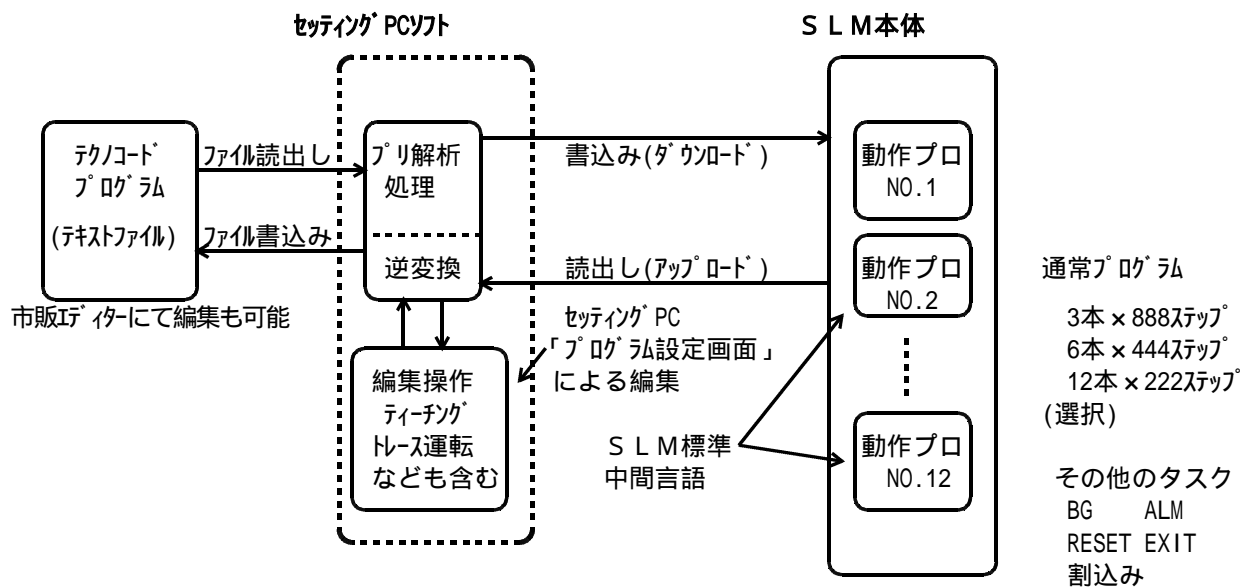


## 5. 運転プログラムの処理と各種運転方法

### 5 - 1 運転プログラムの編集

S L Mの運転プログラムはP C (パソコン)にて作成し、P C内で「プリ解析処理」をおこない、S L Mへダウンロードします。また、アップロードして、読み出すこともできます。



## 5 - 2 運転プログラムの事例

1.6

運転プログラムは、ユーザ殿にて作成していただくもので、文字形式という意味で「テキストプログラム」とも呼びます。

これら運転プログラムで使用する各々の命令については、6項「運転プログラム詳細」を参照下さい。

出荷CDの中には、G言語とテクノ言語の運転プログラム例があります。

また、それらを説明した資料が、テクノHP会員コーナーにあります。あわせてご覧ください。

サンプル運転プログラム説明書 TB04-0422

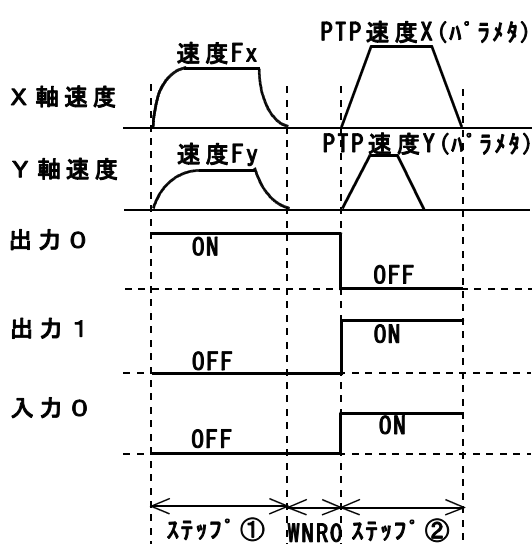
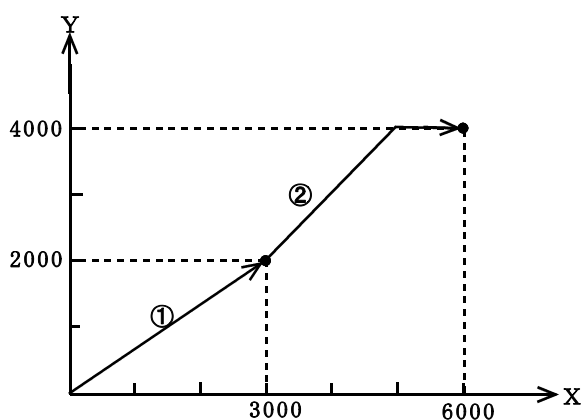
### (1) 運転プログラム例・1

プログラム運転では、ステップの各軸の移動と、入/出力の制御を同時に制御できます。

軸の移動と出力信号の様子を示します。

```

LIN X3000 Y2000 F1000 ONR0 OFR1
PTP X3000 Y2000 WNR0 OFR0 ONR1
    
```



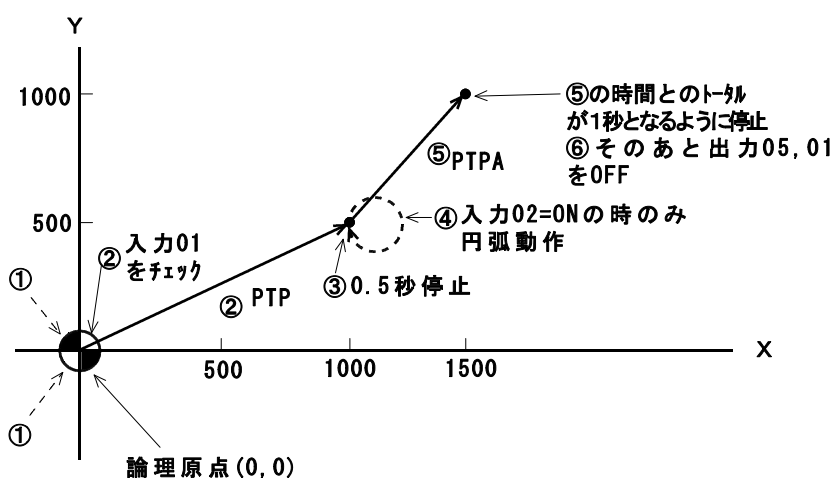
$$\begin{aligned} \text{速度}F_x &= 1000 \times \frac{3000}{\sqrt{3000^2+2000^2}} \\ &= 832\text{pps} \\ \text{速度}F_y &= 1000 \times \frac{2000}{\sqrt{3000^2+2000^2}} \\ &= 555\text{pps} \end{aligned}$$

WNR0 = R0入力がONするまで待つ

## (2) 運転プログラム例・2

ステップ NO.	各ステップ	
1	PTPA X0 Y0 ;	どの位置においても論理座標の(0,0)へ移動
2	PTPA X1000 Y500 WR01 ONR01 ;	入力01が ONの間停止 OFFしたら、出力01をONして(1000,500)へ移動
3	TIMO.5 ;	0.5秒停止
4	CALL SUB1 JNR02 ;	入力02=ONなら、SUB1をCALL。入力02=OFFなら何もせず次へ
5	PTPA X1500 Y1000 TIM1.0 ;	(1500,1000)へ移動し、トータル1秒となるまで停止
6	OFR05 OFR01 ;	出力 05,01を OFF
7	END ;	運転プログラム終了
	SUB1 :	サブルーチン名称ラベル(ステップ NO. は付きません)
8	CIRR X0 Y0 I100 J0 F1000 ;	円弧動作(プリ解析処理により複数ステップ に分割されます)
n 1	END ;	サブルーチンEND

1 円弧精度によりステップNO.は変わります。



1 行の動作を**ステップ**と呼びます。

S L M内部では各行に、**ステップNO.**がつけられます。(プログラム編集時には付けません)

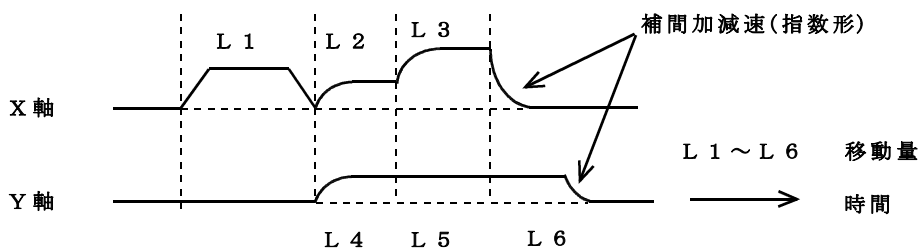
## (3) 運転プログラム例・3

巻線機の「からげ動作」や組立・実装における「はめ込み動作」など、緻密で、なめらか、かつ高速な動作を要求される場合があります。

このような場合は、各軸の移動(速度)のタイミングチャートを作成して、プログラミングする方法を推奨します。

補間時定数の設定により、速度変化部分の丸み具合は変わります。

PTP	X L1					} パス動作によりステップ間は なめらかに連続
LIN	X L2	Y L4	F			
LIN	X L3	Y L5	F			
LIN		Y L6	F			



## 5 - 3 自動運転におけるステップ動作

### 5 - 3 - 1 自動運転

S L M内には、複数の運転プログラムを保存できます。  
自動運転モードにおいて、運転プログラムのNO.を指定して起動することで、その運転プログラムの各ステップを順番に実行していきます。

停止の条件	a. 通常の停止	運転プログラム終了 リセット入力
	b. 途中停止(再開可)	ストップ入力
	c. ステップ間停止	<b>シングル</b> でシングルステップモードにすると 1ステップ実行した所で停止します。
	d. 強制終了	汎用入力条件で、EXITが成立したとき。 E R            E N R
	e. アラーム停止	アラームが発生した時(停止&サーボオフ)

### 5 - 3 - 2 シングル運転(順行)

「自動運転モード」で シングルステップモードの時に、スタート入力するたびに、1ステップずつ実行します。

停止の条件は、連続運転と同様です。

### 5 - 3 - 3 ステップ間停止

シングル運転で、1ステップを完了して停止している状態です。

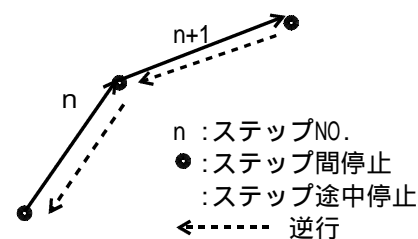
逆行操作やティーチング操作は、ステップ間停止の時に可能です。

### 5 - 3 - 4 ステップ途中停止

そのステップの途中で停止している状態です。

停止操作では、一般的にステップ途中停止

になります

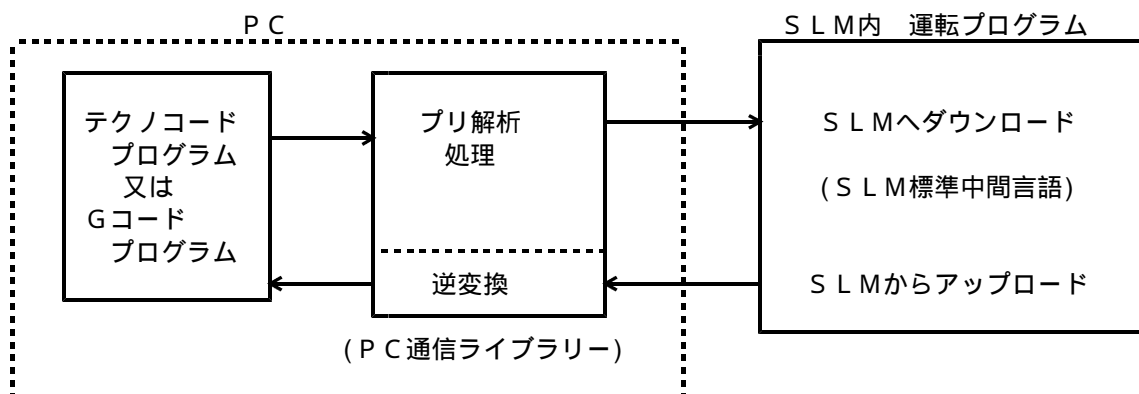


### 5 - 3 - 5 逆行(ステップ逆行)

移動命令(PTP,LIN)に対しては、逆行動作が可能です。(他の命令ではできません)

## 5 - 4 プリ解析処理

「テクノコード」「Gコード」とも運転プログラムは、PC側(PC通信ライブラリー)にて解析をおこない、中間言語レベルの特殊コードとなって SLMへローディングされます。

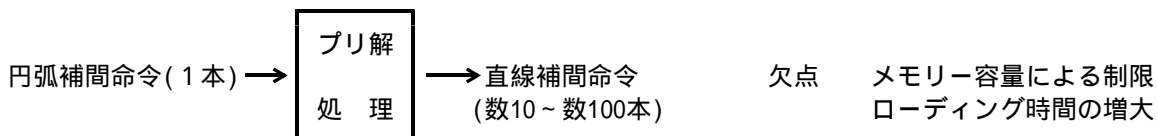


### 【プリ解析処理の機能と特徴】

- |                       |   |
|-----------------------|---|
| a. フォーマットエラーのチェック     | SLMへローディングする前に、運転プログラムの文法上のミスを発見できます。                               |
| b. 中間言語(バイナリーコード)への展開 | SLMの標準的なフォーマットに変換されます。これにより、Tコード/Gコード/他のコード各々の表現形式の違いは吸収されて、統一されます。 |
| c. プリ解加減速             | 直線補間や円弧補間に対して、加減速の指定ができます。  |
| d. 円弧補間の直線分解          | 円弧指令を、微少な直線指令の連続に展開します。円弧精度を指定することにより、分割の度合いが異なります。                 |
| e. 特殊指令への対応(専用化対応)    | 直線補間や円弧補間以外の指令方式に対しても、比較的柔軟に対応できます。                                 |
| f. SLM内部処理の負荷軽減       | SLM内で処理しやすい形式に変換します。  |

### 【プリ解円弧補間の問題点】

プリ解析は、「プリ解円弧によるステップ数の増大」という問題点を持っています。詳細は、補足資料1を参照下さい。

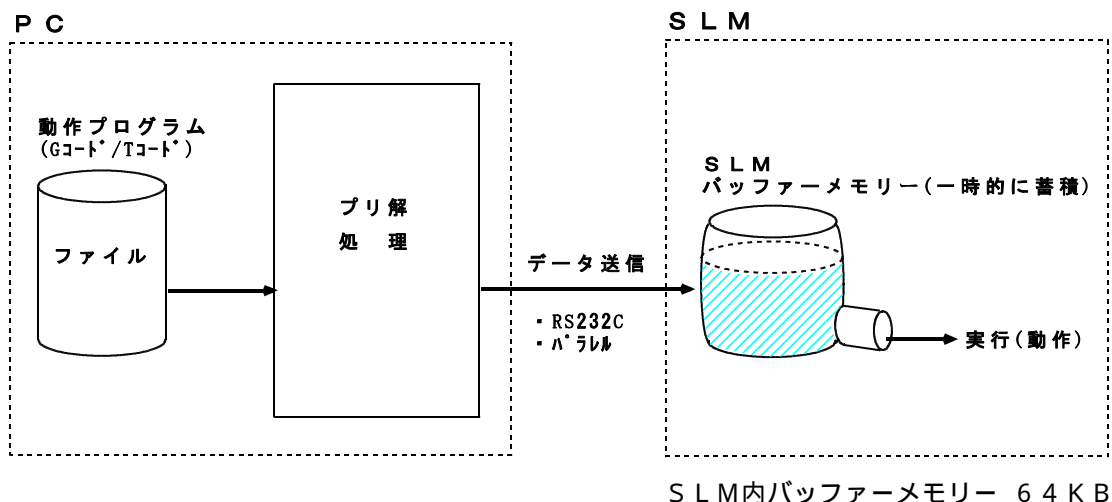


### 【解決する手段】

- |                  |  |
|------------------|--|
| a. 円弧精度を荒くする。    | (1, 10, 100)                               |
| b. 円弧命令の使用を制限する。 | SLMの運転プログラムステップ数(222、444、888ステップ)に応じて使用する。 |
| c. DNC運転方式にする。   | DNC方式では、メモリー容量の制限はありません。非常に有効な手段です。        |
| d. 内部円弧にする       |  |

## 5 - 5 DNC(Direct NC)方式 <オプション>

DNC方式では、PC側で処理された運転プログラムをSLMへ転送しながら、実動作をおこなう方式です。またセッティングPC上で現在実行しているプログラムのステップ番号を表示することができます。



### 【DNC方式の特徴】

- 運転プログラムのトータルのステップ数に制限はありません。
- SLM内のバッファメモリーに蓄積しながら運転することにより、データ通信が瞬間的に止まったり、遅くなっても問題ありません。
- プリ解析、データ通信、実動作が同時進行することで、解析 実行までの待ち時間を最短にできます。
- 特にバス接続による超高速DNCは、金型や光学系の連続微小ブロック加工に最適です。

### 【DNC方式の注意点】

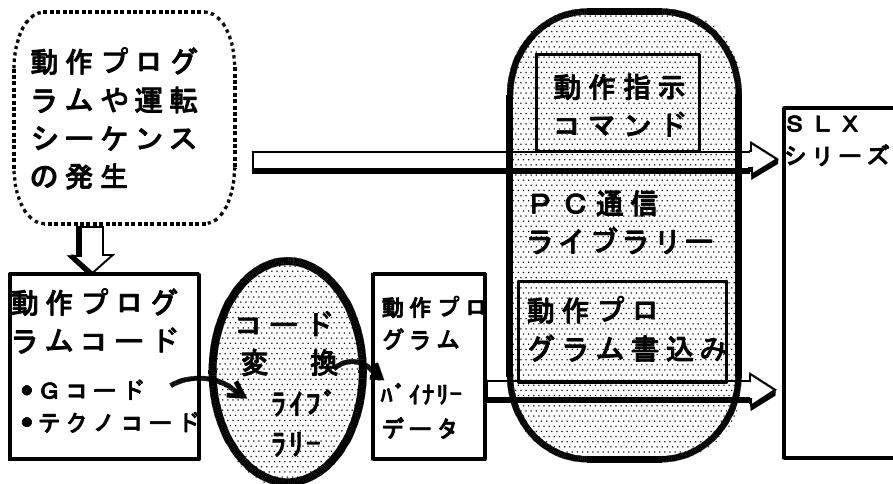
- 一般にSLM内でサブルーチンコールをすることは不可です。
- スタンドアロンのSLMにおいても、PCが常時必要となります。
- RS-232方式では、送信時間の影響から連続微小ステップの処理時間(ブロック処理時間)が数10msec/ステップに制限されます。


## 5 - 6 単独コマンド運転

PC側アプリケーションから、位置決め/補間命令/入出力命令などの単独コマンドを発行して動作させます。

画像処理と組み合わせたアライメント装置や、インプロセス計測による移動指令の修正などをおこなうシステムに有効な方法です。

単独コマンドとメモリ-運転やDNC運転を組み合わせれば、ユーザアプリにおいて、スケジュール機能が実現できます。



 当社提供のライブラリソフト

### 【PC通信ライブラリの使い方の例】

	[ ユーザソフト ]		[ SLM ]
プログラム運転	<b>データ送信</b>	運転プログラム書込み	⇒ プログラムNO.を指定して保存
	<b>動作指示(コマンド)</b>	実行プログラム選択	⇒ プログラムNO.を選択
		プログラム実行開始	⇒ 運転スタート
コマンド運転	<b>動作指示(コマンド)</b>	原点復帰	⇒ 原点復帰動作を実行
		PTP位置決め	⇒ 位置決め動作(1軸~4軸)
		補間指令	⇒ 直線補間動作(1軸~4軸)
		出力直接制御	⇒ 指定された出力信号のON/OFF
状態監視	<b>データ通信</b>	ポジションステータス読出し	⇒ 各軸位置(指令、機械) ⇒ その他内部ステータス
		入出力状態読出し	⇒
			⇒ 現在の全入出力信号のON/OFF情報

1.4

補足資料 1 . 円弧プリ解析処理(ステップ数 / 処理時間)

・ステップ数(変換後の直線ステップ数)。

$$N = \frac{k \cdot \sqrt{r}}{\cos^{-1}\left(\frac{r-r}{r}\right) \times 2}$$

： 円弧の移動角度 [rad]  
 r : 円弧の半径 [pls]  
 : 円弧の精度 [pls]

・ステップ数の概算法

r=10000、 =10 の時 N=72 ( 1 円周の時)  
 Nはrのルートに比例し、 のルートに反比例する

r が半分(1/2)の時、  $N=72 \times \sqrt{\frac{1}{2}} = 51$

が 5 倍の時、  $N=72 \times \frac{1}{\sqrt{5}} = 33$

・実際の変換とその所要時間

r		2 パルス	5 パルス	10 パルス
10000	ステップ数	159	101	72
	処理時間	0.79msec	0.54msec	0.39msec
50000	ステップ数	353	224	159
	処理時間	1.60msec	1.04msec	0.78msec
100000	ステップ数	498	316	224
	処理時間	2.25msec	1.42msec	1.04msec

= 360° ( 1 周円)

処理時間 (msec)

・ PC-AT AMD DURON / 1GHz

上記は一周円の場合なので、半円ではこの半分、1/4円では1/4となります。

又、セッティング PC では、円弧精度 ( ) は 1 0° 単位でしか設定できません。

(セッティング PC の制限です。ユーザアプリケーションでは 1 パルス単位で設定できます。)

1.1

補足資料 2 . プリ解円弧精度と送り速度上限値の関係

円弧補間において、その半径とプリ解円弧精度の指定によって送り速度の上限値が決まります。(次式)

$$\text{送り速度上限値 [pps]} = 2 \times r \times \sin\left(\cos^{-1}\frac{r-r}{r}\right) \times \frac{1000}{RTC}$$

r : 円弧の半径 [pls]

: 円弧の精度 [pls]

RTC : 制御周期 [msec]

標準は 4 msec ですが、高速なシステムでは、2 msec の場合があります

プリ解円弧精度と半径による飽和速度 [KPPS] の関係表 - 1 (制御周期 8 msec 時)

円弧精度	半 径				
	1000	2000	5000	10000	20000
1	11.2	15.8	25.0	35.4	50.0
2	15.8	22.4	35.4	50.0	70.7
3	19.4	27.4	43.3	61.2	86.6
4	22.3	31.6	50.0	70.7	100.0
5	25.0	35.3	55.9	79.0	111.8
6	27.3	38.7	61.2	86.6	122.5
7	29.5	41.8	66.1	93.5	132.3
8	31.6	44.7	70.7	100.0	141.4
9	33.5	47.4	75.0	106.0	150.0
10	35.3	49.9	79.0	111.8	158.1

プリ解円弧精度と半径による飽和速度 [KPPS] の関係表 - 2 (制御周期 8 msec 時)

円弧精度	半 径				
	1000	2000	5000	10000	20000
10	35.3	49.9	79.0	111.8	158.1
20	49.7	70.5	111.7	158.0	223.6
30	60.8	86.3	136.7	193.5	273.8
40	70.0	99.5	157.8	223.4	316.1
50	78.1	111.1	176.3	249.7	353.3
60	85.3	121.6	193.1	273.5	387.0
70	91.9	131.1	208.4	295.3	418.0
80	98.0	140.0	222.7	315.6	446.8
90	103.7	148.3	236.1	334.7	473.8
100	109.0	156.1	248.7	352.7	499.4



## 6. 運転プログラム詳細

### 6-1 動作指令(命令)

#### 6-1-1 テクノコード指令一覧

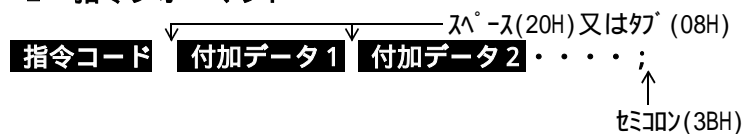
1.1.1.6

指令グループ	指令	名称	付加データ	
グループ 1	a	PTP	インクレ位置決め 軸指定(インクル量)、 OVR:オーバーライド	
		PTPA	論理座標系アブソ位置決め 軸指定(論理座標)、 OVR:オーバーライド	
		PTPB	機械座標系アブソ位置決め 軸指定(機械座標)、 OVR:オーバーライド	
		LIN	インクレ直線補間 軸指定(インクル量)、 F:送り速度	
		LINA	論理座標系アブソ直線補間 軸指定(論理座標)、 F:送り速度	
		LINB	機械座標系アブソ直線補間 軸指定(機械座標)、 F:送り速度	
		SLIN	センサーラッチ直線補間 軸指定(インクル量)、 F:送り速度 OP	
		PTMA	ポイント位置決め P:ポイント番号 OVR:オーバーライド	
		CIRR	インクレ円弧補間 (CW:右回り) 軸指定(インクル量):2軸~3軸指定します。 中心座標を指定した2軸で円弧補間 3軸目を指定すると円弧に同期して移動	
		CIRL	インクレ円弧補間 (CCW:左回り) 中心座標/半径指定:6-2-1を参照下さい。	
	CIRRA1	論理座標系アブソ円弧補間 (CW:右回り) 軸指定(アブソ量):2軸~3軸指定します。 中心座標を指定した2軸で円弧補間 3軸目を指定すると円弧に同期して移動		
	CIRLA1	論理座標系アブソ円弧補間 (CCW:左回り) 中心座標/半径指定:6-2-1を参照下さい。		
	REEL	巻線命令 巻線動作の専用命令。詳細は6-3-8を参照 OP 2.1		
	TURN	接線制御 詳細は機能説明を参照して下さい。 OP		
	SPIN	無限回転軸回転動作 軸指定(回転数:0.1rpm)		
	THSET	工具長補正有効 補正テーブル番号 OP 2		
	THOFF	工具長補正キャンセル OP 2		
	グループ 2	b	CALL	サブルーチンコール ラベル、L:回数
			JMP	ジャンプ ラベル
			PRG	グループの再起動 プログラム番号
IF			計算文	
ELSE			条件判断 -	
ENDIF		-		
-		マクロ命令 計算文		
c		STNE	接線制御有効 詳細は機能説明を参照して下さい。 OP	
		STND	接線制御無効 詳細は機能説明を参照して下さい。 OP	
		CSET	論理座標セットアップ 軸指定(論理座標)	
	END	プログラム終了 -		
	INPE	インプスチェック有効 -		
INPD	インプスチェック無効 -			
グループ 3	TIM	タイマー (0.1秒ステップ) 単位は秒(小数点以下1桁有効) TIM1:0.1秒, TIM1.0:1秒		
	TM	タイマー (RTC単位) 数値:RTC周期単位(2ms~8ms)		
	MOU T	Mコード出力 数値(0~255)		
グループ 4	ONR	汎用出力ON 数値(0~30):出力BIT指定		
	OF R	汎用出力OFF 数値(0~30):出力BIT指定		
	R	汎用入力制御 数値(0~34):入力BIT指定 E:EXIT (強制終了) J:スキップ (ステップ途中終了による飛ばし) W:ウェイト (待ち) S:ストップ (停止) N:負論理指定		
	なし/N E、J、W、S			
グループ 4	DC	径補正キャンセル -		
	DL	径補正方向指定(左側) 径補正テーブル番号(0~19) OP		
	DR	径補正方向指定(右側) 径補正テーブル番号(0~19) OP		
-	PNT	ポイント定義 軸指定(論理座標)		
	:	ラベル定義 最大13文字の文字列		
-	/*...*/	コメント "/" + 文字列 + "*/"		

- ・グループ1は1ステップ内での複数指定は出来ません。
- ・グループ2、3はグループ1と同一ステップでの指定が可能です。ただし、以下の制限があります。
  - ・汎用入出力制御、TIM/TMをグループ1cと同時に指定した場合、無視されます。
  - ・汎用入出力制御をグループ1bと同時に指定した場合、W/Sは無視されます。
- ・グループ3は同一ステップで最大4点までの指定が可能です。
  - 1 プリ解円弧方式では不可。
  - 2 座標の補正は、汎用入力にかかわらず、必ず実行されます。

汎用入力制御は摩耗補正時の軸移動にだけききます。

### 6 - 1 - 2 指令フォーマット



Gコードでは、区切りとしてスペース/タブが基本ですが、省略も可能です。  
(テクノコードでは、省略不可)

1.2

### 6 - 1 - 3. サンプル運転プログラム

運転プログラムのサンプルを別資料で紹介します。(テクノHP:会員コーナ)

「サンプル運転プログラム説明書」 TB04-0422

この資料は、テクノHPの会員コーナからダウンロードできます。

テクノコードプログラム作成例

```
PTPA X0 Y0 Z0 A0 TIM0.5 WR01 ONR02 ONR12 OFR14;  
CALL SUBPROC;  
TIM1.0;  
PTPA X100 Y100 Z100 A100 TIM0.5 JR15 WR00 ONR02 OFR10;  
CALL SUBPROC;  
TIM1.0;  
CIRR X400 Y0 Z100 I200 J0 F1000;  
END;  
:SUBPROC  
PTP X100 Y-100 Z100 A-100;  
PTP X-100 Y100 Z-100 A100 TIM0.5;  
LIN X1500 Y2000 Z2500 A3000 F1000;  
END;
```

6 - 1 - 4 Gコード 指令一覧

Gコード

Gコード	モーダル	対応する テコノド	動作
G91	モーダル	-	インクレ指定
G90		-	論理座標系アブソ指定
(G91)G00	モーダル	PTP	インクレ位置決め
(G90)G00		PTPA	論理座標系アブソ位置決め
G28	非モーダル	PTPB	機械座標系アブソ位置決め
(G91)G01	モーダル	LIN	インクレ直線補間
(G90)G01		LINA	論理座標系アブソ直線補間
		LINB	機械座標系アブソ直線補間
G31	非モーダル	SLIN	インクレセンサー補間 <OP>
(G91)G02	モーダル	CIRR	インクレ円弧補間CW 中心指定、半径指定(CR)
(G90)G02		CIRRA	論理座標系アブソ円弧補間CW
(G91)G03		CIRL	インクレ円弧補間CCW
(G90)G03		CIRLA	論理座標系アブソ円弧補間CCW
-		P**	**平面指定 円弧半径指定で必要
G17	モーダル	PXY	X Y平面指定
G18		PXZ	X Z平面指定
G19		PYZ	Y Z平面指定
G04	非モーダル	TIM	ドウェル
G40	モーダル	DC	径補正キャンセル <OP>
G41		DL	径補正方向指定(左側) <OP>
G42		DR	径補正方向指定(右側) <OP>
G43	モーダル	THSET	工具長補正開始 <OP>
G49		THOFF	工具長補正キャンセル
G61		INPE	インボズチェック有効
G64		INPD	インボズチェック無効
G92	非モーダル	CSET	論理座標設定
G100	モーダル	PTMA	ポイント位置決め <OP>
G110		STNE	接線制御有効 <OP>
G111		STND	接線制御キャンセル <OP>
G112	非モーダル	TURN	T U R N命令 <OP>
-		TLS	トルク制限モード開始 開発中 <OP>
-		TLC	トルク制限モード解除 "
G120	非モーダル	SPIN	無限回転軸回転動作
-		PRG	スレーブプログラム起動
-		JMP	ラベルジャンプ

補助コード

Gコード	テコノド	動作
M**	MOUT**	Mコード出力
M00	MOUT00	プログラムストップ
M01	MOUT01	オプションストップ
M03	-	主軸正転 <OP>
M04	-	主軸逆転
M05	-	主軸停止
M30	END	プログラム運転終了
M98	CALL	サブプログラム呼出
M99	END	サブプログラム終了
S	-	主軸回転速度設定 <OP>
PNT	PNT	ポイント位置決めテーブル <OP>
-	ONR	汎用出力制御
-	OFR	
-	ER/ENR	汎用入力制御
-	JR/JNR	
-	SR/SNR	
-	WR/WNR	
-	マク命令	四則演算、内部データ読出/書込
-	IF	条件判断
-	ELSE	
-	ENDIF	
N		シーケンス番号

## 6 - 1 - 5 Gコード プログラム例

```
N10 G90 G00 X0 Y0 Z0 A0 M11;
M98 P20 L2;
G04 P1.0;
G90 G00 X10.0 Y10.0 Z10.0 A10.0 M12;
M98 P20;
G04 P1.0;
G91 G02 X40.0 Y0 Z10.0 I20.0 J0 F100;
M30;

N20 G91 G00 X10.0 Y-10.0 Z10.0 A-10.0;
G00 X-10.0 Y10.0 Z-10.0 A10.0;
G04 P0.5
G01 X150.0 Y200.0 Z250.0 F200;
M99;
```

- 1 メインプログラムはサブプログラムで指令したモーダルGコードを引き継ぎません。
- 2 サブプログラムの先頭では、モーダルGコードは不定となります。  
従って、サブプログラムではモーダルGコードを設定し直して下さい。

### シーケンス番号指定 (N)

シーケンス番号は、サブプログラム呼び出し (M98) にて使用されるもので、サブプログラム先頭ステップの前に 'N' + 任意の番号 (最大4桁) で記述します。  
シーケンス番号の指定範囲は、0 ~ 9999 (10進数) です。  
シーケンス番号は任意の番号を設定可能ですが、同一番号を複数のステップに設定した場合、最初に見つかったステップを呼び出します。

### プログラム途中再開

Gコードプログラム実行中、実行したプログラムステップにシーケンス番号が設定されている場合、これを記憶します。  
プログラム実行がリセット操作やエラー発生によって中断された場合、次のプログラム起動時最後に記憶したシーケンス番号の位置よりプログラムを再開します。  
記憶したシーケンス番号は、次の操作や動作でクリアされます。

- ・リワインド操作を行った。
- ・プログラムを最後 ("M30") まで実行した。
- ・別プログラムを実行した。
- ・プログラムをダウンロードして上書きした。
- ・バックアップメモリ初期化操作を行った。

中断後のプログラム起動にて、プログラムの先頭から実行を行う場合はリワインド操作を行って下さい。

## 6 - 2 付加データ

付加データには、以下の3種類があります。

### 6 - 2 - 1 付加データアドレス付き数値指定

付加データアドレス付き数値指定は、1つの指令に対して、複数の付加データが必要な指令に対して使用されます。  
「OVR」「Pxy」以外は、テクノコード/Gコード共通です。

( )内のアドレスはGコードの表記です。

アドレス	内容	説明
X	第1軸移動量・移動位置指定	S L Mが制御する軸に対する、移動量・移動位置の指定です。
Y	第2軸移動量・移動位置指定	
Z	第3軸移動量・移動位置指定	設定範囲 - 999999999 ~ 999999999
A	第4軸移動量・移動位置指定	単位はパルスです。
I	第1軸円弧中心座標指定	円弧補間指令での中心位置指定です。
J	第2軸円弧中心座標指定	(現在位置からのインクレ量)
K	第3軸円弧中心座標指定	設定範囲 - 999999999 ~ 999999999
D	第4軸円弧中心座標指定	単位はパルスです。
OVR	オーバーライド指定	サーボパラメータ「PTP速度」に対するオーバーライドを指定します。(テクノコードのみ)
F	速度指定	補間移動時の速度を指定します。
P	ポイント番号	P T M Aで使用するポイントの番号を指定します。
L	回数指定	繰り返し回数を指定します。
R	円弧半径指定	円弧補間指令での円弧半径指定です。
Pxy	円弧補間平面指定	半径指定の円弧補間指令での円弧補間をおこなう軸の指定です。 x= 第1軸の軸名称 y= 第2軸の軸名称

注記 第1～第4軸の小数点以下の桁数を指定できます。(セッティングP C)

指 定	最小単位
1	1 = 1 P
1 0	0.1 = 1 P
1 0 0	0.01 = 1 P
1 0 0 0	0.001 = 1 P
1 0 0 0 0	0.0001 = 1 P

### 6 - 2 - 2 数値指定

数値指定は、T I M指令のように、指令と付加データが1 : 1であるような指令に対して使用されます。

### 6 - 2 - 3 ラベル指定

ラベル指定は、C A L L指令にて使用されるもので、プログラム中に設定されているラベル文字列を指定するものです。  
ラベルは、プログラムステップに対応するものとして、設定する事ができるもので、最大13文字の文字列です。

### 6 - 2 - 4 速度指定「F」の単位

セッティングP CにてFの単位を設定できます。

$F 1 = (\underbrace{1, 10, 100, 1000, 10000}_{\text{選択}}) P / (\underbrace{\text{秒} / \text{分}}_{\text{選択}})$

例 1 P = 1 μm のシステムでmm / s e c  
F 1 = 1 0 0 0 / 秒  
1 P = 0 . 1 μm のシステムでmm / m i n  
F 1 = 1 0 0 0 0 / 分

## 6 - 3 動作指令の詳細

【     】は、同じ意味のGコードです。

### 6 - 3 - 1 PTP : インクレPTP移動指令 【(G91)G00】

付加データとして各軸の移動量と、サーボパラメータのPTP速度に対するオーバーライドを指定します。  
(GコードではOVRはありません)

オーバーライド指定が無いときは100%が指定されたものとして、動作します。

【例】 PTP X100 Y-10 OVR150 ;

操作パネルやPCコマンドによるオーバーライド指定は、さらにこれに乗じます。

### 6 - 3 - 2 PTPA : 論理座標系アブソ位置決め 【(G90)G00】

付加データとして各軸の論理座標上の移動位置と、サーボパラメータのPTP速度に対するオーバーライドを指定します。

オーバーライド指定が無いときは100%が指定されたものとして、動作します。

【例】 PTPA X1000 Y-1000 OVR150 ;

### 6 - 3 - 3 PTPB : 機械座標系アブソ位置決め 【G28】

付加データとして各軸の機械座標上の移動位置と、サーボパラメータのPTP速度に対するオーバーライドを指定します。

オーバーライド指定が無いときは100%が指定されたものとして、動作します。

【例】 PTPB X100 Y-1000 OVR150 ;

### 6 - 3 - 4 LIN : 直線補間移動指令 LIN/LINA/LINB

LIN           インクレ直線補間 【(G91)G01】

LINA          論理座標系アブソ直線補間 【(G90)G01】

LINB          機械座標系アブソ直線補間

付加データとして各軸の移動量と補間移動速度を指定します。

補間移動速度(F)の設定範囲は1~100000000で、単位は「6-2-4 F速度単位」の通りです。

補間移動速度は移動指定軸すべての合成速度です。

【例】 LIN X100 Y-1000 F1500 ;

1.5

### 6 - 3 - 5 SLIN : センサラッチ直線補間 【G31】

仮の目標値に向かって直線補間動作して、センサラッチ信号SENCE\_IがONするとその位置を記憶して停止します。詳細は4-16-7を参照下さい。

【例】 SLIN X1000 Y2000 F1000 ;

### 6 - 3 - 6 PTMA : ポイント指定PTP位置決め移動 【G100】

指定した位置決めポイントへのPTP移動を行います。

付加データとして、位置決めポイント番号と、サーボパラメータのPTP速度に対するオーバーライドを指定します。(GコードではOVRはありません)

オーバーライド指定がない時は100%が指定されたものとして動作します。

アドレスPで指定する位置決めポイントは、運転プログラム中で設定します。(PNT命令)存在しない位置決めポイントを指令した場合はプログラムエラーとなります。

【例】 PTMA P1 OVR20 ;

## 6-3-7 CIRR / CIRL : 円弧補間(ヘリカル)移動指令 【G02 / G03】

CIRR / CIRL インクレ円弧補間(ヘリカル)(CW / CCW) [(G91)G02 / G03]

CIRRA / CIRLA 論理座標系アブソ円弧補間(ヘリカル)(CW / CCW) [(G90)G02 / G03]

2軸の終点(インクレ量又は論理系アブソ位置)を指定します。

3軸の指定は同期軸でヘリカル動作になります。半径指定と中心指定の2通りがあります。

プリ解方式とボード内部方式で動作が異なります。詳細は、「4-7 円弧補間」を参照下さい。

## 【半径指定】

```
CIRR PYA Y300 A400 R500 [L] F1000;
```

半径指定は円弧の半径を直接Rで指定し、以下のように設定できます。

R > 0のとき、円弧角180°以下の円弧

R < 0のとき、円弧角180°以上の円弧

また、半径指定を行う場合は、必ず平面を設定します。XY平面上で円弧補間を行う場合の指定方法は、PXYとします。平面指定はPとし、には軸名称が入ります。ただし、軸名称軸番号が先の方が先に来ます。

Gコードの場合は、G17、G18、G19で指定します。

第4軸を含む円弧はできません。(中心指定では全軸可)

[L]は省略可能な多回転回数の指定です。省略されているときはL1指定とします。

## 【中心指定】

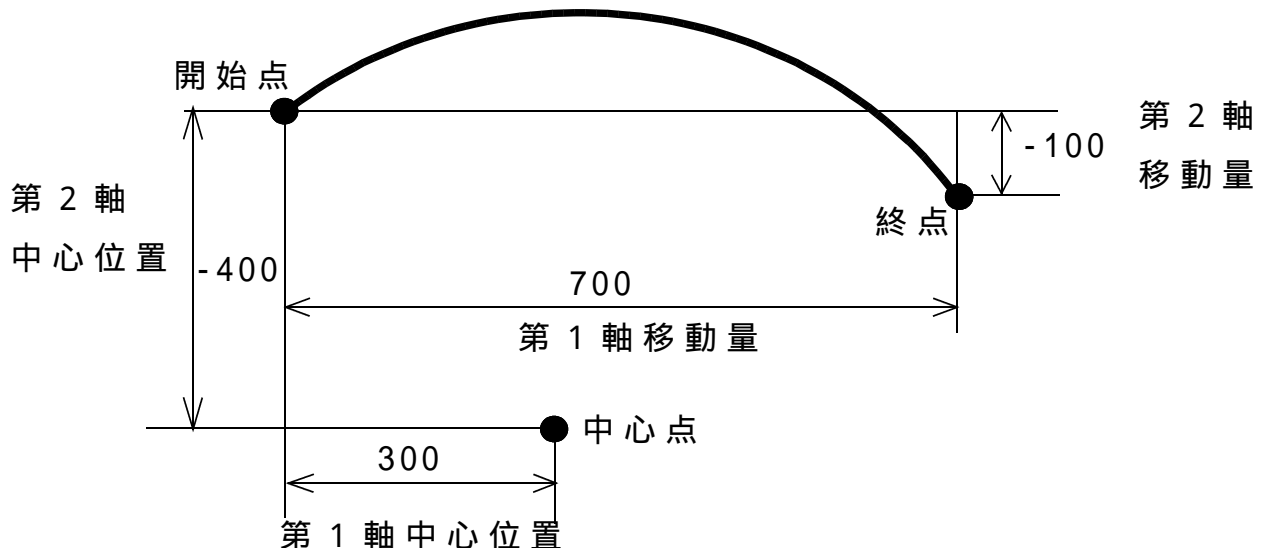
円弧中心位置(I、J、K、D)は、円弧補間移動開始位置からの距離にて指定を行います。

円弧中心位置指定の行われた軸が、円弧補間軸となります。従って、円弧中心位置指定は必ず2軸分行ってください。

円弧中心位置指定の行われた軸以外で、移動量が指定された軸は、円弧補間と同期させて、移動を行いません。(ヘリカル補間移動)

```
CIRR X700 Y-100 I300 J-400 [L] F1000 2.1
```

[L]は省略可能な多回転回数の指定です。省略されているときはL1指定とします。



## 【多回転機能】

多回転指令Lが指定されると(L指定回数 - 1)回の円の回転をした後、L回転目に指定された円弧移動します。例えばL3指定のときは、2回転した後一般の円弧移動になります。

- ・多回転L指令はマクロ変数指定も可能です。
- ・多回転Lの数値範囲は1 ~ 5000です(少数以下は無視)。それ以外はフォーマットエラーです。
- ・ヘリカル補間での第3軸の移動は、円弧の全移動量に同期します。
- ・多回転機能L指令は、プリ解析機能円弧には対応していません。

このとき、直径指定に設定している軸は移動量のみ直径指定として認識され、R値また中心位置の指定値は半径指定と見なされます。

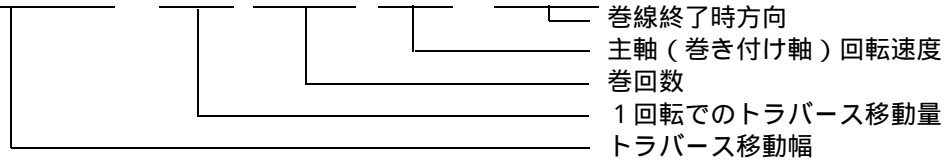
### 6 - 3 - 8 巻線指令 <オプション>

ポピンに巻線を行う専用の命令です。(詳細はTB00-0934を参照してください)

この命令に対応するGコードは有りません。(テクノコードのみです)

巻線指令

REEL { トラバース軸 $\pm$  } P $\pm$  RN $\pm$  S $\pm$  [ RE $\pm$  ] ;



{ トラバース軸 $\pm$  } は次の軸の中の何れかの1軸です X、Y、Z

[表記]  $\pm$  … 数値指定(即値/マクロ変数)を示します。

[ ] … 囲まれた項目が省略できることを示します。

(マクロ変数指定は対応している付加パラメータでのみ可能です。)

- ・ トラバース移動幅 : [ パルス単位、 -1,000,000,000 X/Y/Z/B/C/R/Q/T 1,000,000,000 ]  
A軸は主軸固定として使用するため、トラバース軸としては指定できません。  
トラバース幅が+の場合、最初に+方向に移動して、往復運動を行います。  
-の場合、最初に-方向に移動して、往復運動を行います。  
0の場合、トラバース軸は往復運動をせずに巻き取りのみ実行(主軸のみ回転)します。
- ・ P : ピッチ幅 [ パルス単位、 0.00 < P 100,000,00.00 ]  
主軸一回転当たりのトラバース軸の移動量を指定します。(パルス / REV)
- ・ RN : 総巻数 [ 主軸回転数単位、 -21,474,836.48 RN 21,474,836.47 ]  
総巻数(主軸回転数)を指定します。  
巻き数が+の場合、主軸は+方向に回転します。  
-の場合、主軸は-方向に回転します。  
0の場合、REEL命令は無効(NOP命令と同じ)になります。
- ・ S : 主軸速度 [ rpm単位、 0 < S 320,000 ]  
主軸の回転速度を指定します。  
主軸速度には、サーボパラメータの「PTP時定数」による加減速がかかります。
- ・ RE : 巻き終わり処理選択 [ 単位なし、 0/1/2 ] (省略時は0)  
巻き終わりの処理の方法を指定します。  
マクロ変数指定が可能です。

設定内容は以下の通りです。

- 0 : 自然終了(デフォルト)
- 1 : トラバース始端で終了
- 2 : トラバース終端で終了

#### 【動作】

運転プログラム内で巻線命令を実行する事により巻線動作を行います。

主軸が指定された速度で回転し、トラバース軸は主軸指令に同期して往復動作します。

尚、巻線動作にはオーバーライド指定が反映されません。

主軸速度には、サーボパラメータの「PTP時定数」による加減速がかかります。

トラバース移動は主軸指令に同期しますので主軸と同様の加減速がかかります。

トラバース軸独自の加減速はかかりません。



### 6 - 3 - 9 C S E T : 論理座標値設定

【 G 9 2 】

現在の論理座標系の位置を指定された値に変更（論理座標系のセットアップ）します。  
軸指定は任意の第 1 ~ 4 軸で、指定軸のみ座標変更されます。

【 例 】 C S E T X 0 Y 0 Z 1 0 0 ;

### 6 - 3 - 1 0 C A L L : サブルーチンコール

【 M 9 8 】

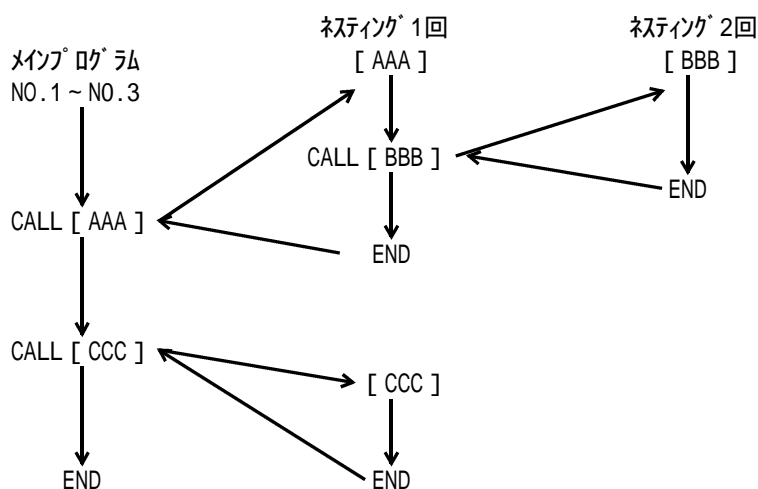
指定したラベルのサブルーチンを実行します。（Gコードでは、P でシーケンス番号を指定）  
付加データとして、ラベル（文字列）、呼び出し回数（L）を指定します。

呼び出し回数は、サブプログラム呼出の繰り返し回数です。（最大 5 0 0 0 0 回）

ここに設定されている回数分、サブプログラム呼出を行います。

【 例 】 C A L L L A B E L L 3 ; (テクノコード)  
M 9 8 P L ; (Gコード)

ネスティングは、最大 2 0 回可能です。



メインプログラムは必ずプログラムの先頭に記述して下さい。  
呼び出されるプログラムの先頭には必ずラベルが必要です。  
メインプログラム・各サブプログラムの最後には必ず“ E N D ” 指令を設定して下さい。

### 6 - 3 - 1 1 J M P : ジャンプ指定

指定したラベルへ実行を移します。

【 例 】 J M P L A B E L ;

### 6 - 3 - 1 2 P R G : スレーブプログラム起動 <オプション>

指定したプログラム番号のプログラムをスレーブプログラムとして実行します。

【 例 】 P R G 2 ;

### 6 - 3 - 1 3 E N D : プログラム終了

【 M 3 0 】 【 M 9 9 】

サブルーチンの終わり(M 9 9)、または運転プログラムの終了(M 3 0)を表します。

1.8

**6 - 3 - 14 接線制御 TURN/STNE/STND 【G112/G110/G111】**

接線制御用の命令です。

4 - 16 - 6 「接線制御機能」を参照下さい。

**6 - 3 - 15 工具長補正 THSET/THOFF 【G43/G49】**

4 - 18 - 2 「工具長補正機能及び摩耗補正機能」を参照下さい。

2.1

**6 - 3 - 16 工具径補正 DC/DL/DR 【G40/G41/G42】**

4 - 18 - 3 「工具径補正機能」を参照下さい。

2.1

**6 - 3 - 17 INPE : INPOS有効モード指定 【G61】**

INPEが指定されると、以後INPDが指令されるまでの各ステップは、インポジションチェック有効にて次のステップへ進みます。

但し、パラメータにてインポジションチェック無効としている場合は、無意コードとして扱います。詳細は、「4 - 9 インポジションチェック」を参照下さい。

**6 - 3 - 18 INPD : INPOS無効モード<デフォルト>指定 【G64】**

INPDが指令されると、以後INPEが指令されるまでの各ステップは、インポジションチェック無効(パスモード)にて次のステップへ進みます。

**6 - 3 - 19 TIM : ステップ実行時間指定 【G04】**  
**TM : ステップ実行時間指定 (RTC周期指定)**

各ステップの所用時間を指定します。

通常は、指定する必要がありません。この場合、移動距離÷送り速度でそのステップ時間となります。

$$\text{ステップ時間} = \text{軸移動時間}$$

時間を指定して、その設定値が軸移動時間を越えている場合、移動完了後にその差分の時間だけ、次のステップへの進行を停止します。

$$\text{ステップ時間} = \text{軸移動時間} + \text{差分の停止時間}$$

1.5

TIM指令の設定範囲は1.0～99999.9で、単位は1秒です。

【例】 TIM150 ;

G04は、他のGコードと同時に使えません。

【例】 G04 P1.0 ; (1秒待ち)

TM指令の設定範囲は1～99999で、単位はRTC周期です。

【例】 TM125 ; . . . RTC周期が8msの場合、この指令は1秒待ちに相当します。

TM指令は、プログラムアップロード時にTIM指令に変換されます。

その際、指定値は以下の式で変換され、端数は切り捨てられますので、御注意下さい。

$$\text{TIM指定値} = \text{TM指定値} \times \text{RTC周期} \div 100$$

RTC周期が8msの場合、

TM125 ;      TIM10 ;

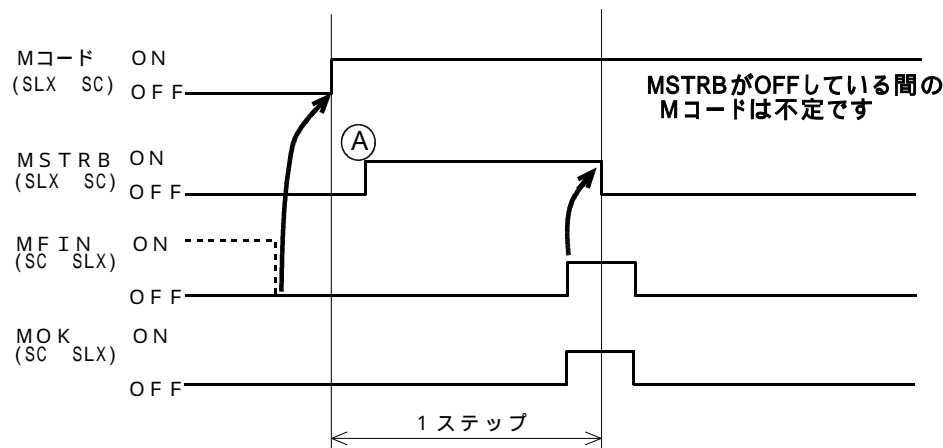
TM12 ;        TIM0 ;

指定したMコードを出力します。

【例】 MOUT 255 ;

シーケンサへステータスや他の情報を渡すために使用することも可能です。  
一般的にはデータ信号 (M0 ~ M7) とストロブ信号を出力します。

- **Mコード出力** M0 ~ M7の出力を利用して8ビットのバイナリデータ出力が行えます。  
(SLM SC) Mコード指令ステップ開始時に、ONとなります。
- **MSTRB出力** Mコード出力のストロブ信号です。  
(SLM SC) Mコード出力後に出力します。  
PLCでは、必ずMSTRBの立ち上がり(図のA)でMコードを読み込んでください 2.1
- **MFIN入力** 本信号の立ち上がりでMSTRB出力がOFFとなり1ステップが終了します。  
(SC SLM)
- **MOK入力** シーケンサ側でMコードの実行が正常におこなえた時、MOK = ONにします。  
(SC SLM) SLMでは、シーケンサ動作の正常 / 異常完了を判断できます。  
(MOKを使用するかどうかはROMスイッチで選択できます)



## 1.4,1.6

## ・ M00 / M01 指令

基本ROMSWにて、M00 / M01 停止機能を有効にすると、M00 / M01 指令時以下の動作を行います。

M00 指令時・・・プログラム実行を停止します。

M01 指令時・・・オプションストップ入力("MOPT")ONの時プログラム実行停止

M00 / M01 停止機能有効の場合、M00 / M01 指令時にMコード出力は行いません。  
また、入力信号Ri15をオプションストップ入力"MOPT"として使用します。

基本ROMSWにて、M00 / M01 停止中出力を有効にすると、M00 又はM01 停止中に、M00 停止中出力("MZSTP")をONにします。

M00 / M01 停止中出力を有効の場合、出力信号"Ro09"は出力信号"MZSTP"として使用します。

M00 / M01 停止機能が無効の場合は、通常のMコードと同様にMコードを出力します。

移動指令と同一行でのMOUT 命令

移動を完了した後にMOUT 出力します。

例 : LIN X\_\_ Y\_\_ M ; 直線補間が完了した後でM 出力

・ Gコードプログラムの予約Mコード

Gコードプログラムでは、以下のMコードの機能が予約されています。  
( 6 - 1 - 4 Gコード指令一覧 補助コード を参照して下さい。 )

**M03 / M04 / M05**

主軸の正転 ( M03 )、主軸の逆転 ( M04 )、主軸の停止 ( M05 ) を行います。  
ロムスイッチの " 主軸Mコード出力 " を有効にした場合、同時にMコード出力を行います。  
ロムスイッチの " 主軸Mコード出力 " が無効の場合は、Mコード出力を行いません。

**M30**

プログラムを終了します。  
本指令ではMコード出力を行いません。

**M98 / M99**

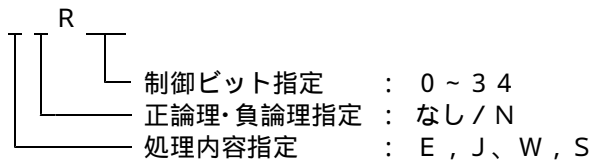
サブプログラムの呼び出し ( M98 )、サブプログラムからの戻り ( M99 ) を行います。  
本指令ではMコード出力を行いません。

1.2,1.6

6 - 3 - 21 汎用入出力処理指定

汎用入力処理と汎用出力処理は、1ステップ内で同時4点までの指定が可能です。

汎用入力処理



2.1

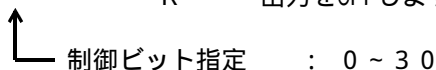
処理名称	機能・内容	表記事例	優先順位
<b>E x i t</b> プログラム強制終了	この信号がアクティブになると強制的に運転プログラムを完了して、停止します。	E R 0 0	第 1
<b>J u m p</b> スキップ	この信号がアクティブになるとそのステップの動作を中断し、次のステップへ進行します。スキップ時の動作は、スキップしたステップの次ステップの指令形式によって変わります。	J N R 0 1	第 2
<b>W a i t</b> ステップ実行開始待ち	各動作ステップの先頭でこの入力をチェックします。アクティブであれば停止します。そのステップが一端開始されるとその後はWAITのチェックはしません。	W R 0 2	第 3
<b>S t o p</b> 一時停止	この信号が アクティブな間、動作を停止します。ステップの実行中いつもチェックします。したがって、STOP条件のON/OFFでステップ動作が停止/再開します。	S N R 0 3	第 3

同一ステップでW・S・J・Eが同時に指定され、各々の入力条件が成立した時の優先順位は、以下の通りです。

- 第1優先    E X I T    E指定の入力が1つでも成立すれば、他の条件にかかわらず強制終了します。
- 第2優先    J u m p    第1優先が成立していない状態では、J指定が1つでも成立すれば、スキップします。
- 第3優先    W a i t  
S t o p    第1・第2優先が成立していない状態では、W又はSの1つでも成立すれば、それに従います。

汎用出力処理

ONR                    R    出力をONします  
OFR                    R    出力をOFFします



汎用出力は、そのステップのはじめに出力されます。

【 例 】 P T P X 1 0 0 0 E R 0 0 J N R 1 0 O N R 0 5 O F R 0 1 ;

### 6 - 3 - 2 2 PNT : 位置決めポイント設定

【PNT】

P T M A 命令で使用する位置決めポイントの座標を設定します。  
付加データとして、ポイント番号と各軸の座標を指定します。  
指定しなかった軸は移動の際に無効軸となります。

1ポイント設定するする毎に1/4ステップ分のプログラム容量を使用します。  
このため、実行可能なプログラムステップ数は以下の通りになります。

$$\text{実行可能ステップ数} = \text{最大ステップ数 (ROMSWで設定)} - \left( \text{ポイント数} \times \frac{16}{64} \right)$$

(端数は切り上げ)

アドレスPで設定できるポイント番号は1からの連番です。(1 2 3 ~)  
それ以外の時はプログラムエラーになります。

[例] PNT P1 X10000 Y20000 A0 ;  
PNT P2 X0 Y0 Z0 A0 ;

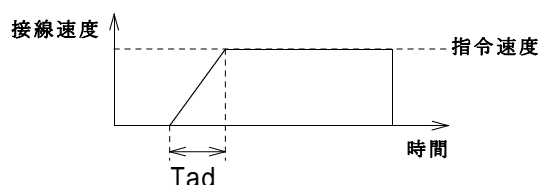
### 6 - 3 - 2 3 プリ解加減速(補間前加減速) <オプション>

補間移動指令 (LIN<sub>x</sub>, CIR<sub>xx</sub> / G01, G02, G03) に加減速指定 (P1 / P2 / P3) を付加する事により、プリ解析処理において、加 / 減速区間微小ブロックの発生が可能です。  
加減速時間(時定数) T a d は、I N I ファイル及び、プログラム変換ライブラリ関数で設定します。

- P 1 . . . 加速区間のステップを生成します。
- P 2 . . . 減速区間のステップを生成します。
- P 3 . . . 加速区間 / 減速区間のステップを生成します。

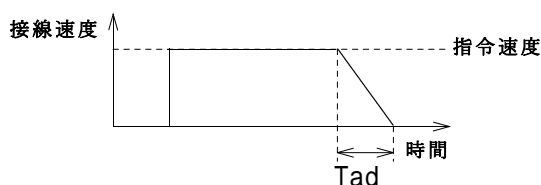
#### 加速(P1)指定

G 0 1 X\_\_ Y\_\_ F\_\_ P 1 ;  
G 0 2 X\_\_ Y\_\_ I\_\_ J\_\_ F\_\_ P 1 ;



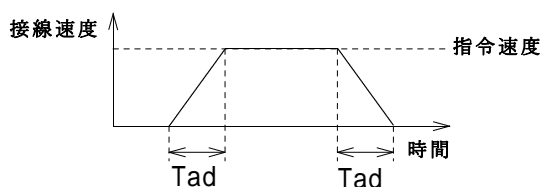
#### 減速(P2)指定

G 0 1 X\_\_ Y\_\_ F\_\_ P 2 ;  
G 0 2 X\_\_ Y\_\_ I\_\_ J\_\_ F\_\_ P 2 ;



#### 加速減速(P3)指定

G 0 1 X\_\_ Y\_\_ F\_\_ P 3 ;  
G 0 2 X\_\_ Y\_\_ I\_\_ J\_\_ F\_\_ P 3 ;



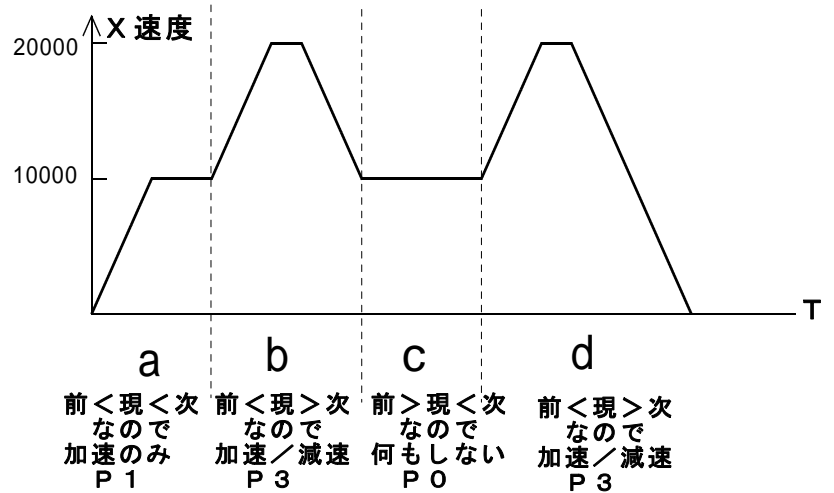
- 注記 1 加速 / 減速区間は、微小直線補間の連続に展開します。  
注記 2 Gコードの場合、加速は速度ゼロから、減速は速度ゼロへ移行します。従って連続補間でのなめらかな速度変更は、従来通りの各軸加減速(指数 / 直線)を用いてください。

## テクノコードでの速減目標速度(F L\_)指定

- 注記3 テクノコードの場合、減速時の目標速度をFLコードにて指定できます。
- 注記4 プリ解加減速は接線速度に対して行います。従って円弧補間の場合、開始角度、終了角度によって、各軸の速度変化が大きくなる場合があります。  
これを滑らかにする場合は、各軸加減速(指数/直線)が利用可能です。
- 注記5 テクノコードの場合、前のステップ・現ステップ・次ステップの指令速度の変化を滑らかにすることができます。  
この時の加減速指定は、前のステップ・現ステップ・次ステップの速度を比べて、一番速い速度のステップで行って下さい。

### プログラム

```
a.LIN X10000 F10000 P1 ;
b.LIN X20000 F20000 P3 ;
c.LIN X30000 F10000 P0 ;
d.LIN X40000 F20000 P3 ;
```



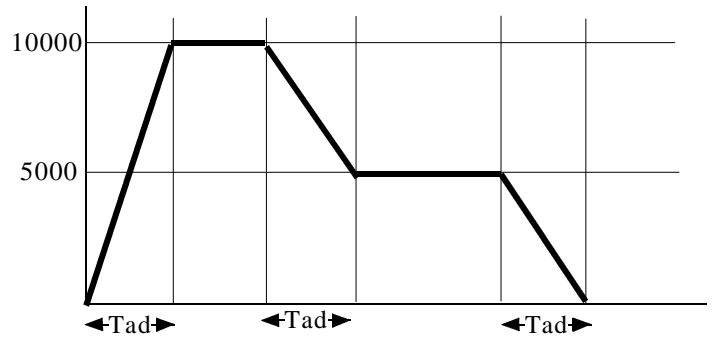
## 動きの例

```

LIN X10000 F10000 P3;
LIN X5000 F5000 P2;

LIN X10000 F10000 FL5000 P3;
LIN X5000 F5000 P2;

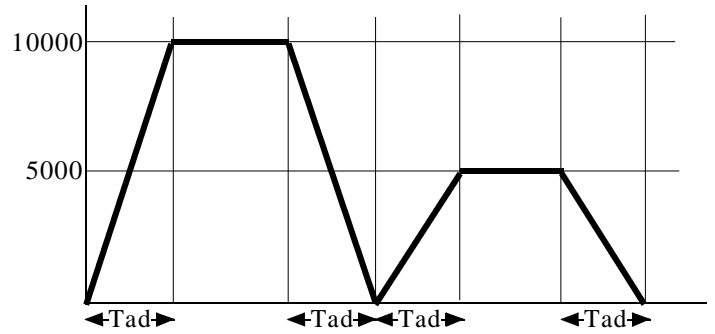
```



```

LIN X10000 F10000 P3;
LIN X5000 F5000 P3;

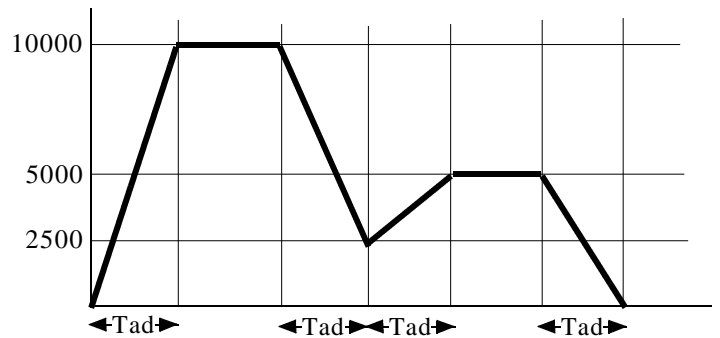
```



```

LIN X10000 F10000 FL2500 P3;
LIN X5000 F5000 P2;

```



## 【注】

FL0はフォーマットエラーです。

P1/P3 指定時、前ステップの速度が該当ステップの速度以上だとフォーマットエラーになります。

P2/P3 指定時、次ステップの速度が該当ステップの速度以上だとフォーマットエラーになります。

P2/P3 指定時、次ステップが P1/P3 指定だと速度 0（または F L 指定速度）まで減速します。

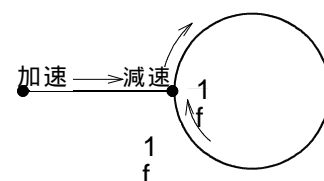
P1/P3 は必ず増速時、 P2/P3 は必ず減速時のステップで指定してください。

## 円弧の半径縮みを最小にする方法。

以下の指令方法で形状補正を併用すると半径縮みは最小になります。

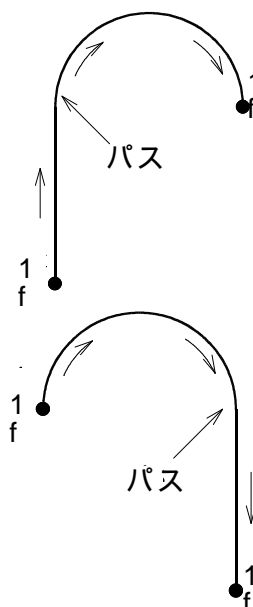
a .

```
G01 X10000 F1000 P3;  
G02 X0 Y0 I10000 J0 P3;
```



b .

```
G01 Y10000 F1000 P1;  
G02 X20000 Y0 I10000 J0 P2;
```



c .

```
G02 X20000 Y0 I10000 J0 F1000 P1;  
G01 Y-10000 P2;
```

1.2

### 6 - 3 - 24 SPIN 回転速度指令 【G120】

4 - 16 - 2 「回転速度指令」を参照下さい。

1.2

### 6 - 3 - 25 主軸指令 【S】【M03/04/05】

4 - 17 「主軸機能」を参照下さい。

### 6 - 3 - 26 ':' : ラベル定義

':' + 文字列でラベルを定義します。

文字列は最大13文字まで指定可能です。

ラベル定義の最大数は、ROMSWのプログラム分割数によってかわります。

### 6 - 3 - 27 '/\*' ... '\*/' : コメント

任意の文字列を"/\*"と"\*/"で囲む事により、運転プログラムテキストにコメントを記述することが可能です。

DNC運転プログラムでは、複数行にまたがるコメントはできません。



## 6 - 4 マクロ機能 <オプション>

運転プログラム中での演算、及び条件判断を行う機能です。  
NCの各種状態や外部からの入力値により、移動位置の変更を行ったり、  
サーボパラメータの変更等を行うことができます。

1.1,1.4

### 6 - 4 - 1 書式

- ・ 変数 = {変数 | 即値} [ 演算子 {変数 | 即値} ] ;
- ・ IF {変数 | 即値} [ 演算子 {変数 | 即値} ] ;  
(1)  
ELSE ;  
(2)  
ENDIF ;  
式の値が真の時(1)の句を実行、偽の時(2)の句を実行。  
ELSE句は省略可能

{ | } ... ' | ' で区切られた内の、どれか1つを選択  
[ ] ... 省略可能  
... 1つ以上の空白、又はタブ(必須です)  
変数 ... レジスタ指定  
即値 ... 500、-1などの数値  
演算子 ... + (加算)  
- (減算)  
\* (乗算)  
/ (除算)  
% (剰余)  
& (AND)  
| (OR)  
^ (XOR)  
<< (左シフト)  
>> (右シフト)  
<  
<=  
>  
>=  
==  
!= } 条件が真ならば1  
偽ならば0

- ・ PTP、PTPA、PTPB、LIN、LINA、LINBの移動量(位置)の指定、  
PTMAのポイント番号、SPINの回転速度指定、TIMに変数を使用することが出来ます。

変数、即値には符号(+、-)とビット反転(~)を付加することができます。

2.1

例)

```
IF #1000 < #1001;          IF #1000 < #1001;
PTP X1000;                 G91 G00 X1000;
TIM1.0;                    G04 P1.0;
PTP X100;                   G00 X100;
ENDIF;                      ENDIF;
PTPA X1000;                 G90 G00 X1000;
END;                         M30;
```

- ・ #1000が0、#1001が1の時の実行順序
- ・ #1000が1、#1001が0の時の実行順序

## 6 - 4 - 2 レジスタ一覧

#1000 ~ 一般レジスタ

#1500 ~ 各種情報

#1500 ~ 全体情報

#1520 ~ 第 1 ~ 5 軸ステータス情報

#1540 ~ 第 1 ~ 5 軸アラーム情報

#1600 ~ 入出力情報

#2000 ~ サーボパラメータ

#2000 ~ 第 1 軸サーボパラメータ

#2100 ~ 第 2 軸サーボパラメータ

#2200 ~ 第 3 軸サーボパラメータ

#2300 ~ 第 4 軸サーボパラメータ

#2400 ~ 第 5 軸サーボパラメータ

#3000 ~ 工具長補正用データ

#4000 ~ ポジションデータ

#4000 ~ 第 1 軸ポジションデータ

#4100 ~ 第 2 軸ポジションデータ

#4200 ~ 第 3 軸ポジションデータ

#4300 ~ 第 4 軸ポジションデータ

#4400 ~ 第 5 軸ポジションデータ

#5000 ~ センサーラッチポジションデータ

#5000 ~ 第 1 ~ 5 軸センサーラッチ論理座標

#5100 ~ 第 1 ~ 5 軸センサーラッチアブソ座標

#6000 ~ H E X 入力

## 6 - 4 - 3 一般レジスタ

# 1000 ~ # 1099

運転プログラム内で自由に使用できる領域です。

本レジスタの内容はバッテリーバックアップされていますので、電源断時でも保持されています。

番号	データ名称	属性	設定値
#1000	一般レジスタ	R/W	-2147483648 ~ 2147483647
#1001	一般レジスタ	R/W	-2147483648 ~ 2147483647
#1002	一般レジスタ	R/W	-2147483648 ~ 2147483647
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
#1095	一般レジスタ	R/W	-2147483648 ~ 2147483647
#1096	一般レジスタ	R/W	-2147483648 ~ 2147483647
#1097	一般レジスタ	R/W	-2147483648 ~ 2147483647
#1098	一般レジスタ	R/W	-2147483648 ~ 2147483647
#1099	一般レジスタ	R/W	-2147483648 ~ 2147483647

表記

番号 … データ指定番号

データ名称 … データ名

属性 … R - 書込可、W - 読込可

設定値 … データ範囲

6 - 4 - 4 各種情報

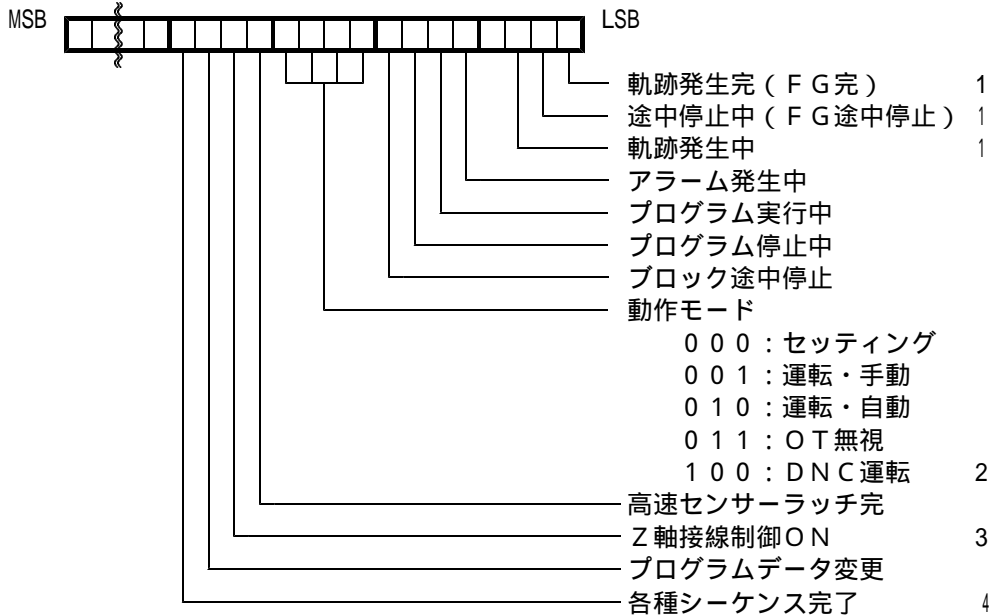
# 1500 ~

番号	データ名称	属性	設定値
#1500	各種状態情報	R	
#1501	アラーム情報	R	
#1502	送りオーバーライト設定	R/W	
#1503	選択・実行プログラム番号	R/W	
#1504	待機・実行ブロック番号	R	

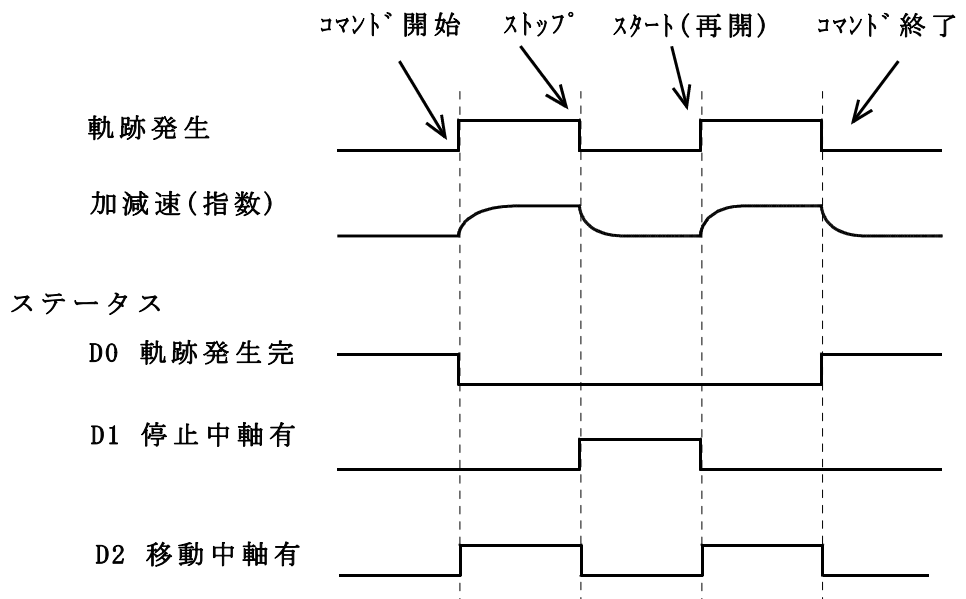
1.5

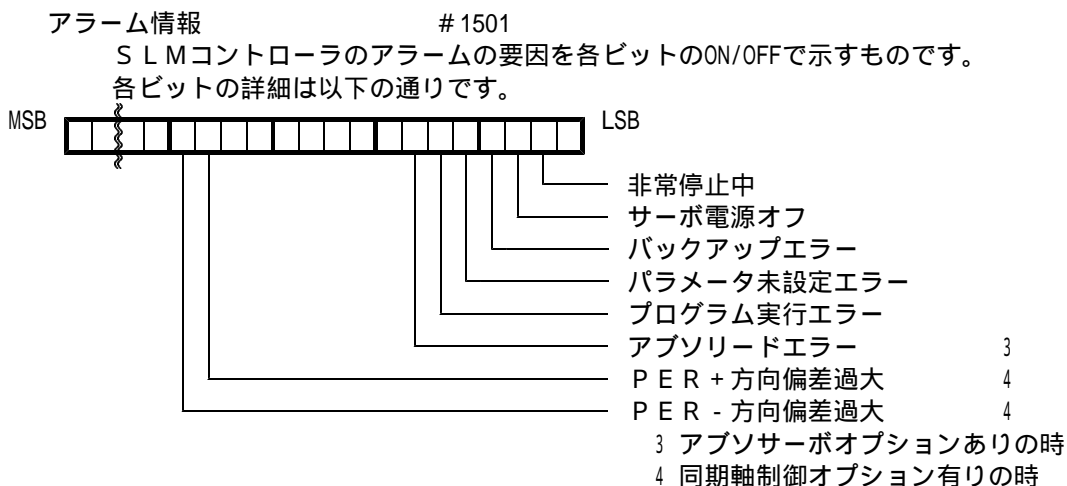
各種状態情報 # 1500

S L Mコントローラの現在の状態を各ビットのON/OFFで示すものです。  
各ビットの詳細は以下の通りです。



- 1 F G ステータスタイミングチャート
- 2 D N C オプション有りの時
- 3 Z 軸接線制御オプション有りの時
- 4 以下の条件が成立しているときに ON します ( A N D 条件 ) 。
  - ・ F G 完
  - ・ M コード出力シーケンス未実行
  - ・ 原点復帰未実行
  - ・ ホーム位置決め未実行





送りオーバーライド設定 # 1502  
現在の送りオーバーライド設定の設定値です。

選択・実行プログラム番号 # 1503  
ステータスがプログラム実行中でない場合 選択プログラム番号  
ステータスがプログラム実行中の場合 実行プログラム番号

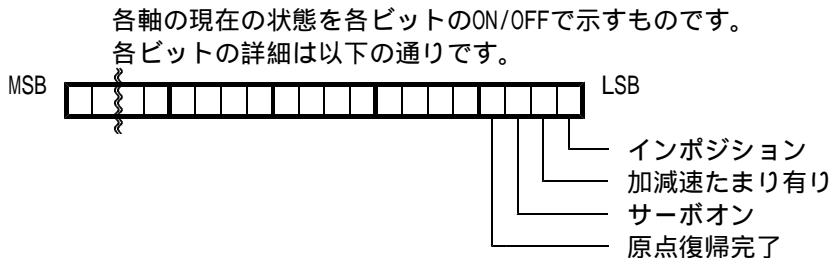
待機・実行ブロック番号 # 1504  
ブロック実行中でない場合 待機ブロック番号  
ブロック実行中の場合 実行ブロック番号

1.6  
注記

# 15 は、実際の動作から変数に値が反映されるまでに数 R T C（制御周期）かかる場合があります。  
正しい値を得るためには、これらのマクロ変数の読み出し前にドウェルを 0.1 秒程度入れる必要があります。

6 - 4 - 5 第 1 軸～第 5 軸ステータス情報 # 1520～# 1524

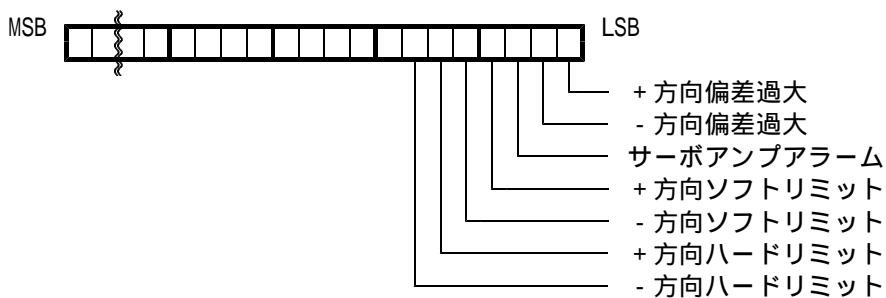
番号	データ名称	属性	設定値
#1520	第 1 軸ステータス情報	R	
#1521	第 2 軸ステータス情報	R	
#1522	第 3 軸ステータス情報	R	
#1523	第 4 軸ステータス情報	R	
#1524	第 5 軸ステータス情報	R	



6 - 4 - 6 第1軸～第5軸アラーム情報 #1540～#1544

番号	データ名称	属性	設定値
#1540	第1軸アラーム情報	R	
#1541	第2軸アラーム情報	R	
#1542	第3軸アラーム情報	R	
#1543	第4軸アラーム情報	R	
#1544	第5軸アラーム情報	R	

各軸のアラームの要因を各ビットのON/OFFで示すものです。各ビットの詳細は以下の通りです。

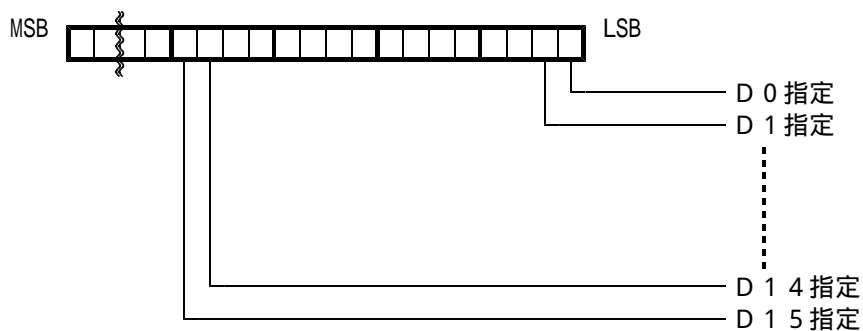


1.1, 1.5

6 - 4 - 7 入/出力パターン #1600～#1615

番号	データ名称	属性	設定値
#1600	入力1パターン (I#0000)	R	
#1601	入力2パターン (I#0001)	R	
#1602	入力3パターン (I#0002)	R	
#1610	出力1パターン (O#0000)	R/W	
#1611	出力2パターン (O#0001)	R/W	

各入出力ポートのデータを16Bitバイナリ値として読み込みます。各ビットの詳細は以下の通りです。



1.2  
6 - 4 - 8 サーボパラメタ #2000 ~

番号	データ名称	属性	設定値
#2000	第1軸 D/A出力ゲイン	R/W	0.004 ~ 256.0
#2001	第1軸 INPOS量	R/W	0 ~ 10000[Pulse]
#2002	第1軸 ER上限値	R/W	100 ~ 10000[Pulse]
#2003	第1軸 ERサチレーション	R/W	100 ~ 10000[Pulse]
#2004	第1軸 PTP時定数	R/W	20 ~ 3000[msec]
#2005	第1軸 PTP速度	R/W	250 ~ 4000000[Pulse/Sec]
#2006	第1軸 JOG送り速度	R/W	250 ~ 4000000[Pulse/Sec]
#2007	第1軸 補間時定数	R/W	-1000 ~ 1000[msec]
#2008	第1軸 +側ソフトリミット	R/W	1 ~ 100000000[Pulse]
#2009	第1軸 -側ソフトリミット	R/W	1 ~ 100000000[Pulse]
#2010	第1軸 形状補正係数	R/W	0 ~ 2000
#2011	第1軸 S字加減速	R/W	0 ~ 100[msec]
#2100	第2軸 D/A出力ゲイン	R/W	0.004 ~ 256.0
#2101	第2軸 INPOS量	R/W	0 ~ 10000[Pulse]
#2102	第2軸 ER上限値	R/W	100 ~ 10000[Pulse]
#2103	第2軸 ERサチレーション	R/W	100 ~ 10000[Pulse]
#2104	第2軸 PTP時定数	R/W	20 ~ 3000[msec]
#2105	第2軸 PTP速度	R/W	250 ~ 4000000[Pulse/Sec]
#2106	第2軸 JOG送り速度	R/W	250 ~ 4000000[Pulse/Sec]
#2107	第2軸 補間時定数	R/W	-1000 ~ 1000[msec]
#2108	第2軸 +側ソフトリミット	R/W	1 ~ 100000000[Pulse]
#2109	第2軸 -側ソフトリミット	R/W	1 ~ 100000000[Pulse]
#2110	第2軸 形状補正係数	R/W	0 ~ 2000
#2111	第2軸 S字加減速	R/W	0 ~ 100[msec]
#2200	第3軸 D/A出力ゲイン	R/W	0.004 ~ 256.0
#2201	第3軸 INPOS量	R/W	0 ~ 10000[Pulse]
#2202	第3軸 ER上限値	R/W	100 ~ 10000[Pulse]
#2203	第3軸 ERサチレーション	R/W	100 ~ 10000[Pulse]
#2204	第3軸 PTP時定数	R/W	20 ~ 3000[msec]
#2205	第3軸 PTP速度	R/W	250 ~ 4000000[Pulse/Sec]
#2206	第3軸 JOG送り速度	R/W	250 ~ 4000000[Pulse/Sec]
#2207	第3軸 補間時定数	R/W	-1000 ~ 1000[msec]
#2208	第3軸 +側ソフトリミット	R/W	1 ~ 100000000[Pulse]
#2209	第3軸 -側ソフトリミット	R/W	1 ~ 100000000[Pulse]
#2210	第3軸 形状補正係数	R/W	0 ~ 2000
#2211	第3軸 S字加減速	R/W	0 ~ 100[msec]
#2300	第4軸 D/A出力ゲイン	R/W	0.004 ~ 256.0
#2301	第4軸 INPOS量	R/W	0 ~ 10000[Pulse]
#2302	第4軸 ER上限値	R/W	100 ~ 10000[Pulse]
#2303	第4軸 ERサチレーション	R/W	100 ~ 10000[Pulse]
#2304	第4軸 PTP時定数	R/W	20 ~ 3000[msec]
#2305	第4軸 PTP速度	R/W	250 ~ 4000000[Pulse/Sec]
#2306	第4軸 JOG送り速度	R/W	250 ~ 4000000[Pulse/Sec]
#2307	第4軸 補間時定数	R/W	-1000 ~ 1000[msec]
#2308	第4軸 +側ソフトリミット	R/W	1 ~ 100000000[Pulse]
#2309	第4軸 -側ソフトリミット	R/W	1 ~ 100000000[Pulse]
#2310	第4軸 形状補正係数	R/W	0 ~ 2000
#2311	第4軸 S字加減速	R/W	0 ~ 100[msec]
#2400	第5軸 D/A出力ゲイン	R/W	0.004 ~ 256.0
#2401	第5軸 INPOS量	R/W	0 ~ 10000[Pulse]
#2402	第5軸 ER上限値	R/W	100 ~ 10000[Pulse]
#2403	第5軸 ERサチレーション	R/W	100 ~ 10000[Pulse]
#2404	第5軸 PTP時定数	R/W	20 ~ 3000[msec]
#2405	第5軸 PTP速度	R/W	250 ~ 4000000[Pulse/Sec]
#2406	第5軸 JOG送り速度	R/W	250 ~ 4000000[Pulse/Sec]
#2407	第5軸 補間時定数	R/W	-1000 ~ 1000[msec]
#2408	第5軸 +側ソフトリミット	R/W	1 ~ 100000000[Pulse]
#2409	第5軸 -側ソフトリミット	R/W	1 ~ 100000000[Pulse]
#2410	第5軸 形状補正係数	R/W	0 ~ 2000
#2411	第5軸 S字加減速	R/W	0 ~ 100[msec]

2.1

## D/A出力ゲイン (現在対応していません)

ER (偏差パルス) に本設定値を乗算した値が、D/Aへの出力値となります。  
サーボのKpと比例関係にありますので、本設定値を変更することによりサーボのKpをソフト的に調整することができます。  
固定小数点型2バイトデータです。

(上位バイト: 整数部 / 下位バイト: 小数部)

【設定例】	指定値	設定値 (hex)
	1.00	0100h
	0.50	0080h
	2.70	02B3h

## INPOS量

インポジションの範囲を、パルス単位の絶対値量で指定します。

## ER上限値 (その軸のFBを有効としたときのみ使います)

偏差の絶対値の上限値を指定します。

ERが本設定値を越えた場合、偏差過大アラームとなります。

## ERサチレーション (現在対応していません)

D/A出力に用いる、偏差量(絶対値)のクランプ値を指定します。

ERが本設定値を越えてかつ、ER上限まで達していない場合、D/A出力基数(通常ER)に本設定値を使用します。

## PTP時定数

PTP移動時、及びJOG移動時の直線加減速の時定数を指定します。  
移動速度100Kppsに達するまでの時間で設定します。

## PTP速度

PTP移動時の速度を指定します。

## JOG速度

JOG移動時の速度を指定します。

## 補間時定数

補間移動時の指数型加減速の時定数を指定します。  
直線加減速オプション有りの時、マイナスの値を入れることで、直線加減速時定数を指定します。

## ソフトリミット

ソフトウェアストロークリミットを、各方向(+/-)別に指定します。  
ポジション値が本設定値を越えた場合、ソフトリミットアラームとなります。

## 形状補正係数

形状補正機能<オプション>のゲインを設定します。  
設定値を小さくするほど、形状補正機能のゲインが大きくなります。  
0を設定すると形状補正機能を無効にします。

これらのパラメータは現在使用しておりません。将来用です。

1.4, 1.5

注1 本マクロ変数でパラメータを読み出した場合、常にバックアップされたパラメータが読み出されます。

注2 本マクロ変数に書き込みを行った場合、書き込んだ値は直ちにSLM内部に反映されます。

注3 異常な値を書き込んだ場合は、軸の制御が正常に行えなくなる可能性がありますので十分に注意して下さい。

注4 補間時定数は、加減速たまり無しの時のみ変更可能です。

#2001 = 100;	}	ここでパラメータ変更
·		この間に変更されたパラメータで動作
·		
#2001 = #2001;		バックアップされたパラメータが読み出されるためパラメータ値が元に戻る。

## 6 - 4 - 9 工具長補正データ # 3000 ~

番号	データ名称	属性	設定値
#3000	工具長補正用データ 1	R/W	-1000000000 ~ 1000000000[Pulse]
#3001	工具長補正用データ 2	R/W	-1000000000 ~ 1000000000[Pulse]
#3002	工具長補正用データ 3	R/W	-1000000000 ~ 1000000000[Pulse]
#3003	工具長補正用データ 4	R/W	-1000000000 ~ 1000000000[Pulse]
#3004	工具長補正用データ 5	R/W	-1000000000 ~ 1000000000[Pulse]
#3005	工具長補正用データ 6	R/W	-1000000000 ~ 1000000000[Pulse]
#3006	工具長補正用データ 7	R/W	-1000000000 ~ 1000000000[Pulse]
#3007	工具長補正用データ 8	R/W	-1000000000 ~ 1000000000[Pulse]
#3008	工具長補正用データ 9	R/W	-1000000000 ~ 1000000000[Pulse]
#3009	工具長補正用データ 10	R/W	-1000000000 ~ 1000000000[Pulse]
#3010	工具長補正用データ 11	R/W	-1000000000 ~ 1000000000[Pulse]
#3011	工具長補正用データ 12	R/W	-1000000000 ~ 1000000000[Pulse]
#3012	工具長補正用データ 23	R/W	-1000000000 ~ 1000000000[Pulse]
#3013	工具長補正用データ 14	R/W	-1000000000 ~ 1000000000[Pulse]
#3014	工具長補正用データ 15	R/W	-1000000000 ~ 1000000000[Pulse]
#3015	工具長補正用データ 16	R/W	-1000000000 ~ 1000000000[Pulse]
#3016	工具長補正用データ 17	R/W	-1000000000 ~ 1000000000[Pulse]
#3017	工具長補正用データ 18	R/W	-1000000000 ~ 1000000000[Pulse]
#3018	工具長補正用データ 19	R/W	-1000000000 ~ 1000000000[Pulse]
#3019	工具長補正用データ 20	R/W	-1000000000 ~ 1000000000[Pulse]
#3100	第 1 軸工具長補正データ	R/W	-1000000000 ~ 1000000000[Pulse]
#3101	第 2 軸工具長補正データ	R/W	-1000000000 ~ 1000000000[Pulse]
#3102	第 3 軸工具長補正データ	R/W	-1000000000 ~ 1000000000[Pulse]
#3103	第 4 軸工具長補正データ	R/W	-1000000000 ~ 1000000000[Pulse]
#3104	第 5 軸工具長補正データ	R/W	-1000000000 ~ 1000000000[Pulse]

工具長補正機能<オプション>用のデータです。



6 - 4 - 1 0 ポジションデータ # 4000 ~

番号#	データ名称	属性	設定値
#4000	第1軸指令位置	R	-1000000000 ~ 1000000000[Pulse]
#4001	第1軸機械位置	R	-1000000000 ~ 1000000000[Pulse]
#4002	第1軸偏差量	R	-1000000000 ~ 1000000000[Pulse]
#4003	第1軸最新ステップ払出量	R	-1000000000 ~ 1000000000[Pulse]
#4004	第1軸絶対位置	R	-1000000000 ~ 1000000000[Pulse]
#4100	第2軸指令位置	R	-1000000000 ~ 1000000000[Pulse]
#4101	第2軸機械位置	R	-1000000000 ~ 1000000000[Pulse]
#4102	第2軸偏差量	R	-1000000000 ~ 1000000000[Pulse]
#4103	第2軸最新ステップ払出量	R	-1000000000 ~ 1000000000[Pulse]
#4104	第2軸絶対位置	R	-1000000000 ~ 1000000000[Pulse]
#4200	第3軸指令位置	R	-1000000000 ~ 1000000000[Pulse]
#4201	第3軸機械位置	R	-1000000000 ~ 1000000000[Pulse]
#4202	第3軸偏差量	R	-1000000000 ~ 1000000000[Pulse]
#4203	第3軸最新ステップ払出量	R	-1000000000 ~ 1000000000[Pulse]
#4204	第3軸絶対位置	R	-1000000000 ~ 1000000000[Pulse]
#4300	第4軸指令位置	R	-1000000000 ~ 1000000000[Pulse]
#4301	第4軸機械位置	R	-1000000000 ~ 1000000000[Pulse]
#4302	第4軸偏差量	R	-1000000000 ~ 1000000000[Pulse]
#4303	第4軸最新ステップ払出量	R	-1000000000 ~ 1000000000[Pulse]
#4304	第4軸絶対位置	R	-1000000000 ~ 1000000000[Pulse]
#4400	第5軸指令位置	R	-1000000000 ~ 1000000000[Pulse]
#4401	第5軸機械位置	R	-1000000000 ~ 1000000000[Pulse]
#4402	第5軸偏差量	R	-1000000000 ~ 1000000000[Pulse]
#4403	第5軸最新ステップ払出量	R	-1000000000 ~ 1000000000[Pulse]
#4404	第5軸絶対位置	R	-1000000000 ~ 1000000000[Pulse]

第1軸～第5軸指令位置

論理座標系\*における、SLMコントローラから出力した指令位置です。

第1軸～第5軸機械位置

論理座標系\*における、エンコーダFBを積算した位置情報です。

\* 論理座標系は、原点設定コマンドにより、原点セットされます。

第1軸～第5軸偏差量

指令位置と機械位置の差分です。

第1軸～第5軸最新ステップ払出量

最新の移動指令で出力した移動量です。

第1軸～第5軸絶対位置

機械原点からの絶対位置です。

6 - 4 - 1 1 センサーラッチポジションデータ # 5000 ~

番号	データ名称	属性	設定値
#5000	第1軸センサーラッチ論理座標	R	-1000000000 ~ 1000000000[Pulse]
#5001	第2軸センサーラッチ論理座標	R	-1000000000 ~ 1000000000[Pulse]
#5002	第3軸センサーラッチ論理座標	R	-1000000000 ~ 1000000000[Pulse]
#5003	第4軸センサーラッチ論理座標	R	-1000000000 ~ 1000000000[Pulse]
#5004	第5軸センサーラッチ論理座標	R	-1000000000 ~ 1000000000[Pulse]
#5100	第1軸センサーラッチアブソ座標	R	-1000000000 ~ 1000000000[Pulse]
#5101	第2軸センサーラッチアブソ座標	R	-1000000000 ~ 1000000000[Pulse]
#5102	第3軸センサーラッチアブソ座標	R	-1000000000 ~ 1000000000[Pulse]
#5103	第4軸センサーラッチアブソ座標	R	-1000000000 ~ 1000000000[Pulse]
#5104	第5軸センサーラッチアブソ座標	R	-1000000000 ~ 1000000000[Pulse]

第1軸～第5軸センサーラッチ論理座標  
センサーラッチ入力があった時の論理座標値です。

第1軸～第5軸センサーラッチアブソ座標  
センサーラッチ入力があった時のアブソ座標値です。

1.4

6 - 4 - 1 2 フィードバックカウンタ積算値

各フィードバックカウンタの積算値です。  
S L M起動時の初期値は0です。  
本変数に書き込むことで積算の初期値を設定できます。  
単位は、F B信号のパルスそのもの(任意分周の前)です。

番号	データ名称	属性	設定値
#5500	第1フィードバックカウンタ積算値	R/W	-2147483648 ~ 2147483647[Pulse]
#5501	第2フィードバックカウンタ積算値	R/W	-2147483648 ~ 2147483647[Pulse]

指令値を超えて動かされたとき上記の値までカウントする可能性があります。

2.1

6 - 4 - 1 3 H E X 入力データ # 6000 ~

番号	データ名称	属性	設定値
#6001	H E X入力1	R	0 ~ 15
#6002	H E X入力2	R	0 ~ 15
#6003	H E X入力3	R	0 ~ 15
#6004	H E X入力4	R	0 ~ 15
#6005	H E X入力5	R	0 ~ 15
#6006	H E X入力6	R	0 ~ 15

H E X入力  
汎用入力のON/OFF状態を4 B I Tバイナリ値として、読み込みます。  
各汎用入力とH E X入力の対応は以下の通りです。

データ 名称	ビット			
	D 3	D 2	D 1	D 0
H E X入力1	R I 03	R I 02	R I 01	R I 00
H E X入力2	R I 07	R I 06	R I 05	R I 04
H E X入力3	R I 11	R I 10	R I 09	R I 08
H E X入力4	R I 15	R I 14	R I 13	R I 12
H E X入力5	R I 19	R I 18	R I 17	R I 16
H E X入力6	R I 23	R I 22	R I 21	R I 20
H E X入力7	R I 27	R I 26	R I 25	R I 24
H E X入力8	R I 31	R I 30	R I 29	R I 28
H E X入力9		R I 34	R I 33	R I 32

## 6 - 4 - 1 4 内部データ

#7000～

## 6 - 4 - 1 4 - 1 ドリフト補正值

番号	データ名称	属性	設定値
#7000	第1軸ドリフト補正值	R	-32767 ~ +32767
#7001	第2軸ドリフト補正值	R	-32767 ~ +32767
#7002	第3軸ドリフト補正值	R	-32767 ~ +32767
#7003	第4軸ドリフト補正值	R	-32767 ~ +32767
#7004	第5軸ドリフト補正值	R	-32767 ~ +32767

## 6 - 4 - 1 4 - 2 軸インタロック

番号	データ名称	属性	設定値
#7100	第1軸インタロック	R	0:フリー、1:インタロック
#7101	第2軸インタロック	R	0:フリー、1:インタロック
#7102	第3軸インタロック	R	0:フリー、1:インタロック
#7103	第4軸インタロック	R	0:フリー、1:インタロック
#7104	第5軸インタロック	R	0:フリー、1:インタロック

## 6 - 4 - 1 4 - 3 軸ネグレクト

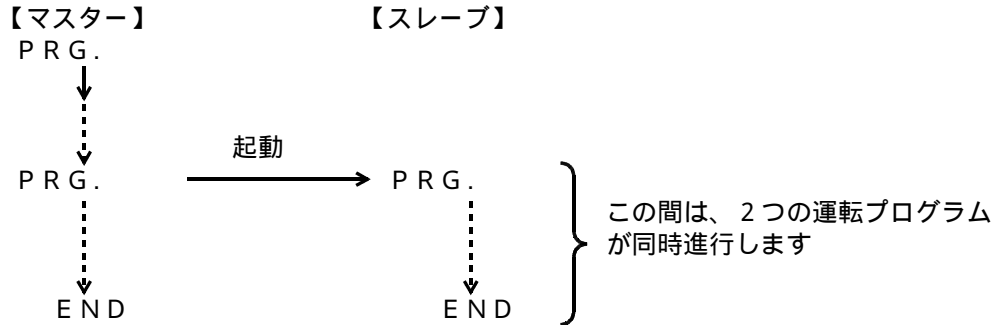
番号	データ名称	属性	設定値
#7200	第1軸インタロック	R	0:フリー、1:ネグレクト
#7201	第2軸インタロック	R	0:フリー、1:ネグレクト
#7202	第3軸インタロック	R	0:フリー、1:ネグレクト
#7203	第4軸インタロック	R	0:フリー、1:ネグレクト
#7204	第5軸インタロック	R	0:フリー、1:ネグレクト

## 6 - 5 マルチタスク

2つの運転プログラム、バックグラウンドタスク及び例外処理のためのタスクをユーザが自由に作成できます。各々のタスクの記述は、通常の運転プログラムと同様にテクノコードでおこないます。

### 6 - 5 - 1 2系列運転 マスタープログラム/スレーブプログラム

2つの運転プログラムを並列に進行できます。待ち合わせやお互いのインターロックも可能です。

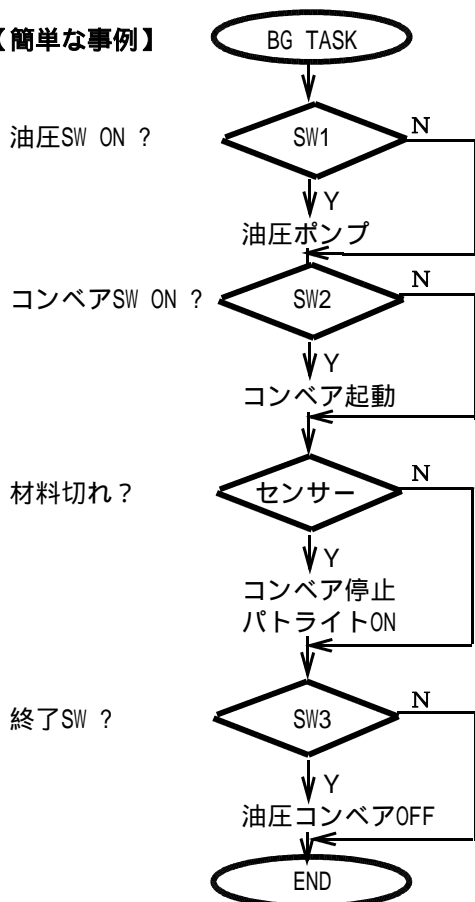


応用例	2ヘッドマシン 2台のマシンの制御 ロボットとローダの制御	組立実装機、加工機 3軸ロボット + 1軸のローダを同時制御
-----	-------------------------------------	-----------------------------------

### 6 - 5 - 2 バックグラウンドタスク(B.G タスク)

自動/手動モードによらず、いつもバックグラウンドタスクが動作します。機械の基本的な制御(油圧やシリンダ、ワーク搬入/搬出など)をB.G タスクで処理すれば、シーケンサは不要です。

#### 【簡単な事例】



#### 入力信号

略称	名称	汎用入力
SW 1	油圧起動	R100
SW 2	コンベア	R101
センサー	材料切れ	R102
SW 3	終了	R103

#### 出力信号

名称	汎用出力
油圧ポンプON	R000
コンベアON	R001
パトライト	R002

```

TIM 0.5 ;
ONR00 JNR00 ;
TIM 0.5 ;
ONR01 JNR01 ;
TIM 0.5 ;
IF #6001&4 ;
OFR01 ;
ONR02 ;
ENDIF ;
IF #6001&8 ;
OFR00 ;
OFR01 ;
ENDIF ;
END
    
```

} OFR01 ONR02 JNR02 ;  
(1ステップでも記述可)

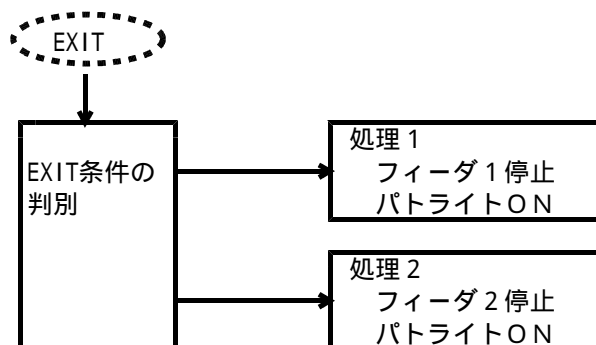
} OFR00 OFR01 JNR03 ;  
(1ステップでも記述可)

### 6 - 5 - 3 E X I Tタスク

運転プログラム中の入力判別でEXIT条件が発生したときに、このタスクが起動します。

例      ネジ締め不良      →      例外処理は、E X I Tタスクで処理します。  
        ワーク / 部品切れ  
        ワーククランプミス

#### 【簡単な事例】

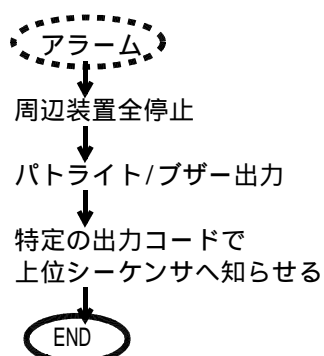


### 6 - 5 - 4 アラームタスク

アラームが発生した時の処理・動作を定義できます。

アラーム情報、各種内部ステータスの変数として読み出せるので、各アラーム要因別に処理することも可能です。

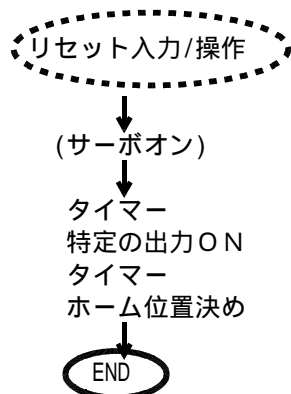
#### 【簡単な事例】



### 6 - 5 - 5 リセットタスク

リセット入力(操作)が発生した場合にリセットタスクが起動します。

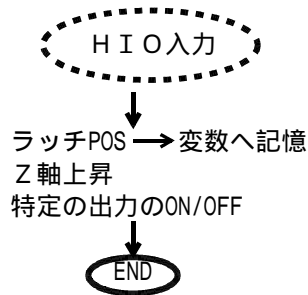
#### 【簡単な事例】



### 6 - 5 - 6 割り込みタスク <開発中>

外部入力を割り込み要因として、割り込みタスクが起動します。  
割り込みタイミングでの位置の計測、特殊な動作シーケンスなどに応用できます。

#### 【簡単な事例】



### 6 - 5 - 7 マルチタスク仕様説明<オプション>

通常のタスク（マスタータスク）の他に、6種のタスクがあり、最大3つのタスク（プログラム）を並列に実行する事が出来ます。

各タスクでそれぞれ起動方法が異なり、ROMSWにて有効にするタスクを選択します。

3つ以上のタスクの実行が重なった時は、下記の優先順位に従って実行します。

（優先順位の低いタスクは実行を停止して、実行タスク数に空きが出来たら再開します。）

各タスクの優先順位

アラームタスク > EXITタスク > 割り込みタスク > リセットタスク > マスタータスク > スレーブタスク > バックグラウンドタスク

1.5

各タスクの特徴

タスク	起動方法	備考
マスター	スタート入力	通常のプログラム
スレーブ	マスタータスクからプログラム番号を指定して起動 (PRG )	単独で実行する時は、マスタータスクとして実行して下さい
バックグラウンド	電源投入	軸移動命令は実行不可 エンドレス実行 (サイクリック) 動作モード、アラーム等に関わらず動作し続けます
リセット	リセット入力	バックグラウンドタスク以外のタスクを強制終了した後、起動されます リセットタスク動作中のリセット入力では、リセットタスクの強制終了になります。
アラーム	アラーム発生時	
EXIT	汎用入力のER/ENRが成立した時	
割り込み	INT0が入力された時	現在未対応

2.1

- 注記
- ・マスタ - /スレーブプログラムのプログラム容量はROMSWの「プログラム分割数」によります。  
マスタ -、スレーブタスク以外のプログラム容量は、それぞれ199ステップ、19レベルです。
  - ・動作の完了していない軸に対して、他のタスクから命令を実行しようとする時、「プログラム実行エラー」になります。
  - ・セッティングPCの"マルチタスク画面"で、各タスクに対して個別にスタート、ストップ、リセットの操作をすることが出来ます。
  - ・各タスクのプログラム記述方法は同一です。

バックグラウンドタスクで実行できない命令

PTP , LIN , SLIN , PTMA , CIRL , CIRR

## 6 - 6 運転プログラム フォーマットエラー / 実行エラー

### 6 - 6 - 1 「テキストフォーマットエラー」

運転プログラムをダウンロードする時にチェックします。  
エラーが発生したステップ番号（行）及び、その前後の行をご確認下さい。

テキストフォーマットエラー要因

<テクノコード / Gコード共通>

- ・プログラム 1 ステップの長さが4096文字以上（コメントは除く）
- ・N番号が10000以上
- ・未定義コードを指定した 

{	有効なコードの詳細については、	}
	6 - 1 - 1 テクノコード指令一覧	
	6 - 1 - 4 Gコード指令一覧	

6 - 2 - 1 付加データアドレス付き数値指定を参照下さい。
----------------------------------
- ・同一コードの2重指定
- ・プログラム 1 ステップ内に、同時に指令できない指令コードが同時に指令されている
- ・文字コードに続く数値指定にて、数字, "-", "+", "." 以外の指定
- ・円弧半径指定が0
- ・補間指令時、F 指定が範囲外、またはF 指定がない  
（F 指定は、モーダルF も含みます）（F の指定範囲は 1 ~ 4000000ppsです）
- ・円弧指令にて、円弧補間を行う2軸のどちらも指令されていない  
（1軸のみ指定した場合、他の軸は0とみなします）
- ・円弧指令にて、半径指定と中心指定が同時に行われていた
- ・半径指定円弧にて、始点と終点が同じ（一周円が指令された）
- ・半径指定円弧にて、始点から終点までの直線距離が円弧直径より大きい  
（終点が円弧上にない）
- ・半径指定円弧にて、平面指定が行われていない
- ・中心指定円弧にて、中心指定が無い
- ・プリ解加減速指定にて、P 1 ~ P 3 以外が指定された

2.1

<テクノコードのみ>

1.5

- ・入出力制御指定 ("ENR", "ER", "JNR", "JR", "SNR", "SR", "WNR", "WR", "OFR", "ONR") を、1 ステップ内に5つ以上行った。
- ・円弧指令以外で平面指定を行った。
- ・"CALL", "JMP" 指令以外でラベル指定を行った。
- ・半径指定円弧にて、平面指定が間違っている

- ・プリ解加減速指定にて、加速("P1", "P3")が指定された時、前ステップのFが現ステップのFより大きいか、同じ
- ・プリ解加減速指定にて、減速("P2", "P3")が指定された時、次ステップのFが現ステップのFより大きいか、同じ
- ・バックグラウンドタスクにて、"PTP", "PTPA", "PTPB", "LIN", "LINA", "LINB", "SLIN", "CIRR", "CIRL", "CIRRA", "CIRLA", "SPIN", "PTMA"指令,または、ポイントデータ指定が行われた
- ・"CALL", "JMP"指令にて、呼び出しラベルの指定がない
- ・"CALL"指令にて、呼び出し回数が50000回以上
- ・D N C 運転にて、"CALL", "JMP", "PTMA"指令、ラベル指定、ポイントデータ指定を行った
- ・"PTMA"指令にて、ポイント指定がない
- ・スレーブタスクプログラム中にて、スレーブタスクプログラム起動指令を指令した。
- ・スレーブタスクプログラム起動指令にて、1 2 以上のプログラム番号を指定した。
- ・演算指令ステップにて、左辺値に対して単項演算子を付加した
- ・プログラムステップのラベルが重複している
- ・総ラベル数以上のラベルを設定した。
- ・プログラムステップに、指令コードがない
- ・"CALL", "JMP"指令にて、呼び出し先ラベルがない
- ・即値指定の"PTMA"指令にて、指定しているポイントがポイントテーブルにない

#### < Gコードのみ >

- ・Mコード指定にて、2 5 5 以上を指定
- ・Sコード指定にて、±320000以上を指定
- ・バックグラウンドタスクにて、"G0", "G1", "G2", "G3", "G28", "G31", "G100", "G120"指令,または、ポイントデータ指定が行われた
- ・アブソ指令に、プリ解加減速を指定した
- ・D N C 運転にて、"M98", "M99", "G100"を指令した
- ・"G100"指令にて、ポイントの指定がない
- ・ポイントデータ指定にて、ポイント番号指定が連番になっていない
- ・"G04"指令にて、Pコード指定がない
- ・"G43"指令にて、Hコード指定がない
- ・"M98"指令にて、呼び出し先がプログラムに存在しない
- ・"G100"指令にて、指定しているポイントがポイントテーブルに存在しない



## 6 - 6 - 2 「プログラム変換計算エラー」

運転プログラムをダウンロードする時にチェックします。

プログラム変換計算エラー要因

- ・円弧補間ステップにて、内部計算で算出された円弧半径が 0 になった

## 6 - 6 - 3 「プログラムバッファオーバーフロー」

運転プログラムをダウンロードする時にチェックします。

プログラムバッファオーバーフロー要因

- ・プログラムステップ数が最大ステップ数を超えた
- ・プログラムステップとポイントデータの合計がプログラム容量を超えた

## 6 - 6 - 4 「プログラム実行エラー」

実運転中にチェックします。

プログラム実行エラー要因

<テクノコード / Gコード共通>

- ・サーボOFF時に、以下の指令を行った
  - PTP移動(ポイント位置決め) "PTPx" / "G0", "G28"
  - 直線補間移動 "LINx", "SLIN" / "G1", "G31"
  - 円弧補間移動 "CIRL", "CIRR", "CIRLA", "CIRRA" / "G2", "G3"
  - 工具長補正指令 "THSET", "THOFF" / "G43", "G49"
  - 論理座標設定指令 "CSET" / "G92"
  - ポイント位置決め指令 "PTMA" / "G100"
  - 接線制御指令 "TURN" / "G112"
  - 無限回転軸回転指令 "SPIN" / "G120"
- ・以下の移動指令の軸指定にて、SPIN動作("SPIN" / "G120")中の軸を指定した
  - PTP移動(ポイント位置決め) "PTPx" / "G0", "G28"
  - 直線補間移動 "LINx", "SLIN" / "G1", "G31"
  - 円弧補間移動 "CIRL", "CIRR", "CIRLA", "CIRRA" / "G2", "G3"
  - ポイント位置決め指令 "PTMA" / "G100"
- ・以下の移動指令の軸指定にて、インタロック中の軸を指定した
  - PTP移動(ポイント位置決め) "PTPx" / "G0", "G28"
  - 直線補間移動 "LINx", "SLIN" / "G1", "G31"
  - 円弧補間移動 "CIRL", "CIRR", "CIRLA", "CIRRA" / "G2", "G3"
  - ポイント位置決め指令 "PTMA" / "G100"
  - 無限回転軸回転指令 "SPIN" / "G120"
- ・主軸オプション無し状態で、主軸ON/OFF(M3/M4/M5)指令を行った
- ・サブプログラム呼び出し指令("CALL" / "M98")のネストが20レベル上になった
- ・ポイント位置決めオプション無し状態で、ポイント位置決め指令を行った

- ・ポイント位置決め指令で指定されたポイントデータが無い
- ・接線制御オプション無し状態で、接線制御有効/無効、接線制御Z軸回転指令を行った
 

接線制御有効指令	"STNE" / "G110"
接線制御無効指令	"STND" / "G111"
接線制御Z軸回転指令	"TURN" / "G112"
- ・接線制御無効の時に、接線制御Z軸回転指令を行った
- ・接線制御Z軸回転指令にて、接線方向指定無し、かつ、次ブロックに第3軸移動が指令されている
- ・接線制御Z軸回転指令にて、接線方向指定無し、かつ、次ブロックに第1軸、第2軸ともに移動指定が無い
- ・工具長補正オプション無し状態で、工具長補正有効/キャンセル指令を行った
 

工具長補正有効	"THSET" / "G43"
工具長補正キャンセル	"THOFF" / "G49"
- ・センサーラッチオプション無し状態で、センサーラッチ直線補間指令を行った
 

センサーラッチ直線補間	"SLIN" / "G31"
-------------	----------------
- ・MOK入力有効状態で、MOK入力がOFFでMコード実行完了した
- ・内部解析の円弧指令ステップで、始点から終点までの直線距離が円弧直径より大きい（終点が円弧上にない）
- ・内部解析の半径指定円弧指令ステップで、始点と終点が同じ（一周円が指令された）
- ・内部解析の中心指定円弧指令ステップで、円弧始点の半径と円弧終点の半径が2パルス以上違っている
- ・他のタスクで使用している軸に対して、指令を行った。

#### <テクノコードのみ>

- ・移動量のレジスタ指定、ポイント番号のレジスタ指定、IFステップ、演算ステップにて、指定された番号のマクロ変数が存在しない。