ABSTRACT

This project presents the development of a maze-solving robot employing advanced technologies for efficient navigation. The robot integrates ultra-sonic sensors for real-time obstacle detection, ensuring smooth traversal through complex maze configurations. Utilizing N20 geared motors equipped with encoders, precise vehicle movement is achieved, enhancing navigation accuracy. An Arduino Mega 2560 serves as the central microcontroller, orchestrating sensor data processing and motor control. The implementation of the flood fill algorithm enables the determination of the shortest path within the maze, optimizing traversal efficiency. Additionally, Bluetooth connectivity facilitated by the HC-05 module allows seamless communication with an Android application, enabling remote control and monitoring of the robot's movements.

Keywords: maze-solving robot, flood fill algorithm, ultrasonic sensors, Arduino Mega 2560, Bluetooth connectivity, N20 geared motors.

INDEX

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Robotics is a field that is always pushing the envelope and solving difficult problems by combining cutting-edge technologies and creative problem-solving techniques. Robots that can solve mazes are among the many applications of robotics that are exciting because they are instances of autonomous systems that can navigate complex surroundings. In this project, we construct a robot that can solve mazes by using state-of-the-art methods to provide smooth control and traversal in maze settings.

The maze-solving robot is fundamentally a representation of autonomous navigation, where it is able to dynamically adjust to its environment through the combination of sensors, algorithms, and control systems. A series of sensors positioned to provide environmental feedback enable the robot to recognize and react to hurdles in real time. Sophisticated control mechanisms that regulate exact movement complement this perceptive capacity, allowing the robot to navigate intricate paths accurately and effectively.

An Arduino Mega 2560 microcontroller is equipped with a complex computational framework that powers the robot's decision-making mechanism. In order to provide intelligent navigation through the labyrinth, this central processing unit coordinates the integration of sensor data processing and motor control algorithms. The flood fill algorithm, a potent method for figuring out the shortest path around the maze, is the foundation of this computational system. The flood fill algorithm minimizes time and energy consumption by optimizing the robot's traversal path through systematic exploration of the maze's structure.

Moreover, the HC-05 module adds Bluetooth connectivity to the robot's control interface, extending beyond its actual parts. With the help of this connectivity, users may effortlessly communicate with an Android application and remotely control and observe the motions of the robot. In addition to improving user experience, this interactive interface highlights the robot's adaptability and agility in a range of operational circumstances.

This research is essentially an example of how technological innovation and interdisciplinary knowledge come together in the field of robotics. Our goal is to create a maze-solving robot that stimulates curiosity and investigation in the fields of science and technology while navigating through mazes on its own by utilizing cutting-edge algorithms and control technologies. As we set out on this project, we see a time when intelligent robots will be essential in enhancing human abilities and reshaping the automation and discovery landscape.

1.1 PURPOSE

The primary objective of this project is to develop and build a maze-solving robot with cutting-edge features including accurate motor control mechanisms, flood fill algorithm implementation for optimal pathfinding, and ultrasonic sensors for obstacle recognition. The main goal is to build an autonomous robot that can effectively and independently navigate through intricate maze layouts. Furthermore, the project intends to offer an Android application-enabled Bluetooth interactive control interface that will let people remotely manipulate the robot's motions and track its advancement in real-time. In addition to its functional objectives, the project aims to stimulate curiosity and interest in technology, engineering, and robotics by providing a forum for community learning and creativity.

1.2 SCOPE

The scope of this project encompasses the design, development, and implementation of a maze-solving robot utilizing advanced technologies such as ultrasonic sensors, precise motor control mechanisms, and the flood fill algorithm. The project includes the following key components:

- Design and development of a maze-solving robot using ultrasonic sensors for obstacle detection.
- Integration of N20 geared motors with encoders to ensure precise movement control within the maze.
- Implementation of the flood fill algorithm for optimized pathfinding, aiming to determine the shortest route to navigate the maze.
- Focus on achieving autonomous navigation, allowing the robot to explore and navigate through various maze configurations.
- Hardware assembly, sensor integration, and algorithm implementation are the primary aspects of the project.
- Basic software development for controlling the robot's movement and processing sensor data.
- Testing and validation to ensure the functionality and performance of the robot in maze navigation tasks.
- Documentation of the design process, implementation details, testing procedures, and results for future reference and replication.
- Potential for future enhancements and extensions, such as additional sensors, advanced algorithms, or improved control mechanisms, depending on project outcomes and requirements.

CHAPTER 2: LITERATURE REVIEW

Swarna Latha, et al [1] The paper discusses swarm robotics implementation using a Flood Fill algorithm for maze-solving, where robot agents share information wirelessly to collectively navigate through the maze in a decentralized manner, proving effective for moderate-sized mazes.

H. Dang, J. Song ,et al [2] Provided an efficient algorithm for maze solvihng robot is the Wall Follower method, which relies on following the wall on one side until the exit is found.

M. Nadour and L. Cherroun [3] This paper showcases the implementation of basic and modified flood-fill algorithms on an Arduino-UNO card for wheeled mobile robot navigation in complex mazes, utilizing ultrasonic sensors for obstacle detection. Experimental results confirm the effectiveness and autonomy of the algorithms, enabling safe and efficient maze-solving by the robot.

I. Elshamarka and A. Bakar Sayuti Saman [4] This paper presents a maze-solving robot utilizing a flood-fill algorithm, ultrasonic range-finders for wall detection, and a PI(D) controller for straight-line correction, demonstrating autonomous navigation capabilities and route optimization.

R. Doni , et al [5] This paper presents the maze recognition technique which is also crucial in robot control systems.

R. Covaci ,et al [6] This paper presents the maze solving algorithm for solving maze and finding shortest path.

P. Studi ,et al [7] The paper presents the development of a Line Maze Solver robot using Arduino Uno R3 and TCRT5000 sensors, employing the Left Hand Rule algorithm and PID tuning for navigation optimization.

S. Iskandar ,et al [8] The paper describes a Line Maze Solving Robot utilizing a Short Path Finder Algorithm with three modes, implementing an improved PID-based motor speed controller to navigate through a maze and find the shortest path.

R. Kumar et al [9] The paper discusses swarm robotics implementation using a Flood Fill algorithm for maze-solving, where robot agents share information wirelessly to collectively navigate through the maze in a decentralized manner, proving effective for moderate-sized mazes.

I. R. Fahmi and D. J. Suroso [10] The paper introduces a simulation program in Python for indoor robot navigation, aiming to teach basic principles joyfully, comparing five maze-solving algorithms and discussing their performance through simulated experiments to aid in robotics education.

D. Horváth ,et al [11] The article presents a technical implementation of controlling a small mobile 3-Pi robot navigating a maze using reflectance infrared sensors and a single-layer neural network with Hebbian learning to adjust weights, showing the efficacy of the bipolar sigmoid activation function for maze-solving.

CHAPTER 3: OVERALL DESCRIPTION

3.1 FUNTIONAL REQUIREMENTS:

- **Obstacle Detection:** The robot must be able to detect obstacles using ultrasonic sensors and adjust its path accordingly to avoid collisions.
- **Precise Movement Control:** The robot should utilize N20 geared motors with encoders to ensure precise movement control, enabling it to navigate through narrow passages and execute turns accurately.
- **Autonomous Navigation:** The robot must be capable of autonomously navigating through maze configurations, exploring different paths to identify the shortest route to the maze exit.
- **Pathfinding Algorithm:** Implementation of the flood fill algorithm to determine the shortest path from the robot's current position to the maze exit, optimizing traversal efficiency.
- **Real-time Decision Making:** The robot's control system should process sensor data and execute navigation commands in real-time, allowing it to adapt its trajectory based on the detected obstacles and maze topology.
- **User Interface:** Integration of Bluetooth connectivity with an Android application to provide a user-friendly interface for remote control and monitoring of the robot's movements.

3.2 NON-FUNCTIONAL REQUIREMENTS:

Performance:

- The robot should be able to navigate in an efficient manner, solving mazes in a reasonable amount of time while reducing traversal errors.

Reliability

- High levels of reliability should be demonstrated by the system, with few instances of sensor problems, motor failures, or software errors that could impair the robot's ability to navigate.

Safety

- Safety should be the first priority in the design and operation of the robot, with the goal of reducing the possibility of unintentional encounters with users or spectators during navigation or accidents with barriers.

User Experience

- To improve the overall user experience when performing tasks involving remote control and monitoring, the Android application's user interface should be clear, responsive, and simple to use.

3.3 HARDWARE REQUIREMENTS:

1. Arduino Mega 2560
2. Ultrasonic Sensors
3. N20 Geared Motors with Encoders
4. Motor Driver
5. PCB's
6. Wheels
7. Battery Pack
8. Bluetooth Module (HC-05)
9. Circuit Component
10. Android Device

3.4 SOFTWARE REQUIREMENTS:

1. Arduino IDE
2. Bluetooth communication libraries for Arduino
3. Programming languages: C/C++ for Arduino
4. Windows XP/2003 or higher version of Windows OS

3.5 FLOW CHARTS

The figure 3.1 is the flow chart which describes the flow the control of the system.

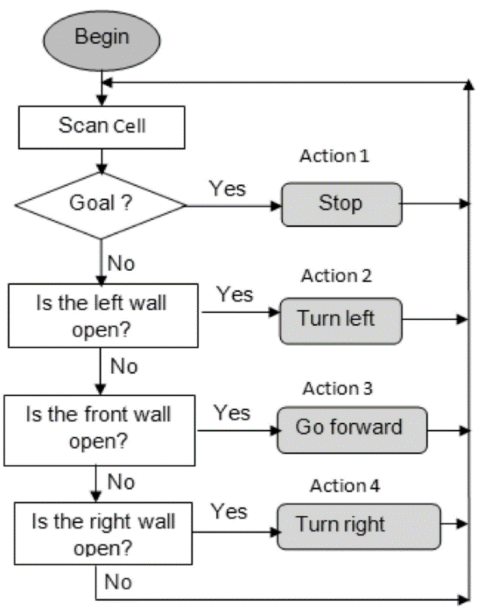

Figure 3.1 Flow chart

CHAPTER 4: IMPLEMENTATION

4.1 Arduino Mega 2560

The Arduino Mega 2560 is the main microcontroller in our maze-solving robot project, handling motor control, sensor data processing, and system administration. The Arduino Mega complies with the project specifications in the following ways:

- **Processing Sensor Data:** The Arduino Mega is a good choice for processing sensor data from ultrasonic sensors used for obstacle detection because of its large memory and processing capacity. It has the ability to interpret sensor inputs quickly and make judgments in real time based on impediments that are recognized.
- **Motor Control:** The Arduino Mega can link with motor driver modules to control the movement of N20 geared motors with encoders thanks to its many digital and analog pins. With its exact control over motor speed, direction, and location, it makes it possible to navigate the maze environment with accuracy.
- **Algorithm Implementation:** To find the shortest route around the maze, the Arduino Mega can run the flood fill algorithm. Because of its processing power, the algorithm can be implemented effectively, optimizing the robot's traversal path and consuming the least amount of time and energy.
- **Bluetooth connectivity:** To provide wireless connectivity with an Android application, the Arduino Mega may speak with the HC-05 Bluetooth module. It can communicate sensor data for monitoring and feedback, and it can receive commands from the application to control the robot's movements.

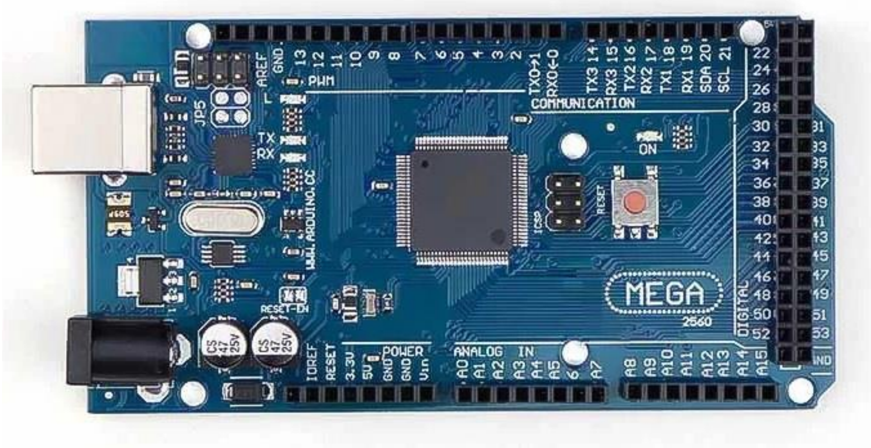

Figure 4.1 Arduino Mega microcontroller

4.2 Ultrasonic Sensors

Ultrasonic sensors are essential to our maze-solving robot project because they identify obstacles and give the machine feedback about its surroundings. This is how ultrasonic sensors fit the project specifications.

- **Obstacle detection:** High-frequency sound waves are emitted by ultrasonic sensors, which track how long it takes for the waves to return after colliding with an obstruction. The robot can effectively navigate around obstacles in its route by using this information to detect their presence and distance.
- **Sensing Range:** Ultrasonic sensors can identify obstacles in the labyrinth environment because of their normal several-meter sensing range. The range of the sensor can be calibrated or changed to suit the particular needs of the project.
- **Accuracy:** The robot can precisely detect obstacles and determine the necessary motions to avoid them thanks to the accurate distance measurements provided by ultrasonic sensors. Ensuring safe and effective navigation through the maze depends on this accuracy.
- **Integration with Arduino:** Ultrasonic sensors are suitable with the project's hardware platform since they can readily interface with the Arduino Mega 2560 microcontroller. The integration procedure is made simpler by the availability of Arduino libraries and code samples for integrating with ultrasonic sensors.
- **Low electricity Consumption:** Because ultrasonic sensors use very little electricity, they can be used in battery-operated devices like the robot that solves mazes. The robot's lifetime and overall energy efficiency are enhanced by its low power consumption.

Figure 4.2 Ultrasonic sensors

4.3 N20 Geared motors with encoders:

N20 geared motors with encoders are used as the propulsion mechanism for our maze-solving robot project, offering accurate movement control and feedback. N20 geared motors with encoders meet the project criteria in the following ways:

- **Propulsion:** To move the robot across the maze environment, N20 geared motors provide the mechanical power required. Their small size and strong torque output allow them to be used for controlling the robot's wheels and navigating through tight spaces.
- **Accurate Movement Control:** Accurate speed and position adjustments are possible because to the exact movement control offered by geared motors with encoders. The robot can keep a constant speed and direction while navigating due to the encoders, which measure the motor shaft's rotation.
- **Speed and RPM:** The N20 geared motors that are being described have a 6 volt rated voltage and a 150 RPM (Rotations Per Minute) rotational speed. This pace can be dynamically changed or regulated to meet the demands of the maze-solving challenge, guaranteeing maximum effectiveness and performance.
- **High Torque:** The robot can overcome obstacles and go uphill in the maze environment thanks to the high torque output of geared motors. High torque and exact control work together to provide dependable and durable movement performance.

Figure 4.3 N20 geared motor with Encoder

4.4 L298N Motor Driver:

The L298N motor driver is a crucial part of our maze-solving robot project since it regulates the direction and speed of the N20 geared motors with encoders. The L298N motor driver complies with the project specifications in the following ways:

- **Motor Control:** Suitable for operating the N20 geared motors utilized in the robot, the L298N motor driver offers bidirectional control for two DC motors. It enables precise movement control by individually regulating the motors' speed and direction.
- **Voltage Compatibility:** The 6-volt power supply used in the project is one of the several operating voltages that the L298N motor driver is compatible with. With its ability to manage voltage inputs as high as 35 volts, it offers versatility and compatibility with various motor setups.
- **Current Handling:** The L298N motor driver is appropriate for driving motors with a range of power requirements since it can tolerate large current loads. It guarantees adequate power delivery for the N20 geared motors with a maximum continuous output current of 2 amps per channel and a peak output current of 3 amps per channel.
- **Pulse Width Modulation (PWM) Control:** The L298N motor driver facilitates PWM control, which enables the motors to be smoothly speed-controlled. Precise movement control in the maze environment is made possible by the motor driver's ability to change the motor's rotating speed through adjustments to the PWM signal.

Figure 4.4 L298N Motor driver

4.5 HC-05 Bluetooth module:

For our maze-solving robot project, the HC-05 Bluetooth module functions as a communication interface between the robot and an Android device, allowing for remote control and monitoring. Here's how the HC-05 Bluetooth module meets the project requirements:

- **Wireless Communication:** With the HC-05 Bluetooth module, the maze-solving robot can communicate wirelessly with an Android device. It establishes a Bluetooth connection, which allows data exchange and control orders to be sent wirelessly.
- **Bluetooth Version:** The HC-05 module uses Bluetooth version 2.0 + EDR (Enhanced Data Rate), which ensures dependable and fast data transmission between devices. It works with a wide range of Bluetooth-enabled devices, such as smartphones, tablets, and PCs.
- **Serial Communication**: The HC-05 module connects with the Arduino Mega 2560 microcontroller via UART (Universal Asynchronous Receiver-Transmitter) serial protocol. It may be set up to send and receive serial data, allowing for bidirectional connection between the robot and the Android device.
- **Range:** The robot may be controlled and monitored within the maze environment thanks to the HC-05 module's usual communication range of up to 10 meters, or roughly 33 feet.

Figure 4.5 HC-05 Bluetooth module

4.6 Lithium-ion Batteries:

Lithium-ion batteries are essential components for any maze-solving robot project since they provide power throughout the system. They comply with the project requirements in the following ways:

- **Power Source:** The maze-solving robot's main power source is lithium-ion batteries, which provide the necessary electricity to run the motors and other electronic parts.
- **Rechargeable:** Lithium-ion batteries have the ability to be recharged after several cycles of operation. With the help of this feature, the robot will always be able to run for longer stretches of time between battery changes.

Figure 4.6 Lithium-ion Batteries

4.7 Jumper wires:

Jumper wires are essential components for maze-solving robot since they establish electrical connections within the system. Here's how they align with the project requirements:

- **Electrical Connections:** Sensors, motors, the Arduino Mega 2560, and other electronic modules are connected electrically to one another using jumper wires in the maze-solving robot.
- **Conductivity:** To provide low resistance and dependable electrical connections, jumper wires are usually composed of conductive materials like copper or tinned copper. This conductivity is necessary to keep the circuit's voltage dips from occurring and to preserve signal integrity.

Figure 4.7 Jumper wires

4.8 PCB's (printed circuit boards)

A printed circuit board (PCB) can be an important part of our maze-solving robot project since it helps assemble and arrange the electrical connections between different electronic parts. This is how PCBs meet the project specifications:

- **Compactness:** PCBs allow for the compact arrangement of electronic components, reducing the overall size and footprint of the robot's electrical system. This is especially important for projects where space is limited, such as mobile robotics applications.

- **Professional Appearance:** When compared to prototypes built on breadboards or point-to-point wiring, PCBs provide a more ordered and professional appearance. It is advantageous to showcase the project and project a sense of professionalism and craftsmanship through its aesthetic attractiveness.

Figure 4.8 PCB's (Printed circuit board)

4.9 Connectivity:

Figure 4.9 describes about the circuit diagram of the project

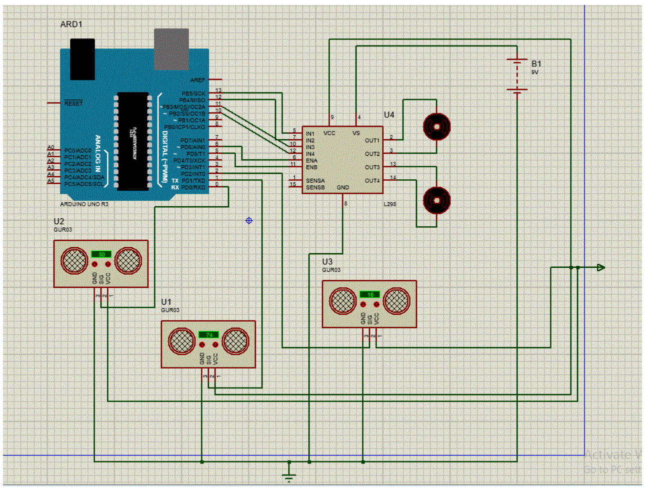

Figure 4.9 Circuit diagram

CAPTER 5: ALGORITHM

The flood fill algorithm is a technique used to fill connected regions of a grid-based map or image. In the context of your maze-solving robot project, the flood fill algorithm can be employed to find the shortest path from the robot's current position to the maze exit.

Here's how the flood fill algorithm could work for our project:

- **Map Representation**: A grid can be used to depict the maze's environment, with each cell representing a distinct location within the maze. Open passageways are represented as cells that can be traversed, while the maze's walls are represented as obstacles.
- **Initialization:** The robot's present location in the maze is designated as the beginning point for the flood fill procedure. Furthermore, the objective destination or maze exit is designated and labeled as the goal.
- **Flood Fill Process:** The robot's position serves as the starting point for the flood fill algorithm, which visits nearby cells to progressively investigate the maze. The program assesses neighbouring cells at each step to find the shortest path to the objective.
- **Traversal Rules:** Depending on how the maze is set up, the flood fill algorithm follows specific traversal rules. For instance, the robot can only move between neighbouring cells that are unobstructed by walls or other objects. The program also remembers which cells have been visited in order to prevent returning to the same spot repeatedly.
- **Path Finding:** As the flood fill algorithm progresses through the maze, it continuously updates the shortest path from the robot's current position to the goal. This process involves evaluating the distance or cost associated with reaching each neighboring cell and selecting the path with the minimum cost.
- **Optimization:** To optimize the pathfinding process, the flood fill algorithm can incorporate heuristics or optimization techniques such as A* (A-star) search. These techniques help prioritize exploration towards the goal while avoiding unnecessary detours or dead ends.
- **Dynamic Updates:** In a dynamic maze environment where obstacles or pathways may change over time, the flood fill algorithm can be adapted to accommodate updates in real-time. This flexibility allows the robot to adjust its pathfinding strategy based on the evolving maze conditions.
- **Path Execution:** The robot can follow this path step-by-step to navigate the maze on its own when the flood fill algorithm has finished its traversal and determined the shortest way from the robot's position to the maze exit.

The figure 6.1 Describes the architecture of the flood fill algorithm

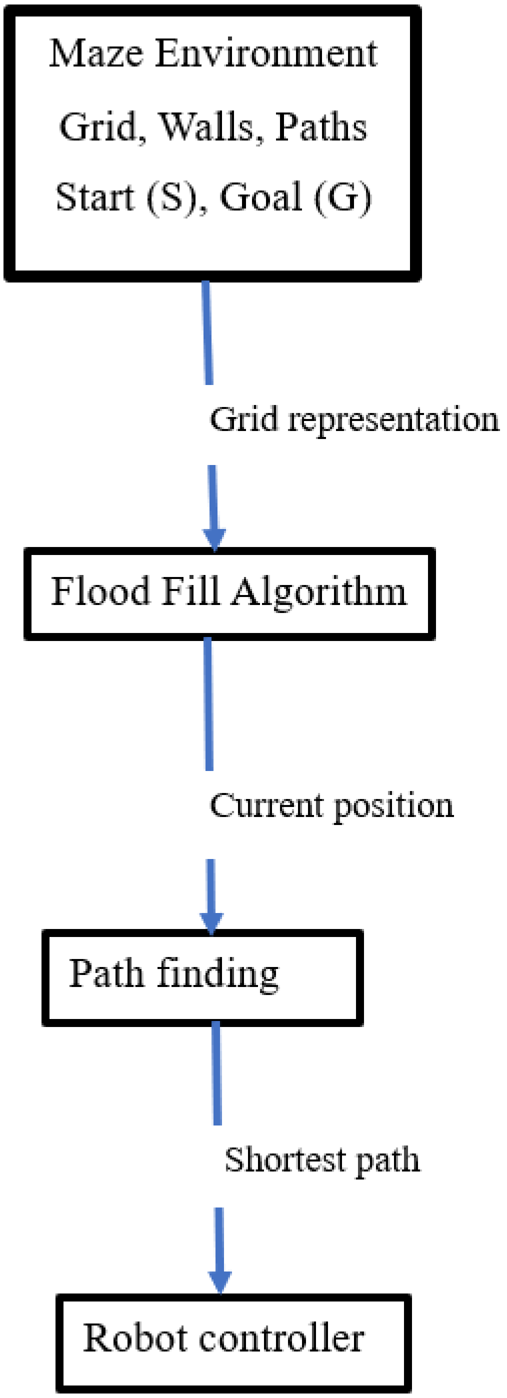

Figure 5.1 Architecture of Flood fill algorithm

CHAPTER 6: RESULTS AND DISCUSSIONS

6.1 Maze solving robot:

The figure 6.1 shows the view of robot developed using different sensors and microcontrollers.

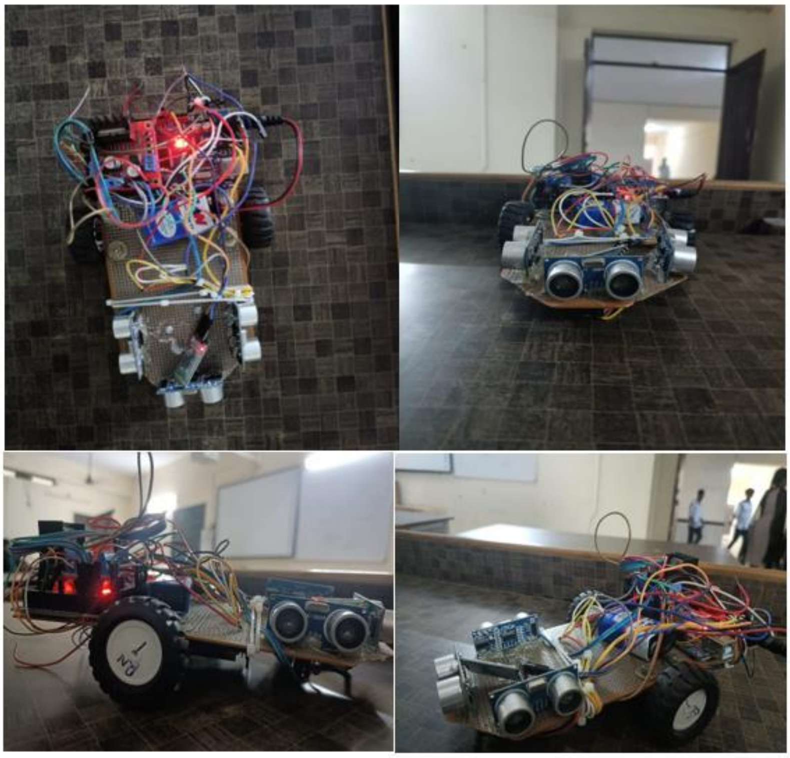

Figure 6.1 Our robot prototype

6.2 Robot in maze :

The figure 6.2 demonstrates how the robot runs in the maze.

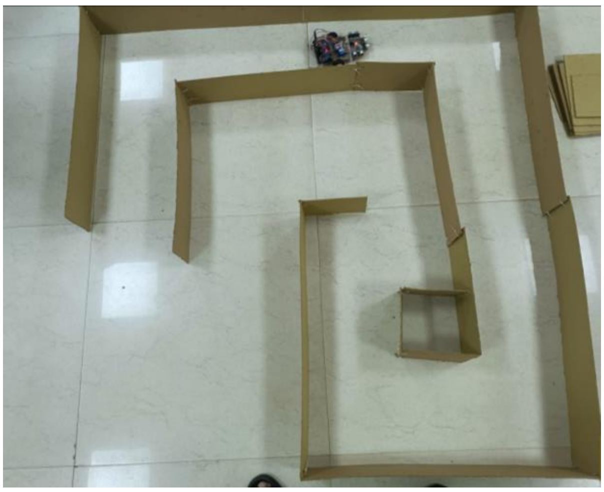

Figure 6.2 Robot running in maze

CONCLUSION

The goal of the maze-solving robot project is to enable autonomous navigation across a complex maze environment by integrating multiple components and algorithms. Considerable progress has been achieved in the design, implementation, and testing of every area of the robot's capabilities over the course of the project. The hardware used in the development of the maze-solving robot was carefully considered, including N20 geared motors with encoders for precise movement, ultrasonic sensors for obstacle detection, an Arduino Mega 2560 microcontroller for behavior control, and an HC-05 Bluetooth module for wireless communication with an Android app. These parts were picked with care to guarantee dependable functioning and strong performance when navigating the maze.

The flood fill method is one of the main algorithms used in the project and is essential to pathfinding. The flood fill algorithm allows for effective navigation across intricate maze configurations by methodically examining the maze and figuring out the shortest path from the robot's current position to the destination. Furthermore, the smooth execution of the computed path is guaranteed by the integration of the flood fill algorithm with the robot's control system.

As the project progressed, a number of obstacles were faced and effectively resolved. These difficulties included software debugging, algorithm optimization, and hardware integration problems. The maze-solving robot's performance was progressively enhanced through iterative testing and improvement, culminating in a reliable and effective navigation system.

To sum up, the maze-solving robot project is an important accomplishment in robotics engineering since it shows how to successfully combine hardware, software, and algorithms to build an autonomous navigation system that can successfully navigate through intricate maze environments. The research has potential for future developments in robotics technology and practical applications with more exploration and improvement.

REFERENCES

[6] R. Covaci, G. Harja, and I. Nascu, “Autonomous Maze Solving Robot ∗,” in 2020 22nd IEEE International Conference on Automation, Quality and Testing, Robotics - THETA, AQTR 2020 - Proceedings, 2020. doi: 10.1109/AQTR49680.2020.9129943.

[7] P. Studi, S. Teknik, F. Teknik, U. N. Surabaya, and M. S. Zuhri, “Maze Solver Robot Dengan Kendali PID Berbasis Arduino Uno Dengan Tunning Melalui Smartphone Android Arief Rahmadiansyah,” Teknik Elektro, vol. 08, 2019.

[8] A. Ma’arif, S. Iskandar, and Iswanto, “New design of line maze solving robot with speed controller and short path finder algorithm,” International Review of Automatic Control, vol. 12, no. 3, 2019, doi: 10.15866/ireaco.v12i3.16501.

[9] R. Kumar et al., “Maze Solving Robot with Automated Obstacle Avoidance,” in Procedia Computer Science, 2016. doi: 10.1016/j.procs.2017.01.192.

[10] I. R. Fahmi and D. J. Suroso, “A Simulation-Based Study of Maze-Solving-Robot Navigation for Educational Purposes,” Journal of Robotics and Control (JRC), vol. 3, no. 1, 2022, doi: 10.18196/jrc.v3i1.12241.

[11] D. Horváth, Z. Červeňanská, and R. Ďuriš, “A movement control of a small mobile 3-pi robot in a maze using artificial neural network,” International Journal of Mechatronics and Applied Mechanics, vol. 1, no. 8, 2020, doi: 10.17683/ijomam/issue8.10.