

## 第4章 制御プログラムの基礎

BL/1 では、それぞれの制御アクチュエータに対応した様々なコマンドが用意されていますが、一度に習得するのは大変な作業となります。ここでは、制御内容に合わせて、いくつかの代表的なコマンドを紹介합니다。これによって、おおまかなプログラムイメージを描いていただけるものと思います。

\* 実際のコマンドの使用方法については、コマンド・リファレンスを参照してください。

なお、実際的な応用例とプログラム記述については、別冊にて、MPC-2000 チュートリアルがあります。また、その内容に沿ったトレーニングキット XY03 も用意されています。

チュートリアルでは、トレーニングキットの装置に基づいて、XY 制御、タッチパネルインターフェース、CUnet の操作、通信の基本的なプログラムを網羅しています。当社 WEB でも常時資料公開しております。

### 4-1 I/O 制御

#### ON/OFF

BL/1 での ON/OFF 制御は、ON/OFF コマンドによって行います。たとえば、0.1 秒での ON/OFF の繰り返しは、以下のとおりです。DO ~ LOOP は制御文で、LOOP に会うと、DO の直後に戻るという意味です。

```
DO
    ON 1
    TIME 100
    OFF 1
    TIME 100
LOOP
```

#### センサ・入力論理検出

次にあるセンサを検出して、検出したら 1 回、ON/OFF するというプログラムにします。

```
DO
    WAIT SW(193)==1
    ON 1
    TIME 100
    OFF 1
    TIME 100
    WAIT SW(193)==0
LOOP
```

最初の WAIT SW(193)==1 は、入力ポート 193 に接続されたセンサがオンになるのを待つという意味です。WAIT SW(193)==0 は、センサがオフになるのを確認しています。これにより、変化に対してのみ ON/OFF を繰り返すこととなります。

#### 条件・論理演算

I/O 制御では、複雑な論理演算をすることもあります。たとえば、先の例では、SW(193) の条件のみでしたが、SW(192) もオンであることを条件に加えるには、以下のようにします。

(SW(193)&SW(192)) は、SW(192) と SW(193) の値の AND 演算です。このため、両方の値が 1 でないと、(SW(193)&SW(192)) の値は、1 になりません。これにより、両方オンという条件になります。

```
DO
    WAIT (SW(193)&SW(192))==1
    ON 1
    TIME 100
    OFF 1
    TIME 100
    WAIT SW(193)==0
LOOP
```

なお、SW 関数には別途反転した値を持つ @SW() 関数が用意されています。

```
WAIT (SW(193)&@SW(192))==1
```

この場合、@SW(192) が 1 は反転論理のため、オフで 1 となります。したがってこの例では、193 がオン、192 がオフという AND 条件で成立することになります。

こうした論理式は WAIT 文のほか、IF,WHILE 文で使用しますが、式の値がすべて 1 になったときに正論理としています。

従って、WAIT SW(192)==1 と WAIT SW(192) は同じタイミング待ちとなります。比較演算子 == は、比較結果が等しいときに 1、等しくないときに 0 となる演算のため、SW(192)==1 では、SW の値が 1 となり、さらに比較演算で 1 に等しくなるとしているからです。

これにより、SW の複雑な論理条件も簡単に記述することができます。

```
IF SW(192)|SW(193)|SW(194)|flag THEN 192,193,194 のいずれかが ON もしくは、変数 flag が 1 になったら成立
IF (SW(192)&SW(193))|SW(194) THEN 192 と 193 が双方 ON もしくは、194 が ON
IF (SW(192)&SW(193))@SW(194) THEN 192 と 193 が双方 ON もしくは、194 が OFF
```

## タイムアウト処理

タイムアウト処理には、timer\_ (ダウンカウント変数) を使います。timer\_ に正数を与えると 0.1 秒ごとにデクリメントされ 0 で停止します。したがって以下のような記述をすることによりタイムアウトを含んだタイミング処理が可能です。

```
timer_=1000
WAIT (SW(192)==1) | timer_==0
IF timer_==0 THEN :GOTO *TMOUT:END_IF
```

なお、timer\_ 変数を外部タスクから参照、変更するには、TIMER() 関数が有効です。

## 4-2 タッチパネル接続

### MEWNET プロトコル

MPC-2000 では、各シリアル・ポートに MEWNET 対応のタッチパネル・ディスプレイを接続することができます。

MPC-2000 に付属のシリアルポートは、RS-232 のみですが、拡張シリアルボード MRS-MCOM では、RS-422 でも使用可能です。プロトコルは MEWNET のみです。MEWNET は松下電工の FA 用メモリリンクプロトコルです。

MEWNET プロトコルでは、PLC の複雑な接点機能に対応した、非常に多くの手順が用意されておりますが、MPC-2000 では、メモリアリアを DT 属性、I/O エリアを R 属性に想定しており、この 2 つに関係したプロトコルのみに対応しています。それ以外の属性を含んだパネル・プログラムや、MEWNET 対応とされている他社パネルでも接続できない場合がありますので事前に接続確認をお願いします。

現在まで、接続の確認がとれているタッチパネルは以下のとおりです。

Panasonic 電工	GT シリーズ (AIGT0030,AIGT2032 など)
デジタル	GP-2000/3000 シリーズ (AGP-3300 など)
三菱	GOT シリーズ (GT-10XX)
キーエンス	VT3 シリーズ (VT3-Q5M,VT3-W4T)

MEWNET を起動するには、以下の一行をプログラム先頭部分に書き加えます。このコマンドは一度実行されると、プログラムの実行状態にかかわらずタッチパネルとリンクされます。リンクされれば、データが共有され、通信を意識することなく、タッチパネルにデータの表示、あるいはタッチパネルからのデータの設定を行うことができます。

```
MEWNET 38400 1
```

第1引数 38400 はボーレートを意味します。反応速度の点から 38400 を推奨します。  
 次の引数は使用するシリアルポートの CH 番号です。(キャラクタフォーマットは 8bit ノンパリティです。)  
 タッチパネル通信には、タスクがひとつ割り当てられますが、シリアルの CH 番号で決定されますので注意してください。

また、1部のタッチパネルで、奇数パリティ固定のものがありますが、この場合はキャラクタフォーマットを指定する以下の定数を追加します。

```
B7O    bit7 奇数パリティ
B7E    bit7 偶数パリティ
B8O    bit8 奇数パリティ
B8E    bit8 偶数パリティ
```

以下は、3800bps bit7 奇数パリティの場合

```
MEWNET 38400 1 B7O
```

使用されるタスク番号は 32-CH 番号となります。

したがって CH1 を指定した場合 (MPC-2000,2100 に付属の CH1) はタスク 31 がタッチパネル通信に引き当てられます。よって、この場合タスク 31 をプログラム中で使用したり不用意に QUIT すると、タッチパネル通信が損なわれます。

## メモリ配置

タッチパネルとのメモリ共有は、MPC 側では、MBK() という予約配列を用います。MBK はワード型の配列で、8192 個確保されています。うち 0 ~ 7835 までをワードデータとして使用します。7836 ~ 7899 までは、システムがタスクごとの実行プログラム番号を常時書き込んでいます。

使用例:

```
IF MBK(100)==10 THEN
  MBK(200)=1000
```

7900 ~ 7999 までは IO エリアとしており、ON/OFF コマンドで操作することができます。

ON 70000 → バンク 0 のポート 0 をオン

OFF 70115 → バンク 0 のポート 15 をオフ

MBK エリア	
0 ~ 7835	<div style="border: 1px solid black; padding: 5px; width: fit-content;">ワードデータ</div> <p>0 ~ 9 は一般的にシステムエリアとして使用される。            変更 S_MBK n m もしくは MBK(m)=n            参照 MBK(m)</p>
7868 ~ 7899(Wrd) 7836 ~ 7899(Lng)	<div style="border: 1px solid black; padding: 5px; width: fit-content;">プログラム番号</div> <p>リアルタイムでプログラム番号が更新されており、            タッチパネル側で参照、表示できる。            注) S_MBK LONG_PRG を実行すると Lng 型となる。</p>
7900 ~ 7999	<div style="border: 1px solid black; padding: 5px; width: fit-content;">IO エリア</div> <p>0 ~ 100 バンク (YY) で 16bit ずつのエリア (XX)            ON 7YYXX, OFF YYXX            OUT 1000 7YY00~Lng            SW(7YYXX), IN(7YY00)</p>

## 4-3 時間管理

MPC-2000 及び 2100 では RTC が搭載され、日時時間を得ることができます。搭載 RTC は、エプソン・トヨコム製 RTC-7301 で月誤差 1 分程度となっています。MPC-1000 にはこの機能はありません。

### 設定

カレンダー IC の使用にはまず初期設定が必要です。設定は SET\_RTC コマンドを用います。

```
SET_RTC 2009 4 1    2009 年 4 月 1 日に設定します。
SET_RTC 12 2 0      12 時 2 分 0 秒に設定します。
```

設定した日時の確認は、date(0),time(0) 関数で行います。

```
#prx date(0)
20090401
#prx time(0)
00120204
#
```

### 時間検出

指定日時を検出するには、以下のように数値比較で行います。指定にはヘキサ型定数で指定します。

```
IF TIME(0)==&H130500 THEN      (13 時 5 分 0 秒)
IF DATE(0)==&H20090401 THEN    ( 2 0 0 9 年 4 月 1 日)
```

以下の例では毎分 5 秒と 15 秒を検出しています。必要な桁のみを有効にすることによって毎時、毎日といった複雑な時間検出も可能です。

```
DO
  WAIT &HFF&TIME(0)==&h0005
  PRINT "time_05"
  WAIT &HFF&TIME(0)==&h0015
  PRINT "time_15"
LOOP
```

### 日時文字列

日時文字列は、DATE\$(0),TIME\$(0) を用います。数値 0～2 によって様式が変わります。

```
10  FORMAT "00000000"
20  a$=DATE$(0)
30  FORMAT "000000"
40  a$=a$+TIME$(0)
60  PRINT a$
#run

2009090900141849
#pr time$(1)
14:19:02
#pr date$(1)
9/9/2009
#
```

## 4-4 軸制御

パルス発生ボードには MPG-2541 と MPG-2314 があります。MPG-2541 は補間や途中停止を含まない単純位置決め用です。対して、MPG-2314 は、直線・円弧補間、センサ停止など複雑な用途に対応することができます。MPG-2314 は 10 枚、MPG-2541 は 8 枚まで拡張できるため、ソフト上では 18 枚×4 軸に対応することができます。(RACK が 16MAX スロットのため、実際にはスロット数制限となります)

### PG のアサイン

どの PG を使用するのかは、PG コマンドで設定します。MPG-2314 が 0～9 の DSW 値に対応します。MPG-2541 の場合は、DSW 値に 10 を加えた値を私用します。(PG 10 等)

### 加速度・速度

ACCEL、FEED、SPEED コマンドが用意されています。

ACCEL は最高速度と最低速度と加速度を決定します。S 字加減速の有無もここで指定します。FEED コマンドは 1～100(%) の引数を与えて最高速度の m% という速度指定を行います。対して SPEED コマンドは、pps 指定です。ACCEL で決定した、最高速度の範囲で m pps として速度を指定します。(ただし分解能は、最高速度の 1/8192pps となります。)

### パルス発生

実際のパルス発生には、以下のようなコマンドがあります。用途により使い分けます。

コマンド	用途	説明
<b>MOVS</b>	位置決め	加減速度パルス発生 絶対位置指定 補間無し
<b>RMVS</b>	位置決め	加減速度パルス発生 相対位置指定 補間無し
<b>MOVL</b>	XY ステージ等	加減速度パルス発生 絶対位置指定 直線補間あり
<b>RMVL</b>	XY ステージ等	加減速度パルス発生 相対位置指定 直線補間あり
<b>MOVT</b>	塗布ロボット NC	軌跡制御連続パルス発生 絶対位置指定 円弧・直線補間
<b>RMVT</b>	塗布ロボット NC	軌跡制御連続パルス発生 相対位置指定 円弧・直線補間
<b>RMVC</b>	スピンドル	無限パルス発生
<b>STOP</b>	汎用	パルス発生を途中で停止させるコマンド
<b>HOME</b>	補助コマンド	原点復帰マクロコマンド

### 設定とエラー

位置決めには、各種異常状態の検出や安全上必要なインターロックがあります。MPG-2541 ではリミット入力のみですが、MPG-2314 ではリミット入力の外、サーボドライバエラー入力、検出停止入力があります。

コマンド / 関数	用途	説明
<b>INCHK</b>	保守	MPG 各種入力 表示
<b>INSET</b>	エラー入力設定	LMT 論理設定、ALM
<b>STOP</b>	停止条件入力	特殊原点復帰や途中停止条件を定める
<b>PGE()</b>	停止原因一読取り	PG 停止後 :EMG,ALM,LMTn,LMTp IN3,IN2,IN1,IN0
<b>LMT()</b>	エラー要因読取り	常時参照 :EMG,ALM,LMTp,LMTn,SLMTp,SLMTn
<b>HPT()</b>	IN0～3 入力	原点入力などの読み取り
<b>RR()</b>	動作状態	PG 動作中かどうかの判定

## パルス発生プログラム例

以下に MPG-2314 を用いた場合の最低限必要なコマンドとそのプログラムを例示します。

PG 0	PG の指定 DSW が 0 の MPG-2314
ACCEL X_A 10000 1000 1000	最高速度 加速度 自起動の設定
INSET X_A LMT_ON	リミット入力の論理設定
CLRPOS	現在位置の初期化
DO	
WAIT SW(192)	タイミング待ち
ROVS X_A 1000	1000 パルス発生
WAIT RR(X_A)==0	パルス出力終了待ち
LOOP	

## パレットサイズ

パルス発生コマンドではありませんが、マトリクス自動演算機能である PALLET コマンドとその関数である PL() が用意されています。マトリクス演算は、4 点あるいは 3 点いずれでも定義可能で、パレットは 0 ~ 63 まで指定することができます。これによりパレット分類や配列組み換えなどのロボット・アプリケーションを容易に構築できます。

## MPC-1000 パルス発生機能について

MPC-1000 では、サブ CPU として PIC16F88 を 2 個搭載し、それぞれパルス発生器として使用することができます。

- \* この PG 機能は、コマンド通信に 0.1 ~ 0.2 秒程度要します。
- \* この PG 機能は、PIC の内部オシレータを使用するため、速度指定には +/-2% 程度の誤差があります。
- \* 加減速付きパルス発生のパルス幅は 15  $\mu$ 秒固定です。

### 【占有ポート】

それぞれのパルス発生器は、PGA,PGB と名づけられており以下のポートを占有します。

	出力ポート	入力ポート
PGA	ON 12 (CW パルス)	SW(192+12) READY
	ON 13 (CCW パルス)	SW(192+13) 通信用
PGB	ON 14 (CW パルス)	SW(192+14) READY
	ON 15 (CCW パルス)	SW(192+15) 通信用

### 【PG の有効化】

PGA,PGB を有効にするには、それぞれ

```
ON PGA
ON PGB
```

を実行します。また、無効化するには、

```
OFF PGA
OFF PGB
```

とします。無効状態では、I/O を占有しなくなり、それぞれ制御用の I/O として使用できます。また、ON/OFF は PG のソフトリセットも兼ねています。パルス発生を途中停止させる場合は、それぞれに対して、OFF PGA,OFF PGB を実行し 10msec 以上のオフ時間を確保してください。

【PG コマンド】 PGX は PGA もしくは PGB

機能	コマンド	範囲	補記	READY
パルス方式	PGX "D" n	0 or 1	0: デフォルト 2PLS 1: 方向指示	
PWM	PGX "W" n	40 ~ 970	DA としても使用可	
PPS 指定パルス発生	PGX "G" pps	20 ~ 9000		
パルスレート設定	PGX "S" pps	20 ~ 9000		
パルス数指定パルス発生	PGX "P" count	-8000000 ~ 8000000		○
加減速テーブル生成	PGX "A" pps	500 ~ 12000		○
速度選択	PGX "F" n	10 ~ 0	n*10 %	
加減速パルス発生 相対	PGX "R" count	-8000000 ~ 8000000		○
加減速パルス発生 座標	PGX "M" count	-8000000 ~ 8000000		○
現在位置クリア	PGX "H" count		現在位置設定	
現在位置取得	PGX "C"		PGA は V_PGA	
バージョン取得	PGX "V"	20091105 以降	PGB は V_PGB	

\*READY に○記号があるものは、指定数パルス発生など実行終了待ちが必要なコマンド

【使用方法 1】 パルス発生として

```

*PGAPGB
TIME 300
ON PGA PGB                PGの有効化
WAIT SW(192+12)==1       有効化確認
PGA "V":PRINT V_PGA      バージョン取得と表示
DO
FOR i=20 TO 6020 STEP 1000
  PGA "G" i                パルスレートを 20 ~ 6020 まで変化させる。
                           (CW)
TIME 100
NEXT
FOR i=6020 TO 20 STEP -1000
  PGA "G" 0-i             パルスレートを 6020 ~ 20 まで変化させる。
                           (負の値で CCW)
TIME 100
NEXT
TIME 100
PGA "G" 0                 G コマンド停止
PGA "S" 2000              パルスレート設定
PGA "P" 1600              1600 パルス発生 CW (加減速なし)
WAIT SW(192+12)
PGA "P" -1600             パルス発生中は READY=0 となる。
WAIT SW(192+12)          1600 パルス発生 CCW (加減速なし)

```

【使用方法 2】 位置制御/パルス発生として

```

PGA "A" 9000              加減速度テーブルを 9000pps/s
WAIT SW(192+12)          テーブル生成終了待
PGA "H" 0                 現在位置指定
,
FOR j_=5 TO 10
  PGA "F" j_              速度指定
FOR i_=1 TO 10
  PGA "R" 800             相対パルス発生
  WAIT SW(192+12)
NEXT
PGA "M" 0                 0 位置に移動。座標移動
NEXT

```

【コマンド解説】 PGX は PGA もしくは PGB の意味です。

パルス方式	PGX "D" n	CW,CCW のニパルス方式か方向指示を指定します。 デフォルトは、ニパルス方式です。 方向指示に変更する場合は、PGX "D" 1 を実行します。
PWM	PGX "W" n	PWM パルス発生です。CW 側のみでパルス発生します。 デフォルトで 1kpps のパルスレートで $n \mu \text{sec}$ のオン時間を規定します。 パルスレートを変更する場合は、PGX "S" pps を先に実行し、PGX "W" n をあとで実行します。 停止させるには、OFF PGA(PGB) するか PGX "W 0" を実行します。
PPS 指定パルス発生	PGX "G" pps	定速パルス発生です。指定したレートでパルスを発生します。正の値で CW, 負の値をいれると CCW となります。 起動後、速度を変更することができます。 停止させるには、OFF PGA(PGB) するか PGX "G" 0 を実行します。
パルスレート設定	PGX "S" pps	PWM とパルス数指定パルス発生のパルスレートを決定します。
パルス数指定 パルス発生	PGX "P" count	定速、指定パルス発生です。正の値で CW, 負の値で CCW 方向にパルス発生します。
加減速テーブル生成	PGX "A" pps	加減速付きパルス発生の速度テーブルを生成します。加速距離は、指定パルスレートの 1/10 で固定されています。 加減速は、フラッシュ ROM に固定されます。 フラッシュ ROM の書き換えは 10 万回未満とされていますので、注意してください。(同じ引数の場合は書き換えを行わないようになっていきます。)
速度選択	PGX "F" n	A コマンドで指定された速度を 10 段階に分割し、 $n/10$ 速度指定を行います。 加減速テーブルの変更は時間がかかりますが、F による速度変更は時間がかかりません。
加減速パルス発生 相対	PGX "R" count	加減速付き、パルス発生です。座標管理されており、M コマンド、C コマンドと併用することができます。
加減速パルス発生 座標	PGX "M" count	加減速付き、パルス発生です。現在位置と指定位置の差分パルス発生します。 座標管理されており、R コマンド、C コマンドと併用することができます。
現在位置クリア	PGX "H" count	現在位置指定です。count を 0 にすると、原点となります。
現在位置取得	PGX "C"	結果は PGA,PGB それぞれ予約変数 V_PGA,V_PGB に返されます。
バージョン取得	PGX "V"	結果は PGA,PGB それぞれ予約変数 V_PGA,V_PGB に返されます。

## 4-5 データ通信

### RS-232/RS-485

MPC-2000 では 10CH のシリアル通信を扱うことができます。CPU ボード単体では、RS-232C しか扱うことはできませんが、MRS-MCOM を用いると、RS-422,RS-485 通信にも対応することができます。受信通信割り込みバッファは各 CH とも 256byte と十分に用意されています。  
但し、MPC-1000 では CH1 を RS-485 として使用する事ができます。

### 1) 設定

設定は CNFG# 1 "38400b8pns1NONE" というように CNFG コマンドを使用します。1200bps から 38400bps の各種フォーマットに対応します。RS-485 通信をする場合は、以下のようにします。

```
CNFG# 5 RS485 "38400b8pns1NONE"
```

予約定数 RS485 を引数に与えることによって、自動的に通信方向を切り替えるようになります。

### 2) 送信

PRINT# コマンドを用います。PRINT# 文は、基本的には文字列を用いてください。変数も扱うことはできますが、書式を規定できません。文字列中に "\n"(LF), "\r"(CR), "\t"(TAB)", を用いることはできますが、その他の制御文字は、CHR\$( ) を用います。

```
PRINT# CHR$(1) "DATA" CHR$(3)
```

### 3) 受信

INPUT# コマンドを用います。INPUT# 文は文字列変数のみを引数に与えることができます。文字列として受け取ったあとに、VAL 関数、GET\_VAL、SERCH コマンドなどによって内容を分析、データを取得します。

### 4) オプション

INPUT#,PRINT# 文には、受信文字数、タイムアウト時間、デリミタ、コードなどを指定することもできます。また、特殊なオプションに COMPOWAY,STR\_LEN があります。COMPOWAY は OMRON が規定する、ベーシック手順のプロトコルですが、このフォーマットの自動送信、受信をサポートしています。SRT\_LEN はヌルコードを含む文字列の送出に使用します。

## CUnet

MPC-2000 では、パソコンとのデータ共有も CUnet によって実現されています。CUnet は(株)ステップテクニカが開発しデバイス販売を行っている FA 用ネットワークで 512byte のメモリイメージをネット・ワーク上で共有することができます。ステーション数は最大 64 台までとなっており、ステーション間で 2.5msec 以内に共有メモリが同期するように設計されています。CUnet を使用するには、MPC-2000 側では、MPC-CUnet2、PC 側には、USB-CUnet が必要となります。

なお、MPC-2000 間のインターロック、機種情報の交換に限れば、IN,OUT,SW,ON,OFF などの IO コマンドで共有メモリを参照変更することができるため、高速分散制御を簡単に構築することができます。

注) MPC-CUnet2 を起動するには、初期化コマンド CUNET の実行が必要です。

PC 側では、アクセル純正のデバイスドライバ、DLL、メモリイメージ監視用ユーティリティが用意されており、VB 等を利用してユーザサイドで CUnet 対応、専用アプリケーションを製作することができます。

USB-CUnet の資料、ソフトウェアについては、当社 Web サイトからダウンロードして御使用ください。

## USB メモリ

MRS-MCOM 及び MPC-1000 には USB メモリ用のポートが付属しています。用途としては、機種切り替え用データ、プログラムや点データのバックアップ保存、リアルタイムでのログデータの書き込みなど、様々です。

大手メーカー製の一般的な USB メモリは使用可能ですが、稀に相性が発生しますので現物での確認が必要です。

### 1) 機種切り替え

```
USB_PSAVE {UPS}
USB_PLOAD {UPL}          *{} 内は省略型
```

ups p(100) 100 "101.p2k" のように、点データのエリア指定して保存することができます。保存したデータは、USB\_PLOAD コマンドで取り出すことができます。取り出し方は上書きのため、特定のエリアのデータ USB メモリから入れ替えて使用することができます。なお、UPS/UPL は MBK データも対象にすることができます。

注意) 点データですべての値が 0 となっているデータは保存されません。これは、大量の点データを保存する場合の時間短縮のためです。

## 2) ログ書き込み

```
USB_WRITE {UWR}
```

装置の稼働率やアナログデータのある時間間隔において書き足す用途は多くあります。ロギング、あるいはデータ・ロガーという用途ですが、MPC-2000では、USB\_WRITE コマンドで簡単に校正することができます。ファイル名は、予約変数 FILE\$ に指定します。

```
FILE$="TEST.TXT"  
DO  
    UWR "1234567890\n"  
    TIME 1000  
LOOP
```

この例では "TEST.TXT" というファイルに延々と文字列 "1234567890\n" を書き足していきます。そして、このプログラムはいつ停止させられても、最後に書き込んだ行まで有効になります。

## 3) 通常のファイル操作

一般的な DOS-BASIC と同様、MPC-2000 にも OPEN,CLOSE が用意されています。OPEN,CLOSE を用いたアクセスでは、PRINT#,INPUT# を用い、シリアル通信と同様に行います。以下は、USB メモリの TEST.CSV を読み出して表示する例です。ポート番号は USB を用います。

注) 扱えるファイルは、テキスト形式のみです。

```
10 OPEN "TEST.CSV"  
20 TIME 200  
30 DO  
50 INPUT# USB a$  
60 PRINT a$  
70 IF LOF(USB)==0 THEN :BREAK :END_IF  
80 LOOP
```

## 4-6 アナログ制御

アナログ制御には、MPC-AD12 を用います。AD/DA ともコマンドで簡単に扱うことができますが、MPC-AD12 は 2 枚まで実装することができ、AD 入力 16CH,DA 出力 8CH まで用意することができます。

### AD 変換

関数 AD() を用います。MPC-AD12 の標準状態では、0 ~ 4095 の値が得られますが、1digit が 1mv に対応します。A=AD(0) で A が 1000 となれば、入力が 1000mV つまり 1V であったということになります。AD 関数には、平均値取得モードもあり、MPC-AD12 によって自動的に平均値計算された値を得ることもできます。

### DA 変換

DA 出力にはコマンド DA を用います。DA 1000 1 と実行すれば、DA-CH1 に 1000mV つまり 1V が出力されます。

### 各種設定

AD コンバータの平均値のサンプル数設定や、AD7890-10 への交換時の設定など、SET\_AD コマンドが用意されています。