CPEG222 Spring 2017 Project 5
Robot Olympics

In this project, we will implement a line following robot.

Goal of this project:

- Integrate everything you learned from the course together

Minimal Hardware:

- PIC32 MX7 or MX4 board
- Two seven-segment display pmods
- One microphone pmod
- One 8LD high brightness LED pmod
- Robot Kit Frame
- One IR sensor pmod
- 4 IR sensors
- One R/C Servo Control Pmod
- 2 R/C Servos
- Oscilloscope (debugging)

Inputs:

- Microphone
- IR Sensors

Outputs:

- On-board LEDs
- 8 LEDs on 8 LD pmod
- Seven-segment Displays
- R/C Servos

Functional description:

You are required to implement a line following robot that has the following functionalities.
The exact number of states and transitions are left for you to decide.

Required Functionalities:

- Follow a 2" wide printed black line on a white background.
- Make turns based on the black line path.
- Use an analog microphone input into and ADC10 pin to trigger the start of the robot off of two load signals (hand claps) within a 1 second time span.
- Using the CoreTimer, calculate the time it takes to complete the course in tenths of a second (i.e. 100s of milliseconds) and display it on the 4 SSDs such that the numbers are right side up (known as Hebling mode). The timer should automatically start and stop.
- Use Analog Output pins configured as PWM waveforms to control the server motors to enable your robot to go straight and turn to follow the path.
- The robot must stop at the solid back end line. There will be a time penalty if the robot goes beyond the end line. After stopping at the end line and waiting for 5 seconds, additional points will be given for robot "celebration" moves.

Line following rules:

- There will be a 12 to 15 foot long track with curves. A 2" wide black starting line and ending line designate the extents of the course.
- At the starting line, the robot will wait for the starting signal (two load noises within a second), detected by the PmodMIC. As soon as it detects the starting signal, it will start the timer and move forward. Incorporate a built-in backup initiate sequence where Btn2 is pushed when in the Ready state to start the robot. This will entail a 10 second penalty.
- There will be both gentle and sharp curves turns on the track with the smallest curve angle of roughly 90 degrees.

Control the speed/direction of the robot:

- Your PWM period should be 16~23ms, which means the frequency should be between 44 Hz and 62.5 Hz as described in class. A 30 second penalty will be given if it is determined that your pulse repetition rate exceeds this.
- The pulse width range is from 0.9 ms to 2.1 ms. When the pulse on width is 1.5 ms the servo is neutral. When the width is smaller than 1.5 ms the servo rotates clockwise. When the width is greater than 1.5 ms the servo rotates counter clockwise.

Operational modes of the project:

1. Initial mode should allow for the IR sensors to sense the presence of the black inkjet line and allow for the IR sensor threshold to be adjusted accordingly. This diagnostic state will also capture the mic input and display the sound level on the 8LD Pmods as in Project 4. LD1 of the 4 on board LEDs will be lit with the other 3 off. This diagnostic mode will be used to test the IR sensors and the microphone input.
2. Once started, capture the elapsed time in tenths of a second using the CoreTimer and display this on the SSDs. A template for this timer will be provided. A 30 second time penalty will be added if it is found that an inaccurate time is used. Also, use the Output Compare modules (check reference manual Chapter 16 for more details) in PWM mode, together with the external R/C Servo Control pmod, to control the two servos. Configure the Pulse Width Modulation (PWM). Hook it up to an oscilloscope and make sure the waveforms are generated correctly and precisely**. Capture these PWM waveforms and include them in your mid-project report as listed. You need to first have a working PWM before attaching the servos!**
3. Attach the Servos and get them working with PWM. You need external power for this one. You can use 4 - AAA batteries as external power (instead of USB supplied power). You are responsible for supplying your own AA batteries.
4. Program your state machine algorithm to finish the path.
5. Stop at the finish line.
6. If desired, perform a celebratory move after waiting 5 seconds.

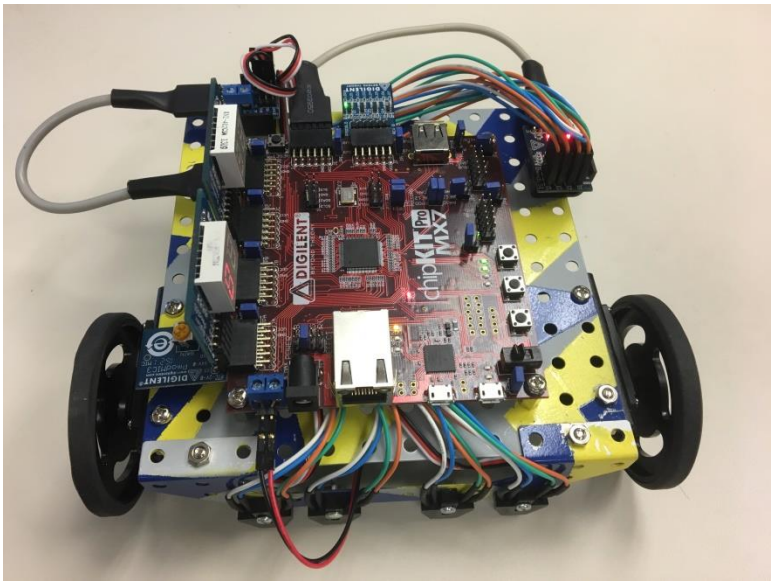
Hardware assembly:

https://reference.digilentinc.com/_media/line-following_servo_robot_kit:srk-line_rm.pdf

We are using a different board so port names/jumper names will not be the same. The picture below shows the connections for the battery & motors. Here is the summary:

- Battery box is connected to J18. Switch the Power Select jumper (J3) to "EXT" (external power).
- Motors are connected to the servo CON3 pmod three-pin jumpers (make sure the orientation is correct) which in turn connect to Output Compare pins on the PIC32.
- Two terminal jumpers on the pmod (J6) are connected to J18 on the board.

- Connect the 4 SSDs to the top rows of Ports A –D using right angle connectors. Program the SSDs to display the numbers right side up.
- Add Mic Pmod connected to and ADC10 input.
- Add an 8LD Pmod.
- Be careful with the orientation of the wires (You can accidentally short GND with VCC).



Midproject Report (uploaded to Sakai):
Your midproject report should include the following:

A list of the states of operation of your robot for the final project
A flowchart for the final project
A plot or screen capture of the following PWM waveforms where you note the pulse period, the pulse repetition rate, the pulse high time, and the duty cycle.
Static/neutral
Counter clockwise rotation (CCW)
Clockwise rotation (CW)
A list or table that covers <u>all</u> of the LED, button, and pmod connections including the MX7/4 port/button/LED, the pin configuration, and the PIC32 pin

Partial Example:

Pmod	MX7 Port	Pin configuration	PIC32 pin
Btn1	Btn1	Digital Input	RG6
LD1	LD1	Digital Output	RG12
SSD1	JA-01	Digital Output	RB2
SSD1	JA-02	Digital Output	RB3
…			
SSD2	JC-01	Analog Input	RC1
…			
Mic	JA-08	Analog Input	AN8
LS1	JE-01	Digital Input	RD14
…			
OC3	JD-07	Output Compare – PWM	OC2

In class mid-project demo:

As a mid-stage demo, show that your robot can do the following:

Use on board Btn1 to change motor operating states between:

Stop/Still (LD 1, 2, 3, and 4 are on)
Forward (LD1 is on/others off)
Turn Right (LD1 and 2 are on/others off)
Turn Left (LD3 and 4 are on/others off)
Reverse (LD4 is on/others off)

Timer (using CoreTimer) captures time in tenths of seconds and displays it on the SSDs. Starts at 0 and then counts when in modes 2 – 5 above. Timer pauses in mode 1 (Stop/Still). Btn 2 resets timer to 0. SSDs don't display leading 0's. In the midproject demo the numbers can be presented either right side up or upside down but must be right side up for the final Robot Olympic demo.

Due Dates:

- Thursday (4/27, 11:55 AM thru Doodle Poll) – Team Name and Race time slot selection due
- Monday (5/1) – MidProject Report due with flowchart, answers to questions, and PWM waveforms (11:55 PM thru Sakai)
- Wednesday (5/3, in class) – MidProject Demo
- Friday (5/12) – Final Project 5 Demo/Race in DuPont Lobby
- Monday (5/15) – Final Class, Hand out Certificates and Review for Final Exam