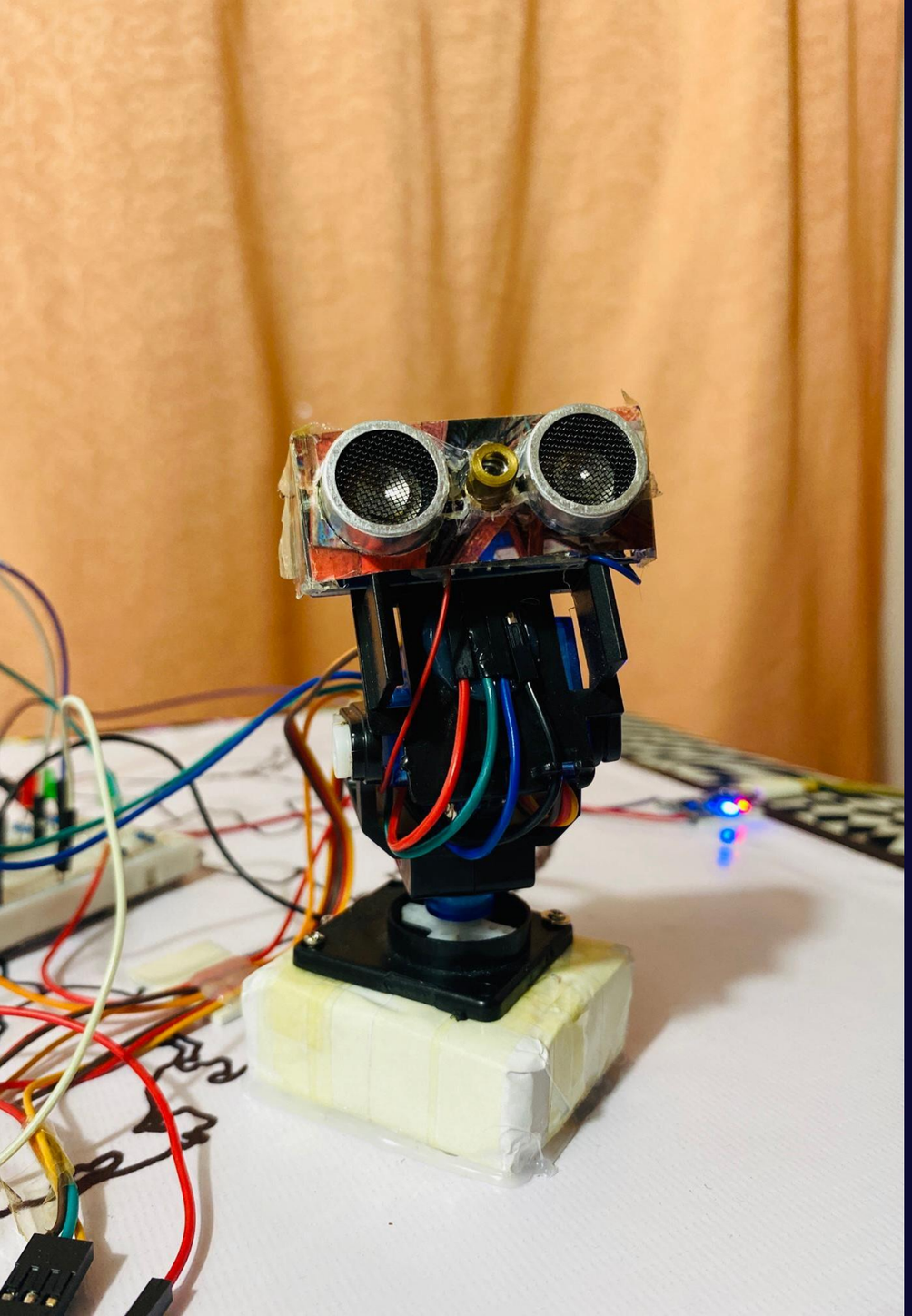
Smart Iron Dome: An IoT-Based Laser Defense System

UNIVERSITY OF SRI JAYEWARDENEPURA
Faculty of Applied Sciences
Department of Physics

Supervisor : K.S. Mannatunga

IBRALEBBE ABDULLAH
AS2021903



Introducing the Smart Iron Dome

Advanced Defense Inspiration

Drawing on cutting-edge defense systems to neutralize aerial threats effectively.

Affordable IoT Prototype Prototype

Developed a cost-effective, IoT-based prototype for robust object tracking.

Precise Laser Interception Interception

Utilizes laser technology to simulate highly accurate and efficient threat interception.

Problem Statement

High Cost Systems

Current defense solutions are prohibitively expensive.

Limited Accessibility

Advanced systems are not widely available.

No Budget Alternatives

Lack of affordable, effective options.

Objectives

Detect with Ultrasonic

Utilize ultrasonic sensors for object detection.

Track with Servos

Employ servo motors for precise tracking.

Display Radar UI

Visualize tracking on a radar interface.

Trigger Laser

Activate laser within a 25 cm range.

Literature Review

Reviewed Existing Systems

Examined various radar and tracking technologies.

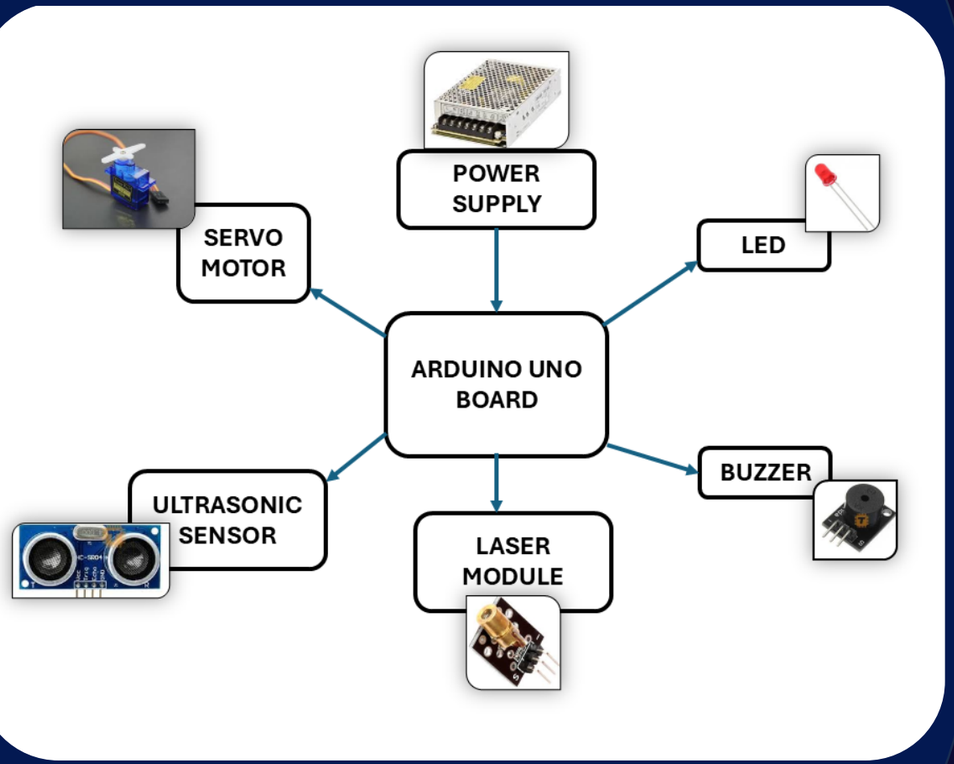
AI and Multiple Sensors

Most solutions relied on AI or complex sensor arrays.

Simple, Affordable Approach

Our project focuses on cost-effective simplicity.

System Architecture



Ultrasonic Sensor

Serves as the primary detection module, emitting ultrasonic waves to detect the proximity and distance of incoming objects with high accuracy, feeding this data for initial threat assessment.

Arduino

The central processing unit of the system, responsible for receiving data from the ultrasonic sensor, calculating target coordinates, and sending precise control signals to the servo motors and laser based on the analyzed data.

Servo Motors + Laser

Controlled by the Arduino, the servo motors precisely orient the laser to track the detected target. The laser then activates within the defined range, simulating neutralization of the aerial threat.

Buzzer

Provides audible alerts for target detection, tracking status, or system errors, offering immediate auditory feedback to the user.

LED Indicator

Offers visual status cues, such as power on, target acquired, or laser active, enhancing user awareness through simple light signals.

PC (Processing UI)

A dedicated user interface running on a PC visualizes the real-time radar data, displaying detected objects, their positions, and the laser's tracking path, providing an intuitive operational overview.

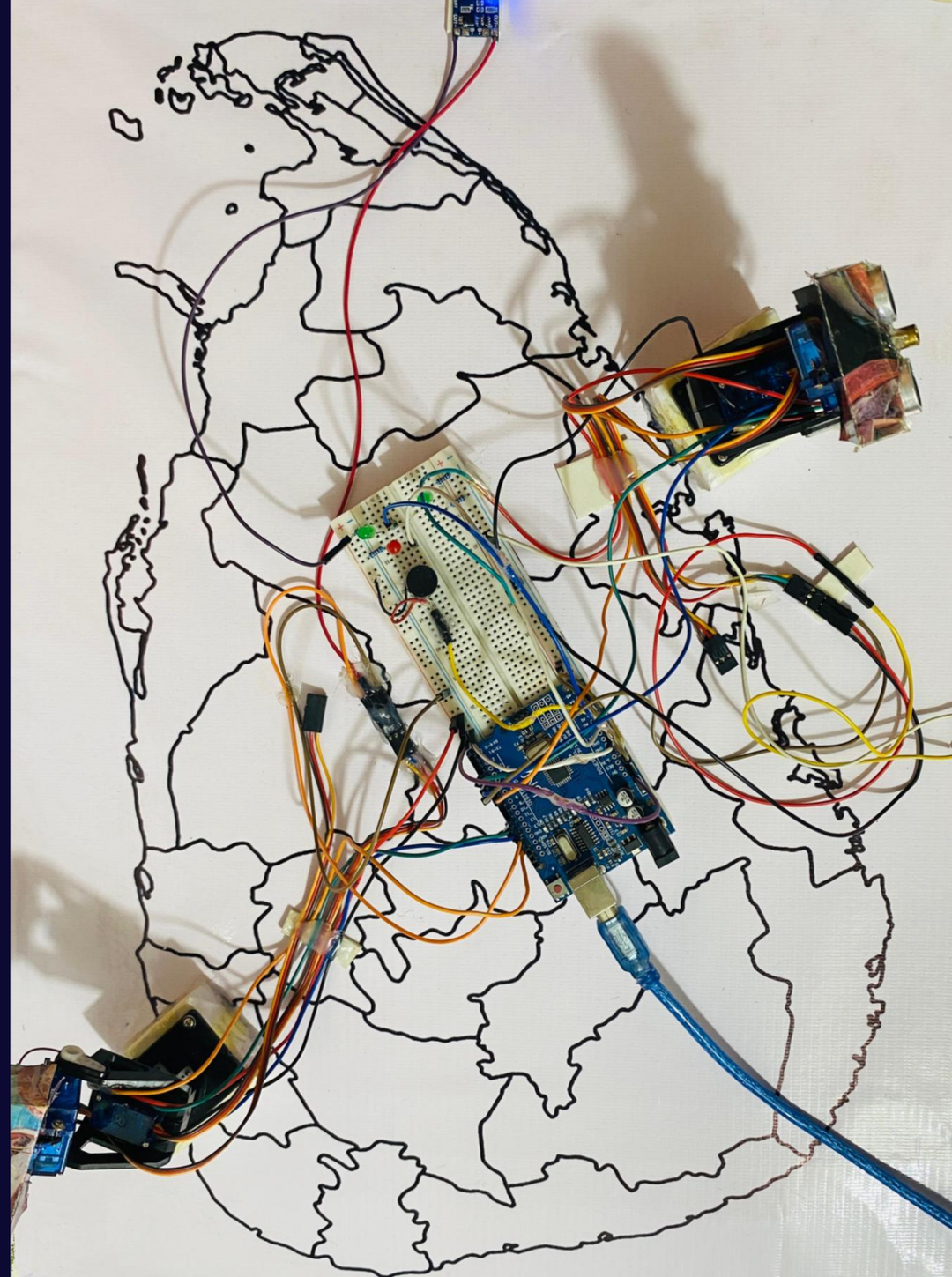
Hardware Components & Methodology

Hardware Components

- Arduino Uno
- 2x HC-SR04 Ultrasonic Sensor
- 2x SG90 Servo Motors
- Laser Module
- Breadboard & Wiring
- Buzzer
- LED
- External Power Supply

Software & Methodology

1. Servo scans using ultrasonic data.
2. Detects object ≤ 25 cm.
3. Stops & activates laser.
4. Visualizes on radar UI.



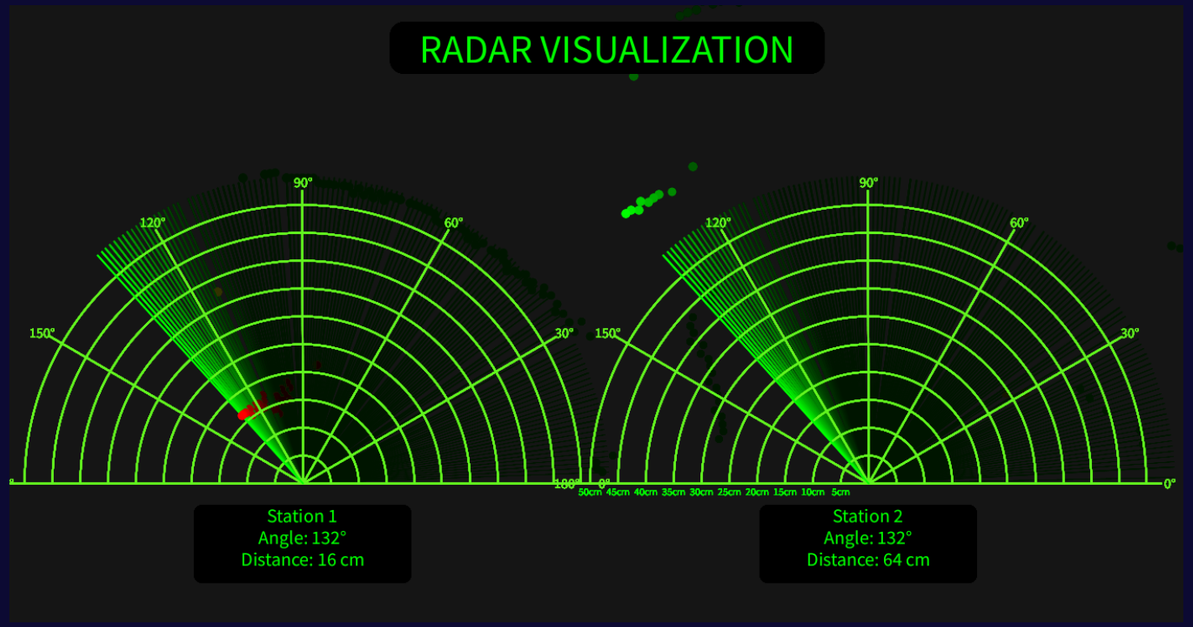
Radar Visualization & Results

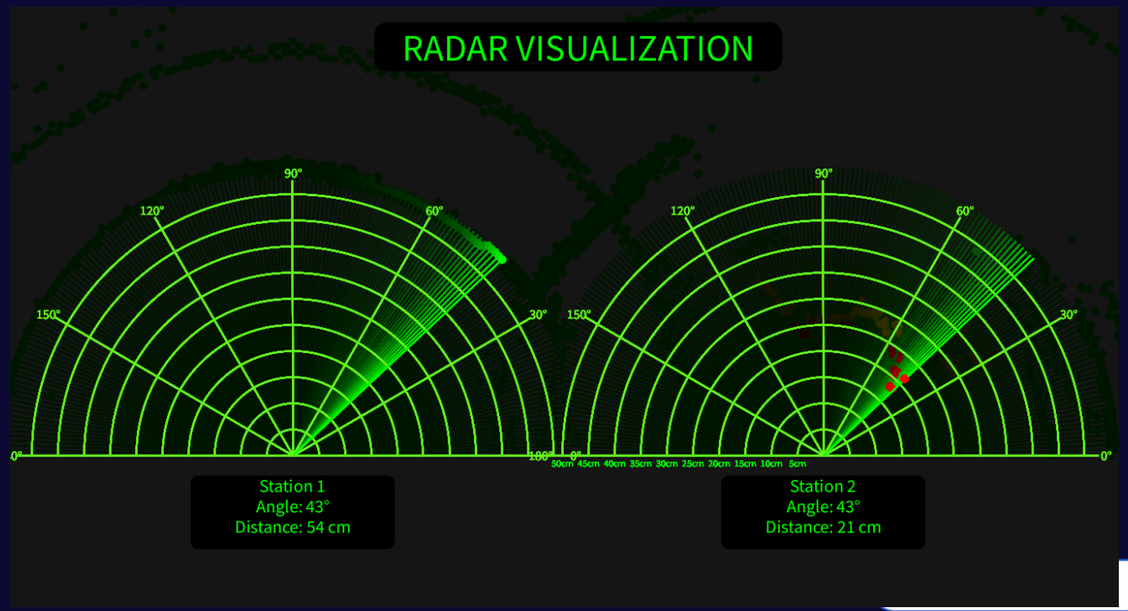
Radar Visualization

- Processing IDE: Radar simulation.
- Displays angle, distance, fading trail.
- Real-time UI animation.
- Dual sweep support.

Results

- Detects objects reliably within 25 cm.
- Smooth servo targeting.
- Radar shows real-time feedback.





Challenges Faced & Future Improvements

1. **Sensor Noise & Delay**
   Challenges encountered during real-time operation.
2. **Servo Jitter**
   Minor inaccuracies in motor movements.
3. **Serial Communication Lag**
   Delays in data transfer to the UI.
4. **PID Control**
   Implement for enhanced precision.
5. **ESP32-CAM & OpenCV**
   Integrate camera for visual tracking.
6. **AI + IoT Dashboard**
   Future integration for advanced monitoring.

FINAL_ARDUINO.ino

```
#include <Servo.h>
//#include <NewPing.h>



const int trigPin1 = 10, echoPin1 = 11;
const int trigPin2 = 3, echoPin2 = 4;

const int laser_1 = 13;
const int laser_2 = 6;
int led_R = A0, led_G = A1, buzz = A2;

Servo servo1, servo2;

int angle1 = 0, angle2 = 0;
bool dir1 = true, dir2 = false;  // direction flags

unsigned long lastMoveTime = 0;
const unsigned long moveInterval = 30; // ms between movements

void setup() {
  pinMode(trigPin1, OUTPUT);
  pinMode(echoPin1, INPUT);
  pinMode(laser_1, OUTPUT);

  pinMode(trigPin2, OUTPUT);
  pinMode(echoPin2, INPUT);
  pinMode(laser_2, OUTPUT);
  pinMode(led_R, OUTPUT);
  pinMode(led_G, OUTPUT);
```

Conclusion & Q&A

1

Affordable Prototype

A cost-effective defense solution.

2

Embedded Control

Combines hardware and software.

3

Strong Foundation

Basis for future innovations.

Thank you for your attention. We welcome your questions now.