MICROMOUSE

Micromouse is a micro sized robot equipped with devices to explore and solve a maze in the shortest time. The robot is built in such a way that it can explore the maze and store information on it. The stored information is later analyzed to generate the optimum path. The concept of micromouse was introduced because of the challenge offered by IEEE Spectrum in May 1977, where they challenged their reader to design and build a maze-solving “micromouse” which must have its own self-contained logic and memory.

Our project “Micromouse” is also guided by the rules of International Micromouse Competition where an autonomous maze-solving robot has to be developed which is capable of solving a 16*16 maze with each cell of size 18cm. The maze can have different configurations, but the start is always at one of the four corners and the destination is always at the center of the maze. We represented Nepal with our robot in the International Micromouse Competition at Techfest 2019/20 at IIT Bombay, India.

Implementation Details

The robot was built of a PCB board to minimize its size. All the electronics components including sensors, microcontrollers and motors were integrated into the PCB itself. A novel algorithm was developed to allow the robot to navigate and solve maze optimally in the shortest time possible.

Components and Materials: PCB board, STM32 Bluepill, MPU-9250 IMU, VL53L0X sensor, N230 Micro-gear Motor, L293D motor-driver, 42D rubber Wheel, 3PI Castor Wheel, Buck Converter, and LiPo Battery

Design Implementation

Designing a micro robot requires a lot of iterations. Our first move towards the development was system architecture realization. It included the considerations of the components required and their systematic interconnections. Information of components and system architecture enabled us to develop a visual realization on KiCAD. The model was designed in every detail possible.

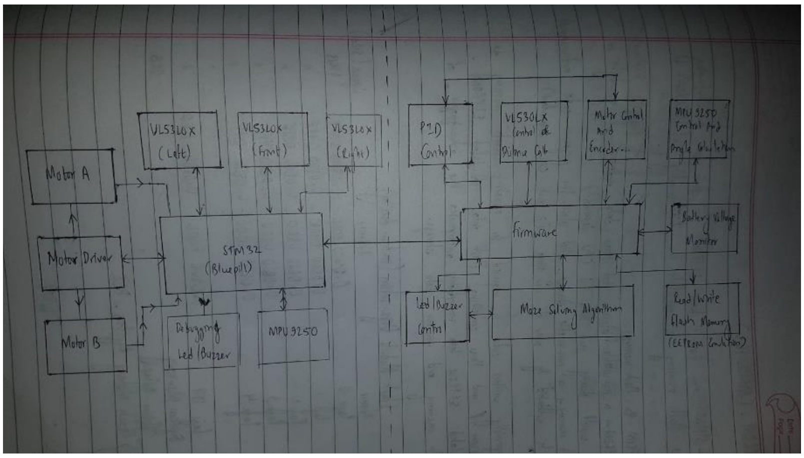


Figure 1 System Block Diagram

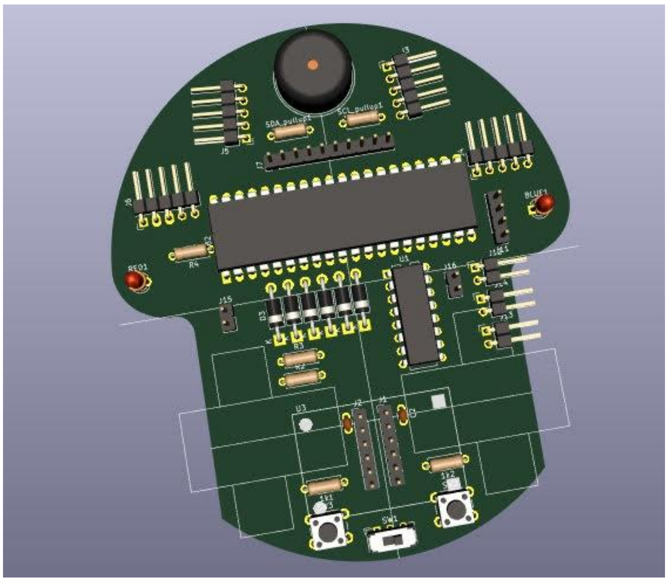


Figure 2 KiCAD Model

The KiCAD model was allocated with spaces for motors and sensors hence enabling us to outline the body of the robot as in Figure 2.

The body of the robot was fabricated using PCB board and the circuit was printed Figure 3. The image of the circuit is transferred onto the PCB through Photolithography process and etching method was applied to remove the unwanted copper from the PCB giving the required circuit only.

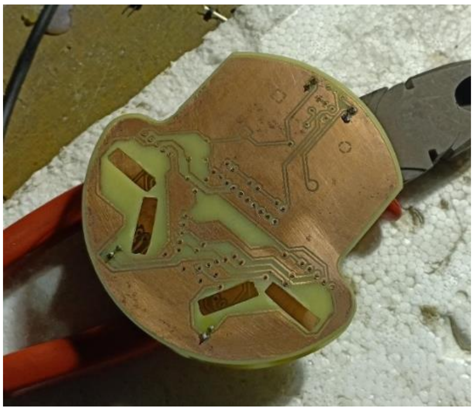


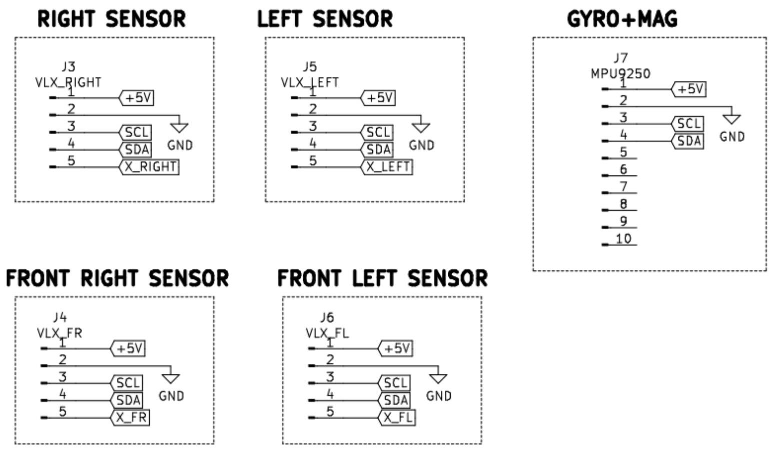
Figure 3 Body of Robot

Working Principle of System

The STM32 microcontroller serves as the central processing unit of the Micromouse, interfacing with various components such as sensors, motor drivers, LEDs, and a buzzer. The sensors, including VL53L0X distance sensors and an MPU-9250 IMU, are connected to the STM32 via the I2C bus. Communication between the STM32 and a PC is established through a UART interface, while GPIO pins are configured as INPUT, OUTPUT, or INTERRUPT based on their specific functions. The firmware embedded in the microcontroller contains all the necessary directives to execute control actions.

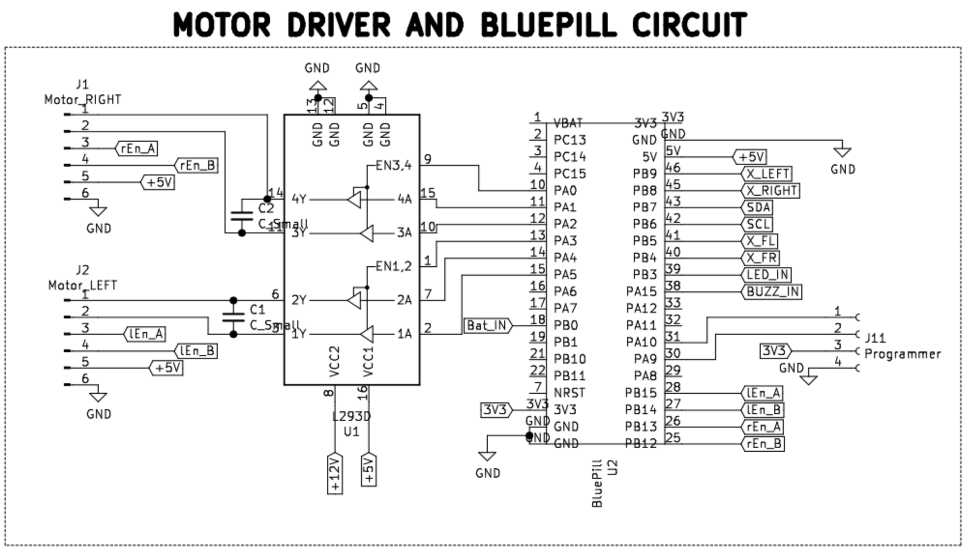
Sensor Data Acquisition and Processing

The STM32 microcontroller operates as the master device, sequentially requesting data from all connected slave devices (sensors) using their unique addresses. It performs calculations on the received data to determine distances and angles, adjusting for any offsets or erroneous values. This processed data is then used to make informed decisions for the Micromouse's navigation.



Motor Control and Feedback

Motor encoders, which are tiny hall-effect sensors, are interfaced with the STM32 through interrupts. The microcontroller counts the interrupts generated by the encoders to precisely control motor movements, ensuring steady motor speeds, accurate travel distances, and computation of the total distance traveled. Based on the encoder feedback, the STM32 sends Pulse Width Modulation (PWM) signals to control motor speed and digital signals (1 or 0) to control motor direction.



PID Control

The PID controller in the STM32 microcontroller performs three main tasks:

1. **Collision Avoidance**: Keeps the Micromouse centered between side walls by balancing the distance readings from the right and left VL53L0X sensors.
2. **Angle Control**: Ensures precise rotations by comparing the desired angle with the current angle from the MPU-9250.
3. **Motor Synchronization**: Synchronizes the pace of both motors by comparing the encoder counts from the right and left motors.

STM32 calculates the control outputs using the tuned gains for each PID controller and sends the appropriate control signals to the motor driver.

Power Management

The Micromouse is powered by three single-celled 680mAh LiPo batteries connected in series, providing a stable voltage in a compact form factor. To monitor battery depletion, a simple voltage divider steps down the input voltage to the in-built 12-bit ADC of the STM32. The ADC samples the battery voltage at the beginning of each loop, and a buzzer sounds when the interpolated voltage falls below a threshold of 9.6V, preventing the robot from running out of power during operation.

Algorithm

The microcontroller used for this project has 128 KB flash memory and SRAM upto 20 KB. Different libraries and Application Programming Interface (API) increased the bulkiness of the code challenging the memory limitation. Hence an efficient algorithm was required to address the code complexity and memory limitation. Since the maze to be solved was standard 16x16 which can be considered as a small maze, and upon evaluation of various algorithms for solving the maze, Tremaux algorithm was found to be the suitable option considering our constraints.

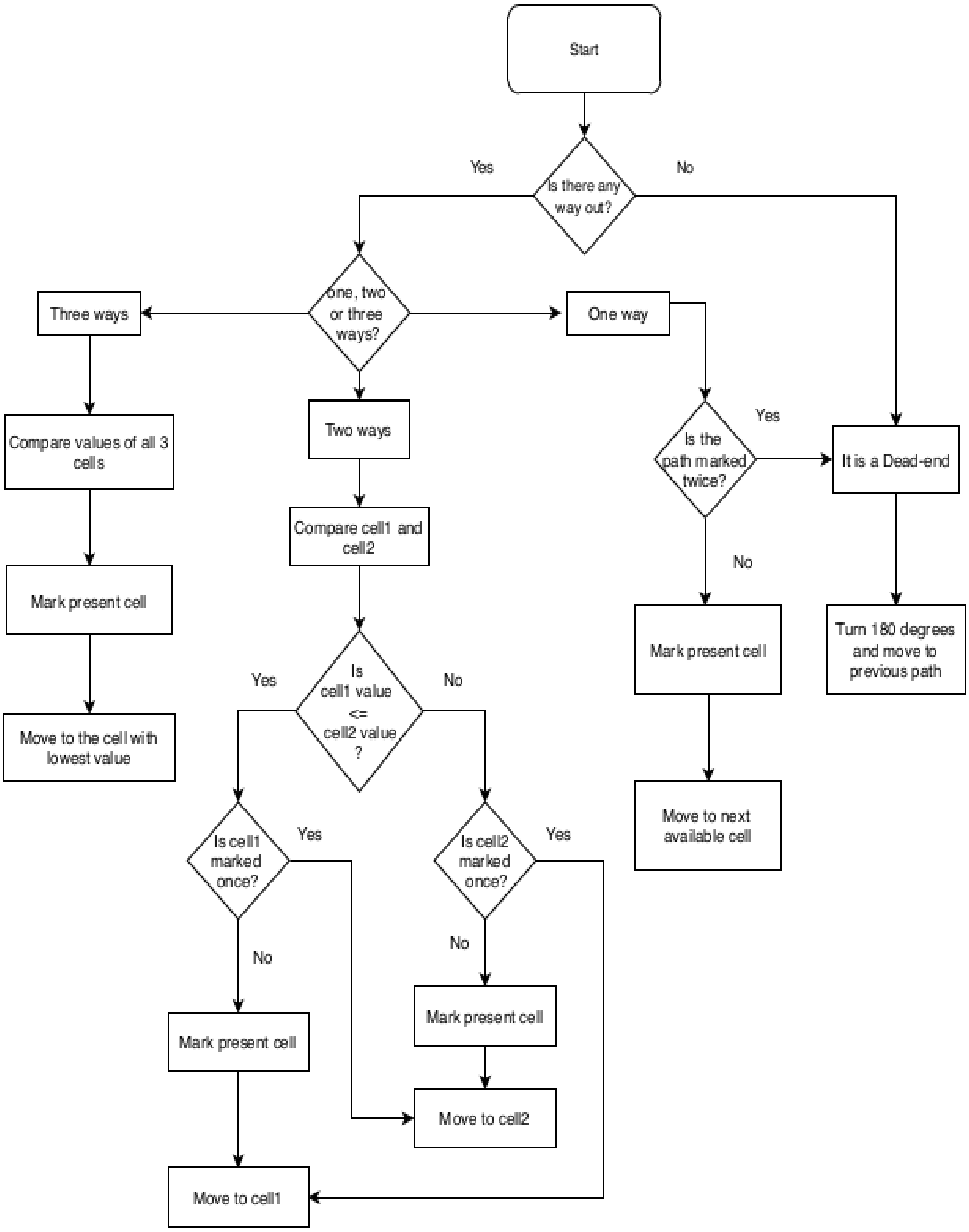
Tremaux algorithm allows robot to explore the maze in a systematic manner marking nodes once, twice or leaving them unmarked. Nodes marked twice are not considered in the final path to avoid dead-ends and loops. Upon solving the maze, the algorithm outputs an optimal path among the visited paths with nodes marked once. The algorithm is deterministic and ensures that every path is explored at least once consuming more time during the dry run. To overcome this problem, Tremaux algorithm was optimized with the potential value algorithm. Potential value algorithm first maps the maze by assigning potential values to each node, generally higher potential on the outer side and gradually decreasing potential towards center (destination) as in Figure 4. The algorithm directs the robot to lower potential constantly prohibiting it from unnecessarily exploring the maze and reducing the search time significantly during the dry run.

Thus, a fused algorithm was implemented on the robot by integrating Potential Value with Tremaux algorithm.

14	13	12	11	10	9	8	7	7	8	9	10	11	12	13	14
13	12	11	10	9	8	7	6	6	7	8	9	10	11	12	13
12	11	10	9	8	7	6	5	5	6	7	8	9	10	11	12
11	10	9	8	7	6	5	4	4	5	6	7	8	9	10	11
10	9	8	7	6	5	4	3	3	4	5	6	7	8	9	10
9	8	7	6	5	4	3	2	2	3	4	5	6	7	8	9
8	7	6	5	4	3	2	1	1	2	3	4	5	6	7	8
7	6	5	4	3	2	1	**0**	**0**	1	2	3	4	5	6	7
7	6	5	4	3	2	1	**0**	**0**	1	2	3	4	5	6	7
8	7	6	5	4	3	2	1	1	2	3	4	5	6	7	8
9	8	7	6	5	4	3	2	2	3	4	5	6	7	8	9
10	9	8	7	6	5	4	3	3	4	5	6	7	8	9	10
11	10	9	8	7	6	5	4	4	5	6	7	8	9	10	11
12	11	10	9	8	7	6	5	5	6	7	8	9	10	11	12
13	12	11	10	9	8	7	6	6	7	8	9	10	11	12	13
14	13	12	11	10	9	8	7	7	8	9	10	11	12	13	14

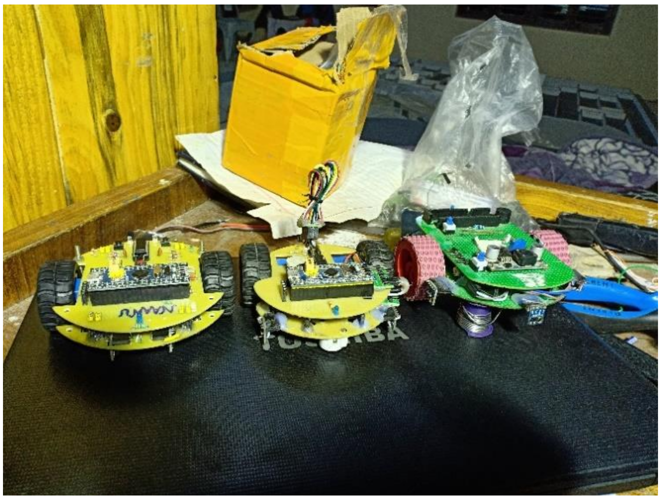
Figure 4 Mapping of values for Potential value algorithm

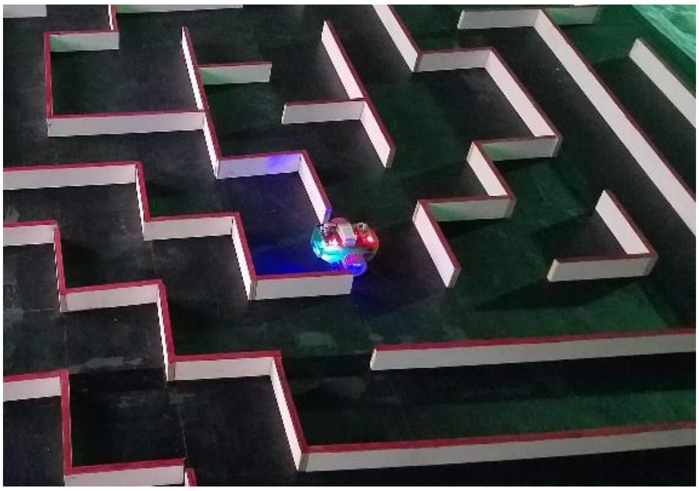
Figure 5 Optimized Algorithm Flowchart



Results

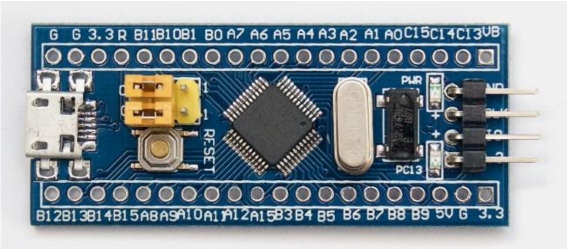
- Built a micro-sized robot.
- Calibrated sensor with tuned PID parameters.
- Straight motion between the walls was obtained with few contacts at times.
- The robot performed accurate 90^{o} rotations.
- The robot explored mazes successfully.
- The robot traversed the shortest path obtained from the paths visited.
- Optimized algorithm reduced the search time by 35.28% and run time by 9.50% on average.
- Participated on International Micromouse Competition at Techfest 2019/20 at IIT Bombay, India.
- Published an article on “Optimization of Tremaux Algorithm Using Potential Value” in International Journal of Advanced Engineering (IJAE). https://ictaes.org/wp-content/uploads/2020/09/IJAE-2020-Vol.03-No.02/7_Sanjaya_Vol3_No2.pdf?ckattempt=1





Components

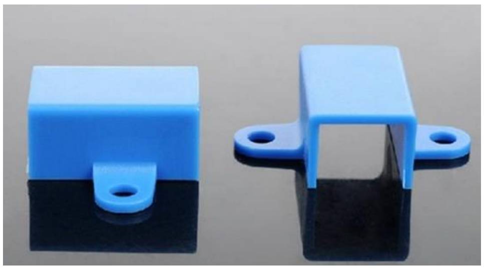
1. **STM32 Bluepill**



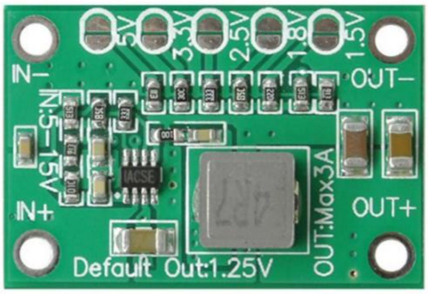
2. **USB-TTL Adapter**

3. **Wheels**

4. **Motor Mount**

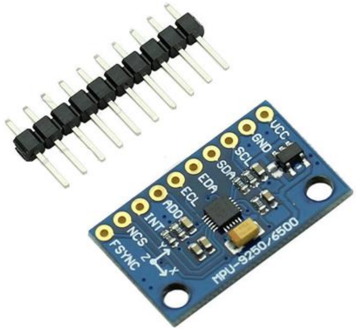


5. **Buck Converter**

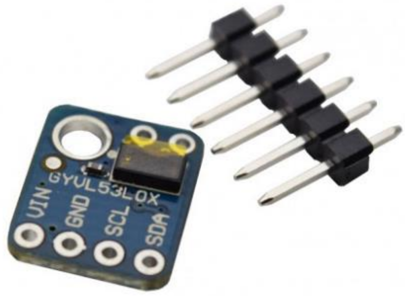


6. **Lipo Battery**

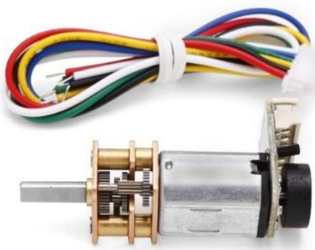
7. **MPU-9250**



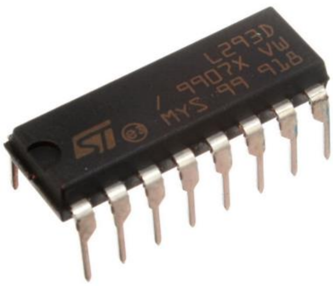
8. **VL53L0X**



9. **N20 Micro-gear Motor**



10. **L293D**



11. **3PI Castor Wheel**


- BLL 852540
3.7V 680mAh
+ 151222