NANO NAVIGATOR

A Theme Based Project Report submitted in partial fulfillment of the academic requirement for the award of the degree of

BACHELOR OF ENGINEERING

In

ELECTRONICS AND COMMUNICATION ENGINEERING

By

CH.NACHIKETH	1602-21-735-021
A.SRIMANIKRISHNA	1602-21-735-049
G.VINEEL	1602-21-735-061

Under the guidance of

Dr. K. Krishna Kishore
Associate Professor,
ECE

Department of Electronics and Communication Engineering

Vasavi College of Engineering (Autonomous)

ACCREDITED BY NAAC WITH 'A++' GRADE

IBRAHIMBAGH, HYDERABAD-500031

2021-2025

CERTIFICATE.

This is to certify that the Theme based project work titled "NANO NAVIGATOR " by

Nachiketh	1602-21-735-021
Srimanikrishna	1602-21-735-049
Vineel	1602-21-735-061

students of the Electronics and Communication Engineering Department, Vasavi College of Engineering in partial fulfillment of the requirement for the award of the degree of Bachelor of Engineering in Electronics and Communication Engineering is a record of the bonafide work carried out by them during the academic year 2023-2024. The result embodied in this Theme based project report has not been submitted to any other university or institute for the award of any degree

Internal Guide
Dr. K. Krishna Kishore
Associate Professor
E.C.E Department

Head of the Department
Dr. E.SREENIVASA RAO
Professor & HOD
E.C.E Department

External Examiner

DECLARATION

This is to state that the work presented in this Theme based project report titled "NANO NAVIGATOR" is a record of work done by us in the Department of Electronics and Communication Engineering, Vasavi College of Engineering, Hyderabad. No part of the thesis is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred to in the text. The report is based on the project work done entirely by us and not copied from any other source. I hereby declare that the matter embedded in this thesis has not been submitted by me in full or partial thereof for the award of any degree/diploma of any other institution or university previously.

Signature of the students

Nachiketh	1602-21-735-021
Srimanikrishna	1602-21-735-049
Vineel	1602-21-735-061

ACKNOWLEDGEMENTS

This satisfaction and euphoria that accompany the successful completion of any task would be incomplete without mentioning the people whose constant guidance and encouragement made it possible. We take pleasure in presenting before you, our project, which is the result of a studied blend of both research and knowledge.

It is our privilege to express our earnest gratitude and venerable regards to our internal guide Dr. Krishna Kishore. K, Associate Professor, E.C.E. Department, Vasavi College of Engineering, Ibrahim Bagh, for abounding and able guidance during the preparation and execution of the project work. We are grateful for his cooperation and his valuable suggestions.

We record with pleasure our deep sense of gratitude to Dr. E. SREENIVASA RAO, Head of the Department, E.C.E. for his stimulating guidance and profuse assistance we have received, which helped throughout the project.

Our sincere thanks to the Principal and Management, of Vasavi College of Engineering, Hyderabad for providing all the facilities in carrying out the project successfully.

Also, we acknowledge with thanks for the support extended by all the staff members and technical staff in shaping our project. We are thankful to one and all who co-operated with us during our project work.

Nachiketh	1602-21-735-021
Srimanikrishna	1602-21-735-049
Vineel	1602-21-735-061

BACHELOR OF ENGINEERING 1

1.ABSTRACT 7

2.INTRODUCTION 8

2.1. General description 8

2.2. Objective 8

2.3. Block Diagram 10

2.4. Organization of the thesis 11

3. LITERATURE SURVEY 13

3.1. Introduction 13

3.2. Over-view of the project 13

3.3. Previous research and studies : 14

3.4. Design and methodologies 15

3.5. Applications and case-studies 16

3.6. Gap analysis 16

4.Hardware Description: 17

4.0. Constructed maze solver: 17

4.1. Arduino 18

4.2. L298 Motor driver 19

4.3 Motors: 20

4.4 Batteries: 21

5. METHODOLGY 21

5.1. Work flow : 21

5.2. Project Overview: 24

5.3Robot Implementation: 25

6. DESIGN 33

6.1. Designing of Robot **33**

6.1.1Chassis and Construction: **34**

6.1.2. Sensors: **34**

6.1.3. Power System: **34**

6.1.4. Control System: **35**

6.2. 3-D Print of Chasis design: **35**

7. RESULTS 37

8. CONCLUSION **38**

9. REFERENCES 41

1.ABSTRACT

This project centers on creating a small, self-sufficient micro-mouse robot using an Arduino microcontroller, tailored to efficiently navigate complex mazes with minimal collisions. The key objective is to integrate sensors like ultrasonic and IR sensors to provide real-time data on the maze environment. The micro-mouse, standing compact at a height of 5 units, aims to intelligently map its surroundings and make optimal decisions to swiftly traverse the maze.

The proposed solution involves sensor systems that empower the robot to gather data crucial for navigating based on its current position and nearby obstacles. Algorithmic intelligence plays a pivotal role, employing a blend of path planning and optimization techniques. These algorithms are designed to enhance the micro-mouse's adaptability across different maze layouts over time, ensuring it can make efficient decisions in diverse scenarios.

Utilizing Arduino as the microcontroller offers a robust and versatile platform capable of executing the sophisticated algorithms necessary for effective maze navigation. This integration not only supports the real-time decision-making process but also enables continuous refinement of the micro-mouse's navigation strategy through iterative learning and adaptation.

Keywords: Micro-mouse robot, Arduino microcontroller,Autonomous navigation, Maze solving, Obstacle avoidance,Sensor integration (ultrasonic, IR sensors),Real-time data, Algorithmic intelligence,Path planning,Decision-making algorithms,Arduino IDE.

2.INTRODUCTION

2.1. General description

The project focuses on developing an autonomous micro-mouse robot utilizing an Arduino microcontroller. Designed to navigate complex mazes efficiently and minimize collisions, the robot integrates ultrasonic and IR sensors for real-time environmental data collection. Standing at a compact height of 5 units, the micro-mouse employs algorithmic intelligence to map its surroundings, make optimal navigation decisions, and adapt to varying maze configurations. The Arduino platform supports the execution of sophisticated algorithms necessary for effective maze traversal, ensuring robust performance and continuous improvement through iterative learning and adaptation.

2.2. Objective

The Nano-navigator's primary goal is to autonomously and efficiently navigate through complex maze environments. To achieve this, it must employ advanced algorithms that can identify the shortest and most optimal routes, minimizing both traversal time and energy consumption. The robot's design should incorporate real-time decision-making capabilities, allowing it to learn and adapt to its surroundings as it navigates the maze. Speed and efficiency are critical, ensuring the robot moves swiftly while maintaining consistent performance.

Moreover, the Nano-navigator needs to be robust and reliable, capable of handling various environmental conditions and unexpected challenges within the maze. Effective sensor integration is essential for gathering the necessary data for navigation. Ultimately, the robot should demonstrate competitive performance, showcasing its ability to navigate mazes more effectively than other similar robots. This combination of intelligent pathfinding, real-time adaptation, and robustness will enable the Nano-navigator to excel in complex maze environments.

Maze navigation: The Nano-navigator project aims to create a miniature robot capable of autonomously navigating through intricate maze environments with high efficiency and precision. By leveraging advanced sensor technology and intelligent algorithms, the robot will autonomously explore and maneuver through complex maze structures, aiming to reach its destination swiftly and accurately.

Path Finding: Central to the Nano-navigator's functionality is its ability to employ sophisticated algorithms for pathfinding. These algorithms will analyze maze layouts in real-time, calculating the shortest and most optimal routes to achieve minimal traversal time and energy consumption. By dynamically adjusting its path based on current sensor data, the robot will continuously optimize its navigation strategy for maximum efficiency.

Real-time Decision Making: The Nano-navigator will integrate real-time decision-making capabilities to assess and respond to immediate maze conditions. By processing sensor data instantaneously, the robot will make swift and informed decisions regarding movement, obstacle avoidance, and path adjustments, ensuring smooth navigation through dynamic maze environments.

Learning and Adaptation: Through iterative learning processes, the Nano-navigator will enhance its navigation capabilities over time. By accumulating data from previous maze runs and adjusting its algorithms accordingly, the robot will learn from experience to improve efficiency, accuracy, and adaptability to varying maze configurations and challenges.

Speed and Efficiency: Designed for competitive performance, the Nano-navigator prioritizes both speed and energy efficiency in maze navigation. By minimizing unnecessary movements and optimizing pathfinding algorithms, the robot aims to achieve rapid traversal of mazes while conserving energy, thereby extending operational endurance and performance longevity.

Robustness and Reliability: Engineered to withstand diverse environmental conditions and unforeseen obstacles within mazes, the Nano-navigator emphasizes robust design and reliability. Built with durable materials and equipped with resilient sensor systems, the robot will maintain consistent performance throughout its missions, ensuring reliable operation in challenging maze environments.

Sensor Integration: The Nano-navigator integrates advanced sensor technologies such as ultrasonic and IR sensors to gather comprehensive data about its surroundings. These sensors enable precise detection of maze walls, obstacles, and other environmental features, providing critical inputs for accurate navigation and real-time decision-making.

Competitive Performance: With a focus on achieving competitive performance standards, the Nano-navigator aims to excel in maze navigation challenges. Through continuous refinement of algorithms and hardware capabilities, the robot strives to outperform competitors by demonstrating superior speed, efficiency, accuracy, and reliability in maze-solving tasks.

By prioritizing these key aspects, the Nano-navigator project aims to push the boundaries of autonomous robotics in maze navigation, showcasing advanced capabilities in pathfinding, real-time decision-making, learning, and competitive performance..

2.3. Block Diagram

2.4. Organization of the thesis

This review is based on the classification of object behaviour system in which tries to describes in following ways:

Title Page: This page includes the title of the thesis, the author's name, the academic institution, and other relevant details.

Table of Contents: This section lists all the chapters, sections, subsections, and other major divisions of the thesis along with their page numbers. It helps readers navigate through the document and locate specific information.

Abstract: The abstract provides a concise summary of the thesis, highlighting the research problem, objectives, methodology, key findings, and conclusions. It gives readers an overview of the entire work in a condensed form.

Introduction: The introduction sets the stage for the research by providing background information, stating the research problem or question, and outlining the objectives and significance of the study. It also introduces the structure and organization of the thesis.

Literature Review: This chapter reviews relevant literature and studies related to the research topic. It provides a comprehensive overview of existing knowledge, identifies gaps or limitations in previous research, and establishes the context for the current study.

Methodology: The methodology section describes the research design, collection methods, and data analysis techniques employed in the study. It explains how the research was conducted and justifies the chosen approach.

Design: The design part describes about the methods got implemented, it includes the designing of scenario, labeling the data and implementing the machine learning algorithm on it.

Results: This chapter presents the findings of the research, usually through a combination of text, tables, and figures. It provides a clear and detailed description of the analyzed data and any statistical or qualitative analysis performed.

Discussion: The discussion section interprets and analyzes the results in relation to the research objectives. It explores the implications of the findings, discusses any limitations or challenges encountered, and relates the results to the existing literature.

Conclusion: The conclusion summarizes the main findings of the study, restates the research objectives, and presents the overall implications and significance of the research. It may also suggest areas for further research or propose practical applications of the findings.

References: This section lists all the sources cited in the thesis, following a specific citation style (e.g., APA, MLA). It provides the necessary information for readers to locate and verify the referenced materials.

3. LITERATURE SURVEY

3.1. Introduction

Navigating through mazes has long been a quintessential challenge in robotics, reflecting advancements in autonomy and navigation technologies. The "Nanonavigator" project aims to tackle this challenge by developing a maze-solving robot equipped with state-of-the-art nanotechnology and artificial intelligence. This literature survey delves into existing research and developments in maze-solving robots, focusing on methodologies, algorithms, and technologies employed to achieve efficient navigation in complex environments. By synthesizing insights from current literature, this survey aims to identify gaps and opportunities for innovation in enhancing the Nanonavigator's capabilities. Through this exploration, we seek to contribute to the evolution of autonomous robotic systems and pave the way for future advancements in maze-solving robotics.

3.2. Over-view of the project

The "Nanonavigator" project focuses on developing a sophisticated micro mouse maze-solving robot using Arduino microcontrollers and IR sensors. Inspired by the challenges posed by maze navigation, this project integrates advanced robotics principles with practical applications of sensor technology.

Key components and features of the project include:

1. **Arduino Microcontroller:** The heart of the Nanonavigator, Arduino facilitates real-time control and coordination of the robot's movements and sensor data processing.

2. **IR Sensors:** Utilizing infrared sensors for detecting walls and obstacles within the maze, enabling the robot to make informed navigation decisions.
3. **Path Planning Algorithms:** Implementing efficient algorithms such as flood fill, A* (A-star),First left algorithm or other maze-solving algorithms to compute optimal paths from the starting point to the destination within the maze.
4. **Compact Design:** Designed to be compact yet efficient, the Nanonavigator incorporates a streamlined chassis and lightweight components to navigate tight corners and narrow passages typical of maze environments.
5. **Learning and Optimization:** Incorporating adaptive learning capabilities to enhance navigation efficiency over successive maze runs, utilizing iterative improvement strategies.

By leveraging these components and technologies, the Nanonavigator aims to demonstrate robust maze-solving capabilities within a miniature robotic platform. This project not only explores the practical application of Arduino and IR sensors in robotics but also serves as a learning platform for understanding autonomous navigation in constrained environments.

3.3. Previous research and studies :

1. **"Development of a Micro Mouse Robot Using Arduino and IR Sensors"**
   - This study focused on the development of a micro mouse robot similar to the Nanonavigator project. It utilized Arduino microcontrollers and IR sensors for maze navigation. The research highlighted the implementation of path planning algorithms and the integration of sensor data to achieve efficient maze-solving capabilities in a compact robot.
2. **"Advanced Maze Solving Algorithms for Autonomous Robots"**
   - This research explored various advanced maze-solving algorithms applicable to autonomous robots. It reviewed algorithms such as Dijkstra's algorithm, A* (A-star), and swarm intelligence techniques like Ant Colony Optimization (ACO) for optimizing pathfinding in maze environments. The study

aimed to identify the most effective algorithm for real-time navigation in complex maze structures.

3. **"Integration of Nanotechnology in Robotics: Challenges and Opportunities"**
    - This study examined the integration of nanotechnology in robotics, focusing on challenges and opportunities. It discussed the use of nanoscale sensors and actuators to enhance robot performance, including agility, precision, and energy efficiency. The research aimed to pave the way for future advancements in miniaturized robotic systems like the Nanonavigator.
4. **"Application of Machine Learning in Maze Solving Robots"**
    - This research investigated the application of machine learning techniques in improving the performance of maze-solving robots. It explored supervised and reinforcement learning approaches to optimize path planning, adapt to dynamic maze environments, and enhance decision-making capabilities based on sensor feedback. The study aimed to demonstrate how AI can enhance the autonomy and efficiency of robotic systems in navigating challenging mazes.

3.4. Design and methodologies

The "Nanonavigator" project integrates the First Left algorithm with Arduino microcontrollers and IR sensors, focusing on robust maze-solving capabilities.

Design: The robot features a compact chassis optimized for agility in maze environments, incorporating durable motors and lightweight materials. Arduino microcontrollers serve as the central control unit, managing motor movements and processing IR sensor data for efficient navigation. IR sensors are strategically placed to detect maze boundaries and obstacles, essential for accurate path planning.

Methodologies: Prototyping begins with iterative design refinements and simulations to validate algorithmic performance and sensor integration. Performance assessments systematically evaluate navigation accuracy and efficiency across maze configurations, informing adjustments to algorithm parameters and sensor placements. Documentation tracks design decisions and experimental outcomes, facilitating iterative improvements in maze-solving capabilities.

3.5. Applications and case-studies

This project demonstrates practical applications and potential in various contexts:

- **Educational Tool:** Enables hands-on learning of robotics principles, algorithm implementation, and sensor integration in maze environments.
- **Research Advancement:** Contributes to autonomous systems research by validating effective maze-solving strategies and sensor-driven navigation.
- **Prototype Development:** Provides a foundational platform for developing autonomous systems applicable in industrial automation and exploration scenarios.

3.6. Gap analysis

Identifying avenues for future enhancements and innovations in maze-solving robotics includes:

- **Algorithmic Optimization:** Refining the First Left algorithm to enhance path efficiency and adaptability across diverse maze complexities.
- **Sensor Fusion Techniques:** Exploring advanced sensor fusion methods to improve maze perception and decision-making under dynamic conditions.
- **Machine Learning Integration:** Incorporating machine learning to enable adaptive learning and decision-making based on real-time sensor feedback and prior maze-solving experiences.

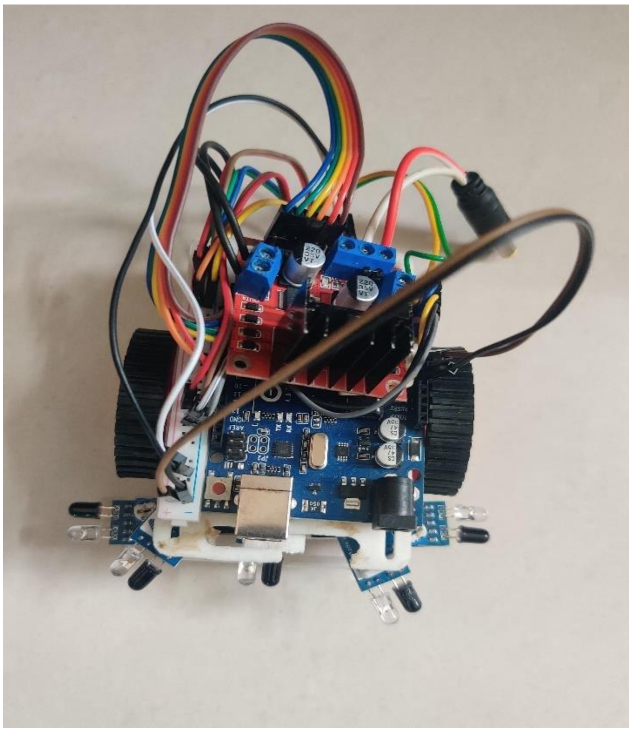
- **Miniaturization and Energy Efficiency:** Addressing challenges in miniaturizing components and optimizing power consumption to extend operational endurance in compact robotic platforms.

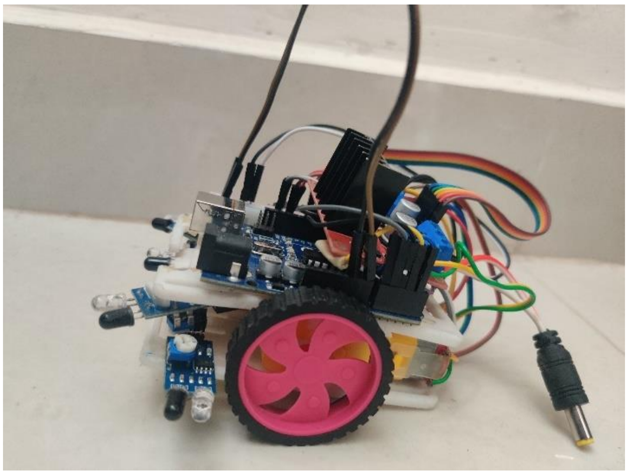
Addressing these gaps aims to elevate autonomous maze-solving capabilities, advancing robotics applications and paving the way for innovative uses in navigation and exploration tasks.

4.Hardware Description:

This chapter covers the important information about all hardware which is used in this project. It is going to give some details for each component including features, specifications and how it operates.

4.0. Constructed maze solver:



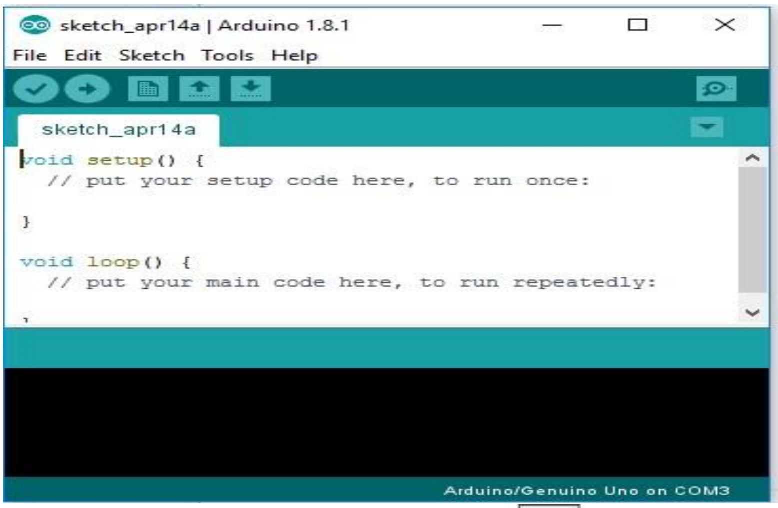


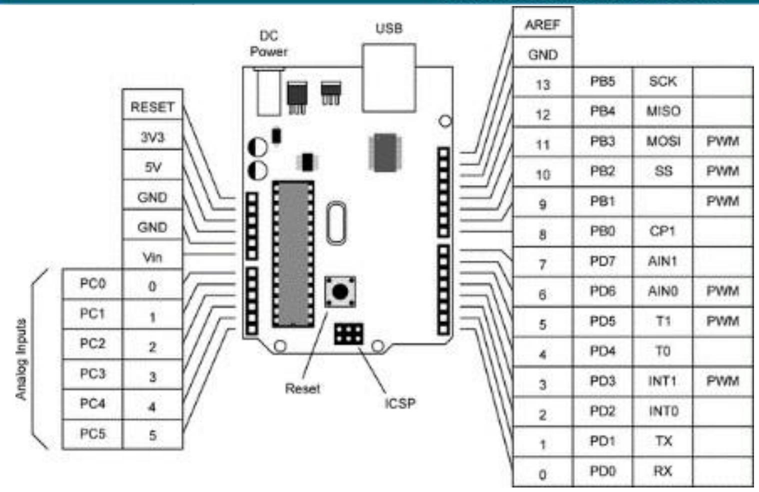
4.1. Arduino

The heart of this project is Arduino. All program of this project is stored in its microprocessor. Arduino is an open source hardware development board. Arduino hardware consists of an open hardware design with an Atmel AVR processor. Arduino programming language is used to program the processor. There are many types of Arduino board available in the market. But, in this project Arduino UNO has been used.

Figure 4.1:Arduino UNO

Arduino UNO is based on the ATmega328P microprocessor. It has 14 input/output pins. 6 digital pin can be used as PWM outputs. It has 6 analog input pins. It has a 16 MHz quartz crystal. It contains USB connection port, dc power port. The microcontroller is programmed via Arduino Software (IDE). Figure 3.3 shows the pin mapping of Arduino UNO and Figure 3.4 shows the IDE of Arduino.





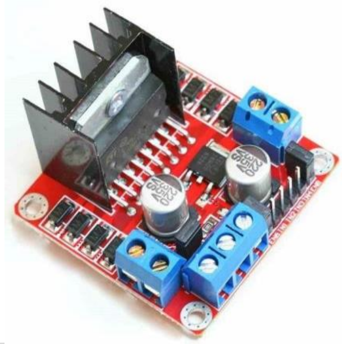
4.2. L298 Motor driver

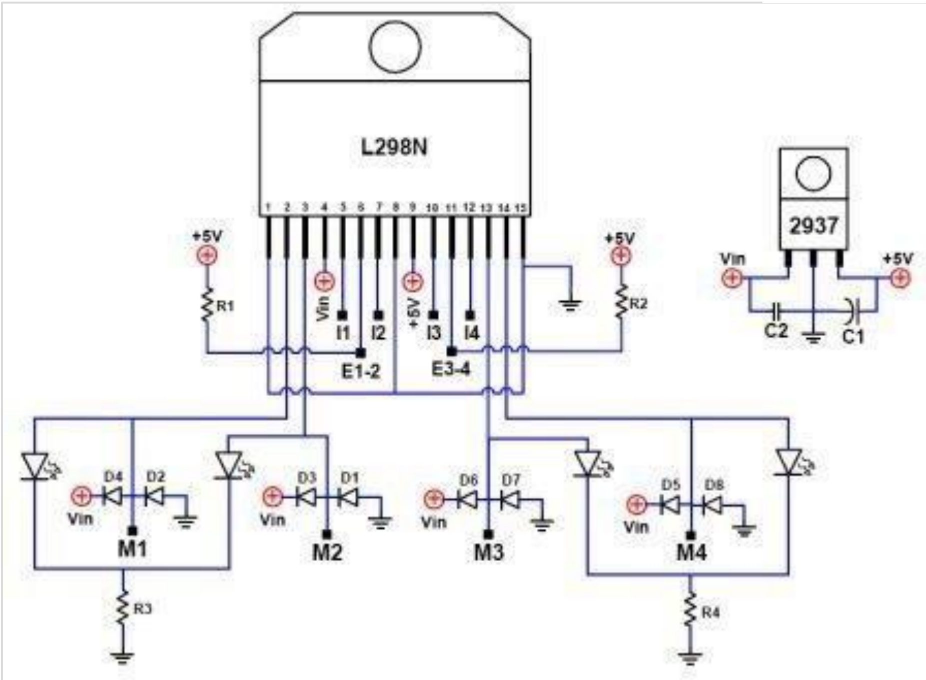
This dual bidirectional motor driver. Most popular L298 Dual H-Bridge Motor Driver Integrated Circuit is used here

Features:

- Logic voltage- 5V
- Drive voltage- 5V to 35V

- Logic Current- 0mA-36mA
- Drive current- 2A (MAX each bridge)
- Maximum Power – 25W





4.3 Motors:

Two DC gear motors have been used to drive this robot. This gear motor is ideal for robotic car or line-tracing robot.

FEATURES:

1. Operating voltage: 3V ~ 6V DC
2. Speed without load: 800g.cm
3. Maximum torque: 90±10rpm
4. Reduction ratio: 1:48
5. Load current: 190mA (max. 250mA)

Figure 4.3 : Dc Gear Motor with wheel

4.4 Batteries:

A battery have been used to power the motor driver. Each battery relates to other in series connection. One 9V battery is used to power the Arduino board, another is used to power the voltage regulator.

Figure 4.4: Battery

5. METHODOLGY

5.1. Work flow :

1. Project Initiation and Planning: The project "Nano Navigator" begins with a clear set of goals and objectives aimed at designing and constructing a micromouse robot capable of autonomously navigating through maze environments. Key considerations include defining the scope of the project, which involves specifying the complexity levels of mazes the robot will navigate and understanding any competition rules if participating in a competition setting. A detailed project timeline is established, outlining milestones for various phases such as design, construction, testing, and refinement. This planning phase ensures that all aspects of the project are systematically addressed, from initial concept to final implementation.

2. Design Phase: In the design phase, the focus is on conceptualizing and creating detailed plans for the Nano Navigator micromouse robot. Initial designs are sketched to visualize the layout and dimensions of the robot chassis, taking into account constraints such as the maximum height of 8cm and requirements for sensor placements, particularly IR sensors crucial for maze navigation. CAD (Computer-Aided Design) software like Fusion 360 or SolidWorks is employed to translate these concepts into precise 3D models. These models incorporate features necessary for mounting components such as the Arduino microcontroller, motor driver, batteries (e.g., two 9V and 4 AA), voltage regulator, and wheels/casters. The design phase aims to optimize the layout for efficient assembly and functionality while adhering to project specifications.

3. Prototyping and Fabrication: Prototyping and fabrication involve transforming the design plans into physical components ready for assembly. Utilizing FlashForge software, 3D printing is employed to fabricate the layers of the robot chassis and other structural parts using suitable filament materials that balance durability and weight considerations. Each layer, including the top layer housing electronics and sensors, the middle layer accommodating power sources and circuitry, and the bottom layer integrating motors and wheels with caster wheels for stability, is carefully printed and inspected for quality. Following successful printing, components are assembled according to the design specifications, ensuring secure attachment and proper alignment to facilitate optimal robot performance.

4. Software and Programming: The software and programming phase focuses on developing the intelligence and control mechanisms necessary for the Nano Navigator robot to navigate mazes autonomously. This involves implementing maze-solving algorithms such as the First Left algorithm, which prioritizes left turns at junctions to systematically explore and navigate maze paths. Programming tasks are carried out using Arduino IDE or a suitable programming environment, where code is written to manage motor movements, interpret sensor data from IR sensors for wall detection, and make navigation decisions based on real-time inputs. Extensive testing and debugging are conducted to refine the algorithm's accuracy and efficiency in maze navigation, ensuring the robot operates reliably under varying maze conditions.

5. Integration and Calibration: Integration and calibration involve merging hardware components with software systems to create a cohesive and functional robot. Sensor calibration is crucial to ensure IR sensors accurately detect maze walls and distances, while motor calibration fine-tunes parameters such as speed and direction for precise movement within maze corridors. System integration focuses on establishing seamless communication between sensors, motors, and the Arduino microcontroller, ensuring coordinated operation during maze navigation. This phase also includes testing the integrated system to verify functionality and performance, addressing any compatibility issues or operational discrepancies that arise.

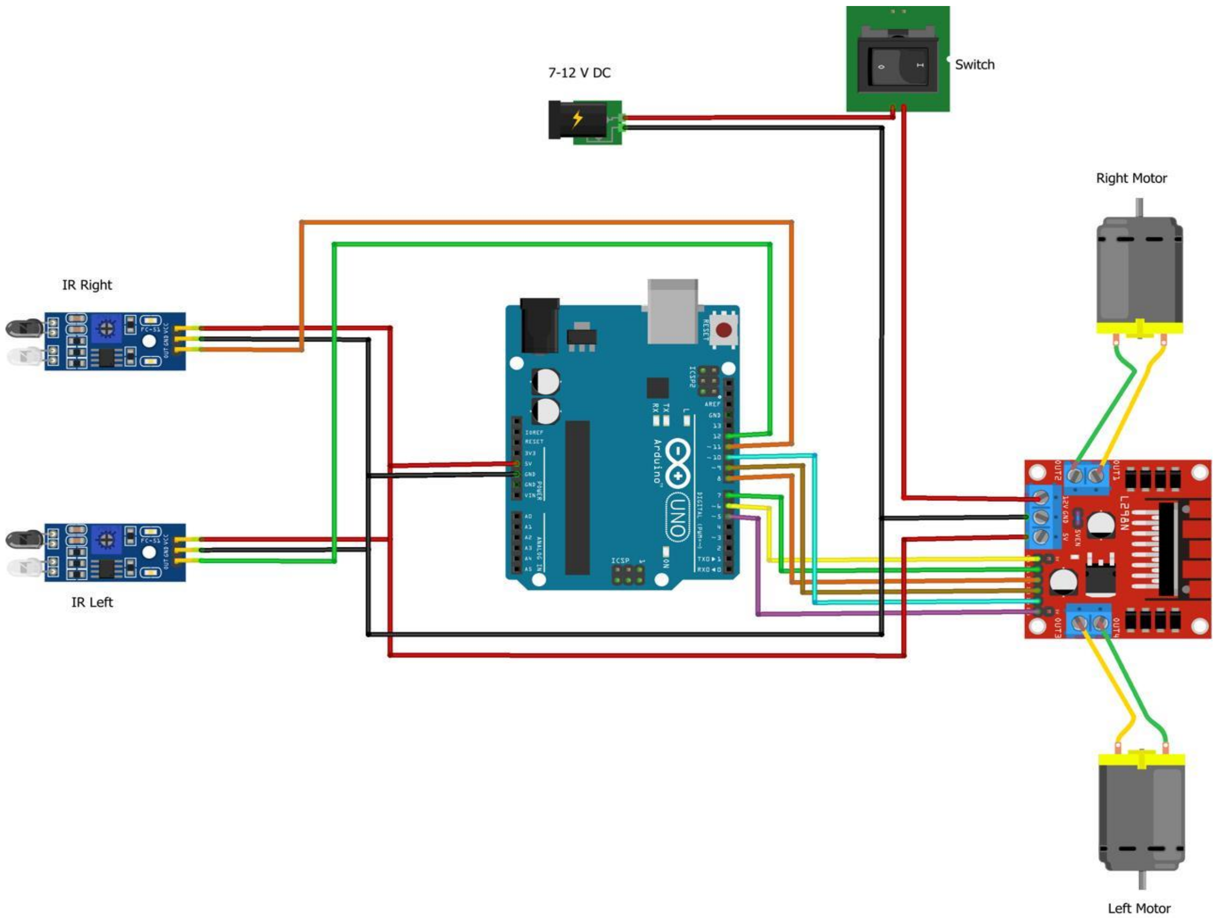
6. Testing and Optimization: Testing and optimization are iterative processes aimed at evaluating and enhancing the Nano Navigator robot's performance in maze-solving tasks. Performance testing involves assessing navigation speed, path efficiency, and successful maze completion rates across different maze configurations, from simple to complex layouts. Data collected during testing informs iterative improvements to algorithms, sensor placements, or mechanical design aspects aimed at enhancing overall performance and reliability. This iterative approach ensures the robot evolves to meet or exceed project goals, delivering robust performance in navigating mazes autonomously.

7. Documentation and Presentation: Documentation and presentation activities encompass capturing and communicating the project's design, development, and outcomes effectively. Technical documentation includes detailed specifications, CAD files, circuit diagrams, and annotated programming code compiled into a comprehensive project report. This report chronicles the design process, challenges encountered, solutions implemented, and test results obtained throughout the project lifecycle. Additionally, preparation for project demonstrations or competitions involves rehearsing presentations that showcase the Nano Navigator robot's capabilities, emphasizing its maze-solving algorithms, innovative design features, and demonstrated performance metrics to judges, stakeholders, or the broader audience.

5.2. Project Overview:

The hardware chosen to complete this project is Arduino based two-wheel mobile robot. Three ultrasonic sonar sensors have been used to map the maze and solve the wall maze.

5.2.1.BLOCK DIAGRAM:



5.3Robot Implementation:

5.3.1Software Development:

1.Algorithm:

The "First Left" algorithm is a straightforward yet effective approach used in maze-solving robotics, particularly in competitions like micromouse challenges. This method guides the robot by consistently favoring left turns at intersections or junctions within the maze. Beginning from a designated starting point, the robot navigates by continuously prioritizing the leftmost path whenever available, thereby systematically exploring the maze. This heuristic strategy simplifies decision-making at junctions: upon encountering a choice point, the robot checks for open pathways to the left first, followed by straight paths if necessary, and finally turns right only as a fallback. When faced with dead ends, the robot backtracks to the last junction where alternate routes are available, ensuring it can explore all possible paths towards the maze's goal. While straightforward, the First Left algorithm balances simplicity with efficiency, making it a popular choice for robots aiming to autonomously solve mazes with predictable wall configurations and structured paths.

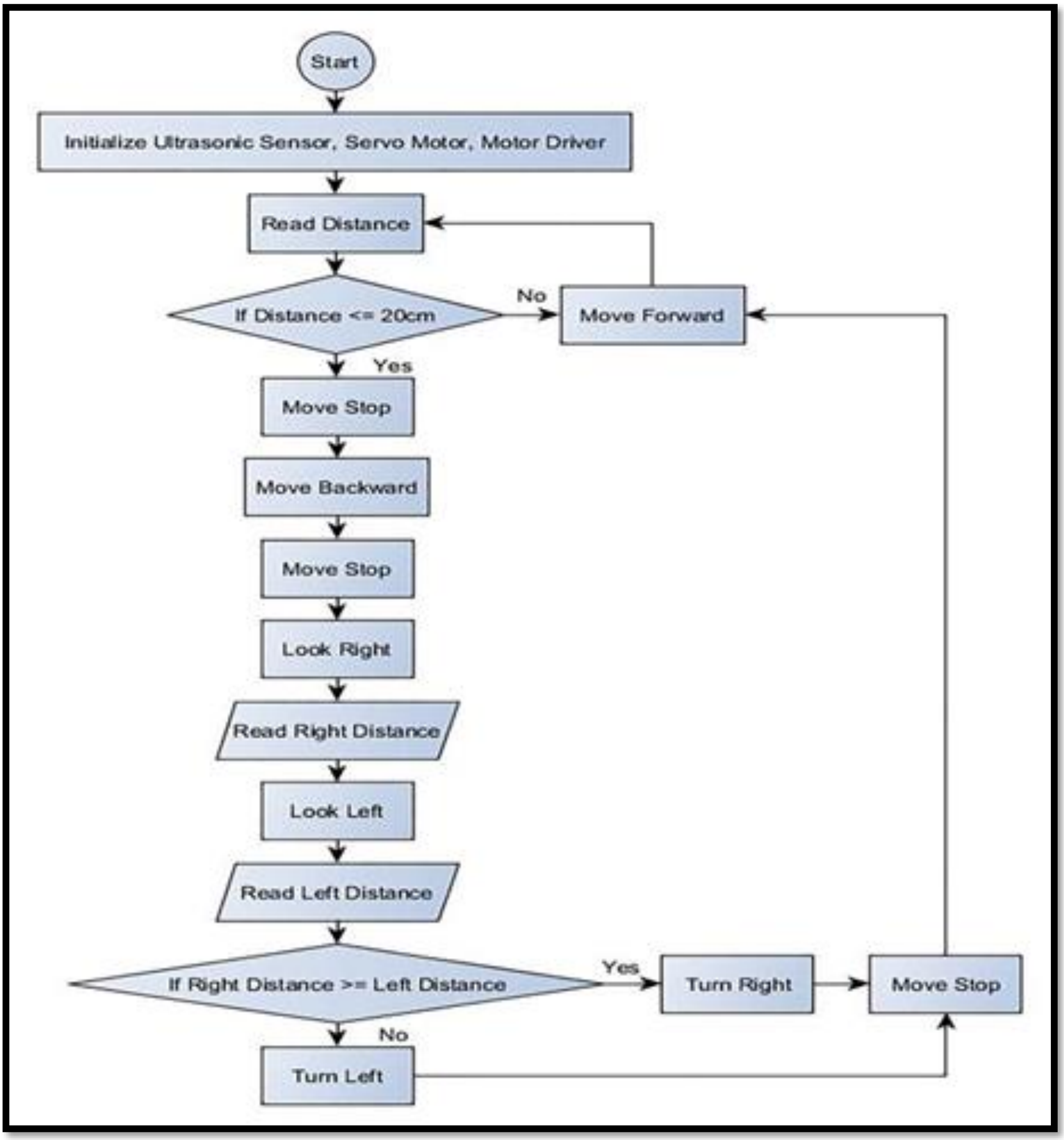
The "First Left" algorithm finds practical applications in various maze-solving robotics scenarios, including micromouse competitions and educational robotics projects. Its appeal lies in its simplicity and effectiveness in navigating structured mazes with predictable wall configurations. By consistently favoring left turns at junctions or intersections, the algorithm streamlines decision-making for the robot, reducing computational complexity and allowing for real-time adaptation to the maze's layout.

In terms of accuracy, the algorithm performs well in mazes where a left-hand wall-following strategy leads to efficient exploration and navigation. This method ensures that the robot systematically covers all accessible areas of the maze, minimizing the chances of missing paths or getting stuck in loops. It is particularly effective in environments with clear pathways and well-defined junctions, enabling the robot to reach the goal efficiently.

Efficiency is another key advantage of the First Left algorithm. Its heuristic approach means the robot makes decisions quickly based on immediate sensory inputs (such as IR sensors or cameras for wall detection), without requiring extensive memory or map planning. This efficiency is crucial in competitive settings where speed and accuracy are paramount. The algorithm's simplicity also makes it accessible for beginners in robotics programming, providing a solid foundation for understanding navigation strategies and maze-solving techniques.

While highly effective in certain maze types, the algorithm may encounter limitations in more complex mazes with irregular wall configurations or multiple interconnected pathways. In such cases, alternative algorithms that incorporate more sophisticated path planning or mapping techniques may be necessary to achieve optimal navigation outcomes. Nonetheless, for straightforward maze-solving tasks where left-hand turns lead to efficient exploration paths, the First Left algorithm remains a robust and practical choice.

2.Flow chart:



3.Flow chart Explaination:

According to the flowchart in figure,
the basic condition for movement of the car will be distance which will be sensed with the help of ultrasonic sensor. So we will be specifying the distance according to which, it will decide to move forward, backward, left, right or stop. The degree of freedom of the ultrasonic sensor is increased with the help of servo motor , since the ultrasonic sensor is

mounted on the servo motor . So, the ultrasonic sensor will also sense the distance to it’s left and right and change the path accordingly.

4.Code:

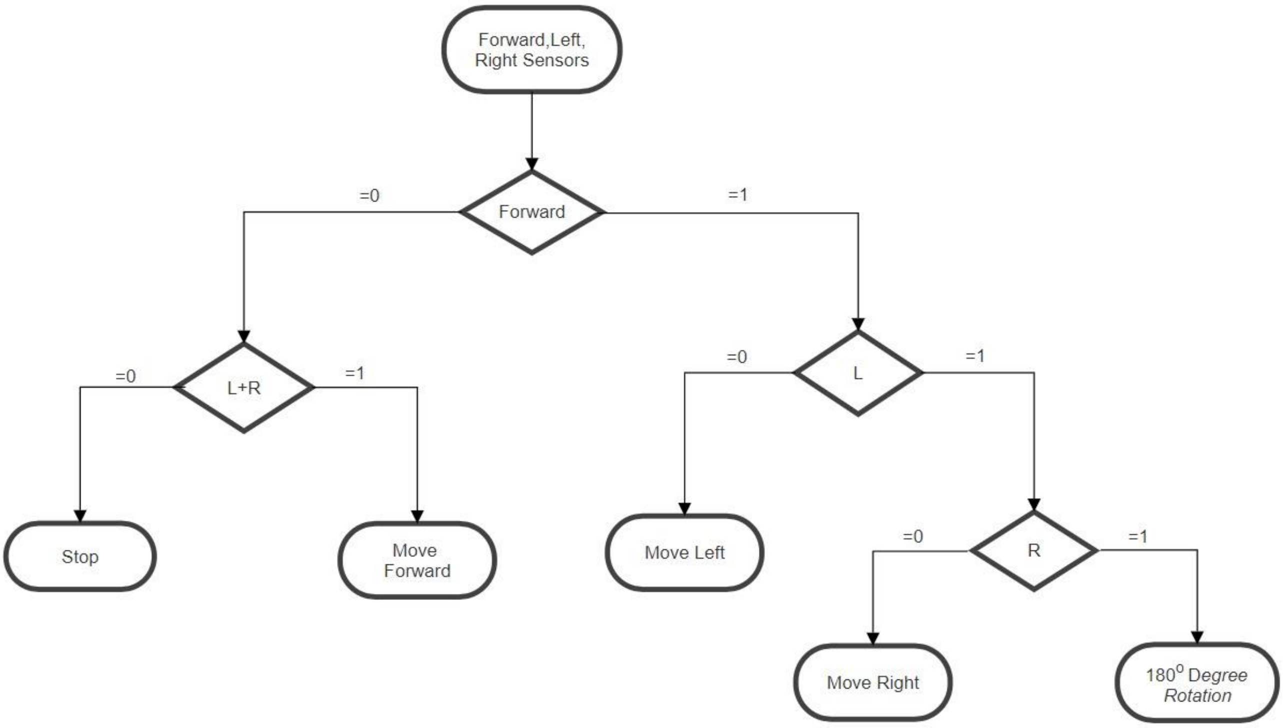
```
#define IR_SENSOR_RIGHT 2
#define IR_SENSOR_LEFT 12
#define IR_SENSOR_FRONT 13
#define MOTOR_SPEED 180

//Right motor
int enableRightMotor=6;
int rightMotorPin1=7;
int rightMotorPin2=8;

//Left motor
int enableLeftMotor=5;
int leftMotorPin1=9;
int leftMotorPin2=10;

void setup()
{
  //The problem with TT gear motors is that, at very low pwm value it does not even rotate.
  //If we increase the PWM value then it rotates faster and our robot is not controlled in that speed and goes out of line.
  //For that we need to increase the frequency of analogWrite.
  //Below line is important to change the frequency of PWM signal on pin D5 and D6
```

```
  //Because of this, motor runs in controlled manner (lower speed) at high
PWM value.
  //This sets frequecny as 7812.5 hz.
  TCCR0B = TCCR0B & B11111000 | B00000010 ;

  // put your setup code here, to run once:
  pinMode(enableRightMotor, OUTPUT);
  pinMode(rightMotorPin1, OUTPUT);
  pinMode(rightMotorPin2, OUTPUT);

  pinMode(enableLeftMotor, OUTPUT);
  pinMode(leftMotorPin1, OUTPUT);
  pinMode(leftMotorPin2, OUTPUT);

  pinMode(IR_SENSOR_RIGHT, INPUT);
  pinMode(IR_SENSOR_LEFT, INPUT);
  pinMode(IR_SENSOR_FRONT, INPUT);
  rotateMotor(0,0);
}

void loop()
{

  int rightIRSensorValue = digitalRead(IR_SENSOR_RIGHT);
  int leftIRSensorValue = digitalRead(IR_SENSOR_LEFT);
  int frontIRSensorValue = digitalRead(IR_SENSOR_FRONT);
```

```
//if(front=low && (right=high||right=low)&& (left=high||right=low))
//move = front(.,.);
if(frontIRSensorValue == LOW && (rightIRSensorValue == HIGH ||
rightIRSensorValue == LOW)&&(leftIRSensorValue == HIGH ||
leftIRSensorValue == LOW ))
 {
  rotateMotor(MOTOR_SPEED, MOTOR_SPEED);
 }

//else if ((front=high)&&(left=low)&&(right=low||right=high))
//move=left(.,~);
else if((frontIRSensorValue == HIGH) && (leftIRSensorValue == LOW
) && (rightIRSensorValue == HIGH || rightIRSensorValue == LOW))
{
  rotateMotor(MOTOR_SPEED, -MOTOR_SPEED);
}

//else if((front=high)&&(left=high)&&(right=low))
//move=right(~,.);
else if((frontIRSensorValue == HIGH) && (leftIRSensorValue == HIGH
) && (rightIRSensorValue == LOW ))
{
  rotateMotor(-MOTOR_SPEED, MOTOR_SPEED);
}

//else if(front=high&&left=high&&right=high)
//move=reverse(,);
```

```
  else if((frontIRSensorValue == HIGH) && (leftIRSensorValue == HIGH
) && (rightIRSensorValue == HIGH ))
  {
    rotateMotor(-MOTOR_SPEED, -MOTOR_SPEED);
  }

  //else
  //move=stop(0,0);
  else
  {
    rotateMotor(0, 0);
  }
}

void rotateMotor(int rightMotorSpeed, int leftMotorSpeed)
{

  if (rightMotorSpeed < 0)
  {
    digitalWrite(rightMotorPin1,LOW);
    digitalWrite(rightMotorPin2,HIGH);
  }
  else if (rightMotorSpeed > 0)
  {
    digitalWrite(rightMotorPin1,HIGH);
    digitalWrite(rightMotorPin2,LOW);
  }
```

```
  else
  {
    digitalWrite(rightMotorPin1,LOW);
    digitalWrite(rightMotorPin2,LOW);
  }

  if (leftMotorSpeed < 0)
  {
    digitalWrite(leftMotorPin1,LOW);
    digitalWrite(leftMotorPin2,HIGH);
  }
  else if (leftMotorSpeed > 0)
  {
    digitalWrite(leftMotorPin1,HIGH);
    digitalWrite(leftMotorPin2,LOW);
  }
  else
  {
    digitalWrite(leftMotorPin1,LOW);
    digitalWrite(leftMotorPin2,LOW);
  }
  analogWrite(enableRightMotor, abs(rightMotorSpeed));
  analogWrite(enableLeftMotor, abs(leftMotorSpeed));
}
```



6. DESIGN

6.1. Designing of Robot

The robot body (chassis) consists of three layers made from 3D printed material. On the top layer, the Arduino and motor driver circuit are mounted. Additionally, three IR sensors have been placed strategically. The power section is located in the middle part of the chassis, accommodating two 9V batteries, a 4 AA battery holder, and a 5V voltage regulator circuit.

At the bottom layer, two gear motors with wheels are securely attached, along with two caster wheels for stability and maneuverability. This configuration provides a sturdy and functional base for the robot's operation.

6.1.1Chassis and Construction:

- **Material:** Lightweight and durable materials suitable for 3D printing.
- **Dimensions:** Total height including sensors should not exceed 8cm, with a tolerance of 3cm.
- **Structure:** Three-layered design for modular assembly:
    - **Top Layer:** Houses Arduino microcontroller, motor driver circuit, and sensors.
    - **Middle Layer:** Contains power management components:
        - Two 9V batteries
        - 4 AA battery holder
        - 5V voltage regulator circuit
    - **Bottom Layer:** Mounts:
        - Two gear motors with wheels for propulsion
        - Two caster wheels for stability and maneuverability

6.1.2. Sensors:

- **Type:** Infrared (IR) sensors for wall detection.
- **Placement:**
    - Positioned on the top layer of the chassis within the permissible height limit of 5cm.
    - Three IR sensors strategically placed for effective wall sensing without violating maze rules.

6.1.3. Power System:

- **Configuration:** Dual power source for reliability:

- Two 9V batteries for motors and high-power consumption components.
- 4 AA batteries for microcontroller and sensors, regulated to 5V.

6.1.4. Control System:

- **Microcontroller:** Arduino platform for its versatility and community support.
- **Motor Driver:** Suitable motor driver circuit for controlling gear motors.
- **Programming:** Implement maze-solving algorithms to navigate efficiently and autonomously.

6.2. 3-D Print of Chasis design:

1. Design Concept:

- Start by creating a detailed 3D model of the micromouse chassis in FlashForge software.
- Sketch out the layout and dimensions based on competition rules, ensuring the chassis does not exceed the maximum height limit (8cm).

2. Modeling Tools:

- Utilize FlashForge's comprehensive modeling tools to design each layer of the chassis:
    - **Top Layer:** Create mounts and spaces for Arduino, motor driver, and IR sensors.
    - **Middle Layer:** Plan compartments for two 9V batteries, a 4 AA battery holder, voltage regulator, and wiring paths.
    - **Bottom Layer:** Design placements for gear motors, wheels, and caster wheels for stability.

3. Iterative Design Process:

- Iterate on the design within FlashForge to refine dimensions and placements:
    - Adjust component positions to ensure they fit securely within the specified height constraints.
    - Fine-tune sensor locations to optimize wall detection and navigation capabilities.

4. Material Selection:

- Choose filament materials compatible with 3D printing and suitable for structural integrity and weight considerations.
- Configure FlashForge settings for optimal print quality and adhesion between layers to ensure a durable chassis.

5. Printing Execution:

- Prepare FlashForge 3D printer with chosen filament and load the finalized chassis design.
- Monitor the printing process to address any issues such as warping or layer misalignment during fabrication.

6. Post-Processing:

- After printing, perform necessary finishing tasks such as removing support structures and smoothing rough edges to prepare for assembly.

7. RESULTS

The "Nanonavigator: A maze-solving robot" project promises significant advancements across several domains of robotics. By integrating sophisticated algorithms like A* and reinforcement learning, the robot aims to drastically improve navigation efficiency, reducing both traversal time and energy consumption in complex mazes. Enhanced sensor fusion and perception capabilities, such as lidar and depth cameras, will bolster reliability by enabling precise obstacle avoidance and error correction. Autonomous features like self-diagnosis and recharging stations will ensure prolonged operation without human intervention, boosting overall autonomy. Real-time mapping capabilities will enhance localization accuracy, allowing the robot to navigate confidently in diverse environments. Moreover, incorporating natural language processing for user interaction will enhance accessibility and usability, making the robot suitable for a wide range of applications. Beyond its practical applications, the project holds educational value, serving as a tool to teach robotics concepts and inspire future STEM professionals. Commercially, successful implementation could lead to applications in logistics, security, and entertainment industries, marking a significant stride in autonomous robotics technology. Overall, the Nanonavigator project not only promises technical advancements but also contributes to research and development in autonomous navigation, positioning it at the forefront of robotic innovation.

8. CONCLUSION

8.1. Conclusion

The project "Nano Navigator: A maze-solving micromouse robot" has successfully achieved its objectives through the implementation of two maze-solving algorithms: the initial wall-following algorithm and the subsequent flood fill algorithm. The wall-following algorithm demonstrated competence in navigating simple maze structures but faced challenges in handling complex layouts and closed-loop configurations. In response, the flood fill algorithm was introduced, significantly enhancing the robot's ability to identify and traverse the shortest paths within mazes of varying complexities.

Throughout the project, extensive testing and refinement were conducted to ensure the robot's reliability and performance in real-world maze scenarios. This iterative process not only validated the effectiveness of the algorithms but also contributed to improving the robot's overall navigation efficiency and accuracy.

Moreover, the project provided valuable hands-on experience in robotics, offering insights into decision-making algorithms, electronics integration (including motor drivers and sensors), and iterative development methodologies. These skills and knowledge gained are invaluable for future advancements in robotics and autonomous systems.

Both maze-solving algorithms have been successfully implemented in the robot, achieving the primary objectives of the project. The initial implementation of the wall-following algorithm demonstrated effective maze-solving capabilities but lacked adaptive intelligence, resulting in suboptimal paths and an inability to navigate closed-loop mazes efficiently. To address these limitations, the flood fill algorithm was subsequently introduced, significantly improving the robot's ability to find the shortest path through mazes, including those with complex configurations. Through rigorous testing in real maze environments, the robot's performance was refined and optimized.

This project has provided valuable insights into robotics and decision-making algorithms. It has deepened understanding of essential electronics components such as motor drivers and sensors, crucial for autonomous navigation. The acquired knowledge is expected to have a profound impact on future endeavors in robotics and automation.

In conclusion, the "Nano Navigator" project has not only demonstrated technical proficiency in maze-solving robotics but has also fostered a deeper understanding of interdisciplinary concepts crucial for innovation in the field of robotics and automation.

8.2 Future scope

- **Advanced Navigation Algorithms**: Integrate more sophisticated algorithms such as A* (A-star) or reinforcement learning-based approaches to improve navigation efficiency and adaptability in complex maze environments.
- **Multi-Robot Collaboration**: Develop capabilities for multiple Nanonavigators to collaborate and solve mazes together, potentially dividing tasks or coordinating exploration to optimize time and efficiency.
- **Sensor Fusion and Perception**: Enhance sensor capabilities by integrating more advanced sensor technologies like lidar or depth cameras for better perception of the maze environment,

enabling the robot to navigate with higher precision and reliability.

- **Autonomous Recharging and Maintenance**: Implement autonomous recharging stations where the Nanonavigator can dock itself when low on battery, ensuring continuous operation without human intervention. Additionally, design self-diagnostic features to detect and potentially repair minor issues autonomously.
- **Mapping and Localization**: Develop mapping and localization capabilities to create detailed maps of the maze environment in real-time, enabling the robot to remember paths and optimize future navigation strategies.
- **Human-Robot Interaction**: Introduce natural language processing (NLP) capabilities for seamless human-robot interaction, allowing users to give commands or receive status updates through voice commands or text-based interfaces.
- **Adaptability to Variable Environments**: Extend the Nanonavigator's capabilities to adapt to dynamic maze configurations or changes in the environment, such as moving obstacles or variable lighting conditions.
- **Integration with IoT and Cloud Services**: Connect the Nanonavigator to IoT platforms and cloud services for remote monitoring, data analytics, and fleet management, allowing for centralized control and monitoring of multiple robots in different locations.
- **Educational and Research Applications**: Explore educational uses by developing curriculum modules that utilize Nanonavigator for teaching robotics, algorithm design, and problem-solving skills. Additionally, the platform could be used for research in autonomous navigation and swarm robotics.
- **Commercial Applications and Scalability**: Investigate potential commercial applications such as in warehouse logistics, security patrolling, or even entertainment sectors,

aiming for scalability and robustness in different operational environments.

9. REFERENCES

[1] Mayur Bhangale, Gaurav Dhabade, Akshay Khairnar, Mamta Bhagat "Self Driving car to demonstrate real time obstacles and object detection" International Research Journal of Engineering and Technology Volume 03 Issue: 11 Nov-2016.

[2] Mohsin Raza "Autonomous Vehicles: Levels, Technologies, Impacts and Concerns" International Journal of Applied Engineering Research Volume 13, Number 16 (2018).

[3] Jianfeng Zhao, Bodong Liang, Quixia Chen "The key technology towards self-driving car" IJIUS 2017.

[4] Shahzeb Ali, Azam Rafique "Self-Driving and Driver Relaxing Vehicle" ICRAI November-2016

[5] Shubh Srivastava, Rajanish Singh "Voice Controlled Robot Car Using Arduino" International Research Journal of Engineering and Technology Volume: 07 Issue: 05 May 2020

[4] "Spectral lines: Announcing the Amazing Micro-Mouse Maze Contest - IEEE Xplore Document", *Ieeexplore.ieee.org*, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/6367601/. [Accessed: 12- Apr- 2017].

[5] MICHAEL GIMS, S. L. D. B. 1999. *Micromouse-Microprocessor Controlled Vehicle.* Bachelor of Engineering, University of East London.

[6] CHANG, Y. C. 2009. *Micromouse Maze Solving Robot.* Bachelor of Engineering, Universiti Teknologi Malaysia.

[7] A. Bakarsayutisaman and I. Abdramane, "Solving a Reconfigurable Maze using Hybrid Wall Follower Algorithm," *International Journal of Computer Applications*, vol. 82, no. 3, pp. 22–26, 2013.

[8] M.M Thu, N.N. Win. "Micromouse Maze Solving", *International Journal of Science, Engineering and Technology Research (IJSETR)* Volume 5, Issue 9, September 2016.

[9] M. Kazimierczuk, *Pulse-width modulated DC-DC power converters*, 1st ed. Chichester, West Sussex: Wiley, 2016.

[10] Arduino - PWM", *Arduino.cc*, 2017. [Online]. Available: https://www.arduino.cc/en/Tutorial/PWM. [Accessed: 13- Apr- 2017].

[11] M. McRoberts, *Beginning Arduino*, 1st ed. [Berkeley, Calif.]: Apress, 2013.

[12] R3, S. Edition, S. Edition, A. 5V/16MHz, A. 3.3V/8MHz, S. Arduino, S. (White), A.
R3, S. V3.2, S. 101, S. OpenLog and S. Module, "Arduino Uno - R3 - DEV-11021 - SparkFun Electronics", *Sparkfun.com*, 2017. [Online]. Available: https://www.sparkfun.com/products/11021. [Accessed: 13- Apr- 2017].

[13] "4. Arduino Technical Details [Book]", *Safari*, 2017. [Online]. Available: https://www.safaribooksonline.com/library/view/arduino-atechnical/9781491934319/ch04.html. [Accessed: 13- Apr- 2017].

[14] "ATmega328P - Microcontrollers and Processors", *Microchip.com*, 2017. [Online]. Available: http://www.microchip.com/wwwproducts/en/ATmega328P. [Accessed: 13- Apr- 2017].

[15] "Ultrasonic Sensor | What is an Ultrasonic Sensor?", *Education.rec.ri.cmu.edu*, 2017. [Online]. Available: http://education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic_sensor/1.html.
[Accessed: 13- Apr- 2017].

[16] "Ultrasonic Sensor - HC-SR04 - SEN-13959 - SparkFun Electronics", *Sparkfun.com*, 2017. [Online]. Available: https://www.sparkfun.com/products/13959. [Accessed: 13- Apr- 2017].

[17] Z. China, "Motor Driver 2A Dual L298 H-Bridge [L298N] - US $4.00 : HAOYU Electronics : Make Engineers Job Easier", *Hotmcu.com*, 2017. [Online]. Available: http://www.hotmcu.com/motor-driver-2a-dual-l298-hbridge-p-45.html. [Accessed: 14- Apr- 2017].

[18] "Bipolar Stepper Motor Driver Using L298", *Archiveclear.jimdo.com*, 2017. [Online]. Available: https://archiveclear.jimdo.com/2016/01/24/bipolar-stepper-motor-driverusing-l298/. [Accessed: 14- Apr- 2017].

[19] "LM7805 Datasheet, PDF - Alldatasheet", *Alldatasheet.com*, 2017. [Online]. Available: http://www.alldatasheet.com/view.jsp?Searchword=Lm7805. [Accessed: 14- Apr- 2017].