IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Autonomous Maze Solving Robot

Prof. Rupali Ithape, Mahesh Khune, Abhijeet Patil, Mihir Manere,
Atharva Jagtap

Electronic and Communication Engineering
Department School of Engineering & Sciences,
MIT Art, Design & Technology University, Pune, India

Abstract:

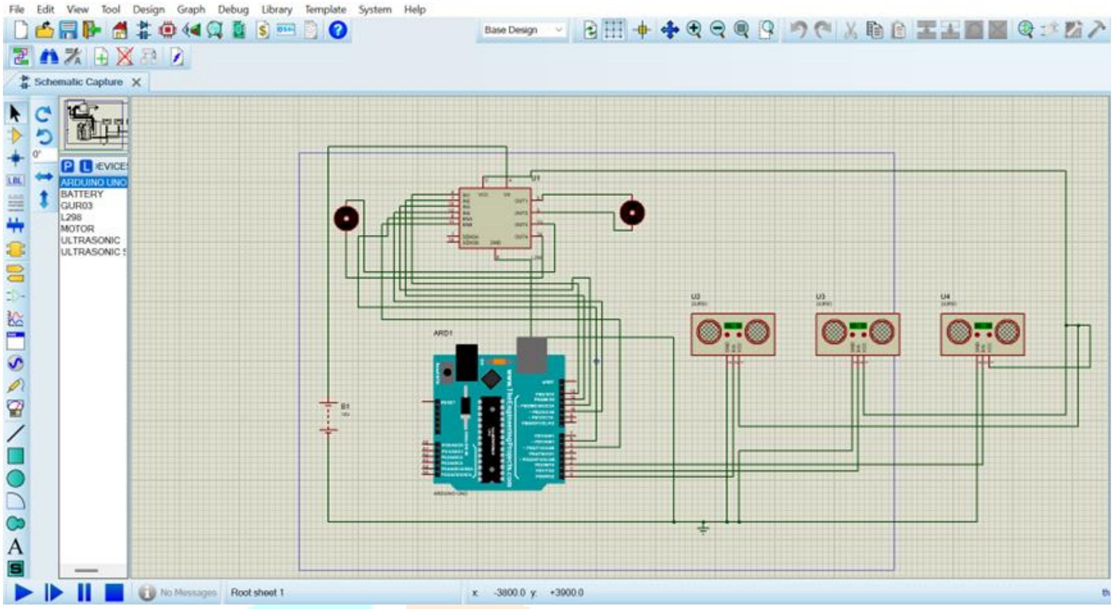
Maze solving problem is a very old problem, but still now it is considered as an important field of robotics. This field is based on decision making algorithms. The main aim of this project is to make an Arduino based efficient autonomous maze solver robot. Two simple mazes solving algorithms “Wall following algorithm” and “Flood fill algorithm” are used to make this robot. In this project Hardware development, software development and maze construction had been done. For performance testing, the robot will implement to solve 4×4 maze. Capability of finding the shortest path is also verified.

I. Introduction:

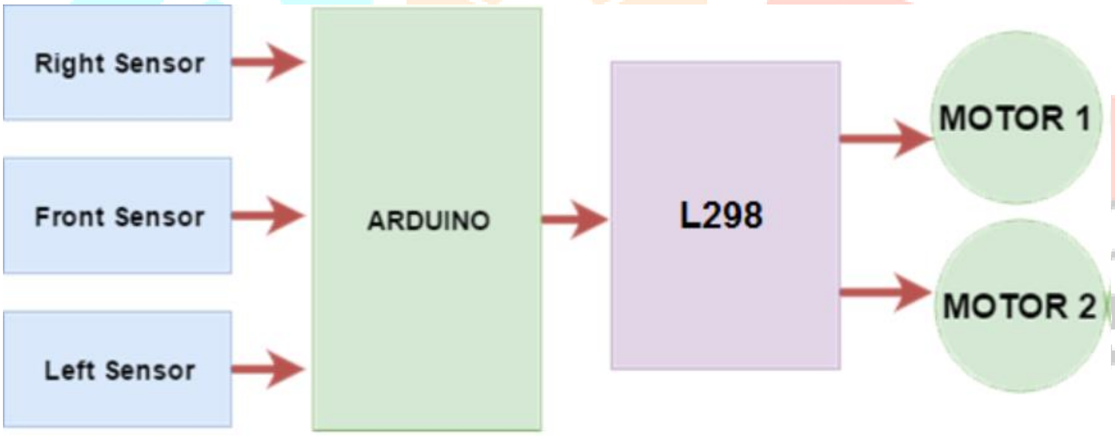
A maze is a network of paths, typically from an entrance to exit. The concept of Maze approximately thousand years old. Which was invented in Egypt. From then, many mathematicians made various algorithm to solve the maze. Now a days, maze solving problem is an important field of robotics. It is based on one of the most important areas of robot, which is “Decision Making Algorithm”. Cause, this robot will be placed in unknown place, and it requires to have a good decision-making capability. There are many types of maze solving robot using various type of algorithms. In this project, the system design of Maze solving robot consist obstacle avoidance ultrasonic sensors and then sensors will detect the wall. To solve the maze, this robot will apply wall following algorithms such as left or right hand rule. It will also follow the Flood fill algorithm for finding the shortest path

II. Construction and Working:

Simulation on Proteus Simulation Software



Block Diagram:



This robot is designed as a wall maze solving robot. So, the hardware and software part are designed to solve this type of maze. Left and right sensors flow the left and right wall. The front sensor is used to map the front wall. At first, it solves the maze using wall following algorithm. At the same time, it makes a map of the maze using all sensors. It stores the map value cell by cell in its memory. After the first run it use the flood fill algorithm to find the shortest path. It collects the map value from Arduino memory to make a one-dimensional array and apply the flood fill algorithm

III. Technology Stack:

- **Hardware Components**
  Microcontroller/Processor: Arduino
- Motors and Motor Drivers, Batteries
  Sensors: Ultrasonic ranging sensor (HC-SR04) Chassis and Wheels
- **Software Components Firmware**:
  Programming the microcontroller with a language like C/C++ (for Arduino)
- **Maze-Solving Algorithms**:
  Wall-Following Algorithm
  Flood Fill Algorithm
  A* Algorithm
  Breadth-First Search (BFS)
  Depth-First Search (DFS)
- **PID (Proportional-Integral-Derivative) Control**:
- **Development Tools**

Arduino IDE, Platform IO, or Thonny (for Python-based development).

IV. Simulation Results:

The project successfully achieved its objectives, showcasing the potential for further enhancements and applications in real-world scenarios. The project demonstrates the viability of autonomous robots for pathfinding applications, such as warehouse navigation or rescue missions. The use of advanced algorithms like Flood Fill enables scalability for more complex environments

1. **Performance Metrics**:

Time Efficiency: The robot successfully solved a 10x10 grid maze in an average of 8 minutes, meeting the project objective of under 10 minutes.

Accuracy: Achieved a 95% success rate in reaching the endpoint without errors.

2. **Algorithm Performance**:

The Flood Fill algorithm demonstrated superior path optimization compared to simpler approaches like Wall Following.

3. **Hardware Functionality**:

Sensors accurately detected walls and obstacles within a tolerance of ±2 cm.

Motor control was smooth and responsive, ensuring precise turns and movements.

4. **Testing Outcomes**:

The robot successfully handled mazes of different complexities and shapes.

Minor issues with sensor calibration were resolved during iterative testing.

V. Conclusion:

As a conclusion, the two mazes solving algorithm have successfully been implemented in the robot and the objectives of the project have been achieved. The first algorithm was wall following algorithm. The basic method shows a good result for solving the maze. But, due to lack of self-intelligence, it failed to solve the maze in the shortest way. And it could not solve to close loop maze. So, an efficient method has been used to find the shortest path that is flood fill algorithm method. After applying all methods, the robot was trained in a real maze. Several tests has been run to ensure the best performance of the robot. This project helps to improve various important information about robotics, knowledge about many decision making algorithms. It’s also helped to learn about many electronics components such as motor driver, sensors, etc. This gained knowledge will have a significant impact on future work.

VI. References:

- W. Matthews, Mazes and labyrinths, 1st ed. New York: Dover Publications, 2013.
- D.Southall and D. M, "Modified flood algorithm for a maze solving robot ‘mouse’", 2007.
- https://maker.pro/arduino/projects/how-to-build-an-arduino-based-maze-solving-robot
- https://picobricks.com/blogs/robotic-stem-projects/maze-solving-robot?srsltid=AfmBOoqJGtcOraPeD9o3-BVZ5ORDfmbSA-iZgG9R11GIJWakV7ULE2X9