MEAM510

Brandon Joel Gonzalez

Lab 4.2 - 11/6/19

PennKey: bgonz

We're Team 7, and our team is "CMPE ROCKZ". My partner is Hyun Ji Jung (Pennkey: hyju).

1. **Programming Design**

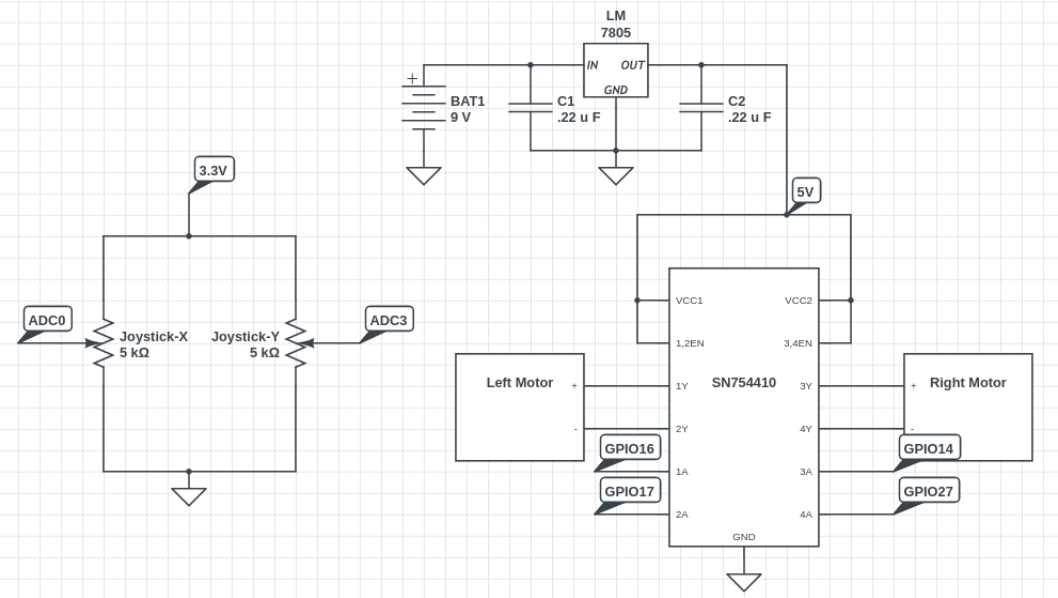
   We broke our code into two files: *Controller* and *Car*. Both of these files are submitted on a zip file on Canvas.

   The *Controller* code was responsible for getting two ADC readings from the joystick, similar to what was done in Lab 4.1. It then converted these ADC readings to corresponding motor commands. Depending on what reading we got, we decided to do one of five things. If the stick was pointed up, then we moved the car forward. If the stick was pointed down, we moved the car backwards. And if the stick was pointed left or right, then we turned the car appropriately. If the stick was in the neutral position, then the car did nothing. This mapping of ADC value to car direction was then sent to the car over UDP, similar to Lab 4.1.

   The *Car* code then listened for these packets, which held the commands for the motors. Once the car received the message, it would go and set the motors appropriately. We had five different functions to handle setting the GPIO motor pins for each appropriate action. As such, we had a system that could read joystick inputs, transmit those commands to the car, and control the car's motors as directed. Neato!

2. **Electronics Design**

Below is a schematic of the circuitry used in our racecar.

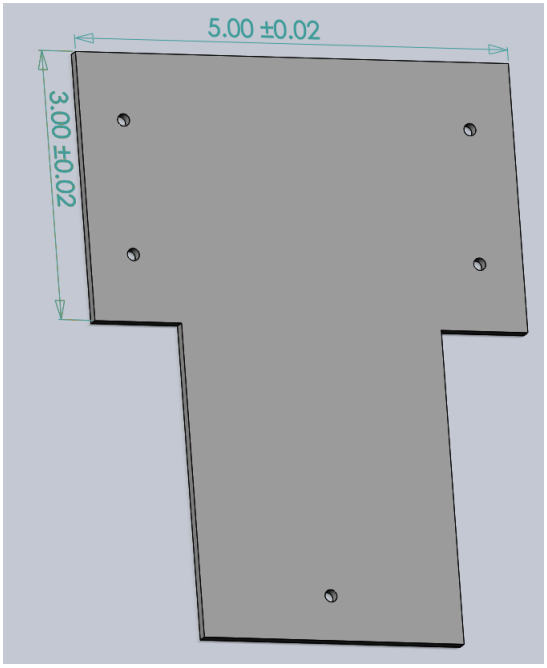


The section on the left is our controller. We used a joystick, which is essentially two potentiometers, to read voltages into our MCU via pins ADC0/ADC3.

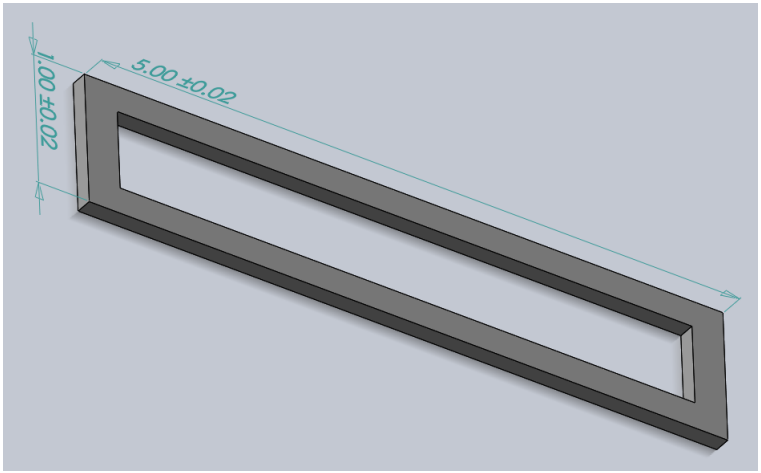
On the right side, we have the circuitry on our car. The car is powered by a 9V battery, which we pass through a linear regulator to get a 5V power source to drive our motors. We then feed this 5V source to power our motor driver. Additionally, we connect our motor driver to GPIO16/GPIO17 to control the left motor and GPIO14/GPIO27 to control the right motor. These two motors are connected to the driver to be rotated in one direction or the other as needed.

3. **Mechanical Design**

Below are images of the parts that we used in our project. We discuss each one in sequence:



This is one of the decks of our car - there were two. The upper deck was mounted on top of the lower deck via standoffs and screws. The upper deck had the breadboard on top, secured by tape. For the lower deck, the motors were mounted on the bottom part of this deck, taped down to the board to ensure stability. The wheels were then attached to the motors. The lower deck also held the battery pack, fastened by more tape.

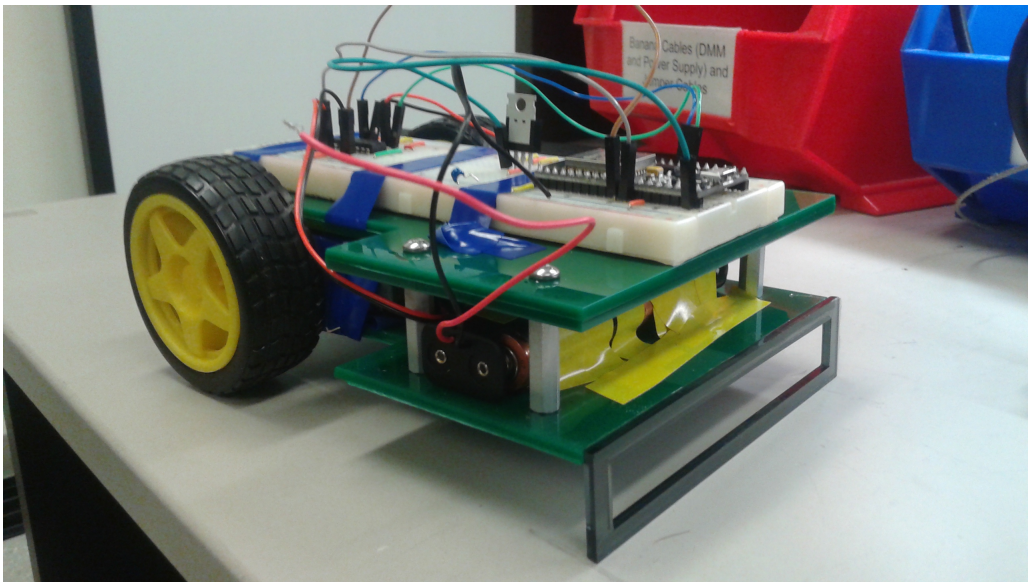


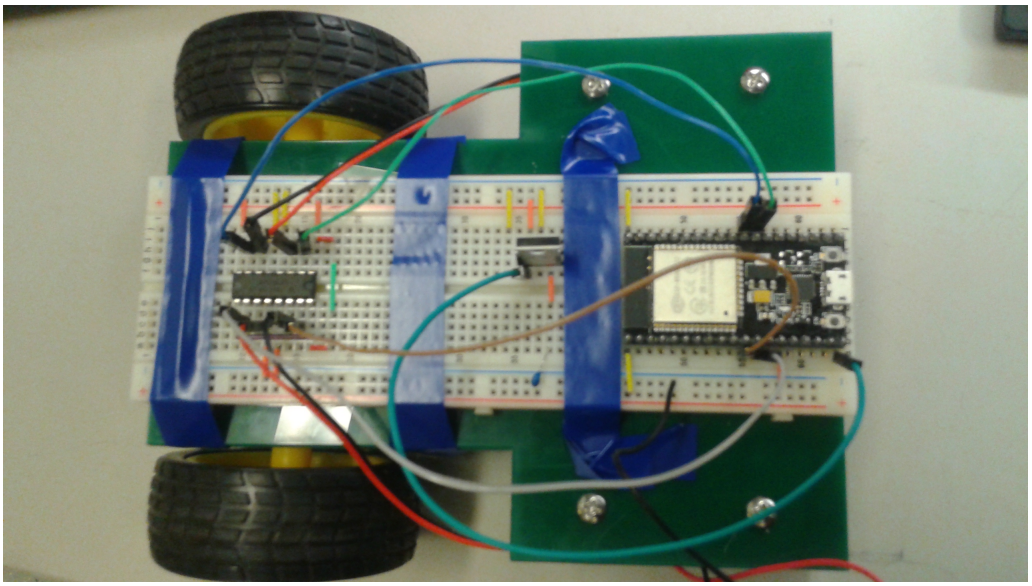
We also attached a bumper to the bottom of the lower deck via glue. This was to ensure that the car didn't drag along the ground as it drove, since this would make it struggle harder to climb an incline. The bumper provided some extra

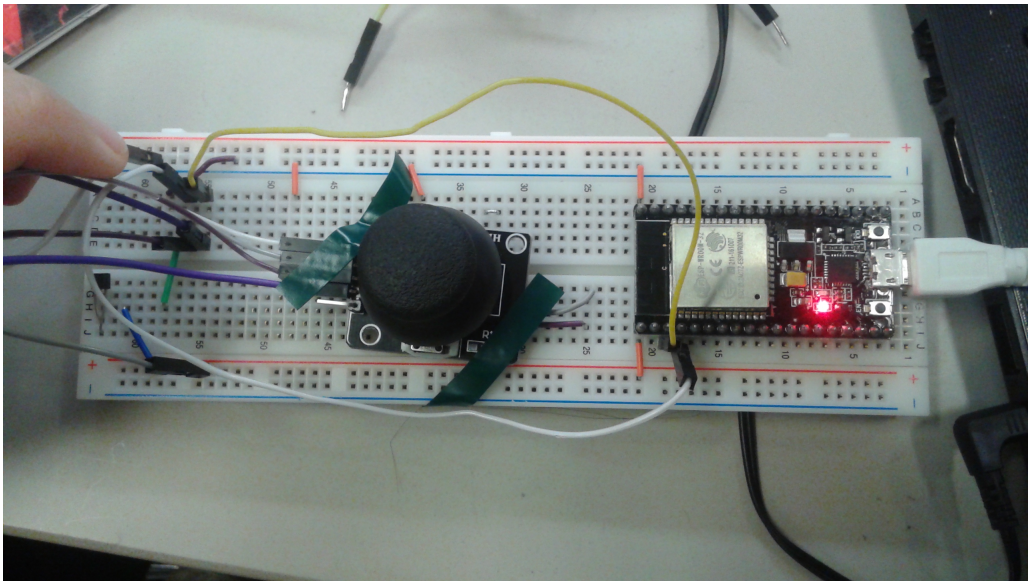
space to lift up the car a bit.

We used a relatively simple driving mechanism consisting of two wheels. If we wanted to move forward, both wheels turned forward; likewise, both wheels turned backwards to go in reverse. To turn left or right, one wheel turned forwards while the other wheel turned backwards. This seemed to work well enough.

Below are two images of *our car*, and *our controller*!







Wow!