MIE444 - Mechatronics Principles
Request for Proposal

Team 11
Andres Cervera Rozo (1005469217)
Liam Toner (1005178754)
Christopher Tong (1005661874)
Nathalie Cristofaro (1004838009)

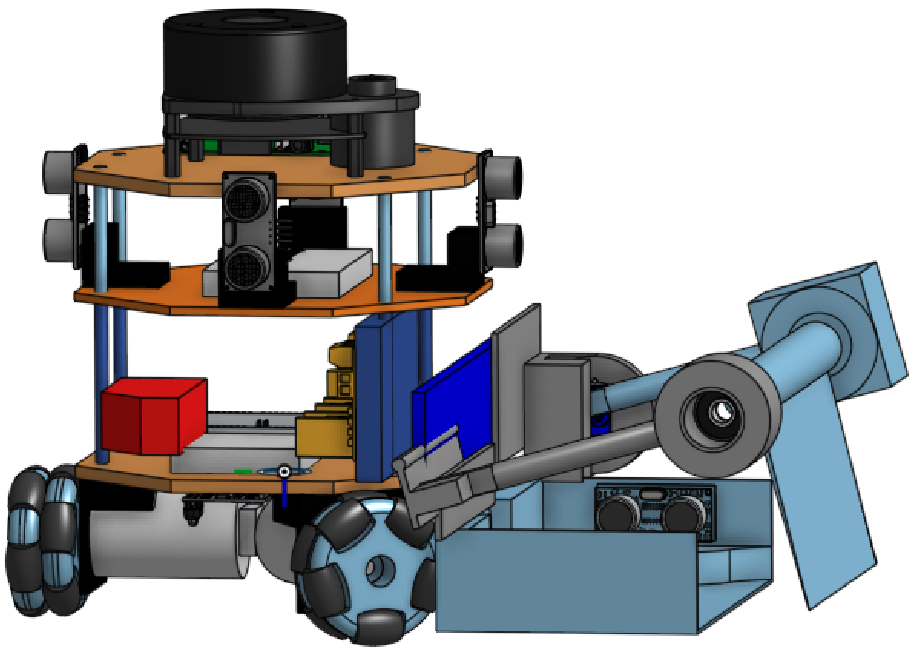


Table of Contents

1.0 Project Requirements

The team has been tasked with designing and prototyping a rover capable of autonomously avoiding obstacles, localization, and object relocation. The following sections state the tasks, functional requirements, and the design constraints for the rover.

1.1 Tasks

The rover should:

- Drive autonomously for at least 20’ in the maze without touching obstacles
- Determine its location in the maze when placed in a random location at a random angle
- Deliver a small load (less than 0.5lbs, 2”x2”x2” in size, beige in color) to a final destination in 5 minutes or less

1.2 Functional Requirements

To perform the expected tasks, the rover should:

- Avoid collisions with walls and obstacles
- Use a localization method to find the loading and delivery locations
- Have a visual method of displaying live rover localization throughout the maze
- Display a signal when it has arrived at the loading zone, detected the block, and completed the delivery
- Use a retrieval mechanism to pick up and store the block above ground

1.3 Design Constraints

The rover must:

- Weigh less than 5lbs
- Not exceed a 12”x12”x12” size
- Not use pre-made kits/equipments
- Be powered by battery packs on board
- Incorporate 3D components
- Not use adhesives or velcro to pick up the block
- Not use touch sensors
- Display the rover localization visually

The following list states the limitations and next objectives for the rover design introduced in Section 2.

Limitations:

- Block retrieval mechanism can be further improved, and will undergo a second design iteration
- Might add second arduino to improve debugging
- For the next iteration, the retrieval mechanism will be designed to better integrate with the robot and its overall footprint will be reduced

2.0 Rover Design

This section will describe the proposed rover chassis and block retrieval designs, along with the final electrical connections layout for the various components that will be mounted on the rover.

2.1 Mechanical Structure

Figure 1 shows the Onshape CAD model of the final proposed design. See appendix A for a drawing with all major dimensions.

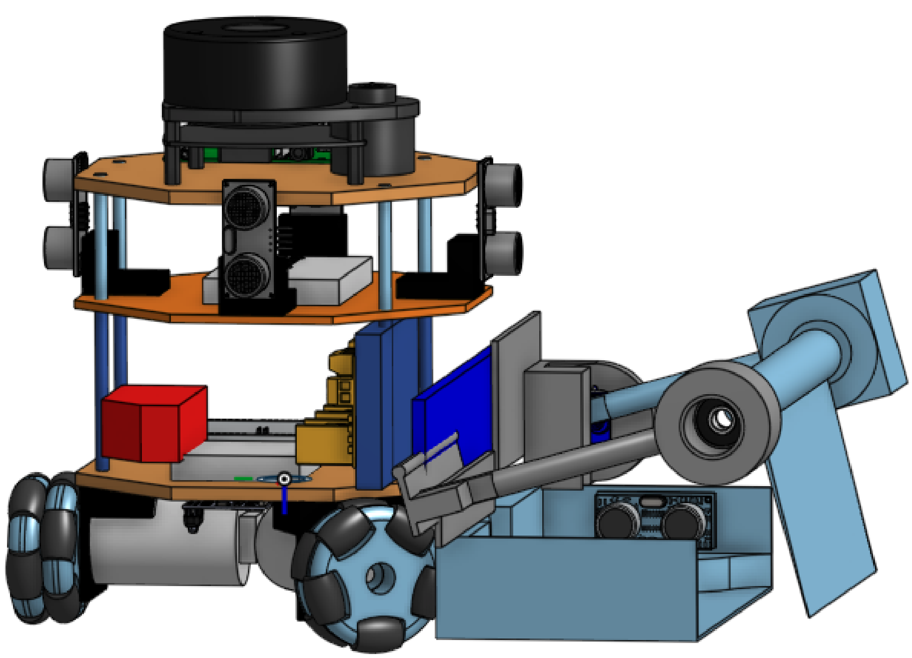


Figure 1: Onshape CAD of First Robot Iteration.

- The mechanical loads that the robot will endure are low, thus the main purpose of the structure is to house the electronics.
- Height of the robot is 9in (23cm) so that the LIDAR is still able to sense all the walls. To leave a buffer of 5.5in between the robot framing and maze walls, the diameter of the robot was made to be 6.5in.
- Based on the heaviest components of the robot which are the motors (~100g each), the 3 layers of ⅛" plywood, and all the electrical hardware, it is estimated that the robot will be 4lbs.
- The robot is divided into 3 layers, which will be elaborated on in Section 2.1.3.
- The LIDAR is on the top layer to ensure that its field of view (FOV) is unobstructed by the robot. The motor controllers and motors are located on the bottom layer, as the wheels must touch the ground. The block retrieval is mounted on the front of the robot to facilitate driving towards the block to retrieve it.

The main design objectives for the mechanical structure were designing for ease of assembly/disassembly and debugging. In the case where the code and hardware must be altered during the design process, ensuring the robot is designed for disassembly will be important to allow for wiring validation and ease of uploading code.

2.1.1 Drive System

The design uses 3 omni wheels in a triangular configuration, mounted 120° apart in order to have the robot rotate in place and translate in each coordinate direction as seen in Figure 2. Omni wheels have rollers that allow for motion normal to the direction of the roller. This configuration and wheel type reduces the amount of turning required. This drive mechanism also allows the robot to perform 'tank turns' in place. Tank turns save time and reduce the likelihood of the robot colliding with walls. Each omni wheel is driven by a 12V DC gearhead motor with an encoder. The omni wheel is mounted to the motor shaft using the coupling in Figure 3. A DC motor was chosen because of its ability to rotate 360°. To have the robot move straight with a 3 wheel, triangular configuration, the team will make use of the equations of motion seen in Figure 4, where the U vector represents the desired wheel driving speeds.

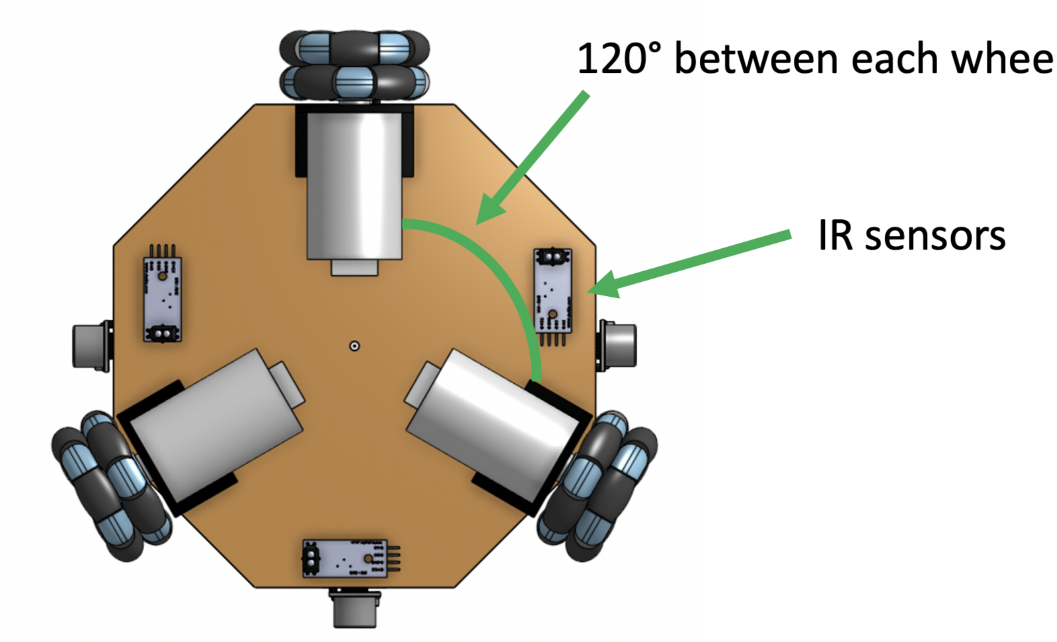


Figure 2: Robot Bottom View.

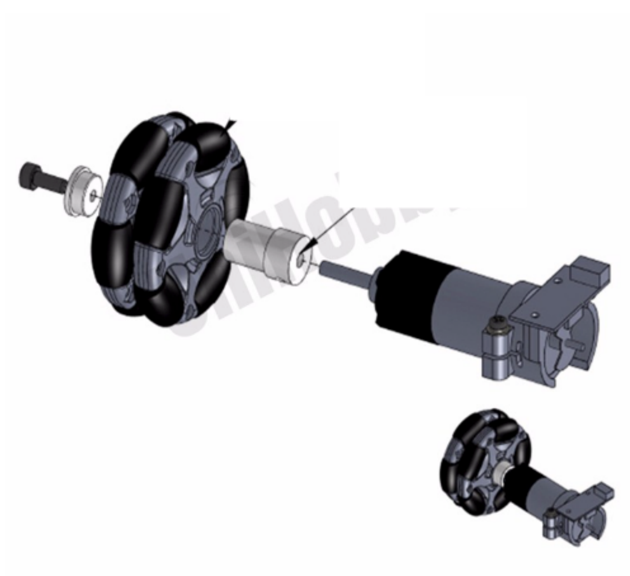


Figure 3: Coupling Mount for Omni Wheel to DC Motor.

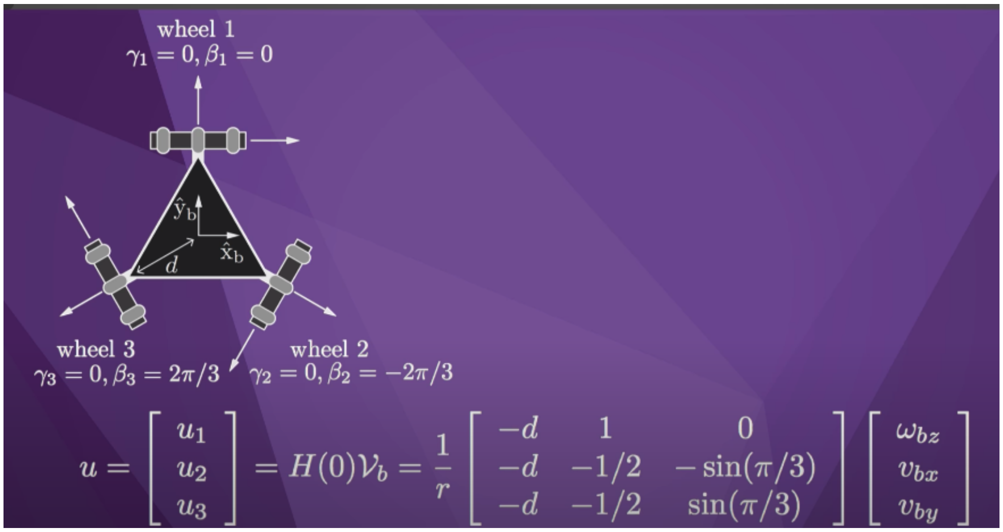


Figure 4: Equations of Motion to Control Three Wheels in a Triangular Configuration.

The wheel motors will be driven by the L298N dual H-bridge motor driver modules to control their speed and direction. Each L298N can drive two motors, so this design can support four DC motors total. With three used in the rover drivetrain, and the fourth slot is currently allocated to the block retrieval mechanism, which will be discussed further in section 2.3. To ensure the motors are moving the robot accurately, the encoder data from each motor will be captured for use with a PID control system which will be implemented on the arduino controller.

2.1.2 Ease of Assembly

For ease of assembly and disassembly, parts that will be frequently taken on and off the robot (i.e. the arduino and breadboard), will be fastened via standoffs. The arduino and pi are already designed with mounting holes, so they will be slid onto the male standoffs that are bolted to the plywood. Standoffs facilitate ease of assembly because screws are not required to hold the arduino in place. If a more permanent solution is required, nuts can be placed on the threaded side of the standoff so that the arduino cannot slide off. The nuts can be tightened by hand and do not require hand tools. The 3 layers of plywood can also be easily assembled using standoffs. All the sensors are mounted to the robot using screws and a 3D printed mount with a heat shrink threaded insert. Table 1 describes the attachment method and mechanical hardware required to mount the electronics on the robot.

Table 1: Electronics Attachment Method and Required Mechanical Hardware.

Hardware	Hole size	Fastener	Bolt holes	Total # of fasteners required	Bracket
Arduino Uno (2)	3.2mm/0.125in	Screw	4	8	Standoff
Raspberry pi 2B+ (1)	2.75mm	Screw	4	4	Standoff
Ultrasound sensor (4)	1mm	M1	2	8	Custom 3D print
VMA326 IR sensor TCRT5000 (3)	X	Screw	2	6	Custom 3D print
VL53L0X Time of flight sensor (1)	X	Screw	2	2	Custom 3D print
Lidar sensor (1)	X	Screw	4	4	Not required, bolted into plywood with screws
L298N Motor controller (2)	3mm	M3	4 (only need 2)	4	Custom 3D print
Servo Motor SG90 (1)	2mm	M2	2	4	Off-the-shelf bracket
DC motor (3)	X	Screw	X	X	Bracket sold in myhal
Custom Brackets	X	All custom brackets will use same fastener size	X	X	

2.1.2 Ease of Debugging

Figure 5 shows how the robot design (without the pickup mechanism showing) can be split up into 3 layers for ease of debugging. Some electronics such as the breadboard and battery, have been represented as dimensionally-accurate coloured blocks for ease of visualization.

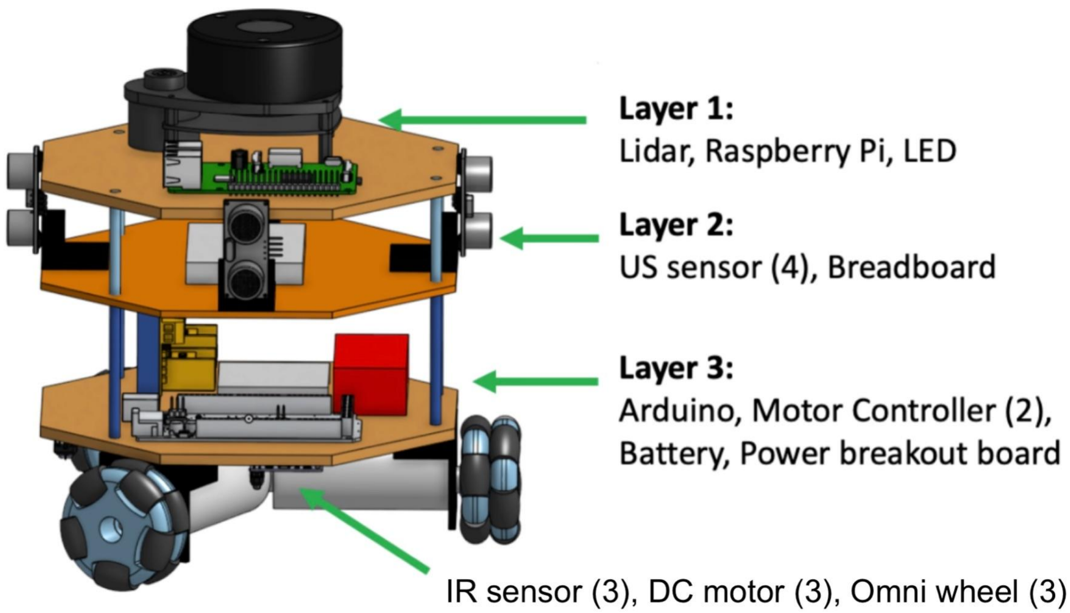


Figure 5: Onshape CAD Model Displaying Robot Layers.

The robot will contain three subsystems: LIDAR, ultrasounds, and motors. For ease of debugging, the robot can be split into 3 layers, sorted by subsystem, as seen in Figures 6-8. This layout will allow for neater wiring and will reduce the EMI interference between wires. Moreover, the robot will be easier to debug since layers can be detached in such a way where one can focus on a single subsystem. This layout also allows for teammates to work on the robot in parallel, since each subsystem can be removed and operated individually. The team is exploring the viability of adding a second Arduino to control the electronics located on layer 3, while having the first control the electronics located on layer 2, which will allow for complete layer separation.

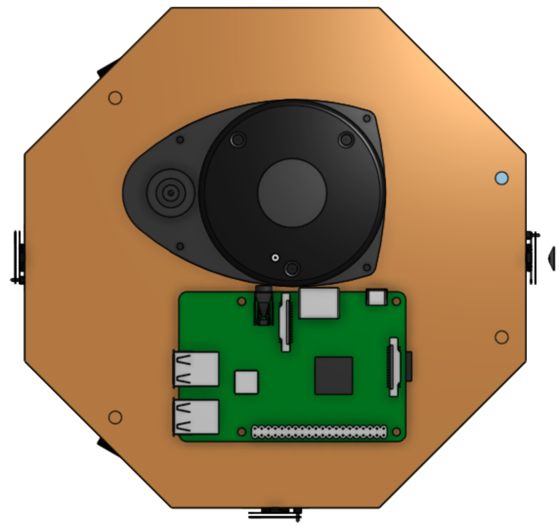


Figure 6: Layer 1 for Testing LIDAR.

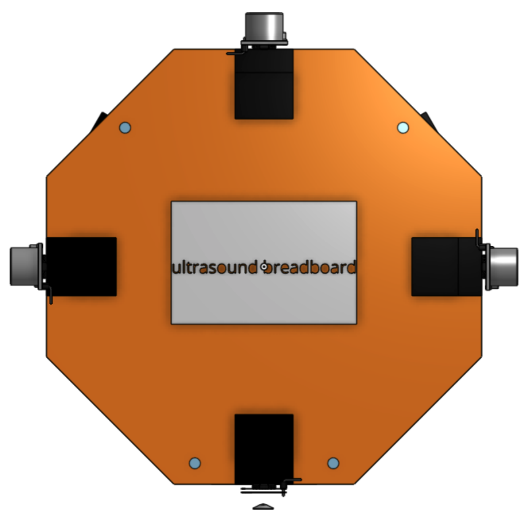


Figure 7: Layer 2 for Testing Ultrasound Sensors.

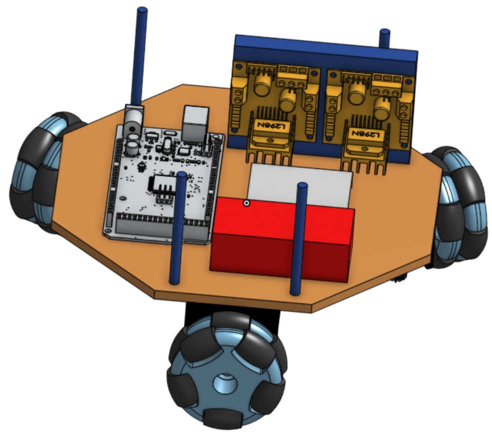


Figure 8: Layer 3 for Testing Motor Control.

2.1.4 Material Selection

The following bullet points state the justification for selecting specific materials for the design chassis and mounts:

- The 3 octagonal layers will be laser cut from ⅛" plywood due to its low cost, sufficient stiffness, and workability.
- All the sensor and motor mounts will be 3D printed from PLA.
- The "pillars" in Figure 8 above, are metal standoffs. Metal was chosen due to its stiffness and strength.

2.2 Electronics

The preliminary electrical layout is outlined in the wiring diagram below (Figure 9). Figure 9 defines the electrical connections of the different parts of the rover without considering where they will be physically located relative to one another. The diagram includes the Arduino and Raspberry pi for control, various sensors (LiDAR, ultrasonic, IR, and TOF) for localization, actuators (DC motors, servo), and their drivers where necessary, as well as a power management system to produce the different voltage levels required throughout the design.

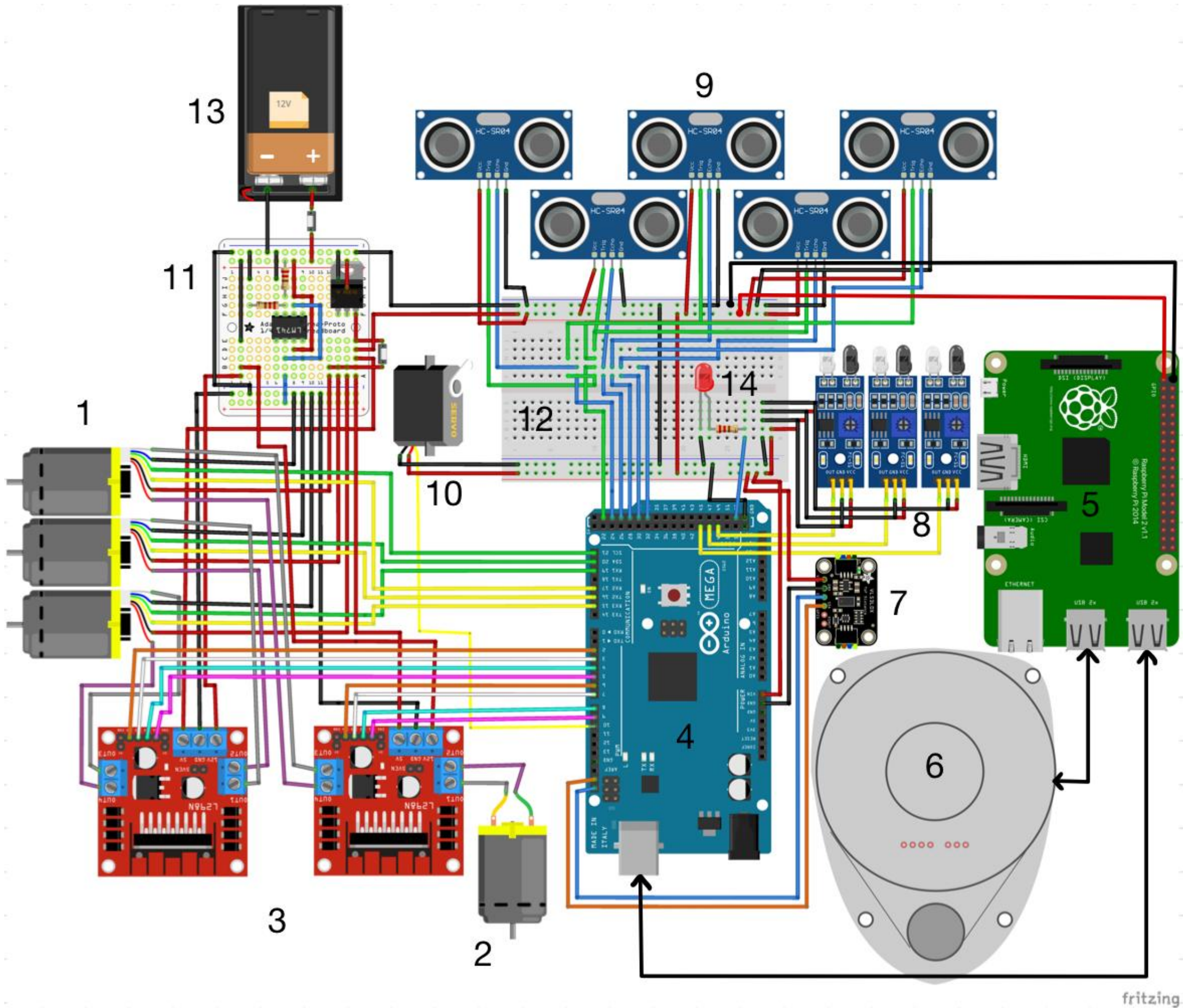


Figure 9: Electrical Wiring Diagram.

1. 3 DC motors with encoders
2. 1 DC motor for retrieval mechanism
3. 2 motor drivers
4. Arduino Mega
5. Raspberry Pi 2B
6. LiDAR sensor
7. ToF (Time of Flight) sensor
8. 3 IR sensors
9. 4 ultrasonic sensors
10. Servo for block retrieval mechanism
11. Power breakout board
12. Breadboard
13. 12V NiMH battery pack
14. Loading zone indicator LED

Due to the limitations of the electrical layout software, a few details are missing from the electrical layout. They are as follows:

- The Raspberry Pi will power the arduino using their respective USB ports (indicated by the black arrow).

- The LiDAR sensor is powered and controlled by a USB cable between it and the Raspberry Pi (indicated by the black arrow).

2.2.1 Power Management System

This robot requires multiple voltage levels to function properly; specifically, the motor drivers require 12V and a number of other components require 5V. Item 11 on the wiring diagram above is the custom board, which will separate different required voltages and distribute them where they are needed. Additionally, this board will feature switches to allow toggling of global robot power, and toggling power to the wheels without turning off the rest of the system to allow for easier sensor troubleshooting.

2.2.2 Sensor Quantities and Locations

The following list states the sensor locations and quantities used the in final rover design:

- 5 ultrasonic sensors: one on the front, back, left, and right side of the robot, plus an additional sensor to confirm the block is onboard.
- 3 IR sensors: three sensors will point down on the lowest layer to monitor which square robot is located on. Three sensors will provide the robot with more feedback, however the team might end up using only two for simplicity.
- 1 ToF sensor: located in the retrieval box of the rover retrieval mechanism to detect the block once the rover is in the loading zone.

2.3 Block Retrieval Mechanism

A bi-directional sweeping design was chosen as the block retrieval mechanism. The design was chosen due to its simple design and programming feasibility.

Figure 10 depicts a labeled Onshape capture of the sweeping design for this robot. Most parts will be 3D printed, however the flap will be made of a row of straws on which a rubbery strip will be glued to grip the block during loading and unloading. See Appendix B for a dimensioned drawing of the retrieval mechanism.

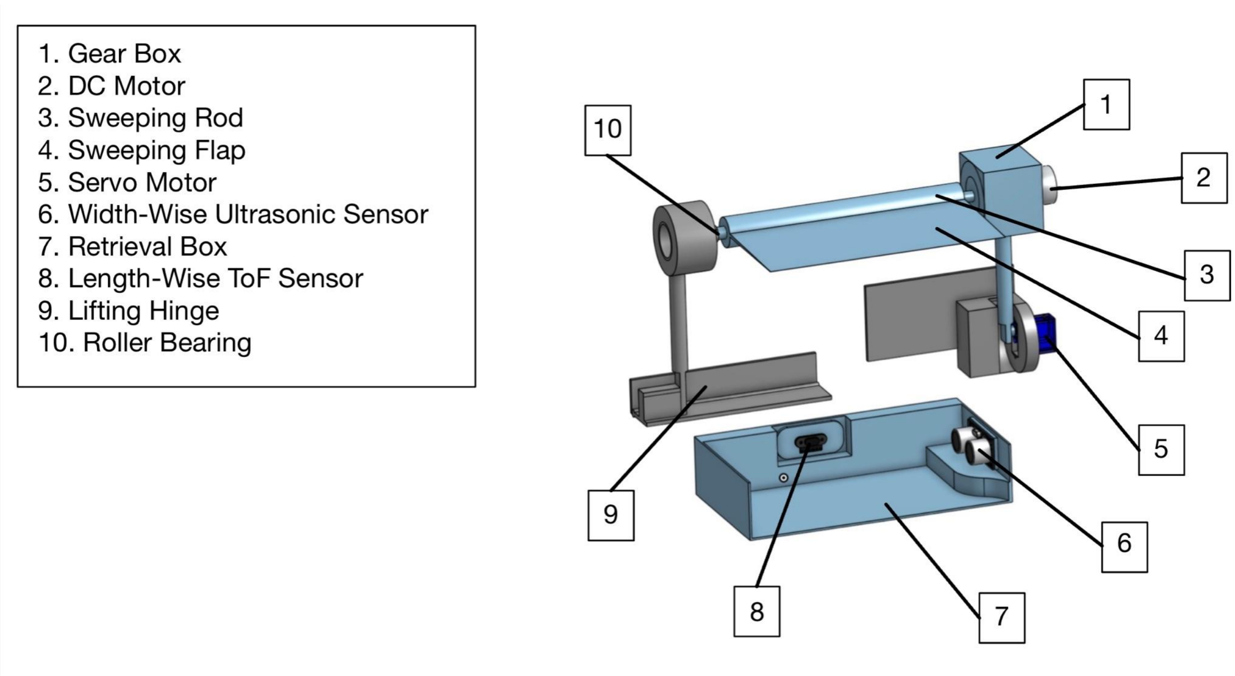


Figure 10: Block Retrieval Onshape Assembly.

The sweeping flap located around the circumference of the sweeping rod is 3 inches long; chosen to be an extra inch longer than the block to ensure there are no gaps between the floor and flap where the block could slide under. The total length of the sweeping rod is 6 inches, which is three times the block length, creating a large intake area.

The motor shaft will slide into a slot on one end of a gear box connected to the sweeping rod. The gear box will decrease the speed of the DC motor so the mechanism can sweep the block into the box at a controlled speed. A DC motor has been selected for the sweeping motor, because it is simple to control, can rotate fully clockwise and counterclockwise, and is inexpensive. The block sweeping mechanism does not require precise increments in speed and requires a full-circle rotation, justifying why a stepper motor and servo motors were not selected, respectively. On the other end of the sweeping rod is a roller bearing to allow for the sweeping rotation and mechanism lifting sequence.

The sweeping motor and bearing enclosures are connected to a lifting mechanism, which is actuated by a servo motor. The mechanism is connected to the robot framing with hinges to allow the entire sweeping mechanism to be lifted upwards during travel to reduce the overall robot footprint. The motor selected to lift the sweeping mechanism is a servo motor, which can attain the higher torques and 180˚ rotation needed to lift the entire sweeping mechanism. This servo motor also slides into a shaft slot, and is enclosed for support.

The retrieval box is located directly underneath the sweeping frame, and its intake area is 6 inches long to match the length of the flaps. The box is 4 inches wide and about 2 inches tall to match the height of the block. One time of flight sensor (ToF) is located along the length of the retrieval box, and will be used to detect the block in the loading area. The sweeping mechanism will be lifted during block localization, and will lower once the block has been detected.

On the width of the box, there is an ultrasonic sensor which will be used to detect if the block is inside the box. The depth of the box is 3.5 inches, thus the 2 inch block will always be detected by the sensor if it is in the box. The sensor will attempt to detect any obstacles that are less than 6 inches away from it, which will determine whether the block is inside the box. In the case where the block enters the box at a high speed, both sensors are protected using wall indentations. The wall along the box width is curved to guide the block safely into the intake area.

2.3.2 Design Programming

With regards to programming, a simple on/off command sent to the sweeping motor will suffice. The block intake procedure in the loading zone is as follows:

1. The block is detected by the time of flight sensor located along the length of the retrieval box.
2. A command is sent to the servo motor to lower the sweeping mechanism.
3. A command is sent to the DC sweeping motor to begin block intake.
4. The block is detected by the ultrasonic sensor on the width of the box.
5. The command to stop the sweeping motors is sent.

When the robot arrives at the unloading zone, the mechanism will sweep in the reverse direction to drive out the block from the retrieval box. Only once the ultrasonic sensor along the width of the retrieval box no longer senses the block in the box will the reverse sweeping motion process stop, and will the sweeping mechanism return to an upright position.

3.0 Bill of Materials

Please find the Bill of Materials attached with the submission of this report. It should be noted that the cost of this robot slightly exceeds the $300 limit. The team has spoken to the professor and agreed that slightly exceeding the budget is acceptable given the added cost and complexity of LiDAR integration. All referenced STL files can be found in the Final Assembly zip folder

4.0 Maze Solving Strategy

There are 2 localization strategies the team is considering: a Simultaneous Localization and Mapping (SLAM) approach, and the more rudimentary ultrasound/IR histogram localization approach. The team is interested in learning more about implementing the industry-standard LiDAR, and thus will be focusing on the map building approach, but will also explore the histogram localization algorithm as a backup.

4.1 Primary Strategy

The Arduino and Ubuntu-based Raspberry Pi (connected through Serial) will process the robot's inputs and outputs. Using a WiFi dongle, this data will be transmitted to a nearby PC where the computationally expensive algorithms will be executed. This near-instant communication is facilitated through the Robotic Operating System (ROS) framework, comprising several nodes and topics as seen in Figure 11.

As the first step, the team will tele-operate the rover through the maze. Using a LiDAR to track the environment, an accurate map of the maze will be generated using the Hector SLAM package on ROS, which will be used as a reference map for future localization.

When driving autonomously, the rover will implement Adjusted Monte Carlo Localization (AMCL) to determine the highest probable location and orientation through a developing 2D particle distribution across the map. With this pose, the rover will use the move_base ROS package to plan and navigate to its designated waypoints on the maze based on a set of global and local cost maps. These maps will account for the rover's position relative to the map, as well as its proximity to nearby walls (from ultrasonic sensors), to output the desired movement direction as shown in Figure 12.

The live location of the rover will be demonstrated using the popular robot visualiser tool, Rviz, where the transform between the rover base_link and the reference map will be visualized in real-time as seen in Figure 13.

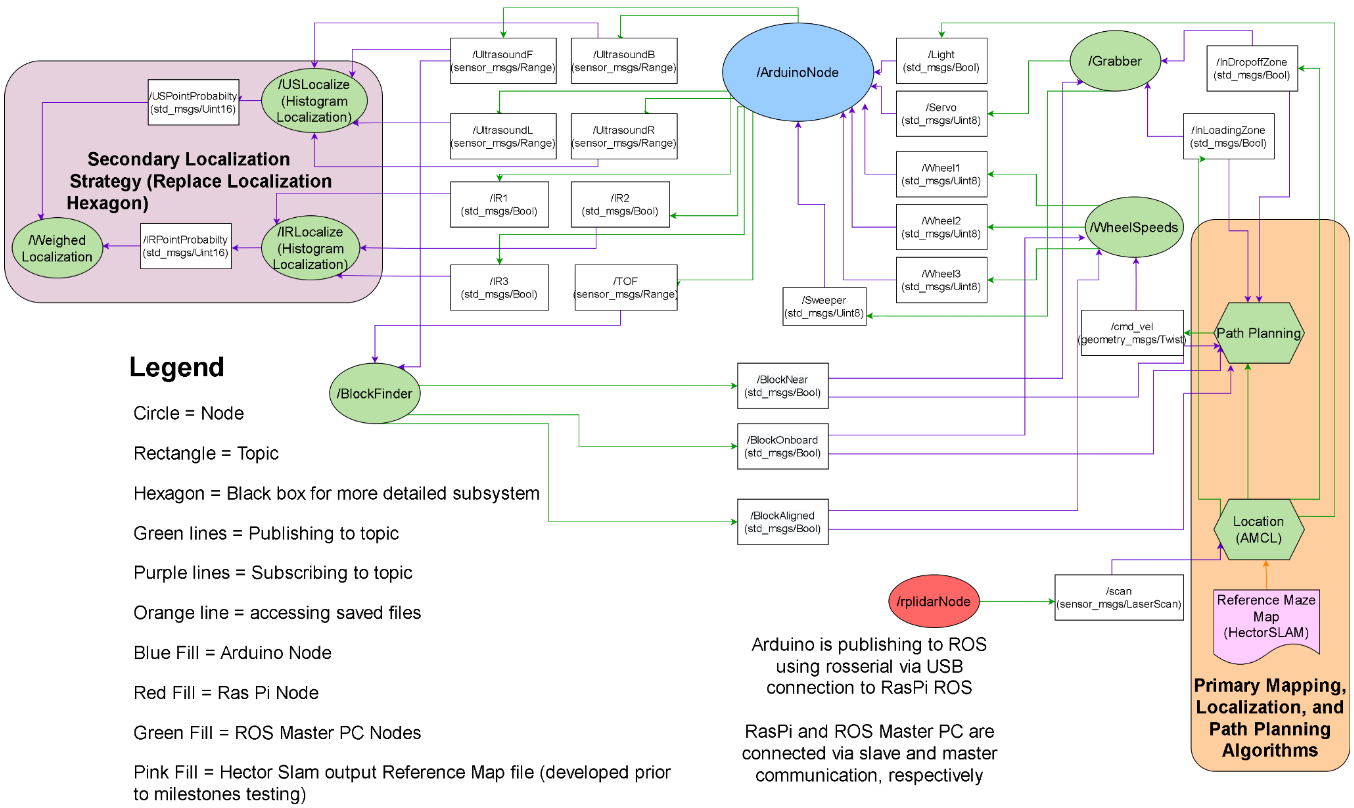


Figure 11: ROS Nodes and Topics Network.

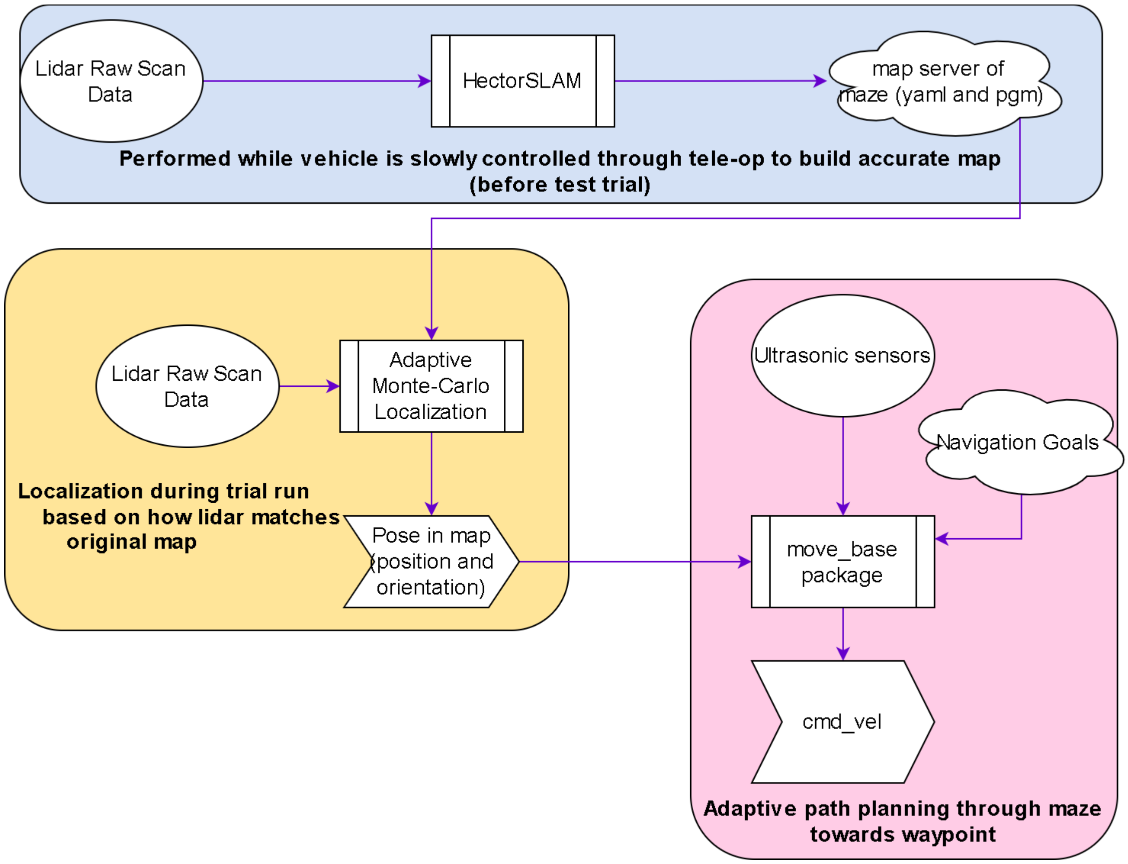


Figure 12: Detailed Primary Localization Algorithm.

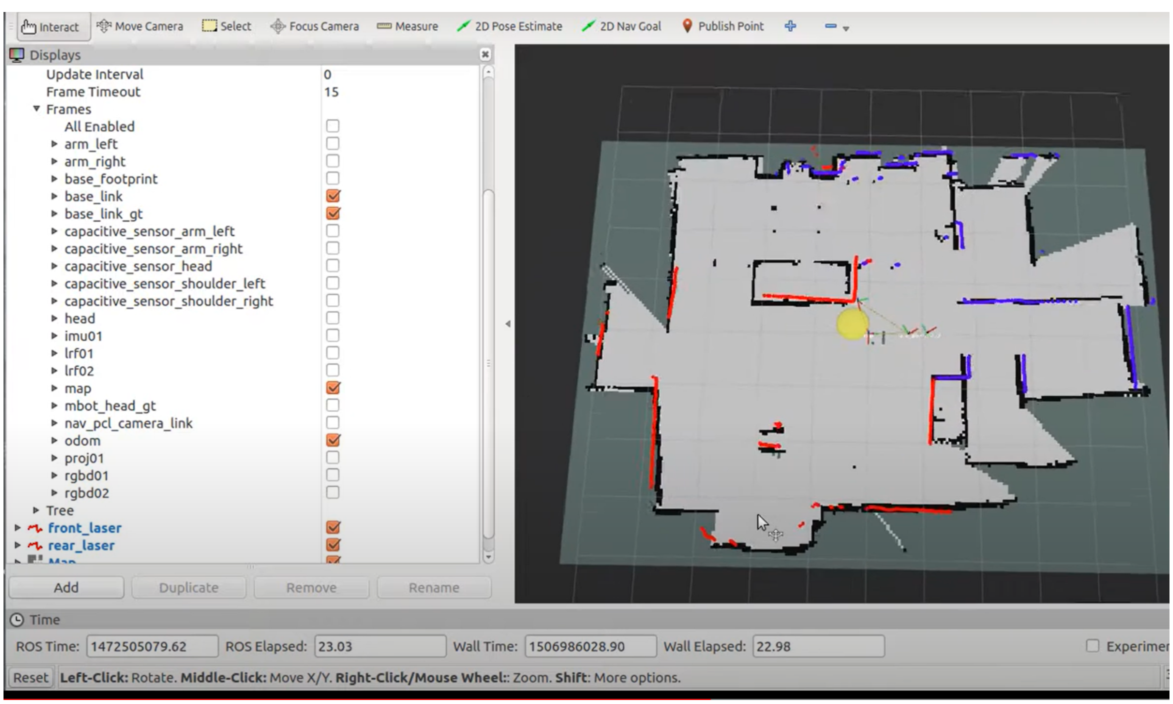


Figure 13: Sample AMCL Visualization On Rviz.

5.0 Contribution Table

Table 2 identifies each team member's individual contributions to this final report.

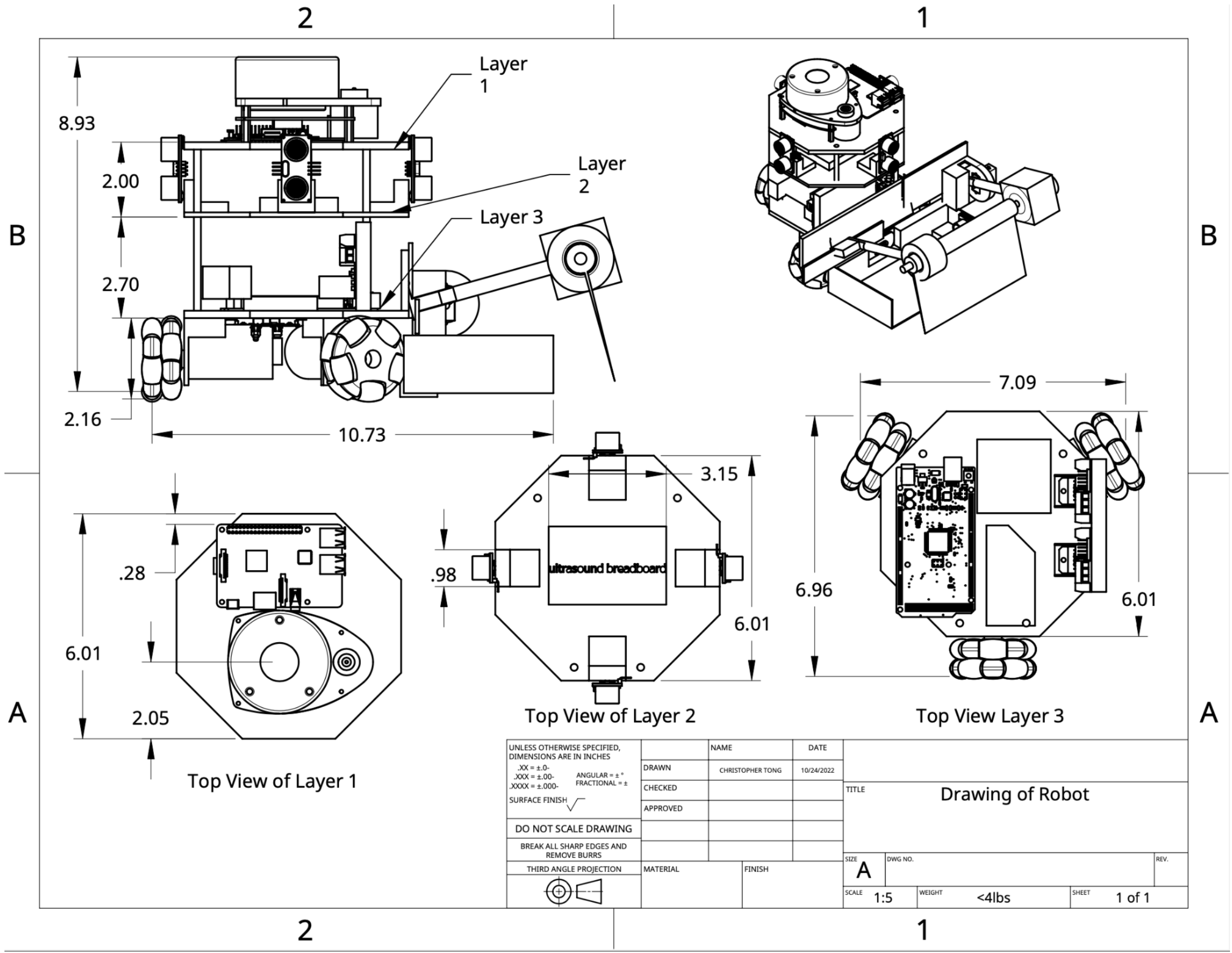
Table 2: Team Contribution Table.

Name	Contributions
Nathalie Cristofaro	- Project requirements section - Block retrieval mechanism report section - Block retrieval mechanism OnShape CAD - Part 3 of video presentation - Editing and revising report - BOM
Liam Toner	- Electrical layout/wiring diagram - Electrical power management - Drivetrain control plan - BOM - Compiled and edited video footage
Christopher Tong	- Robot chassis design section - Robot Onshape CAD chassis - Sensor layout - Researching localization algorithms
Andres Cervera Rozo	- Localization and path planning algorithm - Software architecture with Arduino, Raspberry Pi, and ROS - Drivetrain speed algorithm - Sensor choices - BOM

6.0 Appendices

6.1 Appendix A - Full Rover Major Dimensions

The following drawing outlines the major dimensions of the rover's main components. A pdf version is attached with the submission.



6.2 Appendix B - Full Rover Major Dimensions

The following drawing outlines the major dimensions of the rover's block retrieval mechanism. A pdf version is attached with the submission.