イントロダクション
本書の読み方とライブラリの使い方

本書の読み方

● 斜体の *x* や *y* はモジュールの番号を表す

関数やマクロの名前などに斜体の *x* や *y* が含まれている場合，その部分にはモジュールの番号(1，2，3…)が入ります．ただし，関数は用意してあってもモジュールそのものがない場合は使えないので，モジュールがいくつ備わっているかはデータシートと照らし合わせて確認してください．

● 引き数/戻り値の型は書式で確認

書式の部分には，関数の引き数や戻り値の型が示してあります．変数を渡す場合は，示されている型に合わせてください．型違いはエラーやワーニングの原因となります．

ライブラリの使い方

■ ライブラリ用のヘッダ・ファイルをインクルードする

ライブラリを使用する場合は，そのライブラリに対応するヘッダ・ファイルをインクルードする必要があります．各章のタイトル部分には，対応するヘッダ・ファイルの名前が書いてあります．

例えば，タイマ・モジュール・ライブラリの場合は timer.h というヘッダ・ファイルが必要なので，プロセッサ・ヘッダ・ファイルと同じようにプログラム・ソースの先頭に，

```
#include <timer.h>
```

という1文を追加します．

■ ライブラリに含まれる関数/マクロを使ってプログラムを記述する

● 定義済み定数の使い方

一部の関数には定義済みの定数があります．

例えば，周辺機能の設定を行う関数や，割り込みの設定を行う関数などです．**リスト1**は，timer.h で定義されている定数の一部です．それぞれの値が，レジスタの各ビットに対応するマスクになっていることがわかります．

レジスタの設定値は，これらの定義済み定数を“&”で複数個を組み合わせて作ります．例えば，タイマの設定を行う OpenTimer1() の場合は，

```
OpenTimer1(T1_ON & T1_GATE_OFF & T1_PS_1_8 &
           T1_SYNC_EXT_OFF & T1_SOURCE_INT, 0xFF);
```

といったようにします．

リスト1　timer.hで定義されている定数(抜粋)

```
/* Timer1 Control Register(T1CON)
                    Bit Defines */
#define T1_ON         0xffff
#define T1_OFF        0x7fff

#define T1_IDLE_CON   0xdfff
#define T1_IDLE_STOP  0xffff

#define T1_GATE_ON    0xffff
#define T1_GATE_OFF   0xffbf

#define T1_PS_1_1     0xffcf
#define T1_PS_1_8     0xffdf
#define T1_PS_1_64    0xffef
#define T1_PS_1_256   0xffff
```

● 定義済み定数の省略は避ける

定義済み定数は，設定したい内容によって一部省略できる場合があります．例えば，上の例ではアイドル・モード時の動作はどちらでもよいので，その部分は省略しています．実際には，マスクの組み合わせによってこのビットが‘1’になるため，アイドル・モードのときはタイマを停止するという設定になります．

むやみに省略してしまうと，レジスタの設定が意図しない内容になってしまうため，周辺機能が自分の意図とは違う動きをすることがあります．したがって，定義済み定数はできるだけ省略しないほうがよいでしょう．

■ ライブラリ・ファイルをプロジェクトに追加する

● 標準関数ライブラリは追加の必要なし

C言語の標準関数ライブラリ(libc-coff.a)は，プロジェクトのビルド時に自動的にリンクされるので，特別な操作は不要です．

● その他のライブラリはプロジェクトの設定が必要

算術関数ライブラリ，周辺機能ライブラリ，DSP関数ライブラリについては，必要に応じてプロジェクトに追加する必要があります．本書で解説している周辺機能ライブラリを使う場合は，libp30fxxxx-coff.a(xxxxの部分はデバイスの型名）を追加します．各

表1　各ライブラリのファイル名とヘッダ・ファイル名

ライブラリ種別	ライブラリ・ファイル名	対応するヘッダ・ファイル
C言語標準関数ライブラリ	libc-coff.a(*1)	(*2)
算術関数ライブラリ	libm-coff.a	math.h
周辺機能ライブラリ	libp30fxxxx-coff.a	(*2)
DSP関数ライブラリ	libdsp-coff.a	dsp.h

*1：自動的にリンクされるのでユーザが明示的に追加する必要はない
*2：ヘッダ・ファイルは機能別に用意されている

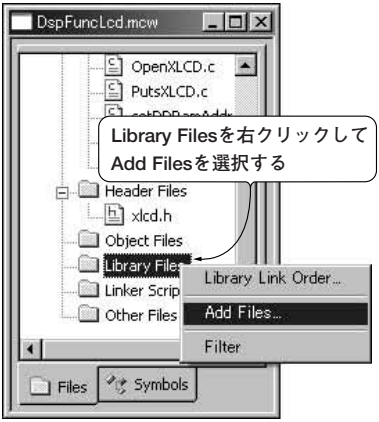

図1　ライブラリの追加方法

ライブラリのファイル名を**表1**に示します．

ライブラリの追加は，MPLAB IDEのプロジェクト・ウィンドウから行います．**図1**のように，プロジェクト・ウィンドウの［Library Files］を右クリックしてメニューを出し，［Add Files...］を選択します．するとファイル選択のダイアログが表示されます．

ライブラリは，通常は“C:¥Program Files¥Microchip¥MPLAB C30¥lib¥”に入っていますから，このフォルダに移動して使用するライブラリを選択します．

◆参考・引用*文献◆

(1) MPLAB C30 Cコンパイラ ユーザーズガイド，DS70046E_JP，2007年，マイクロチップ・テクノロジー．
(2) 16‐Bit Language Tools Libraries，DS51456D，2007，Microchip Technology Inc.
(3)* MPLAB C30 16‐bit Peripheral Libraries，2007，Microchip Technology Inc.