International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Advanced Arduino-Based Air Defence System with Object Detection, Tracking, and Signal Jamming

Piyush Kumar[1], Kishan Jaiswal[2], Pratik Kumar[3], Shreya Narayan[4]

12407905
12407973
12410146
12407895
Department of Computer Science Lovely Professional University, Jalandhar, Punjab, India
Email: jaisukhpal.23875@lpu.co.in

ABSTRACT—

This paper describes the design and implementation of an Arduino-based air defence system to detect, track, and simulate jamming of airborne objects like drones. The system incorporates an ultrasonic sensor to detect the object, servo motors to track the object, a 433 MHz jammer (simulated through an LED/relay), and a 16x2 LCD to display the status. The project shows an economical way to implement low-range drone defence systems using simple electronic components and microcontroller programming.

Keywords—Arduino, air defence system, object detection, signal jamming, servo motors, ultrasonic sensor.

I. INTRODUCTION

As there has been a quick spread of consumer and commercial drones, there has been a requirement for low-cost air defence systems. These conventional systems are not only very complex but also very costly. This project fills this gap by developing an economical solution that is able to detect, track, and neutralize small aerial targets using Arduino Uno and related peripherals.

II. SYSTEM DESIGN AND ARCHITECTURE

A. Components Overview

The main components used in the project are summarized in Table I.

TABLE I. COMPONENTS USED

Component	Quantity	Purpose
Arduino Uno	1	Central microcontroller
Ultrasonic Sensor (HC-SR04)	1	Object detection
Servo Motors (SG90 or MG995)	2	Object tracking
433 MHz Jammer (or LED + Relay for simulation)	1	Signal jamming
16x2 LCD (Parallel)	1	Status display
Breadboard and Jumper Wires	Various	Connections
9V Battery or Adapter	1	Power supply

B. Working Principle.

The ultrasonic sensor identifies. objects in a predetermined range (e.g., 100 cm) by transmitting. Ultrasonic waves and calculating the time for echo reception. When detected:

-. The system engages servo motors to follow. the. Object position.

- The jammer (simulated by LED/relay) turns on to simulate signal jamming.
- The LCD displays real-time status updates.

III. HARDWARE IMPLEMENTATION

A. Arduino Uno

The Arduino Uno takes care of managing sensor input, controlling the servo motor, activating jamming, and providing feedback to the user through the LCD. It interprets sensor readings and makes decisions regarding tracking and jamming.

B. Ultrasonic Sensor (HC-SR04)

The HC-SR04 sensor calculates object distance as follows:

[
$\text{Distance (cm)} = (\text{Time} \times 0.034)/2$
]

A detection range threshold of 100 cm was used for object tracking initiation.

C. Servo Motors

Two servo motors support two-axis rotation:
- *Servo X*: Left-right horizontal movement
- *Servo Y:* Up-down vertical movement

The servo positions are dynamically adjusted to follow the detected object.

D. Signal Jamming Simulation

During simulation mode, jammer activation is indicated by a relay or an LED. Arduino activates the jammer on successful object tracking.

E. 16x2 LCD Display

The LCD displays messages like "Target Detected", "Tracking", "Jamming Active", or "No Target" together with distance readings, improving user feedback.

IV. SOFTWARE PROGRAMMING

The Arduino was programmed with the Arduino IDE. Key features are:

-*Sensor Handling:* Trigger and echo pulses for distance calculation.
-*Motor Control*: PWM signals control servo angles.
- *Jam Activation*: Conditional jammer activation.
- *LCD Display*: Status reports based on system events.

A segment of the fundamental logic:

~~~
cpp
if (distance > 0 && distance < detectionRange) {
// Target detected
// Sweep servos and engage jammer
} else {
// No target
// Reset servos and jammer
}
```
~~~

The full working code is presented in the Appendix section.

V. RESULT AND DISCUSSION

The system was able to detect objects within the set range and accomplish simple tracking through servo movement. Jamming was properly simulated. The project confirms that a minimal microcontroller system can replicate air defence capabilities at a low cost. Drawbacks are the short range of detection and rudimentary tracking mechanism.

VI. CONCLUSION

A sophisticated, Arduino-powered air defence system was successfully developed and tested. The project shows the viability of creating an effective low-cost proof-of-concept that can identify, track, and emulate the neutralization of airborne threats. Future enhancements could include the addition of radar modules, sophisticated tracking software, and actual jamming hardware for actual deployment.

APPENDIX

A. Final Arduino Code

```
#include <Servo.h>
#include <LiquidCrystal.h>  // For non-I2C LCD

// Ultrasonic Sensor
const int trigPin = 9;
const int echoPin = 10;

// Servo Motor Pins
Servo servoX;
Servo servoY;
const int servoXPin = 5;
const int servoYPin = 6;

// Jammer Control (simulate with LED or relay)
const int jammerPin = 8;

// LCD Setup (Non-I2C)
const int rs = 7, en = 4, d4 = 3, d5 = 2, d6 = A0, d7 = A1;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// Detection threshold
const int detectionRange = 100; // cm

void setup() {
  Serial.begin(9600);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(jammerPin, OUTPUT);
  digitalWrite(jammerPin, LOW); // Jammer OFF

  servoX.attach(servoXPin);
  servoY.attach(servoYPin);
  servoX.write(90); // Neutral
  servoY.write(90);

  lcd.begin(16, 2);
  lcd.print("Air Defense Sys");
  delay(2000);
  lcd.clear();
}

void loop() {
  long duration;
  int distance;

  // Trigger pulse
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Echo receive
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;

  // LCD Display
  lcd.setCursor(0, 0);
  lcd.print("Distance: ");
  lcd.print(distance);
  lcd.print("cm ");
  // Detection logic
  if (distance > 0 && distance < detectionRange) {
    lcd.setCursor(0, 1);
    lcd.print("Target Detected ");

    // Simple Servo Sweep (simulate tracking)
    for (int angle = 75; angle <= 105; angle += 5) {
      servoX.write(angle);
      servoY.write(180 - angle);
      delay(50);
    }

    // Activate Jammer
    digitalWrite(jammerPin, HIGH);
    lcd.setCursor(0, 1);
    lcd.print("Jamming Active  ");
    delay(1000);
    digitalWrite(jammerPin, LOW);
  } else {
    lcd.setCursor(0, 1);
    lcd.print("No Target       ");
    digitalWrite(jammerPin, LOW);
  }

  // Reset
  servoX.write(90);
  servoY.write(90);
  delay(500);
}
```

REFRENCES

[1] Arduino Official Documentation. Available: https://www.arduino.cc/

[2] HC-SR04 Ultrasonic Sensor Datasheet.

[3] Servo Motor Control using Arduino.

[4] Basic Principles of Electronic Warfare and Signal Jamming.