Int. J. Microgravity Sci. Appl. Vol. 30 No. 2 2013 (120–130)

IIIIII 技術解説講座 IIIIII

航空機実験用加速度スイッチの作り方（第二回）
初心者のためのマイコン電子工作講座

夏井坂　誠

How to Make an Acceleration Switch for Parabolic Flight Experiments (No. 2)
A lecture for a Beginner to Make a Microcomputer-Controlled, Electronic Device

Makoto NATSUISAKA

Abstract

The lecture introduces how to make a microcomputer-controlled, electronic device for a beginner. A series of lectures provides not only how to measure a physical property with an electronic sensor, convert it to a digit (analog digital conversion), switch on and off an electronic circuit with FET (Field Effect Transistor), and control those with a microcomputer but also practical know-how to design an actual electronic circuit, choose appropriate electronic parts, and mount those to a PCB (printed-circuit board), with explaining how to make "an acceleration switch". The switch can automatically turn on and off a connected device according to an acceleration level measured with an acceleration sensor and contribute to parabolic flight experiments through size reduction of an apparatus, less operation, and precise control of the experiments.

Keyword(s): acceleration, microcomputer, sensor, FET, parabolic flight

1. はじめに

前回は FET を使って電子的に電気回路をスイッチングする方法を勉強しました．さて，今回はいよいよマイコンの使い方を学習します．今さらマイコンの説明は不要かもしれませんが，最初に少しだけ概要を説明します．マイコンは，Microcomputer または Microcontroller の略ということらしいのですが，Microcomputer という名前が示す通り，まずは計算ができます．計算を行うプロセッサとデータを保持するメモリから構成されます．ただ，これだけでは単なる計算器ということになってしまいます．マイコンが，携帯電話，家電，自動車など，身の回りのあらゆるものに使われ，これだけ普及している最大の理由は，もう一つの名前 Microcontroller が示す通り，周辺装置または周辺回路の制御を得意としている点にあります．そのため，プロセッサ＋メモリ以外にも，周辺装置または周辺回路と信号のやり取りを行うためのアナログ入力回路，デジタル入出力回路，A/D（Analog/Digital）変換回路，アウトプットコンペア機能，PWM（Pulse Width Modulation）出力回路などを有し，さらにはクロックを生成する発振回路，外部制御装置（他のマイコンや PC）や周辺装置（表示器や外部メモリなど）との通信回路も有しています．これらインターフェース回路を内蔵しているおかげで，たった一つのマイコンで，計測データの読み取り，条件判断と出力制御，データの表示と保持，さらには他の装置や PC との通信が可能となっています．

マイコンは各社から様々なものが販売されていて，選択に迷いますが，素子性能（CPU クロックの周波数，処理系のデータ幅など），内蔵している周辺回路（「ペリフェラル」と呼ばれる）の機能・性能（入出力ピン数，A/D 変換能力，通信機能，DSP（Digital Signal Processing）機能の有無など），価格，さらには開発環境の入手性＆値段，入門書の有無などを考慮して選びます．本講座では，マイクロチップ・テクノロジー社の dsPIC30F2012 を使用することにします．その理由は，十分な素子性能，周辺回路性能を有していること，開発環境が無償提供されていること，使いやすい基板が適当な値段で入手可能で

宇宙航空研究開発機構　宇宙科学研究所　ISS 科学プロジェクト室　〒305-8505　茨城県つくば市千現 2-1-1
Japan Aerospace Exploration Agency (JAXA), ISS Science Project Office, Institute of Space and Astronautical Science (ISAS), 2-1-1 Sengen, Tsukuba, Ibaraki 305-8505, Japan
(E-mail: natsuisaka.makoto@jaxa.jp)

あることに加えて，親切な解説書（トランジスタ技術 2007 年 8 月号，9 月号）があるためです．というよりも，正直に告白してしまえば，トランジスタ技術 2007 年 8 月号，9 月号があったから，dsPIC30F2012 を選びました．これらの記事では，dsPIC30F2012 の基本的な使い方（開発環境の導入と C 言語によるプログラミング）から，デジタル I/O の使い方，表示モジュールや液晶モジュールとの接続，A/D 変換，PC との通信，音声入出力，DSP 機能を使ったデジタルフィルタの組み方などを丁寧かつ要領良く説明しており，うんざりするユーザ・ガイドの類を読まなくても，dsPIC30F2012 の高機能を使い始めることが可能となっています．本講座もこれをもとにしていますので，まずはバックナンバを購入して下さい．付録の dsPIC30F2012 基板と，付録 CD-ROM に収録されたソフトウェアを利用してプログラムの書き込みを行います．（なお，付録基板はマルツパーツ館からトランジスタ技術連携キット「MDSPIC2012」として単品販売されています．）

2. dsPIC30F2012

2.1 dsPIC30F2012 の概要

本講座で使用する dsPIC30F2012 は，マイクロチップ・テクノロジー社の 16 ビットマイコンになります．電子工作好きの人に愛用されている PIC シリーズの上位バージョンということになりますが，dsPIC シリーズの最大の特徴は，頭 3 文字「dsP」から連想されるように DSP 機能を有している点にあります．高速演算処理が必要な場合，通常のマイコンでは，DSP ユニットを外付けしますが，dsPIC の場合，DSP 機能（高速演算回路と命令セット）を内蔵しており，単体で，デジタルフィルタを構築したり，FFT 処理を行ったり，PID 制御を行うことが可能となっています．その他，I/O ポート，タイマー，インプットキャプチャモジュール，アウトプットコンペアモジュール，直交エンコーダインターフェース，10 ビットA/D コンバータ，12 ビット A/D コンバータ，UART モジュール，SPI モジュール，I²C モジュール，データコンバーターインターフェース（IDC）モジュール，CAN モジュールといった機能・回路も内蔵しており，これ一台でかなりのことを実現できます．

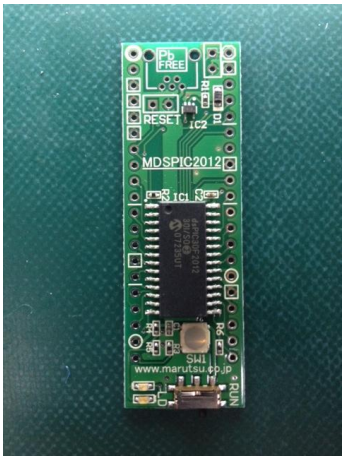

Fig. 1　MDSPIC2012

2.2 dsPIC30F2012 搭載基板 MDSPIC2012

MDSPIC2012（**Fig. 1**）は，マルツパーツ館がトランジスタ技術連携キットとして売り出している dsPIC30F2012 搭載基板です．丸ピン・ヘッダなどを取り付けることによって，DIP（Dual Inline Package）モジュールとして使うことができます．基板の左右に 20 個づつ穴があいているのがわかると思いますが，左上から反時計周りに 1, 2, 3, ... , 40 とピン番号が割振られています．（dsPIC30F2012 自体は 28 ピン構成なので，dsPIC30F2012 自体のピン番号と異なります．以下，ピン番号を使用するときは，MDSPIC2012 のピン番号で呼びますので，お間違え無く．）これらのピンには，**Table 1** に示す通り，様々な役割が割り当てられています．一つのピンに対して，スラッシュ「/」区切りで列記されている場合は，複数の機能が割り当てられていることなります．ちなみに，AN0～AN9 はアナログ入力を，RB0～RB9，RC13～RC14，RD8～RD9，RF2～RF6 はデジタル出力を表しており，AN0～AN9 に対しては A/D 変換を行うことが可能です．各ピンの機能はプログラム上で選択，設定します．

Table 1　Pin assignment of MDSPIC2012

ピン(※)	機能	解説
1	VDD	AGND：アナログ用グランド
2	NC	AN：アナログ入力
3	GND	AVDD：アナログ用電源
4	GND	CLK：外部クロック入出力
5	GND	CN：状態変化通知
6	GND	GND：グランド
7	MCLR	IC：入力キャプチャ
8	EMUD3/AN0/VREF+/CN2/RB0	INT：割り込み入力
9	EMUC3/AN1/VREF-/CN3/RB1	LVDIN：低電圧検出用入力
10	AN2/SS1/LVDIN/CN4/RB2	MCLR：リセットピン
11	AN3/CN5/RB3	OC：出力比較
12	AN4/CN6/RB4	OCFA：フォールト・プロテクション入力
13	AN5/CN7/RB5	OSC：主発振器
14	GND1	PGC/EMUC、PGD/EMUD：ICD-3プログラミング＆デバッギング、I2C、SPI、UART通信
15	OSC1/CLKI	PROG_DATA：プログラミング
16	OSC2/CLKO/RC15	RB、RC、RD、RF：デジタル入出力
17	EMUD1/SOSCI/T2CK/U1ATX/CN1/RC13	SCL、SDA：I2C通信
18	EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14	SDI、SDO、SCK、SS：SPI通信
19	VDD1	SOSC：副発振器
20	IC2/INT2/RD9	T1CK、T2CK：タイマ・クロック入力
21	EMUC2/IC1/INT1/RD8	U1ARX、U1ATX：代替UART入出力
22	SCK1/INT0/RF6	U1RX、U1TX：UART入出力
23	PGD/EMUD/U1TX/SDO1/SCL/RF3	VDD：電源
24	PGC/EMUC/U1RX/SDI1/SDA/RF2	VREF：基準電圧入力
25	GND2	
26	VDD2	
27	CN18/RF5	
28	CN17/RF4	
29	AN9/OC2/RB9	
30	AN8/OC1/RB8	
31	EMUD2/AN7/RB7	
32	AN6/OCFA/RB6	
33	AGND	
34	AVDD	
35	SCK	
36	SDO	
37	SDI	
38	SS	
39	GND	
40	PROG_DATA	

※ピン番号はMDSPIC2012のもの（dsPIC30F2012のものと違うことに注意）

2.3　マイコンのプログラミング

マイコンに所定の動作をさせるためには，プログラミングを行います．マイコンのプログラミングでは，通常，アセンブリ言語または C 言語が使われます．メモリ容量の少ない一昔前のマイコンでは，よりシンプルなアセンブリ言語が使われることが多かったのですが，マイコンの高性能化に伴い，より複雑な制御が可能となり，最近では C 言語でもプログラミング可能なマイコンが大半を占めるようになっています．アセンブリ言語は，機械語と一対一に対応しているために，マイコンの内部動作を見通しやすく，動作タイミングの見積もりや正確な設定がしやすいといった長所がある一方，C 言語などの高級言語に比べると命令セットが少ないため，複雑な制御を行う場合，長々とした記述が必要になるといった欠点があります．また，機種依存性があるために，一度組んだプログラム（リソース）の利活用が効かないなどの欠点もあります．これに対して C 言語は，より多くの命令セットを有し，複雑な制御を，よりシンプルに記述することが可能となっています．また，機種依存性のある部分をコンパイラ任せにできるため，プログラマは制御の内容に集中してプログラムを書くことが可能となっています．本講座は，あくまでもマイコンを道具として使用するというスタンスで進めますので，プログラミングは C 言語で行います．（C 言語というと「ポインタが・・・」と拒否感を示す人がいますが，本講座のレベルでポインタを使うことはありませんので，あまり難しく考えず，まずは挑戦してみて下さい．C 言語に関しては良書が多数出版されていますので，一番簡単な入門書を購入して，そちらをご一読下さい．）

dsPIC のプログラミングの流れを **Fig. 2** に示します．

1) MPLAB X IDE（マイクロチップ・テクノロジー社が提供する統合開発環境）上で，プロジェクトを作成．
2) プロジェクトにリンカ・スクリプト・ファイル等関連ファイルを追加．
3) プロジェクトにソース・ファイル（.c ファイル）を追加して，プログラムを書き込む．
4) ビルド（ソース・ファイルと関連ファイルから 16 進数（HEX）ファイル（.hex ファイル）を作成すること）．
5) MDSPIC2012 と PC を接続（シリアル-USB 変換ケーブルを使用）．
6) 書き込みソフト（dspicguy.exe）を用いて，プログラムを MDSPIC2012 にダウンロード．

1)~4)で使用するソフトウェア MPLAB X IDE と MPLAB XC16 C Compiler は，マイクロチップ・テクノロジー社のウェブ・ページから無償版を入手可能です．トランジスタ技術の付録 CD-ROM には，一世代前の MPLAB IDE と MPLAB C30 C Compiler が集録されていますが，これは Windows XP SP2 までしか対応していませんので，本講座では最新版の MPLAB X IDE と MPLAB XC16 C Compiler を使用することにします．MPLAB X IDE は，Windows 2000~7 だけでなく Linux, Mac OS にも対応しています．なお，通常，5)，6)の書き込み作業には，書き込み器（マイクロチップ・テクノロジー社の ICD-3 や PICkit3 など）が必要となりますが，MDSPIC2012 にはブート・ローダという書き込みプログラムが搭載されていて，書き込み器を必要としません．

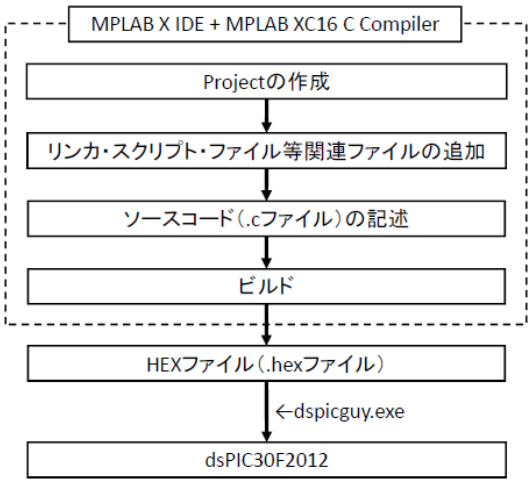

Fig. 2 Work flow of the dsPIC30F2012 programming

3.　試してみよう

3.1　プログラミングの準備

では，実際にマイコンを動かしてみましょう．まずは，以下の手順で，プログラミング環境を整えて下さい．

【準備】統合開発環境と C コンパイラのインストール

dsPIC30F2012 のプログラムを作成するには，まずはマイクロチップ・テクノロジー社のウェブ・サイトから統合開発環境 MPLAB X IDE をダウンロードして，パソコンにインストールする必要があります．さらに，C 言語でプログラムできるように C コンパイラ MPLAB XC16 C Compiler をインストールする必要があります．どちらも無償版がありますので，以下の手順でインストールして下さい．インストール作業に先立ち，一点注意しておきたいことがあります．関係するフォルダ名に日本語が使われていると，うまく動かなくなる可能性があります．自分は MPLAB IDE を使用していたときにユーザ名に日本語（2 バイト文字）を使用していたために，痛い目に遭いました．（関連ファイルまたはフォルダを My Documents などに置いた場合，ディレクトリ名にユーザ名が含まれるので，MPLAB IDE 側で認識できなくなる．）ユーザ名，フォルダ名などには英語（半角英数字）を使って下さい．また，インストール・ディレクトリは，なるべくデフォルトのものを使って下さい．

それでは，以下の手順に則って，インストール作業を行って下さい．（以下，筆者の使用している Windows XP

版での説明になりますが，Windows 7 でもほぼ同様の手順となります.）

1) マイクロチップ・テクノロジー社のウェブ・サイト「http://www.microchip.co.jp/」を開く.

2) ページ下部の製品一覧から「>開発ツール」，「MPLAB X IDE」をクリック，ページ左の MPLAB X Links から「MPLAB X FREE DOWNLOAD」をクリック.

3) 使用している OS にあった，「MPLAB X IDE v1.70」（執筆時最新版）をクリック＆保存.

4) ダウンロードフォルダを開けて，「MPLABX-v1_70-windows-installer.exe」をダブルクリック.

5) Setup - MPLAB X IDE v1.70 画面は，そのまま「Next >」.

6) License Agreement 画面は，「I accept the agreement」にチェックのうえ，「Next >」.

7) Installation Directory 画面は，デフォルト（C:¥Program Files¥Microchip¥MPLABX）のまま，「Next >」.

8) Ready to Install 画面は，そのまま「Next >」．インストールが実行される.

9) Completing the MPLAB X IDE v1.70 Setup Wizard 画面では，下部のチェックボックスにチェックが入っていることを確認のうえ，「Finish」.

10) すると，MPLAB XC Compilers 画面（ http://www.microchip.com/pagehandler/en_us/devtools/mplabxc/）が開くので，左下方にあるツリーから Downloads - XC16 の使用している OS バージョンをクリック＆保存.（本講座で使用する dsPIC 用の C コンパイラは XC16.）

11) ダウンロードされた C コンパイラ・インストーラ（xc16-v1.11-windows-installer.exe）をダブルクリック＆実行.

12) Setup - MPLAB XC16 C Compiler 画面は，「Next >」.

13) License Agreement 画面では，I accept the agreement にチェックのうえ，「Next >」.

14) Choose Installer 画面では，Install compiler にチェックのうえ，「Next >」.

15) Installation type 画面は，Install MPLAB XC16 C Compiler on this computer にチェックのうえ，「Next >」.

16) License Activation Key 画面の入力ボックスは，ブランクにしたまま「Next >」.（ブランクにしたままにすることによって，無償版としてインストールされる.）

17) No activation key entered の確認画面が表示されるので，「はい」.

18) Install Free or Evaluation Compiler 画面は，Run the compiler in Free mode にチェックのうえ，「Next >」.

19) Installation Directory は，デフォルト（C:¥Program Files¥Microchip¥xc16¥v1.11）のままで，「Next >」.

20) Ready to Install 画面は，そのまま「Next >」．インストールが実行される.

21) Completing the MPLAB XC16 C Compiler Setup Wizard 画面は，そのまま「Finish」.

このまま，プログラミングに取り掛かっても良いのですが，MPLAB X IDE の設定を一部手直ししておきます.

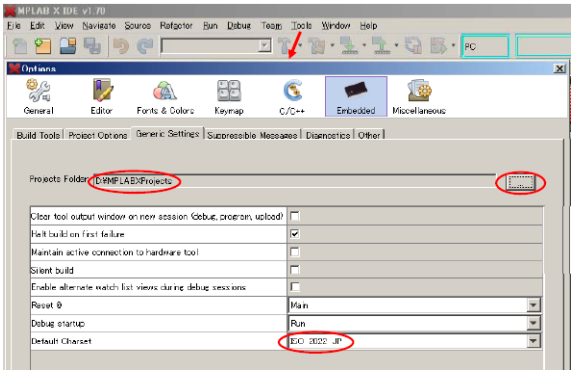

Fig. 3　Settings of MPLAB X IDE

22) MPLAB X IDE を起動して，Tools>Options>Embedded>Generic Settings．Projects Folder の右の「...」を押す.（**Fig. 3**）筆者の場合，デフォルトで，「C:¥Documents and Settings¥ユーザ名¥MPLABXProjects」に設定されていたが，フォルダ名，ファイル名，パスなどに空白を含むと，後で書き込みソフトを使ったときに，書き込みエラーが起こるので，空白を含まないパスに変更した．筆者の場合，データは D:ドライブに置くことにしているので，フォルダ「MPLABXProjects」をデフォルトの位置から D:ドライブのルートに移動した上で，Please select the Project Folder 画面で，ファイル名が「D:¥MPLABXProjects」となるようにして「保存」．空白および日本語を含まなければどこでも良い．（Windows 7 の場合，デフォルトで，「C:¥Users¥ユーザ名¥MPLABXProjects」となるので，ユーザ名が半角英数字であれば，本操作は不要.）さらに，「Default Charset」で「ISO-2022-JP」を選んで，「OK」.（**Fig. 3.** 使用している OS に合わせて適当な日本語文字セットを選ぶ．ここが適切に設定されていないと，コメント文の日本語が文字化けする.）

なお，この後の説明で使用するので，MPLAB X IDE のペイン名を **Fig. 4** に示します.

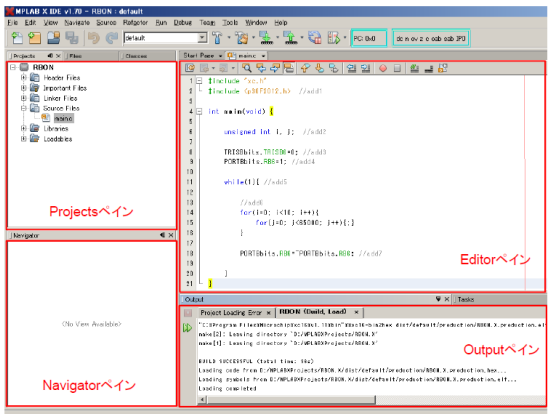

Fig. 4　Panes of the MPLAB X IDE

3.2　プログラミング

次に，プログラミングに移ります．プログラムは，**Fig. 2** に示した手順で行います．ここでは，デモ・プログラムとして，dsPIC30F2012 の RB6 ポートに LED（＋電流制限抵抗）をつないで，LED を点滅させるプログラムを作成します．まずは，以下の手順でプログラミングの準備を進めて下さい．

【手順 1】プロジェクトの作成

プログラミングの準備は以下の通り進めます．プログラムは，ユーザが C 言語で書き込むソース・ファイルの他，ヘッダ・ファイル，リンカ・スクリプト・ファイルなど，複数のファイルを必要としますので，まずはこれらをまとめて管理するための，プロジェクトを作成します．

1) MPLAB X IDE を起動．Learn & Discovery タブから「Create New Project」をクリック．または，メニューバーから「File>New Project」．（「New Project」ボタンを押しても良い．）（**Fig. 5**）（Learn & Discovery タブのある Start Page をいつも利用したい場合は，タブの下部にある Show on Startup にチェックを入れておく．また，Start Page が表示されなくなった場合は，Help>Start Page で再表示できる．）

2) Project Wizard が起動するので，Categories から「Microchip Embedded」を，Projects から「Standalone Project」を選び，「Next >」．

3) Select Device の Family から「16-bit DSCs (dsPIC30)」を，Device から「dsPIC30F2012」を選び，「Next >」．

4) Select Tool は，そのまま「Next >」．

5) Select Compiler は，「XC16 (v1.11) [C:¥Program Files¥Microchip¥xc16¥v1.11¥bin]」を選び，「Next >」．

6) Project Name を入力．本講座では，「RBON」とする．なお，3.1 項 22)で，Projects Folder を設定変更しているが，Project Location が「C:¥Documents and Settings¥ユーザ名¥MPLABXProjects」などのように，変更されておらず，空白を含んだままとなっている場合，後で書き込みソフトを使ったとき，書き込みエラーが起こるので，「D:¥MPLABXProjects」など，空白を含まないパスに変更すること．（Windows 7 の場合，デフォルトで，「C:¥Users¥ユーザ名¥MPLABXProjects」となるので，ユーザ名が半角英数字であれば，本操作は不要．）最後に，Set as main project にチェックを入れて，「Finish」．

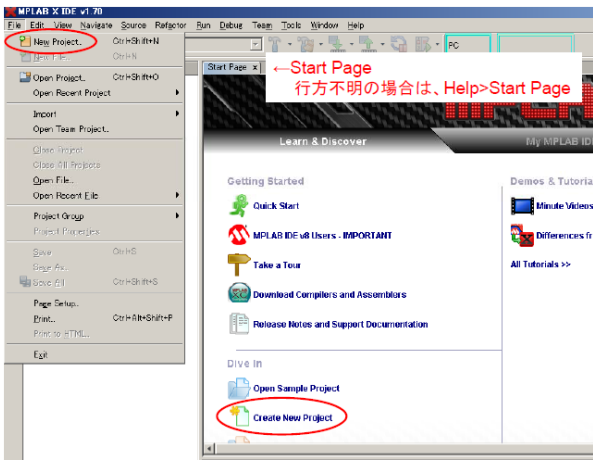

Fig. 5　How to create a new project

7) Project Wizard が終了し，MPLAB X IDE のメイン画面に戻る．左上の Projects ペイン内に Project が作成されていることを確認．

【手順 2】リンカ・スクリプト・ファイルの追加

次に，リンカ・スクリプト・ファイルを追加します．

1) Project ペイン内の Linker Files を選択した状態で，右クリック．「Add Existing Item」を選択．dsPIC30F2012 用リンカ・スクリプト・ファイル「p30F2012.gld」を選択．（デフォルト・インストールの場合，リンカ・スクリプト・ファイルは以下のフォルダに置かれている．

「C:¥ProgramFiles¥Microchip¥xc16¥v1.11¥support¥dsPIC30F¥gld」（筆者の場合，その都度，ディレクトリを変更するのが面倒臭いので，デスクトップなど，わかりやすい場所に，上記フォルダのショートカットを置いている．）さらに，Store path as: で「Auto」にチェックを入れて，「Select」を押すと，Project ペイン内の Linker Files の下に，p30F2012.gld が追加される．

【手順 3】ソース・ファイルの作成

次に，ソース・ファイルを作成，追加します．

1) Project ペインに表示されている Source Files フォルダを選択して，右クリック，「New」から「mainXC16.c」を選択．Name and Location で File Name を付ける．（ここでは「main」とする．）そして「Finish」．または，Source Files フォルダを選択した状態で，メニューバーから「File>New File」．（「New File」ボタンを押しても良い．）File Wizard が起動したら，Choose File Type の Categories から「Microchip Embedded – XC16 Compiler」を，File Types から「mainXC16.c」を選択して，「Next >」．Name and Location で File Name を付ける．（ここでは「main」とする．）そして「Finish」．

Project ペインで，作成した main.c をダブルクリックしてください．すると Editor ペイン（**Fig. 4**）に MPLAB XC16 C Compiler のエディット画面が現れます．そちらに，下記のようなテンプレートが表示されるはずです．

```
/*
* File:   main.c
* Author: ユーザ名
*
* Created on 作成した日時
*/

#include "xc.h"
```

```
int main(void) {
    return 0;
}
```

「/*」と「*/」は，これらに挟まれた部分がコメント文であることを示しています．コメント文は，プログラマがプログラムを見やすくするために加える注釈で，コンパイル時は無視されるので，マイコンの動作には全く影響を与えません．また，

```
// ・・・
```

というコメントの付け方もあります．//以降，そのエディタ行内に書かれた内容がコメントであることを示しています．プログラムを後で読み返したり，修正したりするときのために，なるべくコメントは書き加えておきましょう．（その他，修正を加えるときに，一気にプログラムを書き換えると，やっぱり元に戻したいとなったときに困るので，修正を加える箇所を削除するのではなくて，コメント化しておいて，必要な修正作業を加えたりします．）

次に書かれている「#include "xc.h"」は，ヘッダ・ファイルになります．C 言語のプログラミングでは，他のプログラムでも使用されるような汎用性の高い記述は，ヘッダ・ファイルとして別に作成しておき（上記の場合「xc.h」），これを使用するプログラムでは，冒頭に「#include “ヘッダ・ファイル名”」または「#include <ヘッダ・ファイル名>」と読み込み指示を 1 行書きこむだけにします．これにより，メイン・プログラムがすっきりと見通しが良くなりますし，プログラムを組むたびに，同じ内容をくどくど書く必要が無くなります．

```
int main(void) {
    return 0;
}
```

の部分は，引数が整数型（int）で，戻り値の無い（void）main 関数を記述しています．C 言語のプログラムは，関数の組み合わせで記述されるのですが，関数は以下の通り表現されます．

```
戻り値の型 関数名(引数の型と名前) {
・・・
}
```

引数は，ある関数が計算に使用する変数（数学関数における独立変数のようなもの）で，戻り値は，計算の結果得られた値を受け取る変数（数学関数における従属変数のようなもの）で，「{」と「}」の間に処理内容が書き込まれます．なお，C 言語では，必ず main 関数が必要で，main 関数を一番最初に実行する約束になっています．（詳しくは C 言語の教科書を読んで下さい．）

また，「;」はプログラムの区切りを表しています．C 言語では，行という概念が無いので，ひとつ命令を書いたら「;」を加えて区切りとします．（ただし「# include」や「}」の後ろには書きません．）

【手順 4】コーディング

プログラムを記述することをコーディングと言います．ここからコーディングを開始します．今回目標とするプログラムの具体的な内容は，以下の通りです．

1) dsPIC30F2012 の RB6 ポート（MDSPIC2012 のピン番号で言うと 32 ピン）に LED（＋電流制限抵抗）を接続．
2) RB6 ポートに DC5V を出力，LED を点灯．
3) 一定時間待った後，今度は RB6 をオフして，LED を消灯．
4) 2)~3)を繰り返す．

結果として，RB6 に接続された LED は点滅することになります．

まずは，Editor ペイン上で，以下の通り，追記作業を行って下さい．（適宜，スペースや空行が挟まれていますが，C 言語では空白はすべて無視されますので，適当にスペースや空行を入れても，問題ありません．むしろプログラムを見やすくするために，積極的にスペースや空行を入れた方が良いでしょう．）なお，MPLAB X IDE の Editor 上では，途中まで入力すると入力候補がプルダウン表示されます．スペルミスを避けるために，積極的に活用して下さい．

```
#include "xc.h"
#include <p30F2012.h>   //add1

int main(void) {

    unsigned int i, j;   //add2

    TRISBbits.TRISB6=0; //add3
    PORTBbits.RB6=1; //add4

    while(1){ //add5

        //add6
        for(i=0; i<10; i++){
            for(j=0; j<65000; j++){;}
        }

        PORTBbits.RB6=~PORTBbits.RB6; //add7

    }
}
```

以下，追記した部分を解説します．

add1：

「#include <p30F2012.h>」は，dsPIC30F2012 を使用するためのおまじないです．dsPIC30F2012 を使う場合は，必ず追記して下さい．

add2：
「unsigned int i, j;」は，関数の中で使用する変数を宣言しています．独自に変数を使う場合は，必ず変数宣言が必要になります．変数宣言，変数の型などについては，必ず C 言語の教科書の一番最初に書いてあるので，わからない方は，教科書をあたって下さい．
add3：
「TRISBbits.TRISB6=0;」は，RB6 ポートをデジタル出力ポートに設定しています．既述の通り，dsPIC30F2012 のピンには複数の機能が割り振られていますので，事前に機能を指定しなければなりません．
add4：
「PORTBbits.RB6=1;」は，RB6 ポートの出力をオンにしています．結果として DC5V が出力されます．「PORTBbits.RB6=0;」と書くと，出力がオフされます．
add5：
「while(条件式){・・・}」という記述は，条件式が真の場合は，ひたすら・・・を繰り返すという内容になります．ここでは，条件式が「1」となっています．条件式が「1」というのは，「真」を表しており，「while(1){・・・}」という記述は，条件式が常に真ということになりますので，無限ループを表します．
add6：
「for (i=N_0; i<N_m; i++){・・・}」という記述は，・・・の処理を N_m-N_0 回繰り返しなさいという命令です．i は処理回数をカウントするための変数で（add2 で符号無し整数として宣言されている），i をはじめ N_0 として，一回毎に i を 1 づつ大きくして（「i++」），i が N_m を超えない範囲で（すなわち，N_m -1 まで）処理を繰り返しなさいという意味になります．このプログラムでは処理内容が「;」なので何もしないことになります．これは，時間稼ぎに使われる表現です．すなわち，処理としては何もしないのですが，for ループを繰り返すことによって，時間を消費しますので，待ち時間を発生させていることになります．（dsPIC は非常に高速なので，for 文を加えないと，超高速で点滅を繰り返すことになり，人間の目では点滅を認識できなくなります．）プログラム中で for ループが二重になっているのは，実行回数をカウントする変数 i が符号無し整数であり，65535（処理系による）までしか表現できず，十分な待ち時間を作れないので，i で 65000 回ループさせたものを，さらに j で 10 回ループさせて，点滅を確認できる時間間隔を作っています．この二重ループのおかげで，最終的に数 100msec の待ち時間を発生させています．ここで，「i++」という表現は，「i=i+1」の略記で，これは「前の i（右辺の i）に 1 を加えたものを，次の i（左辺の i）とする」ことを表しています．プログラムにおける「i=i+1」という表現は，数学における「i=i+1」と意味が異なりますので，注意して下さい．
add7：
「PORTBbits.RB6=~PORTBbits.RB6;」は，RB6 の出力を反転させています．右辺の頭にある「~」（チルダ）は，論理を反転させるときに使われます．ここを「PORTBbits.RB6=0」と書いても良いのですが，そうするとforの二重ループがもう一つ必要になり，くどい記述となってしまいます．この書き方の方がすっきりすることをおわかりいただけるでしょうか？

なお，無限ループ「while(1){・・・}」を使ったことには，もう一つの理由があります．今回使用する MDSPIC2012 には，プログラムの書き込みを行うためのブートローダというプログラムがプリインストールされているのですが，これがプログラムメモリの最後に書き込まれていて，無限ループさせないとプログラム終了後，プログラム書き込みモードになってしまうからです．制御を繰り返す必要のない，単発処理のプログラムを書く場合も，「while(1){;}」という無限ループをプログラム末に加えて下さい．これによって，ブート・ローダの起動を回避できます．

【手順 5】ビルド

ソース・ファイルを書き上げたら，ビルドを行います．ビルドは，C 言語の文法チェックから機械語への変換，関連ファイル（ヘッダ・ファイルやライブラリ・ファイル）との結合作業（リンク）をひとまとめに行ってくれるコマンドで，最終的に 16 ビットで記述された HEX ファイル（拡張子「.hex」）が出力されます．これが dsPIC に書き込まれる最終的なファイルとなります．

では，プルダウン・メニュの「Run」から「Build Main Project」を選択して，ビルドを実行して下さい．Output ペイン（**Fig. 4**）に「BUILD SUCCESSFUL (total time: xxx ms)」と表示されれば成功です．エラーとなってしまった場合は，ソース・ファイルを修正します．スペルミスをしていないか，「;」を忘れていないか？「}」が足りなくないか？全角文字，スペースを使用していないか？などチェックしてみて下さい．修正後は，再度ビルドを行います．

3.3　プログラムの書き込み

それでは，HEX ファイルを dsPIC30F2012 に書き込んで，動作確認をしてみましょう．ただ，その前に少し準備が必要です．

【準備 1】シリアル-USB 変換ケーブルの準備

プログラムの書き込みは，MDSPIC2012 にあらかじめ書き込まれているプログラム「ブート・ローダ」を使用します．すでに説明した通り，ブート・ローダを使えば書き込み器が不要となるのですが，PC と MDSPIC2012 を接続するためのシリアル-USB 変換ケーブル（**Fig. 6**）が必要になります．シリアル-USB 変換ケーブルは，ネット検索すれば販売店がいくつも出て来ますので，自分の PC に合ったものを購入，ドライバをインストールのうえ，使えるようにしておいて下さい．

Fig. 6　Serial USB convertor cable

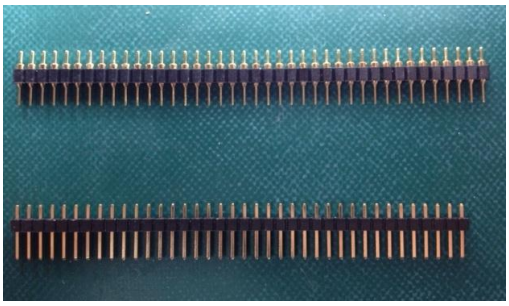

Fig. 7　Pin headers

【準備 2】書き込みソフトのインストール

もうひとつ，PC 側に書き込みソフトをインストールする必要があります．トランジスタ技術 2007 年 8 月号に付属する CD-ROM から「dspicguy.exe」，「dspicguy.bat」，「hexconv.com」，「loadspic.exe」の 4 つのファイルを PC の同じディレクトリにコピーして下さい．コピーするだけでインストール完了です．（パスに日本語名（2 バイト文字）や空白が含まれないディレクトリにコピーすること．）

【準備 3】基板の DIP 化

MDSPIC2012 は，そのままでも使えます．トランジスタ技術 2007 年 8 月号では，そのまま使う方法を紹介しています．まずはそういった使い方でも良いのですが，丸ピン・ヘッダをハンダ付けして，DIP 化してしまった方が，ブレッド・ボードに差したりできて，何かと便利ですので，丸ピン・ヘッダのハンダ付け作業を行って下さい．ピン・ヘッダの購入には，ちょっと注意が必要です．ピン・ヘッダというと普通は角型のピンのもの（**Fig. 7** 下のもの）を指すようで，今回使用する丸ピン・ヘッダを購入する際は，必ず「丸」が付いたもの（**Fig. 7** 上のもの）を選んで下さい．また，各社で呼び方が異なるようですので，ご注意下さい．（丸ピン IC 連結ソケット，丸ピンプラグなど）丸ピン・ヘッダには，色々なタイプがあるのですが，MDSPIC2012 は 40 ピンなので，一列 40 ピンのものを購入して，半分に折って使用します．参考まで，以下に筆者がよく使うものを載せておきます．

○　秋月電子通商
　丸ピン IC 連結ソケット（両端オスピン・1 列 40P）
　通販コード　P-01382

○　マルツパーツ館
　【HW309-1X40PIN】丸ピンプラグ［40 ピン×1 列］

あと，ハンダ付け用品（ハンダ・コテ，ハンダ，ハンダ・スタンドとスポンジ，ハンダ吸い取り線または除去器など）をお持ちでなければ，それらも購入して下さい．電子部品のハンダ付けには，20～30W くらいのセラミック・タイプ（昇温が速い）のもので，コテ先が細めのものが使い易いと思います．また，太めの配線や端子など，より大きな熱量を必要とするときに，熱量を切り替えられるボタンが付いたもの，また，コテ先自体の温度をコントロールできるもの（ソルダ・ステーションなどと呼ばれる）などありますので，予算に余裕があるのであれば，そちらを購入してください．ハンダは，環境を考慮して，鉛フリーのものが主流となっていますが，鉛フリーハンダは部品や基板との濡れが悪く，融点も高いため，ハンダ付けを上手に行うのが難しいので，特段要求が無ければ，鉛を 40%程度含む，すず鉛系の共晶ハンダ（Sn60-Pb40~Sn63-Pb37：融点 190°）を使用して下さい．（濡れの悪さに加えて，ウィスカの発生が懸念されるので，宇宙機では，鉛フリーハンダの使用が禁止されています．）また，金属表面から酸化被膜を除去して，ハンダの濡れを良くするために，フラックスを使用するのですが，あらかじめフラックスが添加されたハンダ（ヤニ入りハンダなどと呼ばれる）が売られていますので，これを使用してください．色々な太さのものがあるのですが，挿入実装部品などをハンダ付けするのであれば，0.8mm 程度のものが使いやすいかもしれません．（表面実装部品をハンダ付けするのであれば，もっと細いものを選んでください．）他にも，ハンダを失敗したときにハンダを除去するための，ハンダ吸い取り線（ウィッキングワイヤー），ハンダ除去器も用意しておくと良いでしょう．（加熱とハンダの吸い取りを同時にできる，数万円もするハンダ除去器も売られています．）

丸ピン・ヘッダは以下の要領で取り付けます．（ハンダ付けは練習すれば上達しますので，未経験の人は，丸ピン・ヘッダや安いユニバーサル基板を余分に買っておいて，事前に練習してから行って下さい．）

1) 丸ピン・ヘッダを半分に折る．（20 ピン×2 個にする．）

2) MDSPIC2012 の両脇の穴に丸ピン・ヘッダをしっかり差し込み（差し込む向きに注意．），テープなどでしっかり固定する．丸ピン・ヘッダが基板に垂直になっていることを確認．なお，IC ソケット（40 ピン 600MIL）（**Fig. 8** 上）を購入して，そこにあらかじめ差した状態（**Fig. 8** 下の状態）でハンダ付けすれば，丸ピン・ヘッダを基板に垂直に取り付けることが可能です．この後，このままブ

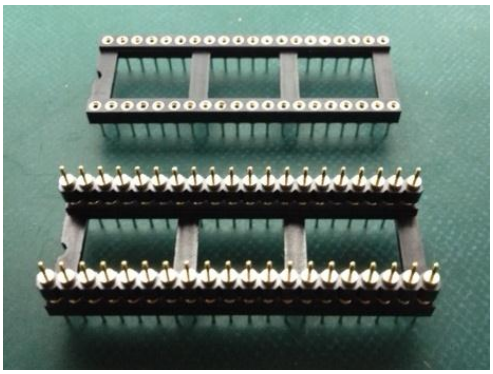

Fig. 8　IC socket (40pin, 600MIL) & pin header

Fig. 9　Dsub9 connector

レッド・ボードに差して動作確認を行いますので，IC ソケットは外さず，そのままの状態にしておいて下さい．（どうしても，外したいというときは，マイナスの精密ドライバなどを使って，ちょっとずつ慎重にはずして下さい．丸ピンは弱いので，簡単に曲がったり，折れたりします．）

なお，ハンダ付けは以下の要領で行って下さい．

1) 基板のランド（基板穴の周りの金属面）とピンをアルコールで洗浄．コテ先を水でぬらしたスポンジで洗浄．

2) ハンダ付けは，ランドとピンの両方にコテ先を当てて予熱，十分温まったらハンダを供給してコテ先を離す．ハンダの供給量は，ハンダ面が富士山のようにやや凹んだ円錐形になるくらいの量で，表面に金属光沢が残っていなければならない．（表面が濁ってしまうのは，加熱のし過ぎ．）ハンダが濡れ広がっていかないときは加熱不十分なので，部品を壊さない程度に良く加熱する．

3) 最後に，アルコールなどで，ハンダ面の周りのヤニをふき取る．

【準備 4】Dsub9 ピン・ケーブルの作製

シリアル-USB 変換ケーブルと MDSPIC2012 を接続するための，Dsub9 ピン・ケーブルを自作します．Dsub9 ピン・コネクタのメスを購入して下さい．Dsub9 ピンの 3 番ピンと 5 番ピンにケーブルを取り付けて下さい．3 番ピンがデータ線，5 番ピンがグランドになりますので，色つきのケーブルを使って，識別できるようにして下さい．

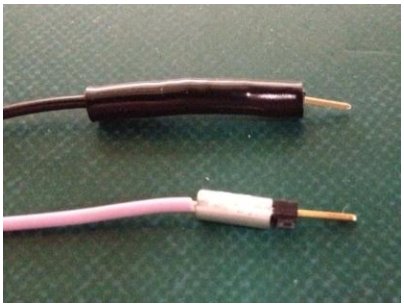

Fig. 10　Connecting sleeves

（番号はピンの横に小さく書かれています．）ハンダ付け後は，接合部を熱収縮チューブで覆って下さい．（**Fig. 9**）ケーブルの反対側は，ブレッド・ボードに差し込みますので，0.6mm くらいのすずメッキ線，または，角ピン・ヘッダなどを取り付けます．筆者は，連結スリーブ（または接続スリーブ）というチューブ状の圧着端子（**Fig. 10** 下）で，角ピン・ヘッダとケーブルをかしめ，最後に熱収縮チューブで覆いました．（**Fig. 10** 上）連結スリーブや角ピン・ヘッダを持っていなければ，すずめっき線のような単線（ブレッド・ボードに挿せるもの）とケーブルを直接ハンダ付けしても OK です．

【準備 5】ブレッド・ボードへの組み込み

Fig. 11 の通りデモ回路をブレッド・ボードに組み込みます．なお，組み込み時は電源をオフにしておいて下さい．

まず，MDSPIC2012 をブレッド・ボードに取り付けて下さい．MDSPIC2012 は IC ソケットに差したままの状態で差し込みます．MDSPIC2012 を差したら，シリアル-USB 変換ケーブルにつながれた Dsub9 ピン・コネクタの 3 番ピンと MDSPIC2012 の PROG_DATA（40 番ピン）とを，Dsub9 ピン・コネクタの 5 番ピンと MDSPIC2012 のグランド（39 番ピン）を接続して下さい．さらに，

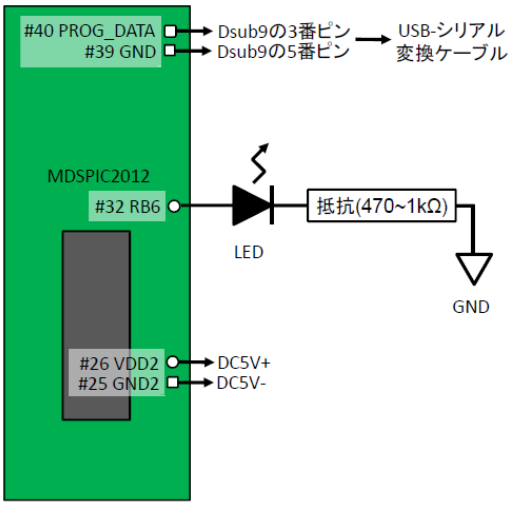

Fig. 11　Wiring for the download of the demo program

DC5V（+）を MDSPIC2012 の VDD（26 番ピン）に，DC5V のリターン（－）を MDSPIC2012 のグランド（25 番ピン）に接続して下さい．LED（+電流制限抵抗）は，MDSPIC2012 の 32 番ピン（RB6）につないで下さい．LED の電流制限抵抗は，200Ω より大きなものを使うようにして下さい．dsPIC30F2012 が流せる，または吸い込める電流は 1 ピンあたり 25mA，全ピン合計で 200mA までなので，あまり小さな抵抗を使うと破損につながります．実際は，5~10mA も流れれば十分かと思いますので，470Ωから 1kΩのものを使って下さい．（**Fig. 12**）

【手順 1】プログラムの書き込み

さて，配線が終わったらいよいよ書き込みを行います．以下の手順で進めて下さい．

1) MDSPIC2012 のスライド・スイッチ SW2 を「LD」（LOAD）にセット．

2) 電源オン．MDSPIC2012 上の赤 LED が点灯．

3) プッシュ・スイッチ SW1 を押す．（ペン先など先が鋭いもので押すと，スイッチが壊れる可能性があるので注意．）赤 LED と緑 LED が同時に点灯．

4) パソコンで dspicguy.exe を起動して，「Open HEX file」ボタンを押して，書き込み対象の HEX ファイルを選択（**Fig.13**）．シリアル-USB 変換ケーブルの接続された COM ポート番号を選択．COM ポート番号の調べ方は，「マイコンピュータを右クリック，プロパティを開ける」→「ハードウェア・タブでデバイスマネージャ・ボタンを押して，ポート（COM と LPT）からシリアル-USB 変換ケーブルの COM 番号を読みとる」．COM ポート番号が 10 より大きい場合，dspicguy の COM ポート番号はプルダウンで，1~9 番しか選べないので，COM ポート番号の付け替えを行う必要がある．そのような場合は，「USB ポートの競合」をキーワードとして，ネット検索すれば，丁寧に説明してくれているサイトが見つかるので，そちらを参照のこと．また，dspicguy.exe のチェック・ボタン「Quick RUN」，「Stay on Top」，「Silent」は，各々

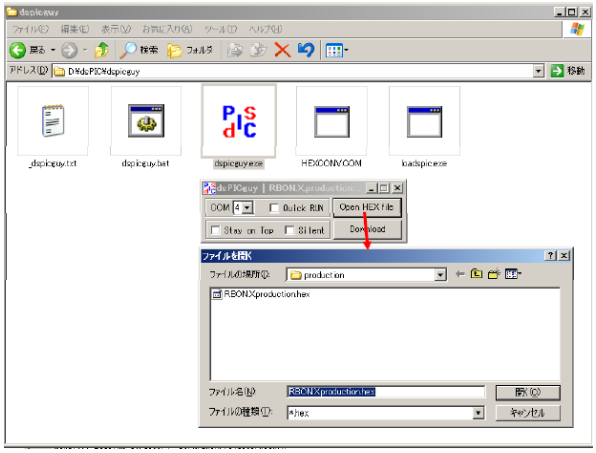

Fig. 13　How to download a HEX file

「書き込み直後からプログラムを実行」，「dspicguy を画面上に表示したままにする」，「ダウンロード終了を知らせるビープ音を鳴らさない」．

5) 「Download」で書き込み実行．ダウンロード中は DOS 窓が開く．正常終了の場合，基板上に緑 LED が点灯．

6) スライド・スイッチ SW2 を「LD」から「RUN」に切り替えるとプログラムがスタート．（Quick RUN をオンにしていた場合は，書き込み直後からスタート．）

さて，どうでしょうか？LED がチカチカ点滅してくれれば成功です（**Fig.14**）．点滅しないときは，配線を確認して下さい．それでも，ダメなときはプログラムが間違っているのかもしれません．ちなみに筆者も今回所定の動作とならずトラブル・シュートに時間を取られました．原因は，dspicguy で HEX ファイルを書き込むとき，HEX ファイルを「Document and Settings」フォルダの下に置いていたためでした．MPLAB X IDE のデフォルト・フォルダに置いただけなんですが・・・dspicguy，MPLAB X IDE，MPLAB XC16 C Compiler ともども，フォルダ名，ファイル名，パスなどに，日本語（2 バイト文字），スペースなどを嫌うことがあるようです．どうし

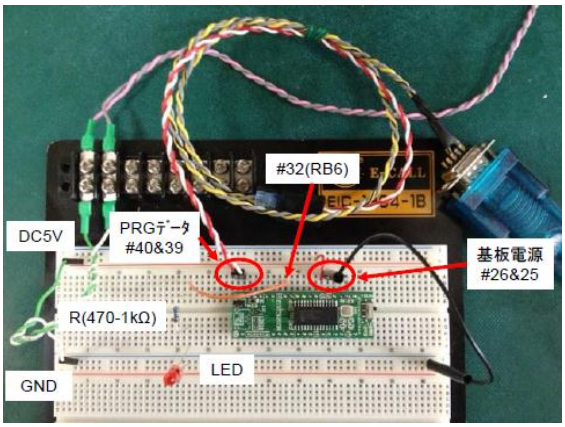

Fig. 12　Demo circuit assembled on a bread board

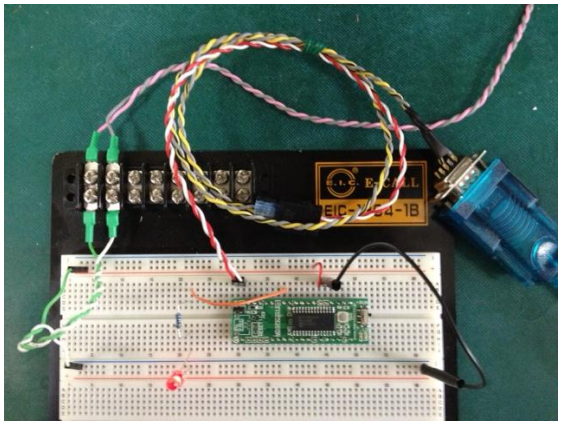

Fig. 14　Nominal action of the demo program

てもというときは，そちらも疑ってみて下さい．うまくいった場合は，プログラムに手を加えてみて下さい．for ループの i の回数を変更すれば，点滅の間隔を変えることが可能です．for ループ関連個所をコメント・アウトしてしまえば，点滅ではなくて，点灯とすることも可能です．また，RB6 以外のポートを点灯させることも可能です．RB0~9 を使うのであれば，6 の部分を 0~9 の数字で置き換えれば良いですし，RC，RD，RF などのポートを使うのであれば，B6 の部分を Cx，Dx，Fx（x はポート番号）と置き換えれば OK です．また，複数の LED を点灯させることも可能です．色々挑戦してみて下さい．

4. 結び

4.1 just do it!

今回は，マイコンの使い方を勉強しました．本当は，もっと詳しく書きたいこともあったのですが，はじめての人がなるべく興味を失わないように，余計なことは極力書かないようにしました．それでも，くどくど長い記述となってしまい，途中で挫折してしまった人もいるかもしれません．ただ，すべてを理解できなくても，ここに書かれた内容を鵜呑みにして，そのまま実行してもらえれば，確実に動作します．ですので，挫折してしまった場合は，今一度手を動かしながら，読み返して下さい．プログラミングや電子工作は，英会話と同じように，座学だけでは絶対に上達しません．実技を重ね，失敗を重ねて上達するものです．また，すべてのルールを理解しないとできないというものでもありません．まずは，動かしてみて下さい．退屈な教科書やユーザガイドを開くのは，動かしてみた上で，わからないところ，改良したいところが出てきてからで十分です．Just do it!

今回は，LED の点灯で終わってしまいましたので，次回は，A/D 変換を学習して，加速度センサの出力電圧を読みとれるようにします．

4.2 参考情報

皆さまの利便性を考えて，最後にパーツリストを載せておきます．ハンダ付けなどは大抵最初失敗しますので，予備品も含めて購入して下さい．

- MDSPIC2012
- シリアル-USB 変換ケーブル
- Dsub9 ピン・コネクタ（メス）
- ケーブル（AWG22~26 のもの）
- ワイヤ・ストリッパ
- 熱収縮チューブ
- ハンダ・コテ（20~30W のセラミック・タイプ）
- ハンダ（ヤニ入りすず鉛共晶ハンダ φ0.5~0.8 のもの）
- ハンダ・スタンドとスポンジ
- ハンダ吸い取り線または除去器
- 丸ピン・ソケット（40 ピン 600MIL）
- 丸ピン・ヘッダ（1 列 40 ピン）
- ブレッド・ボード
- LED
- 抵抗（470Ωから 1KΩ）
- 実体鏡（あれば）
- 連結スリーブと圧着工具
- DC5V 電源（安定化電源，AC アダプタなど）

販売店リストも添付しておきます．

- マルツパーツ館　http://www.marutsu.co.jp/index.php
  マルツエレック社の販売店．本講座で使用する dsPIC30F2012 モジュール基板 MDSPIC2012 や 3 軸加速度センサモジュール MM-2860（サンハヤト㈱製）の販売．店舗販売&通販.
- 秋月電子通商　http://akizukidenshi.com/
  バルク品などお値打ち品が見つかる可能性大．店舗販売&通販.
- 千石電商　https://www.sengoku.co.jp/
  バルク品などお値打ち品が見つかる可能性大．店舗販売&通販.
- RS コンポーネンツ　http://jp.rs-online.com/web/
  法人契約が必要となりますが，在庫品に関しては翌日受け取りが可能なので重宝しています．通販のみ.

参考文献

1) MPLAB X IDE ユーザガイド（DS52027A_JP）マイクロチップ・テクノロジー社
2) dsPIC30F ファミリーリファレンスマニュアル（DS70046B_JP）マイクロチップ・テクノロジー社
3) トランジスタ技術 2007 年 8，9 月号　CQ 出版

(2013 年 4 月 17 日受理)