

実数・関数演算ルーチンの改訂版

金野茂男

1. 始めに

最近、先年に著者の書き上げた実数・関数演算ルーチンを使用して、新たなH I Dの開発を思い立った。が、プログラムの先頭部分での変数や配列変数の宣言部で、アセンブラエラーが発生することに気が付いた。2007年に作成した「光学マウスキットを応用した卓上型経路長・面積計 - その2」⁽¹⁾中のPIC18F252に、その実数・関数演算ルーチンが納まっている。このプログラムで解決方法を探した。要は以下の通りである。2010年になり、マイクロチップ社の提供している最新の「MPLAB IDE v8.53」をマイクロチップ社のURLからダウンロードした。この「MPLAB IDE v8.53」中の「MPASMWIN」中のアセンブラ「MPASM V5.36」で、参考文献(1)に付属のアセンブラプログラムをアSEMBルすると、変数・配列宣言部分がエラーを出すようになってしまっていた。なぜそのようなになったのかはわからない。それではと思い、同じ文献に付属のHEXファイルのPICへの書き込みも試みた。正常に書き込めた。書き込みの点では、最新の「MPLAB IDE v8.53」には問題はない事はわかった。あれやこれやの試行錯誤の結果、アSEMBルが正常行えるようにプログラムを修正することができた。以下その報告をする。

著者の公開しているPIC関係には、上述の実数・関数ルーチンが納まっているのは少なくない。万が一、著者と同様のアSEMBルの失敗に陥っているならば、本論文を参考にして、プログラムの変数宣言部分に若干の変更を加えればよいであろう。

2. プログラムの変更

以下に参考文献(1)に添付していたアSEMBラプログラムの先頭にある宣言部分に、「MPLAB IDE v8.53」で正常にアSEMBルできるように修正を加えたプログラムを掲載している。極力、前のプログラムを消さないようにして、修正を加えている。「;」より右の文は注釈行とし、アSEMBルから除外しているところである。前のプログラムと、ここで紹介しているプログラムを比較してみるのも良いであろう。

参考文献(1)の旧アSEMBラを修正し、ハード回路はそのままに、最新「MPLAB IDE v8.53」でアSEMBルし、そして同じPIC18F252に正常に書き込みができ、前の通り、装置が正常に動作することは確認済みである。

著者の他の装置の場合でも、該当する箇所を下のプログラムを参照しながら、修正していけばよい。或いは、著者の提供しているプログラムを参考にして作成しているプログラムでも、同じ箇所を修正すれば、正常にアSEMBリングができよう。

```
*****
;
*****
;
;org h'000' ;バンク0に設ける

;変数の宣言
CBLOCK h'000'

    temp0    ;res 1    ;一時的カウンタ変数
    temp1    ;res 1    ;
    temp2    ;res 1    ;
    counter  ;res 1    ;カウンタ変数
    current  ;res 1    ;一時変数
    charnum  ;res 1    ;表示する文字数を入れる
    LCD_mode ;res 1    ;

    mosi_data ;res 1 ;MOSIポートに送り出すデータを納める
    miso_dta  ;res 1 ;MISOポートから出力されるデータを納める
    motion    ;res 1 ;MOTIONレジスタの内容を受け取る変数
    delta_x   ;res 1 ;delta_Xレジスタの内容を納める変数
    delta_y   ;res 1 ;delta_Yレジスタの内容を納める変数

    sum_x     ;res 1 ;X方向の移動量 連続2バイト - 3 2 7 6 8 ~ + 3 2 7 6 8 b15=符号ビット
    sum_y     ;res 1 ;Y方向の移動量 連続2バイト - 3 2 7 6 8 ~ + 3 2 7 6 8 b15=符号ビット

    figure100 ;res 1 ;1バイト16進数を10進数に変換したときに
    figure10  ;res 1 ;桁の数値が収まる変数
    figure1   ;res 1 ;
```

```

sign      ;res 1 ; +、 - のLCD用データが収まる

sig       ;res 1 ; 数値の仮数符号（第1ビット）と指数符号（第0ビット）が納まる
fig:8     ;res 8 ; 数値の仮数が納まる。8桁分用意する。
exp       ;res 1 ; 数値の指数値が納まる

xsig      ;res 1 ; 数値の仮数符号（第1ビット）と指数符号（第0ビット）が納まる
xfig:8    ;res 8 ; 数値の仮数が納まる。8桁分用意する。
xexp      ;res 1 ; 数値の指数値が納まる

ysig      ;res 1 ; 数値の仮数符号（第1ビット）と指数符号（第0ビット）が納まる
yfig:8    ;res 8 ; 数値の仮数が納まる。8桁分用意する。
yexp      ;res 1 ; 数値の指数値が納まる

sumxsig   ;res 1 ; 数値の仮数符号（第1ビット）と指数符号（第0ビット）が納まる
sumxfig:8 ;res 8 ; 数値の仮数が納まる。8桁分用意する。
sumxexp   ;res 1 ; 数値の指数値が納まる

sumysig   ;res 1 ; 数値の仮数符号（第1ビット）と指数符号（第0ビット）が納まる
sumyfig:8 ;res 8 ; 数値の仮数が納まる。8桁分用意する。
sumyexp   ;res 1 ; 数値の指数値が納まる

xi_1sig   ;res 1 ; 数値の仮数符号（第1ビット）と指数符号（第0ビット）が納まる
xi_1fig:8 ;res 8 ; 数値の仮数が納まる。8桁分用意する。
xi_1exp   ;res 1 ; 数値の指数値が納まる

yi_1sig   ;res 1 ; 数値の仮数符号（第1ビット）と指数符号（第0ビット）が納まる
yi_1fig:8 ;res 8 ; 数値の仮数が納まる。8桁分用意する。
yi_1exp   ;res 1 ; 数値の指数値が納まる

xisig     ;res 1 ; 数値の仮数符号（第1ビット）と指数符号（第0ビット）が納まる
xifig:8   ;res 8 ; 数値の仮数が納まる。8桁分用意する。
xiexp     ;res 1 ; 数値の指数値が納まる

yisig     ;res 1 ; 数値の仮数符号（第1ビット）と指数符号（第0ビット）が納まる
yifig:8   ;res 8 ; 数値の仮数が納まる。8桁分用意する。
yiexp     ;res 1 ; 数値の指数値が納まる

tanasig   ;res 1 ;
tanafig:8 ;res 8 ;
tanaexp   ;res 1 ;

tanctsig  ;res 1 ;
tanctfig:8 ;res 8 ;
tanctexp  ;res 1 ;

pathsig   ;res 1 ; 数値の仮数符号（第1ビット）と指数符号（第0ビット）が納まる
pathfig:8 ;res 8 ; 数値の仮数が納まる。8桁分用意する。
pathexp   ;res 1 ; 数値の指数値が納まる

stepsig   ;res 1 ; 数値の仮数符号（第1ビット）と指数符号（第0ビット）が納まる
stepfig:8 ;res 8 ; 数値の仮数が納まる。8桁分用意する。
stepexp   ;res 1 ; 数値の指数値が納まる

asig      ;res 1 ;
afig:8    ;res 8 ;
aexp      ;res 1 ;

bsig      ;res 1 ;
bfig:8    ;res 8 ; 3角形のヘロンの公式の3辺の数値が納まる
bexp      ;res 1 ;

csig      ;res 1 ;
cfig:8    ;res 8 ;

```

```

cexp ;res 1 ;

dsig ;res 1 ;
dfig:8 ;res 8 ;
dexp ;res 1 ;

ssig ;res 1 ;
sfig:8 ;res 8 ;作業変数群
sexp ;res 1 ;

areasig ;res 1 ;
areafig:8 ;res 8 ;面積値が納まる
areaexp ;res 1 ;

parasig ;res 1 ;
parafig:8 ;res 8 ;縮尺からの換算数値が納まる
paraexp ;res 1 ;

num_counter ;res 1 ;算用数値が入る
number:4 ;res 4 ;整数値最大4桁の換算数値が納める
fig_number ;res 1 ;上記数値を決める時に使用するいわゆる桁数
unit ;res 1 ;単位対応の数値が納まる。 0=mm,1=cm,2=m,3=km
my_status ;res 1 ;状態指定ビットとして使用する

```

ENDC

```
#define input_flag my_status,0 ;=0で入力中、=1で入力終了
```

```

;*****
;*****
;
;
; L C D の制御用ポートの割り当て。
#define contl_E PORTA,2 ; 0=L C D 不動作、 1=L C D 動作
#define contl_RW PORTA,1 ; 0=L C D へ書き込み 1=L C D から読み出し
#define contl_RS PORTA,0 ; 0=命令、 1=記号ノバイト

#define LED_monitor PORTA,3 ;ポートAの第3ビットをモニターに使用
#define lcd_data PORTB ;L C D データ用ポートのBポートにラベル名を付ける。

```

```

; L C D に転送する各命令のコードの定義
; 使用する L C D のテクニカルノートを参照すること
;
; 2進数 処理
clear equ 1 ; 00000001 表示をクリア
home equ 2 ; 00000010 カーソルをホームへ
shift_l equ 24 ; 00011000 表示を左へ移動
shift_r equ 28 ; 00011100 表示を右へ移動
crsr_l equ 16 ; 00010000 カーソルを左へ移動
crsr_r equ 20 ; 00010100 カーソルを右へ移動
crsr_e equ 32 ; 00100000 ファンクションをセット
blink_c equ 11 ; 00001011 表示オフ、カーソルオン、ブリンクオン
no_crsr equ 8 ; 00001000 表示オフ、カーソルオフ、ブリンクオフ
ret_bak equ 192 ; 11000000 改行とリターン

```

```

; L C D に表示する文字コードの定義
column equ b'00111010'
degree equ b'11011111'
period equ b'00101110'
charlimit equ 16 ; L C D の 1 行の表示可能文字数
allchar equ 31 ; L C D に表示できる総文字数

```

```

; L C D に表示する各文字の文字数
Laboname equ 15 ; '**Kinno Labo**'
DeviseName equ 13 ; ' Opt Sencer '

```

```

; A D N S - 5 0 3 0 端子と接続する P I C の端子の命名
#define NRESET PORTC,4 ;リセットピン。L 入力でアクティブ
#define NCS PORTC,3 ;チップ選択。L 入力でアクティブ
#define SCLK PORTC,2 ;シリアルクロック入力
#define MOSI PORTC,1 ;シリアルデータ入力
#define MISO PORTC,0 ;シリアルデータ出力

;
#define center_sw PORTC,7 ;
#define right_sw PORTA,5 ;クロススイッチの名前
#define left_sw PORTC,6 ;
#define up_sw PORTA,4 ;
#define down_sw PORTC,5 ;

;*****
;***** 浮動小数点演算用変数定義 バンク 5 に設ける *****
;*****
;org h'500' ;バンク 5 に設ける
CBLOCK h'500'

xsig_B5 ;res 1 ;先行数値の仮数符号 (第 1 ビット) と指数符号 (第 0 ビット) が納まる
xfig_B5:8 ;res 8 ;先行数値の仮数が納まる。8 桁分用意する。
xexp_B5 ;res 1 ;先行数値の指数値が納まる

ysig_B5 ;res 1 ;後行数値の仮数符号 (第 1 ビット) と指数符号 (第 0 ビット) が納まる
yfig_B5:8 ;res 8 ;後行数値の仮数が納まる。8 桁分用意する。
yexp_B5 ;res 1 ;後行数値の指数値が納まる

zsig_B5 ;res 1 ;結果数値の仮数符号 (第 1 ビット) と指数符号 (第 0 ビット) が納まる
zfig_B5:8 ;res 8 ;結果数値の仮数が納まる。8 桁分用意する。
zexp_B5 ;res 1 ;結果数値の指数値が納まる

xwork_B5:32 ;res 32 ;3 2 バイト配列で加減算を行うための配列その 1
ywork_B5:32 ;res 32 ;3 2 バイト配列で加減算を行うための配列その 2
zwork_B5:32 ;res 32 ;3 2 バイト配列で加減算を行うための配列その 3

asig_B5 ;res 1 ;仮数符号 (第 1 ビット) と指数符号 (第 0 ビット) が納まる
afig_B5:8 ;res 8 ;ワーク配列。浮動小数点型の仮数が納まる。8 桁分用意する。
aexp_B5 ;res 1 ;ワーク変数。浮動小数点型の指数値が納まる

bsig_B5 ;res 1 ;仮数符号 (第 1 ビット) と指数符号 (第 0 ビット) が納まる
bfig_B5:8 ;res 8 ;ワーク配列。浮動小数点型の仮数が納まる。8 桁分用意する。
bexp_B5 ;res 1 ;ワーク変数。浮動小数点型の指数値が納まる

csig_B5 ;res 1 ;仮数符号 (第 1 ビット) と指数符号 (第 0 ビット) が納まる
cfig_B5:8 ;res 8 ;ワーク配列。浮動小数点型の仮数が納まる。8 桁分用意する。
cexp_B5 ;res 1 ;ワーク変数。浮動小数点型の指数値が納まる

dsig_B5 ;res 1 ;仮数符号 (第 1 ビット) と指数符号 (第 0 ビット) が納まる
dfig_B5:8 ;res 8 ;ワーク配列。浮動小数点型の仮数が納まる。8 桁分用意する。
dexp_B5 ;res 1 ;ワーク変数。浮動小数点型の指数値が納まる

counter_B5 ;res 1 ;表示した文字数のカウンタ変数
current_B5 ;res 1 ;一時変数
point_B5 ;res 1 ;小数点用のカレント変数
repeat_B5 ;res 1 ;漸化式の反復回数

;関数演算時に使用するワーク変数
esig_B5 ;res 1 ;仮数符号 (第 1 ビット) 指数符号 (第 0 ビット) が納まる
efig_B5:8 ;res 8 ;ワーク配列。浮動小数点型の仮数が納まる。8 桁分用意する。
eexp_B5 ;res 1 ;数値の指数値が納まる

fsig_B5 ;res 1 ;仮数符号 (第 1 ビット) 指数符号 (第 0 ビット) が納まる
ffig_B5:8 ;res 8 ;ワーク配列。浮動小数点型の仮数が納まる。8 桁分用意する。
fexp_B5 ;res 1 ;数値の指数値が納まる

```

```

gsig_B5      ;res 1      ;仮数符号(第1ビット) 指数符号(第0ビット)が納まる
gfig_B5:8    ;res 8      ;ワーク配列。浮動小数点型の仮数が納まる。8桁分用意する。
gexp_B5      ;res 1      ;数値の指数値が納まる

hsig_B5      ;res 1      ;仮数符号(第1ビット) 指数符号(第0ビット)が納まる
hfig_B5:8    ;res 8      ;ワーク配列。浮動小数点型の仮数が納まる。8桁分用意する。
hexp_B5      ;res 1      ;数値の指数値が納まる

ksig_B5      ;res 1      ;仮数符号(第1ビット) 指数符号(第0ビット)が納まる
kfig_B5:8    ;res 8      ;ワーク配列。浮動小数点型の仮数が納まる。8桁分用意する。
kexp_B5      ;res 1      ;数値の指数値が納まる

psig_B5      ;res 1      ;仮数符号(第1ビット) 指数符号(第0ビット)が納まる
pfig_B5:8    ;res 8      ;ワーク配列。浮動小数点型の仮数が納まる。8桁分用意する。
pexp_B5      ;res 1      ;数値の指数値が納まる

tri_fun_sel_B5 ;res 1      ;三角関数の種類を選択する変数
cord_para0_B5 ;res 1      ;三角関数演算ルーチン内で使用できる変数その0
cord_para1_B5 ;res 1      ;三角関数演算ルーチン内で使用できる変数その1

exp_para0_B5  ;res 1      ;指数関数ルーチンで使用する変数その0
exp_para1_B5  ;res 1      ;指数関数ルーチンで使用する変数その1
expvar_B5     ;res 1      ;指数関数ルーチンで使用する変数

log_para0_B5  ;res 1      ;対数関数ルーチンで使用する変数その0
log_para1_B5  ;res 1      ;対数関数ルーチンで使用する変数その1
logvar_B5     ;res 1      ;対数関数ルーチンで使用する変数

```

ENDC

;定数の宣言

limit_B5 equ 32

3. 終わりに

マイクロチップ社の旧MPLABで、正常に開発できていたプログラムが、最新MPLABでは正常にアセンブリができなくなっている。この理由はわからない。不正常的の1つは、旧バージョンでは問題の無かった配列変数を簡単に定義できた「RES」ディレクティブである。新バージョンでは受け付けてくれない。アセンブラ文章の書き方によっては「RES」を受け付けるようになるが、他の所でおかしくなる。何故か。エラーメッセージも良くわからない。PICSTART Plusライターに付属しているマニュアルには、何らの注釈もなく、どのPICでも、どのMPLABのバージョンでもRESの使用ができるような説明なのだが。

その他、不可思議な箇所は幾つかある。が、とにかく上記のように旧プログラムを修正すれば、プログラムは正常に動作する。

今後は、その時点での開発環境を詳細に記述すべきであることを改めて認識した。今回の開発環境はパソコンはWindows XPである。

参考文献

(1) 光学マウスキットを応用した卓上型経路長・面積計 - その2」金野茂男、小山高専電子制御工学科、2007年6月著者のURLで公开发表。

2010年8月11日