AIR DEFENCE RADAR SYSTEM

PROJECT PHASE-1(7ETRP1)

SESSION:- 2025-26

CLASS:- 4TH YEAR E&TC 'B'

- Submitted to: -

  DR. SANGITA SOLANKI

- Submitted by: -

  Ankush Kumar Verma(22T7113)

  Omprakash Alawa (22T7149)

  Yash sisodiya (22T7179)

  Priyansh Vimal (22T7181)

  Sarthak Verma (22T7183)

- Signature: -

DR. SANGITA SOLANKI [Mentor]

DR. SANJIV TOKEKAR [H.O.D.]

INTRODUCTION

- Air defense systems play a critical role in ensuring the security of nations by protecting against potential airborne threats such as drones, aircraft, and missiles.
- With the growing accessibility of technology, small-scale prototypes of such systems can be developed using platforms like Arduino to simulate their operation.
- This project demonstrates the core functionalities of an air defense system , such as detection, tracking.

OBJECTIVES TARGETED

1. *Real-Time Target Tracking:- Continuously monitor the position, speed, and trajectory of multiple targets simultaneously.*
2. *Environmental Adaptability:- Ensure reliable performance in diverse conditions such as mountainous terrain, urban clutter, and adverse weather.*
3. *Integration with Defence Infrastructure:- Seamlessly connect with missile systems, command centers, and communication networks for coordinated response.*
4. *Scalability and Modularity:- Design the system to be scalable for different operational zones and modular for easy upgrades.*

OBJECTIVE ACHIEVED

📡 Wireless Data Transmission

Transmit sensor data wirelessly to a remote server or dashboard using NodeMCU's built-in Wi-Fi.

📊 Real-Time Monitoring Interface

Display object distance, speed, and direction on a blink app.

⚠️ Alert System

Trigger alarms (buzzer, LED, or notification) when an object is detected within a critical range.

🔌 Low-Cost and Scalable Design

Demonstrate how low-cost microcontrollers can be used for basic surveillance and monitoring tasks.

📚 Educational Value

Provide a hands-on learning experience in IoT, embedded systems, and basic radar principles.

PROBLEM STATEMENT

- **Detection Range:** The effectiveness of the radar system may be limited by the range capabilities of the ultrasonic sensors used. Factors such as sensor resolution, signal attenuation, and environmental conditions (e.g., weather, obstacles) can impact the system's detection range and accuracy.
- **Object Classification:** The radar system may have limitations in accurately classifying detected objects as potential threats. The system's classification algorithms may be affected by factors such as object size, shape, and movement patterns, leading to false positives or false negatives in threat detection.
- **Hardware Constraints**: The capabilities of Arduino microcontrollers and ultrasonic sensors may impose limitations on the system's performance and functionality. Memory constraints, processing power, and sensor accuracy may influence the system's overall effectiveness and reliability.
- **Regulatory Compliance:** The deployment and testing of the radar system with defense capabilities may be subject to legal and regulatory requirements related to defense technology development and use. Compliance with relevant laws, standards, and regulations must be considered throughout the project to ensure ethical and legal use of the system.

MATERIALS REQUIRED

- Arduino microcontroller board (e.g., Arduino Uno or Arduino Mega)
- Resistor
- Radar sensor module (e.g.,HC-SR04 Ultrasonic Sensor).
- Display module (e.g., LCD or Laptop)·
- Motorized mechanism (e.g., servo motor) for radar scanning.
- Buzzer
- Power supply (e.g., batteries or AC adapter)
- Breadboard and jumper wires for prototyping
- Red and Green LED light

HARDWARE USED

1. Ultrasonic Sensor

The ultrasonic sensor (e.g., HC-SR04) emits sound waves and measures the time it takes for the echo to return, calculating the distance to an object. This mimics radar detection of aerial threats.

2. Servo motor

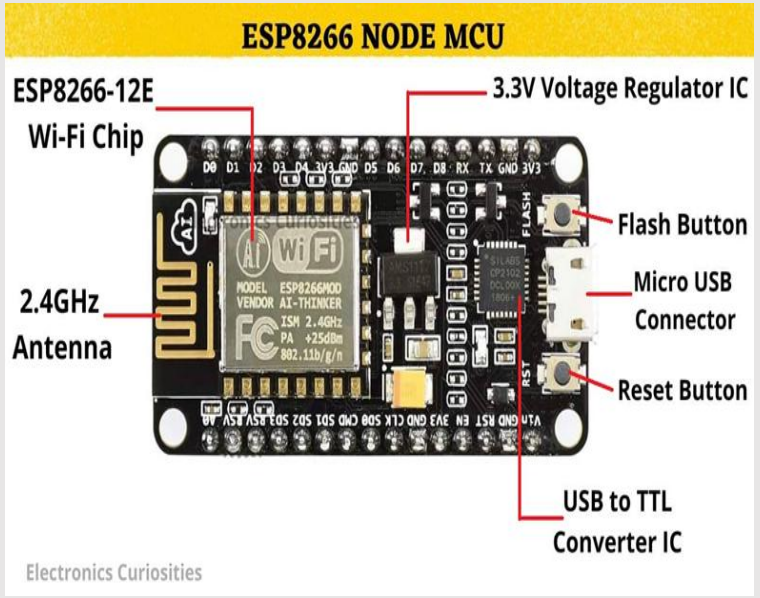
The servo motor rotates the ultrasonic sensor (or other detection module) across a defined angular range (e.g., 0° to 180°), allowing the system to scan a wide area for objects.

3. 1 KΩ Resistor - Limits current to protect components like the transistor and sensors.

HARDWARE USED

4.Node MCU Microcontroller board

NodeMCU combines both hardware and software components, primarily utilizing the ESP8266 chip from Espressif Systems. This microcontroller integrates WiFi capabilities, making it ideal for IoT projects



5.Breadboard - Used for circuit prototyping and easy connections without soldering.

HARDWARE USED

Jumper Wires - Connect all the components together to form the working circuit.

Buzzer- The buzzer sounds an alarm when an object is detected within a critical or predefined distance, simulating how real radar systems alert operators of incoming threats.

Led- to show the status of tracked object.

SOFTWARE USED

1. Arduino IDE- Used to write, compile, and upload the program (code) into the Arduino UNO board.
2. .Embedded C / Arduino Programming Language - The logic that controls fire detection, robot movement, and water pumping is written in this language.
3. Blynk – it is used for real time tracking and controlling the environmental parameters or potential threats.

WORKING PRINCIPLE

🔄 Scanning Mechanism (Servo Motor)

- The servo motor rotates the ultrasonic sensor across a defined angular range (e.g., 0° to 180°).
- This simulates radar scanning by sweeping the sensor to detect objects in different directions.

📡 Object Detection (Ultrasonic Sensor)

- The ultrasonic sensor emits sound waves and measures the time it takes for the echo to return.
- Based on this time, it calculates the distance to any object in its path.

🧠 Data Processing (NodeMCU)

- The NodeMCU reads the distance and angle data from the sensor and servo.
- It processes this information to determine the location of detected objects.

📊 Visualization (Optional Web Interface)

- The NodeMCU can send data via Wi-Fi to a web dashboard or serial monitor.
- This allows users to **visualize object positions** in a radar-like display.

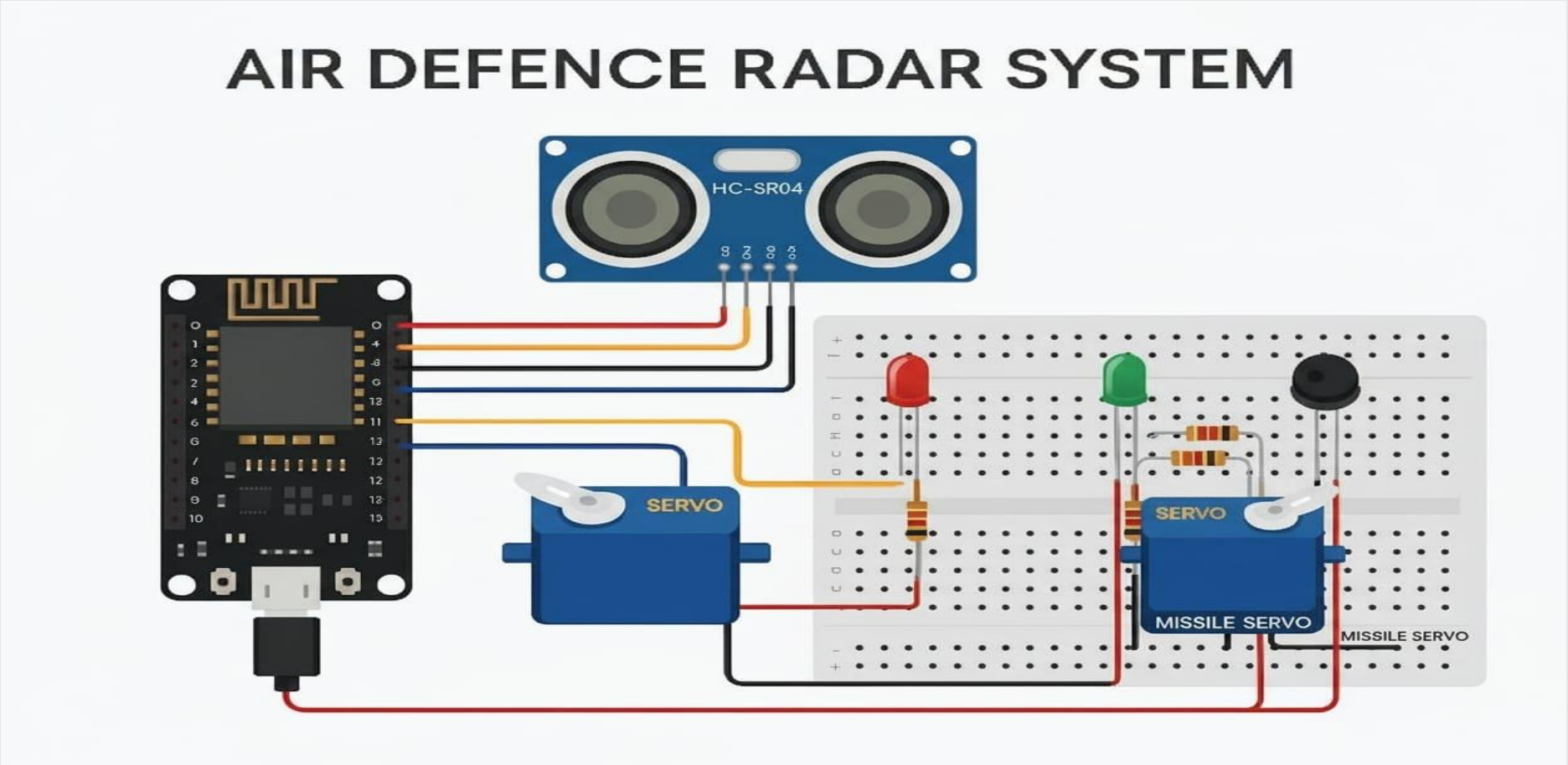
🔊 **Alert System (Buzzer)**

- If an object is detected within a **critical range**, the **buzzer** is activated. This simulates a threat alert, warning users of nearby intrusions.

🔁 **Continuous Monitoring**

- The system continuously scans, detects, and updates object positions in real time. It mimics how actual radar systems monitor airspace for threats.

CIRCUIT DIAGRAM



PROJECT CODE

```
// --- Air Defence Radar System with Missile Launch Integration ---
// Components: HC-SR04, 2x Servo, LEDs, Buzzer, Blynk
// --- Blynk Template Info ---
#define BLYNK_TEMPLATE_ID "TMPL3H3WBOkgn“
#define BLYNK_TEMPLATE_NAME "Air defence radar system “
#define BLYNK_AUTH_TOKEN "S9PosdsO2tzeexvOIEmzMam66lm1x63S“
// --- Libraries ---
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>
// --- WiFi Credentials ---
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "Redmi 12 5G";
char pass[] = "00000000";
// --- Pin Configuration ---
#define TRIG_PIN D5
#define ECHO_PIN D6
#define RADAR_SERVO_PIN D7
#define MISSILE_SERVO_PIN D4
#define RED_LED D1
#define GREEN_LED D2
#define BUZZER_PIN D3

Servo radarServo;
Servo missileServo;

int pos = 0;
int distance;
bool missileLaunched = false;
```

PROJECT CODE

```
// --- Setup ---
void setup() {
  Serial.begin(115200);

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

  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);

  digitalWrite(RED_LED, LOW);
  digitalWrite(GREEN_LED, HIGH);
  digitalWrite(BUZZER_PIN, LOW);

  radarServo.attach(RADAR_SERVO_PIN);
  missileServo.attach(MISSILE_SERVO_PIN);

  missileServo.write(0); // Missile ready position
  Blynk.begin(auth, ssid, pass);
   Serial.println(" 🔗 Connecting to WiFi & Blynk...");
}
// --- Main Loop ---
void loop() {
  Blynk.run();
  // Sweep radar left to right 0 to 180°
  for (pos = 0; pos <= 180; pos += 3) {
    radarServo.write(pos);
    delay(25);
    measureDistance();
  }
 // Sweep radar right to left 180 to 0°
  for (pos = 180; pos >= 0; pos -= 3) {
    radarServo.write(pos);
    delay(25);
    measureDistance();
  }
}
```

PROJECT CODE

```
// --- Measure Distance ---
void measureDistance() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  long duration = pulseIn(ECHO_PIN, HIGH);
  distance = duration * 0.034 / 2; // in cm
  Serial.print("Angle: ");
  Serial.print(pos);
  Serial.print("° | Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  Blynk.virtualWrite(V1, pos);      // Radar angle
  Blynk.virtualWrite(V0, distance);  // Live distance
// --- Security Alert ---
if (distance > 0 && distance <= 15) {
  digitalWrite(RED_LED, HIGH);
  digitalWrite(GREEN_LED, LOW);
  digitalWrite(BUZZER_PIN, HIGH);
  Blynk.virtualWrite(V2, 1);  // App alert ON
   if (!missileLaunched) {
    launchMissile();         // Fire only once
    missileLaunched = true;
  } }
  else {
  digitalWrite(RED_LED, LOW);
  digitalWrite(GREEN_LED, HIGH);
  digitalWrite(BUZZER_PIN, LOW);
  Blynk.virtualWrite(V2, 0);
  missileLaunched = false;  // Reset when safe
}}
```

PROJECR CODE

```
// --- Missile Launch Function ---
void launchMissile() {
  Serial.println("🚀 Target Detected! Launching Missile...");

  missileServo.write(120);  // Fire
  delay(150);
  missileServo.write(0);   // Reset

  Blynk.virtualWrite(V12, "🚀 Missile Launched!");
}
```

TECHNICAL DISCUSSION

- 📡 Challenges
- • Range Constraints: Ultrasonic sensors are limited to short-range detection (~2–4 meters).
- • Environmental Sensitivity: Accuracy can be affected by temperature, humidity, and surface texture.
- • No target classification: The system cannot distinguish between types of objects (e.g., bird vs drone).
- • Manual calibration: Servo angles and detection thresholds require manual tuning.

- 🔧 Potential Enhancements
- • Replace ultrasonic sensors with microwave or LiDAR for longer range and better precision.
- • Add AI-based object classification using camera modules and machine learning.
- • Integrate wireless communication (e.g., Wi-Fi, LoRa) for remote monitoring.
- • Use GPS modules for geolocation tracking of threats.

RESULTS AND OUTCOMES

- ✅ **Functional Object Detection**
- • The system reliably detects objects within a set threshold distance using an HC-SR04 ultrasonic sensor, confirming its effectiveness for short-range surveillance.
- • Real-time scanning was achieved through servo motor rotation, enabling horizontal and vertical coverage of the monitored area.
- 🛠️ **Automated Threat Response**
- • Upon detecting an object, the system triggers an alert mechanism (buzzer or LED), simulating a defensive response.
- • The prototype can track object position and display it visually using software like Processing IDE, enhancing situational awareness.
- 💡 **Proof of Concept Validation**
- • The project successfully demonstrates that Arduino-based platforms can be used to build basic radar systems for defense simulations.
- • It validates the integration of servo motors, sensors, and microcontroller logic for coordinated scanning and detection.

APPLICATIONS

- **Integration with Defense Mechanisms:**
- Design and implement methods for integrating the radar data with various cost-effective radar system for defense defensive actions like alarms, automated barriers, or communication protocols Investigate the feasibility of using Arduino as a central control unit for coordinating and ultrasonic sensors. It aims to provide different defensive responses.
- **Educational and Research Purposes:**
- Develop a low-cost, accessible platform for studying the fundamental principles of radar systems and their potential applications in defense scenarios. Serve as a foundation for further research into more advanced radar technologies and integration with more sophisticated control systems.

🚀 FUTURE SCOPE AND ENHANCEMENTS

- The current prototype opens up several avenues for future development and deployment:
- • *Integration with AI and machine learning:* Future systems can incorporate AI for intelligent target classification, threat prioritization, and predictive analytics.
- • *Advanced sensor technologies:* Upgrading to radar modules with higher range and precision (e.g., millimeter-wave radar) can improve detection accuracy.
- • *Wireless communication and IoT:* Adding wireless modules (e.g., Wi-Fi, LoRa) enables remote monitoring and control, making the system suitable for distributed defense networks.
- • *Scalability and modularity:* The system can be scaled to cover larger areas or integrated into mobile platforms like drones or autonomous vehicles.
- • *Enhanced user interface:* Developing a graphical interface for real-time visualization of detected threats and system status can improve usability.
- • *Cybersecurity measures:* As the system becomes more connected, implementing robust security protocols will be essential to prevent unauthorized access or manipulation.

REFERENCES

- https://www.scribd.com/
- https://projecthub.arduino.cc/hashan_sudeera/radar-system-using-arduino-dedc7d
- https://researchgate.com/researchpdf
- GitHub - Ruthumani001/-Arduino-Controlled-Radar-Guided-Defence-System: Arduino-based radar-guided missile defense prototype using ultrasonic sensors and servo motors.
- Some references are also taken from ChatGPT.