ENGR 270 Lab #1 Online – Introduction to Development Environment

Objective

Introduce Microchip's MPLAB microprocessor Integrated Development Environment (IDE).

Preparation

Complete the following steps before starting to work on the experiments in this lab:

1) Install MPLAB X IDE Version 5.1 for your operating system from the following link: https://www.microchip.com/development-tools/pic-and-dspic-downloads-archive
   Note: You may have to search in the page links under "MPLAB X IDE Archives" section to find the correct version. During installation use only the default settings and only install MPLAB X IDE.
2) Complete lecture and assignment in Chapter 1. "Introductions"

Experiment 1. Introduction

In this experiment, you are expected to create a project, compile and simulate it in MPLAB for PIC18F1220 Microcontroller. You need to use the same process in all future labs so it is important to gain a solid understanding of the process and MPLAB.

Step I. Create a Project

1) Start MPLAB

2) Select menu "File>New Project"; in the pop up window :
   * Select Categories "Microchip Embedded" and Projects "Standalone Project", Click "Next"
   * Select "Advanced 8-bit MCUs" from Family list and "PIC18F1220." from Device list, Click "Next"
   * Click "Next"
   * Select "Simulator" from the Tools Menu, Click "Next"
   * Select "mpasm(v5.82)[C:\P…" from the Compile Toolchains, Click "Next"
   * *Enter Project Name, Location and folder, check "Set as main project", click "Finish"*
   Note: It is recommended that you create a new project directory for each new lab experiment to avoid the most common problem of including old files in new projects.

Step II. Add a Source File

1) Once project is created, new window pops up, on the upper left sub-window right-click on "Source Files" under your project name and select "New > AssemblyFile.asm".
   Note: If "AssemblyFile.asm" is not found then click on "Other", in the pop up window, click on "assembler" under categories and then select "AssemblyFile.asm" under file types.

2) Enter a "File Name" of your choice. main.asm is always a good choice, but you can choose something more descriptive if you like. Don't enter anything in the Folder: area. You should see a full path under "Created File:" which should end with "\main.asm", Click "Finish"

Note: On the upper-middle/right side of the window should be a tab labeled "main.asm" This is where you'll enter your program's source code.

3) Copy the following code into the main.asm file created in the last step. Save your using menu “File>Save".
Note: All instruction must be indented. Labels and #define must start on column 1. You can indent entire sections of text by selecting the text and hitting Tab key.

```
;---------------------------------------------------------------------------
; FILE: Intro.asm
; DESC: Experiment 1 – Simple Counter
; DATE: 5-18-06
; AUTH: Class
; DEVICE: PICmicro (PIC18F1220)
;---------------------------------------------------------------------------

   list          p=18f1220               ; Set processor type
   radix        hex                     ; Sets the default radix for data exp.
   ; Disable Watchdog timer,  Low V. Prog, and RA6 as a clock
   config       WDT=OFF, LVP=OFF, OSC = INTIO2
#define         PORTA                   0xF80
#define         PORTB                   0xF81
#define         TRISA                   0xF92
#define         TRISB                   0xF93
#define         ADCON1                  0xFC1

#define         COUNT                   0x080
#define         LASTIN                  0x081
#define         INPUT                   0x082
#define         TEMP                    0x083
   org          0x000                   ; Set the program origin (start) to absolute 0x000

; Initialize all I/O ports
   CLRF         PORTA                   ; Initialize PORTA
   CLRF         PORTB                   ; Initialize PORTB
   MOVLW        0x7F                    ; Set all A/D Converter Pins as
   MOVWF        ADCON1                  ; digital I/O pins
   MOVLW        0x00                    ; Value used to initialize data direction
   MOVWF        TRISB                   ; Set Port B RB<7:0> as outputs
   MOVLW        0x01                    ; Value used to initialize data direction
   MOVWF        TRISA                   ; Set Port A Pin 0 RA<0> as input
   MOVLW        0x00                    ; W = 0
   MOVWF        COUNT                   ; COUNT = WREG
   MOVWF        LASTIN                  ; LASTIN = WREG

Loop:
   MOVFF        PORTA, INPUT            ; INPUT = PORTA
   MOVF         INPUT, 0                ; W = PORTA
   XORWF        LASTIN, 0               ; W = W XOR LASTIN
   ANDLW        0x1                     ; W = W AND 0x1
   MOVFF        INPUT, LASTIN           ; LASTIN = PORTA
   MOVWF        TEMP                    ; TEMP = W
   BTFSC        TEMP, 0                 ; If TEMP<0> = 0 Then Skip Next Command
   CALL         IncL
   GOTO         Loop

IncL:
   MOVF         COUNT, 0                ; W = COUNT
   ADDLW        1                       ; W = W + 1
   MOVWF        COUNT                   ; COUNT = W
   MOVWF        PORTB                   ; PORTB = W
   RETURN

   end          ; Indicates the end of the program.
```

4) Enable Absolute Addressing
* Right click on the project name and select “Properties”

* Click on "mp*asm (Global Options)"*

** Click on "Unlock" if not grayed out*

** Click the checkbox for "Build in absolute mode"*

** Click "OK"*

Step III. Assembling a Project

1) Select menu "Production>Build Main Project" to assemble the code.
   Notes:
   **All instruction must be indented. Labels and #define must start on column 1. You can indent entire sections of text by selecting the text and hitting Tab key.*
   ** Select "Absolute" if prompted with "Absolute or Relocatable" code dialog box.*
   ** The lower-right window should have a tab labeled "Output - Lab1(Build, Load)" and the bottom of that window should include the messages "BUILD SUCCESSFUL" and "Loading Complete." Any messages in red indicate an error and messages in blue are warning. Both have to be investigated and corrected before a successful assembly.*

2) Disassembly listing can be generated by selecting menu item
   "Window>Debugging>Output>Disassembly Listing File

   Note: This step is only required for the report. If the menu is grayed out (not available) then right-click on the project name and select properties. In the property window, select "Loading", check "Load symbols…" and Click on "Apply. After the change, you need to redo step 1 to generate the disassembled listing.

Step IV. Initializing Simulator and Debugger

Use the simulation feature of MPLAB to test the functionality of your Project by following the steps shown:

1) In your source code listing, click on the line number next to the code line that says

   "CLRF PORTA ; Initialize PORTA"

   The entire line should turn red. This indicates that you've created a *breakpoint* at that line. Each time the simulator is about to execute a line with a breakpoint, the simulator will pause and return control to you.

2) Select menu "Debug>New Watch..." and under "Enter or Select Watch Expression" enter PORTA and click OK (remember that names like "PORTA" are case-sensitive: "PORTA" is different from "PortA"). Repeat this step for "PORTB" and "WREG".

3) Select menu "Window>Simulator>Stimulus" You'll see a new tab labeled "Stimulus." Click on the area beneath "Pin" and select RA0 (which specifies bit 0 of PORTA). Click on the box under Action and select "Toggle."
   Note: Toggle means invert the value from 0 to 1 and from 1 to 0.

4) Select “Debug>Debug Main Project." Your main dialog box should now say something like:
   i) Launching
   ii) Initializing simulator
   iii) User program running
   iv) User program stopped
   v) Breakpoint hit at line 21 in file ...

Step V. Simulation and Debugging

1) In addition to Special Function Registers, you can watch General Function Registers such as location 0x80. Click on the Variables tab, *double-click* <Enter new watch>, enter "0x80" and click on “OK”. You can repeat this step to add other watch locations such as 0x81, 0x82 and 0x83.

2) At this point you're ready to begin simulating/debugging your code. Select menu “Debug>Step in” or F7 to single-step your code, one instruction at a time. As the code executes, the current instruction will be highlighted in your source-code window. Hit F7 and observe what happens. You should see the program advance line by line, until it reaches the GOTO Loop" statement, at which point it will return to a previous instruction. Your variables should all show a Value of 0x00. This pattern will repeat indefinitely, until you change the value of your simulated input. Proceed as follows:

   a) In the Stimulus window, click the arrow under the "Fire" column;
   b) Click on the "Variables" tab to examine your variables;
   c) Continue hitting F7 and observe what happens. The PORTA line should turn red and show a value of 0x01 (this reflects the fact that you just toggled bit 0 of that port). Keep hitting F7 and observe what happens. After several steps, the program should reach the line labeled "IncL:
   d) After 3 more steps, you should see the value of address 0x80 changes to 0x01. After one more step, PORTB should also change to 0x01.
   e) The program will now return to looping indefinitely, until you switch to the Stimulus window and again click Fire. Then go back to Variables and hit F7 and you'll again observe PORTB incrementing after 10-20 instructions.

   Note: You can hold down F7 instead of repeatedly pressing it. You can also experiment with other options such as using “F5” to continue to breakpoint available under “Debug” menu. To set a break point, click on line number to set a break point (red square indicates breakpoint is set). The simulator output is update only when debugger is paused.

   You can select "Debug>Finish Debugger Session." menu to end simulation

Report Requirements

This lab and associated report must be completed individually. All reports must be computer printed (formulas and diagrams may be hand drawn) and at minimum:

For each experiment include:

- Clear problem statement in your words.
- Answer to any specific experiment questions (if any)
- Pseudo code which may be written in C-like syntax
- Disassembled code available after successful assembling
  Note:
  Disassembly listing can be generated by selecting menu item
  "Window>Debugging>Output>Disassembly Listing File
  If the menu is grayed out (not available) then right-click on the project name and select properties. In the property window, select "Loading", check "Load symbols…" and Click on "Apply. After the change, you need to recompile in order to generate the disassembled listing.
- Test plan which describes the input values and expect output/memory values for a successful design.
- Simulator output which shows the stimuli, relevant memory locations values showing validation based on test plan.

For the whole report include:

- A Cover sheet with your name, class, lab and completion date.
- A Lessons Learned section which summarizes your learning from this lab.
- A New Experiment section that has description of a new experiment and the experiment's results. Experiment should be related to material covered in class but not similar to one of the experiments in this lab.