MW 2411 Lab #6
Amazing Ball Control
Summer 2022

1 Overview

In this lab, you will let the Amazing Ball Board move the ball in a circle on the touchscreen.

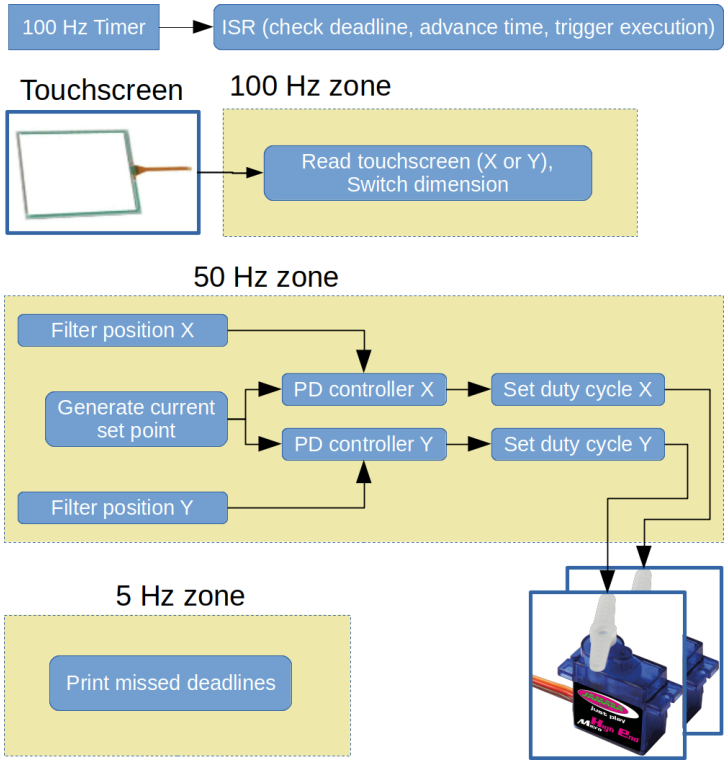

Figure 1: Design of the Amazing Ball project.

General remarks:

- Values read from the touchscreen require a filter, since they can be very noisy.
- A target position (two set points (X/Y)) is generated with a frequency of 50 Hz. The target position moves in a circle.
- To match the target position and the actual position a PD controller (like a PID controller, but without the I part) is used. This controller will tilt the plate to move the ball (actual position) to the current set point (moving in a circle).
- To check if the code is executing fast enough a counter for missed deadlines is increased every time an execution period starts and the previous execution period has not ended yet.

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

1. Set up a timer. The timer interrupt should advance the time for the calculation of the current set point and serve as a source of time for the frequency domains.
2. Read the touchscreen X and Y position with a frequency of 100 Hz (that is, at every 10 ms you must read either X or Y and then alternate between them).
3. Implement low pass filters for X and Y position (Butterworth 1st order, corner frequency 3 Hz).
4. Implement PD controllers for X and Y dimension. Before trying to optimize the parameter for P and D, make sure the plate is leveled when the PD controller outputs zero (What duty cycle results in a leveled plate?). Then choose a static set point, push the ball by hand and optimize P and D parameters. Be ready to remove the ball at any time when optimizing the PD controller. Start with a small ball for the rough setup, then switch to a medium sized ball. (The medium sized ball is much heavier and can destroy the touchscreen.)
5. Generate a set point for X and Y dimensions based on the current time. The set points must describe a circle. Make the radius, center and speed configurable using `#define`.
6. The timer interrupt handler must check whether the previous execution phase has completed by the deadline or not.
7. Print the current number of deadline misses to the LCD with a frequency of 5 Hz.

2 Procedure

1. You can find an MPLAB X IDE project with template code on the Moodle course page.
2. Reuse the code you developed in the previous lab (to read from the touchscreen and actuate on the servos).
3. Note that it takes about 10ms for the touchscreen output signal to be stable when the touchscreen is switched from one operation mode to another.
4. Make your solution simple (i.e., there is no need for more than 1 timer interrupt for instance).
5. A demo program is provided in compiled form on the course website.
6. Printing too much debug information may cause deadline misses, so be careful about how often `lcd_printf(...)` is called.
7. Practical information on PID controller optimization:
   `www.youtube.com/watch?v=JFTJ2SS4xyA&t=390`

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 7, each lab group will be asked to demonstrate and explain their Lab 6 code to the lab instructor.