Project: Line Following Robot (2023)

Atul Gadhiya
Technische Hochschule
Deggendorf Cham,
Germany
atul.gadhiya@stud.thdeg.de

Abstract- In this report, we proposed an autonomous line following robot that follows a designated Path by hurdling obstacles. The use of robots in this modern era is no longer restricted to sectors such as manufacturing, production, industrial etc., but also promises a future in households. Through this project, we propose the use of a line-following robot in logistical warehouses to maintain the location of the goods precisely while avoiding obstacles with predefined shapes in straight-line courses. Along with the use of low-cost ultrasonic sensor, the IR sensors were utilized to precisely follow the line and transmit the required data to the microcontroller under different circumstances. The obtained result from the experiment demonstrates the capability of the robot that follows a predetermined course across a floor surface while avoiding obstacles in straight lines.

Keywords— Adruino, ESP32S2, Microcontroller, Line follower Robot, Autonomous driving, Object detection, Object avoiding, IR sensors, Ultrasonic Sensors.

I. INTRODUCTION

Nowadays the need of a simple man to live an easier, reliable, efficient lifestyle has led engineers to search for a complex solution. One of the ways to reach this high level of efficient, less time-consuming life is to rely on an alternative source such as robots. In recent years, a growth in the number of research institutions and Firms have shown an enormous interest in intelligent service robots. The unavailability of skilled workers has forced the developed and emerging countries to focus the usage of robotic applications [1],Logistics [2] and Health sector[3], Automobiles sector[4] , are some of the examples, where the human work forces can be replaced by robots. Recently, the usage of robots in production, helped the factories to increase their production capacity and quality without any human intervention.

Line follow robots are autonomous driving robots that are particularly designed to follow a track or a path. To bring this into practice, the robot is programmed to move along a path with a defined speed. In this report, we are presenting the design and implementation of a line follower robot using ESP32S2 microcontroller, IR sensors and an L298N motor driver. In addition to that, to increase the quality and performance of the robot, ultrasonic sensors were introduced into the design to avoid obstacles along the straight path.

The ESP32 is a low power consumption and high-performance microcontroller which consist of several peripherals and wireless communication facilities which are suited for robotic applications. Arduino IDE is the software platform used for programming the microcontroller. The basic idea involved in the working of an infrared sensor can be compared with that of an object detection sensor. According to the literature source, the working of IR sensors can be understood as follows, when a ray of light incident on a coloured surface, a percentage of light gets partially reflected and partially absorbed. This basic principle of reflection and absorption of light, helps the prototype, to detect the blackline or white line. The values can be calculated by using Planck's radiation law, Wien's displacement law and Stephan's Boltzmann law, which can be seen in the source. Moreover, the reflection of the light can vary depending on the surface. For example, the light is easily reflected by a white surface, whereas a black surface absorbs the light [6].

In conclusion, project explains a line follow robot that can detect obstacles and follow a predefined path. In the upcoming section of the report, explanation on the design, hardware, and software, the implementation of the model and finally the working of the prototype is conducted.

A. Methodology

The research was commenced by gaining information from the literature sources on the working principles, requirements, and components such as sensors, microcontroller, motor etc. that are essential for the development of the project. Initially, a schematic diagram of the robot, which can follow the path was developed. It generally consisted of a circuit design, structural design, and the integration of the software to the robot. To detect and hurdle the obstacle lying on the path, an ultrasonic sensor was mounted onto the prototype.

Further steps in the development consisted of improvement of the code, which resulted in better detection of the obstacles. The robot was programmed using Arduino IDE software 2.0.3. The computational software that was used to check the corrections was KiCAD. After the successful completion of the simulations, the software was integrated to the hardware.

B. Background

Technology plays a vital role in today's world. In these circumstances choosing a desired robot is more important. An autonomous robot which can follow a specified or predefined path using the sensor is available in the market. Here we proposed a line following the robot with minimum performance and additional features. We used IR sensors for detecting the predetermined path that is often indicated by a line drawn on the ground or another flat surface. These sensors will provide the necessary instruction to the respective motors.

The obstacles in the path are detected using Ultrasonic sensors and provide information to the microcontroller. Moreover, the values that are provided by the sensors will be used for calculating the speed and direction of the robot. These robots can be mainly used in the industries, factories, transportation, and logistics.

C. Purpose of study

In modern factories, the main problem we are facing is logistics and transportation across various buildings and different blocks. Normally we use vehicles as the carriers to move things line follow robot which can travel through a predefined path or through a straight line by avoiding obstacles. Also, we used a blackline instead of rail or track to transport the goods from one place to. In this project we have developed an. Thereby we can reduce not only the construction cost but also the time for transportation.

D. Scope of the project

In this project, our main goal is to make a line follow robot which can be used in industries and warehouses for transporting the goods from one place or station to another station without any human intervention. Later we added an Ultrasonic sensor to avoid obstacles with limited size. This will help in the industries and factories to transport goods into a different location by avoiding obstacles. The growth in sensor technology, artificial intelligence and robotics can expand this idea and can be used in various industries and warehouses or even for delivering the goods.

II. Literature review

Considerable dedication and resources have been invested in creating mechanisms for enabling an infrared (IR) sensor-controlled robot to utilize a visual system to autonomously navigate along a predefined route. This development involves the implementation of complex technologies and methodologies to enable the robot to interpret visual cues and make real-time decisions to stay on track. The aim is to create a reliable and efficient system that seamlessly integrates the capabilities of both IR sensors and a visual system, allowing the robot to successfully follow a designated path with precision and accuracy.

There are mainly two methods that were used to run the robot. The first step is to establish a mathematical model of the vehicle and its surroundings, simulate its outputs, and then feed it to a robot designed specifically for the task. In the second approach, a visual servo system and a kinematic model were combined, and the robot is often built around the technique for resolving the problem. Simpler models and techniques, such as visual serving, are used to reduce processing burden because the processing resources available to these robots are limited due to their size.

Using the advanced model of the Arduino microcontroller system, called ESP32S2 serves as the core component and is responsible for overseeing the motor control process. The ESP32 microcontroller efficiently handles and regulates the motor control functions. The motor control system is based on an ESP32 microcontroller, which acts as the primary controller for managing the motor operations. Additionally, an IR sensor is integrated into the system to facilitate motor control. The model employs an L298N module to transfer the output signals required for controlling the motor's speed and direction. The motor's speed is regulated using a PWM signal. In addition to that the obstacles and objects in the robot path can be detected by using an Ultrasonic sensor.

III. Requirements

A. Esp32 Microcontroller

The ESP32S2 is the cheapest and low power consumption microcontroller, which is highly capable and accurate for robotic application. Along with that , it has a dual core processor with wifi and Bluetooth connectivity which makes it suitable for IOT based projects. The high processing power, low power consumption and ample memory space in the processor Figure 1: ESP3makes it efficient and suitable for developing our project. In addition to that, it has several features including Multiple communication interfaces, Multiprotocol support and the rich set of Peripherals.

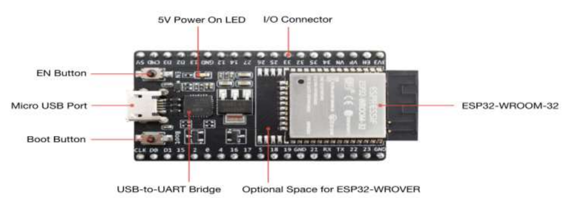

Figure 1: ESP32 [7]

The ESP32 microcontroller serves as the primary control unit in the line-following robot, taking charge of managing and overseeing its overall operation. The ESP32 is responsible for several key tasks within the robot's functionality. Initially, The microcontroller reads data from sensors which helps to find the robot position and detect the path to be followed. Two infrared sensors are employed to detect whether the line is present or not. Secondly, the ESP32 processes the sensor data to determine the robot's precise position relative to the line and calculates the necessary control signals needed to maintain alignment with it. With the provided sensor data, the microcontroller will provide the control signal for the respective motor to navigate through the path. We used a PWM signal to control and regulate the motor speed and direction. Moreover, The ESP32 can communicate with external devices such as computer and mobile applications which will give an advantage to enlarge the project. Its built-in Wi-Fi and Bluetooth capabilities enable seamless data exchange and facilitate commands or data transmission to and from the robot. All in all, the ESP32 microcontroller is an excellent choice for a line-following robot's primary control unit. It organizes sensor data well, calculates control signals, and allows connectivity with external devices. With its versatility and advanced features, the ESP32 microcontroller excels at performing various tasks in this application.

B. IR Sensor

IR sensors are the most common sensors that we used to detect and measure infrared radiation from the obstacles. The main advantage of an IR sensor is that it is able to detect the infrared light even in the daytime. It is mainly used to detect the white and black surfaces. IR sensor works by measuring the reflected right from the object which emits from IR LED. The main components of the IR sensors are the diode send rays and diode receives rays. This shown in Figure 2: Function of IR sensor

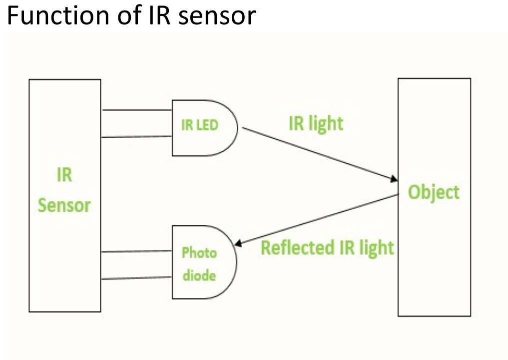

Figure 2: Function of IR sensor [8]

The robot detects the white surface when the emitted rays reflect back to the photodiode and it will detect as black if the photodiode cannot receive a reflection. We used 2 IR sensors for this project

C. Ultrasonic Sensor

The HC-SR04 Figure 3 is a low-cost, easy-to-use distance sensor. It can detect an object in the range of 2cm to 400cm, which is equal to an inch to 13 feet. They're mostly used in a variety of industries, including industrial automation, robotics, automotive, healthcare, and security systems. The transducers in the sensor are used for the transmitter and for the receiver and it serves for a particular function. Transmitter emits the Ultrasonic waves and the receiver receives the signal when an object is detected.

The sensors will calculate the distance by calculating the time taken for the waves to return[11].

Technical Specification:
Power Supply − +5V DC
Quiescent Current − <2mA
Working Current − 15mA
Effectual Angle − <15°
Ranging Distance − 2cm – 400 cm
Resolution − 0.3 cm
Measuring Angle − 30 degree

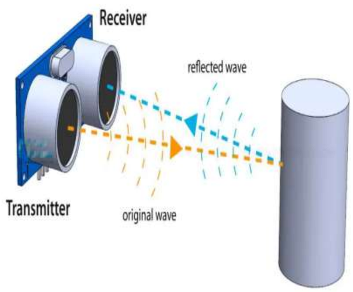

Figure 3: Ultrasonic principle [9]

The HC-SR04 sensor emits ultrasound at 40,000 Hz, which travels through the air. If an object is in its path, the sound waves bounce back to the module. By measuring the time it takes for the waves to return and using the speed of sound, the distance can be calculated. To generate the ultrasound, the Trig pin is set to a High State for 10 µs, sending out an 8 cycle burst. The Echo pin goes high immediately after the burst is sent and listens for the reflected wave. If no reflection is received within 38ms, the Echo pin goes low. The duration of the Echo pin being high helps determine the distance traveled by the sound wave and the distance to the object.

D. L298N MOTOR DRIVER

The L298N Figure 4: L298N motor driver pinouts motor driver IC is widely used in embedded systems, particularly in robotics. Microcontrollers that operate at low voltages and currents (such as 5V) are incapable of providing the higher currents required by motors. Motor driver ICs is utilized to address this issue. A motor driver functions as a current amplifier, amplifying a low current signal to produce a high current signal capable of operating a motor. It also allows you to regulate the direction of the motor. Motor drivers are classified into several types based on parameters such as maximum supply voltage, maximum output current, power dissipation capacity, load voltage, and number of outputs, among others.

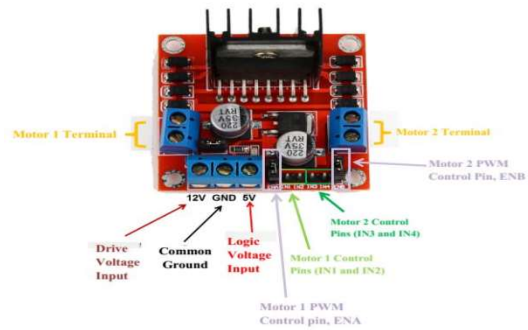

Figure 4: L298N motor driver pinouts [10]

The L298N motor driver IC has numerous essential characteristics that make it a viable choice in a variety of applications. It can manage larger voltage requirements due to its high maximum supply voltage of 46V. The IC can generate a maximum output DC current of 4A, which is sufficient for driving motors. Another benefit is that it has a low saturation voltage, which reduces power loss and ensures efficient operation. Furthermore, the L298N has over-temperature protection to defend against excessive heat and potential damage. Furthermore, it can interpret logical "0" input voltage signals as low as 1.5V, making it compatible with a variety of microcontrollers and logic systems.

In this application, the L298N motor driver IC is used to regulate and steer the stepper motor's motion based on the robot's position relative to a line. The L298N provides fine control over the motor's speed and direction by utilizing input and PWM signals. This control mechanism provides for perfect navigation and alignment by directing the robot along the defined line.

E. DC Motor

A motor Figure 5 :DC Motor is an electrically powered device that transfers electrical energy into mechanical energy. It is largely utilized in manufacturing and robotics. They generate rotational motion by interacting with a magnetic field and an electric current. This is accomplished by the use of magnetic fields and the passage of electric current. The Lorentz force is experienced by a current-carrying conductor when it is put in a magnetic field. This force causes the conductor (in the form of coils) in a DC motor to rotate, resulting in mechanical motion of the motor's shaft.

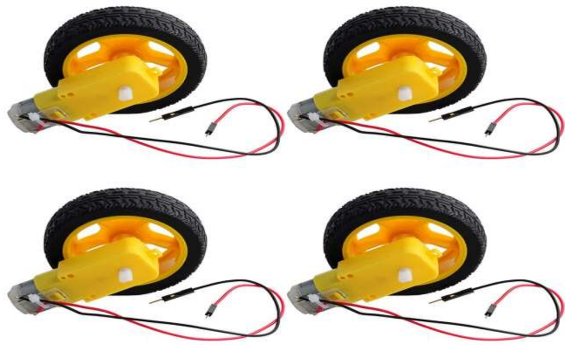

Figure 5 :DC Motor [11]

The speed and direction of the motor can be modified as needed by regulating the current and magnetic field.

F. Robot Car Kit

The robot car kit we are utilizing comprises a pair of geared motors that serve the purpose of propelling the robot and enabling it to change directions. The motor speed and output is determined by the output. Also, The robot car kit has an ON/OFF switch for controlling the power supply and a battery holder is connected to power the robot. This kit is used to build a simple line following the robot, including motors, power supplies, and important control systems.

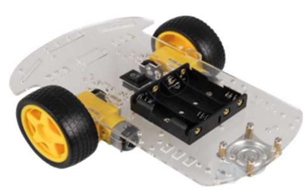

Figure 6: Robot Kit [12]

Moreover, sensors, microcontrollers, and programming are necessary for comprehensive customization and control of the robot's behavior. The Robot's chase is 200mm length and 100mm width and it is made up of strong acrylic with multiple holes to fix the hardware components. Furthermore, the robot has tires of 60 x 27 mm in size, which is used for mobility.

G. Power Supply

Lithium-ion batteries are widely used in portable devices such as electric vehicles, computers, electronics, and cell phones due to their rechargeable nature. They are also finding increased applications in military and aerospace fields. In our project we required a 12V supply for this project to improve robots performance and ability to navigate. So we used 4 batteries that typically provide a voltage output of 4v each. However, we didn't use the Lithium-ion battery since it cannot give the required voltage. The single use battery is found to be efficient and can generate 4-5V. We connected the 4 batteries in series to get the desired voltage. This configuration allows us to meet the voltage requirement for the robot's efficient operation.

IV. SCHEMATIC DIAGRAM

A schematic diagram Figure 7: Circuit diagram is a visual representation of a system, device, or process, using standardized symbols and lines to depict its essential components. It provides an overview of the system's parts, focusing on the key elements while omitting unnecessary details. In some cases, the schematic diagram may include exaggerated or simplified representations to aid in understanding the system. For the specific case of a "line-taking robot," the primary component is the Esp32 microcontroller. KiCAD, a software package developed for schematic design and layout, was used to build the schematic diagram for this robot. The diagram highlights the robot's key features and components, such as the Esp32 microcontroller, an intelligent sensor with an IR-LED for infrared lighting, a quad comparator IC (LM324) with a tuning potentiometer for reference voltage, and the motor control IC (L298N) connected to the H-bridge drivers that control the motors with reduction gears. The schematic also includes connectors to ease the joining of various modules and sheets, resulting in a coherent and functional entity. The tachometer, an important sensor device, is designed with a combination of IR-LEDs and photodiodes. Before being incorporated into the final application, each component of the system is developed and implemented individually, fulfilling a specific role.

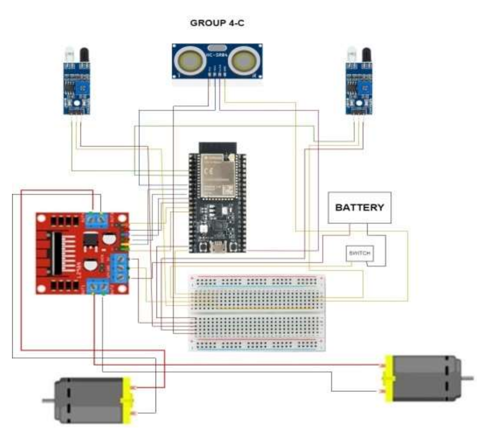

Figure 7: Circuit diagram

V. DEVELOPMENT

A. SOFTWARE REQUIREMENT

- Operating System Windows 10
- Arduino IDE
- Serial Communication Tools

The Arduino programming environment incorporates C++ methods and functions, which are utilized to create code for Arduino boards. The Arduino modules are Arduino based microcontroller based development board packages which allows a human-readable programming language, such as C++, to develop the code.

The Arduino IDE is designed to establish a connection with Arduino hardware, enabling the upload and communication of programs. Sketches, which are software programs developed using the Arduino IDE, play a central role in this process. These sketches are created within the IDE's text editor and are saved with a specific file extension. The text editor offers useful features such as text replacement and searching, facilitating efficient code development. Throughout the process, the message area provides feedback and displays information during tasks such as saving and exporting code.

B. Flow of Robot

The Figure 8: Flow chart of the whole process depicts the sequence of actions and the logical design of the robot based on its input and output parameters. It visually represents how the robot's behavior and decision-making process are structured, considering the logic behind the input it receives and the corresponding output it produces. There are mainly four steps for controlling a robot. Initially, the robot uses its sensors that allow it to determine its location on the track.

These sensors are responsible for constantly monitoring and giving feedback to microcontroller if the Robot is in a defined path or not. If both the sensors detect the white line, the robot will move forward along the track. The robot is programmed in such a way that if the left sensor detects the path ,the speed of the left wheel increases by keeping the right wheel aligned by sending the input voltage as zero and vice versa. It is programmed to detect objects or edges of the object.

There are two conditions to identify the object. On one hand, if the ultrasonic sensor does not detect any object, the robot will move forward in the path. On the other hand, if the robot detects an object, the robot will move left and avoid the obstacle and then rejoins the line by a shape of half rectangle. In this case, the robot will proceed without interruptions or external intervention. Since no obstacles or diversions are detected along path, it will follow its present direction and speed without making any changes. The robot will continue to follow its defined directions until it either discovers the track again or reaches its target. This function enables the robot to cover a broad area without regularly pausing, enhancing its efficiency and saving time. The logic is explained in Table 1: Movements on sensor value

Table 1: Movements on sensor value

Input			Output		Movements of robot
Left sensor	Right sensor	Ultrasonic sensor	Left Actuator	Right Actuator	
LS	RS	US	LM	RM	
0	0	0	1	1	FORWARD
1	0	0	0	1	LEFT
0	1	0	1	0	RIGHT
0	0	1	0	0	Deviate from the track and join again after certain movemennt
1	1	0	0	0	STOP

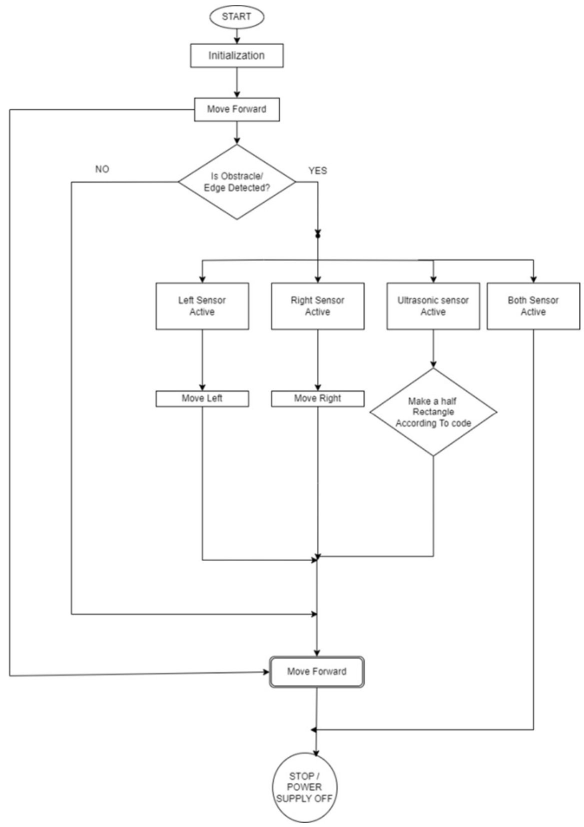

Figure 8: Flow chart of the whole process

VI. TESTING AND RESULTS

The line following robot is designed to navigate a path by following a line using 2 IR sensors and 1 ultrasonic sensor.

The purpose of this testing is to evaluate the performance and functionality of the robot's line following an obstacle avoidance capabilities.

During the testing, the robot's movements and sensor readings were observed and recorded. The following procedures were carried out to assess the robot's performance.

To ensure accurate line following and obstacle detection, calibration of the infrared sensors and the ultrasonic sensor is essential. The report provides guidelines and procedures for calibrating these sensors to achieve optimal performance.

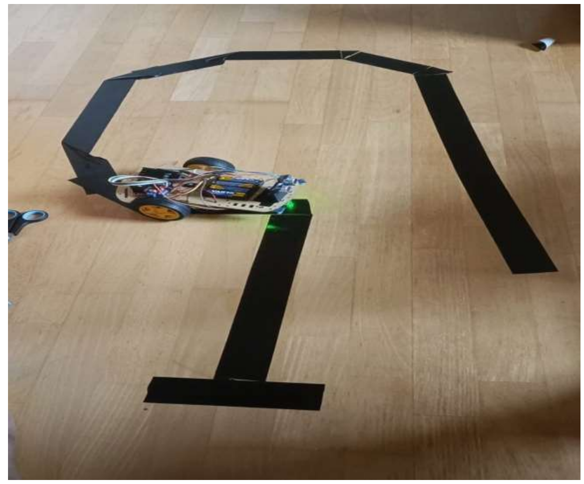

Figure 9: Tested path

Analyze the data collected during the testing phase, including the accuracy of line following, the detection range of the ultrasonic sensor, and the success rate of obstacle avoidance.

Compare the robot's performance against predetermined performance metrics and project requirements.

Identify any limitations or areas for improvement, such as response time, sensitivity to different line colors, or the need for sensor calibration.

A. Line Following Test:

- Place the robot on a black line or track,which shown in Figure 9: Tested path with sufficient contrast between the line and the surface.Table 1: Movements on sensor value

- Activate the robot by uploading the provided code to the microcontroller.

-Start the robot at the beginning of the line and observe its behavior.

-Document the robot's response to different line configurations, including straight lines, curves, intersections, and T-junctions.

-Measure the robot's accuracy in staying on the line, recording any deviations or corrections made.

-Conduct multiple repetitions of the line following tests to ensure consistency in the robot's performance.

- Observe the robot's behavior as it attempts to follow the line.

- Record any deviations from the line, such as overshooting or undershooting.

Sl.No	Line width	Time(S)	Distance(M)
1	1.5	5	2.4
2	1.3	5.3	2.4
3	1.2	5.8	2.4

B. Ultrasonic Sensor Test:

The purpose of this testing is to ensure reliable obstacle detection capabilities, which are crucial for effective obstacle avoidance and navigation.

- Position obstacles of varying distances in front of the robot.

- Ensure that the ultrasonic sensor is facing forward and fixed place.

-Connect the ultrasonic sensor to the appropriate pins on the ESP32 microcontroller according to the wiring specifications.

-Ensure that the ESP32 microcontroller is properly programmed with the necessary code to read and process the ultrasonic sensor data.

-Create a testing environment with a variety of obstacles positioned at different distances from the sensor.

-Set up a monitoring system to record the sensor readings during the testing process. Monitor the ultrasonic sensor readings displayed in the serial monitor.

- Compare the measured distances with the actual distances to evaluate the accuracy of the sensor.

Before proceeding with the testing, perform accuracy checks and calibration of the ultrasonic sensor:

1. Place a known obstacle at a specific distance from the sensor.
2. Place a known obstacle at a specific distance from the sensor.
3. Adjust any calibration factors, if necessary, to improve the accuracy of the sensor readings.

C. Obstacle Avoidance Test:

- Start the line following robot and ensure the ultrasonic sensor is actively measuring the distance**Error! Reference source not found.**.

- Gradually move various obstacles within the sensor's detection range and record the corresponding sensor readings.

- Verify that the sensor detects the obstacles reliably and consistently, providing accurate distance measurements.

- Test the sensor's performance with different obstacle shapes, sizes, and materials to assess its versatility. Position obstacles incrementally at different distances from the sensor, ranging from the minimum to the maximum range.

- Observe the robot's response when it detects an obstacle.

- Verify that the robot stops or changes direction to avoid the obstacle.

- Assess the robot's ability to navigate around the obstacle and continue following the line.

- Record the sensor readings for each obstacle distance and evaluate the consistency and reliability of the measurements.

- Determine the maximum detection range of the sensor and verify that it aligns with the specifications provided by the manufacturer.

D. Turn (After obstacle avoidance) Testing:

Prepare a testing track with a clearly defined line path using contrasting colors for accurate line detection. Position the line following robot at the start of the line and ensure all sensors are active.

- Start the line following robot and monitor its behavior as it follows the line.

- Observe the robot's behavior when making turns.

- Ensure that the robot makes appropriate turns at predefined angles.

- Assess the accuracy and precision of the robot's turning capabilities.

-Conduct multiple repetitions of the test to ensure consistency in the robot's performance.

- Record any instances of turning errors or deviations.

Measure the time taken by the robot to detect the line deviation, execute the turn, and successfully reconnect with the line.

Sl.No	Line width	Time	Distance
1	1.5	7.9	2.4
2	1.3	10.9	2.4
3	1.2	12.6	2.4

E. Robustness and Reliability Testing:

- Run the robot through various line configurations, such as curves, intersections, and sharp turns**Error! Reference source not found.**.

- Assess the robot's ability to adapt to different line patterns and successfully navigate the path.

- Evaluate the robot's reliability by conducting multiple test runs and checking for consistency in performance.

Sl.No	Line width	Time	Distance
1	1.5	7.9	2.4
2	1.3	10.9	2.4
3	1.2	12.6	2.4

Throughout the testing, it is essential to document any issues or areas of improvement. This includes noting any discrepancies in sensor readings, unexpected behavior, or limitations in the robot's performance. To ensure the validity of the testing process, multiple iterations of the tests may be conducted to validate the results and identify any potential improvements or modifications required.

VII. Conclusion & Future Scope

In this work, the proposed autonomous line following robot with obstacle detection and avoidance in the straight path is successfully completed. The performance and efficiency of the robot in different environments, was validated. Apart from the minor errors, occurred due to the inefficiency of the components used, the prototype works as defined. The sensors used were not efficient enough to follow a path especially in the curved and sharp edges. Even though, the prototype shows promising results, further improvements in the results can be achieved by incorporating better components such as sensors like QRE sensors, ultrasonic sensors mounted one servo motor, which detects bigger objects lying within 180°. Summarizing this report, the robot is successful in detecting and avoiding the obstacle within a specified shape.

REFERENCES

[1] W. Rahmaniar and A. Wicaksono, “Design and Implementation of a Mobile Robot for Carbon Monoxide Monitoring,” Journal of Robotics and Control (JRC), vol. 2, no. 1, 2020.

[2] L. K. Amifia, M. I. Riansyah, and P. D. Putra, “Design of Logistic Transporter Robot System,” Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, vol. 6, no. 1, p. 19, 2020

[3] Wang, Z., Majewicz Fey, A. “Deep learning with convolutional neural network for objective skill evaluation in robot-assisted surgery”, Int J CARS 13, 1959–1970, 2018.

[4]M. G. Bechtel, E. Mcellhiney, M. Kim, and H. Yun, "DeepPicar: A LowCost Deep Neural Network-Based Autonomous Car," IEEE 2 International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Hakodate, pp. 11-21, 2018

[5] Atkeson C.G. et al. “What Happened at the DARPA Robotics Challenge Finals” The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue. Springer Tracts in Advanced Robotics, vol 121. Springer, Cham, 2018.

[6] N. H. B. Abd Khafar, D. M. Soomro, R. Abd Rahman, M. N. Abdullah, I. A. Soho, and A. A.

Rahimoon, "Intelligent navigation system using ultrasonic sensor for visually impaired person (VIP)," in 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS), 2019, pp. 1-4

[7]https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html

[8] https://robu.in/ir-sensor-working/

[9]https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

[10]https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

[11] https://www.amazon.de/Elektro-Getriebe-Doppelwellengetriebe-Kunststoffreifen-Arduino/dp/B096Z5C3R4/ref=asc_df_B096Z5C3R4/?tag=googshopde-21&linkCode=df0&hvadid=546555495341&hvpos=&hvnetw=g&hvrand=6670663819344863714&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9041610&hvtargid=pla-1394103073416&th=1

[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.eurocircuits.de/blog/kicad-eurocircuits-akzeptiert-kicad-daten-als-natives-format/

APPENDIX 1

CODE OF THIS ROBOT

```
#include <Ultrasonic.h>

#define Left_IR 2
#define Right_IR 3

//ultrasonic
#define TRIGGER_PIN 5
#define ECHO_PIN 18

#define SOUND_SPEED 0.034
#define CM_TO_INCH 0.393701

//left motor data
#define Lmotor_ENA 11
#define LEFT_MOTOR_INPUT1 7
#define LEFT_MOTOR_INPUT2 6

//right motor data
#define Rmotor_ENB 10
#define RIGHT_MOTOR_INPUT3 8
#define RIGHT_MOTOR_INPUT4 9

int SPEED = 125; // speed of this
line following robot
float DIST;
boolean object;

//front movement
void ROBO_FRONT() {
 analogWrite(Lmotor_ENA, SPEED);
 analogWrite(Rmotor_ENB, SPEED);
 digitalWrite(LEFT_MOTOR_INPUT1,
HIGH);
 digitalWrite(LEFT_MOTOR_INPUT2,
LOW);
 digitalWrite(RIGHT_MOTOR_INPUT3,
HIGH);
 digitalWrite(RIGHT_MOTOR_INPUT4,
LOW);
}


//turn left
void ROBO_TURN_LEFT() {
 analogWrite(Lmotor_ENA, SPEED);
 analogWrite(Rmotor_ENB, SPEED);
 digitalWrite(LEFT_MOTOR_INPUT1,
LOW);
 digitalWrite(LEFT_MOTOR_INPUT2,
HIGH);
 digitalWrite(RIGHT_MOTOR_INPUT3,
HIGH);
 digitalWrite(RIGHT_MOTOR_INPUT4,
LOW);
}


//turn right
void ROBO_TURN_RIGHT() {
 analogWrite(Lmotor_ENA, SPEED);
 analogWrite(Rmotor_ENB, SPEED);
 digitalWrite(LEFT_MOTOR_INPUT1,
HIGH);
 digitalWrite(LEFT_MOTOR_INPUT2,
LOW);
 digitalWrite(RIGHT_MOTOR_INPUT3,
LOW);
 digitalWrite(RIGHT_MOTOR_INPUT4,
HIGH);
}


//stop
void ROBO_STOP() {
 digitalWrite(LEFT_MOTOR_INPUT1,
LOW);
 digitalWrite(LEFT_MOTOR_INPUT2,
LOW);
 digitalWrite(RIGHT_MOTOR_INPUT3,
LOW);
 digitalWrite(RIGHT_MOTOR_INPUT4,
LOW);
}


void turn() {

    Serial.println("turn
activated");
    ROBO_TURN_RIGHT();
    delay(500);
    ROBO_FRONT();
    delay(1500);
    ROBO_TURN_LEFT();
    delay(400);
    ROBO_FRONT();
    delay(450);
    ROBO_TURN_LEFT();
    delay(100);
```

```
}

//ultrasonic reading
long ULTRASONIC_READING(){

  digitalWrite(TRIGGER_PIN, LOW);
  delayMicroseconds(2);
  //trigPin on HIGH state for 5
micro seconds
  digitalWrite(TRIGGER_PIN, HIGH);
  delayMicroseconds(5);
  digitalWrite(TRIGGER_PIN, LOW);
  long dure = pulseIn(ECHO_PIN,
HIGH);
  return dure*SOUND_SPEED/2;

}

void setup() {
 Serial.begin(115200);

 pinMode(ECHO_PIN, INPUT);
pinMode(TRIGGER_PIN, OUTPUT);

 pinMode(Left_IR, INPUT);
 pinMode(Right_IR, INPUT);
 pinMode(Lmotor_ENA, OUTPUT);
 pinMode(LEFT_MOTOR_INPUT1,
OUTPUT);
 pinMode(LEFT_MOTOR_INPUT2,
OUTPUT);
 pinMode(RIGHT_MOTOR_INPUT3,
OUTPUT);
 pinMode(RIGHT_MOTOR_INPUT4,
OUTPUT);
 pinMode(Rmotor_ENB, OUTPUT);

}

void loop() {
 bool IR_L = digitalRead(Left_IR);
 bool IR_R =
digitalRead(Right_IR);
  DIST = ULTRASONIC_READING();

  Serial.println(DIST);

  if (DIST >=20)  {

   if (IR_L == 0 && IR_R == 0) {
 ROBO_FRONT();
Serial.println("forward");
 } else if (IR_L == 1 && IR_R ==
1) {
 ROBO_STOP();
 } else if (IR_L == 0 && IR_R ==
1) {
 ROBO_TURN_LEFT();
 Serial.println("forward left");
 } else if (IR_L == 1 && IR_R ==
0) {
 ROBO_TURN_RIGHT();
 Serial.println("forward right");
 }
 } else {
    Serial.println("object
detected");
    turn();
  }
}
```