RIDWAN HOSSAIN
100747897

2022/05/20

- Researched how to communicate with COM ports in Linux, as it is not referred to COM3 or COM4 in Linux systems
- Discovered that the serial ports COM3 AND COM4 are referred to as /dev/ttyS2 and /dev/ttyS3 in Linux. Switching to these port references raised a ‘serial.serialutil.SerialException: Could not configure port: (5, 'Input/output error')’ error. To resolve this, researched how to remedy this error.
- Discovered using the ‘dmesg’ command that the arduino boards are connected to ports /dev/ttyACM0 and /dev/ttyACM1. Upon switching the port references to those, the port error goes away and the code runs. However, the speed and rpms do not read on the gauge cluster, even though the gauge cluster is most definitely connected, as indicated by the gauge c;uster buzzing every time the code is run.
- Through manual testing, it was discovered the speedometer moves when the left turn signal is activated. From this, it is clear that the gauge cluster is connected, however, the signals are most likely being sent to the incorrect pins on the arduino board.
- Looked through the previous capstone team’s button mapping configuration to see which connection the left turning indicator is assigned to.
- Discovered the left turn signal is associated with the joystick button event that produces 0 in the terminal window.Therefore, areas of interest are at points which ‘event.button’ is equal to 0.
- Through trial and error, discovered that /dev/ttyACM0 is responsible for the connection to the tachometer (rpms) and /dev/ttyACM1 is responsible for the speedometer (speed)
- Figured out how to send a value to the speedometer. Now need to figure out how to correctly map the vehicle speed in Carla to the correct value for the speedometer. Went through the previous capstone groups ‘cluster.ino’ arduino file to see what values the cluster should receive.
- It is possible that the arduino board may not have the correct program loaded into it when in Linux, but is running the correct program when in Windows. Therefore, I am going to download the Arduino IDE on Linux to reupload the ‘cluster.ino’ program onto the board.
- After setting up the Arduino IDE and granting read/write permissions to the serial ports the boards are connected to, the cluster.ino programs were uploaded. However, after uploading, no changes can be found in the behavior of the gauge cluster. Furthermore, after uploading, the serial ports the boards were connected to changed to /dev/ttyACM2 and /dev/ttyACM3, so I went back into testbed.py and changed the port references.

- It seems the next steps are to make a program to be uploaded to the board from scratch, while reverse engineering the previous capstone team's Arduino files. This entire process would be much easier if the previous capstone team's documentation was not so abysmal.

- Observed that unplugging the board and plugging it back in may cause the port numbers to switch, make sure to update the port references each time that happens. Use the 'dmesg' command to obtain serial port information.

- First step is to be able to send values to the Arduino board and read them. I will make a sample program to see if I can obtain the speed value from Carla on the Arduino end.

- Was able to successfully communicate values to the Arduino board and have them read on the gauge cluster. Next step is to send correct speed values to the Arduino board as the existing get_speed function is sending strange values to the board for some reason (may need to create another function to obtain speed)