iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70761

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Obstacle Avoidance Robot

Shantanu Sinha[1], Swati Singh[2], Vikhyat Gupta[3], Dr. Abhinav Bhargava[4], Prof. Rahul Sharma[5]
[1, 2, 3, 4]Bachelor of Technology in Electronics and Communication Engineering, Lakshmi Narain College of Technology and Science, Bhopal
[5]Project Guide, [6]Project In-charge, Department of Electronics and Communication Engineering Lakshmi Narain College of Technology and Science, Bhopal

Abstract: Obstacle Avoiding Robot is an intelligent device that can automatically sense the obstacle in front of it and avoid them by turning itself in another direction. This design allows the robot to navigate in an unknown environment by avoiding collisions, which is a primary requirement for any autonomous mobile robot.
The application of the Obstacle Avoiding robot is not limited and it is used in most of the military organizations now which helps carry out many risky jobs that cannot be done by any soldiers. Here an Ultrasonic sensor is used to sense the obstacles in the path by calculating the distance between the robot and obstacle. If robot finds any obstacle it changes the direction and continue moving.
Obstacle Avoiding Robot is an intelligent device that can automatically sense the obstacle in front of it and avoid them by turning itself in another direction. This design allows the robot to navigate in an unknown environment by avoiding collisions, which is a primary requirement for any autonomous mobile robot. Obstacle avoiding robots can be used in almost all mobile robot navigation systems.
They can be used for household work like automatic vacuum cleaning. They can also be used in dangerous environments, where human penetration could be fatal.

I. INTRODUCTION

An obstacle avoidance robot is an autonomous mobile robot designed to detect and avoid obstacles in its path without human intervention. It uses sensors, such as ultrasonic, infrared, or LiDAR, to continuously scan the surrounding environment. When an obstacle is detected, the robot's microcontroller processes the sensor data and makes real-time decisions to change its direction or stop, ensuring smooth navigation. These robots are widely used in applications such as automated vacuum cleaners, autonomous vehicles, and robotics competitions. The primary objective of an obstacle avoidance robot is to move safely and efficiently within a defined space while preventing collisions, making it a fundamental project in robotics and embedded systems.

A. Background

Early Attempts at Measurement
It is an autonomous robot which will be able to avoid every obstacle in its path. It will use an ultrasonic distance sensor and a servo motor. The robot will check how far the nearest obstacle is (in every direction) and then decide upon the actions to be taken. The servo controls the direction in which the distance sensor faces and if the robot is hindered by an obstacle, the servo will rotate the sensor in different directions.
Once the robot is convinced that a certain direction is clear of any obstacles, it will turn the robot in that particular direction and then move in a straight line along that direction till the next obstacle is found. If there is no way to go ahead the robot executes a full 180° turn. Now day's many industries are using robots due to their high level of performance and reliability and which is a great help for human beings. The obstacle avoidance robotics is used for detecting obstacles and avoiding the collision. This is an autonomous robot.
The design of the obstacle avoidance robot requires the integration of many sensors according to their task. Obstacle detection is the primary requirement of this autonomous robot.
The robot gets the information from the surrounding area through mounted sensors on the robot. Some sensing devices used for obstacle detection like bump sensors, infrared sensors, ultrasonic sensors, etc. The ultrasonic sensor is most suitable for obstacle detection and it is of low cost and has a high ranging capability.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

1) Circuit Diagram

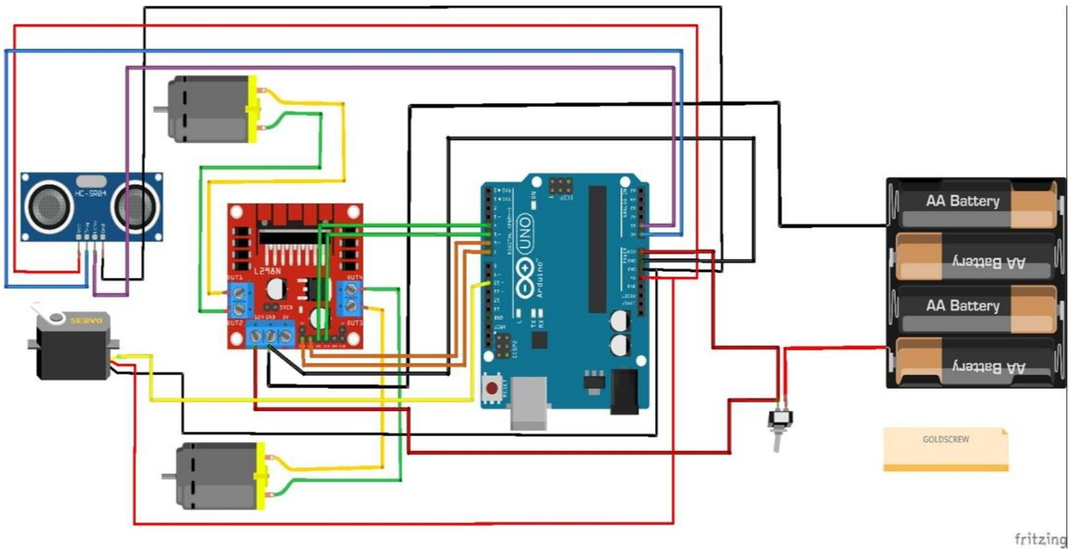


Fig. 1.1: Circuit Diagram of the Project

2) Approach

The development of an obstacle avoidance robot follows a systematic approach that encompasses the selection of components, circuit design, algorithm development, and testing, all aimed at creating a reliable and autonomous mobile system capable of identifying and navigating around obstacles. The first step in the approach involves defining the robot's functional requirements. For an obstacle avoidance robot, the core objective is to enable the robot to move autonomously in its environment while detecting and avoiding any physical obstructions in its path. To meet this objective, the design begins with selecting a suitable microcontroller, such as an Arduino Uno, due to its affordability, ease of programming, and a wide array of compatible sensors and modules. The Arduino serves as the brain of the robot, processing inputs from sensors and controlling the actuators based on pre-programmed logic.

Once the microcontroller is selected, the next phase involves choosing the appropriate sensors for obstacle detection. Ultrasonic sensors, particularly the HC-SR04, are commonly used due to their accuracy, low cost, and reliable performance in measuring distances by emitting sound waves and calculating the time taken for the echo to return. These sensors are typically mounted on the front of the robot, but additional sensors can be placed on the sides or back for comprehensive obstacle detection, depending on the complexity of the application. Infrared (IR) sensors may also be integrated to detect nearby objects or differentiate surface colors, especially in line-following scenarios. The placement of these sensors is critical and must be strategically designed to maximize the field of view and ensure accurate detection of obstacles from different angles.

After determining the sensors, attention is turned to the mechanical design and locomotion system of the robot. A two-wheeled differential drive robot with a free-rotating caster wheel is a common and effective configuration. It provides stability and maneuverability, allowing the robot to pivot in place and make sharp turns to avoid obstacles. The wheels are usually powered by DC motors or geared motors, controlled via a motor driver module such as the L298N. The motor driver acts as an interface between the low-power Arduino and the high-power motors, enabling the microcontroller to regulate speed and direction. Power supply management is another crucial consideration. A rechargeable battery pack is typically used to power both the Arduino and the motors. Voltage regulators may be included to ensure stable power delivery to sensitive components.

The software and control logic form the backbone of the obstacle avoidance functionality. Using the Arduino IDE, a program is written to continuously monitor sensor inputs, calculate distances to nearby objects, and make navigation decisions. The logic may follow a simple rule-based algorithm: if an obstacle is detected within a certain threshold distance (e.g., 20 cm), the robot stops, reverses briefly, and then turns either left or right depending on sensor readings. More advanced models may implement fuzzy logic or machine learning to improve decision-making in dynamic environments. The timing and responsiveness of the code are critical; delays must be minimized to ensure real-time reaction. Efficient coding practices, such as using non- blocking functions and implementing state machines, can enhance the robot's performance.

II. LITERATURE SURVEY

Amit Kumar Bharti, Arvind Kumar Bharati, Asharaf Raza, Ashwani Kumar, Aamir [1], in 2022, In today's world Robotics is a fast-growing and very interesting field. The concept of Robotics is now used in every sector whether it is in manufacturing industry, medical, transport etc. Obstacle avoidance is one of the features that is needed for the automated mobile robots. In this there is a robot that consist of Arduino UNO (Microcontroller) and sensor that detect presence of obstacles. Programming is done by the Arduino software. The ultrasonic sensor is highly accurate in detecting obstacles in the surroundings. This is a wheeled robot

Faiza Tabassum, Susmita Lopa, Muhammad Masud Tarek & Dr. Bilkis Jamal Ferdosi [2], in 2017, Obstacle detection and avoidance can be considered as the central issue in designing mobile robots. This technology provides the robots with senses which it can use to traverse in unfamiliar environments without damaging itself. In this paper an Obstacle Avoiding Robot is designed which can detect obstacles in its path and maneuver around them without making any collision. It is a robot vehicle that works on Arduino Microcontroller and employs three ultrasonic distance sensors to detect obstacles.

Pavithra A C, Subramanya Goutham [3] in 2018, The project is design to build an obstacle avoidance robotic vehicle using ultrasonic sensors for its movement. A microcontroller (ATmega328) is used to achieve the desired operation. A robot is a machine that can perform task automatically or with guidance. The project proposes robotic vehicle that has an intelligence built in it such that it directs itself whenever an obstacle comes in its path.

R Chinmayi; Yogesh Kumar Jayam; Venkatesh Tunuguntla; Jaideep Venkat Dammuru; [4] in 2018. This is a cost effective obstacle avoidance circuit developed in Amrita University. This robot is driven with an Arduino board controlled by an ultrasonic sensor. The obstacle is being detected at a distance of 15 cm when it senses the right path to move with the level of distance to the next obstacle. The number of vehicles is tremendously increasing day to day and the risk factor of accidents also increases with it. The Bluetooth module acts as an interface to communicate with the device using android apps.

III. TOOLS & TECHNOLOGIES

The development of an obstacle avoidance robot involves a combination of hardware tools and software technologies to ensure autonomous navigation and efficient performance. The primary hardware tools include an Arduino Uno microcontroller, which acts as the brain of the robot, ultrasonic sensors (such as HC-SR04) for detecting obstacles, and a motor driver module (like L298N) to control the DC motors. Additional components like jumper wires, a chassis, caster wheels, a battery pack, and a breadboard are essential for assembling the physical system.

Tools & Technologies used-

- Arduino Uno
- HC-SR04 Ultrasonic Sensor
- LM-298N motor driver module
- 5V DC motors
- Wheels & Chassis
- Battery & Jumper Wires
- Arduino IDE
- Tinkercad for virtual Implementation

A. Arduino Uno

In this project, the Arduino Uno plays a central role as the main microcontroller that coordinates all operations of the obstacle avoidance robot. It receives input from ultrasonic sensors, processes the distance data, and makes real-time decisions to control the movement of the robot. The Arduino Uno reads signals from the HC-SR04 ultrasonic sensor to detect nearby obstacles and sends corresponding commands to the motor driver (L298N) to change the direction of the DC motors. Its simplicity, low cost, and open-source environment make it ideal for beginners and hobbyists. Using the Arduino IDE, code is written in C/C++ and uploaded to the board via USB, allowing the robot to operate autonomously. The Arduino Uno's multiple digital and analog I/O pins make it easy to connect various components, while its stable performance ensures reliable control. Overall, the Arduino Uno serves as the brain of the system, enabling smart and efficient obstacle avoidance.

The versatility of the Arduino Uno also allows integration with wireless modules, such as the HC-05 Bluetooth module or NRF24L01 RF transceivers, to wirelessly transmit sensor data to a mobile phone or a remote receiver. While the Uno lacks built-in Wi-Fi (unlike the ESP8266 or ESP32), adding such modules extends its functionality into the realm of wireless communication and

Internet of Things (IoT). Although this adds complexity, it remains an excellent learning experience and demonstrates the flexibility of the Arduino Uno platform. Additionally, real-time clock (RTC) modules, such as the DS1307, can be added to timestamp each reading, which is particularly useful when data is stored for later analysis.

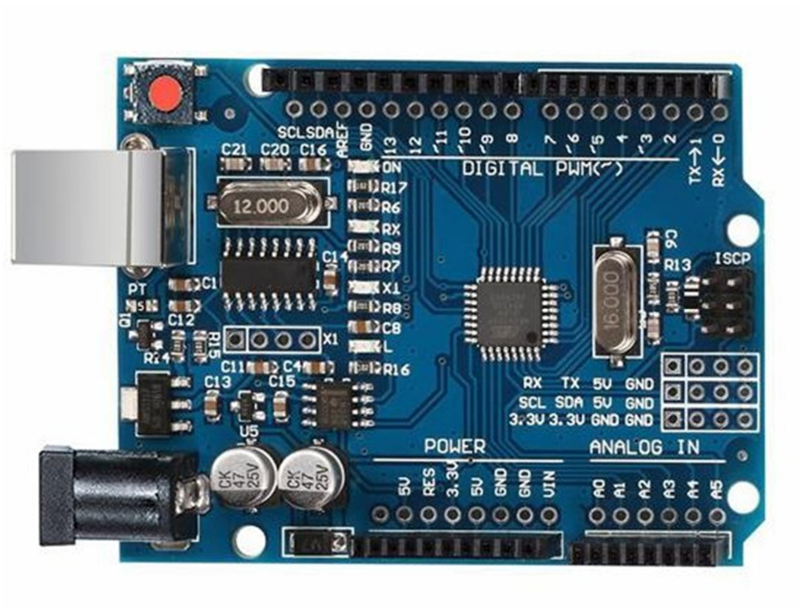


Fig. 3.1: Arduino Uno

B. HC-SR04 Ultrasonic Sensor

The HC-SR04 Ultrasonic Sensor is a key component in the obstacle avoidance robot, serving as the robot's "eyes" by enabling it to detect objects in its path and measure distances accurately. This sensor works on the principle of sound wave reflection. It emits ultrasonic sound waves at a frequency of 40 kHz from its transmitter, and when these waves hit an object, they bounce back to the sensor's receiver. The sensor then calculates the time taken for the echo to return and, using the speed of sound, determines the distance to the object. This time-of-flight measurement allows the robot to identify obstacles within a specific range, usually between 2 cm and 400 cm, with an accuracy of about 3 mm. In this project, the HC-SR04 is mounted on the front of the robot chassis and connected to the Arduino Uno, which is responsible for triggering the sensor and processing the echo response. The sensor has four pins: VCC, GND, Trigger (Trig), and Echo. The Trig pin is connected to a digital output pin on the Arduino and is used to send a 10-microsecond HIGH pulse to initiate the ultrasonic burst. The Echo pin is connected to a digital input pin and returns a pulse width that represents the time delay between sending and receiving the signal. The Arduino uses this pulse width to calculate distance using the formula: Distance = (Time × Speed of Sound) / 2. This division by two accounts for the round-trip of the wave. In the context of the robot, the sensor continuously monitors the environment in front of the robot by sending and receiving ultrasonic signals at regular intervals. The Arduino Uno is programmed to check whether the measured distance falls below a predefined threshold (for example, 20 cm). If an obstacle is detected within this range, the Arduino stops the motors, makes the robot reverse slightly, and then turns left or right, depending on the availability of space. This logic enables the robot to avoid collisions and navigate through complex spaces without external input. The simplicity and efficiency of the HC-SR04 make it a perfect match for such applications, especially in educational and hobbyist projects where low cost and ease of use are critical. Additionally, the sensor is reliable under normal lighting and environmental conditions and does not require calibration or complex setup.

Fig 3.2: HC-SR04 Ultrasonic Sensor

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

C. LM-298N motor driver module

The L298N motor driver module plays a crucial role in this obstacle avoidance robot project by acting as the interface between the Arduino Uno microcontroller and the robot's DC motors. The Arduino itself cannot supply enough current to drive motors directly, so the L298N module is used to control the power supplied to the motors safely and efficiently. It is based on the L298 dual H-bridge motor driver IC, which allows for independent control of the speed and direction of two DC motors simultaneously. In this project, the L298N receives control signals from the Arduino and uses an external power source, typically a battery pack, to drive the motors that move the robot. The module has inputs for IN1, IN2, IN3, and IN4, which are connected to digital pins on the Arduino to determine motor direction. It also includes ENA and ENB pins to enable or disable each motor, which are often connected to PWM (Pulse Width Modulation) pins on the Arduino to control motor speed.

The wiring of the L298N module involves connecting the motor terminals to the output pins (OUT1, OUT2, OUT3, OUT4), the power supply (usually 6V to 12V) to the module's VCC and GND terminals, and a 5V connection to power the logic circuit if needed. A jumper on the board often enables the onboard 5V regulator, which powers the logic side of the L298N. In operation, when the Arduino detects an obstacle via the ultrasonic sensor, it sends a signal to the L298N to stop or change the direction of the motors, allowing the robot to avoid the obstacle. For example, if an object is detected ahead, the Arduino will reverse the motors briefly and then rotate the robot by running one motor forward and the other in reverse.

The L298N is particularly advantageous because it allows precise control over each motor, enabling smooth movement, sharp turns, and the ability to stop instantly when needed. Its robustness, affordability, and ease of use make it a popular choice in robotic applications. Moreover, it protects the Arduino from high current loads and voltage spikes that can occur when motors start or stop abruptly. The dual H-bridge design also provides the flexibility to expand the robot's movement capabilities, such as adding a third motor or integrating with a servo for advanced motion. Overall, the L298N motor driver is essential in translating the microcontroller's low-power signals into real motor movements, making it a vital component of the robot's mobility system.

Fig. 3.3: L298N Motor

D. 5V DC Motor

In this obstacle avoidance robot project, the 5V DC motors are used as the primary actuators responsible for the movement of the robot. These motors convert electrical energy into mechanical energy, enabling the robot to move forward, reverse, or turn in different directions.

Typically, two 5V DC motors are attached to the rear wheels of the robot chassis, forming a differential drive system. This configuration allows the robot to be highly maneuverable—by varying the speed and direction of each motor independently, the robot can pivot, turn left or right, or move in a straight line.

The motors operate on a 5V power supply, which is compatible with the battery pack used in the project. Since the Arduino cannot drive these motors directly due to current limitations, the 5V DC motors are controlled through the L298N motor driver module. The driver receives signals from the Arduino based on the readings from the ultrasonic sensor, and it supplies the necessary voltage and current to the motors. For example, when no obstacle is detected, both motors are powered to move the robot forward. If an obstacle is detected ahead, the Arduino signals the driver to stop the motors, reverse them, or rotate the robot by controlling the direction of each motor separately.

Fig. 3.4: 5V DC Motor

IV. WORKING

The ultrasonic sensor emits the short and high-frequency signal. These propagate in the air at the velocity of sound. If they hit any object, then they reflect an echo signal to the sensor. The ultrasonic sensor consists of a multi-vibrator, fixed to the base. The multi-vibrator is a combination of a resonator and a vibrator. The resonator delivers ultrasonic wave generated by the vibration. The ultrasonic sensor consists of two parts; the emitter which produces a 40 kHz sound wave and the detector detects a 40 kHz sound wave and sends an electrical signal back to the microcontroller.

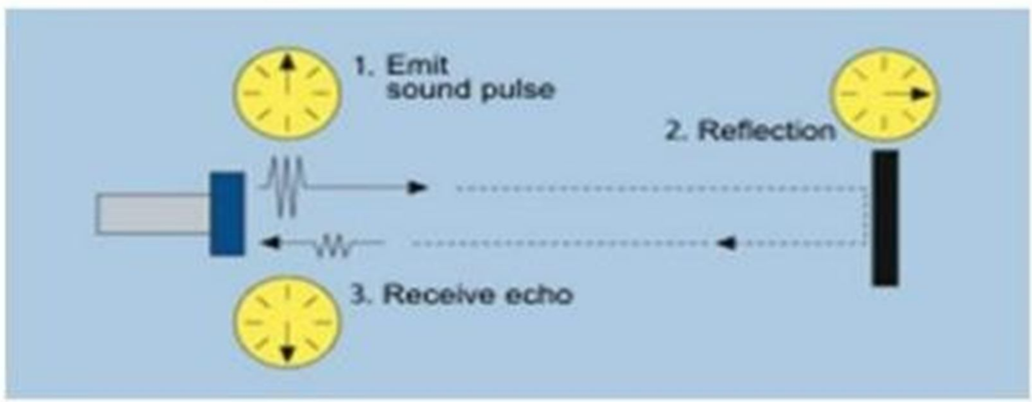


Fig. 4.1: Working of the Sensor

A. *Operation of the Ultrasonic Sensor*

When an electrical pulse of high voltage is applied to the ultrasonic transducer it vibrates across a specific spectrum of frequencies and generates a burst of sound waves. Whenever any obstacle comes ahead of the ultrasonic sensor the sound waves will reflect in the form of echo and generates an electric pulse. It calculates the time taken between sending sound waves and receiving the echo. The echo patterns will be compared with the patterns of sound waves to determine the detected signal's condition.

Note: The ultrasonic receiver shall detect signal from the ultrasonic transmitter while the transmit waves hit on the object. The combination of these two sensors will allow the robot to detect the object in its path. The ultrasonic sensor is attached in front of the robot and that sensor will also help the robot navigate through the hall of any building.

Applications of Ultrasonic Sensor:

1) Automatic change overs of traffic signals
2) Intruder alarm system
3) Counting instruments access switches parking meters
4) Back sonar of automobiles

A. *Features of Ultrasonic Sensor*

1) Compact & Lightweight
2) High Sensitivity & High Pressure
3) High Reliability
4) Power Consumption of 20mA
5) Pulse in-out Communication
6) Narrow Acceptance Angle

7) Provides exact, non-contact separation Estimation within 2cm-3m
8) The explosion point LED shows estimations in the advancement
9) 3-pin Header makes it simple to connect utilizing a servo development link.

B. Advantages

Obstacle avoiding robots can be used in almost all mobile robot navigation systems. **They can be used for household work like automatic vacuum cleaning**. They can also be used in dangerous environments, where human penetration could be fatal.

1) Very quick to compute.
2) Simple detection of edges and their orientation.
3) Improved Signal to noise ratio with a better detection with noise.
4) Quick response time.
5) Not affected by noise.
6) Detect object at distance.

C. Disadvantages

A disadvantage with obstacle avoidance based on edge detecting is the need of the robot to stop in front of an obstacle in order to provide a more accurate measurement.

1) Maybe Subject to false positives &false negatives.
2) Sensitive to noise.
3) Maybe inaccurate.
4) Cannot detect distant objects.
5) Produces false reading.

V. SOURCE CODE

```
//Object avoiding robot car code
//First install NewPing.h library #include <Servo.h>
#include <NewPing.h>

//L298N motor control pins
const int LeftMotorForward = 10; const int LeftMotorBackward = 11; const int RightMotorForward = 12; const int
RightMotorBackward = 13;

//Ultrasonic sensor pins #define trig_pin A1 #define echo_pin A2
#define maximum_distance 200 boolean goesForward = false; int distance = 100;

NewPing sonar(trig_pin, echo_pin, maximum_distance); //sensor function Servo servo_motor;

void setup(){

pinMode(RightMotorForward, OUTPUT); pinMode(LeftMotorForward, OUTPUT); pinMode(LeftMotorBackward, OUTPUT);
pinMode(RightMotorBackward, OUTPUT);
servo_motor.attach(9); //servo motor pin servo_motor.write(115);
delay(2000);
distance = readPing(); delay(100);
distance = readPing(); delay(100);
distance = readPing(); delay(100);
distance = readPing(); delay(100);
}
void loop(){
int distanceRight = 0; int distanceLeft = 0;
delay(50);
```

```
if (distance <= 40){ moveStop(); delay(300); moveBackward(); delay(400); moveStop(); delay(300);
distanceRight = lookRight(); delay(300);
distanceLeft = lookLeft(); delay(300);

if (distance >= distanceLeft){ turnRight();
moveStop();
}
else{ turnLeft(); moveStop();
}
}
else{ moveForward();
}
distance = readPing();
}
int lookRight(){ servo_motor.write(50); delay(500);
int distance = readPing(); delay(100); servo_motor.write(115); return distance;
}

int lookLeft(){ servo_motor.write(170);
delay(500);
int distance = readPing(); delay(100); servo_motor.write(115); return distance; delay(100);
}

int readPing(){ delay(70);
int cm = sonar.ping_cm(); if (cm==0){
cm=250;
}
return cm;
}

void moveStop(){

digitalWrite(RightMotorForward, LOW); digitalWrite(LeftMotorForward, LOW); digitalWrite(RightMotorBackward, LOW);
digitalWrite(LeftMotorBackward, LOW);
}

void moveForward(){ if(!goesForward){ goesForward=true;
digitalWrite(LeftMotorForward, HIGH); digitalWrite(RightMotorForward, HIGH);
digitalWrite(LeftMotorBackward, LOW); digitalWrite(RightMotorBackward, LOW);
}
}

void moveBackward(){ goesForward=false;
digitalWrite(LeftMotorBackward, HIGH); digitalWrite(RightMotorBackward, HIGH);
digitalWrite(LeftMotorForward, LOW); digitalWrite(RightMotorForward, LOW);

}

void turnRight(){
digitalWrite(LeftMotorForward, HIGH); digitalWrite(RightMotorBackward, HIGH);
```

```
digitalWrite(LeftMotorBackward, LOW); digitalWrite(RightMotorForward, LOW);
delay(500);
digitalWrite(LeftMotorForward, HIGH); digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW);
digitalWrite(RightMotorBackward, LOW);

}

void turnLeft(){

digitalWrite(LeftMotorBackward, HIGH); digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorForward, LOW); digitalWrite(RightMotorBackward, LOW);

delay(500);

digitalWrite(LeftMotorForward, HIGH); digitalWrite(RightMotorForward, HIGH);

digitalWrite(LeftMotorBackward, LOW); digitalWrite(RightMotorBackward, LOW);}
```

VI. RESULT

The result is obtained for obstacle avoidance robot using Arduino, if the robot moves forward if any obstacle detect it check for other directions and moves where there is no obstacles it moves in forward direction, to sense the obstacle ultrasonic sensor is used. We used servo motor to rotate the ultrasonic sensor.

The sonar system is used in HC-SR04 ultrasonic sensor to determine distance to an object like bats do. It offers excellent non-contact range detection from about 2 cm to 400 cm or 1feet to 13 feet. Its operation is not affected by sunlight or black material. The ultrasonic sensor emits the short and high frequency signal. If they detect any object, then they reflect back echo signal which is taken as input to the sensor through Echo pin .Firstly user initialize Trigger and Echo pin as low and push the robot in forward direction. When obstacle is detected Echo pin will give input as high to microcontroller. Pulse In function is used for calculating the time of distance from the obstacle. Every time the function waits for pin to go high and starts timing, then timing will be stopped when pin go to low. It returns the pulse length in microseconds or when complete pulse was not received within the timeout it returns. The timing has been determined means it gives length of the pulse and will show errors in shorter pulses. Pulses from 10microseconds to 3 minutes in length are taken into consideration. After determining the time, it converts into a distance. If the distance of object is moderate then speed of robot get reduced and will take left turn, If obstacle is present in left side then it will take right turn. If the distance of object is short then speed of robot get reduced and will turn in backward direction and then can go in left or right direction. This robot was built with an Arduino development board on which microcontroller is placed.

TABLE I. INPUT PINS FOR MOVEMENT

Movement	Pin10	Pin11	Pin 12	Pin 13
Forward	1	0	0	1
Backward	0	1	1	0
Left	1	0	1	0
Right	0	1	0	1

Fig. 6.1: IN/OP Pins

Arduino board is connected with DC Motor through Motor driver board (pin10, pin11, pin12, pin13) which provides power to the actuators. Actuators are used to move robot in Forward, Backward, Left and Right directions. The brief description of inputs pins for movement of robot is given in below in table. The movement of robot will be stop whenever there is an obstacle is present on its path which can be detected by ultrasonic sensors. Ultrasonic sensors give time in length to the microcontroller as an input for further actions

Photograph of the Project

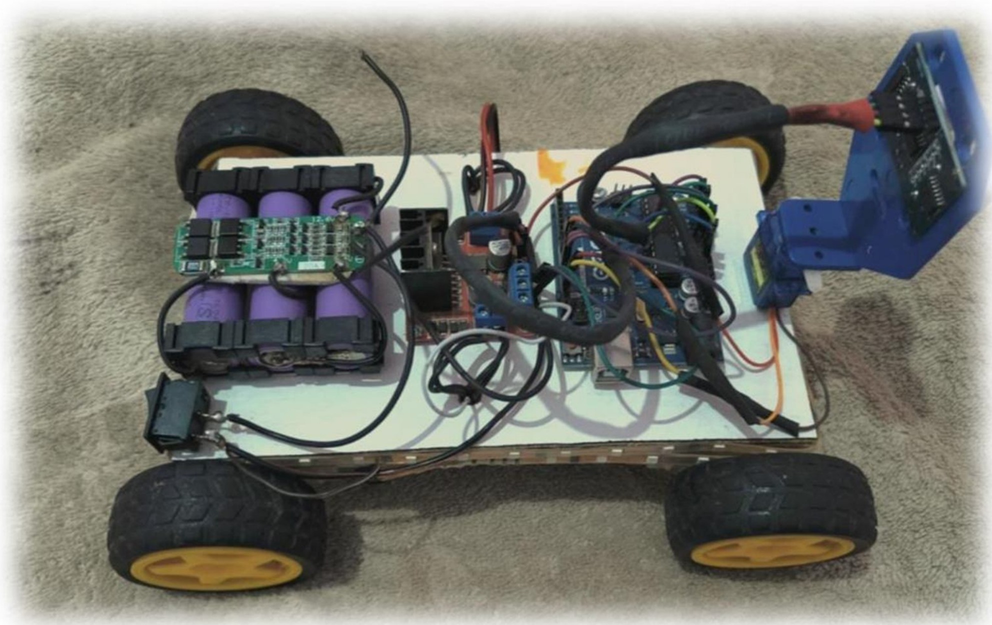


Fig. Obstacle Avoidance Robot

VII. EXPECTATION & ACHIEVEMENTS

A. *Expectations of the project*

At the beginning of the obstacle avoidance robot project, the core expectations centered around developing a simple yet functional robotic system capable of navigating autonomously while avoiding obstacles in its path. The primary goal was to understand and implement the integration of sensors, actuators, and microcontroller-based logic to achieve basic autonomous mobility. It was expected that the robot would detect obstacles using the ultrasonic sensor, process that information via the Arduino Uno, and respond by altering its movement to prevent collisions. The project aimed to demonstrate how real-time sensor feedback could influence physical responses, giving the robot the ability to operate independently in an environment without human intervention.

From a learning perspective, the expectation was to gain hands-on experience in programming embedded systems, wiring and interfacing electronic components, and designing a basic robotic system. Additionally, it was anticipated that this project would help develop problem-solving skills through practical experimentation, debugging, and iterative improvements. Another expectation was that the robot would work reliably in controlled indoor environments and serve as a foundation for more advanced projects in the future, such as line-following robots, maze solvers, or path-planning autonomous systems.

B. *Achievements of the Project*

The project successfully met several of its core goals. A working obstacle avoidance robot was built using key components such as the Arduino Uno, HC-SR04 ultrasonic sensor, L298N motor driver, and 5V DC motors. The robot was able to detect obstacles within a certain range (typically 15–20 cm) and respond appropriately by stopping, reversing, and turning to avoid collisions. This demonstrated that the robot could achieve autonomous navigation in a basic environment, fulfilling the primary functional requirement. One of the most significant achievements was the correct integration and synchronization of hardware and software. The ultrasonic sensor was accurately interfaced with the Arduino, and the distance-measuring code was written and calibrated successfully.

The logic implemented allowed real-time response to environmental changes, proving that the robot could interact with and respond to its surroundings. The L298N motor driver was also effectively used to control motor direction and speed, ensuring smooth and coordinated movements of the robot. From an educational standpoint, the project achieved its learning objectives. It enhanced the understanding of embedded systems, microcontroller programming, sensor applications, and actuator control. It also fostered practical skills in circuit design, soldering, troubleshooting, and using the Arduino IDE. The experience helped bridge the gap between theoretical knowledge and real-world application. Additionally, the robot's behavior was tested in various scenarios, such as approaching walls, navigating around objects, and moving through narrow paths. In each case, the system performed within expected parameters, proving its reliability under controlled conditions. The final prototype was compact, efficient, and could be operated with minimal supervision, showcasing a well- rounded implementation of basic robotics principles.

VIII. SHORTCOMING & LIMITATIONS

Although the obstacle avoidance robot demonstrates basic autonomous navigation and obstacle detection, it still has several notable shortcomings and limitations that affect its overall efficiency, scalability, and real-world applicability. These limitations span across its hardware components, software design, sensing capabilities, power management, and environmental adaptability. Understanding these constraints is essential for evaluating the robot's performance and identifying areas for future improvement.

A. Limited Field of View

One of the most significant limitations of the current design is the narrow field of view provided by a single forward-facing ultrasonic sensor. The HC-SR04 ultrasonic sensor can only detect obstacles directly in front of the robot, usually within a 15° to 30° cone of vision. This means obstacles on the sides or behind the robot may go undetected, potentially leading to collisions during turns or while navigating tight spaces. In real-world environments, where obstacles can appear from various directions, such a limited detection range is insufficient for robust navigation. This limitation makes the robot less suitable for complex tasks such as exploring unknown areas or navigating crowded indoor spaces.

B. Basic Decision-Making Logic

The robot operates based on simple rule-based logic coded into the Arduino Uno. The program typically follows a basic pattern: move forward, stop when an obstacle is detected, reverse briefly, and turn in a predefined direction. While this approach is easy to implement and sufficient for controlled environments, it lacks adaptability and intelligence. The robot does not assess the optimal direction to turn based on multiple sensor inputs, nor does it evaluate previous decisions to improve future movements. Without algorithms like path planning, obstacle mapping, or machine learning, the robot cannot optimize its route or adapt to dynamic changes in the environment.

C. No Environmental Mapping or Memory

Another limitation is the robot’s inability to create or store a map of its surroundings. Since it does not use technologies like Simultaneous Localization and Mapping (SLAM) or store previous positions, the robot cannot remember where it has been or where obstacles are located beyond the immediate sensor range. This restricts the robot to reactive behavior rather than proactive navigation. It simply responds to obstacles as they are detected without understanding the layout of the environment. As a result, the robot may revisit the same locations or get trapped in loops, which reduces its effectiveness in more complex applications.

D. Sensor Inaccuracy and Environmental Sensitivity

The HC-SR04 ultrasonic sensor, while affordable and easy to use, can be affected by environmental factors. For instance, the sensor may struggle to detect very soft, curved, or narrow objects that do not reflect sound waves effectively. Surfaces that absorb or deflect sound, such as cloth or angled glass, may yield inaccurate distance readings or go undetected altogether. Additionally, ambient noise, temperature, and humidity can impact the speed of sound and affect distance calculations. These inaccuracies can lead to delayed responses or incorrect obstacle detection, causing inefficient or unsafe navigation.

E. Lack of Side and Rear Obstacle Detection

In many real-world scenarios, obstacles may approach the robot from the sides or rear. However, the current design only monitors the front direction, leaving blind spots in other directions. This limitation is particularly problematic when the robot reverses or turns, as it may hit objects outside the sensor’s detection range. Without additional sensors mounted on the sides or rear, the robot lacks the comprehensive awareness needed for safe and reliable movement in cluttered or unpredictable environments.

F. No Feedback Control or Position Tracking

The robot does not incorporate feedback mechanisms such as wheel encoders, gyroscopes, or accelerometers. As a result, it lacks knowledge of its own speed, orientation, and displacement. Movement is controlled in open-loop mode, meaning that once a command is sent to the motors, the system assumes the robot moves as expected, without confirming it. This can lead to errors in positioning, especially on uneven terrain or low-friction surfaces where wheel slip occurs. Precise navigation or path-following tasks become challenging or impossible without feedback for correction.

G. Limited Power Management

The robot is powered by a standard battery pack, which typically supplies 6V to 12V to the motor driver and the Arduino. However, without efficient power management or monitoring, battery performance can degrade over time. As the battery voltage drops, motor speed and sensor performance can decline, leading to inconsistent behavior. Furthermore, the project does not include features such as battery level monitoring or power-saving modes.

H. Limited Scalability and Modularity

While the current system is simple and easy to build, it is not very scalable. The Arduino Uno has a limited number of GPIO pins and memory, which restricts the number of additional sensors, actuators, or modules that can be connected. Expanding the robot to include more sophisticated sensors like LiDAR, cameras, or GPS modules would require upgrading to a more powerful microcontroller or microprocessor platform. Similarly, without a modular design approach, integrating new features may involve rewiring or redesigning the entire system, which limits the flexibility for future enhancements.

IX. ISSUES & CHALLENGES

Although the obstacle avoidance robot was built successfully and demonstrated core functionalities, several issues and challenges were encountered during its design, assembly, and testing phases. These challenges ranged from hardware-related problems to software limitations, as well as issues in integrating the various components effectively. Some of the key difficulties encountered are discussed below:

A. Sensor Accuracy and Reliability

One of the major challenges was ensuring the accuracy and reliability of the HC-SR04 ultrasonic sensor. The sensor relies on sound waves to measure distance, but its accuracy can be affected by several environmental factors. For instance, soft or irregularly shaped objects, such as cloth or angled surfaces, do not reflect sound waves as efficiently as flat, hard surfaces. This led to inaccurate distance measurements or the robot failing to detect obstacles in certain scenarios. Additionally, ambient noise and temperature fluctuations also influenced the sensor's performance, affecting its ability to consistently measure distances. The challenge here was calibrating the sensor to handle a variety of objects and environmental conditions while maintaining reliable obstacle detection.

B. Limited Field of View

The single ultrasonic sensor mounted on the front of the robot was another limitation that posed challenges. The sensor's narrow detection angle, typically around 15-30 degrees, only provided limited coverage of the environment. This meant that obstacles approaching from the sides or rear of the robot could not be detected. As the robot navigated, it often found itself unable to avoid obstacles unless they were directly in its forward path. This issue was particularly challenging when the robot was attempting to turn or reverse, as it had no real- time awareness of obstacles outside its direct line of sight. To address this limitation, additional sensors could be added to the sides or rear of the robot, but the additional complexity and wiring posed further challenges.

C. Motor Control and Precision

The use of DC motors to drive the robot posed its own set of challenges, primarily related to motor control and precision. While the L298N motor driver was effective in controlling the motors' direction, achieving precise control over the robot's movement proved difficult. The robot's motion was not always smooth, especially when it transitioned between forward and reverse directions. This often resulted in erratic movements or difficulty in making sharp turns. Additionally, there was no feedback mechanism in place, such as wheel encoders or PID (Proportional-Integral-Derivative) control, to track the robot's position or speed accurately. The absence of precise feedback meant that the robot had difficulty maintaining consistent speed or direction, which was particularly noticeable on uneven or slippery surfaces.

D. Power Supply and Battery Life

The power supply was another issue that posed challenges during the project. The robot was powered by a basic battery pack, but the voltage and current requirements of the DC motors often resulted in power depletion more quickly than expected. As the battery drained, the motors became less responsive, and the sensor readings became less accurate. The lack of a power management system meant that the robot could lose its functionality mid-operation due to insufficient power. In addition, since the motors and the Arduino were sharing the same power source, voltage drops occurred, affecting the robot's overall performance. The challenge was to ensure that the robot had enough battery life to complete its tasks without sudden shutdowns, which required optimizing power usage and considering battery capacity for longer operational times.

E. Code Optimization and Logic Flaws

While the Arduino code for the robot was functional, it was often not optimal in terms of performance and flexibility. The robot's decision-making logic was simple and reactive, which meant it only responded to immediate obstacles. There were instances where the robot would get stuck in a loop, making inefficient movements or retracing its steps unnecessarily. For example, after avoiding an obstacle, the robot might have turned into another obstacle or failed to move in the most efficient direction. This was largely due to the lack of advanced algorithms for decision-making, such as pathfinding algorithms (e.g., A* or Dijkstra's) or state-based logic to guide the robot in a more intelligent manner. Additionally, the absence of error handling in the code sometimes caused the robot to behave unpredictably when encountering situations that the logic was not designed to handle. For instance, if an object was too close or the sensor readings fluctuated, the robot could become unresponsive or perform actions that did not make sense (e.g., turning in the wrong direction).

F. Integration of Components

The integration of the various components—Arduino, HC-SR04 sensor, L298N motor driver, and 5V DC motors—was a complex task that presented numerous challenges. One issue was ensuring that the wiring and connections were secure and correctly routed to avoid short circuits or power issues. The L298N motor driver required careful attention to ensure that the motor connections and the power supply were set up correctly, as improper wiring could result in the motors not working or the Arduino failing to receive proper input signals. Additionally, the communication between the components often faced issues, such as signal interference between the motor driver and the sensor, causing instability in motor control. Testing the robot in real-time required debugging multiple connections and adjusting the code to ensure that the motors responded appropriately to sensor input.

G. Limited Debugging and Testing Environment

Another challenge faced during the project was the testing environment. The robot was initially tested on flat, well-defined surfaces, which did not fully replicate the challenges it would face in real-world environments. When the robot was tested on uneven or sloped terrain, its performance significantly degraded. Additionally, testing the robot in environments with obstacles of different shapes, sizes, and materials highlighted the sensor's inability to detect all types of obstacles accurately. The lack of a sufficient debugging process also meant that troubleshooting took longer than expected. While the robot performed well in controlled tests, real-world conditions uncovered numerous edge cases that required additional adjustments to the code and hardware setup. Debugging these issues was time-consuming, as it involved testing multiple scenarios and fine-tuning sensor calibration, motor behavior, and decision-making logic.

H. Lack of Advanced Features

Finally, the absence of advanced features like real-time communication, remote control, and environment mapping meant that the robot was not fully autonomous in dynamic environments. Features like path planning, SLAM (Simultaneous Localization and Mapping), or the ability to recognize and avoid multiple types of obstacles would have made the robot significantly more advanced. However, implementing these features required more complex hardware and software, which were beyond the scope of this initial project.

X. FUTURE SCOPE

The obstacle avoidance robot project has considerable potential for expansion and improvement. While the initial implementation serves as a functional and educational starting point, there are many opportunities to enhance the system's capabilities and adapt it to more complex environments and tasks. By incorporating advanced technologies, algorithms, and hardware improvements, the project could evolve into a more intelligent, versatile, and autonomous robot. Below are several potential areas for future development and expansion:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

A. Enhanced Sensor Integration and Coverage

One of the major limitations of the current project is the use of a single ultrasonic sensor, which only provides a narrow field of view. To improve the robot's ability to navigate in dynamic environments, additional sensors could be added:

Multiple Ultrasonic Sensors: Adding sensors to the sides and rear of the robot could increase its awareness of obstacles in all directions, reducing blind spots and preventing collisions from multiple angles.

Infrared Sensors: These sensors can be added alongside ultrasonic sensors to improve obstacle detection in situations where ultrasonic waves may be ineffective, such as in very close proximity or in certain materials that absorb sound.

LiDAR (Light Detection and Ranging): For more accurate and detailed environmental mapping, LiDAR sensors can be added. LiDAR can scan a larger area around the robot, providing precise distance measurements and creating a more accurate map of the environment. Cameras and Computer Vision: Incorporating cameras and implementing computer vision algorithms (e.g., OpenCV) would allow the robot to not only detect obstacles but also recognize objects, track movements, and follow specific paths or targets.

These additional sensors would enable more sophisticated behavior, including detecting and avoiding objects in complex and cluttered environments.

B. Path Planning and Navigation Algorithms

The robot currently follows simple reactive behaviors, responding to obstacles with predefined movements (e.g., stop, reverse, turn). However, more advanced robots use path planning algorithms to optimize their movement, allowing them to navigate efficiently through complex spaces. A few improvements in this area could include:

SLAM (Simultaneous Localization and Mapping): SLAM allows the robot to build a map of its environment while simultaneously tracking its position within that map. Implementing SLAM would make the robot capable of navigating in unknown spaces, learning about its environment, and avoiding previously encountered obstacles.

Dijkstra's Algorithm: Another common pathfinding algorithm, Dijkstra's algorithm could be used to find the shortest path in a graph-like structure, allowing for more efficient movement in environments with multiple obstacles.

By incorporating these algorithms, the robot could transition from reactive to proactive behavior, improving its ability to plan and follow optimal paths in real-time.

C. Improved Motor Control and Feedback

In the current project, the robot operates with simple motor control via the L298N motor driver. However, more precise control is possible through the use of feedback systems. By adding feedback mechanisms, the robot's movements could become smoother and more accurate, particularly in challenging environments:

Encoders: Adding encoders to the wheels would allow the robot to track its precise position and movement. This would help correct any errors in movement caused by wheel slippage, uneven terrain, or motor variations.

PID Control: Implementing a PID controller (Proportional-Integral-Derivative) could optimize the speed and direction of the motors, reducing overshoot and improving response times, especially in dynamic environments.

These improvements would lead to more stable and reliable motion, especially in environments with obstacles or irregular surfaces.

D. Power Efficiency and Battery Management

As mentioned earlier, the robot's performance is affected by power limitations, and its battery life is a significant constraint. To improve the robot's performance and extend its operational time, several power-related upgrades could be implemented:

Battery Management System (BMS): Adding a BMS would allow the robot to monitor its battery voltage and consumption in real-time, providing better power management and preventing sudden shutdowns due to low battery.

Solar Charging: For outdoor robots, integrating solar panels could allow the robot to recharge itself while it is operating, extending its operational time and reducing the need for manual recharging.

More Efficient Power Supply: Using a higher-capacity battery or a more efficient voltage regulator could ensure that the motors, sensors, and microcontroller receive the appropriate power without draining the battery too quickly.

E. Communication and Remote Control

The current robot operates autonomously without any external control. However, for more complex tasks or in real-world applications, enabling communication with the robot can provide several benefits:

Wireless Control: Integrating a wireless communication module, such as Bluetooth or Wi-Fi, would allow the robot to be remotely controlled via a smartphone, computer, or other devices. This would also enable remote troubleshooting or control in case the robot faces an issue it cannot overcome autonomously.
Real-Time Monitoring: Adding real-time data streaming, such as sending sensor readings or positional information to a remote device, would provide users with insights into the robot's status and allow them to monitor its behavior during operation.
Voice Control: By integrating speech recognition technology, the robot could respond to verbal commands, making it more accessible and user-friendly in specific applications.

XI. ACKNOWLEDGEMENT

Words shall never be able to pierce through the Gamut's of emotions that are Suddenly exposed during the routine of our college life. We would like to thank, Dr. Abhinav Bhargava, Project Guide for guiding and correcting us at each instant and for his continuous motivation. At the same time, we would also like to thank Prof. Rahul Sharma, Project In-charge of Electronics and Communication Engineering, for his well-wishes and to provide her kind support and valuable knowledge on the subject from time to time.
We express our heart-felt thankfulness to Dr. Abhinav Bharagava, H.O.D. of Electronics and Communication Engineering, LNCTS, Bhopal for his unfailure inspiration, whole hearted co-operation and constant encouragement throughout the progress of the dissertation and incisive comments gave immense confidence to complete the work. Also, we are grateful to Dr. Ashok Kumar Rai, Director Administration, LNCT Group for providing us a positive learning environment and Dr. V. N. Bartaria Principal, LNCTS for giving us a chance to fulfill our potential.

REFERENCES

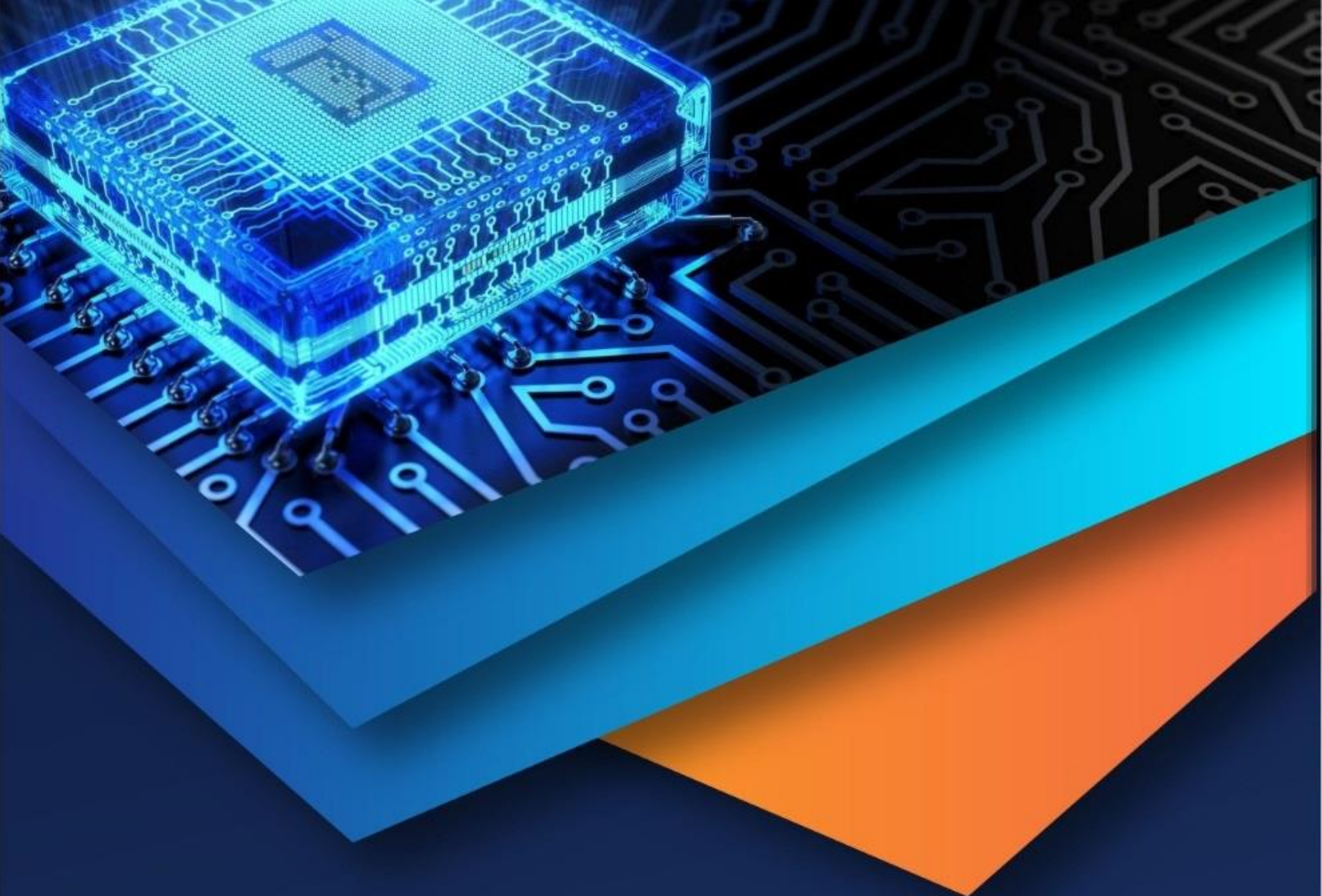
[1] Amit Kumar Bharti, Arvind Kumar Bharati, Asharaf Raza, Ashwani Kumar, Aamir https://doi.org/10.22214/ijraset.2022.43056, May 2022.
[2] Faiza Tabassum, Susmita Lopa, Muhammad Masud Tarek & Dr. Bilkis Jamal Ferdosi , in 2017 Pavithra A C, Subramanya Goutham in 2018
[3] R Chinmayi; Yogesh Kumar Jayam; Venkatesh Tunuguntla; Jaideep Venkat Dammuru; in 2018

About Us

My name is Shantanu Sinha. At present I am pursuing my Bachelor of Technology in Electronics and Communication in Lakshmi Narain College of Technology & Science, Bhopal. My current CGPA is 6.70. I have done my schooling from Saraswati Vidya Mandir Munger Bihar. I am an active learner and Iam a hard worker.

My name is Swati Singh. At present I am pursuing my Bachelor of Technology in Electronics and communication in Lakshmi Narain College of Technology & Science, Bhopal. My current CGPA is 7.4. I have done my schooling from D.A.V. Public School Pandopara Chhattisgarh. I’m an active learner and I’m a hard worker.

My name is Vikhyat Gupta. At present I am pursuing my Bachelor of Technology in Electronics and communication in Lakshmi Narain College of Technology & Science, Bhopal. My current CGPA is 7.1. I have done my schooling from St. Paul EL Higher Secondary School Gwalior. I’m an active learner and I’m a hard worker.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)