ECE 2534 Fall 2012 – Lab 2
A Digital Combination Lock

This lab is an individual assignment.

Introduction

In this laboratory exercise, you will continue to learn about microcontrollers and common peripherals. Your task is to write software for the Cerebot board and two peripheral modules to emulate a padlock. The peripherals are the PmodENC rotary encoder module and the PmodOLED display. You will also need to use the PIC's timer module to enforce time limits on opening the lock.

Design Specification

In this assignment, you are to emulate a single-dial combination padlock such as the Master Lock padlock. Because there is no shackle on our Cerebot board, your system should indicate the locked and unlocked states with the LEDs and with the OLED display. As the user turns the rotary switch to enter the combination, the OLED will provide visual feedback on the position of the rotary dial.

An actual Master Lock provides a rotary field of forty distinct positions. Your design should support a rotary field of only fifteen distinct positions, numbered zero through fourteen. The OLED should always indicate the current position of the rotary action by displaying the field position. When the rotary switch is rotated counter-clockwise, the displayed count should increase. The count should roll over to zero as it passes the largest rotary value, which is fourteen for your design. When turned clockwise, the count should decrease, rolling over to fourteen when passing zero. Notice that the displayed roll-over may not correspond to a 360-degree turn of the physical switch. When the program first initializes, assume that the rotary switch is at position zero.

When the device is locked, all of the LEDs on the Cerebot board should be off, and the OLED should display the message "LOCKED". Your system should be in the locked state at initialization. The operation of unlocking should be identical to that of an actual Master Lock padlock:

1. TURN RIGHT two or more whole turns (of the displayed values, not of the physical switch) and stop at the first number in the combination.
2. TURN LEFT one whole turn (of the displayed values) past the above number and stop at the second number in the combination.
3. TURN RIGHT to the third number in the combination and stop.
4. Press and release the rotary switch. If the combination was entered correctly, the lock should open.

To indicate the unlocked state, LED1 of the Cerebot board should be on, and the message

"OPEN" should replace the "LOCKED" message on the OLED display. To enter the locked state again, the user should press the BTN1 switch on the Cerebot.

Unlike an actual Master Lock, your system should enforce a *time constraint* of 15 seconds on the unlocking procedure. When the user begins the unlocking procedure, the initial movement of the rotary switch should cause a count-down clock to appear on the OLED. The count-down values should be updated every second, to display the following sequence: "15", "14", "13", …, "3", "2", "1", "Time out!". Each of these values, including "Time out!", should be displayed for 1 second. If a time-out occurs before the unlocking procedure has been completed, then the count-down field should go blank; the user is then free to start the unlocking procedure again, with another 15-second limit on successfully entering the combination. If the user successfully completes the unlocking procedure during the 15-second limit, then the count-down field should go blank immediately, and the "OPEN" message should appear at approximately the same time.

Additional Requirements

For this assignment, you must use the PmodOLED peripheral module and the PmodENC rotary encoder module. You do not need any other Pmods for this assignment. You can find more information about the PmodENC at Digilent's web site,
http://www.digilentinc.com/Products/Catalog.cfm?NavPath=2,401&Cat=9.
The relevant documents are the reference manual (PmodENC_rm.pdf) and the schematics (PmodENC_revA_sch.pdf).

The PmodOLED peripheral should be attached to connector JD on the Cerebot board, just as in Lab 1. The PmodENC rotary encoder module has a single-row header containing only 6 pins, and it should be attached to the *bottom half* (pins 7 to 15) of connector JA on the Cerebot board. (By connecting the PmodENC to the lower row of JA, there should be less strain on the connector when you press the rotary switch.)

Create a new MPLAB project named CombinationLock to contain your code. Borrow source files, as needed, from your Lab 1 solution. Follow the software model given in Lab 1, with the code subdivided into modules using *.c and *.h files.

Create a new code module for the rotary encoder, using file names Rotary.c and Rotary.h. When your program accesses the PmodENC peripheral, it should do so through interface functions contained in Rotary.c. For example, your source file Rotary.c should contain an initialization function initRotary(), which you can model after initButtons() and initLEDs() from Lab 1.

At initialization, your OLED display should resemble the following:

E	C	E		2	5	3	4		L	a	b		2		
J	a	n	e		D	o	e								
L	O	C	K	E	D										
0															

As shown above, the first line of the OLED display should show "ECE 2534 Lab 2". The second line display should contain your name, such as "Jane Doe" (or the first sixteen characters of your

name, if it is long). The third line should display the lock status (“LOCKED” or “OPEN”). The bottom row of the OLED should display the rotary position (a value in the range 0 to 14) at the left.

The count-down field is at the lower right of the display. An example of the display during an unlocking procedure is shown below. The rotary dial has been moved to position 5, and 12 seconds remain before the time-out occurs.

E	C	E		2	5	3	4		L	a	b		2		
J	a	n	e		D	o	e								
L	O	C	K	E	D										
5														1	2

Your system should continually update the position of the rotary encoder on the OLED, regardless of the lock status or count-down value.

Specify your lock’s three-number combination (the "unlocking code") using #define statements similar to the following. Place these statements near the top of your main.c file. These represent the three numerical values, in order, which are required to unlock the Master Lock:

```
#define UNLOCK_CODE1   11   /* first (turn-RIGHT) target */
#define UNLOCK_CODE2    2   /* second (turn-LEFT) target */
#define UNLOCK_CODE3   13   /* third (turn-RIGHT) target */
```

As always in this course, good commenting style is required in your source code.

Validation

After you have finished your work, demonstrate your working program to a GTA who is on duty in the CEL. Take a hard copy of the validation sheet (provided separately), along with your Cerebot board, to the CEL. For some sections of 2534, a validation deadline will be announced. Please note that the CEL GTAs are willing to offer assistance when they are able, but they will give priority to validations over debugging as the deadline draws near. Any GTA in the CEL can validate your work – not just GTAs for 2534.

Hand in your signed validation sheet to your instructor at the beginning of the lecture that follows the project submission deadline.

Submitting your code to Scholar

Create a ZIP archive of all of your CombinationLock source files (all *.c and *.h files). Use the following file name, and upload it to Scholar before the announced submission deadline:

<Last name>_<First name>_Lab<lab number>.zip