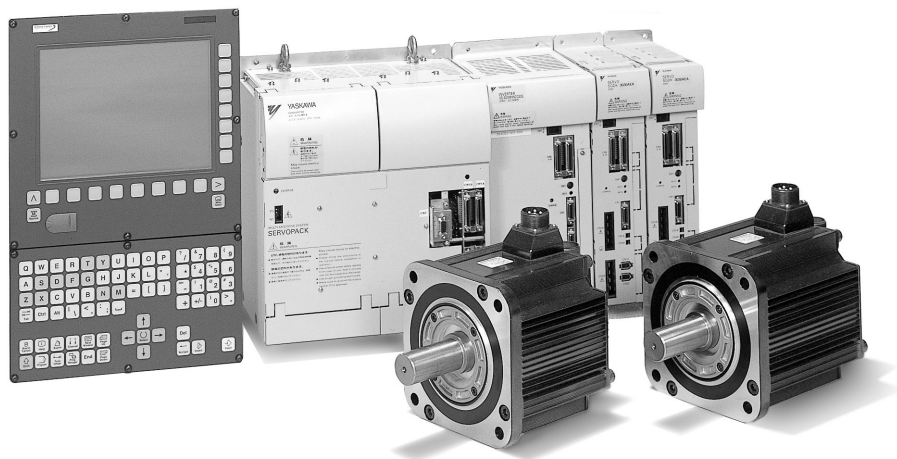


Yaskawa Siemens CNC シリーズ

シンクロナイズドアクション説明書



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が、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。

Yaskawa Siemens 840DI

シンクロナイズドアクション説明書

	概略説明	1
	詳細説明	2
	補足条件	3
機能説明	データの説明 (MD, SD)	4
	信号の説明	5
製造業者／サービス文書	例	6
	データ, フィールド, リスト	7

対象制御装置

制御装置
Yaskawa Siemens 840DI

03.01 版

Yaskawa Siemens 文書

版の履歴

今回の版の概略説明および今までに作成された版を下記に示します。

「備考」欄のコードが、各版のステータスを示しています。

「備考」欄のステータスコードの意味は次のとおりです。

- A..... 新規作成
- B..... 新しいオーダ番号で印刷し直した未改訂の文書
- C..... 新しいステータスの改訂版
前回の版以降に実際に変更があったページには、そのページのヘッダ部分に新しい版のコードが示されています。

版	オーダ番号	備考
03.01	NCSI-SP02-16	A

書面による許可なしに、本文書の一部または全部を使用、複製することはできません。違反行為があった場合、損害賠償金が課せられます。使用モデルまたはデザインの特許登録による著作権を含むすべての権利を当社は所有しています。

本文書に説明のない他の機能でも制御装置で実行できる場合がありますが、そのような機能は新しい制御装置やサービス時に利用できるとは限りません。

本文書の記述と、対象となるハードウェアおよびソフトウェアとが一致しているかどうかは十分に確認されています。しかし相違点がまったくないとは言えず、完全に一致しているとは保証できません。本文書に記載されている情報は定期的に検討され、必要な変更は次の版に反映されます。さらなる改善のために皆様のご意見をお待ちしています。

本内容は予告なしに変更されることがあります。

関連マニュアル

- 関連するマニュアルについては，下表に示すものがあります。必要に応じてご覧ください。
- 製品の仕様，使用制限などの条件を十分ご理解いただいたうえで，製品をご活用ください。

マニュアル名称	資料番号
Yaskawa Siemensu 840DI 結合説明書 ハード編	NCSI-SP02-01
Yaskawa Siemensu 840DI 結合説明書 機能編	DE0400309
Yaskawa Siemensu 840DI PLC トレーニングマニュアル	DE0400515
Yaskawa Siemensu 840DI ユーザーズマニュアル 操作編	NCSI-SP02-04
Yaskawa Siemensu 840DI ShopMill セットアップマニュアル	NCSI-SP02-05
Yaskawa Siemensu 840DI ユーザーズマニュアル プログラミング編 ISO 互換 G コード説明書（マシニング用）（制作中）	NCSI-SP02-20
Yaskawa Siemensu 840DI ユーザーズマニュアル プログラミング編 ISO 互換 G コード説明書（旋盤用）（制作中）	NCSI-SP02-21
Yaskawa Siemensu 840DI ユーザーズマニュアル プログラミング編 基本説明書	NCSI-SP02-06
Yaskawa Siemensu 840DI ユーザーズマニュアル プログラミング編 上級説明書	NCSI-SP02-07
Yaskawa Siemensu 840DI ユーザーズマニュアル プログラミング編 サイクル説明書	NCSI-SP02-08
Yaskawa Siemensu 840DI ユーザーズマニュアル プログラミング編 計測サイクル説明書	NCSI-SP02-09
Yaskawa Siemensu 840DI 保守説明書	NCSI-SP02-10
Yaskawa Siemensu 840DI 保守説明書 サービスマンハンドブック（制作中）	NCSI-SP02-19
Yaskawa Siemensu 840DI 保守説明書 別冊付録 一覧表	NCSI-SP02-11
Yaskawa Siemensu 840DI 保守説明書 別冊付録 アラーム診断ガイド	NCSI-SP02-12
Yaskawa Siemensu 840DI API 取扱説明書 HMI プログラミングパッケージ 基礎編	NCSI-SP02-13
Yaskawa Siemensu 840DI API 取扱説明書 HMI プログラミングパッケージ COM および OPC クライアント編	NCSI-SP02-14
Yaskawa Siemensu 840DI API 取扱説明書 HMI プログラミングパッケージ インストールガイド	NCSI-SP02-15
Yaskawa Siemensu 840DI シンクロナイズドアクション説明書（本書）	NCSI-SP02-16
Yaskawa Siemensu 840DI SINCOM コンピュータリンク説明書（制作中）	NCSI-SP02-17
Yaskawa Siemensu 840DI ツールマネージメント説明書（制作中）	NCSI-SP02-18

1 概略説明	1-1
2 詳細説明	2-1
2.1 シンクロナイズドアクションの構成要素	2-2
2.1.1 モーションシンクロナズアクションの定義	2-8
2.1.2 シンクロナイズドアクションの実行	2-8
2.1.3 可能なアクションのリスト	2-9
2.2 リアルタイム評価および計算	2-10
2.3 シンクロナイズドアクション用の特殊リアルタイム変数	2-15
2.3.1 マーカ／カウンタ変数	2-15
2.3.2 タイマ	2-16
2.3.3 シンクロナイズドアクションパラメータ	2-16
2.3.4 R パラメータ	2-17
2.3.5 マシンおよびセッティングデータ	2-17
2.3.6 FIFO 変数（循環メモリ）	2-18
2.3.7 シンクロナイズドアクションに関連するシステム変数のリスト	2-21
2.4 シンクロナイズドアクション中のアクション	2-40
2.4.1 補助機能 M, S, および H の PLC への出力	2-43
2.4.2 リアルタイム変数の設定（書込み）および読取り	2-45
2.4.3 SW カム位置および時間の変更（セッティングデータ）	2-46
2.4.4 FCTDEF	2-47
2.4.5 多項式評価 SYNFACT	2-49
2.4.6 オンラインツールオフセット FTOC	2-55
2.4.7 RDISABLE	2-56
2.4.8 STOPREOF	2-57
2.4.9 DELDTG	2-57
2.4.10 プログラムされた軸モーションのディスエーブル	2-59
2.4.11 指令軸のスタート	2-60
2.4.12 シンクロナイズドアクションからの軸送り	2-62
2.4.13 シンクロナイズドアクションからの軸のスタート／停止	2-63
2.4.14 シンクロナイズドアクションからの主軸モーション	2-63
2.4.15 シンクロナイズドアクションからの実際値の設定	2-67
2.4.16 カップリング軸およびカップリングの起動／停止	2-68
2.4.17 シンクロナイズドアクションからの測定値	2-71
2.4.18 チャンネル同期用待機マーカの設定および削除	2-75
2.4.19 アラームの設定／エラーリアクション	2-76
2.5 テクノロジサイクルのコール	2-77
2.5.1 シンクロナイズドアクション, テクノロジサイクル, パートプログラム （および PLC）の協調	2-80
2.6 シンクロナイズドアクションの制御および保護	2-82
2.6.1 PLC を介しての制御	2-82
2.6.2 保護されたシンクロナイズドアクション	2-84
2.7 特定の運転テータスでのシンクロナイズドアクションに対する 制御系の応答	2-85
2.7.1 電源オン	2-85
2.7.2 リセット	2-85
2.7.3 NC 停止	2-86
2.7.4 運転モードの変更	2-86
2.7.5 プログラムの終了	2-87
2.7.6 プログラム終了に対するアクティブなシンクロナイズドアクションの 応答および運転モードの変更	2-87
2.7.7 ブロックサーチ	2-88
2.7.8 ASUP によって中断させられたプログラム	2-88
2.7.9 REPOS	2-88
2.7.10 アラームに対する応答	2-89
2.8 構成	2-90
2.8.1 構成可能性	2-90

2.9 診断	2-92
2.9.1 シンクロナイズドアクションのステータスの表示	2-93
2.9.2 リアルタイム変数の表示	2-93
2.9.3 リアルタイム変数のログ	2-94
3 補足条件	3-1
4 データの説明 (MD, SD)	4-1
4.1 一般マシンデータ	4-2
4.2 チャンネル別マシンデータ	4-3
4.3 軸／主軸別マシンデータ	4-6
4.4 セットアップデータ	4-9
5 信号の説明	5-1
6 例	6-1
6.1 シンクロナイズドアクション中の条件例	6-2
6.2 シンクロナイズドアクションからの SD/MD の読取りおよび書込み	6-3
6.3 適用制御例	6-5
6.3.1 可変上限値を使ったクリアランス制御	6-5
6.3.2 送り制御	6-6
6.3.3 正規化されたパスの機能としての制御速度	6-8
6.4 2 軸間の安全クリアランスの監視	6-9
6.5 実行時間を R パラメータに保存する	6-9
6.6 連続測定時の「センタリング」	6-10
6.7 シンクロナイズドアクションによっての軸カップリング	6-13
6.7.1 マスタ軸へのカップリング	6-13
6.7.2 マスタ軸カップリングによる非円形グラインディング	6-14
6.7.3 即時の切断	6-17
6.8 テクノロジサイクル「主軸の位置決め」	6-19
6.9 TCC/BAZ エリアでのシンクロナイズドアクション	6-20
7 データフィールド, リスト	7-1
7.1 インタフェース信号	7-2
7.2 マシンデータ	7-2
7.3 アラーム	7-3

1 概略説明

シンクロナイズドアクションの定義

モーションシンクロナイズドアクション（以降はシンクロナイズドアクションという）とは、パートプログラムの実行と同期して NCK の補間サイクル中に評価される、ユーザがプログラムする命令の総称です。シンクロナイズドアクションの中にプログラムされた条件が満足されるか、あるいは何も条件が規定されていないならば、命令に割当てられたアクションがパートプログラムと同期して実行されます。

アプリケーション

下記にシンクロナイズドアクションの幅広いアプリケーションのいくつかを示します。これらの例からも分かるように、シンクロナイズドアクションとしてプログラムできるアクションは非常に便利です。

- 補助機能の PLC への出力
- リアルタイム変数の読書き
- 軸と主軸の位置決め
- 以下のシンクロナイズドプロシージャの起動：
 - 読込みディスエーブル
 - 移動距離削除
 - エンド前処理停止
- テクノロジサイクルの起動
- 機能値のオンライン計算
- オンラインツールオフセット
- カップリングモーションの起動と停止
- 測定の実施
- シンクロナイズドアクションのイネーブル／ディスエーブル

シンクロナイズドアクションのアプリケーションは「詳細説明」のセクションで詳しく説明されています。

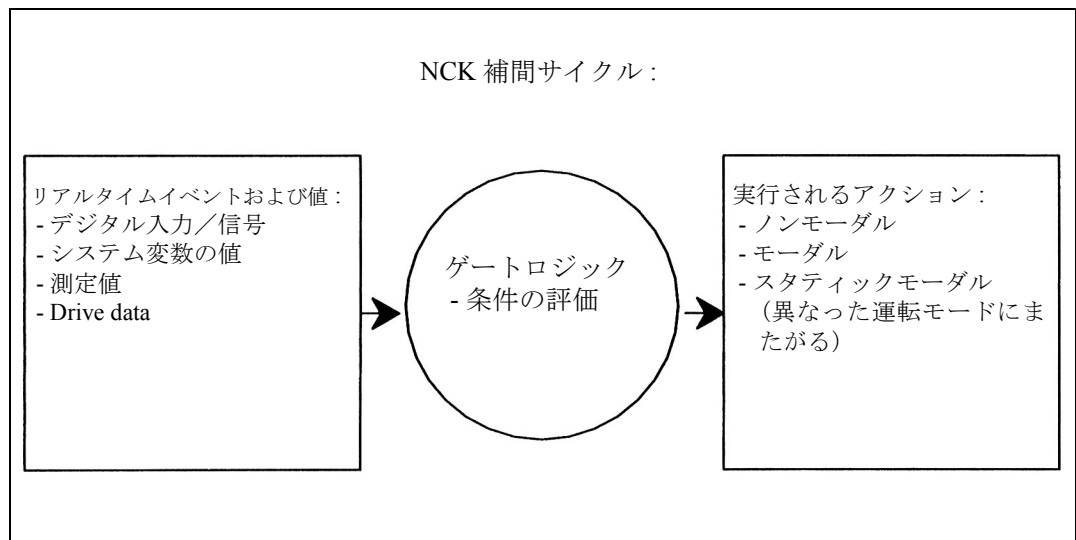


図 1.1 シンクロナイズドアクションの概略図

シンクロナイズドアクションのプログラミング方法の詳細については次を参照してください：

参照： プログラミングガイド・上級編

本書の概要：

- シンクロナイズドアクションの基本条件が 2 章で説明されています。
- 必要となるマシンデータが 4 章で説明されています。
- アプリケーション例が 6 章で説明されています。

(注) 本書はソフトウェアバージョン 5 で提供されている機能に適用されます。ソフトウェアバージョン 3 およびそれ以前のシンクロナイズドアクションについては次を参照してください：

参照： シンクロナイズドアクション (S5)

2 詳細説明

2.1 シンクロナイズドアクションの構成要素

シンクロナイズドアクションの構成

構成要素：	有効性 識別番号	頻度	条件および アクション用の G コード	条件	アクション コードワード (固定)	アクション用 の G コード	アクションあ るいはテクノ ロジサイクル (2.5 「テクノ ロジサイクル のコール」を 参照)
例：	IDS=1	EVERY	G70	\$AAA_IM[B]> 15	DO	G71	POS[X]= 100

シンクロナイズドアクションの構成要素：

- 有効性：
 - 識別番号がある場合
 - 識別番号がない場合
- 頻度
- 条件およびアクション用の G コード (SW 5 以降)
- 条件
- アクション用の G コード (SW 5 以降)
- アクション／テクノロジサイクル

各構成要素を次に詳しく説明します。

有効性, ID 番号

シンクロナイズドアクションの有効性の範囲を定義するには次の 3 つの方法があります：

- ステータスなし
- ID
- IDS

指定された有効性なし

指定された有効性のないシンクロナイズドアクションは、ノンモーダルアクションを実行します。つまり、このシンクロナイズドアクションはその次の実行可能ブロックにしか適用されません。

実行可能ブロックは次のいずれかです。

- モーションブロック、または
- マシンアクション（前処理停止、補助機能出力）を含むブロック

ノンモーダルシンクロナイズドアクションは、AUTOMATIC モードでしか動作できません。

ID

有効性識別子 ID 付きのシンクロナイズドアクションは、その後にプログラムされたブロックでモーダル的にアクティブです。このシンクロナイズドアクションは、AUTOMATIC モードでしか動作できません。

制限：

- 同一の識別番号を持つ別のシンクロナイズドアクションがプログラムされるまで、ID アクションは有効なままです。あるいは、
- CANCEL(i) でキャンセルされるまで、ID アクションは有効なままです（2.5.1「シンクロナイズドアクション、テクノロジーサイクル、パートプログラム（および PLC）」参照）。

IDS

ボキャブラリワード "IDS" を使用してプログラムされる静的にアクティブなシンクロナイズドアクションは、すべての運転モードで動作できます。このシンクロナイズドアクションは、スタティックシンクロナイズドアクションとも呼ばれます。

ID または IDS を使用してプログラムされたシンクロナイズドアクションは、パートプログラムから削除されます。

識別番号

1 ～ 255 までの識別番号は、モーダルシンクロナイズドアクション (ID, IDS) に割当てられます。これらの番号は、シンクロナイズドアクションの相互協調で重要となります。2.5.1「シンクロナイズドアクション、テクノロジーサイクル、パートプログラム（および PLC）の協調」を参照してください。1 ～ 64 の識別番号を持つモーダル／スタティックシンクロナイズドアクションは、PLC からディスエーブルまたはイネーブルすることができます。2.6.1「PLC を介しての制御」を参照してください。

スタティックシンクロナイズドアクションのアプリケーション：

- AC グラインディング（JOG モードでもアクティブ）
- 安全統合用のゲートロジック
- すべての運転モードでの監視機能、マシンステータスに対するリアクション
- ツールチェンジの最適化
- 周期マシン

例：

IDS=1 EVERY \$A_IN[1]==1 DO POS[X]=100	すべての運転モード
ID=2 EVERY \$A_IN[1]==0 DO POS[X]=0	AUTOMATIC

（注）プログラム実行中は、次のアクションは AUTOMATIC モードでしか動作できません：

STOPREOF,
DELDTG

頻度

ボキャブラリワード（下表を参照）は、その後に指定された条件が何回スキャンされなければならないか、またその条件が満足された場合、関連アクションが何回実行されなければならないかを示すためにプログラムされます。こうしたボキャブラリワードは、シンクロナイズドアクション条件に絶対必要な構成要素です。

表 2.1 頻度ボキャブラリワードの影響

ボキャブラリワード	スキャン頻度
なし	スキャン頻度がプログラムされていない場合、アクションは補間サイクルごとに周期的に実行される。
WHENEVER	条件が満足されれば、関連アクション／テクノロジーサイクルが補間サイクルごとに周期的に実行される。
FROM	条件が1度満足されると、シンクロナイズドアクションがアクティブである間は、アクション／テクノロジーサイクルが補間サイクルごとに周期的に実行される。
WHEN	条件が満足されると直ちにアクション／テクノロジーサイクルが1回実行される。アクションが1回実行されると、条件はそれ以上チェックされることがない。
EVERY	条件が満足されれば、アクション／テクノロジーサイクルが1回起動される。アクション／テクノロジーサイクルは、条件が"FALSE"ステータスから"TRUE"ステータスへ切替るたびに実行される。 ボキャブラリワード WHEN とは対照的に、条件のチェックは、シンクロナイズドアクションが削除あるいはディスエーブルされるまで、アクション／サイクル実行後も継続される。

テクノロジーサイクルの詳細については、2.5「テクノロジーサイクルのコール」を参照してください。

削除

アクティブなシンクロナイズドアクションがパートプログラムから CANCEL で選択解除（削除）された場合でも、現在アクティブなアクションは有効なままです。位置決めモーションはプログラムされたとおりに完了されます。指令 CANCEL をプログラムすることにより、モーダル、つまり、静的にアクティブなシンクロナイズドアクションを削除することができます。

シンクロナイズドアクションが、自身が開始させた位置決め軸モーションの進行中に削除されたとしても、その位置決めモーションは正しく実行されるまで継続します。チャンネル停止も同様にシンクロナイズドアクション／テクノロジーサイクルからの位置決め動作をキャンセルします。

条件およびアクション用の G コード

SW バージョン 5 以降では、G コードをシンクロナイズドアクションにプログラムすることができます。これにより、現在のパートプログラムステータスから独立して、定義された設定値を条件評価のために存在させ、アクション／テクノロジーサイクルを実行することができます。シンクロナイズドアクションは、トリガ条件が満足されると、任意の時間に定義された初期ステータスからアクションを実行するよう要求されるので、シンクロナイズドアクションをプログラム環境から分離する必要があります。

アプリケーション：

G コード G70, G71, G700, G710 による条件評価およびアクション用の測定系の定義

（注）SW バージョン 5 では、シンクロナイズドアクションでの G コードの使用はこれら 4 つの G コードに制限されています。

条件について指定された G コードは条件評価について有効ですが、別の G コードがアクションについて指定されていない場合はそのアクションについても有効です。
条件の各部分について、G コードグループの G コードは 1 つしかプログラムできません。

条件

アクション／テクノロジーサイクルの実行を条件（論理式）に依存させることができます。

条件は補間サイクル中にチェックされます。条件がプログラムされていない場合、アクションは IPO サイクルごとに 1 回実行されます。

バージョン 3 以前の SW では、2 つの条件、すなわち、リアルタイム変数と前処理中に計算された式との比較および 2 つのリアルタイム変数の比較が許可されています。

例：

```
WHENEVER $AA_IM[X] > 10.5*SIN(45) DO ...    あるいは
WHENEVER $AA_IM[X] > $AA_IM[X1] DO ...
```

SW バージョン 4 では、追加条件、つまり、ブール演算を使用した比較の結合が利用できます。NC 言語で書かれたブール演算子がこの目的に使用されます。

NOT, AND, OR, XOR, B_OR, B_AND, B_XOR, B_NOT.

例：

```
WHENEVER ($A_IN[1]==1) OR ($A_IN[3]==0) DO ...
```

; 入力 1 が適用されるか、あるいは、入力 3 が適用されない場合 ...

2 つかそれ以上のリアルタイム式を 1 つの条件の中で比較できます。

同じタイプの変数間での比較あるいは部分式間での比較を実行できます。

例：

```
WHEN $AA_IM[X2] <= $AA_IM[X1] +.5 DO $AA_OVR[X1]=0
```

; 安全クリアランスが超過されると、停止

リアルタイム式を適用するためのオプションについては、2.2「リアルタイム評価および計算」で説明しています。条件がプログラムされているときは、2.3.7「シンクロナイズドアクションに関連するシステム変数のリスト」に示されているすべてのシステム変数を指定することができます。以下のデータも同様に指定することができます。

- マシンデータ、例、\$SMN_..., \$SMC_..., \$SMA_...
- セッティングデータ、例、\$SSN_..., \$SSC_..., \$SSA_...

（注）・GUD 変数は使用できません

- R パラメータは \$R... を使用して指定されます
- 加工中に設定値が変化するセッティングデータおよびマシンデータは、\$SS_... / \$SM_... を使用してプログラムされなければなりません。

条件のその他の例については 6.1「シンクロナイズドアクション中の条件例」を参照してください。

アクション用の G コード

G コードは、ブロック中のすべてのアクション、およびテクノロジサイクルについて、条件用の G コードとは異なる G コードを指定することができます。テクノロジサイクルがアクション部分に含まれている場合、G コードは、テクノロジサイクルエンド後も次の G コードまで、すべてのアクションに対してモーダル的にアクティブなままです。

1 アクション部分につき、G コードグループの G コードは 1 つしかプログラムできません。

アクション

すべてのシンクロナイズドアクションには 1 つあるいは複数のプログラムされたアクションか、1 つのテクノロジサイクルが含まれます。これらは、適切な条件が満足されると実行されます。1 つのシンクロナイズドアクションの中に複数のアクションがプログラムされている場合、これらのアクションは同一の補間サイクル内で実行されます。

例： WHEN \$AA_IM[Y] >= 35.7 DO M135 \$A_OUT[1]=1
 Y 軸の実際値が 35.7 であるか、それ以上である場合、M135 が PLC に出力され、同時に出力 1 がセットされる。

プログラム／テクノロジサイクル

プログラム（名称）をアクションとして指定することもできます。このプログラムは、シンクロナイズドアクションの中に個別にプログラムできるアクションであればどのアクションでも含めることができます。このようなプログラムはテクノロジサイクルとも呼ばれます。テクノロジサイクルとは、補間サイクル中に逐次的に処理される一連のアクションのことです。2.5 「テクノロジサイクルのコール」を参照してください。

アプリケーション：単一軸プログラム、周期マシン

処理

パートプログラムのブロックは、プログラム前処理段階で準備され、次に保存され、補間レベル（メインラン）で逐次的に実行されます。変数はブロック準備中にアクセスされます。リアルタイム変数（たとえば、実際値）が使用されると、ブロック準備は中断され、前のブロックまでの現在のリアルタイム値の供給が可能となります。

シンクロナイズドアクションは、準備されたブロックと合わせて、前処理された形で補間器へ送られます。使用されるリアルタイム変数は補間サイクル中に評価されます。ブロック準備は中断されません。

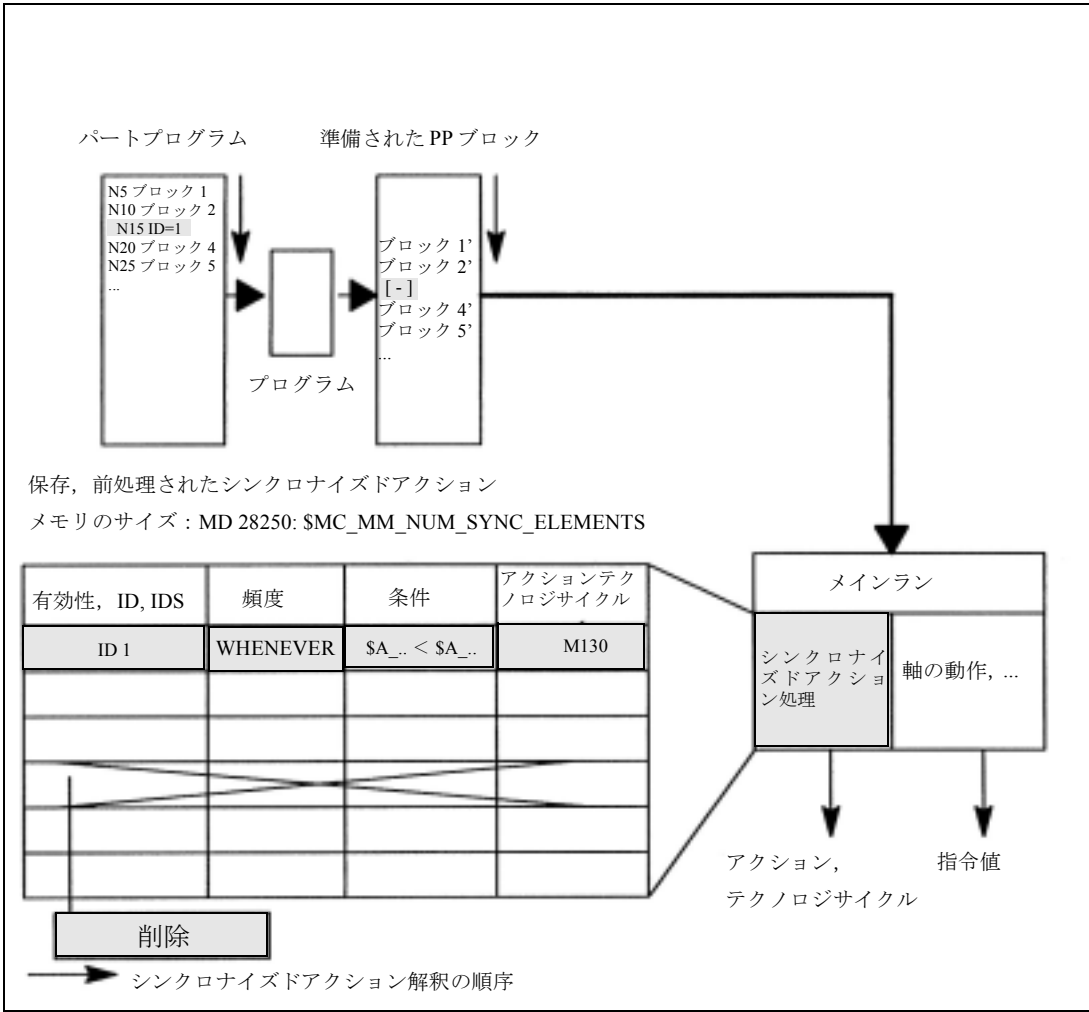


図 2.1 シンクロナイズドアクションの処理を説明する概略図

シンクロナイズドアクションの処理

起動されるアクションがシンクロナイズドアクション中に含まれているかどうかを判断するためのシンクロナイズドアクションのチェックは、補間サイクル中に実行されます。アクションの左側にプログラムされた前提条件が満足された場合、アクションはパス制御と同期して実行されます。

処理の順序

モーダル的にアクティブなシンクロナイズドアクション命令は、ID 番号の順に、1 補間サイクル内で処理されます（つまり、ID 番号 1 を持つブロックは ID 番号 2 を持つブロックより前に）。モーダルシンクロナイズドアクション命令が実行された後に、ノンモーダルアクション命令がプログラムされた順に処理されます。

2.1.1 モーションシンクロナスアクションの定義

プログラムの定義

モーションシンクロナスアクションを次の方法で定義することができます。

- パートプログラムで
- スタティックシンクロナイズドアクションは、PLC によって起動される非同期サブプログラム (ASUP) で

2.1.2 シンクロナイズドアクションの実行

実行のための条件

モーションシンクロナスアクションの中にプログラムされたアクションは、次の場合に実行されます。

- シンクロナイズドアクションが存在していて、CANCEL(ID) によって選択解除されていない場合（2.5.1 「シンクロナイズドアクション，テクノロジサイクル，パートプログラム（および PLC）の協調」を参照），
- シンクロナイズドアクションがディスエーブルされていない場合。つまり，LOCK(ID) がない場合（2.5.1 「シンクロナイズドアクション，テクノロジサイクル，パートプログラム（および PLC）の協調」を参照），
- プログラムされた頻度がキャブラリワードの結果，アクションの評価が行われる場合，あるいは，
- 適切な条件が満足された場合。

詳細については，以下のサブセクションを参照してください。

2.1.3 可能なアクションのリスト

- 補助機能 M, S, および H の PLC への出力
- 次の機能性を得るために、リアルタイム変数を設定する（書込む）ことができます。
 - オーバレイドモーション (\$AA_OFF), オブション。
 - 送り制御 (\$AC_OVR, \$AA_OVR),
プログラムされた軸モーションのディスエーブル
 - ...
- SW カム位置および時間（セッティングデータ）の変更, および他のセッティングデータの変更
- FCTDEF からの係数およびリミットの変更
- SYNFACT（多項式評価）
- FTOC（オンラインツールオフセット）
- RDISABLE（読み込みディスエーブル）
- STOPREOF（前処理停止キャンセル）
- DELDTG（移動距離削除）
- カーブテーブル値の計算
- シンクロナイズドアクションからの軸送り
- 軸フレーム
- シンクロナイズドアクションからの軸の動作／位置決め
- シンクロナイズドアクションからの主軸モーション
- シンクロナイズドアクションからの実際値設定（プリセット）
- カップリングおよびカップリングモーションの起動／停止
- シンクロナイズドアクションからの測定値
- チャンネル同期用の待機マーカの設定および削除
- アラーム／エラーリアクションの設定
- 延長停止および後退（機能 M3 の説明）
- 2.3.7「シンクロナイズドアクションに関連するシステム変数のリスト」のリストからのシステム変数の読取り, および適切にタグが付けられている場合, システム変数の書込み。

これらのアクションは 2.4「シンクロナイズドアクション中のアクション」で詳しく説明されています。

2.2 リアルタイム評価および計算

制限

リアルタイムで実行される計算は、NC 言語で実行することのできる計算のサブセットを意味しています。これはデータタイプ REAL, INT, CHAR および BOOL に制限されます。

アプリケーションの範囲

"Real-time expression" (リアルタイム式) という用語は、補間サイクル中に実行することのできるすべての計算のことを意味しています。リアルタイム式は、条件中および、NC アドレスおよび変数への割当ての中で使用されます。

リアルタイム変数

すべてのリアルタイム変数は補間サイクル中に評価され (読取られ)、アクションの絶対必要な構成要素として書込まれます。

リアルタイム変数識別子

リアルタイム変数とは、以下で始まるすべての変数のことをいいます。

\$A... (メインラン変数) あるいは
\$V... (サーボ値)。

これらの変数を明白に識別するために、\$\$ を使用して、変数をシンクロナイズドアクションの中にプログラムすることができます。

例、\$AA_IM[X] または \$\$AA_IM[Y]: 機械座標系における X 軸または Y 軸用の実際値

(注) 加工中に設定値が変化するセッティングおよびマシンデータは、
\$\$S._... / \$\$M._... を使用してプログラムされなければなりません。

データタイプ

論理演算によって同一式内に結合できるのは、データタイプが同じリアルタイム変数同士だけです。NC 言語で書かれた完全な式とは対照的に、関連するリアルタイム変数のデータタイプで計算が実行されます。

... DO \$R10 = \$AC_PARAM[0]; 許可された REAL, REAL

... DO \$R10 = \$AC_MARKER[0]; 許可されていない REAL, INT

次のリアルタイム評価の例は、すでに SW バージョン 3.2 で利用することができました (例ではこの SW バージョンのリアルタイム変数しか使用しません)

SW 3.2 用の例 1

比較の左側には、リアルタイムで計算された比較変数があり、右側には、\$\$ で始まる、許可されたリアルタイム処理変数ではない式があります。

WHEN \$AA_IM[X] > \$A_INA[1] DO M120

X 軸の実際値がアナログ入力 1 で適用された値を超えた場合、次のブロックにプログラムされたモーションの実行中に、M120 が出力されます。

このプログラミングを使用すると、アナログ入力での値が解釈の瞬間に生成されるかぎり、補間サイクルごとに実際値が再評価されます。

SW 3.2 用の例 2

比較の左側には、リアルタイムで計算された比較変数があり、右側には、(\$\$ で始まる) シンクロナイズドアクション用に許可された変数のうちの 1 つである式があります。

WHEN \$AA_IM[X] > \$\$A_INA[1] DO M120

\$\$ 変数が右側にプログラムされているので、IPO サイクル中に X 軸の現在の実値値がアナログ入力 1 と比較されます。

両方の変数が補間サイクル中に互いに比較されます。

SW 3.2 用の例 3

\$\$ 変数を比較の左側にプログラムすることもできます。

WHEN \$\$AA_IM[X] > \$\$A_INA[1] DO M120

例 2 と同様。左側と右側が常にリアルタイムで比較されます。

SW バージョン 4 での拡張

シンクロナイズドアクションで利用できるリアルタイム変数のリストが 2.3.7「シンクロナイズドアクションに関連するシステム変数のリスト」にあります。以降のソフトウェアバージョンで追加された新しいシステム変数がテーブルに示されています。

- ・ マシンおよびセッティングデータ

マシンおよびセッティングデータの場合、\$\$\$... あるいは \$\$M... がオンラインアクセス用にプログラムされなければなりません。解釈／デコーディング中に評価されるアクセス命令は、\$ 記号で始まらなければなりません。シンクロナイズドアクションから合法的にアクセスされるリアルタイム変数は、\$ 記号を 1 つだけ先頭に付けて指定されます。

基本算術演算

REAL および INT タイプのリアルタイム変数を、以下の算術演算によって論理的に結合することができます。

- 加算
- 減算
- 乗算
- 除算
- 整数除算
- モジュロ除算

これらの演算によって結合できるのは同じタイプの変数だけです。

式

基本算術演算からの式を括弧に入れたり、ネスティングしたりすることができます。

次ページの演算子の優先順位を参照してください。

比較

次の比較演算子を使用できます。

==	等しい
<>	等しくない
<	より小さい
>	より大きい
<=	以下
>=	以上

ブール演算子

次のブール演算子を使用できます。

NOT	
AND	,
OR	,
XOR	排他的 OR

ビット演算子

次のビット演算子を使用できます。

B_OR	ビットシリアル OR
B_AND	ビットシリアル AND
B_XOR	ビットシリアル排他的 OR
B_NOT	ビットシリアル否定

オペランドは、INT タイプの変数および定数です。

演算子の優先順位

希望する論理結果を複数の式中で生成するためには、次の演算子の優先順位が計算および条件で守られるべきです。

1. NOT, B_NOT	否定, ビットシリアル否定
2. *, /, DIV, MOD	乗算, 除算
3. +, -	加算, 減算
4. B_AND	ビットシリアル AND
5. B_XOR	ビットシリアル排他的 OR
6. B_OR	ビットシリアル OR
7. AND	AND
8. XOR	排他的 OR
9. OR	OR
10.	未使用
11.	比較演算子
==	等しい
<>	等しくない
>	より大きい
<	より小さい
>=	以上
<=	以下

必要な場合には丸括弧が使用されるべきです。

条件用の論理演算結果は、BOOL データタイプでなければなりません。

複数の式の例：

WHEN (\$AA_IM[X] > WERT) AND (\$AA_IM[Y] > WERT1) DO ...

機能

REAL タイプのリアルタイム変数は、機能値（サイン、コサインなど）を作成するのに使用できます。

次の機能が可能です。

SIN, COS, ABS, ASIN, ACOS, TAN, ATAN2, TRUNC, ROUND, LN,
EXP, ATAN, POT, SQRT, CTAB, CTABINV

例：

... DO \$AC_PARAM[3]=COS(\$AA_IM[X])

上記の機能の実行方法の説明については、次を参照してください。

参照： /PG/, プログラミングガイド

/PGA/, プログラミングガイド 上級編

インデックス

リアルタイムフィールド変数のインデックスが、リアルタイム変数であっても構いません。

例：

```
WHEN ... DO $AC_PARAM[ $AC_MARKER[1] ] = 3
```

インデックス \$AC_MARKER[1] は各補間サイクルで評価されます。

制限事項：

- リアルタイム変数でインデックスをネスティングすることは許可されていません。
- リアルタイムで自己生成しない変数によってリアルタイムインデックスを生成することはできません。次のプログラミングを使用するとエラーが発生します。
\$AC_PARAM[1]=\$P_EP[\$AC_MARKER[0]].

2.3 シンクロナイズドアクション用の特殊リアルタイム変数

シンクロナイズドアクションの中に指定されるシステム変数の完全なリストについては、2.3.7「シンクロナイズドアクションに関連するシステム変数のリスト」を参照してください。いくつかの特殊リアルタイム変数の特性については以下で説明します。

- マーカ／カウンタ変数
 - チャンネル別マーカ
- タイマ
- シンクロナイズドアクションパラメータ
- R パラメータ
- マシンおよびセッティングデータ
- FIFO 変数（循環メモリ）

SW バージョン 4

特殊リアルタイム変数，すなわち，タイマ，R パラメータ，マシンおよびセッティングデータ，FIFO 変数は，SW バージョン 4 から利用できます。

2.3.1 マーカ／カウンタ変数

チャンネル別マーカ

変数 \$AC_MARKER[n] は，データタイプ INTEGER でマーカあるいはカウンタとしての役割を果たします。

n: マーカ番号：0-n

1 チャンネルあたりのマーカ数はマシンデータ

MD 28256: NUM_AC_MARKER を介してセットされます。

マーカは同じ名前で各チャンネルに 1 度だけ存在します。

これらのマーカはダイナミックストレージに保存され，電源オン，および NC リセット，プログラムエンド時に 0 にリセットされて，すべてのプログラム実行で同一のスタート条件を保証します。

マーカ変数をシンクロナイズドアクションの中に読書きできます。

2.3.2 タイマ

システム変数 \$AC_TIMER[n] を使用すると、定義された遅延時間の後に、アクションをスタートさせることができます。

n: タイマ変数の番号

単位 : 秒

データタイプ : REAL

利用できるタイマ変数の数は、マシンデータ

MD 28258: MM_NUM_AC_TIMER にプログラムされます。

タイマの設定

タイマ変数のインクリメントは、値の割当てによってスタートされます :

\$AC_TIMER[n]= 値

n: タイマ変数の番号

値 : スタート値 (通常は 0)

タイマの停止

タイマ変数のインクリメントは、負の値の割当てによって停止されます :

\$AC_TIMER[n]=-1

タイマの読取り

タイマ変数が実行中であろうと停止中であろうと、現在の時間値を読取ることができます。-1 の割当てによってタイマ変数が停止された後も、現在の時間値は保存されたままです、読取ることができます。

例

デジタル入力検出後に、アナログ出力 500ms を介した実際値の出力 :

WHEN \$A_IN[1]==1 DO \$AC_TIMER[1]=0 ; タイマをリセットし、スタート

WHEN \$AC_TIMER[1]>=0.5 DO \$A_OUTA[3]=\$AA_IM[X] \$AC_TIMER[1]=-1

2.3.3 シンクロナイズドアクションパラメータ

\$AC_PARAM[n] 変数は、シンクロナイズドアクションの中でバッファとしての役割を果たします。

データタイプ : REAL

n: パラメータ番号 0 - n

1 つのチャンネルで利用できる AC パラメータ変数の数は、マシンデータ

MD 28254: \$SMC_MM_NUM_AC_PARAM を介してプログラムされます。

これらのパラメータは同じ名前でも各チャンネルに 1 度だけ存在します。

\$AC_PARAM 変数はダイナミックストレージに保存され、電源オン、および NC リセット、プログラムエンド時に 0 にリセットされて、すべてのパートプログラム実行で同一のスタート条件を保証します。\$AC_PARAM 変数をシンクロナイズドアクションの中に読書きすることができます。

2.3.4 R パラメータ

定義

R パラメータは、バッテリーでバックアップされたメモリに保存される、リアルタイムの変数です。

このため、R パラメータは、プログラムエンドおよび RESET、電源オン後もその設定値を保持します。

シンクロナイズドアクションでのアプリケーション

R パラメータの前に \$ 記号をプログラムすることにより、R パラメータをシンクロナイズドアクションの中で使用することもできます。

例：

```
WHEN $AC_MEA== 1 DO $R10= $AA_MM[Y]
```

；有効な測定が利用できれば、測定値を R パラメータに転送する

(注) 一定の R 変数の適用は、標準的にパートプログラムに行うか、あるいは、シンクロナイズドアクションに行うのが望ましいと言えます。シンクロナイズドアクションで使用された R 変数を、後になってパートプログラムに「標準的に」適用しなければならない場合、同期を保証するために、STOPRE 命令がプログラムされなければなりません。

例：

```
WHEN $A_IN[1] == 1 DO $R10 = $AA_IM[Y]
G1 X100 F150
STOPRE
IF R10 > 50 .... ; R パラメータの評価
```

2.3.5 マシンおよびセッティングデータ

SW バージョン 4 以降では、シンクロナイズドアクションからマシンおよびセッティングデータを読み書きすることが可能です。アクセスは、次の基準にしたがってプログラムされなければなりません。

- 加工中に変化しない MD, SD, および
- 加工中に設定値が変化する MD, SD。

変化しない MD, SD の読取り

設定値が変化しないマシンおよびセッティングデータは、通常のパートプログラム指令の場合と同じように、シンクロナイズドアクションから指定されます。これらのマシンおよびセッティングデータは \$ 記号で始まります。

例：

```
ID=2 WHENEVER $AA_IM[z]<$SA_OSCILL_REVERSE_POS2[Z]-6 DO
$AA_OVR[X]=0
```

；（オペレーション中はスタティックなままであると想定される）逆転レンジ 2 が揺動用に指定されます。

逆転レンジ内のイン送り揺動の完全な例については、6.2「シンクロナイズドアクションからの SD/MD の読取りおよび書込み」および次を参照してください。

参照 :/FB/, P5, 揺動機能

変化する MD, SD の読取り

加工中に値が変化するマシンデータおよびセッティングデータは，\$\$ 記号を先頭に付けて，シンクロナイズドアクションから指定されます。

例：

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

この状況では，逆転位置がオペレータアクションによっていつでも変更可能であるということが想定されます。

MD, SD の書込み

前提条件：

現在設定されているアクセス許可レベルが書込みアクセスを許可していなければなりません。変更が直ちに有効とならないかぎり，シンクロナイズドアクションから MD および SD を変更することに意味はありません。変更の有効性は，次の参照ですべての MD およびセッティングデータについて個別に説明されています。

参照： /LIS/, リスト

指定：

変更されるマシンおよびセッティングデータは，\$\$ 記号を先頭に付けて指定されなければなりません。

例：

```
ID=1 WHEN $AA_IW[X]>10 DO $$SN_SW_CAM_PLUS_POS_TAB_1[0]= 20
                                $$SN_SW_CAM_MINUS_POS_TAB_1[0]= 30
```

; SW カムの切換え位置の変更

2.3.6 FIFO 変数（循環メモリ）

アプリケーション

関連データ順序の保存を可能にするために，最高 10 個の FIFO 変数が提供されます：

\$AC_FIFO1[n] から \$AC_FIFO10[n]。

構成

図 2-3 は，FIFO 変数のメモリ構成を示しています。

数

利用できる AC FIFO 変数の数は，マシンデータ

MD 28260: NUM_AC_FIFO にプログラムされます。

サイズ

1 つの FIFO 変数に保存できる値の数は，マシンデータ

MD 28264: LEN_AC_FIFO を介して定義されます。

すべての FIFO 変数の長さは同じです。

データタイプ

FIFO 変数中の値は、REAL データタイプです。

インデックスの意味

インデックス n :

インデックス 0 ～ 5 には特殊な意味があります：

$n=0$: 変数がインデックス 0 を使用して書込まれると、新しい値が FIFO に保存されます。変数がインデックス 0 を使用して読取られると、最も古い要素が FIFO から読取られ、削除されます。

$n=1$: 最も古い保存要素へのアクセス

$n=2$: 最新の保存要素へのアクセス

$n=3$: 全 FIFO 要素の合計

MD 28266: MODE_AC_FIFO によって合計モードが決定されます：

ビット 0 = 1 新しく要素が保存されるたびに合計を更新する

ビット 0 = 0 合計なし

$n=4$: FIFO で利用できる要素の数

FIFO 中のすべての要素に読取りおよび書込みアクセスすることができます。

たとえば、第 1 FIFO 変数用の要素数をリセットすることによって、FIFO 変数がリセットされます。 $\$AC_FIFO1[4]=0$

$n=5$ FIFO の始めを基準にした現在の書込みインデックス

$n=6 \sim 6+n_{\max}$:

n 番目の FIFO 要素へのアクセス：

(注) FIFO アクセスは、R パラメータアクセスの特殊な形です（以下を参照）。

FIFO 値は、R パラメータエリア、つまり、スタティックストレージエリアに保存されます。

FIFO 値は、プログラムエンド、リセット、あるいは電源オンによって削除されません。R パラメータがアーカイブに入れられると、FIFO 値は同時に保存されます。

マシンデータ

MD 28262: START_AC_FIFO は、

R パラメータエリアへの FIFO 変数の保存の開始をマークする、R パラメータの番号を定義します。

1 チャンネル中の現在の R パラメータ数がマシンデータ

MD 28050: MM_NUM_R_PARAM にプログラムされます。

以下の2つの図では、FIFO 変数に保存されたベルト上のパーツの長さを概略的に説明しています。

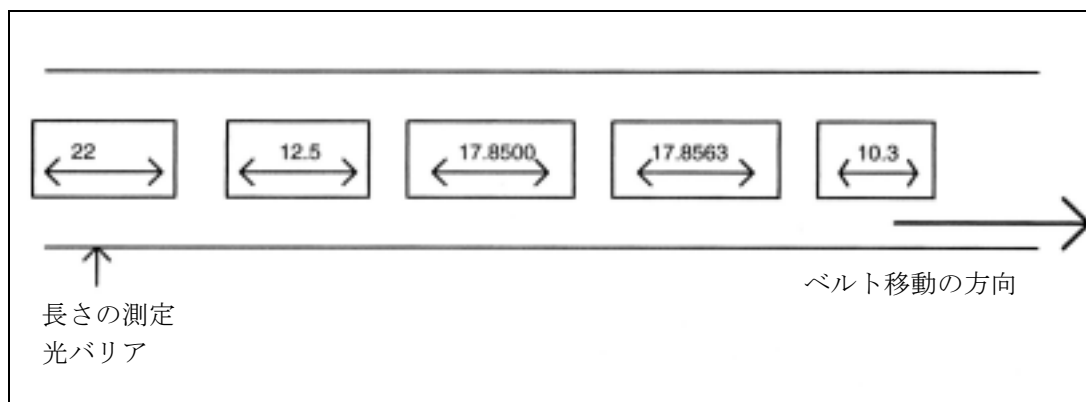


図 2.2 コンベヤベルト上の各パーツの長さ

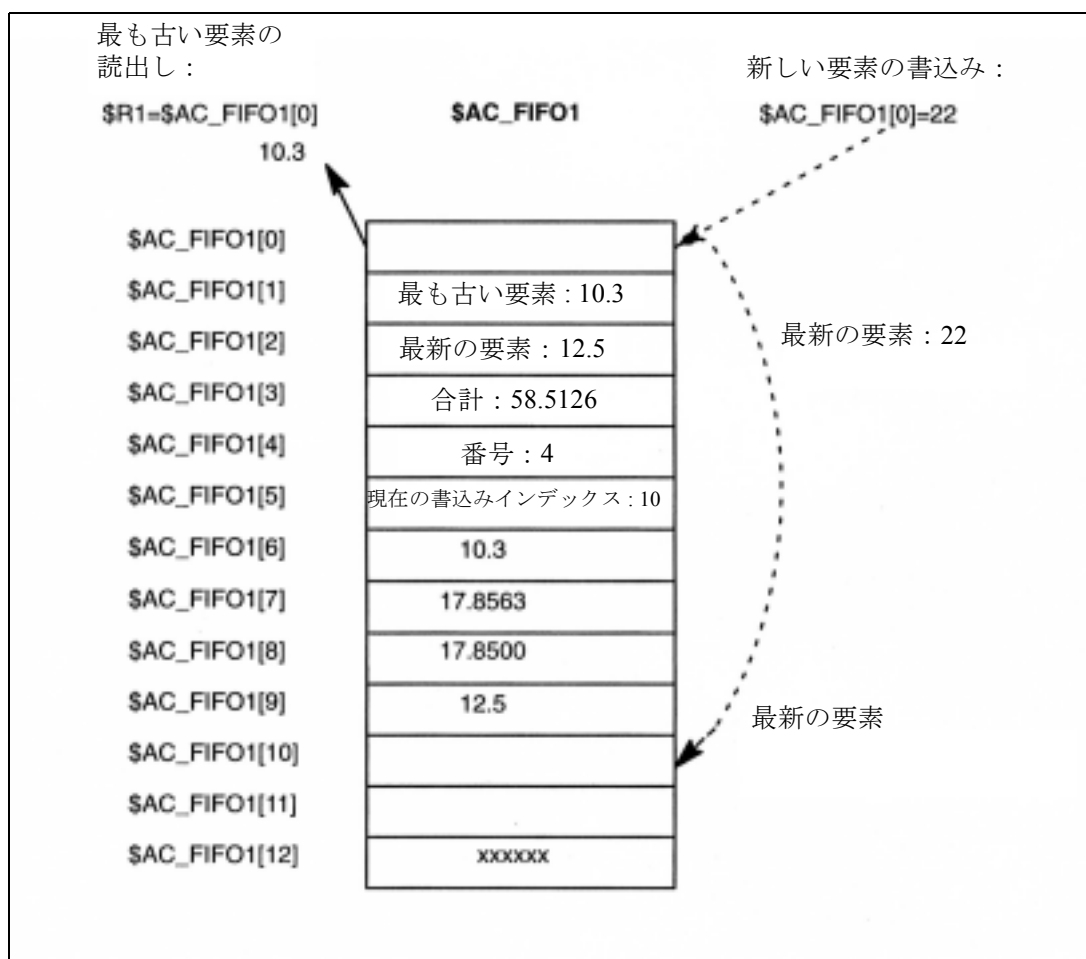


図 2.3 FIFO 変数の例

2.3.7 シンクロナイズドアクションに関連するシステム変数のリスト

概要

下表は、シンクロナイズドアクションから読取りまたは書込みアクセス可能な（カテゴリに基づいて分類された）すべてのシステム変数のリストを示しています。

アクセスオプションが示されています。

記号の説明：

r	読取り
w	書込み
R	暗示的なプリプロセッサ停止を伴う読取り
W	暗示的なプリプロセッサ停止を伴う書込み
PP	パートプログラム
SA	シンクロナイズドアクション
SW	SW バージョンについては（注）を参照してください。

（注）タイプは、システム変数が導入されたソフトウェアバージョン（たとえば、/4）を示しています。ただし、そのシステム変数が SW 2 から存在していない場合に限りです。

SA アクセスは、シンクロナイズドアクションからのシステム変数へのアクセス権が導入されたソフトウェアバージョンを示しています。ただし、アクセス権がシステム変数の導入時から利用できなかった場合に限りです。

シンクロナイズドアクション用のシステム変数の名前の構成要素 "ACT"（たとえば、\$AA_VACTM）は、補間器で計算され、軸制御用の入力変数として使用される指令値を示しています。名前のプレフィックス "\$VA_..." は、エンコード情報を評価することによって再生成される、マシン軸の真の実際値を示しています。

ユーザ変数

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$R[n], Rn	DOUBLE	スタティックメモリ中の算術変数。R パラメータは、\$R または \$R[i] を使用してシンクロナイズドアクションから指定される。そうでない場合は、...Rn または R[n] が使用される。	カウンタ	r/w	r/w /4
\$AC_PARAM[n]	DOUBLE	モーションシンクロナイズドアクション用の算術変数	カウンタ	R/W	r/w
\$AC_MARKER[n]	INT	モーションシンクロナイズドアクション用のマーカ変数	カウンタ	R/W	r/w

2.3.7 シンクロナイズドアクションに関連するシステム変数のリスト

名称	タイプ／SW	説明／値	インデックス	PP アクセス	SA アクセス／SW
\$AC_FIFOi[n]	DOUBLE/4	i: 1 ～ 10, FIFO 変数の番号 n: パラメータ番号, 0 ～最大 FIFO 要素。 n の意味： n=0: 変数がインデックス 0 を使用して書込まれると, 新しい値が FIFO に保存される。変数がインデックス 0 を使用して読取られると, 最も古い要素が FIFO から読取られ, 削除される。 n=1: 最も古い要素への読取りアクセス n=2: 最新の要素への読取りアクセス n=3: ビット 0 が MD\$MC_MM_MODE_FIFO でイネーブルされた場合, FIFO に保存されているすべての要素の合計。 n=4: 現在の FIFO 要素数への読取りアクセス n= 5 ～ m: 個々の FIFO 要素への読取りアクセス。5 は最も古い要素であり, 6 は 2 番目に古い要素である。	パラメータ番号	R/W	r/w

入力／出力

名称	タイプ／SW	説明／値	インデックス	PP アクセス	SA アクセス／SW
\$A_IN[n]	BOOL	NC のデジタル入力	入力番号	R	r
\$A_OUT[n]	BOOL	NC のデジタル出力	出力番号	R/w	r/w
\$A_INA[n]	DOUBLE	NC のアナログ入力	入力番号	R	r
\$A_OUTA[n]	DOUBLE	NC のアナログ出力 書込みオペレーション後, 次の補間サイクルまで値はアクティブにならない; その後, 値を読返すことができる。	出力番号	R/w	r/w
\$A_INCO[n]	BOOL	比較器入力	入力番号	R	r
\$A_PBB_IN[n]	INT /5	データバイト (8 ビット) を PLC I/O から直接読取る	PLC 入力エリアでの位置オフセット	R/	r/
\$A_PBB_OUT[n]	INT /5	データバイト (8 ビット) を PLC I/O に直接書込む	PLC 出力エリアでの位置オフセット	R/w	r/w
\$A_PBD_IN[n]	INT /5	データダブルワード (32 ビット) を PLC I/O から直接読取る	PLC 入力エリアでの位置オフセット	R/	r/

名称	タイプ／ SW	説明／値	インデックス	PP アクセス	SA アクセス ／ SW
\$A_PBD_OUT[n]	INT/5	データダブルワード（32 ビット）を PLC I/O に直接書込む	PLC 出力エリアでの位置オフセット	R/w	r/w
\$A_PBR_OUT[n]	DOUBLE /5	実数データ（32 ビット）を PLC I/O に直接書込む	PLC 出力エリアでの位置オフセット	R/w	r/w
\$A_PBW_IN[n]	INT /5	データワード（16 ビット）を PLC I/O から直接読取る	PLC 入力エリアでの位置オフセット	R/	r/
\$A_PBW_OUT[n]	INT /5	データワード（16 ビット）を PLC I/O に直接書込む	PLC 出力エリアでの位置オフセット	R/w	r/w

タイマ

名称	タイプ／ SW	説明／値	インデックス	PP アクセス	SA アクセス ／ SW
\$A_YEAR	INT	システム時間，年		R	r /4
\$A_MONTH	INT	システム時間，月		R	r /4
\$A_DAY	INT	システム時間，日		R	r /4
\$A_HOUR	INT	システム時間，時		R	r /4
\$A_MINUTE	INT	システム時間，分		R	r /4
\$A_SECOND	INT	システム時間，秒		R	r /4
\$A_MSECOND	INT	システム時間，ミリ秒		R	r /4
\$AC_TIMER[n]	DOUBLE /4	タイマ単位，秒。時間は補間サイクルの整数倍で内部的にカウントされる。タイマ変数のインクリメントは，値の割当て \$AC_TIMER[n]=スタート値> によってスタートされる。タイマ変数のインクリメントは，負の値：\$AC_TIMER[n]=-1 の割当てによって停止される。タイマ変数が実行中であろうと，停止中であろうと，現在の時間値を読取ることができる。-1 の割当てによってタイマ変数が停止された後でも，現在の時間値は保存されたままで，読取ることができる。	カウンタ	R/W	r/w
\$AC_TIME	DOUBLE	秒で表した，ブロックスタートからの時間		R	r /4
\$AC_TIMEC	DOUBLE	補間サイクルで表した，ブロックスタートからの時間		R	r /4

チャンネルステータス

名称	タイプ / SW	説明 / 値	インデックス	PPアクセス	SA アクセス / SW
\$AC_MEA[n]	INT	TRUE (1) の場合、プローブがトリガされる	プローブ番号	r	r /4
\$A_PROBE[n]	INT /4	\$A_PROBE[1]: 第 1 プローブのステータス \$A_PROBE[2]: 第 2 プローブのステータス 0: たわんでいない 1: たわんでいる	プローブ番号	R	r
\$AC_TRAFO	INT	マシンデータ \$MC_TRAFO_TYPE n を基準にした、アクティブな変換のコード番号	-	R	r /4
\$AC_LIFTFAST	INT	急速上昇: 0: 逆ストロークがアクティブでなかった 1: 逆ストロークがアクティブであった	-	R/W	r/w
\$AC_ALARM_STAT	INT/5	シンクロナイズドアクション (SYNFCT) 用の (選択された) アラームリアクション	-	R/	r/
\$AC_STAT	INT/4	チャンネルステータス -1: 無効 0: チャンネルリセット 1: チャンネル中断 2: チャンネルアクティブ		r	r /4
\$AC_PROG	INT/4	プログラムステータス -1: 無効 0: プログラムリセット 1: プログラム停止 2: プログラムアクティブ 3: プログラム待機 4: プログラム中断		r	r /4
\$AC_IPO_BUF	INT/4	補間バッファのレベルを満たす		r	r /4
\$AC_SYNA_MEM	INT/4	モーションシンクロナイズドアクション用のフリーメモリは、\$MC_MM_NUM_SYNC_ELEMENTS で割当てられたメモリの要素がどれだけ現在もフリーであるかを示す。		r	r /4

ASUP

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$AC_ASUP	INT /4	<p>参照 : /FB/, K1, (モードグループ, チャンネル, プログラム運転モード)</p> <p>注釈 :</p> <p>ASUP: 起動の理由</p> <p>Act. cau.: 起動の原因</p> <p>Act. by: 起動源</p> <p>Cont.: 再開のためのオプション</p> <p>BIT 0:</p> <p>Act. cau.: ユーザ割込み "ASUP with Blsync" (Blsync を使用した ASUP) (ブロック同期),</p> <p>Act. by: Vdi 信号, デジタル/アナログインタフェース,</p> <p>Cont. : Reorg または Ret</p> <p>BIT 1:</p> <p>Act. cau.: ユーザ割込み "ASUP" (停止が生じたブロックの後の位置が, Repos を使用したプログラム再開のために保存される)</p> <p>Act. by: Vdi 信号, デジタル/アナログインタフェース</p> <p>Cont.: Reorg または Ret</p> <p>BIT 2:</p> <p>Act. cau.: ユーザ割込み "ASUP from Ready channel status" (チャンネルステータスレディからの ASUP) ,</p> <p>Act. by: Vdi 信号, デジタル/アナログインタフェース</p> <p>Cont.: Reorg または Ret</p> <p>BIT 3:</p> <p>Act. cau.: ユーザ割込み "ASUP in a manual mode and channel status not READY" (手動モードおよびチャンネルステータス READY でない, での ASUP)</p> <p>Act. by: Vdi 信号, デジタル/アナログインタフェース</p> <p>Cont. by: Reorg または Ret</p> <p>BIT 4:</p> <p>Act. cau.: ユーザ割込み "ASUP". (Repos を使用したプログラム再開のために, 割込み時の位置が保存される)。</p> <p>Act. by: Vdi 信号, デジタル/アナログインタフェース</p> <p>Cont. by: Reorg または Ret</p> <p>BIT 5:</p> <p>Act. cau.: サブプログラム反復のキャンセル</p> <p>Act. by: Vdi 信号</p> <p>Cont. by: REPOS システム ASUP の使用</p> <p>BIT 6:</p> <p>Act. cau.: デコーディング単一ブロックの起動</p> <p>Act. by: Vdi 信号 (+BTSS)</p> <p>Cont. by: REPOS システム ASUP の使用</p> <p>BIT 7:</p> <p>Act. cau.: 移動距離削除の起動</p> <p>Act. by: Vdi 信号</p> <p>Cont. by: Ret システム ASUP の使用</p> <p>BIT 8:</p> <p>Act. cau.: 軸同期の起動</p> <p>Act. by: Vdi 信号 Cont. by: REPOS システム ASUP の使用</p>		R	r /4

2.3.7 シンクロナイズドアクションに関連するシステム変数のリスト

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$AC_ASUP	INT/4	<p>BIT 9: Act. cau.: モード切換え Act. by: Vdi 信号 Cont. by: REPOS または RET システム ASUP の使用 (MD を参照)</p> <p>BIT 10: Act. cau.: 教示のもとでの, あるいは教示停止後のプログラム再開 Act. by: Vdi 信号 Cont. by: Ret システム ASUP の使用</p> <p>BIT 11: Act. cau.: オーバストア選択 Act. by: PI サービス Cont. by: REPOS システム ASUP の使用</p> <p>BIT 12: Act. cau.: Repos 付きリアクション補正ブロック付きのアラーム Act. by: 内部 Cont. by: REPOS システム ASUP の使用</p> <p>BIT 13: Act. cau.: G33 を使用した後退動作, および停止 Act. by: 内部 Cont. by: Ret システム ASUP の使用</p> <p>BIT 14: Act. cau.: ドライラン送り速度の起動 ; Act. by: Vdi 信号 Cont. by: REPOS システム ASUP の使用</p> <p>BIT 15: Act. cau.: ドライラン送り速度の停止 ; Act. by: Vdi 信号 Cont. by: REPOS システム ASUP の使用</p> <p>BIT 16: Act. cau.: ブロックスキップの起動 ; Act. by: Vdi 信号 Cont. by: REPOS システム ASUP の使用</p> <p>BIT 17: Act. cau.: ブロックスキップの停止 ; Act. by: Vdi 信号 Cont. by: REPOS システム ASUP の使用</p> <p>BIT 18: Act. cau.: マシンデータの起動 Act. by: PI サービス Cont. by: REPOS システム ASUP の使用</p> <p>Bit 19: Act. cau.: ユーザデータの起動 Act. by: PI サービス Cont. by: REPOS システム ASUP の使用</p>		R	r /4

パス

名称	タイプ／SW	説明／値	インデックス	PP アクセス	SA アクセス ／ SW
\$AC_PATHN	DOUBLE	正規化されたパスパラメータ，値は， 0＝ブロックの始め，および 1＝ブロックエンド		R	r
\$AC_DTBW	DOUBLE	ワーク座標系におけるブロックの始めからのジオメトリ距離。プログラムされた位置だけが距離計算を決定する。軸がカップリングモーション軸である場合，軸カップリングの結果生じた位置構成要素は無視される。		R	r
\$AC_DTBB	DOUBLE	基準座標系におけるブロックの始めからのジオメトリ距離。プログラムされた位置だけが距離計算を決定する。軸がカップリングモーション軸である場合，軸カップリングの結果生じた位置構成要素は無視される。		R	r
\$AC_DTEW	DOUBLE	ワーク座標系におけるブロックエンドからのジオメトリ距離。プログラムされた位置だけが距離計算を決定する。軸がカップリングモーション軸である場合，軸カップリングの結果生じた位置構成要素は無視される。		R	r
\$AC_DTEB	DOUBLE /3	基準座標系におけるブロックエンドからのジオメトリ距離。プログラムされた位置だけが距離計算を決定する。軸がカップリングモーション軸である場合，軸カップリングの結果生じた位置構成要素は無視される。		R	r
\$AC_PLTBB	DOUBLE /3	基準座標系におけるブロックの始めからのパス距離		R	r
\$AC_PLTEB	DOUBLE	基準座標系におけるブロックエンドまでのパス距離		R	r
\$AC_DELT	DOUBLE	モーションシンクロナイズドアクション用に移動距離を削除した後の，ワーク座標系におけるパス移動距離		r	r

速度, チャンネル別

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$AC_OVR	DOUBLE	シンクロナイズドアクション用のパスオーバーライド：乗算オーバーライド構成要素が、オペレータオーバーライドおよびプログラムされたオーバーライド、変換オーバーライドのほかに動作する。合計係数は200%に制限される。これは補間サイクルごとに再度書込まなければならない。そうでない場合は、値100%が適用される。\$AA_OVR[S1]により主軸オーバーライドが変更される。次のマシンデータによって定義されるオーバーライドは超過されない。 MD 12100: OVR_FACTOR_LIMIT_BIN, MD 12030: OVR_FACTOR_FEEDRATE[30], MD 12010: OVR_FACTOR_AX_SPEED[30]		r/w	r/w /4
\$AC_VC	DOUBLE	シンクロナイズドアクション用の加算パス送り速度オーバーライド。補正值は、G0 または、G33, G331, G332, G63 の場合はアクティブにならない。補正值は、補間サイクルごとに再度書込まなければならない。そうでない場合は、値0が適用される。0 オーバーライドによって補正值がキャンセルされる；そうでない場合は、オーバーライドは補正值に影響を及ぼさない。補正值の結果として、合計送り速度は負になることができない。最高値は、最大軸速度および加速が超過されないように制限される。他の送り速度構成要素の計算は、\$AC_VCの影響を受けない。次のマシンデータによって定義されたオーバーライド値は、超過されない。 MD 12100: OVR_FACTOR_LIMIT_BIN, MD 12030: OVR_FACTOR_FEEDRATE[30], MD 12010: OVR_FACTOR_AX_SPEED[30], MD 12070: OVR_FACTOR_SPIND_SPEED 加算送り速度オーバーライドは、結果として生じる送り速度が、プログラムされた送り速度の最大オーバーライド値を超過しないように制限される。		r/w	r/w /4
\$AC_VACTW	DOUBLE	ワーク座標系におけるパス速度		R	r
\$AC_VACTB	DOUBLE	基準座標系におけるパス速度		R	r

速度，軸別

名称	タイプ／SW	説明／値	インデックス	PPアクセス	SA アクセス ／ SW
\$AA_OVR[X]	DOUBLE	モーションシンクロナイズドアクション用の軸オーバーライド。乗算オーバーライド構成要素が、オペレータオーバーライドおよびプログラムされたオーバーライド、変換オーバーライドのほかに動作する。合計係数は200%に制限される。これは補間サイクルごとに再度書込まなければならない。そうでない場合は、値 100% が適用される。 \$AA_OVR[S1] によって主軸オーバーライドが変更される。次のマシンデータによって定義されたオーバーライドは超過されない。 MD 12100: OVR_FACTOR_LIMIT_BIN, MD 12030: OVR_FACTOR_FEEDRATE[30], MD 12010: OVR_FACTOR_AX_SPEED[30], \$AA_OVR_FACTOR_SPIND_SPEED	軸	r/w	r/w /4
\$AA_VX[X]	DOUBLE	モーションシンクロナイズドアクション用の加算軸送り速度オーバーライド。補正値は補間サイクルごとに再度書込まなければならない。そうでない場合は、値 0 が適用される。0 オーバライドにより補正値がキャンセルされる；そうでない場合は、オーバーライドは補正値に影響を及ぼさない。補正値の結果として、合計送り速度は負になることができない。最高値は、最大軸速度および加速が超過されないように制限される。他の送り速度構成要素の計算は、\$AA_VC の影響を受けない。次のマシンデータによって定義されたオーバーライド値は超過されない。 MD 12100: OVR_FACTOR_LIMIT_BIN, MD 12030: OVR_FACTOR_FEEDRATE[30], MD 12010: OVR_FACTOR_AX_SPEED[30], MD 12070: OVR_FACTOR_SPIND_SPEED 加算送り速度オーバーライドは、結果として生じる送り速度が、プログラムされた送り速度の最大オーバーライド値を超過しないように制限される。	軸	r/w	r/w /4
\$AA_VACTW[X]	DOUBLE	ワーク座標系における軸速度	軸	R	r
\$AA_VACTB[X]	DOUBLE	基準座標系における軸速度	軸	R	r
\$AA_VACTM[X]	DOUBLE	機械座標系における軸速度指令値。この指令値は、交換軸および PLC 軸によっても読取ることができる。	軸	R	r
\$VA_VACTM[X]	DOUBLE /4	機械座標系における軸速度実際値。エンコーダリミット周波数が超過された場合、この変数は未定義の値を返す。	軸	R	r

シンクロナイズドアクション用の多項式の値

名称	タイプ／SW	説明／値	インデックス	PP アクセス	SA アクセス／SW
\$SAC_FCTiLL, \$SAC_FCTLL[j]	DOUBLE /4	シンクロナイズドアクション用の多項式の下限 (SYNFCT) i: 1 ～ 3, 評価機能 FCTDEF 1 ～ 3 j: 多項式番号		R/W	r/w
\$SAC_FCTiUL, \$SAC_FCTUL[j]	DOUBLE /4	シンクロナイズドアクション用の多項式の上限 (SYNFCT) i: 1 ～ 3, 評価機能 FCTDEF 1 ～ 3 j: 多項式番号		R/W	r/w
\$SAC_FCTiC[n], \$SAC_FCT0[n]	DOUBLE /(4)	i: 1 ～ 3, 多項式 1 ～ 3 ; 係数 n : 0 ～ 3 多項式 n 用の a_0 係数		R/W	r/w
\$SAC_FCT1[n]	DOUBLE /(4)	多項式 n 用の a_1 係数		R/W	r/w
\$SAC_FCT2[n]	DOUBLE /(4)	多項式 n 用の a_2 係数		R/W	r/w
\$SAC_FCT3[n]	DOUBLE /(4)	多項式 n 用の a_3 係数		R/W	r/w

位置およびパス

名称	タイプ／SW	説明／値	インデックス	PP アクセス	SA アクセス／SW
\$AA_IW[X]	DOUBLE	ワーク座標系 (WCS) における実際値	軸	R	r
\$AA_IEN[X]	DOUBLE	ワークゼロ座標系 (WOS) における実際値	軸	R	r
\$AA_IB[X]	DOUBLE	基準座標系 (BCS) における実際値	軸	R	r
\$AA_IBO[X]	DOUBLE	基準ゼロ座標系における実際値	軸	R	r
\$AA_IM[X]	DOUBLE	機械座標系 (MCS) における実際値	軸	R	r
\$AA_ENC_ACTIVE[X]	BOOL /4	アクティブな測定系が、エンコーダリミット周波数 (有効な値) 以下で動作している	軸	R	r
\$AA_ENCi_ACTIVE[X]	BOOL /4	i: 1 ～ 2 エンコーダ番号 ; 測定系 i がエンコーダリミット周波数 (有効な値) 以下で動作している	軸	R	r
\$VA_IM[X]	DOUBLE /4	(アクティブな測定系によって測定された) 機械座標系におけるエンコーダ実際値。実際値補正は補正される (リードスクリュー誤差補正, バックラッシュ補正, 4 分の 1 円弧誤差補正)。	軸	R	r
\$VA_IM1[X]	DOUBLE /4	(エンコーダ 1 によって測定された) 機械座標系における実際値。補正は補正される。	軸	R	r
\$VA_IM2[X]	DOUBLE /4	(エンコーダ 2 によって測定された) 機械座標系における実際値。補正は補正される。	軸	R	r
\$AA_MW[X]	DOUBLE	ワーク座標系における測定値	軸	r	r
\$AA_MWi[X]	DOUBLE /4	ワーク座標系における測定値 i: 1 ～ 4 トリガイイベント 1 ～ 4 用	軸	r	r
\$AA_MM[X]	DOUBLE	機械座標系における測定値	軸	r	r/4

名称	タイプ／SW	説明／値	インデックス	PP アクセス	SA アクセス ／ SW
\$AA_MMi[X]	DOUBLE /4	機械座標系における測定値 i: 1 ～ 4 トリガイメント 1 ～ 4 用	軸	r	r
\$AA_MEAACT[X]	BOOL /4	軸の測定が軸 X についてアクティブ であれば、値は TRUE (1) である。	軸	r	r
\$AC_DRF[X]	DOUBLE	DRF オフセット	軸	R	r /4
\$AC_PRESET[X]	DOUBLE	最後のプリセット値	軸	R	r /4
\$AA_OFF[X]	DOUBLE	プログラムされた軸用のオーバーレイ ドモーション	軸	R/w	r/w
\$AC_RETPOINT[X]	DOUBLE	再位置決め用の、輪郭上の戻り点	軸	R	r /4
\$AA_SOFTENDP[X]	DOUBLE	ソフトウェアリミット位置、正方向	軸	R	r /4
\$AA_SOFTENDN[X]	DOUBLE	ソフトウェアリミット位置、負方向	軸	R	r /4
\$AA_DTBW[X]	DOUBLE	モーションシンクロナイズドアク ション中の位置決め軸および同期軸 用の、ワーク座標系におけるブロッ クの始めからの軸パス。プログラム された位置だけがパス計算を決定す る。軸がカップリングモーション軸 である場合、軸カップリングの結果 生じた位置構成要素は無視される。	軸	R	r
\$AA_DTBB	DOUBLE	モーションシンクロナイズドアク ション中の位置決め軸および同期軸 用の、基準座標系におけるブロッ クの始めからの軸パス。プログラム された位置だけがパス計算を決定す る。軸がカップリングモーション軸 である場合、軸カップリングの結果 生じた位置構成要素は無視される。	軸	R	r
\$AA_DTEW	DOUBLE	モーションシンクロナイズドアク ション中の位置決め軸および同期軸 用の、ワーク座標系におけるブロッ クエンドまでの軸パス。プログラム された位置だけがパス計算を決定す る。軸がカップリングモーション軸 である場合、軸カップリングの結果 生じた位置構成要素は無視される。	軸	R	r
\$AA_DTEB	DOUBLE	モーションシンクロナイズドアク ション中の位置決め軸および同期軸 用の、基準座標系におけるブロッ クエンドまでの軸パス。プログラム された位置だけがパス計算を決定す る。軸がカップリングモーション軸 である場合、軸カップリングの結果 生じた位置構成要素は無視される。	軸	R	r
\$AA_DTEPW	DOUBLE	ワーク座標系におけるイン送り揺動 用の軸の移動距離	軸	R	r
\$AA_DTEPB	DOUBLE	基準座標系におけるイン送り揺動 用の軸の移動距離	軸	R	r
\$AA_DELT	DOUBLE	モーションシンクロナイズドアク ション用に軸の移動距離を削除した 後の、ワーク座標系における軸の移 動距離	軸	r	r

2.3.7 シンクロナイズドアクションに関連するシステム変数のリスト

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$AA_OSCILL_REVERSE_POS1[X]	DOUBLE	揺動用の現在の逆転位置 1。シンクロナイズドアクションでは、セッティングデータ \$SA_OSCILL_REVERSE_POS1 の値はオンラインで評価される。	軸	R	r
\$AA_OSCILL_REVERSE_POS2[X]	DOUBLE	揺動用の現在の逆転位置 2。シンクロナイズドアクションでは、セッティングデータ \$SA_OSCILL_REVERSE_POS2 の値はオンラインで評価される。	軸	R	r

ドライブデータ

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$AA_LOAD[X]	DOUBLE	% で表したドライブ負荷 (611D 用のみ)	軸	R	r
\$AA_TORQUE[X]	DOUBLE	Nm で表したドライブトルク指令値 (611D 用のみ)	軸	R	r
\$AA_POWER[X]	DOUBLE	W で表したアクティブなドライブパワー (611D 用のみ)	軸	R	r
\$AA_CURR[X]	DOUBLE	軸あるいは主軸用の現在の実際値 (611D 用のみ)	軸	R	r
\$VA_DPE[X1]	BOOL / 5	マシン軸のパフォーマンスリリースのステータス	マシン軸	R/	r/

ドライブステータス / リンク

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$VA_EG_SYNCDIFF[a]	DOUBLE / 5	同期差	追従軸	R/	r/
\$VA_COUP_OFFSETS[SPI(2)]	DOUBLE / 3	同期主軸用の位置オフセット (実際値)	追従主軸	R	r / 4

ツールマネージメント

参照： /FBW/, ツールマネージメント

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$AC_TC_FCT	INT /5	指令番号。指令番号は、希望するオペレーションを指定する。	-	R/	r/
\$AC_TC_LFN	INT /5	新しいツールの移動元ロケーション番号。 0: 新しいツールはない。	-	R/	r/
\$AC_TC_LFO	INT /5	(交換される) 古いツールの移動元ロケーション番号。 0: 古いツールはない。	-	R/	r/
\$AC_TC_LTN	INT /5	新しいツールの移動先ロケーション番号。 0: 新しいツールはない。	-	R/	r/
\$AC_TC_LTO	INT /5	(交換される) 古いツールの移動元ロケーション番号。 0: 古いツールはない。	-	R/	r/
\$AC_TC_MFN	INT /5	新しいツールの移動元マガジン番号。0: 新しいツールはない。	-	R/	r/
\$AC_TC_MFO	INT /5	(交換される) 古いツールの移動元マガジン番号。 0: 古いツールはない。	-	R/	r/
\$AC_TC_MTN	INT /5	新しいツールの移動先マガジン番号。0: 新しいツールはない。	-	R/	r/
\$AC_TC_MTO	INT /5	(交換される) 古いツールの移動先マガジン番号。 0: 古いツールはない。 -	-	R/	r/
\$AC_TC_STATUS	INT /5	\$AC_TC_FCT を使用して読取られる指令のステータス	-	R/	r/
\$AC_TC_THNO	INT /5	新しいツールがロードされるツールホルダの番号 (主軸番号)。	-	R/	r/
\$AC_TC_TNO	INT /5	(ロードされる) 新しいツールの NCK 内部の T 番号。 0: 新しいツールはない。	-	R/	r/

軸ステータス

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$AA_STAT[X]	INT /4	軸ステータス : 0: 軸ステータスは利用できない 1: 移動動作アクティブ 2: 軸が補間エンドに到達した。 チャンネルの軸のみ 3: 軸が定位置にある [イグザクトストップ (粗)], 全軸 4: 軸が定位置にある [イグザクトストップ (微)], 全軸	軸	R	r

2.3.7 シンクロナイズドアクションに関連するシステム変数のリスト

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$AA_TYP[X]	INT /4	軸タイプ : 0: 他のチャンネルにある軸 1: 同一チャンネルのチャンネル軸 2: ニュートラル軸 3: PLC 軸 4: 揺動軸 5: JOG モードで現在移動中のニュートラル軸 6: マスタ値カップリング追従軸 7: カップリングモーション追従軸 8: 指令軸 9: コンパイルサイクル軸	軸	R	r
\$AA_FXS[X]	INT	ステータス "travel to fixed stop" (固定停止点への移動) 0: 軸は停止していない 1: 定始点アプローチ, 成功 2: 固定停止点アプローチ, 失敗	軸	R	r /4
\$AA_OFF_LIMIT[X]	INT/4	軸補正 \$AA_OFF[X] 用のリミット値 0: リミット値が到達されていない 1: リミット値が到達された, 正の軸方向 -1: リミット値が到達された, 負の軸方向	軸	R	r
\$AA_LEAD_SP[MV]	DOUBLE /4	シュミレートされたマスタ値 - 位置	マスタ値	R/W	r/w
\$AA_LEAD_SV[MV]	DOUBLE /4	シュミレートされたマスタ値 - 速度	マスタ値	R/W	r/w
\$AA_LEAD_P[MV]	DOUBLE /4	現在のマスタ値 - 位置 (減少されたモジュロ) マスタ値 (MV) がモジュロ軸である場合, 次が常に真である : $0 \leq \$AA_LEAD_P[MV] \leq \$MA_MODULE_RANGE[MV]$	マスタ値	R	r
\$AA_LEAD_P_TURN[MV]	DOUBLE /4	現在のマスタ値 - モジュロ減少の結果失われた位置構成要素。(コントローラが内部計算に使用する) 実際のマスタ値位置は, $\$AA_LEAD_P[MV] + \$AA_LEAD_P_TURN[MV]$ である。MV がモジュロ軸である場合, $\$AA_LEAD_P_TURN$ は $\$MA_MODULE_RANGE$ の整数倍である。MV がモジュロ軸でない場合, $\$AA_LEAD_P_TURN$ は常に 0 である。 例 1: $\$MA_MODULE_RANGE[MV] = 360$ $\$AA_LEAD_P[MV] = 290$ $\$AA_LEAD_P_TURN[MV] = 720$ (コントローラが内部計算で使用する) 実際のマスタ値位置は 1010 である。 例 2: $\$MA_MODULE_RANGE[MV] = 360$ $\$AA_LEAD_P[MV] = 290$ $\$AA_LEAD_P_TURN[MV] = -360$ (コントローラが内部計算で使用する) 実際のマスタ値位置は -70 である。	マスタ値	R	r

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$AA_LEAD_V[MV]	DOUBLE /4	際のマスタ値 - 速度	マスタ値	R	r
\$AA_SYNC[FA]	INT /4	マスタ値カップリングにおける追従軸 のカップリングステータス 0: 同期なし 1: 粗同期 2: 微同期 3: 粗および微	追従軸	R	r
\$AA_COUP_ ACT[SPI(2)]	INT /4	追従主軸の現在のカップリングステータス	追従主軸	R	r
\$AA_ACT_INDEX_ AX_POS_NO[X]	INT /5	0: インデックス軸なし。したがって、 インデックス位置は利用できない。 > 0: 到達あるいはオーバトラベルされる 最後のインデックス位置の番号		R/	r/
\$AA_EG_BC[a]	STRING /5	ブロックチェンジ条件 TYPE_STRING 用の第 2 寸法が自動的に 最大ストリング長になる	追従軸	R/	r/
\$AA_EG_NUM_LA[a]	INT /5	EGDEF を使用して指定されたリーディング 軸の数	追従軸	R/	r/
\$AA_EG_SYNFA[a]	DOUBLE /5	追従軸の同期位置	追従軸	R/	r/
\$AA_IBN[X]	DOUBLE /5	基準ゼロ座標系 (BCS) における実際値		R/	r/
\$AA_IEN[X]	DOUBLE /5	設定可能なゼロ座標系 (SZS) における 実際値		R/	r/
\$AA_PROG_INDEX_ AX_POS_NO[X]	INT /5	0: インデックス軸なし。したがって、 インデックス位置は利用できない。 あるいは、インデックス軸が現在インデックス 位置にアプローチしていない。 > 0: プログラムされたインデックス位置の 番号		R/	r/
\$AA_REF[X]	INT /5	軸ステータス : 0: 軸は基準化されていない 1: 軸が基準化されている		R/	r/
\$AA_SCPAR	INT /5	現在のサーボパラメータセットを読み取る		R/	r/

2.3.7 シンクロナイズドアクションに関連するシステム変数のリスト

主軸データ

名称	タイプ／SW	説明／値	インデックス	PP アクセス	SA アクセス ／ SW
\$AC_MSNUM	INT	マスタ主軸の番号	R	r/4	
\$AC_SDIR[n]	INT	現在アクティブな主軸回転 3: 主軸, 時計方向 4: 主軸, 反時計方向 5: 主軸停止	主軸番号	R	r /4
\$AC_SMODE[n]	INT	現在アクティブな主軸モード: 0: 主軸は利用できない 1: 速度制御モード 2: 位置決めモード 3: 同期モード 4: 軸モード	主軸番号	R	r /4
\$AA_S[n]	DOUBLE	実際の主軸速度その符号により回転 方向が識別される	主軸番号	R	r /4
\$AA_COUP_OFFS[S2]	DOUBLE	同期主軸用の位置オフセット (指令 値)	追従主軸	R	r /4

PLC 変数の読書き

名称	タイプ／SW	説明／値	インデックス	PP アクセス	SA アクセス ／ SW
\$A_DBB[n]	INT /4	データバイト (8 ビット) の PLC か らの読取りまたは PLC への書込み	I/O エリアで のオフセット	R/w	r/w
\$A_DBW[n]	INT /4	データワード (16 ビット) の PLC か らの読取りまたは PLC への書込み	I/O エリアで のオフセット	R/w	r/w
\$A_DBD[n]	INT /4	データダブルワード (32 ビット) の PLC からの読取りまたは PLC への書 込み	I/O エリアで のオフセット	R/w	r/w
\$A_DBR[n]	DOUBLE /4	実数データ (32 ビット) の PLC から の読取りまたは PLC への書込み	I/O エリアで のオフセット	R/w	r/w

リンク変数

名称	タイプ／ SW	説明／値	インデックス	PP アクセス	SA アクセス ／ SW
\$A_GG[n]	INT /5	SA から、G グループの現在の G 機能を読取る（インデックスは PLC インタフェースと同じ）	G グループの番号	r/	r/

軸コンテナ回転

名称	タイプ／ SW	説明／値	インデックス	PP アクセス	SA アクセス ／ SW
\$AC_AXCTSWA[n]	BOOL /5	軸コンテナ回転のイネーブル 1: チャンネルが、軸コンテナ名 n について軸コンテナ回転をイネーブルし、回転が現在も進行中である。 0: 軸コンテナ回転が完了した。	軸コンテナ	r/	r/
\$AN_AXCTAS[n]	INT /5	軸コンテナ回転、現在の回転角度。 軸コンテナ名 n について、軸コンテナが前進させたスロットの数を指定する。 値の範囲は、0 ～軸コンテナ -1 中に割当てられたスロットの最大数である。	軸コンテナ	r/	r/
\$AN_AXCTSWA[n]	BOOL /5	軸コンテナ回転アクティブ 1: 軸コンテナ回転が軸コンテナ名 n 上で現在実行されている。 0: 軸コンテナ回転は現在アクティブでない。	軸コンテナ	r/	r/

安全統合 (S. I.)

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$A_INSE[n]	BOOL /4	安全統合入力信号のイメージ (外部 NCK インタフェース)	入力番号	R	r
\$A_INSEW[n]	INT /4	安全統合入力信号のイメージ (外部 NCK インタフェース)	入力ワードの番号	R	r
\$A_INSEP[n]	BOOL /4	安全入力信号のイメージ (外部 PLC インタフェース)	入力番号	R	r
\$A_INSEPW[n]	INT /4	安全統合入力信号のイメージ (外部 PLC インタフェース)	入力ワードの番号	R	r
\$A_OUTSEW[n]	INT /4	安全統合出力信号のイメージ (外部 NCK インタフェース)	出力ワードの番号	R/W	r/w
\$A_OUTSEP[n]	BOOL /4	安全統合出力信号のイメージ (外部 PLC インタフェース)	出力番号	R	r
\$A_OUTSEPW[n]	INT /4	安全統合出力信号のイメージ (外部 PLC インタフェース)	出力ワードの番号	R	r

S. I. : サーボ補間器インタフェース

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$A_INSI[n]	BOOL /4	安全統合入力信号のイメージ (内部 NCK インタフェース)	入力番号	R	r /4
\$A_INSIW[n]	INT /4	安全統合入力信号のイメージ (内部 NCK インタフェース)	入力ワードの番号	R	r /4
\$A_INSIP[n]	BOOL /4	安全統合入力信号のイメージ (内部 PLC インタフェース)	入力番号	R	r /4
\$A_INSIPW[n]	INT /4	安全統合入力信号のイメージ (内部 PLC インタフェース)	入力ワードの番号	R	r /4
\$A_OUTSI[n]	BOOL /4	安全統合出力信号のイメージ (内部 NCK インタフェース)	出力番号	R/W	r/w /4
\$A_OUTSIW[n]	INT /4	安全統合出力信号のイメージ (内部 NCK インタフェース)	出力ワードの番号	R/W	r/w /4
\$A_OUTSIP[n]	BOOL /4	安全統合出力信号のイメージ (内部 PLC インタフェース)	出力番号	R	r /4
\$A_OUTSIPW[n]	INT /4	安全統合出力信号のイメージ (内部 PLC インタフェース)	出力ワードの番号	R	r /4

安全統合用のマーカおよびタイマ

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$A_MARKERSI[n]	BOOL /4	安全統合プログラミング用のマーカ	マーカ番号	R/W	r/w
\$A_MARKERSIP[n]	BOOL /4	PLC 安全統合マーカのイメージ	マーカ番号	R	r
\$A_TIMERSI[n]	DOUBLE /4	タイマの単位、秒。時間は補間サイクルの整数倍で内部的にカウントされる。タイマ変数のインクリメントは、 \$A_TIMERSI[n]=<start value> を割当てることによってスタートされる。インクリメントは、負の値： \$A_TIMERSI[n]=-1 を割当てることによって停止される。タイマ変数が実行中であろうと、停止中であろうと、現在の時間値を読取ることができる。-1 の割当てによってタイマ変数が停止された後も、現在の時間値は保存されたままで、読取ることができる。	タイマ番号	R/W	r/w

診断

名称	タイプ / SW	説明 / 値	インデックス	PP アクセス	SA アクセス / SW
\$A_PROTO	BOOL /4	ロギング機能 : 0: 停止 1: 起動	R/W	r/w	

2.4 シンクロナイズドアクション中のアクション

アクション

アクションコード DO ... のあと、各シンクロナイズドアクションには以下が含まれます。

- 1 つまたは複数（最大 16）のアクション、または、テクノロジサイクル
（以下の説明では、これら 2 つの構成要素のことを一般にアクションと呼びます）。

これらは、適切な条件が満足されると実行されます。

複数のアクション

シンクロナイズドアクションの中に含まれる複数のアクションは、適切な条件が満足されると、同一の補間サイクル中で起動されます。

可能なアクションのリスト

以下のアクションを、シンクロナイズドアクションの "Action"（アクション）セクションにプログラムすることができます。

表 2.1 シンクロナイズドアクション中のアクション

... DO ...	意味	参照
Mxx Sxx Hxx	PLC への補助機能出力	2.4.1「補助機能 M, S, および H の PLC への出力」
SETAL(nr)	アラーム／エラーリアクションの設定	2.4.19「アラームの設定／エラーリアクション」
\$A... = ... \$V... = ... \$AA_OFF = \$AC_OVR = \$AA_OVR = \$AC_VC = \$AA_VC = \$\$SN_SW_CAM_ ... \$AC_FCT...	リアルタイム変数の書込み： - オーバレイドモーション - 速度制御： パス速度 軸速度 加算パス送り速度オーバーライド 加算軸補正值 SW カム位置の変更（セッティングデータ）およびその他の全 SD FCTDEF パラメータの上書き	2.4.2「リアルタイム変数の設定（書込み）および読取り」 2.4.3「SW カム位置および時間の変更（セッティングデータ）」 2.4.4「FCTDEF」
RDISABLE STOPREOF DELDTG FTOC SYNFCT ZYKL_T1 (e.g.)	シンクロナイズドアクション手順： 読み込みディスエーブルの起動 エンド送り停止 移動距離削除 オンラインツールオフセット 多項式評価 テクノロジサイクルのコール	2.4.7「RDISABLE」 2.4.8「STOPREOF」 2.4.9「DELDTG」 2.4.6「オンラインツールオフセット FTOC」 2.4.5「多項式評価 SYNFCT」 2.5「テクノロジサイクルのコール」

... DO ...	意味	参照
\$AA_OVR[x]= 0 AXIS_X (e.g.) POS[u]= ... FA[u]= ... MOV[u]= >0 MOV[u]= <0 MOV[u]= =0	位置決め軸の制御： 軸モーションのディスエーブル 軸プログラムの呼出し 位置 軸送り速度の定義 指令軸の移動，連続： - 前方 - 後方 - 停止	2.4.10 「プログラムされた軸モーションのディスエーブル」 2.4.11 「指令軸のスタート」 2.4.11 「指令軸のスタート」 2.4.12 「シンクロナイズドアクションからの軸モード」 2.4.13 「シンクロナイズドアクションからの軸スタート」 " " "
SPOS M3, M4, M5, S = \$AA_OVR[S1]= 0	主軸： 位置 回転方向，停止，速度 主軸モーションのディスエーブル	2.4.14 「シンクロナイズドアクションからの軸モーション」
PRESETON(,)	実際値の設定	2.4.15 「シンクロナイズドアクションからの実際値の設定」
LEADON LEADOF TRAILON TRAILOF	カップリングの起動／停止： スレーブ軸とマスタ軸をカップリングする カップリングのキャンセル 非同期カップリングモーション ON 非同期カップリングモーション OFF	2.4.16 「カップリング軸およびカップリングの起動／停止」
MEAWA, MEAC	移動距離を削除しない測定 周期的な測定	2.4.17 「シンクロナイズドアクションからの測定値」
SETM CLEARM	チャンネル同期： 待機マーカの設定 待機マーカのクリア	2.4.18 「チャンネル同期用待機マーカの設定および削除」
LOCK UNLOCK RESET	シンクロナイズドアクションの協調： - シンクロナイズドアクション／テクノロジーサイクルのディスエーブル - シンクロナイズドアクション／テクノロジーサイクルのイネーブル - テクノロジーサイクルのリセット	2.5.1 「シンクロナイズドアクション，テクノロジーサイクル，パートプログラム（および PLC）」

2.4.1 補助機能 M, S, および H の PLC への出力

補助機能出力に関する一般情報については、次を参照してください。

参照： /FB/, H2, PLC に対する補助機能出力

例

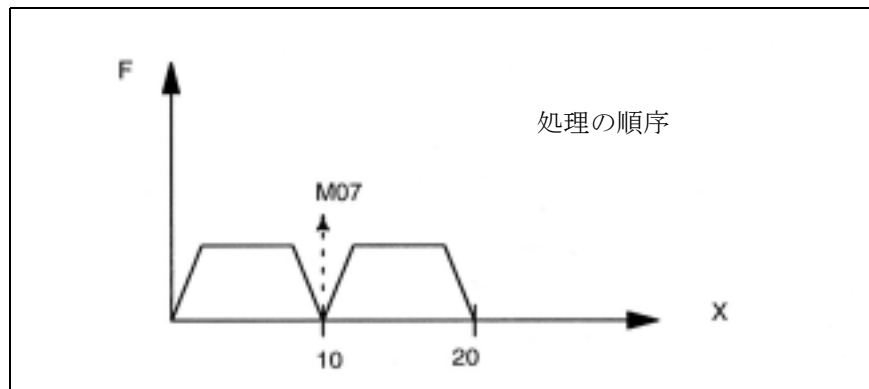
シンクロナイズドアクションに補助機能出力を実装する利点が、次の例で説明されています。特定の位置での冷却剤のスイッチオン

シンクロナイズドアクションを使用しない場合のソリューション：3つのブロック

N10 G1 X10 F150

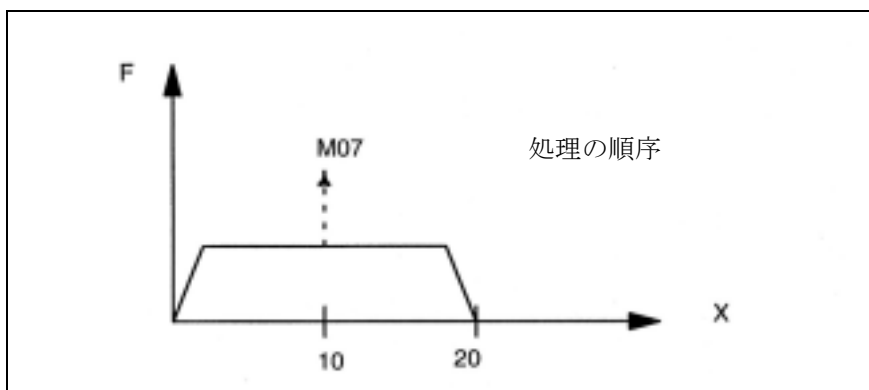
N20 M07

N30 X20



シンクロナイズドアクションを使用した場合のソリューション：1つのブロック

N10 WHEN \$AA_IM[X] >= 10 DO M07 G1 X20 F150



PLC への補助機能出力

補助機能 M, S, または H をシンクロナイズドアクションとして PLC に出力することができます。

条件が満足されれば, 補間サイクル中に (PLC 上の割込みと同様) 直ちに出力が起こります。

次の MD にプログラムされるタイミングは, シンクロナイズドアクションに影響しません。

AUXFU_11110: AUXFU_GROUP_SPEC (補助機能グループ指定) あるいは
AUXFU_M_SYNC_TYPE (M 機能の出力タイミング) /
AUXFU_S_SYNC_TYPE (S 機能の出力タイミング) /
AUXFU_H_SYNC_TYPE (H 機能の出力タイミング) /

プログラミング

頻度ボキャブラリワード WHEN または EVERY を使用した補助機能のプログラミングは, ID または IDS を使用しないシンクロナイズドアクション (つまり, ノンモーダルアクション) に対してのみ可能です。

例

WHEN \$AA_IM[X] > 50 DO H15 S3000 M03

; X 軸の実際値が 50 よりも大きければ, H15 を出力し, 新しい主軸速度と新しい回転方向を設定する

制限

10 個の補助機能しか同時に (つまり, PLC の OB40 サイクルで) 出力できません。

パートプログラムおよびシンクロナイズドアクションからの補助機能出力の合計数は, 常に 10 を超えてはなりません。

1 シンクロナイズドアクションブロックあるいは 1 テクノロジサイクルブロックあたりの補助機能の最大数は:

- 5 つの M 機能
- 3 つの S 機能
- 3 つの H 機能

予め定義された M 機能をシンクロナイズドアクションによってプログラムすることはできません。M 機能はアラームによって拒絶されます。

WHEN ... DO M0 ; アラーム

しかし, 主軸 M 機能 M3, M4, M5 および M17 を, テクノロジサイクルエンドとしてプログラムすることはできます。

確認応答

補助機能出力を含むテクノロジサイクルブロック (2.5 「テクノロジサイクルのコール」を参照) は, ブロック中のすべての補助機能に対して PLC から確認応答が送られるまで, 完全に処理されません。テクノロジサイクル中の次のブロックの処理は, 前のブロック中のすべての補助機能に対して PLC から確認応答が送られるまで, 実行されません。

SW 5

SW 5 以降では、他にもいくつかの確認応答方法が導入されています。

- ブロックチェンジ遅延を伴わない補助機能出力

最初に、PLC での並列処理としての、高速補助機能 (QUICK)、次に確認応答が予想された補助機能出力。

ユーザは、H 補助機能用のデータタイプとして INT か REAL のいずれかを選択することができます。PLC ユーザプログラムは、その定義に基づいて値を解釈しなければなりません。H 補助機能用の INT 値の範囲は次のように拡大されました：-2 147 483 648 ～ 2 147 483 647。

参照： /FB/, H2, SW5 における PLC に対する補助機能出力

2.4.2 リアルタイム変数の設定（書込み）および読取り

書込み

2.3.7「シンクロナイズドアクションに関連するシステム変数のリスト」のリストで "Write from synchronized actions"（シンクロナイズドアクションからの書込み）というアクセス権を示す+符号でマークされているリアルタイム変数を、シンクロナイズドアクションに含まれるアクションの中に書込むことができます。

- ・ マシンおよびセッティングデータ、例, \$MN_..., \$MC_..., \$MA_...
または \$SN_..., \$SC_..., \$SA_...

（注）メインラン中にオンラインで書込まなければならないマシンおよびセッティングデータは、\$\$_... を使用してプログラムされなければなりません。

有効性

シンクロナイズドアクションから書込まれたマシンデータは、すぐに有効となるようにコード化されなければなりません。そうでない場合、変更された値は処理実行の残りで利用できません。変更後の新しいマシンデータ値の有効性に関する詳細については、次を参照してください。

参照： /LIS/, リスト

例：

```
... DO $MN_MD_FILE_STYLE = 3           ; マシンデータの設定
... DO $SA_OSCILL_REVERSE_POS1 = 10    ; セッティングデータの設定
... DO $A_OUT[1]=1                     ; デジタル出力の設定
... DO $A_OUTA[1]= 25                   ; アナログ値の設定
```

読取り

機能用の入力量としてリアルタイム変数に割当てるため、および条件を公式化するために、シンクロナイズドアクション中の変数に読取りアクセスすることができます。これらの変数は、2.3.7「シンクロナイズドアクションに関連するシステム変数のリスト」のリストでは "Read from synchronized actions"（シンクロナイズドアクションからの読取り）というアクセス権を示す文字rによって示されています。

- ・ マシンデータ、セッティングデータ、例 \$SN_..., \$SC_..., \$SA_...

（注）処理中に量が増える可能性のあるマシンおよびセッティングデータをメインラン中にオンラインで指定する必要がある場合、そのセッティングデータは\$\$_... を使用してプログラムされなければなりません。量が増えない変数の場合は、識別子の前に\$記号を1つ入力するだけで十分です。

例：

```
WHEN $AC_DTEB < 5 DO ...                ; 条件中のブロックエンドからの距離を読取る
DO $R5= $A_INA[2]                        ; アナログ入力2の値を読取り、算術変数を割当てる
```


2.4.3 SW カム位置および時間の変更（セッティングデータ）

概要

"Software cams"（ソフトウェアカム）機能を使用すると、位置に依存したカム信号を PLC あるいは NCK I/O に出力させることができます。

参照： /FB/, N3, ソフトウェアカム, ポジションスイッチ信号

機能

シンクロナイズドアクションをプログラムすることにより、信号出力が設定されるカム位置を変更することができます。カム位置を変更するために、現行のセッティングデータに対して書込みが行われます。シンクロナイズドアクションを介して次のセッティングデータを変更することができます：

\$\$\$SN_SW_CAM_MINUS_POS_TAB_1[0..7]; マイナスカムの位置

\$\$\$SN_SW_CAM_MINUS_POS_TAB_2[0..7]; マイナスカムの位置

\$\$\$SN_SW_CAM_PLUS_POS_TAB_1[0..7]; プラスカムの位置

\$\$\$SN_SW_CAM_PLUS_POS_TAB_2[0..7]; プラスカムの位置

例 1

カム位置の変更：

ID=1 WHEN \$AA_IW[x] > 0

DO \$\$\$SN_SW_CAM_MINUS_POS_TAB_1[0] = 50.0

次のセッティングデータを介してリード時間または遅延時間を変更することができます：

\$\$\$SN_SW_CAM_MINUS_TIME_TAB_1[0..7]; マイナスカム上でのリードまたは遅延時間

\$\$\$SN_SW_CAM_MINUS_TIME_TAB_2[0..7]; マイナスカム上でのリードまたは遅延時間

\$\$\$SN_SW_CAM_PLUS_TIME_TAB_1[0..7]; プラスカム上でのリードまたは遅延時間

\$\$\$SN_SW_CAM_PLUS_TIME_TAB_2[0..7]; プラスカム上でのリードまたは遅延時間

例 2

リード／遅延時間の変更：

ID=1 WHEN \$AA_IW[x] > 0

DO \$\$\$SN_SW_CAM_MINUS_TIME_TAB_1[0] = 1.0

(注) カムの直前に、シンクロナイズドアクションを介してソフトウェアカムを速度の関数として設定してはなりません。設定位置と関連カム位置の間には少なくとも 2～3 の補間サイクルがなければなりません。

2.4.4 FCTDEF

アプリケーション

以下のサブセクションで説明されるアクション "Online tool offset FTOC" (オンライン ツールオフセット FTOC) および "Polynomial evaluation SYNFACT" (多項式評価 SYNFACT) では、入力量と出力量の相互関係が多項式の形で定義されることが必要となります。

FCTDEF はこのタイプの多項式を定義します。

グライディングホイールのオンラインドレッシングでの、多項式のアプリケーションの特殊な例については、2.4.6 「オンラインツールオフセット FTOC」を参照してください。多項式を介した負荷依存送りおよびクリアランス制御の例については、2.4.5 「多項式評価 SYNFACT」を参照してください。

多項式の特徴

FCTDEF によって定義される多項式には次の特徴があります：

- ・ 多項式は、FCTDEF コールによってパートプログラム中に生成されます。
- ・ 定義された多項式のパラメータはリアルタイム変数です。
- ・ 個々の多項式のパラメータは、リアルタイム変数の書込みに使用されるのと同じ方法で上書きすることができます。通常、パートプログラム中およびシンクロナイズドアクションのアクションセクション中で許可されています。2.4.2 「リアルタイム変数の設定（書込み）および読取り」を参照してください。

(注) SW バージョン 4 以降では、現行の多項式の有効性リミットおよび係数をシンクロナイズドアクションから変更することが可能です。
例：WHEN ... DO \$AC_FCT1[1]=0.5

多項式の数

SW バージョン 4 以降では、同時に定義できる多項式の数を

MD 28252 : \$MC_MM_NUM_FCTDEF_ELEMENTS

に指定することができます。

ブロック同期多項定義

FCTDEF(

多項式番号,

下限,

上限,

a_0 ,

a_1 ,

a_2 ,

a_3)

出力量 y と入力量 x の関係は次のとおりです :

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

この関数で指定されたパラメータは次のシステム変数に保存されます :

\$AC_FCTLL[n]: 下限, n: 多項式番号

\$AC_FCTUL[n]: 上限, n: 多項式番号

\$AC_FCT0[n]: a_0 係数, n: 多項式番号

\$AC_FCT1[n]: a_1 係数, n: 多項式番号

\$AC_FCT2[n]: a_2 係数, n: 多項式番号

\$AC_FCT3[n]: a_3 係数, n: 多項式番号

この関係を基礎として、関連するシステム変数を介して多項式を直接書込むこと、あるいは変更することも可能です。多項式の有効性の範囲は、リミット \$AC_FCTLL[n] および \$AC_FCTUL[n] を介して定義されます。

多項式評価のコール

保存されている多項式を、次の機能に関連して使用することができます :

- オンラインツールオフセット, FTOC()

- 多項式評価, SYNFACT().

参照 : /PG/, プログラミングガイド 基本編

/PGA/, プログラミングガイド 上級編

/FB/, W4 "Grinding"

2.4.5 多項式評価 SYNFACT

アプリケーション

評価機能をシンクロナイズドアクションのアクションセクションに適用することにより、加工プロセスと同期して、変数を読み取り、多項式でその変数を評価し、その結果を別の変数に書込むことが可能となります。この機能性は、たとえば、次のタスクを実行するのに使用できます：

- ドライブ負荷の関数としての送り
- センサ信号の関数としての位置
- パス速度の関数としてのレーザパワー
- ...

SYNFACT() 評価機能

SYNFACT() には次のパラメータがあります：

SYNFACT(多項式番号,
 リアルタイム変数出力,
 リアルタイム変数入力)

多項式の定義については、2.4.4 「FCTDEF」を参照してください。

SYNFACT の動作原理

"Polynomial number" (多項式番号) によって識別された多項式は、"Real-time variable input" (リアルタイム変数入力) の値で評価されます。その結果は、最大および最小リミットによって制限され、"Real-time variable output" (リアルタイム変数出力) に割当てられます。

例：

FCTDEF(1,0,100,0,0.8,0,0) ; 多項式 1 はすでに定義されている

...

シンクロナイズドアクション：

ID=1 DO SYNFACT(1,\$AA_VC[U1], \$A_INA[2])

; 軸 U1 の加算補正值は、補間サイクルごとに、多項式 1 を基礎にしてアナログ入力値 2 から計算される。

"Real-time variable output" (リアルタイム変数出力) について、加工プロセスに次のように統合される変数を選択することが可能です。

- 加算制御係数 (例、送り速度) として、
- 乗算制御係数 (例、オーバライド) として、
- 位置オフセットとして、あるいは
- 直接

加算送り速度制御

加算制御の場合、プログラムされた値（適用制御では F ワード）が加算係数によって補正されます。

$$F_{\text{アクティブ}} = F_{\text{プログラムされた}} + F_{\text{AC}}$$

以下は、"Real-time variable output"（リアルタイム変数出力）の設定例です：

\$AC_VC 加算パス送り速度オーバーライド、

\$AA_VC[axis] 加算軸送り速度オーバーライド

パス送り速度の加算制御の例

プログラムされた送り速度（軸あるいはパス関連の）は、（正の）X 軸の電流（たとえば、イン送りトルク）によって加算制御されなければなりません。作動点は 5 A にセットされます。送り速度は、± 100 mm/min の間で変更されます。軸電流のずれの大きさは、± 1 A です。

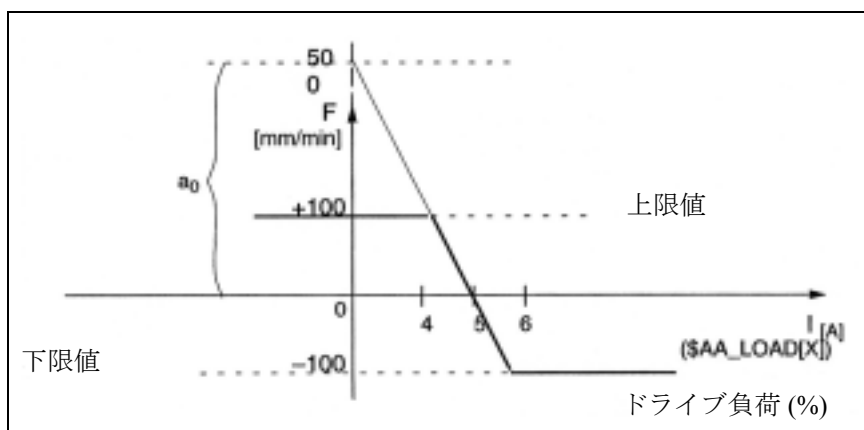


図 2.4 加算制御の例

係数の定義については、2.4.4 「FCTDEF」も参照してください。

$$y = f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

$$a_1 = -\frac{100\text{mm}}{1\text{ min} \cdot \text{A}}$$

$$a_1 = -100 \Rightarrow \text{制御定数}$$

$$a_0 = -(-100) \cdot 5 = 500$$

$$a_2 = 0 \quad (2 \text{ 乗成分はない})$$

$$a_3 = 0 \quad (3 \text{ 乗成分はない})$$

$$\text{上限} = 100$$

$$\text{下限} = -100$$

定義される多項式（番号 1）は次のようになります。

FCTDEF(1, -100, 100, 500, -100, 0, 0)

図 2.4 で与えられた例が、この関数によって完全に定義されます。

適用制御機能は、次のシンクロナイズドアクションによって起動されます。

ID = 1 DO SYNFACT(1, \$AC_VC[x], \$AA_LOAD[x])

; 軸 x の送り速度用の加算補正值は、補間サイクルごとに、多項式 1 を介して、パーセントドライブ負荷値から計算されます。

乗算制御

乗算制御の場合、F ワードは係数（適用制御の場合はオーバライド）によって乗算されます。F アクティブ = F プログラムされた ・ 係数 AC

加工プロセスで乗算係数として機能する変数 \$AC_OVR が、リアルタイム変数出力として使用されます。

乗算制御の例

プログラムされた送り速度（軸あるいはパス関連の）は、ドライブ負荷の関数として乗算制御されなければなりません。作動点は、ドライブ負荷が 30 % で 100 % にセットされます。軸はドライブ負荷が 80 % で停止しなければなりません。プログラムされた値 +20 % に対応する、超過速度は許可されます。

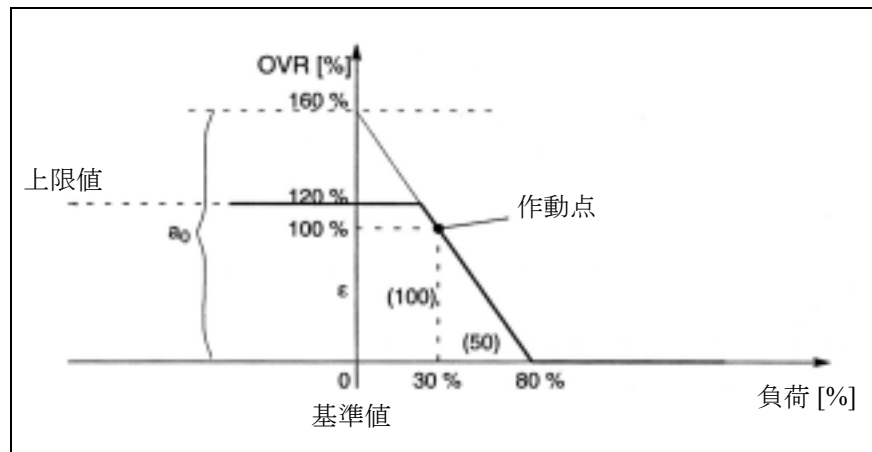


図 2.5 乗算制御の例

係数の定義については、2.4.4 「FCTDEF」 も参照してください。

$$y = f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

$$a_1 = - \frac{100 \%}{(80 - 30) \%} = -2$$

$$a_0 = 100 + (2 \cdot 30) = 160$$

$$a_2 = 0 \quad (2 \text{ 乗成分はない})$$

$$a_3 = 0 \quad (3 \text{ 乗成分はない})$$

$$\text{上限} = 120$$

$$\text{下限} = 0$$

多項式（番号 2）を次のように定義することができます。

FCTDEF(2, 0, 120, 160, -2, 0, 0)

図 2-5 で与えられた例が、この関数で完全に定義されます。

関連シンクロナイズドアクションを次のようにプログラムできます。

ID = 1 DO SYNFACT(2, \$AC_OVR, \$AA_LOAD[x])

;パスオーバーライドは、補間サイクルごとに、多項式 2 を介して、x 軸用のパーセントドライブ負荷から計算されます。

制限付きの位置オフセット

システム変数 \$AA_OFF は、直ちに有効となる軸別オーバーライドを制御します（基準座標系）。オーバーライドのモードは、

MD 36750: \$MA_AA_OFF_MODE で定義されます。

0: 比例評価

1: 積分評価

SW バージョン 4 以降では、絶対的に補正される値（リアルタイム変数出力）を、セッティングデータ

SD 43350: \$SA_AA_OFF_LIMIT に保存されている値に制限することができます。

制限が到達されたかどうかを確認するために、軸別システム変数

\$AA_OFF_LIMIT[axis] を別のシンクロナイズドアクションの中で評価することができます。

値 -1: 補正值のリミットが負の方向で到達されました。

値 1: 補正值のリミットが正の方向で到達されました。

値 0: 補正值がリミット範囲内にありません。

アプリケーション:

クリアランス制御をレーザ加工オペレーションに実装するために、システム変数 \$AA_OFF に関連して、機能 SYNFACT を使用することができます。以下を参照してください。

例

タスク:

レーザ加工オペレーションにおけるセンサ信号の関数としてのクリアランス制御

レーザヘッドが仕上げられた金属のブランクから確実に後退するように、補正值は負の Z 方向に制限されます。(0 オーバーライド (2.4.10 「プログラムされた軸モーションのディスエーブル」を参照) による) "Stop axis" (軸停止) または "Set alarm" (アラームの設定 (2.4.19 「アラームの設定/エラーリアクション」を参照)) などのユーザリアクションをリミット値が到達されたときに起動させることができます。

補足条件:

センサ \$A_INA[3] の入力量の積分評価。

補正值は、基準座標系に、つまり、キネマティックス変換の前に適用されます。プログラムされたフレーム (TOFRAME) は無効です。つまりこの機能は、オリエンテーション方向の 3D クリアランス制御には使用できません。"clearance control" (クリアランス制御) 機能は、動的応答の高いクリアランス制御系あるいは 3D クリアランス制御系を実装

するのに使用できます。次を参照してください。

参照： /FB/, TE1, クリアランス制御

参照： /PG/, プログラミングガイド 基本編

入力量と出力量の相互依存性は、次の図で説明される関係によって保証されます。

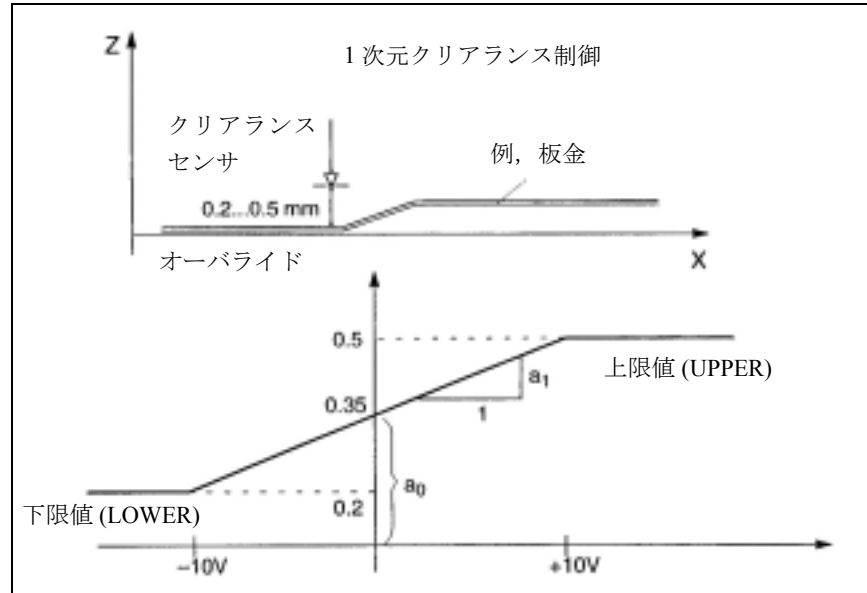


図 2.6 クリアランス制御

%_N_AON_SPF

PROC AON ; クリアランス制御用のサブプログラム ON

FCTDEF(1, 0.2, 0.5, 0.35, 1.5 EX-5); 多項定義：補正は 0.2 ～ 0.5 の範囲に適用される。

ID=1 DO SYNFACT(1,\$AA_OFF[Z], \$A_INA[3]) ; クリアランス制御アクティブ

ID = 2 WHENEVER \$AA_OFF_LIMIT[Z] < 0 DO \$AA_OVR[X] = 0

; リミット範囲が超過されるたびに、X をディスエーブルする。

RET

ENDPROC

%_N_AOFF_SPF

PROC AOFF ; クリアランス制御用のサブプログラム OFF

CANCEL(1) ; クリアランス制御用のシンクロナイズドアクションを
キャンセルする

CANCEL(2) ; リミット範囲のチェックをキャンセルする

RET

ENDPROC

%_N_MAIN_MPF

; メインプログラム

; MD 36750 は、電源オンの前に、積分評価用に 1 に

; セットされた。


```

SSA_AA_OFF_LIMIT[Z]= 1 ; 補正用のリミット値
AON                      ; クリアランス制御 ON
...
G1 X100 F1000
AOFF                     ; クリアランス制御 OFF
M30
    
```

その他の例

適用制御（クリアランス制御）に関連して、多項式リミットの動的適用を説明した例については、6.3.1「可変上限値を使ったクリアランス制御」を参照してください。パス送り速度に適用される適用制御の例については、6.3.2「送り制御」を参照してください。

2.4.6 オンラインツールオフセット FTOC

オンラインツールオフセット

ワークの加工と、グラインディングアプリケーションのグラインディングホイールのドレッシングを同じチャンネルかあるいは別々のチャンネル（加工チャンネルとドレッシングチャンネル）に実装することができます。

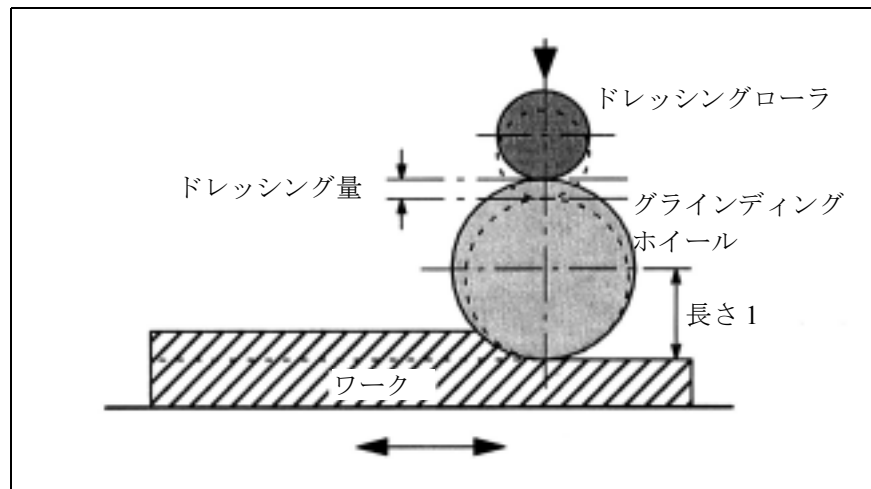


図 2.7 ドレッシングローラを使用した、加工中のドレッシング

参照： /FB/, W4 "Grinding"

補足条件

シンクロナイズドアクション FTOC は SW バージョン 3.2 以降で利用できます。

オンラインオフセットを使用すると、FCTDEF（2.4.4「FCTDEF」を参照）を使用して、基準値（たとえば、軸の実際値）の関数としてプログラムされた多項式に基づいて、オーバーレイドモーションをジオメトリ軸について実装することができます。

FTOC のプログラミング

オンラインオフセットは次のように指定されます。

FTOC(多項式番号,
 Read_real_main_variable, ; 基準値
 長さ 1_2_3,
 チャンネル番号,
 主軸番号)

パラメータ

多項式番号： FCTDEF を使用して予めプログラムされた機能の番号。

Read_real_main_variable: 2.3.7「シンクロナイズドアクションに関連するシステム変数のリスト」で示されている、REAL タイプのすべてのメイン変数が使用できる。

長さ 1_2_3: オフセット値が加算される摩耗パラメータ

- チャンネル番号： オフセットが適用されなければならない目標チャンネル。平行チャンネルからの同時ドレッシングが可能です。チャンネル番号が指定されていない場合、オフセットはアクティブなチャンネルに適用されます。FTOCON を使用したオンラインオフセットは、オフセット目標チャンネルで起動されなければなりません。
- 主軸番号： 主軸番号は、アクティブでないグライインディングホイールをドレッシングする必要がある場合にプログラムされます。前提条件は、"Constant grinding wheel peripheral speed" (一定グライインディングホイール周速) または "Tool monitoring" (ツール監視) がアクティブであることです。主軸番号がプログラムされていない場合、アクティブなツールが補正されます。

例

アクティブなグライインディングホイールの長さを補正する

%_N_DRESS_MPF

FCTDEF(1,-1000,1000,-\$AA_IW[V],1) ; 機能の定義

ID=1 DO FTOC(1,\$AA_IW[V],3,1) ; オンラインツールオフセットの選択：
; V 軸のモーションから得られた、アクティブな
; グライインディングホイールの長さ 3 はチャンネル
; ル 1 で補正される。

WAITM (1,1,2) ; 加工チャンネルとの同期

G1 V-0.05 F0.01, G91

G1 V -....

...

CANCEL(1) ; オンラインオフセットの選択解除

...

(注) シンクロナイズドアクションには、頻度がキャブラリワードもいかなる条件もプログラムされていません。したがって、FTOC アクションは、すべての補間サイクルで、\$AA_IW[V] 以外に依存することなくアクティブです。

2. 4. 7 RDISABLE

プログラムされた読み込みディスエーブル RDISABLE

アクティブなセクション中の RDISABLE 指令は、関連条件が満足されると、ブロック処理を停止させます。プログラムされたモーションシンクロナスアクションの処理は継続します。読み込みディスエーブルは、RDISABLE 用の条件が満足されなくなると直ちに再度キャンセルされます。

イグザクトストップは、読み込みディスエーブルが現在もアクティブであるかどうかに関係なく、RDISABLE を含むブロックのエンドで開始されます。

アプリケーション：この方法は、たとえば、外部入力の関数としてプログラムを補間サイクル中にスタートさせるのに使用することができます。

RDISABLE の例

プログラムされた読み込みディスエーブル

```
WHENEVER $A_INA[2]<7000 DO RDISABLE
```

...

N10 G01 X10 ; N10 の処理中に条件が満足された場合、N10 の終わりで
RDISABLE は有効になる。

N20 Y20

入力 2 での電圧が (1000 が 1 V に対応すると仮定して) 7 V 以下に下がれば、プログラム
処理は停止されます。

この方法のアプリケーション例：障害物がパスから取除かれるまで、読み込みディスエー
ブル

2.4.8 STOPREOF

STOPREOF を使用した前処理停止のエンド

STOPREOF 指令を含むモーションシンクロナイズドアクションは、条件が満足されると、現
行の前処理停止をキャンセルします。

STOPREOF は、常にボキャブラリワード 'WHEN' を使用して、ノンモーダル指令として
プログラムされなければなりません。

アプリケーション：ブロックエンドでの高速プログラム分岐。

STOPREOF の例

プログラム分岐

```
WHEN $AC_DTEB<5 DO STOPREOF
```

```
G01 X100
```

```
IF $A_INA[7]>5000 GOTO ラベル 1
```

ブロックエンドまでの距離が 5 mm 以下であれば、エンド前処理停止。入力での電圧が
(1000 が 1 V に対応すると仮定して) 5 V 以下に下がれば、ラベル 1 まで前方ジャンプする。

2.4.9 DELDTG

移動距離の削除

シンクロナイズドアクションを使用すれば、パスおよび指定された軸について移動距離
の削除を条件の関数として起動させることができます。

- 高速／準備された、移動距離の削除

パス用の高速／準備された DDTG

高速／準備された移動距離削除は、時間が重要なアプリケーションに使用されます。つ
まり、

- 移動距離削除から次のブロックの始めまでの時間を非常に短くする必要がある場
合、あるいは
- 移動距離削除が起動される確立が高い場合。

DELDTG

移動距離削除は、シンクロナイズドアクション DELDTG を使用してプログラムされます。移動距離が削除された後、残りのパス距離は \$AC_DELT に保存されます。このように、連続パスモードは、高速移動距離削除によってブロックエンドで中断されます。

制限事項：

パス用の移動距離削除は、ノンモーダルシンクロナイズドアクションとしてしかプログラムできません。ツール半径補正がアクティブな場合は、高速移動距離削除は使用できません。

DELDTG の例

```
... DO DELDTG
N100 G01 X100 Y100 F1000
N110 G01 X...
IF $AC_DELT > 50
...
```

軸用の高速／準備された DDTG

軸用の高速／準備された移動距離削除は、ノンモーダルアクションとしてプログラムされなければなりません。

アプリケーション：

パートプログラムにプログラムされた位置決めモーションは、軸の移動距離削除によって停止されます。1つの指令で複数の軸を同時に停止させることができます。

```
... DO DELDTG(axis1, axis2, ...)
```

DELDTG（軸）の例

```
WHEN $A_INA[2]>8000 DO DELTG(X1)
                                ; 入力 2 での電圧が 8 V 以下に下がった場合、軸 X1 用の
                                ; 移動距離を削除
POS[X1] = 100                  ; 次の位置
R10 = $AA_DELT[X 1]            ; 軸の移動距離を R10 に転送する
```

移動距離が削除された後、軸の移動距離は変数 \$AA_DELT[axis] に保存されます。

(仮定：1000 は 1 V に対応する)。

2.4.10 プログラムされた軸モーションのディスエーブル

タスク

軸は、加工ルーチン内にプログラムされますが、特定の状況ではブロックの始めからスタートされてはなりません。

ソリューション

シンクロナイズドアクションを使用して、軸をスタートさせる時間になるまで、0 オーバライドを維持します。

例：

```
WHENEVER $A_IN[1]==0 DO $AA_OVR[W]=0  
G01 X10 Y25 F750 POS[W]=1500 FA[W]=1000
```

；位置決め軸は、パス加工と非同期でスタートされる

；イネーブル信号はデジタル入力を介してセットされる

(注) 軸モーションディスエーブルを PLC 軸（たとえば、マガジン軸）についてプログラムすることもできます。

2. 4. 11 指令軸のスタート

概要

シンクロナイズドアクションから，パートプログラムとは完全に非同期で，軸の位置決めおよびスタート，停止を行うことが可能です。このプログラミング方法は，周期オペレーションまたは主にイベント制御されたオペレーションに推奨されます。シンクロナイズドアクションからスタートされる軸を指令軸と呼びます。

補足条件

軸をパートプログラムとシンクロナイズドアクションの両方から同時に動作することはできませんが，逐次的に動作することはできます。シンクロナイズドアクションから軸を動作させてから，パートプログラムに再度軸をプログラムした場合，遅延が生じることがあります。

(注) MD 30450: IS_CONCURRENT_POS_AX は，軸が主として指令軸として使用されるか，あるいは，パートプログラムによるプログラミングに使用されるかを示します。

0: 競合軸でない

1: 競合軸（指令軸）

例 1

```
...
ID=1 EVERY $A_IN[1]==1 DO POS[X]=100
...
```

例 2

軸モーションをテクノロジサイクルの形で開始させることができます。
(2.5「テクノロジサイクルのコール」を参照)。

メインプログラム：

```
...
ID=2 EVERY $A_IN[1]==1 DO AXIS_X
...
```

軸プログラム：

```
AXIS_X:
        M100
        POS[X]=100
        M17
```

プログラミング

位置決め軸モーションは，次のパートプログラムから来ているので，シンクロナイズドアクションの中にプログラムされます。

```
ID = 1 EVERY $AA_IM[B] > 75 DO POS[U]=100
```

プログラムされた位置の評価は、セッティング G70 と G71 のどちらが現在のパートプログラムブロックでアクティブであるかに基づいて、インチ法あるいはメートル法で実行されます。

SW 5 では、G70/G71 および G700/G710 をシンクロナイズドアクションの中に直接プログラムすることもできます。

これにより、パートプログラムでのプログラミングから独立して、指令軸動作のインチ／メートル評価を定義することができます。

```
ID = 1 WHENEVER $A_OUT[1] == 1 DO G710 POS[X]=10
```

```
ID = 2 EVERY G710 $AA_IM[Z] > 100 DO G700 POS[Z2]
```

(注) G70, G71, G700, G710 しかシンクロナイズドアクションの中にプログラムできません。2.1「シンクロナイズドアクションの構成要素」を参照してください。

シンクロナイズドアクションブロックの中にプログラムされた G 機能は、そのシンクロナイズドアクションについてのみ、あるいはそのテクノロジーサイクル内でのみ有効です。G 機能はパートプログラム中の以降のブロックには影響しません。

参照： /PG/ Kap. 3 "Positional Parameters"

アブソリュート／インクリメンタルエンド位置

エンド位置をアブソリュートあるいはインクリメンタルにプログラムすることができます。現在処理中のメインプログラムブロックで G90 と G91 のどちらがアクティブであるかに基づいて、位置はアブソリュートあるいはインクリメンタルにアプローチされます。値をアブソリュート値と解釈しなければならないか、インクリメンタル値と解釈しなければならないかを明示的にプログラムすることが可能です。

IC: インクリメンタル

AC: アブソリュート

DC: 直接。つまり、最短経路を経由して回転軸を位置決めする

ACN: モジュロ回転軸を絶対的に、モーションの負方向に位置決めする

ACP: モジュロ回転軸を絶対的に、モーションの正方向に位置決めする

CAC: 軸をコード化された位置へ絶対的に移動させる

CIC: 軸をコード化された位置へ相対的に移動させる

CDC: 回転軸をコード化された位置へ最短経路を経由して移動させる

CACN: モジュロ回転軸をコード化された位置へ負方向に移動させる

CACP: モジュロ回転軸をコード化された位置へ正方向に移動させる

コード化された位置は、マシンデータに保存される設定値です。

例 1 固定値

```
ID = 1 EVERY $AA_IM[B] > 75 DO POS[U]=IC(10)
```

; イベントが発生すれば、U 軸を 10 前進させる

例 2 現在値

移動パスがリアルタイム変数からリアルタイムで生成されます。

```
ID = 1 EVERY $AA_IM[B] > 75 DO POS[U]=$AA_MW[V]-$AA_IM[W] + 13.5
```


軸フレーム

以下ではシンクロナイズドアクションおよび軸フレームの応答について説明しています。

影響

位置決めモーションがシンクロナイズドアクションから実行されると、ツール長補正と同様に、プログラム可能および設定可能なフレームの軸オフセット、スケーリングおよびミラーリング機能（G54 など）がすべて動作可能になります。

現在のブロックで動作可能なフレームはどれでも有効になります。回転が現在のブロックでアクティブである場合、アラームが出力され、位置決めモーションから開始された位置決めモーションを拒絶します。

例

TRANS X20

IDS= 1 EVERY \$A_IN==1 DO POS[X]=40

G1 Y100 ; この入力の設定されれば、X が 60 に位置決めされる

...

TRANS X-10

G1 Y10 ; この入力の設定されれば、X が 30 に位置決めされる

抑止

フレームおよびツール長の影響は、MD 32074: FRAME_OR_CORRPOS_NOTALLOWED によって抑止することができます。

2.4.12 シンクロナイズドアクションからの軸送り

送り速度

軸送り速度をエンド位置に追加してプログラムすることができます。

```
ID = 1 EVERY $AA_IM[B] > 75 DO POS[U]=100 FA[U]=990
```

指令軸用の軸送り速度は、モーダルアクションを実行します。軸送り速度は、アドレス FA のもとにプログラムされます。デフォルトは軸別マシンデータ

MD 32060: POS_AX_VELO を介して設定されます。

送り速度値は、固定量にプリセットされるか、リアルタイム変数からリアルタイムで生成されます。

計算された送り速度の例

```
ID = 1 EVERY $AA_IM[B] > 75 DO POS[U]=100 FA[U]=$AA_VACTM[W]+100
```

送り速度値は、直線送りあるいは回転送りとしてプログラムされます：

送りタイプは、セッティングデータ：

SD 43300: \$SA_ASSIGN_FEED_PER_REV_SOURCE によって決定されます。このデータの変更は、オペレータ入力によって行うか、PLC から行うか、あるいはパートプログラムから行います。パートプログラムコンテキストと同期して、送りタイプを指令 FPRAON, FPRAOF によって切替えることができます。次も参照してください。

参照： /FB/, V1 送り機能

(注) モーションシンクロナイズドアクションからの軸送り速度は、補助機能として PLC に出力されません。そうでない場合、パラレル軸のテクノロジーサイクルが互いにブロックし合います。

2.4.13 シンクロナイズドアクションからの軸のスタート／停止

スタート／停止

エンド位置が指定されていなくても、シンクロナイズドアクションから指令軸を停止させることができます。この場合、新しいモーション指令または位置決め指令によって別のモーションがセットされるまで、あるいは、停止指令によって軸が停止されるまで、軸はプログラムされた方向へ移動されます。この方法は、たとえば、継続的に回転する回転軸をプログラムするのに使用できます。

スタートおよび停止は、位置決めモーションと同じ方法でプログラムされます。

MOV[axis]= 値

値のデータタイプは INT です。

値の符号がモーションの方向を決定します。

> 0: 正の方向へ軸モーション

< 0: 負の方向へ軸モーション

== 0: 軸モーションの停止

動作中のインデックス軸が指令 MOV[axis]=0 によって停止された場合、その次のインデックス位置が JOG モードの場合と同様の方法でアプローチされます。

FA[axis]= 値（上記を参照）を使用して、モーション用の送り速度をプログラムすることができます。軸送り速度がプログラムされていない場合、送り速度値は、シンクロナイズドアクションからすでに起動されている軸モーションから得られるか、あるいは、MD 32060: POS_AX_VELO を介して設定された軸速度から得られます。

例

... DO MOV[u]=0 ; 条件が満足されれば直ちに軸モーションを停止する

2.4.14 シンクロナイズドアクションからの主軸モーション

一般事項

位置決め軸と同様に、シンクロナイズドアクションから主軸をスタート、位置決め、停止させることも可能です。パートプログラムにプログラムされた主軸モーションをブロックすることによって、あるいは、シンクロナイズドアクションから軸モーションを制御することによって、主軸動作を定義された点からスタートさせることができます。

スタート／停止

これら 2 つの機能の使用は、周期オペレーションあるいは主にイベント制御されたオペレーションに推奨されます。

イベントが発生するまで停止

アプリケーション：

主軸は加工ルーチン内にプログラムされますが、特定の状況ではブロックの始めからスタートされてはなりません。シンクロナイズドアクションを使用して、主軸がスタートされるまで 0 オーバライドを維持します。

例：

```
ID=1 WHENEVER $A_IN[1]==0 DO $AA_OVR[S1]=0
```

```
G01 X100 F1000 M3 S1=1000
```

；主軸はパス加工と非同期でスタートされる；

；スタート指令はデジタル入力を介して設定される

補助機能，速度，位置

これらの機能は、パートプログラムで使用される方法と全く同じ方法で、シンクロナイズドアクションのアクションセクションにプログラムされます。

指令：S= ..., M3, M4, M5, SPOS= ...

例：

```
ID = 1 EVERY $A_IN[1]==1 DO M3 S1000
```

```
ID = 2 EVERY $A_IN[2]==1 DO SPOS=270
```

数字の拡張子が指定されない場合、指令はマスタ主軸に適用されます。数字の拡張子を指定することにより、各主軸を個別的に起動させることができます。

```
ID = 1 EVERY $A_IN[1]==1 DO M1=3 S1=1000 SPOS[2]=90
```

この位置決め方法のプログラミングでは、位置決め軸の場合と同様の規則が適用されます（上記を参照）。

同時指令が、軸／主軸について、同時にアクティブなシンクロナイズドアクションを介して入力された場合、その指令はプログラムされた順序で適用されます。

例：

```
ID=1 EVERY $A_IN[1]==1 DO M3 S300 ; 回転方向および速度
```

```
ID = 2 EVERY $A_IN[2]==1 DO M4 S500; 回転方向および速度
```

```
ID=3 EVERY $A_IN[3]==1 DO S1000
```

；アクティブな主軸回転について新しい速度の設定

```
ID=4 EVERY ($A_IN[4]==1 ) AND ($A_IN[1]==0) DO SPOS=0
```

；主軸の位置決め

送り速度

次の指令を使用して、"Position spindles"（主軸の位置決め）用の送り速度をシンクロナイズドアクションからプログラムすることができます。

FA[Sn]= ...

：

SW リミットスイッチ，作業エリア制限

SW リミットスイッチおよび作業エリア制限によって加えられた制限は，シンクロナイズドアクションから起動された軸／主軸動作にも適用されます。

制限がシンクロナイズドアクションからの動作に与える影響

G25/G26 によってプログラムされた作業エリア制限は，セッティングデータ SD 43400: \$SA_WORKAREA_PLUS_ENABLE の関数として考慮されます。

パートプログラムにおける G 機能 WALIMON / WALIMOF による作業エリア制限の起動および停止は指令軸に影響しません。

軸協調

位置決め指令 (POS, MOV) が，パス軸あるいは PLC 軸としてすでに動作している軸についてシンクロナイズドアクションからスタートされた場合，処理はアラームを伴って強制終了されます。

PP および SA による交互の軸動作

通常，軸はモーシオンブロック中のパートプログラム (PP) から動作されるか，位置決め軸としてシンクロナイズドアクション (SA) から動作されます。しかし，同一の軸が，（パス軸または位置決め軸として）パートプログラムと，シンクロナイズドアクションの両方から交互に移動されなければならない場合，この 2 つの軸モーシオンの間に協調転送が生じます。

例

; パートプログラムとシンクロナイズドアクションの両方から交互に X 軸を移動する

N10 G01 X100 Y200 F1000 ; パートプログラムにプログラムされた X 軸

..

N20 ID=1 WHEN \$A_IN[1]==1 DO POS[X]=100 FA[X]=200

; デジタル入力 that 適用されれば，シンクロナイズド
アクションから

; 位置決めをスタートさせる ...

CANCEL(1) ; シンクロナイズドアクションの選択

...

N100 G01 X100 Y200 F1000 ; X: パス軸

; シンクロナイズドアクションから X が位置決め
されるように，

; デジタル入力が 1 にあれば，モーシオンの前に遅延。

瞬時遷移

指令軸と主軸との間の遷移が可能です。

初期状況

複数のシンクロナイズドアクションが同時にアクティブであることが可能なので，軸がすでにアクティブなときに軸モーシオンがスタートされるという状況が生じる可能性があります。

応答

この場合、最も新しく起動されたモーションが適用できます。POS モーションと MOV モーションを交互に起動させることができます。

モーション方向における逆転がこの状態になると、軸がまず減速され、次に反対方向に位置決めされます。

例：

```
ID=1 EVERY $AC_TIMER[1] >= 5 DO POS[V]=100 FA[V]=560
```

```
ID=2 EVERY $AC_TIMER[1] >= 7 DO POS[V]=$AA_IM[V] + 2 FA[V]=790
```

; \$AC_TIMER[1] のプログラミングのため、ID=2 を使用したシンクロナイズドアクションが最も新しく起動されたアクションとなる。このアクションの指令が ID=1 ... 中の指令に代わって適用される。

したがって、指令軸が動作している間に、指令軸用のエンド位置および送り速度を調整することができます。

例：信号による起動

```
ID=1          EVERY $A_IN[1]==1 DO POS[U]=$AA_IM[U]+$AA_IM[V]*.5
              FA[U]=$AA_VACTM[U]+10
```

正常遷移

x でマークされた遷移が正常です：

in ↓ to →	POS	MOV=1 MOV= - 1	MOV=0	SPOS	M3 M4	M5	LEADON	TRAIL- ON
静止軸								
軸モード	x	x	x	x	x	x	x	x
位置制御された主軸	x	x	x	x	x	x		
速度制御された主軸				x	x	x		
動作中の軸								
軸モード	x	x	x				x	x
位置制御された主軸								
速度制御された主軸				x	x	x		

x でマークされていない遷移はアラームで拒絶されます。

例：正常遷移

```
N10 WHEN $AA_IM[Y] >= 5 DO MOV[Y]=-1          ; 位置 +5 から負の方向に軸を
                                              ; スタートさせる
```

```
N20 WHEN TRUE DO POS[Y]=20 FA[Y]=500          ; ブロックが到達すると、
                                              ; Y 軸をスタートさせる
```

軸カップリング用の瞬時遷移

位置決め軸モーションと、シンクロナイズドアクションを介してプログラムされた軸カップリングの結果生じた動作を交互に起動させることができます。

-2.4.16「カップリング軸およびカップリングの起動／停止」および次を参照してください。

参照： /M3/, カップル軸とマスター値カップリング

マスタ値カップリングにおける正常遷移は、上記の表では LEADON でマークされています。カップリング軸モーションにおける正常遷移は、TRAILON でマークされています。

2.4.15 シンクロナイズドアクションからの実際値の設定

アプリケーション

機械座標系における制御原点を再定義するのに PRESETON 機能を使用することができます。

機能

プリセットが適用されると、軸は動作されません。新しい位置値が現在の軸位置の代わりに入力されるだけです。

プログラミング

値は、1 つの軸についてしか各シンクロナイズドアクションにプログラムできません。

例

```
WHEN $AA_IM[a] >= 89.5 DO PRESETON(a, 10.5)
```

PRESETON (軸, 値)

軸: 制御原点が変更されなければならない軸

値: 制御原点を変更しなければならない量

許可されるアプリケーション

シンクロナイズドアクションからの PRESETON を次の軸についてプログラムすることができます。

- ・ パートプログラムからスタートされたモジュロ回転軸、および
- ・ シンクロナイズドアクションからスタートされたすべての指令軸

制限

変換に関連する軸について PRESETON をプログラムすることはできません。

例

"On-the-fly parting" (即時の切断) アプリケーションに関連した PRESETON の使用法の例については、6.7.3 「即時の切断」を参照してください。

2. 4. 16 カップリング軸およびカップリングの起動／停止

概要

下記の機能についての詳細は以下を参照してください。

参照： /FB/, M3, カップリング軸

- カップリング軸
スレーブ軸は、カップリング係数を介してマスタ軸にリンクされます。
- カーブテーブル
カーブテーブルはマスタ値とスレーブ値との（複雑な）関係を表します。下記のこと
がマスタ値として適用されます：
 - 制御装置によって生成された指令値
 - エンコーダによって測定された実際値
 - 外部で指定された数量
 スレーブ軸がカーブテーブルによってマスタ軸にリンクされているような状況が，特
にシンクロナイズドアクションと関係があります。
- マスタ値カップリング
下記のマスタ値カップリングがパートプログラム用に実装できます。
 - 軸マスタ値カップリング
 - パスマスタ値カップリング
 シンクロナイズドアクションでは軸マスタ値カップリングしか利用できません。

カップリング軸

シンクロナイズドアクションから、カップリング係数を使用してスレーブ軸とマスタ軸
との間の割当てを定義したり同時に起動したりできます。

... DO TRAILON(FA, LA, Kf)

ただし：

FA	スレーブ軸
LA	マスタ軸
Kf	カップリング係数

カップリングされた軸のグルーピングを分離するための指令は下記の通りです：

... DO TRAILOF(FA, LA, LA2)

ただし：

FA	スレーブ軸
LA	マスタ軸
LA2	マスタ軸 2, オプション

カーブテーブル

カーブテーブルに保存されているマスタ量とスレーブ量との関係は、他の REAL 関数
(例えば SIN, COS) と同様の方法でシンクロナイズドアクションで利用することができ
ます：

スレーブ値の計算

カーブテーブル n に基づいてマスタ値から計算されたスレーブ値は、算術変数に割り当てなければなりません。

例：

```
... DO $R17=CTAB(LW, n, grad)
```

ただし：

LW	マスタ値
n	カーブテーブルの番号
grad	ピッチパラメータ，結果

(上記以外の 2 つのオプションのスケーリング用パラメータ：

- スレーブ軸
- マスタ軸)

例：

```
DEF REAL GRADIENT
...
WHEN $A_IN[1] == 1 DO $R17 = CTAB(75.0, 2, GRADIENT)
```

マスタ値の計算

シンクロナイズドアクションから、スレーブ値に対する具体的なマスタ値をカーブテーブルに基づいて計算できます。

例：

```
... DO $R18=CTABINV(FW, aprLW, n, grad)
```

ただし：

FW	マスタ値
aprLW	カーブテーブル逆関数が不明瞭な場合にはっきりしたマスタ値を指定できるような近似マスタ値
n	カーブテーブルの番号
grad	ピッチパラメータ，結果

(上記以外の 2 つのオプションのスケーリング用パラメータ：

- スレーブ軸
- マスタ軸)

機能 CTAB および CTABINV は、シンクロナイズドアクションの条件セクションにもアクションセクションにもプログラムできます。

軸マスタ値カップリング

NR という番号を付けて保存されたカーブテーブルに基づいた追従軸 FA とリーディング軸 LA とのカップリングは、シンクロナイズドアクションのアクションセクションで下記のようにコールされます：

```
... DO LEADON(FA; LA, NR)
```

ただし：

FA	追従軸
LA	リーディング軸
NR	カーブテーブルの番号

シンクロナイズドアクションからの軸カップリングの停止

別の条件が満足されて軸マスタ値カップリングを再びキャンセルしなければならない場合、アクションは下記のようにプログラムしなければなりません：

```
... DO LEADOF(FA, LA)
```

システム変数

システム変数リストの中で指定されるようなマスタ値カップリングのシステム変数は、パートプログラムおよびシンクロナイズドアクションから読取り／書込みできます。

2.3.7「シンクロナイズドアクションに関連するシステム変数のリスト」を参照してください。

同期検出

システム変数 \$AA_SYNC[ax] はパートプログラムおよびシンクロナイズドアクションから読取ることができ、追従軸 FA が同期しているかどうか、およびどのような方法で同期しているかを示します：

- 0：同期していない
- 1：同期（粗）（MD 37200:COUPLE_POS_TOL_COARSE へのアクセス）
- 2：同期（微）（MD 37210:COUPLE_POS_TOL_FINE へのアクセス）

アプリケーションの定義

パートプログラムの中で直接起動されたカップリングはブロックリミットで起動されます。シンクロナイズドアクションからカップリングを起動する追加のオプションを使用すると、イベント制御された差動起動を実行することができます。例えば：

- 特定の軸のパスについてのブロックの先頭から、
- 特定の移動距離についてのブロックエンドまで、
- デジタル入力信号の発生、または
- これら全ての組合せ

2.1「シンクロナイズドアクションの構成要素」の条件を参照してください。

カップリング機能およびカーブテーブルのプログラミングに関するその他の情報については下記を参照してください。

参照： /PGA/, プログラミングガイド 上級編

(注) シンクロナイズドアクションを介してカップリングされたときにどのような動作ステータスになっていても、軸は制御系によって同期させられます。詳細については機能 M3 の説明を参照してください。

例

6.7.3「即時の切断」を参照してください。カーブテーブルを使用して実行される軸カップリングの説明例が記載されています。

2.4.17 シンクロナイズドアクションからの測定値

概要

下記の測定機能はパートプログラム用です：
MEAS, MEAW, MEASA, MEAWA, MEAC

参照： /PGA/, プログラミングガイド 上級編
/FB/, M5, 計測機能

シンクロナイズドアクションでは下記のもののしか使用できません：

- MEAWA 移動距離の削除なしの軸測定
- MEAC 連続した軸測定

測定機能は、パートプログラムのモーションブロックでは1度に1ブロックずつに制限されていても、シンクロナイズドアクションからは何度でも起動したり停止したりできます：

(注) スタティックシンクロナイズドアクションを使用すると、JOG モードでも測定が可能です。

プログラミング

MEAWA[axis]=(mode, trigger_event_1, trigger_event_2,
trigger_event_3, trigger_event_4)
; 移動距離の削除なしで軸測定を起動する

MEAC[axis]=(mode, meas._memory, trigger_event_1, trigger_event_2,
trigger_event_3, trigger_event_4)
; 連続した軸測定を起動する

軸： 測定が行なわれる軸

表 2.2 モードの意味：

10 の桁	1 の桁	意味
	0	測定ジョブを強制終了する
	1	同時に 4 つまでトリガイイベントを起動できる
	2	連続的に 4 つまでトリガイイベントを起動できる
	3	連続的に 4 つまでトリガイイベントを起動できるが、START 時にトリガイイベント 1 の監視をしない
0		アクティブな測定系
1		第 1 測定系
2		第 2 測定系

10 の桁	1 の桁	意味
3		両方の測定系

Trigger_event_1 ~ trigger_event_4:

- 1: 立ち上がりエッジプローブ 1
- 1: 立ち下がりエッジプローブ 1 オプション
- 2: 立ち上がりエッジプローブ 2 オプション
- 2: 立ち下がりエッジプローブ 2 オプション

測定メモリ : FIFO 変数の番号

測定値は機械座標系に排他的に与えられます。

MEAWA

... DO MEAWA[axis]=(, , , ,) ; 移動距離の削除なしの軸測定

移動距離の削除はシンクロナイズドアクションの中で明確にコールできます。2.4.9「DELDTG」および下記の例を参照してください。

GEO 軸および変換に関連する軸は個々にプログラムすることができます。

プログラミング :

プログラミング方法はパートプログラムで使用方法と同じです。

(注) システム変数 \$AC_MEA には、シンクロナイズドアクションからコールされる測定の有効性についての有益な情報は含まれていません。1つの軸に対しては、1度に1つの測定ジョブしかアクティブにできません。

システム変数 :

\$AA_MEAACT[axis]	軸の即時の測定ステータスを供給する
1	測定はアクティブ
0	測定は非アクティブ
\$A_PROBE[probe]	測定プローブの即時のステータスを供給する
1	プローブは切換えられる, ハイ (high)
0	プローブは切換えられない, ロー (low)

機械座標系での2つのプローブ (エンコーダ) を使用した測定値 :

\$AA_MM1[axis] トリガイイベント 1, エンコーダ 1
 \$AA_MM2[axis] トリガイイベント 1, エンコーダ 2
 \$AA_MM3[axis] トリガイイベント 2, エンコーダ 1
 \$AA_MM4[axis] トリガイイベント 2, エンコーダ 2

MEAC

... DO MEAC[axis]=(mode, No_FIFO, trigger events)

変数 \$AC_FIFO (2.3.6「FIFO 変数 (循環メモリ)」参照) は、周期的な測定プロセスから測定された値を保存するためのものです。モードおよびトリガイイベントについては上記を参照してください。

例 :

下記の例については2つのFIFOがマシンデータにセットアップされています。

マシンデータ

MD 28050: MM_NUM_R_PARAM = 300
 MD 28258: MM_NUM_AC_TIMER = 1
 MD 28260: NUM_AC_FIFO = 2 ; 2 つの FIFO
 MD 28262: START_AC_FIFO = 100 ; 第 1 の FIFO が R100 でスタートする
 MD 28264: LEN_AC_FIFO = 22 ; FIFO ごとに 22 個までの値を保存できる
 MD 28266: MODE_AC_FIFO = 0 ; 合計なし

例 1

プローブ 1 の全ての立ち上がりエッジを X0 と X100 の間のパスに記録しなければならない。エッジは 22 個までしか発生しないと考えられる。

プログラム 1 :

```
DEF INT NUMBER
DEF INT INDEX_R
N0 G0 X0
N1 MEAC[X]=( 1, 1, 1) POS[X]=100
                                ; モード = 1, 同時
                                ; FIFO 番号 = 1
                                ; トリガイイベント 1= 立ち上がりエッジ, プローブ 1
N2 STOPRE                      ; 前処理を停止する
N3 MEAC[X]=( 0)                ; 連続測定を強制終了する
N4 NUMBER= $AC_FIFO1[4]        ; FIFO 変数に転送された測定値の数
N5 NUMBER= NUMBER - 1
N6 FOR INDEX_R= 0 TO NUMBER
N7 R[INDEX_R]= $AC_FIFO1[0]    ; FIFO 要素を R0 -... に入力する
N8 ENDFOR                      ; 読取られた後 FIFO 変数は空になる
```

例 2

X0 と X100 の間のパス上のプローブ 1 の全ての立ち上がりエッジおよび立ち下がりエッジを記録しなければならない。発生するトリガイイベントの数は知らされていない。このため 1 つのシンクロナイズドアクション中の並行したオペレーションとして、測定値をフェッチし、R1 に昇順に保存しなければならない。保存された測定値の数が R0 に入力される。

プログラム 2 :

```

N0 G0 X0 ; 開始点への早送り
N1 $AC_MARKER[1]=1 ; 算術変数 R[...] 用インデックスとしてのマーカ 1
N2 ID=1 WHENEVER $AC_FIFO1[4]>=1
DO $R[$AC_MARKER[1]]=$AC_FIFO1[0] $AC_MARKER[1]=$AC_MARKER[1]+1
    ; チェックとしてのシンクロナイズドアクション :
    ; FIFO 変数に 1 つ以上の測定値が保存されている場合, FIFO から 1 番古
    ; い値を読み取り現在の R[...] に保存し, R 用のインデックスを 1 だけ増やす
N3 MEAC[X]=( 1, 1, 1, -1) POS[X]=100 ; 連続測定を起動する, X = 100 となるような動作
    ; モード = 1, 同時
    ; no_FIFO = 1
    ; トリガイベント 1= 1, 立ち上がりエッジプロープ 1
    ; トリガイベント 2= -1, 立ち下がりエッジプロープ 1
N4 MEAC[X]=(0) ; 測定を選択解除する
N5 STOPRE ; 前処理を停止する
N6 R0= $AC_MARKER[1] ; R0 に記録された値の数
    
```

例 3 :

10 回測定した後の, 明確な移動距離の削除を伴う連続測定

プログラム 3 :

```

N1 WHEN $AC_FIFO1[4]>=10
DO MEAC[X]=(0) DELDTG(X) ; シンクロナイズドアクションとしての終了条件
    ; 10 以上の測定値が FIFO 変数に保存されていた
    ; ら, 連続測定を選択解除し, 移動距離を削除する
N2 MEAC[X]=( 1,1,1,-1) G01 X100 F500 ; パートプログラムから連続測定アクティブ
    ; モード = 1, 同時
    ; No_FIFO = 1, FIFO 変数 1
    ; トリガイベント 1= 1, 立ち上がりエッジプロープ 1
    ; トリガイベント 2= -1, 立ち下がりエッジプロープ 1
N3 MEAC[X]=(0) ; 連続測定を選択解除する
N4 R0= $AC_FIFO1[4] ; 測定値の実際の数
    
```

複数の測定に伴う優先順位

1つの軸に対して、同時に複数の測定ジョブをアクティブにすることはできません。

同一の軸に対して測定ジョブが開始されたら、トリガイベントは再起動され、測定結果はリセットされます。

測定ジョブがあらかじめ起動されていない場合、「測定ジョブ停止」(モード0)がプログラムされてもシステムはまったく反応しません。

パートプログラムから開始された測定ジョブにシンクロナイズドアクションから影響を及ぼすことはできません。

パートプログラムからの測定ジョブが軸に対してすでにアクティブになっている場合、同一の軸に対してシンクロナイズドアクションから測定ジョブが開始されると、アラームが発生します。測定ジョブがシンクロナイズドアクションからすでに行なわれている最中であれば、同時にパートプログラムから測定ジョブを開始することはできません。

測定ジョブおよびステータス変更

シンクロナイズドアクションから測定ジョブが実行されたら、制御系は下記のように応答します：

ステータス	応答
運転モードの切替え	モーダルシンクロナイズドアクションによって起動された測定ジョブは、運転モードの変更の影響を受けない。この測定ジョブはブロックリミットを超えてアクティブなままになる。
RESET (リセット)	測定ジョブは強制終了される
ブロックサーチ	測定ジョブは収集されるが、プログラムされた条件が満足されるまで起動されない
REPOS	起動された測定ジョブは影響を受けない。
プログラムエンド	スタティックシンクロナイズドアクションから開始された測定ジョブはアクティブなままになる。

2.4.18 チャンネル同期用待機マーカの設定および削除

概要

チャンネルでの運転の順序の協調に関することが下記の参照に記載されています。

参照： /FB/, K1, 運転モードの種類とプログラム運転モード

この参照で説明されている機能のうち、下記のものはシンクロナイズドアクションで正當に使用できます：

待機マーカの設定

指令 SETM (マーカ番号) はパートプログラムおよびシンクロナイズドアクションのアクションセクションにプログラムできます。この指令は、指令が適用されるチャンネル (自身のチャンネル) 用のマーカ (マーカ番号) を設定します。

待機マーカの削除

指令 CLEARM (マーカ番号) はパートプログラムおよびシンクロナイズドアクションのアクションセクションにプログラムできます。この指令は、指令が適用されるチャンネル (固有のチャンネル) 用のマーカ (マーカ番号) を削除します。

2.4.19 アラームの設定／エラーリアクション

エラー状況

エラーテータスに対するリアクション方法の1つに「アラームの設定」があります。

アプリケーション：

シンクロナイズドアクションからサイクルアラームを設定するために SETAL 指令をプログラムすることができます。

エラーに対する応答として下記のリアクションをプログラムすることもできます：

- 軸を停止する（2.4.10「プログラムされた軸モーションのディスエーブル」を参照）
- 出力を設定する（2.4.2「リアルタイム変数の設定（書込み）および読取り」を参照）
- 2.4「シンクロナイズドアクション中のアクション」に記載されているその他のアクション

アラームのセット例

```
ID=67 WHENEVER $AA_IM[X1] - $AA_IM[X2] < 4.567 DO SETAL(61000)
```

；距離（軸 X1 の実際値 - 軸 X2 の実際値）が臨界値 4.567 未満になれば，
；アラームをセットする。

サイクルおよびサイクルアラーム

サイクルおよびサイクルアラームに関する情報については下記の参照を参照してください。

参照： /PGZ/, プログラミングガイド サイクル

2.5 テクノロジサイクルのコール

定義

テクノロジサイクルとは、補間サイクルで連続して実行されるアクションの順序です。2.4「シンクロナイズドアクション中のアクション」に記載されているアクションは、プログラムを構成するために組み合わせることができます。ユーザの観点からすれば、これらのプログラムはパラメータを使用しないサブプログラムです。

チャンネル内での並行処理

同一のチャンネル内で複数のテクノロジサイクルまたはアクションを同時に処理できます。これらのサイクルおよびアクションは、チャンネル内で1回の補間サイクルで並行して処理されます。

処理の順序

処理の順序に関して、ユーザは下記のオプションから最も適当な方法を選択しなければなりません：

- 1つのシンクロナイズドアクション中の複数のアクション：
全てのアクションが、条件が満足される補間サイクル中で同時に実行されます。
- テクノロジサイクルは複数のアクションから構成されています：
テクノロジサイクル中のアクションは、補間サイクル中で連続して処理されます。補間サイクルごとに1つのブロックが処理されます。単一サイクルアクションとマルチサイクルアクションとは区別しなければなりません。テクノロジサイクルは、最後のアクションが実行され終わったとき（通常は複数の補間サイクルが終わった後）に終了します。

テクノロジサイクルでの変数の割当てのような指令は、1回の補間サイクルで処理されます。その他の指令（例えば指令軸動作など。2.4.11「指令軸のスタート」を参照）は完了するのに複数の補間サイクルが必要です。機能が完了したら（例えば軸の位置決めでのイグザクトストップなど）、それ以降の補間サイクルで次のブロックが実行されます。ブロックごとに少なくとも1回の補間サイクルが必要です。1つのブロックに複数の単一サイクルアクションが含まれていれば、これらは全て1回の補間サイクル中で処理されます。図2.8にはどのアクションが単一サイクルでどのアクションがマルチサイクルかということを示すための例が示されています。

アプリケーション

テクノロジサイクルのプリケーション例の1つとして別々の軸プログラムを使用してそれぞれの軸を動かすことがあげられます。

プログラミング

テクノロジサイクルは、モーダル／スタティックシンクロナイズドアクション中の条件の関数として起動できます。

プログラムエンドは M02 / M17 / M30 / RET でプログラムします。

サーチパス

コールサーチパスはサブプログラムおよびサイクルについての場合と同じです。

例：

...

```
ID=1 EVERY $AA_IM[Y]>=10 DO AX_X      ; AX_X サブプログラム
      ; X 軸についての軸プログラム用の名称
AX_X:      ; 軸プログラム
POS[X]=$R[7] FA[X]=377
$A_OUT[1]=1
POS[X]=R10
POS[X]=-90
M30
```

(注) テクノロジサイクルが処理されている最中に条件が再び満足されても、サイクルは再び開始したりしません。テクノロジサイクルが **WHENEVER** タイプのシンクロナイズドアクションから起動されていて、関連する条件がサイクルの最後でまだ満足されていれば、サイクルは再び開始します。

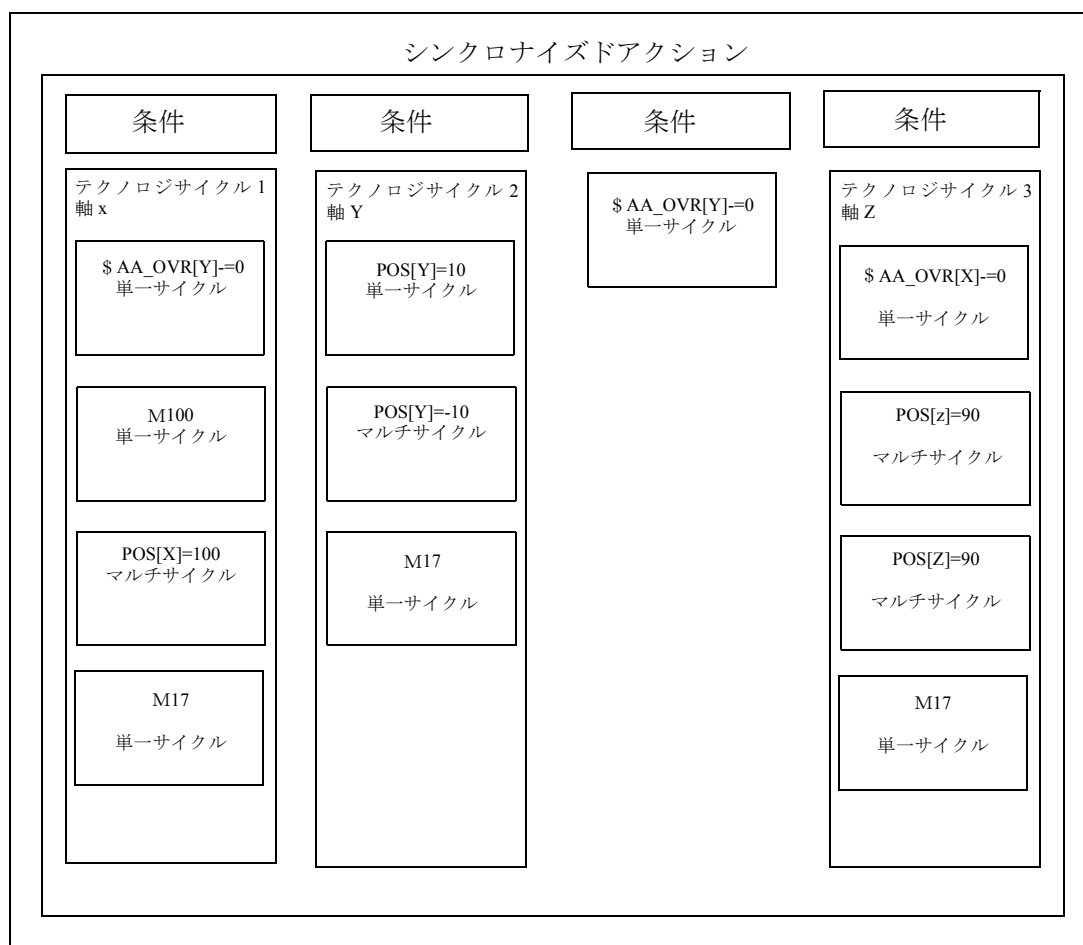


図 1.8 様々なテクノロジサイクル

例 (2) 協調させた軸モーションに対して：

デジタル NC 入力を設定することによって異なった軸プログラムを開始することができる。

メインプログラム：

...

```
ID=1 WHEN $A_IN[1]==1 DO AXIS_X  
ID=2 WHEN $A_IN[2]==1 DO AXIS_Y  
ID=3 WHEN $A_IN[3]==1 DO AA_OVR[Y]=0  
ID=4 WHEN $A_IN[4]==1 DO AXIS_Z  
M30
```

軸プログラム：

AXIS X:

```
$AA_OVR[Y]=0  
M100  
POS[X]=100  
M17
```

AXIS Y:

```
POS[Y]=10  
POS[Y]=-10  
M17
```

AXIS Z:

```
$AA_OVR[X]=0  
POS[Z]=90  
POS[Z]=-90  
M17
```

2.5.1 シンクロナイズドアクション、テクノロジサイクル、パートプログラム（および PLC）の協調

テクノロジサイクルの制御

テクノロジサイクル／シンクロナイズドアクションは、これらがアクションとしてプログラムされているシンクロナイズドアクションの識別番号によって制御されます。

協調の意味

キーワード	意味	PP	SA
	パートプログラムでの正当なコール シンクロナイズドアクション／テクノロジサイクルでの正当なコール	+	+
LOCK(ID) (ロック)	テクノロジサイクルがディスエーブルされる。 アクティブなアクションが中断させられる。		+
UNLOCK(ID) (ロック解除)	UNLOCK は中断させられた位置からテクノロジサイクルを続行させる。中断させられた位置決めオペレーションが続行される。		+
RESET(ID) (リセット)	テクノロジサイクルを強制終了する。アクティブな位置決めオペレーションが強制終了される。テクノロジサイクルは再び始まると、サイクルの第 1 ブロックから処理される。シンクロナイズドアクションのタイプに応じて、条件が再び満足されたときにアクションがもう 1 度実行される。完了された WHEN タイプのシンクロナイズドアクションは RESET 後は再び処理されない。		+
CANCEL(ID) (キャンセル)	シンクロナイズドアクションが削除される。	+	

- PLC による LOCK(ID), UNLOCK(ID), 2.6.1 「PLC を介しての制御」を参照

(注) シンクロナイズドアクションにはテクノロジサイクルコールが含まれています。ID 番号と関連するテクノロジサイクルとの間の割当てが明白であることを保証するため、同一のブロックにこれ以上アクションをプログラムすることはできません。

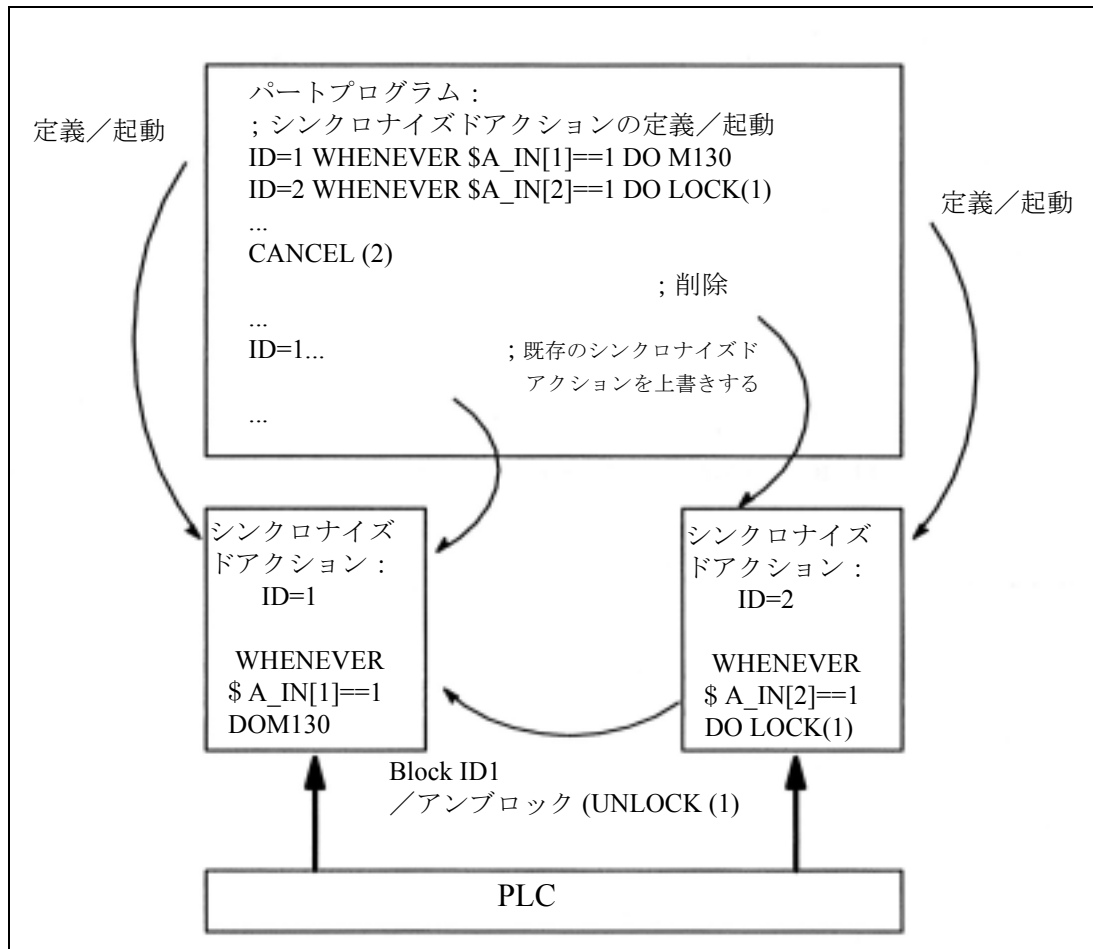


図 2.1 セットアップ/モーダルシンクロナイズドアクションのロック/削除

2.6 シンクロナイズドアクションの制御および保護

2.6.1 PLC を介しての制御

機能

モーダルシンクロナイズドアクション (ID, IDS) は、PLC からロックまたはイネーブルすることができます。

- 全てのモーダルシンクロナイズドアクションをディスエーブルする
- 個々のシンクロナイズドアクションを選択的にディスエーブルする

制御範囲

PLC は、ロックすることによって先頭の 64 個のモーダルシンクロナイズドアクションを制御することができます (ID, IDS1-64)。PLC によってロックできるシンクロナイズドアクションはインタフェースの 64 ビット配列 : DB21-30, DBB308-315 の中に保存され、NC によって "I" というタグが付けられます。保護されたシンクロナイズドアクションはロックできるアクションとしてタグが付けられることはありません。2.6.2「保護されたシンクロナイズドアクション」を参照してください。

全てのシンクロナイズドアクションをディスエーブルする

PLC アプリケーションプログラムは、すでに NC で定義され、起動に備えて保存されている全てのモーダルシンクロナイズドアクションをディスエーブル（起動しないようにロック）するために DB 21-30, DBB1 ビット 2 をセットすることができます。この場合、保護されたシンクロナイズドアクションは例外となります。2.6.2「保護されたシンクロナイズドアクション」を参照してください。

DB 21-30, DBB1 ビット 2 を 0 にセットすると、PLC による通常のロックが再びキャンセルされます。

選択的にディスエーブルする

PLC インタフェースでは、64 個の ID (1-64) それぞれにつき 1 ビットずつが確保されています (DB21-30, DBB 300 ビット 0 ~ DB21-30 DBB 307 ビット 7)。

これらの機能の初期設定は「イネーブル」（ビット = 0）です。割当てられたビットがセットされると、条件の評価および関連する機能の実行が NCK の中でディスエーブルされます。

選択的ディスエーブルのキャンセル

DB21-30, DBB 300 ビット 0 ~ DB21-30 DBB 307 ビット 7 において、モーダルシンクロナイズドアクション (ID, IDS) 番号に対応するビットを 0 にセットすると、ディスエーブルされていたアクションを PLC がイネーブルします。

参照 : /LIS/, リスト, インタフェース信号

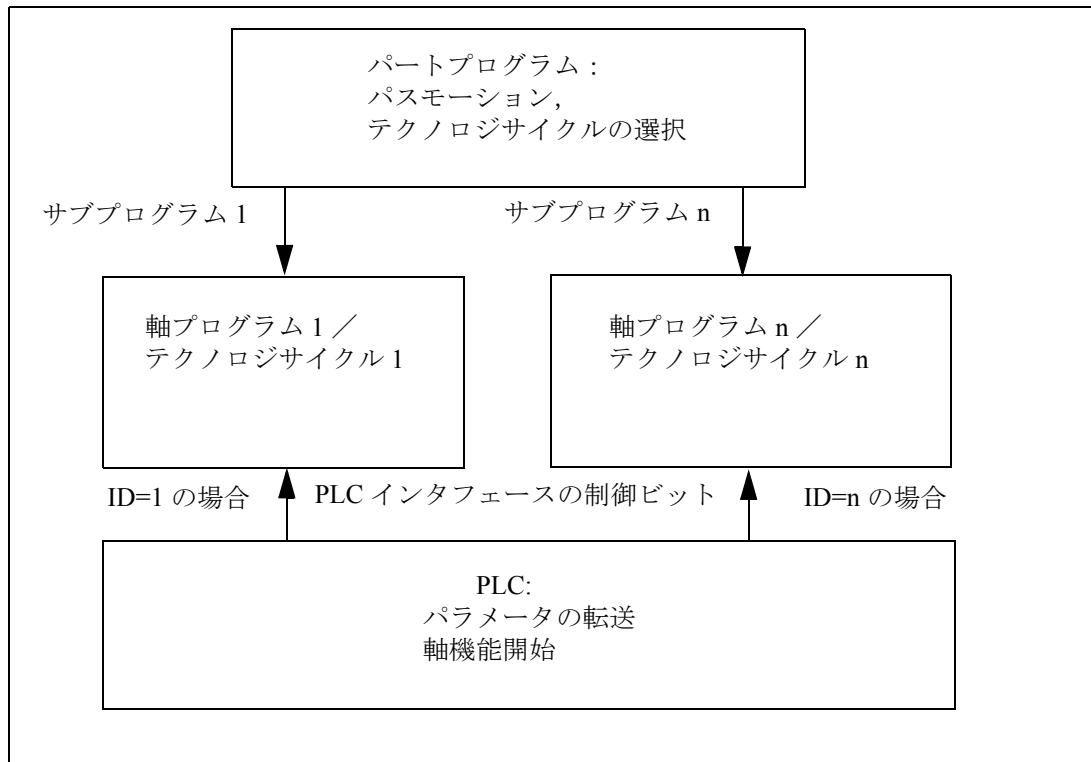


図 2.10 軸プログラム／テクノロジーサイクル

PLC データの読取り／書込み

SW バージョン 4 以降では、NCK と PLC の間で VDI インタフェースを介してパラメータを転送することによって PLC データをパートプログラムから読取ったり書込んだりできます。

これはオプションです：PLC 変数

参照： /FB/, P3, 基本的 PLC プログラム

パラメータはシンクロナイズドアクションからでもアクセスできるので、軸機能が開始する前に PLC データをパラメータ化するために NCK に転送することができます。アドレスすべきシステム変数は 2.3.7 「シンクロナイズドアクションに関連するシステム変数のリスト」に記載されています。

2.6.2 保護されたシンクロナイズドアクション

機能

保護されたシンクロナイズドアクションのエリアを定義するためにマシンデータ

MD 11500:PREVENT_SYNACT_LOCK をプログラムすることができます。保護エリア内の ID 番号が付いたシンクロナイズドアクションは、いったん定義された後は下記のことができません：

- 上書き
- 削除 (CANCEL)
- ディスエーブル (LOCK)

保護されたシンクロナイズドアクションは PLC によってディスエーブルすることもできません。これらはインタフェース中のロックできないシンクロナイズドアクションとして PLC に示されます。2.6.1「PLC を介しての制御」を参照してください。

(注) この機能性は安全統合システムに対しても使用されます。

アプリケーション

エンドユーザが、機械メーカーによって定義された特定のステータスに対するリアクションを変更してはいけません。

ゲートロジックを定義およびチェックできるようにするため、機械メーカーがシステムをスタートアップさせる時点ではシンクロナイズドアクションはまだ保護されていません。しかしシステムがエンドユーザに配布される前に、機械メーカーは保護されたシンクロナイズドアクションとして使用したシンクロナイズドアクションの範囲を宣言します。このため、エンドユーザがこの保護エリア内で独自のシンクロナイズドアクションを定義するのを防ぐことができます。

2.7 特定の運転テータスでのシンクロナイズドアクションに対する制御系の応答

2.7.1 電源オン

電源オンの間はシンクロナイズドアクションはアクティブになりません。電源オン後ただちにアクティブになる必要があるスタティックシンクロナイズドアクションは、PLCによって開始する ASUP 内で起動しなければなりません。

参照： /FB/, P3, 基本的 PLC プログラム

/FB/, K1, モードグループ, チャンネル, プログラム運転
モード

この組合せは、「全ての運転モードでの ASUP」という機能性を有する SW 4 がインストールされているという条件のもとでしか使用できません。

例：

- 適応制御
- 安全統合, シンクロナイズドアクションによって公式化されたゲートロジック

2.7.2 リセット

位置決め軸モーション

シンクロナイズドアクションから開始した全ての位置決めモーションは、NC リセットによって強制終了されます。アクティブになっているテクノロジーサイクルはリセットされます。

ID

ローカルに（つまり ID=... によって）プログラムされたシンクロナイズドアクションは NC リセットによって選択解除されます。

IDS

(IDS = ... によってプログラムされた) スタティックシンクロナイズドアクションは NC リセット後もアクティブなままになります。NC リセット後、静的アクションからモーションを再開することができます。

アクションに応じたその他のリアクション

リセット (続き)

シンクロナイズドアクション/ テクノロジサイクル	モーダルおよびノンモーダル アクティブなアクションは強制終了され、シンクロナイズドアクションはキャンセルされる	静的 (IDS) アクティブなアクションは強制終了され、テクノロジサイクルはリセットされる
軸/位置決め主軸	モーションは強制終了される	モーションは強制終了される
速度制御された主軸	\$MA_SPIND_ACTIVE_AFTER_RE-SET= =TRUE: 主軸はアクティブなままになる \$MA_SPIND_ACTIVE_AFTER_RE-SET= =FALSE: 主軸は停止する	\$MA_SPIND_ACTIVE_AFTER_RE-SET= =TRUE: 主軸はアクティブなままになる \$MA_SPIND_ACTIVE_AFTER_RE-SET= =FALSE: 主軸は停止する
マスタ値カップリング	\$MC_RESET_MODE_MASK, Bit13== 1: マスタ値カップリングはアクティブなままになる \$MC_RESET_MODE_MASK, Bit13== 0: マスタ値カップリングはキャンセルされる	\$MC_RESET_MODE_MASK, Bit13== 1: マスタ値カップリングはアクティブなままになる \$MC_RESET_MODE_MASK, Bit13== 0: マスタ値カップリングはキャンセルされる
測定オペレーション	シンクロナイズドアクションから開始された測定オペレーションは強制終了される	スタティックシンクロナイズドアクションから開始された測定オペレーションは強制終了される

2.7.3 NC 停止

スタティックシンクロナイズドアクションから開始されていたモーションは NC 停止の際もアクティブなままになります。

モーダルおよびノンモーダルアクションから開始された軸モーションは中断され、その後 NC スタートによって再開されます。速度制御された主軸はアクティブなままになります。

現在のブロックにプログラムされたシンクロナイズドアクションはアクティブなままになります。

例：

出力を設定する：... DO \$A_OUT[1] = 1

2.7.4 運転モードの変更

関連するシンクロナイズドアクションがスタティックシンクロナイズドアクションか、ローカルにプログラムされたシンクロナイズドアクションかによって応答が異なります。運転モードの変更後も、ボキャブラリワード IDS によって起動されたシンクロナイズドアクションはアクティブなままになります。その他の全シンクロナイズドアクションは運転モードの変更に応じて停止し、位置決めのために自動モードに切替えられたときに再起動します。

例：

```
N10    WHEN $A_IN[1] == 1 DO DELDTG
N20    G1          X10 Y 200 F150 POS[U]=350
```

ブロック N20 には STOP 指令が含まれています。運転モードは JOG に切り替わります。中断される前に移動距離の削除がアクティブになっていなかった場合、AUTO モードが再び選択されるとブロック N10 にプログラムされたシンクロナイズドアクションが再起動され、プログラムが続行されます。

2.7.5 プログラムの終了

プログラム終了後もスタティックシンクロナイズドアクションはアクティブなままになります。

モーダルおよびノンモーダルシンクロナイズドアクションは強制終了されます。

M30 ブロックにプログラムされたスタティックおよびモーダルシンクロナイズドアクションはアクティブなままになります。

それらは M30 ブロックの前で CANCEL を使用して強制終了できます。FCTDEF を使用してプログラムされた多項式係数はプログラム終了後もアクティブなままになります。

2.7.6 プログラム終了に対するアクティブなシンクロナイズドアクションの応答 および運転モードの変更

2.7.4 「運転モードの変更」 および 2.7.5 「プログラムの終了」を参照してください。

シンクロナイズドアクション/ テクノロジーサイクル	モーダルおよびノンモーダルアクションは 強制終了される	スタティックアクション (IDS) は アクティブなままになる
軸/位置決め主軸	M30 は軸/主軸が静止するまで遅延する	モーションは続行する
速度制御された主軸	プログラムエンド: \$MA_SPIND_ACTIVE_AFTER_RE-SET= =TRUE: 主軸はアクティブなままになる \$MA_SPIND_ACTIVE_AFTER_RE-SET= =FALSE: 主軸は停止する 主軸はモードが変更してもアクティブなままになる	主軸はアクティブなままになる
マスタ値カップリング	\$MC_RESET_MODE_MASK, Bit13==1: マスタ値カップリングはアクティブなままになる \$MC_RESET_MODE_MASK, Bit13==0: マスタ値カップリングはキャンセルされる	スタティックシンクロナイズドアクションから スタートしたカップリングはアクティブな ままになる
測定オペレーション	シンクロナイズドアクションからスタートした 測定オペレーションは強制終了される	スタティックシンクロナイズドアクションから スタートした測定オペレーションはアク ティブなままになる

2.7.7 ブロックサーチ

一般事項

ブロックサーチの間に変換されたプログラム中のシンクロナイズドアクションが収集されますが、条件は評価されません。アクションは実行されません。シンクロナイズドアクションの処理は NC スタートまで開始されません。

IDS

ボキャブラリワード IDS を使用してプログラムされ、すでにアクティブになっているシンクロナイズドアクションは、ブロックサーチの間も有効です。

多項式係数

多項式係数はブロックサーチの間に計算されて収集されます。つまり多項式係数はシステム変数に書込まれます。

2.7.8 ASUP によって中断させられたプログラム

ASUP スタート

モーダルおよび静的モーションシンクロナスアクションはアクティブなままになり、非同期サブプログラム (ASUP) 中でも有効です。

ASUP エンド

非同期サブプログラムが Repos によって続行されない場合、サブプログラム中で変更されたモーダルおよび静的モーションシンクロナスアクションはメインプログラム中で有効なままになります。

シンクロナイズドアクションからスタートした位置決めモーションは、運転モードの切替えと同じように応答します：

ノンモーダルおよびモーダルアクションからスタートしたモーションは停止され、(プログラムされていれば) Repos によって続行されます。スタティックシンクロナイズドアクションからスタートしたモーションは中断されずに続行されます。

2.7.9 REPOS

ブロックの残りでは、シンクロナイズドアクションは中断ブロックでの場合と同じように扱われます。

非同期サブプログラム中のモーダルシンクロナイズドアクションに対する変更は中断されたプログラムの中では有効ではありません。

FCTDEF を使用してプログラムされた多項式係数は ASUP および REPOS の影響を受けません。

非同期サブプログラムではコールプログラムからの係数が適用されます。非同期サブプログラムからの係数は続行コールプログラム中でも適用され続けます。

シンクロナイズドアクションからスタートした位置決めモーションが運転モードの切替えまたは割込みルーチンの開始によって中断された場合、それらの位置決めモーションは Repos に応答して続行されます。

2.7.10 アラームに対する応答

シンクロナイズドアクションによってスタートした軸および主軸モーションは、モーション停止命令を含むアラームに応答して減速します。（出力の設定というような）その他のアクションは全て実行され続けます。

シンクロナイズドアクションによってアラームが起動されると、アクションは次の補間サイクルで処理されなくなります。つまりアラームは1度だけしか出力されません。その他のアクション全ての処理は通常どおり続行されます。

テクノロジーサイクルがモーション停止を含むアラームを発生させると、関連するサイクルの処理が停止します。

2.8 構成

2. 8. 1 構成可能性

シンクロナイズドアクション要素の数

プログラム可能なシンクロナイズドアクションブロックの数は、シンクロナイズドアクション要素の構成可能な数に完全に依存します。モーションシンクロナスアクション用の保存要素（シンクロナイズドアクション要素）の数は、マシンデータ

MD 28250: MM_NUM_SYNC_ELEMENTS

で定義されます。

このデータは制御系で使用可能なブロックの数とは無関係に設定できるので、リアルタイムで評価される式の複雑さとアクションの数をフレキシブルに設定することができます。

要素の使用

下記のそれぞれの場合ごとにシンクロナイズドアクション要素が 1 つ必要です：

- 条件中の比較式
- 基本アクション
- シンクロナイズドアクションブロック

例：

下記のシンクロナイズドアクションブロック用に計 4 個の要素が必要です。

WHENEVER (\$AA_IM[x] > 10.5) OR (\$A_IN[1]==1) DO

要素 1	要素 2	要素 3

\$AC_PARAM[0]=\$AA_im[y]+1

要素 4

MD 28250: \$MC_MM_NUM_SYNC_ELEMENTS の初期設定は、SW バージョン 3 以前用に最大プリセット（16 個のシンクロナイズドアクション）を起動できるように選択されます。

(注) ユーザはシンクロナイズドアクションをプログラムしたくなければ、
約 16 KB の DRAM メモリをうかせるために MD 28250:
MM_NUM_SYNC_ELEMENTS の値を 0 にリセットすることができます。

表示

シンクロナイズドアクション用のステータス表示（2.9「診断（MMC102/MMC103 を使用した場合のみ）」を参照）は、シンクロナイズドアクション用に与えられたメモリがあとどれだけ使用できるかということを示します。このステータスはシンクロナイズドアクションから変数 \$AC_NUM_SYNC_ELEM で読取ることもできます。

アラーム

プログラムの実行中に使用可能な要素が全て使われてしまった場合、アラームが発生します。

ユーザはシンクロナイズドアクション要素の数を増やすか、自分のプログラムを修正することによって応答できます。

FCTDEF 機能の数

各ブロックについてプログラム可能な FCTDEF 機能の数は、マシンデータ
MD 28252: MM_NUM_FCTDEF_ELEMENTS
を介して構成できます。

全ての制御タイプについての初期設定は 3 です。
制御別最大値については下記の参照を参照してください。
参照： /LIS/, リスト

補間サイクル

補間レベルで必要な時間はプログラムされたシンクロナイズドアクションの数にともな
なって増加します。スタートアップさせる技術者が補間サイクルを延ばさなければなら
ない可能性があります。

補間サイクルを延ばすためのガイド値

基準として、SW バージョン 4.2 を使用した 840D / NCU3 で測定された、シンクロナ
イズドアクション内でオペレーションを行うために必要な個々の時間が下記に示されてい
ます：
他の制御タイプおよびソフトウェアバージョンについての時間は異なっている可能性が
あります。

NC 言語	必要な時間	総計
	合計	太字になっている 文字
条件が満足されていない場合の シンクロナイズドアクション用の基本ロード： WHENEVER FALSE DO \$AC_MARKER[0]=0	10 μ s	~10 μ s
変数の読取り： WHENEVER \$AA_IM[Y]>10 DO \$AC_MARKER[0]=1	11 μ s	~1 μ s
変数の書込み： DO \$R2=1	11-12 μ s	~1-2 μ s
セッティングデータの読取り／書込み： DO \$\$\$N_SW_CAM_MINUS_POS_TAB_1[0]=20	24 μ s	~14 μ s
乗算などの基本算術演算： DO \$R2=\$R2*2	22 μ s	~12 μ s
(cos などの) 三角関数： DO \$R2=COS(\$R2)	23 μ s	~13 μ s
位置決め軸モーションの開始： WHEN TRUE DO POS[z]=10	83 μ s	~73 μ s

2.9 診断

診断の機能性

下記の特殊テストツールはシンクロナイズドアクションを診断するためのものです：

- ステータス表示
- 全てのシンクロナイズドアクション変数の現在値が表示できる。
(リアルタイム変数を表示する)
- 変数特性を補間サイクルグリッドに記録できる。
(リアルタイム変数をログする)

この機能性はオペレータインタフェースの中で下記の方法で構成されます：

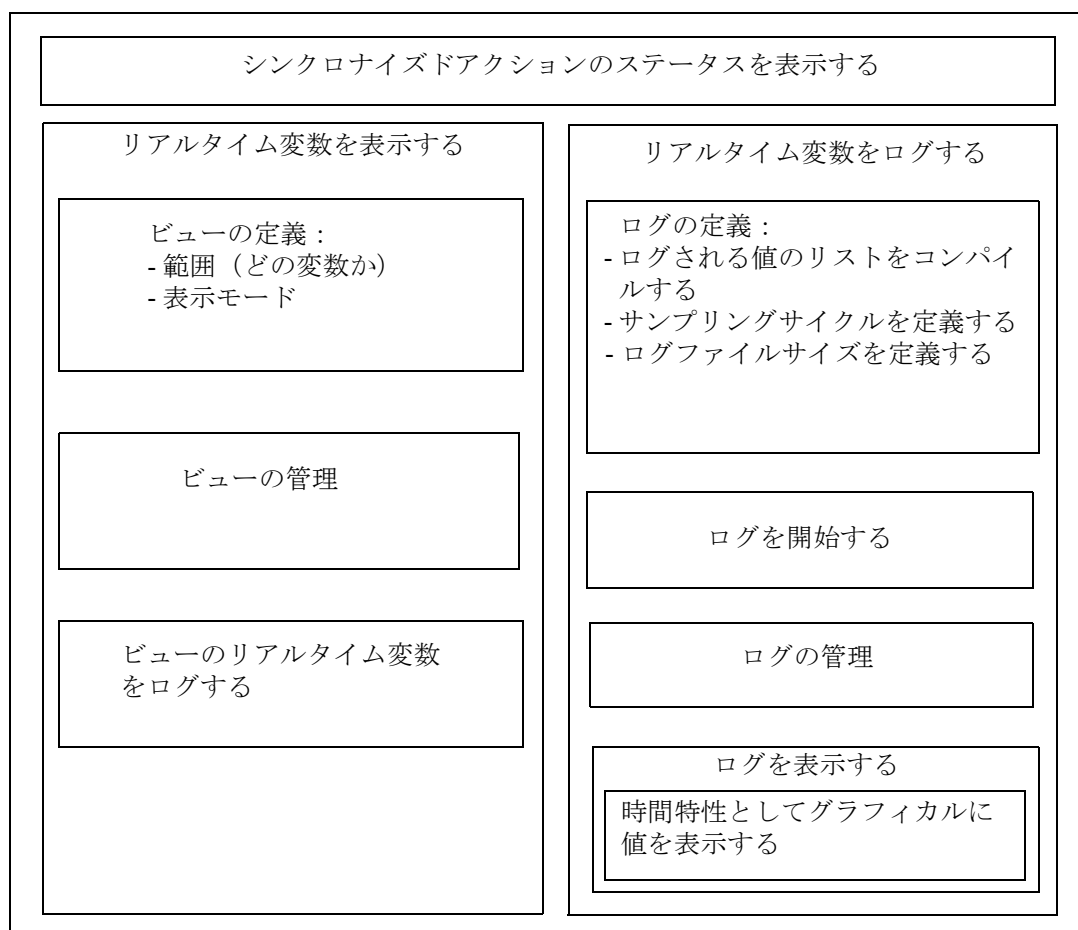


図 2.11 シンクロナイズドアクション用テストツールの機能性

これらの機能の使い方の説明が下記の参照に記載されています：

/BA/, Operator's Guide

ビューの管理

ユーザによって定義されたビューは、ユーザが選択した名称で保存し、その後再びコールすることができます。ビューに含まれた変数は修正することができます（編集ビュー）。

ビューのリアルタイム変数の表示

ビューに割当てられた値は、それらに対応するユーザに定義されたビューをコールすることによって表示されます。

2.9.3 リアルタイム変数のログ

初期状況

シンクロナイズドアクション中にイベントを追跡できるようにするためには、補間サイクル中にアクションステータスを監視する必要があります。

方法

ログ定義中に選択された値は指定されたサイクル中に定義されたサイズのログファイルに書込まれます。ログファイルの内容を表示するための特殊機能があります。

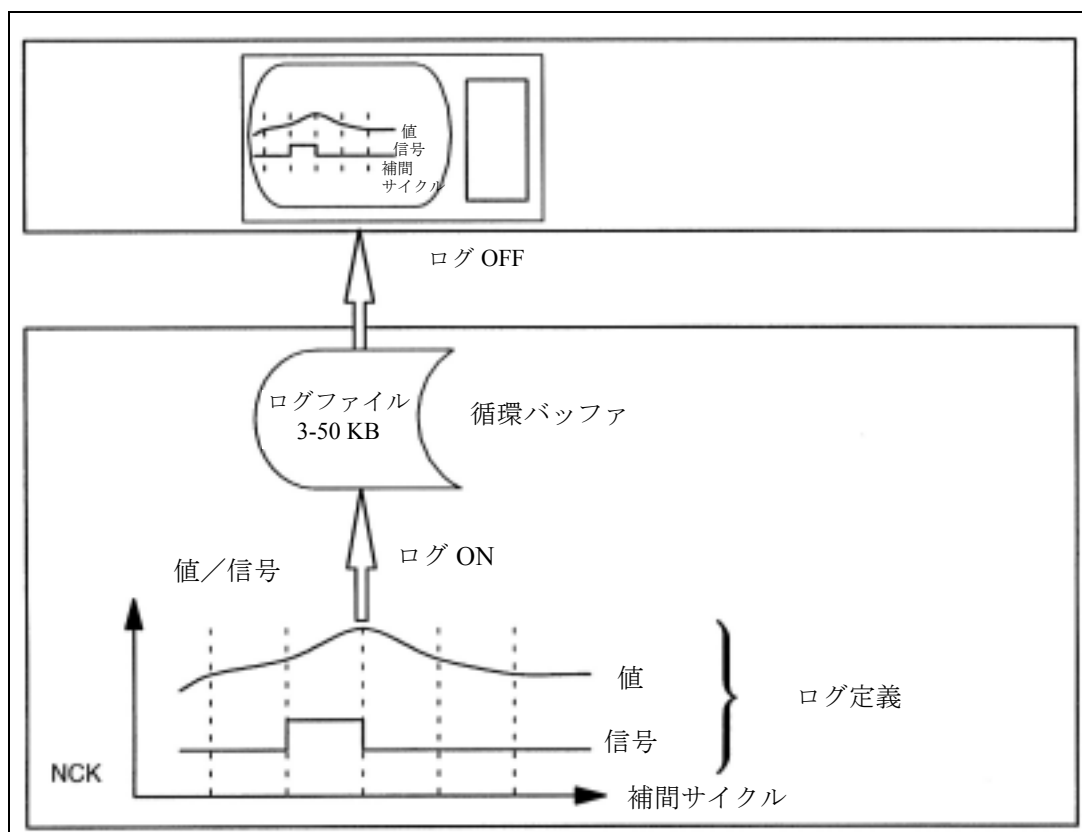


図 2.1 「リアルタイム変数のログ」プロセスの概略図

操作

ログ機能の使用に関する情報に付いては、下記の参照を参照してください：

参照： /BA/, Operator's Guide

ログ定義

ログ定義には指定された変数を 6 つまで入れることができます。これらの変数の値は指定されたサイクル中にログファイルに書込まれます。ログするために選択できる変数のリストが表示されます。サイクルは補間サイクルの倍数で選択できます。ファイルサイズは KB で選択できます。ログ定義は初期化するまでは、必要な値を得るために NCK で起動することはできません。

ログファイルのサイズ

ログファイルのサイズとして 3 KB（最小）から 50 KB（最大）の範囲の値を選択できます。

保存方法

有効なログファイルのサイズを超えてしまった場合、1 番古い入力为上書きされます。つまりファイルは循環バッファ原理で動きます。

ログの開始

初期化されたログ定義の 1 つに従ったログが、

- オペレータ入力または
- パートプログラムからのシステム変数 \$A_PROTO=1 の設定

によって開始されます。

開始する瞬間は、マシン上のオペレーションが起動されるまでログされる予定の変数が変更されないように選択されます。開始点は初期化される予定の最後のログ定義を参照します。

ログの停止

この機能は NCK でのログデータの取得を終了させます。ログされたデータが入っているファイルは保存および評価のために使用できるようになります（グラフィックログ）。ログは、

- オペレータ入力または
- パートプログラムからのシステム変数 \$A_PROTO=0 の設定

によって停止できます。

「グラフィックログ」機能

ログの測定値（6 まで）はサンプリング時間関数としてグラフィカルに表示されます。変数の名称はそれぞれの値の特性に従って下降順で指定されます。スクリーン表示は自動的に用意されます。グラフィックの選択されたエリアをズームすることができます。

ログの管理

ユーザが選択した名称で複数のログ定義を保存できます。それらは初期化および記録の開始のため、または修正および削除のために後でコールすることができます。

3 補足条件

可用性／性能範囲

「シンクロナイズドアクション」機能パッケージによって提供される性能範囲は、下記のことによって左右されます：

- 制御系のタイプ
 - HW
 - SW
- 「アクション」によって開始できる機能の可用性：
 - 標準機能
 - オプションとして使用できる機能

制御系の性能とタイプ、およびオプションとして与えられた機能が SW バージョン別のカタログに示されています：

参照： /BU/, Ordering Information, Catalog NC60.1 および
/LIS/, リスト

シンクロナイズドアクションに関連する機能は、下記のことにも左右されます：

- マシンおよびセッティングデータを含むシンクロナイズドアクションから読取り / 書き込みすることができるシステム変数のリスト
 - 使用可能なシステム変数の数は、インストールされた SW バージョンによって決まります。

特定の SW バージョンで使用できるシステム変数が下記の参照に示されています：

参照： /PGA/, プログラミングガイド 上級編, 付録
(関連 SW バージョンで)

SW バージョン 4 での拡張

SW バージョン 4 では下記の拡張が導入されました：

- シンクロナイズドアクション用の診断機能
- さらにどのくらいリアルタイム変数を使用できるか
- シンクロナイズドアクション中の複雑な条件
 - 基本算術演算
 - 機能
 - リアルタイム変数を使用したインデックス
 - セッティングおよびマシンデータへのアクセス
 - 論理演算子
- 構成可能性
 - 同時にアクティブになるシンクロナイズドアクションの数
 - シンクロナイズドアクション用の特殊変数の数
- シンクロナイズドアクションから指令軸／軸プログラム／テクノロジーサイクルを起動する

-
- シンクロナイズドアクションからの PRESET
 - シンクロナイズドアクションからのカップリングおよびカップリング軸
 - 起動
 - 停止
 - パラメータ化
 - シンクロナイズドアクションからの測定機能の使用
 - SW カム
 - 位置の再定義
 - リードタイムの再定義
 - 前処理停止なしの移動距離削除
 - 静的シンクロナイズドアクション（AUTO 以外のモードが可能）
 - シンクロナイズドアクション：
 - 上書きおよび削除に対する保護
 - 停止，続行，削除
 - テクノロジサイクルのリセット
 - PLC からのパラメータ化，起動，ディスエーブル
 - オーバーレイドされたモーション／最適化されたクリアランス制御
 - シンクロナイズドアクションからのチャンネルの協調
 - シンクロナイズドアクションからの ASUP のスタート
 - ノンモーダルな補助機能出力
 - 安全論理演算用のすべての安全統合機能。変更禁止。

SW バージョン 5 での拡張

SW バージョン 5 では下記の拡張が導入されました：

- PLC 用にタグ可能なシンクロナイズドアクション
- 追加されたリアルタイム変数の可用性
- PLC I/O へのアクセス（オプション）

4 データの説明 (MD, SD)

4.1 一般マシンデータ

11500 MD 番号	PREVENT_SYNACT_LOCK 保護されたシンクロナイズドアクション	
初期設定 : 0, 0	最小入力リミット : 0	最大入力リミット : 255
電源オン後に 変更が有効になる :	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 :	<p>保護されたシンクロナイズドアクションエリアの先頭の ID および最後の ID。</p> <p>このエリア内の ID がついたシンクロナイズドアクションは、プログラム中で上書きまたはディスエーブル (NC: CANCEL, LOCK) できない。保護されたシンクロナイズドアクションは、PLC によってディスエーブルする (LOCK) こともできない。</p> <p>代表的なアプリケーション : 機械メーカーは安全ロジックを非同期サブプログラムの中に定義する。このロジックは電源オン時に PLC によってスタートする。使用される ID の範囲はこのマシンデータを介してロックアウトされるので、機械メーカーによって統合された安全ロジックをエンドユーザが変更したり停止したりするのを防止できる。</p> <p>(注) : 保護するアクションを定義している間は、シンクロナイズドアクションの保護をキャンセルしなければならない。そうしなければ、変更のたびにロジックを再定義するために電源を入れなおさなければならない。</p> <p>0.0 という設定は、保護されているシンクロナイズドアクションはないということの意味する。つまり機能はオンにならない。値は絶対値として読取られる。任意の順番で上下の値を指定できる。</p>	

4.2 チャンネル別マシンデータ

28250 MD 番号	MM_NUM_SYNC_ELEMENTS シンクロナイズドアクション中での式のための要素の数	
初期設定 : 159	最小入力リミット : 0	最大入力リミット : 2000
電源オン後に 変更が有効になる :	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 :	<p>シンクロナイズドアクションの構成要素は、制御系の保存要素に保存される。1つのアクションには最低4つの要素が必要である。構成要素が必要とする要素数は下記の通りである :</p> <ul style="list-style-type: none"> - 条件中の各オペランド 1 要素 - 各アクション >= 1 要素 - 各割当て 2 要素 - 複雑な式の中でオペランドが1つ増えるごと 1 要素ずつ増える。 <p>1つの要素はメモリのおよそ 64 バイト分を使用する。</p>	
参照	Programming Guide Advanced	

28252 MD 番号	MM_NUM_FCTDEF_ELEMENTS FCTDEF の要素の数	
初期設定 : 3	最小入力リミット : 0	最大入力リミット : 100
電源オン後に 変更が有効になる :	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 :	シンクロナイズドアクションが機能を使用できるように、機能を制御系に保存するために保存要素が必要である。この MD はこれらの要素の数を指定する。	

28254 MD 番号	MM_NUM_AC_PARAM \$AC_PARAM のパラメータの数	
初期設定 : 50	最小入力リミット : 0	最大入力リミット : 10000
電源オン後に 変更が有効になる :	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 :	シンクロナイズドアクション用のチャンネル別 \$AC_PARAM パラメータの数	

28256 MD 番号	MM_NUM_AC_MARKER \$AC_MARKER のマーカの数	
初期設定 : 8	最小入力リミット : 0	最大入力リミット : 10000
電源オン後に 変更が有効になる :	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 :	シンクロナイズドアクション用のチャンネル別 \$AC_MARKER マーカの数	

28258 MD 番号	MM_NUM_AC_TIMER \$AC_TIMER のタイマ変数の数	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 10000
電源オン後に 変更が有効になる :	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 :	シンクロナイズドアクション用のチャンネル別 \$AC_TIMER タイマ変数の数	

28260 MD 番号	NUM_AC_FIFO \$AC_FIFO1, \$AC_FIFO2 などの変数の数	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 10
電源オン後に 変更が有効になる :	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 :	シンクロナイズドアクション用の \$AC_FIFO1 ~ \$AC_FIFO10 の FIFO 変数の数	
アプリケーション例	FIFO 変数は、例えば製品を追跡するために使用することができる : 情報 (例えば製品の長さなど) はベルトコンベア上の各部分について別々の FIFO 変数中にバッファリングできる。	
関連性	MD 28262: START_AC_FIFO	

28262 MD 番号	START_AC_FIFO R パラメータから FIFO 変数を保存する	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 10000
電源オン後に 変更が有効になる :	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 :	<p>FIFO 変数保存エリアの先頭での R パラメータの番号 番号が小さい R パラメータは全てパートプログラム内で必要に応じて使用できる。 FIFO 範囲を超えている R パラメータはパートプログラムから書込むことはできない。 FIFO エリアの先頭 : $\\$MC_MM_NUM_R_PARAM = \\$MC_START_FIFO + \\$MC_NUM_AC_FIFO * (\\$MC_LEN_AC_FIFO + 6)$ で R パラメータから FIFO 変数をすべて保存できるように、R パラメータの数をマシンデータ MD 28050: \$MC_MM_NUM_R_PARAM に設定しなければならない。 FIFO 変数の名称は \$AC_FIFO1 ~ \$AC_FIFOn である。それらはフィールドとして設定されている。 0 ~ 5 のインデックスには特別な意味がある : n=0: 変数がインデックス 0 で書込まれると、新たな値が FIFO に保存される。変数がインデックス 0 で読取られると、1 番先に保存された要素が FIFO から削除される。 n=1: 最初に保存された要素にアクセスして読取る n=2: 最後に保存された要素にアクセスして読取る n=3: FIFO 要素の総計 n=4: FIFO で使用できる要素の数 n=5: FIFO の先頭を基準にした現在の書込みインデックス</p>	

28262 MD 番号	START_AC_FIFO R パラメータから FIFO 変数を保存する
関連性	MD 28260: NUM_AC_FIFO

28264 MD 番号	LEN_AC_FIFO \$AC_FIFO ... FIFO の変数の長さ	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 10000
電源オン後に 変更が有効になる :	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 :	\$AC_FIFO1 ~ \$AC_FIFO10 の FIFO 変数の長さ。 1 つのチャンネル内の FIFO 変数はすべて同じ長さである。	
関連性	MD 28262, MD 28260	

28266 MD 番号	MODE_AC_FIFO FIFO の処理モード	
初期設定：0	最小入力リミット：0	最大入力リミット：***
電源オン後に 変更が有効になる：	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：4.1	
意味	FIFO 処理モード： Bit 0 = 1：書き込みアクセスオペレーションのたびに FIFO 要素の総計が計算される。 Bit 0 = 0: 総計は計算されない	
関連性	MD 28260: NUM_AC_FIFO	

4.3 軸／主軸別マシンデータ

30450 MD 番号	IS_CONCURRENT_POS_AX 競合する位置決め軸	
初期設定 : 0	最小入力リミット : 0	最大入力リミット :
電源オン後に 変更が有効になる :	保護レベル : 2 / 7	単位 : 1
データタイプ : Boolean	適用開始 SW バージョン : 1	
意味 :	0: 軸は競合する位置決め軸ではない 1: 軸は競合する位置決め軸であり、現在はパートプログラムによってプログラムすることはできない。	
参照	2.4.11 「指令軸のスタート」を参照	

32070 MD 番号	CORR_VELO ハンドル, 外部 ZO, 連続ドレッシング, クリアランス制御用の軸速度	
初期設定 : 100	最小入力リミット : 0	最大入力リミット : 正
電源オン後に 変更が有効になる :	保護レベル :	単位 : %
データタイプ : DWORD	適用開始 SW バージョン : 3.2	
意味 :	JOG 速度 MD: JOG_VELO, MD: JOG_VELO_RAPID, MD: JOG_REV_VELO, MD: JOG_REV_VELO_RAPID を基準にしたシンクロナイズドアクションを介しての, ハンドルオーバーライド, 外部ゼロオフセット, 連続ドレッシング, クリアランス制御 \$AA_OFF 用の軸速度の制限。 最大可能速度は MD: MAX_AX_VELO の最大速度設定と一致する。 この値になったら制限がかかる。 この最大設定を超過するとアラームが発生する。 直線または回転軸速度への変換は MD: IS_ROT_AX に従って行なわれる。	
アプリケーション例	オーバーレイされたモーションの移動についての速度制限	

32074 MD 番号	FRAME_OR_CORRPOS_NOTALLOWED フレームおよびツール長補正の有効性	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 0xFF
電源オン後に 変更が有効になる :	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 : 4.2	
意味 :	<p>シンクロナイズドアクションからスタートしたインデックス軸, PLC 軸, および指令軸についてのフレームおよびツール長補正の有効性がこのマシンデータにプログラムされる。</p> <p>Bit 0 = 0: フレームまたは補正值が有効。</p> <p>ビット割当て :</p> <p>Bit 0 = 1: プログラムされたゼロオフセット (TRANS) はインデックス軸について無効。</p> <p>Bit 1 = 1: スケール修正 (SCALE) はインデックス軸について無効。</p> <p>Bit 2 = 1: 方向転換 (MIRROR) はインデックス軸について無効。</p> <p>Bit 3 = 1: DRF オフセットは軸について無効。</p> <p>Bit 4 = 1: 外部ゼロオフセットは軸について無効。</p> <p>Bit 5 = 1: オンラインツールオフセットは軸について無効。</p> <p>Bit 6 = 1: シンクロナイズドアクションオフセットは軸について無効。</p> <p>Bit 7 = 1: コンパイルサイクルオフセットは軸について無効。</p> <p>Bit 8 = 1: 軸フレームは PLC 軸について有効。</p> <p>Bit 8 = 0: 軸フレームは PLC 軸について無効。</p> <p>(ビット評価は適合性のためである)</p> <p>Bit 9 = 1: 軸フレームは指令軸について無効。</p> <p>Bit 9 = 0: 軸フレームは指令軸について有効。</p>	

32920 MD 番号	AC_FILTER_TIME 適応制御用のフィルタ平滑定数	
初期設定 : 0.0	最小入力リミット : 0.0	最大入力リミット : 正
電源オン後に 変更が有効になる :	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 : 2.1	
意味 :	<p>メインラン変数 \$AA_LOAD, \$AA_POWER, \$AA_TORQUE および \$AA_CURR によって下記の実際ドライブ値が得られる :</p> <ul style="list-style-type: none"> - ドライブ負荷 - アクティブドライブパワー - ドライブトルク指令値 - 軸または主軸の実際の現在値 <p>測定された値は, ピーク防止用のために PT1 フィルタを介して平滑化することができる。フィルタ時定数は MD: AC_FILTER_TIME (適応制御用のフィルタ平滑定数) に定義できる。</p> <p>611-D に統合されたフィルタに加え, PT1 フィルタがドライブトルク指令値または実際の現在値について作用します。平滑化を強くきかせた値と細かく平滑化を弱くきかせた値の両方がシステムで必要な場合は, 2 つのフィルタが連続して接続される。</p> <p>平滑化時間を 0 秒と入力すると, フィルタが停止する。</p>	
この MD は次の場合は無意味	611A 付きの FM-NC	
アプリケーション例	AC 制御用の実際の現在値の平滑化	

36750 MD 番号	AA_OFF_MODE シンクロナイズドアクションを使用した軸オーバーライドに対する値の割当ての効果	
初期設定 : 0	最小入力リミット : 0	最大入力リミット : 1
電源オン後に 変更が有効になる :	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 : 3.2	
意味 :	<p>メインラン変数 \$AA_OFF を使用すると、プログラムされた軸に対するオーバーレイされたモーションをシンクロナイズドアクションの中で実行できる。これはオプションである。</p> <p>計算のモードは軸 MD: AA_OFF_MODE に定義される。</p> <p>MD: AA_OFF_MODE=0: 絶対値</p> <p>MD: AA_OFF_MODE=1: 相対値</p>	
アプリケーション例	<ul style="list-style-type: none"> ・レーザ加工用のクリアランス制御（積分） ・ジョイスティック制御された軸の移動（比例） 	

4.4 セットアップデータ

43350 MD 番号	AA_OFF_LIMIT \$AA_OFF のクリアランス制御用の補正值の上限	
初期設定 : 1.0 Ex+8	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 : 即時	保護レベル : 2 //	単位 : mm
データタイプ : DOUBLE	適用開始 SW バージョン : 4.2	
意味 :	<p>シンクロナイズドアクションから変数 \$AA_OFF を使用してプリセットできる補正值の上限</p> <p>リミット値は有効な絶対補正量に適用される。</p> <p>レーザ加工オペレーションでのクリアランス制御用のアプリケーション : レーザヘッドがメタルブランク中にはまってしまうのを防ぐために補正值が制限される。</p> <p>補正值が制限範囲内にあるかどうかを決めるためにシステム変数 \$AA_OFF_LIMIT をスキャンすることができる。</p>	

5 信号の説明

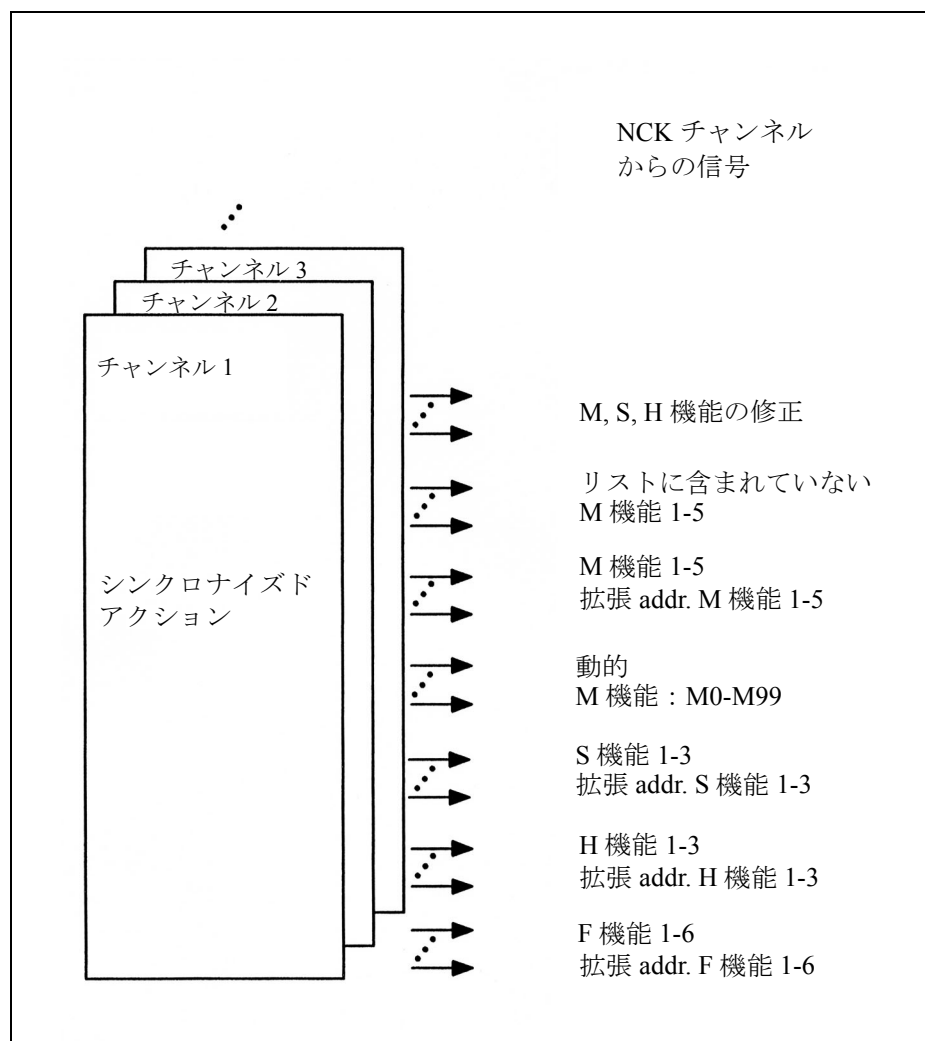


図 5.1 シンクロナイズドアクション用の PLC インタフェース信号

下記の参照に示された信号と一致するシンクロナイズドアクションからの補助機能出力によって生成された信号

参照 : /FB/, H2, PLC に対する補助機能出力

チャンネルへの信号

PLC アプリケーションプログラムは割当てられたシンクロナイズドアクションのディスエーブルを要求するのに、信号

DB 21 - 30 DBB 300 ビット 0 ～

DB 21 - 30 DBB 307 ビット 7

を使用します。この場合、DBB 300 ビット 0 は最初のモーダルシンクロナイズドアクション (ID=1/IDS=1) に対応し、DBB 307 ビット 7 は 64 番目のモーダルシンクロナイズドアクション (ID=64/IDS=64) に対応します。

(注) ディスエーブルを開始したインスタンス (NCK または PLC) しか再びディスエーブルをキャンセルすることはできません。

チャンネルからの信号

チャンネルは PLC によってどのシンクロナイズドアクションがディスエーブルできるかを PLC ユーザプログラムに指示するのに、信号

DB 21 - 30 DBB 308 ビット 0 ～

DB 21 - 30 DBB 315 ビット 7

を使用します。この場合、DBB 308 ビット 0 は最初のモーダルシンクロナイズドアクション (ID=1/IDS=1) に対応し、DBB 315 ビット 7 は 64 番目のモーダルシンクロナイズドアクション (ID=64/IDS=64) に対応します。

全てのシンクロナイズドアクションのディスエーブル

グローバル信号

DB21-30 DBB1 ビット 2

は、全てのモーダル／静的シンクロナイズドアクションを、それらのシンクロナイズドアクションが保護されていない限りディスエーブルします。

6 例

6.1 シンクロナイズドアクション中の条件例

ブロックエンドからのパス距離

ブロックエンドからの軸の距離：10 mm 未満（ワーク座標系）：

```
... WHEN $AC_DTEW <= 10 DO ...
G1 X10 Y20
```

パスエンドからの軸の距離

```
... WHEN $AA_DTEW[X] <= 10 DO ...
POS[X]= 10
```

ブロックの先頭からのパス距離

基準座標系でブロックの先頭からの距離が 20 mm を超えているパス：

```
...WHEN $AC_PLTBB >= 20 DO ...
```

機能比較条件

MCS の軸 Y の実際値が $\sin(R10)$ の 10 倍より大きい：

```
... WHEN $AA_IM[y] > 10*SIN(R10) DO ...
```

1 ステップずつの位置決め

入力 1 が設定されるたびに軸位置が 1 ステップずつ進みます。システムのコールドリストスタートができるように、入力をリセットしななければなりません。

```
G91
EVERY $A_IN[1]==1 DO POS[X]= 10
```

補間サイクルごとの OVR

プログラムされた信号が届くまでパスモーションを選択的にディスエーブルするために、補間サイクルごとに \$AC_OVR を 0 に設定しなければなりません（キーワード WHENEVER）。

```
WHENEVER $A_IN[1]==0 DO $AC_OVR= 0
```

その他のシステム変数

シンクロナイズドアクション中に読取ることができるシステム変数のリストが下記の参照および 2.3.7「シンクロナイズドアクションに関連するシステム変数のリスト」に記載されています。

参照： /PGA/, プログラミングガイド 上級編

このリストはシンクロナイズドアクションの条件中で評価できる数量の全範囲を示しています。

6.2 シンクロナイズドアクションからの SD/MD の読取りおよび書込み

グラインディングオペレーション用のインフィードおよび揺動

加工中に変更されなかった値を持つセッティングデータが通常の名称でアドレスされます（パートプログラムの場合と同様に）。

例：シンクロナイズドアクションからの揺動

NC 言語注釈

N610 ID=1 WHENEVER \$AA_IM[Z]>\$SA_OSCILL_REVERSE_POS1[Z]

DO \$AC_MARKER[1]=0

；機械座標系での往復軸の現在位置が逆転エリア 2 の先頭に達していないときは常に、インフィード軸の軸オーバーライドを 0 に設定する。

N620 ID=2 WHENEVER \$AA_IM[Z]<\$SA_OSCILL_REVERSE_POS2[Z]-6

DO \$AA_OVR[X]=0 \$AC_MARKER[0]=0

；機械座標系での往復軸の現在位置が逆転位置 1 と等しいときは常に、往復軸の軸オーバーライドを 0 に設定し、インフィード軸の軸オーバーライドを 100% に設定する（このことによってその前のシンクロナイズドアクションがキャンセルされます！）。

N630 ID=3 WHENEVER \$AA_IM[Z]==\$SA_OSCILL_REVERSE_POS1[Z]

DO \$AA_OVR[Z]=0 \$AA_OVR[X]=100

；部分インフィードの移動距離が 0 と等しいときは常に、往復軸の軸オーバーライドを 100% に設定する（このことによってその前のシンクロナイズドアクションがキャンセルされます！）。

N640 ID=4 WHENEVER \$AA_DTEPW[X]==0

DO \$AA_OVR[Z]=100 \$AC_MARKER[0]=1 \$AC_MARKER[1]=1

N650 ID=5 WHENEVER \$AC_MARKER[0]==1 DO \$AA_OVR[X]=0

N660 ID=6 WHENEVER \$AC_MARKER[1]==1 DO \$AA_OVR[X]=0

；ワーク座標系での往復軸の現在位置が逆転位置 1 と等しいときは常に、往復軸の軸オーバーライドを 100% に設定し、インフィード軸の軸オーバーライドを 0 に設定する（このことによって 2 番目のシンクロナイズドアクションがキャンセルされます！）。

N670 ID=7 WHEN \$AA_IM[Z]==\$SA_OSCILL_REVERSE_POS1[Z]

DO \$AA_OVR[Z]=100 \$AA_OVR[X]=0

加工中に（例えばオペレータ入力またはシンクロナイズドアクションを介して）値が変更される可能性のあるセッティングデータは，
\$\$\$... を付けてプログラムしなければなりません：

例：オペレータインタフェースを介して揺動位置が変更されるシンクロナイズドアクションからの揺動

N610 ID=1 WHENEVER \$AA_IM[Z]>\$\$SA_OSCILL_REVERSE_POS1[Z] DO

\$AC_MARKER[1]=0

；機械座標系での往復軸の現在位置が逆転エリア 2 の先頭に達していないときは常に，インフィード軸の軸オーバーライドを 0 に設定する。

N620 ID=2 WHENEVER \$AA_IM[Z]<\$\$SA_OSCILL_REVERSE_POS2[Z]-6

DO \$AA_OVR[X]=0 \$AC_MARKER[0]=0

；機械座標系での往復軸の現在位置が逆転位置 1 と等しいときは常に，往復軸の軸オーバーライドを 0 に設定し，インフィード軸の軸オーバーライドを 100% に設定する（このことによってその前のシンクロナイズドアクションがキャンセルされます！）。

N630 ID=3 WHENEVER \$AA_IM[Z]==\$SA_OSCILL_REVERSE_POS1[Z]

DO \$AA_OVR[Z]=0 \$AA_OVR[X]=100

；部分インフィードの移動距離が 0 と等しいときは常に，往復軸の軸オーバーライドを 100% に設定する（このことによってその前のシンクロナイズドアクションがキャンセルされます！）。

N640 ID=4 WHENEVER \$AA_DTEPW[X]==0

DO \$AA_OVR[Z]=100 \$AC_MARKER[0]=1 \$AC_MARKER[1]=1

N650 ID=5 WHENEVER \$AC_MARKER[0]==1 DO \$AA_OVR[X]=0

N660 ID=6 WHENEVER \$AC_MARKER[1]==1 DO \$AA_OVR[X]=0

；ワーク座標系での往復軸の現在位置が逆転位置 1 と等しいときは常に，往復軸の軸オーバーライドを 100% に設定し，インフィード軸の軸オーバーライドを 0 に設定する（このことによって 2 番目のシンクロナイズドアクションがキャンセルされます！）。

N670 ID=7 WHEN \$AA_IM[Z]==\$SA_OSCILL_REVERSE_POS1[Z]

DO \$AA_OVR[Z]=100 \$AA_OVR[X]=0

6.3 適用制御例

一般手順

下記の例は多項式評価機能 SYNFACT() を使用したものです。

1. 入力値と出力値との間の関係を表す（入出力値はリアルタイム変数）
2. この関係を制限付きの多項式として定義する
3. 位置オフセットを使用して：MD および SD を設定する
 - MD 36750: \$AA_OFF_MODE
 - SA 43350: \$SA_AA_OFF_LIMIT（オプション）
4. シンクロナイズドアクションで制御装置を起動する

6.3.1 可変上限値を使ったクリアランス制御

動的な上限値を持つ多項式例

クリアランス制御を行なうために、出力の上限値（\$AA_OFF，軸 V でのオーバーライド値）は、主軸オーバーライド（アナログ入力 1）の機能として変化します。多項式 1 に対する上限は、アナログ入力 2 の機能として動的に変化します。

多項式 1 はシステム変数を介して直接定義されます：

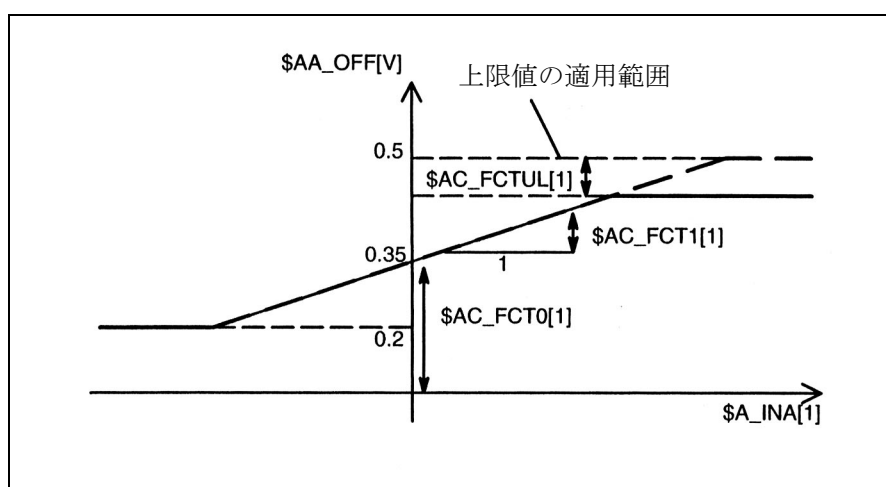


図 6.1 上限値が変化するクリアランス制御

```

$AC_FCTLL[1]=0.2          ; 下限値
$AC_FCTUL[1]=0.5          ; 上限の開始値
$AC_FCT0[1]=0.35          ; ゼロ交差点  $a_0$ 
$AC_FCT1[1]=1.5 EX-5      ; ピッチ  $a_1$ 
STOPRE                    ; 下記の（注）を参照
...
ID=1 DO $AC_FCTUL[1]=$A_INA[2]*0.1+0.35
                                ; アナログ入力 2 を介して上限値を動的に適用する、無条件
ID=2 DO SYNFACT(1, $AA_OFF[V], $A_INA[1])
                                ; オーバライドを使用したクリアランス制御
                                ; 無条件
...

```

（注）システム変数がパートプログラムに適用される場合は、ブロック同期で書き込みがおこなわれるように STOPRE をプログラムしなければなりません。下記は多項式定義用の同等の表記法です
FCTDEF(1,0.2, 0.5, 0.35, 1.5EX-5).

6.3.2 送り制御

アナログ入力電圧を使用した適用制御の例

（\$A_INA[1] を介して測定された）プロセス量は、パス（または軸）送り速度オーバライドによって与えられた加算制御係数によって 2 V に調整しなければなりません。送り速度オーバライドは ± 100 [mm/min] の制限範囲内で適用しなければなりません。

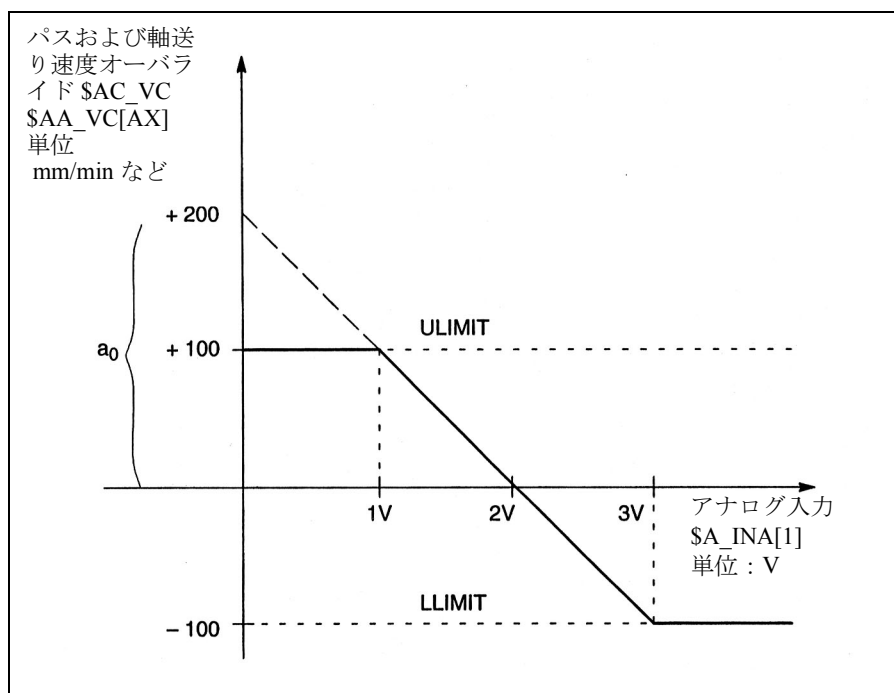


図 6.2 適用制御の説明図

係数の求め方：

$$y = f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

$$a_1 = - \frac{100 \text{ mm}}{1 \text{ min } 1 \text{ V}}$$

$a_1 = -100 \rightarrow$ 制御定数, ピッチ

$$a_0 = -(-100) \cdot 2 = 200$$

$a_2 = 0$ (2 乗成分は無視できる)

$a_3 = 0$ (3 乗成分は無視できる)

上限 = 100

下限 = -100

FCTDEF (多項式番号,

LLIMIT,

ULIMIT,

a_0 , ; y for x = 0

a_1 , ; ピッチ

a_2 , ; 2 乗成分

a_3) ; 3 乗成分

上記で求められた値を使用して, 多項式は次のように定義されます：

FCTDEF(1, -100, 100, 200, -100, 0, 0)

軸送り速度について適用制御機能を起動するのに下記のシンクロナイズドアクションが使用できます：

ID = 1 DO SYNFACT(1, \$AA_VC[X], \$A_INA[1])

パス送り速度について起動する場合：

ID = 2 DO SYNFACT(1, \$AC_VC, \$A_INA[1])

6.3.3 正規化されたパスの機能としての制御速度

乗算系の適用

正規化されたパスは入力量：\$AC_PATHN として適用されます。

0：ブロックの先頭の場合

1：ブロックエンドの場合

変動量 \$AC_OVR は、3 次多項式に従って \$AC_PATHN の機能として制御しなければなりません。オーバーライドはモーションの間 100% から 1% まで減少しなければなりません。

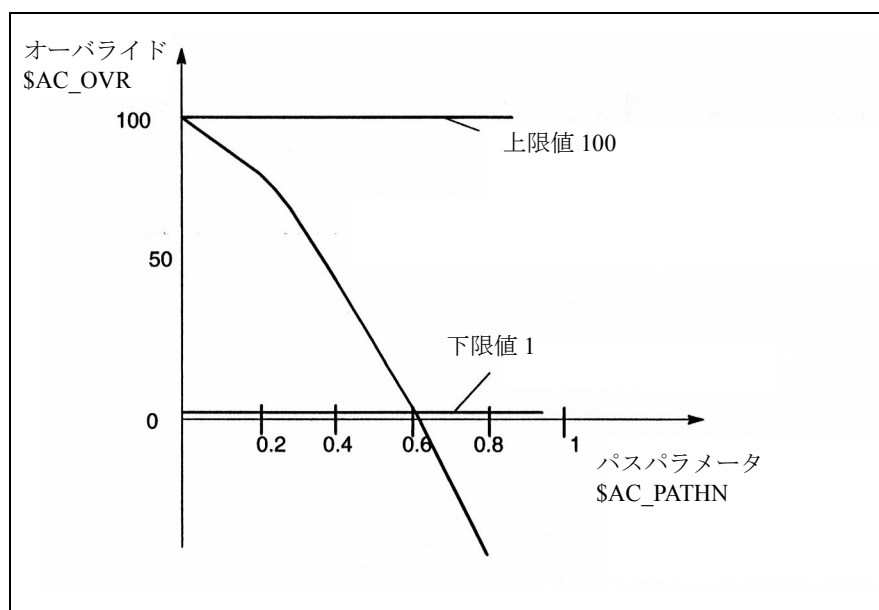


図 6.3 速度を連続的に調整する

多項式 2：

下限： 1

上限： 100

a_0 ： 100

a_1 ： -100

a_2 ： -100

a_3 ： 使用されない

これらの値を使用して多項式は下記のように定義されます：

FCTDEF(2, 1, 100, 100, -100, -100)

；パスの機能として可変オーバーライドを起動する：

ID= 1 DO SYNFACT(2, \$AC_OVR, \$AC_PATHN)

G01 X100 Y100 F1000

6.4 2 軸間の安全クリアランスの監視

タスク

軸 X1 および X2 は、ワークの積み下ろしに使用される別々に制御された 2 つの運搬装置を動かします。

軸が衝突するのを防ぐため、軸の間では安全クリアランスが保たれなければなりません。安全クリアランスが違反されると、軸 X2 は減速します。軸 X1 が再び安全クリアランスエリアからでるまでこのインタロックが適用されます。

軸 X1 が軸 X2 に向かって動き続け、その結果近い方の安全バリアを超えた場合、軸 X1 は安全位置まで移動させられます。

NC 言語

注釈

ID=1 WHENEVER \$AA_IM[X2] - \$AA_IM[X1] < 30 DO \$AA_OVR[X2]=0 ; 安全バリア
ID=2 EVERY \$AA_IM[X2] - \$AA_IM[X1] < 15 DO POS[X1]=0 ; 安全位置

6.5 実行時間を R パラメータに保存する

タスク

R パラメータ 10 で始まるパートプログラムブロックについての実行時間を保存します。

プログラム

注釈

; 下記はシンボリックプログラミングを使用しない例である：
る：

IDS=1 EVERY \$AC_TIMEC==0 DO \$AC_MARKER[0] = \$AC_MARKER[0] + 1

; ブロック変更時に R パラメータポインタを進める

IDS=2 DO \$R[10+\$AC_MARKER[0]] = \$AC_TIME

; それぞれの場合のブロックの先頭からの現在の時間を R パラメータに書込む。

; 下記はシンボリックプログラミングを使用する例である：

DEFINE INDEX AS \$AC_MARKER[0] ; シンボリックプログラミングに対する宣言

IDS=1 EVERY \$AC_TIMEC==0 DO INDEX = INDEX + 1

; ブロック変更時に R パラメータポインタを進める

IDS=2 DO \$R[10+INDEX] = \$AC_TIME

; それぞれの場合のブロックの先頭からの現在の時間を R パラメータに書込む。

6.6 連続測定時の「センタリング」

概要

ギアの歯間のすきまは連続して測定されます。すきま寸法は、すきま全部の合計と歯の数から計算されます。加工を継続するために必要な中心位置は、第1測定点プラス平均すきまサイズの1/2の位置です。測定値が補間サイクルごとに1つずつ確実に得られるような測定速度が選択されます。

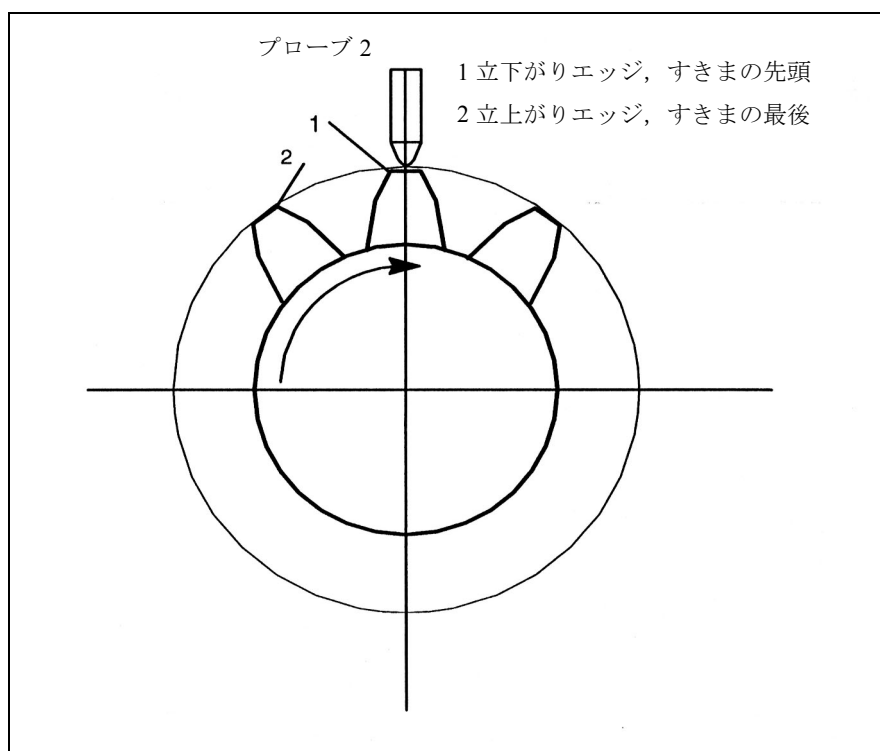


図 6.4 ギアの歯と歯の間のすきま測定の説明図

%_N_MEAC_MITTEN_MPF

; 測定値の間の差の表示がある回転軸 B (BACH) を使用して測定する

;***** ローカルユーザ変数の定義 ***

N1 DEF INT NO.TEETH ; ギアホイールの歯の数を入力する

N5 DEF REAL HYS_POS_EDGE ; ヒステリシス正エッジプローブ

N6 DEF REAL HYS_NEG_EDGE ; ヒステリシス負エッジプローブ

;***** 同期マーカ用コードネームの定義 *****

define M_TEETH as \$AC_MARKER[1] ; 計算用 ID マーカ : 歯ごとの正/負エッジ

define Z_MW as \$AC_MARKER[2] ; ID カウンタ MW FIFO を読取る

define Z_RW as \$AC_MARKER[3] ; ID カウンタ MW, 歯のすきまの計算

;***** ギア測定のための値の入力 *****

N50 NO.TEETH=26 ; 測定するギアホイールの歯の数を入力する

N70 HYS_POS_EDGE = 0.160 ; ヒステリシス正エッジプロープ

N80 HYS_NEG_EDGE = 0.140 ; ヒステリシス負エッジプロープ

開始: ; ***** 変数の割当て *****

R1=0 ; ID2 すきま寸法の計算結果

R2=0 ; ID2 すきま全部の合計の計算結果

R3=0 ; 読み込まれた第 1 要素の内容

R4=0 ; R4 が歯と歯の間の距離に対応している

R5=0 ; 計算されたすきま位置, 最終結果

R6=1 ; MOV を使用して ID 3 BACH を起動する

R7=1 ; ID 5 MEAC を起動する

M_TEETH=NO.TEETH*2 ; 歯ごとに ID 負/正エッジを計算する

Z_MW=0 ; 歯の数まで ID カウンタ MW FIFO を読取る

Z_RW=2 ; ID カウンタが歯のすきまの差を計算をする

R13=HYS_POS_EDGE ; 計算レジスタ内のヒステリシス

R14=HYS_NEG_EDGE ; 計算レジスタ内のヒステリシス

;***** 軸の移動, 測定, 計算 *****

N100 MEAC[BACH]=(0) ; 測定ジョブをリセットする

; FIFO1[4] 変数をリセットし, 定義された測定トレースを保証する

N105 \$AC_FIFO1[4]=0 ; FIFO1 をリセットする

STOPRE

;***** 歯の数に達するまで FIFO を読取る *****

; FIFO1 がまだ空になっていなくて, 全部の歯を測定し終えていない場合, 測定値を
FIFO 変数からシンクロナイズドアクションパラメータへ移動させ, 測定値カウンタを増
加させる。

ID=1 WHENEVER (\$AC_FIFO1[4]>=1) AND (Z_MW<M_TEETH)

DO \$AC_PARAM[0+Z_MW]=\$AC_FIFO1[0] Z_MW=Z_MW+1

; 2 つの測定値が使用できれば計算を始める。

; すきま寸法のみとすきま全部の測定値カウンタを 2 つだけ増加させる。

ID=2 WHENEVER (Z_MW>=Z_RW) AND (Z_RW<M_TEETH)

DO \$R1=(\$AC_PARAM[-1+Z_RW]-\$R13)-(\$AC_PARAM[-2+Z_RW]-\$R14)

Z_RW=Z_RW+2 \$R2=\$R2+\$R1


```
;***** MOV を使用して軸 BACH を連続回転用回転軸として起動する *****
WAITP(BACH)
ID=3 EVERY $R6==1 DO MOV[BACH]=1 FA[BACH]=1000      ; 起動
ID=4 EVERY $R6==0 DO MOV[BACH]=0                    ; 停止

; 連続して測定し、FIFO 1、MT2 負、MT2 正エッジに保存する
; 2 つの歯の立下がりエッジと立上がりエッジの間の距離を測定する (プローブ 2)
N310 ID=5 WHEN $R7==1 DO MEAC[BACH]=(2, 1, -2, 2)
N320 ID=6 WHEN (Z_MW>=M_TEETH) DO MEAC[BACH]=(0)    ; 測定を強制終了する
M00
STOPRE

;***** FIFO 値のフェッチおよび保存 ***
N400 R3=$AC_PARAM[0]      ; 読込まれる第 1 要素の内容
                           ; FIFO1[4] 変数をリセットし、次の測定ジョブ用に定義
                           ; された測定トレースを保証する

N500 $AC_FIFO1[4]=0

;***** 個々の歯の間の差を計算する
N510 R4=R2/(NO.TEETH)/1000 ; R4 が歯と歯の間の平均距離と一致する
                           ; 新しい SW バージョンでは "/1000" が省略されている

;***** 中心位置の計算 *****
N520 R3=R3/1000 ; 角度に変換する第 1 測定位置
N530 R3=R3 MOD 360 ; 第 1 測定点のモジュロ
N540 R5=(R3-R14)+(R4/2) ; すきま位置を計算する
M00
stopre
R6=0                      ; BACH 軸の回転を停止する
gotob start
M30
```

6.7 シンクロナイズドアクションによっての軸カップリング

6.7.1 マスタ軸へのカップリング

タスクの割当て

周期カーブテーブルは多項式セグメントを使用して定義されます。マスタ軸動作およびマスタ軸とスレーブ軸との間のカップリングプロセスの起動／停止は算術変数によって制御されます。

%_N_KOP_SINUS_MPF

N5 R1=1 ; ID 1, 2 カップリングを起動／停止する
; LEADON (CACB, BACH)

N6 R2=1 ; ID 3, 4 マスタ軸動作のオン／オフ : MOV BACH

N7 R5=36000 ; BACH 送り速度／分

N8 STOPRE

;**** 多項式セグメントを使用して番号 4 の周期テーブルを定義する ****

N10 CTABDEF (YGEO,XGEO,4,1)

N16 G1 F1200 XGEO=0.000 YGEO=0.000 ; 最初の位置にアプローチする

N17 POLY PO[XGEO]=(79.944,3.420,0.210) PO[YGEO]=(24.634,0.871,-9.670)

N18 PO[XGEO]=(116.059,0.749,-0.656) PO[YGEO]=(22.429,-5.201,0.345)

N19 PO[XGEO]=(243.941,-17.234,11.489) PO[YGEO]=(-22.429,-58.844,39.229)

N20 PO[XGEO]=(280.056,1.220,-0.656) PO[YGEO]=(-24.634,4.165,0.345)

N21 PO[XGEO]=(360.000,-4.050,0.210) PO[YGEO]=(0.000,28.139,-9.670)

N22 CTABEND ; **** テーブル定義終了 ****

; マスタ軸およびカップリング軸を早送りモードで基本設定まで移動させる

N80 G0 BACH=0 CACH=0 ; チャンネル軸の名称

N50 LEADOF(CACH,BACH) ; 既存のカップリングをオフにする

N235 ;***** 軸 CACH 用のカップリングモーションの起動 *****

N240 WAITP(CACH) ; 軸をチャンネルと同期化させる

N245 ID=1 EVERY \$R1==1 DO LEADON(CACH, BACH, 4) ; テーブル 4 を使用したカ
ップリング

N250 ID=2 EVERY \$R1==0 DO LEADOF(CACH, BACH) ; カップリングを停止する

N265 WAITP(BACH)

N270 ID=3 EVERY \$R2==1 DO MOV[BACH]=1 FA[BACH]=R5 ; R5 での送り速度で
マスタ軸を継続的に回転
させる

N275 ID=4 EVERY \$R2==0 DO MOV[BACH]=0 ; マスタ軸を停止する

N280 M00

N285 STOPRE

N290 R1=0 ; カップリング条件を停止する

N295 R2=0 ; マスタ軸の回転用の条件を停止する

N300 R5=180 ; BACH 用の新たな送り速度

N305 M30

6.7.2 マスタ軸カップリングによる非円形グラインディング

タスクの割当て

軸 CACH 上で回転する非円形ワークはグラインディングによって加工しなければなりません。グラインディングホイールとワークとの間の距離は軸 XACH によって制御され、ワークの回転角度に左右されます。回転角度と割当てられた動作との間の相互関係はカーブテーブル 2 で定義されます。ワークはカーブテーブル 1 で定義されたワーク輪郭によって決められる速度で動かなければなりません。

ソリューション

CACH はマスタ軸として指定されます。CACH は下記のことを制御します。

- テーブル 2 を介しての軸 XACH の補正モーション
- テーブル 1 を介しての「ソフトウェア軸」CASW

軸 CACH の軸オーバーライドは軸 CASW の実際値によって決まるので、必要とされる軸 CACH の輪郭に依存した速度が提供されます。

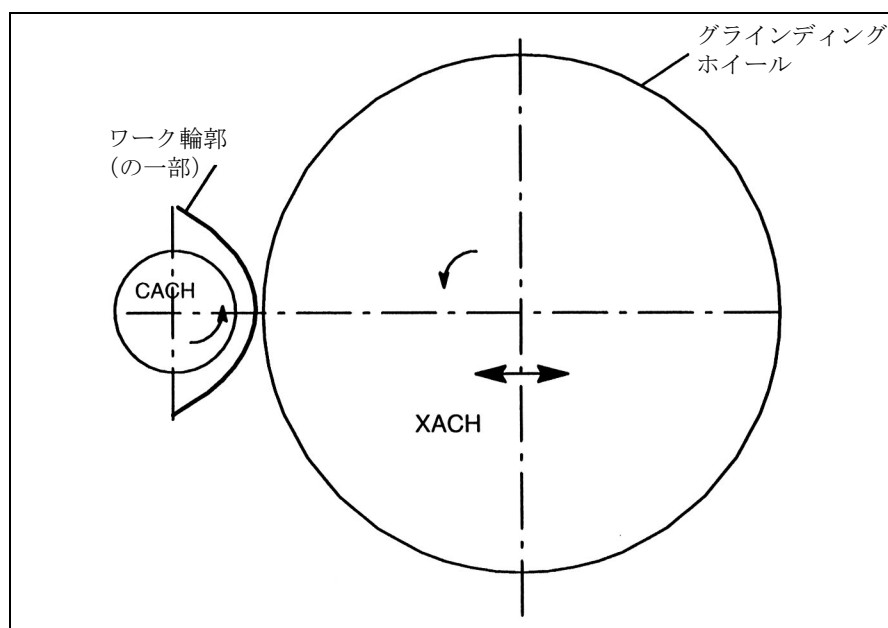


図 6.5 非円形輪郭グラインディングの説明図

```
%_N_CURV_TABS_SPF
```

```
PROC CURV_TABS
```

```
N160 ; ***** テーブル 1 オーバライドの定義 *****
```

```
N165 CTABDEF(CASW,CACH,1,1) ; 周期テーブル 1
```

```
N170 CACH=0 CASW=10
```

```
N175 CACH=90 CASW=10
```

```
N180 CACH=180 CASW=100
```

```
N185 CACH=350 CASW=10
```

```
N190 CACH=359.999 CASW=10
```

```
N195 CTABEND
```

N160 ; **** テーブル 2 XACH の直線補正モーションの定義 ****

CTABDEF(YGEO,XGEO,2,1) ; 周期テーブル 2

N16 XGEO=0.000 YGEO=0.000

N16 XGEO=0.001 YGEO=0.000

N17 POLY PO[XGEO]=(116.000,0.024,0.012) PO[YGEO]=(4.251,0.067,-0.828)

N18 PO[XGEO]=(244.000,0.072,-0.048) PO[YGEO]=(4.251,-2.937)

N19 PO[XGEO]=(359.999,-0.060,0.012) PO[YGEO]=(0.000,-2.415,0.828)

N16 XGEO=360.000 YGEO=0.000

N20 CTABEND

M17

%_N_NONCIRC_MPF

; 非円形加工用のカップリング軸グルーピング

; XACH はグライディングホイールのインフィード軸である

; CACH は回転軸およびマスタ軸としてのワーク軸である

; アプリケーション : 非円形の輪郭をグランドする

; テーブル 1 は CACH の位置の関数として軸 CACH 用にオーバーライドをミラーリングする

; スクラッチング用にハンドルインフィードを使用した XGEO 軸のオーバーライド

N100 DRFOF ; ハンドルオーバーライドを選択解除する

N200 MSG("select DRF, (handwheel 1 active) and select INCREMENT.== handwheel override ACTIVE")

N300 M00

N500 MSG() ; メッセージをリセットする

N600 R2=1 ; LEADON テーブル 2, ID=3/4 を使用して起動,
CACH 対 XACH

N700 R3=1 ; LEADON テーブル 1, ID=5/6 を使用して起動,
CACH 対 CASW, オーバライド

N800 R4=1 ; 連続回転用回転軸 CACH, ID=7/8 によって始まる

N900 R5=36000 ; FA[CACH] 連続回転用回転軸の速度

N1100 STOPRE

N1200 ; ***** マスタ軸を追従軸に対しセットする *****

; マスタおよび追従軸を開始位置まで移動させる

N1300 G0 XGEO=0 CASW=10 CACH=0

N1400 LEADOF(XACH,CACH) ; XACH 補正動作のカップリング OFF

N1500 LEADOF(CASW,CACH) ; CASW オーバライドテーブルのカップリング OFF

N1600 CURV_TABS ; テーブル定義を有するサブプログラム

N1700 ; ***** LEADON 補正モーション XACH を起動する *****

N1800 WAITP(XGEO) ; 軸をチャンネルと同期させる

N1900 ID=3 EVERY \$R2=1 DO LEADON(XACH,CACH,2)

N2000 ID=4 EVERY \$R2=0 DO LEADOF(XACH,CACH)

```

N2100                                ;***** LEADON CASW オーバライドテーブルを起動する ****
N2200 WAITP(CASW)
N2300 ID=5 EVERY $R3==1 DO LEADON(CASW,CACH,1)
                                ; マスタ軸 CACH の CTAB カップリング ON
N2400 ID=6 EVERY $R3==0 DO LEADOF(CASW,CACH)
                                ; マスタ軸 CACH の CTAB カップリング OFF

N2500                                ; ** ID 10 * を使用して位置 CASW から CASH
                                ; オーバライドを制御する

N2700 ID=11 DO $$AA_OVR[CACH]=$AA_IM[CASW]          ; 「軸位置」 CASW を OVR
                                                    CACH に割当てて

N2900 WAITP(CACH)

N3000 ID=7 EVERY $R4==1 DO MOV[CACH]=1 FA[CACH]=R5
                                ; 連続回転用回転軸としてスタートする
N3100 ID=8 EVERY $R4==0 DO MOV[CACH]=0          ; 連続回転用回転軸として停止する

N3200 STOPRE

N3300 R90=$AA_COUP_ACT[CASW]                ; チェックのための CASW 用のカップ
                                                    リングのステータス

N3400 MSG("activate CASW override table with LEADON "<<R90<<"," go to END with NC
START")

N3500 M00                                ;***** NC ストップ *****
N3600 MSG()
N3700 STOPRE                ; 前処理停止
N3800 R1=0                ; 連続回転用回転軸としての ID=2 CASW 軸を使用してを停止する
N3900 R2=0                ; ID=6 を使用して FA XACH およびマスタ軸 CACH を LEADOF する
N4000 R3=0                ; ID=7/8 を使用して TAB1 CASW を LEADOF する, CACH 対
                        CASW, オーバライドテーブル
N4100 R4=0                ; 連続回転用回転軸として軸を停止する, ID=4 CACH
N4200 M30

```

拡張オプション

上記の例は下記の構成要素によって拡張できます：

- グラインディングホイールまたはワークを同一シャフト（カムシャフト）上で、1つの非円形オペレーションから次の非円形オペレーションに移すための Z 軸の導入
- 例えばインレット用およびアウトレット用というようにカムが異なった輪郭を有する場合のテーブル間の切換え
ID = ... < 条件 > DO LEADOF(XACH, CACH) LEADON(XACH, CACH, < 新たなテーブル番号 >)
- 2.4.6 「オンラインツールオフセット FTOC」に基づくオンラインツールオフセットを使用したグラインディングホイールのドレッシング。

6.7.3 即時の切断

タスクの割当て

切断ツールの運転エリアまで入ってきている連続して通る押出し加工材料は、同じ長さでいくつかに切断しなければなりません。

X 軸：押出し加工材料が動く方向を表す軸。WCS

X1 軸：押出し加工材料のマシン軸，MCS

Y 軸：切断ツールが押出し加工材料を「追跡する」方向を表す軸

この例では、切断ツールのインフィードは PLC によって制御されることを前提としています。PLC インタフェースの信号を、押出し加工材料と切断ツールを同期させるかどうかを決めるために評価することができます。

アクション

カップリングの起動，LEADON

カップリングの停止，LEADOF

実際値の設定，PRESETON

NC プログラム	注釈
%_N_SCHERE1_MPF	
;\$PATH=/_N_WKS_DIR/_N_DEMOFBE_WPD	
N100 R3=1500	; 切断される部分の長さ
N200 R2=100000 R13=R2/300	
N300 R4=100000	
N400 R6=30	; Y 軸のスタート位置
N500 R1=1	; ベルト軸用のスタート条件
N600 LEADOF(Y,X)	; 既存のカップリングの削除
N700 CTABDEF(Y,X,1,0)	; テーブルの定義
N800 X=30 Y=30	; 値のペア
N900 X=R13 Y=R13	
N1000 X=2*R13 Y=30	
N1100 CTABEND	; テーブルの定義の終了
N1200 PRESETON(X1,0)	; 開始時の PRESET
N1300 Y=R6 G0	; Y 軸のスタート位置
	; 軸は直線
N1400 ID=1 WHENEVER \$AA_IW[X]>\$R3 DO PRESETON(X1,0)	; 長さ R3 移動した後の PRESET,
	; 次の切断を新たにスタートする
N1500 WAITP(Y)	
N1800 ID=6 EVERY \$AA_IM[X]<10 DO LEADON(Y,X,1)	; X < 10 の場合, テーブル 1 を使用して Y を X にカップリングする
N1900 ID=10 EVERY \$AA_IM[X]>\$R3-30 DO LEADOF(Y,X)	; X が切断長さの先頭から 30 を超えた時点でカップリングを停止する
N2000 WAITP(X)	
N2100 ID=7 WHEN \$R1==1 DO MOV[X]=1 FA[X]=\$R4	; 連続した動作で押出し加工材料軸をセットする
N2200 M30	

6.8 テクノロジサイクル「主軸の位置決め」

アプリケーション

ツール変更を開始する主軸は PLC プログラムと共同して、下記のことを行なわなければなりません

- 開始位置まで移動する、または
- 挿入予定のツールと同じ位置に位置決めされる

2.4.11「軸指令のスタート」、2.6.1「PLC を介しての制御」を参照してください。

協調

PLC および NCK は、SW バージョン 4 以降の標準データを使用して協調させられます (2.3.7「シンクロナイズドアクションに関連するシステム変数のリスト」を参照してください)。

- \$A_DBB[0] 開始位置への 1 回の移動
- \$A_DBB[1] 目標位置への 1 移動
- \$A_DBW[1] 位置値 +/-, PLC が最短ルートを計算する

シンクロナイズドアクション

%_N_MAIN_MPF

...

IDS=1 EVERY \$A_DBB[0]==1 DO NULL_POS ; PLC によって \$A_DBB[0] が設定されたら、開始位置へ移動する

IDS=2 EVERY \$A_DBB[1]==1 DO ZIEL_POS ; PLC によって \$A_DBB[1] が設定されたら、主軸を \$A_DBW[1] に保存された位置まで移動させる

...

テクノロジサイクル NULL_POS

%_N_NULL_POS_SPF

PROC NULL_POS

SPOS=0 ; ツール変更用ドライブを開始位置まで移動させる

\$A_DBB[0]=0 ; NCK で実行される開始位置

テクノロジサイクル ZIEL_POS

%_N_ZIEL_POS_SPF

PROC ZIEL_POS

SPOS=IC(\$A_DBW[1]) ; PLC によって主軸を \$A_DBW[1] に保存されている位置値まで移動させる、インクリメンタル指令

\$A_DBB[1]=0 ; NCK で実行される目標位置

6.9 TCC/BAZ エリアでのシンクロナイズドアクション

概要

下記の図はツール変更サイクルの構造を示しています。

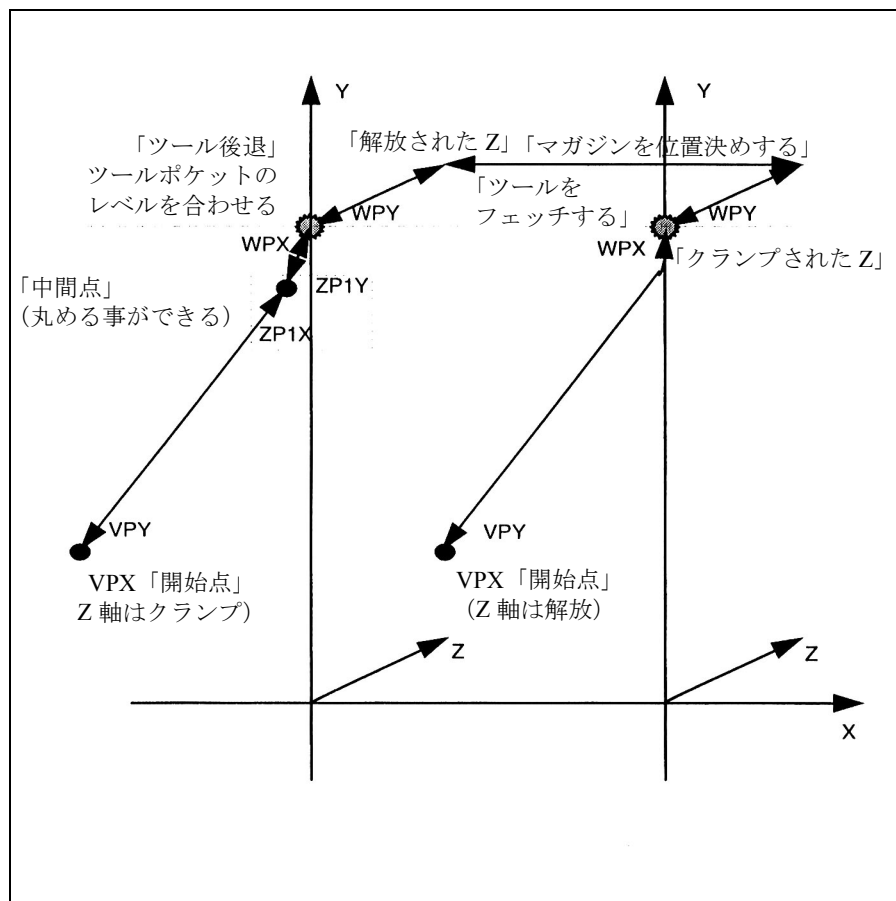
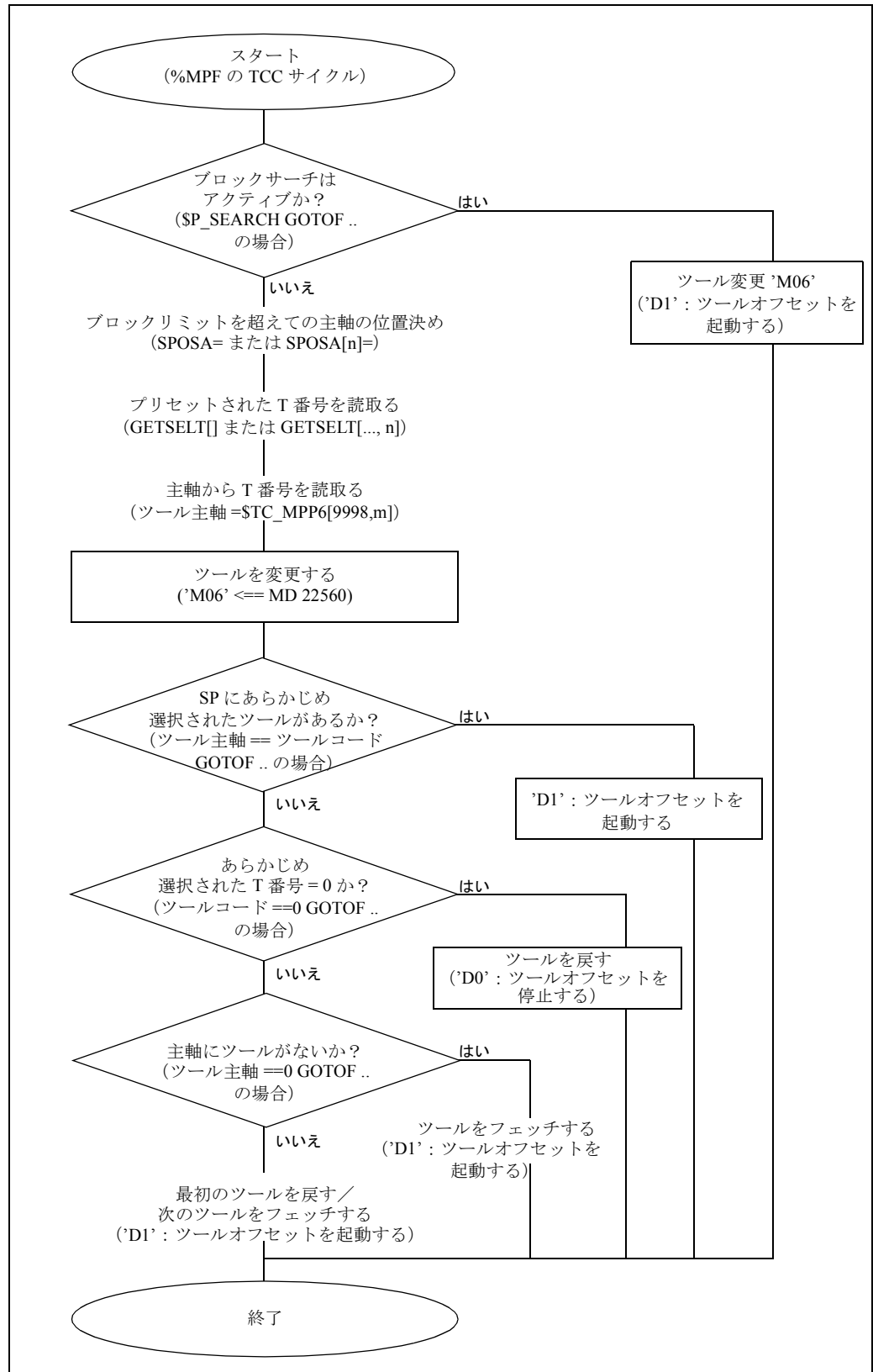


図 6.6 ツール変更サイクルの順番

フローチャート用フローチャート



NC プログラム

注釈

```

% N_WZW_SPF
; $PATH=/_N_SPF_DIR
N10 DEF INT toolcode, ToolSpindle
N15 WHEN $AC_PATHN<10 DO $AC_MARKER[0]=0 $AC_MARKER[1]=0 $AC_MARKER[2]=0
N20 ID=3 WHENEVER $A_IN[9]==TRUE DO $AC_MARKER[1]=1 ; マガジン軸が移動するとき
                                                    marker to = 1
N25 ID=4 WHENEVER $A_IN[10]==TRUE DO $AC_MARKER[2]=1 ; マガジン軸が移動するとき
                                                    marker to = 1
N30 IF $P_SEARCH GOTOF tcc_preprocessing ; ブロックサーチはアクティブか? ->
N35 SPOSA=0 D0
N40 GETSELT(toolcode) ; あらかじめ選択された T 番号を読む
N45 ToolSpindle=$TC_MPP6[9998,1] ; 主軸のツールを読む
N50 M06
N55 IF ToolSpindle== ToolCode GOTOF tool_in_spindle IF ToolCode==0 GOTOF return1 IF
ToolSpindle==0 GOTOF
fetch1
; ***** ツールをフェッチする, および戻す *****
return1 fetch1:
N65 WHENEVER $AA_VACTM[C2]<>0 DO $AC_MARKER[1]=1 ; マガジン軸が移動すると
                                                    marker to = 1
N70 G01 G40 G53 G64 G90 X=magazine1VPX Y= magazine1VPY Z= magazine1Zclamped F70000
M=QU(120)
M=QU(123) M=QU(9)
N75 WHENEVER $AA_STAT[S1]<>4 DO $AC_OVR=0 ; 主軸定位置
N80 WHENEVER $AA_VACTM[C2]<>0 DO $AC_MARKER[1]=1 ; マガジン軸移動要求
N85 WHENEVER $AC_MARKER[1]==0 DO $AC_OVR=0 ; 軸が移動しない場合 override=0
N90 WHENEVER $AA_STAT[C2]<>4 DO $AC_OVR=0 ; マガジン軸が定位置 (微) にない
                                                    場合 override=0
N95 WHENEVER $AA_DTEB[C2]>0 DO $AC_OVR=0 ; マガジン軸の移動距離
                                                    > 0 の場合 override=0

N100 G53 G64 X=magazine1ZP1X Y=magazine1ZP1Y F60000
N105 G53 G64 X=magazine1WPX Y=magazine1WPY F60000
N110 M20 ; ツールを解放する
N115 G53 G64 Z=MR_magazine1Zreleased F40000
N120 WHENEVER $AA_VACTM[C2]<>0 DO $AC_MARKER[2]=1;
N125 WHENEVER $AC_MARKER[2]==0 DO $AC_OVR=0
N130 WHENEVER $AA_STAT[C2]<>4 DO $AC_OVR=0
N135 WHENEVER $AA_DTEB[C2]>0 DO $AC_OVR=0
N140 G53 G64 Z= magazine1Zclamped F40000
N145 M18 ; ツールをクランプする
N150 WHEN $AC_PATHN<10 DO M=QU(150) M=QU(121) ; 条件が常に満足される
N155 G53 G64 X=magazine1VPX Y=magazine1VPY F60000 D1 M17
; ***** ツールを戻す *****
return1:
N160 WHENEVER $AA_VACTM[C2]<>0 DO $AC_MARKER[1]=1
N165 G01 G40 G53 G64 G90 X=magazine1VPX Y=magazine1VPY Z=magazine1Zclamped F70000
M=QU(120)
M=QU(123) M=QU(9)
N170 WHENEVER $AA_STAT[S1]<>4 DO $AC_OVR=0
N175 WHENEVER $AA_VACTM[C2]<>0 DO $AC_MARKER[1]=1
N180 WHENEVER $AC_MARKER[1]==0 DO $AC_OVR=0
N185 WHENEVER $AA_STAT[C2]<>4 DO $AC_OVR=0
N190 WHENEVER $AA_DTEB[C2]>0 DO $AC_OVR=0
N195 G53 G64 X=magazine1ZP1X Y=magazine1ZP1Y F60000
N200 G53 G64 X=magazine1WPX Y=magazine1WPY F60000
N205 M20 ; ツールを解放する
N210 G53 G64 Z=magazine1Zreleased F40000
N215 G53 G64 X=magazine1VPX Y=magazine1VPY F60000 M=QU(150) M=QU(121) D0 M17

```

***** ツールをフェッチする *****

fetch1:

N220 WHENEVER \$AA_VACTM[C2]<>0 DO \$AC_MARKER[2]=1

N225 G01 G40 G53 G64 G90 X=magazine1VPX Y=magazine1VPY Z=magazine1Zreleased F70000

M=QU(120)

M=QU(123) M=QU(9)

N230 G53 G64 X=magazine1WPX Y=magazine1WPY F60000

N235 WHENEVER \$AA_STAT[S1]<>4 DO \$AC_OVR=0

N240 WHENEVER \$AA_VACTM[C2]<>0 DO \$AC_MARKER[2]=1

N245 WHENEVER \$AC_MARKER[2]==0 DO \$AC_OVR=0

N250 WHENEVER \$AA_STAT[C2]<>4 DO \$AC_OVR=0

N255 WHENEVER \$AA_DTEB[C2]>0 DO \$AC_OVR=0

N260 G53 G64 Z=magazine1Zclamped F40000

N265 M18 ; ツールをクランプする

N270 G53 G64 X= magazine1VPX Y= magazine1VPY F60000 M=QU(150) M=QU(121) D1 M17

***** 主軸のツール *****

tool_in_spindle:

N275 M=QU(121) D1 M17

***** ブロックサーチ *****

tc_block search:

N280 STOPRE

N285 D0

N290 M06

N295 D1 M17

7 データフィールド, リスト

7.1 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			
21-30	300.0-	モーダルシンクロナイズドアクション ID または IDS 1-	
21-30	307.7	64 ディスエーブル。NCK チャンネルへの要求	
21-30	308.0-	モーダルシンクロナイズドアクション ID または IDS 1-	
21-30	315.7	64 ディスエーブル。NCK チャンネルによる確認	

7.2 マシンデータ

番号	識別子	名称	参照
一般 (\$MN_...)			
11110	AUZFU_GROUP_SPEC	補助機能グループの指定	H2
11500	PREVENT_SYNACT_LOCK	保護されたシンクロナイズドアクション	
チャンネル別 (\$MC_...)			
28250	MM_NUM_SYNC_PREVENTS	シンクロナイズドアクションでの式用の要素の数	
28252	MM_NUM_FCTDEF_PREVENTS	FCTDEF 要素の数	
28254	MM_NUM_AC_PARAM	\$AC_PARAM パラメータの数	
28256	MM_NUM_AC_MARKER	\$AC_MARKER マーカの数	
28258	MM_NUM_AC_TIMER	\$AC_TIMER タイマ変数の数	
28260	NUM_AC_FIFO	\$AC_FIFO1, \$AC_FIFO2 変数の数	
28262	START_AC_FIFO	R パラメータから FIFO 変数を保存する	
28264	LEN_AC_FIFO	\$AC_FIFO ~ FIFO までの変数の長さ	
28266	MODE_AC_FIFO	FIFO 処理モード	

軸別 (\$MA_...)			
30450	IS_CONCURRENT_POS_AX	競合する位置決め軸	P2
32060	POS_AX_VELO	位置決め軸速度用の初期設定	P2
32070	CORR_VELO	ハンドル, 外部 ZO, 連続ドレッシング, クリアランス制御用の軸速度 (SW 3 以降)	H1
32074	FRAME_OR_CORRPOS_NOTALLOWED	フレームおよびツール長補正の有効性	
32920	AC_FILTER_TIME	適応制御用のフィルタ平滑化時間定数 (SW 2 以降)	
36750	AA_OFF_MODE	シンクロナイズドアクション用の軸オー バライドに対する値の割当ての効果 (SW3 以降)	
37200	COUPLE_POS_TOL_COARSE	「同期 (粗)」用のスレッシュホールド値	S3
37210	COUPLE_POS_TOL_FINE	「同期 (微)」用のスレッシュホールド値	S3
セッティングデータ (\$SA_...)			
43300	ASSIGN_FEED_PER_REV_SOURCE	位置決め軸/主軸用の回転送り速度	V1
43350	AA_OF_LIMIT	\$AA_OFF クリアランス制御用のオフセッ ト値の上限	
43400	WORKAREA_PLUS_ENABLE	正の方向の作業エリア制限	A3

7.3 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: /DA/, 診断ガイド

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>