Line Following Robot – Project Report

# 1. Introduction

In today’s world of automation and intelligent systems, robotics has emerged as a pivotal technology across various industries. One of the most fundamental and widely studied robotic applications is the **line following robot** — an autonomous mobile robot capable of detecting and following a predefined path or line, typically black on a white surface. This concept is not only simple but also forms the foundation for more complex robotic navigation systems used in industrial automation, logistics, and intelligent transport systems.

The **motivation** behind developing a line following robot lies in its practical real-world applications and its educational value. From automatic guided vehicles in warehouses to robotic assistants in hospitals, the ability to navigate autonomously is a crucial feature. Moreover, for students and hobbyists, building a line following robot offers hands-on experience with sensors, embedded systems, control logic, and real-time decision making.

This project aims to design and build a cost-effective and efficient line following robot using an Arduino microcontroller, IR sensors, and basic motor control logic. The simplicity of hardware combined with intelligent software logic makes this robot a powerful educational tool while also demonstrating key principles in robotics and automation.

# 2. Objective

To build a robot that can autonomously follow a black line on a white surface using IR sensors and Arduino.

# 3. Research and Background

Line following robots are often considered the stepping stone into the world of mobile robotics due to their simplicity and practical relevance. These robots are designed to follow a path, typically a black line on a white surface, using infrared (IR) sensors. The IR sensors help in identifying contrast by detecting the difference in reflectivity between the black line and the surrounding white background. This contrast detection allows the robot to determine its position relative to the line. The robot can then take corrective actions to stay on course.

The behaviour of the robot is controlled by a microcontroller, such as an Arduino, which processes the sensor inputs and commands the motors accordingly. The basic logic relies on reading sensor outputs and making directional decisions such as turning left, right, or moving forward. Line following can be accomplished with as few as two sensors, but using three or more improves accuracy. With multiple sensors, the robot can detect sharp turns and recover quickly from deviations. This capability makes it suitable for navigating more complex paths.

More advanced versions of line followers employ **PID (Proportional-Integral-Derivative)** control to enhance performance. PID control continuously calculates an error value and adjusts the motor speeds in a smooth and balanced manner. This reduces jerky movement and increases precision, especially at higher speeds or during sharp turns. Research has shown that integrating PID control in even simple robots significantly improves tracking accuracy. Thus, line following robots not only offer a platform for hands-on learning but also a testbed for implementing and analyzing control algorithms.

# 4. Theory and Working Principle

4.1 IR Sensors

- IR Transmitter emits infrared light.

- IR Receiver (Photodiode) detects the reflected light.

- A black surface absorbs more IR, giving low output; white reflects IR, giving high output.

4.2 Motor Driver

- L298N Dual H-Bridge is used to control the direction and speed of two DC motors.

4.3 Line Detection Logic

- The robot adjusts its motors based on sensor readings:

- Center sensor on line → go straight.

- Left sensor on line → turn left.

- Right sensor on line → turn right.

# 5. Components Required

• Arduino Uno - 1

• IR Sensor Module - 3

• L298N Motor Driver - 1

• DC Motors (BO type) - 2

• Robot Chassis - 1

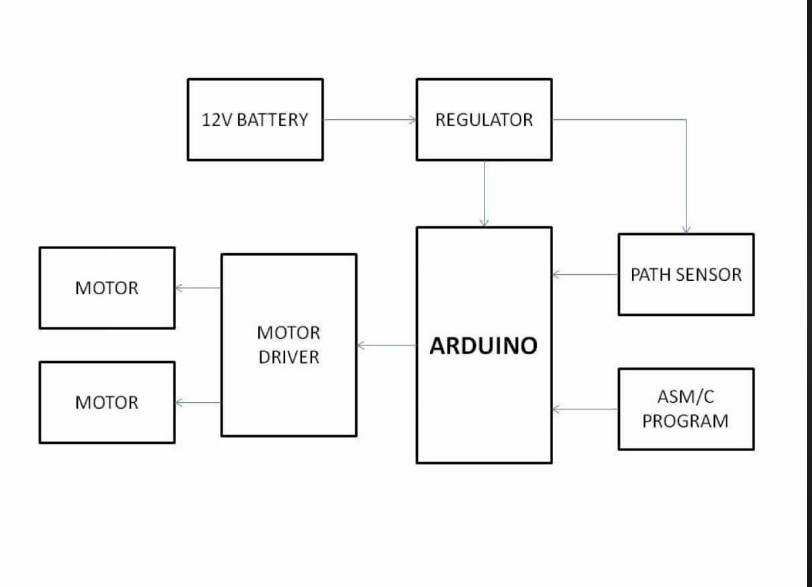
• Wheels - 2

• Caster Wheel - 1

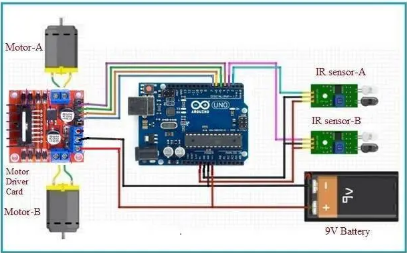
• Jumper Wires - As required

• 12V Battery + Switch - 1

# 6. Block Diagram



# 7. Circuit Diagram



# 8. Arduino Code

#define leftSensor A0

#define centerSensor A1

#define rightSensor A2

#define in1 8

#define in2 9

#define in3 10

#define in4 11

void setup() {

pinMode(leftSensor, INPUT);

pinMode(centerSensor, INPUT);

pinMode(rightSensor, INPUT);

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(in3, OUTPUT);

pinMode(in4, OUTPUT);

}

void loop() {

int left = digitalRead(leftSensor);

int center = digitalRead(centerSensor);

int right = digitalRead(rightSensor);

if (center == 1 && left == 0 && right == 0) {

forward();

} else if (left == 1 && center == 0 && right == 0) {

turnLeft();

} else if (right == 1 && center == 0 && left == 0) {

turnRight();

} else {

stopMotors();

}

}

void forward() {

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

}

void turnLeft() {

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

}

void turnRight() {

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

}

void stopMotors() {

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

# 9. Mathematical Model

Sensor outputs: L, C, R ∈ {0, 1}

Motor control logic:

| L | C | R | Action | M\_L | M\_R |

|---|---|---|------------|-----|-----|

| 0 | 1 | 0 | Forward | 1 | 1 |

| 1 | 0 | 0 | Turn Left | 0 | 1 |

| 0 | 0 | 1 | Turn Right | 1 | 0 |

| 0 | 0 | 0 | Stop | 0 | 0 |

Error = (L × -1) + (C × 0) + (R × 1)

Correction = Kp × Error + Kd × d(Error)/dt + Ki × ∫Error dt

# 10. Results and Observations

- Successfully follows black line on white surface.

- Accurate turns.

- Affected by surface and ambient light.

- Tuning speed improves stability.

# 11. Conclusion

The line following robot developed in this project successfully demonstrates the core principles of robotics, embedded systems, and real-time sensor integration. By using infrared sensors for line detection and an Arduino-based control system, the robot was able to navigate a predefined path with simple logic and effective motor control. This project serves as a valuable introduction to autonomous systems and highlights the practical implementation of hardware and software working in unison.

Through this project, key concepts such as sensor data acquisition, motor actuation, and logical decision-making were explored and applied. It also emphasizes the importance of tuning and calibration for consistent robot behavior under different lighting and surface conditions. The robot’s responsiveness and ability to follow the line accurately validate the approach used.

Although the system performs well in basic scenarios, there are several areas for future enhancement. Implementing **PID control** can significantly improve line tracking precision and stability. Adding **junction detection logic** would enable the robot to make intelligent decisions at path intersections. Integrating **color sensors** could allow it to follow colored lines or perform tasks based on color cues. Furthermore, enabling **maze-solving algorithms** could transform the robot into a more advanced autonomous navigation system capable of complex tasks.

