Laboratory 3
Programming and Interrupts: An Exercise

Due Date: March 10, 2014
Points: 20 Points
Work individually.

Objective: Utilize your now finely honed skills to create Interrupt Service Routines to utilize interrupt concepts for UART activity, PWM interaction, and timer events.

Activities: The activities of this laboratory session are to create interrupt service routines for a timer function and also for PWM functions, and include them with UART activity. Please demonstrate each of the following sections to a lab assistant.

1) Interrupt section. Before writing the routines involved in this laboratory, read the User Manual section(s) that deal with the interrupts as they are implemented in the PIC32MX system. (Specifically, Section 8 of User Manual and individual subsections, such as the UART stuff). Identify the bits of the status register(s) that will be utilized.

2) Modify the echo routine that you wrote for the previous laboratory to function by using interrupts with the UART. That is, instead of using the flag in the status register to identify when a character is available, use the receiver interrupt feature (receiver interrupts when there is a new character). For this laboratory, don't worry about the transmit interrupt. (That is, when a transmit interrupt occurs, ignore it.) This time make the routine just echo – no repeat line to follow the line typed in. Note that this time the loop executed by the main routine will not be looking at the flag. Also note that the transmit portion of the job is speed limited by the input part of the job, so no speed checking is needed. Finally, just to make this interesting, implement this echo scheme in a manner that utilizes the rotating bit pattern of the previous lab. That is, the LEDs should show a continuously rotating pattern while the echo is happening. Also, to make things even more interesting, use a mailbox technique to send the character from the interrupt service routine to the main routine, and have the main routine print out the character.

3) Timer Subsystem. Read the subsection dealing with the Timers (Section 14). Utilizing the facilities of the Timers, create an Interrupt Service Routine that will generate a rotating pattern on the four LEDs on the trainer board Have the pattern rotate two times at a rate of half a second per LED, and then five times at a quarter second per LED (and then repeat). Note that interaction with the LEDs will be accomplished with programmed I/O utilizing the appropriate addresses. In other words, the task described here is to create an Interrupt Service Routine (ISR) that will respond to the Timer system set up so that an interrupt will initially occur in every half second, then proceed at the intervals described above. As in the previous section, implement the I/O activity in the main routine, and use the ISR only for the half-second timing activity. When the timer message arrives, rotate the pattern and see if you have done that enough times. If so, change the timing.

4) PWM Subsystem. Read the section for the PWM activity (see section 16). This document describes the functionality of the unit and how it can be used to create a pulse-width mechanism. Modify the program already created for the previous part of this laboratory to add a PWM system that has a period close to 25 µsec. The duty cycle should change with each period to make a pulse train where square wave cycles (cycles with a 50% duty cycle) are followed by cycles with a 25% duty cycle (then back to 50 %, and back to 25%, etc.) Hence, each time a period completes, an ISR should intervene and change the width based on what the previous width was. Demonstrate this with an oscilloscope. Remember that this action must be happening during the time in which the LED timing routine is also functioning.

Note: with all of the multiple interrupt activities mentioned above, make sure that the ISR that you create is capable of handling more than one interrupt at a time, should that eventuality arise.