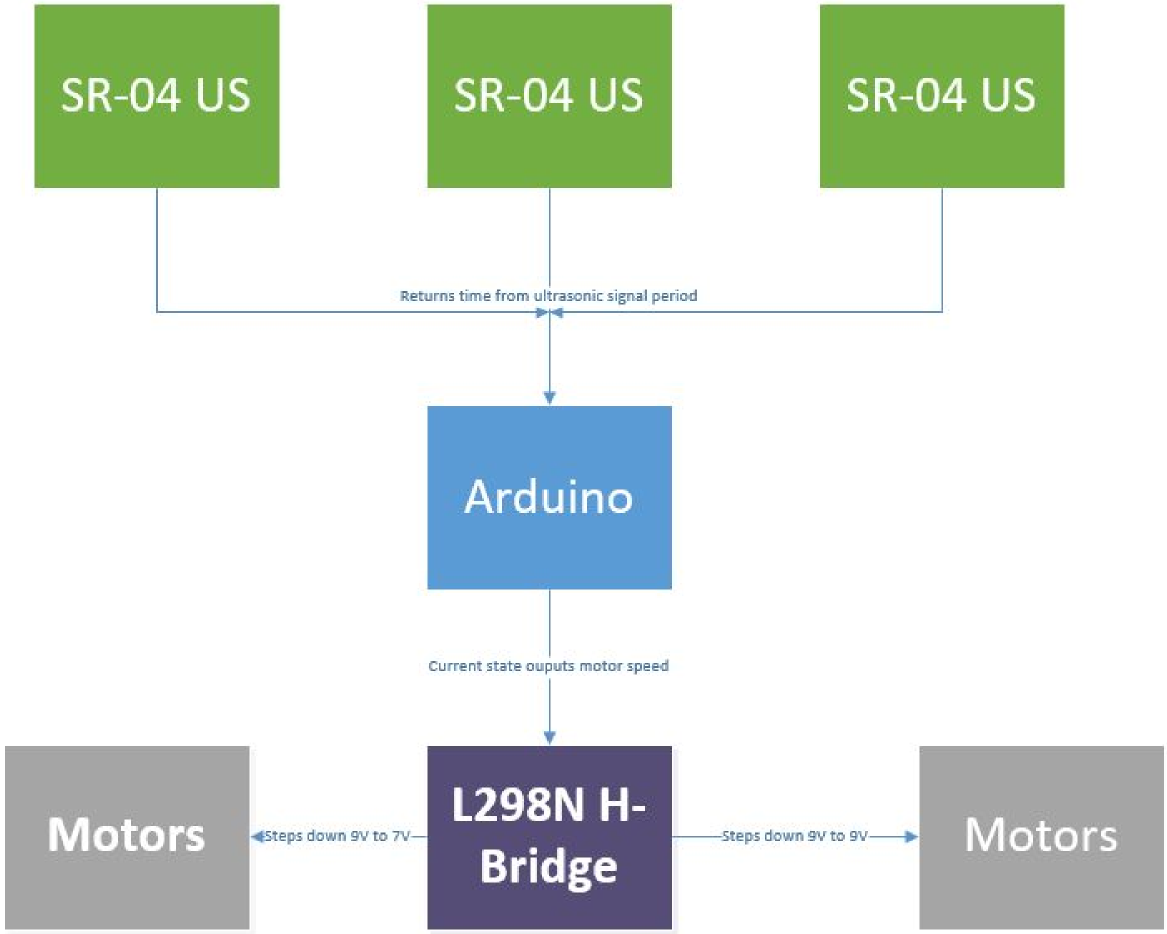
Ecocar Development Challenge Report - Team Robot

System Engineering Report

Accident and Hazard Identification

There are many potential hazards and accidents that the design may encounter. In any machine with an electrical system, there is a potential for any common current or voltage hazards to arise. In the design, a breadboard was used instead of a PCB, which makes use of many wires. Since some components, such as the V_{IN} and V_{OUT} pins of the H-bridge, are in close proximity, there exists the potential for soldering errors to arise. In the case where the soldering overlaps, a short circuit is created which causes a fire hazard. With respect to current as well, there is the potential for certain components (ie. the LED) to draw too much current (ie. from a lack of using a proper resistor) and overheat. Also, in the design a 9V battery was used to power the electrical circuit, and a 6V motor to power the wheels. The H-Bridge is used to step down the voltage to a proper value for the motors. In the case where the H-Bridge malfunctions, the voltage delivered to the motor will be too high and will result in overheating.

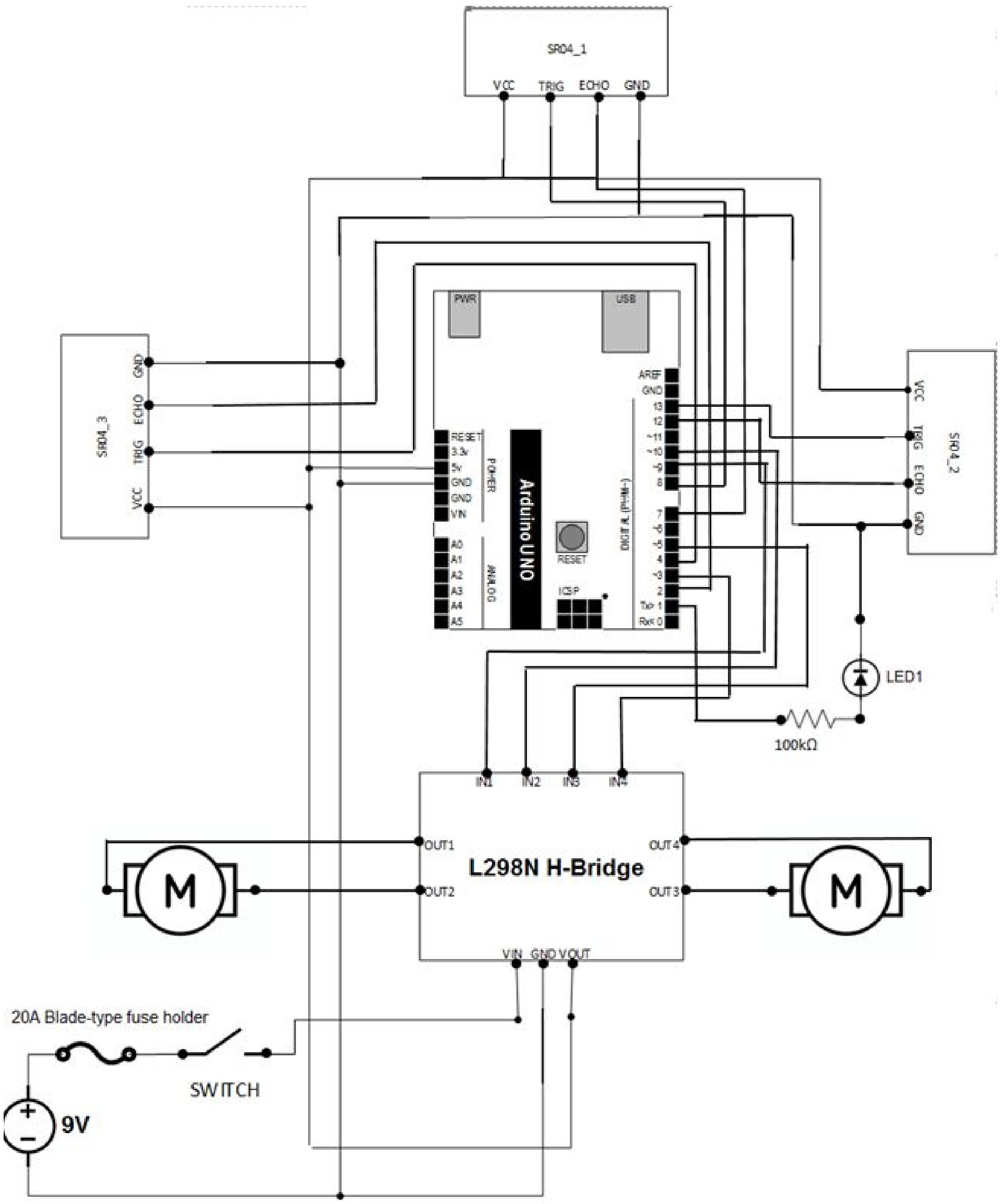
Functional Control System Block Diagram



System Safety Concept

In having numerous potential risks of hazards with this design, preventive solutions must be implemented in order to ensure safety. Evidently, current requirements are always to be kept in consideration. Although the probability of running a high current through an arduino is very unlikely, a razor blade fuse has been implemented in the circuit to prevent the possibility of overcurrent. Similarly, to prevent overcurrent in the LED, a resistance of 100k ohms was placed in series. Equally, another preventive measure to avoid overcurrent, specifically in the case of a short circuit, was to use insulated wires and minimize the number of exposed wires.

Schematic



All wiring used has a wire gauge of 22 AWG

Bill of Materials

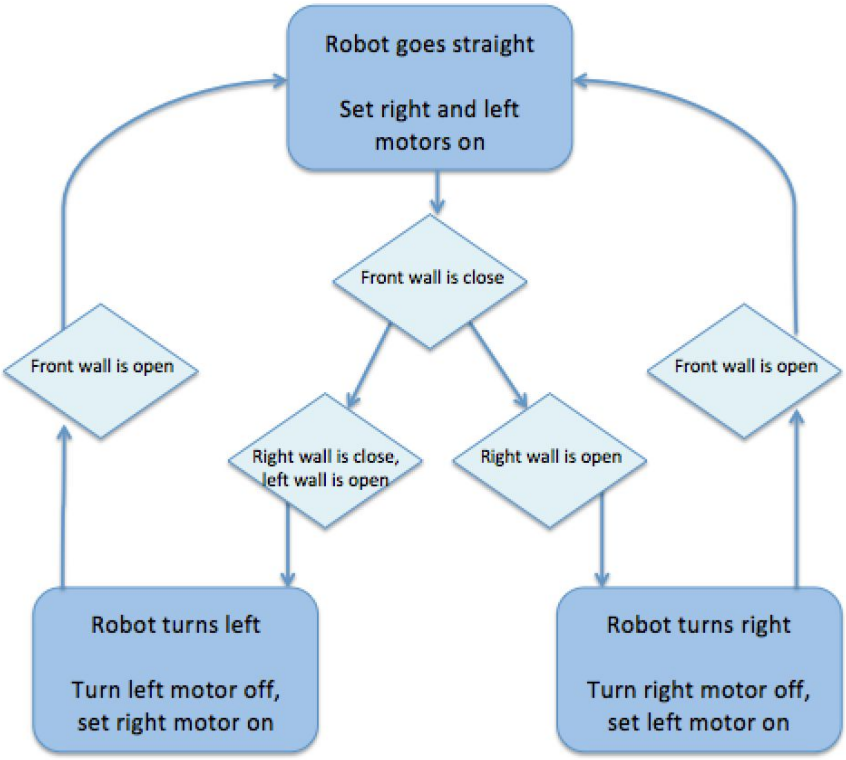
QTY	Item	Price/Unit	Price
1	Fuse Holder	0.64	0.64
2	SR-04 US sensor	3.5	7
1	Switch	2.5	2.5
1	Motor driver	10.09	10.09
1	Front wheel	4.7	4.7
2	Back wheel	1.52	3.04
1	9v Battery	6.76	6.76
Total			**$34.73**

Reflection

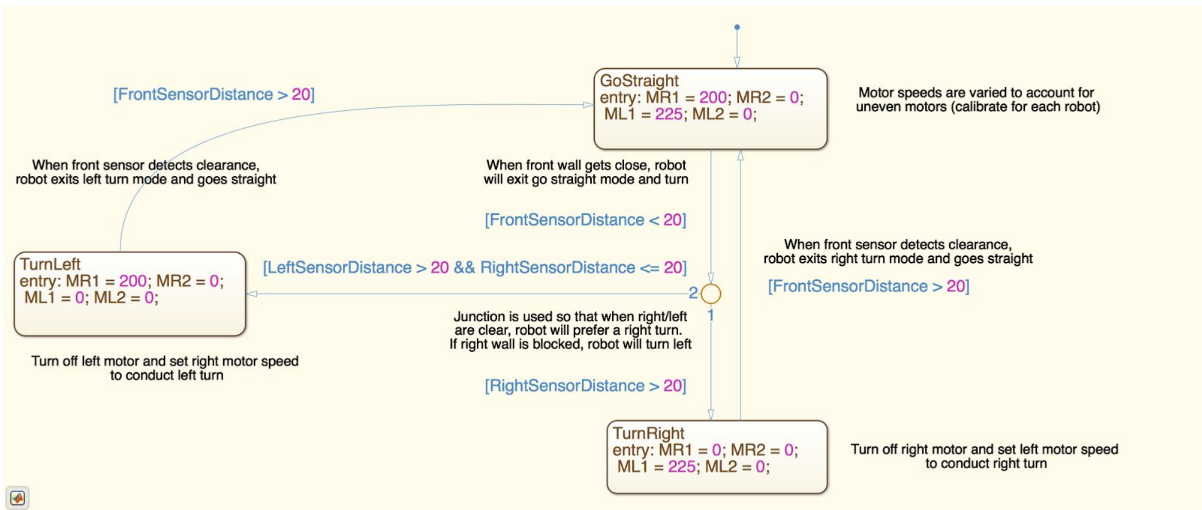
The circuit topology has had many iterations over the course of the development challenge. Primarily, it was assumed that a dc motor can simply be wired to the Arduino without any additional components. However, due to current and voltage restrictions, the motors require more than 5V and the 4mA that the Arduino can supply. As a result, the L298N H-Bridge was introduced to the circuit. Connecting a 9V battery to the H-Bridge eliminates the aforementioned restrictions with the Arduino acting as a speed controller through PWM and decreased duty cycles. Additionally, the voltage supply was initially vague as it was unsure as to how much voltage the entire circuit would consume. Ultimately, after consulting with EcoCAR mentors, 9V is more than sufficient voltage for this circuit. Furthermore, all components of the circuit are mounted using a breadboard as it allows for flexibility and versatility relative to a printed PCB or a perf board. Moreover, a blade fuse holder with through hole mounting is included in the circuit in series with the voltage supply. Specifically, this fuse holder was chosen because of its ability to be easily mounted onto the breadboard. Lastly, the glue to the entire circuit is the wiring. In terms of specification, a mix of jumper wire and hook up wire having a wire gauge of 22 AWG depending on the need for male-to-male or female-to-male connection. This specific wire gauge was used simply because it fits tightly into the breadboard and Arduino pin holes.

Controls System Report

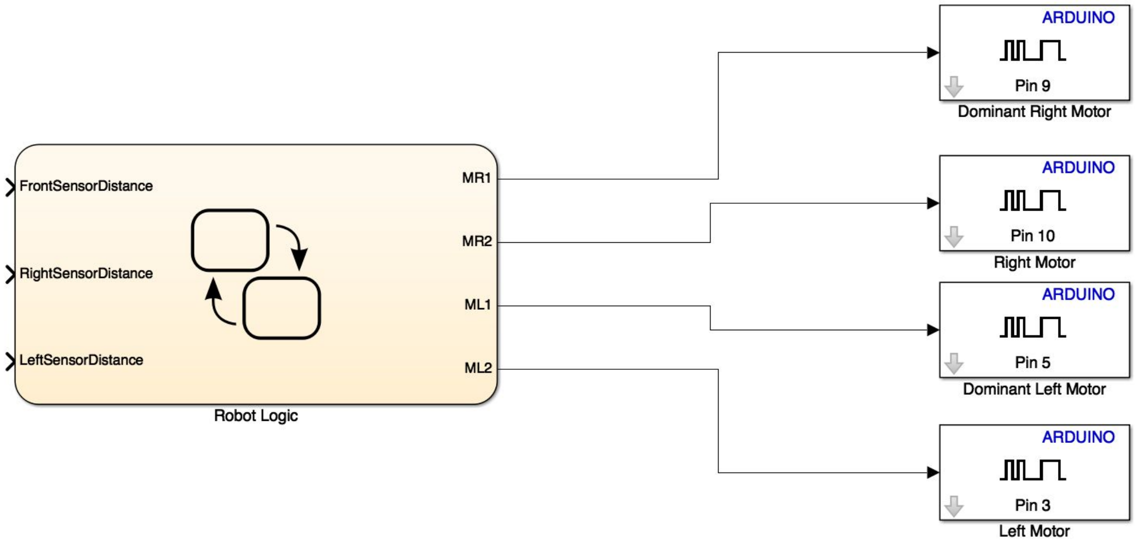
Flowchart of Logic



Simulink Stateflow Diagram



Simulink Model



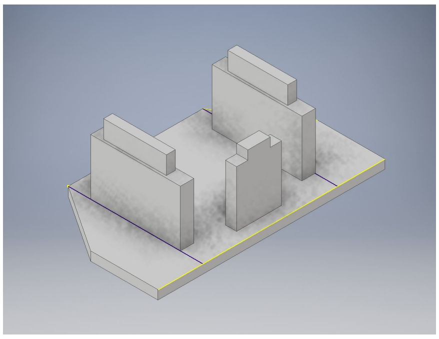
Reflection

When first implementing the logic through Simulink's stateflow, several test cases were used to test the basic logic. Three constant blocks were used to simulate sensor data. A value of 8cm was used to simulate a blocked path, and a value of 20cm was used to simulate an open path. An example of a test case was setting the right and front sensor distance to 8cm, and the left sensor to 20cm. The test case was successful, as the logic set the left motor speed to 0 and the right motor speed to 200, which effectively makes a left turn. Several similar test cases were simulated and confirmed the basic logic of the stateflow. In order to run more effective test cases, the code was uploaded onto the arduino to collect actual distance data in the maze. The test cases conducted were similar to the above test cases - blocking the robot from the right, or from the right and front, etc. Several logic holes were found - for example, the initial condition to do a left turn involved a condition for the right wall to be within 10cm of the right sensor. However, the robot would not turn left during times it was blocked front and right, only because the right wall was further than 10cm. Also, when the wall distance was greater than the maximum 200cm set by the S-function, the sensor would output a 0cm distance. This was accounted for during the testing period by placing wooden blocks to decrease the distance from the far wall. The most difficult part of the challenge was the inability to readily test simulink code due to software issues and only being able to test on a robot at MARC. In the future, it would be beneficial for the Ecocar team to run a quick set-up workshop where the controls lead could facilitate the setup of Matlab, Simulink and the C-Compiler so that everyone could start off at the same level and address compatibility issues immediately. Furthermore, it would be beneficial to have separate deadlines for mechanical, electrical and controls. If the deadline to complete the mechanical and electrical part of the robot was a week before the controls deadline, it would allow the controls member time to conduct effective test cases on the robot.

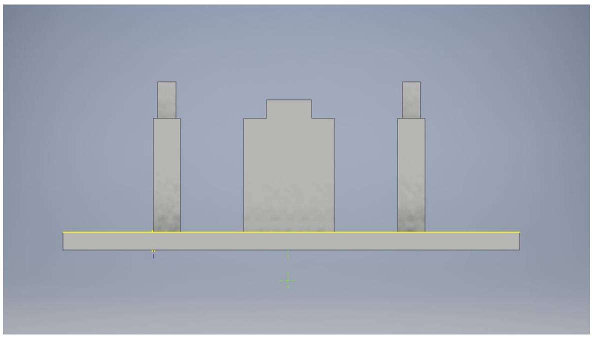
Mechanical Report

CAD 3D Views

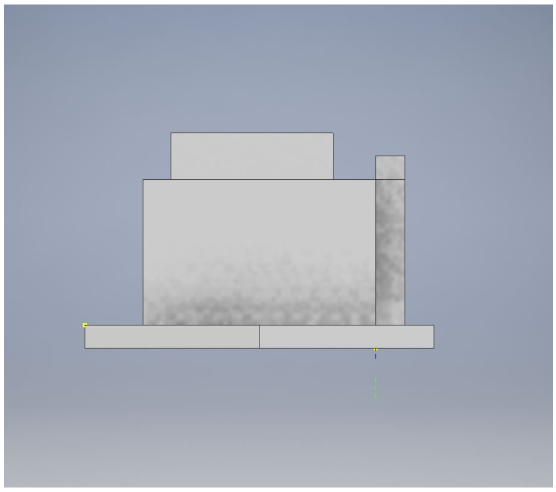
Main Base:



Front

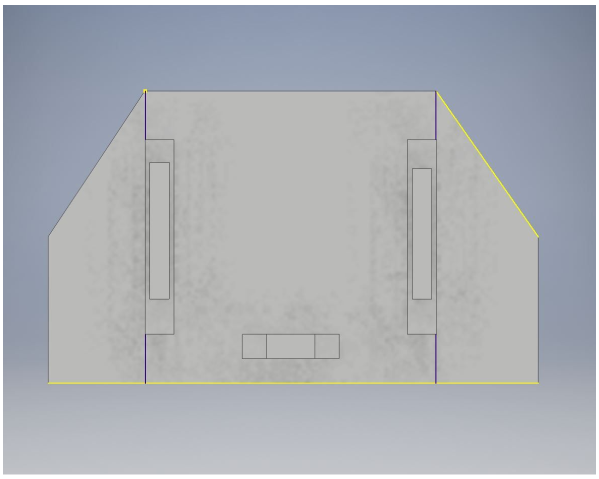


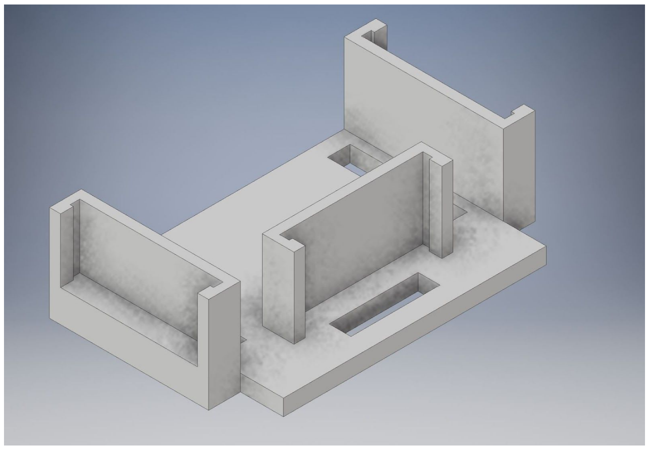
Side View



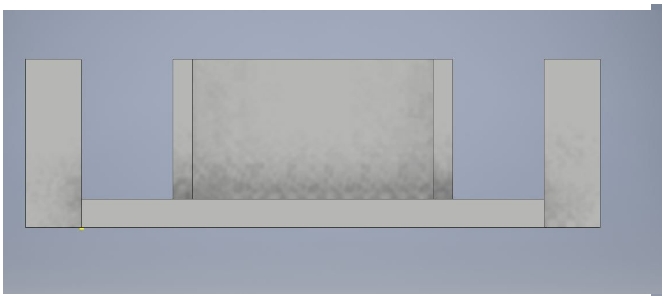
Top View

Top Deck:

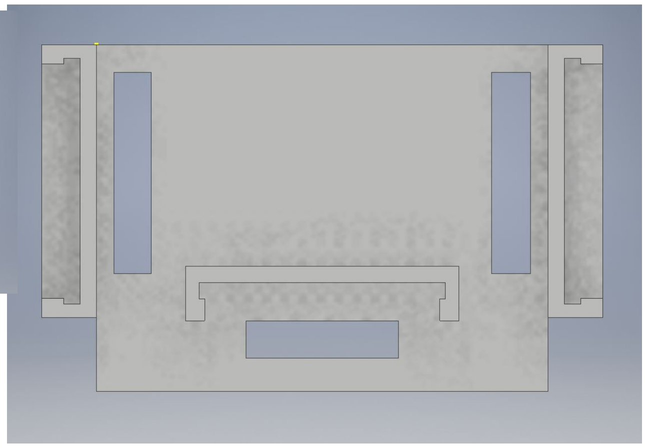




Front View

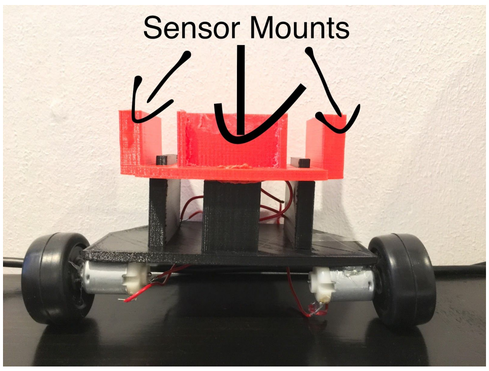


Top View



Side View:

Assembled View



Reflection

The goal of the 3d printed chassis design was to provide a chassis that could house all of the required electronics, and materials needed without using glue or screws. The approach chosen to achieve this goal was to design two separate chassis pieces that were designed to fit together tightly without the need for glue. The first piece was the main base, where the motors and wheels would be attached to. A separate motor mount was created that would attach to the bottom of this base. The center of the base was designed and measured so a circuit board could sit tightly in the middle; The the walls holding up the top deck acted as the support points. The top deck was designed to fit perfectly on top of the main base. The main base was shaped so the top base clicked tightly on top without the need for screws or glue. On the top base mounts for the 3 sensors were designed so the sensors could be securely attached to the chassis. These sensor mounts were designed specifically so that the sensors could be slide in and out quickly. But would remain secure when in motion. The middle section of the top deck was measured to tightly fit a circuit board between the sensor mounts, all of this without the need for glue or screws. In addition to being designed to fit tightly together without glue, it was designed specifically for 3D printing. The gaps for the fitted components were designed specifically with the 1mm tolerance of the 3D printer taken into account. The shapes used for the chassis were chosen because they could easily be 3D printed without flaw. This is the design process chosen for this chassis.