UGC & Govt. Approved

Sonargaon University (SU)

সোনারগাঁও ইউনিভার্সিটি (এসইউ)

DEVELOPMENT OF AN ARDUINO NANO-BASED ADVANCED AIR DEFENCE SYSTEM FOR THREAT DETECTION AND NEUTRALIZATION

PROJECT & THESIS

Submitted By

Shoeb Akter	ID:ME2201026324
Shuvo Chandra Das	ID:ME2203028296
Md Kamruzzoha	ID:ME2203028326
Shahidul Islam Mokul	ID:ME2203028331
Md Rakib	ID:ME2002021208

Supervisor

Md. Minhaz Uddin

Assistant Professor

Department of Mechanical Engineering

Submitted to the

DEPARTMENT OF MECHANICAL ENGINEERING

SONARGAON UNIVERSITY (SU)

In partial fulfillment of the requirements for the award of the degree

of

BACHELOR OF SCIENCE IN MECHANICAL ENGINEERING

MAY, 2026

DEVELOPMENT OF AN ARDUINO NANO-BASED ADVANCED AIR DEFENCE SYSTEM FOR THREAT DETECTION AND NEUTRALIZATION

A project submitted by

Shoeb Akter

Shuvo Chandra Das

Md Kamruzzoha

Shahidul Islam Mokul

Md Rakib

In partial fulfillment of the requirements for the award of the degree of Bachelor of Science in Mechanical Engineering

Under the supervision of

Md. Minhaz Uddin
Assistant Professor
Department of Mechanical Engineering

DEPARTMENT OF MECHANICAL ENGINEERING
SONARGAON UNIVERSITY (SU)

Spring-2026

Batch:28th

MAY-2026

SONARGAON UNIVERSITY (SU)

Department of Mechanical Engineering

Certificate of Approval

This Is To Certify That The Thesis/Project Titled "Development Of An Advanced Air Defence System For Threat Detection And Neutralization" Has Been Successfully Completed By Shoeb Akter, Shuvo Chandra Das, Md Kamruzzoha, Shahidul Islam Mokul, And Md Rakib Id: Me2201026324, Me2203028296, Me2203028326, Me2203028331 And Me2002021208 Department Of Department Of Mechanical Engineering Sonargaon University (Su) In Partial Fulfillment Of The Requirements For The Degree Of Bachelor Of Science In Mechanical Engineering. The Work Has Been Carried Out Under My/Our Supervision and Is Found to Be Satisfactory in Terms of Quality, Scope, And Originality. This Project Has Not Been Submitted Elsewhere for The Award of Any Other Degree.

……………………………………

Md. Minhaz Uddin

Assistant Professor

Department of Mechanical Engineering

DECLARATION

We do hereby solemnly declare that, the work presented here in this project report has been carried out by us and has not been previously submitted to any University Organization for award of any degree or certificate.

We hereby ensure that the work presented here does not breach any existing copyright. We further undertake to indemnify the university against any loss or damage arising from breach of the foregoing obligation.

Name: Shoeb Akter
ID: ME2201026324

Name: Shuvo Chandra Das
ID: ME2203028296

Name: Md Kamruzzoha
ID: ME2203028326

Name: Shahidul Islam Mokul
ID: ME2203028331

Name: Md Rakib
ID: ME2002021208

ACKNOWLEDGEMENT

First of all, we express our gratitude to Almighty Allah for giving us the strength and ability to complete this thesis successfully. This thesis is accomplished under the supervision of Md. Minhaz Uddin, Assistant Professor, Department of Mechanical, Sonargaon University. It is a great pleasure to acknowledge our profound gratitude and respect to our supervisor for his consistent guidance, encouragement, helpful suggestion, constructive criticism and endless patience through the progress of this work. The successful completion of this thesis would not have been possible without his persistent motivation and continuous guidance.

The authors are also grateful to Prof. Md. Mostofa Hossain, Head of the Department of Mechanical Engineering and all respected teachers of the Mechanical Engineering Department for their co-operation and significant help for completing the thesis work successfully.

Authors:

Shoeb Akter

Shuvo Chandra Das

Md Kamruzzoha

Shahidul Islam Mukul

Md Rakib

ABSTRACT

The Air Defence System is an Arduino-based prototype designed to simulate real-time target detection, tracking, and automated engagement using a simplified radar interface displayed on a laptop screen. The system employs an ultrasonic sensor mounted on a servo motor to scan a wide arc, emitting pulses and measuring the echo time to calculate the distance of approaching objects. The Arduino Nano acts as the central controller, processing the sensor data and sending it to a laptop, where the radar interface visually represents the target's position and movement. When an object enters a predefined critical range, the system triggers a simulated missile launch mechanism to demonstrate automated Defence response. This setup eliminates additional hardware displays or warning buzzers, focusing on software-based visualization for monitoring and tracking. The project highlights the integration of embedded systems, real-time data acquisition, and automated control, providing a cost-effective and educational platform. It serves as a practical demonstration of radar-based surveillance, responsive target tracking, and automated engagement principles, offering insights into modern Defence system concepts in a safe and controlled laboratory environment.

TABLE OF CONTENTS

DECLARATION..ix
ACKNOWLEDGEMENT..v
ABSTRACT..vi
LIST OF FIGURES..ix,
LIST OF TABLE..x

CHAPTER

1. Introduction..1

1.1 Overview 1

1.2 Problem Statement.. 2

1.3 Objectives 3

1.4 Work Flow Diagram..3

1.5 Research Outline... 4

2. Literature Review...5

2.1 Introduction 5

2.2 Literature Review 5

2.3 Summary.. 8

3. System Architecture...9

3.1 Introduction 9

3.2 Block Diagram... 9

3.3 Circuit Diagram 10

3.4 Working Principle.. 10

3.5 Cost Analysis..11

4. Hardware and Software Analysis 12

4.1 Introduction..12

4.2 Hardware & Software 12

4.3 Arduino Nano 12

4.4 SMPS 16
4.5 Servo Motor 20
4.6 Ultrasonic Sensor 22
4.7 Buzzer 25
4.8 Projectile 26
4.9 Arduino IDE 27
4.10 Easy EDA 31
4.11 Complete Project Prototype 32

5. Result Analysis 33
5.1 Result Analysis with Mathematical Data 33
5.2 Advantages 34
5.3 Application 34
5.4 Future Scope of Work 35

6. Future Work and Conclusion 36
6.1 Conclusion 36

References 37
Appendix 40

LIST OF FIGURES

Figure

1.1 Work Flow Diagram 4
3.1 Block Diagram of our Project 9
3.2 Circuit Diagram of our Project 10
4.1 Arduino Nano.. 13
4.2 Arduino Nano Shematic Diagram... 13
4.3 How Arduino Nano Looks Like 14
4.4 Micro Controller IC AT mega 328p 15
4.5 SMPS 16
4.6 SMPS Circuit 17
4.7 SMPS Circuit Connection.. 19
4.8 DC Power Supply Way .... .19
4.9 Servo Motor 20
4.10 Schematic Diagram of Servo Motor 21
4.11 Ultrasonic Sensor Working Procedure.. 23
4.12 Ultrasonic Sensor Pin out.. 23
4.13 Buzzer 25
4.14 Projectile…………………………………………………………………………26
4.15 Arduino Software Interface IDE 28
4.15 Easy EDA Software Interface 31
5.1 Our Final Project 32

LIST OF TABLES

Table

1 List of Component with Price 11

2 Pin Configuration of Ultrasonic Sensor 24

CHAPTER -1

INTRODUCTION

1.1 Overview

In an era marked by technological advancements, radar systems stand at the forefront of technological innovation at the beginning evolved in secrecy through various countries during world conflict, RADAR, or Radio Detection and varying, has evolved into a multifaceted device for item detection. utilizing radio waves, it determines critical parameters such as range, altitude, course, and velocity of items. The coinage of the time period "RADAR" by means of the us navy in 1940 marked a milestone in its nomenclature.

Modern-day radar applications are giant, starting from air traffic control and astronomy to air protection and antimissile structures. It performs a pivotal function in numerous domain names like maritime navigation, plane anti-collision structures, and outer space surveillance. at the coronary heart of missile guidance systems, radar technology is a linchpin for particular targeting and navigation. This project embarks at the ambitious endeavor of developing automatic missile launching gadgets, poised to revolutionize Defence technology and surveillance. Envisaged for deployment throughout navy branches, along with army, army, and aircraft, these devices promise heightened readiness and reaction competencies. precipitated through interruptions in radar indicators, those computerized launchers represent a bounce forward in Defence mechanisms, reflecting the relentless pursuit of innovation to make sure safety in an ever-evolving landscape.

Missile Defence systems play a crucial role in modern security by detecting, tracking, and intercepting incoming threats. Traditional radar-based Defence systems are highly sophisticated but expensive, limiting their accessibility for research and educational purposes. An Arduino-based missile Defence radar system offers a cost-effective alternative, utilizing microcontrollers and radar sensors to detect and track moving objects in real-time. This survey explores existing radar Defence technologies, compares them with Arduino-based implementations, and evaluates their feasibility for low-cost Defence applications. The study aims to highlight advancements, challenges, and future prospects of affordable radar-based threat detection systems.

Several research efforts have focused on developing radar-based Defence systems, ranging from military grade solutions to low-cost experimental models. High-end systems like THAAD, Iron Dome, and Aegis use advanced radar and AI-driven tracking for missile interception. In contrast, Arduino-based radar systems have been explored for small-scale applications, leveraging ultrasonic, Doppler, and microwave sensors for object detection. Studies have demonstrated Arduino's capability to interface with radar modules for motion tracking and distance measurement. Researchers have also integrated machine learning for target classification and IoT for remote monitoring. This section reviews these contributions, comparing methodologies, sensor technologies, and system architectures to assess their relevance for low-cost missile Defence applications.

1.2 Problem Statement

Modern security threats, including unauthorized aerial intrusions and low-altitude projectiles, demand responsive and cost-effective detection and Defence systems. Traditional missile Defence technologies are often expensive, complex, and inaccessible for educational or small-scale applications. There is a need for a simplified, affordable, and functional prototype to demonstrate the core principles of radar-based object detection and interception. The Arduino Missile Defence Radar System addresses this gap by providing a low-cost solution for detecting and tracking incoming objects using ultrasonic sensors and servo motors. It simulates the working of real missile Defence systems by scanning the environment, identifying threats within a specified range, and responding with a simulated countermeasure. This project enhances understanding of radar mechanisms, target tracking, and automated response systems using Arduino technology. It serves as an educational tool for students and hobbyists, promoting hands-on learning in electronics, automation, and Defence simulation while contributing to innovation in low-cost Defence technologies.

1.3 Objectives

- To develop an Arduino-based radar and target detection system using ultrasonic sensors and servo motor technology.
- To implement real-time target tracking and visual monitoring through a laptop-based display interface.
- To simulate an automated Defence response system for educational and surveillance applications.

1.4 Work Flow Diagram

The Air Defence System project follows a structured five-stage workflow to ensure accurate object detection and real-time tracking capabilities. It begins with a detailed study of radar principles, ultrasonic sensing, and target detection techniques. This is followed by the design phase, where schematic layouts are developed to optimize the placement of the ultrasonic sensor, servo mechanism, and data communication setup. Next, essential components like the Arduino board, ultrasonic sensor, servo motor, and display units are collected. In the assembling stage, these components are integrated according to the design to enable smooth scanning and detection. Finally, the complete system is tested for performance and accuracy, and refined to ensure reliable operation before visualizing targets through radar simulation software.

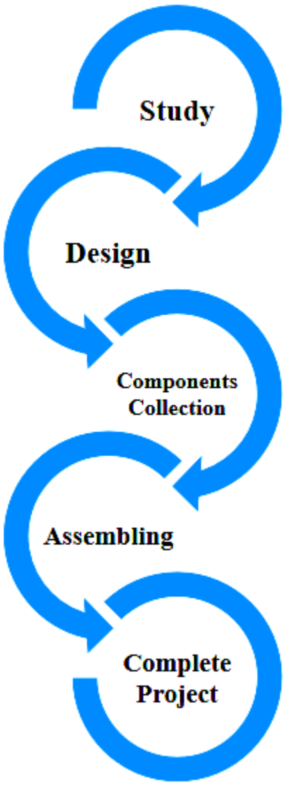

Figure 1.1: Work Flow Diagram

1.5 Research Outline

This project book consists of Six chapter. The first chapter contains the statement of the Introduction, Overview, Problem Statement, Work Flow Diagram, objectives of the study and Methodology. Chapter two contains system Literature e Review details. Chapter three System Architecture, Discussed in Block Diagram, Circuit Diagram, Working principle and Cost analysis. Chapter Four describes the hardware implementation with component details and the software which we have used for our work. Chapter Five deals with the result Analysis, shows the complete prototype, Advantage, Application and Limitation of the project that we have built. In the final chapter, we have discussed the, Future Scope, conclusion of the project.

CHAPTER -2

LITERATURE REVIEW

2.1 Introduction

In this chapter we discuss our system overview. In this paper they also work about Air Defence System. In below section we will describe some previous literature review. From previous literature we gather more knowledge to build this project and we successfully made it.

2.2 Literature Review

D.A. Gheorghe, Ahire Dhanshri, and Ahire Priyanka describe RADAR as an object detection system that operates using microwaves, which are a form of radio waves. These microwaves help in identifying the range, altitude, direction, and speed of objects. The radar system functions by transmitting radio wave pulses from an antenna or radar dish, which then bounce back after hitting an object in their path. Arduino, a versatile single-board microcontroller, simplifies the integration of electronics into multidisciplinary projects. This project aims to develop an efficient and cost-effective radar system that incorporates all the essential functionalities typical of conventional radar technology. [1]

T.V. Karthikeyan and A.K. Kapoor present a missile launcher system that operates based on signals received from a RADAR. The system utilizes an Arduino Uno microcontroller, a servo motor, and an ultrasonic sensor for its functioning. The initial step involves programming the system using prior coding knowledge. This code is first tested through software simulations before being implemented on the hardware via the Arduino Uno. The ultrasonic sensor's movement is controlled by a servo motor attached to it, which rotates at predetermined angles. When an object is detected, the corresponding angle is transmitted as input to the servo motor controlling the missile launcher. [2]

Macias-Ayala, Gonzalo and Lopez Velasquez, Maria, Radar technology has been a cornerstone of United States military air Defence since the 1930s. With microcontroller applications, we can utilize the technology to develop and prototype further on a smaller scale. Ultrasonic radar technology is most applicable in the Defence sector for its clear advantage in detection and versatility. Engineers are responsible for maintaining the successful operation of these systems. Additionally, creating plans for safe/standard procedure and maintainability records. The goal of the project is to enhance the target tracking capabilities by integrating real-time distance measurement and adaptable control of the launching mechanism.[3]

S. Chowdhury, S. Mazumdar, S. Giri, and A. Bhattacharya developed a system that employs an ultrasonic distance sensor to detect the distance of objects, with the collected data processed and displayed on a computer. The sensor, attached to a servo motor, captures polar distance measurements over a 180-degree sweep. The primary components include an Arduino UNO, an HC-SR04 ultrasonic sensor, and a servo motor. Data processing and visualization are managed using the Arduino IDE and Processing software. This system has diverse applications, including air traffic control, maritime navigation, and weather monitoring, highlighting its adaptability for security and mapping purposes. By utilizing open-source hardware and software, the project emphasizes the ease of access and educational benefits of incorporating radar technology into both practical and experimental projects. [4]

M. Tejashwini, C. Rohith, and G.S. Amrutha describe radar as a detection system that utilizes radio waves to identify characteristics of objects, such as their distance and angle. In this project, a radar system was developed using ultrasonic sensors to detect nearby objects. Ultrasonic technology enables precise, non-contact distance measurement, making it ideal for accurate and automated detection. The movement of the sensors is managed using a compact servo motor, which allows for controlled scanning of the surroundings. [5]

T. K. T. Mach et al.,[6] presented an exploration of the advancements in ultrawideband (UWB) radar technology applied to people counting in the Internet of Things (IoT) applications. We introduce a novel lightweight convolutional neural network (CNN) model specifically designed for implementation on microcontrollers with low power consumption. Our proposed model successfully overcomes the limitations of existing methods, achieving an impressive accuracy rate of 99.38% for counting to ten people while randomly walking in a small area of 5×5 m. In addition, its lightweight architecture enables effortless integration with resource-constrained microcontrollers, enabling efficient execution of intelligent IoT tasks. Notably, our method maintains the accuracy even after quantization, with the model retaining 98.22% accuracy while reducing its size by more than half. Benchmarking results demonstrate the model's efficiency, with inference times of less than 48 MS on a wide range of STM32 microcontrollers. Furthermore, the model with the most compact size specification achieves a remarkable inference time of only 3.8 MS per prediction on an STM32 microcontroller.

M. Rajeswari et al.,[7] Radar technology has been a cornerstone of numerous applications, from aviation to meteorology, offering invaluable capabilities for object detection and tracking. Traditional radar systems, while powerful, often come with significant costs and complexities, limiting their accessibility in various fields. In response to these limitations, this research endeavors to design and implement a radar system using Arduino microcontrollers and ultrasonic sensors. This innovative approach leverages the flexibility and affordability of the Arduino platform and the distance measurement precision of ultrasonic sensors to create a radar-like system. The research encompasses the development of the hardware, data acquisition, signal processing, and real-time visualization, providing an accessible and cost-effective alternative for radar-based applications. The study explores the integration of Arduino technology and ultrasonic sensors into radar systems, with a focus on simplicity, affordability, and practicality. By bridging the gap between traditional radar systems and DIY electronics, this research aims to democratize radar technology and open doors for applications in robotics, smart cities, industrial automation, and education.

J. Verastegui et al.,[8] operates several radars for different applications, from the main radar, an incoherent scatter radar used mainly for ionospheric activity observations, to ionosondes and wind profilers. Most of these radars use a centralized modular control system that commands all the radar sequences that require the radar modules, these tasks and sequences are controlled by pulsed digital signals. The device responsible for this operation is called the Radar Controller. A large number of customized Radar Controller versions were developed and built at JRO for decades, since the utilization of its first acquisition system. The current version of the Radar Controller is based on an RTL design written on VHDL language that implements a custom arbitrary waveform generator connected to an SRAM memory that stores all the data a given waveform needs.

P. Kaniewski et al.,[9] presented a concept of a high-accuracy positioning system, using ultrawideband (UWB) radio modules. The system is dedicated for supporting operation of a handheld ground-penetrating radar (GPR), as the use of information about the antenna coordinates at the moments of scanning enables correction of the radar signals and facilitates creation of high-resolution subsurface images. The presented system is self-contained and easy-deployable; it has centimeter-level accuracy and provides positioning data with high repeatability. The system's structure, mathematical model, positioning algorithm and chosen simulation results are presented in the paper.

F. Sickinger et al.,[10] An automotive radar sensor for cocoon functions or automated parking requires very small dimensions to access new mounting positions like B-pillar and side skirts.

To minimize the dimensions of radar sensors, new concepts are necessary. A new system approach for radar sensors is presented. The new radar sensor system is divided in two major units. The sensor unit consists of a small serializer board and Low Temperature Cofired Ceramic (LTCC) miniature frontend. The external radar Electrical Control Unit (ECU) provides the signal processing performance and the power supply for the sensor unit. For the automotive radar band (76-81 GHz), RX- and TX antennas have been simulated, manufactured and the radiation pattern has been measured and a full prototype has been built.

X. Quan et al.,[11] The impulse radio ultra-wideband (IRUWB) radar technology is attracted strong attention for various applications such as crowdedness measurement, building energy management system, vital sign monitoring, and counting the number of inbound and outbound people. In this paper, we propose a shopping store management system based on IR-UWB radar sensors. In order to manage a shopping store, checking the number of entered customers and sweet spot of a store is necessary. Here, sweet spot means the average residence time of customers in one hour. To get these two information, two applications of IR-UWB radar sensor are required. One is counting the number of inbound and outbound people, and the other one is crowdedness measurement. To validate this system, we install this system in a real shopping store. The performance of the shopping store management system is given in this paper, and the results show that our system is applicable.

2.3 Summary

From the literature discussed above, we gained a lot of knowledge and we were inspired to do this project. We were able to do it with everyone tireless work.

CHAPTER -3

SYSTEM ARCHITECTURE

3.1 Introduction

The project objectives, methods, literature evaluation, and other details were all clarified in the preceding chapter. The block diagram, circuit diagram, operating principle, and final project instrument cost analysis will all be covered in this chapter.

3.2 Block Diagram

Here is the block diagram of the Air Defence System with all the essential components. All of the components are shown in below as a block in this diagram. Here we use Arduino Nano for main brain of our project. Other equipment's are attached with this micro-controller and work together for perform as our desire outcome.

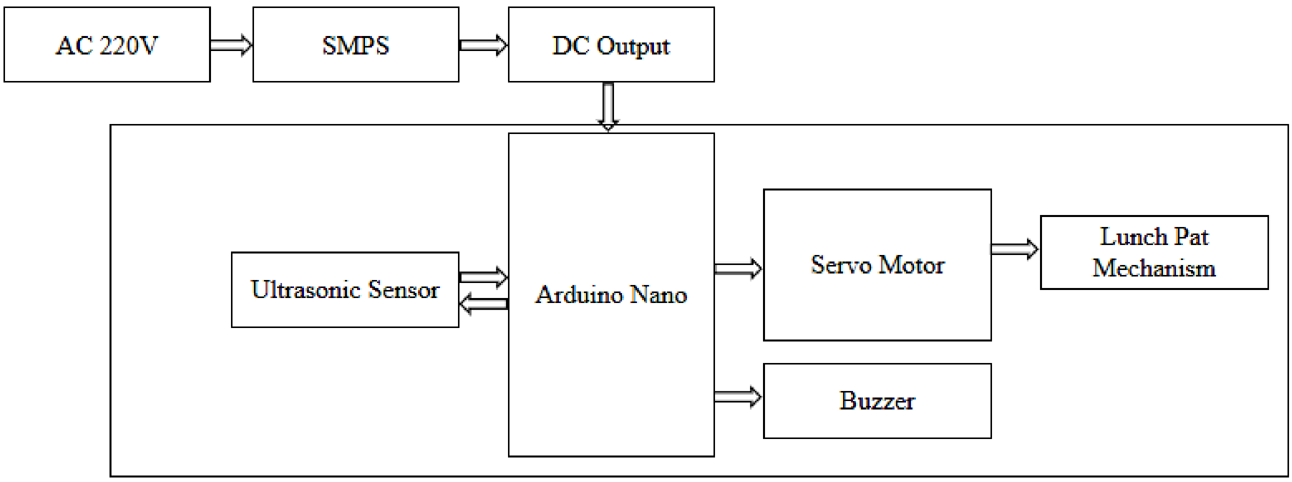

Figure 3.1: Block Diagram of Our Project

3.3 Circuit Diagram

In this part we show our project circuit design and connect out instrument through standard wire.

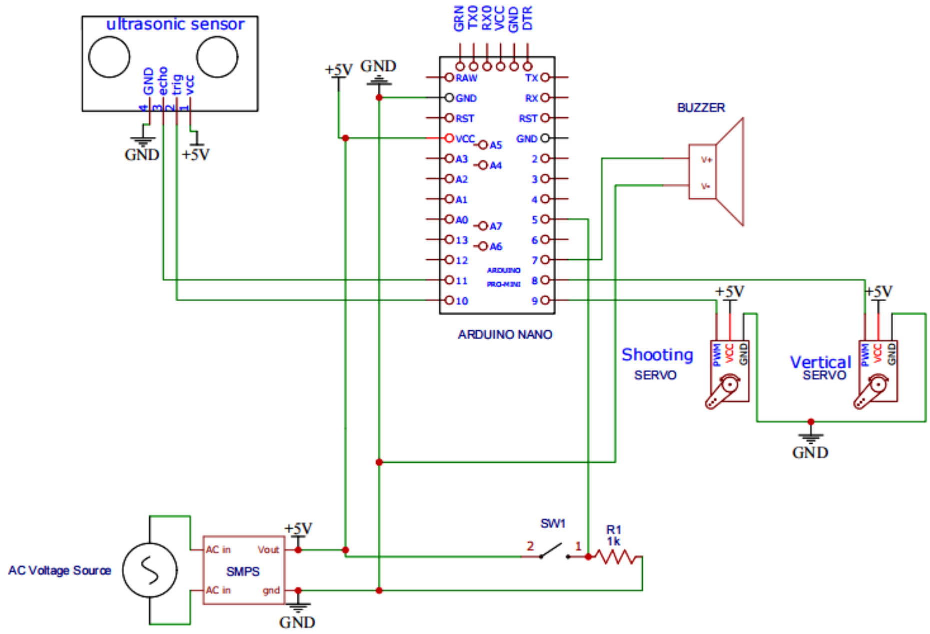

Figure 3.2: Project Circuit Diagram of Our System

3.4 Working Principle

The Air Defence System operates by integrating an Arduino Nano, ultrasonic sensor, servo motor, buzzer, SMPS, and a launch pad mechanism to simulate automated target detection and engagement. The SMPS supplies stable power to all components. The ultrasonic sensor emits high-frequency sound waves and receives reflected echoes from objects. The Arduino Nano calculates the target distance using echo return time. The sensor is mounted on a servo motor that rotates continuously, enabling radar-like scanning of the surrounding area. When an object enters the predefined warning range, the buzzer is activated to provide an audible alert indicating target detection. As the target moves closer into the critical range, the Arduino triggers the launch pad mechanism to simulate missile launch. The servo motor ensures accurate target alignment and smooth tracking. A laptop interface displays real-time target position and movement, demonstrating monitoring, warning, and automated response capabilities.

3.5 Cost Analysis

Table 1: List of Component with Price

Sl.no	Particulars	Specification	Qty.	Unit Price (Taka)	Total Price (Taka)
1	SMPS	5V 7A	1	650	650
2	Arduino Nano	IC AT mega328p	1	700	700
3	Ultrasonic Sensor		1	350	350
4	Servo Motor	SG90	2	450	900
5	Buzzer		1	50	50
6	Others				1200
				Total	3850/=

CHAPTER -4

HARDWARE & SOFTWARE ANALYSIS

4.1 Introduction

In this section, we will discuss elaborately about “Air Defence System” and the component description, features, working procedure of our all equipment. The system hardware fabricates composed of micro-controller unit, power unit, source unit, power store unites, sensor unit and many more related components.

4.2 Hardware & Software

Hardware

- SMPS
- Arduino Nano
- Servo Motor
- Buzzer
- Ultrasonic Sensor
- Projectile

Software

- Arduino IDE
- Easy EDA

4.3 Arduino Nano

Arduino is open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments. Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling Lights, motors, and other actuators.

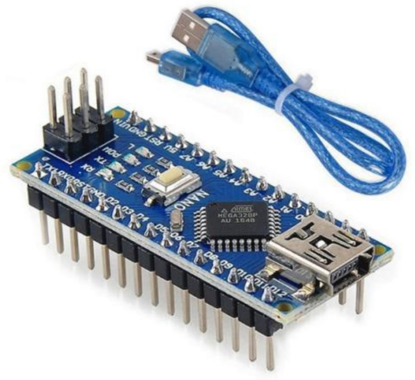

Figure 4.1: Arduino Nano

The micro-controller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, Maxims'). Arduino Nano is a surface mount breadboard embedded version with integrated USB. It is a small, complete, and breadboard friendly component. It has everything that Decimal/ Duemilanove has (electrically) with more analog input pins and onboard +5V AREF jumper. Physically, it is missing power jack. The Nano can automatically sense and switch to the higher potential source of power.

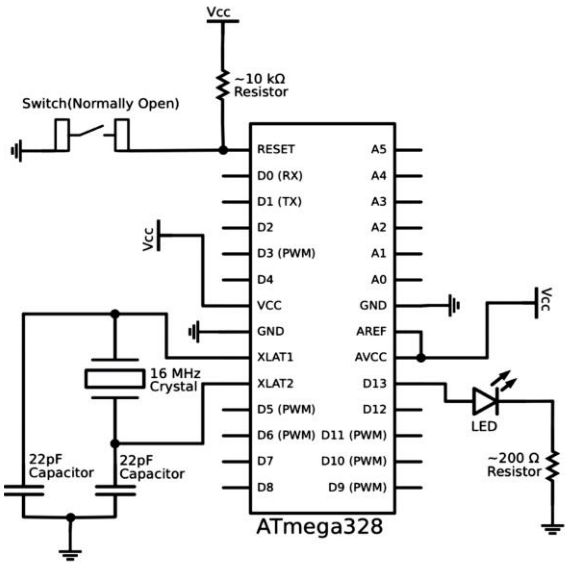

Figure 4.2: Arduino Nano Schematic Diagram

Nanos got the breadboard-ability of the Boarding and the Minibus with smaller footprint than either, so users have more breadboard space. It's got a pin layout that works well with the Mini or the Basic Stamp (TX, RX, ATN, and GND on one top, power and ground on the other).

This new version 3.0 comes with ATMEGA328 which offer more programming and data memory space. It has two layers. That make it easier to hack and more affordable. One of the best features of Arduino Nano is, it's easy to use, compact and also small.

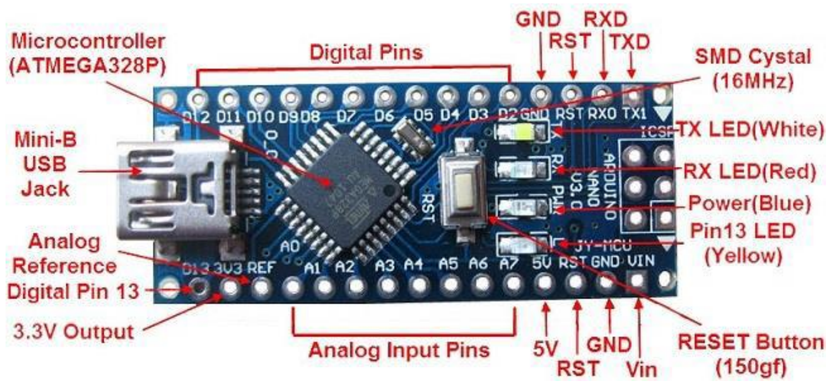

Figure 4.3: How Arduino Nano looks like

Specifications

- Micro-controller: Atmel ATmega328
- Operating Voltage (logic level): 5 V
- Input Voltage (recommended): 7-12 V
- Input Voltage (limits): 6-20 V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- Analog Input Pins: 8
- DC Current per I/O Pin: 40 mA
- Flash Memory: 32 KB (of which 2KB used by boot loader)
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Dimensions: 0.70" x 1.70"

Features

- Automatic reset during program download
- Power OK blue LED
- Green (TX), red (RX) and orange (L) LED

- Auto sensing/switching power input
- Small mini-B USB for programming and serial monitor
- ICSP header for direct program download
- Manual reset switch

Micro-controller IC ATmega328p

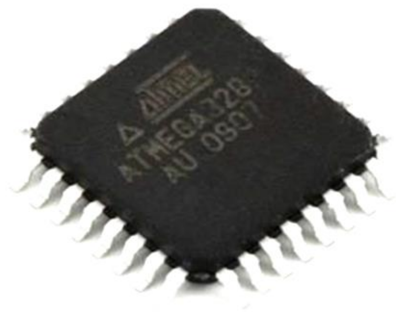

Figure 4.4: Micro-controller IC AT mega 328p

The high-performance Microchip Pico Power 8-bit AVR RISC-based micro-controller combines 32KB ISP flash memory with read-while-write capabilities, 1024B EEPROM, 2KB SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible timer/counters with compare modes, internal and external interrupts, serial programmable USART, a byte-oriented 2-wire serial interface, SPI serial port, a 6-channel 10-bit A/D converter (8-channels in TQFP and QFN/MLF packages), programmable watchdog timer with internal oscillator, and five software selectable power saving modes. The device operates between 1.8-5.5 volts. By executing powerful instructions in a single clock cycle, the device achieves throughput's approaching 1 MIPS per MHz, balancing power consumption and processing speed.

4.4 Switch Mode Power Supply (SMPS)

A switched-mode power supply (switching-mode power supply, switch-mode power supply, switched power supply, SMPS, or switcher) is an electronic power supply that incorporates a switching regulator to convert electrical power efficiently. Like other power supplies, an SMPS transfers power from a DC or AC source (often mains power) to DC loads, such as a personal computer, while converting voltage and current characteristics. Unlike a linear power supply, the pass transistor of a switching-mode supply continually switches between low-dissipation, full-on and full-off states, and spends very little time in the high dissipation transitions, which minimizes wasted energy. A hypothetical ideal switched-mode power supply dissipates no power. Voltage regulation is achieved by varying the ratio of on-to-off time (also known as duty cycles). In contrast, a linear power supply regulates the output voltage by continually dissipating power in the pass transistor. This higher power conversion efficiency is an important advantage of a switched-mode power supply. Switched-mode power supplies may also be substantially smaller and lighter than a linear supply due to the smaller transformer size and weight.

Figure 4.5: SMPS

Switching regulators are used as replacements for linear regulators when higher efficiency, smaller size or lighter weight are required. They are, however, more complicated; their switching currents can cause electrical noise problems if not carefully suppressed, and simple designs may have a poor power factor. Switched-mode power supplies are classified according to the type of input and output voltages. The four major categories are:

- AC to DC
- DC to DC

- DC to AC
- AC to AC

A basic isolated AC to DC switched-mode power supply consists of:

- Input rectifier and filter
- Inverter consisting of switching devices such as MOSFETs
- Transformer
- Output rectifier and filter
- Feedback and control circuit

The input DC supply from a rectifier or battery is fed to the inverter where it is turned on and off at high frequencies of between 20 KHz and 200 KHz by the switching MOSFET or power transistors. The high-frequency voltage pulses from the inverter are fed to the transformer primary winding, and the secondary AC output is rectified and smoothed to produce the required DC voltages. A feedback circuit monitors the output voltage and instructs the control circuit to adjust the duty cycle to maintain the output at the desired level.

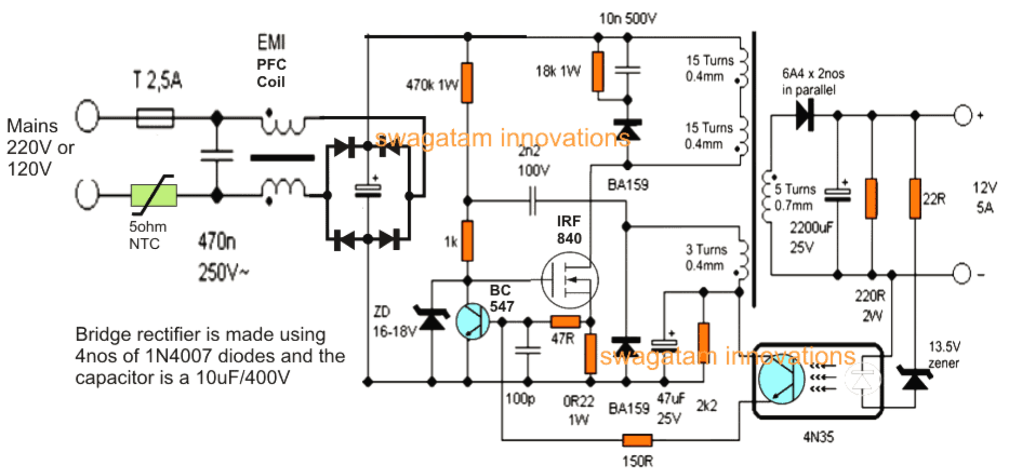

Figure 4.6: SMPS Circuit

Basic working concept of an SMPS

A switching regulator does the regulation in the SMPS. A series switching element turns the current supply to a smoothing capacitor on and off. The voltage on the capacitor controls the time the series element is turned. The continuous switching of the capacitor maintains the voltage at the required level.

Design basics

AC power first passes through fuses and a line filter. Then it is rectified by a full-wave bridge rectifier. The rectified voltage is next applied to the power factor correction (PFC) pre-regulator followed by the downstream DC-DC converter(s). Most computers and small appliances use the International Electrotechnical Commission (IEC) style input connector. As for output connectors and pin outs, except for some industries, such as PC and compact PCI, in general, they are not standardized and are left up to the manufacturer.

There are different circuit configurations known as topologies, each having unique characteristics, advantages and modes of operation, which determines how the input power is transferred to the output. Most of the commonly used topologies such as flyback, push-pull, half bridge and full bridge, consist of a transformer to provide isolation, voltage scaling, and multiple output voltages. The non-isolated configurations do not have a transformer and the power conversion is provided by the inductive energy transfer.

Advantages of switched-mode power supplies

- Higher efficiency of 68% to 90%
- Regulated and reliable outputs regardless of variations in input supply voltage
- Small size and lighter
- Flexible technology
- High power density
- Disadvantages:
- Generates electromagnetic interference
- Complex circuit design
- Expensive compared to linear supplies

Switched-mode power supplies are used to power a wide variety of equipment such as computers, sensitive electronics, battery-operated devices and other equipment requiring high efficiency. Linear voltage IC regulators have been the basis of power supply designs for many years as they are very good at supplying a continuous fixed voltage output. Linear voltage regulators are generally much more efficient and easier to use than equivalent voltage regulator circuits made from discrete components such a Zener diode and a resistor, or transistors and even op-amps. The most popular linear and fixed output voltage regulator types are by far the 78… positive output voltage series, and the 79… negative output voltage series.

These two types of complementary voltage regulators produce a precise and stable voltage output ranging from about 5 volts up to about 24 volts for use in many electronic circuits. There is a wide range of these three-terminal fixed voltage regulators available each with its own built-in voltage regulation and current limiting circuits. This allows us to create a whole host of different power supply rails and outputs, either single or dual supply, suitable for most electronic circuits and applications. There are even variable voltage linear regulators available as well providing an output voltage which is continually variable from just above zero to a few volts below its maximum voltage output.

Switch Mode Power Supply

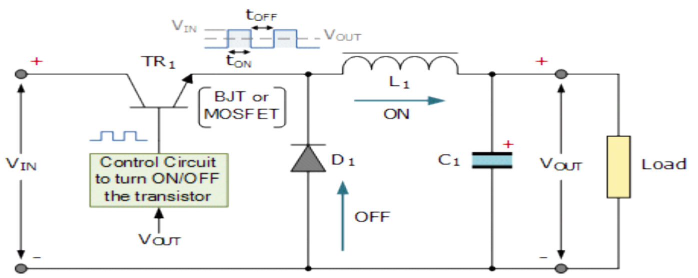

Figure 4.7: SMPS Circuit Connection

Most DC. power supplies comprise of a large and heavy step-down mains transformer, diode rectification, either full-wave or half-wave, a filter circuit to remove any ripple content from the rectified DC. producing a suitably smooth DC. voltage, and some form of voltage regulator or stabilizer circuit, either linear or switching to ensure the correct regulation of the power supplies output voltage under varying load conditions. Then a typical DC. power supply would look something like this:

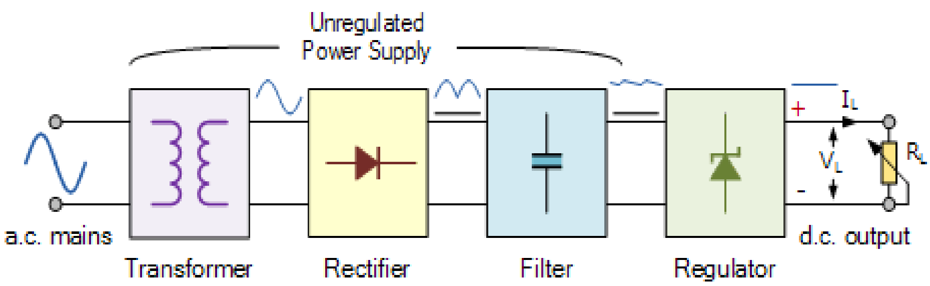

Figure 4.8: DC Power supply way

These typical power supply designs contain a large mains transformer (which also provides isolation between the input and output) and a dissipative series regulator circuit. The regulator circuit could consist of a single Zener diode or a three-terminal linear series regulator to produce the required output voltage. The advantage of a linear regulator is that the power supply circuit only needs an input capacitor, output capacitor and some feedback resistors to set the output voltage.

4.5 Servo Motor

A servo motor is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through servo mechanism. If motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor. We can get a very high torque servo motor in a small and light weight packages. Doe to these features they are being used in many applications like toy car, RC helicopters and planes, Robotics, Machine etc.

Servo motors are rated in kg/cm (kilogram per centimeter) most hobby servo motors are rated at 3kg/cm or 6kg/cm or 12kg/cm. This kg/cm tells you how much weight your servo motor can lift at a particular distance. For example: A 6kg/cm Servo motor should be able to lift 6kg if the load is suspended 1cm away from the motors shaft, the greater the distance the lesser the weight carrying capacity. The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor.

Figure 4.9: Servo Motor

Servo Mechanism

It consists of three parts

- Controlled device
- Output sensor
- Feedback system

All motors have three wires coming out of them. Out of which two will be used for Supply (positive and negative) and one will be used for the signal that is to be sent from the MCU. Servo motor is controlled by PWM (Pulse with Modulation) which is provided by the control wires. There is a minimum pulse, a maximum pulse and a repetition rate. Servo motor can turn 90 degrees from either direction form its neutral position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position, such as if pulse is shorter than 1.5ms shaft moves to 0° and if it is longer than 1.5ms than it will turn the servo to 180°.

Servo motor works on PWM (Pulse width modulation) principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically, servo motor is made up of DC motor which is controlled by a variable resistor (potentiometer) and some gears. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. Potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on required angle.

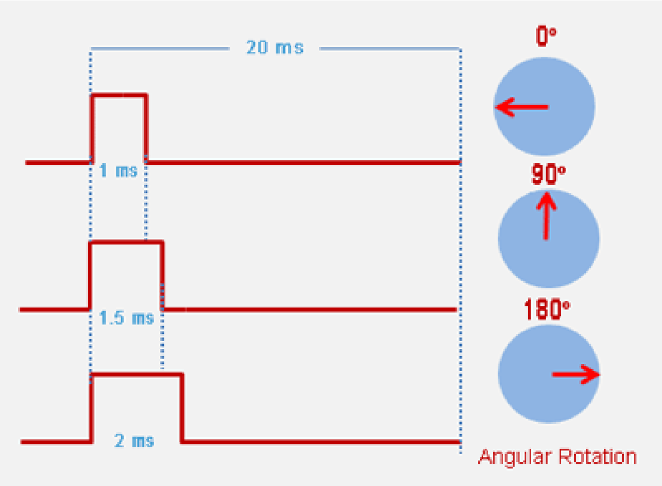

Figure 4.10: Schematic Diagram of servo motor

Servo motor can be rotated from 0 to 180 degree, but it can go up to 210 degrees, depending on the manufacturing. This degree of rotation can be controlled by applying the Electrical Pulse of proper width, to its Control pin. Servo checks the pulse in every 20 milliseconds. Pulse of 1 MS (1 millisecond) width can rotate servo to 0 degree, 1.5ms can rotate to 90 degree (neutral position) and 2 MS pulse can rotate it to 180 degrees. All servo motors work directly with your +5V supply rails but we have to be careful on the amount of current the motor would consume, if you are planning to use more than two servo motors a proper servo shield should be designed.

4.6 Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound.

HC-SR04 Ultrasonic Sensor - Working

As shown above the HC-SR04 Ultrasonic (US) sensor is a 4-pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that –

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below.

Figure 4.11: Ultrasonic Sensor Working Procedure

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave, we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a micro-controller or microprocessor.

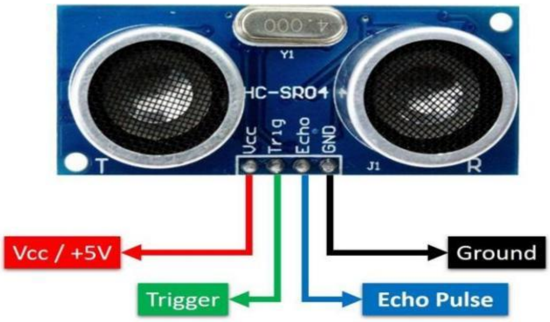

Figure 4.12: Ultrasonic Sensor Pin Out

Ultrasonic Sensor Pin Configuration

Table 2: Pin Configuration of Ultrasonic Sensor

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

HC-SR04 Sensor Features

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

4.7 Buzzer

Piezo elements are actually one of the coolest crystals you'll play with. Flex them, they generate a voltage (pretty high, actually). Pass power through them, and they'll flex. Do it fast enough, and they'll start to make a sound! So besides making annoying little beep noises, you could use it as a sensor to detect a knock or other similar vibration.

Figure 4.13: Buzzer

<u>Specifications</u>

- Operating Voltage: 1.0 ~ 20.0V
- Rated Voltage: 3.0V
- Diameter: Ø12.5mm
- Total Height: 6.3mm
- Capacitance: 15,000pF
- Current Consumption: ≤2mA
- Resonance Frequency: 4,000Hz
- Sound Pressure Level at 10cm: ≥75db
- Self-Drive: No

4.8 PROJECTILE

The projectile used in this project is a lightweight PVC-made prototype missile designed for educational and experimental purposes in the Air Defence System. The structure is fabricated using Polyvinyl Chloride (PVC) material due to its low weight, durability, low cost, and ease of manufacturing. The projectile is mounted on the launch pad mechanism to simulate automated missile launching during target detection.

The projectile features a cylindrical body with a pointed nose cone to improve directional movement and provide a simple aerodynamic structure. A sensor mounting arrangement is integrated on the upper section of the body to support heat detection and target sensing operations. The projectile is designed to identify heat-emitting enemy objects by using thermal or infrared sensing principles. Such targets may include moving vehicles, engines, or other heat-generating sources.

The main purpose of this projectile is to demonstrate the working concept of a heat-seeking missile system in a safe laboratory environment. It does not contain any explosive material or actual weapon mechanism; instead, it serves as a prototype model for understanding missile guidance, target tracking, and automated Defence response systems.

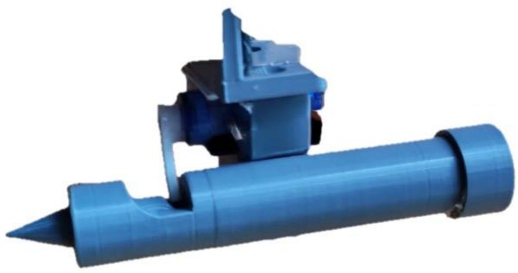

Figure 4.14: PVC Projectile

The projectile system provides the following features:

- Lightweight PVC body construction
- Low-cost and easy fabrication process
- Pointed nose cone for improved movement direction
- Heat-based target detection concept
- Compatibility with automated launch mechanism
- Suitable for educational and research applications

This prototype helps students and researchers understand the basic principles of guided projectile systems, embedded control technology, and Defence simulation techniques. Future improvements may include advanced infrared sensors, wireless communication modules, improved stabilization systems, and intelligent target tracking algorithms for higher accuracy and performance.

SOFTWARE

4.9 Arduino IDE

The digital microcontroller unit named as Arduino Nano can be programmed with the Arduino software IDE. There is no any requirement for installing other software rather than Arduino. Firstly, Select "Arduino Nano from the Tools, Board menu (according to the microcontroller on our board). The IC used to name as ATmega328 on the Arduino Nano comes pre burned with a boot loader that allows us to upload new code to it without the use of an external hardware programmer.

Communication is using the original STK500 protocol (reference, C header files). We can also bypass the boot loader and programs the microcontroller through the ICSP (In Circuit Serial Programming) header. The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU boot loader, which can be activated by:

On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.
The Arduino Nano is one of the latest digital microcontroller units and has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL at (5V) with serial communication, which is available on digital pins 0 -(RX) for receive the data and pin no.1 (TX) for transmit the data. An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an .in file is required.

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board.

The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial Communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Nano's digital pins. The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. Arduino programs are written in C or C++ and the program code written for Arduino is called sketch. The Arduino IDE uses the GNU tool chain and AVR Lab to compile programs, and for uploading the programs it uses argued. As the Arduino platform uses Atmel microcontrollers, Atmel's development environment, AVR Studio or the newer Atmel Studio, may also be used to develop software for the Arduino.

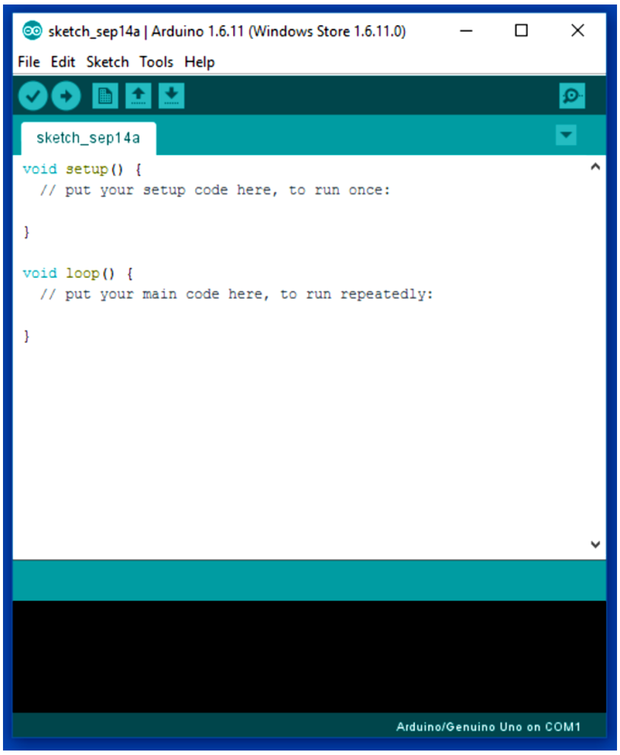

Figure 4.15: Arduino Software Interface IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuine hardware to upload programs and communicate with them.

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension. ion. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Sketchbook

The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook. You can view or change the location of the sketchbook location from with the Preferences dialog. Beginning with version 1.0, files are saved with a .ino file extension. Previous versions use the. pde extension. You may still open. pde named files in version 1.0 and later, the software will automatically rename the extension to .ino.

Tabs, Multiple Files, and Compilation

Allows you to manage sketches with more than one file (each of which appears in its own tab). These can be normal Arduino code files (no visible extension), C files (.c extension), C++ files (.cpp), or header files (.h).

Uploading

Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. The boards are described below. On the Mac, the serial port is probably something like /dev/tty.usbmodem241 (for an Uno or Mega2560 or Leonardo) or /dev/tty.usbserial-1B1 (for a Duemilanove or earlier USB board), or /dev/tty.USA19QW1b1P1.1 (for a serial board connected with a Keyspan USB-to-Serial adapter). On Windows, it's probably COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. On Linux, it should be /dev/ttyACMx , /dev/ttyUSBx or similar. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. With older boards (pre-Diecimila) that lack auto-reset, you'll need to press the reset button on the board just before starting the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

When you upload a sketch, you're using the Arduino bootloader, a small program that has been loaded on to the microcontroller on your board. It allows you to upload code without using any additional hardware. The bootloader is active for a few seconds when the board resets; then it starts whichever sketch was most recently uploaded to the microcontroller. The bootloader will blink the on-board (pin 13) LED when it starts (i.e. when the board resets).

Libraries

Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code.

There is a list of libraries in the reference. Some libraries are included with the Arduino software. Others can be downloaded from a variety of sources or through the Library Manager. Starting with version 1.0.5 of the IDE, you do can import a library from a zip file and use it in an open sketch. See these instructions for installing a third-party library.

Third-Party Hardware

Support for third-party hardware can be added to the hardware directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the hardware directory, then unzip the third-party platform into its own sub-directory. (Don't use "Arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

For details on creating packages for third-party hardware, see the Arduino IDE 1.5 3rd party Hardware specification.

Serial Monitor

This displays serial sent from the Arduino or Genuine board over USB or serial connector. To send data to the board, enter text and click on the "send" button or press enter. Choose the baud rate from the drop-down menu that matches the rate passed to Serial. Begin in your sketch. Note that on Windows, Mac or Linux the board will reset (it will rerun your sketch) when you connect with the serial monitor. Please note that the Serial Monitor does not process control characters; if your sketch needs a complete management of the serial communication with control characters, you can use an external terminal program and connect it to the COM port assigned to your Arduino board.

4.10 Easy EDA

Easy EDA is a web-based EDA tool suite that enables hardware engineers to design, simulate, share-publicly and privately-and discuss schematics, simulations and printed circuit boards. Other features include the creation of a bill of materials, Gerber files and pick and place files and documentary outputs in PDF, PNG and SVG formats. Easy EDA allows the creation and editing of schematic diagrams, SPICE simulation of mixed analogue and digital circuits and the creation and editing of printed circuit board layouts and, optionally, the manufacture of printed circuit boards.

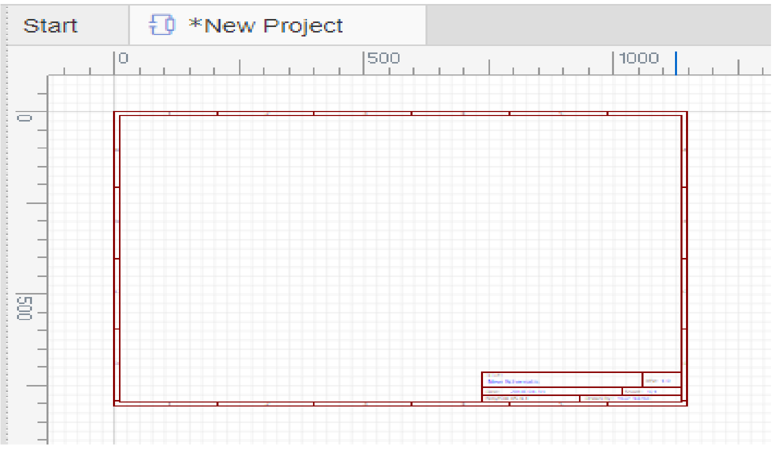

Figure 4.15: Easy EDA Software Interface.

Subscription-free membership is offered for public plus a limited number of private projects. The number of private projects can be increased by contributing high quality public projects, schematic symbols, and PCB footprints and/or by paying a monthly subscription. Registered users can download Gerber files from the tool free of charge; but for a fee, Easy EDA offers a PCB fabrication service. This service is also able to accept Gerber file inputs from third party tools.

4.11 Complete Project prototype

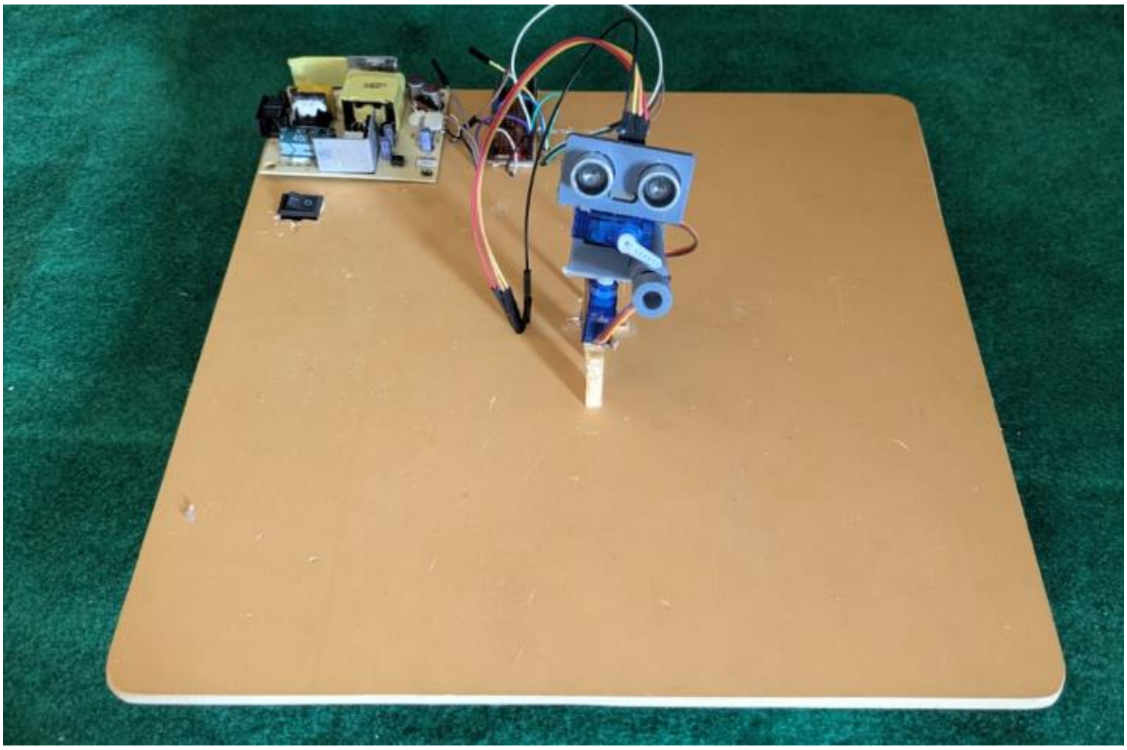

Figure 5.1: Our final project

CHAPTER -5

RESULT ANALYSIS

5.1 Result Analysis with Mathematical Data

The system was tested at different object distances to analyze detection accuracy and response time. The ultrasonic sensor successfully detected targets within the effective range and the servo motor completed smooth scanning operation.

Distance Calculation

Formula: $D = (V \times T) / 2$

Where:

- • D = Distance (cm)
- • V = Speed of sound (0.034 cm/µs)
- • T = Echo time (µs)

Example Calculation:
$D = (1176 \times 0.034) / 2 = 20$ cm

Experimental Results

Calculated Distance (cm)	Echo Time (µs)	System Response
20 cm	1176	Missile simulation activated
24 cm	1411	Warning detection

Scanning Performance
Scanning Angle = 180°
Delay per step = 31 ms
Total scan time = 10 seconds
Full forward-backward scan ≈ 20 seconds

Power Analysis
Total Current = 50mA + 15mA + 1000mA + 2mA = 1067mA ≈ 1.07A
Power Consumption = $V \times I = 5 \times 1.07 = 5.35$W
The analysis shows that the system operated accurately with low power consumption and reliable real-time detection performance.

5.2 Advantages

There are certainly many advantages of our project and some of the major ones have been given below:

- Cost-effective system
- Real-time target detection
- 180-degree scanning capability
- Compact and user-friendly design
- Low power consumption
- Easy prototyping and maintenance
- Arduino-based automated control
- Educational and research friendly

5.3 Application

The application areas for this project in this modern and practical world are huge and some of these are given below:

- Defence training simulations
- Object detection experiments
- Surveillance and monitoring prototypes
- Educational demonstrations of radar technology

5.4 Discussion

While working on our project, we did face some difficulties as it is a very complex system but the end results, we came up with were quite satisfactory. We have put the whole system through several tasks to validate our work and also have taken necessary notes for future improvements. Some future recommendations that we have involves improvement in system design and wiring, adding features for more efficient.

The Air Defence System project demonstrates the practical application of embedded systems for automated target detection and response. By integrating an Arduino Nano with an ultrasonic sensor and servo motor, the system effectively simulates radar-based area scanning and target tracking. The inclusion of a buzzer enhances situational awareness by providing an early warning before engagement. The launch pad mechanism represents automated Defence action in a safe, simulated manner. Real-time visualization through a laptop interface improves monitoring and analysis. Overall, the project highlights automation, control accuracy, and real-time decision-making, making it a valuable educational model for understanding modern Defence and surveillance technologies.

5.5 Future Scope of Work

- Real-field Defence and surveillance applications.
- Integration of renewable energy power systems.
- Wireless remote monitoring and control.
- GPS-based real-time tracking.
- Multi-target detection and tracking system.

CHAPTER -6

CONCLUSION

6.1 Conclusion

The Air Defence System project successfully demonstrates a simplified and safe simulation of modern automated Defence technology using embedded systems. By integrating an Arduino Nano, ultrasonic sensor, servo motor, buzzer, SMPS, and a launch pad mechanism, the system is able to detect, track, and respond to targets in real time. The radar-like scanning achieved through servo-controlled ultrasonic sensing provides effective area monitoring, while distance calculation enables accurate target identification within predefined ranges. The buzzer adds an important early-warning feature, improving system awareness before engagement.

This project highlights automation, precision, and reliability in Defence applications. The SMPS ensures stable power, while the servo provides accurate tracking alignment. The simulated missile system shows how sensors can trigger automatic responses without human input. The laptop interface enables real-time monitoring and better system understanding. Overall, it is a strong learning platform for sensor integration, microcontroller programming, and control systems, with future scope for AI tracking and advanced surveillance.

REFERENCES

[1] D.A. Gheorghe, Ahire Dhanshri, and Ahire Priyanka, *"Radar System Using Arduino"*, presented at the National Conference on Emerging Trends in Engineering & Technology, pp. 53-56.

[2] T.V. Karthikeyan and A.K. Kapoor, *"Guide Missiles"*, Defence Research and Development Laboratory, Hyderabad.

[3] G. Macias-Ayala and M. Lopez Velasquez, *"Automatic Radar Defence System"*, 2024, Engineering – Undergraduate Research Projects. Available at: https://trace.tennessee.edu/utk-enginunder/5

[4] S. Chowdhury, S. Mazumdar, S. Giri, and A. Bhattacharya, *"Radar System Utilizing Arduino"*, published on ResearchGate, 2024.

[5] M. Tejashwini, C. Rohith, and G.S. Amrutha, *"A Short-Range Radar Using Arduino"*, *Journal of Engineering*, 2023.

[6] T. K. T. Mach, C. T. Pham and M. Le, "UWB Impulse Radar for People Counting With Convolutional Neural Network on Microcontrollers," *in IEEE Sensors Journal, vol. 24, no. 9, pp. 15643- 15650, 1 May1, 2024, doi: 10.1109/JSEN.2024.3380571.*

[7] N.Rajeswari, J. Lingappa, M. G. Basha, V. Deepa, V. Sravani and K. Srilatha, "To Design and Implement a Radar System using Arduino and Ultrasonic Sensor," *2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp. 208-212, doi: 10.1109/ICACRS58579.2023.10404253.*

[8] J. Verastegui, I. Manay, E. Pacheco and M. Milla, "Design of a Programmable Radar Controller ASIC on VHDL for a Modular Radar System," *2021 IEEE International Conference on Aerospace and Signal Processing (INCAS), Lima, Peru, 2021, pp. 1-4, doi: 10.1109/INCAS53599.2021.9666921.*

[9] P. Kaniewski, T. Kraszewski and P. Pasek, "UWBBased Positioning System for Supporting Lightweight Handheld Ground-Penetrating Radar," *2019 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS), Tel-Aviv, Israel, 2019, pp. 1-4, doi: 10.1109/COMCAS44984.2019.8958144.*

[10] F. Sickinger, C. Sturm, L. Janda, O. Stejskal and M. Vossiek, "Automotive Satellite Radar Sensor System based on an LTCC Miniature Frontend," *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), Munich, Germany, 2018, pp. 1-4, doi: 10.1109/ICMIM.2018.8443535.*

[11] X. Quan, D. H. Yim and S. H. Cho, "Shopping store management system based on IR-UWB radar sensors," *2016 IEEE International Conference on Network Infrastructure and Digital Content (ICNIDC), Beijing, China, 2016, pp. 481-485, doi: 10.1109/ICNIDC.2016.7974621.*

[12] A. Kumar and R. Singh, "Arduino Based Ultrasonic Radar System for Object Detection*," International Journal of Innovative Research in Science and Engineering, vol. 10, no. 4, pp. 112-118, 2022.*

[13] S. Patel, R. Mehta and K. Sharma, "Smart Surveillance Radar Using Arduino and Ultrasonic Sensors," *International Conference on Smart Electronics and Communication, 2023, pp. 245-250.*

[14] M. Hasan, F. Rahman and T. Ahmed, "Development of a Low-Cost Radar System for Security Applications*," Journal of Embedded Systems and Applications, vol. 8, no. 2, pp. 55-61, 2021.*

[15] R. K. Gupta and P. Verma, "Real-Time Object Tracking System Using Arduino
and Servo Motor*," International Journal of Engineering Research & Technology (IJERT), vol. 11, no. 6, pp. 321-326, 2022.*

[16] L. Wang and Y. Chen, "Microcontroller-Based Radar Monitoring System for Indoor Surveillance*," IEEE International Conference on Consumer Electronics,2020, pp. 98-102, doi: 10.1109/ICCE46568.2020.9043175.*

[17] D. Roy and S. Dutta, "Implementation of Ultrasonic Sensor-Based Radar for Obstacle Detection*," International Journal of Advanced Research in Computer and Communication Engineering, vol. 9, no. 8, pp. 14-19, 2020.*

[18] P. Sharma and V. Tiwari, "IoT Enabled Radar Security System Using Arduino," Proceedings of the *International Conference on Automation and Computing* Technologies*, 2021, pp. 201-206.*

[19] H. Lee, J. Kim and S. Park, "Low-Power Embedded Radar System for Smart Monitoring*," IEEE Sensors Conference, 2022, pp. 1-4, doi: 10.1109/SENSORS52175.2022.9967342.*

[20] M. Ali, S. Hossain and N. Karim, “Arduino Based Defence Surveillance Prototype with Real-Time Visualization*,” International Journal of Scientific Research in Engineering and Technology, vol. 7, no. 5, pp. 88-94, 2023.*

[21] K. Ramesh and P. Kumar, “Design and Analysis of Radar Systems Using Embedded Controllers*,” Journal of Electronics and Communication Engineering, vol. 15, no. 3, pp. 44-50, 2021.*

[22] S. Ibrahim and A. Noor, “Automated Object Detection and Tracking Using Ultrasonic Radar*,” 2022 International Conference on Robotics and Artificial Intelligence, Kuala Lumpur, Malaysia, 2022, pp. 132-137.*

[23] T. Nguyen and V. Tran, “Embedded Radar Prototype for Smart Defence Applications*,” IEEE Asia-Pacific Conference on Applied Electromagnetics, 2021, pp. 77-81, doi: 10.1109/APACE51902.2021.9689921.*

[24] J. Peterson and R. Clarke, “Radar Signal Processing Techniques for Low-Cost Surveillance Systems*,” International Journal of Radar Engineering, vol. 12, no. 1, pp. 23-31, 2020.*

[25] A. Islam, M. Sarker and R. Hasan, “Real-Time Radar Visualization System Using Processing IDE and Arduino*,” International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 2023, pp. 415-420, doi: 10.1109/ICCIT60459.2023.10441278.*

APPENDIX

<u>Micro-Controller Programming:</u>

```
#include <Servo.h>

const int trigPin = 8;
const int echoPin = 9;
const int switchPin = 7;  // Switch connected here

long duration;
int distance;

Servo servo1; // Scanning servo (original)
Servo servo2; // Mirror servo
Servo servo3; // Alert servo - rotates 90° on detection

const int detectionThreshold = 30; // cm - distance to trigger servo3 rotation

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(switchPin, INPUT_PULLUP);  // Switch connects to GND when ON
  Serial.begin(9600);

  servo1.attach(10);
  servo2.attach(11);
  servo3.attach(12);
  servo3.write(0);
}

void loop() {
  bool isOn = digitalRead(switchPin) == HIGH;

  if (isOn) {
    // Scan forward
    for (int i = 15; i <= 165; i++) {
      if (digitalRead(switchPin) == LOW) break; // Stop scanning if switch is turned OFF

      servo1.write(i);
      servo2.write(i);
      delay(30);
      distance = calculateDistance();

      if (distance > 0 && distance < detectionThreshold) {
        servo3.write(90);
      } else {
        servo3.write(0);
      }
```

```
      Serial.print(i);
      Serial.print(",");
      Serial.print(distance);
      Serial.print(".");
    }

    // Scan backward
    for (int i = 165; i > 15; i--) {
      if (digitalRead(switchPin) == LOW) break; // Stop scanning if switch is turned OFF

      servo1.write(i);
      servo2.write(i);
      delay(30);
      distance = calculateDistance();

      if (distance > 0 && distance < detectionThreshold) {
        servo3.write(90);
      } else {
        servo3.write(0);
      }

      Serial.print(i);
      Serial.print(",");
      Serial.print(distance);
      Serial.print(".");
    }
  } else {
    // Switch OFF: stop all servos
    servo1.write(0);
    servo2.write(0);
    servo3.write(0);
    delay(100); // Small delay to reduce CPU usage
  }
}

int calculateDistance() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

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

  return distance;
}
```