ECE 218
EXERCISE 1. LEARNING ABOUT MPLAB X IDE AND C

This exercise will use the MPLAB X IDE from Microchip and the X16 compiler. The tools are installed in the ECE 100, 102, and 106 lab computers, but you are encouraged to install them on your laptop as well. They can be found here:

- MPLAB X IDE: http://www.microchip.com/mplab/mplab-x-ide
- XC16 C compiler: http://www.microchip.com/mplab/compilers [be sure to select this particular 16-bit version of the compiler]

You will also need the Microstick II development kit, which is available from the bookstore, or directly from Microchip.
http://www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=DM330013-2

STARTING A NEW PROJECT

Before starting a new project, it is important to have your Microstick II kit connected to your computer's USB port, and to have the correct PIC microcontroller (the PIC24HJ128GP502) installed on the Microstick II.

Open the MPLAB X IDE application, and create a new project (File-New Project).

- Choose category "Microchip Embedded", and Project type "Standalone Project".

In the next screen, you will be asked to select a microcontroller family, and a particular device. On the Microstick II kit you have choices of microcontrollers, but we will install and use the following 28-pin chip:

- Family: 16-bit MCUs (PIC24)
- Device: PIC24HJ128GP502

For hardware tools, you will choose the Starter Kit (PKOB) - Microstick II starter kit. Note that it will not show up in this window if it is not connected to the USB port.

For the compiler, you will choose the X16.

In the next window you sill select a name for the project, a location, and a folder. Note that the project location is a top level of your project and the project folder defaults to a folder.X subfolder. To keep things simple, check the box for "Use project location as the project folder".

- Project name: Exercise1
- Project location: Browse to the desktop and create a folder for your project labeled Exercise1.
- Check "Use project location as the project folder" so location and folder are the same.
- Leave the default Encoding standard as it is, and select "Finish" to complete the process.

Identify the following windows in the MPLAB X IDE:

- Project and Files (upper left) – this is a hierarchical list of all the files (source, header, linker, etc) in the projects that are opened. Note that you can have more than one project open at the same time.
- Navigator (bottom left) – This is where the IDE settings for your project are displayed. You can find the PIC24 chip that you selected, and the X16 compiler, for example.
- Editor window (upper right)
- Output (bottom right)

C PROGRAM FOR BLINKING LIGHT

The source file for this first example is called "main_blink.c" and you can download it from Nexus. Place it in your project folder. The next step is to add it to your project.

- Right-click (on a PC) the "Source Files" entry in the Project window and select "Add Existing Item". **Be sure to select the "Relative" option for storing the path to the file.**
- Navigate to your project directory where your downloaded main_blink.c program is, and select it. It should show up under the Source File heading.

Next we will open the file in the editor (double-click on it) and take a look at all the parts. ***Notice the following types of instructions at the beginning, and briefly describe what they do:***

- /* */ ______________________________
- // ______________________________
- #include ______________________________
- #pragma ______________________________
- #define ______________________________

On the Microstick II board, pin 2 (RA0) is connected via a jumper to the LED on the board. The main_blink.c program defines the "led" variable to be pin RA0 (LATA0).

Notice there are two functions in this program, what are their names?

______________________________ ______________________________

Notice the following instructions and describe what they do, both on this sheet and as comments in code.

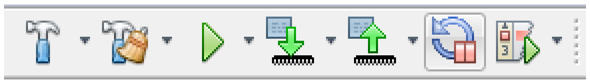
- uint16_t i, j

  ____

- for (i=0; i<iend; i++) ____
- AD1PCFGL = 0xffff; ____
- while (1) {} ____
- delay(); ____
- led = ~led ____

Executing Program

Next we will learn about the MPLab X tools for compiling, linking, assembling and downloading machine code to the PIC part on the Microstick II board. These tools are mostly found at the top center of the IDE. Explanations of the tools are given in the attached reference page, so refer to them as you follow the instructions for your blink example.



Build your project to make sure there are no problems with the code or your setup.

Debug Project to download the program to the Microstick II board in debug mode.

Observe that the light blinks. Hooray!

Pause the program and observe where the program counter is.

Put your cursor at the led = ~led statement, and run the program to the cursor.

Pause the program again and set a breakpoint at the led=~led statement.

Continue to the breakpoint. Continue repeatedly to manually turn the LED on and off.

Program Modifications

Modify the program so that it blinks once a second. There are many ways to accomplish this.

Once modified, use the Make and Program Device tool to program the modified version to the device. Using this tool, you will not be able to use the debugging tool menu.

ADDING HARDWARE

Now we add 2 more LEDs with different configurations of output ports, latched and open drain.

Second LED on RA1:

<u>Add an LED and a 300Ω resistor (in series) on the breadboard, connected to pin RA1 (pin 3). Modify your code so that it blinks both LEDs. The pin diagram for the</u> PIC24HJ128GP502 is given in Figure 1.

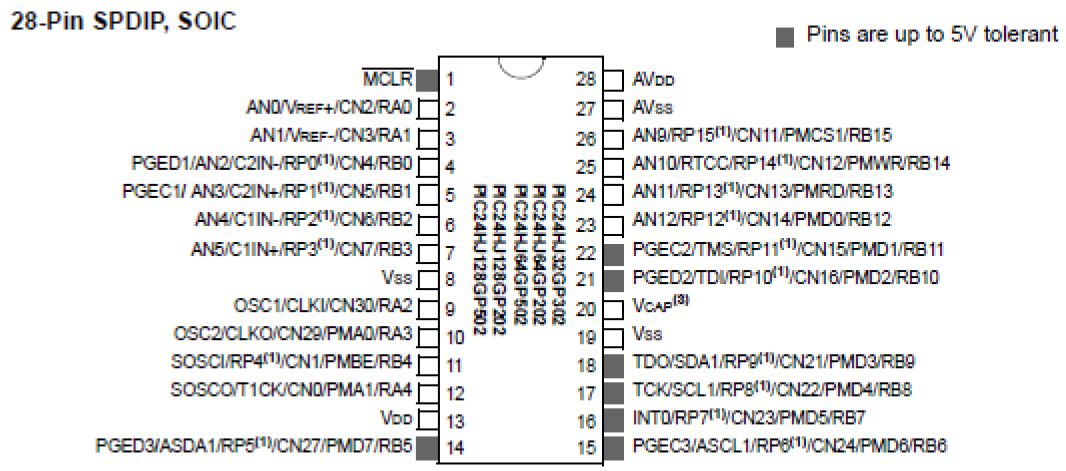

28-Pin SPDIP, SOIC
Pins are up to 5V tolerant
MCLR 1
AN0/VREF+/CN2/RA0 2
AN1/VREF-/CN3/RA1 3
PGED1/AN2/C2IN-/RP0(1)/CN4/RB0 4
PGEC1/ AN3/C2IN+/RP1(1)/CN5/RB1 5
AN4/C1IN-/RP2(1)/CN6/RB2 6
AN5/C1IN+/RP3(1)/CN7/RB3 7
VSS 8
OSC1/CLKI/CN30/RA2 9
OSC2/CLKO/CN29/PMA0/RA3 10
SOSCI/RP4(1)/CN1/PMBE/RB4 11
SOSCO/T1CK/CN0/PMA1/RA4 12
VDD 13
PGED3/ASDA1/RP5(1)/CN27/PMD7/RB5 14
28 AVDD
27 AVSS
26 AN9/RP15(1)/CN11/PMCS1/RB15
25 AN10/RTCC/RP14(1)/CN12/PMWR/RB14
24 AN11/RP13(1)/CN13/PMRD/RB13
23 AN12/RP12(1)/CN14/PMD0/RB12
22 PGEC2/TMS/RP11(1)/CN15/PMD1/RB11
21 PGED2/TDI/RP10(1)/CN16/PMD2/RB10
20 VCAP(3)
19 VSS
18 TDO/SDA1/RP9(1)/CN21/PMD3/RB9
17 TCK/SCL1/RP8(1)/CN22/PMD4/RB8
16 INT0/RP7(1)/CN23/PMD5/RB7
15 PGEC3/ASCL1/RP6(1)/CN24/PMD6/RB6
PIC24HJ32GP302
PIC24HJ64GP202
PIC24HJ64GP502
PIC24HJ128GP202
PIC24HJ128GP502

Figure 1. Pinout of PIC24HJ128GP502[1]

To complete the exercise, demonstrate both LEDs blinking.

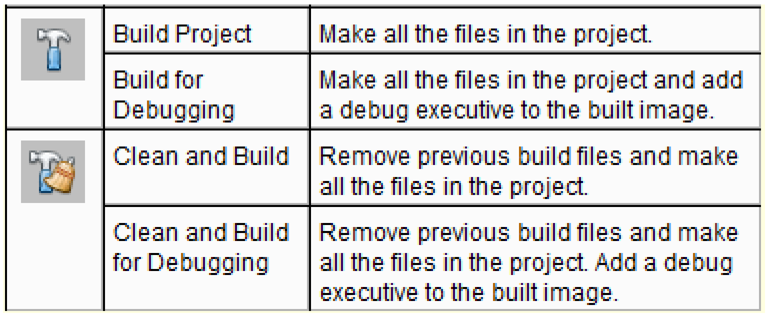
SAVE YOUR WORK

Save your project, exit MPLAB X IDE, and move your project to a flash drive or a google drive for future use. You will want to backup this drive to another location periodically.

[1] Microchip PIC24HJ32GP302/304, PIC24HJ64GPX02/X04 and PIC24HJ128GPX02/X04 16-bit Microcontrollers Data Sheet. http://ww1.microchip.com/downloads/en/DeviceDoc/70293D.pdf

MPLAB X IDE Tool Menu Icon Reference

The first time you compile a program, it is a good idea to use the Build Project tool.



	Build Project	Make all the files in the project.
	Build for Debugging	Make all the files in the project and add a debug executive to the built image.
	Clean and Build	Remove previous build files and make all the files in the project.
	Clean and Build for Debugging	Remove previous build files and make all the files in the project. Add a debug executive to the built image.

After building, you can "run" your code – note that the "Run Project" function does a "build" for you if you did not do it first, and it does NOT include the debug executive, so it will just run from the beginning, and you cannot set breakpoints, etc.

Run Project

If it makes sense to hold the device in Reset after it is programmed then use "Hold in Reset". Another click on the tool will start the program running.

Hold in Reset

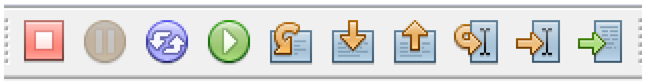
To Build and program the device (without the debug ability), use "Make and Program Device" (this is the same as "Run Project".)

Make and Program Device

When we have programs that need to be debugged, we use the Debug Project tool:

Debug Project

This tool builds your code, and programs the device with your program and the debug executive code, and starts a debug session. Once the debug session starts, the Debug icons will appear on the toolbar. NOTE – if your window is too narrow, the Debug icons are hidden. You can mouse over these tools to learn their names:



Finish Debug Session: Ends the current debug session and closes the connection to the debug tool.

Pause: Pauses debug operation without finishing the session. Watch windows are updated.

Reset: Resets processor

Continue: Resumes debugging until the next breakpoint or the end of the program is reached.

Step over: Executes one source line of the program. If line is a function call, it executes the function and then stops.

Step into: Executes one source line of the program. If line is a function call, it stops at the first statement in the function.

Run to Cursor: Run to the cursors location in the file and stop execution

Set PC at Cursor: Sets the PC value to the line address of the cursor

Focus Cursor at PC: Moves the cursor to the current PC address and centers this address in the window.

Breakpoints: You can also set breakpoints to run the program to a particular line in the code and then halt. Set a breakpoint by clicking on the left margin of the line in the Source Editor. The number of the line will turn to a red square. Click in the same place to clear the breakpoint. To run to the breakpoint, use the "Continue" tool.