VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi - 590018

###### A Mini project report on

**“Development Of Line Following Robot”**

*Submitted in partial fulfilment for the award of the degree of*

***Bachelor of Engineering***

***In***

***Electronics and Communication Engineering***

***Submitted by***

|  |  |
| --- | --- |
| **Varsha. V** | **1AT20EC159** |
| **Sneka. P** | **1AT20EC142** |
| **Sanjay. S** | **1AT20EC133** |

***Under the Guidance of***

**Dr. Mangala Gowri**

**Associate Professor Department of ECE**



Atria Institute of Technology

Department of Electronics and Communication Engineering

2023-2024



**Atria Institute of Technology**

**Affiliated to Visvesvaraya Technological University Anand Nagar, Hebbal, Bangalore-560024**

**Department of Electronics and Communication Engineering**

## CERTIFICATE

Certified that the project work entitled ***“Development Of Line Following Robot***”

carried out by

|  |  |
| --- | --- |
| Varsha. V | 1AT20EC159 |
| Sneka. P | 1AT20EC142 |
| Sanjay. S | 1AT20EC133 |

Bonafied students of **ATRIA INSTITUTE OF TECHNOLOGY,** in partial fulfilment for the award of **Bachelor of Engineering in Electronics and Communication** under Visvesvaraya Technological University, Belagavi during the year 2023-2024. It is certified that they have completed the project satisfactorily at Atria Institute of Technology.

Signature of Principal

**Dr.** Principal

Atria Institute of Technology

Signature of HOD

**Dr. Arun Balodi**

Professor and Head Dept. of ECE, AIT

Signature of Guide **Dr. Mangala Gowri** Associate Professor Dept. of ECE, AIT

**External VIVA**

##### Internal Examiner External Examiner

Signature: Signature:

Name: Name:

Date: Date:

## ABSTRACT

Line following robots have gained significant attention in various industries

and applications, ranging from warehouse automation to educational robotics.

This abstract presents a comprehensive overview of an autonomous line following

robot, highlighting its key components, functionalities, and potential applications.

Here we are developing line following robot by implementing smart dustbin which

can sense any dry waste which allows the user to dump the waste.

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without acknowledging those who made it possible.

We wish to express our profound and sincere gratitude to **Dr.** , Principal, Atria Institute of Technology, for providing all the help.

We express our sincere gratitude to **Dr. Arun Balodi**, Head, Department of Electronics and Communication, Atria Institute of Technology, for the support extended to us.

We express our sincere gratitude to the Project co-coordinators, **Dr. Mangala Gowri**, Associate Professor, Department of Electronics and Communication, Atria Institute of Technology, for the support extended to us.

We also express our sincere gratitude to our Internal guide, **Prof. Poornima B, Prof. Rajagopalan Nadathur and Prof. Gayathri Joshi** for being patient and taking out her precious time to help us in more ways than one.

Finally, we also thank the entire staff of the Dept. of Electronics and Communication Engineering, friends and family for their valuable suggestions and encouragement in all means during the course of the project.

**Varsha. V**

**Sneka. P**

**Sanjay. S**

## DECLARATION

We the students of final year, Bachelor of Engineering, Electronics and Communication Engineering, Atria Institute of Technology, Bangalore-560024, hereby declare that the project work entitled ***“Development Of Line Following Robot”*** has been carried out at **ATRIA INSTITUTE OF TECHNOLOGY** under the guidance of **Dr. Mangala Gowri,** Associate Professor, Department of Electronics and Communication Engineering, Atria IT, Bangalore.

We declare that the work submitted in this report is our own and has not been previously submitted for the fulfilment of the B.E degree at the Visvesvaraya Technological University, Belagavi or any other Institution/University.

**Place: Bengaluru Varsha. V**

**Sneka. P**

**Sanjay. S**

### Table of Contents

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **1** | **INTRODUCTION** | 2 |
|  | 1.1. Scope of Work | 4 |
|  | 1.2. Problem Statement | 4 |
|  | 1.3. Objectives | 4 |
|  | 1.4. Literature Survey | 5 |
| **2** | **SOFTWARE USED** | 8 |
|  | 2.1. Software Requirements | 8 |
|  | 2.1.1 Arduino IDE | 9 |
| **3** | **HARDWARE USED** | 10 |
|  | 3.1. Hardware Requirements | 11 |
|  | 3.1.1 Brief description on hardware | 12 |
| **3** | **METHODOLOGY** | 11 |
|  | 3.1. Introduction | 11 |
|  | 3.2. Block Diagram | 12 |
| **4** | **IMPLEMENTATION** | 15 |
| **5** | **RESULTS** | 22 |
| **6** | **APPLICATIONS, ADVANTAGES & DISADVANTAGES** | 28 |
|  | 6.1. Applications | 28 |
|  | 6.2. Advantages | 29 |
|  | 6.3. Disadvantages | 30 |
| **7** | **CONCLUSION & FUTURE SCOPE** | 32 |
|  | **REFERENCES** | 34 |

I

### TABLE OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure no.** | **Title** | **Page no.** |
| 1.1 | Line following robot | 2 |
| 1.2 | Smart dustbin | 3 |
| 2.1 | Arduino IDE | 8 |
| 3.1 | Block Diagram | 12 |
| 4.1.1 | Image 1 of line following robot | 17 |
| 4.1.2 | Code for an line following robot | 18 |
| 4.1.3 | Output of line following robot | 19 |
| 4.2.1 | Image 2 of smart dustbin | 20 |
| 4.1.2 | Code for an smart dustbin | 21 |
| 4.2.3 | Output of smart dustbin | 22 |
| 5.1 | Output of line following robot and smart dustbin | 24 |
| 6.1 | Application | 25 |
| 6.2 | Advantages | 27 |
| 6.3 | Disadvantages | 28 |
| 7.1 | Conclusion | 31 |
| 7.2 | Future Scope | 31 |

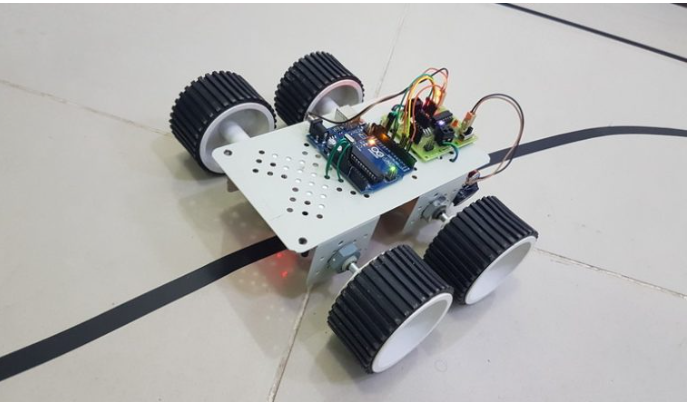
II

# CHAPTER - 1

Dept, ECE, AIT 2023-2024 Page | 1

**Chapter 1: INTRODUCTION**

Line following robots have emerged as fascinating and versatile machines that can navigate autonomously along a pre-determined path. These robots utilize a combination of sensors, algorithms, and control mechanisms to detect and follow lines, opening a wide range of applications in various fields, from industrial automation to educational robotics. The concept of line following robots revolves around the idea of mimicking human-like behaviors in machines. Humans often use visual cues to follow paths or road markings, and line following robots replicate this behavior by utilizing sensors to detect lines and adjust their movements accordingly.

This capability allows them to perform tasks efficiently and accurately in structured environments. Line following robots typically employ optical sensors, such as infrared or reflective sensors, mounted on their undersides. These sensors continuously scan the surface beneath the robot and detect the presence or absence of a line. By analyzing the sensor data, the robot's control system can make real-time decisions and adjustments to stay on the desired path. The control algorithms used in line following robots play a vital role in their performance.

**Fig 1.1**

Common techniques include proportional-integral-derivative (PID) control, where the robot adjusts its position and orientation based on the deviation from the line. Other approaches, such as fuzzy logic or machine learning algorithms, may be employed to enhance the robot's adaptability to different line patterns and environmental condition.

Through line following robot we can develop smart dustbin which represents a significant advancement in waste management systems, leveraging technology to enhance efficiency, cleanliness, and sustainability. Smart dustbins incorporate various sensors, connectivity, and intelligent features to revolutionize the way waste is collected, monitored, and managed in both urban and residential environments.

Traditional waste management methods often rely on manual processes, leading to inefficiencies, overflow, and inadequate maintenance. Smart dustbins address these challenges by integrating innovative technologies to streamline the waste disposal process and promote a cleaner and greener environment. At its core, a smart dustbin is equipped with sensors that can detect the fill level of the bin. These sensors can employ different technologies such as ultrasonic, infrared, or weight sensors to accurately measure the amount of waste present. This real-time monitoring allows waste management personnel or authorities to optimize collection routes and schedules, ensuring bins are emptied promptly when needed, thus reducing unnecessary trips and associated costs.

**Fig 1.2**

##### Scope of Work:

We demonstrate our project development of line following robot using

Arduino software/hardware. In this project, it can be developed by making the robot

carry smart dustbin. Likewise, this robot can be represented in four different methods

such a robot which follows the line, remote controlled robot with the help of MP3

player IR remote, voice-controlled robot and obstacle detection and avoider.

##### 

##### Problem Statement:

Before the invention of line following robots, there were several challenges and limitations in various fields that required automated systems. They are

* Manual labor
* Lack of precision
* Limited flexibility
* Safety risk

These robots helped improve efficiency, reduce cost, enhance safety, and open new possibility for automation and research in various fields.

##### Aim:

To implement the development of line following robot and Dut with the help of Arduino.

* 1. **Objectives:**
* Creating a circuit design for line following robot and dustbin.
* Using the reference paper and reviews for the design, components and

programming required.

* Implementation of the programme for line following robot and dustbin

using Arduino hardware and Arduino software.

* Building the circuit with respect to the circuit diagram.
* Verifying the project.

##### Literature Survey:

1. ***“Line Following Robot Using Arduino for Hospital”, Jagruti Chaudhari, Asmita Desai, S. Gavarskar, ICCT, IEEE 2019.***

Paper got published in 2019

Abstract: This paper describes the line following robot using Arduino

for surveying, inspecting, and enhancing the transportation of necessary

materials inside the healthcare institutions, industries also. The proposed

system spot the black path and proceed in its direction on to the ground.

This system eases the work of material conveyance as well as minimizes

the manpower. This technology targets on the secured, punctual and

constructing transportation of goods. This paper aims to implement

controlled movement of robot by tuning control parameters and thus achieve

better performance. This robot is predominantly design to proceed in

a predefined path. To locate this path two sensors are used. Robots like this

are mainly used in industrial plants comprising of pick and place facility.

This robot carries components from desired source to destination by

following fixed path. Recently lot of research has been done to empower

the automation in hospitals as well in industries. This robot is made to supply

the essential goods such injections, medicine, etc. This paper is divided

into hardware and software modules.

1. “Vision-Based Line Following Robot In Webots” Alfian Maarif, Aninditya Anggari Nuryono, Iswanto, FORTEI, IEEE 2020.

Paper got published in 2020

Abstract: Line following robot is one of the popular robots commonly used for educational purposes. The most widely used sensors for the robots are photoelectric sensors. However, it is irrelevant, along with the development of autonomous vehicles and robotic vision. Robotic vision is a robot that can obtain information through image processing by the camera. The camera installed on the line following robot aims to detect image-based lines and to navigate the robot to follow the path. This paper proposed a method of image preprocessing along with its robot action for line-following robots. This includes image preprocessing such as dilation, erosion, Gaussian filtering, contour search, and centerline definition to detect path lines and to determine the proper robot action. The implementation of the robot is simulated using Webots simulator. OpenCV and Python are utilized to design line detection systems and robot movements. The simulation result shows that the method is implemented properly, and the robot can follow a different type of path lines such as zigzag, dotted, and curved line. The resolution of the cropped-image frame is the fundamental parameter in detecting path lines.

***[3] “Line following robots on factory floors”, Significance and***

***Simulation study using Coppelia Sim. Saharsh Oswal1 and***

***Saravanakumar, School of Mechanical Engineering, Vellore Institute***

***of Technology, India 2021.***

Paper got published in 2021

Abstract: Line following robots has been around for many years now.

Research and development of these robots have been keenly studied and

examined for many industrial applications. Line following robots can

provide maximum effectiveness to in-house transportation of jobs inside a

factory, which is traditionally done by forklifts and different types of cranes.

These traditional methods are high maintenance and extremely unsafe for the

employees. This paper presents the design, assembly and dynamic simulation

of a line following robot integrated with a proximity sensor for collision

avoidance and vision sensors for line tracking. The model is designed using

Autodesk Inventor 2018 and the assembly and simulation is carried out using

Coppelia Sim software. This simulation study intends to provide significance

of the implementation of such robots on a factory floor, which can cover a

path of distance 10 meters in approximately 8 seconds.

# CHAPTER - 2

**Chapter 2: SOFTWARE USED**

##### Software Requirement:

* + 1. Arduino IDE

**2.1.1. Arduino IDE:**

****

The Arduino IDE (Integrated

Development Environment) is a software

application specifically designed to program

Arduino microcontrollers. It provides a

user-friendly interface that simplifies the

process of writing, compiling, and uploading

code to Arduino boards. The Arduino IDE

supports multiple platforms, making it

accessible to a wide range of users, from

beginners experienced developers.

**Fig 2.1**

Features of the Arduino IDE:

1. Code Editor: The IDE includes a text editor where users can write their Arduino

code.

2. Library Manager: The IDE includes a library manager that allows users to easily

browse, install, and update libraries.

3. Serial Monitor: The Serial Monitor is a built-in feature that allows users to

communicate with the Arduino board through the serial port.

4.Board Manager: The IDE's Board Manager provides a simple way to install and

manage different Arduino board packages.

5. Sketch Compilation and Upload: The IDE compiles the Arduino code into

machine-readable instructions that can be understood by the microcontroller. It

also provides a straightforward interface to upload the compiled code to the Arduino

board via a USB connection. This process is commonly referred to as "sketch

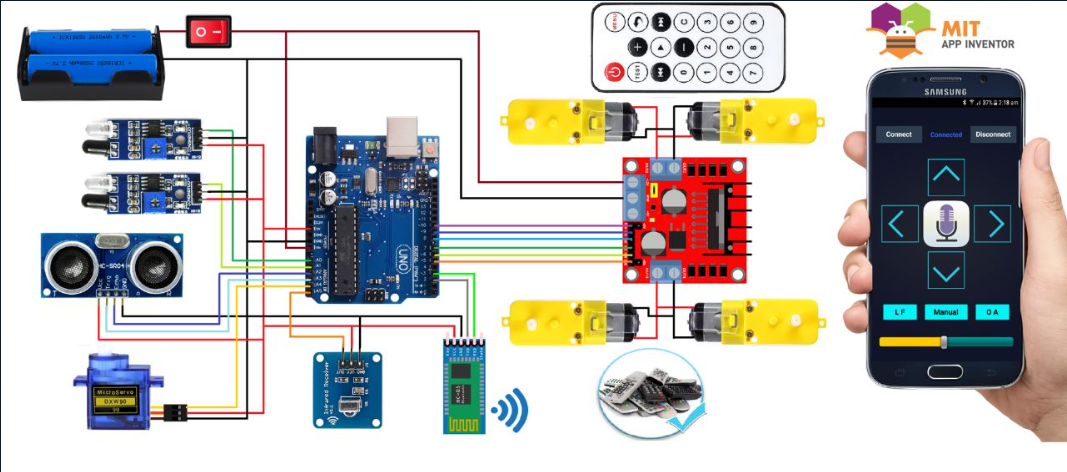
compilation and upload."

# CHAPTER - 3

**Chapter 3: HARDWARE USED**

##### Hardware Requirements:

1. 5mm Acrylic Sheet 20cm x 12cm
2. DC Gear Motor x 4,
3. Arduino UNO x 2,
4. IR Sensor x 2,
5. L298 Motor Driver,
6. HC-05 Bluetooth Module x 2,
7. IR Receiver Module,
8. MP3 Player IR Remote,
9. sg90 servo motor,
10. Ultrasonic Sensor Holder,
11. Ultrasonic Sensor hc-sr04,
12. 4Pcs Smart Robot Car Tyres Wheels,
13. Male to Female jumper Wires,
14. On/Off Switch,
15. 18650 Battery Holder – 2 Cells ,
16. 18650 Battery Cell 3.7V x 4,
17. dustbin,
18. 9V battery.

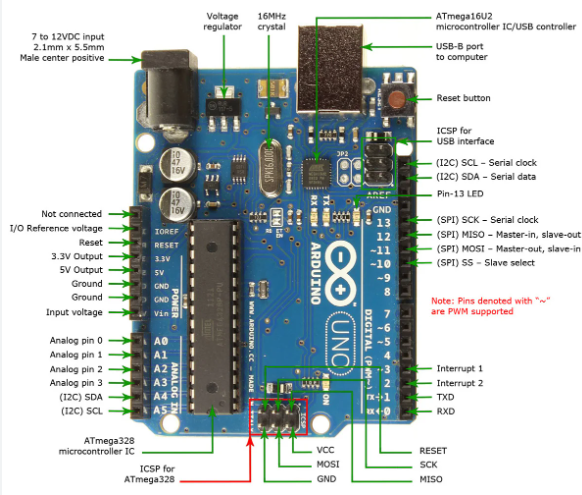


**Fig 3.1**

##### Brief description of hardware:

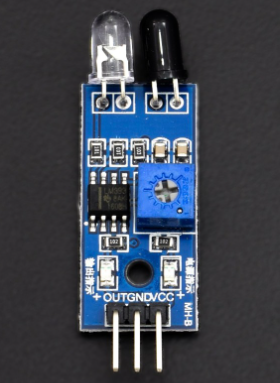
From the above fig 3.1 is used to build line following robot which consists of

1.Arduino uno:

 The Arduino Uno is a widely used microcontroller board based on the ATmega328P. It offers 14 digital I/O pins, 6 PWM output pins, and 6 analog input pins. It supports USB and UART communication interfaces, and can be powered via USB or an external source. Programming is done through the Arduino IDE using simplified C/C++ language. The board is expandable with various shields, and its open-source nature allows customization. The Uno is beginner-friendly and popular for prototyping and DIY projects due to its simplicity and extensive community support.

**Fig 3.2.1**

2.IR sensor:

IR sensors are commonly used in line-following robots to detect and track lines on the ground. They consist of an IR emitter and receiver, detecting infrared light reflected from the line. IR sensors offer benefits such as accurate line detection, compactness, affordability, easy interfacing with microcontrollers, fast response times, versatility for different line types, and robustness against ambient light. These qualities make IR sensors an advantageous component for line-following robots, enabling precise line tracking and effective navigation.

**Fig 3.2.2**

3.IR receiver module and mp3 IR remote:



**Fig 3.2.3**

An IR receiver is a component that detects infrared signals and converts them into electrical signals. The receiver consists of an IR sensor, which detects the infrared light emitted by an IR remote control, and a demodulator, which extracts the modulated data from the received signal.

An IR remote, also known as an infrared remote control, is a handheld device that transmits infrared signals to control electronic devices wirelessly. It typically consists of buttons or keys that are associated with specific functions or commands.

4.L298 motor driver and dc motor:

 A DC motor is an electrical device that converts electrical energy into mechanical motion through the interaction of magnetic fields. It consists of a stationary stator and a rotating rotor. DC motors find applications in robotics, automation, and electric vehicles.

A motor driver is an electronic circuit or module used to control and regulate the operation of motors. It provides the necessary power, voltage, and current to drive the motor effectively. Motor drivers are responsible for controlling speed, direction, and torque, and they offer protection features.

**Fig 3.2.4**

5.Bluetooth module:



The HC-05 Bluetooth module is a widely used module for wireless communication. In the

**Fig 3.2.5**

context of a line-following robot, the HC-05

module enables remote control and serves as an interface between the robot and a controlling

device. It allows commands to be sent wirelessly from a smartphone or computer, providing

flexibility and convenience.

Overall, the HC-05 Bluetooth module enhances the functionality and control options

of a line following robot.

6.Servo motor:



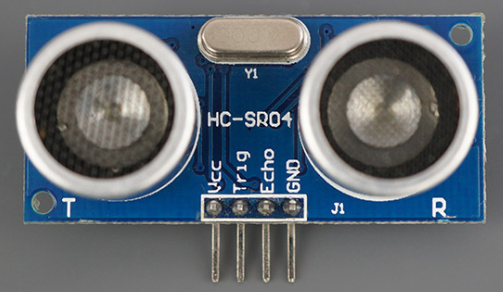
The SG90 servo motor is a small and lightweight motor

**Fig 3.2.6**

that offers precise control over angular motion. In a line-following robot, it is commonly used for steering control. The servo motor allows for accurate adjustments to the robot's direction based on the line being followed. It is compatible with microcontrollers, compact in size, and power-efficient. These qualities make the SG90 servo

motor a suitable choice for integration into line-following robots, providing precise positioning and efficient steering control.

7.Ultrasonic sensor:



The ultrasonic sensor is a device that uses

high-frequency sound waves to detect objects

and measure distances. In a line-following robot, the ultrasonic sensor plays a crucial role in

obstacle avoidance, line detection and

correction, and providing environmental sensing capabilities. It helps the robot navigate around

**Fig 3.2.7**

obstacles, detect deviations from the line, and enhances its awareness of the

surroundings. The sensor's versatility and ease of integration make it a valuable component for improving the navigation and safety of line-following robots.

# CHAPTER 4

### Chapter 4: IMPLEMENTATION

##### Image 1: Line following robot

|  |
| --- |
|  |

**Fig4.1.1:Development of line following robot**

**4.1.1. Code for an development of line following robot**

#include <SoftwareSerial.h>

SoftwareSerial BT\_Serial(2, 3); // RX, TX

#include <IRremote.h>

const int RECV\_PIN = A5;

IRrecv irrecv(RECV\_PIN);

decode\_results results;

#define enA 10//Enable1 L298 Pin enA

#define in1 9 //Motor1  L298 Pin in1

#define in2 8 //Motor1  L298 Pin in1

#define in3 7 //Motor2  L298 Pin in1

#define in4 6 //Motor2  L298 Pin in1

#define enB 5 //Enable2 L298 Pin enB

#define servo A4

#define R\_S A0 //ir sensor Right

#define L\_S A1 //ir sensor Left

#define echo A2    //Echo pin

#define trigger A3 //Trigger pin

int distance\_L, distance\_F = 30, distance\_R;

long distance;

int set = 20;

int bt\_ir\_data; // variable to receive data from the serial port and IRremote

int Speed = 130;

int mode=0;

int IR\_data;

void setup(){ // put your setup code here, to run once

pinMode(R\_S, INPUT); // declare if sensor as input

pinMode(L\_S, INPUT); // declare ir sensor as input

pinMode(echo, INPUT );// declare ultrasonic sensor Echo pin as input

pinMode(trigger, OUTPUT); // declare ultrasonic sensor Trigger pin as Output

pinMode(enA, OUTPUT); // declare as output for L298 Pin enA

pinMode(in1, OUTPUT); // declare as output for L298 Pin in1

pinMode(in2, OUTPUT); // declare as output for L298 Pin in2

pinMode(in3, OUTPUT); // declare as output for L298 Pin in3

pinMode(in4, OUTPUT); // declare as output for L298 Pin in4

pinMode(enB, OUTPUT); // declare as output for L298 Pin enB

irrecv.enableIRIn(); // Start the receiver

irrecv.blink13(true);

Serial.begin(9600); // start serial communication at 9600bps

BT\_Serial.begin(9600);

pinMode(servo, OUTPUT);

 for (int angle = 70; angle <= 140; angle += 5)  {

   servoPulse(servo, angle);  }

 for (int angle = 140; angle >= 0; angle -= 5)  {

   servoPulse(servo, angle);  }

 for (int angle = 0; angle <= 70; angle += 5)  {

   servoPulse(servo, angle);  }

delay(500);

}

void loop(){

if(BT\_Serial.available() > 0){  //if some date is sent, reads it and saves in state

bt\_ir\_data = BT\_Serial.read();

Serial.println(bt\_ir\_data);

if(bt\_ir\_data > 20){Speed = bt\_ir\_data;}

}

if (irrecv.decode(&results)) {

Serial.println(results.value,HEX);

bt\_ir\_data = IRremote\_data();

Serial.println(bt\_ir\_data);

irrecv.resume(); // Receive the next value

delay(100);

}

     if(bt\_ir\_data == 8){mode=0; Stop();}    //Manual Android Application and IR Remote Control Command

else if(bt\_ir\_data == 9){mode=1; Speed=130;} //Auto Line Follower Command

else if(bt\_ir\_data ==10){mode=2; Speed=255;} //Auto Obstacle Avoiding Command

analogWrite(enA, Speed); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1 Speed

analogWrite(enB, Speed); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2 Speed

if(mode==0){

//===============================================================================

//                          Key Control Command

//===============================================================================

     if(bt\_ir\_data == 1){forword(); }  // if the bt\_data is '1' the DC motor will go forward

else if(bt\_ir\_data == 2){backword();}  // if the bt\_data is '2' the motor will Reverse

else if(bt\_ir\_data == 3){turnLeft();}  // if the bt\_data is '3' the motor will turn left

else if(bt\_ir\_data == 4){turnRight();} // if the bt\_data is '4' the motor will turn right

else if(bt\_ir\_data == 5){Stop(); }     // if the bt\_data '5' the motor will Stop

//===============================================================================

//                          Voice Control Command

//===============================================================================

else if(bt\_ir\_data == 6){turnLeft();  delay(400);  bt\_ir\_data = 5;}

else if(bt\_ir\_data == 7){turnRight(); delay(400);  bt\_ir\_data = 5;}

}

if(mode==1){

//===============================================================================

//                          Line Follower Control

//===============================================================================

if((digitalRead(R\_S) == 0)&&(digitalRead(L\_S) == 0)){forword();}  //if Right Sensor and Left Sensor are at White color then it will call forword function

if((digitalRead(R\_S) == 1)&&(digitalRead(L\_S) == 0)){turnRight();}//if Right Sensor is Black and Left Sensor is White then it will call turn Right function

if((digitalRead(R\_S) == 0)&&(digitalRead(L\_S) == 1)){turnLeft();} //if Right Sensor is White and Left Sensor is Black then it will call turn Left function

if((digitalRead(R\_S) == 1)&&(digitalRead(L\_S) == 1)){Stop();}     //if Right Sensor and Left Sensor are at Black color then it will call Stop function

}

if(mode==2){

//===============================================================================

//                          Obstacle Avoiding Control

//===============================================================================

 distance\_F = Ultrasonic\_read();

 Serial.print("S=");Serial.println(distance\_F);

  if (distance\_F > set){forword();}

    else{Check\_side();}

}

delay(10);

}

long IRremote\_data(){

     if(results.value==0xFF02FD){IR\_data=1;}

else if(results.value==0xFF9867){IR\_data=2;}

else if(results.value==0xFFE01F){IR\_data=3;}

else if(results.value==0xFF906F){IR\_data=4;}

else if(results.value==0xFF629D || results.value==0xFFA857){IR\_data=5;}

else if(results.value==0xFF30CF){IR\_data=8;}

else if(results.value==0xFF18E7){IR\_data=9;}

else if(results.value==0xFF7A85){IR\_data=10;}

return IR\_data;

}

void servoPulse (int pin, int angle){

int pwm = (angle\*11) + 500;      // Convert angle to microseconds

 digitalWrite(pin, HIGH);

 delayMicroseconds(pwm);

 digitalWrite(pin, LOW);

 delay(50);                   // Refresh cycle of servo

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ultrasonic\_read\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

long Ultrasonic\_read(){

  digitalWrite(trigger, LOW);

  delayMicroseconds(2);

  digitalWrite(trigger, HIGH);

  delayMicroseconds(10);

  distance = pulseIn (echo, HIGH);

  return distance / 29 / 2;

}

void compareDistance(){

       if (distance\_L > distance\_R){

  turnLeft();

  delay(350);

  }

  else if (distance\_R > distance\_L){

  turnRight();

  delay(350);

  }

  else{

  backword();

  delay(300);

  turnRight();

  delay(600);

  }

}

void Check\_side(){

    Stop();

    delay(100);

 for (int angle = 70; angle <= 140; angle += 5)  {

   servoPulse(servo, angle);  }

    delay(300);

    distance\_L = Ultrasonic\_read();

    delay(100);

  for (int angle = 140; angle >= 0; angle -= 5)  {

   servoPulse(servo, angle);  }

    delay(500);

    distance\_R = Ultrasonic\_read();

    delay(100);

 for (int angle = 0; angle <= 70; angle += 5)  {

   servoPulse(servo, angle);  }

    delay(300);

    compareDistance();

}

void forword(){  //forword

digitalWrite(in1, HIGH); //Right Motor forword Pin

digitalWrite(in2, LOW);  //Right Motor backword Pin

digitalWrite(in3, LOW);  //Left Motor backword Pin

digitalWrite(in4, HIGH); //Left Motor forword Pin

}

void backword(){ //backword

digitalWrite(in1, LOW);  //Right Motor forword Pin

digitalWrite(in2, HIGH); //Right Motor backword Pin

digitalWrite(in3, HIGH); //Left Motor backword Pin

digitalWrite(in4, LOW);  //Left Motor forword Pin

}

void turnRight(){ //turnRight

digitalWrite(in1, LOW);  //Right Motor forword Pin

digitalWrite(in2, HIGH); //Right Motor backword Pin

digitalWrite(in3, LOW);  //Left Motor backword Pin

digitalWrite(in4, HIGH); //Left Motor forword Pin

}

void turnLeft(){ //turnLeft

digitalWrite(in1, HIGH); //Right Motor forword Pin

digitalWrite(in2, LOW);  //Right Motor backword Pin

digitalWrite(in3, HIGH); //Left Motor backword Pin

digitalWrite(in4, LOW);  //Left Motor forword Pin

}

void Stop(){ //stop

digitalWrite(in1, LOW); //Right Motor forword Pin

digitalWrite(in2, LOW); //Right Motor backword Pin

digitalWrite(in3, LOW); //Left Motor backword Pin

digitalWrite(in4, LOW); //Left Motor forword Pin

}

**4.1.3 Output of line following robot:**

The line follower robot senses a black line by using a sensor and then sends the signal to Arduino. Then Arduino drives the motor according to sensors' output. Here in this project, we are using two IR sensor modules namely the left sensor and the right sensor.



##### Image 2

**4.2.1. Fig: Smart dustbin**

|  |
| --- |
|  |

4.2.2. Code for an Smart Dustbin:

#include <Servo.h>

const int trigPin = 2;  // Ultrasonic sensor trigger pin

const int echoPin = 3;  // Ultrasonic sensor echo pin

const int servoPin = 9;  // Servo motor control pin

Servo servo;             // Servo object

const int obstacleThreshold = 30;  // Distance threshold for obstacle detection (in cm)

void setup() {

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  servo.attach(servoPin);

}

void loop() {

  long duration, distance;

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  distance = duration \* 0.034 / 2;  // Calculate distance in cm

  if (distance <= obstacleThreshold) {

    // Obstacle detected, move servo motor

    servo.write(90);  // Adjust the angle as needed

    delay(1000);     // Delay to allow the servo to reach the desired position

  }

  else {

    // No obstacle detected, reset servo motor position

    servo.write(0);  // Adjust the angle as needed

  }

  delay(100);  // Delay between distance measurements

}

4.2.3. Output of smart dustbin:

It gives a **real time indicator** of the garbage level in a trashcan at any given time. Using that data, we can then optimize waste collection routes and ultimately reduce fuel consumption. It allows trash collectors to plan their daily/weekly pick up schedule. The level of dustbin filled or not is done by using an Ultrasonic sensor or IR sensor.

|  |
| --- |
| Fig :output of smart dustbin |

# CHAPTER - 5

**Chapter 5: RESULTS**

**5.1. Result of image 1**

**Fig5.1.** Output of line following robot and smart dustbin

|  |
| --- |
|  |

# CHAPTER - 6

**Chapter 6: APPLICATIONS, ADVANTAGES AND DISADVANTAGES**

**6.1. APPLICATIONS:**

Line following robots have several practical applications across various industries

and domains. Here are some common applications of line following robots:

*1*.Industrial Automation: Line following robots are extensively used in

industrial settings for tasks such as material handling, assembly line

operations, and quality control.

*2*.Warehouse Logistics: Line following robots can be employed in

warehouses and distribution centers to navigate and transport items.

*3*.Food and Beverage Industry: Line following robots find applications

in food and beverage production facilities.

*4*.Healthcare: Line following robots can be utilized in hospitals

and healthcare facilities to perform various tasks.

*5*.Education and Research: Line following robots serve as educational

tools to introduce students to the basics of robotics, programming,

and automation.

*6*.Security and Surveillance: Line following robots can be employed for

security purposes in public spaces, such as shopping malls or airports.

*7*.Agriculture: Line following robots have applications in the field

of agriculture. They can be used for crop monitoring, data collection,

or precision farming tasks.

*8*.Entertainment and Recreation: Line following robots are also popular in

the entertainment and recreational domain. They can be used in

amusement parks or robotic competitions to provide interactive

experiences.

#### 6.2. Advantages:

Line following robots offer several advantages in various applications.

* + Automation: Line following robots automate repetitive tasks that involve following a specific path or line.
  + Precise and Consistent Path Following: Line following robots are designed

to accurately track and follow lines or paths with high precision.

* + Versatility: Line following robots can be programmed to follow different

types of lines or paths, allowing them to adapt to various environments and

tasks.

* + Cost-Effectiveness: Line following robots often utilize relatively simple

hardware and sensors, making them cost-effective compared to more complex robotic systems.

* + Easy Implementation and Integration: Line following robots are relatively

easy to implement and integrate into existing systems.

* + Learning and Educational Tool: Line following robots serve as excellent

educational tools to introduce students and enthusiasts to robotics,

programming, and automation.

* Scalability: Line following robots can be scaled up or down depending

on the application requirements. Multiple robots can be deployed to

work collaboratively or independently, depending on the task. This

scalability makes them adaptable to different environments, production

volumes, or complexity levels.

* Safety Enhancement: Line following robots can enhance safety in

certain applications.

#### DISADVANTAGES

* + - It's very costly depending on the system used, the number of detectors purchased.
    - Time consuming.
    - Lack of qualified professional.

# CHAPTER - 7

**Chapter 7: CONCLUSION AND FUTURE SCOPE**

#### CONCLUSION:

As we had defined the aim earlier i.e., to implement image processing using Verilog, we successfully got the result as expected. As specified in methodology, the image input when given to MATLAB, converts the input image format into hexadecimal which when given to Verilog, performs the operations as specified. The image after the operation is performed is given as an output and can be viewed in bmp format on the computer screen. Thus we got the output as expected in this project

#### FUTURE SCOPE:

The future scope of line following robots is quite promising, with several potential advancements and applications. Here are some possible areas of development:

1. Advanced Navigation: Line following robots can be equipped with more sophisticated navigation systems, such as computer vision algorithms and machine learning techniques, to enhance their ability to follow complex lines and adapt to changing environments. This could involve using more advanced sensors, such as LiDAR or depth cameras, to improve their perception capabilities.

2. Autonomous Operations: Currently, line following robots are mostly programmed to follow predetermined paths. However, the future may bring more autonomy, allowing them to make decisions and navigate dynamically based on their surroundings. This could involve incorporating artificial intelligence and decision-making algorithms, enabling them to handle unexpected obstacles and optimize their path planning.

3. Multi-Tasking Capabilities: Line following robots can be developed to perform additional tasks simultaneously while following a line. For instance, they could have integrated sensors for environmental monitoring, data collection, or even robotic arm attachments to manipulate objects along the line. This would expand their functionality and make them more versatile in various applications.

4. Swarm Robotics: Line following robots can be used in swarm robotics, where

multiple robots collaborate to accomplish tasks collectively. Swarm robots can work together to follow complex lines, explore unknown environments, or perform coordinated actions. This approach can bring advantages in terms of efficiency, fault tolerance, and scalability.

5. Industrial Automation: Line following robots can play a significant role in industrial automation processes. They can be employed in assembly lines, material handling, and

logistics, where they follow lines to transport goods, perform quality inspections, or deliver items. Increasing their capabilities and reliability can lead to improved efficiency and reduced human intervention in these industrial applications.

6. Educational Tools: Line following robots have been widely used as educational tools to introduce basic robotics concepts and programming to students. The future can bring more advanced educational platforms and kits, enabling students to experiment with AI, machine learning, and advanced control algorithms, fostering their interest in robotics and technology.

7. Healthcare and Service Applications: Line following robots can find applications in

healthcare and service sectors, assisting with tasks like medication delivery in hospitals, navigation within large facilities, or providing assistance to individuals with mobility issues. With further advancements, line following robots can be made more robust, reliable, and capable of interacting with humans in a safe and helpful manner.

# REFERENCES

### REFERENCES:

1. **“Line Following Robot Using Arduino for Hospital”,** Jagruti Chaudhari, Asmita Desai, S. Gavarskar, ICCT, IEEE 2019.
2. **“Vision-Based Line Following Robot In Webots”** Alfian Maarif, Aninditya Anggari Nuryono, Iswanto, FORTEI, IEEE 2020.
3. **“Line following robots on factory floors”,** Significance and Simulation study using Coppelia Sim. Saharsh Oswal1 and Saravanakumar, School of Mechanical Engineering, Vellore Institute of Technology, India 2021.