Rob Chirpich – 2896710 – EECS388 – Project Report

|  |  |
| --- | --- |
| A screenshot of text  Description automatically generated  A close up of text on a black background  Description automatically generated | The purpose of this lab project was to design an implementation for an (almost) self-driving car which could be accomplished by utilizing a plethora of wires, a HiFive1 board, a Raspberry Pi 4, a Linux PC, a servomotor, and a Lidar sensor. According to the write-up, the wiring of this set-up was to be done by the TA’s, but unfortunately, not all of the stations were wired correctly, which caused a bit of grief when the program was not functioning as it was expected to. To fix this, some minor adjustments had to be made to the wiring via removal of a few excess wires and then making sure that the Lidar sensors power and ground lines were connected to the 5 Volt and Ground lines on the Raspberry Pi 4 as well as connecting the servomotors power, ground, and PWM lines to the HiFive1’s 5 Volt, Ground, and PIN\_19 lines. Once this was completed, then all code will be able to function properly.  To the left is the entirety of the comm.c file for this project. First, a servo function was created to control the servomotor such that it would pulse for the duration of a turn from the provided video which was used for input in this project. The servo function utilizes the maximum pulse, minimum pulse, and the period of the servo to control the length of the pulse and delay of the servomotor.  Next, the set-up of the UART channels, distance, angle, and PINs was completed and initialized so that the main loop (the while-loop in this implementation) could function properly. The first thing to occur in this while-loop is the calculation of distance from the Lidar sensor to an object located in front of it. |
| ... CODE WAS GIVEN IN WRITE-UP ...  import struct  ... CODE WAS GIVEN IN WRITE-UP ...  ser1 = serial.Serial(port='/dev/ttyAMA1', baudrate=115200)  ser2 = serial.Serial(port='/dev/ttyAMA2', baudrate=115200)  while(1):      if curFrame < NFRAMES:          cam\_start = time.time()            ret, img = cap.read()          if not ret:              break          prep\_start = time.time()          img2 = img          img = cv2.resize(img, (200, 66))          img = img / 255.          pred\_start = time.time()            rad = model.y.eval(feed\_dict={model.x: [img]})[0][0]          deg = rad2deg(rad)          ser1.write(struct.pack("<f", deg))          pred\_end = time.time()  **cv2.putText(img2, str(deg), (0, 25), cv2.FONT\_HERSHEY\_SIMPLEX,**  **1, (255, 255, 255), 2, cv2.LINE\_AA)**  **cv2.imshow('Video with Angle', img2)**  **cv2.waitKey(1)**  ... CODE WAS GIVEN IN WRITE-UP ...  ser1.close()  ser2.close()  ... CODE WAS GIVEN IN WRITE-UP ... | If the calculated distance is greater than 50 cm, then the program determines if the Raspberry Pi 4 is ready to output an angle, if it is, then an angle is read-in from the Raspberry Pi 4 which can then be used to determine which way the servomotor must turn in order to keep the vehicle within the bounds of the track in the video. If this angle is between -30 and 30 degrees, then the LED on the HiFive1 is turned on so that a white color is displayed, signaling that the vehicle is in the “center” position. If the angle is greater than 30 degrees, then the LED on the HiFive1 is turned on so that a green color is displayed, signaling that the vehicle is turning “right”, otherwise the LED on the HiFive1 is turned on so that a blue color is displayed, signaling that the vehicle is turning “left”. In either the green or blue LED case, the servomotor is turned on via a call to servo() with the current angle that results in the servomotor acting as a steering wheel for the vehicle. If the distance was less than 50 cm, then the LED on the HiFive1 is turned on and displays a red color and the servomotor stops running until the object is removed from the Lidar sensors view.  In the dnn.py file, a DNN inference occurs every 50 ms which reads the video image and then writes the angle to the HiFive1 via serial channel. Next, cv2 is utilized to display the video with an overlay depicting the steering angle, this is accomplished by the bolded text on the left. It should be noted that the HiFive1 is reading the distance from the Lidar sensor and the steering angle from the Raspberry Pi 4 concurrently so that the vehicle is able to drive and steer at the same time. |