC ソフトウェアバージョンに対して，また NC タイプに応じて提供されています。ユーザは，ユーザデータ用に，名前とアドレスのパラメータをこのリストに追加してください。マシンシリーズによってリストが変わるのであれば，それぞれのマシンシリーズに対する全リストを作成することをおすすめします。
- この全リストは，それぞれのマシンシリーズあるいはそれぞれのマシン用に，選択変数のリストを生成する基礎として使うことができます。

- "selected list" （選択リスト）を基礎に，*.awl ファイルをすぐに生成することができ，続いて STEP 7 コンパイラを使用してロードできるデータブロックを生成するのに使うことができます。このため，*.awl ファイルは常に，マシン別 STEP 7 プロジェクトストレージ <sub-catalog name> （サブカタログ名）.s7d に保存されていなければなりません。

  （注）*.awl ファイルには，アドレスパラメータのデータと，NC 変数の名前または ALIAS 名の両方が入っています。データブロックは，アドレスパラメータ（1 パラメータにつき 10 バイト）だけが入っているファイルから生成します。

- 生成したデータブロックは，STEP 7 仕様にしたがって，マシン別ファイルストレージに常に保存しておいてください。
- GET/PUT (FB 2/3) ブロックが，NC アドレスに対してパラメータ化されていることを，記号を使って確認するには，生成されたデータブロックの名前を，STEP 7 記号リストに載せてください。

基本画面／メインメニュー

NC VAR セレクタを選択（スタート）すると，すべての入力オプション（上にあるメニューバー）が表示されている基本画面が現れます（図 1.128)。表示される他のすべてのウィンドウは汎用ウィンドウです。

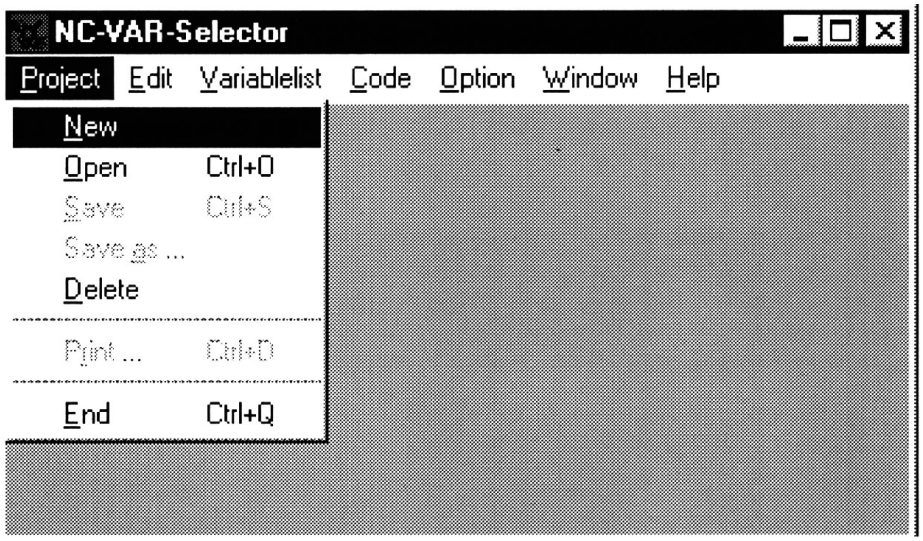


図 1.128 基本メニューの基本画面

メニュー項目プロジェクト

プロジェクトファイル（選択変数のファイル）に関する操作は，すべてこのメニュー項目から行ってください (図 1.128)。

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，メニュー項目 Project にある End を選択してください。

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）を作成するには，メニュー項目 Project で行います。

NEW を選択すると，選択変数用のウィンドウが表示されます（図 1.129）。

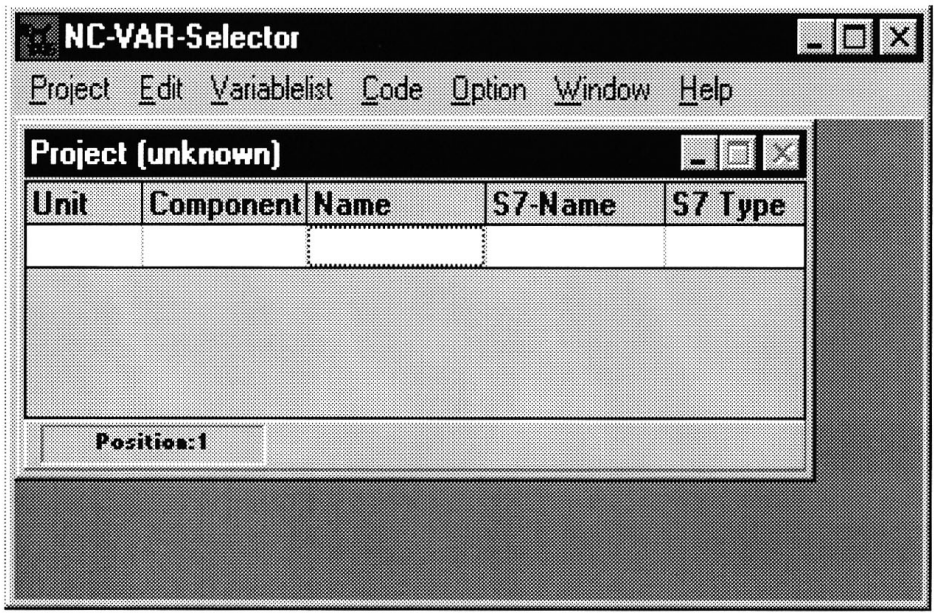


図 1.129 新規プロジェクト用選択変数のウィンドウ

選択変数用のウィンドウの数は，1 つに限られています。それぞれのプロジェクトに対して，最大で 819 の変数をプログラムすることができます。

現在あるプロジェクトを開く

現在あるプロジェクト（変数がすでに選択されている）を開くには，メニュー項目 Project にある Open を選択してください（図 1.129）。ファイルを選択するウィンドウが表示されて，*.var の拡張子 がついた適切なプロジェクトを選択できます。

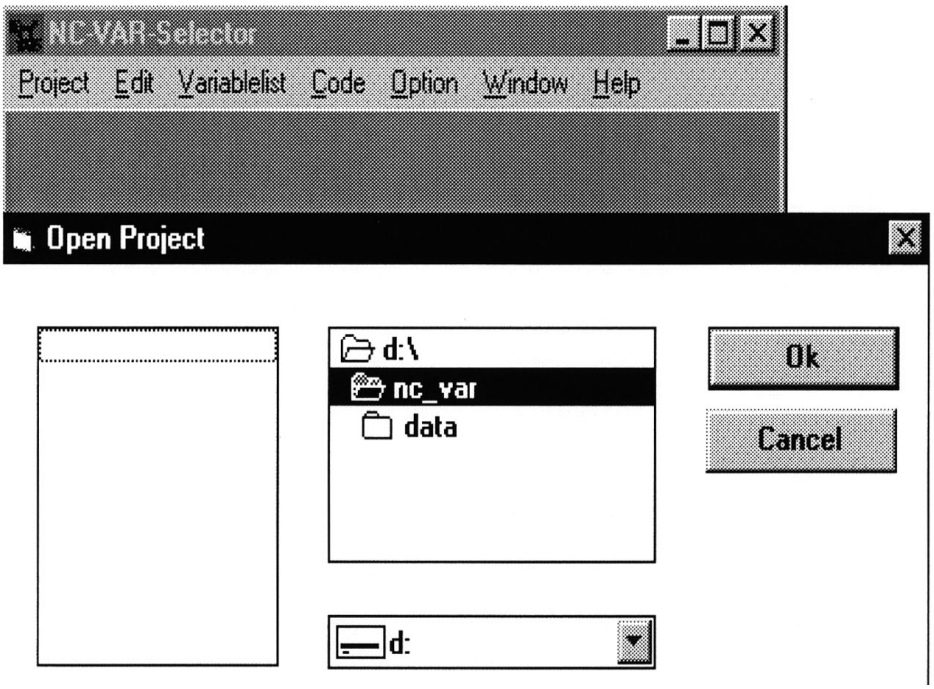


図 1.130 現在あるプロジェクトの選択ウィンドウ

プロジェクトを選択した後で，新しい変数を追加する場合は，NCK 変数の全リストを選択してください（「全リストの選択」を参照してください）。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，Project, Store, Save as... のどのメニューからでも保存することができます（NCVAR バージョン 4.01.01 以前のものは，.S7D パスからのみ保存できる）。

Store を選択すると，すでに指定されているパスで，変数リストを保存します。プロジェクトパスがわからない場合は，Store as... からの方法で保存します。

Store as... 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。

プロジェクトの削除

プロジェクトファイルを削除するには，メニュー項目 Project のコマンド Delete を選択します。

プロジェクトの印刷

プロジェクトを印刷するには，メニュー項目 Project のコマンド Print を選択します。

メニュー項目 Edit

以下のようなすべてのオペレータ動作は，このメニューから直接実行することができます。これらの動作をキャンセルする場合も，Edit で行うことができます。

- 変数の転送
- 変数の削除
- エイリアス名の変更
- 変数の検索

アンドゥ動作

プロジェクトファイルの作成に関するオペレータ動作（変数の転送，変数の削除，エイリアス名の変更）をアンドゥするには，このメニューで行うことができます。

メニュー項目 Complete list

すべての変数をインストールしなければならない基本リストのパスは，ソフトウェアをインストールしている間に要求されます。このリストは，全リストとして選択できます。

全リストの生成

このメニュー項目は，ユーザがご使用になるものではありません。NC ソフトウェアの開発中に，リストを生成するために使われます。

全リストへ変数を追加する

このメニュー項目は，ユーザがご使用になるものではありません。NC ソフトウェアの開発中に，補足リストを生成するために使われます。

全リストの選択

メニュー項目の Complete List, Select で，NC バージョンのすべての NC 変数リストが選択され，表示されます (図 1.131 参照)。

選択された全リストの最後の 2 つは，プルダウンメニューの Complete list に追加されます。ここをマウスでクリックすれば，再び直接選択することができます。

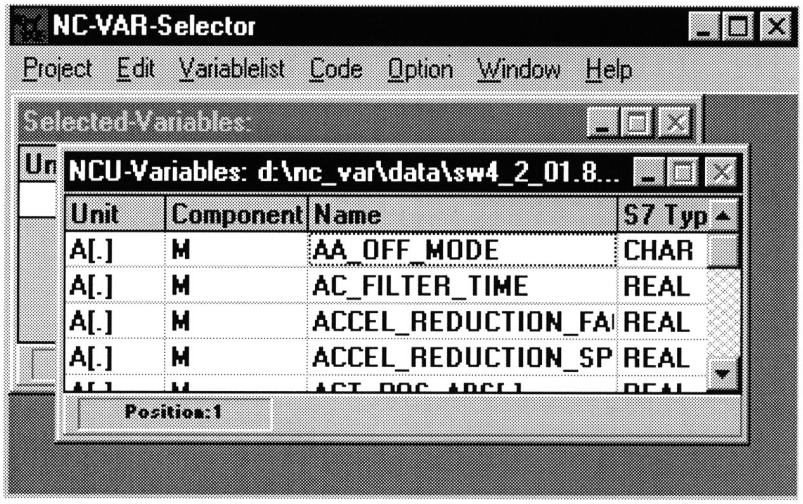


図 1.131 選択された全リストのウィンドウ

フィールド変数（例えば，軸エリア，T エリアデータ，など）は，括弧を使って表示されます ([.])。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へ転送されると，必要な追加情報を要求してきます。

ディスプレイサブセット

表の中の項目をどれでもダブルクリックすると（変数の項目以外），フィルタ基準がプリセットできるウィンドウが表示されます (図 1.132)。

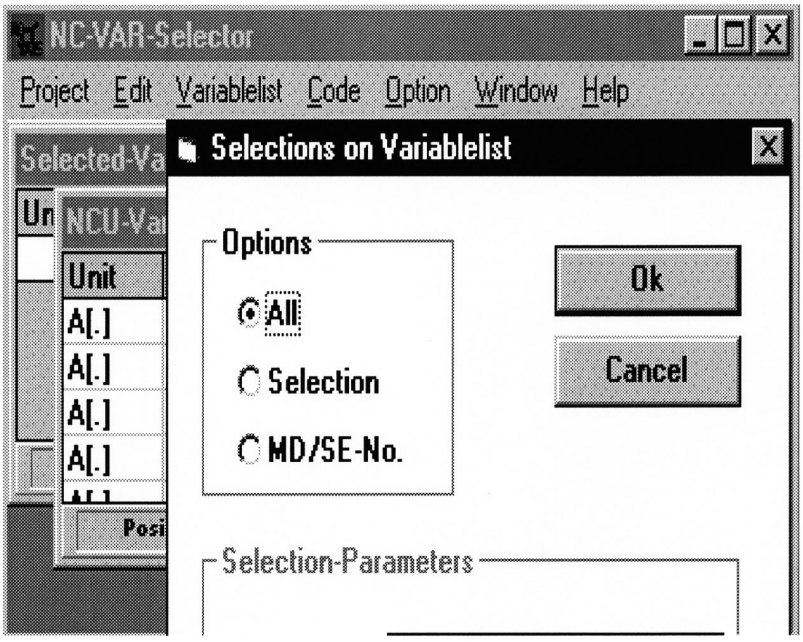


図 1.132 変数表示リストのフィルタ基準のウィンドウ

3 個の選択項目があります。

- すべてのデータを表示
- エリア，ブロック，名前，タイプを特定（組合わせも含む）
- MD/SE データ番号を表示

下記の記号でこのようなこともできます。

* 検索基準を思い通りの範囲に広げる

? 検索文字列の文字を取替える

検索基準の例

検索基準の名前 : CHAN*　　検索結果 :　CHAN_NAME
chanAlarm
chanStatus
channelName
chanAssignment

検索基準の名前 : CHAN?????　検索結果 :　CHAN_NAME
chanAlarm

変数の選択

変数を選択するには，マウスを 1 回クリックしてください。選択された変数は，ドラッグアンドドロップで，選択変数のウィンドウに移動してください。変数をダブルクリックすると，転送動作がすぐに実行されます。下記の要領で，1 度にいくつかの変数を選択することができます。

- 最初の変数をクリックする
- シフトキーを押し下げたままにする
- 最後の変数をクリックする

移動は（アイコンは，ランプオフ，ランプオン），ドラッグアンドドロップで行います（マウスのボタンを押したままで，マウスのポインタを変数のリストまでドラッグします）。

この動作も，メニュー項目 Edit で行うことができます。

エイリアス名

変数名は，32 文字までの長さになっています。生成される変数が，データブロック内で明確に識別されるには，選択した名前にいくつかのアスキー文字を加えます。しかしながら，STEP 7 コンパイラは，明白な STEP 7 変数として，24 個のアスキー文字だけを識別します。変数名は，最後の 8 文字の位置によってのみ区別できるので，長すぎる名前にはエイリアス名を使います。

変数が選択されると，使用する STEP 7 の名前の長さがチェックされます。24 文字よりも長い場合は，エイリアスとして使う追加の名前を入力してください。この場合，エイリアス名が正しいかユーザご自身で確認してください。

入力されたエイリアスは，Option メニューの中で，ユーザによって常にアクティブにすることができます。エイリアス名は，変数が転送される時はいつでも，入力できます。

S7 変数の名前のフィールドをダブルクリックすれば，あとからエイリアス名を編集することができます。メニュー項目 Edit で，これをキャンセルすることもできます。

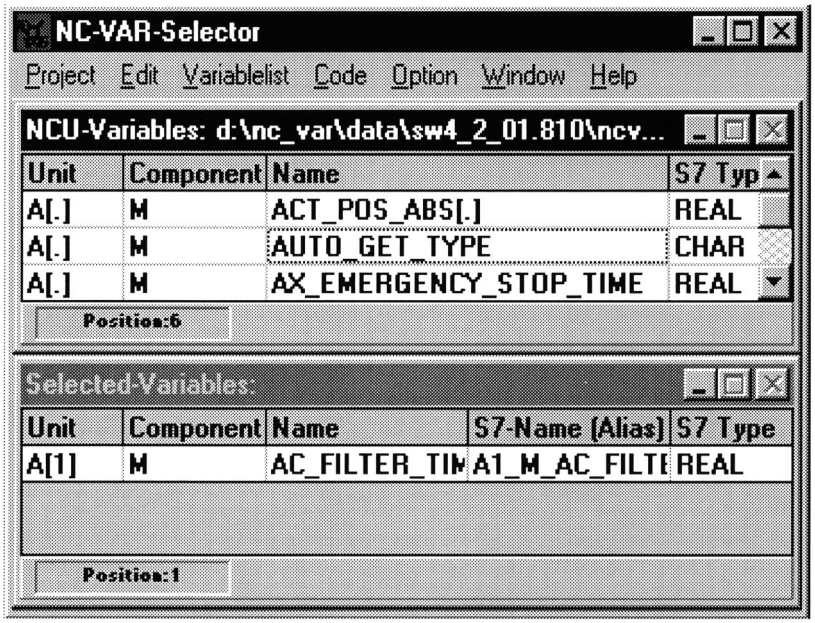


図 1.133 全リストと選択変数の画面

スクロール

すべての変数をウィンドウに表示することができない場合，スクロールバーを表示します。表示されない変数は，画面をスクロールすると現れます（ページを上下する）。

マルチディメンション構造における変数

変数をマルチディメンション構造から選択した場合，エリア番号だけでなく，列および／または行番号も入力して，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は，NC 変数の文書に載っています。（Manufacturer Documentation Lists, Section Variables）。

ブロック番号，または行，または列インデックスにゼロ (0) を入力すると，S7 PLC にある変数をこれらのデータに対して，ポインタとして使うことができます。

"PUT" または "GET" 機能を介して，これらのデータを読取りあるいは書込みをする時，オプションのパラメータである "Unit X", "Column X", "Line X" に必要な情報を入力してください。

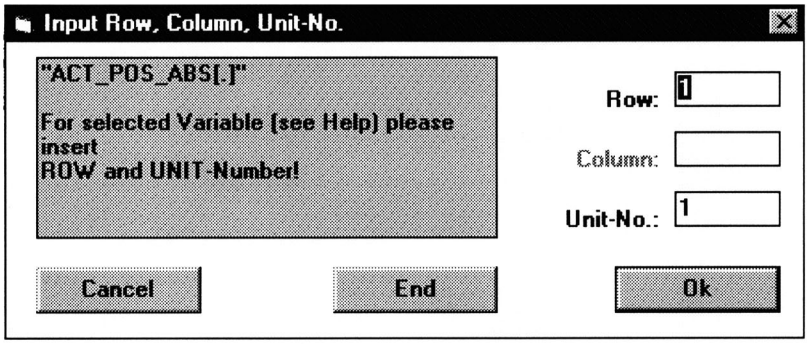


図 1.134 行，列，ブロック番号のエントリフィールド

変数の削除

変数の削除を行うには，選択変数のウィンドウで，削除したい変数を選択して（1回クリックします）Delete キーを押してください。ダブルクリックをすると削除できません。いくつかの変数を同時に削除することもできます（変数の選択」を参照してください)。

削除についても，メニュー項目 Edit でアンドゥすることができます。

（注）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時は，変更する前の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。このように，FB"GET" または FB"PUT" における変数の割当てが，再度コンパイルした後でも正しいままであるかを確認することができます。

選択リストの保存

変数を選択すれば，プロジェクト名をつけて保存することができます。ファイルは S7 プロジェクトベースで保存されます。この場合，変数選択リストおよび生成された STEP 7 ファイルは，S7 プロジェクトパス <program file>.S7D 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（図 1.135)。このウィンドウで，ファイルのプロジェクトパスとファイル名を特定してください。ファイルは S7D パスにのみ保存されることになります。

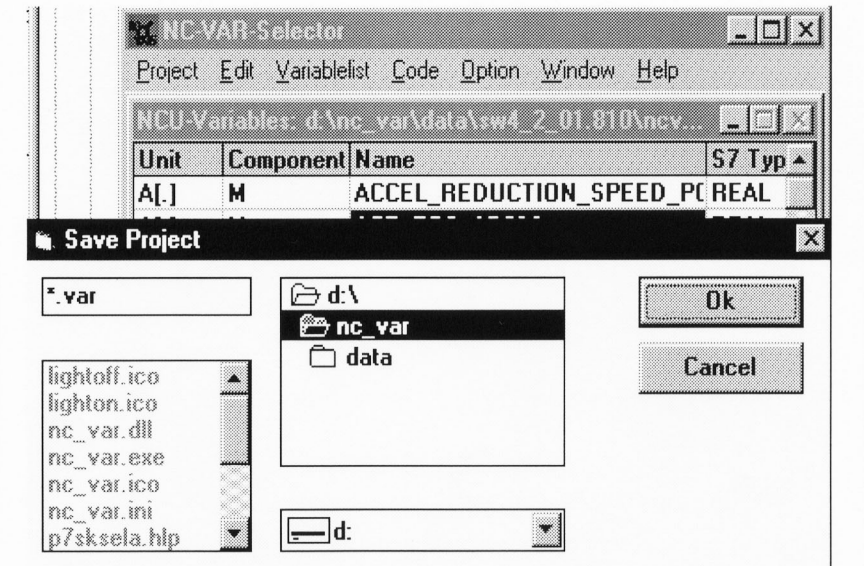


図 1.135 保存するファイルのプロジェクトパスとファイル名のウィンドウ

コード生成

このメニュー項目では 2 つのオプション選択があります。

1. 設定 (コードを生成するためにデータブロック番号を入力する)
2. 生成 (データブロックを作成)

DB 番号の設定

コードを生成するためには，このメニュー項目で DB 番号を入力します。

生成

このメニュー項目で，選択変数リストの STEP 7 ファイルを，拡張子 .awl でセットします。Select をクリックすると，2 つのファイルが生成されます。

1. .awl ファイルは，STEP 7 コンパイラ用の入力として使用します。
2. .var ファイルには，ASCII 文字の選択変数が入っています。

ファイルは S7 プロジェクトベースで保存されます。この場合，変数選択リストおよび生成された STEP 7 ファイルは，S7 プロジェクトパス <program file>.S7D に保存されます。

保存するすべての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，生成された .stl ファイルのプロジェクトパスとファイル名を特定してください。このようなファイルは .S7D パスにのみ保存されることになります。

メニュー項目 Option

メニュー項目 Option で，以下の選択ができます。

- カレントランゲージ
- エイリアス入力モード (常に 24 文字以下)

メニュー項目 Windows

メニュー項目 Windows で，変数のウィンドウの表示のしかたを変えることができます。以下の方式があります。

- ウィンドウを重ねずに並べていく
- 1 つのウィンドウを別のウィンドウの下に重ねていく
- 1 つのウィンドウを別のウィンドウの上に重ねていく

メニュー項目 Help

ここでは，以下のサブメニューの詳細情報を得ることができます。

- The Operator’s Guide （オペレータガイド）
- The Description of Variables （変数の説明）

著作権およびバージョン番号もここに表示されます。

■ スタートアップ，インストール

Windows のアプリケーションである NC Var selector は，パッケージソフトに添付されている SETUP プログラムを使用して，インストールします。

全リストを作成するために，読取りをする入力ファイルが保存される NC_VAR_INI ファイルにあるデフォルトパスをセットすることができます。

1.15.4 ブロック説明

■ FB 1: RUN_UP 基本プログラム，起動セクション

機能説明

NCK と PLC は，同期して立ち上がります。マシンデータで定義された NC 構成およびもっとも重要なパラメータの妥当性を確認するため，NC/PLC ユーザインタフェース用のデータブロックを作成します。エラーが発生した場合には，FB 1 はエラー識別子を診断バッファに送り，PLC を STOP 状態に切替えます。

コントローラを順序正しくスタートアップするために，NCK と PLC が同期していることは，きわめて重要なことです。なぜならば，この 2 つのシステムはそれぞれ独自のタイプの電源投入方法があるからです。したがって，起動ルーチンにおいて CPU は「補助的なスタートアップ機能」を実行して，その方法が正しく機能していることを確実にするために，ID 情報を交換します。スタートアップの方法は非同期なので，1 つの CPU が，ある一定の状況において，別の CPU が立ち上がってくるまで当然「待機」しなければなりません。

PLC 314 と PLC 315-2DP は，「コールド再スタート」のみに有効で，「ウォーム再スタート」は実行できません。すなわち，システムを初期化した後，オペレーティングシステムは，構成ブロック OB 100 を経由して実行し，OB 1 の初めで常に周期運転を開始します。

ユーザは，ご自身のアプリケーションに関連する FB 1 パラメータを与えるだけです。関連インスタンス DB 7 にあるあらかじめ設定されている値を割当てる必要はありません。ブロックを，OB 100 に呼出すだけです。

出力パラメータ

PLC のユーザは，FB 1 の出力パラメータによって，制御システム構成の情報を得ることができます。これらのデータは，周期プログラムセクションにアクセスすることができます。このアクセス方法には，以下の 2 つがあります。

1. シンボリック形式で，データブロック DB 7 (FB 1 のインスタンス) に直接アクセスする方法（例えば，Lgp_par.MaxChan では，この場合の gp_par が DB 7 のシンボリック名です）。
2. FB 1 をパラメータ化する時，フラグ，データ要素を，適切なパラメータに割当てます（例えば，MaxChan:=MW 20)。チャンネルの最大数についての情報は，ユーザプログラムの残り部分にある，marker word 20 でスキャンできます。

（注）STEP 7 アプリケーション Communication Configuration （コミュニケーション構成）を介して，MPI に接続されているオペレータコンポーネントに，追加の SDB 210 を生成してください。対応する手順については，Installation and Start-up Guide（インストールおよびスタートアップガイド）の説明をお読みください。

注意

下記事項は，ポインタパラメータ MCP1In, MCP1Out, MCP1StatSend, MCP1StatRec (または，第 2 の制御パネルおよびハンドヘルドユニットのパラメータ) が，これらのポインタの初期設定を入出力するデータブロックに割当てられている時にだけ，当てはまります。

下記事項は，ポインタパラメータ MCP1In, MCP1Out, MCP1StatSend, MCP1StatRec (および第 2 の制御パネルおよびハンドヘルドユニットのパラメータ) が，データブロックエレメントに割当てられている時にだけ，当てはまります。

1. このオペレータコンポーネントの In パラメータ（MCP1In など）を，データブロックにパラメータ化してください。
2. パラメータ化された DB 番号は，他のポインタがデータブロックにパラメータ化されている場合，そのパラメータと同じでなければなりません（入出力あるいはフラグを示す他のパラメータ）。
3. オペレータコンポーネントのすべてのポインタを，データブロックにパラメータ化する必要はありません。

YS840DI における宣言

```
FUNCTION_BLOCK FB 1
VAR_INPUT
    MCPNum:         INT:=1;          //0: MCP なし
                                     //1: 1 MCP ( デフォルト )
                                     //2: 2 MCP
    MCP1In:         POINTER;         // MCP 1 アドレス指定スタート　信号入力
    MCP1Out:        POINTER;         // MCP 1 アドレス指定スタート　信号出力
    MCP1StatSend:   POINTER;         // MCP 1 転送の DW ステータス
    MCP1StatRec:    POINTER;         // MCP 1 転送の DW ステータス
    MCP1BusAdr:     INT:=6;          // デフォルト
    MCP1Timeout:    S5TIME:= S5T#700MS;
    MCP1Cycl:       S5TIME:= S5T#200MS;
    MCP2In:         POINTER;         // MCP 2 アドレス指定スタート　信号入力
    MCP2Out:        POINTER;         // MCP 2 アドレス指定スタート　信号出力
    MCP2StatSend:   POINTER;         // MCP 2 転送の DW ステータス
    MCP2StatRec:    POINTER;         // MCP 2 転送の DW ステータス
    MCP2BusAdr:     INT;
    MCP2Timeout:    S5TIME:=S5T#700MS;
    MCP2Cycl:       S5TIME:= S5T#200MS;
    MCPMPI:         BOOL:= FALSE;
    MCP1Stop:       BOOL:= FALSE;
    MCP2StopI:      BOOL:= FALSE;
    MCP1NotSend:    BOOL:= FALSE;
    MCP2NotSend     BOOL:= FALSE;
```

```
HHU:              INT;              // 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転送データ
BHGOut:           POINTER;          // ハンドヘルドユニットの受信データ
BHGStatSend:      POINTER;          // 転送 HHU の DW ステータス
BHGStatRec:       POINTER;          // 受信 HHU の DW ステータス
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
MMCToIF:          BOOL:=TRUE;
HWheelMMC:        BOOL:=TRUE;       //MMC を介してのハンドル選択
MsgUser:          INT:=10;          //DB2 のユーザエリア数
UserIR:           bool:=  FALSE;    //OB40 のユーザプログラム
                                    // 注意，ローカルデータ拡張
IRAuxfuT:         bool:= FALSE;     //OB40 の T 機能評価
IRAuxfuH:         bool:= FALSE;     //OB40 の H 機能評価
IRAuxfuE:         ool:= FALSE;      //OB40 の DL 機能評価
UserVersion:      Pointer;          // バージョンディスプレイに出力される変
                                    数文字列へのポインタ
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;
END_VAR
```

正式パラメータの説明

以下は，RUN_UP 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
MCPNum	I	Int	0 から 2	アクティブ MCP の数 0: MCP はインストールされていない
MCP1In MCP2In	I	ポインタ	I0.0 から I120.0，F0.0 から F248.0，または DBn DBX0.0 から DBXm.0	マシン制御パネルの入力信号用スタートアドレス
MCP1Out MCP2Out	I	ポインタ	Q0.0 から Q120.0，F0.0 から F248.0，または DBn DBX0.0 から DBXm.0	マシン制御パネルの出力信号用スタートアドレス
MCP1StatSend MCP2StatSend	I	ポインタ	Q0.0 から Q124.0，F0.0 から F252.0，または DBn DBX0.0 から DBXm.0	マシン制御パネルへの転送のダブルワードステータス用スタートアドレス DW#16#08000000: タイムアウト，それ以外なら 0
MCP1StatRec MCP2StatRec	I	ポインタ	Q0.0 から Q124.0，F0.0 から F252.0，または DBn DBX0.0 から DBXm.0	マシン制御パネルによるデータ受信のダブルワードステータス用スタートアドレス DW#16#00040000: タイムアウト，それ以外なら 0
MCP1BusAdr MCP2BusAdr	I	Int	1...15	マシン制御パネルのバスアドレス
MCP1Timeout MCP2Timeout	I	S5time	700 ms を推奨	マシン制御パネル用周期的ハートビートモニタリング
MCP1Cycl MCP2Cycl	I	S5time	200 ms を推奨	マシン制御パネルへの信号の周期的アップデートにかかる時間
MCPMPI	I	Bool		1: すべてのマシン制御パネルは MPI バスに接続（GD パラメータ化を除く），SW 4.x 以降
MCP1Stop MCP2Stop	I	Bool		0: マシン制御パネル信号の転送スタート 1: マシン制御パネル信号の転送ストップ
MCP1NotSend MCP2NotSen	I	Bool		0: オペレーション起動送受信 1: マシン制御パネル信号のみを受信（近日入荷），(SW バージョン 4 以降)
HHU	I	Int		HHU インタフェース 0: HHU なし 1: HHU から SDB 210 構成の MPI (SW 3.x 用) 2: FB1 パラメータ BHGMPI がすでに TRUE にセットされている場合，HHU から OPI，または MPI（SW 4.x 以降）。
BHGIn	I	ポインタ	I0.0 から I124.0，F0.0 から F252.0，または DBn DBX0.0 から DBXm.0	スタートアドレス HHU から PLC のデータを受信
BHGOut	I	ポインタ	Q0.0 から Q124.0，F0.0 から F252.0，DBn DBX0.0 から DBXm.0	スタートアドレス HHU から PLC のデータを転送
BHGStatSend	I	ポインタ	Q0.0 から Q124.0，M0.0 から M252.0，DBn DBX0.0 から DBXm.0	HHU への転送データのダブルワードステータス用スタートアドレス DW#16#08000000: タイムアウト，それ以外なら 0

信号	種類	タイプ	数値の範囲	メモ
BHGStatRec	I	ポインタ	Q0.0 から Q124.0，M0.0 から M252.0，または DBn DBX0.0 から DBXm.0	HHU からの受信データのダブルワードステータス用スタートアドレス DW#16#00040000: タイムアウト，それ以外なら
BHGInLen	I	バイト	HHU デフォルト : B#16#68 (68 バイト)	ハンドヘルドユニットからの受信データ量
BHGOutLen	I	バイト	HHU デフォルト : B#16#14 (20 バイト)	ハンドヘルドユニットへの転送データ量
BHGTimeout	I	S5time	700 ms を推奨	ハンドヘルドユニット用周期的ハートビートモニタリング
BHGCycl	I	S5time	100 ms を推奨	ハンドヘルドユニットへの信号の周期的アップデートにかかる時間
BHGRecGDNo	I	Int	HHU デフォルト : 2	GD 円番号受信
BHGRecGBZNo	I	Int	HHU デフォルト : 2	GI 番号受信
BHGRecObjNo	I	Int	HHU デフォルト : 1	受信 GI のオブジェクト番号
BHGSendGDNo	I	Int	HHU デフォルト : 2	GD 円番号転送
BHGSendGBZNo	I	Int	HHU デフォルト : 1	GI 番号転送
BHGSendObjNo	I	Int	HHU デフォルト : 1	転送 GI のオブジェクト番号
BHGMPI	I	Bool		1: ハンドヘルドユニットを MPI にカップリング（SDB 210 config. 以外）。パラメータ HHU は 2 にセット。
BHGStop	I	Bool		0: ハンドヘルドユニット信号の転送スタート 1: ハンドヘルドユニット信号の転送ストップ
BHGNotSend	I	Bool		0: オペレーション起動送受信 1: ハンドヘルドユニット信号のみを受信（近日入荷），(SW バージョン 4 以降)
NCCyclTimeout	I	S5time	200 ms を推奨	周期的ハートビートモニタリング NCK
NCRunupTimeout	I	S5time	50 s を推奨	パワーアップモニタリング NCK
ListMDecGrp	I	Int	0...16	拡張 M グループデコーディングのアクティブ状態 0 = アクティブではない 1...16: M groups の数
NCKomm	I	Bool		PLC NC コミュニケーションサービス (FB2/3/4/5/7:Put/Get/PI_SERV/GETGUD) true: アクティブ
MMCToIF	I	Bool		インタフェースへの MMC 信号の転送 (運転モード，ハンドル選択，プログラム調整など)。 True: アクティブ
HWheelMMC	I	Bool		True: MMC を介してのハンドル選択 False: ユーザプログラムを介してのハンドル選択
MsgUser	I	Int	0...25	メッセージ用ユーザエリア量 (DB2)

信号	種類	タイプ	数値の範囲	メモ
UserIR	I	Bool		ユーザ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	デフォルト，false	OB40 の T 機能の評価
IRAuxfuH	I	Bool	デフォルト , false	OB40 の H 機能の評価
IRAuxfuE	I	Bool	デフォルト , false	OB40 の DL 機能の評価
UserVersion	I	ポインタ		文字列変数へのポインタ。関連文字列変数オーバジョンディスプレイに出力。（最大 41 文字），（SW バージョン 4 以降）
MaxBAG	Q	INT	1..10	モードグループの数 (SW 4 以降)
MaxChan	Q	INT	1..10	チャンネルの数 (SW 4 以降)
MaxAxis	Q	INT	1..31	軸の数 (SW 4 以降)
ActivChan	Q	ARRAY[1..10] OF BOOL		アクティブなチャンネルのビット文字列 (SW 4 以降)
ActivAxis	Q	ARRAY[1..31] OF BOOL		アクティブな軸のビット文字列 (SW 4 以降)
UDInt	Q	Int		DB20 の整数マシンデータの数 (SW 4 以降)
UDHex	Q	Int		DB20 の 16 進数マシンデータの数 (SW 4 以降)
UDReal	Q	Int		DB20 の実数マシンデータの数 (SW 4 以降)

モニタリング MCP/HHU

下記の，マシン制御パネルとのコミュニケーションに関するステータス情報は，エラーイベントに出力されます。

適用機種	ビット番号	説明
MCP1StatRec MCP2StatRec BHGStatRec	10	受信者：タイムアウト
SendMCP2StatSendBHGStatSend	27	送信者：タイムアウト

加えて，PLC　の診断バッファにエラーエントリを生成して，MMC 上に以下のエラーメッセージを表示します。

- 400260 : MCP 1 または
- 400261 : MCP 2
- 400262 : HHU

この場合，MCP あるいはハンドルヘルドユニットからの入力信号（MCP1In/MCP2In，または BHGIn）は，0 で初期化されます。PLC と MCP/HHU が再同期できれば，コミュニケーションは自動的に再開して，BP によってエラーメッセージはリセットされます。

■ FB 2 GET, NC 変数の読取り

機能説明

ユーザ PLC プログラムでは，FB GET を使用して NCK から変数を読取ることができます。

FB 2 は，ユーザエリアからの「インスタンス DB」も含みます。（SW 3.7 以降では，マルチインスタンス機能）。

FB 2 が，制御入力 Req において正の信号立ち上がりで呼出されると，ADDR1-ADDR8 で参照されている NCK 変数を読取るジョブがスタートします。つぎに，RD1 から RD8 で参照されている PLC オペランドエリアへ，その変数をコピーします。

読取りプロセスが無事に完了すると，ステータスパラメータ NDR に論理的「1」が示されます。

読取りプロセスは，いくつかの PLC 周期の間続きます（通常は，1 から 2）。ブロックは，周期的モードの時だけ呼出すことができます。

すべてのエラーは，Error（エラー）および State（ステータス）で示されます。

NCK 変数を参照するために，すべての必要な変数を，"NC Var Selector" （NC Var セレクタ）ツールで最初に選択して（本書の セクション 3.2 を参照してください），これを STL ソースとしてデータブロックに生成します。信号リストにあるこの DB に名前を割当ててください。

FB 2 を呼出す時，NCK 変数アドレス（Addr1 to Addr8）の実パラメータとして "DB name.S7 name" を転送します。

変数アドレス指定

多くの NC 変数用に，NC VAR セレクタ内のエリア番号および／または行あるいは列番号を選択する必要があります。基本的なタイプでは，これらの変数を選択することができます。すなわち，エリア／列／行は，0 にあらかじめ設定されています。

NC VAR セレクタで指定されているエリア番号，行および列の内容は，FB において「0」でチェックします。「0」があれば，値は入力パラメータへ転送されます。"GET" FB を呼出す前に，ユーザは望むパラメータを指定してください (UnitX/ ColumnX/LineX)。

!

重要

PLC と NC の間でのコミュニケーション（NC 変数の読取り／書込み，FB 2，3，5，または 汎用 PI サービス，FB 4）が，電源オフあるいは，緊急停止／確認応答／リセットによって中止になった後，コールド再スタートまたはリセット (信号 : Req = 0) の後で最初の OB 1 内で，スタートジョブを削除してください。

FB 2 は，基本プログラムパラメータ NCKomm ="1" がセットされている時だけ NC 変数を読取ることができます (OB 100 では FB1, DB7)。

チャンネル別変数を読取りしているとき，1 つのチャンネルの変数だけが，Addr1 to Addr8 を介してジョブ（FB 2 呼出し）にアドレス指定します。

エリア V および H では，異なる論理軸番号は 1 つのジョブの中に割当てないでください。（割当ててしまうと，エラーになります。Error:=TRUE, State:=W#16#02)。

1 つのグループの中にある NCK 変数をジョブの中で組合わせることができます。

	エリア				
グループ 1	C[1]	N	B	Q	T
グループ 2	C[2]	N	B	Q	T
グループ 3	V[.]	H[.]			

上記の表で，例としてグループ 1 およびグループ 2 について説明しているように，同じ規則が，チャンネル 3 から 10 までに当てはまります。

宣言

```
FUNCTION_BLOCK FB 2
VAR_INPUT
    Req :       BOOL ;
    NumVar :    INT;
    Addr1 :     ANY ;
    Unit 1 :    BYTE ;
    Column1 :   WORD ;
    Line1 :     WORD ;
    Addr2 :     ANY ;
    Unit 2 :    BYTE ;
    Column2 :   WORD ;
    Line2 :     WORD ;
    Addr3 :     ANY ;
    Unit 3 :    BYTE ;
    Column3 :   WORD ;
    Line3 :     WORD ;
    Addr4 :     ANY ;
    Unit 4 :    BYTE ;
    Column4 :   WORD ;
    Line4 :     WORD ;
    Addr5 :     ANY ;
    Unit 5 :    BYTE ;
    Column5 :   WORD ;
    Line5 :     WORD ;
    Addr6 :     ANY ;
    Unit 6 :    BYTE ;
    Column6 :   WORD ;
    Line6 :     WORD ;
    Addr7 :     ANY ;
    Unit 7 :    BYTE ;
    Column7 :   WORD ;
    Line7 :     WORD ;
    Addr8 :     ANY ;
    Unit 8 :    BYTE ;
    Column8 :   WORD ;
    Line8 :     WORD ;
    FMNCNo :    int;1)
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
```

1) FM-NC のみ

正式パラメータの説明

以下は，GET 機能のすべての正式パラメータの表です。

信号	種類	タイプ	数値の範囲	メモ
Req	I	Bool		正の信号エッジでジョブがスタート
NumVar	I	Int	1 から 8 (Addr1 to Addr8 の使用に対応)	読取る変数の数
Addr1 to Addr8	I	Any	[DBName].[VarName]	NC Var セレクタからの変数識別子
Unit 1to Unit 8	I	Byte		エリアアドレス，変数アドレス指定用オプション
Column1 to Column8	I	Word		列アドレス，変数アドレス指定用オプション
Line1 to Line8	I	Word		行アドレス，変数アドレス指定用オプション
Error	Q	Bool		ジョブの確認応答は否定，またはジョブの実行は不可能
NDR	Q	Bool		ジョブは問題なく実行。データは使用可
State	Q	Word		エラー識別子を見ること
RD1 to RD8	I/Q	Any	P#Mm.n BYTE x... P#DBnr.dbxm.n BYTE x	読取りデータのターゲットエリア

エラー識別子

ジョブの実行が失敗すると，ステータスパラメータ Error の "logical 1" で示されます。エラーの原因は，"State" を出力するブロックでコード化されます。

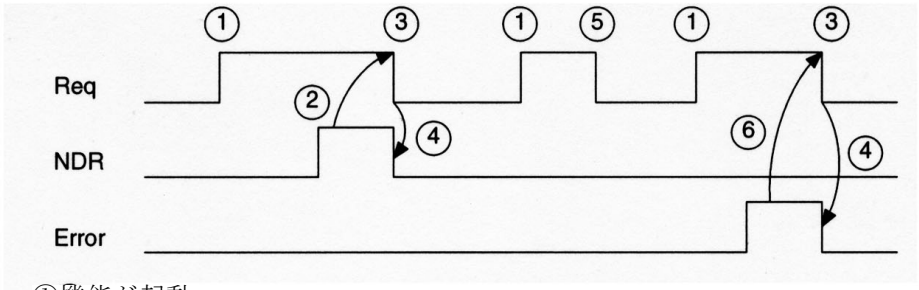
ステータス		意味	注記
WORT-H	WORT-L		
1 から 8	1	アクセスエラー	エラーが発生した変数の，上位バイト番号
0	2	ジョブにおけるエラー	ジョブにおける変数のまちがったコンパイル
0	3	確認応答は否定，ジョブは実行できない	内部エラー，トライ : NC リセット
1 から 8	4	有効なローカルユーザメモリが不十分	読取り変数は，RD 1 で指定するよりも長くなる（...RD 8 ）。エラーが発生した変数の上位バイト番号
0	5	フォーマット変換エラー	ダブルタイプ変数の変換エラー：S7 REAL エリア内に変数はない
0	6	FIFO フル	キューが長すぎるのでジョブを繰返すこと
0	7	オプションがセットされていない	BP パラメータ "NCKomm" はセットされていない
1 から 8	8	不適切なターゲットエリア (RD)	RD1 から RD8 はローカルエリアではない
0	9	転送が占有する	ジョブを繰返すこと
1 から 8	10	変数アドレス指定のエラー	単位または値 0 を含む列／行
0	11	無効な変数アドレス	アドレス（または変数名），エリア，単位をチェック
0	12	NumVar = 0	パラメータ NumVar をチェック

構成処置

NC 変数を読取る前に，以下の構成処置を行ってください。

- NC-VAR セレクタによる変数選択
- 所望のプロジェクトカタログの *.VAR ファイルに選択された変数を保存 (*.S7D)
- STEP 7 の *.STL ソースファイルを生成
- 関連アドレスデータで DB を生成
- 生成された DB 用に，シンボルテーブルへ記号を入力，これによりユーザプログラム内のアドレスパラメータへのシンボリックアクセスが可能
- FB2 パラメータをセット

タイミングダイアグラム



①機能が起動
②確認応答は肯定 : 新しいデータを受信
③確認応答受信後，機能起動をリセット
④ FB を使用して信号変換
⑤確認応答受信以前に機能起動をリセットした場合，出力信号はアップデートされず，影響を受けているアクティブなファンクションの運転シーケンスのみアップデートされる。
⑥確認応答は否定 : エラーが発生，エラーコードは出力パラメータステータスへ

呼出しの例

アドレスデータが DB120 に保存されているチャンネル 1 から，チャンネル別マシンデータ 3 個を読取る。

NC VAR セレクタでデータを選択して，DB120.VAR ファイルに保存；DB120.AWL ファイルを生成

エリア	ブロック	名前	タイプ	番号	バイト	S7 名
C[1]	M	MD 20070: AXCONF_MACHAX_USED[1]	char	20070	1	C1AxConfMachAxUsed1
C[1]	M	MD 20070: AXCONF_MACHAX_USED[2]	char	20070	1	C1AxConfMachAxUsed2
C[1]	M	MD 20090: SPIND_DEF_MASTER_SPIND	char	20090	1	C1SpindDefMasterSpind

以下の事をするために S7 (エイリアス) 名を選択します。

A. チャンネル指定を名前に組込むため。

B. [ ] は，STEP 7 の記号として認められていないので，これを取り除くため。

S7 記号表に名前を入力します。(例えば，DB120 に NCVAR):

記号	オペランド	データタイプ
NCVAR	DB120	DB120

DB120.STL ファイルをコンパイルして PLC へ転送してください。

インスタンス DB 110 での FB 2 のパラメータ化

```
DATA_BLOCK DB 110                    //FB2 用インスタンスとして割当てられて
                                     いないユーザ DB
FB 2
BEGIN
END_DATA_BLOCK
Function FC "Variable Call" : VOID
   I 7.7;                            //MCP 上に割当てられていないキー
   S     F 100.0;                    //Req 起動
   A     F 100.1;                    // メッセージ NDR 終了
   R     F 100.0;                    // ジョブエンド
   A     I 7.6;                      // マニュアルでエラー確認応答
   A     F 102.0;                    // エラーが存在
   R     F 100.0;                    // ジョブエンド
   Call fb 2, db 110(
                Req :=               F 100.0,
                NumVar :=            3, //3 つの変数を読取る
                Addr1 :=             NCVAR.C1AxConfMachAxUsed1,
                Addr2 :=             NCVAR.C1AxConfMachAxUsed2,
                Addr3 :=             NCVAR.C1SpindDefMasterSpind,
                Error :=             F102.0,
                NDR :=               F100.1,
                State :=             FW104,
                RD1 :=               P#DB99.DBX0.0 BYTE 1,
                RD2 :=               P#DB99.DBX1.0 BYTE 1,
                RD3 :=               P#M110.0 INT 1);
```

変数のアドレス指定の例

アドレスデータが基本タイプとして DB 120 に保存されているチャンネル 1 から，R パラメータ 2 個を読取る。R パラメータ番号は，パラメータ行 X を介してパラメータ化される。

```
DATA_BLOCK DB 120
VERSION : 0.0
STRUCT
    C1_RP_rpa0_0:
    STRUCT
    SYNTAX_ID :         BYTE := B#16#82;
    area_&_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 FB 2 , DB110 (
                Req :               F 0.0,
                NumVar :            2,
                Addr1 :             "NCVAR".C1_RP_rpa0_0,
                Line1 :             W#16#1,
                Addr2 :             "NCVAR".C1_RP_rpa0_0,
                Line2 :             W#16#2,
                FMNCNo :            1,          //FM-NC のみ
                Error :             F 1.0,
                NDR :               F 1.1,
                State :             FW 2,
                RD1 :               P#F 4.0 REAL 1,
                RD2 :               P#F 24.0 REAL 1);
```

データタイプ

NC VAR セレクタ内では，NCK データタイプは，変数でリスト出力します。以下は，S7 データタイプへの割当ての表です。

表 1.46 データタイプの分類

NCK データタイプ	S7 データタイプ
double	REAL
float	REAL
long	DINT
integer	DINT
uint_32	DWORD
int_16	INT
uint_16	WORD
unsigned	WORD
char	CHAR or BYTE
string	STRING
bool	BOOL

■ FB 3 PUT, NC 変数の書込み

機能説明

ユーザ PLC プログラムでは，FB PUT を使用して NCK エリアに変数を書込むことができます。

すべての FB 3 呼出しを，ユーザエリアから個々のインスタンス DB へ割当ててください。(マルチインスタンス機能は，SW 3.7 以降)。

FB 3 が，制御入力 Req において正の信号立ち上がりで呼出されると，SD 1 to SD 8 でローカルに参照されている PLC オペランドエリアのデータとともに，Addr1 to Addr8 で参照されている NC 変数の上書きをするジョブがスタートします。書込みプロセスが無事に完了すると，ステータスパラメータ Done に論理的「1」が示されます。

書込みプロセスは，いくつかの PLC 周期の間続きます（通常は，1 から 2)。ブロックは，周期的モードの時だけ呼出すことができます。

すべてのエラーは，Error(エラー) および State(ステータス) で示されます。NCK 変数を参照するために，すべての必要な変数を，"NC Var Selector" （NC Var セレクタ）ツールで最初に選択して，これを STL ソースとしてデータブロックに生成します。信号リストにあるこの DB に名前を割当ててください。FB 3 を呼出す時，NCK 変数アドレス（Addr1 to Addr8）の実パラメータとして "DB name.S7 name" を転送します。

変数アドレス指定

多くの NC 変数用に，NC VAR セレクタ内のエリア番号および／または行あるいは列番号を選択する必要があります。基本的なタイプでは，これらの変数を選択することができます。すなわち，エリア／列／行は，0 にあらかじめ設定されています。

NC VAR セレクタで特定されているエリア番号，行および列の内容は，FB において「0」でチェックします。「0」があれば，入力パラメータからの値が受け入れられます。"PUT" FB を呼出す前に，ユーザは望むパラメータを指定してください (UnitX/ColumnX/LineX)。

マシンデータ , GUD

パスワードなしで，マシンデータおよび GUD を定義するためには，アクセスしたいデータの保護レベルを 1 番低いレベルに再定義してください。この手順については，Installation Guide(インストールガイド) (詳細はプロテクションレベルコンセプトのセクション) および Programming Guide Advanced (上級プログラミングガイド)(ユーザデータ用プロテクションレベル) を参照してください。

!

重要

PLC と NC の間でのコミュニケーション（NC 変数の読取り／書込み，FB 2，3，5，または 汎用 PI サービス，FB 4）が，電源オフあるいは，緊急停止／確認応答／リセットによって中止になった後，コールド再スタートまたはリセットの後で最初の OB 1 内で，スタートジョブを削除してください (信号 : Req = 0)。

FB 3 は，基本プログラムパラメータ NCKomm が "1" にセットされている時だけ NC 変数を書込むことができます (OB 100 では FB1, DB7)。チャンネル別変数を書込んでいるとき，同一のチャンネルからの変数だけが，Addr1 to Addr8 を介してジョブ（FB 3 コール）にアドレス指定します。

エリア V および H では，異なる論理軸番号は 1 つのジョブの中に割当てないでください。(割当ててしまうと，エラーになります。Error:=TRUE, State:=W#16#02)。

1 つのグループの中にある NCK 変数をジョブの中で組合わせることができます。

	エリア				
グループ 1	C[1]	N	B	Q	T
グループ 2	C[2]	N	B	Q	T
グループ 3	V[.]	H[.]			

上記の表で，例としてグループ 1 およびグループ 2 について説明しているように，同じ規則が，チャンネル 3 から 10 までに当てはまります。

宣言

```
FUNCTION_BLOCK FB 3
VAR_INPUT
   Req :       BOOL ;
   NumVar :    INT;
   Addr1 :     ANY ;
   Unit 1 :    BYTE ;
   Column1 :   WORD ;
   Line1 :     WORD ;
   Addr2 :     ANY ;
   Unit 2 :    BYTE ;
   Column2 :   WORD ;
   Line2 :     WORD ;
   Addr3 :     ANY ;
   Unit 3 :    BYTE ;
   Column3 :   WORD ;
   Line3 :     WORD ;
   Addr4 :     ANY ;
   Unit 4 :    BYTE ;
   Column4 :    WORD ;
   Line4 :     WORD ;
   Addr5 :     ANY ;
   Unit 5 :    BYTE ;
   Column5 :   WORD ;
   Line5 :     WORD ;
   Addr6 :     ANY ;
   Unit 6 :    BYTE ;
   Column6 :   WORD ;
```

```
    Line6 :     WORD ;
    Addr7 :     ANY ;
    Unit 7 :    BYTE ;
    Column7 :   WORD ;
    Line7 :     WORD ;
    Addr8 :     ANY ;
    Unit 8 :    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
```

正式パラメータの説明

以下は，PUT 機能のすべての正式パラメータの表です。

信号	種類	タイプ	数値の範囲	メモ
Req	I	Bool		正の信号エッジでジョブがスタート
NumVar	I	Int	1 から 8(Addr1 to Addr8 の使用に適用)	書込まれる変数の数
Addr1 to Addr8	I	Any	[DBName].[VarName]	NC Var セレクタからの変数識別子
Unit 1 to Unit 8	I	Byte		エリアアドレス，変数アドレス指定用オプション
Column 1 to Column 8	I	Word		列アドレス，変数アドレス指定用オプション
Line 1 to Line 8	I	Word		行アドレス，変数アドレス指定用オプション
Error	Q	Bool		ジョブの確認応答は否定，またはジョブの実行は不可能
Done	Q	Bool		ジョブは問題なく実行
State	Q	Word		エラー識別子を見ること
SD1 to SD8	I/Q	Any	P#Mm.n BYTE x...P#DBnr.dbxm.n BYTE x	書込まれるデータ

エラー識別子

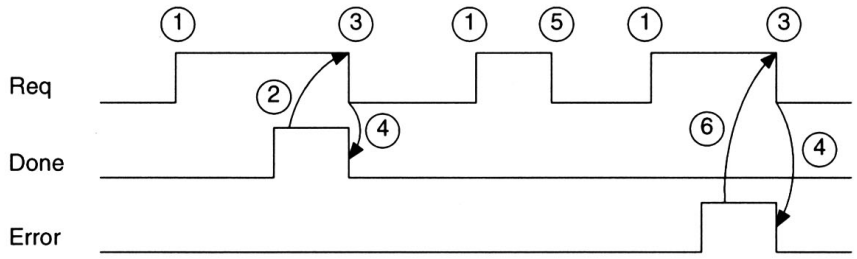
ジョブの実行が失敗すると，ステータスパラメータ Error に "logical 1" で示されます。エラーの原因は，"State" を出力するブロックでコード化されます。

ステータス		意味	注記
WORT-H	WORT-L		
1 から 8	1	アクセスエラー	エラーが発生した変数の，上位バイト番号
0	2	ジョブにおけるエラー	ジョブにおける変数のまちがったコンパイル
0	3	確認応答は否定，ジョブは実行できない	内部エラー，トライ：ジョブをチェック，NC リセット
1 から 8	4	データエリアまたはデータタイプはマッチしない，あるいは文字列になにもない	SD 1 to SD 8 に書込まれるデータをチェック。エラーが発生した変数の，上位バイト番号
0	6	FIFO フル	キューが長すぎるのでジョブを繰返すこと
0	7	オプションがセットされていない	BP パラメータ "NCKomm" はセットされていない
1 から 8	8	不適切なターゲットエリア (RD)	RD1 to RD8 はローカルエリアではない
0	9	転送が占有する	ジョブを繰返すこと
1 から 8	10	変数アドレス指定のエラー	単位または 0 を含む列／行
0	11	無効な変数アドレス，または変数が読取りのみ	アドレス（または変数名），エリア，単位をチェック
0	12	NumVar = 0	パラメータ NumVar をチェック

構成処置

NC 変数書込みに，読取りと同じ構成処置を行ってください。DB において読取りまたは書込みが行われるすべての NC 変数のアドレスデータを保存することは有用です。

タイミングダイアグラム



①機能が起動
②確認応答は肯定：変数が書込まれる
③確認応答受信後，機能起動をリセット
④ FB による信号変換
⑤確認応答受信以前に機能起動をリセットした場合，出力信号はアップデートされず，影響を受けている起動している機能の運転シーケンスのみアップデートされる。
⑥確認応答は否定 : エラーが発生，エラーコードは出力パラメータステータスへ

呼出しの例

チャンネル 1 から，チャンネル別マシンデータ 3 個を書込む

NCK VAR セレクタで 3 個のデータを選択して，DB120.VAR ファイルに保存

エリア	ブロック	名前	タイプ	バイト	S7 名
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 記号エディタで，DB 120 に NCVAR を入力します。

記号	オペランド	データタイプ
NCVAR	DB120	DB120

DB120.STL ファイルをコンパイルして PLC へ転送してください。

インスタンス DB 111 での FB 3 の呼出しおよびパラメータ化：

```
DATA_BLOCK DB 111  // FB 3 用インスタンスとしてユーザ DB は割当てられない
FB 3
BEGIN
Function FC "Variable Call" : VOID
END_DATA_BLOCK
    I 7.7;                          //MCP 上に割当てられていないキー
    S       F 100.0;                //Req 起動
    A       F 100.1;                // メッセージ終了
    R       F 100.0;                // ジョブエンド
    A       I 7.6; /                / マニュアルで確認応答
    A       F 102.0;                // エラーが存在
    R       F 100.0;                // ジョブエンド
    Call fb 3, db 111(
            Req :=      F 100.0,
            NumVar :=   3,          //3 つの変数を書込む
            Addr1 :=    NCVAR.rpa_5C1RP,
            Addr2 :=    NCVAR.rpa_11C1RP,
            Addr3 :=    NCVAR.rpa_14C1RP,
            FMNCNo :=   1,          //FM-NC のみ
            Error :=    F102.0,
            Done :=     F100.1,
            State:=     FW104,
            SD1 :=      P#DB99.DBX0.0 REAL 1,
            SD2 :=      P#DB99.DBX4.0 REAL 1,
            SD3 :=      P#F110.0 REAL 1);
```

変数のアドレス指定の例

アドレスデータが基本タイプとして DB 120 に保存されているチャンネル 1 から，R パラメータ 2 個を書込む。R パラメータ番号は，パラメータ行 X を介してパラメータ化される。

```
DATA_BLOCK DB 120
VERSION : 0.0
STRUCT
   C1_RP_rpa0_0:
   STRUCT
   SYNTAX_ID :          BYTE := B#16#82;
   area_&_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 FB 3 , DB 122 (
                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 :=        F 11.0,
                Done :=         F 11.1,
                State :=        FW 12,
                SD1 :=          P#F 4.0 REAL 1,
                SD2 :=          P#F 24.0 REAL 1);
```

■ FB 4：PI_SERV，汎用 PI サービス

機能説明

FB PI_SERV を使って，NCK エリアにあるインスタンスサービスプログラムをスタートすることができます。この項では，実行できるサービスについて述べます。PI サービスを介した要求は，特定の機能を行う NCK にあるプログラムセクションの実行を開始します（例えば，ツール管理でマガジン内の空きロケーションを検索する）。

ユーザエリアからの個々のインスタンス DB を，それぞれの FB 4 呼出しに割当ててください。マルチインスタンス機能は，SW バージョン 3.7 以降に適用されます。くわしくは，STEP 7 取扱い説明書を参照してください。

指定されたサービスは，パラメータ PI サービスを介して参照されます。選択された PI サービスは，異なるデータタイプでは，ユーザが割当てる追加入力の変数を使用して提供されます（Addr1...Addr4 は文字列用，WVar1...Wvar10 は整数用，またはワード変数用）。制御入力 Req の，正のエッジ変化によって FB 4 が呼出されると，ジョブがスタートします。ジョブが問題なく実行されると，ステータスパラメータ "Done" にロジック信号 "1" が表示されます。すべてのエラーは，Error(エラー) および State(ステータス) で示されます。

データブロック "PI" (DB 16) には，有効な PI サービスが内部に記述されています。信号リストにあるこの DB に名前を割当ててください。"DB-Name.PI-Name" は，FB 4 を呼出す時，PI サービスの実パラメータとして転送されます。

PI サービスの実行は，いくつかの PLC 周期の間続きます（通常は，1～2）。ブロックは，サイクリックモードの時だけ呼出すことができます。

（注）

PLC と NC の間でのコミュニケーション（NC 変数の読取り／書込み，FB 2，3，5，または 汎用 PI サービス，FB 4）が，電源オフあるいは，緊急停止／確認応答／リセットによって中止になった後，コールド再スタートまたはリセットの後で最初の OB 1 内で，スタートジョブを削除してください (信号 : Req = 0)。

FB 4 は，基本プログラムパラメータ NCKomm が "1" にセットされている時だけ PI サービスをスタートすることができます (OB 100 では FB1, DB7)。

宣言

```
FUNCTION_BLOCK FB 4
VAR_INPUT
   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
```

正式パラメータの説明

以下は，PI_SERV 機能のすべての正式パラメータの表です。

信号	種類	タイプ	数値の範囲	メモ
Req	I	Bool		ジョブ要求
PIService	I	Any	[DBName].[VarName] デフォルトは :"PI".[VarName]	PI サービス説明 1)
Unit	I	Int	1 ...	エリア番号
Addr1 to Addr4	I	Any	[DBName].[VarName]	選択された PI サービスに応じた文字列仕様を参照
WVar1 to WVar10	I	Word	1 ...	整数またはワード変数。選択された PI サービスに応じた仕様 （Wvar10 SW バージョン 4 以降）
Error	Q	Bool		ジョブの確認応答は否定，またはジョブは実行できない
Done	Q	Bool		ジョブは問題なく実行
State	Q	Word		エラー識別子を見ること

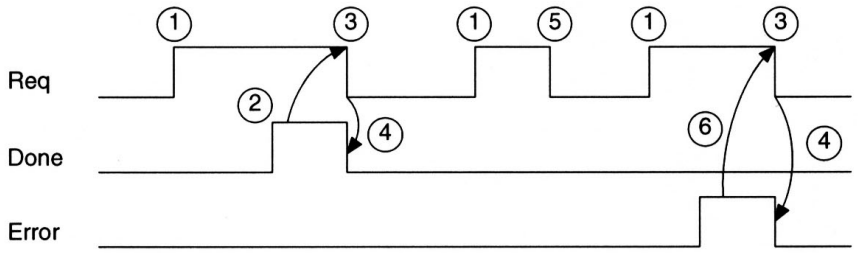
1) 基本プログラムディスクの README ファイル参照

エラー識別子

ジョブの実行が失敗すると，ステータスパラメータ Error に "logical 1" で示されます。エラーの原因は，State を出力するブロックでコード化されます。発生するであろうエラー識別子は以下のとおりです。

ステータス	意味	注記
3	確認応答は否定，ジョブは実行されない	内部エラー， 可能な処置：NC リセット
6	FIFO フル	キューがいっぱいなのでジョブを繰返すこと
7	オプションはセットされていない	BP パラメータ "NCKomm" はセットされていない
9	転送が占有する	ジョブを繰返すこと

タイミングダイアグラム



① 機能が起動
② 確認応答は肯定：PI サービスが実行される
③ 確認応答受信後，機能起動をリセット
④ FB による信号交換
⑤ 確認応答受信以前に機能起動をリセットした場合，出力信号はアップデートされず，影響を受けている起動している機能の運転シーケンスのみアップデートされる。
⑥ 確認応答は否定：エラーが発生，エラーコードは出力パラメータステータスへ

PI サービスの概要

以下の項目では，PLC からスタートすることのできる PI サービスの概要について述べています。変数 (Unit, Addr..., WVar...) を入力した汎用 FB 4 の意味およびアプリケーションは，個々の PI サービスによって異なります。

PI サービス	機能	適用可能性
SELECT	1 つのチャンネルに対して処理用にプログラムを選択	○
ASUP	割当て割込み	○
FINDBL	ブロックサーチ起動	○
SETUFR	ユーザフレーム起動	○
CONFIG	マークしているマシンデータの再構成	○
CANCEL	実行キャンセル	○
DELETO	ツール削除	○
CREATO	ツール生成	○
CREACE	切削エッジ生成	○
TMCRTO	ツール生成	○
TMFDPL	ロードするために空きロケーションを検索	○
TMMVTL	ロード，アンロードのツール用マガジン ロケーションを用意	○
TMPOSM	マガジンロケーションまたはツールの位置付け	○
LOGIN	パスワード起動	○
LOGOUT	パスワードリセット	○
MMCSEM	さまざまな PI サービスのセマフォ	○

PI サービス	機能	適用可能性
CRCEDN	新しい切削エッジ作成	○
DELECE	切削エッジ削除	○
TMFPBP	空きロケーション検索	○
TSEARC	検索画面を使っての複合検索	○
TMPCIT	ワークカウンタにインクリメント値をセット	○
DIGION	座標読取りオン	○
DIGIOF	座標読取りオフ	○
NCRES	NC リセット開始	○
TMRASS	アクティブステータスリセット	○
ACTDEF	定義起動 (GUD またはマクロ)	○
○ : PI サービスは有効		

PI サービス :SELECT

チャンネル上の処理のためにプログラムを選択

機能 :

NCK に保存されているプログラムで，1 つのチャンネルを処理するために選択します。これは，ファイルを実行する場合のみ可能となります。プログラミングガイド（"File and Program Management" （ファイルおよびプログラム管理）セクション，"Program memory" （プログラムメモリ）サブセクション）の説明通りに，パス名およびプログラム名を入力してください。パス名およびプログラム名の表記については，FB 4 の例を参照にしてください。

表 1.47 実行可能なブロックタイプ

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

表 1.48 実行可能なブロックタイプ

ブロックタイプ	
サブプログラム	SPF
サブサイクル	CYC
非同期サブプログラム	ASP
バイナリファイル	BIN

表 1.49 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.SELECT	プログラム選択
Unit	INT	1 から 10	チャンネル
Addr1	STRING		パス名
Addr2	STRING		プログラム名

PI サービス : ASUP

割当て割込み

機能 :

NCK に保存されているプログラムで，割込み信号をチャンネルに割当てます。これは，実行するファイルでのみ可能となります。プログラミングガイド（"File and Program Management" (ファイルおよびプログラム管理) セクション，"Program memory"（プログラムメモリ）サブセクション）の説明通りに，パス名およびプログラム名を入力してください。パス名およびプログラム名の表記については，FB 4 の例を参照にしてください。

表 1.50 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.ASUP	割当て割込み
Unit	INT	1 から 10	チャンネル
WVar1	INT		割込み番号
WVar2	INT		優先度
WVar3	INT	0/1	LIFTFAST
WVar4	INT	0/1	BLSYNC
Addr1	STRING		パス名
Addr2	STRING		プログラム名

（注）SETINT 命令は，割当ての実行にも使用します。
PI サービス ASUP は，起動するチャンネルが Reset ステータスの時だけ実行します。

参照： /PA/, Programming Guide

PI サービス : FINDBL

検索起動

機能 :

チャンネルは，ブロック検索モードへ切換えられて，適切な確認応答の後，転送されます。ブロック検索は，NCK によってただちに実行されます。検索ポインタが，NCK のこのポイントにすでに在るようにしてください。この検索は , NC RESET でいつでも中断することができます。検索が問題なく完了すると，標準処理モードが自動的に再起動します。NC Start は，検索ターゲットの位置するところから働きます。オペレータには，衝突しないアプローチパスを規定する責任があります。

表 1.51 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.FINDBL	ブロック検索
Unit	INT	1 から 10	チャンネル
WVar1	WORD	x	事前処理モード
x: 事前処理モードを定義 x = 1: 計算なし x = 2 計算あり x = 3 メインブロックの考慮あり			

PI サービス :
SETUFR

ユーザフレーム起動

機能 :

ユーザフレームは，NCK にロードします。あらかじめ FB 3 で変数を書込んで，必要なフレームの値をすべて NCK へ転送してください。

表 1.52 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.SETUFR	ユーザフレーム起動
Unit	INT	1 から 10	チャンネル

PI サービス :
CONFIG

再構成

機能 :

"Reconfigure" (再構成) コマンドで，ユーザあるいは PLC によって連続的に入力されているマシンデータに対し擬似的な並列起動を行うことができます。このコマンドは，RESET ステータスあるいはプログラム割込み（ブロックリミットでの NC ストップ）の後の制御でのみ起動します。これらの状態が実行されない場合（ステータス = 3)，FB 4 エラーチェックバックメッセージが出力されます。

表 1.53 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.CONFIG	再構成
Unit	INT	1	
WVar1	INT	1	分類

PI サービス :
CANCEL

実行キャンセル

機能 :

CANCEL コマンドで，"Cancel" (キャンセル) 機能が起動します（MMC のキーによる）。

表 1.54 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.CANCEL	キャンセル

PI サービス :
DELETO

ツール削除

機能 :

すべての切削エッジの転送された T 番号に割当てられたツールを削除します（TO, TU のいくつか，TUE および TG (タイプ 400)，TD および TS ブロック内）。

表 1.55 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.DELETO	ツール削除
Unit	INT	1, 2	TOA
WVar1	INT		T 番号

PI サービス :
CREATO

ツール作成

機能 :

T 番号を特定して，ツールを生成します。ツールディレクトリエリア (TV) に存在するツールを入力します。最初の切削エッジ "D1" （ゼロ内容）は，TO ブロックにあるツールオフセット用に作成されます。"D1" （ゼロ内容）は，TUE ブロック（がある場合）の OEM 切削エッジデータ用にも作成されます。TU ブロックが存在すれば，ツール用のデータブロックが実装されています。

表 1.56 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.CREATO	ツール作成
Unit	INT	1, 2	TOA
WVar1	INT		T 番号

PI サービス : CREACE

切削エッジ作成

機能 :

TO, TS (がある場合) にある転送された T 番号で転送されたツールのために，次に高い D 番号／次の割当てられていない D 番号で，切削エッジを作成します。OEM 切削エッジデータ用の切削エッジは，TUE ブロック（がある場合）に同時にセットアップします。

表 1.57 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.CREACE	ツール切削エッジ作成
Unit	INT	1, 2	TOA
WVar1	INT		T 番号

PI サービス : TMCRTO

ツール作成

機能 :

識別子，二重番号および T 番号 (オプション) を特定して，ツールを作成します。ツールディレクトリエリア (TV) に存在するツールを入力します。最初の切削エッジ "D1" （ゼロ内容）は，TO ブロックにあるツールオフセット用に作成します。"D1" （ゼロ内容）は，TS ブロックのモニタリングデータ用にもセットアップして，同時に TUE ブロック（がある場合）の OEM 切削エッジデータ用にも，ゼロ内容でセットアップします。TD ブロックには，識別子，二重番号，および任意に入力された，または NCK によって割当てられた T 番号用の切削エッジ（=1）の番号があります。TU ブロックが存在すれば，ツール用のデータブロックが実装されています。

PI の実行後，作成されたツール用の T 番号は，TnumWZV で，TV ブロックに保存します。

（注）この PI サービスの前後 , 関連パラメータ WVar1 で，この PI サービス用に MMCSEM PI サービスを呼出してください。詳しくは，PI サービス MMCSEM を参照してください。

表 1.58 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.TMCRTO	ツール作成
Unit	INT	1, 2	TOA
WVar1	INT		T 番号
WVar2	INT		二重番号
Addr1	STRING	最大 32 文字	ツール識別子

T 番号 > 0 は，T 番号を入力してください。
T number = -1 は，NCK で T 番号を割当ててください。
Example shows T number = -1 ==> NCK による T 番号割当て。

PI サービス：
TMFDPL

ロード用空きロケーションの検索

機能：

(パラメータ割当てに依存する)

Location_number_where = -1, magazine_number_where = -1:

指定されたエリア（= チャンネル）のすべてのマガジンを通して，T 番号によって指定されたツールのための空きロケーションを検索します。検索中に見つかったマガジンおよびロケーション番号は，PI の実行後 , チャンネル（コンポーネント magCMCmdPar1 (マガジン番号)）および magCMCmdPar2 (ロケーション番号) の構成ブロックに，リスト出力されます。Location_number_ID および magazine_number_ID は，検索基準としてセットできるか，またはセットできません（=-1）。PI は，確認応答肯定となるか，確認応答否定となるか検索結果によって決まります。

Location_number_where = -1, magazine_number_where = magazine number:

T 番号によって指定されたツールをロードするための空きロケーションの検索を，指定されたマガジンで実行します。Location_number_ID および magazine_number_ID は，検索基準としてセットできるか，またはセットできません（=-1）。PI は，確認応答肯定となるか，確認応答否定となるか検索結果によって決まります。

Location_number_where = location number, magazine_number_where = magazine number:

指定されたロケーションは，指定されたツールにロードする空きがあるかどうかを立証するため調査されます。Location_number_ID および magazine_number_ID は，検索基準としてセットできるか，またはセットできません（=-1）。PI は，確認応答肯定となるか，確認応答否定となるか検索結果によって決まります。

コマンドパラメータ 1 および 2 は，ソースに置きます。

ロード : ソースが内部ロードマガジンである場合，コマンドパラメータはデータ転送先（リアルマガジン）に置きます。

アンロード : ソースは，常にリアルマガジンにあります。

（注）この PI サービスの前後 , 関連パラメータ WVar1 で，この PI サービス用に MMCSEM PI サービスを呼出してください。詳しくは，PI サービス MMCSEM を参照してください。

表 1.59 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.TMFDPL	ロード用空きロケーション
Unit	INT	1, 2	TOA
WVar1	INT		T 番号

WVar2	INT		Location_num-ber_where
WVar3	INT		Magazine_num-ber_where
WVar4	INT		Location_num-ber_ID
WVar5	INT		Magazine_num-ber_ID

PI サービス :
TMMVTL

ツールのロード，アンロード用マガジンロケーションの用意

機能 :

この PI サービスは，ツールのロードおよびアンロードのどちらにも使われます。PI がロード運転を開始するのかアンロード運転を開始するのかは，リアルロケーションと，"from" パラメータおよび "to" パラメータの間の割当てによって決まります。: Loading =>"from" = ロードポイント／ステーション , unloading =>"to" = ロードポイント／ステーション。

PI サービス TMMVTL は，すべての動作に使用されます。

1) ロードおよびアンロード (ロードポイント <-> マガジン)

2) ロードおよびアンロード (ロードポイント <-> バッファ , 例えば，スピンドル)

3) マガジン内の再ロケーション

4) 異なるマガジン間の再ロケーション

5) マガジンとバッファの間の再ロケーション

6) バッファ内の再ロケーション

以下の TM ブロックからの変数は，ケース 1), 3), 4), 5) を監視するために使用します。

magCmd (エリア番号 = TO unit, 行 = マガジン番号)

magCmdState <- 「確認応答」

以下の TMC ブロックからの変数は，ケース 2), 6) を監視するために使用します。

magCBCmd (エリア番号 = TO unit)

magCBCmdState <- 「確認応答」

ロード機能

ロードするために，特定されたリアルマガジンを特定されたチャンネルに用意します。例えば，特定されたロードポイント／ステーションで，選択されたロケーションへロードするためのマガジンの移動 (location_number_from, magazine_number_from) およびツールの入力，などです。location_number_to = -1 の場合，T 番号によって指定されたツール用の空きロケーションは，最初にマガジン内でシークされ，そのマガジンが移動します。PI の実行後，検索中に見つかったロケーション番号は，チャンネルのリアルマガジンのコンポーネント magCMCmdPar2 にある TM エリアに，リスト出力されます。

location_number_to = -2 および有効なマガジン番号で，指定されたマガジンの 1 番近い空き位置へ，ツールをロードします。PI の実行後，ツールをロードしたロケーション番号は，チャンネルのリアルマガジンのコンポーネント magCMCmdPar2 にある TM エリアに，リスト出力されます。

アンロード機能

ツール番号で指定されたツールは，指定されたロードポイント／ステーションでアンロードします (location_number_to, magazine_number_to), すなわち，アンロードする位置へマガジンを移動して，ツールを移動します。ツール用のマガジンロケーションは，TP ブロック内で "free" 状態として表示されています。T 番号，またはロケーションおよびマガジン番号のどちらを介しても，ツールを指定することができます。値 -1 を，使用していない指定ポイントに入力します。

（注）この PI サービスの前後 , 関連パラメータ Wvar1 で，この PI サービス用に MMCSEM PI サービスを呼出してください。詳しくは，PI サービス MMCSEM を参照してください。

表 1.60 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.TMMVTL	ロード，アンロードツール用に用意するマガジンロケーションを作成
Unit	INT	1, 2	TOA
WVar1	INT		T 番号
WVar2	INT		Location_num-ber_from
WVar3	INT		Magazine_num-ber_from
WVar4	INT		Location_number_to
WVar5	INT		Magazine_number_to

PI サービス : TMPOSM

マガジンロケーションまたはツールの位置付け

機能 : (パラメータ割当てに依存する)

ツールを介して直接指定された，あるいは限定されたマガジンロケーションを，指定された位置へ PI サービスを介して移動します（ロードロケーションの前，など)。

PI サービスで，さまざまな方法によって限定することのできるマガジンロケーションを，指定されたロードロケーションの前に移動します。ロードロケーションを，PI パラメータ location number_from および magazine number_from で指定してください。移動するマガジンロケーションは，以下のとおり限定します。

- ツールの T 番号：ツールが移動に位置付けられるところ；パラメータ tool identifier, duplo number, location number_from および magazine number_from は，関連性がない (すなわち，値 "," "-0001","-0001","-0001")

または

- ツール識別子および二重番号：ツールが移動に位置付けられるところ；パラメータ T number, location number_from および magazine number_from は，関連性がない (すなわち，値はそれぞれ "-0001")

または

- パラメータ location number_from および magazine number_from で，ロケーションを直接指定 ; ツールを特定したパラメータ T number, tool identifier および duplo number は，関連性がない (すなわち，値 "-0001", "", "-0001")。

表 1.61 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.TMPOSM	マガジンロケーションまたはツールの位置付け
Unit	INT	1, 2	TOA
Addr1	STRING	最大 32 文字	ツール識別子
WVar1	INT		T 番号
WVar2	INT		二重番号
WVar3	INT		Location_num-ber_from
WVar4	INT		Magazine_num-ber_from
WVar5	INT		Location num-ber_ref
WVar6	INT		Magazine num-ber_ref

PI サービス : LOGIN

パスワード作成

機能 :

パラメータ化されたパスワードを NCK へ転送します。パスワードは通常 8 文字で作成します。もし必要であれば，パスワードの中にブランクを入れてください。

例

Password : STRING[8] := ’SUNRISE _ ’;

表 1.62 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.LOGIN	パスワード作成
Unit	INT	1	NCK
Addr1	STRING	8 文字	パスワード

PI service: LOGOUT

パスワードリセット

機能 :

NCK へ最後に転送されたパスワードをリセットします。

表 1.63 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.LOGOUT	リセットパスワード
Unit	INT	1	NCK

PI サービスのセマフォ

MMCSEM

MMC および PLC による使用

それぞれのチャンネルには，10 のセマフォがあります。これらは，MMC/PLC の重要な機能を保護します。セマフォを，対応する機能番号にセットすると，NCK によって取出されたデータに関する重要なセクションを含む機能がある場合，いくつかの MMC/PLC ユニットがこれに同期します。セマフォは MMC/PLC によって管理されています。セマフォ値 1 は，指定された機能番号のセマフォの Test & Set 運転，と規定されています。PI サービスのリターン値は，このオペレーションの結果を表しています。

リターン値 OK:

セマフォがセットされる，重要な機能の呼出しができる

リターン値 REJECTED:

セマフォはすでにセットされているが，重要な機能を現在呼出すことができない。オペレーションを後で繰返してください。

（注）オペレーションの完了（この PI サービスのデータ読取り）
時，セマフォを再びイネーブルにすることが絶対に必要です。

パラメータ：

WVar1= 機能番号

この機能番号は，PI サービスを表しています :

1: TMCRTO (ツール作成)

2: TMFDPL (ロード用空きロケーションの検索)

3: TMMVTL (ツールのロード，アンロード用マガジンロケーションの用意)

4: TMFPBP (ロケーションの検索)

5: TMGETT (ツール番号の検索)

6: TSEARC (ツールの検索)

7 ... 10: リザーブ

WVar2= セマフォ値

0: セマフォをリセット

1: セマフォをテストおよびセット

パラメータ化

表 1.64 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.MMCSEM	新しい切削エッジの作成
Unit	INT	1, 2 から 10	チャンネル
WVar1	INT	1 から 10	機能番号
WVar2	Word	0, 1	セマフォ値

新しい切削エッジの作成

CRCEDN

機能

切削エッジ番号を特定してツールエッジを作成します。

現在あるツールの T 番号が，PI サービスにおいてパラメータ "T number" で指定されている場合，この特別なツール用に切削エッジをセットアップします（この場合，パラメータ "D number" （作成される切削エッジの番号）の数値の範囲は，00001 - 00009 です)。正の T 番号をパラメータとして指定した場合，および T 番号が入力されたツールが存在しない場合，PI サービスは中止されます。T 番号（絶対 D 番号の型）の値を 00000 で入力した場合，D 番号の数値の範囲は，00001 から 31999 まで拡大します。特定された D 番号で，切削エッジをセットアップします。指定された切削エッジがすでに存在する場合，PI サービスは中止されます。

パラメータ化

表 1.65 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.CRCEDN	新しい切削エッジの作成
Unit	INT		TOA
WVar1	INT		作成される切削エッジのツール番号。00000　の設定で，切削エッジはどんな特定ツールも参照できないことになる(絶対 D 番号)。
WVar2	INT	1 - 9 または 01 - 31999	ツール切削エッジのエッジ番号

ツール切削エッジの削除

削除

機能

現在あるツールの T 番号が，PI サービスにおいてパラメータ "T number" で指定されている場合，このツール用切削エッジを削除します（この場合，パラメータ "D number" （作成される切削エッジの番号）の数値の範囲は，00001 - 00009 です)。正の T 番号がパラメータとして指定された場合，および T 番号が入力されたツールが存在しない場合，PI サービスは中止されます。T 番号（絶対 D 番号の型）の値を 00000 で入力した場合，D 番号の数値の範囲は，00001 から 31999 まで拡大します。指定された切削エッジがすでに存在する場合，PI サービスは中止されます。

パラメータ化

表 1.66 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.DELETE	切削エッジ削除
Unit	INT		TOA
WVar1	INT		削除される切削エッジのツール番号。00000　の設定で，切削エッジはどんな特定ツールも参照できないことになる（絶対 D 番号)。
WVar2	INT	1 - 9 または 01 - 31999	削除されるツール切削エッジのエッジ番号

空きロケーション検索

TMFPBP

機能 : (パラメータ割当てに依存する)

FB7 の説明を参照してください。

サーチマスクによる複合検索

TSEARC

機能 : (パラメータ割当てに依存する)

PI サービスを使って，サーチドメインの中（特定のロケーションでスタートおよび終了する，1 つ以上のマガジン内）にある指定されたプロパティを持つツールを検索することができます。この指定されたプロパティは，ツールおよびツールの切削エッジのデータのみを参照します。PI サービスは，ツール管理が起動している時のみ有効です。PI サービス用に検索の指示およびヒットする数を定義することができます（例えば，プロパティが一致している次のツールに対する「1 つのツール」なのか，あるいは指定されたプロパティを持つすべてのツールに対するものなのか)。このサービスで，サーチで見つかったツール用の内部 T 番号のリストを元の状態に復元します。

サーチ基準は，ANDed のみです。サーチ基準のために，アプリケーションが OR オペレーションを定義する必要がある場合，ANDed　基準で一連の問合せをまず実行してから，個々の問合せの結果を，組合わせ／評価してください。

PI サービスのパラメータを割当てるには，まず必要なツールのプロパティを，TF ブロックにある変数サービスを介して定義します。この定義を行うためには，比較オペレータデータ (parDataT..) にある比較論理 (==, <, >, <=, >=, &&) のタイプを入力し，TF ブロック内のオペランドマスク (parMaskT..) にある適切な比較基準を選択して（すなわち，どのツールデータが比較されるのか？），オペランドデータの比較値を入力します。その結果，PI サービスが開始されて，正しく完了した後，TF ブロックからの変数サービスで，変数 resultNrOfTools および 変数 resultNrOfToolNr にある結果リストにおいてヒットした数を読出します（すなわち，サーチ—resultNrOfTools quantity において見つかったツールの内部 T 番号のリスト)。PI サービスを，準備段階から正しく完了するまでの間，セマフォでカプセル化してください。これは，排他アクセスを確保して，TSEARC PI サービスと共に TF ブロックを排他的に使用するための唯一の方法です。セマフォ機能用に与えられた機能番号 (PI サービス MMCSEM) は，TSEARC 用の機能番号です。

サービスが間違って構成された場合，誤作動が起こります。中でも，ツールがまったく見つからない場合にでも，結果が返ってくることもあります。(resultNrOfTools = 0).

サーチドメインは，次のパラメータによって下記のように定義することができます。

MagNrFrom, PlaceNrFrom, MagNrTo, PlaceNrTo:

MagNrFrom	PlaceNrFrom	MagNrTo	PlaceNrTo	サーチエリア
WVar1	WVar2	WVar3	WVar4	
#M1	#P1	#M2	#P2	マガジン #M1 でスタートするロケーション，マガジン #M2 までのロケーション #P1, ロケーション #P2 を検索
#M1	-1	#M1	-1	マガジン #M1 にあるすべてのロケーション - これのみ - を検索
#M1	-1	-1	-1	マガジン #M1 でスタートするすべてのロケーションを検索
#M1	#P1	-1-	1	マガジン #M1 でスタートするすべてのロケーション，およびロケーション #P1 を検索
#M1	#P1	#M1	-1	マガジン #M1 でスタートする マガジン #M1 にあるロケーション，およびこのマガジンにあるロケーション #P1 を検索
#M1	#P1	#M2	-1	マガジン #M1 でスタートするロケーション , およびこのマガジンまでにあってマガジン #M2 を含むロケーション #P1 を検索
#M1	-1	#M2	#P2	マガジン #M1 までにスタートしてマガジン #M2 を含むロケーション，およびこのマガジンにあるロケーション #P2 を検索
#M1	-1	#M2	-1	マガジン #M1 までにスタートしてマガジン #M2 を含むロケーションを検索
-1	-1	-1	-1	すべてのマガジンロケーションを検索

対称検索 (パラメータ "SearchDirection" 参照) をするには , 検索ドメインを 1 つのマガジンに加えるだけです（上記の表にあるケース 2 および 5)。別のサーチドメインを指定した場合，サービスは誤作動を起こします。対称検索のために，パラメータ，参照ロケーションを指定してください (パラメータ "SearchDirection" 参照)。参照ロケーションは，パラメータ MagNrRef および PlaceNrRef に指定されます。参照ロケーションは，バッファロケーション（マガジンバッファからのロケーション，すなわち位置変更，グリッパー ...)，あるいはロードポイント（内部ロードマガジンからのロケーション）です。指定された参照ロケーションの前にあるマガジンロケーションを参照して，検索は対称に実行されます。

マガジンへの多重割当てを，TPM ブロック内で参照ロケーション用に構成してください。そうでない場合は，誤作動が起きます。参照ロケーションの前にあるマガジンロケーションが，検索ドメインの外にある場合，サービスは，一致するロケーションを見つけられなかったかのように対応します。

（注）この PI サービスの前後 , 関連パラメータ WVar1 で，この PI サービス用に MMCSEM PI サービスを呼出してください。詳しくは，PI サービス MMCSEM を参照してください。

PI サービスは，NCK SW バージョン 4.2 から有効になります。

パラメータ化

表 1.67 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.TSEARC	サーチスクリーン形式を使用した複合検索
Unit	INT	1, 2	TOA
WVar1	INT		検索を始めるマガジンの MagNrFrom Magazine 番号
WVar2	INT		マガジン MagNrFrom にあるロケーションの PlaceNrFrom ロケーション番号 , 検索を開始するマガジンにおいて
WVar3	INT		検索を終了するマガジンのマガジン MagNrTo 番号
WVar4	INT		検索を終了するマガジン MagNrTo にあるロケーションのロケーション PlaceNrTo 番号
WVar5	INT		対称検索を行って参照するマガジン (内部) のマガジン MagNrRef 番号。（このパラメータは，「対称的な」検索方向にのみ関連する。）
WVar6	INT		対称検索を行って参照するマガジン MagNrRef ロケーションのロケーション PlaceNrRef 番号。このパラメータは，「対称的な」検索方向にのみ関連する。
WVar7	INT	1, 2, 3	SearchDirection (検索方向) は，要求する検索方向を特定する 1 検索ドメインの最初のロケーションから前へ進む 2 検索ドメインの最後のロケーションから逆戻りする 3 MagNr-Ref および PlaceNr-Ref によって指定されたロケーションの前に位置するリアルマガジンロケーションで，対称的に行う。
WVar8	INT	0, 1	KindofSearch 値 0: このプロパティのすべてのツールを見つける Value 1: このプロパティの最初のツールを見つける

ワークカウンタのインクリメント値のセット

TMPCIT

機能:

主軸ツールのワークカウンタをインクリメントする。

パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.TMPCIT	ワークカウンタのインクリメント値をセットする
Unit	INT	1 から 10	TOA
WVar1	WORD	0 ... 最大	主軸番号；チャンネル内のバッファマガジンの主軸ロケーションタイプと共に，ロケーションデータにあるタイプインデックスに対応する。000 = メイン主軸
WVar2	WORD	0 ... 最大	インクリメント値；ワークカウンタがインクリメントされた後の主軸が回転した数を示す。

アクティブステータスのリセット

TMRASS

機能:

摩滅したツールのアクティブステータスをリセットする

この PI サービスを使って，すべてのツールに対して，ツールステータスのアクティブおよびディスエーブルを検索します。これらのツール用のアクティブステータスはキャンセルされます。場合によっては，この PI サービスにかかる時間が，VDI 信号 "tool disable ineffective" の負のエッジとなって，プログラムを終了，あるいはチャンネルをリセットします。この PI サービスは，ディスエーブルツールが最終的に使用できなくなる時がわかるので，主に PLC 用になっています。

パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI. TMRASS	アクティブステータスをリセット
Unit	INT	1 から 10	TO エリア

デジタイジングオン (DIGION)

機能:

指定されたチャンネルでデジタイジングを選択する

パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.DIGION	デジタイジングオン
Unit	INT	1 から 10	チャンネル

デジタイジングオフ (DIGIOF)

機能:

特定のチャンネルでデジタイジングを停止する

パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.DIGIOF	デジタイジングオフ
Unit	INT	1 から 10	チャンネル

NC Reset 開始

NCRES

機能:

NCK Reset を開始する。Unit およびパラメータ WVar1 を 0 に割当ててください。

パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.NCRES	NC Reset を開始
Unit	INT	0	0
WVar1	WORD	0	0

GUD またはマクロアクティブの定義

ACTDEF

機能:

NCK に保存されているプログラムを，処理のため 1 つのチャンネルに選択します。これは，ファイルが実行されている時のみ可能となります。パス名およびプログラム名を，Programming Guide (プログラムガイド)（セクション "File and program management" (ファイルおよびプログラム管理), サブセクション "Program memory" (プログラムメモリ)）の説明にしたがって入力してください。パスの表記およびプログラム名については，FB 4 の例もあわせて参照してください。

パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.ACTDEF	GUD あるいはマクロを起動
Unit	INT	1	
Addr1	STRING		パス名
Addr2	STRING		GUD あるいはマクロ

呼出しの例

チャンネル 1 にプログラムを選択する（メインプログラムおよびワークプログラム）

S7 SYMBOL エディタで，DB 16 に PI，DB 124 に STR を入力します。

パラメータ化

記号	オペランド	データタイプ
PI	DB16	DB16
STR	DB124	DB124

```
DATA_BLOCK DB 126                    // FB 4 のインスタンスとして割当てられていないユーザ DB
FB 4
BEGIN
END_DATA_BLOCK
DATA_BLOCK db 124
    struct
    PName:          string[32]:= '_N_TEST_MPF ';
    Path:           string[32]:= '/_N_MPF_DIR/';          // メインプログラム
    PName_WST:      string[32]:= '_N_ABC_MPF';
    Path_WST:       string[32]:= '/_N_WKS_DIR/_N_ZYL_WPD';
                                                          // ワークプログラム
    end_struct
BEGIN
END_DATA_BLOCK
Function FC "PICall" : VOID
    call fb4,db126(
    I 7.7;                      //MCP 上に割当てられていないキー
    S           F 0.0;          //Req 起動
    A           F 1.1;          // メッセージ終了
    R           F 0.0;          // ジョブ終了
    A           I 7.6;          // マニュアルでエラー確認応答
    A           F 1.0;          // エラーが存在する
    R           F 0.0;          // ジョブ終了
                Req :=          F0.0,
                PIService :=    PI.SELECT,
                Unit :=         1,              // CHAN 1
                Addr1 :=        STR.Path,
                Addr2 :=        STR.PName,      // メインプログラム選択
                                //Addr1:=STR.Path_WST,
                                //Addr2:=STR.PName_WST,
                                // ワークプログラム選択
                Error :=        F1.0,
                Done :=         F1.1,
                State :=        MW2);
```

■ FB 5: GETGUD, GUD 変数読取り

機能の説明

FB GETGUD を使用して，PLC ユーザプログラムで，NCK およびチャンネルエリアにある GUD 変数（GUD = Global User Data, グローバルなユーザが定義するデータ）を読取ることができます。

GUD 変数の名前には，大文字を使用してください。すべての FB 5 呼出しを，ユーザエリアから，個々のインスタンス DB へ割当ててください。(マルチインスタンス機能は，SW 3.7 以降)。FB 5 が，制御入力 Req において正のエッジ交換で呼出されると，ジョブがスタートします。このジョブには，データタイプ "STRING" でパラメータ "Addr" において読取りを行う GUD 変数の名前があります。GUD 変数の名前を指すポインタを , <Data-block Name> で，パラメータ "Addr" <Variable name> にシンボリックに割当てます。この変数についての追加の情報は，パラメータ "Area", "Unit", "Index1" および "Index2" に詳しく記しています (ブロックパラメータの表を参照してください)。

パラメータ "CnvtToken" が起動している場合，変数ポインタ（トークン）が，この GUD 変数用にオプションとして生成されます。このポインタは，NC のシステム変数用に，VAR セレクタ（1.15.3「■ NC VAR セレクタ」を参照）を介して生成されます。GUD 変数に有効なポインタを生成するのは，この方法だけです。GUD 変数用にポインタが生成されると，ポインタを参照にして，FB 2 および FB 3 (GET, PUT) を介して GUD 変数を読取り／書き込みすることができます。GUD 変数を読取ることができるのは，この方法だけです。FB 2 または FB 3 がパラメータ化している場合，変数ポインタ用に，パラメータ Addr1 to Addr8 だけをパラメータ化する必要があります。GUD 変数フィールドは，例外となります。これらの中にある Line1...Line 8 を，この変数のフィールドインデックスでパラメータ化してください。

読取りプロセスが無事に完了すると，ステータスパラメータ NDR に論理的「1」が示されます。

読取りプロセスは，いくつかの PLC 周期にわたって続きます（通常は，1 から 2）。ブロックは，周期的モードの時だけ呼出すことができます。

すべてのエラーは，Error および State で示されます。

!

重要

PLC と NC の間でのコミュニケーション（NC 変数の読取り／書込み，FB 2，3，5，または 汎用 PI サービス，FB 4）が，電源オフあるいは，緊急停止／確認応答／リセットによって中止になった後，コールド再スタートまたはリセットの後で最初の OB 1 内で，スタートジョブを削除してください (信号 : Req = 0)。

FB 5 は，基本プログラムパラメータ NCKomm が "1" にセットされている時だけ GUD 変数を読取ることができます (OB 100 では FB1, DB7。セクション 4.1 を参照)。

宣言

```
FUNCTION_BLOCK FB 5                    // サーバ名
    KNOW_HOW_PROTECT
    VERSION: 3.0
VAR_INPUT
    Req:          bool;
    Addr:         any;                 // 変数文字列名
    Area:         byte;                // エリア : NCK = 0, チャンネル channel = 1
    Unit:         byte;
    Index1:       INT;                 // フィールドインデックス 1
    Index2:       INT;                 // フィールドインデックス 2
    CnvtToken:    BOOL ;               //10 バイトトークンに変換
    VarToken:     ANY;                 //10 バイトで変数トークンを構築
END_VAR
VAR OUTPUT
    Error: bool;
    Done: bool;
    State: word;
END_VAR
VAR_IN_OUT
    RD:           any;
END_VAR
BEGIN
END_FUNCTION_BLOCK
```

正式パラメータの説明

以下は，GETGUD 機能のすべての正式パラメータの表です。

信号	種類	タイプ	数値の範囲	メモ
Req	I	Bool		正のエッジでジョブを開始
Addr	I	Any	[DBName].[VarName]	データタイプ STRING の変数にある GUD 変数名
Area	I	Byte		エリアアドレス : 0: NCK 変数 2: チャンネル変数
Unit	I	Byte		NCK エリア : Unit : 0 チャンネルエリア : チャンネル番号
Index1	I	Int		変数のフィールドインデックス 1 フィールドインデックスが使われていない場合は，変数の値は "0"。
Index2	I	Int		変数のフィールドインデックス 2 フィールドインデックスが使われていない場合は，変数の値は "0"。
CnvtToken	I	Bool		変数トークンの生成を起動
VarToken	I	Any		10 バイトトークンにアドレス指定（例を参照）
Error	Q	Bool		ジョブは確認応答は否定，または実行されない
Done	Q	Bool		ジョブは問題なく実行
State	Q	Word		エラー識別子を見ること
RD	I/Q	Any	P#Mm.n BYTE x...P#DBnr.dbxm.n BYTE x	書込まれるデータ

エラー識別子

ジョブの実行が不可能な場合，ステータスパラメータ Error (エラー) の "logic 1" でその不履行が示されます。エラーの原因は，"State" (ステータス) を出力するブロックでコード化されます。

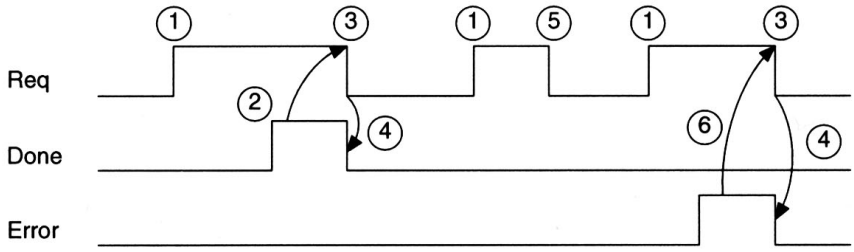
ステータス		意味	注記
WORT-H	WORT-L		
0	1	アクセスエラー	
0	2	ジョブにおけるエラー	ジョブにある変数のまちがったコンパイル
0	3	確認応答は否定，ジョブは実行されない	内部エラー , トライ : NC リセット
0	4	データアクセスまたはデータタイプが一致していない	RD で読取られるデータをチェック ;
0	6	FIFO フル	キューがいっぱいなので，ジョブを繰返すこと

ステータス		注記	意味
WORT-H	WORT-L		
0	7	オプションはセットされていない	BP パラメータ "NCKomm" はセットされていない
0	8	不適切なターゲットエリア (RD)	RD はローカルエリアではない
0	9	転送が占有する	ジョブを繰返すこと
0	10	アドレス指定におけるエラー	Unit は 値が 0
0	11	変数のアドレスが無効	アドレス，エリア , 単位をチェック (または変数名),

構成方法

GUD 変数を読取るためには，GUD 変数名を変数文字列に保存してください。この変数文字列のあるデータブロックを，記号リストに定義してください。そうすれば，パラメータ "Addr" は，FB GETGUD にシンボリックに割当てられることになります。変数の構成は，変数ポインタを受取るために，PLC のどこのデータエリアでも，オプションで定義することができます（下記の例にある仕様を参照)。

パルスダイアグラム



① 機能が起動
② 確認応答は肯定：変数は読取られている
③ 確認応答受信後，機能の起動をリセット
④ FB を使用して信号交換
⑤ 確認応答受信以前に機能起動をリセットした場合，出力信号はアップデートされず，影響を受けている起動している機能の運転シーケンスのみアップデートされる。
⑥ 確認応答は否定 : エラーが発生，エラーコードは出力パラメータステータスへ

呼出しの例

GUD 変数を，"GUDVAR1" の名前で整数変数として読取ります。(FB2 の表「SIMATIC データタイプにおける NC データタイプの割当て」を参照)

インスタンス DB 111 のある FB 5 の呼出しおよびパラメータ化

```
DATA_BLOCK DB GUDVAR                    // 記号リストにおいて割当てを行う
STRUCT
   GUDVar1 : STRING[32] := 'GUDVAR1'; // ユーザによる名前の定義
   GUDVar1T :
   STRUCT
      SYNTAX_ID : BYTE;
      area_&_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
DATA_BLOCK DB 111                       //FB 5 用インスタンスとして使用していないユーザ DB
FB 5
BEGIN
END_DATA_BLOCK
// チャンネル 1 からユーザが定義したチャンネル変数を，引続き変数を書込めるように，変換して変数ポインタへ読取る
Function FC "Variable Call" : VOID
   I 7.7;                               //MCP 上に割当てられていないキー
   S F 100.0;                           //Req 起動
   A F 100.1;                           // メッセージ終了
   R F 100.0;                           // ジョブエンド
   A I 7.6;                             // マニュアルでエラー確認応答
   A F 102.0;                           // エラーが存在
   R F 100.0;                           // ジョブエンド
   Call fb 5, db 111(
               Req :=          F 100.0,    // 読取りのためにエッジをスタート
               Addr :=         GUDVAR.GUDVar1,
               Area :=         B#16#2,     // チャンネル変数
               Unit :=         B#16#1,     // チャンネル 1
               Index1 :=       0,          // フィールドインデックスなし
               Index2 :=       0,          // フィールドインデックスなし
               CnvtToken :=    TRUE,       //10 バイトトークンに変換
               VarToken :=     GUDVAR.GUDVar1T,
               Error :=        F102.0,
               Done :=         F100.1,
               State :=        FW104,
               RD :=           P#DB99.DBX0.0 DINT 1
               );
```

■ FB 7: PI_SERV2　汎用 PI サービス

機能説明

FB 7 の詳しい説明については，FB 4 の説明を参照してください。FB 4 と違う点は，Wvar1 の数と，引続き起こるパラメータの数だけです。FB 7 では，Wvar1 から Wvar16 までを，VAR_INPUT で定義します（FB 4 は，Wvar 1 から Wvar 10 まで)。他のすべてのパラメータは，FB 4 に示されます。この PI サーバは，FB 4 で前もって実装されているすべての PI サービスに使用できます。

宣言

```
FUNCTION_BLOCK FB 7
Var_INPUT
    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;
    WVar11 :     WORD;
    WVar12 :     WORD;
    WVar13 :     WORD;
    WVar14 :     WORD;
    WVar15 :     WORD;
    WVar16 :     WORD ;
END_VAR

VAR OUTPUT
    Error :      BOOL;
    Done :       BOOL;
    State :      WORD;
END_VAR
```

正式パラメータの説明

以下は，PI_SERV 機能のすべての正式パラメータの表です。

信号	種類	タイプ	数値の範囲	メモ
Req	I	Bool		ジョブ要求
PIService	I	Any	[DBName].[VarName] デフォルトは :"PI".[VarName]	PI service description
Unit	I	Int	1 ...	エリア番号
Addr1 to Addr4	I	Any	[DBName].[VarName]	選択された PI サービスにしたがって指定された文字列を参照
WVar1 to WVar16	I	Word	1 ...	整数またはワード変数。選択された PI サービスにしたがって特定
Error	Q	Bool		ジョブは確認応答は否定，または実行されない
Done	Q	Bool		ジョブは問題なく実行
State	Q	Word		エラー識別子を見ること

追加の PI サービスの概要

以下の項目では，PLC からスタートすることのできる PI サービスの概要について述べています。変数 (Unit, Addr..., Wvar...) を入力した汎用 FB 7 の意味およびアプリケーションは，個々の PI サービスによって異なります。

PI サービス	機能
TMFPBP	空きロケーションを検索

空きロケーションを検索

TMFPBP

機能 : (パラメータ設定に依存する)

このサービスを使用して，特定された基準（ツールサイズおよびロケーションタイプ）を満たす空きロケーション用に，その適切なパラメータ内で命名されたマガジンを検索します。空きロケーションの検索結果は，サービスが正しく機能している場合に，TMC ブロックの変数 magCMCmd-Par1 (マガジン番号) および magCMCmdPar2 (ロケーション番号) から取出すことができます。この PI サービスは，変数 magCMCmdPar1 および magCMCmdPar2 に結果を保存するので，いくつかの制御ユニットあるいは PLC が 1 つの NC で運転されている場合は，機能番号 _N_TMFDPL のセマフォメカニズム (PI サービス MMCSEM) を使って保護してください。検索は，設定パラメータ MagazinNumber_From, LocNumber_From, MagazinNum-ber_To, LocNumber_To により，下記の方法であらかじめ定義することができます。

Magazin-Number_From	Loc_Num-ber_From	Magazine Number_To	Loc_Num-ber_To	検索エリア
WVar1	WVar2	WVar3	WVar4	
#M1	#P1	#M1	#P1	マガジン #M1 にあるロケーション #P1 のみをチェック
#M1	#P1	#M2	#P2	マガジン #M1 でスタートするロケーション，マガジン #M2 までのロケーション #P1，ロケーション #P2 を検索
#M1	-1	#M1	-1	マガジン #M1 にあるすべてのロケーション - これのみ - を検索
#M1	-1	-1	-1	マガジン #M1 でスタートするすべてのロケーションを検索
#M1	#P1	-1	-1	マガジン #M1 でスタートするすべてのロケーションおよびこのマガジンにあるロケーション #P1 を検索
#M1	#P1	#M1	-1	マガジン #M1 でスタートする マガジン #M1 にあるロケーション，およびこのマガジンにあるロケーション #P1 を検索
#M1	#P1	#M2	-1	マガジン #M1 でスタートするロケーション，およびこのマガジンまでにあってマガジン #M2 を含むロケーション #P1 を検索
#M1	-1	#M2	#P2	マガジン #M1 までにスタートしてマガジン #M2 を含むロケーション，およびこのマガジンにあるロケーション #P2 を検索
#M1	-1	#M2	-1	マガジン #M1 までにスタートしてマガジン #M2 を含むロケーションを検索
-1	-1	-1	-1	すべてのマガジンロケーションを検索

（注）この PI サービスの前後 , 関連パラメータ Wvar1 で，この PI サービス用に MMCSEM PI サービスを呼出してください。詳しくは，PI サービス MMCSEM を参照してください。

パラメータ化

表 1.68 パラメータ化

信号	タイプ	数値の範囲	意味
PIService	ANY	PI.TMFPBP	空きロケーションを検索
Unit	INT	1 ... max. TOA	TOA
WVar1	INT		MagazinNumber _From: 検索を始めるマガジンのマガジン番号 (検索エリアのスタート)
WVar2	INT	:	Loc_Number_From 検索を始めるマガジン MagazinNumber_From にあるロケーションのロケーション番号
WVar3	INT		MagazinNumber_To: 検索を終了するマガジンのマガジン番号
WVar4	INT		Logic_Number_To 検索を終了する MagazinNum-ber_To にあるロケーションのロケーション番号
WVar5	INT		Magazin Number_ref
WVar6	INT		LacaTion Number_ref
WVar7	INT	0, 1 .. 7	必要な左半分のロケーションの数
WVar8	INT	0, 1 .. 7	必要な右半分のロケーションの数
WVar9	INT	0, 1 .. 7	上方向へ要求する半分のロケーションの数
WVar10	INT	0, 1 .. 7	下方向へ要求する半分のロケーションの数
WVar11	INT	0, 1 .. 7	望みのロケーションタイプの数
WVar12	INT	0: デフォルト 1: 前方 2: 後方 3: 対称的	要求された検索方向を特定

■ FB 29: 信号レコーダとデータトリガ診断

機能

信号レコーダの説明

「診断」FB は，PLC ユーザプログラム上のさまざまな診断ルーチンを実行することができます。診断ルーチンログは，ステータスおよび信号切換えを信号で伝えます。8 までのブール信号 (パラメータ Signal_1 to Signal_8) は，信号の 1 つが切替わるごとに，リングバッファに記録されます。パラメータ Var1(バイト値)，Var2 および Var3(整数値) の最新の情報もリングバッファに保存されます。すでに終わった OB 1 周期の数も，追加の情報としてリングバッファに保存されます。この情報は，OB 1 周期グリッドにおいて，信号および値をグラフィカルに評価できるようにします。最初の「診断」FB を，新しい PLC 周期に呼出すので，パラメータ NewCycle を TRUE にセットしてください。「診断」FB を同じ OB 1 周期の中で何

度も呼出す場合は, 2 回目とそれに続く呼出しでは パラメータ NewCycle を FALSE にセットしてください。リングバッファはユーザご自身でセットアップしてください。リングバッファの DB は，パラメータ BufDB にある診断 FB へ送ってください。リングバッファは，ソースコードで指定されているように，配列構造を使用してください。この配列は，配列要素をいくつでも持つことができます。250 の配列要素での使用をおすすめします。パラメータ ClearBuf を使って，リングバッファをクリアして，BufAddr ポインタ (I/O パラメータ) をスタートにセットします。FB の関連インスタンス DB は，ユーザエリアからの DB です。

機能

データトリガの説明

データトリガ機能は，許可されたメモリ呼び出しにある，特定の値（ビットも含む）のトリガを可能にします。トリガされるセルは，パラメータ TestVal の診断で比較される前に，データブロック内で，ビットマスキング（パラメータ AndMask) を行って四捨五入します。

（注）機能のソースコードは，Diagnose.awl 名で，基本プログラムライブラリのソース内容において有効です。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において定義されます。機能呼出しについてもこの機能で説明しています。DB 番号および呼出しを，変更してください。

```
FUNCTION_BLOCK FB 29
VAR_INPUT
    Func : INT ; // 機能番号
//0 = 機能なし , 1 = 信号レコーダ , 2 = データトリガ
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 ;
    Datdr : POINTER ;                    // エリアポインタをテストするワードへ
    TestVal : WORD ;                     // トリガの値
    AndMask : WORD ;                     //AND- Mask をテストするワードへ
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

```
DATA_BLOCK DB 81
TITLE =
//FB 29 のリングバッファ DB
VERSION : 1.0
STRUCT
    Feld : ARRAY [0 .. 249 ] OF STRUCT / / この構造はどんなサイズでも可
    Cycle : INT ;                          // バッファ内の最後に入るストレージへの
                                             デルタ周期
        Signal_1 : BOOL ;                  //FB 29 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
```

正式パラメータの説明

以下は，診断機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
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 to Signal_8	I	Bool		切換えをチェックするビット信号
NewCycle	I	Bool		前述の説明を参照
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		リングバッファ DB およびリセットポインタ BufAddr を削除
BufAddr	I/Q	Int		読取りデータのターゲットエリア
機能 2 のパラメータ				
DataAdr	I	Pointer		ポインタをテストされるワードへ
TestVal	I	Word		比較値
AndMask	I	Word		説明を参照
TestIsTrue	Q	Bool		比較の結果

構成ステップ

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記録するのに適するように，データを定義します。診断 FB を呼出すために，適したポイント，またはユーザプログラム内のポイントを見つけます。リングバッファ用のデータブロックを作成してください（呼び出しの例を参照）。診断 FB を，パラメータでユーザプログラムに呼出します。機能 1 では，パラメータ ClearBuf でリングバッファをクリアすることをおすすめします。レコーディング段階が終了すると（機能 1），データビューにあるデータブロックを開いて，STEP 7 にあるリングバッファ DB を読取ります。この時，リングバッファ DB の内容を分析することができます（例えば，信号タイミングチャートにおいて）

呼出しの例

```
FUNCTION FC 99: VOID
TITLE =
VERSION : 0.0
BEGIN
NETWORK
TITLE = NETWORK
CALL FB 29 , DB 80 (
    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 100.4,
            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
```

■ FC 2: GP_HP 基本プログラム，周期的セクション

機能の説明

NCK-PLC インタフェースは，周期的モードで完全に処理されます。基本プログラムの実行時間を最短にするために，制御／ステータス信号のみを周期的に転送します。すなわち，NCK が要求した時，補助機能および G 機能の転送だけが行われます。

宣言

FUNCTION FC 2: VOID

// パラメータなし

呼出しの例

時系列の点から，基本プログラムは，ユーザプログラムの前に処理してください。したがって，OB 1 の最初に 基本プログラムを呼出してください。

下記の例は，OB 1 の標準宣言および基本プログラムの呼出し（FC2）について，機械操作パネル信号の転送（FC19），およびエラーと制御メッセージの取得（FC 10）について述べています。

```
ORGANIZATION_BLOCK OB 1
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 FC 2;                          // 最初の FC で基本プログラムを呼出す
    // ここにユーザプログラムを挿入
    CALL FC 19(                         // インタフェースへの MCP 信号
        ModeGroupNo := B#16#1,          // モードグループ番号 1
        ChanNo :=      B#16#1,          // チャンネル番号 1
        SpindleIFNo := B#16#4,          // スピンドルインタフェース番号 = 4
        FeedHold :=    f22.0,           // フィード停止信号モーダル
        SpindleHold := db2.dbx151.0);   // メッセージデータブロックのスピンドル
                                           停止モーダル
CALL FC 10(    // エラーおよび運転メッセージ
        ToUserIF:=          TRUE,       //DB2 からインタフェースへの転送信号
        Quit :=             I6.1);      //16.1 を介してのエラーメッセージの確認応
                                           答
END_ORGANIZATION_BLOCK
```

■ FC 3: GP_PRAL 基本プログラム，アラーム駆動セクション

機能の説明

NCK から PLC へのブロック同期転送データ（補助機能および G 機能）は，パートプログラムのアラーム駆動セクションで処理されます。補助機能は，ノーマルスピードおよび高速補助機能にさらに分割されます。NC ブロックの高速機能はバッファされ，確認応答が NC へ転送されます。これらは，つぎの OB1 周期のスタート時に，アプリケーションインタフェースへ送られます。ノーマル補助機能は，1 つの周期の間に存在していた時のみ確認されます。これにより，アプリケーションは NC へ読取りディスエーブルを発することができるようになります。

高速補助機能は，つぎつぎと順に指定され，ユーザプログラムを無駄にすることはありません。これは，基本プログラムの中でメカニズムによって確実になります。G 機能はすぐに評価されて，アプリケーションインタフェースへ送られます。

NC プロセス割込み

NC によって割込みが引起された場合（すべての IPO 周期において可能な），ビットは，基本プログラムの OB 40 のローカルデータ ("GP_IRFromNCK") にセットされます。(FB 1 パラメータ UserIR := TRUE の場合のみ)。このデータは，他のイベントにはセットされません（I/O によるプロセス割込み）。この情報は，必要なアクションを開始するために，ユーザプログラムにある関連割込みルーチンへ，分岐できるようにします。マシン用ユーザプログラムのジョブ制御処理及び，高速処理を可能にするため，以下の NC 機能が，PLC ユーザプログラムのために，割込み処理ルーチン（OB 40 プログラムセクション）の中に準備されています。

- 選択された補助機能
- ツール管理オプションのツール変更機能
- PLC を介して，位置決め軸，割出し軸およびスピンドルの起動に達する位置

上記に示した機能は，マシンで反応を開始するために，OB 40 にあるユーザプログラムによって評価することができ，または評価してください。例えば，旋盤で T 指令された時，回転切換えメカニズムを起動することができます。くわしくは，対応する PLC プログラミングマニュアルのプログラミングハードウェア割込み（時間遅延，割込み可能性など）に関する箇所をお読みください。

補助機能

高速，または「標準確認応答」の補助機能は通常，それらがどのように割当てられているかにかかわらず，割込み制御によって，あるいは割込み制御なしで処理されます。補助機能（T, H, DL）がユーザプログラムによって，割込み駆動ベース上において単独で実行されるように定義するために，FB 1 にある基本プログラムパラメータをセットすることができます。。

初期のバージョンでは，割込みを介して割当てられない機能は，周期的基本プログラムによってのみ有効となります。これらの機能の信号変化は，PLC 周期において有効です。

補助機能グループ (T, H, DL) の選択が，割込み制御で行われた場合，選択した機能の割込みは，ユーザプログラムによってのみ処理することができます。

ビットは，ユーザプログラムのために，チャンネルを特定してローカルデータ "GP_AuxFunction" にセットされます（"GP_AuxFunction[1]" をセットした場合，補助機能は，第 1 チャンネルに対して有効となります)。

信号の変化および機能値は，関連チャンネル DB においてユーザが使用できます。この割込み制御機能用の変換信号は，少なくとも OB 1 周期が完全に 1 回実行された後，周期的基本プログラムセクションにおいて，ゼロにリセットします（最大で，約 2 OB サイクル)。

ツールチェンジ

ツール管理機能について，マガジンへの重工具交換指令や，主軸の工具交換は，割込みによって供給されます。これは，OB 40 にあるローカルデータビット "GP_TM" をセットして行います。PLC ユーザプログラムは，ツールチェンジのためにツール管理 DB (DB 72 または DB 73) をこのようにチェックして，ツールチェンジ運転を開始します。

到達位置

OB 40 ローカルデータのビット構造 "GP_InPosition" は，マシンの軸別になっています。(それぞれのビットは，軸／スピンドルに対応しています。例えば，GP_InPosition[5] は，5 番目の軸に対応しています)。

機能が，FC 18 (スピンドル制御) を介して起動した場合，関連 "GP_InPosition" ビットによって，上記の FC リストにある "InPos" 信号の瞬時の評価を実行することができます。この機能を使って，例えば，割出し軸のクランプを即座に起動することができます。

宣言

FUNCTION FC 3: VOID

// パラメータなし

呼出しの例

時系列の点から，基本プログラムは，その他のアラーム駆動ユーザプログラムを実行する前に処理してください。したがって，OB 40 の最初に基本プログラムを呼出してください。下記の例は，OB 40 の標準宣言および基本プログラムの呼出しについて述べています。

```
ORGANIZATION_BLOCK OB 40
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;
// 基本プログラムに割当てられる
GP_IRFromNCK : BOOL;                // ユーザ用 NCK による割込み
GP_TM : BOOL;                       // ツール管理
GP_InPosition : ARRAY [1..3] OF BOOL; // 位置決め軸 , 割出し軸，スピンドル向け
                                    軸
GP_AuxFunction : ARRAY [1..10] OF BOOL; // 補助機能向けチャンネル
GP_FMBlock : ARRAY [1..10] OF BOOL; //FM への転送ブロック向けチャンネル
    // さらにローカルユーザデータは，このポイントから前への定義をする
END_VAR
BEGIN
    CALL FC 3;
    // ここでユーザプログラムを挿入してください
END_ORGANIZATION_BLOCK
```

■ FC 7: TM_REV リボルバでのツールチェンジ用転送ブロック

機能の説明

このブロックは，リボルバが変化した後にユーザが呼出します。この呼出しを行うには，リボルバ番号（DB 73 のインタフェース番号に対応）を，パラメータ "ChgdRevNo" で指定してください。このブロックを呼出す時，データブロック DB 73 にある関連 "Interface active" （インタフェースアクティブ）のビット，FC 7 のワード 0 を，パラメータ "Ready" := TRUE が返ってきた後で，リセットします。

ブロック FC TM_REV は，この転送に適切なインタフェース（DB 73, ワード 0）の起動信号が，ツール管理機能によって提供されている場合にのみ，スタートします (パラメータ "Start" = "TRUE" でスタート)。

出力パラメータ "Ready" は，ジョブが正しく実行されている時は，値 TRUE にセットされています。この時，ユーザは，パラメータ "Start" = "FALSE" をセットするか，再びブロックの呼出しはしないでください。パラメータ "Ready" を "FALSE" にセットして , パラメータ "Error" にあるエラーコードをインタープリタしてください。エラーコードが 0 の場合 , このジョブをつぎの PLC 周期の中で，繰返してください（例えば，"Start" は，"TRUE" にセットしたまま)。これは，転送ジョブがまだ完了していないことを意味しています（FC 7 呼出しおよびパルスダイアグラムの例を参照)。パラメータ "Start" は，引続き起こるジョブの信号エッジを必要としません。



警告

転送の中途終了は許可されていません（例えば，外部信号リセットによる)。パラメータ Ready および Error が <> になるまで，パラメータ Start は常に "1" 信号にセットしてください。

<> 0 のエラーコードは，まちがったパラメータ化であることを示しています。

（注）ツール管理に関するさらに詳しい説明（PLC に関することも）は，ツール管理の機能の説明を参照してください。さらに，ツール管理のための PI サービスは，FB 4, FC 8 および FC 22 を介して提供しています（この取扱説明書の対応するセクションも参照してください)。マシンデータ 20310 ビット 12 を，円形マガジン用に，1 にセットしないでください。

マニュアルリボルバ切換え

マニュアル動作でリボルバを回転させる場合，この情報をツール管理システムに送る必要があります。FC 8 の非同期転送機能を，リボルバの変更する位置への転送に使用してください。これは，シーケンスの中で最初のマニュアル回転をした時にのみ行ってください。この場合，FC 8 を介して，非同期転送の下記のパラメータ化が必要となります。

TaskIdent = 4
TaskIdentNo = チャンネル
NewToolMag = リボルバのマガジン番号
NewToolLoc = ツールのオリジナルロケーション
OldToolMag = バッファ（スピンドル）のマガジン番号 = 9998
OldToolLoc = スピンドルのバッファ番号
Status = 1

この方法によって，前の T コマンドが再びプログラムされる場合，同じ T コマンドを，ツール管理インタフェースで繰返すこともできます。

機能の宣言

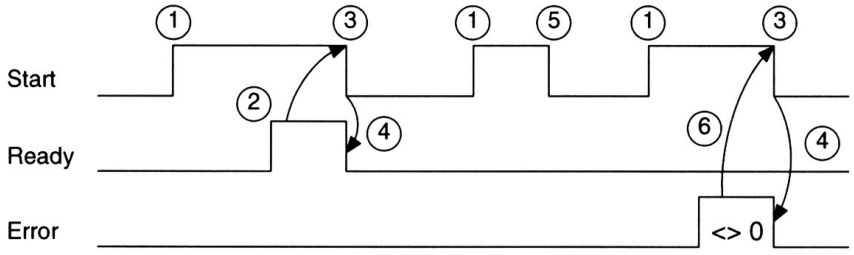
```
STL 表現
FUNCTION FC 7 : 無効
// 名前 :TM_REV
VAR_INPUT
    Start:          BOOL;
    ChgdRevNo:      BYTE;
END_VAR
VAR OUTPUT
    Ready:          BOOL;
    Error:          INT;
END_VAR
BEGIN
END_FUNCTION
```

正式パラメータの説明

以下は，TM_REV 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
Start	I	Bool		1 = 転送スタート
ChgdRevNo	I	Byte	1..	リボルバインタフェースの番号
Ready	Q	Bool		1 = 転送完了
Error	Q	Int	0..3	エラーチェックバック 0: エラーは発生していない 1: リボルバは存在しない 2: パラメータ "ChgdRevNo" にある不正なリボルバ番号 3: 不正なジョブ (選択されたリボルバの「インタフェースアクティブ」信号 = "FALSE")

タイミングダイアグラム



①機能が起動
②確認応答は肯定：ツール管理は転送されている
③確認応答が到着した時，機能起動信号をリセット
④ FC を使用して信号交換
⑤この信号チャートは，許可されていない。ツール管理からの新しいツール位置を NCK に転送しなければならないので，普通はジョブを終了すること。
⑥確認応答は否定 : エラーが発生，エラーコードは出力パラメータステータスへ

呼出しの例

```
CALL FC 7       // リボルバ用ツール管理転送ブロック
                Start :=            m 20.5, // スタート := "1 " => 転送の開始
                ChgdRevNo :=        DB61.DBB 1,
                Ready :=            f 20.6,
                Error :=            DB61.DBW 12);
a f 20.6;                           // 応答準備完了
r f 20.5;                           // リセットスタート
jc m001;                            // すべてが OK であればジャンプ
l db61.dbw 12;                      // エラー情報
ow w#16#0;                          // エラー評価
jn fehl;                            //<> 0 であれば，エラー取扱いへジャンプ
f001:                               // 残っているプログラムの実行を開始する
fehl:
r f 20.5                            // エラーが存在する場合，リセットスタート
```

■ FC 8: TM_TRANS ツール管理用転送ブロック

機能の説明

ツール位置を変える，あるいはステータスを変更する場合は，FC TM_TRANS を呼出します。この FC は，パラメータ "TaskIdent" でパラメータ化します。

1. ロード／アンロード位置用
2. スピンドル変更位置用
3. 転送識別子としてのリボルバ変更位置用
4. 非同期転送
5. ロケーション予約での非同期転送

インタフェース番号は，パラメータ "TaskIdentNo" で示されます。

ローディングポイント 5 の例 :

パラメータ "TaskIdent":=1 および "TaskIdentNo":=5).

さらに，現在のツール位置およびステータスデータ（この後に続くテキストにある，パラメータ "Status" のリスト）も，この転送機能で送られます。

（注）FC 8 は，古いツールの現在の位置を NCK に知らせます。
NCK は古いツールと新しいツールの位置が，変更になるまで
どこであったのかがわかります。

いわゆる「古いツール」（例えば，ロード中の）なしで転送する場合，値 "0" は，パラメータ "OldToolMag", "OldToolLoc" に割当てられます。.

ブロック FC TM_TRANS は，この転送に適切なインタフェース（ワード 0 の DB 71, DB 72, DB 73）の起動信号が，ツール管理機能によって供給された場合のみ，パラメータ "Start" = "TRUE" でスタートします。

ジョブが正しく実行されている場合，出力パラメータ "Ready" は，値 TRUE にセットされています。この時，ユーザは，パラメータ "Start" = "FALSE" をセットする，あるいは再びブロックの呼出しはしないでください。パラメータ "Ready" が FALSE の場合 , パラメータ "Error" にあるエラーコードをインタープリタしてください（FC 8 の呼出しの例，およびパルスダイアグラムを参照してください)。

エラーコードが 0 の場合 , このジョブをつぎの PLC 周期の中で，繰返してください（例えば，"Start" は，"TRUE" にセットしたまま)。これは，転送運転がまだ完了していないことを意味しています

パラメータ "Status" への値の割当てを 100 よりも少なくした場合，データブロック DB 7 または DB 72, DB 73 のワード 0 にある関連インタフェースは，停止します（プロセス終了)。インタフェースの適切なビットは，FC 8 によって 0 にセットします。

パラメータ "Start" は，引続き起こるジョブに対する信号エッジを必要としません。つまり，新しいパラメータは，"Ready = TRUE" を受けた時，即時に "Start = TRUE" で割当てることができます。

非同期転送

PLC からツール管理へ，ツールの位置変更を転送するために，"TaskIdent" := 4 または 5 でブロック FC TM_TRANS を呼出します (例えば，コマンドがアクティブの時，あるいは独立している位置変更が PLC によって制御されている時の電源割込み)。この呼出しは，ツール管理機能によるインタフェース起動は一切必要ありません。

パラメータ "TaskIdent" = 5 の場合 , ツール管理システムは，位置変更を加えるためにロケーションの確保をします。ただし，ツールが実際のマガジンからバッファに移送された場合のみ，ロケーションは確保されます。

関連 NC チャンネルは，パラメータ "TaskIdentNo" で構成してください。ツールの以前の位置は，パラメータ "OldToolMag", "Old-ToolLoc" で指定します。また，ツールの現在の位置は，パラメータ "NewTool-Mag", "NewToolLoc" で指定します。Status = 1 に指定してください。

指定されたツールは，ステータス 5 で，ロケーション "OldToolMag", "OldToolLoc" に残ります。このロケーションは，バッファとしてください（例えば，スピンドル）。実際のマガジンおよびロケーションを，パラメータ "NewToolMag", "NewToolLoc" で指定してください。これにより，ロケーションはバッファの位置になります。ツール管理システムで，特定のマガジンロケーションの位置を 伝える場合は，常にこの方法で行ってください。この方法は，サーチにおける位置合わせに使用します。



警告

転送の中途終了は許可されていません（例えば，外部信号リセットによる）。パラメータ Ready および Error が <> になるまで，Start パラメータは常に 信号 "1" にセットしてください。

<> 0 のエラーコードは，まちがったパラメータ化であることを示しています。

（注）ツール管理に関するさらに詳しい説明（PLC に関することも）は，ツール管理の機能の説明を参照してください。さらに，ツール管理のための PI サービスは，FB 4, FC 8 および FC 22 を介して提供しています（この取扱説明書の対応するセクションも参照してください）。

機能の宣言

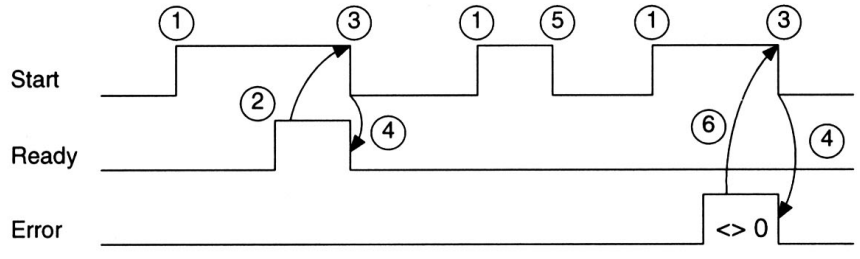
```
STL 表現
FUNCTION FC 8 : void
// 名前 :TM_TRANS
VAR_INPUT
   Start:          BOOL;
   TaskIdent:      BYTE;
   TaskIdentNo:    BYTE;
   NewToolMag:     INT;
   NewToolLoc:     INT;
   OldToolMag:     INT;
   OldToolLoc:     INT;
   Status:         INT;
END_VAR
VAR OUTPUT
   Ready:          BOOL;
   Error:          INT;
END_VAR
BEGIN
END_FUNCTION
```

正式パラメータの説明

以下は，TM_TRANS 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
Start	I	Bool		1 = 転送スタート
TaskIdent	I	Byte	1..5	インタフェースまたはタンク識別子 1: ロード／アンロード位置 2: スピンドル変更位置 3: リボルバ変更位置 4: 非同期転送 5: 確保された非同期転送
TaskIdentNo	I	Byte	1..	関連インタフェース番号またはチャンネル番号
NewToolMag	I	Int	0..	新しいツールの現在のマガジン番号
NewToolLoc	I	Int	0.. 最大　ロケーション番号	新しいツールの現在のロケーション番号
OldToolMag	I	Int	0..	新しいツールの現在のマガジン番号
OldToolLoc	I	Int	0.. 最大　ロケーション番号	新しいツールの現在のロケーション番号
Status	I	Int	1..5, 103..105	転送運転に関するステータス情報
Ready	Q	Bool		1 = 転送完了
Error	Q	Int	0..65535	エラーチェックバックその他の値 : 0: エラーは発生していない 1: "TaskIdent" 不明 2: "TaskIdentNo" 不明 3: 不正なタスク (選択されたリボルバの信号「インタフェースアクティブ」= "FALSE") その他の値 : 番号は，この転送によって起こる，NCK にあるツール管理機能のエラーメッセージに対応しています。

タイミングダイアグラム



①機能が起動
②確認応答は肯定：ツール管理は転送されている
③確認応答が到着した時，機能起動信号をリセット
④ FC を使用して信号交換
⑤この信号チャートは，許可されていない。ツール管理からの新しいツール位置を NCK に転送しなければならないので，普通はジョブを終了すること。
⑥確認応答は否定 : エラーが発生，エラーコードは出力パラメータステータスへ

ステータスリスト出力

Status（ステータス）= 1:

運転が完了 (ロード／アンロード／再ロード，変更準備，変更)。

FC 8 (FC TM_TRANS) のパラメータ "NewToolMag", "NewToolLoc", "OldToolMag" および "OldToolLoc" を，インタフェース内で指定されているツールターゲット位置にパラメータ化してください（"Prepare change" （変更準備）をのぞく)。くわしくは，FC TM_TRANS のパラメータの説明，あるいは PLC にあるツール管理の概要セクションを参照してください。

1. ロード／アンロード／再ロードの場合，要求したターゲットアドレスにツールは到着します。DB 71.DBX(n+0).3 "position at loading point" （ローディングポイントの位置付け）にあるインタフェース内のビットがイネーブルの場合，ステータス 1 は，機能終了に使用することはできません。ステータス 5 を使って正しい終了を行ってください。
2. 「変更準備」の場合，新しいツールが使用可能です。例えば，ツールオーバッファ (グリッパ) に位置付けします。また，古いツールのターゲット（マガジン，ロケーション）は，バッファで新しいツールの位置付けが行われた後，ツール変更位置へ移動する場合もあります。しかしながら，古いツールは，まだスピンドルに残っています。ツール変更の準備は，このように完了します。この確認応答の後，"Change" コマンドを受取ります。パラメータ "NewToolMag", "NewToolLoc", "OldToolMag" および "OldToolLoc" にある位置は，現在のツールの位置に対応しています。
3. 「変更」の場合（スピンドル，またはリボルバ)，インタフェースにアドレス指定されているツールは，要求されたターゲットアドレスへ到達します。ツール変更運転は，このように完了します。

Status (ステータス) = 2:

"new" （新しい）ツールは，使用可能になりません。このステータスは，"Change tool" （変更ツール）コマンドと共に使用することだけが認められています。このステータスを適用した場合，提案したツールの変更を PLC で行わないでください。提案した (新しい) ツールは，NCK にあるツール管理機能によって無効となります。新しいコマンドは，二重ツールのあるツール管理によって，出力されます。パラメータ "NewToolMag", "NewToolLoc", "OldToolMag", OldToolLoc" の位置は，ツールのオリジナル位置に対応しています。

Status (ステータス) = 3:

エラーが発生しています。ツール位置を変更しないでください。その間に発生するマガジン位置のいかなる変更も，例えば，FC TM_TRANS を介したステータス = 105 で，前もって通知してください。唯一ツール位置が，ツール管理機能によって考慮されます。

Status (ステータス) = 4:

マガジンの "Old Tool" は，パラメータ "OldTool Mag" と "OldToolLoc" に指定してください。このステータスは，ツール変更用の準備と共に使用することだけが認められています（スピンドルに変更)。このステータスが，NCK にあるツール管理機能へ転送された後，ツール管理は，引続き起こるコマンドにおいて指定されたマ

ガジン位置を考慮に入れるようにします。ただし，位置が自由の場合にのみ行うことができます。パラメータ "NewToolMag" および "NewToolLoc" は，考慮されません。

Status (ステータス) = 5:

運転は完了。「新しい」ツールは，パラメータ "NewToolMag", "NewToolLoc" において指定された位置にあります。この場合，指定されたツールは，実際にはこの位置にはありませんが，まだ同じマガジンロケーションにはあります。ただし，このマガジンロケーションは，パラメータにおいてセットされた位置へ移動しています（例えば，ツール管理位置)。このステータスは，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のみ使用します。ステータス 5 は，マガジンの現在の位置を調整するため，および引続き起こるコマンド用のサーチを向上するための，ツール管理機能をイネーブルにします。このステータスは，ロード，アンロードおよび再ロード運転と共に，およびツール変更用の準備と共に使用することだけが認められています。パラメータ "OldToolMag" および "OldToolLoc" オーバッファのデータでパラメータ化してください。

- ロード，再ロード :
  ロードまたは再ロードにおいて，ツール用ロケーションは NCK 内ですでに確保されています。マシンオペレータは，ターゲットロケーションにツールを挿入してください。
  注意 : ロケーションの確保は，制御システムのスイッチを再びオンにした時，キャンセルします。
- 変更用の準備 :
  まだ実行するツールの動きは，ツールを交換した後までは行われません。
- ロード位置での位置決め :
  DB 71.DBX (n+0).3 "position at loading point" （ロード位置での位置決め）にあるインタフェース内のビットがイネーブルの場合，ステータス 5 (ステータス 1 ではない) のみ機能の終了に使用できます。

Status (ステータス) = 103:

「新しい」ツールを挿入することができる。このステータスは，PLC が新しいツールを拒否した場合，ツール変更用の準備と共に使用することだけが認められています (MD:MC_TOOL_MANAGEMENT_MASK, Bit 4)。ツール位置は，変更しないままです。このステータスは，NCK において前処理が続いている事を確実にすることが必要です。(これ以外の場合，処理は停止します)。

参照： 1.19 工具補正（W1）

Status （ステータス）= 104:

「新しい」ツールは，パラメータ "NewToolMag", "NewToolLoc" で指定されている位置にある。このステータスは，ツールがまだ同じロケーション内のマガジンにある場合にのみ認められています。「古い」ツールは，パラメータ "OldToolMag", "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，新しいツールは 実際にはこの位置にはありませんが，まだ同じマガジンロケーションにはあります。しかしながら，このマガジンロケーションは，パラメータにおいてセットした位置へ移動します（例えば，ツール管理位置)。このステータスは，「ツール変更の準備」のために，リボルバ，チェーンタイプマガジンおよびディス

クマガジンにだけ使用します。ステータス 104 は，マガジンの現在の位置を調整するため，および引続き実行されるコマンド用のサーチを向上するためのツール管理機能をイネーブルにします。

Status（ステータス）= 105:

関与しているすべてのツールは，指定されたバッファに到達した（運転がまだ完了していない場合の標準的なケース)。ツールは，指定されたツール位置にあります（パラメータ "NewToolMag", "NewToolLoc", "OldToolMag", "OldTool-Loc"）。

ステータス定義

確認応答ステータスに関しては，ステータス情報番号 1 から 5 は，一般的にはコマンドの終了を導くものです。FC 8 がステータスの 1 つを受取った場合，FC 8 で指定されたインタフェースの "Interface active bit" （インタフェースアクティブビット）は，「0」にリセットされ（インタフェースリスト DB 1 から DB 73 を参照），このように運転を完了します。ステータス情報 103 から 105 の場合，応答が異なります。FC 8 がステータス情報のこれらの項目の 1 つを受取った場合，このインタフェースの「インタフェースアクティブビット」は，「1」のままです。PLC にあるユーザプログラムでは，これ以上の処理を要求されます（例えば，マガジン位置決めの続行)。ステータス情報のこの項目は，一般的に，運転がまだ進行している間，1 つあるいは両方のツールの位置変更を転送するために使用します。

呼出しの例

```
CALL FC 8       // ツール管理転送ブロック
                Start :=          m 20.5,    //Start( スタート ) := "1 " => 転送開始
                TaskIdent :=      DB61.DBB 0,
                TaskIdentNo :=    DB61.DBB 1,
                NewToolMag :=     DB61.DBW 2,    // 新しいツールの現在位置
                NewToolLoc :=     DB61.DBW 4,
                OldToolMag :=     DB61.DBW 6,    // 古いツールの現在位置
                OldToolLoc :=     DB61.DBW 8,
                Status :=         DB61.DBW 10,    // ステータス
                Ready :=          m 20.6,
                Error :=          DB61.DBW 12);
a f 20.6;                         // 質問準備完了
r f 20.5;                         // リセットスタート
jc m001;                          // すべてが OK の場合，ジャンプ
lDB61.dbw 12;                     // エラー情報
ow w#16#0;                        // エラー評価
jn fehl;                          // エラー処理へジャンプ
m001:                             // 標準ブランチ
fehl:                             // エラー処理
r f 20.5:                         // リセットスタート
```

■ FC 9: ASUP 非同期サブプログラムの起動

機能の説明

FC ASUP は，NC 内の選択された機能を起動することができます。ASUP を PLC からスタートする前に，NC プログラムまたは FB 4 (PI サービス ASUP) で選択して，パラメータを設定してください。このように準備ができると，ASUP は PLC からいつでもスタートすることができます。このチャンネル上を走っている NC プログラムは，非同期サブプログラムによって中断します。その時同じチャンネルにある 1 つの ASUP だけがスタートできます。スタートパラメータを，1 つの PLC 周期の中で 2 つの FC 9 を論理「1」にセットする場合，ASUP は，呼出されたシーケンスの中でスタートします。

エラーが発生して，ASUP を終了 (Done) する時は，スタートパラメータを論理的「0」にセットしてください。

ジョブの処理を行うために，すべての FC ASUP は，グローバルユーザメモリエリアからの , それぞれの WORD パラメータ（Ref）が必要となります。このパラメータは，内部でのみ使用するもので，ユーザによる変更はしないでください。パラメータ Ref は，最初の OB1 周期において値 0 で初期化します。このため，すべての FC 9 は完全に呼出してください。または，スタートアップ時に，ユーザが値 0 でパラメータ Ref を初期化することができます。条件付き呼出しは，パラメータ Start=1, で起動します。これは，パラメータ Done が 1 から 10 へ移行するまでアクティブのままです。

（注）機能は，Low-High エッジでのみ開始します。

アラーム

400902	FC 9 にあるパラメータ ChanNo は許可されない
説明	パラメータ化されたチャンネルは存在しない
反応	アラーム表示および PLC STOP
処置	パラメータを正しくセット
継続	コールド再スタート後

宣言

```
FUNCTION FC 9: VOID                    //ASUP
VAR_INPUT
    Start:        BOOL;
    ChanNo:       INT;
    IntNo:         INT;
END_VAR
```

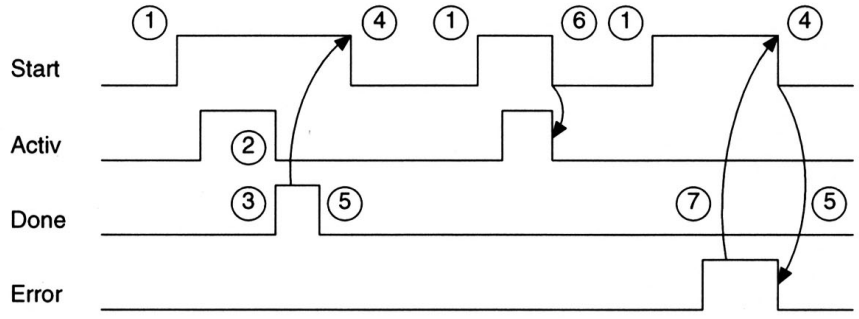
```
VAR OUTPUT
    Active:     BOOL;
    Done:       BOOL;
    Error:      BOOL;
    StartErr:   BOOL;
END_VAR
VAR_IN_OUT
    Ref:        WORD;
END_VAR
```

正式パラメータの説明

以下は，ASUP 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
Start	I	Bool		
ChanNo	I	Int	1 から 10	NC チャンネルの番号
IntNo	I	Int	1 - 8	割込み番号
Activ	Q	Bool		1 = アクティブ
Done	Q	Bool		1 = ASUP 終了
Error	Q	Bool		
StartErr	Q	Bool		1 = 割込み番号は割当てられない
Ref	I/Q	Word	グローバル変数 (MW, DBW,..)	FC 9 ごとに 1 ワード (内部使用)

タイミングダイアグラム



①機能が起動
② ASUP アクティブ
③確認応答は肯定：ASUP 終了
④確認応答受信後，機能の起動をリセット
⑤ FC による信号交換
⑥確認応答受信以前に機能開始をリセットした場合，出力信号はアップデートされず，影響を受けているアクティブな機能の運転シーケンスのみアップデートされる。
⑦確認応答は否定 : エラーが発生

呼び出しの例

```
CALL FC9(    // チャンネル 1 割込みにおける非同期サブプログラムがスタート
     Start :=        E45.7,
     ChanNo:=        1,
     IntNo:=         1,
     Active:=        F204.0,
     Done:=          F204.1,
     Error:=         F204.4,
     StartErr:=      F204.5,
     Ref:=           FW 200);
```

■ FC 10: AL_MSG エラーおよび運転メッセージ

機能の説明

FC AL_MSG を使用して，DB 2 に保存されている信号を評価し，運転メッセージと，発生／消滅したエラーとしてそれらの信号を MMC 上に表示します。立ち上がり信号（正のエッジ）は，エラーの場合も運転メッセージの場合でも即座に表示されます。立ち上がり信号（負のエッジ）は，運転メッセージの場合はただちにキャンセルされます。たとえ信号が存在しなくとも，「確認応答」パラメータが発生するまで，つまりユーザがメッセージを確認するまで，エラーメッセージは MMC に保存されたままです。パラメータ "ToUserIF" は，フィード，読取りおよび NC スタート停止の信号と，現在の軸，スピンドル，チャンネルインタフェースに対するフィード停止信号用のグループ信号を転送するために使用します。グループ信号は，アラーム確認応答のいかんを問わず，DB 2 にあるステータス情報から直接ユーザインタフェースへ転送されます。

1. パラメータ "ToUserIF" が FALSE にセットされている場合，信号はユーザインタフェースへ転送されません。この場合，ユーザの PLC プログラムにおいて，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は手を打ってください。
2. パラメータ "ToUserIF" が TRUE にセットされている場合，上記にリストされているすべての信号は，グループ信号としてそれぞれのケースでユーザインタフェースへ送られます。その結果，ユーザ PLC プログラムは，メッセージまたはアラーム出力と接続している DB 2 を介して，これらの信号に影響を与えます。適切な情報は，ユーザインタフェースにおいて上書きします。

上記 2 で説明した手順の代替として，FC AL_MSG が呼出された後，インタフェース信号へのディセーブルあるいは停止信号ステータスを適用して，メッセージ出力なしでディセーブルあるいは停止信号に，ユーザが影響を与えることができます。以下のプログラムは，この方法を説明しています。

```
CALL FC 10(
                ToUserIF:=          TRUE,
                Quit:=              e 6.1);
a f 50.0;       // チャンネル 1 へのフィードディセーブル
to db 21;
s DBX 6.0       // ディセーブルコンディションにセット,
                FC AL_MSG if F50.0 signal（信号）= "0" を介してリセット
```

エラーおよび運転メッセージは，データブロック DB 2 においてユーザで提供してください (1.15.5「■ DB2 のメッセージ信号」の DB2 の説明を参照).

機能の宣言

```
STL 表現
FUNCTION FC 10: Void
// 名前 :          AL_MSG
VAR_INPUT
    ToUserIF :    BOOL;
    Quit :        BOOL;
END_VAR
END_FUNCTION
```

正式パラメータの説明

以下は，AL_MSG 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
ToUserIF	I	Bool		1 = 周期ごとのユーザインタフェースへの転送信号
Quit	I	Bool		1 = エラーメッセージの確認応答

呼出しの例

```
CALL FC 10(                    // エラーおよび運転メッセージ
    ToUserIF:=  TRUE,          //DB2 からの信号は,
                               インタフェースへ転送される
    Quit :=     I6.1);         // エラーメッセージは
                               入力 16.1 を介して確認応答する
```

■ FC 12: AUXFU 補助機能でのユーザ呼出しインタフェース

機能の説明

入力パラメータに転送されたチャンネルが，新しい補助機能を持つ場合，一般的に FC AUXFU は，基本プログラムにあるイベント駆動ベースに呼出します。PLC ユーザは，補助機能がチャンネル DB の周期的ポーリングを避けて処理するように，プログラムインストラクションで，FC AUXFU を拡張することができます。このメカニズムで，ジョブ制御の基盤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きブロックとして提供されます。この場合，基本プログラムはチャンネル番号と共にパラメータ "Chan" を提供します。PLC ユーザは，新しい機能が有効であるのはどのチャンネルなのかわかります。そのチャンネルにおいて，新しい補助機能を，補助機能変更信号によって確定することができます。

宣言

```
FUNCTION FC 12: VOID                    // 補助機能のイベント制御
VAR_INPUT
   Chan:        BYTE;
END_VAR
BEGIN
   BE;
END_FUNCTION
```

正式パラメータの説明

以下は，AUXFU 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
Chan	I	バイト	1 から 10	NC チャンネルの番号

例

```
FUNCTION FC 12: VOID                // 補助機能のイベント制御
VAR_INPUT
   Chan:            バイト;     // 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;             // チャンネル番号 (1,2,..)
+ 20;               // チャンネル DB オフセット
T ChanDB;           // チャンネル DB 番号保存
AUF DB[ChanDB];     // チャンネル DB は間接的にオープン
                    // 補助機能の変更信号を走査している，など
BE;
END_FUNCTION
```

■ FC 13: BHGDisp ハンドヘルドユニット用ディスプレイ制御

機能の説明

このブロックは，ハンドヘルドユニット（HHU）にあるディスプレイを制御します。ディスプレイに出力される情報は，文字列データ ChrArray に 32 文字で保存されます。したがって，データブロックを作成した時に，この文字列用に，32 文字で固定された文字の割当てが必要となります。オプションの数値コンバータを使い，パラメータ "Convert" を TRUE にセットして，この文字列にある変数コンポーネントを挿入することができます。表示される変数は，ポインタ Addr を介して参照します。パラメータ データタイプ には，このパラメータのフォーマット説明があります（パラメータの表を参照)。変数のバイトの数は，フォーマット説明にリンクしています。文字列の中で右揃えしたアドレスは，パラメータ StringAddr によって特定します。書込まれた文字の数は，パラメータの表で示されます。パラメータ Row を "0" にセットすると，表示はされなくなります。(例えば，1 つまたはいくつかの PLC 周期において，いくつかの変数を表示出力なしに文字列に入力する場合)。

信号

バイト 1 および入力する文字を，ハンドヘルドユニットの出力信号によってブロックを通して接続します。これらは PLC ユーザプログラムでの書込みはできません。

追加のパラメータ

ハンドヘルドユニットの入出力データ用ポインタパラメータを，OB 100 のスタートアップ時に FB 1, DB 7 においてパラメータ化してください。パラメータ BHGIn は，ハンドヘルドユニット（PLC によって受信したデータ）からの PLC の入力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（PLC によって転送されたデータ）への PLC の出力データに対応しています。これらの 2 つのポインタは，SDB2100 において，MPI リンクで同じくパラメータに設定された関連データエリアのスターティングポイントにセットしてください。

（注）情報をディスプレイするために数値コンバータを使用する場合，PLC 周期の時間を減らす目的で，すべての PLC 周期において変換を実行するのは避けた方が良いでしょう。この場合，パラメータ "Convert" 用の HHU から PLC への入力信号 "Acknowledgment digital display" （確認応答デジタルディスプレイ）(DB m+5.7) を使用することをおすすめします。この方法で，最近のほとんどの数値情報をディスプレイできることになります。

機能の宣言

```
STL representation
DATA_BLOCK "strdat"                    // データブロック番号は
                                       記号ファイルで定義される
   STRUCT
   disp:          STRING [32]:= 'Line 1 Line 2 '; //32 文字が定義される
   END_STRUCT;
BEGIN
END_DATA_BLOCK
FUNCTION FC 13: VOID
VAR INPUT
   Row :          Byte ;          // 行ディスプレイ ( 表を参照 )
   ChrArray :     STRING ;        // 少なくとも文字列 [32] に転送
   Convert :      BOOL ;          // 数値変換起動
   Addr :         POINTER;        // 変換される変数へのポイント
   Data Type :    Byte ;          // 変数データタイプ
   StringAddr :   INT;            / 右揃え文字列アドレス (1...32)
   Digits :       BYTE ;          / 十進の場所の番号 (1...3)
END VAR
VAR OUTPUT
   Error :        BOOL ; /        / 変換または文字列エラー
END VAR
```

正式パラメータの説明

以下は，BHGDisp 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
Row	I	バイト	0 ~ 3	行ディスプレイ 0: ディスプレイ出力なし 1: 行 1 2: 行 2 3; 行 1 および 行 2
ChrArray	I	String	>= 文字列 [32]	この文字列は，すべてのディスプレイ内容を含む
Convert	I	Bool		数値変換の起動
Addr	I	Pointer		変換される変数のポイント
DataType	I	バイト	1 ~ 8	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 (パラメータ Digits を参照)
StringAddr	I	Int	1 ~ 32	変数 ChrArray にあるアドレス
Digits	I	バイト	1 ~ 4	以下の符号のあるデータタイプ Real にのみ適切 1: 符号なしの 6.1 デジット 2: 符号なしの 5.2 デジット 3: 符号なしの 4.3 デジット 4: 符号なしの 3.4 デジット
Error	Q	Bool		変化エラー，数値オーバフロー，または StringAddr におけるエラー

数値の範囲

表 1.69 データタイプの数値の範囲

データタイプ	数値で表すことのできる範囲
BOOL	0, 1
BYTE	0 から 255
WORD	0 から 65535
INT	- 32768 から + 32767
DWORD	0 から 9999999
DINT	- 9999999 から + 9999999
REAL (Digits := 1)	- 999999.9 から + 999999.9
REAL (Digits := 2)	- 99999.99 から + 99999.99
REAL (Digits := 3)	- 9999.999 から + 9999.999
REAL (Digits := 4)	- 999.9999 から + 999.9999

呼出しの例

```
CALL FC 13 (
    Row :=          MB 26,
    ChrArray :=     "strdat".disp,      // 記号表にある strdat 名の DB,
                                        データ要素のディップは,
                                        文字列 [32] として宣言,
                                        および文字によって完全に割当てられる
    Convert :=      F 90.1,
    Addr :=         P#M 20.0,           // 変換される番号
    Data Type :=    FB 28,              // 変数のデータタイプ
    StringAddr :=   FW 30,
    Digits :=       B#16#3,             // 十進ポイント後の 3 つの場所
    Error :=        F 90.2);
```

■ FC 17: Y/Delta スター／デルタ切換え

機能の説明

スター／デルタ切換えブロックは，定義された切換えロジック（時間制御された）を両方向に履行します。このブロックは，デジタルのメインスピンドル駆動にのみ使用するものなので，それぞれのスピンドルに別々に呼出してください。

切換え運転は，4 段階あるシーケンスで，2 つの別々のコンタクタを介して実行されます。

ステップ 1: 関連軸 DB (DB 31, ... DBX21.5) にあるインタフェース信号 "Motor selection in progress" （進行中のモータ選択），およびモータ選択 A A（DB31, ... DBX21.3）を介した切換えプロセスのシグナリングを削除。

ステップ 2: チェックバック信号 "Pulses enabled" (パルスイネーブル) = 0 (DB31, ... DBX93.7), およびドライブからアナウンスされるモータ選択の確認応答があるとすぐに，現在の通電されているコンタクタのスイッチが切れます。

ステップ 3: その他のコンタクタは，ユーザパラメータ "TimeVal" を使用してセットした時間が経過した後，通電されます。

ステップ 4: さらに遅れた後，切換えはドライブへ信号で送られます (DB31, ... DBX21.5)。

アラーム

401702	FC 17 にあるパラメータ ChanNo は認められない
説明	パラメータに設定されたスピンドルは存在しない
反応	アラーム表示および PLC STOP
処置	パラメータを正しくセット
継続	コールド再スタート後

エラーメッセージ

パラメータ "SpindleIFNo" の値が許容範囲内でない場合，PLC は，アラームメッセージ番号 401702 の出力にしたがって停止します。

特殊機能

パラメータ "TimeVal" が "0" にセットされている場合，100 ms の初期設定を使用します。値が 50 ms よりも少ない場合は , 50 ms の最小設定となります。

ブロックを無条件で呼出してください。

機能の宣言

```
STL 表現
    VAR_INPUT
    YDelta:         BOOL ;      // スター = 0, デルタ = 1
    SpindleIFNo:    INT;        // マシン軸番号
    TimeVal:        S5TIME;     // タイム値
    TimerNo:        INT;        // 切換え時間用ユーザタイマ
END_VAR
VAR OUTPUT
    Y:              BOOL ;      // スターコンタクタ
    Delta:          BOOL ;      // デルタコンタクタ
END_VAR
VAR_IN_OUT
    Ref:            WORD;       // ブロックステータスワード ( インスタンス )
END_VAR
```

正式パラメータの説明

以下は，YDelta 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
YDelta	I	Bool		0 = スター 1 = デルタ信号の切換えエッジが，切換え運転を開始する
SpindleIFNo	I	Int	1..	スピンドルインタフェースの番号
TimeVal	I	S5time	0..	切換え時間
TimerNo	I	Int	10..	プログラミング待ち時間用タイマ
Y	Q	Bool		スターコンタクタを通電させる
Delta	Q	Bool		デルタコンタクタを通電させる
Ref	I/Q	Word		ステータス情報用インスタンス 内部使用

呼出しの例

```
CALL FC 17 (
            YDelta :=       i 45.7,     // スター / デルタ
            SpindleIFNo:=   4,
            TimeVal:=       S5T#150ms,
            TimerNo :=      10,         // タイマ 10
            Y:=             q 52.3,     // スターコンタクタ
            Delta:=         q 52.4,     // デルタコンタクタ
            Ref:=           mw 50);     // インスタンス
```

■ FC 18: SpinCtrl スピンドル制御

機能の説明

FC SpinCtrl を使用して，スピンドルおよび軸を PLC から制御することができます。

参照： 1.17 主軸機能（S1）
2.10 補助位置決め軸（P2）
2.15 割出し軸（T1）

このブロックには，以下の機能があります。

- スピンドル位置
- スピンドル回転
- スピンドル振動
- 割出し軸
- 位置決め軸

それぞれの機能は，適切な開始信号（スタート，ストップ）の正のエッジによって起動します。この信号は，機能が InPos="1" によって正で，あるいは Error = "1" によって負で確認応答されるまで，論理「1」ステータスにとどまります。出力パラメータは，関連トリガ信号がリセットして，機能が終了した時に削除されます。

PLC を介して軸およびスピンドルの制御を可能にするには，PLC を起動してください。このためには，例えば，パラメータ "Start" または "Stop"　を起動して，FC "SpinCtrl" を呼出します。この場合，FC "SpinCtrl" は , NC からのスピンドル／軸の制御を要求します。

NC は，関連スピンドル／軸インタフェース (DB 31,...)（インタフェース信号のリストを参照）内のバイト 68 にあるこのスピンドル／軸のステータスをフィードバックします。軸／スピンドルを PLC 制御のもとで運転すると，アクティブ状態の移動コマンドを，適切な軸インタフェースを介して評価することができます。

完了すると ("InPos" が True, "Start" はゼロになる)，軸／スピンドルのチェック機能は，FC "SpinCtrl" によってニュートラルの状態に切り換ります。

そうでない場合，PLC ユーザプログラムから，呼出し FC "SpinCtrl" の前の PLC にチェックが要求されます。

この機能を続けて何回か呼出すと，FC におけるスイッチオーバのプロセスが省かれるので，スピンドル／軸からのより良い応答反応が得られます。

PLC ユーザプログラムを通しての起動は，バイト 8 にある対応するスピンドルインタフェース内で実行されます。

チェックを返した後，スピンドルは再び NC プログラムによってプログラミングすることができます。

（注）"InPos" 信号，あるいはエラーの場合は "Error" 信号が，エッジを 1 から 0 へ遷移するまで，FC 18 を周期的に呼出してください。"InPos"/"Error" 信号が 0 の状態の時にだけ，その軸／スピンドルを再びスタートまたは停止できます（つぎのスタートまたは停止は，少なくとも PLC 周期 1 回分遅らせてからにしてください）。これは，軸のインタフェースにあるデータバイト 8 の割当てを変更する時にも当てはまります。
パラメータ "Start" または "Stop" を使用して機能を中断することはできませんが，軸のインタフェース信号を使用してのみ中断することができます（例えば，distance-to-go 削除）。軸のインタフェースは，評価する必要のある軸のステータス信号も戻します（例えば，イグザクトストップ，移動コマンド）。
パラメータ "InPos" の意味は，機能「スピンドル回転」および「スピンドル振動」に対して以下のように定義されます。
セットスピードが出力される → エラーなしで機能がスタート
要求されたスピンドルスピードまでのスピンドル加速を，スピンドルインタフェースを介して評価してください。
いくつかの軸を同時に，あるいは時間をずらしたシーケンスで，ブロック FC 15, 16, 18 によって動作させることができます。上限は，軸の最大番号によって定義されます。NCK は，それぞれの軸／スピンドルの個々のインタフェースを介して，PLC 機能要求 (FC 15, 16, 18) を処理します。



警告

いくつかのブロックの呼出し (FC 15, FC 16, FC 18) を，PLC ユーザプログラムにおいて同じ軸／スピンドル用にプログラムした場合，その機能をユーザプログラムにおいて条件付呼出しによってインターロックさせてください。スタートしたブロックの条件付呼出し（パラメータ Start or Stop = TRUE) を，出力パラメータ "Active" または "InPos" が "1" から "0" へ変わるまで周期的に呼出してください。

アラーム

401802	FC 18 にあるパラメータ AxisNo は不正
説明	パラメータに設定された軸は存在しない
反応	アラーム表示および PLC STOP
処置	パラメータを正しくセット
継続	コールド再スタート後

機能

1. スピンドル位置 :

以下の信号が関連しています :

Start: 信号開始
Funct: "1" = スピンドル位置
Mode: 位置決めモード 1, 2, 3, 4
AxisNo: マシン軸の番号
Pos: 位置
FRate: 位置決め速度。FRate = 0 の時 , MD 35300 からの値 :
SPIND_POSCTRL_VELO (位置制御起動スピード) を使用
InPos: 位置が "Exact stop fine" に到達した時，"1" にセット
Error: 位置決めエラー = "1"
State: エラーコード

2. スピンドル回転 :

以下の信号が関連しています :

Start: 回転スタートの開始信号
Stop: 回転停止の開始信号
Funct: "2" = スピンドル回転
Mode: 位置決めモード 5 (回転方向 M4)
モード位置決め <>5 (回転方向 M3)
AxisNo: マシン軸の番号
FRate: スピンドルスピード
InPos: 機能はエラーなしでスタート
Error: 位置決めエラー = "1"
State: エラーコード

3. スピンドル振動 :

以下の信号が関連しています :

Start: 振動スタートの開始信号
Stop: 振動停止の開始信号
Funct: "3" = スピンドル振動
AxisNo: マシン軸の番号
Pos: ギア段階セット
InPos: Error: 位置決めエラー = "1"
State: エラーコード

振動スピードは，マシンデータ SPIND_OSCILL_DES_VELO からのものです。

MD 35010: GEAR_STEP_CHANGE_ イネーブル = 0	機能	MD 35010: GEAR_STEP_CHANGE_ イネーブル = 1	機能
Pos = 0	Oscill.	Pos = 0	
Pos = 1	Oscill.	Pos = 1	ギア段階変更 M41 での振動
Pos = 2	Oscill.	Pos = 2	ギア段階変更 M42 での振動
Pos = 3	Oscill.	Pos = 3	ギア段階変更 M43 での振動
Pos = 4	Oscill.	Pos = 4	ギア段階変更 M44 での振動
Pos = 5	Oscill.	Pos = 5	ギア段階変更 M45 での振動

4. 割出し軸移動 :

以下の信号が関連しています :

Start:	信号開始
Funct:	"4" = 割出し軸
Mode:	位置決めモード 0, 1, 2, 3, 4
AxisNo:	マシン軸の番号
Pos:	割出し位置
FRate:	位置決め速度。FRate = 0 の時 , マシンデータ POS_AX_VELO からの値 (ユニットをマシンデータ内にセット) を使用します。
InPos:	位置に到達した時，"Exact stop fine" で , "1" にセット
Error:	位置決めエラー = "1"
State:	エラーコード

5. 軸の位置 :

以下の信号が関連しています :

Start:	信号開始
Funct:	"5 to 8" = 軸位置
Mode:	位置決めモード 0, 1, 2, 3, 4
AxisNo:	マシン軸の番号
Pos:	位置
FRate:	位置決め速度。FRate = 0 の時 , マシンデータ POS_AX_VELO からの値 (ユニットをマシンデータ内にセット) を使用します。
InPos:	位置に到達した時，"Exact stop fine" で , "1" にセット
Error:	位置決めエラー = "1"
State:	エラーコード

宣言

```
FUNCTION FC 18: VOID                    //SpinCtrl
VAR_INPUT
    Start:      BOOL;
    Stop:       BOOL;
    Funct:      BYTE;
    Mode:       BYTE;
    AxisNo:     INT;
    Pos:        REAL
    FRate:      REAL
END_VAR
VAR OUTPUT
    InPos:      BOOL;
    Error:      BOOL;
    State:      BYTE;
END_VAR
```

正式パラメータの説明

以下は，SpinCtrl 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
Start	I	Bool		PLC からのスピンドル制御スタート
Stop	I	Bool		PLC からのスピンドル制御ストップ
Funct	I	バイト	1 から 8	1: スピンドル位置 2: スピンドル回転 3: スピンドル振動 4: 割出し軸 5: PosAxis メートル 6: PosAxis インチ 7: ハンドルオーバライドでの PosAxis メートル 8: ハンドルオーバライドでの PosAxis インチ
Mode	I	バイト	0 から 5	0: 絶対位置への位置決め 1: インクリメントに位置決め 2: 位置決め最短パス 3: 絶対位置 , 正のアプローチ方向 4: 絶対位置 , 負のアプローチ方向 5: M4 に関する回転方向
AxisNo	I	Int	1 - 31	移動する軸／スピンドルの番号
Pos	I	Real	± 0.1469368 I -38 から ± 0.1701412 I +39	回転軸 : 度割出し軸 : 割出し位置直線軸 : mm または inch
FRate	I	Real	± 0.1469368 I -38 から ± 0.1701412 I +39	回転軸およびスピンドル : r/min 下記の Frate についての情報を載せた表を参照
InPos	Q	Bool		1 = 到達した位置または実行された機能
Error	Q	Bool		1 = エラー
State	Q	バイト	0 から 255	エラーコード

FRate

SW 4 以降では，フィード速度を下記のように FC 18 において指定することができます。

1. 切削速度の単位は m/min または ft/min
2. 一定の研削ホイールの表面スピードは m/s または ft/s

これらの選択できる速度設定は，この機能が NC によって起動している時のみ使用することができます。エラーなく起動した事を確かめるフィードバックメッセージは，チャンネルインタフェースにあるバイト 84 において見つけることができます。

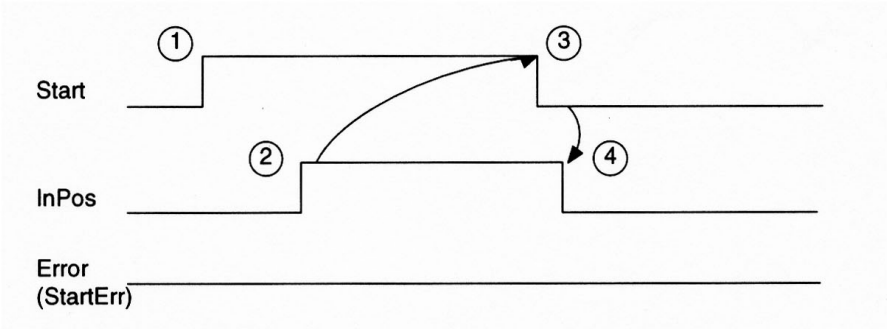
エラー識別子

ジョブの実行が失敗すると，ステータスパラメータ Error の "logical 1" で示されます。エラーの原因は，"State" を出力するブロックでコード化されます。

表 1.70 エラー識別子

ステータス	意味
PLC 処理によるエラー	
1	いくつかの軸／スピンドル機能が同時に起動した
20	機能がその位置に到達せずにスタートした
30	軸／スピンドルがまだ動作中に NC へ転送された
40	軸が NC プログラムでプログラムされた
NCK 処理によるエラー。アラーム番号は，840D 用診断ガイドに記述してある。	
100	アラーム番号 16830 に対応
105	アラーム番号 16770 に対応
106	アラーム番号 22052 に対応
107	アラーム番号 22051 に対応
108	アラーム番号 22050 に対応
109	アラーム番号 22051 に対応
110	速度／スピードが負
111	指令速度がゼロ
112	無効なギア段階
115	プログラムされた位置は到達していない
120	割出し軸ではない
121	割出し位置エラー
125	DC (1 番短いパス) は不可
126	絶対マイナス値は不可
127	絶対プラス値は不可
130	ソフトウェアリミットスイッチはプラス
131	ソフトウェアリミットスイッチはマイナス
132	作業エリア制限プラス
133	作業エリア制限マイナス
134	
135	
136	
システムまたは他の重大なアラーム	
200	システムアラーム番号 450007 に対応

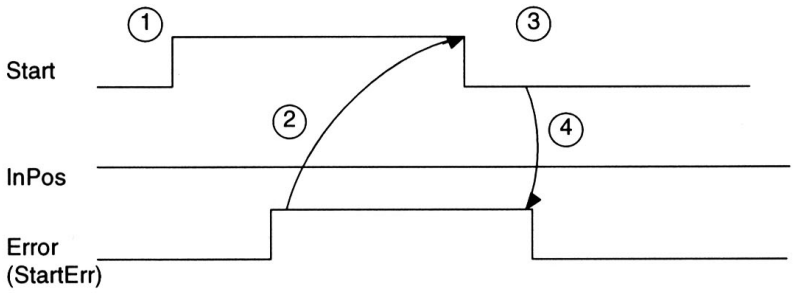
タイミングダイアグラム



① スタートまたはストップを通して正のエッジにより機能起動
② 確認応答は肯定 : 実行された機能 / 到達した位置
③ 確認応答受信の時，機能起動信号をリセット
④ FC による信号交換

タイミングダイアグラム

(エラー状態)



① スタートまたはストップを通して正のエッジにより機能起動
② 確認応答は否定：エラーが発生
③ 確認応答受信の時，機能起動信号をリセット
④ FC による信号交換

呼出しの例

1. スピンドル位置

```
// 確認応答は肯定でスタートリセット :
A M112.0;       // InPos
R M 100.0;      // スタート
// 確認応答は否定 , エラー評価後 ( ステータス : MB114)
A M113.0;       // エラー
A I 6.4;        // キー T12
R M 100.0;      // スタート
//T13 でスタート
A I 6.3;        // キー T13
AN F 112.0;     // InPos または Error = 0 の時のみ，新しいスタート
AN F 113.0;
S M 100..0;
    CALL FC 18 (
                Start :=        F100.0,
                Stop :=         FALSE,
                Funct :=        B_16_1,     // スピンドル位置
                Mode :=         B_16_2,     //1 番短いパス
                AxisNo :=       5,
                Pos :=          MD104,
                FRate :=        MD108,
                InPos :=        F112.0,
                Error :=        F113.0,
                State :=        FY114);
```

2. スピンドル回転スタート :

```
    CALL FC 18 (
                Start :=        F100.0,
                Stop :=         FALSE,
                Funct :=        B_16_2,     // スピンドル回転
                Mode :=         B_16_5,     //M4 用としての回転方向
                AxisNo :=       5,
                Pos :=          0.0,
                FRate :=        MD108,
                InPos :=        F112.0,
                Error :=        F113.0,
                State :=        FY114);
```

3. スピンドル振動スタート：

```
CALL FC 18 (
        Start :=    F100.0,
        Stop :=     FALSE,
        Funct :=    B#16#3,     // スピンドルを振動
        Mode :=     B#16#0,
        AxisNo :=   5,
        Pos :=      0.0,
        FRate :=    MD108,
        InPos :=    F112.0,
        Error :=    F113.0,
        State :=    FY114);
```

4. 割出し軸の移動

```
CALL FC 18 (
        Start :=    F100.0,
        Stop :=     FALSE,      // 見つからない
        Funct :=    B#16#4,     // 割出し軸の移動
        Mode :=     B#16#0,     // 絶対位置
        AxisNo :=   4,
        Pos :=      MD104,      //REAL: 1.0;2.0;.. において指定
        FRate :=    MD108,
        InPos :=    F112.0,
        Error :=    F113.0,
        State :=    FY114);
```

5. 軸位置

```
CALL FC 18 (
        Start :=    F100.0,
        Stop :=     FALSE,      // 見つからない。
        Funct :=    B#16#5,     // 軸位置
        Mode :=     B#16#1,     // 相対位置決め
        AxisNo :=   6,
        Pos :=      MD104,
        FRate :=    MD108,
        InPos :=    F112.0,
        Error :=    F113.0,
        State :=    FY114);
```

■ FC 19: MCP_IFM　MCP 信号のインタフェースへの転送

機能の説明

FC MCP_IFM (M 型) を使用して，マシン制御パネル (MCP) から NCK/PLC インタフェース上の対応する信号へ，以下のものを転送します。

オペレーティングモード,

軸選択,

WCS/MCS 切換えコマンド,

移動キー,

オーバライドおよび

キースイッチ

基本プログラムでは (FC 2), FC 27 は，マシン制御パネルまたは運転パネルから選択されたモードを可能になるように，運転パネルあるいは NCK/PLC への MMC からのハンドル選択，モードおよび他の運転信号を転送します。インタフェースへの MMC 信号の転送は，FB 1 (DB 7) "MMCToIF" のパラメーターにある値 "FALSE" を設定して停止することができます。

以下の仕様は，アクティブ運転モードまたは選択座標系による，フィードオーバライド，軸移動キーおよび INC キーに適用します。

- フィードオーバライド :

  フィードオーバライドは，選択されたチャンネルのインタフェース，および軸のインタフェースへ転送されます。

  MMC 信号 "Feed override for rapid traverse effective" （早送りフィードオーバライド有効）がセットされている場合（スイッチ設定 ” ゼロ ” を除く），NC チャンネルに対してのインタフェースバイト "Rapid traverse override"（早送りオーバライド）(DBB 5) に加えて，フィードオーバライド信号が転送されます。
- INC および軸移動キーのためのマシン機能 :

  MCS [1)] が選択されると，信号は，選択されたマシン軸のインタフェースへ転送されます。

  WCS [2)] が選択されると，信号は，パラメータに設定されたチャンネルのジオメトリ軸インタフェースへ転送されます。

  MCS と WCS の間で，システムが切換わると，アクティブな軸の選択は通常取消されます。

MMC からのハンドル選択信号はデコードされて，選択されたハンドルのマシン軸インタフェース，あるいはジオメトリ軸インタフェース内で起動します（FB 1 のパラメータ "HWheelMMC := TRUE" の場合のみ)。

マシン制御パネルの LED は，選択したものに応じて制御されます。

フィードおよびスピンドルのスタート／ストップは，インタフェースには転送されませんが，"feedHold" および "spindleHold" 信号として，モーダルで出力されます。ユーザは，これらの信号を，フィードあるいはスピンドルストップに導く他の信号にリンクすることができます（例えば，FC 10: AL_MSG にある適切な入力信号を使います)。同時に，関連 LED が起動します。

1) MCS= マシン座標系

2) WCS= ワーク座標系

マシン制御パネルが故障すると，ここから出力する信号はゼロにあらかじめセットされます。これは，出力信号 "FeedHold" および "SpindleHold" に当てはまります。

同じ PLC 周期の中で，FC 19 または FC 25 の複数の呼出しができます。この場合，周期の最初の呼出しが LED ディスプレイを駆動します。さらに，パラメータに設定されたブロックのすべての動きは，最初の呼出しで実行します。以下の呼出しでは，チャンネルおよびモードグループインタフェースの処理レベルを減らすことだけを行います。ジオメトリ軸は，方向性のあるデータとともに，周期における最初のブロック呼出しにおいてのみ提供されます。

1 つのブロック処理では，周期の最初の呼出しでのみ，選択／選択解除を行うことができます。

パラメータ ModeGroupNo が，B#16#10 によって増加した場合，二次マシン制御パネルを処理することができます。モードグループ番号は，パラメータがセットされた時，より低いニブル（4 ビットよりも低い）にあります。ModeGroupNo = 0 あるいは B#16#10 は，モードグループ信号が処理されていないことを表しています。

ChanNo = 0 は，チャンネル信号が処理されていないことを表しています。

INC 選択は，モードグループインタフェースにのみ転送されます。これにより，ランタイムが改善されます。このコマンドの起動は，DB10.DBX 57.0. 2 を介した電源投入後すぐにこのブロックによって行われ，マシン制御パネルを，このブロックで並行して操作することができます。2 つのマシン制御パネルのサポートは，マシン制御パネルブロックにある一定の制限によってなされています（2 つのマシン制御パネルの軸選択の相互インターロッキングがあるので，標準軸番号 10 から 31 に対してはサポートしていません）。

アラーム

401901, 401902	FC 19 にあるパラメータ ModeGroupNo は不正，FC 19 にあるパラメータ ChanNo は不正
説明	パラメータに設定されたモードグループ，チャンネルが存在しない
反応	アラーム表示および PLC STOP
処置	パラメータを正しくセット
継続	コールド再スタート後

機能の宣言

```
FUNCTION FC 19: void
//NAME : MCP_IFM
VAR_INPUT
    ModeGroupNo :   BYTE;
    ChanNo :        BYTE;
    SpindleIFNo :   BYTE;
END_VAR
VAR OUTPUT
    FeedHold :      BOOL;
    SpindleHold :   BOOL;
END_VAR
BEGIN
END_FUNCTION
```

正式パラメータの説明

以下は，MCP_IFM 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	注
BAGNo	I	Byte	0 - b#16# および b#16#10 -b#16#1A	転送されるモード信号を操作するモードグループ番号 ModeGroupNo >= b#16#10 は，二次マシン制御パネルへのアクセスを表している
ChanNo	I	Byte	0 - B#16#0A	チャンネル信号用チャンネル番号
SpindleIFNo	I	Byte	0 - 31(B#16#1F)	スピンドルとして宣言した軸インタフェースの番号
FeedHold	Q	Bool		マシン制御パネルからのフィードストップ，モーダル
SpindleHold	Q	Bool		マシン制御パネルからのスピンドルストップ，モーダル

ユーザインタフェース上の MCP の選択した信号の宛先

キースイッチ

ソース :MCP スイッチ	ターゲット : インタフェース DB
Position 0	DB10.DBX56.4
Position 1	DB10.DBX56.5
Position 2	DB10.DBX56.6
Position 3	DB10.DBX56.7

運転モードおよびマシン機能

ソース :MCP キー	インタフェース DB (パラメータ ChanNo)
AUTOMATIC	DB11, ... DBX0.0
MDA	DB11, ... DBX0.1
JOG	DB11, ... DBX0.2
REPOS	DB11, ... DBX1.1
REF	DB11, ... DBX1.2
TEACH IN	DB11, ... DBX1.0

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)
INC 1 ... 10 000, INC Var.	DB21, ... DBB13, ビット 0 から 5
INC 1 ... 10 000, INC Var.	DB21, ... DBB17, ビット 0 から 5
INC 1 ... 10 000, INC Var.	DB21, ... DBB21, ビット 0 から 5

ソース :MCP キー	ターゲット : インタフェース DB (すべての DB 軸)
INC 1 ... 10 000, INC Var.	DB31, ... DBB5, ビット 0 から 5

方向キー，早送りオーバライド

転送は，選択した軸によって異なります。関連インタフェースビットは，選択されなかった軸を削除します。

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
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 DB 軸)
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

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)
フィードレートオーバライド	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 DB 軸)
フィードレートオーバライド	DB31, ... DBB0
スピンドルオーバライド	DB31, ... DBB19 (パラメータ SpindleIFNo)

チャンネル信号

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
ブロック	DB21, ... DBX0.4

フィードレート，スピンドル信号

ソース :MCP キー	ターゲット :FC がパラメータを出力
フィードストップ フィードイネーブル	パラメータ : "FeedHold" ラッチ，LED が駆動
スピンドルストップ スピンドルイネーブル	パラメータ :"SpindleHold" ラッチ，LED が駆動

ディスプレイ制御のためのユーザインタフェースからのチェックバック信号

運転モードおよびマシン機能

ターゲット :MCP LED	ソース : インタフェース DB (パラメータ ModeGroupNo)
AUTOMATIC	DB11, ... DBX6.0
MDA	DB11, ... DBX6.1
JOG	DB11, ... DBX6.2
REPOS	DB11, ... DBX7.1
REF	DB11, ... DBX7.2
TEACH IN	DB11, ... DBX7.0

INC が選択したものについての LED チェックバック信号を獲得するために，すべての信号は OR されます。

ターゲット :MCP LED	ソース : インタフェース DB (パラメータ ChanNo)
INC 1 ... 10 000, INC Var.	DB21, ... DBB41, ビット 0 から 5
INC 1 ... 10 000, INC Var.	DB21, ... DBB47, ビット 0 から 5
INC 1 ... 10 000, INC Var.	DB21, ... DBB53, ビット 0 から 5

ターゲット :MCP LED	ソース : インタフェース DB (すべての DB 軸)
INC 1 ... 10 000, INC Var.	DB31, ... DBB65, ビット 0 から 5

チャンネル信号

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 Start	DB21, ... DBX35.0
NC Stop	DB21, ... DBX35.2 or DB21, ... DBX35.3
block	DB21, ... DBX0.4

（注）方向キー上の LED は，方向キーを作動することによって駆動します。
軸の選択用 LED および WCS/MCS は，適切なキーを作動することによって駆動します。

呼出しの例

```
CALL FC 19(          // マシン制御パネル M 型は，インタフェースへ信号を送る
     ModeGroupNo :=  B#16#1,          // モードグループ番号 1
     ChanNo :=       B#16#1,          // チャンネル番号 1
     SpindleIFNo :=  B#16#4,          // スピンドルインタフェース番号 = 4
     FeedHold :=     f22.0,           // フィードストップ信号モーダル
     SpindleHold :=  db2.dbx151.0);   // メッセージデータブロック内のスピンドルストップモーダル
```

この構成により，最初のモードグループ，最初のチャンネルおよびすべての軸へ，信号を転送します。スピンドルオーバライドは，4 番目の軸／スピンドルインタフェースへ転送されます。信号を保持しているフィードは，ビットメモリ 22.0 へ送られ，スピンドルストップ信号は，データブロック DB2，データビット 151.0 へ送られます。

選択した軸のルーティング

軸選択キーと適切な軸またはスピンドルの間で，フレキシブルな割当てを行うために，FC 19 を修正したり書きなおしたりする必要はありません。以下に述べている解決策を当てはめれば，要求されるフレキシビリティを獲得できます。

1. FC 19 を呼出した後，あたらしく定義された軸選択キーに関する情報（RLO）は，軸番号によって識別されたキー選択へ転送されます。
2. FC 19 を呼出した後，軸番号によって識別された LED に関する情報（RLO）は，新しい軸選択キーの LED およびその前の軸 LED の RLO へ転送され，削除されます。

例 :

スピンドルは第 4 軸として定義され、軸キー 9 経由で選択されなければなりません。

STL 抜出し :

```
a i 5.2;              //9 番目の軸の選択
= i 4.2;              //4 番目の軸を選択する
    call fc 19(
                 ModeGroupNo := b#16#1,
                 ChanNo :=      b#16#1,
                 SpindleIFNo :=  b#16#4,
                 FeedHold :=     m 30.0,
                 SpindleHold :=  m 30.1);
a q 2.5;              //LED 4 番目の軸
= q 3.3;              //LED 9 番目の軸
clr;
= q 2.5;              //LED 4 番目の軸
```

■ FC 21: Transfer PLC-NCK データ交換

機能の説明

Transfer（転送）ブロックが呼出されると，選択された機能コードにしたがって，PLC と NCK の間で，データの交換が行われます。周期の初めにではなく，FC 21 が呼出されてすぐにデータが転送されます。

"Enable"（イネーブル）信号を使用して，ブロックを起動します。FC 21 は，"Enable" = "1" の時だけ処理を行います。

機能の説明，STL 表現

```
VAR_INPUT
    Enable :    BOOL;
    Funct :     BYTE;
    S7Var :     ANY;
    IVar1 :     INT;
    IVar2 :     INT;
END_VAR
VAR_OUTPUT
    Error :     BOOL;
    ErrCode :   INT
END_VAR
```

正式パラメータの説明

以下は，Transfer （転送）機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
Enable	I	Bool		1 = FC 21 起動
Funct	I	Byte	1.5	1: チャンネルへの同期をとった動き 2: チャンネルからの同期をとった動き 3: データ読取り 4: データ書込み 5: チャネルへの制御信号 6: 軸への制御信号
S7Var	I	Any	S7 データ保存エリア	"Funct" 次第
IVAR1	I	Int	0..	"Funct" 次第
IVAR2	I	Int	1..	"Funct" 次第
Error	Q	Bool		
ErrCode	Q	Int		"Funct" 次第

機能

1: チャンネルへの同期をとった動きの信号 :

2: チャンネルからの同期をとった動きの信号 :

同期をとった動きは，PLC によってディセーブルあるいはイネーブルになります。データエリアは，DB21 から DB30, DBB 300 から 307 (チャンネルへ) および DBB 308 から 315 (チャンネルから) にあるユーザインタフェース上に保存されます。パラメータ "S7Var" は，この機能用に評価されませんが，実際のパラメータ（呼出しの例を参照）に割当ててください。データは，FC 21 が処理されるとすぐに，NC から，および NC へ転送されます。以下は適切な信号です。

信号	種類	タイプ	数値の範囲	メモ
Enable	I	Bool		1 = FC 21 起動
Funct	I	Byte	1, 2	1: チャンネルへの同期をとった動き
				2: チャンネルからの同期をとった動き
S7Var	I	Any	S7 データ保存エリア	使用していない
IVAR1	I	Int	1 ..MaxChannel	チャンネル番号
Error	Q	Bool		
ErrCode	Int			1: "Funct" 無効 10: チャンネル番号無効

呼出しの例

```
FUNCTION FC 100: VOID
VAR_TEMP
   myAny:      ANY;
END_VAR
BEGIN
NETWORK
...
// NC チャンネル 1 にある ID3, ID10 および ID31 で，同期をとった動きを停止する
                                                    :
SYAK:          AUF DB21
   SET;
   S DBX300.2;      //ID3
   S DBX301.1;      //ID10
   S DBX303.6;      //ID31
   L B#16#1;
   T MB11;
   SPA TRAN;
// NCK チャンネル 1 からの同期をとった動き :
SYVK:          L B#16#2;
               T MB11;
TRAN: CALL FC 21 (
               Enable :=      F 10.0,      //True の場合 , FC21 起動
               Funct :=       MB 11,
               S7Var :=       #myAny,      // 使用していない
               IVAR1 :=       1,           // チャンネル番号
```

```
                IVAR2 :=        0,
                Error :=        F10.1,
                ErrCode :=      FW 12);
...
END_FUNCTION
```

機能

3, 4: Rapid PLC-NCK データ交換

一般事項

別々になった内部のデータエリアでは，PLC と NCK との間で，高速のデータ交換をすることができます。内部データフィールドのサイズは，あらかじめ 1024 バイトにセットされています。NCK へは，FC21 を介して PLC からアクセスしています（読取り／書込み)。このエリア（構造）の割当てを，NC パートプログラムおよび PLC ユーザプログラムの両方においてまったく同様に宣言してください。これらのデータは，コマンド $A_DBB[x], $A_DBW[x], $A_DBD[x], $A_DBR[x] によって , NC パートプログラムからアクセスすることができます。(プログラミングガイドを参照)。具体的なアドレスは，パラメータ IVAR1 にあるバイトオフセット（0 から最大。サイズ - 1）によって指定されているデータフィールドです。この場合，位置合わせはデータフォーマットにしたがって選択してください。すなわち，Dword は 4 バイト以上で，ワードは 2 バイト以上でスタートします。バイトは，データフィールド内のどのオフセットにも位置付けることができるので，1 つのビットでのアクセス運転はサポートされず，FC 21 によるバイトアクセス運転に変換されます。データタイプの情報およびデータ量は，S7Var を介して転送されたパラメータ ANY から獲得します。データの整合性は，1 バイトおよび 2 バイトのアクセス運転だけが保証されています。これよりも長いデータタイプあるいはフィールド転送の場合，ブロックの有効性および整合性を見極めるために FC 21 で使用するパラメータ IVAR2 に，セマフォをプログラムしてください。NC 内で，すなわちパートプログラム内で，セマフォバイトを書込み，あるいは削除することによって，この動作をサポートします。

セマフォバイトは，IVAR2 の 0 から 1023 の間の値で識別します。

PLC は，FC 21 を使ってセマフォバイトの読取りと書込みを行います。セマフォ変数のセットアップが必要なのは，PLC プログラマだけです。パートプログラムを介して NC からアクセスするには，以下に示すフローチャートにしたがって，個々の命令を使用し，セマフォの機能をプログラムしてください。読取り変数と書込み変数では，シーケンスが異なります。

個々の変数あるいは配列は，セマフォ技術で直接サポートすることができます。転送する構造を，個々のジョブにさらに分割してください。専用のセマフォシステムをプログラミングの時，プログラマはデータ保全性を確実にしてください。

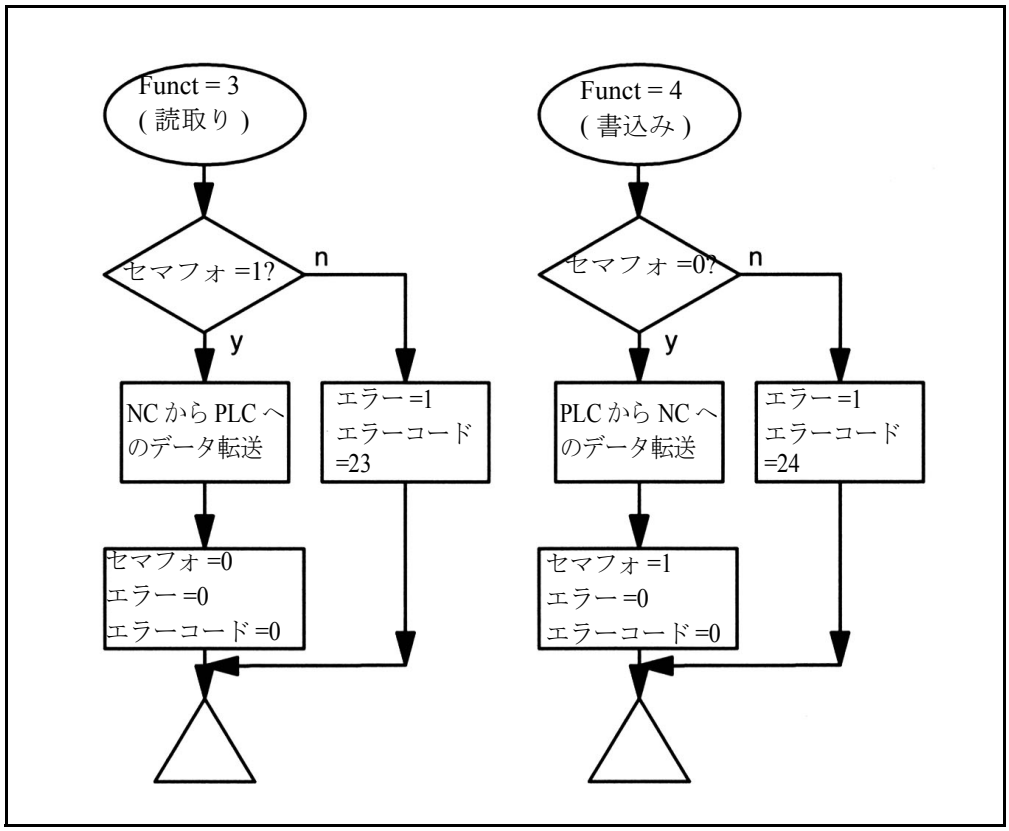


図 1.136 セマフォを使用したデータ交換

1 または 2 バイトの長さのデータ転送は，セマフォを必要としません。

IVAR2 が，-1 にセットされている場合，セマフォなしでデータ転送できます。

以下の信号が関連しています :

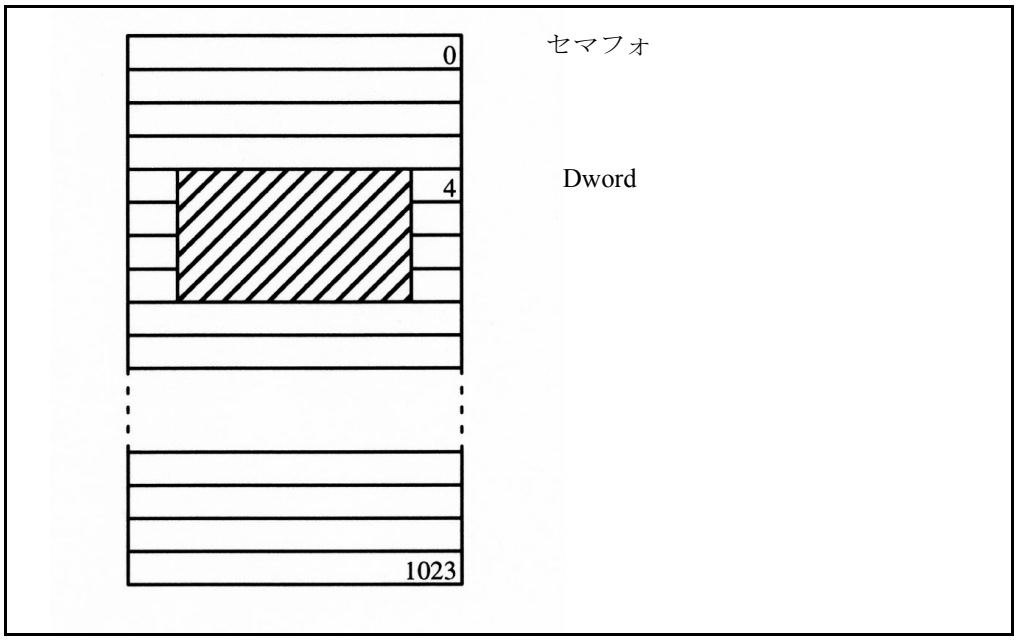
信号	種類	タイプ	数値の範囲	メモ
Enable	I	Bool		1 = FC 21 起動
Funct	I	Byte	3, 4	3: データ読取り 4: データ書込み
S7Var	I	Any	S7 データ保存エリア	ソース／宛先データ保存エリア
IVAR1	I	Int	0..1023	オフセット位置
IVAR2	I	Int	-1 .. 1023	セマフォバイト セマフォなしでの転送 : -1
Error	Q	Bool		
ErrCode	Q	Int		20: 位置合わせエラー 21: 不正なオフセット位置 22: 不正なセマフォバイト 23: 読取る新しいデータはない 24: データ書込みはできない

呼出しの例

1. バイト 0 にあるセマフォでオフセット位置 4 のダブルワードを読取って，MD 100 に保存する :

データタイプ Dword (4 バイト)

オフセット位置 4




```
CALL FC 21 (
        Enable :=       F 10.0,         //True の場合 , FC21 起動
        Funct :=        B#16#3,         // データ読取り
        S7Var :=        P#M 100.0 DWORD 1,
        IVAR1 :=        4,
        IVAR2 :=        0,
        Error :=        F10.1,
        ErrCode :=      FW12);
AN F10.1;           //1 の間はイネーブル , 値を読取るまで
R F10.0;
```

セマフォなしでオフセット位置 8 のワードを読取って，MW 104 に保存する。

```
CALL FC 21 (
        Enable :=       F 10.0,         //True の場合 , FC21 起動
        Funct :=        B#16#3,         // データ読取り
        S7Var :=        P#M 104.0 DWORD 1,
        IVAR1 :=        8,
        IVAR2 :=        -1,
        Error :=        F10.1,
        ErrCode :=      FW12);
```

機能

5: チャンネルへのアップデート制御信号 :

機能 5 の目的は，重要な制御信号を，周期的データ転送間において高速で送信することです。ユーザインタフェース DB21 から DB30 のバイト 6 および 7 のデータは，NC へ転送されます。チャンネルは，パラメータ "IVAR1" で指定されます。これにより，例えば，フィードディスエーブルあるいは読取りディスエーブルが PLC 周期の外側へ転送されます。

以下の信号が関連しています :

信号	種類	タイプ	数値の範囲	メモ
Enable	I	Bool		1 = FC 21 起動
Funct	I	Byte	5	5: チャンネルへの制御信号
S7Var	I	Any	S7 データ保存エリア	使用していない
IVAR1	I	Int	1.. 最大　チャンネル	チャンネル番号
Error	Q	Bool		
ErrCode	Q	Int		1: "Funct" 不正 10: チャンネル番号不正

6: 軸へのアップデート制御信号

機能 6 の目的は，重要な制御信号を，周期的データ転送間において高速で送信することです。アプリケーションインタフェース DB31 から DB61 のバイト 2 のデータは，NC へ転送されます。転送は，すべての起動している軸で実行されます。これにより，例えば，サーボイネーブルが PLC 周期の外側へ転送されます。

以下の信号が関連しています :

信号	種類	タイプ	数値の範囲	メモ
Enable	I	Bool		1 = FC 21 起動
Funct	I	Byte	6	6: 軸への制御信号
S7Var	I	Any	S7 データ保存エリア	使用していない
IVAR1	I	Int	0	
Error	Q	Bool		
ErrCode	Q	Int		1: "Funct" 不正

■ FC 22: TM_DIR ツール管理の方向選択

機能の説明

ブロック TM_DIR は，マガジンまたはリボルバの位置決めに対して，実際の位置および指令の位置を基本とした，できる限り短いルートを提供します。

"1" と同じ長さの信号を Start 入力に適用して，すべての出力パラメータを周期的にアップデートします。つぎの PLC 周期において，パラメータ入力のために変化が起こります（例えば，位置の値）。出力信号は，スタート信号が "0" レベルのとき，定義されません。

特殊位置決め入力 "Offset" > 0 で，方向の選択をする場合，新しい指令の位置は，以下の公式にしたがって，指令位置，特殊位置およびマガジンロケーションの数から計算します。

新しい指令位置 = (指令位置 - (特殊位置 -1) neg. modulo 算出された数。新しい指令位置は，マガジンを位置付けるロケーション番号に対応しています。このため，ユーザによって要求された指令位置は，特殊位置のロケーション番号に対応しています。特殊位置決めの有無にかかわらず，方向性の最適化がなされています。それぞれのマガジンに適切なパラメータの設定を行って，ブロックを呼出してください。

> 警告
>
> ツール管理機能と共に，あるいは DB 74 データブロックを以下の例に示すようにセットアップした後でのみ，ブロックの呼出しをします。この例は，2 つのマガジンがある場合です。最初のマガジンには 10 のロケーションがあり，2 番目のマガジンには 12 のロケーションがあります。実際のマガジンに適応する時は，データ AnzMag, MagNo[.] を変更する必要があるだけです。

```
DATA_BLOCK DB 74
STRUCT
    P: ARRAY [1..9] OF DINT;
    w1: WORD;
    AnzMag: BYTE;
    res:BYTE;
        MagNo: array [1..16] of struct // バイト 40
        AnzPlatz:INT;
        res1:BYTE;
        res2:BYTE;
    end_struct;
end_struct
BEGIN
    P[4]:=L#320; // 絶対に必要 !!!
    AnzMag:=b#16#2;              // マガジンの総数 = 2
    MagNo[1].AnzPlatz:=10;       // マガジンの総ロケーション 1 = 10
    MagNo[2].AnzPlatz:=12;       // マガジンの総ロケーション 2 = 12
END_DATA_BLOCK
```

（注）ツール管理に関するさらに詳しい説明（PLC に関することも）は，ツール管理の機能の説明を参照してください。さらに，PI サービスはツール管理のために，FB 4, FC 7 および FC 8 を介して提供されます（この取扱説明書の対応するセクションも参照してください)。

機能の説明

```
STL 表現
FUNCTION FC 22 : void
// 名前 : TM_DIR
VAR_INPUT
   MagNo:      INT;
   ReqPos:      INT;
   ActPos:      INT;
   Offset:      BYTE;
   Start:       BOOL;
END_VAR
VAR OUTPUT
   Cw:         BOOL;
   Ccw:        BOOL;
   InPos:      BOOL;
   Diff:       INT;
   Error:      BOOL;
END_VAR
BEGIN
END_FUNCTION
```

正式パラメータの説明

以下は，TM_DIR 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
MagNo	I	Int	1..	マガジン番号
ReqPos	I	Int	1..	指令ロケーション
ActPos	I	Int	1..	実際のロケーション
Offset	I	Byte	0..	特殊位置決め
Start	I	Bool		計算のスタート
Cw	Q	Bool		1 = マガジンを時計方向に移動
Ccw	Q	Bool		1 = マガジンを反時計方向に移動
InPos	Q	Bool		1 = 位置にある
Diff	Q	Int	0..	差動パス (1 番短いパス)
Error	Q	Bool		1 = エラー

呼出しの例

```
CALL FC 22(                    // ツール管理方向選択
     MagNo :=      2,          // マガジン番号
     ReqPos :=     fw 20,      // 宛先位置
     ActPos :=     fw 22,      // 現在の位置
     Offset :=     b#16#0,     // 特殊位置用オフセット
     Start :=      m 30.4,     // スタート起動
                               // パラメータリターン
     Cw :=         f 30.0,     // 時計方向にマガジンを移動
     Ccw :=        f 30.1,     // 反時計方向にマガジンを移動
     InPos :=      f 30.2,     // 位置にあるマガジン
     Diff :=       fw 32,      // 差動パス
     Error :=      f 30.3      エラーが発生
```

■ FC 24: MCP_IFM2 MCP 信号のインタフェースへの転送

機能の説明

FC MCP_IFM2 (薄型オペレータパネルの M 型) を使用して，マシン制御パネル (MCP) から NCK/PLC インタフェース上の対応する信号へ，以下のものを転送します。

- モード
- 軸選択
- WCS/MCS 切換えコマンド
- 移動キー
- オーバライドまたはオーバライドシュミレーション信号

基本プログラムでは (FC 2)，ハンドル選択，運転モードおよびその他の運転信号は，オペレータパネル (OP) あるいは MMC から NCK/PLC への転送を続けます。このような運転モードは，MCP または OP どちらからも選択することができます。インタフェースへの MMC 信号の転送は，FB 1 (DB 7) "MMCToIF" のパラメーターにある値 "FALSE" を設定して停止することができます。"MMCTolF" は，セットとリセットを行って，周期的プログラムにおいて起動／停止をすることもできます（例えば、R gp_par.MMCTolF）。

以下の文は，アクティブ運転モードまたは選択座標系の機能として，フィードオーバライド，軸移動および INC ² キーに適用します。

- フィードオーバライド :

フィードオーバライドは，選択されたチャンネルのインタフェース，および軸のインタフェースへ転送されます。

MMC 信号 "Feed override for rapid traverse active" （早送りフィードオーバライド有効）がセットされている場合（スイッチ設定 " ゼロ " を除く），インタフェースバイト "Rapid override"（早送りオーバライド）(DBB 5) と共に，NC チャンネルへ，フィードオーバライド信号が転送されます。"Rapid traverse override effective" （早送りオーバライド実効）もこの MMC 信号でセットします。

- INC および軸移動キーのためのマシン機能 :

MCS が選択されると，選択されたマシン軸のインタフェースへ，信号が転送されます。

WCS が選択されると，パラメータに設定されたチャンネルのジオメトリ軸インタフェースへ，信号が転送されます。

現在選択されている軸は，通常 MCS と WCS　の間での切換えで取消されます。

MMC からのハンドル選択信号はデコードされて，適切なハンドルの関連 (マシン) 軸インタフェース，あるいはジオメトリ軸インタフェース内で起動します（FB 1 のパラメータ "HWheelMMC := TRUE" の場合のみ)。

マシン制御パネルの LED は，選択したものに応じて制御されます。

フィードおよびスピンドルのスタート／ストップは，インタフェースには転送されませんが，"feedHold" および "spindleHold" 信号として，モードで出力されます。ユーザは，これらの信号を，フィードあるいはスピンドルストップに導く他の信号にリンクすることができます（例えば，FC 10: AL_MSG にある適切な入力信号を使います)。同時に，関連 LED が起動します。

スピンドル方向 (+,-) はどちらも直接切換えはできませんが，出力パラメータ "SpindleDir" として有効です。例えば，FC 18 をパラメータに設定することが可能です。スピンドルイネーブル信号は，パラメータ "SpindleHold" を介して切換えることもできます。スピンドルを直接移動するための 1 つの方法として，軸としてあらかじめ選択しておく方法があります。こうすることにより，(軸の) 方向キーを介して転送することができます。

マシン制御パネルが故障すると，ここから出力する信号はゼロに初期化されます。これは，出力信号 "FeedHold" および "SpindleHold" に当てはまります。

同じ PLC 周期の中で，FC 24, FC 19 または FC 25 の複数の呼出しができます。この場合，周期の最初の呼出しが LED ディスプレイを駆動します。さらに，パラメータに設定されたブロックのすべての動きは，最初の呼出しで実行します。以下の呼出しでは，チャンネルおよびモードグループインタフェースの処理レベルを減らすことだけを行います。ジオメトリ軸は，方向性のあるデータとともに，周期における最初のブロック呼出しにおいてのみ提供されます。1 つのブロック処理では，周期の最初の呼出しでのみ，選択／選択解除を行うことができます。

パラメータ ModeGroupNo が，B#16#10 によって増加した場合，二次マシン制御パネルを処理することができます。モードグループ番号は，パラメータがセットされた時，より低いニブル（4 ビットよりも低い）にあります。ModeGroupNo = 0 あるいは B#16#10 は，モードグループ信号が処理されていないことを表しています。

ChanNo = 0 は，チャンネル信号が処理されていないことを表しています。

INC 選択は，モードグループインタフェースにのみ転送されます。これにより，ランタイムが改善されます。このコマンドの起動は，DB10.DBX 57.0. 2 を介した電源投入後すぐにこのブロックによって行われ，マシン制御パネルを，このブロックで並行して操作することができます。2 つのマシン制御パネルのサポートは，マシン制御パネルブロックにある一定の制限によってなされています（2 つのマシン制御パネルの軸選択の相互インターロッキングがあるので，標準軸番号 10 から 31 に対してはサポートしていません)。

アラーム

402401, 402402	FC 24 にあるパラメータ ModeGroupNo は不正，FC 24 にあるパラメータ ChanNo は不正
説明	パラメータに設定されたモードグループ，チャンネルは存在しない
反応	アラーム表示および PLC STOP
処置	パラメータを正しくセット
継続	コールド再スタート後

機能の宣言

```
FUNCTION FC 24: void
// 名前 :          MCP_IFM2
VAR_INPUT
    ModeGroupNo :    BYTE;
    Chao :           BYTE;
    SpindleIFNo :    BYTE;
END_VAR
VAR OUTPUT
    FeedHold :       BOOL;
    SpindleHold :    BOOL;
    SpindleDir:      BOOL;
END_VAR
BEGIN
END_FUNCTION
```

正式パラメータの説明

以下は，MCP_IFM2 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
BAGNo	I	バイト	0 - b#16#0A および b#16#10 - b#16#1A	転送されるモード信号のモードグループ番号 ModeGroupNo >= b#16#10 は，二次マシン制御パネルへのアクセスを表している
Chao	I	バイト	0 - B#16#0A	チャンネル信号用チャンネル番号
SpindleIFNo	I	バイト	0 - 31(B#16#1F)	スピンドルとして宣言した軸インタフェースの番号
FeedHold	Q	Bool		マシン制御パネルからのフィードストップ，モーダル
SpindleHold	Q	Bool		マシン制御パネルからのスピンドルストップ，モーダル
SpindleDir	Q	Bool		スピンドル方向 0 は + (反時計方向)1 は - (時計方向)

呼出しの例

```
CALL FC 24(                    // 薄型マシン制御パネル , インタフェースへの M 型信号
     ModeGroupNo :=   B#16#1,      // モードグループ信号 1
     Chao :=          B#16#1,      // チャンネル番号 1
     SpindleIFNo :=   B#16#4,      // スピンドルインタフェース番号 = 4
     FeedHold :=      f22.0,       // フィードストップ信号モーダル
     SpindleHold :=   db2.dbx151.0); // メッセージ DB 内のスピンドルストップモーダル
     SpindleDir:= f22.1); // スピンドル方向フィードバック
```

この構成により，最初のモードグループ，最初のチャンネルおよびすべての軸へ，信号を転送します。スピンドルオーバライドは，4 番目の軸／スピンドルインタフェースへ転送されます。信号を保持しているフィードは，ビットメモリ 22.0 へ送られ，スピンドルストップ信号は，データブロック DB2，データビット 151.0 へ送られます。パラメータ SpindleDir を介して提供されたスピンドル方向フィードバック信号は，追加の FC 18 を呼出すための直接入力として使用することができます。

■ FC 25: MCP_IFT MCP/OP 信号のインタフェースへの転送

機能の説明

FC MCP_IFT (T 型) を使用して，マシン制御パネル (MCP) から NCK/PLC インタフェースの対応する信号へ，以下のものを転送します。

オペレーティングモード,

4 軸の方向キー,

WCS/MCS 切換えコマンド,

オーバライドおよび

キースイッチ

基本プログラムでは (FC 2), FC 27 は，マシン制御パネルまたは運転パネルから選択されたモードを可能になるように，運転パネルあるいは NCK/PLC への MMC からのハンドル選択，モードおよび他の運転信号を転送します。インタフェースへの MMC 信号の転送は，FB 1 (DB 7) "MMCToIF" のパラメータにある値 "FALSE" を設定して停止することができます。

以下の仕様は，アクティブ運転モードまたは選択座標系による，フィードオーバライド，軸移動キーおよび INC キーに適用します。

- フィードオーバライド :

  フィードオーバライドは，選択されたチャンネルのインタフェース，および軸のインタフェースへ転送されます。

  MMC 信号 "Feed override for rapid traverse effective" （早送りフィードオーバライド有効）がセットされている場合（スイッチ設定 " ゼロ " を除く），NC チャンネルに対してのインタフェースバイト "Rapid traverse override"（早送りオーバライド）(DBB 5) に加えて，フィードオーバライド信号が転送されます。

- INC および軸移動キーのためのマシン機能 :
  MCS 1) (マシン座標系) が選択されると，信号は，選択されたマシン軸のインタフェースへ転送されます。
  WCS 2) (ワークピース座標系) が選択されると，信号は，パラメータに設定されたチャンネルのジオメトリ軸インタフェースへ転送されます。

MMC からのハンドル選択信号はデコードされて，選択されたハンドルのマシン軸インタフェース，あるいはジオメトリ軸インタフェース内で起動します（FB 1 のパラメータ "HWheelMMC := TRUE" の場合のみ)。

マシン制御パネルの LED は，選択したものに応じて制御されます。

フィードおよびスピンドルのスタート／ストップは，インタフェースには転送されませんが，"feedHold" および "spindleHold" 信号として，モーダルで出力されます。ユーザは，これらの信号を，フィードあるいはスピンドルストップに導く他の信号にリンクすることができます（例えば，FC 10: AL_MSG にある適切な入力信号を使います)。同時に，関連 LED が起動します。

マシン制御パネルが故障の場合，ここからの信号はゼロにセットされます。

SW バージョン 4 以降では，同じ PLC 周期の中で，FC 19 または FC 25 の複数の呼出しができます。この場合，周期の最初の呼出しが LED ディスプレイを駆動します。さらに，パラメータに設定されたブロックのすべての動きは，最初の呼出しで実行します。以下の呼出しでは，チャンネルおよびモードグループインタフェースの処理レベルを減らすことだけを行います。ジオメトリ軸は，方向性のあるデータとともに，周期における最初のブロック呼出しにおいてのみ提供されます。

1 つのブロック処理では，周期の最初の呼出しでのみ，選択／選択解除を行います。

パラメータ ModeGroupNo が，B#16#10 によって増加した場合，二次マシン制御パネルを処理することができます。モードグループ番号は，パラメータがセットされた時，より低いニブル（4 ビットよりも低い）にあります。ModeGroupNo = 0 あるいは B#16#10 は，モードグループ信号が処理されていないことを表しています。

ChanNo = 0 は，チャンネル信号が処理されていないことを表しています。

くわしくは，1.15.4「■ FC19:MCP_IFM MCP 信号のインタフェースへの転送」の FC 19 の説明を参照してください。

1) MCS= マシン座標系

2) WCS= ワーク座標系

アラーム

402501, 402502	FC 25 にあるパラメータ ModeGroupNo は不正，FC 25 にあるパラメータ ChanNo は不正
説明	パラメータに設定されたモードグループ，チャンネルは存在しない
反応	アラーム表示および PLC STOP
処置	パラメータを正しくセット
継続	コールド再スタート後

機能の宣言

```
FUNCTION FC 25: void
//NAME :        MCP_IFT
VAR_INPUT
    ModeGroupNo :   BYTE;
    Chao :          BYTE;
    SpindleIFNo :   BYTE;
END_VAR
VAR OUTPUT
    FeedHold :      BOOL;
    SpindleHold :   BOOL;
END_VAR
BEGIN
END_FUNCTION
```

正式パラメータの説明

以下は，MCP_IFT 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
BAGNo	I	バイト	0 - b#16#0A および b#16#10 - b#16#1A	転送するモード信号のモードグループ番号 ModeGroupNo >= b#16#10 は，二次マシン制御パネルへのアクセスを表している
Chao	I	バイト	0 - B#16#0A	チャンネル信号用チャンネル番号
SpindleIFNo	I	バイト	0 - 31 (B#16#1F)	スピンドルとして宣言した軸インタフェースの番号
FeedHold	Q	Bool		マシン制御パネルからのフィードストップ，モーダル
SpindleHold	Q	Bool		マシン制御パネルからのスピンドルストップ，モーダル

呼出しの例

```
CALL FC 25(             // マシン制御パネル , インタフェースへの T 型信号
        ModeGroupNo :=  B#16#1,         // モードグループ番号 1
        Chao :=         B#16#1,         // チャンネル番号 1
        SpindleIFNo :=  B#16#4,         // スピンドルインタフェース
                                        番号 = 4
        FeedHold :=     f22.0,          // フィードストップ信号モー
                                        ダル
        SpindleHold :=  db2.dbx151.0);  // メッセージデータ内のスピ
                                        ンドルストップモーダル
```

この構成により，最初のモードグループ，最初のチャンネルおよびすべての軸へ，信号を転送します。スピンドルオーバライドは，4 番目の軸／スピンドルインタフェースへ転送されます。信号を保持しているフィードは，ビットメモリ 22.0 へ送られ，スピンドルストップ信号は，データブロック DB2，データビット 151.0 へ送られます。

■ FC 26: HPU_MCP HPU 信号のインタフェースへの転送

機能の説明

FC HPU_MCP (ハンドヘルドプログラミングユニットのマシン制御パネル信号) を使用して，マシン制御パネル (MCP) から NCK/PLC インタフェース上の対応する信号へ，以下のものを転送します。

- モード
- WCS/MCS 切換えコマンド
- 移動キー，および
- オーバライド

基本プログラムでは (FC 2)，ハンドル選択，運転モードおよびその他の運転信号は，オペレータパネル (OP) あるいは MMC から NCK/PLC への転送を続けます。このような運転モードは，MCP または OP どちらからも選択することができます。インタフェースへの MMC 信号の転送は，FB 1 (DB 7) "MMCToIF" のパラメータにある値 "FALSE" を設定して停止することができます。

以下の文は，アクティブモードまたは座標系にしたがって，フィードオーバライド，軸移動および INC ² キーに適用します。

フィードレートオーバライド :

フィードオーバライドは，選択されたチャンネルのインタフェース，および軸のインタフェースへ転送されます。

MMC 信号 "Feedrate override for rapid traverse active" (早送りオーバライド有効) がセットされている場合（スイッチ設定 " ゼロ " を除く），インタフェースバイト "Rapid override"（早送りオーバライド）(DBB 5) と共に，NC チャンネルへ，フィードオーバライド信号が転送されます。"Rapid traverse override effective" （早送りオーバライド実効）もこの MMC 信号でセットします。

INC および軸移動キーのためのマシン機能 :

- MCS が選択されると，選択されたマシン軸（6 軸）のインタフェースへ，信号が転送されます。
  WCS が選択されると，パラメータに設定されたチャンネルのジオメトリ軸インタフェースへ，最初の 3 軸の信号が転送されます。
  残りの 3 軸は，選択されたマシン軸のインタフェースへ転送されます。

MMC からのハンドル選択信号はデコードされて，適切なハンドルの関連 (マシン) 軸インタフェース，あるいはジオメトリ軸インタフェース内で起動します（FB 1 のパラメータ "HWheelMMC := TRUE" の場合のみ)。

マシンのコントロールパネルにある LED は , 選択によって変わります。

フィードおよびスピンドルのスタート／ストップは，インタフェースには転送されませんが，"FeedHold" および "SpindleHold" 信号として，モードで出力されます。ユーザは，これらの信号を，フィードあるいはスピンドルストップに導く他の信号にリンクすることができます（例えば，FC 10: AL_MSG にある適切な入力信号を使います)。同時に，関連 LED が起動します。

マシン制御パネルが故障の場合，ここから出力す信号はゼロにセットされます。

同じ PLC 周期の中で，FC 19 または FC 25 の複数の呼出しができます。この場合，周期の最初の呼出しが LED ディスプレイを駆動します。さらに，パラメータに設

定されたブロックのすべての動きは，最初の呼出しで実行します。以下の呼出しでは，チャンネルおよびモードグループインタフェースの処理レベルを減らすことだけを行います。ジオメトリ軸は，方向性のあるデータとともに，周期における最初のブロック呼出しにおいてのみ提供されます。

パラメータ ModeGroupNo が，B#16#10 によって増加した場合，二次マシン制御パネルを処理することができます。モードグループ番号は，パラメータがセットされた時，より低いニブル（4 ビットよりも低い）にあります。ModeGroupNo = 0 あるいは B#16#10 は，モードグループ信号が処理されていないことを表しています。

ChanNo = 0 は，チャンネル信号が処理されていないことを表しています。

INC の選択したものは，モードグループインタフェースにのみ転送されます。これにより，ランタイムが改善されます。このコマンドの起動は，DB10.DBX 57.0. 2 を介した電源投入後すぐにこのブロックによって行われ，マシン制御パネルを，このブロックで並行して操作することができます。2 つのマシン制御パネルのサポートは，マシン制御パネルブロックにある一定の制限によってなされています（2 つのマシン制御パネルの軸選択の相互インターロッキングがあるので，標準軸番号 10 から 31 に対してはサポートしていません）。

1) MCS = マシン座標系

2) WCS = ワーク座標系

アラーム

402601, 402602	FC 26 にあるパラメータ ModeGroupNo は不正，FC 26 にあるパラメータ ChanNo は不正
説明	パラメータに設定されたモードグループ，チャンネルは存在しない
反応	アラーム表示および PLC STOP
処置	パラメータを正しくセット
継続	コールド再スタート後

機能の宣言

```
FUNCTION FC 26: void
// 名前 :          HPU_MCP
VAR_INPUT
    ModeGroupNo :    BYTE;
    Chao :           BYTE;
END_VAR
BEGIN
END_FUNCTION
```

正式パラメータの説明

以下は，HPU_MCP 機能のすべての正式パラメータについての表です。

信号	種類	タイプ	数値の範囲	メモ
BAGNo	I	Byte	0 - b#16#0A および b#16#10 - b#16#1A	転送されるモード信号のモードグループ番号 ModeGroupNo >= b#16#10 は，二次マシン制御パネルへのアクセスを表している
Chao	I	Byte	0 - B#16#0A	チャンネル信号用チャンネル番号

ユーザインタフェース上の MCP の選択した信号の宛先

MCP 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 (パラメータ ModeGroupNo) モードグループ 1 の表現
AUTOMATIC	DB11.DBX0.0
MDA	DB11.DBX0.1
JOG	DB11.DBX0.2
REPOS	DB11.DBX1.1
REF	DB11.DBX1.2
TEACH_IN	DB11.DBX1.0

インクリメント

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)
INC 1 ... 10 000, INC Var.	DB21, ... DBB13, ビット 0 から 5
INC 1 ... 10 000, INC Var.	DB21, ... DBB17, ビット 0 から 5
INC 1 ... 10 000, INC Var.	DB21, ... DBB21, ビット 0 から 5

運転モードおよびマシン機能

ソース :MCP キー	ターゲット : インタフェース DB (6 軸 DB)
INC 1 ... 10 000, INC Var.	DB31, ... DBB5, ビット 0 から 5

方向キー，早送りオーバライド

転送は，選択した軸によって異なります。関連インタフェースビットは，選択されなかった軸を削除します。

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
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 (6 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

オーバライド

ソース :MCP 設定	ターゲット : インタフェース DB (パラメータ ChanNo)
フィードレートオーバライド	DB21, ... DBB4

ソース :MCP 設定	ターゲット : インタフェース DB (6 軸 DB)
フィードレートオーバライド	DB31, ... DBB0
スピンドルオーバライド	DB31, ... DBB19 (パラメータ SpindleIFNo)

チャンネル信号

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
ブロック	DB21, ... DBX0.4

ディスプレイ制御のためのユーザインタフェースからのチェックバック信号

運転モードおよびマシン機能

表 1.71 運転モードおよびマシン機能

ターゲット :MCP 出力	ソース : インタフェース DB (パラメータ ModeGroupNo) モードグループ 1 の表現
AUTOMATIC	DB11, ... DBX6.0
MDA	DB11, ... DBX6.1
JOG	DB11, ... DBX6.2
REPOS	DB11, ... DBX7.1
REF	DB11, ... DBX7.2
TEACH IN	DB11, ... DBX7.0

WCS/MCS 出力は，キー操作によって動作します。

呼出しの例

```
CALL FC 26(       //HPU のマシン制御パネルは，インタフェースに信号を送る
          ModeGroupNo :=   B#16#1,      // モードグループ番号 1
          Chao :=          B#16#1);     // チャンネル番号 1
```

これらのパラメータ設定を行って，最初にパラメータに設定されたマシン制御パネルからの信号を，最初のモードグループ，最初のチャンネルおよびすべての軸へ送ります。

■ FC 19, FC 24, FC 25, FC 26 ソースコード説明

タスク

アプリケーションインタフェースへのマシン制御パネル
(FC 19 M バージョン , FC 24 薄型バージョン , FC 25 T バージョン , FC 26 HPU バージョン)。

関連ブロック

DB 7 (SW バージョン 3 までは，DB 5), モードグループの番号 , チャンネル , 軸

DB 7, マシン制御パネルへのポインタ

DB 8, つぎの周期へ保存

FC 20, エラーメッセージの出力

使用するリソース

なし

一般情報

ブロック FC 19 (M バージョン), FC 24 (薄型バージョン), FC 25 (T バージョン) および FC 26 (HPU バージョン) は，マシン制御パネルを，アプリケーションインタフェースへ，およびアプリケーションインタフェースから転送します。入力パラ

メータの "ModeGroupNo" は，ブロックで処理するモードグループを指定します。パラメータ "ModeGroupNo" は，マシン制御パネルの番号も指定します。"ChanNo" では，処理するチャンネルを指定します。パラメータ "SpindleIFNo" は，スピンドルの軸インタフェースを指定します。スピンドルオーバライドは，このスピンドルインタフェースへ転送されます。パラメータのエラーチェックが行われます。出力パラメータ "FeedHold" および "SpindleHold" は，フィード／スピンドルがディスエーブルである，フィード／スピンドルがイネーブルである，および万一ディスエーブルの時に論理的 1 で戻ってくる 4 個のキーのから，出力パラメータ "FeedHold" および "SpindleHold" が生成します。つぎの周期の情報は，マシン制御パネル番号によって，0 から 3 バイトあるいは 62 から 65 バイトで DB8 に保存されます。この情報は，エッジトリガフラッグ，フィード値および選択された軸番号です。
このブロックは，DB 7 MCP1In および MCP1Out (MCP2In および MCP2Out) にあるポインタパラメータを介して，ユーザデータとともに提供されます。絶対アドレス指定を避けるために，より遠いポインタを介して，DB7 の VAR セクションから間接的にこのポインタをアドレス指定します。この追加のポインタは，FB1 において記号で確定されます。

ブロック説明

4 つのブロックは，すべて同じ構造です。ブロックは，個々のサブタスク用に別々のセクションの形で構成されています。

ネットワーク入力において，さまざまなパラメータがローカル変数へコピーされます。マシン制御パネル信号（入力／出力エリアのユーザデータ）も，DB 7 (gp_par) にあるさまざまなポインタを使って，ロケーションとロケーションの間でコピーされます。これらのローカル変数は，効率の点からも，ブロック内で処理されます。スタートアップするために，いくつかの変数を初期化します。

ブロックにおいてさらに処理を行うために，エッジ評価，軸選択，方向キーおよび早送りオーバレイによる MCS/WCS 切換えを，Network Global_in で確定します。ユーザ別変更を，プログラムのこのセクションで行ってください。ユーザ別変更は，通常は軸選択が主となります。

キースイッチの情報のみ , NC ネットワークにおいてコピーします。

モードグループネットワークは, NCK への動的信号として，キーのモードを転送します。モードグループ番号が 0 の場合，このネットワークは処理されません。番号が大きすぎる場合は，401901 あるいは 402501 が出力され，制御スイッチはストップモードになります。

ジオメトリ軸をインクリメントするための選択は，ネットワークチャンネルにあるすべてのジオメトリ軸（ソフトウェアバージョン 4 以前）用に，キーからインタフェースへ直接送られます。NC スタート，ストップ，リセットおよびシングルブロックの機能は，チェックバック信号に対応して起動します。あらかじめ対応するものが選択されていれば，ジオメトリ軸の方向キーが提供されます。そうでなければ，クリアとなります。チャンネル番号が 0 の場合，このネットワークは処理されません。番号が大きすぎる場合は，401902 あるいは 402502 が出力され，制御スイッチはストップモードになります。スピンドルネットワークは，SpindleIFNo を介して構成されたインタフェースへ，スピンドルオーバライドを転

送します。

軸ネットワークは，選択された軸インタフェースへ，フィードレートオーバライドを転送します。方向キーを，選択された軸／スピンドルへ割当てます。軸が前もって選択されている場合，方向の情報は，0 にセットされます。選択された軸の INC チェックバックが表示されます。

出力パラメータが準備され，INC マシン機能の LED 信号が，global_out ネットワーク内で生成されます。

出力ネットワークは，マシン制御パネルの出力信号を，VAR_TEMP イメージから論理アドレスへ転送します。つぎの周期のデータも保存されます。

軸選択の選択

10 以上の軸を使用している場合 , ネットワーク Global_in を変更する必要があります。さらに多くのキーおよびマシン制御パネル LED を使用する場合，以下に示す措置をとってください。

1. コマンド UD DW#16#Value で，軸選択のために定義されたすべての LED を削除します。現時点では，ビットマスクで処理できるのは，9 の軸選択となっています。

2. コメント "mask all axis selection keys"（すべての軸選択キーをマスキング）のあるコマンド UW W#16# で，新しい方向選択が行われているかどうかをチェックします。ビット列はここで変更してください。

3. ジャンプ列は，新しいジャンプラベルで拡張してください。新しいジャンプラベルは，ラベル m009 の前に降順で挿入してください。選択情報は，ラベル m009 および m008 で表すように，新しいジャンプラベル用に拡張してください。

1.15.5 信号／データの説明

■ NCK/PLC, MMC/PLC, MCP/PLC のインタフェース信号

一般事項

NCK/PLC, MMC/PLC および MCP/PLC のインタフェース信号は，別資料の入出力信号一覧に記載してあります。

参照： /LIS/, Lists

YS840DI 用のこのリストは，対応する信号を記述している機能の説明のうち，関連するものを参照にしてリスト化しています。ユーザインタフェースへ転送する前に，基本プログラムによって評価され，規定される NCK 信号を，以下のセクションに記載しています。

■ デコードされた M 信号

一般事項

デコードされた M 信号は，以下の表にあるデータビットに入力します。NC チャンネル 1 では DB 21 に，NC チャンネル 2 では DB 22 に入力します。信号の長さは，PLC 1 周期です。

ユーザは，信号を評価することによって，それぞれの M グループに対してアクティブ信号を簡単に引出すことができます。

（注）スピンドル用にデコードされた M 機能はありません (M3, M4, M5, M70)。

DB21/22 内のアドレス	変数	タイプ	初期値	コメント
DBX 194.0 .. 7	M_Fkt_M0 .. M7	Bool		M 信号 M0 .. M7
DBX 195.0 .. 7	M_Fkt_M8 .. M15	Bool		M 信号 M8 .. M15
DBX 196.0 .. 7	M_Fkt_M16 .. M23	Bool		M 信号 M16 .. M23
DBX 197.0 .. 7	M_Fkt_M24 .. M31	Bool		M 信号 M24 .. M31
DBX 198.0 .. 7	M_Fkt_M32 .. M39	Bool		M 信号 M32 .. M39
DBX 199.0 .. 7	M_Fkt_M40 .. M47	Bool		M 信号 M40 .. M47
DBX 200.0 .. 7	M_Fkt_M48 .. M55	Bool		M 信号 M48 .. M55
DBX 201.0 .. 7	M_Fkt_M56 .. M63	Bool		M 信号 M56 .. M63
DBX 202.0 .. 7	M_Fkt_M64 .. M71	Bool		M 信号 M64 .. M71
DBX 203.0 .. 7	M_Fkt_M72 .. M79	Bool		M 信号 M72 .. M79
DBX 204.0 .. 7	M_Fkt_M80 .. M87	Bool		M 信号 M80 .. M87
DBX 205.0 .. 7	M_Fkt_M88 .. M95	Bool		M 信号 M88 .. M95
DBX 206.0 ..3	M_Fkt_M96 .. M99	Bool		M 信号 M96 .. M99

■ G 機能

一般事項

G 機能は，NC チャンネル 1 では DB 21 へ，NC チャンネル 2 では DB 22 へ入力します。データバイト（8 ビット整数）に入力された変数の意味は，1.15.5「■ G 機能（値）」を参照してください。
電源を投入すると，すべての G グループのユーザインタフェースで値ゼロが定義されます（すなわちアクティブ G グループは不定となります）。
NC プログラムを終了あるいは中断した場合，グループの最後のステータスは保持されます。

SW 3.1 およびそれ以前のバージョン : NC プログラムのスタートに続いて，最初の 8 G グループの値を，NC プログラムにリスト化された値として，マシンデータによって定義されたデフォルト設定にしたがって上書きします。

SW バージョン 3.2 以降 : NC プログラムのスタートに続いて，MD 22510 において指定された 8 G グループの値を，NC プログラムにリスト化された値として，マシンデータによって定義されたデフォルト設定にしたがって上書きします。

DB21/22 内のアドレス	変数	タイプ	初期値	コメント
BB 208	G_FKT_GR_1	Byte	0	グループ 1 のアクティブ G 機能
DBB 209	G_FKT_GR_2	Byte	0	グループ 2 のアクティブ G 機能
DBB 210	G_FKT_GR_3	Byte	0	グループ 3 のアクティブ G 機能
DBB 211	G_FKT_GR_4	Byte	0	グループ 4 のアクティブ G 機能
DBB 212	G_FKT_GR_5	Byte	0	グループ 5 のアクティブ G 機能
DBB 213	G_FKT_GR_6	Byte	0	グループ 6 のアクティブ G 機能
DBB 214	G_FKT_GR_7	Byte	0	グループ 7 のアクティブ G 機能
DBB 215	G_FKT_GR_8	Byte	0	グループ 8 のアクティブ G 機能
DBB 216	G_FKT_GR_9	Byte	0	グループ 9 のアクティブ G 機能
DBB 217	G_FKT_GR_10	Byte	0	グループ 10 のアクティブ G 機能
DBB 218	G_FKT_GR_11	Byte	0	グループ 11 のアクティブ G 機能
DBB 219	G_FKT_GR_12	Byte	0	グループ 12 のアクティブ G 機能
DBB 220	G_FKT_GR_13	Byte	0	グループ 13 のアクティブ G 機能
DBB 221	G_FKT_GR_14	Byte	0	グループ 14 のアクティブ G 機能
DBB 222	G_FKT_GR_15	Byte	0	グループ 15 のアクティブ G 機能

DB21/22 内のアドレス	変数	タイプ	初期値	コメント
DBB 223	G_FKT_GR_16	Byte	0	グループ 16 のアクティブ G 機能
DBB 224	G_FKT_GR_17	Byte	0	グループ 17 のアクティブ G 機能
DBB 225	G_FKT_GR_18	Byte	0	グループ 18 のアクティブ G 機能
DBB 226	G_FKT_GR_19	Byte	0	グループ 19 のアクティブ G 機能
DBB 227	G_FKT_GR_20	Byte	0	グループ 20 のアクティブ G 機能
DBB 228	G_FKT_GR_21	Byte	0	グループ 21 のアクティブ G 機能
DBB 229	G_FKT_GR_22	Byte	0	グループ 22 のアクティブ G 機能
DBB 230	G_FKT_GR_23	Byte	0	グループ 23 のアクティブ G 機能
DBB 231	G_FKT_GR_24	Byte	0	グループ 24 のアクティブ G 機能
DBB 232	G_FKT_GR_25	Byte	0	グループ 25 のアクティブ G 機能
DBB 233	G_FKT_GR_26	Byte	0	グループ 26 のアクティブ G 機能
DBB 234	G_FKT_GR_27	Byte	0	グループ 27 のアクティブ G 機能
DBB 235	G_FKT_GR_28	Byte	0	グループ 28 のアクティブ G 機能
DBB 236	G_FKT_GR_29	Byte	0	グループ 29 のアクティブ G 機能
DBB 237	G_FKT_GR_30	Byte	0	グループ 30 のアクティブ G 機能
DBB 238	G_FKT_GR_31	Byte	0	グループ 31 のアクティブ G 機能
DBB 239	G_FKT_GR_32	Byte	0	グループ 32 のアクティブ G 機能
DBB 240	G_FKT_GR_33	Byte	0	グループ 33 のアクティブ G 機能
DBB 241	G_FKT_GR_34	Byte	0	グループ 34 のアクティブ G 機能
DBB 242	G_FKT_GR_35	Byte	0	グループ 35 のアクティブ G 機能
DBB 243	G_FKT_GR_36	Byte	0	グループ 36 のアクティブ G 機能
DBB 244	G_FKT_GR_37	Byte	0	グループ 37 のアクティブ G 機能
DBB 245	G_FKT_GR_38	Byte	0	グループ 38 のアクティブ G 機能
DBB 246	G_FKT_GR_39	Byte	0	グループ 39 のアクティブ G 機能

■ G 機能 (値)

G 機能 (値)

G 機能のリストは，以下を参照してください。

参照： /PG/, Programming Guide Fundamentals

2 番目の列で指定された番号は，グループ内において機能を一意に識別します。対応する番号は，PLC インタフェースへ転送されます。

■ DB2 のメッセージ信号

一般事項

ユーザは DB2 を使用して，制御パネル上に個々の信号用メッセージを表示することができます。インタフェース信号のリストが現れると，信号はあらかじめ定義されたグループに分けられます。メッセージが発生，消滅，あるいは確認応答された場合，メッセージ番号の列に入力された番号は，MMC へ転送されます。テキストは，それぞれのメッセージ用に MMC に保存することができます（SINUMERIK 840D の Installation and Start-up Guide （インストールおよびスタートアップガイド）のセクション "Message numbers" （メッセージ番号）を参照)。

（注）ユーザエリアの数は，FB 1 を介してパラメータに設定することができます。
構成を変更した後 (FB 1: MsgUser), DB 2/3 を削除してください。

DB 2 のチャンネルエリア

エリア	アドレス	メッセージ番号
チャンネル 1	DBX0.0 - DBX11.7	510.000-510.231
チャンネル 1, ジオメトリ軸	DBX12.0 - DBX17.7	511.100-511.315
チャンネル 2	DBX18.0 - DBX29.7	520.000-520.231
チャンネル 2, ジオメトリ軸	DBX30.0 - DBX35.7	521.100-521.315
チャンネル 3	DBX36.0 - DBX47.7	530.000-530.231
チャンネル 3, ジオメトリ軸	DBX48.0 - DBX53.7	531.100-531.315
チャンネル 4	DBX54.0 - DBX65.7	540.000-540.231
チャンネル 4, ジオメトリ軸	DBX66.0 - DBX71.7	541.100-541.315
チャンネル 5	DBX72.0 - DBX83.7	550.000-550.231
チャンネル 5, ジオメトリ軸	DBX84.0 - DBX89.7	551.100-551.315
チャンネル 6	DBX90.0 - DBX101.7	560.000-560.231
チャンネル 6, ジオメトリ軸	DBX102.0 - DBX107.7	561.100-561.315
チャンネル 7	DBX108.0 - DBX119.7	570.000-570.231
チャンネル 7, ジオメトリ軸	DBX120.0 - DBX125.7	571.100-571.315
チャンネル 8	DBX126.0 - DBX137.7	580.000-580.231
チャンネル 8, ジオメトリ軸	DBX138.0 - DBX143.7	581.100-581.315
チャンネル 9 および 10 は，現在実装されていない		

DB 2 のユーザエリア

エリア	アドレス	メッセージ番号
軸／スピンドル 1	DBX144.0 - DBX145.7	600.100-600.115
軸／スピンドル 2	DBX146.0 - DBX147.7	600.200-600.215
軸／スピンドル 3	DBX148.0 - DBX149.7	600.300-600.315
軸／スピンドル 4	DBX150.0 - DBX151.7	600.400-600.415
軸／スピンドル 5	DBX152.0 - DBX153.7	600.500-600.515
軸／スピンドル 6	DBX154.0 - DBX155.7	600.600-600.615
軸／スピンドル 7	DBX156.0 - DBX157.7	600.700-600.715
軸／スピンドル 8	DBX158.0 - DBX159.7	600.800-600.815
軸／スピンドル 9	DBX160.0 - DBX161.7	600.900-600.915
軸／スピンドル 10	DBX162.0 - DBX163.7	601.000-601.015
軸／スピンドル 11	DBX164.0 - DBX165.7	601.100-601.115
軸／スピンドル 12	DBX166.0 - DBX167.7	601.200-601.215
軸／スピンドル 13	DBX168.0 - DBX169.7	601.300-601.315
軸／スピンドル 14	DBX170.0 - DBX171.7	601.400-601.415
軸／スピンドル 15	DBX172.0 - DBX173.7	601.500-601.515
軸／スピンドル 16	DBX174.0 - DBX175.7	601.600-601.615
軸／スピンドル 17	DBX176.0 - DBX177.7	601.700-601.715
軸／スピンドル 18	DBX178.0 - DBX179.7	601.800-601.815
軸 19 から 31 は，現在実装されていない		

DB 2 のユーザエリア

エリア	アドレス	メッセージ番号
ユーザエリア 0	DBX180.0 - DBX187.7	700.000 - 700.063
ユーザエリア 1	DBX188.0 - DBX195.7	700.100 - 700.163
ユーザエリア 2	DBX196.0 - DBX203.7	700.200 - 700.263
ユーザエリア 3	DBX204.0 - DBX211.7	700.300 - 700.363
ユーザエリア 4	DBX212.0 - DBX219.7	700.400 - 700.463
ユーザエリア 5	DBX220.0 - DBX227.7	700.500 - 700.563
ユーザエリア 6	DBX228.0 - DBX235.7	700.600 - 700.663
ユーザエリア 7	DBX236.0 - DBX243.7	700.700 - 700.763
ユーザエリア 8	DBX244.0 - DBX251.7	700.800 - 700.863
ユーザエリア 9	DBX252.0 - DBX259.7	700.900 - 700.963
ユーザエリア 10	DBX260.0 - DBX267.7	701.000 - 701.063
ユーザエリア 11	DBX268.0 - DBX275.7	701.100 - 701.163
ユーザエリア 12	DBX276.0 - DBX283.7	701.200 - 701.263
ユーザエリア 13	DBX284.0 - DBX291.7	701.300 - 701.363
ユーザエリア 14	DBX292.0 - DBX299.7	701.400 - 701.463
ユーザエリア 15	DBX300.0 - DBX307.7	701.500 - 701.563
ユーザエリア 16	DBX308.0 - DBX315.7	701.600 - 701.663
ユーザエリア 17	DBX316.0 - DBX323.7	701.700 - 701.763
ユーザエリア 18	DBX324.0 - DBX331.7	701.800 - 701.863
ユーザエリア 19	DBX332.0 - DBX339.7	701.900 - 701.963
ユーザエリア 20	DBX340.0 - DBX347.7	702.000 - 702.063
ユーザエリア 21	DBX348.0 - DBX355.7	702.100 - 702.163
ユーザエリア 22	DBX356.0 - DBX363.7	702.200 - 702.263
ユーザエリア 23	DBX364.0 - DBX371.7	702.300 - 702.363
ユーザエリア 24	DBX372.0 - DBX379.7	702.400 - 702.463

1.15.6 データフィールド，リスト

（注）FB, FC, DB の割当ておよびインタフェース信号の概要については，以下を参照してください。

参照： /LIS/, Lists.

■ FB/FC 割当て

FB 番号	FC 番号	意味
1		基本プログラム
2 ~ 29		シーメンス用に予約
	1	基本プログラムの初期化
	2 ~ 29	シーメンス用に予約
	30 ~ 35	ShopMill, ManualTum についてのメモを参照
	30 ~ 127	ユーザ割当て用
30 ~ 127		ユーザ割当て用

（注）FC および FB の割当てについては , 1.15.6「■ NC 用基本 PLC プログラムに必要なメモリ」を参照してください。

■ DB 割当て

（注）NC-MD にしたがって必要な数だけ DB をセットアップします。

表 1.72　データブロックの概要

DB 番号	宛先	名前	パッケージ
1		シーメンス用に未使用	BP
2 ~ 4	PLC-MELD	PLC メッセージ	BP
5 ~ 8		シーメンス用に未使用	
9	NC-COMPILE	NC コンパイル周期用インタフェース	BP
10	NC INTER-FACE	中央 NC インタフェース	BP
11	BAG 1	インタフェースモードグループ	BP
12		コンピュータリンクおよびトランスポートシステムインタフェース	
13 ~ 14		シーメンス用に未使用	
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未使用	
19		MMC インタフェース	
20		PLC マシンデータ	
21 ~ 30	CHAEL 1	インタフェース NC チャンネル	BP
31 ~ 61	AXIS 1	軸／スピンドル用インタフェース，またはユーザ割当て用	BP
62 ~ 70		ユーザ割当て用	
71 ~ 74		ツール管理	BP
75 ~ 76		M グループデコーディング	
77 ~ 80		シーメンス用に未使用	
81 ~ 89		ShopMill, ManualTurn についての記載を参照	
81 ~ 127		ユーザ割当て用	

（注）起動していないチャンネル，軸／スピンドルおよびツール管理機能のデータブロックは，ユーザの要求通りに割当てられます。

ManualTurn

ManualTurn は，FC 30 から 35 および DB 81 から 89 を使用します。

ManualTurn は，従来の周期制御の旋削マシン用オペレーティングシステムです。構成するマシンが，最大で 2 つの軸および 1 つのスピンドルの旋削マシンでない場合でも，躊躇することなく上記の FC および DB を使用することができます。あなたのマシンがこのタイプのマシンで，CNC 機能に加えて従来のオペレーティングメソッドを必要とする場合，上記の FC および DB を使用しないでください。

ShopMill

ShopMill は，FC30 から 35 および DB 81 から 89 を使用します。

ShopMill は，作業場における 2 1/2 フライスマシン用オペレーティングシステムです。構成するマシンが，2 1/2 マシニングオペレーション用フライスマシンでない場合でも，躊躇することなく上記の FC および DB を使用することができます。ただし，あなたのマシンをこのタイプのアプリケーションに使用する意向の場合，上記の FC および DB を使用しないでください。

■ 割当てタイマ

タイマ番号	意味
0 ~ 9	シーメンス用に未使用
10 ~ 127	ユーザ割当て用

■ NC 用基本 PLC プログラムに必要なメモリ

一般事項

基本プログラムは，基本機能とオプショナル機能から成っています。基本機能には，NC と PLC の間の周期的信号交換があります。オプションには，必要な時に使用することができる FC などがあります。

以下は，基本機能およびオプションに必要なメモリの一覧表です。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。

表 1.73 ブロックに必要なメモリ

ブロックタイプ，番号	機能	コメント	ブロックサイズ (バイト)	
			ロードメモリ	作業メモリ
基本プログラムにある基本機能				
FB 1, 15, 16, 17,18		ロードすること	3616	3052
FC 1, 2, 3, 4, 11,20		ロードすること	7208	6608
DB 4, 5, 7, 8, 17,19		ロードすること	2490	966
DB 2, 3, 6		BP によって生成	992	812
OB 1, 40, 100		ロードすること	490	282
		合計	14796	11720
PLC/NCK, PLC/MMC インタフェース				
DB 10	PLC/NCK 信号	ロードすること	318	262
DB 11	PLC/ モードグループ信号	BP によって生成	80	44
DB 21, 30	PLC/ チャンネル信号	NC-MD の機能として BP によって生成	各 352	各 316
DB 31, ...61	PLC/ 軸およびスピンドル信号	NC-MD の機能として BP によって生成	各 180	各 144
基本プログラムオプション マシン制御パネル				
FC 19	MCP 信号 , M 変数の転送	MCP の M 変数をインストールする時にロードすること	1498	1258
FC 25	MCP 信号 , T 変数の転送	MCP の T 変数をインストールする時にロードすること	1358	1160
FC 24	MCP 信号 , スリム変数の転送	MCP の スリム変数をインストールする時にロードすること	1358	1160
FC 26	CP 信号 , HPU 変数の転送	HPU 用にロードすること	1358	1160
FC 14	MPI/OPI 転送	MCPNum > 0 で転送すること	942	802
ハンドヘルドユニット				
FC 13	制御 HHU 表示	ハンドヘルドユニット用にロードできる	1264	1044

ブロックタイプ，番号	機能	コメント	ブロックサイズ (バイト)	
			ロードメモリ	作業メモリ
エラー／ステータスメッセージ				
FC 10	FM/BM 収集	FM/BM の使用時にロード	1572	1350
ASUP				
FC 9	ASUP スタート	PLC ASUP の使用時	656	538
基本プログラムオプション 同時軸				
FC 15	直線／回転軸の位置決め	PLC による軸位置決め用にロード	656	546
FC 16	割出し軸の位置決め	PLC による軸位置決め用にロード	674	560
スター／デルタ切換え				
FC 17	MSD のスター／デルタ切換え	スター／デルタ 切換え用にロード	612	494
スピンドル制御				
FC 18	スピンドル制御	PLC からのスピンドル制御用にロード	826	676
PLC/NC コミュニケーション				
FB 2	NC 変数読取り	"NC 変数の読取り " 用にロード	396	224
DB n 1)	NC 変数読取り	FB 2 呼出しにつきインスタンス DB は 1	各 426	各 270
FB 3	NC 変数書込み	"NC 変数書込み " 用にロード	396	224
DB m 1)	NC 変数書込み	FB 3 呼出しにつきインスタンス DB は 1	各 426	各 270
FB 4	PI サービス	PI サービス用にロード	334	214
DB o 1)	PI サービス	FB4 呼出しにつきインスタンス DB は 1	各 234	各 130
DB 16	PI サービス説明	PI サービス用にロード	1190	408
FB5	GUD 変数読取り	PI サービス用にロード	532	365
DB p	GUD 変数読取り	FB5 呼出しにつきインスタンス DB は 1	各 308	各 166
FB 6	ジェネラルコミュニケーション	NC 変数および PI サービスの読取り／書き込み用にロード	5986	5228
DB 15	ジェネラルコミュニケーション	FB6 用インスタンス DB	440	172
ツール管理				
FC 6	基本機能	ツール管理オプション用にロード	1382	1182
FC 7	機能リボルバ転送	ツール管理オプション用にロード	530	430
FC 8	転送機能	ツール管理オプション用にロード	1002	834
FC 22	方向選択	方向選択が必要な場合ロード	404	300
DB 71	ロード位置	NC-MD の機能として BP によって生成	76+30*B	40+30*B
DB 72	スピンドル	NC-MD の機能として BP によって生成	76+48*Sp	0+48*Sp
DB 73	リボルバ	NC-MD の機能として BP によって生成	76+44*R	40+44*R

DB 74	基本機能	NC-MD の機能として BP によって生成	136+(B+Sp+R)*20	100+(B+Sp+R)*20
コンパイル周期				
DB 9	インタフェース PLC コンパイル周期	NC オプションの機能として BP によって生成	472	436
1) : DB 番号は，PLC ユーザで指定してください。				

例：

上記の表で指定している必要メモリを基礎にして，構成の 2 つのサンプルにおける必要メモリを計算します（以下の表を参照してください）。

ブロックタイプ，番号	機能	コメント	ブロックサイズ (バイト)	
			ロードメモリ	作業メモリ
最小構成 (1 スピンドル , 2 軸および T-MCP)				
表 1.73 を参照	基本プログラム，ベース		14796	11720
	インタフェース DB		1290	1054
	MCP		2300	1962
		合計	18386	14736

ブロックタイプ，番号	機能	コメント	ブロックサイズ (バイト)	
			ロードメモリ	作業メモリ
最大構成 (2 チャンネル , 4 スピンドル , 4 軸，T-MCP)				
表 1.73 を参照	基本プログラム，ベース		14796	11720
表 1.73 を参照	インタフェース DB		2542	2090
表 1.73 を参照	MCP		2300	1962
表 1.73 を参照	エラー／ステータスメッセージ		1572	1350
表 1.73 を参照	ASUP	1 ASUP 開始	656	538
表 1.73 を参照	同時軸	2 タレット用	674	560
表 1.73 を参照	PLC/NC コミュニケーション	1 x 読取り変数および 1 x 書込み変数	8070	6388
表 1.73 を参照	ツール管理	1 つのローディングポイントにつき 2 タレット	3430	2854
表 1.73 を参照	コンパイル周期		472	436
		合計	34512	27898

1.15.7 STEP7 でのプログラミングにおける役立つポイント

一般事項

STEP7 での複雑なマシニングシーケンスのプログラミングにおける役に立つポイントを，以下に述べます。これは，主にデータタイプ POINTER および ANY の取扱いについての情報です。データタイプ POINTER および ANY の構造についての詳細は，STEP 7 取扱い説明書 "Designing user programs" にあるセクション "CPU register and storage of data" を参照してください。

■ データのコピー

以下の例は，1 つの DB から別の DB へ高速でデータをコピーする方法です。

```
// DB xx.[AR1] がソース
// DI yy.[AR2] が宛先
    AUF         DB 100;     // ソース DB
    LAR1        P#20.0;     // データバイト 20 でのソーススタートアドレス
    AUF         DI 101;     // 宛先 DB
    LAR2        P#50.0;     // データバイト 50 での宛先スタートアドレス
//AR1, AR2, DB, DI をあらかじめロード
    L       42;         //84 バイト転送
F001:
    L       DBW [AR1,P#0.0];    // ワード毎にコピー
    T       DIW [AR2,P#0.0];
    +AR1        P#2.0;
    +AR2        P#2.0;
    TAK;
    LOOP        F001;
```

■ ANY および POINTER

一般事項

以下のプログラミングの例は，異なるプログラミングのメカニズムを説明したものです。これらは，FC または FB 内のデータタイプ "POINTER" あるいは "ANY" によって，入出力および変数遷移 (VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT) がどのようにアクセスするのかを実証しています。アクセスオペレーションは，プログラミングに使用することのできる部分的記号メソッドで説明してあります。

■ POINTER および ANY がパラメータとして有効な場合の，FC にある POINTER および ANY の使用

機能の説明

FC 99 には，POINTER または ANY として定義された入力パラメータがあります。この例は，POINTER または ANY のパートコンポーネントにアクセスさせるルートプログラムを示しています。この場合，POINTER または ANY でパラメータに設定された DB はオープンとなり，アドレスオフセットは，クロスエリアポインタとしてアドレスレジスタ AR1 に保存されます。このように，POINTER, ANY を介してアドレス指定された変数のデータ要素（通常は，構造および配列）へのアクセスが可能になります。このアクセス動作は，この例の中では，関連プログラムシーケンスの最後で説明されています。データタイプ ANY では，データタイプおよびそのデータ要素の数に基づいて変数をアクセスした時，チェックあるいは分岐の実行も可能になります。

```
FUNCTION FC 99: VOID
VAR_INPUT
    Row : BYTE ;
    Convert : BOOL ;     // 数値変換起動
    Addr : POINTER ;     // 変数を指定
Addr1 : ANY;
END_VAR
VAR_TEMP
    dbchr : WORD ;
    Number : WORD ;
    Type : BYTE;
END_VAR
BEGIN
NETWORK
TITLE =
//POINTER
    L P##Addr;
    LAR1 ;                          // ポインタをフェッチ
    L W [AR1,P#0.0];//DB 番号をフェッチ
    T #dbchr;
    L D [AR1,P#2.0]; // ポインタの一部をオフセット
    LAR1 ;
    AUF DB [#dbchr]; // 変数の DB をオープン
    L B [AR1,P#40.0];               // アドレスオフセット 40 のポインタを介してバイト値をフェッチ
//ANY
    L               P##Addr1;
    LAR1 ; //ANY をフェッチ
    L       B [AR1,P#1.0];          // フェッチタイプ
    T #typ;
    L       W [AR1,P#2.0];          // フェッチ番号
    T       #Anzahl;
```

```
    L     W [AR1,P#4.0];          // フェッチ DB 番号
    T #dbchr;
    L     D [AR1,P#6.0];          // ポインタの一部をオフセット
    LAR1 ;
    AUF DB [#dbchr]; // 変数の DB をオープン
    L     B [AR1,P#0.0];          //ANY を介してバイト値をフェッチ
```

■ POINTER および ANY がパラメータとして有効な場合の，FC にある POINTER および ANY の使用

機能の説明

FC 99 には，POINTER または ANY として定義された入力パラメータがあります。この例は，POINTER または ANY のパートコンポーネントにアクセスさせるルートプログラムを示しています。この場合，POINTER または ANY でパラメータに設定された DB はオープンとなり，アドレスオフセットは，クロスエリアポインタとしてアドレスレジスタ AR1 に保存されます。このように，POINTER, ANY を介してアドレス指定された変数のデータ要素（通常は，構造および配列）へのアクセスが可能になります。このアクセス動作は，この例の中では，関連プログラムシーケンスの最後で説明されています。データタイプ ANY では，データタイプおよびそのデータ要素の数に基づいて変数をアクセスした時，チェックあるいは分岐の実行も可能になります。

```
FUNCTION BLOCK FB 99
    VAR_INPUT
        Row : BYTE ;
        Convert : BOOL ;                    // 数値変換起動
        Addr : POINTER ;                    // 変数を指定
        Addr1 : ANY;
    END_VAR
    VAR_TEMP
        dbchr : WORD ;
        Number : WORD ;
        Type : BYTE;
    END_VAR
BEGIN
NETWORK
TITLE =
//POINTER
    L     P##Addr;
    LAR1 ;                              // インスタンス DB からのポインタをフェッチ
    L     DIW [AR1,P#0.0]; //DB 番号をフェッチ
    T #dbchr;
    L     DID [AR1,P#2.0];              // ポインタの一部をフェッチ
    LAR1 ;
    AUF DB [#dbchr];                    // 変数の DB をオープン
    L     B [AR1,P#40.0];               // アドレスオフセット 40 のポインタを介してバイ
```

```
                                    ト値をフェッチ
//ANY
    L     P##Addr1;
    LAR1 ;                          // インスタンス DB からの ANY をフェッチ
    L      DIB [AR1,P#1.0];         // フェッチタイプ
    T      #typ;
    L      DIW [AR1,P#2.0];         // フェッチ番号
    T     #Anzahl;
    L     DIW [AR1,P#4.0];          // フェッチ DB 番号
    T     #dbchr;
    L     DID [AR1,P#6.0];          // ポインタの一部をオフセット
    LAR1 ;
    AUF DB [#dbchr];                // 変数の DB をオープン
    L     B [AR1,P#0.0];            //ANY を介してバイト値をフェッチ
```

■ FC または FB への転送用の POINTER あるいは ANY 変数

STEP7 のバージョン 1 以降では，VAR_TEMP において POINTER または ANY を定義することができます。以下の 2 つの例は，ANY がどのように提供されるのかを示しています。

いくつかの ANY パラメータは，FB (FC) において定義されます。他の FB (FC) へ転送するために，特定の ANY パラメータを選択リストから選んでください。これは，VAR_TEMP の ANY を使用してのみ行うことができます。Addr1 から Addr4 を選択するために，1 から 4 をパラメータ WhichAny にセットします。

（注）アドレスレジスタ AR2 は，ブロック内で使用します。ただし，このレジスタは，マルチインスタンス DB にも使用します。何故ならば，関連 FB は，マルチインスタンス DB として宣言することができないからです。

```
FUNCTIONBLOCK FB 100
    CODE_VERSION1               // バージョン 2 以降では，マルチインスタンス
                                   DB を停止
    STEP7
    VAR_INPUT
        WhichAny : INT;
        Addr1 : ANY;        // あらかじめ決められた命令に従う
        Addr2 : ANY;
        Addr3 : ANY;
        Addr4 : ANY;
    END_VAR
    VAR_TEMP
        dbchr : WORD ;
        Number : WORD ;
        Type : BYTE;
        Temp_addr : ANY;
    END_VAR
BEGIN
```

```
NETWORK
TITLE =
   L    WhichAny;
   DEC 1;
   L    P#10.0;                    //ANY 毎に 10 バイト
   *I;
   LAR2;
   L    P##Addr1;
   +AR2;                           //ANY のスタートアドレスを追加
   L    P##Temp_addr;
   LAR1 ;                          //VAR_TEMP からのポインタをフェッチ
   L    DID [AR2,P#0.0];           // ポインタ値を VAR_TEM へ転送
   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 FB 101, DB 100
        (ANYPAR := #Temp_addr);              //ANYPAR はデータタイプ ANY
```

2. すでにコンパイルしてあるパラメータ ANY を，別の FB (FC) へ転送してください。これは，VAR_TEMP に保存されている ANY を使用してのみ行うことができます。

```
FUNCTIONBLOCK FB 100
   VAR_INPUT
      DB Number: INT;
      DBOffset : INT;
      Data type: INT;
      Number: INT;
   END_VAR
   VAR_TEMP
      dbchr : WORD ;
      Temp_addr : ANY;
   END_VAR
BEGIN
NETWORK
TITLE =
   L   P##Temp_addr;
   LAR1 ;                          //VAR_TEMP からのポインタをフェッチ
   L   B#16#10;                    //ANY 識別子
   T   LB [AR1,P#0.0];
   L   Data type;
   T   LB [AR1,P#1.0];
   L   Number;
   T   LW [AR1,P#2.0];
   L   DB Number;
   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 FB 101, DB 100
     (ANYPAR := #Temp_addr);              //ANYPAR はデータタイプ ANY
```

■ マルチインスタンス DB

STEP7 のバージョン 2 以降では , FB はマルチインスタンス機能があります。すなわち，FB はマルチインスタンス DB を組込んでいます。マルチインスタンス DB の最も重要な特性は，FB の様々なインスタンスに使用できることで（STEP 7 取扱い説明書を参照），それゆえに DB データ量をふやすことができます。しかしながら，このタイプの FB のルーチンおよびコードサイズは通常のインスタンス DB と比べて大きいので，実際に使用するという時にのみ，マルチインスタンス DB を起動してください。

（注）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 FB において，複雑なプログラムを実行する時，ある一定のルールに従うことは，プログラマーにとって重要です。

マルチインスタンス DB では，変数のスタートアドレス (VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR) を，DI データブロックレジスタおよびアドレスレジスタ AR2 とともに転送します。マルチインスタンス FB 内で変数をアクセスする時，コンパイラがアドレスレジスタ AR2 を介してアクセスオペレーションを単独で制御します。しかしながら，複雑なプログラムセクションも同じ FB においてアドレスレジスタと共に動作しなければならない時（例えば，データのコピー），レジスタを変更する前に，AR2 の古い内容をセーブしてください。インスタンス変数 (VAR_INPUT, VAR_OUTPUT, VAR_IN_OUT, VAR) をアクセスする前に，AR2 の内容をそれらのオリジナルステータスへ再保存してください。インスタンス FB の AR2 の内容は，ローカル変数（VAR_TEMP) へセーブすることが 1 番良いでしょう。コマンド "Load pointer to an instance variable"（インスタンス変数へのロードポインタ）で，インスタンスデータの初めの部分でスタートするポインタ値を提供します。この変数を，ポインタを介してアクセスするには，AR2 に保存されたオフセットを追加してください。

例

```
FUNCTION_BLOCK FB 99
VAR_INPUT
   varin: INT;
END_VAR
VAR
   variable1: ARRAY[0..9] OF INT;
   variable2: INT;
END_VAR
```

```
BEGIN
    L P##variable1;                 //ARRAY の最初へのポインタ
                                    //8500 0010 がアキュムレータに現在保存されている
    //AR2 には，クロスエリアポインタもある。クロスエリアアクセスを使用する場合，これらの 2 つのポインタを追加する時に，エリアをマスク処理してください。
    UD DW#16#00FF_FFFF;         // エリアをマスク処理
    LAR1 //AR1 へロード
    TAR2;
    +AR1 AR2;                       //AR2 インスタンスオフセットを追加
    //AR1 を介して間接的にアクセスすることのできる変数の配列
    L DIW [AR1, P#0.0];             // 例えば，最初の要素へアクセス
END_FUNCTION_BLOCK
```

■ 文字列

一般事項

データタイプ STRING は，基本プログラムのある一定のサービスにおいて必要となります。パラメータ割当てのための文字列構造および取扱い手順に関するいくつかの追加事項を以下に示します。

STRING の構造

STRING タイプのデータは，通常データブロックに保存（定義されて）します。文字列の定義の方法は 2 通りあります。

1. データタイプ STRING のみを変数に割当てます。STEP7 コンパイラは，自動的に 254 文字の長さを生成します。
2. データタイプ STRING は，矩形の括弧に入れた文字列（例えば，[32] ）とともに変数に割当てます。この方法で，STEP7 コンパイラは，入力に応じた長さの文字列を生成します。

STRING データタイプの変数用に，定義により定められたものより 2 バイト多く常に保存します。STEP7 コンパイラは，最初のバイトにおいて可能な限りの最大数の文字を保存します。2 番目のバイトには，実際に使用する文字の数があります。文字（1 文字に付き 1 バイト）は，3 番目のバイトから前方へ保存します。

文字列パラメータは，通常 POINTER あるいは ANY を使用して，基本プログラムのブロックへ割当てます。このような割当ては通常，記号的プログラミングメソッドを使用して作り出してください。パラメータ化した文字列を含むデータブロックは，記号リストに保存してください。文字列変数のフルストップおよび記号的名前に従った記号データブロック名を使用して，基本プログラムブロックを作成します。

■ データブロック構造体用オフセットアドレスの確定

一般情報

構造化 DB 内でオフセットアドレスの記号確定において問題が発生します。例えば，Array あるいは Structure は，DB 内のいずれかの場所に保存されます。スタートアドレスで記号的にアドレスレジスタをロードした後，アドレスレジスタを介して Array あるいは Structure の個々の要素にアクセスすることでしょう。アドレスレジスタを記号的にロードする方法の 1 つは，その入力パラメータがポインタとなる FC を使用することです。Array あるいは Structure のアドレスは，プログラムの中でこの FC の入力パラメータへ記号で割当てられます。FC にあるプログラムコードが入力パラメータからオフセットアドレスを確定して，アドレスレジスタ（AR1）にあるオフセットアドレスを呼出し機能へ送ります。このように間接的な記号アドレス指定でも，可能となっています。

```
FUNCTION FC 99: VOID
VAR_INPUT
Addr : POINTER ; // 変数を指定
END_VAR
BEGIN
NETWORK
TITLE =
    L P##Addr;
    LAR1 ;                          // アドレスからのポインタをフェッチ
    L D [AR1,P#2.0];                // 変数ポインタのオフセットパート
    LAR1 ;
END_FUNCTION
```

1.15.8 マシンデータ，アラーム

■ マシンデータ

■ 一般マシンデータ

10100 MD 番号	PLC_CYCLIC_TIMEOUT 周期的 PLC 監視時間	
初期設定 : 0.1	最小入力リミット : 0	最大入力リミット : plus
変更を有効にするための条件 : POWER ON	保護レベル : 2 / 7	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : SW1	
意味 :	周期的 PLC 監視時間 このマシンデータは，PLC がそのライフ信号を実行した後の，最大監視時間を定義します。目盛り付きスケールは，補間周期で内部的に実行します。	

14504 MD 番号	MAXNUM_USER_DATA_INT 周期的 PLC 監視時間 (INT)	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 256
変更を有効にするための条件 : 電源オン	保護レベル : 2 / 7	単位 : -
データタイプ : DWORD	適用開始 SW バージョン : 4.1	
意味 :	ユーザマシンデータの最大量（整数フォーマットで表示）	
関連性	MD 14506, 14508	

14506 MD 番号	MAXNUM_USER_DATA_HEX Number of user data (HEX)	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 256
変更を有効にするための条件 : POWER ON	保護レベル : 2 / 7	単位 : -
データタイプ : DWORD	適用開始 SW バージョン : 4.1	
意味 :	ユーザマシンデータの最大量（16 進フォーマットで表示）	
関連性	MD 14504, 14508	

14508 MD 番号	MAXNUM_USER_DATA_FLOAT Number of user data (FLOAT)	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 32
変更を有効にするための条件 : 電源オン	保護レベル : 2 / 7	単位 : -
データタイプ : DWORD	適用開始 SW バージョン : 4.1	
意味 :	IEEE フォーマットでのユーザマシンデータの最大量 (浮動小数点フォーマットで表示)	
関連性	MD 14504, 14506	

14510 MD 番号	USER_DATA_INT[n] ユーザデータ (INT)	
初期設定 : 0	最小入力リミット : -32768	最大入力リミット : 32767
変更を有効にするための条件 : 電源オン	保護レベル : 2 / 7	単位 : -
データタイプ : 1)	適用開始 SW バージョン : 4.1	
意味 :	ユーザマシンデータ (整数フォーマットで表示)	
関連性	MD 14512, 14514	

14512 MD 番号	USER_DATA_HEX[n] ユーザデータ (HEX)	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 0x0FF
変更を有効にするための条件 : 電源オン	保護レベル : 2 / 7	単位 : -
データタイプ : 1)	適用開始 SW バージョン : 4.1	
意味 :	ユーザマシンデータ (16 進フォーマットで表示) 1)	
関連性	MD 14510, 14514	

14514 MD 番号	USER_DATA_FLOAT[n] ユーザマシンデータ (FLOAT)	
初期設定 : 0	最小入力リミット : -3.40e38	最大入力リミット : 3.40e38
変更を有効にするための条件 : 電源オン	保護レベル : 2 / 7	単位 : -
データタイプ : 2)	適用開始 SW バージョン : 4.1	
意味 :	:IEEE フォーマットでのユーザマシンデータ (浮動小数点フォーマット)	
関連性	MD 14510, 14512	

1) 整数／ 16 進フォーマットのマシンデータは，NCK の DWORD と同じように扱います。データは，NCK-PLC インタフェースに保存され，PLC の電源投入の間，PLC ユーザで DB 20 から読取ることができます。

2) 浮動少数フォーマットのマシンデータは，NCK の FLOAT（4 バイト IEEE）と同じように扱います。データは，NCK-PLC インタフェースに保存され，PLC の電源投入の間，PLC ユーザで DB 20 から読取ることができます。

■ チャンネル別マシンデータ

28150 MD 番号	MM_NUM_VDIVAR_ELEMENTS PLC 変数書込み用の要素の番号	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : plus
変更を有効にするための条件 : 電源オン	保護レベル : 2 / 7	単位 : -
データタイプ : DWORD	適用開始 SW バージョン : 4.1	
意味 :	この MD は，ユーザが PLC 変数をプログラミング（$A_OUT_DBx）できるようにするための要素の番号を定義します。この番号は，ブロックサーチにも適用しますが，動作には同期しません。それぞれの要素のメモリスペースは，約 24 バイトです。PLC 変数を高速で順次書込みをするためのそれぞれの書き込み命令を実行するために，1 つの要素が必要となります。	

■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

1.16 原点復帰 (R1)

1.16.1 概略説明

■ 概略説明

マシン原点

工作機械のマシン原点はマシン軸の実際位置が "0" である点として定義されます。

原点復帰

マシン原点は，NC 制御装置に接続された位置測定系および原点復帰機能（レファレンシング機能とも呼ばれる）によって軸別に NC 制御装置に伝達されます。

位置測定系

下記の位置測定系を制御装置に接続することができます：

相対回転測定系

相対直線測定系

距離コード化基準マーク（Heidenhain 社製）を使用した直線測定系

絶対回転測定系

絶対直線測定系

1.16.2 詳細説明

基準化の必要性

電源投入後に正確なマシン原点の検出ができるように，制御装置を各マシン軸の位置測定系と同期させなければなりません。このプロセスを基準化と言います。

位置測定系

下記の位置測定系を制御装置に接続することができます：

- 相対回転測定系
- 相対直線測定系
- 距離コード化基準マーク（Heidenhain 社製）を使用した直線測定系
- 絶対回転測定系
- 絶対直線測定系

接続された位置測定系は MD 34200: ENC_REFP_MODE（基準化モード）を使用して下記の 8 つの基準化グループに細分することができます：

- 絶対エンコーダを使用した基準化（MD 34100: REFP_SET_POS を受けつける）（SW2.2 以降）
- 相対測定系を使用した基準化
- 距離コード化基準マーク付きの直線測定系を使用した基準化
- BERO および 1 回の信号エッジ検出を使用した基準化
- BERO が基準カムの代わりをする基準化
- すでに基準化されたエンコーダに合わせて測定系を校正する基準化（ SW3.2 以降，NCU570 以外）
- スピンドルアプリケーション用に構成されたアプローチ速度での BERO を使用した基準化（SW3.6 以降）

カム

カムには次のようなタスクがあります：

- ゼロマークにアプローチする際の移動の方向を選択する
- 必要となる場所でゼロマークを選択する

REF アクティブマシンファンクション DB11, ...DBX5.2

原点復帰は REF マシンファンクションを起動させる（インタフェース信号 "REF active machine function"(DB11, ... DBX5.2)）ことによって行います。REF マシンファンクションの選択は，JOG モードおよび MDA モードで，インタフェース信号 "REF machine function " (DB11, ... DBX1.2) で行います。

軸別基準化

インタフェース信号 "plus/minus travel keys"（DB31, ... DBX4.7 および 4.6）で，各マシン軸ごとに別々に軸別基準化を開始します。8 軸まで同時に基準化できます。マシン軸を特定の順番で基準化しなければならない場合，下記のオプションが利用できます：

- オペレータがスタートアップの正しい順番を守る。
- PLC ユーザプログラムがスタートアップ時に順番を調べるか，または順番そのものを定義する。
- 順番を MD 34110: REFP_CYCLE_NR（チャンネル別基準化を参照）に定義する。

チャンネル別基準化

インタフェース信号 "activate referencing" (DB21, ... DBX) でチャンネル別基準化がスタートします。制御装置は，インタフェース信号 " referencing active" (DB21, ... DBX33.0) によって，スタートが成功したことを確認応答します。チャンネルに割当てられた各マシン軸はチャンネル別基準化を使用して基準化できます（プラス／マイナストラベルキーをシュミレートすることによって，制御装置で内部的に行われます)。軸別 MD 34110: REFP_CYCLE_NR（チャンネル別基準化の軸の順番）を使用してマシン軸を基準化するときの順番を定義できます。
REFP_CYCLE_NR に入力された全軸がそれぞれの終点に到達したら，インタフェース信号 "all axes referenced" (DB21, ... DBX36.2) がイネーブルされます。

特別機能

- 軸別基準化とチャンネル別基準化は相互排他的ではありません。
- インタフェース信号 "reset" (DB21, ... DBX35.7 / DB11, ... DBX0.7) で基準化をキャンセルできます。この信号が出された時点で基準点に到達していない軸は基準化されません。インタフェース信号 "referencing active" がリセットされ，アラーム 20005 が出力されます。
- 下記のリミットモニタは基準化されたマシン軸に対して非アクティブです：
  - 作業エリア限界
  - ソフトウェアリミットスイッチ
  - プロテクションゾーン
- 基準化中は（アラームが発生した時以外)，定義された軸別に観測が行なわれます。
- 原点復帰は，MD 34010: REFP_CAM_DIR_IS_MINUS に記録されている方向の方向キーのみを使用してスタートさせることができます（例外：距離コード化測定系の場合は，どちらの方向キーを使用しても原点復帰をスタートさせることができます)。

■ 相対測定系を使用した基準化

基準化の順番

相対測定系用の基準化の順番は次の 3 つのフェーズに分けることができます：

フェーズ 1：基準カムへの移動

フェーズ 2：ゼロマークとの同期

フェーズ 3：原点への移動

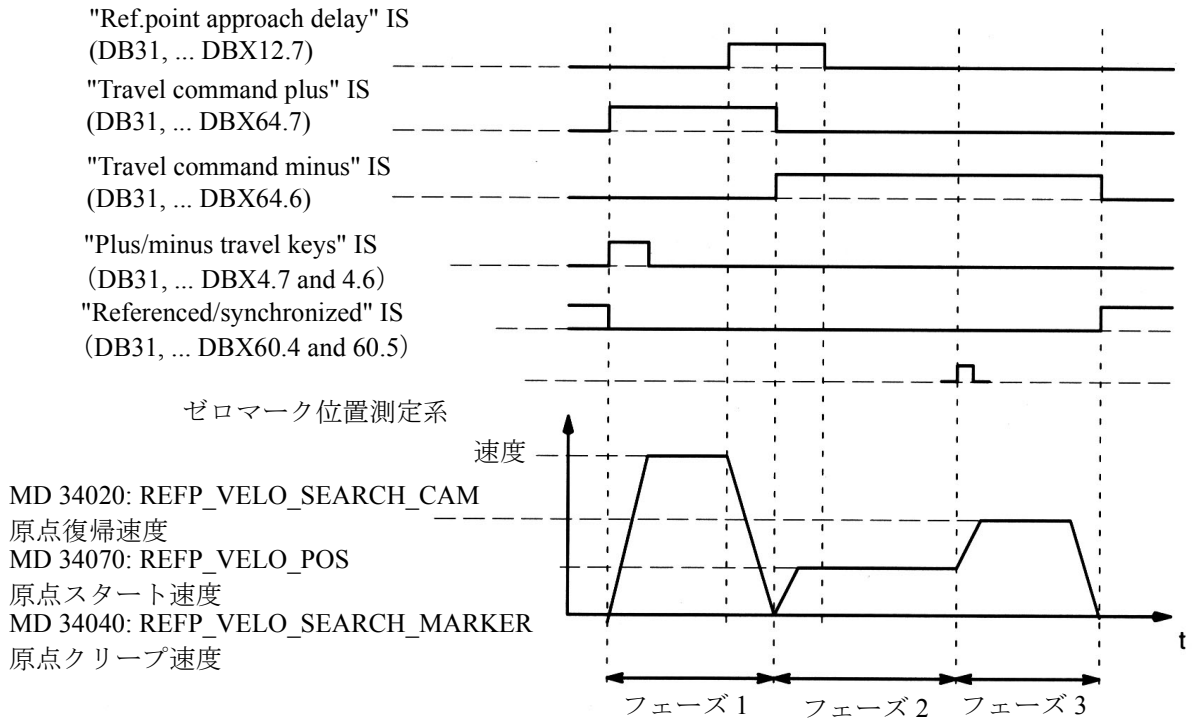


図 1.137 相対測定系を使用した基準化の手順（例）

■ フェーズ 1：基準カムへの移動

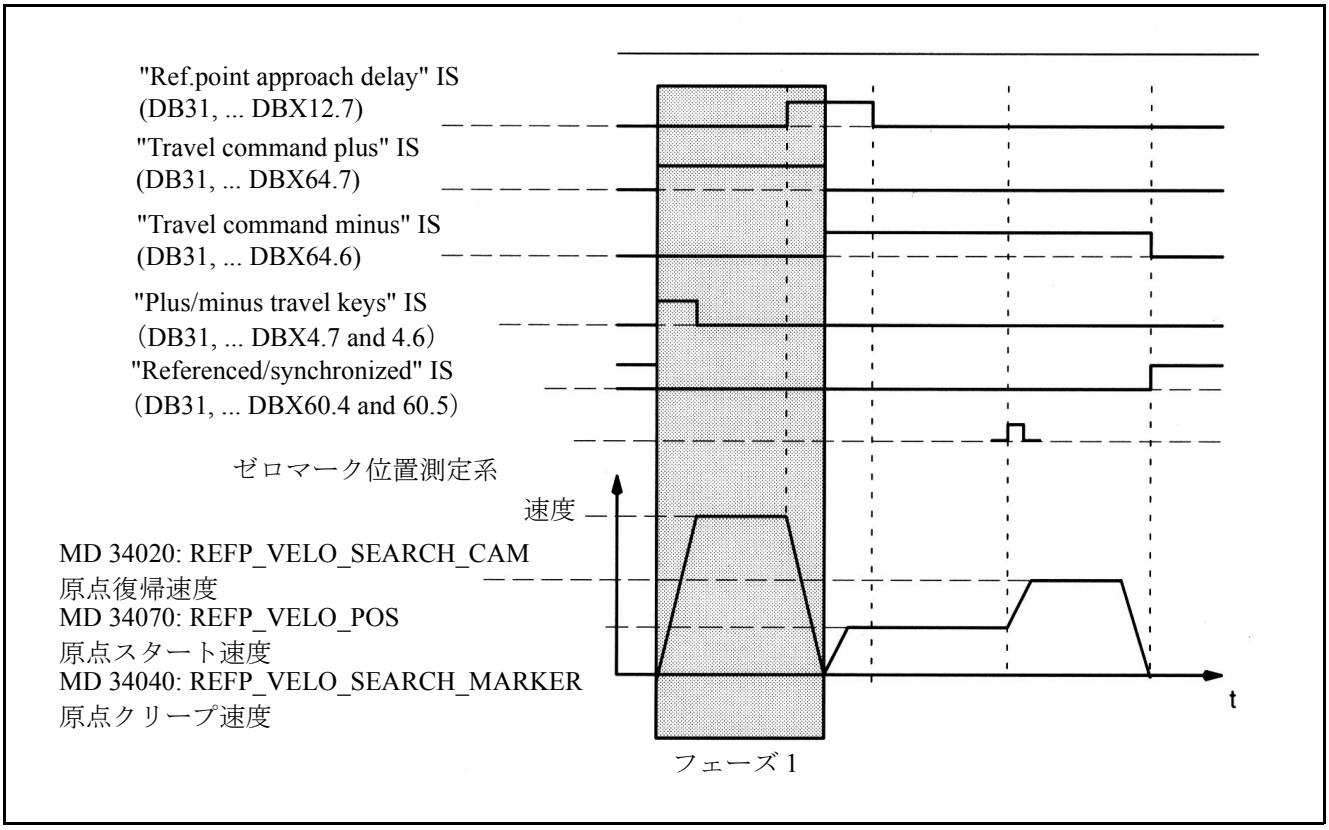


図 1.138 フェーズ 1：基準カムへの移動

フェーズ 1 のスタート

フェーズ 1 はインタフェース信号 " Plus/minus travel keys"（DB31 から 48，DBX4.7 および 4.6）（軸別基準化にだけ適用される）でスタートします。MD 11300: JOG_INC_MODE_LEVELTRIGGRD（JOG ／連続モードの INC/REF）は，フェーズ 1 ～ 3 の間，プラス／マイナストラベルキーを押したままにしておくか，それとも 1 回のキーストロークがフェーズ 1 ～ 3 を終了させるのに十分であるかを定義します。インタフェース信号 "referenced/synchronized 1 or 2"（DB31, ... DBX60.4 および 60.5）およびインタフェース信号 "all axes referenced" (DB 21-30, DBX36.2) はリセットされます。

フェーズ 1 は下記の 3 つの状態からスタートすることができます：

1. マシン軸が基準カムの前にある場合
2. マシン軸が基準カム上にある場合
3. マシン軸に基準カムがない（例えば回転軸）場合

マシン軸が基準カムの前にある場合

マシン軸は，MD 34010:REFP_CAM_DIR_IS_MINUS（マイナス方向での原点復帰）で指定された方向へ，MD 34020:REFP_VELO_SEARCH_CAM（基準カムより前の原点復帰速度）で定義された速度にまで加速します。インタフェース信号 "reference point approach delay" (DB31, ... DBX12.7) で，いつ基準カムに到達したか，どの位置でマシン軸が制動停止したかが制御装置に通知されます。この間，マシン軸は下記の式によって計算される距離だけさらに移動します：

$$\text{最小距離} = \frac{(\text{MD: REFP_VELO_SEARCH_MARKER で定義される速度})^2}{2 * (\text{MD: MAX_AX_ACCEL で定義される加速})}$$

マシン軸がフェーズ 2 で正確な基準点クリープ速度で基準カムから離れることを保証するためにこの最小距離が必要です。

フェーズ 1 は完了し，それに続いてフェーズ 2 のプロセスが行なわれます。

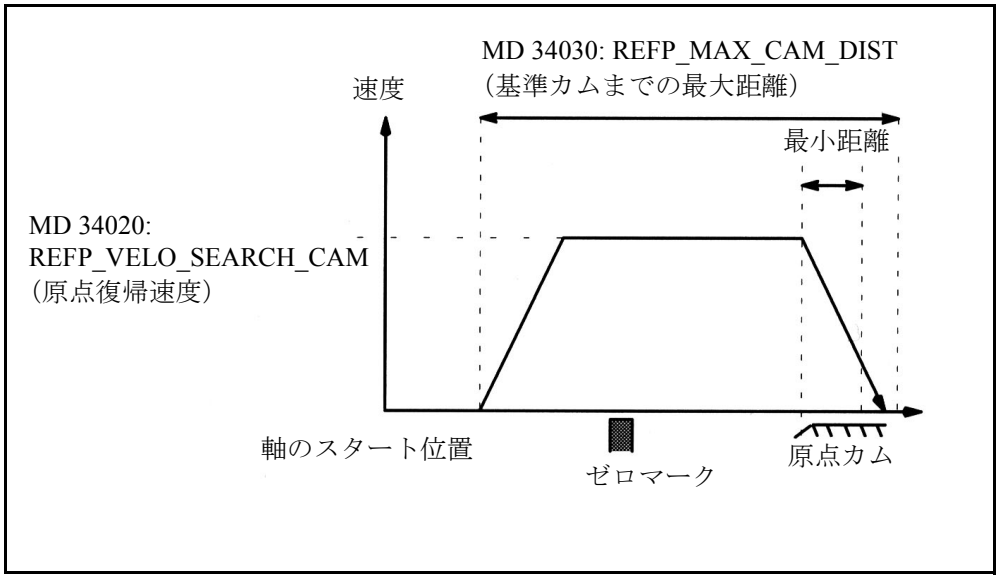


図 1.139 遅延のための最小距離

マシン軸が基準カム上にある場合

マシン軸はスタート位置にとどまります。フェーズ 1 は完了しているので，直接フェーズ 2 がスタートします。

マシン軸に基準カムがない場合

MD 34000: REFP_CAM_IS_ACTIVE（基準カムを有する軸）に "0" を入力することによって基準点カムがないマシン軸を識別します。これは，全トラバース範囲にわたってゼロマークが 1 つしかないマシン軸，または 1 回転ごとのゼロマークがが 1 つしかない回転軸の場合です。これらのマシン軸はそれぞれのスタート位置にとどまります。フェーズ 1 は完了していると見なされ，直接フェーズ 2 がスタートします。

フェーズ 1 の特徴

- フィードレイトオーバライドスイッチがアクティブです。
- フィード停止（チャンネル別および軸別）がアクティブです。
- NC 停止／NC スタートを使用してマシン軸を停止させたり再スタートさせたりできます。
- マシン軸が，スタート位置から基準カムの方向に最大距離（MD 34030:REFP_MAX_CAM_DIST で定義）だけ移動しても基準カムに到達しなかった場合（インタフェース信号 "reference point approach delay" (DB31, ... DBX12.7) は出力されない），軸は停止し，アラーム 20000 "reference cam not reached" が出力されます。

■ フェーズ 2: ゼロマークとの同期

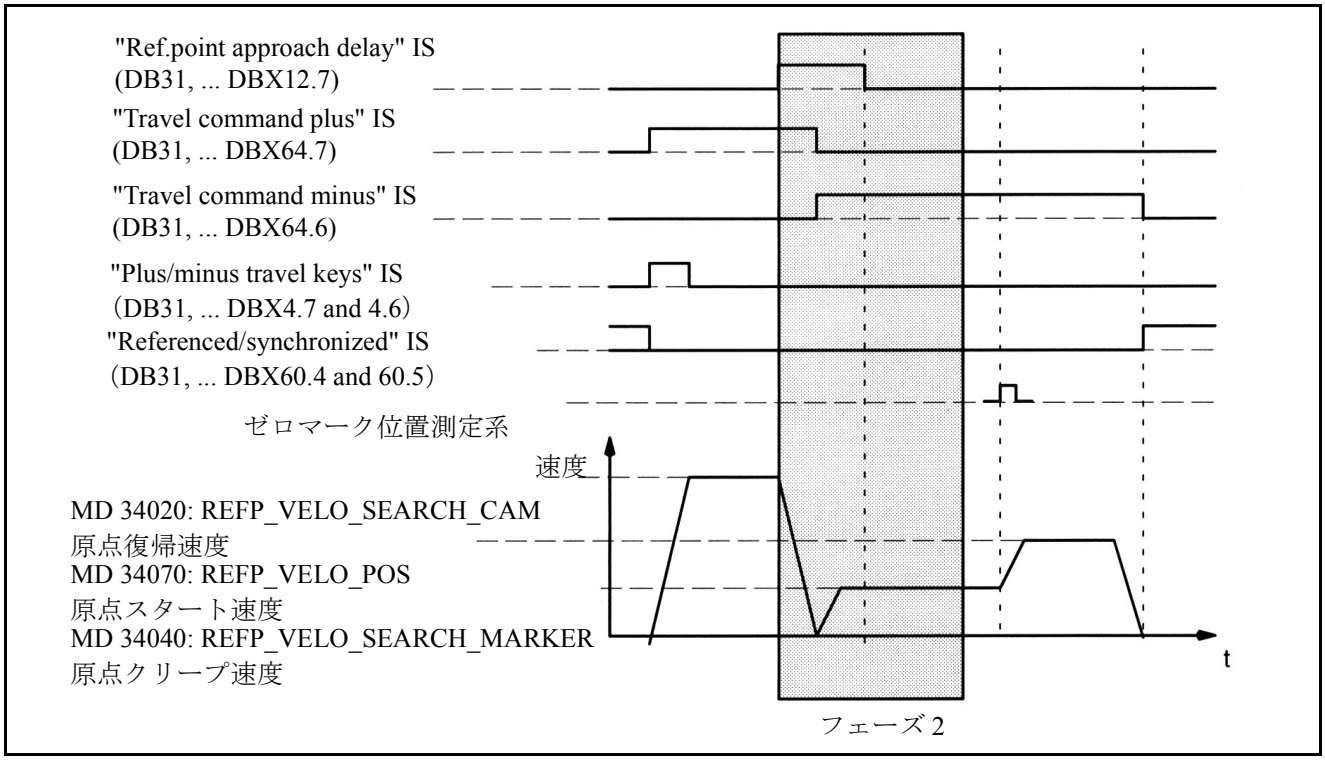


図 1.140 フェーズ 2: ゼロマークとの同期

フェーズ 2 のスタート

フェーズ 1 が正常に完了し（アラームが出ない），マシン軸が基準カム上にあればフェーズ 2 が自動的にスタートします。基準カム上になければ（インタフェース信号 "reference point approach delay" (DB31, ... DBX12.7) がリセットされている），アラーム 20001 "No cam signal exists" が出力されます。フェーズ 1 での減速中に，基準カムが短すぎてマシン軸がそれを超えて移動すると，アラーム 20001 が発生する可能性があります。

同期のタイプ

MD 34050: REFP_SEARCH_MARKER_REVERSE を使用してゼロマークのサーチ方向を定義します：

- 基準カム信号の立下がりエッジとの同期
- 基準カム信号の立上がりエッジとの同期

実際速度がフェーズ 2 用のセットポイント（MD 35150: SPIND_DES_VELO_TOL からの公差）に達することがなければ，フェーズ 1 が繰返されます。このことは，スタート時に軸が丁度カム上にあった場合に起こります。

基準カム信号の立下がりエッジとの同期

マシン軸は，MD 34010:REFP_CAM_DIR_IS_MINUS（マイナス方向での原点復帰）で指定された方向とは反対の方向へ，MD 34040:REFP_VELO_SEARCH_MARKER（原点クリープ速度）で定義された速度にまで加速します。軸が基準カムを離れる（インタフェース信号 "reference point approach delay" (DB31, ... DBX12.7) がリセットされる）と，制御装置は最初のゼロマークと同期化されます。マシン軸は一定の速度で移動し続けます。フェーズ 2 は完了し，それに続いてフェーズ 3 がスタートします。

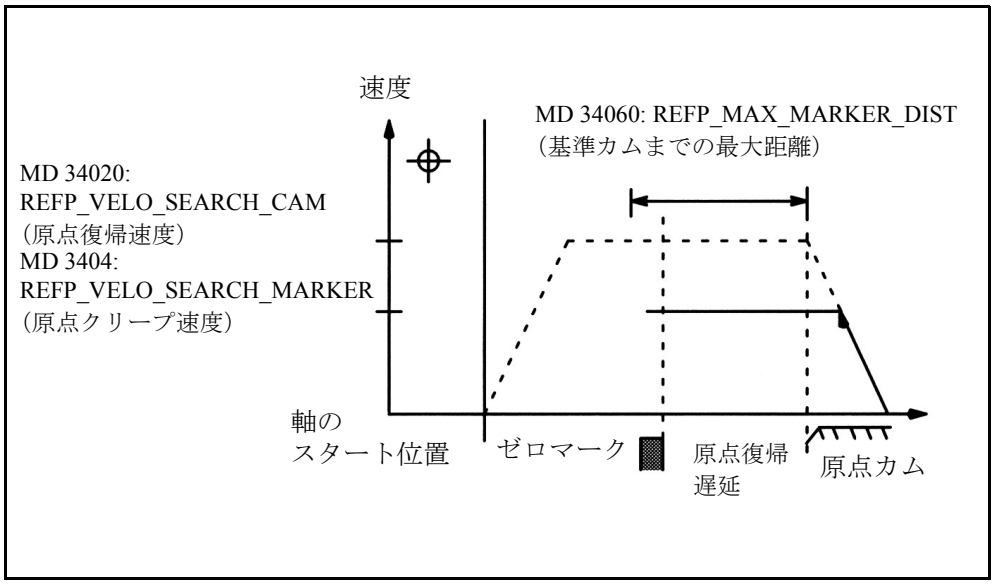


図 1.141 基準カム信号の立下がりエッジとの同期

基準カム信号の立上がりエッジとの同期

マシン軸は，MD 34010:REFP_CAM_DIR_IS_MINUS（マイナス方向の基準点へのアプローチ）で指定された方向とは反対の方向へ，MD 34020:REFP_VELO_SEARCH_CAM（原点復帰速度）で定義された速度にまで加速します。軸が基準カムを離れる（インタフェース信号 "reference point approach delay" がリセットされる）と，マシン軸は減速停止し，MD 34040:REFP_VELO_SEARCH_MARKER（原点クリープ速度）で定義された速度にまで反対の方向に加速し，その速度で基準カムに向かいます。基準カムに到達すると（インタフェース信号 "reference point approach delay" がイネーブルされ），制御装置は最初のゼロマークと同期化されます。マシン軸は定速でそのまま移動し続けます。

MD 34050:REFP_SEARCH_MARKER_REVERSE がイネーブルになると，カムも REFP_VELO_SEARCH_CAM で定義された速度でアプローチされます。この設定はより長いカムに対して有効です。フェーズ 2 は完了し，それに続いてフェーズ 3 がスタートします。

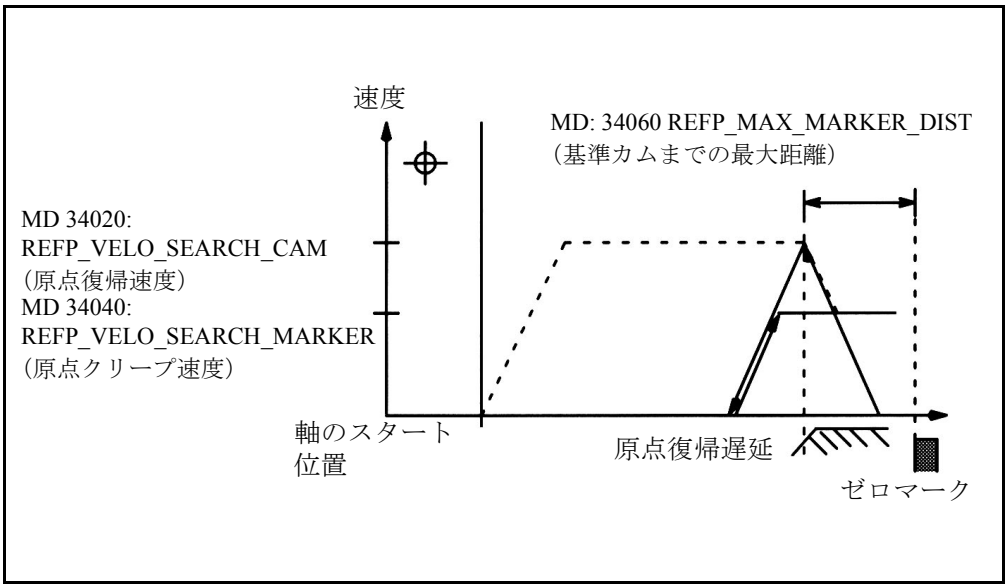


図 1.142 基準カム信号の立上がりエッジとの同期

基準カムの位置合わせ

位置測定系に周期的に繰返される複数のゼロマークがある場合（モータに接続された相対回転間接位置測定系内のような）は，基準カムが正確に位置合わせされていなければなりません。基準カムが正確に位置合わせされていなければ，誤ったゼロマークが算出されるか，またはアラーム 20002 "zero mark missing" が出力されます。

基準カムはインタフェース信号 "reference point approach delay" によって PLC から NCK へ報告されます。基準カムを検出するときの制御装置 (NCK) の応答時間は下記の要因の影響を受けます：

- 基準カムスイッチの切換え精度
- 基準カムスイッチ（NC 接点）の遅延
- PLC 入力時の遅延
- PLC サイクルタイム
- NCK/PLC インタフェースを更新するためのサイクルタイム
- 補間サイクル
- 位置制御サイクル

同期に必要となる基準カムの信号エッジは 2 つのゼロマークの間で位置合わせされることが実算でわかっています。

警告

基準カムが正確に位置合わせされていなければ，誤ったゼロマークが算出される可能性があります。この場合，制御装置は誤ったマシン原点を想定し，軸を誤った位置に動かします。ソフトウェアリミットスイッチ，保護エリア制限および作業エリア制限は誤った位置で作動するので，マシンを保護できません。いずれの場合にも，距離はエンコーダの ± 1 回転に対応しています。

フェーズ 2 の特徴

- （フィードオーバライドスイッチの）フィードオーバライドはアクティブではありません。設定は 100% として定義されています。0% では手順がキャンセルされます。
- フィード停止（チャンネル別および軸別）がアクティブです。軸は停止し，アラーム 20005 が出力されます。
- マシン軸を NC 停止／ NC スタートで停止させたり再スタートさせたりできません。
- 基準カムを離れた（IS: "reference point approach delay" がリセットされた）後で，マシン軸が MD 34060:REFP_MAX_MARKER_DIST（基準マークまでの最大距離）で定義されたパスを移動してもゼロマークが検出されなければ，軸は停止して，アラーム 20002 "zero mark missing" が出力されます。

■ フェーズ 3：基準点への移動

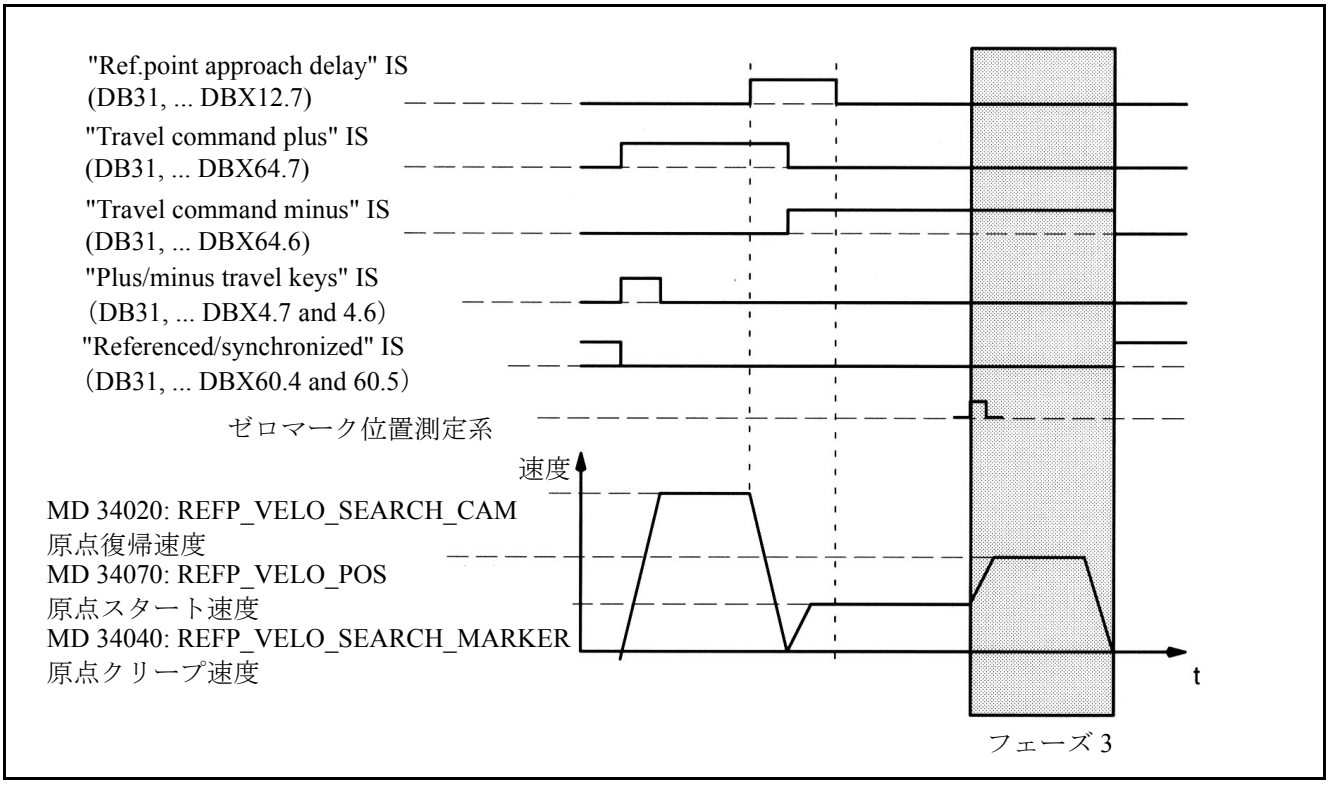


図 1.143 フェーズ 3：原点への移動

フェーズ 3 のスタート

フェーズ 2 が正常に完了して（アラームが出ない），フェーズ 3 は自動的にスタートします。マシン軸はフェーズ 2 の終わりに基準クリープ速度で移動しているので，フェーズ 3 への遷移は速度が 0 に落ちることなく行なわれます。

距離の符号に応じて，マシン軸は MD 34070: REFP_VELO_POS（原点スタート速度）で定義された速度にまで加速し，MD 34080:REFP_MOVE_DIST（距離コード化システム用の原点距離／移動先）と MD 34090: REFP_MOVE_DIST_CORR（原点オフセット）の和から計算された距離だけ移動します。この距離はフェーズ 2 で検出されたゼロマークと基準点との間の距離と正確に一致します。

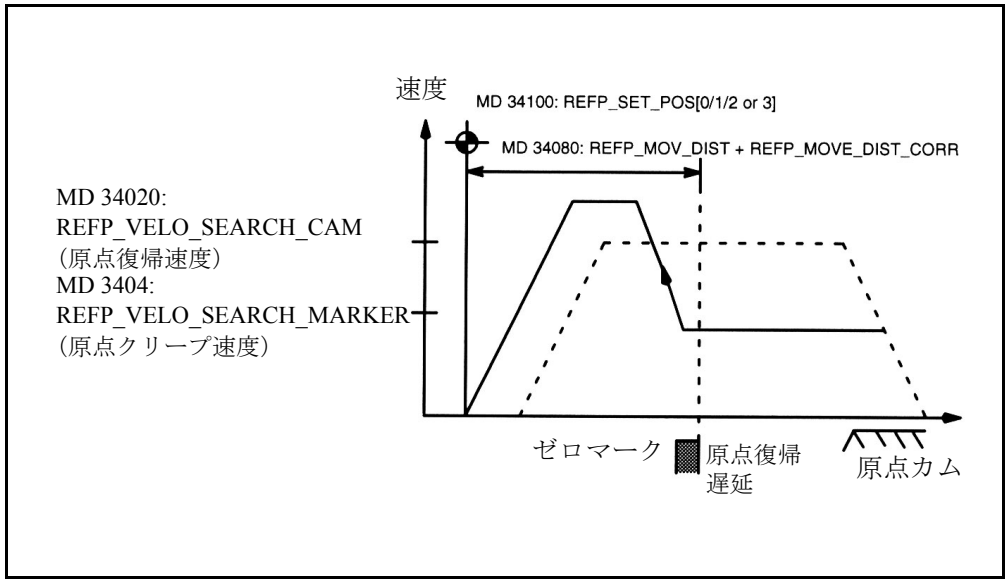


図 1.144 原点位置

原点到達

マシン軸が原点に到達したら（軸は静止），MD 34100: REFP_SET_POS（基準値）からのインタフェース信号 "Reference point value 1 to 4" (DB31, ... DBX2.4 -2.7) によって選択された基準点が , 新しい基準位置（「信号の立上がりエッジ」で選択された値）として制御装置に受入れられます。この時点でマシン軸は基準化され，インタフェース信号 "referenced/synchronized 1 or 2"（DB31, ... DBX60.4 または 60.5（どの位置測定系がアクティブになっているかによる）がイネーブルになります。

フェーズ 3 の特徴

- フィードオーバライドスイッチがアクティブです。
- フィード停止（チャンネル別および軸別）がアクティブです。
- NC 停止／ NC スタートでマシン軸を停止させたり再スタートさせたりできます。

フェーズ 3 の特別機能

MD 34080: REFP_MOVE_DIST（距離コード化システム用の原点距離／移動先）と MD 34090: REFP_MOVE_DIST_CORR（基準点オフセット）からの合計距離が，基原点クリープ速度から制止までのマシン軸の制動距離より短い場合，原点は反対方向からアプローチされます。

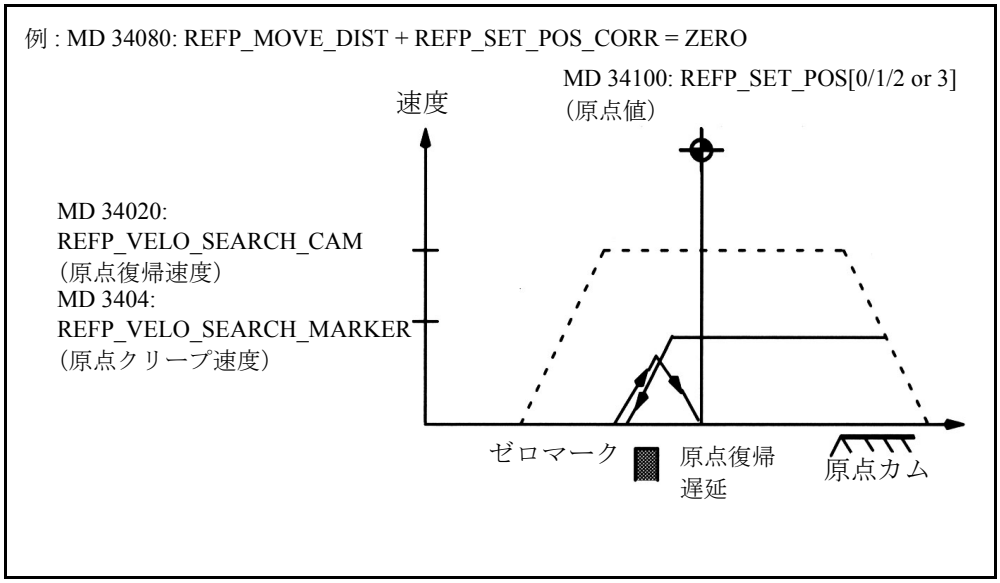


図 1.145 制動距離 > 原点距離 + 原点オフセット

■ 電源をオフにした後の相対エンコーダ用の実際値バッファ

目的

これまでは，相対エンコーダが付いている軸は，電源を投入する度，または制御装置がパーキングする度に再度原点復帰しなければなりませんでした。

SW 4.1 以降では，従来の工作機械でも，電源オフ／オン後に再度明確に基準化することなく元の位置情報によって、従来の機械工具を操作する事ができます。

条件

電源オフ／オン後も引続き元の位置情報を使用可能にするためには，その軸が電源オフ中に動かされていないことが大切です。

この目的で適当な保持ブレーキを使用することができます。

(注)・このファンクションを使用するとき，マシン設計の予防措置がなにもとられていなければ，何度か電源オフ／オンオペレーションを行なった後に，位置誤差が積重なる可能性があります。

・この機能は "exact stop fine"（正確停止（微））軸信号に常時リンクされています。この信号をサポートしていない軸またはスピンドルはこの機能を使用できません。

自動基準化

電源オフ／オン後の「自動」基準化のための前提条件は下記のとおりです：

1. 軸を制御するために使用されるエンコーダが（機械的に）校正されている。
2. 「自動」基準化が MD 34210: ENC_REFP_STATE=1 または 2 に入力されている。
3. 軸がエンコーダを使用して基準化されている。ここで MD 34210: ENC_REFP_STATE の内部的な値が 1 から 2 に変えられる。
4. 電源オフ／オン後，MD 34210: ENC_REFP_STATE=2 になる。これは，軸が電源オフ／オン前に基準化されてそのまま静止していた，つまり制御装置上では元の位置が有効で，それを内部的に使用できることを意味します。

（注）エンコーダが停止中に校正を無効にするような動作が起こらないようにするのはユーザの責任です。

エンコーダに電源投入されるとき，NC（コントローラ，補間器およびブロック処理）は保存された元の絶対値と同期化されます。同期化が完了し，スピンドルが作動し続けられるようになるまで軸動作はディスエーブルされます。

電源オン時の応答

- 軸が基準化されいて，かつ静止状態である場合 (MD 34210: ENC_REFP_STATE=2):
  - 相対エンコーダを使用した軸の従来の電源オン応答とは違って，インタフェース信号 "referenced" は自動的に現れます。軸位置は "0" に初期化されることはなく，保存された絶対値でスタートします。
  - 明確な原点復帰は必要ありませんが，もし必要なら行なっても問題ありません。
- 軸が基準化されていない，および（または）静止状態でない場合 (MD 34210: ENC_REFP_STATE=1):
  - 明確な原点復帰が必要です。軸はスタート位置 "0" に初期化されます。"axis is referenced"（軸が初期化された）ステータスは現れません。
  - MD 34210: ENC_REFP_STATE=2 は次の原点復帰によって制御装置内で内部的に設定されます（静止／正確停止）。トラバース中は（静止／正確停止なし），MD 34210: ENC_REFP_STATE は一時的に "1" に戻ります。

（注）正確停止信号を出すことができないスピンドルはステータス MD 34210: ENC_REFP_STATE=2 に到達することができません。つまりここでは「保存された実際値」が有効ではありません。

■ 距離コード化基準マーク付きの直線測定系を使用した基準化

基本概念

距離コード化基準マーク付きの直線測定系の場合，マシン軸を基準化するために，基準カムを評価したり，定義された基準点にアプローチしたりする必要はありません。このタイプの測定系は，ラインングリッドおよびこれと平行に伸びている基準マークトラックで構成されています。2 つの連続した基準マークの間のこの距離は，マシン軸の絶対位置がこの距離から求められるように可変に定義されます。

（Heidenhain 社製の）距離コード化基準マーク付きの直線測定系では，マシン軸が正の方向へ移動するか負の方向へ移動するかに関わらず，20 mm だけトラバースした後（つまり 2 つの基準マークを通過した時）に絶対位置値が得られます。

基準化の順番

基準化の順番は次の 2 つのフェーズに細分できます：

フェーズ 1: 同期させた 2 つの基準マークの通過

フェーズ 2: 固定移動先への移動

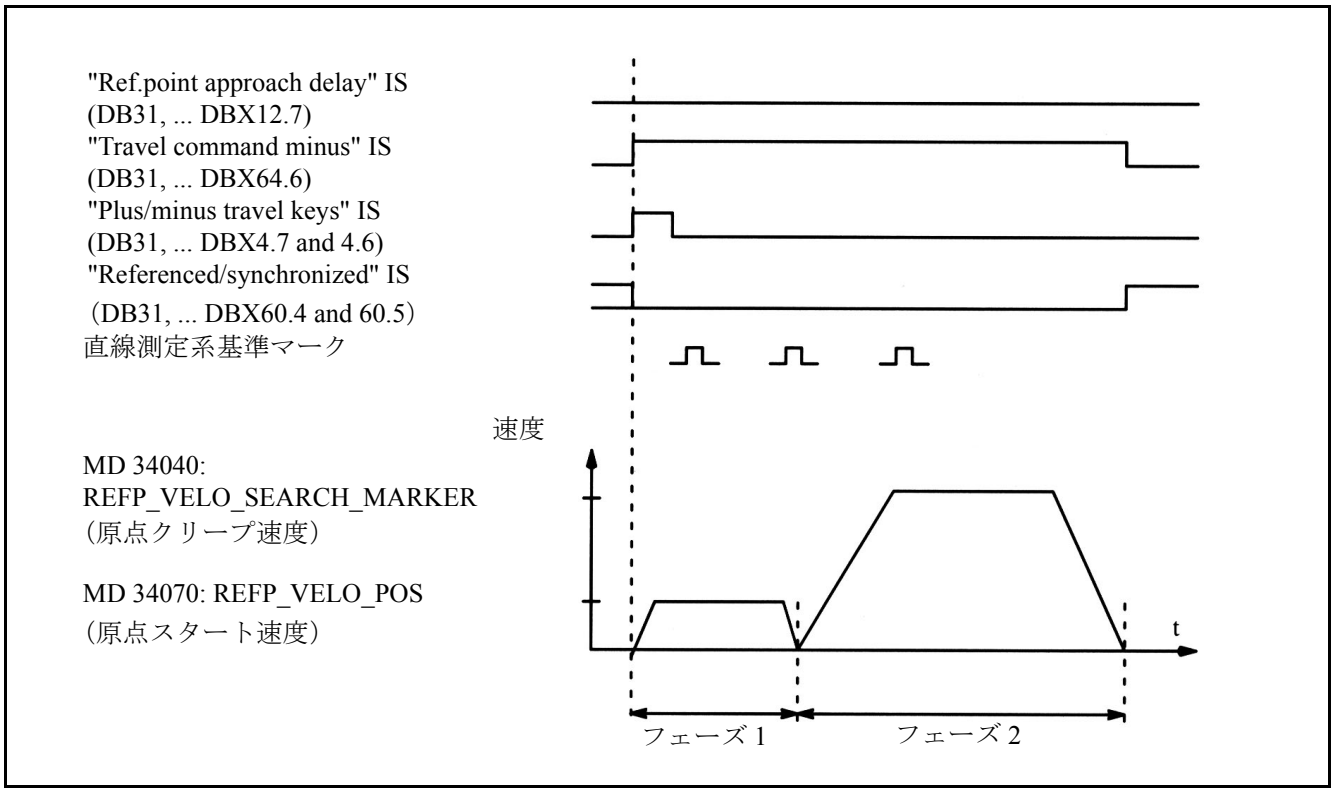


図 1.146 直線コード化基準マーク

■ フェーズ 1: 同期させた 2 つの基準マークの通過

フェーズ 1 のスタート

フェーズ 1 はインタフェース信号 " Plus/minus travel keys"（DB31, ... DBX4.7 および 4.6）（軸別基準化にだけ適用される）によって始まります。MD 11300: JOG_INC_MODE_LEVELTRIGGRD（ジョグ／連続モードの NC/REF）は，フェーズ 1 および 2 の間，プラス／マイナストラベルキーを押したままにしておくか，それとも 1 回キーを押すとフェーズ 1 および 2 がスタートし，もう 1 回キーを押すと終了するようにするかを定義します。

インタフェース信号 "referenced/ synchronized 1 or 2"（DB31, ... DBX60.4 and 60.5）およびインタフェース信号 "all axes referenced" (DB21, ... DBX36.2) はリセットされます。

基準カム

距離コード化基準マークを使用した直線測定系は基準化用の基準カムを必要としません。しかし，チャンネル別基準化およびパートプログラム内の基準化 (G74) をする間，2 つの連続した基準マークを通過しない場合に備えて，マシン軸のトラバース範囲内に基準カムをインストールする必要があります。基準カムの目的は，マシン軸がフェーズ 1 のスタート時に基準カム上に位置している時に，MD 34010: REFP_CAM_DIR_IS_MINUS（マイナス方向への原点復帰）で指定された方向とは反対のトラベル方向を選択することです（この選択はプラス／マイナストラベルキーをオーバライドします)。その結果，軸はカムから離れる方向に移動します。これにより，マシン軸が 2 つの基準マークを検出する前にトラバース範囲限界（ハードウェアリミットスイッチまたは非常停止）に到達することが防止されます。

軸は，2 番目のゼロマークを検出する前にカムに到達したら，すぐに反転して反対方向にサーチを開始します。

基準カムを使用しない場合

マシン軸はトラベルキー（プラスまたはマイナス）の方向へ MD 34040:REFP_VELO_SEARCH_MARKER（原点クリープ速度）で指定された速度にまで加速し，2 つの（同期）基準マークを通過した後停止します。

基準カムを使用する場合

基準カム上に位置していなければ，マシン軸はトラベルキー（プラスまたはマイナス）の方向へ MD 34040:REFP_VELO_SEARCH_MARKER（基準点クリープ速度）で指定された速度にまで加速し，2 つの（同期）基準マークを通過した後停止します（フェーズ 2)。基準カム上に位置していれば，マシン軸は MD 34010: REFP_CAM_DIR_IS_MINUS（マイナス方向の基準点へのアプローチ）で指定された方向とは反対の方向へ MD: REFP_VELO_SEARCH_MARKER（基準点クリープ速度）で指定された速度にまで加速し，プラスのトラベルキーが押されているかマイナスのトラベルキーが押されているかに関わらず，2 つの（同期）基準マークを通過した後停止します（フェーズ 2)。

基準マーク距離

2 つの基準マークの間の距離が 2 * MD 34300: ENC_REFP_MARKER_DIST（基準マーク距離）を超えていて，従って誤っていることが制御装置でわかった場合は，マシン軸は押されたトラベルキー（プラスまたはマイナス）と反対の方向へ MD 34040: REFP_VELO_SEARCH_MARKER（基準点クリープ速度）で定義された半分の速度にまで加速し，2 つの基準マークを通過します。2 つの基準マークの間の距離が 2 * MD 34300: ENC_REFP_MARKER_DIST（基準マーク距離）を超えていることを制御装置が検出した場合は，軸は停止して，アラーム 20003 "error in measuring system" が出力されます。

フェーズ 1 は完了し，それに続いてフェーズ 2 がスタートします。

フェーズ 1 の特徴

マシン軸が，スタート位置から MD 34300: REFP_MAX_MARKER_DIST（基準マークまでの最大距離）で定義された距離だけ移動しても 2 つの基準マークを通過しなければ，軸は停止して，アラーム 20004 "reference mark missing" が出力されます。

■ フェーズ 2: 固定移動先へのトラベル

フェーズ 2 のスタート

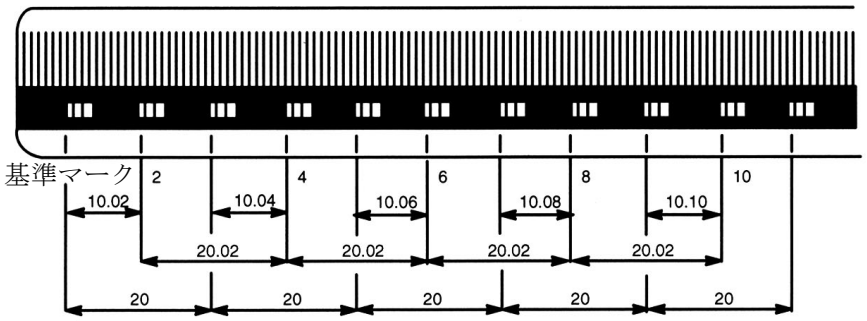
フェーズ 1 が正常に完了する（アラームが出ない）と，フェーズ 2 が自動的にスタートします。

基準点の設定

2 つの連続した基準マークの間の距離は，直線測定系上で基準マークおよび実際位置を（直線測定系上の最初の基準マークに関して）正確に識別できるように可変に定義されます。

直線測定系上の実際位置は，基準点を設定するために（マシン原点に関して）正確なマシン軸位置に割当てられます。これは MD 34090: REFP_MOVE_DIST_CORR（基準点／絶対オフセット）に直線測定系上の絶対オフセットを入力することによって達成されます。

直線測定系をマシン系と反対方向にインストールするかどうかを定義するために，MD 34320: ENC_INVERS（マシン系と反対方向の直線測定系）も設定しなければなりません。



距離コード化基準マークを使用した DIADUR 目盛りガラススケール
最小目盛 20 μm の場合の mm で表された寸法

図 1.147 直線目盛りガラススケール

絶対オフセットの計算

マシン原点と直線測定系上の最初の基準マーク位置との間の絶対オフセットは直接 MD 34090: REFP_MOVE_DIST_CORR （基準点／絶対オフセット）に入力することができます。

REFP_MOVE_DIST_CORR（絶対オフセットは測定が困難なことが多いのでよく問題となります）は，マシン軸がどのような位置にあっても，下記の式から計算することができます（その位置は，レーザ干渉計を使用してマシン原点を基準にして実際のマシン位置を正確に測定できる位置であるべきです）：

マシン系と直線測定系が同方向の場合：

絶対オフセット＝マシンの実際位置 - 直線測定系の実際位置

マシン系と直線測定系が逆方向の場合：

絶対オフセット＝マシンの実際位置 - 直線測定系の実際位置

直線測定系の実際位置は，2 つの連続した基準マークを通過した（同期化）時に，NC 画面の "actual position"（実際位置）の下に表示されます。この場合，MD: REFP_MOVE_DIST_CORR の値は 0 になっていなければなりません。

警告

絶対オフセットが計算されてその値が REFP_MOVE_DIST_CORR に入力された時，位置測定系を再度基準化しなければなりません。

フェーズ 2 の短縮

MD 34330: STOP_AT_ABS_MARKER（移動先がない直線測定系）の値によって設定方法が異なります。下記の設定が可能です：

- （短縮された）距離コード化直線測定系を使用した移動先がないアプローチ
- （ノーマルの）距離コード化直線測定系を使用した移動先があるアプローチ

移動先がないアプローチ

マシン軸は静止したままです。この時点でマシン軸は基準化され，インタフェース信号 "referenced/synchronized 1 or 2"（DB31, ... DBX60.4 or 60.5)（どの位置測定系がアクティブであるかによる）がイネーブルされます。

移動先があるアプローチ

マシン軸は MD 34070: REFP_VELO_POS（基準点スタート速度）で定義された速度にまで加速し，MD 34100: REFP_SET_POS（基準点値）で定義された移動先へアプローチします。この時点でマシン軸は基準化され，インタフェース信号 "referenced/synchronized 1 or 2"（どの位置測定系がアクティブであるかによる）がイネーブルされます。

フェーズ 2 の特徴

- フィードレートオーバライド（フィードレートオーバライドスイッチ）がアクティブです。
- フィード停止（チャンネル別および軸別）がアクティブです。
- NC 停止／スタートを使用してマシン軸を停止したり再スタートしたりできます。

■ パートプログラム別の基準化

パートプログラムを使用して基準化する場合

マシン軸は，パートプログラムの実行中に原点を失うことがあります。そのような場合にはパートプログラム中で基準化を行う必要があります。マシン軸は次ような場合に原点を失う可能性があります：

- 軸がパーキング中で，再プログラムされようとしている場合
- 回転軸がエンコーダ限界を頻繁に超えて原点をなくした場合
- 実際値が PRESETON を使用して設定された場合

プログラミング

パートプログラム別の基準化はコマンド G74 で実行します。複数の軸を同時に基準化することができます。個々のフェーズの順番は，プラス／マイナストラベルキーのかわりに G74 コマンドでスタートさせたプロセス以外は軸別基準化と同じです。

（注）パートプログラムファンクション (G74) を使用して基準化を行う前に下記を参照してください：

参照： YS840DI ユーザーズマニュアル・プログラミング編

■ 絶対値エンコーダを使用した基準化

絶対値エンコーダを有する軸は，制御装置に電源が投入された時，関連した軸がすでに校正済みであることをシステムが検出すれば，自動的に基準化されます（絶対値の転送）。この絶対値の転送は，電源投入時などに軸動作なしで行なわれます。軸が自動的に基準化されるためには，下記の 2 つの条件が満たされる必要があります：

- 軸が，位置制御を行なうのに使用する絶対値エンコーダを有する
- 絶対値エンコーダが校正されている (MD 34210: ENC_REFP_STATE[n] = 2)

校正

絶対値エンコーダを有する軸の場合，測定系は，基準カムにアプローチすることで校正されます（同期化はされません）。このために，絶対値エンコーダの実際値はスタートアップ中に一度設定され，その後に制御装置に送られます。絶対値エンコーダは原則として下記の 3 通りの方法で校正することができます：

- オペレータによる校正
- プローブを使用した自動校正
- BERO を使用した自動校正

絶対値エンコーダの他の原点記録方法はサポートされていません。

■ オペレータによる校正

一般手順

校正されるべき軸は定義された位置へ動かされます。それから適切な実際値が設定されます。

詳細手順

1. MD 34200: ENC_REFP_MODE を 0 にセットし，電源を投入することによって起動させます（ENC_REFP_MODE = 0 は軸の実際値がセットされたことを意味します）。
2. 軸を JOG モードで既知の位置へトラバースさせます。その位置にアプローチする方向は，MD 34010: REFP_CAM_DIR_IS_MINUS（0 = 正の方向，1 = 負の方向）に保存された方向と同じでなければなりません。

   （注）この既知の位置へは，連結部のバックラッシュのせいで誤差が生じないように，常に低速で定義された方向からアプローチしなければなりません。

3. アプローチされた位置に対応する実際値を MD 34100: REFP_SET_POS に入力します。この値は指定された設計値（例えば固定点）でもよいし，この時点で測定器具を使用して求めた測定値でもかまいません。
4. 「校正」ファンクションをイネーブルするために MD 34210: ENC_REFP_STATE を "1" にセットします。
5. 変更されたマシンデータはリセット後に有効になります。
6. JOG-REF モードに切換えます。

7. ステップ 2 で使用したトラベルキーを押すと，現在のオフセットが MD 34090: REFP_MOVE_DIST_CORR に入力され，ENC_REFP_STATE は "2" に変わります。つまり，軸は事実上校正されたことになります。

   （注）

   正しいトラベルキーを押すと軸は動きません。REFP_SET_POS に入力した値は軸位置の実際値画面に表示されます。

8. JOG-REF 運転モードを終了します。校正オペレーションはこれで完了です。

複数の軸の校正

時間を短縮するため，複数の軸を全部，あるいはいくつかを（マシンの構造上可能であれば）それぞれの校正位置（ステップ 1 ～ 5）へ動かし，それから JOG-REF に切換える（ステップ 6）ことによって，複数の軸を校正することができます。各軸は JOG-REF に切換えた後，個別に校正することができます（ステップ 7）。

■ プローブを使用した自動校正

一般手順

NC プログラムが校正軸をトラバースさせます。軸はプローブからの信号に応答して停止し，それから適切な実際値が設定されます。

（注）

プローブを使用して自動的に絶対値エンコーダを校正する場合，軸がかってに動くので衝突が発生しないように十分注意してください。

NC プログラム

ユーザは，関連するマシンの仕様を考慮に入れながら，絶対値エンコーダの校正用の NC プログラムを生成しなければなりません。

プローブを使用した絶対値エンコーダの校正用の NC プログラムは，各軸に対して下記の 4 つのステップを順番に実行しなければなりません：

1. プローブの応答から検出される軸の校正位置へのアプローチ。その位置へは何度か同じ方向からアプローチしなければなりません。得られた測定値 $AA_IM をできるだけ正確なものにするために，アプローチごとで速度を変える（徐々に遅くしていく）必要があります。
2. NC プログラムが MD 34100: REFP_SET_POS と $AA_IM との差を計算する。
3. NC プログラムが MD 34090 REFP_MOVE_DIST_CORR にその差を入力する。
4. NC プログラムが MD 34210: ENC_REFP_STATE=2 にセットする。

詳細手順

1. MD 20700: REFP_NC_START_LOCK=0（基準化されていない軸に対してもプログラムスタートはイネーブル）
2. 全ての軸を校正するために，MD 34100: REFP_SET_POS をマシン上の校正点用の正確な実際位置にプリセットしてください。

3. リセットオペレーションによって，変更された MD が起動されます。
4. NC プログラムをスタートしてください。
5. MD 20700: REFP_NC_START_LOCK=1 に設定してください（基準化されていない軸に対してプログラムスタートはディスエーブル)。
6. 電源を投入してください。新しい値が MD 34100: REFP_MOVE_DIST_CORR に入力され，アクティブなままになります。

   （注）

   校正後に電源オフ／オンせずに軸のオペレーションをそのまま続けると，まだ内部的に使用されている元の値が再度 MD に保存され（自動オーバフロー補正の過程で)，新しい値に元の値が上書きされる可能性があります。

7. 校正の結果をチェック／確認してください。

■ BERO を使用した校正

条件

BERO を使用して絶対値エンコーダを調整する場合，BERO は取付けたままにしておくことも，校正のために一時的に取付けることも可能です。エンコーダは校正された後，適切なインタフェース信号 "Position measurement system 1/2" がアクティブであれば基準化されたままになります。測定系が接続された後，インタフェース信号 "Axis referenced" が再度イネーブルになります。

詳細手順

1. MD 34200: ENC_REFP_MODE=2 にセットしてください（"Referencing with BERO" が選択されます)。
2. BERO の機械的に定義された位置に MD 34100: REFP_SET_POS をプリセットしてください。
3. 原点復帰を開始してください。原点復帰は JOG-REF でユーザがスタートさせることも，プログラムモードの G74 でスタートさせることもできます。
4. BERO は，取付けたままにしておく必要がなければこの時点で取外すことができます。この場合，ENC_REFP_MODE は 0 ("referencing with absolute value encoder") にセットしなければなりません。

測定系の電源の接続

電源が接続された時に軸と測定系との間の同期が調べられます。軸がすでに校正されて同期化されていれば，それ以上校正する必要はありません。

制限事項 :

スピンドルが限界周波数を超えて運転される場合は，リミット MD 36302: ENC_FREQ_LIMIT_LOW を，絶対トラックの最低値にセットしなければなりません。

■ 基準点の確認

条件

「電源切断後の実際値のバッファリング」ファンクション（1.16.2 「■電源をオフにした後の相対エンコーダ用の実際値バッファ」を参照）を使用すると，絶対エンコーダを有していない軸は，マシンが電源切断された時にそれらの軸が正確停止ステータスになっていれば，電源が切断されている間も基準化されたままになります。

この条件が常に満たされているとは限らないので，位置がわからなくなれば，電源投入時に軸を再度基準化しなければなりません。基準カムがない軸については，基準位置をトラバースエリア内に設定すれば十分です。軸を既知の位置にトラバースさせ，基準点を設定し，軸が基準化されたことを示すことによってこのことは達成されます。

（注）測定系のゼロマークは基準点の設定には使用されません。

詳細手順

1. MD 34200: ENC_REFP_MODE を 0 に設定し，電源を投入することによって起動させてください。この時点でこの軸に対して基準点を設定することができます。
2. 軸を JOG モードで既知の位置へトラバースさせてください。その位置にアプローチする方向は，MD 34010: REFP_CAM_DIR_IS_MINUS（0 = 正の方向，1 = 負の方向）に保存された方向と同じでなければなりません。

   （注）この既知の位置へは，ドライブトレインにあるバックラッシュのせいで誤差が生じないように，常に低速で定義された方向からアプローチしなければなりません。

3. アプローチされた位置に対応する実際値を MD 34100: REFP_SET_POS に入力してください。この値は指定された設計値（例えば固定点）でもよいし，この時点で測定器具を使用して求めた測定値でもかまいません。
4. JOG-REF モードに切換えてください。
5. ステップ 2 で使用されたトラベルキーを押すと，現在のオフセットが MD 34090: REFP_MOVE_DIST_CORR に入力されます。

   （注）正しいトラベルキーを押すと軸は動きません。MD 34100: REFP_SET_POS に入力された値は軸位置の実際値画面に表示されます。

6. 軸は基準化されます。

■ 追加情報

その他の基準化方法

EQN 1325，LC181，および同じような設計のエンコーダについては，ゼロマークへの基準化と校正，距離コード化マーク，および 2 つのエッジ評価を使用した BERO はサポートされていません。MD 34200: ENC_REFP_MODE に誤った値を指定した場合には，ソフトウェアバージョン 3.6 以降ではアラーム 26015 が出力されます。

2 つのエンコーダを有する軸

軸が 2 つのエンコーダを有していて (MD 30200: NUM_ENCS=2) 少なくともそれらのうちの 1 つが絶対エンコーダであれば，MD 34102: REFP_SYNC_ENCS=1 はイネーブルされていなければなりません（1.16.2「■基準化測定系への実際値調整」を参照）。

BERO と同期化する際の特別機能

BERO による信号の遅延によって生じた位置誤差は，信号ランタイム補償を入力することによって NC 内で内部的に修正できます。

MD 34200: ENC_REFP_MODE = 2 または 7 という設定に加え，動作のプラス方向またはマイナス方向に対する信号ランタイム補償値（待ち時間）を，MD 31122: BERO_DELAY_TIME_PLUS または MD 31123: BERO_DELAY_TIME_MINUS に入力します。

- MD 34200: ENC_REFP_MODE = 7 という設定は，位置の同期化が，固定速度および（または）MD 34040: REFP_VELO_SEARCH_MARKER に定義された速度に対してのみ行なわれるということを意味します。

  MD 34040 に設定された速度は JOG-REF モード での基準化の間，および G74 を有するパートプログラムに対しても有効です。
- MD 34200: ENC_REFP_MODE = 2 という設定では，特定の速度 / 速度を指定せずに位置の同期化を行ないます。

（注）NC によって信号ランタイムの補償を行なうには対応するドライブを使用する必要があります。
信号ランタイムは，普通は内容を変更する必要がないので，
納品時にプリセットされています。

■ 実際値調整による基準化

■ 基準化測定系への実際値調整

ファンクション

いくつかの測定系が使用されているとき，ゼロマークへの基準化，Bero または基準化に使用されるエンコーダの絶対位置を使用して，他の全ての軸の測定系を調整できます。MD 34102: REFP_SYNC_ENCS は 1 にセットされます。

基準化プロセスが全ての測定系を基準化します。基準位置は，基準化のために使用される測定系の基準点に関連する MD から単独で得られます。

（注）2 つの測定系の内の 1 つが絶対エンコーダであれば，両方の測定系が常に同じ値を有するようにするため，常に REFP_SYNC_ENCS=1 とセットされるべきです。

例

1 つのエンコーダを基準化するともう 1 つのエンコーダにも作用する 2 つの相対エンコーダ

```
CHANDATA(1)
$MA_NUM_ENCS[AX1]=2
$MA_REFP_SYNC_ENCS[AX1]=1
$MA_REFP_MODE[0,AX1]=1
$MA_REFP_MODE[1,AX1]=1
```

■ 2基準化測定系に合わせた実際値の調整（対抗システムについてのみ）

ファンクション

制御装置用に選択されたのではない測定系から基準値を受入れるためには，MD 34200: ENC_REFP_MODE=6 とセットしてください。基準化プロセス用のトラベルパスは MD 34080: REFP_MOVE_DIST に入力されます。この値は測定系のバックラッシュよりも高くするべきであり，その値から基準値が得られます。

この設定は，例えばモータについている間接的絶対値エンコーダ，およびテーブルについている直接的相対エンコーダを使用した測定系の構成に使用されます。バックラッシュを除いた後は，2 つの測定系内での実際値の変更は，リードスクリュー誤差以外は，一致します。この時点では直接的相対エンコーダによって絶対値エンコーダのおおまかな位置が選定され，その後 NC によって直接的エンコーダの完全な精度に修正されます。

従って直接的相対測定系は，MD: REFP_MOVE_DIST に保存されたパスがトラバースした後で基準化されます。この MD: REFP_MOVE_DIST が 0 に設定されていれば，測定系調整はトラベルキーが作動した後ただちに実行されます。このため，バックラッシュは間接的測定系のピッチの半分未満である必要があります。

例

```
CHANDATA(1)
$MA_NUM_ENCS[AX1]=2
$MA_ENC_TYPE[0,AX1]=4
$MA_ENC_TYPE[1,AX1]=1
$MA_REFP_MODE[0,AX1]=0
$MA_REFP_MODE[1,AX1]=6
$MA_REFP_MODE_DIST[1,AX1]=2.0
```

1.16.3 補足条件

■ 絶対エンコーダについての一般補足条件

エンコーダの選択

Heidenhain 社製の絶対エンコーダ（タイプ EQN 1325）、および以下の機能を有する類似の構造を持つエンコーダがサポートされています :

- 生信号出力を有する、互いに 90° オフセットされた 2 つの相対トラック
- EnDat プロトコルを介して値を読取ることのできる、1 つの絶対トラック（単回転あるいは複数回転）

FDD あるいは MSD については、対応するドライブを使用しなくてはなりません。

NC ソフトウェアの可用性

- 絶対エンコーダを運転する NC ソフトウェアは，不可欠な構成要素です。
- 重要な違いは，無限回転軸において，整数ギア伝達比をサポートしているかどうかということです。

  さらに，絶対位置を失うことなく絶対エンコーダによって表示される範囲よりも，移動距離の方が長いことです。

■ 校正

システムの校正時期

校正プロセスは，マシン原点とエンコーダ原点との間のオフセットを測定して不揮発性メモリに保存します。通常は，校正は一回だけ，つまり最初のスタートアップのときにだけ必要です。制御装置はオフセットを知っているので，いつでもエンコーダ絶対値から絶対マシン位置を計算することができます。このステータスは MD 34210: ENC_REFP_STATE=2 で識別されます。

オフセットは MD 34090: REFP_MOVE_DIST_CORR (SRAM) に保存されます。

校正プロセスは以下の場合はもう一度行う必要があります。

- オフセット値が失われた場合（SRAM が消去されたか，バッテリ電圧が低下したか，PRESET された場合）
- エンコーダあるいはエンコーダ内蔵のモータを取付けたか，取外したか，あるいは交換した場合
- モータ（絶対エンコーダ付き）と負荷との間の既存のギアユニットを変更した場合
- エンコーダと負荷との間の機械的な接続が切離され，その後正確に元通りに再接続されたのではない場合

注意 :

制御装置は校正の必要があることを常に検出できるとは限りません。制御装置は，校正の必要があることを検出すると，マシンデータ MD 34210: ENC_REFP_STATE を 0 または 1 にセットします。制御装置は以下の状態を検出できます。

- オフセット値の喪失（SRAM が消去されたか，バッテリ電圧が低下したか，PRESET された場合）
- エンコーダと負荷との間のギア比が異なる別のギア速度への切替わり

これ以外の場合は，ユーザが MD 34210: ENC_REFP_STATE を上書きする必要があります（PLC 介入，サイクル，オペレータ入力を介して）。

データのバックアップ

マシンデータがバックアップされると，MD 34210: ENC_REFP_STATE のステータスも保存されます。

このブロックをロードすると，軸は校正されたものと見なされます。

ブロックを別のマシンから持ってきた場合（たとえばシリーズインストレーションの場合）は，データをロードして起動した後に校正を実施しなければなりません。

MD 30250 についての注記を守ってください。1.16.4「■軸／スピンドル別マシンデータ」（SW 3.6 以降）も参照。

■ トラバース範囲が大きい場合の特殊機能

固有のトラバース範囲

直線スケール（LC181 など）で検出される位置は全スケール長にわたって固有です（オーバフローしません）。

ロータリエンコーダ EQN 1325 の絶対値は，4096 回転を超えるとオーバフローします。

絶対位置は次の最大値が守られる場合に限って固有です。

- 回転軸
  - 負荷側に付いているエンコーダの場合⇒ 4096 回転（負荷側）
  - モータ側に付いているエンコーダの場合⇒ 4096 回転（モータ側）
- 直線軸
  - モータ側に付いているエンコーダ⇒ 4096（有効スピンドルリード）

例

有効スピンドル上昇が 10 mm の直線軸が，-20.48 ～ 20.48 m の距離をカバーする場合。

全範囲を使用するのであれば，エンコーダ原点とマシン原点とを正確にマッチングさせることができないので，有効トラバース範囲内でエンコーダ値がオーバフローします。それでもマシンの実際値は NC 内の内部オフセットを使用して正確に計算されます。これにより，エンコーダの回転数とは関係なく，全トラバース範囲が使用できます。

マシン原点の選択（スピンドルリードが 10 mm の場合）:

- トラバース範囲長が最大 20.48 m
  ⇒マシン原点は自由に選ぶことができます。
- トラバース範囲が 20.48 m から 40.96 m まで
  ⇒両端からのマシン原点間の距離 ≦ 20.48 m
  （長さが 40.96 m の場合，マシン原点は中央）

特殊機能

- 直線軸トラバース範囲が，EQN 1325（MD 34220: ENC_ABS_TURNS_MODULO = 4096）で指定されたエンコーダ 4096 回転を超えるマシンの場合 :
  - SW 3.6 以降は，表示されるトラバース範囲は，ソフトウェア的にサーボモータのフルナンバーフォーマットにまで拡大されました（相対エンコーダと同じになりました）。
  - エンコーダをオフにしたとき（電源オフ／オン，パーキング），回転エンコーダ（EQN1325 など）の軸が MD 34220: ENC_ABS_TURNS_MODULO で定義されたトラバース範囲の半分（EQN1325 の場合は 2048 回転）を超えて移動されることがないようにしなければなりません。この場合，ソフトウェアは最短パス検出を使用して新しい位置を再構築することができます。

- 無限に回転する絶対エンコーダを使用する回転軸の場合 :
  - SW 3.6 以降は，伝達比の数は無制限となっています。これに必要なオーバフロー補正はソフトウェアによって内部的に実行されます。
  - 回転が無制限の回転軸の実際値は，電源オン後に，モジュロオフセットを使用して，基本範囲（たとえば 0 ～ 360°）にまで減少させられます。そうしないと，軸が最初に原点に配置されるときに，長い距離をトラバースしなければならないことになります。減少させる範囲は MD 34220: ENC_ABS_TURNS_MODULO（EQN 1325 の場合に重要）で設定します。
  - エンコーダをオフにしたとき，軸が絶対エンコーダの有効範囲の半分を超えて移動されることがないようにしなければなりません。つまり，エンコーダ動作はエンコーダ回転数 ENC_ABS_TURNS_MODULO/2 を超えてはなりません。

    推奨事項 : ハンドブレーキをかけておく。
  - 制御装置は，校正を必要とするすべての場合を検出できるわけではなく，また，調整可能な絶対エンコーダ位置が無効であるすべての場合（たとえば，エンコーダの負荷からの切離し，特に制御装置が停止中か，エンコーダが非アクティブか，パーキング状態の場合）を検出できるわけでもありません。制御装置が校正の必要性を検出した場合は，MD 34210: ENC_REFP_STATE を 0 あるいは 1 にセットすることによって通知します（1.16.3「■校正」を参照）。

    校正の必要性が検出されたならば，MD 34210: ENC_REFP_STATE=0 にセットするのはユーザの責任です。
- 有限回転軸の場合
  - SW 3.6/4.1 以降は，内部ソフトウェア拡張は，正と負の値（たとえば -180 ～ 180°）が表現可能なトラバース範囲に含まれていたことを意味します。

1.16.4 データの説明 (MD, SD)

■ チャンネル別マシンデータ

20700 MD 番号	REFP_NC_START_LOCK 原点なしで NC スタート不可	
初期設定 : 1	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : リセット	保護レベル : 2/7	ユニット : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	0: 系列の 1 つあるいはすべての軸がまだ原点復帰されていなくても，パートプログラムあるいはパートプログラムブロック（MDA およびオーバストア）をスタートさせるための "NC start" インタフェース信号 (DB21, ... DBX7.1) は有効である。軸が正しい位置に到達できるように，他の方法（スクラッチ法，自動ゼロオフセット計算など）でワークピース座標系を正しい値に設定しなければならない。 1: すべての軸が原点復帰されている場合にのみ NC スタート可。	

■ 軸／スピンドル別マシンデータ

番号	識別子	名称	参照
30200	NUM_ENCS	エンコーダの数	1.8（G2）

30240 MD 番号	ENC_TYPE **エンコーダタイプ**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 5
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	ユニット : -
データタイプ : BYTE	適用開始 SW バージョン : 1.1	
意味 :	エンコーダタイプ : 0: シュミレーション 1: 生信号エンコーダ（高分解能） 2: 方形波エンコーダ（標準、エンコーダマーク数の倍数） 3: EnDat インタフェース付きの絶対エンコーダ	
関連性	Drive MD 1011: ACTUAL_VALUE_CONFIG, Bit 3	

34000 MD 番号	REFP_CAM_IS_ACTIVE **原点カム付きの軸**	
初期設定 : 1	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	ユニット : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	全移動範囲中でゼロマークが一つしかないマシン軸か、あるいは 1 回転当たりのゼロマークが一つしかない回転軸は、MD: REF_CAM_IS_ACTIVE によって原点カム付きの機械軸としては識別されない。プラス／マイナス移動キーを押すと、原点カムを有する機械軸として識別された機械軸が、MD 34040: REFP_VELO_SEARCH_MARKER（原点復帰クリープ速度）で定義された速度になるまで加速する。	
この MD は次の場合は無意味	距離コード化された基準マークを有する直線測定系	

34010 MD 番号	REFP_CAM_DIR_IS_MINUS **マイナス方向での原点復帰**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	ユニット : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	**REF_CAM_DIR_IS_MINUS = 0** プラス方向での原点復帰 **REF_CAM_DIR_IS_MINUS = 1** 相対測定系の場合 : 機械軸が原点カムの前にあれば、プラス／マイナス移動キーを押すと、マシン軸が、MD: REFP_CAM_DIR_IS_MINUS で指定された方向に、MD 34020: REFP_VELO_SEARCH_CAM（原点復帰アプローチ速度）で定義された速度になるまで加速する。間違った移動キーを押すと、原点復帰はスタートしない。 機械軸が原点マーク上にあれば、機械軸は、MD: REFP_CAM_DIR_IS_MINUS で指定された方向とは反対の方向に、MD 34020: REFP_VELO_SEARCH_CAM（原点復帰速度）で定義された速度になるまで加速する。 距離コード化された基準マークを有する直線測定系の場合 : 原点カムを有するマシン軸（距離コード化された原点マークを有する直線測定系は基準カムを必要としない）がカム上にある場合は、プラスかマイナスかには関係無く移動キーが押されると、MD: REFP_CAM_DIR_IS_MINUS で指定された方向とは反対の方向に、MD 34040: REFP_VELO_SEARCH_MARKER（原点復帰クリープ速度）で定義された速度になるまで加速する。	

34020 MD 番号	REFP_VELO_SEARCH_CAM **原点復帰速度**	
初期設定 : 5000	最小入力リミット : 0	最大入力リミット : プラス
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : mm/min, r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	原点復帰速度とは、移動キーが押されたときに、機械軸が原点カムの方向に移動するとき（フェーズ 1）の速度である。この値は、機械軸がハードウェアリミットスイッチに到達する前に停止させられるだけの大きさに設定されているべきである。	
この MD は次の場合は無意味	距離コード化された基準マークを有する直線測定系	

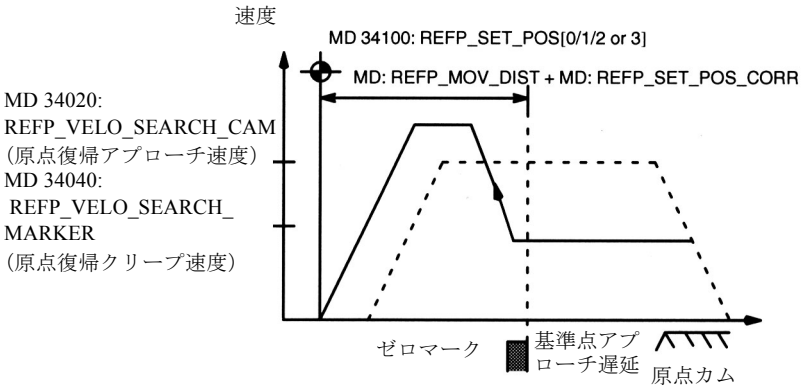
34030 MD 番号	REFP_MAX_CAM_DIST **原点カムまでの最大距離**	
初期設定 : 10000	最小入力リミット : 0	最大入力リミット : プラス
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	機械軸が原点カムの方向にスタート位置から最大距離（MD: REFP_MAX_CAM_DIST で定義された距離）だけ移動しても原点カムに到達しなかった場合は（"reference point approach delay" インタフェース信号 (DB31, ... DBX12.7) がリセット)、軸が停止してアラーム 20000 "reference cam not reached"（原点カムに到達せず）が出力される。	
この MD は次の場合は無意味	距離コード化された基準マークを有する直線測定系	

34040 MD 番号	REFP_VELO_SEARCH_MARKER[n] **原点復帰クリープ速度 [エンコーダ番号]: 0, 1**	
初期設定 : 300	最小入力リミット : 0	最大入力リミット : プラス
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : mm/min, r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	1) 相対測定系の場合 : 原点復帰クリープ速度とは、原点カムが最初に検出されてから最初のゼロマークに同期するまでの間（フェーズ 2）の軸の移動速度である。 移動方向 : 指定された方向 (MD 34010: REFP_CAM_DIR_IS_MINUS) とは逆の方向 MD 34050: REFP_SEARCH_MARKER_REVERSE（原点カム上での方向反転）が有効であれば、原点カム信号の立上がりエッジで同期させる場合は、軸は MD 34020: REFP_VELO_SEARCH_CAM で定義されている速度で移動する。 2）距離コード化された基準マークを有する直線測定系 : 軸はこの速度で 2 つの基準マークを通過する。最高速度は、直線測定系での最小基準マーク距離（x min）を移動するのに要する時間が位置制御同期一つ分より長くなるような速度に制限されなければならない。以下に示す式では、基本距離 [グラジュエーションサイクルの倍数]、グラジュエーションサイクル [mm]、および測定長 [mm] で最高速度が求まる。 $x_{min}\,[mm] = \frac{\text{基本距離}}{2} * \text{グラジュエーションサイクル} - \frac{\text{測定長}}{\text{基本距離}}$ $\text{最大距離 [m/s]} = \frac{x_{min}\,[mm]}{\text{位置コントローラサイクル [ms]}}$ この速度制限は他の測定系にも当てはまる。 移動方向 : - MD: REFP_CAM_DIR_IS_MINUS で定義されている方向 - 軸がカム上ですでに位置決めされていれば、軸は反対方向に移動される。 3) BERO が負荷側にある間接測定系（スピンドルに適している） この速度で、BERO に関連するゼロマークがサーチされる（ゼロマーク選択は VDI 信号による）。実際の速度が MD 34040:REFP_VELO_SEARCH_MARKER[n] で定義され、MD 35150: SPIND_DES_VELO_TOL で定義された公差範囲内にあればゼロマークは受入れられる。	

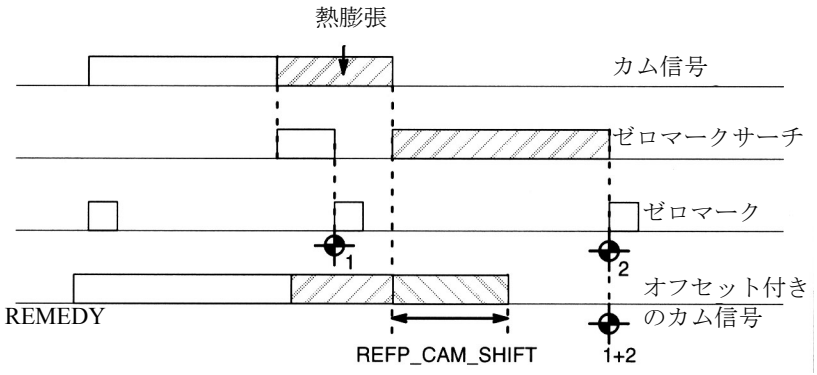
34050 MD 番号	REFP_SEARCH_MARKER_REVERSE[n] **原点カム上での方向転換 [エンコーダ番号]: 0, 1**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	MD: REFP_MARKER_REVERSE を使用してゼロマークをサーチする方向を設定することができる。 REFP_SEARCH_MARKER_REVERSE = 0 原点カム信号の立下がりエッジで同期 機械軸は、MD 34010: REFP_CAM_DIR_IS_MINUS（マイナス方向での原点復帰）で指定された方向とは反対の方向に、MD 34040: REFP_VELO_SEARCH_MARKER（原点復帰クリープ速度）で定義された速度になるまで加速する。 機械軸が原点カムから外れると（"reference point approach delay" インタフェース信号 (DB31, ... DBX12.7) がリセットされると）、制御装置は最初のゼロマークに同期する。 MD: REFP_SEARCH_MARKER_REVERSE = 1 原点カム信号の立上がりエッジで同期 機械軸は、MD 34010: REFP_CAM_DIR_IS_MINUS（マイナス方向での原点復帰）で指定された方向とは反対の方向に、MD 34040: REFP_VELO_SEARCH_MARKER（原点復帰クリープ速度）で定義された速度になるまで加速する。 軸が原点カムから外れると（"reference point approach delay" インタフェース信号 (DB31-48, DBX12.7) がリセットされると）、軸は減速停止し、そして MD: REFP_VELO_SEARCH_MARKER で定義された速度になるまで、原点カムへの方向とは逆の方向へ加速する。 軸が基準カムに到達すると（"reference point approach delay" インタフェース信号 (DB31, ... DBX12.7) がイネーブルされると）、制御装置は最初のゼロマークと同期する。	
この MD は次の場合は無意味	距離コード化された基準マークを有する直線測定系	

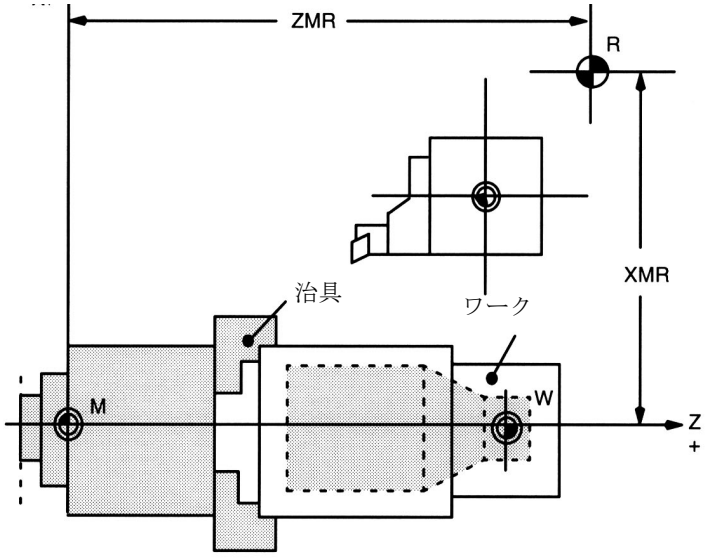
34060 MD 番号	REFP_MAX_MARKER_DIST[n] **基準マークまでの最大距離** **距離コード化された測定系についての 2 つの基準マークまでの最大距離** **[エンコーダ番号]**: 0, 1	
初期設定 : 20	最小入力リミット : 0	最大入力リミット : プラス
変更が有効になるための条件 : 電源オン	保護レベル :	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	相対測定系の場合 : 原点カムから外れた（"reference point approach delay" インタフェース信号がリセットされた）後で、軸が MD: REFP_MAX_MARKER_DIST で定義された距離だけ移動してもゼロマークを検出しなかった場合は、軸が停止してアラーム 20002 "zero mark missing"（ゼロマークが見つからない）が出力される。 距離コード化された基準マークを有する直線測定系の場合 : 軸が、スタート位置から MD: REFP_MAX_MARKER_DIST で定義された距離だけトラベルしても 2 つのゼロマークを通過しなければ、軸は停止してアラーム 20004 "reference mark missing"（基準マークが見つからない）が出力される。	
アプリケーション例	相対測定系で、いつでも同じゼロマークが同期用に使用されることが大切である場合は（誤ったマシン原点を検出することがないように）、MD: REFP_MAX_MARKER_DIST で指定されている最大値が 2 つの基準マーク間の距離を超えることのないようにしなければならない。	

34070 MD 番号	REFP_VELO_POS **原点復帰スタート速度**	
初期設定 : 10000	最小入力リミット : 0	最大入力リミット : プラス
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : mm/min, r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	相対測定系の場合 : 軸は、最初のゼロマークとの同期から原点到着までの間はこの速度で移動する。 距離コード化された基準マークを有する直線測定系の場合 : 軸は、同期の時点（2 つのゼロマークを通過した時点）から原点到着までの間はこの速度で移動する。	

34080 MD 番号	REFP_MOVE_DIST[n] **距離コード化された測定系についての原点復帰距離／目標位置 [エンコーダ番号]: 0, 1**	
初期設定 : -2.0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 7/2	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	相対測定系 : 最初のゼロマークとの同期の後で、軸は MD 34070: REFP_VELO_POS（原点復帰スタート速度）で定義された速度にまで加速し、MD: REFP_MOVE_DIST と MD 34090: REFP_MOVE_DIST_CORR（原点オフセット）とを加算して計算した距離だけ移動する。 この距離は、フェーズ 2 で検出されたゼロマークと原点との間の距離に正確にマッチする。  距離コード化された基準マークを有する直線測定系の場合 : 軸は MD: REFP_VELO_POS で定義された速度まで加速し、MD 34100: REFP_SET_POS で定義された移動先にアプローチする。移動先は絶対位置 (G90) として MD に入力する。 MD 34200: ENC_REFP_MODE が 6 の場合、この移動後に測定系相互間で実際値が合わされる。入力する距離は、バックラッシュ量よりも大きくなっているべきである。	
この MD は次の場合は無意味	距離コード化された基準マークを有する直線測定系 MD 34330: REFP_STOP_AT_ABS_MARKER（移動先のない直線測定系）が 1 にセットされている場合	
関連性	MD 34330: REFP_STOP_AT_ABS_MARKER	

34090 MD 番号	REFP_MOVE_DIST_CORR[n] **原点オフセット／絶対オフセット、距離コード化 [エンコーダ番号]: 0, 1**	
初期設定 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	• ゼロマーク付きの相対エンコーダ : ゼロマークが検出されると、軸は MD 34080: REFP_MOVE_DIST + REFP_MOVE_DIST_CORR の距離だけゼロマークから離れる。軸はこの距離だけ移動してから原点に到着する。プリセット MD 34100: REFP_SET_POS を実際値として設定する。 REFP_MOVE_DIST+REFP_MOVE_DIST_CORR の距離だけ軸が移動するとき、オーバライドスイッチと MD : JOG_INC_MODE_IS_CONT（連続／ジョグモード）は有効です。 • 距離コード化された測定系 : REFP_MOVE_DIST_CORR が絶対オフセットの働きをする（機械原点と測定系の最初の基準マークとの間のオフセットを定義する）。 • 絶対値エンコーダ : REFP_MOVE_DIST_CORR が絶対オフセットの働きをする（機械原点と絶対測定系のゼロ点との間のオフセットを定義する）。 注記 : 絶対エンコーダの場合は、この MD は測定時に制御装置によって変更される。 このため、手動入力した後、あるいはパートプログラムでのこの MD を変更した後で、新しい値がアクティブとなるように（失われることのないように）、電源オンリセットすべきである。	

34092 MD 番号	REFP_CAM_SHIFT **ゼロマークを有する相対測定系の原点カムオフセット**	
初期設定 : 0.0	最小入力リミット : 0.0	最大入力リミット :
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : mm
データタイプ : DOUBLE	適用開始 SW バージョン : 3.2	
意味 :	原点カム信号が出ても、ゼロマークサーチは直ちには開始されず、REFP_CAM_SHIFT の距離だけ遅れて開始される。このようにして、温度によって原点カムが伸びた場合にも、ゼロマークを適切に選択することによって、ゼロマークサーチの再現性が保証される。 補間サイクルで制御装置が原点カムオフセットを計算するとき、実際のカムオフセットは、REFP_CAM_SHIFT から REFP_CAM_SHIFT+(MD 34040: REFP_VELO_SEARCH_MARKER* 補間サイクル) となる。 原点カムオフセットはゼロマークサーチの方向に作用する。 原点カムオフセットは MD 34000: REFP_CAM_IS_ACTIVE=1 の場合にのみ有効である。  	

34100 MD 番号	REFP_SET_POS[n] **相対測定系の原点および MD 34330: REFP_STOP_AT_ABS_MARKER[n] = 0 の場合の距離コード化測定系の原点位置**	
初期設定 : 0.0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	• ゼロマーク付きのインクリメンタルエンコーダ : ゼロマークが検出され、REFP_MOVE_DIST + REFP_MOVE_DIST_CORR（ゼロマークからの距離）の距離だけ移動した後で現在の軸の位置として設定される値。カム信号 (INTSIG DB31, ...DBX2.4-2.7) の立上がりエッジで、原点番号に対応した REFP_SET_POS が軸位置として設定される。 • 距離コード化測定系 : REFP_STOP_AT_ABS_MARKER が 0 (FALSE) にセットされ、2 つのゼロマークが通過されたときにアプローチされる目標位置。 • 絶対値エンコーダ : REFP_SET_POS は正しい実際値に対応している。 マシンの動作は MD34210: ENC_REFP_STATE のステータスによる : ENC_REFP_STATE = 1 の場合は、REFP_SET_POS の値が絶対値として転送される。 ENC_REFP_STATE = 2 および REFP_STOP_AT_ABS_MARKER = 0 (FALSE) の場合は、軸は ,REFP_SET_POS に保存されている目標位置にアプローチする。 (INTSIG DB31, ...DBX2.4-2.7) を介して設定された REFP_SET_POS 値が使用される。	
図 1.148	 M 機械原点 W ワーク座標 R 基準点 XMR X 方向の基準点 (MD: REFP SET POS [x]) ZMR Z 方向の基準点 (MD: REFP SET POS [z])	
関連性	"Reference point value 1 to 4" インタフェース信号 (DB31, ... DBX2.4 -2.7)	

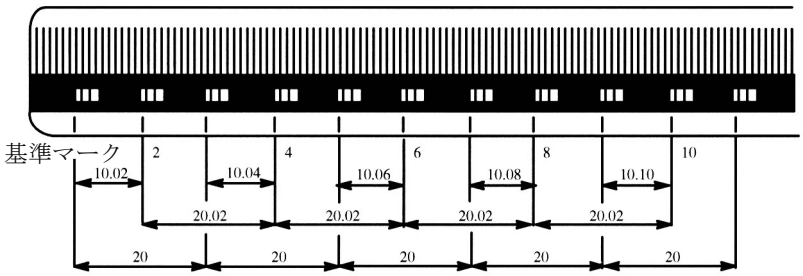
34102 MD 番号	REFP_SYNC_ENCS 測定系調整有効	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 3.1	
意味 :	このマシンデータを使用することで、関連する軸のすべての測定系について、基準測定系を調整することができる。調整は、原点復帰中に行われるか、あるいは閉ループ制御用に選択された校正済みの絶対値エンコーダがオンになったときに行われる。 値 : 0: 測定系は調整されない。測定系は個別に基準化されなければならない。 1: 基準測定系を基準にして軸のすべての測定系が調整される。	
関連性		

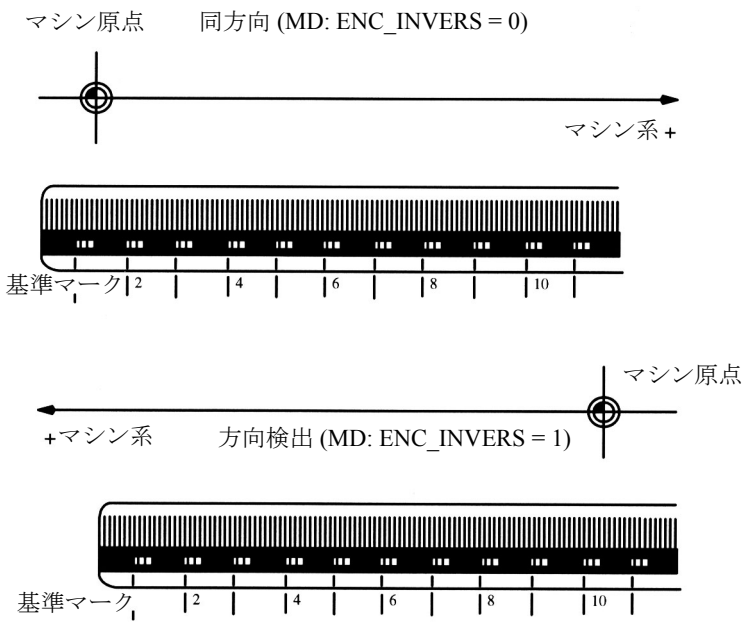
34110 MD 番号	REFP_CYCLE_NR **系列別原点復帰軸シーケンスの軸の順番**	
初期設定 : 0	最小入力リミット : -1	最大入力リミット : 8 (840D), 5 (FM-NC/810D)
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 1.1	
意味 :	MD: REFP_CYCLE_NR = 0 -----> 軸別原点復帰動作 軸別原点復帰は、"plus/minus travel keys" インタフェース信号 (DB31, ... DBX4.7 and 4.6) がスタートした状態で、各軸ごとに別個にスタートされる。 最大 8 軸が同時に可能。 特定のシーケンスでマシン軸を原点復帰動作するのに以下の方法が使用できる : ・ オペレータがスタートアップ時にシーケンスを確認する。 ・ PLC ユーザプログラムがスタートアップ時にシーケンスをチェックするか、あるいはシーケンスそのものを定義する。 ・ 系列別基準化ファンクションを使用する。 MD: REFP_CYCLE_NR = 1 -----> 系列別原点復帰動作 系列別原点復帰は、"activate referencing" インタフェース信号 (DB21, ... DBX1.0) でスタートする。制御装置は、スタートが成功すれば "referencing active" インタフェース信号 (DB21, ... DBX33.0) で応答する。 系列に関連する各軸は、系列別原点復帰で原点復帰することができる（プラス／マイナス移動キーをシミュレートすることによって制御装置で内部的に行われる）。 軸別の MD: REFP_CYCLE_NR を使用することによって、マシン軸の原点復帰動作のシーケンスを定義することができる : -1 にセットした場合 : 軸は系列別原点復帰によってはスタートしない。NC スタートはこの軸を原点復帰しなくても可能。 0 にセットした場合 : 軸は系列別原点復帰によってはスタートしない。NC スタートはこの軸を原点復帰しなければ不可。 1 にセットした場合 : 軸は系列別原点復帰によってスタートする。 2 にセットした場合 : MD: REFP_CYCLE_NR を 1 にセットされたすべての軸が原点復帰されていれば、軸は系列別原点復帰によってスタートする。 3 にセットした場合 : MD: REFP_CYCLE_NR を 1 にセットされたすべての軸が原点復帰されていれば、軸は系列別原点復帰によってスタートする。 4 ～ 8 にセットした場合 : 上記と同じ（追加の軸について）。 系列別の MD 20700: REFP_NC_START_LOCK（原点復帰なしで NC スタート不可）をゼロにセットすることは、系列のすべての軸について -1 を入力するのと同じである。	
この MD は次の場合は無意味	軸別原点復帰動作	
関連性	"activate referencing" interface signal (DB21, ... DBX1.0) "referencing active" interface signal (DB21, ... DBX33.0)	

34200 MD 番号	ENC_REFP_MODE[n] **原点復帰動作モード [エンコーダ番号]: 0 ～ 最大エンコーダ数 - 1**	
初期設定 : 1	最小入力リミット : 0	最大入力リミット : 7
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 1.1	
意味 :	MD:ENC_REFP_MODE を使用することで、接続された測定系は原点復帰用に 2 つのグループに分けられる。 • ENC_REFP_MODE = 0 絶対エンコーダの場合 : MD 34100: REFP_SET_POS の設定 その他のエンコーダの場合 : 原点復帰不可（SW2.2 以降） • ENC_REFP_MODE = 1 相対測定系用の原点復帰 : 相対回転系 相対直線系 エンコーダのゼロパルス （絶対エンコーダ以外） • ENC_REF_MODE = 2 : 1 エッジ検出付きの BERO; 絶対エンコーダでも可能。原点復帰されると、エンコーダは "adjusted"（調整済み）になる。 • ENC_REFP_MODE = 3 距離コード化された基準マークを有する直線測定系での原点復帰動作 : 距離コード化された基準マークを有する直線測定系 (Heidenhain) • ENC_REF_MODE = 4 : 基準点 BERO の正のエッジと負のエッジが連続して通過され、実際値が登録される。計算された平均値が同期点である。2 エッジ評価を使用すると、BERO のドリフトあるいは熱膨張を補償することができる。 • ENC_REF_MODE = 5: BERO が通過されエッジが検出されるとゼロマークサーチが開始し、次に検出されたゼロマークで測定系の原点復帰が完了する。 • ENC_REFP_MODE = 6 原点設定済みのエンコーダ（NCU570 を除く）（SW3.2 以降）を基準にした測定系の校正 • ENC_REFP_MODE = 7 軸およびスピンドルに構成されたアプローチ速度を有する BERO（SW3.6 以降） MD 34040: REFP_VELO_SEARCH_MARKER[n] 原点復帰クリープ速度 [エンコーダ番号]	

34210 MD 番号	ENC_REFP_STATE[n] **絶対値エンコーダステータス [エンコーダ番号]: 0 ～ 最大エンコーダ数 - 1**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 2
変更が有効になるための条件 : 即時	保護レベル : 4/7	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 2.2	
意味 :	• 絶対値エンコーダ : ENC_REFP_STATE = 0: 初めてインストールしたときの初期設定 ENC_REFP_STATE = 1: エンコーダ調整イネーブル（ただしエンコーダは未調整） ENC_REFP_STATE = 2: エンコーダ調整済み • 相対エンコーダ : ENC_REFP_STATE = 0: 初期設定：自動基準化なし ENC_REFP_STATE = 1: 自動原点復帰イネーブル（ただし原点復帰がまだ行われていないか停止していない） ENC_REFP_STATE = 2: エンコーダは基準化済みで停止している（エンコーダが次に起動されたときに自動原点復帰が有効になる）	
アプリケーション例	MD ENC_REFP_STATE は以下のようにインストールする人およびオペレーティングシステムによって変更可能である : • 絶対値エンコーダ : - インストールする人が変更する場合 : このエンコーダの調整が必要であれば、インストールする人がデータを "1" にセットしなければならない。 - オペレーティングシステムが変更する場合 : 調整が成功した場合 1 ==> 2 調整が無効になった場合 2 ==> 0 または 1 SRAM の損失とギア速度変更（ギア比の変更による）はオペレーティングシステムによって識別される。 機械の機械的な構成要素の変更（エンコーダの変更、モータとエンコーダの変更など）は検知されない。 • 相対エンコーダ : - インストールする人が変更する場合 : 自動原点復帰を実行するのであれば、インストールする人がデータを "1" にセットしなければならない。 - オペレーティングシステムが変更する場合 : 軸が原点復帰され停止している場合 1 ==> 2 原点位置が無効になったか軸が停止していない場合 2 ==> 1 絶対値エンコーダの場合とは異なって、電源がオフになっているかエンコーダが非アクティブのときは位置の変更は検知されない。	
この MD は次の場合は無意味	ゼロマーク付きの相対エンコーダ（標準エンコーダ） 距離コード化された原点マークを有する測定系	

34300 MD 番号	ENC_REFP_MARKER_DIST[n] **距離コード化スケール用の基準マーク距離 [エンコーダ番号]: 0 ～ 最大エンコーダ数 - 1**	
初期設定 : 10	最小入力リミット : -	最大入力リミット : -
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	2 つの基準マーク間の距離が ENC_REFP_MARKER_DIST の 2 倍を超えており、したがって正しくないことを制御装置が検出した場合は、押された移動キー（プラスあるいはマイナス）の方向とは反対の方向に、MD 34040: REFP_VELO_SEARCH_MARKER（原点移動クリープ速度）で定義された速度の半分の速度になるまで加速し、2 つの基準マークを通過する。2 つの基準マーク間の距離が ENC_REFP_MARKER_DIST の 2 倍を超えていることを依然として制御装置が検出した場合は、軸は停止したままとなって、アラーム 20003 "error in measurement system"（測定系エラー）が出力される。	
この MD は次の場合は無意味	距離コード化された基準マークのない測定系	

34310 MD 番号	ENC_MARKER_INC[n] **距離コード化スケール用の 2 つの基準マーク間の距離 [エンコーダ番号]: 0 ～ 最大エンコーダ数 - 1**	
初期設定 : 0.02	最小入力リミット : 0	最大入力リミット : プラス
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	通過された基準マークの位置が直線測定系で正確に測定されるように、2 つの基準マーク間の距離が可変的に定義される。2 つの基準マーク間の距離は MD:ENC_MARKER_INC に入力する。図 1.149 の例では、この値は 20.02 - 20 = 0.02 [mm]	
この MD は次の場合は無意味	相対測定系	
図 1.149	 距離コード化基準マーク付きの、目盛入り DIADUR ガラス板 単位は mm （最小目盛は 20 μ m） 2 つの基準マーク間の距離は 0.02 mm。	
例外、エラー	Heidenhain 社製の距離コード化基準マーク付き直線測定系の場合は、2 つの基準マーク間の距離は常に最小目盛に等しい。	

34320 MD 番号	ENC_INVERS[n] **機械システムとは逆方向の直線測定系 [エンコーダ番号]: 0 ～ 最大エンコーダ数 - 1**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	• 測定系 • 距離コード化測定系の場合 : 基準点を設定するときには、直線測定系上の実際位置（距離コード化基準マークで決定された）が正確な軸位置（機械原点を基準にした）に割当てられる。このことは、直線測定系上の絶対オフセットを MD 34090 MD: REFP_MOVE_DIST_CORR（基準点／絶対オフセット）に入力することによって達成される。直線測定系が機械システムに対して同方向かあるいは逆方向のどちらにインストールされているかを定義するために、MD: ENC_INVERS もセットしなければならない。	
この MD は次の場合は無意味	相対センサあるいは距離コード化基準マーク	
図 1.150		

34330 MD 番号	REFP_STOP_AT_ABS_MARKER[n] 目標位置なし距離コード化直線測定系 [エンコーダ番号]: 0 ～ 最大エンコーダ数 - 1	
初期設定 : 1	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : mm, 度
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	• 距離コード化測定系 : REFP_STOP_AT_ABS_MARKER = 0: 原点復帰サイクルの最後に、MD 34100: REFP_SET_POS で入力した位置にアプローチされる（フェーズ 2 の通常の場合） REFP_STOP_AT_ABS_MARKER = 1: 2 番目の基準マークが検出された後で軸に制動がかかる（フェーズ 2 の短縮）。 • 絶対値エンコーダ : MD REFP_STOP_AT_ABS_MARKER は、G74 での、あるいは JOG-REF で移動キーが押されたときの、MD 34210: ENC_REFP_STATE = 2 の軸の応答を定義する : REFP_STOP_AT_ABS_MARKER = 0: 軸は MD: REFP_SET_POS に入力された位置にまで移動する。 REFP_STOP_AT_ABS_MARKER = 1: 軸は移動しない。	
アプリケーション例	距離コード化リファレンスマークを有する直線測定系上での原点復帰の最後で固定目標値へのアプローチが行われないようにするためには、MD: STOP_AT_ABS_MARKER を有効化しなければならない。しかし、固定移動先は相対測定系の場合と同様にアプローチされることが多い。	
この MD は次の場合は無意味	ゼロマーク付きの相対エンコーダ（標準エンコーダ）	
関連性	MD 34080: REFP_MOVE_DIST (Reference point distance/destination point for distance-coded system)	

36302 MD 番号	ENC_FREQ_LIMIT_LOW **エンコーダの再同期限界周波数**	
初期設定 : 99.9	最小入力リミット : 0	最大入力リミット : 100
変更が有効になるための条件 : NEW_CONF	保護レベル : 2/7	単位 : %
データタイプ : DOUBLE	適用開始 SW バージョン : 3	
意味 :	エンコーダ周波数監視ファンクションにはヒステリシスがある。エンコーダがオフになるエンコーダ限界周波数は MD 36300: ENC_FREQ_LIMIT で定義される : ENC_FREQ_LIMIT_LOW はエンコーダが再びオンになるときの周波数である。 ENC_FREQ_LIMIT はヘルツで直接入力し、ENC_FREQ_LIMIT_LOW は ENC_FREQ_LIMIT のパーセントとして入力する。 MA_ENC_FREQ_LIMIT_LOW の初期設定で通常は十分である。しかし、EnDat インタフェースを有する絶対値エンコーダ上での絶対値トラックの限界周波数は、相対値トラックの限界周波数よりも低い。 ENC_FREQ_LIMIT に低い値を設定することにより、絶対トラックの限界周波数よりも周波数が低くなるまではエンコーダが再びオンにならないようにでき、したがって絶対トラックによって許可されるまでは基準化されないようにできる。この基準化はスピンドルについては自動的に実施される。 例 : EQN 1325: 相対トラックの限界周波数 : 430 kHz ===> ENC_FREQ_LIMIT = 430000 Hz 絶対トラックの限界周波数 : エンコーダ約 2000 rpm （2048 個のマーク）、つまり限界周波数 (2000/60) * 2048 Hz = 68 kHz ===> ENC_FREQ_LIMIT_LOW = 68/430 = 15 %	
関連性		

36310 MD 番号	ENC_ZERO_MONITORING **ゼロマーク監視**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : *
変更が有効になるための条件 : NEW_CONF	保護レベル : 2/7	単位 : -
データタイプ : DWORD	適用開始 SW バージョン : 1	
意味 :	この MD はゼロマーク監視ファンクションを起動する。 0: ゼロマーク監視なし >0: 何回ゼロマークエラーが検出されたらモニタが応答する（アラームが出る）かを指定する。 =100: エンコーダ監視停止（アラーム 25000, 25001） 絶対エンコーダ EQN 1325 はゼロマークがないので、BERO と組合わせてゼロマークとしてしか使用できない。ゼロマークモニタは非アクティブのままにしておく（初期設定 0 のままにしておく）べきである。	
関連性		

30250 MD 番号	ACT_POS_ABS **停止時の絶対エンコーダの位置**	
初期設定 : 0.0	最小入力リミット : -	最大入力リミット : -
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : -
データタイプ : DOUBLE	適用開始 SW バージョン : 3.6	
意味 :	現在位置がこの MD に保存される（内部フォーマットで)。 この値は次の目的で電源オン（あるいはエンコーダの起動）時に使用される : • 回転絶対値エンコーダの場合 : 現在位置を復元するため（エンコーダに保存されている位置（曖昧である可能性あり）とともに） • 相対エンコーダの場合 : MD 34210: ENC_REFP_STATE ≠ 0（等価基準点として）ファンクションが有効時に、電源オフ時の実際値を保存するため 注 : 移動中にこの MD は制御装置内で内部的に変更される。このため、以前にバックアップしておいた MD ブロックを入力すると、絶対エンコーダの校正が破壊される。 SW をグレードアップする場合は、グレードアップの直前に、古い SW リリースの MD ブロックをバックアップしておいて、軸を一切動かすことなく、新しい SW リリースをインストールすることが望ましい。SW 3.6 については保護レベル 1 をセットすべきである。SW 4 以降では保護レベル 2 で十分である。SW をグレードアップした後で、エンコーダ校正を明示的に確認（チェック／調整）すべきである。	

1.1.5 信号の説明

■ チャンネル別信号

■ チャンネルへの信号

DB21, ... DBX1.0 データブロック	Activate referencing（基準化の起動） 系列への信号 (PLC → NCK)	
エッジ評価 : あり	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 または信号遷移 0 ---> 1	系列別原点復帰は "activate referencing"（基準化起動）インタフェース信号で起動する。起動が成功すると、制御装置は "referencing active"（基準化アクティブ）インタフェース信号で確認する。系列に割当てられている各マシン軸は、系列別原点復帰（プラス／マイナス移動キーを内部的にシュミレートすることによって制御装置で起動される）で基準化することができる。軸別の MD 34110: REFP_CYCLE_NR（の軸シーケンス）を使用することによって、基準化される軸のシーケンスを定義することができる。MD: REFP_CYCLE_NR に入力されたすべての軸が終点に到着すると、"all axes referenced"（全軸原点復帰完了）インタフェース信号 (DB21, ... DBX36.2) がイネーブルになる。	
アプリケーション例	軸を特定のシーケンスで基準化する場合、以下のオプションが使用できる : • オペレータがスタートアップ時にシーケンスが正しいか確認する。 • PLC ユーザプログラムがスタートアップ時にシーケンスをチェックするか、あるいはシーケンスそのものを定義する。 系列別原点復帰ファンクションを使用する。	
関連性	"referencing active" interface signal (DB21, ... DBX33.0) "all axes referenced" interface signal (DB21, ... DBX36.2)	

DB21, ... DBX33.0 データブロック	Referencing active（基準化アクティブ） 系列への信号 (PLC → NCK)	
エッジ評価 : あり	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 または信号遷移 0 ---> 1	系列別原点復帰が "activate referencing" インタフェース信号で起動され、起動が成功したことが "Referencing active"（基準化アクティブ）インタフェース信号で確認応答された。系列別原点復帰が機能中。	
信号ステータス 0 あるいは信号遷移 1 ---> 0	• 系列別原点復帰が完了した。 • 軸別基準化が機能中。 • 基準化が非アクティブ。	
この信号は次の場合は無意味	スピンドル	
関連性	"activate referencing" interface signal (DB21, ... DBX1.0)	

■ チャンネルからの信号

DB21, ... DBX36.2 データブロック	All axes referenced（全軸基準化完了） 系列からの信号 (PLC → NCK)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 ---> 1	全軸（直線軸と回転軸）が原点復帰完了された。 MD 20700: REFP_NC_START_LOCK（基準点なしで NC スタート不可）がゼロ。 NC スタートを禁止するような 2 つの位置測定系が軸に接続されている場合は、軸を基準化したいのであれば、アクティブになっている方を基準化しなければならない。この信号が存在している場合に限って、パートプログラム処理の NC スタートコマンドが受入れられる。MD 34110: REFP_CYCLE_NR ≠ -1 で、軸がパーキング位置（位置測定系が非アクティブでコントローライネーブルがキャンセルされていれば）になければ、軸は基準点を有していなければならない。	
信号ステータス 0 あるいは信号遷移 1 ---> 0	系列上の一つあるいはそれ以上の軸がまだ基準化されていない。	
例外、エラー	系列のスピンドルはこのインタフェース信号に影響を与えない。	
関連性	"referenced/synchronized 1" interface signal (DB31, ... DBX60.4) "referenced/synchronized 2" interface signal (DB31, ... DBX60.5)	

■ 軸／スピンドル別信号

■ 軸／スピンドルへの信号

DB31, ... DBX2.4 to 2.7 データブロック	Reference point value 1 to 4（基準点値 1 ～ 4） 軸／スピンドルへの信号 (PLC → NCK)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 ---> 1	軸が基準点に到着すると、コード化されたどの基準カムがアプローチされたかが NCK に通知される。"reference point value 1 to 4" インタフェース信号は、基準点が到達されるまで、あるいは新しいコード化基準カムがアプローチされるまではイネーブルされたままになる。 軸が基準点に到着すると（軸停止）、MD 34100: REFP_SET_POS から "reference point value 1 to 4" インタフェース信号によって選択された基準点値が、制御装置によって新しい基準位置として受入れられる。	
信号ステータス 0 あるいは信号遷移 1 ---> 0	影響なし。	
この信号は次の場合は無意味	距離コード化基準マークを有する直線測定系。	
アプリケーション例	移動距離が長い工作機械では、軸が移動する全範囲にわたって 4 つのコード化された基準カムを配置して、4 つの異なった基準点にアプローチするようにすることで、有効な基準点に到着するのに要する時間を短くすることができる。	
例外、エラー	軸が基準点に到着したにもかかわらず、4 つの "reference point value 1 to 4" インタフェース信号のいずれもがイネーブルされなかった場合は、基準点の値は自動的に 1 にセットされます。	
関連性	MD 34100: REFP_SET_POS (reference point value)	

DB31, ... DBX12.7 **データブロック**	Reference point approach delay（**基準点アプローチ遅延**） **軸／スピンドルへの信号** (PLC → NCK)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 ---> 1	軸はリファレンスカム上にある。	
信号ステータス 0 あるいは信号遷移 1 ---> 0	マシン軸は基準カムの手前にある。適切な長さ（移動範囲の最後まで延びている）の基準カムが使用されているので、軸は基準カムの後ろにくることはない。	
関連性	"reference point value 1 to 4" interface signal (DB31, ... DBX2.4 to 2.7)	

■ 軸／スピンドルからの信号

DB31, ... DBX60.4 **データブロック**	Referenced/synchronized 1（**基準化済み／同期化済み** 1） **軸／スピンドルからの信号** (NCK → PLC)	
エッジ評価 :	信号更新 :	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 ---> 1	軸 : 原復帰中に、軸が基準点に到達したか（相対測定系の場合）、あるいは目標位置に到達したとき（距離コード化基準マークを有する直線測定系の場合）、軸は基準化されたと見なされ、"referenced/synchronized 1" インタフェース信号（原点復帰中にどちらの位置測定系がアクティブかによる）がイネーブルされる。 スピンドル : 電源オンの後で、ゼロマークを通過するかあるいは BERO 近接スイッチを通過すると、スピンドルが同期化される（一番早くはスピンドルが 360° 回転したときに同期化される）。	
信号ステータス 0 あるいは信号遷移 1 ---> 0	位置測定系 1 を有するマシン軸／スピンドルは基準化／同期化されていない。	
関連性	"position measurement system 1" interface signal (DB31, ... DBX1.5)	
参照	1.17「主軸機能（S1）」	

DB31, ... DBX60.5 **データブロック**	Referenced/synchronized 2（**基準化済み／同期化済み 2**） **軸／スピンドルからの信号 (NCK → PLC)**	
エッジ評価 :	信号更新 :	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 ---> 1	軸 : 原点復帰中に、軸が基準点に到達したか（相対測定系の場合）、あるいは目標位置に到達したとき（距離コード化基準マークを有する直線測定系の場合）、マシン軸は基準化されたと見なされ、"referenced/synchronized 2" インタフェース信号（基準化中にどちらの位置測定系がアクティブかによる）がイネーブルされる。 スピンドル : 電源オンの後で、ゼロマークを通過するかあるいは BERO 近接スイッチを通過すると、スピンドルが同期化される（一番早くはスピンドルが 360° 回転したときに同期化される）。	
信号ステータス 0 あるいは信号遷移 1 ---> 0	位置測定系 2 を有するマシン軸／スピンドルは基準化／同期化されていない。 軸 : アラーム 21610 が発行された。 スピンドル : エンコーダ限界周波数を超えた。	
関連性	"position measurement system 2" interface signal (DB31, ... DBX1.6), MD 34102 = 0	
参照	1.17 主軸機能（S1）	

1.1.6 例

なし

1.1.7 データフィールド、リスト

■ インタフェース信号

DB 番号	ビット、バイト	名称	参照
モードグループ別			
11, ...	1.2	REF マシンファンクション	1.10（K1）
11, ...	5.2	アクティブな REF マシンファンクション	1.10（K1）
チャンネル別			
1, ...	1.0	原点復帰起動	
21, ...	28.7	(MMC ---> PLC) REF	1.10（K1）
21, ...	33.0	原点復帰アクティブ	
21, ...	36.2	基準点を有していなければならない全軸が基準化されている。	
軸別			
31, ...	1.5 / 1.6	位置測定系 1/ 位置測定系 2	1.1（A2）
31, ...	2.4-2.7	基準点値 1 ～ 4	1.1（A2）
31, ...	12.7	原点復帰遅延	1.1（A2）
31, ...	60.4 / 60.5	基準化済み、同期化済み 1/ 基準化済み、同期化済み 2	1.1（A2）

■ マシンデータ

番号	識別子	名称	参照
一般事項 ($MN_ ...)			
11300	JOG_INC_MODE_LEVELTRIGGRD	JOG ／連続モードでの INC/REF	2.4（H1）
チャンネル別 ($machine code_ ...)			
20700	REFP_NC_START_LOCK	基準点なしの NC スタート不可	
軸別 ($MA_ ...)			
30200	NUM_ENCS	エンコーダの数	1.8（G2）
30240	ENC_TYP	エンコーダタイプ	
31122	BERO_DELAY_TIME_PLUS	プラス方向の BERO 遅延時間	1.17（S1）
31123	BERO_DELAY_TIME_MINUS	マイナス方向の BERO 遅延時間	1.17（S1）
34000	REFP_CAM_IS_ACTIVE	基準カム付きの軸	
34010	REFP_CAM_DIR_IS_MINUS	マイナス方向の原点復帰	
34020	REFP_VELO_SEARCH_CAM	原点復帰速度	
34030	REFP_MAX_CAM_DIST	基準カムまでの最大距離	
34040	REFP_VELO_SEARCH_MARKE[n]	原点復帰クリープ速度 [エンコーダ番号]	
34050	REFP_SEARCH_MARKER_REVERS[n]	基準カムへの方向変更 [エンコーダ番号]	
34060	REFP_MAX_MARKER_DIS[n]	基準マークまでの最大距離 ; 距離コード化スケールでの 2 つの基準マークまでの最大距離 [エンコーダ番号]	
34070	REFP_VELO_POS	原点復帰開始速度	
34080	REFP_MOVE_DIS[n]	距離コード化系用の基準点距離／目標位置 [エンコーダ番号]	
34090	REFP_MOVE_DIST_COR[n]	基準点／絶対オフセット、距離コード化 [エンコーダ番号]	
34092	REFP_CAM_SHIFT	等距離ゼロマーク付きの相対測定系用の電子基準カムシフト	
34100	REFP_SET_POS[n]	基準点値 [基準点番号]	
34102	REFP_SYNC_ENCS	基準測定系に対する実際値補償	
34110	REFP_CYCLE_NR	軸別基準化時の軸の順番	
34200	ENC_REFP_MOD[n]	基準化モード [エンコーダ番号]	
34210	ENC_REFP_STAT[n]	絶対値エンコーダのステータス [エンコーダ番号]	
34220	ENC_ABS_TURNS_MODULO	回転エンコーダ用の絶対値エンコーダ範囲	2.12（R2）
34300	ENC_REFP_MARKER_DIS[n]	距離コード化スケール（目盛付き）用の基準マーク距離 [エンコーダ番号]	
34310	ENC_MARKER_INC[n]	距離コード化スケール（目盛付き）上での 2 つの基準マーク間の距離 [エンコーダ番号]	
34320	ENC_INVER[Encode]	機械系とは逆方向の直線測定系 [エンコーダ番号]	
34330	REFP_STOP_AT_ABS_MARKE[n]	目標位置のない距離コード化直線測定系 [エンコーダ番号]	

番号	識別子	名称	参照
軸別 ($MA_ ...)			
35150	SPIND_DES_VELO_TOL	スピンドル速度公差	1.17（S1）
36302	ENC_FREQ_LIMIT_LOW	エンコーダ再同期限界周波数	
36310	ENC_ZERO_MONITORING	ゼロマーク監視	
30250	ACT_POS_ABS	停止時の絶対エンコーダの位置	

■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

1.17 主軸機能 (S1)

1.17.1 概略説明

スピンドルのアプリケーション

スピンドルの基本的な機能は，加工を容易にするためにツールまたはワークに回転動作を与えるものです。

そのため，マシンのタイプにより，スピンドルは次の機能をサポートしなければなりません。

スピンドルを回転させる方向の定義 (M3, M4)

スピンドル速度の定義 (S...)

方向づけ（オリエンテーション）を伴わないスピンドルの停止 (M5)

方向づけ（オリエンテーション）／スピンドル位置決めを伴うスピンドルの停止 (SPOS および SPOSA)

ギアチェンジ (M40 ～ M45)

スピンドル／軸の機能性（スピンドルが回転軸に，または回転軸がスピンドルになる）

ねじ切り

補正チャックを使用した，または使用しないタッピング

回転フィードレート (G95)

定切削速度 (G96)

プログラム可能なスピンドル速度制限 (G25, G26, LIMS)

スピンドルまたはスピンドルモータ上の位置エンコーダアセンブリ

スピンドルの最低速度および最高速度監視

位置制御のオン／オフ切換え (SPCON, SPCOF, M70)

1.17.2 詳細説明

スピンドルの最大数

スピンドルは有効なチャンネルに割当てられます。

■ スピンドルモード

スピンドルモード

スピンドルには以下のモードがあります：

- 制御モード　　1.17.2「■スピンドル制御モード」を参照してください
- 発振モード　　1.17.2「■スピンドルモード発振モード」を参照してください
- 位置決めモード　　1.17.2「■スピンドル位置決めモード」を参照してください
- 同期モード
  同期スピンドル (S3)　　2.13「主軸同期」を参照してください
- リジッドタッピング　　1.18.2「■ G331, G332 用フィード」を参照してください

参照： YS840DI ユーザーズマニュアル・プログラミング編

軸モード

スピンドルと軸のオペレーションに同じモータが使われていれば，スピンドルはスピンドルモードから軸モード（回転軸）へ切換えができます。

- 軸モード　　1.17.2「■軸モード」を参照してください

1) メモリ 2 が 原因

スピンドルモードの変更

インタフェース信号またはプログラミングコマンドは，スピンドルモードと軸オペレーションの切換えに使われます。

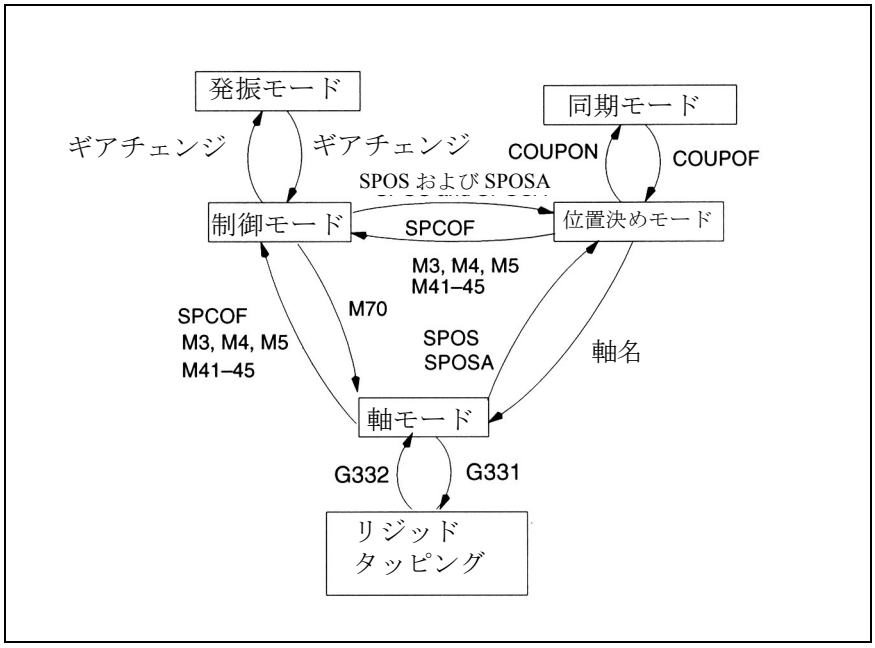


図 1.151 スピンドルモードの切換え

- 制御モード → 発振モード
  新しい S 値と関連した自動ギアステップ選択 （M40）を使用するか，あるいは M41 ～ M45 を使用した新しいギアステップが指定されると，スピンドルは発振モードに変わります。スピンドルは新しいギアステップが現在の実際のギアステップと同じでない場合のみ，発振モードに変わります。
- 発振モード → 制御モード
  新しいギアを使用中の場合，「発振モード」インタフェース信号 (DB31, ... DBX84.6) はリセットされ，スピンドルは「ギアチェンジ」インタフェース信号 (DB31, ... DBX16.3) で制御モードに変更されます。最後にプログラムされたスピンドル速度（S 値）に戻ります。
- 制御モード → 位置決めモード
  回転中のスピンドルを方向づけ（M3 または M4）して停止させるか，あるいは停止中のスピンドルの方向づけ（M5）を変更する必要がある場合，SPOS または SPOSA を使用して制御モードを位置決めモードに変更できます。
- 位置決めモード → 制御モード
  M3, M4 または M5 は，スピンドルの方向が解除される場合，制御モードに変更するために使用します。最後にプログラムされたスピンドル速度（S 値）は復帰します。
- 位置決めモード → 発振モード
  スピンドルの方向が解除される場合，M41 ～ M45 は発振モードに変更するために使用します。ギアチェンジが完了すると，最後にプログラムされたスピンドル速度（S 値）と M5 （制御モード）は復帰されます。
- 位置決めモード → 軸モード
  スピンドルが方向を伴って停止した場合，割当てられた軸名は，軸モードへの

変更をプログラムするために使用します。ギアステップは維持されます。

- 制御モード → 軸モード
  制御モードから軸モードへの変更は，プログラミング M70 により行われます。この場合，回転するスピンドルが M5 と同じ方法で減速され，位置制御が起動され，ゼロパラメータセットが選択されます。
- 軸モード → 制御モード
  軸モードから制御モードにするには，M3, M4 または M5 を使用します。最後にプログラムされたスピンドル速度（S 値）に戻ります。
- 軸モード → 発振モード
  軸モードを発振モードにするには，M41 ～ M45 を使用します（プログラムされたギアステップが実際のギアステップと同じでないときのみ)。ギアチェンジが完了すると，最後にプログラムされたスピンドル速度（S 値）と M5 （制御モード）に戻ります。

■ スピンドル制御モード

制御モードはどんなとき使われるか？

スピンドルが以下の機能を有する制御モードにあるとき：

- 定スピンドル速度 S.... M3/M4/M5 および G93, G94, G95, G97
- 定切削速度 G96 S.... M3/M4/M5
- 定スピンドル速度 S.... M3/M4/M5 および G33
- 定スピンドル速度 S.... M3/M4/M5 および G63

要求事項

- スピンドルは同期させる必要はありません
- スピンドル位置を検出する実際値エンコーダは，タイムレシプロカルフィードコーディング (G93)，mm/min または inch/min で表示されるフィードレート (G94) およびリジッドタッピング (G63) 用には M3/M4/M5 では必要ありません。
- スピンドル位置を検出する実際値エンコーダは，回転フィードレート (G95), 定切削速度 (G96, G97), ねじ切り (G33), タッピング (G331, G332)，およびアクティブ位置制御 (SPCON, M70) 用には M3/M4/M5 では必ず必要です。

位置制御オン／オフ

スピンドルは位置制御を伴って，または伴わずに作動することができます。

- SPCON （位置制御オン）
- SPCOF （位置制御オフ）

速度制御モード

速度制御モードは，定スピンドル速度が必要で，スピンドルの位置は関係ない（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）場合に特に適しています。

- 速度制御モードは M3, M4, M5 または SPCOF を有するパートプログラムで起動します。
- "control mode"（制御モード）インタフェース信号 (DB31, ... DBX84.7) がイネーブルになります。
- 位置制御なしで動作させると，"position controller active（位置コントローラアクティブ）インタフェース信号 (DB31, ... DBX61.5) がリセットされます。
- 加速は MD 35200: GEAR_STEP_SPEEDCTRL_ACCEL （位置制御モードでの加速）のギアステップに従って定義され，物理的条件に対応します。

「位置コントローラアクティブ」

DB31, ... DBX61.5

位置制御は，スピンドル位置が長い期間制御される必要がある場合，または同期スピンドルのセットポイント値リンケージが起動される場合に特に適しています。

- 位置制御モードは起動 SPCON（スピンドル番号）を使用してパートプログラムで起動します。
- 「位置コントローラアクティブ」インタフェース信号 (DB31, ... DBX61.5) はイネーブルになります。
- 加速は MD 35210: GEAR_STEP_POSCTRL_ACCEL （位置制御モードでの加速）に定義されています。

一般スピンドルリセット

MD 35040: SPIND_ACTIVE_AFTER_RESET にはリセットまたはプログラム終了後 (M2, M30) のスピンドルの動きが定義されています。

- SPIND_ACTIVE_AFTER_RESET がリセットされると，スピンドルはアクティブ加速で，直ちに減速停止します。最後にプログラムされたスピンドル速度と回転方向は削除されます。
- SPIND_ACTIVE_AFTER_RESET がイネーブルになると，先にプログラムされたスピンドル速度（S 値）と最後にプログラムされたスピンドル回転（M3, M4, M5）は維持されます。スピンドルは減速しません。

  定切削速度 (G96) がリセットまたはプログラム終了前にアクティブになると，現行のスピンドル速度（100% スピンドルオーバライドを基準にして）が最後にプログラムされたスピンドル速度として受入れられます。スピンドルは，"delete distance-to-go, spindle rest" （移動距離削除，スピンドルリセット）インタフェース信号 (DB31, ... DBX2.2) を使用して，パートプログラムあるいは FC18 の外部で停止させることができます。

■ スピンドルモード発振モード

発振とは？

本書でいう発振とは，スピンドルモータが時計回りと反時計回りに交互に回転することをいいます。この発振動作は新しいギアステップに切換えるのを容易にします。

発振モード

- MD 35010: GEAR_STEP_CHANGE_ENABLE はイネーブルでなければなりません。
- "oscillation mode"（発振モード）インタフェース信号 (DB31, ... DBX84.6) はイネーブルになります。
- 加速は MD 35410: SPIND_OSCILL_ACCEL に定義されています。

IS 「発振速度」DB31, ... DBX18.5

スピンドルは新しいギアステップが自動ギアステップ選択 (M40) または M41 ～ M45 （「ギアチェンジ」インタフェース信号がイネーブル）を使って定義されると発振モードに入ります。「ギアチェンジ」インタフェース信号 (DB31, ... DBX82.3) は新しいギアステップが現行の実際のギアステップと同じでない場合のみイネーブルになります。

「発振速度」インタフェース信号が新しいギアステップを定義せずにイネーブルになった場合，スピンドルは発振モードに変わりません。

発振は「発振速度」インタフェース信号 (DB31, ... DBX18.5) を伴って開始します。「PLC を介しての発振」インタフェース信号 (DB31, ... DBX18.4) の設定は次の場合を区別します。

- NCK を介しての発振
- PLC を介しての発振
- FC 18 （機能の説明 P3 を参照）を伴う発振

基本的に，新しいギアステップは発振させなくても噛み合います。

発振時間

発振モードの発振時間は各回転方向のマシンデータに定義されます。

- MD 35440: SPIND_OSCILL_TIME_CW の M3 方向での発振時間（以下 tl とする）
- MD 35450: SPIND_OSCILL_TIME_CCW の M4 方向での発振時間（以下 t2 とする）

NCK を介しての発振

相 1:

「発振速度」インタフェース信号 (DB31, ... DBX18.5) では，スピンドルモータは MD 35400: SPIND_OSCILL_DES_VELO （発振速度）に定義される速度まで加速（発振加速を伴って）します。開始方向は MD 35430: SPIND_OSCILL_START_DIR で定義されています。時間 t1 （または t2）は MD 35430: SPIND_OSCILL_START_DIR で定義されている開始方向に従って開始されます。その時間は常に決定的であり，発振速度に達したということではありません。

相 2:

時間 t1 (t2) が終了するとき，スピンドルモータは MD 35400: SPIND_OSCILL_DES_VELO に定義されている速度に反対方向に加速します。時間 t2 (t1) が開始します。

相 3:

時間 t2 (t1) が終了するとき，スピンドルモータは MD 35400: SPIND_OSCILL_DES_VELO で定義されている速度に反対側に（相 1 と同じ方向）加速します。時間 t1 (t2) が開始します。相 3 が終わると相 2 に戻ります。

PLC を介しての発振

「発振速度」インタフェース信号で，スピンドルモータは MD 35400: SPIND_OSCILL_DES_VELO （発振速度）に定義されている速度に加速（発振加速を伴って）します。回転の方向は，"set counterclockwise" （逆時計回りに設定）インタフェース信号 (DB31, ... DBX18.7) および "set direction of rotation" （回転方向を設定）インタフェース信号 (DB31, ... DBX18.6) で定義されています。

発振動作および t1 と t2 の 2 つの時間（時計回りおよび逆時計回り回転）は PLC においてシミュレートされる必要があります。

発振モードの終了

「ギアチェンジ」インタフェース信号 (DB31, ... DBX16.3) は NCK に新しいギアステップ（「実際のギアステップ」インタフェース信号 (DB31, ... DBX16.0-16.2)）が有効であり，発振が終了したことを知らせます。「発振速度」インタフェース信号 (DB31, ... DBX18.5) がまだイネーブルであるかどうかは関係ありません。実際のギアステップは設定ギアステップに対応していなければなりません。報告された実際のギアステップはパラメータの設定の選択に関連しています。最後にプログラムされたスピンドル速度（S 値）とスピンドルの回転方向 (M3, M4 または M5) は再びアクティブになります。

発振モードの終了で，スピンドルは制御モードに戻り，SPCON または SPCOF に定義されているモードに従って自動的にオペレーションを始めます。

すべてのギア別制限値（ギアステップの最低／最高速度など）は実際のギアステップ用に定義された値に対応し，スピンドルが停止するとアクティブでなくなります。

ブロックチェンジ

スピンドルが発振モードに切換えられると（「ギアチェンジ」インタフェース信号 (DB31, ... DBX82.3) がイネーブルになる），パートプログラムの実行が中断されます。新しいブロックは実行されません。発振モードが「ギアチェンジ」インタフェース信号を伴って終了すると，図 1.152 にあるようにパートプログラムの実行が再開します。新しいブロックが実行されます。

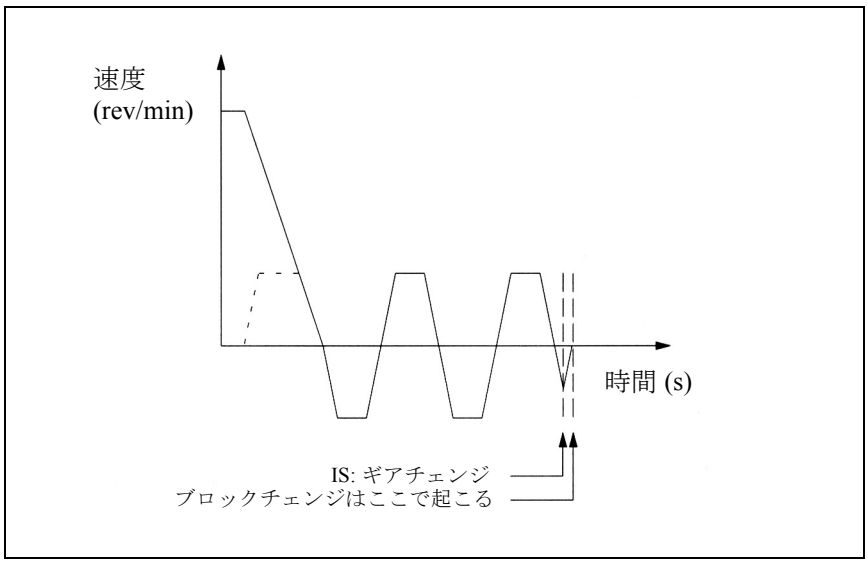


図 1.152 発振モードに続くブロックチェンジ

特別機能

- 加速は MD 35410: SPIND_OSCILL_ACCEL で定義されています。
- 「発振速度」インタフェース信号 (DB31, ... DBX18.5) がリセットされると，発振は停止します。しかし，スピンドルは発振モードを維持します。
- ギアチェンジは常に「ギアチェンジ」インタフェース信号 (DB31, ... DBX16.3) を伴って完了しなければなりません。
- 同期は間接的測定系で消失します。MD 31050: ENC_IS_DIRECT = 0 であるとき，「基準化／同期化」インタフェース信号 (DB31, ... DBX60.4) は消去されます。ゼロマークは次回交差したとき基準化されます。

ギアチェンジ中のリセット

以下において，

- 「リセット」インタフェース信号 (DB21, ... DBX7.7)
- 「NC 停止」インタフェース信号 (DB21, ... DBX7.4)

次の場合，スピンドルを停止させることはできません。

- スピンドルがギアチェンジのために発振モードにある
- 「ギアチェンジ」インタフェース信号 (DB31, ... DBX16.3) がない

これらの場合，リセットを選択するとアラーム 10640 "spindle cannot stop during gear change" （ギアチェンジ中にスピンドルは停止できません）が表示されます。

ギアステップが変更されている場合，リセット要求は実行され，アラームがまだインタフェースにある場合はキャンセルされます。

（注）強制終了のオプション：
実際のギアステップ（「実際のギアステップ」インタフェース信号 (DB31, ... DBX16.0, 1,2)）に対応する確認応答を有する「移動距離を削除，スピンドルリセット」インタフェース信号 (DB31, ... DBX16.3)

■ スピンドル位置決めモード

位置決めモードが使われるのは？

スピンドル位置決めモードは定義された位置にスピンドルを停止させ，位置制御を解除するまでアクティブにします。スピンドルは次の機能を伴って位置決めモードに入ります。n = スピンドル番号であるパラメータ [n] は，主スピンドルには必要ありません。

- 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] = ...
- SPOSA [n] = AC(.....)
- SPOSA [n] = IC(.....)
- SPOSA [n] = DC(.....)

SPOS [n] = AC(.....)

絶対的位置へのスピンドル位置決め（0 ～ 359.999 度）。位置決め方向は，現在のスピンドル回転（スピンドルが回転している）方向または移動距離によって決まります。

SPOS [n]=IC(.....)

最後にプログラムされた位置と関連した相対位置 (+/- 999999.99 度) へのスピンドル位置決め。位置決め方向は，移動させる距離の表示により定義されます。

SPOS [n] = DC(.....)

絶対的位置決めへの（0 ～ 359.999 度）最短パスを通るスピンドル位置決め。位置決め方向は，現在のスピンドル回転（スピンドルが回転している）方向，または自動的に制御（スピンドルが静止）によって決まります。

SPOS[n] = ...

SPOS [n] = DC(.....) と同一の機能。

SPOS [n] = ACP(.....)

プラス方向から位置にアプローチ。

位置決めが負の回転方向をなすとき，スピードは停止させられ，正のアプローチを実行させるために反対方向に加速させられます。

SPOS [n]=ACN(....)

負の方向からアプローチする位置にアプローチ。

位置決めが正の回転方向をなすとき，スピードは停止し，負のアプローチを実行する反対方向に加速します。

ブロックチェンジ

SPOS および SPOSA は，ブロックチェンジに関して以外は同じ機能を持っています：

- SPOS を使用したプログラミング

  ブロックチェンジは，ブロック中にプログラムされたすべての機能がブロックエンド基準（すべての補助機能が PLC によって確認応答され，すべての軸が終了点に到達する，など）を取得し，スピンドルがその位置（「スピンドル用正確停止（微）」インタフェース信号 (DB31, ... DBX60.7)）に到達したとき起こります。

- SPOSA を使用したプログラミング

  ブロックチェンジはブロック中（スピンドルを除く）にプログラムされたすべての機能がブロックエンド基準を取得すると起こります。SPOSA がブロック中に唯一入力されると，ブロックチェンジが即座に実行されます。その時，スピンドル位置決めは複数のブロックに渡って行われます（WAITS を参照）。

コーディネーション

動作のシーケンスのコーディネーションは次で行われます：

- 主スピンドルの WAITS
- 主スピンドルなどのスピンドルの WAITS[n]
- スピンドルの最大数までの複数のスピンドルの WAITS[n, m,...,q]

次のブロックの実行は，スピンドルまたは SPOS を使用してプログラムされたスピンドルが位置（正確停止（微））に達するまで保留されます。

フィード

位置決め速度は MD 35300: SPIND_POSCTRL_VELO で設定され，以下のプログラミングにより変更することができます：

- FA[Sn]　　n = スピンドル番号
- FA[Sn]=0　　設定された速度が起動される

スピードは deg/min で示されます。

回転からの位置決め

スピンドルは位置決めが（プログラム中の SPOS または SPOSA コマンドによって）開始されると，速度制御モードまたは位置制御モードに入ります。次のシーケンスが得られます：

- ケース 1: スピンドルは制御モード，エンコーダリミット周波数オーバ
- ケース 2: スピンドルは制御モード，エンコーダリミット周波数オーバしない
- ケース 3: スピンドルは制御モード
- ケース 4: スピンドル速度 < 位置制御起動速度

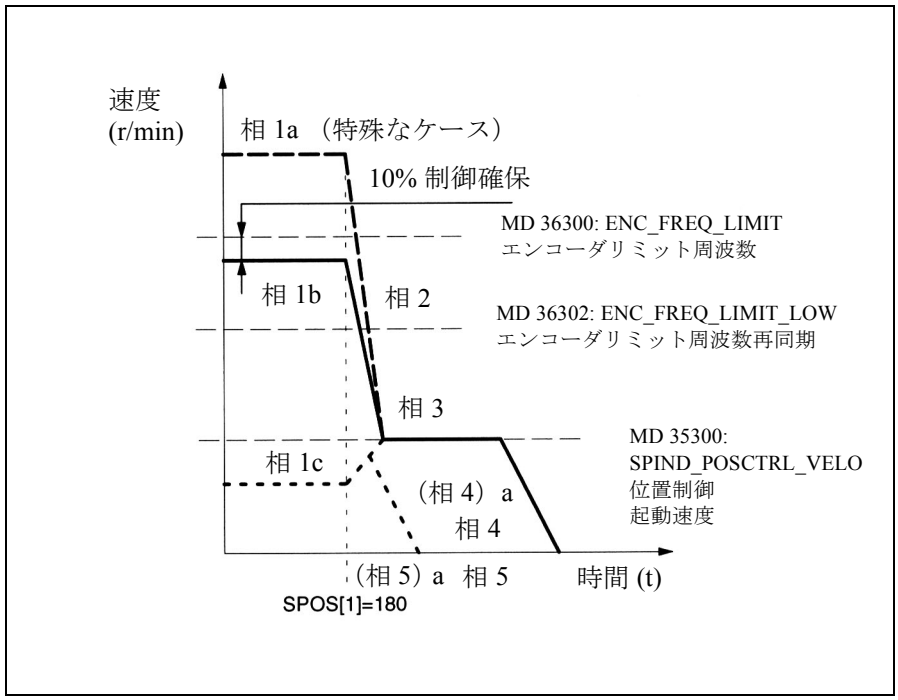


図 1.153 回転からの位置決め

（注）MD 36302: ENC_FREQ_LOW 中の速度は位置制御起動速度 (MD 35300: SPIND_POSCTRL_VELO) より速くなければなりません。

相 1

相 1a から可能な位置決め：

スピンドルがエンコーダリミット周波数より高速で回転。同期化されていない。

相 1b から可能な位置決め：

スピンドルがエンコーダリミット周波数より低速で回転。同期化されている。

（注）位置制御が起動されると，速度はスピンドル最高速度またはエンコーダリミット周波数の 90% までしか上がりません（10% 制御確保が必要です）。

相 1c から可能な位置決め：

スピンドルはプログラムされたスピンドル速度で回転し，それは MD 35300: SPIND_POSCTRL_VELO （位置制御起動速度）で定義されている速度より低速。同期化されている。

相 2

- スピンドル速度 > 位置制御起動速度

  SPOS コマンドが有効になると，スピンドルは MD 35200: GEAR_STEP_SPEEDCTL_ACCEL に保存されている加速から位置制御起動速度にブレーキを始めます。速度がエンコーダリミット周波数を下回ると，スピンドルは同期化されます。

- スピンドル速度 < 位置制御起動速度

  SPOS または SPOSA のプログラミングはスピンドルを位置制御モード（すでにこのモードになっていなければ）に切換えます。
  MD 35210: GEAR_STEP_POSCTRL_ACCEL （位置制御モードでの加速）が起動します。移動先への移動パスが計算されます。

  移動先に可能な限り最大速度（MD 35300: SPIND_POSCTRL_VELO より高速ではない）で到達するなど，スピンドルはプログラムされた移動先に時間に関しては最適に移動します。適当な補足条件により，相 2 - 3 - 4 - 5 または 2 - 4a - 5a でのオペレーション上のシーケンスが実行されます。

相 3

- スピンドル速度 > 位置制御起動速度

  MD 35300: SPIND_POSCTRL_VELO で定義された位置制御起動速度に到達すると，

  - 位置制御が起動し（すでにアクティブでなければ），

  - 移動距離（移動先までの）が計算され

  - MD 35210: GEAR_STEP_POSCTRL_ACCEL （位置制御モードでの加速）に定義されている加速が起動または維持されます。

- スピンドル速度 < 位置制御起動速度

  移動先に到達するため，スピンドルは MD 35300: SPIND_POSCTRL_VELO（位置制御起動速度）に設定されている速度まで加速します。このスピードは超過されません。ブレーキング開始点計算により，どの瞬間にスピンドルが MD 35210: 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位置にアプローチできるかが検出されます。

相 4

- スピンドル速度 > 位置制御起動速度

  スピンドルは MD 35210: GEAR_STEP_POSCTRL_ACCEL で計算されたブレーキング点から移動先までブレーキをかけます。

- スピンドル速度 < 位置制御起動速度

  相 3 のブレーキ開始点で検出された時間に，スピンドルは MD 35210: GEAR_STEP_POSCTRL_ACCEL で設定された加速から停止になるまでブレーキをかけます。

相 4a:

SPOS コマンドが有効になると，移動先はスピンドルが MD 35300: SPIND_POSCTRL_VELO の値まで加速されるほど接近します。スピンドルは MD 35210: GEAR_STEP_POSCTRL_ACCEL に設定されている加速で停止になるまでブレーキをかけます。

相 5

- スピンドル速度 > 位置制御起動速度

  位置制御はアクティブな状態を維持し，スピンドルをプログラムされた位置に保ちます。

  （注）スピンドル位置実際値エンコーダの最高エンコーダリミット周波数が制御（オーバステップが可能）によりモニターされます。位置制御モードでは，指令速度が測定系制限速度の 90% まで減少します。この場合，「プログラムされた速度が速すぎる」インタフェース信号がイネーブルになります。"MS limit frequency exceeded" （MS リミット周波数オーバ）が指令速度の減少後に検知されると，アラームが出力されます。

- スピンドル速度 < 位置制御起動速度（相 5, 5a）

  スピンドルは静止し，移動先に到達しています。位置制御がアクティブになり，スピンドルをプログラムされた位置に保ちます。

スピンドル実際位置とプログラムされた位置（スピンドルセットポイント位置）の距離が正確停止（微／粗）制限（MD 36010: STOP_LIMIT_FINE および MD 36000: STOP_LIMIT_COARSE に定義されている）の設定より短い場合，「正確停止（微／粗）を使用して到達した位置」インタフェース信号 (DB31, ... DBX60.6 および DBX60.7) が設定されます。

（注）「正確停止（微)」が報告されると，位置決め処理が完了したと考えられます。これはパートプログラムからの SPOS，FC 18 を伴う PLC によるシンクロナイズドアクション，およびスピンドル位置決めに適用されます。

静止からの位置決め

静止からの位置決め

静止からの位置決めについては 2 つのケースで異なります。

- ケース 1: 同期化されていません。スピンドルが制御と運転（ツール変更などのため）の切換え後に位置決めされるのはこのケースです。
  MD 31040: ENC_IS_DIRECT = 0.
- ケース 2: 同期化されています。制御と運転の切換え後，スピンドルが M3 または M4 を伴う最低 1 回の回転を通じて回転し，M5 （ゼロマークを伴う同期）を伴って停止するのはこのケースです。

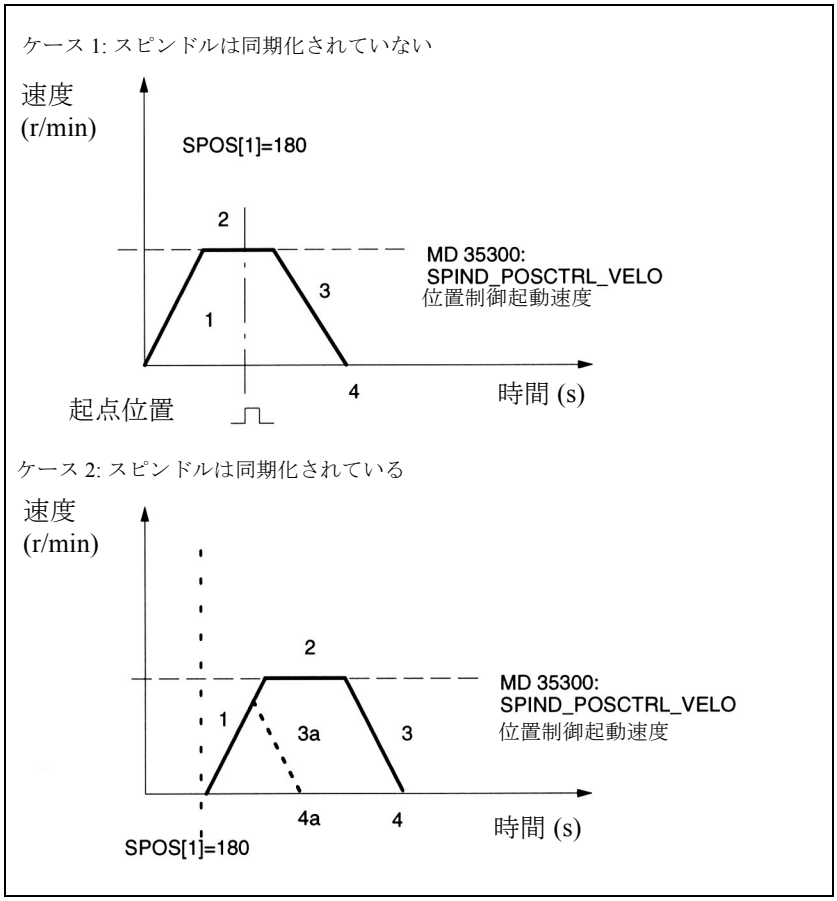


図 1.154 静止スピンドルを使用した位置決め

相 1

- ケース 1: スピンドルは同期化されていない

  SPOS または SPOSA がプログラムされる時，スピンドルは MD 35200: GEAR_STEP_POSCTRL_ACCEL （位置制御モードでの加速）からの加速度で加速します。回転方向は MD 35350: SPIND_POSITIONING_DIR に定義されています。

  例外：ACN, ACP, IC が位置決めに使われる場合，プログラムされた移動方向が起動します。

  スピンドルはスピンドル位置実際値エンコーダの次の起点位置を伴って同期し，位置制御モード（1.17.2「■基準化／同期化」も参照）に切換わります。MD 34060: REFP_MAX_MARKER_DIST に保存されているパス中の起点位置は，それが検知されるのを確認するために監視されます（IC を除く）。MD 35300: SPIND_POSCTRL_VELO （位置決め速度）に入力されている速度に到達したがスピンドルがまだ同期化されていない場合，スピンドルは位置決めクリープ速度（スピンドルがさらに加速されない）で回転し続けます。

- ケース 2: スピンドルは同期化されている

  SPOS または SPOSA のプログラミングはスピンドルを位置制御モードに切換えます。MD 35210: GEAR_STEP_POSCTRL_ACCEL （位置制御モードでの加速）からの加速度が有効になります。回転方向はプログラムされた動作 (ACP;

ACN; IC; DC) または移動距離により定義されています。MD 35300: SPIND_POSCTRL_VELO （位置制御起動速度）にある速度は超過されません。移動先への移動パスが計算されます。

移動先に可能な限り最大速度（MD 35300: SPIND_POSCTRL_VELO より高速ではない）で到達するなど，スピンドルはプログラムされた移動先へ時間に関しては最適に移動します。適当な補足条件により，相 1 - 2 - 3 - 4 または 1 - 3a - 4a でのオペレーション上のシーケンスが実行されます。

相 2

- ケース 1: スピンドルは同期化されていない

  スピンドルが同期化されると，位置制御が起動されます。スピンドルは，プログラムされたスピンドル位置に定義された加速で正確にアプローチすることができるブレーキング開始点が検知されるまで MD 35300: SPIND_POSCTRL_VELO で定義されている速度で回転します。

- ケース 2: スピンドルは同期化されている

  移動先に到達するため，スピンドルは MD 35300: SPIND_POSCTRL_VELO で設定されているまで加速します。ブレーキング開始点計算により，どの瞬間にスピンドルが MD 35210: GEAR_STEP_POSCTRL_ACCEL で定義されている加速度で，正確にプログラムされた位置にアプローチできるかが検知されます。

  相 1 のブレーキ開始点で検出された瞬間に，スピンドルは MD 35210: GEAR_STEP_POSCTRL_ACCEL で設定された加速から停止になるまでブレーキをかけます。

相 3

相 1 のブレーキ開始点で検出された瞬間に，スピンドルは MD 35210: GEAR_STEP_POSCTRL_ACCEL で設定された加速度から停止になるまでブレーキをかけます。

相 3a:

SPOS コマンドが有効になると，移動先はスピンドルが MD 35300: SPIND_POSCTRL_VELO の値まで加速されるほど接近します。スピンドルは MD 35210: GEAR_STEP_POSCTRL_ACCEL に設定されている加速度で停止になるまでブレーキをかけます。

相 4, 4a

スピンドルは静止し，移動先に到達しています。位置制御がアクティブになり，スピンドルをプログラムされた位置に保ちます。スピンドル実際位置とプログラムされた位置（スピンドルセットポイント位置）の距離が正確停止（微／粗）制限（MD 36010: STOP_LIMIT_FINE および MD 36000: STOP_LIMIT_COARSE に定義されている）の設定より短い場合，「正確停止（微／粗）を使用して到達した位置」インタフェース信号 (DB31, ... DBX60.6 および DBX60.7) が設定されます。

相 3:

相 2 のブレーキ開始点で検出された瞬間に，スピンドルは MD 35210: GEAR_STEP_POSCTRL_ACCEL で設定された加速度から停止になるまでブレーキをかけます。

相 4:

スピンドルは静止し，移動先に到達しています。位置制御がアクティブになり，スピンドルをプログラムされた位置に保ちます。スピンドル実際位置とプログラムされた位置（スピンドルセットポイント位置）の距離が正確停止（微／粗）制限（MD 36010: STOP_LIMIT_FINE および MD 36000: STOP_LIMIT_COARSE に定義されている）の設定より短い場合，「正確停止（微／粗）を使用して到達した位置」インタフェース信号が設定されます。

位置決め処理の強制終了

位置決め処理は「移動距離削除／スピンドルリセット」インタフェース信号で中断させることができます。位置決め処理は MD 35040: SPIND_ACTIVE_AFTER_RESET （独自のスピンドルリセット）とは別に，各リセット（オペレータパネルリセット／「移動距離削除／スピンドルリセット」インタフェース信号）で中断されます。

特別機能

- 加速度は以下のマシンデータに定義されています。
  MD 35210: GEAR_STEP_POSCTRL_ACCEL （位置制御モードでの加速）
  MD 35200: GEAR_STEP_SPEEDCTRL_ACCEL （速度制御モードでの加速）
- スピンドルオーバライドスイッチはアクティブです。
- 位置決め (SPOS および SPOSA) は各「リセット」でキャンセルされます。
- NC 停止で位置決めはキャンセルされます。
- 位置決め速度は FA[Sn] を使用してプログラムすることもできます。

■ 軸モード

軸モードを使うのは？

特定の加工作業においては（端面加工を伴う旋盤など）, スピンドルは M3, M4 および M5 を伴って回転し SPOS および SPOSA を伴って位置決めされなければならないだけでなく，特有の識別子（例，"C"）を有する軸として扱われなければなりません。

要求事項

- 同じスピンドルモータがスピンドルモードと軸モードに使われます。
- 同じ位置測定系または別の位置測定系がスピンドルモードと軸モードで使うことができます。
- 位置実際値エンコーダは軸モードの強制的要求事項です。
- 位置制御が M70 を使用して起動されるなど，軸が同期化されていないのであれば，まず G74 を参照しなければなりません。そうしたときのみ機械的位置がプログラムされた位置と一致します。

例 :
M70
G74 C1=0 Z100
G0 C180 X50

機能

軸モードがアクティブで回転軸が参照されれば，全ての軸機能を使うことができます。

参照： 2.12 回転軸（R2）

最も重要な機能は :

- 軸名を有するプログラミング
- ゼロオフセット (G54, G55, TRANS, ...) の使用
- G90, G91, IC, AC, DC, ACP, ACN
- 運動変換（例，伝送）の使用
- 他の軸を有する補間（パス補間）
- 位置決め軸としてのプログラミング

特別機能

- フィードオーバライドスイッチがアクティブです。
- 「リセット」インタフェース信号 (DB21, ... DBX7.7) は軸モードを終了させません。
- 「軸／スピンドルでない」インタフェース信号 (DB31, ... DBX60.0) がゼロに設定されている場合，DB31, ... 中のインタフェース信号 DBB16 ～ DBB19 および DBB82 ～ DBB91 は不適切です。
- 軸モードはすべてのギアステップにおいて起動できます。位置実際値エンコーダがモータ（間接測定系）にインストールされていれば，位置決めと輪郭精度はギアステップにより異なります。
- ギアステップは軸モードがアクティブであると変更できません。スピンドルは M41 ～ M45 を使用して制御モードに切換えなければなりません。
- 軸モードでは，インデックスがゼロであるパラメータの設定の MD は応答を実行するために有効になります。

軸モードへの移行

- 軸名を使用してまたは M70 を介してスピンドルをプログラミングすることによる軸モードへの移行
- 以下はパラメータセットを変更するための関連データです。
  MD 31050: DRIVE_AX_RATIO_DENOM （分母測定ギアボックス）
  MD 31060: DRIVE_AX_RATIO_NUMERA （分子負荷ギアボックス）
  MD 32200: POSCTRL_GAIN （KV 係数）
  MD 32800: EQUIV_CURRCTRL_TIME （フィードフォワード制御用の等価時定数電流制御回路）
  MD 32810: EQUIV_SPEEDCTRL_TIME (フィードフォワード制御用の速度制御回路の定等価時間)
  MD 32910: DYN_MATCH_TIME （動的マッチング用の定時間）
  MD 36200: AX_VELO_LIMIT （速度監視制限）
  PLC によるパラメータセット変更については以下を参照してください。

参照： 1.1 各種インタフェース信号（A2）

- マシンデータに保存されている軸の動的制限は軸オペレーションに適用できます。
- 軸は MD と FFWON および FFWOF コマンドに指定されているように現在のフィードフォワード制御に切換わります。

スピンドルモードへの変更

- パラメータセット（設定 1 ... 5）は現在有効なギアステップに従って選択されます。
- フィードフォワード制御機能は補正チャックを使用するタッピングを除いて常に起動されています。MD 32620: FFW_MODE （フィードフォワードタイプ）は常に 0 以外の値でなければなりません。
- フィードフォワード制御は，位置決めの最中にアラームが発生することがあるため , 常に 100% の値でオペレーションされなければなりません。

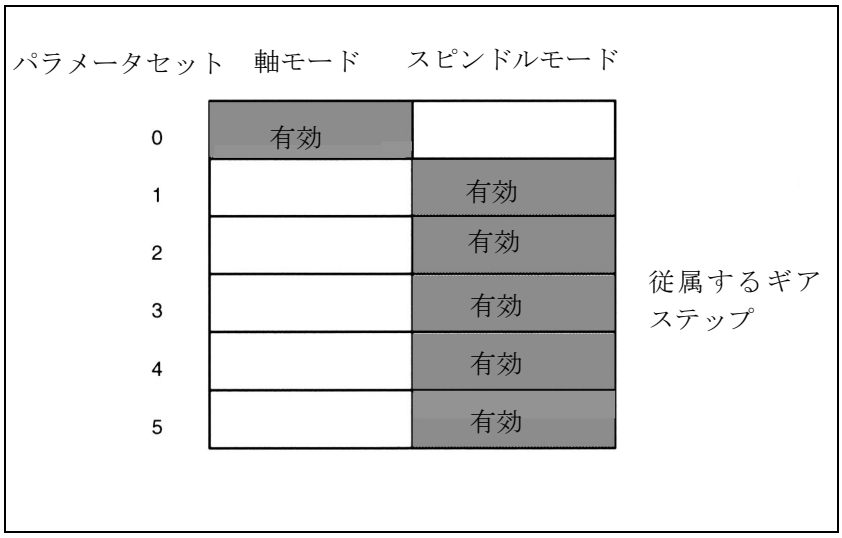


図 1.155 軸およびスピンドルモードのパラメータセットの有効性

■ デフォルトモード設定

マシンデータ

スピンドル運転モードのデフォルト設定を使用すると，電源オン，NC-START，リセット後の基本スピンドル設定が以下のマシンデータによって設定されます。

MD 35020: SPIND_DEFAULT_MODE および
MD 35030: SPIND_DEFAULT_ACT_MASK

マシンデータの設定については 1.17.4「■軸／スピンドル別マシンデータ」を参照してください。

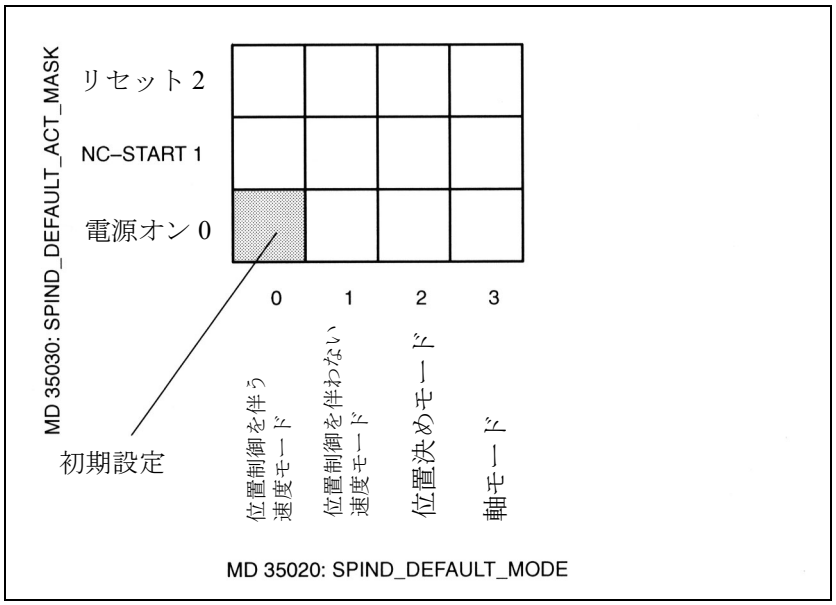


図 1.156 スピンドル運転モードの設定可能なデフォルト

■ 基準化／同期化

（注）スピンドル同期化と回転軸基準化については，以下を参照してください。

参照： 1.16 原点復帰（R1）

なぜ同期化するのか？

スイッチがオンになったときの正確な位置を知るために，制御装置はスピンドル上の位置測定系で同期化される必要があります。この処理を同期化といいます。

以下の場合，同期化されたスピンドルのみが使われます。

- スレッド切削
- リジッドタッピング
- 軸としてのプログラミング

なぜ基準化するのか？

電源がオンになったときのマシンの原点を正確に検知するために，制御装置は回転軸上の位置測定系と同期化される必要があります。この処理を基準化といいます。軸を基準化するのに必要なオペレーションのシーケンスを基準点アプローチといいます。

基準化された軸のみがマシン上のプログラムされた位置に正確にアプローチします。

位置測定系のインストール場所

位置測定系は以下のようにインストールできます。

- ゼロマークエンコーダとしてスピンドル上の Bero 近接スイッチと組合わせてモータ上に置きます。
- ゼロマークエンコーダとして測定ギアボックスを介してスピンドル上の Bero 近接スイッチと組合わせてモータ上に置きます。
- スピンドル上に直接置きます。
- ゼロマークエンコーダとして（1:1 ではない比率に限る）測定ギアボックスを介してスピンドル上の Bero 近接スイッチと組合わせて

2 つの位置測定系がある場合，両方を同じまたは別の場所にインストールすることができます。

同期化手順

スピンドルのスイッチがオンになると，スピンドルは以下のように同期化されます。

- スピンドルはスピンドル速度（S 値）とスピンドル回転（M3 または M4）を伴って始動され，次の位置測定系のゼロマークまたは次の Bero 信号を使って同期化されます。
- スピンドルには SPOS または SPOSA （1.17.2「■スピンドル位置決めモード」を参照）を伴って静止状態からの位置決めが必要です。スピンドルは次の位置測定系のゼロマークまたは次の Bero 信号を使って同期化されます。そしてプログラムされた位置に位置決めされます。
- スピンドルは SPOS または SPOSA を伴って動作（M3 または M4 への）から同期化することができます。

  反応は以下のとおりです。

- SPOS = Pos, SPOS = DC(Pos) および SPOS = AC(Pos) は動作方向を保持し，位置へのアプローチがされます。

- SPOS = ACN(Pos) または SPOS = ACP(Pos) により，位置は常に負または正方向の動作を伴ってアプローチされます。必要であれば，動作の方向は位置決めの前に反転します。

- 手順をパートプログラム，FC 18 または同期化動作のいずれから開始しても違いはありません。
- JOG モードで方向によってゼロマークを交差させることで，速度制御モードに合わせます。

（注）スピンドルの同期化の最中に，可能なすべての基準点値と基準点オフセットは選択された測定系に適しているとして有効になります。測定系へのオフセットは類似しています。
以下のマシンデータに注意してください。
MD 34080: REFP_MOVE_DIST （距離コード化系の基準点距離／移動先）
MD 34090: REFP_MOVE_DIST_CORR （基準点オフセット／絶対オフセット，距離コード）
MD 34100: REFP_SET_POS （距離コード化系に関連しない基準点値）

（注）SPOS=IC(...) およびパス＜360 度で基準化されていないスピンドルが位置決めされる場合，ゼロマークが交差せず，スピンドル位置がまだゼロマークで同期化されないことがあります。
これが発生するのは：
・電源オン後
・「位置決め 2 および 1 のスピンドルの再同期」インタフェース信号 (DB31,...DBX17.5, 17.4) がイネーブル

BERO を伴う同期の特別機能

BERO 信号の遅延により引き起こされる位置の変造は，信号ランタイム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。

正または負方向の動作の信号ランタイム補正値（デッドタイム）が MD 34200: ENC_REFP_MODE = 2 または 7 （SW4.1 以降）と合わせて，MD 31122: BERO_DELAY_TIME_PLUS または MD 31123: BERO_DELAY_TIME_MINUS に入力されます。

- MD 34200: ENC_REFP_MODE = 7 （SW3.6 から）を設定することで，MD 34040: REFP_VELO_SEARCH_MARKER で定義されている定められた速度でのみ位置同期を実行します。MD 34040 で設定されている速度は JOG-REF モードでの基準化および G74 を使用するパートプログラムでも有効です。
- MD 34200: ENC_REFP_MODE = 2 を設定することで，特定の速度を指定せずに位置同期を実行することができます。

（注）
信号ランタイムは内容が変更されなくてもよいように，伝達される前に設定されます。

基準化シーケンス

スピンドルが制御電源を入れた後軸モードでプログラムされる場合，軸は基準化されている必要があります。制御のスイッチがオンになっていれば，スピンドルは以下のように同期化されます（条件は回転ごとに 1 つのゼロマーク）。

- 手順については，次を参照してください。
  参照： 1.16 原点復帰（R1）
- スピンドルに使用された位置測定系が回転軸にも使用される場合，回転軸はスピンドルが同期化されると同時に基準化（同期化手順を参照）されます。

位置測定系スピンドル

スピンドルおよび軸モードに単一のモータが使用される場合，スピンドルはスピンドルモードから軸モード（回転軸）に切換えることができます。スピンドルのスピンドルおよび軸モードへの切換えについては 1.17.2「■スピンドルモード」を参照してください。

スピンドル（スピンドルモードおよび軸モード）には 1 つまたは 2 つの位置測定系を取り付けることができます。2 つの位置測定系を使うと，1 つの位置測定系をスピンドルに，もう 1 つを回転軸に割当てる，または 2 つの位置測定系をスピンドルに割当てることができます。2 つの位置測定系は，どちらも制御によって更新されるが，アクティブになるのは 1 つだけです。「位置測定系 1 および 2」インタフェース信号 (DB31, ... DBX1.5 および DBX1.6) はアクティブな位置測定系を選択するのに使用されます。

アクティブな位置測定系は以下の機能で必要です：

- スピンドルの位置制御 (SPCON)
- スピンドル位置決め (SPOS および SPOSA)
- ねじ切り (G33)
- 補正チャックを使用しないタッピング (G331, G332)
- 回転フィードレート (G95)
- 定切削速度 (G96, G97)
- スピンドル速度表示
- 軸モード
- 同期軸セットポイント値リンク

「再同期スピンドル」DB31, ... DBX16.4

以下の場合，スピンドル位置測定系は再同期化されなければなりません：

- 位置エンコーダはモータ上にあり，Bero 近接スイッチがスピンドル上に搭載され，ギアチェンジが実行されます。スピンドルが新しいギアステップ（同期化手順を参照）で回転しているとき，同期は内部から作動されます。
- マシンには垂直および水平スピンドルの選択スイッチがあります。2 つの異なった位置エンコーダが使用されるが（1 つは垂直スピンドル，もう 1 つは水平スピンドル），制御では 1 つの実際値入力が使用されます。システムが垂直から水平スピンドルへ切換わると，スピンドルは再同期化されなければなりません。同期は「再同期スピンドル 1」インタフェース信号 (DB31, ... DBX16.4) または「再同期スピンドル 2」インタフェース信号 (DB 31, ... DBX16.5) により作動します。スピンドルは制御モードでなければなりません。

■ ギアステップ変更

なぜギアステップが必要なのか？

ギアステップはスピンドル上で，低いスピンドル速度で高いトルクを発生させるために，モータ回転速度を段階的に落とすのに使われます。

ギアステップの番号

各スピンドルには 5 つずつのギアステップが設定されています。スピンドル上にインストールされたスピンドルモータに正のトランスミッション率 (1:1) または一定のトランスミッションがあれば，MD 35010: GEAR_STEP_CHANGE_ENABLE（ギアチェンジが可能）はゼロに設定される必要があります。

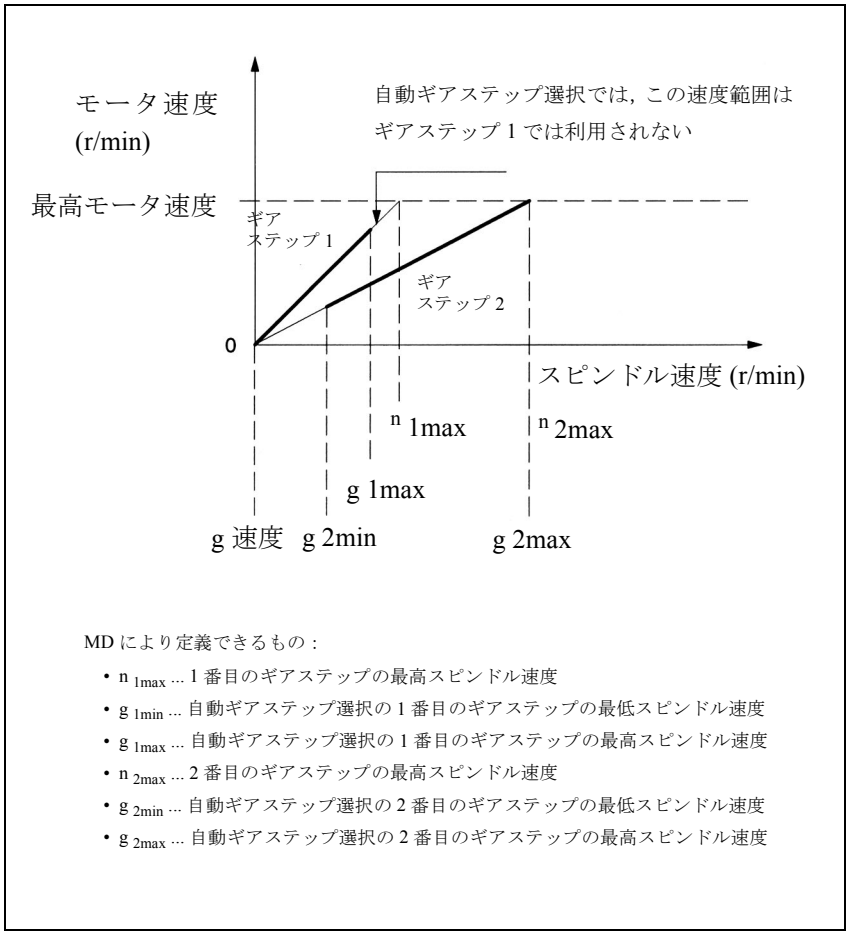


図 1.157 ギアステップ選択を使用したギアステップ変更

ギアステップの定義

ギアステップは以下のように定義されます：

- パートプログラム中の永久的定義 (M41 ～ M45)
- プログラムされたスピンドル速度 (M40) による自動的定義
- 機能モジュール FC 18 を使用する PLC による

M40 の場合，スピンドルは S 値を伴う自動ギアステップ選択のために制御モードでなければなりません。そうでない場合は，ギアステップ変更は拒否され，アラーム 22000 "Gear change not possible" （ギアチェンジ不可能）が出力されます。

M41 ～ M45

ギアステップは M41 ～ M45 を使用したパートプログラムでは永久的に定義されます。M41 ～ M45 で現在の（実際の）ギアステップと同じでないギアステップが定義されていれば，「ギアチェンジ」インタフェース信号 (DB31, ... DBX82.3) と「A ～ C のギアステップ設定」インタフェース信号 (DB31, ... DBX82.0 ～ DBX82.2) がイネーブルになります。プログラムされたスピンドル速度 （S 値）はこの永久的に定義されたギアステップを参照します。永久的に定義されたギアステップの最高速度を超過したスピンドル速度がプログラムされると，速度はギアステップの最高速度に限定され，「プログラムされた速度が速すぎる」インタフェース信号 (DB31, ... DBX83.1) がイネーブルになります。

M40

パートプログラム中の M40 では，ギアステップは制御により自動的に定義されます。制御装置はどのギアステップがプログラムされたスピンドル速度（S 値）に可能かをチェックします。現在の（実際の）ギアステップと同じでないギアステップが見つかると，「ギアチェンジ」インタフェース信号と「A ～ C のギアステップ設定」がイネーブルになります。

自動ギアステップ選択機能は最初に，プログラムされたスピンドル速度と現在のギアステップの最低および最高速度を比較します。

比較が正であれば，新しいギアステップは定義されません。比較が負であれば，それぞれのギアステップ（ギアステップ 1 から始まる）で結果が正になるまで比較が続けられます。5 つのギアステップで比較が正にならない場合，ギアステップ変更は始められません。必要であれば，速度は最高速度に限定されるかまたは，現在選択されているギアステップの最低速度まで上げられ，それに応じて「プログラムされた速度が速すぎる」または「プログラムされた速度が遅すぎる」インタフェース信号がイネーブルになります。

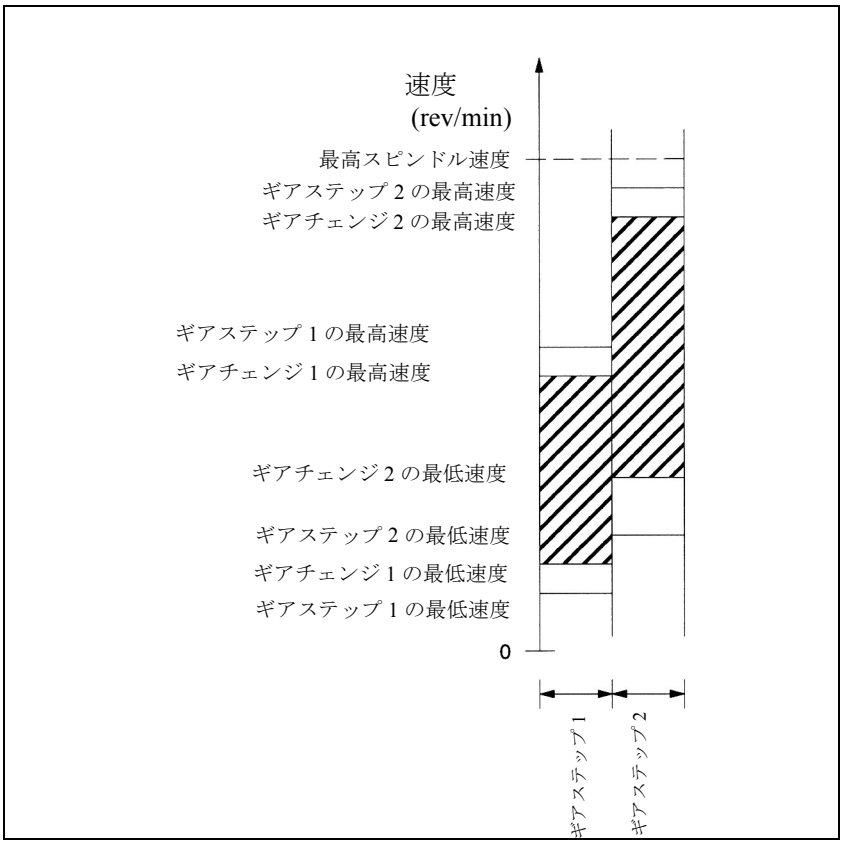


図 1.158 自動ギアステップ選択 (M40) での速度範囲例

静止スピンドルを伴うギアステップ変更

新しいギアステップが M40 と，M41 ～ M45 で選択されたスピンドル速度により定義されると，「A ～ C のギアステップ設定」(DB31, ... DBX82.0 ～ DBX82.2) および「ギアチェンジ」インタフェース信号 (DB31, ... DBX82.3) がイネーブルになります。「発振速度」インタフェース信号 (DB31, ... DBX18.5) がイネーブルになる点で，スピンドルは発振加速または速度制御／位置制御の加速度を伴って停止するまで減速します。

M40 と S 値または M41 ～ M45 までのギアチェンジに続くパートプログラムの次のブロックは実行されません（「読取りディスエーブル」インタフェース信号 (DB21, ...DBX6.1) の起動と効果は同じ）。

スピンドルが停止（「静止軸／スピンドル」インタフェース信号 (DB31, ... DBX61.4)）する最後に，発振は「発振速度」インタフェース信号 (DB31, ... DBX18.5) （1.17.2「■スピンドルモード発振モード」を参照）を使用して起動されます。

新しいギアが使用中であれば，PLC アプリケーションにより「実際のギアステップ」(DB31, ... DBX16.0 ～ DBX16.2) および「ギアチェンジ」インタフェース信号 (DB31, ... DBX16.3) がイネーブルになります。ギアチェンジが完了（スピンドル発振モードが解除される）し，システムは新しい実際のギアステップのパラメータの設定をロードします。NCK は「ギアチェンジ」インタフェース信号 (DB31, ... DBX82.3) をリセットし，そこで PLC アプリケーションが「ギアチェンジ」インタ

フェース信号をリセットします。

M3 または M4 がギアステップの前にアクティブであると，スピンドルは新しいギアステップから最後にプログラムされたスピンドル速度に加速します。パートプログラムの次のブロックが実行されます。

パラメータセット

5 つのギアステップに次の割当てでそれぞれ 1 つのパラメータセットが提供されます：

パラメータセットインデックス	VDI インタフェース	データブロックのデータ	内容
0	-	軸モードのデータ	K v 係数監視
1	000	1 番目のギアステップ	M40 速度 最低
	001	のデータ	／最高速度 加速
2	010	2 番目のギアステップのデータ	
3	011	3 番目のギアステップのデータ	
4	100	4 番目のギアステップのデータ	
5	101	5 番目のギアステップのデータ	
	110		
	111		

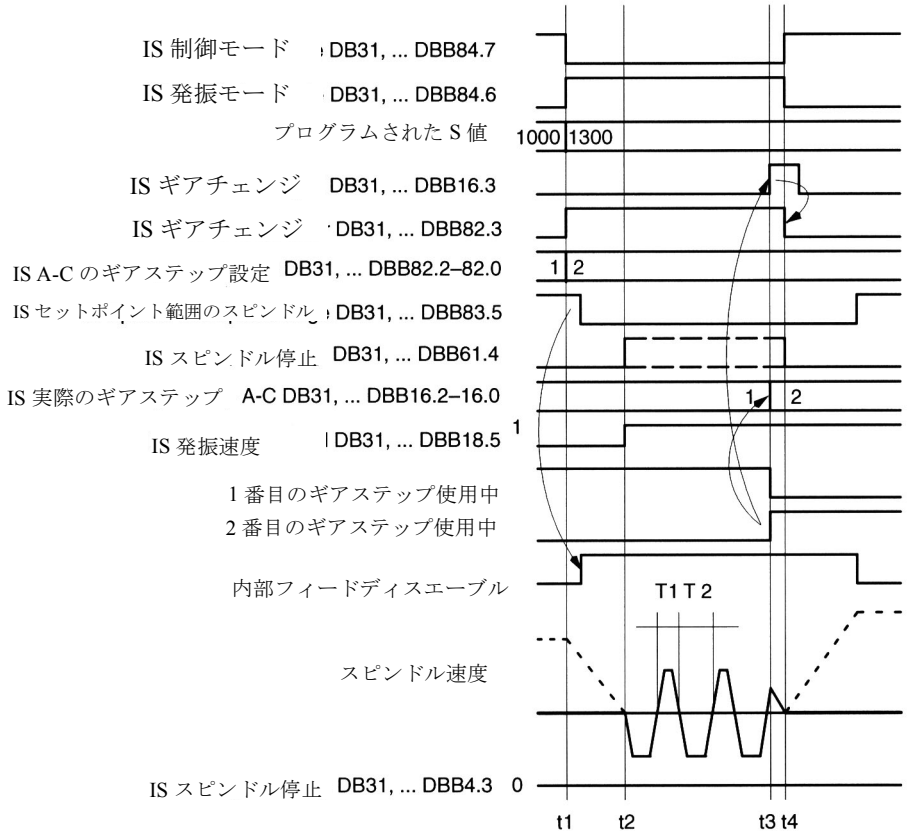
特別機能

- スピンドルにブレーキをかけるには，PLC アプリケーションが「スピンドル停止」インタフェース信号 (DB31, ... DBX8.3) をイネーブルにする必要はありません。スピンドルはギアチェンジが必要になると内部で制御により停止させられます。
- 「発振速度」インタフェース信号 (DB31, ... DBX18.5) は機械的なギアの連動をサポートするためのものです。ギアステップ変更手順で内部制御メカニズムに影響はなく，必要に応じてのみ設定されます。

SW 4.4 およびそれ以降では，ギアチェンジに関して以下のことに注意してください。

1. ギアチェンジは「ランプ機能発生器急速停止アクティブ」(DB31..48,DBB20.1) を選択することで中断させられることはありません。セットポイント 0 が出力されます。
   ギアチェンジは PLC 信号「ギア変更」(DB31..48, DBB16.3) により通常どおり確認応答されます。
2. 「ランプ機能発生器急速停止アクティブ」信号はギアチェンジを終了させるまえに PLC によりリセットされる必要があります。
3. ギアチェンジは NC リセットでアラームの出力なしに終了させられます。
4. 実際のギアステップ (DB31..48, DBB16.0..2) は，アクティブなパートプログラムがなく，スピンドルが FC 18 またはシンクロナイズドアクションにより起動されない場合，ギアチェンジ（NC → PLC 「ギアチェンジ」DB31..48, DBB82.3 が 0）の外で NC により常に受け入れられます。
   実際のギアステップが NC により受け入れられると，NC はそのギアステップを PLC としては通常どおりギアステップセットポイント (DB31..48, DBB82.0..2) に戻します。

スピンドルギアステップ変更の標準的時系列のシーケンス：



t1　S1300 がプログラムされると，NCK は新しいギアステップ（2 番目のギアステップ）を検知し，「ギアチェンジ」インタフェース信号をイネーブルにし，次のパートプログラムブロックの実行を防止します。

t2　スピンドルは静止し，発振が開始します（NCK を介しての発振）。発振速度インタフェース信号は t2 時間によりイネーブルにされなければなりません。

t3　新しいギアステップが使用中です。PLC ユーザーは新しい（実際の）ギアステップを NCK に伝送し，「ギアチェンジ」インタフェース信号をイネーブルにします。

t4　この時点で，NCK は「ギアチェンジ」インタフェース信号をキャンセルし，発振を終了させ，次のパートプログラムブロックの実行をイネーブルにし，スピンドルを新しい S 値 (S1300) に加速します。

図 1.159 静止スピンドルを使用したギアステップ変更

■ 選択可能なスピンドル

アプリケーション

SW の 4.3 および以降のバージョンでは，「選択可能なスピンドル」機能により，設定するスピンドル（「物理的スピンドル」）からチャンネルの実際の割当てにかかわらず，使用されたスピンドルを基準化してパートプログラムを書くことができます。

「軸交換」によりロードされたまたはされていない物理的スピンドルは，パートプログラムで明確に指定される必要はありません。

機能

各スピンドルは MD35000: SPIND_ASSIGN_TO_MACHAX[AX...] で設定可能な番号によりマシン軸に割当てられます。この番号はスピンドルがアクティブに扱われているチャンネルに係わらず常にスピンドルにあてられます。

チャンネルスピンドルは，パートプログラムで使われている論理的スピンドル番号とチャンネルに存在する物理的スピンドルの中間レベルが導入されるため切換えられます。これはパートプログラムで使用される各論理的スピンドルを設定データ （SD 42800: SPIND_ASSIGN_TAB[...]; スピンドル番号コンバータ）を含むテーブルにある物理的スピンドルに割当てることで行われます。
スピンドル番号コンバータは以下によりスピンドルのプログラミングに有効です：

- パートプログラム
- シンクロナイズドアクション

スピンドル番号コンバータは機能ブロック FC18 を使用する PLC コマンドでは有効ではありません。物理的スピンドルは常に軸の概念の範囲内で割当てられなければなりません。

論理的スピンドルは SD 42800: SPIND_ASSIGN_TAB[...] を変更することで切換えられます。切換えはパートプログラム，PLC および／または MMC から始められます。

（注）SD 42800: SPIND_ASSIGN_TAB[0] は論理的マスタスピンドルを含んでいます。これは表示の目的でのみ使用されます。
この設定データは "SETMS （論理的スピンドル）" によりパートプログラムで定義されています。
値 φ は使用されない主軸に対して SD42800 で入力されなくてはなりません。
スピンドル切換えに影響されるシステム変数は：
$P_S, $P_SDIR, $P_SMODE, $P_GWPS, $AC_SDIR, §
AC_SMODE, $AC_MSNUM, $AA_S. 以下も参照のこと：

参照： YS840DI ユーザーズマニュアル・プログラミング編

コンバートされた物理的スピンドル番号は，補助機能出力で拡張アドレスとして常に出力されます。

補足条件

- 切換え可能なチャンネルスピンドルは軸交換機能の代用にはなりません。
- パラメータ表示によりチャンネルに割当てられたスピンドルのみ切換えられます。
- 別のチャンネルで現在アクティブなスピンドルに切換えの表示が出ると，設定の相違によって，物理的スピンドルに "Auto-Get" 機能が起動されるかまたはアラーム 16105 "Assigned spindles do not exist" （割当てられたスピンドルが存在しません）が出力されます。
- SD 42800: SPIND_ASSIGN_TAB[...] が PLC または MMC から定義されると，テーブルが訂正されたチャンネルにはリセットステータスがなければならず，切換えられるスピンドルは現在起動しているパートプログラムで使用されていないものでなければなりません。同期反応は STOPRE プロセッサ停止によって行われます。
- 論理的および物理的スピンドルの間の複数のマッピングは NC においてインタロックされません。しかし，論理的スピンドルが MMC 上に表示されたとき，

変換テーブルに従って不明瞭さが現れます。

起動

SD 42800: SPIND_ASSIGN_TAB[...] は MD 20092: SPIND_ASSIGN_TAB_ENABLE=1 をイネーブルにすることにより起動されます。

SD 42800 の初期設定

NC がスタートアップスイッチ位置 1 （SRAM 削除）においてスイッチオンされると，SD 42800 が初期設定になります。論理的および物理的スピンドルの番号が 同一のものになります。

SD 42800: SPIND_ASSIGN_TAB[1] = 1

SD 42800: SPIND_ASSIGN_TAB[2] = 2

SD 42800: SPIND_ASSIGN_TAB[3] = 3

SD 42800: SPIND_ASSIGN_TAB[4] = 4

SD 42800: SPIND_ASSIGN_TAB[5] = 5

...

例

仮定：スピンドル設定：

スピンドル番号とマシン軸を定義する

MD 35000: SPIND_ASSIGN_TO_MACHAX [AX4] = 1

MD 35000: SPIND_ASSIGN_TO_MACHAX [AX5] = 2

MD 35000: SPIND_ASSIGN_TO_MACHAX [AX6] = 3

MD 35000: SPIND_ASSIGN_TO_MACHAX [AX7] = 5

マシン軸をチャンネルに含める

MD 20070: AXCONF_MACHAX_USED[0] = 4

MD 20070: AXCONF_MACHAX_USED[1] = 5

MD 20070: AXCONF_MACHAX_USED[2] = 6

MD 20070: AXCONF_MACHAX_USED[3] = 7

マスタスピンドルを定義する

MD 20090: SPIND_DEF_MASTER_SPIND = 1

スピンドル番号コンバータ

MD 20092: SPIND_ASSIGN_TAB_ENABLE = 1 スピンドル番号コンバータを起動する

SD 42800: SPIND_ASSIGN_TAB[0] = 1 設定どおりのマスタスピンドル

SD 42800: SPIND_ASSIGN_TAB[1] = 1 テーブルの初期設定

SD 42800: SPIND_ASSIGN_TAB[2] = 2

SD 42800: SPIND_ASSIGN_TAB[3] = 3

SD 42800: SPIND_ASSIGN_TAB[4] = 0 論理的スピンドルが割当てられていない

M3 S1000　　拡張アドレス = 1, M 1=3 S 1=1000 が出力される

番号が "1" （物理的マスタスピンドルの番号）に設定されたスピンドルが回転。

...

...

SD 42800: SPIND_ASSIGN_TAB[1] = 5 論理的スピンドル 1 から物理的スピンドル 5 まで割当てる

SD 42800: SPIND_ASSIGN_TAB[2] = 3 論理的スピンドル 2 から物理的スピンドル 3 まで割当てる

注意 : 物理的スピンドル 3 はここで 2 度割当てられた。

論理的スピンドル 2 および 3 をプログラムするとき，物理的スピンドル 3 は常に当てられる。基本的マシン表示では，両方のスピンドルが回転する。

SETMS (2)　SD 42800:SPIND_ASSIGN_TAB[0] = 2 が NCK により内部的に定義される。

...

M5　マスタスピンドル = 拡張アドレス = 2,
変換されたスピンドル番号が M3=5 と出力され，番号 "3" で設定された物理的スピンドルが停止する。

...

GET (S4)　論理的スピンドル "4" が表示されないため，アラーム 16105 が表示される。

..

RELEASE (S1)　チャンネルスピンドル "1" = 物理的。スピンドル "5" がイネーブルになる

..

M30

■ プログラミング

（注）スピンドルのプログラミングについては以下を参照してください。

参照： YS840DI ユーザーズマニュアル・プログラミング編

SETMS

MD 20090: SPIND_DEF_MASTER_SPIND （チャンネルのマスタスピンドルの初期設定）に保存されているスピンドルがマスタスピンドルです。

SETMS(n)

番号 (n) を伴うスピンドルはマスタスピンドル（MD 20090: SPIND_DEF_MASTER_SPIND の初期設定とは異なる場合がある）です。

マスタスピンドルは以下の機能のために定義されなければなりません。

- G95　回転フィードレート
- G96 S...　m/min 表示での定切削速度
- G97　G96 をキャンセルして最後のスピンドル速度をフリーズさせる
- G63　補正チャックを使用したタッピング
- G33　ねじ切り
- G331/G332　リジッドタッピング
- G4 S...　スピンドル回転での休止時間
- スピンドル番号の特定のない M3, M4, M5, S, SPOS, SPOSA, M40, M41 ～ M45 および WAITS のプログラミング。

M3

マスタスピンドルの回転方向右

M1=3

スピンドル番号 1 の回転方向右

M4

マスタスピンドルの回転方向左

M2=4

スピンドル番号 2 の回転方向左

M5

マスタスピンドルの方向づけのないスピンドル停止

M1=5

スピンドル番号 1 の方向づけのないスピンドル停止

S....

マスタスピンドルの rpm 表示でのスピンドル速度

S2=....

スピンドル番号 2 の rpm 表示でのスピンドル速度

SPOS =270

270 度位置でのマスタスピンドルのスピンドル位置決め

SPOS[n]=270

270 度位置でのスピンドル番号 [n] のスピンドル位置決め。ブロックチェンジはスピンドルが位置決めされているときのみ実行されます。

SPOSA=90

マスタスピンドルのスピンドル位置決め。ブロックチェンジは即座に実行されます。更なるパートプログラム処理に関係なく，スピンドルが位置に到達するまでスピンドル位置決めは続けられます。

SPOSA[n]=90

90 度位置でのスピンドル番号 [n] のスピンドル位置決め。ブロックチェンジは即座に実行されます。更なるパートプログラム処理に関係なく，スピンドルが位置に到達するまでスピンドル位置決めは続けられます。

SPOS=DC(Pos)

SPOS[n]=DC(Pos)

SPOSA=DC(Pos)

SPOSA[n]=DC(Pos)

動作方向は動作中の位置決めのため保持され，位置にアプローチします。静止からの位置決めの場合，位置には最短パスでアプローチします。

SPOS=ACN(Pos)

SPOS[n]=ACN(Pos)

SPOSA=ACN(Pos)

SPOSA[n]=ACN(Pos)

位置は常に動作の負の方向でアプローチします。必要であれば，動作方向は位置決めの前に反転します。

SPOS=ACP(Pos)

SPOS[n]=ACP(Pos)

SPOSA=ACP(Pos)

SPOSA[n]=ACP(Pos)

位置は常に動作の正の方向でアプローチします。必要であれば，動作方向は位置決めの前に反転します。

SPOS=IC(Pos)

SPOS[n]=IC(Pos)

SPOSA=IC(Pos)

SPOSA[n]=IC(Pos)

移動パスが特定されます。移動方向は移動パスの正面のサインから得られます。スピンドルが動作中であれば，移動方向はプログラムされた方向に移動させるために必要に応じて反転します。

移動中にゼロマークが交差した場合，基準がないか，または新しい基準がインタフェース信号で必要ならば，スピンドルはゼロマークを伴って自動的に同期化されます。

M40

マスタスピンドルの自動ギアステップ選択

M1=40

スピンドル番号 1 の自動ギアステップ選択

M41 ～ M45

マスタスピンドルにギアステップ 1 ～ 5 を選択

M1=41 ～ M1=45

スピンドル番号 1 にギアステップ 1 ～ 5 を選択

M70

スピンドルを停止に持ちこみ，位置制御を起動させ，ゼロパラメータセットを選択し，軸モードを起動

M1=70

スピンドルを停止に持ちこみ，位置制御を起動させ，ゼロパラメータセットを選択し，スピンドル番号 1 に軸モードを起動させる

（注）M 機能の M03, M04, M05 および M70 は DB21, ... DBB194 および DBB202 では出力されません。これらの M 機能は DB21, ... DBB68 ff. および DB, DB31, ... DBB86 ff. にある特殊軸で拡張された M 機能として提供されます。

SPCON

マスタスピンドルではスピンドル位置制御オン

SPCON(n)

スピンドル番号 (n) ではスピンドル位置制御オン

SPCOF

マスタスピンドルではスピンドル位置制御オフ，速度制御モードを起動

SPCOF(n)

マスタスピンドルではスピンドル位置制御オフ，番号 (n) を伴うスピンドルでは速度制御モードを起動

C30 G90 G1 F3600

回転軸 C （軸モードにあるスピンドル）は 3600 deg/min =10 rpm の速度で 30 度位置に移動

G25 S....

プログラム可能な最低スピンドル速度制限

G25 S2....

スピンドル番号 2 のプログラム可能な最低スピンドル速度制限

G26 S....

プログラム可能な最高スピンドル速度制限

G26 Sn....

スピンドル番号 n のプログラム可能な最高スピンドル速度制限

LIMS=

マスタスピンドルの G96 のプログラム可能な最低スピンドル速度制限

WAITS

パートプログラム , マスタスピンドルの同期コマンド

- 続くブロックの実行は SPOSA でプログラムされたスピンドルが正確停止（微）の位置に到達するまでは中止されます。
- スピンドルが M5 の後の静止になるまで待機します。

WAITS(n)

スピンドル n の同期コマンド

WAITS(n,m)

スピンドル n および m の同期コマンド

FA[Sn]

[deg/min] で示されるスピンドルの位置決め速度（軸フィード）。MD : POSCTRL_VELO で設定された FA[Sn]=0 での値が再度有効になります。

OVRA[Sn]

[%] で示されるスピンドル n の軸オーバライド値のプログラミング

ACC[Sn]

[%] で示されるスピンドル n の軸加速容量

■ 外部からのプログラミング (PLC, MMC)

回転送り速度での移動はチャンネル別設定データ MD 42600: JOG_FEED_PER_REV_SOURCE （JOG モードでの回転送り速度制御）から JOG モードの軸設定データ MD 43300: ASSIGN_FEED_PER_REV_SOURCE （スピンドルの回転フィードレート）を介して選択できます。設定データから以下の設定ができます。

- >0: 回転フィードレートが得られる回転軸／スピンドルのマシン軸番号。
- -1: 回転フィードレートは軸／スピンドルがアクティブな特定のチャンネルのマスタスピンドルから得られます。
- 0: 機能は選択解除されます。

FPRAON (S2)

マスタスピンドルから得られる，スピンドル S2 からの回転フィードレート

FPRAON (S2, A)

軸 A から得られる，スピンドル S2 からの回転フィードレート。回転フィードレート値は FA[Sn] で指定される必要がある

FPRAOF (S2)

スピンドル S2 の回転フィードレートオフ

■ スピンドル監視

速度範囲

スピンドル監視および現在アクティブな機能（G94, G95, G96, G97, G33, G331, G332, など）により，スピンドルの許容速度範囲が決定します。

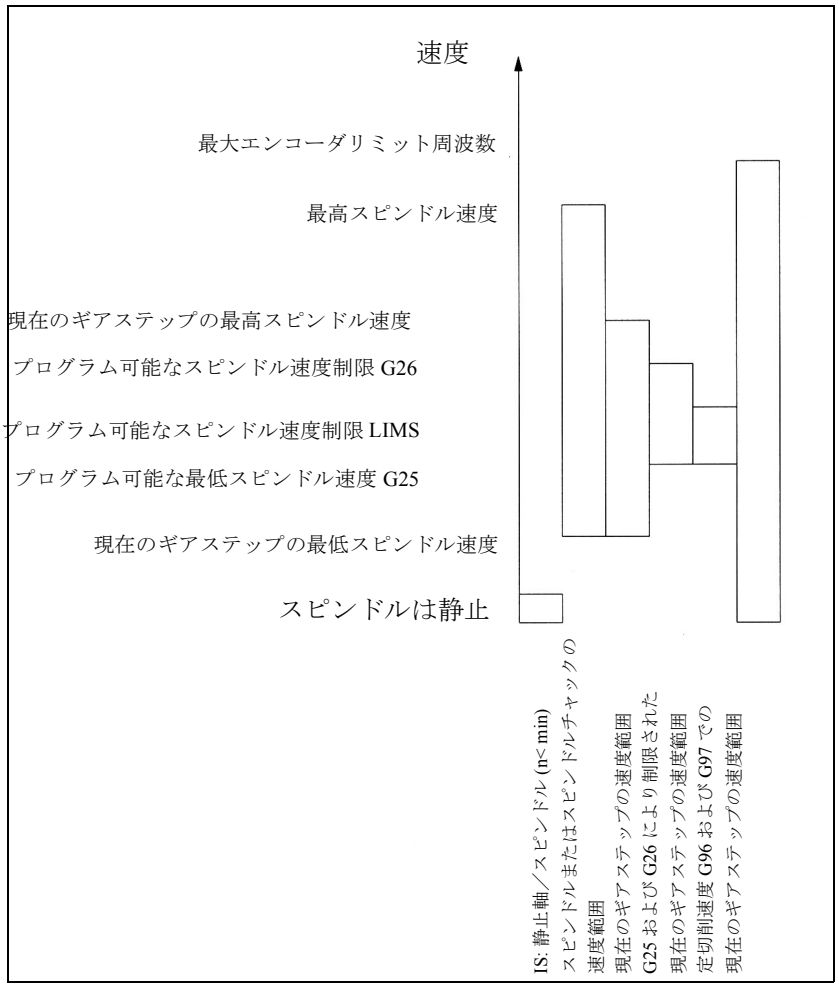


図 1.160 スピンドル監視／速度の範囲

■ 静止軸／スピンドル (n < n min)

実際のスピンドル速度が MD 36060: STANDSTILL_VELO_TOL （最高速度「静止軸／スピンドル」）で定義された値を下回るなど，軸／スピンドルが静止であるときのみ，ツール交換，マシンドアを開ける，パスフィードイネーブルなど，マシンの特定の機能を実行することができます。

- IS 「静止軸／スピンドル」インタフェース信号 (DB31, ... DBX61.4) はイネーブルになります。
- MD 35510: SPIND_STOPPED_AT_IPO_START がイネーブルになると，次の作業ブロックがイネーブル（位置決め軸と G0 を伴う位置決めブロックに影響せず，移動パス軸は停止しない）になります。

監視はすべてのスピンドルモードおよび軸モードで有効になります。

■ セットポイント範囲でのスピンドル

機能

「セットポイント範囲でのスピンドル」監視はプログラムされたスピンドル速度に到達したか，スピンドルが静止にあるか（「静止軸／スピンドル」），またはまだ加速またはブレーキ相にあるかをチェックします。

スピンドルモード制御モードでは，速度セットポイント（プログラムされた速度 _ アクティブな制限を含むスピンドルオフセット）は実際の速度と比較されます。速度セットポイントからの実際の速度のずれがスピンドル速度許容値 (MD 35150: SPIND_DES_VELO_TOL) より大きい場合：

- 「セットポイント範囲でのスピンドル」インタフェース信号 (DB31, ... DBX83.5) がゼロに設定されます。
- MD 35500: SPIND_ON_SPEED_AT_IPO_START がイネーブルになると，次の作業ブロックはイネーブル（位置決め軸と G0 を伴う位置決めブロックに影響せず，移動パス軸は停止しない）になりません。

■ 最高スピンドル速度

最高スピンドル速度

スピンドル（ワークを伴うスピンドルチャックまたはツールを伴うスピンドル）により超過することがないと予想される最高速度は，速度監視のため定義されます。最高スピンドル速度は MD 43230: SPIND_MAX_VELO_LIMIT に入力されています。NCK はこの値への過度なスピンドル速度セットポイントを制限します。実際のスピンドル速度がスピンドル速度許容値 MD 35150: SPIND_DES_VELO_TOL での許容にも係わらず最高制限を超過し続ける場合，ドライブに故障が発生し，「速度制限オーバ」インタフェース信号 (DB31, ... DBX83.0) がイネーブルになります。アラーム 22100 "chuck speed exceeded" （チャック速度オーバ）がまた出力され，チャンネルの全ての軸およびスピンドルが停止します。

PLC からの速度制限

スピンドル速度は PLC により指定された値に制限されます。この値は MD 35160: SPIND_EXTERN_VELO_UNIT に保存され，「速度制限」インタフェース信号 (DB31, ... DBX3.6) により起動します。

■ ギアステップの最低／最高速度

最高速度

ギアステップの最高速度は MD 35130: GEAR_STEP_MAX_VELO_LIMIT に入力されています。この速度（セットポイント）は現在使用中のギアステップにおいて超過することはありません。プログラムされたスピンドル速度が制限されていると，「セットポイント速度が制限されている」インタフェース信号（プログラムされた速度から高く) がイネーブルになります。

最低速度

ギアステップの最低速度は MD 35140: GEAR_STEP_MIN_VELO_LIMIT に入力されています。過度に小さい S 値をプログラムすると，速度をこの制限（セットポイント）より低くできません。そこで,「セットポイント速度増加」インタフェース信号（プログラムされた速度から低く）がイネーブルになります。

最低ギアステップ速度は速度モードでのみオペレーション可能で，ギアステップの速度は以下をとおして最低制限を下回ることしかできません：

- スピンドルオーバライド 0%
- M5
- S0
- 「スピンドル停止」インタフェース信号 (DB31, ... DBX8.3)
- 「コントローライネーブル」インタフェース信号 (DB31, ... DBX2.1) キャンセル
- 「リセット」(DB21, ... DBX7.7)
- 「移動距離削除／スピンドルリセット」インタフェース信号 (DB31, ... DBX2.2)
- 「発振速度」インタフェース信号 (DB31, ... DBX18.5)
- 「NC- 停止軸＋スピンドル」インタフェース信号 (DB21, ... DBX7.4)
- 「軸／スピンドルディスエーブル」インタフェース信号 (DB31, ... DBX1.3)
- 「S 値削除」インタフェース信号 (DB31, ... DBX16.7)

■ 最大エンコーダリミット周波数

警告

実際のスピンドル位置エンコーダの最大エンコーダ周波数制限は制御（制限は超過されることがあります）により監視されます。スピンドルモータ，ギア，測定ギア，エンコーダおよびマシンデータの設定が，実際のスピンドル位置エンコーダから最高速度が超過するのを防止する，ということはマシンツール製造業者の責任において安全確保をされるものです。

最高エンコーダ周波数オーバ

スピンドル速度が最大エンコーダリミット周波数（エンコーダの最高機械的速度制限は超過してはならない）を超過する速度（プログラムされた大きな S 値）に到達すると，同期は失われます。
スピンドルは回転を続けるが，機能性は減少します。

以下の機能では，スピンドル速度はアクティブな測定系が再びエンコーダリミット周波数を下回ってオペレーションするまで減少します：

- ねじ切り (G33)
- リジッドタッピング (G331, G332)
- 回転フィードレート (G95)
- 定切削速度 (G96, G97)
- SPCON （位置制御されたスピンドルオペレーション）

最大エンコーダ周波数（プログラムされた小さい S 値，スピンドルオーバライドスイッチなど）を下回る速度に到達すると，スピンドルは自動的に次のゼロマークまたは次の Bero 信号で同期化されます。新しい同期は常に，同期を失い最大エンコーダリミット周波数が現在超過していない選択された位置測定系で機能します。

エンコーダリミット周波数を超過すると，「基準化／同期化 1」または，「基準化／同期化 2」インタフェース信号が関連する測定系でリセットされ，「エンコーダリミット周波数 1 オーバ」または「エンコーダリミット周波数 2 オーバ」インタフェース信号がイネーブルになります。

スピンドルが軸モードにあると，最大エンコーダリミット周波数は超過してはなりません。最高速度 (MD 32000: MAX_AX_VELO) は最大エンコーダリミット周波数を下回っていなければならず，そうでなければアラーム 21610 が出力され，軸は静止に持ちこまれます。

最大エンコーダ周波数を下回る速度

最大エンコーダ周波数制限を超過し，続いて速度が最大エンコーダリミット周波数（プログラムされた小さい S 値，スピンドルオフセットスイッチ変更など）より落ちるとき，スピンドルは自動的に次のゼロマークまたは次の Bero 信号で同期化されます。新しい同期は常にアクティブな位置測定系で機能します。

特別機能

以下の機能がアクティブであれば，最大エンコーダ周波数は超過しません。

- スピンドル位置決めモード
- ねじ切り (G33)
- リジッドタッピング G331, G332 （G63 には適用されない）
- 回転フィードレート (G95)
- 定切削速度 (G96, G97)
- SPCON （位置制御されたスピンドルオペレーション）

■ 移動先監視

移動先監視

位置決め（スピンドルは位置決めモードにある）の間，システムはスピンドル（実際の位置を基準に）からプログラムされたスピンドル設定位置（移動先）までの距離を監視します。

2 つの制限値がスピンドル設定位置から始まる相対パスとして MD 36000: STOP_LIMIT_COARSE （正確停止制限（粗））および MD 36010: STOP_LIMIT_FINE に定義されています。スピンドルの位置決めは，連結したスピンドル測定エンコーダ，バックラッシュ，伝送比などのように常に正確です。

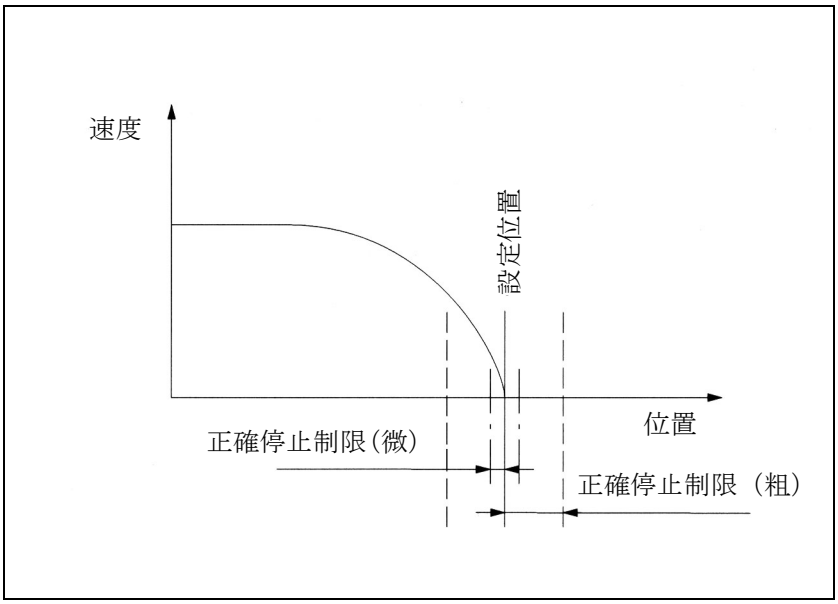


図 1.161 スピンドルの正確停止ゾーン

「粗／微の正確停止を使用して到達した位置」DB31, ... DBX60.7 DB31, ...DBX60.6

MD 36000: STOP_LIMIT_COARSE および MD 36010: STOP_LIMIT_FINE （正確停止（粗／微)）で規定されている 2 つの制限が「(DB31, ... DBX60.7) および「正確停止（微）を使用して到達した位置」インタフェース信号 (DB31, ... DBX60.6) を伴って PLC に出力されます。

SPOS を使用したブロックチェンジ

スピンドルが SPOS を使用して位置決めされると，ブロックチェンジは「正確停止（微）を使用して到達した位置」インタフェース信号を伴う移動先監視に依存します。ブロック中にプログラムされた他のすべての機能はエンド基準（PLC により確認応答されるすべての補助機能など）を実行していなければなりません。

SPOSA を使用すると，ブロックチェンジは移動先の監視に依存しません。

1.17.3 補足条件

この機能の詳細に規定された補足条件はありません。

1.17.4 データの説明 (MD，SD)

■ チャンネル別マシンデータ

20090 MD 番号	SPIND_DEF_MASTER_SPIND チャンネル上のマスタスピンドルの初期設定	
初期設定 : 1	最小入力リミット : 1	最大入力リミット :5
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 1.1	
意味 :	各チャンネルは以下の機能のためのマスタスピンドルを持たなければならない。 • G95　回転フィードレート • G63　補正チャックを使用したタッピング • G33　スレッド切削 • G331/G332 リジッドタッピング • G4 S...　スピンドル回転での休止時間 マスタスピンドルはスピンドル番号を指定せずにコマンド M3, M4, M5, S, SPOS, WAITS, SPOSA, M40, M41 ～ M45 を使用してプログラムすることが可能である。 MD 20090: SPIND_DEF_MASTER_SPIND では，マスタスピンドルのリセット値は新しいマスタスピンドルが SETMS プログラムコマンドで定義されるまで適用される。STEMS 設定は NC 開始で削除される。続く M02/M30 および新しい NC 開始で，SPIND_DEF_MASTER_SPIND で定義されているスピンドルは常にマスタスピンドルである。	
アプリケーション	旋盤には 2 つのスピンドル（主スピンドルおよび回転ツールとしての補助スピンドル）がある。主スピンドルは MD: SPIND_DEF_MASTER_SPIND ではマスタスピンドルとして示されている。SETMS プログラムコマンドは穴ぐりおよび面エンド加工（フライス）で補助スピンドルをマスタスピンドルとして示すために使われる。それゆえ M3, M4 および M5 （アドレスエクステンションを伴わない）を使用して補助スピンドルをプログラム，またはリジッドタッピングを実行することが可能である。	

20092 MD 番号	SPIND_ASSIGN_TAB_ENABLE スピンドルコンバータのイネーブル／ディスイネーブル	
初期設定 : 0,0,0,0, 0,0,0,0, 0,0,0,0, 0,0,0,0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 7/7	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 4.3	
意味 :	値 0: スピンドルコンバータ機能は停止している。 SD 42800: SPIND_ASSIGN_TAB[..] の内容は評価されない。 値 1: スピンドルコンバータは起動している。論理的スピンドルから物理的スピンドルへの変換が起こる。詳細は SD 42800: SPIND_ASSIGN_TAB を参照。 注記 :「SRAM 削除」位置 "1" への開始スイッチ）後，スピンドルコンバータは停止させられる。	
例外，エラー		
関連性	SD 42800: SPIND_ASSSIGN_TAB	

22400 MD 番号	S_VALUES_ACTIV_AFTER_RESET RESET がアクティブによる S 機能	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2 / 7	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : SW1	
意味 :	1: 主処理の最後の S 値設定はリセット後も有効。 0: リセット後，様々な S 値は 0 に等しくなり，再プログラムされなければならない。	

■ 軸／スピンドル別マシンデータ

31122 MD 番号	BERO_DELAY_TIME_PLUS 正方向での BERO 遅延時間	
初期設定 : 0.000110	最小入力リミット : 0.0	最大入力リミット : 1.0
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 4.1	
意味 :	MD 34200: ENC_REFP_MODE （基準化モード）= 7 の設定に関連して，マシンデータは BERO（ゼロマーク）を伴う位置を決定するとき，正方向の動作において信号ランタイム補正を起こす。 BERO 検知範囲の標準的全遅延時間は正方向に交差するために入力される。 含まれる時間：BERO エッジ遅延時間 信号デジタル化時間 測定値処理時間，など。 時間は使用したハードウェアに左右される。デフォルト値は標準的なものである。例外的な環境においてのみ，カスタマイズされた測定が必要になる。 最低値 "0.0" を入力すると，補正のスイッチがオフになり，これは MD 34200: ENC_REFP_MODE = 7 と関連して了解できるものである。 マシンデータは各エンコーダごとに提供される。	
関連性	MD 34200: ENC_REFP_MODE （基準化モード） MD 34040: REFP_VELO_SEARCH_MARKER[n] （基準点クリープ速度 [エンコーダ番号]）	

31123 MD 番号	BERO_DELAY_TIME_MINUS 負方向での BERO 遅延時間	
初期設定 : 0.000078	最小入力リミット : 0.0	最大入力リミット : 1.0
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 4.1	

31123 MD 番号	BERO_DELAY_TIME_MINUS **負方向での BERO 遅延時間**
意味 :	MD 34200: ENC_REFP_MODE （基準化モード）= 7 の設定に関連して，マシンデータは BERO（ゼロマーク）を伴う位置を決定するとき，負方向の動作において信号ランタイム補正を起こす。 BERO 検知範囲の標準的全遅延時間は負方向に交差するために入力される。 含まれる時間：BERO エッジ遅延時間 信号デジタル化時間 測定値処理時間，など。 時間は使用したハードウェアに左右される。デフォルト値は SIEMENS 製品の標準的なものである。例外的な環境においてのみ，カスタマイズされた測定が必要になる。 最低値 "0.0" を入力すると，補正のスイッチがオフになり，これは MD 34200: ENC_REFP_MODE = 7 と関連して了解できるものである。 マシンデータは各エンコーダごとに提供される。
関連性	MD 34200: ENC_REFP_MODE （基準化モード） MD 34040: REFP_VELO_SEARCH_MARKER[n] （クリープ速度 [エンコーダ番号]）

35000 MD 番号	SPIND_ASSIGN_TO_MACHAX **マシン軸へのスピンドルの割当て**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 5
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 1.1	
意味 :	NCK/PLC ユーザーインタフェースがすべての軸とスピンドルについて同じになるために，スピンドルは常に制御で内部的にマシン軸上に位置付けられる必要がある。SPIND_ASSIGN_TO_MA-CHAX はスピンドルとして使われていたマシン軸を入力するために使われる。	
アプリケーション	3 つのマシン軸 (X, Y および Z) と 1 つのスピンドルを持つフライス盤の例： SPIND_ASSIGN_TO_MACHAX [1] = 0 → X SPIND_ASSIGN_TO_MACHAX [AX2] = 0 → Y SPIND_ASSIGN_TO_MACHAX [AX3] = 0 → Z SPIND_ASSIGN_TO_MACHAX [AX4] = 1 → スピンドル番号 1 は 4 番目のマシン軸	
関連性	MD 30300: IS_ROT_AX （回転軸／スピンドル） MD 30310: ROT_IS_MODULO （回転軸／スピンドルのモジュロ変換） このマシンデータは必ず設定される必要がある。設定しない場合は，アラーム 4210 "rotary axis declaration missing"（回転軸宣言不明）および 4215 "modulo axis declaration missing" （モジュロ軸宣言不明）が表示される。	

35010 MD 番号	GEAR_STEP_CHANGE_ENABLE **ギアチェンジは可能**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン :	

35010 MD 番号	GEAR_STEP_CHANGE_ENABLE **ギアチェンジは可能**
意味 :	スピンドルモータがスピンドルに直接 (1:1) または不変伝送比を伴って接続されると，GEAR_STEP_CHANGE_ENABLE （ギアステップ変更可能）はゼロに設定されなければならない。M40 ～ M45 を使用してギアステップを変更することはできない。スピンドルモータがスピンドルに可変ギアステップを伴うギアを介して接続されると，GEAR_STEP_CHANGE_ENABLE は 1 に設定されなければならない。ギアは 5 つまでのギアステップを持つことができ，M40, M41 ～ M45 を使って選択できる。
関連性	MD 35110: GEAR_STEP_MAX_VELO （ギアステップ変更の最高速度） MD 35120: GEAR_STEP_MIN_VELO （ギアステップ変更の最低速度） GEAR_STEP_MAX_VELO および GEAR_STEP_MIN_VELO は全体の速度範囲を含まなければならない。

35020 MD 番号	SPIND_DEFAULT_MODE **基本スピンドル設定**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 3
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 2	
意味 :	SPIND_DEFAULT_MODE は MD 35030: SPIND_DEFAULT_ACT_MASK に定義される時間にスピンドルの運転モードを起動する。適切なスピンドル運転モードは以下の値を使って選択できる： 0　速度モード，位置制御選択解除 1　速度モード，位置制御起動 2　位置決めモード 3　軸モード	
関連性	MD 35030: SPIND_DEFAULT_ACT_MASK （基本スピンドル位置設定を起動）	

35030 MD 番号	SPIND_DEFAULT_ACT_MASK **初期スピンドル設定を起動**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 2
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 2	
意味 :	SPIND_DEFAULT_ACT_MASK は MD 35020: SPIND_DEFAULT_MODE で定義されている運転モードが有効になる時間を指定する。基本スピンドル設定は以下の値で以下の時点で割当てられる： 0　電源オン 1　電源オンおよび NC プログラム開始 2　電源オンおよびリセット (M2/M30)	
例外，エラー	MD 35040: SPIND_ACTIVE_AFTER_RESET = 1 であれば，以下の補足条件が適用される。 - SPIND_DEFAULT_ACT_MASK は 0 に設定されなければならない。 - これが不可能であれば，スピンドルは起動の前に静止しなければならない。	
関連性	MD 35020: SPIND_DEFAULT_MODE （基本スピンドル設定） MD 35040: SPIND_ACTIVE_AFTER_RESET （リセットでアクティブになるスピンドル）	

35040 MD 番号	SPIND_ACTIVE_AFTER_RESET リセット後アクティブになるスピンドル	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1

35040 MD 番号	SPIND_ACTIVE_AFTER_RESET リセット後アクティブになるスピンドル	
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	SPIND_ACTIVE_AFTER_RESET はリセット (DB21, ... DBX7.7) およびプログラム終了 (M2, M30) 後のスピンドルの反応を定義する。これはスピンドル制御モードでのみアクティブになる。 SPIND_ACTIVE_AFTER_RESET = 0: 制御モード : - スピンドルは停止し，M2/M30 および MCP リセットにのみ適用される - プログラムは強制終了され，M2/M30 に適用される。 発振モード : - アラーム 10640 "spindle cannot stop during gear change" （スピンドルはギアチェンジ中に停止できない） - 発振は強制終了されない - 軸は停止する - プログラムはギアチェンジまたはスピンドルリセット後に強制終了され，アラームは消去される。 位置決めモード : - 停止する 軸モード : - 停止する --- SPIND_ACTIVE_AFTER_RESET= 1: 制御モード : - スピンドルは停止しない - プログラムは強制終了させられる 発振モード : - アラーム 10640 "spindle cannot stop during gear change" （スピンドルはギアチェンジ中に停止できない） - 発振は強制終了されない - 軸は停止させられる - プログラムはギアチェンジまたはスピンドルリセット後に強制終了され，アラームは消去され，スピンドルはプログラムされた M および S 値を伴って回転を続ける。 位置決めモード : - 停止する。 軸モード : - 停止する。 「スピンドルリセット」インタフェース信号 (DB31, ... DBX2.2) は，SPIND_ACTIVE_AFTER_RESET に関係なく常に有効である。	
この MD は次の場合は無意味	制御モードを除くすべてのスピンドルモード	
関連性	「リセット」インタフェース信号 (DB21, ... DBX7.7) 「スピンドルリセット」インタフェース信号 (DB31, ... DBX2.2)	

35100 MD 番号	SPIND_VELO_LIMIT **最高スピンドル速度**	
初期設定 : 10 000	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	SPIND_VELO_LIMIT はスピンドル（ワークまたはツールを伴うスピンドルチャック）が超過してはならない最高スピンドル速度を定義する。最高スピンドル実際速度が超過すると，スピンドル速度許容値 (MD 35150: SPIND_DES_VELO_TOL) があっても，ドライブに故障が起こり，「速度制限オーバ」インタフェース信号 (DB31, ... DBX83.0) がイネーブルになる。アラーム 22050 "maximum speed reached" （最高速度に到達）も出力され，チャンネル上のすべての軸およびスピンドルが停止（ただしエンコーダが正しく機能していること）に持ちこまれる。	

35100 MD 番号	SPIND_VELO_LIMIT 最高スピンドル速度
関連性	MD 35150: SPIND_DES_VELO_TOL （スピンドル速度許容値） 「速度制限オーバ」インタフェース信号 (DB31, ... DBX83.0) アラーム 22050 "maximum speed reached"（最高速度に到達）

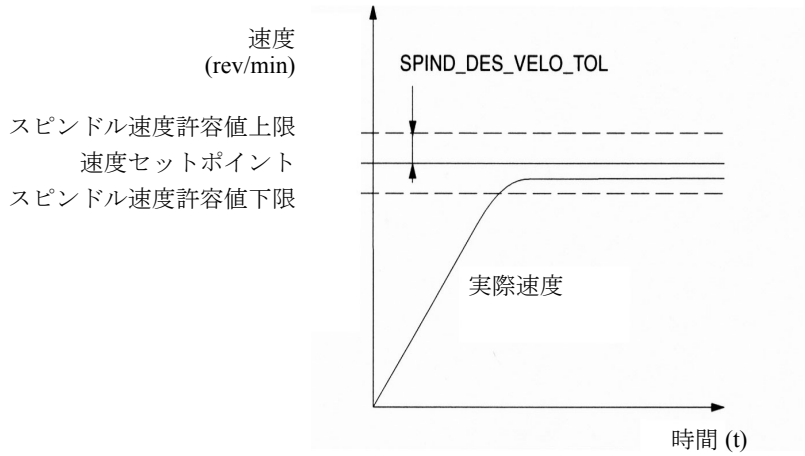
35110 MD 番号	GEAR_STEP_MAX_VELO[n] ギアステップ変更 [ギアステップ番号]: 0...5 （インデックス 0 はスピンドルには関係ない）の最高速度	
初期設定 : 500, 500, 1000, 2000, 4000, 8000	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	GEAR_STEP_MAX_VELO は自動ギアステップ変更 (M40) におけるギアステップの最高速度を定義する。ギアステップはギアステップ間のプログラム可能なスピンドル速度範囲の格差を避けるという意味で, GEAR_STEP_MAX_VELO および MD 35120: GEAR_STEP_MIN_VELO_LIMIT により定義される必要がある。 不正解 GEAR_STEP_MAX_VELO [ギアステップ 1] =1000 GEAR_STEP_MIN_VELO [ギアステップ 2] =1200 正解 GEAR_STEP_MAX_VELO [ギアステップ 1]=1000 GEAR_STEP_MIN_VELO [ギアステップ 2] =950	
関連性	MD 35010: GEAR_STEP_CHANGE_ENABLE （ギアステップ変更は可能） MD 35120: GEAR_STEP_MIN_VELO （ギアステップ変更の最低速度） MD 35140: GEAR_STEP_MIN_VELO_LIMIT （ギアステップの最低速度） MD 35130: GEAR_STEP_MAX_VELO_LIMIT （ギアステップの最高速度）	

35120 MD 番号	GEAR_STEP_MIN_VELO[n] ギアステップ変更の最低速度 [ギアステップ番号]: 0...5	
初期設定 : 50, 50, 400, 800, 1500, 3000	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	GEAR_STEP_MIN_VELO は自動ギアステップ変更 (M40) におけるギアステップの最低速度を定義する。 詳細については MD 35120: GEAR_STEP_MAX_VELO を参照。	
関連性	MD 35110: GEAR_STEP_MAX_VELO （ギアステップ変更の最高速度） MD 35010: GEAR_STEP_CHANGE_ENABLE （ギアステップ変更は可能） MD 35140: GEAR_STEP_MIN_VELO_LIMIT （ギアステップの最低速度） MD 35130: GEAR_STEP_MAX_VELO_LIMIT （ギアステップの最高速度）	

35130 MD 番号	GEAR_STEP_MAX_VELO_LIMIT[n] ギアステップの最高速度 [ギアステップ番号]; 0...5	
初期設定 : 500, 500, 1000, 2000, 4000, 8000	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 :r/min

35130 MD 番号	GEAR_STEP_MAX_VELO_LIMIT[n] ギアステップの最高速度 [ギアステップ番号]; 0...5
データタイプ : DOUBLE	適用開始 SW バージョン :
意味 :	GEAR_STEP_MAX_VELO_LIMIT はギアステップを定義する。この速度は現在使用中のギアステップで超過することはない。
例外，エラー	• 位置制御が起動されていれば，値は 90% （制御確保）まで制限される。 • 使用中のギアステップの最高速度を上回っている S 値がプログラムされると，速度はギアステップ（ギアステップ選択 M41 ～ M45 を使用して）の最高速度まで制限され，「プログラムされた速度が早すぎる」インタフェース信号がイネーブルになる。 • ギアチェンジのための最高速度を上回っている S 値がプログラムされると，新しいギアステップが選択される（自動ギアステップ選択 M40 を使用して）。 • 最も高いギアステップの最高速度を上回っている S 値がプログラムされると，速度はギアステップの最高速度に制限される（自動ギアステップ選択 M40 を使用して）。 • 適切なギアステップのない S 値がプログラムされると，ギアチェンジは起こらない。
関連性	MD 35010: GEAR_STEP_CHANGE_ENABLE （ギアステップ変更は可能） MD 35110: GEAR_STEP_MAX_VELO （ギアステップ変更の最高速度） MD 35120: GEAR_STEP_MIN_VELO （ギアステップ変更の最低速度） MD 35140: GEAR_STEP_MIN_VELO_LIMIT （ギアステップの最低速度）

35140 MD 番号	GEAR_STEP_MIN_VELO_LIMIT[n] ギアステップの最低速度 [ギアステップ番号]: 0...5	
初期設定 : 5, 5, 10, 20, 40, 80	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	GEAR_STEP_MIN_VELO_LIMIT はギアステップの最低速度を定義している。速度は極めて低い S 値がプログラムされていても，この値を下回ることはない。速度は 1.17.2「■ギアステップの最低／最高速度」にリストされている信号／コマンド／ステータスの結果としてのみ，この最低値を下回ることがある。	
この MD は次の場合は無意味	• スピンドルモード発振モード • スピンドルモード位置決めモード，軸モード	
アプリケーション	最低速度を下回ると，モータの円滑な動作は保証されない。	
関連性	MD 35010: GEAR_STEP_CHANGE_ENABLE （ギアステップ変更は可能） MD 35110: GEAR_STEP_MAX_VELO （ギアステップ変更の最高速度） MD 35120: GEAR_STEP_MIN_VELO （ギアステップ変更の最低速度） MD 35130: GEAR_STEP_MAX_VELO_LIMIT （ギアステップの最高速度）	

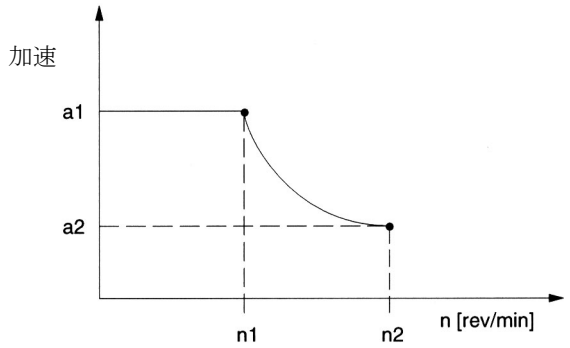
35150 MD 番号	SPIND_DES_VELO_TOL スピンドル速度許容値	
初期設定 : 0.1 1 = 100% 0.1 = 10%	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : 係数
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	スピンドルモード制御モードでは，設定速度 （プログラムされた速度 x スピンドルオフセット，制限つき）が実際速度と比較される。 • 実際速度が設定速度から SPIND_DES_VELO_TOL よりはずれると，「スピンドルがセットポイント範囲内」インタフェース信号 (DB31, ... DBX83.5) がゼロに設定される。 • 実際速度が最高スピンドル速度 (MD 35100: SPIND_VELO_LIMIT) から SPIND_DES_VERLO_TOL より超過すると，「速度はセットポイントを下回っている」インタフェース信号 (DB31, ... DBX83.0) がイネーブルになり，アラーム 22050 "maximumspeed reached" （最高速度に到達）が出力される。チャンネル上のすべての軸およびスピンドルは停止させられる。	
この MD は次の場合は無意味	スピンドルモード発振モード スピンドルモード位置決めモード	
図 1.162		
関連性	MD 35500: SPIND_ON_SPEED_AT_IPO_START MD 35100: SPIND_VELO_LIMIT （最高スピンドル速度） 「スピンドルはセットポイント範囲内」インタフェース信号 (DB31, ... DBX83.5) 「速度制限オーバ」インタフェース信号 (DB31, ... DBX83.0) アラーム "maximum speed reached" （最高速度に到達）	

35160 MD 番号	SPIND_EXTERN_VELO_LIMIT PLC によるスピンドル速度制限	
初期設定 : 1000	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	スピンドル速度の制限は SPIND_EXTERN_VELO_UNIT に入力されており，「速度制限」インタフェース信号 (DB31, ... DBX3.6) がイネーブルになるとき考慮される。NCK はこの値を極めて高いスピンドル速度を制限するために適用する。	

35200 MD 番号	GEAR_STEP_SPEEDCTRL_ACCEL[n] 速度制御モードでの加速 [ギアステップ番号]: 0...5	
初期設定 : 30, 30, 25, 20, 15, 10	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : rev./s ^{2}
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	スピンドルが速度制御モードにあると，加速度は GEAR_STEP_SPEEDCTRL_ACCEL に入力される。 スピンドルは機能 SPCOF を伴って速度制御モードに入る。	
例外，エラー ,	このように速度制御モード (GEAR_STEP_SPEEDCTRL_ ACCEL) での加速度は，現在の制限に到達させられるよう設定される。	
関連性	MD 35210: GEAR_STEP_POSCTRL_ACCEL （位置制御モードでの加速） MD 35220: ACCEL_REDUCTION_SPEED_POINT （速度制限により減少される加速）	

35210 MD 番号	GEAR_STEP_POSCTRL_ACCEL[n] 位置制御モードでの加速度 [ギアステップ番号]: 0...5	
初期設定 : 30, 30, 25, 20, 15,10	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : rev./s ^{2}
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	位置制御モードでの加速度は，現在の制限に到達させられないよう設定される。	
関連性	MD 35200: GEAR_STEP_SPEEDCTRL_ACCEL MD 35220: ACCEL_REDUCTION_SPEED_POINT	

35220 MD 番号	ACCEL_REDUCTION_SPEED_POINT 減少した加速度の速度制限	
初期設定 : 1.0	最小入力リミット : 0.0	最大入力リミット : 1.0
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : 係数
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	スピンドルのトルクは低い速度範囲では一定で，定義された速度（上限速度範囲）を上回って小さくなる。減少したトルクを伴う上位速度範囲は MD: ACCEL_REDUCTION_SPEED_POINT に入力されていなければならない速度で開始する。	
この MD は次の場合は無意味	全速度範囲に渡る一定トルクを伴うスピンドル	
関連性	MD 35210: GEAR_STEP_POSCTRL_ACCEL （位置制御モードでの加速度） MD 35200: GEAR_STEP_SPEEDCTRL_ACCEL （速度制御モードでの加速度） MD 35230: ACCEL_REDUCTION_FACTOR （減少した加速度）	

35230 MD 番号	ACCEL_REDUCTION_FACTOR 減少した加速	
初期設定 : 0	最小入力リミット : 0.	最大入力リミット : 1.0
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : 係数
データタイプ : DOUBLE	適用開始 SW バージョン :	
意味 :	スピンドルのトルクは低い速度範囲では一定で，定義された速度（上限速度範囲）を上回って小さくなる。減少したトルクを伴う上位速度範囲は MD 35220: ACCEL_REDUCTION_SPEED_POINT （減少した加速の速度制限）に定義されている速度で開始する。ACCEL_REDUCTION_FACTOR に定義されている値はギアステップの最高速度 (MD 35130: GEAR_STEP_MAX_VELO_LIMIT) の最大許容加速である。 注意 : 減少した加速は % 表示で入力される。スピンドルが位置制御モードまたは速度制御モードにあるかによって，減少した加速は位置制御モードでの加速 (MD 35210: GEAR_STEP_POSCTRL_ACCEL) または速度制御モードでのアクティブなギアステップの加速 (MD 35200: GEAR_STEP_SPEEDCTRL_ACCEL) に関係する。	
この MD は次の場合は無意味	全速度範囲に渡る一定トルクを伴うスピンドル	
図 1.163	 a1 ... MD: GEAR_STEP_POSCTRL_ACCEL [degrees/s] a2 ... MD: ACCEL_REDUCTION_FACTOR [%] n1 ... MD: ACCEL_REDUCTION_SPEED_POINT [%] n2 ... MD: GEAR_STEP_MAX_VELO_LIMIT [rev/min]	
関連性	MD 35210: GEAR_STEP_POSCTRL_ACCEL （位置制御モードでの加速） MD 35200: GEAR_STEP_SPEEDCTRL_ACCEL （速度制御モードでの加速） MD 35220: ACCEL_REDUCTION_SPEED_POINT （減少される加速の速度制限） MD 35130: GEAR_STEP_MAX_VELO_LIMIT （ギアステップの最高速度）	

35300 MD 番号	SPIND_POSCTRL_VELO 位置制御起動速度	
初期設定 : 500	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	位置制御モードにないスピンドルを位置決めするとき，位置制御はスピンドルが MD: SPIND_POSCTRL_VELO に定義されている速度に到達するまで起動されない。. 速度はパートプログラムから FA[Sn] を使って変更することができる。 様々な補足条件（回転からの位置決め，静止からの位置決め）においての特性をオペレーションするスピンドルの詳細については，1.17.2「■スピンドル位置決めモード」を参照のこと。	
関連性	同期が得られなければ，MD 35350: SPIND_POSITIONING_DIR （静止からの位置決めの間の回転方向）	

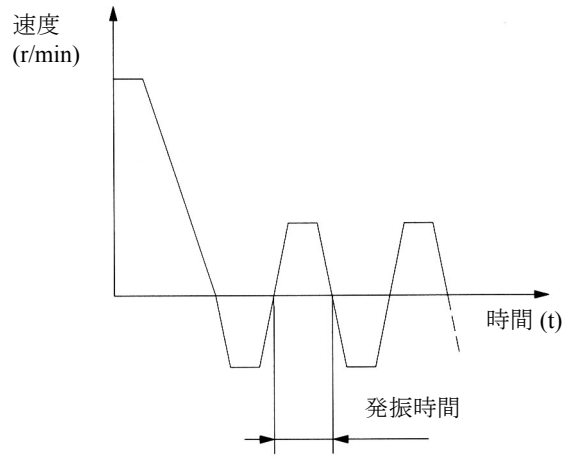
35350 MD 番号	SPIND_POSITIONING_DIR 非同期スピンドルの回転方向の位置決め	
初期設定 : 3	最小入力リミット : 3	最大入力リミット : 4
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 1.1	
意味 :	SPOS または SPOSA がプログラムされると，スピンドルは位置制御モードに切換えられ，スピンドルが同期化されなければ MD 35210: GEAR_STEP_POSCTRL_ACCEL （位置制御モードでの加速）に定義されている加速で加速される。回転方向は MD 35350: SPIND_POSITIONING_DIR で定義されている。 SPIND_POSITIONING_DIR = 3 →　時計回り方向 SPIND_POSITIONING_DIR = 4 →　反時計回り方向	
関連性	MD 35300: SPIND_POSCTRL_VELO （位置制御起動速度）	

35400 MD 番号	SPIND_OSCILL_DES_VELO 発振速度	
初期設定 : 500	最小入力リミット : 0	最大入力リミット : MD にある値 : GEAR_STEP_MAX_VELO_LIMIT
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	発振の間，「発振速度」インタフェース信号 (DB31, ... DBX18.5) はスピンドルモータのモータ速度を選択するのに使われる。このモータ速度は MD: SPIND_OS-CILL_DES_VELO に定義されている。この MD に定義されているモータ速度は，現在のギアステップとは独立している。AUTOMATIC および MDA 表示では，発振速度はギアが変更されるまで「スピンドルセットポイント」ウィンドウに表示される。	
この MD は次の場合は無意味	発振モードを除くすべてのスピンドルモード	
アプリケーション	スピンドルモータへのおよびスピンドルモータからの動作は，ギアホイールの歯が互いによくかみ合うため，新しいギアステップを行うのを容易にする。	
例外，エラー ,	MD 35410: SPIND_OSCILL_ACCEL に定義されている発振加速はこの MD に定義されている発振速度に有効である。	
関連性	MD 35410: SPIND_OSCILL_ACCEL （発振中の加速） 「PLC を介しての発振」インタフェース信号 (DB31, ... DBX18.4) 「発振速度」インタフェース信号 (DB31, ... DBX18.5)	

35410 MD 番号	SPIND_OSCILL_ACCEL 発振加速	
初期設定 : 16	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : rev./s ^{2}
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	指定された加速はスピンドルモータへの発振速度 (MD 35400: SPIND_OSCILL_DES_VELO) の出力にのみ有効である。発振速度は「発振」速度インタフェース信号を使用して選択される。	
この MD は次の場合は無意味	発振モードを除くすべてのスピンドルモード	
関連性	MD 35400: SPIND_OSCILL_DES_VELO （発振速度） 「発振速度」インタフェース信号 (DB31, ... DBX18.5) 「PLC を介しての発振」インタフェース信号 (DB31, ... DBX18.4)	

35430 MD 番号	SPIND_OSCILL_START_DIR **発振開始方向**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 4
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 1.1	
意味 :	「発振速度」インタフェース信号では，スピンドルモータは MD 35400: SPIND_OSCILL_DES_VELO に示された速度に加速する。**開始方向**は「PLC を介しての発振」がイネーブルでなければ SPIND_OSCILL_START_DIR により定義される。 SPIND_OSCILL_START_DIR = 0 → 開始方向は最後の回転方向 SPIND_OSCILL_START_DIR = 1 → 開始方向は最後の回転方向の逆 SPIND_OSCILL_START_DIR = 2 → 開始方向は最後の回転方向の逆 SPIND_OSCILL_START_DIR = 3 → 開始方向は M3 SPIND_OSCILL_START_DIR = 4 → 開始方向は M4	
この MD は次の場合は無意味	発振モードを除くすべてのスピンドルモード	
関連性	MD 35400: SPIND_OSCILL_DES_VELO (発振速度) 「発振速度」インタフェース信号 (DB31, ... DBX18.5) 「PLC を介しての発振」インタフェース信号 (DB31, ... DBX18.4)	

35440 MD 番号	SPIND_OSCILL_TIME_CW M3 **方向の発振時間**	
初期設定 : 1	最小入力リミット : 0 0 は **1 つの**補間サイクル (MD 10070: IPO_SYSCLOCK_TIME_RATIO) を表す	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	ここで定義されている発振時間は M3 方向でアクティブである（図 1.164 の MD 35450: SPIND_OSCILL_TIME_CCW を参照）。	
この MD は次の場合は無意味	・発振モードを除くすべてのスピンドルモード ・PLC を介しての発振（「PLC を介しての発振」インタフェース信号 (DB31, ... DBX18.4) がイネーブルになる）	
関連性	MD 35450: SPIND_OSCILL_TIME_CCW （M4 方向の発振時間） MD 10070: IPO_SYSCLOCK_TIME_RATIO （補間サイクル） 「発振速度」インタフェース信号 (DB31, ... DBX18.5) 「PLC を介しての発振」インタフェース信号 (DB31, ... DBX18.4)	

35450 MD 番号	SPIND_OSCILL_TIME_CCW M4 方向の発振時間	
初期設定 : 0,5	最小入力リミット : 0 0 は 1 つの補間サイクル (MD 10070: IPO_SYSCLOCK_TIME_RATIO) を表す	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : s
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	ここで定義されている発振時間は M4 方向でアクティブである（図 1.164 を参照)。	
この MD は次の場合は無意味	・発振モードを除くすべてのスピンドルモード ・PLC を介しての発振（「PLC を介しての発振」インタフェース信号 (DB31, ... DBX18.4) がイネーブルになる）	
図 1.164		
関連性	MD 35440: SPIND_OSCILL_TIME_CW （M3 方向の発振時間）MD 10070: IPO_SYSCLOCK_TIME_RATIO （補間サイクル） IS 「発振速度」(DB31, ... DBX18.5) IS 「PLC による発振」(DB31, ... DBX18.4)	

35500 MD 番号	SPIND_ON_SPEED_AT_IPO_START セットポイント範囲にあるスピンドルを伴うフィードイネーブル	
初期設定 : 1	最小入力リミット : 0	最大入力リミット : 2
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
データタイプ : BYTE	適用開始 SW バージョン : 4.2	
意味 :	Bit 0: パス補間は影響されない Bit 1: フィード補間はスピンドルが指定された速度（許容範囲は MD 35150 で定義されている）に到達するまでイネーブルにならない（位置決め軸は移動を続ける）。	
	SW 4.2 およびそれ以降のバージョン Byte = 0: パス補間は影響されない Byte = 1: フィード補間はスピンドルが指定された速度（許容範囲は MD 35150 で定義されている）に到達するまでイネーブルにならない（位置決め軸は移動を続ける）。 Byte = 2: 1 に加えて，連続パスモード (G64) では作業ブロックが急速移動 (G0) 後にロードされると，スピンドルが速度セットポイント範囲内でなければパス速度は減速される。 Byte = 3: 1 および 2 の反応は PLC (FC18) および同期化動作によってスピンドル制御に適用される。	
アプリケーション	スピンドルが加速相にあると（プログラムされたセットポイント速度にまだ到達していない），パスフィードは一般的に以下のようにディスエーブルにされなければならない。 • 「スピンドルはセットポイント範囲内」インタフェース信号 (DB31, ... DBX83.5) が求められ，「フォワードフィードディスエーブル」インタフェース信号 (DB21, ... DBX6.0) がイネーブルになる。位置決め軸も停止させられる。 • MD 35500 は 1 に設定される（上記を参照） • MD 35500 は 2 に設定される（上記を参照） • G0 を有するブロックでは MD によって設定された影響する係数はアクティブではない。 • MD 35500 により起動されたパスフィードの特定は以下に従って有効になる。 - 速度の特定 - 回転方向の特定 - NC 開始 速度許容値ウィンドウが少なくとも 1 回到達した後動作を停止する。	
関連性	MD 35150: SPIND_DES_VELO_TOL （スピンドル速度許容値） " セットポイント範囲のスピンドル " インタフェース信号 (DB31, ... DBX83.5)	

35510 MD 番号	SPIND_STOPPED_AT_IPO_START 静止スピンドルを伴うフィードイネーブル	
初期設定 : 100	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	スピンドルが停止されると (M5)，MD SPIND_STOPPED_AT_IPO_START がイネーブルでスピンドルが制御モードにあれば，パスフィードはディスエーブル（位置決め軸は移動を継続する）になる。 スピンドルが静止になる（「静止軸／スピンドル」インタフェース信号 (DB31, ... DBX61.4) がイネーブルになる）と，パスフィードはイネーブルになる。	
アプリケーション	MD 35500: SPIND_ON_SPEED_AT_IPO_START および SPIND_STOPPED_AT_IPO_START は実際スピンドル速度（制御モード）に従ってパスフィードを制御するのに使われる。 • スピンドルが加速相にあれば（プログラムされたセットポイントに到達していなければ），パスフィードはディスエーブルになる。 • 実際速度がスピンドル速度許容値 (MD 35150:SPIND_DES_VELO_TOL) より小さく速度セットポイントからずれると，パスフィードはイネーブルになる。 • スピンドルがブレーキ相にあると，パスフィードはディスエーブルになる。 • スピンドルが静止していると報告されると（「静止軸／スピンドル」インタフェース信号 DB31, ... DBX61.4），フィードはイネーブルになる。 • G0 を伴うブロックでは，変更はアクティブではない。	
関連性	MD 35500: SPIND_ON_SPEED_AT_IPO_START （セットポイント範囲のスピンドルのフィードイネーブル)	

■ スピンドル別設定データ

<table>
<tr><td>42800
SD 番号</td><td colspan="2">SPIND_ASSIGN_TAB
スピンドル番号コンバータ</td></tr>
<tr><td>初期設定 :
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18},
{0, 1, 2, ..., }</td><td>最小入力リミット : 0</td><td>最大入力リミット :
MD_SPIND_ASSIGN_TAB_LENGTH</td></tr>
<tr><td>変更は即座に有効</td><td>保護レベル : 7/7</td><td>単位 :</td></tr>
<tr><td>データタイプ : BYTE</td><td colspan="2">適用開始 SW バージョン : 4.3</td></tr>
<tr><td>意味 :</td><td colspan="2">スピンドルコンバータは物理的（= 内部的，設定された）スピンドル番号にプログラムされた（= 論理的）スピンドル番号を設定する。
設定データ (SD) のインデックスはプログラムされたスピンドル番号またはプログラムされたアドレスエクステンションに対応する。SD は実際に存在する物理的スピンドルを含む。</td></tr>
<tr><td>アプリケーション</td><td colspan="2">スピンドル 1, ..., 3 は常にチャンネルにプログラムされている。実際にチャンネルにあるスピンドルは，プログラムが開始される前に割当てテーブルにある MMC, PLC または パートプログラム により入力される。

SD 42800: SPIND_ASSIGN_TAB[1]=15
SD42800: SPIND_ASSIGN_TAB[2]=18
SD42800: SPIND_ASSIGN_TAB[3]=3
SD42800: SPIND_ASSIGN_TAB[4]=0
M1=3 S1=1000 ; 物理的スピンドル 15 は位置決めされた方向に 1000r/min で回転する。
SPOS[2]=0 ; 物理的スピンドル 18 は 0 度に位置決めされる。
M3=5 ; 物理的スピンドル 3 の停止
R1=$AA_S[4] ; 物理的スピンドルが決定されないためアラーム 16105 が報告される。</td></tr>
<tr><td>例外，エラー ,</td><td colspan="2">注記 :
• ゼロインデックス (SPIND_ASSIGN_TAB[0]) はチャンネルで選択されたマスタスピンドルを表示するためのみに使われ，上書きしてはならない。
• スピンドルコンバータへの変更は即座に有効になる。
それゆえ，プログラムが起動している間，スピンドルコンバータをパートプログラムで使用されているスピンドルの MMC または PLC から変更することは適切ではない。
• 「SRAM 削除」後，論理的および物理的スピンドルの番号は同一のものになる。</td></tr>
</table>

43210 SD 番号	SPIND_MIN_VELO_G25 プログラムされたスピンドル速度制限 G25	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 正
変更は即座に有効	保護レベル : MMC-MD 9220	単位 : r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	スピンドル速度は SPIND_MIN_VELO_G25 に入力されている最高スピンドル速度制限を超過することは許されない。NCK はこの値に対して過度に低いスピンドル速度設定を制限する。 スピンドル速度は以下の場合のみ下限を下回ることができる。 ・スピンドルオーバライド 0% ・M5 ・S0 ・「スピンドル停止」インタフェース信号 (DB31, ... DBX8.3) ・「コントローラディスエーブル」インタフェース信号 (DB31, ... DBX2.1) ・「リセット」インタフェース信号 (DB21, ... DBX35.7) ・「スピンドルリセット」インタフェース信号 (DB31, ... DBX2.2) ・「発振速度」インタフェース信号 (DB31, ... DBX18.5) ・S 値を削除	
この SD は次の場合は無意味	制御モードを除くすべてのスピンドルモード (SPOS, SPOSA)	
アプリケーション	スピンドルに搭載されているのは，過度に低い速度でバランスが取れない（チャンバーから流れ出る水の形を取った分銅），水で調節されたツール（研削ホイールなど）である。	
例外，エラー，	SPIND_MIN_VELO_G25 の値は以下を使って変更できる。 ・パートプログラム中の G25 S.... ・MMC を介したオペレータコマンド SPIND_MIN_VELO_G25 の値は電源を切る形のリセット後も保持される。	
関連性	SD 43220: SPIND_MAX_VELO_G26 SD 43230: SPIND_MAX_VELO_LIMS （プログラムされたスピンドル速度制限 G96）	

43220 SD 番号	SPIND_MAX_VELO_G26 プログラムされたスピンドル速度制限 G26	
初期設定 : 1000	最小入力リミット : 0	最大入力リミット : 正
変更は即座に有効	保護レベル : MMC-MD 9220	単位 : r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	最高スピンドル速度は SPIND_MAX_VELO_G26 に入力されており，スピンドルはその速度を超えてはならない。NCK はこの値への過度のスピンドル速度セットポイントを制限する。	
この SD は次の場合は無意味	制御モードを除くすべてのスピンドルモード	
例外，エラー，	SD: SPIND_MIN_VELO_G26 の値は以下を使って変更できる。 ・パートプログラム中の G26 S.... ・MMC を介したオペレータコマンド SPIND_MIN_VELO_G26 の値は電源を切る形のリセット後も保持される。	
関連性	SD 43210: SPIND_MIN_VELO_G25 （プログラムされたスピンドル速度制限 G25） SD 43230: SPIND_MAX_VELO_LIMS （プログラムされたスピンドル速度制限 G96）	

43230 SD 番号	SPIND_MAX_VELO_LIMS プログラムされたスピンドル速度制限 G96	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 正
変更は即座に有効	保護レベル : MMC-MD 9220	単位 : r/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	速度が一定であれば (G96 および G97), 一定してアクティブである制限に加えて SPIND_MAX_VELO_LIMS に入力されている更なる制限がスピンドル上で働く。さらに，制限はパートプログラムで LIMS=..... を使用して書くことができる。	
この SD は次の場合は無意味	G96 および G97 (定切削速度) を除くすべてのスピンドル機能	
アプリケーション	カットオフなどの極めて小さい加工径では，ワーク（旋盤上の）を伴って定切削速度 (G96) で回転するスピンドルはますます高速になり，対面する軸 X= 0 であれば，理論上無限の高速セットポイントに到達する。このようなケースでは，スピンドルは現在のギアステップ（適用できれば G26 により制限される）で最高スピンドル速度まで回転 する。スピンドルが特に低い速度まで制限されると，LIMS=.... SPIND_MAX_VELO_ を設定することができる。	
例外，エラー ,	SD 43210: SPIND_MIN_VELO_LIMS の値は以下を使用して変更することができる。 ・パートプログラム中の LIMS S.... ・MMC を介したオペレータコマンド SPIND_MIN_VELO_LIMS の値は電源を切るか，またはリセット後も保持される。	
関連性	SD 43220: SPIND_MAX_VELO_G26 （最高スピンドル速度） SD 43210: SPIND_MIN_VELO_G25 （最高スピンドル速度）	

1.17.5 信号の説明

■ 軸／スピンドル別信号

■ 軸／スピンドルへの信号

DB31, ... DBX2.2 **データブロック**	**スピンドルリセット／移動距離削除** **軸／スピンドルへの信号** (PLC -> NCK)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号遷移 0 → 1	MD 35040: SPIND_ACTIVE_AFTER_RESET とは別に，スピンドルリセットには以下のような効果がある。 制御モード : 発振モード : 位置決めモード : 軸モード :	 - スピンドル停止 - プログラム継続 - スピンドルは以下の M および S プログラムコマンドを使用して起動継続する。 - 発進は中断される - 軸は移動を継続する - プログラムは現在のギアステップで継続される - 以下の M 値およびそれ以上の S 値を伴って，必要に応じて「プログラムされた速度が速すぎる」(DB31, ... DBX83.1) インタフェース信号がイネーブルになる。 - 停止させられる - 停止させられる
信号ステータス 0 あるいは信号 遷移 1 → 0	影響なし	
関連性	MD 35040: SPIND_ACTIVE_AFTER_RESET （独自のスピンドルリセット） 「リセット」インタフェース信号 (DB21, ... DBX7.7) 「移動距離削除」インタフェース信号 (DB31, ... DBX2.2) は同じ信号の別名	

DB31, ... DBX16.7 **データブロック**	**S 値を削除** **軸／スピンドルへの信号** (PLC -> NCK)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号遷移 0 → 1	制御モード :	- スピンドル停止 - プログラム継続 - スピンドルは M3 または M4 がアクティブであれば，以下の S 値を使用して起動継続する。
	発振モード, 軸モード, 位置決めモード :	- 信号は適切な機能に有効ではない。しかし，「制御モード」が再び選択されると，新しい S 値がプログラムされなければならない。
信号ステータス 0 あるいは信号遷移 1 → 0	影響なし	
アプリケーション	外部信号（検出プローブなど）のため移動動作を終了させる。	

DB31, ... DBX16.5, 16.4 **データブロック**	**スピンドル 1 および 2 を再同期化** **軸／スピンドルへの信号** (PLC -> NCK)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号遷移 0 → 1	スピンドル上の位置測定系と 0 度位置の間の同期が失われているため，スピンドルは再同期化されなければならない。	
信号ステータス 0 あるいは信号遷移 1 → 0	影響なし	
この信号は次の場合は無意味	制御モードを除くすべてのスピンドルモード	
アプリケーション	マシンには垂直および水平のスピンドルのセレクターがある。**2 つの**異なった位置エンコーダが使用される（1 つは垂直のスピンドルに，もう 1 つは水平のスピンドルに）が，制御ではただ 1 つの実際値入力が使用される。システムが垂直から水平のスピンドルに切換わると，スピンドルは再同期化されなければならない。同期化は「再同期化スピンドル 1 および 2」インタフェース信号で行われる。	
関連性	「基準化／同期化 1」インタフェース信号 (DB31, ... DBX60.4) 「基準化／同期化 2」インタフェース信号 (DB31, ... DBX60.5)	

DB31, ... DBX16.3 **データブロック**	**ギアチェンジ** **軸／スピンドルへの信号** (PLC -> NCK)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	新しいギアステップが「実際ギアステップ A ～ C」が用いられると，「ギアチェンジ」インタフェース信号がイネーブルになる。これにより NCK に正しいギアステップが首尾よく用いられていることが伝えられる。ギアステップ変更は完了（スピンドル発振モードが選択解除される）し，スピンドルは最後にプログラムされたスピンドル速度で新しいギアステップで加速し，パートプログラムの次のブロックが実行される。「ギアチェンジ」インタフェース信号が NCK によりリセットされ，それにより PLC アプリケーションが「ギアチェンジ」信号をリセットする。	
信号ステータス 0 あるいは信号 遷移 1 → 0	影響なし	
この信号は次の場合は無意味	発振モードを除くすべてのスピンドルモード	
例外，エラー ,	PLC アプリケーションが NCK により設定されたギアステップとして出されたものと異なる実際のギアステップで NCK に報告すると，ギアチェンジは首尾よく完了されたとして扱われ，実際のギアステップ A ～ C が起動される。	
関連性	「実際のギアステップ A ～ C」(DB31, ... DBX16.2 ～ 16.0) 「ギアステップ A ～ C の設定」インタフェース信号 (DB31, ... DBX82.2 ～ 82.0) 「チェンジギア」インタフェース信号 (DB31, ... DBX82.3) 「発振速度」インタフェース信号 (DB31, ... DBX18.5)	

DB31, ... DBX16.2 - 16.0 **データブロック**	**実際のギアステップ A ～ C** **軸／スピンドルへの信号** (PLC -> NCK)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 （レベルオペレーションされた）	新しいギアが用いられると「実際のギアステップ A ～ C」および「ギアチェンジ」インタフェース信号は PLC ユーザーによりイネーブルになる。これにより NCK に正しいギアステップが首尾よく用いられていることが伝えられる。 ギアステップ変更は完了（スピンドル発振モードが選択解除される）し，スピンドルは最後にプログラムされたスピンドル速度で新しいギアステップで加速し，パートプログラムの次のブロックが実行される。 実際のギアステップの詳細はコード化されている。 5 つのギアステップに次の割当てでそれぞれ 1 つのパラメータセットが提供されます。	

パラメータブロック番号	VDI インタフェース	データブロックのデータ	内容
0	-	軸モードのデータ	Kv 係数 監視
1	000	1 番目のギアステップのデータ	M40 速度
	001		最低／最高速度 加速
2	010	2 番目のギアステップのデータ	その他
3	011	3 番目のギアステップのデータ	
4	100	4 番目のギアステップのデータ	
5	101	5 番目のギアステップのデータ	
	110		
	111		

例外，エラー，	PLC アプリケーションが NCK により設定されたギアステップとして出されたものと異なる実際のギアステップで NCK に報告すると，ギアチェンジは首尾よく完了されたとして扱われ，実際のギアステップ A ～ C が起動される。
関連性	「実際のギアステップ A ～ C」インタフェース信号 (DB31, ... DBX82.0 ～ DBX82.2) 「チェンジギア」インタフェース信号 (DB31, ... DBX82.3) 「チェンジギア」インタフェース信号 (DB31, ... DBX16.3) 「発振速度」インタフェース信号 (DB31, ... DBX18.5) ギアステップのパラメータセット

DB31, ... DBX17.6 **データブロック**	**M3/M4 を反転** **軸／スピンドルへの信号** (PLC -> NCK)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	スピンドルモータの回転方向は以下の機能で変更する。 ・M3 ・M4 ・M5 ・動作中の SPOS/SPOSA，静止からの SPOS/SPOS にはアクティブではない。	
アプリケーション	マシンには垂直および水平スピンドルの選択スイッチがある。機械上のデザインでは，水平のスピンドル上では垂直のスピンドルより **1 つ多いギアホイール** を組み込まれている。スピンドルが常に M3 を使用して右に回転するなら，回転方向はそれゆえ垂直のスピンドル上で変更される。	

DB31, ... DBX17.5, 17.4 **データブロック**	**位置決め 2 および 1 中のスピンドルの再同期化** **軸／スピンドルへの信号** (PLC -> NCK)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 4.0
信号ステータス 1	スピンドルは位置決め中に再同期化される。	
信号ステータス 0 あるいは信号 遷移 1 → 0	影響なし	
この信号は次の場合は無意味	位置決めモードを除くすべてのスピンドルモード	
アプリケーション	スピンドルには間接的測定系があり，モータとクランプの間にスリップを引き起こすことがある。位置決め処理を開始するとき信号 =1 であれば，古い基準は削除され，終了位置の前に再度検索されたゼロマークにアプローチする。	
関連性	「基準化／同期化 1」インタフェース信号 (DB31, ... DBX60.4) 「基準化／同期化 2」インタフェース信号 (DB31, ... DBX60.5)	

DB31, ... DBX18.7, 18.6 **データブロック**	**回転方向を逆時計回りに設定／回転方向を時計回りに設定** **軸／スピンドルへの信号** (PLC -> NCK)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	「PLC を介した発振」インタフェース信号がイネーブルになると，2 つのインタフェース信号「回転方向セットポイント左」および「回転方向セットポイント右」が発振速度の回転方向を設定するのに使用される。スピンドルモータの発振動作の時間は対応する長さの時間の「回転方向セットポイント左および右」インタフェース信号をイネーブルにすることにより定義される。	
この信号は次の場合は無意味	発振モードを除くすべてのスピンドルモード	
アプリケーション	「PLC を介した発振」インタフェース信号を参照	
例外，エラー，	・両方のインタフェース信号が同時にイネーブルになると，発振速度は出力されない。 ・両方のインタフェース信号が同時にイネーブルにならなければ，発振速度は出力されない。	
関連性	「PLC を介した発振」インタフェース信号 (DB31, ... DBX18.4) 「発振速度」(DB31, ... DBX18.5)	

DB31, ... DBX18.5 **データブロック**	**発振速度** **軸／スピンドルへの信号** (PLC -> NCK)	
エッジ評価 : なし	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	ギアステップが変更される場合（「ギアチェンジ」インタフェース信号 (DB31-48, DBX82.3) がイネーブルになる），スピンドルは発振モードに変更される。 スピンドルは「発振速度」インタフェース信号 (DB31-48, DBX18.5) がイネーブルになった時点に従って，異なった加速レベルで静止まで減速する。 1 番目 「ギアチェンジ」インタフェース信号がイネーブルになる**前**に NCK により「発振速度」インタフェース信号がイネーブルになる。スピンドルは発振中の加速を伴って静止に持ちこまれる (MD: SPIND_OSCILL_ACCEL)。発振はスピンドルが静止していると即座に開始する。 2 番目 「ギアチェンジ」インタフェース信号がイネーブルになった**後**およびスピンドルが停止していると，NCK により「発振速度」インタフェース信号がイネーブルになる。位置制御は起動停止させられる。スピンドルは速度制御モードで加速を伴って減速する。 「発振速度」がイネーブルになると，スピンドルは発振加速 (MD:SPIND_OSCILL_ACCEL) を伴って発振を開始する。 「PLC を介した発振」インタフェース信号 (DB31, ... D18.4) が**イネーブルでなければ**,「発振速度」インタフェース信号は NCK 上で自動発振を起動する。回転方向の 2 つの時間は MD: SPIND_OSCILL_TIME_CW （M3 方向の発振時間）および MD: SPIND_OSCILL_TIME_CCW （M4 方向の発振時間）中に入力されている。 「PLC を介した発振」インタフェース信号が**イネーブル**になると，「発振速度」インタフェース信号は「回転方向セットポイント左および右」インタフェース信号と関連してのみ速度の出力を行う。回転方向の連続変更などの発振が「回転方向セットポイント左および右」インタフェース信号（PLC を介した発振）を伴って PLC アプリケーションにより実行される。	
信号ステータス 0 あるいは信号 遷移 1 → 0	スピンドルは発振しない。	
この信号は次の場合は無意味	発振モードを除くすべてのスピンドルモード	
アプリケーション	発振速度は新しいギアステップの採用を促進するために使用される。	
関連性	PLC を介した発振 (DB31, ... DBX18.4) 回転方向を逆時計回りに設定 (DB31, ... DBX18.7) 回転方向を時計回りに設定 (DB31, ... DBX18.6)	

DB31, ... DBX18.4 **データブロック**	**PLC を介した発振** **軸／スピンドルへの信号** (PLC -> NCK)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	「PLC を介した発振」インタフェース信号が**イネーブルでなければ**,「発振速度」インタフェース信号は NCK 上で自動発振を起動する。回転方向の 2 つの時間は MD 35440: SPIND_OSCILL_TIME_CW （M3 方向の発振時間）および MD 35450: SPIND_OSCILL_TIME_CCW （M4 方向の発振時間）中に入力されている。 「PLC を介した発振」インタフェース信号が**イネーブル**になると,「発振速度」インタフェース信号は「回転方向セットポイント左および右」インタフェース信号と関連してのみ速度の出力を行う。回転方向の連続変更などの発振が「回転方向セットポイント左および右」インタフェース信号（PLC を介した発振）を伴って PLC ユーザーにより実行される。	
アプリケーション	新しいギアステップが NCK による数回の試みにも係わらず用いられなければ, システムは PLC を介した発振に切換えられる。そのとき PLC ユーザーにより両方の時間は変更される。これにより好ましくないギアホイール位置であっても確実なギアステップの変更ができる。	
関連性	MD 35440: SPIND_OSCILL_TIME_CW （M3 方向の発振時間） MD 35450: SPIND_OSCILL_TIME_CCW （M4 方向の発振時間） 「発振速度」インタフェース信号 (DB31, ... DBX18.5) 「逆時計回りに設定」インタフェース信号 (DB31, ... DBX18.7) 「時計回りに設定」インタフェース信号 (DB31, ... DBX18.6)	

■ 軸／スピンドルからの信号

DB31, ... DBX60.0 **データブロック**	**スピンドル／軸ではない** **軸／スピンドルからの信号** (NCK -> PLC)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	マシン軸は以下のスピンドルモードでスピンドルとしてオペレーションされる ・制御モード ・発振モード ・位置決めモード ・リジッドタッピング ・同期化モード 軸へのインタフェース信号 (DB31, ... DBB12 ～ 15) および軸からのインタフェース信号 (DB31, ... DBB74 ～ 81) は無効。 スピンドルへのインタフェース信号 (DB31, ... DBB16 ～ 19) およびスピンドルからのインタフェース信号 (DB31, ... DBB82 ～ 91) は有効。	
信号ステータス 0 あるいは信号 遷移 1 → 0	マシン軸は軸としてオペレーションされる 軸へのインタフェース信号 (DB31, ... DBB12 ～ 15) および軸からのインタフェース信 (DB31, ... DBB74 ～ 81) は有効。スピンドルへのインタフェース信号 (DB31, ... DBB16 ～ 19) およびスピンドルからのインタフェース信号 (DB31, ... DBB82 ～ 91) は無効。	
アプリケーション	スピンドルが時々マシンツール（スピンドル／ C- 軸を伴う旋盤またはリジッドタッピングのスピンドル／回転軸を伴うフライス盤）上で回転軸として使用される場合,「スピンドル／軸ではない」信号が, マシン軸が軸モードまたはスピンドルモーにあるかを判断するために使用される。	

DB31, ... DBX82.3 **データブロック**	**ギアチェンジ** **軸／スピンドルからの信号** (NCK -> PLC)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	ギアステップは以下のように選択される。 ・パートプログラム (M41 ～ M45) により永久的に ・プログラムされたスピンドル速度 (M40) により自動的に M41 ～ M45: ・ギアステップは M41 ～ M45 を使用して永久的に選択される。ギアステップが M41 ～ M45 により指定され，現在の（実際の）ギアステップと同じでなければ，「チェンジギア」インタフェース信号 (DB31, ...DBX82.3) および「ギアステップ A ～ C の設定」インタフェース信号がイネーブルになる。 M40: ・パートプログラム中の M40 によりギアステップは制御により自動的に選択される。制御装置はどのギアステップがプログラムされたスピンドルに適しているかをチェックする。提案されたギアステップが現在（実際）のギアステップと同じでなければ，「チェンジギア」インタフェース信号 および「ギアステップ A ～ C の設定」インタフェース信号がイネーブルになる。 ・信号 = 1 であれば，チャンネルは「ギアチェンジ待機中」のオペレーションメッセージを表示する。	
例外，エラー ,	「チェンジギア」インタフェース信号は，現在の実際のギアステップと**同じでない**新しいギアステップが選択されたときのみイネーブルになる。	
関連性	「ギアステップ A ～ C の設定」インタフェース信号 (DB31, ... DBX82.0 ～ 82.2) 「実際のギアステップ A ～ C」インタフェース信号 (DB31, ... DBX16.0 ～ D16.2) 「チェンジギア」インタフェース信号 (DB31, ... DBX16.3)	

DB31, ... DBX82.2 - 82.0 **データブロック**	**ギアステップ A ～ C の設定** **軸／スピンドルからの信号** (NCK -> PLC)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	ギアステップは以下のように選択される。 ・パートプログラム (M41 ～ M45) により永久的に ・プログラムされたスピンドル速度 (M40) により自動的に M41 ～ M45: ・ギアステップは M41 ～ M45 を使用して永久的に選択される。M41 ～ M45 コマンドが，現在の（実際の）ギアステップと同じでないギアステップを選択すると，「チェンジギア」インタフェース信号 (DB31-48, ...DBX82.3) および「ギアステップ A ～ C の設定」インタフェース信号がイネーブルになる。 M40: ・パートプログラム中の M40 によりギアステップは制御により自動的に選択される。制御装置はどのギアステップがプログラムされたスピンドル速度（S 値）に適しているかをチェックする。現在（実際）のギアステップと同じでないギアステップが見つかると，「チェンジギア」(DB31, ... D82.3) および「ギアステップ A ～ C の設定」インタフェース信号がイネーブルになる。 設定されたギアステップはコード化された形で以下のように出力される。 1 番目のギアステップ　0 0 0 (C B A) 1 番目のギアステップ　0 0 1 2 番目のギアステップ　0 1 0 3 番目のギアステップ　0 1 1 4 番目のギアステップ　1 0 0 5 番目のギアステップ　1 0 1 無効な値　1 1 0 無効な値　1 1 1	
この信号は次の場合は無意味	発振モードを除く他のスピンドルモード	
関連性	「チェンジギア」インタフェース信号 (DB31, ... D82.3) 「ギアステップ A ～ C の設定」インタフェース信号 (DB31, ... DBX16.0 ～ 16.2) 「チェンジギア」インタフェース信号 (DB31, ... DBX16.3)	

DB31, ... DBX83.7 **データブロック**	**実際の回転方向右** **軸／スピンドルからの信号** (NCK -> PLC)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	スピンドルが回転していると，実際の回転方向が右である「実際の回転方向右」= 1 が表示される。実際の回転方向はスピンドル位置測定エンコーダから得られる。	
信号ステータス 0 あるいは信号 遷移 1 → 0	スピンドルが回転していると，実際の回転方向が左である「実際の回転方向右」= 0 が表示される。	
この信号は次の場合は無意味	・静止スピンドル（回転方向の評価は不可能） ・位置測定エンコーダのないスピンドル	
関連性	「静止スピンドル」インタフェース信号 (DB31, ... DBX61.4)	

DB31, ... DBX83.5 **データブロック**	**セットポイント範囲のスピンドル** **軸／スピンドルからの信号** (NCK -> PLC)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	「セットポイント範囲のスピンドル」インタフェース信号はプログラムされた，制限されたと考えられるスピンドル速度に到達したかを報告する。スピンドル制御モードでは，速度セットポイント（プログラムされた速度＋制限を含むスピンドルオーバライド）が実際速度と比較される。実際速度が設定された速度から MD 35150: SPIND_DES_VELO_TOL にあるスピンドル速度許容値より**小さく**ずれる場合，「セットポイント範囲のスピンドル」インタフェース信号がイネーブルになる。	
信号ステータス 0 あるいは信号 遷移 1 → 0	「セットポイント範囲のスピンドル」インタフェース信号はスピンドルがまだ加速またはブレーキ相にあるかを報告する。 スピンドル制御モードでは，速度セットポイント（プログラムされた速度 x スピンドルオーバライド）が実際速度と比較される。実際速度が設定された速度から SPIND_DES_VELO_TOL にあるスピンドル速度許容値より**大きく**ずれる場合，「スピンドルはセットポイント範囲」インタフェース信号がリセットされる。	
この信号は次の場合は無意味	速度モード（制御モード）を除くすべてのスピンドルモード	
アプリケーション	アプリケーション スピンドルが加速相にあると（プログラムされた速度セットポイントに到達していない）パスフィードは通常ディスエーブルになる。これは以下のように実行される。 ・「セットポイント範囲のスピンドル」インタフェース信号は評価され，「フィードディスエーブル」インタフェース信号 (DB21, ... DBX6.0) がイネーブルになる。位置決め軸も停止させられる。 ・MD 35500: SPIND_ON_SPEED_AT_IPO_START （セットポイント範囲のスピンドルのフィードイネーブル）がイネーブルになり，NCK は内部でスピンドルがセットポイント範囲にあるかを評価する。パスフィードはスピンドルがセットポイント範囲にあるときのみイネーブルになる。位置決め軸はこの機能により停止させられることはない。	
関連性	MD 35500: SPIND_DES_VELO_TOL （スピンドル速度許容値）	

DB31, ... DBX83.2 **データブロック**	**セットポイント速度増加（プログラムされた速度が低すぎる）** **軸／スピンドルからの信号** (NCK -> PLC)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	スピンドル速度 (1/min) または定切削速度 (m/min または ft/min) がプログラムされると，値は以下の制限のうちひとつを下回るものになる。 ・指定されたギアステップの最低速度 ・最低スピンドル速度 ・VDI を通しての速度制限 ・プログラムされたスピンドル速度制限 G25 ・G96 のプログラムされたスピンドル速度制限 スピンドル速度は最低制限までに制限される。	
信号ステータス 0 あるいは信号 遷移 1 → 0	スピンドル速度 (1/min) または定切削速度 (m/min または ft/min) がプログラムされると，値は制限のどれをも下回らない。	
アプリケーション	「セットポイント速度増加」インタフェース信号はプログラムされた速度が達成されないことを検知するのに使用される。PLC ユーザーはこのステータスを許容可能だと受け入れフィードをイネーブルにするか，またはパスフィードをディスエーブルにし，および／または「スピンドルがセットポイント範囲内」インタフェース信号が処理されるチャンネルを完了することができる。	

DB31, ... DBX83.1 **データブロック**	**制限されたセットポイント速度（プログラムされた速度が速すぎる）** **軸／スピンドルからの信号** (NCK -> PLC)	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	スピンドル速度 (1/min) または定切削速度 (m/min または ft/min) がプログラムされると，値は以下の制限のうちひとつを超過するものになる。 ・選択されたギアステップの速度 ・最高スピンドル速度 ・VDI を通しての速度制限 ・プログラムされたスピンドル速度制限 G26 ・G96 のプログラムされたスピンドル速度制限 スピンドル速度は最高制限までに制限される。	
信号ステータス 0 あるいは信号 遷移 1 → 0	スピンドル速度 (1/min) または定切削速度 (m/min または ft/min) がプログラムされると，値は制限のどれをも超過しない。	
アプリケーション	「制限されたセットポイント速度」インタフェース信号はプログラムされた速度が達成されないことを検知するのに使用される。PLC ユーザーはこのステータスを許容可能だと受け入れフィードをイネーブルにするか，またはパスフィードをディスエーブルにし，および／または「スピンドルがセットポイント範囲内」インタフェース信号が処理されるチャンネルを完了することができる。	

DB31, ... DBX83.0 **データブロック**	**速度制限オーバ** **軸／スピンドルからの信号** (NCK -> PLC)	
エッジ評価 : あり	信号の更新：周期的	信号が有効になる SW バージョン：1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	実際の速度が最高スピンドル速度 MD 35100: SPIND_VELO_ LIMIT をスピンドル速度許容値 MD 35150: SPIND_DES_VELO_TOL より超過すると，「速度制限オーバ」インタフェース信号がイネーブルになり，アラーム 22050 "maximum speed reached" （最高速度に到達）が出力される。チャンネル上のすべての軸とスピンドルは停止させられる。	
関連性	MD 35150: SPIND_DES_VELO_TOL （スピンドル速度許容値） MD 35100: SPIND_VELO_LIMIT （最高スピンドル速度） アラーム 22050 "maximum speed reached" （最高速度に到達）	

DB31, ... DBX84.7 **データブロック**	**アクティブなスピンドル制御モード** **軸／スピンドルからの信号** (NCK -> PLC)	
エッジ評価 : あり	信号の更新：周期的	信号が有効になる SW バージョン：1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	スピンドルは以下の機能を有する制御モードにある。 ・スピンドルの回転方向詳細 M3/M4 またはスピンドル停止 M5 ・M41...M45 または自動ギアステージ切換え	
関連性	「スピンドル発振モード」インタフェース信号 (DB31, ... DBX84.6) 「スピンドル位置決めモード」インタフェース信号 (DB31, ... DBX84.5)	

DB31, ... DBX84.6 **データブロック**	**アクティブなスピンドルモード発振モードデータブロック** **軸／スピンドルからの信号** (NCK -> PLC)	
エッジ評価 : あり	信号の更新：周期的	信号が有効になる SW バージョン：1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	新しいギアステップが自動ギアステップ選択 (M40) または M41 ～ M45 （「ギアチェンジ」インタフェース信号がイネーブルになる）を使用していると定義されれば，スピンドルは発振モードになる。「ギアチェンジ」インタフェース信号は，現在の実際のギアステップと**同じでない**新しいギアステップが選択されたときのみイネーブルになる。	
関連性	「スピンドル制御モード」インタフェース信号 (DB31, ... DBX84.7) 「スピンドル位置決めモード」インタフェース信号 (DB31, ... DBX84.5 「ギアチェンジ」インタフェース信号 (DB31, ... DBX82.3)	

DB31, ... DBX84.5 **データブロック**	**アクティブなスピンドル位置決めモード** **軸／スピンドルからの信号 (NCK -> PLC)**	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	スピンドルは以下の機能を有する位置決めモードにある。 ・SPOS [n] = ... ・SPOS [n] = AC(.....) ・SPOS [n] = IC(.....) ・SPOS [n] = DC(.....) ・SPOS [n] = ACN(.....) ・SPOS [n] = ACP(.....) ・SPOSA [n] = ... ・SPOSA [n] = AC(.....) ・SPOSA [n] = IC(.....) ・SPOSA [n] = DC(.....) ・SPOSA [n] = ACN(.....) ・SPOSA [n] = ACP(.....) コマンドパラメータ [n] はマスタスピンドルには必要ではない。	
関連性	「スピンドル制御モード」インタフェース信号 (DB31, ... DBX84.7) 「スピンドルモード発振モード」インタフェース信号 (DB31, ... DBX84.6)	

DB31, ... DBX84.3 **データブロック**	**リジッドタッピングがアクティブ** **軸／スピンドルからの信号 (NCK -> PLC)**	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	スピンドルはリジッドタッピング機能 (G331/G332) 中に内部的に軸モードに切換えられる。リジッドタッピングでは，スピンドル速度はまた rpm で表示される S.... を伴ってプログラムされるが，回転方向はスレッドリードのリーディングサインとして保存されている。以下のようなスピンドル別インタフェース信号は，どれも更新されるものはない。 「スピンドルリセット」 「同期スピンドル」 「M3/M4 変換」 「セットポイント範囲のスピンドル」 「プログラム可能な速度が速すぎる」	
アプリケーション	以下のような特定の機能はリジッドタッピング中に使用されてはならない。 ・「制御イネーブル」インタフェース信号 (DB31, ... DBX2.1) リセット ・「設定フィード停止」インタフェース信号 (DB31, ... DBX8.3) ・リセット ・リジッドタッピング中に緊急停止を起動するとき，ツールとワークは共にロックされていなければならない。	

DB31, ... DBB86, DBB87 **データブロック**	**スピンドルの M 機能** **軸／スピンドルからの信号 (NCK -> PLC), 軸別**	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	M 機能は DBs 31, ... では軸別に，DBs 21, ... ではチャンネル別に出力される。 **スピンドルの選択された M 機能**は PLC に「スピンドルの M 機能」インタフェース信号で出力される。スピンドルがアドレスエクステンションを伴ってまたは伴わずにプログラムされたか，およびどのチャンネルでプログラムされたかは関係ない。選択された以下の M 機能はここで出力される。 ・M3 ・M4 ・M5 ・M70	
関連性	「スピンドルの S 機能」インタフェース信号 (DB31, ... DBB88 ～ 91), 軸別 「スピンドルの M 機能」インタフェース信号 (DB21, ... DBB58 および DBB68 ～ DBB97), チャンネル別	

DB31, ... DBB88-91 **データブロック**	**スピンドルの S 機能** **軸／スピンドルからの信号 (NCK -> PLC), 軸別**	
エッジ評価 : あり	信号の更新 : 周期的	信号が有効になる SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	S 機能は DBs 31, ... では軸別に，DBs 21, ... ではチャンネル別に出力される。 **スピンドルの選択された S 機能**は PLC に「スピンドルの S 機能」インタフェース信号で出力される。スピンドルがアドレスエクステンションを伴ってまたは伴わずにプログラムされたか，およびどのチャンネルでプログラムされたかは関係ない。 選択された以下の S 機能はここで出力される。 ・S.... r/min で表示されるスピンドル速度として（プログラムされた値） ・S.... m/min または ft/min で表示される定切削速度として 選択された以下の S 機能はここでは出力されない。 ・S.... プログラムされたスピンドル速度制限 G25 として ・S.... プログラムされたスピンドル速度制限 G26 として ・S.... スピンドルが制御上定義されていない場合，r/min で表示されるスピンドル速度として ・S.... スピンドル回転での休止時間として	
関連性	「スピンドルの M 機能」インタフェース信号 (DB31, ... DBB86 および 87), 軸別 「スピンドルの S 機能」インタフェース信号 (DB21, ... DBB60 および DBB98 ～ DBB115), チャンネル別	

1.17.6 例

- 例なし -

1.17.7 データフィールド，リスト

■ インタフェース信号

DB 番号	ビット，バイト	名称	参照
軸別			
31, ...	0	フィードオーバライド	1.18（V1）
31, ...	1.7	オーバライドアクティブ	1.18（V1）
31, ...	1.6	位置測定系 2	1.1（A2）
31, ...	1.5	位置測定系 1	1.1（A2）
31, ...	1.4	フォローアップモード	1.1（A2）
31, ...	1.3	軸／スピンドルディスエーブル	1.1（A2）
31, ...	2.2	スピンドルリセット／移動距離削除	1.1（A2）
31, ...	2.1	コントローライネーブル	1.1（A2）
31, ...	3.6	スピード／スピンドル速度制限	1.2（A3）
31, ...	16.7	S 値削除	
31, ...	16.5	再同期スピンドル 2	
31, ...	16.4	再同期スピンドル 1	
31, ...	16.3	ギアチェンジ	
31, ...	16.2-16.0	実際のギアステップ A ～ C	
31, ...	17.6	M3/M4 変換	
31, ...	17.5	位置決め中にスピンドルを再同期化 2	
31, ...	17.4	位置決め中にスピンドルを再同期化 1	
31, ...	18.7	回転方向セットポイント左	
31, ...	18.6	回転方向セットポイント右	
31, ...	18.5	発振速度	
31, ...	18.4	PLC を介した発振	
31, ...	19.7-19.0	スピンドルオフセット H - A	1.18（V1）
31, ...	60.7	正確停止（微）で到達した位置	1.3（B1）
31, ...	60.6	正確停止（粗）で到達した位置	1.3（B1）
31, ...	60.5	基準化／同期化 2	1.16（R1）

DB 番号	ビット，バイト	名称	参照
31, ...	60.4	基準化／同期化 1	1.16（R1）
31, ...	60.3	エンコーダリミット周波数オーバ 2	1.2（A3）
31, ...	60.2	エンコーダリミット周波数オーバ 1	1.2（A3）
31, ...	60.0	軸／スピンドルではない	
31, ...	61.7	パワーコントローラアクティブ	1.1（A2）
31, ...	61.6	速度コントローラアクティブ	1.1（A2）
31, ...	61.5	位置コントローラアクティブ	1.1（A2）
31, ...	61.4	静止軸／スピンドル (n < n min)	1.1（A2）
31, ...	82.3	ギアチェンジ	
31, ...	82.2-82.0	ギアステップ設定 A-C	
31, ...	83.7	実際の回転方向右	
31, ...	83.5	セットポイント範囲のスピンドル	
31, ...	83.2	セットポイント速度増加	
31, ...	83.1	制限されたセットポイント速度	
31, ...	83.0	速度制限オーバ	
31, ...	84.7	アクティブなスピンドル制御モード	
31, ...	84.6	アクティブなスピンドル発振モード	
31, ...	84.5	アクティブなスピンドル位置決めモード	
31, ...	84.3	補正チャックを使用したタッピングアクティブ	
31, ...	86 および 87	スピンドルの M 機能	
31, ...	88-91	スピンドルの S 機能	

■ マシンデータ

番号	識別子	名称	参照
一般事項 ($MN_ ...)			
12060	OVR_SPIND_IS_GRAY_CODE	グレイコーディングを使用したスピンドルオーバライド	1.18（V1）
12070	OVR_FACTOR_SPIND_SPEED	スピンドル速度オーバライドスイッチの評価	1.18（V1）
12080	OVR_REFERENCE_IS_PROG_FEED	オーバライド参照速度	1.18（V1）
チャンネル別 ($machine code_ ...)			
20090	SPIND_DEF_MASTER_SPIND	チャンネルのマスタスピンドルの初期設定	
20092	SPIND_ASSIGN_TAB_ENABLE	スピンドルコンバータのイネーブル／ディスエーブル	
20118	GEOAX_CHANGE_RESET	自動ジオメトリ軸変更許可	
22400	S_VALUES_ACTIVE_AFTER_RESET	アクティブなリセットを介した S 機能	
軸別 ($MA_ ...)			
30300	IS_ROT_AX	回転軸	2.12（R2）
30310	ROT_IS_MODULO	モジュロ変換	2.12（R2）
30320	DISPLAY_IS_MODULO	位置表示	2.12（R2）
31050	DRIVE_AX_RATIO_DENOM	分母ロードギアボックス	1.8（G2）
31060	DRIVE_AX_RATIO_NUMERA	分子ロードギアボックス	1.8（G2）
31122	BERO_DELAY_TIME_PLUS	正方向での BERO 遅延時間	
31123	BERO_DELAY_TIME_MINUS	負方向での BERO 遅延時間	
32200	POSCTRL_GAIN	サーボゲイン係数	1.8（G2）
32800	EQUIV_CURRCTRL_TIME	フィードフォワード制御の等価時定数電流制御回路 K3	2.5（K3）
32810	EQUIV_SPEEDCTRL_TIME	フィードフォワード制御の速度制御回路の定等価時間	2.5（K3）
32910	DYN_MATCH_TIME	動的マッチングの定時間	1.8（G2）
34040	REFP_VELO_SEARCH_MARKER	参照ポイントクリープ速度	1.16（R1）
34060	REFP_MAX_MARKER_DIST	ゼロマーク距離の監視	1.16（R1）
34080	REFP_MOVE_DIST	距離コード化系の参照ポイント距離／移動先	1.16（R1）
34090	REFP_MOVE_DIST_CORR	参照ポイントオフセット／絶対オフセット，距離コード化	1.16（R1）
34100	REFP_SET_POS	参照ポイント値	1.16（R1）
34200	ENC_REFP_MODE	基準化モード	1.16（R1）
35000	SPIND_ASSIGN_TO_MACHAX	スピンドルをマシン軸に割当てる	
35010	GEAR_STEP_CHANGE_ENABLE	ギアチェンジが可能	
35020	SPIND_DEFAULT_MODE	基本スピンドル設定	
35030	SPIND_DEFAULT_ACT_MASK	初期スピンドル設定を起動	
35040	SPIND_ACTIVE_AFTER_RESET	リセット後にアクティブなスピンドル	
35100	SPIND_VELO_LIMIT	最高スピンドル速度	
35110	GEAR_STEP_MAX_VELO[n]	ギアチェンジの最高速度	
35120	GEAR_STEP_MIN_VELO[n]	ギアチェンジの最低速度	
35130	GEAR_STEP_MAX_VELO_LIMIT[n]	ギアステップの最高速度	

番号	識別子	名称	参照
35140	GEAR_STEP_MIN_VELO_LIMIT[n]	ギアステップの最低速度	
35150	SPIND_DES_VELO_TOL	スピンドル速度許容値	
35160	SPIND_EXTERN_VELO_LIMIT	PLC からのスピンドル速度制限	
35200	GEAR_STEP_SPEEDCTRL_ACCEL[n]	スピード制御モードでの加速	
35210	GEAR_STEP_POSCTRL_ACCEL[n]	位置制御モードでの加速	
35220	ACCEL_REDUCTION_SPEED_POINT	減少した加速の速度制限	
35230	ACCEL_REDUCTION_FACTOR	減少した加速	
35300	SPIND_POSCTRL_VELO	位置制御起動速度	
35350	SPIND_POSITIONING_DIR	非同期スピンドルの回転方向の位置決め	
35400	SPIND_OSCILL_DES_VELO	発振速度	
35410	SPIND_OSCILL_ACCEL	発振中の加速	
35430	SPIND_OSCILL_START_DIR	発振開始方向	
35440	SPIND_OSCILL_TIME_CW	M3 方向での発振時間	
35450	SPIND_OSCILL_TIME_CCW	M4 方向での発振時間	
35500	SPIND_ON_SPEED_AT_IPO_START	セットポイント範囲のスピンドルのフィードイネーブル	
35510	SPIND_STOPPED_AT_IPO_START	静止軸を伴うフィードイネーブル	
35590	PARAMSET_CHANGE_ENABLE	PLC から可能なパラメータセット定義	1.1（A2）
36060	STANDSTILL_VELO_TOL	スレッショルド速度 「静止軸／スピンドル」	1.2（A3）
36200	AX_VELO_LIMIT	速度のスレッショルド監視	1.2（A3）

■ 設定データ

番号	識別子	名称	参照
スピンドル別 ($SS_ ...)			
42600	JOG_FEED_PER_REF_SOURCE	JOG モードでの回転フィードレート制御	1.18（V1）
42800	SPIND_ASSIGN_TAB	スピンドル番号コンバータ	
42900	MIRROR_TOOL_LENGTH	ミラーツール長オフセット	1.19（W1）
42910	MIRROR_TOOL_WEAR	ツール長補正のミラー磨耗値	1.19（W1）
42920	WEAR_SIGN_CUTPOS	加工面のミラー磨耗値	1.19（W1）
42930	WEAR_SIGN	すべての磨耗値の変換サイン	1.19（W1）
42940	TOOL_LENGTH_CONST	加工面変更時のツール長補正の割当てを保持 (G17 ～ G19)	1.19（W1）
43210	SPIND_MIN_VELO_G25	プログラムされたスピンドル速度制限 G25	
43220	SPIND_MAX_VELO_G26	プログラムされたスピンドル速度制限 G26	
43230	SPIND_MAX_VELO_LIMS	G96 を伴うプログラムされたスピンドル速度制限	
43300	ASSIGN_FEED_PER_REF_SOURCE	位置決め軸／スピンドルの回転フィードレート	1.18（V1），1.10（P2）

■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

1.18 送り機能 (V1)

1.18.1 概略説明

送りのタイプ

送りはマシン速度（軸あるいは軌跡速度）を決定します。送りは，輪郭上あるいはツール中心点パス上（G コマンドによる）でツールオフセット公差が設けられている場合でも，すべてのタイプの補間に適用されます。

次に示すタイプの送りを使用することで，様々なアプリケーション（旋盤加工，フライス加工，穴あけ加工など）に対応できます。

- 早送り (G0)
- 逆時間指定送り (G93)
- 毎分送り (G94)
- 毎回転送り (G95)
- 周速一定送り (G96)
- 定速 (G97)
- ねじ切り (G33)
- 補正チャックを使用したタッピング (G63)
- ソリッドタッピング (G331，G332)

プログラム可能なランイン，ランアウトパス

ねじ切りのランインパスとランアウトパスがプログラムできます。ねじ切りが定義された距離内で加減速されます。

（注）非常に短い距離が指定された場合は，軸が過負荷になることがあります。

送りの軸割当て

技術要件に合わせて，送りは軸に対して可変的に割当てることができます。次の内容がサポートされています。

- 作業平面用とインフィード軸用に分けた送り
- 送りの可変軸割当て
- 位置決め軸用の送り

送り制御

加工中あるいは試験目的で，変化する要件にシステムを最適化するために，プログラムされた送り速度をマシンの制御パネル，操作パネル，PLC で変更したり，プログラムコマンドを使用して変更したりできます。

補間送り

軌跡が定義できるように，DIN 66025 に送り速度プログラミングが拡張されているので，直線および立体パスも定義できます。立体パスは直接プログラムすることもスプライン補間としてプログラムすることもできます。

次の指令をプログラムすることができます。

- FNORM ：DIN 66025 に従った応答（初期設定）
  ブロック中にプログラムされた F 値は，パス全体にわたる定数として指定され，その後はモーダル値として適用されます。
- FLIN ：ブロック中にプログラムされた F 値は，ブロックの始点での現在値からブロック終点に向けて直線的にパス距離にわたって位置決めされ，その後はモーダル値として適用されます。
- FCUB ：ブロック中にプログラムされた F 値でブロックとブロックをスプラインによって接続します。スプラインは前の送りに対する接線から開始し，次の送りに対する接線で終了します。ブロック中に F アドレスが無い場合は，最後にプログラムされた値が代わりに使用されます。
- FPO ：F アドレス [構文 : F=FPO (...,...,...)] が，それがプログラムされているブロックの終点に向けて現在値から複数の送り速度を規定します。終わりの値がモーダル値として適用されます。

FLIN と FCUB が圧縮 COMPON と一緒に使用されていれば，MD 20172: $MC_COMPRESS_VELO_TOL を使用することで，送りについて公差を定義できます。

送り速度のプログラミングの詳細についてはプログラミング説明を参照してください。

1.18.2 詳細説明

■ 送り速度 F

送り速度 F

送り速度は，関連するすべての軸の速度成分の幾何学的合計です。このため，送り速度は補間軸の個々のモーションからつくり出されます。

標準設定はプログラムされているジオメトリ軸の軸速度をそのまま使用します。"FGROUP" コマンドを使用することによって，送り速度の計算において他のジオメトリ軸および（または）同期軸を含めることができます。

送り速度 F は加工速度を決定します。送り速度 F は，ツールオフセットについて公差が設定されている場合でも，すべてのタイプの補間に適用されます。アドレス F の下でプログラムされた値は，別の F 値あるいは別のタイプの送り速度がプログラムされるまではプログラム中に留まります。

送り速度 F の値の範囲

詳しくはプログラミングマニュアルを参照してください。

PLC インタフェースでの F 値

現在の送り速度の F 値は，補助ファンクション (DB21, ... DBB158 to 193) について必ずチャンネル別 PLC インタフェース中で入力されます。

関連するインタフェース信号（変更信号，F 値）については次を参照してください。

参照： 1.9 補助機能出力 (H2)

遷移円付きの送り

詳しくはプログラミングマニュアルを参照してください。

パスの凹凸部での送り速度

円形ブロックあるいは円形スプラインブロックおよび起動している工具径補正（G41/G42）については，プログラムされた送り速度は中心点パスあるいは輪郭（凹部か凸部かによって）に作用します。このために G コマンドが提供されています。

- CFTCP: プログラムされた送りは中心点パスに作用します。
- CFC: プログラムされた送りは輪郭に作用します。
- CFCIN: プログラムされた送りは凹部スプラインの輪郭にのみ作用します。

最大送り速度

最大送り速度は，関連する直線または回転軸 (MD 32000: MAX_AX_VELO) の最大速度から得られます。つまり，最大速度が最も低い軸によって最大送り速度が決まります。この最大送り速度でクランプされます。

G0 がプログラムされている場合は，MAX_AX_VELO で制限されたパス速度で移動が行われます。

パス軸の制限速度

特定の軸速度を制限するためのコマンド FL[x]=... を使用することで，パス軸（ジオメトリ軸および同期軸）について制限速度をプログラムすることができます。

このため，作業平面とインフィード軸について別の送りをプログラムすることができます。このことは，パスに関連する補間用とインフィード軸用の両方について送り速度を指定できることを意味しています。インフィード軸は，選択された加工平面に対して垂直な軸として指定します。軸速度（パス速度）を制限するためにインフィード軸のフィードをプログラムすることができます。フレームを通しての座標回転は含められません。つまり，インフィード軸は標準座標の軸でなければなりません。このファンクションは，カッタの切削性能はフェース側の方がカッタ円周を横切る方向より劣るということに対する対策となります。

プログラミング例：

. .. .. G94	送りタイプの選択 (mm/min)
X30 Y20 F200	送り速度 = 200 mm/s
FL[Z]=50 Z-30	Z 軸の最大送り速度 : 50 mm/s

■ 送りタイプ G93，G94，G95

適用範囲

送りタイプ G93，G94 および G95 は，自動モードではグループ 1 の G コード（G0 を除く）について有効になります。

JOG モードでの移動に G94 あるいは G95 を使用できます。

逆時間指定送り (G93)

ブロックの後退について，送り速度をプログラムするよりも時間をプログラムする方が簡単な場合に逆時間指定送り速度を使用します。

逆時間指定送り速度は次の式から求められます。

$$F = \frac{v}{s};\quad F\ [1/min]$$

F ： 逆時間指定送り速度

v ： mm/min あるいは inches/min で表した希望するパス速度

s ： mm あるはインチで表したパス長

プログラミング例：

N10 G1 G93 X100 Y200 F2; プログラムされたパスは 0.5 分で移動します。

（注）G41/G42 が有効なときは G93 は使えません。ブロックごとでブロック長が大きく異なる場合は，G93 用にブロック毎に別の F 値をプログラムしてください。

毎分送り (G94)

毎分送りは，直線軸あるいは回転軸について次の単位を使用してプログラムします。

- メートル系の場合は [mm/min，度 /min]
- インチ系の場合は [inch/min，度 /min]

毎回転送り (G95)

毎回転送りはマスタスピンドルについて次の単位を使用してプログラムします。

- メートル系の場合は [mm/rev]
- インチ系の場合は [inch/rev]
- 回転軸については [度 /rev]

送り速度は，次の式にしたがってスピンドルの実際速度から求めます。

V = n * F

V： 送り速度

n ： マスタスピンドルの速度

F ： プログラムされた回転送り

（注）プログラムされた F 値は，送りタイプ G93，G94 および G95 が切替わると削除されます。
JOG モードでは，軸／スピンドルの応答は，SD 41100: JOG_REV_IS_ACTIVE （JOG 用の回転送りレートがアクティブ）の設定データにも依存します。

- この設定データがアクティブであれば，軸／スピンドルは，マスタスピンドルによって，回転送りレート MD 32050: JOG_REV_VELO（JOG 用の回転送りレート）あるいは MD 32040: JOG_REV_VELO_RAPID（早送りオーバレイを使用する JOG 用の回転送りレート）で常に移動します。
- この設定データがアクティブでなければ，軸／スピンドルの応答は，設定データ SD 43300: ASSIGN_FEED_PER_REV_SOURCE（位置決め軸／スピンドルの回転送りレート）によります。
- この設定データがアクティブでなければ，回転付きのフレームがアクティブになっているジオメトリ軸の応答は，チャンネル別設定データ SD 42600: JOG_FEED_PER_REV_SOURCE（回転付きのフレームがアクティブになっているジオメトリ軸についての JOG モードでの回転送りレート）によります。

"Revolutional feedrate active"（毎回転送り指定）DB31, ... DBX62.2

このインタフェース信号を使用して，プログラムされた有効な毎回転送り (G95) が表示されます。

アラーム

- F 値がプログラムされていなければ，アラーム 10860 "no feedrate programmed" が出力されます。
  このアラームは G0 ブロックでは発生しません。
- 負の送り速度がプログラムされていれば，アラーム 14800 "programmed path velocity smaller than or equal to zero" が出力されます。
- 毎回転送り (G95) がプログラムされているにもかかわらずマスタスピンドルが定義されていない場合は，アラーム 10810 "no master spindle defined" が出力されます。

■ 送りタイプ G96，G97

周速一定速度 (G96)

ワークピースの作業直径とは関係なく切削条件を一定にするには，旋盤については周速一定速度を使用します。こうすることにより，ツールは最適切削性能範囲内で使用されることになり，ツール寿命が長くなります。

G96 の選択 :

G96 がプログラムされている場合は，S 値はフェーシング軸についての mm/min あるいは inch/min で表した切削速度であると解釈されます。

加工するにしたがってワークピースの直径が減少する場合は，周速が一定となるように回転速度が増加します。

G96 がパートプログラム中で最初に選択されるときは，周速一定速度を mm/min あるいは inch/min で入力しなければなりません。一旦このコマンドが選択されたら，別の切削速度を入力することができます。制御装置は自動的に毎回転送りに切替わります（G95 の場合と同様)。つまり，プログラムされた送り F は mm/rev あるいは inch/rev と解釈されます。

プログラムされた切削速度 (S_{G96}) とワークピースに関連するフェーシング軸の実際値（半径 r）によって，制御装置は次の式を用いてスピンドル速度を計算します。

$$n = \frac{S_{G96}}{2 * \pi * r}$$

π = 円周率

例 : S_{G96} = 230 m/min

- r = 0.2 m → n = 183.12 r/min

- r = 0.1 m → n = 366.24 r/min

⇒ ワークピースの直径が小さくなればなるほど，速度は速くなります。

ジオメトリ軸は G96 についてはフェーシング軸として定義しなければなりません。その位置がマスタスピンドルの速度に影響する平面軸は，チャンネル別の MD 20100: DIAMETER_AX_DEF （フェーシング軸ファンクションを有するジオメトリ軸）を介して定義されます。

ファンクション G96 では，フェーシング軸上のマシン原点とワークピースの原点がスピンドル回転の中心になければなりません。

フェーシング軸に作用する予め定義されたオフセットがスピンドル速度の計算で考慮されます。しかしツールオフセットとフレームは考慮されません。

（注）G96 はワークピースの加工時（送り F が有効の場合の G1, G2，G3，スプライン補間など）にしか有効になりません。
G96 と G0 ブロックについてのスピンドル速度の応答は，チャンネル別の MD 20750: ALLOW_G0_IN_96（G96 を有する G0 ロジック）で定義できます。
周速一定速度では切替えは行うことができません。
スピンドルオーバライドスイッチは計算されたスピンドル速度に作用します。フェーシング軸に対する DRF オフセットは，スピンドル速度セットポイント計算には影響しません。
スピンドルの作業範囲（G96 速度検出および送りオーバライド後）

G96 の速度制限

「周速一定速度」ファンクションの場合は，最大スピンドル速度は SD 43230: SPIND_MAX_VELO_LIMS（G96 を使用したスピンドル速度制限）で定義することができ，パートプログラム中では（マスタスピンドルの場合）プログラミングコマンド "LIMS" で定義することができます。最も新しく変更された値（LIMS あるいは SD）が有効です。

LIMS は G96 を使用した場合にのみ有効になります。

MD 10710: PROG_SD_RESET_SAVE_TAB[n]（更新される設定データ）によっては，制御装置の電源がオフにされても LIMS を使用した速度制限は保存されたままとなります。G96 が再び起動されると速度制限も起動されます。

G26 あるいは SD 43220: SPIND_MAX_VELO_G26（最大スピンドル速度）で定義された最大許容スピンドル速度でクランプされます。

速度制限（G26 あるいは SPIND_MAX_VELO_G26）を超えるような不正なプログラミングが行われると，"programmed speed too high" インタフェース信号 (DB31, ... DBB83.1) が出力されます。

直径が大きなパートでも滑らかに回転するように，スピンドル速度は最低レベル未満にならないようクランプされます。この速度は，SD 43210: SPIND_MIN_VELO_G25（最低スピンドル速度）および MD 35140: GEAR_STEP_MIN_VELO_LIMIT を使用して各ギアステップごとに定義されます。最低スピンドル速度は G25 を使用してパートプログラム中で変更できます。

設定データの効果の詳細については次を参照してください。

参照： 1.17 主軸機能 (S1)

様々なスピンドル速度制限を下図に示します。

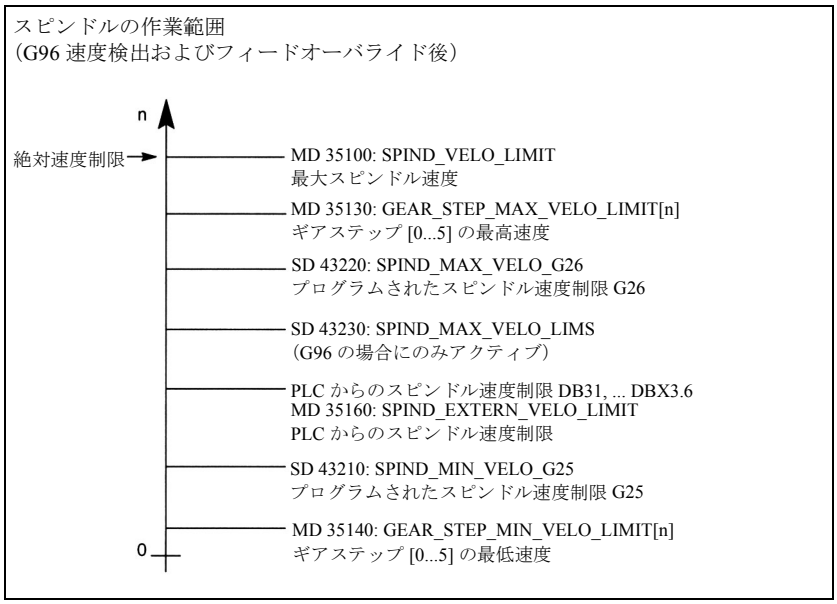


図 1.165 スピンドル速度制限

定速 G97

G97 は "constant cutting speed" インタフェース信号 (G96) を無効にし，計算された最後のスピンドル速度を保存します。送りは毎回転送り (G95) として解釈されます。

G97 が有効のときは，S 値を再プログラミングして新たにスピンドル速度を定義できます。G96 の下で定義された切削速度がこれによって影響されることはありません。

G97 を使用することで，加工が行われていないときのフェーシング軸上の動作が原因で発生する速度変化を防止することができます。

アラーム

- F 値がプログラムされていなければ，アラーム 10860 "no feedrate programmed" が出力されます。
  このアラームは G0 ブロックでは発生しません。
- 負の送り速度がプログラムされた場合は，アラーム 14800 "programmed path velocity smaller than or equal to zero" が出力されます。
- G96/G97 が有効であるにもかかわらず，その G96/G97 について MD 20100: DIAMETER_AX_DEF（フェーシング軸ファンクションを有するジオメトリ軸）でフェーシング軸が定義されていなければ，アラーム 10870 "no facing axis defined" が出力されます。
- G96 が有効なときに，LIMS プログラムコマンドを使用して負の最大スピンドル速度が定義されている場合は，アラーム 14820 "negative maximum spindle speed programmed for G96" が出力されます。
- G96 が初めて選択されたときに，周速一定速度がプログラムされなかった場合は，アラーム 10900 "no S value programmed for constant cutting speed" が出力されます。

■ G33 での送り

アプリケーション

ファンクション G33 を使えばピッチが一定のねじ山を加工することができます。

速度 S，送り F，ピッチ

G33 ねじ切りには毎回転送り [mm/revolution] が使用されます。毎回転送りは，ねじピッチ [mm/revolution] をプログラムすることによって定義します。ねじ方向の軸速度はプログラムされたスピンドル速度 S とねじピッチから次のように計算されます。

送り F [mm/min] = 速度 S [rev/min] * ねじピッチ [mm/rev]

加速の終点で，スピンドルの実際値（マスタスピンドル上の SPCON を使用したスピンドルセットポイント）と軸セットポイントとの間に位置カップリングが確立されます。この時点では，スピンドルのゼロマーク（ゼロマークオフセットを含む）を基準にした軸位置は，あたかもねじ開始位置（ゼロマークプラス SF）が超えられたときにブロックの開始で軸が急激に加速した状態となります。次の軸エラーについて補正が行われます。

最低スピンドル速度

低速でも滑らかな回転となるように，スピンドル速度は最低レベル未満とならないように監視されます。この速度は，各ギアステップについて，SD 43210: SPIND_MIN_VELO_G25（最低スピンドル速度）および MD 35140: GEAR_STEP_MIN_VELO_LIMIT で定義されます。最低スピンドル速度は G25 でパートプログラム中で変更できます。

NC 停止，単一ブロック

NC 停止と単一ブロックは（ブロック境界においても），ねじ切りチェーニングが完了してからでないと有効にはなりません。その後のすべての G33 ブロックと最初の G33 以外のブロックは，あたかも単一ブロックのように移動します。

終点に達する前にねじ切りを安全に中断できるよう改善されています。このことは後退モーションを起動することによって可能となります。

アラーム

- プログラミングが不正であると次のアラームが出力されます。
  アラーム 16005　"Impermissible path for retraction"（後退パス不可）
  アラーム 16710　"Master spindle not programmed"（マスタスピンドルがプログラムされていない）
  アラーム 16720　"Thread lead is zero"（ねじピッチがゼロ）
  アラーム 16730　"Incorrect parameter"（パラメータが不正）
  アラーム 16740　"No geometry axis programmed"（ジオメトリ軸がプログラムされていない）
- 軸別の「フィードホールド」などによって，G33 が有効なときにパス軸が停止すると，アラーム 22250 "axis stop with thread cutting"（ねじ切り中に軸停止）が出力されます。
- スピンドルオーバライドが 200% にセットされているなどして，G33 中がアクティブなときにスピンドル速度が高すぎる場合は，アラーム 22270 "spindle speed too high for thread cutting"（ねじ切り用のスピンドル速度が高すぎる）が出力されます。

プログラム可能なランインおよびランアウトパス

ランインおよびランアウトパスが短すぎて送り速度を調整する必要がある場合，パートプログラムインストラクション DITS（ねじ開始ずれ）および DITE（ねじ終了ずれ）を使用してパスランプの加減速を定義することができます。

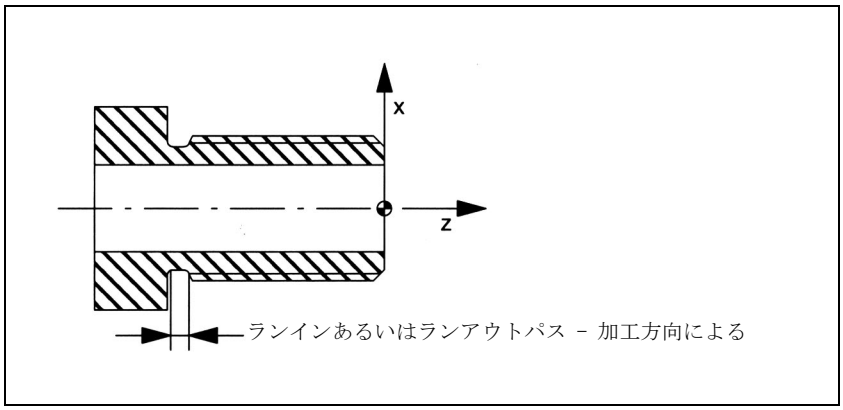


図 1.166 ランインあるいはランアウトパスが短すぎる

- ランインパスが短すぎる場合
  ねじ切りの始まりでのリップのために，ツール開始ランプのためのスペースは少ししかありません。このため，DITS を使用して開始ランプを短くしなければなりません。
- ランアウトパスが短すぎる場合
  スレッドの始まりでのリップのために，ツール開始ランプのためのスペースは少ししかありません。このため，ワークピースとツールエッジが衝突する可能性があり，DITE を使用してツールランプを短くしなければなりません（ただし衝突の可能性は依然として残ります）。
  対策：短いねじをプログラムする，スピンドル速度を下げる。

DITS と DITE でプログラムできるのはパスであって位置ではありません。このパートプログラムインストラクションは，設定データ THREAD_RAMP_DISP[0,1] に対応しています。

ランインおよび（または）ランアウトパスが非常に短い場合は，構成で許可されているよりも高いレートでねじ切り軸が加速します。この場合，その加速によって軸が過負荷になります。

アラーム 22280 "Programmed run-in path too short" がそのねじ切りランインについて出力されます（ MD 11411: ENABLE_ALARM_MASK で構成されている場合）。このアラームは情報用であり，パートプログラムの実行には何ら影響しません。

（注）DITE はスレッドの終わりでは丸め公差として作用します。これによって軸動作が滑らかになります。

適合性

MD 20650 ：THREAD_START_IS_HARD が省略されて，代わりに SD 42010: THREAD_RAMP_DISP[0] あるいは SD 42010 ：THREAD_RAMP_DISP[1] が使用されます。

SD 42010 ：THREAD_RAMP_DISP[0] = 0 および SD 42010: THREAD_RAMP_DISP[1] = 0 を使用した新しい設定データの応答は，以前の応答 MD 20650 ：THREAD_START_IS_HARD = 1 と同じです。

SD 42010 ：THREAD_RAMP_DISP[0] = -1 および SD 42010 ： THREAD_RAMP_DISP[1] = -1 を使用した新しい設定データの応答は，以前の応答 MD 20650 ：THREAD_START_IS_HARD = 0 と同じです。

補足条件

コマンド DITS および（または）DITE が入っているブロックが補間器にロードされると，DITS にプログラムされているパスが SD 42010 THREAD_RAMP_DISP[0] にコピーされ，DITE にプログラムされているパスが SD 42010 THREAD_RAMP_DISP[1] にコピーされます。

プログラムされたランインパスは，現在の設定（インチ系，メートル系）にしたがって処理されます。

ランイン／減速パスが最初のねじブロックの前あるいはその途中でプログラムされていない場合は，そのパスは SD 42010 THREAD_RAMP_DISP[0,1] の現在の内容から決定されます。

起動

このファンクションはねじ切り中は常に有効です。

例 :

N...

N59 G90 G0 Z100 X10 SOFT M3 S500

N60 G33 Z50 K5 SF=180 DITS=1 DITE=3 ;Rounding begins at Z=53

N61 G0 X20

N...

ねじ切りのための停止

（注）非破壊中断ファンクションはねじ切りにのみ使用して，G33 でのタッピングには使用すべきではありません。

後退動作

ねじ切り用の後退動作は次のキーワードで制御されます。

- LFON ⇒ねじ切りについて Liftfast を有効
- LFOF ⇒ねじ切り切削について Liftfast を無効

これらの G ファンクションは常にプログラム可能です。NC リセットおよび（または）NC スタートについての初期設定は MD 20150: GCODE_RESET_VALUES に保存されます。

次のタイミングでねじ切り中に後退動作が始まります。

- 高速入力
  （"LIFTFAST" オプション用に SETINT LIFTFAST を使用したプログラミング）
- NC 停止
- 暗示的に NC 停止を招くアラーム

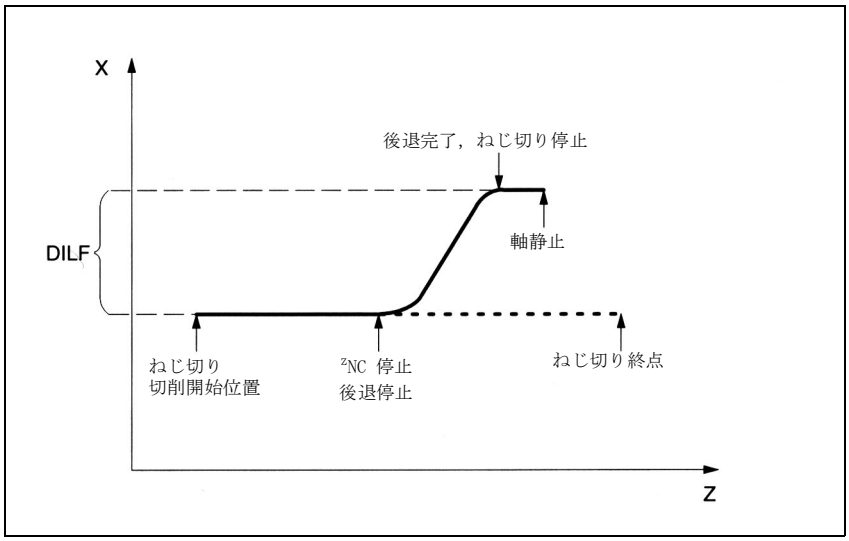


図 1.167 後退動作を通しての G33 の中断

後退パス

後退パスは MD 21200: LIFTFAST_DIST で構成します。必要であれば，パートプログラム中のこのパスを任意の点で DILF を書込むことで変更できます。

NC リセット後，MD 21200: LIFTFAST_DIST に入力された値は常に有効になります（デフォルト値）

後退方向

後退方向の決定方法は，変数 ALF と組合わせることによって次のキーワードで制御できます。

- LFTXT
  後退動作平面はパス接線とツール方向で決定されます。この G コード（初期設定）を使用すると高速リフト時の応答をプログラムできます。
  ALF プログラミングの詳細についてはプログラミングマニュアルを参照してください。
- LFWP
  後退動作平面は G コード G17，G18 あるいは G19 で選択された有効な作業平面となります。後退平面の方向はパス接線とは独立しています。このため，軸に平行な高速リフトをプログラムできます。

これらの G ファンクションは常にプログラム可能です。NC リセットおよび（または）NC スタートについての初期設定は MD 20150: GCODE_RESET_VALUES に保存されます。

後退動作平面中では，従来どおり ALF を使用して 45 度単位で方向をプログラムできます。LFTXT の場合は，ツール方向での後退は ALF=1 について定義されていました。

LFWP では，作業平面中の方向は次の割当てによって得られます。

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

（注）後退動作平面プログラミング用の拡張はねじ切りとは独立して使用できます。

後退速度

後退速度は，最高軸速度で行ないます。これは，MD (MD 32000: MAX_AX_VELO) で構成できます。

後退加速，ジャーク

加速は，最大許容値で行ないます。これは，MD (MD 32300: MAX_AX_ACCEL) で構成できます。

例

```
N55 M3 S500 G90 G18                    ; 有効な加工面を設定する
...
N65 MSG ("thread cutting")
MM_THREAD:
N67 $AC_LIFTFAST=0                     ; ねじ切り開始前にリセットする
N68 G0 Z5
N69 X10
N70 G33 Z30 K5 LFON DILF=10 LFWP ALF=7
                                       ; ねじ切り用高速リフトを有効にする
                                       ; 後退距離 =10 mm
                                       ; 後退面 Z/X (G18 のため )
                                       ; 後退方向 -X (with ALF=3
                                       ; 後退方向 +X)
N71 G33 Z55 X15
N72 G1                                 ; ねじ切りを選択する
N69 IF $AC_LIFTFAST GOTOB MM_THREAD
                                       ; ねじ切りが中断された場合
N90 MSG ("")
...
N70 M30
N55 M3 S500 G90 G0 X0 Z0
...
N87 MSG (" タッピング ")
N88 LFOF                               ; タッピング前に高速後退
N89 CYCLE...                           ; G33 でタッピングサイクルを無効にする
N90 MSG("")
...
N99 M30
```

制御応答

電源オンおよびリセット中は，該当する G グループの MD 20150: GCODE_RESET_VALUES から，構成パス (MD) と LFON あるいは LFOF および LFTXT あるいは LFWP のステータスで，後退パスが設定されます。

■ G63 用フィード (補正チャックを使用したタッピング)

アプリケーション

G63 は，補正チャックを使用したタップで，ねじをタッピングするためのサブファンクションです。

位置エンコーダは必要ありません。

速度 S, 送り F, ねじリード

送り F [mm/min] = 速度 S [rev/min] * ねじリード [mm/rev]

アラーム

- プログラムが不正なときには，次のアラームが出力されます :
  Alarm 16700 "Incorrect feed type"
  Alarm 16710 "Master spindle not programmed"
- G63 がアクティブのときにパス軸が停止する場合，たとえば，軸別 「フィードホールド」から，アラーム 22200 "axis stop with tapping" が出力されます。

■ G331, G332 用フィード (ソリッドタッピング)

アプリケーション

G331 (タッピング) および G332 (タッピング後退) は，スピンドルが位置制御モードで技術的に運転可能である場合，補正チャックなしで，ねじをタップするのに使用できます。

速度 S, 送り F, ねじリード

G331 および G332 では，毎回転送りが有効です [mm/rev]。毎回転送り速度は，ねじリード増加量をプログラミングして定義します [mm/revolution]。

ねじ方向の軸速度は，プログラムされたスピンドル速度 S およびねじリードから計算します :

送り F [mm/min] = 速度 S [rev/min] * ねじリード [mm/rev]

アラーム

プログラムが不正のときには，次のアラームが出力されます :

Alarm 16700 "Incorrect feed type"

Alarm 16720 "Thread lead is zero"

Alarm 16760 "S value missing"

■ 位置決め軸 FA 用送り

位置決め軸用送り FA

軸別送り FA が，位置決め軸の速度を定義します。

パートプログラムブロックでは，軸ごとに，1 つの送りが定義できます。ブロックごとに最大 5 つの送りが定義できます。

定義された軸の FA アドレスにプログラムされた値は，新しい FA 値 がプログラムされるまで，そのプログラムで有効のままです。

送り FA がプログラムされてない場合，軸は，MD 32060: POS_AX_VELO (軸速度を位置決めするリセット値) に定義された値で移動します。

プログラムエンド時，あるいはリセット時，FA 値は MD: POS_AX_VELO にプログラムされている最新の値あるいは入力された値にリセットされます。これは，MD 22410: F_VALUES_ACTIVE_AFTER_RESET (RESET を介して有効な F ファンクション) における設定によって決まります。

MD 32000: MAX_AX_VELO に定義された最高軸速度は超えることができません。

位置決め送り FA 値範囲

PLC インタフェースに対する F 値

現行ブロックの位置決め軸のための F 値は，補助ファンクション (DB21, ... DBB158 ～ 193) の PLC インタフェースに入力されます。

チャンネル別インタフェース信号 (変更信号 , 位置決め軸のマシン軸番号 , F 値):

位置決め軸のための F 値は，軸別 PLC インタフェース (DB31, ... DBB78 - 81) にも入力されます。

この F 値は，新しい値で上書されるまで保留されます。

アプリケーション：

プログラムされた F 値の PLC ユーザプログラムによる監視および 制限値を超えるときに PLC から軸別フィードオーバライドを選択する

（注）できれば，PLC に F ファンクションの出力をしないようにすることをお薦めします。輪郭モードの減速につながる可能性があるからです。
パラメータは，MD 22240: AUXFU_F_SYNC_TYPE (F ファンクションの出力時間) に定義されます。

■ 位置決めの送りタイプ - G94

毎分送り (G94)

位置決め軸送りは，G94 送りタイプで常に移動します。

毎分送りは，直線軸または回転軸に関して次の単位でプログラムされます :

- メートル系 [mm/min, 度 /min]
- インチ系 [inch/min, 度 /min]

アラーム

負の軸速度がプログラムされるとき , アラーム 14810 "negative axis velocity programmed for positioning axis" が出力されます。

■ 送り速度プログラミングの補足条件

測定単位

送り速度に使用される測定単位は，MD 12240: SCALING_SYSTEM_IS_METRIC (制御装置における，メートル系 / インチ系) に入力された値および MD 30300: IS_ROT_AX に入力された軸タイプ (回転軸あるいは直線軸) で決まります。

送りタイプの標準設定

G94 は，標準設定としてスクリーンに表示されます。送りタイプのリセット値 (標準プログラム設定) は，パートプログラムの開始時にのみ表示されます。リセット値は，MD 20150: GCODE_RESET_VALUES で設定されます。

F 値の適用範囲

MD 22140: F_VALUES_ACTIVE_AFTER_RESET は，最も新しくプログラムされた F 値をリセット後有効にすべきかどうかを定義するのに使用できます。

- プログラムされた送り速度 : F=....
- プログラムされた送り速度用オーバライドチェンジ : OVR=...
- プログラムされた位置決め送り : FA=....
- プログラムされた位置決め送り用オーバライドチェンジ : OVRA[U]=....

構文についての詳しい情報は，プログラミングマニュアルを参照してください。

スピンドルの位置決め

G95, G96, G97 or G33 が有効のときは，スピンドルの位置決めは起こりません。スピンドルの位置決め後に招かれた送り速度 = 0 がすでに完了しているからです。

⇒× プログラムされた軸位置にまだ到達していない場合は，ブロックが終了できません。

■ 送り制御

送り速度プログラミングおよび制御

下図は，送り速度プログラミングと制御の可能性を示しています。

詳しくは，

- 送り速度プログラミング，1.18.2 「 ■送り速度 F」,「■位置決め軸 FA 用フィード」および「■送り速度プログラミングの捕捉条件」を参照のこと。
- マシン制御パネルでの送りオーバライド，1.18.2「■マシン制御パネルでの送りオーバライド」を参照のこと。
- プログラム可能な送りオーバライド，1.18.2「■プログラム可能な送りオーバライド」を参照のこと。
- ドライラン送りの起動，1.18.2「■ドライラン送り」を参照のこと。

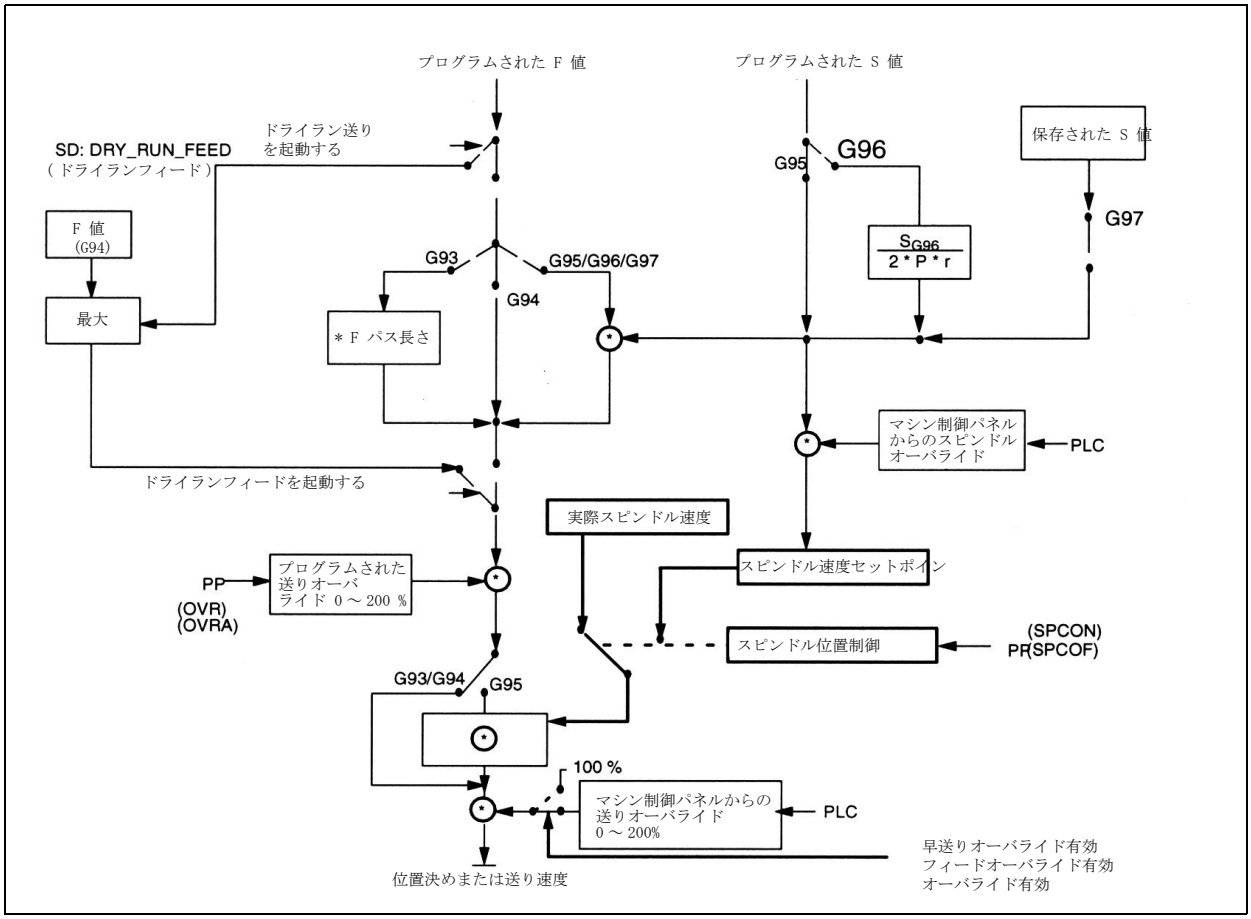


図 1.168 送りレートプログラミングおよび制御

■ フィードディスエーブルおよびフィード／スピンドル停止

一般事項

フィードディスエーブルあるいはフィード / スピンドル停止によって，ブレーキング特性およびパス輪郭に従って，軸が停止します。(例外 : G33 ブロック).

フィードディスエーブル

「フィードディスエーブル」というインタフェース信号 (DB21, ... DBX6.0) によって，チャンネルのすべての軸 (ジオメトリ軸および追加軸) が停止します。

下記のファンクションが有効チャンネル別フィードディスエーブルは，次のようになります。

- G33 ：無効
- G63 ：有効
- G331, G332 ：無効

「フィードホールド」DB21, ...DBX12.3 ジオメトリ軸

ジオメトリ軸 1,2 および 3 のための「フィードホールド」インタフェース信号 (DB21, ... DBX12.3，次の信号) を使って，JOG モードで軸を停止します。

「フィードホールド」DB31, ...DBX4.3 軸別

軸別の 「フィードホールド」インタフェース信号 (DB31, ... DBX4.3) を使って，マシン軸を停止します。

自動モード :

- 「フィードホールド」がパス軸について実行される場合，現行のブロックで移動するすべての軸および軸グループに関連しているすべての軸が停止します。
- 「フィードホールド」が位置決め軸について実行される場合，その軸だけが停止します。

現行の軸だけが，JOG モードで停止します。

下記のファンクションが有効のとき，軸別フィードディスエーブルは次のとおりです。

- G33 ：有効 (ダイナミック特性値によっては輪郭のずれが生じることがある)
- G63 ：有効
- G331, G332 ：無効

「スピンドル停止」DB31, ... DBX4.3

「スピンドルストップ」インタフェース信号 (DB31, ... DBX4.3) を使って，スピンドルを停止します。

次のファンクションがアクティブのとき，スピンドルストップは，次のとおりです。

- G33 ：有効
  (ダイナミック特性値によっては輪郭のずれが生じることがある)
- G63 ：有効
- G331, G332 ：無効

■ マシン制御パネルでの送りオーバライド

一般事項

オペレータは，送りオーバライドスイッチを使って，即時に効果的にプログラムされたパーセンテージに関連し送り速度を増減することができます。この送りには，オーバライド値を掛けます。

0 ～ 200% の間のオーバライドは，送り速度のためにプログラムできます。

早送りオーバライドスイッチを使って，パートプログラムのテスト時に移動速度を減速します。

0 ～ 100% の間のオーバライドは，早送りについてプログラムできます。

フィードは，位置決め軸について軸別に変更できます。オーバライドは，0 ～ 200% の間で変更可能です。

スピンドルオーバライドを使って，スピンドル速度と切削速度を変更できます (G96 で)。オーバライドは，0 ～ 200% の間で変更可能です。

オーバライドは，マシン別加速度および速度制限を超えたり，輪郭のエラーを発生してはなりません。

フィードオーバライドは，パス軸と位置決め軸について別々に変更できます。

オーバライドはプログラム値または制限値に作用します (たとえば，スピンドル速度については，G26, LIMS)。

チャンネル別 「送りオーバライド」DB21, ...DBB4
「早送りオーバライド」DB21, ...DBB5

各送りタイプのオーバライド係数について，1 つのイネーブル信号と 1 バイトの信号が，PLC インタフェースに用意されています。

「送りオーバライド」インタフェース信号 (DB21, ... DBB4)

「送りオーバライドが有効」(DB21, ... DBX6.7)

「早送りオーバライド」インタフェース信号 (DB21, ... DBB5)

「早送りオーバライド有効」インタフェース信号 (DB21, ... DBX6.6)

オーバライド用インタフェースは，バイナリコード化またはグレイコード化されたフォーマットで，PLC によって使用できます。

MD12020: OVR_FEED_IS_GRAY_CODE (グレイコード化された送りオーバライドスイッチ) および MD 12040: OVR_RAPID_IS_GRAY_CODE (バイナリコード化された早送りオーバライドスイッチ) を使って，バイナリコードとグレイコードのいずれが有効であるかを定義します。

次の恒久的割当てが，バイナリコーディングに適用できます :

コード	オーバライド係数
00000000	0.00 ＝ 0%
00000001	0.01 ＝ 1%
00000010	0.02 ＝ 2%
00000011	0.03 ＝ 3%
00000100	0.04 ＝ 4%
・	・
・	・
・	・
01100100	1.00 ＝ 100%
・	・
・	・
・	・
11001000	2.00 ＝ 200%

グレイコーディングの場合，スイッチ設定に該当するオーバライド係数が MD 12030: OVR_FACTOR_FEEDRATE [n](送りオーバライドスイッチの評価) あるいは MD 12050: OVR_FACTOR_RAPID_TRA [n](早送りオーバライドスイッチの評価) に入力されます。

有効な送りオーバライドは，現行チャンネルに割当てられたすべてのパス軸に作用します。

有効な早送りオーバライドは，現行の軸に割当てられた早送りで移動するすべての軸に作用します。

専用の早送りオーバライドスイッチを使用しない場合は，システムは早送りオーバライドと送りオーバライドとの切替えが可能です。この場合，100% を超える送りオーバライドは，100% の早送りオーバライドに制限されています。

有効であるオーバライドは，PLC または操作パネルから選択できます。早送りオーバライドを操作パネルから起動するとき，PLC 基本プログラムは 「早送り用送りオーバライドを選択する」インタフェース信号 (DB21, ... DBX25.3) を 「早送りオーバライドが有効」(DB21, ... DBX6.6) に送り，さらに，「送りオーバライド」インタフェース信号 (DB21, ... DBB4) を 「早送りオーバライド」インタフェース信号 (DB21,

... DBB5) にコピーします。

PLC からの起動時，PL C ユーザプログラムは，「早送りオーバライドが有効」インタフェース信号 (DB21, ... DBX6.6) をイネーブルし，送りオーバライド用インタフェース信号 (DB21, ... DBB4) を早送りオーバライドのためのインタフェース信号 (DB21, ... DBB5) にコピーする必要があります

次のファンクションが有効な時，チャンネル別送りと早送りオーバライドは，次のようになります :

- G33 ：無効
- G63 ：無効
- G331, G332 ：無効

送りオーバライドの基準速度

標準値からずれる送りオーバライドの基準速度を指定することができます。マシンデータ MD 12082:OVR_REFERENCE_IS_MIN_FEED は，この選択を行なうのに使用されます。

軸別 「送りオーバライド」DB31, ...DBB0

送りオーバライド係数についての 1 つのイネーブル信号と 1 バイト信号が，各位置決め軸の PLC インタフェースで指定できます。

「送りオーバライド」インタフェース信号 (DB31, ... DBB0)

「オーバライドを起動」インタフェース信号 (DB31, ... DBX1.7)

送りオーバライドのインタフェースは，バイナリコード化またはグレイコード化されたフォーマットで，PLC により使用することができます。

MD 12000: OVR_AX_IS_GRAY_CODE (グレイコード化された軸送りオーバライドスイッチ) を使って，バイナリコードとグレイコードのいずれが有効であるのかを定義します。

次の恒久的割当ては，バイナリコーディングに適用します :

コード	オーバライド係数
00000000	0.00 ＝ 0%
00000001	0.01 ＝ 1%
00000010	0.02 ＝ x 2%
00000011	0.03 ＝ 3%
00000100	0.04 ＝ 4%
・	・
・	・
・	・
01100100	1.00 ＝ 100%
・	・
・	・
・	・
11001000	2.00 ＝ 200%

グレイコーディングの場合，スイッチの設定に対応したオーバライド係数を，MD 12010: OVR_ FACTOR_AX_ SPEED [n](軸送りオーバライドスイッチの評価) に入力します。

次のファンクションが有効なとき，軸別送りオーバライドは次のとおりです :

- G33 ：無効
- G63 ：無効 (NC はオーバライドを恒久的に 100% にセットする)
- G331, G332 ：無効 (NC はオーバライドを恒久的に 100% にセットする)

「スピンドルオーバライド」DB31, ...DBB0

スピンドルオーバライド係数には，1 イネーブル信号および 1 バイト信号

「スピンドルオーバライド」インタフェース信号 (DB31, ... DBB19), SW 2, DBB0 まで

「オーバライドを起動」インタフェース信号 (DB31, ... DBX1.7)

「スピンドルオーバライド」のインタフェースは，バイナリコード化またはグレイコード化されたフォーマットで，PLC によって指定できます。

MD 12060: OVR_SPIND_IS_GRAY_CODE (スピンドルオフセットスイッチがグレイコード化される) を使って，バイナリコードとグレイコードのいずれが有効であるのかを定義します。

下記の恒久的割当ては，バイナリコーディングに適用されます :

コード	オーバライド係数
00000000	0.00 ＝ 0%
00000001	0.01 ＝ 1%
00000010	0.02 ＝ 2%
00000011	0.03 ＝ 3%
00000100	0.04 ＝ 4%
・	・
・	・
・	・
01100100	1.00 ＝ 100%
・	・
・	・
・	・
11001000	2.00 ＝ 200%

グレイコーディングの場合，スイッチ設定に対応するオーバライド係数は，MD 12070: OVR_FACTOR_SPIND_SPEED [n]
(スピンドルオーバライドスイッチの評価) に入力します。

次のファンクションが有効なとき，スピンドルオーバライドは次のとおりです :

- G33 ：有効
  (スピンドルが位置制御モードの場合，オーバライドスイッチが作動可能)
- G63 ：無効 (NC はオーバライドを恒久的に 100% にセットする)
- G331,G332 ：有効

オーバライド係数を制限する

バイナリコード化されたインタフェースでは，最大オーバライド係数は，MD 12100:OVR_FACTOR_LIMIT_BIN (バイナリコード化されたオーバライド係数の制限) を使って，送り，軸送りおよびスピンドル速度をさらに制限することができます。

詳しくは，1.18.4 「■一般マシンデータ」を参照してください。

オーバライド有効 DB21, ...DBX6.6 DB21, ... DBX6.7 DB31, ... DBX7.1

マシン制御パネルにある選択スイッチでセットされたオーバライド値は，「早送りオーバライドが有効」(DB21, ... DBX6.6) あるいは 「送りオーバライドが有効」(DB21, ... DBX6.7) あるいは 「オーバライドが有効」(DB31, ...DBX1.7) インタフェース信号がイネーブルの場合には，すべての操作モードとマシンファンクションについて即時有効になります。

0% というオーバライド係数は，送り禁止として機能します。

オーバライド無効

オーバライドが無効のとき (すなわち，上記インタフェース信号は "0" にセットされている)，オーバライド係数 "1" が NC で内部的に使用されます (すなわち，オーバライドは 100% である)。この場合，PLC インタフェースに入力された値は影響しません。

例外として，バイナリインタフェースはゼロが設定され，グレイコード化されたインタフェースには，1 番目のスイッチが設定されます。これらの場合には，PLC インタフェースに入力されたオーバライド係数が使用されます。バイナリインタフェースの場合，オーバライド係数は 0 です。グレイコード化されたインタフェースの場合，1 番目のスイッチ設定のためにマシンデータに入力された値が，オーバライド値として出力されます。これは，"0" で初期化される必要があります。

0% のオーバライド係数は，送り禁止として機能します。

スピンドルオーバライドのリファレンス

MD 12080: OVR_REFERENCE_IS_PROG_FEED (オーバライド基準速度) が，スピンドルオーバライドが下記のいずれを適用するのかを指定します。

- MD または SD によって制限される速度を適用，あるいは
- プログラムされた速度を適用

■ プログラム可能な送りオーバライド

ファンクション

プログラム可能な送りオーバライドを使って，パートプログラムのコマンドにより，パスと位置決め軸の速度レベルを変更できます。

分離した送りオーバライドは，位置決め軸についてプログラムできます。

プログラミング

送りオーバライドは，下記のコマンドで変更できます :

OVR= .... 送り速度 F に速度変更

OVRA[X1]= ... 位置決め軸速度 FA に速度変更

プログラム可能範囲は，0 から 200% までです。

標準設定 = 100%

適用範囲

「早送り／送りオーバライドが有効」(DB21, ... DBB6) および 「軸別オーバライドを起動」(DB31, ... DBX1.7) インタフェース信号は，プログラム可能な送りオーバライドには関連しません。プログラム可能な送りオーバライドは，これらの信号の停止時にも有効なままです。

有効なオーバライドは，プログラム可能な送りオーバライドとマシン制御パネルからの送りオーバライドとの積から計算します。

標準設定は 100% です。送りオーバライドがまだプログラムされていないとき，および，リセット後に，MD 22410:F_VALUES_ACTIVE_AFTER_RESET (リセットを介して，F ファンクション有効) がイネーブルでない場合は，この設定が適用されます。

OVR も，早送り (G0) の場合，有効です。ただし，自動的に，最大 100% に制限されます。

OVR は，G33 で有効にはなりません。

■ ドライラン送り

アプリケーション

たとえば，プログラムまたはプログラムセクションが送り速度を早くして実行できるようにするため，ワークピース加工なしにパートプログラムをテストするときに，ドライラン送りを使用します。

「ドライラン送りを選択」DB21, ... DBX24.6"

ドライラン送りは，PLC または操作パネルから起動できます。

操作パネルからの起動時，「ドライラン送りを選択」インタフェース信号 (DB21, ... DBX24.6) がイネーブルし，PLC 基本プログラムによって 「ドライラン送りを起動」インタフェース信号 (DB21, ... DBX0.6) に送ります。

PLC での選択時,「ドライラン送りを起動」インタフェース信号を，PLC ユーザプログラムでセットする必要があります。

プログラムの実行は，G94 でトリガされます。

ドライラン送りも，G93, G95 and G33 の速度より優先します。

プログラムされた速度は，ドライラン送りと比較され，移動は，2 つのフィードレート値のうち大きい方で実行されます。

ドライラン送りの変更

ドライラン送り (SD 42100: DRY_RUN_FEED) は,「パラメータ」ユーザエリアの操作パネルで変更できます。

プログラムが始動してしまうと変更はできません。

適用範囲

「ドライラン送りを起動」インタフェース信号がイネーブルの間は，SD: DRY_RUN_FEED に定義された送り値が，プログラムされた速度よりも大きい場合は，プログラムされた速度をオーバライドします。

ドライラン送りは，自動モードで選択でき，自動モードの中断時またはブロックのエンドで有効になります。

SW バージョン 4.2 以降

あらかじめ，ドライランフィードレートが自動モードを中断すれば，またはブロックのエンドでのみ，起動できます。MD 10704: DRYRUN_MASK = 1 がイネーブルの場合，ドライランフィードレートも，プログラムの実行中に起動できます (パートプログラムブロックにおいて)。

（注）加工中の起動によって，少しの間軸を停止させる制御において内部認識タスクがトリガされます。加工中のワークピースの表面仕上げに影響することがあります。

■ 一つのブロック中の複数の送り値

アプリケーション

このファンクションは，主に研削に使用します。

機能性

「一つのブロック中の複数の送り」というファンクションを使えば，NC ブロックの 6 種類の送り値，滞在時間，あるいは後退モーションを，外部のデジタルおよび／またはアナログの入力により同期して起動することができます。

後退

後退は，IPO サイクルで前に定義された量だけ，行なわれます。

信号

HW 入力信号は，「一つのブロック中の複数の送り」ファンクション用の 1 入力バイトにまとめられます。ファンクションの恒久的割当ては，このバイト内で適用されます :

表 1.74 「一つのブロック中の複数の送り」用入力バイト

	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
入力番号	E7	E6	E5	E4	E3	E2	E1	E0
フィードアドレス	F7	F6	F5	F4	F3	F2	ST	SR

E7 ~ E2　　送り F7 から F2 までを起動する

E1　　滞在時間 ST (秒) を起動する

E0　　後退モーション SR を起動する

優先順位

信号は，E0 から始まり，昇順で割付けられています。

したがって，後退モーション (SR) が最高の優先順位になり，送り F7 が最低になります。

SR および ST は，F2 ～ F7 で起動された送り動作を終わらせます。SR もまた，ST すなわち，完了ファンクションを終了させます。

最高の優先順位を持つ信号は，現行の送りを決定します。MD 21230:MULTFEED_STORE_MASK (このファンクションの入力信号には，「一つのブロック中の複数の送り」が保存されています) は，いずれの場合も (F2 ～ F7), 最高の優先順位を失ったときの応答を定義するのに使用できます 。

ブロック終了基準は下記の場合に満たされます :

- プログラムされたエンド位置に到達したとき
- 後退モーションが終わるとき (SR)
- 滞在時間が経過したとき (ST)

移動距離削除

後退モーションあるいは滞在時間によって，移動距離が削除されます。

ハードウエアの割当て

上記入力バイトは，チャンネル別 MD 21220: MULTFEED_ASSIGN_FASTIN (「一つのブロック内の複数の送り」に NCK I/Os の入力バイトを割当てる) を使って，最大 2 ディジタル入力バイトまたは NCK I/Os のコンパレータ入力バイトに割当てることができます。

MD 21220: MULTFEED_ASSIGN_FASTIN は，さらに入力ビットを変更するのに使用できます。

2 つ目のバイトを入力する場合，1 番目のバイトと 2 番目のバイトの内容が OR されてから，使用されます。

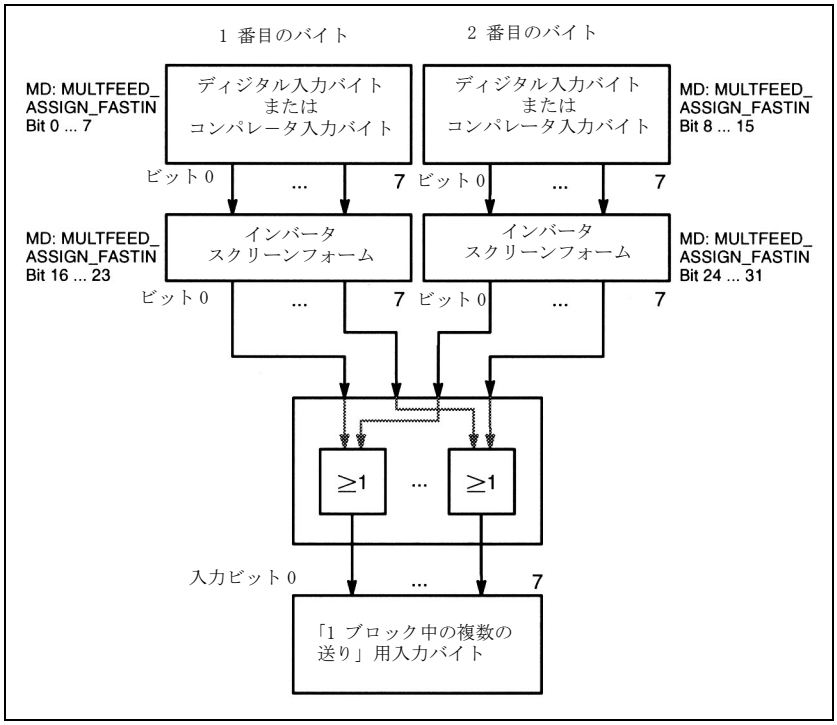


図 1.169 「1 ブロック中の複数の送り」のための信号ルーティング

パスモーションのプログラミング

送り速度は，アドレス F にプログラムされており，入力信号が表われるまで有効です。この値は，モーダルで機能します。

F2=... から F7=... までは，ブロック中に最大 6 つの他の送りをプログラムするために，送り速度に加えて使用できます。

数字の表記は，変更時に送りを起動する入力のビット番号を示しています。

つまり , F7=1000 ; 7 は入力ビット 7 を表します。

ビット 2 から 7 は，フィードの数字表記ができます。プログラム値は，モーダルではなく機能します。F でプログラムされた送り速度は，次のブロックに適用されます。

滞在 (スパークアウト時間) および後退パスは，ブロック内の別々のアドレス下でプログラムします :

ST=... 滞在 (切削スパークアウト時間)

SR=... 後退パス

これらのアドレスは，モーダルではありません。

軸モーションのプログラミング

軸送りは，アドレス FA の下でプログラムされており，入力信号が表われるまで有効です *。モーダルで有効です。

FMA[2,x]=... から FMA[7,x]=... までは，ブロック内の軸につき最大 6 つまでの送りをプログラムすることに使用できます。

四角い括弧の中の最初の表記は，変更時に送りを起動する入力のビット番号を示しています。2 番目の表記は，送りが適用される軸を示しています。

つまり，FMA[3,Y]=1000　;Y 軸用軸送り，

; 3 は，入力ビット 3 を示す

ビット 2 から 7 までは，軸送りの数字表記が可能です。FMA の下でプログラムされた値は，モーダルではなく有効です。FMA 下でプログラムされた送りは次のブロックに適用されます。

滞在 (スパークアウト時間) および後退パスは，また，単一軸について定義できます :

STA[x]=...　軸別滞在 (スパークアウト時間)

SRA[x]=...　軸別後退パス

括弧の中の表記は，スパークアウト時間と後退パスが適用される軸を示しています。

STA[X]=2.5 ; スパークアウト時間は 2.5 秒である

SRA[X]=3.5 ; 後退パスは 3.5 である (単位 例 : mm)

これらのアドレスは他には適用されません。

スパークアウト時間 (滞在) または後退パスが，外部入力のためにプログラムされている場合，この軸は，このブロックの PSOA 軸 (ブロックのエンドを超える位置決め軸) としてプログラムしてはなりません。

スパークアウト時間または後退パスの入力が起動されているとき，パス軸あるいは特定の単一軸の移動距離が削除され，滞在または後退が開始します。

（注）後退パスの単位は，現行で有効な測定単位を適用します (ミリメートルまたはインチ) 。
戻しは，常に，そのときの動きの反対方向に行なわれます。
SR/SA は，戻し量の値を常にプログラムします。記号はプログラムされません。
多様な軸の同期コマンドの入力状態を参照することも可能です。
一つのブロック中の複数の送りについては，ルックアヘッドも有効になります。このように，現行の送りがルックアヘッド値で制限されます。

アプリケーション

たとえば，代表的なアプリケーションを次に示します :

- アナログキャリパ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よって，多様な送り速度値，滞在，および後退パスが起動できます。限界値は設定データを介して定義されます。
- インフィードから作業フィードへの近接スイッチによる切換え

例

円錐リングの内部切削の場合は，実際直径はキャリパで決定され，制限によっては，荒削り，仕上げ，微仕上げに必要な送り値が起動されます。キャリパの位置もまた終点を指定します。このように，ブロック終了基準は，インフィード軸のプログラムされた軸位置によってだけでなく，キャリパによっても決まります。

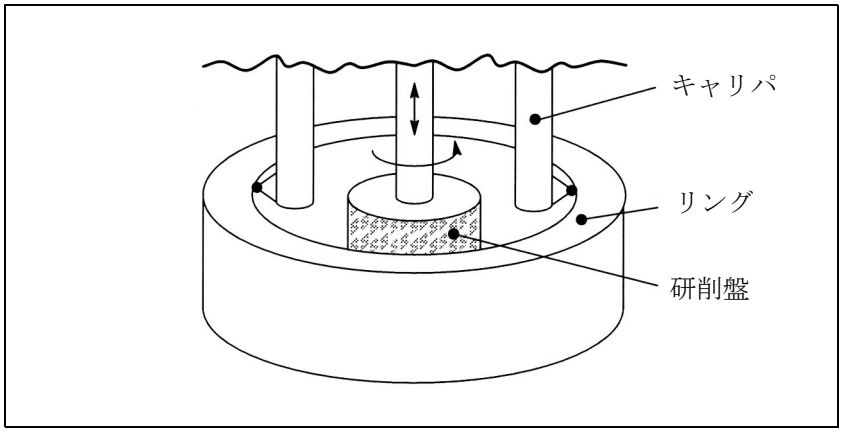
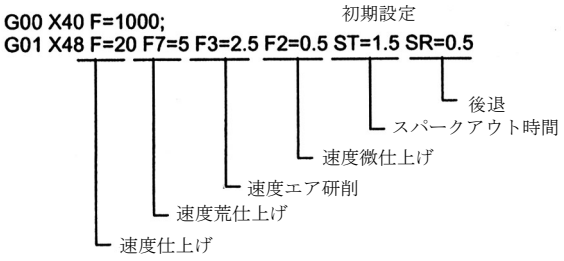


図 1.170 キャリパ

パートプログラム



（注）軸の速度／パス (F 値) は，100% 速度です。「一ブロック中の複数の送り値」インタフェース信号 (F2 から F7 までの値) は，軸の速度／軌跡速度以下である送りを得るために使用することができます。
「一ブロック中の複数送り」という機能性は，シンクロナイズドアクションに関連づけて利用できます。

■ 固定送り値

ファンクション

マシンデータは，4 固定送り値を定義するのに使用でき，インタフェース信号を介して，起動できます。ファンクションは，自動モードと JOG モードで使用できます。

自動モードでの応答

すでに選択されている固定送りは，プログラムされた速度の代わりに使用します。次の MDs とインタフェース信号を使って，パス／ジオメトリ軸について固定フィードを選択することができます :

MD 12202: $MN_PERMANENT_FEED[n] (直線軸の固定速度)

MD 12204: $MN_PERMANENT_ROT_AX_FEED[n] (回転軸の固定速度)

Fixed feed x for path/geometry axes"「パス／ジオメトリ軸用固定速度 x 」インタフェース信号
(DB21, 22, ... ,DBX29.0, 29.1, 29.2, 29.3)
プログラムされた速度を使用する代わりに，輪郭が起動された固定速度で移動します。

JOG モードでの応答

インタフェース信号を介して選択された固定速度は，セットされた JOG 速度の代わりに使用します。

移動方向は，インタフェース信号を使って指定されます。

固定速度は，次のインタフェース信号を使って，パス／ジオメトリ軸について選択できます。

MD 12202: $MN_PERMANENT_FEED[N](直線軸のための固定速度)

MD 12204: $MN_PERMANENT_ROT_AX_FEED[n](回転軸のための固定速度)

"Activate fixed feed x for machine axes"「マシン軸用固定速度 x を起動」インタフェース信号

(DB31, 32, ... ,DBX3.2, 3.3, 3.4, 3.5)

セットされた JOG/JOG 早送り速度 の代わりに，軸が起動された固定速度で移動します。

補足条件

- 固定速度は，スピンドル，位置決め軸およびタッピングについては有効になりません。
- 固定速度での移動時，オーバライドが MD 12200: $MN_RUN_OVERRIDE_0 (オーバライド 0 で移動する) によって決まります。
- 固定速度の選択時，DRF オフセットを起動することはできません。
- 固定速度は常に直線速度です。回転速度は，内部で，直線速度に変換されます。

マシンデータ

12200 MD 番号	$MN_RUN_OVERRIDE_0 オーバライド 0 で移動する	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 7/2	単位 : -
データタイプ : BOOLEAN		
意味 :	= 0 オーバライド 0 が有効であり，ブレーキ (従来のモード , 安全ファンクション) を意味する。ハンドルの使用時，ジオメトリと輪郭のハンドルについて，マシン軸には $MA_HANDWH_STOP_COND により，また，パルスが集められているかどうかを決定するためには $MC_HANDW_CHAN_STOP_COND を介して，ビット 0 と ビット 1 が指定される。 = 1 オーバライド 0 は，有効ではない。すなわち，ハンドルを使って，JOG モードで，固定速度により移動することも可能である。	
関連性	$MA_HANDWH_STOP_COND $MC_HANDW_CHAN_STOP_COND	

12202 MD 番号	$MN_PERMANENT_FEED[n] 直線送りのための固定速度	
初期設定 :	最小入力リミット :	最大入力リミット :
変更が有効になるための条件 : 電源オン	保護レベル : 7/2	単位 : mm/min, Inch/min
データタイプ : DOUBLE		
意味 :	自動 モードの場合 : 固定速度がインタフェース信号を介して起動されるとき，プログラムされた送り速度の代わりに固定速度が使用される。 JOG モードの場合 : 固定速度をインタフェース信号を介して起動し，直線軸用移動キーを使用するとき，移動は，選択された方向に固定速度により実行される。 n = 0, 1, 2, 3 は固定速度 1, 2, 3, 4 を意味する。	
例外，エラー	$MA_MAX_AX_VELO によって定義された最高速度が有効である。 オーバライドの効果は，$MN_RUN_OVERRIDE_0 によって決まる。	
関連性	$MN_RUN_OVERRIDE_0	

12204 MD 番号	$MN_PERMANENT_ROT_AX_FEED[n] 回転軸の固定速度	
初期設定 :	最小入力リミット :	最大入力リミット :
変更が有効になるための条件 : 電源オン	保護レベル : 7/2	単位 : deg/min
データタイプ : DOUBLE		
意味	自動モードの場合 : 固定速度がインタフェース信号を介して起動されるとき，プログラムされた送り速度の代わりに固定速度が使用される。 JOG モードの場合 : 固定速度がインタフェース信号を介して起動され，直線軸のために移動キー w を使用するとき，選択された方向で，固定速度で移動が実行される。 n = 0, 1, 2, 3 は，固定フィード 1, 2, 3, 4 を意味する。	
例外，エラー	$MA_MAX_AX_VELO によって定義される最高速度が有効である。 オーバライドの効果は，$MN_RUN_OVERRIDE_0 によって決まる。	
関連性	$MN_RUN_OVERRIDE_0	

インタフェース信号

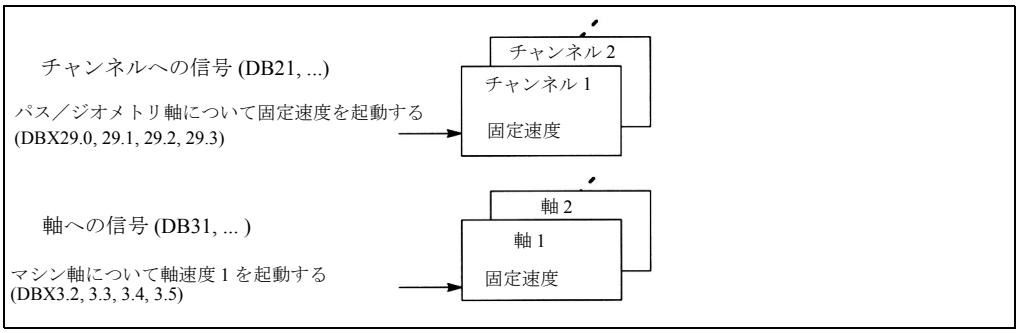


図 1.171 固定速度用インタフェース信号の概観

DB 21, 22, ... DBX29.0 DBX29.1 DBX29.2 DBX29.3 データブロック	**パス／ジオメトリ軸の固定速度 1 を起動する** **パス／ジオメトリ軸の固定速度 2 を起動する** **パス／ジオメトリ軸の固定速度 3 を起動する** **パス／ジオメトリ軸の固定速度 4 を起動する** **チャンネルへの信号 (PLC → NCK)**
エッジ評価 : なし	信号更新 : 周期的
説明	次の信号は，固定速度ファンクションを選択／選択解除し，どの固定速度がパス／ジオメトリ軸について有効であるかを指定する。 ビット 3　ビット 2　ビット 1　ビット 0　意味 0　0　0　0　固定速度が選択解除される 0　0　0　1　固定速度 1 が選択される 0　0　1　0　固定速度 2 が選択される 0　1　0　0　固定速度 3 が選択される 1　0　0　0　固定速度 4 が選択される
関連性	$MN_PERMANENT_FEED[n] $MN_RUN_OVERRIDE_0

DB 31, 32, ... DBX3.2 DBX3.3 DBX3.4 DBX3.5 データブロック	**マシン軸の固定速度 1 を起動する** **マシン軸の固定速度 2 を起動する** **マシン軸の固定速度 3 を起動する** **マシン軸の固定速度 4 を起動する** **軸への信号 (PLC → NCK)**
エッジ評価 : なし	信号更新 : 周期的
説明	次の信号は，固定速度ファンクションを選択／選択解除し，どの固定速度がマシン軸について有効であるかを指定する。 ビット 5　ビット 4　ビット 3　ビット 2　意味 0　0　0　0　固定速度が選択解除される 0　0　0　1　固定速度 1 が選択される 0　0　1　0　固定速度 2 が選択される 0　1　0　0　固定速度 3 が選択される 1　0　0　0　固定速度 4 が選択される
関連性	$MN_PERMANENT_FEED[n] $MN_RUN_OVERRIDE_0

1.18.3 補足条件

ファンクション 「一ブロック中の複数のフィード」のみが同期化アクションのファンクションと関連づけて利用できます。

参照： /FBSY/, Synchronized Actions

1.18.4 データの説明 (MD，SD)

■ 一般マシンデータ

10704 MD 番号	DRYRUN_MASK ドライランフィードレートを起動する	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : -
データタイプ : BYTE	適用開始 SW バージョン : 4.2	
意味 :	DRYRUN_MASK = 1 がイネーブルである場合，ドライランフィードレートも，プログラム実行中に起動することができる (パートプログラムブロック中で)。	
関連性	SD 42100: DRY_RUN_FEED	

12000 MD 番号	OVR_AX_IS_GRAY_CODE グレイコード化された軸フィードオーバライドスイッチ	
初期設定 : 1	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	このマシンデータを用いて，軸フィードオーバライドスイッチを PLC インタフェースのインタフェースコーディングに適用する。 1: 「フィードオーバライド」インタフェース信号 (DB31, ... DBB0) の 5 つのビットが，グレイコードとして解釈される。読み値は，スイッチ設定に相当する。MD 12010: OVR_FACTOR_AX_SPEED [n] のテーブルから正しいオーバライド係数を選択するためのポインタとして使用する。 0: PLC インタフェースのフィードオーバライドバイトが，パーセントで表されるオーバライド値をバイナリ 8- ビットの表記として解釈される。	
関連性	「フィードオーバライド」インタフェース信号 (DB31, ... DBB0), (軸別) MD 12010: OVR_FACTOR_AX_SPEED [n] (軸フィードオーバライドスイッチの評価)	

<table>
<tr><td>12010
MD 番号</td><td colspan="2">OVR_FACTOR_AX_SPEED [n]
軸フィードオーバライドスイッチの評価</td></tr>
<tr><td>初期設定 : 表参照</td><td>最小入力リミット : 0.00</td><td>最大入力リミット : 2.00</td></tr>
<tr><td>変更が有効になるための条件 :
電源オン</td><td>保護レベル : 2</td><td>単位 : 係数</td></tr>
<tr><td>データタイプ : DOUBLE</td><td colspan="2">適用開始 SW バージョン : 1.1</td></tr>
<tr><td>意味 :</td><td colspan="2">グレイコード化されたインタフェースの軸フィードオーバライドスイッチにおける各設定を評価するための係数が，このマシンデータに入力される必要がある。係数はスイッチ設定の順に (係数 1 → スイッチ設定 1, 係数 2 → 2 スイッチ設定 2, など) 入力しなければならない。
最大 31 のスイッチ設定のある軸フィードオーバライドが使用可能。任意の係数を割当てることができるが，1 番目のスイッチ設定にはゼロを割当てることを推奨する。1 番目のスイッチ設定はゼロで初期化することが望ましい。

<table>
<tr><th>スイッチ設定</th><th>軸フィード
オーバライド係数
(標準値)</th><th>指数 [n]</th></tr>
<tr><td>1</td><td>0.0</td><td>0</td></tr>
<tr><td>2</td><td>0.01</td><td>1</td></tr>
<tr><td>3</td><td>0.02</td><td>2</td></tr>
<tr><td>4</td><td>0.04</td><td>3</td></tr>
<tr><td>5</td><td>0.06</td><td>4</td></tr>
<tr><td>6</td><td>0.08</td><td>5</td></tr>
<tr><td>7</td><td>0.10</td><td>6</td></tr>
<tr><td>8</td><td>0.20</td><td>7</td></tr>
<tr><td>9</td><td>0.30</td><td>8</td></tr>
<tr><td>10</td><td>0.40</td><td>9</td></tr>
<tr><td>11</td><td>0.50</td><td>10</td></tr>
<tr><td>12</td><td>0.60</td><td>11</td></tr>
<tr><td>13</td><td>0.70</td><td>12</td></tr>
<tr><td>14</td><td>0.75</td><td>13</td></tr>
<tr><td>15</td><td>0.80</td><td>14</td></tr>
<tr><td>16</td><td>0.85</td><td>15</td></tr>
<tr><td>17</td><td>0.90</td><td>16</td></tr>
<tr><td>18</td><td>0.95</td><td>17</td></tr>
<tr><td>19</td><td>1.00</td><td>18</td></tr>
<tr><td>20</td><td>1.05</td><td>19</td></tr>
<tr><td>21</td><td>1.10</td><td>20</td></tr>
<tr><td>22</td><td>1.15</td><td>21</td></tr>
<tr><td>23</td><td>1.20</td><td>22</td></tr>
<tr><td>24</td><td>1.20</td><td>23</td></tr>
<tr><td>25</td><td>1.20</td><td>24</td></tr>
<tr><td>26</td><td>1.20</td><td>25</td></tr>
<tr><td>27</td><td>1.20</td><td>26</td></tr>
<tr><td>28</td><td>1.20</td><td>27</td></tr>
<tr><td>29</td><td>1.20</td><td>28</td></tr>
<tr><td>30</td><td>1.20</td><td>29</td></tr>
<tr><td>31</td><td>1.20</td><td>30</td></tr>
</table>

マシンデータの指数 [n] は次のようにコード化される :
[スイッチ設定]: 0 ～ 30</td></tr>
<tr><td>この MD は次の場合は無意味</td><td colspan="2">MD 12000: OVR_AX_IS_GRAY_CODE = 0</td></tr>
<tr><td>関連性</td><td colspan="2">「 フィードオーバライド」インタフェース信号 (DB31, ... DBB0), (軸別)</td></tr>
</table>

12030 MD 番号	OVR_FACTOR_FEEDRATE[n] **パスフィードオーバライドスイッチの評価**	
初期設定 : 表参照	最小入力リミット : 0.00	最大入力リミット : 2.00
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : 係数
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	グレイコード化されたインタフェースのパスフィードオーバライドスイッチにおける各設定を評価するための係数が，このマシンデータに入力される必要がある。 係数はスイッチ設定の順に (係数 1 → スイッチ設定 1, 係数 2 → 2 スイッチ設定 2, など) 入力しなければならない。 最大 31 のスイッチ設定のあるパスフィードオーバライドスイッチが使用可能。任意の係数を割当てることができるが，1 番目のスイッチ設定にはゼロを割当てることが望ましい。1 番目のスイッチ設定はゼロで初期化することが望ましい。	

スイッチ設定	早送り オーバライド (標準値)	指数 [n]
1	0.0	0
2	0.01	1
3	0.02	2
4	0.04	3
5	0.06	4
6	0.08	5
7	0.10	6
8	0.20	7
9	0.30	8
10	0.40	9
11	0.50	10
12	0.60	11
13	0.70	12
14	0.75	13
15	0.80	14
16	0.85	15
17	0.90	16
18	0.95	17
19	1.00	18
20	1.05	19
21	1.10	20
22	1.15	21
23	1.20	22
24	1.20	23
25	1.20	24
26	1.20	25
27	1.20	26
28	1.20	27
29	1.20	28
30	1.20	29
31	1.20	30

スイッチ設定フィードオーバライド係数 (標準値) 指数 [n]31 番目のスイッチ設定は，速度制御の特殊ファンクションを備えている：
31 番目のオーバライド係数の設定は，速度制御がパスフィードを増やすために保持するダイナミックリザーブを定義する。
設定は，実際に使用される最大オーバライド係数に相当しなければならない。
31 番目の値のファンクションは，バイナリコード化されたインタフェースの使用時には，MD 12100:OVR_FACTOR_LIMIT_BIN (バイナリコード化されたオーバライドスイッチのための制限) の効果に一致する。

参照： 1.3 連続送り，イグザクトストップモードと先読み (B1)

マシンデータの指数 [n] は下記のようにコード化される :

[スイッチ設定]: 0 ～ 30

この MD は次の場合は無意味	MD 12020: OVR_FEED_IS_GRAY_CODE = 0
関連性	「フィードオーバライド」インタフェース信号 (DB21, ... DBB4)

12050 MD 番号	OVR_FACTOR_RAPID_TRA[n] 早送りオーバライドスイッチの評価	
初期設定 : 表参照	最小入力リミット : 0.00	最大入力リミット : 1.00
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : 係数
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	グレイコード化されたインタフェースの早送りオーバライドスイッチにおける各設定を評価するための係数が，このマシンデータに入力される必要がある。 係数はスイッチ設定の順に (係数 1 → スイッチ設定 1, 係数 2 → 2 スイッチ設定 2, など) 入力しなければならない。 最大 31 のスイッチ設定のある早送りオーバライドスイッチが使用可能。任意の係数を割当てることができるが，1 番目のスイッチ設定にはゼロを割当てることが望ましい。1 番目のスイッチ設定はゼロで初期化することが望ましい。	
この MD は次の場合は無意味	MD 12040: OVR_RAPID_IS_GRAY_CODE = 0	
関連性	「早送りオーバライド」インタフェース信号 (DB21, ... DBB5)	

スイッチ設定	早送り オーバライド (標準値)	指数 [n]
1	0.0	0
2	0.01	1
3	0.02	2
4	0.04	3
5	0.06	4
6	0.08	5
7	0.10	6
8	0.20	7
9	0.30	8
10	0.40	9
11	0.50	10
12	0.60	11
13	0.70	12
14	0.75	13
15	0.80	14
16	0.85	15
17	0.90	16
18	0.95	17
19	1.00	18
20	1.05	19
21	1.10	20
22	1.15	21
23	1.20	22
24	1.20	23
25	1.20	24
26	1.20	25
27	1.20	26
28	1.20	27
29	1.20	28
30	1.20	29
31	1.20	30

マシンデータの指数 [n] は，下記のとおりコード化される :
[スイッチ設定]: 0 ～ 30

12070 MD 番号	OVR_FACTOR_SPIND_SPEED[n] スピンドルオーバライドスイッチの評価	
初期設定 : 表参照	最小入力リミット : 0.00	最大入力リミット : 2.00
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : 係数
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	グレイコード化されたインタフェースのスピンドルオーバライドスイッチにおける各設定を評価するための係数が，このマシンデータに入力される必要がある。 係数はスイッチ設定の順に (係数 1 → スイッチ設定 1, 係数 2 → 2 スイッチ設定 2, など) 入力しなければならない。 最大 31 のスイッチ設定のあるスピンドルオーバライドスイッチが使用可能。任意の係数を割当てることができるが，1 番目のスイッチ設定にはゼロを割当てることが望ましい。1 番目のスイッチ設定はゼロで初期化することが望ましい。	

スイッチ設定	スピンドル オーバライド (標準値)	指数 [n]
1	0.50	0
2	0.55	1
3	0.60	2
4	0.65	3
5	0.70	4
6	0.75	5
7	0.80	6
8	0.85	7
9	0.90	8
10	0.95	9
11	1.00	10
12	1.05	11
13	1.10	12
14	1.15	13
15	1.20	14
16	1.20	15
17	1.20	16
18	1.20	17
19	1.20	18
20	1.20	19
21	1.20	20
22	1.20	21
23	1.20	22
24	1.20	23
25	1.20	24
26	1.20	25
27	1.20	26
28	1.20	27
29	1.20	28
30	1.20	29
31	1.20	30

31 番目のスイッチ設定は，速度制御の特殊ファンクションを備えている：
31 番目のオーバライド係数は，速度制御がスピンドルフィードを増やすために保留するダイナミックリザーブを定義する。設定は，実際に使用される最大オーバライド係数に相当しなければならない。31 番目の値のファンクションは，バイナリコード化されたインタフェースの使用時に，MD 12100:OVR_FACTOR_LIMIT_BIN (バイナリコード化されたオーバライドスイッチのための制限) の効果に一致する。

参照： 1.3 連続送り，イグザクトストップモードと先読み (B1)

マシンデータの指数 [n] は下記のようにコード化される :

[スイッチ設定]: 0 ～ 30

この MD は次の場合は無意味	MD 12060: OVR_SPIND__IS_GRAY_CODE = 0

12070 MD 番号	OVR_FACTOR_SPIND_SPEED[n] スピンドルオーバライドスイッチの評価
関連性	「スピンドル」インタフェース信号 (DB31, ... DBB0)

12020 MD 番号	OVR_FEED_IS_GRAY_CODE グレイコード化されたパスフィードオーバライドスイッチ	
初期設定 : 1	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	このマシンデータを使って，PLC インタフェースのインタフェースコーディングにパスフィードオーバライドスイッチを適用する。 1: 「フィードオーバライド」PLC インタフェース信号ビットがグレイコードとして解釈される。読み値は，スイッチ設定に相当する。MD 12030: OVR_FACTOR_FEEDRATE [n] の表から正しいオーバライド係数を選択するためのポインタとして使用される。 0: PLC インタフェースのフィードオーバライドバイトは，パーセントで示されるオーバライド値をバイナリ 8 ビットで表示として解釈される。	
関連性	「フィードオーバライド」インタフェース信号 (DB21, ... DBB4) MD 12030: OVR_FACTOR_FEEDRATE [n](連続パスオーバライドスイッチの評価)	

12100MD 番号	OVR_FACTOR_LIMIT_BIN バイナリコード化されたオーバライドスイッチの制限	
初期設定 : 1.2	最小入力リミット : 0.00	最大入力リミット : 2.00
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	パス，軸，スピンドルのフィードのためにバイナリコード化されたインタフェースを使用するとき，このマシンデータはオーバライド係数の追加的制限として使用できる。 最大値 • チャンネル別フィードオーバライドの場合，200% • チャンネル別早送りオーバライドの場合，100% • 軸別フィードオーバライドの場合，200% • スピンドルオーバライドの場合，200% 上記の最大値は，MD: OVR_FACTOR_LIMIT_BIN に入力された制限値より小さい場合は，有効となります。 例 : OVR_FACTOR_LIMIT_BIN = 1.20 ⇒ 下記のための最大オーバライド係数 • チャンネル別フィードオーバライド =120% • チャンネル別早送りオーバライド =100% • 軸別フィードオーバライド =120% • スピンドルオーバライド =120% この値は，また，パスとスピンドルのフィードレートを増やすための速度制御によって保持されるダイナミックリザーブを定義する。 参照： 1.3 連続送り，イグザクトストップモードと先読み (B1)	

12040 MD 番号	OVR_RAPID_IS_GRAY_CODE グレイコード化された早送りオーバライドスイッチ	
初期設定 : 1	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	このマシンデータを使って，PLC インタフェースのインタフェースコーディングに早送りオーバライドスイッチを適用する。 1: 「早送りオーバライド」PLC インタフェース信号の 5 ビット がグレイコードとして解釈される。読み値は，スイッチ設定に相当する。MD 12050: OVR_FACTOR_RAPID_TRA. [n] の表から正しいオーバライド係数を選択するためのポインタとして使用される。 0: PLC インタフェースの早送りオーバライドバイトは，パーセントで示されるオーバライド値をバイナリ 8 ビット表示として解釈される。	
関連性	「早送りオーバライド」インタフェース信号 (DB21, ... DBB5) MD 12050: OVR_FACTOR_RAPID_TRA.[n](早送りオーバライドスイッチの評価)	

12060 MD 番号	OVR_SPIND_IS_GRAY_CODE グレイコード化されたスピンドルオーバライドスイッチ	
初期設定 : 1	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	このマシンデータを使って，PLC インタフェースのインタフェースコーディングにパスフィードオーバライドスイッチを適用する。 1: 「スピンドルオーバライド」PLC インタフェース信号の 5 ビット がグレイコードとして解釈される。読み値は，スイッチ設定に相当する。MD 12070:OVR_FACTOR_SPIND_ SPEED [n] の表から正しいオーバライド係数を選択するためのポインタとして使用される。 0: PLC インタフェースのスピンドルオーバライドバイトは，パーセントで示されるオーバライド値をバイナリ 8 ビット表示として解釈される。	
関連性	「スピンドルオーバライド」インタフェース信号 (DB31, ... DBB0) MD 12070: OVR_FACTOR_SPIND_SPEED[n](スピンドルオーバライドスイッチの評価)	

12080 MD 番号	OVR_REFERENCE_IS_PROG_FEED **オーバライド参照速度**	
初期設定 : 1	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2/4	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 2.1	
意味 :	この MD には，インタフェース信号によって与えられるスピンドルオーバライドに MD/SD によって制限される速度あるいはプログラムされた速度を適用するかどうかを入力する。 1: プログラムされた速度に関して，スピンドルオーバライドが機能する。(プログラムされた速度 = スピンドルオーバライド 100%) 0: MD あるいは SD によって制限された速度で，スピンドルオーバライドが機能する。(MD/SD による制限速度 = スピンドルオーバライド 100%)	
関連性	速度制限は，下記の MDs あるいは SDs の影響を受ける : MD 35100: SPIND_VELO_LIMIT Maximum スピンドル最高速度 MD 35130: GEAR_STEP_MAX_VELO_LIMIT ギアステップの最高速度 MD 35160: SPIND_EXTERN_VELO_LIMIT PLC によるスピンドル速度制限 SD 43220: SPIND_MAX_VELO_G26 スピンドル最高速度 SD 43230: SPIND_MAX_VELO_LIMSG G96 でスピンドル速度制限	

12082 MD 番号	OVR_REFERENCE_IS_MIN_FEED **パスオーバライドの基準を定義**	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になる条件 : 電源オン	保護レベル : 2 / 7	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 4.1	
意味 :	マシン制御パネルを介して指定されたパスフィードの基準速度は，標準とは異なった設定が可能である。 0: 標準 : オーバライドは，プログラムされたフィードに関連する。 1: 例外 : オーバライドが，プログラムされたフィードまたはパスフィードリミットに関連しており，これにより値より小さいものになる。大幅にフィードを減らすためにも (可能な軸ダイナミックスのため)，オーバライド値 (0 から 100% の範囲で) の効果は常に一目でわかる。	

■ チャンネル別マシンデータ

20172 MD 番号	COMPRESS_VELO_TOL 圧縮のためのパスフィードの最大許容ずれ	
初期設定 : 1000	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 電源オン	保護レベル : 2 / 7	単位 : mm/min
データタイプ : DOUBLE	適用開始 SW バージョン : 3.2	
意味 :	パスフィードの圧縮の最大許容ずれを示す。値が大きいほど，たくさんの短いブロックが一つの長いブロックに圧縮できる。圧縮可能ブロックの最大数は，スプラインバッファのサイズによって制限される。	
関連性	$MA_COMPRESS_POS_TOL[AXn] $MC_COMPRESS_BLOCK_PATH_LIMIT	
参照	プログラミングガイド	

20750 MD 番号	ALLOW_G0_IN_G96 G96 のための G0 ロジック	
初期設定 : 1	最小入力リミット : 0	最大入力リミット :1
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	このマシンデータを使って，周速一定制御 (G96) の選択時に，G0 ブロック内のスピンドルの速度応答を定義する。 1: G0 ブロックでは，スピンドル速度が，G0 に等しくなかった前のブロックの最後の値で一定のままになる。G0 を含まないブロックが連続する前には，スピンドル速度が次のブロックのフェーシング軸の位置に属する値まで加速される。 0: G0 ブロックでは，スピンドル速度が，フェーシング軸の位置に従って変化する。	
この MD は次の場合は無意味	フィードタイプ ≠ G96	

21220 MD 番号	MULTFEED_ASSIGN_FASTINNCK I/Os の入力バイトの 「一つのブロック中の複数のフィード」への割当て	
初期設定 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : HEX
データタイプ : DWORD	適用開始 SW バージョン : 2.1	
意味 :	MD: MULTFEED_ASSIGN_FASTIN (NCK I/Os の入力バイトの 「1 つのブロック中の複数のフィード」への割当て) では，NCK I/Os の 2 以下のディジタル入力バイトまたは NCK I/Os のコンパレータ入力バイトは，「1 つのブロック中の複数フィード」ファンクションの入力バイトに割当てることができる。割当てられた入力信号は，マシンデータで反転できる。 MD は次のようにコード化される : Bit 0 ～ 7　使用済み 1 番目のディジタル入力バイトあるいはコンパレータ入力バイトの数 Bit 8 ～ 15: 使用済み 2 番目のディジタル入力バイトあるいはコンパレータ入力バイトの数 Bit 16 - 23: 1 番目のバイトを示すインバータスクリーンフォーム Bit 24 ～ 31: 2 番目のバイトを示すインバータスクリーンフォーム Bit=0: 反転しない Bit=1: 反転する 2 つ目のバイトを入力する場合，使用前に，1 番目のバイトと 2 番目のバイトの内容が OR される。ディジタル入力の数は，次のように指定しなくてはならない : 1: 使用中バイト用 2 ～ 5: 外部バイト用 コンパレータ入力バイト数は，次のように指定しなければならない : 128: コンパレータ 1 用 (80Hex に対応する) 129: コンパレータ 2 用 (81Hex に対応する)	
アプリケーション例	ファンクション 「1 ブロック中の複数フィード」の場合，外部ディジタル入力バイト 3 を 1 番目のバイトとして使用し，コンパレータ 2 を 2 番目のバイトとして使用する。反転も次のように実行される : ディジタル入力バイトのビット 0, 2, 3 コンパレータ入力バイトのビット 0, 1, 5, 7 ⇒ MD: MULTFEED_ASSIGN_FASTIN=A30D8103 (十六進法フォーマット)	

21230 MD 番号	MULTFEED_STORE_MASK 「1 ブロック中の複数のフィード」ファンクション用入力信号を保存する	
初期設定 : 0	最小入力リミット : ***	最大入力リミット : ***
変更が有効になるための条件 : 電源オン	保護レベル : 2/7	単位 : HEX
データタイプ : BYTE	適用開始 SW バージョン : 2.1	
意味 :	ファンクション 「1 ブロック中の複数フィード」の フィード F2 ～ F7 用信号の優先順位は，入力バイトのビット番号が大きくなるほど，低くなる。最高優先順位の信号が，そのときのフィードを決定する。 MD: MULTFEED_STORE_MASK (「一ブロック中の複数のフィード」ファンクションの入力信号を保存する) を用いて，最高優先順位入力を省くとき，応答を指定する : セットされたビット 2 ～ 7 には，入力信号が省略されて，優先順位の低いものが出てても，それぞれ最高優先順位入力信号によってすでに選択されている関連フィード (F2 ～ F7) が保留されるという効果がある。 MD は次のようにコード化される : ビット 0 ～ 1　　無意味 ビット 2 ～ 7:　　フィード信号の保存応答 ビット 8 ～ 31:　　予備	

22410 MD 番号	F_VALUES_ACTIVE_AFTER_RESET リセット後アクティブな F ファンクション	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
変更が有効になるための条件 : 電源オン	保護レベル : 2	単位 : -
データタイプ : BOOLEAN	適用開始 SW バージョン : 1.1	
意味 :	1: 最も最近プログラムされた F, FA, OVR および OVRA の値は，RESET 後にもアクティブになる。 0: 多様な値が，リセット後のデフォルト値にセットされる。	
関連性	MD 22240: AUXFU_F_SYNC_TYPE F ファンクションの出力時間	

■ チャンネル別設定データ

42100 SD 番号	DRY_RUN_FEED ドライランフィード	
初期設定 : 5000	最小入力リミット : 0	最大入力リミット : 正
変更が有効になるための条件 : 即時	保護レベル : MMC-MD 9220	単位 : mm/min
データタイプ : DOUBLE	適用開始 SW バージョン : 1.1	
意味 :	アクティブなドライラン用フィードレートは，この設定データに入力される必要がある。 この設定データは，「パラメータ」ユーザーエリアの操作パネルで変更できる。 入力されたドライランフィードは常に直線フィードとして解釈される (G94)。 ドライランフィードを PLC インタフェースを介して起動する場合，ドライランフィードは，リセット後にプログラムされたフィードの代わりにパスフィードとして使用される。 プログラムされた速度が，このデータに保存された速度より大きい場合は，プログラムされた速度が移動に使用される。	
アプリケーション例	プログラムテスト	
関連性	"Activate dry run feed" 「ドライランフィードを起動する」インタフェース信号 (DB21, ... DBX0.6) "Dry run feed selected" 「ドライランフィードが選択される」インタフェース信号 (DB21, ... DBX24.6)	

42600 MD 番号	JOG_FEED_PER_REV_SOURCEJOG モードでは，回転するフレームが作動するジオメトリ軸用回転フィードレート	
初期設定 : 0	最小入力リミット :	最大入力リミット :
変更が有効になるための条件 : 電源オン	保護レベル : 7	単位 :
データタイプ : DWORD	適用開始 SW バージョン : 3	
意味 :	0= 回転フィードレートがアクティブではない >0= 回転フィードレートを得るための回転軸／スピンドルのマシン軸指数 - 1= 軸／スピンドルがアクティブであるチャンネルのマスタスピンドルから得られた回転フィドレート	
関連性	SD 53000: ASSIGN_FEED_PER_REV_SOURCE (位置決め軸／スピンドル用回転フィードレート)	

43300 MD 番号	ASSIGN_FEED_PER_REV_SOURCE 位置決め軸／スピンドル用回転フィードレート	
初期設定 : 0	最小入力リミット : 0	最大入力リミット :
変更が有効になるための条件 : 電源オン	保護レベル : 7	単位 :
データタイプ : DWORD	適用開始 SW バージョン : 3	
意味 :	0= 回転フィードレートがアクティブではない >0= 回転フィードレートが得るための回転軸／スピンドルのマシン軸指数 -1= 軸／スピンドルがアクティブであるチャンネルのマスタスピンドルから得られた回転フィドレート	
関連性	SD 52011: JOG_FEED_PER_REV_SOURCE (JOG モードで，フレームが回転して有効になるジオメトリ軸用回転フィードレート)	

1.18.5 信号の説明

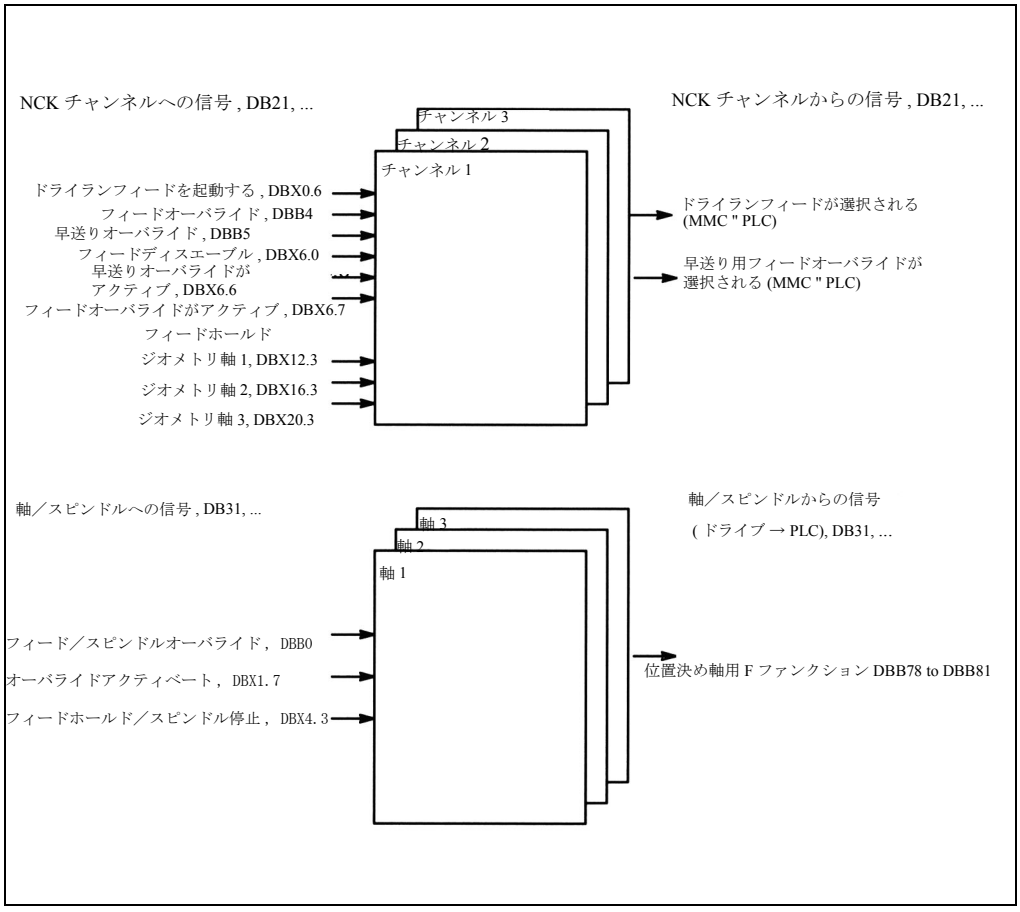


図 1.172 フィード用 PLC インタフェース信号

■ チャンネル別信号

■ チャンネルへの信号

DB21, ... DBX0.6 データブロック	ドライランフィードを起動 チャンネルへの信号 (PLC → NCK)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	SD 42100: DRY_RUN_FEED に定義されたドライランフィードは，これがプログラムされたフィードより大きい場合，プログラムされたフィード (G01, G02, G03 で) の代わりに移動に使われる。 ドライランフィードは，リセット後，アクティブである。このインタフェース信号は，チャンネルが " リセット " ステータスであるとき，NC 始動の際に評価される。ドライランフィードは，PLC または操作パネルから起動できる。 操作パネルからの起動時，「ドライランフィードが選択される」PLC インタフェース信号がイネーブルされ，PLC 基本プログラムによって 「ドライランフィードを起動」インタフェース信号に送る。 PLC で選択されると，「ドライランフィードを起動」インタフェース信号が，PLC ユーザプログラムで設定するのに必要である。	
信号ステータス 0 あるいは信号遷移 1 → 0	プログラムされたフィードは移動に使用する。 リセット後にアクティブとなる。	
アプリケーション例	フィードレートを大きくして，ワークピースプログラムをテストする。	
例外，エラー	信号が，G33 ブロック内で "0" に変わるとき，プログラムフィードはブロックの終了まで起動しない。NC 停止がトリガされなかったからである。	
関連性	「ドライランフィードレートが選択される」インタフェース信号 (DB21, ... DBX24.6) SD 42100: DRY_RUN_FEED (ドライランフィード)	

DB21, ... DBB4 **データブロック**	**フィードオーバライド** **チャンネルへの信号 (PLC → NCK)**	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	フィードオーバライドは，バイナリまたはグレイコーディングで，PLC を介して定義できる。 バイナリコード化では，フィード値が，単位 % で解釈される。 バイトのバイナリ値に従って，0% から 200% までのフィード変更が可能。恒久的割当ては，次のとおり : コード フィード → オーバライド係数 00000000 → 0.00 → 0% 00000001 → 0.01 → 1% 00000010 → 0.02 → 2% 00000011 → 0.03 → 3% ・ → ・ 11001000 → 2.00 → 200% デュアル値 E ≧ 200 は 200% に制限される。MD 12100: OVR_FACTOR_LIMIT_BIN (バイナリコード化されたオーバライドスイッチの制限) を，最大フィードオーバライドの追加制限として使用できる。 グレイコード化では，個々のスイッチ設定が次のコードに割当てられる :	

スイッチ設定	コード	フィード オーバライド
1	00001	0.0
2	00011	0.01
3	00010	0.02
4	00110	0.04
5	00111	0.06
6	00101	0.08
7	00100	0.10
8	01100	0.20
9	01101	0.30
10	01111	0.40
11	01110	0.50
12	01010	0.60
13	01011	0.70
14	01001	0.75
15	01000	0.80
16	11000	0.85
17	11001	0.90
18	11011	0.95
19	11010	1.00
20	11110	1.05
21	11111	1.10
22	11101	1.15
23	11100	1.20
24	10100	1.20
25	10101	1.20
26	10111	1.20
27	10110	1.20
28	10010	1.20
29	10011	1.20
30	10001	1.20
31	10000	1.20

表 1.75 フィードオーバライドのグレイコーディング

表で指定されたフィードオーバライド係数は，NC マシンデータ 12030:OVR_FACTOR_FEEDRATE [n] に保存される。

表には，デフォルト値が含まれる。標準マシン制御パネルの可能なスイッチ設定数は，”Configuring guid”に記載されている。

関連性	「フィードオーバライドがアクティブ」インタフェース信号 (DB21, ... DBX6.7)MD 12030: OVR_FACTOR_FEEDRATE [n](パスフィードオーバライドスイッチの評価)MD 12100: OVR_FACTOR_LIMIT_BIN (バイナリコード化されたオーバライドスイッチ用リミット)

DB21, ... DBB5 **データブロック**	**早送りオーバーライド** **チャンネル (PLC → NCK) への信号**	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	早送りオーバライドは，バイナリまたはグレイコーディングで，PLC を介して定義できる。バイナリコーディングでは，早送りオーバライドが，単位 % で解釈される。バイトのバイナリ値に従って，0% から 200% までのフィード変更が可能。 恒久的割当ては，次のとおり : コード → 早送りオーバライド係数 00000000 → 0.00 → 0% 00000001 → 0.01 → 1% 00000010 → 0.02 → 2% 00000011 → 0.03 → 3% . → . 01100100 → 1.00 → 100% デュアル値 > 100 は 100% に制限される。 MD 12100: OVR_FACTOR_LIMIT_BIN (バイナリコード化されたオーバライドスイッチの制限) を，最大早送りオーバライドの追加制限として使用できる。 グレイコーディングでは，個々のスイッチ設定が次のコードに割当てられる :	

スイッチ	設定コード	早送りオーバライド係数 (標準値)
1	00001	0.0
2	00011	0.01
3	00010	0.02
4	00110	0.04
5	00111	0.06
6	00101	0.08
7	00100	0.10
8	01100	0.20
9	01101	0.30
10	01111	0.40
11	01110	0.50
12	01010	0.60
13	01011	0.70
14	01001	0.75
15	01000	0.80
16	11000	0.85
17	11001	0.90
18	11011	0.95
19	11010	1.00
20	11110	1.00
21	11111	1.00
22	11101	1.00
23	11100	1.00
24	10100	1.00
25	10101	1.00
26	10111	1.00
27	10110	1.00
28	10010	1.00
29	10011	1.00
30	10001	1.00
31	10000	1.00

表 1.76 早送りオーバライドのグレイコーディング

表で指定された早送りオーバライド係数は，NC マシンデータ MD 12050:OVR_FACTOR_RAPID_TRA[n] に保存される。表には，デフォルト値が含まれる。標準マシン制御パネルの可能なスイッチ設定数は,”Configuring guid”に記載される。

関連性	「早送りオーバライドがアクティブ」 インタフェース信号 (DB21, ... DBX6.6) MD 12050: OVR_FACTOR_RAPID_TRA[n] (早送りオーバライドの評価) MD 12100: OVR_FACTOR_LIMIT_BIN (バイナリコード化されたオーバライドスイッチの制限)

DB21, ... DBX6.0 **データブロック**	**フィードディスエーブル チャンネルへの信号 (PLC → NCK)**	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは 信号遷移 0 → 1	信号は，すべての操作モードでのチャンネルにおいてアクティブである。 • 信号は，G33 (スレッド) がアクティブでない限り，互いに補間関係にあるすべての軸 (ジオメトリおよび同期化) をディスエーブルする。すべての軸は，パス輪郭に従って静止される。フィードディスエーブルのキャンセル時 (0 信号)，中断されたパートプログラムが続行する。 • 信号は，すべての位置決め軸用フィードディスエーブルをトリガする。移動軸は，ブレーキングが制御され (ランプストップ) 停止する。アラームは出力されない。 • 位置制御が保留される，すなわち次のエラーが削除される。 • 「フィードディスエーブル」がアクティブで，軸についてトラベルリクエストが発行される場合，このリクエストが保留され，「フィードディスエーブル」がキャンセルされるとすぐに実行される。軸が他の軸と補間関係にある場合は，これらの軸にも適用される。	
信号ステータス 0 あるいは信号 遷移 1 → 0	• フィードはチャンネルにおいてすべての軸についてイネーブルである。 • 「フィードディスエーブル」のキャンセル時に，トラベルリクエスト (" トラベルコマンド ") が，軸または軸のグループのために存在する場合，これが即時実行される。	
アプリケーション例	マシン制御パネルで FEED OFF を選択して，加工を停止する。	
例外，エラー	G33 がアクティブのとき，フィードディスエーブルが非アクティブである。	

DB21, ... DBX6.6 **データブロック**	**早送りオーバライドがアクティブ チャンネルへの信号 (PLC → NCK)**	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号 遷移 0 → 1	PLC インタフェースに入力された 0 から最大 100% までの早送りオーバライドは，チャンネル別に機能する。MD 12040: OVR_RAPID_IS_GRAY_CODE (グレイコード化された早送りオーバライドスイッチ)，MD 12050: OVR_FACTOR_RAPID_TRA [n](早送りオーバライドスイッチの評価) で早送りオーバライドが指定される。	
信号ステータス 0 あるいは信号 遷移 1 → 0	PLC インタフェースに入力された早送りオーバライドは無視される。早送りオーバライドが非アクティブのとき，NC は常に，内部オーバライド係数として 100% を使用する。 例外として，バイナリインタフェースはゼロが設定され，グレイコード化されたインタフェースは 1 番目のスイッチ設定になる。いずれの場合も，PLC インタフェースに入力されたオーバライド係数を使用する。バイナリインタフェースの場合は，オーバライド係数が 0 である。グレイコード化されたインタフェースの場合は，1 番目のスイッチ設定としてマシンデータに入力された値が，オーバライド値として出力される。	

DB21, ... DBX6.6 データブロック	早送りオーバライドがアクティブ チャンネルへの信号 (PLC → NCK)
アプリケーション例	オーバライド値は，マシン制御パネルで早送りオーバライドスイッチを使用して，通例は，オーバライド値を選択する。 「早送りオーバライドアクティブ」インタフェース信号を使用して，PLC ユーザプログラムから早送りオーバライドスイッチをイネーブルすることができる。たとえば，キースイッチを使って，新しい NC プログラムのスタートアップ中に行なう。
例外，エラー	G33, G63, G331 あるいは G332 がアクティブのとき，早送りオーバライドが非アクティブである。
関連性	「早送りオーバライド」インタフェース信号 (DB21, ... DBB5)

DB21, ... DBX6.7 データブロック	フィードオーバライドアクティブ チャンネルへの信号 (PLC → NCK)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	PLC インタフェースに入力された 0 から最大 200% までのフィードオーバライドは，パスフィードについてアクティブであり，したがって，関連軸については自動的にアクティブになる。 JOG モードでは，フィードオーバライドが軸に直接作用する。 オーバライド係数は， MD 12020: OVR_FEED_IS_GRAY_CODE (グレイコード化されたパスフィードオーバライドスイッチ), MD 12030: OVR_FACTOR_FEEDRATE [n](パスフィードオーバライドスイッチの評価) で指定される。	
信号ステータス 0 あるいは信号遷移 1 → 0	PLC インタフェースで入力されたフィードオーバライドは無視される。 フィードオーバライドが非アクティブのとき，NC は常に，100% を内部オーバライド係数として使用する。 例外として，バイナリインタフェースはゼロが設定され，グレイコード化されたインタフェースには 1 番目のスイッチが設定される。いずれの場合も，PLC インタフェースに入力されたオーバライド係数を使用する。バイナリインタフェースの場合は，オーバライド係数が 0 である。グレイコード化されたインタフェースの場合は，1 番目のスイッチ設定としてマシンデータに入力された値が，オーバライド値として出力される。	
アプリケーション例	オーバライド値は，マシン制御パネルでフィードオーバライドスイッチを使用して，通例，オーバライド値を選択する。「フィードオーバライドがアクティブ」インタフェース信号を使用して，PLC ユーザプログラムからフィードオーバライドスイッチをイネーブルすることができる。たとえば，キースイッチを使って，新しい NC プログラムのスタートアップ中に行なう。	
例外，エラー	G33, G63, G331 あるいは G332 がアクティブのとき，フィードオーバライドがアクティブである。	
関連性	「フィードオーバライド」インタフェース信号 (DB21, ... DBB4)	

DB21, ... DBX6.7 データブロック	フィードオーバライドアクティブ チャンネルへの信号 (PLC → NCK)

DB21, ... DBX12.3 16.3 20.3 データブロック	フィードホールド (ジオメトリ軸 1 ～ 3) チャンネルへの信号 (PLC → NCK)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	信号は JOG モードでのみアクティブになる。 • 信号はジオメトリ軸を停止させる。移動軸は，ブレーキングが制御され (ランプストップ) 停止する。アラームは出力されない。 • 位置制御が保留される，すなわち次のエラーが削除される。 • アクティブな「フィードホールド」で，軸についてトラベルリクエストが発行される場合，このリクエストは保留され，「フィードホールド」がキャンセルされるとすぐに実行される。軸が他の軸と補間関係にある場合は，これらの軸にも適用される。	
信号ステータス 0 あるいは信号遷移 1 → 0	• フィードはジオメトリ軸についてイネーブルされる。 • トラベルリクエスト (" トラベルコマンド ") が，「フィードホールド」のキャンセル時に，ジオメトリ軸のために存在する場合，これが即時実行される。	

DB21, ... DBX24.6 データブロック	ドライランフィードが選択される チャンネルへの信号 (MMC → PLC)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	ドライランフィードが選択される。 プログラムされたフィードの代わりに，SD 42100: DRY_RUN_FEED に入力されたドライランフィードを使用する。 操作パネルからの起動時，ドライランフィード信号は，PLC インタフェースに自動的に入力され，PLC 基本プログラムによって，「ドライランフィードを起動」PLC インタフェース信号に送られる。	
信号ステータス 0 あるいは信号遷移 1 → 0	ドライランフィードは選択されない。プログラムされたフィードがアクティブである。	
関連性	「ドライランフィードを起動」インタフェース信号 (DB21, ... DBX0.6) SD: DRY_RUN_FEED (ドライランフィード)	

DB21, ... DBX25.3 データブロック	早送り用フィードオーバライドが選択される チャンネルへの信号 (MMC _ PLC)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	フィードオーバライドスイッチは，早送りオーバライドスイッチとしてもアクティブである。100% を超えるオーバライド値は，100% 早送りオーバライドについて，最大値に制限される。 「早送り用フィードオーバライドが選択される」インタフェース信号は，操作パネルから PLC インタフェースに自動的に入力され，PLC 基本プログラムによって 「早送りオーバライドがアクティブ」PLC インタフェース信号に変換される。 「フィードオーバライド」インタフェース信号 (DB21, ... DBB4) は，PLC 基本プログラムによって，「早送りオーバライド」インタフェース信号 (DB21, ... DBB5) にコピーされる。	
信号ステータス 0 あるいは信号遷移 1 → 0	フィードオーバライドスイッチが，早送りオーバライドスイッチとして起動されない。	
アプリケーション例	独立した早送りオーバライドスイッチが使用できないとき，この信号を使用する。	

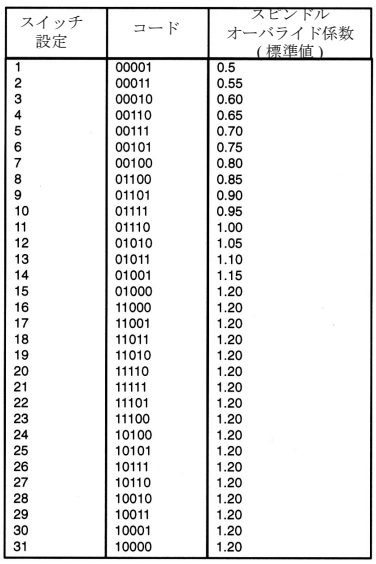
■ 軸 / スピンドル別信号

■ 軸／スピンドルへの信号

DB31, ... DBB0 データブロック	フィードオーバライド (軸別) 軸への信号 (PLC → NCK)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	軸別フィードオーバライドは，バイナリまたはグレイコーディングで，PLC を介して定義できる。バイナリコーディングでは，フィード値が，単位 % で解釈される。バイトのバイナリ値に従って，0% から 200% までのフィード変更が可能。 恒久的割当ては，次のとおり : コード → 軸別フィードオーバライド係数 00000000 → 0.00 → 0% 00000001 → 0.01 → 1% 00000010 → 0.02 → 2% 00000011 → 0.03 → 3% . . . 01100100 → 1.00 → 100% . . . 11001000 → 2.00 → 200% デュアル値 ＞ 200 は 200% に制限される。 MD 12100: OVR_FACTOR_LIMIT_BIN (バイナリコード化されたオーバライドスイッチの制限) を，最大フィードオーバライドの追加制限として使用できる。	
信号ステータス 1 あるいは信号遷移 0 → 1	グレイコーディングでは，個々のスイッチ設定が次のコードに割当てられる : 表 1.77 軸別フィードオーバライドのグレイコーディング 軸フィードオーバライドについて，表で指定された係数は，NC-MD 12010:OVR_FACTOR_AX_SPEED [n] に保存される。表には，デフォルト値が含まれる。標準マシン制御パネルの可能なスイッチ設定数は,”Configuring guid”に記載されている。	

スイッチ設定	コード	軸フィードオーバライド係数 (標準値)
1	00001	0.0
2	00011	0.01
3	00010	0.02
4	00110	0.04
5	00111	0.06
6	00101	0.08
7	00100	0.10
8	01100	0.20
9	01101	0.30
10	01111	0.40
11	01110	0.50
12	01010	0.60
13	01011	0.70
14	01001	0.75
15	01000	0.80
16	11000	0.85
17	11001	0.90
18	11011	0.95
19	11010	1.00
20	11110	1.05
21	11111	1.10
22	11101	1.15
23	11100	1.20
24	10100	1.20
25	10101	1.20
26	10111	1.20
27	10110	1.20
28	10010	1.20
29	10011	1.20
30	10001	1.20
31	10000	1.20

DB31, ... DBB0 データブロック	 フィードオーバライド (軸別) 軸への信号 (PLC → NCK)
関連性	「オーバライドがアクティブ」インタフェース信号 (DB31, ... DBX1.7) MD 12010: OVR_FACTOR_AX_SPEED [n] (軸フィードオーバライドスイッチの評価) MD 12100: OVR_FACTOR_LIMIT_BIN (バイナリコード化されたオーバライドスイッチの制限)

DB31, ... DBB19 **データブロック**	**スピンドルオーバライド** **スピンドルへの信号 (PLC → NCK)**	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	スピンドルオーバライドは，バイナリまたはグレイコーディングで，PLC を介して定義できる。オーバライド値は，スピンドルに対して，プログラムされた速度セットポイントのパーセンテージを決める。バイナリコーディングの場合，オーバライドは，単位 % で解釈される。バイトのバイナリ値に従って，0% から 200% までの速度変更が可能。 恒久的割当ては，次のとおり : コード　　スピンドルオーバライド係数 00000000　　0.00 → 0% 00000001　　0.01 → 1% 00000010　　0.02 → 2% 00000011　　0.03 → 3% .　　. .　　. .　　. 01100100　　1.00 → 100% .　　. .　　. .　　. 11001000　　2.00 → 200% バイナリ値 ＞ 200 は 200% に制限される。 MD 12100: OVR_FACTOR_LIMIT_BIN (バイナリコード化されたオーバライドスイッチの制限) を，最大スピンドルオーバライドの追加制限として使用できる。	
信号ステータス 1 あるいは信号遷移 0 → 1	グレイコーディングでは，個々のスイッチ設定が次のコードに割当てられる :  **表 1.78 スピンドルオーバライド用グレイコーディング** 表で指定されたスピンドル用係数は，MD 12070: OVR_FACTOR_SPIND_SPEED [n] に保存されている。表には，デフォルト値が含まれる。標準マシン制御パネルの可能なスイッチ設定数は，”Configuring guid” に記載されている。	

スイッチ設定	コード	スピンドルオーバライド係数 (標準値)
1	00001	0.5
2	00011	0.55
3	00010	0.60
4	00110	0.65
5	00111	0.70
6	00101	0.75
7	00100	0.80
8	01100	0.85
9	01101	0.90
10	01111	0.95
11	01110	1.00
12	01010	1.05
13	01011	1.10
14	01001	1.15
15	01000	1.20
16	11000	1.20
17	11001	1.20
18	11011	1.20
19	11010	1.20
20	11110	1.20
21	11111	1.20
22	11101	1.20
23	11100	1.20
24	10100	1.20
25	10101	1.20
26	10111	1.20
27	10110	1.20
28	10010	1.20
29	10011	1.20
30	10001	1.20
31	10000	1.20

DB31, ... DBB19 データブロック	スピンドルオーバライド スピンドルへの信号 (PLC → NCK)
関連性	「オーバライドがアクティブ」インタフェース信号 (DB31, ... DBX1.7) MD 12070: OVR_FACTOR_SPIND_SPEED [n](スピンドルオーバライドスイッチの評価) MD 12100: OVR_FACTOR_LIMIT_BIN (バイナリコード化されたオーバライドスイッチの制限)

DB31, ... DBX1.7 **データブロック**	**オーバライドアクティブ** **軸／スピンドルへの信号 (PLC → NCK)**	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	フィードオーバライドアクティブ : • PLC に入力された 0 から最大 200% までの軸別フィードオーバライドを使用する。MD 12000: OVR_AX_IS_GRAY_CODE (グレイコード化された軸フィードオーバライドスイッチ), MD 12010: OVR_FACTOR_AX_SPEED [n](軸フィードオーバライドスイッチの評価) でオーバライド係数を指定する。 スピンドルオーバライドアクティブ : • 0 から最大 200% のスピンドルオーバライドを使用する。MD 12060: OVR_SPIND_IS_GRAY_CODE (グレイコード化されたスピンドルオーバライドスイッチ), MD 12070: OVR_FACTOR_SPIND_SPEED [n](スピンドルオーバライドスイッチの評価) で，オーバライド係数を指定する。	
信号ステータス 0 あるいは信号遷移 1 → 0	現行の軸別フィードオーバライドまたはスピンドルオーバライドはアクティブでない。フィードオーバライドが非アクティブの場合，内部オーバライド係数として 100% を使用する。例外として，バイナリインタフェースはゼロに設定され，グレイコード化されたインタフェースは 1 番目のスイッチ設定になる。これらの場合には，PLC インタフェースに入力されたオーバライド係数を使用する。バイナリインタフェースの場合，オーバライド係数 = 0 である。グレイコード化されたインタフェースでは，1 番目のスイッチ設定用マシンデータに入力された値は，オーバライド値として出力される。	
アプリケーション例	通常，軸別フィードオーバライドスイッチあるいはマシン制御パネルのスピンドルオーバライドスイッチを使って，オーバライド値を指定する。「フィードオーバライドアクティブ」信号を使って，PLC ユーザプログラムからフィードオーバライドスイッチをイネーブルできる。たとえば，キースイッチを使って，新しい NC プログラムのスタートアップ中に行なえる。	
例外，エラー	• "spindle oscillation mode"(スピンドル振動モード) で，スピンドルオーバライドは，常に 100% で受け取られる。 • スピンドルオーバライドは，リミットが介入する前に，プログラムされた値で機能する (例 G26, LIMS...)。 • 次のファンクションがアクティブのとき，フィードオーバライドは，次のとおり。 - G33 : 非アクティブ - G63 : 非アクティブ (オーバライドが，NC では 100% にセットされる) - G331, G332 : 非アクティブ (オーバライドが，NC では 100% にセットされる) • 次のファンクションがアクティブのとき，スピンドルオーバライドは， - G63 : 非アクティブ (オーバライドが，NC では 100% にセットされる)	
関連性	「フィード／スピンドルオーバライド」インタフェース信号 (DB31, ... DBB0)	

DB31, ... DBX4.3 **データブロック**	**フィードレートホールド／スピンドル停止** **(軸別) 軸／スピンドルへの信号 (PLC → NCK)**	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	信号はすべてのモードでアクティブである。フィードホールド : • 信号は軸用フィードホールドをトリガする。移動軸は，ブレーキングが制御されて停止する (ランプストップ)。アラームは出力されない。 • 互いに補間関係にあるすべてのパス軸について，これらの軸のうち任意の 1 つの軸について「フィードホールド」が起動されるとき，信号が「フィードホールド」をトリガする。この場合，すべての軸がパス輪郭に従って停止する。フィードホールド信号のキャンセル時，中断されたパートプログラムの実行が再開する。 • 位置制御が保留される，すなわち次のエラーが削除される。 • アクティブな「フィードホールド」で，トラベルリクエストが発行される場合，このリクエストは，保留され，「フィードホールド」がキャンセルされるとすぐに実行される。軸が他の軸と補間関係にある場合は，これらの軸にも適用される。 スピンドル停止 : • スピンドルは，加速特性に従って，停止する。 • 位置決めモードでは，「スピンドル停止」信号を起動すると，位置決めプロセスが中断する。この応答は，個々の軸について適用される。	
信号ステータス 0 あるいは信号遷移 1 → 0	フィドホールド : • フィードが軸についてイネーブルになる。 • 「フィードホールド」のキャンセル時にトラベルリクエスト (トラベルコマンド) がアクティブの場合，即時にこれが実行される。 スピンドル停止 : • 速度がスピンドルについてイネーブルされる。 • 加速特性によって，または位置決めモードで，スピンドルが前の速度セットポイントまで加速され，位置決めが再開される。	
アプリケーション例	フィードホールド : • 「フィードホールド」がアクティブのとき，マシン軸の移動動作は始動しない。たとえば，軸モーションを許容しない一定の操作ステータス (ドアが閉まらない，など) が有力な場合。 スピンドル停止 : • ツールを変更する。 • セットアップ中に，補助ファンクション (M, S, H, T, D および F のファンクション) を入力する。	
例外，エラー	G331 または G332 がアクティブのとき，スピンドル停止が非アクティブである。	

■ 軸／スピンドルからの信号

DB31, ... DBX62.2 データブロック	回転フィードレートアクティブ 軸への信号 (NCK → PLC)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	G95 (回転フィードレート) を JOG モードあるいは 自動モードでプログラミングする場合	
アプリケーション例		
関連性	SD 41100: JOG_REV_IS_ACTIVE (JOG 用回転フィードレートがアクティブ) SD 42600: JOG_FEED_PER_REV_SOURCE (JOG モードで，回転するフレームがアクティブであるジオメトリ軸用回転フィード) SD 43300: ASSIGN_FEED_PER_REV_SOURCE (位置決め軸／スピンドル用回転フィードレート) MD 32040: JOG_REV_VELO_RAPID (早送りオーバレイ付き JOG モード用回転フィードレート) MD 32050: JOG_REV_VELO (JOG モード用回転フィードレート)	

DB31, ... DBB78-81 データブロック	位置決め軸用 F ファンクション 軸への信号 (NCK → PLC)	
エッジ評価 : なし	信号更新 : 周期的	信号が有効となる最初の SW バージョン : 1.1
信号ステータス 1 あるいは信号遷移 0 → 1	そのときのブロックにプログラムされた位置決め軸の F 値が，軸別 PLC インタフェース信号に入力される。 DB 番号とマシン軸番号の間の割当ては，軸名を使って実行される。この値は，他の値で上書されるまで保留される。 フォーマット : リアルフォーマットで 2 進数	
アプリケーション例	プログラムされた F 値を PLC で修正する，たとえば，定義された軸別フィードオーバライドを上書する	
関連性	MD 22240: AUXFU_F_SYNC_TYPE F （ファンクションの出力時間）	

1.18.6 例

なし

1.18.7 データフィールド，リスト

■ インタフェース信号

DB 番号	ビット，バイト	名称	参照
チャンネル別			
21, ...	0.6	ドライランフィードを起動する	
21, ...	4	フィードオーバライド	
21, ...	5	早送りオーバライド	
21, ...	6.0	フィードディスエーブル	
21, ...	6.6	早送りオーバライドがアクティブ	
21, ...	6.7	フィードオーバライドがアクティブ	
21, ...	12.3	1 フィードホールド，ジオメトリ軸 1	
21, ...	16.3	2 フィードホールド，ジオメトリ軸 2	

DB 番号	ビット，バイト	名称	参照
21, ...	20.3	フィードホールド，ジオメトリ軸 3	
21, ...	24.6	ドライランフィードが選択される	
21, ...	25.3	早送り用フィードオーバライドが選択される	
21, ...	29.0	パス／ジオメトリ軸用固定フィード 1 を起動する	
	29.1	パス／ジオメトリ軸用固定フィード 2 を起動する	
	29.2	パス／ジオメトリ軸用固定フィード 3 を起動する	
	29.3	パス／ジオメトリ軸用固定フィード 4 を起動する	
軸／スピンドル別			
31, ...	0	フィード／スピンドルオーバライド	
31, ...	1.7	オーバライドアクティブ	
31, ...	3.2	マシン軸用固定フィード 1 を起動する	
	3.3	マシン軸用固定フィード 2 を起動する	
	3.4	マシン軸用固定フィード 3 を起動する	
	3.5	マシン軸用固定フィード 4 を起動する	
31, ...	4.3	フィードホールド／スピンドル停止	
31, ...	62.2	回転フィードレートがアクティブ	
31, ...	78 ～ 81	位置決め軸用 F ファンクション	
31, ...	83.1	プログラムされた速度が高すぎる	1.17 (S1)

■ 7.2 マシンデータ

番号	識別子	名称	参照
一般事項 ($MN_ ...)			
10704	DRYRUN_MASK	ドライランフィードを起動する	
10710	PROG_SD_RESET_SAVE_TAB[n]	更新される設定データ	1.10 (K1)
12000	OVR_AX_IS_GRAY_CODE	グレイコード化されたパスフィードオーバライドスイッチ	
12010	OVR_FACTOR_AX_SPEED[n]	軸フィードオーバライドスイッチの評価	
12020	OVR_FEED_IS_GRAY_CODE	グレイコード化されたパスフィードオーバライドスイッチ	
12030	OVR_FACTOR_FEEDRATE[n]	パスフィードオーバライドスイッチの評価	
12040	OVR_RAPID_IS_GRAY_CODE	グレイコード化された早送りオーバライドスイッチ	
12050	OVR_FACTOR_RAPID_TRA[n]	早送りオーバライドスイッチの評価	
12060	OVR_SPIND_IS_GRAY_CODE	グレイコード化されたスピンドルオーバライドスイッチ	
12070	OVR_FACTOR_SPIND_SPEED[n]	スピンドル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 (SW2 以降)	
12082	OVR_REFERENCE_IS_MIN_FEED	パスオーバライド用基準を決定する	
12100	OVR_FACTOR_LIMIT_BIN	バイナリコード化されたオーバライドスイッチ用	
12200	RUN_OVERRIDE_0	オーバライド 0 で移動する	FBMA
12202	PERMANENT_FEED[n]	直線軸用固定フィード	FBMA
12204	PERMANENT_ROT_AX_FEED[n]	回転軸用固定フィード	FBMA
12240	SCALING_SYSTEM_IS_METRIC	G2 メートル法	1.8 (G2)

番号	識別子	名称	参照
軸別 ($MA_ ...)			
30300	IS_ROT_AX	回転軸	2.12 (R2)
32000	MAX_AX_VELO	最高軸速度	1.8 (G2)
32060	POS_AX_VELO	位置決め速度用初期設定	1.10 (P2)
32300	MAX_AX_ACCEL	軸加速	1.4 (B2)
35100	SPIND_VELO_LIMIT	最高スピンドル速度	1.17 (S1)
35130	GEAR_STEP_MAX_VELO_LIMIT	ギアステップ最高速度	1.17 (S1)
35140	GEAR_STEP_MIN_VELO_LIMIT	ギアステップ最高速度	1.17 (S1)
35160	SPIND_EXTERN_VELO_LIMIT	PLC からのスピンドル速度リミット	1.17 (S1)
チャンネル別 ($machine code_ ...)			
20100	DIAMETER_AX_DEF	フェーシング軸ファンクション付きジオメトリ軸	1.14 (P1)
20172	COMPRESS_VELO_TOL	圧縮用パスフィードからの最大許容ずれ	
20150	GCODE_RESET_VALUES	G グループの初期設定	1.10 (K1)
20660	THREAD_AUTO_LIFTFASTANGLE	後退角度を決定する (ねじ切り); SW バージョン 4.3 以降には含まれない	1.10 (K1)
20750	ALLOW_GO_IN_G96	G96 での G0 ロジック	
21200	LIFTFAST_DIST	K1 輪郭からの高速後退用移動パス	1.10 (K1)
21220	MULTFEED_ASSIGN_FASTIN	「一つのブロック中の複数のフィード」への NCK I/Os の入力バイトの割当て (SW2 以降)	
21230	MULTFEED_STORE_MASK	「一つのブロック中の複数のフィード」ファンクションのために入力信号を保存する (SW2 以降)	
22240	AUXFU_F_SYNC_TYPE H2	F ファンクション用出力時間 H2	
22410	F_VALUES_ACTIVE_AFTER_RESET	リセット後，F ファンクションアクティブ	

■ 設定データ

番号	識別子	名称	参照
チャンネル別 ($SC_ ...)			
42000	THREAD_START_ANGLE K1	スレッド用開始角度	1.10 (K1)
42010	THREAD_RAMP_DISP K1	ねじ切り時のフィード軸のランインおよびランアウトパス K1	1.10 (K1)
42100	DRY_RUN_FEED Dry run feed V1	ドライランフィード	1.18 (V1)
42600	JOG_FEED_PER_REV_SOURCE	JOG モードで，回転するフレームが作用するジオメトリ軸用回転フィードレート	
43300	ASSIGN_FEED_PER_RES_SOURCE	位置決め軸用回転フィードレート	
軸別 ($SA_ ...)			
43210	SPIND_MIN_VELO_G25	スピンドル最高速度	1.17 (S1)
43220	SPIND_MAX_VELO_G26	スピンドル最高速度	1.17 (S1)
43230	SPIND_MAX_VELO_LIMS	G96 によるスピンドル速度リミット	1.17 (S1)

■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

1.19 工具補正 (W1)

1.19.1 概要

YS840DI を使って工具補正データの計算ができます。

1. 長さ補正
2. 半径補正
3. フレキシブルなツールオフセットメモリにツールデータを保存
   - 0 ～ 32000 までの T 番号を使用するツール識別
   - 最多 9 つの切削エッジを使用するツールの定義
   - 25 までのツールパラメータタイプによって記述される切削エッジ

     ジオメトリ：長さ　　摩耗：長さ　　基本／アダプタ寸法

     ジオメトリ：半径　　摩耗：半径

     テクノロジ
4. ツール選択の選択方法：即時または選択可能な M 機能を介して
5. 工具径補正
   - 選択および選択解除ストラテジの構成方法：ノーマルまたは輪郭に関連して
   - 全ての補間タイプにアクティブな補正 :

     直線

     円

     螺旋

     スプライン

     多項式
   - 外側コーナでの補正の選択方法：遷移円／楕円 (G450) または等距離インタセクション (G451)
   - 輪郭に対する G450/G451 機能 のパラメータ起動適応
   - G450 と DISC パラメータを使った外側コーナのフリー移動
   - 補正面で選択可能な軸動作なしダミーブロック数
   - 選択可能な衝突監視：

     次の場合輪郭違反が予測されます：

     パス距離が工具半径より短い

     内側コーナの幅がツール直径より小さい
   - 工具径補正を一定に保つ（SW4 以降）
   - 多項式用のインタセクション処理（SW4 以降）

向き調整可能なツールホルダ (SW4 以降)

SW リリース 4 現在，使用可能なこの機能を使用して，工具長補正の公差で傾斜面の加工が可能です。ただし，ツールホルダ (NC 軸なし）のキネマティックスを使ってツールの静的向き調整が認められる場合です。さらに複雑な 5 軸変換はできません。

参照： 3.1 3 ～ 5 軸刃先方向制御（F2）

ツールデータとツールホルダデータを適切に選択すると，工具長補正を考慮に入れることができるキネマティックスを記述します。制御は，現在のフレームから直接いくつかの記述データを得ることができます。

（注）ツールと工具補正，および工具補正 （TLC と TRC）に関する一般的／個別的プログラミング機能の技術的記述の詳細情報は，下記の文書を参照してください。

参照： YS840DI ユーザーズマニュアル・プログラミング編

フラット／固有 D 番号構造

SW4 以降，固有の D 番号を使った簡単なオフセット選択は，管理機能なしに基本バージョンで使用可能です。

SW5 以降，固有の D 番号を介したオフセット選択は，管理機能を使って利用可能です（1.19.2「■ツール管理または「フラット D 番号」」参照)。

工具補正特殊処理（SW5 以降）

設定データ SD 42900 ～ SD 42950 は，ツール長とツール摩耗の符号の評価を制御するのに使用できます。同じデータが，ジオメトリ軸をミラーリングしたり，設定データを介して加工面を変更する時，摩耗コンポーネントの応答に適用されます。

参照： YS840DI ユーザーズマニュアル・プログラミング編

G461/G462 （SW5 以降）

工具径補正の起動および起動解除を使って，いくつかの状況で内側コーナのソリッド加工ができるように，コマンド G461 と G462 が導入されており，従ってアプローチ／後退ストラテジが工具径補正のために拡張しています。

G461：最後の TRC とその前のブロックの間のインタセクションが可能でない場合，制御は，その中心点が未修正ブロックの終点と一致し，半径が工具半径に等しい円を使って，このブロックのオフセットカーブを拡張してインタセクションを計算します。

G462：最後の TRC ブロックとその前のブロックの間のインタセクションが可能でない場合，制御は，工具径補正を使って最後のブロックの終点で直線を挿入してインタセクションを計算します（ブロックはそのエンド接線によって拡張されます)。

G40 から G41/42 への変更（SW 5 以降）

G40 から G41/G42 への変更またはその反対は，関連するツール点方向を持つツールのツールチェンジとしては処理されません（旋削ツールと研削ツール）。

1.19.2 詳細説明

■ ツール

ツール選択

T 機能は，プログラム内でツールを選択するのに使用されます。MD 22550:TOOL_CHANGE_MODE（M 機能を使った新たなツールオフセット）の設定は，新しいツールが T 機能の実行時即時ロードされるかどうかを確定します。

即時ツールチェンジ

TOOL_CHANGE_MODE = 0

新しいツールが T 機能の実行時即時ロードされます。この設定は主にツールタレット付き旋削マシンに使用されます。

"M06" との変化

TOOL_CHANGE_MODE = 1

新しいツールが T 機能を使ってロード準備されます。この設定は，加工プロセスを中断することなく新しいツールをツールチェンジ位置へ配置するために，主にツールマガジン付きフライス盤に使用されます。MD 22560 : TOOL_CHANGE_M_CODE （ツールチェンジ用 M 機能）は，スピンドルから古いツールを除去し，スピンドルに新しいツールをロードするために使用されます。このツールチェンジは，DIN 66025 に従って M 機能 M06 を使ってプログラムされる必要があります。

次のツールは，MD 20121:TOOL_PRESEL_RESET_VALUE を使って事前選択されます。この場合，MD 20110: RESET_MODE_MASK に従って，RESET とスタートアップの工具長補正値を考慮に入れなければなりません。

T の値範囲

T 機能は整数を受入れます。

- T0 から（ツールなし）
- T32000 まで（ツール番号 32000)

ツール切削エッジ

各ツールは 9 つまでの切削エッジを持つことができます。9 つのツール切削エッジは，D 機能 D1 ～ D9 まで割当てられます。

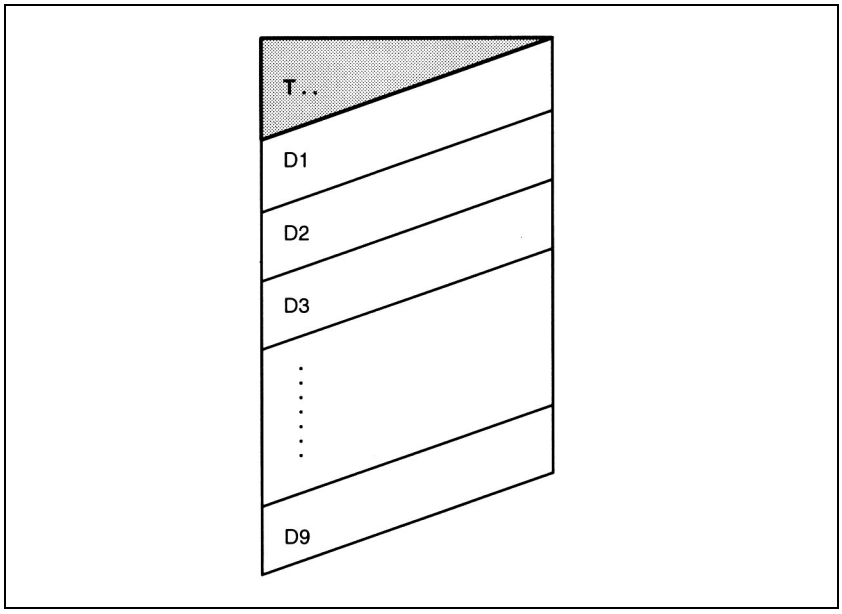


図 1.173 9 つの切削エッジ（D1 ～ D9 まで）を使ったツール T...

D ファンクション

ツール切削エッジは，D1（エッジ 1）から D9（エッジ 9) を使ってプログラムされます。ツール切削エッジは常に，現在アクティブなツールを参照します。アクティブツールのない (T0) のアクティブツール切削エッジ（D1 ～ D9）は非アクティブです。ツール切削エッジ D0 は，アクティブツールのツールオフセットを全て選択解除します。

ツールチェンジ時の切削エッジの選択

新しいツール（新しい T 番号）がプログラムされ，古いツールが交換される場合，切削エッジの選択に次のオプションが使用できます。

1. 切削エッジ番号がプログラムされる
2. 切削エッジ番号が MD 20270 :CUTTING_EDGE_DEFAULT によって指定される
   - = 0　M06 の後，切削エッジの自動選択なし
   - <>0　M06 の後に選択される切削エッジの番号
   - = -1　古いツールの切削エッジ番号が保存され，後で新しいツール用にも選択され，そして M06 の後にも新しいツール用に選択される。

工具補正の起動

D1 から D9 はアクティブツールの切削エッジに対する工具補正を起動します。工具長補正と工具径補正は，異なる時点で起動できます。

- 工具長補正 (TLC) は，TLC が動作する軸の最初の移動動作時に実行されます。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。
- 工具径補正 (TRC) は，G41/G42 がアクティブ面 （G17，G18 または G19) にプログラムされる時アクティブになります。G41/G42 を使った工具径補正の選択は，G0（早送り）または G1（直線補間）を使ったプログラムブロックでのみ可能です。

■ オフセットメモリ構造

ツールオフセットメモリサイズ

各チャンネルは，専用のオフセットメモリ (TO ユニット）を持つことができます。MD 28085: MM_LINK_TOA_UNIT（チャンネルへの TO ユニットの割当て）は，どの TO メモリが適切なチャンネルに存在するかを定義するために設定されます。NCK が管理する全てのツールのツール切削エッジの最大数は，MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA (NCK のツール切削エッジ番号）を使って設定されます。

ツール

ツールオフセットメモリは，番号 T1 ～ T32000 を持つツールで構成されます。各ツールは，TOA を介して，または個々に「新しいツール」ソフトキーを使って設定できます。必要でないオフセットには値ゼロを割当ててください（これはオフセットメモリが生成されない場合の初期設定です)。オフセットメモリの個々の値（ツールパラメータ）は，システム変数を使ってプログラムから読取られ書込まれます。

（注）ツール（T1 ～ T32000) は，ツールオフセットメモリに昇順または連続して保存される必要はなく，最初のツールが番号 T1 に割当てられる必要はありません。

ツール切削エッジ

各ツールは 9 つまでの切削エッジ （D1 ～ D9) を持つことができます。最初の切削エッジ (D1) は，新しいツールがツールオフセットメモリにロードされるとき自動的にセットアップされます。他の切削エッジ（8 つまで）は，「新しい切削エッジ」ソフトキーを使って連続的に隣接してセットアップされます。異なる番号のツール切削エッジがこの方法で各ツールに割当てられます。

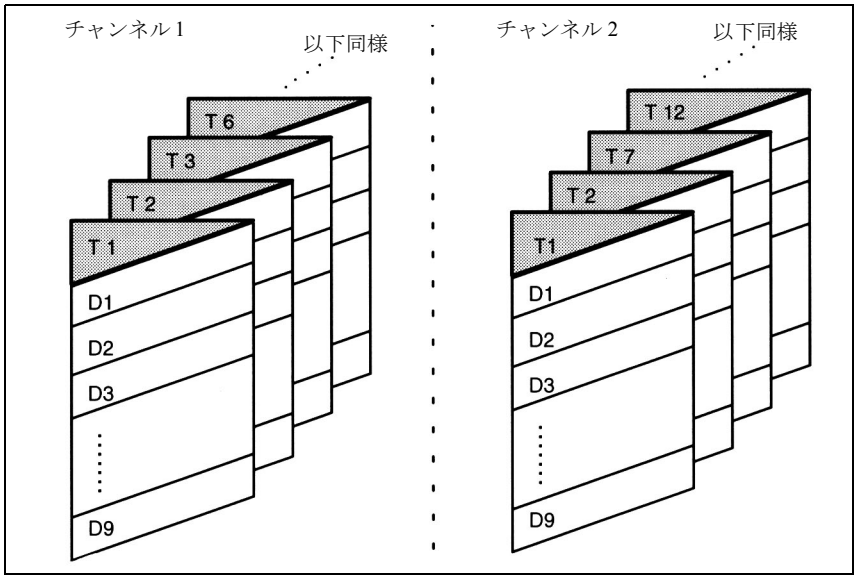
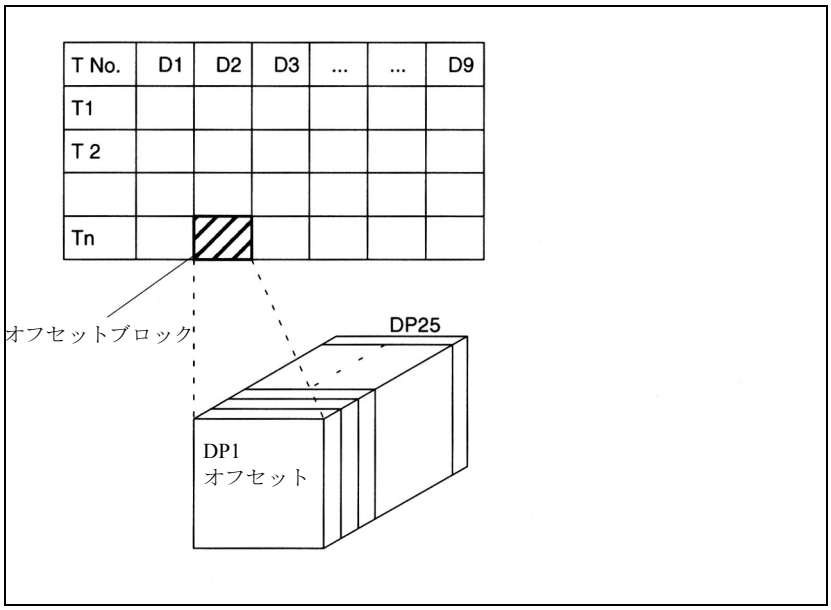


図 1.174 2 チャンネルに対するツールオフセットメモリ構造の例

■ ツールオフセットの計算

SW3.x およびそれ以前

D 番号と T 番号がツールオフセットの計算に必要です。



例 :

上記のオフセットは NC 内で計算されます :

パートプログラム内での呼出し

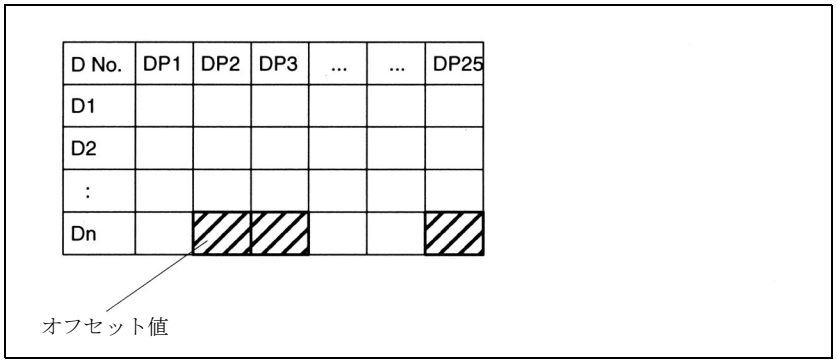
...

Tn D2

...

SW 4 以降（同様に可能）

D 番号は工具補正値を計算するのに十分です（MD 別に設定できる）。



例 :

上記のオフセットは NC 内で計算されます :

パートプログラム内での呼出し

...

D_n

■ NC アドレス T および M のアドレス拡張（SW5 以降）

SW5 以降，ユーザはマシンデータ T_M_ADDRESS_EXT_IS_SPINO を使って，ツール管理が起動しない時でさえ，T と M のアドレス拡張がスピンドル番号として解釈されるかどうか定義できます。すると同じルールが，「ツール管理」機能がアクティブである時のように，D 番号と T 番号の間の参照に適用されます。

D 番号への影響

オフセットデータセットは D 番号によって確定されます。

D アドレスはアドレス拡張を使ってプログラムできません。

D アドレスの評価は，常に，現在アクティブであるツールを参照します。
T_M_ADDRESS_EXT_IS_SPINO = TRUE の場合，プログラムされた D アドレスは，マスタスピンドルに関連するアクティブツールを参照します（ツール管理機能の場合と同じ）。

T 番号への影響

「ツール管理」機能がアクティブである場合，マスタスピンドル（またはマスタツールホルダ）に関してプログラムされた値は，プログラム済み／アクティブ番号 T として表示されます。

ツール管理がアクティブでない場合，プログラムされた全ての値 T は，どのアドレス拡張がプログラムされたかに関係なく，プログラム済み／アクティブとして表示されます。

T_M_ADDRESS_EXT_IS_SPINO = TRUE（アドレス拡張としてのスピンドル番号）の場合，マスタスピンドルに関してプログラムされた値 T のみが，プログラム済み／アクティブとして表示されます。

例

次の例はマシンデータの影響を説明します。

2 つのスピンドルが使用されます。スピンドル 1 はマスタスピンドルです。

M6 がツールチェンジ信号として定義されました。

T1 = 5

M1= 6

T2 = 50

M2 = 6

D4

- ツール管理がアクティブである時，D4 はツール "5" を参照します。
  T2=50 は，パス補正に影響しない補助スピンドルへツールを割当てます。パスは，マスタスピンドル用にプログラムされたツールによって排他的に決定されます。
- ツール管理が非アクティブで T_M_ADDRESS_EXT_IS_SPINO =FALSE である場合，D4 はツール "50" を参照します。T および M のどちらのアドレス拡張も NCK で評価されません。各ツールチェンジコマンドは，新しいパス修正を定義します。

- ツール管理がアクティブでなく，T_M_ADDRESS_EXT_IS_SPINO = TRUE の場合，D4 はツール "5" を参照します（ツール管理がアクティブである時と同じ）。
  アドレス拡張 1 (T1=..., M1=...) はマスタスピンドルのアドレスを指定します。

  （注）以前は，ツール管理が起動しない時，各ツールチェンジコマンド (T または M でプログラム済み) を使用すると，ツールオフセットがパス内で再計算されていました。アドレス拡張はこのオペレーションでこれ以上定義されません。拡張の意味は，ユーザによって（PLC ユーザプログラム内で）定義されます。

■ フリー D 番号割当て（SW5 以降）

NCK では，D 番号をツールオフセットデータセットの「相対」D 番号として管理できます。対応する D 番号は各 T 番号に割当てられます。D 番号の最大数は以前は 9 に制限されていました。

機能 （SW 5 以降）

SW 5 以降，D 番号を割当てる機能は大幅に拡張しています。

- 許される D 番号の最大数はマシンデータによって定義されます (MD 18105: MM_MAX_CUTTING_EDGE_NO)。デフォルト値は，既存のアプリケーションと互換性を維持するために 9 です。
- ツール別の切削エッジ数（またはオフセットデータセット）もマシンデータ (MD 18106: MM_MAX_CUTTING_EDGE_PERTOOL) によって定義できます。これを使ってユーザは，監視目的に，切削エッジ数が各ツールに対して実際の切削エッジの実際数に構成されるようにカスタマイズできます。
- NCK では，D 番号の名称変更が可能で，したがってどの D 番号でも切削エッジへ割当てることが可能です。

（注）

相対 D 番号の割振りに加えて，D 番号は，（「フラット D 番号構造」機能内の）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) として割当てることもできます。

切削エッジ番号 CE

D 番号を名称変更すると，これらの切削エッジに定義された番号を記載したツールカタログの情報が失われます。したがって名称変更のオペレーション後，カタログのどの切削エッジが参照されるかを決定することができません。

この情報は再ツーリング手順に必要なので，切削エッジ番号 CE が各切削エッジに導入されました。この番号は，D 番号が名称変更される時そのまま保存されます。

D 番号は，パートプログラムの切削エッジオフセットを識別します。SW5 以降，このオフセット番号 D は，切削エッジ番号 CE（ツールカタログ内の番号）と別に管理されます。どの番号でも使用できます。番号は，パートプログラムとディスプレイのオフセットを識別するために使用されます。

再ツーリング中，CE 番号は実際の物理的切削エッジを識別します。ツールチェンジ中，切削エッジ番号 CE は，オフセット選択時 NCK によって評価されません (OPI を介してのみ使用可能)。切削エッジ番号は，システムパラメータ $TC_DPCE[t,d] を使って定義されます。

- t は内部 T 番号を表します。
- d は D 番号を表します。

書込みアクセスは，衝突がないかどうか監視されます。つまりツールの全ての切削エッジ番号は異なっていなければなりません。パラメータ $TC_DPCE は，切削エッジパラメータデータセット $TC_DP1,..., $TC_DP25 のコンポーネントです。

最大切削エッジ数 (MD 18105) が，ツール (MD 18106) 別の最大切削エッジ数より大きい場合，$TC_DPCE をパラメータ化することのみが実用的です。

この場合，デフォルト切削エッジ番号は切削エッジの分類番号と同じです。ツー

ルのオフセットは番号 1 から始まって生成され，ツール別の最大切削エッジ数までインクリメントされます (MD 18106: MM_MAX_CUTTING_EDGE_PER_TOOL)。
MM_MAX_CUTTING_EDGE_NO <=
MM_MAX_CUTTING_EDGE_PER_TOOL の場合，CE 切削エッジ番号は D 番号と等しくなります（前回応答に一致)。読取りオペレーションは CE=D を戻します。書込みオペレーションはアラームメッセージなしで無視されます。

（注）アクティブツールオフセットのオフセット値 $TC_DP1,...,
$TC_DP25 は，n=1,...,25 のパラメータ $P_AD[n] を使って読取られます。アクティブオフセットの CE 切削エッジ数は，
n=26 を使って戻されます。

コマンド

最大切削エッジ数 (MD 18105) が，ツール別の切削エッジ数 (MD 18106) より大きい場合，次のコマンドが使用できます。

- CHKDNO - 既存の D 番号が固有かどうか確認します；TO ユニット内に定義された全ツールの D 番号は一度以上発生しないかもしれません。交換ツールに公差は考慮されません。
- GETDNO - ツールの切削エッジに対する D 番号を取得します。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 より大きい値が戻されます。
- SETDNO - ツールの切削エッジ CE の D 番号を設定または変更します；入力パラメータに一致するデータセットが存在しない場合，FALSE が戻されます。構文エラーはアラームを生成します。D 番号は 0 に設定できません。
- GETACTTD - 絶対 D 番号に対応付けられた T 番号を取得します；システムは，番号が固有かどうか確認しません。TO ユニット内の D 番号のいくつかが同じであれば，検索で見つかった最初のツールの T 番号が戻されます。このコマンドは「フラット」D 番号を使った使用に適しません。なぜならこの場合（データベースに T 番号がない)，常に値 1 が戻されるからです。
- DZERO - TO ユニットの全ての D 番号に無効としてタグを付けます；このコマンドは，再ツーリング中サポートに使用されます。このコマンドでタグを付けられたオフセットデータセットは，CHKDNO 言語のコマンドによっては検証されません。これらのデータセットは，再度 SETDNO を使って D 番号を設定することで再びアクセスできます。

（注）最大切削エッジ数が，ツール別の最大切削エッジ数より少ない (MM_MAX_CUTTING_EDGE_NO <
MM_MAX_CUTTING_EDGE_PERTOOL) 場合，定義された言語コマンドはシステムに影響しません。
既存のアプリケーションが引き続き動作するために，この関係が NCK にプリセットしています。
個別コマンドは，プログラミングガイドに詳しく述べられています。

参照： YS840DI ユーザーズマニュアル・プログラミング編

起動

固有 D 番号および定義された言語コマンドを使って加工するには，ツールに自由に D 番号を命名できなければなりません。この目的で次の条件が満たされなければなりません。

- マシンデータ MD 18105: MM_MAX_CUTTING_EDGE_NO が，MD 18106: MM_MAX_CUTTING_EDGE_PERTOOL より大きくなければならない。
- 「フラット D 番号」機能は起動しない。
  (-> MD 18102: MM_TYPE_OF_CUTTING_EDGE).

例

MM_MAX_CUTTING_EDGE_NO = 1

1 つのオフセットの最大値がツール別に定義できます（D 番号 = 1 で）。

（注）
「フラット D 番号」機能がアクティブである時，1 つのみの D オフセットを TO ユニットに定義できます。

MM_MAX_CUTTING_EDGE_NO = 9999

ツールは固有の D 番号を割当てられることができます。

例

- D 番号 1，2，3 は T 番号 1 へ割当てられます。
- D 番号 10，20，30，40，50 は T 番号 2 へ割当てられます。
- D 番号 100，200 は T 番号 3 へ割当てられます。
- 以下同様

CHKDNO; MAX_CUTTING_EDGE_NO = 9999

次のデータは固有の D 番号かどうかチェックされます。

- D 番号 1，2，3 を持つ T 番号 1
- D 番号 10，20，30，40，50 を持つ T 番号 2
- D 番号 100，200，30 を持つ T 番号 3
  （定義上のタイプミス：30 が 300 の代わりに入力された）

CHKDNO	D=30 が二度存在するので，上記の配列がチェックされる時 FALSE 状態が戻されます。
CHKDNO(2，3，30)	D=30 が二度存在するので，指定された D 番号がチェックされる時 FALSE 状態が戻されます。
CHKDNO(2，3，100)	D=100 が一度存在するだけなので TRUE が戻されます。
CHKDNO(1，3)	三番目のツール D=30 と二番目のツール D=30 の間に対立がありますが，TRUE 状態が戻されます。

MM_MAX_CUTTING_EDGE_PERTOOL = 1

1 つの切削エッジを持つツールのみが使用されます。マシンデータの値 1 は，1 つ

のツールに対する二番目の切削エッジの定義を禁止します。

MM_MAX_CUTTING_EDGE_PERTOOL = 12

12 までの切削エッジが 1 つのツールに定義できます。

(ツール別に 9 つの切削エッジが SW4 まで認められました)。

サンプルプログラム

D 番号の名称変更

切削エッジ CE = 3 の D 番号は 2 ～ 17 まで名称変更できます。次の仕様が適用されます。

- 内部 T 番号 T = 1
- D 番号 = 2
- 1 つの切削エッジを持つツール
  $TC_DP2[1, 2] = 120
  $TC_DP3[1, 2] = 5.5
  $TC_DPCE[1, 2] = 3 ：切削エッジ番号 CE
- MM_MAX_CUTTING_EDGE_NO = 20

パートプログラム内で，このオフセットは T1,....D2 を使って標準としてプログラムされます。

ユーザは切削エッジ 3 の D 番号を変数 (DNoOld) へ割当て，新しい D 番号に変数 DNoNew を定義します。

```
def int DNoOld, DNoNew = 17
DNoOld = GETDNO( 1, 3 )
SETDNO( 1, 3, DNoNew )
```

新しい D 値 17 は，その後切削エッジ CE=3 へ割当てられます。

この切削エッジのデータは，システムパラメータを介して，およびプログラムで NC アドレス D を使って，現在 D 番号 17 でアドレス指定されます。

このオフセットは，現在，T1,....D17 を使ってパートプログラムにプログラムされ，データは次のようにアドレス指定されます。

```
$TC_DP2[ 1, 17 ] = 120
$TC_DP3[ 1, 17 ] = 5.5
$TC_DPCE[ 1, 17 ] = 3：切削エッジ番号 CE
```

（注）もう 1 つの切削エッジがツールに定義されている場合：例えば
$TC_DPCE[1, 2] = 1 ; = CE,
切削エッジ 1 の D 番号 2 は，切削エッジ 3 の D 番号と同じ名称でアドレス指定できません：つまり SETDNO(1, 3, 17) は FALSE を戻します。

DZERO - D 番号を無効にする

このコマンドを起動すると，TO ユニットのツールの D 番号を全て無効にします。有効な D 番号が再度 NCK で使用できるまでオフセットを起動できません。コマン

ド SETDNO を使って D 番号を再度割当ててください。

次のツールが定義されなければなりません（全て切削エッジ番号 1 を使って）:

T1, D1　　切削エッジ CE=1 の D 番号

T2, D10　　切削エッジ　CE=1 の D 番号

T3, D100　　D 切削エッジ CE=1 の D 番号

その後次のコマンドがプログラムされます :

DZERO

オフセットの 1 つが現在起動される場合（例えば T3 D100 で），D100 は現在定義されていないのでアラームが生成されます。

D 番号は下記を使って再定義されます。

SETDNO(1, 1, 100);T=1，切削エッジ 1 は（新しい）D 番号 100 を割当てられる

SETDNO(2, 1, 10); T=2，切削エッジ 1 は（古い）D 番号 10 を割当てられる

SETDNO(3, 1, 1);　T=3，切削エッジ 1 は（新しい）D 番号 1 を割当てられる

（注）電源異常の場合，DZERO コマンドは NCK を D 番号に関して無定義状態のままにします。このような場合，正常に戻った時 DZERO コマンドを繰返してください。

再ツーリングプログラムのオペレーション原理

タスクは必要なツールと切削エッジを確実に使用可能にすることです。NCK のマガジンとツールメモリは，どんな状態でもあり得ます。新たな加工オペレーション用のパートプログラム内の D 番号は，一般的に実際の切削エッジの D 番号と一致しません。再ツーリングプログラムは次の表示ができます。

DZERO　；TO ユニットの全ての D 番号が無効としてタグが付けられる

....　；ツールとその切削エッジ番号をチェックするためのマガジン位置上の 1 つまたは複数のループ。
まだディスエーブルされていない ($TC-TP8) 望ましい切削エッジ番号 CE (GETDNO) を持つツールが見つかった場合，新しい D 番号が切削エッジ(SETDNO) へ割振りされる。

....　；負荷と無負荷運転が実行される。
ツールステータス「無負荷」および「負荷」を使って作業できる。

CHKDNO　；負荷／無負荷および D 番号の名称変更オペレーションが完了する。
個々のツールまたは D 番号あるいはその両方がチェックされ，衝突が戻り値に従って自動的に処理される。

■ ツールチェンジ中のエラーの修正ブロック（SW5 以降）

ツール準備がパートプログラムにプログラムされた後，NCK がエラーを検出する（例えば，プログラムされた T 番号のデータセットが NCK に存在しない）場合，加工プログラムはアラームメッセージで終了します（SW4 まで）。

SW5 以降，ユーザは，エラー状況（パートプログラムのプログラミングエラー，ツールデータセットが NCK に存在しない）を確認し，適切な対策を開始し，加工を続けることができます。

ツールチェンジは，マシンデータ MD 22550: TOOL_CHANGE_MODE 次第で様々な方法でプログラムできます。

TOOL_CHANGE_MODE = 0

T=「T 番号」；1 つの NC ブロック内のツール準備とツールチェンジ
つまり，T がプログラムされる時，新しい D オフセットが起動する
NCK で（参照：マシンデータ MD 20270:
CUTTING_EDGE_DEFAULT）

TOOL_CHANGE_MODE = 1

T= 「T 番号」；ツール準備

M06 ；ツールチェンジ
（ツールチェンジ M コードの番号が設定することができる）
つまり，M06 がプログラムされると，新しい D オフセットが起動する
NCK で（参照 : マシンデータ MD 20270:
CUTTING_EDGE_DEFAULT）

ツール管理がアクティブでない場合，次の問題が生じる場合があります。

- D オフセットデータセットの欠落
- パートプログラムのエラー

（注）「マガジンにツールがない」という問題は検出できません。
なぜなら NCK は，工具補正中マガジン情報に全くアクセスしていないからです。

D オフセットデータセット欠落

プログラム実行は，無効 D 値（マシンデータ MD 22550 値から独立）を含むブロックで中断されます。オペレータはプログラムを修正するか，または欠落したデータセットを再ロードしてください。

このためには，D 番号機能に D 番号が必要かまたは T 番号も同様に必要です。これらのパラメータは，起動されたアラーム 17181（17180 と同じ機能）と共に渡されます。

パートプログラムのエラー

エラーが発生した場合の介入オプションは，ツールチェンジがプログラム（マシンデータ MD 22550: TOOL_CHANGE_MODE を介して定義）された方法に依存します。

T プログラミングを使ったツールチェンジ (TOOL_CHANGE_MODE = 0)

この場合，NCK で利用可能な「修正ブロック」機能が使われます。NC プログラムは，プログラムされた T 値が検出された NC ブロックで停止します。「修正ブロック」は，プログラムが再開されると再度実行されます。

オペレータは次のように介入できます。

- パートプログラムを修正する。
- MMC から，欠落切削エッジオフセットデータを再ロードする。
- 「オーバストア」によって欠落切削エッジオフセットデータを NCK に含める。

オペレータの処置後，START キーを押すとエラーを起こしたブロックが再度実行されます。エラーが修正されるとプログラムは継続されます。そうでない場合，アラームが再び出力されます。

T および M06 プログラミング (TOOL_CHANGE_MODE = 1) を使ったツールチェンジ

この場合，エラーはツール準備（T プログラミング）を含む C ブロックで検出されますが，このエラーは最初無視できます。処理は，ツールチェンジリクエスト（通常 M06）が実行されるまで続きます。プログラムはこの時点で停止されなければなりません。

プログラムされた T アドレスは，M06 コマンドの前に何行でもプログラム行を含むことができるか，または 2 つの指令が異なる（サブ）プログラムに表示できます。この理由で，既に実行されたブロックまたは訂正ブロックの修正はできません。

オペレータは，TOOL_CANGE_MODE = 0 に関して同じ介入オプションを持ちます。欠落データを再ロードできます。しかしこの場合，T は「オーバストア」でプログラムされなければなりません。

プログラムエラーが発生した場合，エラーのある行は修正できません (Txx)。プログラムが停止しアラームを生成した行のみが編集できます。

シーケンスは次の通りです。

```
Txx     ; エラー！ xx を持つデータセットが存在しない
        ; ステータスを保存する ; xx を保存する
        ; プログラムを継続する
....
M06     ;「xx 欠落」を検出 -> アラーム出力
        ; プログラムを停止する
        ; ブロックを修正する。例えば Tyy M06 を使って開始
        ; ブロック Tyy M06 は正しく解釈される
```

; 処理を継続する

21 このプログラム：点が再度実行される時次のことが生じます。

```
Txx       ; エラー！ xx のデータセットが存在しない
          ; ステータスを保存する ; xx を保存する
          ; プログラムを継続する
....
Tyy M06   ; フラグ「xx 欠落」を検出 -> さらなるリアクションなしに拒否する,
          ; なぜなら Tyy M06 は正しい -> プログラムは停止しない（正しい)。
```

必要であれば，T 呼出しの起点は，プログラムの終了後修正できます。マシンのツールチェンジ論理がこれを処理できない場合，プログラムを強制終了し，エラー点を修正してください。

1 つのみのデータセットが欠落している場合，NCK へ伝達され，Txx が「オーバストア」にプログラムされ，プログラムは続いて再開されます。

「欠落 D 番号」の場合のように，「欠落 T 番号」に必要なパラメータ（T 番号）は，適切なアラーム (17191) を介してユーザがアクセスできます。

■ フラット D 番号構造（SW4 以降）

SW4 以降では，D 番号を使った簡単なツール管理が可能です（交換ツールなし，マガジンなし）。

この機能はツール管理の基本レベル（ツール管理機能の起動なし）で使用でき，旋削マシン用に意図されています。研削ツールはこの機能を使って定義できません。

起動

MD18102: MM_TYPE_OF_CUTTING_EDGE は，アクティブな D 番号のタイプを定義するために使用できます。

- 値 = 0 = 以前と同様 = デフォルト設定
- 値 = 1 = 絶対直接 D プログラミングを使ったフラット D 番号構造
- 値 = 2 = 相対間接 D プログラミングを使ったフラット D 番号構造

切削エッジは，PI コマンドまたは NC プログラミングコマンドを介して，個々に削除できます。

指定番号を持つ切削エッジも，MMC を使って選択的に生成できます。

■ 新しい D 番号を生成する（修正ブロック）

工具補正は，システムパラメータ $TC_DP1... $TC_DP25 を使ってプログラムできます。内容は以前と同じ意味を持ちます。

構文変更：T 番号は指定されません。

- 機能「フラット D 番号」がアクティブ

  $TC_DPx[d] = 値：　　x= パラメータ番号である場合，d=D 番号

  つまりこの構文を持つデータのみが，「フラット D 番号」機能が起動された時，NCK へロードできます。

- 「フラット D 番号」機能が非アクティブ

  $TC_DPx[t][d] = 値：　t=T 番号を使って，d=D 番号

D 番号は各ツールに一度だけ割当てられることができます。つまり D 番号は正確に 1 つのオフセットデータブロックを表します。

存在しない D 番号が初めて生成されると，新しいデータブロックが NCK メモリに保存されます。

MD 18100 MM_NUM_CUTTING_EDGES_IN_TOA は，D の最大数またはオフセットデータブロック（最大 600）を設定するために使用されます。

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実行されます。つまり，「フラット D 番号」機能が生成されるバックアップファイルは，機能を起動していない制御の NCK へ入力できません。同じことが転送のための反転にも適用されます。

D 番号範囲

1 ～ 99 999 999

■ PLC から D 番号を読取る

プログラマは，パートプログラムに D 番号を指定する様々なオプションを使用できます。

1 つのオプションは，システムパラメータ $A_DNO[n] (n=1,...9) から D 番号を読取ることです（"DNO" は D 番号を表します)。

前提条件

MD 18102 MM_TYPE_OF_CUTTING_EDGE = 2

パラメータは，VDI（PLC と NC 間のインタフェース）を介して PLC から絶対 D 番号を取得します。

例

パートプログラムでのオフセット選択

D9 がプログラムに書込まれます。$A_DND[9] を使って，呼出し時点で 9 番目のテーブル位置に保存された絶対 D 番号が読取られます。

表 1.79 システムパラメータ $A_DNO[n]; (n=1,...9)

意味	パラメータは，D 番号を含む n 番目の位置の値を読取る。 パラメータは，通常，間接 D プログラミングに使用される（パラメータを介して）: D=$A_DNO[1] パラメータは，VDI の D 番号テーブルにあるフィールドエレメント（= 具体的な D 番号）を読取る。PLC は，NCK が補助機能 T を PLC へ出力した後，この T 番号に対するマッチング D 番号をフィールドへ書込む。 それらに対して D 番号が定義されていない場合，フィールドエレメントは値ゼロを持つ。 PLC は，1 T 番号に対する最大 9 つの異なる D 番号を PLC によって VDI インタフェースに保存することができる。D 番号自体が 0 ～ 3200 の間の値を取ることができる。つまりテーブルはギャップを持つ場合もある。
データタイプ	INT
値の範囲	1 ～ 99 999 999
索引	n=1 ~ 9 索引 n は読取られる D 番号テーブル内の位置を示す。
アクセス	パートプログラムで読取る
前処理停止	可能
同期化動作	同期化動作で読取る
妥当性	「フラット D 番号」または PLC のツール管理と結合してのみ適切である。

■ D 番号プログラミング

MD 18102 は，D 番号のプログラミング方法を定義するのに使用できます。

D0 は，以前の意味「NCK のアクティブオフセットの選択解除」を有します。

D のアドレスを拡張する

D のアドレス拡張はできません。1 つのアクティブオフセットデータブロックのみが，所定時ツールパスに対して可能です。

直接，絶対プログラミング

パートプログラム内のプログラミングは，以前と同じように実行されます。プログラムされた D 番号の値範囲のみが増大します。

例 1:

MD 22550 TOOL_CHANGE_MODE = 0

MD 18102 MM_TYPE_OF_CUTTING_EDGE = 1

MD 20270 CUTTING_EDGE_DEFAULT = -1

...

D92

X0　　D92 のオフセット値で移動

T17　　PLC へ出力 T=17

X1　　D92 のオフセット値で移動

D16

X2　　D16 のオフセット値で移動

D32000

X3　　D32000 のオフセット値で移動

T29000　　PLC へ出力 T=29000

X4　　D32000 のオフセット値で移動

D1

X5　　D1 のオフセット値で移動

...

例 2:

MD 22550 = 0

T1

T2

T3

D777　　待機なし，D777 が起動される。

T3 = プログラムされている，ディスプレイのアクティブツール

D777 = プログラムされているアクティブオフセット

（注）・ツールチェンジおよび
・具体的なツールへの D オフセットの割当ては，NCK の責任ではありません。

間接パラメータ化プログラム

D=$A_DNO[n]

意味：VDI D 番号テーブルの位置 n（=1 ～ 9）で入力された D 番号を選択します。

パラメータ $A_DNO は，メインランの前に強制同期を取って準備段階中に読取られます。

VDI インタフェースで使用可能な D 番号は，$A_DND を使って読取られます。

（注）PLC は，NCK メインランが D 番号を読取る前に，以前プログラムされた T 値に一致する D 番号の書込みを確実に終えていなければなりません。（NCK で）このメカニズムは，NCK 内の PLC の一般待機を開始します。

間接索引プログラミング

Dn または D=n

意味：VDI D 番号テーブルの位置 n(=1 ～ 9）で入力された D 番号を選択します。この構文は従来のものと同じですが，索引 n に確定された D 番号用のオフセットブロックを内部的に起動します。

間接 D プログラミングシーケンスの例：

MD 20270 = 0 にします

MD 22550 = 0 にします

スピンドル no. 2 がマスタスピンドルでないようにします

パートプログラム	動作
T 4	ツールチェンジコマンド：NCK は T 補助機能として値 4 を出力する。PLC はそれを評価し，VDI の関連 D 番号を提供する。受取られたアドレス拡張がマスタスピンドルの番号である場合，PLC のみがこれを行う。単純通信プロトコルを使って，NCK は，関連 D 番号が後続のプログラムされた D に使用可能かどうか検出することができる。
X1	
...	
D1	NCK のフィードは，メインランと同期化され，新しい D 番号が VDI に存在するかどうかチェックし（必要な場合待つ），位置 1 に望ましい（絶対）D 番号を受入れる：例えば番号 4711。NCK は現在，オフセットブロック 4711 を決定しジオメトリを計算する。NC は，VDI インタフェースへ D 番号テーブルの完全な内容を伝達する（これは常に，ツールチェンジコマンドが検出される時起こる）。
D=$P_DNO[2]	内容は D2 のプログラミングに対応する（間接プログラミング）。
D3	NC は，記憶位置 3 の絶対 D 番号にアクセスする：新しい T がプログラムされていないので同期化動作は必要ではない。
T2 = 12000000	NC はツールチェンジを検出しない：なぜならスピンドル no.2 がマスタスピンドルの番号でないため。
D4	NC は記憶位置 4 の絶対 D 番号にアクセスする：新しい T がプログラムされていないので同期化動作は必要ではない。

T 番号を PLC へ出力する前に，NCK は待機される D 番号に関する VDI のステータスを保存します。カウンタがこの目的のために VDI に含まれています。PLC は，D 番号がリフレッシュされる都度カウンタを増分します。必要であれば，NCK からの次の D 番号リクエストは，カウンタ値が代わるまで待ちます。VDI には，以前プログラムされた値に属する新しい D 番号テーブルが入ります。



重要

D は，アドレス拡張なしにプログラムのみが可能です。D は常にマスタスピンドルを参照します。T はアドレス拡張を使ってプログラムできます。PLC はマスタスピンドルに関する T 値または M6 コマンドを使って，VDI へ D オフセットの書込みのみをしなければなりません。NCK は，VDI からの D 番号の読取り中，同期動作を実行する時これを事実と見なします。

例

```
MD 22550 $MC_TOOL_CHANGE_MODE = 1
MD 22560 $MC_TOOL_CHANGE_M_CODE = 6
                ; つまり，M6 プログラミングを使ったツールチェンジ
                  スピンドル番号 3 がマスタスピンドルでないようにします
T1
M6              ; ツールチェンジコマンド
T3
T3 = 11
D=2             ; VDI のカウンタが変わるまで待ちます：これは，PLC が T1 に対
                  する D 値を D 番号テーブルに書込み終えたことを示します（例え
                  ば記憶位置 2 = 4711：その後 NCK のオフセット 4711 を起動しま
                  す）
                  T3 = プログラムされた T,
                  T1 = ディスプレイの中のアクティブ T
```

パートプログラムを介した D 番号削除

- フラット D 番号を使って
  $TC_DP1[d] = 0
  オフセットブロックは NCK の番号 d で削除されます。
  メモリは以後別の D 番号の定義に開放されます。
- D 番号なし
  $TC_DP1[t][d] = 0 =
  ツールの切削エッジが削除されます。
- $TC_DP1[0] = 0
  NCK の全ての D オフセットを削除します。

アクティブデータブロック （D 番号）は削除できません。したがって，削除する前に D0 をプログラムする必要があります。

ツール MD

次のマシンデータは，ツールと切削エッジ（D 番号）が NCK で動作する方法に影響します。

- MD20270: CUTTING_EDGE_DEFAULT
- MD20130: CUTTING_EDGE_RESET_VALUE
- MD20120: TOOL_RESET_VALUE
- MD20121: TOOL_PRESEL_RESET_VALUE
- MD22550: TOOL_CHANGE_MODE
- MD22560: TOOL_CHANGE_M_CODE
- MD20110: RESET_MODE_MASK
- MD20112: START_MODE_MASK

MD 1802 = 2 の場合，MD 20270 と MD 20130 の値は，NC ブロックにプログラムされたように絶対 D 番号でなく相対 D 番号を参照します。

■ T 番号のプログラミング

機能「フラット D 番号構造」がアクティブである時，NC アドレス T は継続して評価されます。つまり，プログラムされた T 番号とアクティブ T 番号が表示されます。しかし，NC はプログラムされた T 値に関係なく D 番号を確定します。

NC は，チャンネルにつき 1 マスタスピンドルを認識します（スピンドル番号は MD を介して設定可能）。

オフセットと M6 コマンド（ツールチェンジ）は，マスタスピンドルに関してのみ計算されます。

アドレス拡張 T は，スピンドル番号として解釈されます（例えば T2 = 1：ツール 1 はスピンドル 2 に選択される)：ツールチェンジは，スピンドル 2 がマスタスピンドルの場合のみ検出されます。

■ M6 のプログラミング

MD 22550 は，ツールコマンドが M 機能で実行されるかどうかを定義し，T はツール準備コマンドとして使用されます。

MD 22560 は，ツールチェンジのための M 機能の名称を定義します。デフォルトは M6 です。

NC は，チャンネルごとに 1 マスタスピンドルを認識します（スピンドル番号は MD を介して設定可能）。

オフセットと M6 コマンド（ツールチェンジ）は，マスタスピンドルに関してのみ計算されます。

M6 のアドレス拡張はスピンドル番号として解釈されます。

例

スピンドル 1 とスピンドル 2 が定義されます。

次のようにしてください：

MD 20090 = 2，つまりスピンドル no. 2 はマスタスピンドル。

M6　　；ツールチェンジが望ましい時，コマンドは暗黙にマスタスピンドルを参照します。

M1=6　；スピンドル no.2 はマスタスピンドルなのでツールチェンジはありません。

M2=6　；スピンドル no.2 はマスタスピンドルなので，ツールチェンジが実行されます。

■ プログラムテスト

MD 20110，ビット 3 は，アクティブツールとツールオフセットが次のように伝達されるのを定義するために使用されます。

- (=1) 最後のテストプログラムからテストモードで終わる。
- (=0) 最後のプログラムからプログラムテストの起動前に終わる。

前提条件

MD 20110 のビット 0 と 6 がイネーブルされなければなりません。

■ ツール管理または「フラット D 番号」

ツール管理

NCK のアクティブツール管理は，次の仮定に基づいて作動します。

1. ツールはマガジンの中で管理されます。
2. 切削エッジが監視される：リミットに達すると，ツールはディスエーブルされます。
3. ツール交換の背景にある理念：ツールは，その識別子に基づいてのみ選択用にプログラムされます。NCK は，以後そのストラテジに従って特定のツールを選択します。

ツール管理の意義は，特定のツールが定義され，NCK に使用されて初めて意味があるということです。

フラット D 番号

フラット D 番号は，ツール管理が NCK の外部で実行され，T 番号に関係しないことを意味します。

ツール管理とフラット D 番号の非混合

NCK に対するツール管理の主な理由は時間の節約にあるので，NCK と PLC に関するツール管理機能を混合したり分散したりすることは意味がありません。これは，極めて時間に厳格なタスクが NCK で実行されて初めて成功します。

しかし，これは「フラット D 番号」には当てはまりません。

（注）「フラット D 番号構造」と「ツール管理」の両機能の起動が監視されます。両方が同時に起動されない場合，「ツール管理」が優先します。

■ ツール切削エッジ

ツール切削エッジ

次のデータは，ツール切削エッジを固有に記述するために使用されます。

- ツールタイプ（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

ツールパラメータ

ジオメトリの記述，テクノロジの記述およびツールタイプは，各ツールタイプのツールパラメータへマップされます。次のツールパラメータがこの目的で提供されます。

ツールパラメータ	意味	メモ	拡張用に予約済み
1	ツールタイプ		
2	切削エッジ位置	旋削ツールのみ	
ジオメトリ - 工具長補正			
3	長さ 1		
4	長さ 2		
5	長さ 3		
ジオメトリ - 工具径補正			
6	半径 1/ 長さ 1	3D 正面フライス用　1.19.2「■ジオメトリ ― 工具径補正」参照	
7	長さ 2	3D 正面フライス用	
8	半径 1	3D 正面フライス用	
9	半径 2	3D 正面フライス用	
10	角度 1	3D 正面フライス用	
11	角度 2	3D 正面フライス用	
摩耗 - 工具長補正			
12	長さ 1		
13	長さ 2		
14	長さ 3		
摩耗 - 工具径補正			
15	半径 1/ 長さ 1	3D 正面フライス用 1.19.2「■ジオメトリ ― 工具径補正」参照参照	
16	長さ 2	3D 正面フライス用	
17	半径 1	3D 正面フライス用	
18	半径 2	3D 正面フライス用	
19	角度 1	3D 正面フライス用	
20	角度 2	3D 正面フライス用	
ベース寸法／アダプタ寸法 - 工具長補正			
21	基本長さ 1		
22	基本長さ 2		

ツールパラメータ	意味	メモ	拡張用に予約済み
23	基本長さ 3		
テクノロジ			
24	自由切削角	旋削ツールのみ	
25			予約済み

予約済み ... このツールパラメータは使用されないという意味です（拡張用に予約済み）。

■ ツールタイプ（ツールパラメータ）

表 1.80 使用可能なツールタイプと最低パラメータ必要条件の表

ツールタイプ		タイプ	切削エッジ位置	ジオメトリ―長さ1	―長さ2	―長さ3	ジオメトリ―半径長さ1	―長さ2	―半径1	―半径2	―角度1	―角度2	摩耗長さ	摩耗半径	基本寸法―長さ1	―長さ2	―長さ3	フリ―切削角
フライス加工ツール																		
110	ボールエンドミル	x		x	x	x			x				x	x	x	x	x	
120	エンドミル	x		x					x				x	x	x	x	x	
121	面取り付きエンドミル	x		x					x				x	x	x	x	x	
130	アングルヘッドミル	x		x	x	x			x				x	x	x	x	x	
131	面取り付きアングルヘッドミル	x		x	x	x			x				x	x	x	x	x	
140	正面フライス	x		x					x				x	x	x	x	x	
145	スレッドミル	x		x					x				x	x	x	x	x	
150	サイドミル	x		x					x				x	x	x	x	x	
155	円錐台ミル	x		x	x	x			x				x	x	x	x	x	
ドリルツール																		
200	ドリル	x		x									x		x	x	x	
205	回転チップドリル	x		x									x		x	x	x	
210	ボーリングロッド	x		x									x		x	x	x	
220	センタドリル	x		x									x		x	x	x	
230	さらもみ	x		x									x		x	x	x	
231	もみ下げ	x		x									x		x	x	x	
240	タップ／粗スレッド	x		x									x		x	x	x	
241	タップ／微スレッド	x		x									x		x	x	x	
242	タップ／ Whitworth（ウィット）スレッド	x		x									x		x	x	x	
250	リーマ	x		x									x		x	x	x	
研削ツール																		
400		x		x	x	x			x				x		x	x	x	
から																		
499		x		x	x	x			x				x		x	x	x	
旋削ツール																		

ツールタイプ		タイプ	切削エッジ位置	ジオメトリ―長さ1	―長さ2	―長さ3	ジオメトリ―半径長さ1	―長さ2	―半径1	―半径2	―角度1	―角度2	摩耗長さ	摩耗半径	基本寸法―長さ1	―長さ2	―長さ3	フリ―切削角
500	荒削りバイト	x	x	x	x	x			x				x	x	x	x	x	x
510	仕上げ切削バイト	x	x	x	x	x			x				x	x	x	x	x	x
520	溝切りバイト	x	x	x	x	x			x				x	x	x	x	x	x
530	突切りバイト	x	x	x	x	x			x				x	x	x	x	x	x
540	スレッド切削バイト	x	x	x	x	x			x				x	x	x	x	x	x
特殊ツール																		
700	スロッティングソー	x		x	x	x							x		x	x	x	

意味

3 桁の番号がツールタイプの定義に使用されます。タイプ別にツールを分類する主な理由は，オペレータ／セットアップエンジニア／プログラマによる加工セクションへの正しいツールの選択を容易にすることです。ツールタイプの従属目的は，入力に必要な基本ツールパラメータを定義することです（表 1-87 参照）。

特別機能

- ツールタイプを各ツールの切削エッジに対して指定してください。
- ツールタイプは，定義された値が受入れられるものに限ります（表 1-87 参照）。
- ツールタイプ 0（ゼロ）は，有効なツールが定義されないことを意味します。

スロッティングソータイプ用ツールオフセットデータ

次のオフセットデータ（TOA データ）は，スロッティングソータイプ（タイプ 700）に指定できます。

	ジオメトリ	摩耗	基本	
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
ソーブレード補正 直径 d	$TC_DP6	$TC_DP15		mm
スロット幅 b	$TC_DP7	$TC_DP16		mm
プロジェクション k	$TC_DP8	$TC_DP17		mm

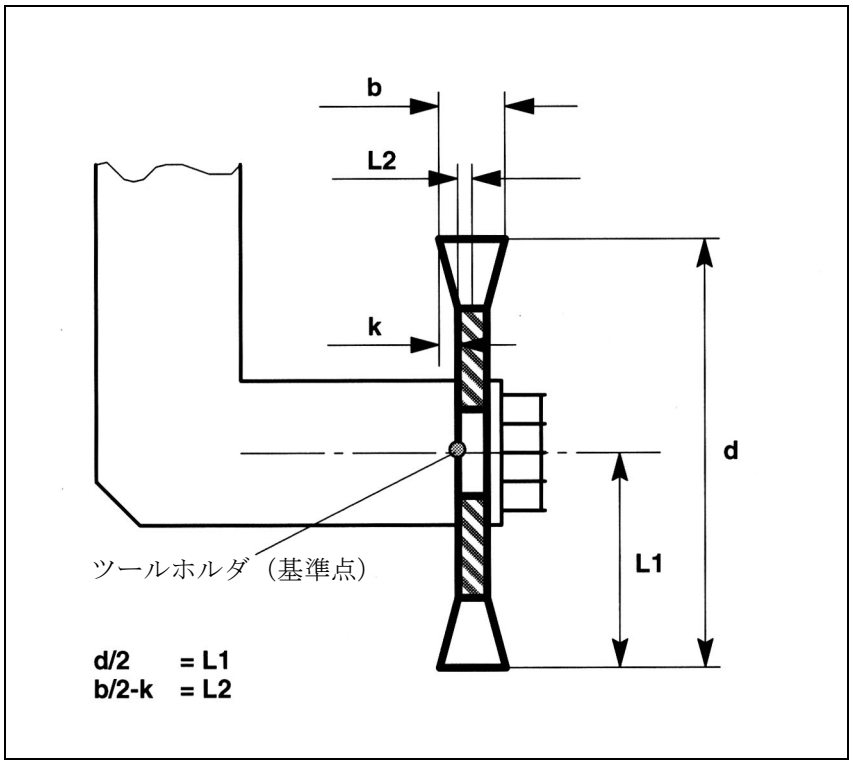


図 1.175 スロッティングソーのジオメトリ（アングルヘッドカッタに類似）

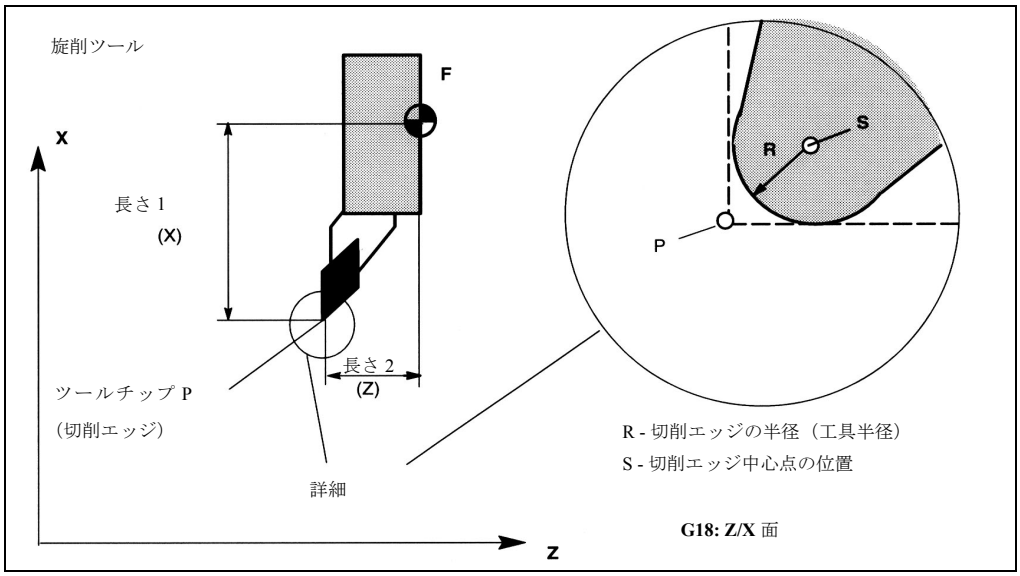
ソーブレード幅は G40 ～ G42 を使って考慮されます（G40 ソーブレード補正なし，G41 ソーブレード補正左，G42 ソーブレード補正右）。

■ ツールエッジ位置（ツールパラメータ 2）

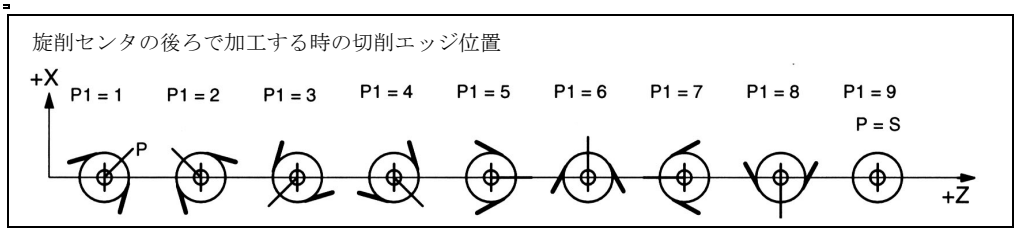
意味

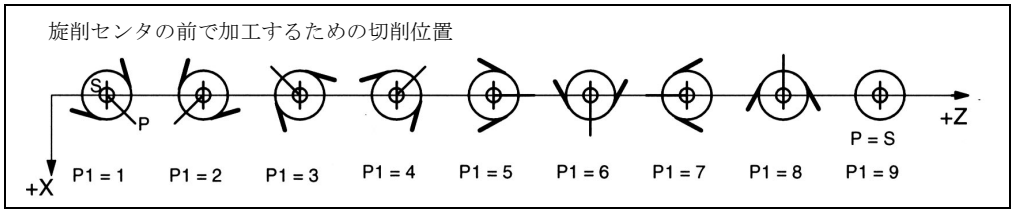
旋削ツール（ツールタイプ 5xx）の場合，制御には切削エッジ半径（ツールパラメータ 8）と切削エッジ位置が必要です。切削エッジ半径は，旋削ツールがどのようにツールホルダに固定されるかを記述します。図 1.175 に示すように，切削エッジの中心点 S に対する点 P の位置が，切削エッジ位置（ツールパラメータ 2）の識別子（値 1 ～ 9 の値）として入力されます。

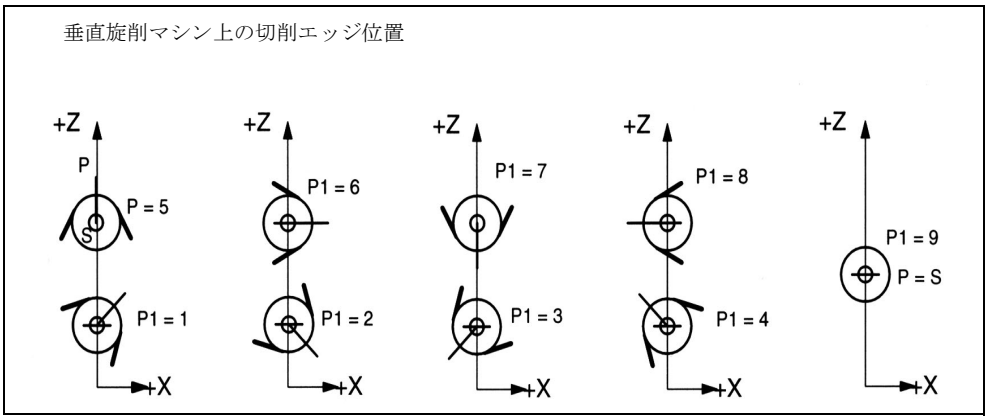
図 1.176 旋削ツールの切削エッジ位置の定義



ツールパラメータ (P2) は切削エッジの位置を指定します。P2=1 から P2=9 が可能です。







特殊機能

- 切削エッジ中心点 S が，工具長補正を計算するため点 P の替わりに基準点として使用される場合，識別子 9 が切削エッジ位置に入力されなければなりません。
- 識別子 0（ゼロ）は切削エッジ位置として認められません。

■ ジオメトリ - 工具長補正（ツールパラメータ 3 ～ 5）

ツール長 l ～ 3

工具長補正のためのツールの長さは，ツール長 1 ～ 3（ツールパラメータ 3 ～ 5）として入力されます。次の長さ仕様が，ツールタイプに従って入力される必要があります。

- ツールタイプ 12x，140，145，150　：ツール長 1
- ツールタイプ 13x　：ツール長 1-3 （面 G17-G19 に依存）
- ツールタイプ 2xx　：ツール長 1
- ツールタイプ 5xx　：ツール長 1-3

重要

ツールタイプに関係なく，全 3 つのパラメータ 3 ～ 5（ツール長 1 ～ 3）が 3 つのジオメトリ軸に計算されます。

特定のツールタイプ（ツール長 1 のみがツールタイプ 2xx に必要）に対して必要以上に多いツール長が，ツールパラメータ 3 ～ 5 に入力される場合，これらの追加ツール長はアラームの出力なしにジオメトリ軸へ計算されます。

（注）ツールパラメータ 3 ～ 5（ツール長 1 ～ 3）のツールサイズ（長さ）入力，およびそれらが 3 つのジオメトリ軸（平面 G17 ～ G19）にどのように計算されるかについての情報は，次の文書を参照してください。

参照： /BA/, Operator’s Guide

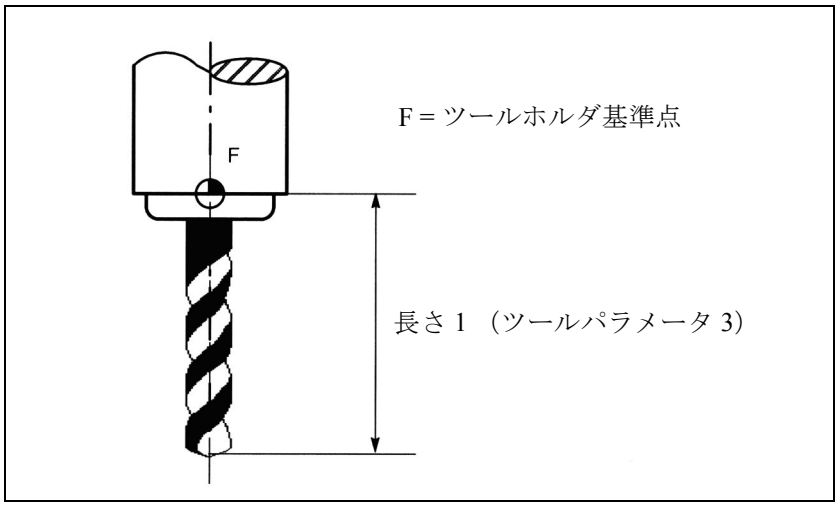


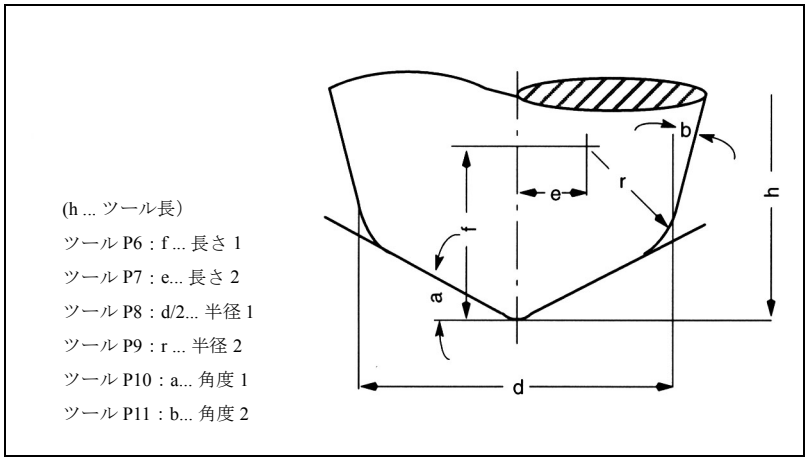
図 1.177 ツール長（ツールパラメータ 3）を持つドリル（ツールタイプ 200）

特殊機能

ツールのアクティブサイズは，ジオメトリ工具長補正（ツールパラメータ 3 ～ 5）と摩耗工具長補正（ツールパラメータ 12 ～ 14）が合計されて初めて定義されます（1.19.2「■ MG のプログラミング」参照）。ベース寸法／アダプタ寸法も，完全な工具長補正をジオメトリ軸に計算するため加算されます（1.19.2「■ツール管理または「フラット D 番号」」参照）。

■ ジオメトリ - 工具径補正（ツールパラメータ 6 ～ 11）

ジオメトリ工具径補正はツールの形状を定義します。



（注）ツールジオメトリの記述
図 に示すツール記述は正面フライスのみに必要です。
それ以外の場合：
6 ～ 11 のパラメータの内，ツールパラメータ 6（工具半径 1）のみが使用されます。
ツールパラメータ 6 ～ 11 へのツール形状の入力（工具径補正用半径）およびそれらが 3 つのジオメトリ軸に計算される方法についての詳細は，次の文書を参照してください。

参照： YS840DI ユーザーズマニュアル・プログラミング編
3.12 3 次元工具径補正（W5）

ツール長 1

使用されません（図 1.177 参照）。

ツール長 2

使用されません（図 1.177 参照）。

工具半径 1

工具半径は，次のツールタイプに対してツールパラメータ 6（工具半径 6）に入力されなければなりません。

- ツールタイプ 1xx　　フライス加工ツール
- ツールタイプ 5xx　　旋削ツール

工具半径は，ドリルツール（ツールタイプ 2xx）に対して入力する必要はありません。切削エッジ位置は，旋削ツール（ツールタイプ 5xx，1.19.2「■ PLC から D 番号を読取る」参照）に対して追加入力される必要があります。

工具径補正は，工具半径を現在の面（G17 ～ G19，1.19.2「■ツール切削エッジ」参照）に計算するために使用されます。

工具半径 2
ツール角度 1
ツール角度 2

使用されません（注記参照）。

■ 摩耗 - 工具長補正（ツールパラメータ 12 ～ 14)

意味

ジオメトリ工具長補正（ツールパラメータ 3 ～ 5）は，ツールサイズを定義するために使用される一方，摩耗工具長補正は，アクティブツールサイズの変化を修正するために使用できます。アクティブツールサイズは次の理由で変化します。

- ツール測定器のツール取付具とマシンツールのツール取付具の間の相違。
- 加工により寿命中に生じたツールの摩耗。
- 仕上しろの定義。

アクティブツールサイズ

ジオメトリ工具補正（ツールパラメータ 3 ～ 5）と摩耗工具長補正（ツールパラメータ 12 ～ 14）は，アクティブツールのサイズに達するまで合計されます（ジオメトリツール長 1 が摩耗ツール長 1 に加算される，以下同様)。

■ 摩耗 - 工具径補正（ツールパラメータ 15 ～ 20）

意味

ジオメトリ工具径補正（ツールパラメータ 6 ～ 11）は，ツールの形状を定義するのに使用される一方，摩耗工具径補正は，アクティブツール形状の変化を修正するのに使用できます。

アクティブツールの形状は次の理由で変化します。

- 加工による寿命中に生じるツールの摩耗
- 仕上しろの定義

アクティブツールの形状

ジオメトリ工具径補正（ツールパラメータ 6 ～ 11）と摩耗工具径補正（ツールパラメータ 15 ～ 20）が，アクティブツールの形状に達するまで合計されます（ジオメトリ工具半径 1 が摩耗工具半径 1 に加算される，以下同様)。

■ ベース寸法／アダプタ寸法 - 工具長補正（ツールパラメータ 21 ～ 23）

意味

ベース寸法／アダプタ寸法は，ツールホルダ（ツールサイズ）の基準点がツールホルダの基準点と異なる時使用されます。次の場合生じます。

- ツールとツールアダプタは別々に測定されますが，マシンに一体で取付けられます（ツールサイズとアダプタサイズは，切削エッジに別々に入力されます)。
- ツールは，別の位置にある第二ツール取付具（例えば立てスピンドルと横スピンドル）の中で使用されます。
- ツールタレットのツール取付具は様々な位置にあります。

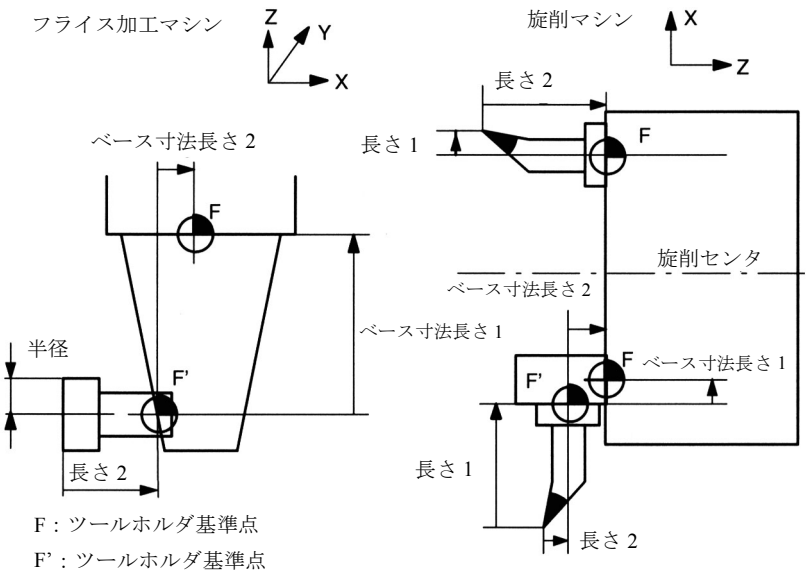


図 1.178 ベース寸法／アダプタ寸法 TLC への適用例

ベース長さ 1 ～ 3

ツールホルダ基準点 F とツールホルダ基準点 F ' の不一致が 3 つのジオメトリ軸（三次元）上で修正されるには，全てのベース長さがツールタイプに関係なくアクティブです。言い替えれば，工具長補正（長さ 1）を持つドリル（ツールタイプ 200）も，3 軸にベース寸法／アダプタ寸法を持つことができます。

（注）ベース／アダプタ寸法工具長補正についての詳細は，次の文書を参照してください。

参照： YS840DI ユーザーズマニュアル・プログラミング編

■ テクノロジ - ツールチェンジ角度（ツールパラメータ 24）

意味

レリーフ切削で移動動作が生成されるいくつかの旋削サイクルは，輪郭違反がないかどうかアクティブツールのツールチェンジ角度を監視します。

値範囲

角度（先頭符号なし 0 ～ 90 度）が，ツールチェンジ角度としてツールパラメータ 24 に入力されます。

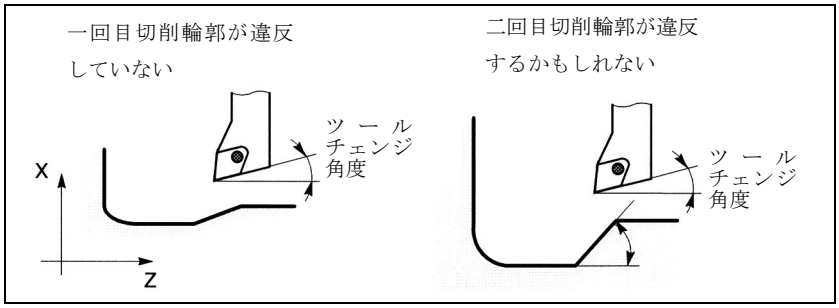


図 1.179 旋削ツールのツールチェンジ角度

縦／正面

ツールチェンジ角度は，加工のタイプ（縦または正面）に従って異なって入力されます。ツールが縦と正面の両方に使用される場合，2 つの切削エッジは異なるツールチェンジ角度に対して入力されなければなりません。

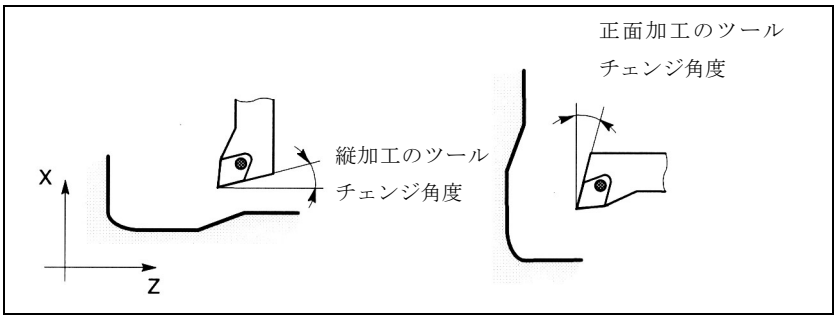


図 1.180 縦および正面加工のツールチェンジ角度

（注）ツールチェンジ角度（パラメータ 24）にゼロが入力される場合，旋削サイクルでレリーフ切削を監視しません。ツールチェンジ角度についての詳細は，次の文書を参照してください。

参照： YS840DI ユーザーズマニュアル・プログラミング編

■ 関連するツール点方向を持つツール（SW 5 以降）

SW 4.x およびそれ以前

G40 から G41/G42 への変更またはその反対は，関連するツール点方向を持つツールのツールチェンジとして処理されます。これは，変換（例えば TRANSMIT）がアクティブであるとき前処理停止が生じるため，意図されたパート輪郭からずれを起こします。

SW 5 以降

次の変更がなされました。

1.G40 から G41/G42 の変更またはその反対は，もはやツールチェンジとして処理されません。したがって前処理停止はトランスミットで生じません。

2. ブロックのスタートとエンドの切削エッジ中心点間の直線は，アプローチと後退ブロックのインタセクションを計算するために使用されます。ツール基準点と切削エッジ中心点の間の相違がこの動作に重ね合わせられます。KONT でのアプローチおよび／または後退の場合，動作は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。したがってジオメトリカルな関係は，関連するツール点方向付きおよびツール点方向なしのツールに関して同じです。アプローチおよび／または後退ブロックが隣接ブロックと交差しない場合，前の動きとの相違は比較的稀な状況でしか起こりません（下図参照）。

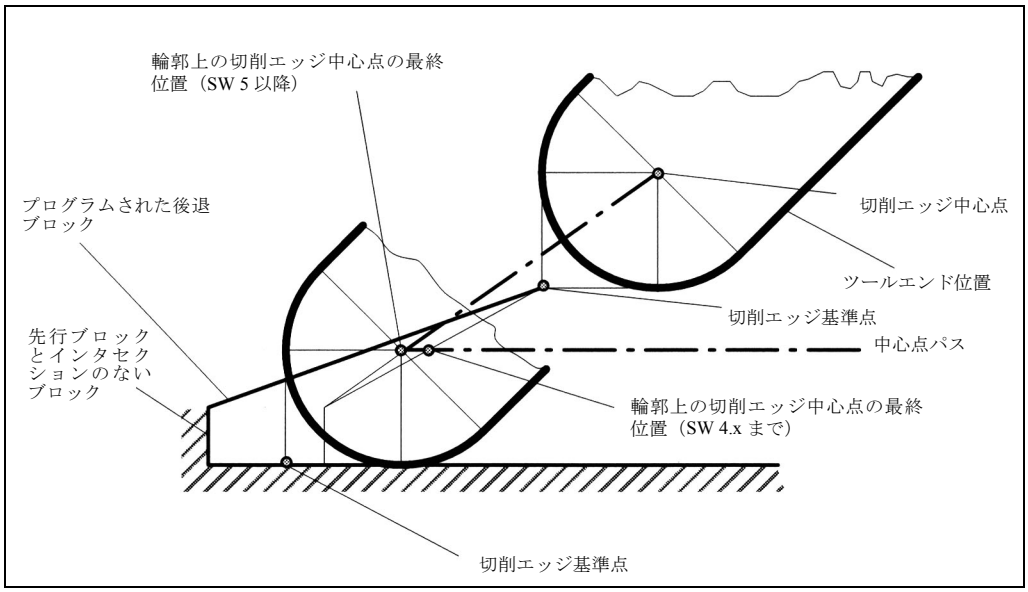


図 1.181 関連するツール点方向を持つツールの後退動作

3. 分母次数 > 4 のある有理多項式を持つ円ブロックおよび移動ブロック内で，切削エッジ中心点と切削エッジ基準点の間の距離が変わるアクティブ工具径補正を使ったツールチェンジの実行は認められません。他のタイプの補間では，以前のバージョンと異なり，変換（例えばトランスミット）がアクティブである時のみチェンジが認められます。
4. 可変ツール方位を使った工具径補正の場合，切削エッジ基準点から切削エッジ中心点への変換は，単純ゼロオフセットで実行できません。したがって，関連するツール点方向を持つツールは，3D 円周フライス加工が禁止されます（アラーム）。

（注）以前と同様に，関連するツール点方向のない定義済みツールタイプは認められるので，問題は正面フライスには関係ありません。（明示的に認められていないタイプを持つツールは，指定半径を持つボールエンドミルとして処理されます。ツール点方向の仕様は無視されます）。

■ 工具径補正：2D (TRC)

（注）工具径補正 (TRC) についての情報は次の文書を参照してください。

参照： YS840DI ユーザーズマニュアル・プログラミング編

プログラミング編のみが，工具径補正 (TRC) の完全な技術的説明を含んでいます。

TRC の必要性

パートプログラムにプログラムされたワークの輪郭（ジオメトリ）は，生産に使用する工具から独立しているべきです。このため，現在のオフセットメモリから工具の長さと半径の値を読出す必要があります。工具径補正は，工具半径からプログラム済み輪郭への距離（一定）を計算するのに使用できます。

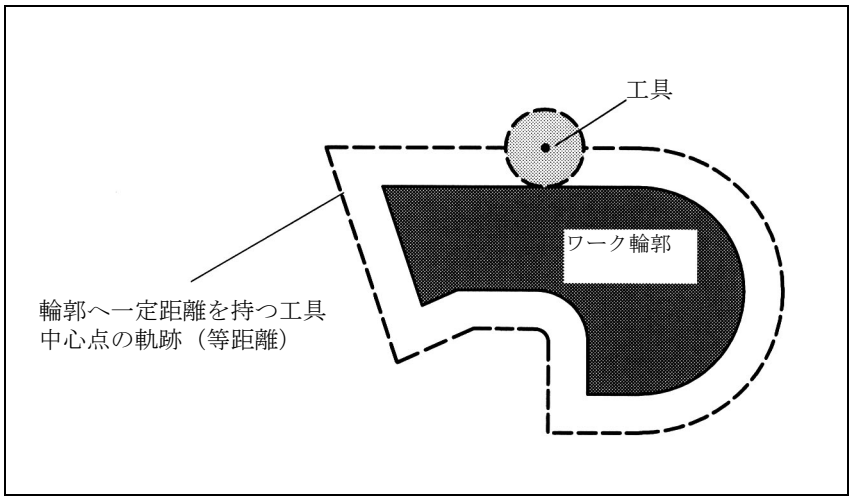


図 1.182 等距離を持つワーク輪郭（ジオメトリ）

面上の TRC

工具径補正は，次のタイプの補間について現在の平面（G17 ～ G19）においてアクティブです。

- 直線補間 ... G0, G1
- 円補間 ... 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

■ TRC (G41/G42) の選択

工具径補正 (TRC) は，プログラムされた輪郭へ一定距離の軌跡を計算します。補正は，動作向きへ向かって輪郭の左側または右側で実行できます。

- G41 ... 動作向きへ向かって輪郭の右側での TRC
- G42 ... 動作向きへ向かって輪郭の左側での TRC
- G40 ... TRC の選択解除（1.19.2「■ジオメトリー工具長補正（ツールパラメータ 3～5)」を参照)。

ダミーブロック

一般的に，工具径補正 (TRC) がアクティブである時，現在面のジオメトリ軸上に位置を持つプログラムブロックのみがプログラムされます。しかし，ダミーブロックもアクティブ TRC でやはりプログラムできます。ダミーブロックは，現在面のジオメトリ軸上に位置を含まないプログラムブロックです。

- インフィード軸上の位置
- 補助機能
- その他

ダミーブロックの最大数は，MD 20250:CUTCOM_MAXNUM_DUMMY_BLOCKS（TRC を使って移動のないブロック数）に定義できます。

特殊機能

- 工具径補正は，G0（早送り）または G1（直線補間）を使ったプログラムブロックでのみ選択できます。
- 工具は，工具径補正選択を使ってプログラムブロックの最後にロード（T 機能）され，そしてツール切削エッジ（ツールオフセット）（D1 ～ D9）が起動されなければなりません。
- 工具径補正は，D0 のツール切削エッジ／ツールオフセットを使って選択されません。
- 工具径補正が選択される時に，1 つのみのジオメトリ軸が面にプログラムされる場合，第二軸は自動的に面に追加されます（最終プログラム位置)。
- ジオメトリ軸が，工具径補正選択を使ってブロックの現在の面にプログラムされない場合，選択は生じません。
- 工具径補正が，工具径補正選択に続いてブロック内で選択解除 (G40) される場合，選択は生じません。
- 工具径補正が選択されると，アプローチ応答が NORM/KONT によって決定されます（1.19.2「■ツールエッジ位置（ツールパラメータ 2)」参照)。

■ アプローチおよび後退応答 (NORM/KONT)

指令 NORM と KONT を使って，アプローチ応答（G41/42 を使って工具径補正の選択）および後退応答（G40 を使って工具径補正の選択）を制御できます。

NORM　... 始点／終点でノーマルの設定（初期処理）

KONT　... 始点／終点で輪郭を辿る

特別機能

- ツールスタート位置が輪郭の後ろにある場合，KONT のみが NORM と異なる。

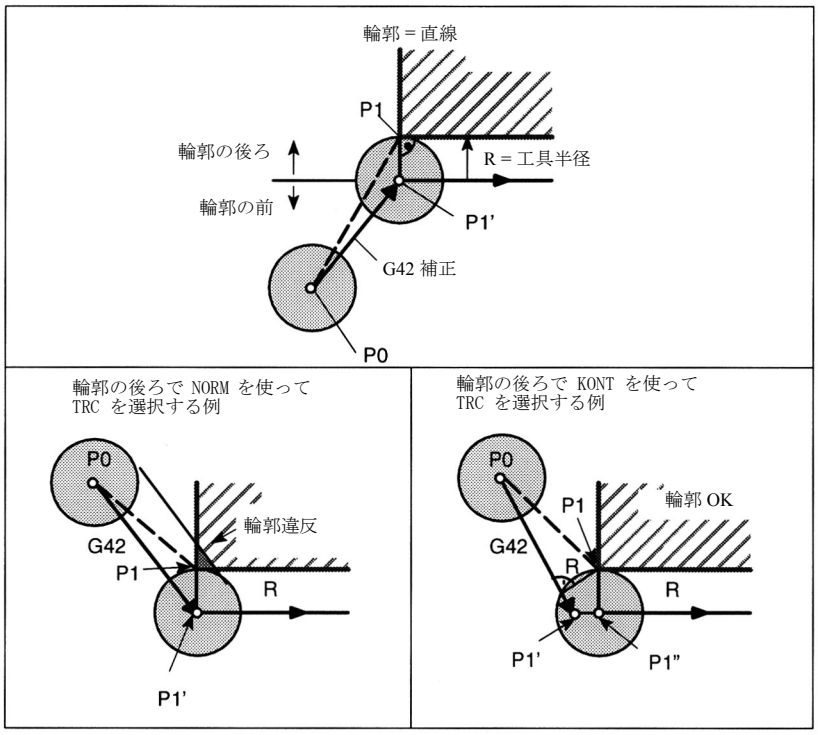


図 1.183 輪郭の前後で KONT または NORM を使って TRC を選択する例

- KONT と G450/G451（外部コーナでコーナ応答）は一般的効果を持ち，TRC を使ってアプローチと後退を決定します。
- 工具径補正が選択解除される時，後退アプローチは KONT/NORM 指令によって決定されます（1.19.2「■ツールエッジ位置（ツールパラメータ 2)」参照)。

■ ソフトアプローチおよび後退（SW 4.3 以降）

意味

ソフトアプローチと後退機能 (SAR) は，始点の位置に関係なく，輪郭の始点へ正接アプローチを達成するのに使用されます。

アプローチ応答は多様で，追加パラメータ範囲を使って特殊ニーズに合わせられます。

ソフトアプローチとソフト後退の 2 つの機能は，ほぼ左右対称です。したがって，次のセクションはアプローチの詳細記述に絞られます：後退で生じる相違が特別参照されます。

アプローチ動作

最大 4 つのコンポーネント動作：

- 動作の始点 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 と P_2 は，定義されたパラメータとジオメトリ条件に従って省略できます。

後退動作

後退の場合，点は逆向きに横断します。つまり P_4 でスタートし P_0 で終わります。

パラメータ

ソフトアプローチおよび後退機能の応答は，9 までのパラメータによって決定します。

- アプローチおよび後退輪郭を定義するための非モーダル G コード。
  この G コードは省略されない場合があります。
  - G147：直線アプローチ
  - G148：直線アプローチ
  - G247：クォッドラントアプローチ
  - G248：クォッドラントアプローチ
  - G347：半円形アプローチ
  - G348：半円形アプローチ

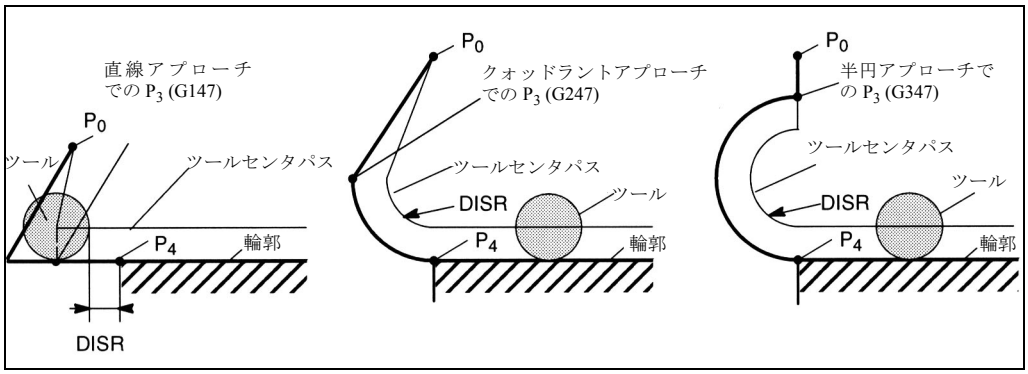


図 1.184 G147 ～ G347 および DISR に依存するアプローチ応答（工具径補正の同時起動時）

- アプローチおよび後退輪郭を定義するモーダル G コード
  この G コードは，アプローチ輪郭がクォッドラントまたは半円の場合のみ関係します。アプローチおよび後退向きは次のように確定されます。
  - G140：アクティブ工具径補正を使って，アプローチおよび後退向きを定義する（G140 は初期処理）。
    正の工具半径： G41 アクティブ→左から右へアプローチ
    G42 アクティブ→右からアプローチ
    どの工具径補正もアクティブでない場合 (G40)，応答は G143 と同じです。この場合アラームは出力されません。アクティブツールの半径が 0 の場合，アプローチと後退サイドは工具半径が正であるかのように確定されます。
  - G141：左から輪郭へアプローチ，または左へ後退。
  - G142：右から輪郭へアプローチ，または右へ後退。
  - G143：アプローチ向きの自動確定，つまり，輪郭は次のブロック (P_4) の始点の接線から見て，始点が位置する側からアプローチされます。

（注）先行ブロックの終点の接線に沿って後退が行われます。終点が明示的にプログラムされない場合，つまり暗示的に決定される場合，G143 は後退には使用できません。これはアプローチ側と終点位置の間に相互依存性があるからです。この場合，G143 がプログラムされるとアラームが出力されます。G140 がアクティブの場合同じことが適用され，非アクティブ工具径補正の結果，G143 への自動切替えが起こります。

• 始点から終点まで各ブロックに動作の下位区分を定義するモーダル G コード (G340, G341)。

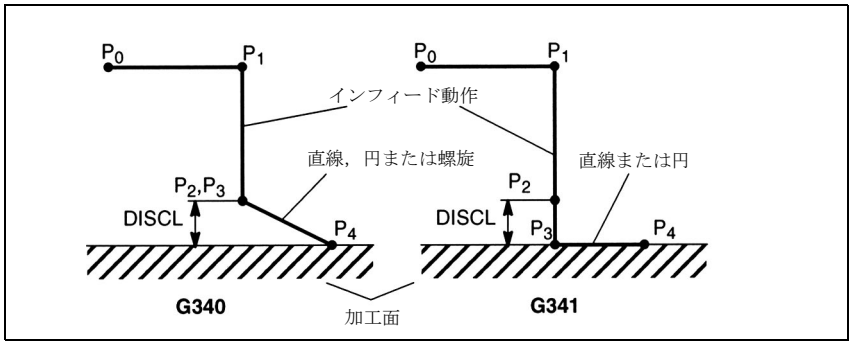


図 1.185 G340/G341 に依存するアプローチ動作のシーケンス

- G340：図 1.185 に示される P_0 ～ P_4 へのアプローチ特性。

  G247 または G347 がアクティブ（クォッドラントまたは半円）で，始点 P_3 が終点 P_4 によって定義された加工面の外側にある場合，円の代わりに螺旋が挿入されます。点 P_2 は定義されないかまたは P_3 と一致しません。
  円形面または螺旋軸は，SAR ブロック (G17 ～ G19) でアクティブである面によって決定されます。つまり，スタート接線の突出しは，接線自体の代わりに，円を定義するために次のブロックによって使用されます。
  点 P_0 から 点 P_3 への動作は，SAR ブロックの前に有効であった速度で 2 つの直線に沿って起こります。

- G341：P_0 から P_4 へのアプローチ特性は図 1.185 に示されます。

  P_3 と P_4 は加工面内に位置しています。その結果，円が G247 または G347 を使って螺旋の代わりに常に挿入されます。

  （注）アクティブな回転フレームは，アクティブ面 G17 ～ G19（アクティブ面に垂直な円形面，螺旋軸，インフィード動作）の位置が関連する全てのケースに含まれます。

• DISR：アプローチ直線の長さまたはアプローチ弧の半径を指定します（図 1.184 を参照）。

直線に沿ったアプローチ／後退の場合，DISR はカッタエッジから輪郭の始点までの距離を指定します。つまり，アクティブ TRC を使った直線の長さは，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。

次の場合，直線を使ったアプローチと後退時にアラームが出力されます。

- DISR が負でその量が工具半径より大きい（生じるアプローチ線の長さが，ゼロに等しいかまたはゼロ以下）。

円を使って，DISR は常にツールセンタパスの半径を指定します。工具径補正が起動される場合，円は内部的に生成され，その半径はこの場合も同様に，ツールセンタパスがプログラムされた半径から導かれる大きさになります。

次の場合，円を使ったアプローチと後退時アラームが出力されます。

- 内部的に生成された円の半径がゼロまたは負。

- DISR がプログラムされない。

- それが値 ≦ 0 を持つ。

• DISCL：点 P_2 から加工面までの距離を指定します（図 1.185 参照）。

点 P_2 の位置が，絶対基準によって円面に垂直な軸に指定される場合，値は形式 DISCL = AC(....) でプログラムされなければなりません。

DISCL がプログラムされない場合，点 P_1，P_2 および P_3 は G340 に一致し，アプローチ輪郭は P_1 ～ P_4 まで生成されます。

システムは，DISCL によって定義された点が P_1 と P_3 の間にあるかどうか監視します。つまり，加工面に垂直なコンポーネントを持つ全ての動作（例えば P_3 から P_4 へのインフィード動作，アプローチ動作）に関してして，このコンポーネントは同じ先頭符号を持たなければなりません。方向を変えることは認められません。この条件に違反するとアラームが出力されます。

方向逆転の検出では，MD 20204: WAB_CLEARANCE_TOLERANCE で定義された公差が認められます。しかし P_2 が P_1 と P_3 で定義された範囲外にあり，ずれがこの公差に等しいかまたは以下である場合，P_2 は，P_1 および／または P_3 によって定義された面にあると仮定されます。

例 :

アプローチは，点 P_1 の位置 Z=20 で始まる G17 を使ってなされます。P_3 によって定義された SAR 面は Z=0 にあります。したがって，DISCL で定義された点はこれらの 2 点間になければなりません。MD 20204=0.010。P_2 が 20.000 と 20.010 の間または 0 と -0.010 の間にある場合，値 20.0 または 0.0 はプログラムされていると仮定されます。P_2 の Z 位置が 20.010 より大きいかまたは -0.010 以下の場合，アラームが出力されます。

加工面を基準にした始点 P_0 と終点 P_4 の相対位置に依存して，インフィード動作は負（アプローチの場合は垂直）または正（後退の場合は垂直）の方向に実行されます。つまり，G17 の場合は，終点 P_4 の Z コンポーネントは，始点 P_0 の Z コンポーネントより大きくなることができます。

• X...Y... Z.... を使って終点 P_4 （または後退で P_0）を一般的にプログラミングする

- アプローチ用に終点 P_4 をプログラミングする可能な方法

終点 P_4 は実 SAR ブロック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確定することができます。ジオメトリ軸を移動させることなく，さらなるブロック（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。

少なくとも 1 つのジオメトリ軸（G17 を使って X または Y）が，加工面にプログラムされる場合，終点は，アプローチのため実 SAR ブロックに既にプログラムされていると見なされます。加工面に垂直な軸（G17 を使って Z）の位置のみが SAR ブロックにプログラムされる場合，このコンポーネントは SAR

ブロックから得られますが，面の位置は次のブロックから得られます。この場合，加工面に垂直な軸も次のブロックにプログラムされるとアラームが出力されます。

例：

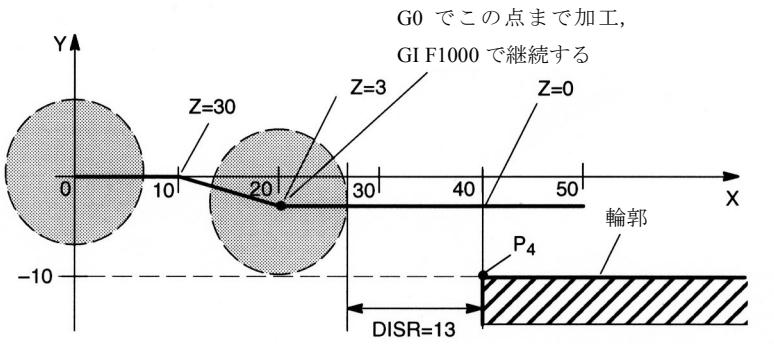


図 1.186 例

$TC_D[1,1]=120　　：フライス加工ツール T1/D1

$TC_DP6[1,1]=7　　：7mm の半径を持つツール

```
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 は次のものに置換えることができます：

1.

```
N30 G41 G147 DISCL=3 DISR=13 X40 Y-10 Z0 F1000
```

または

2.

```
N30 G41 G147 DISCL=3 DISR=13 F1000
N40 G1 X40 Y-10 Z0
```

- 後退用に終点 P_0 をプログラミングする方法

エンド位置は，どれほど多くの軸がプログラムされていても常に SAR ブロックから得られます。次の状況を区別して説明します。

1. ジオメトリ軸が SAR ブロックにプログラムされない。

   この場合，輪郭は点 P_2（または P_1 と P_2 が一致する場合は点 P_1）で終わります。加工面を記述する軸の位置は，後退輪郭（直線または弧の終点）によって決定されます。これに垂直な軸のコンポーネントは，DISCL によって定義されます。この場合 DISCL = 0 であると，動作は完全に面内で起こります。

2. 加工面に垂直な軸のみが SAR ブロックにプログラムされる。
   この場合，輪郭は点 P_1 で終わります。他の二つの軸の位置は 1 と同じ方法で決定されます。

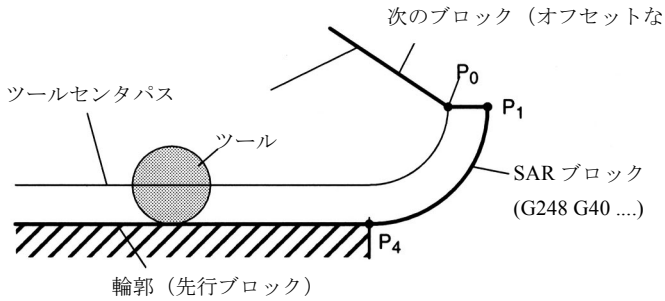


図 1.187 TRC の同時起動解除を使った SAR での後退

SAR 後退ブロックが，1 と 2 のケースで工具径補正を起動解除するためにも使用される場合，P_1 から P_0 へ追加パスが挿入され，工具径補正が後退輪郭のエンドで起動解除される時に動作が起こらないようにします。つまりこの点は，修正される輪郭上の位置でなく，ツール中心点を定義します。

3. 加工面の最低 1 つの軸が，プログラムされる。加工面の第二軸は，先行ブロックのその最終位置からモーダルに決定できます。加工面に垂直な軸位置は，この軸がプログラムされるか否かに依存して，1 または 2 に説明したように生成されます。この方法で生成される位置は，終点 P_0 を定義します。

   工具径補正の選択解除に特別な方法は必要ありません。なぜならプログラムされた点 P_0 は，既に速やかに完全な輪郭のエンドにツール中心点の位置を定義しているからです。

   SAR 輪郭のスタートおよび終点（P_0 と P_4）は，アプローチおよび後退で一致が可能です。

• 先行ブロック（通常 G0）の速度

P_0 から P_2 への全ての動作はこの速度で実行されます。つまり，加工面に平行な動作および安全クリアランスまでのインフィード動作の部分。

• FAD を使ってフィードをプログラミングする

- G340 でプログラムされる FAD：P_2 または P_3 から P_4 への送り速度

- G341 でプログラムされる FAD：P_2 から P_3 への加工面に垂直なインフィード動作の送り速度

FAD がプログラムされない場合，輪郭のこの部分は，先行ブロックからモーダルにアクティブである速度で移動します。ただし，速度を定義する F コマンドが SAR ブロックにプログラムされない場合です。

- プログラムされる応答：

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 のアクティブ G コードに従って動作します（フィードタイプ：G93, G94 等）

FAD=PM(...)　→プログラムされた値は，グループ 15 のアクティブ G コードに関係なく直線フィード（G94 のように）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 のアクティブ G コードに関係なく直線フィード（G94 のように）として解釈されます。

例：

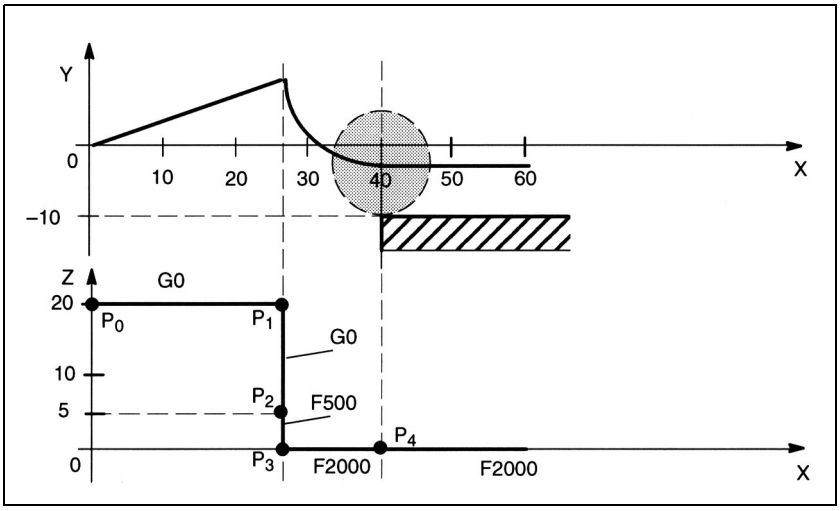


図 1.188 例

$TC_DP1[1,1]=120　：フライス加工工具 T1/D1

$TC_DP6[1,1]=7　：7mm の半径を持つ工具

```
N10 G90 G0 X0 Y0 Z20 D1 T1
N20 G41 G341 G247 DISCL=AC(5) DISR=13 FAD 500 X40 Y-10 Z=0 F2000
N30 X50
N40 X60
...
```

- プログラムされるフィード F

  このフィード値は，点 P_3（FAD がプログラムされない場合は点 P_2）から有効です。F コマンドが SAR ブロックにプログラムされない場合，先行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次のブロックには使用されません。

速度

図 1.189 および図 1.190 に示すアプローチディスプレイでは，新しい速度が SAR ブロックに続くブロックにプログラムされないと仮定されます。そうでない場合，新しい速度が点 P_4 の後有効になります。

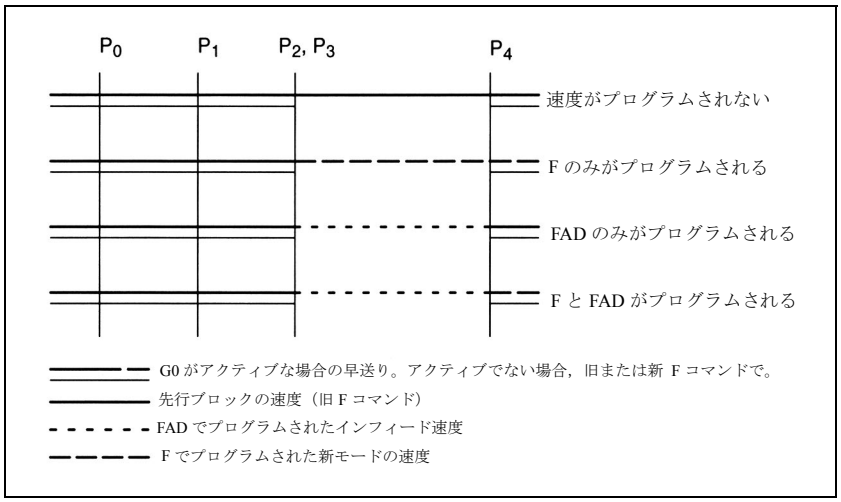


図 1.189 G340 を使ったアプローチ時の SAR サブブロックの速度

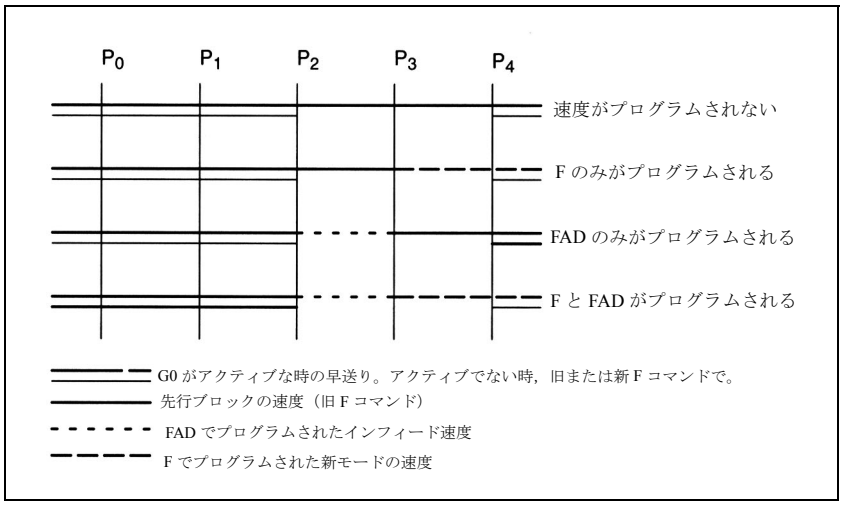


図 1.190 G341 を使ったアプローチ時の SAR サブブロックの速度

後退時，先行ブロックからのモーダルフィード機能と SAR ブロックにプログラムされたフィード速度はスワップされます。つまり，実際の後退輪郭（直線，円，螺旋）は，旧送り速度で移動する一方，F コマンドでプログラムされた新速度は，点 P_2 から点 P_0 まで有効です。レベル後退がアクティブで，FAD がプログラムされる場合，P_3 から P_2 へのパスは FAD で移動され，そうでない場合，旧速度が適用されます。先行ブロックにプログラムされた最終 F コマンドは，常に P_4 から P_2 の移動に適用されます。G0 はこれらのブロックに影響しません。

P_2 から P_0 への移動は，SAR ブロックにプログラムされた F コマンドで起こり，F コマンドがプログラムされない場合，先行ブロックからのモーダル F コマンドで起こります。これは，G0 がアクティブでなければ適用されます。

早送りが P_2 から P_0 のブロックの後退に使用される場合，G0 が，SAR ブロックの前または SAR ブロック自体の中で起動されなければなりません。追加 F コマンドが実 SAR ブロックにプログラムされる場合，これは有効ではありません。しかしこれは次のブロックに対してモードダルでアクティブなまま残ります。

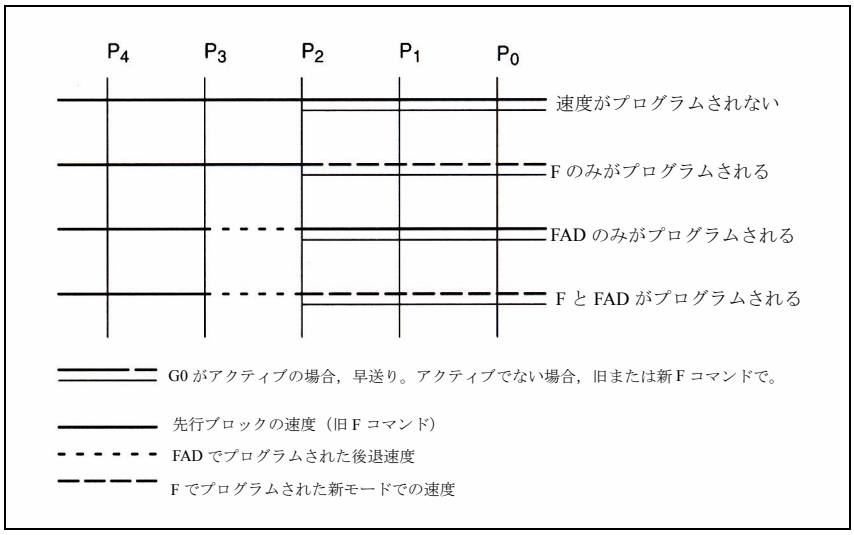


図 1.191 後退時 SAR サブブロックの速度

システム変数

点 P_3 と P_4 は，アプローチ中システム変数として WCS（ワーク座標系）で読取られます。

- $P_APR：WCS の P_3（始点）を読取る
- $P_AEP：WCS の P_4（輪郭始点）を読取る
- $P_APDV=1 $P_APR と $P_AEP の内容が有効である場合。つまりそれらが，プログラムされた最終 SAR アプローチブロックに属する位置の値を含む場合。$P_APDV=0 の場合，旧 SAR アプローチブロックの位置が読取られる。

SAR ブロックと読取りオペレーションの間で WCS を変更しても位置の値に影響しません。

補足条件

- さらなる NC コマンド（補助機能出力，同期軸動作，位置決め軸動作など）が，SAR ブロックにプログラムできます。これらはアプローチの最初のサブブロックで，後退の最終サブブロックで実行されます。
- 終点 P_4 が，SAR ブロックからでなく後続の移動ブロックから取得される場合，実際の SAR 輪郭（直線，クォッドラントまたは半円）はこのブロックで移動します。元の SAR ブロックの最終サブブロックは，ジオメトリ軸の移動情報を含みません。しかしさらなる動作（単一軸）がこのブロックで実行しなければならないので，これは常に出力されます。
- 少なくとも次の 2 つのブロックを常に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を定義するブロック

  さらに多くのブロックがこれらのブロック間にプログラムできます。
  MD 20202: WAB_MAXNUM_DUMMY_BLOCKS は中間ブロックの最大数を制限します。
- 工具径補正がアプローチブロックで同時に起動される場合，SAR 輪郭の最初の直線ブロックは起動が起こるブロックです。SAR 機能によって生成される完全な輪郭は，あたかもそれが明示的にプログラムされたかのように工具径補正によって処理されます（衝突監視，インタセクションの計算，アプローチ応答 NORM/KONT）。
- インフィード動作向きおよび円面または螺旋軸の位置は，アクティブ面 (G17 ～ G19) によって排他的に定義されます - 適切な場合アクティブフレームを使って回転される。
- アプローチ時，前処理停止は，SAR ブロックと接線向きを定義する次のブロックの間に挿入されてはなりません。この場合，明示的にプログラムされても，またはコントローラによって自動的に挿入されても，前処理停止は結果的にアラームを発します。

アラーム

プログラミングが不正確な場合，次のアラームが出力されます。

アラーム 10740	「SAR プログラムに空ブロックが多すぎる」
アラーム 10741	「SAR インフィード動作時，向き逆転」
アラーム 10742	「無効 SAR 距離」
アラーム 10743	「二度以上 SAR がプログラムされた」
アラーム 10744	「SAR 方向が定義されていない」
アラーム 10745	「SAR エンド位置が固有でない」
アラーム 10746	「SAR に対する前処理停止」
アラーム 10747	「後退方向が SAR に定義されていない」

REPOS での応答

SAR サイクルが中断され再位置決めされる場合，SAR サイクルは RMI の中断点で再開します。RME の接点は最終 SAR ブロックの終点であり，RMB でこれは最初の SAR ブロックの始点です。

RMI が DISPR（割込み点の前方距離 DISPR で再アプローチ）と共にプログラムされる場合，再アプローチ点は，割込みサブブロック前の SAR サイクルのサブブロックに表示できます。

例 1

条件は次の通りです。

- ソフトアプローチがブロック N20 で起動される。
- X=40（終点）：Y=0：Z=0
- アプローチ動作がクォッドラント (G247) で実行される。
- アプローチ向きがプログラムされず，G140 が有効である。つまり，TRC がアクティブ (G42) でオフセット値が正 (10) であるので，輪郭は右からアプローチされる。
- 内部的に生成されたアプローチ円（SAR 輪郭）が半径 20 を持ち，ツールセンタパスの半径がプログラムされた値 DISR=10 に等しい。
- G341 により，アプローチ動作が面内の円で起こり，(20, -20, 0) の始点になる。
- DISCL=5 であるので，点 P_2 は位置 (20，-20，5) にあり，Z30 により，点 P_1 は (20，-20，30) で N10 にある。

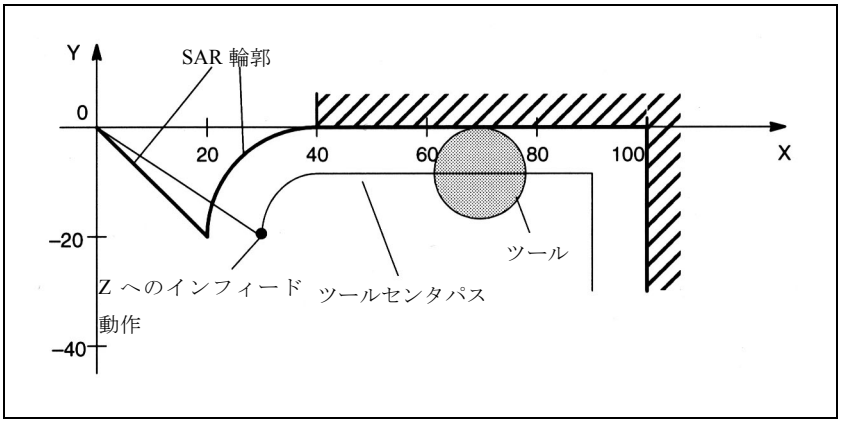


図 1.192 輪郭例 1

パートプログラム：

```
$TC_DP1[1,1]=120                    ：ツール定義 T1/D1
$TC_DP6[1,1]=10                     ：半径
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

アプローチに関する条件は次の通りです。

- ソフトアプローチがブロック N20 で起動される。
- アプローチ動作がクォッドラント (G247) を使って実行される。
- アプローチ向きがプログラムされず，G140 が有効である。つまり TRC がアクティブであるので（GA1），輪郭は左からアプローチされる。
- 輪郭オフセット OFFN=5 (N10)。
- 現在の工具半径が 10 であるので，WRK の有効な補正半径は 15：したがって SAR 輪郭の半径は 25 に等しく，その結果ツールセンタパスの半径は DISR=10 に等しい。
- Z 位置のみが N20 にプログラムされているので，円の終点は N30 から生じる。
- インフィード動作
  - 早送りで Z20 から Z7 (DISCL=AC(7))
  - その後 FAD=200 で Z0 へ
  - X-Y 面のアプローチ円および F1500 を持つ次のブロック（この速度が次のブロックでアクティブであるには，アクティブ G コード G0 は，G1 を使って N30 内で上書きしなければならない。そうでない場合，輪郭は G0 を使って継続加工される）

次の条件が後退に適用されます。

- ソフトアプローチがブロック N60 で起動される。
- 後退動作がクォッドラント (G248) および螺旋 (G340) で実行される。
- G340 の影響がないので，FAD がプログラムされない。
- DISCL=6 であるので，始点は Z=2：終点は Z=8
- DISR=5 で，SAR 輪郭の半径は 20，ツールセンタパスの半径は 5
- 円ブロックの後，後退動作は Z8 から Z20 へ至り，その動作は X70 Y0 の終点まで X-Y 面に平行である。

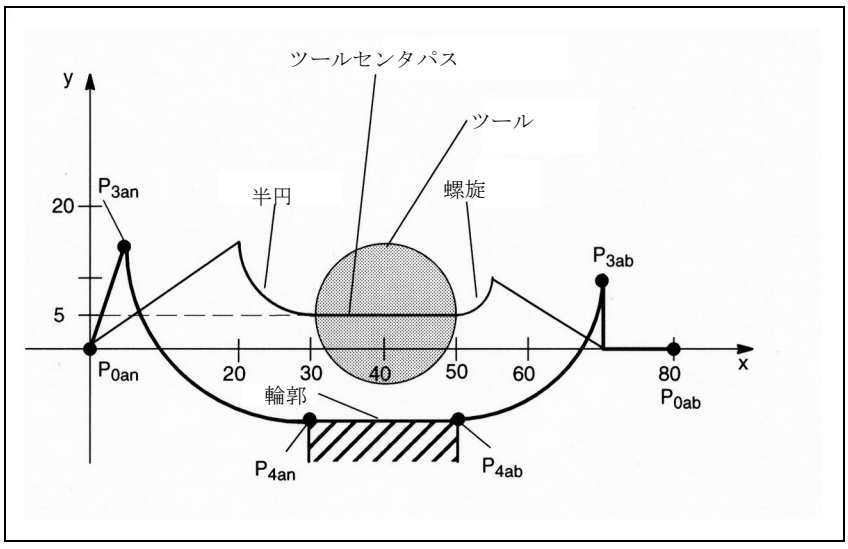


図 1.193 輪郭例 2

パートプログラム

$TC_DP1[1,1]=120　　：ツール定義 T1/D1

$TC_DP6[1,1]=10　　：半径

N10 G0 X0 Y0 Z20 G64 D1 T1 OFFN = 5　　(P_{0an})

N20 G41 G247 G341 Z0 DISCL = AC(7) DISR = 10 F1500 FAD=200　　(P_{3an})

N30 G1 X30 Y-10　　(P_{4an})

N40 X40 Z2

N50 X50　　(P_{4ab})

N60 G248 G340 X70 Y0 Z20 DISCL = 6 DISR = 5 G40 F10000　　(P_{3ab})

N70 X80 Y0　　(P_{0ab})

N80 M 30

（注）この方法で生成される輪郭は，SAR アプローチブロックで起動し SAR 後退ブロックで起動解除される工具径補正により修正されます。工具径補正は，工具半径 (10) と輪郭オフセット (5) の合計である 15 の有効半径に認められます。したがってアプローチブロック内で，ツールセンタパスの結果的に生じる半径は 10，後退ブロックで 5 です。

■ 工具径補正 (G40) の選択解除

工具径補正の選択解除は，G40 指令で実行されます。

特別機能

- 工具径補正の選択解除は，G0（早送り）および G1（直線補間）のあるプログラムブロック内でのみ実行できます。
- 工具径補正が有効な時，D0 がプログラムされると，補正は選択解除されず，エラーメッセージ 10750 が出力されます。
- ジオメトリ軸が，工具径補正選択解除を使ってブロックにプログラムされる場合，補正は，現在面になくても選択解除されます。

■ 外側コーナでの補正

機能 G450/G451 は，外側コーナで不連続ブロック遷移を使って応答を制御するのに使用できます。

- G450 ... 遷移円を使った不連続ブロック遷移
- G451 ... 等距離のインタセクションを使った不連続ブロック遷移

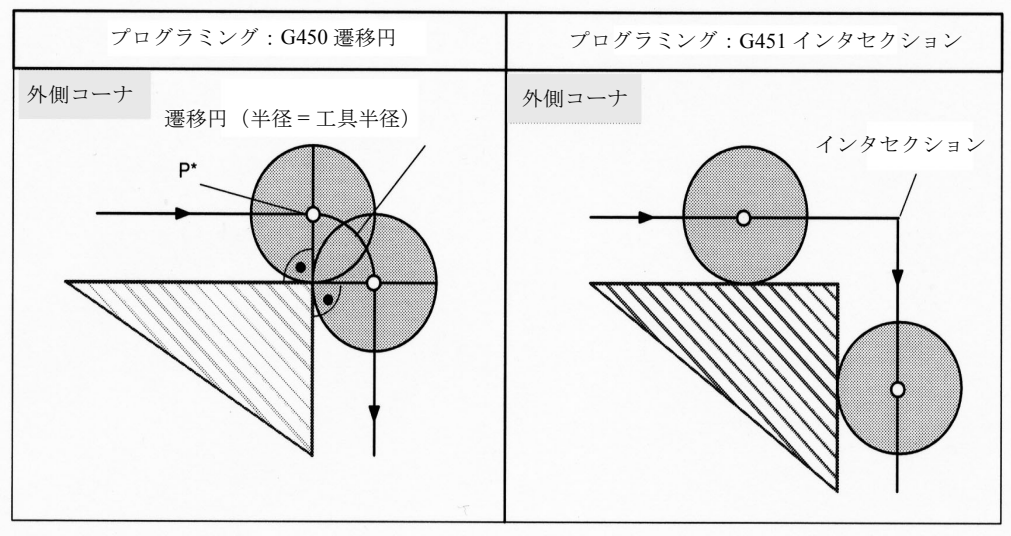


図 1.194 G450 と G451 を使った 90 度の外側コーナの例

G450 遷移円

機能 G450 が外側コーナにある時，ツールの中心点は，工具半径を使って円形軌跡を描きます。円形の軌跡は，前のパスセクション（プログラムブロック）の終点であるノーマルの位置（パス接戦に垂直）からスタートし，新しいパスセクション（プログラムブロック）の始点であるノーマルの位置で終わります。

外側コーナが非常に広い場合，G450（遷移円）と G451（インタセクション）を使った異なる応答は一層類似してきます ---> 非常に広い外側コーナを参照。

外側コーナが鋭い角度の場合，ツールは輪郭から離昇しなければなりません ---> DISC を参照。

DISC

G450 遷移円は鋭角の外側輪郭コーナを生成しません。なぜなら遷移円を通るツール中心点のパスが，切削エッジが外側コーナ（プログラムされた位置）で停止するように制御されるからです。鋭角の外側コーナが G450 で加工されるとき，DISC 指令が，オーバシュートをプログラムするために使用できます。この遷移は遷移円を円錐セクションに変換し，切削エッジは外側コーナから離昇します。DISC 指令の値範囲は，1 段階ずつ 0 ～ 100 まで拡張します。

DISC = 0 ... オーバシュートがディスエーブル，遷移円がアクティブ。

DISC = 100 ... オーバシュートが，インタセクション (G451) に類似する応答を理論的に生成するのに十分な大きさである。

MD 20220: CUTCOM_MAX_DISC（DISC の最大値）は，DISC を使ってプログラムできる最大値の定義に使用できます。DISC を使って 50 を超える値は，一般にはお奨めできません。

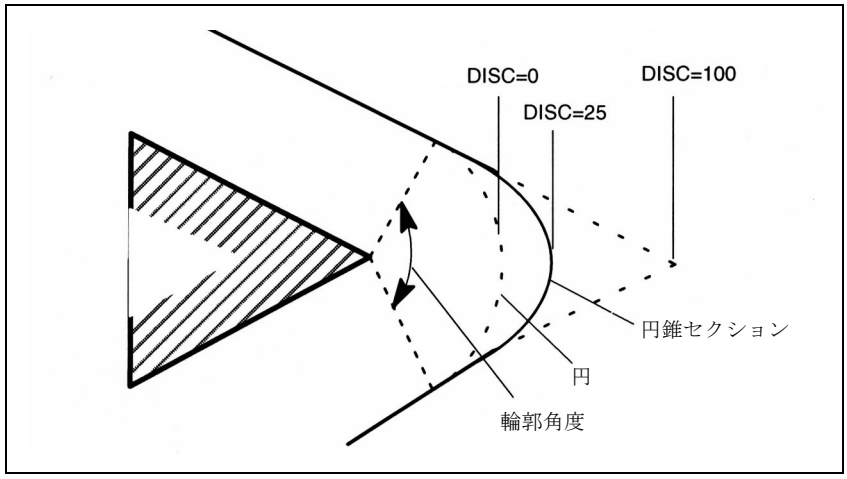
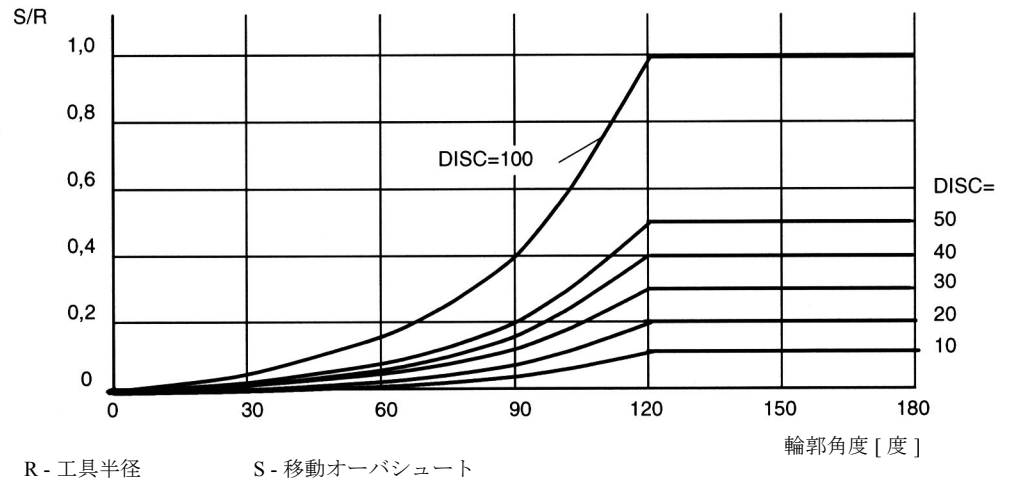


図 1.195 例：オーバシュート DISC = 25



R - 工具半径　　S - 移動オーバシュート

S/R - 測定されるオーバシュート（工具半径に関して）

図 1.196 輪郭角度に依存する DISC を使ったオーバシュート

G451 インタセクション

G 機能 G451 がアクティブである時，アプローチ位置（インタセクション）は，その距離が工具半径（ツールの中心点パス）に対応するパスライン（直線，円，または螺旋のみ）から計算されます。スプラインと多項式は，原則として拡張されません。

外側コーナが非常に鋭い場合，G451 は結果的に過剰なアイドルパスになります
→「非常に鋭い外側コーナ」を参照。

非常に鋭い外側コーナ

外側コーナが非常に鋭い場合，G451 は結果的に過剰なアイドルパスになり得ます（図 1.197 参照）。したがって外側コーナが非常に鋭い場合，システムは G451（インタセクション）から G450（適切な DISC を持つ遷移円）へ自動的に切替わります。この自動切替え（インタセクション→遷移円）に続いて移動することができる輪郭角度は，MD 20210: CUTCOM_CORNER_LIMIT に定義できます（工具径補正を使った補正ブロック用の最大角度）。

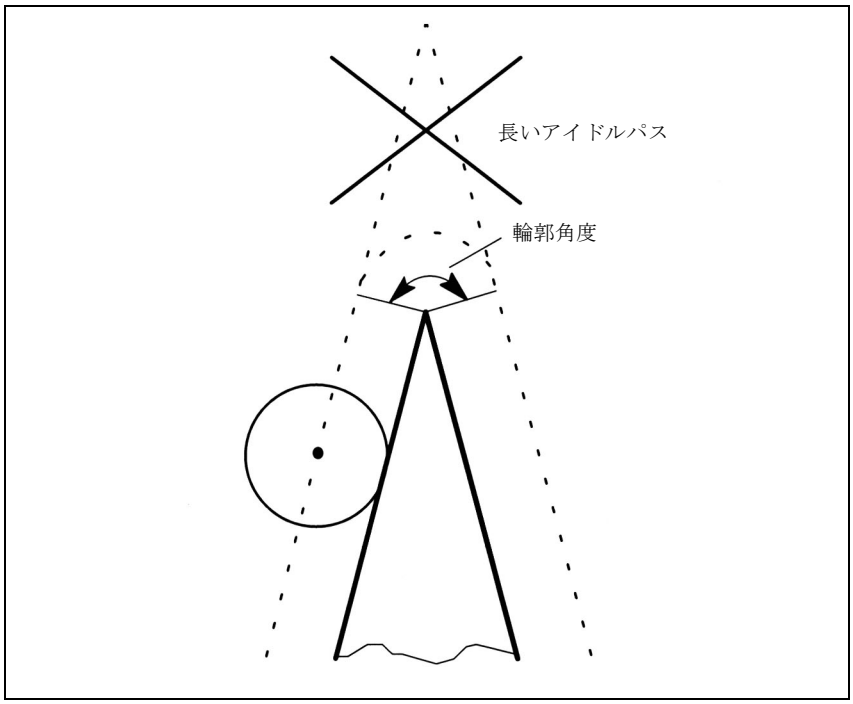


図 1.197 遷移円への自動切替えの例

非常に広い外側コーナ

外側コーナが非常に広い場合，G450（遷移円）と G451（インタセクション）を使った応答は，ますます類似します。この場合，遷移円の挿入はお奨めできません。5 軸加工のこれらの外側コーナで遷移円を挿入することが許されない一つの理由は，制御モード (G64) の速度を制限するからです。外側コーナが非常に広い場合，システムは，G450（適切な DISC を使って遷移円）から G451（インタセクション）へ自動的に切替わります。この自動切替え（遷移円→インタセクション）に続いて移動されることができる輪郭角度は，MD 20230: CUTCOM_CURVE_INSERT_LIMIT に定義できます（工具径補正を使ったインタセクション計算のための最大角度)。

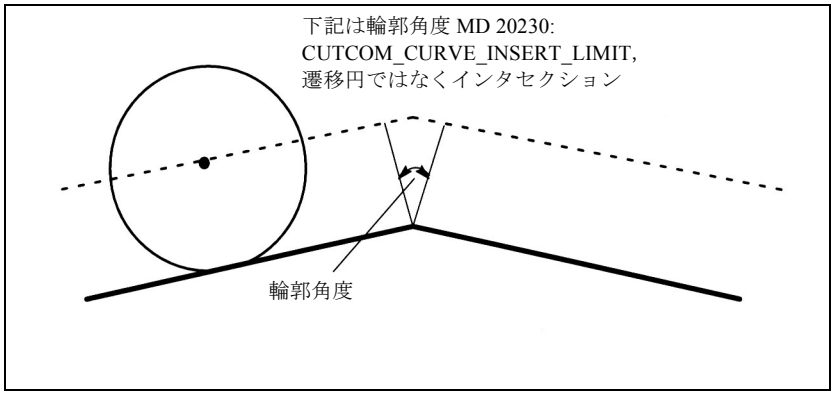


図 1.198 インタセクションへの自動切替えの例

■ 内側コーナでの補正

インタセクション

2 つの連続ブロックが内側コーナを形成する場合，2 つの等距離のインタセクションを見つける試みがなされます。インタセクションが見つかる場合，プログラムされた輪郭はインタセクションまで短縮されます（エンドで短縮される最初のブロック，初めに短縮される二番目のブロック）。

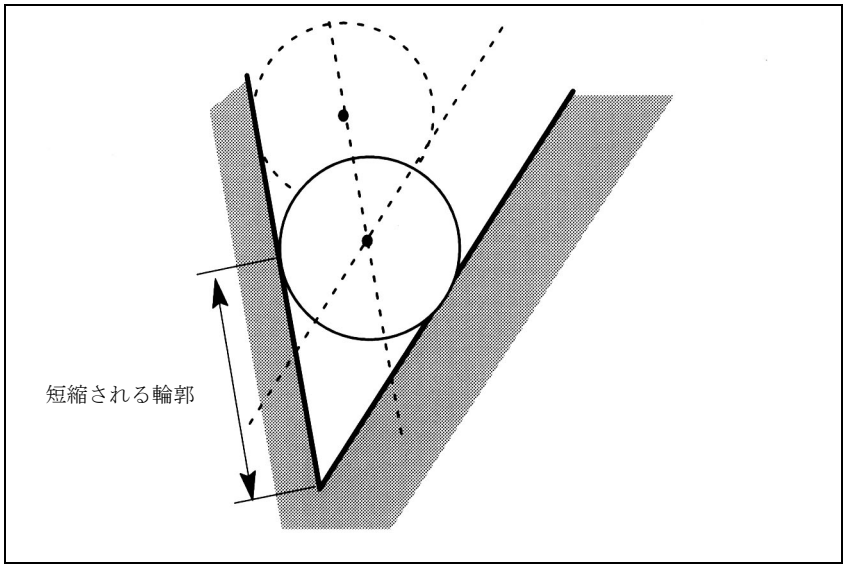


図 1.199 短縮される輪郭の例

インタセクションなし

いくつかのケースで，内側コーナのインタセクションは，2 つの連続ブロック間に見つかりません（図 1.200 参照）。

予想輪郭計算

インタセクションが 2 つの連続ブロック間に見つけられない場合，制御は次のブロックを自動的にチェックし，このブロックの等距離のインタセクションを見つけようとします（図 1.200：インタセクション S 参照）。次のブロックのこの自動チェック（予想輪郭計算）は，常に，マシンデータによって定義された最大数のブロックが達成されるまで実行されます。予想チェックに使用される最大ブロック数は，MD20240: CUTCOM_MAXNUM_CHECK_BLOCKS に入力できます。インタセクションが，チェックに定義されたブロック数内に見つけられない場合，プログラム実行は中断され，アラーム 10751 が出力されます。衝突監視が CDON で起動している場合，2 つ以上の連続ブロックのチェックのみが可能です（1.19.2「■磨耗－工具径補正（ツールパラメータ 15～20)」を参照)。

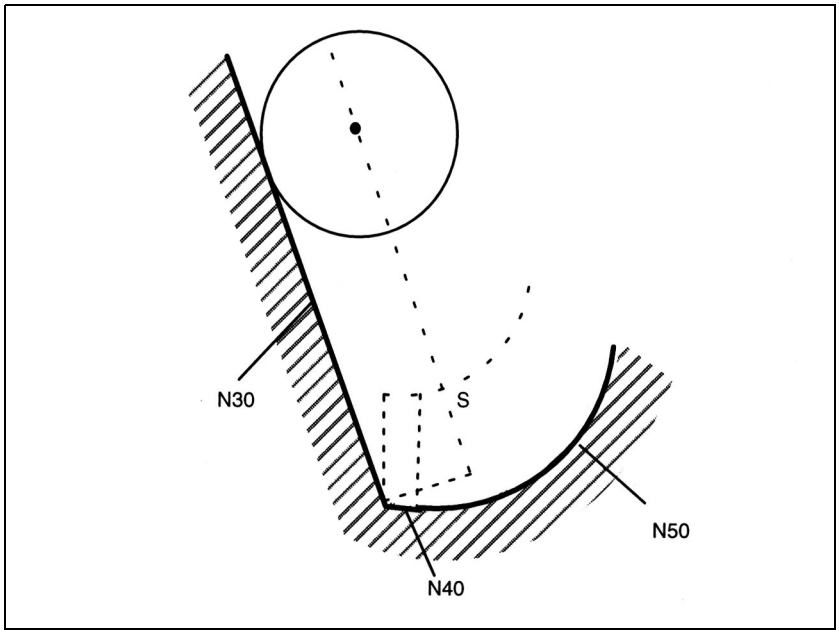


図 1.200 N30 と N40 の間にインタセクションがない場合，ブロック N30 とブロック N50 の間のインタセクションが計算される

複数のインタセクション

---> 衝突監視も参照（1.19.2「■磨耗－工具径補正（ツールパラメータ 15～20)」）

内側コーナのいくつかのケースで，等距離の複数インタセクションが，数個の連続ブロックで見つかります。これらの場合，最後のインタセクションが有効なインタセクションとして常に使用されます（図 1.201 参照)。予想チェックに使用される最大ブロック数は，MD 20240: CUTCOM_MAXNUM_CHECK_BLOCKS に入力できます。衝突監視が CDON で起動している場合，2 つ以上の連続ブロックのチェックのみが可能です（1.19.2「■磨耗－工具径補正（ツールパラメータ 15～20)」を参照)。

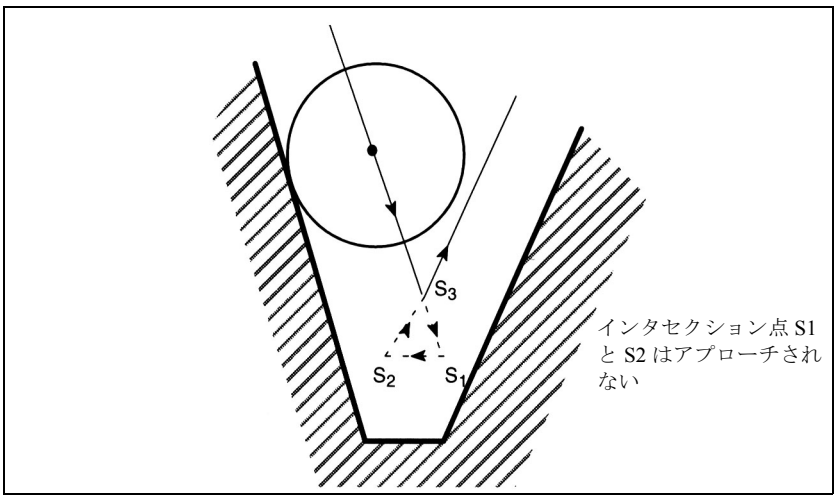


図 1.201 例：輪郭違反のない TRC を持つ内側コーナ（予想される 3 ブロック）

特別機能

次のブロックで複数のインタセクションが見つかる場合，次のブロックのスタートに最も近いインタセクションに適用されます。

■ 衝突監視とボトルネック検出

衝突監視（ボトルネック検出）は，非連続ブロックの等距離が交差するかどうかを調べます。インタセクションが見つかると，応答は，複数インタセクションを持つ内側コーナの場合と同じです。つまり，最後に見つけられたインタセクションが有効です（図 1.201 も参照）。

予想チェックに使用される最大ブロック数は，MD 20240: CUTCOM_MAXNUM_CHECK_BLOCKS に入力できます。

プログラミング

衝突監視は，プログラム内で起動または起動解除できます。

- CDON ... 衝突監視オン
- CDOF ... 衝突監視オフ

CDOF の場合，インタセクションの検索は 2 つの連続ブロックを調べるだけです。2 つの連続ブロックの間にインタセクションが見つからない場合，プログラム実行が中断され，アラーム 10751 が出力されます。外側コーナの場合，インタセクションは常に 2 つの連続ブロック間で見つかります。

ブロックの除外

連続しない 2 つのブロック間にインタセクションが検出される場合，補正面のこれらのブロック間にプログラムされた動作はどれも実行されません。除外されたブロックに含まれる他の全ての動作および実行指令（M コマンド，位置決め軸の移動など）は，NC プログラムにプログラムされるシーケンスでインタセクション位置で実行されます。

警告 10763

衝突またはボトルネック検出機能の結果，ブロックが除外された場合，警告 107663 が発せられ，プログラムは中断します。

特別機能

非連続ブロックのインタセクションがチェックされるとき，調べられるのはプログラムされた元の輪郭ではなく，対応付けられた計算済み等距離です。このため，「ボトルネック」が誤って外側コーナで検出される場合があります。この理由は，計算されたツールパスが DISC>0 の時，プログラムされた元の輪郭に等距離で走らないからです。

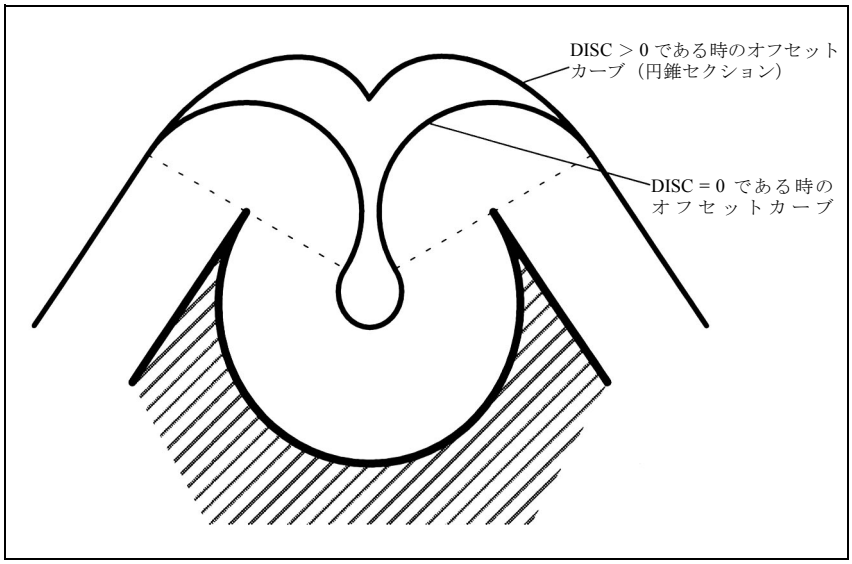


図 1.202 ボトルネック検出と外側コーナ

■ 可変補正値を持つブロック

補足条件

可変補正値は，全ての補間タイプにプログラムされます（円およびスプライン補間でさえ）。

符号チェンジ（およびこのように補正側でのチェンジ）もまた認められます。

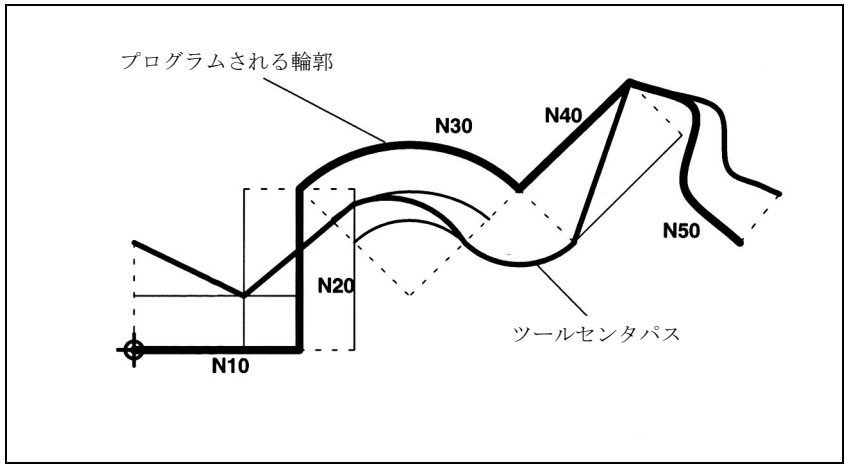


図 1.203 可変補正値を使った工具径補正

インタセクション計算

可変補正値を持つブロックのインタセクションが計算される時，オフセットカーブ（ツールパス）のインタセクションは，補正値が一定であるという仮定に基づいて常に計算されます。

可変補正値を持つブロックが，走行向きへ調べられる 2 つのブロックの最初のブロックである場合，ブロックエンドの補正値が計算に使用されます，そうでない場合，ブロックスタートの補正値が使用されます。

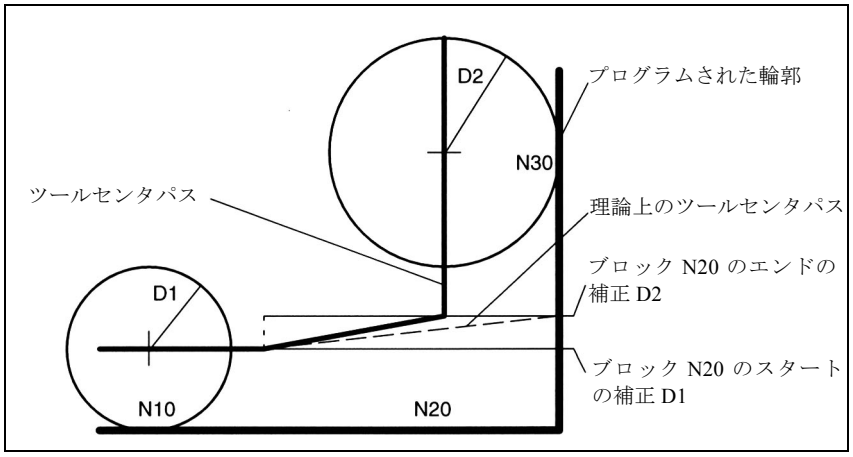


図 1.204 可変補正値を使ったインタセクション計算

制限

補正半径が，円の内側の加工のためのプログラム済み円半径より大きくプログラムされる場合，加工オペレーションは，アラーム 10758「可変補正値を使った曲率半径が小さすぎる」として拒否されます。

■ 工具径補正を一定に保つ（SW 4 以降）

意味

「工具径補正を一定に保つ」機能は，多数のブロックに対する工具径補正を抑制するために使用され，それによって，プログラム済みツールセンタパスと前ブロックの工具径補正によって設定され走行される実際のツールセンタパスの間の相違は，オフセットとして保留されます。

数個の移動ブロックが逆転点にラインフライス加工中必要である時，この方式を使用することは利点がありますが，工具径補正（フォローストラテジ）によって生成される輪郭には望ましくありません。

起動

「工具径補正を一定に保つ」機能は，G コード CUTCONON（カッタ補正が一定してオン）で起動し，G コード CUTCONOF（カッタ補正が一定してオフ）で起動解除されます。

CUTCONON と CUTCONOF はモーダル G コードグループを形成します。基本設定は CUTCONOF です。

工具径補正のタイプ（$2^1/_2$ D, 3D 正面フライス，3D 円周フライス加工）から独立して使用できます。

ノーマルのケース

通常，工具径補正は，オフセット抑制の起動前に既にアクティブで，オフセット抑制が再度起動解除される時やはりアクティブです。

CUTCONON 前の最終移動ブロック内で，ブロック終点のオフセット点がアプローチされます。オフセット抑制がアクティブである次の全ブロックは，オフセットなしに移動します。しかしそれらは，最終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。これらのブロックはどの補間タイプ（直線，円，多項式）でも持つことができます。

オフセット抑制の起動解除ブロック，つまり CUTCONOF を含むブロックはノーマルにオフセットされ，これは始点のオフセット点内でスタートします。1 つの直線ブロックが，前ブロックの終点，つまりアクティブ CUTCONON で最後にプログラムされた移動ブロックとこの点の間に挿入されます。

円面がオフセット面（縦の円）に垂直である円ブロックは，あたかも CUTCONON がプログラムされたかのように処理されます。オフセット抑制のこの暗黙の起動は，オフセット面に移動動作を含み，そしてそのような円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，縦の円は円周フライス加工中にのみ生じることができます。

例 :

```
N10                              ：ツール d1 の定義
N20 $TC_DP1[1,1]= 110            ：タイプ
N30 $TC_DP6[1,1]= 10.            ：半径
N40
N50 X0 Y0 Z0 G1 G17 T1 D1 F10000
N60
N70 X20 G42 NORM
N80 X30
N90 Y20
N100 X10 CUTCONON：オフセット抑制を起動する
N110 Y30 KONT   ：オフセット抑制の起動解除中
                 ：必要な場合バイパスサイクルを挿入する
N120 X-10 CUTCONOF
N130 Y20 NORM   ：TRC の起動解除でバイパスサイクルなし
N140 X0 Y0 G40
N150 M30
```

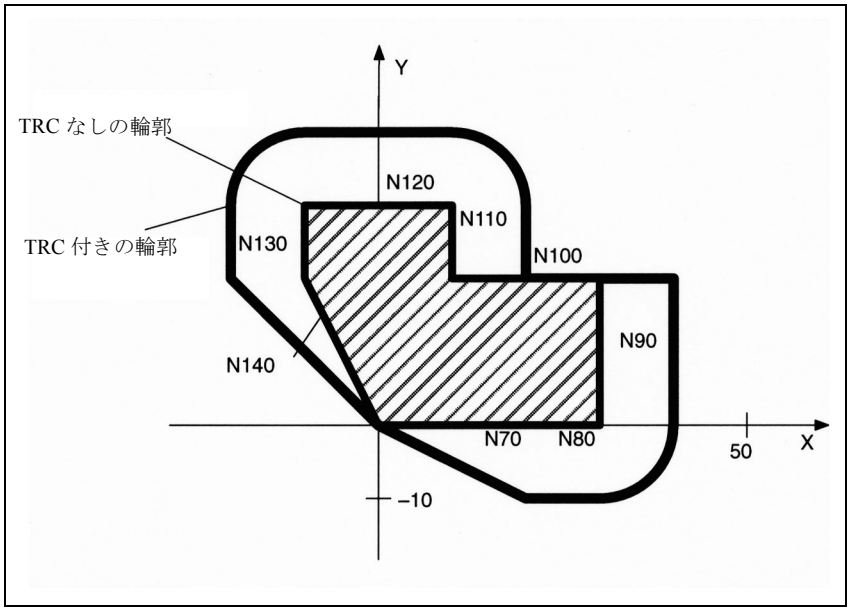


図 1.205 オフセット抑制のためのサンプルプログラム

特殊ケース

1. 工具径補正が起動していない (G40) 場合，CUTCONON は影響しません。アラームは生成されません。
   しかし G コードはアクティブなままです。工具径補正が，G41 と G42 を使って後のブロックで起動される時これは重要です。
2. 7 番目の G コードグループ（工具径補正：G40/G41/G42）を G コードの CUTCONON とチェンジすることが認められます。G40 へのチェンジは即時アクティブです。
   前ブロックを移動するのに使用されたオフセットが移動します。
3. CUTCONON または CUTCONOF が，アクティブ補正面に移動なしにプログラムされる場合，効果はそのような移動動作を持つ次のブロックまで遅れます。
4. CUTCONON がアクティブ工具径補正を使ってプログラムされ，プログラムの終了前にキャンセルされない場合，移動ブロックは，最後の有効なオフセットを使って移動します。同じことが，プログラムの最終移動ブロックの G41 または G42 の再プログラミングに当てはまります。
5. 工具径補正が G41 または G42 で起動し，CUTCONON もアクティブである場合，オフセットの起動は CUTCONOF を持つ次の移動ブロックまで遅れます。
6. CUTCONOF を使った輪郭での再スタート時，17 番目の G コードグループ（工具補正：NORM/KONT を使ったアプローチと後退) が評価されます。つまり，次のサイクルが KONT に必要な場合挿入されます。次のサイクルは，G41 または G42 での工具径補正の起動に関して同じ条件で挿入されます。
7. 抑制された工具径補正を持つブロック数は制限されます（MD: CUTCOM_MAXNUM_SUPPR_BLOCKS)。それを越える場合，加工は強制終了し，エラーメッセージがでます。CUTCONON 前の最終ブロックの内部ブロック処理が，再位置決め時に再開しなければならないので，制限が必要です。
8. 工具径補正が既にアクティブである時，G41 または G42 を再プログラミングした後の応答は，オフセット抑制に似ています。次のずれが適用されます。
   - 直線ブロックのみが認められます。
   - G41 または G42 を含む単一移動ブロックは，次のブロックにある始点のオフセットブロックで終わるように修正されます。したがって，ダミー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 を含む移動ブロックのシーケンスで，最終ブロックに適用されます。
   - 輪郭は，17 番目のグループの G コード（工具補正：NORM/KONT でのアプローチと後退）から独立して，常に NORM を使って再アプローチされます。

9. G41/G42 が連続移動ブロックで数回プログラムされる場合，全ブロックが，最終ブロックを除き CUTCONON に関して加工されます。
10. 輪郭抑制のタイプは，連続移動ブロックのシーケンスの第一移動ブロックでのみ評価されます。
    CUTCONON および G41 か G42 の両方が，最初のブロックにプログラムされる場合，輪郭抑制起動解除への応答は，CUTCONON によって決定されます。
    G41 から G42 またはその反対へのチェンジは，再スタート時オフセット側（輪郭の右または左）をチェンジする手段としてこの場合有意義です。
    オフセット側のチェンジ (G41/G42) も，輪郭抑制がアクティブであっても，後にプログラムできます。
11. 衝突監視およびボトルネック検出は，アクティブ輪郭抑制を使って全ブロックに対して起動解除されます。

■ 変更アラーム応答

工具径補正アラームが発生する場合,

- メインランでの処理は，到達できる次のブロックエンドで停止します。つまり，一般的に補間済み現行ブロックの最後で。
- 軸が静止した時，アクティブなルックアヘッドで。

前処理停止アラームおよびアクティブ工具径補正

SW 4 以降の応答

設定データ ($SC_STOP_CUTCOM_STOPRE) が挿入され，したがって工具径補正の応答は前のステータスと比較して変更されないまま残り，および／またはアラームがアクティブ工具径補正中に前処理停止に対して発せられ，プログラムがその値に依存して停止します。

ユーザはこのアラームを認知し，NC プログラムを NC スタートを使って継続するか，またはリセットを使って強制終了できます。

■ 多項式に対するインタセクションプロシージャ

アクティブな工具径補正を使った 2 つのカーブが外側コーナを形成する場合，含まれるカーブ（直線，円，多項式）のタイプに関係ないグループ（ツールオフセット：G450/G451 を使ったコーナ応答)

- 円錐カットが挿入されコーナを辿るか，または
- 含まれるカーブが，18 番目のグループ（ツール補正：G450/G451 に対するコーナ応答）の G コードに依存して，含まれるカーブのタイプ（直線，円，多項式）から独立して外挿されます。

インタセクションを 起動する G451 を使って見つけられないか，または 2 つのカーブが形成する角度が鋭い場合，挿入モードへの切替えは自動です。

MD: CUTCOM_INTERS_POLY_ENABLE を使って多項式のインタセクションプロシージャが可能です。この MD が非アクティブに設定される場合，応答はソフトウェアバージョン 3 以前の応答と同じです。

■ G461/G462：アプローチ／後退ストラテジ拡張（SW 5 以降）

いくつかの特殊ジオメトリの状況で，以前の実行と比べ，拡張されたアプローチと後退ストラテジが工具径補正の起動／起動解除に必要です（下図参照）。

（注）

次のセクションは，工具径補正の起動解除に関する状況を専ら参照します。アプローチに関する応答は事実上同じです。

例

G42 D1 T1 ：工具半径 20mm

...

G1 X110 Y0

N10 X0

N20 Y10

N30 G40 X50 Y50

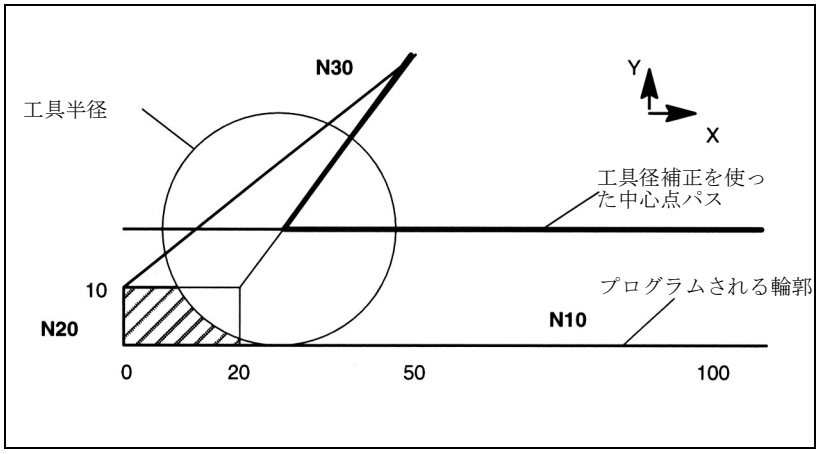


図 1.206 SW 4.x までの後退ストラテジ

アクティブツール補正を持つ最後のブロック（N20) は非常に短いので，オフセットカーブと先行ブロック（または以前のブロック）の間のインタセクションは，現在の工具半径にもはや存在しないかもしれません。したがって，次のブロックと先行ブロックのオフセットカーブの間でインタセクションを見つける試みがなされます。つまり例の中で N10 と N30 の間です。後退ブロックに使用されるカーブは，真のオフセットカーブではなく，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。インタセクションが見つかる場合，アプローチされます。この場合，図の中の陰影部分は，使用されるツールで加工可能であっても加工されません。

G460

G460 を使ったアプローチ／後退ストラテジは以前と同じです。

G461

インタセクションが，最終 TRC ブロックと前のブロックの間で可能である場合，このブロックのオフセットカーブは，中心点が非修正ブロックの終点と一致し，半径が工具半径に等しい円を使って拡張されます。

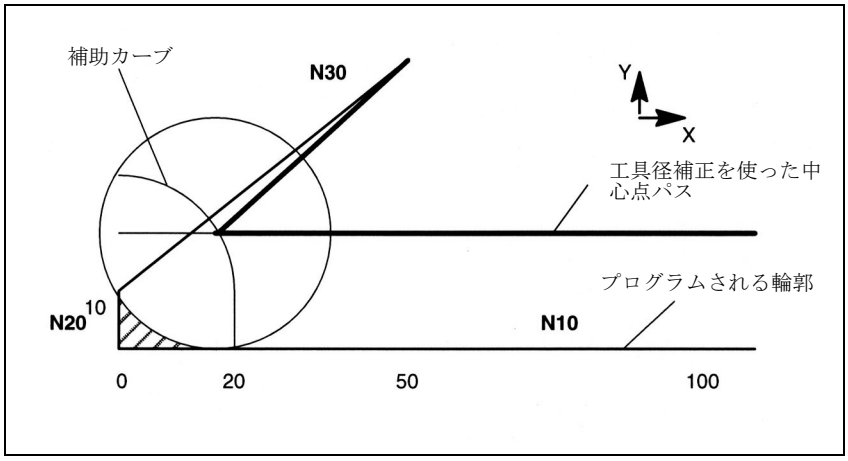


図 1.207 G461 に対する後退挙動

制御は，この円と先行ブロックの 1 つを交差させようとします。CDOF がアクティブである場合，検索はインタセクションが見つかると終了します。つまりシステムは，前のブロックとのさらなるインタセクションが存在するかどうかチェックしません。

CDON がアクティブである場合，検索は，最初のインタセクションが見つかった後さらにインタセクションを求めて継続します。

この方法で見つけられるインタセクション点は，先行ブロックの新しい終点と停止ブロックの始点です。挿入された円は，インタセクションを計算するために排他的に使用され，移動動作を生じません。

（注）インタセクションが見つからない場合，アラーム 10751 （衝突の危険）が出力されます。

G462

インタセクションが最終 TRC ブロックと先行ブロックの間で可能でない場合，G462（初期設定）を使って後退時，直線が，工具径補正を持つ最終ブロックの終点（ブロックはエンド接線によって拡張される）に挿入されます。下図参照。

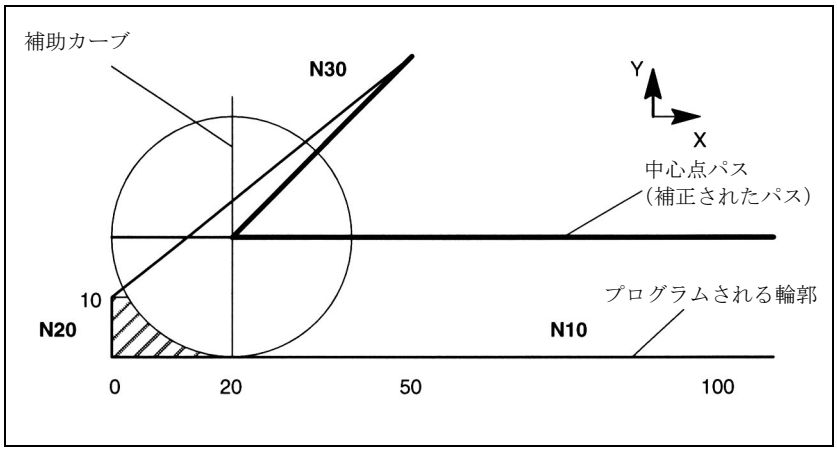


図 1.208 G462 に関する後退動作

インタセクションを求める検索は，G461 に対する処理と同じです。

G462 を使って，プログラム例の中で N10 と N20 が生成するコーナは，使用するツールで事実上可能な全範囲までクリアされません。しかし例の中で，N20 の左へのパート輪郭（プログラムされた輪郭と異なる）が，10m より大きい y 値を使ってさえ違反が許されない場合，この動作が必要です。

KONT がアクティブな（スタートまたは終点の丸い輪郭を走行する）場合，動作は，終点が輪郭の前または後ろにあるかどうかによって異なります。

輪郭の前の終点

終点が輪郭の前にある場合，後退動作は NORM の場合と同じです。この特性は，G451 の最終輪郭ブロックが直線または円で拡張されても変わりません。したがって，輪郭終点周辺の輪郭違反を避けるための追加バイパスストラテジは必要ありません。

輪郭の後ろの終点

終点が輪郭の後ろにある場合，円または直線は，G450/G451 に依存して常に挿入されます。この場合 G460 ～ G462 は影響しません。

この状況で，最終移動ブロックが先行ブロックとインタセクションを持たない場合，挿入された輪郭エレメントとの，またはバイパス円の終点からプログラムされた終点とのインタセクションが結果的に生じます。

挿入された輪郭エレメントが円 (G450) で，先行ブロックと交差する場合，これは，NORM および G461 で生成されるインタセクションと同じです。しかし，一般に円の残りのセクションはなお移動しなければなりません。インタセクションの計算は，後退ブロックの直線セクションにはもはや必要ありません。

二番目のケースで（挿入された輪郭エレメントと先行ブロックの間にインタセクションが見つからない場合），後退直線と先行ブロックの間のインタセクションが移動します。

したがって G461 と G462 がアクティブである場合，G460 と異なる動作は，NORM がアクティブであるか，または KONT を使った動作がジオメトリ条件により NORM と同じである場合のみ生じます。

（注）アプローチ動作は後退動作と左右対称です。
アプローチ／後退動作は，アプローチ／後退ブロックの G コマンドの状態によって決定します。したがってアプローチ動作は，後退動作から独立して設定できます。

例：

アプローチに G461 を使用するプログラム：

```
N10 $TC_DP1[1,1]=120 ：ツールタイプミル
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
```

■ 向き調整可能なツールホルダ（SW 4 以降）

■ 一般事項

はじめに

ツールの向きは，マシンツールの 1 クラスに関して変更が可能です（例えば異なるツールをロードする時)。しかしそのマシンの運転中，設定された向きは恒常的で移動中に変更できません。このためキネマティックス的向き変換（3，4 または 5 軸変換 TRAORI) は，そのようなマシンに必要でなく , また意味がありません。しかし，関与する数学でユーザが困らないように，向きチェンジで生じるツール長コンポーネントの変更を考慮に入れる必要があります。制御はこれらの計算を実行します。

必要なデータ

向き調整可能なツールホルダに対する制御工具補正が考慮に入れられる場合，次のものが存在しなければなりません。

- ツールデータ（ジオメトリ，摩耗 ...)
- ツールホルダデータ（向き調整可能ツールホルダのジオメトリのデータ）

ツールホルダ

制御に定義されるツールホルダは，「向き調整可能ツールホルダ」機能に指定されなければなりません。これは NC プログラムコマンドを使って行われます。

TCARR = m

m はツールホルダ数です。ツールホルダは，そのジオメトリを記述する関連ツールホルダデータブロックを持っています。

ツールホルダとそのブロックを起動すると，次の移動ブロックからすぐに実行できます。

割当てツール／ツールホルダ

以前アクティブであったツールが，新しいツールホルダへ割当てられます。

組合せ

制御の観点から，ツールホルダ数 m とツール数 T は自由に組合わせられます。しかし実際の適用で，いくつかの組合せが加工またはメカニカルな理由で除外できます。制御は，組合せが意味をなすかどうかはチェックしません。

向き調整可能ツールホルダ

例：ツールの向き調整用の 2 つの軸を使ったカルダンツールホルダ

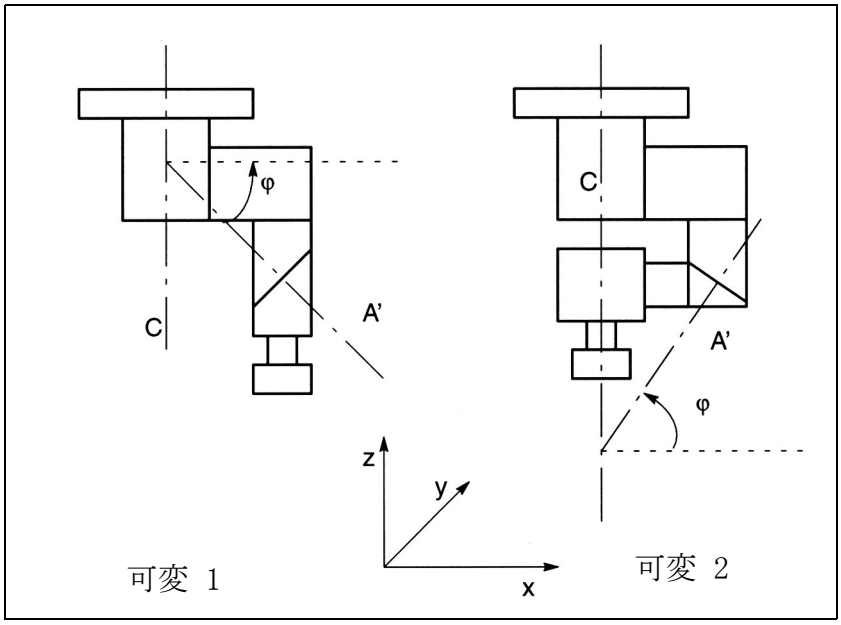


図 1.209 カルダンツールホルダ

ツールホルダキネマティックスの記述

向き調整可能ツールホルダのキネマティックスは，補正メモリに格納されている 17 REAL 値から成るデータブロックによって記述されます。詳細はセクション「向き調整可能ツールホルダ使用時のプロシージャ」に説明しています。ユーザはブロックのデータを編集できます。

ツールホルダデータブロックを処理する

2 つのオプションが可能です。

- パートプログラムからツールホルダデータブロックへ明示的に入力
- フレームからいくつかの値（角度）を自動転送
  この可変性は，TCOFR：(Tool Carrier Orientation FRame：ツールキャリア向き調整フレーム）がツールホルダ選択に指定される必要があります。ツール長の計算に使用されるツール向きは，ツールが変更されるその時点でアクティブなフレームから再度決定されます。

Z 方向を基準にした向き調整

G 機能 TOFRAME は，このフレーム中の Z 方向も現在のツールの向きであるようにフレームを定義します。

下記の場合	仕様
ツールホルダがアクティブでない	Z 方向きが新しいフレームにある。 旧 Z 方向と同じ G17 で 旧 Y 方向と同じ G18 で 旧 X 方向と同じ G19 で
向きチェンジなしツールホルダがアクティブ	

アクティブフレームに対する TCOABS

絶対ツールホルダの向き調整は

TCOABS：（Tool Carrier Orientation ABSolut：ツールキャリア向き調整絶対）によって設定されます。

工具長補正のため考慮に入れる向き調整は，アクティブフレームの向き調整から独立しています。

TCOABS または TCOFR の内 1 つのみが有効です。

フレームの変更

ユーザはツールの選択後フレームを変更できます。これは工具長補正のコンポーネントに影響しません。

ツールホルダデータの角度：

ツールホルダデータに記憶された回転角度は，フレームによって定義される回転角度により影響されません。TCOFR から TCOABS への変更時，ツールホルダデータの元の（プログラムされた）角度または回転が再度起動されます。

工具径補正

CUT2D または CUT3DFS がアクティブである時，TRC は現在のツール向き調整を考慮に入れます。

回転しないツールホルダ

内部的に使用されるツール方向調整は，アクティブ面のみに依存します (G17-G19)。

CUT3DC および CUT3DF

正面フライス用の 3D 工具補正

アクティブ 5 軸変換を持つ正面フライス用の 3D ツールは，「向き調整可能ツールホルダ」機能に影響されません。向き調整情報はアクティブなキネマティックス 5 軸変換によって決定されます。

他の全ての工具補正タイプ

CUT3DC および CUT3DF を除いて，全て G コードグループ 22 の補正タイプです。

応答は，補正に使用される面に関して同じままです。これは，アクティブフレームからのツール向き調整から独立して決定されます。

CUT2DF および CUT3DFF の場合，TRC に使用される補正面は，現在のツール向き調整から独立したフレームから決定されます。アクティブ面 (G17/G18/G19) が考慮されます。

制限されるツールホルダ向き調整

定義されたツールホルダキネマティックスで到達されない向き調整がフレームによって定義される場合，アラームがでます。次のキネマティックスはどの向き調整も達成できません。

- キネマティックスの定義に必要な 2 つの回転軸が互いに垂直でなく，ツール向きを定義するツール軸が第二回転軸に垂直でないか，または
- 2 つに満たない軸が定義された時。

曖昧性

フレームによって指定される特定のツール向き調整は，一般的に，2 つの回転角度ペアによって 2 つの角度に設定できます。制御は，回転角度がプログラムされた回転角度になるだけ近い 1 つの設定を選択します。

ツールホルダデータに角度を保存する

曖昧性が生じる場合，ツールホルダデータにフレームから求められる近似角度を保存する必要があります。セクション「曖昧性」を参照。

■ プロシージャ

キネマティックスチェーンの表示

キネマティックスチェーンのコンセプトは，基準点とツールチップの間のキネマティックス的関係を記述するために使用されます。

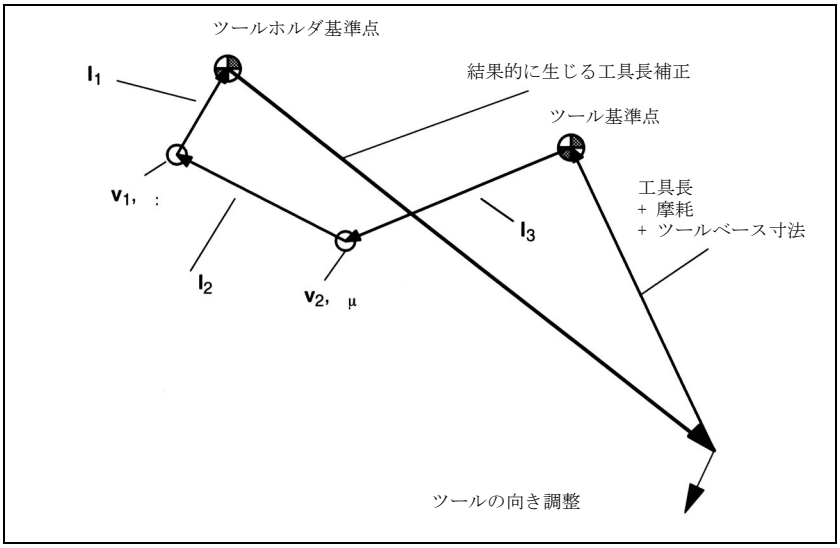


図 1.210 向き調整を使ってツールを記述するキネマティックスチェーン（一般的なケース）

チェーンは，略図の中のツールホルダデータブロックに必要な全てのデータを指定します。特定のキネマティックスの具体的なケースを記述するには，チェーンの関連コンポーネントが，実際のベクトル，長さおよび角度に割当てられなければなりません。チェーンは最大配列を表します。個々のコンポーネントは，単純なケースではゼロかもしれません。

チェーン／データブロックのコンポーネント

ツールホルダデータブロックの全ての値は，タイプ REAL です。それらは全てゼロにプリセットされます。

記述	NCK 変数
オフセットベクトル l_1 の x コンポーネント	$TC_CARR1
オフセットベクトル l_1 の y コンポーネント	$TC_CARR2
オフセットベクトル l_1 の z コンポーネント	$TC_CARR3
オフセットベクトル l_2 の x コンポーネント	$TC_CARR4
オフセットベクトル l_2 の y コンポーネント	$TC_CARR5
オフセットベクトル l_2 の z コンポーネント	$TC_CARR6
回転軸 v_1 の x コンポーネント	$TC_CARR7
回転軸 v_1 の y コンポーネント	$TC_CARR8
回転軸 v_1 の z コンポーネント	$TC_CARR9
回転軸 v_2 の x コンポーネント	$TC_CARR10
回転軸 v_2 の y コンポーネント	$TC_CARR11
回転軸 v_2 の z コンポーネント	$TC_CARR12
回転角度 $_1$（度）	$TC_CARR13
回転角度 $_2$ (度）	$TC_CARR14
オフセットベクトル l_3 の x コンポーネント	$TC_CARR15
オフセットベクトル l_3 の y コンポーネント	$TC_CARR16
オフセットベクトル l_3 の z コンポーネント	$TC_CARR17

回転を記述するデータブロックは，回転軸の向きをその基本設定と角度 $_1$ または $_2$ に指定するベクトル V_1 または V_2 で構成されます。回転角度は，時計回りに回転され，回転ベクトルの向きから見て，正向きへカウントされます。

2 つのツールホルダ角度 $_1$ および $_2$ は，現在選択されているアクティブ面 (G17-G19) から独立してフレームを使って決定されます。

基本設定（2 つの角度 -- と --- がゼロに等しい）のツール向き調整は，（標準ケースに関して）次の通りです：

G17 に対して：Z に並行,

G18 に対して：Y に並行

G19 に対して：X に並行

例

関連データ値は，示されるツールホルダに対して下記に述べられています。

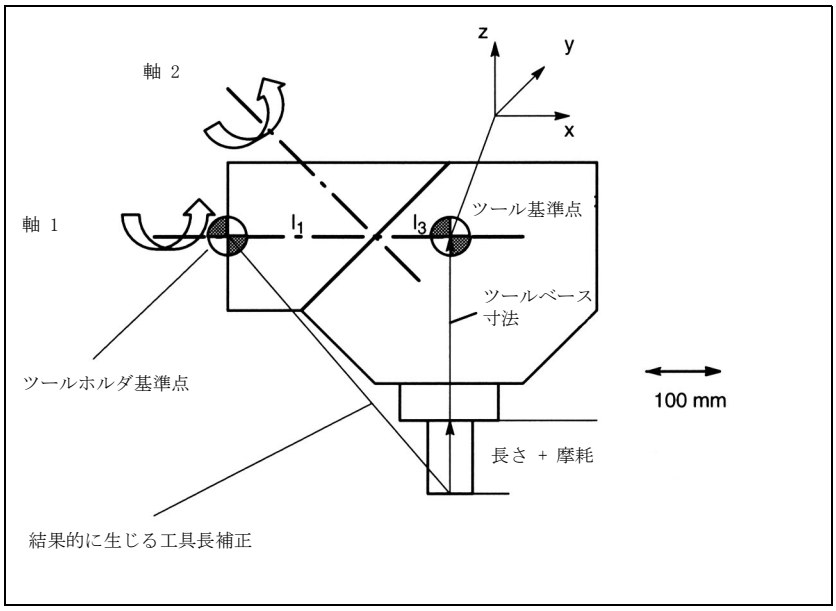


図 1.211 ツールホルダデータの割当て

割当ては，データブロックの次の値に対してなされました。

- 2 つの回転軸が一点で交差する。したがって l_2 の全てのコンポーネントはゼロです。
- 第一回転軸は x-z 面にあり，第二回転軸は x 軸に平行。これらの条件は向き v_1 と v_2（それらがゼロでない場合，長さは無関係）を定義します。
- ツールホルダの基準点 y は，2 つの回転軸から見て負の x 向き 200mm のところにある。この条件は l_1 を定義します。

記述	NCK 変数	値
オフセットベクトル l_1 の x コンポーネント	$TC_CARR1	-200
オフセットベクトル l_1 の y コンポーネント	$TC_CARR2	0
オフセットベクトル l_1 の z コンポーネント	$TC_CARR3	0
オフセットベクトル l_2 の x コンポーネント	$TC_CARR4	0
オフセットベクトル l_2 の y コンポーネント	$TC_CARR5	0
オフセットベクトル l_2 の z コンポーネント	$TC_CARR6	0
回転軸 v_1 の x コンポーネント	$TC_CARR7	1
回転軸 v_1 の y コンポーネント	$TC_CARR8	0
回転軸 v_1 の z コンポーネント	$TC_CARR9	0
回転軸 v_2 の x コンポーネント	$TC_CARR10	1
回転軸 v_2 の y コンポーネント	$TC_CARR11	0
回転軸 v_2 の z コンポーネント	$TC_CARR12	1
回転角度 $_1$（度）	$TC_CARR13	0
回転の角度 $_2$（度）	$TC_CARR14	0
オフセットベクトル l_3 の x コンポーネント	$TC_CARR15	-100
オフセットベクトル l_3 の y コンポーネント	$TC_CARR16	0

記述	NCK 変数	値
オフセットベクトル l_3 の z コンポーネント	$TC_CARR17	0

ベース寸法の x コンポーネント：0

ベース寸法の y コンポーネント：0

ベース寸法の z コンポーネント：250

説明

例に選ばれたツールホルダのキネマティックスは，2 つの回転軸が 45 度の角度を成しており，向き調整がどの値でも取れないことを意味します。具体的な条件で，この例は負の X コンポーネントを持つ向き調整の表示を認めません。

（注）

必要なデータは，ツールホルダのジオメトリから明確に決定できません。つまり，ユーザ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。したがって前の例では，第二軸までベース寸法に 1 つのみの z コンポーネントを指定することが可能です。この場合 l_2 はゼロではなく，第二軸のこの点と第一軸の先の点の間の距離のコンポーネントを含みます。第一軸の点も自由に選べます。どの点が選択されるかによって，l_1 は基準点（これも自由に選べる）が達せられるように選択されなければなりません。

規則

一般的に，軸の回転によって変わらないベクトルのコンポーネントは，回転の「前」と「後」のベクトルに分散できます。

ゼロベクトル

ベクトル v_1 と v_2 はゼロでも構いません。この時，関連回転角度（アクティブフレームから明示的にプログラムされるかまたは計算される）も，回転軸の向きが定義されないので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，ツールホルダが起動される時アラームがでます。

2 つに満たない回転軸

記述されるツールホルダが 1 つの面でツールを回転させるだけの場合，回転角度を定義しないオプションが有意義です。したがって，感知する最低データブロックは，ツールホルダデータに 0 でない 1 つのみの入力を含みます。つまり，回転角度が 1 つのフレームから決定される場合，軸に平行な回転角度を記述する v_1 または v_2 コンポーネントの 1 つの値。

さらに特殊なケース

2 つのベクトル v_1 と v_2 は互いに直線です。しかし向き調整の自由度は失われます。つまりそのようなキネマティックスは，1 つのみの回転軸が定義される場合と同じです。全ての可能な向き調整は 1 つの円錐シース上にあります。円錐シースは，ツール向き調整が t および v_1 か v_2 が互いに直線になる場合，直線に変形します。したがって，向き調整の変更はこの特殊ケースで可能ではありません。円錐シースは，ツール向き調整 t および v_1 か v_2 が互いに垂直である場合，円面へ変形します（つまり，全ての向き調整が 1 つの面に可能です）。

2 つのベクトル v_1 と v_2 がゼロになるのは認められます。向き調整変更は，この時可能ではありません。この特殊ケースで，ゼロでない様々な長さ l_1 と l_2 は，追加工具長補正として働き，その場合個々の軸のコンポーネントは，面 (G17 ～ G19) の変更によって影響されません。

ツールホルダデータブロック数

使用可能なツールホルダデータブロック数は，NCK で MD: 18088 $MN_MM_NUM_TOOL_CARRIER に定義されます。

■ プログラミング

ツールホルダ選択

ツールホルダは TCARR = m を使って選択され，m はツールホルダ数です。

ツールホルダデータブロックへのアクセス

次のアクセスがパートプログラムから可能です。

$TC_CARRn[m] = 値　　：これはツールホルダ m に対する n パラメータの前の値を新しい値「値」と取替えます。

値 = $TC_CARRn[m]　　：値は REAL パラメータでなければなりません。ツールホルダ m のパラメータは，ツールホルダデータブロックが既に定義されている場合，読取られることができます。

ツールホルダデータブロック数は，MD 18088: $MN_MM_NUM_TOOL_CARRIER（定義可能なツールホルダデータブロック合計数）によって定義される範囲内になければなりません。NC チャンネルに対して，この数はアクティブツールホルダデータブロックチャンネル数で割って定義できます。例外：MD 28085: $MC_MM_LINK_TOA_UNIT が使用され，標準から偏向する値を選択する場合。

ユーザが設定しない場合，値は 0 にプリセットされます。

全てのツールホルダデータブロックをキャンセルする

パートプログラム内から，1 つのコマンドを使って全部のツールホルダデータブロックの全ての値を削除することができます。

$TC_CARR1[0] = 0

起動

ツールホルダとツールの両方が起動されると，ツールホルダはアクティブになります。ツールホルダのみの選択は効果がありません。

ツールホルダ選択の効果は，G コードまたはモーダル G コードグループ TCOABS/TCOFR （ツールホルダ変更中に応答）に左右されます。TCOABS を使って，回転角度との両方に対してツールホルダデータに保存された値が，ツール向き調整の決定に使用されます。TCOFR を使って，これらの 2 つの角度は現在のフレームから決定されます。しかしツールホルダデータに保存されている値は変更されません。これらは，回転角度が 1 つのフレームから計算される時に生じる曖昧さの解決にも使用されます。ここでは，プログラムされた角度から最も偏向しない角度が，様々な可能な角度から選択されます。

TCOABS/TCOFR グループの中の G コードを変更すると，ツールホルダがアクティブである時，工具長補正の再計算をします。

■ 向き（オリエンテーション）に関する補足条件

向きのオプション

一定のキネマティックスを記述する所定のデータブロックに対して，可能な全ての空間的な向きは，次の条件が満たされる場合のみ表示できます。

- 回転軸を記述する 2 つのベクトル v_1 と v_2 が定義されなければなりません。つまりゼロであってはなりません。
- 2 つのベクトル v_1 と v_2 は互いに垂直でなければなりません。
- ツールの向きは，第二回転軸に垂直でなければなりません。

実行不可能な向き

これらの条件が満たされず，アクティブフレームによって達成できない向きが TCOFR で要求される場合，アラームがでます。

回転ベクトルと回転角度の関係

回転方向を記述するベクトル v_1 または v_2 がゼロの場合，関連回転角度もゼロでなければなりません。そうでない場合アラームが出力されます。アラームは，ツールホルダが起動されるまで，つまりツールホルダがチェンジされるまで発せられません。

（注）向き調整可能ツールホルダと向き変換を結合しないでください。
参照：3.1 3～5 軸刃先方向制御（F2）
向きをオーバレイすると，望ましくない結果を生じます。
例：
ツールのみが 1 つの長さコンポーネントを持ち，基本設定で Z 向きを指します。ツールホルダは，ツールを Y 軸回りを 45 度回転させ，回転後も同じ X と Y 値を持つコンポーネントを持ちます。A3 = -1 C3 = -1 が，起動された 5 軸変換を使って向きベクトルとして現在プログラムされる場合（これは Y 軸回りの 45 度回転にも対応する），ツールはさらに 45 度回転させられ，現在，ツールは水平面に位置し，負の X 軸方向を指します。つまりツールは同じ位置に留まりません。回転はこのようになされますが，しかしツールチップはその位置を保持します（ただし，直線軸動作がプログラムされていないことが条件）。

ツールオフセット（微）および向き調整可能なツールホルダ

ツールオフセットとツールホルダの結合は認められません。アクティブツールホルダに対するツールオフセット（微）起動，およびその反対にアクティブツールオフセット（微）でのツールホルダの起動は，アラームを発します。

補助機能出力

定数および／または M コードがオプションとして出力されるために，MD 22530: $MC_TOCARR_CHANGE_M_CODE が設定できます。コード番号はツールホルダ番号から得られます。

参照： 1.9 補助機能出力（H2）

■ リセット中の制御応答，プログラムスタート，Repos

RESET，プログラムスタート

MD 20126: $MC_TOOL_CARRIER_RESET_VALUE を使って，リセットまたはプログラムスタート中に自動的にツールホルダを選択できます。処理は，MD 20120: $MC_TOOL_RESET_VALUE を介して制御されるツール選択に似ています。リセットまたはプログラムスタート中の応答は，ツール選択に関して，MD 20110 $MC-RESET_MODE_MASK または MD 20112 $MC_START_MODE_MASK の中の同じビット（ビット 6）を使って制御されます。

ASUP，Repos

ツールホルダは，非同期サブプログラム ASUP 内で変更できます。中断されたプログラムが REPOS を使って継続される場合，新しいツールホルダのアプローチ動作が考慮に入れられ，プログラムはこの動作で継続されます。ここでの処理は，ASUP 内のツールチェンジに似ています。

■ G91 エクステンション

前提条件，基本

G91 を使った増分寸法プログラミングは，オフセット値が，ツールオフセットが選択される時プログラムされた増分寸法に加えて移動するように，SW バージョン 4.3 以降定義されます。

スクラッチなどのアプリケーションに対して，増分寸法を使って，プログラムされたパスのみを走行する必要があります。起動されたツールオフセットは移動しません。

シーケンス

- ツールチップを使ってワークをスクラッチする
- ツールオフセットによって実際位置を減じた後，それを基本フレームに保存する（実際値を設定する）
- ゼロ位置から相対的に移動する

起動

設定データ SD 42442: TOOL_OFFSET_INCR_PROG を使って，修正済みゼロオフセットが，FRAME および軸のインクリメンタルプログラミングで移動するか否か，またはプログラムされたパスのみが移動されるか否かをユーザは定義できます。

補足条件

応答が，プログラム終了およびリセット (MD 20110: RESET_MODE_MASK, Bit 6=1) 後なお有効であるように設定する場合，およびインクリメンタルパスがパートプログラムブロックにプログラムされる場合，オフセットを，常にプログラムされたパスに追加して移動します。

（注）この設定をすると，パートプログラムは絶対プログラミングによって始まります。

■ ツールオフセットの特殊機能

設定データ SD 42900 - SD 42940 は，ツール長と摩耗符号の計算を制御するのに使用できます。

同じことが，ジオメトリ軸のミラーリング時または設定データを介した加工面の変更時，摩耗コンポーネントの応答に適用されます。

参照： YS840DI ユーザーズマニュアル・プログラミング編

（注）下記の摩耗値が参照される場合，これは，摩耗値 （$SCPX3 ～ $SCPX11) とセットアップ値 ($ECPX3 ～ $ECPX11) を使って，実際の摩耗値 ($TC_DP12 ～ $TC_DP20) および合計オフセットの合計を参照します。合計オフセットの詳しい情報は，/FBW/, Description of Functions, Tool Management を参照してください。

必要な設定データ

- SD 42900 MIRROR_TOOL_LENGTH（長さコンポーネントおよびツールベース寸法のコンポーネントのミラーリング）
- SD 42910 MIRROR_TOOL_WEAR（ツール長コンポーネントの摩耗値のミラーリング）
- SD 42920 WEAR_SIGN_CUTPOS（摩耗コンポーネントの符号評価）
- SD 42930 WEAR_SIGN（摩耗寸法の符号の反転）
- SD 42940 TOOL_LENGTH_CONST（ツール長コンポーネントのジオメトリ軸への割当て）

a) ツール長のミラーリング (SD 42900 MIRROR_TOOL_LENGTH)

設定データがゼロ以外の場合：

ツール長コンポーネント ($TC_DP3, $TC_DP4 and $TC_DP5)，およびその関連軸がミラーリングされるツールベース寸法のコンポーネント ($TC_DP21, $TC_DP22 and $TC_DP23) もまたミラーリングされます。符号反転によって

摩耗値はミラーリングされません。これらが同様にミラーリングされる場合，設定データ $SC_MIRROR_TOOL_WEAR はイネーブルでなければなりません。

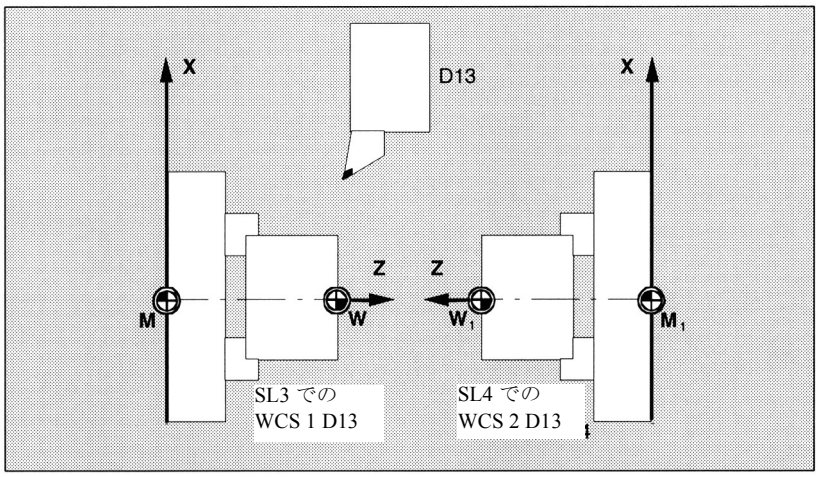


図 1.212 ツール長ミラーリングの例

SD 42910 MIRROR_TOOL_WEAR

設定データがゼロ以外の場合：

その関連軸がミラーリングされるツール長コンポーネントの摩耗値もまたミラーリングされます（符号反転によって）。

b) 摩耗符号評価 (SD 42920 WEAR_SIGN_CUTPOS)

設定データがゼロ以外の場合：

関連するツール点方向を持つツール（旋削バイトおよび研削ツール - ツールタイプ 400-599）の場合，摩耗コンポーネントの符号評価は，加工面のツール点方向に依存します。このデータ設定は，関連するツール点方向を持たないツールタイプに影響しません。

次の表で，その符号が SD 42920（0 でない）によって反転される寸法には × 印が記されています。

切削エッジ位置	長さ 1	長さ 2
1		
2	×	
3	×	×
4		×
5		
6		
7	×	
8		×
9		

（注）SD 42920 および 42910 は独立しています。例えば，測定パラメータが両設定データによって変更される場合，符号は変わりません。

SD 42930 WEAR_SIGN

設定データがゼロ以外の場合：

全ての摩耗寸法の符号を反転します。これはツール長，および工具半径，丸め半径など全てに影響します。

正の摩耗寸法を入力すると，ツール長が短くなります。

例：

次のセクション「修正された設定データの起動」を参照。

c) ツール長および加工面の変更 (SD 42940 TOOL_LENGTH_CONST)

ツール長コンポーネント（長さ，摩耗およびツールベース寸法）とジオメトリ軸の間の割当ては，加工面が変更 (G17 ～ G19) されても修正されません。

次の表は，旋削ツールと研削ツール（ツールタイプ 400 ～ 599）のツール長コンポーネントとジオメトリ軸の間の割当てを示します：

加工面	長さ 1	長さ 2	長さ 3
G17	Y	X	Z
G18*)	X	Z	Y
G19	Z	Y	X
-G17	Z	X	Y
-G18	Y	Z	X
-G19	X	Y	Z

*) 記載された 6 つの値の 1 つに等しくなく 0 でない各値は，G18 でのように評価されます。

次の表は，他の全てのツール（ツールタイプ <400 および >599）のツール長コンポーネントとジオメトリ軸の間の割当てを示します：

加工面	長さ 1	長さ 2	長さ 3
G17*)	Z	Y	X
G18	Y	X	Z
G19	X	Z	Y
-G17	Z	X	Y
-G18	Y	Z	X
-G19	X	Y	Z

*) 記載された 6 つの値の 1 つに等しくなく 0 でない各値は，G17 でのようになります。

（注）表の中の表示で，ジオメトリ軸 1 ～ 3 は X，Y，Z の名称を持つと仮定します。軸識別子でなく軸の順位が，オフセットと軸の間の割当てを決定します。
3 つのツール長コンポーネントが，上記 6 つの異なるタイプに配置できます。

■ 修正された設定データの起動

上述の設定データが修正される場合，ツールコンポーネントはツールエッジが次に選択されるまで再反映されません。ツールが既に有効で，このツールのデータが反映されることになる場合，そのツールが再度選択されなければなりません。

例 :

N10 $SC_WEAR_SIGN = 0	摩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 1 mm の摩耗寸法，結果的に生じる工具半径 101 mm
N100 T1 D1 G41 X150 Y20	
....	
N150 G40 X300N10	
....	
N200 $SC_WEAR_SIGN = 1	全ての摩耗値の符号反転：新たな半径 99 mm は新たな選択 (D1) で起動される。D1 がない場合，半径は 101 mm のまま。
N300 D1 G41 X350 Y-20	
N310	

結果的に生じるツール長が，軸のミラーリングステータスの変更により修正される場合，同じことが適用されます。修正されたツール長コンポーネントを起動するには，ツールがミラーコマンドの後再度選択されなければなりません。

■ 回転可能なツールホルダおよび新しい設定データ

設定データ SD 42900 - SD 42940 は，有効な回転可能ツールホルダのコンポーネントに影響しません。しかし，回転可能なツールホルダに関する計算は，常に，結果的に生じる長さ（ツール長 + 摩耗 + ツールベース寸法）を持つツールに対して可能です。結果的に生じる合計長の計算を使って，設定データによって生じる全ての修正が可能です。

（注）回転可能なツールホルダが使用される場合，ミラーリングされない基本システムに全ツールを定義することが実用的です。たとえミラーリングされる加工にのみ使用されるツールさえもです。ミラーリングされた軸で加工する場合，ツールホルダは，ツールの実際位置が正しく記述されている通りに回転されます。全てのツール長コンポーネントは，各軸のミラーリングステータスに依存して，設定データを介する各コンポーネント評価の制御の必要なく，正しい向きに自動的に働きます。
回転ツールの物理的手段がマシンに用意されず，様々な向きを持つツールが恒常的にインストールされる場合，回転可能なツールホルダの機能性を使用するのが実用的です。ツールの寸法決めは基本的向きに一貫して実行され，加工に適する寸法が，仮想ツールホルダを回転させることによって決定できます。

■ 合計オフセットとセットアップオフセット（SW 5 以降）

■ 一般事項

合計オフセットは，加工中プログラム可能なプロセスオフセットとして取り扱われることができ，ワークを指定寸法から偏向させる全てのエラーサイズ（摩耗を含む）で構成されます。

合計オフセットは，一般化された摩耗形式です。

これらは切削エッジデータのコンポーネントです。合計オフセットのパラメータは，切削エッジのジオメトリのデータを参照します。

合計オフセットのオフセットデータは，DL 番号（DL：位置依存，使用位置に関するオフセット）によってアドレス指定されます。

対照的に，D 番号の摩耗値は，切削エッジの物理的摩耗を記述します。つまり特殊状況では，合計オフセットは切削エッジの摩耗に一致することが可能です。

合計オフセットは，アクティブまたは非アクティブツール管理を使って，またはフラット D 番号機能を使って，一般的な使用が意図されています。

マシンデータが合計オフセットを類別するために使用されます。

- 合計オフセット（微）への類別
- 合計オフセット（粗）（セットアップオフセット）への類別

セットアップオフセットは，加工の前にセットアップエンジニアが入力するオフセットです。これらの値は，NCK に別に保存されます。オペレータは，その後 MMC を介して「合計オフセット（微）」のみアクセスします。「合計オフセット（微）」および「合計オフセット（粗）」は，NCK で内部的に追加されます。この値は，後に合計オフセットとして参照されます。

■ 機能の記述

いくつかの合計オフセット（DL 番号）は，D 番号別に定義できます。これによって，ユーザは，例えばワーク位置依存オフセット値を決定し，それらを切削エッジへ割当てることができます。合計オフセットは摩耗と同じ効果を持ちます。つまり D 番号のオフセット値へ追加されます。

データは，通常 D 番号へ割当てられます。

ユーザはマシンデータに次の設定を定義できます。

- アクティブ合計オフセット
- NCK メモリで生成される DL データセットの最大量の定義
- D 番号へ割当てられる DL 番号の最大量の定義
- 合計オフセット（微／粗）がデータバックアップ中保存されるかどうかの定義
- 下記の場合どの合計オフセットが起動されるかの定義
  - 新しい切削エッジオフセットが起動される時
  - オペレータパネル RESET が実行される時
  - オペレータパネル START が実行される時，または
  - プログラムが終わりに達した時。

名称は，ツールおよび切削エッジに対応するマシンデータの論理に合わせられます。

セットアップオフセットと合計オフセット（微）は，システムパラメータと OPI サービスによって読取りと書込みが可能です。

（注）ツール管理がアクティブである時，マシンデータを使用して，プログラムされたツールチェンジ中，起動されたツールの合計オフセットが変更されずそのまま残るか，またはゼロに設定されるかどうかを定義できます。

オフセットパラメータ $TC_DPx の概要

次の一般的データは，切削エッジを記述するために以前定義されました：

$TC_DP1	ツールタイプ
$TC_DP2	切削エッジ位置

ジオメトリおよび摩耗のパラメータ

ツールジオメトリのオフセットは，パラメータ $TC_DP3 ～ $TC_DP11 へ割当てられます。パラメータ $TC_DP12 ～ $TC_DP20 を使って，ユーザはこれらの各パラメータに対する摩耗を命名できます。

ジオメトリ	摩耗	長さオフセット
$TC_DP3	$TC_DP12	長さ 1
$TC_DP4	$TC_DP13	長さ 2
$TC_DP5	$TC_DP14	長さ 3
ジオメトリ	摩耗	半径オフセット
$TC_DP6	$TC_DP15	半径
$TC_DP7	$TC_DP16	コーナ半径（ツールタイプ 700：スロッティングソー）
ジオメトリ	摩耗	さらなるオフセット
$TC_DP8	$TC_DP17	長さ 4 (ツールタイプ 700：スロッティングソー）
$TC_DP9	$TC_DP18	長さ 5
$TC_DP10	$TC_DP19	角度 1（ツール面とトーラス面の間の角度）
$TC_DP11	$TC_DP20	角度 2（縦軸とトーラス面の上端の間の角度）

ベース寸法／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

テクノロジ

$TC_DP8	クリアランス角度 1.) クリアランス角度は，手動ターン用にここに保存される：ツールタイプ 5xx。旋削バイトの標準サイクルの場合と同じ意味。 2.) ドリルのチップ角度は，ショップミル用にここに保存される：ツールタイプ 2xx。 3.) 切削バイトの標準サイクルに使用：ツールタイプ 5xx。このツールの第二切削エッジ角度である。
$TC_DP10	1.) 切削レートの値は，手動ターン用にここに保存される。 2.) 様々な状態のツールタイプ 1xx と 2xx のビットコード化値は，ショップミル用にここに保存される。

合計オフセットおよびセットアップオフセットのパラメータ ($TC_SCPxy-, $TC_ECPxy)

パラメータの番号付けは，パラメータ $TC_DP3 ～ $TC_DP11 の番号付けに合わせられます。

パラメータの効果は摩耗に似ています（ツールジオメトリへ追加）。6 つまでの合計／セットアップパラメータが，切削エッジパラメタ別に定義できます。

オフセットが追加されるツールジオメトリパラメータ	合計／セットアップパラメータ（長さオフセット）	ツール摩耗パラメータ
$TC_DP3	長さ 1 $TC_SCP13, $TC_SCP23,$TC_SCP33, $TC_SCP43,$TC_SCP53,$TC_SCP63 $TC_ECP13, $TC_ECP23,$TC_ECP33, $TC_ECP43,$TC_ECP53,$TC_ECP63 太字の数字 1, 2, ... 6 は，最大 6 つのオフセット（位置依存または類似）のパラメータを識別する。カラム 1 に指定されるパラメータに対して DL =1, ... 6 を使ってプログラムされるオフセット。	$TC_DP12
$TC_DP4	長さ 2 $TC_SCP14, $TC_SCP24,$TC_SCP34, $TC_SCP44,$TC_SCP54,$TC_SCP64 $TC_ECP14, $TC_ECP24,$TC_ECP34, $TC_ECP44,$TC_ECP54,$TC_ECP64	$TC_DP13
$TC_DP5	長さ 3 その他	$TC_DP14
	半径オフセット	
$TC_DP6	半径	$TC_DP15
$TC_DP7	コーナ半径	$TC_DP16
	さらなるオフセット	
$TC_DP8	長さ 4	$TC_DP17
$TC_DP9	長さ 5	$TC_DP18
$TC_DP10	角度 1 ... など	$TC_DP19
$TC_DP11	角度 2 $TC_SCP21, $TC_SCP31,$TC_SCP41, $TC_SCP51,$TC_SCP61,$TC_SCP71 $TC_ECP21, $TC_ECP31,$TC_ECP41, $TC_ECP51,$TC_ECP61,$TC_ECP71 太字の数字 2, 3, ... 7 は，最大 6 つのオフセット（位置依存または類似）のパラメータを識別する。カラム 1 に指定されるパラメータに対して DL =1, ... 6 を使ってプログラムされるオフセット。	$TC_DP20

補足条件

切削エッジの DL データセットの最大数および NCK の合計オフセットの合計数は，マシンデータによって定義されます。デフォルト値はゼロ：つまりどの合計オフセットもプラグラムできません。

「監視機能」を起動すると，摩耗または「合計オフセット」の有無に関してツールを監視できます。

追加合計／セットアップデータセットは，追加バッファメモリを使用します。8 バ

イトがパラメータ別に必要です。合計オフセットデータセットには，8 バイト * 9 パラメータ = 72 バイトが必要です。セットアップデータセットには同等量のメモリが必要です。一定数のバイトもまた，内部的管理データに必要です。

■ 起動

機能はマシンデータ内で起動されなければなりません。

システムパラメータ $TC_ECPx と $TC_SCPx および OPI インタフェースを介して定義されたセットアップと合計オフセット（微）は，パートプログラム内で起動できます。

これは言語コマンド DL=「番号」をプログラミングすることで実行されます。新しい D 番号が起動されると，新しい DL がプログラムされるか，またはマシンデータ MD 20272: SUMCORR_DEFAULT に定義された DL 番号が起動されます。

DL プログラミング

合計オフセットは，常に次のコマンドでアクティブ D に関してプログラムされます。

DL = ’n’.

合計オフセット ’n’ はアクティブ D 番号の摩耗に追加されます。

（注）セットアップオフセットおよび合計オフセット（微）を使用する場合，両オフセットは結合され，ツール摩耗へ追加されます。

合計オフセットは，次のコマンドで選択解除されます。

DL = 0

（注）DL0 は認められません。オフセットが選択解除（D0 および T0）されると，合計オフセットも起動解除されます。
存在しない合計オフセットをプログラムすると，存在しない D オフセットの場合同様，アラームがでます。
定義された摩耗は，オフセットの唯一のコンポーネント（システムパラメータ $TC_DP12 から $TC_DP20）です。

重要

D オフセットがアクティブ（選択解除にも適用）である時，合計オフセットがプログラムされる場合，D コマンドをプログラムする時と同じ影響をパスに与えます。したがって，アクティブ半径補正は隣接ブロックの参照を失います。

コンフィギュレーション

MM_KIND_OF_SUMCORR，ビット 4=0：標準設定

合計オフセットの内 1 セットのみが DL 番号別に存在します。

ここで一般的に合計オフセットを参照します。

これは $TC_SCPx で表されるデータを参照します。

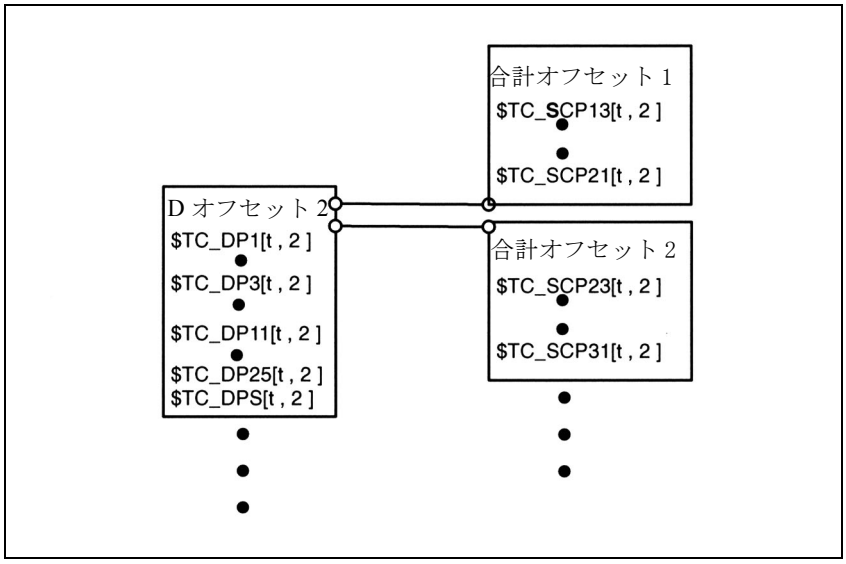


図 1.213 MD 18112 MM_KIND_OF_SUMCORR，ビット 3 = 0

ツール T = t はアクティブです。図中のデータを使って，次のようにプログラムされます。

D2　　：切削エッジオフセット：$TC_DP3,... $TC_DP11 + 摩耗 ($TC_DP12,...$TC_DP20) + アダプタ寸法

...

DL=1　：合計オフセット 1 が前のオフセット D2 へさらに追加されます。$TC_SCP13,...$TC_SCP21

...

DL=2　：合計オフセット 2 が，合計オフセット 1 の代わりにオフセット D2 へ追加されます。$TC_SCP23,...$TC_SCP31

...

DL=0　：選択解除合計オフセット：D2 のデータのみが現在有効です。

MM_KIND_OF_SUMCORR，ビット 4=1：

セットアップオフセットが使用可能です。

合計オフセットは，現在，合計オフセット（微）（$TC_SCPx で表される）およびセットアップオフセット（$TC_ECPx で表される）で構成されます。したがって，2 つのデータセットが 1 つの DL 番号に存在します。合計オフセットは，対応するコンポーネント ($TC_ECPx + $TC_SCPx) を追加して計算されます。

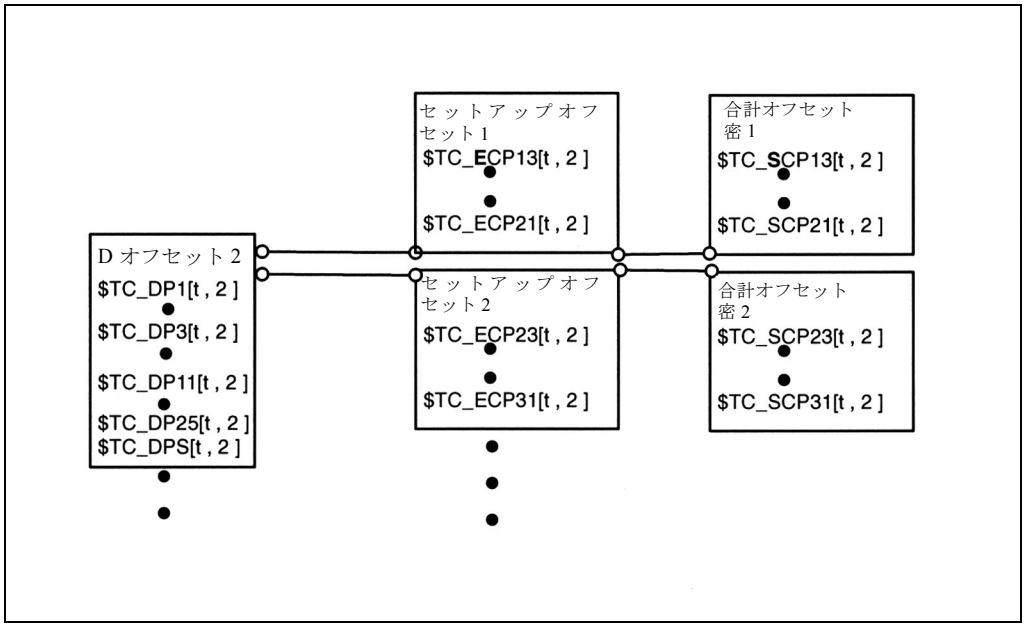


図 1.214 MD 18112 MM_KIND_OF_SUMCORR，ビット 4 = 1 セットアップオフセット + 合計オフセット（微）

ツール T = t はアクティブです。図中のデータを使って，次のようにプログラムされます。

D2　　：切削エッジオフセット：$TC_DP3,... $TC_DP11 + 摩耗 ($TC_DP12,...$TC_DP20) + アダプタ寸法

...

DL=1　　：合計オフセット 1 が前のオフセット D2 へさらに追加されます。$TC_ECP13 + $TC_SCP13,...$TC_ECP21 + $TC_SCP21

...

DL=2　　：合計オフセット 2 が，合計オフセット 1 の代わりにオフセット D2 へ追加されます。$TC_ECP23 + $TC_SCP23,...$TC_ECP31 + $TC_SCP31

...

DL=0　　：選択解除合計オフセット：
D2 のデータのみが現在有効です。

パートプログラム内の読取り／書込み

合計オフセットパラメータの個々のセットは，システムパラメータ $TC_SCP の番号範囲にしたがって区別されます。

各パラメータの意味は，ジオメトリパラメータ $TC_DP3 ～ $TC_DP11 に似ていま

す。長さ 1，長さ 2 および長さ 3 のみが，基本的機能（切削エッジの最初の合計オフセット用パラメータ $TC_SCP13 ～ $TC_SCP15）に対してイネーブルされます。

R5 = $TC_SCP13[t, d] ：R パラメータの値を，ツール (t) 用切削エッジ (d) の合計オフセット 1 の最初のコンポーネント値へ設定します。

R6 = $TC_SCP21[t, d] ：R パラメータの値を，ツール (t) 用切削エッジ (d) の合計オフセット 1 の最後のコンポーネント値へ設定します。

R50 = $TC_SCP23[t, d] ：R パラメータの値を，ツール (t) 用切削エッジ (d) の合計オフセット 2 の最初のコンポーネント値へ設定します。

$TC_SCP43[t, d] = 1.234：ツール (t) 用切削エッジ (d) の合計オフセット 4 の最初のコンポーネント値を値 1.234 へ設定します。

上記のステートメントは，セットアップオフセットへも適用されます（NCK がこのオプションで構成される場合），つまり，

R5 = $TC_ECP13[t, d] ：R パラメータの値を，ツール (t) 用切削エッジ (d) のセットアップオフセット 1 の最初のコンポーネント値へ設定します。

R6 = $TC_ECP21[t, d] ：R パラメータの値を，ツール (t) 用切削エッジ (d) のセットアップオフセット 1 の最後のコンポーネント値へ設定します。

以下同様。

セットアップオフセットを使って加工する時，合計オフセット（微）は，$TC_SCPx パラメータを使って書込まれます。

新しい合計オフセットを生成する

オフセットデータセット (x) がまだ存在しない場合，それはそのパラメータ (y) の 1 つへの最初の書込みで生成されます。

$TC_SCPxy[t, d] = r.r：合計オフセット x のパラメータ y は，値 ’r.r.’ を割当てられます。x の他のパラメータはゼロです。

セットアップオフセットを使って加工する時，合計オフセット（微）は，$TC_SCPx パラメータを使って書込まれます。

（注）セットアップオフセットを使って加工する時，データセットが まだ [t, d] 用に存在しなかった場合，セットアップオフセット用のデータセットは，データセットが合計オフセット（微）用に生成される時，生成されます。

新しいセットアップオフセットを生成する

オフセットデータセット (x) がまだ存在しない場合，それはそのパラメータ (y) の 1 つへの最初の書込みオペレーションで生成されます。

$TC_ECPxy[t, d] = r.r：セットアップオフセットのパラメータ y は，値 'r.r.' を割当てられます。x の他のパラメータはゼロです。

（注）セットアップオフセットを使って加工する時，データセットがまだ [t, d] に対して存在しなかった場合，セットアップオフセット用データセットは，1 つのデータセットが合計オフセット（微）に対して生成される時，生成されます。

DELDL - 合計オフセットの削除

合計オフセットは，一般的に，所定時所定位置で切削エッジを使って加工する時のみ関係します。NC 言語コマンド DELDL を使って，（メモリを解放するために）切削エッジから合計オフセットを削除できます。

status = DELDL(t, d)：ツール t の切削エッジ d の合計オフセットを全部削除します。t，d はオプションパラメータです。

d が指定されない場合，ツール t の全切削エッジの合計オフセットが全て削除されます。d と t が指定されない場合，TO ユニットの全ツールの切削エッジの合計オフセットが全て削除されます（コマンドがプログラムされるチャンネルに対して）。

セットアップオフセットを使って加工する時，DELDL コマンドは，セットアップオフセットと指定された切削エッジの合計オフセット（微）の両方を削除します。

（注）データセットに使用されるメモリは，削除後開放されます。
削除された合計オフセットは，その後起動またはプログラムできません。

アクティブツールの合計オフセットおよびセットアップオフセットは，削除できません（D オフセットまたはツールデータの削除に似ています）。

「ステータス」リターン値は，削除コマンドの結果を示します。

0　削除がうまく行われた

-1　（1 つの切削エッジ）が削除されなかったか，または（いくつかの切削エッジ）が完全に削除されなかった。

データバックアップ

データは，一般的ツールデータバックアップ中保存されます（D 番号データセットアップのコンポーネントとして）。

現在のステータスが復元されるように合計オフセットのバックアップをお奨めします。合計オフセットは，マシンデータによってデータバックアップから除外することができます（セットアップオフセットおよび合計オフセット（微）用に別に設定できます）。

■ 例

例 1

マシンデータを使って，オフセットおよび合計オフセットのどれも工具交換時アクティブでないように定義されます ($MC_CUTTING_EDGE_DEFAULT=0, $MC_SUMCORR_DEFAULT=0)。

T5 M06 ：ツール番号 5 がロードされる - オフセットはどれも非アクティブ

D1 DL=3 ：オフセット D1 + D1 の合計オフセット 3 が起動される

X10

DL=2 ：オフセット D1 + 合計オフセット 2 が起動される

X20

DL=0 ：合計オフセット選択解除，オフセット D1 のみが現在アクティブ

D2 ：オフセット D2 が起動される - 合計オフセットはオフセットに含まれない

X1

DL1 ：オフセット D2 + 合計オフセット 1 が起動される

X2

D0 ：オフセット選択解除

X3

DL2 ：影響なし - D0 の DL2 はゼロ（T0 D2 のプログラミングに同じ）

例 2

マシンデータを使って，オフセット D2 および合計オフセット DL=1 がツールチェンジ時に起動されるように定義されます ($MC_CUTTING_EDGE_DEFAULT=2, $MC_SUMCORR_DEFAULT=1)。

T5 M06 ：ツール番号 5 がロードされる - D2 + DL=1 がアクティブ（ =MDen の値）

D1 DL=3 ：オフセット D1 + D1 の合計オフセット 3 が起動される

X10

DL=2 ：オフセット D1 + 合計オフセット 2 が起動される

X20

DL=0 ：合計オフセット選択解除，オフセット D1 のみが現在アクティブ

D2 ：オフセット D2 が起動される - 合計オフセット DL=1 が起動される

X1

DL=2 ：オフセット D2 + 合計オフセット 2 が起動される

D1 ：オフセット D1 + 合計オフセット 1 が起動される

1.19.3 補足条件

■ フラット D 番号構造

研削ツール

研削ツール（ツールタイプ 400 ～ 499）は，簡単なツール管理構造（フラット D 番号）を使って定義できません。

ブロック検索

PLC へ T 番号を出力すると，NCK 内で同期プロセスを起動します。絶対間接 D プログラミングで，PLC は VDI を介して D 値を戻します。NCK は，T 番号の出力後，PLC から「D 番号書込み終了」の応答があるまで待ちます。計算なしのブロック検索で，この同期プロセスは，最初の有効 T 番号が再出力されるまで起動解除されなければなりません。

これは，NCK が D プログラミングを待つべきでないことを意味します。

（注）MD $MC_AUXFU_AT_BLOCK_SEARCH_END を使って，ブロック検索の終わりに，PLC へ補助機能を出力するタイミングを制御できます - 終了時または NC スタート時自動的に。

Reorg

ここで定義される（唯一の）書込み可能変数 $A_MONIFACT は，メインランデータによって保存されます。書込みプロセスはメインランと同期を取って行われるので，特別な方法は Reorg に必要ありません。

1.19.4 データの説明 (MD，SD)

■ 一般マシンデータ

18088 MD 番号	MM_NUM_TOOL_CARRIER	
初期設定：0	最小入力リミット：0	最大入力リミット：99999999
変更が有効になるための条件： 電源投入	保護レベル：2 / 7	単位：-
データタイプ：DWORD	適用開始 SW バージョン：4.1	
意味：	TO ユニット範囲内で回転可能ツールに対して定義できる最大数。値はアクティブ TO ユニット数で割られる。整数は，どれだけ多くのツールホルダが TO ユニット別に定義されることができるかを示す。 ツールホルダの定義に関するデータは，システム変数 $TC_CARR1, ...$TC_CARR17 を使って設定される。	
アプリケーション例	2 チャンネルがアクティブで，各チャンネル（= プリセット）に 1 つの TO ユニットがある。 3 つのキャリアは，チャンネル 2 の 1 つ，チャンネル 1 に定義されることになる。設定される値は 6/2=3 であるので，6 である。つまり各 TO ユニットに最大 3 キャリアの定義。	

18102 MD 番号	MM_TYPE_OF_CUTTING_EDGE フラット D 番号管理の起動	
初期設定：0	最小入力リミット：0	最大入力リミット：2
変更が有効になるための条件： 電源投入	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン : 4	
意味：	この MD は「フラット D 番号管理」機能を起動する。各値は，D プログラミングのタイプを決定する：直接または間接プログラミング。 MD= 0　標準値はゼロである。これは，NCK が T および D 番号を管理することを意味する。 1　値 1 は D 番号が直接および絶対値としてプログラムされることを意味する。 2　値 2 は D 番号が間接および相対値としてプログラムされることを意味する：つまりプログラムされた D 番号は，テーブル VDI の索引である。PLC は，絶対 D 番号をこのテーブルに書込む。 NCK はその番号を読取り，対応するオフセットを選択する。起動（0 から >0 へ値を変更）または起動解除（>0 から 0 へ値を変更）すると，バッファメモリが再構成される。つまりデータは削除される。 ビット 0 が MD 18080: MM_TOOL_MANAGEMENT_MASK でイネーブルされない場合，値 >0 のみが NCK によって受入れられる。つまり，ツール管理機能は，同時にアクティブであってはならない。	
関連性		

18105 MD 番号	MM_MAX_CUTTING_EDGE_NO D 番号の最大値	
初期設定：9	最小入力リミット：1	最大入力リミット：32000
変更が有効になるための条件： 電源投入	保護レベル：7/2	単位：番号
データタイプ :DWORD	適用開始 SW バージョン：5.1	
意味：	最大 D 番号値の定義 ツール別 D 番号の最大数は影響されない。 （MD 18106: MM_MAX_CUTTING_EDGE_PERTOOL に定義されている）。 D 番号割当て監視は，新しい D 番号の定義によってのみ有効である。既存のデータセットは，このマシンデータが変更される場合，自動的に検証されない。 マシンデータは，「フラット D 番号」機能を使って評価されない。 最大 D 番号値とツール別最大切削エッジ番号の間の関係が変更されると，バッファメモリの必要性も変化する（両初期値は同じ）。	
関連性	MD 18106: MM_MAX_CUTTING_EDGE_PERTOOL	

18106 MD 番号	MM_MAX_CUTTING_EDGE_PERTOOL ツール別切削エッジの最大数	
初期設定：9	最小入力リミット : 1	最大入力リミット : 12
変更が有効になるための条件： 電源オン	保護レベル：7/2	単位：番号
データタイプ：DWORD	適用開始 SW バージョン：5.1	
意味：	ツール（T 番号）別切削エッジ（D オフセット）の最大数の定義。 このマシンデータは，ツールに（例えば，1 つのみの切削エッジを持つツールに対して）実際に使用可能な切削エッジの最大数を割振りすることによって，データ定義中，高いレベルのセキュリティを提供する。 12 までのオフセットがツール別に割振りされることができる。 マシンデータは，「フラット D 番号」機能を使って評価されない。 最大 D 番号値とツール別最大切削エッジ数の間の関係が変更されると，バッファメモリの必要性も変わる（両初期値は同じ）。	
関連性	MD 18105: MM_MAX_CUTTING_EDGE_NO	

18108 MD 番号	MM_NUM_SUMCORR NCK の全ての合計オフセット数	
初期設定：-1	最小入力リミット：-1	最大入力リミット：*)
変更が有効になるための条件： 電源オン	保護レベル：7/2	単位：番号
データタイプ：DWORD	適用開始 SW バージョン：5.1	
意味：	値　意味 -1　合計オフセット数は，切削エッジ数と切削エッジ別合計オフセット数の積に等しい。 ＞0　システムが，実際の合計オフセット数用のバッファメモリのみを確保するために，上記の積より少ない値を選択できる。「セットアップ補正」がアクティブである場合，合計オフセットに必要なメモリは倍になる。 *)　最大値は，「切削エッジ数」の上限とマシンデータの「切削エッジ別合計オフセット数」の積から計算される。	
関連性	MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA MD 18110: MM_MAX_SUMCORR_PERCUTTING_EDGE MD 18112: MM_KIND_OF_SUMCORR	

18110 MD 番号	MM_MAX_SUMCORR_PER_CUTTEDGE 切削エッジ別合計オフセットの最大数	
初期設定：1	最小入力リミット：1	最大入力リミット：6
変更が有効になるための条件： 電源オン	保護レベル：7/2	単位：番号
データタイプ：	DWORD 適用開始 SW バージョン：5.1	
意味：	切削エッジ別合計オフセットの最大数を定義する このマシンデータの影響は，MD 18108 の値に依存する。 MM_NUM_SUMCORR > 1 このデータはメモリを確保せず，監視目的専用に使用される。 MM_NUM_SUMCORR = -1 マシンデータはメモリを予約する。	
関連性	MD 18108: MM_NUM_SUMCORR MD 18100: MM_NUM_CUTTING_EDGES_IN_TOA	

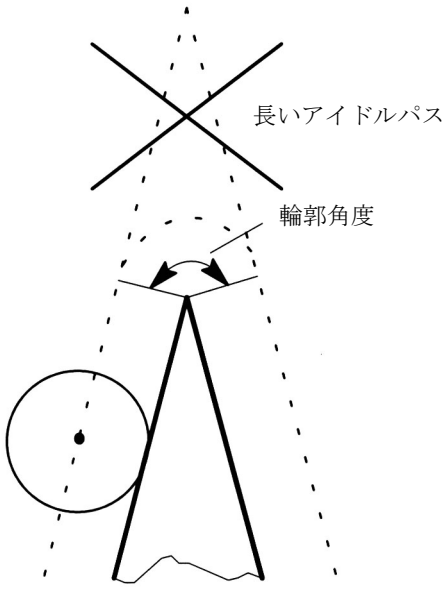
18112 MD 番号	MM_KIND_OF_SUMCORR NCK の合計オフセットのプロパティ	
初期設定：0	最小入力リミット：0	最大入力リミット：0x1F
変更が有効になるための条件： 電源オン	保護レベル：7/2	単位：ビットマスク
データタイプ：INT	適用開始 SW バージョン：5.1	
意味：	NCK の合計オフセットのプロパティを定義する： ビット番号 値 意味 0 0 合計オフセットは, データバックアップ中ツールデータを使ってセーブされる。 1 0x1 合計オフセットは, データバックアップ中ツールデータを使ってセーブされない。 1 0 セットアップオフセットは, データバックアップ中ツールデータを使ってセーブされる。 1 0x2 セットアップオフセットは, データバックアップ中ツールデータを使ってセーブされない。 2 0 「ツール管理」機能を使用する場合：ツールを 起動する時，既存の合計／セットアップオフセットは変更されない。 1 0x4 ツールを起動する時, 既存の合計／セットアップオフセットはゼロに設定される。 セットアップオフセットは変更されない。 3 0 「ツール管理」プラス「アダプタ」機能を使用する場合：合計オフセットの変換が実行される。 1 0x8 合計オフセットの変換はない。 4 0 セットアップオフセットデータセットはない。 1 0x10 セットアップオフセットデータセットが生成される。合計オフセットは，その時セットアップオフセットおよび合計オフセット（微）の合計から計算される。 ビット 0，1，2，3 の状態への変更は，メモリ構成に影響しない。ビット 4 への変更は，次の電源投入時，バッファメモリを再構成する。	
関連性	MD 18080: MM_TOOL_MANAGEMENT_MASK MD 18086: MM_NUM_MAGAZINE_LOCATION MD 18104: MM_NUM_TOOL_ADAPTER MD 20310: TOOL_MANAGEMENT_MASK	

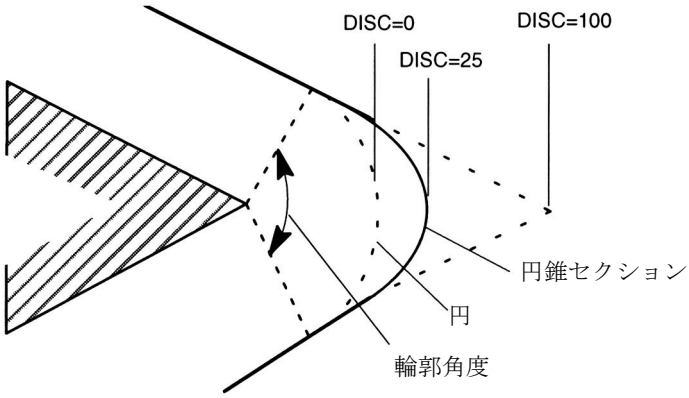
■ チャンネル別マシンデータ

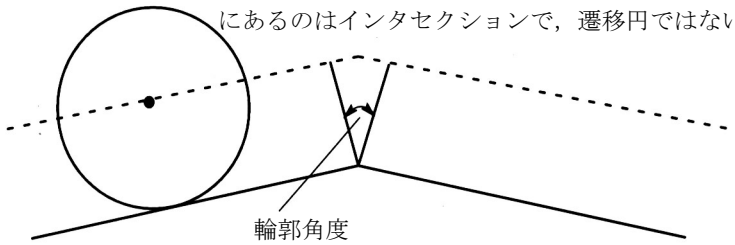
20126 MD 番号	TOOL_CARRIER_RESET_VALUE リセット時のアクティブツールホルダ	
初期設定：0	最小入力リミット：0	最大入力リミット：プラス
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：-
データタイプ：DWORD	適用開始 SW バージョン：4.1	
意味：	MD: $MC_RESET_MODE_MASK に依存して，および MD: $MC_START_MODE_MASK にしたがってパートプログラムをスタートするために，パワーアップおよびリセット中またはパートプログラム終了時，工具長補正を選択するのに使用されるツールホルダを指定する。このデータはツール管理なしに有効である。	
参照	FBW	

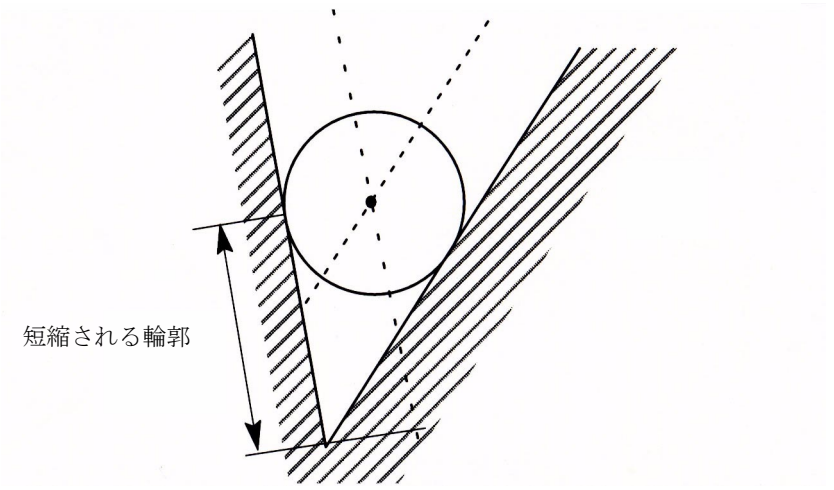
20202 MD 番号	WAB_MAXNUM_DUMMY_BLOCKS SAR を使った移動動作のないブロックの最大数	
初期設定：5, 5, 5, 5, 5, ...,,	最小入力リミット：1	最大入力リミット：10
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：-
データタイプ：BYTE	適用開始 SW バージョン：4.3	
意味：	SAR ブロックと，アプローチまたは後退接線の向きを決定する移動ブロックの間に表示できるブロックの最大数。	

20204 MD 番号	WAB_CLEARANCE_TOLERANCE SAR 用の DISCL 定義の点に対して定義される公差	
初期設定：0.001	最小入力リミット：0	最大入力リミット：
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：mm
データタイプ：REAL	適用開始 SW バージョン：4.4	
意味：	アプローチおよび後退中，初期面からインフィード移動が低速 (G341) で実行される DISCL 定義の点，または実際のアプローチ動作がそこでスタート (G340) し，初期面とアプローチ面の間になければならない点。 点がこの間隔外にあり，ずれがこのマシンデータに等しいかまたは以下の場合，点は，アプローチまたは後退面内にあると仮定される。ずれがそれより大きい場合，アラーム 10741 (PRAL_WAB_DIRECTION_CHANGE) が発せられる。	

20210 MD 番号	CUTCOM_CORNER_LIMIT **工具径補正を使った補正ブロックの最大角度**	
初期設定：100.0	最小入力リミット：0.0	最大入力リミット：150.0
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：度
データタイプ：DOUBLE	適用開始 SW バージョン：1.1	
意味：	外側コーナが非常に鋭角である場合，G451 は無効なパスになる（図 1.197 参照）。したがって，外側コーナが非常に鋭角である時，システムは，G451（インタセクション）から G450（適切であれば DISC を使って遷移円）へ自動的に切替わる。この自動切替え（インタセクション ---> 遷移円）にしたがって移動が可能な輪郭の角度は，CUTCOM_CORNER_LIMIT に定義される。	
		

20220 MD 番号	CUTCOM_MAX_DISC DISC の最大角度	
初期設定：50.0	最小入力リミット：0.0	最大入力リミット：75.0
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：-
データタイプ：DOUBLE	適用開始 SW バージョン：1.1	
意味：	G450 遷移円は，鋭い外側輪郭コーナを生成できない。なぜなら遷移円を通るツール中心点のパスが，切削エッジが外側コーナ（プログラムされた位置）で停止するように制御されるからである。鋭角の外側コーナが G450 で加工される場合，オーバシュートをプログラムするために，DISC 指令を使ってプログラムできる。これは，遷移円を円錐セクションへ変換し，切削エッジは外側コーナから離昇する。 DISC 指令の値範囲は，0 から理論上 100 まで 1 ステップずつ拡張する。 DISC = 0 ... オーバシュートがディスエーブルで，遷移円がアクティブ DISC = 100 ... オーバシュートが十分大きく，インタセクション (G451) に似た応答を理論的に生成する。 CUTCOM_MAX_DISC に記憶された値より大きくプログラムされた DISC の値は，メッセージの出力なしにこの最大値に制限される。したがって，パス速度の極度の非直線変更は避けられることができる。	
		
特殊ケース：	一般的に，DISC に 50 以上の値を入力しても意味がない。したがって，値 > 75 の入力はできない。	

20230 MD 番号	CUTCOM_CURVE_INSERT_LIMIT **工具径補正を使ったインタセクション計算のための最大角度**	
初期設定：10.0	最小入力リミット：0.0	最大入力リミット：150.0
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：-
データタイプ：DOUBLE	適用開始 SW バージョン：1.1	
意味：	外側コーナが非常に平坦な場合，G450（遷移円）と G451（インタセクション）を使った応答は類似してくる。この場合，遷移円を挿入することは望ましくない。5 軸加工でこれらの外側コーナに遷移円の挿入が認められない 1 つの理由は，輪郭モード (G64) での速度に制限を課すからである。したがって，外側コーナが非常に平坦な場合，システムは，G450（適切な場合 DISC を使って遷移円）から G451（選択されるインタセクション）へ自動的に切替わる。この自動切換（遷移円 ---> インタセクション）にしたがって移動されることができる輪郭角度は，CUT-COM_ CURVE_INSERT_LIMIT に定義される。	
	輪郭角度 CUTCOM_CURVE_INSERT_LIMIT の下にあるのはインタセクションで，遷移円ではない  	

20240 MD 番号	CUTCOM_MAXNUM_CHECK_BLOCKS **工具径補正を使った予想輪郭計算用のブロック**	
初期設定：3	最小入力リミット：2	最大入力リミット：10
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：-
データタイプ：BYTE	適用開始 SW バージョン：1.1	
意味：	内側コーナと「ボトルネック」のいくつかのケースでは，等距離の複数インタセクションが数個の連続ブロックに見つけられる。これらの場合，常に最後のインタセクションが，有効インタセクションとして使用される（図参照）。予想チェックに使用されるブロックの最大数は，CUTCOM_MAXNUM_CHECK_BLOCKS に入力できる。CDON を使って衝突監視が起動されている場合，2 つまたはそれ以上の連続ブロックのチェックのみが可能（1.19.2「■磨耗 ― 工具径補正（ツールパラメータ 15～20)」参照)。  	

20250 MD 番号	CUTCOM_MAXNUM_DUMMY_BLOCKS TRC を使った移動動作のないダミーブロックの最大数	
初期設定：3	最小入力リミット：0	最大入力リミット：5
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：-
データタイプ：BYTE	適用開始 SW バージョン：1.1	
意味：	一般的に，工具径補正 (TRC) がアクティブである時，現在面のジオメトリ軸に位置するプログラムブロックのみがプログラムされる。しかし現在面のパス情報を含まないダミーブロックも，アクティブ TRC を使ってプログラムされることができる。例えば ・インフィード軸上の位置 ・補助機能 ・一般事項：メインランへ至り，そこで実行されるブロック そのようなダミーブロックの最大数は，この MD によって指定される。最大数を超える場合，アラーム 10762 「アクティブ工具径補正を使った 2 つの移動ブロックの間に空白ブロックが多すぎる」が起動される。 （注） これらの MD に関する限り，コメントブロック，演算ブロックおよび空白ブロックはダミーブロックではなく，したがってどの数量でもプログラムされることができる（アラームが起動されることなく）。	

20252 MD 番号	CUTCOM_MAXNUM_SUPPR_BLOCKS オフセット抑制を使ったブロックの最大数	
初期設定：5	最小入力リミット：0	最大入力リミット：10
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：-
データタイプ：BYTE	適用開始 SW バージョン：4	
意味：	「半径オフセットを一定に保つ」インタフェース信号（CUTCONON またはアクティブ TRC 中 G41/G42 の再プログラミング）が有効であるアクティブ工具径補正ブロックの最大数を示す。 （注）この状況で再位置決めを行うためにも，アクティブ CUTONON でブロック数を制限する必要である。マシンデータのこの値を増やすと，NC ブロック用にメモリの増大が必要になる。	
関連性		

20256 MD 番号	CUTCOM_INTERS_POLY_ENABLE 多項式に可能なインタセクションプロセス	
初期設定：1	最小入力リミット：-	最大入力リミット：-
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：BOOLEAN	適用開始 SW バージョン：4	
意味：	このマシンデータが TRUE の場合，工具径補正のアクティブ時，多項式（スプライン）を含む外側コーナでの遷移は，インタセクションプロシージャを使って処理されることができる。マシンデータが FALSE の場合，常に円錐（円）カットが挿入される。 FALSE の場合，応答は 4.0 以前のソフトウェアリリースでの応答と同じである。	
関連性		

20270 MD 番号	CUTTING_EDGE_DEFAULT プログラミングなしツール切削エッジの基本設定	
初期設定：1	最小入力リミット：-2	最大入力リミット：MD_SLMAX-CUTTINGEDGENUMBER
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：2	
意味：	MD：CUTTING_EDGE_DEFAULT = 0 ツールのチェンジは自動的にオフセットを選択解除する。 MD：CUTTING_EDGE_DEFAULT > 0 切削数の設定が，ツールチェンジの後常に自動的にアクティブになる。 MD：CUTTING_EDGE_DEFAULT = -1： 旧ツールの切削エッジ数が保持され，ツールチェンジの後新ツールに選択される。 MD：CUTTING_EDGE_DEFAULT = -2： 前ツールの二番目の切削エッジが使用されていた場合，新ツールへチェンジ後，新ツールの二番目の切削エッジが加工に使用される。選択された最終オフセットのオフセット値は，D 番号がプログラムされるまでアクティブのままである。ツール管理：ツールチェンジ（例えば M06 ）と補正選択の間に，軸がなお移動しなければならない場合，この設定が選択されるべきである。 例： MD: CUTTING_EDGE_DEFAULT = 1; 切削エッジがプログラムされない場合，ツールチェンジ後最初の切削エッジがアクティブである。	

20360 MD 番号	TOOL_PARAMETER_DEF_MASK ツールパラメータの影響の定義	
初期設定： 0x0, 0x0, 0x0, 0x0, 0x0, ...	最小入力リミット：0	最大入力リミット：0xFFFF
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：-
データタイプ：DWORD	適用開始 SW バージョン：4.1	
意味：	ビット 0(LSB)：旋削および研削ツールの場合，フェーシング軸の摩耗パラメータは直径値として計算される。	

22530 MD 番号	TOCARR_CHANGE_M_CODE ツールホルダチェンジ用の M コード	
初期設定：0	最小入力リミット：-99999999	最大入力リミット：99999999
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：-
データタイプ：DWORD	適用開始 SW バージョン：4.1	
意味：	このマシンデータの値は，ツールホルダが VDI インタフェースで起動される時，M コード出力の番号を決定する。 - MD が正である場合，常に一定の M コードが出力される。 - MD が負である場合，ツールホルダの番号がマシンデータ値に追加され，その合計が出力される。	
アプリケーション例	MD 値 = -200 TOCARR = 7 がプログラムされると，M207 が出力される。	
例外，エラー	出力されるコード番号またはこの MD 自体の値が，0 ～ 6，17 または 30 の値の 1 つである場合，M コードは出力されない。生成されたこのような M コードと他の機能の間の衝突監視はない。	

22530 MD 番号	TOCARR_CHANGE_M_CODE ツールホルダチェンジ用の M コード
参照	1.9 補助機能出力（H2）

22550 MD 番号	TOOL_CHANGE_MODE M 機能を使った新たなツールオフセット	
初期設定：0	最小入力リミット：0	最大入力リミット：1
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：BYTE	適用開始 SW バージョン：1.1	
意味：	T 機能はプログラム内でツールを選択するのに使用される。このマシンデータの設定は，T 機能の実行時，即時新しいツールがロードされるかどうかを決定する。 MD: TOOL_CHANGE_MODE = 0 新しいツールが，T 機能の実行時即時ロードされる。この設定は，主にツールタレット付き旋削マシンに使用される。 MD: TOOL_CHANGE_MODE = 1 新しいツールが，T 機能の実行時ローディングに用意される。この設定は，加工プロセスを中断することなく新しいツールをツールチェンジ位置へ配置するために，主にフライス加工マシン上で使用される。MD 22560:TOOL_CHANGE_M_CODE は，古いツールをスピンドルから除去し，新しいツールをスピンドルにロードするために使用される。このツールチェンジは，DIN 66025 にしたがって M 機能 M06 を使ってプログラムされる必要がある。	
関連性	MD 22560: TOOL_CHANGE_M_CODE	

22560 MD 番号	TOOL_CHANGE_M_CODE ツールチェンジ用の M 機能	
初期設定：6	最小入力リミット：0	最大入力リミット：9999 9999
変更が有効になるための条件： 電源オン	保護レベル：2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：1.1	
意味：	T 機能のみが，ツールチェンジに新しいツールを用意するために使用される場合（この設定は，加工プロセスを中断することなくツールチェンジ位置に新しいツールを配置するために，ツールマガジン付きのフライス加工マシンに主に使用される），別の M 機能を使ってツールチェンジを起動しなければならない。TOOL_CHANGE_M_CODE に入力される M 機能は，ツールチェンジを起動する（スピンドルから古いツールを除去し，スピンドルに新しいツールを挿入する）。このツールチェンジは，DIN 66025 にしたがって M 機能 M06 を使ってプログラムされる必要がある。	
関連性	MD 22550：TOOL_CHANGE_MODE	

22562 MD 番号	TOOL_CHANGE_ERROR_MODE **プログラムされたツールチェンジ時のエラー応答**	
初期設定：0	最小入力リミット：0	最大入力リミット：3
変更が有効になるための条件： 電源オン	保護レベル：7/2	単位：-
データタイプ：DWORD	適用開始 SW バージョン：5.1	
意味：	マシンデータの初期値 = 0 は，設定 TOOL_CHANGE_MODE = 0 を使って加工する時，変更しなければならない（ツールチェンジは，T コマンドでのみプログラムされる）。 この場合マシンデータの値は影響しない。 ビット番号　値　　意味 0　0　デフォルト応答 プログラムは，エラーが生じた NC ブロックで停止する。 1 0x1　ツールチェンジを準備したブロックでエラーが生じる場合，準備コマンド (T) を参照するアラームは，関連ツールチェンジコマンド (M06) がプログラム実行時に解釈されるまで無視される。その時初めてシステムは，準備コマンドによって起動されたアラームを出力する。したがってこのブロックが達せられるまで，ユーザは修正できない。プログラムが再開されると，エラーのある NC ブロックは再度解釈され，準備コマンドが自動的に内部的に再度実行される。 1　0　ツール管理がアクティブである時のみ（この場合デフォルト応答）。ツールチェンジ準備中，NCK は，そのデータがマガジンへ割当てられたツールのみを検出する。 1 0x2　ツール管理がアクティブである時のみ NCK は，データが NCK 内で既知であるがマガジンへ割当てられていないツールもロードする。この場合，NCK は，プログラムされたスピンドル位置へツールデータを自動的に割当てようと試みる。前提条件：プログラムされたツールに対して，使用可能な 1 つのツールのみが存在する（マガジン位置へまだ割当てられていない）。	
関連性	MD 22550: TOOL_CHANGE_MODE	

20132 MD 番号	SUMCORR_RESET_VALUE **合計オフセットを選択するための番号**	
初期設定：0	最小入力リミット：0	最大入力リミット：6
変更が有効になるための条件： RESET	保護レベル：7/2	単位：番号
データタイプ：DWORD	適用開始 SW バージョン：5.1	
意味：	電源投入，リセットまたはパートプログラム終了時（RESET_MODE_MASK に依存して）およびパートプログラムスタート時（START_MODE_MASK に依存して），合計オフセットの選択に使用される合計オフセット番号を定義する。 マシンデータ MD 18108: MM_NUM_SUMCORR は，入力のための最大値を確定する。	
関連性	MD 20110: RESET_MODE_MASK MD 20112: START_MODE_MASK MD 20130: CUTTING_EDGE_RESET_VALUE MD 18108: MM_NUM_SUMCORR	

20272 MD 番号	SUMCORR_DEFAULT 新しい切削エッジオフセットを起動するための番号	
初期設定：0	最小入力リミット：0	最大入力リミット：6
変更が有効になるための条件： RESET	保護レベル：7/2	単位：番号
データタイプ：DWORD	適用開始 SW バージョン：5.1	
意味：	新しい合計オフセットが起動される時，起動される切削エッジの合計オフセットを定義する。	
関連性	MD 20270: CUTTING_EDGE_DEFAULT	

20096 MD 番号	T_M_ADDRESS_EXT_IS_SPINO スピンドル番号として解釈されるアドレス拡張	
初期設定：FALSE	最小入力リミット：-	最大入力リミット：-
変更が有効になるための条件： 電源オン	保護レベル：7/2	単位：ブール
データタイプ：BOOLEAN	適用開始 SW バージョン：5.1	
意味：	「ツール管理」機能が非アクティブである時のみ関連する： 値　　意味 FALSE　NC アドレス T と M のアドレス拡張の内容は，NCK によって評価されない。PLC はプログラムされる拡張の意味を確定する。 TRUE　NC アドレス T と M のアドレス拡張は，スピンドル番号として解釈される。 NCK は，ツール管理がアクティブである時と同じ方法で拡張を処理する。したがってプログラムされる D 番号は，プログラムされた主スピンドル番号を常に参照する（「ツールチェンジコマンド番号」= TOOL_CHANGE_M_CODE：初期値は 6）。	
関連性	MD 20090: SPIND_DEF_MASTER_SPIND MD 22550: TOOL_CHANGE_MODE MD 22560: TOOL_CHANGE_M_CODE	

■ 設定データ

42442 SD 番号	TOOL_OFFSET_INCR_PROG **相対プログラミングを使ったツールオフセットの後退**	
初期設定：1	最小入力リミット：0	最大入力リミット：1
変更が有効になるための条件： 電源オン	保護レベル：2 / 7	単位：-
データタイプ：BOOLEAN	適用開始 SW バージョン：4.3	
意味：	0： 相対プログラミングが軸に使用される時，プログラムされた位置デルタのみが移動される。工具長オフセットは，絶対位置が指定される時，単に後退させられる。 1： 相対プログラミングが軸に使用される時，工具長オフセットへの変更は，ツールチェンジ後後退させられる。 （SW バージョン 3 まで標準応答）	
関連性	SD 42440: FRAME_OFFSET_INCR_PROG	

42470 SD 番号	CRIT_SPLINE_ANGLE **スプラインと多項式補間のリミット角度およびコンプレッサ**	
初期設定：0.0, ...	最小入力リミット：-	最大入力リミット：-
変更が有効になるための条件： リセット	保護レベル：2/7	単位：度
データタイプ：DOUBLE	適用開始 SW バージョン：4.1	
意味：	MD 42470 は，多項式補間 (POLY) 用の正接遷移であると見なすことができる 2 つの隣接ブロックの間の最大角度を決定する。角度が小さい場合，正確な正接遷移を得るために多項式が適用される。A または C スプライン補間の場合，ブロックの遷移はコーナに大きい角度を使って形成される（スプラインは，プログラムされたブロックに正接して終了し再スタートする）。オンラインコンプレッサがアクティブ (COMPON) である時，圧縮は大きな角度を使ってブロック境界をカバーしない。つまり大きな角度は保留される。	
関連性		

42480 SD 番号	STOP_CUTCOM_STOPRE **工具径補正用のアラームリアクションと前処理停止**	
初期設定：1	最小入力リミット：-	最大入力リミット：-
変更が有効になるための条件： リセット	保護レベル：7	単位：-
データタイプ：BOOLEAN	適用開始 SW バージョン：4.1	
意味：	この設定データが TRUE の場合，前処理停止のブロック処理とアクティブ工具径補正は停止され，ユーザ確認後初めて継続される。 値が FALSE の場合，処理はプログラムのそのような箇所で中断されない。	
関連性		

42900 SD 番号	MIRROR_TOOL_LENGTH **工具長オフセットのミラー**	
初期設定：0	最小入力リミット：-	最大入力リミット：-
変更が有効になるための条件： リセット	保護レベル：7	単位：-
データタイプ：BOOLEAN	適用開始 SW バージョン：5.1	
意味：	この設定データがゼロに等しくない場合，その関連軸がミラーされるツール長コンポーネント ($TC_DP3[..., ...] から $TC-DP5[...,...]) およびツールベース寸法のコンポーネント ($TC_DP21[..., ...] から $TC_DP23[..., ...]) もミラーされる。つまり，それらの符号は反転される。摩耗値はミラーされない。これらも同様にミラーされる場合，設定データ $SC_MIRROR_TOOL_WEAR がイネーブルしなければならない。	
関連性	SD 42910: MIRROR_TOOL_WEAR SD 42920: WEAR_SIGN_CUTPOS SD 42930: WEAR_SIGN SD 42940: TOOL_LENGTH_CONST	

42910 SD 番号	MIRROR_TOOL_WEAR **ツール長コンポーネントの摩耗値のミラー**	
初期設定：0	最小入力リミット：-	最大入力リミット：-
変更が有効になるための条件： リセット	保護レベル：7	単位：-
データタイプ：BOOLEAN	適用開始 SW バージョン：5.1	
意味：	この設定データがゼロに等しくない場合，その関連軸がミラーされるツール長コンポーネントの摩耗値もまたミラーされる。 つまりそれらの符号は反転される。摩耗値は，実際のツール摩耗値，合計オフセットおよびセットアップオフセットの合計である。	
関連性	SD 42900: MIRROR_TOOL_LENGTH SD 42920: WEAR_SIGN_CUTPOS SD 42930: WEAR_SIGN SD 42940: TOOL_LENGTH_CONST	

42920 SD 番号	WEAR_SIGN_CUTPOS **加工面の摩耗値のミラー**	
初期設定：0	最小入力リミット：-	最大入力リミット：-
変更が有効になるための条件： リセット	保護レベル：7	単位：-
データタイプ：BOOLEAN	適用開始 SW バージョン：5.1	
意味：	この設定がゼロに等しくない時，関連するツール点方向を持つツール（旋削および研削ツール，ツールタイプ 400 から 599）の場合，加工面の摩耗コンポーネントの符号評価はツール点方向に依存する。この設定データは，関連するツール点方向を持たないツールタイプの場合無視される。次のテーブルで，$SC_WEAR_SIGN_CUTPOS がゼロでない場合，その符号が反転されるコンポーネントには × 印が付いている。 ツール点方向　　長さ 1　　長さ 2 1 2　　× 3　　×　　× 4　　× 5 6 7　　× 8　　× 9 $SC_WEAR_SIGN_CUTPOS および $SC_MIR-ROR_ TOOL_WEAR によって制御される符号評価は互いに独立している。つまりコンポーネントの符号が 2 つの設定データによって反転される場合，生じる符号は一定である。	
関連性	SD 42910: MIRROR_TOOL_WEAR SD 42900: MIRROR_TOOL_LENGTH SD 42930: WEAR_SIGN SD 42940: TOOL_LENGTH_CONST	

42930 SD 番号	WEAR_SIGN **全ての摩耗値の符号反転**	
初期設定：0	最小入力リミット：-	最大入力リミット：-
変更が有効になるための条件： リセット	保護レベル：7	単位：-
データタイプ：BOOLEAN	適用開始 SW バージョン：5.1	
意味：	この設定データがゼロに等しくない場合，全ての摩耗寸法の符号が反転される。これは，ツール長および，工具半径，丸め半径などの他の変数の両方に影響する。正の摩耗寸法を入力すると，ツールは「短く」「細く」なる。	
関連性	SD 42910: MIRROR_TOOL_WEAR SD 42920: WEAR_SIGN_CUTPOS SD 42900: MIRROR_TOOL_LENGTH SD 42940: TOOL_LENGTH_CONST	

42940 SD 番号	TOOL_LENGTH_CONST 加工面（G17 ～ G19）の変更時，工具長補正の割当ての維持	
初期設定：0	最小入力リミット：-	最大入力リミット：-
変更が有効になるための条件： リセット	保護レベル：7	単位：-
データタイプ：DWORD	適用開始 SW バージョン：5.1	
意味：	この設定がゼロに等しくない場合，ツール長コンポーネント（長さ，摩耗，ツールのベース寸法）とジオメトリ軸の間の割当ては，加工面 (G17 ～ G19) が変更されても変更されない。ジオメトリ軸への工具長補正の割当ては，下記のテーブルにしたがって設定データの値によって決定される。研削および旋削ツール（ツールタイプ 400 ～ 599）と他のツール（フライス加工ツールが代表的）の割当ての間で区別がなされる。しかし，ジオメトリ軸 1 ～ 3 が X，Y，Z の名称を持つと仮定した表の中で，軸識別子ではなく軸シーケンスがオフセットを軸へ割当てる際の決定要素である。 研削および旋削ツール（ツールタイプ 400 ～ 599）に対する割当て： 面　長さ 1　長さ 2　長さ 3 17　Y　X　Z 18*　X　Z　Y 19　Z　Y　X -17　X　Y　Z -18　Z　X　Y -19　Y　Z　X * ゼロでなく，またリストされた 6 つの値の 1 つに等しくない全ての値は，値 18 と同じ方法で扱われる。数字が同じで異なる符号を持つ値の場合，長さ 3 の割当ては同じであるが，長さ 1 と長さ 2 はスワップされる。 研削および旋削ツール以外の全てのツール（ツールタイプ <400 または >599）に対する割当て： 面　長さ 1　長さ 2　長さ 3 17*　Z　Y　X 18　Y　X　Z 19　X　Z　Y -17　Z　X　Y -18　Y　Z　X -19　X　Y　Z * ゼロでなく，リストされた 6 つの値の 1 つと等しくない全ての値は，値 17 と同じ方法で扱われる。数字が同じで異なる符号を持つ値の場合，長さ 1 の割当ては同じであるが，長さ 2 と 3 はスワップされる。	
関連性	SD 42910: MIRROR_TOOL_WEAR SD 42920: WEAR_SIGN_CUTPOS SD 42930: WEAR_SIGN SD 42900: MIRROR_TOOL_LENGTH	

1.19.5 信号の説明

なし

1.19.6 例

■ 角度調整可能なツールホルダの例

次の例は，Y 軸回り一回転によって完全に記述されるツールホルダを使用しています。したがって，回転軸（ブロック N20）を定義するのに 1 つの値のみを入力すれば充分です。

ブロック N50 ～ N70 は，半径 5mm および長さ 20mm を使ってエンドミルを記述します。

ブロック N90 は，Y 軸回り 37 度回転を定義します。

ブロック N120 は工具径補正を起動し，Y 軸回り 37 度回転を使って次のブロックのオフセットを記述する全ての設定がなされます。

```
N10                          ：ツールホルダ 1 の定義
N20 $TC_CARR8[1] = 1  ：Y 方向への最初の回転軸のコンポーネント
N30
N40                          ：ツールオフセットメモリ T1/D1 の定義
N50 $TC_DP1[1,1] = 120  ：シャフト型フライス加工カッタ
N60 $TC_DP3[1,1]= 20     ：長さ 1
N70 $TC_DP6[1,1] = 5      ：半径
N80
N90 ROT Y37                  ：y 軸回り 37° 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
```

1.19.7 データフィールド，リスト

■ インタフェース信号

DB 番号	ビット，バイト	名前	参照
チャンネル別			
21, ...	61.0	T 機能 1 チェンジ	
21, ...	62.0	D 機能 1 チェンジ	
21, ...	116 ~ 119	T 機能 1	
21, ...	128 ~ 129	D 機能 1	
21, ...	214	グループ 7 用アクティブ G 機能	1.10（K1）
21, ...	223	グループ 16 用アクティブ G 機能	1.10（K1）
21, ...	224	グループ 17 用アクティブ G 機能	1.10（K1）
21, ...	225	グループ 18 用アクティブ G 機能	1.10（K1）
21, ...	230	グループ 23 用アクティブ G 機能	1.10（K1）

■ マシンデータ

番号	識別子	名前	参照
一般事項 ($MN_ ...)			
18082	MM_NUM_TOOL	NCK が管理できるツール数 (SRAM)	2.14（S7）
18088	MM_NUM_TOOL_CARRIER	ツールホルダ数	
18094	MM_NUM_CC_TDA_PARAM	TOA データ数	FBW, 2.14（S7）
18096	MM_NUM_CC_TOA_PARAM	ツール別にセットアップされ CC によって評価されることができる TOA データ数 CC	FBW, 2.14（S7）
18100	MM_NUM_CUTTING_EDGES_IN_TOA	TOA モジュール別ツールオフセット	2.14（S7）
18102	MM_TYPE_OF_CUTTING_EDGE	アクティブフラット D 番号管理 (SW4 以降)	
18105	MM_MAX_CUTTING_EDGE_NO	スピンドル番号として解釈されるアドレス拡張（SW5 以降）	
18106	MM_MAX_CUTTING_EDGE_PERTOOL	ツール別切削エッジの最大数（SW5 以降）	
18108	MM_NUM_SUMCORR	NCK の全ての合計オフセット数（SW5 以降）	
18110	MM_MAX_SUMCORR_PER_CUTTEDGE	切削エッジ別合計オフセットの最大数（SW5 以降）	
18112	MM_KIND_OF_SUMCORR	NCK の合計オフセットのプロパティ（SW5 以降）	
チャンネル別 ($machine code_ ...)			
20110	RESET_MODE_MASK	パワーアップとリセットパートプログラム後の制御基本設定の定義	1.11（K2）
20120	TOOL_RESET_VALUE	パワーアップ中あるいはリセット時またはパートプログラム終了時，工具長補正が MD 20110 の機能として選択されるツールの定義	1.11（K2）
20121	TOOL_PRESEL_RESET_VALUE	パワーアップ中あるいはリセットまたはパートプログラム終了時，工具長補正が MD 20110 の機能として選択される事前選択ツールの定義	1.11（K2）
20126	TOOL_CARRIER_RESET_VALUE	リセット時のアクティブツールホルダ	

番号	識別子	名前	参照
20130	CUTTING_EDGE_RESET_VALUE	パワーアップ中あるいはリセットまたはパートプログラム終了時，工具長補正が MD 20110 の機能として選択されるツール切削エッジの定義	1.11（K2）
20140	TRAFO_RESET_VALUE	パワーアップ中あるいはリセットまたはパートプログラム終了時，MD 20110 の機能として選択される変換ブロックの定義	1.11（K2）
20202	WAB_MAXNUM_DUMMY_BLOCKS	SAR を使った移動動作のないブロックの最大数	
20204	WAB_CLEARANCE_TOLERANCE		
20210	CUTCOM_CORNER_LIMIT	工具径補正を使った補正ブロックの最大角度	
20220	CUTCOM_MAX_DISC	DISC の最大値	
20230	CUTCOM_CURVE_INSERT_LIMIT	工具径補正を使ったインタセクション計算の最小値	
20240	CUTCOM_MAXNUM_CHECK_BLOCKS	工具径補正を使った予想輪郭計算用のブロック	
20250	CUTCOM_MAXNUM_DUMMY_BLOCKS	TRC を使った移動動作のないダミーブロックの最大数	
20252	CUTCOM_MAXNUM_SUPPR_BLOCKS	オフセット抑制を使ったブロックの最大数（SW4 以降）	
20256	CUTCOM_INTERS_POLY_ENABLE	多項式に可能なインタセクションプロセス（SW4 以降）	
20270	CUTTING_EDGE_DEFAULT	ツールチェンジ後選択される切削エッジ	
20360	TOOL_PARAMETER_DEF_MASK	ツールパラメータの影響を定義する	
20610	ADD_MOVE_ACCEL_RESERVE	オーバレイ動作の加速保留	1.10（K1）
21080	CUTCOM_PARALLEL_ORI_LIMIT	3D 半径補正を使ったパス正接とツールの向きとの間のリミット角度	3.13（W5）
22530	TOCARR_CHANGE_M_CODE	ツールホルダチェンジ用の M コード	
22550	TOOL_CHANGE_MODE	M 機能を使った新しいツールオフセット	
22560	TOOL_CHANGE_M_CODE	ツールチェンジ用の M 機能	
28085	MM_LINK_TOA_UNIT	TO ユニットのチャンネルへの割当て	FBW, 2.14（S7）
20132	SUMCORR_RESET_VALUE	合計オフセットを選択するための番号（SW5 以降）	
20272	SUMCORR_DEFAULT	新しい切削エッジオフセットを起動するための番号（SW5 以降）	
22562	TOOL_CHANGE_ERROR_MODE	プログラムされたツールチェンジ時のエラー応答（SW5 以降）	
?????	T_M_ADDRESS_EXT_IS_SPINO	アドレス拡張としてのスピンドル番号（SW5 以降）	FBW, 2.14（S7）

■ 設定データ

番号	識別子	名前	参照
42442	TOOL_OFFSET_INCR_PROG	相対プログラミングでのツールオフセットの後退	
42470	CRIT_SPLINE_ANGLE	スプラインおよび多項式補間のリミット角度とコンプレッサ	
42480	STOP_CUTCOM_STOPRE	工具径補正に対するアラーム反応および前処理停止（SW4 以降）	
42900	MIRROR_TOOL_LENGTH	工具長オフセットのミラー	
42910	MIRROR_TOOL_WEAR	工具長補正の摩耗値のミラー	
42920	WEAR_SIGN_CUTPOS	加工面の摩耗値のミラー	
42930	WEAR_SIGN	全ての摩耗値の符号の反転	
42940	TOOL_LENGTH_CONST	加工面 （G17 ～ G19）の変更時，工具長補正の割当てを維持する	

■ アラーム

アラームの詳細説明については次を参照してください。

参照： アラーム一覧

あるいは MMC 101/102/103 を有するシステムのオンラインヘルプ

Yaskawa Siemens CNC シリーズ

本製品の最終使用者が軍事関係であったり，用途が兵器などの製造用である場合には，「外国為替及び外国貿易法」の定める輸出規制の対象となることがありますので，輸出される際には十分な審査及び必要な輸出手続きをお取りください。

製品改良のため、定格、寸法などの一部を予告なしに変更することがあります。
この資料についてのお問い合わせは、当社代理店もしくは、下記の営業部門にお尋ねください。

製造

株式会社 安川電機　　シーメンスAG

販売

シーメンス・ジャパン株式会社

工作機械営業本部
東京都品川区大崎1-11-1 **ゲートシティ大崎ウエストタワー**　〒141-8644
TEL（03）3493-7411　FAX（03）3493-7422

アフターサービス

カスタマーサービス事業本部
TEL 0120-996095（フリーダイヤル）　FAX (03)3493-7433

シーメンス・ジャパン株式会社
http://www.siemens.co.jp

02- WSC02, 02-WSC09

無断転載・複製を禁止