MSS-M-1 Case Study Embedded Control Solutions

Line Following Robot

SUBMITTED BY

DHANANJAY GAJERA -22207436
HIREN MORADIYA- 22202824
MILAN BUHA -12200060
NAVEEN KUMAR MOORAM-12201340
VISHVA MUNJANI-22204929

FROM

ARTIFICIAL INTELLIGENCE FOR SMART SENSORS AND ACTUATORS

UNDER THE GUIDANCE OF

Prof. Schmid Joseph

Faculty of Applied Natural Science and Industrial Engineering
Deggendorf Institute of Technology
Cham, 93413

Development and Implementation of Line Following Robot

Vishva Munjani, Naveen Kumar Mooram, Hiren Moradiya, Dhananjay Gajera, Milan Buha

Abstract: **This paper discusses the algorithm for a robot that can follow a path and can be any contrasting color. The line controlling components are more adaptable, produce more accurate results, and handle material things. As it provides many benefits in our lives, the working environment of the vehicles ranges from small offices with carpet floors to enormous harbour dockside areas. The goal of this project is to create a working prototype of a black line tracking robot that can drive on a flat, white surface while tracking a black line with the help of two DC gear motors, two driving wheels, and a third wheel that rotates the vehicle 360 degrees. With the help of the ESP32 microcontroller, the prototype can synchronize the commands from the sensors and manage the delay by moving along the black line on the floor. The microcontroller is connected to a sensor that continuously reflects the surface conditions, an IR sensor that controls the movement and direction of the vehicle, acting as the stem, and a distance sensor that acts as brakes when required, in order to follow the line. The robot's control algorithm uses the sensor input to adjust the robot's movement and keep it on the correct path. The project aims to demonstrate the capabilities of the robot in following a path and the potential applications in industrial automation and transportation. As a result, we have to showcase the robot's ability to navigate a predetermined path and to execute the planning, design, development, and evaluation of the outcome.**

Keywords: Line Following Robot, ESP32, IR Sensor, L298N Motor driver

1. INTRODUCTION

Line following robots are automated vehicles that are designed and track the black line predetermined which is black color marked by a line on the ground. These robots have a wide range of applications, including manufacturing, material handling, and education. In this report, we present the design and implementation of a line following robot using two, an ESP32 microcontroller, IR sensors and an L298N motor driver. The ESP32 is a low power consumption, high performance Arduino esp32 microcontroller with multiple peripherals and wireless communication capabilities, making as well as suited for building robotics applications. The L298N motor driver is a dual H-bridge motor driver that allows the robot to control the speed and direction of two motors. The IR sensors were used to detect the line using IR module and provide input to the microcontroller. We used the Arduino programming language to write a program that reads sensor data and controls the motors to keep the robot on the line.

The line following robot was designed with the goal of creating a reliable and accurate robot that can follow a line under a variety of lighting conditions. In the following sections of this report, we describe the hardware and software components of our line following robot in detail, as well as the steps we took to build and test it. We also discuss the challenges we faced and the lessons we learned during the project. In addition, we present the results of our tests and provide an analysis of the robot's performance.

1.1 METHODOLOGY

First, we proceeded by reviewing the literature and gathering background knowledge on line-following robots and their numerous parts, including sensors, microcontrollers, and motors. Start by designing the robot's hardware and software, including the circuit schematic and programming code, after reading the relevant literature.

The reasoning behind the robot's intelligence is derived. Using the software Arduino® 2.0. 3. The Arduino has the code burned onto it. The computation software Ki CAD® is used to verify the precision and viability of the program, as well as the electronic components. After a successful simulation, the software is put into action in the hardware.

The programming, electrical, and electronic portions are finished, and then stable, dependable, and adaptable mechanical design and production are then completed. Finally, the system is evaluated, and any errors are eliminated.

1.2 BACKGROUND

A sort of autonomous robot called a line following robot is made to go along a predetermined path that is often indicated by a line drawn on the ground or another flat surface. These robots have sensors that can detect the line and provide instructions to the robot's motors to stay on it, such as infrared or optical sensors. Many different industries, including manufacturing, logistics, and transportation, use line-following robots.

As technology grows in importance in today's world, it is critical to understand not only how to use it, but also how to create it. As an engineer, you need deep knowledge of other disciplines. Most projects are confined to a specific area. It would also limit their own ideas and creativity. This project combines mechanical, electronic, electrical, and programming skills, encouraging connections between various disciplines rather than studying individual subjects in isolation. Provides a visual understanding of mathematics and science. Build your logical thinking.

Although there are many uses for line following robots, there are still many challenges that need to be solved before they can be made more effective, dependable, and efficient. This project attempts to address one or more of these problems in order to contribute to ongoing research.

1.3 PURPOSE OF STUDY

Carriers are used in business to move things across production lines, which are frequently spread throughout various buildings on different blocks. Trucks or vehicles with human drivers were frequently utilized. This part, which was unreliable and useless, was the assembly line's weakest link. By using carts that follow a line rather than constructing railroad tracks, which is both expensive and inconvenient, the plan aims to simplify this component of the economy.

1.4 SCOPE OF THE PROJECT

The goal of this project is to develop a line-following robot that can precisely follow a line in challenging circumstances.

The robot can be improved more so that the user can select a dark line on a white background. Instead of using a user interface, the robot might be programmed to determine what kind of line it is. In order to drive a convectional vehicle without a differential steering system, the motor control could be adjusted. It is possible to make the robot all - wheel drive. Additional sensors might be added to the robot to help it recognize obstacles and, if feasible, avoid them so it can return to the line. In other words, it needs to be able to forecast the path beyond the obstruction. Additionally, speed control could be added. It is possible to integrate position and distance sensing devices that can communicate data to a mother station for use in locating a lost carrier.

1.5 OBJECTIVE

- To develop a line-following robot that can precisely follow a specified line using motors, sensors, and a microcontroller.
- To evaluate the robot's performance under various circumstances, such as variable line widths, surface textures, and lighting.
- To create a control algorithm capable of successfully guiding the robot down a line.
- The robot must be robust to environmental parameters like noise and light. It must be easy to calibrate the modest amount of darkness along line Scalability must be a key design principle.
- More sensors provide reliable solutions for the robot.

2. LITERATURE REVIEW

A lot of work and effort has been put into developing systems in case study that allow an IR sensor Controlling robot to use a visual system to follow a designated path. Unsurprisingly, most of this research has focused on creating or altering developing different application related component so they can operate without human supervision on regular roadways which is black path.

Smaller devices typically opt for one of two methods. The first process includes developing a mathematical model of the vehicle and its surrounds, simulating its outputs, and then putting it to a robot specifically created for the task. In the second approach, a visual servo system and a kinematic model were integrated; and once again, the robot is typically constructed around the technique of resolving the issue. Simpler models and techniques, such visual serving, are implemented to minimize the processing load so because processing resources available to these robots are quite restricted because of their size.

An ESP32 Microcontroller based project give advance model of Arduino microcontroller system and built 3 sub system. ESP32 microcontroller which control the process of controlling motor and IR sensor for motor controlling model L298N transfer output for the motor operate speed and PWM signal. There are four conditions for the controlling motor because in zigzag path it will react differently so that enhancement of motor transferring command for the robot in different application.

3.TECHNICAL REQUIRMENTS

3.1 ESP32 MICROCONTROLLER

Espressif Systems created a microcontroller called the ESP32. It is an Internet of Things (IoT) system-on-a-chip (SoC) that is inexpensive and low-power (IoT). Embedded system and other applications. The ESP32[2] has a dual-core processor with both Wi-Fi and Bluetooth capabilities, making it well-suited for a variety of wireless applications. It also has several peripherals and features, including:

Multiple communication interfaces: The ESP32 has multiple UART, SPI, I2C, and I2S, SDIO, ETH, PWM interfaces, which can be used to communicate with a wide range of external devices.

Multiprotocol support: The ESP32 supports a variety of wireless protocols, including Wi-Fi, Bluetooth, and BLE (Bluetooth Low Energy). This makes it a versatile platform for building IoT and other wireless applications.

Rich set of peripherals: The ESP32 has several built-in peripherals, including ADC (analog-to-digital converter), DAC (digital-to-analog converter), and PWM (pulse width modulation). It also has number of general-purpose I/O (GPIO) pins that can be used to interface with external devices.

Low power consumption: The ESP32 is designed to be power efficient, with several power-saving modes that can be used to extend the battery life of portable devices.[4]

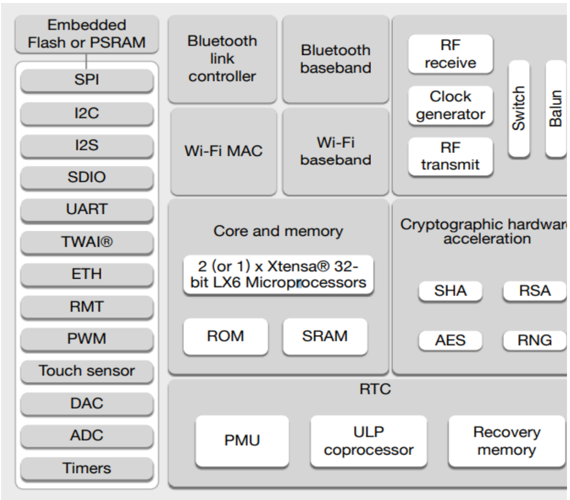

Fig 1. Block Configuration[2]

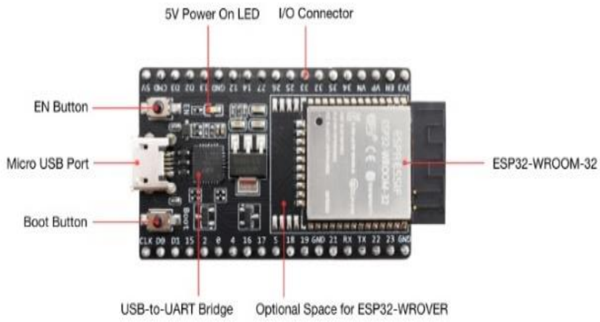

Fig 2. ESP32 Module Description [4]

ESP32 is a versatile and feature-rich microcontroller that is lot of application, including IoT, robotics, and home automation. It is a popular choice for building wireless projects due to its low cost, low power consumption, and wide range of features.

We are using ESP32 microcontroller as main control unit in a line following robot to handle the overall operation and control of the robot. ESP32 would be responsible for:

Reading sensor data: The ESP32 can be connected to sensors that are used to detect the position of the robot relative to the line. We use infrared sensors to detect the presence or absence of the line.

Processing sensor data: The ESP32 can use the sensor data to determine the position of the robot relative to the line and calculate the necessary control signals to keep the robot aligned with the line.

Sending control signals to the motors: The ESP32 can use its communication interfaces (e.g., UART, I2C, etc.) to send control signals to the motors that drive the robot. Here we use PWM signals to control the speed and direction of the motors.

Communicating with external devices: Depending on your specific application, the ESP32 might need to communicate with external devices, such as a computer or smartphone, to receive commands or send data. The ESP32 has built-in Wi-Fi and Bluetooth capabilities, which can be used for this purpose.[5]

Overall, the ESP32 can be used as the main control unit in a line following robot to handle the sensor data, calculation of control signals, and communication with external devices. It is a versatile and feature-rich microcontroller that can be programmed to perform a wide range of tasks in this application.

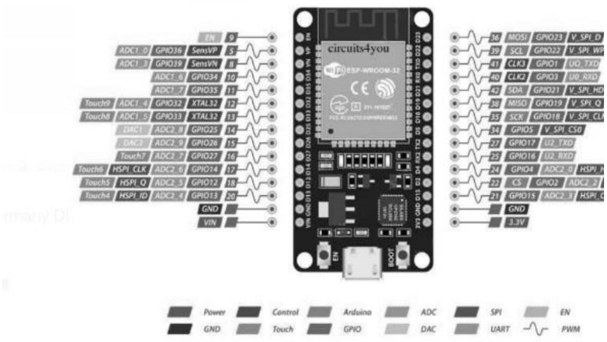

Fig 3. ESP32 Pin Description [5]

- 16 PWM output channels
- 18 Analog-to-Digital Converter (ADC) channels
- 10 Capacitive Sensing GPIOs
- 3 UART interfaces, 3 SPI interfaces
- 2 I2C interfaces, 2 Digital-to-Analog Converters (DAC)
- 2 I2S interfaces.
- Enable (EN) is the 3.3V regulator's enable pin.
- It is pulled up, so connect to ground to disable the 3.3V regulator. This means that you can use this pin connected to a pushbutton to restart your ESP32.

3.2 IR SENSOR

Sensors that detect and measure the intensity of infrared radiation are called as infrared (IR) devices. They are widely utilized in various fields, such like robotics, automation, and environmental monitoring. A beam of IR radiation is generated by an IR sensor, and the amount of reflected radiation then is detected. The sensor could determine the distance to the target since such amount of reflected radiation is directly proportional to the distance between it. By measuring the amount of reflected radiation and comparing it to a threshold value, IR sensors can also be used to identify if an object is present or not. [6]

There are several types of IR sensors, including photodiodes, phototransistors, and thermopiles. Photodiodes and phototransistors are sensitive to IR radiation in the visible and near-infrared range, while thermopiles are sensitive to IR radiation in the mid-infrared range.IR sensors are used in a variety of applications, including line following robots, where they are used to detect the presence of a line or track. They are also used in security systems, to detect the presence of intruders, and in temperature sensing applications, to measure the temperature of an object or environment.IR sensors are typically small and inexpensive, making them an attractive choice for many applications. However, they are sensitive to ambient light and may not perform well in brightly lit environments. They may also have difficulty detecting transparent or highly reflective objects.

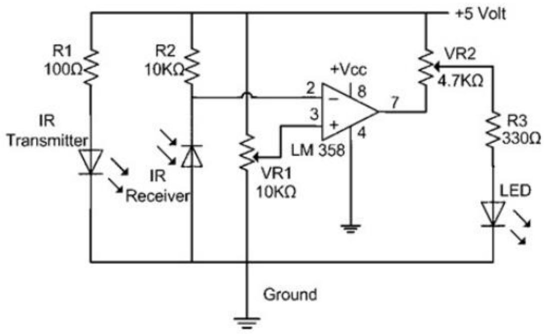

Fig 4. IR Sensor circuit [6]

An IR sensor with a distance range of 3 to 30 cm and a voltage range of 3.3 to 5 volts can be used in our line following robot project to detect the presence of a line. The sensor can be mounted on the front of the robot, facing downward towards the ground. When the sensor is placed over a line, it will detect the reflected IR radiation and send a signal to the microcontroller, indicating the presence of the line. we will connect it to an esp32 microcontroller, and program the microcontroller to read the sensor's output. The microcontroller can then use the sensor's readings to adjust the robot's speed and direction, allowing it to follow the line.[6]

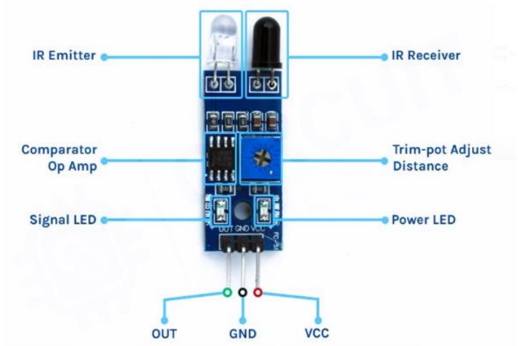

Fig 5. IR Sensor Module [7]

It is important to note that the accuracy of the sensor's readings may be affected by the reflectivity of the surface on which the robot is traveling. A surface that reflects IR radiation poorly may result in lower sensor readings, while a highly reflective surface may result in higher readings. You may need to adjust the sensor's threshold values or use additional sensors to compensate for these variations in reflectivity.

3.3 L298N MOTOR DRIVER

A DC motor can only be entirely underneath our control if we have full control over its speed and direction of rotating. Combining these two techniques allows you to achieve this goal.

PWM serves to modulate speed.

H-Bridge: used it to adjust the direction of spinning.

The L298N is a popular stepper motor driver that is widely used to control the movement of stepper motors in various applications, such as robotics, CNC machines, and 3D printers. The L298N is a dual H-bridge which is control the motor driver, which means it can drive two DC motors or one stepper motor. It uses high-power transistor switches to control the flow of current through the motors and can operate at a maximum current of 2A per channel.

The L298N has two input channels, labelled IN1 and IN2, which can be used to control the forward and backward the direction of the line following robot motor. By setting these inputs to different logic levels (e.g., high or low), the motor can be made to rotate in one direction or the other. [9] The L298N also has two PWM (pulse width modulation) inputs, labelled EN1 and EN2, which can be used to control the speed of the motor by varying the duty cycle of the PWM signal. The L298N can be used with a microcontroller or other control device to control the movement of a stepper motor easily and accurately. It is a versatile and widely used component in many different types of applications.

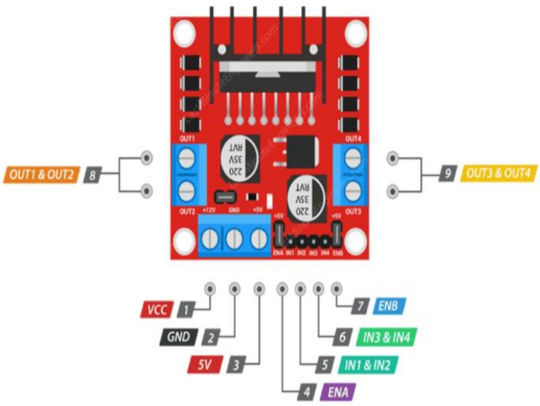

Fig 6. L298N Module [9]

Motor output voltage	5V – 35V
Motor output voltage (Recommended)	7V – 12V
Logic input voltage	5V – 7V
Continuous current per channel	2A
Max Power Dissipation	25W

Fig 7. L298N Voltage Configuration [9]

In our case we will Connect the L298N to a power source and the stepper motor. The L298N will typically require a DC power supply of around 12V to 24V, depending on the specific needs of your motor. Connect the input and PWM signals from the microcontroller or control device to the L298N. These signals will be used to control the direction and speed of the motor.[9]

In this application, the L298N is used to control the movement of the stepper motor in response to the position of the robot relative to the line. By using the input and PWM signals to control the direction and speed of the motor, you can use the L298N to accurately steer the robot along the line.

3.3 DC MOTOR

These sturdy plastic gearbox motors, commonly referred to as "TT" motors, are an inexpensive and simple method to get your creations rolling.

DC Gearbox Motor, "TT Motor," 200RPM, 3 to 6VDC From 3VDC to 6VDC can be utilized to power these motors. They will definitely move a little more swiftly at higher voltages. We found this information by running one motor from a benchtop supply. 3 VDC, 120 RPM no load, and 1.1 Amps even before device switched off. 1.2 Amps when the motor stalled at 185 RPM at 4.5 VDC with really no load. The device halted at 250 RPM no-load at 6 VDC and 1.5 Amps.

Fig 8. DC Motor [10]

Keep in mind that they are relatively simple motors that lack speed control, built-in encoders, or positional feedback.

Rotation stops when the voltage is applied! If you need precision movement, a different feedback method is necessary because there will be difference from motor to motor.

3.4 ROBOT CAR KIT

The robot car kit we are using consists of two geared motors, which are used to propel the robot and allow it to change direction. These motors are supplied with a voltage between 3 and 9 volts, which determines their speed and power. The kit also includes an on and off switch, which allows you to control the robot's power supply, and a battery holder, which holds the batteries that power the motors. Overall, the car kit provides the necessary components for building a basic line following robot, including the motors, power supply, and control mechanisms.

However, additional components such as sensors, microcontrollers, and programming may be required to fully customize and control the robot's behavior. Chassis Dimensions are 200 x 100 mm and Tires Dimensions are 60 x 27 mm. It is made of sturdy acrylic with many holes for mounting components.

Fig 9. Robot kit [12]

4. SYSTEM DESIGN AND ANALYSIS

KiCad is software for the simulation and layout. Which is easy to understand.

KiCad can be regarded as developed enough to be used for the effective creation and upkeep of intricate electronic boards. With KiCad, there are no limits on board size, and the software is readily capable of handling up to 32 copper layers, 14 technical layers, and 4 auxiliary layers. All the files required to construct printed circuit boards, as well as drilling files, component positioning files, Gerber files for photo-plotters, and many more, can be generated using KiCad.

KiCad is the best tool for projects aimed at producing electronic gear with an open-source flavor because it is open source (GPL licensed).

Fig 10 KiCad [11]

simplicity in use and creation. It required to have clear, straightforward, and organic concepts for developers. In order to make development easier, the code produced also required to be synchronous.

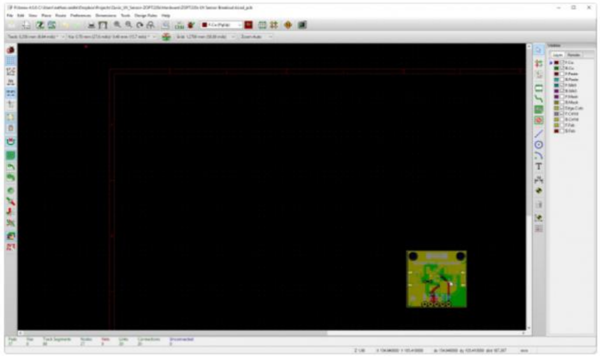

Fig 11 Ki cad background

Reusability and extensibility We were able to create a straightforward yet effective pipeable interface by keeping reusability in mind. Schematics can be added to other Schematics as input or as an output. Schematics for components and modules, for instance, can be used to design an application.

Atomicity. The CLI blueprints contained numerous flaws that were a direct result of side effects caused by our designs. We made the choice to completely eliminate side effects from our code when we were developing Schematics.

All the modifications are kept in memory and are only put into effect when they have been validated. For instance, creating a file that already exists is incorrect and would cause any previously applied changes to be discarded.

Schematics required to enable use cases that were asynchronous in nature because many workflows are (for example, contacting web servers). This seemed to be at odds with the initial objective of keeping the debugging process synchronous, but we were able to create a design that made everything function as a whole. A Schematics has synchronous input but asynchronous output, and the library will wait for everything to finish before moving on to the next stage. Developers can reuse in this manner without even being aware that a Schematic is asynchronous.

4.1 CIRCUIT VIEW

The actual electrical connections are shown in a circuit diagram. Physical design or wiring diagram refers to a diagram that shows the components are connected of the wires and the components they connect.

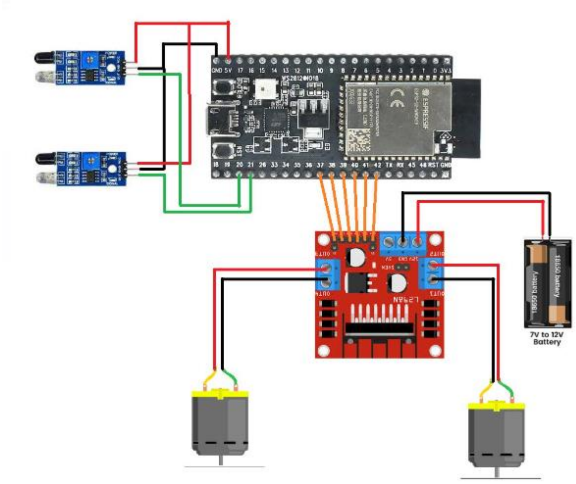

Fig 12 Circuit connection Configuration

This diagram develop to image and connection in the model and enhancement of the all creating IC. It will create 3D image and all connection in the motor, l298N module, IR sensor and cells. Dc motor control by the motor driver l298N and it transfer 12 voltage to the motor output which is also transfer 5 voltage in the microcontroller for the basic requirements of the ESP32.

4.2 SCHEMATIC DIAGRAM

A schematic diagram is a visual representation of a process, device, or other object's parts using standardized, sometimes abstract symbols and lines. Schematic diagrams only depict the crucial components of a system, yet some details may be exaggerated or introduced to the diagram to make the observer understand the system.

The schematic for such a "line-taking robot" The Esp32 microcontroller is the key part. A schematic is developed using KiCAD. The following list summarizes the primary features of the equipment: Microcontroller Esp32.

The intelligent sensor is an IR-LED with IR lighting.

The quad comparator IC LM324 and a tuning potentiometer for reference voltage.

The IC (L298N) motors there under control of the H-bridge engine have connected reduction gears. connectors to join the various sheets into a single functional object.

As a sensing element, a combination of IR-LEDs and photodiodes is used in the design of the tachometer. Each piece of equipment was created/implemented separately for its purpose before being combined into a single whole application.

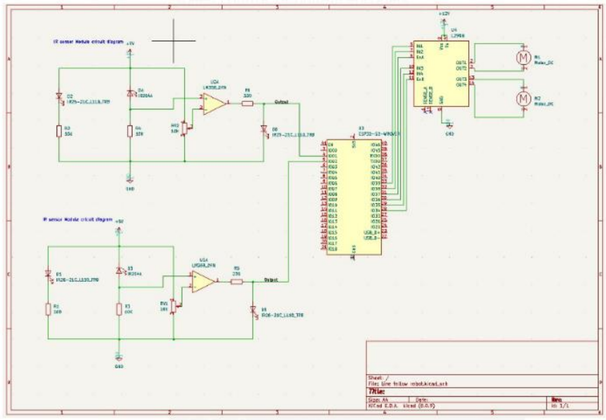

Fig 13. Circuit connection Configuration:

4.3 LAYOUT DIAGRAM

A PCB (printed circuit board) layout is a plan for arranging electrical components and their connections on a physical board. It is an important step in the design and manufacturing of electronic devices because it ensures that the components are properly connected and work as intended. Detailed drawings of the components, their locations, and the connections between them are typically included in the layout.

In the case of our project, the PCB layout must precisely position the IR sensor, ESP32 microcontroller, and motor driver to ensure proper communication and functionality. We must ensure that the sensor is placed in an area where it can detect the black line accurately, that the microcontroller is placed in an area where it can easily receive the sensor's input and send the appropriate output to the motor driver, and that the motor driver is placed in an area where it can easily receive the microcontroller's output and control the motors.

Our PCB layout for a line following robot that detects a black line on the ground using an infrared (IR) sensor. The input of the sensor is linked to a processing unit, which is an ESP32 microcontroller. The input from the IR sensor is processed by the microcontroller and used to determine the position of the line. It then sends an output to a motor driver, which controls the robot's motors. The motors propel the robot along the path of the line.

This PCB layout is intended to control the movement of a line-following robot by utilizing an IR sensor, an ESP32 microcontroller, and a motor driver. The sensor detects the position of the line, the microcontroller processes the information, and the motor driver controls the robot's movement based on the microcontroller's output.

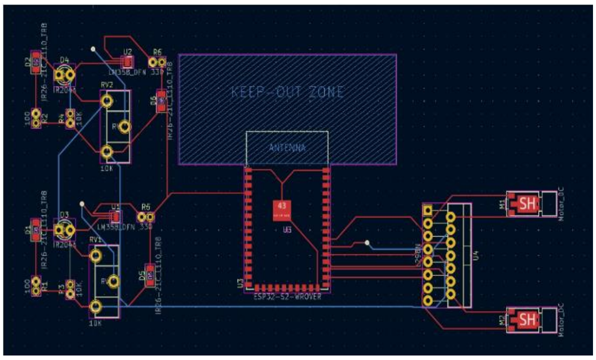

Fig14. Layout Configuration

5. DEVELOPMENT

5.1 SOFTWARE REQUIREMENT

- Operating System: Windows 10
- IDE: Arduino

Arduino methods and functions are introduced to the C++ that is used to create the Arduino code. Human-readable programming languages includes C++. A text editor for writing code, a message area, a text console, a toolbar with buttons for basic functions, and several menus are all part of the Arduino Integrated Development Environment (IDE), also known as the Arduino software. In order to upload programs and communicate with them, it connects to the Arduino hardware. Sketches are software programs developed using the Arduino Software (IDE). A sketch is generated and then processed and converted to machine code. The text editor is utilized to create these sketches, that are then saved with the file extension. The editor offers functions for text replacement and text searching. While saving and exporting, the message section provides feedback and displays.

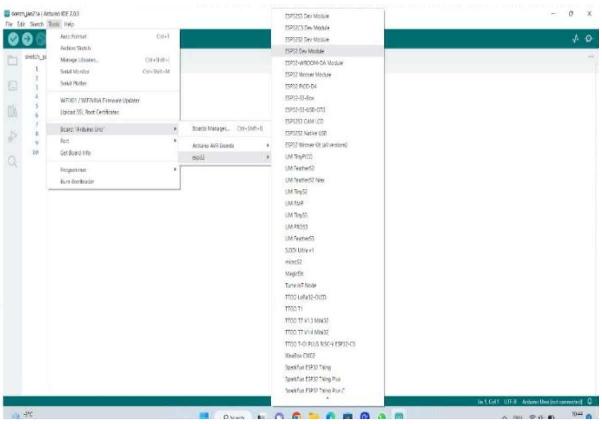

Fig15. Arduino Configuration

1. Open-source software called Arduino IDE is mostly used for authoring and compiling code into Arduino Modules.

2. Due to the native Arduino software's ease of use, even a layperson with no prior tech knowledge can learn how to compile code.

3. It is supported by the Java Programming language, which can be easily accessed by MAC, Windows, and Linux operating systems and has built-in functions and commands that are necessary for debugging, editing, and compiling the code in the environment.4. A wide selection of Arduino modules are available, including the Uno, Mega, Leonardo, Micro, and many others.

5. On the board of each of them is a microcontroller that is actually programmed and takes data in the form of code.

6. After being developed on the IDE platform, the main program, often referred to as a sketch, would eventually produce a Hex File that is transmitted and uploaded into the controller on the board.

7. The IDE environment primarily consists of two fundamental components: Editor and Compiler, the former of which is used to write the necessary code and the latter of which is used to compile and upload the code into the specified Arduino Module.

8. The languages C and C++ are supported in this environment.

5.2 SOFTWARE AND HARDWARE REQUIRMENT

5.2.1 FLOW OF PROJECT

The figure shows the flow of the robot and design logic according to the input and output of the logic. There are four steps for controlling robot.

➢ The robot is equipped with sensors that are used to determine its position on the track. The robot will continuously check the input of these sensors to determine whether it is currently standing on the track or not. If both sensors sense the track, the robot will change its direction and begin tracking the line in the opposite direction. If this condition is met, the robot will move forward and backward along the track. However, if the sensors do not detect the track, the robot will return to the start command and repeat the loop until the track is detected.

➢ The robot is programmed to utilize its left sensor to detect the line track. If the left sensor detects the line track, the robot will respond by rotating the left wheel in the opposite direction and the right motor in the forward direction. This will cause the robot to change direction and follow the line track. However, if the left sensor does not detect the line track, the robot will come to a stop. This is a safety feature to prevent the robot from wandering off the line track and potentially causing damage or injury.

- The robot is also programmed to move in the opposite direction, to the right, using similar sensor input. When the right sensor detects the line track, the robot will respond by rotating the right wheel in the reverse direction and the left motor in the forward direction. This action will cause the robot to change its direction and move to the right. The robot will continue to move in this direction until the sensors no longer detect the line track, at which point the robot will stop and return to the start command to repeat the loop until the track is detected again. The goal of this is to ensure that the robot is able to navigate the track in both directions, providing more flexibility and adaptability to different scenarios.
- When both sensors do not detect the black line, it indicates that the robot is not on the track. In this case, the robot will continue to move forward without interruption or any other device intervention. The robot will maintain its current direction and speed without making any changes. This is because the robot is not detecting any obstacle or deviation from the track, so it will continue to move forward as programmed until it detects the track again or reaches its destination. This feature allows the robot to cover a large area without stopping, making it more efficient and time-saving."
- It will check the input of the sensor which gives the robot is stand on the track or not so the both sensor is the sense the track then it transfers the reverse of tracking the direction. If yes then move forward and backward otherwise go on below loop of the start command.
- If the left sensor senses the track of line, then it will left wheel rotate reverse direction and right motor rotate forward direction and this condition is not identify then robot will stop.
- The same things happen to move robot in right way, and this is right motor in reverse direction and left motor rotate forward direction.
- If both input is not in black line, then motor goes in forward direction without interrupt and device.

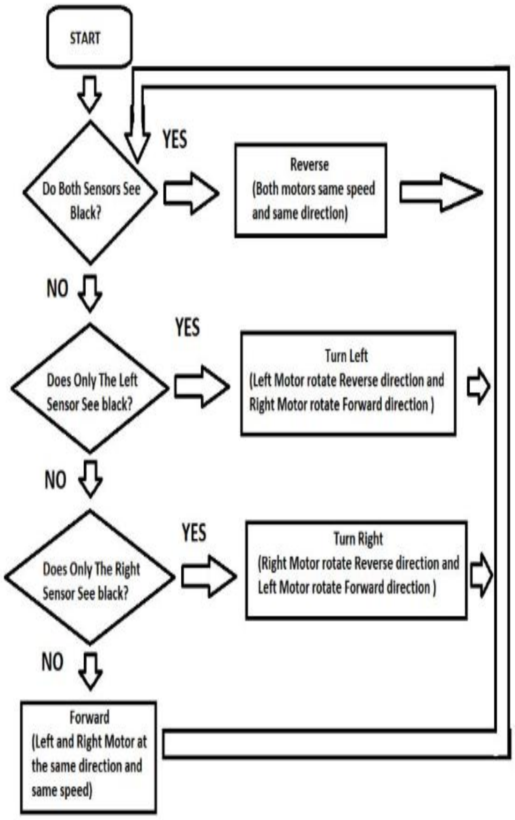

Fig 16. Flow chart

5.2.2 IR MODULE TESTING

We have to initialize global variable for the pin of esp32 GPIO pin. int IR_module_GPIO_pin = 2.

Following that is our setup function. We initialize the serial with a 9600 baud rate in the setup function. Then we initialize the IR Sensor pin as input and the LED pin as output, and finally we print a sentence to see if the serial monitor window is functioning properly or not.

void setup ()
{
Serial.begin(9600); // Init Serial at 9600 Baud Rate.
pinMode (IR_module_GPIO_pin, INPUT); // IR Sensor pin INPUT
}
void loop ()
{
Int sensorStatus_Module = digitalRead(IR_module_GPIO_pin); // Set the GPIO as Input
if (sensorStatus_Module == 1) // Check if the pin high or not
{
Serial.println("Motion Detected in the module!"); // print Motion Detected! on the serial.

```
}
else
{
//else turn on the onboard LED
Serial.println("Motion Ended in the module!"); // print Motion Ended! on the serial
}
}
```

We then go on to our infinite loop. In the endless loop, we first use the digitalRead () function to read the sensor pin and then save the result in the sensor Status Module variable. The sensor's output is then checked to see if it is high or low; if it is high, no motion is detected; if it is low, motion is detected; we also print this status in the serial monitor window.

IR module is transmitted data in serial monitor and check both IR module using Esp32 microcontroller.

5.2.3 L298N MODULE LOGIC

We have to initialized motor enable pin for the motor selection and motor logic high and low for the output in L298N.

```
#Define motor_enable 5
#Define motor_in1 7
#Define motor_in2 8
```

The setup portion of the code configures every motor control pin, including the direction and speed control pins, as a digital OUTPUT. The direction control pins are also pulled low to initially disable both motors.

```
void setup ()
{
    // Set all the motor control pins to outputs
    pinMode (motor_enable, OUTPUT).
    pinMode (motor_in1, OUTPUT).
    pinMode (motor_in2, OUTPUT);


    // Turn off motors - Initial state which off the motor
    digitalWrite(motor_in1, LOW);
    digitalWrite (motor_in2, LOW).
}
```

If Motor both input is high, then motor goes in forward direction, so we have to test it according the input and apply 12 volts to the motor driver. It works depend on the speed of the motor.

5.3 why we need a Motor Driver Circuit

A Motor Driver Circuit is a device that is used to control the speed and direction of a DC motor. The primary function of a motor driver circuit is to provide the necessary current and voltage to operate the motor, as well as to control the direction of the motor's rotation.

The reason for using a motor driver circuit is that most DC motors require more than 250mA of current to operate, which is beyond the current handling capacity of most microcontroller devices such as Atmega16/32, 8051, and other ICs. If a motor is connected directly to a microcontroller, the IC will burn out due to the enormous current consumption. Additionally, motors require a greater amount of voltage to operate at high speed, which is not provided by microcontroller devices.

There are various types of motor driver circuits available in the market, but the most widely used ones are the L293 and L298 series, including the L293D, L293NE, and L298N. These ICs are specifically designed to handle two DC motors at once and use the H-bridge configuration for motor control. The H-bridge is a simple circuit that allows for bidirectional control of a motor and is suitable for motors with low current ratings.

In summary, a Motor Driver Circuit is necessary for controlling DC motors as it provides the necessary current and voltage to operate the motor and allows for bidirectional control. The L293 and L298 series, including the L293D, L293NE, and L298N, are popular and widely used motor driver ICs that are suitable for most projects.

In above all the driver we prefer L293D and L298N because this driver is good for our project.

The basic difference between L293D and L298N Motor Driver?

The L293D and L298N are popular choices for motor driver circuits due to their versatility and robustness. The L293D is a dual H-bridge motor driver IC that can control the speed and direction of two DC motors simultaneously. It has a maximum current rating of 600mA per channel and can supply up to 36V DC to the motors. It also has built-in protection features, such as thermal shutdown and current limiting, which prevent damage to the IC and connected motors in case of overloading or overheating.

The L298N is also a dual H-bridge motor driver IC that can control the speed and direction of two DC motors simultaneously. It has a maximum current rating of 2A per channel and can supply up to 46V DC to the motors. It also has built-in protection features, such as thermal shutdown and overcurrent protection, which ensure the safety of the IC and connected motors. Additionally, it also has an enable pin that can be used to turn off the motor driver and stop the motors, which provides more control over the motor operation.

In summary, both L293D and L298N are popular choices for motor driver circuits due to their versatility and robustness. They are both dual H-bridge motor driver ICs that can control the speed and direction of two DC motors simultaneously. They both have built-in protection features which protect the IC and connected motors from damage. L298N has more current rating and voltage supply capacity than L293D and have an enable pin. Depending on the specific requirements of the project, either of these motor driver ICs could be a suitable choice for the project.

5.4 BATTERY, LITHIUM-ION

Fig17. Battery cell [15]

Lithium-ion batteries are a type of rechargeable battery. Portable devices like electric automobiles, computers, electronics, and cell phones typically use it. They are also increasingly used in military and aerospace applications.[15]

The voltage output of a lithium-ion battery is 3.7V in storage mode and 4.2V in full charging mode. In this project, two lithium-ion batteries are linked in series, resulting in a total battery voltage of 8.4V when fully charged. Consult the article on how a lithium-ion battery works to learn more about them.

Why do we not use this battery because this battery only generates the 4-5V? In my project, we need the 12V so that the robot works well and we can increase the speed also.

In this project, we use a 4-5V cell and this cell is not rechargeable this product is for one-time use. In this project, we use 4 cells and join in the parallel cell

6. TESTING PROJECT

6.1 DIRECTION OF ROBOT DIRECTION

In this project we employed infrared transmitters and infrared receivers, we commonly known as photo diodes, in this line-following robot. where Light will be transmitted and received through them. Basically, Infrared light is transmitted through IR. When infrared light strikes on the white surface then light will be bounces back and is captured by the photodiodes, they causing minor voltage changes. IR light is absorbed by a black line surface when it strikes it.

The photo diode does not receive any light or rays due to the black surface's lack of reflection.

Here, in this line-following robot, the IC receives a 1 as input when the sensor detects a white surface and a 0 as input when the sensor detects a black line.

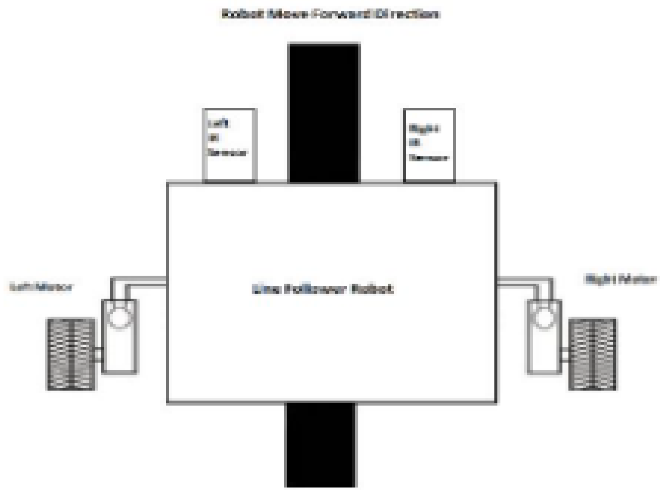

Fig18. Robot model direction

The 2Array IR Sensors part of the gadget consists of IR diodes, a potentiometer, a comparator (OpAmp), and LEDs. A potentiometer is used to set the reference voltage at the comparator's first terminal, and IR sensors are utilized at the comparator's second terminal to sense the line and cause a change in voltage.

After comparing the two voltages, the comparator produces a digital signal. This line follower circuit using two sensors used two comparators.

Regulatory Section: The IC L298N manages the line follower robot's complete function. The line follower is driven by the driver circuit, which receives instructions from this IC after reading these signals.

The driver portion consists of the motor driver and two more personnel.

How does a Line Following Robot Navigates?

Typically, line-following robots use two motor sets. Based on the signals from the left and right sensors, respectively, both motors rotate.

Basically, the robot perform 4 sets of movement

- Right side
- Left side.
- Forward
- Reverse.

The left sensor is on top of the black line, whereas the right sensor is on the white surface, the robot should move left side.

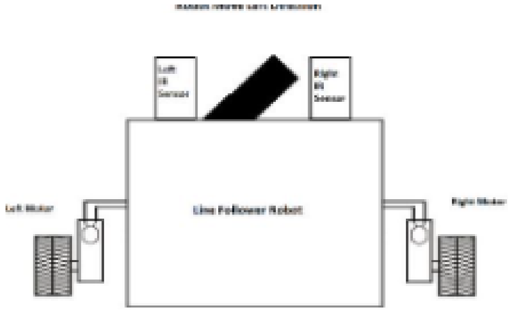

Fig19. Right direction

Robot will turn right until both sensors reach a white surface if the right sensor detects a black line. Robot resumes forward movement as soon as a white surface appears.

The right sensor is on top of the black line, whereas the left sensor is on the white surface, the robot should move right side.

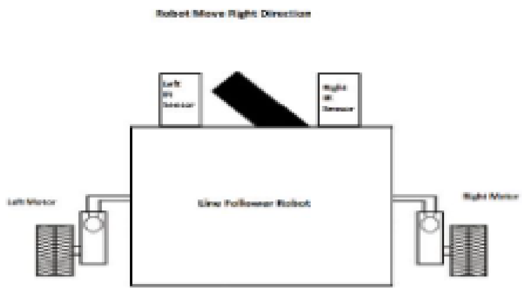

Fig20. Left direction

If the left sensor detects a black line, the robot will turn right until both sensors come to a white surface.

Robot starts to move again whenever a white surface is visible.

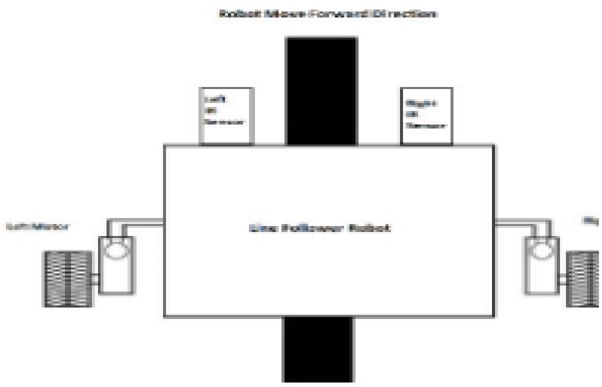

Fig21. Forward direction

Robot will move forward until any of the sensors detect a black line.

When both the sensors are detect a black line then the robot should reverse.

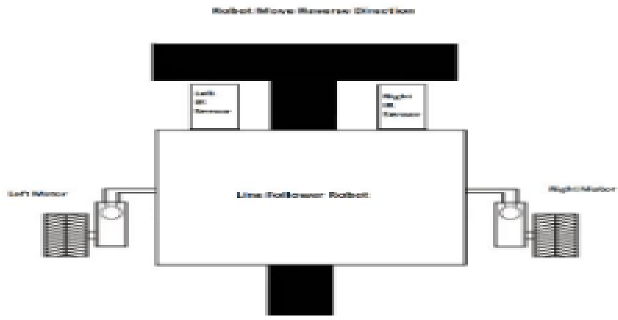

Fig 22. Stop

Any track can be used for the robot's line-following track. We have used the following. We had to make sure that the surface was transparent and the black line was opaque.

Dc motor test

When the motor is start at that time input is 12v and once robot complete the path at that time we checked the motor output is 10v. Input power will always be greater than output power due to friction and other loses.

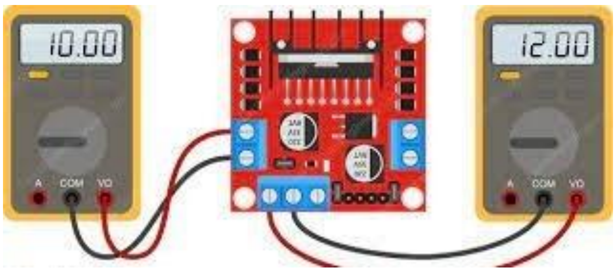

Fig23. Voltage measurement [17]

7. DEMONSTRATION

7.1 BASIC OPERATION:

The basic operations of line followers are:

An optical sensor mounted on the tip of the robot detects the line position which is the black path. For this purpose, an IR LED and photodiode combination known as a light sensor was used and it transfer the high and low signal after that develop the logic according to microcontroller. This results in a high resolution and robust acquisition process.

Steering robots require a steering mechanism for tracking because sensor position is perfect mounted in the robot using potential meter. To overcome this obstacle, two motors that regulate the motion of the wheels are used.

If it doesn't detect a black surface, the robot moves in a circle until it finds a line and battery cell used for the transfer voltage for motor driver.

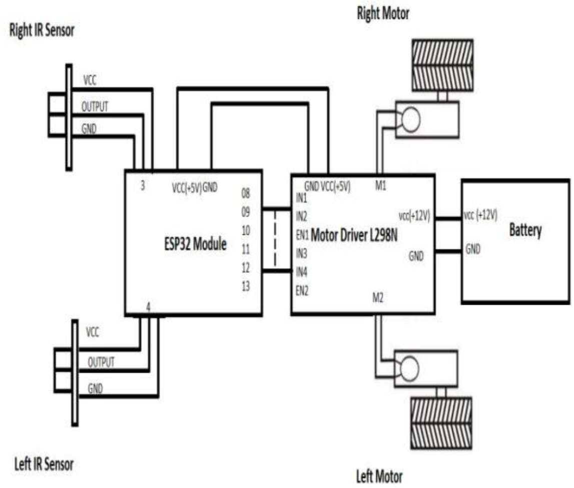

Fig 24. Block diagram of the project

The basic structure shows the input and output of the sensor and what are the controlling logic will put in the microcontroller and it will give the output to the controller as well as motor driver.

There are divided in the four parts.

a. IR sensor
b. ESP32 microcontroller
c. Motor driver
d. Battery

The robot diagram is used to represent the fictional portion of the robot that moves current and voltage in accordance with the appliances. Motor drivers are used to control DC motors, and they pass control of the motor's ability to travel forward, backward, right, or left depending on the IR sensor to the microcontroller.

IR sensors have a potential meter that they can use to modify the black line's length and position objects in line with the track.

Fig 25 IR LED or IR transmitter.[13]

An infrared transmitter, also known as an IR LED, is a type of light-emitting diode (LED) that emits infrared radiation. Although an IR LED appears to be a regular LED, the radiation it emits cannot be seen by the human eye.

The image below shows an infrared LED.

IR receiver or photodiode:

An IR transmitter's radiation is picked up by infrared receivers or infrared sensors. Photodiodes and phototransistors are two types of infrared receivers. Because they only pick up infrared radiation, infrared photodiodes are distinct from ordinary photodiodes. A photodiode or an IR receiver is depicted in the image below.

Fig 26 IR Receiver [19]

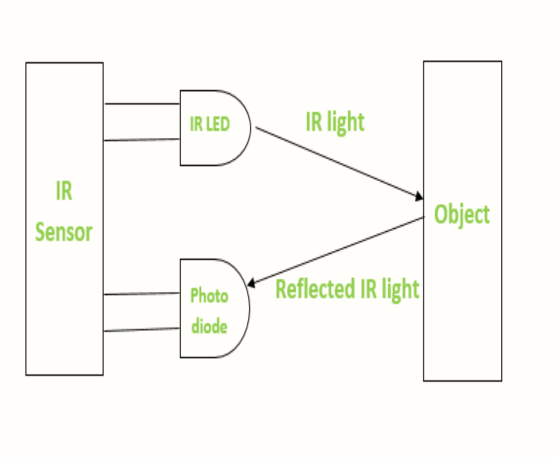

Fig 27 Object detection

Depending on the wavelength, voltage, packaging, or other parameters, there really are multiple kinds of IR receivers. When used in an infrared transmitter-receiver pair, the wavelength of the receiver should coincide with the wavelength of the transmitter.

The emitter is an IR LED, while the detector is an IR photodiode. The amount of IR light an IR photodiode gets from an IR LED influences both its output voltage and resistance. This is the basic operation of the IR sensor.

7.2 ESP32 WORKING LOGIC

Input		output				Movement of robot
Left Sensor	Right Sensor	Left Motor		Right Motor		
LS	RS	LM1	LM2	RM1	RM2	
1	0	0	1	1	0	Left
0	1	1	0	0	1	Right
0	0	1	0	1	0	Forward
1	1	0	1	0	1	Reverse

Fig 28. Command of the controller

There are four main condition to rotate the motor according to the sensor input. And it will create same logic in the program.

1. Left direction rotate.
2. Right direction rotates.
3. Forward direction
4. Reverse direction

```
embadded.ino
void loop() {
  bool dummy_left = digitalRead(left_Sensor);
  //bool centerV = digitalRead(center);
  bool dummy_right = digitalRead(right_Sensor);

  Serial.println(rightV);

  if (dummy_left == 0  && dummy_right == 0) {
    carforward();
    Serial.println("forward");
  } else if (dummy_left == 1  &&  dummy_right == 1) {
    carStop();
  } else if (dummy_left == 0  &&  dummy_right == 1) {
    carturnleft();
  } else if (dummy_left == 1  &&  dummy_right == 0) {
    carturnright();
```

Fig 29. IR sensor coding for controlling motor

First condition show that if left sensor give output 1 and right motor will be 0 so that motor turn in right motor is working and it is ON condition so that robot rotate in the left direction. After that the black track is shown that motor goes in the right direction for the tracking depends on the sensor input left motor is working.

Forward direction has both motor is working condition and reverse direction both motor goes in reverse direction.

Working

The ESP32 is a microcontroller developed by Espressif Systems that is widely used in line following robots due to its high accuracy and advanced features. One of the key features of the ESP32 is its dual-core processor, which allows it to handle multiple tasks simultaneously and perform complex calculations with high accuracy.

The ESP32 also has built-in Bluetooth and Wi-Fi capabilities, which allows it to communicate wirelessly with other devices and networks. This can be useful in line following robots, as it allows the robot to receive commands and transmit data to a remote control or monitoring system.

The ESP32 also has a wide range of peripheral interfaces, such as I2C, SPI, UART, and PWM, which allows it to interface with a variety of sensors and actuators. This is useful in line following robots, as it allows the robot to use various sensors, such as infrared or ultrasonic sensors, to detect the line and make precise adjustments to its movement.

Another advantage of the ESP32 microcontroller is its low power consumption which allows the robot to run longer on a battery and reduces the need for frequent recharging. The ESP32 also has a built-in low-power sleep mode, which allows it to conserve energy when it is not performing any critical functions.

In conclusion, the ESP32 microcontroller has many advanced features that make it well-suited for line following robots, including high accuracy, built-in wireless connectivity, a wide range of peripheral interfaces, and low power consumption. These features make it easy to build a highly accurate and reliable line following robot that can perform complex tasks and operate in various environments.

The ESP32 is a microcontroller developed by Espressif Systems that is widely used in line following robots due to its high accuracy and advanced features. One of the key features of the ESP32 is its dual-core processor, which allows it to handle multiple tasks simultaneously and perform complex calculations with high accuracy.

The ESP32 also has built-in Bluetooth and Wi-Fi capabilities, which allows it to communicate wirelessly with other devices and networks. This can be useful in line following robots, as it allows the robot to receive commands and transmit data to a remote control or monitoring system.

The ESP32 also has a wide range of peripheral interfaces, such as I2C, SPI, UART, and PWM, which allows it to interface with a variety of sensors and actuators. This is useful in line following robots, as it allows the robot to use various sensors, such as infrared or ultrasonic sensors, to detect the line and make precise adjustments to its movement.

Another advantage of the ESP32 microcontroller is its low power consumption which allows the robot to run longer on a battery and reduces the need for frequent recharging. The ESP32 also has a built-in low-power sleep mode, which allows it to conserve energy when it is not performing any critical functions.

In conclusion, the ESP32 microcontroller has many advanced features that make it well-suited for line following robots, including high accuracy, built-in wireless connectivity, a wide range of peripheral interfaces, and low power consumption. These features make it easy to build a highly accurate and reliable line following robot that can perform complex tasks and operate in various environments.

The heart of the project and mind of the project is microcontroller there is no analog pin in the controller, so we are adjusting the speed in the PWM and change the duty cycle according the input and controlling the motor driver. in detail

The microcontroller is the heart and mind of the line following robot project as it controls all the functions of the robot. In the case where the microcontroller lacks an analog pin, the speed of the motors can be adjusted by using Pulse Width Modulation (PWM).

PWM is a technique used to control the speed of a motor by adjusting the duty cycle of a square wave signal that is applied to the motor driver. The duty cycle is the ratio of the time the signal is high to the total time of the signal. By adjusting the duty cycle, we can control the average voltage applied to the motor, and thus control its speed.

In a line following robot, the PWM signal is used to control the speed of the motors based on the input from the sensors that detect the line. The sensors provide input to the microcontroller, which then adjusts the duty cycle of the PWM signal to control the speed of the motors.

For example, when the robot is following the line, the sensors detect the line and provide input to the microcontroller, which then adjusts the duty cycle of the PWM signal to increase or decrease the speed of the motors, thus controlling the robot's movement and keeping it on the correct path.

It is also important to note that the microcontroller is also responsible for controlling the direction of the motors through the motor driver. By controlling the direction of the motors, the robot can turn left or right, depending on the sensor input, and stay on the line.

In summary, the microcontroller is the heart and mind of the line following robot project, it controls all the functions of the robot, including the speed and direction of the motors, using PWM signals. By adjusting the duty cycle of the PWM signal, the microcontroller can control the speed of the motors and keep the robot on the correct path, even without the use of analog pins.

There is more accurate speed control if we write analog read in the code according change duty cycle and this controller set 16 bit so we have 1024 bit for the controlling analog pin and adjust and divided the byte in the 0 to 1024 bit sample for the data.

So that speed is controlled according the result. There is one condition that motor have to go on forward direction and move forward if both sensors not sense the black line and both IR sensor have logic 0 and so that controller give the command to the controller enable bit both will be high and PWM output is 120 so that the speed of the project is low and it will easy to go forward as per the fig 27.

7.3 SPEED CONTROL

By providing PWM from an Arduino to the L293D's enable pin, which modifies the voltages across the motor, the speed of the de motor may be adjusted. Speed is correspondingly decreased. The direction and speed are also controlled by the command.

PWM, often known as pulse width modulation, is a technology that produces analog signals using digital means. Create a square wave, which is an on/off signal, using digital control. This on/off pattern controls the voltage between the board's full Vcc (for example, 5V for UNO, 3.3V for MKR boards), and off (0 volts), during the duration of the signal's on and off states. Change can be used to simulate it. The pulse width is the length of the "on-time." This pulse width can be modulated or changed to produce various analog values. The signal will be a continuous voltage between 0 and Vcc, controlling the brightness of the LED, for instance, if you repeat this on/off pattern on an LED quickly enough.

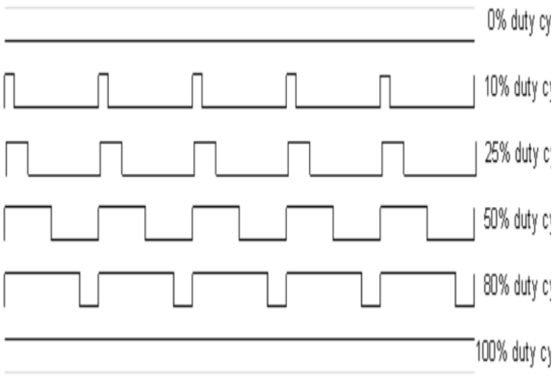

Fig 30. PWN Cycle change

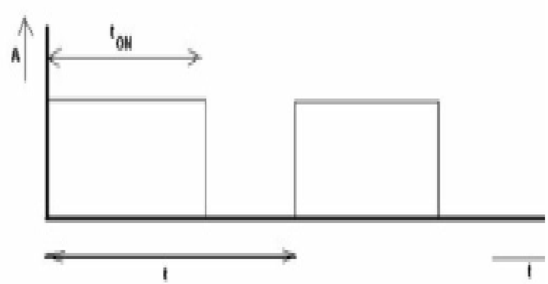

Fig 31. ON and OFF cycle

7.4 PICTURE OF LINE FOLLOWING ROBOT

This picture shows some angle which is capture by the camera and shows the model. It will design as a battery cell.

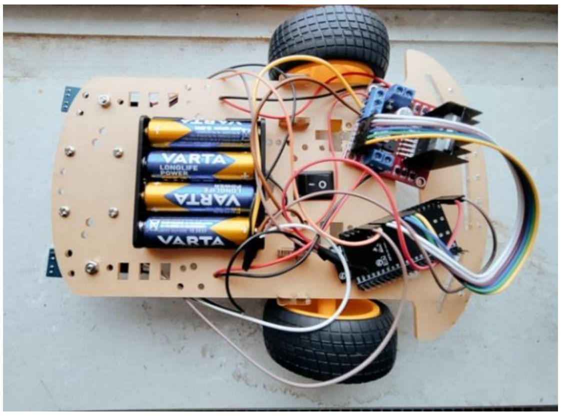

Fig 32 Top View of the robot

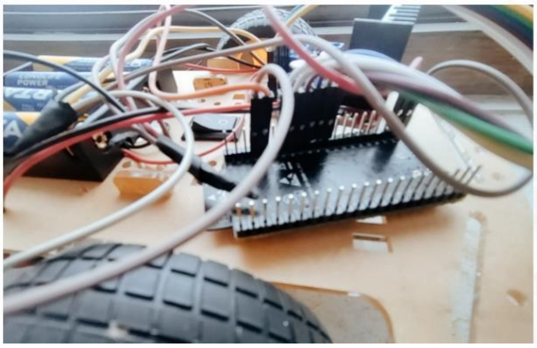

Fig 33 Component wiring part

This image shows the ESP32 wiring part and mounting in the devices so that it cannot change in the motor components. There are predefine wire used such as male to male and female to male etc. switch is used for the on and off supply voltage in the cell so that it cannot discharge the battery.

7.5 Design of path

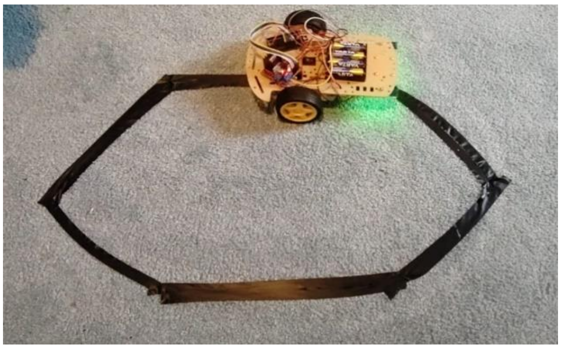

Fig 34 hexagonal path of the project

Fig 35 Curve path of the project

This image shows the path of the robot which give accurate result of the project. In the first phase of the project have only one line track but after improving the result of the project it will give round and zig zag path which will show best result of the project.

First image shoe the unusual path called as hexagonal path and robot have to sense the object according to the sensing the path and second image shows the S path which is also give the right output of the robot.

8. Conclusion & Future Scope

The line-following robot is a vehicle system that which can follow the track and it can move and alter the robot's position with respect to the line in the road.

This report represents the photodiode sensor it means IR sensor-based line following robot which is track the black line on the white surface so that IR signal will receive the data and if the path is out of track so what is condition will execute the in the background of the robot and execute in the microcontroller.

It is also understanding electronics, mechanical and software part of the system. Each task's accomplishment showed the potential of mechatronic systems and a successful group behavior.

Future enhancement:

- This project can be enhanced by the following features
- In future we can upgrade this system to change the sensor or used more number of inputs for the accurate result.
- We can also add some more technology.
- Add battery life for the more time working model.
- Use of color sensor.
- Use of camera for more accurate and improved route tracking.
- It will be useful to do this using RGB color schemes and to find a path by sensing an undetectable magnetic field on the surface.
- sends a message of alert when a physical object is approaching.
- This line-following robot can also be controlled via Bluetooth and Wi-Fi modules.
- We can alter the path of the line-following robot at intersections by using voice command controlling

REFERENCES

[1] Arduino software information and download device for the general information about coding in cand c++: https://www.arduino.cc/en/software

[2] http://esp32.net/

[3] Md Younus , [2] Pooja Gadekar, [3] Adhiraj Walse Government College of Engineering Jalgaon, Maharashtra

[4] https://www.conrad.de/de/p/espressif-entwicklungsboard-esp32-devkitc-ve-2383855.html

[5] https://www.upesy.com/blogs/tutorials/best-tutorials-for-esp32-with-arduino-code

[6] https://www.circuits-diy.com/ir-sensor-module-circuit/

[7] https://circuitdigest.com/microcontroller-projects/interfacing-ir-sensor-module-with-arduino

[8] International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 13, 2019 (Special Issue) © Research India Publications. http://www.ripublication.com

[9] information regarding the details of the project. https://www.researchgate.net/publication/334123838_Design_and_Construction_of_Line_Following_Robot_using_Arduino

[10] https://de.farnell.com/pro-elec/pel00885/inline-motorgearbox/dp/3383480

[11] https://www.eurocircuits.de/blog/kicad-eurocircuits-akzeptiert-kicad-daten-als-natives-format/

[12] https://www.reichelt.com/li/de/roboter-fahrgestell-kit-fuer-raspberry-pi-arduino-robot-car-kit-05-p258657.html?CCOUNTRY=457&LANGUAGE=de&&r=1

[13] https://www.conrad.de/de/p/osram-fotodiode-5-mm-1100-nm-20-sfh-203-154002.html?hk=SEM&WT.mc_id=google_pla&gclid=CjwKCAiAoL6eBhA3EiwAXDom5vgr_HiMPwVsACxOfiMFGgvE-RS9TUrZ9I5z4Y1HWh2JBRcDDi234RoCN3gQA

vD_BwE

[14]https://www.ijeart.com/download_data/IJEART03403.pdf

[15] https://www.sparkfun.com/products/12895

[16]https://www.fierceelectronics.com/sensors/what-ir-sensor

[17] https://electrical-engineering-portal.com/download-center/books-and-guides/electrical-engineering/motors-installation

[18] https://www.slideshare.net/UllaAhmed1/line-follower-robot-86810174

[19] https://ifuturetech.org/product/ir-receiver/

[20] https://www.reichelt.de/roboter-fahrgestell-kit-fuer-raspberry-pi-arduino-robot-car-kit-05-p258657.html

[21]https://www.ebay.de/itm/265640015328?chn=ps&norover=1&mkevt=1&mkrid=707-134425-418520&mkcid=2&mkscid=101&itemid=265640015328&targetid=1404115578893&device=m&mktype=pla&googleloc=9041610&poi=&campaignid=17935704717&mkgroupid=139162549385&rlsatarget=pla-1404115578893&abcId=9301059&merchantid=7364532&gclid=CjwKCAjwtp2bBhAGEiwAOZZTuILwDbp6pZ_3SpTn98XugdA6Rg-WF40koBS9A9oaen9T_8xKfByOhBoCon0QAvD_BwE

Appendix I

PROJECT CODE

```
#define left_Sensor 2
#define right_Sensor 3

//motor  left one
#define Left_Motor_ENA 5
#define Left_Motor_IN1 7
#define Left_Motor_IN2 6

//motor right two
#define Right_Motor_ENB 10
#define Right_Motor_IN3 8
#define Right_Motor_IN4 9

int Speed = 121; // speed of this robot

void setup() {
  Serial.begin(9600);
  pinMode(left_Sensor, INPUT);
  //pinMode(center, INPUT);
  pinMode(right_Sensor, INPUT);

  pinMode(Left_Motor_ENA, OUTPUT);
  pinMode(Left_Motor_IN1, OUTPUT);
  pinMode(Left_Motor_IN2, OUTPUT);
  pinMode(Right_Motor_IN3, OUTPUT);
  pinMode(Right_Motor_IN4, OUTPUT);
  pinMode(Right_Motor_ENB, OUTPUT);
}

void loop() {
  bool dummy_left = digitalRead(left_Sensor);
  //bool centerV = digitalRead(center);
  bool dummy_right = digitalRead(right_Sensor);

  Serial.println(rightV);

  if (dummy_left == 0  && dummy_right == 0) {
    carforward();
    Serial.println("forward");
  } else if (dummy_left == 1  &&  dummy_right == 1) {
    carStop();
  } else if (dummy_left == 0  &&  dummy_right == 1) {
    carturnleft();
  } else if (dummy_left == 1  &&  dummy_right == 0) {
    carturnright();

  }

void carforward() {
  analogWrite(Left_Motor_ENA, Speed);
  analogWrite(Right_Motor_ENB, Speed);
  digitalWrite(Left_Motor_IN1, HIGH);
  digitalWrite(Left_Motor_IN2, LOW);
  digitalWrite(Right_Motor_IN3, HIGH);
  digitalWrite(Right_Motor_IN4, LOW);
}
void carturnleft() {
  analogWrite(Left_Motor_ENA, Speed);
  analogWrite(Right_Motor_ENB, Speed);
  digitalWrite(Left_Motor_IN1, LOW);
  digitalWrite(Left_Motor_IN2, HIGH);
  digitalWrite(Right_Motor_IN3, HIGH);
  digitalWrite(Right_Motor_IN4, LOW);
}
void carturnright() {
  analogWrite(Left_Motor_ENA, Speed);
  analogWrite(Right_Motor_ENB, Speed);
  digitalWrite(Left_Motor_IN1, HIGH);
  digitalWrite(Left_Motor_IN2, LOW);
  digitalWrite(Right_Motor_IN3, LOW);
  digitalWrite(Right_Motor_IN4, HIGH);
}
void carStop() {
  digitalWrite(Left_Motor_IN1, LOW);
  digitalWrite(Left_Motor_IN2, LOW);
  digitalWrite(Right_Motor_IN3, LOW);
  digitalWrite(Right_Motor_IN4, LOW);
}
```

Appendix II

Robot Movements