1601 MPL

Transitioning to MPLAB® X IDE

For users of MPLAB IDE 8

Lab Manual

Rob Ostapiuk
Microchip Technology Inc.

MICROCHIP
Technical Training

v3.0 March 2012

Transitioning to MPLAB® X IDE
For users of MPLAB IDE 8

Table of Contents

Lab Exercise 1: How to create a new standalone project 1-1
Lab Exercise 2: How to convert a legacy project to the MPLAB® X format 2-1
Appendix A: MPLAB® X IDE Quick Reference Guide A-1
Section 1: Managing Projects
1.1 How to open a project..... A-3
1.2 How to add existing files to a project A-4
1.3 How to create new files in a project A-5
1.4 How to remove a file from a project..... A-7
1.5 How to permanently delete a file A-7
1.6 How to save a file or project A-7
1.7 How to close a project A-8
1.8 How to modify project settings *(choose debug tool, build tool, etc.)* A-8
Section 2: Building Projects
2.1 How to build a project A-9
2.2 How to build and run a project with a debugger A-9
2.3 How to build and run a project without a debugger A-10
Section 3: Debugging Projects
3.1 How to set or change the debugger A-11
3.2 How to control program execution when debugging A-12
3.3 How to set and clear breakpoints A-12
3.4 How to use the Stopwatch..... A-13
3.5 How to display and use Watches A-14
3.6 How to view Embedded Memory *(SFRs, RAM, Flash, EEPROM and Configuration bits)* A-15

Lab Exercise 1

How to create a new Standalone Project

Purpose

This exercise will walk you through the steps required to create a new Standalone Project in MPLAB® X IDE. A Standalone Project is the most common kind of project which is used to create a complete application from source code and libraries that can "stand on its own", requiring nothing else to execute on a PIC® microcontroller. (Contrast this to library projects and prebuilt/hex file projects). Upon completion of this exercise, you will possess the basic skills necessary to create a project that may be used as a framework for more advanced development tasks.

Objective

The project you create will use the MPLAB X IDE's project wizard to collect the information required to generate the project files. You will then add some existing source files to the project, build it, program the target device and run the application. If everything is done correctly, the project will build without error and you will see the text "MPLAB X" on the LCD display and LEDs D6 and D7 will blink. Additionally, a number will appear on the second line of the LCD representing the LED blink delay. Pushing switches S4 and S3 will increment or decrement the blink delay respectively (you may need to alter the delay by a significant amount to notice a difference).

Tools

- MPLAB X IDE 1.0+
- MPLAB C30 3.31
  Lite/Eval version works

If not using hardware:

- Proteus VSM Viewer Plug-in

If using hardware:

- REAL ICE™ or MPLAB ICD 3
- Explorer 16
- PIC24FJ128GA010 PIM

Procedure

Launch MPLAB® X IDE by double clicking its icon on the desktop or selecting its entry in the Start/Applications menu.

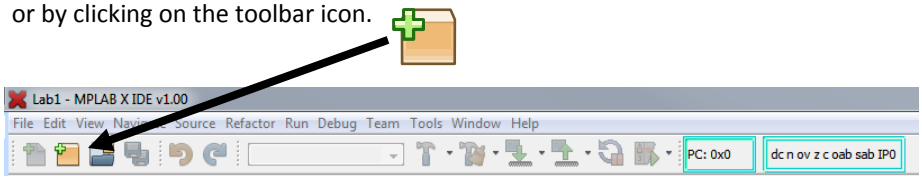
MPLAB X IDE

Information

You must use the wizard in MPLAB X IDE. A new project may not be created manually as in earlier versions.

1 Start the new project wizard

Launch the new project wizard by selecting from the menu: **File** ▸ **New Project...** or by clicking on the toolbar icon.



2 Choose Project Type

Select **Microchip Embedded** under **Categories** and select **Standalone Project** under **Projects**. Click **Next >** when done.

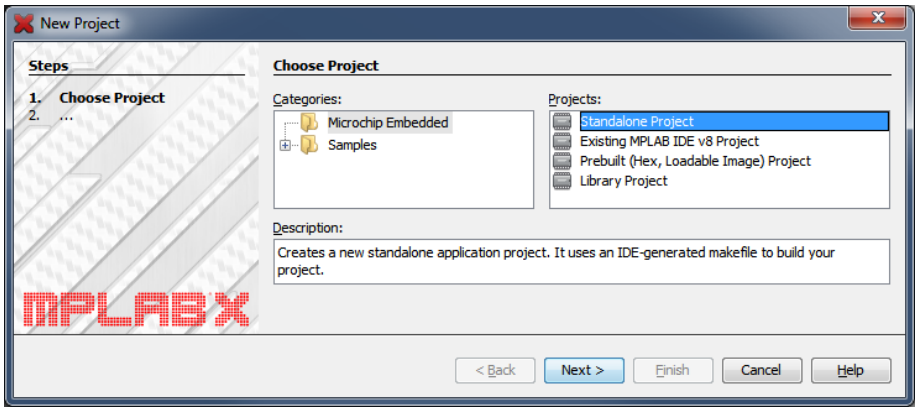


Figure 1.1
Choose Project step of the New Project wizard

The Categories list contains two options. Selecting one will change the items in the Projects list. The Samples category simply provides you with a list of sample projects you can use to test drive MPLAB X IDE without having to create a project from scratch or to convert an existing project of your own.
For this exercise, we will be using the Microchip Embedded category. This provides us with four project options.

1. **Standalone Project:** A regular project that will contain one or more source files and will be built into a single hex file for programming onto a PIC® Microcontroller. This is the one we will use for this exercise.
2. **Existing MPLAB IDE v8 Project**: This will prompt you for an MPLAB 8 *.mcp file and will ask you a few questions about the device and tools. We will cover this option in Lab 2.
3. **Prebuilt Project**: In MPLAB IDE v8, hex files could be imported without having a project to contain them. In MPLAB X IDE, you must have a project for everything, even an imported hex file. The steps to bring in a hex file are no more numerous in this version. The wizard will create a project "wrapper" around the hex file as it asks you for the very same information that you had to provide manually in version 8. For more information on creating prebuilt pro-

jects, please see: http://microchip.wikidot.com/mplab:how-to-create-a-prebuilt-project

4. **Library Project**: In MPLAB IDE v8, you built a library by creating an ordinary project and then selecting a library as a target in the linker options. In MPLAB X IDE, you create a library project from the very beginning. For more information about creating libraries, please see: http://microchip.wikidot.com/mplab:how-to-create-a-library-project

3 Select Device

For the purposes of this exercise, chose **16-bit MCUs (PIC24)** from the **Family** combo box and **PIC24FJ128GA010** from the **Device** combo box. Click **Next >** when done.

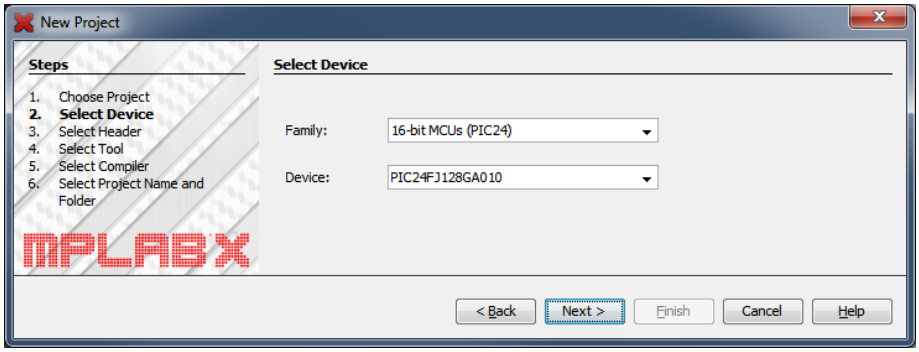
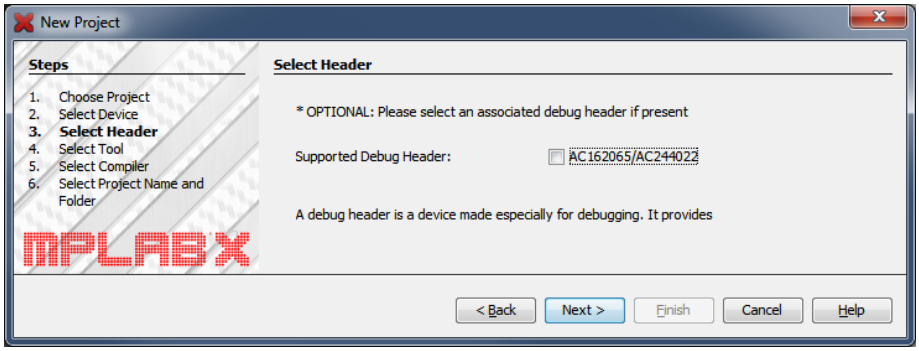


Figure 1.2
Select Device step of the New Project wizard

4 Select Header

No headers are being used with this device. Click **Next >** to advance to the next step.



Some devices use small header boards with a special MCU for debugging. This special MCU has extra pins for MPLAB REAL ICE communication and therefore allows the use of all pins on the part for the application. The header board is not used or needed for programming. However, when debugging very low pin-count devices the header must be used while it is optional for high pin-count devices like the PIC24FJ128GA010.

Figure 1.4
An example of some header boards for an 8-pin device and a 64-pin device

Tips and Tricks

Recently Used Devices

After using MPLAB X IDE for the first time, it will keep track of the devices you have selected. If you select **Recently Used** in the Family box, you will be presented with a short list of the devices you have used most recently.

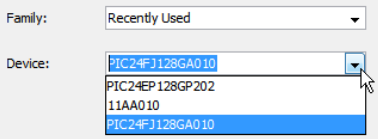


Figure 1.3
Select Header step of the New Project wizard

5 Select Tool

Choose your development tool:

A. If attending the class online or you have no hardware tools available, choose the **Proteus VSM Viewer** (see figure 1.5).

B. If attending the class in-person, you may be using either a **Real ICE** or **ICD 3**. ***Choose the serial number*** immediately below the tool you are using (see figure 1.6 for an example with a Real ICE).

Click Next > when done.

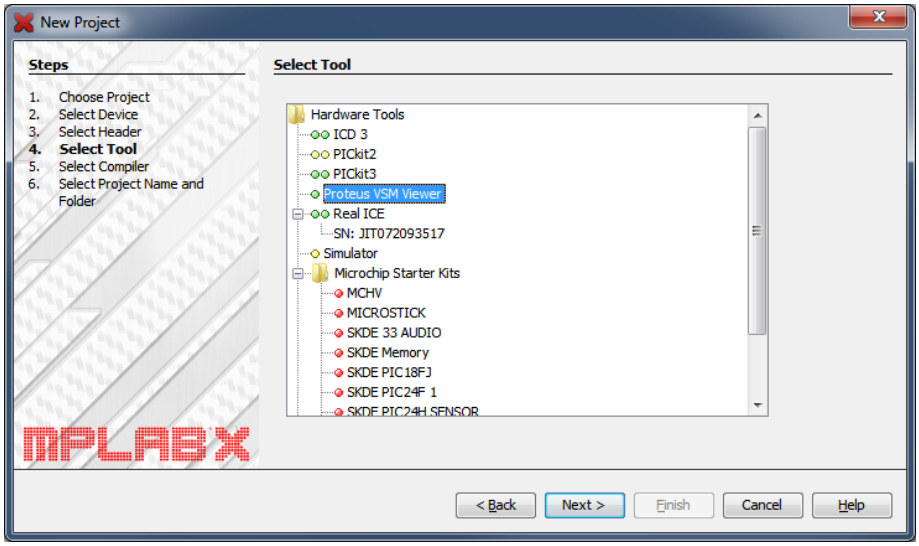


Figure 1.5
Select Tool step of the New Project wizard (Using Proteus VSM Viewer)

Information

Tool support for the project's selected PIC® Microcontroller

Programmer Support
Debugger Support

Production Tested
Beta Support
Not Supported

When you work with a hardware debugger such as the REAL ICE, ICD3 or PICkit3, be sure to click on the corresponding serial number below the tool's name. If you click the name instead, the IDE will not communicate with the desired tool, but will still allow you to complete the creation of the project. It does this so that you can create a project that will eventually use one of these tools even if they aren't currently connected to the system.

Selecting the serial number is important because this allows multiple tools to be connected simultaneously, each assigned to a different, open project.

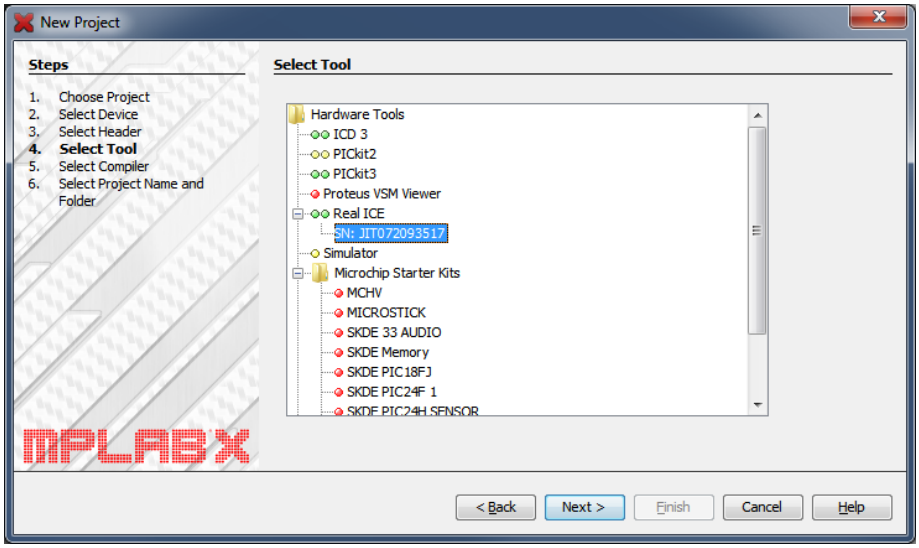


Figure 1.6
Select Tool step of the New Project wizard (Using MPLAB Real ICE)

Information

If you plug in a tool with a different serial number from the one assigned to the project, but it is the only one connected to the system, MPLAB X IDE will ask you if you want to use the currently connected tool instead.

6 Select Compiler

Select the specific version of the compiler you wish to use (the one given to you for this class). Figure 1.7 shows MPLAB C30 version 3.31 selected.

Click Next > when done.

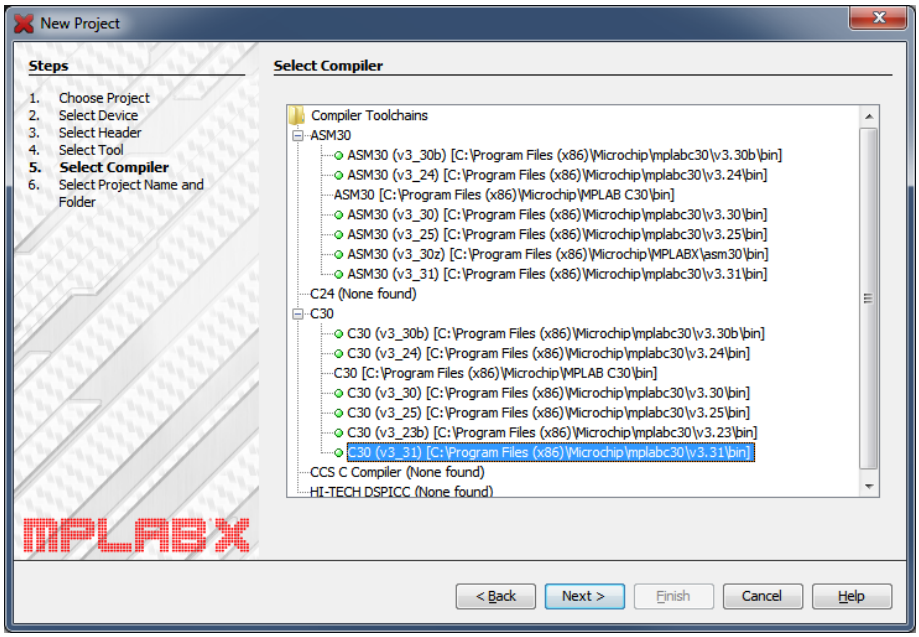


Figure 1.7
Select Compiler step of the New Project wizard

Information

The debug tool and build tool are stored with the project properties in MPLAB X IDE.

MPLAB X IDE supports the installation of multiple versions of the same compiler as can be seen above in Figure 1.7. Each node under the "C30" is a different version of the MPLAB C30 compiler. Each project may have a different compiler version assigned to it and the version may be changed at any time in the project's properties. This makes it easy to keep legacy versions installed for maintaining old projects while using the latest version for a new project. Our compilers are now installed into directories that include the version number as part of their path, so old and new versions can happily coexist on the same machine.

7 Select Project name and Location

For the purposes of this lab, we will be specifying the name of an existing folder which contains the source code you will use shortly. Normally, you can name the project anything you want and place it anywhere you like. However, in this case it is very important that you enter the name EXACTLY as shown to create the project in the same folder as the existing source code. This eliminates the need to copy files in your file manager. (See figure 1.8 for screenshot)

For the project name, enter "**Lab1**" without any spaces.
For the project location, enter "**C:\MASTERS\1601**".

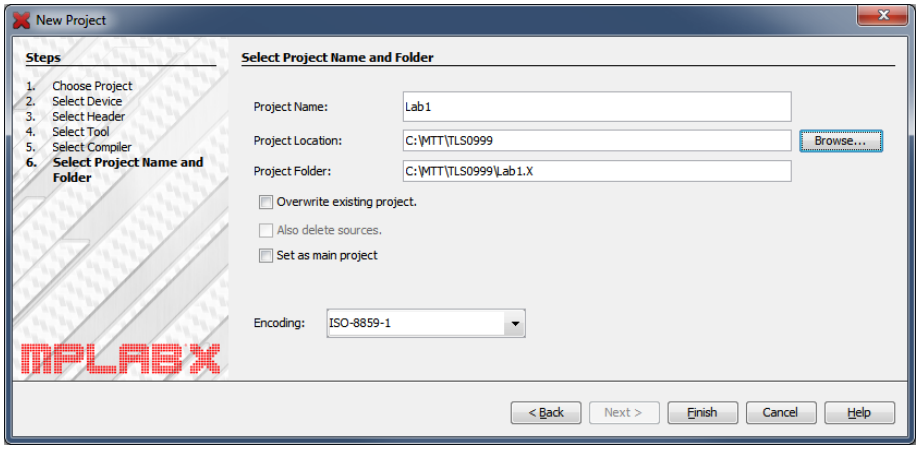
The project folder should automatically be populated with:
"**C:\MASTERS\1601\Lab1.X**"

Click Finish when done.

Information

If you are using a Mac or Linux machine, please enter the path to the location where you installed the TLS0999 directory.
For example:
/home/user/MTT/TLS0999

Figure 1.8
Select Project Name and Folder step of the New Project wizard



8 Add Files to Project

Now we need to add some files to our project. To save time, code has been written for you and if you performed step 7 according to the directions, the files should be in the project directory, Lab1.X. To add files to the project, **right click** on the logical folder where you wish the files to be added and select **Add Existing Item…** from the popup menu (figure 1.9). You will be presented with a standard "open file" dialog box from which you may add files to the project. Multiple files may be selected by holding down the Ctrl key as you click on them.

Using the technique described above, add the following files to the project in the specified logical folders:

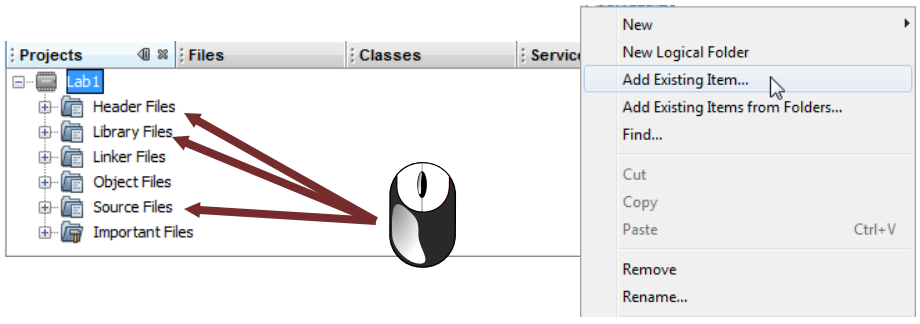
Header Files
C:\MASTERS\1601\Lab1.X**Lab1.h**
C:\MASTERS\1601\Lab1.X**libLCD.h**

Library Files
C:\MASTERS\1601\Lab1.X**libLCD24.x.a**

Source Files
C:\MASTERS\1601\Lab1.X**Lab1.c**

> **Information**
>
> The logical folders are strictly there for your convenience. You are not required to put specific file types into any of the folders. In fact, you can put any file type into any folder. You can add new logical folders and you can delete or rename existing folders. The folders themselves have no special meaning to the IDE or compiler. The Linker Files logical folder is the lone exception to this rule at the time of writing.

Figure 1.9
Adding files to the project's logical folders



As you add the files to the project tree, you may notice the **Store path as** options at the bottom right of the **Select Item** dialog box (Figure 1.10). For this exercise, just leave it set to **Relative**. When creating your own projects, you may want to change the setting based on the following guide:

Absolute: Specifies a fixed path to a file.

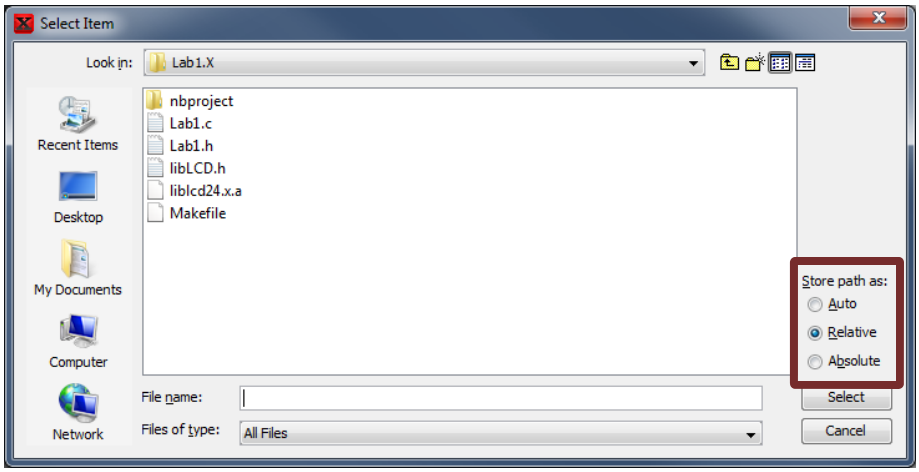


Figure 1.10
Select Item dialog box opened by selecting Add Existing Item...

This setting is most useful for referring to resources in directories outside of the project directory such as library code or files used by all of your projects. In that situation, you are free to move the project directory anywhere you like, as long as the external files remain in their fixed locations outside of the project directory.

Relative: Specifies a path relative to the project directory.
This setting is most useful for referring to files inside your project directory or within subdirectories of your project directory. This allows the project directory to be moved anywhere you like without breaking links to the files it contains.

Auto: Automatically specifies *Relative* paths for files inside the project directory and *Absolute* for files outside of the project directory. This is a good all around choice for most applications.

9 Build and Run the Project

To build and run the project, click on the **Debug Project** button on the tool-bar or select from the menu **Debug ▶ Debug Project (Lab 1)**.

The lab should build successfully, and MPLAB X IDE should automatically program and run the code on the device if using hardware or in the Proteus VSM Viewer simulator. You should see LEDs D6 and D7 blinking.

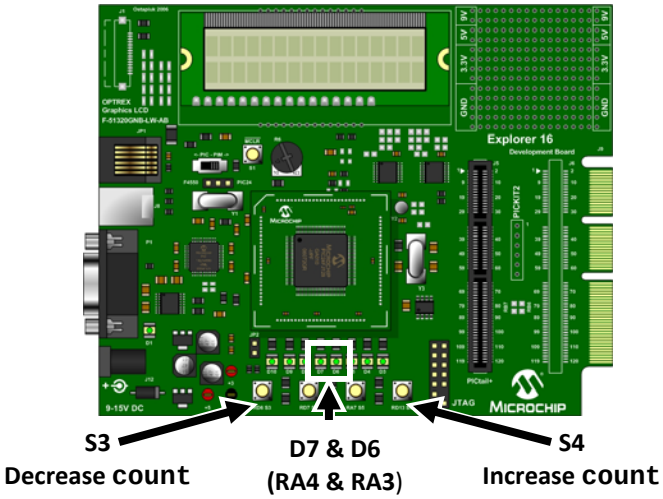
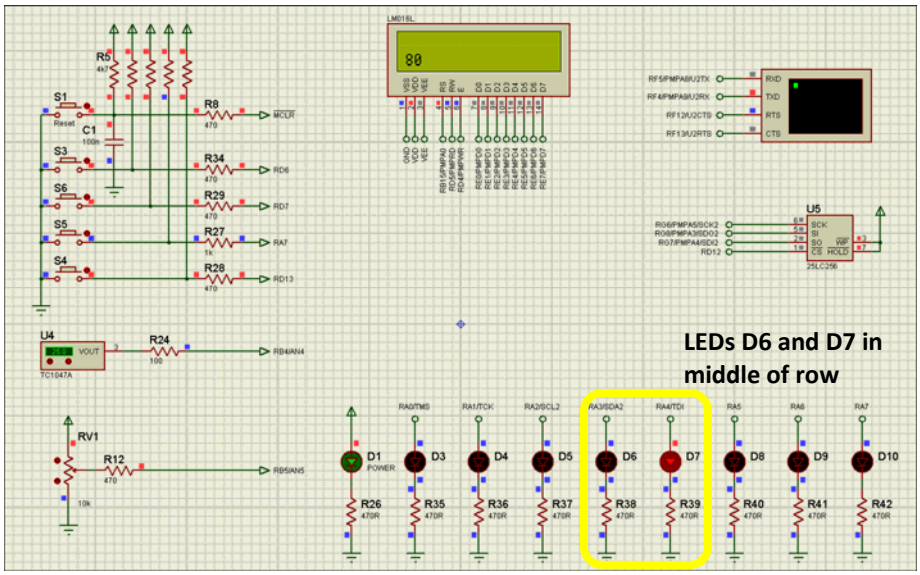


Figure 1.11
Location of LEDs D6 and D7 on the Explorer 16 Development Board

See Figure 1.12 for Proteus VSM Viewer screenshot

Figure 1.12
Location of LEDs D6 and D7 on the Explorer 16 Development Board as seen in Proteus VSM Viewer



10 End the debug session

Click on the **Pause** button

Click on the **Finish Debugger Session** button

Results

In this exercise, you used the project wizard to setup the project's required parameters and added existing source files to the project. Once the project was complete, you built and programmed the target. If everything was done correctly, you should have seen the value of the count variable (initially 80) on the LCD and two blinking LEDs (D6 & D7).

Conclusions

This exercise provided the foundation for everything you will ever do in MPLAB X. Although we focused on a standalone project, the process is nearly identical for all the other project types, the difference being in the way the project is built. You now possess the skills necessary to create projects that will serve as a framework for more advanced development tasks.

Lab Exercise 2

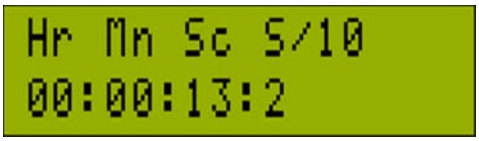
How to convert a legacy project to the MPLAB® X IDE Format

Purpose

This exercise will demonstrate the steps required to convert a project created with MPLAB IDE version 8.14.03A or later to the new MPLAB X IDE format. Upon completion of this exercise you will know how to prepare a project for conversion and how to use the IDE's built-in conversion utility.

Objective

The project we will be converting is a simple clock application written for use on the Explorer 16 Demo Board using a PIC24FJ128GA010. The clock will display hours, minutes, seconds and tenths of a second on the LCD character module of the demo board (either on live hardware or as a simulation using Proteus VSM Viewer).



Tools

- MPLAB X IDE 1.0+
- MPLAB C30 3.31
- REAL ICE™ or MPLAB ICD 3
- Explorer 16
- PIC24FJ128GA010 PIM

The Proteus VSM Viewer plugin for MPLAB X IDE may be used to simulate the project instead of running on live hardware.

Lab Exercise 2

Procedure

If the project from Lab 1 is still open, close it now by right clicking on its top node in the project tree and selecting **Close** from the popup menu.

Information

You must use the wizard in MPLAB X IDE. A new project may not be created manually as in earlier versions.

1 Start the new project wizard

Launch the new project wizard by selecting from the menu: **File > New Project...** or by clicking on the toolbar icon:

2 Choose Project Type

Select **Microchip Embedded** under **Categories** and select **Existing MPLAB IDE v8 Project** under **Projects**. Click **Next >** when done.

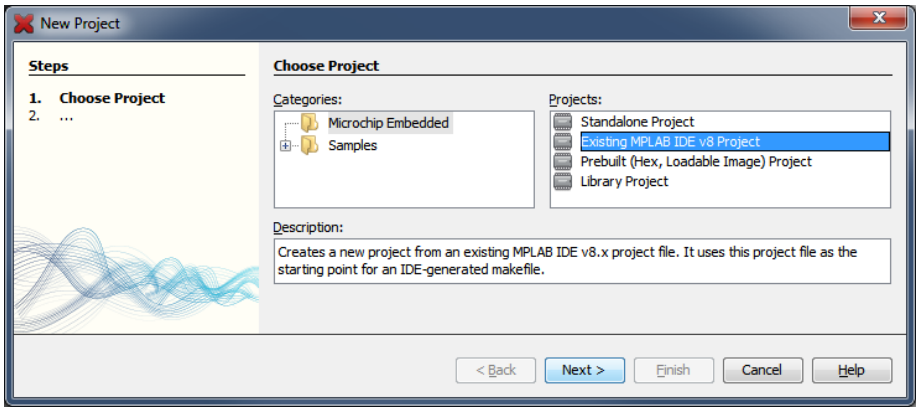


Figure 2.1
Choose Project step of the New Project wizard

3 Select MPLAB 8 Project File (*.mcp)

Click **Browse...** and select the following file:

C:\MASTERS\1601\Lab2\Lab2.mcp

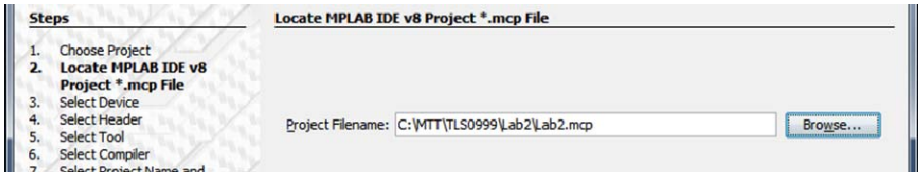


Figure 2.2
Legacy project file selection

Click **Next >**

4 Select the Family and Device

For this lab, change the device to the PIC24FJ128GA010. This project defaults to a different device, but is designed to work with several different microcontrollers.

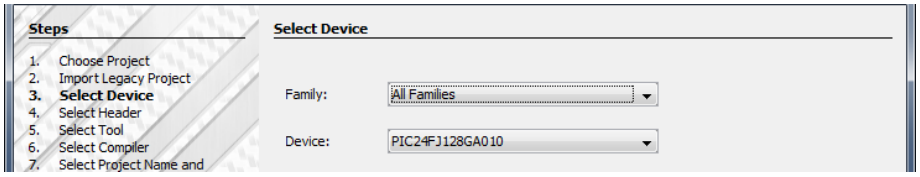


Figure 2.3
Device selection

Click **Next >**

5 Select Header

Since we are not using a header here, leave the box unchecked.

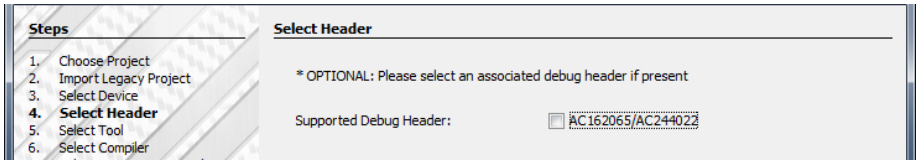


Figure 2.4
Header selection (none)

Click Next >

6 Select Tool

As with the first lab, select Proteus VSM Viewer. If you are using hardware, be sure to select the serial number of the tool you are using.

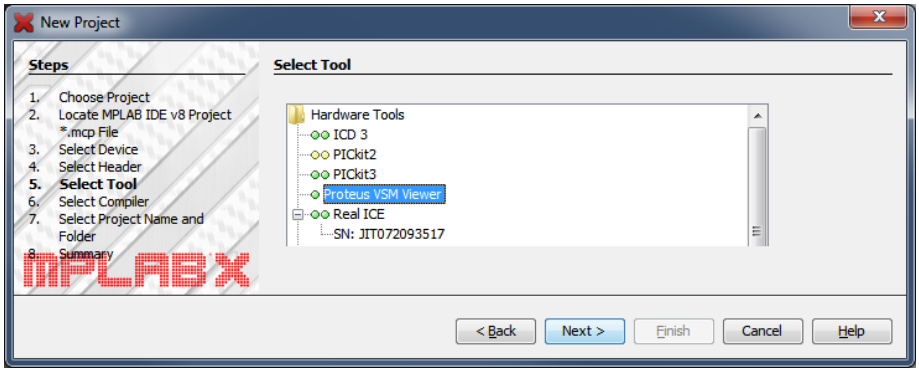
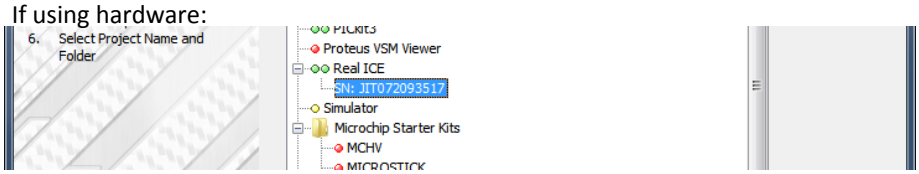


Figure 2.5
Debugging/Programming Tool Selection

If using hardware:



Click Next >

7 Select Compiler

As with the first lab, select the C30 compiler installed on your system by clicking on its version number. C30 version 3.31 is shown in this example.

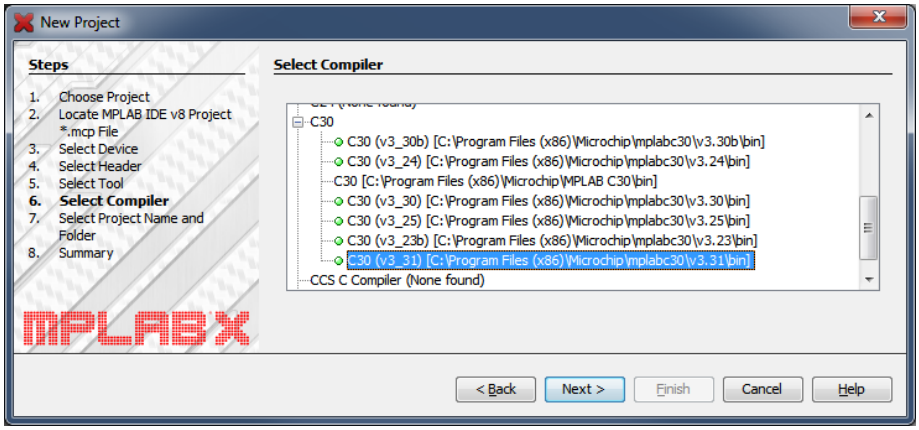


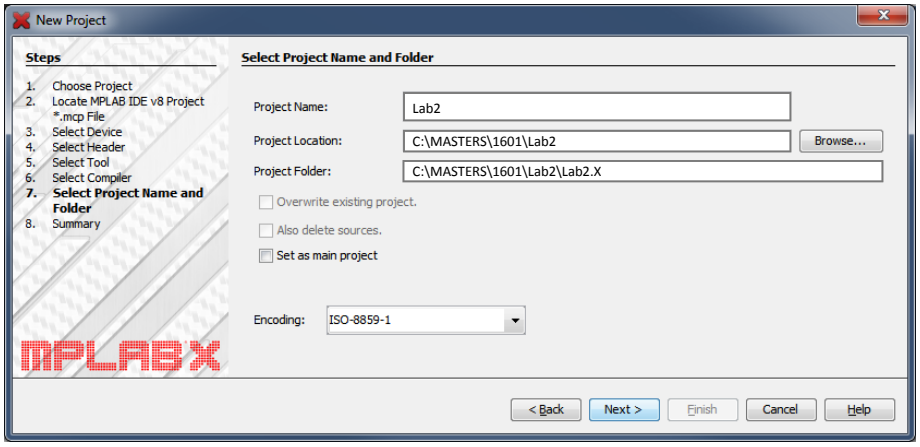
Figure 2.6
Compiler Selection

Click Next >

8 Select Project Name and Folder

For the purposes of this exercise, leave all settings here as they appear by default. This will create a subdirectory within the MPLAB 8 project's directory that will contain all the MPLAB X project files. The new MPLAB X project will reference the original source files in the MPLAB 8 project directory. Therefore, both the MPLAB 8 and MPLAB X project will share the same source files allowing you to work with the project in either version of the IDE.

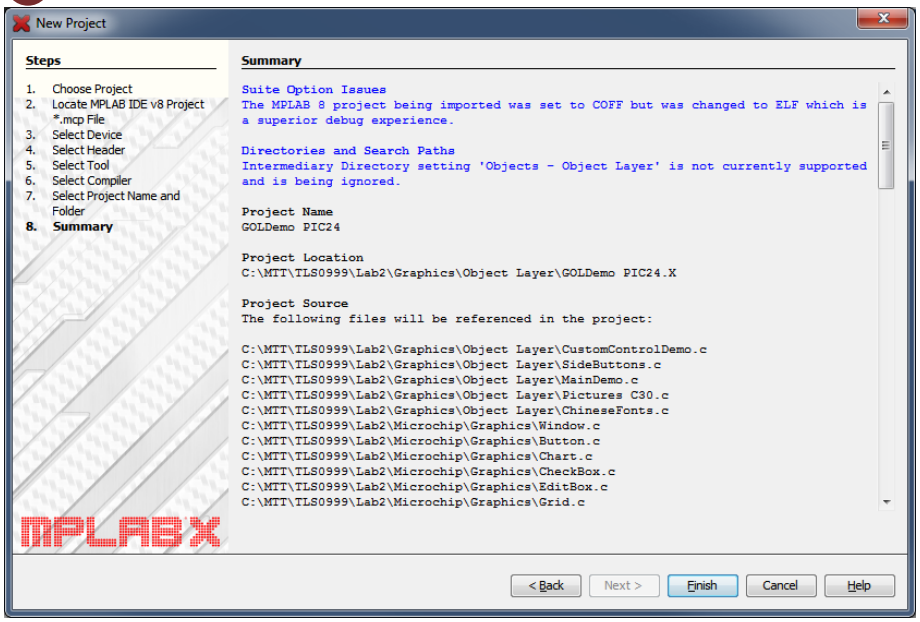
Figure 2.7
Select project name and folder (leave defaults)



Click **Next >**

9 Summary

Figure 2.8
Wizard summary screen



Click **Finish**

Information

The red text In the summary doesn't indicate an error—just important information about differences and limitations.

You should now have the new project open. The main difference between this and a standalone project created from scratch is that the source files reside outside of the MPLAB X project directory. The source files are all referenced by relative paths starting with ".." meaning "up" one directory level, which takes us into the MPLAB 8 project directory (the parent of the MPLAB X project directory.

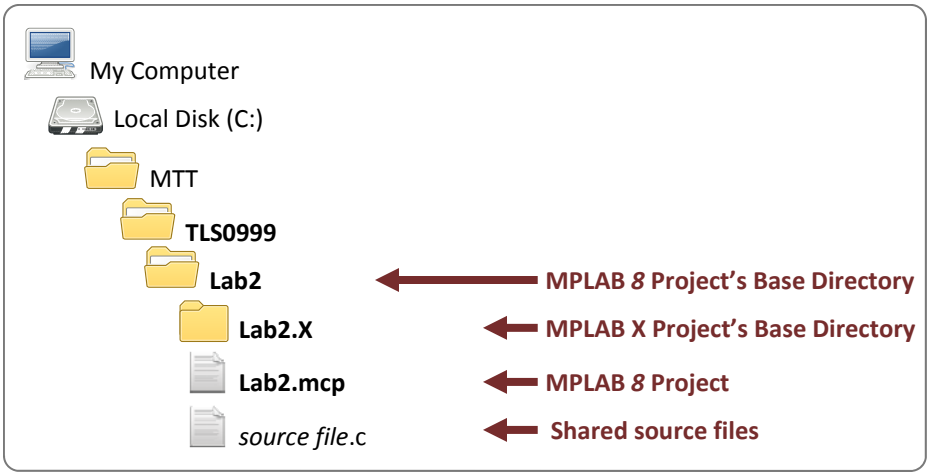


Figure 2.9
Converted project directory structure

10 Build and Run the Project

To build and run the project, click on the **Debug Project** button on the toolbar or select from the menu **Debug ▶ Debug Project (Lab 2)**.

The lab should build successfully. You should see instructions on the LCD to push switch S3 to start.

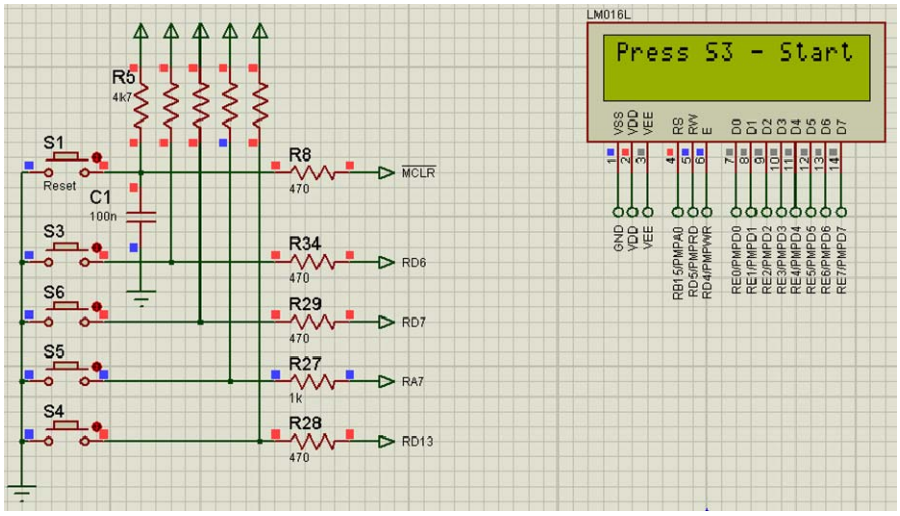


Figure 2.10
Proteus VSM simulation results.

Push S3 (either on the Explorer 16 board if using hardware, or on the schematic if using Proteus VSM Viewer.

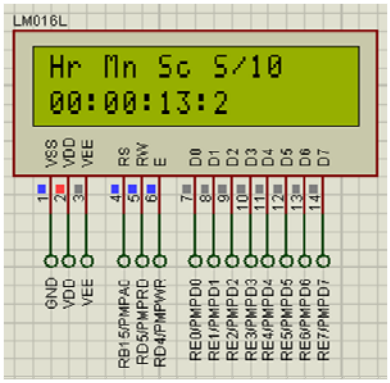


Figure 2.11
Running clock on the LCD

11 Halt the program and end the debug session when done

 Click on the **Pause** button

 Click on the **Finish Debugger Session** button

Results

In this exercise, you used the project wizard to import and convert an existing MPLAB 8 project. The code should have compiled and run without modification and a clock should be running on the character LCD of the demo board as shown in figure 2.11.

Conclusions

MPLAB X IDE makes it easy to convert legacy projects to the new format. Not only that, but you can continue to work on the project from either version of the IDE since they share the same source files.