Journal Report 8
10/18/19-10/25/19
Hari Shanmugaraja
Computer Systems Research Lab
Period 5, White

Daily Log

Monday October 28

Tried reading analog values in from the radio. Since steering servo is analog, and Pi only has digital pins, it read in 0s and 1s, changing the analog signal. We experimented to see what would happen if we read the 0s and 1s in and tried writing 0s and 1s to analog, even though analog is a range of values. We were able to get a reaction out of the servo. That is, when we turned the tire right on the transmitter, the servo oscillated around the center and when we released the tire on the transmitter, the wheels went full left.

Wednesday October 30

After the presentations, we tried using an Arduino board, which can work with analog servos, to read in and write values to the servo. We are hoping that we will be able to link up the arduino and Pi, using the Arduino specifically to interact with the analog steering servo. We wrote a sketch that reads in analog signal and prints it out. Had trouble with computer not recognizing the USB cable plugged in.

Friday November 1

Hari was out today. After presentations, looked more into the USB cable issue. After some searching, we believe the cable we are using cannot move information, and only power. Next week, we plan on getting a USB B cable

Wednesday November 6

Used arduino to control steering servo. analogWrite() function that comes standard with the Arduino's PWM pins is much more precise than the PWM we were using with the Pi.

Friday November 8

Found a an arduino library called Servo.h. Not only does it let us write analog values to the steering servo super precisely, it also has the read() function. The read() function takes the wheels pointing straight as 90 degrees, with the positive direction going towards the left. We wrote a program today that will write analog values to the servo and then give us the angle it is at. We now realize that the read() function does not actually read the angle of the servo, but output the last write() call. Since we have spent nearly three weeks trying to read an analog signal from the radio, but have not been able to, we are considering separating the steering servo from the rest of the car when we train it. A possible way forward might be to wire a breadboard with 3 or 5 buttons, with each button representing some wheel angle. For example, one button could send a signal to the servo to turn to 90 degrees, while another might tell it to turn to 45 degrees. We might have to use the buttons as labels for our inputs.

Timeline

Date	Goal	Met
Today minus 2 weeks	Have the car move on its own via the pi interfacing with the esc	No, but we have a feeling it might work now that we have the Arduino
Today minus 1 week	Have the car navigate a hallway on its own with hard coding	No, we have not started we meed the previous step to be completed
Today	Have the Raspberry Pi gather Lidar data	No, we have not started
Today plus 1 week	Build a program where we throw a object in front of the car and avoids the obstacle	No, we have not started
Today plus 2 weeks	Gather training data with Lidar	No, we have not started yet

Reflection

We are seriously behind, as we have run into issues with reading the input that we give out to the car. Tony and I are considering two new options. Option 1 is to get rid of the radio, and control the car's steering using buttons that are connected on a circuit. The second option is to bypass the radio, setting up a localhost server on the pi, or maybe even source an Intel Edison from Mr. Kosek, and control the car via a website from Socket.io. Option 1 is easy, but option 2 is more feasible in the long run.