

3 「Download MPLAB X IDE」 ボタンをクリック

microchip.com/en-us/tools-resources/develop/mplab-x-ide#

MPLAB® X IDE

MPLAB® X Integrated Development Environment (IDE)

MPLAB X Integrated Development Environment (IDE) is an expandable, highly configurable software program that incorporates powerful tools to help you discover, configure, develop, debug and qualify embedded designs for most of our microcontrollers and digital signal controllers. MPLAB X IDE works seamlessly with the MPLAB development ecosystem of software and tools, many of which are completely free.

[Download MPLAB X IDE](#) **クリック**

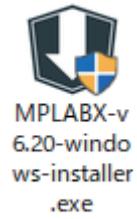
MPLAB X IDE brings a host of features to help you quickly debug your projects and minimize your development time. Some newer features include:

4 もしくは画面をスクロールして「MPLAB X IDE Windows」の「Download」ボタンをクリックします。

Please see the release notes for more details and information.

Title	Version Number	Date	
MPLAB X IDE (Windows)	28db195f... 7782 6.20	31 Jan 2024	Download クリック
MPLAB X IDE (Linux)	cecef0a... 7560 6.20	31 Jan 2024	Download
MPLAB X IDE (macOS)	18ba349f... a537 6.20	31 Jan 2024	Download
MPLAB X IDE Release Notes	6.20	31 Jan 2024	Download

- 5 インストーラーの「MPLAB-v6.20-windows-installer.exe」がダウンロードされていれば OK です。
※バージョンナンバーは 2024.02 現在



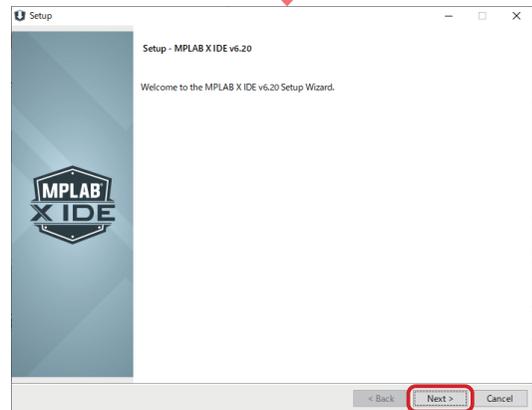
MPLAB X IDE のインストール手順

1 ダウンロードした MPLABX インストーラをダブルクリックして起動します。

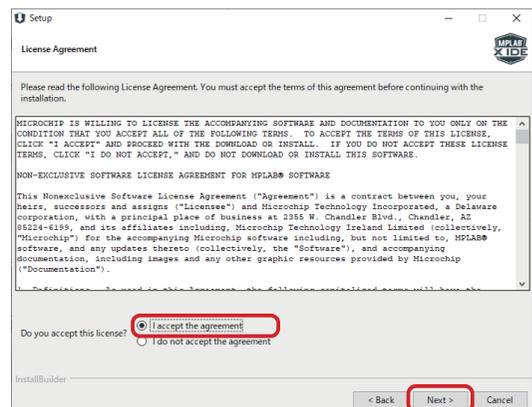
2 OS の「ユーザーアカウント制御」ダイアログが出る場合は「はい」をクリックします。



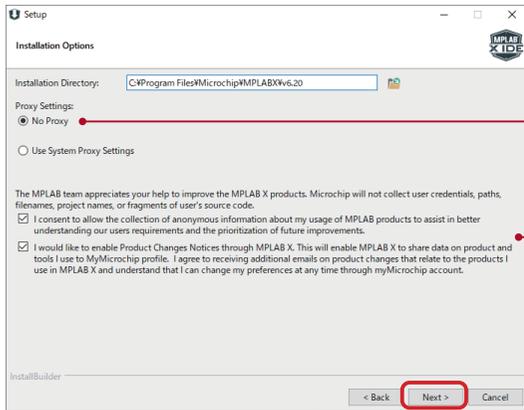
3 ウェルカム画面が表示されるので、「Next >」をクリックします。



4 License Agreement 画面
「I accept the agreement」をチェックして「Next >」をクリックして進めます。



次のページへ



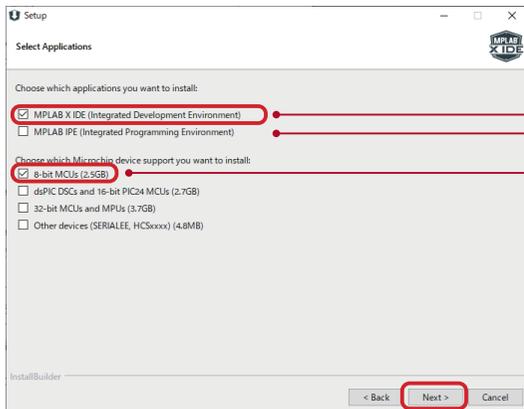
5 Installation Options 画面

インストール先を変更したい場合は、フォルダアイコンをクリックしディレクトリを指定します。通常は**デフォルト**で構いません。

PC のプロキシ環境が不明の場合は、**No Proxy** で構いません。

MPLAB X 開発用の情報収集に同意するかどうかは任意です。

「Next >」をクリックして進めます。



6 Select Applications 画面

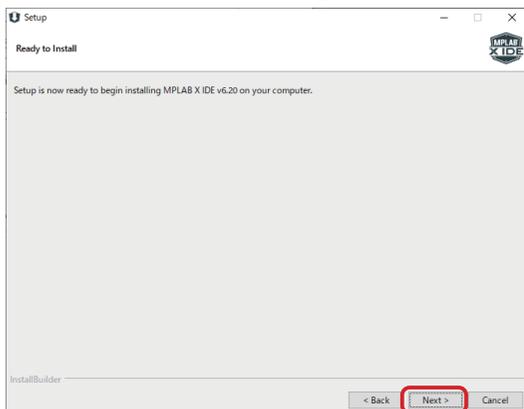
MPLAB X IDE は本書で使用するので**必須**です。

MPLAB X IPE は書き込み専用ソフトです。本書での解説はなく、**インストール不要**です。

デバイスの選択

本書で使用する「PIC16F627A」は 8bit マイコンです。他機種を使用しなければ、**8bit MCUs** だけで構いません。

「Next >」をクリックして進めます。



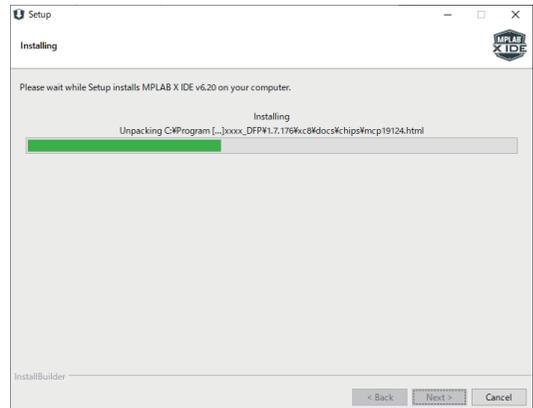
7 Ready to Install 画面

「Next >」をクリックして進めます。

次のページへ

8 Installing 画面

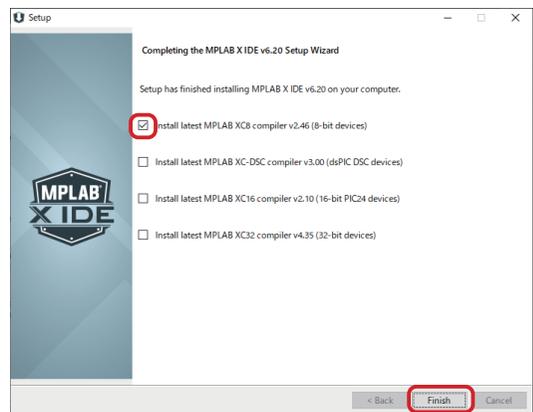
インストール中です。進捗バーが完了するまでお待ちください。



9 インストール完了画面

チェックボックスをチェックしておくと、XC コンパイラのダウンロードページが WEB ブラウザで開きます。続きは「MPLAB XC8 (C コンパイラ) のダウンロード手順」へ。

「Finish」をクリックすると MPLAB X のインストールは完了です。



MPLAB XC8 (C コンパイラ) のダウンロード手順

- 1 前ページの続きで自動でページが開けば、2 へ進んでください。

WEB ページが自動で開かない場合は、以下のリンクを WEB ブラウザで開いてください。



<https://www.microchip.com/en-us/development-tools-tools-and-software/mplab-xc-compilers>

- 2 「View Downloads」 ボタンをクリックします。

MPLAB® XC Compilers

Available as free, unrestricted-use downloads, our award-winning MPLAB® XC C Compilers are comprehensive solutions for your project's software development. Finding the right compiler to support your device is simple:

- MPLAB XC8 supports all 8-bit PIC® and AVR® microcontrollers (MCUs)
- MPLAB XC16 supports all 16-bit PIC MCUs and dsPIC® Digital Signal Controllers (DSCs)
- MPLAB XC32/32++ supports all 32-bit PIC and SAM MCUs and MPUs

The first versions of the MPLAB XC compilers that supported Catalina were:

- MPLAB XC8 - v2.20
- MPLAB XC16 - v1.50
- MPLAB XC32 - v2.41

[View Downloads](#) **クリック**

3 「Compiler Downloads」をクリックします。

Downloads, Documentation and Other Resources

Documentation **Compiler Downloads** Functional Safety Compiler Downloads Compiler FAQs

Documentation

- Installing and Licensing MPLAB XC C Compilers
- MPLAB XC HPA Renewal Instructions
- MPLAB XC License Server Manual
- Setting up the Site Licenses
- Quick Guide to Microchip Development Tools

XC8 Documents

- MPLAB XC8 C Compiler User Guide for PIC MCUs
- MPLAB XC8 C Compiler Legacy User Guide
- MPLAB XC8 C Compiler User's Guide for AVR MCUs
- MPLAB XC8 User's Guide for Embedded Engineers Code Example
- MPLAB XC8 User's Guide for Embedded Engineers

4 Windows 用の「MPLAB XC8 Compiler v2.32」をクリックします。

※バージョンナンバーは 2021.10 現在

Title	Date Published	Size
Windows (x86/x64)		
MPLAB® XC8 Compiler v2.32 SHA-256:4e38738bc2e19f27d4eb7469859a634951e4806cd8a5a8028459b2804d7455a4	2/18/2021	68.1 MB
MPLAB XC16 Compiler v1.70 SHA-256: 0a04017197d3086e652de08d74fa0d32f4cf7dc326a6faf70732833e938bd5c9	3/26/2021	102.5 MB
MPLAB XC32/32++ Compiler v3.01 SHA-256: 7f55ae8facb770de92f115fdd28b7cab81d12440efd1a8301dfa578f25b6de7a	6/2/2021	389.7 MB
Linux 32-Bit and Linux 64-Bit (Requires 32-Bit Compatibility Libraries)		

5 ダウンロードが完了したら、フォルダを開いてください。

Linux 32-Bit and Linux 64-Bit (Requires 32-Bit Compatibility Libraries)

MPLAB XC8 Compiler v2.32 SHA-256:57e52eab9...	2/18/2021	70.4 MB
MPLAB XC16 Compiler v1.70 SHA-256:...	3/26/2021	102.5 MB

xc8-v2.32-full-installer.exe

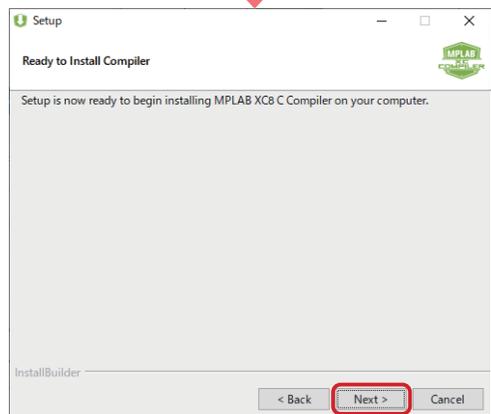
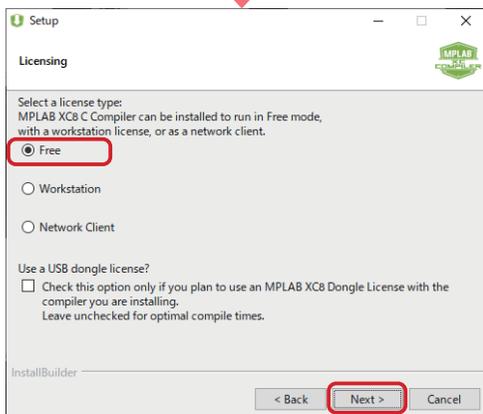
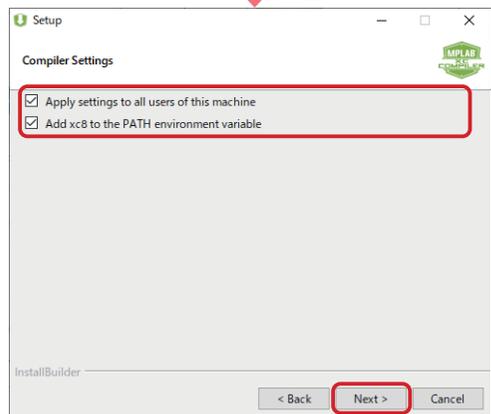
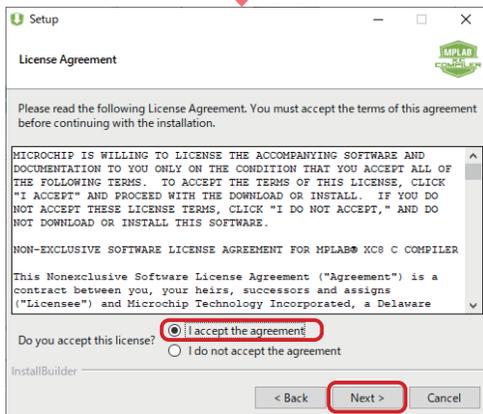
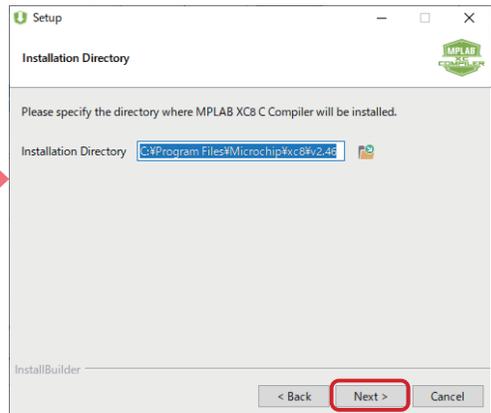
フォルダ内に「xc8-v2.32-full-install-windows-x64-installer.exe」があれば OK です。



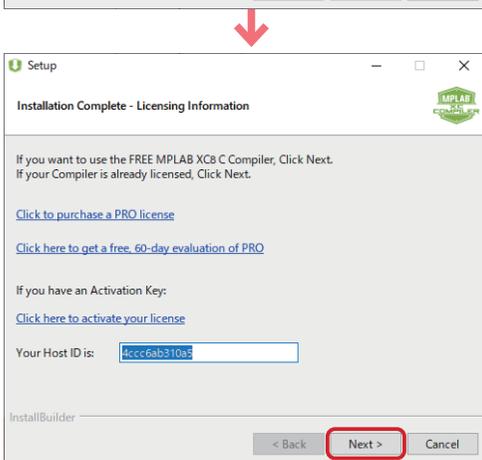
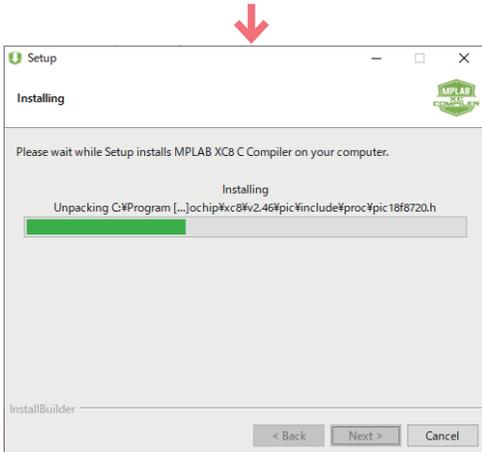
xc8-v2.32-full-install-windows-x64-install...

MPLAB XC8 (C コンパイラ) のインストール手順

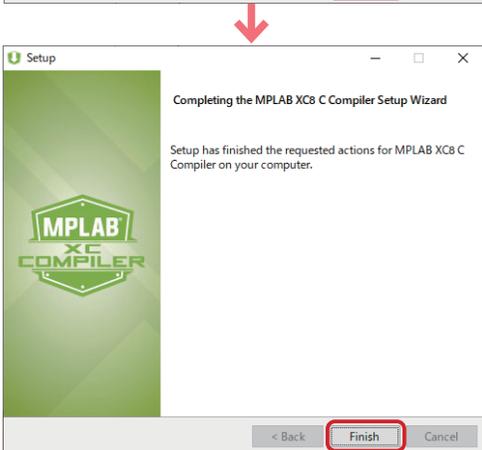
- 1 ダウンロードした XC8 インストーラをダブルクリックして起動します。
- 2 チェックの入っている箇所を確認しながら「Next >」をクリックして進めます。



次のページへ



FREE 版を使用するので、ライセンス登録不要です。
Next をクリックして進めます。



「Finish」をクリックしてウィンドウを閉じると
「MPLAB XC」のインストール完了です。

MPLAB を使用したプログラミング方法（コーディングからビルドまで）

1. 「MPLAB」の起動

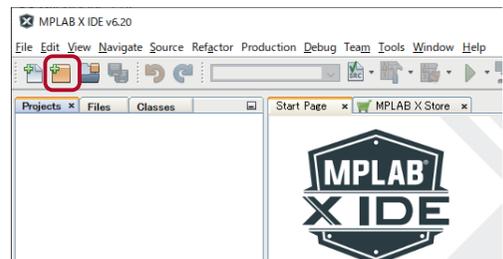
MPLAB インストール後、デスクトップに作成されたショートカットアイコン「MPLAB X IDE v*.**」をダブルクリックし起動します。
（v*.**はバージョン番号で、随時更新されます）



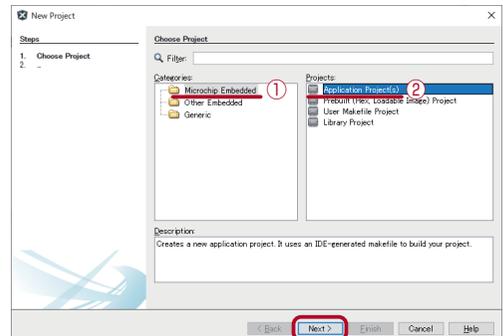
2. プロジェクトの作成

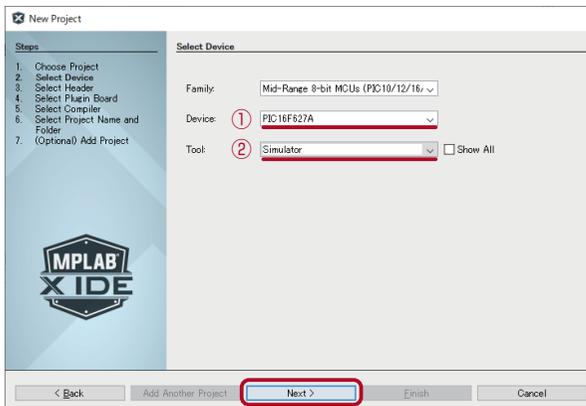
プロジェクトとは、プログラムを作成するためのファイル群です。
プロジェクトを作成していきましょう。

- 1 [New Project ...] アイコン  をクリックすると、プロジェクトウィザードが起動します。



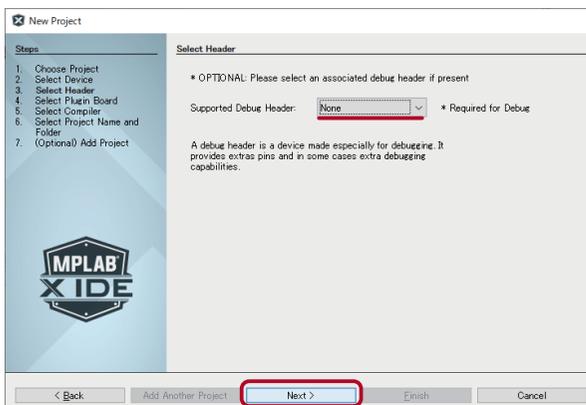
- 2
 - ① [Categories] で **Microchip Embedded** を
 - ② [Projects] で **Application Project(s)** を
選択し
[Next >] をクリックしてください。



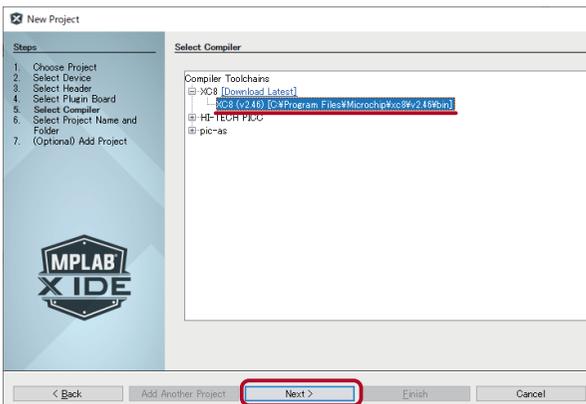


- 3 ① 使用する PIC の型を選びます。
[Device] 覧で「PIC16F627A」を選択してください。
[Device] 覧に PIC16F627A と キーボードから打って指定することもできます。

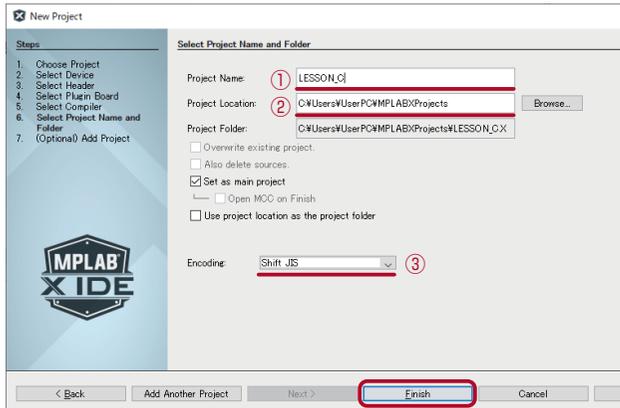
② [Tool] 覧は、お使いの PIC ライターがリスト内にあれば選択します。無い場合は [Simulator] を選択し、[NEXT >] をクリックしてください。



- 4 デバッグヘッダは本書では未使用です。
[None] を選択し、[NEXT >] をクリックしてください。

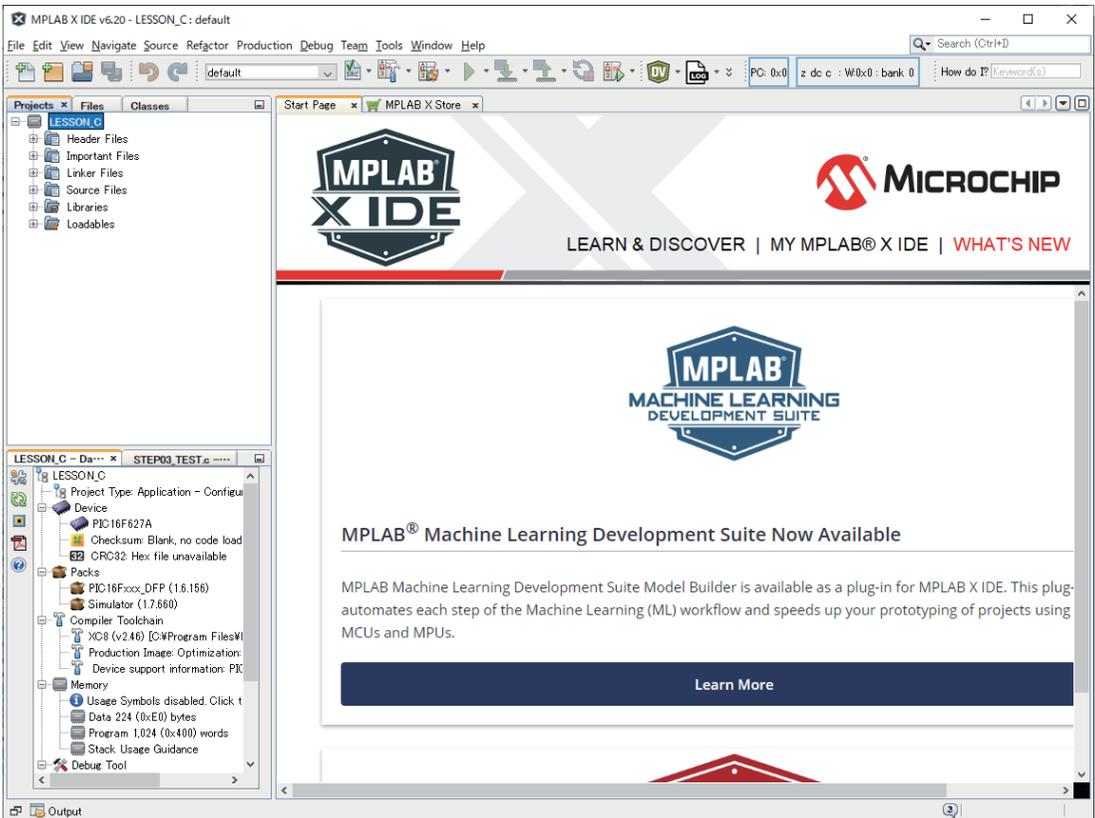


- 5 使用する C コンパイラを選びます。
[XC8] を選択し、[NEXT >] をクリックしてください。



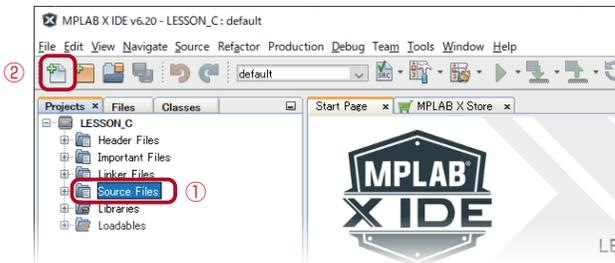
- 6
- ① プロジェクト名をつけます。
例では [LESSON_C] としています。
 - ② プロジェクトの保存先はデフォルトでは、C:\Users\ユーザー名 です。
[Browse...] をクリックして保存先を変更することもできます。
 - ③ [Shift_JIS] を選んでおけば、ソース内のコメント文で日本語が使えます。

[Finish] をクリックすると、プロジェクトが作成されます。

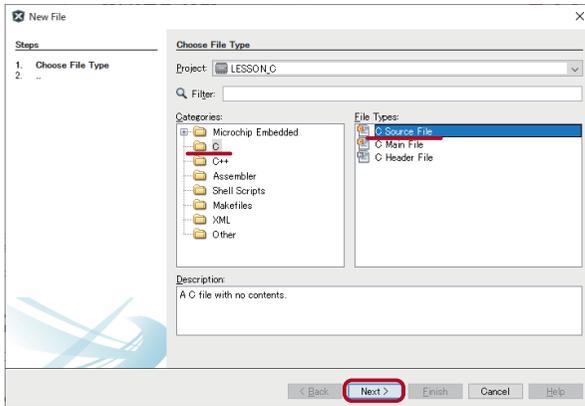


3. ソースファイルの作成

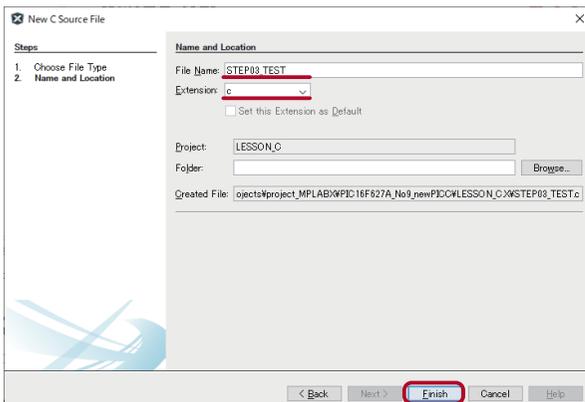
ソースファイルを作成しましょう。
ソースファイルにマイコンが行う処理を記述します。



- 1 ① ファイルペインの [Source Files] を選択し、
② [New File ...] アイコン  をクリックします。



- 2 [Categories] で [C] を
[File Types] で [C Source File] を選択し
[Next >] をクリックしてください。



- 3 ファイル名をつけます。
例では [STEP03_TEST] としています。

[Extension] は [c] を選択します。

[Finish] をクリックすると、空のソース
ファイルが作成されます。

- 4 弊社サイトに解答例ソースをご用意しています。以下の URL からダウンロードしてください。

<http://www.adwin.com/product/ECB-900T.html>

ダウンロードした zip ファイルを解凍すると、フォルダの中に「STEP03_TEST.c」というファイルがあります。テキストエディタ（メモ帳など）で開き、作成したソースファイルにコピー＆ペーストしてください。



注意

C 言語を記述する時は下記の点に注意してください。

- ・基本的に半角英数字を使用する
- ・大文字と小文字は異なる文字として扱われる
- ・空白は半角スペースもしくはタブを使用する

タスクペインが表示されていなくても、この後の「ビルド」操作を行なうと「output」ウィンドウが表示されます。メニューの Window > Output でも表示できます。ウィンドウはタブ部分をドラッグすれば位置を移動できます。ウィンドウ位置は、メニューの Window > Reset Windows で初期化して戻せるので色々動かしてみてください。

```

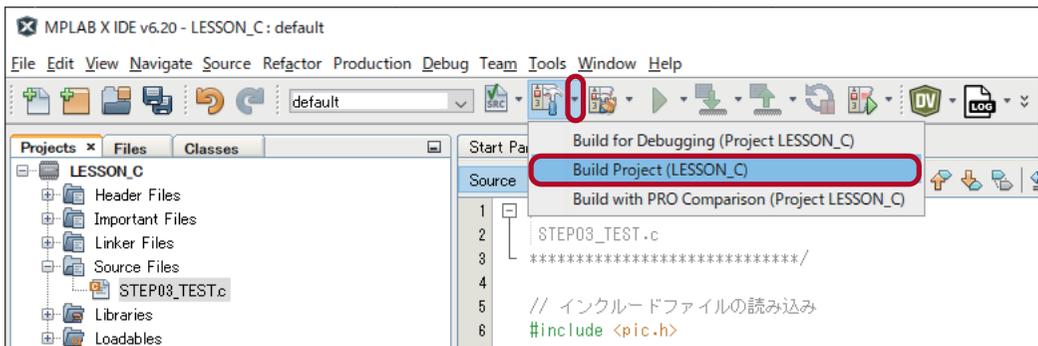
1 // STEP03_TEST.c
2 // *****
3 //
4 //
5 // インクルードファイルの読み込み
6 #include <pic.h>
7 //
8 // コンフィギュレーションワードの設定
9 #pragma config OPD=OFF, LVP=OFF, BOREN=ON, MCLR=ON, PWRTE=ON, WDTE=OFF, FOSC=INTOSC10
10 //
11 // メイン関数
12 void main(void)
13 {
14     // 1,2,17,18端子を入出力端子に設定
15     CMCON = 0x07;
16 //
17 // 電圧レベルの初期設定
18 PORTA = 0xFF;
19 PORTB = 0xFF;
20 //
21 // 入出力設定
22 TRISA = 0x21;
23 TRISB = 0x00;
24 //
25 // 永続ループ
26 while(1)
27 {
28     // SW1がONの場合
29     if (RAD == 0)
30     {
31         // LED1点灯
32         RBO = 0;
33     }
34     // その他の場合 (SW1がOFFの場合)
35     else
36     {
37         // LED1消灯
38         RBO = 1;
39     }
40 }

```

4. ビルド

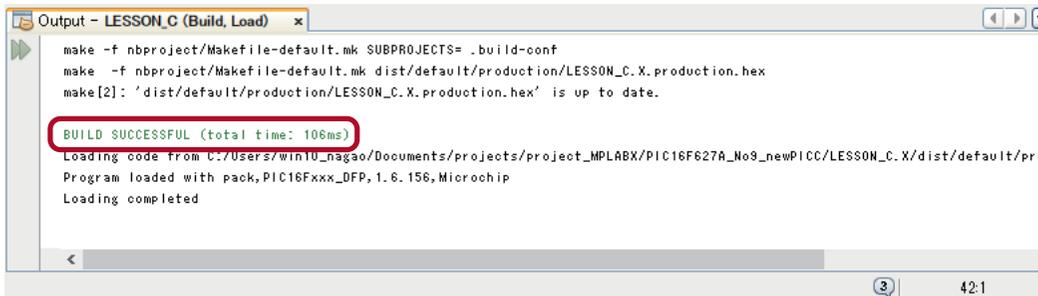
プロジェクトをビルドします。

- 1 メニューバーの Production > Build Project (プロジェクト名) をクリックするか、ツールバーの  の▼をクリックして表示される Build Project (プロジェクト名) を選択すると、ビルドが開始します。ビルドが終了すると「Output」ウインドウに結果が表示されます。



ビルドの結果：BUILD SUCCESSFUL (ビルド成功)

ソースファイルに間違いがなければ「Output」ウインドウに "BUILD SUCCESSFUL" と表示されます。



ビルドの結果：BUILD FAILED (ビルド失敗)

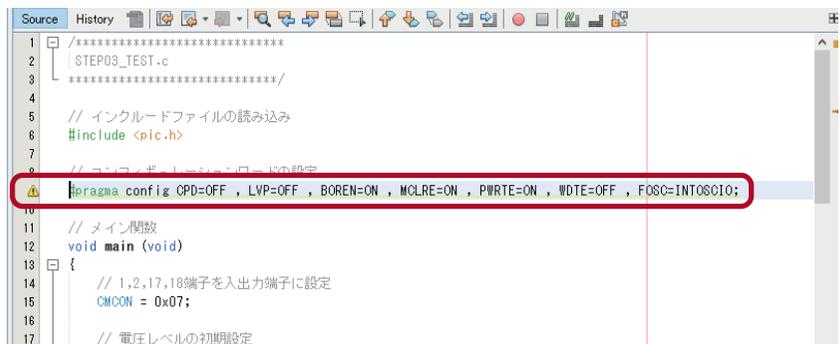
ソースファイルに間違いがあれば「Output」ウインドウに "BUILD FAILED" と表示されます。その場合、最初に起きたエラーからつぶしていきましょう。青文字リンクのエラー行をクリックしてください。



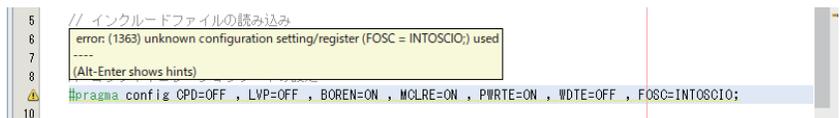
すると、ソースファイルの間違いがある箇所を示してくれます (下図赤枠)。

この場合、9 行目の記述に間違いがあることがわかります。

よく見てみると、行末に ; が付いておりこれがエラーの原因です。; は不要です。



カーソルを  に合わせると、エラーメッセージが表示されるので、これもエラー解決のヒントになります。



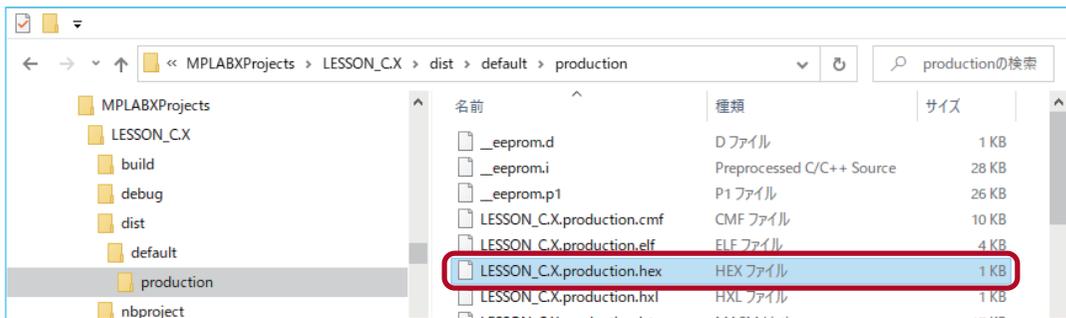
間違いを修正し、もう一度ビルドします。

エラーが無くなるまで、修正とビルドを繰り返します。

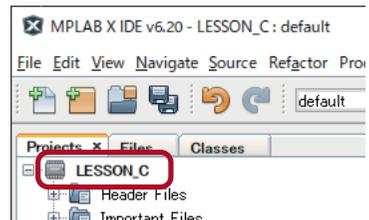
2 ビルドが成功すると、HEX ファイルが生成されます。

HEX ファイルの生成ディレクトリは、 ¥プロジェクト名.X¥ dist ¥ default ¥ production
 HEX ファイル名は、 プロジェクト名.X.production.hex です。

この HEX ファイルが、「実行ファイル」となります。

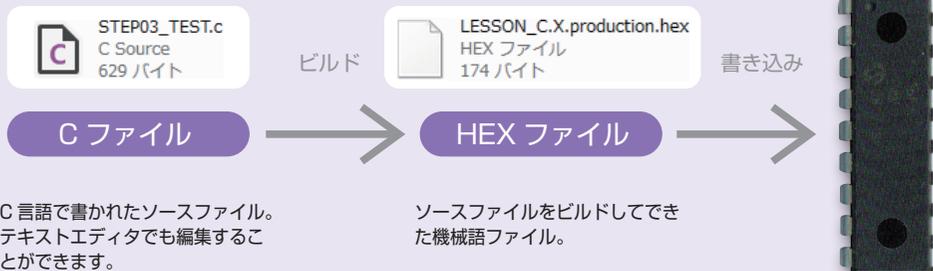


HEX ファイルは、MPLAB のプロジェクトを右クリックして「Export Hex」を選択すれば、任意のディレクトリに書き出すことができます。



ビルド

ビルドとは、C 言語で記述したソースファイルをマイコンが読める機械語に変換し、更にそのほかにも必要なファイルをつなぎ合わせて、実行ファイルを作成することです。



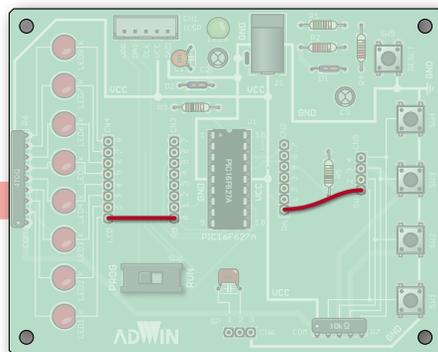
C 言語で書かれたソースファイル。テキストエディタでも編集することができます。

ソースファイルをビルドしてできた機械語ファイル。

c ファイル ,hex ファイルのアイコンはパソコン環境により異なります。

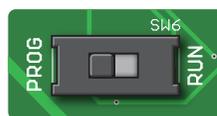
- 3 「STEP03_TEST.c」では、RBO と RAO を使うプログラムになっているので、右図のように実験基板を配線しましょう。
LED1 を RBO, SW1 を RAO に線材で接続します。

実体配線図



- 4 「LESSON_C.X.production.hex」を PIC に書き込みます。
書き込み方法は、ご使用になる PIC ライターのマニュアルを参照してください。
スライドスイッチは、弊社製品の PIC ライター「A+ タイプ」か「B タイプ」を使って、実験基板と接続して書き込む際に使います。

書き込み時に電源供給の必要はありませんが、AC アダプタは挿したままでも構いません。
プログラム書き込み時は、スライドスイッチのつまみを「PROG」側にし、書き込み後、プログラムを実行させたい時は「RUN」側にしてください。



書き込み時



実行時

プログラムを実行し、SW1 を押すと LED1 が点灯します。

PICKit を ICSP で使用するには

Microchip 社製 PIC ライター「PICKit」を使用するにはケーブルを自作する必要があります。
ケーブルを自作される場合は、お客様の責任の下行ってください。問題が起きても補償いたしかねますのでご了承ください。

端子の対応は右表のとおりです。
PICKit の 6 番ピンは、NC (ノンコネクション) で OK です。
実験基板に使用しているコネクタは JST 製です。
コネクタハウジング 型番：XHP-5
コンタクトピン 型番：BXH-001T-P0.6
その他の必要な周辺回路については、お使いの PICKit のマニュアルをご参照ください。

PICprogrammer		PICKit 2 or 3
VPP	↔	VPP/MCLR
DAT	↔	PGD/ICSPDAT
CLK	↔	PGC/ICSPCLK
VCC	↔	VDD
GND	↔	VSS/GND

【ご注意】 PIC Programmer の書き込みエラー

弊社製 PIC ライター「PIC Programmer」をご購入いただければ、付属のマニュアルに沿ってお使いいただけますが、実験基板の裏に記した型番が「ADW-076」の場合、書き込み不良が起きることがあります。

Picpro の [Program] ボタンを押すと、

「The programmer does not recognise the [16F627A] processor Continue anyway?」

とメッセージが出て書き込みが失敗する場合、回避策として以下の方法をお試しください。

実験基板の Reset ボタンを押したまま PICpro アプリの「Program」ボタンをクリック。

書き込み完了ダイアログが出るまで Reset ボタンは押しっぱなしにする。

お手数をお掛けして申し訳ございませんが、何卒よろしくお願いいたします。