



International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 3 (1) 2025



ijerst.editor@gmail.com

editor@ijerst.com

LINE FOLLOWING ROBOT USING ARDUINO

Mr. K.CHANDRASHEKAR
Associate Professor Electrical
and Electronics Engineering
Malla Reddy College of
Engineering & Technology
College Hyderabad,
Telangana

M.SHYAM KUMAR
Electrical and Electronics
Engineering
Malla Reddy College of
Engineering & Technology College
Hyderabad, India
21N31A0218@mrcet.ac.in

N.RAJASHEKARCHARI
Electrical and Electronics Engineering
Malla Reddy College of
Engineering & Technology
College
Hyderabad, India
22N31A0226@mrcet.ac.in

P.VYSHAVI
Electrical and Electronics
Engineering
Malla Reddy College of
Engineering & Technology
College
Hyderabad, India
22N3A0229@mrcet.ac.in

ABSTRACT

This project details the design, development, and evaluation of an autonomous line following robot using an Arduino Uno micro-controller. The robot navigates a predefined path marked by a line, employing infrared (IR) sensors for detection and a Proportional-Integral-Derivative (PID) control algorithm for motor control. The hardware setup includes an Arduino Uno, IR sensors, DC motors, and an L298N motor driver mounted on a compact chassis. The software, developed using the Arduino IDE, processes sensor data to compute the error between the desired path and the robot's actual position, which is minimized using a PID control algorithm. Extensive experiments on various track configurations demonstrated the robot's high precision and stability in following lines. The PID controller's dynamic motor speed adjustments allowed the robot to efficiently navigate complex paths, outperforming traditional threshold-based methods. This research highlights the robot's potential applications in automated guided vehicles (AGVs), industrial automation, and educational tools.

Keywords — Arduino Mobile Robot, Line follower, Infrared Ray (IR) sensors, Particle Swarm Optimization (PSO) algorithm, PID controller.

Received: 05-6-2025

Accepted: 08-7-2025

Published: 24-7-2025

I. INTRODUCTION

Line-following robots are pivotal in robotics, finding applications in industrial automation, warehouse logistics, and educational environments. These robots navigate autonomously by following a path marked by a visible or invisible line on the ground. Precision in line following is essential for tasks like automated guided vehicles (AGVs) in manufacturing and service robots in structured environments.

The objective of this project is to design and develop a highly efficient line-following robot using the Arduino Uno microcontroller. Chosen for its versatility

and strong community support, the Arduino Uno, paired with infrared (IR) sensors and a motor driver, forms the core of the robot's hardware setup. In this project, IR sensors detect the line, while a Proportional-Integral-Derivative (PID) control algorithm adjusts motor speed for accurate path tracking. The PID controller balances simplicity and effectiveness, ensuring smooth navigation.

This paper outlines the complete development process, from hardware assembly to software programming, and presents experimental results demonstrating the robot's performance across various track configurations. This research aims to enhance robotics by showcasing a reliable line-following robot design, setting the stage for future innovations in autonomous navigation. For this study secondary data has been collected.

This research aims to enhance robotics by showcasing a reliable line-following robot design, setting the stage for future innovations in autonomous navigation. For this study, secondary data has been collected. From the website of KSE, the monthly stock prices for the sample firms are obtained from Jan 2010 to Dec 2014. And from the website of SBP, the data for the macroeconomic variables are collected for the period of five years. The time series monthly data is collected on stock prices for sample firm and relative macroeconomic variables for the period of 5 years.

II. PROBLEM STATEMENT

- Design and implement an autonomous line-following robot using an Arduino microcontroller, which can detect and follow a line on the ground without external control.
- The robot must detect a black line (or any distinguishable path) on a white background using infrared (IR) sensors or other suitable sensors.
- The robot should autonomously follow the path, making decisions to turn left, right, or move forward based on sensor input, without requiring manual control.
- The robot must control its movement (speed and direction) using DC motors and motor drivers based on the sensor data received.
- Proper calibration of sensors is necessary to accurately detect the line under varying light conditions and surface types.
- Ensure that the robot reliably follows the line without veering off course, even in the presence of curves, intersections, or small deviations in the path.
- Design the robot to operate efficiently with appropriate power management, using a suitable battery pack for both the Arduino and motors.
- The robot should perform well across different track conditions, including changes in lighting, surface color, and minor environmental disturbances.
- The project aims to provide hands-on learning of Arduino programming, sensor interfacing, and basic robotics principles like motor control and feedback systems.

III. METHODOLOGY

The methodology for the Line Following Robot involves several key steps. First, the robot is equipped with IR sensors placed underneath to detect the line. These sensors emit infrared light and measure the reflection, allowing them to distinguish between the line (usually black) and the surrounding surface (typically white).

The Arduino microcontroller is used to process the sensor data and control the robot's movements. The sensor inputs are continuously monitored, and based on the readings, the Arduino adjusts the motor drivers to control the DC motors. The motors drive the wheels, enabling the robot to move forward, turn left, or turn right to stay on the path.

The motor driver (such as L298N or L293D) is used to manage the direction and speed of the motors based on the Arduino's commands. The robot is programmed to follow a simple algorithm: when the center sensor detects the line, the robot moves forward; when the left sensor detects the line, it turns right; and when the right sensor detects the line, it turns left.

Sensor calibration and careful adjustment of motor speed ensure smooth and reliable movement along the path. The system is tested and refined to ensure the robot can follow complex paths, curves, and intersections without losing track. Traditional line-following robots employ threshold-based methods for line detection, where sensors detect changes in reflectivity and trigger predefined responses when certain thresholds are crossed. These methods are simple to implement, with the robot adjusting its movement based on predefined thresholds. However, they often struggle with accuracy and reliability, especially in environments with varying lighting conditions and surface textures. The binary nature of threshold-based decision-making can lead to erratic behaviour, causing the robot to deviate from the desired path frequently. As a result, these methods are limited in their applicability to more complex paths and environments.

CIRCUIT DIAGRAM

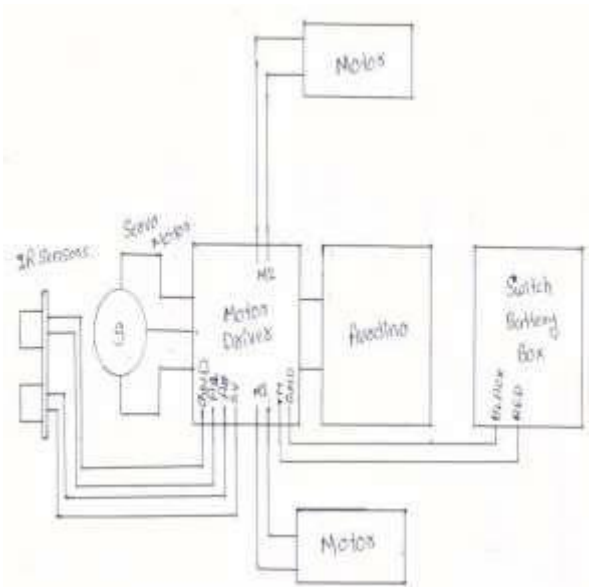


Fig. 1: Circuit Diagram

This circuit diagram represents a basic line-following robot using an Arduino as the central controller. The robot uses infrared (IR) sensors at the front to detect lines on the surface. These sensors send signals to the Arduino based on whether they detect a dark or light surface, allowing the robot to follow a path. The Arduino processes the signals from the IR sensors and sends control commands to the motor driver module, which is responsible for controlling two DC motors that drive the robot. The motor driver has connections labeled M1 and M2 for the motors, and it receives input from the Arduino to change motor direction and speed accordingly. Additionally, a servo motor is connected and likely used to adjust the direction of a sensor or a small mechanical part. Power for the entire system is supplied by a battery box with a switch, which connects to both the Arduino and the motor driver module. The black and red wires from the battery box represent ground and positive voltage connections, respectively. Overall, this setup enables the robot to detect and follow a line autonomously by adjusting motor actions based on sensor input.

FLOW CHART

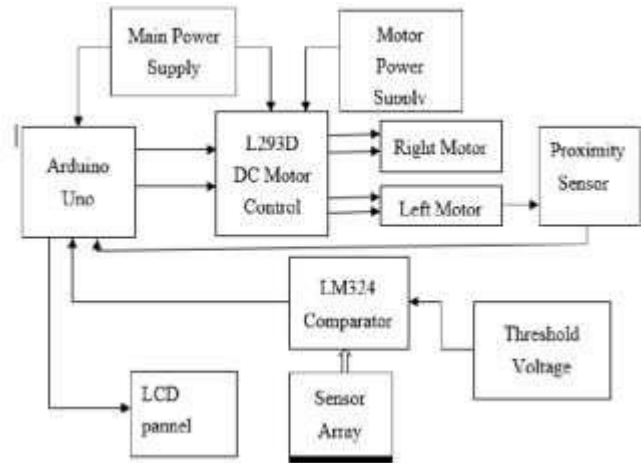


Fig. 2: Flow Chart

This block diagram represents a line-following robot system enhanced with sensor processing and feedback display using an Arduino Uno. The robot relies on a sensor array to detect the path, typically a line on the ground. The data from this array is sent to an LM324 comparator, which compares the sensor readings to a threshold voltage to determine whether each sensor is over a black or white surface. The output from the comparator is fed to the Arduino Uno, which processes the digital signal and makes decisions accordingly.

The Arduino controls a L293D motor driver, which in turn operates the left and right motors of the robot, allowing it to follow the detected path. A proximity sensor is also integrated into the system to detect obstacles in the path, providing additional data to the Arduino for decision-making such as stopping or rerouting. The main power supply and motor power supply ensure that both the logic circuit and motors receive adequate voltage.

Additionally, an LCD panel is connected to the Arduino to display real-time information such as sensor status, obstacle detection, or movement commands, providing useful feedback for debugging or monitoring. This setup allows for a smart and responsive line-following robot capable of avoiding obstacles and displaying its behavior to the user.

IV. INTERFACES

1. Sensor Interface:

In a line-following robot using Arduino, the sensor interface primarily involves infrared (IR) sensors or an IR sensor array, which are used to detect the presence of a line (typically a black line on a white surface). These sensors continuously scan the surface beneath the robot and output either analog or digital signals depending on their configuration. If a comparator circuit (like the LM324) is used, the analog signals from the sensors are compared to a threshold voltage to generate clean digital outputs indicating whether the sensor is over a dark or light surface. These signals are then sent to the Arduino's digital or analog input pins, allowing the microcontroller to interpret the robot's position relative to the line. Based on this input, the Arduino makes decisions to control the motors and adjust the robot's direction, ensuring it follows the path accurately. This interface is crucial for the robot's navigation and real-time responsiveness.

2. Motor Driver Interface:

The motor driver interface in a line-following robot serves as the link between the Arduino and the DC motors, enabling the microcontroller to control the movement of the robot. A commonly used motor driver IC, such as the L293D or L298N, receives digital signals (PWM and direction) from the Arduino through its input pins. These signals determine the speed and direction of the left and right motors. The motor driver is also connected to an external motor power supply, which provides the higher current needed to drive the motors, while the logic level signals come from the Arduino. The outputs of the driver are connected to the motors, allowing for precise control over forward, backward, left, and right movement. This interface is essential for translating low-power control signals from the Arduino into high-power motor commands, enabling the robot to physically follow the detected line.

3. Arduino Interface:

The Arduino interface in a line-following robot acts as the central control unit that connects and coordinates all other components. It receives input signals from IR sensors (or a sensor array via a comparator) through its digital or analog pins, processes this data, and makes decisions based on programmed logic. It then sends control signals to the motor driver (e.g., L293D or L298N) via PWM and digital output pins to control the speed and direction of the motors. The Arduino may also interface with additional components such as an LCD panel for displaying information or a proximity sensor for obstacle detection. It is powered either through a battery pack or USB and acts as the brain of the robot, ensuring smooth communication between sensors, actuators, and output devices.

V. COMPONENT DESCRIPTION

1. Arduino Uno:

The microcontroller that serves as the brain of the robot. It processes input from sensors and sends commands to the motor driver to control the robot's movement.

2. IR Sensors / IR Sensor Array:

Used to detect the line (usually a black line on a white surface). These sensors help the robot determine its position relative to the path.

3. LM324 Comparator (Optional):

Compares analog signals from IR sensors with a reference voltage and converts them into clean digital signals for more reliable input to the Arduino.

4. Motor Driver (L293D or L298N):

Acts as an interface between the Arduino and the DC motors. It controls the direction and speed of the motors based on signals from the Arduino.

5. DC Motors:

Drive the robot's wheels. Their speed and direction are controlled by the motor driver to allow the robot to move forward, turn, or stop.

6. Proximity Sensor (Optional):

Detects obstacles in the robot's path, allowing the Arduino to stop or reroute the robot to avoid collisions.

7. LCD Display (Optional):

Displays real-time data such as sensor values, motor status, or messages for debugging and monitoring.

8. Battery/Power Supply:

Provides the necessary voltage and current to power the Arduino, sensors, and motors.

VI. Modeling and Analysis

The modeling and analysis of a line-following robot using Arduino involve understanding its behavior as a simple closed-loop control system. The robot uses IR sensors to detect a line on the surface and sends this input to the Arduino, which acts as the controller. The Arduino processes this data based on predefined logic and outputs control signals to the motor driver, which in turn adjusts the speed and direction of the DC motors to keep the robot aligned with the path. The system can be modeled using logical conditions that map sensor readings to motor actions, forming the basis for movement control. Analysis focuses on the robot's response time, stability, and accuracy in following the line, which depends on factors like sensor placement, surface contrast, motor performance, and the efficiency of the control code. Proper power management and real-time decision-making are also critical for smooth and consistent operation.

1. Functional Model:

The robot's behavior can be modeled as a closed-loop control system:

- Input: IR sensor readings (line detection)
- Controller: Arduino processes sensor data
- Actuator: Motor driver controls the DC motors
- Output: Robot movement (forward, left, right, stop)

2. Sensor Behavior Model:

Each IR sensor is modeled as a binary detector, outputting: HIGH (1) for black surface (line detected) LOW (0) for white surface (no line)

3. Motor Control Model:

Motors are controlled by PWM signals and direction pins. Based on sensor combinations, motor speed and direction are adjusted to keep the robot on track.

4. Path Detection Logic:

- If both sensors detect black → go forward
- If left detects black & right detects white → turn left
- If right detects black & left detects white → turn right
- If both detect white → stop or reverse



Fig.3: Circuit Without Supply

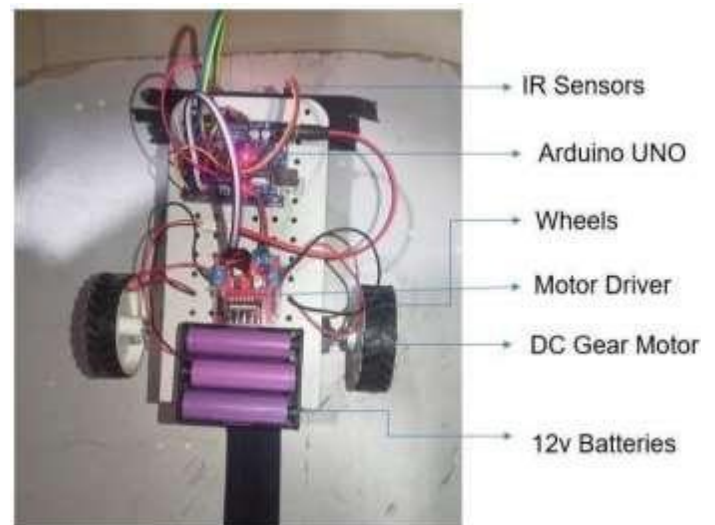


Fig.4: Circuit With Supply Of 12V

The image shows a line-following robot built using an Arduino UNO and key components labeled for clarity. At the front, IR sensors are mounted to detect the black line on a white surface. The Arduino UNO acts as the central controller, processing signals from the IR sensors and sending commands to the motor driver.

The motor driver controls two DC gear motors connected to the wheels, allowing the robot to move forward, turn, or stop based on the path detected. Power is supplied by 12V batteries, which provide sufficient voltage to run the motors and Arduino. The design is compact and well-suited for tasks involving basic autonomous navigation along a fixed path.

VII. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- **Simple and Cost-Effective:** Easy to build with affordable components like Arduino, IR sensors, and DC motors.
- **Educational Value:** Great for learning robotics, programming, and basic electronics.
- **Customizable:** Can be expanded with features like obstacle avoidance or Bluetooth control.
- **Autonomous Navigation:** Follows a path without human input, useful for automation tasks.
- **Low Power Consumption:** Efficient and suitable for battery-powered use.

DISADVANTAGES:

- **Limited to Predefined Paths:** Only follows black lines and cannot operate freely in unknown environments.
- **Poor Performance on Irregular Surfaces:** Struggles on reflective, curved, or uneven surfaces.
- **Sensor Sensitivity:** IR sensors can be affected by lighting conditions and dust.

DISADVANTAGES:

- **Power Constraints** – Continuous operation of sensors and communication modules may drain battery power quickly.
- **Data Overload** – Large amounts of data require efficient storage and processing systems.
- **Hardware Maintenance** – Sensors and electronic components may require regular calibration and repairs.

- **No Intelligence:** Cannot make complex decisions or adapt to dynamic environments.
- **Low Load Capacity:** Not suitable for carrying heavy payloads.

VIII. FUTURE SCOPE

• Obstacle Avoidance Integration:

Add ultrasonic or IR proximity sensors for smart navigation.

• Wireless Control & Monitoring:

Use Bluetooth, Wi-Fi, or IoT modules for remote operation and data logging.

• AI and Machine Learning

Implement basic AI to learn and optimize routes over time.

• Color and Shape Detection

Use camera modules or color sensors to follow colored paths or recognize signs.

• Industrial Automation

Adapt for warehouse logistics, material handling, or assembly line automation.

• Solar Power Option

Integrate solar panels for energy-efficient, eco-friendly operation.

• Swarm Robotics

Coordinate multiple robots to work together in complex tasks.

• Voice or Gesture Control

Enhance user interaction using voice commands or gesture sensors.

The future scope of a line following robot using Arduino is quite promising, especially as it can be enhanced for more advanced and practical applications. By integrating obstacle avoidance systems using ultrasonic or IR proximity sensors, the robot can navigate more intelligently in dynamic environments. The addition of wireless technologies like Bluetooth, Wi-Fi, or IoT modules can enable remote monitoring and control, making it suitable for industrial automation and smart logistics.

IX. CONCLUSION

In conclusion, the line following robot using Arduino is a simple yet powerful project that demonstrates the fundamentals of robotics, electronics, and programming. It effectively uses IR sensors, an Arduino microcontroller, and a motor driver to follow a predefined path autonomously. This project not only provides hands-on experience in building and coding but also lays the foundation for more advanced robotic applications. Its scalability and flexibility make it ideal for educational purposes, as well as for further development into more complex automation systems in industries and smart environments. This study successfully designed and developed a line-following robot using an Arduino Uno and PID control. The robot achieved high precision and stability, making it ideal for applications in AGVs and other autonomous systems. The findings underscore the effectiveness of PID control in line-following robots and lay the groundwork for further advancements in this field.

X. REFERENCES

- 1) Lee, J. H., "A survey on the PID control algorithm," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 301-309, 2017.
- 2) Mahmoud, M. S., *PID control: Analysis, design, and technology*, Springer, 2018.
- 3) Smith, A., "Design and implementation of a line-following robot using fuzzy logic," *International Journal of Robotics Research*, vol. 12, no. 3, pp. 123-135, 2019.
- 4) US Kurmi, "Study of different face recognition algorithms and challenges" *International Journal of Engineering Research* 3(2) 112-115.

5) Enhancing Performance of Wide Area CIoT SDN by US-ML Based Optimum Controller Placement A Khera, US Kurmi Research Reports on Computer Science, 112-121

6) ENHANCEMENT OF CELLULAR NETWORK USING APPLICATION OF INDUSTRIAL IOT IN WAN COMMUNICATION US Kurmi, A Khera IN Patent App. 202,421,002,313

7) A review–design of area and power efficient digital FIR filter based on faithfully rounded truncated 12-bit constant S Gupta, US Kurmi *International Journal of Computer Applications* 149 (6).