ULTRASONIC RADAR USING ESP32

A MINI-PROJECT REPORT

Submitted by

SENTHILNAATHAN K	**210701238**
SHAMINI P N	**210701239**

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE,

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2024

BONAFIDE CERTIFICATE

Certified that this project **"ULTRASONIC RADAR USING ESP32"** is the bonafide work of **"SENTHILNAATHAN K (210701238), SHAMINI P N (210701239)** who carried out the projectwork under my supervision.

SIGNATURE
Dr. P. Kumar, M.E., Ph.D.,

HEAD OF THE DEPARTMENT
PROFESSOR
Dept. of Computer Science and Engg,
Rajalakshmi Engineering College,
Chennai - 602105

SIGNATURE
Mrs. Anitha, M.Tech.,

SUPERVISOR
PROFESSOR
Dept. of Computer Science and Engg,
Rajalakshmi Engineering College,
Chennai - 602105

Submitted to Project Viva-Voce Examination held on ______________________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

This project introduces an IoT-based radar system leveraging an ultrasonic sensor and a servo motor, both controlled by an ESP32 microcontroller. The system integrates with the Blynk application for real-time monitoring and control, while data visualization is achieved using the Processing IDE. Designed to be cost-effective, efficient, and user-friendly, this radar system addresses various applications including obstacle detection, distance measurement, and surveillance.

The ultrasonic sensor, mounted on the servo motor, scans the environment by rotating, while the ESP32 processes and transmits the collected data to the Blynk app over Wi-Fi. This setup allows users to monitor radar readings in real-time from anywhere. The Processing IDE provides a graphical representation of the radar's scanning area, displaying detected objects and their distances.

By combining IoT technology with traditional sensor systems, this project offers an innovative solution that enhances remote monitoring and control capabilities. Its potential applications extend to smart home and security systems, where such a radar system can significantly improve automation and safety. Overall, this project showcases the practicality of integrating IoT with sensor technology to create accessible and versatile solutions for everyday use.

ACKNOWLEDGEMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our Chairman **Mr. S. Meganathan, B.E., F.I.E**., for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan, Ph.D.,** for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan, B.E., M.S.,** for providing us with the requisite infrastructure. We also express our sincere gratitude to our college Principal**, Dr. S.N. Murugesan M.E., PhD.,** and **Dr. Kumar M.E., Ph.D.,** Head of the Department of Computer Science and Engineering and our project guide **Mrs. Anitha, M.Tech.,** for his encouragement and guiding us throughout the project and to our parents,friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

SENTHILNAATHAN K
SHAMINI P N

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

An ultrasonic radar monitoring system using Arduino is a project that employs sound waves to detect the presence and measure the distance of objects. It functions by emitting high-frequency sound waves from an ultrasonic sensor, which bounce off objects and return to the sensor. The Arduino, a small, programmable microcontroller, calculates the distance based on the time it takes for the sound waves to travel to the object and back. This information is then displayed on a screen or can be used to trigger other actions, like sounding an alarm or sending a notification.

if we focus on the software we use two software one name is Arduino IDE, with the help of this we embed the code in our arduino and the other name is Processing app which is used for the processing of our project. Now next comes to its implementation as we know that in Arduino there are fourteen digital pins, eight analogue pins, one pin, one set reset, one analog reference, one 3v, one 5v and three pins are ground and if we see in the side of Ultrasonic sensor there are four pins total that is echo, trig,Vcc and ground. Also there are three connections in our servo motor signal, negative voltage and positive voltage. Then we come to the working of our project where all the connections finish and our ultrasonic start working means start to rotate with the help of the servo motor. If there is no object in the range of the radar than it will show only a green line and green graph in the pc screen, but if we put some object in the range of the radar means an ultrasonic sensor so, when it comes ahead of the object than it will show a red line in the graph with distance, range and angle of the object. So, our project is basically used in local path areas to detect the object distance, range and angle.

CHAPTER 2

LITERATURE SURVEY

The evolution and research efforts in radar have been enormously successful, and have vitally changed computing.[1] a radar system based on the Internet of Things (IoT) that locates objects within 150 degrees by using ultrasonic sensors. A servo motor drives the system's rotation, and the results of detection are shown on a Graphical User Interface (GUI) created with Java Swing and applets. In addition, the system comes with a feature that notifies a user's mobile device when an object is observed. The suggested system is capable of efficiently and effectively detecting things in its immediate surroundings in a range of circumstances, including environmental monitoring and security surveillance. The proposed IoT-based radar system is a desirable addition to the present line of IoT devices since it can usually improve object detection skills and provide users with real-time notifications.

. [2] This innovative method uses the precision of ultrasonic sensors for distance observation and the ability to adapt and affordability of the Arduino platform to create a system that functions similarly to a radar. The research gives a cost-effective substitute for radar-based applications by addressing hardware development, collecting data, signal processing, and real-time visualization. The study investigates how Arduino technology and ultrasonic sensors might be integrated into radar systems, highlighting ease of use, affordability, and simplicity.

There are some ways to show radar working data. There are also some modified radar systems that have advance technology of handling the systems. These modified system are used at higher levels to get or extract the helpful or important data.

[3] The primary objectives of this project are to develop a sophisticated distance measurement system using ultrasonic sensors. This system will be designed to accurately measure distances in various environments and conditions. To achieve

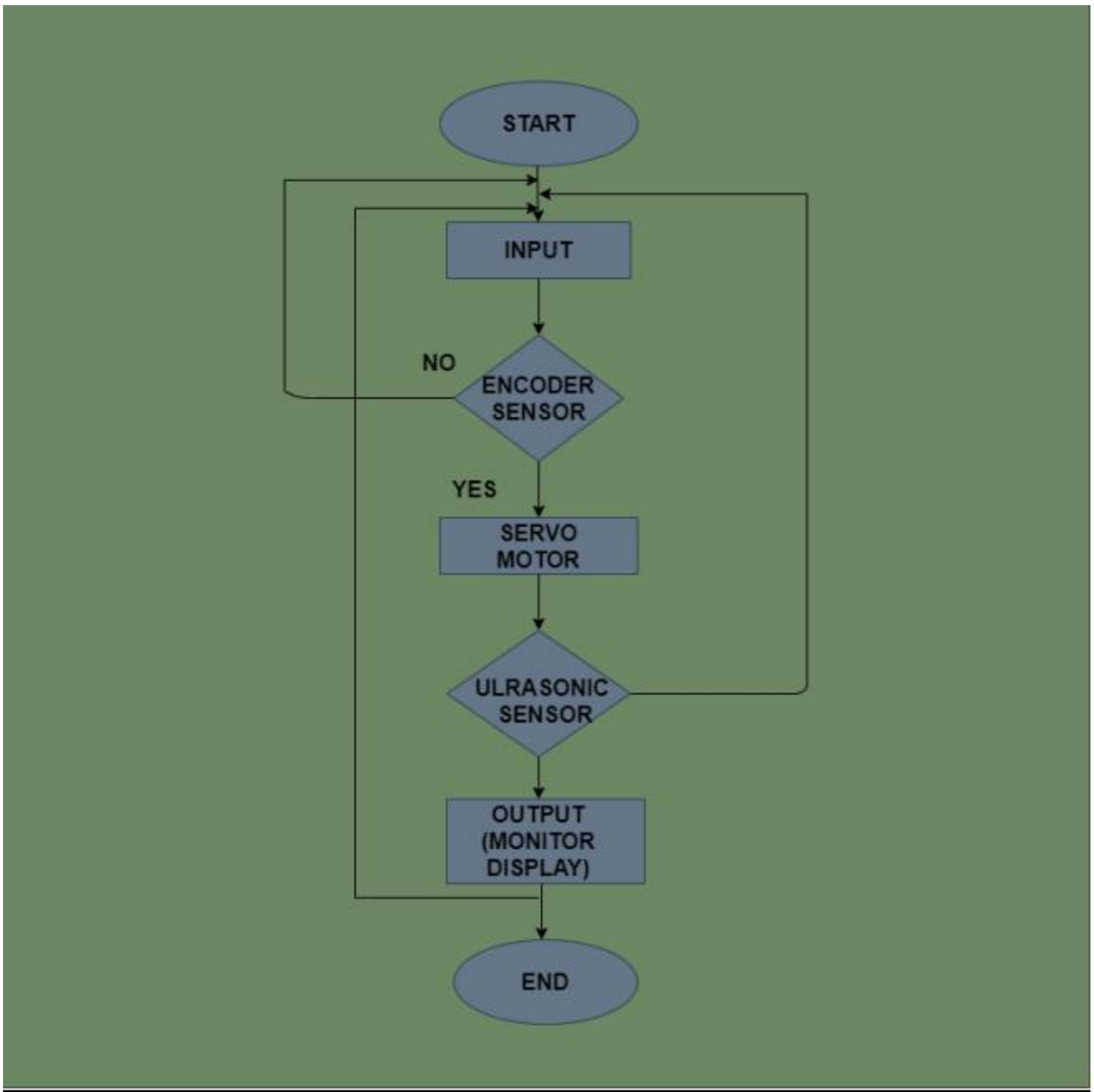
this, an Arduino-based control system will be implemented to handle data acquisition and processing. The system will incorporate advanced signal processing techniques to ensure precise and reliable distance measurements. In addition, the project aims to create a user-friendly graphical interface on a computer to visualize the distance measurements. This interface will provide real-time updates and allow for easy interpretation of the data. The graphical representation will enhance the usability of the system, making it accessible to users with varying levels of technical expertise. Furthermore, the system will be enhanced to provide real-time monitoring of objects and their distances. This will involve integrating the system with advanced monitoring and alerting capabilities, allowing for proactive identification of changes in distance measurements and potential object detection. The real-time monitoring feature will significantly expand the practical applications of the system, enabling its use in scenarios where timely and accurate distance measurements are crucial.

2.1 EXISTING SYSTEM

Current radar systems are generally complex and costly, often using advanced technology for applications like air traffic control, weather monitoring, and security. These systems can be difficult to set up and maintain, making them impractical for small-scale or budget-conscious projects. Moreover, many traditional radar systems lack easy-to-use interfaces and remote monitoring capabilities, which limits their accessibility and convenience for everyday users and simple IoT applications. This project aims to address these limitations by providing a straightforward, cost-effective alternative using readily available components.

CHAPTER 3

PROJECT DESCRIPTION



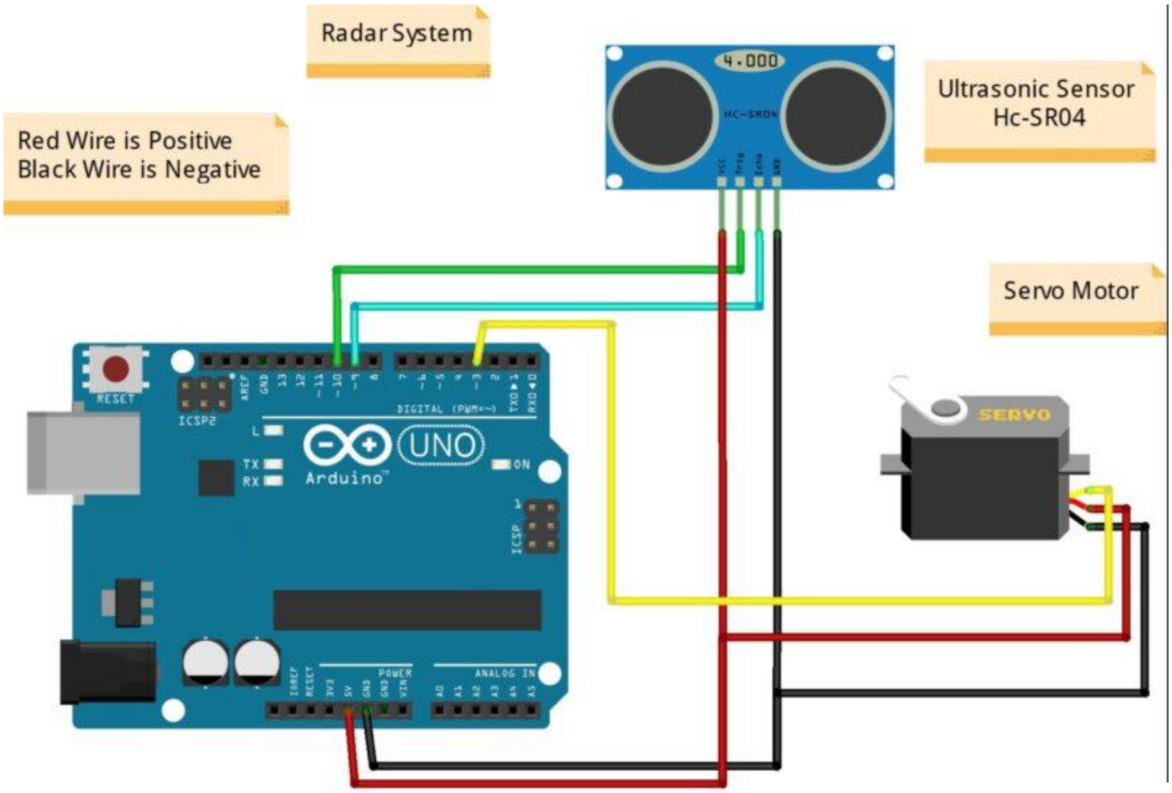
This project involves the design and implementation of a radar system utilizing an ultrasonic sensor and a servo motor, both controlled by an ESP32 microcontroller. The ultrasonic sensor is strategically mounted on the servo motor, allowing it to rotate and scan the surrounding environment for objects. As the servo motor rotates, the ultrasonic sensor continuously measures the distance to any objects in its path.

The ESP32 microcontroller, serving as the central processing unit, collects the data from the ultrasonic sensor. This data is then processed and sent to the Blynk application over a Wi-Fi connection, enabling real-time monitoring and control. The Blynk application allows users to view the radar readings remotely, making it convenient for various applications such as home security and automation.

In addition to the Blynk application, the data is visualized using the Processing IDE. This software creates a radar-like display, graphically representing the scanned area and highlighting the detected objects along with their respective distances. This visual representation helps users to easily understand and interpret the data collected by the radar system.

By combining these components, the project aims to create a cost-effective, user-friendly radar system suitable for a wide range of applications including obstacle detection, distance measurement, and environmental scanning. This system leverages the capabilities of the ESP32 microcontroller and IoT technologies to provide an accessible solution for real-time monitoring and data visualization.

3.1 PROPOSED SYSTEM



The proposed system leverages the ESP32 microcontroller's Wi-Fi capabilities to create an IoT-enabled radar system. The ultrasonic sensor measures distances to objects, and the servo motor rotates the sensor to cover a wide area. The ESP32 collects the sensor data and transmits it to the Blynk application, where users can monitor the radar's readings in real-time. The Processing IDE is used to visualize the data, providing a graphical representation of the radar's scanning area and detected objects. This system is cost-effective, easy to set up, and provides remote monitoring capabilities, making it suitable for various applications such as home security and automation.

3.2 REQUIREMENTS

3.2.1 HARDWARE REQUIREMENTS

- ESP 32
- Power Supply
- Wi-Fi
- Ultrasonic sensor
- Servo motor
- Jumper wires
- Laptop
- Mobile device

3.2.2 SOFTWARE REQUIREMENTS

- Arduino IDE
- Processing IDE
- Blynk IoT

3.3 ARCHITECTURE DIAGRAM

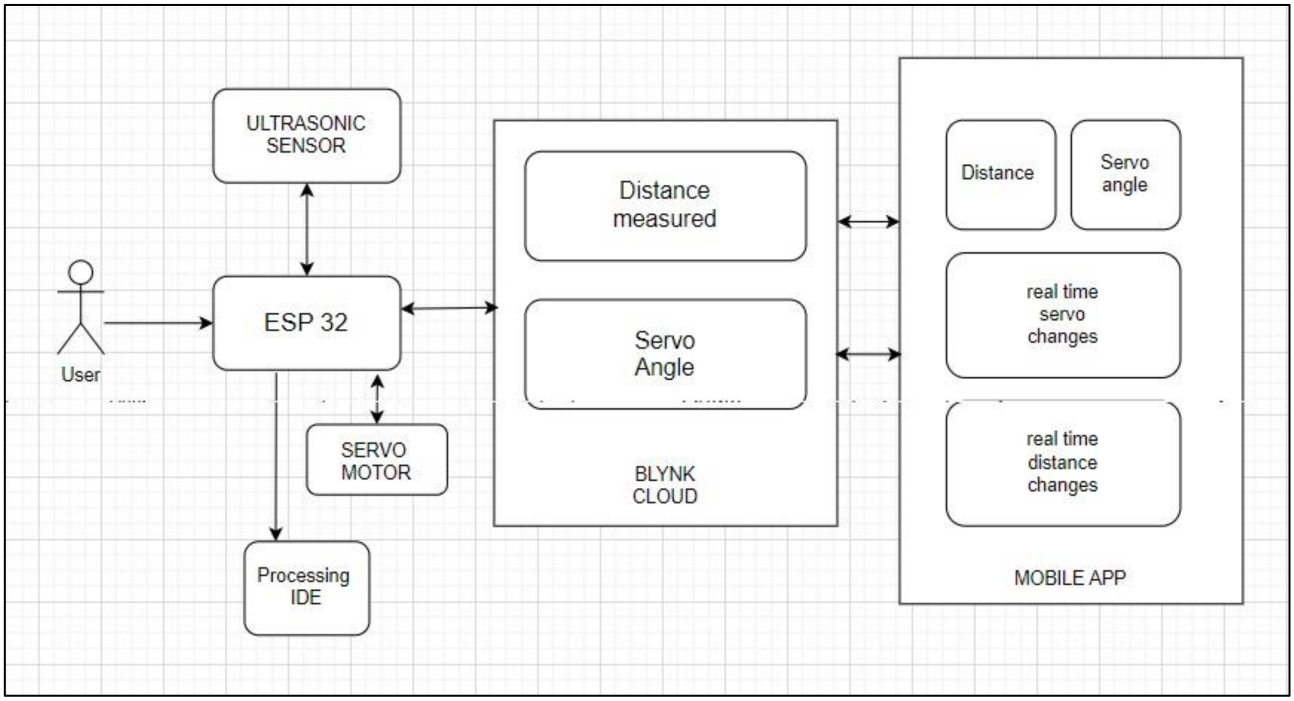
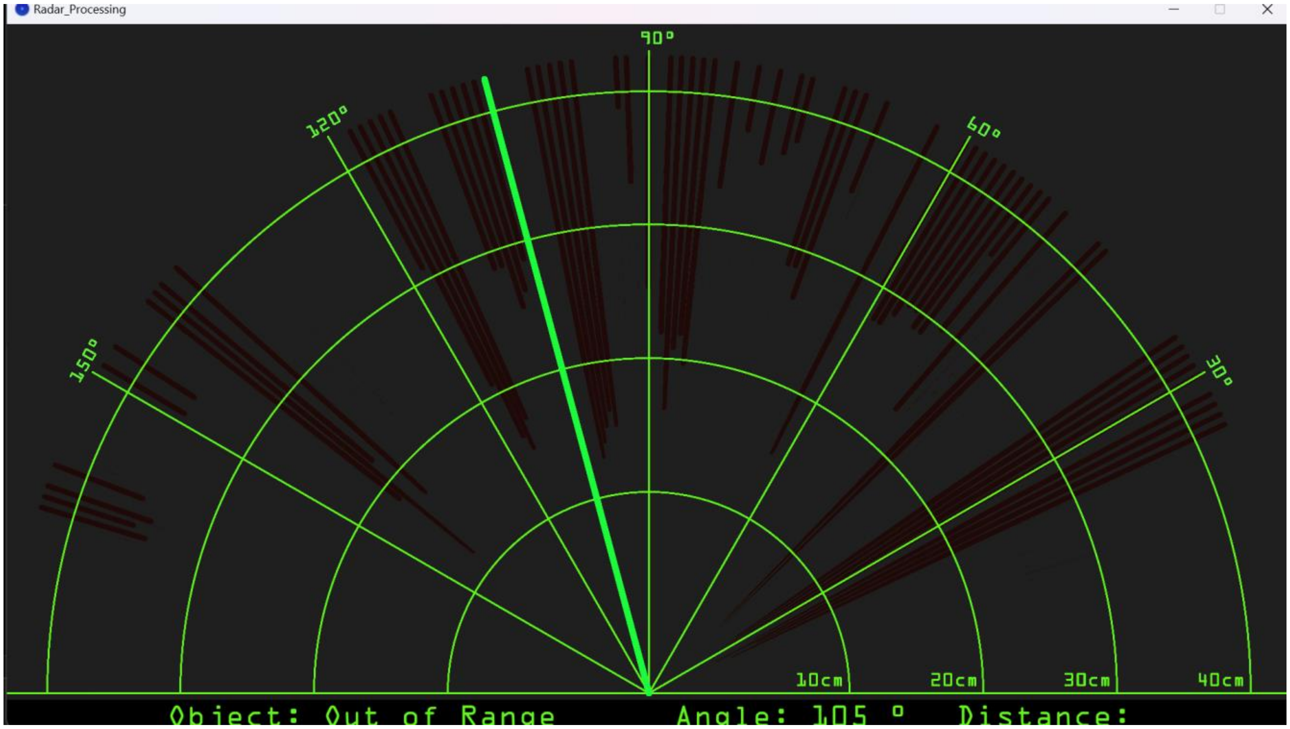


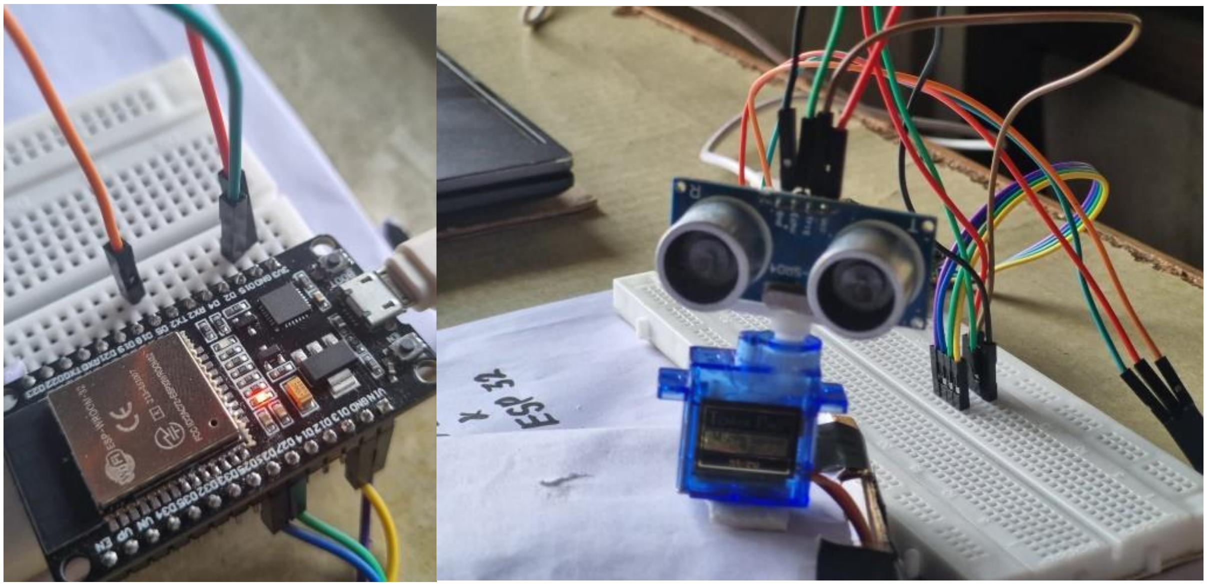
Figure 2

The figure above represents the architecture of the IoT-based radar system includes an ultrasonic sensor, a servo motor, an ESP32 microcontroller, the Blynk application, and the Processing IDE. The ultrasonic sensor measures distances to objects and is mounted on the servo motor, which rotates to scan a wider area. The ESP32 microcontroller processes the sensor data, controls the servo motor, and communicates with the Blynk application over Wi-Fi. The Blynk app enables real-time monitoring and remote control of the radar system. Data visualization is handled by the Processing IDE, which provides a graphical representation of the scanning area and detected objects. This integration of components creates a robust system for obstacle detection, distance measurement, and surveillance.

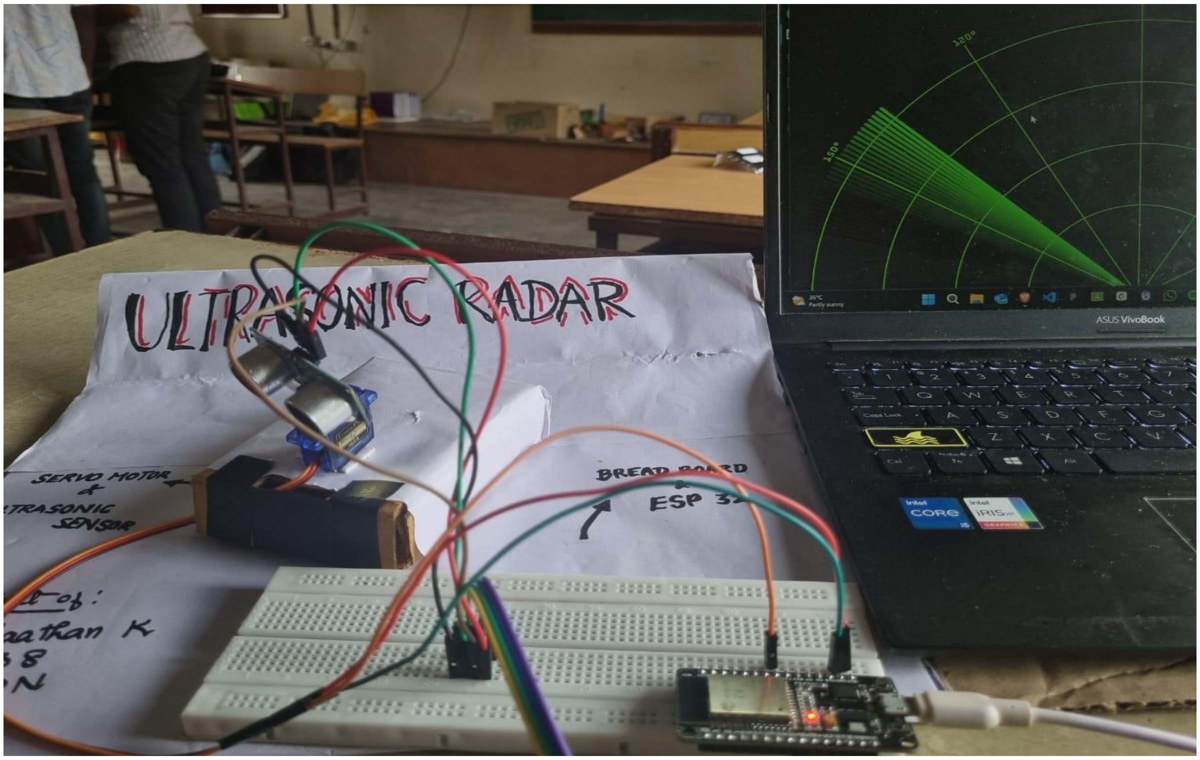
3.4 OUTPUT



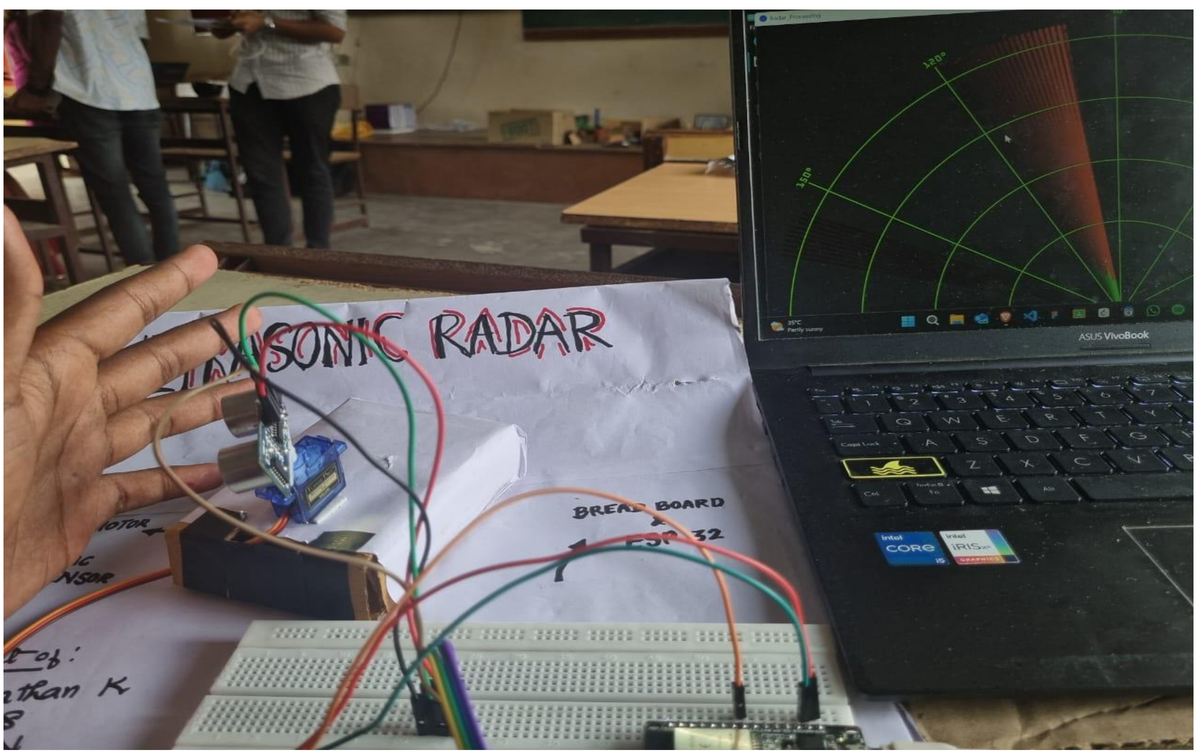
Processing IDE's output



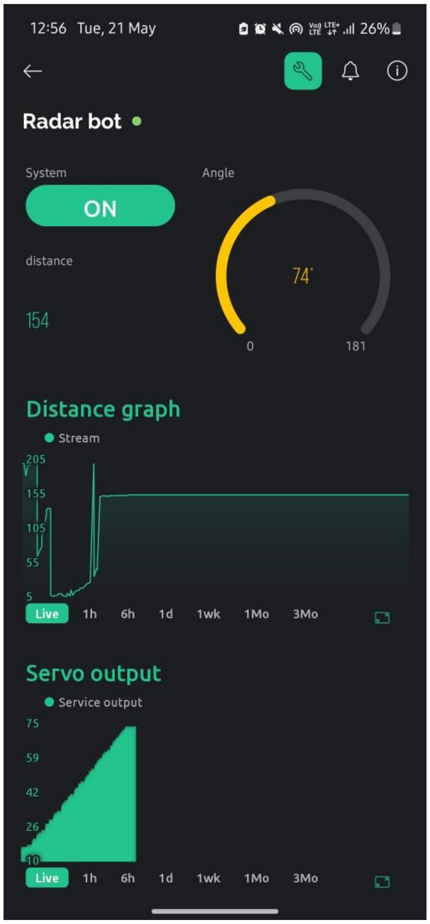
ESP 32 and ultrasonic sensor in connection



Total setup of system



Object detection



Blynk monitor

CHAPTER 4
CONCLUSION AND FUTURE WORK

The development of an IoT-based radar system using an ultrasonic sensor, servo motor, and ESP32 microcontroller demonstrates a practical and affordable solution for obstacle detection and distance measurement. By integrating with the Blynk application and Processing IDE, the system offers real-time monitoring and visualization, enhancing user interaction and control. This project successfully showcases the potential of combining IoT technologies with traditional sensor systems to create innovative and accessible solutions for everyday applications.

Future enhancements to this project could include integrating additional sensors to improve accuracy and coverage, such as infrared or LIDAR sensors. Implementing machine learning algorithms for object recognition and classification could further enhance the system's capabilities. Expanding the system's connectivity options to include other IoT platforms and improving the user interface for easier interaction are also potential areas for development. Moreover, the system could be adapted for various other applications, such as autonomous navigation for robots or drones, by enhancing its detection range and resolution.

APPENDIX I

Arduino.ino

```
#define BLYNK_TEMPLATE_ID "TMPL3JoxujNOo"
#define BLYNK_TEMPLATE_NAME "WEIGHTSYSTEM"
#define BLYNK_AUTH_TOKEN "aC0fyuJ08vBDgm2GSwpujykXpRoVzuPz"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <Servo.h>
#define trigPin 33
#define echoPin 32

char auth[] = "aC0fyuJ08vBDgm2GSwpujykXpRoVzuPz"; // Blynk auth token
char ssid[] = "Galaxy S21 FE 5G b537"; // Your Wi-Fi network SSID
char pass[] = "12345678"; // Your Wi-Fi network password

long duration;
int distance;
bool systemState = false; // Variable to store the system state (on/off)

Servo myservo;

int calculateDistance()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  Blynk.virtualWrite(V2, distance);
  return distance;
}

void setup()
```

```
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Blynk.begin(auth, ssid, pass);
  myservo.attach(5);
  Serial.begin(9600);
}

BLYNK_WRITE(V4) // Function to handle the button press
{
  int pinValue = param.asInt();
  systemState = (pinValue == 1); // Set systemState based on button press
}
void loop()
{
  Blynk.run();

  if (systemState) // Only run if system is turned on
  {
    int i;
    for (i = 15; i <= 165; i++)
    {
      myservo.write(i);
      delay(15);
      calculateDistance();
      Serial.print(i);
      Serial.print(",");
      Serial.print(distance);
      Serial.print(".");
      Blynk.virtualWrite(V1, i);
    }
    for (i = 165; i >= 15; i--)
    {
      myservo.write(i);
      delay(15);
      calculateDistance();
      Serial.print(i);
      Serial.print(",");
      Serial.print(distance);
      Serial.print(".");
      Blynk.virtualWrite(V1, i);
```

```
    }
  }
}
```

Processing IDE

```
import processing.serial.*; // imports library for serial communication
import java.awt.event.KeyEvent; // imports library for reading the data from the
serial port
import java.io.IOException;

Serial myPort; // defines Object Serial
// defubes variables
String angle="";
String distance="";
String data="";
String noObject;
float pixsDistance;
int iAngle, iDistance;
int index1=0;
int index2=0;
PFont orcFont;

void setup() {

 size (500,300); // ***CHANGE THIS TO YOUR SCREEN RESOLUTION***
 smooth();
 myPort = new Serial(this,"COM8", 9600); // starts the serial communication
 myPort.bufferUntil('.'); // reads the data from the serial port up to the character '.'. So
actually it reads this: angle,distance.
 orcFont = loadFont("OCRAExtended-30.vlw");
}

void draw() {

  fill(98,245,31);
  textFont(orcFont);
  // simulating motion blur and slow fade of the moving line
  noStroke();
  fill(0,4);
  rect(0, 0, width, height-height*0.065);

  fill(98,245,31); // green color
  // calls the functions for drawing the radar
  drawRadar();
```

```
  drawLine();
  drawObject();
  drawText();
}

void serialEvent (Serial myPort) { // starts reading data from the Serial Port
  // reads the data from the Serial Port up to the character '.' and puts it into the String
variable "data".
  data = myPort.readStringUntil('.');
  data = data.substring(0,data.length()-1);

  index1 = data.indexOf(","); // find the character ',' and puts it into the variable
"index1"
  angle= data.substring(0, index1); // read the data from position "0" to position of the
variable index1 or thats the value of the angle the Arduino Board sent into the Serial
Port
  distance= data.substring(index1+1, data.length()); // read the data from position
"index1" to the end of the data pr thats the value of the distance

  // converts the String variables into Integer
  iAngle = int(angle);
  iDistance = int(distance);
}

void drawRadar() {
  pushMatrix();
  translate(width/2,height-height*0.074); // moves the starting coordinats to new
location
  noFill();
  strokeWeight(2);
  stroke(98,245,31);
  // draws the arc lines
  arc(0,0,(width-width*0.0625),(width-width*0.0625),PI,TWO_PI);
  arc(0,0,(width-width*0.27),(width-width*0.27),PI,TWO_PI);
  arc(0,0,(width-width*0.479),(width-width*0.479),PI,TWO_PI);
  arc(0,0,(width-width*0.687),(width-width*0.687),PI,TWO_PI);
  // draws the angle lines
  line(-width/2,0,width/2,0);
  line(0,0,(-width/2)*cos(radians(30)),(-width/2)*sin(radians(30)));
  line(0,0,(-width/2)*cos(radians(60)),(-width/2)*sin(radians(60)));
  line(0,0,(-width/2)*cos(radians(90)),(-width/2)*sin(radians(90)));
```

```
  line(0,0,(-width/2)*cos(radians(120)),(-width/2)*sin(radians(120)));
  line(0,0,(-width/2)*cos(radians(150)),(-width/2)*sin(radians(150)));
  line((-width/2)*cos(radians(30)),0,width/2,0);
  popMatrix();
}

void drawObject() {
  pushMatrix();
  translate(width/2,height-height*0.074); // moves the starting coordinats to new
location
  strokeWeight(9);
  stroke(255,10,10); // red color
  pixsDistance = iDistance*((height-height*0.1666)*0.025); // covers the distance
from the sensor from cm to pixels
  // limiting the range to 40 cms
  if(iDistance<40){
    // draws the object according to the angle and the distance
  line(pixsDistance*cos(radians(iAngle)),-pixsDistance*sin(radians(iAngle)),(width-
width*0.505)*cos(radians(iAngle)),-(width-width*0.505)*sin(radians(iAngle)));
  }
  popMatrix();
}

void drawLine() {
  pushMatrix();
  strokeWeight(9);
  stroke(30,250,60);
  translate(width/2,height-height*0.074); // moves the starting coordinats to new
location
  line(0,0,(height-height*0.12)*cos(radians(iAngle)),-(height-
height*0.12)*sin(radians(iAngle))); // draws the line according to the angle
  popMatrix();
}

void drawText() { // draws the texts on the screen

  pushMatrix();
  if(iDistance>40) {
  noObject = "Out of Range";
  }
  else {
```

```
  noObject = "In Range";
  }
  fill(0,0,0);
  noStroke();
  rect(0, height-height*0.0648, width, height);
  fill(98,245,31);
  textSize(25);

  text("10cm",width-width*0.3854,height-height*0.0833);
  text("20cm",width-width*0.281,height-height*0.0833);
  text("30cm",width-width*0.177,height-height*0.0833);
  text("40cm",width-width*0.0729,height-height*0.0833);
  textSize(40);
  text("Object: " + noObject, width-width*0.875, height-height*0.0277);
  text("Angle: " + iAngle +" °", width-width*0.48, height-height*0.0277);
  text("Distance: ", width-width*0.26, height-height*0.0277);
  if(iDistance<40) {
  text("        " + iDistance +" cm", width-width*0.225, height-height*0.0277);
  }
  textSize(25);
  fill(98,245,60);
  translate((width-width*0.4994)+width/2*cos(radians(30)),(height-height*0.0907)-
width/2*sin(radians(30)));
  rotate(-radians(-60));
  text("30°",0,0);
  resetMatrix();
  translate((width-width*0.503)+width/2*cos(radians(60)),(height-height*0.0888)-
width/2*sin(radians(60)));
  rotate(-radians(-30));
  text("60°",0,0);
  resetMatrix();
  translate((width-width*0.507)+width/2*cos(radians(90)),(height-height*0.0833)-
width/2*sin(radians(90)));
  rotate(radians(0));
  text("90°",0,0);
  resetMatrix();
  translate(width-width*0.513+width/2*cos(radians(120)),(height-height*0.07129)-
width/2*sin(radians(120)));
  rotate(radians(-30));
  text("120°",0,0);
  resetMatrix();
```

```
  translate((width-width*0.5104)+width/2*cos(radians(150)),(height-height*0.0574)-
width/2*sin(radians(150)));
  rotate(radians(-60));
  text("150°",0,0);
  popMatrix();
}
```

REFERENCE

[1] B. Jadhav, P. Wanjale, R. Gavli, K. Thopate, N. Maharnawar and N. Karmankar, "IoT Based Radar System using Ultrasonic Sensor for Enhanced Object Detection and Tracking," *2023 Global Conference on Information Technologies and Communications (GCITC)*, Bangalore, India, 2023, pp. 1-5, doi: 10.1109/GCITC60406.2023.10426516

[2] M. 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

[3] Balasaheb Jadhav, Pooja Wanjale, Renuka Gavli, Kaushalya Thopate, Niranjan Maharnawar, Nikhil Karmankar, "IoT Based Radar System using Ultrasonic Sensor for Enhanced Object Detection and Tracking", *2023 Global Conference on Information Technologies and Communications (GCITC)*, pp.1-5, 2023.

[4] A. U. Kulkarni, A. M. Potdar, S. Hegde and V. P. Baligar, "RADAR based Object Detector using Ultrasonic Sensor," *2019 1st International Conference on Advances in Information Technology (ICAIT)*, Chikmagalur, India, 2019, pp. 204-209, doi: 10.1109/ICAIT47043.2019.8987259.

[5] Jeevan C N, Asha K, Sharan A C, Tejas N M, Vishwas R S, Suryanarayana N K, "Object Detection System using Arduino for Military Application", *2023 2nd International Conference for Innovation in Technology (INOCON)*, pp.1-4, 2023

[6] Ritesh Tandon, Hari Hara Viswanathan, "Smart Parking System For Indian Roads", *2023 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pp.1-5, 2023

[7] M. V. Paulet, A. Salceanu and O. M. Neacsu, "Ultrasonic radar," 2016 International Conference and Exposition on Electrical and Power Engineering (EPE), Iasi, Romania, 2016, pp. 551-554, doi: 10.1109/ICEPE.2016.7781400.