Radar-Based Obstacle Detection System using ESP8266

Table of Contents

1. Introduction

Radar technology has been one of the most revolutionary developments of the 20th century. Originally developed for military applications during World War II, radar has now become an essential tool in civil, industrial, and academic domains. Its capability to detect objects at a distance by transmitting and receiving signals has enabled breakthroughs in aviation, navigation, weather monitoring, and defense.

In modern times, with the growing demand for IoT (Internet of Things), compact radar-like systems are being developed for real-world tasks such as robotics, automation, and smart cities. Traditional radar systems are expensive and complex, but by using microcontrollers such as the ESP8266 NodeMCU, it is possible to build a low-cost, portable, and Wi-Fi enabled radar system.

This project uses an ultrasonic sensor as the detection module, mounted on a servo motor to simulate radar scanning. Data is processed by the ESP8266 and displayed both on a local OLED display and a web-based radar interface accessible over Wi-Fi. Alerts are provided using a buzzer and a laser pointer.

2. Objectives

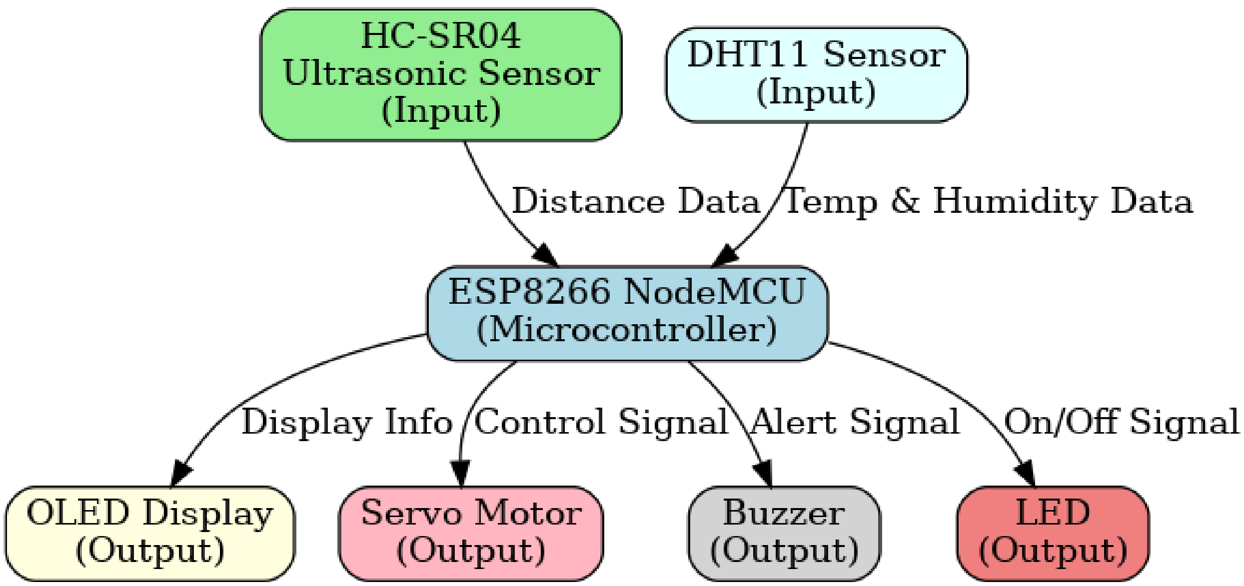
The main objectives of the project are:

- To design and implement a functional radar system for obstacle detection.
- To integrate ESP8266 NodeMCU as the system's controller with Wi-Fi connectivity.
- To provide both **local visualization (OLED)** and **remote monitoring (Web interface)**.
- To implement real-time alerts using a buzzer and laser diode.
- To ensure system flexibility with both **Access Point (AP)** and **Station Mode (STA)** Wi-Fi configurations.

- To make the system portable, low-cost, and adaptable to multiple environments.

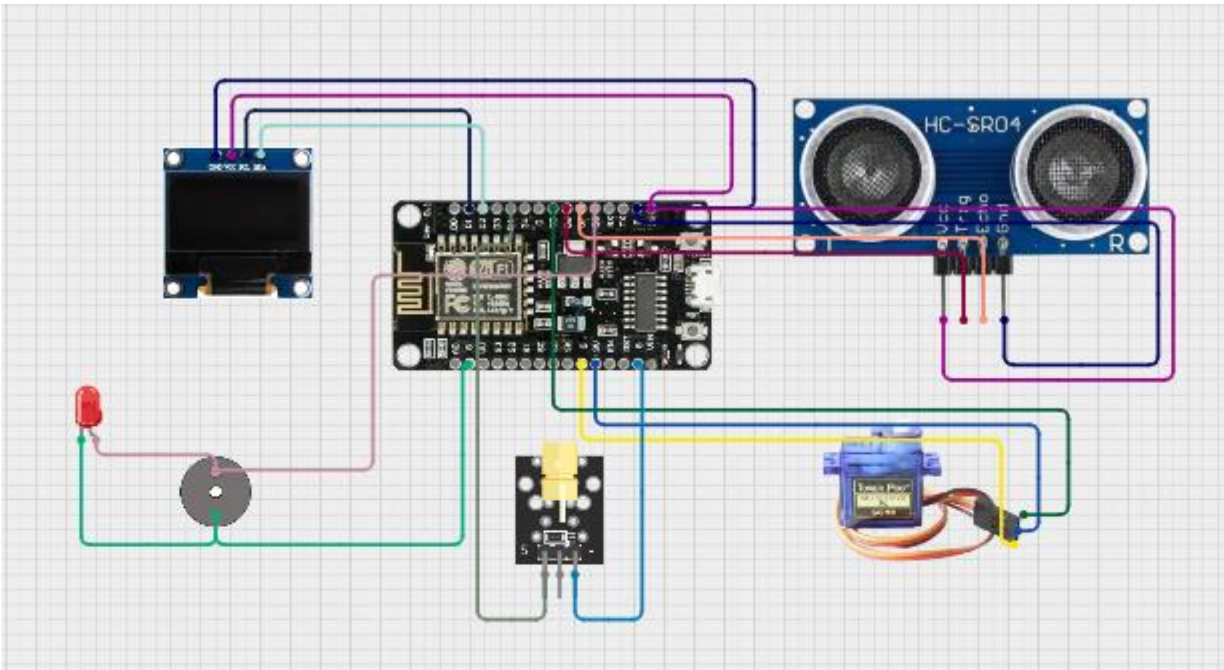
3. System Design

- 3.1 Block Diagram



The block diagram represents the interconnection between ESP8266, ultrasonic sensor, servo motor, OLED display, and alert systems.

- **3.2 Circuit Diagram**



Connections:

- Ultrasonic sensor → ESP8266 pin(D6 – Trig ,D7 - Echo).
- Servo motor → ESP8266 pin(D5).
- OLED → I2C pin(D1 – SCL ,D2 – SDA).
- Buzzer → ESP8266 pin(D8).
- Laser → ESP8266 pin(D4).
- LED → ESP8266 pin(D8).

4. Hardware Implementation

The system uses the following components:

1. **ESP8266 NodeMCU**
    a. 32-bit microcontroller with built-in Wi-Fi.
    b. Operates at 80 MHz with 4 MB flash memory.
    c. Handles data collection, processing, and web server hosting.
2. **Ultrasonic Sensor (HC-SR04)**
    a. Range: 2 cm to 400 cm.
    b. Frequency: 40 kHz.
    c. Accuracy: ±3 mm.
3. **Servo Motor (SG90/MG90S)**
    a. Rotates the ultrasonic sensor between 0°–180°.
    b. Provides scanning functionality.

4. **OLED Display (0.96" I2C)**
    a. Resolution: 128x64 pixels.
    b. Displays angle and distance in real-time.
5. **Buzzer & Laser Diode**
    a. Provide audio-visual alerts when obstacles are too close.
6. **Power Supply**
    a. 5V USB or rechargeable battery.
7. **Enclosure**
    a. Professional housing for protection and portability.

5. **Software Implementation**

5.1 Code

1. Include Libraries

You'll need libraries for each sensor and device:

```
#include <Wire.h>

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <DHT.h>

#include <Servo.h>
```

2. **Define Pins and Setup Components**

Define pin connections for ultrasonic sensor, DHT11, buzzer, LED, and servo.

```
// Ultrasonic Sensor Pins

#define TRIG_PIN D5

#define ECHO_PIN D6

// DHT11 Sensor

#define DHTPIN D2

#define DHTTYPE DHT11 DHT dht(DHTPIN, DHTTYPE);

// OLED Display Setup

#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 64 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);

// Servo Motor Servo servo;

#define SERVO_PIN D7

// Buzzer & LED

#define BUZZER_PIN D3
```

```
#define LED_PIN D4
```

3. **Setup Function**

Initialize everything inside `setup()`.

```
void setup() {
 Serial.begin(115200);
 // DHT Sensor
 dht.begin();
 // OLED Display
 if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
 Serial.println("OLED allocation failed");
   for(;;);
 }
 display.clearDisplay();
 // Servo
 servo.attach(SERVO_PIN);
 servo.write(0); // initial position
 // Buzzer & LED
 pinMode(BUZZER_PIN, OUTPUT);
 pinMode(LED_PIN, OUTPUT);
 // Ultrasonic Pins
 pinMode(TRIG_PIN, OUTPUT);
 pinMode(ECHO_PIN, INPUT);
}
```

4. **Ultrasonic Sensor Function**

Measure distance using HC-SR04.

```
long getDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long duration = pulseIn(ECHO_PIN, HIGH);
  long distance = duration * 0.034 / 2; // in cm
  return distance;
}
```

5. **Loop Function**

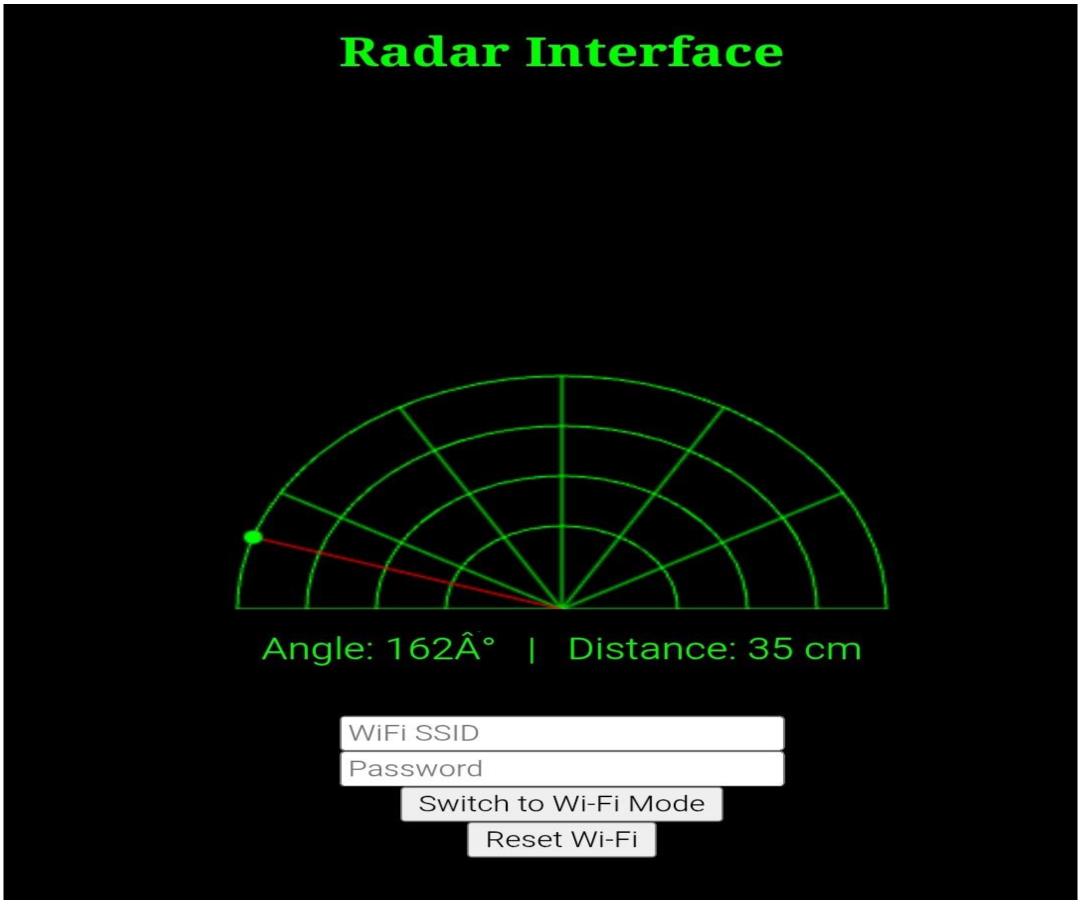
Here’s where data is read and actions are taken.

void loop() {

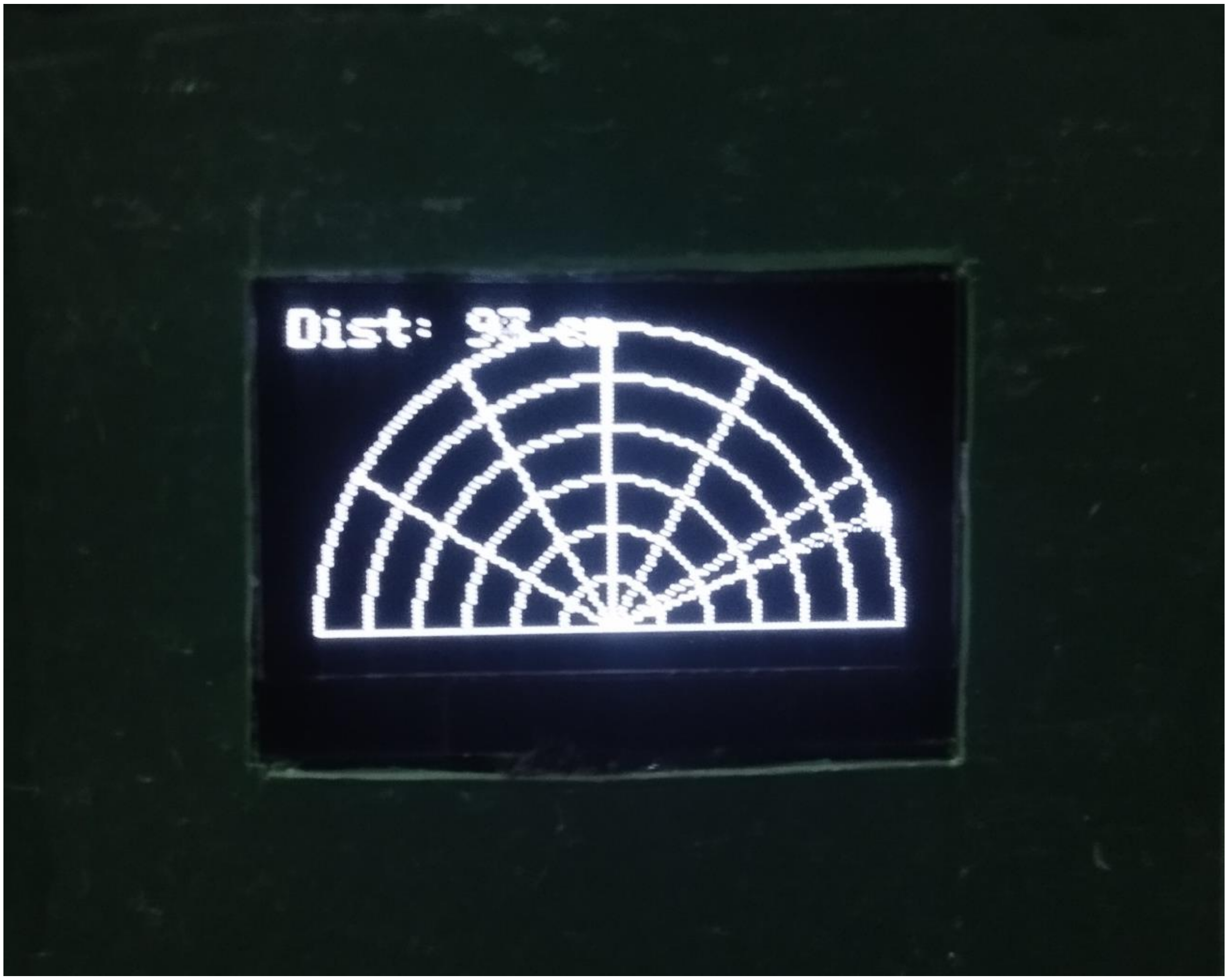
```
  // Read Ultrasonic Distance
  long distance = getDistance();
  // Read Temperature and Humidity
  float temp = dht.readTemperature();
  float hum = dht.readHumidity();
  // Display on OLED
  display.clearDisplay();
  display.setTextSize(1);
```

```
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0,0);
  display.print("Dist: "); display.print(distance); display.println(" cm");
  display.print("Temp: "); display.print(temp); display.println(" C");
  display.print("Hum : "); display.print(hum); display.println(" %");
  display.display();
  // Control Servo Motor Example
  if (distance < 20) {
   servo.write(90); // open
  } else {
   servo.write(0); // close
  }
  // Alert with Buzzer & LED
  if (temp > 30) { // high temperature alert
   digitalWrite(BUZZER_PIN, HIGH);
   digitalWrite(LED_PIN, HIGH);
  } else {
   digitalWrite(BUZZER_PIN, LOW);
   digitalWrite(LED_PIN, LOW);
  }
  delay(1000); // 1-second refresh
}
```

- **5.2 Interface**
  - Web Interface



- **built in interfaces**



6. **Results & Analysis**

The system was tested under different conditions:

- Successfully detected obstacles up to **4 meters**.
- OLED displayed **angle + distance** accurately.
- Web interface provided **real-time radar-like sweeping visualization**.
- Buzzer and laser alerts worked instantly for close obstacles.

Advantages:

- Low cost and portable.
- Dual-display output.
- Wi-Fi enabled for remote monitoring.

Limitations:

- Limited range compared to LiDAR.
- Servo speed limits scanning refresh rate.
- Ultrasonic sensors give inaccurate readings on soft/angled surfaces.

7. Applications

- **Security Systems** – detecting unauthorized movement.
- **Smart Parking** – guiding vehicles.
- **Robotics** – autonomous navigation.
- **Military & Surveillance** – perimeter detection.
- **Industrial Automation** – collision prevention in factories.

8. Conclusion & Future Work

The project successfully demonstrates a **low-cost radar-like obstacle detection system** using ESP8266 and ultrasonic sensors. With dual visualization (OLED + Web) and alert mechanisms, it is suitable for real-time applications in robotics, automation, and security.

Though limited in range and accuracy compared to advanced sensors, it provides an excellent foundation for **IoT-enabled radar systems** and strengthens knowledge of **microcontrollers, sensors, and IoT applications**.

9. Future Scope

- IoT cloud integration for remote access worldwide.
- AI-based object classification.
- Faster and more accurate sensors (LiDAR, mmWave radar).
- Mobile app visualization.
- Solar-powered operation.
- Integration into **drones & autonomous vehicles**.