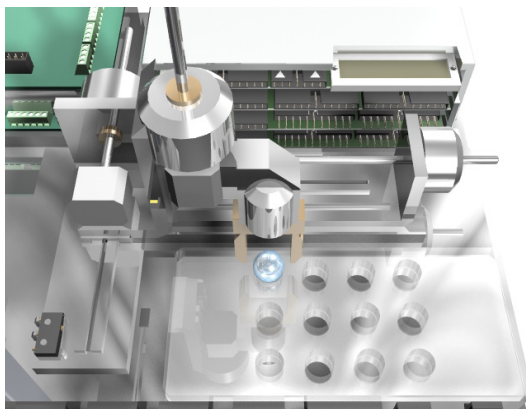


第7章 プログラム例

7-1 ロボットアプリケーション

ここでは、簡単なプログラム例で、基本的なプログラム記述を例示します。
想定は、以下のような4×3のマトリクス配置されたガラス玉をつかんで、原点で排出するものです。
こうしたアプリケーションでもPGの設定や原点復帰などが必要となります。



```
CONST HAND 15
CONST READY 0
CONST BUSY 1
CONST START 192
PG 1
ACCEL X_A|Y_A|SACL 50000
ACCEL Z_A|SACL 40000
GOSUB *HOME
PALLET 1 P(1) P(2) P(3) P(4) 4 3
```

```
DO
ON READY
QUIT 1:TIME 10
OFF BUSY
WAIT M_SW(START)
OFF READY
FORK 1 *BUSY
FOR I=1 TO 12
JUMP PL(1;I)
WAIT RR(ALL_A)==0
ON HAND
TIME 300
JUMP 0 0 0 0
OFF HAND
TIME 300
NEXT
LOOP
END
*BUSY
```

I/O の定義

このように CONST を用いて、I/O をシンボル化しておくことでプログラムの見通しが良くなる。

PG の選択。DSW1 の PG を選択

速度・加速度の設定を行う。
このように軸ごとに設定可。SACL は S 字指定
サブルーチンコール
パレット宣言。ここでは 4 点指定。

繰り返し制御文

スタートスイッチ待ち。チャタがあるときは M_SW

別タスクで、BUSY(点滅)表示
順次処理、1～12 のワークを取り出す。

パルス発生(移動)終了の監視
ガラス玉をつかむ

排出位置に運ぶ
ガラス玉を放す

順次処理終端
繰り返し終端

別タスクのプログラム

DO	
ON BUSY	LED 点滅を繰り返す単純なプログラム
TIME 100	
OFF BUSY	
TIME 100	
LOOP	
*HOME	原点復帰プログラム
STOP ALL_A VOID	
RMVS 1000 1000 0 -1000	原点位置脱出
WAIT RR(ALL_A)==0	
SPEED 1000	速度 1kPPS に設定
STOP ALL_A IN0_ON	停止条件を指定する。(原点センサ)
RMVS -100000 -100000 0 100000	原点サーチ
WAIT RR(ALL_A)==0	停止したら終了
STOP ALL_A VOID	停止条件を解除
CLRPOS	現在位置を 0 にする
SPEED 30000	速度をもとに戻す
RETURN	サブルーチンからもどる。

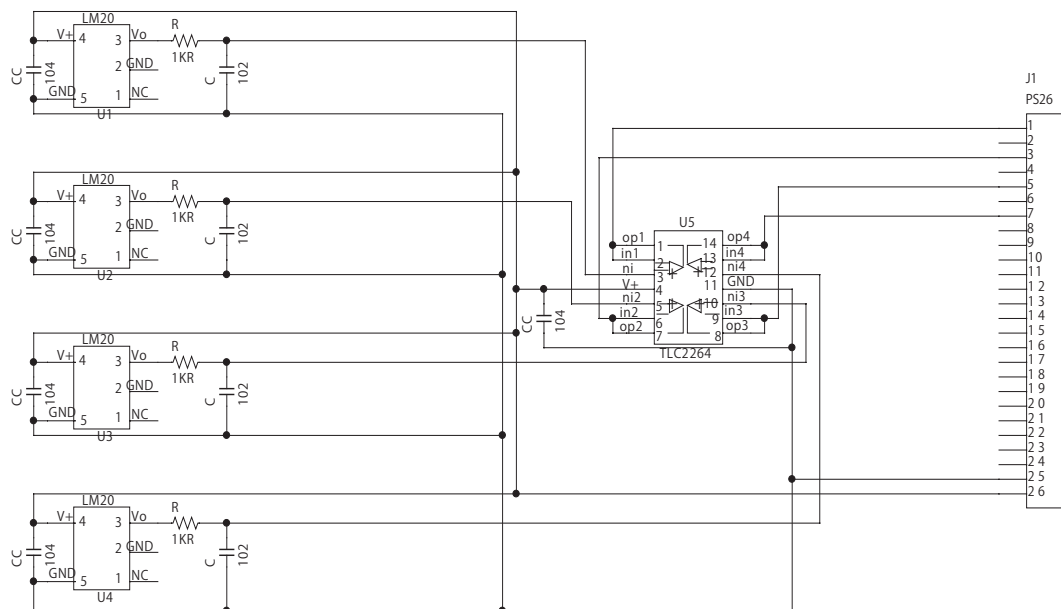
ここでは、プログラムの見通しをよくするために、いくつか条件を設定しています。パワーオン時はほぼ原点の近くに XY があること。それにより、原点サーチにより停止すれば必ずそこが原点だという前提です。実際には、原点サーチ抜け後、センサ状態などの確認が必要です。

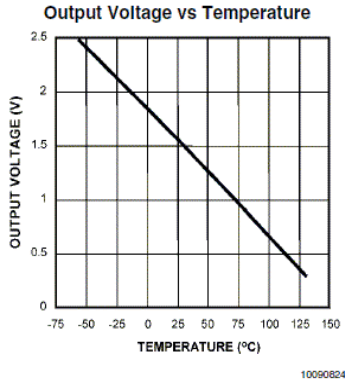
7-2 温度計測とデータロガー

MPC-AD12 と温度センサ LM20 を組み合わせることにより、多点温度計測ロガーを構成できます。センサ一部の回路図は以下のとおりです。OP アンプとセンサ IC の間は、1m 程度は引き伸ばすことができます。以下回路図で J1 は MPC-AD12 に接続、センサ部の電源は MPC-AD12 からの給電で動作します。

ここでは、カレンダー IC に同期して 10 秒ごとに温度データを計測し USB メモリに記録保存するプログラムを例示します。

【LM20 参考回路図】





```

DIM DATA(4)
TIME 500
FILE$="log_S.txt"
err_cnt=0
err$=" "
*ondo
FORK 1 *disp_lcd
ON_ERROR *usb_err
DO
  WAIT TIME(255)%16==0
  FOR I=0 TO 3
  DATA(I)=(1866-AD(1,I))*100/117
  NEXT
  DISP$=""
  FOR I=0 TO 3
  FORMAT " CH0_"
  DISP$=DISP$+STR$(I)
  FORMAT "00.0"
  DISP$=DISP$+STR$(DATA(I))
  NEXT
  FORMAT "000"
  OND1$=STR$(DATA(0))
  OND2$=STR$(DATA(1))
  I$=DATE$(3)+" "+TIME$(1)+DISP$+"n"
  ON 768
  USB_WRITE I$
  PRINT I$
  TIME 1000
  OFF 768
  LOOP
*usb_err
RST_USB
INC err_cnt
FORMAT "00"
err$=STR$(err_cnt)
PRINT "ERRORUSB"
TIME 500
RESUME
END
*disp_lcd
DO
  lcd$=err$+OND1$+OND2$
  PR_LCD lcd$
  TIME 5000
  LOOP

```

以上のようにして簡単なデータロガーを構成することができます。

【温度データへの変換方法】

LM20 は左のような特性を持ちます。
 0度で 1.86V で、1度ごとに -11.7mV 変化します。
 MPC-AD12 は出荷状態で 1mV/1digit です。
 このため、以下の演算で 3 桁の温度データを得ることができます。

$$A=(1866-AD(0))*100/117$$

A が 254 であれば、25.4 度ということになります。

USB メモリのログファイルを指定。
 (八文字 .TXT DOS 形式)

LCD への表示タスクを起動
 USB メモリ書き込みエラー規定
 繰り返し
 10 秒ごとののタイミング待ち
 4ch の繰り返し
 センサ電圧を温度に変換

カウント繰り返し CH0 ~ CH3

チャンネルごとに文字列に変換

LCD 用の文字列生成

日付データ文字列生成

日付、時間、温度データを合成
 書き込みランプ点灯
 USB メモリに書き込む
 FTM に表示

書き込みランプ消灯

ここは、USB メモリ書き込みエラーの処理

エラーカウントを文字列に変換

エラー発生コマンドにもどる。

MPC の LCD にエラー回数と
 CH0,CH1 の温度を表示

USB メモリに書き込まれるデータは以下のようになります。普通の PC で参照できます。

```
"log_S.txt"  
2009-05-29 10:36:30 CH0_22.2 CH1_21.5 CH2_21.6 CH3_22.0  
2009-05-29 10:36:40 CH0_22.2 CH1_21.5 CH2_21.6 CH3_22.1  
2009-05-29 10:36:50 CH0_22.2 CH1_21.5 CH2_21.6 CH3_22.1
```

なお、USB メモリはフラッシュメモリを内蔵したメモリデバイスのため、10 万回程度の書き換えにしか耐えられません (メーカーによってバラつきあり)。1 秒に 1 回の書き込みを実行すると、24 時間で、86400 回の書き込みが発生し、メモリを劣化させます。

当社の試験では、1 秒間隔の書き込みで 1 週間ほど連続運転すると破損するものもありました。

このため、なるべく書き込みは MPC 内にバッファして USB_WRITE を実行する回数を減らすようにしてください。

また、USB メモリは、書き込みエラーが発生する可能性のあるデバイスです。

稼動を停止させないために、適切なエラー処理 (ON_ERROR) を組み込んでください。