

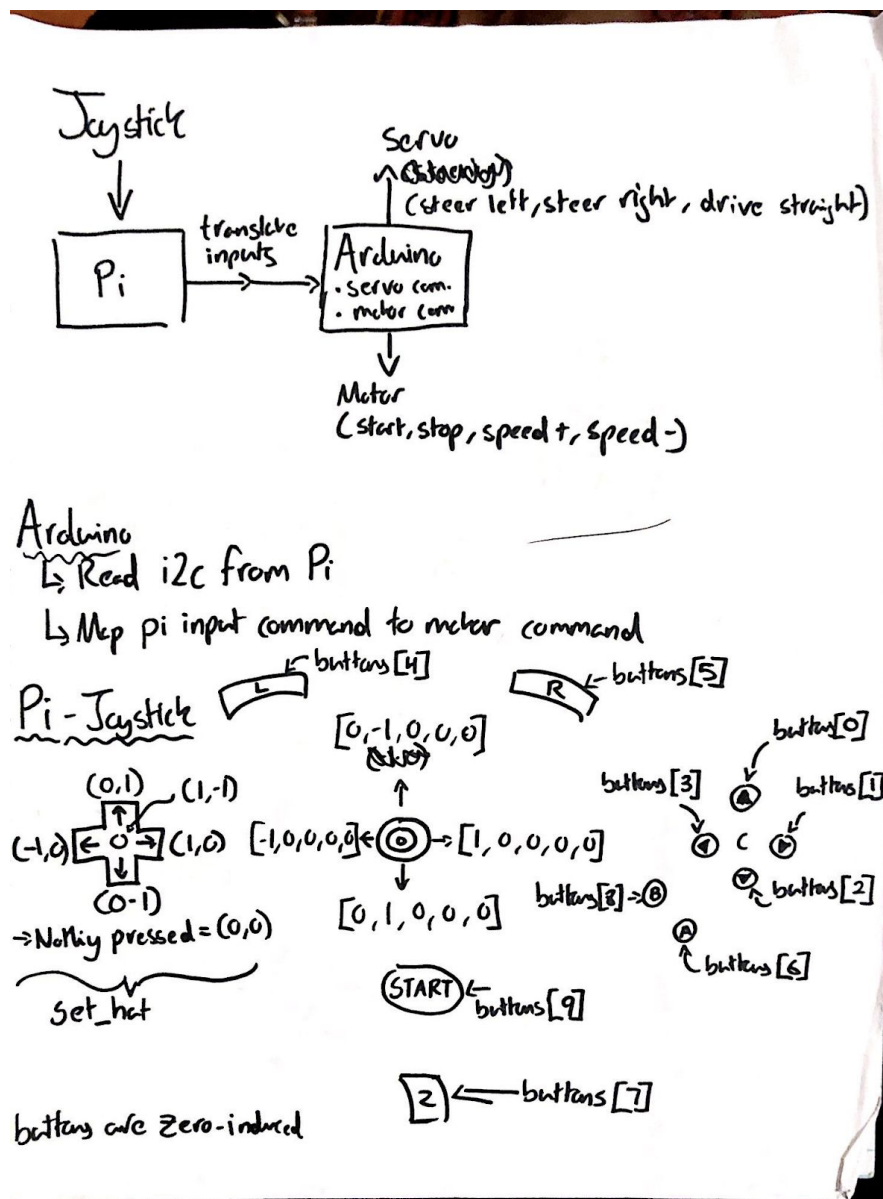
ESE 421: Code Instructions (11/10/18)

Adnan Jafferjee and Saumeel Desai

joystickControl.ino and SendJoystickCommandsToArduino.py

What you see is what you get. Run joystickControl.ino on Arduino and SendJoystickCommandsToArduino.py on the Pi simultaneously.

The image below maps buttons pressed on the joystick to number combinations on the input arrays that the joystick reads. Mappings between the joystick buttons and corresponding commands have been commented into the joystickControl.ino Arduino function.



HSV_CV.py

HSV_CV.py is our most functional road detection Python script. It does the following:

- Takes in an image of a road (**change the directory that you read images from at the top of the script**)
- Converts it to HSV color space
- Applies lower and upper thresholds that separate regions with colors (defined in HSV space) that are similar to the road into a mask¹
- Performs Canny edge detection on a selected region of interest (bottom right corner of the original image) of the mask
- Uses Hough line detection to pick out desired lines from the image
- Returns calculated Ψ_R and X_0
- Plots detected line on image, as well as plots of intermediate detection steps

We still need to implement a heuristic to choose one line (as opposed to several) from the detected edges. Likely picking the longest line is a good enough “dumb” heuristic for right now. We also need to send the best estimated line Ψ_R and X_0 to the Arduino for heading control.

headingControl.ino

Just upload onto your Arduino and run it! There may be bugs in communicating with the PiCamera as we have yet to test that section. The filter should default to relying on the gyro if that is the case. Control gains and time estimates were tuned based on GPS feedback, they need to be adjusted for the camera.

¹ The thresholding portion was adapted from code that Ryan Kortvelesy shared on Piazza. Thanks Ryan!