24-354 Self Driving Robot- Final Report

# Summary

*The objective of this project is to create a robot capable of navigating a course with a variety of obstacles using a variety of different sensors and measurement metrics. The design criteria for the robot include path navigation through a course with multiple 90 degree bends, color and obstacle detection, turn indication, position and velocity estimation, and control through a remote device, also known as ‘untethered’. These deliverables will be accomplished through a variety of different sensors, motors, and LEDs. A control algorithm will be implemented to determine the robot’s path and movements based on ultrasonic readings. A ring of LEDs fixed to the robot will indicate any turns that need to be made, or if the wall color has changed.*

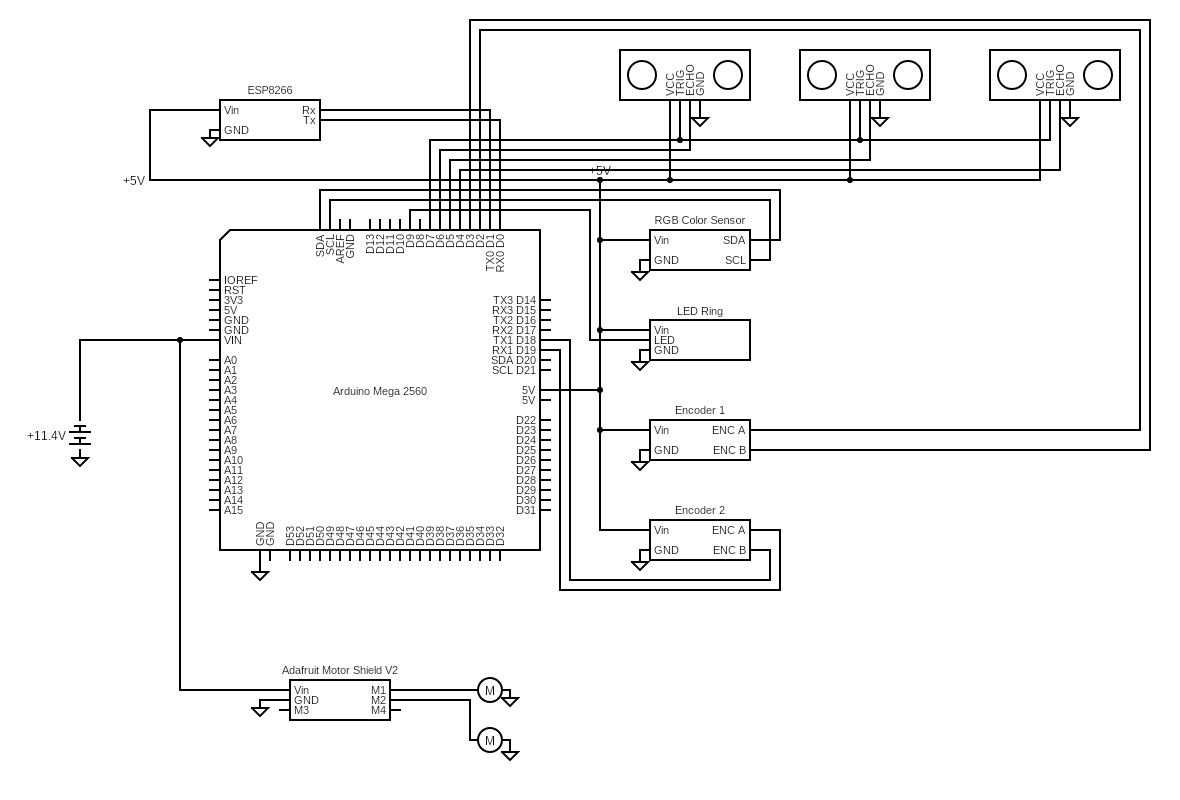
*When tested, our robot met all of the design criteria and navigated the course successfully. It also was able to indicate when a wall was colored red vs. blue. Lastly, this project was completed under the predetermined $150 budget.*

# Electrical Design

There were four major domains of this project handled with electronics: Control Logic, Actuation, Sensing, and Communication. All of the control logic was handled onboard an Arduino Mega microcontroller. This controller was selected because it was able to provide us with a sufficient number of digital/analog/interrupt pins, output current, and output voltage to support all of our hardware. Actuation was handled using an Adafruit V2 Motor shield, which allowed us to drive two 12V geared motors with a current exceeding the 200mA limit of the Arduino. A 11.1V 450mAh 45C LiPo battery was selected in order to meet the power demands of all electrical systems on the robot for a duration far exceeding a 60s test. We selected 12V geared motors (20:1) with built in quadrature encoders to meet both the speed and torque requirements of the track, as well as increasing our ability to measure wheel rotation.

Sensing required a means to measure the robot’s position/velocity, distance from the track walls, and the color of nearby walls. Each motor utilized a quadrature encoder, requiring two interrupt pins for each motor with a total of four. Pins 2, 3, 18, and 19 were used, as the Wifi module required pins 20 and 21 for serial communication. Wall distance sensing was done using three HC-SR04 ultrasonic sensors mounted on the front left and right side of the robot. In order to conduct distance sensing in a non-blocking manner, the ultrasonic sensors were attached to digital input pins which could take advantage of the PinChangeInterrupt library, allowing ‘non-interrupt’ pins to detect changes in state as if they were interrupt pins. We used a TCS-34725 RGB light sensor for color sensing to give us the ability to detect color changes at a range >10cm as well as its cheap cost and high degree of support in documentation.

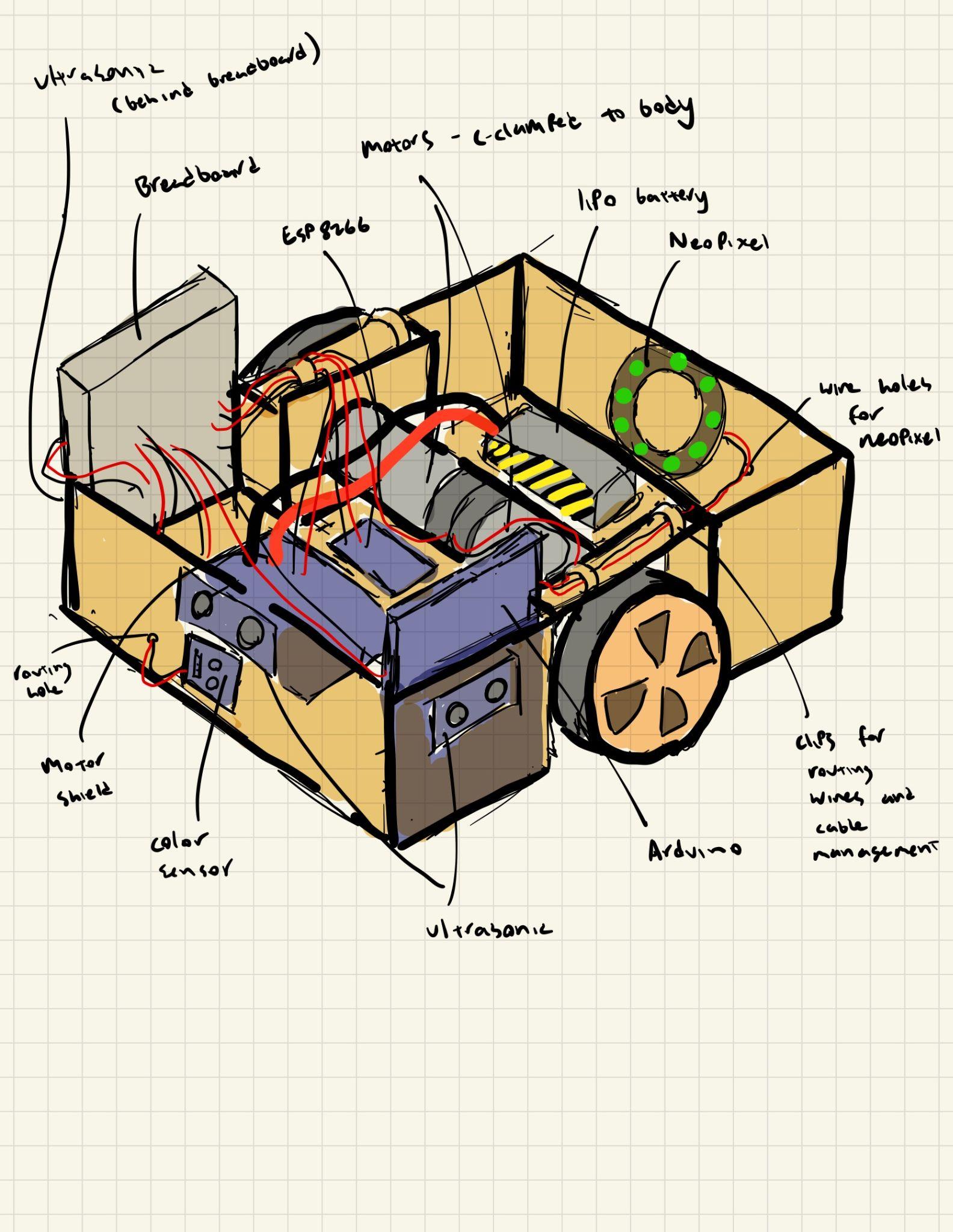
Communication was implemented using both Wifi transmissions of logged sensor data, as well as visual LED signals to indicate the robot’s planned actions. Wireless logging was implemented using an ESP01 Wifi module due to its relatively low cost and power requirements. Data would be sent through serial communication from the Arduino to the ESP, and then sent wirelessly through UDP to a python server running on a laptop. The data could then be printed out for debugging or saved to a csv file for later analysis. To indicate actions such as path correction, sharp turns, or color changes, a 12 LED neopixel ring was used. The ring was selected, as the 12 LEDs allowed us to implement a large set of pixel shapes/colors to indicate various sensor readings. See the ‘pizzaz’ section for more information.



*Fig. 1 - System circuit diagram*

# Physical System Design

Our design goal was to create a compact and reliable mechanical system to allow for easy and repeatable course navigation. We kept the width of the car body to roughly 20 cm as the width of the course was only 23.5cm and we wanted as much clearance as possible on either side. It was also designed with serviceability in mind, in the event that we had to diagnose electrical issues. There are a couple important aspects to note. First, the location of the wheels and hardware components inside the car are shifted to the middle of the body so that the center of mass will be almost directly over the driving motors. This is to ensure that the car has the lowest possible risk of slipping while on an uphill incline. The wheels driven by our motos were custom 3d printed using PLA and TPU. The diameter of the wheel was chosen based on the required course completion speed of 60s. We also included a ball bearing mounted under the car body towards the rear of the vehicle to ensure bar balance. Secondly, there are holes in the body that allow electrical components such as the arduino and the neopixel to stay securely in one place, while still being accessible for pin swapping and breadboard prototyping. We split major systems into different breadboards for easier troubleshooting. This proved useful when we encountered issues with our arduino not picking up signals from the echo pins on our ultrasonic sensors. If we had more time, we would have switched to a protoboard once our breadboard prototype was finished, especially because we had issues with wires falling out of the breadboard in the weeks leading up to testing.



*Fig. 2 - labeled model of physical system.*

*CAD drawings of the car body and wheels can be found in the appendix.*

Our motors were selected based on the required stall torque and no-load speed. The required torque was determined using simple static and dynamic hand calculations (found in appendix). We also determined the speed our motor needed to run at under load based on the time requirement for course completion. The torque calculation assumes a 0.5 kg car mass, a 10cm wheel radius, and 5% grade with no forces except for gravity. To account for inertial and frictional forces, we searched for motors with a torque rating of at least 0.01 Nm (10 x 10-3). The velocity calculation assumes a steady velocity across the entire track with 10cm wheel radius. To account for acceleration and stoppages, we search for a rated speed of at least 250 rpm. This yields a theoretical FoS of ~10 on torque and ~2.5 on speed operating each motor within rating.

| 1.22 x 10-3 Nm (12.44 x 10-3 Kg-cm) | 103 rpm |
| --- | --- |
| 𝛕desired = 10 x 10-3 Nm | ⍵desired = 250 rpm |

# Testing Approach and Results

## Estimation of Uncertainty

We used dead reckoning to estimate the position of the robot with a no-slip wheel assumption. This creates uncertainty in distance measurements, as wheel slippage is known to happen in the real world. To estimate the uncertainty, we commanded the robot to drive forwards 90cm for 10 trials, measuring the true displacement each time. We found that the average error for a 90cm trajectory was 0.62cm, yielding an uncertainty rate of 0.7% for distance traveled.

In terms of measuring the uncertainty of our ultrasonic sensors, we verified that each sensor could detect the distance from a wall within during the testing of our sensor platform. A table summarizing these results can be found in the appendix.

## Course Navigation

Prior to testing, we ensured that the ultrasonic sensors were working properly by gauging the LED ring state while positioning our hands at various distances from the sensors. In a similar manner, both a blue and red wall were placed in front of the color sensor to ensure that color was read properly. Additionally, we tested the robot on isolated corner segments to ensure that corner turning events would be triggered properly.

The robot was deployed on the test track a number of times. Due to the autonomous nature of the project, not much was done while testing was conducted, however we did take videos to monitor the performance of the robot as well as the LED indications of the robot’s planned motions. Track testing was mostly successful with all systems functioning as intended. These videos ([top view](https://drive.google.com/file/d/1CBtHd5NBnwomrab4rudmcJJ-_8MGw21M/view?usp=share_link), [rear view](https://drive.google.com/file/d/1Kib2lTRF1KeOozwrpPFM0bvo9Wr-hHvU/view?usp=share_link)) of a successful track test demonstrates the robot’s ability to meet the 60 second speed requirement, ability to detect a red wall (indicated by a purple :) on the ring), and the ability to traverse a 5 degree grade both up and down.

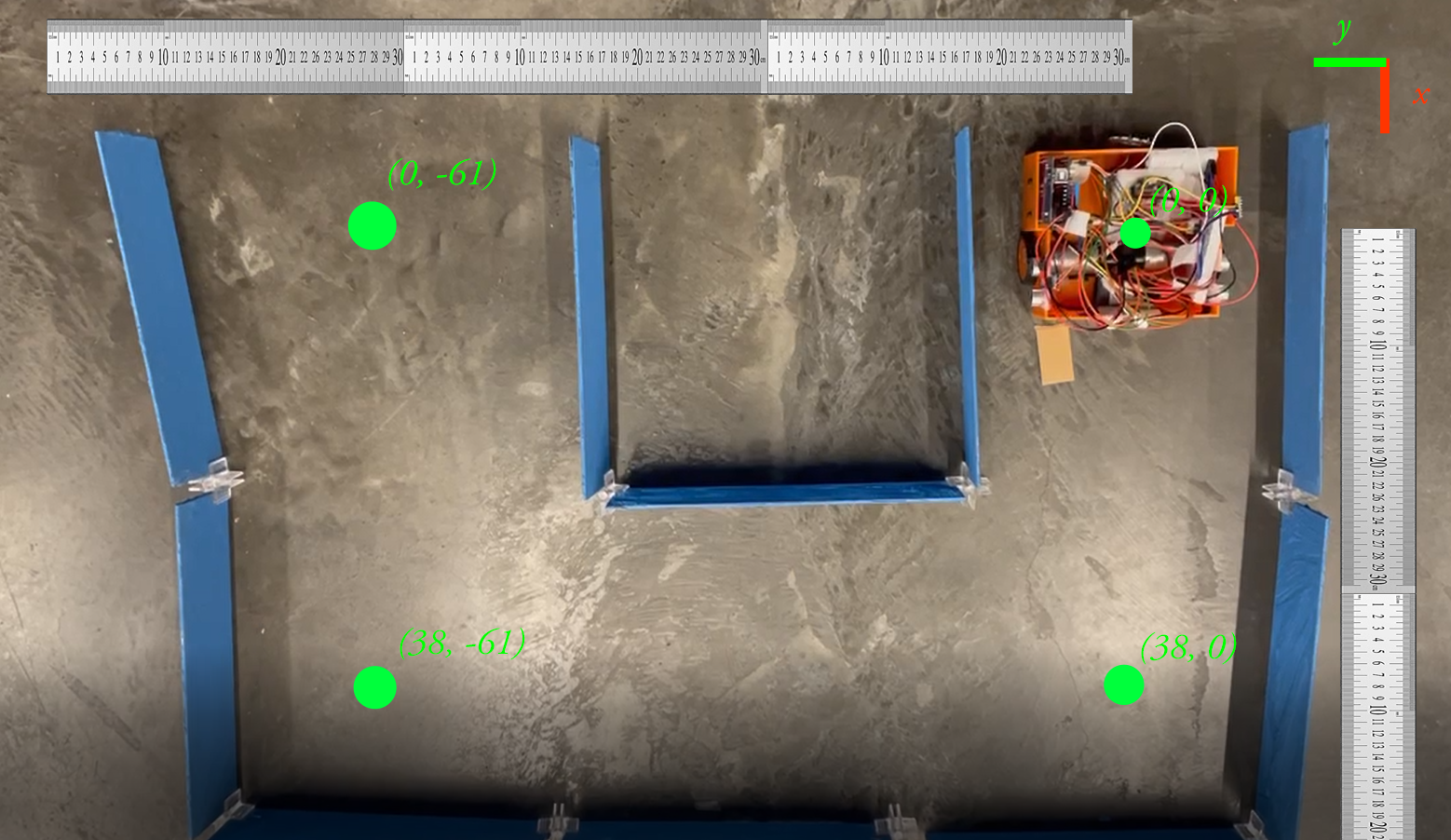
****

*Fig. 3- A pink smiley face indicates that a red wall was detected, and the course was completed.*

There were a number of trials in which the robot would turn too much in the straight wall segments, causing failure in the corners. Further tuning in the PID controls could be done to increase the repeatability of our algorithm. Another source of failure in testing was ‘random stoppages.’ After some investigation we determined that this was caused by drawing too much power from the Arduino (200mA limit), and this problem was eliminated by decreasing the brightness of our LED signals.

## Wireless Data Logging

Data Logging was implemented wirelessly using the ESP01 Wifi module. The Arduino would first compute sensor values and state estimation values and create an array of floats in the form of [timestamp, x, y, yaw, speed, US\_L, US\_F, US\_R]. These values were then converted into a buffer of bytes, sent to the ESP01 over serial, then sent to a python server over UDP. The python server would receive each buffer, unpack the values back into floating point numbers, and save the values to a .csv file for later use. The data logging module was tested separately from the primary track test with the below image showing 4 waypoints used to validate the quality of our uploaded data. The waypoints are visual estimates for robot position during the test and may be imprecise. A higher fidelity test would require grid lines on the floor to accurately pinpoint the robot’s position in worldspace as it drove. Follow these links for a full [video](https://drive.google.com/file/d/1CWYPMOFFQXfwwg1u7zzY7qb77Y4llg-x/view?usp=share_link) of test and [log](https://drive.google.com/file/d/1uqhFyg3MVA67pBxW_S4h4IjgvWamHqnz/view?usp=share_link) output.



*Fig. 4 - Logging Test with Visual Overlay*

**Table 1 - Logged (x,y) values at each sample point**

| **Waypoint** | **(0,0)** | **(38,0)** | **(38,-61)** | **(0,-61)** |
| --- | --- | --- | --- | --- |
| **Log Value** | **(0.00, 0.00)** | **(34.45, 0.78)** | **(38.58, 59.24)** | **(0.18, -64.49)** |

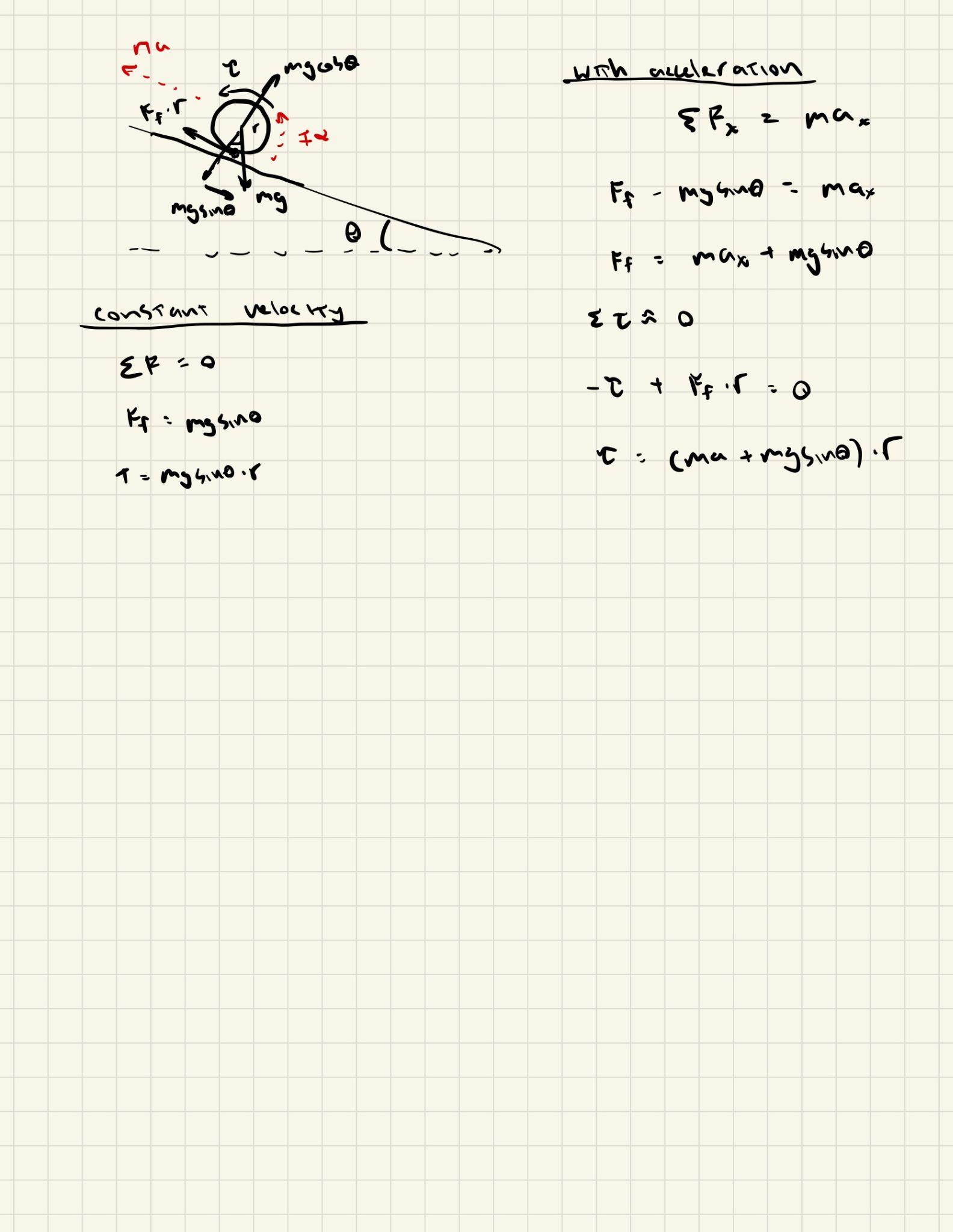
# Obstacles Encountered

We encountered a number of obstacles while designing and building our final prototype. The first obstacle was the lack of interrupt pins. Our motors each required two interrupt pins, meaning there were only enough extra interrupt pins for two ultrasonic sensors before we reached the maximum of 6 interrupt pins on the arduino mega. We solved this issue by installing a 3rd party arduino library PinChangeInterrupt that converted all pins to interrupt pins. This allowed us to use all 3 ultrasonic sensors along with our moto encoders.

The second, and most notable obstacle we encountered was with the power supplied to our system. The number of sensors and LEDs being used caused the robot to operate near the arduino’s current limit of 200mA. This was causing our arduino to heat up and stop reading our sensors. We tried many different approaches to fix this, including operating the robot at different levels of battery charge, plugging our battery into different locations in the motor shield, and repeatedly checking our pin connections to ensure that our system has not short circuited. Eventually, we discovered that reducing the brightness of the LEDs in our neopixel reduced the amount of current drawn from the arduino enough to allow it to operate normally.

# Appendix

**Torque calculations**

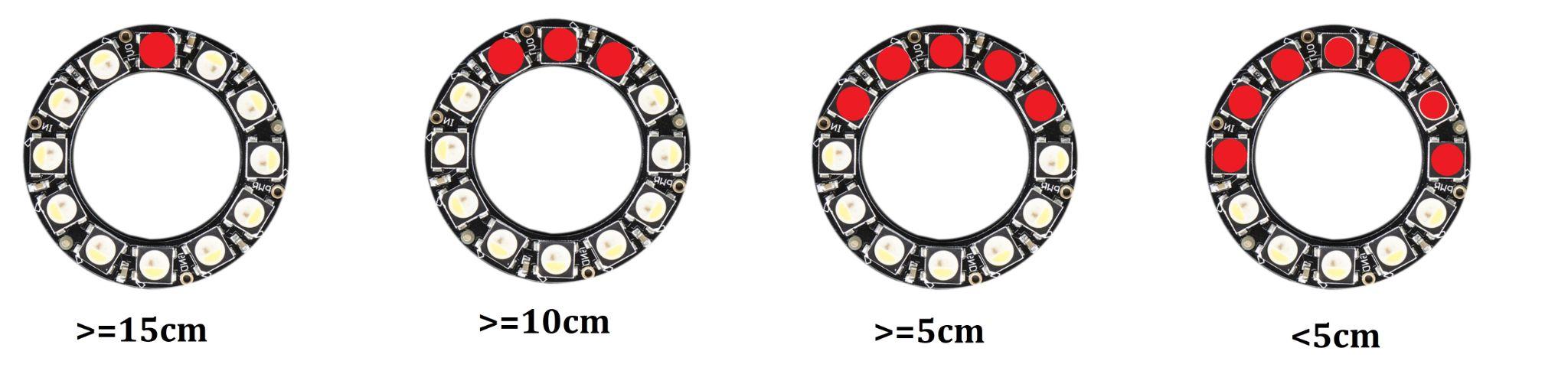


*Fig. 4- Calculations to determine stall torque at steady state and a chosen acceleration.*

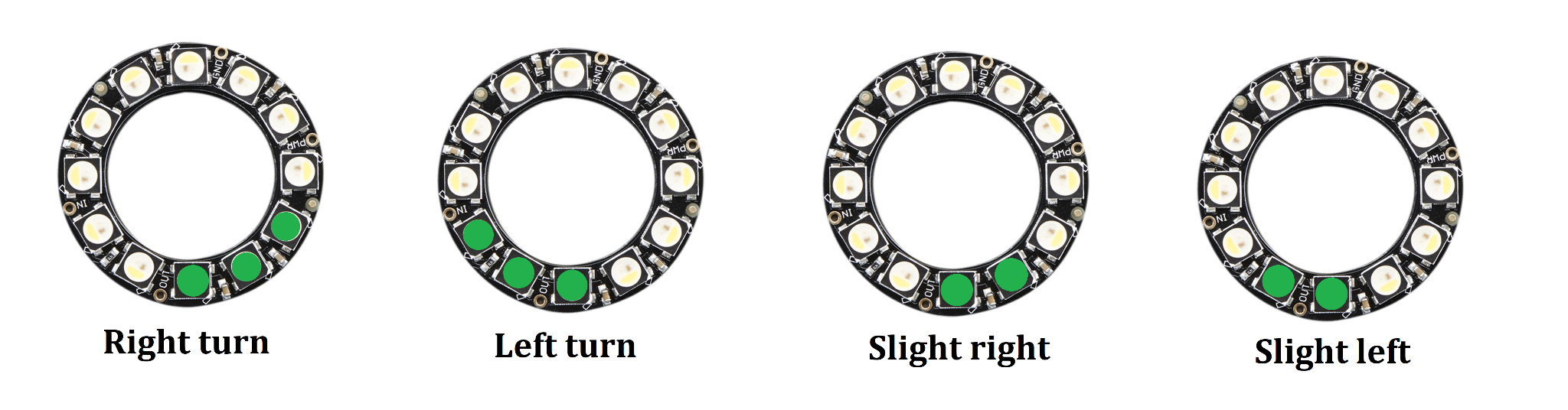
**Table 2- Bill of Materials and selection criteria**

| Component | Price | Role | Selection criteria |
| --- | --- | --- | --- |
| **1x Arduino Mega** | $20.00 | Microcontroller | Provided |
| **2x ESP8266 Wifi Module** | *$5.50 x 2*  $11.00 | Remove tether from robot | Provided |
| **1x Adafruit Motor Shield V2** | $17.00 | Regulates power | Provided |
| **3x HC-SR04 Ultrasonic Distance Sensor** | *$2.00 x 2*  $6.00 | Measure wall distance | Cheap and accurate |
| **1x Adafruit RGB Ring**  L[ink](https://www.amazon.com/Adafruit-NeoPixel-Ring-Integrated-Drivers/dp/B00KBXT9I0/ref=sr_1_1?crid=MS3B3X5EUM31&keywords=adafruit+rgb+ring&qid=1668129221&sprefix=adafruit+rgb+rin%2Caps%2C77&sr=8-1) | $14.64 | Indicate robot actions and environment | Display versatility |
| **2x Geared Motor with encoder**  20:1 gearing  12V  300 rpm rated speed  0.5 kgcm rated torque  [Link](https://www.amazon.com/gp/product/B07GNGCNK1/ref=ox_sc_saved_image_5?smid=A15GNHST7KG47K&psc=1)  12 fin quadrature encoder pair (960 ppr) | *$14.89 x 2*  $29.78 | Drive wheels and record distance traveled | Desired torque speed curve |
| **1x RGB Light Sensor**  [Link](https://www.amazon.com/gp/product/B0749GNL56/ref=ox_sc_saved_image_2?smid=A30QSGOJR8LMXA&psc=1) | $6.99 | Record wall color | Cheap and accurate |
| **1x LiPo 11.4V 3S Battery**  [Link](https://www.amazon.com/gp/product/B0714GN73V/ref=ox_sc_saved_image_3?smid=A35GESQNTB15AT&psc=1) | $12.34 | Power system | Cheap and correct voltage |
| **1x 12V to 5V DC Converter**  [Link](https://www.amazon.com/gp/product/B08VHZJ3C8/ref=ox_sc_saved_image_4?smid=ANYJQ8VGX4SR1&psc=1) | $7.99 | Convert power source for arduino | Convert desired voltages |
| **2x Speed Sensor** | *$1.50 x 2*  $3.00 | Initial encoders (sensor platform project) | Cheap and easy to work with |
| **3D Prints [PLA and TPU]** | $5.81 | Car body and wheels | Cheap and customizable |
|  |  |  |  |
| **Total** | $134.55 |  |  |

**LED turn indications**



*Fig. 5- Front wall detection based on front ultrasonic distance measurement*

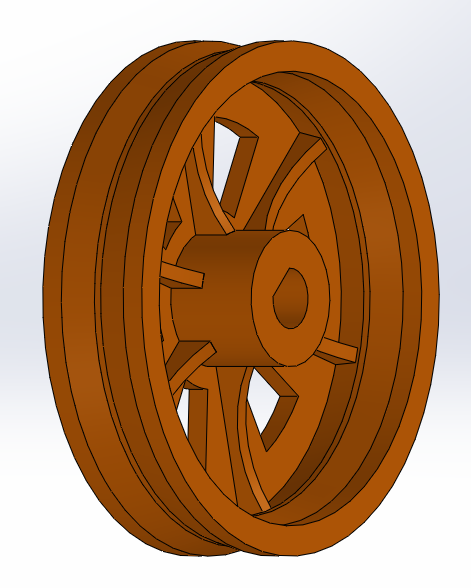


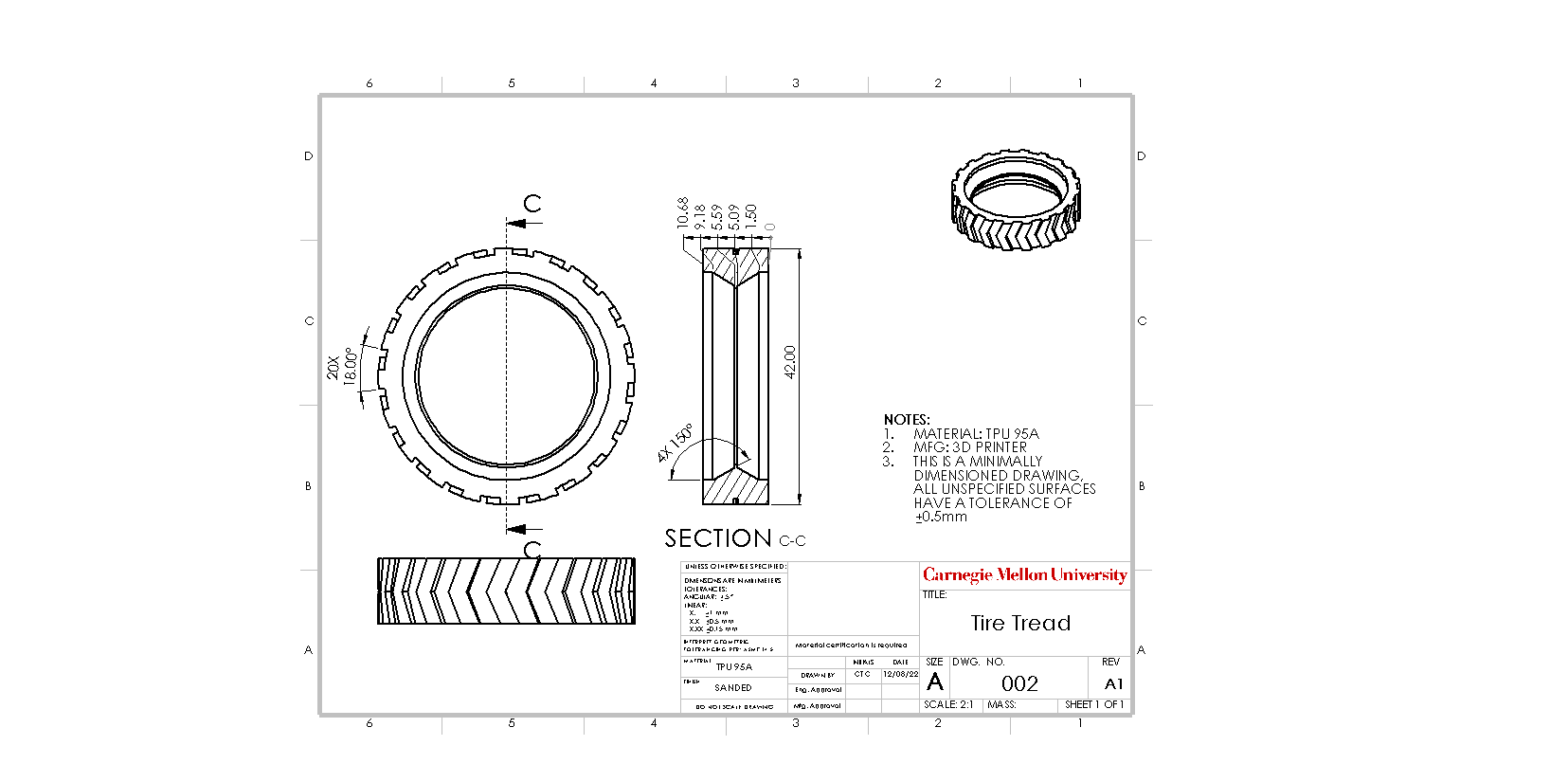
*Fig. 6- Side wall detection based on the difference between the left and right ultrasonic sensor readings*

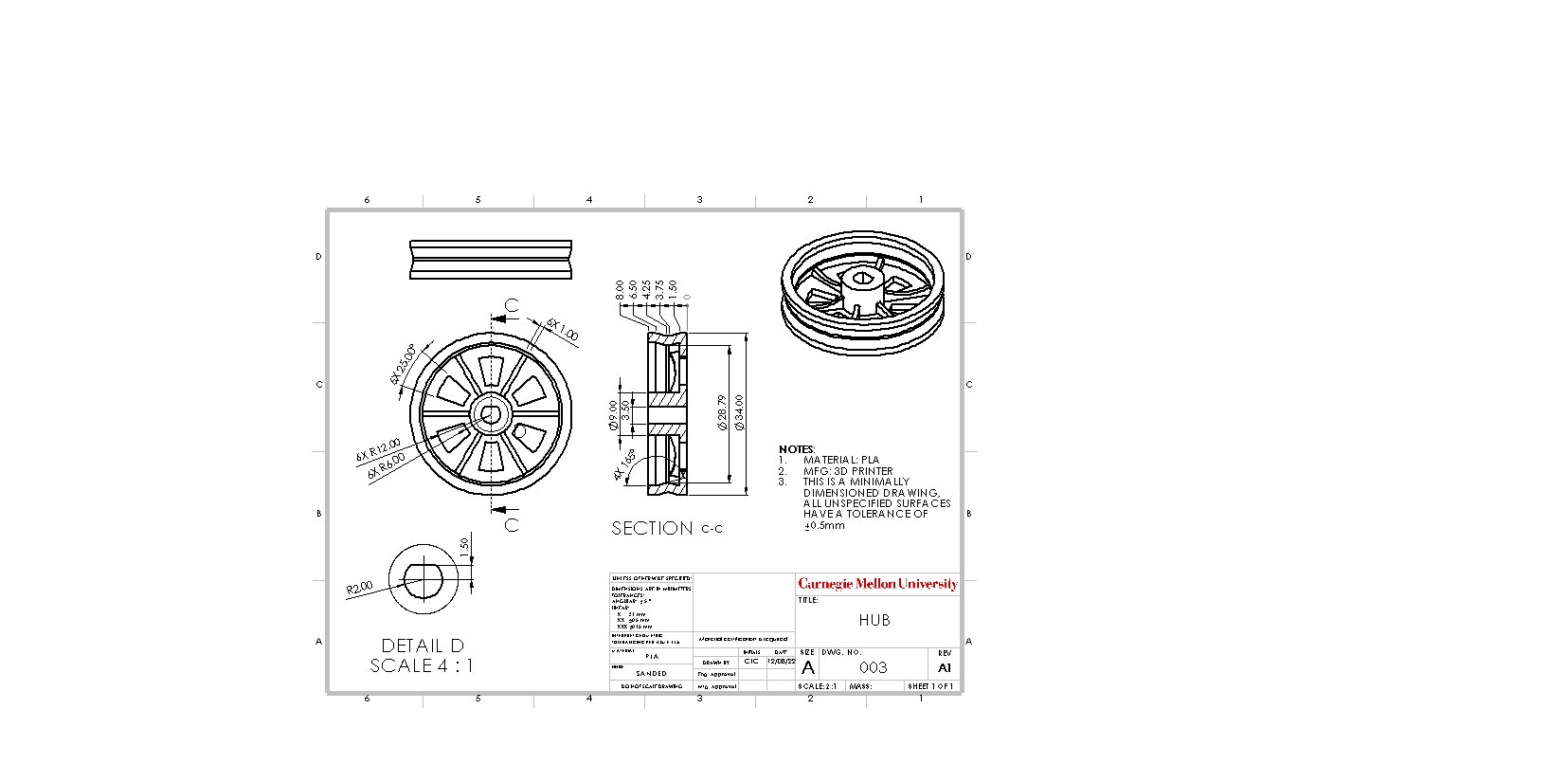
**Table 3- Ultrasonic uncertainty measurements from sensor platform project**

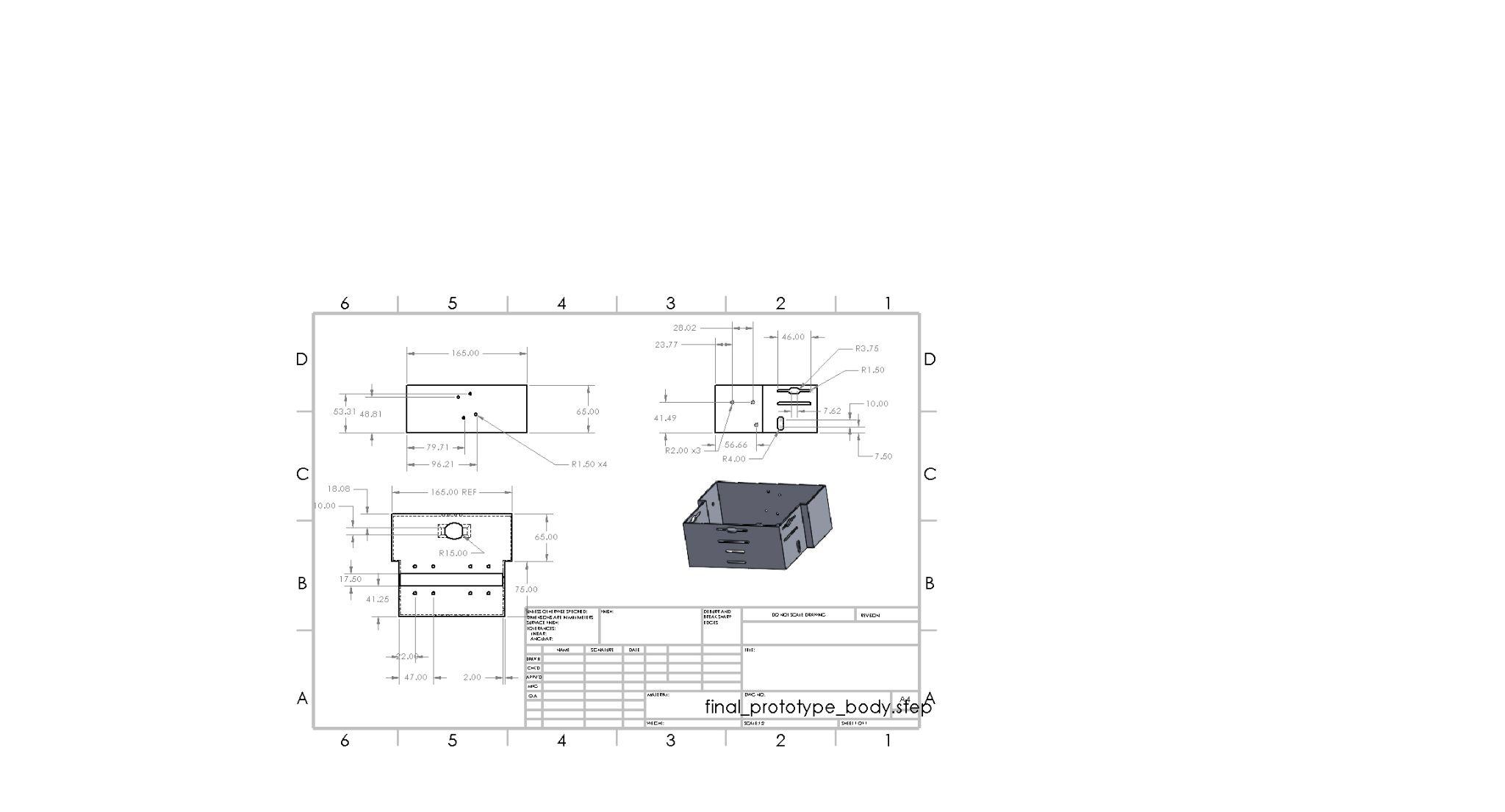
| Trial # | Video link | Log Name  [link](https://drive.google.com/file/d/1RCbpeX414qxB_zVlNvcn6o6i0F3mvK-C/view?usp=sharing) | 15cm reading | 10cm reading | 5cm reading |
| --- | --- | --- | --- | --- | --- |
| 1 | [link](https://drive.google.com/file/d/1Vqt0nxiTIejLFRXh8dSLPNwX9EdEBSqw/view?usp=sharing) | echo\_distance1.txt | 15.17 | 9.70 | 4.92 |
| 2 | [link](https://drive.google.com/file/d/178Trqtln77ksgYPNPstnyHG7aubQm7yh/view?usp=sharing) | echo\_distance2.txt | 14.82 | 10.00 | 4.78 |
| 3 | [link](https://drive.google.com/file/d/1GdOMGMkZGpbNOrcRe3vCf0H5UIfE0HTS/view?usp=sharing) | echo\_distance3.txt | 14.89 | 9.83 | 5.05 |

**Mechanical Part Pictures and Drawings**

****

****

****

****