「はじめての PIC アセンブラ入門」訂正と補足

光永 法明

平成 20 年 5 月 22 日

1　訂正 (第 4 版)

「はじめての PIC アセンブラ入門」(第 4 版) について、表 1 の通りお詫びして訂正します。

表 1: 第 4 版の正誤表

	誤	正
p.32 本文 6 行目	movf に対して movfw と覚えやすく	movwf に対して movfw と覚えやすく

2　訂正 (第 3 版, 第 4 版)

「はじめての PIC アセンブラ入門」(第 3 版, 第 4 版) について、表 2 の通りお詫びして訂正します。

表 2: 第 3 版, 第 4 版の正誤表

	誤	正
p.155 (周期) を求める式内	(PR レジスタ)	{(PR2 レジスタの値)+ 1}
p.156 リスト 6-2 の T1MS の定義内の数値	T1MS equ D'125'	T1MS equ D'124'

3　訂正 (初版から第 4 版)

「はじめての PIC アセンブラ入門」(第 1 版から第 4 版) について、表 3(本文), 4 (Appendix C の comf, decf, rlf, rrf, sublw, swapf の例題) の通りお詫びして訂正します。

表 3: 第 1 版から第 4 版の正誤表 (本文)

	誤	正
p.37 下から 2 行目	リスト 2-1	リスト 3-1

表 4: 第 1 版から第 4 版の正誤表 (Appendix)

	誤	正
COMF (p.256), DECF (p.257) の例題の例 2 の実行後の w レジスタの値	0x13	- (実行前と同じ)
RLF (p.267), RRF (p.268) の例題の例 2 の実行後の w レジスタの値	0x20	- (実行前と同じ)
SUBLW (P.270) の例題の例 2 の実行後の STATUS レジスタの値	DC C	C
SWAPF (p.271) の例題の例 2 の実行後の w レジスタの値	0xA5	- (実行前と同じ)

4 訂正 (初版, 第 2 版, 第 3 版)

「はじめての PIC アセンブラ入門」(第 1 版, 第 2 版, 第 3 版) について、表 5 の通りお詫びして訂正します。

5 訂正 (初版, 第 2 版)

「はじめての PIC アセンブラ入門」(第 1 版, 第 2 版) について、以下の通りお詫びして訂正します。

5.1 コラム (p.34) について

コラム内の define は #define に訂正します (define でもアセンブルはエラーにはなりません)。

5.2 図 4-1 (p.61) について

矢印の向きが 1 部間違っています。正しくは、30 ピン (RD_7/PSP_7) については双方向矢印、32 ピン (V_{DD}) については IC へ入る向きの矢印となります。

5.3 表 5-2 (p.102) について

5 行目と 6 行目の ADCON0< 5 >, ADCON0< 4 > を、それぞれ ADCON1< 5 >, ADCON1< 4 > に訂正します。

5.4 p.140 本文 6 行目について

「1/2 から 1/128 にするときは」を「1/2 から 1/256 にするときは」と訂正します。

表 5: 第 1 版, 第 2 版, 第 3 版の正誤表

	誤	正
p.17 表 2-1 (10 進数のところ)	2 5 5	255
p.18 12 行目	8 桁と 2 桁の差 になります	2 進数では 8 桁に 16 進数では 2 桁 になります
p.23 図 2-14 の脚注	物理的に存在しない	間接アクセス用。p.30 2-17 節 レジスタ・ファイルの間接アクセスを参照。
p.25 図 2-16 の脚注	-n = POR リセット後の値	-n = POR リセット後の値 (-0: 0, -1: 1, -x: 不定)
p.25 下から 3 行目	-x は	-n は
p.32 本文 5 行目から 6 行目	movf 命令 の保存先がファイル・レジスタになっている のと同じです。movf に対して movfw と	movf 命令 で保存先を w レジスタにする のと同じです。movwf に対して movfw と
p.41 リスト 3-3, 9〜10 行目 のコメント部	w=1, z=0, C=/BO=0	w=1, z=0, C=/BO=1
p.41 リスト 3-3, 13〜14 行目 コメント部	小さかったら C=/BO=1	小さかったら C=/BO=0
p.41 本文 5 行目	C フラグのチェックは btfss でも	C フラグのチェックは btfsc でも
p.58 下から 5 行目	FSR と INDF を 使った	FSR と INDF を 使って
p.60 コラム 4, 5 行目	AARGB1 と BARGB0 を 掛けた結果	AARGB0 と BARGB0 を 掛けた結果
p.60 コラム 4, リスト 3-A 14 行目 (LUM0808NAP の次の行)	BTFSC STATUS, C	BCF STATUS, C
p.153 図 6-4		TMR0IF のところの横に長い矢印のところに「20[ms]」と追加
p.204 本文の下から 1 行目 p.205 図 7-12 内	PGP	PGM
P.205 図 7-11 のキャプション	OSCCAl	OSCCAL
p.269 SUBLW の (i)	この命令は 引き算方向が 逆なので 注意する	この命令は 定数 k から w レジスタの値を引くので 注意する
p.270 SUBWF の (i)	この命令は 引き算方向が 逆なので 注意する	この命令は ファイル・レジスタの値から w レジスタの値を引くので 注意する

表 6: タイマ 2 の周波数と周期について (p.155-156) の正誤表

	誤	正
p.155 の 1 行目	PR2 レジスタを 10	PR2 レジスタを 9
p.155 の 4 行目	PR2 に設定した 10 になると	PR2レジスタ に設定した 9 になった 次のカウントアップのときに
(周波数) と (周期) を求める式内	(PR レジスタ)	{(PR2 レジスタの値)+ 1}
62.5[kHz] を求めている式の分子	4 × 10 × 1	4 × (9+1) × 1
周期 (1/62.5[kHz]) を求める式内	× 10 ×	× (9+1) ×
「●割り込みを使ってサーボを動かすプログラム」の 4 行目	125	124
p.156 リスト 6-2 の T1MS の定義内の数値	T1MS equ D'125'	T1MS equ D'124'

5.5 リスト 5-15 (p.143) について

8 行目 (MYTIMER の定義) と 9 行目 (W_TEMP の定義) の間に、

```
PCLATH_TEMP     equ     0x7d
```

を追加してください。このアドレスもバンク切替えの影響が出ないアドレスです。

また、21 行目の PCLATH を保存する部分 (movfw 命令) ですが、", W" は不要です。

```
        movfw   PCLATH      ; PCLATH を保存する
```

としてください。

5.6 タイマ 2 の周波数と周期について (p.155-156)

表 6 のように訂正します。

5.7 図 7-8 (p.197) について

LED は、抵抗を通して GP1 へ接続してください (図 1)。

5.8 リスト 7-10(p.200) について

プルアップ設定の movlw 命令の値を訂正します。正しくは以下です。

```
movlw   B'111100'
movwf   WPU
bcf     OPTION_REG, NOT_GPPU
```

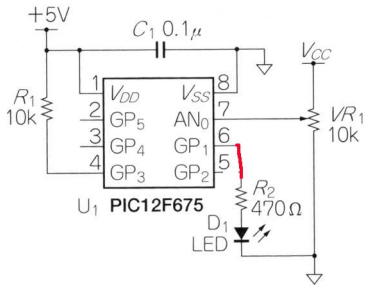


図 1: LED は、抵抗を通して GP1 へ接続してください (図 7-8 (p.197) の訂正)

5.9 図 7-10 (p.203) について

ANS3:ANS0 の解説について訂正します。正しくは以下です。

ANS3:ANS0 : アナログ選択ビット
(AN_3 から AN_0 についてディジタル/アナログ機能の選択)
'1' = アナログ入力 (1)
'0' = ディジタル入出力 (ポート) またはその他の機能

6 訂正 (初版)

「はじめての PIC アセンブラ入門」(初版) について、以下の通りお詫びして訂正します。

6.1 図 2-10 (p.20) の計算について

図 2-10 (p.20) の減算の答えに間違いがありました。正しくは 011 です (図 2)。

$$\begin{array}{r} 0\ 1\ 1 \\ +\ 0\ 0\ 1 \\ \hline 1\ {}^{1}0\ 0 \end{array} \qquad \begin{array}{r} 1\ 0\ 0 \\ -\ 0\ 0\ 1 \\ \hline {}^{1}0\ {}^{1}1\ 1 \end{array}$$

図2-10　2進数の加算と減算の筆算

図 2: 図 2-10 の計算の訂正

6.2 p.45 上から 2 行目について

「レジスタ REGLOOP1 に」は削除してください。

6.3 p.112, p.113 RCSTA レジスタ

p.112 の下から 3 行目と, p.113 図 5-15 のキャプションに RXSTA レジスタとありますが、正しくは RCSTA レジスタです。

6.4 LCD の DB_0 から DB_3 の接続について

回路図では LCD の下位 4 ビット DB_0 から DB_3 が、グラウンドに繋がっていますが、接続は不要です (図 3)。p.122 本文の最下行に「DB0 から DB3 はグラウンドに接続し」とありますが、「DB0 から DB3 には何もつながず」と訂正します。

6.5 図 4-6 全体配線図 (p.69) について

図 4-6 全体配線図 (p.69) では VR2 の中点から LCD の 5 ピンにつながっていますが、回路図通り 3 ピン (V_O) につないでください (図 4)。

6.6 リスト 5-1(p.101)

リスト 5-1 (p.101) の最終行は正しくは bcf STATUS, RP0 です。

6.7 リスト 5-5(p.108)

リスト 5-5 (p.108) について、本文とは逆に明るくなると点滅するようになっていました。12 行の btfsc を btfss としてください (図 5)。ダウンロードページのリストは訂正済みのものです。

7 補足

以下の通り本文に補足させていただきます。

7.1 AKI-PIC プログラマーについて

現在は AKI-PIC プログラマー ver.4 (完成ボード) として完成品も販売されています (2007 年 11 月確認)。また、サポートページが開設されています。

秋月電子 AKI-PIC プログラマー サポートページ
http://akizukidenshi.com/down/tk/picpgm_v4/index.htm

7.2 PCL レジスタ操作によるジャンプについて (p.53)

PCL レジスタ操作によるジャンプについて、第 1 版 p.53 下から 9 行目で「桁上がりのないアドレスになっていることを確かめてください」と書いていますが、「ジャンプ先の命令の一連のアドレス内で桁上がりがないことを確かめてください」と訂正します。PCLATH レジスタは、PCL

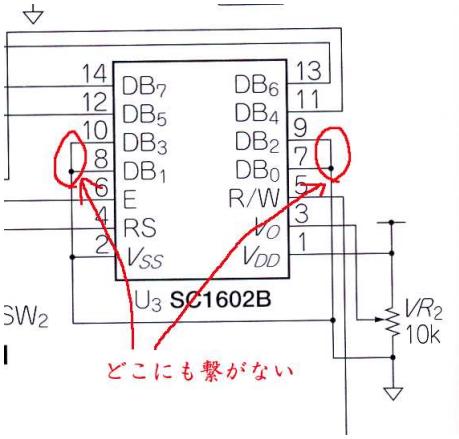


図 3: 回路図の訂正

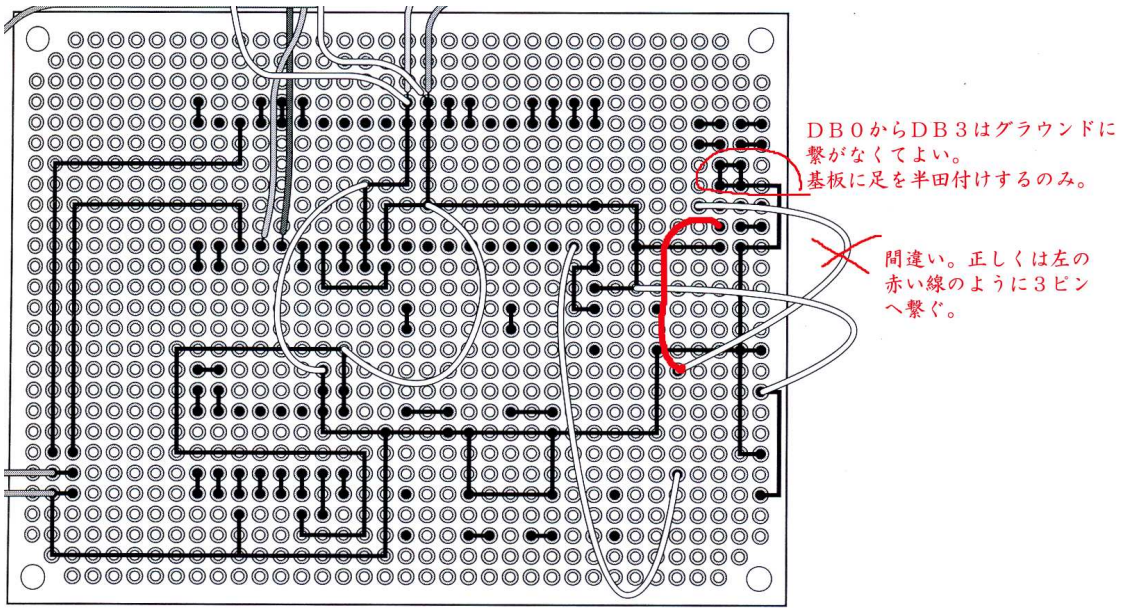


図 4: 図 4-6 全体配線図の訂正

リスト5-5　暗くなると点滅するプログラムのメイン・ルーチンと追加サブルーチン

```
loop
        ; A-D変換する
        bsf     ADCON0,GO     ; A-D変換開始          ┐①
                                                     │A-D変換
        btfsc   ADCON0,GO     ; A-D変換終了まで待つ   ┘
        goto    $-1

        movfw   ADRESH        ; wに変換結果を代入     ┐②
        sublw   D'128'        ; 暗さの閾値(128)と比較をする │暗いか
                              ; w−128の結果が正(Cが1)なら暗い┘チェック  ┐メイン・
        btfss   STATUS,C                                              │ルーチン
        goto    blink_LED ←── ③ 暗ければ1回点滅                       ┘
```

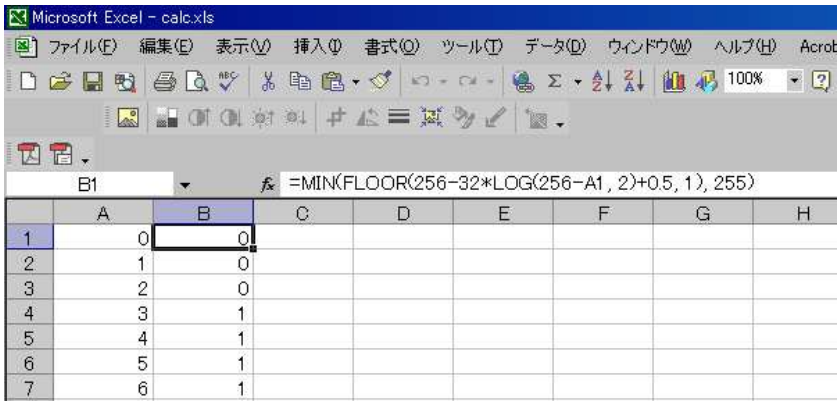
図 5: 訂正後のリスト 5-5 の抜粋

レジスタ操作後の値に応じて変えることはできないため、あらかじめジャンプ先の PCLATH に指定する値が同じになるようにプログラム時に配置する必要があるためです。

7.3　表 5-3 (p.105) の求め方

表 5-3 (p.105) を表計算で値を求める方法ですが、Excel ですと次のようにして求められます。

A1 を 0 に B1 を $= MIN(FLOOR(256 - 32 * LOG(256 - E1, 2) + 0.5, 1), 255)$ とします。A2 は $= A1 + 1$ とし B2 は B1 のコピーとします (図 6)。そして 2 行目を 3 行目から 256 行までにコピーします。B の列が求める数値です。



	A	B	C	D	E	F	G	H
1	0	0						
2	1	0						
3	2	0						
4	3	1						
5	4	1						
6	5	1						
7	6	1						

図 6: 表計算ソフトで表 5-3 の値を求める

計算の意味ですが、MIN() で最大値が 255 を超えないようにしています。FLOOR() 関数の中身は本文の式に 0.5 を足したものです。FLOOR() 関数は切捨て計算を行うので、0.5 をあらかじめ足しておくことで、四捨五入の計算としています。

なお、リスト 5-3 全体と表計算の例はダウンロードページ
http://www.cqpub.co.jp/shoseki/mycomputer/pic004/index.html

に掲載していますので、ご利用ください。

7.4 10 進数と 16 進数の変換

7-8 章 (p.190) のリスト 7-9(p.191) に出てくる数値を 10 進数に変換すると以下のようになります。

```
0x835A3FC1          -2091237439
  0x5C3BE5              6044645
0xFFFFFEA7                 -345
  0xA6F75E             -5834914
```

負数を 10 進数から 16 進数への変換を手計算する場合には、以下のようにします。

1. 正の数に変換します
2. 2 進数に変換します
3. ビットを反転します
4. 1 を足します
5. 16 進数に変換します

負数の 16 進数を 10 進数へ変換する場合には、以下のようにします。

1. 1 を引きます
2. 2 進数に変換します
3. ビットを反転します
4. 10 進数に変換します

10 進数から 16 進数への変換は、Windows 2000/XP に付属の電卓（アクセサリ→電卓）を利用して変換することができます。

1. 表示から関数電卓を選択し、関数電卓モードにします (図 7)
2. 10 進モードで 10 進数を入力します
3. 16 進モードにします (図 8)

16 進数から 10 進数へ Windows 2000/XP に付属の電卓で変換すると、そのままでは正の値として扱われてしまいます。そこで負の数（最上位ビットが 1、つまり最上位の桁が 0x8 から 0xf）であることがわかっている場合、次のようにすれば計算できます (0xA6F75E の場合です)。

1. 電卓を関数電卓、16 進数モードにします
2. FFFFFF から A6F75E を引きます
3. 10 進数モードにします
4. 1 を足します

図 7: Windows の電卓を関数電卓モードにする

図 8: Windows の電卓を 16 進数モードにする

5. 正負を反転します

(FFFFFF の桁数は変換する数の桁数に合わせます。)

あるいは 16 進計算機能つきの関数電卓を利用するのも便利です。シャープの EL-501EX などは安価ですが 16 進計算機能がついています。この電卓の場合は、桁数が 6 桁（24 ビット）の場合には先頭に FF をつけ 8 桁（32 ビット）にし（たとえば FFA6F75E）変換するとよいようです。

また 16 ビットの数値の変換なら MPASM に任せることもできます。w レジスタに代入する場合は次のようにします。

```
        movlw ((0x1234) >>8 & 0xff)    ; 上位バイト（8 ビット右にシフトして 0xff と
論理積をとる）
        movlw ((0x1234)     & 0xff)    ; 下位バイト（0xff と論理積をとる）
        movlw high(0x1234)             ; 上位バイト
        movlw low(0x1234)              ; 下位バイト

        movlw (-(D'30000') >>8 & 0xff) ; 上位バイト
        movlw (-(D'30000')     & 0xff) ; 下位バイト
        movlw high(-D'30000')          ; 上位バイト
        movlw low(-D'30000')           ; 下位バイト
```

7.5 「Source file path exceeds 62 characters」というエラーがでる

「Source file path exceeds 62 characters」というエラーはパス名が長すぎる場合に起こるエラーです。パス名というのは、ファイルの場所を表すもので、絶対パスというドライブ名からファイルにたどりつくまでの一連のディレクトリ名とファイル名をつないだものと、相対パスというあるディレクトリから対象のファイルまでのディレクトリ名とファイル名であらわす方法が使われています。MPLAB からアセンブラ MPASM はファイル名を絶対パスで受け取り、アセンブルを実行します。

たとえば、c ドライブに home というディレクトリをつくり、その中に、very_very_long_directory_name というディレクトリを、さらにその中に 0123456789012345678901234546789012345678901234567890123456789 というディレクトリ、その中に、PROG1 ディレクトリをつくって、PROG1.ASM を置いたとします。このとき PROG1.ASM の絶対パスは、ドライブレターからファイル名までを ¥ でつなげた、

```
C:\HOME\VERY_VERY_LONG_DIRECTORY_NAME\0123456789012345678901234546789012345678901234567
7890123456789\PROG1\PROG1.ASM
```

となります (Windows では大文字、小文字は内部では区別しません)。ところが MPASM にはパス名の長さに制限があるため、以下のようなエラーが出ます。

```
Executing: "C:\Program Files\Microchip\MPASM Suite\MPAsmWin.exe" /q /p16F877A "prog1.as
m" /l"prog1.lst" /e"prog1.err"
Error[173] C:\HOME\VERY_VERY_LONG_DIRECTORY_NAME\012345678901234567890123454678901234
5678901234567890123456789\PROG1\PROG1.ASM 14 : Source file path exceeds 62 characters
(C:\HOME\VERY_VERY_LONG_DIRECTORY_NAME\012345678901234567890123454678901234567890123
567890123456789\PROG1\PROG1.ASM)
```

```
Halting build on first failure as requested.
BUILD FAILED: Thu Apr 28 01:01:46 2005
```

エラーメッセージでは 62 文字を超えているとありますが、実際の制限はもう少し小さいようです。このような場合にはディレクトリ名を短くする、ディレクトリの階層（ドライブからファイルまでの間にあるディレクトリの数）を少なくすることでエラーを回避できます。

ソースファイルの入ったディレクトリをデスクトップやマイドキュメントにおくと、実際の絶対パスは、

```
c:\Documents and Settings\Administrator\デスクトップ\PIC\PROG1\prog1.asm
```

といったように、思ったよりも長くなり、このエラーに出会う可能性が高くなるようです。もしデスクトップにディレクトリを作られているようでしたら、パス名が短めのディレクトリにソースをおくようにするとエラーが出にくくなります。

7.6 MPLAB IDE エディタでの日本語表示について

著者のところでは、インストールを行った時点で日本語の表示と入力ができている (v.6.60, v7.40) ため、解決策となるか分かりませんが、以下を確認してみてください。

ソースファイルを開いて一番手前にある状態で、MPLAB IDE の Edit の中から、Properties をクリックし、ウィンドウを開きます。Text タブで、SelectFont をクリックし、適当な日本語を含むフォント (たとえば MS ゴシックや Terminal) を選び、OK をクリックして、閉じます。また Text タブ内の National Language Code Page が、932 (ANSI/OEM - Japanese Shift-JIS) となっていなければ変更します。フォントの変更はすぐに反映されますが、Code Page の変更は、一度ファイルを閉じ、もう一度開くと変更が有効になるようです。

7.7 予期しないアセンブラのエラー

書籍の通り、プログラムをエディタで打ち込んだつもりなのに、アセンブルでエラーが起こることがあります。見ためには問題ないように見える場合には、行頭など空白部分に全角スペースが含まれている場合があります (以下では全角空白 (スペース) を□で表します)。

図 9 は、問題の再現用に用意したソースを MPLAB のエディタで表示しているところです。このソースをアセンブルすると次のようなエラーになります。

```
Make: The target "H:\mydocs\test-src\lcd.o" is out of date.
Executing: "C:\Program Files\Microchip\MPASM Suite\MPAsmWin.exe" /q /p16F877A
"lcd.asm" /l"lcd.lst" /e"lcd.err"
Error[122]   H:\MYDOCS\TEST-SRC\LCD.ASM 6 : Illegal opcode (PORTB)
Error[116]   H:\MYDOCS\TEST-SRC\LCD.ASM 8 : Address label duplicated or
different in second pass (□□)
Halting build on first failure as requested.
BUILD FAILED: Sat Dec 22 10:47:08 2007
```

6, 8 行目でエラーが出ています。ところが 7, 9 行目ではエラーが出ていません。6 から 9 行目の行頭は一見すると正しく空白が入っているように見えます。しかし、6 行目のエラーは PORTB と

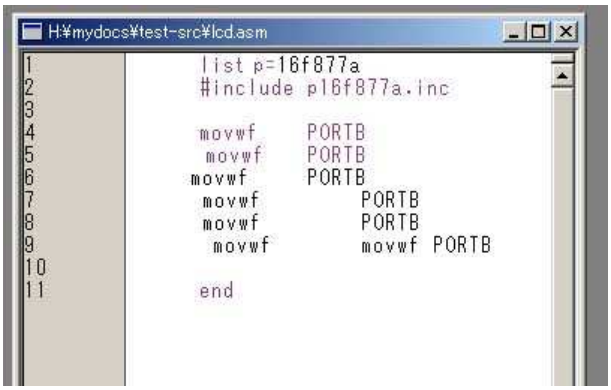


図 9: 一見問題がないように見えるソースファイルの例

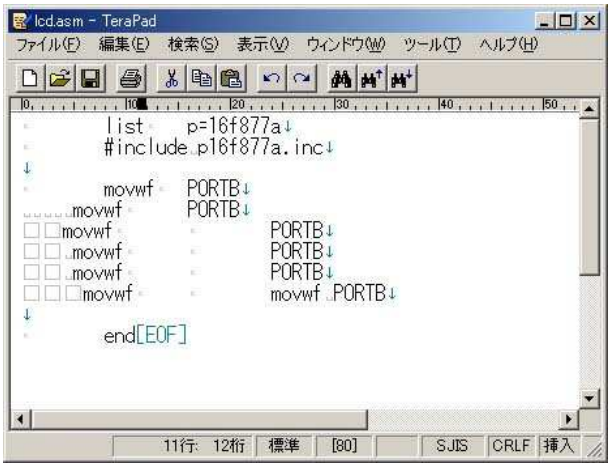


図 10: 同じソースファイルを TeraPad で表示したところ

いう opcode (アセンブラ命令, 擬似命令) は存在しないというエラーです。8 行目は、同じラベルが再定義されているというエラーです。これは、6 行目、7 行目、8 行目の行頭に全角空白が含まれているためです。TeraPad というエディタ (フリーウェアです。検索で簡単に見つけられます) で、全角空白、半角空白、TAB を表示するように設定して、見てみると図 10 のようになります。

6 行目の行頭は半角空白、7, 8, 9, 10 行目の行頭は全角空白になっています。MPASM は、半角空白と TAB については空白文字として扱いますが、全角空白（日本語空白）は空白としてではなく表示できる文字として扱ってしまいます。そのため、

6 行目 行頭は □□ movwf というラベル、命令は PORTB （Illegal opcode）

7 行目 行頭は □□ というラベル、命令は movwf PORTB (エラーにならない)

8 行目 行頭は □□ というラベル、命令は movwf PORTB。ラベルの 2 重定義エラー。

9 行目 行頭は □□□ movwf というラベル。命令は movwf PORTB (エラーにならない)

と MPASM は解釈しています。なお、上記の表示をするため、TeraPad は、オプションの表示タブで、図 11 のように全角空白などを表示する設定をしています。

図 11: TeraPad のオプション

8 参考情報

サポートページ: http://mycomputer.cqpub.co.jp/pic004/
ダウンロードページ: http://www.cqpub.co.jp/shoseki/mycomputer/pic004/
書籍紹介: http://www.cqpub.co.jp/hanbai/books/37/37391.htm
シリーズ一覧: http://www.cqpub.co.jp/hanbai/series/micon.htm

マイクロチップテクノロジージャパン: http://www.microchip.co.jp/
Microchip: http://www.microchip.com/

秋月電子 AKI-PIC プログラマー サポートページ:
http://akizukidenshi.com/down/tk/picpgm_v4/index.htm

9 改定履歴

2008/5/22 第 3 版，第 4 版: 訂正情報を追加
2008/4/25 初版〜第 4 版：Appendix C の訂正情報を追加
2007/12/22 全角空白が原因で起こるエラーについての補足情報を追加
2007/12/7 初版〜第 3 版の訂正情報を表 1 にまとめ直し、訂正情報と AKI-PIC プログラマ完成版について補足を追加
2007/11/2 初版〜第 3 版：p.32，p.41，p.60 について訂正情報を追加
2006/12/7 初版：p.45，RCSTA レジスタ (p.112，p.113)，LCD 接続 (p.122) の訂正情報を修正・追加。PCL レジスタ操作によるジャンプについて (p.53) の補足を追加。初版・第 2 版：コラム (p.34)，表 5-2(p.102)，本文 (p.140)，タイマ 2 の周波数・周期の求め方 (p.155-156) についての訂正情報を追加

2006/10/23 図 7-8（p.197），リスト 7-10(p.200)，図 7-10（p.203）の訂正情報を追加
2006/10/17 図 4-1（p.61），リスト 5-15（p.143）の訂正情報，MPLAB IDE エディタでの日本語表示についての補足情報を追加
2006/1/3　リスト 5-1（p.101）の訂正情報を追加