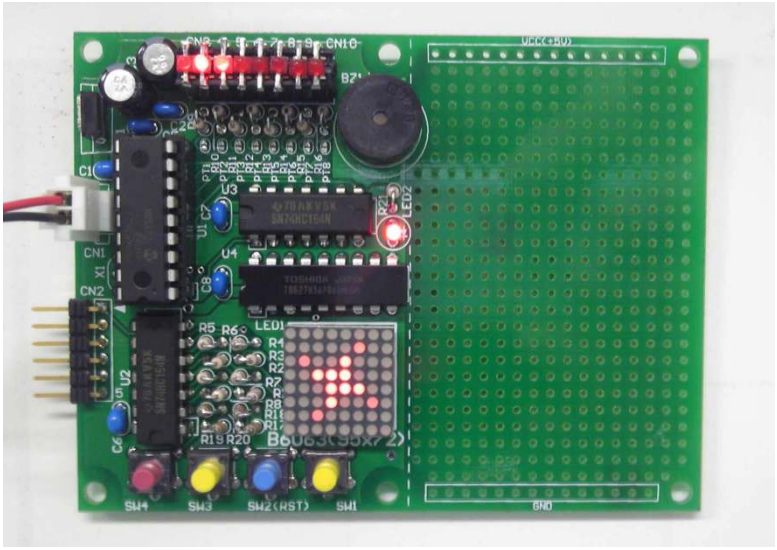
ｔｋ－ＰＩＣ１８２７

開発環境の構築手順書

(Ver.2.00)



最新版の開発環境とドキュメントを手に入れましょう

PIC の開発環境やマイクロチップテクノロジーが公開しているドキュメントは，ときどきバージョンアップされます。その際に報告されていた不具合を一緒に修正することもあります。それで，マイクロチップテクノロジーの WEB ページは定期的に確認することをおすすめします。

このマニュアルは 2012 年 12 月 14 日現在の情報に基づいています。そのため，最新版と異なるかもしれませんが，それについて弊社は免責としマニュアルの更新の義務は負いません。いつでもマイクロチップテクノロジーの最新の情報を確認してください。

Toyo-linx,Co.,Ltd.

1.「MPLAB X IDE」と「MPLAB XC8」のダウンロード

（以下は 2012 年 12 月 14 日現在の情報です。WEB ページは予告なく変更されます。）

マイクロチップテクノロジーの WEB ページからダウンロードします。検索サイト（Google, Bing, YAHOO など）から「マイクロチップテクノロジー」で検索してください。「マイクロチップ・テクノロジー・ジャパン株式会社」のアドレスが見つかるので開いてください。

開発ツールをダウンロードする前に日本語の資料を入手しておきましょう。ダウンロードのメニューから「ツール/ユーザガイド」選択します。

ダウンロードファイルの一覧から「MPLAB X IDE ユーザガイド」,「PICkit3 プログラマ/デバッガ ユーザガイド」,「PICkit3 インサーキットデバッガの使い方」をダウンロードしてください。

次に，開発ツールをダウンロードします。ページの下に「製品一覧」という項目があります。そこの「開発ツール」の欄に「MPLAB X IDE」という項目がありますのでクリックしてください。ブラウザによりますが，別タブで「MPLAB X IDE」のページが開きます（英文サイト）。

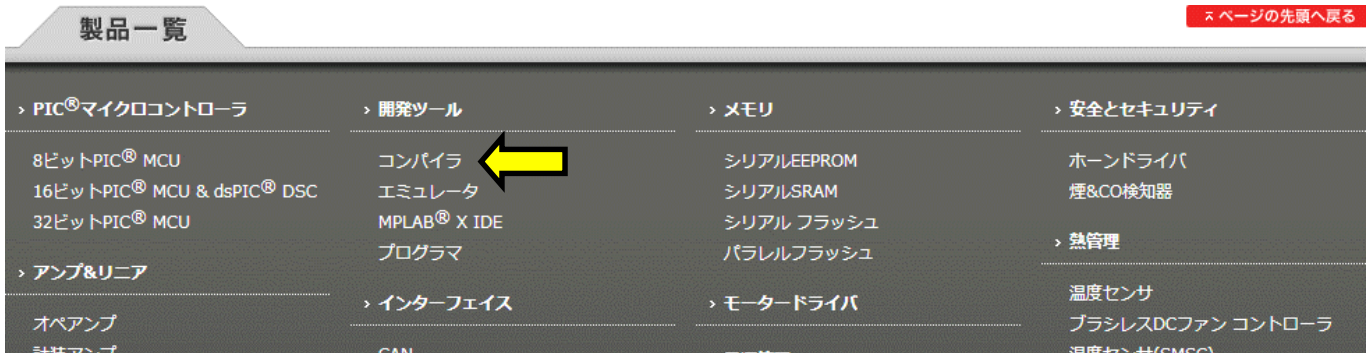
このページに「MPLAB X FREE DOWNLOAD」というボタンがあります。そこをクリックすると，ダウンロードファイルの一覧が開きます。使用するパソコンの OS に対応する「MPLAB X IDE v1.51」と「MPLAB XC8 v1.12」の二つのファイルをダウンロードしてください。（v以降の数字はバージョンによって変化します。このマニュアルのパソコンの OS は Windows を使っています。）

Title	Date Published	Size	D/L
Windows (x86/x64)			
MPLAB® X IDE v1.51	11/08/2012	312Mb	
MPLAB® X IDE Release Notes / User' Guide v1.51 (supersedes info in installer)	09/11/2012	150KB	
MPLAB® C18 Lite Compiler for PIC18 MCUs v3.40	05/30/2012	73Mb	
MPLAB® XC8 Compiler v1.12	12/4/2012	168Mb	
MPLAB® XC16 Compiler v1.10	05/30/2012	98Mb	
MPLAB® XC32 Compiler v1.11a	10/04/2012	105Mb	
Linux 32-Bit and Linux 64-Bit (Requires 32-Bit Compatibility Libraries)			
MPLAB® X IDE v1.51	11/08/2012	260Mb	
MPLAB® X IDE Release Notes / User' Guide v1.51 (supersedes info in installer)	09/11/2012	150KB	
MPLAB® C18 Lite Compiler for PIC18 MCUs v3.40	05/30/2012	70Mb	
MPLAB® XC8 Compiler v1.12	12/4/2012	172Mb	
MPLAB® XC16 Compiler v1.10	05/30/2012	81Mb	
MPLAB® XC32 Compiler v1.11	10/04/2012	104Mb	
Mac (10.X)			
MPLAB® X IDE v1.51	11/08/2012	231Mb	
MPLAB® X IDE Release Notes / User' Guide v1.51 (supersedes info in installer)	09/11/2012	150KB	
MPLAB® C18 Lite Compiler for PIC18 MCUs v3.40	05/30/2012	73Mb	
MPLAB® XC8 Compiler v1.12	12/4/2012	169Mb	
MPLAB® XC16 Compiler v1.10	05/30/2012	75Mb	
MPLAB® XC32 Compiler v1.11	10/04/2012	107Mb	

マイクロチップ・テクノロジー・ジャパンのページから日本語の資料を入手しましたが，英文の資料が最新版になるのでこちらも入手しておきましょう。先ほどの「MPLAB X IDE」のページの「MPLAB X Documentation」というボタンをクリックしてください。ドキュメントの一覧が開きます。「MPLAB X IDE User's Guide」と「Using PICkit3 for MPLAB X IDE」をダウンロードしてください。

Title	Date Published	Size	D/L
MPLAB® X IDE User's Guide	11/29/2011	515 KB	
Using PICKit™ 3 for MPLAB® X IDE	11/10/2008	399 KB	
Using MPLAB® ICD 3 In-Circuit Debugger for MPLAB® X IDE	07/01/2011	573 KB	
MPLAB® ICD 3 for MPLAB® X IDE User's Guide	07/01/2011	573 KB	
Using MPLAB® REAL ICE In-Circuit Emulator for MPLAB® X IDE	07/01/2011	573 KB	

「MPLAB　XC8」のドキュメントも入手しておきましょう。日本語のページに戻って，ページの下の「製品一覧」という項目の「開発ツール」の欄にある「コンパイラ」という項目をクリックしてください。ブラウザによりますが，別タブで「MPLAB　XC　Compilers」のページが開きます（英文サイト）。このページの下の部分に「MPLAB　XC8　User Guide」があります。クリックしてダウンロードしてください。なお，「MPLAB　XC8」の日本語の資料はまだないようです。



2. 開発環境のインストール

「MPLAB X IDE」も「MPLAB XC8」も英語のソフトです。説明文は全て英文になります。

最初に「MPLAB X IDE」をインストールします。ダウンロードしたファイルをクリックしてください。あとは画面の指示に従ってインストールします。

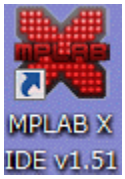
次に「MPLAB XC8」をインストールします。ダウンロードしたファイルをクリックしてください。あとは画面の指示に従ってインストールします。

いずれも途中でいくつかのダイアログが開きますが，基本的にそのままの設定でかまいません。使用許諾に同意するかどうか尋ねるダイアログのときは「agree」を選択してください。また，プロダクトキーの入力が求められることがありますが，FREE 版で使用するときは未入力でかまいません（未入力のときに FREE 版になる）。

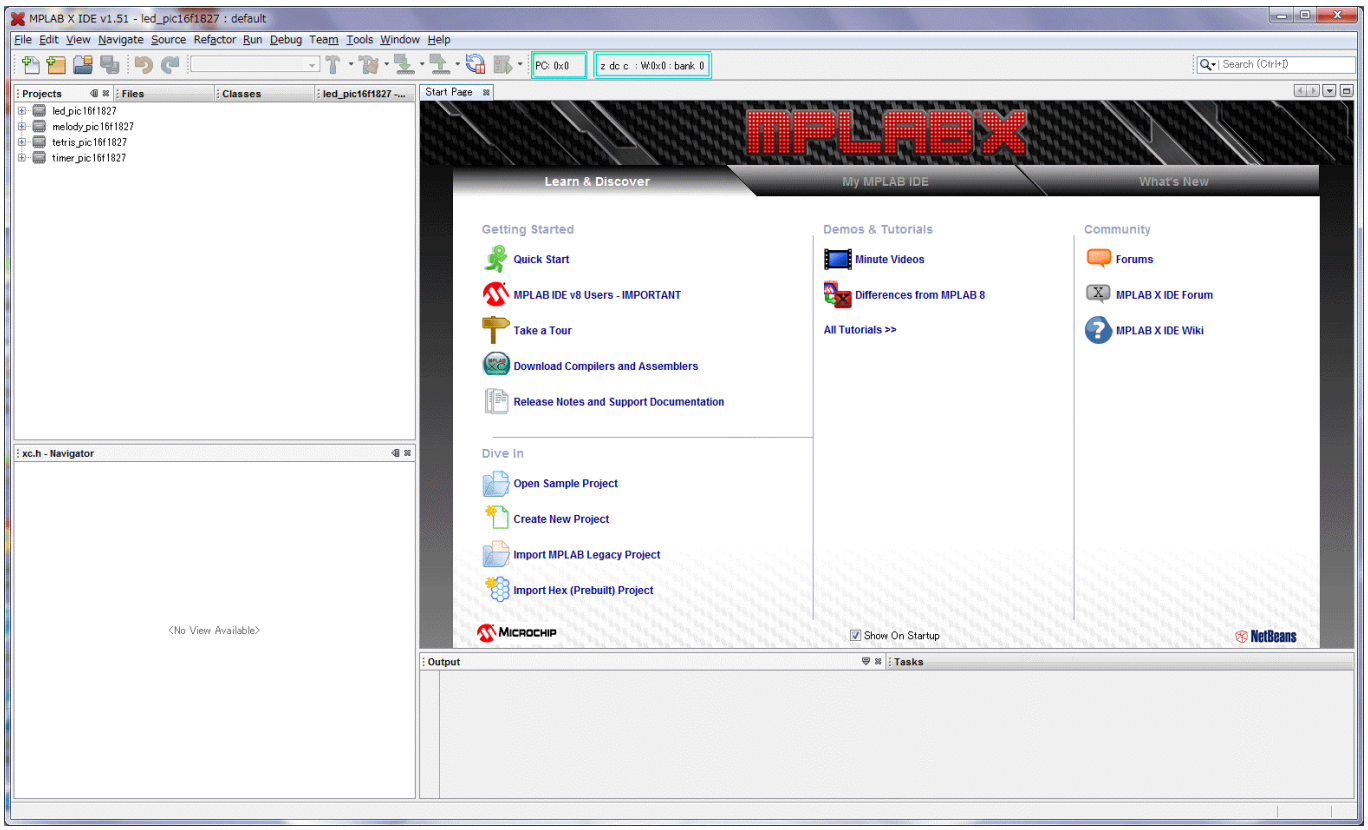
3. プロジェクトの新規作成

プロジェクトを新規作成し，ソースリストを入力，ビルドした後，PIC16F1827 にダウンロードするところまで見てみましょう。ここで作成するプロジェクト名は「00_tutorial」です。

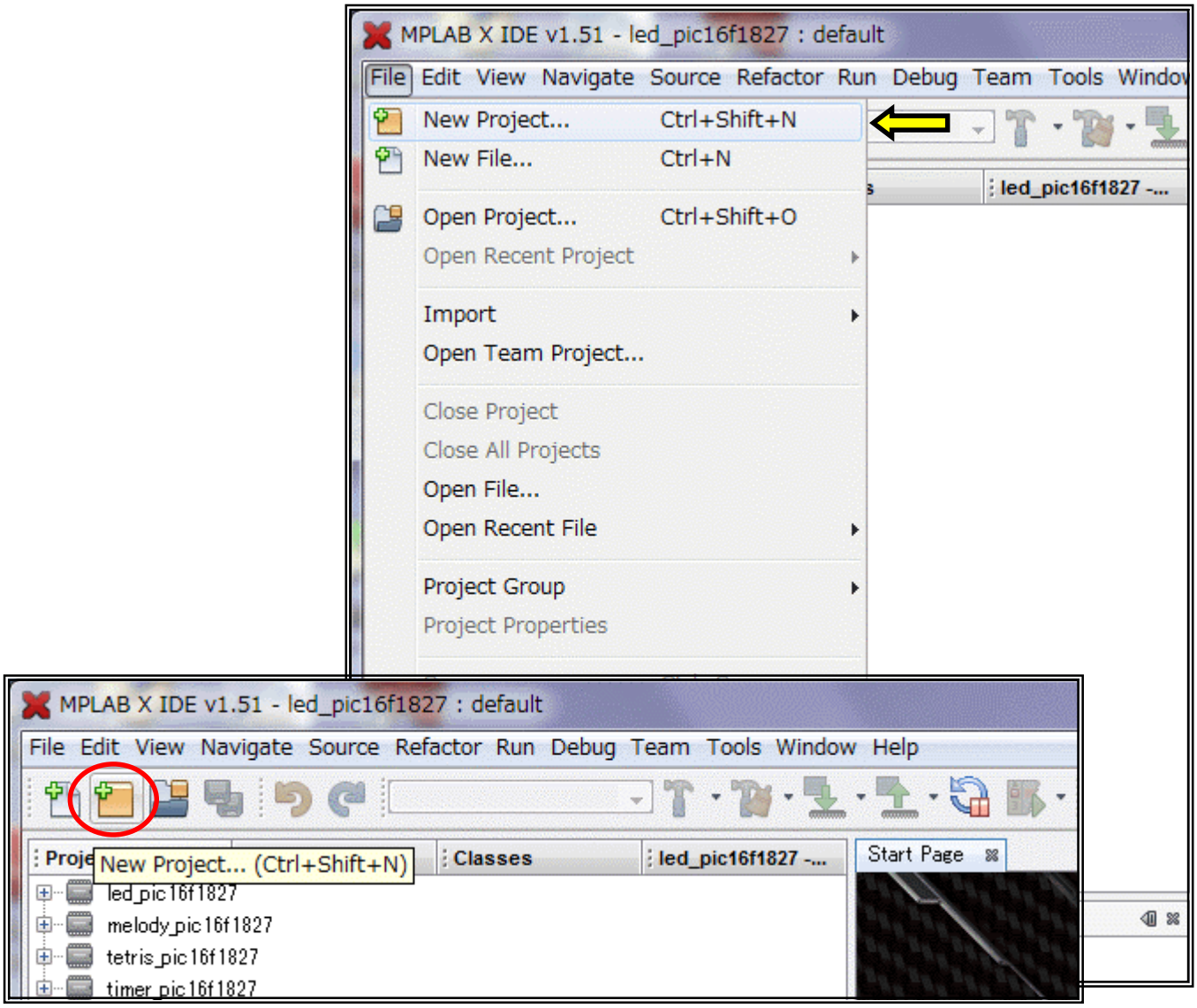
デスクトップに「MPLAB X IDE」のショートカットアイコンが作られていますのでクリックして起動してください。（アイコンの数字はバージョンによって変更されます。アイコンが作られていないときはスタートメニューから起動してください。）



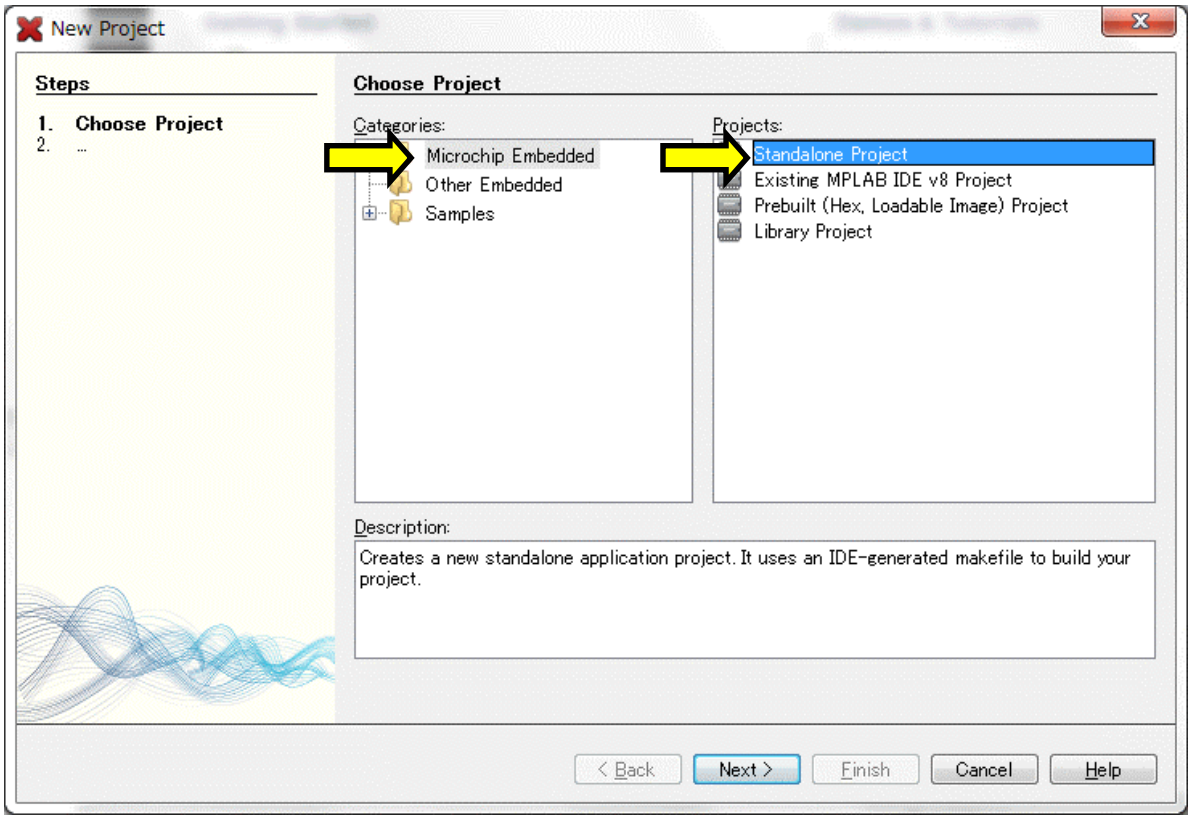
しばらくすると，次のようなウィンドウが開きます。ここからスタートしていきます。（画面をキャプチャしたこのパソコンはすでにプロジェクトをいくつか作っていますので，作成済みのプロジェクトがプロジェクトウィンドウに表示されています。）



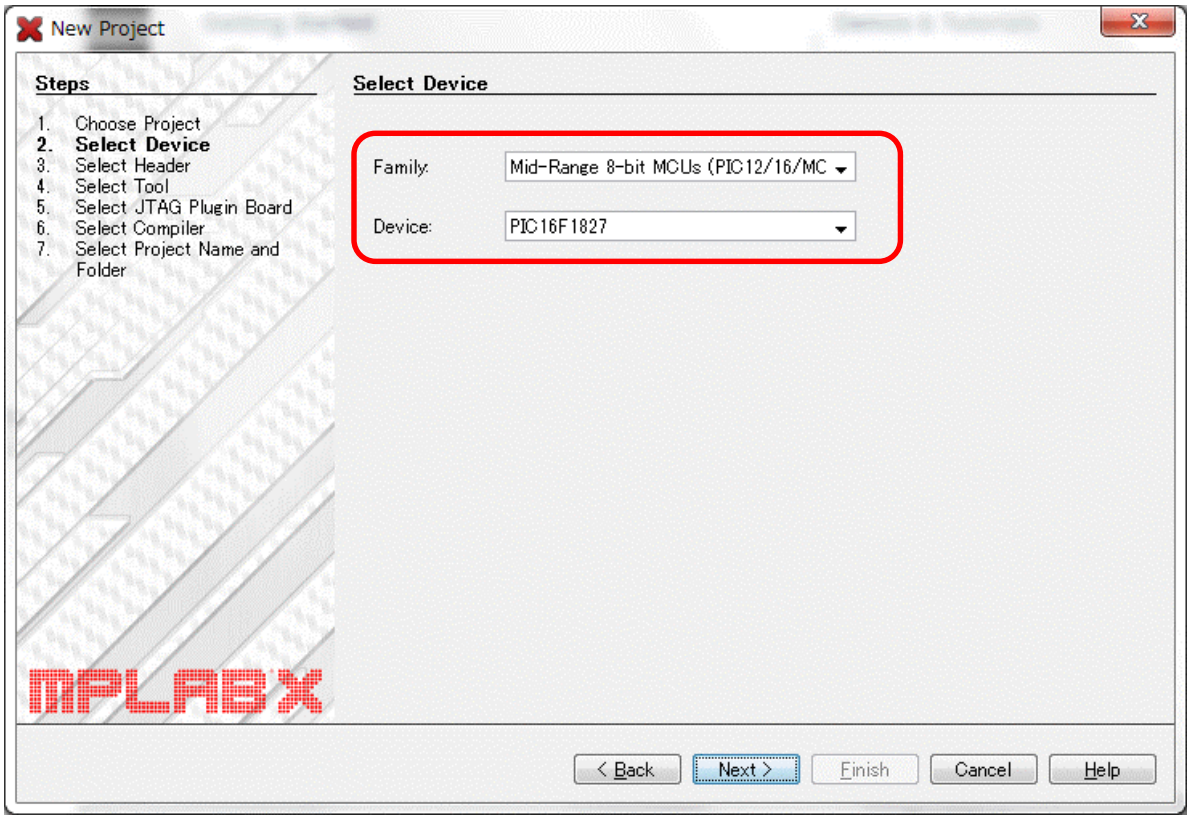
メニューから[File]→[New Project]と選択するか,「New Project」ボタンをクリックしてください。



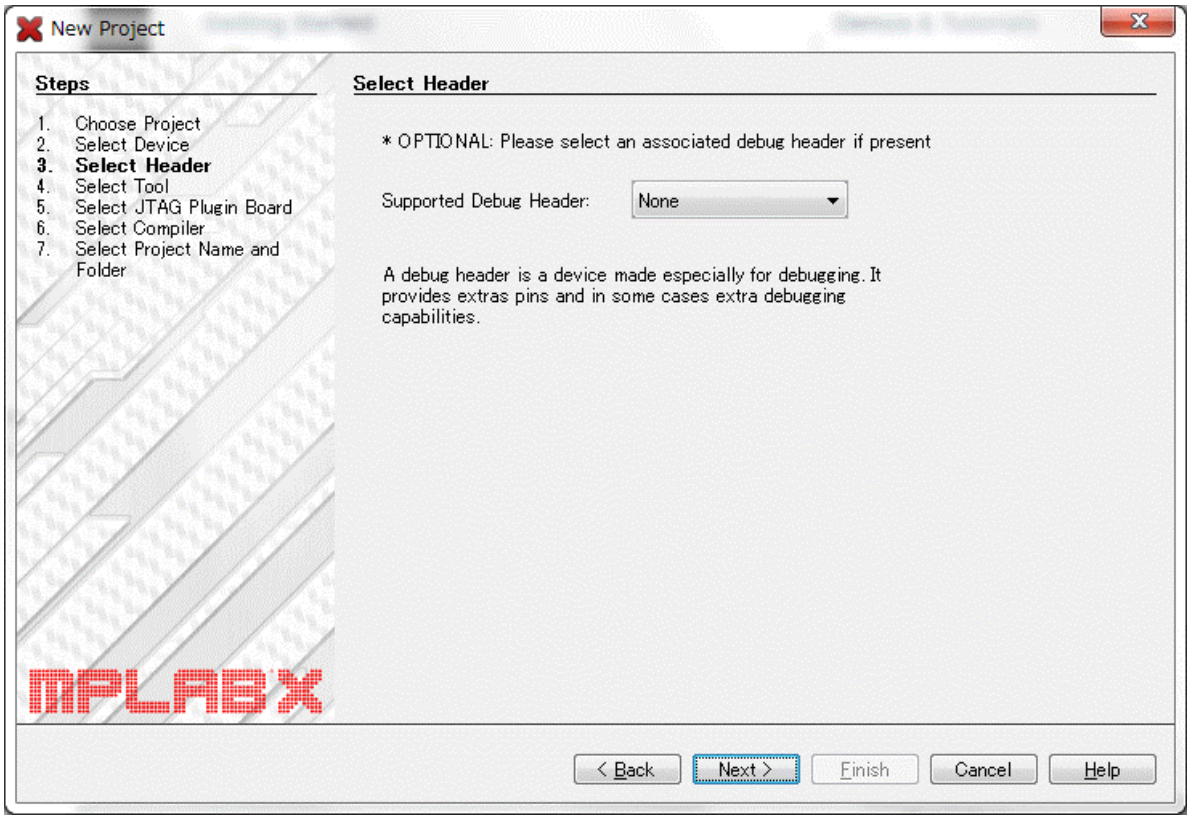
「1. Choose Project」ダイアログで,「Categories:」は「Microchip Embedded」,「Projects:」は「Standard Project」を選択し,「Next>」をクリックします。



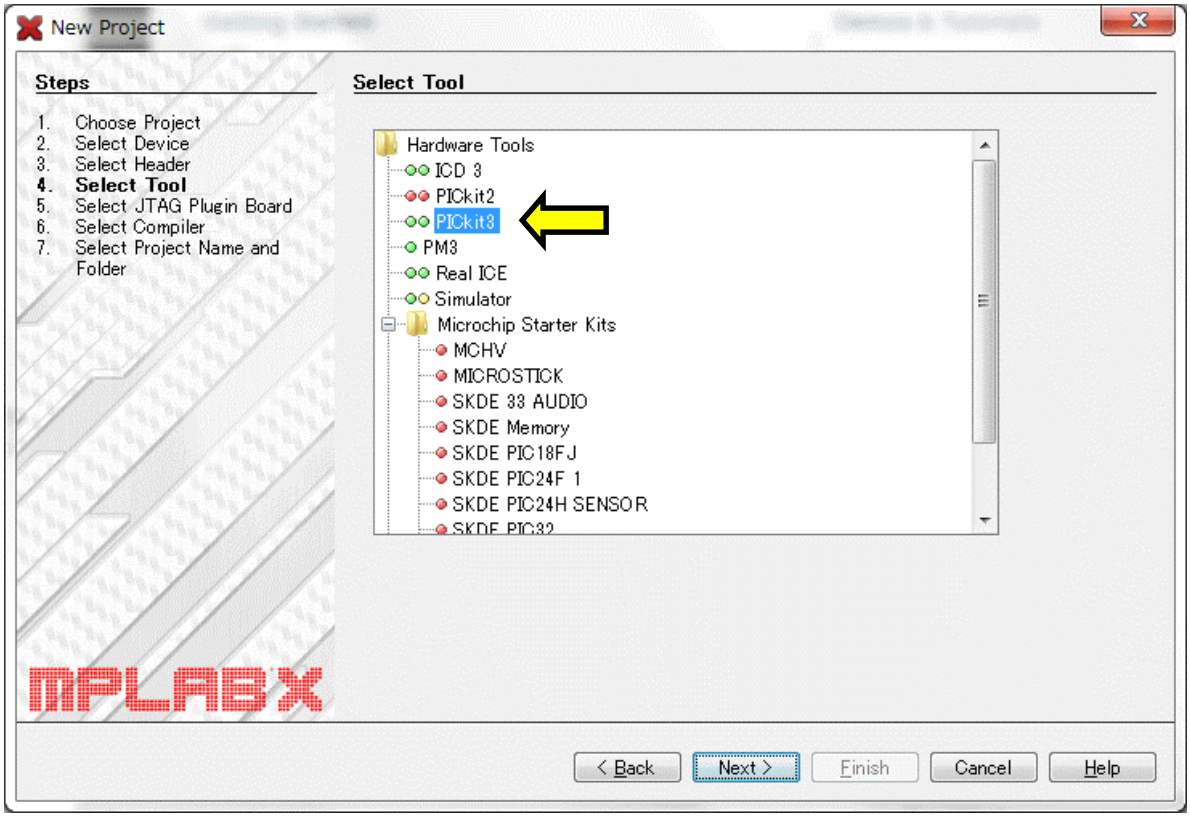
「2. Select Device」ダイアログで,「Family:」は「Mid-Range 8-bit MCUs(PIC12/16/MCP)」,「Device:」は「PIC16F1827」を選択し,「Next>」をクリックします。



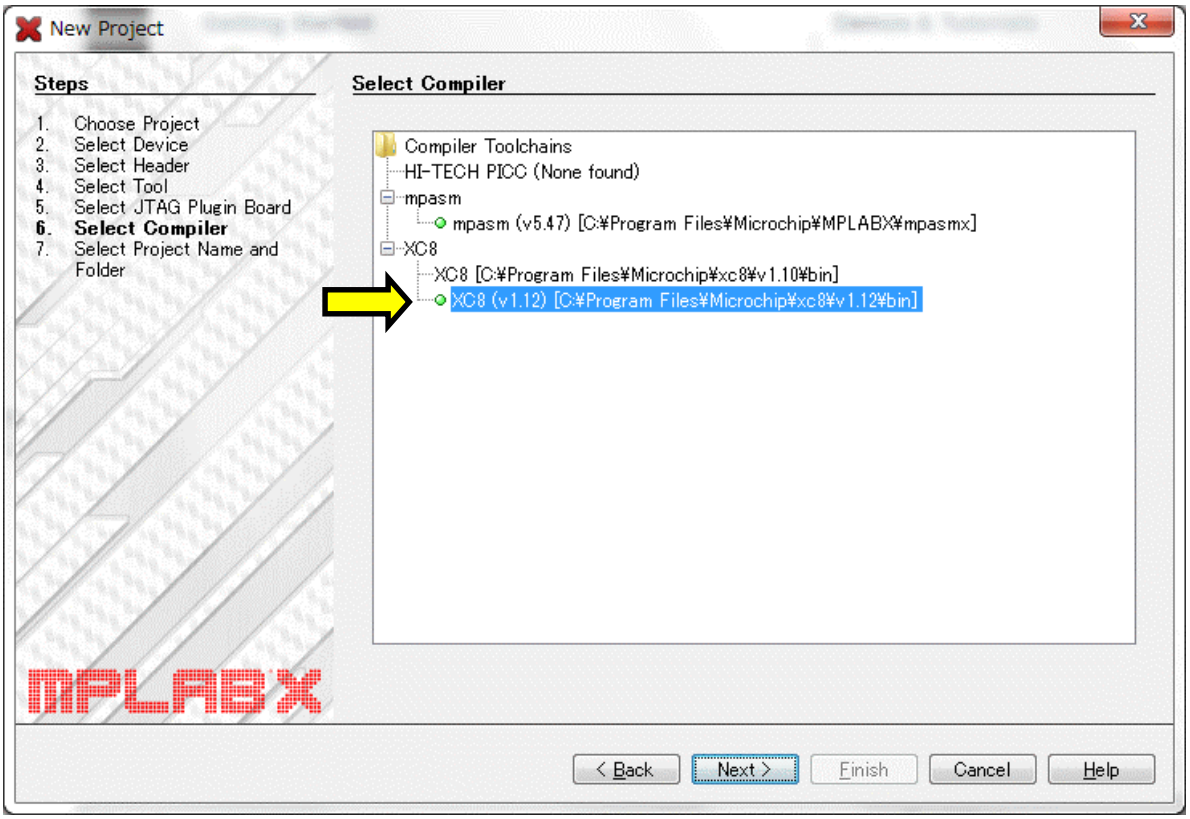
「3. Select　Header」ダイアログは変更しません。「Next＞」をクリックします。



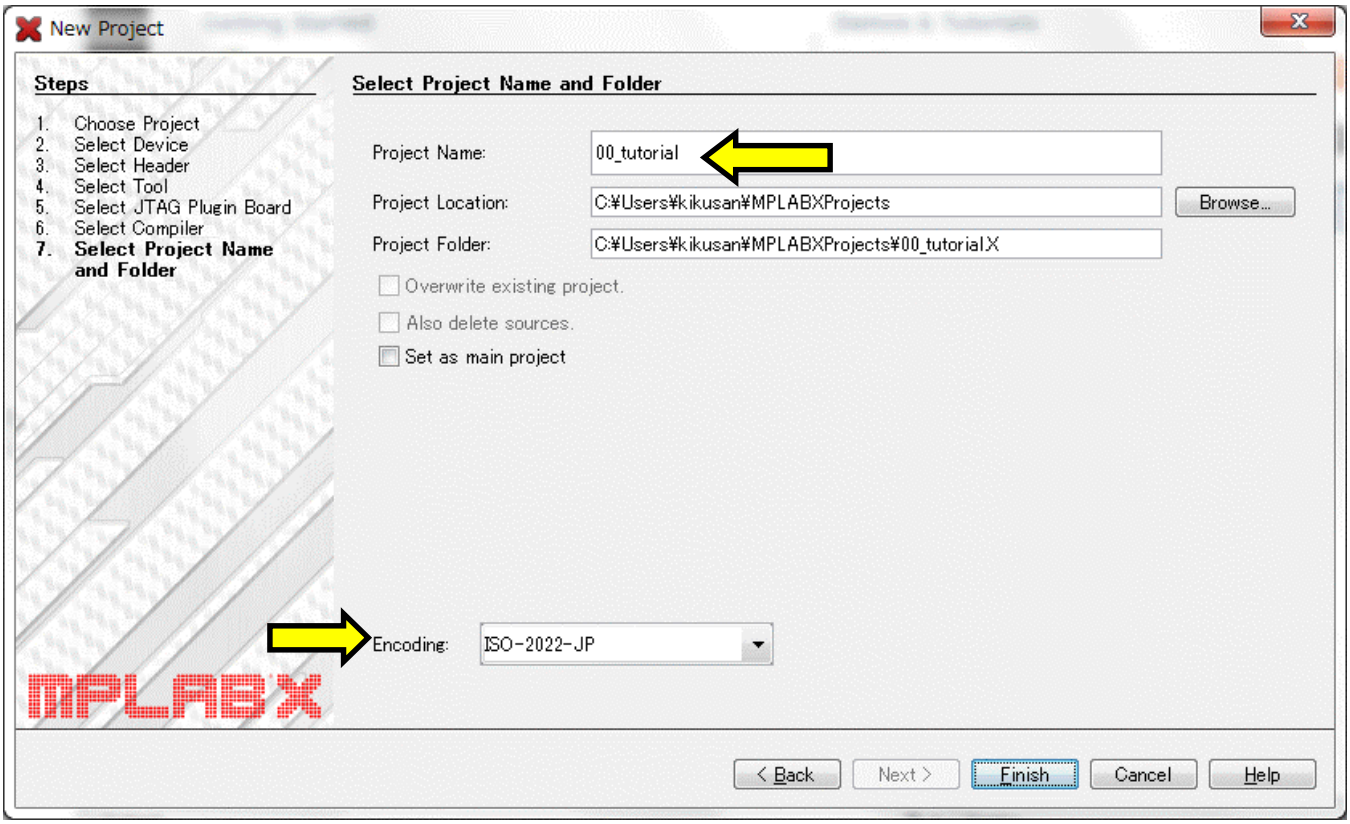
「4. Select　Tool」ダイアログで「PICkit3」を選択し,「Next＞」をクリックします。



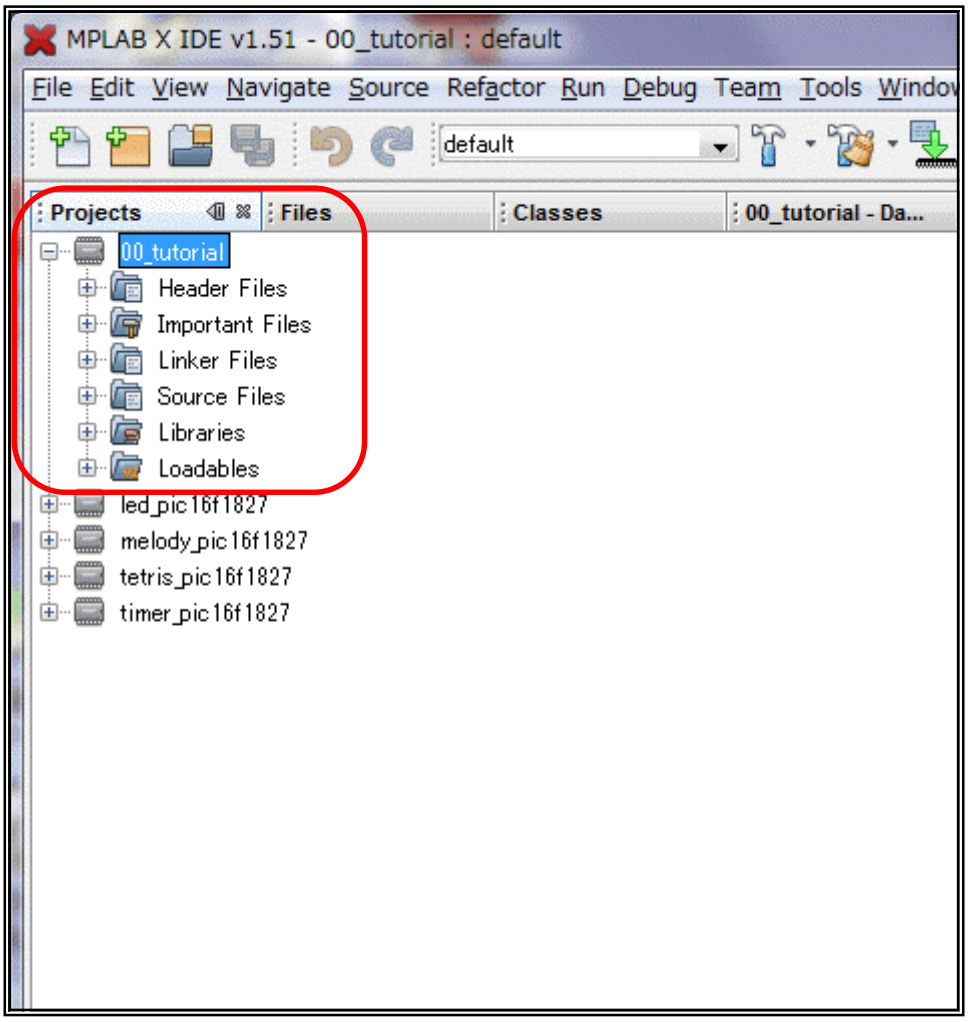
「6. Select Compiler」ダイアログで「XC8(v1.12)」を選択し，「Next＞」をクリックします。(数字はバージョンによって変更されます。)



「7. Select Project Name and Folder」ダイアログで，「Project Name:」に「00_tutorial」を入力し，「Encoding:」を「ISO-2022-JP」か「ISO-2022-JP-2」に変更します(ソースリストに日本語のコメントを入力できるようになります)。最後に「Finish」をクリックします。



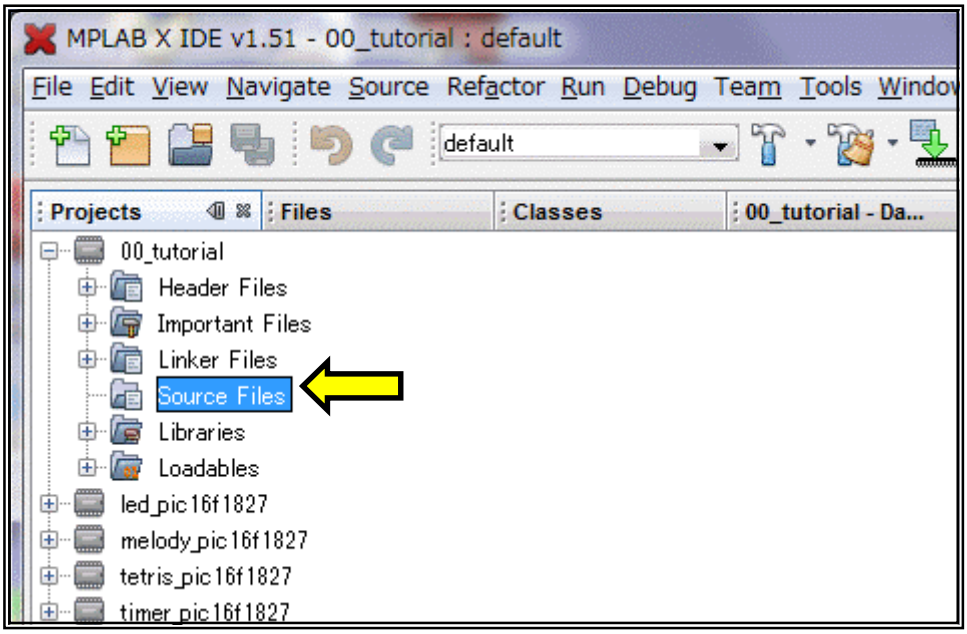
「Projects」ウィンドウに「00_tutorial」が追加されます。



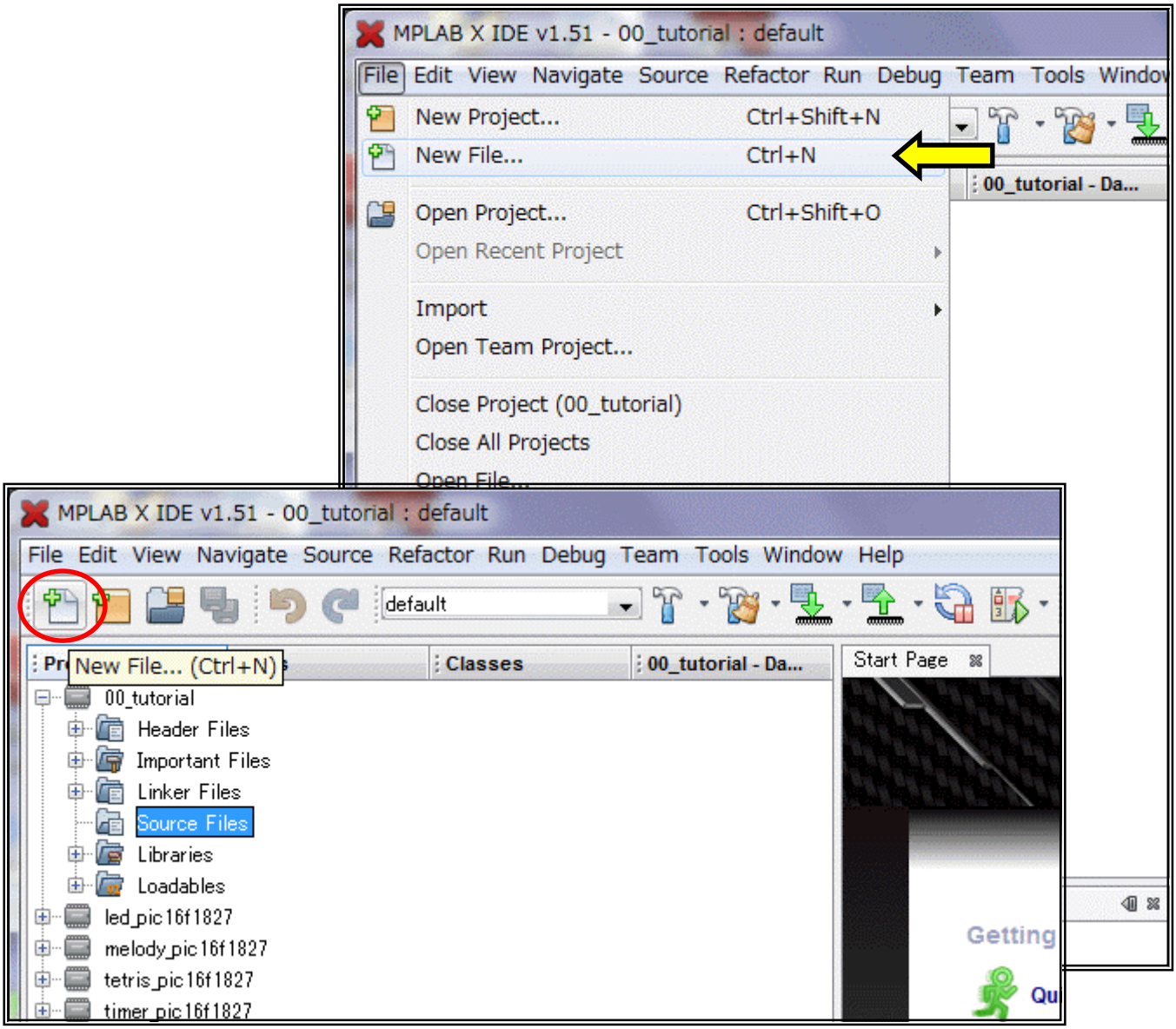
4. ソースリストの入力

続いてソースリストを入力します。ここで作成するソースファイル名は「00_tutorial. c」です。

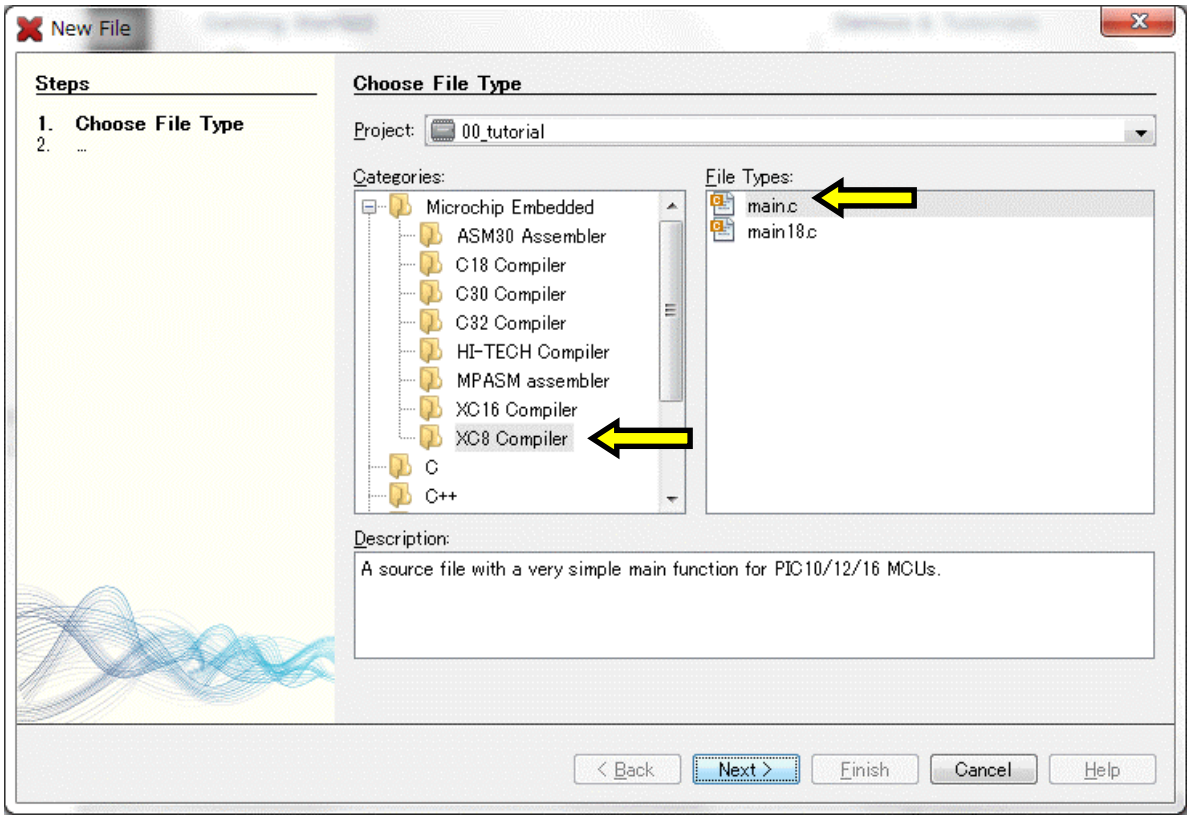
まず，プロジェクトの「Source Files」をクリックします。



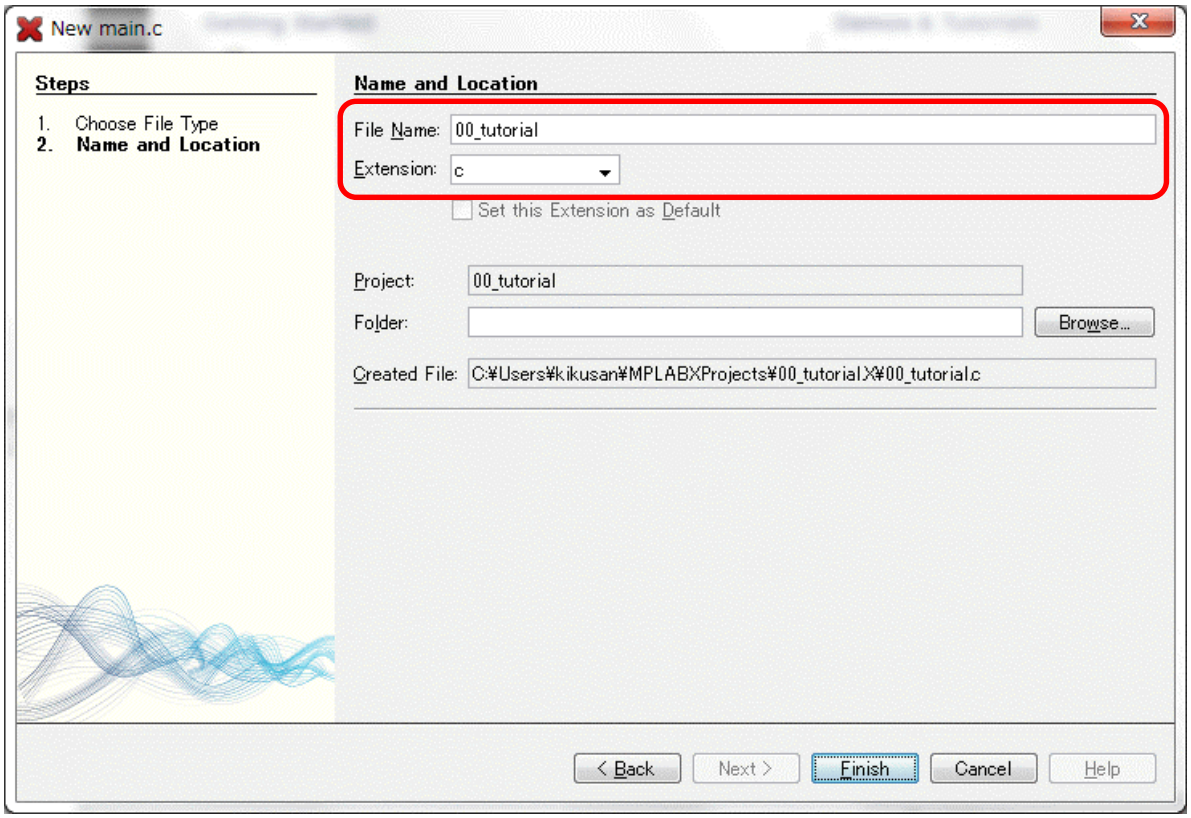
次に，メニューから[File]→[New　File]と選択するか，「New　File」ボタンをクリックしてください。



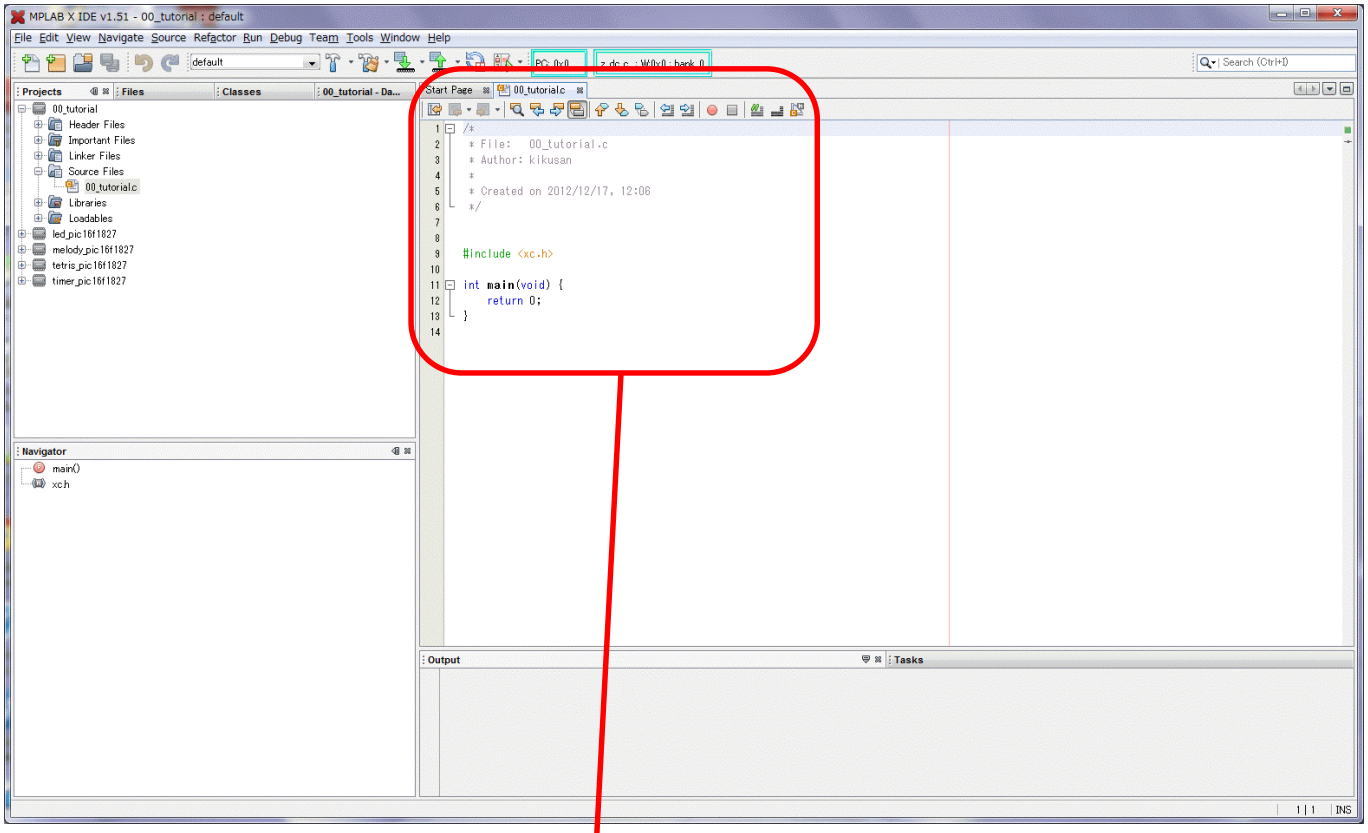
「1. Choose File Type」ダイアログで,「Categories:」は「XC8 Compiler」,「File Types:」は「main. c」を選択し,「Next>」をクリックします。

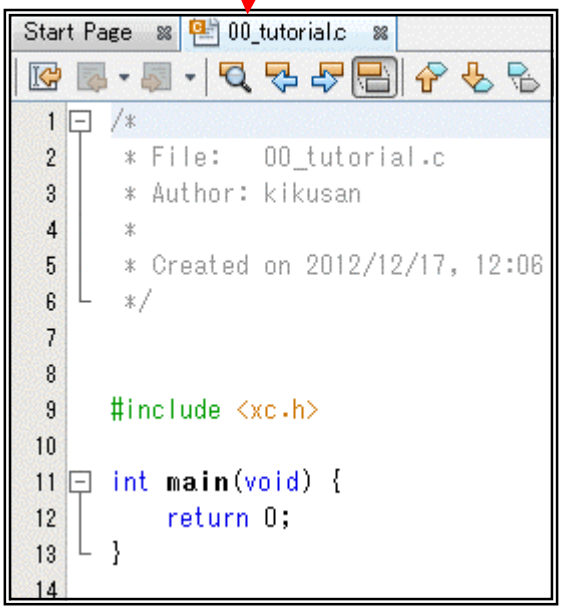


「2. Name and Location」ダイアログで,「File Name:」に「00_tutorial」を入力,「Extension:」は「c」を選択します。最後に「Finish」をクリックします。

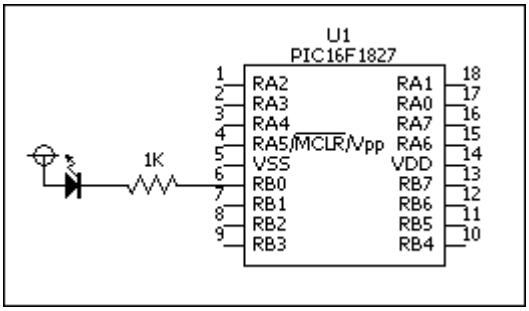


すると，ソースファイルが自動生成されます。





このソースファイルを変更していきます。今回のプログラムは RB0 につないだ LED を点滅するというものです。回路は右のようになります。(この回路は説明のために考えたもので，プリント基板には実装されていません。実際に LED を光らせたい場合はユニバーサルエリアに組み立ててください。巻末の回路図参照。)



次のようにソースファイルを変更してください。今回は手順の説明なので，プログラムの詳細はコメントを見てください。もちろん，コメントは入力しなくても動作します。

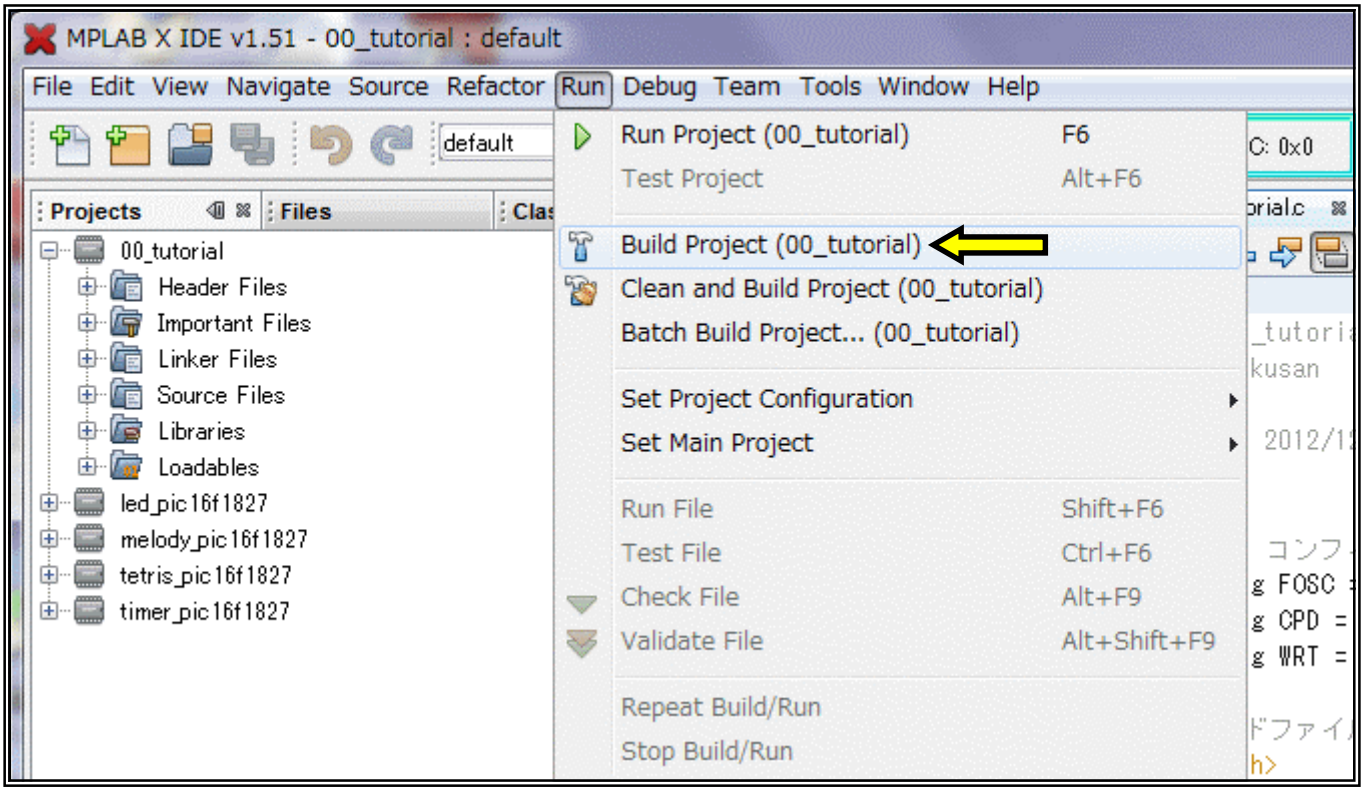
Start Page | 00_tutorial.c

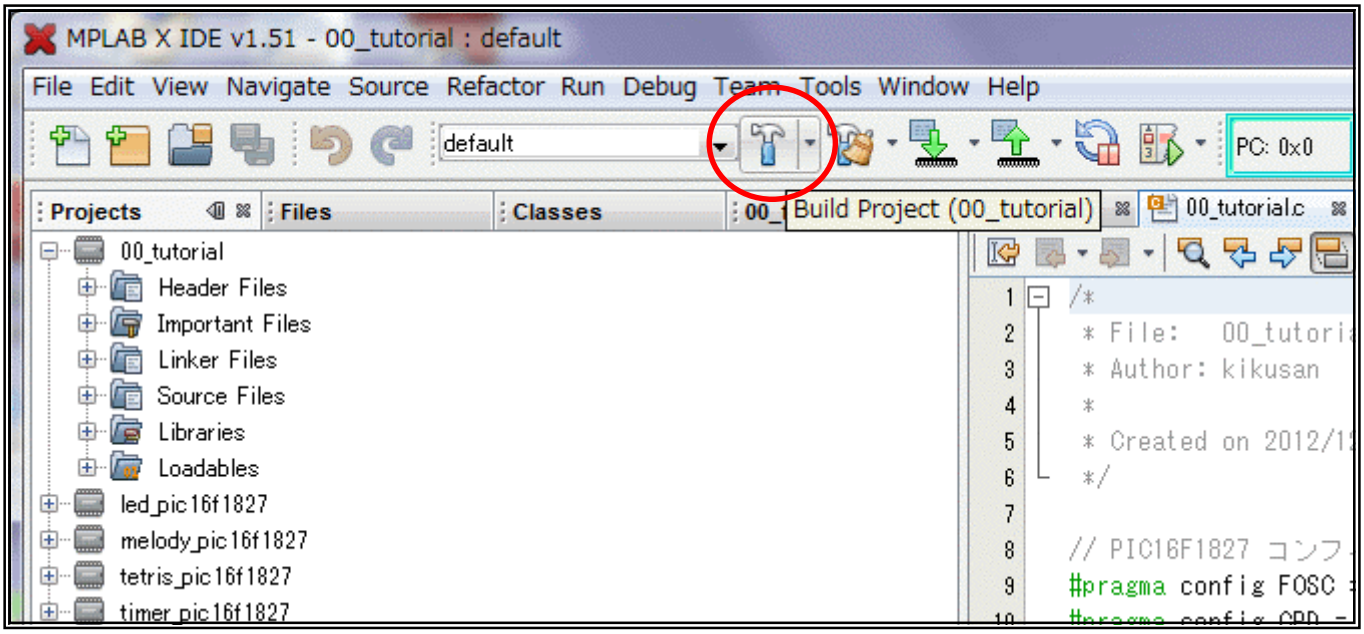
```
/*
 * File:   00_tutorial.c
 * Author: kikusan
 *
 * Created on 2012/12/17, 12:06
 */

// PIC16F1827 コンフィグレーションビットの設定
#pragma config FOSC = INTOSC, WDTE = OFF, PWRTE = ON,MCLRE = OFF, CP = OFF
#pragma config CPD = OFF, BOREN = ON, CLKOUTEN = OFF, IESO = ON,FCMEN = ON
#pragma config WRT = OFF,PLLEN = ON, STVREN = ON, BORV = LO, LVP = OFF

// インクルードファイル
#include <xc.h>

//メインルーチン
void main(void) {
    unsigned long i;

    // オシレータ制御レジスタのイニシャライズ
    OSCCON = 0xf0;  // 4xPLL-Enable,8MHz = 32MHz

    // PORTBのイニシャライズ
    ANSELB = 0x00;  // 端子はデジタルI/Oとして使う
    TRISB = 0x00;   // 出力ポートに設定
    PORTB = 0xff;   // 初期出力値

    // メインループ
    while(1){
        RB0 = 0;                    // LEDオン
        for(i=0; i<100000; i++){}   // ウェイト
        RB0 = 1;                    // LEDオフ
        for(i=0; i<100000; i++){}   // ウェイト
    }
}
```

入力の途中でも，ときどきキーボードから「Ctrl」+「S」(ファイルの上書き保存)でセーブする習慣をつけましょう。入力が終わったときにもセーブします。

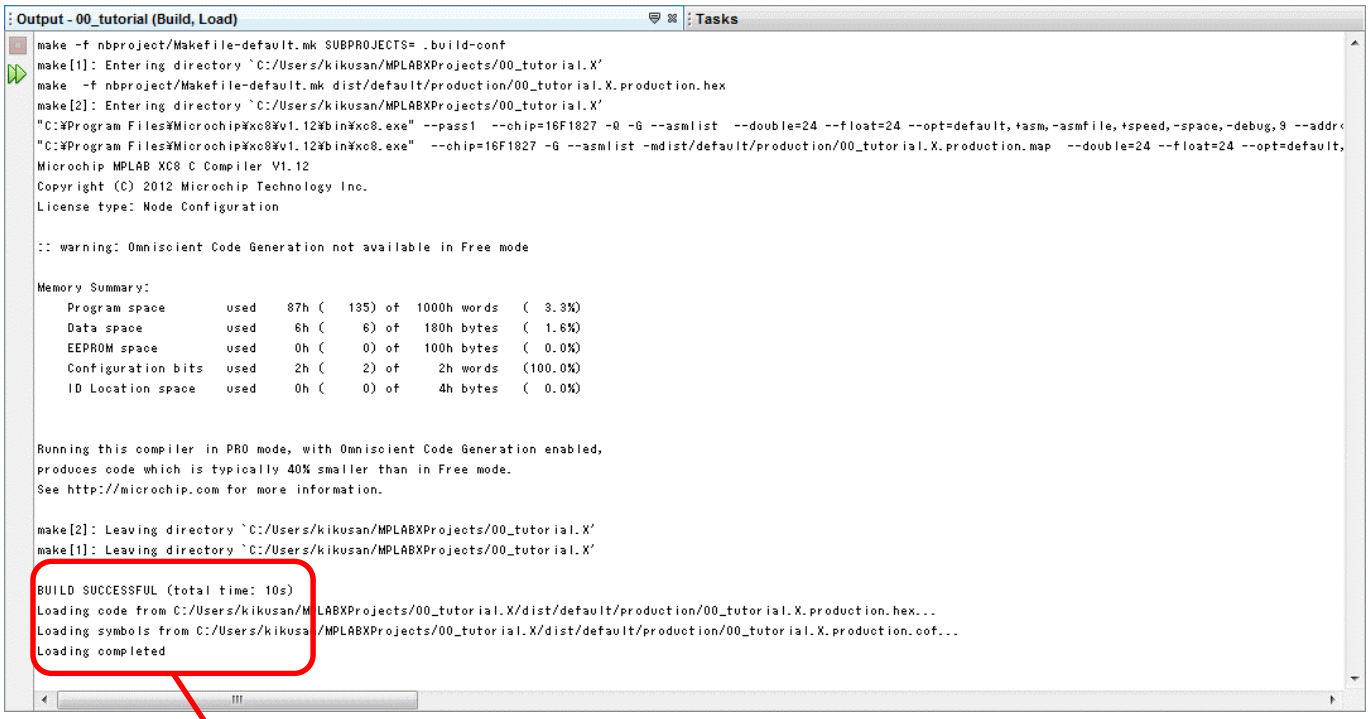
5. ビルド

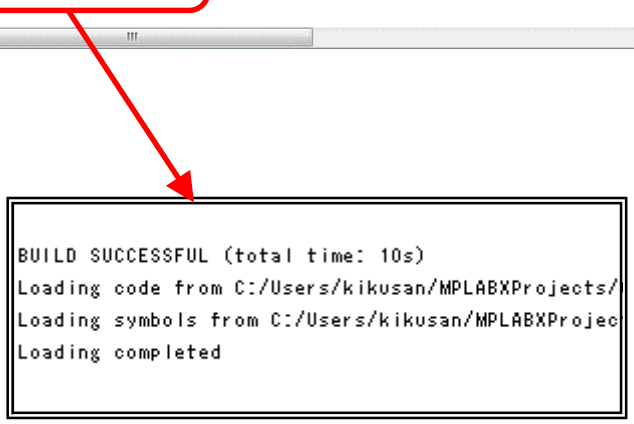
ソースリストの入力が終わったらビルドします。メニューから[Run]→[Build　Project]と選択するか,「Build Project」ボタンをクリックしてください。





ビルドの状態は「Output」ウィンドウに表示されます。ビルドが成功すると次のように表示されます。





エラーが出るとすれば，今回はタイプミスがほとんどだと思います。エラーが表示されたら，エラー箇所を探して修正してください。

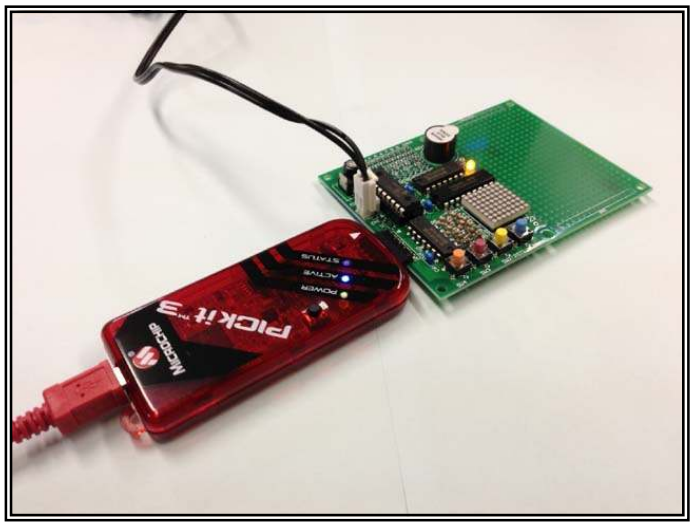
6. PICkit3 を使って PIC16F1827 にダウンロード

ビルドが成功したら PIC16F1827 にプログラムをダウンロードしましょう。

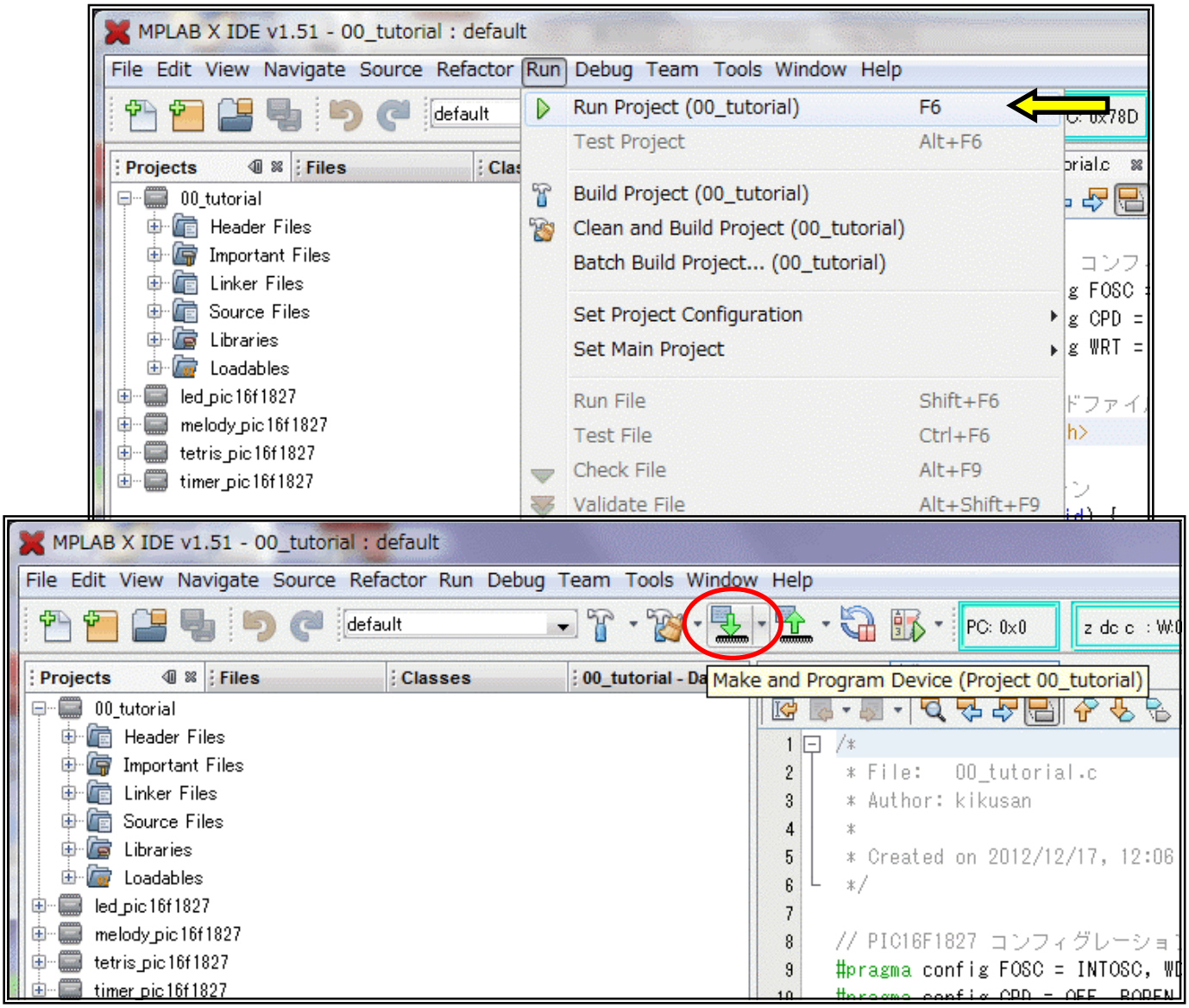
まず，「PICkit3」とパソコンを，「PICkit3」に付属している USB ケーブルで接続します。「PICkit3」のドライバは「MPLAB X IDE」のインストールの際にパソコンにコピーされます。ドライバのインストールが必要な場合は，パソコンの画面の指示に従ってインストールしてください。

次に tk-PIC1827 の CN2 と「PICkit3」を接続します。

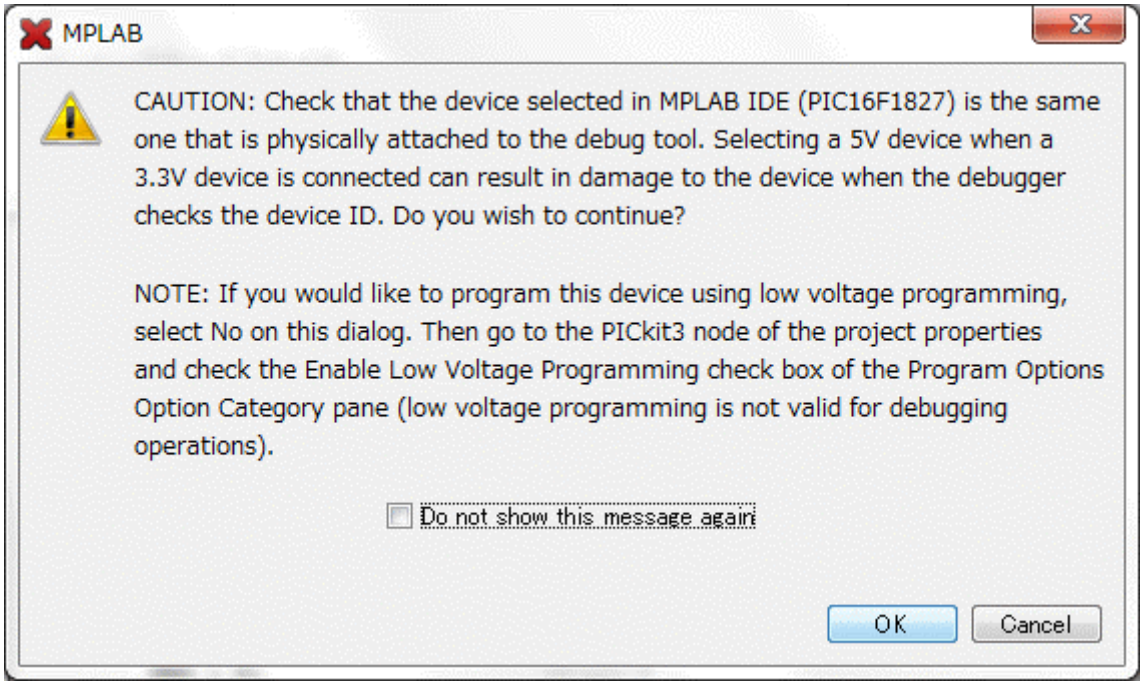
最後に tk-PIC1827 の電源を入れてください。



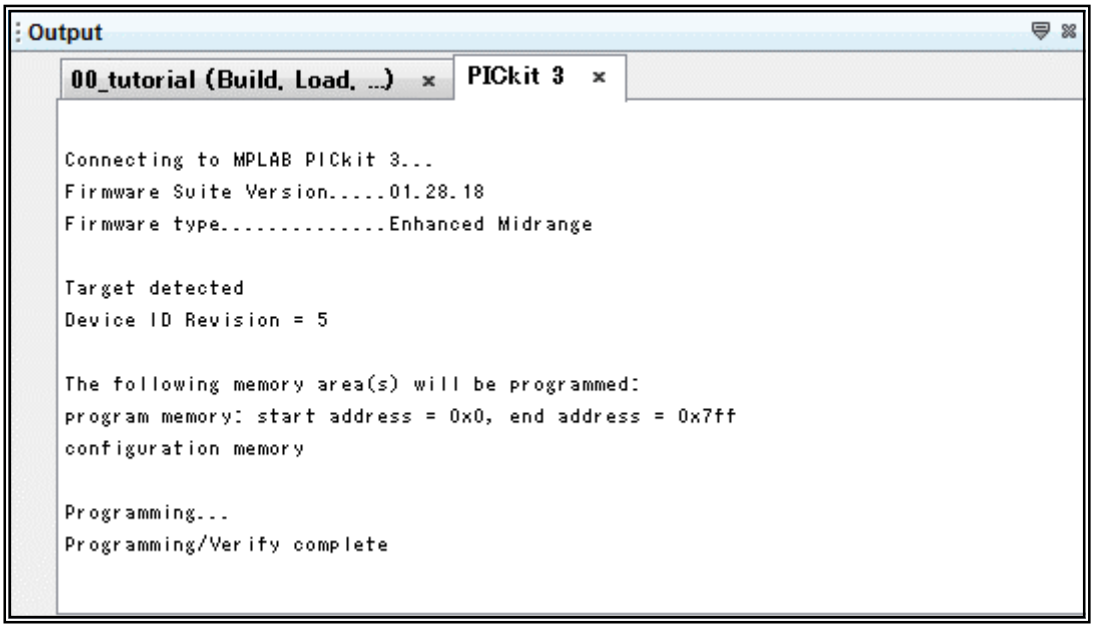
接続が終わったら，メニューから[Run]→[Build Project]と選択するか，「Make and Program Device」ボタンをクリックしてください。必要な場合は自動的にビルドされ，ダウンロードが始まります。



途中で，次のような警告のダイアログが開くことがありますが，「OK」をクリックしてください。



「PICkit3」の情報は「Output」ウィンドウの「PICkit3」タブに表示されます。右のように表示されれば PIC16F1827 へのダウンロードは成功です。



ダウンロードが終わるとプログラムがスタートします。回路図どおり RB0 に LED を接続していれば，LED が点滅します。あとは「PICkit3」を外した状態でも，電源を入れればダウンロードしたプログラムがスタートします。

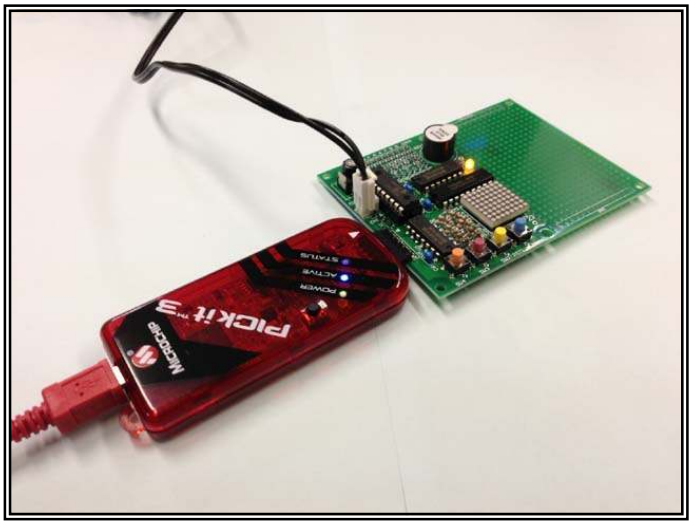
■ 「PICkit3」接続時の注意事項 ■

「PICkit3」は PIC16F1827 の RA5(MCLR/Vpp)，RB6(ICSPCLK)，RB7(ICSPDAT)を使って通信します。このマニュアルで取り上げたプログラム例ではこれらのポートを使用しないので問題ありませんが，これらのポートを使う回路を追加した場合ダウンロードに支障が出る可能性があります。その際は，ダウンロードの際に回路を切り離せるようにするとよいでしょう。

なお，回路設計時の注意事項の詳細は，マイクロチップテクノロジーの資料か，このマニュアルの付録をご覧ください。

7. PICkit3 を使ったデバッグ

「PICkit3」はプログラムのダウンロードだけでなく，プログラムを実機で動かしながらデバッグする，デバッガとして使うことができます。

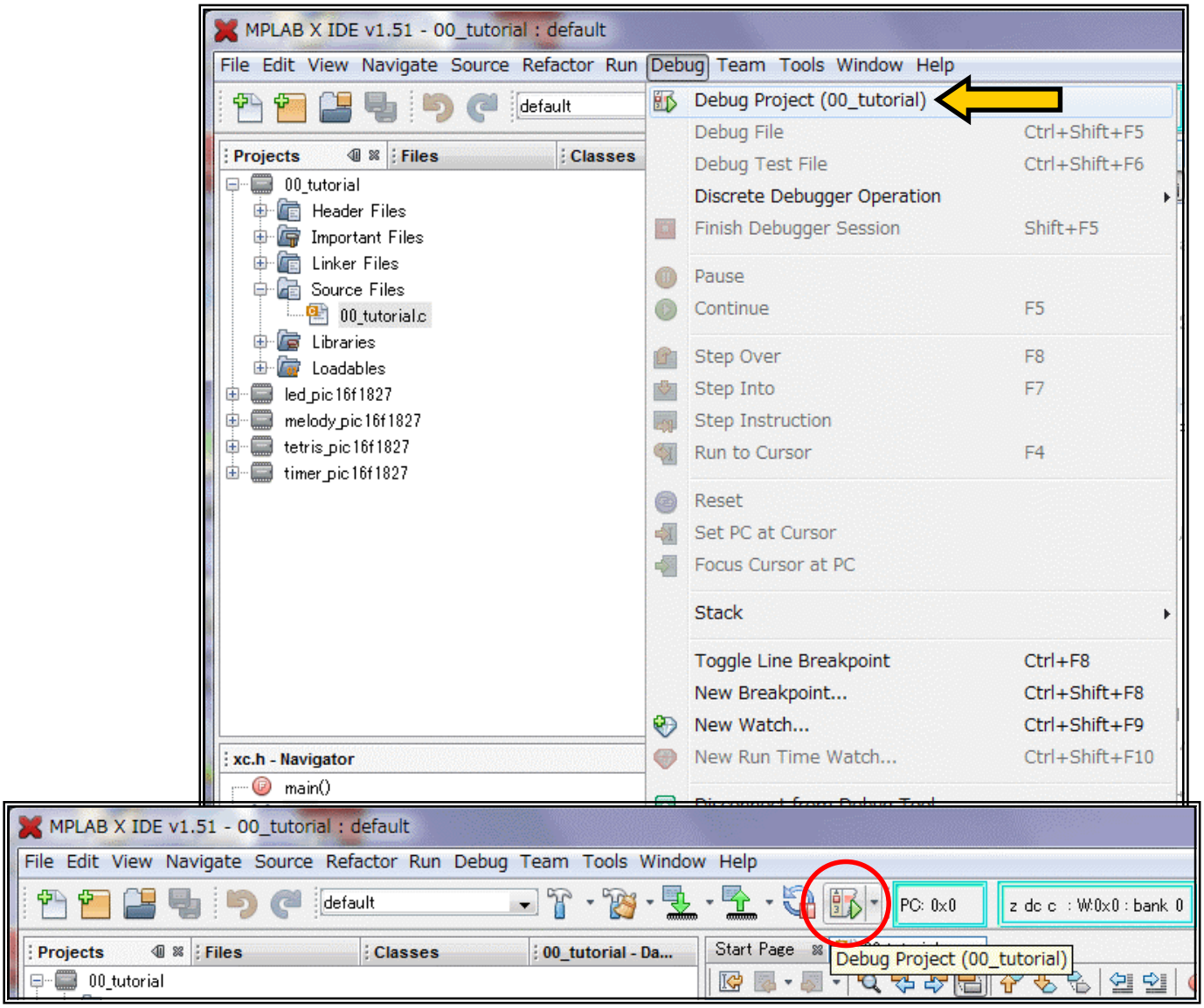


まず，「PICkit3」とパソコンを，「PICkit3」に付属している USB ケーブルで接続します。「PICkit3」のドライバは「MPLAB X IDE」のインストールの際にパソコンにコピーされます。ドライバのインストールが必要な場合は，パソコンの画面の指示に従ってインストールしてください。

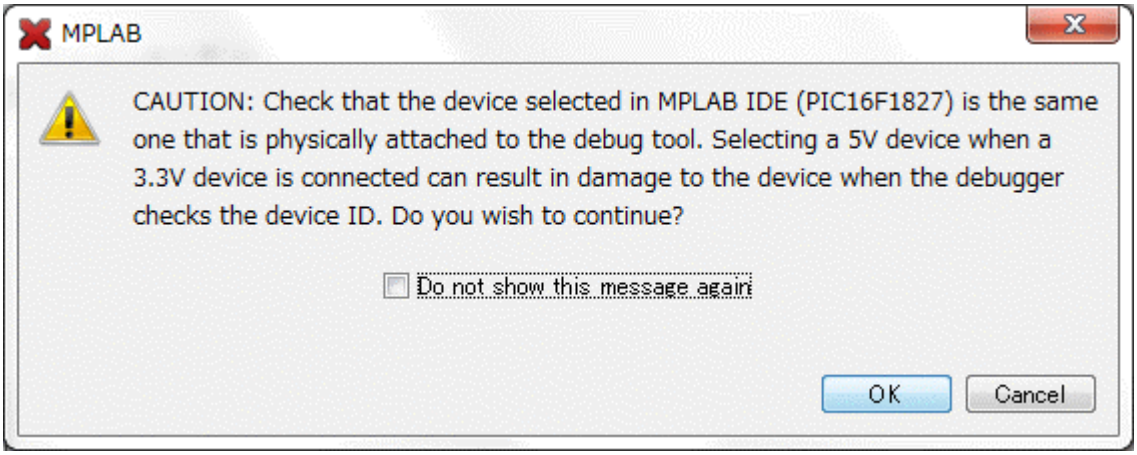
次に tk-PIC1827 の CN2 と「PICkit3」を接続します。

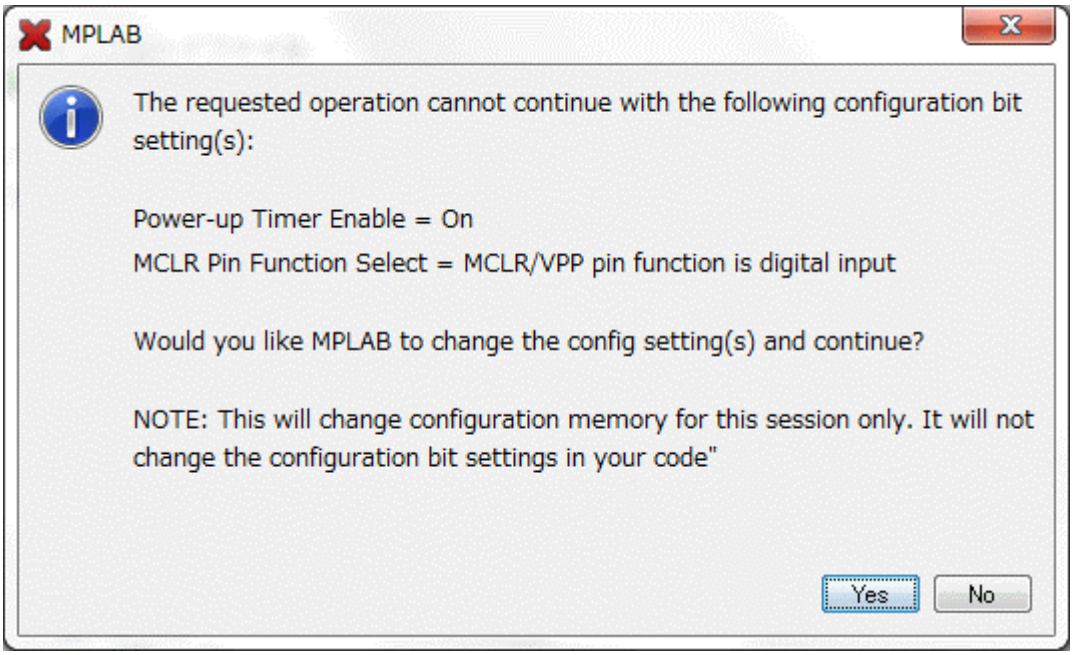
最後に tk-PIC1827 の電源を入れてください。

接続が終わったら，メニューから[Debug]→[Debug Project]と選択するか，「Debug Project」ボタンをクリックしてください。必要な場合は自動的にビルドされ，ダウンロードが始まります。

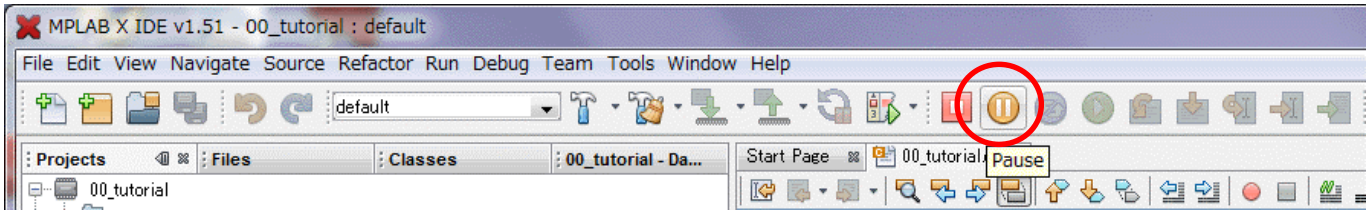


途中で，次のような警告や確認のダイアログが開くことがありますが，「OK」，「Yes」をクリックしてください。



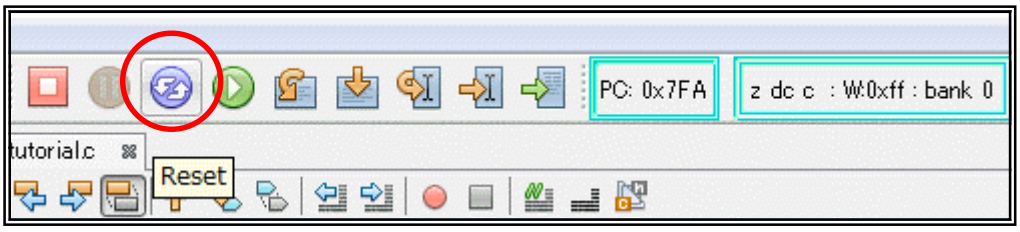


ダウンロードが終わると自動的にプログラムがスタートしますので，まずはプログラムを一時停止します。「Pause」ボタンをクリックします。

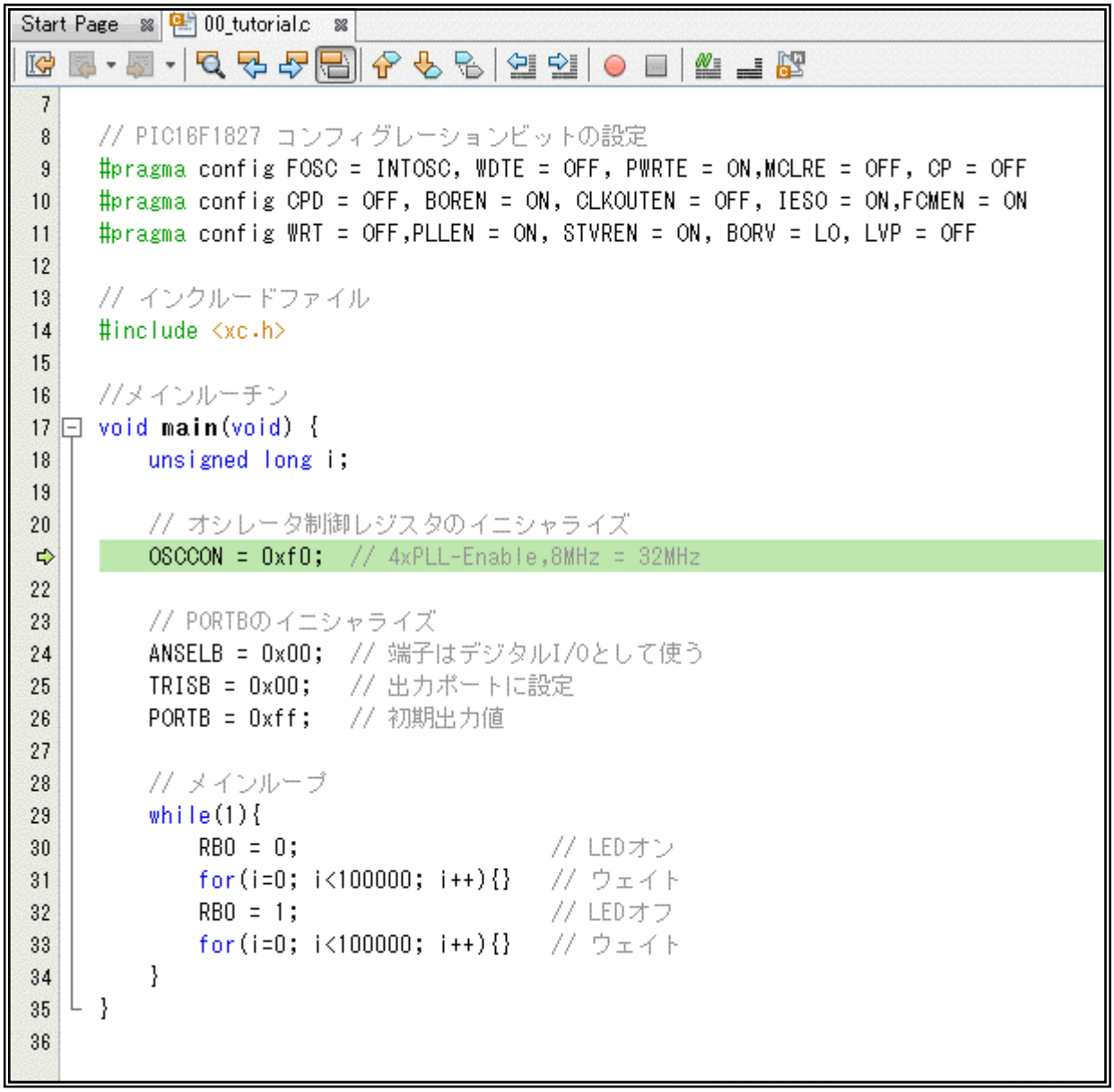


■ **リセット**

プログラムをスタートアドレスから動かしてみましょう。「Reset」ボタンをクリックしてください。

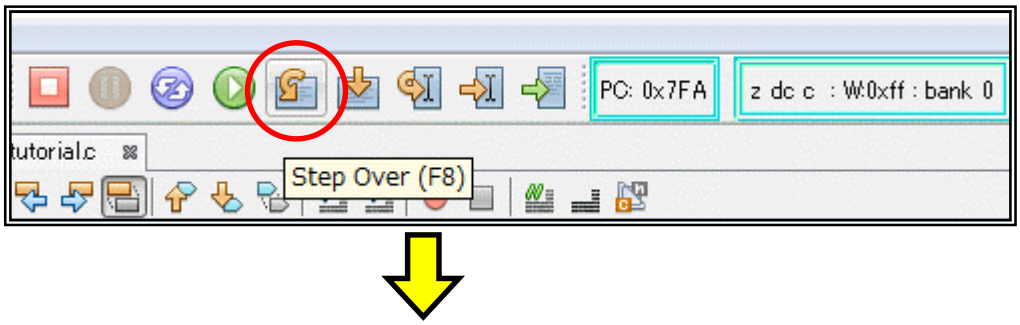


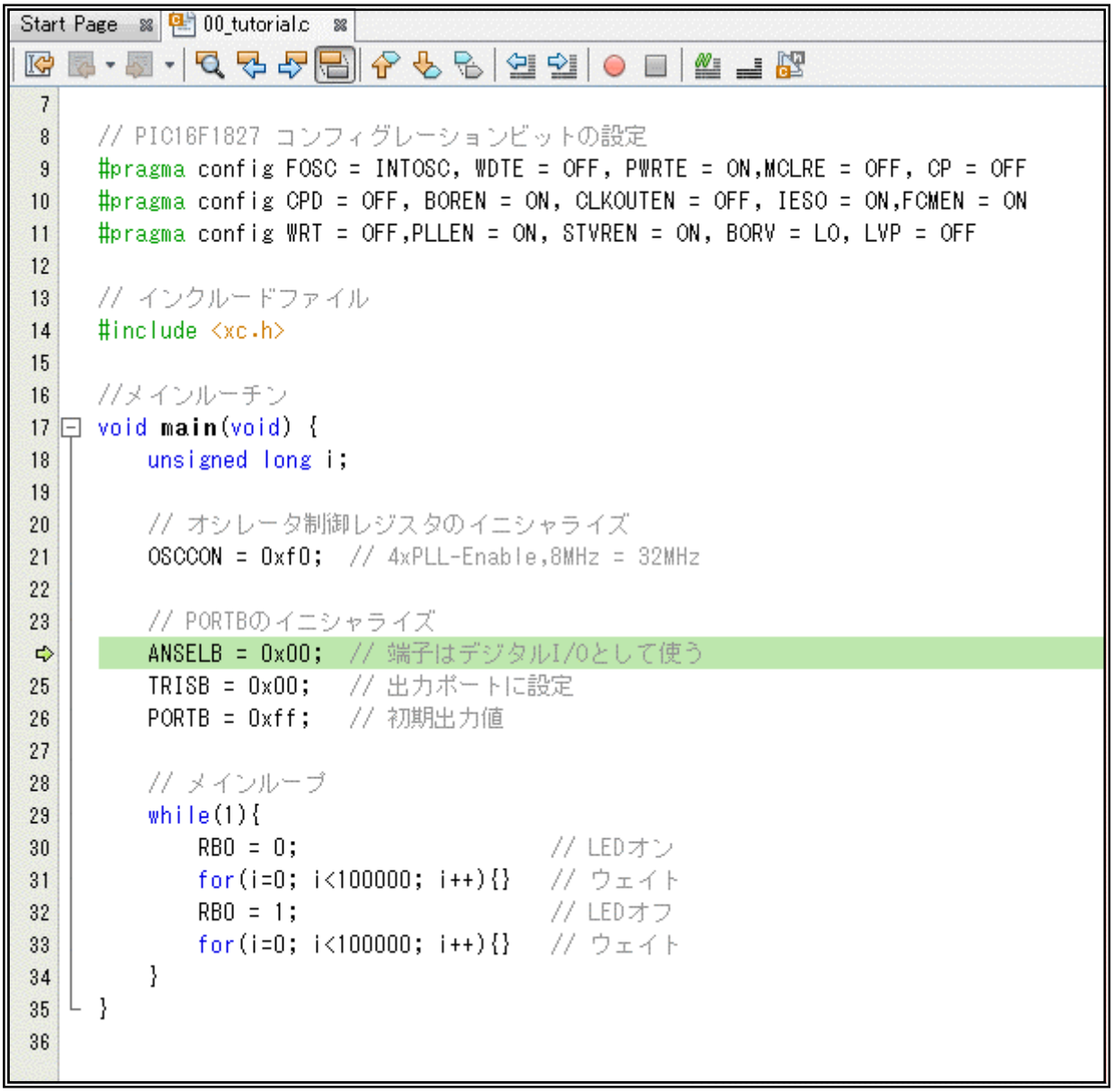
そうすると，スタートアドレスの行の色が変わり，左端に「→」が表示されます（下図の 21 行）。これが，現在のプログラムカウンタが示している行をあらわします。



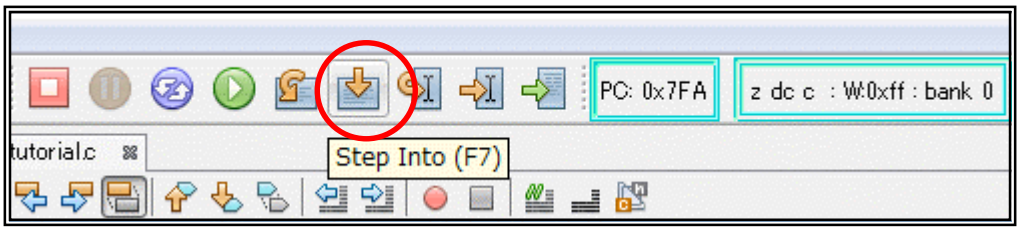
■ **ステップ実行**

デバッグの基本は一行ずつ実行してプログラムが設計どおり動いていくか確認することです。「Step Over」ボタンをクリックすると一行ずつ実行します。





なお，似たような機能に「Step Into」ボタンがあります。「Step Over」は関数をコールしている行を一行として実行しますが，「Step Into」は関数の中に入って一行ずつ実行します。状況に応じて使い分けましょう。



■ レジスタファイルの値の確認

デバッグの際に，レジスタファイルの値を確認したいことがあります。そのときはマウスカーソルを，値を見たいレジスタファイルのラベルに重ねてください。例えば，レジスタファイル「ABSELB」にマウスカーソルを重ねると，次のように表示されます。

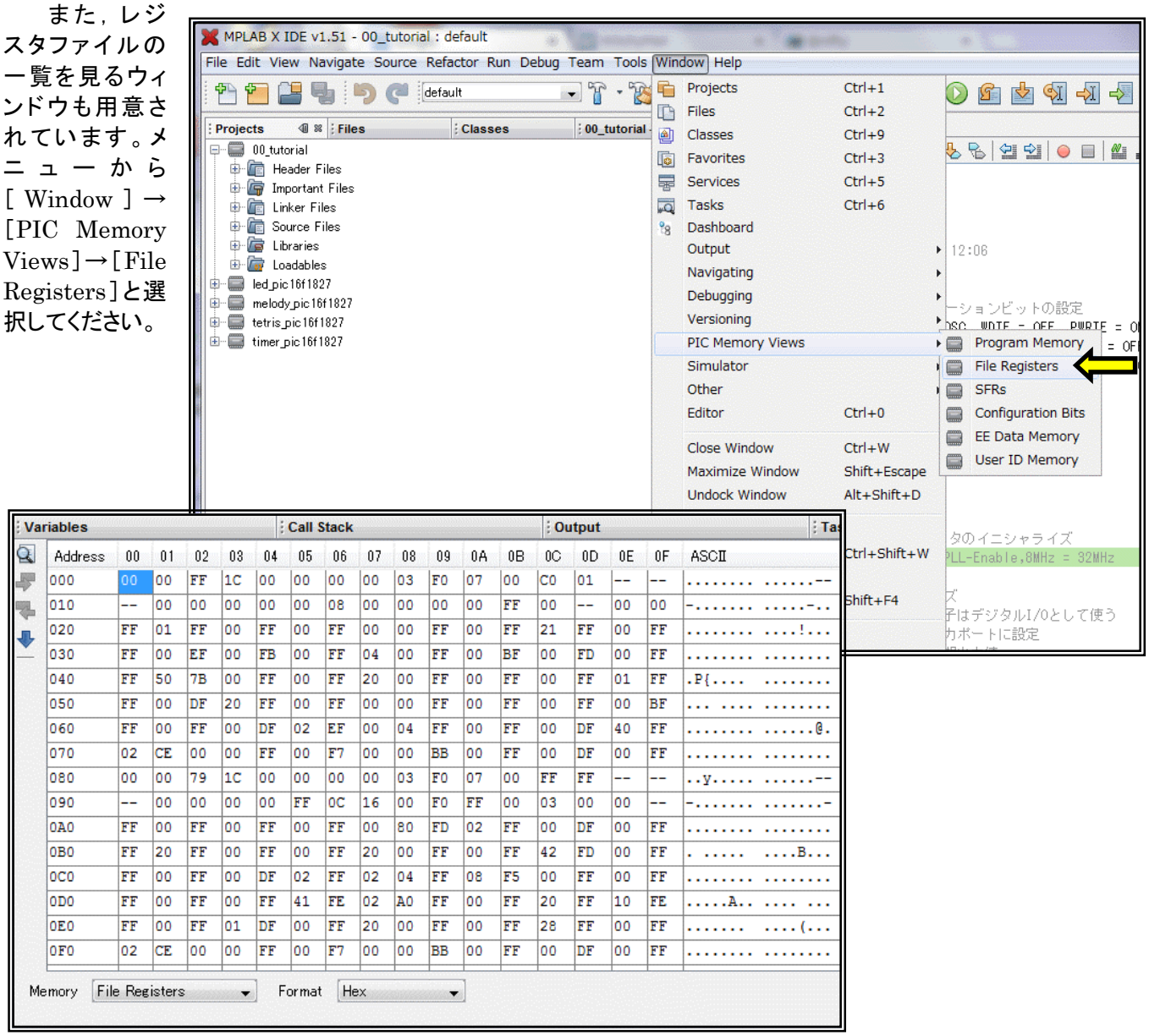
```
22
23      //  address = 0x18D,   ANSELB = 0xFE
24      ANSELB = 0x00;  // 端子はデジタルI/Oとして使う
25      TRISB = 0x00;   // 出力ポートに設定
```

「ANSELB」のアドレスが 0x18D 番地で，現在の値は 0xFE なのがわかります。ここで，「Step Over」ボタンをクリックして一行実行します。そのあと，同じように「ABSELB」にマウスカーソルを重ねると，次のように表示されます。

```
22
23      //  address = 0x18D,   ANSELB = 0x00
24      ANSELB = 0x00;  // 端子はデジタルI/Oとして使う
25      TRISB = 0x00;   // 出力ポートに設定
```

プログラムどおり，0x00 になったことがわかります。

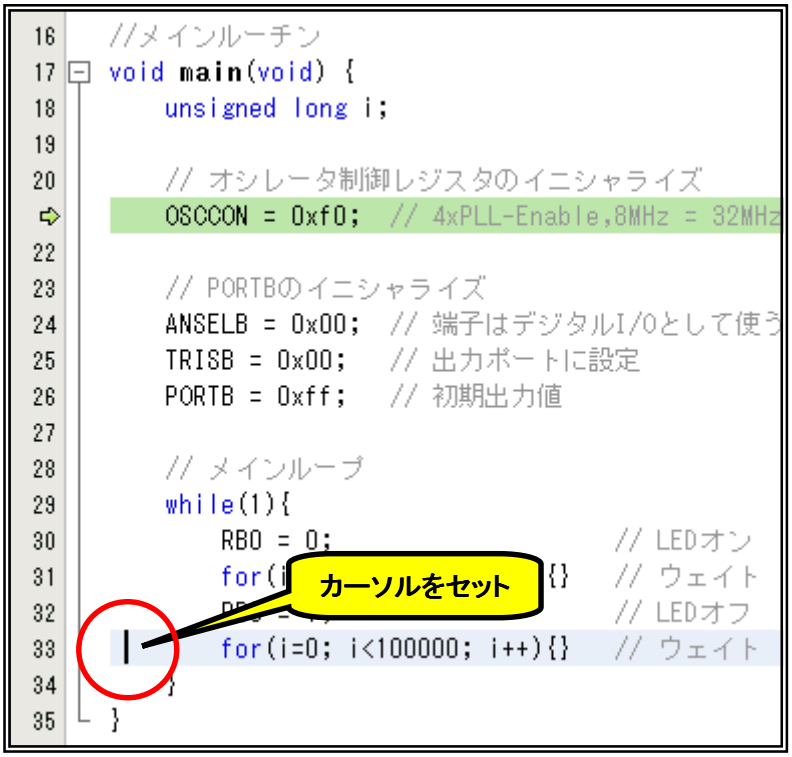
また，レジスタファイルの一覧を見るウィンドウも用意されています。メニューから［Window］→［PIC Memory Views］→［File Registers］と選択してください。

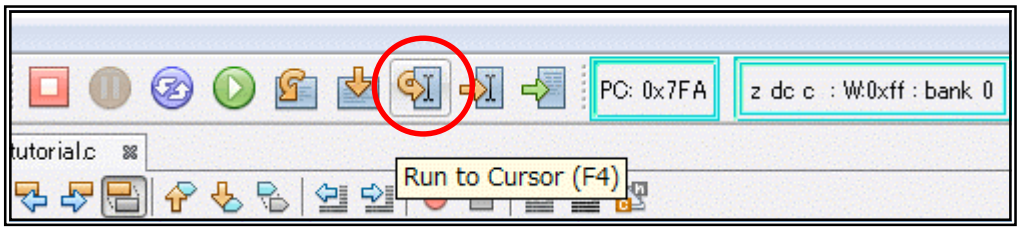


■ カーソル行まで実行

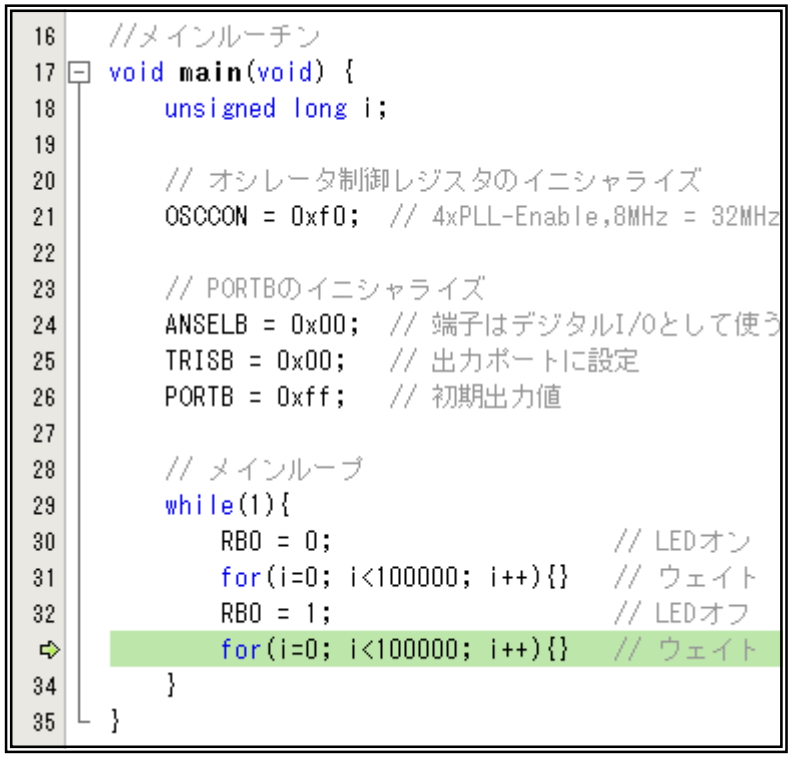
「Step Over」や「Step Into」を使えば一行ずつプログラムを実行することができますが，ある行までプログラムを実行してそこで止めたい，ということがあります。

右の例ではプログラムカウンタが 21 行にあります。メインループの最後の行，33 行まで実行してプログラムを止めたい場合，カーソルを 33 行にもっていって「Run to Cursor」ボタンをクリックします。





そうすると，プログラムカウンタの位置からスタートし，カーソルのある行まで実行すると停止します。

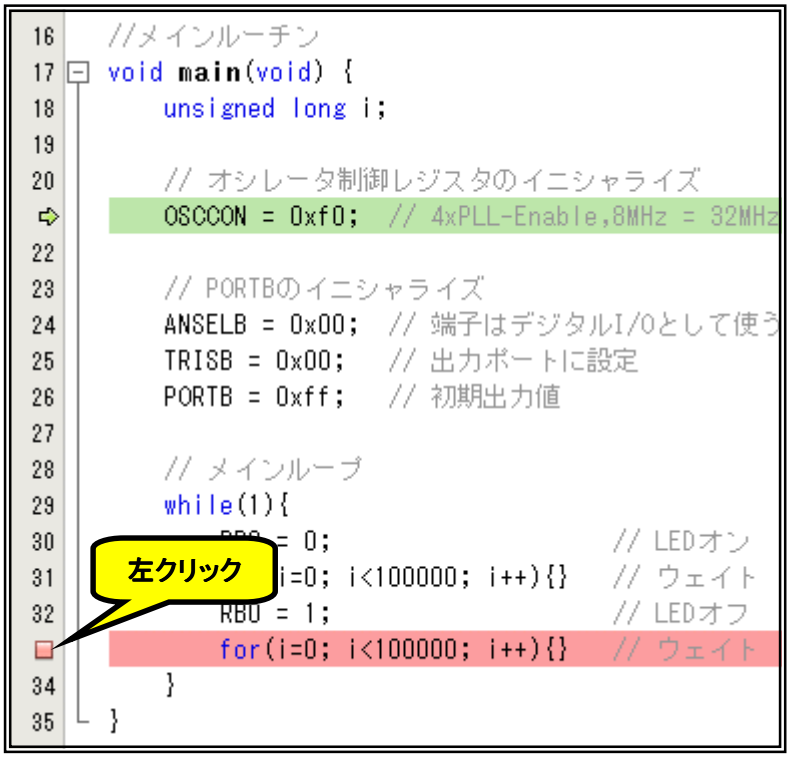


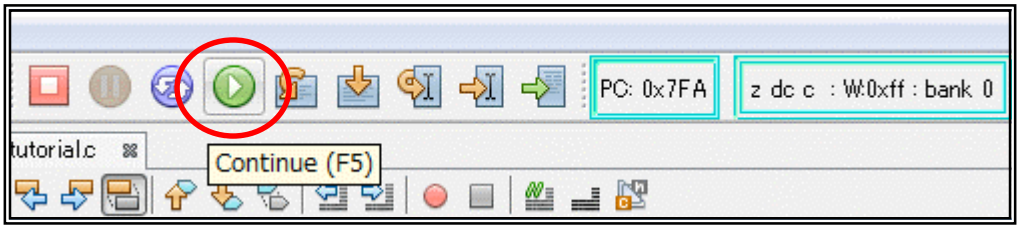
■ **ブレイク機能**

同じような機能で，ブレイク機能があります。プログラムの実行を止めたい行（ブレイクポイント）を登録し，プログラム実行中にその行に到達したら停止します。

右の例で，33 行で停止したい場合，「33」という数字のところで左クリックします。すると，33 行が赤くかわり，ブレイクが張られたことが明示されます。

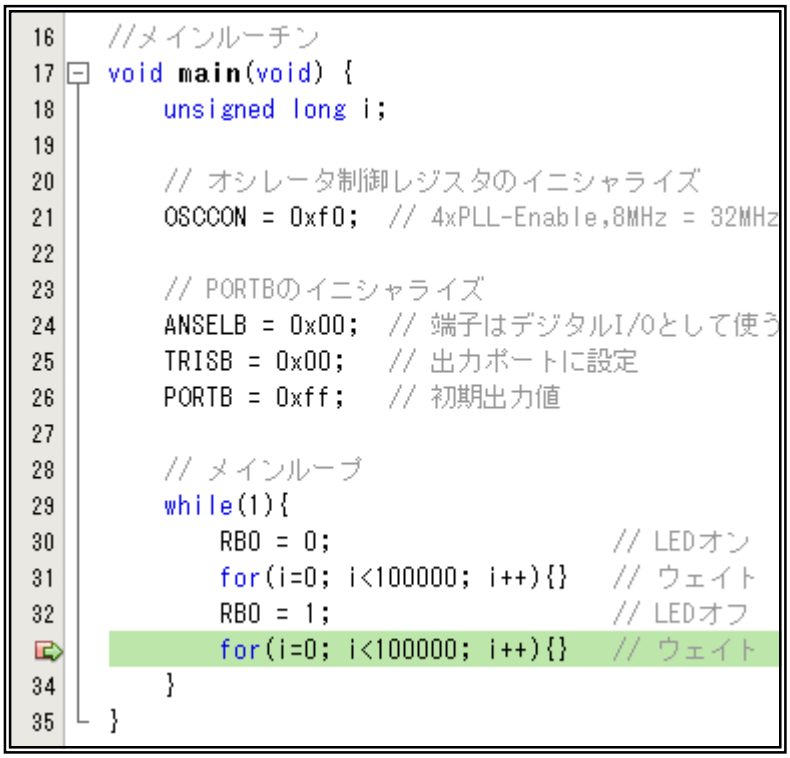
この状態で，「Continue」ボタンをクリックするとプログラムカウンタの行から実行します。





カーソル行まで実行すると停止します。

なお，ブレイクポイントは一つまで指定できます。また，ブレイクポイントが指定されている間は「Run to Cursor」機能は使用できません。

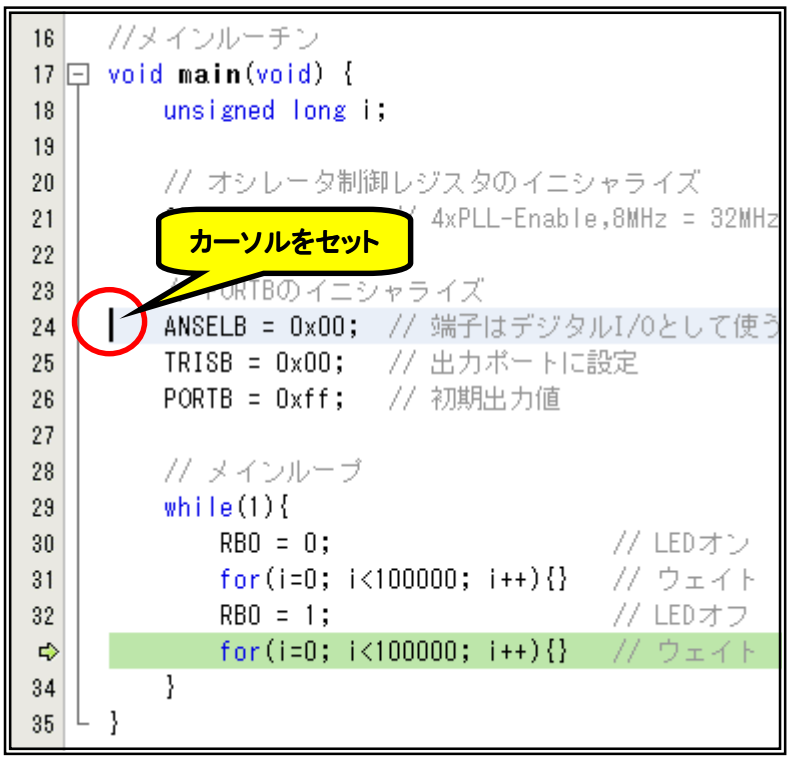


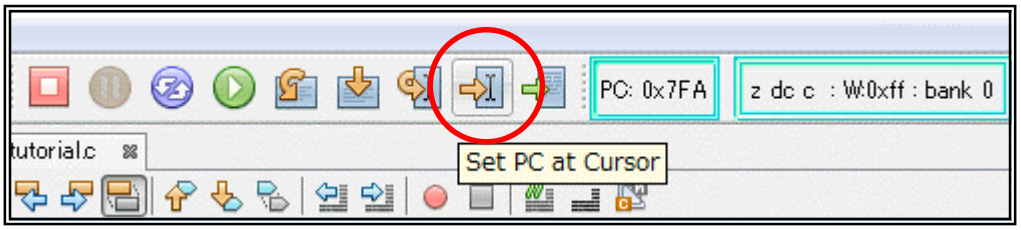
■ プログラムカウンタの指定

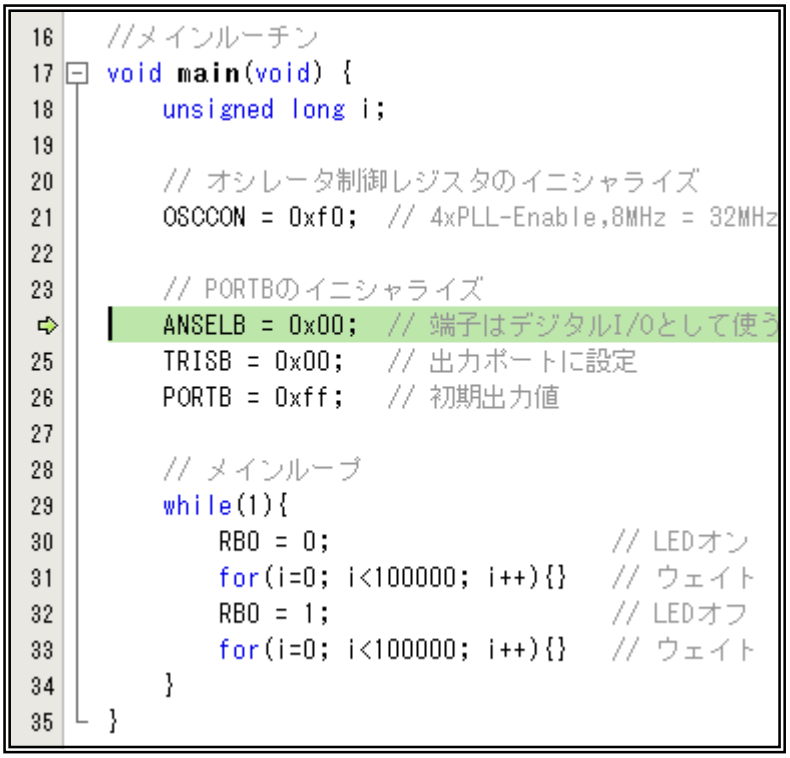
プログラムカウンタの位置を任意の場所に設定することができます。右の例では現在 33 行にプログラムカウンタがあります。これを 24 行に設定しましょう。

まずプログラムカウンタを設定したい 24 行にカーソルをあわせます。

この状態で「Set PC at Cursor」ボタンをクリックすると，プログラムカウンタがカーソル行にセットされます。





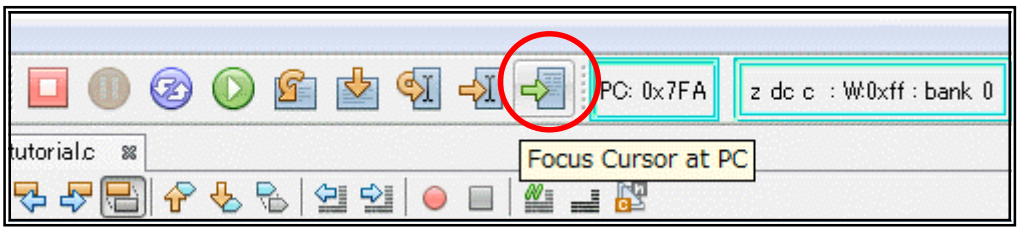


■ プログラムカウンタの位置を表示する

画面上でプログラムコードを読んでいくと，今プログラムカウンタがどこにあるのかわからなくなることがあります。そのようなときに，ワンクリックでプログラムカウンタのある行の画面に戻ることができます。

右の例ではプログラムカウンタの位置が画面に表示されていません。ここで，「Focus Cursor at PC」ボタンをクリックすると，プログラムカウンタが示している 33 行を含む形でソースリストが画面に表示されます。

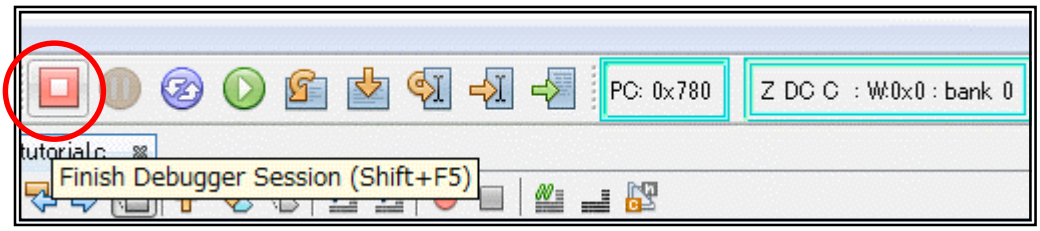
```
16   //メインルーチン
17   void main(void) {
18       unsigned long i;
19
20       // オシレータ制御レジスタのイニシャライズ
21       OSCCON = 0xf0;  // 4xPLL-Enable,8MHz = 32MHz
22
23       // PORTBのイニシャライズ
24       ANSELB = 0x00;  // 端子はデジタルI/Oとして使う
25       TRISB = 0x00;   // 出力ポートに設定
26       PORTB = 0xff;   // 初期出力値
27
28       // メインループ
29       while(1){
30           RB0 = 0;                    // LEDオン
```




```
16   //メインルーチン
17   void main(void) {
18       unsigned long i;
19
20       // オシレータ制御レジスタのイニシャライズ
21       OSCCON = 0xf0;  // 4xPLL-Enable,8MHz = 32MHz
22
23       // PORTBのイニシャライズ
24       ANSELB = 0x00;  // 端子はデジタルI/Oとして使う
25       TRISB = 0x00;   // 出力ポートに設定
26       PORTB = 0xff;   // 初期出力値
27
28       // メインループ
29       while(1){
30           RB0 = 0;                    // LEDオン
31           for(i=0; i<100000; i++){}   // ウェイト
32           RB0 = 1;                    // LEDオフ
             for(i=0; i<100000; i++){}   // ウェイト
34       }
35   }
```

■ デバッガの終了

デバッグ機能を終えて通常の MPLAB に戻るときは，「Finish Debugger Session」ボタンをクリックします。



なお，デバッグ時にダウンロードしたプログラムには，ユーザプログラムと一緒にデバッガ実行プログラムがダウンロードされており，リセット（電源オン）でデバッガ実行プログラムが起動します。そのため，このままだとユーザプログラムを起動することができません。それで，あらためてプログラムをビルドし「PICkit3」を使ってユーザプログラムのみをダウンロードしてください。

■ 「PICkit3」接続時の注意事項 ■

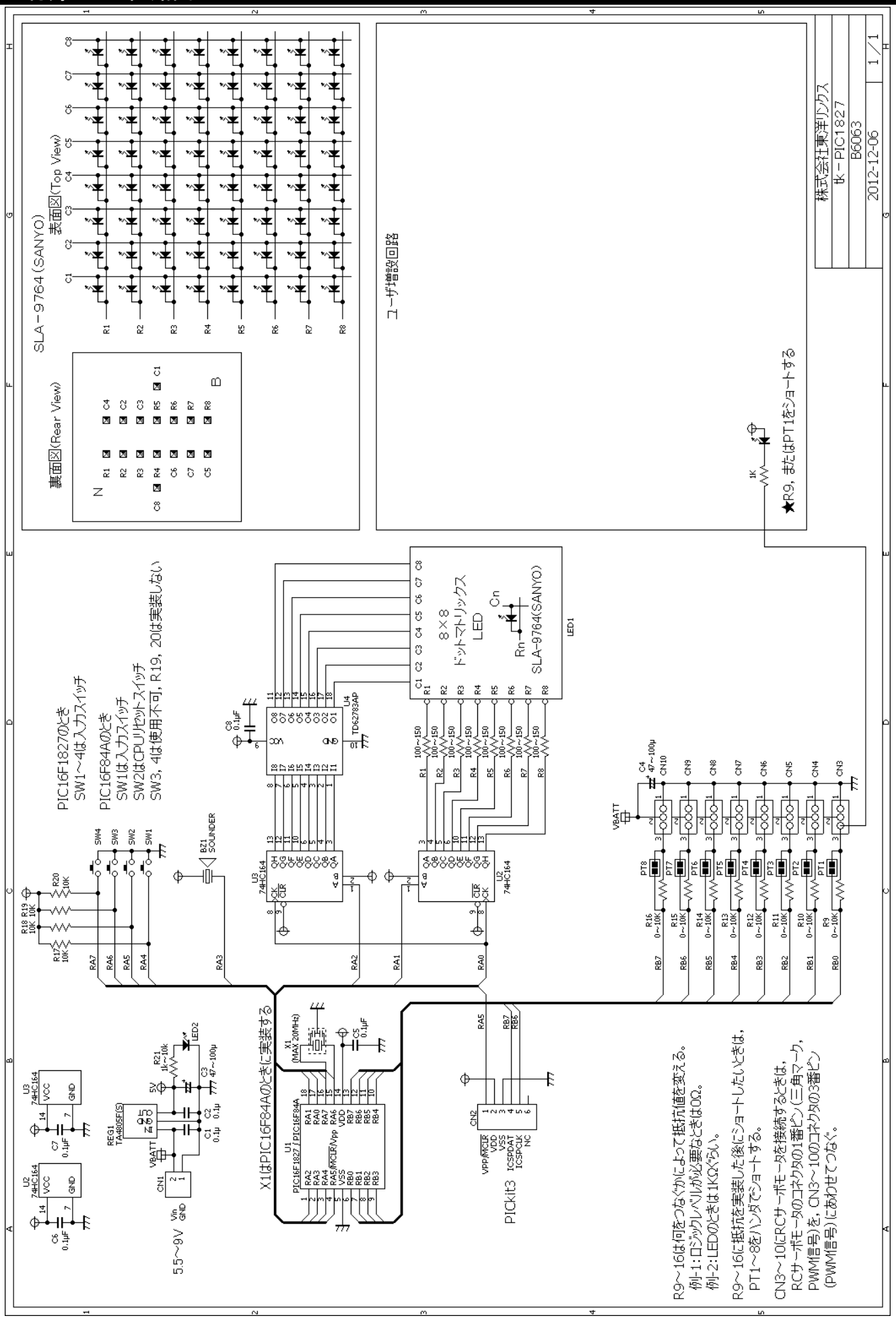
「PICkit3」は PIC16F1827 の RA5（MCLR/Vpp），RB6（ICSPCLK），RB7（ICSPDAT）を使って通信します。このマニュアルで取り上げたプログラム例ではこれらのポートを使用しないので問題ありませんが，これらのポートを使う回路を追加した場合デバッグできません。「PICkit3」を使ってデバッグするのであれば，回路設計時に RA5，RB6，RB7 は使用しないでください。

なお，回路設計時の注意事項の詳細は，マイクロチップテクノロジーの資料か，このマニュアルの付録をご覧ください。

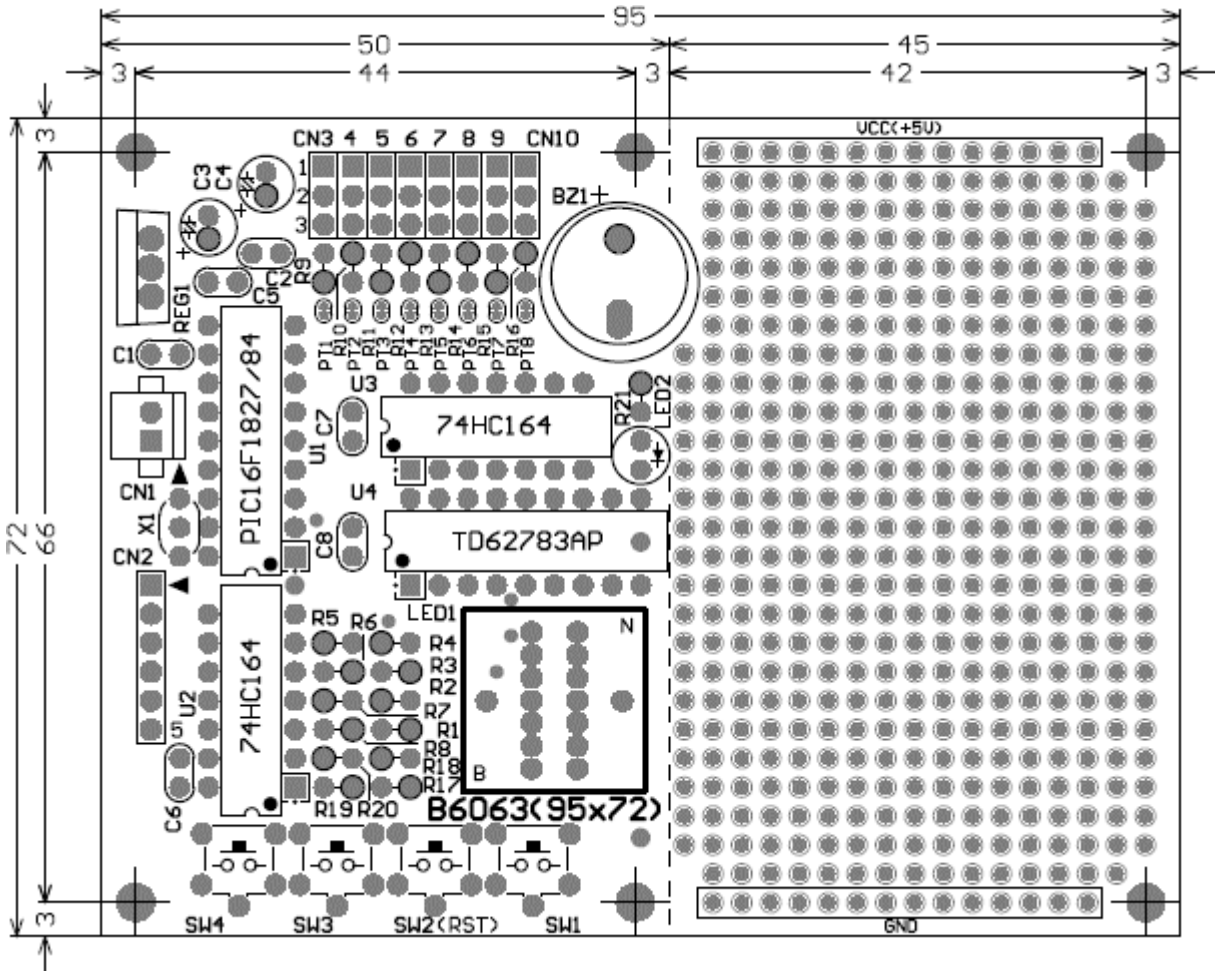
◆

このマニュアルではごく一部の機能しかご紹介していません。「MPLAB X IDE」，「MPLAB XC8」，「PICkit3」には，他にも便利な機能がたくさんあります。HELP ファイルや，マイクロチップテクノロジーの資料を見て活用してください。

付録-1 ： 回路図


SLA-9764(SANYO)
表面図(Top View)
裏面図(Rear View)
ユーザ増設回路
株式会社東洋リクス
tk-PIC1827
B6063
2012-12-06
1/1
PIC16F1827のとき
SW1~4は入力スイッチ
PIC16F84Aのとき
SW1は入力スイッチ
SW2はCPUリセットスイッチ
SW3, 4は使用不可, R19, 20は実装しない
X1はPIC16F84Aのときに実装する
★R9, またはPT1をショートする
8×8
ドットマトリックス
LED
SLA-9764(SANYO)
5.5~9V
PICkit3
R9~16は何をつなぐかによって抵抗値を変える。
例-1:ロジックレベルが必要なときは0Ω。
例-2:LEDのときは1KΩぐらい。
R9~16に抵抗を実装した後にショートしたいときは,
PT1~8をハンダでショートする。
CN3~10にRCサーボモータを接続するときは,
RCサーボモータのコネクタの1番ピン(三角マーク,
PWM信号)を, CN3~10のコネクタの3番ピン
(PWM信号)にあわせてつなぐ。

付録-2 ： シルク図（実装図）


95
50
45
3
44
3
42
3
72
66
3
3
CN3 4 5 6 7 8 9 CN10
VCC(+5V)
BZ1
C3
C4
REG1
C5
C1
CN1
X1
CN2
PIC16F1827/84
U1
C7
U3
74HC164
R21
LED2
U4
TD62783AP
C8
LED1
R5
R6
R4
R3
R2
R7
R1
R8
R18
R17
74HC164
U2
C6
R19
R20
B6063(95x72)
SW4
SW3
SW2(RST)
SW1
GND

付録-3 ： PICkit3 接続時の注意事項

回路とコネクタのピン配置

1ピン インジケータ

*デバッガとして使う場合、オシレータが必要です。
**デバイスにAVDDラインとAVSSラインがある場合、デバッガを使うには、両ラインを接続する必要があります。

ターゲット コネクタのピン配置

ピン	信号
1	MCLR/VPP
2	VDDターゲット
3	Vssグランド
4	PGD (ICSPDAT)
5	PGC (ICSPCLK)
6	未接続*

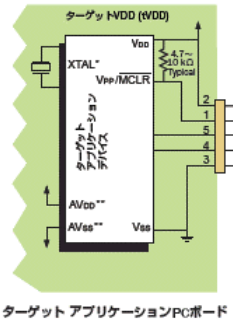
*将来用に予約済み

PICkit 3コネクタのピン配置

ピン	信号
1	MCLR/VPP
2	VDDターゲット
3	Vssグランド
4	PGD (ICSPDAT)
5	PGC (ICSPCLK)
6	未接続*

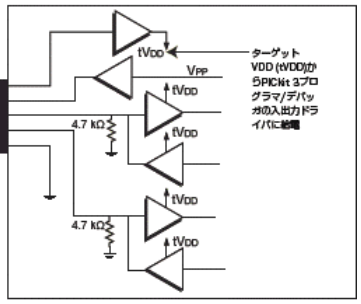
*将来用に予約済み

正しい設計



ターゲット アプリケーションPCボード

PICkit 3内部回路(略図)



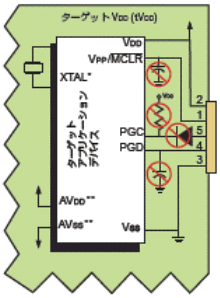
推奨設定

コンポーネント	設定
オシレータ	・ OSCビットを正しく設定 ・ 動作している事
電源	ターゲットが供給
WDT	無効(デバイスごとに異なる)
コード保護	無効
テーブル読み出し保護	無効
LVP	無効
BOD	VDD>BOD VDD min
JTAG	無効
AVDDとAVss	接続する必要あり
PGCx/PGDx	該当する場合、正しいチャンネルが選択されている事
プログラミング	VDD電圧がプログラミング仕様を満たす事

Note: 詳細は、PICkit 3オンラインヘルプを参照してください。

ターゲット回路設計上の注意

誤った設計



ターゲット アプリケーションPCボード

- **VDD に静電容量が 100 µF を超えるコンデンサを接続しないでください。**
  PICkit 3から電源を供給する場合、総負荷によってはターゲットに速やかに給電されません。
- **MCLR にコンデンサを接続しないでください。**
  コンデンサを接続すると、VPP が高速に遷移できません。
- **PGC/PGD にプルアップ抵抗を接続しないでください。**
  両ラインには MPLAB REAL ICE 内に 4.7 kΩのプルダウン抵抗が存在するため、プルアップ抵抗を接続すると分圧されます。
- **PGC/PGD に多重化しないでください。**
  両ラインは PICkit 3への通信専用です。
- **PGC/PGD にコンデンサを接続しないでください。**
  コンデンサを接続すると、プログラミングとデバッグの通信中にデータラインとクロックラインが高速に遷移できません。
- **PGC/PGD にダイオードを接続しないでください。**
  ダイオードを接続すると、PICkit 3とターゲットPIC® MCUの間で双方向通信ができなくなります。
- **推奨ケーブル長より長いケーブルを使わないでください。**
  最大ケーブル長については、PICkit 3 オンラインヘルプまたはユーザガイドの「Hardware Specification」を参照してください。

上記の資料は，マイクロチップテクノロジーの資料から抜粋しました。tk-PIC1827 は次のとおりこれらの条件を満たしています。

■ Vpp/MCLR(RA5)を 10KΩ程度でプルアップする。コンデンサを接続しない。

→R18(10KΩ)でプルアップしている。コンデンサは接続していない。**デバッガ使用時のアプリケーションでは SW2 を入力スイッチとしては使わないでください。**

■ PGC(ICSPCLK，RB6)/PGD(ICSPDAT，RB7)に，プルアップ抵抗，コンデンサ，ダイオードを接続しない。

→RB6とRB7はユーザに開放しているため未使用。**デバッガ使用時のアプリケーションでは RB6，RB7 は使わないでください。**

★部品が足りない場合やお問い合わせはメールか FAX でお願い致します。

（株）東洋リンクス

〒102-0093 東京都千代田区平河町 1-2-2 朝日ビル

TEL ： 03-3234-0559　FAX ： 03-3234-0549

URL ： http://www2.u-netsurf.ne.jp/~toyolinx

E-Mail ： toyolinx@va.u-netsurf.jp

※本書の内容は将来予告無しに変更することがあります（2013 年 2 月作成）

20130212