FABRICATION AND PERFORMANCE OF AUTONOMOUS VEHICLE

Ms. D. Sravana Lakshmi, Mr.G. Bhaskar, Mr. K. Jagan Jeevan Kumar,Mr. S. Lokesh Department of Mechanical Engineering, Nadimpalli Satyanarayana Raju Institute of Technology,Visakhapatnam.
Mrs. B Usha Rani, Sr. Assistant Professor, Department of Mechanical Engineering, Nadimpalli Satyanarayana Raju Institute of Technology, Visakhapatnam.

Abstract:

Line following and obstacle avoidance are essential functionalities in robotics. A Line Following Robotutilizes arrays of optical sensors to track and follow lines autonomously. Equipped with four sensors, it ensures precise movement along the line. DC gear motors drive its wheels, controlled by an Arduino Uno interfacing algorithms for speed regulation and steering. Additionally, an LCD interface displays the traveled distance. This versatile robot finds applications in industrial automation, household tasks, and guiding systems in museums. Conversely, an obstacle avoidance robotic vehicle, built with an ATmega328 microcontroller and ultrasonic sensors, demonstrates intelligent navigation. When obstacles are detected, the microcontroller redirects the vehicle by actuating motors through a motor driver. Unlike infrared sensors, the ultrasonic sensors offer compatibility and reliability.

This project amalgamates both functionalities to create a comprehensive robotic system capable of navigating along lines while autonomously avoiding obstacles. Key components include the Arduino UNO, motor shield L293d, ultrasonic sensor HC-SR04, DC motors, and servo motor.represents a significant leap towards a future where vehicles navigate with unprecedented efficiency and safety, heralding a new era of transportation.

Keywords:

Line Following Robot, Optical Sensors, DC Gear Motors, Arduino Uno.

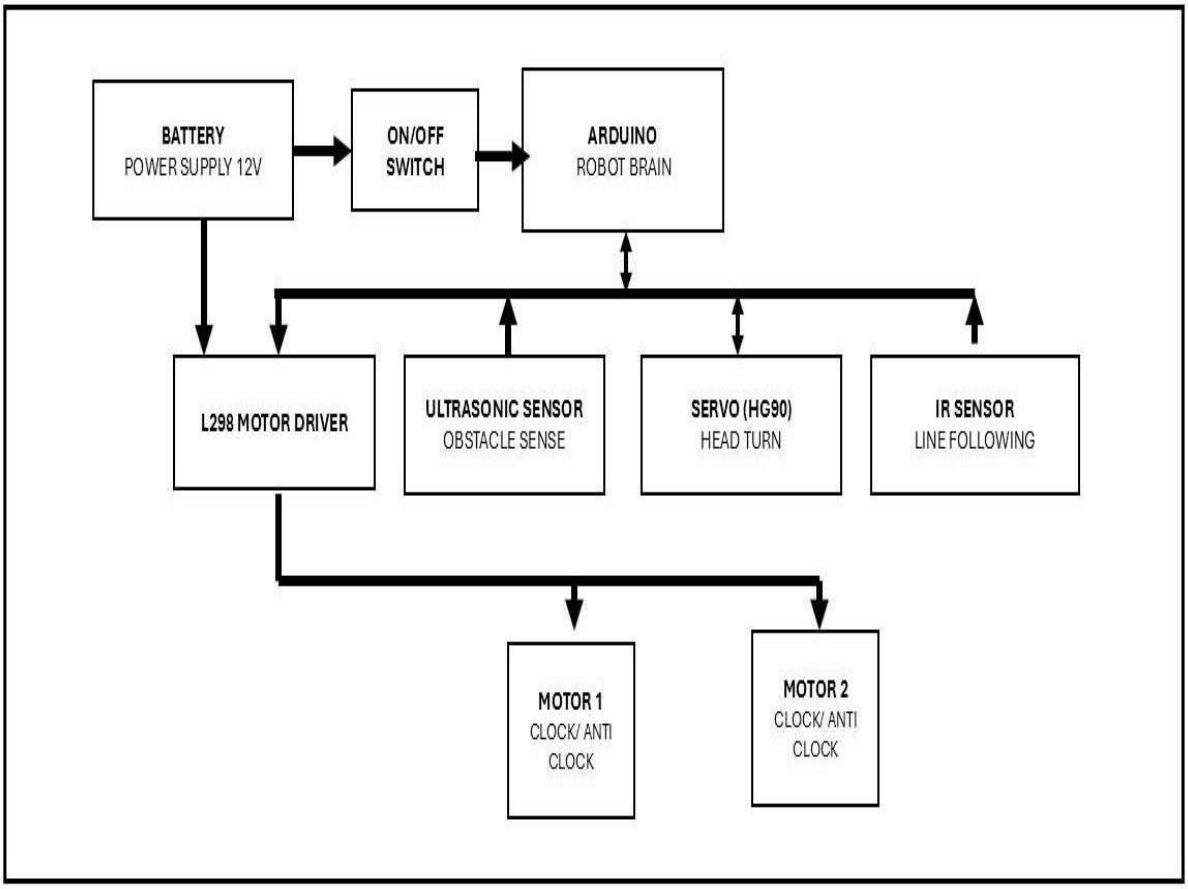
1. Introduction:

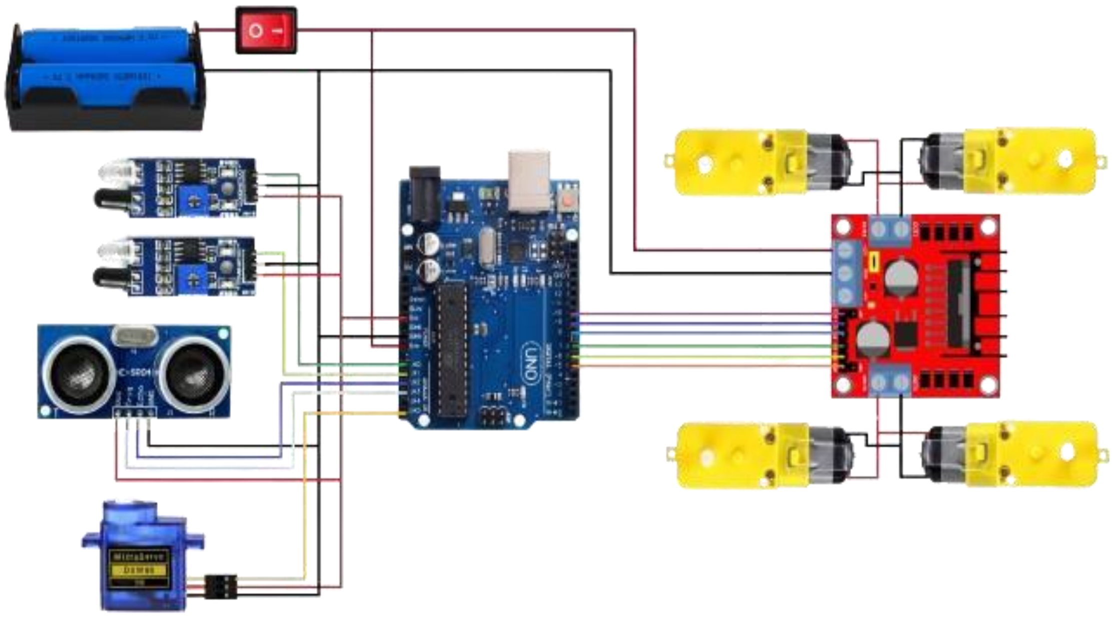
The objective of developing a path Following and Obstacle Avoidance Robot is to create a versatile robotic system capable of autonomously navigating complex environments with precision and efficiency. This entails integrating two fundamental functionalities: line following and obstacle avoidance.

Firstly, the objective is to enable the robot to follow predefined paths marked by lines accurately. By utilizing arrays of optical sensors, the robot detects and tracks lines on the surface, ensuring precise movement along the desired path. The goal is to implement robust algorithms that interpret sensordata and control the robot's motion, allowing it to follow lines autonomously. Additionally, the systemaims to achieve adaptability to different line patterns and environmental conditions, ensuring reliable performance across various scenarios.

Secondly, the objective extends to equipping the robot with obstacle avoidance capabilities to navigatesafely in dynamic environments. By incorporating sensors such as ultrasonic or infrared sensors, the robot detects obstacles in its path and autonomously maneuvers to avoid collisions. The objective is todevelop intelligent algorithms that analyze sensor data, identify obstacles, and generate appropriate navigation commands to steer the robot away from potential collisions. This involves integrating obstacle detection and avoidance seamlessly with the line following functionality, ensuring smoothand uninterrupted navigation along the predefined path. Furthermore, the objective includes optimizing the robot's hardware and software components for efficiency and reliability. This encompasses selecting suitable actuators such as DC motors for propulsion and motor drivers for precise control of movement. Additionally, the objective involves designing a robust microcontroller- based system, such as Arduino or Raspberry Pi, capable of executing complex algorithms for sensor data processing and decision-making.

Overall, the objective of developing a Line Following and Obstacle Avoidance Robot is to create a versatile and intelligent robotic system capable of navigating autonomously in diverse environments, contributing to advancements in robotics for applications such as industrial automation, logistics, and service robotics.





ROLE OF OBSTACLE AVOIDANCE AND PATH FOLLOWING ROBOT IN DIFFERENT INDUSTRIES

Obstacle avoidance and path-following robots play significant roles across various industries, offering solutions to diverse challenges and enhancing operational efficiency. Here's a breakdown of their roles in different sectors:

- ➢ **Manufacturing:** In manufacturing, these robots are employed for material handling, assembly line operations, and warehouse navigation. They ensure safe and efficient movementof goods within the facility, optimize workflow, and minimize the risk of collisions between robots and workers or equipment.
- ➢ **Logistics and Warehousing:** Obstacle avoidance and path-following robots are utilized for automated guided vehicle (AGV) systems and autonomous mobile robots (AMRs) in logistics and warehousing operations. They navigate through warehouses, transporting goods between storage locations, picking stations, and shipping docks. These robots streamline order fulfilment processes, reduce manual labour, and improve inventory management.
- ➢ **Security and Surveillance:** Obstacle avoidance and path-following robots are employed for security patrols, surveillance, and monitoring in various environments. They navigate through indoor and outdoor spaces, avoiding obstacles and following predefined routes to detectand respond to security threats effectively
- ➢ **Healthcare:** In healthcare facilities, obstacle avoidance and path-following robots assist in delivering medical supplies, transporting equipment, and navigating hospital corridors. They contribute to the efficient flow of materials and resources within the hospital environment, freeing up healthcare professionals to focus on patient care tasks.
- ➢ **Agriculture:** Agriculture benefits from these robots for tasks such as crop monitoring, spraying pesticides, and harvesting. They navigate through fields, avoiding obstacles and following crop rows to perform targeted operations with precision. These robots optimizeresource utilization, reduce labour costs, and improve crop yields.

METHODOLOGY

AIM

This vehicle is a self-driving car that employs artificial intelligence and computer vision. In this project, we create a car with a camera that detects the track or path around it. If the automobile is about to steer into one, it should respond. The "eye" of self-driving vehicles is computer vision with an AI-based algorithm. The primary goal of computer vision is to provide a smooth self driving experience. The foundation of Artificial Intelligence technology is computer vision. AI assists computers in decoding and comprehending visual input obtained from diverse sources. It entails employing AI algorithms to do autonomous visual understanding. The facial recognition technology is the best example of computer vision. This model employs the Canny edge detectiontechnique. This variant is mostly utilized in industries to assist in moving and transferring things from one end to the other. As a result, it reduces the need for human intervention and makes the work easier.

EXISTING SYSTEM:

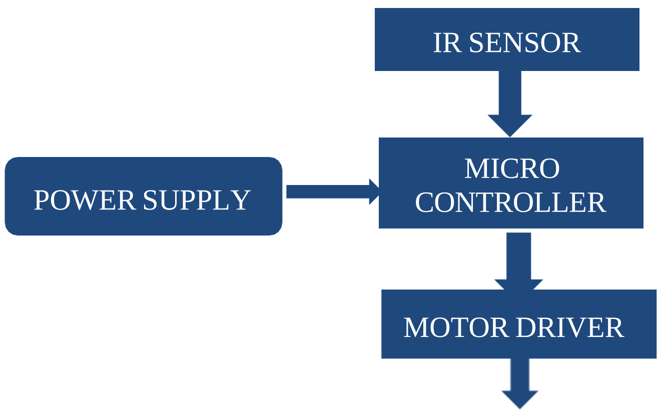


Fig Existing System

The system uses a microcontroller with an IR sensor to reach the destination. It follows the desiredpath or line and moves automatically. The IR sensor senses the and the information is given to themicrocontroller.

This system is a basic system which has low features, and it also has various disadvantages likepoorworking conditions at night time and in cloudy weather.

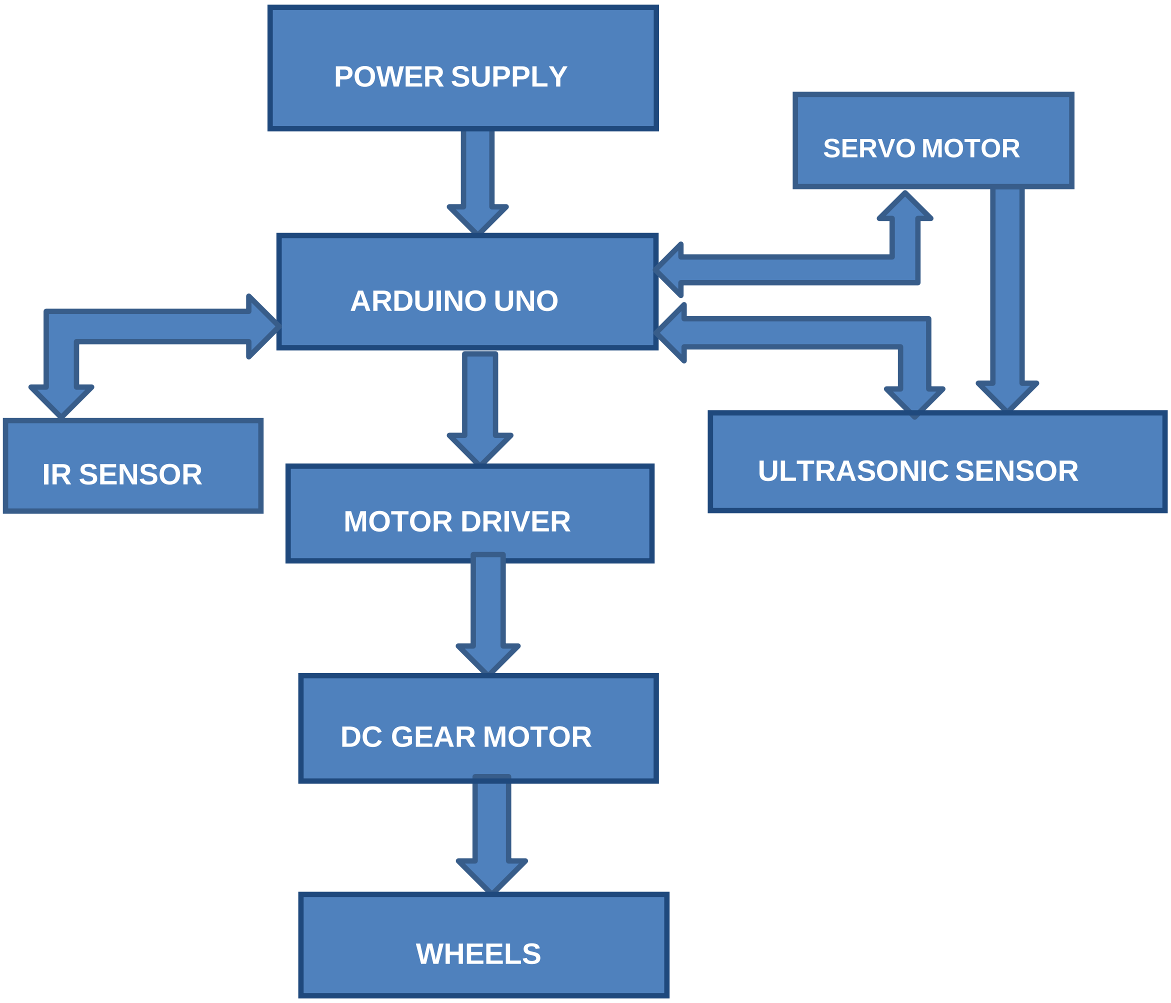


Fig Projected System

WORKING OF PROJECTED SYSTEM:

The integrated obstacle avoiding and path-following robot functions by utilizing an array of sensors, algorithms, and actuators to autonomously traverse its surroundings. Equipped with sensors such as ultrasonic, infrared, the robot continuously gathers data about its environment. This data is then processed inreal-time by the onboard computer, enabling the robot to detect obstacles along its path. Simultaneously, algorithms interpret sensor data to ensure the robot stays on its intended path or trajectory, which may involve following marked lines or utilizing GPS coordinates. In the event of an obstacle detection, the control systemrecalculates a new trajectory to navigate around the obstacle while maintaining its course towards the destination. Actuators, such as motors or servos, are then utilized to adjust the robot's motion based on the calculated trajectory, ensuring safe navigation and collision avoidance. This process operates within a feedback loop, allowing the robot to continuously adapt its behaviour to changes in its environment.Through the integration of obstacle avoidance and path-following capabilities, the robot achieves autonomous navigation in dynamic environments, effectively avoiding obstacles and adhering to predefined paths to fulfil its objectives.

METHODS AND ALGORITHM:

The method used in obstacle-avoiding path-following robots that incorporate bothultrasonic sensors and IR sensors typically involves the following steps:

- **Sensor Data Acquisition:** The robot is equipped with both ultrasonic sensors and IR sensors placed strategically around its body. These sensors continuously collect data aboutthe robot's surroundings.
- Obstacle Detection:

- **Ultrasonic Sensors:** Ultrasonic sensors emit high-frequency sound waves and measurethe time taken for the waves to reflect back from obstacles. The robot's onboard computer processes this data to detect obstacles within a certain range.
- **IR Sensors:** IR sensors emit infrared light and measure the reflection or absence of thislight to detect obstacles. They are effective at detecting nearby objects and can be usedin conjunction with ultrasonic sensors for enhanced obstacle detection.

- **Path Following:** Algorithms interpret sensor data to ensure the robot stays on its intendedpath or trajectory. This could involve following lines marked on the ground or using othervisual cues. IR sensors can be particularly useful for path following when the robot is designed to follow lines or tracks with contrasting colors.
- **Collision Avoidance:**When an obstacle is detected, the robot's control system calculates anew trajectory to avoid the obstacle while staying on course to reach its destination.
- The combination of ultrasonic sensors and IR sensors allows for robust obstacle detectionand accurate calculation of safe paths to navigate around obstacles.
- **Actuator Control:**Based on the calculated trajectory, the robot's actuators, such as motorsor servos, are instructed to adjust the robot's motion accordingly. This may involve steeringthe wheels, adjusting the speed, or changing the direction of movement to avoid collisionsand stay on track.
- **Feedback Control:**The robot continuously monitors its environment using sensors and adjusts its behavior in real-time based on new information. This closed-loop feedback

  system ensures that the robot can adapt to changes in its surroundings and navigate effectively.

  By integrating both ultrasonic sensors and IR sensors, the robot can effectively detect obstacles and follow predefined paths autonomously in dynamic environments

4.1 HARDWARE COMPONENTS REQUIREMENTS

1. ARDUINO UNO
2. ULTRASONIC SENSOR
3. IR SENSOR
4. L298N MOTOR DRIVER
5. DC GEAR MOTOR

INTRODUCTION TO ARDUINO UNO:

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller (MCU) and developed by Arduino.cc and initially released in 2010. The microcontroller board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/Opins (six capable of PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE(Integrated Development Environment), via a type B USB cable. It can be powered by a USB cableor a barrel connector that accepts voltages between 7 and 20 volts, such as a rectangular 9-volt battery. It has the same microcontroller as the Arduino Nano board, and the same headers as the Leonardo board.

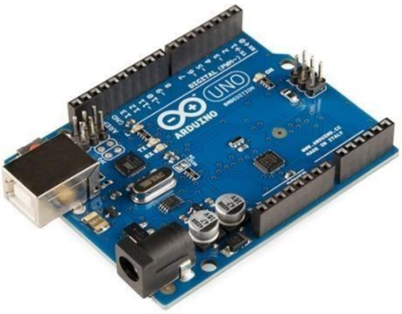
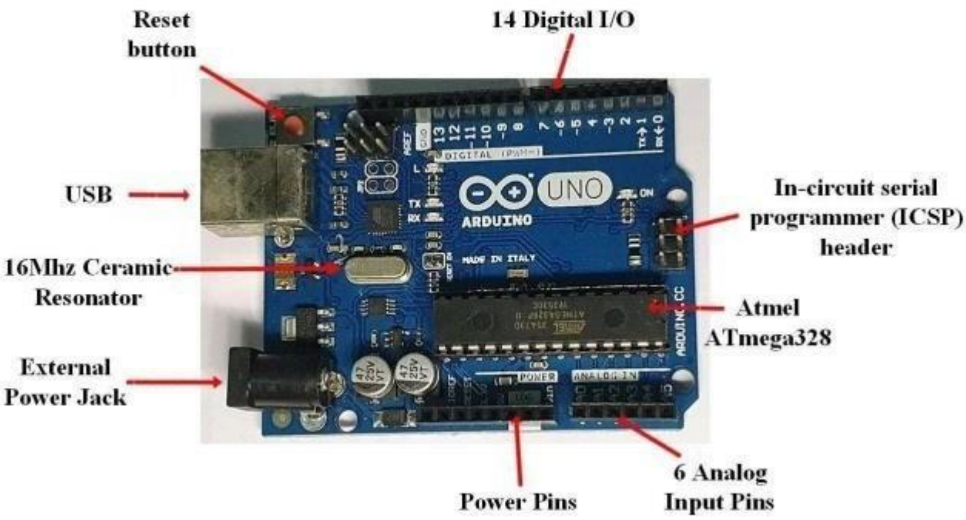


Fig Arduino UNO

4.1.1.1 POWER OVERVIEW OF ARDUINO UNO R3

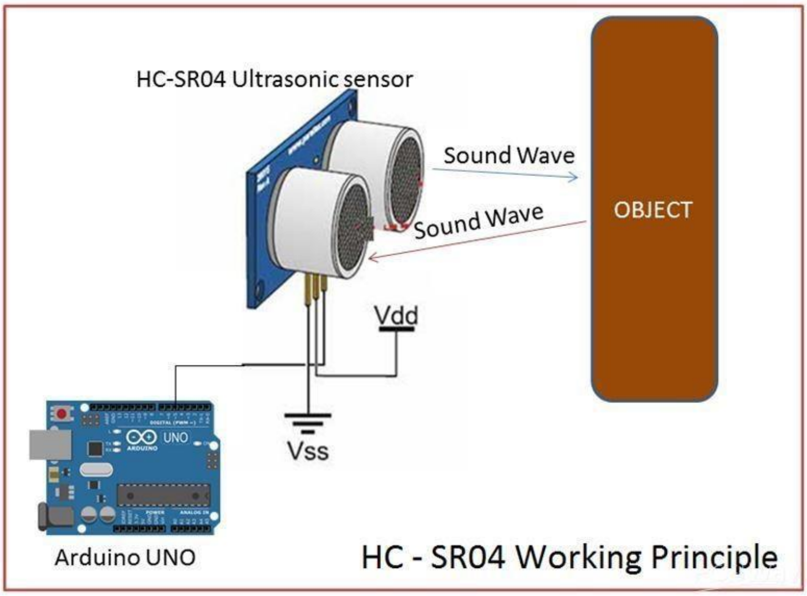


ULTRASONIC SENSOR

A transmitter and receiver are included with the HC-SR04 ultrasonic sensor. This sensor is used to determine the distance between the target and the sensor. The distance between the sensor and an object is determined by the amount of time it takes to transmit and receive the waves. Using non-contact technology, this sensor detects sound waves. The required distance for the target maybe measured without causing damage using this sensor, and accurate details can be obtained. Thissensor has a range of 2 to 400 centimetres.

4.1.2.1 HC-SR04 ULTRASONIC SENSOR WORKING

HC-SR04 has 4 pins: Vcc, Trigger, Echo, and Ground. This sensor is used to determine the precisedistance between the sensor and the target. This sensor is mostly used to detect sound waves. Whenthis module is given power, it emits sound waves that travel through the air and strike the requiredobject. These waves strike and return from the object, which the receiver module catches. Becausethe time required to travel a greater distance is longer, both the distance and the time spent are directly related. When the trigger pin is held high for 10μs, ultrasonic waves are generated that travel at the speed of sound. As a result, it generates eight cycles of sound bursts, which are collected within the Echo pin. This ultrasonic sensor is connected to an Arduino board to determinethe required distance between sensor and object. The following formula can be used to calculate the distance. $S = (V \times t)/2$ S - required distance V - sound's speed t - time taken for sound waves to return back after striking the object. Because the time it takes for the waves to travel and returnfrom the sensor is twice as long, the real distance can be estimated by dividing the value by two .



Ultrasonic Sensor Working

4.1.1 IR SENSOR

IR sensor is an electronic device, that emits the light in order to sense some object of the surroundings. An IR sensor can measure the heat of an object as well as detects the motion. Usually, in the infrared spectrum, all the objects radiate some form of thermal radiation. These types of radiations are invisible to our eyes, but infrared sensor can detect these radiations.

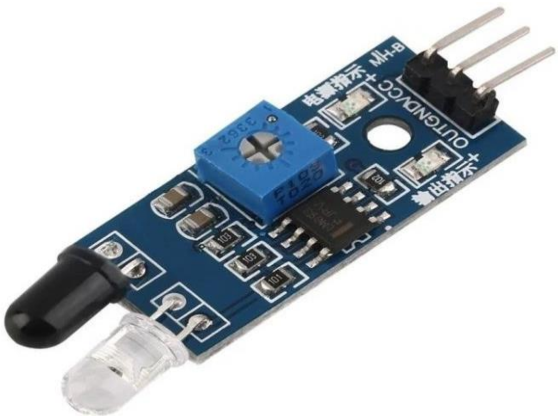


FIG IR Sensor

DC GEAR MOTOR

A gear motor combines a motor and a gearbox into one unit. When a gear head is added to a motor, the speed is reduced but the torque output is increased. In terms of gear motors, the most significant criteria are speed (rpm), torque (lb-in), and efficiency (percent). To choose the best gear motor for your application, you must first calculate the load, speed, and torque requirements for your application. ISL Products has a wide range of Spur Gear Motors, Planetary Gear Motors,and Worm Gear Motors 29 to satisfy your needs. Most of our DC motors can be combined with one of our one-of a-kind gearheads to create a highly efficient gear motor solution.

Fig DC Gear Motor

Raspberry Board

The Raspberry Pi is based on the Broadcom BCM2835 chip and features a 700 MHz ARM1176JZF-S core CPU and 256 MB of SDRAM. The USB 2.0 ports employ only external data communication methods. A micro-USB converter with a 500 MA minimum range powers the Raspberry Pi (2.5 watts). The graphics specialist chip is designed to reduce the time it takes to calculate and manipulate images. It has a 11 Broadcom video core IV cable, which is helpful if you want to utilise your Raspberry Pi to play games and watch videos.

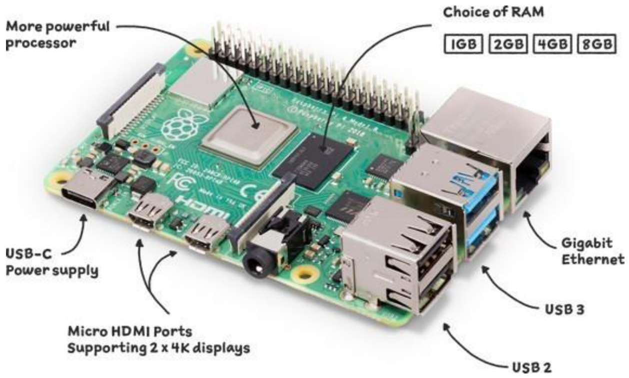


FIGURE 5.1 RASPBERRY PI

Features:

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
- 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previousboards)
- 2 × micro-HDMI® ports (up to 4kp60 supported)
- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.1, Vulkan 1.0
- Micro-SD card slot for loading operating system and data storage
- 5V DC via USB-C connector (minimum 3A*)

- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- Operating temperature: 0 – 50 degrees C ambient.

CAMERA

The camera plays a pivotal role in an autonomous car by providing crucial visual input for navigation and decision-making processes. Here are some key aspects of the camera's role inan autonomous car:

FIG CAMERA

Environmental Perception: The camera captures real-time images of the car's surroundings, providing valuable information about the road, traffic, pedestrians, obstacles, and othervehicles.

Object Detection and Recognition: Using computer vision techniques, the camera helps in detecting and recognizing various objects such as pedestrians, vehicles, cyclists, road signs, traffic signals, and lane markings. This information is essential for understanding the dynamicenvironment and making safe navigation decisions.

PROGRAMMING THE ROBOT

The Arduino microcontroller communicates with the PC via the USB connection. Data istransferred between the board and the PC bit by bit. An adaptor is used for power supply to the board and a USB programmer is used to burn the hardware program (written in Arduino IDE) into the board.

Arduino Code

```
define enA 10//Enable1 L298 Pin enA#define in1 9 //Motor1 L298 Pin in1 #define in2 8 //Motor1 L298 Pin in1 #define in3 7 //Motor2 L298 Pin in1 #define in4 6 //Motor2 L298 Pin in1 #define enB 5 //Enable2 L298 Pin enB

#define L_S 12 //ir sensor Left #define R_S 13 //ir sensor Right

#define echo A2                       //Echo pin #define trigger A3 //Trigger pin

#define servo A5int Set=20;
int distance_L, distance_F, distance_R;

void setup(){ // put your setup code here, to run once Serial.begin(9600); // start serial communication at

9600bps
```

pinMode(R_S, INPUT); // declare if sensor as inputpinMode(L_S, INPUT); // declare ir sensor as input

pinMode(echo, INPUT);// declare ultrasonic sensor Echo pin as input pinMode(trigger, OUTPUT); // declare ultrasonic sensor Trigger pin as Output

pinMode(enA, OUTPUT); // declare as output for L298 Pin enApinMode(in1, OUTPUT); // declare as output for L298 Pin in1 pinMode(in2, OUTPUT); // declare as output for L298 Pin in2 pinMode(in3, OUTPUT); // declare as output for L298 Pin in3 pinMode(in4, OUTPUT); // declare as output for L298 Pin in4 pinMode(enB, OUTPUT); // declare as output for L298 Pin enB

//analogWrite(enA, 100); // Write The Duty Cycle 0 to 255 Enable Pin A for Motor1 Speed
//analogWrite(enB, 100); // Write The Duty Cycle 0 to 255 Enable Pin B for Motor2 SpeedpinMode(servo, OUTPUT);

```
for (int angle = 70; angle <= 140; angle += 5)     {servoPulse(servo, angle); }
for (int angle = 140; angle >= 0; angle -= 5)      {servoPulse(servo, angle); }

for (int angle = 0; angle <= 70; angle += 5)       {servoPulse(servo, angle); }

distance_F = Ultrasonic_read();delay(500);
}


void loop(){
//==============================================
//                Line Follower and Obstacle Avoiding
//==============================================

distance_F =  Ultrasonic_read(); Serial.print("D F=");Serial.println(distance_F);


//if Right Sensor and Left Sensor are at White color then it will call forword functionif((digitalRead(R_S) == 0) && (digitalRead(L_S) == 0) && (distance_F > Set)){
{forword();}

}
```

```
//if Right Sensor is White and Left Sensor is Black then it will call turn Left functionelse if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 1)){turnLeft();}

else if((digitalRead(R_S) == 0)&&(digitalRead(L_S) == 0) && (distance_F < Set))
{Stop();
Check_side();
//compareDistance();
}

else if((digitalRead(R_S) == 1)&&(digitalRead(L_S) == 1) ){Stop();}


delay(10);
}

void servoPulse (int pin, int angle){
int pwm = (angle*11) + 500;                    // Convert angle to microseconds digitalWrite(pin, HIGH);
delayMicroseconds(pwm);  digitalWrite(pin, LOW);delay(50); // Refresh cycle of servo
}


//*********************Ultrasonic_read***********************
**** long Ultrasonic_read(){ digitalWrite(trigger, LOW); delayMicroseconds(2); digitalWrite(trigger, HIGH); delayMicroseconds(10); long time = pulseIn (echo, HIGH);return time / 29 / 2;
}

void compareDistance(){if(distance_L > distance_R){ turnLeft();
delay(250);forword(); delay(500);turnRight();delay(300);forword(); delay(500);turnRight();delay(300);
forword(); delay(500);turnLeft(); delay(250);
```

```
}
else{ turnRight();delay(300);forword(); delay(500);turnLeft(); delay(250);forword(); delay(500);turnLeft();
delay(250);forword(); delay(500);turnRight();delay(300);
}
}
void Check_side(){Stop();
delay(100);
for (int angle = 70; angle <= 140; angle +=5) { servoPulse(servo,  angle);  } delay(300);
distance_R = Ultrasonic_read();Serial.print("D R=");Serial.println(distance_R); delay(100);
for (int angle = 140; angle >= 0; angle -=5) { servoPulse(servo, angle); } delay(500);
distance_L = Ultrasonic_read();Serial.print("D L=");Serial.println(distance_L); delay(100);
for (int angle = 0; angle <= 70; angle +=5) { servoPulse(servo, angle); } delay(300); compareDistance();
}

void forword(){ //forword analogWrite(enA, 100); digitalWrite(in2, LOW); //Left Motor backword Pin
digitalWrite(in1, HIGH); //Left Motor forword Pin digitalWrite(in4, HIGH);
//Right Motor forword Pin digitalWrite(in3, LOW); //Right Motor backword Pin analogWrite(enB, 100);
```

void backword(){ //backword digitalWrite(in1,HIGH); //Left Motor backword Pin digitalWrite(in2, LOW); //Left Motor forwordPin digitalWrite(in3, LOW); //Right Motor forword Pin digitalWrite(in4, HIGH); //Right Motor backword Pin
}

void turnRight(){ //turnRight analogWrite(enA,150); digitalWrite(in1, HIGH); //Left Motor backword Pin digitalWrite(in2, LOW); //Left Motor forword Pin digitalWrite(in3, HIGH);
//Right Motor forword Pin digitalWrite(in4,LOW); //Right Motor backword Pin analogWrite(enB, 100);
}

void turnLeft(){ //turnLeft analogWrite(enA, 100); digitalWrite(in1, LOW); //Left Motor backword Pin digitalWrite(in2, HIGH); //LeftMotor forword Pin digitalWrite(in3, LOW);
//Right Motor forword Pin digitalWrite(in4,HIGH); //Right Motor backword Pin analogWrite(enB, 150);
}

void Stop(){ //stop
digitalWrite(in1, LOW); //Left Motor backword Pin digitalWrite(in2, LOW); //Left Motor forword Pin digitalWrite(in3, LOW); //Right Motor forword Pin digitalWrite(in4, LOW); //Right Motor backword Pin
}

RESULT

Arduino-controlled robot car which moves around detecting obstacles in its way and avoiding them. During operation of the robot, the ultrasonic sensor sends out an ultrasoundwave to the front position (90 degrees), right position (36 degrees), and left position (144degrees). When the wave strikes an obstacle, it bounces back and the distance is stored forthe front, right, and left position. After this, the microcontroller compares the values basedon its algorithm and determines whether to move forward or change path.

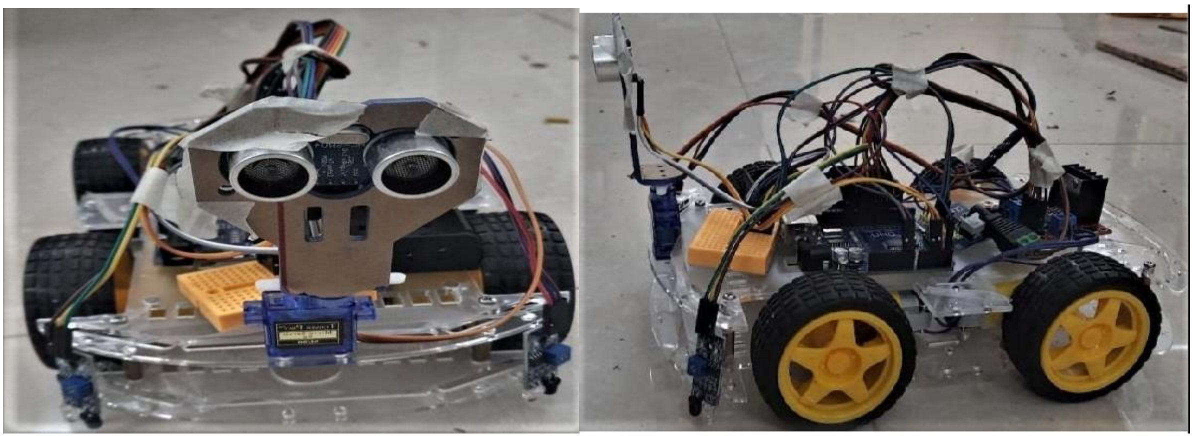


FIG Result

CONCLUSION

The proposed system gives the desired output through AI and ML. The scope of the proposedmethodology lies in achieving an efficient robot that completes the task effectively. Theuse of this robot improves workplace safety in a number of ways. For starters, they can accomplish tasks that humans might consider unsafe. Second, they run in a smooth, regulated manner, reducing the chances of human-operated vehicles damaging other workers. Furthermore, improved workplace safety can lower costs and save money by minimizing the amount of harm that human operators can cause. This technology is designed to be precise and accurate. Of course, robots can make mistakes, but when compared to human errors, the chances of making a mistake are quite low. In conclusion, obstacle avoidance path following robots play a crucial role in advancing automation across differentindustries. Their ability to navigate safely and effectively around obstacles leads to improvedsafety, efficiency, and productivity. As technology continues to evolve, these robots are likely to become even more sophisticated and integrated into various sectors, contributing tothe overall progress and innovation in robotics.

REFERENCES

[1] K.Hao,J.L.Zhao,B.B.Wang,Y.L.Liu,andC.Q.Wang,"Te application of an adaptive genetic algorithm based on colli- sion detection in path planning of mobile robots,"*Compu- tational Intelligence and Neuroscience*, vol. 2021, Article ID 5536574, 20 pages, 2021.

[2] Z. Wang, H. Ding, Z. Yang, B. Li, Z. Guan, and L. Bao, "Rank- driven salp swarm algorithm with orthogonal opposition- based learning for global optimization," *AppliedIn- telligence*, vol. 52, no. 7, pp. 7922–7964, 2022.

[3] L. Zhang, Y. J. Zhang, and Y. F. Li, "Mobile robot path planning based on improvedlocalized particle swarm opti- mization," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6962–6972, 2021.

[4] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation ofmobile robots in complex clut- tered environments: a survey," *Robotica*, vol. 33, no.3, pp. 463–497, 2015.

[5] Z.Q.Jian,S.Y.Zhang,S.T.Chen,Z.X.Nan,, "A global-local coupling two-stage path planning method for mobile robots," *IEEE Robotics and Automation Letters*, vol. 6,no. 3, pp. 5349–5356, 2021.

[6] W. Li, M. Tan, L. Wang, and Q. Z. Wang, "A cubic spline method combing improvedparticle swarm optimization for robot path planning in dynamic uncertain environment," *International Journal of Advanced Robotic Systems*, vol. 17, no. 1, Article ID 172988141989166, 2020.

[7] T. Hossain, H. Habibullah, R. Islam, and R. V. Padilla, "Local path planning for autonomous mobile robots by integrating modifed dynamic-window approach and improved follow the gap method," *Journal of Field Robotics*, vol. 39, no. 4, pp. 371–386, 2022.

[8] Q. W. Qian, J. F. Wu, and Z. Wang, "Optimal path planning for two-wheeled self-balancing vehicle pendulum robot based on quantum-behaved particle swarm optimization algo- rithm," *Personal and Ubiquitous Computing*, vol. 23,no. 3-4, pp.393–403, 2019.

[9] Z. A. Zhang, Y. X. Wan, Y. Wang, X. Q. Guan, W. Ren, and G. Li, "Improved hybridA∗ path planning method for spherical mobile robotbased on pendulum," *International Journal of Advanced Robotic Systems*, vol. 18, no. 1, Article ID 172988142199295, 2021.

[10] X.B.Cheng,L.C.Zhu,H.H.Lu,J.Z.Wei,andN.Wu,"Robot path planning based on an improved salp swarm algorithm," *Journal of Sensors*, vol.2022, Article ID 2559955,16 pages, 2022.

[11] L. W. Yang, L. X. Fu, P. Li, J. L. Mao, and N. Guo, "An efective dynamicpath planning approach for mobile robots based on ant colony fusion dynamic windows,"*Machines*, vol. 10, no. 1, p. 50, 2022.

[12] Y. Jin, M. Yue, W. J. Li, and J. Y. Shangguan, "An improved target- oriented path planning algorithm for wheeled mobile robots," *Proceedingsof the Institution of Mechanical Engineers - Part C: Journal of MechanicalEngineering Science*, vol. 236,no. 22, pp. 11081–11093, 2022.

[13] L. S. Liu, J. X. Yao, D. W. He et al., "Global dynamic path planningfusion algorithm combining jump-A∗ algorithm and dynamic window approach," *IEEE Access*, vol. 9,pp. 19632– 19638, 2021.

[14] J. F. Lian, W. T. Yu, K. Xiao, and W. R. Liu, "Cubic spline interpolation-based robotpath planning using a chaotic adaptive particle swarm optimization algorithm," *Mathe- matical Problems in Engineering*, vol. 2020, Article ID 1849240, 20 pages, 2020.

[15] B. Hichri, A. Gallala, F. Giovannini, and S. Kedziora, "Mobile robots path planning and mobile multirobots control: a re- view," *Robotica*, vol. 40, no.12, pp. 4257–4270,2022.