

マグネット・アクチュエータ・ドライバとスピード・コントローラを一つの PIC に内 蔵した 3 チャネル赤外線受信機を紹介します.マグネット・アクチュエータもスピー ド・コントローラと同様, PWM 信号でコントロールします.

オリジナルのプログラムはイギリスの Andy Birkett 氏によって, C言語で書かれたもので, PIC12C508A(509より少しメモリの容量が少ない品種)用です.

オリジナルのプログラムをそのままコンパイルしても日本メーカの送信機の信号タイミングにマッチしないことと、本書で紹介している受信機とPICの出力ピンの割り当てに 違いがあるため、筆者が国内3社の送信機で使えるように、また本書で紹介しているほかの受信機と互換性が保てるようにプログラムの一部を変更しました.

7.1 C言語とCC5X Cコンパイラ

オリジナルのソースは, CC5X C Compilerを使ってコンパイルすることができます.CC5X C CompilerはノルウェのB Knudsen Data社のコンパイラで, CC5Xのフリー・エディションが提供されています.このバージョンを使って1Kワードまでのコードをコンパイルすることができるので,インドア・プレーンで使用する受信機のプログラムを組むのには十分です.

比較的頻繁にバージョン・アップされるので,最新版のCC5X C Compiler FreeをB Knudsen Data社のホーム・ページからダウンロードしてください.マニュアルもダウンロードすることができ ます.

http://www.bknd.com/cc5x/index.shtml

このコンパイラはMPLAB version 6.30 以降で, MPLAB統合環境の中で使用することができます.アセンブラのMPASMを使うのとまったく変わらない使い方で,Cコンパイラを使っていることを意識させません.

ダウンロードしたら解凍してください.解凍したファイルの中のINSTALL.TXTに使い方が書いてありますが難しいことはありません.これから説明するとおりに設定すれば使えるようになります.

CC5X コンパイラは専用のインストーラがなく,適当なフォルダを作って,その中にすべてのファイ ルをコピーして使うようになっています.アンインストールするときは,作ったフォルダを削除するだ けです.通常はC:¥Program Filesの下にcc5xとでもして,cc5xフォルダに解凍したすべての ファイルをコピーします.作業用のフォルダをC:¥Program Files¥cc5xの下にprojects とでもして作っておくとよいでしょう.これから先の説明は上記の設定で進めて行きます.MPLABの インストールについては,付録Dの「初めてのPICプログラミング」で詳しく説明しています.

準備として,付属CD-ROMのled-test.cというC言語のソース・ファイルを先ほど作った

MPLAB IDE v6.40	- 🗆 🗙
<u>Eile E</u> dit <u>V</u> iew <u>Project</u> <u></u>	ebugger Programmer Tools Configure Window Help
] D 📽 🖬 % 🐂 🖷	- ° 🖉 🖬 🖡
🔲 Untitled Wor 🗌 🗖	
New Project	×
Project Name	
led-test	
Project Directory	
C:\Program Files	vcc5x\projects
Help	0K Cancel
	PIC12F629

図7.1 プロジェクト名とプロジェクト・ディレクトリを指定

projectsフォルダにコピーしておいてください.このソース・ファイルは,付録Dの「初めてのPIC プログラミング」で取り上げている,アセンブラで書かれた発光ダイオードの点滅プログラムをC言語 で書きなおしたものです.興味のある人はソース・プログラムを比較してみてください.

CC5X コンパイラをMPLABで使う

MPLABでCC5Xを使えるようにするための設定がいくつかあります.まずMPLABを立ち上げます.

図7.1は、led-testというプロジェクト名を新たに作って、C:¥Program Files¥cc5x ¥projectsディレクトリで作業を行う設定をするところです.ディレクトリは[Browse]ボタンで 設定します.設定が済んだら[OK]ボタンをクリックします.

次に,図7.2のように,[Configure]から[Select Device]をクリックして,ターゲット・デバイスを 選択します(図7.3).

デバイス選択がおわったら, [Configure]から[Configuration Bits]を選択してコンフィギュレーシ



図7.2 デバイス選択へ

Select Device	\mathbf{X}
De <u>v</u> ice:	
PIC12F629	•
_Microchip Programme	r Tool Support
O PICSTART Plus	MPLAB ICD 2
PRO MATE II	O PICkit 1
MPLAB PM3	
MPLAB SIM MPLAB SIM30	MPLAB ICD 2
MPLAB ICE 2000	MPLAB ICE 4000
PCM12XB0	No Module
<u>ok</u>	<u>Cancel H</u> elp

図7.3 ターゲット・デバイスの選択

ョン・ビットの設定を行いますが,今回コンパイルするled-test.cのソース・プログラムには, すでにコンフィギュレーション・ビットの設定を書き込んであるので,ここでの設定は省略できます.

ここで,ひとまずソース・ファイルの内容を説明しておきましょう.**リスト**7.1は,PIC12F629の ためのプログラム・ソースled-test.cの全リストです.

#pragma config |= 0x2184がコンフィギュレーション・ビットの設定部分です.ここでは パワー・アップ・タイマだけをONにしています.この部分はあとで説明します.

では次に, [Project]から[Select Language Toolsuite]を選択します(図7.4).

図7.5に示すように,[Active Toolsuite]に[B Knudsen Data CC5X]を選択し,[Toolsuite]

リスト7.1 LEDが0.5秒点灯、0.5秒消灯を繰り返すプログラム

```
#include "hexcodes.h"
                                // インライン・アセンブラをMPASMと違和感
                                // なく使えるようにするための
                                // ヘッダ・ファイル
//#include "12f629.h"
                                // MPLAB でコンパイルする場合は記述不要
                                // コンフィギュレーション・ビットの設定
#pragma config |= 0x2184
void delay(char millisec)
                                // ウェイト・ルーチン
{
                                // delay timeの調整はMPLAB SIMで行う
       uns8
              i;
                                // 8bit unsigned(MIN 0, MAX 255)
       do {
              TMR0 = 0;
              while (TMR0 < 127)
       } while (-- millisec > 0);
       for (i = 0; i < 178; i++)
void main(void)
                                // メイン
                                // 初期設定
       #pragma update_RP 1
                                // Bank1に切り替え
       #asm
                                // インライン・アセンブラを使って
              DW CALL(0x3FF) // メモリの最終番地をコール
       #endasm
                                // インライン・アセンブラの終了
       OSCCAL = W;
                                // OSCCALレジスタに内部発振周波数
                                // 校正値をセット
       TRISIO = 0b.0000.0000;
                                // すべてのポートを出力に設定
       OPTION REG = 0b.0000.0011; // TMR0のプリスケーラを 1:16に設定
       CMCON = 0b.0000.0111; // コンパレータ未使用
       #pragma update RP 0
                               // Bank0に戻る
       GPIO = 0b.0000.0000;
                               // ゼロで初期化
       for(;;) {
                                // メイン・ループ
              GPIO4 = 1;
                                // GP₄をHighにしてLEDを点灯
              delay(500);
                               // 499.992ms
              nop2();
                               // cc5xではqoto $+1を生成する
              nop2();
                               // nop();を二つ記述するのと同じ
              nop2();
                               // ここまでで0.5秒経過
              GPIO4 = 0;
                               // GP₀をLowにしてLEDを消灯
              delay(500);
                               // 499.992ms
              nop2();
                                // 時間調整
              nop2();
                                // 時間調整
       }
}
```

505 M	PLAB	IDE vi	6.40							- DX
<u>F</u> ile	<u>E</u> dit	View	<u>P</u> roject	<u>D</u> ebugger	Programmer	Tools	<u>C</u> onfigure	<u>W</u> indow	<u>H</u> elp	
	ഷം	XIE	Project	t Wizard						
	ed-te ed-te Sou Hea Obj Libr	st.mc st.mcp rce File der File ect Files ary File ter Scrip	New Open Close Set Ac Quickb Clean Build	tive Project	m file)	Alt+F10 Ctrl+F1	• •			
			Build (Options			<u> </u>			
			Find in	Project File	es					
			Save F Save F Add Fi Remov	Project Project As les to Proje re File From	ct Project		,			
			Select Select Set La	Language T Version-Co nguage Too	oolsuite ntrol System I Locations					
				PIC	12F629					

図7.4 [Select Language Toolsuite]選択画面へ

📉 MPLAB IDE v6.40	×
Eile Edit View Project Debugger Programmer Tools Configure Window Help	
┃	
Eled-t Select Language Toolsuite	
- So He Active Toolsuite: B Knudsen Data CC5X	
Lit Toolsuite Contents	
CCSX C Compiler MPASM MPLINK	
Location	
LCNProgram FilesNccbxNECaX.EXE Browse	
Help OK Cancel	
PIC126629	

図7.5 CC5X C Compiler を設定

Contents]ではCC5X C Compilerを選択します.[Location]にC:¥Program Files¥cc5x.EXE と表示されるので,[OK]ボタンをクリックします.

次に,インクルード・ファイルのパスを設定します.[Project]から[Build Options]で[Project]を選択します.

図7.6のように, [Include Path, \$(INCDIR)]の右にある[Browse]ボタンでインクルード・ファイル が置かれているCC5Xフォルダを選択します.ところがこのパスにはスペースが含まれていて,この ままの設定ではパスが通りません.図7.7のように,スペースの前後をダブル・クォートでくくってあ げる必要があります.

uild Options For Project "led-test.mcp"	?>
General CC5X C Compiler MPASM MPLINK	
Output Directory, \$(BINDIR):	
	Browse,
Intermediates Directory, \$(TMPDIR):	
	Browse
Assembler Include Path, \$(AINDIR):	
	Browse
Include Path, \$(INCDIR):	
C:\Program Files\cc5x\	Browse
Library Path, \$(LIBDIR):	
	Browse
Linker-Script Path, \$(LKRDIR):	
	Browse
Help Suite Defaults	
OK キャンセル	道用(<u>A</u>)

図7.6 ビルド・オプション設定画面

C:\Program""Files\cc5x\ Brows	Include Path, \$(INCDIR):	
	C:\Program''''Files\cc5x\	Browse.

図7.7 パスのスペースをダブル・クォートでくくる



図7.8 プロジェクト・ファイルにソース・ファイルを追加

[適用]ボタンをクリックしてから[OK]ボタンをクリックして,ウィンドウを閉じます.

これでいつでもCC5X コンパイラをMPLABで使うことができます.早速led-test.cのソース・ プログラムを読み込んで,コンパイルしてみましょう.

ここからの手順は,アセンブラのソース・プログラムを読み込むときと同じです(図7.8). C言語で書かれたソース・プログラムを表示したものを図7.9に示します.[Project]から[Build]を選