# 16.317: Microprocessor Systems Design I

Spring 2014

Homework 6

Due **Friday, 5/2/14**

**Notes:**

- Only typed solutions will be accepted for this assignment, as you must submit an assembly or C file for each part.
- As noted in class, you must work in a group of 2 or 3 students on this assignment. **Please be sure to list the names of ALL group members on your final submission(s).**
- This assignment is worth a total of 50 points.

For each part of this assignment, you will write a program to be executed on the PIC16F1829 microcontroller included with your PICkit3 Starter Kit. Each problem will be graded based on the following:

- Your source code for the exercise, written either in C or PIC assembly language. **You must submit the actual .c or .asm files—do not simply copy and paste your code into another document.**
- A demonstration of your code working correctly.
  - The key points to be demonstrated are described with each exercise.
  - For each demonstration, you may either show your working project to an instructor or provide a video of the project working correctly. Any video submissions must demonstrate all of the key points.

1. (15 points) This program should repeatedly go through the sequence below, showing each set of values on the LEDs for approximately one second before switching. A value of '1' indicates the LED is on; a value of '0' indicates the LED is off.

   0111 → 0101 → 0001 → 1001 → 1000 → 1010 → 1110 → 0110 → 0111

   Note that:
   - After returning to the first state (0111), the sequence should restart with the 0111 → 0101 step, and then follow the remaining steps given.
   - The leftmost LED (DS1) should show the most significant bit of each step in the sequence, while the rightmost LED (DS4) should show the least significant bit. These LEDs are connected to Port C, but in the reverse order (DS4 holds the most significant bit within that port).

**Demonstration (5 points):** You must demonstrate at least two complete cycles of the sequence above, showing all eight steps in the sequence.

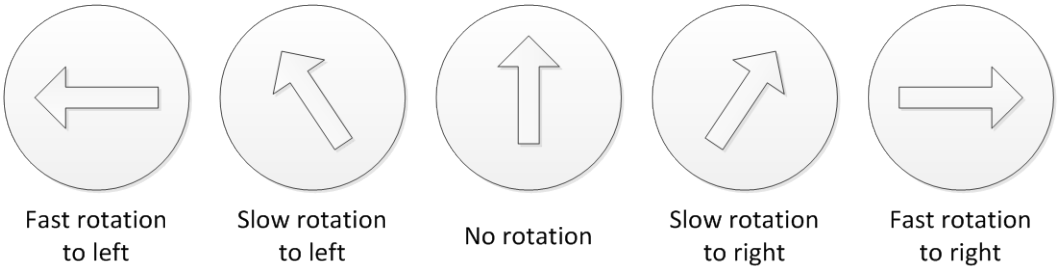
2. (15 points) This program should count the number of times the pushbutton on the board has been pressed and display that count in binary using the LEDs, with the leftmost LED (DS1) displaying the most significant bit of the count. (See the note above regarding how the LEDs are actually connected to Port C.)

   Note that, since there are only four LEDs, the maximum count value you can represent is 15. Once the count reaches 16, the LEDs should all reset to 0 and restart the count at that point.

**Demonstration (5 points):** You must show your project correctly handling at least 20 button presses and displaying the appropriate count on the LEDs.

3. (20 points) This program should rotate the LEDs using both variable delay and direction. The delay and direction should be controlled by the on-board potentiometer, which is read by the analog to digital converter (ADC) on the microcontroller. (Note: we will cover ADC operation in class on Friday, 4/25.)

   When the arrow on the potentiometer is pointing straight up (toward the edge of the board where the PICkit3 programmer connects), the LEDs should not change at all. (Note that it may be difficult to determine this exact position.) If the arrow points to the left of that position, the LEDs should rotate left; if the arrow points to the right of that position, the LEDs should rotate right. The speed of rotation is dependent on how far from the center the arrow is, as shown below:




   In addition to using the potentiometer to control direction and speed of rotation, this exercise should use the switch to enable and disable rotation. If the switch is pressed while the LEDs are rotating, the system should pause in its current position, regardless of whether the ADC input value changes. If the switch is pressed while everything is paused, the system should resume rotating in the appropriate direction based on the ADC input value.

   **Demonstration (8 points):** You must show that the potentiometer controls the direction and speed of rotation in the manner described—both slow and fast rotation in either direction, as well as pausing when the potentiometer is centered. You must also show that the button toggles the mode of operation—press the button to pause the system, and move the potentiometer to show that it does not affect the LEDs when paused. Press the button again to show that rotation resumes at that point.