

SEL言語

説明書 第5版

STEP	A/O	N	OP-CODE	OP-RND1	OP-RND2	POST
1			HOME	11		
2			VEL	100		
3			ACC	0.2		
4			TAG	1		
5			LET	300	0	
6			LET	301	0	
7			OFST	11	0	
8			MOVL	18		
9			WTON	17		
10			BTOF	312		
11			TAG	2		
12			OFST	11	0	
13			MOVL	17		
14			EXSR	1		
15			OFST	1	#300	
16			OFST	10	#301	
17			MOVL	1		
18			EXSR	1		
19			ADD	300	20	
20			CPEQ	300	80	600
21		N600	GOTO	2		
22			LET	300	0	
23			ADD	301	30	
24			CPEQ	301	120	601
30			WTON	16		
31			BTNT	311		
32			TIMW	0.5		
33			BTOF	310		
34			WTON	15		
35			EDSR			
36						

目 次

1. SEL 言語で取り扱う数値と記号	1
1.1 SEL 言語で取り扱う数値一覧表	1
1.2 SEL 言語で使われる記号	3
2. 入出力ポート	4
2.1 入力ポート	4
2.2 出力ポート	4
2.3 スーパー SEL タイプ入出力ポート一覧	5
2.4 DS タイプ入出力ポート一覧	6
3. フラグ	8
4. 変数	9
4.1 変数の意味	9
4.2 変数の種類	10
5. タグ	12
6. サブルーチン	13
7. 軸の指定	14
7.1 軸 No. と表示	14
7.2 軸パターン	15
8. SEL 言語の構造	16
8.1 ポジション部	16
8.2 命令部	17
9. パラメーター一覧表	19
9.1 多軸共通パラメータ	19
9.2 単軸共通パラメータ	21
9.3 軸別パラメータ	22
10.機能別 SEL 言語命令コード一覧表	24
11.アルファベット順 SEL 言語命令コード一覧表	27

12. SEL 言語	30
12.1 数値演算命令	30
12.2 算術演算命令	32
12.3 関数演算命令	35
12.4 論理演算命令	38
12.5 比較演算命令	41
12.6 タイマー命令	42
12.7 入出力・フラグ操作命令	44
12.8 プログラム制御命令	50
12.9 タスク管理命令	53
12.10 資源管理命令	57
12.11 ポジション操作命令	58
12.12 アクチュエータ制御宣言	63
12.13 アクチュエータ制御命令	73
12.14 構造化 IF 命令	84
12.15 構造化 DO 命令	87
12.16 多分岐命令	89
12.17 外部入出力命令	93
12.18 ストリング処理命令	96
13. エラーコード一覧表	102



はじめに

SEL 言語は、数多くあるロボット言語の中で最もシンプルなタイプの言語です。
「高度な制御を簡単な表現で実現する」という難問を SEL 言語が見事に解決しました。

一般的に多く使われている BASIC（ベーシック）を基本とする言語は文章形式であり、文章の解釈に時間がかかります。高度な表現になればなるほど、インタプリタ*¹ではリアルタイム制御に適應できなくなります。そこでコンパイル*²などという余分な操作が増えます。さらに MS-DOS の知識、疑似命令などどんどん難しくなります。

同じことを実現するには、簡単にできるほうが良いに決まっています。
さあ、シンプルそしてインタプリタでありながら高速動作ができる SEL 言語を駆使して、高度な制御を実現しようではありませんか。

注) * 1: インタプリタとは、命令を実行するときにコンピュータ言語に翻訳しながら動作することをいいます。

注) * 2: コンパイルとは、あらかじめコンピュータ言語に翻訳しておくことをいいます。

1. SEL 言語で取り扱う数値と記号

1.1 SEL 言語で取り扱う数値一覧表

プログラム上で必要な各種機能を数値を使って表わします。

機 能	スーパーSELタイプ		DSタイプ		備 考
	グローバル領域	ローカル領域	グローバル領域	ローカル領域	
入力ポート	000～287 (288点)		001～015 (15点)		機種により異なります
出力ポート	300～587 (288点)		300～307 (8点)		機種により異なります
フラグ	600～887 (288点)	900～999 (100点)	600～887 (288点)	900～999 (100点)	
変数 (整数)	200～299 (100点)	1～99 (99点)	200～299 (100点)	1～99 (99点)	99はINB,OUTBで使用
変数 (実数)	300～399 (100点)	100～199 (100点)	300～399 (100点)	100～199 (100点)	199はPPUT,PGETで使用
ストリング	300～999 (700字)	1～199 (199字)			
タグNo.		1～64 (64点)		1～64 (64点)	
サブルーチンNo.		1～64 (64点)		1～64 (64点)	
軸No.	1～8		1		機種により異なります
軸パターン	2進数8桁				指定軸の桁を1にします
ポジションNo.	1～2000		1～500		
プログラムNo.	1～64		1～32		
ステップNo.	1～3000		1～1000		
タスクレベル	1～5		1～5		
資源No.	1～9		1～9		
チャンネルNo.	1～2				機種により異なります
タイマー		1		1	
	どのプログラムからも共通に使用することができます	各プログラムで個別に参照します プログラム起動時にクリアされます	どのプログラムからも共通に使用することができます	各プログラムで個別に参照します プログラム起動時にクリアされます	変数99,199は、本システムが演算用に使用する特別な変数です この2つの変数の汎用的な使用は避けて下さい

注) 変数99および199は、本システムが演算用に使用する特別なレジスターです。

● バッテリーバックアップ領域について

電源再投入時のコントローラ内部の状態は、バッテリーバックアップしている領域以外はクリアされます。(非常停止時も同様です)

- ・ プログラム (停止)
- ・ 出力ポート (クリア)
- ・ ローカルフラグ (クリア)
- ・ ローカル変数 (クリア)
- ・ 原点位置 (クリア)
- ・ グローバルフラグ (保持)
- ・ グローバル変数 (保持)

● SEL 言語で取り扱える数値の範囲について

整数および実数が取り扱えます。しかし、これらには以下の制約事項がありますので十分に注意して下さい。

① コントローラ内部での扱について (コンピュータとして)

整数は±2,147,483,648 また実数は単精度浮動小数点数として $\pm 3.4 \times 10^{38}$ が理論的に扱えます。

② 現実としての制約

開発初期からの基本的プログラミングツールをLCD (液晶表示) のティーチングボックスとしてきた結果、プログラムからの入出力に関しては以下の制約があります。

整数－9,999,999～99,999,999 実数－999,999～9999,999 または－.999999～.99999999 つまり符号小数点を含めて8桁の値がプログラムから取り扱える数値の限度です。また浮動小数点で演算されますと有効数字は7桁までしか保証されませんし、それには浮動小数点特有の誤差が含まれることがあることをご承知置き下さい。

③ 位置データについて

位置データは内部的には整数データとして扱っていますが、それを演算の過程で実数に取り込むと実数の扱いになります。±9999,999が扱えれば本来は問題ないのですが、これを位置データとしてでなく一般データとして内部演算した結果の場合 (乗除算を繰り返しますと) は最後の桁の精度は問題になってくる場合があります。

ご使用に当たってこれらの点を十分に配慮してください。特に実数を用いた比較演算でCPEQ命令を用いますと一致を見ることはほとんどありません。この場合は大小関係を併用して見るコマンドCPLE / CPGE を用いる必要があります。

1.2 SEL 言語で使用する記号

SEL 言語で使用する記号の内容を説明します。

記号	意 味	説 明
ZR	ゼロ	演算結果が0になった時、出力をオン
EQ	イコール	操作1＝操作2が成立する時、出力をオン
NE	ノット・イコール	操作1≠操作2が成立する時、出力をオン
GT	グレーター・ザン	操作1＞操作2が成立する時、出力をオン
GE	グレーター・イコール	操作1≥操作2が成立する時、出力をオン
LT	レス・ザン	操作1＜操作2が成立する時、出力をオン
LE	レス・イコール	操作1≤操作2が成立する時、出力をオン
PE	ポジション・エンド	動作完了時、出力をオン（パス、円、円弧の連続移動時は2ポイント手前でオン）
CP	コンプリート	命令完了時、出力をオン
TU	タイム・アップ	時間経過時、出力をオン
XX	ポジションデータ無し	ポジションに有効な値が無い時、出力をオン
—		以上の記号に下線が引かれているものは、出力を必要とします。
ON	オン	オン
OF	オフ	オフ
NT	ノット	反転
FN	フォワード・オン	指定した入出力・フラグがオンの間に前進
FF	フォワード・オフ	指定した入出力・フラグがオフの間に前進
BN	バックワード・オン	指定した入出力・フラグがオンの間に後退
BF	バックワード・オフ	指定した入出力・フラグがオフの間に後退

2. 入出力ポート

2.1 入力ポート

リミットスイッチ、センサースイッチ等の入力ポートとして使用します。

入力番号割付け	スーパー SEL タイプ
001～023	標準
024～047	オプション
048～071	オプション
072～095	オプション

入力番号割付け	DS タイプ
001～015	標準

2.2 出力ポート

各種出力ポートとして使用します。

出力番号割付け	スーパー SEL タイプ
300～323	標準
324～347	オプション
348～371	オプション
372～395	オプション

出力番号割付け	DS タイプ
300～307	標準

2.3 スーパー SEL タイプ入出力ポート一覧表

ポートNo.	機 能	説 明
000	外 部 起 動 入 力	オンになった瞬間に指定されたプログラムを実行します。
001	汎 用 入 力	ユーザーが自由に使うことができる入力ポートです。
002	非 常 停 止 b 接 点 入 力	オフになるとコントローラは非常停止します。
003	シ ス テ ム 予 約	使用不可（将来新機能が付く時に使用されるポートです）
004	シ ス テ ム 予 約	使用不可（将来新機能が付く時に使用されるポートです）
005	汎 用 入 力	ユーザーが自由に使うことができる入力ポートです。
006	汎 用 入 力	ユーザーが自由に使うことができる入力ポートです。
007	汎 用 入 力	ユーザーが自由に使うことができる入力ポートです。
008	PRG No.01（汎用入力）	10^0 位 008～014をBCDコードとして起動プログラムNo.を指定します。 入力ポート000（外部起動入力）がオンになった瞬間に指定されていたプログラムが実行されます。 10^1 位
009	PRG No.02（汎用入力）	
010	PRG No.04（汎用入力）	
011	PRG No.08（汎用入力）	
012	PRG No.10（汎用入力）	
013	PRG No.20（汎用入力）	
014	PRG No.40（汎用入力）	
015 ┆ 023	汎 用 入 力	ユーザーが自由に使うことができる入力ポートです。
024 ┆ 287	拡 張 入 力	拡張I/Oカードユニット、高速入力ユニットを増設することにより使用可能になる汎用入力ポートです。
300	非常停止／アラーム出力	非常停止時、エラー時にオンになります。
301	レ デ ィ ー 出 力	コントローラの準備ができている時にオンになります。
302 ┆ 323	汎 用 出 力	ユーザーが自由に使うことができる出力ポートです。
324 ┆ 587	拡 張 出 力	拡張I/Oカードユニットを増設することにより使用可能になる汎用出力ポートです。

2.4 DS タイプ入出力ポート一覧表

プログラムモード

ポートNo.	機 能	説 明
	PRG No.01	10^0 位 BCDコードとして起動プログラムNo.を指定します。 10^1 位 入力ポート000（外部起動入力）がオンになった瞬間に指定されていたプログラムが実行されます。
	PRG No.02	
	PRG No.04	
	PRG No.08	
	PRG No.10	
	PRG No.20	
	N C	
	CPU リセット 入 力	コントローラの再起動をします。
000	外 部 起 動 入 力	オンになった瞬間に指定されたプログラムを実行します。
001 ┆ 015	汎 用 入 力	ユーザーが自由に使うことができる入力ポートです。
300	非常停止／アラーム出力	非常停止時、エラー時にオンになります。
301	レ デ ィ ー 出 力	コントローラの準備ができている時にオンになります。
302 ┆ 307	汎 用 出 力	ユーザーが自由に使うことができる出力ポートです。



ポジショナーモード

ポートNo.	機 能	説 明
	N C	ポジショナーモード時はNCとなります。
	N C	
	N C	
	N C	
	N C	
	N C	
	N C	
	CPU リセット 入 力	コントローラの再起動をします。
000	外 部 起 動 入 力	オンになった瞬間に指定されたポジションへ移動します。
001	ホ ー ル ド 入 力	オンするとサーボアクチュエータが減速停止します。
002 └ 003	N C	002～003は、ポジショナーモード時はNCとなります。
004	ポジション No.1 入力	10⁰ 位 004～014をBCDコードとしてポジションNo.を指定します。 10¹ 位 入力ポート000（外部起動入力）がオンになった瞬間に指定されていたポジションに移動します。 10² 位 ポジションNo.の指定がない場合は、原点復帰を行います。
005	ポジション No.2 入力	
006	ポジション No.4 入力	
007	ポジション No.8 入力	
008	ポジション No.10 入力	
009	ポジション No.20 入力	
010	ポジション No.40 入力	
011	ポジション No.80 入力	
012	ポジション No.100 入力	
013	ポジション No.200 入力	
014	ポジション No.400 入力	
015	N C	015は、ポジショナーモード時はNCとなります。
300	非常停止／アラーム出力	非常停止時、エラー時にオンになります。
301	レ デ ィ ー 出 力	コントローラの準備ができている時にオンになります。
302	位 置 決 完 了 出 力	ポジション移動が完了するとオンになります。
303 └ 307	N C	303～307は、ポジショナーモード時はNCとなります。

3. フラグ

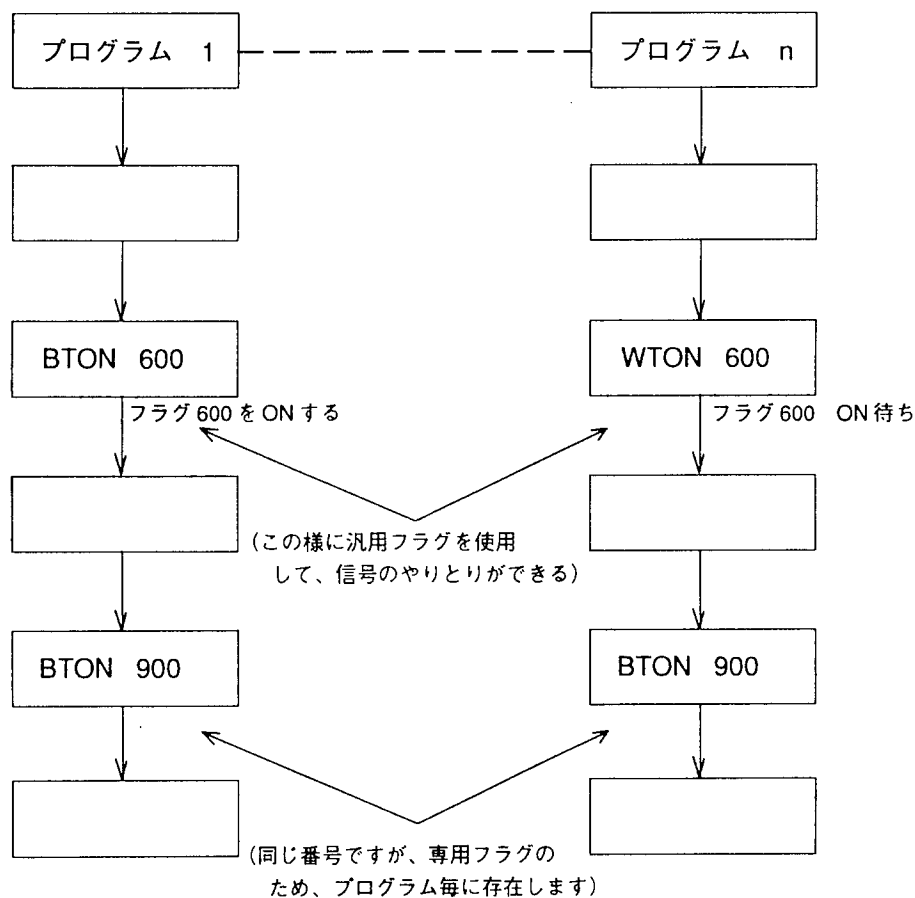
フラグは旗（Flag：フラッグ）の意味ですが、中身は“メモリー”で、データのセット・リセットを行います。シーケンサでいう“補助リレー”にあたります。

フラグは、すべてのプログラムで使用できる汎用フラグ（グローバルフラグ）600～887番と、個々のプログラムでのみ使用できる専用フラグ（ローカルフラグ）900～999番の2種類があります。

汎用フラグ（グローバルフラグ）は、電源をOFFしても保存（バッテリーバックアップ）されます。

専用フラグ（ローカルフラグ）は、電源OFFでクリアされます。

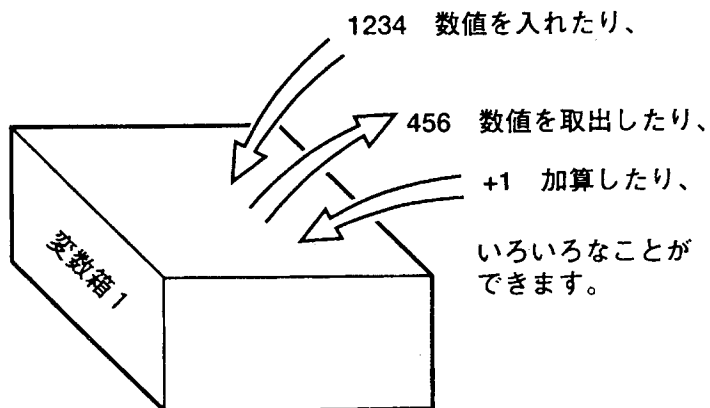
フラグ番号	600～887	全プログラムで使用可	“汎用フラグ（グローバルフラグ）”
フラグ番号	900～999	各プログラムで個別に使用可	“専用フラグ（ローカルフラグ）”



4. 変数

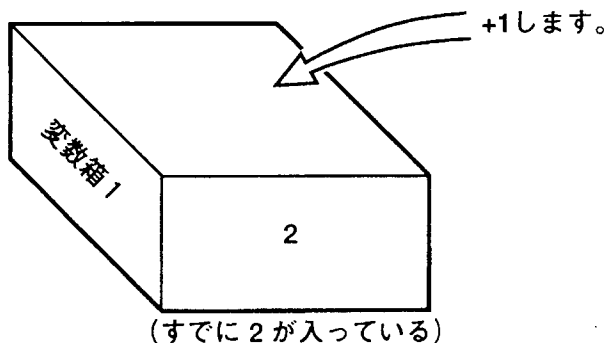
4.1 変数の意味

変数とは、ソフトウェアの専門用語です。わかりやすく表現しますと“数値を入れる箱”と考えてください。数値を入れたり、数値を取り出したり、加減算したり、いろいろなことができます。



命令	操作 1	操作 2
ADD	1	1

この命令の場合は、図のように変数1の箱にすでに2が入っていれば、+1されて結果は3になります。



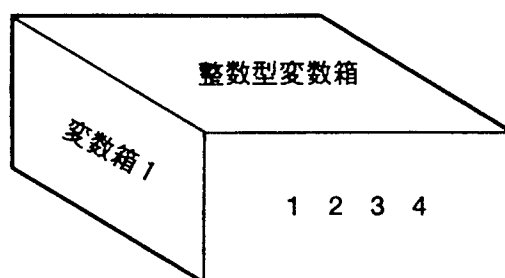
4.2 変数の種類

変数は、次のように2種類あります。

①整数型変数

小数点以下が取り扱えない変数です。

[例] 1234



整数型変数番号	200～299	全プログラムで使用可	“グローバル整数変数”
整数型変数番号	1～99	各プログラムで個別に使用可	“ローカル整数変数”

(注) 変数 99 は、本システムが使用する整数演算用の特別なレジスターです。

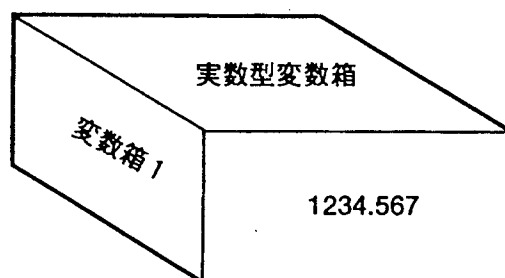
プログラム上で入力できるのは、-9,999,999 から 99,999,999 までです。

②実数型変数

実際の数値のことで、小数点以下も取り扱うことができる変数です。

[例] 1234.567

↑
(小数点)



実数型変数番号	300～399	全プログラムで使用可	“グローバル実数変数”
実数型変数番号	100～199	各プログラムで個別に使用可	“ローカル実数変数”

(注) 実数 199 は、本システムが使用する実数演算用の特別なレジスターです。

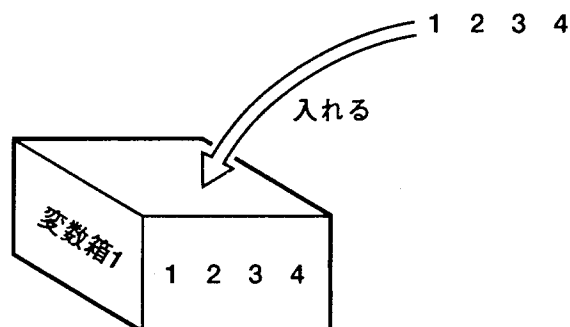
プログラム上で入力できるのは、-99,999.9 から 999,999.9 までの（符号付8桁）の数です。

③ “*”（アスタリスク）のついた変数

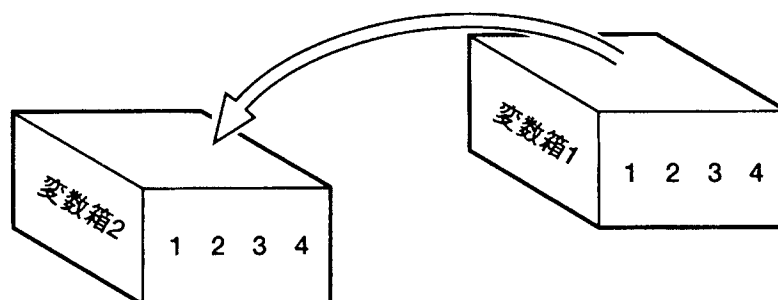
“*”（アスタリスク）は、変数を指定する時に使います。

次の例は、変数箱2へ変数箱1の中身を入れることになります。変数箱1に、“1234”が入っていれば、変数箱2へ“1234”が入ることになります。

命令	操作1	操作2
LET	1	1234

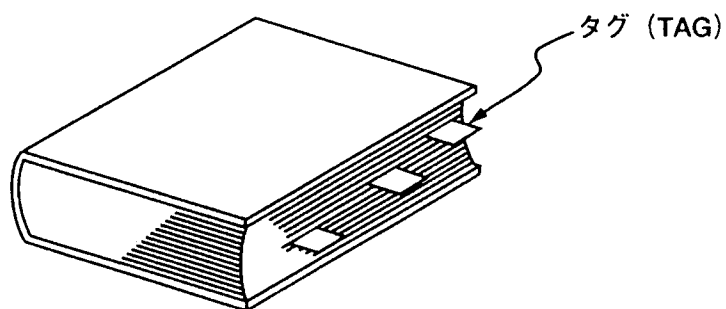


命令	操作1	操作2
LET	2	* 1



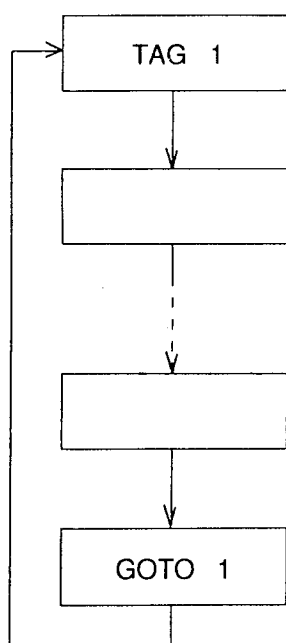
5. タグ (TAG)

“タグ (TAG)”とは“見出し”のことです。
よく本で見たいところにラベルを貼ります。これと意味あいは同じです。
ジャンプ命令“GOTO”で指定される飛び先が“タグ (TAG)”になります。



命令	操作 1
TAG	タグ No. (1～64 の整数値)

各プログラムで個別に使用可。



6. サブルーチン

プログラムの中で何回も繰り返し使用する部分を切り出して、“サブルーチン”として登録しておけば、少ないステップ数で処理できます。(最大15までネスティングできます)

各プログラムで個別に使用可。

命令	操作 1
EXSR	サブルーチン No. (1~64 の整数値、変数も可)

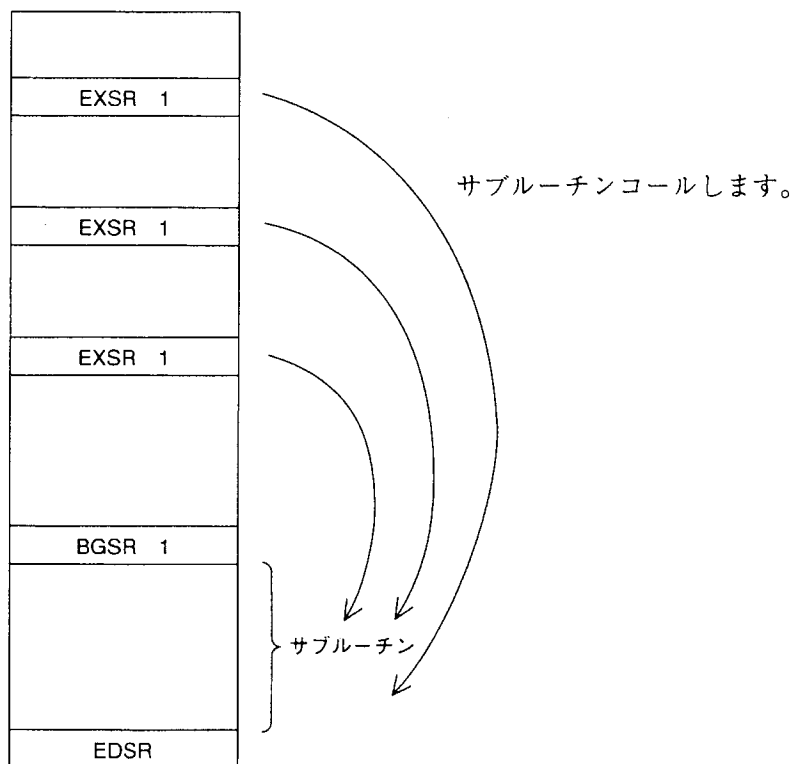
サブルーチン実行命令

命令	操作 1
BGSR	サブルーチン No. (1~64 の整数値)

サブルーチン開始宣言

命令	操作 1
EDSR	_____

サブルーチン終了宣言



7. 軸の指定

軸の指定をするには、軸 No. で行う場合と軸パターンで行う場合があります。

7.1 軸 No. と軸の表示

多軸を表現するために次のように表示します。

軸 No.	軸の表示
1 軸	1 軸
2 軸	2 軸
3 軸	3 軸
4 軸	4 軸
5 軸	5 軸
6 軸	6 軸
7 軸	7 軸
8 軸	8 軸



このような表示にセットされていますが、パラメータにて1文字の英数字に変更可能です。

何軸かのうちのひとつの軸だけを指定する場合には、軸 No. にて行います。

- ・ 軸 No. で指定する命令
BASE, PPUT, PGET

(注) DS タイプは、1 軸のみの表示となります。



7.2 軸パターン

どの軸を使用するかを、“1”あるいは“0”を用いて表わします。

	(上位)					(下位)		
軸 No.	8 軸	7 軸	6 軸	5 軸	4 軸	3 軸	2 軸	1 軸
使用する	1	1	1	1	1	1	1	1
使用しない	0	0	0	0	0	0	0	0

[例] 1 軸と 2 軸を使用する場合

2 軸
↓
00000011・・・となります。(前にある 0 は必要ありません。0 をとり、11 とします)
↑
1 軸

[例] 1 軸と 8 軸を使用する場合

8 軸
↓
10000001・・・となります。(この場合、0 は 8 軸の位置を表わすために必要です)
↑
1 軸

何軸かのうち、同時にいくつかの軸を指定する必要がある場合には、軸パターンにて行います。

・軸パターンで指定する命令

OFST,GRP,SVON,SVOF,HOME,JFWN,JFWF,JBWN,JBWF,STOP,PTST,PRED

(注) DS タイプの軸パターンは、1 軸のためあらかじめ設定されています。

8. SEL 言語の構造

SEL 言語は、ポジション部（ポジションデータ＝座標値、他）と命令部（アプリケーションプログラム）に分かれています。

8.1 ポジション部 (Position)

ポジション部には、座標値、速度、加速度を格納します。

ポジション No.	速度	加速度	* 2 標準 0.3G							
			1 軸	2 軸	3 軸	4 軸	5 軸	6 軸	7 軸	8 軸
1										
2										
3										
4										
1997										
1998										
1999										
2000										

* 1 アクチュエータの機種によって異なります。

* 2 ポジションデータに速度、加速度を設定した場合、アプリケーションプログラムに設定したデータより優先されます。アプリケーションのデータを有効にしたい場合は、X、XXまたは0と設定します。

（注）DS タイプは、1 軸のみとなります。また、ポジション No. は 500 までとなります。

8.2 命令部

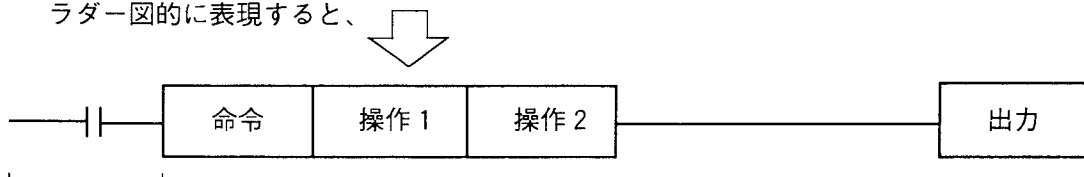
SEL 言語の最大の特徴は、極めてシンプルな命令の構造にあります。構造がシンプルなため、コンパイル（コンピュータ言語に翻訳）する必要がなく、インタプリタ（翻訳しながら動作する）で、高速動作します。

8-2-1. SEL 言語の構造

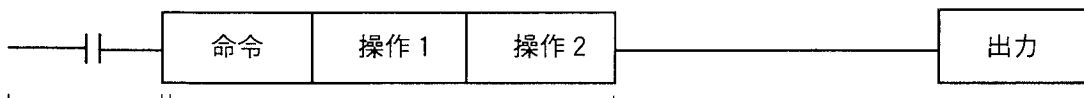
命令の 1 ステップは、次の構造になっています。

拡張条件 (AND・OR)	入力条件 ((入出力・フラグ)	命令・宣言			出力部 (出力ポート・フラグ)
		命令・宣言	操作 1	操作 2	

ラダー図的に表現すると、



(1) 命令の前にある条件は、極めて巧妙な仕掛けで BASIC (ベーシック) 言語の “IF～THEN・・・” に相当しています。



IF～THEN・・・・・・・・・・・・・・・・・・・・・・・・ELSE・・・・・・・・



次へステップ

① 入力条件が成立した時は命令を実行し、出力指定があれば出力ポートを ON し、入力条件が成立しない時は後の命令の如何 (ex.WTON,WTOF) を問わずに次のステップに進みます。

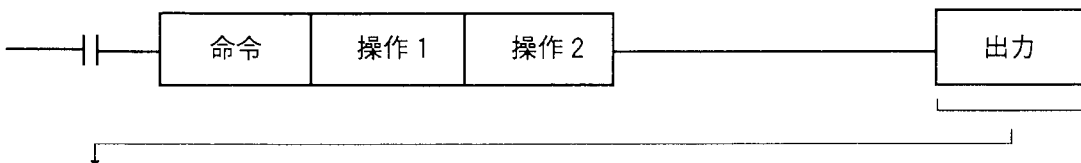
当然出力ポートには何も起こりませんが注意が必要です。

② 条件設定のない場合には、無条件に命令を実行します。

③ 条件を逆条件 (一般的にいう b 接点 ) で使用したい時は、条件のところに “N” (NOT) をつけます。

④ 条件には、入力ポート、出力サポートフラグが使用できます。

(2) 命令、操作 1、操作 2 の後にある出力は、次のような動作となります。



① アクチュエータ動作制御命令等では、命令実行開始と同時に OFF となり、実行完了で ON となります。演算命令等では、結果がある特定の値になると ON し、それ以外では OFF となります。

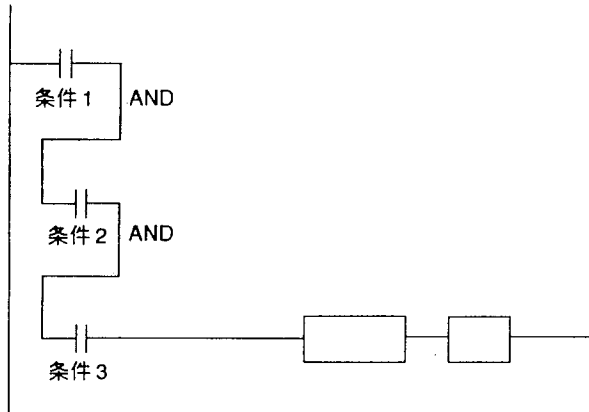
② 出力部には、出力ポートとフラグが使用できます。

8-2-2. 拡張条件

条件を複雑に組み合わせることも可能です。

AND 拡張

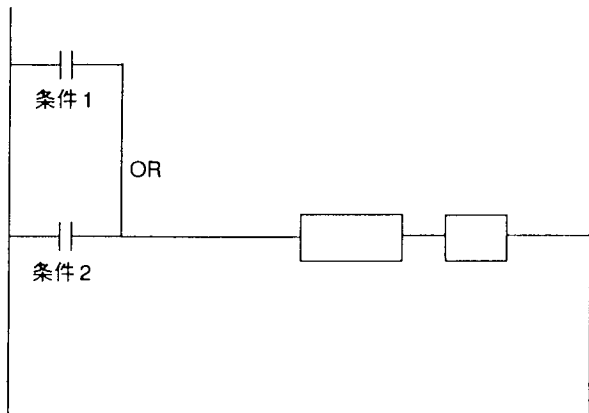
(ラダー図的表現)



(SEL 言語)

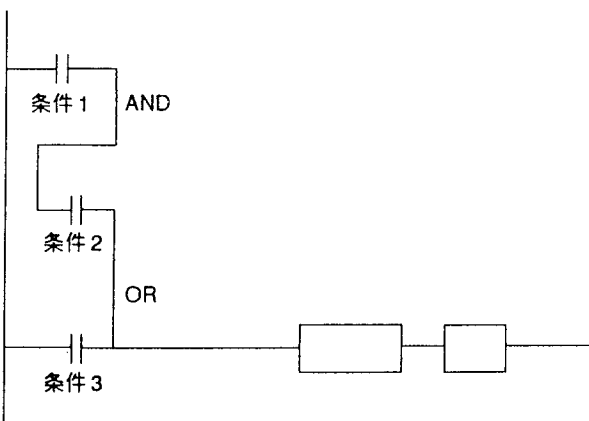
拡張条件	入力条件	命 令			出力部
		命令	操作 1	操作 2	
	条件 1				
AND	条件 2				
AND	条件 3	命令	操作 1	操作 2	

OR 拡張



拡張条件	入力条件	命 令			出力部
		命令	操作 1	操作 2	
	条件 1				
OR	条件 2	命令	操作 1	操作 2	

AND 拡張と OR 拡張



拡張条件	入力条件	命 令			出力部
		命令	操作 1	操作 2	
	条件 1				
AND	条件 2				
OR	条件 3	命令	操作 1	操作 2	

9. パラメーター一覧表（参考値を表示します。出荷時の設定は機種毎に異なります。）

9.1 多軸共通パラメータ

(1) サーボパラメータ

No.	パラメータ名	設定値	内容
1	Axis Size	8	軸数
2	Numerator	1	分子
3	Denominator	1	分母
4	Override (%)	100	オーバーライド
5	Acceler (G)	0.30	加速度係数 (G)
6	Acceler Max (G)	1.00	最大加速度係数 (G)
7	Drive Vel (mm/s)	100	運転速度 (mm/s)
8	Drive Vel Max (mm/s)	1000	最大速度 (mm/s)

(2) プログラムパラメータ

No.	パラメータ名	設定値	内容
1	AutoStart Program	0	自動スタートプログラム No.
2	Emergency Program	0	非常停止プログラム No.
3	Program Size	64	プログラム本数
4	Task Size	16	タスク本数
5	Step Size	3000	プログラムステップ数
6	Time Slice	0.01	タイムスライスチック値

(3) ポイントパラメータ

No.	パラメータ名	設定値	内容
1	Point Size	2000	ポイントデータ数

(4) 円弧パラメータ

No.	パラメータ名	設定値	内容
1	Circle Angle (°)	15.0	スライス角度 (°)
2	Circle Delt (mm/s)	0	速度増分 (mm/s)

(5) シリアル I/O パラメータ

No.	パラメータ名	設定値	内容
1	Terminal ID	99	マルチドロップ局コード
2	Time Out (S)	0	タイムアウト (S)
3	Baud Rate (bit/s)	3	ボーレート (bit/s)
4	Char Length	0	キャラクター長
5	Parity	0	パリティ
6	Stop Bit	0	ストップビット

(6) パラレル入力ポートパラメータ

No.	パラメータ名	設定値	内容
1	Device No.	1~3	デバイス No.
2	Unit ID	1	ユニット No.
3	Scan Time (ms)	1	スキャンタイム (ms)
4	Unit ID	1	ユニット No.
5	Scan Time (ms)	1	スキャンタイム (ms)
6	Unit ID	1	ユニット No.
7	Scan Time (ms)	1	スキャンタイム (ms)
8	Unit ID	1	ユニット No.
9	Scan Time (ms)	1	スキャンタイム (ms)

9.2 単軸共通パラメータ

(1) サーボパラメータ

No.	パラメータ名	設定値	内容
1	Axis Size	1	軸数
2	Numerator	1	分子
3	Denominator	1	分母
4	Override (%)	100	オーバーライド
5	Acceler (G)	0.30	加速度係数 (G)
6	Acceler Max (G)	1.00	最大加速度係数 (G)
7	Drive Vel (mm/s)	100	運転速度 (mm/s)
8	Drive Vel Max (mm/s)	1000	最大速度 (mm/s)

(2) プログラムパラメータ

No.	パラメータ名	設定値	内容
1	AutoStart Program	0	自動スタートプログラム No.
2	Emergency Program	0	非常停止プログラム No.
3	Program Size	32	プログラム本数
4	Task Size	8	タスク本数
5	Step Size	1000	プログラムステップ数
6	Time Slice	0.01	タイムスライスチック値

(3) ポイントパラメータ

No.	パラメータ名	設定値	内容
1	Point Size	500	ポイントデータ数

(4) シリアル I/O パラメータ

No.	パラメータ名	設定値	内容
1	Terminal ID	99	マルチドロップ局コード
2	Time Out (S)	0	タイムアウト (S)
3	Baud Rate (bit/s)	3	ボーレート (bit/s)
4	Char Length	0	キャラクター長
5	Parity	0	パリティ
6	Stop Bit	0	ストップビット

9.3 軸別パラメータ

(1) 軸別サーボパラメータ

No.	パラメータ名	設定値	内容
1	Axis Name	1~8	軸名
2	Servo Service	400	サーボサービス回数 (回数/s)
3	Numerator	1	分子
4	Denominator	1	分母
5	Override (%)	100	オーバーライド (%)
6	Acceler (G)	0.30	加速度係数 (G)
7	Jog Vel (mm/s)	30	ジョグ速度 (mm/s)
8	Pend Band	10	位置決め幅 (パルス)
9	Soft Limit Offset	2.00	ソフトリミットオフセット
10	Soft Limit (+)	9999	ソフトリミット (+)
11	Soft Limit (-)	0	ソフトリミット (-)

(2) 軸別原点復帰パラメータ

No.	パラメータ名	設定値	内容
1	Home Dir	0	原点復帰方向
2	Home Type	0	原点復帰方法
3	Home Sequence	1	順番
4	Home Sw Pol	1	リミット入力極性
5	Home Z Edge	1	Z相検出エッジ
6	Home Creep Vel	100	クリープ速度
7	Home Back Vel	10	追い込み速度
8	Home Z Vel	5	Z相サーチ速度
9	Home Offset	0	オフセット移動量
10	Home Deviation	667	押付け偏差 (パルス)
11	Home Current	60	電流制限



(3) 軸別モータパラメータ

No.	パラメータ名	設定値	内容
1	Motor RPM Max	4000	モータ最大回転数
2	Encoder Pulse	400	エンコーダパルス数
3	Screw Lead	8	スクリューリード (mm)
4	Multiple	4	通倍率
5	Brake Time	0.1	ブレーキ時間
6	Position Gain	60	位置ゲイン
7	Speed Gain	80	速度ゲイン
8	F/F Gain	0	フィード/フォワードゲイン
9	Integral Gain	30	積分ゲイン
10	Total Gain	150	トータルゲイン
11	Int. Volt. Lmt	60	積分電圧リミッタ
12	Over Speed	410	オーバースピード定数
13	Error Range	2666	累積誤差
14	Motor Max Cur	90	モータ最大回転数
15	Motor Over Load	16300	モータ過負荷下限値

10. 機能別 SEL 言語命令コード一覧表

区分	条件	命令	操作1	操作2	出力	機能	ページ	多軸	単軸
数値演算	任意	LET	代入変数	代入数	ZR	代入	30	○	○
	任意	TRAN	複写先変数	複写元変数	ZR	複写	30	○	○
	任意	CLR	消去開始変数	消去終了変数	ZR	変数消去	31	○	○
算術演算	任意	ADD	被加変数	加数	ZR	加算	32	○	○
	任意	SUB	被減変数	減数	ZR	減算	32	○	○
	任意	MULT	被乗変数	乗数	ZR	乗算	33	○	○
	任意	DIV	被除変数	除数	ZR	除算	33	○	○
	任意	MOD	剰余代入変数	除数	ZR	剰余算	34	○	○
関数演算	任意	SIN	正弦代入変数	演算数 (ラジアン)	ZR	正弦	35	○	○
	任意	COS	余弦代入変数	演算数 (ラジアン)	ZR	余弦	35	○	○
	任意	TAN	正接代入変数	演算数 (ラジアン)	ZR	正接	36	○	○
	任意	ATN	逆正接代入変数	演算数 (ラジアン)	ZR	逆正接	36	○	○
	任意	SQR	平方根代入変数	演算数	ZR	平方根	37	○	○
論理演算	任意	AND	被論理積変数	演算数	ZR	論理積	38	○	○
	任意	OR	被論理和変数	演算数	ZR	論理和	39	○	○
	任意	EOR	被排他的論理和変数	演算数	ZR	排他的論理和	40	○	○
比較	任意	CPXX	比較変数	比較数	EO GE NE LT GT LE	比較	41	○	○
タイマー	任意	TIMW	待ち時間 (秒)		TU	時間待ち	42	○	○
	任意	TIMC	プログラム No.			時間待ち解除	42	○	○
	任意	GTTM	時間代入変数			時間取得	43	○	○
入出力・フラグ操作	任意	BTXX	開始出力・フラグ	終了出力・フラグ		出力・フラグ [ON OF NT]	44	○	○
	任意	WTXX	入出力・フラグ	待ち時間	TU	入出力・フラグ [ON OF] 待ち	45	○	○
	任意	IN	先頭入出力・フラグ	終了入出力・フラグ		2進数入力 (Max31bit)	46	○	○
	任意	INB	先頭入出力・フラグ	変換桁数		BCD 入力 (Max8 桁)	47	○	○
	任意	OUT	先頭入出力・フラグ	終了入出力・フラグ		2進数出力 (Max31bit)	48	○	○
	任意	OUTB	先頭入出力・フラグ	変換桁数		BCD 出力 (Max8 桁)	49	○	○
プログラム制御	任意	GOTO	ジャンプ先タグ No.			ジャンプ	50	○	○
		TAG	宣言タグ No.			ジャンプ先宣言	50	○	○
	任意	EXSR	実行サブルーチン No.			サブルーチン実行	51	○	○
		BGSR	宣言サブルーチン No.			サブルーチン開始	51	○	○
		EDSR				サブルーチン終了	52	○	○
タスク管理	任意	EXIT				プログラム終了	53	○	○
	任意	EXPG	実行プログラム No.		CP	プログラム起動	53	○	○
	任意	ABPG	停止プログラム No.		CP	他プログラム停止	54	○	○
	任意	*SLPG				タスク休止	54	○	○
	任意	*WUPG	起床プログラム No.		CP	他タスク起床	54	○	○
	任意	*GTPG	タスク状態格納変数	取得プログラム No.		タスク状態取得	55	○	○
	任意	*GTPR	タスクレベル格納変数			タスクレベル取得	55	○	○
	任意	*STPR	タスクレベル			タスクレベル変更	56	○	○
	任意	*SLIC	タスクレベル	チェック値 (秒)		タイムスライス値設定	56	○	○
資源	任意	*GTRS	獲得資源 No.			資源獲得	57	○	○
	任意	*RLRS	返却資源 No.			資源返却	57	○	○

※未公開コマンドにつき、使用不可

注) 多軸・単軸欄で表記されている○印は、多軸または単軸で使用できる命令語を表わします。

区分	条件	命令	操作1	操作2	出力	機能	ページ	多軸	単軸
ボジション操作	任意	PGET	軸 No.	ボジション No.		位置を変数 199 に代入	58	○	○
	任意	PPUT	軸 No.	ボジション No.		変数 199 の値を代入	58	○	○
	任意	PCLR	開始ボジション No.	終了ボジション No.		ポイントデータ消去	59	○	○
	任意	PCPY	複写先ボジション No.	複写元ボジション No.		ポイントデータ複写	59	○	○
	任意	PRED	読取り軸パターン	格納先ボジション No.		軸の現在位置読取り	60	○	○
	任意	PTST	確認軸パターン	確認ボジション No.	XX	ボジションデータ確認	60	○	○
	任意	PVEL	速度 (mm/sec)	代入先ボジション No.		ボジション速度代入	61	○	○
	任意	PACC	加速度 (G)	代入先ボジション No.		ボジション加速度代入	61	○	○
	任意	PAXS	軸パターン代入変数 No.	ボジション No.		軸パターン読取り	62	○	○
	任意	PSIZ	サイズ代入変数 No.			ボジションサイズ確認	62	○	○
アクチュエータ制御宣言	任意	VEL	速度 (mm/sec)			速度設定	63	○	○
	任意	OVRD	速度比 (%)			速度係数設定	63	○	○
	任意	ACC	加速度 (G)			加速度設定	64	○	○
	任意	SCRV	比率 (%)			S 字モーション比率設定	65	○	○
	任意	OFST	設定軸パターン	オフセット値 (mm)		オフセット設定	66	○	○
	任意	ATRG	位置比 (%)			アーチトリガー設定	67	○	
	任意	DEG	分割角度 (°)			分割角度設定	68	○	
	任意	BASE	基準軸 No.			基準軸設定	68	○	
	任意	GRP	有効軸パターン			グループ軸設定	69	○	
	任意	HOLD	一時停止入力ポート			一時停止ポート宣言	70	○	○
	任意	CANC	中止完了入力ポート			中止完了ポート宣言	71	○	○
	任意	AXST	ステータス格納変数	取得軸 No.		軸ステータス取得	72	○	○
	任意	SVXX	操作軸パターン			サーボ [ON OF]	73	○	○
アクチュエータ制御命令	任意	HOME	原点復帰軸パターン		PE	原点復帰	73	○	○
	任意	MOV P	移動先ボジション No.	PE		ボジション指定移動	74	○	○
	任意	MOV L	移動先ボジション No.	PE		ボジション指定補間移動	76	○	
	任意	MV P I	移動量ボジション No.	PE		ボジション相対移動	77	○	○
	任意	MV L I	移動量ボジション No.	PE		ボジション相対補間移動	79	○	
	任意	PATH	開始ボジション No.	終了ボジション No.	PE	パス移動	80	○	○
	任意	CIR	通過ボジション No.	通過ボジション No.	PE	円移動	81	○	
	任意	ARC	通過ボジション No.	終了ボジション No.	PE	円弧移動	82	○	
	任意	J X W X	動作軸パターン	起動入出力・フラグ	PE	ジョグ [FN FF BN BF]	83	○	○
	任意	STOP	停止軸パターン		PE	軸の減速停止	83	○	○
構造化 I F	任意	I F X X	比較変数	比較数		比較 [EQ NE GT GE LT LE]	84	○	○
	任意	I S X X	比較カラム	カラム or リテラル		文字列比較 [EQ NE]	85	○	○
		ELSE				IF 命令条件不成立実行先宣言	86	○	○
		EDIF				IF 終了宣言	86	○	○
構造化 D O	任意	D W X X	比較変数	比較数		ループ [EQ NE GT GE LT LE]	87	○	○
	任意	LEAV				DO からの脱出	87	○	○
	任意	ITER				DO の繰返し	88	○	○
		EDDO				DO 終了宣言	88	○	○
多分岐	任意	SLCT				多分岐開始宣言	89	○	○
		W H X X	比較変数	比較数		値分岐 [EQ NE GT GE LT LE]	90	○	○
		W S X X	比較カラム	カラム or リテラル		文字列分岐 [EQ NE]	91	○	○
		OTHE				条件不成立時分岐先宣言	92	○	○
		EDSL				SLCT 終了宣言	92	○	○

注) 多軸・単軸欄で表記されている○印は、多軸または単軸で使用できる命令語を表わします。

区分	条件	命令	操作 1	操作 2	出力	機能	ページ	多軸	単軸
外部入出力	任意	OPEN	チャンネル No.			SIO オープン	93	○	○
	任意	CLOS	チャンネル No.			SIO クローズ	93	○	○
	任意	READ	読み込みチャンネル No.	読み込みカラム No.		SIO 入力	94	○	○
	任意	WRIT	書き出しチャンネル No.	書き出しカラム No.		SIO 出力	95	○	○
	任意	SCHA	文字コード			終了文字設定	95	○	○
ストリング処理	任意	SCPY	複写先カラム	カラム or リテラル		文字列複写	96	○	○
	任意	SCMP	比較カラム	カラム or リテラル	EQ NE	文字列比較	97	○	○
	任意	SGET	転送先変数	転送元カラム		文字取得	97	○	○
	任意	SPUT	転送先カラム	書き込みデータ		文字配置	98	○	○
	任意	STR	変換先カラム	変換元変数		10 進文字列変換	98	○	○
	任意	STRH	変換先カラム	変換元変数		16 進文字列変換	99	○	○
	任意	VAL	変換先変数	変換元カラム		10 進数値変換	100	○	○
	任意	VALH	変換先変数	変換元カラム		16 進数値変換	101	○	○
	任意	SLEN	作業文字数			作業文字数設定	101	○	○

注) 多軸・単軸欄で表記されている○印は、多軸または単軸で利用できる命令語を表わします。
但し、外部入出力区分・ストリング処理区分における命令語は、DS タイプでは使用できません。

[DS タイプ専用命令語]

区分	条件	命令	操作 1	操作 2	出力	機能	参照ページ
アクチュエータ制御命令	任意	MOVD	移動位置		PE	ダイレクト指定移動	75
	任意	MVDI	移動量		PE	インクリメンタル移動	78

11. アルファベット順 SEL 言語命令コード一覧表

条件	命令	操作 1	操作 2	出力	機能	ページ	多軸	単軸
任意	ABPG	停止プログラム No.		CP	他プログラム停止	54	○	○
任意	ACC	加速度 (G)			加速度設定	64	○	○
任意	ADD	被加変数	加数	ZR	加算	32	○	○
任意	AND	被論理積変数	演算数	ZR	論理積	38	○	○
任意	ARC	通過ポジション No.	終了ポジション No.	PE	円弧移動	82	○	
任意	ATN	逆正接代入変数	演算数 (ラジアン)	ZR	逆正接	36	○	○
任意	ATRG	位置比 (%)			アーチトリガー設定	67	○	
任意	AXST	ステータス格納変数	取得軸 No.		軸ステータス取得	72	○	○
任意	BASE	基準軸 No.			基準軸設定	68	○	
	BGSR	宣言サブルーチン No.			サブルーチン開始	51	○	○
任意	BT ××	開始出力・フラグ	終了出力・フラグ		出力・フラグ [ON OF NT]	44	○	○
任意	CANC	中止完了入力ポート			中止完了ポート宣言	71	○	○
任意	CIR	通過ポジション No.	通過ポジション No.	PE	円移動	81	○	
任意	CLOS	チャンネル No.			SIO クローズ	93	○	○
任意	CLR	消去開始変数	消去終了変数	ZR	変数消去	31	○	○
任意	COS	余弦代入変数	演算数 (ラジアン)	ZR	余弦	35	○	○
任意	CP ××	比較変数	比較数	EO GE NE LT LE	比較	41	○	○
任意	DEG	分割角度 (°)			分割角度設定	68	○	
任意	DIV	被除変数	除数	ZR	除算	33	○	○
任意	DW ××	比較変数	比較数		ループ [EQ NE GT GE LT LE]	87	○	○
	EDDO				DO 終了宣言	88	○	○
	EDIF				IF 終了宣言	86	○	○
	EDSL				SLCT 終了宣言	92	○	○
	EDSR				サブルーチン終了	52	○	○
	ELSE				IF 命令条件不成立実行先宣言	86	○	○
任意	EOR	被排他的論理和変数	演算数	ZR	排他的論理和	40	○	○
任意	EXIT				プログラム終了	53	○	○
任意	EXPG	実行プログラム No.		CP	プログラム起動	53	○	○
任意	EXSR	実行サブルーチン No.			サブルーチン実行	51	○	○
任意	GOTO	ジャンプ先タグ No.			ジャンプ	50	○	○
任意	GRP	有効軸パターン			グループ軸設定	69	○	
任意	※ GTPG	タスク状態格納変数	取得プログラム No.		タスク状態取得	55	○	○
任意	※ GTPR	タスクレベル格納変数			タスクレベル取得	55	○	○
任意	※ GTRS	獲得資源 No.			資源獲得	57	○	○
任意	GTTM	時間代入変数			時間取得	43	○	○
任意	HOLD	一時停止入力ポート			一時停止ポート宣言	70	○	○
任意	HOME	原点復帰軸パターン		PE	原点復帰	73	○	○
任意	IF ××	比較変数	比較数		比較 [EQ NE GT GE LT LE]	85	○	○
任意	IN	先頭入出力・フラグ	終了入出力・フラグ		2進数入力 (Max31bit)	46	○	○
任意	INB	先頭入出力・フラグ	変換桁数		BCD 入力 (Max8 桁)	47	○	○
任意	IS ××	比較カラム	カラム or リテラル		文字列比較 [EQ NE]	85	○	○
任意	ITER				DO の繰返し	88	○	○
任意	J × W ×	動作軸パターン	起動入出力・フラグ	PE	ジョグ [FN FF BN BF]	83	○	○
任意	LEAV				DO からの脱出	87	○	○
任意	LET	代入変数	代入数	ZR	代入	30	○	○

※未公開コマンドにつき、使用不可

注) 多軸・単軸欄で表記されている○印は、多軸または単軸で使用できる命令語を表わします。

条件	命令	操作 1	操作 2	出力	機能	ページ	多軸	単軸
任意	MOD	剰余代入変数	除数	ZR	剰余算	34	○	○
任意	MOVD	移動位置		PE	ダイレクト指定移動	75		DS
任意	MOVL	移動先ポジション No.		PE	ポジション指定補間移動	76	○	
任意	MOVLP	移動先ポジション No.		PE	ポジション指定移動	74	○	○
任意	MULT	被乗変数	乗数	ZR	乗算	33	○	○
任意	MVDI	移動量		PE	インクリメンタル移動	78		DS
任意	MVLI	移動量ポジション No.		PE	ポジション相対補間移動	79	○	
任意	MVPI	移動量ポジション No.		PE	ポジション相対移動	77	○	○
任意	OFST	設定軸パターン	オフセット値 (mm)		オフセット設定	66	○	○
任意	OPEN	チャンネル No.			SIO オープン	93	○	○
任意	OR	被論理和変数	演算数	ZR	論理和	39	○	○
	OTHE				条件不成立時分岐先宣言	92	○	○
任意	OUT	先頭入出力・フラグ	終了入出力・フラグ		2進数出力 (Max31bit)	48	○	○
任意	OUTB	先頭入出力・フラグ	変換桁数		BCD 出力 (Max8 桁)	49	○	○
任意	OVRD	速度比 (%)			速度係数設定	63	○	○
任意	PACC	加速度 (G)	代入先ポジション No.		ポジション加速度代入	61	○	○
任意	PATH	開始ポジション No.	終了ポジション No.	PE	パス移動	80	○	○
任意	PAXS	軸パターン代入変数 No.	ポジション No.		軸パターン読取り	62	○	
任意	PCLR	開始ポジション No.	終了ポジション No.		ポイントデータ消去	59	○	○
任意	PCPY	複写先ポジション No.	複写元ポジション No.		ポイントデータ複写	59	○	○
任意	PGET	軸 No.	ポジション No.		位置を変数 199 に代入	58	○	○
任意	PPUT	軸 No.	ポジション No.		変数 199 の値を代入	58	○	○
任意	PRED	読取り軸パターン	格納先ポジション No.		軸の現在位置読取り	60	○	○
任意	PSIZ	サイズ代入変数 No.			ポジションサイズ確認	62	○	○
任意	PTST	確認軸パターン	確認ポジション No.	XX	ポジションデータ確認	60	○	○
任意	PVEL	速度 (mm/sec)	代入先ポジション No.		ポジション速度代入	61	○	○
任意	READ	読込みチャンネル No.	読込みカラム No.		SIO 入力	94	○	○
任意	※ RLRS	返却資源 No.			資源返却	57	○	○
任意	SCHA	文字コード			終了文字設定	95	○	○
任意	SCMP	比較カラム	カラム or リテラル		文字列比較	97	○	○
任意	SCPY	複写先カラム	カラム or リテラル	EQ NE	文字列複写	96	○	○
任意	SCRV	比率 (%)			S 字モーション比率設定	65	○	○
任意	SGET	転送先変数	転送元カラム		文字取得	97	○	○
任意	SIN	正弦代入変数	演算数 (ラジアン)	ZR	正弦	35	○	○
任意	SLCT				多分岐開始宣言	89	○	○
任意	SLEN	作業文字数			作業文字数設定	101	○	○
任意	※ SLIC	タスクレベル	チェック値 (秒)		タイムスライス値設定	56	○	○
任意	※ SLPG				タスク休止	54	○	○
任意	SPUT	転送先カラム	書き込みデータ		文字配置	98	○	○
任意	SQR	平方根代入変数	演算数	ZR	平方根	37	○	○
任意	STOP	停止軸パターン		PE	軸の減速停止	83	○	○
任意	※ STPR	タスクレベル			タスクレベル変更	56	○	○
任意	STR	変換先カラム	変換元変数		10 進文字列変換	98	○	○
任意	STRH	変換先カラム	変換元変数		16 進文字列変換	99	○	○
任意	SUB	被減変数	減数	ZR	減算	32	○	○
任意	SVXX	操作軸パターン			サーボ [ON OF]	73	○	○

※未公開コマンドにつき、使用不可

注) 多軸・単軸欄で表記されている○印は、多軸または単軸で利用できる命令語を表わします。

但し、MOVD,MVDI は、DS タイプ専用命令語です。

条件	命令	操作1	操作2	出力	機能	ヘルプ	多軸	単軸
	TAG	宣言タグ No.			ジャンプ先宣言	50	○	○
任意	TAN	正接代入変数	演算数 (ラジアン)	ZR	正接	36	○	○
任意	TIMC	プログラム No.			時間待ち解除	42	○	○
任意	TIMW	待ち時間 (秒)		TU	時間待ち	42	○	○
任意	TRAN	複写先変数	複写元変数	ZR	複写	30	○	○
任意	VAL	変換先変数	変換元カラム		10 進数値変換	100	○	○
任意	VALH	変換先変数	変換元カラム		16 進数値変換	101	○	○
任意	VEL	速度 (mm/sec)			速度設定	63	○	○
	WH××	比較変数	比較数		値分岐 [EQ NE GT GE LT LE]	90	○	○
任意	WRIT	書き出しチャンネル No.	書き出しカラム No.		SIO 出力	95	○	○
	WS××	比較カラム	カラム or リテラル		文字列分岐 [EQ NE]	91	○	○
任意	WT××	入出力・フラグ	待ち時間	TU	入出力・フラグ [ON OF] 待ち	45	○	○
任意	※ WUPG	起床プログラム No.		CP	他タスク起床	54	○	○

※未公開コマンドにつき、使用不可

注) 多軸・単軸欄で表記されている○印は、多軸または単軸で使用できる命令語を表わします。



●CLR (変数消去)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	CLR	変数No.	変数No.	ZR

〔機能〕 操作1の変数から操作2の変数をクリアします。
クリアされた変数の内容は0になります。
出力は操作1の変数に0が代入されたときオンになります。

〔例1〕 CLR 1 5 変数1～5をクリアします。

〔例2〕 LET 1 10 変数1に10を代入します。
LET 2 20 変数2に20を代入します。
CLR *1 *2 変数1の内容10から変数2の内容20までの変数をクリアします。

12.2 算術演算命令

●ADD (加算)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	ADD	変数Na	データ	Z R

〔機能〕 操作1の変数の内容と操作2の値を加算し、操作1の変数に代入します。
出力は演算結果が0になったときオンになります。

〔例1〕

LET	1	3	変数1に3を代入します。
ADD	1	2	変数1の内容3に2を足します。 変数1には3+2で5が入ります。

〔例2〕

LET	1	2	変数1に2を代入します。
LET	2	3	変数2に3を代入します。
LET	3	2	変数3に2を代入します。
ADD	*1	*3	変数1の内容2の変数に変数3の内容2を足します。 変数2には3+2で5が入ります。

●SUB (減算)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	SUB	変数Na	データ	Z R

〔機能〕 操作1の変数の内容から操作2の値を減算し、操作1の変数に代入します。
出力は演算結果が0になったときオンになります。

〔例1〕

LET	1	3	変数1に3を代入します。
SUB	1	2	変数1の内容3から2を引きます。 変数1には3-2で1が入ります。

〔例2〕

LET	1	2	変数1に2を代入します。
LET	2	3	変数2に3を代入します。
LET	3	2	変数3に2を代入します。
SUB	*1	*3	変数1の内容2の変数から変数3の内容2を引きます。 変数2には3-2で1が入ります。



●MULT (乗算)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｻｸﾞ)	命令・宣言			出力部 (出力・ﾌｻｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	MULT	変数No.	データ	ZR

〔機能〕 操作1の変数の内容に操作2の値を乗算し、操作1の変数に代入します。

出力は演算結果が0になったときオンになります。

- 〔例1〕
- | | | | |
|------|---|---|--|
| LET | 1 | 3 | 変数1に3を代入します。 |
| MULT | 1 | 2 | 変数1の内容3に2を掛けます。
変数1には 3×2 で6が入ります。 |
- 〔例2〕
- | | | | |
|------|----|----|---|
| LET | 1 | 2 | 変数1に2を代入します。 |
| LET | 2 | 3 | 変数2に3を代入します。 |
| LET | 3 | 2 | 変数3に2を代入します。 |
| MULT | *1 | *3 | 変数1の内容2の変数に変数3の内容2を掛けま
す。
変数2には 3×2 で6が入ります。 |

●DIV (除算)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｻｸﾞ)	命令・宣言			出力部 (出力・ﾌｻｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	DIV	変数No.	データ	ZR

〔機能〕 操作1の変数の内容を操作2の値で除算し、操作1の変数に代入します。

出力は演算結果が0になったときオンになります。

(注) 操作1が整数型変数の場合は少数点以下は切り捨てられます。

- 〔例1〕
- | | | | |
|-----|---|---|--|
| LET | 1 | 6 | 変数1に6を代入します。 |
| DIV | 1 | 2 | 変数1の内容6を2で割ります。
変数1には $6 \div 2$ で3が入ります。 |
- 〔例2〕
- | | | | |
|-----|----|----|---|
| LET | 1 | 2 | 変数1に2を代入します。 |
| LET | 2 | 6 | 変数2に6を代入します。 |
| LET | 3 | 2 | 変数3に2を代入します。 |
| DIV | *1 | *3 | 変数1の内容2の変数を変数3の内容2で割りま
す。
変数2には $6 \div 2$ で3が入ります。 |

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令 ・ 宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	MOD	変数No.	データ	Z R

操作2は整数のみ入力可です。

[例 2]	LET	1	2	変数 1 に 2 を代入します。
	LET	2	7	変数 2 に 7 を代入します。
	LET	3	3	変数 3 に 3 を代入します。
	MOD	* 1	* 3	変数 1 の内容 2 の変数を変数 3 の内容 3 で割った 余りを求めます。 変数 2 には $7 \div 3 = 2$ 余り 1 で、1 が代入されま す。

● S I N (正弦演算)

【機能】 操作1の変数に操作2の正弦を代入します。
出力は演算結果が0になったときオンになります。
操作1の設定範囲は実数型変数100～199、300～399を指定して下さい。
操作2の単位はラジアンです。

[例2]	LET	1	100	]	変数1に100を代入します。
	LET	101	30		$30 \times \pi \div 180$ (ラジアン)
	MULT	101	3.141592		(30° をラジアンに変換して
	DIV	101	180		変数101に代入します。)
	SIN	*1	*101		変数1の内容100の変数に変数101
					1の内容の正弦、0.5を代入します。

【機能】 操作 1 の変数に操作 2 の余弦を代入します。
出力は演算結果が 0 になったときオンになります。
操作 1 の設定範囲は実数型変数 1 0 0 ～ 1 9 9、3 0 0 ～ 3 9 9 を指定して下さい。
操作 2 の単位はラジアンです。

[例 2]	LET	1	100		変数1に100を代入します。
	LET	101	60	]	60× π ÷180 (ラジアン)
	MULT	101	3.141592		(60°をラジアンに変換して
	DIV	101	180		変数101に代入します。)
	COS	*1	*101		変数1の内容100の変数に変数101
					1の内容の余弦、0.5を代入します。

拡張条件 (AND・OR)	入力条件 (入出力・ワキ)	命令 ・ 宣言			出力部 (出力・ワキ)
		命令・宣言	操作 1	操作 2	
自由	自由	T A N	変数No.	データ	Z R

(注1) ラジアン = 角度 $\times\pi\div180$

【例2】	LET	1	100		変数1に100を代入します。
	LET	101	45		45× π ÷180 (ラジアン)
	MULT	101	3.141592	]	(45°をラジアンに変換して
	DIV	101	180		変数101に代入します。)
	TAN	*1	*101		変数1の内容100の変数に変数101
					の内容の正接、1を代入します。

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	A T N	変数No.	データ	Z R

(注1) ラジアン = 角度 $\times\pi\div180$

— 36 —

● SQR (平方根演算)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｻｸ)	命令・宣言			出力部 (出力・ﾌｻｸ)
		命令・宣言	操作1	操作2	
自由	自由	SQR	変数No.	データ	ZR

〔機能〕 操作1の変数に操作2の平方根を代入します。
出力は演算結果が0になったときオンになります。

〔例1〕 SQR 1 4 変数1に4の平方根2を代入します。

〔例2〕 LET 1 10 変数1に10を代入します。
 LET 2 4 変数2に4を代入します。
 SQR *1 *2 変数1の内容10の変数に変数2の内容4の平方根を代入します。

●AND (論理積)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	A N D	変数No.	データ	Z R

出力は演算結果が0になったときオンになります。

[例2]	LET	1	2	変数1に2を代入します。
	LET	2	2 0 4	変数2に2 0 4を代入します。
	LET	3	1 7 0	変数3に1 7 0を代入します。
	AND	* 1	* 3	変数1の内容2を変数とした内容2 0 4と変数3の内容1 7 0の論理積1 3 6を変数1の内容2の変数に代入します。

2 進数

$$\begin{array}{r} 11001100 \\ \text{AND } 10101010 \\ \hline 10001000 \end{array}$$

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令 ・ 宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	OR	変数No.	データ	Z R

出力は演算結果が0になったときオンになります。

[例 2]	LET	1	2	変数 1 に 2 を代入します。
	LET	2	2 0 4	変数 2 に 2 0 4 を代入します。
	LET	3	1 7 0	変数 3 に 1 7 0 を代入します。
	OR	* 1	* 3	変数 1 の内容 2 を変数とした内容 2 0 4 と変数 3 の内容 1 7 0 の論理和 2 3 8 を変数 1 の内容 2 の変数に代入します。

$$\begin{array}{r} 11001100 \\ \text{OR } 10101010 \\ \hline 11101110 \end{array}$$



拡張条件 (AND・OR)	入力条件 (入出力・ﾌｫﾝｸ)	命令・宣言			出力部 (出力・ﾌｫﾝｸ)
		命令・宣言	操作 1	操作 2	
自由	自由	E O R	変数No.	データ	Z R

出力は演算結果が0になったときオンになります。

[例2]	LET	1	2	変数1に2を代入します。
	LET	2	2 0 4	変数2に2 0 4を代入します。
	LET	3	1 7 0	変数3に1 7 0を代入します。
	EOR	* 1	* 3	変数1の内容2を変数とした内容2 0 4と変数3の内容1 7 0の排他的論理和1 0 2を変数1の内容2の変数に代入します。

— 40 —

12.5 比較演算命令

● CPXX (比較)

拡張条件 (AND・OR)	入力条件 (入出力・フラグ)	命令・宣言			出力部 (出力・フラグ)
		命令・宣言	操作1	操作2	
自由	自由	CPXX	変数No.	データ	EQ NE GT GE LT LE

〔機能〕 操作1の変数の内容と操作2の値を比較し条件を満たしていれば、出力をオンにします。

変数の値は変化しません。

条件を満たしていない場合は出力をオフにします。

CPXX

EQ	...	操作1 = 操作2
NE	...	操作1 ≠ 操作2
GT	...	操作1 > 操作2
GE	...	操作1 ≥ 操作2
LT	...	操作1 < 操作2
LE	...	操作1 ≤ 操作2

〔例1〕

LET	1	10		変数1に10を代入します。
CPEQ	1	10	600	変数1の内容が10ならばフラグ600をオンにします。
600 ADD	2	1		フラグ600がオンになれば変数2に1を足します。

〔例2〕

LET	1	2		変数1に2を代入します
LET	2	10		変数2に10を代入します。
LET	3	10		変数3に10を代入します。
CPNE	*1	*3	310	変数1の内容2の変数と変数3の内容が等しくなければ出力310をオンにします。

●TIMW (タイマー)

【機能】 操作1で指定した時間、プログラムの進行を停めて待ちます。
設定範囲は0.01～9.9で、単位は秒です。
出力は時間が過ぎて次のステップへ移る時にオンになります。

[例2]	LET 1 10	変数1に10を代入します。
	TIMW *1	変数1の内容の10秒間待ちます。

〔機能〕 操作1で指定した並列動作している他のプログラムのタイマーをキャンセルします。

[例 2]	LET 1 10	変数 1 に 10 を代入します。
	TIMC * 1	変数 1 の内容 10 のプログラムの 時間待ちを解除します。

[例 3] プログラム 1 ： ： ： T I M C 1 0 ：	プログラム 1 0 ： T I M W 8 2 0 (入力 8 待ち) (入力 8 待ち) ：	プログラム 1 0 は入力 8 を 2 0 秒間待ちます。 プログラム 1 0 の時間待ちを解除します。
--	---	--

- 42 -



● G T T M (時間取得)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	G T T M	変数No.		

〔機能〕 システム時間を操作 1 の変数に読みみます。時間の単位は 1 0 ミリ秒です。
ここで取得される時間は基数のない値です。よってこの命令は 2 回呼び出し、その差から経過時間を知るために使われます。

〔例 1〕

G T T M 1		変数 1 に基準となる時間を読みみます。
A D D 1	5 0 0	終了時間を 5 秒後に設定します。
G T T M 2		変数 2 に現在のシステム時間を読みみます。
D W L E 1	* 2	5 秒経ったら E D D O の次のステップへ進みます。
:		この間の処理を 5 秒間繰り返します。
:		
G T T M 2		変数 2 に現在のシステム時間を読みみます。
E D D O		

〔例 2〕

L E T 1	5	変数 1 に 5 を代入します。
G T T M * 1		変数 1 の内容 5 の変数に現在のシステム時間を格納します。

●BTXX (出力ポート・フラグ 操作)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令 ・ 宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	B T X X	出力・ﾌﾗｸﾞ	出力・ﾌﾗｸﾞ	

【機能】 操作1で指定された出力ポート・フラグから操作2で指定された出力ポート・フラグまでをオン、オフ、反転します。

BTXX			
└──┬──	ON	...	状態をオンにします。
└──┬──	OF	...	状態をオフにします。
└──┬──	NT	...	状態を反転します。

- | | | | | |
|------|------|-----|-----|------------------------------------|
| [例1] | BTON | 300 | | 出力ポート300をオンにします。 |
| [例2] | BTOF | 300 | 307 | 出力ポート300～307をオフにします。 |
| [例3] | LET | 1 | 600 | 変数1に600を代入します。 |
| | BTNT | *1 | | 変数1の内容600のフラグを反転します。 |
| [例4] | LET | 1 | 600 | 変数1に600を代入します。 |
| | LET | 2 | 607 | 変数2に607を代入します。 |
| | BTON | *1 | *2 | 変数1の内容600から変数2の内容607までのフラグをオンにします。 |



●WT×× (入出力ポート・フラグ待ち)

拡張条件 (AND・OR)	入力条件 (入出力・フラグ)	命令・宣言			出力部 (出力・フラグ)
		命令・宣言	操作1	操作2	
自由	自由	WT××	入出力・フラグ	時間	TU

〔機能〕 操作1で指定された入出力ポート・フラグがオン／オフになるのを待ちます。

操作2の時間を設定する事により、一定時間で打ち切ることが出来ます。

設定範囲は0.01～9.9秒です。

出力は一定時間が過ぎた時、オンになります。(操作2がある場合のみ)

(注) WT××命令の出力には、ローカルフラグを設定出来ません。

WT××

└─ ON	...	入出力ポート・フラグのオン待ち
└─ OF	...	入出力ポート・フラグのオフ待ち

〔例1〕 WT ON 15 入力ポート15オンを待ちます。

〔例2〕 WT OF 308 10 出力ポート308オフを10秒間待ちます。

〔例3〕 LET 1 600 変数1に600を代入します。
WT ON *1 変数1の内容600のフラグがオンになるのを待ちます。

〔例4〕 LET 1 8 変数1に8を代入します。
LET 2 5 変数2に5を代入します。
WT OF *1 *2 変数1の内容8の入力ポートがオフになるのを変数2の内容の5秒待ちます。



● I N (2進数 入出力・フラグ読み)

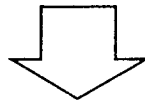
拡張条件 (AND・OR)	入力条件 (入出力・フラグ)	命令・宣言			出力部 (出力・フラグ)
		命令・宣言	操作1	操作2	
自由	自由	I N	入出力・フラグ	入出力・フラグ	

〔機能〕 操作1 から2 までの入出力ポートまたは、フラグを2進数として、変数9 9に読みます。

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	・・・	2進数
15	14	13	12	11	10	9	8	・・・	入力ポートNo
ON OFF OFF OFF OFF ON OFF ON									



1		0		0		0		0		1		0		1	...	2進数
2^7	+	0	+	0	+	0	+	0	+	2^2	+	0	+	2^0		
128	+	0	+	0	+	0	+	0	+	4	+	0	+	1	=	133



133 ・・・・・・・ 変数9 9

(注) 入力できる最大限度は31ビットです。

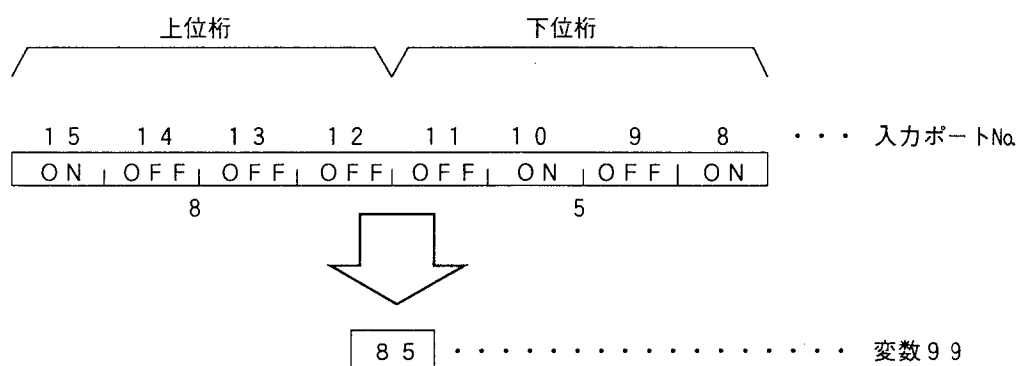
〔例1〕 I N 8 15 入力ポート8～15を2進数として、変数9 9に読みます。

〔例2〕 L E T 1 8 変数1に8を代入します。
 L E T 2 15 変数2に15を代入します。
 I N * 1 * 2 変数1の内容8～変数2の内容15のポートを2進数として、変数9 9に読みます。

● I N B (B C D 入出力・フラグ読み)

拡張条件 (AND・OR)	入力条件 (入出力・フラグ)	命令・宣言			出力部 (出力・フラグ)
		命令・宣言	操作1	操作2	
自由	自由	I N B	入出力・フラグ	B C D 桁数	

〔機能〕 操作1から操作2の桁数分の入出力ポートまたは、フラグをB C D値として、変数9 9に読みます。



(注1) 入力できる最大限度は8桁(32ビット)です。

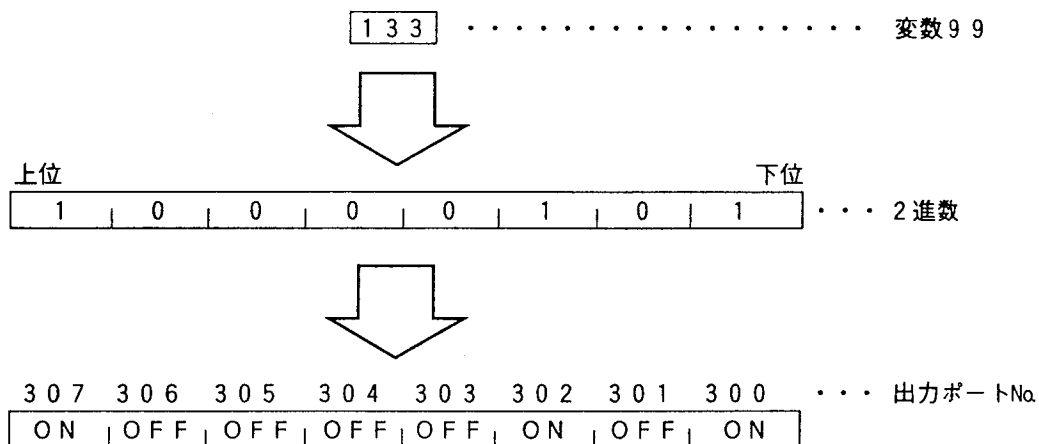
(注2) 使用する入出力ポート・フラグは4×n(桁数)です。

- [例1] I N B 8 2 入力ポート8から2桁分(15迄)をB C D値として、変数9 9に読みます。
- [例2] L E T 1 8 変数1に8を代入します。
- L E T 2 2 変数2に2を代入します。
- I N B * 1 * 2 変数1の内容8の入力ポートから変数2の内容の2桁(15迄)をB C D値として、変数9 9に読みます。

● OUT (2進数 出力)

拡張条件 (AND・OR)	入力条件 (入出力・フラグ)	命令・宣言			出力部 (出力・フラグ)
		命令・宣言	操作1	操作2	
自由	自由	OUT	入出力・フラグ	入出力・フラグ	

〔機能〕 操作1から2までの出力ポートまたは、フラグへ、変数99の値を書出します。



(注) 出力できる最大限度は31ビットです。

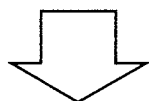
- [例1] OUT 300 307 出力ポート300～307へ、変数99の値を2進数として書出します。
- [例2] LET 1 300 変数1に300を代入します。
 LET 2 307 変数2に307を代入します。
 OUT *1 *2 変数1の内容300から変数2の内容307の出力ポートへ、変数99の値を2進数として書出します。

●OUTB (BCD 出力)

拡張条件 (AND・OR)	入力条件 (入出力・フラグ)	命令・宣言			出力部 (出力・フラグ)
		命令・宣言	操作 1	操作 2	
自由	自由	OUTB	入出力・フラグ	BCD 桁数	

〔機能〕 操作 1 から 2 までの出力ポートまたは、フラグへ、変数 9 9 の値を B C D コードで書出します。

85 変数 99



307 306 305 304 303 302 301 300 ... 出力ポートNo.

ON	OFF	OFF	OFF	OFF	ON	OFF	ON
----	-----	-----	-----	-----	----	-----	----

(注1) 出力できる最大限度は8桁(32ビット)です。

(注2) 使用する出力ポート・フラグは $4 \times n$ (桁数) です。

【例1】 OUTB 300 2 出力ポート300から2桁（307迄）に、変数
99の値をBCD値として書出します。

[例 2]	LET	1	3 0 0	変数 1 に 3 0 0 を代入します。
	LET	2	2	変数 2 に 2 を代入します。
	OUTB	* 1	* 2	変数 1 の内容 3 0 0 から変数 2 の内容 2 桁分 (3 0 7 迄) の出力ポートへ、変数 9 9 の値を B C D として書出します。



12.8 プログラム制御命令

●GOTO (ジャンプ)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	GOTO	タグNo.		

〔機能〕 操作1で指定したタグNo.の位置へジャンプします

〔注〕 GOTO命令は、同一プログラム内のみ有効です。

- 〔例1〕 TAG 1 タグをセットします。
 :
 :
 :
 GOTO 1 タグ1へジャンプします。
- 〔例2〕 LET 1 10 変数1に10を代入します。
 GOTO *1 変数1の内容10のタグへジャンプします。

●TAG (タグ宣言)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作1	操作2	
		TAG	タグNo.		

〔機能〕 操作1で指定したタグNo.を設定します。

〔例1〕 GOTO命令を参照して下さい。



● EXSR (サブルーチン実行)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	EXSR	サブルーチンNo.		

〔機能〕 操作1で指定したサブルーチンを実行します。

〔注〕 同一プログラム内のサブルーチンのみ有効です。

〔例1〕 EXSR 1 サブルーチン1を実行します。

：

EXIT

BGSR 1

サブルーチン1開始

：

：

EDSR

サブルーチン1終了

〔例2〕 LET 1 10 変数1に10を代入します。
EXSR *1 変数1の内容10のサブルーチンを実行します。

● BGSR (サブルーチン開始)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
		BGSR	サブルーチンNo.		

〔機能〕 操作1で指定したサブルーチンの開始を宣言します。

〔例1〕 EXSR命令を参照して下さい。

● E D S R (サブルーチン終了)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
		E D S R			

〔機能〕 サブルーチンの終了を宣言します。
 サブルーチンの終わりに必ず必要となります。
 この後の処理は呼び出した E X S R の次のステップに移ります。

〔例1〕 E X S R 命令を参照して下さい。

12.9 タスク管理命令

●EXIT (プログラム終了)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	EXIT			

〔機能〕 プログラムを終了します。

もしEXIT命令が無く最後のステップまで来た場合、処理は先頭に戻ります。

(注) 終了時の状態

- ・出力ポート・・・・・・・・・・保持
- ・ローカルフラグ・・・・・・・・消滅
- ・ローカル変数・・・・・・・・消滅
- ・現在値・・・・・・・・・・保持
- ・グローバルフラグ・・・・・・・・保持
- ・グローバル変数・・・・・・・・保持

〔例1〕 :
 :
EXIT プログラムを終了します。

●EXP G (他プログラム起動)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	EXP G	プログラムNo.		CP

〔機能〕 操作1で指定した他のプログラムを起動し、並列処理を行います。

出力はプログラム（タスク）が起動に成功した時オンになります。

〔例1〕 EXP G 10 プログラム10を起動します。

〔例2〕 LET 1 10 変数1に10を代入します。
EXP G *1 変数1の内容10のプログラムを起動します。

● A B P G (他プログラムの停止)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	A B P G	プログラムNo.		C P

[機能] 操作 1 で指定したプログラムを強制的に終了させます。
出力はそのプログラム (タスク) が終了した時オンになります。

(注) A B P G 命令は実行中の命令が完了した時点でプログラムを停止します。

[例 1] A B P G 1 0 プログラム 1 0 を停止させます。

[例 2] L E T 1 1 0 変数 1 に 1 0 を代入します。
 A B P G * 1 変数 1 の内容 1 0 のプログラムを停止させます。

● S L P G (タスク休止) ※未公開コマンドにつき、使用不可

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	S L P G			

[機能] 自己タスクを休止します。
再び実行状態にするには他のプログラムから W U P G 命令を実行します。

[例 1] S L P G 自己タスクを休止します。

● W U P G (他タスク起床) ※未公開コマンドにつき、使用不可

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	W U P G	プログラムNo.		C P

[機能] 操作 1 で指定した S L P G 命令で休止しているプログラムを実行状態にします。
起床に成功すると出力部をオンにします。

[例 1] W U P G 1 プログラム 1 を実行状態にします。

[例 2] L E T 1 1 0 変数 1 に 1 0 を代入します。
 W U P G * 1 変数 1 の内容 1 0 のプログラムを実行状態にします。

●GTPG（タスクの状態取得）※未公開コマンドにつき、使用不可

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	GTPG	変数No.	プログラムNo.	

〔機能〕 操作1の変数に操作2のプログラムのタスク状態を読みみます。

読みまれる値は下記の様になっています。

実行状態	・・・	TTS_RUN	01	
実行可能状態	・・・	TTS_RDY	02	
待ち状態	・・・	TTS_WAI	04	
強制待ち状態	・・・	TTS_SUS	08	
二重待ち状態	・・・	TTS_WAS	0C (12)	WAI+SUS (4+8)
休止状態	・・・	TTS_DMT	10 (16)	

〔例1〕 GTPG 10 5 変数10にプログラム5のタスク状態を読みみます。

〔例2〕 LET 1 10 変数1に10を代入します。
 LET 2 5 変数1に5を代入します。
 GTPG *1 *2 変数1の内容10の変数に変数2の内容5のプログラムのタスク状態を読みみます。

●GTPR（タスクレベル取得）※未公開コマンドにつき、使用不可

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	GTPR	変数No.		

〔機能〕 操作1の変数に自己タスクのタスクレベルを格納します。

タスクレベルの範囲は1～5です。

〔例1〕 GTPR 1 変数1にタスクレベルを格納します。

〔例2〕 LET 1 10 変数1に10を代入します。
 GTPR *1 変数1の内容10の変数にタスクレベルを格納します。



● S T P R (タスクレベル変更) ※未公開コマンドにつき、使用不可

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	S T P R	タスクレベル		

〔機能〕 操作1の値に自己タスクのタスクレベルを変更します。
タスクレベルの範囲は1～5で、小さい方が優先されます。

〔例1〕 S T P R 1 タスクレベルを1に変更します。

〔例2〕 L E T 1 3 変数1に3を代入します。
S T P R * 1 変数1の内容3にタスクレベルを変更します。

● S L I C (タイムスライス値変更) ※未公開コマンドにつき、使用不可

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	S L I C	タスクレベル	チック値	

〔機能〕 操作1のタスクレベルを操作2のチック値に変更します。
タスクレベルは1～5までの範囲で指定し、チック値は10ミリ秒単位で指定します。
チック値に0を指定した場合、タイムスライスは行われなくなります。

〔例1〕 S L I C 2 0.02 タスクレベル2のチック値を0.02秒に変更します。

〔例2〕 L E T 100 0.5 変数100に0.5を代入します。
S L I C 3 *100 タスクレベル3のチック値を変数100の内容0.5秒に変更します。

〔例3〕 L E T 1 5 変数1に5を代入します。
L E T 100 0 変数100に0を代入します。
S L I C *1 *100 変数1の内容5のタスクレベルのチック値を変数100の内容0（タイムスライスを行わない）に変更します。



12.10 資源管理命令

● G T R S (資源獲得) ※未公開コマンドにつき、使用不可

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｻｸﾞ)	命令・宣言			出力部 (出力・ﾌｻｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	G T R S	資源No.		

〔機能〕 資源の獲得をします。

資源が獲得できないときは獲得できるまで待ちます。

資源No.は 1 ～ 9 まだが指定できます。

〔例 1〕 X 軸を資源 1 と決めます (使用者が決める物なので、考えられる総ての物に使用できます)。

プログラム 1	プログラム 2	
G T R S 1	:	プログラム 1 が資源 1 を獲得します。
M O V L 1	G T R S 1	プログラム 2 は資源 1 を獲得できません。
R L R S 1	(獲得待ち)	資源 1 が返却されたのでプログラム 2 が獲得します。
:	M O V P 2	プログラム 2 が X 軸を使用できます。
:	R L E S 1	プログラム 2 が資源を返却します。

こうすることで複数のプログラムで同じ軸 (資源) を使用しても軸使用中エラーを出さない様に出来ます。

〔例 2〕

L E T 1	5	変数 1 に 5 を代入します。
G T R S * 1		変数 1 の内容 5 の資源を獲得します。

● R L R S (資源返却) ※未公開コマンドにつき、使用不可

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｻｸﾞ)	命令・宣言			出力部 (出力・ﾌｻｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	R L R S	資源No.		

〔機能〕 確保している資源を返却します。

資源No.は 1 ～ 9 まだが指定できます。

〔例 1〕 G T R S を参照してください。



12.11 ポジション操作命令

● PGET (位置データ読取り)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	PGET	軸No.	ポジションNo.	

〔機能〕 操作 2 で指定したポジションデータの操作 1 で指定した軸No.のデータを、変数 1 9 9 に読取ります。

実行時、取り込むデータが X X X. X X の場合は、変数 1 9 9 にデータを入れません (実行しない)。

〔例 1〕 PGET 2 3 ポジション 3 の 2 軸のデータを変数 1 9 9 に読取ります。

〔例 2〕 LET 1 2 変数 1 に 2 を代入します。
 LET 2 3 変数 2 に 3 を代入します。
 PGET * 1 * 2 変数 2 の内容 3 のポジションの、変数 1 の内容 2 軸のデータを変数 1 9 9 に読取ります。

● PPUT (位置データ書き込み)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	PPUT	軸No.	ポジションNo.	

〔機能〕 操作 2 で指定したポジションデータの操作 1 で指定した軸No.に、変数 1 9 9 の値を書込みます。

〔例 1〕 LET 1 9 9 1 5 0 変数 1 9 9 に 1 5 0 を代入します。
 PPUT 2 3 ポジション 3 の 2 軸に変数 1 9 9 の内容 1 5 0 を書込みます。

〔例 2〕 LET 1 9 9 1 5 0 変数 1 9 9 に 1 5 0 を代入します。
 LET 1 2 変数 1 に 2 を代入します。
 LET 2 3 変数 2 に 3 を代入します。
 PPUT * 1 * 2 変数 2 の内容 3 のポジションの、変数 1 の内容 2 軸に、変数 1 9 9 の内容 1 5 0 を書込みます。

●PCLR (ポジションデータクリア)

拡張条件 (AND・OR)	入力条件 (入出力・ワウ)	命令 ・ 宣言			出力部 (出力・ワウ)
		命令・宣言	操作 1	操作 2	
自由	自由	P C L R	ポジションNo.	ポジションNo.	

【機能】 操作1で指定したポジションNo.から操作2で指定したポジションNo.を消去します。
消去されたデータは、××. ×××になります（0. 000にはなりません）。

[例 1]	P C L R	1 0	2 0	ポジションNo.1 0～2 0のデータを消去します。
[例 2]	L E T	1	1 0	変数1に1 0を代入します。
	L E T	2	2 0	変数2に2 0を代入します。
	P C L R	* 1	* 2	変数1の内容1 0のポジションから変数2の内容2 0のポジションのデータを消去します。

●PCPY (ポジションデータコピー)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令 ・ 宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	PCPY	ポジションNo.	ポジションNo.	

[機能] 操作1で指定したポジションNo.に操作2で指定したポジションデータを複写します。

[例 1]	P C P Y 2 0	1 0	ポジションNa.2 0 へポジションNa.1 0 のデータを複写します。
[例 2]	L E T 1	2 0	変数 1 に 2 0 を代入します。
	L E T 2	1 0	変数 2 に 1 0 を代入します。
	P C P Y * 1	* 2	変数 1 の内容 2 0 のポジションに、
			変数 2 の内容 1 0 のポジションのデータを複写します。

● P R E D (座標値 読取り)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｫﾝｸｼｮﾝ)	命令・宣言			出力部 (出力・ﾌｫﾝｸｼｮﾝ)
		命令・宣言	操作1	操作2	
自由	自由	P R E D	軸パターン	ポジションNo.	

【機能】 操作1で指定した軸の現在位置を操作2で指定したポジションに読み込みます。

[例 1]	P R E D 1 1	1 0	1、2 軸の現在位置をポジションNo.1 0に読み込みます。
-------	-------------	-----	--------------------------------

[例2]	LET 1	10	変数1に10を代入します。
	PRED 11	*1	1、2軸の現在位置を変数1の内容10のポジションに読み込みます。

●P T S T (ポジションデータチェック)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｫﾝｸｼｮﾝ)	命令 ・ 宣言			出力部 (出力・ﾌｫﾝｸｼｮﾝ)
		命令・宣言	操作 1	操作 2	
自由	自由	P T S T	軸パターン	ポジションNo.	<u>X X</u>

【機能】 操作2で指定したポジションNo.の操作1で指定した軸パターンに、有効なデータが有るか確認します。
軸パターンで指定されたデータが総て無効(XX, XXX)の場合に出力がオンになります。
0は有効なデータとして扱われます。

【例1】	P T S T 1 1	1 0 3 0 0	ポジション10の1、2軸に有効な値がなければ 出力300をオンにします。 ポジションデータが下の様な場合には、出力300はオフになります。
------	-------------	-----------	---

[例 2]	LET 1	1 1	変数 1 に 1 1 を代入します。
	PTST 1 1	* 1 6 0 0	変数 1 の内容 1 1 のポジションの 1、2 軸のデータに有効な値が無ければフラグ 6 0 0 をオンにします。
			ポジションデータが下の様な場合には、フラグ 6 0 0 がオンになります。

	速度	加速度	1軸	2軸	3軸
1 0	×××	××××	1 0 0 . 0 0 0	5 0 . 0 0 0	××××. ×××
1 1	×××	××××	××××. ×××	××××. ×××	2 0 0 . 0 0 0

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令 ・ 宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	P A X S	変数No.	ポジションNo.	

	1 軸	2 軸					
9 8	××.×××	××.×××	・ ・ ・	0 0	=	0+0	= 0
9 9	1 0 0.×××	××.×××	・ ・ ・	0 1	=	0+1	= 1
1 0 0	××.×××	1 5 0.0 0 0	・ ・ ・	1 0	=	2+0	= 2
1 0 1	1 0 0.0 0 0	5 0.0 0 0	・ ・ ・	1 1	=	2+1	= 3

拡張条件 (AND・OR)	入力条件 (入出力・ワウ)	命令 ・ 宣言			出力部 (出力・ワウ)
		命令・宣言	操作 1	操作 2	
自由	自由	P S I Z	変数No.		

- 62 -



12.12 アクチュエータ制御宣言

●VEL (速度設定)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	VEL	速度		

〔機能〕 操作1の値にアクチュエータの移動速度を設定します。

単位はmm/sです。

接続されているアクチュエータの機種によって最高速度が異なりますので、それ以下を設定して下さい。

〔注1〕 小数点以下の数値は扱えません。エラーとなります。

〔注2〕 最低速度は1mm/sです。

〔例1〕	VEL	100		速度を100 mm/sに設定します。
	MOVP	1		ポイント1へ100 mm/sで移動します。
〔例2〕	VEL	500		速度を500 mm/sに設定します。
	MOVP	2		ポイント2へ500 mm/sで移動します。
〔例3〕	LET	1	300	変数1に300を代入します。
	VEL	*1		速度を変数1の内容300 mm/sに設定します。

●OVRD (オーバーライド)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	OVRD	速度比		

〔機能〕 操作1の比率に従って速度を低下させます（速度係数設定）。比率の設定範囲は1～100%です。

〔注〕 オーバーライドを掛けた結果が1以下になっても1以下の速度になる事はありません。

小数点以下の値は切り捨てます。

〔例1〕	VEL	100		速度を100 mm/sに設定します。
	OVRD	50		速度を50%に落とします。 よって、実際の速度は50 mm/sになります。
〔例2〕	LET	1	30	変数1に30を代入します。
	VEL	100		速度を100 mm/sに設定します。
	OVRD	*1		速度を変数1の内容30%に落とします。 実際の速度は30 mm/sになります。

●ACC（加速度設定）

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	ACC	加速度		

〔機能〕 操作1の値にアクチュエータの移動加速度を設定します。

単位はGです。

接続されているアクチュエータの機種や負荷によって最大加速度は異なります。

定格加速度は0.3Gです。

ACC命令で加速度を設定していない場合は、定格加速度0.3Gで動作します。

〔例1〕 ACC 0.3 加速度を0.3Gに設定します。

〔例2〕 LET 100 0.1 変数100に0.1を代入します。
 ACC *100 加速度を変数100の内容0.1Gに設定します。

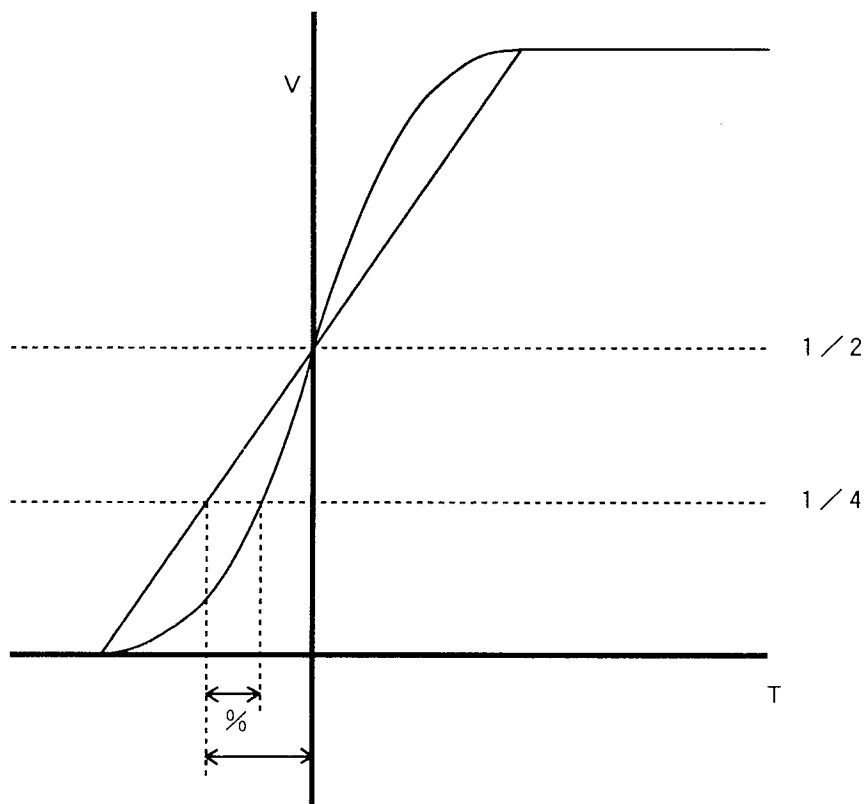
● S C R V (S字モーション比率設定)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	S C R V	比率		

〔機能〕 操作1の値にアクチュエータのS字モーション制御の比率を設定します。

設定範囲は0～50(%)の整数です。

この命令で比率を設定していない時、または0%を設定した時は台形モーションになります。



〔例1〕 S C R V 30

S字モーション比を30%にします。

〔例2〕 L E T 1 50
S C R V *1

変数1に50を代入します。
S字モーション比を変数1の内容50%にします。



●O F S T（オフセット設定）

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｸｸ)	命令・宣言			出力部 (出力・ﾌｸｸ)
		命令・宣言	操作1	操作2	
自由	自由	O F S T	軸パターン	オフセット値	

【機能】 操作1で指定されたアクチュエータ移動に於いて、目標値に操作2のオフセット値を加算し、目標値を設定し直して動作します。

オフセットの設定単位はmmで、有効分解能は0.001mmです。

オフセットは動作範囲内ならばマイナスの値も指定できます。

（注） O F S T命令は、当該プログラム以外では無効です。複数のプログラムでO F S Tを有効にするには、それぞれのプログラムでO F S T命令を実行する必要があります。

【例1】 O F S T 11001100 50 3、4、7、8軸の指定位置が50mm加算されます。

【例2】 L E T 1 100 変数1に100を代入します。
 O F S T 10101010 *1 2、4、6、8軸の指定位置が変数1の内容の100mm加算されます。



● A T R G (アーチモーションのトリガー (引き金))

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｻｸﾞ)	命令・宣言			出力部 (出力・ﾌｻｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	A T R G	位置比率		

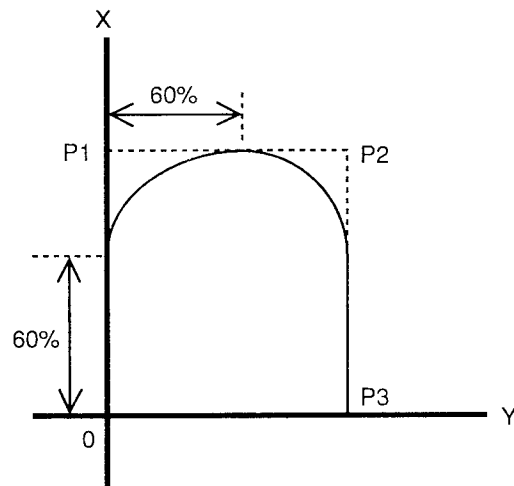
〔機能〕 A R C H命令の移動開始位置の比率を設定します。

(注) 位置比率は移動距離にもよりますが、50～60(%)以上を指定して下さい。
あまり低い比率を指定すると、軸使用中エラー(C2)が発生してしまいます。

〔例1〕

HOME	1	1	原点復帰を行います。
VEL	1	00	速度を100 mm/sに設定します。
ACC	0.	3	加速度を0.3 Gに設定します。
ATRG	6	0	位置比率を60%に設定します。
ARCH	1	3	アーチ移動を行います。

	速度	加速度	X軸	Y軸
1	XXX	XXXX	200.000	XXXX. XXX
2	XXX	XXXX	XXXX. XXX	200.000
3	XXX	XXXX	0.000	XXXX. XXX



〔例2〕

LET	1	70	変数1に70を代入します。
ATRG	*	1	変数1の内容70%に位置比率を設定します。



●DEG（円弧角度設定）

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	DEG	分割角度		

〔機能〕 操作1の値にCIR（円移動）、ARC（円弧移動）命令で参照される分割角度を設定します。

CIR、ARC命令は、ここで設定した角度毎に円を分割して通過点を計算します。

角度を小さくすると円はより正確になりますが、小さくしすぎると速度が低下します。

DEG命令で設定しない場合は、15°が設定されています。

〔例1〕 DEG 10 分割角度を10°に設定します。

〔例2〕 LET 1 10 変数1に10を代入します。
 DEG *1 分割角度を変数1の内容10°に設定します。

●BASE（軸のベース指定）

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	BASE	軸No.		

〔機能〕 操作1で指定した軸Noを1軸目として順次数えていきます。

〔例1〕 HOME 00000011 1、2軸が原点復帰します。
 BASE 3 3軸目を、1軸と考えます。
 HOME 00000011 3、4軸が原点復帰します。

以後1～6軸の指定（軸No、軸パターン、ポジションデータ等）で3～8軸が動作します。

〔例2〕 LET 1 5 変数1に5を代入します
 BASE *1 変数1の内容、5軸を1軸と考えます。

以後1～4軸の指定で5～8軸が動作します。



● G R P (グループ軸設定)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	G R P	軸パターン		

〔機能〕 操作 1 で指定された軸パターンのポジションデータだけを有効にします。

指定外の軸のデータは無いものとして扱われます。

複数のプログラムが同時に動作するとき、軸を割り当てる事により同一のポジションデータを有効に使う事が出来ます。

〔例 1〕 G R P 0 0 0 0 0 0 1 1
 C I R 1

1、2 軸のデータを有効にします。

3～8 軸にデータがあっても軸パターンエラーになりません。



●HOLD（ホールド：軸の一時停止ポート宣言）

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｹﾞ)	命令・宣言			出力部 (出力・ﾌｹﾞ)
		命令・宣言	操作1	操作2	
自由	自由	HOLD	入力ポート		

〔機能〕 移動命令実行中に、減速停止させる入力ポートの宣言をします。

指定された入力ポートがオンになると、移動中のアクチュエータは減速停止します。

その後、指定された入力ポートがオフになると、再び動作を開始します。

HOLD宣言が有効なのは同じタスク（プログラム）で動かされた軸のみです。

他のタスクで動作している軸には影響しません。

〔注1〕 PATH移動中にHOLDを行うと、次のポジション（教示点）で停止します。

他の移動命令（MOV P、MOV L、CIR、ARC等）では即時、減速停止します。

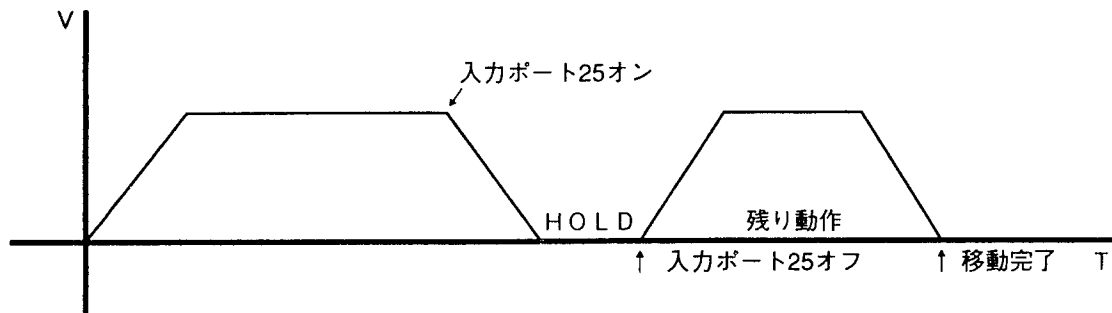
〔注2〕 1Aではユニークな原点復帰のシーケンスを採用しており、サーボロックさせてストローク端を検出しています。従って、その瞬間にHOLDを行うとHOLD解除後にアラーム（サーボの暴走）が発生する可能性があります。それを避ける為に、HOLD宣言を原点復帰後に行うか、どうしても先に宣言したい場合には原点近傍検出（エリアリミット）スイッチを設け、この付近では絶対にHOLDを行わないロジックを組んで下さい。

〔注3〕 HOLDとCANC宣言は、1つのプログラム内で同時に設定することは出来ません。

後から宣言した方が有効になります。

〔例1〕 HOLD 25

入力ポート25がオンになると減速停止します。



〔例2〕 LET 1 16

HOLD *1

変数1に16を代入します

変数1の内容16の入力ポートがオンになると減速停止します。

● C A N C (キャンセル：軸の中止完了ポート宣言)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｧｸﾄ)	命令・宣言			出力部 (出力・ﾌｧｸﾄ)
		命令・宣言	操作1	操作2	
自由	自由	C A N C	入力ポート		

〔機能〕 移動命令実行中に、減速停止させる入力ポートの宣言をします。

指定された入力ポートがオンになると、移動中のアクチュエータは減速停止します。

その後、指定された入力ポートがオフになると、その地点のまま次のステップへ進みます。

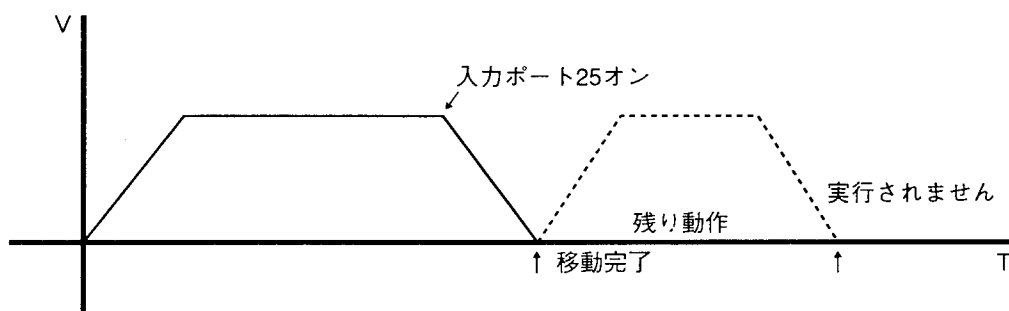
C A N C宣言が有効なのは同じタスク（プログラム）で動かされた軸のみです。

他のタスクで動作している軸には影響しません。

〔注1〕 H O L DとC A N C宣言は、1つのプログラム内で同時に設定することは出来ません。

後から宣言した方が有効になります。

〔例1〕 C A N C 2 5 入力ポート2 5 がオンになると減速停止します。



〔例2〕 L E T 1 1 6 変数1に1 6を代入します。
C A N C * 1 変数1の内容1 6の入力ポートがオンになると減速停止します。



● A X S T (軸ステータス取得)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｸｸ)	命令・宣言			出力部 (出力・ﾌｸｸ)
		命令・宣言	操作 1	操作 2	
自由	自由	A X S T	変数No.	軸No.	

【機能】 操作 1 の変数に操作 2 の軸のステータス（エラーコード）を格納します。

格納されるエラーコードは 1 6 1 (A 1) ～ 1 7 5 (A F) で、コードパネルに表示されるエラーコードと同じものです。

一覧表では 1 6 進数で書かれているので 1 0 進数に変換して考える必要があります。

最初が A ではじまるエラーコードのみ取得できます。

【例 1】 A X S T 1 2 変数 1 に 2 軸のステータスを読み込みます。この命令の後、変数 1 に 1 6 2 が入っていたとすると、
1 6 2 ÷ 1 6 = 1 0 (=A)・・・2 となりエラーコード A 2、モータ過負荷エラーが 2 軸で起こったことが解ります。

【例 2】 L E T 1 1 0 変数 1 に 1 0 を代入します。
 L E T 2 2 変数 2 に 2 を代入します。
 A X I S * 1 * 2 変数 1 の内容 1 0 の変数に変数 2 の内容 2 軸のステータスを格納します。

12.13 アクチュエータ制御命令

●SVXX (サーボオン/オフ)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	SVXX	軸パターン		

〔機能〕 操作1の軸パターンで指定された軸のサーボをオン/オフします。

SVXX
└─ ON サーボをオンします。
└─ OF サーボをオフします。

〔例1〕 SVON 11001100 3、4、7、8軸のサーボをオンにします。すでにオンになっている軸には影響しません。

●HOME (原点復帰)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	HOME	軸パターン		PE

〔機能〕 操作1の軸パターンで指定された軸を原点復帰します。

原点復帰する軸は自動的にサーボオンになります。

出力は原点復帰開始時にオフになり完了時にオンになります。

〔例1〕 HOME 11001100 3、4、7、8軸を原点復帰します。

●MOV P (PTPポジションデータ指定移動)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令 ・ 宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	MOV P	ポジションNo.		P E

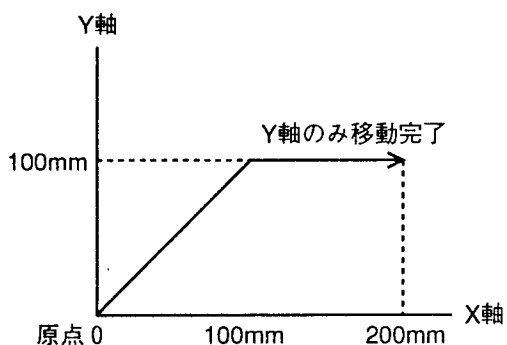
[機能] 操作1で指定されるポジションNo.の位置へ
補間なしでアクチュエータを移動します (Point To Point: ポイント トゥ ポイント)。
出力は軸移動開始時にオフになり、完了時にオンになります。

【例 1】	MOV P 1	ポジション Na.1 の位置 (2 0 0, 1 0 0) へ軸を移動します。
-------	---------	---

[例2]	LET 1 2	変数1に2を代入します。
	MOV P *1	変数1の内容2のポジションNaの位置(100, 100)へ軸を移動します。

No.	速度	加速度	X軸	Y軸
1	×××	×××	200.000	100.000
2	×××	×××	100.000	100.000

原点位置からポジションNo.1の位置(200, 100)へ移動する場合の経路



それぞれの軸が指定された速度で移動します。



●MOV D (ダイレクト指定移動)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｸｸ)	命令・宣言			出力部 (出力・ﾌｸｸ)
		命令・宣言	操作1	操作2	
自由	自由	MOV D	移動位置		P E

〔機能〕 操作1で指定された位置へアクチュエータを移動させます。
出力は軸移動開始時にオフになり、完了時にオンになります。

〔例1〕 MOV D 1 0 0 位置1 0 0へ軸を移動します。

〔例2〕 L E T 1 1 0 0 変数1に1 0 0を代入します。
 MOV D *1 変数1の内容1 0 0の位置へ軸を移動します。

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	MOV L	ポジションNo.		P E

— 76 —

●MVP I (インクリメンタルPTP移動)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	MVP I	ポジションNo.		PE

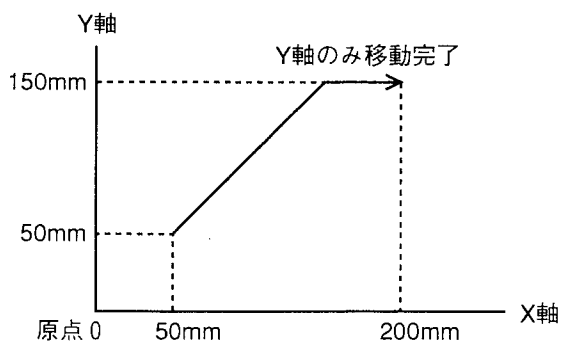
〔機能〕 操作1で指定されるポジションNo.を現在位置からの移動量として、補間なしでアクチュエータを移動させます。
出力は軸移動開始時にオフになり、完了時にオンになります。

〔例1〕 MVP I 1
現在位置が(50, 50)、ポジションNo.1が(150, 100)の場合、現在位置からX方向に150、Y方向に100の位置(200, 150)に移動します。

〔例2〕 LET 1 2
MVP I *1
変数1に2を代入します。
現在位置から変数1の内容2のポジションNo.(100, 100)を移動量として移動します。

No.	速度	加速度	X軸	Y軸
1	×××	××××	150.000	100.000
2	×××	××××	100.000	100.000

(50, 50) からポジションNo.1 (150, 100) を移動量として移動する場合の経路



それぞれの軸が指定された速度で移動します。



●MVD I (インクリメンタル移動)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	MVD I	移動量		P E

〔機能〕 操作1で指定された値を移動量として軸を移動させます。
出力は軸移動開始時にオフになり、完了時にオンになります。

〔例1〕 MVD I 3 0 現在位置から十方向へ3 0 mm移動します。

〔例2〕 L E T 1 - 1 0 0 変数1に- 1 0 0を代入します。
 MVD I * 1 現在位置から変数1の内容- 1 0 0 mm、つまり
 一方向へ1 0 0 mm移動します。

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令 ・ 宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	M V L I	ポジションNo.		P E

— 79 —

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｫﾝｸｼｮﾝ)	命令 ・ 宣言			出力部 (出力・ﾌｫﾝｸｼｮﾝ)
		命令・宣言	操作 1	操作 2	
自由	自由	P A T H	開始ポジションNo.	終了ポジションNo.	P E

(注1) PATH命令は、多次元移動させることが可能です。

[例 2]	LET	1	50	変数1に50を代入します。
	LET	2	100	変数2に100を代入します。
	PATH	*1	*2	変数1の内容50から変数2の内容100までの ポジションを連続移動します。



●C I R（円移動）

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	C I R	通過ポジションNo.	通過ポジションNo.	P E

〔機能〕 現在の位置を起点として、操作 1、2 の通過ポジションを順に通る円移動を行います。

よって操作 1 と 2 を入れ替えれば、逆回りの円移動になります。

出力は円移動開始時にオフになり、完了時にオンになります。

ただし、その後に他の C I R や P A T H、A R C 命令が連続してある場合には、最後のポイントの 2 つ手前のポイント（円周上の（分割角度 * 2）度手前）でオンになります。

（注） C I R 命令は任意の直行平面にて有効となります。

ポジションデータから自動的に判断されます。

通常は X Y 平面が第一優先になりますが、B A S E 命令によって変更することが出来ます。

〔例 1〕 C I R 1 0 0 1 0 1 現在位置からポジション 1 0 0、1 0 1 を順に通過する円移動を行います。

〔例 2〕 L E T 1 5 変数 1 に 5 を代入します。
 L E T 2 6 変数 2 に 6 を代入します。
 C I R * 1 * 2 現在位置から変数 1、2 の内容 5、6 のポジションを順に通過する円移動を行います。

●A R C (円弧移動)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令 ・ 宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	A R C	通過ポジションNo.	終了ポジションNo.	P E

〔機能〕 現在位置から操作1のポジションを通り、操作2のポジションまで円弧移動を行います。

出力は円弧移動開始時にオフになり、完了時にオンになります。

ただし、その後に他のARCやPATH、CIR命令が連続してある場合には、最後のポイントの2つ手前のポイント（円周上の（分割角度*2）度手前）でオンになります。

(注) A R C 命令は任意の直行平面にて有効となります。

ポジションデータから自動的に判断されます。

通常はXY平面が第一優先になりますが、BASE命令によって変更することが出来ます。

【例1】 A R C 1 0 0 1 0 1 現在位置からポジション100を通りポジション101までの円弧移動を行います。

【例 2】	LET	1	5	変数 1 に 5 を代入します。
	LET	2	6	変数 2 に 6 を代入します。
	ARC	* 1	* 2	現在位置から変数 1 の内容 5 のポジション通り、 変数 2 の内容 6 のポジションまでの円弧移動を行います。

● J X W X (ジョグ移動)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	J X W X	軸パターン	入出力・ﾌﾗｸﾞ	P E

〔機能〕 操作1で指定された軸を操作2で指定された入出力ポート・ﾌﾗｸﾞ(ｸﾞﾛｰﾊﾞﾙ)が条件に合っている間移動し続けます。

軸がソフトリミットまで行った場合は停止します。

J X W X

B	F	...	指定ポートがオフの間後退します。
B	N	...	指定ポートがオンの間後退します。
F	F	...	指定ポートがオフの間前進します。
F	N	...	指定ポートがオンの間前進します。

(注) 本命令には、HOLDは無効です。

〔例1〕 J B W F 1 1 0 0 1 1 0 0 1 0 入力1 0がオフの間、3、4、7、8軸を後退させます。

〔例2〕 L E T 5 2 0 変数5に2 0を代入します。
J F W N 1 0 1 0 1 0 1 0 * 5 変数5の内容2 0の入力がオンの間、2、4、6、8軸を前進させます。

● S T O P (減速停止)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	S T O P	軸パターン		P E

〔機能〕 操作1の軸パターンで指定された軸を減速停止させます。

〔例1〕 S T O P 1 1 0 0 1 1 0 0 3、4、7、8軸を減速停止します。

12.14 構造化 I F 命令

● I F X X (構造化 I F)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	I F X X	変数No.	データ	

【機能】 操作 1 の変数の内容と操作 2 の値を比較し、条件が成立した場合は次のステップに進みます。

条件が成立しない場合は、対応した E L S E 命令があればその次、なければ対応した E D I F 命令の次のステップに進みます。

入力条件が成立せず、I F X X 命令が実行されない場合は対応した E D I F の次のステップに進みます。

ネストは I S X X、D W X X と併せて 15 段まで可能です。

I F X X					
┌	EQ	...	操作 1	=	操作 2
	NE	...	操作 1	≠	操作 2
	GT	...	操作 1	>	操作 2
	GE	...	操作 1	≥	操作 2
	LT	...	操作 1	<	操作 2
	LE	...	操作 1	≤	操作 2

[例 1]	┌	600	I F E Q	1	1	軸を選択します。
		┌	I F G E	2	0	移動方向を選択します。
			J F W N	0 1	5	1 軸を前進させます。
			E L S E			
			J B W N	0 1	5	1 軸を後進させます。
			E D I F			
		┌	E L S E			
			I F L T	2	0	移動方向を選択します。
			J B W N	1 0	5	2 軸を後進させます。
			E L S E			
			J F W N	1 0	5	2 軸を前進させます。
			E D I F			
			E D I F			

変数 1 で 1、2 軸を、変数 2 で前後進（＋／－）を選択してジョグ移動します。

ﾌﾗｸﾞ 6 0 0 がオフの時は何もせず、最後の E D I F の次のステップに進みます。

● I S X X (ストリング比較)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	I S X X	カラムNo.	カラムNo.・文字リテラル	

〔機能〕 操作 1 と操作 2 のカラムの文字列を比較し、条件が成立した場合は次のステップに進みます。
条件が成立しない場合は対応した E L S E 命令があればその次、なければ対応した E D I F 命令の次のステップに進みます。
比較は S L E N 命令で設定した長さだけ行われます。
操作 2 が文字リテラルの場合はその長さ分行われます。
入力条件が成立せず、I S X X 命令が実行されない場合は E D I F の次のステップに進みます。
ネストは I F X X、D W X X と併せて 1 5 段まで可能です。

```

I S X X
├── EQ      . . .   操作 1  =  操作 2
└── NE      . . .   操作 1  ≠  操作 2

```

〔例 1〕

SCP Y	1 0	' 前進'	
SCP Y	1 4	' 後進'	
LET	1	5	
LET	2	1 4	
SLEN	4		比較数を 4 文字に設定します。
6 0 0 ISEQ	1	' 1 軸'	軸を選択します。
ISEQ	5	1 0	移動方向を選択します。
JFWN	0 1	5	1 軸を前進させます。
ELSE			
JBWN	0 1	5	1 軸を後進させます。
EDIF			
ELSE			
ISNE	* 1	* 2	移動方向を選択します。
JFWN	1 0	5	2 軸を前進させます。
ELSE			
JBWN	1 0	5	2 軸を後進させます。
EDIF			
EDIF			

カラム 1 ～ 4 で 1、2 軸をカラム 5 ～ 8 で前後進を選択してジョグ移動します。
ﾌﾗｸﾞ 6 0 0 がオフの時は何もせず、最後の E D I F の次のステップに進みます。
カラム 1 ～ 8 に次のようなデータがある場合は 1 軸を前進させます。

1	2	3	4	5	6	7	8
1	軸	前	進				



●ELSE（エルス）

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｻｸﾞ)	命令・宣言			出力部 (出力・ﾌｻｸﾞ)
		命令・宣言	操作1	操作2	
		ELSE			

〔機能〕 ELSE 命令は IFXX 命令、ISXX 命令と併せて任意に使用され、条件が成立しなかった時に実行される命令部を宣言します。

〔例1〕 IFXX、ISXX を参照してください。

●EDIF（IFXX 終了）

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｻｸﾞ)	命令・宣言			出力部 (出力・ﾌｻｸﾞ)
		命令・宣言	操作1	操作2	
		EDIF			

〔機能〕 IFXX 命令、ISXX 命令の終了を宣言します。

〔例1〕 IFXX、ISXX を参照してください。

12.15 構造化DO命令

●DWXX (DO WHILE)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	DWXX	変数No.	データ	

【機能】 操作1の変数の内容と操作2の値を比較し、条件が成立してる間EDDOまでの命令を実行します。
 条件が成立しなくなった場合は、対応したEDDO命令の次のステップに進みます。
 LEAV命令により強制的にループを終わらせる事が出来ます。
 入力条件が成立せず、DWXX命令が実行されない場合は対応したEDDOの次のステップに進みます。
 ネストはIFXX、ISXXと併せて15段まで可能です。

DWXX				
	EQ	...	操作1	= 操作2
	NE	...	操作1	≠ 操作2
	GT	...	操作1	> 操作2
	GE	...	操作1	≥ 操作2
	LT	...	操作1	< 操作2
	LE	...	操作1	≤ 操作2

【例1】 008 DW EQ 1 0 変数1が0の間EDDO命令までの
 : 命令を繰り返します。
 :
 EDDO

始めにDWXXに来た時、入力8がオフの場合は何もせずにEDDOの次のステップに進みます。

●LEAV (DO WHILEからの抜けだし)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	LEAV			

【機能】 DOXXのループを抜けてEDDOの次のステップに移行します。

【例1】 DW EQ 1 0 変数1が0の間EDDO命令までの命令を繰り返
 : します。
 600 LEAV フラグ600がオンなら強制的にループを終わら
 : せEDDO命令の次のステップに進みます。
 EDDO
 :



● I T E R (繰返し)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	I T E R			

〔機能〕 D O X X のループの途中で強制的に E D D O に制御を移します。

〔例 1〕 D W E Q 1
 :
 6 0 0 I T E R
 :
 E D D O



変数 1 が 0 の間 E D D O 命令までの命令を繰り返します。

ﾌﾗｸﾞ 6 0 0 がオンなら強制的に E D D O 命令に制御を移し、終了判定を行います。

● E D D O (D O W H I L E 終了)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
		E D D O			

〔機能〕 D W X X で始まったループの終了を宣言します。

D W X X の条件が成立しない場合は、この命令の次のステップに進みます。

〔例 1〕 D W X X を参照してください。

12.16 多分岐命令

● S L C T (選択グループの始め)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令 ・ 宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	S L C T			

【機能】 E D S L 命令までにある、条件の成立する W H × × 命令、W S × × 命令、またどの条件にも合わなければ O T H E 命令の次のステップに多分岐します。

[例 1]	SCPY 1	' 右'	カラム 1、2 に '右' を代入します。
	:		
	6 0 0 S L C T		条件が合う W × × × へ飛びます。
	WSEQ 1	' 右'	カラム 1、2 に '右' が入っている時、
	:		この部分の命令を実行します。
	WSEQ 1	' 左'	'左' が入っていた場合には、
	:		ここの命令が実行されます。
	O T H E		どちらでもなければ、
	:		ここが実行されます。
	E D S L		ﾌﾗｸﾞ 6 0 0 がオフの時、条件のうちどれかが実行された後、処理はここに移ります。



●WHXX (真の場合に選択 変数)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
		WHXX	変数No	データ	

【機能】 SLCT～EDSL 命令の間で使し、操作 1 の変数の内容と操作 2 の値を比較し条件が成立した場合に、次のWXXX、もしくはOTHE、EDSL までの命令を実行します。

WHXX	EQ	...	操作 1 = 操作 2
	NE	...	操作 1 ≠ 操作 2
	GT	...	操作 1 > 操作 2
	GE	...	操作 1 ≥ 操作 2
	LT	...	操作 1 < 操作 2
	LE	...	操作 1 ≤ 操作 2

【例 1】	LET	1	20	変数 1 に 20 を代入します。
	LET	2	10	変数 2 に 10 を代入します。
	:			
	SLCT			多分岐します。
	WHEQ	1	10	変数 1 内容が 10 なら①が実行されますが変数 1 は 20 なので次の条件を参照します。
	:			
	①			
	:			
	WHGT	1	*2	変数 1 の内容が変数 2 の内容より大きい場合に実行されます。
	:			
	②			変数 1 (=20) > 変数 2 (=10) なので、②が実行されます。
	:			
	OTHE			どの条件も成立しなかった場合に実行
	:			されます。②が実行されたので、③は実行されま
	③			は実行されません。
	:			
	EDSL			いずれかの条件が成立し、その部分の命令が実行
	:			された後は、ここに処理が移ります。この場合は
	④			②、④と実行されます。
	:			

※ 複数の条件が成立する可能性がある場合、先にあるWXXXが有効となり、後にある命令は実行されません。
条件の厳しいもの、優先順位の高いものから先に記述して下さい。

●WSXX (真の場合に選択 文字)

拡張条件 (AND・OR)	入力条件 (入出力・ワカ)	命令・宣言			出力部 (出力・ワカ)
		命令・宣言	操作1	操作2	
		WSXX	カラムNo.	カラムNo.・文字リテラル	

【機能】 SLCT～EDSLの間で使用し、操作1と操作2のカラムの文字列を比較し、条件が成立した場合に、次のWXXX、もしくはOTHE、EDSLまでの命令を実行します。

比較はSLEN命令で設定した長さだけ行われます。

操作2が文字リテラルの場合はその長さ分行われます。

WSXX
└─ EQ ... 操作1 = 操作2
└─ NE ... 操作1 ≠ 操作2

【例1】

SLEN 3		比較文字数を3にします。
SCPY 1	'ABC'	カラム1に'ABC'を代入します。
LET 1	2	変数1に2を代入します。
:		
SLCT		多分岐します。
WSEQ 1	'XYZ'	カラム1～3が'XYZ'なら①が実行されますが、カラム1～3は'ABC'なので実行されません。
:		
①		
:		
WSEQ 2	*1	カラム2からSLENで指定された文字数が変数1が示すカラムの内容と同じならば、②が実行されます。
:		
②		
:		
OTHE		どの条件も成立しなかった場合に実行されます。
:		②が実行されたので、③は実行されません。
③		
:		
EDSL		いずれかの条件が成立し、その部分の命令が実行された後は、ここに処理が移ります。この場合は②、④と実行されます。
:		
④		
:		

※ 複数の条件が成立する可能性がある場合、先にあるWSXXが有効となり、後にある命令は実行されません。
条件の厳しいもの、優先順位の高いものから先に記述して下さい。

●O T H E（その他の場合の選択）

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｻｸﾞ)	命令・宣言			出力部 (出力・ﾌｻｸﾞ)
		命令・宣言	操作1	操作2	
		O T H E			

〔機能〕 S L C T～E D S L 命令の間で使用し、どの条件も成立しなかった時に実行される命令を宣言します。

〔例1〕 S L C T、W H X X、W S X Xを参照してください。

●E D S L（選択グループの終わり）

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｻｸﾞ)	命令・宣言			出力部 (出力・ﾌｻｸﾞ)
		命令・宣言	操作1	操作2	
		E D S L			

〔機能〕 S L C T 命令の終了を宣言します。

〔例1〕 S L C T、W H X X、W S X X 命令を参照してください。

12.17 外部入出力命令

●OPEN (チャンネルオープン)

拡張条件 (AND・OR)	入力条件 (入出力・ワケ)	命令・宣言			出力部 (出力・ワケ)
		命令・宣言	操作1	操作2	
自由	自由	OPEN	チャンネルNo.		

〔機能〕 操作1で指定されたチャンネルをオープンします。
 これ以降指定されたチャンネルは送受信可能となります。
 この命令を実行する前にSCHA命令によって終了文字を設定しておく必要があります。

〔例1〕 SCHA 10 終了文字に10 (=LF) を指定します。
 OPEN 1 チャンネル1をオープンします。

〔例2〕 SCHA 13 終了文字に13 (=CR) を指定します。
 LET 1 2 変数1に2を代入します。
 OPEN *1 変数1の内容2のチャンネルをオープンします。

●CLOS (チャンネルクローズ)

拡張条件 (AND・OR)	入力条件 (入出力・ワケ)	命令・宣言			出力部 (出力・ワケ)
		命令・宣言	操作1	操作2	
自由	自由	CLOS	チャンネルNo.		

〔機能〕 操作1で指定されたチャンネルをクローズします。
 これ以降指定されたチャンネルは送受信不可能となります。

〔例1〕 CLOS 1 チャンネルをクローズします。

〔例2〕 LET 1 2 変数1に2を代入します。
 CLOS *1 変数1の内容2のチャンネルをクローズします。



● R E A D (リード)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	R E A D	チャンネルNo.	カラムNo.	

〔機能〕 操作1のチャンネルから操作2のカラムへ文字列を読み込みます。

SCHA命令で指定した文字が来ると読み込みを終了します。

カラムはローカル、グローバルどちらでもかまいません。

〔例1〕	SCHA 1 0		終了文字にLF (= 1 0)を設定します。
	OPEN 1		チャンネル1を開きます。
	READ 1	2	チャンネル1からカラム2へ文字列をLFが来るまで読み込みます。
	CLOS 1		チャンネル1を閉じます。
〔例2〕	SCHA 1 3		終了文字にCR (= 1 3)を設定します。
	OPEN 2		チャンネル2を開きます。
	LET 1	2	変数1に2を代入します。
	LET 2	3	変数2に3を代入します。
	READ * 1	* 2	変数1の内容2のチャンネルから変数2の内容3のカラムに文字列をCRが来るまで読み込みます。
	CLOS 2		チャンネル2を閉じます。



●WRIT (ライト)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	WRIT	チャンネルNo.	カラムNo.	

〔機能〕 操作1のチャンネルへ操作2のカラムから文字列を書出します。

SCHA命令で指定した文字を書出すと終了します。

カラムはローカル、グローバルどちらでもかまいません。

〔例1〕	SCHA 10			終了文字にLF (= 10) を設定します。
	OPEN 1			チャンネル1を開きます。
	WRIT 1	2		カラム2からチャンネル1へ文字列をLFまで書出します。
	CLOS 1			チャンネル1を閉じます。
〔例2〕	SCHA 13			終了文字にCR (= 13) を設定します。
	OPEN 2			チャンネル2を開きます。
	LET 1	2		変数1に2を代入します。
	LET 2	3		変数2に3を代入します。
	WRIT *1	*2		変数1の内容2のチャンネルへ変数2の内容3のカラムから文字列をCRまで書出します。
	CLOS 2			チャンネル2を閉じます。

●SCHA (終了文字設定)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	SCHA	文字コード		

〔機能〕 READ、WRIT命令で使用される終了文字の設定をします。文字は0～255 (BASICなどで使われるキャラクターコードです) までの値が指定出来ます。

〔例1〕 READ、WRIT命令を参照してください。

12.18 スtring処理命令

● S C P Y (文字列複写)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	S C P Y	カラムNo.	カラムNo.・文字リテラル	

〔機能〕 操作1のカラムへ操作2のカラムから文字列を複写します。

複写は S L E N 命令で設定した長さだけ行われます。

操作2が文字リテラルの場合はその長さ分行われます。

- 〔例1〕
- | | | |
|-------------|-----------|----------------------------|
| S C P Y 1 | ' A B C ' | カラム1へ' A B C 'を複写します。 |
| S L E N 10 | | 作業長を10バイトに設定します。 |
| S C P Y 100 | 200 | カラム100へカラム200から10バイト複写します。 |
- 〔例2〕
- | | | |
|------------|-----|--|
| L E T 1 | 300 | 変数1に300を代入します。 |
| L E T 2 | 400 | 変数2に400を代入します。 |
| S L E N 5 | | 作業長を5バイトに設定します。 |
| S C P Y *1 | *2 | 変数1の内容の300のカラムへ変数2の内容400のカラムから5バイト複写します。 |



● SCMP (文字列比較)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	SCMP	カラムNo.	カラムNo.・文字リテラル	EQ

〔機能〕 操作1のカラムと操作2のカラムを比較します。
 比較はSLEN命令で設定した長さだけ行われます。
 操作2が文字リテラルの場合はその長さ分行われます。

- 〔例1〕 SCMP 1 'ABC' 600 カラム1～3が'ABC'の時、フラグ600がオンになります。
- 〔例2〕 SLEN 5 作業長を5バイトに設定します。
 SCMP 10 30 999 カラム10とカラム30から5バイトが一致したらフラグ999をオンにします。
- 〔例3〕 LET 1 10 変数1に10を代入します。
 LET 2 20 変数2に20を代入します。
 SLEN 3 作業長を3バイトに設定します。
 SCMP *1 *2 310 変数1の内容10のカラムと変数2の内容20のカラムから3バイトが一致したら出力310をオンにします。

● SGET (文字取得)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	SGET	変数No.	カラムNo.	

〔機能〕 操作1の変数に操作2のカラムから1文字を代入します。

- 〔例1〕 SCPY 100 '1' カラム100に'1'を複写します。
 SGET 1 100 変数1にカラム100の'1'(49)を代入します。
- 〔例2〕 LET 1 3 変数1に3を代入します。
 LET 2 1 変数2に1を代入します。
 SCPY 1 'A' カラム1に'A'を複写します。
 SGET *1 *2 変数1の内容3の変数に変数2の内容1のカラムの'A'(65)を代入します。

● S P U T (文字セット)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令 ・ 宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	S P U T	カラムNo.	データ	

【機能】 操作1のカラムに操作2のデータをセットします。
コントロールコード (20) H以下のコードの時、利用する命令です。

[例1] S P U T 5 1 0 カラム5に10 (LF) をセットします。

[例 2]	LET	1	1 0 0	変数 1 に 1 0 0 を代入します。
	LET	2	5 0	変数 2 に 5 0 を代入します。
	SPUT	* 1	* 2	変数 1 の内容 1 0 0 のカラムに変数 2 の内容 5 0 (' 2 ') をセットします。

●STR (文字列変換 10進)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令 ・ 宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作 1	操作 2	
自由	自由	S T R	カラムNo.	データ	

【機能】 操作1のカラムに操作2のデータを10進数の文字列に変換したものを複写します。
SLEN命令で設定した長さにゼロサプレスを行って合わせます。
データが長さより大きい場合でもSLEN命令の設定を優先します。

(注) 操作 2 のデータが有効数字 8 桁以上 10 桁を持つ整数の場合、8 桁以上の数値の変換は保証されません。
(7 桁までの数値が正しく変換されます)

[例1]	SLEN 5.3 STR 1 カラム1～9には	整数部5桁、小数部3桁を設定します。 1 2 3 1 2 3を文字列に変換してカラム1に複写します。
------	-------------------------------	--

1	2	3	4	5	6	7	8	9
		1	2	3		0	0	0

がセットされます。

[例 2]	LET	1	1 0	変数 1 に 1 0 を代入します。
	LET	1 0 0	9 8 7, 6 5 4 3	変数 1 0 0 に 9 8 7, 6 5 4 3 を代入する。
	SLEN	2, 3		整数部 2 桁、小数部 3 桁を設定します。
	STR	* 1	* 1 0 0	変数 1 0 0 の内容を文字列に変換し変数 1 の内容 1 0 のコラムに複写します

ラム 10 ~ 15 には

10	11	12	13	14	15
8	7	.	6	5	4

がセットされます。

長さよりデータが大きかった為100の位の9と小数第4位の3が切り捨てられます。



● S T R H (文字列変換 16進)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌﾗｸﾞ)	命令・宣言			出力部 (出力・ﾌﾗｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	S T R H	カラムNo.	データ	

〔機能〕 操作1のカラムに操作2のデータを16進数の文字列に変換したものを複写します。

S L E N命令で設定した長さに整数部だけをゼロサブレスを行って合わせます。

データが長さより大きい場合でもS L E N命令の設定を優先します。

〔例1〕

S L E N 5

S T R H 1 2 5 5

整数部5桁を設定します。

2 5 5を16進数の文字列F Fに変換してカラム1に複写します。

カラム1～5には

1	2	3	4	5
			F	F

がセットされます。

〔例2〕

L E T 1 1 0

L E T 2 9 8 7 . 3 6 5 4 3

S L E N 2, 3

S T R H * 1 * 2

変数1に10を代入します。

変数2に9 8 7 . 6 5 4 3を代入する。

整数部2桁、小数部3桁を設定します。

変数2の内容9 8 7 . 3 6 5 4 3を16進数の文字列D Bに変換して変数1の内容10のカラムに複写します。

カラム10～11には

10	11
D	B

がセットされます。

小数部であるS L E N命令の、3と変数2の、6 5 4 3は無視されます。

整数部は16進で表すと'3 D B'ですが、長さは2桁なので3桁目の3は切り捨てられます。



●VAL (文字列 データ変換 10進)

拡張条件 (AND・OR)	入力条件 (入出力・ワザ)	命令・宣言			出力部 (出力・ワザ)
		命令・宣言	操作1	操作2	
自由	自由	VAL	変数No.	カラムNo.	

〔機能〕 操作1の変数に操作2のカラムの文字列を10進数としバイナリーに変換して代入します。
変換はSLEN命令で設定した長さだけ行われます。

- 〔例1〕
- | | | |
|---------|-------------|---|
| SCPY 10 | ' 1 2 3 4 ' | カラム10に' 1 2 3 4 'を複写します。 |
| SLEN 4 | | 作業長を4バイトに設定します。 |
| VAL 1 | 10 | 変数1にカラム10の' 1 2 3 4 'をバイナリーに変換した値1 2 3 4を代入します。 |
- 〔例2〕
- | | | |
|---------|-------------|---|
| LET 1 | 100 | 変数1に100を代入します。 |
| LET 2 | 20 | 変数2に20を代入します。 |
| SCPY 20 | ' 1 2 3 4 ' | カラム20に' 1 2 3 4 'を複写します。 |
| SCPY 24 | ' . 5 6 7 ' | カラム24に' . 5 6 7 'を複写します。 |
| SLEN 8 | | 作業長を8バイトに設定します。 |
| VAL *1 | *2 | 変数1の内容100の変数に変数2の内容20のカラム' 1 2 3 4 . 5 6 7 'をバイナリーに変換した値、1 2 3 4 . 5 6 7を代入します。 |



●VALH (文字列 データ変換 16進)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	VALH	変数No.	カラムNo.	

〔機能〕 操作1の変数に操作2のカラムの文字列を16進数としバイナリーに変換して代入します。

変換はSLEN命令で設定した長さだけ行われます。

整数部だけが変換され、小数部は無視されます。

〔例1〕

SCPY 10	'1234'	カラム10に'1234'を複写します。
SLEN 4		作業長を4バイトに設定します。
VALH 1	10	変数1にカラム10の'1234'を16進としてバイナリーに変換した値4660を代入します。

〔例2〕

LET 1	100	変数1に100を代入します。
LET 2	20	変数2に20を代入します。
SCPY 20	'ABCD'	カラム20に'ABCD'を複写します。
SLEN 4		作業長を4バイトに設定します。
VALH *1	*2	変数1の内容100の変数に変数2の内容20のカラム'ABCD'を16進としてバイナリーに変換した値43982を代入します。

●SLEN (レングス設定)

拡張条件 (AND・OR)	入力条件 (入出力・ﾌｳｸﾞ)	命令・宣言			出力部 (出力・ﾌｳｸﾞ)
		命令・宣言	操作1	操作2	
自由	自由	SLEN	レングス (長さ)		

〔機能〕 スtring命令の作業する長さを設定します。

下記の命令を使用する前に必ず設定する必要があります。

SCMP	...	小数部無効
SCPY	...	〃
ISXX	...	〃
STRH	...	〃
VAL, VALH	...	〃
STR	...	小数部有効

〔例1〕 上記の各命令の例を参照してください。

13. エラーコード一覧表

エラーコード	エラー名称	エラー内容
A 1	外部割り込みエラー	1. モーター過電流 2. 回生電流過大 (マイナス負荷過大) 3. ドライバのオーバーヒート
A 2	モータ過負荷エラー	機械的負荷増大等によるモータの過負荷
A 3	偏差エラー	機械的負荷増大等によりモータが指令に追従できなくなった
A 4	ソフトリミットエラー	パラメータのソフトリミット以上に動作させようとした
A 5	ボールセンスエラー	ボールセンスができない
B 0	プログラム無しエラー	プログラムデータが存在しない
B 1	プログラム実行中エラー	実行中のプログラムを再実行した
B 2	プログラムオーバーエラー	パラメータに設定されたタスク本数以上のタスクを実行した
B 3	サブルーチン No. 多重定義	サブルーチン No. が重複して使用された
B 4	タグ No. 多重定義	タグ No. が重複して使用された
B 5	サブルーチン No. 未定義	定義されていないサブルーチン No. が指定された
B 6	タグ No. 未定義	定義されていないタグ No. が指定された
B 7	サブルーチンベアエラー	BGSR と EDSR がベアになっていない
B 8	ステップ 1 が BGSR エラー	プログラムのステップ 1 が BGSR 命令になっている
B 9	D O ベアエラー	DWXX と EDDO がベアになっていない
B A	D O ネストオーバーエラー	D O の重複数が 15 を越えて設定された
B B	I F ベアエラー	IF XX と EDIF がベアになっていない
B C	E L S E エラー	ELSE が IF と EDIF の間以外の場所で使用された
C 0	原点復帰未完エラー	原点復帰を行わずに指定ポジションへ移動させようとした
C 1	位置指定エラー	位置データがないポジションへ移動させようとした
C 2	軸使用中エラー	移動中の軸に再度移動指定をした
C 3	ソフトリミットエラー	プログラムの中でソフトリミット以上に移動させる指定をした
C A	カラムエラー	カラム No. を 1～999 以外で指定した
C B	チャンネル No. エラー	デバイスを 1～2 以外で指定した
C C	ターミネーターエラー	終了文字が設定されていない
C D	資源 No. エラー	資源 No. を 1～9 以外で指定した
C E	S モーションパーセントエラー	S モーションパーセントを 0～50% 以外で指定した
C F	アーチトリガーエラー	アーチモーショントリガーを 50～100% 以外で指定した
D 0	加速度エラー	加速度を上限值以上で指定した
D 1	速度無しエラー	プログラム中で速度設定がされていない
D 2	オーバーライドエラー	オーバーライドが 1～100% 以外で指定された
D 3	角度エラー	角度が 0.1～120 度以外で指定された
D 4	軸パターンエラー	軸パターンの指定が正しくない C 1 でも D 4 が表示されます
D 5	軸 No. エラー	軸 No. が 1～8 以外で指定された
D 6	軸指定エラー	円弧で 2 軸以外の指定が行われた
D 7	プログラム No. エラー	パラメータのプログラム本数以上のプログラム No. を指定した
D 8	ポジション No. エラー	パラメータのポイントデータ数以上のポイントを指定した
D 9	ポジションデータエラー	ポイントデータが負のデータで指定された
D A	フラグ No. エラー	フラグ No. の指定が正しくない
D B	変数エラー	変数の指定が正しくない
D C	桁オーバーエラー	桁 (8 桁)・バイナリー (32 ビット) の指定がオーバーした
D D	ゼロ割り算エラー	0 で割り算された
D E	円移動初期計算エラー	円移動の出来ない位置データを指定した
D F	タスクレベルエラー	タスクレベルを 1～5 以外で指定した
E 0	未定義命令エラー	未定義の命令を実行させようとした
E 1	サブルーチンネストオーバーエラー	サブルーチンの重複数を 15 を越えて指定した
E 2	サブルーチンネストアンダーエラー	BGSR と EDSR がベアになっていない
E 3	制御欄エラー	拡張条件の使用方法が間違っている
E G	EMG エラー	エマージェンシー (非常停止) が入力された
F 0	割り込みエラー	モータ CPU との割り込み処理の数が一致しない

※ DS タイプは、エラーコードの先頭に E が付き、3 桁にて表示します。



株式会社 **アイエイアイ**

本社・工場	〒424-0103 静岡県静岡市清水区尾羽 577-1	TEL 054-364-5105 FAX 054-364-2589
東京営業所	〒105-0014 東京都港区芝 3-24-7 芝エクスージビルディング 4F	TEL 03-5419-1601 FAX 03-3455-5707
大阪営業所	〒530-0002 大阪市北区曽根崎新地 2-5-3 堂島 TSS ビル 4F	TEL 06-6457-1171 FAX 06-6457-1185
名古屋営業所	〒460-0008 名古屋市中区栄 5-28-12 名古屋若宮ビル 8F	TEL 052-269-2931 FAX 052-269-2933
盛岡営業所	〒020-0062 岩手県盛岡市長田町 6-7 クリエ 21 ビル 7F	TEL 019-623-9700 FAX 019-623-9701
仙台営業所	〒980-0802 宮城県仙台市青葉区二日町 14-15 アミ・グランデ二日町 4F	TEL 022-723-2031 FAX 022-723-2032
新潟営業所	〒940-0082 新潟県長岡市千歳 3-5-17 センザビル 2F	TEL 0258-31-8320 FAX 0258-31-8321
宇都宮営業所	〒321-0953 栃木県宇都宮市東宿郷 5-1-16 ルーセントビル 3F	TEL 028-614-3651 FAX 028-614-3653
熊谷営業所	〒360-0847 埼玉県熊谷市龍原南 1 丁目 312 番地あかりビル 5F	TEL 048-530-6555 FAX 048-530-6556
茨城営業所	〒300-1207 茨城県牛久市ひたち野東 5-3-2 ひたち野うしく池田ビル 2F	TEL 029-830-8312 FAX 029-830-8313
多摩営業所	〒190-0023 東京都立川市柴崎町 3-14-2B0SEN ビル 2F	TEL 042-522-9881 FAX 042-522-9882
厚木営業所	〒243-0014 神奈川県厚木市旭町 1-10-6 シャンロック石井ビル 3F	TEL 046-226-7131 FAX 046-226-7133
長野営業所	〒390-0852 長野県松本市島立 943 ハーモネットビル 401	TEL 0263-40-3710 FAX 0263-40-3715
甲府営業所	〒400-0031 山梨県甲府市丸の内 2-12-1 ミサトビル 3 F	TEL 055-230-2626 FAX 055-230-2636
静岡営業所	〒424-0103 静岡県静岡市清水区尾羽 577-1	TEL 054-364-6293 FAX 054-364-2589
浜松営業所	〒430-0936 静岡県浜松市中区大工町 125 大発地所ビルディング 7F	TEL 053-459-1780 FAX 053-458-1318
豊田営業所	〒446-0056 愛知県安城市三河安城町 1-9-2 第二東祥ビル 3F	TEL 0566-71-1888 FAX 0566-71-1877
金沢営業所	〒920-0024 石川県金沢市西念 3-1-32 西清ビル A 棟 2F	TEL 076-234-3116 FAX 076-234-3107
京都営業所	〒612-8401 京都市伏見区深草下川原町 22-11 市川ビル 3 F	TEL 075-646-0757 FAX 075-646-0758
兵庫営業所	〒673-0898 兵庫県明石市榑屋町 8 番 34 号大同生命明石ビル 8F	TEL 078-913-6333 FAX 078-913-6339
岡山営業所	〒700-0973 岡山市北区下中野 311-114 OMOTO-ROOT BLD. 101	TEL 086-805-2611 FAX 086-244-6767
広島営業所	〒730-0802 広島市中区本川町 2-1-9 日宝本川町ビル 5F	TEL 082-532-1750 FAX 082-532-1751
松山営業所	〒790-0905 愛媛県松山市榑味 4-9-22 フォーレスト 21 1F	TEL 089-986-8562 FAX 089-986-8563
福岡営業所	〒812-0013 福岡市博多区博多駅東 3-13-21 エフビル WING 7F	TEL 092-415-4466 FAX 092-415-4467
大分出張所	〒870-0823 大分県大分市東大道 1-11-1 タンネンバウム Ⅲ 2F	TEL 097-543-7745 FAX 097-543-7746
熊本営業所	〒862-0954 熊本県熊本市中央区神水 1-38-33 幸山ビル 1F	TEL 096-386-5210 FAX 096-386-5112

お問い合わせ先

アイエイアイお客様センター エイト

(受付時間) 月～金 24 時間 (月 7 : 00AM～金 翌朝 7 : 00AM)
土、日、祝日 8 : 00AM～5 : 00PM
(年末年始を除く)

フリー
コール **0800-888-0088**

FAX: 0800-888-0099 (通話料無料)

ホームページアドレス <http://www.iai-robot.co.jp>

IAI America Inc.

Head Office: 2690 W, 237th Street Torrance, CA 90505
TEL (310) 891-6015 FAX (310) 891-0815
Chicago Office: 110 East State Parkway, Schaumburg, IL 60173
TEL (847) 908-1400 FAX (847) 908-1399
Atlanta Office: 1220 Kennestone Circle Suite 108 Marietta, GA 30066
TEL (678) 354-9470 FAX (678) 354-9471
website : www.intelligentactuator.com

IAI Industrieroboter GmbH

Ob der Röth 4, D-65824 Schwalbach am Taunus, Germany
TEL 06196-88950 FAX 06196-889524

IAI (Shanghai) Co., Ltd.

SHANGHAI JIAHUA BUSINESS CENTER A8-303, 808, Hongqiao Rd. Shanghai 200030, China
TEL 021-6448-4753 FAX 021-6448-3992
website : www.iai-robot.com

IAI Robot (Thailand) Co., LTD.

825 PhairojKijja Tower 12th Floor, Bangna-Trad RD., Bangna, Bangkok 10260, Thailand
TEL +66-2-361-4458 FAX +66-2-361-4456

製品改良のため、記載内容の一部を予告なしに変更することがあります。
Copyright © 2014. Dec. IAI Corporation. All rights reserved.