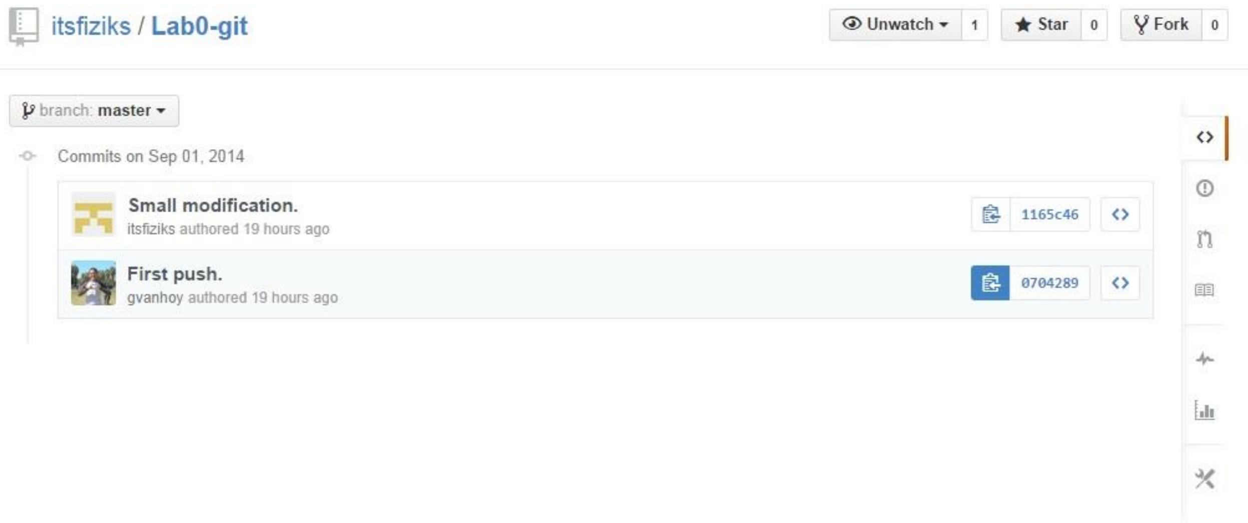
Lab 0: Introduction to Microchip 16-bit 28-pin Starter Board and PIC Development Tool

Starts: **NOW!**
Demo Due (Step G in this handout): September 11 – September 19, 2014
Points: 50
Points deductions:

- Your must demonstrate your working lab 0 code on the PIC24F demo kit in a stand-alone mode and briefly explain your C code to either TA or the instructor by the due date. Otherwise, 50% will be deducted from your lab 0 points.
- After your demo, ***submit your C file and screenshot of your commit history*** on D2L Dropbox by ***11.59 PM on Sept 19, 2014***. Without a C file, 20% will be deducted from your lab0 points. Without a commit history 10% will be deducted from your lab0 points. The commit history looks something like this on github



Commits on Sep 01, 2014

Small modification.
itsfiziks authored 19 hours ago
1165c46

First push.
gvanhoy authored 19 hours ago
0704289

Lab Overview: In this lab, you will familiarize yourself with PIC microcontrollers, the 16-bit 28-pin starter board, the PICKit3 in-circuit debugger, and C-based software development using the MPLAB Integrated Development Environment.

Datasheets and References

[Microchip 16-bit 28-pin Starter Board User's Guide]

[PIC24FJ64GA002 Datasheet]

Provided Software Code:

Source Code for Lab 0 Assignment: **lab0.c** (available on class D2L)

When writing your code, follow C Coding and Commenting Style Guidelines (see page 3)

Lab Procedure and Demo:

A. Install Microchip **MPLAB IDE version 8.92**
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2115

search for MPLAB IDE v8.92
or
MPLAB®X IDE
http://www.microchip.com/pagehandler/en-us/family/mplabx/

For MPLAB® X IDE user guide,
http://ww1.microchip.com/downloads/en/DeviceDoc/52027B.pdf

B. Install MPLAB C Compiler for PIC24 MCUs (**MPLAB XC16**). Use MPLAB®X IDE link above and then click on MPLAB®X FREE DOWNLOAD tab.

C. Install HyperTerminal or TeraTerm in order to communicate between your PIC24F demo board and the computer (your laptop/PC) [to receive and send characters].
Note: HyperTerminal is not installed by default in Windows Vista or Windows 7 (there is a free trial version available on Internet).
TeraTerm is available at: http://ttssh2.sourceforge.jp/index.html.en

D. ***Replace*** the dsPIC33F microcontroller chip that may be pre-installed on the 16-bit 28-pin starter board **with the PIC24F microcontroller chip** provided i***n the small black antistatic box***. ***Note:*** For all lab assignments throughout this course), we will use the PIC24F microcontroller.

E. Read the Microchip 16-bit 28-pin Starter Board **User's Guide** to familiarize you with your development board.

If using MPLAB X IDE, also read MPLAB® X IDE user guide http://ww1.microchip.com/downloads/en/DeviceDoc/52027B.pdf. There is also video tutorial available on the MPLAB® X IDE webpage

F. Follow the tutorials starting on page 4 (for MPLAB IDE v8.92) or page 9 (for MPLAB X IDE). Once your board starts working, make sure you understand the C code and then continue to step G below.

Note: Every time *you modify change the programming code*, step ***Building the code*** and step ***Running the application (code) or program the code in stand-alone operation*** must be redone **in order for the modified code to be working on the starter board.**

G. **Demo**:
G1. Modify the provided lab0.c code to achieve the following requirements:
a. The only LED that should be illuminated is the LED that is blinking. Modify the software code to ensure that only the currently blinking LED is illuminated.
Hint: Be careful to make sure you only change the output (latch) settings for the outputs connected to the LEDs.
b. SW1 (Switch 1) of the 16-bit 28-pin Starter Board is connected to a pin within PORTB of the PIC microcontroller. Determine which pin on the PIC that SW1 is connected to. Assign the appropriate value to the appropriate TRISB register bit to configure this bit as an input pin.

c. Modified the main loop of the software application such that whenever SW1 is continuously pressed, the currently selected LED will blink twice as fast. When SW1 is released the LEDs will blink at the initially defined rate.
   i. To get the LED to blink twice as fast, you can simply assign one half of the value already calculated to the PR1 register in the lab0.c code. Be sure to reset this value when the button is released.
   ii. Whenever you update the PR1 register, the TMR1 register should be reset to 0. Otherwise, you may notice some undesired behavior.
   Hint: The TMR1 register should be reset when assigning a ***new*** value to the PR1 register.
   iii. Extra Credit (5 point): Provide an explanation and demonstration of why not resetting the TMR1 register lead to undesired behavior.

G2. Stand-Alone Operation:

Follow step V of the tutorials to program your device in a stand-alone operation. If successfully, your board should be working with your modified C code after the PICkit3 is removed from the starter board.

Stand-Alone Program Demo and Code Explanation

During September 11 – September 19, 2014, you have to demonstrate your Lab 0 assignment to the TAs and/or instructor during the *scheduled* lab and *open* lab sections.

During this demo, you will be asked to provide both a demonstration of your working stand-alone programmed board and an explanation of the required modifications made to the provided software code.

C Code Submission on D2L

All C files for each lab assignment and final project should be submitted on D2L in the Dropbox. For scheduled lab assignments, your C programming code must be submitted by 11:59PM on the Friday of the week during which the lab assignment is due.

C Coding and Commenting Style Guidelines

a. All files should have the names and date. If working in a group, then all group members should be listed.
b. All functions should have a descriptive note of what it does.
c. All variables should have a self-descriptive name or comments describing what each one is used for.
d. Any PIC configuration settings should have comments indicating what the configuration corresponds to.
e. In later lab assignments, using an interrupt (instead of a busy wait) for the peripherals is a much preferred way to write your C programming code

Note: Class examples don't always adhere to this standard, be more diligent then we were.

Tutorial for MPLAB v8.92

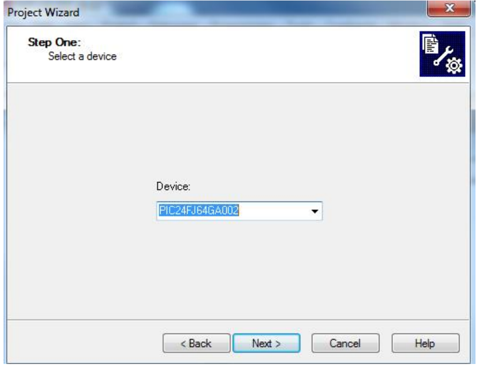
Note: This is a modified version of Chapter 2 of the 16-bit 28-pin Starter Board User's Guide

i. Creating the project

i.1 **Create a new folder** named it Lab0, to contain all files that will be created for this lab and download the C (*lab0.c*) code from D2L. Keep the *lab0.c* code in this Lab0 folder.

i.2 **Select a Device (Step One):**

- Start MPLAB IDE.
- Close any workspace that might be open (*File>Close Workspace*).
- From the *Project* menu, select *Project Wizard.*
- On the Welcome screen, click **Next >** to display the Project Wizard Step One dialog (see Figure below). From the **Device:** drop-down list, select **PIC24FJ64GA002**



Device: PIC24FJ64GA002

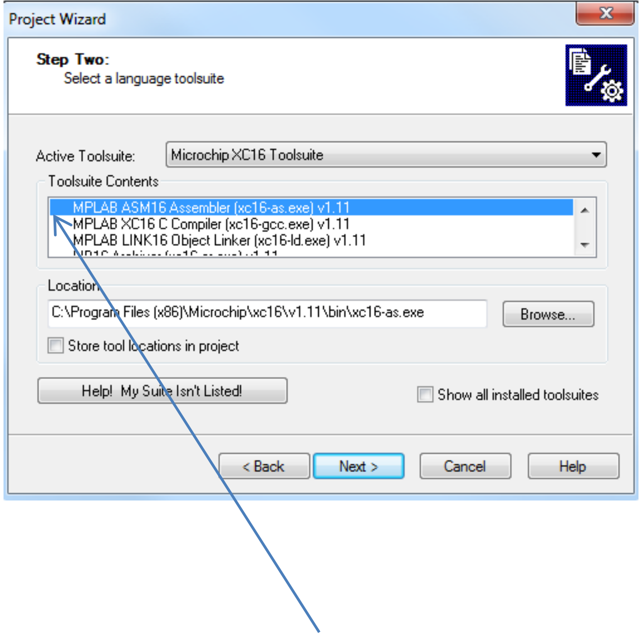
The device we are utilizing is **PIC24FJ64GA002**.

Please make sure to select this device.

- Then click **Next >**.The Project Wizard Step Two dialog appears.

i.3 **Select a Language Toolsuite (Step Two):**

- Select *Microchip XC16 Toolsuite* (assuming that you already installed the software). ***For each item*** under ***Toolsuite Contents***, use Browse to point to the location where the associated file is located. They should be located in *C:\Program Files (x86)\Microchip \xc16\v1.11*\bin directory.
  Note: the directory path depends on where you install the software!



Active Toolsuite: Microchip XC16 Toolsuite

Toolsuite Contents

MPLAB ASM16 Assembler (xc16-as.exe) v1.11
MPLAB XC16 C Compiler (xc16-gcc.exe) v1.11
MPLAB LINK16 Object Linker (xc16-ld.exe) v1.11

Location

C:\Program Files (x86)\Microchip\xc16\v1.11\bin\xc16-as.exe

□ Store tool locations in project

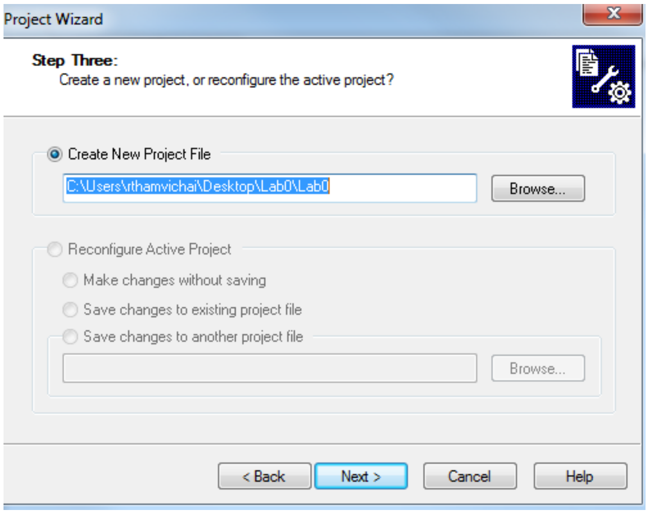
□ Show all installed toolsuites

- Once each item under Contents is point to the correct location, click **Next** > to continue. The Project Wizard Step Three dialog appears.

Note: if you do not point to the correct location, **X** will appear in front of that item. Make sure that no **X** shows up in front of each item.

i.4 **Name Your Project (Step Three):**

- Click **Browse...** and navigate to the Lab0 folder that you just created in step i.1 in order to place your project in that folder.
- In the **File name:** text box, type ***Lab0***
- Click **Next** > to continue. The Project Wizard Step Four dialog appears.


Project Wizard

Step Three:
Create a new project, or reconfigure the active project?

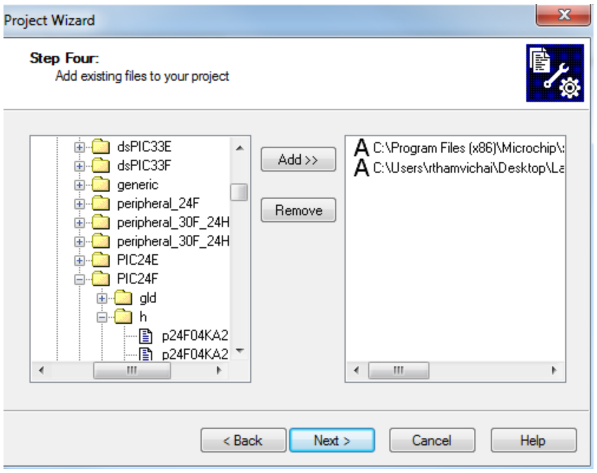
● Create New Project File: C:\Users\rthamvichai\Desktop\Lab0\Lab0

Browse...
Reconfigure Active Project
Make changes without saving
Save changes to existing project file
Save changes to another project file
Browse...
< Back
Next >
Cancel
Help

i.5 **Add Files to the Project (Step Four):**

- Add lab0.c file to the project:
  Locate the Lab0 folder and select the lab0.c file and Click **Add** >> to include the file in the project.
- Add the device specific header file for the PIC24FJ64GA002 to your project. Locate the header file *(p24fj64ga002.h)* and Click **Add** >> to include the file in the project.

This header file *(p24fj64ga002.h)* should be located in the directory ***C:\Program Files(x86)\Microchip\xc16\v1.11\support\PIC24F\h\***

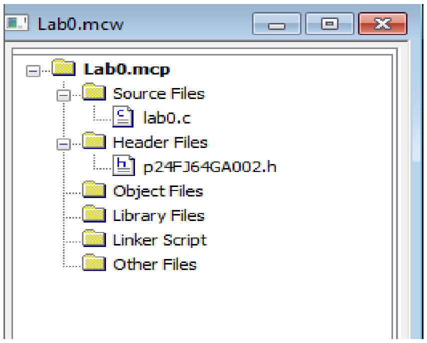

Project Wizard

Step Four:
Add existing files to your project

C:\Program Files (x86)\Microchip\:
C:\Users\rthamvichai\Desktop\La

- There should now be 2 files in the project. Click **Next >** to continue.
- When the summary screen appears, click **Finish.** Now, the MPLAB IDE Project window shows the lab0.c file in the **Source Files** folder, and the p24FJ64GA002.h file in the **Header Files** folder shown in Figure below
- Double-click the lab0.c file in the project window to open the file.

  Read the comments within the provided software code to get an ideal on how this provided code serial works!



ii. Building the code

ii.1 Before building your code, you must specify a heap size. This can be done using the following steps:

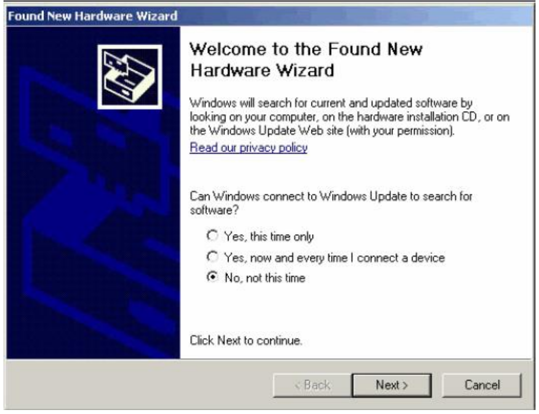
- From the *Project* menu, select **Build Options...->Project**
- Select the XC16 LINK tab
- Enter a size of heap of 512 under Heap Size

ii.2 From the *Project* menu select **Build All**. The Build Output window appears and it should show BUILD SUCCEEDED.

iii. Programming the Microcontroller chip in DEBUG mode

iii.1 Connect the 16-Bit 28-Pin Starter Development Board to your computer (laptop/PC) with the USB cable.

- The Found New Hardware Wizard dialog may appear as shown in Figure below.



Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

Read our privacy policy

Can Windows connect to Windows Update to search for software?

○ Yes, this time only
○ Yes, now and every time I connect a device
● No, not this time

Click Next to continue.

- Select **No, not this time**, then click **Next >** to continue. The Select Installation Location dialog appears.
- Select **install from a list or specific location**, then click **Next >** to continue. The Search and Installation Options dialog appears.

- Make sure that the 16-Bit 28-Pin Starter Development Board CD-ROM is inserted in the CD-ROM drive. Select the **Search for the best driver in these locations** radio button, and then select the **Search removable media (floppy, CD-ROM)** check box, and then click **Next >** to continue.
- Windows installs the USB driver. Select **Finish** to close the Found New Hardware Wizard window.

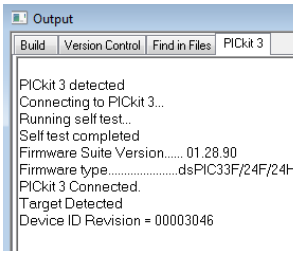
iii.2 **PICkit3 connection**

- Connect the PICkit3 to your computer with USB cable
- Connect the PICkit3 to ***J6*** position on the 16-Bit 28-Pin Starter Development Board
- Make sure that ***SW2*** on the 16-Bit 28-Pin Starter Development Board is in the "USB/DEBUG" position.
-

iii.3 **Enabling the PICKit3 Connection**

- From the *Debugger* menu, click *Select Tool>PICkit3* to designate the PICkit3 as the debug tool in MPLAB IDE.
- If needed, from the *Debugger* menu, select *Reconnect* to connect the debugger to the device.

The MPLAB IDE should report that it found the device as shown in Figure below.

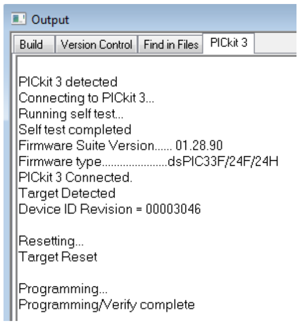


```
PICkit 3 detected
Connecting to PICkit 3...
Running self test...
Self test completed
Firmware Suite Version...... 01.28.90
Firmware type.....................dsPIC33F/24F/24H
PICkit 3 Connected.
Target Detected
Device ID Revision = 00003046
```

Note: MPLAB IDE may need to download new firmware if this is the first time the PICkit3 is being used with the device. Allow it to do so. If any errors are shown, double click the error message to get more information.

iii.4 **Programming the device** (**Debug mode**):

- From the *Debugger* menu, select *Program* to program the part. The output window displays the program status as shown below.



```
PICkit 3 detected
Connecting to PICkit 3...
Running self test...
Self test completed
Firmware Suite Version...... 01.28.90
Firmware type.....................dsPIC33F/24F/24H
PICkit 3 Connected.
Target Detected
Device ID Revision = 00003046

Resetting...
Target Reset

Programming...
Programming/Verify complete
```

iv. **Running the application (code) in DEBUG mode**

iv.1 **Start HyperTerminal (or TeraTerm)** to communicate with the PIC24F starter board in order to receive and send characters.

Read the comments within the provided software code to determine the correct serial configuration settings (baud rate and protocol).

iv.2 **Executing the Application:** Select *Debugger>Run* to execute the code. All four LEDs on the development board should light up. Only the LED at the D4 position should be blinking. If the HyperTerminal/TeraTerm is working, the following sentence should appear on its window

Hello!

Select LED to Toggle (4-7):

This will allow you to enter a number (4, 5, 6, or 7) in that window in order to change LED that will be blinking.

Study (make sure you understand) the lab0.c code when you have it working on the board.

iv.3 **Debugging the code: Follow Section 2.8** (starting on page 30 of the Microchip 16-bit 28 pin Starter Board User's Guide) to debug your code.
This is a very useful way to understand your code and will be very helpful later on when working on other lab assignments.

v. Programming the device (PIC24F) FOR STAND-ALONE OPERATION

There are 2 ways that you can do this step.
First way:

- From the *Programmer* menu, select the *Select Programmer>PICkit3* option.
- Make sure that ***SW2*** on the 16-Bit 28-Pin Starter Development Board is in the "PROGRAM" position.

  Note: If you were previously using the PICkit3 as a debugger tool, you will receive a warning message indicating that the tool cannot be enabled as a programmer and a debugger at the same time. Click **OK** in the warning message to continue.
- Select *Programmer-> Program* to program your code in the device.
- Once programming is complete, **SW2** must be switched back to the "USB/Debug" position for UART communication via the USB bridge.
- At this point, you can remove the PICkit3 connected to J6.
- If your board is still working (LEDs are lit up and one of them is blinking), this means you have successfully program your device in the stand-alone operation. You can change the blinking LED by entering a number on the HyperTerminal or TeraTerm window.
- Push the "RESET" switch if you want your board to start working from the initial mode (all LEDS are lit up and D4 LED is blinking).

Second way:

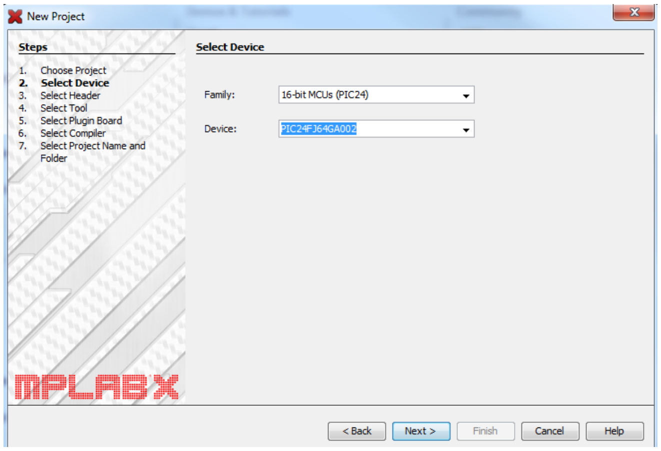
In the provided C code, the configuration bits are specified within the software code rather than through MPLAB. In order to program the PIC for stand-alone operation, you will need to modify these configuration bit settings.

Refer to both comments within the provided C code and the MPLAB documentation to determine the correct settings for programming your PIC for stand-alone operation.

Tutorial for MPLAB X IDE

Follow steps in Chapter 3 of the MPLAB® X IDE user guide http://ww1.microchip.com/downloads/en/DeviceDoc/52027B.pdf **with the following modifications:**

1) Create a new folder, named it lab0, on your computer (C drive or desktop) download the C (*lab0.c*) code from D2L. Keep the *lab0.c* code in this Lab0 folder.
2) Follow Step 1 of Chapter 3
3) In Step 2 (Select Device), choose **16-bit MCUs (PIC24)** for **Family** and **PIC24FJ64GA002** for **Device**. and then click **Next>**

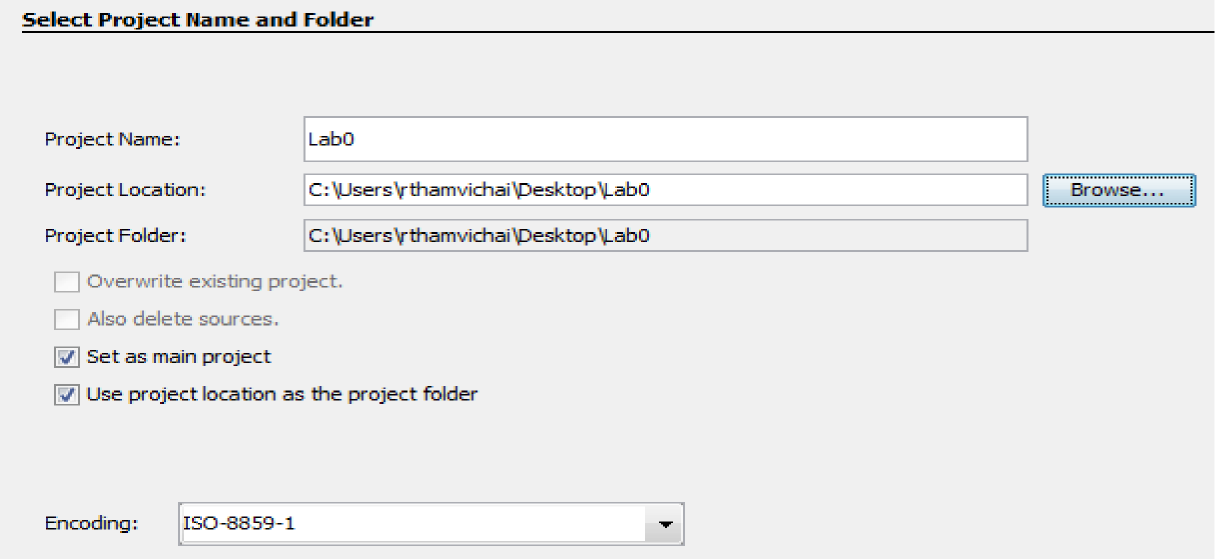


Select Device

Family: 16-bit MCUs (PIC24)

Device: PIC24FJ64GA002

4) Step 3 (Select Header), click **Next** > (to skip this step).
5) Step 4 (Select Tool), choose **PICkit3** and then then click **Next>**
6) Step 6 (Select Compiler), choose **XC16 (v1.11)** and then click **Next>**
7) Step 7 (Select Project Name and Folder), use Browse to go to the folder lab0 that you just created in step 1 above. Give a name to your project (similar to Figure below). Click **Finish**.



Select Project Name and Folder

Project Name: Lab0

Project Location: C:\Users\rthamvichai\Desktop\Lab0

Project Folder: C:\Users\rthamvichai\Desktop\Lab0

□ Overwrite existing project.

□ Also delete sources.

☑ Set as main project

☑ Use project location as the project folder

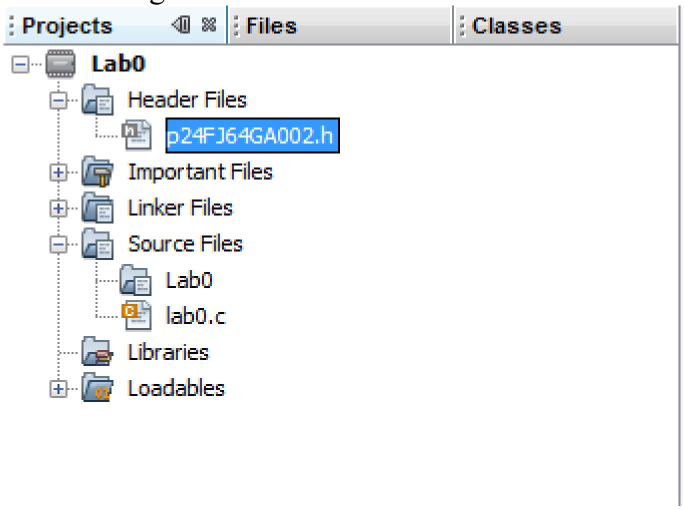
Encoding: ISO-8859-1

8) Continue reading section 3.3.2 - 3.3.5 in the MPLAB X IDE user guide. Again, we are using ***PICkit3*** as a debugger/programmer tool.

9) Section 3.3.6 (page 50) of the user guide:

9.1 **Add the lab0.c code (to Source Files folder) and the device specific header file for the PIC24FJ64GA002.h (to Header Files folder) to your project**.
The lab0.c code should already be in your folder from step 1 (otherwise, download it from the class D2L)
The header file *(p24fj64ga002.h)* should be located in the directory ***C:\Program Files(x86)\Microchip\xc16\v1.11\support\PIC24F\h\***

Note: Existing files can be added to a project by doing one of the following:
• Right clicking on the project in the Project/File window and selecting "Add Existing Item"
• Right clicking on a logical folder (e.g., Source Files) in the Project/File window and selecting "Add Existing Item"



9.2 Double-click the lab0.c file in the Projects window to open the file.
Read the comments within the provided software code to get an ideal on how this provided code serial works!

10) Continue reading section 3.3.7 (you can skip pages 51-53) in the MPLAB X IDE user guide.

11) Connect the starter board and PICkit3 to the computer

11.1 Connect the 16-Bit 28-Pin Starter Development Board to your computer (laptop/PC) with the USB cable. You may need to install the software for the new hardware (if yes, read chapter 2 of the user guide)

11.2 Connect the PICkit3 to your computer with USB cable

11.3 Connect the PICkit3 to ***J6*** position on the 16-Bit 28-Pin Starter Development Board

11.4 Make sure that ***SW2*** on the 16-Bit 28-Pin Starter Development Board is in the "USB/DEBUG" position.

12) Running the application (code) in DEBUG mode

12.1 **Start HyperTerminal (or TeraTerm)** to communicate with the PIC24F starter board in order to receive and send characters.

Read the comments within the provided software code to determine the correct serial configuration settings (baud rate and protocol).

12.2 **Executing the Application:** follow section 3.4.1 and 3.4.2 (Running and Debugging code). At this point, if the HyperTerminal/TeraTerm is working, the following sentence should appear on its window

Hello!
Select LED to Toggle (4-7):

This will allow you to enter a number (4, 5, 6, or 7) in that window in order to change LED that will be blinking.

Study (make sure you understand) the lab0.c code when you have it working on the board.

12.3 **Debugging the code:** Follow the rest of section 3.4 to learn how to debug (also single step through) your code.

This is a very useful way to understand your code and will be very helpful later on when working on other lab assignments.

13) **Programming the device (PIC24F) FOR STAND-ALONE OPERATION**

There are 2 ways that you can do this

First way:

- Make sure that ***SW2*** on the 16-Bit 28-Pin Starter Development Board is in the "PROGRAM" position.
- Click on Make and Program Device Project Icon to program your device
- Once programming is complete, **SW2** must be switched back to the "USB/Debug" position for UART communication via the USB bridge.
- At this point, you can remove the PICkit3 connected to J6 of the board.
- If your board is still working (LEDs are lit up and one of them is blinking), this means you have successfully program your device in the stand-alone operation. You can change the blinking LED by entering a number on the HyperTerminal or TeraTerm window.
- Push the "RESET" switch if you want your board to start working from the initial mode (all LEDS are lit up and D4 LED is blinking).

Second way:

In the provided C code, the configuration bits are specified within the software code rather than through MPLAB. In order to program the PIC for stand-alone operation, you will need to modify these configuration bit settings.

Refer to both comments within the provided C code and the MPLAB documentation to determine the correct settings for programming your PIC for stand-alone operation.