Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

Line Follower Robot

The domain of the Project

Embedded Systems and IoT

Mentor

MEHAK MAJEED
(Junior Engineer , ATFAAL Innovations)

By

Ms. DONTHI REDDY KAVYA
(B.Tech, 4th year pursuing)

Period of the project

July 2025 to August 2025

SURE ProEd

Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

Declaration

The project titled "**LINE FOLLOWER ROBOT**" has been mentored by **MEHAK MAJEED**, organised by SURE Trust, from March 2025 to August 2025, for the benefit of the educated unemployed rural youth for gaining hands-on experience in working on industry relevant projects that would take them closer to the prospective employer. I declare that to the best of my knowledge the members of the team mentioned below, have worked on it successfully and enhanced their practical knowledge in the domain.

Name	**Signature**
Ms.DONTHI REDDY KAVYA	D·Reddykavya

Mentor	**Signature**
MEHAK MAJEED (Junior Engineer—ATFAAL innovations)	

Seal & Signature

Prof. Radhakumari
Executive Director & Founder
SURE ProEd

Table of contents

Executive Summary

This project presents the design and implementation of a **Bluetooth-Controlled Line Follower Robot** using an **ESP32 microcontroller**, an **L298N motor driver**, and **infrared (IR) sensors**. The robot operates in two modes: manual activation through Bluetooth commands and autonomous line-following navigation.

The ESP32 acts as the central control unit, receiving user commands via Bluetooth and processing real-time sensor inputs to guide the robot. Two IR sensors detect the presence of a black line on a contrasting surface, while the ESP32 applies decision-making logic to control the direction of movement. The L298N motor driver enables bidirectional control of two DC motors, ensuring smooth navigation.

Key functionalities include:

- **Bluetooth Integration** – Remote ON/OFF control using a smartphone or paired device.
- **Line Detection –** Dual IR sensors for reliable tracking of black line paths.
- **Motor Control –** Precise forward, left, and right movements managed via the L298N driver.
- **Failsafe Mechanism –** Automatic stop when the robot loses the line.

This system demonstrates the integration of **IoT communication (Bluetooth)** with **embedded control systems** for robotics. It is low-cost, scalable, and ideal for educational, hobbyist, and automation applications, providing a practical foundation for intelligent robotics and autonomous navigation research.

Introduction

The field of robotics is one of the most rapidly growing areas in science and technology, with applications ranging from industrial automation to smart home systems and autonomous vehicles. A **line follower robot** is one of the simplest yet most effective examples of autonomous robotic systems. It uses sensors to detect and follow a predefined path, usually a black line on a white surface. Despite its simplicity, the line follower concept forms the basis for advanced navigation technologies used in **automated guided vehicles (AGVs)**, **warehouse robots**, and **delivery bots**.

This project focuses on the design and implementation of a **Bluetooth-Controlled Line Follower Robot**. The robot combines **autonomous navigation capabilities** with **wireless Bluetooth control**, making it more versatile than a conventional line follower. At the core of the system is the **ESP32 microcontroller**, which offers built-in Bluetooth, fast processing, and flexibility for real-time decision-making. The **L298N motor driver** is used to drive two DC motors, providing the required power and directional control, while **infrared (IR) sensors** serve as the robot's vision system to detect line patterns on the ground.

When operating in autonomous mode, the IR sensors continuously scan the surface below. If both sensors detect the black line, the robot moves forward. If the left sensor detects black while the right sensor sees white, the robot turns left. Conversely, if the right sensor detects black while the left sees white, the robot turns right. In the absence of a line, the robot halts automatically to avoid misalignment. This **real-time control logic** ensures accurate tracking of the path.

An additional feature of this project is **Bluetooth connectivity**, which allows the robot to be activated or deactivated remotely using a smartphone. This enhances flexibility by enabling the user to start or stop the robot at will, making it suitable for semi-autonomous applications.

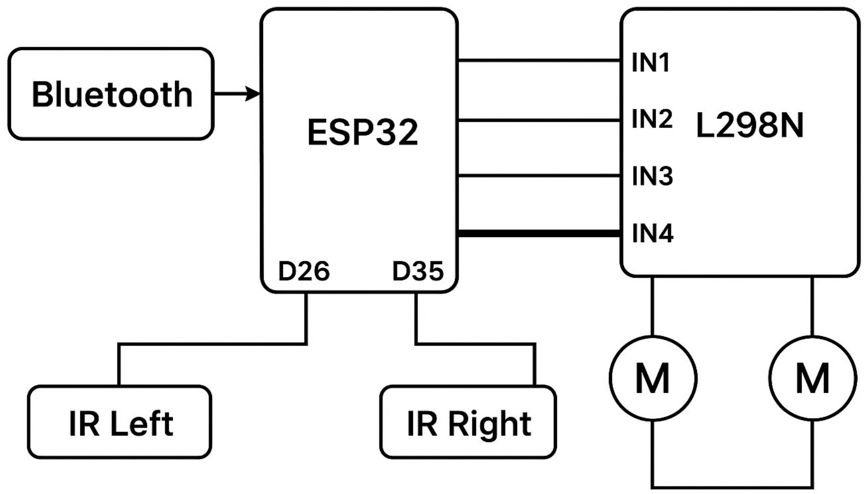
The significance of this project lies in its **practicality and educational value**. On one hand, it provides students and hobbyists with hands-on experience in robotics, sensor interfacing, and embedded programming. On the other hand, it demonstrates concepts directly applicable in **logistics automation**, where robots follow fixed paths to move goods, or in **service industries**, where guided robots are used for assistance.

In addition, the project emphasizes the importance of integrating **IoT technologies** with robotics. The use of Bluetooth not only simplifies wireless control but also demonstrates how microcontrollers like the ESP32 can bridge autonomous functions with user interactivity. This makes the system **scalable and adaptable** for future enhancements such as Wi-Fi integration, obstacle detection using ultrasonic sensors, or cloud-based monitoring.

In summary, this project represents a **low-cost, efficient, and versatile robotic platform** that highlights the fundamentals of embedded systems.

System Diagram

System Diagram



Mechanical Representation

The mechanical structure of the line follower robot consists of a rectangular chassis that serves as the base for mounting all components. Two DC motors are fixed on either side of the chassis, each connected to a wheel to provide differential drive. A caster wheel is placed at the front to support balance and smooth turning. At the front underside of the chassis, two IR sensors are mounted close to the ground to detect the black line path. A rechargeable battery pack is placed on the top of the chassis to supply power, while the ESP32 controller and L298N motor driver are also mounted on the chassis for compact integration.

Fig. 1.1: Mechanical Components of Line Follower Robot

Fig. 1.2: Assembled Mechanical Structure of Line Follower Robot

Electrical Representation

The electrical representation of the line follower robot is shown in below figure. The system consists of two IR sensors that act as inputs to the ESP32 microcontroller. Based on the sensor inputs, the ESP32 generates control signals which are sent to the L298N motor driver. The motor driver then controls the two DC motors to achieve forward, left, or right movement. A rechargeable 12V battery supplies power to the motor driver and ESP32, with regulated 5V used for the IR sensors.

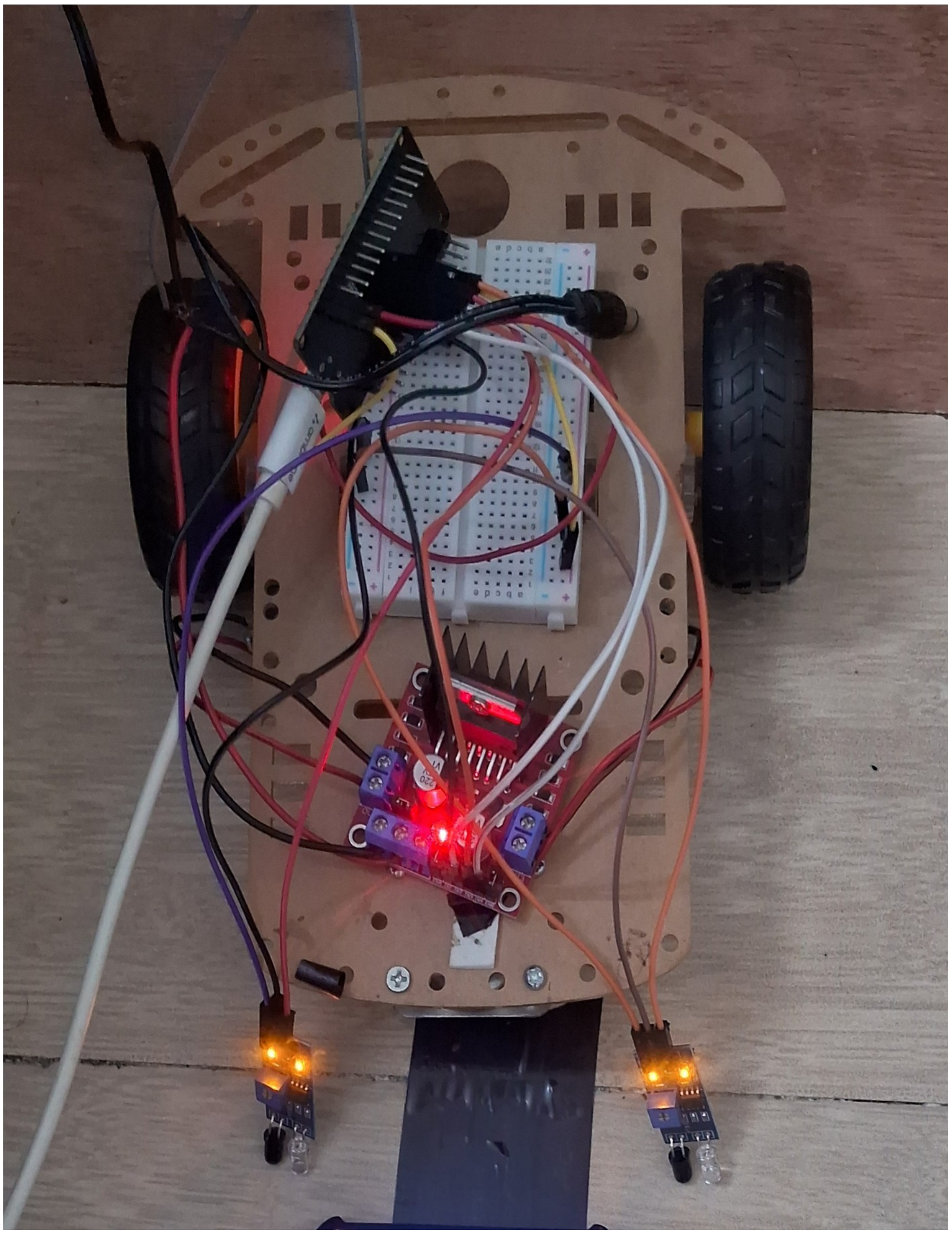


Fig. 1.3: Electrical Representation of Line Follower Robot

Electronics Representation

The electronics representation of the line follower robot is shown in Fig. 3. The IR sensors are connected to the ESP32 microcontroller through GPIO pins for left and right line detection. The ESP32 provides four output signals to the L298N motor driver inputs (IN1–IN4), which control the direction of the left and right DC motors. The motor driver outputs (OUT1–OUT4) are connected to the motors to enable forward, reverse, and turning motions. A 12V battery powers the motor driver and motors, while a regulated 5V supply powers the ESP32 and IR sensors. A common ground is shared between all components to ensure stable operation.

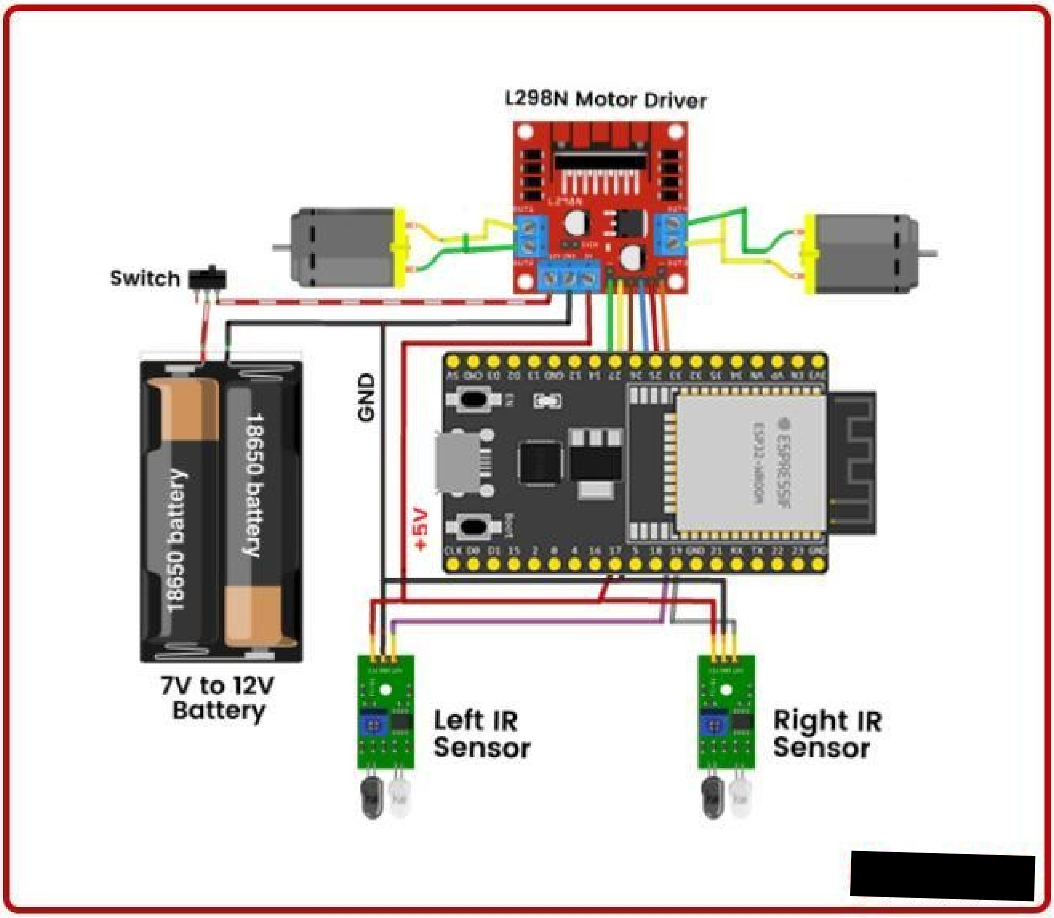


Fig. 1.4: Electronic Circuit Representation of the Line Follower Robot

Project Objectives

The main objective of this project is to design and implement a **Bluetooth-Controlled Line Follower Robot** that combines the concepts of **autonomous navigation, embedded systems, and wireless communication**. The project aims to build a reliable, low-cost robotic system capable of following a black line path while also being controlled remotely via Bluetooth.

The detailed objectives are:

1. **Design and Hardware Setup**
   - To build a compact robotic platform using ESP32, L298N motor driver, DC motors, IR sensors, and Bluetooth communication.
   - To establish proper wiring and interfacing of the ESP32 with the motor driver and sensors for accurate control.
2. **Sensor Integration**
   - To implement dual infrared (IR) sensors for detecting black line paths on a contrasting surface.
   - To ensure reliable line detection under varying lighting and surface conditions.
3. **Autonomous Navigation**
   - To program the ESP32 for real-time decision-making based on sensor inputs.
   - To enable smooth forward movement, left and right turns, and automatic stopping when the line is undetected.
4. **Bluetooth Control**
   - To integrate Bluetooth connectivity for remote ON/OFF control of the robot using a smartphone or paired device.
   - To ensure user-friendly wireless control with minimal delay in communication.
5. **Motor Control and Power Management**
   - To drive two DC motors through the L298N motor driver for precise directional control.
   - To optimize motor performance for smooth navigation and minimal power consumption.
6. **Failsafe Mechanism**
   - To implement safety features where the robot halts automatically when both sensors fail to detect the line.

- To prevent random or unintended movements that may damage the robot or environment.

7. **Low-Cost and Scalable Design**

- To create a cost-effective robotic solution using affordable components.
- To ensure the system is modular and easily extendable for future upgrades (e.g., Wi-Fi, obstacle detection, speed control).

8. **Educational and Practical Relevance**

- To provide hands-on learning of embedded systems, robotics, and IoT integration.
- To demonstrate practical applications in **automation, logistics, and service robotics**.

9. **Testing and Validation**

- To test the robot's performance in different track layouts (straight, curved, junctions).
- To evaluate accuracy, response time, and robustness under real conditions.

Innovative Part in This Project

1. **Bluetooth Integration for Control**
   - The robot can be **started and stopped wirelessly** using Bluetooth commands (S = Start, X = Stop).
   - This adds flexibility compared to normal line followers that start as soon as power is given.
2. **Robot ON/OFF Flag (robotActive)**
   - A software-controlled flag manages whether the robot should actively follow the line or remain stopped.
   - Ensures that motor control only works when the robot is intentionally started via Bluetooth.
3. **Serial Monitoring for Debugging**
   - Real-time **IR sensor readings** and received **Bluetooth commands** are printed on the Serial Monitor.
   - This allows the user to **observe robot decisions** (forward, left, right, stop) and fine-tune sensor alignment.
4. **Failsafe Stop Mechanism**
   - If both sensors detect white (robot leaves the track), the robot **immediately stops motors** instead of moving randomly.
   - Prevents uncontrolled movement outside the track.

Methodology and Results

The methodology of this project is divided into several systematic stages to ensure the successful design, implementation, and testing of the **Bluetooth-Controlled Line Follower Robot**. The approach follows a logical sequence starting from hardware setup, followed by software development, integration, and validation.

1. System Design and Planning

- The overall design of the robot was planned to integrate **ESP32** as the control unit, **L298N motor driver** for motor actuation, **DC motors** for movement, **IR sensors** for line detection, and **Bluetooth** for wireless control.
- A block diagram of the system was created to represent the interaction between sensors, controller, motors, and communication modules.
- The flow of data was defined: IR sensors detect line → ESP32 processes input → movement decision → motor driver drives motors.

2. Hardware Setup

- **ESP32 Microcontroller**: Selected for its built-in Bluetooth, GPIO flexibility, and fast processing.
- **IR Sensors**: Placed at the front of the robot to detect black/white surface contrast.
- **Motor Driver (L298N)**: Connected between ESP32 and DC motors to enable forward, backward, left, and right movement.
- **DC Motors & Chassis**: Two-wheel drive with caster wheel support was used for mobility.
- **Power Supply**: A rechargeable 12V battery was used to power the system, with proper regulation for ESP32 and sensors.

3. Software Development

- **Arduino IDE with ESP32 libraries** was used for coding and uploading programs to the microcontroller.
- **Bluetooth Module**: ESP32's onboard Bluetooth was configured to communicate with a paired smartphone. Commands such as S (start) and X (stop) were defined.
- **Sensor Reading**: IR sensor inputs were read as digital signals (1 = black, 0 = white).

- **Decision Logic**:
    - Left = 1, Right = 1 → Move Forward
    - Left = 1, Right = 0 → Turn Left
    - Left = 0, Right = 1 → Turn Right
    - Left = 0, Right = 0 → Stop Motors
- **Motor Control**: Appropriate HIGH/LOW signals were sent from ESP32 GPIOs to the L298N motor driver for movement.

4. Integration of Bluetooth Control

- The ESP32 was programmed to continuously check for incoming Bluetooth commands.
- On receiving S, the robot entered **Line Follower Mode**.
- On receiving X, the robot stopped and remained idle until the next command.
- This allowed both **autonomous navigation** and **remote user control**.

5. Testing and Calibration

- The IR sensors were calibrated to correctly differentiate between black and white surfaces under various lighting conditions.
- The robot was tested on straight, curved, and junction tracks.
- Motor direction was adjusted by modifying wiring and control logic to ensure correct left/right turns.
- Bluetooth range and response time were verified to ensure smooth operation.

6. Validation and Optimization

- The robot's performance was validated against the defined objectives: accurate line following, quick response, and reliable wireless control.
- Adjustments were made to sensor placement for better accuracy.
- Power consumption was optimized by fine-tuning motor control signals and ensuring minimal idle usage.

7. Final Deployment

- The complete system was assembled with all components fixed securely on the chassis.

- Final trials were conducted on different track patterns to confirm stability and reliability.
- The project was successfully demonstrated with both **Bluetooth control** and **autonomous line-following functionality**.

8. Pictures

ESP32

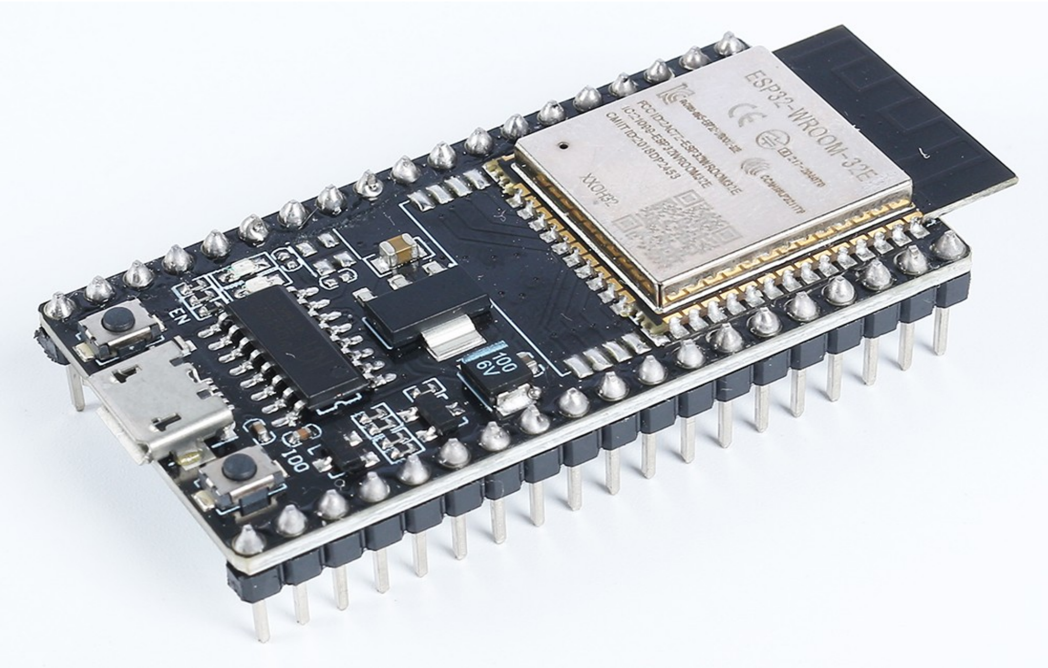


Fig-2.1

L298N MOTOR DRIVER

Fig-2.2

BLUETOOTH CONTROL

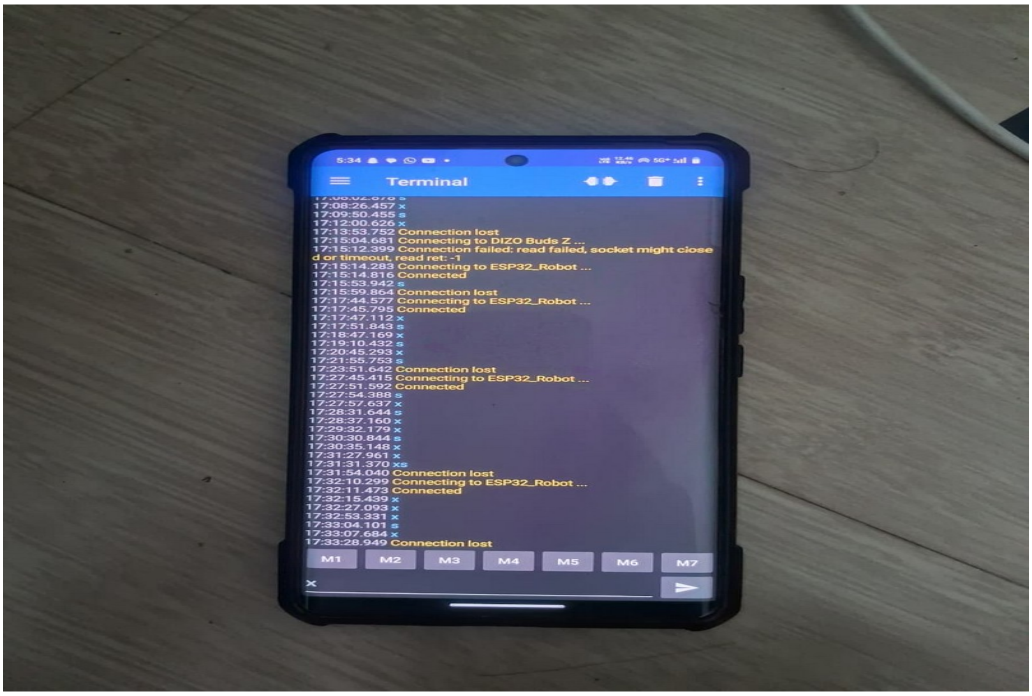


Fig-2.3

RESULT

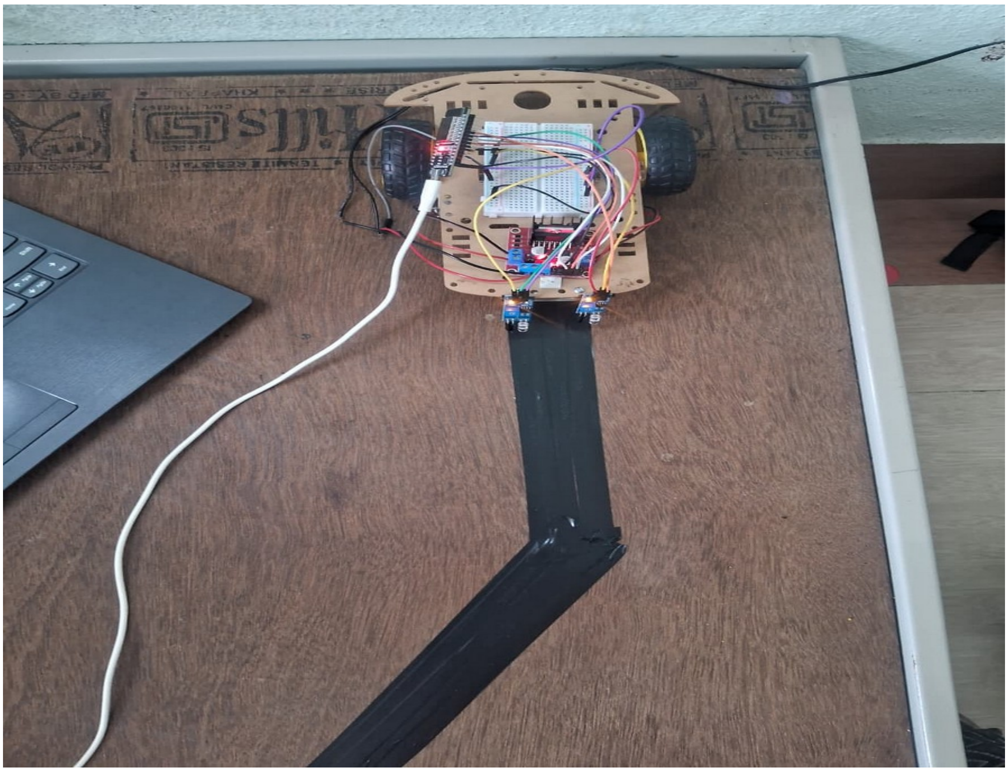


Fig-2.4

Test Reports

1. Test Setup

- **Hardware**: ESP32, L298N motor driver, 2 DC motors, 2 IR sensors, 12V battery, Smartphone with Bluetooth terminal.
- **Connections**:
- Motors: IN1=26, IN2=25, IN3=33, IN4=32
- IR Sensors: Left=GPIO34, Right=GPIO35
- Bluetooth: ESP32 paired as ESP32_Robot.
- **Procedure**: Upload code → Pair phone → Send commands (S/X) → Place robot on line → Observe motor and sensor behavior.

2. Quantitative Targets for ESP32 Bluetooth + Line Follower Robot

1. **Robot Speed**
    - Target: **0.25 – 0.30 m/s** (25–30 cm/s) depending on DC motor RPM and PWM control.
2. **Bluetooth Range**
    - Target: **~10 meters indoors** (ESP32 Classic Bluetooth standard).
3. **IR Sensor Detection Time**
    - Target: **<100 ms** response time for line detection and correction.
4. **Line Following Accuracy**
    - Target: **≥90% accuracy** on test track without deviating from the line.
5. **Turning Capability**
    - Target: Successfully handle **90° turns** on the line.
6. **Payload Capacity**
    - Target: **0.5 – 1.0 kg** (small object prototype, given motor torque).
7. **Power Supply & Runtime**
    - Target: **30–45 minutes** continuous operation on a **12V rechargeable battery**.
8. **Bluetooth Command Latency**

    Target: **<200 ms** from mobile app command to motor response.

3. Results

Test Case	Procedure	Expected Result	Observed Result	Status
Bluetooth pairing	Pair phone with ESP32	Successful pairing	YES	Pass
Start/Stop command	Send S and X	Robot starts and stops accordingly	YES	Pass
Forward movement	Both sensors on black	Robot moves forward	YES	Pass
Left turn	Left=Black, Right=White	Robot turns left	YES	Pass
Right turn	Left=White, Right=Black	Robot turns right	YES	Pass
Off-line detection	Both sensors on white	Robot stops	YES	Pass

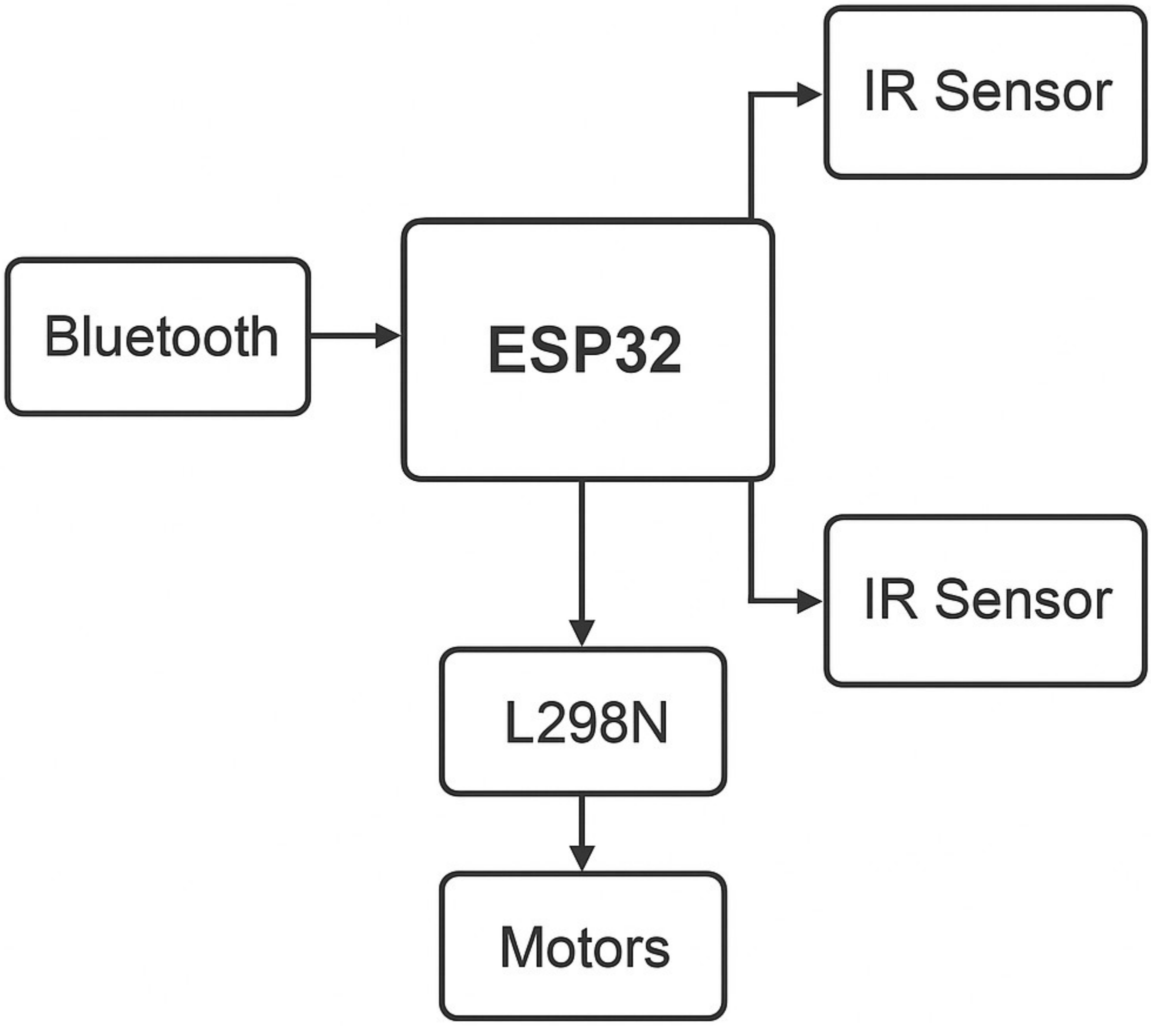
4. Observations

- Motors responded correctly to all movement commands.
- Robot followed the black line smoothly on straight and curved paths.
- Bluetooth control was responsive with minimal delay.
- IR sensors gave stable readings under normal light, but ambient light affected accuracy.

5. Analysis

- The system performed as expected in all test cases.
- The logic for line following and Bluetooth start/stop worked correctly.
- Slight performance issues observed in sharp turns due to only two sensors.
- Power stability and ambient light conditions play a key role in sensor accuracy.

Block Diagram of the Project



Social / Industry relevance of the project

The development of a Bluetooth-controlled line follower robot holds significant importance both socially and industrially.

1. Industrial Relevance

- **Automation in Manufacturing**: Line follower robots are widely used in industries for material handling, automated assembly lines, and warehouse logistics, reducing human labor and improving efficiency.
- **Smart Transportation Systems**: Similar principles are applied in automated guided vehicles (AGVs) and autonomous mobile robots (AMRs), which are extensively used in smart factories and logistics hubs.
- **Cost-Effective Solutions**: Using simple sensors and microcontrollers makes the system affordable, adaptable, and scalable for small- and medium-scale industries.
- **Bluetooth Integration**: The inclusion of Bluetooth enables wireless monitoring and control, which enhances flexibility in industrial automation and remote management.

2. Social Relevance

- **Educational Applications**: The project serves as an excellent learning tool for students and enthusiasts to understand robotics, embedded systems, and control theory.
- **Accessibility and Assistance**: Such robots can be adapted for assisting differently-abled individuals in carrying out small tasks by remotely controlling robots through mobile devices.
- **Research and Innovation**: Provides a foundation for further research in autonomous robotics, Internet of Things (IoT) integration, and smart infrastructure.
- **Employment and Skill Development**: Encourages skill-building in embedded systems, robotics, and wireless communication, which are highly valued in the job market.

Learning and Reflection

Working on the Bluetooth-controlled line follower robot provided me with valuable insights into embedded systems, robotics, and IoT-based applications. The project strengthened my understanding of how hardware and software components interact to achieve real-time automation. By integrating the ESP32 microcontroller with IR sensors, motor drivers, and Bluetooth communication, I learned how to build a system that combines both autonomous decision-making and manual control through wireless connectivity.

One of the key learnings was the importance of sensor calibration and precise motor control. I realized that even small variations in sensor readings could impact the path-following accuracy, requiring careful fine-tuning of both the hardware connections and the control algorithm. Additionally, I gained practical experience in configuring and programming the L298N motor driver to control the speed and direction of the DC motors effectively.

From a software perspective, I developed a deeper understanding of using Arduino IDE for ESP32 programming and how to implement Bluetooth communication using the BluetoothSerial library. This helped me to successfully establish wireless connectivity between the robot and a smartphone, enabling flexible control modes.

The project also enhanced my problem-solving skills. I encountered challenges such as incorrect motor direction, unstable line detection, and initial Bluetooth pairing issues. Overcoming these difficulties improved my debugging skills and reinforced the importance of iterative testing in embedded systems development.

Reflecting on the overall experience, this project not only helped me apply theoretical knowledge from microcontrollers and robotics but also prepared me for tackling real-world IoT and automation problems. It has motivated me to further explore advanced robotics concepts, such as PID control for smoother line following, integration of machine vision for intelligent navigation, and cloud connectivity for remote monitoring.

In conclusion, this project served as an excellent learning platform, strengthening my technical, analytical, and practical skills while enhancing my confidence in working with embedded systems and IoT technologies.

Main Code and File Structure

File and Folder Structue(Arduino IDE)

The Line Follower Robot was implemented on the ESP32 using the Arduino IDE in C++. The code, written in a single sketch file (main.ino), handles Bluetooth communication, IR sensor inputs, and motor control through the L298N driver.

Folder Structure:

```
ESP32_LineFollower_Robot/
└── main.ino
```

Code Architechture

1. Setup & Initialization

- setup() configures:
    - Bluetooth (SerialBT.begin("ESP32_Robot"))
    - Motor driver pins (IN1–IN4)
    - IR sensor pins (IR_LEFT, IR_RIGHT)
    - Stops motors at startup

2. Bluetooth Control

- Reads commands from mobile via Bluetooth.
- Commands:
    - S → Start line following mode.
    - X → Stop robot.

3. Line Following Logic

- Inside loop(), when robot is active:
    - Reads IR sensors (digitalRead(IR_LEFT), digitalRead(IR_RIGHT)).
    - Decision logic:
        - Both sensors on black → move forward
        - Left black, right white → turn left
        - Left white, right black → turn right
        - Both white → stop

4. Motor Control Functions

- moveForward(), turnLeft(), turnRight(), stopMotors()
- Control L298N motor driver through GPIO pins.

Main Code

1. Library and Global Declarations

```
#include "BluetoothSerial.h"   // Bluetooth library

BluetoothSerial SerialBT;      // Bluetooth Serial object

const int IN1 = 26;  // Left Motor pin 1

const int IN2 = 25;  // Left Motor pin 2

const int IN3 = 33;  // Right Motor pin 1

const int IN4 = 32;  // Right Motor pin 2

const int IR_LEFT  = 34;   // Left IR sensor input

const int IR_RIGHT = 35;   // Right IR sensor input

// Robot Status

bool robotActive = false;  // ON/OFF flag
```

2. Main Code (setup & loop)

```
//Setup Function

void setup() {

  Serial.begin(115200);                // Debug serial monitor

  SerialBT.begin("ESP32_Robot");       // Bluetooth pairing name

  Serial.println("Bluetooth Line Follower Ready!");

  // Motor pin setup

  pinMode(IN1, OUTPUT);

  pinMode(IN2, OUTPUT);
```

```
  pinMode(IN3, OUTPUT);

  pinMode(IN4, OUTPUT);

  // Sensor pin setup

  pinMode(IR_LEFT, INPUT);

  pinMode(IR_RIGHT, INPUT);

  stopMotors();  // Ensure motors are off at start

}

// Main Loop

void loop() {

  // Bluetooth command check

  if (SerialBT.available()) {

    char cmd = SerialBT.read();   // Read command from phone

    Serial.print("Bluetooth Command: ");

    Serial.println(cmd);

    if (cmd == 'S' || cmd == 's') {

      robotActive = true;   // Start line following

      Serial.println("Robot STARTED (Line Follow Mode)");

    }

    else if (cmd == 'X' || cmd == 'x') {

      robotActive = false;  // Stop the robot

      stopMotors();

      Serial.println("Robot STOPPED");

    }
```

```
}

if (robotActive) {

  int leftSensor  = digitalRead(IR_LEFT);   // Read left sensor

  int rightSensor = digitalRead(IR_RIGHT);  // Read right sensor

  // Debugging output

  Serial.print("Left Raw: ");

  Serial.print(leftSensor);

  Serial.print(" | Right Raw: ");

  Serial.println(rightSensor);

  // Convert sensor values: 1 = Black, 0 = White

  bool leftBlack  = (leftSensor == 1);

  bool rightBlack = (rightSensor == 1);

  // Movement logic

  if (leftBlack && rightBlack) {

    moveForward();    // Both black → forward

  }

  else if (leftBlack && !rightBlack) {

    turnLeft();       // Left black → turn left

  }

  else if (!leftBlack && rightBlack) {

    turnRight();      // Right black → turn right

  }

  else {
```

```
      stopMotors();     // Both white → stop
    }
  }
}
```

3. Sensor and Motor Control Code

```
// Motor Control Functions
// Move forward
void moveForward() {
  digitalWrite(IN1, HIGH);  // Left motor forward
  digitalWrite(IN2, LOW);
  digitalWrite(IN3, LOW);   // Right motor forward
  digitalWrite(IN4, HIGH);
}
// Turn left
void turnLeft() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);  // Left motor backward
  digitalWrite(IN3, LOW);
  digitalWrite(IN4, HIGH);  // Right motor forward
}
// Turn right
void turnRight() {
  digitalWrite(IN1, HIGH);
```

```
  digitalWrite(IN2, LOW);   // Left motor forward

  digitalWrite(IN3, HIGH);

  digitalWrite(IN4, LOW);   // Right motor backward

}

// Stop motors

void stopMotors() {

  digitalWrite(IN1, LOW);

  digitalWrite(IN2, LOW);

  digitalWrite(IN3, LOW);

  digitalWrite(IN4, LOW);

}
```

Conclusion and Future Scope

Conclusion

The development of the **ESP32 Bluetooth-Controlled Line Follower Robot** has provided valuable insights into embedded systems, robotics, and wireless communication. The project successfully demonstrated how IR sensors can be used for real-time line detection and navigation, while Bluetooth connectivity enabled remote control and flexibility in operation. The integration of the ESP32 microcontroller with the L298N motor driver facilitated efficient control of the DC motors, resulting in smooth and reliable motion along the predefined path.

This project not only enhanced technical knowledge in hardware–software integration but also improved problem-solving and debugging skills. It showcased the importance of microcontroller programming, sensor interfacing, and motor control in building intelligent robotic systems. Furthermore, it emphasized the role of wireless technologies in modern robotics, opening possibilities for future enhancements such as IoT-based monitoring, advanced sensor fusion, or AI-driven decision-making.

In conclusion, the project achieved its objectives and proved to be a practical implementation of embedded system concepts in real-world robotics. It serves as a strong foundation for further exploration in automation, smart systems, and advanced robotics applications.

Future Scope

The Bluetooth-controlled line follower robot developed in this project can be further enhanced and expanded in several directions to increase its efficiency, reliability, and practical applications:

1. **Integration with IoT and Cloud Services** – By connecting the robot to IoT platforms, real-time monitoring, data logging, and remote control through mobile applications or web dashboards can be implemented. This enables smart automation and predictive maintenance in industries.
2. **Advanced Navigation and Obstacle Avoidance** – The addition of ultrasonic, LIDAR, or computer vision modules will allow the robot to detect and avoid obstacles, making it suitable for more complex environments beyond simple line following.

3. **Renewable Energy Integration** – Incorporating solar-powered charging systems can make the robot self-sustainable, extend operational time, and reduce dependency on manual recharging.

4. **Swarm Robotics Applications** – Multiple robots can be coordinated to work as a team for tasks such as automated material handling in warehouses, agricultural monitoring, or collaborative search-and-rescue missions.