CS 431 Lab #5
Joystick ADC, PWM, and Servo Motors

Spring 2015

Demonstration due the week of March 18th, 2015.

1 Overview

In this lab you will be using pulse width modulation to control the servo motors on the dsPIC33F board. You must write a program that does all of the following in order from 1 to 6.

1. Display a prompt on the LCD for specifying the joystick's maximum value in the x-dimension. You must also display the current digital value of the joystick's x-position as it updates in real time. When the joystick's trigger button is pressed, keep the digital value of the joystick's x-position at the time of the trigger press displayed on the LCD and stop further updates to this value. You must use debouncing to determine when the joystick's trigger button is pressed. (12.5 points)
2. Same as (1) but instead for the minimum in the x-dimension. (12.5 points)
3. Same as (1) but instead for the maximum in the y-dimension. (12.5 points)
4. Same as (1) but instead for the minimum in the y-dimension. (12.5 points)
5. Display a prompt on the LCD for specifying the servo motor pulse width in the x-dimension in microseconds. The joystick's current x-position must determine the current pulse width in the x-dimension as it updates in real time, such that it is possible to balance the ball at any position on the platform in the x-dimesion by moving the joystick to the corresponding x-position. When the joystick trigger button is pressed, keep the x-dimension pulse width at the time of the trigger press displayed on the screen and stop further updates to both the LCD as well as to the servo motors. You must use debouncing to determine when the joystick's trigger button is pressed. (25 points)
6. Same as (5) but instead for the servo motor y-dimension and the joystick's current y-position. (25 points)

2 Procedure

1. Before getting started, read sections 3.5, 3.6, and 4.9 in the CS 431 Laboratory Manual.

2. Use your code/project from a previous lab as a starting point for coding this lab and remove all unnecessary functionality. In particular, you may want to reuse your debouncing code from lab01.

3. A demo program that demonstrates the features that you need to implement is provided in compiled form on the course website. To upload it to your dsPIC33F, download load-lab05.bat and lab05.hex and follow the demo upload instructions posted on the course Labs page.

4. It is good practice to put associated reusable code into new .c and .h files. In this lab, you are should put the following two functions into flexmotor.c and flexmotor.h.

   (a) `void motor init(uint8 t chan)` Perform any necessary initialization of the specified motor channel. Use Timer2 for the duty cycle.

   (b) `void motor set duty(uint8 t chan, uint16 t duty us)` Set the duty cycle of the specified motor channel using the provided duty in mircoseconds. Use Timer2 for the duty cycle.

5. Write lab05.c such that it fulfills the requirements specified in the Overview section.

6. For details regarding PWM, ADC, and motor operation, see the following sources: lab manual, lecture 13, dsPIC33F.11 manual, dsPIC33F.13 manual, dsPIC33F.14 manual.

3 Post Lab

At the start of Lab06, each lab group will be asked to demonstrate and explain their Lab05 code to the TA.

The following questions are provided for your lab group to think about. No written response is required.

1. Why is debouncing necessary in this lab?

2. How could you complete this lab if you could not first calibrate the joystick positions?