SIMPLE MAZE SOLVING ROBOT

Group Name: **G-5**

Members:

- **Ayush Sinha–Data Science and Engineering**
- **Akshat Singh–Computer Science and Engineering**
- **Yasaswini Pagadala–Electrical Engineering**
- **Siddhant Tyagi – Engineering Physics**
- **Mihir Chandra – Civil Engineering**

Tutor Names:

- Sachin Ruhela
- Dhruv Singhla

Summary:

The simple maze solving robot is a project that involves building a small autonomous robot with three degrees of freedom, capable of navigating through a maze using sensors and programming logic. The robot is designed to use an ultrasonic sensor to detect obstacles and follow a predetermined path to reach the end of the maze.

The robot construction involves assembling a basic chassis with motors and wheel, adding an Arduino board to control its movements and attaching an ultrasonic sensor to enable obstacle detection.

Overall, the project will involve combining basic engineering principles with programming logic to create a functional robot.

Requirements:

- 5MM Acrylic Sheets-1
- Ultrasonic Sensor-2 (Though **3** is preferred)
- 75mm Rubber Wheels-2
- Castor Wheel-1
- Jumper Wires
- L298 Motor Driver-1
- Battery 12V LIPO-1
- M2, M4 Nut & Bolt
- L Bracket (**20x20** profile)
- Aluminum Profile (**20x20** mm)
- Buck Converter-1
- XT 60 M-F Connector-1
- Copper Clad (**1x1** feet)-1
- Microcontroller Board (Node MCU/ Arduino Board with Cable)-1
- Allen Bolt/Screws

Methodology:

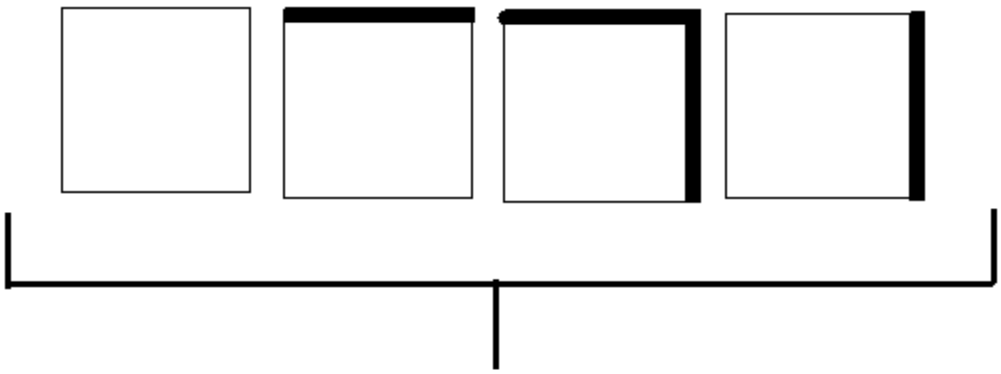
Mazes can be represented as a grid of cells, where some cells are blocked and others are open. The goal is to navigate through the open cells to reach the destination while avoiding obstacles. Solving mazes efficiently requires the use of intelligent algorithms that can explore the maze and find the optimal or near-optimal path.

These include **DFS** (**D**epth **F**irst **S**earch), **BFS** (**B**readth **F**irst **S**earch), **Flood Fill**, **A*** and **Wall Following Algorithm**.

The one we used was-

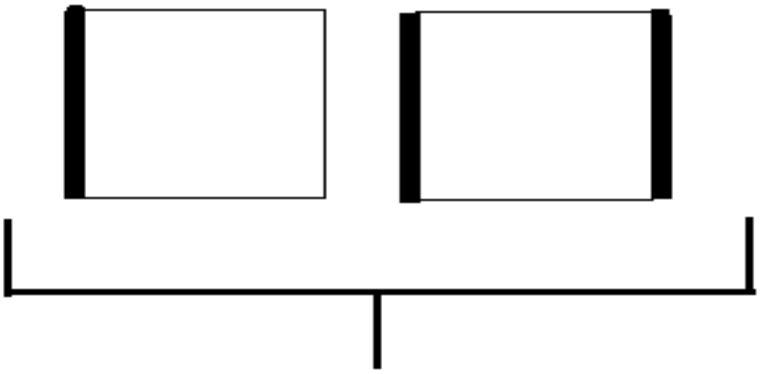
Algorithm: **Left-Hand Wall Follower Algorithm**

***Case1: When the robot doesn't sense any wall at a distance less than or equal to a particular value on its left side, it will turn left.



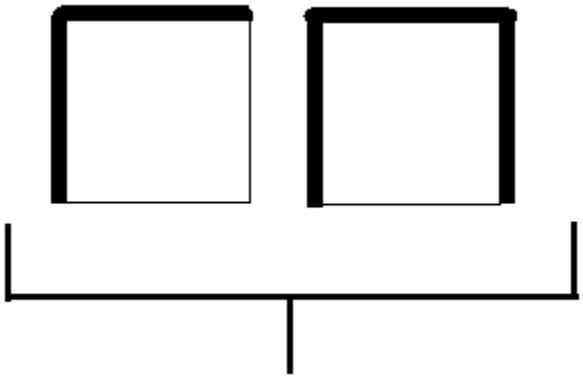
In all these cases, the robot will turn left.

***Case2: At a distance less than or equal to a particular value, if the robot senses a wall on its left side and no wall in front of it, it won't change its path and keep on moving parallel to that wall at a fixed distance.



In all these cases, the robot won't change its path.

***Case3: At a distance less than or equal to a particular value, if the robot senses a wall on both left and front sides, it will turn right.

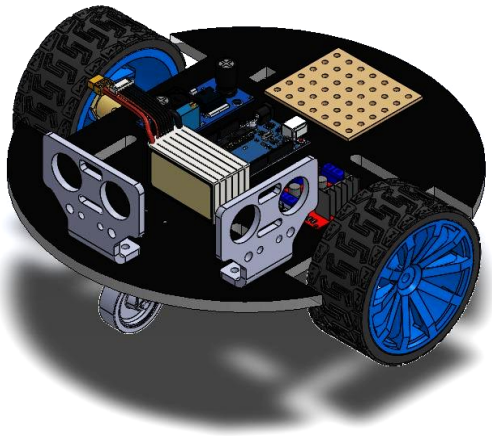


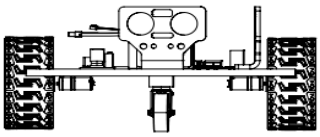
In all these cases, the robot will turn right

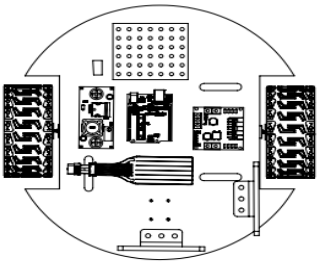
While this algorithm will eventually solve the maze, it is **not** an optimal solution. There are plenty of cases where the bot might end up **stuck in a loop**, such as the **introduction of free walls**.

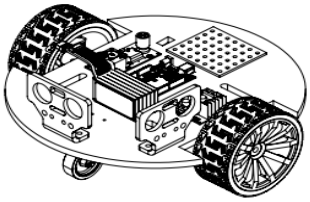
*** Principle behind ultrasonic sensors: Transmitters of ultrasonic sensors generate ultrasonic waves and when there is an obstacle in this case a wall, an echo will be generated. This echo will be captured by receivers. Based on the time gap between emitting and echo receiving, the sensors will calculate the distance between the wall and the robot. Based on the values obtained, the robot will move according to the above three cases.

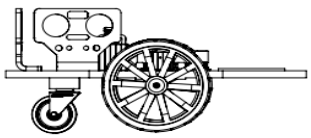
Potential CAD Model:











*** A circular chassis was chosen to provide swift motion during turning; avoid the robot getting stuck during turns.

*** Caster wheel is used to provide the robot with sufficient balance during movement.

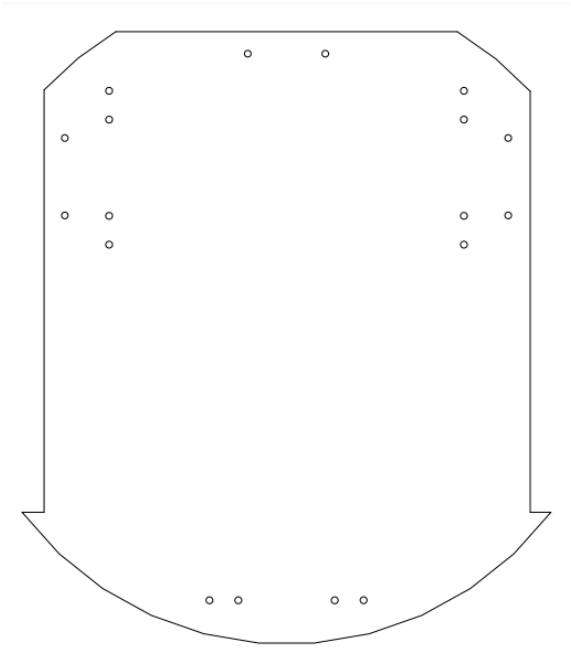
*** Frames for ultrasonic sensors are used to provide additional support to ultrasonic sensors and reduce the movement of ultrasonic sensors while traversing the maze.

**** The sensors are arranged in an L-shape manner; one pointing in the front and other in the left. The left one shall be used to follow the left wall while the front one shall be used to determine when the robot needs to take a turn.

UPDATE:

In reference to the first point made regarding the shape of the chassis, we decided to slightly deviate from an all circular look. Reason being the presence of a lot of empty space after the placement of components and the overall size of the bot was quite large.

Thus, we decided to reduce the breadth of bot by cutting the circular portion from the sides while keeping the front and back still round for ease of turning, as already mentioned.



Also, we added a second story to the bot for mounting ultrasonic sensors, since the reduction of size caused the components to look crumpled and increased chances of interference of signals to the sensors.

Fig - Top Plate

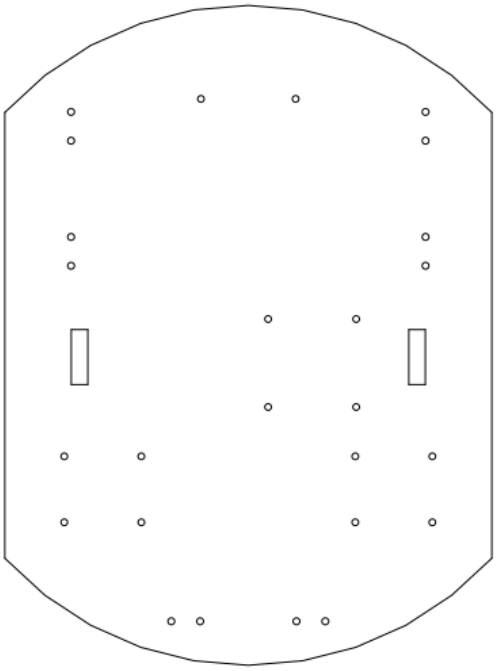
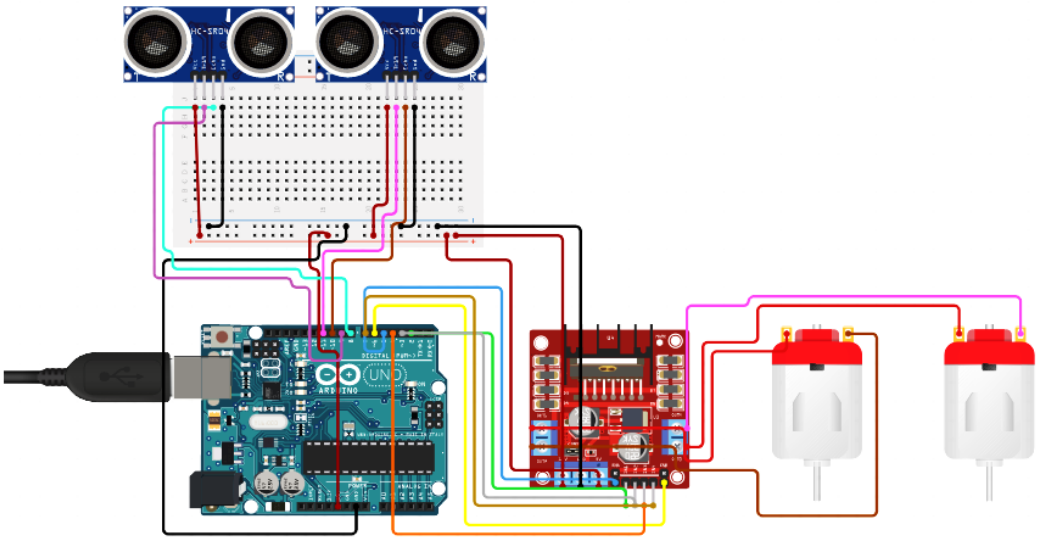


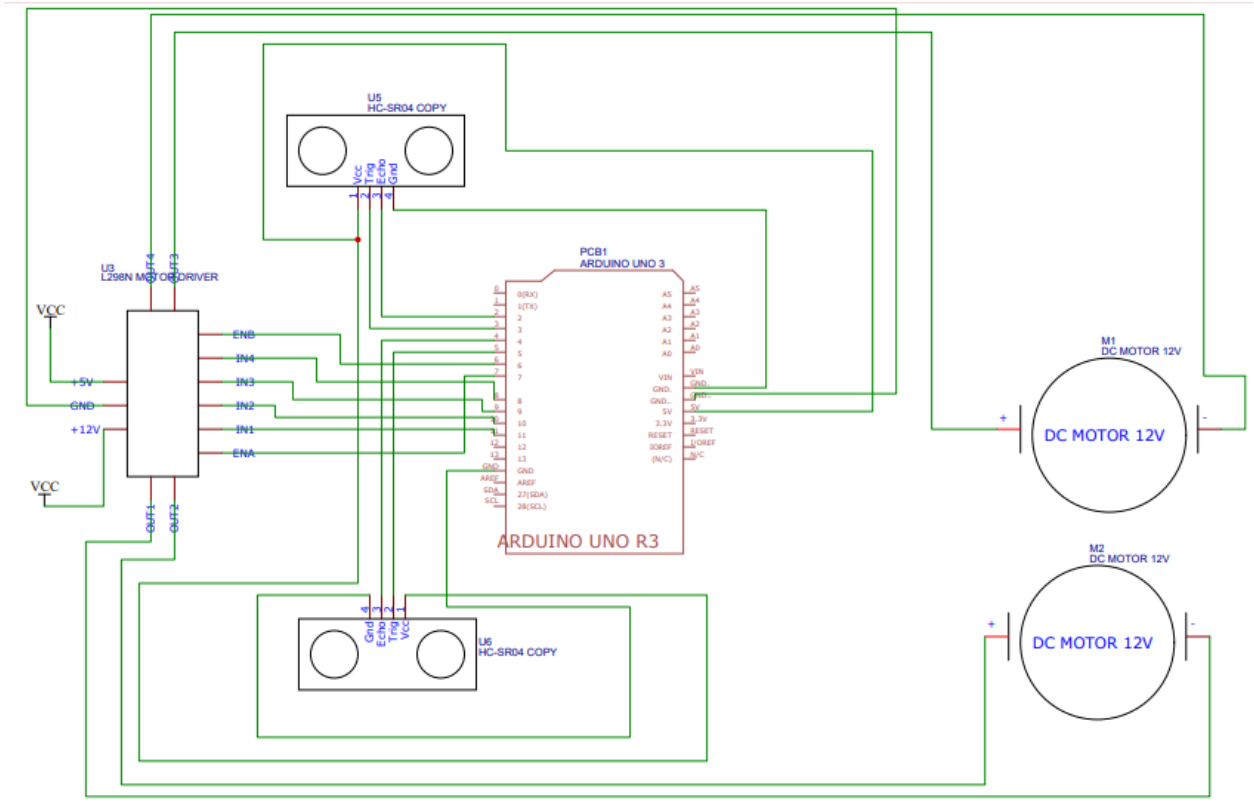
Fig - Bottom Plate

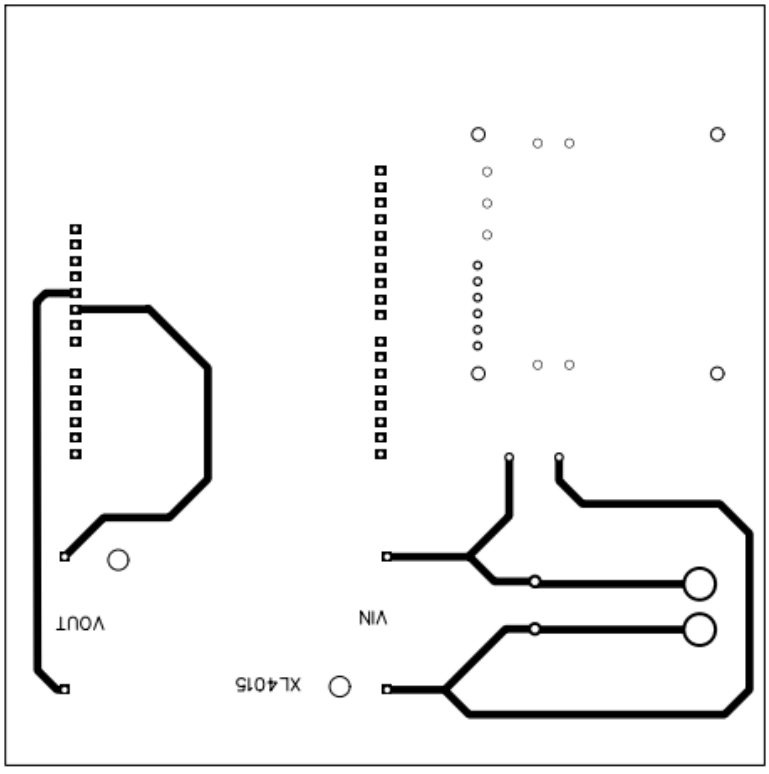
Primitive Circuit Diagram:



Citation: Made using circuit.io

PCB Fabrication:





Limitations of the model:

*** As we are using the left-hand wall follower algorithm, the robot may take a longer path than the shortest possible path to complete the maze.

*** As we are making a robot with three base wheels, there are more chances for it to topple.

*** As we are using sensors only on the left and front sides of the robot, it doesn't have any information about its right side. So, if the width of the path is not enough and all three sides have walls then, while taking the right turn, it will collide with the right wall.

*** As the ultrasonic sensors were placed at a height, the robot is unable to detect the obstacles of less height than this. So, in this case, it will topple down. As it is a robot with three base wheels, there are more chances for it to topple.

Applications:

*** Shortest distance maze solving robot: This is an advanced version of this prototype which uses breadth first search algorithm. This robot will be programmed in such a way that it measures the distance of different paths then finds and stores the shortest path. So this can be used as a form of navigation to achieve the shortest path for situations like: In case of medical emergency, rescue operations, etc.

*** Advanced versions of this can be used for military purposes despite using human efforts like in the cases of underground transportation, finding unknown paths etc.