INTELLIGENT LINE FOLLOWER AND HAND TRACKING ROBOT USING ARDUINO UNO

A THESIS SUBMITTED TO
THE FACULTY OF ARCHITECTURE AND ENGINEERING
OF
EPOKA UNIVERSITY

BY

DYLBER CAUSHI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRONICS AND COMMUNICATION
ENGINEERING

FEBRUARY, 2024

Approval sheet of the Thesis

This is to certify that we have read this thesis entitled **“**Intelligent Line Follower Robot Using Freenove Starter Kit for Arduino UNO**”** and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

____

Assoc. Prof. Dr. Arban Uka
Head of Department
Date: February, 29, 2024

Examining Committee Members:

Dr. Florenc Skuka __________________

Prof. Dr. Bekir Karlik __________________

Dr. Valmir Bame __________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Surname: Dylber Caushi

Signature: _______________

ABSTRACT

INTELLIGENT LINE FOLLOWER ROBOT AND HAND TRACKING ROBOT USING ARDUINO UNO

Caushi, Dylber
M.Sc. Department of Computer Engineering
Supervisor: Prof. Dr. Bekir Karlik

The integration of robotics in the various fields of automation achieved remarkable advancement. Robots are specially designed machines that offer solutions to complex problems and bring efficiency and advancement in various fields. Robots are fast, accurate, and dedicated as compared to humans. The robots are controlled through sensors, programming, and manipulators. Line follower robots are one of the basic types of robots and follow white lines on black surfaces or black lines on white surfaces by using IR sensors. These robots can be used for delivery purposes, self-driving cars, transportation, and industrial purposes.

This thesis is separated in two parts, the first one discusses the design, development, and implementation of a color-based line follower robot using AI (or machine learning) freenove starter kit for Arduino UNO and the second one is about hand tracking robot using an ultrasonic sensor, 2 IR sensors, Arduino UNO and L293D motor driver.

Line Follower study aims to increase speed, efficiency, and accuracy and to reduce errors while tracking the way. The project explores the capabilities of the Arduino platform in creating an intelligent and automatic robotic system that is capable of following a predefined path. The thesis explores the importance of the line follower robot and also covers the drawbacks.

The study uses many components to design the robot. The IR sensor, Arduino, moto driver, and motors. The IR (infrared) sensors sense the path (the white or black lines) and send the data to Arduino. The color sensor and KNN algorithm are used to

choose the desired way of robot. Arduino is the brain of the project and decides what to do and when to do it. The Arduino is programmed using a software called Arduino IDE. Arduino controls the motor driver connected to it. The motor driver controls the motor like its speed and power. Over all a controlled system is designed to follow the predefined path with high accuracy.

The second section of this paper describes the design and implementation of a hand-tracking robot that uses an Arduino UNO microcontroller, an L293D motor driver, ultrasonic sensors, and two infrared (IR) sensors. The primary goal of this robotic system is to track the movement of a hand in its area, showcasing the use of robotic following systems and sensor integration for real-time monitoring. The robot uses ultrasonic sensors for distance measuring and obstacle identification, while infrared sensors detect the presence and movement of a hand. The L293D motor driver provides precise control over the robot's motions, allowing it to modify its course and speed in reaction to the hand's location, assuring smooth and responsive tracking.

Overall this thesis contributes to the growing field of robotics by providing a comprehensive overview of line follower and hand tracking robots using Arduino UNO. The project aims to make both robots smarter to enhance capability on their predefined path with high accuracy and wide usage in our daily life.

Keywords: *Arduino UNO, motor driver, line follower, hand tracking, ultrasonic sensors, IR sensors, programming, Arduino IDE.*

ABSTRAKT

ROBOTI INTELIGJENT QE NDJEKE LINJEN ME MATERIALE NGA FREENOVE PER ARDUINO UNO

Caushi, Dylber

Master Shkencor, Departamenti i Inxhinierisë Kompjuterike

Udhëheqësi: Prof.Dr. Bekir Karlik

Integrimi i robotikës në fusha të ndryshme të automatizimit arriti një përparim të jashtëzakonshëm. Robotët janë makina të dizajnuara posaçërisht të cilat ofrojnë zgjidhje për probleme komplekse dhe sjellin efikasitet dhe avancim në fusha të ndryshme. Robotët janë të shpejtë, të saktë dhe të përkushtuar në krahasim me njerëzit. Robotët kontrollohen përmes sensorëve, programimit dhe impulseve. Robotët "Ndjekës të Linjës" janë një nga llojet bazë të robotëve dhe ndjekin vija të bardha në sfond të zi ose vija të zeza në sfond të bardhë duke përdorur sensorë IR. Këta robotë mund të përdoren për qëllime dërgese, makina vetë-drejtuese, transport dhe qëllime industriale.

Kjo tezë është e ndarë në dy pjesë, e para diskuton projektimin, zhvillimin dhe zbatimin e një roboti ndjekës të linjës me bazë ngjyrash duke përdorur pjese nga kompania Freenove AI (ose mësimin e makinerive) për Arduino UNO dhe e dyta ka të bëjë me robotin i cili ndjek dorën duke përdorur një sensor ultrasonic, 2 sensorë IR, drejtues motori dhe Arduino UNO.

Studimi i robotit qe ndjek nje linje te caktuar synon të rrisë shpejtësinë, efikasitetin dhe saktësinë dhe të reduktojë gabimet gjatë gjurmimit të rrugës. Projekti shpjegon aftësitë e platformës Arduino në krijimin e një sistemi robotik inteligjent dhe automatik që është i aftë të ndjekë një rrugë të paracaktuar. Teza tregon rëndësinë e robotit ndjekës të linjës dhe gjithashtu mbulon të metat.

Studimi përdor shumë komponentë për të dizajnuar robotin. Sensori IR, Arduino UNO, drejtuesi i motorit dhe motorët. Sensorët IR (infra të kuqe) dallojne vijat (vijat e bardha ose të zeza) dhe i dërgojnë të dhënat në Arduino. Sensori i ngjyrave dhe algoritmi KNN përdoren për të zgjedhur mënyrën e dëshiruar të robotit. Arduino është truri i projektit dhe vendos se çfarë të bëjë dhe kur ta bëjë atë. Arduino është programuar duke përdorur një softuer të quajtur Arduino IDE. Arduino kontrollon drejtuesin e motorit të lidhur me të. Drejtuesi i motorit kontrollon motorin si shpejtësinë dhe fuqinë e tij. Mbi të gjitha, një sistem i kontrolluar është projektuar për të ndjekur rrugën e paracaktuar me saktësi të lartë.

Pjesa e dytë përshkruan projektimin dhe zbatimin e një roboti I cili ndjek dorën duke përdorur një mikrokontrollues Arduino UNO, një drejtues motori L293D, sensorë ultrasonic dhe dy sensorë infra të kuqe (IR). Qëllimi kryesor i këtij sistemi robotik është të ndjeke lëvizjen e një dore në zonën e saj, duke shfaqur përdorimin e sistemeve robotike ndjekëse dhe integrimin e sensorëve për monitorim në kohë reale. Roboti përdor sensorë tejzanor për matjen e distancës dhe identifikimin e pengesave, ndërsa sensorët infra të kuqe zbulojnë praninë dhe lëvizjen e një dore. Drejtuesi i motorit L293D siguron kontroll të saktë mbi lëvizjet e robotit, duke e lejuar atë të modifikojë kursin dhe shpejtësinë e tij në reagim ndaj vendndodhjes së dorës, duke siguruar gjurmim të qetë dhe të përgjegjshëm.

Në përgjithësi kjo tezë kontribuon në fushën në rritje të robotikës duke ofruar një përmbledhje gjithëpërfshirëse të robotëve ndjekës të linjës dhe gjurmimit të dorës duke përdorur Arduino UNO. Projekti synon t'i bëjë të dy robotët më të zgjuar për të rritur aftësinë në rrugën e tyre të paracaktuar me saktësi të lartë dhe përdorim të gjerë në jetën tonë të përditshme.

Keywords: *Arduino UNO, kontrollues motori, ndjekesi i linjes, ndjekesi I dores, sensor ultrasonic, sensor IR, programim, Arduino IDE.*

ACKNOWLEDGEMENTS

First and foremost, I immensely thank full to my project supervisor Prof. Dr. Bekir Karlik for their invaluable guidance. Support and insightful feedback throughout the research process.

It is not a matter of percept, but a solemn duty to acknowledge the efforts of all who have contributed towards the completion of this Thesis. I would like again to express my sincere gratitude and respect to my supervisor Prof. Dr. Bekir Karlik for his patience, devotion, guidance, and support. It has been an honor and blessing to learn from him. I greatly appreciate his contribution of time and his ideas to make my experience productive and enhance my motive toward achieving the given tasks.

I also want to acknowledge the prayers, support, and encouragement of my loving parents and friends, especially Xhois Shaholli.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Nowadays robots become very advanced and can be adapted to any type of work from daily home cleaning to complex industrial tasks. Robots play various crucial roles from medical to local transportation. In the era of rapid technological advancement, the demand for intelligent and adaptable robots has increasing day by day. One of the simple types of robots is a line follower robot. Line follower robots follow the line using sensors.

The last decade, intelligent car automation system has been using machine learning algorithms. ANN is a subset of machine learning and very important part in deep learning algorithms. Ozyigit and Karlik, has presented car suspension systems by using machine learning. In their paper, using Artificial Neural Network they have explained how to obtained faster results by reducing the calculations time. ANN provides them values of the control force in an exceptionally brief amount of time, by applying multilayer perceptron and backpropagation algorithms. [1]

The main aim of the project is to develop two robots, line follower and a hand tracking robot using machine learning algorithm (K-Nearest Neighbor) for Arduino UNO.

The first robot will follow a black line on the white surface and decide which path to go at a roundabout. Many line follower robots have been made but the main challenge is to make them fast responsive, precise, and adaptable to diverse environments. The main goal of our project is to bring improvement in terms of sensor accuracy, control algorithm, and fast response. This theoretical study and the practical implementation of the project will make robotic education more engaging and applicable to real-world scenarios.

The second robot, hand tracking, seeks to demonstrate real-time dynamic tracking capabilities, using infrared and ultrasonic sensors to identify and track the movement of a hand in its proximity. Conversely, the line follower robot is meant to walk courses designated by lines, employing similar sensor technology to accomplish accurate and responsive movement. These goals are founded in the overarching goal of promoting educational robotics by providing venues for students and enthusiasts alike to engage with fundamental and advanced topics in robotics engineering, sensor integration, and programming.

A similar analysis, describes the design and development of a robotic shopping assistant system that assists disabled or elderly people by carrying their shopping loads. Using mobile devices for user engagement, the system sends mobile robots to customers depending on the shopping lists they send to the supermarket's computer. These robots, outfitted with ultrasonic sensors, Arduino microcontrollers, and Bluetooth connection, follow customers throughout the supermarket, overcoming obstacles and keeping a safe distance. The system prioritizes ease of use, efficiency, and safety, with superior navigation and obstacle avoidance capabilities. It also investigates the mechanical architecture of the shopping cart, strengthening its mobility and stability with a differential wheeled drive and a suspension system, highlighting robots' potential for improving accessibility and convenience in retail situations. [2]

1.2 Thesis Objective

The first main objective of the study is to make an intelligent line follower robot that follows a black line on a white surface and make decisions if there are more than one line, for example in a roundabout there is also a red and yellow line were. The aims and goals of the project are given below.

- To develop a physical line follower robot using machine learning with Freenove kit for Arduino UNO.
- The robot must be capable of following the black line.
- To improve sensor accuracy and control algorithm.
- To enhance the stability and accuracy of the robot using the PID algorithm.

- To make the robot insensitive to the environment changes.
- To detect and avoid obstacles.
- To make the robot adaptable to real-world uses such as to improve work in warehouses, factories, and transportation.

The second purpose of the thesis, is to design, build, and empirically evaluate a hand-tracking robot, with a focus on the system's capacity to properly identify and follow hand motions. This goal is driven by a desire to contribute to the educational and enthusiast robotics communities by offering an in-depth example on the use of combination of sensors, control systems, and programming in a real robotics project. By traversing the challenges of sensor-based tracking and motion control, the project hopes to provide insights into creating responsive and interactive robotic systems with popular open-source hardware and software platforms.

1.3 Scope of works

The thesis aims to use a Freenove-kit for Arduino to design, implement, and optimize a line follower robot and also a hand tracking robot. The research explores the effect of the different sensor configurations on the performance and responsiveness of the robot during line following scenario. The project uses Arduino to control the movements of the robot and make the right decision at the right time because Arduino is intelligent in decision-making due to the program installed in it through Arduino IDE software. For example, the robot will never be distracted from the predefined path because the Arduino can see the path through IR sensors and keep the robot on track. The research involves a comprehensive assessment of the line follower's components and aims to enhance the robot's control algorithm, response speed, and adaptability.

The scope of our study also includes advanced features of the robot since we have done twice this robot, first time using the Arduino UNO and the second time using the Raspberry Pi which can handle more complex algorithms and faster data processing. We have tried multiple problems with the robot like: obstacle detection and avoidance, object tracking but, here we have worked on finding the shortest path in a maze. This robot can handle indoor and outdoor environments, but higher capabilities are indoor environments.

The hand tracking robot contains a thorough study of the hardware selection process, with a particular emphasis on the Arduino UNO microcontroller's and L293D motor driver's roles in controlling the robot's motions and processing sensor data. The use of ultrasonic and IR sensors for real-time hand identification and tracking is explored, as well as the creation of a control algorithm that allows the robot to read sensor data and alter its motions accordingly. The paper's comprehensive analysis of the robot's design intends to shed insight on the complexities of constructing sensor-based robotic systems and stimulate future innovation in the field of robotics.

1.4 Organization of the thesis

This thesis is divided into 6 chapters. The organization is done as follows:

In Chapter 1, the problem statement, thesis objective, and scope of works are presented. Chapter 2, includes the literature review. In Chapter 3 basic functions of the Arduino program are explained. Chapter 4, consists of the methodology followed in this study. In Chapter 5, Machine Learning Algorithm. In Chapter 6, Results and Discussion. In Chapter 7, Conclusion and recommendations for further research are stated.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Line follower robots play a very important role in today’s world because such types of robots are adaptable to every type of work. The line following robots are useful for disabled persons, especially for blind people. It can reduce the overall cost of the industries and can be used in extreme environmental conditions where humans can’t work. Many line follower robots have been made for different purposes using different technologies and components. Some robots are made for industrial purposes and some are made for general work like for home usage or public transport.

People have different approaches to making a line-follower robot. Someone may use a servo motor for steering purposes while another person may control the voltage and speed of the motors to turn the robot left or right. The main goal is to reduce the cost and increase the stability of following the line and the robot has full control of its movements. The stability and smooth controls of that robot are very crucial when it is considered for public transport because if the desired control and stability are not achieved it may cost human lives.

In the field of line follower robots for industrial purposes, there is a research paper titled” Line Follower Robot for Industrial Manufacturing Process” published in 2017. The paper explores the significance of the line follower robot in industrial applications and explains its construction. The researchers used Arduino, a motor driver, and an array of IR sensors. The robot will follow black lines on the white surface. The robot has 3 wheels the front wheel is independent and the back wheels are controlled by the motors. The robot needs improvement in control algorithm, stability, and design, and in ambient light, the sensors may fail to observe the line [3].

Another research was made on the design, implementation, and technical issues of the line follower robot. The research suggests that the robot must follow the predefined line and must take smooth turns to not go off the track. The research further highlights the rule of the physical shape of the robot in stability and speed. For example, the weight and wheel radius must be considered while designing a robot. The weight of his designed robot is 300 mg and it can be lighter. The researchers didn’t mention how to minimize the shaking of robots across the line and how to improve the controlling strategies [4].

A research paper titled” Line Follower Robot: Fabrication and Accuracy Measurement by Data Acquisition” investigates line robot follower and its fabrications, performance, and accuracy measurement. The research focuses on the construction of a simple LFR using a microcontroller. The study evaluates the robot's efficiency and sensor responsiveness by collecting actual data from the sensors and providing feedback to the CPU (central processing unit) based on these replies. The study also demonstrates mistake correction in following the line. The suggested robot design of the paper has a battery of 7.2 A but only 600 mA has served to each motor and because of this, the motor was running at 103 rpm. which was half of its RPM rating. This reduction is due to the architectural limitations of the robot [5].

In the field of autonomous robotics, an innovative contribution is presented by a research paper on the multiple source multiple destination robot (MDR-1). The robot can choose a specific line among other lines autonomously. Every line has been colored differently. The robot can reach its destinations by choosing the right line from different colored lines. The robots have another feature that sets it apart from other robots, is its ability to detect obstacles on its path. The MDR-1 robot has impressive performance in detecting obstacles and multi-colored line selection but it is crucial to recognize its complexity and challenges in making such robots. Its complexity and cost of the design might it less accessible for people or educational purposes [6].

Artificial Intelligence in the automotive industries is widely used especially in the self-driving technologies and its main purpose is improvement of safety, fuel efficiency and helps drivers for long trips on highways or when they get microsleep.

Mercedes Benz in his last type “EQS”, which is fully electric and released in 2022, has done an incredible composition of machine learning. Mercedes at their own page in April 2022, set a new record with their safety standard by releasing the Attention Assis. A camera and a sensor placed at the steering wheel detects driver’s eyelid movement, motions like fatigue and inattention, considering various factors such as driving behaviors, steering movements, driving time and so many, and after that car warns the driver with audio and visuals to take break. [7]

In a paper of Ramya, how AI is integrated Automotive Embedded Systems he talks about different usage of AI, like Autonomous Driving, Decision making and planning, risk reduction etc. Autonomous driving into cars systems needs techniques like machine learning and computer vision to detect and collect as many data as possible like immediate surroundings, traffic elements (laws, traffic flow, certain routes) and then executing the safest and fastest maneuvers, this leads to passenger, protection. Talking about risk and protection, AI systems are able to collect data also if there are any changes of forecast or construction in the road so examining these components is crucial. [8]

All the models represented in the above discussion have some issues, like some have high design costs, and some robot models are complex in design and have efficiency and performance issues. In contrast, our proposed simple robot models aim to address these challenges and optimize implementation complexity, energy efficiency, and design cost. The main focus of our study is to propose such a model which is immune to environmental changes and works well in ambient light. Our proposed model is simple in design and can be used for education purposes as well as for other tasks with some modification.

Hand tracking robots have sparked widespread attention due to their potential uses in gesture control, interactive learning, and rehabilitation. The Arduino UNO, known for its accessibility and adaptability, acts as the computing brain for many robotic applications. The L293D motor driver, working in tandem, is critical to motor control, allowing for accurate motions and reactions. The integration of ultrasonic and infrared sensors in hand tracking robots is a critical area of study. Ultrasonic sensors, which are known for their capacity to estimate distance using sound waves, are useful in determining the location of a hand relative to the robot. Meanwhile, IR sensors, which detect infrared light produced or reflected by objects, are critical for identifying hand gestures and motions.

2.2 Hardware Components

To make a robot the selection of hardware components is a very difficult task because these hardware components determine the functionality and adaptability of certain robots. To make a line follower robot we carefully choose a set of hardware components. The components include the IR sensors for line detection and environment observation, Arduino UNO for processing and decision making on the response of IR sensors, L293D motor driver for controlling DC motors speed and power, DC motors to rotate the wheels, battery for power supply, wheels for movement and frame to support all the components. These components are discussed in detail in the following section.

2.2.1 Infrared (IR) Sensors Array

Infrared (IR) sensors play a very crucial role in the functionality of the line follower robot. infrared sensors provide real-time feedback to the Arduino about the environment. An array of 8-channel Infrared sensors is placed beneath the robot. IR sensors emit invisible infrared light. These sensors measure the reflection of the light from the surface to observe the environment.

The predefine line should be black on the white surface or it may be white on the black surface. But in this case, the robot will follow the black line on the white surface. The robot distinguishes between the line and the surface through a reflection pattern. The reflection pattern changes when the surface beneath changes. For example black color absorbs almost all incident light and white color reflects the light. The amount of reflection is changed to the voltage by IR sensors. With the help of this information, Arduino decides whether the robot follows the line o it is distracted. The IR sensor array is shown in Figure 1.

Figure 1. IR Sensor ***[9]***

The sensitivity of the sensor can be changed by turning the potentiometer. Turning the potentiometer clockwise makes the IR sensor more sensitive while turning the potentiometer anticlockwise makes the IR sensor less sensitive. The sensor has two parts the emitter circuit and the receiver circuit. Both circuits are collectively known as photo couplers or opt couplers. The infrared light is emitted by IR LED and detected by IR photodiode.

2.2.2 Color Sensor

In robotics, a color sensor is a sophisticated device which helps to measure RGB colors and their intensity. They are widely used in food manufacturing, printing, medical or environment protection, even at the project we are working on. [10]

At a medical paper, by Saad Hussein and Ersin Kayahan, they realized that many people wanted to get everyday results of the urine, so a simple and manageable device was needed. The problem was already there, so a simple device came in their mind, a color sensor, an Arduino UNO and some tapes that will react with the urine and make a specific color if there was a problem.

Figure 2. TCS34725 Color Sensor

2.2.3 Arduino UNO

Arduino UNO is the brain of the line follower robot. it makes real-time decisions based on the input data received from the IR sensors. Arduino has a microcontroller on its board. Arduino determines the DC motor response based on the input IR sensor data and keeps the robot on the line. Arduino functions depend upon the code. This code is written in software called Arduino IDE software and uploaded to the microcontroller of the Arduino.

Arduino has 14 digital input output pins, and 16 analog inputs and is operated at a voltage of 5 volts. The Arduino Uno microcontroller plays a big part in robots. Based on the ATmega328P chip, provides the processing power necessary for sensor data interpretation and motor control. It commands things like motors sensors and actuators. Motors move the robot. They take orders from the Arduino's PWM, or pulse width modulation, signals. Sensors spot changes in what's around the robot. They tell the Arduino what they see, and the Arduino uses this to tell the robot what to do. Actuators follow the Arduino's orders to perform specific tasks

Arduino UNO is easy to use, cost-effective, and energy efficient. Its other versions are also available such as Arduino Mega and Arduino Mini but they are expensive and used for complex projects. The following figure shows Arduino UNO.

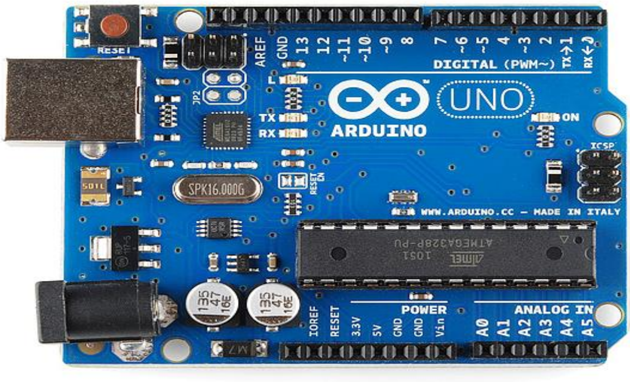

Figure 3. Arduino UNO

2.2.4 L293D Motor Driver

The L293D motor driver is a crucial component of the line follower robot. The L293D motor drive acts as an intermediator between Arduino and DC motors. The motor driver controls the power and speed of the DC motors. The motor driver follows the Arduino commands. According to these commands motor driver varies the power and the speed of the DC motors.

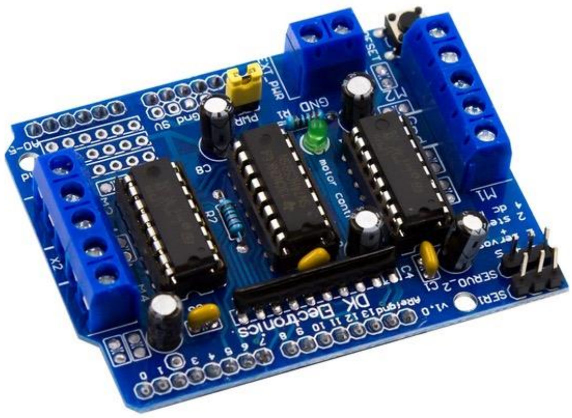

Figure 4. L293D Motor Driver

For example, if the robot goes off the track the Arduino will take some necessary actions based on the input IR data. To me on the track, Arduino will send commands to the motor driver to vary the speeds of the wheels to turn the robot.

2.2.5 DC Motors

DC motors are also important components of the line follower robot. DC motors significantly impact the motor speed. The motor's supply voltage and speed are controlled by the motor driver. The selection of a DC motor is crucial while considering some factors like torque, speed, and power efficiency.

Figure 5. DC motor

2.2.6 Wheels

The selection of proper wheels for line follower robots is crucial because they must meet the surface condition and must not slip on the surface.

Figure 6. Wheels of the robots

2.2.7 Power Supply

To maintain the operational capability of the line follower robot we need a voltage source in the form of a battery. The battery should be reliable and must have a good lifetime to make continuous power supply to all the components, to allow the robots to carry out their tasks. Some factors like weight, voltage, and capacity must be considered while selecting a battery to keep the balance between performance and the overall weight of the robot.

Figure 7. Battery

2.2.8 Chassis and Frame

The chassis forms the structural foundation of the robot, providing support for all components. It must be designed with considerations for stability, weight distribution, and ease of assembly. Material selection, such as lightweight metals or durable plastics, is important in chassis design.

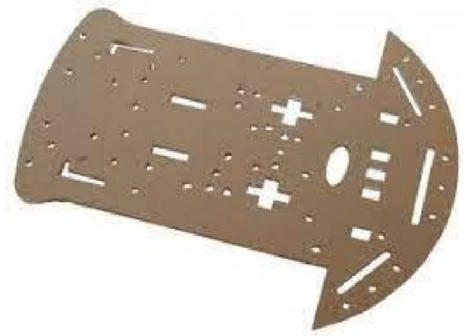

Figure 8. Chassis and Frame for LFR

2.2.9 Ultrasonic sensor

An ultrasonic sensor uses ultrasonic sound waves to detect the distance to an item, It has a transmitter, which delivers the pulses, and a receiver, which receives information from the nearby item, the pulse is sent to the motor driver and Arduino. It works with 5V DC power supply and has a ranging distance between 2-400cm and an angle of 30 degree. The principle of this sensor is from bats which use sound waves to detect objects. The formula to find the distance is pretty simple:

$$\boldsymbol{distance} = \frac{\boldsymbol{speed\ of\ sound \times time\ taken}}{\mathbf{2}} \qquad \text{(Equation 1)}$$

Speed of sound that travels through air is 344 m/s, time taken is calculated by the sensor and is the time when the signal is triggered and received back after the signal is reflected back by the object, and the number is divided by 2 since there are 2 pulses one that is triggered and the other one that is received. This figure shows the exact sensor that we have used. [11]

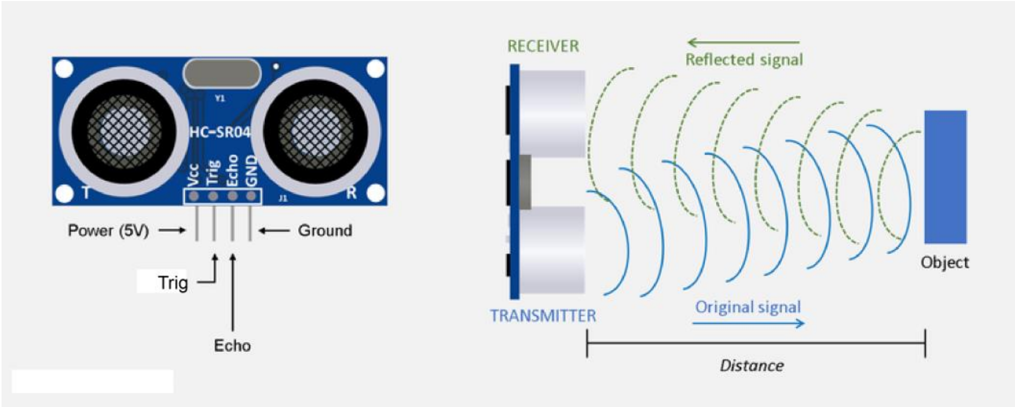

Figure 9. Ultrasonic Sensor

2.2.10 IR Sensors

An infrared sensor is an electrical gadget that emits light in order to detect nearby things. Typically, everything in the infrared region emits some form of heat radiation. These radiations are invisible to human sight, but an infrared sensor can detect them. It is made up of an IR LED that sends light and a photo diode that receives it. It uses the same idea as an ultrasonic sensor. [12]

Figure 10. IR Sensors

2.3 Software Components

Arduino IDE is the only software used in this project. Arduino IDE is used for programming Arduino IDE microcontroller. The program is uploaded through a cable form connected to a PC and Arduino. The programming language is based on C++ and is designed to be easy for non-programmers or beginners.

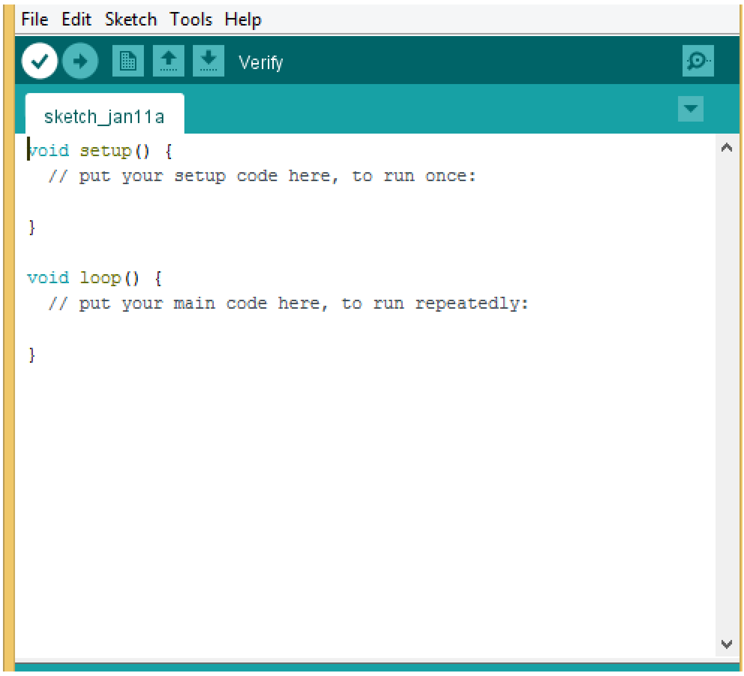

Figure 11. Arduino IDE software interface

The toolbar of the interface has 5 options (buttons). The first option (like a tick) is for verifying the code. When the code the written the tick button is clicked, and it compiles the code and checks for errors. The option is used for uploading the code to the Arduino microcontroller. It is indicated by the right arrow.

The third option is used to open a new window and write a new code. This option is represented by a page-like structure. The 4th option is represented by an up arrow and is used to open files already present in the storage. A down arrow saves the written code in the PC storage.

CHAPTER 3

CODE IMPLEMENTATION

The line follower robot is controlled by Arduino and it is the code that able Arduino for this function. Every component of the robot is controlled by this code written for the Arduino microcontroller. The code is written in software called Arduino IDE and uploaded to the Arduino microcontroller through a cable. The Arduino code is divided into two sections.

3.1 Section 1

Without Obstacle detection (Existing Code)

The code is written in a simplified version of C++. The code is divided into parts and each part is explained.

Variable Declaration: these lines declare the integer variables to store pin numbers for motor speed control, IR sensor reading and threshold values for sensors.

Setup Function: the function here is the configuration of Arduino pins for motor control. The function “setup” calls setup motor (1,1) to initialize the motor to move forward.

Brake Function: sets both DC motors for the brake when the input signals to its pins are high.

Setup Motor Function: configures the direction of each motor based on the values of the two variables forward a and forward b.

Change Speed Function: changes the duty cycles of PWM signals.

Read Eye Function: this function reads analog values from the sensor and compares them with threshold values, if the reading is below the threshold it returns 0, otherwise 1.

```
int pin_in1 = 8;
int pin_in2 = 9;
int pin_ena = 11; // PWM speed control of left motor
int pin_in3 = 6;
int pin_in4 = 7;
int pin_enb = 10; // PWM speed control of right motor
Int line0, line1, line2, line3, line4, line = 0, no_line, speed_a,
speed_b;
int threshold = (400 + 19) / 2;
int max_speed = 255, loss = 50, turning_speed = 255;
void setup() {
pinMode(pin_in1, OUTPUT);
pinMode(pin_in2, OUTPUT);
pinMode(pin_ena, OUTPUT);
pinMode(pin_in3, OUTPUT);
pinMode(pin_in4, OUTPUT);
pinMode(pin_enb, OUTPUT);
pinMode(LED_BUILTIN, OUTPUT);
setup_motors(1, 1);
}
void brake() {
digitalWrite(pin_in1, HIGH);
digitalWrite(pin_in2, HIGH);
digitalWrite(pin_in3, HIGH);
digitalWrite(pin_in4, HIGH);
}
void setup_motors(int forward_a, int forward_b) {
// Configures motor A direction based on forward_a and motor B direction
based on forward_b.
}
void change_speed(int speed_a, int speed_b) {
analogWrite(pin_ena, speed_a);
analogWrite(pin_enb, speed_b);
}
int read_eye(int eye) {
// Reads the analog value from the specified eye sensor.
}
void loop() {
// Continuously reads sensor values, controls motors, and adjusts speed
based on sensor inputs.
}
```

Variable Declaration

Setup Function

Brake Function

Setup Motors Function

Change Speed Function

Read Eye function

Loop Function

Figure 12. Code Implementation, no obstacles

3.2 Section 2

With Obstacle Detection

Obstacle detection is a good feature of the line follower robot. with this feature, the robot is capable of avoiding obstacles coming in its path deciding to turn left or right, and managing to come back to the predefined path. To add this feature, we need a servo motor and ultrasonic sensor. A portion of the code is given which uses a servo motor and ultrasonic sensor to avoid obstacles.

```
void loop() {
    // Read ultrasonic sensor to get distance
    distance_F = Ultrasonic_read();
    Serial.print("D F=");
Serial.println(distance_F);
    // If both right and left sensors detect white
color (on a line), move forward
    if ((digitalRead(R_S) == 0) && (digitalRead(L_S)
== 0)) {
        if (distance_F > Set) {
            forword(); // move forward
        } else {
            Check_side(); // if obstacle detected,
check sides
        }
    }
    // If right sensor detects black and left sensor
detects white, turn right
    else if ((digitalRead(R_S) == 1) &&
(digitalRead(L_S) == 0)) {
        turnRight();
    }
    // If right sensor detects white and left sensor
detects black, turn left
    else if ((digitalRead(R_S) == 0) &&
(digitalRead(L_S) == 1)) {
        turnLeft();
    }
    delay(10); // a small delay for stability
}
```

Ultrasonic Sensor Function

Side Function

Figure 13. Code Implementation, Obstacle Detection

3.2.1 Line Following

The robot will continue its forward movement through the line until an obstacle comes in its path. The line follower robot checks the distance ahead by using an ultrasonic sensor **(‘distance _F’)**. If the distance ahead is greater than the predefined threshold **(‘set’)** the forward function **(‘forward’)** moves the robot forward. If the distance is less than or equal to the threshold the robot stops and calls a function called check side **(‘check_side’).**

3.2.2 Check Side Function

When the robot stops the check side function rotates the servo motor (to which ultrasonic is attached) to check the left and right sides. The Ultrasonic sensor scans the right side and measures the distance (distance_R) then scans the left side to measure the distance (distance_L). The ‘CompareDistance()’ is called to compare the distance of left side and right side and determine which side has more space. The robot turns to that side which has more space.

3.3 Section 3

Hand tracking Robot

3.3.1 Code Implementation

Note: Comments are part of code explanation

```
#include<NewPing.h>
#include<Servo.h>
#include<AFMotor.h>
#define RIGHT A2   // Right IR sensor connected to
analog pin A2 of Arduino Uno:
#define LEFT A3   // Left IR sensor connected to
analog pin A3 of Arduino Uno:
#define TRIGGER_PIN A1 // Trigger pin connected to
analog pin A1 of Arduino Uno:
#define ECHO_PIN A0   // Echo pin connected to
analog pin A0 of Arduino Uno:
#define MAX_DISTANCE 200   // Maximum ping distance:
unsigned int distance = 0;    //Variable to store
ultrasonic sensor distance:
unsigned int Right_Value = 0; //Variable to store
Right IR sensor value:
unsigned int Left_Value = 0;  //Variable to store
Left IR sensor value:
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE);
//NewPing setup of pins and maximum distance:
AF_DCMotor Motor1(1,MOTOR12_1KHZ);
AF_DCMotor Motor2(2,MOTOR12_1KHZ);
AF_DCMotor Motor3(3,MOTOR34_1KHZ);
AF_DCMotor Motor4(4,MOTOR34_1KHZ);
 Servo myservo; //create servo object to control the
servo:
 int pos=0;     //variable to store the servo
position:
```

Variable Declaration

Create Motor Objects

Figure 14. Code Implementation, Hand Tracking Robot

```
void setup() { // the setup function runs only once
when power on the board or reset the board:
Serial.begin(9600); //initailize serial
communication at 9600 bits per second:
myservo.attach(10); // servo attached to pin 10 of
Arduino UNO
{
for(pos = 90; pos <= 180; pos += 1){     // goes from
90 degrees to 180 degrees:
  myservo.write(pos);                     //tell servo
to move according to the value of 'pos' variable:
  delay(15);                              //wait 15ms
for the servo to reach the position:
}
for(pos = 180; pos >= 0; pos-= 1) {      // goes from
180 degrees to 0 degrees:
  myservo.write(pos);     //tell servo to move
according to the value of 'pos' variable:
  delay(15);                              //wait 15ms
for the servo to reach the position:
  }
for(pos = 0; pos<=90; pos += 1) {        //goes from
180 degrees to 0 degrees:
  myservo.write(pos);      //tell servo to move
according to the value of 'pos' variable:
  delay(15);                           //wait 15ms for
the servo to reach the position:
  }
}
   pinMode(RIGHT, INPUT); //set analog pin RIGHT as
an input:
   pinMode(LEFT, INPUT);  //set analog pin RIGHT as
an input:
}
```

Variable

Declaration

Figure 15. Code Implementation, Hand Tracking Robot

```
// the loop function runs forever
void loop() {

delay(50);                                   //wait
50ms between pings:
distance = sonar.ping_cm(); //send ping, get
distance in cm and store it in 'distance' variable:
Serial.print("distance");
Serial.println(distance);                        //
print the distance in serial monitor:
    Right_Value = digitalRead(RIGHT);            //
read the value from Right IR sensor:
    Left_Value = digitalRead(LEFT);              //
read the value from Left IR sensor:
Serial.print("RIGHT");
Serial.println(Right_Value);      // print the right
IR sensor value in serial monitor:
Serial.print("LEFT");
Serial.println(Left_Value);          //print the
left IR sensor value in serial monitor:
if((distance > 1) && (distance < 15)){
//check whether the ultrasonic sensor's value stays
between 1 to 15.
   //If the condition is 'true' then the statement
below will execute:
  //Move Forward:
  Motor1.setSpeed(130);  //define motor1 speed:
  Motor1.run(FORWARD);   //rotate motor1 clockwise:
  Motor2.setSpeed(130);  //define motor2 speed:
  Motor2.run(FORWARD);   //rotate motor2 clockwise:
  Motor3.setSpeed(130);  //define motor3 speed:
  Motor3.run(FORWARD);   //rotate motor3 clockwise:
  Motor4.setSpeed(130);  //define motor4 speed:
  Motor4.run(FORWARD);   //rotate motor4 clockwise:
}else if((Right_Value==0) && (Left_Value==1)) {
//If the condition is 'true' then the statement
below will execute:

  //Turn Left
  Motor1.setSpeed(150);  //define motor1 speed:
  Motor1.run(FORWARD);   //rotate motor1 clockwise:
  Motor2.setSpeed(150);  //define motor2 speed:
  Motor2.run(FORWARD);   //rotate motor2 clockwise:
  Motor3.setSpeed(50);  //define motor3 speed:
  Motor3.run(FORWARD);  //rotate motor3 clockwise:
  Motor4.setSpeed(50);  //define motor4 speed:
  Motor4.run(FORWARD);  //rotate motor4 clockwise:
  delay(150);
```

Loop Function

IR Sensor

Ultrasonic Condition

1st Condition

IR Sensor

2nd Condition

IR Sensor

Figure 16. Code Implementation, Hand Tracking Robot

```
}else if((Right_Value==1)&&(Left_Value==0)) {
//If the condition is 'true' then the statement
below will execute:

  //Turn Right
  Motor1.setSpeed(50);  //define motor1 speed:
  Motor1.run(FORWARD);  //rotate motor1 clockwise:
  Motor2.setSpeed(50);  //define motor2 speed:
  Motor2.run(FORWARD);  //rotate motor2 clockwise:
  Motor3.setSpeed(150);  //define motor3 speed:
  Motor3.run(FORWARD);   //rotate motor3 clockwise:
  Motor4.setSpeed(150);  //define motor4 speed:
  Motor4.run(FORWARD);   //rotate motor4 clockwise:
  delay(150);

}else if(distance > 15) {//If the condition is
'true' then the statement below will execute:
  //Stop
  Motor1.setSpeed(0);    //define motor1 speed:
  Motor1.run(RELEASE);   //stop motor1:
  Motor2.setSpeed(0);    //define motor2 speed:
  Motor2.run(RELEASE);   //stop motor2:
  Motor3.setSpeed(0);    //define motor3 speed:
  Motor3.run(RELEASE);   //stop motor3:
  Motor4.setSpeed(0);    //define motor4 speed:
  Motor4.run(RELEASE);   //stop motor4:
}
}
```

3rd Condition

IR Sensor

4th Condition

IR Sensor

Figure 17. Code Implementation, Hand Tracking Robot

CHAPTER 4

METHODOLOGY

4.1 Introduction

Line follower robots need an efficient control strategy. An efficient control system is the heart of the robot because it determines the overall performance of the line follower robot in terms of stability, its ability to navigate the predefined line autonomously, and give well response to dynamic conditions. Different methodologies have been adopted by different researchers and all have the same goal to obtain the stable performance of the robot. Controlling strategies for making line-follower robots has some challenges. Whenever the sensors observe that the robot has lost track the control system of the robot will try to overcome that error and try to bring the robot on track but if the control system is inefficient the will oscillate across the line and will be unable to become stable and gain the speed. So to avoid these circumstances we need an intelligent and well effective control system for line follower robots.

Considering all the challenges, we adopted a proportional integral derivative (PID) controller strategy for developing a fast, precise, and stable robot using a freenove starter kit for Arduino. This approach is very useful in dealing with the difficulties of controlling a robot and achieves a high level of precision and adaptability in the robot’s movement.

PID controllers play a very important role in industrial applications, control system designing for robots, speed control systems, and temperature control systems. The special ability of a PID control system is its feedback mechanism which makes it able to improve the stability and performance of dynamic systems.

4.2 PID Controller

A proportional integral derivative (PID) controller is a smart way of designing control systems for industries, robots, temperature control, and MPPT controllers. Almost 95% of industries use PID control systems. The PID controller acts like the brain of the robot because it controls all of the robot’s actions and makes decisions.

4.2.1 Working Principle of PID Controller

The proportional integral derivative (PID) controller uses a feedback control mechanism and operates on the principle of minimizing the error between the set point and the actual value of the system and the difference between these values is called error. The controller archives the desired response with the help of its three components proportional (P), integral (I), and derivative. These three components work like three friends, every one contributes to the overall control signal and enables the system to overcome the disturbance in the control system. each component is explained in detail in the following section.

4.2.2 Proportional component (P)

The term proportional in the PID controller is corresponding to the current error. The robot is outfitted with a sensor array. The sensor array computes the difference between the desired location (the middle of the line) and the actual position observed by the sensor. The difference between the desired position and the actual position is known as an error. The proportional term immediately responds to the error and determines the correction needed to fix the error. In the case of a line follower robot, when the robot goes off the line the proportional term tries to bring it back to the line. if the robot is to the right of the line the proportional term will drive the robot to the left and vice versa. using only proportional term may result in steady-state error. The steady-state error exists when the system is in a steady state. It is a constant deviation between the set point and the actual point of a system in a steady state.

$$\boldsymbol{P(t) = K_p \times e(t)} \quad \text{(Equation 2)}$$

Where

- $P(t)$ is the proportional term output at time t.
- K_p is the proportional gain.
- $e(t)$ is the current error.

4.2.3 Integral Component (I)

The integral term eliminates the steady state error that exists in the steady state system despite the proportional correction. The integral term adjusts the output based on the integral of the error. For example, the robot tends to drift to one side the integral will try to fix this and it makes sure that the robot follows the line perfectly.

$$\boldsymbol{I(t) = K_i \times \int_0^t e(t)dt} \qquad \text{(Equation 3)}$$

Where

- $I(t)$ is the integral term output at time t.
- K_i is the integral gain.
- $e(t)$ is the error at time t.

4.2.4 Derivative Component (D)

The derivative term describes the future behavior of the robot based on the rate of change of the error. The derivative term eliminates damping oscillations and overshoots from the control system. The derivative term predicts the movements of the robot like how fast it's going away or toward the line. If the robot approaches to line so quickly the P term, makes it slow down and eliminates the damping oscillations.

$$\boldsymbol{D(t) = K_d \times \frac{de(t)}{dt}} \qquad \text{(Equation 4)}$$

Where

- $D(t)$ is the derivative term output at time t.
- K_d is a derivative gain
- $\frac{de(t)}{dt}$ is the derivative of the error with respect to time.

4.2.5 PID Controller Output

The total output u(t) of the PID controller is the sum of the three terms Proportional, Integral, and Derivative.

$$\boldsymbol{u(t) = P(t) + I(t) + D(t)} \qquad \text{(Equation 5)}$$

The overall output response of the PID controller defends the tuning of its three constants K_p, K_i and K_d. Adjusting proper values for these three constants will give the desired response and will minimize the damping oscillations and overshoots.

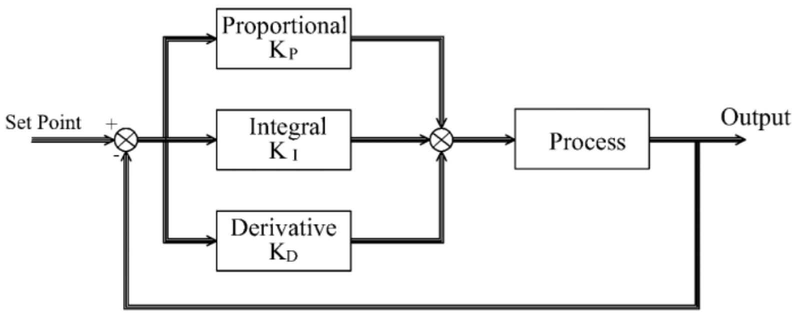

Figure 18. Closed loop operation of PID controller

The above figure shows the closed loop operation of the PID controller. The figure illustrates the functions of the different components and the structure of the PID controller. The first summing junction calculates the error. The proportional, integral, and derivative terms try to eliminate the error in their way as discussed in the above discussion. The total output of the PID controller is the sum of the P, D and I terms.

The feedback loop plays a vital role in control systems. Feedback control systems are always more stable than the others. Feedback loop feedback the output and finds the difference between the set point and the feedback signal. This error signal represents the deviation of the system response from the desired set point. Based on this error the system always tries to correct itself.

4.3 Enhanced navigation through Obstacle Avoidance

The navigation system of the line follower robot can be enhanced by adding a servo moor and ultrasonic sensor. By adding these two components and make some in the code for these components the line follower robot become able to avoid obstacles in coming in his path and come back to the track. Traditional line follower robots lack the ability to navigate around obstacles and the lack of this feature limit them in dynamic environments.

4.3.1 Function of Servo motor

Servo motor plays an important role to steer the robot and maintain it on the track, but here it is used to turn the ultrasonic sensor (mounted on it) 90° in either direction and total of 180°. When an obstacle is detected in the path, the servo motor turns the ultrasonic sensor to analyze the left and right side for alternate path.

Figure 19. Servo motor

4.3.2 Function of Ultrasonic Sensor

An ultrasonic sensor is an electronic device that emits ultrasonic waves to measure the distance of an object by receiving reflected weave through a receiver. The ultrasonic sensor has a transmitter and a receiver. The transmitter emits the sound waves and the receiver receives the reflected waves. The ultrasonic sensor is shown in the following figure, and the transmitter and receiver are represented by T and R respectively.

Figure 20. Ultrasonic sensor

The distance of the object from the sensor is calculated through the following equation.

$$\boldsymbol{Distance = \frac{Time \times Speed\ of\ the\ sound\ waves}{2}} \qquad \text{(Equation 6)}$$

The distance is calculated and the value is coveted to voltage, based on these data Arduino decides to turn the robot left or right.

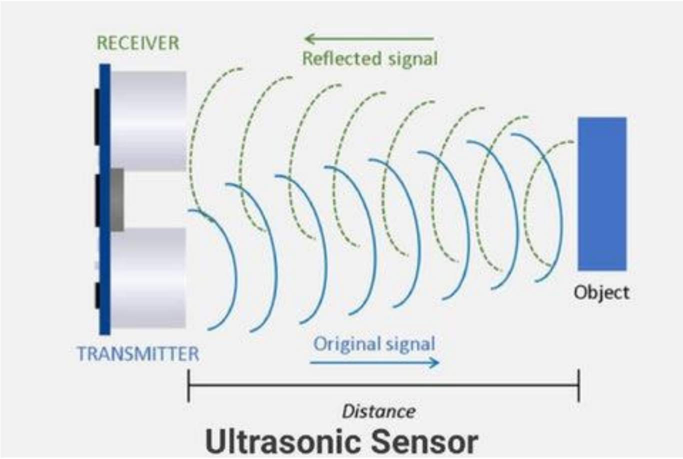

Figure 21. Ultrasonic Sensor

4.4 Hardware Setup

The hardware we assembled to make a functional line follower robot is,

- Chassis and frame
- DC motors and wheels
- 8 IR sensor array
- Arduino
- Motor driver

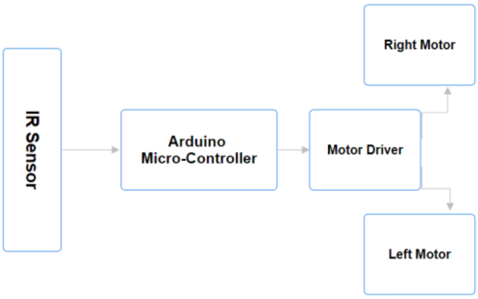

Figure 22. Line follower robot block diagram

4.4.1 Chassis and frame Assembly

The chassis and frame hold and give support to all the components. The selection of chassis is very important in the case of making a robot because the overall structure of a line follower robot depends on the frame and the structure directly affects the performance of the robot. The frame selected for this robot is strong enough to support the components and the battery weight

4.4.2 Motors and wheels integration

DC motors are mounted on the frame on the left and right sides. The wheels are attached to the motor shaft and the 3rd wheel is independent and mounted on the front and can rotate 360°. DC motors connected through wires to the L293D motor driver.

4.4.3 IR sensor array installation

An array of 8 IR sensors is mounted on the front of the robot chassis. Each sensor has three pins VCC (power), GND (ground), and OUT (output signal). The VCC pin of each IR sensor is connected to the Arduino 5v output of the Arduino. And GND pin of each sensor is connected to the GND pin of the Arduino. The output pins of the sensors are connected to the digital input pins of Arduino. Every sensor has two diodes one emits the IR light and the other receives the reflected light. The amount of received light is converted to voltage and sent to the Arduino.

4.4.4 Arduino Installation

Arduino is mounted on the frame, and IR sensors and motor drivers are directly connected to Arduino. we used Arduino UNO which is powered by ATMega328P microcontroller. Arduino UNO has 14 digital input output pins. It receives data from IR sensors based on that data it gives commands to the motor driver.

4.4.5 Motor Driver

The motor driver is installed between Arduino and DC motors. The motor driver receives commands from Arduino and controls DC motor speed and power. We used the L293D Motor Driver. L293D motor driver has 16 pins and two H bridge circuits to control 2 DC motors at once.

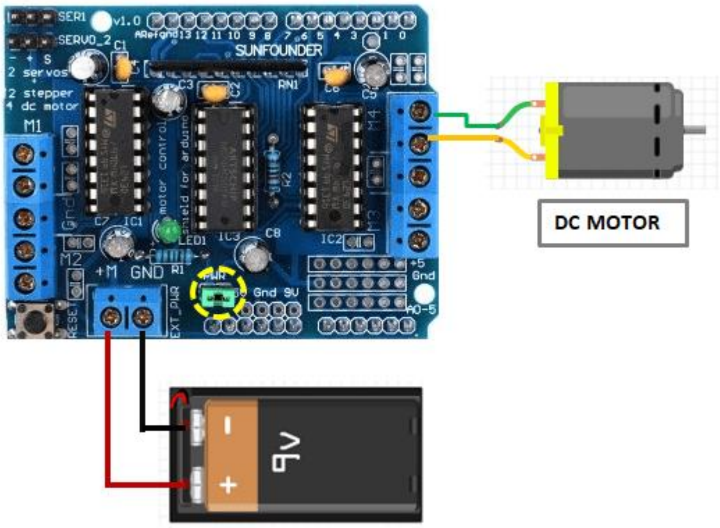

Figure 23. Motor driver, battery and DC motor connection

4.4.6 Battery connection

The battery is the power source for the robot. All the components of the robot use the battery as a power source. This battery is not rechargeable and can be replaced win down. The battery is connected to the VCC and GND of the Arduino.

4.4.7 Final model

The final model is given in figure below. This line follower robot model is fully functional and follows the predefined line.

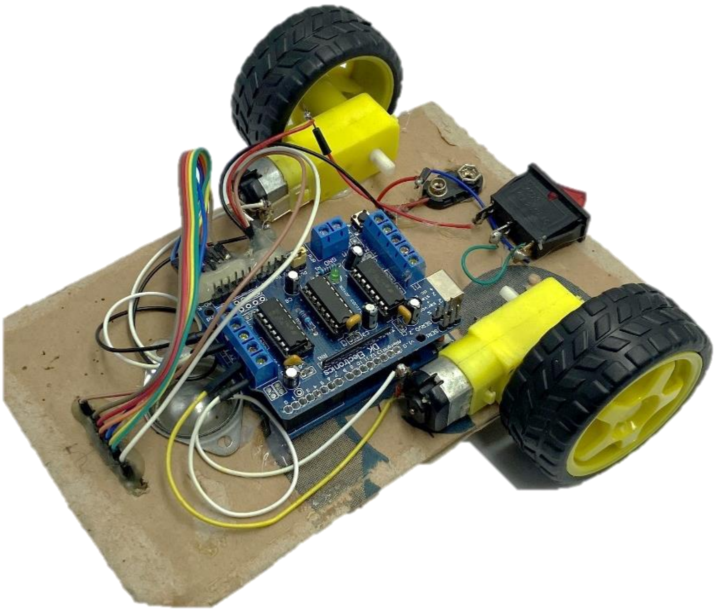

Figure 24. Final line follower robot model

4.4.8 Software Configuration

The software used to write a program for the Arduino microcontroller is Arduino IDE. The program defines the algorithm used. The program is written through Arduino IDE and when uploaded to the Arduino, it becomes capable of processing data, making decisions, and controlling the motors.

4.5 Calibration

Calibration is a crucial step while designing a robot. calibration involves the configuration of software parameters to match the specific characteristics of sensors, the environment, and the robot.

The calibration of the sensor threshold determines the difference between the lines and the background. It determines the analog voltage levels corresponding to the robot positions on the line, off the line, or at the edge of the line. These threshold values help the robot algorithm to make a correct decision based on the sensor readings. The environment is properly set for the robot like a black line is set as a path on the white surface.

4.6 System Architecture

This section of the thesis gives details about how the different components work together and communicate together to make a robot functional. A block diagram is shown in Figure 26 which in principle illustrates the interconnections between components within the robot. It depicts the flow of information from sensors to the microcontroller, and from the microcontroller to the motor drivers. This visualization is crucial for understanding the system's architecture.

All the components of a line follower robot are controlled by Arduino directly or indirectly. From the block it is obvious that Arduino is directly connected to IR sensors and the motor driver, motor drive is connected to two DC motors. The motor driver takes all the instructions from Arduino to control the motor's speed.

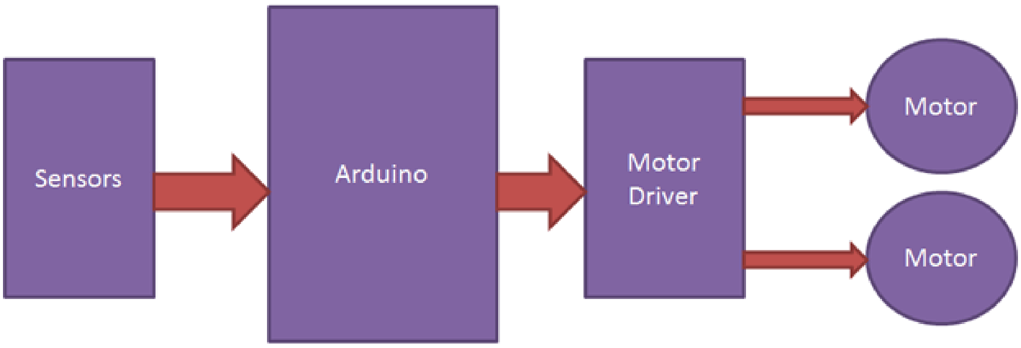

Figure 25. Block diagram of LFR information flow

4.6.1 IR sensor working principle

The eyes of the line follower robot are IR sensors. Using 8 IR sensors array increased the observing efficiency of the robot. Each IR sensor has an infrared emitter LED and an infrared photodiode as shown in the following figure.

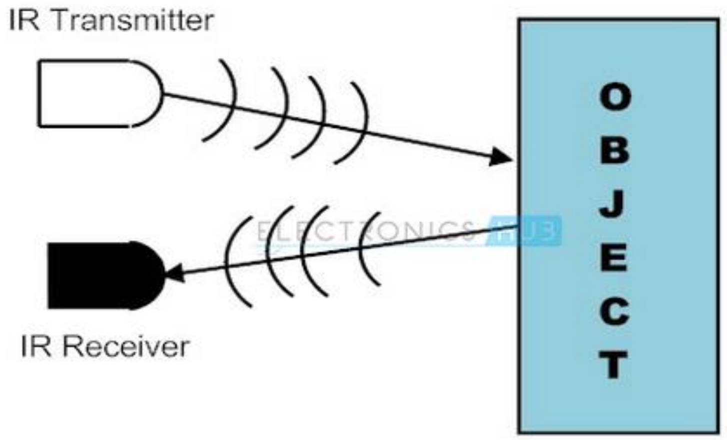

Figure 26. IR emitter and IR photodiode of IR sensor

The infrared emitting LED emits infrared light that is invisible to the human naked eye. Whenever the voltage is given to IR emitting LED it emits infrared light. When the reflected infrared light falls on the photodiode, it causes a change in the resistance of the photodiode. The extent of change in resistance depends on the amount of reflected light. The white surface reflects almost all of the incidence light and the black surface absorbs the light so these changes are sent to the Arduino in the form of high or low voltage values. Based on this data Arduino further decides what to do.

4.6.2 Raspberry Pi, Integration with Color Sensor

Raspberry Pi is a computing device featuring a processor capable of handling complex algorithms, in this case input from color sensors which will use a photodiode array to detect the color. It is “the brain” of the robot makes real time decisions, guiding the robot to stay along the track with precision.

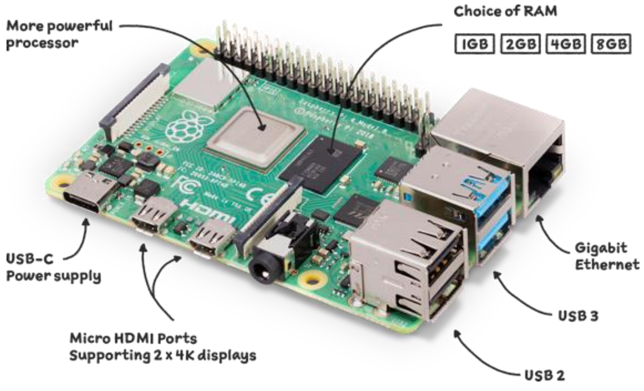

Figure 27. Raspberry Pi

Integrating a Raspberry Pi with a color sensor helps us to differentiate the main color with other colored lines that are on the same track. This process is achieved by doing some steps which will convert raw color sensor data to color temperature.

Figure 28. TCS34725 Color Sensor

By using RGB (red, green, and blue) values sensed by the sensor we estimate the color temperature of the light source that is illuminating and based on concept of blackbody radiation, a specific temperature can be mapped on the CIE 1931 Planckian Locus.

4.6.3 Calculation Process

First step is to translate the color data from a specific format to a universally recognized color space and this is by transforming RGB values of the sensor to

CIE1931 XYZ color space. This transformation is achieved by some specific set formulas which are known as color matching functions. In python, we can achieve it using conversion matrix:

$$\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{Y} \\ \boldsymbol{Z} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{0.412453} & \mathbf{0.357580} & \mathbf{0.180423} \\ \mathbf{0.212671} & \mathbf{0.715160} & \mathbf{0.072169} \\ \mathbf{0.019334} & \mathbf{0.119193} & \mathbf{0.950227} \end{bmatrix} \times \begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{G} \\ \boldsymbol{B} \end{bmatrix} \qquad \text{(Equation 7)}$$

There are various Python packages that can help you convert RGB values to the CIE 1931 XYZ color space, including ‘matplotlib’, ‘color-science’. Each of these libraries has its own functions or methods for handling color space changes. We might apply the transformation using mathematical formulas and a particular transformation matrix. If the RGB values are gamma-corrected, they are linearized before being converted to XYZ using the transformation matrix appropriate for our RGB color space, but this can get difficult, necessitating a thorough grasp of color science fundamentals.

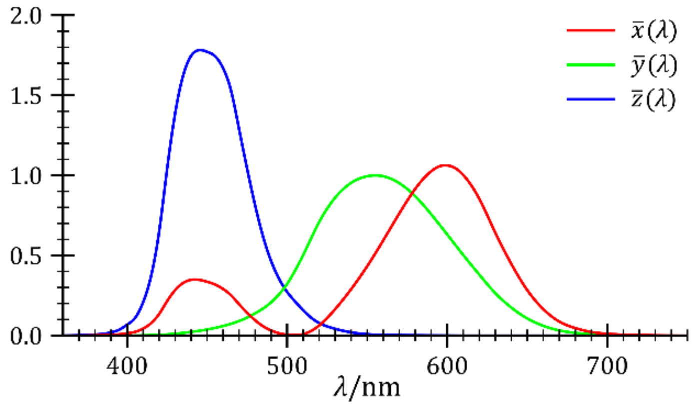

Figure 29. Color Matching Functions (not normalized curve)

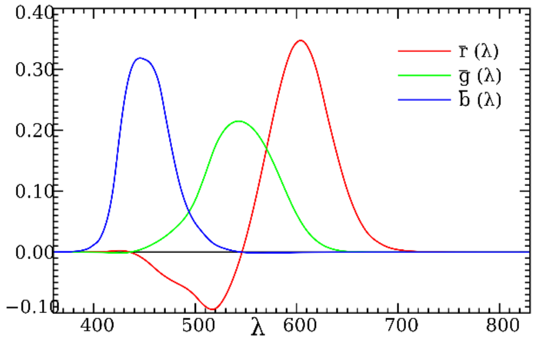

Figure 30. Normalized curve

After getting XYZ, we need to transform again these values into chromaticity coordinates so colors can be expressed independently. This transformation is done by normalizing the X, Y and Z values, the normalized values, x and y, chromaticity coordinates are calculated as below:

$$\boldsymbol{x = \frac{X}{X+Y+Z}} \qquad \boldsymbol{y = \frac{Y}{X+Y+Z}} \qquad \text{(Equation 8)}$$

Last thing that we are going to do is comparison between chromaticity coordinates and the Planckian locus CIE1931, this is called Robertson's method where it will find the closest point to our (x and y) coordinates and that give the color temperature.

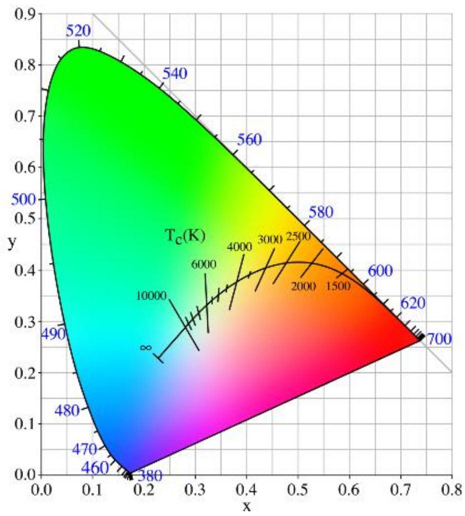

Figure 31. CIE 1931 Color space, x and y chromaticity diagram

4.6.4 Data processing by Arduino

Arduino UNO has 8bit ATmega328P microcontroller. Arduino UNO has a crustal oscillator, voltage regulator, and other components to support the microcontroller. Arduino UNO has 14 digital input/output pins of which 6 can be used as PWM output and 6 are analogue inputs. These pins can be used by directing them with special commands in the Arduino program. Arduino uses the decision-making algorithm present in the code uploaded to it. Through the algorithm, Arduino sends the control signal to the motor driver to control the DC motor operation.

4.6.5 Motor controls signal by Arduino to L293D

Control signals come from Arduino to the L293D motor driver. Arduino employs PWM (pulse width modulation) to regulate the speed of DC motors. Arduino controls the voltage by rapidly turning on and off the electricity. The average voltage provided to DC motors is determined by the duty cycle, or the ratio of time the signal is on to time it is off in a certain period of time. The PWM square wave has a specific frequency. if we assume the duty cycles in percentage then when the duty cycle is 100%, the pulse is high and the motors receive full power and rotate at full speed while the 0% means the pulses are low and the DC motors are receiving o voltage so the motors are stopped.

Arduino has an 8-bit register to control the PWM signal. We have about 255 voltage values between 0v and 5v. The 0 value signifies 0% duty cycle and 255 means 100% duty cycle, other duty cycles can also be obtained by intermediate values like 64 for 25%, 127 for 50%, and 191 for 75% duty cycle.

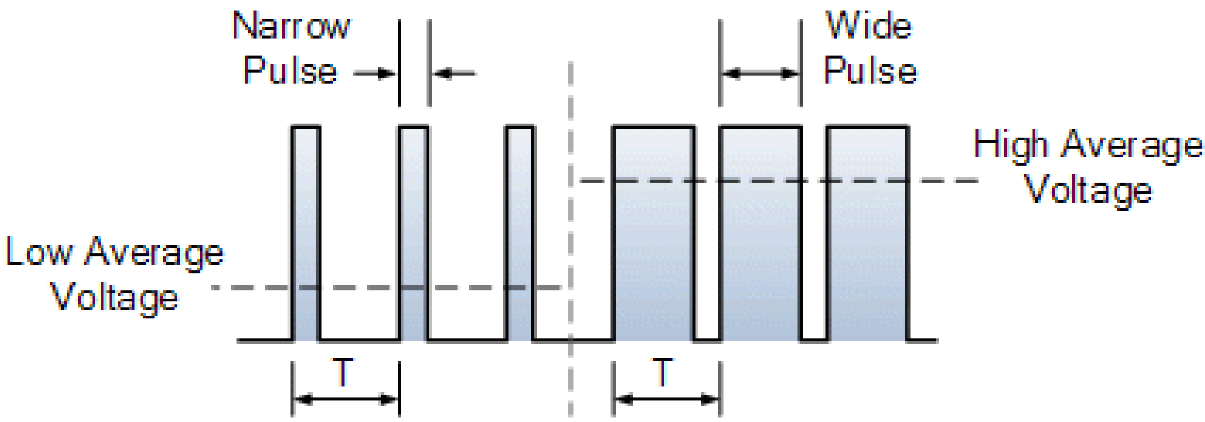

Figure 32. Voltage control by PWM signals

4.6.6 Operation of L293D motor driver

The motor driver receives the control signals from Arduino and decides the voltage and polarity for DC motors. This motor driver comes with a 16-pin DIP IC package and has two H bridges and a shift register. Figure below shows the motor driver, battery, and DC motor.

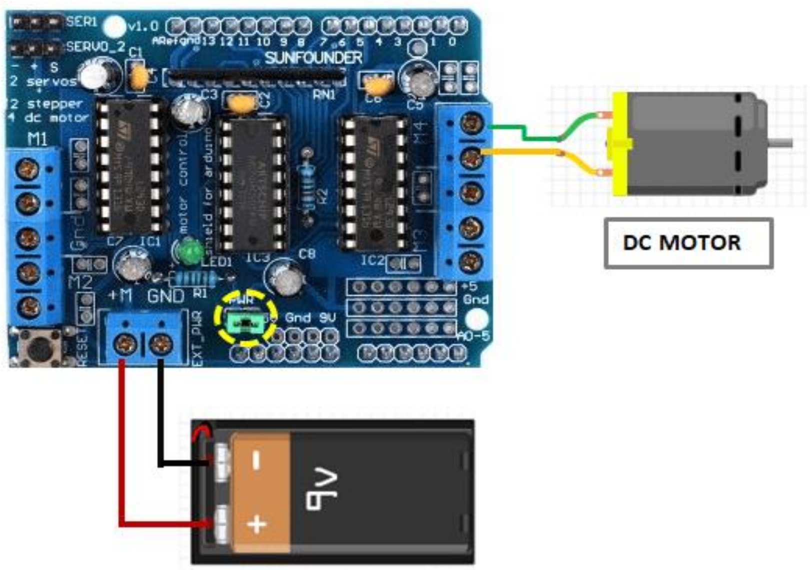

Figure 33. L293D motor driver connected with battery and DC motor

The figure above shows the pinout of the l293D IC. There is a total of 16 pins of which two pins are power supply pins VCC1 and VCC2.

- VCC1 is the power supply for the circuitry and its voltage should be 5V, usually, VCC1 is connected to the Arduino 5V.
- VCC2 is the power supply for the H bridge and its voltage ranges from 4.5V to 36V and is use to run the DC motor.
- There are four output terminals two controls two motors. One motor is connected to the output terminals OUT1 and OUT2 while another motor is connected to terminals OUT3 and OUT4.
- ENA and ENB are used to control the speed of the DC motors, when they are high the speed of the motor is high and when they get low the motors stop. Arduino sends PWM signals to these pins to control the speed of the motors.
- IN1, IN2, IN3, and IN4 pins control the direction of rotation of the motors.
- Pins 4 and 5 are the ground pins of the IC and provide ground connection as well as act as a heat sink for IC.

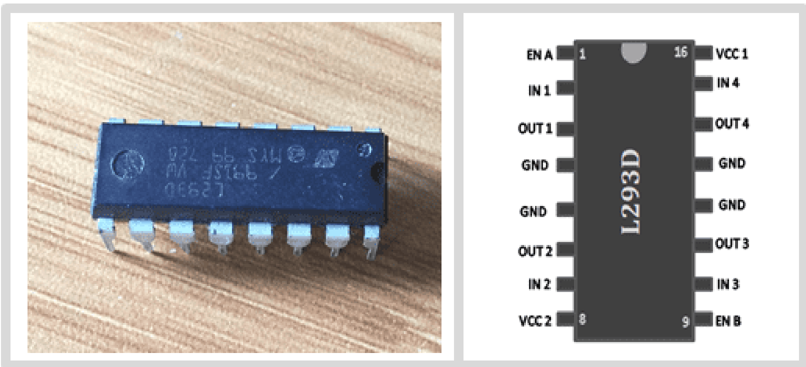

Figure 34. Motor driver IC

4.6.7 H-Bridge Working Mechanism

L293D motor drives have two H bridges. Each H bridge has four MOSFET transistor switches S1, S2, S3 and S4. At the same time, two particular switches must be activated to maintain the flow of current in a particular direction to rotate the motors in reverse or forward directions. The working of the H bridge is shown in the following figure.

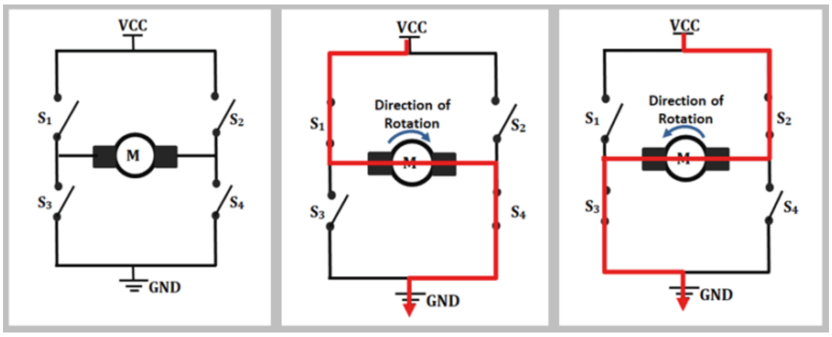

Figure 35. The working mechanism of H bridge

When all four switches are open, no electricity passes to the motor. When S1 and S3 are closed and S2 and S4 are open, the current flows from left to right via the motor, which rotates clockwise. Similarly, when switches S2 and S3 are closed and switches S1 and S2 are closed, the current flows from right to left and the motor rotates counterclockwise. When S1 and S2 or S3 and S4 are switched together, a phenomenon known as shoot-through occurs, which can destroy the transistor.

L293D motor driver has 2 H-Bridge circuits and controls 2 DC motors simultaneously. The four pins IN1, IN2, IN3, and IN4 control the switches of the H-Bridge circuit of the L293D motor driver module.

The rotation of the DC motors depends on the input values to these four IN pins. If IN1 is low and IN2 is high the motor will rotate in a particular direction say clockwise and if IN1 is high and IN2 is low, then the motor will rotate in the opposite direction say anticlockwise. The motor will stop rotating when both of the inputs have the same signal, when both IN1 and IN2 are both low, or when they are both high. The following table shows the corresponding direction of the motor based on the state of the inputs IN1, IN2, IN3, and IN4.

Table 1. The direction of rotation of motion based on the state of IN

State of IN 1	**State of IN 2**	**Direction of Rotation of Motion**
Low	Low	Stop
Low	High	Clockwise
High	Low	Anticlockwise
High	High	Stop

CHAPTER 5

MACHINE LEARNING ALGORITHM

5.1 Introduction to Machine Learning

Machine learning is a part of artificial intelligence (AI) that allows systems to learn and improve over time without being manually programmed. The premise behind machine learning is that algorithms can learn from data, find patterns, and make choices with little human interaction. This learning process is similar to how a human learns via experience, with a machine's "experience" coming from the data it is fed. As a machine learning model is provided with additional data, it may update and optimize its algorithms to improve performance, similar to how an electronic circuit modifies its parameters for peak performance. [13]

There are three main forms of machine learning:

1. Supervised Learning: The model learns from labelled data and makes predictions or judgments based on previous results. It's similar to how an electrical system is tuned using a given set of inputs and intended outputs.
2. Unsupervised Learning: The model learns from unlabeled data by identifying hidden patterns or inherent structures in incoming data. This is analogous to examining raw data in electronics to find underlying patterns without knowing what to search for.
3. Reinforcement Learning: The model learns via trial and error, making decisions based on input from its own actions and experiences. This is similar to adaptive systems in electronics, which modify their behavior to improve some type of input feedback. [14]

5.2 Introduction k-NN Algorithm, future work!

Unlike the first line follower, this advanced robot can handle more complex tasks and have higher efficiency in data computation. To achieve this kind of thing, Arduino Uno is not feasible because of the memory and limited processing power, so we have to integrate a Raspberry Pi. Here we will use machine learning algorithms, particularly the k-NN (k-Nearest Neighbors) which empowers the robot with enhanced decision making. This makes the robot to learn from the environment and improve decision making and adaptability.

k-Nearest Neighbors algorithm is a supervised simple, yet advanced in machine learning which is used for observation and classification of datasets. In simple terms, let's say we have plant with corn and see what will grow near it, we would predict that there will grow a corn. "k" it's a parameter that represents the number of nearest neighbors, to find nearest neighbors, algorithm has to calculate the distance between points by using some formulas which are known as Euclidian. [15]

$$\boldsymbol{Distance} = \sqrt{(\boldsymbol{p1} - \boldsymbol{q1})^{\boldsymbol{2}} + (\boldsymbol{p2} - \boldsymbol{q2})^{\boldsymbol{2}}} \qquad \textit{Equation 9.}$$

$$\boldsymbol{Distance} = \sqrt{(\boldsymbol{p1} - \boldsymbol{q1})^{\boldsymbol{2}} + (\boldsymbol{p2} - \boldsymbol{q2})^{\boldsymbol{2}} + \cdots +} \, (\boldsymbol{pn} - \boldsymbol{qn})^{\boldsymbol{2}} \qquad \textit{Equation 10.}$$

Equation 9: distance between two points in 2D space

Equation 10: distance in multidimensional space

In a study to predict real time bus travel using k-NN algorithm in India they were able to predict and perform better results. k-NN technique involves employing GPS technology to monitor buses in real time, estimating distances between stops using the haversine formula, and analyzing the spatial dynamics of bus movement. This study uses the k-Nearest Neighbors (k-NN) method, a kind of supervised machine learning, to examine historical journey time data, allowing the system to forecast future travel times based on comparable previous conditions. The study tackles unpredictability in urban traffic patterns, which has a substantial influence on bus journey times, by dividing data into distinct temporal segments, such as peak vs. off-peak hours and weekdays vs. weekends. [16]

The approach improves forecast accuracy by using a time-discretization mechanism, which divides the route into smaller chunks for more in-depth study. This technique enables more precise predictions that can respond to abrupt changes in traffic circumstances. Furthermore, the study incorporates Kalman filtering, a mathematical approach that is utilized in real time to modify forecasts by accounting for the inherent uncertainty and noise in GPS data and traffic circumstances. This dual method, which combines historical data analysis with real-time modification, provides a strong foundation for predicting bus trip times, demonstrating its potential to dramatically enhance public transportation efficiency in densely crowded metropolitan locations such as Chennai.

5.3 Object recognition

Now that our line follower is ready to detect any color, we can make the track more complex by adding a STOP sign, this complex duty offers us the opportunity to implement in our robot machine learning and object recognition by using features that STOP sign has, color-based segmentation. This is going to be a hard try with our only RGB color sensor since there are many factors that affects our sensor to fail, like light intensity. By integrating a camera to our Raspberry Pi, we can use shape recognition and since most of the signs are designed with a single background color it would be much easier for us to make the difference.

CHAPTER 6

RESULTS AND DISCUSSIONS

Both robots are presented in this thesis, Intelligent Line follower and Hand tracking robot. LFR was successfully practically implemented using Arduino UNO. The robots are programmed using Arduino IDE software and the PID algorithm is implemented in the programming code of LFR. The algorithm makes the robot capable of following the line precisely and with high accuracy. Even if the robot goes off the track it resets itself online.

The final model of the line follower robot is given in the following figure. This model successfully implemented and followed the predefined line with high accuracy and stability.

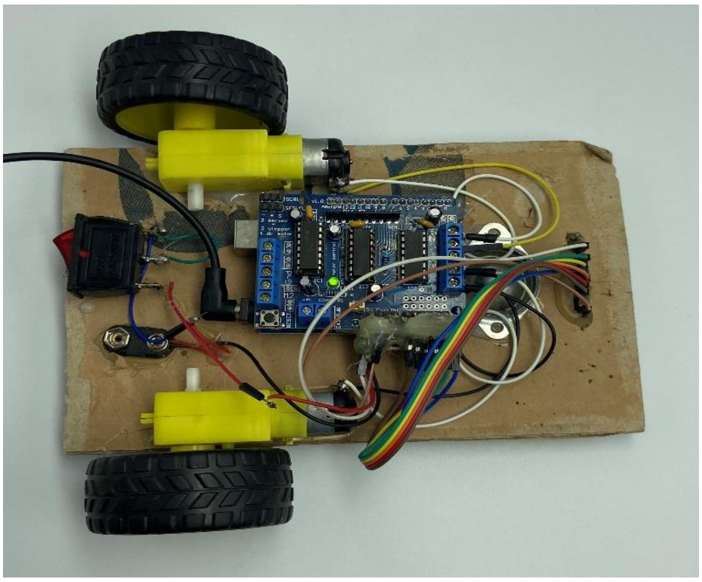

Figure 36. Finalized model of the presented Intelligent line follower robot

The creation of a hand tracking robot prototype utilizing the Arduino UNO to help elderly people in supermarkets is a big step toward improving the elderly's shopping experience and autonomy. This project not only highlights the practical application of robots in everyday life, but it also shows the possibility for combining basic, low-cost technologies to address real-world problems. The robot uses hand tracking technology to closely follow users without requiring physical touch, removing the need for elderly people to push heavy shopping carts, which may be physically and emotionally taxing. The project serves as a prototype that could be refined and expanded upon, with future iterations possibly incorporating advanced features such as obstacle avoidance, voice commands, and more sophisticated user interaction capabilities.

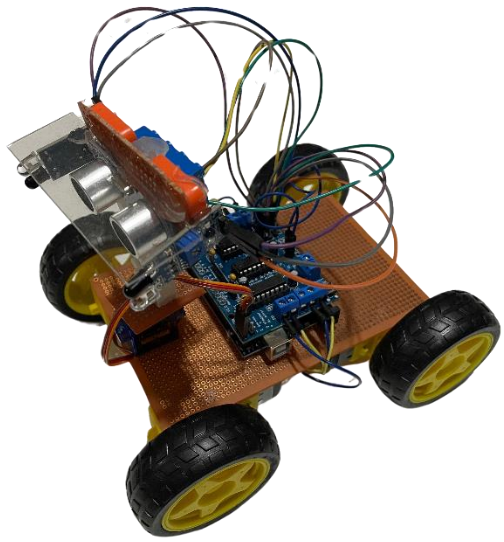

Figure 37. Finalized model of Hand Tracking Robot

6.1 Performance of the robot

The line follower robot gives outstanding performance in following a black line on a white surface. Through the systematic calibration of IR sensors and PID controller, the robot exhibits a precise and stable movement along the predefined path. The overall design of the robot is good but needs a little reduction in the size. The robots give consistence performance in various scenarios like straight lines, long turns, and sharp turns. Overall the model is successfully implemented and showcases very good performance.

The presented line follower robot is capable of handling the change in line width and ambient light conditions. The line follower robots maintain their stable performance in the presence of environmental disturbance.

The existing model of the line follower robot can be upgraded by integrating the raspberry pi microprocessor with obstacle avoidance feature. To add this feature a servo motor and ultrasonic sensor is needed. This feature enables the line follower robot to make real time decision in unpredictable and complex environment. Obstacle avoidance feature reduces the dependency of the robot on the predefined track. This feature increases the safety and stability because it minimizes the risk of collision and damage to the environment.

The presented model achieved the balance between speed and stability ensuring smooth movement. The fine-tuning of PID controllers results in a controlled and stable motion along the path. The algorithm minimizes overshooting and oscillations and maintains a robot on the track.

6.2 Applications and usage of line follower robot

There are many applications of these robots including medicine, transportation, military etc. The following are some applications of the line follower robot:

Our initiative's primary goal is to create and install robotic assistants specifically designed to meet the demands of the older population, particularly in large retail venues. This attempt aims to make it easier for customers to acquire and move things from the point of purchase to their automobiles or straight to their homes. A critical component of this goal is the development of a strong municipal infrastructure that supports and improves the operational efficacy of these robotic helpers.

In military operations, the use of modern sensors particularly intended to detect gas leaks represents a significant technical leap with the potential to transform safety regulations and reconnaissance missions. By incorporating these sensors into military equipment, soldiers may proactively identify and examine regions suspected of being contaminated by toxic gasses, eliminating the need to physically approach these potentially dangerous zones.

CHAPTER 7

CONCLUSIONS

7.1 Conclusion

This thesis presents a Intelligent Line follower robot which is a dynamic and simple robotic system designed to follow a predefined black line on a white surface using an IR sensor and a color sensor. The primary objective was to design such a robot capable of navigating a predefined route with high accuracy. The robot is made from the integration of different components; each component has its own rule in the functionality of a line follower robot. An array of 8 infrared sensors work as an eye of the line follower robot. IR sensors observe the position of the robot on the line and convert these positions and physical data to voltage signals. These signals are processed by Arduino, which acts as the brain of the line follower robot and real-time take decision. L293d motor driver receives the control signals from Arduino and controls the speed and direction of rotation of the DC motor. The line follower robot uses the PID controller algorithm for the controller robot's functions. The proportional, integral, and derivative terms work together to process the sensor data and make real-time decisions to control the speed of DC motors and maintain the robot on the line. The PID algorithm minimizes the error and maintains the robot on the track with high accuracy and stability. The results show that the finalized model was able to successfully follow the black line on the white surface with high accuracy. Line follower robots have a lot of industrial and domestic applications. These types of robots are adaptable to every daily life task such as house cleaning, transportation, carrying goods delivery services, etc.

7.2 Recommendations for future research

While the current iteration of the line follower robot demonstrates promising performance, there are opportunities for future enhancements. Potential areas of improvement include:

- Integration of advanced sensors for improved line detection in diverse environments.
- Implementation of machine learning algorithms for adaptive navigation and obstacle avoidance.
- Expansion of the robot's capabilities to handle dynamic, unstructured environments.
- Improvement in the design and overall shape can increase the performance like a robot having a balanced structure can easily turn in the short curves.
- Implementing a Raspberry Pi for more complex tasks such as in military where a camera is needed to check for the enemy.

These enhancements have the potential to further broaden the applicability and effectiveness of the robots in real-world scenarios.

REFERENCES

[1] B. K. Hamdi Alper Özyiğit, "Active suspension control for vehicles and numerical calculations by using artificial neural networks," Research Gate, Ankara, Turkey, 2001.

[2] L. K. D. A. Y. V. A. S. Narendra Kumar Sharivas, "Human Assistance Robot by Using Arduino," International Journal of Scientific Research in Computer Science, India, 2017.

[3] A. P. R. T. A. T. N. R. A. a. B. N. Pathak, "Line follower Robot for industrial manufacturing process," *International Journal of Engineering Inventions,* pp. 10-17, 2017.

[4] M. S. M. a. G. M. Pakdaman, "A line follower robot from design to implementation: Technical issues and problems," *The 2nd International Conference on Computer and Automation Engineering,* pp. 5-9, 2010.

[5] I. S. I. W. K. K. I. K. Kaiser F, "Line follower robot: Fabrication and accuracy measurement by data acquisition," *International Conference on Electrical Engineering and Information & Communication Technology,* pp. 1-6, 2014.

[6] K. A.-N. A. R. K. K. S. a. B. M. Hasan, " Sensor based autonomous color line follower robot with obstacle avoidance," *IEEE Business Engineering and Industrial Applications Colloquium (BEIAC),* pp. 598-603, 2013.

[7] H. A. Desk, "Mercedes-Benz short film explains microsleep detection technology of EQS EV," 2022.

[8] Ramya, "Integration of Artificial Intelligence in Automotive Embedded Systems," 2024.

[9] china: roboway.

[10] E. K. Saad Hussein, "LOW COST PORTABLE URINE ANALYSIS SYSTEM DESIGN BASED ON COLOR," Kocaeli, Turkey, 2020.

[11] osoyoo, "Micro bit Lesson — Using the Ultrasonic Module," osoyoo, 2018.

[12] Electronicsforu, "Understanding IR Sensors and IR LEDs: Functions, Differences, and Applications," Electronicsforu, 2023.

[13] I. H. Sarker, "Machine Learning: Algorithms, Real-World Applications and Research," Springer Nature Singapore Pte Ltd 2021, 2021.

[14] G. Bonaccorso, "Machine Learning Algorithms," Packt Publishing, Birmingham, 2017.

[15] Z. Z, "Introduction to machine learning: k-nearest neighbors," 4 June 2016.

[16] R. J. S. S. A. &. L. V. B. Anil Kumar, "Real time bus travel time prediction using k-NN," Taylor and Francis Online, 2017.

[17] "SriTu Hobby," [Online]. Available: https://srituhobby.com/how-to-make-a-line-follower-robot-using-arduino-and-l298n/.

[18] "ictransistors.com," GNS components Limited , [Online]. Available: https://www.ictransistors.com.

[19] "circuit digest," [Online]. Available: https://circuitdigest.com.

[20] "mytectutor.," [Online]. Available: https://mytectutor.com/l293d-motor-driver-shield-for-arduino/.

[21] "robocraze," [Online]. Available: https://robocraze.com.

[22] "Electronic Tutorials," [Online]. Available: https://www.electronics-tutorials.ws/.

[23] "mytectutor," [Online]. Available: https://mytectutor.com/.

[24] "mytectutor," [Online]. Available: https://mytectutor.com/.

[25] "mytectutor," [Online]. Available: https://mytectutor.com/l293d-motor-driver-with-arduino-controlling-dc-and-stepper-motors/.

[26] A. M. A. a. A. H. Ali, " Tuning PID Controllers for DC Motor by Using Microcomputer," *Int. J. Appl. Eng. Res,* pp. pp.202-206, 2019.

[27] "circuitdigest," [Online]. Available: https://circuitdigest.com.

[28] "mytectutor," [Online]. Available: https://mytectutor.com/.

[29] "robocraze," [Online]. Available: https://robocraze.com/.

[30] "makerselectronics," [Online]. Available: https://makerselectronics.com/product.

[31] "hackster.io," [Online]. Available: https://www.hackster.io.

[32] r. pi.

[33] D. Das, "circuitdiges," 28 December 2022. [Online]. Available: https://circuitdigest.com/microcontroller-projects/interface-l293d-motor-driver-with-arduino.

APPENDIX

The figure shown below represents the circuit diagram of Arduino, L293D IC and DC motors. The connection between these components is shown by virtual wires.

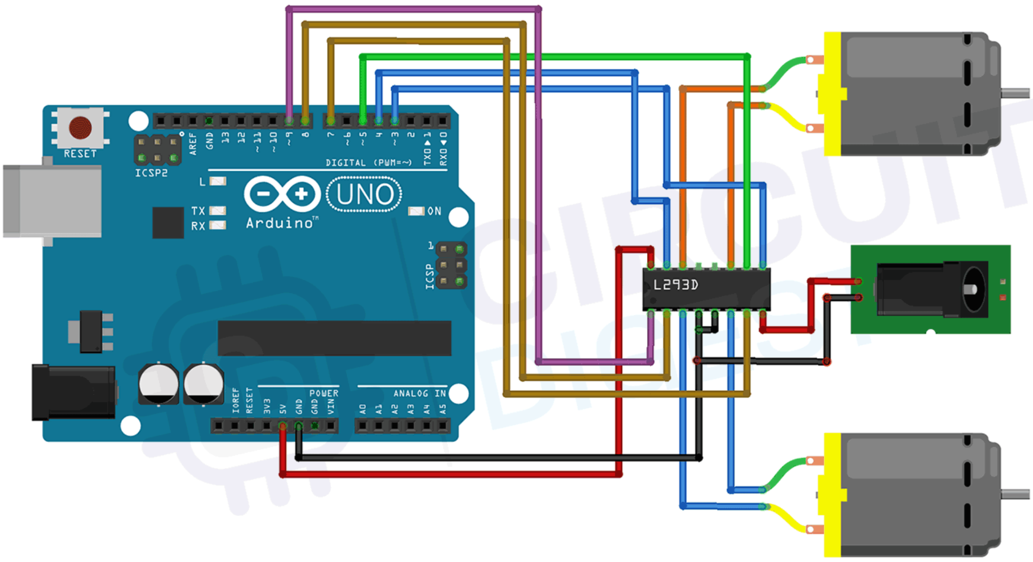

Figure 38. Circuit diagram of DC motor with Arduino and L293D IC

In the figure below a pinout of the L293D motor driver IC is shown. Each pin of the IC is properly labeled.

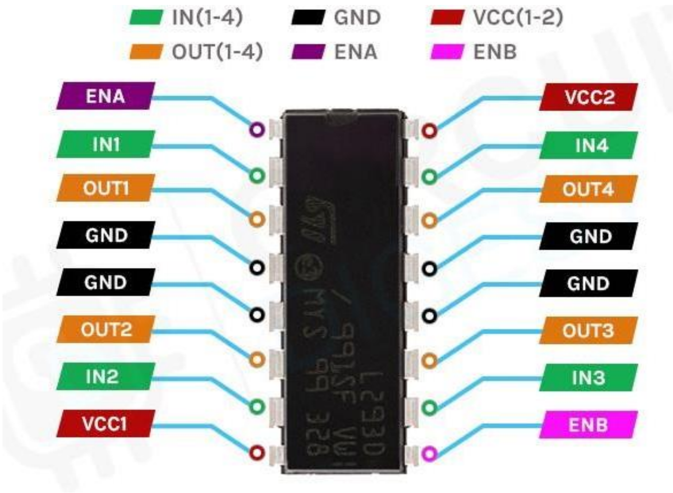

Figure 39. Pinout of L293D motor driver IC

A labeled diagram of the L293D motor is given in the following figure.

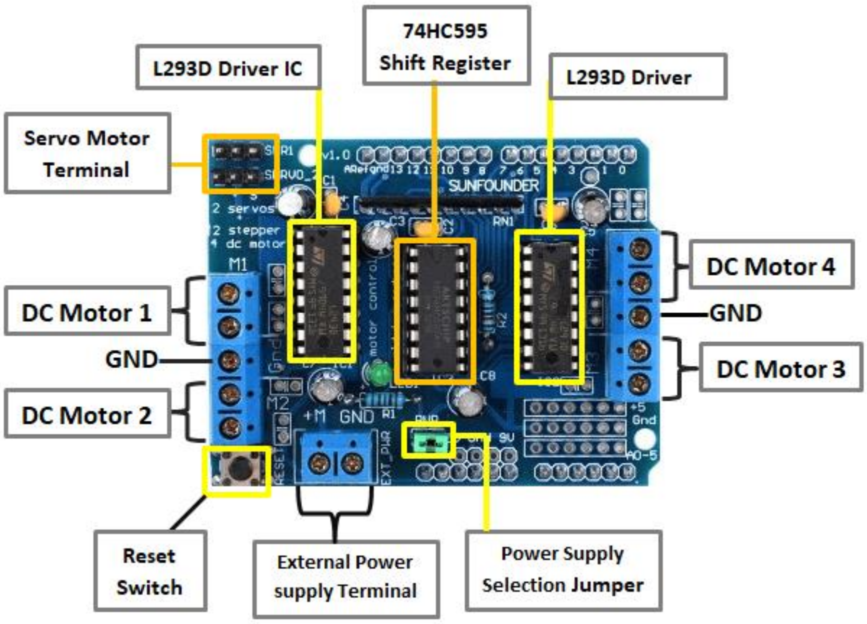

Figure 40. Labeled diagram of L293D motor driver