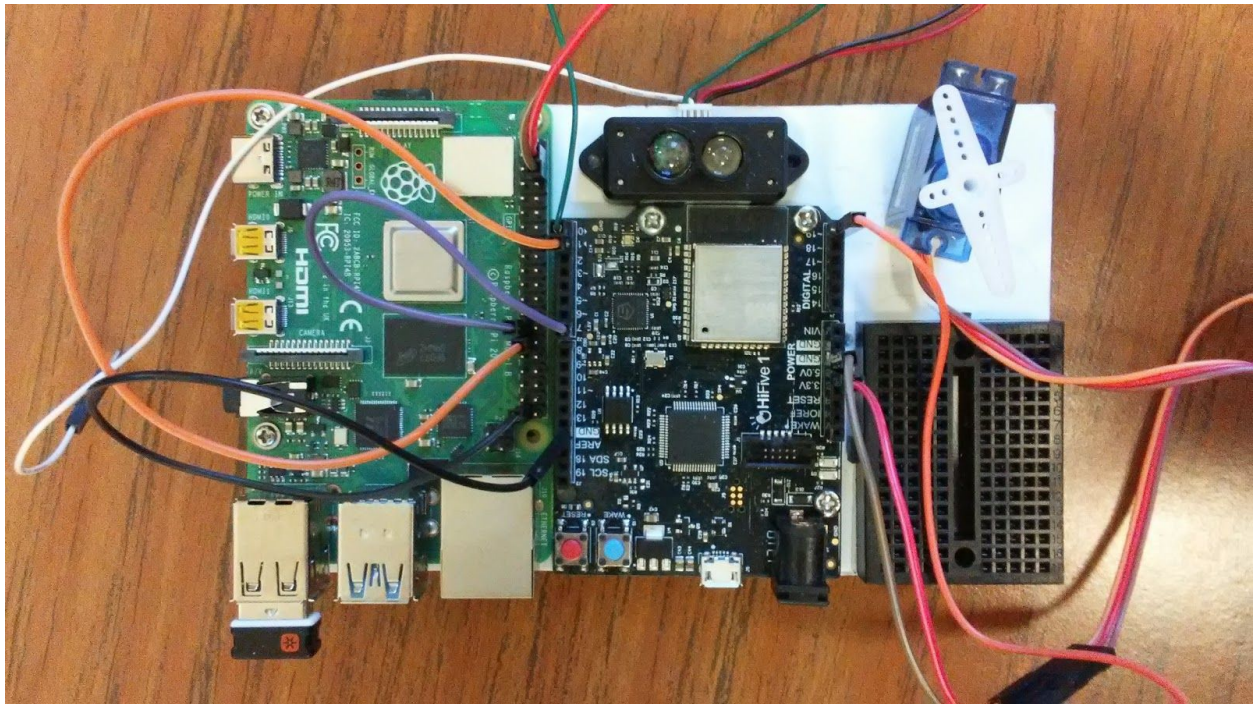


EECS 388 #Project

# Putting All Things Together: (Almost) Self-Driving Car

In this project, you will put all things together to create a self-driving car system prototype. (Note that, although we don't use a real camera here, you can easily attach a webcam to Pi 4's USB port, and open the camera device instead of the video file to process real-time video.)

## Part 0: Setup the system (for TAs)



Connect Lidar's TX to UART0 RX (pin 0). Lidar's power and ground lines may be connected to the Pi 4's 5V and GND lines, respectively.

Connect the servo motor's Vcc (red, center), ground (brown), and PWM (orange) to the HiFive1 board's 5V, GND, and PIN 19 lines, respectively.

Serial connections between the Pi 4 and the HiFive1 are the same as Lab #10

/dev/ttyAMA1      Pi 4 → HiFive1: Send steering angle to HiFive1 (uart1).  
/dev/ttyAMA2      HiFive1 → Pi 4: Receive HiFive1's console (uart0) output

## Part 1. Programming the HiFive1 board

In your system, the Raspberry Pi 4 will perform a DNN inference at every 50ms using a captured camera image (read from the stored video file) to produce the car's steering angle. The steering angle value is then sent to the HiFive 1 board.

The HiFive1 read the steering angle data from the Pi 4 and control the servo motor based on the angle value. In addition, if the received angle is smaller than negative 30 deg, a blue LED should be on to indicate "left". If the angle is bigger than positive 30 deg, a green LED should be on to indicate "right". Lastly, if the angle is between -30 and + 30 deg, all three red, blue, green LEDs should be on (which will produce the white color), to indicate the car is in "center" position.

At the same time, the HiFive1 also read the distance data from the connected Lidar. And if the read distance is below 50cm, it turns on the red LED and stops the servo motor.

Note that HiFive1 should deal with multiple inputs (pi4 and lidar) and outputs (servo and leds) simultaneously.

Once you complete the system, shows it to your TA.

## Part 2. Programming the Raspberry Pi 4 board

Your task is to modify the dnn.py so that it can display the current video frame and the DNN's output angle onto the monitor screen in real-time. Think of this as making a car's cockpit UI. You can use any UI library you want to use to accomplish this.

## Grading

Part 1: 50%

Part 2: 20%

Report: 30%