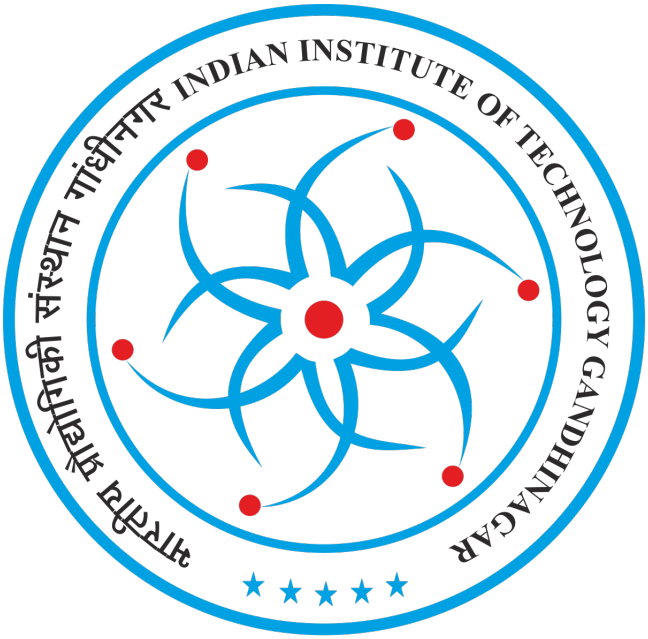


Indian Institute of Technology Gandhinagar

Micro Mouse Project Report
Team Alpha

Maker Bhavan

Aditya Mehta
22110150
Chemical Engineering

Aditya Baderia
23110015
Electrical Engineering

May 2, 2024

1 Problem Statement and Introduction

The problem statement focuses on designing and building a Micromouse with maximum size of 15 cm X 15 cm, which is capable of navigating and solve a maze autonomously. There are no restrictions on the height of the robot. We are provided with a maze problem to resolve. The Micromouse robot should demonstrate autonomous navigation, obstacle avoidance, data collection, and communication capabilities. Students will engage in an iterative design process for the robot, covering aspects of robotics, programming, and autonomous navigation algorithm design.

2 Materials Used

1. Flying Fish IR Sensor (For Obstacle Detection)
2. ESP32 Wroom32 Module
3. L298N Motor Driver
4. N20 DC Motors with Encoder
5. 12V LiPo Battery

2.1 Fabricated Components

1. Wheels
2. PCB
3. Caster Wheel

3 Why this Design?

The IR Sensors are put on an angle of 45 degrees on the PCB due to the reason that some of the mazes are angles and not always rectangular. So, the IR Sensors kept at angles will help in those mazes too.

4 Algorithm Used (Flood Fill Algorithm)

4.1 Why?

The micromouse robot, as per the international micromouse competition regulations, should be completely autonomous, should traverse the maze by itself while discovering the walls and mapping them in the memory.

At the time of placing the robot on the maze, only the starting position, size of the maze and the destination position with respect to the starting position is known by the robot.

4.2 The Algorithm

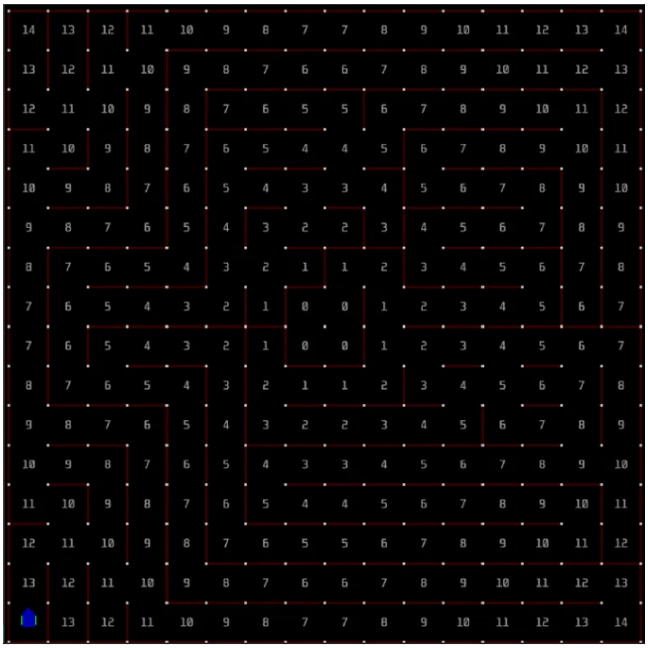
Flood fill is an algorithm mainly used to determine a bounded area connected to a given node in a multi-dimensional array.

We define another 8*8 maze, lets call it the Flood array. We assigned zero to the four destination cells, 1 to the cell which is immediately accessible by the destination cells, and so on. The cell at the cell will get the largest number.

Ideally, if there are no walls, the micromouse will follow the path from highest number to zero. Initially, this is what your Flood array looks like.

As the mouse follows the path, it will encounter walls. This will disrupt the path. Hence, the flood fill will update as new walls are encountered, changing the path to the end point. As the mouse traverses the maze, it will find possible paths to the centre.

In micromouse competitions, the micromouse traverses the maze 4-5 times, which allows it to explore the entire maze and find the shortest path.



While moving continuously through the maze, we need to:

1. Find the value of neighbouring cells from flood array (above)
2. Travel to the neighbouring value with the least value.
3. Detecting walls to its left, right and front.
4. Update the newly found walls in the maze array.
5. Perform the flood fill for the entire flood array.
6. Back to step 1, until the robot moves to the desired position.

4.2.1 The fast run

Once we decide that the mouse has discovered enough cells to find an optimum path, you can bring the mouse back to the starting square, and do the fast run. In the process, the mouse finds the values of its neighbouring cells (from the flood array) and travels to the neighbouring cell with the value 1 less than the present cell.

Back to step 1, and continue until the robot moves to the desired position. During the fast run, we don't need to update the maze array or the flood array as the mouse will only be moving to the cells that are already discovered.

4.3 Code for the Algorithm that we used

The code is uploaded as another file.

5 Main Code

The code is uploaded as another file.

6 Procedure and Timeline

6.1 Understanding the Concept

Our first step was to understand the concept behind the micro mouse and its requirements.

This meant, we needed to understand the concepts of flood fill algorithm, PCB fabrication, 3D printing and circuitry to make our micromouse.

6.2 Designing and 3D Printing the main body

After finalizing the design, we designed it on our own using the software Autodesk Inventor according to the components of our robot such as battery and motors.

6.3 3D Printing other Parts

The Parts such as tyres and the cover for the motor (so that the motor does not come out) were also printed according to the dimensions in our main body.

For using the facility of 3D Printing, we designed the parts using the software Autodesk Inventor.

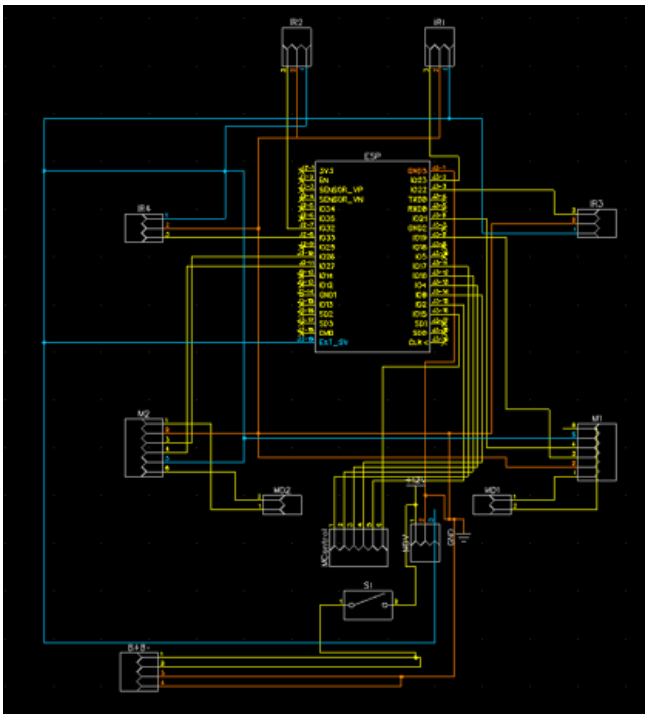
Also, we learnt how to design the basic parts computationally using Python libraries such as CadQuery.

6.4 PCB Design and Prototyping

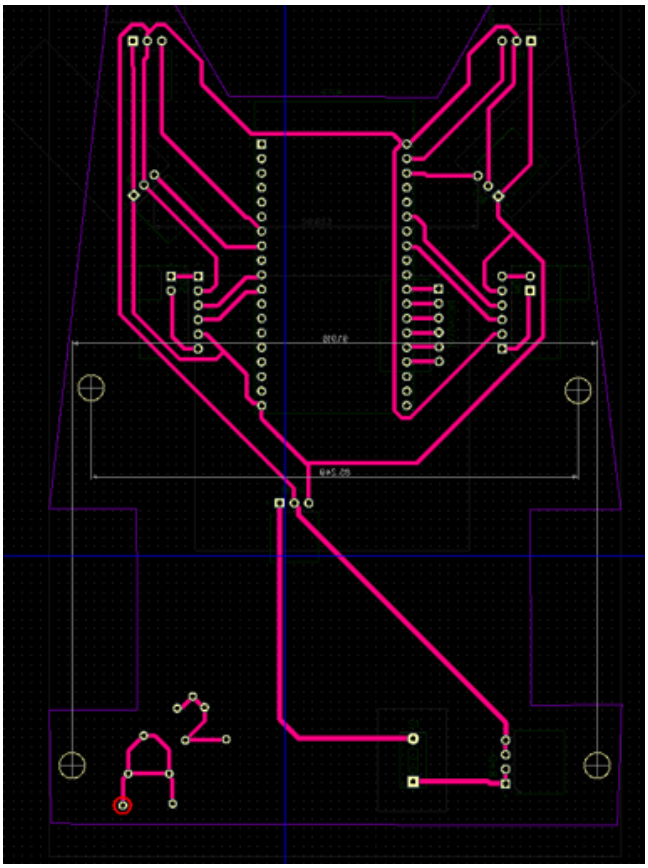
We have designed our own PCB. We had to carefully place the components. We also made sure to widen the grooves where more current was expected, ie, the grooves from the battery and from the motor driver.

While finalising the PCB, we unexpectedly couldn't route the PCB as the ground and power connections for 2 of the IR Sensors were blocked off. To fix this, we flipped the IR sensors' pins so that the connections could be made.

To make the PCB, we made the schematic in a software known as DipTrace. All the components required we kept in the schematic tab and joined them with proper connections. Our Schematic is as shown below:



After this, the layout has to be made in the same software. The final layout of the PCB is as shown below:

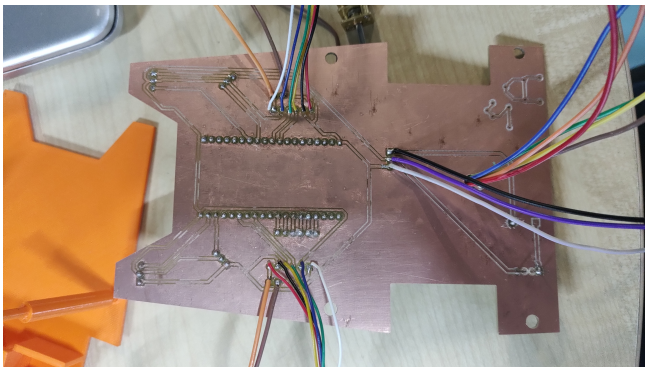


Finally, in another software known as CopperCAM, we make the PCB.

6.5 Soldering

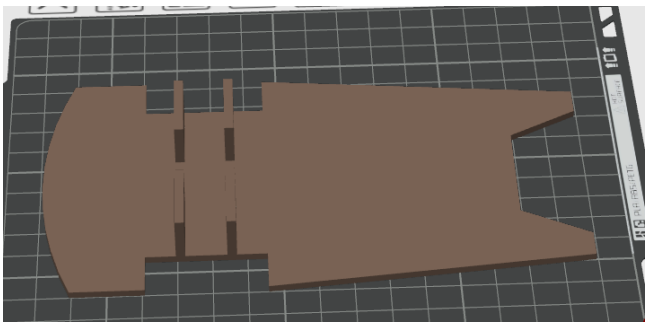
After making the PCB, we need to solder the connection so that the components such as ESP32 and IR Sensors which we have used can be mounted on it.

The final PCB after soldering looks like this:



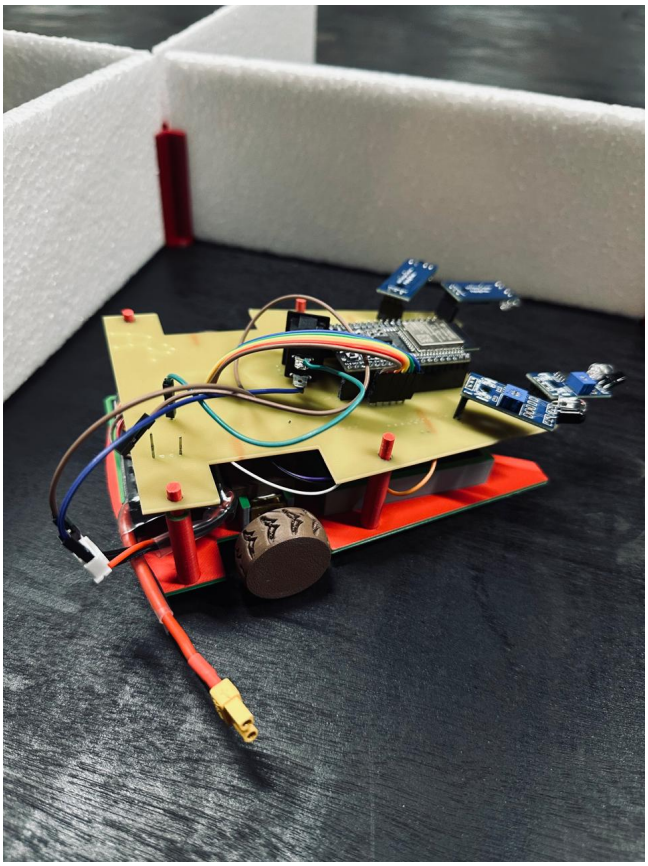
After making this PCB, we needed to make many changes in the main design of our micromouse.

Finally, we came to the most appropriate design which looked like this:



We included a battery holder, a motor driver holder, columns to place the PCB and a caster wheel with a ball bearing, for smoother locomotion.

The final MicroMouse Robot looks like this:



6.6 Checking the calibration

We wrote a basic code to check the calibration of the circuit (just to check if the motors are operating properly according to the outputs of the IR Sensors).

6.7 Flood Fill Simulation

Then as mentioned above that we used the Flood Fill Algorithm. We wrote the code for the same. In order to check the code, we need to simulate it. So, we used a mms micromouse simulator available on git hub. The image of the simulator was shown above.

6.8 Main Code

The final part of the project is writing the main code for the micromouse which take the data from the IR Sensors and operates the motors accordingly and then according to the data recieved from the flood fill algorithm it moves forward and accomplish the task.

7 Learning Outcomes

7.1 Fundamental of Micromouse Robot Design

Through this project, we learned the fundamentals of robot design, ie, integrating physical, electrical and coding elements.

We also gained a lot of practical experience through this project, like setting up IR sensors, how to upload code to ESP32 without fail, how to effectively solder components and how to rotate the micromouse in place.

7.2 Fundamentals of Digital Fabrication for PCB Prototyping techniques

With the help of this project, we learned various techniques of PCB Prototyping, like using space effectively and finding creative solutions for making sure all components are connected with each other.

7.3 Floodfill Algorithm

The flood fill algorithm as explained above is implemented.

7.4 Computational 3D Designing

We also learnt computational 3D Printing using Python Libraries such as Cadquery.

8 Previous Efforts before the final design

Before all this stuff and steps taken, this was not the method which we thought of. Our first method includes the use of STM32 instead of ESP32 along with the emitters and recievers separetely. We made the schematic for making the PCB for our system. But then, due to large number of components in very small place, we need to make a double sides PCB which is really very tough. So, finally the thought was to make single sided PCB only and hence to moves to ESP32 and using IR Sensors.

References

[1] M. A. Dharmasiri, "Micromouse from scratch— Algorithm- Maze traversal—Shortest path—Floodfill," Medium, Dec. 12, 2021. [Online]. Available: https://medium.com/@minikiraniamayadharmasiri/micromouse-from-scratch-algorithm-maze-traversal-shortest-path-floodfill-741242e8510

[2] "Building — Bulebule 1.0.0 documentation." [Online]. Available: https://bulebule.readthedocs.io/en/latest/building.html