

## Daily Log

### Wednesday January 29

Discovered a new problem with our data gathering code. We have the Arduino print to serial, and when we read it in Python, it goes down in order rather than reading the last line printed to serial, which means the data about the car will lag behind after the first data point.

### Friday January 31

Tried to fix the problem by making recordtrainingdata write out two files, one for driving information and the other for LIDAR. We plan tried to the two files up with a timestamp and find the latest line printed to serial, but did not do it successfully. We also figured out why the time the LIDAR spent spinning was inconsistent, even though we tell it to pause every 500 data points to get information about wheel direction and speed. We found that the LIDAR program does not print values continuously; instead, it prints them out in batches of 200-250.

While we were hoping to start gathering training data this week, we found even more problems than we thought we had.

### Monday February 3

Began writing the two separate files - one to log lidar data into a text file, the other logging steering and motor data. We are switching because having two files sounds simpler than having one. In addition, the constant starting and stopping of the lidar sensor is not good for the sensor.

### Wednesday February 5

Continued writing the two data logging files. The lidar sensor is attached to the Raspberry Pi, where the text file will be created. On the other hand, the breadboard and arduino are connected to my computer, and a text file with steering and motor data will be created on my computer. Both files put time stamps using `time.time()`, and we plan on syncing them up. We know the clock on my computer and the Raspberry Pi is about two seconds off, so we plan on adding some offset to the times in the text files. We are planning to map lidar output to two integers: the value written to steering in `writeMicroseconds()` and the value written to the motor in `writeMicroseconds()`. The car's input pins will be wired directly to the breadboard, at least for now. I am still trying to get the Xbees to work, which would allow us to get rid of the direct wiring for a remote control.

### Friday February 7

Wrapped up the two data logging files. With the lidar data gathering program, I tried to code a feature to stop the sensor when a key is pressed, but did not finish it.

I also took xbees with me over the weekend and got the wireless to work perfectly. We now send throttle and steering data in one line formatted exactly how we wanted it to. The code involved to format our data properly lacks sophistication, but it was the only one of over 10 ideas to work.

## Timeline

Date	Goal	Met
Today minus 2 weeks	Have our data logging program work reliably	Yes
Today minus 1 week	Make the collection of data wireless using the XBees	Yes
Today	Make the collection of data wireless using the XBees	Yes
Today plus 1 week	Control the car wirelessly using our controller	No
Today plus 2 weeks	Control the car wirelessly using our controller	No

## Reflection

This week we made more significant progress, but as the school's end nears, we realize the speed we are currently progressing is not enough. We have started to take equipment home to expedite the progress, and this has begun to put us back on course.