MW 2411 Lab #5
Touchscreen and Servos

1 Overview

In this lab, you will use the Touchscreen and Servos on the Flex-UI/dsPIC33F to keep a ball on the corners, as demonstrated by the Figure below.
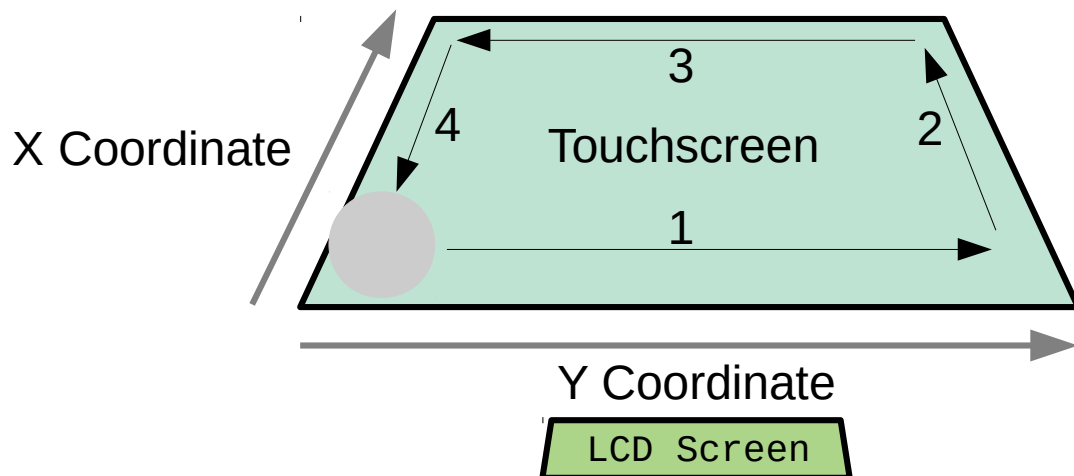

Figure 1: Demonstration of the behavior of the ball on top of the Touchscreen.

You must write a program that does all of the following.

1. Implement two functions to handle the servos:

   (1) which initializes and configures the servos. This function receives the servo number as parameters (X or Y) and does not return values.

   (2) which sets the duty cycle of the servo. This function does not return a value and receives the servo number (X or Y) and the duty cycle in microseconds as parameters.

2. Implement three functions to handle the touchscreen:

   (1) which initializes and configures the touchscreen (does not receive parameters nor return anything).

   (2) which changes the dimension in which the touchscreen reads from (horizontal or vertical). This function receives the dimension as parameter and configures the touchscreen accordingly.

   (3) which reads the current ball position on the touchscreen. This function does not receive parameters but returns the position as an unsigned int of 16 bits.

3. Assuming that the ball starts at the front left corner, as shown in the Figure above, you must write code that keeps the ball on one of the touchscreen corners, also as demonstrated by the Figure above. At every 5 seconds, your program controls the servos to change the ball position (steps 1, 2, 3, and 4 in the Figure). Iterate over those 4 steps infinitely.

4. Print the current ball position (horizontal and vertical) on the LCD at every iteration.

2 Procedure

1. Before starting, read the sections 3.5, 3.6, 4.8, and 4.9.

2. Note that it takes about 10ms for the touchscreen output signal to be stable when the touchscreen is switched from one operation direction to another.

3. Before testing the servos, test the duty cycle of the servos using the oscilloscope to not damage them with wrong values (see Figures 10 and 11 in the lab manual).

4. You can find an MPLAB X IDE project with template code on the Moodle course page.

5. If you split your code into multiple files make sure to have a header file (.h) for the function header and related declaration/definitions and a .c file for the implementation.

6. You don't need to use timer interrupts in this lab.

Due date of code submission can be found on the Moodle submission page for this lab. Only one member of the group must upload the code (all .c and .h files that your project uses compressed in one zip file). At the start of Lab 6, each lab group will be asked to demonstrate and explain their Lab 5 code to the lab instructor.