CPE 490 Lab 3

## Goals

1. Learn how to use the Explorer 16 board with MPLAB .
2. Learn about configuration bits and how to set them
3. Learn about the clock options for the dsPIC33FJ256GP710A and how to get different instruction speeds using the same external oscillator.

## Overview

This lab will allow you to design, write and verify c using the Explorer 16 Demo board. You should also be comfortable in knowing how to set configuration bits, how to setup the main oscillator, and use counting loop structures in your C code.

## Report

This will be a normal report write up. Make sure to include the signatures when you turn in the report.

## Explorer 16 Tutorial

1. Plug in the 120 VAC to 9 VDC power supply to a wall socket. Connect the other end to the Explorer 16 demo PCB using the power jack.
2. Connect the PICkit 3 to the demo board, 6 pin, connector that is labeled “PICKIT” and J14 on the silkscreen of the demo board. MAKE SURE to align arrows on the PICkit 3 and the arrow on the silkscreen.
3. Connect the red cord from the PICkit 3 to a USB port of your computer. You should hear a double chime sound if your PC speakers are on.
4. Launch MPLAB. Go to the file Menu -> Recent Workspaces and pull up last week’s lab.
5. From the project window in the upper left of screen, click on the project and then right click on the project, select properties at the bottom of the list of choices
6. In the window that has opened up go to the hardware tool window and select the serial number under the PICKit 3 folder. If no serial number you PC has not recognized the PICKit 3.
7. Click on apply and then ok
8. Add the following code to your C source file make sure it is outside of main() toward the top of your program (more on this later in the lab):

#pragma config FWDTEN = OFF

1. Click on the Debug main project ICON 
2. The lights on the Explorer 16 board should blink on and off (maybe not at .5 second interval).
3. Get a signoff that the lights are blinking from Mr. Clark or Mr. Barlow.

Signature \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Configuration Bits

As we discussed in last lecture with MPLAB X you must include statements in your code to set the configuration bits. You can do this with the macros defined in the header file, or, you can do them from the configuration bits window, then have MPLAB X generate pragma statements which you will paste into your source code.

To get the configuration bits window: Select ‘Window’ on the menu bar, then select ‘PIC Memory Views’, and then select ‘Configuration Bits’.

From the newly made Configuration Bits Window every bit is described along with the 16 bit (only 8 bits are used) pattern for every configuration group. These groups are described in section 22 of your data sheet.

From the data sheet and what we talked about in lecture set the bits so that the default is taken for groups FBS, FSS, FGS, and FCID groups and you modify the correct bits in groups FOSCSEL, FOSC, ,and FWDT (remember the only bit in FWDT group that you must worry about is FWDTEN).

Once all the bits are correct click on the generate source code to output button. You can copy this code and past it into your source code, making sure that it comes before the main() function call. In your written report make sure to include the code so I can see the selections you have made.

## Oscillator Configuration

Using the given macro calls the dsPIC is configured to use the 8 MHz external oscillator with the PLL. Function. Looking at figure 9-1 in the data sheet shows our clock path from the OSC1 and OSC2 pins where our 8 MHz crystal is to the PLL and what comes out is Fosc. To calculate Fosc the following is given in section 9 of the data sheet:



Fin in our case is the 8MHz clock. Looking over pages 148 to 153 in the data sheet and recalling the last lecture figure out how to initialize the dsPIC to get a Fosc of 32 MHz and include in your initialization portion of your code. Realize that to get the instruction cycle, or clock speed driving Timer1 you need Fcy:



So considering the last lab Timer 1 will be driven at a frequency of 16 MHz. For help remembering see Figure 12-1 in the data sheet and realize that Tcy = 1/Fcy

## Make a Template

Take your current c source file and save it as template.c Strip out code that is specific and leave the include statement, the pragma statements, the clock setup, and anything else that you would think should be in a template.

In MPLAB X select ‘Tools’ from the menu bar, next choose Templates, next in the template manger window select the C folder and then press the ‘Add …’ button. Find your newly created template file to add to the C folder. Now when you add a source file to a project you can go to ‘Other’ tab and find your template.

## Lab 3 Design

Make sure to create a new project for Lab 3 at this point. Using what you learned from the previous labs, design a C program for the dsPIC33FJ256GP710A that will:

1. Using a loop that will index the PORT A single output that will be true, write code that will set each of the lower 8 bits in PORT A to true one at a time.
2. The bit will stay true for 1 second then turn off and the next bit higher will turn on immediately and stay on for 1 second and so on.
3. When the PORT A bit 7 is true for one second and turns off PORTA bit 0 will turn on.

## Verification

This time we will be using the Explorer 16 board to run and verify our code. That does not mean that you can’t use MPLAB SIM and the Logic Analyzer tool that we learned in the last lab to debug if you want. You will have to change the debugger back to MPLAB SIM to do so.

Think of way to verify the Explorer 16 hardware, that the PORTA pin is staying on for exactly one second and include this in you lab write up. Make sure to list any lab equipment that you use. Hint: To get some easy places to probe the demo board, the following is from the user guide for the Explorer 16.



## Witness

Have an instructor witness your hardware verification measure. **A witnessing signature is not verification! In your report you must describe how you verified the design to the input requirements.**

Signature \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_