A Project Report

on

Collision Avoiding System

1. Problem Statement:

In many autonomous and semi-autonomous systems, the risk of collision with nearby obstacles poses a serious safety concern. Accidents can occur due to the lack of real-time obstacle detection and delayed response mechanisms, especially in robotics, automated vehicles. There is a need for a compact, cost-effective, and reliable solution that can detect obstacles in real-time and take immediate action to prevent collisions. This project aims to develop a sensor-based Collision Avoiding System that uses an ultrasonic sensor, Hedy MCU, and servo motor to detect nearby objects and automatically trigger alerts and braking actions to enhance safety and prevent damage.

2. Abstract:

Collisions in automobiles have become a major safety concern; cases of damage and death due to automobile collisions are frequently reported. The number of pedestrians knocked to death by automobiles or vehicles crashing into each other is also on the rise in cities and highways. The cost of a life cannot be estimated, the cost of damages to the automobiles also impacts negatively on investment. In most cases drivers fail to notice the presence of obstacles ahead and brakes need a driver's response to operate thus increasing the response time, hence reducing their reliability. Many approaches to automobile crash avoidance systems have been proposed in recent time, however, such approaches mainly concentrate on steering maneuvering control. In addition, these approaches do not take into consideration the safety distance to stoppage of the automobile, more so, many approaches only provide a warning signal to the driver for activating the automatic braking. This always provides a room for human error.

This project presents the development of an automatic microcontroller based crash avoidance system that employs obstacle detection and distance measurement using ultrasonic sensors to detect obstacles and their distances. Once the obstacle has been detected and safe separation distance is reached, the automobile performs safety distance induced braking at that distance to bring the automobile to a stoppage. From the results obtained after tests, the system activates the buzzer and LED at a distance of 150mm and brakes at a distance 70mm of with respect to the obstacle.

Keywords: Collision Avoidance System, Object Detection, Ultrasonic Sensors, Hedy MCU

3. Introduction :

The problem of vehicle accident is a part of an endless list of disaster that could occur anywhere anytime. About 1.2 million people die on the world's roads annually, making it a leading cause of death globally. Collision avoidance system is a system made of sensors placed inside a car to caution its driver of any obstruction that could possibly lie ahead on the road. Some of the dangers detected by these sensors include the proximity of the car to other cars around it, and its exact distance from the cars. The ultrasonic sensor is modified to measure the proximity with reference to the previous car for the rear end. According to the closeness of cars suitable cautioning signals and counter measures are taken. Approaches to automobile crash avoidance systems have been based on controlling the steering maneuvering, these approaches do not take into consideration the safety distance to stoppage of the automobile. Thus provides a room for human error. In this project an automatic micro controller(Hedy MCU) based crash avoidance system is developed that employ obstacle detection and distance measurement using ultrasonic sensors to detect obstacles and their distances for automatic safety braking of the car without the driver's consent thus preventing collision.

4. PCB Designing :

The design of a Printed Circuit Board (PCB) is a fundamental phase in the development of any electronic system. It involves translating an abstract electrical schematic into a physical representation that can be manufactured and assembled to create a functioning electronic circuit. A well-designed PCB ensures not only the correct electrical interconnections among components but also promotes signal integrity, mechanical stability, and thermal management.

4.1 Steps of PCB Designing:

The process of designing a PCB involves several essential stages that transform an electronic schematic into a functional, manufacturable hardware layout. The steps involved are:

1. Conceptualization and Schematic Design:

   The PCB design process begins with the creation of a schematic diagram, which serves as the blueprint of the circuit. This diagram outlines the logical connections between various electronic components, such as resistors, capacitors, microcontrollers, and sensors. The schematic is developed using Computer-Aided Design (CAD) software tools like KiCad, Eagle, or Altium Designer. At this stage, it is crucial to ensure that all components are correctly placed and interconnected based on the design requirements.

2. Component Footprint Assignment

   Each component in the schematic is with a specific footprint—a graphical representation of the physical layout and pin configuration. Accurate footprint selection is critical to ensure compatibility between the electronic components and the board layout during the fabrication and assembly processes.

3. Layout Design

   Following schematic creation, the layout design phase commences. Here, the components are arranged on a virtual board canvas according to electrical, mechanical, and thermal considerations. The goal is to minimize the length of critical signal paths, reduce electromagnetic interference, and ensure ease of assembly. The placement of components plays a significant role in the board's performance, especially in high-frequency or high-power applications.

4. Routing of Electrical Connections

Routing involves drawing copper traces that form the actual electrical connections between components. Designers must consider factors such as trace width, spacing, and impedance. In multilayer PCBs, different layers may be used for power, ground, and signal routing, enhancing signal integrity and simplifying layout complexity.

5. Design Rule Checks (DRC)

Once routing is completed, Design Rule Checks (DRC) are performed to verify that the PCB layout adheres to the manufacturing constraints and electrical design rules. These checks help identify issues such as overlapping traces, incorrect clearances, and missing connections, which could otherwise lead to functional failures or production defects.

6. Generation of Manufacturing Files

After successful validation, the design is exported into a set of standardized files known as Gerber files. These files include information about each layer of the PCB, including copper layers, solder mask, silkscreen, and drilling instructions. Gerber files are submitted to PCB manufacturers for fabrication.

7. Review and Finalization

Before sending the design for fabrication, a final review is conducted. This may include 3D visualization, thermal simulations, and electrical validation. Upon satisfactory completion, the board is ready for manufacturing and subsequent assembly.

4.2 Types of PCB

Printed Circuit Boards (PCBs) are categorized based on the number of conductive layers they contain. These classifications include single-sided, double-sided, and multi-layer PCBs. Each type serves different design complexities and application requirements.

1. Single-Sided PCB

A Single-Sided PCB as shown in Figure 1 is the most basic form of printed circuit board, consisting of a single layer of conducting material (usually copper) on one side of an insulating substrate. the other side remains unused for circuitry**.**

In this configuration, all the electrical components and their respective interconnections are placed and routed on the same side. Due to its simplicity, single-sided PCBs are commonly used in low-cost and low-complexity applications where minimal routing is required.

They are widely used in consumer electronics, LED lighting systems, and power supply modules. However, the limited space for routing restricts their application in more complex circuit designs.

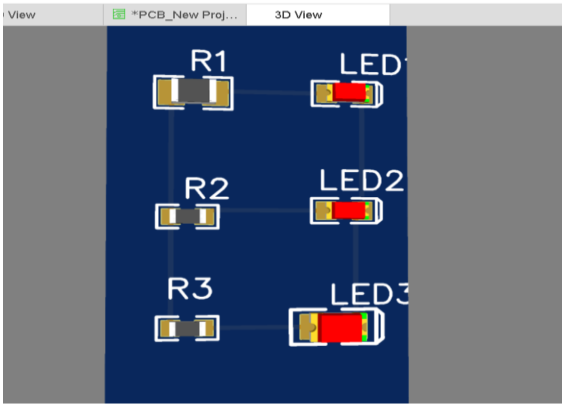

Figure 1. Single Sided PCB

2. Double-Sided PCB

A double-sided PCB as shown in Figure 2 contains two conductive layers—one on the top side and one on the bottom—separated by an insulating core. Components may be mounted on both sides of the board, and electrical interconnections between the layers are established using plated-through holes known as vias**.**

Double-sided PCBs offer significantly more design flexibility than single-sided boards, allowing for more complex and dense circuitry. They enable the crossing of signal paths without the need for jumpers, thereby making efficient use of board space. This type of PCB is suitable for a wide range of applications such as industrial control systems, instrumentation, automotive circuits, and consumer electronics where moderate circuit complexity is required.

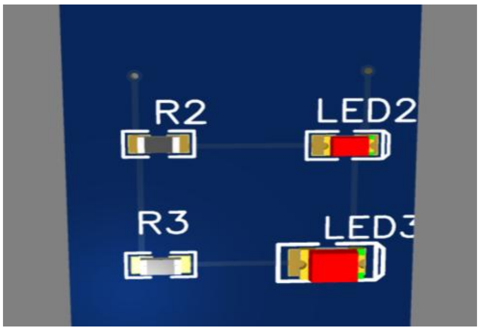

Figure 2.a Top side of Double Sided PCB

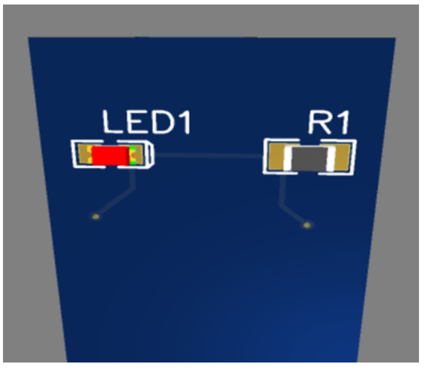

Figure 2.b Bottom side of Double Sided PCB

3. Custom PCB

Custom Printed Circuit Boards (PCBs) as shown in Figure 3 are designed specifically to meet the unique requirements of a particular application. Unlike standard off-the-shelf PCBs, which are mass-produced to fit common needs, custom PCBs are tailored in terms of size, shape, layer count, material, and even component placement to suit specialized use cases.

The key feature of custom PCBs is their flexibility. Designers can specify the board's dimensions, layer count (single-sided, double-sided, or multi-layer), trace width, and component configuration to meet the precise needs of the application. This customization allows for efficient use of space, ensuring that the PCB fits into specific devices or enclosures.

Figure 3. Custom House Shaped PCB

5. Introduction to Hedy MCU:

An MCU (Microcontroller Unit) is a compact integrated circuit that contains a processor core, memory, and programmable input/output peripherals. Microcontrollers are the brains of embedded systems and are commonly used in various electronic devices, from consumer electronics to industrial machines.

The Hedy MCU series, similar to other MCUs like Arduino or ESP32, is designed to provide a simple yet powerful platform for developers, hobbyists, and engineers to design embedded systems. These microcontrollers are typically used in applications like automation, robotics, sensor networks, and IoT (Internet of Things).

Here are the key features of the Hedy MCU**:**

1. Processor Core**:** Typically based on 32-bit or 8-bit processors (e.g., ARM Cortex-M), offering sufficient processing power for embedded systems applications.
2. Memory**:** Equipped with flash memory for program storage and RAM for temporary data storage, with varying sizes depending on the model.
3. Peripherals**:** Includes essential GPIO pins**,** Analog-to-Digital Converters (ADC)**,** Pulse Width Modulation (PWM) outputs, UART**,** SPI, and I2C interfaces for sensor and actuator connections.
4. Low Power Consumption**:** Designed for energy efficiency, making them ideal for battery-powered or portable applications, with optimized power management.
5. Connectivity**:** Supports wireless communication technologies such as Wi-Fi**,** Bluetooth**,** and Zigbee for IoT and remote applications. Some models also support Ethernet for long-range communication.

6. Methodology

This project uses a sensor-based method to build a real-time collision avoiding system that improves safety in autonomous movement. The system uses an ultrasonic sensor to detect nearby objects, a microcontroller to process the data and make decisions, and components like LEDs, buzzers, and a servo motor to give warnings and apply brakes when needed.

6.1 Block Diagram of Collision Avoiding System:

The system is developed step by step, starting with connecting sensors, programming the microcontroller, and controlling the actuators. The main goal is to sense obstacles early and respond quickly to avoid crashes by alerting users and triggering the brake. The block diagram of the Collision Avoiding System is shown in Figure 4.

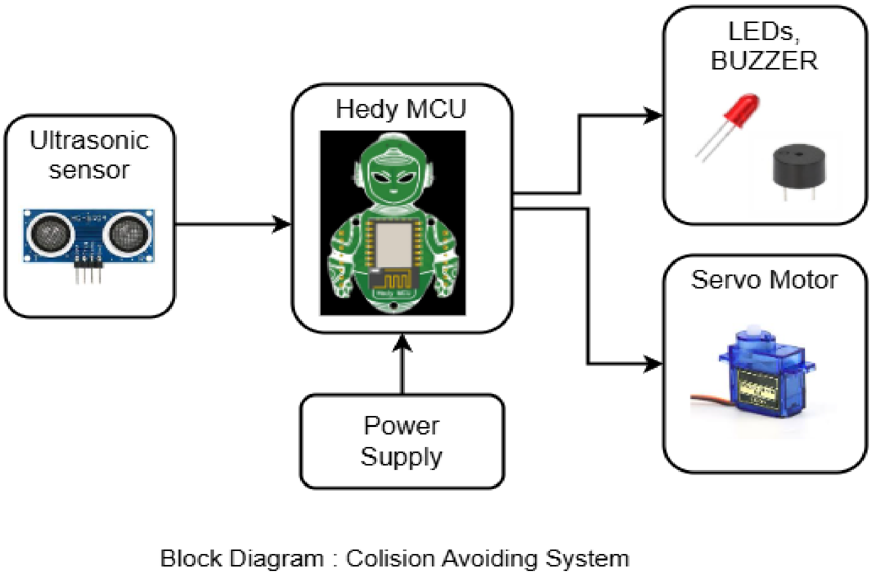

Figure 4: Block Diagram of Collision Avoiding System

This block diagram shows a Collision Avoiding System using an ultrasonic sensor, Hedy MCU, LEDs, buzzer, and a servo motor. The ultrasonic sensor detects obstacles and sends distance data to the Hedy MCU. Based on this input, the MCU activates the buzzer/LEDs for alerts and controls the servo motor to avoid collisions. All components are powered by a central power supply.

6.2 Circuit Diagram

The Collision Avoiding System circuit diagram shown in Figure 5.

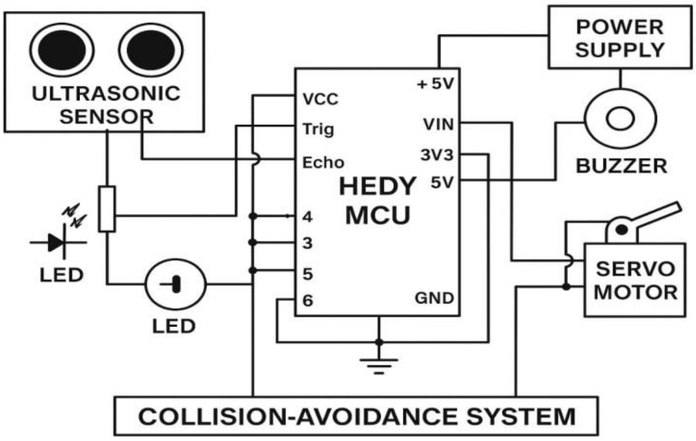

Figure 5: Circuit Diagram of Collision Avoiding System

6.3 Components required:

The various components required to develop Collision Avoiding System are as given below:

6.3.1 Hedy MCU (ESP8266):

The Hedy MCU, based on the ESP8266 microcontroller, acts as the brain of the system. It controls how all the components work together in the Collision Avoiding System.

❖ Role in the project:

- Reads data from the ultrasonic sensor to detect obstacles.
- Processes the data to calculate the distance.
- Decides what action to take:
  Turns ON the LED and buzzer if an obstacle is near.
  Rotates the servo motor to avoid the obstacle.
- Runs the Arduino code that controls the logic of the system.
- Can also be programmed via Wi-Fi (since ESP8266 supports wireless features).
  The Hedy MCU is shown in Figure 6.

Figure 6: Hedy MCU

❖ Why use it?

Compact and powerful also Supports multiple I/O pins for connecting sensors. It is easy to program using the Arduino IDE and Suitable for IoT and automation project.

6.3.2 Ultrasonic Sensor (HC-SR04):

It is used to detect obstacles in front of the system by measuring the distance between the sensor and nearby objects.

- The sensor continuously sends out ultrasonic waves.
- When these waves hit an object, they bounce back to the sensor.
- The system calculates the time taken for the echo to return, and based on that, it measures how far the object is.

  The Ultrasonic Sensor is shown in Figure 7.

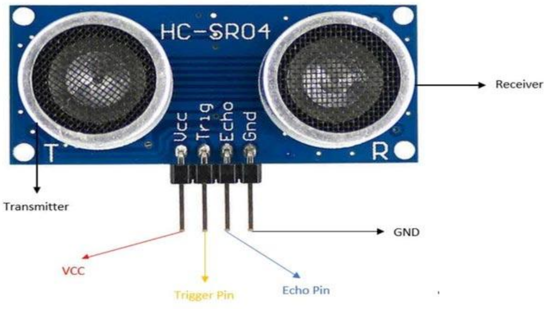

Figure 7: Ultrasonic Sensor

6.3.3 LED

In this project, the LED is used as a visual warning signal shown in Figure 8. It turns ON when the ultrasonic sensor detects that an obstacle is too close.

❖ Role of the LED:

- Acts as an alert to show that there is an object in the path.
- Turns OFF when no obstacle is detected

Figure 8: LED

6.3.4 Buzzer:

The buzzer shown in Figure 9 is used as an audio alert to warn that an obstacle is too close to the system.

❖ Role of the Buzzer:

- Provides a sound warning when the system detects an object nearby.
- Works along with the LED to alert users.
- Controlled by the Hedy MCU (ESP8266) based on the ultrasonic sensor data.

Figure 9: Buzzer

6.3.5 Servo Motor:

The Servo Motor shown in Figure 10 is used to change the direction of the system when an obstacle is detected and It is controlled by the microcontroller (ESP8266) based on distance data from the ultrasonic sensor.

If an object is detected too close, the servo rotates to a certain angle (e.g., 90°) to simulate the system turning away from the obstacle

If there's no obstacle, the servo stays in its default position (e.g., 0°), and the system continues in the same direction.

Figure 10: Servo Motor

7. Implementation

The collision Avoiding System is implemented using Arduino-compatible code uploaded to the Hedy MCU. The code is designed to read distance values from the ultrasonic sensor, process the data, and trigger appropriate actions such as blinking LEDs, sounding the buzzer, and controlling the servo motor. Based on the measured distance, the system alerts the user and activates the braking mechanism when necessary. This section explains the logic and functionality of the Arduino code used to control the entire system.

7.1 Arduino Code:

```
#include <Servo.h>
#define TRIG_PIN D6       // Ultrasonic sensor trigger pin
#define ECHO_PIN D5       // Ultrasonic sensor echo pin
#define SERVO_PIN D8      // Servo motor control pin
#define BUZZER D3         // Buzzer (Active Low)
#define LED_PIN D4        // LED Indicator (Active Low)

#define DISTANCE_THRESHOLD 150  // Warning starts at this distance (mm)
#define BRAKE_THRESHOLD 70      // Brake engages at this distance (mm)
#define BEEP_DELAY 200          // Beep delay in milliseconds

Servo brakeServo;
void setup()
{
Serial.begin(115200); // Initialize serial communication for debugging

pinMode(TRIG_PIN, OUTPUT);   // Configure ultrasonic trigger pin as output
pinMode(ECHO_PIN, INPUT);    // Configure ultrasonic echo pin as input
pinMode(BUZZER, OUTPUT);     // Configure buzzer pin as output
pinMode(LED_PIN, OUTPUT);    // Configure LED pin as output
```

```
brakeServo.attach(SERVO_PIN, 500, 2400); // Attach servo with defined pulse
brakeServo.write(0);          // Set initial servo position (0° - No brake)

digitalWrite(BUZZER, HIGH);   // Ensure buzzer is OFF at startup
digitalWrite(LED_PIN, HIGH);  // Ensure LED is OFF at startup

 Serial.println("Automatic Braking System Initialized");
 }

 Void loop()
{
 long distance = measureDistance(); // Measure the obstacle distance
 Serial.print("Obstacle Distance: ");
 Serial.print(distance);
 Serial.println(" mm");

 if (distance <= BRAKE_THRESHOLD) // If distance reaches braking point
{
 Serial.println("Obstacle too close! Applying brakes...");
 applyBrakes();
 }
 else if (distance <= DISTANCE_THRESHOLD) // If within warning zone
{
 Serial.println("Warning! Obstacle detected in range.");
warningSignal(distance);
}
else // If safe distance
{
 Serial.println("Safe distance maintained.");
 digitalWrite(BUZZER, HIGH); // Turn OFF buzzer
 digitalWrite(LED_PIN, HIGH); // Turn OFF LED
 brakeServo.write(0);         // Ensure brake is released
 }
 }
```

```
long measureDistance()
{
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);

  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH);
  return (duration * 0.343) / 2; // Convert duration to distance in mm
  }

void applyBrakes()
{
  digitalWrite(BUZZER, LOW); // Activate buzzer (Active Low)
  digitalWrite(LED_PIN, LOW); // Activate LED (Active Low)
  brakeServo.write(90);      // Move servo to 90° to engage brake
  delay(500);                // Hold brake position
  digitalWrite(BUZZER, HIGH); // Turn OFF buzzer after braking
}
void warningSignal(long distance)
{
nt=map(distance,BRAKE_THRESHOLD,DISTANCE_THRESHOLD,50,500);
digitalWrite(BUZZER, LOW); // Activate buzzer (Active Low)
digitalWrite(LED_PIN, LOW); // Activate LED (Active Low)
delay(beepDelay);
digitalWrite(BUZZER, HIGH); // Deactivate buzzer
digitalWrite(LED_PIN, HIGH); // Deactivate LED
delay(beepDelay);
}
```

7.2 Collision Avoiding System Working

The working of Collision Avoiding System is divided into three different parts (i) Obstacle Detection, (ii) Warning Signal Activation (iii) Automatic Braking Mechanism.

(i) Obstacle Detection:

- The Ultrasonic Sensor continuously sends ultrasonic waves and it receives the echo after hitting an object.
- It sends ultrasonic waves, and based on the time taken for the waves to reflect back, it calculates the distance.
- It is calculated using the formula:

$$Distance = Time \times Speed\ of\ Sound/2$$

(ii) Warning Signal Activation:

- If an obstacle is detected within a predefined safe distance, the Hedy MCU processes the data.
- The MCU triggers visual (LED) and audio (buzzer) alerts to warn the driver of a potential collision.

(iii) Automatic Braking Mechanism:

- If the distance between the vehicles continues to decrease despite the warning signals, the Hedy MCU activates the servo motor, which automatically applies the brakes, bringing the vehicle to a stop.
- This prevents a collision by ensuring a safe stopping distance.

7.3 Hardware Model:

Figure 11: Hardware Model

8. Applications

The applications of Collision Avoiding System are as follows:

- **(i) Autonomous Vehicles:** Used in self-driving cars to detect obstacles, maintain safe distances, and prevent collisions with pedestrians or other vehicles.
- **(ii) Robotics:** Implemented in mobile robots to enable smooth navigation in dynamic environments by avoiding walls, objects, and people.
- **(iii) Industrial Automation**: Applied in automated guided vehicles (AGVs) and robotic arms to prevent damage to machinery and ensure worker safety in factories.
- **(iv) Drones and UAVs**: Helps unmanned aerial vehicles avoid obstacles in the air or during landing/takeoff, ensuring stable and safe flight operations.
- **(v) Smart Wheelchairs**: Enhances mobility for people with disabilities by providing automatic obstacle detection and route adjustment.
- **(vi) Automated Parking Systems**: Used in vehicles for safe and accurate parking by detecting surrounding objects and guiding movements.
- **(vii) Home Automation (Smart Robots)**: In vacuum robots and other home assistants to prevent bumps and ensure efficient path planning.

9. Results

The Collision Avoiding System was tested under various conditions to evaluate its response to different obstacle distances. The results confirm that the system accurately detects objects within the defined thresholds and responds accordingly. When an object is detected within 150mm, the LED and buzzer are activated to alert the user. If the distance further decreases to 70mm, the braking mechanism is triggered through the servo motor. The table 1 below summarizes the system's behaviour based on specific distance parameters.

Table 1: System's behaviour based on specific distance parameters

Parameters	Values	Output
DISTANCE_THRESHOLD	150 mm	LED & Buzzer ON
BREAK_THRESHOLD	70 mm	Break Applied

10. Conclusion:

The Collision Avoiding System presents an intelligent safety mechanism that leverages real-time obstacle detection and automated braking to enhance vehicular safety and reduce the likelihood of accidents. By integrating components such as the ESP8266 microcontroller, ultrasonic sensors, and servo motors, the system achieves prompt and precise responses to potential hazards. Its modular and scalable design enables broad applicability across various domains, including autonomous vehicles, robotics, and industrial automation, thereby demonstrating its potential as a versatile and effective safety solution.

10.1 Important Findings:

- **Reliable Obstacle Detection:** The ultrasonic sensor consistently detects obstacles within a predefined range, enabling timely alerts.

- **Immediate Alert Mechanism:** The integration of LED and buzzer provides instant visual and auditory warnings upon obstacle detection.
- **Automated Directional Control**: The servo motor, controlled by the Hedy MCU (ESP8266), effectively alters the system's direction to avoid collisions.
- **Cost-Effective Implementation**: Utilizing readily available components like the ultrasonic sensor and ESP8266 microcontroller makes the system affordable and accessible.
- **Scalability for Future Enhancements:** The system's design allows for easy integration with additional sensors or modules, offering potential for enhanced functionalities.

.11. Future Scope:

- **AI & Machine Learning Integration:** AI can improve obstacle detection accuracy by analyzing patterns and predicting potential collisions.
- **Integration with Cloud & IoT:** The system can send real-time alerts to cloud servers and allow remote monitoring.
- **Advanced Braking Systems:** Integration with ABS and electronic braking systems for smoother and more effective stopping.
- **Autonomous Vehicle Applications:** Expansion into self-driving cars, industrial automation, and smart city traffic management.

12. References:

[1] tps://www.academia.edu/85920554/Design_and_Implementation_of_Car_Collision_Detection_Warning_System

[2] https://staff.futminna.edu.ng/CPE/content/journal/PF1790/4.pdf

[3] https://www.academia.edu/86227342/COLLISION_AVOIDANCE_SYSTEM?source=swp_share