

MITY SERVO

VEAシリーズ用

QMCL解説書

(OS:270-580対応)

株式会社 MSテクノ

2006. 05. 01

第6版 2021. 09. 28改

目 次

1. QMCL 言語説明	-----	2~3
2. QMCL 詳細		
2-1 . 変数	-----	4~5
2-2 . 入力、出力命令	-----	5
2-3 . 表示	-----	5~6
2-4 . 数値指定	-----	6
2-5 . 四則及び論理演算命令	-----	7~8
2-6 . メモリ内容への直接アクセス命令	-----	9
2-7 . モータ制御のための命令	-----	9~13
2-8 . 計時命令	-----	13
2-9 . ジャンプ命令	-----	13~15
2-10 . 戻り命令	-----	15
2-11 . 実行停止命令	-----	15
3. プログラミング上の注意		
3-1 . 演算命令の実行順序	-----	16
3-2 . 負数の表現	-----	16
3-3 . 小数点演算	-----	16
3-4 . 除算	-----	17
3-5 . 乗算	-----	17
3-6 . PLSの初期値	-----	17
3-7 . プログラム作成	-----	17~18
4. 命令実行の原理		
4-1 . プログラム構成	-----	19
4-2 . エディター及びプログラムについて	-----	19
5. キー操作		
5-1 . エディタ・キー操作	-----	20~21
6. プログラムの書き込み方法		
6-1 . プログラム例	-----	22
6-2 . 中間言語に翻訳	-----	22
6-3 . プログラム入力	-----	23
6-4 . プログラム実行	-----	23
7. ディスプレイ・キー配線図		
7-1 . 7SEG配置	-----	24
7-2 . KEYコード配置	-----	24
8. エラーコード表	-----	25

1 QMCL 言語説明

QMCLには表面言語と中間言語とがあります。コンパイラを通して使用する時は表面（高級）言語でプログラムを作成できます。コンパイラを用いることなく直接MIITYサーボに命令を与えたい、あるいはプログラムを入力したい場合は中間言語を使用すると便利です。

中間言語 コード	表面言語 ニーモニック	機 能	説 明
C0	C0. (OUT 0)	Out 信号出力	8 b i t の出力信号
C1	C1. (OUT 1)	"	"
C2	C2. (OUT 2)	"	"
C3	C3. (OUT 3)	"	"
C4	C4. (IN 0)	入力信号読み取り	8 b i t の入力信号
C5	C5. (IN 1)	"	"
C6	C6. (IN 2)	"	"
C7	C7. (IN 3)	"	"
C8	C8	送信命令	MIITY 同士の送信命令
C9	C9	受信命令	MIITY 同士の受信命令
CA	CA	10進で表示	
CB	PLS2	2ndエンコーダ カウント	2ndエンコーダ のカウント値が格納。(4Byte)
CC			
CD			
CE	\$	1バイト HEX	1バイト16進数指定。
CF	\$	2バイト HEX	2バイト16進数指定。
D0	=	イコール	行先頭の場合はNOP。
D1	+	加算記号	
D2	-	減算記号	
D3	×	乗算記号	
D4	÷	除算記号	
D5	(シフトL) × 2 ⁿ	数値の左シフト	数値 × 2 ⁿ の演算
D6	(シフトR) / 2 ⁿ	数値の右シフト	数値 ÷ 2 ⁿ の演算
D7	AND	論理積	
D8	OR	論理和	
D9	EOR	排他的論理和	
DA	NOT	反転	
DB	ABS	絶対値	
DC	PEEK	1バイトデータ リード	特定番地より1バイトデータ読み込み。
DD	POKE	1バイトデータ ストア	特定番地に1バイトデータ書き込み。
DE	DPEEK	2バイトデータ リード	特定番地より2バイトデータ読み込み。
DF	DPOKE	2バイトデータ ストア	特定番地に2バイトデータ書き込み。

中間言語 コード	表面言語 ニーモニック	機 能	説 明
E 0	HZS	現在指令周波数	現在出力している周波数が格納。
E 1	HZP	設定指令周波数	出力周波数指令命令。
E 2	PLS	1stエンコーダカウンタ	1stエンコーダのカウンタ値が格納。(4Byte)
E 3	POS	ターゲットポジション	位置決め目標値設定。
E 4	MAXHZ	位置決最大周波数	位置決め時の上限周波数を設定。
E 5	MINHZ	位置決低速PSG	位置決め低速時の減速率。
E 6	VFA	実トルク	ベクトル制御時の実トルク(READ ONLY)
		周波数比例電圧	高速時のトルクコントロール (VFモード[PWM:2]時有効)
E 7	VFB	トルクリミット	ベクトル制御時のトルクの上限值
		バイアス電圧	低速時トルクコントロール (VFモード[PWM:2]時有効)
E 8	SFT	加減速度	設定周波数への加減速度を設定。
E 9	PSG	ポジションゲイン	目標値への傾きを設定。 位置決め制御のスタート・ストップ信号も兼用
E A	TIC1	内部タイマ-1設定	2.44msecのタイムカウンタ
E B	TIC2	内部タイマ-2設定	2.44msecのタイムカウンタ
E C	HZF	速度検出	フィードバック周波数
E D	PLSI	Z相入力時パルス	Z相入力時に設定値をパルス値に置換
E E	KED	キーコード	キー入力の内容が格納。
E F	SEVCC	パワーコントロール	=0 パワーOFF、=1 パワーON
F 0	JSR	サブルーチンジャンプ	サブルーチン実行。
F 1	JMP	無条件ジャンプ	無条件にジャンプ。
F 2	JMI	条件ジャンプ	結果が<0ならジャンプ。
F 3	JEQ	条件ジャンプ	結果が=0ならジャンプ。
F 4	JPL	条件ジャンプ	結果が≥0ならジャンプ。
F 5	JNE	条件ジャンプ	結果が≠0ならジャンプ。
F 6	BRA	相対ジャンプ	設定数値分次行よりジャンプ。
F 7	CALL	マシン語サブルーチン	マシン語サブルーチン実行。
F 8	ONTIM1	サブルーチン計時命令	65msecごとにサブルーチン実行。
F 9	ONTIM2	サブルーチン計時命令	可変時間ごとのサブルーチン実行。
F A	RTS	帰還命令	サブルーチンよりメインルーチンへ帰還。
F B	OFFRTS	不帰還命令	メインルーチンへの戻りをキャンセル。
F C	AOFRTS	全不帰還命令	メインルーチンへの戻りを全てキャンセル。
F D	SCNO	シリアルチャネル	シリアル通信の子機No.を設定
F E			
F F	STOP	ストップ	プログラムストップ

2 QMCL 詳糸田

2-1. 変数

(1) ユーザ変数 (2バイトメモリA0~A9, B0~B9) 20種類

変数名 A0, A1, A2, A3, A4, A5, A6, A7, A8, A9,
B0, B1, B2, B3, B4, B5, B6, B7, B8, B9

ユーザ変数とはプログラムを作成する時に、ユーザが自由に使用できる2バイト(8ビット×2)のメモリエリアで、プログラム上、16ビット(0~65535)のデータ格納に使用します。

使用例 $A1 = A0 \times A9 + B8$

A0の内容とA9の内容を乗算し、B8の内容を加えて結果をA1に格納します。(ただし、演算途中および結果が2バイトを越えてはならない)もし越えた場合はオーバーフローした数値下位2バイト分が結果となります。

(2バイトを越えてしまう時は、下記の特別変数を使用します。)

(2) 特別変数 (AA~AF, BA~BF) ・ ・ 下記の6ペアは4バイト長専用メモリエリアです。

変数名 (AA, AB), (AC, AD), (AE, AF), (BA, BB), (BC, BD), (BE, BF)

この変数は4バイトの乗除算に用いるための変数です。(4バイト宣言不要)

BEに結果の上位2バイト、BFに結果の下位2バイトが格納されます。

但し、4バイト数値同士の演算はできません。

4バイトの演算を実行する場合、答えは必ず4バイト変数に格納して下さい。

$BE = A1 \times A0$

$BE = BE \div A1$

使用例 円周の計算例

(小数点以下の計算)

$BE = A0 \times 314$

$BE = BE / 100$

$A1 = BF$

上位	下位
AA	AB
AC	AD
AE	AF
BA	BB
BC	BD
BE	BF

(3) システム変数

MI TYサーボを動作させるためにシステムで使用している変数名に以下のものがあります。

a) HZS [中間コード E0] : 現在指令周波数

サーボ動作中、自動的に現在の周波数が格納されます。

使用例 $A0 = HZS$

とすると、変数A0に現在の周波数値が格納されます。数値は分解能1/100Hzの整数倍で表されます。(HZS)が500ならば、実際の指令周波数は5Hzです。

※この変数はユーザが自由にセットしてはいけません。(READ ONLY)

ユーザ設定は、HZP [中間コード E1] : (設定周波数)で行います。

b) PLS [中間コード: E2] : 現在パルス

エンコーダ1からの入力パルスを常にカウントした数値が格納されます。

速度制御、位置制御であれエンコーダを接続している限り、自動的にパルスをカウントし、格納します。

$AA = PLS$ とすると変数AAに現在のカウント値の上位2バイト、ABに下位2バイトが格納されます。

$PLS = 1000$ とすると、それまでのカウント値とは無関係に1000がセットされ、この後はこの値から上/下カウントされることとなります。

c) PLS2 [中間コード: CB] (2ndENC仕様の場合)

エンコーダ2からの入力パルスを常にカウントした数値が格納されます。

速度制御、位置制御であれエンコーダを接続している限り、自動的にパルスをカウントし、格納します。

d) H Z F [中間コード: E C]

エンコーダ1のパルスを読み取り、周波数の数値として演算した数値が格納されます。
数値は分解能1/100Hzの整数倍で表されます。

H Z Fが6000ならば、実際のフィードバック周波数は60Hzです。

※ H Z F 2 の中間コードはありません。 H Z F 2 のアドレス \$ E F 3 A から
読取りを行います。 例えば A 2 に読取る場合 D P E E K A 2 \$ E F 3 A となります。

e) K E D [中間コード: E E]

キー入力を行った時のキーコード内容が格納されます。(※キーコード0~31:参照7-2項)
この内容は65ms毎に更新されます。

キー入力されていない時は-1 (F F F F) となります。

3ケタの数字の打ち込み例 (オンタイム1 サブルーチンを利用した場合)

```

ONTIM1  T00
          :
T00  JMI  T04  KED          ; キー入力されているかを判定
      JPL  T10  KED-10     ; 0~9までのキーかを判定
      JEQ  T02  A1         ; 1回目か継続(桁上)かを判定
      JPL  T02  A0-1000    ; 3桁以内かを判定
      A0=A0*10+KED        ; 桁上げ
      JMP  T04
T02  A0=KED              ; キー数値をA0に格納
      A1=1                ; 次からは桁上げ
T04  CA40=A0             ; 表示
      RTS                  ; メインルーチンへ帰還
T10  :
```

2-2. 入力、出力命令

(1) 出力 C 0 [中間コード: C 0] 外部にON/OFF情報を出力します。

C 0 = 1 2 8 ; D 7 を出力します。

C 0 = C 0 O R 1 ; D 7 を維持し D 0 を出力します。

(2) 入力 C 4 [中間コード C 4] 外部の接点情報を入力する命令です。

使用例

B 0 = C 4 A N D 1 2 8

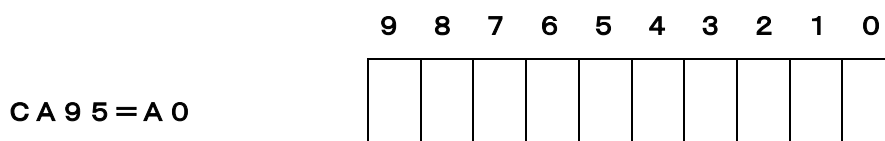
外部の信号を入力し D 7 = 1 と論理積を求めその値を B 0 に格納します。

B 0 = 1 2 8 または 0 (B 0 = 0 : O F F , B 0 ≠ 0 : O N)

2-3. 表示

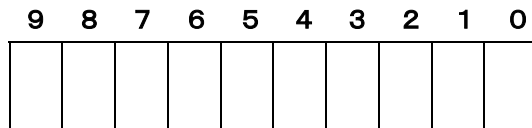
(1) 10進数表示 C A m n [中間コード C A X X]

表示部のm桁からn桁に変数名(あるいは定数)で指定した内容が表示されます。



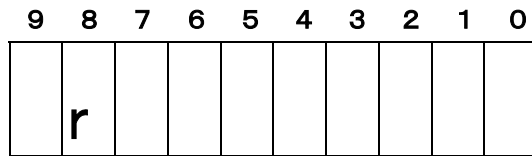
┌──────────┐
9~5にA0の内容が10進数で表示される

CA40=PLS-1000



4～0にシステム変数PLS（エンコーダカウント値）から1000を引いた数値が10進数で表示される。

CA88=22



8の桁に22 (=r) が表示される

1桁の特殊コードは以下の通りです。

10…A	16…blank	22…h
11…B	17…-	23…o
12…C	18…/	24…P
13…D	19…H	25…r
14…E	20…J	26…U
15…F	21…L	27…y

※ 正・負に変化する数値を表示するためのプログラム例を示します。HZSには正・負の値が入っている可能性があります。

プログラム例

JMI B00 HZS ; (HZS)<0ならB00へジャンプします。
CA40=HZS÷2³ ; 0.125×(HZS)を4～0桁に表示します。

RTS

B00 CA30=ABS HZS÷2³ ; 0.125×(HZS)の絶対値を求めて3～0に表示します。

CA44=17 ; 4桁に(-)を表示します。

RTS

(2) 16進数表示 CBmn [中間コード CBXX]

表示部のm桁からn桁に変数名(o r定数)で指定した内容が16進数で表示されます。

2-4. 数値指定

(1) 10進数

整数の10進数指定は通常の数を用いる時と同様に行うことができます。

A0=100 A0に100を格納します。

A1=-3 A1に-3を格納します。(内容は\$FFFFD)

(2) 16進数

プログラムの中で16進数を使用したい時にはその数の先頭に\$を付けることにより指定することができます。

\$77 [中間コード: CE77]

\$123 [中間コード: CF0123]

\$を付けるとコンパイラにより、表面言語を中間言語に変換する時に自動的に、CE(1バイト)、CF(2バイト)のいずれかが判断されて数値変換されます。

\$77=119(10進)

\$123=291(10進)

の意味となります。

2-5. 四則及び論理演算命令

(1) 等号 (=) [中間コードD0]

この記号はまた特殊な意味 [NOP: No Operation] をもっており命令文の先頭にD0 (=) があるとその行の命令は実行されません。

(2) 加算 (+) [中間コード: D1]

使用例 $A0 = B0 + B1$ (A0D0B0D1B1) …中間言語
変数B0にB1を加算し、A0に格納されます。

(3) 減算 (-) [中間コード: D2]

使用例 $B0 = B0 - 1$ (B0D0B0D201) …中間言語
変数B0の内容から1を減算した値が新たにB0に格納されます。

(4) 乗算 (×) [中間コード: D3]

使用例 $A1 = A2 \times 10$ (A1D0A2D310) …中間言語
変数A2の内容に10を乗算した値が変数A1に格納されます。

(5) 除算 (÷) [中間コード: D4]

使用例 $B2 = B0 \div 5$ (B2D0B0D405) …中間言語
B0の内容を5で除算した値がB2に格納されます。

(6) 左シフト (×2^n) [中間コード: D5]

左へのnビットシフト命令です。これはある値を2^n倍することと同じです。

使用例 $A0 = B1 \times 2^1 (= B1 \times 2) \rightarrow (A0D0B1D501)$
B1の内容を左へ1ビットシフトしその値をA0に格納します。

$A0 = B2 \times 2^3 (= B2 \times 8) \rightarrow (A0D0B2D508)$

B2の内容を左へ3ビットシフトしその値をA0に格納します。

(7) 右シフト (÷2^n) [中間コード: D6]

右へのnビットシフト命令です。これはある数を2^nで除算することと同じです。 使用例

$A0 = B1 \div 2^3$ (A0D0B1D603)

($A0 = B1 \div 8$) (A0D0B1D408)

B1の内容を右へ3ビットシフトしその内容がA0に格納されます。この時B1の内容は変化しません。

(8) 論理積 (AND) [中間コード: D7]

2つの入力(変数あるいは定数)の論理的な積を求めます。

真理値表は下表の通りです。

2入力の対応するビットがともに1の時、結果が1となります。

例えば $\$55 \text{ AND } \$33 \rightarrow \$11$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

0 1 0 1 | 0 1 0 1 = \$55

AND

0 0 1 1 | 0 0 1 1 = \$33

↓

0 0 0 1 | 0 0 0 1 = \$11

(メモリの内容を示しています。)

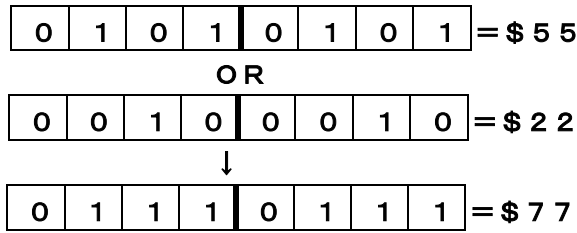
※QMCLでは $C0 = C0 \text{ AND } 128$ のような命令によりD7ビットのみONしたまま他をOFFとして出力を行うこともできます。

(9) 論理和 (OR) [中間コード: D8]

2つの入力(変数あるいは定数)の論理的な和を求めます。

真理値表は右表の通りです。2入力の対応するビットのいずれかが1の時結果は1となります。

例えば \$55 OR \$22 → \$77



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

※QMCLでは C0=C0 OR 64命令によりすでに出力しているビットはそのまま

64 (

0	1	0	0	0	0	0	0
---	---	---	---	---	---	---	---

) との論理和を求めることにより

ビット6をONした出力を出すことができます。

(上は C0=C0 OR \$40とも同一です)

(10) 排他的論理和 (EOR) [中間コード: D9]

2つの入力(変数あるいは定数)の排他的な論理和を求めます。

その真理値表は下表の通りです。言い替えると2入力の対応するビットが同じ時は0、違っている時は1とします。

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

(11) 論理反転 (NOT) [中間コード: DA]

入力の論理を反転(NotあるいはInvertともいう)します。

使用例 A1=NOT A0 (A1D0DAA0)

A0の内容を反転してA1に格納します。 A0=1のとき A1=\$FFFEとなります。

A0=0のとき A1=\$FFFFとなります。

(12) 絶対値 (ABS) [中間コード: DB]

与えられた変数の絶対値を求めます。

X=-3のとき Y=ABS X=3です。

使用例 A1=ABS A0 (A1D0DBA0)

A0の絶対値を求めA1に格納します。 A0=1のとき A1=1となります。

A0=\$FFFFのときA1=1となります。

A0=\$FFFEのときA1=2となります。

2-6. メモリ内容への直接アクセス（読み出し書き込み）命令

(1) 内容（データ）の読み出し

a) 1バイト読み出し PEEK [中間コード: DC]

b) 2バイト読み出し DPEEK [中間コード: DE]

命令で指定された番地の内容を読み出し指定された変数に格納します。

使用例 DPEEK A0 \$FE50 (DEA0CFFE50)

\$FE50と\$FE51番地の内容を読み出してA0に格納します。

PEEK B0 \$FF0D (DCB0CFFF0D)

\$FF0D番地の内容を読み出してB0に格納します。

DPEEK A0 A2 (DEA0A2)

A2の内容（数値）を番地とし、その番地と次の番地の内容を読み出してきてA0に格納します。

(2) データの書き込み

a) 1バイトデータの書き込み POKE [中間コード: DD]

b) 2バイトデータの書き込み DPOKE [中間コード: DF]

命令で指定した番地に変数の内容を書き込みます。

使用例 DPOKE \$F200 A0 (DFCFF200A0)

A0の内容を\$F200番地と\$F201番地に書き込みます。

POKE \$F300 B0 (DDCFF300B0)

B0の下位バイトの内容を\$F300番地に書き込みます。

DPOKE A2 A0 (DFA2A0)

A0の内容をA2の内容番地と、その次の番地に書き込みます。

2-7. モータ制御のための命令

(1) 周波数比例電圧設定 (VFA) [中間コード: E6]

周波数に比例して出力したい電圧の傾きを設定します。

下記（図2-1）に参考図を示しています。

[V/Fモード(オフソループ)時のみ有効]

(2) トルクリミット値 or バイアス電圧設定 (VFB)

[中間コード: E7]

(2-1) トルクリミット値・ベクトル制御時

[システムパラメータ No.16 PWM Mode = 0 or 3 or 8]

各周波数における最大トルクを1000として、トルク制御が可能です。パラメータにより出力トルクは異なります。

(2-2) バイアス電圧・・・VFモード時

[システムパラメータ No.16 PWM Mode 2]

ゼロ周波数で出力したい電圧を設定します。

(図2-2)に参考図を示します。

(※設定値については、各OSの

パラメータ表の設定範囲を御参照下さい。)

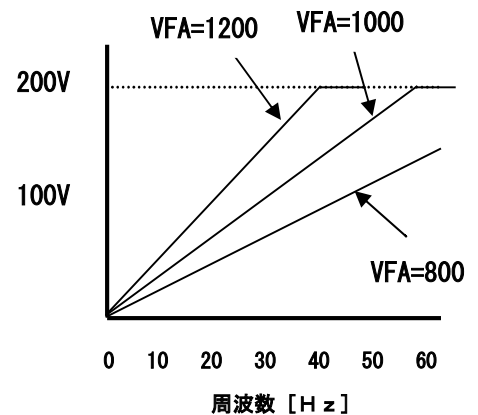


図2-1 VFAと周波数

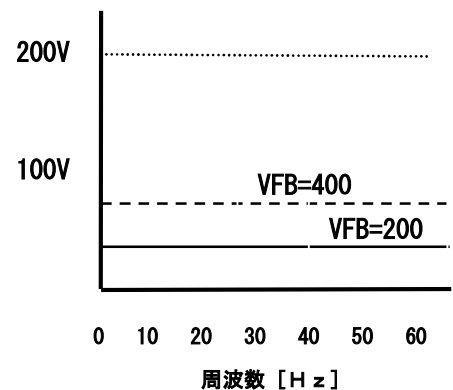


図2-2 VFBと周波数

(3) 加速度設定 (SFT) [中間コード: E8]

モータを始動、停止させる時の加減速度の設定を行います。一般的なソフトのスタートを意味しています。

$$\text{加減速度の設定} \quad SFT = \frac{Hz \times 20}{t(\text{sec})}$$

SFT=6000の時約0.2秒で60Hz

SFT=1200の時約1秒で60Hz

SFT=120の時約10秒で60Hz

に上昇することとなります。

SFT	周波数変化	加減速時間
6000	60Hz	0.2sec
1200	60Hz	1sec
120	60Hz	10sec

(4) 指令周波数設定 (HZP) [中間コード: E1]

モータへの指令周波数の設定を行います。

実際の周波数Hz (motor) は次式で与えられます。

$$Hz(\text{motor}) = (HZP) \times (1/100) \quad [Hz]$$

(HZP) > 0ならば正転, (HZP) < 0ならば逆転となります。

(HZP) = 0ならば停止します。

HZP=6000の場合・・・指令周波数正転60Hzです。

※このコマンドは、速度制御のスタート信号も兼ねており、このコマンドに数値が与えられると、SFTに従って周波数を出力します。

(5) 目標位置設定 (POS) [中間コード: E3]

位置決め制御を行いたい時の目標値を設定する命令です。エンコーダカウント値(システム変数: PLS)がPOSの内容と同一になるまでモータは回転し一致すると停止します。

(6) 最大周波数設定 (MAXHZ) [中間コード: E4]

位置決め制御においてはHZPを設定せずに最大周波数のみを設定して制御します。

例えば MAXHZ=3000としておくと最大30Hzの周波数を用いて位置決めが行われます。下記(図2-3)に参考図を示します。

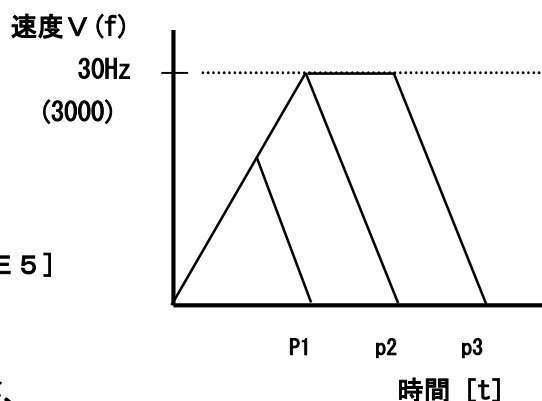


図2-3 MAXHzとPOS

(7) 最小周波数設定 (MINHZ) [中間コード: E5]

位置決め制御において低速制御時の最小周波数を設定できます。

この数値が大きい程位置決め時間は速くなりますが、位置決め精度は悪くなってしまいますのでご注意ください。

(8) パワーコントロール命令 (SEVCC) [中間コード: EF]

モータ通電ON/OFFのための命令です。

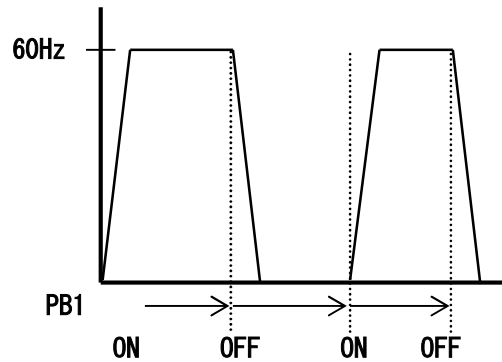
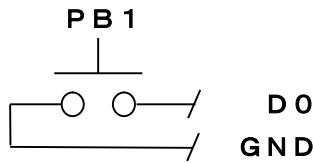
SEVCC=0 (パワーをOFFにします。) : モータはフリーラン状態となります。

SEVCC=1 (パワーをONにします。) : モータに通電されます。

[例 1] 入力を使ったモータ運転(ベクトル制御の場合)

入力D0がONされている間、周波数を60Hzに設定し、モータを回転させます。

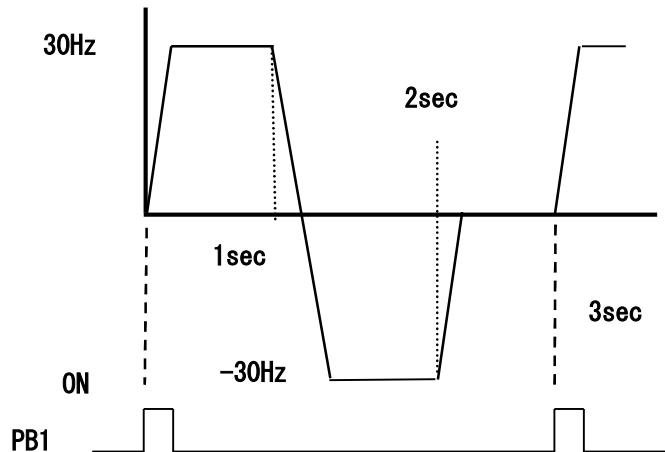
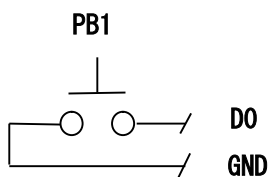
QMCLソースプログラム	行NO.	QMCL中間言語
CALL \$460	000	F7CF0460FF ; パラメータモード設定
VFB=1000	001	E7D01000FF ; トルクリミット設定
SFT=2000	002	E8D02000FF ; 加減速度設定
SEVCC=1	003	EFD001FF ; モータ通電ON
L00 JNE L01 C4 AND 1	004	F507C4D701FF ; 入力D0 (ON?)
HZP=0	005	E1D000FF ; モータ (停止)
JMP L00	006	F104FF ; 入力読み込みへ
L01 HZP=6000	007	E1D06000FF ; モータ (60Hzで回転)
JMP L00	008	F104FF ; 入力読み込みへ



PB1をON/OFFするたびに上図のような動作をくり返すためのプログラムです。

[例 2] タイマ正逆運転

QMCLソースプログラム	行NO.	QMCL中間言語
CALL \$460	000	F7CF0460FF ; パラメータモード設定
VFB=1000	001	E7D01000FF ; トルクリミット設定
SFT=6000	002	E8D06000FF ; 加減速度設定
SEVCC=1	003	EFD001FF ; モータ通電ON
L00 JEQ L00 C4 AND 1	004	F304C4D701FF ; スタート待ち
L01 HZP=3000	005	E1D00960FF ; モータ30Hz正転
TIC1=410	006	EAD00410FF ; タイマ1秒にセット
L02 JNE L02 TIC1	007	F507EAFF ; タイマアップ待ち
HZP=-3000	008	E1D0D20960FF ; モータ30Hz逆転
TIC1=410	009	EAD00410FF ; タイマ1秒にセット
L03 JNE L03 TIC1	010	F510EAFF ; タイマアップ待ち
JMP L00	011	F104FF ; プログラム繰り返し



PB1を1度ONすると1秒間正転し、その後1秒間逆転して再度PB1がONか否かを見に行きます。

(9) 位置決めゲイン設定 (PSG) [中間コード: E9]

最大周波数を使用した位置決めの時、減速曲線(直線)を指定することができます。駆動されるメカニズム(機械)の慣性モーメントあるいは摩擦力に応じた適切な値を設定することができます。(図2-4)にPSGの値と減速直線の概略を示します。メカニズムの動きにオーバーシュートが発生するような場合にはPSGを小さくしてお使い下さい。

位置決め制御を行う時には、PSGにある値(ゼロ以外の)を設定します。

また、PSG設定が位置決め制御のスタート命令も兼ねています。

このPSGとSFTには次の関係があります。

設定値 = $SQR(SFT \times \text{エンコーダ補正值})$

例えば

SFT = 6000

エンコーダ補正值 = 800 のとき

(モータ: 4pole、エンコーダ: 2500パルス)

$$PSG = \sqrt{6000 \times 800} = 2190$$

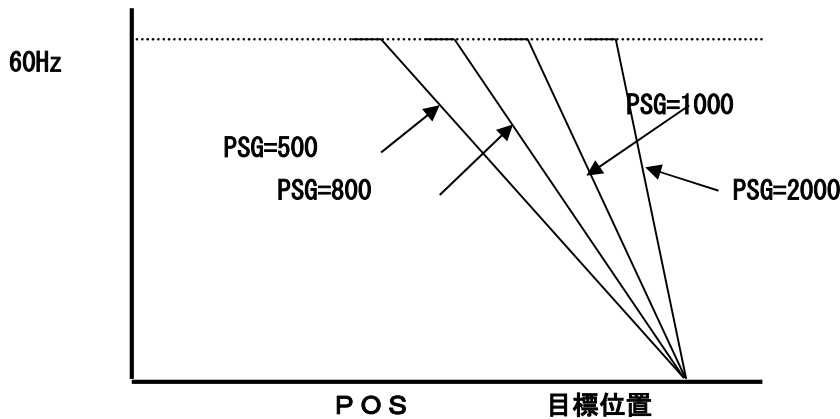


図2-4 POSと周波数

POS = PLS となると PSG = 0 (位置決め完了) となり自動的にモータは停止し、[HZ P=0] (停止しているが負荷の反力には耐え停止続ける) を維持します。サーボロック状態で保持する場合は、位置決め完了後、もう一度、PSGをループさせます。(次頁にて使用例を示します。)

使用例

```
      POS = 2000
      PSG = 1000
      :
L00   JNE L00 PSG
```

PSGを1000と与えて、PSGが0になるのを待ち0になれば位置決め完了と判断し、QMCLではPSG=0として、L00の次の命令の実行に入ります。PSGが大きい程、減速所要時間は短くなります。

サーボロック状態の使用例（PSGをループさせる）

```

L 0 1   P S G = 5
          J N E   L 0 1   C 4   A N D   1   ; C4-D0 ONの間 サーボロック ※
          P S G = 0
          H Z P = 0
          :
          ; ※2018.06.21 訂正 (JNE ← JEQ)

```

2-8. 計時命令

QMCLにはモータあるいはメカニズムの動きを制御する時に必要となる計時のカウントのためのタイマ（時間待ちを行う）命令が2ヶ用意されています。

TIC2がTIC1より優先します。

(1) 待ち時間（タイマ）設定（TIC1, TIC2） [中間コード：EA, EB]

TIC1はタイマ1用、TIC2はタイマ2用の時間設定命令です。

TICn = xxx (n = 1, 2)

(xxx = 1, 2...65535)

xxxの1は2.44msecに対応しています。

TIC1 = 100はタイマ1の待ち時間を0.244秒とすることを意味しています。

使用例

```

          T I C 2 = 1 0 0
L 0 0   J N E   L 0 0   T I C 2
          H Z P = 2 4 0
          :

```

プログラムの上の部分を実行してきたQMCLはここでタイマ2を0.244秒と設定します。この時モータの動きは前の状態を継続しています。L00は0.244秒の待ち時間の実行です。0.244秒経つとQMCLは次の命令HZP=240（30HZ周波数設定）を実行し、以下のプログラムの実行に移って行きます。

2-9. ジャンプ命令

プログラムの分岐命令をして以下のものが用意されています。

(1) サブルーチンジャンプ（JSR） [中間コード：F0]

メインプログラムからラベルあるいは行番号で指定されたサブルーチンへジャンプする場合用います。

使用例 JSR 130 (F0130)

QMCLコマンド行の130行目から書かれているサブルーチンへジャンプします。

(2) 無条件ジャンプ（JMP） [中間コード：F1]

無条件に指定されたラベルあるいは行番号へジャンプする場合に用います。

使用例 JMP 50 (F150)

(3) 条件ジャンプ

条件ジャンプとして次の四つのものが用意されています

ニーモニック	中間コード	意味	説明
JMI	F2	Jump Minus	条件文が負<0の時ジャンプ
JEQ	F3	Jump Equal ZERO	条件文が零=0の時ジャンプ
JPL	F4	Jump Plus	条件文が正≥0の時ジャンプ
JNE	F5	Jump Not Equal ZERO	条件文が≠0の時ジャンプ

使用法 JEQ LOO [条件文]

条件文としては 変数 (ユーザ変数およびシステム変数) あるいは直接演算式を書くことができます。

使用例: JEQ 50 A0-A1 (F350A0D2A1)

ユーザ変数A0とA1の差が0となった時50行目へジャンプします。0でない時は、すぐ下の命令が実行されます。

(4) 相対ジャンプ命令 (BRA) [中間コード: F6]

無条件ジャンプの一種ですが、指定された変数の内容に従ってジャンプ先を変えることができます。ユーザ変数0の場合、次行を実行します。変数の内容が飛び行数になります。

使用法 : BRA [ユーザ変数]

使用例		(中間言語)	
BRA	B0	(F6B0)	
JMP	LOO	(F1nn)	※ (nn、mm、ee) には、ジャンプ先の
JMP	LO1	(F1mm)	行番号が入ります。
JMP	LO2	(F2ee)	

ユーザ変数B0に入っている数値 (0、1あるいは2) に従い、
0の場合LOO : 1の場合LO1 : 2の場合LO2 にジャンプします。

(5) 実時間時計利用命令 (略称: ONTIM命令)

QMCLは65msec単位と(0.244msec×n)単位の計時を行っている時計を各1ヶ持っています。

(n: ONTIM2制御時間(\$F01C)設定)

65msec毎あるいは(0.244msec×n)毎に行いたい処理を実行させるために利用することができます。

時計1利用宣言 ONTIM1 (ラベル名あるいは行番号) [中間コード: F8]

時計2利用宣言 ONTIM2 (ラベル名あるいは行番号) [中間コード: F9]

使用法

```
                  ONTIM1 100      (F80100)
LOO  A0=A0+A1
```

65msec毎に行番号100より始まるサブルーチンへジャンプし、その処理が終了するとこのルーチンへ戻り直ちにLOO以下の命令を実行していきます。

```
                  ONTIM2 50      (F950)
LO1  B0=A0×B1
```

(0.244msec×n)毎に実行中の行数を戻り行として記憶し、行番号50より始まるサブルーチンへジャンプし、その処理が終了すると記憶した戻り行の命令を実行していきます。また、ONTIM1より優先して実行します。

ONTIM命令を複数回続けて使用した場合には最後のものが有効となります。

時計の利用終了宣言は、OFTIM命令により行われます。

使用例 OFTIM1 [中間コード; F800] : 時計1不使用

(6) 特殊ジャンプ (CALL) [中間コード: F7]

マシン語で書かれたサブルーチンへジャンプする時に利用します。

QMCLには実行時間の高速化を目的としたマシン語のサブルーチンをいくつか有しています。

また、マシン語で作成したサブルーチンを実行する事もできます。

これらを使用したい時に利用する命令です。

使用例 CALL \$460 (F7CF0460)

システムパラメータモードをコールします。

2-10. 戻り命令

(1) 戻り命令 (RTS) [中間コード: FA]

主ルーチンからジャンプしてきた当該サブルーチンの終了宣言と、もとの主ルーチンの戻りを指令する命令です。 ONTIM命令、JSR命令で実行するサブルーチンの終了行に必要です。

(2) 戻り禁止命令 (OFFRTS) [中間コード: FB]

サブルーチンでの処理の結果によっては先の主ルーチンへ戻る必要がない場合が生じます。このような時のサブルーチン・プログラムの終了宣言命令として利用すると便利です。

(3) 戻り先テーブルクリア命令 (AOFRTS) [中間コード: FC]

プログラムの実行によって順次自動的に作成されているサブルーチンからの戻り先を記憶したテーブルを全てクリア(消去)する命令です。

何らかの理由でプログラムが正常に動作していないことが検出された時などに本命令を利用し、正常な動きに戻すことができます。

2-11. 実行停止命令 (STOP) [中間コード: FF]

モータを停止させパワー部への電源を遮断し(モータはフリーラン) QMCLをエディタモードへセットする命令です。

※QMCLでクリアするとはメモリの内容を`F`という文字にするということと同義です。1行の命令がFFF...となっていることは何もプログラムが書かれていないことを意味しています。これはつまり”STOP”命令ということにもなります。

3-4. 除算

QMCLでの除算は正の数同士の演算しか許されていません。負の数の演算を行う場合には先ず与えられた数の絶対値を求めた後、演算を行って下さい。

使用例 $A0 = (-9) / 3$

プログラム例

```
A1=-9
A2=ABS A1 /3
JPL L00 A2
A0=-A2
L00      :
```

3-5. 乗算

QMCLでは2バイトの整数(4バイト特殊変数は除く)しか取り扱っていませんので2つの乗算結果が32767を越える時は負の数となってしまいますのでご注意ください。(3-3. 負数の表現 参照)

3-6. PLSの初期値

システム変数PLSにはエンコーダからのパルス数のカウント値が常に保存されています。原点位置設定としてPLS=0とすると原点近傍でその内容は1と65535(=-1)を交互にとったりして数値のつながりと動きの実感がつかみにくなります。このようなことを避けたい場合、初期値としてたとえばPLS=1000としておくとう便利です。原点へ戻すことは1000への設定となり999、1000、1001といった値をとることとなり、そのつながりが容易につかめます。ただし表示する場合は、1000を減算することをお勧めします。

プログラム例

```
PLS=1000
POS=2000
PSG=10
L00 CA40=PLS-1000
      (1000を減算してPLSをディスプレイ4~0に表示)
JNE L00 PSG
      :
```

3-7. プログラム作成

(1) 10進数の内部表現

10進数は内部では2文字単位(1byte)で表現されています。

使用例 HZP=960の内部表現は(E1 D0 09 60)のようになります。

直接キーより入力する時は として下さい。

(2) 1行の命令は16文字以内

QMCLのコマンド行は16文字(8byte)以内となっています。長い命令文を作成したい場合には2つに分けて作成して下さい。

使用例 $A0=BA+1000+A2+A1$ この命令文は

(A0 D0 BA D1 10 00 D1 A2 D1 A1)の20文字となっています。このような場合は次のように命令を分けて記述して下さい。

```
A0=BA+1000
A0=A0+A2+A1
```

(3) 条件ジャンプ文の演算式の最初は文字でJNE、JMI等の条件ジャンプ文の条件部分に演算式を用いたい場合、その先頭には必ず文字変数を使用して下さい。

```
使用例 JMI L20 10-A0 . . . 誤
        JMI L20 A0-10 . . . 正
```

(4) ジャンプ文の飛び先指定

ジャンプ文の飛び先に演算式を用いることはできません。必ず指定の行番号あるいはラベルをご使用下さい。

```
JMP 20+A0 . . . 誤
JMP 100 . . . . . 正
JMP L01 . . . . . 正
```

(5) NOPコマンド

1つのプログラムのある行を無視したい場合にNOP [ノーオペレーションの意味、中間コード D0 (=と同一)] を使用します。

今ある行の命令として $A0=A1+2$

(中間言語表現: A0 D0 A1 D1 02)があったとします。

上記命令文の先頭にD0を追加します。

(中間言語表現: D0 A0 D0 A1 D1 02)

D0を追加したことにより、上記命令文は無視されます。D0を削除すると再度実行されます。デバッグの途中などでNOPを使うと効率良くバグつぶしを進めることができます。

(6) ONTIM1, 2の時間制約

ONTIM1, 2により起動され、実行されるサブルーチンは、それぞれ65msec、(0.244msec×n)以内に処理が終了されなければなりません。

QMCLのコマンド行1行は約0.1msecで処理されるとし、サブルーチンの総処理時間を算出し、その値がそれぞれの知的時間内に入っていることをご確認下さい。

※TIC1, 2命令をONTIM1, 2の中で使用することは禁止されています。

(7) プログラム入力

QMCLの中間言語で直接MITYサーボにコマンドを入力しプログラムを作成する場合、1行のコマンド入力に引続き“FF”を挿入して下さい。

```
使用例 HZP=32 → E1 D0 32 FF . . . . . FF
        A4=A0*2 → A4 D0 A0 D3 02 FF . .
```

※直接プログラム入力する際は行の終わりの桁には、“FF”を書き込んでください。

4 命令実行の原理

4-1. プログラム構成

M I T Yサーボのプログラムは行単位で構成されます。

QMCLソースプログラム	行No.	QMCL中間言語
CALL \$460	000	F7CF0460FF
VFB=1000	001	E7D01000FF
SFT=2000	002	E8D02000FF
SEVCC=1	003	EFD001FF
L00 JNE L01 C4 AND 1	004	F507C4D701FF
HZP=0	005	E1D000FF
JMP L00	006	F104FF
L01 HZP=1920	007	E1D01920FF
JMP L00	008	F104FF

L00、L01は、ジャンプ先の目安のためのラベルで、コマンドではありません。QMCL中間言語でFFとは、行先頭にあるときはSTOPコマンドであり、それ以外の時は行エンドです。以降説明ではFFは省きます。

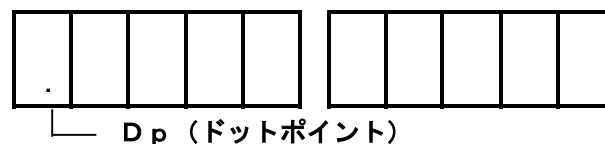
命令語の表し方は、英数字の組み合わせ、1コマンド2文字で構成され、1行最大8コマンド16文字で構成されることになります。したがって、1命令は1行に納める必要があります。また、同一行内に2命令のプログラムはできません。

プログラムの実行順序は、行No.の若い順から実行されます。

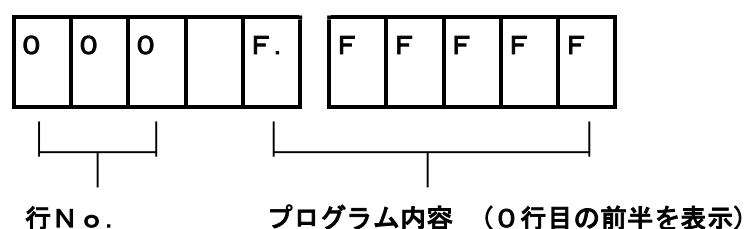
4-2. エディタ及びプログラムについて


エディタとは修正編集という意味で、つまりプログラムの修正編集に必要なものです。フラッシュメモリのプログラムを選択する様に短絡ピンがセットされていない時に電源を投入するとエディタとなり、ディスプレイの左端にDp（ドットポイント）が点灯してコマンド待ちとなります。

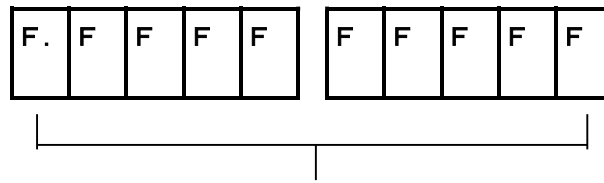
エディタに於いてプログラムを書き込める状態になった事をプログラムモードといい、作成したプログラムをM I T Yサーボに入力する事ができます。エディタの時の表示は下記ようになります。また、プログラムストップ時のエディタはディスプレイ左にストップした行数を表示します。



プログラムモード (F CR) になった時の表示を下記に示します。



ここで  キーを6回押すと表示が下記の様になり、その行の後半のプログラムを表示します。



プログラム内容 (0行目の後半を表示)

プログラムモードではDp (ドットポイント) の表示している桁が入力中の桁でカーソルとなります。以降カーソルと呼びます。

5 キー操作

5-1. キー操作




① , 


キー操作のキャンセルができ、CLRは1キャラクタのキャンセル、L. CLRは全てのキャンセルとなります。






② , 




プログラムエリアのクリアができます。

(下記のX、Yは行数及び数値を入力します。)

   でX行よりプログラムがクリアされます。

   で0行よりX行までプログラムがクリアされます。

     でX行よりY行までクリアとなります。


③ , , 

X行よりプログラムを実行します。

④ , , 




X行よりプログラムモードになります。

⑤ , 

マシン語の操作が出来るモードとなります。このモードからエディタへ戻るときは  キーを押して下さい。

⑥ , , 

メモリーの内容を表示します。

   とすれば、A0の変数の内容を10進数で表示します。

A0~AF, B0~BF, C0~C7が可能です。

⑦

メモリーの内容を10進数で書き替えることができます。

とすれば、A2の変数の内容が30になります。

A0~AF, B0~BF, C0~C1が可能です。

Dpの点灯している1文字が削除され、それ以降の文字が繰り上がります。

1行全て、"F"となります。(1行クリア)

Dpの点灯している所が、"F"となり桁が繰り上がります。

表示している行も含め、それ以降のプログラムが1行上がり、新たな行が追加されます。

表示行が削除され、それ以降の行が繰り上がり、表示行には次行のプログラムが表示されます。

エディタへ戻ります。

表示している行のプログラムを記憶して、次行を表示します。このキーを押さないとプログラムは記憶されません。

表示している行の進み、戻しをします。

行内でのカーソルの移動をします。

⑨ QMCLパラメータモードになります。このモードからエディタモードに戻るには キーを押して下さい。

⑩ ユーザモードになります。このモードからエディタモードに戻るには キーを押して下さい。

⑪ QMCLパラメータ設定値を初期化します。

⑫ ユーザパラメータの設定値を0にします。

6 プログラムの書き込み方法

タイマーで寸動運転するプログラムを作り、MITTYサーボに書き込んで動かしてみます。

6-1. プログラム例

```
CALL $460
VFB=1000
SFT=6000
SEVCC=1
L00 HZP=960 ; 30HZ指令
TIC1=410 ; 1秒タイマー
L01 JNE L01 TIC1
HZP=0 ; 停止
TIC1=205 ; 0.5秒タイマー
L02 JNE L02 TIC1
JMP L00
```

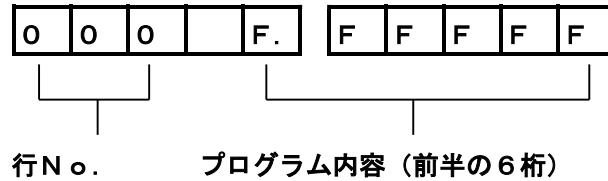
6-2. 中間言語に翻訳 (コマンド表で翻訳、または、コンパイラソフトで自動翻訳)

行No.	QMCL中間言語
000	F7 CF 04 60 FF
001	E7 D0 10 00 FF
002	E8 D0 60 00 FF
003	EF D0 01 FF
004	E1 D0 09 60 FF
005	EA D0 04 10 FF
006	F5 06 EA FF
007	E1 D0 00 FF
008	EA D0 02 05
009	F5 09 EA
010	F1 04

6-3. プログラム入力

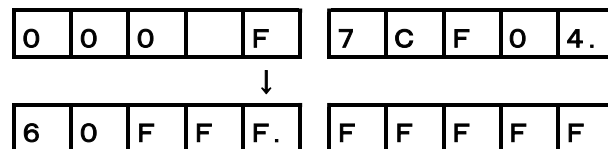
(1) MITYサーボの電源をONします。エディタとなります。まず、中に入っているプログラムをいったん、クリアします。N1 CR でクリアされます。プログラムモードにします。F 0 CR で、0行目よりプログラムモードとなります。

(0行目のみ 0 を省略し、F CR でも可)



(2) 2. で中間言語に翻訳したプログラムをキーボードより入力していきます。

0行目 F 7 C F 0 4 6 0 と入力する。



(3) 0行目を入力したら、CR キーを押して下さい。プログラムは0行目として記憶され、表示は次の行に進みます。



(4) 以降、10行目まで同様に入力していきます。



(5) 10行目の入力が終了したら忘れずに CR キーを押します。



(6) 以上でプログラムの入力は終わりました。ここで、END キーを押してエディタに戻します。



6-4. プログラム実行

入力されたプログラムを実行します。JOB CR キーを押して下さい。0行目よりプログラムは実行され、運転します。このプログラムは停止するプログラムではありませんので停止するのは電源OFFにします。

また、100行目からのプログラムの場合

F 1 0 0 CR ; プログラム100行目を指定
 JOB 1 0 0 CR ; 100行目よりプログラムを実行

7 ディスプレイ・キー 配置図

7-1. 7SEG配置

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

7-2. KEYコード配置

(1) 内部KEY

7	8	9	13
4	5	6	12
1	2	3	11
0	16	17	10

19	23	27	31
18	22	26	30
15	21	25	29
14	20	24	28

(2) エディタKEY

7	8	9	D
4	5	6	C
1	2	3	B
0	STOR L. CLR	LOAD CLR	A

INC N1	INC	END	JOB
DEC N2	RUN INC	LINE DEL	MONI TOR
F	DATA ←	↓	OPT ION
E	ADR →	↑	CR

8 エラーコード表

エラーコード表

Er-0	過電流 (OC: モータ加速時)
Er-1	過電流 (OC: モータ加速時以外)
Er-2	過電圧 (OV: 回生動作時)
Er-3	過電圧・不足電圧 (OV: 回生動作時以外)
Er-7	演算除算エラー (0で除算した場合のエラー)
Er-8	TRAPエラー (マシン語ミス・ノイズによるCPU-RUN不能)
Er-9	タイマー割り込みが、ルチンまたはマシン語が、ルチンでリターン命令なし
Er-10	エンコーダ信号がカウントされない。
Er-11	システムパラメータ設定によるオーバロード検出
Er-12	エンコーダ相順が、モータ相順と不一致。 モータ正転指令で、エンコーダカウント値がアップカウント
Er-13	ホールセンサ電源電圧供給なし。(あるいは未接続)
Er-70	メモリ書込みエラー
Er-71	メモリクリアエラー
Er-72	PROM書込みエラー
Er-73	PROMクリア (\$FF) されていない。
Er-80	ジャンプ先が2048を超えている。
Er-81	データにCA, CBがある。
Er-82	データ入力ミス。
Er-83	先頭がD1~DB, CE, CFになっている。
Er-84	表示内容ミス。 CA10=200など。
Er-85	= がない。
Er-86	先頭が0~9 (数字) になっている。
Er-87	10進数にA~Fがある。
Er-88	表示指定ミス。 CA13= など。(正: CA31=)
Er-90	サブルーチンにリターンがない。
Er-91	リターンの戻り先がない。
Er-92	VFBが大き過ぎる。(31以下のこと)
Er-93	VFBデータ入力エラー
Er-94	VFA加減し過ぎる。ただし通電中
Er-95	VFAデータ入力エラー
Er-96	シリアル入力エラー

表示は、左にトラベル行数、右にエラーNo. が表示され、エディタモードとなります。
出力は全てOFFします。OC, OVの時はアラーム信号もOFFします。

初版 1999.09.07
第2版 2000.02.28
第3版 2002.11.12
第4版 2015.06.25
第5版 2018.06.22
第6版 2021.09.28