**LINE FOLLOWING ROBOT**

**J COMPONENT PROJECT REPORT FOR THE COURSE BECE301L – DIGITAL SIGNAL PROCESSING**

**BY**

**KALPANA RAJ-21BEC1074 SHUBHAM SHARMA – 21BEC1527 DEBARGHYA BANERJEE-21BEC1083**

**SUBMITTED TO Dr. RAMESH R**

Associate Professor, VIT Chennai



**SCHOOL OF ELECTRONICS ENGINEERING VELLORE INSTITUE OF TECHNOLOGY CHENNAI – 600127**

**JULY 2023**

**CERTIFICATE**

This is to certify that the Project work titled “**Line Following Robot**” is being submitted by **KALPANA RAJ (21BEC1074), SHUBHAM SHARMA (21BEC1527)** and **DEBARGHYA BANERJEE (21BEC1083)** for the course

B**ECE301L – DIGITAL SIGNAL PROCESSING**, is a record of bonafide work done under my guidance. The contents of this project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University

# Dr. RAMESH R

Associate Professor

School of Electronics Engineering (SENSE) VIT University, Chennai

Chennai – 600 127

# ABSTRACT

This project focuses on the design, development, and implementation of a line follower robot using Arduino. The objective of the project is to demonstrate the practical application of Arduino microcontroller in signal processing and automation. The line follower robot is designed to autonomously track a line using integrated sensors, motors, and programming.

Throughout the project, various components were utilized, including infrared sensors for line detection, motors for movement, and an Arduino microcontroller for control and decision- making. The programming code was developed to enable the robot to interpret sensor data, make decisions based on predefined algorithms, and navigate along the line.

The project successfully achieved its objective, as the line follower robot demonstrated the ability to accurately follow different types of lines on various surfaces. The robot showcased its versatility by tracking both straight and curved lines, as well as maneuvering through mazes.

# ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr**. Ramesh R,** Associate Professor, School of Electronics Engineering, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr. Susan Elias,** Dean of the School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of the Department **Dr. Mohanaprasad. K for** his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

**KALPANA RAJ SHUBHAM SHARMA DEBARGHYA BANERJEE**

# TABLE OF CONTENTS

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER** |  | **TITLE** | **PAGE** |
| **NO.** |  |  | **NO.** |
|  | **ABSTRACT** |  | **III** |
|  | **ACKNOWLEDGMENT** |  | **IV** |
|  | **TABLE OF CONTENT** |  | **V** |
|  | **LIST OF FIGURES** |  | **VI** |
| **1** | **LIST OF TABLES**  **INTRODUCTION** |  | **VII**  **8** |
|  | 1.1 OBJECTIVES |  | 8 |
|  | 1.2 SCOPE |  | 8 |

1. DESIGN/ IMPLEMENTATION 8-21
   1. [INTRODUCTION 9](#_TOC_250006)
   2. DESIGN APPROACH 10-13
   3. OVERVIEW OF SOFTWARE 14
   4. SPECIFICATION OF COMPONENTS 14-21
2. RESULT AND ANALYSIS/TESTING 22
   1. WORK DONE 22
   2. ANALYSIS OF RESULTS 22
3. CONCLUSION AND FUTURE ENHANCEMENT 23
   1. CONCLUSION 23
   2. FUTURE ENHANCEMENT 23

|  |  |  |
| --- | --- | --- |
| **5** | **APPENDIX** | **24-27** |
| **6** | **REFERENCES** | **27** |
| **7** | **BIO- DATA** | **28** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure** | **Title** | **Page** |
| **No.** |  | **No.** |
| 2.1 | LINE FOLLOWER ROBOT IN FACTORY | 9 |
| 2.2 | ROBOT CONNECTIONS WITH DIFFERENT  COMPONENTS | 11 |
| 2.3 | ALGORITHM FOR LINE FOLLOWER ROBOT | 12 |
| 2.4 | FLOW CHART OF LINE FOLLOWER ROBOT | 13 |
| 2.5 | ASSEMBLED LINE FOLLOWER ROBOT | 13 |
| 2.6 | ARDUINO UNO | 14 |
| 2.7 | INFRARED SENSOR | 15 |
| 2.8 | BATTERY OPERATED MOTOR | 16 |
| 2.9 | L298N MOTOR DRIVER | 17 |
| 2.10 | JUMPER WIRES | 18 |
| 2.11 | 9V BATTERY | 18 |
| 2.12 | MOTOR WHEEL | 19 |
| 2.13 | CASTOR WHEEL | 20 |
| 2.14 | ROBOT CAR CHASSIS | 20 |
| 3.2 | LINE FOLLOWER ROBOT GOING ALONG THE CURVED LINE | 22 |

# LIST OF TABLES

**Table No.**

# Title Page

**No.**

* 1. LIST OF FUTURE IMPROVEMEN 23

# PURPOSE

**CHAPTER 1 INTRODUCTION**

The purpose of this project is to design, develop, and implement a line follower robot using Arduino. The project aims to showcase the application of Arduino microcontroller in robotics and automation.

By creating a line follower robot, the project intends to demonstrate the integration of sensors, motors, and programming to achieve autonomous line tracking capabilities. The purpose also includes enhancing understanding and practical skills in electronics, programming, and mechanical design.

# SCOPE

The scope of this project includes documenting the entire process of building a line follower robot using Arduino. It encompasses the design and construction of the robot's mechanical structure, selection and integration of electronic components, development of the Arduino programming code, and testing and calibration procedures.

The scope also involves discussing the challenges faced during the project and providing suggestions for future improvements. The project report aims to serve as a comprehensive guide for individuals interested in learning about line follower robots, Arduino-based projects, and robotics in general.

# CHAPTER 2

**DESIGN AND IMPLEMENTATION**

# INTRODUCTION

A line follower robot is a type of autonomous robot that can follow a line drawn on a surface. The robot uses sensors to detect the line and then controls its movement accordingly. Line follower robots are often used in educational settings to teach students about robotics and programming. They can also be used in industrial applications, such as in warehouses and factories as shown in figure 2.1.



FIGURE 2.1: Line follower Robot in a factory

In this project, I built a line follower robot using an Arduino microcontroller. The robot uses four infrared sensors to detect the line. The sensors are connected to the Arduino, which uses a simple algorithm to control the movement of the robot. The robot is able to follow a line on a variety of surfaces, including black lines on white paper, white lines on black paper, and even curved lines.

The purpose of this project was to learn about the basics of programming, IOT and digital signal processing. I also wanted to build a robot that could be used for educational purposes. The robot that I built is simple to build and program, and it can be used to teach students about line following, sensors, and microcontrollers.

# DESIGN APPROACH

The design of the line follower robot was based on the following principles:

* + - The robot should be able to follow a line on a variety of surfaces, including black lines on white paper, white lines on black paper, and even curved lines.
    - The robot should be simple to build and program.
    - The robot should be affordable to build.

The robot was designed using the following components:

* + - Arduino Uno microcontroller
    - Infrared sensors - 2
    - BO motors - 2
    - L298N motor driver-1
    - Jumper wires
    - 12V battery/ 9V battery
    - Motor Wheels 2
    - Castor Wheel 1
    - Hobby Robot Chassis – 1

The Robot design connections are shown in the fig. 2.2 given below:

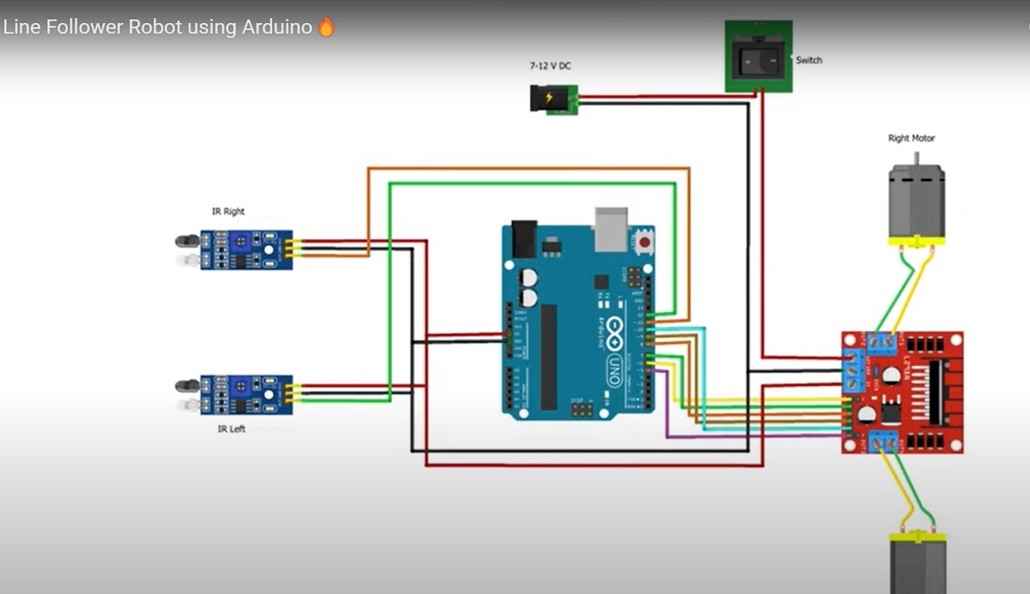


FIGURE 2.2: Robot Connections with different components

## Algorithm

The algorithm for the line follower robot is as follows:

1. The robot starts by moving forward.
2. The infrared sensors detect the line.
3. The Arduino microcontroller compares the sensor readings to determine if the robot is on the line.
4. If the robot is on the line, the microcontroller continues to move the robot forward.
5. When both the sensors sees black, the robot moves forwards
6. When the “left” Sensor sees white and “Right” sensor sees black, the robot moves right.
7. When the “Right” Sensor sees white and “Left” sensor sees black, the robot moves left.
8. When both the sensor senses white, the robot stops.
9. If the robot is not on the line, the microcontroller adjusts the speed of the motors to keep the robot on the line.
10. Steps 2-7 are repeated continuously until the robot reaches the end of the line.

The algorithm is simple to understand and implement. It is also very efficient, which allows the robot to follow a line quickly and accurately, the algorithm and flow chart is shown in the figure 2.3 and figure 2.4.

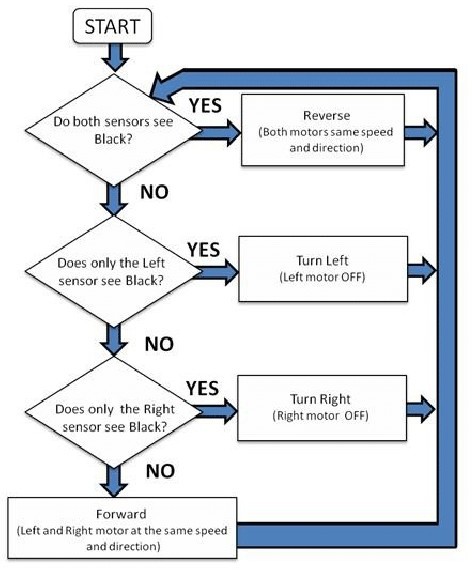


FIGURE 2.3: Algorithm for line follower Robot

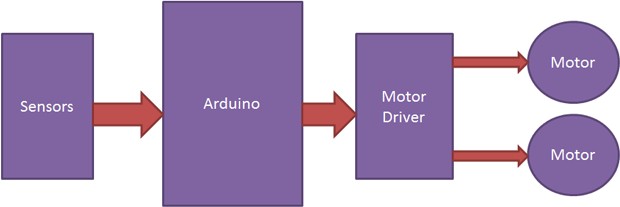


FIGURE 2.4: Flow Chart of Line Follower Robot

The assembled design of the line follower robot is made from acrylic chassis. The sensors are mounted on the chassis in a way that they are able to detect the line. The motors are connected to the chassis and the Arduino microcontroller, which is programmed with the algorithm that controls the movement of the robot. The robot is able to follow a line on a variety of surfaces, including black lines on white paper, white lines on black paper, and even curved lines. The robot is also able to follow more complex paths, such as mazes. The final design of the Line Follower Robot is show in the figure 2.5.

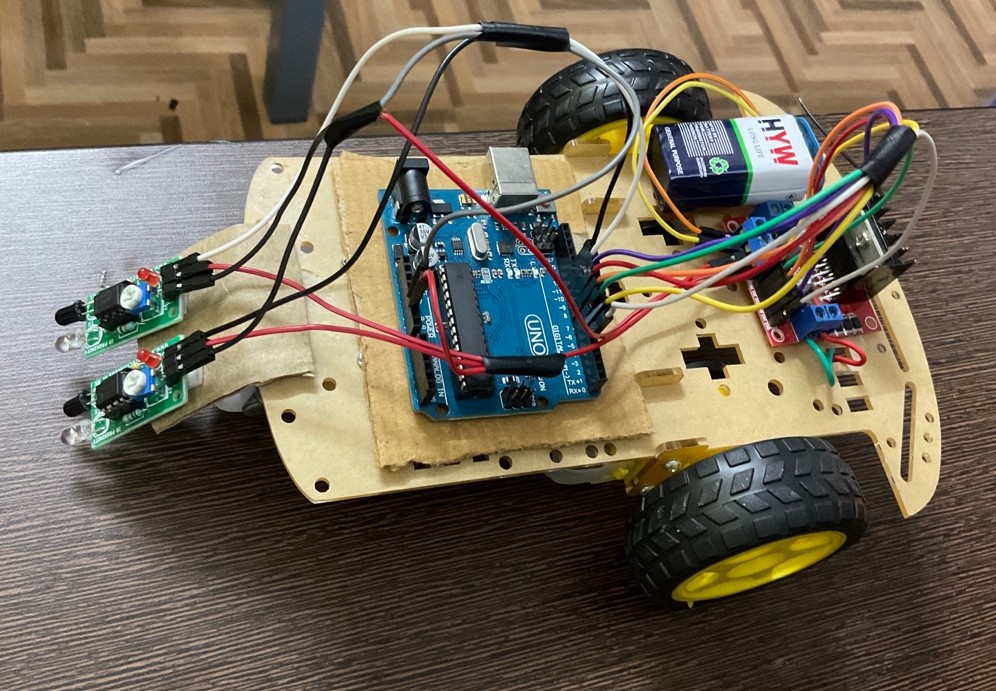


FIGURE 2.5: Assembled Line follower Robot

# OVERVIEW OF SOFTWARE

The Arduino IDE is a free and open-source integrated development environment (IDE) used to write and upload programs to Arduino boards. The IDE is available for Windows, macOS, and Linux. The Arduino IDE has a simple and easy-to-use interface. It includes a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus.

To code the line follower robot, we used the Arduino IDE and the code provided in the project. The code contains the logic for the movement of the robot, including its turning mechanism, speed of the wheels, direction of the motor, and response time. The code also uses signal processing to detect the line and control the movement of the robot.

In short, the code for the line follower robot is a simple and effective way to control the movement of the robot. The code is easy to understand and modify, and it can be used to create a variety of line follower robots.

# SPECIFICATION OF COMPONENTS

Here is the list of components used in building the Line Follower Robot with their specifications:

1. Arduino Uno microcontroller

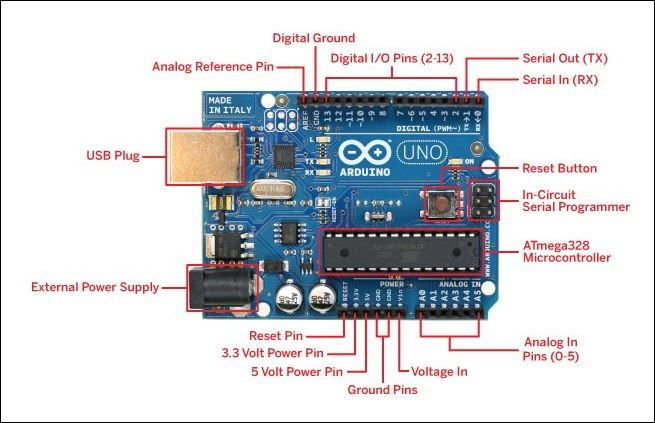


FIGURE 2.6: Arduino Uno

## Specifications:

* + Microcontroller: ATmega328P
  + Operating Voltage: 5V
  + Input Voltage (recommended): 7-12V
  + In-out Voltage (limit): 6-20V
  + Digital I/O Pins: 14 (of which 6 provide PWM output)
  + PWM Digital I/O Pins: 6
  + Analog Input Pins: 6
  + DC Current per I/O Pin: 20 mA
  + DC current for 3.3V Pin: 50 mA
  + Flash Memory: 32 KB (ATmega328P) of which 0.5 KB used by bootloader
  + SRAM: 2 KB (ATmega328P)
  + EEPROM: 1 KB (ATmega328P)
  + Clock Speed: 16 MHz
  + LED\_BUILTIN: 13
  + Length: 68.6 mm
  + Width: 58.4 mm
  + Weight: 25 g

1. Infrared sensors – 2

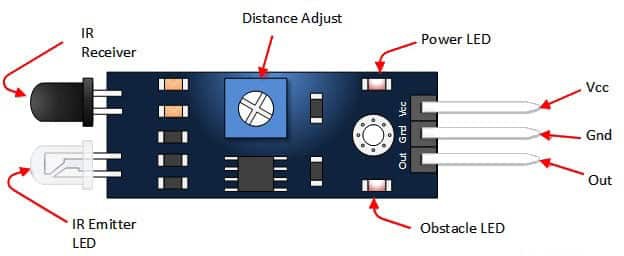


FIGURE 2.7: Infrared Sensor

## Specifications:

* + Board Size: 3.2 x 1.4cm
  + Working voltage: 3.3 to 5V DC
  + Operating voltage: 3.3V: ~23 mA, to 5V: ~43 mA
  + Detection range: 2cm – 30cm (Adjustable using potentiometer)
  + Active output level: The output is “0” (Low) when an obstacle is detected

1. BO motors – 2

FIGURE 2.8: Battery Operated Motor

## Specifications:

* + Operating Voltage (VDC): 3~12
  + Shaft Length (mm): 8.5
  + Shaft Diameter (mm): 5.5 (Double D-type)
  + No Load Current: 40-180mA.
  + Rated Speed (After Reduction): 100 RPM
  + Rated Torque: 1 Kg-cm

1. L298N motor driver-1

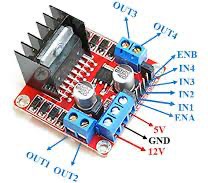


FIGURE 2.9: L298N motor driver

## Specifications:

* + Driver Model: L298N 2A
  + Driver Chip: Double H Bridge L298N
  + Motor Supply Voltage (Maximum): 46V
  + Motor Supply Current (Maximum): 2A
  + Logic Voltage: 5V
  + Driver Voltage: 5-35V
  + Driver Current:2A
  + Logical Current:0-36mA
  + Maximum Power (W): 25W
  + Current Sense for each motor
  + Heatsink for better performance
  + Power-On LED indicator

1. Jumper wires



FIGURE 2.10: Jumper Wires

## Specifications:

* + Current: 4-20 mA
  + Voltage: 12 V
  + Rated Pressure: 25 kPA
  + Pitch: 2.54 mm
  + Cable Length: 20 cm - 8 Inch

1. 12V battery/ 9V battery

FIGURE 2.11: 9V battery

## Specifications:

* + Model Number: 9V 6F 22
  + Battery Type: Zinc Carbon
  + Size: 6F22 006P
  + Jacket: Metal
  + Single Battery Dimensions (mm): L- 26. 5, H - 48. 5, W - 17. 5 (Max)
  + Nominal Voltage(V) :9V
  + Discharge Resistance(Ω): 620
  + Cut-off Voltage(V): 5.4

1. Motor Wheels 2



FIGURE 2.12: Motor Wheel

## Specifications:

* + Diameter: 65 mm
  + Width: 28 mm
  + The material of Wheel: High-strength plastic
  + The material of Tyre: Rubber
  + Color: Yellow

1. Castor Wheel - 1



FIGURE 2.13: Castor Wheel

## Specifications:

* + Size: 4 inch
  + Material: Nylon
  + Color: White
  + Tread Width: 50 mm
  + Weight capacity: Approx 2 Kg

1. Hobby Robot Chassis – 1

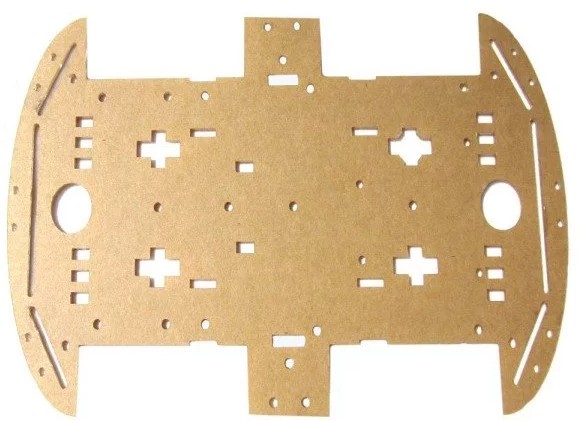


FIGURE 2.14: Robot Car chassis

## Specifications:

* + Length: about 25.3cm
  + Width: about 14.8cm
  + Material: Acrylic
  + Color: Transparent

# CHAPTER 3

**RESULT AND ANALYSIS/TESTING**

# WORK DONE

The following work was done for the line follower robot project:

* + - The design of the robot was created.
    - The components for the robot were purchased.
    - The robot was assembled.
    - The code for the robot was written.
    - The robot was tested.

# ANALYSIS OF RESULTS

The robot was able to follow a line on a variety of surfaces, including black lines on white paper, white lines on black paper, and even curved lines as shown in figure 3.2. The robot can also follow more complex paths, such as mazes.

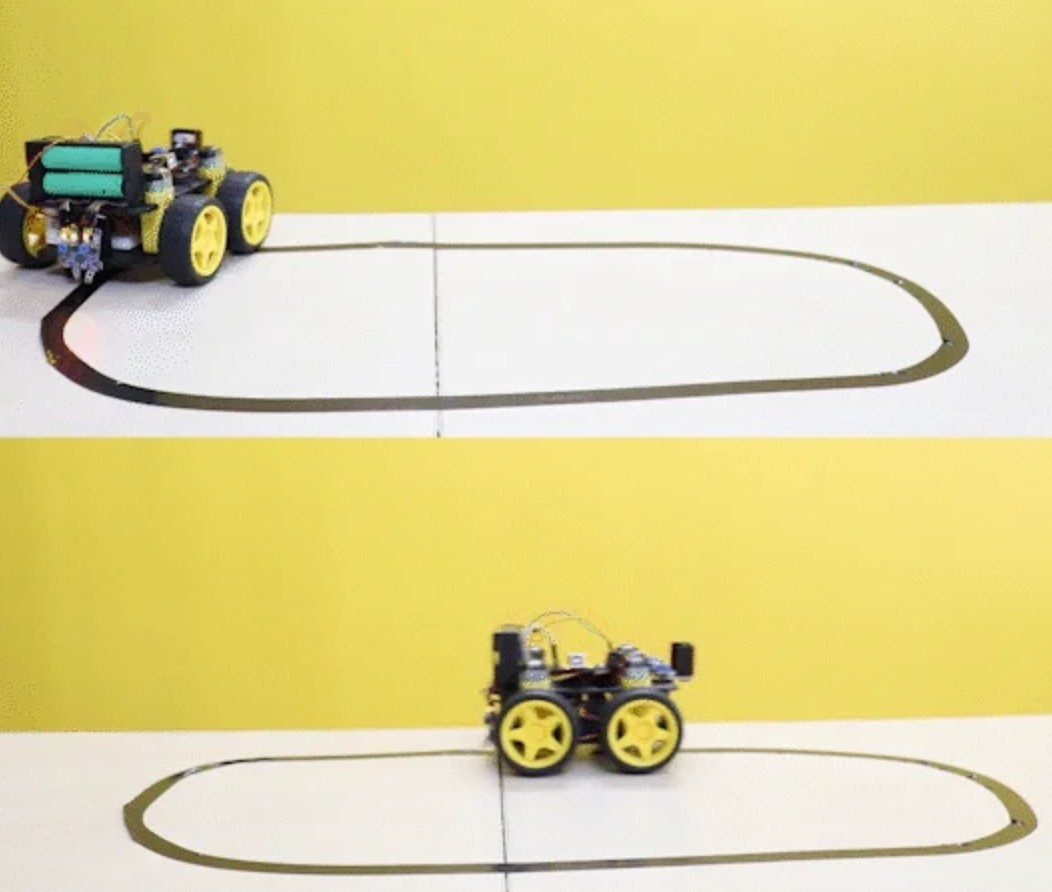


FIGURE 3.2: Line follower robot going along the curved line

# CHAPTER 4

**CONCLUSION AND FUTURE ENHANCEMENT**

# CONCLUSION

The line follower robot project was a success. The robot was able to follow a line on a variety of surfaces, and it was able to follow more complex paths. The project was a valuable learning experience, and it helped us gain more knowledge of working of sensors for signal processing and.

There are a few areas where the project could be improved. The design of the robot could be made more compact and efficient. The code for the robot could be made more modular and easier to understand. The robot could also be equipped with more sensors to improve its performance.

The overall cost of making this project was Rs 1700/-

# FUTURE ENHANCEMENT

The line follower robot project has a lot of potential for improvement in both the software and hardware design. This could be very useful for future implementation and work. The table 4.2 lists some improvement which can we implement.

|  |  |  |
| --- | --- | --- |
| **Area** | **Enhancement** | **Region of Improvement** |
| Design | Make the robot more compact and efficient. | Use smaller components, a more powerful motor, and stronger materials. |
| Code | Make the code more modular and easier to understand. | Break the code down into smaller,  more manageable sections and use more descriptive variable names and comments. |
| Sensors | Equip the robot with more sensors. | Add a camera, a range sensor, and a touch sensor. |
| Intelligence | Add intelligence to the robot. | Give the robot the ability to learn the shape of the line, avoid obstacles, and navigate more complex paths. |

TABLE 4.2: List of Future improvement

# APPENDIX

## CODE

The code used in the project is given below:

#define IR\_SENSOR\_RIGHT 11

#define IR\_SENSOR\_LEFT 12

#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);

rotateMotor (0, 0);

}

void loop ()

{

int rightIRSensorValue = digitalRead (IR\_SENSOR\_RIGHT); int leftIRSensorValue = digitalRead (IR\_SENSOR\_LEFT);

//If none of the sensors detects black line, then go straight

if (rightIRSensorValue == LOW && leftIRSensorValue == LOW)

{

rotateMotor (MOTOR\_SPEED, MOTOR\_SPEED);

}

//If right sensor detects black line, then turn right

else if (rightIRSensorValue == HIGH && leftIRSensorValue == LOW)

{

rotateMotor (-MOTOR\_SPEED, MOTOR\_SPEED);

}

//If left sensor detects black line, then turn left

else if (rightIRSensorValue == LOW && leftIRSensorValue == HIGH)

{

rotateMotor (MOTOR\_SPEED, -MOTOR\_SPEED);

}

//If both the sensors detect black line, then stop 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));

}

The code used in the line follower robot project follows all the algorithms and the robot works perfectly fine. The code was written in the Arduino IDE and uses signal processing to detect the line and control the movement of the robot. The code is simple to understand and modify, and it can be used to create a variety of line follower robots.

The code has been tested on a variety of surfaces, including black lines on white paper, white lines on black paper, and even curved lines. The robot has been able to follow the line on all of these surfaces without any problems.

# REFERENCES

These are the following links which were referred during the making of this project: https://circuitdigest.com/microcontroller-projects/arduino-uno-line-follower-robot

https://circuitdigest.com/microcontroller-projects/line-follower-robot-using-arduino

https://quartzcomponents.com/blogs/electronics-projects/line-follower-robot-using-arduino

https://projecthub.arduino.cc/lightthedreams/line-following-robot-34b1d3

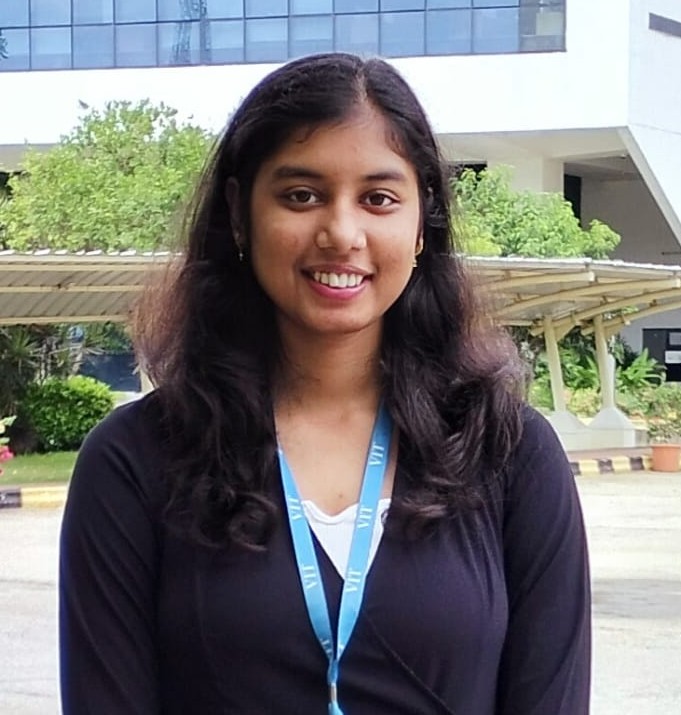
# BIODATA

NAME: **Shubham Sharma**

MOBILE NUMBER: **8700087743**

## E-MAIL: [shubham.sharma2021d@vitstudent.ac.in](mailto:shubham.sharma2021d@vitstudent.ac.in)

PERMANENT ADDRESS: **Orchid Island, Haryana**

NAME: **Kalpana Raj**

MOBILE NUMBER: **8618820047**

## E-MAIL: [kalpana.raj2021@vitstudent.ac.in](mailto:kalpana.raj2021@vitstudent.ac.in)

PERMANENT ADDRESS: **Vidyanagar township, Karnataka**

NAME: **Debarghya Banerjee**

MOBILE NUMBER: **8250485713**

## E-MAIL: [debarghya.banerjee2021@vitstudent.ac.in](mailto:debarghya.banerjee2021@vitstudent.ac.in)

PERMANENT ADDRESS: **Deshbandhu Nagar, West Bengal**