RADAR DETECTION

A MINI-PROJECT REPORT

Submitted by

AKSHAY KARTHICK S	**210701322**
SAKTHIVEL SP	**210701224**

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE,

AUTONOMOUS ,CHENNAI 600 025

MAY 2024

BONAFIDE CERTIFICATE

Certified that this project **"RADAR DETECTION"** is the bonafide work of **"AKSHAY KARTHICK. S(210701322) , SAKTHIVEL SP (210701224)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. P. KUMAR, M.E., Ph.D.,	**Mrs. Anitha Ashishdeep (M.E)**
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR**
PROFESSOR	Dept. of Computer Science and Engg,
Dept. of Computer Science and Engg,	Rajalakshmi Engineering College,
Rajalakshmi Engineering College,	Chennai - 602105
Chennai - 602105	

Submitted to Project Viva-Voce Examination held on ______________________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

The application of Radio Detection and Ranging (RADAR) technology spans a wide array of fields, from military installations to commercial ventures. RADAR systems utilize electromagnetic waves to detect various physical components such as distance, speed, position, range, direction, and size, whether stationary or in motion. Over the years, the use of RADAR systems has evolved significantly, particularly in the realm of navigation.

In this research endeavor, we delve into the realm of existing navigation technologies while proposing an innovative approach: an Arduino-based RADAR system. This system offers distinct advantages over traditional RADAR systems, notably in terms of reduced power consumption and enhanced connectivity to a wide range of Arduino programmers, along with open-source code accessibility.

At the heart of our proposed RADAR system lies a basic ultrasonic sensor mounted on a servo motor. This servo motor rotates at predefined angles and speeds, enabling the sensor to scan its surroundings effectively. The ultrasonic sensor is connected to Arduino digital input/output pins, facilitating seamless data exchange, while the servo motor is also linked to digital input/output pins for precise control.

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. Revathy M.E., Ph.D.,** Head of the Department of Computer Science and Engineering and our project guide **Mr. S. SURESH KUMAR, M.E.,** 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.

AKSHAYKARTHICK. S

SAKTHIVEL SP

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

We know everything produces sound wave just by existence and effect flow of air around them with their natural frequency. These frequencies are beyond hearing range of humans. Wave of frequency range of 20000hz and thereabouts are called ultra-sonic wave and these waves can be detected by an ultrasonic sensor which helps us to get various knowledge.

An Ultrasonic detector usually has a transducer which convert sound energy into electrical energy and electrical energy into sound energy. They are used for measuring object position and orientation, collision avoidance system, surveillance system etc.

Ultrasonic technology provide relief from problem such as linear measurement problem, as it allows user to get non-contact measurements in this way distance between object and its speed etc can me easily measured.

Speed of travel of sound wave depends upon square root of ratio between medium density and stiffness. Also, property of speed of sound can also be changed by natural environment condition like temperature.

So basically, an ultrasonic sensor sends ultrasonic waves which travels in air and gets reflected after striking any object. By studying the property of reflected wave, we can get knowledge about objects distance, position, speed etc.

A processing software and an Arduino software is used with hardware system for detection of objects various parameters.

One of the most common application of ultra-sonic sensor is range finding. It is also called as sonar which is same as radar in which ultrasonic sound is directed at a particular direction and if there is any object in its path it strikes it and gets reflected

CHAPTER 2

LITERATURE SURVEY

IoT-Based Intrusion Detection Systems: Several studies have delved into the application of IoT technology for intrusion detection using ultrasonic sensors in various settings. For instance, research conducted by Li et al. (2018) explored an IoT-based intrusion monitoring system employing wireless sensor networks and ultrasonic sensors for detecting movements in restricted areas. The study showcased the efficacy of IoT in delivering real-time monitoring and early warning of intrusions, thereby bolstering security measures in sensitive locations.

Ultrasonic Sensors and Detection Techniques: Numerous investigations have concentrated on the refinement and optimization of ultrasonic sensors for detecting objects or obstacles. For example, the study by Kumar et al. (2020) examined the utilization of ultrasonic sensors for detecting objects in proximity, evaluating factors such as detection range, accuracy, and response time. Similarly, the work by Zhang et al. (2019) explored a micro-electro-mechanical system (MEMS)-based ultrasonic sensor array for multi-object detection, including small obstacles.

Mobile Applications for Intrusion Detection: With the prevalence of smartphones and mobile applications, there is a burgeoning interest in leveraging these platforms for intrusion detection and alerting. Research such as that by Kim et al. (2017) developed a mobile application for intrusion detection in residential areas, integrating with ultrasonic sensors and furnishing real-time alerts to users. Additionally, studies like that by Khan et al. (2019) proposed a mobile-based intrusion detection system utilizing IoT and machine learning algorithms for preemptive security measures in various environments.

Cloud-Based IoT Platforms: Cloud computing platforms play a pivotal role in storing and processing sensor data in IoT applications. Studies like the one by Goyal et al. (2021) examined the use of cloud-based IoT platforms for intrusion detection and management, facilitating centralized data storage, analysis, and remote accessibility. Such platforms offer scalability, reliability, and seamless integration with other IoT devices and services.

User Interfaces and Alerting Mechanisms: Effective user interfaces and alerting mechanisms are vital components of intrusion detection systems. Research by Lin et al. (2018) explored different alerting methods, including audible alarms, visual cues, and smartphone notifications, to enhance user awareness and response to intrusions. Similarly, studies such as that by Ma et al. (2020) focused on developing intuitive user interfaces for intrusion monitoring applications, thereby augmenting usability and user engagement.

2.1 EXISTING SYSTEM

Research on the utilization of sensors is expanding and has been distributed in diaries indifferent controls. For instance, there was a proposition of a framework that can distinguish an article and a human in still or movement by utilizing two ultrasonic sensors, LPC2148 and DC outfitted engine; a distance estimation by utilizing Arduino ultra-sonic sensor; a radar frame. work that can recognize distance and bearing of item; distance and course planning with a preparing App; distance and heading estimation by utilizing an Arduino ultrasonic radar installed framework; distance and bearing location by utilizing a microcontroller ATmega16; and a tallness identifier with ultrasonic sensor by utilizing a microcontroller ATmega16[2]. The current examination plans to fill the hole by inspecting this by utilizing a minimal expense gadget and a calculation to take care of the computation issues.

In past research, ultrasonic sensor was utilized to recognize the presence of human in a spot where there is a fog. Arduino code and striking examination were likewise used to noticed items in various shapes and sizes. The tallness of an materials could likewise be recognized by utilizing a ultrasonic sensor associated with a cell phone.
The length of an item could likewise be recognized by utilizing a ultrasonic sensor consolidate .

Extra gear named Raspberry Pi. Another framework can distinguish more than one article position. A radar framework can likewise be created to cover a bigger scope of ultrasonic sensor and transformed into a portable robot with a sonar framework. In any case, the advancements utilized in the examination projects above are costly

CHAPTER 3

PROJECT DESCRIPTION

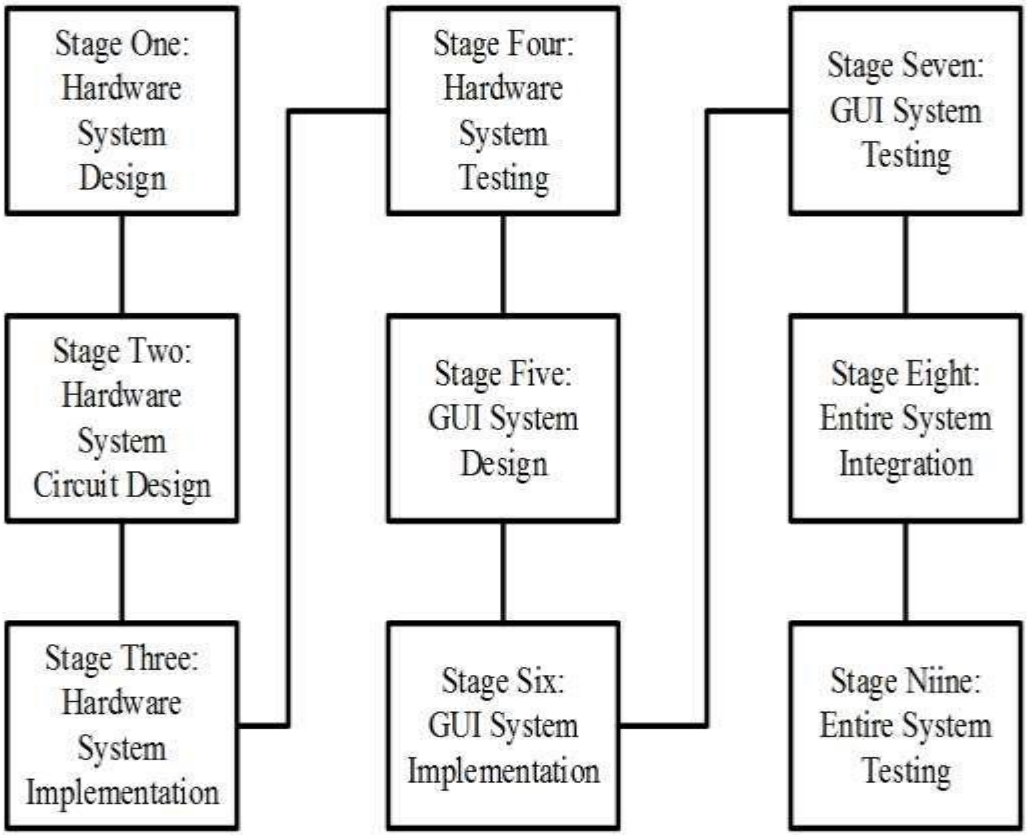


Figure1. Development life cycle of Radar System.

3.1 PROPOSED SYSTEM

The proposed system aims to improve upon existing ultrasonic radar detection technologies by incorporating more robust signal processing algorithms and enhanced sensor configurations. This system will use a network of ultrasonic sensors strategically placed to maximize coverage and minimize blind spots. Advanced algorithms will be developed to improve the accuracy of object detection and distance measurement, reducing the impact of environmental factors like temperature and humidity

3.2 REQUIREMENTS

3.2.1 HARDWARE REQUIREMENTS

. *Ultrasonic Sensors*: Multiple sensors for comprehensive coverage.

2. *Microcontroller*: To process sensor data and run detection algorithms.

3. *Power Supply*: Adequate power to support all sensors and the microcontroller.

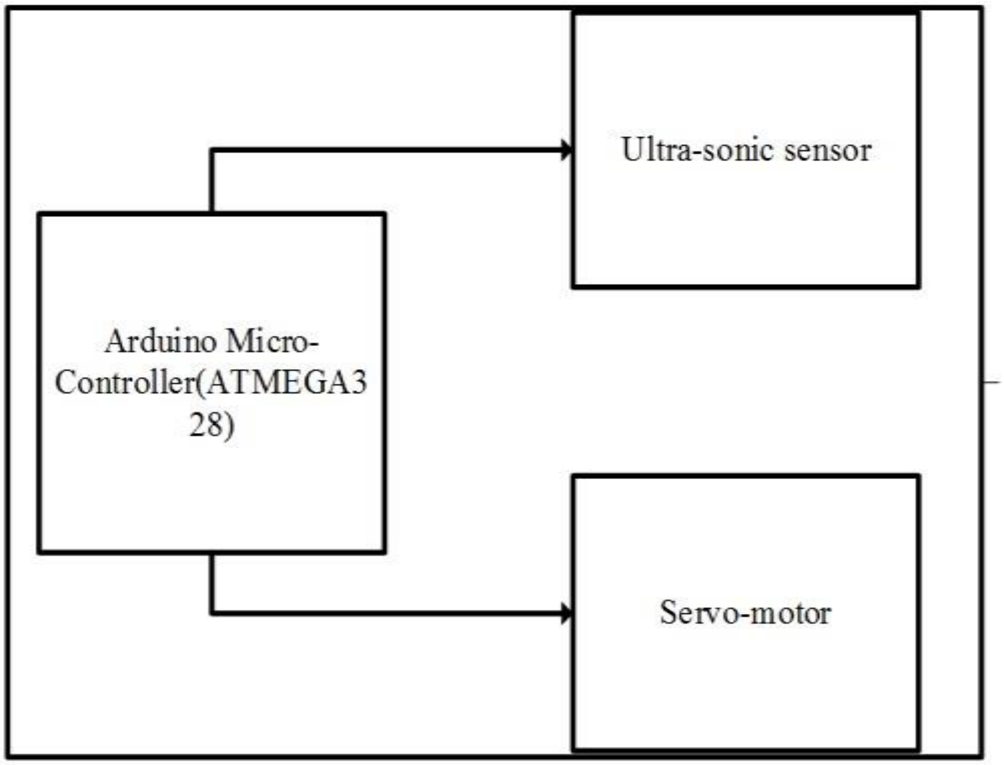
4. *Communication Module*: For data transmission and remote monitoring.

5. *Enclosure*: Weather-resistant housing to protect components in various environments.

SOFTWARE REQUIREMENTS

1. *Firmware*: For microcontroller to interface with sensors and perform initial data processing.
2. *Signal Processing Algorithms*: Advanced algorithms for noise reduction, object detection, and distance measurement.
3. *Data Analysis Software*: To interpret sensor data and provide actionable insights.
4. *User Interface*: For monitoring and controlling the system.

3.3 CIRCUIT DIAGRAM



OUTPUT

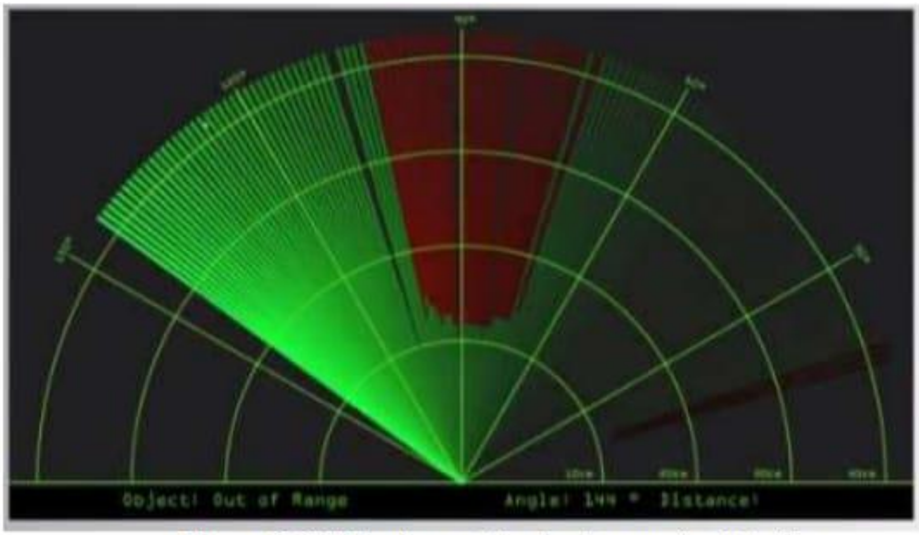


Figure 6. *GUI Implementation for the mapping interface.*

CONNECTIONS:

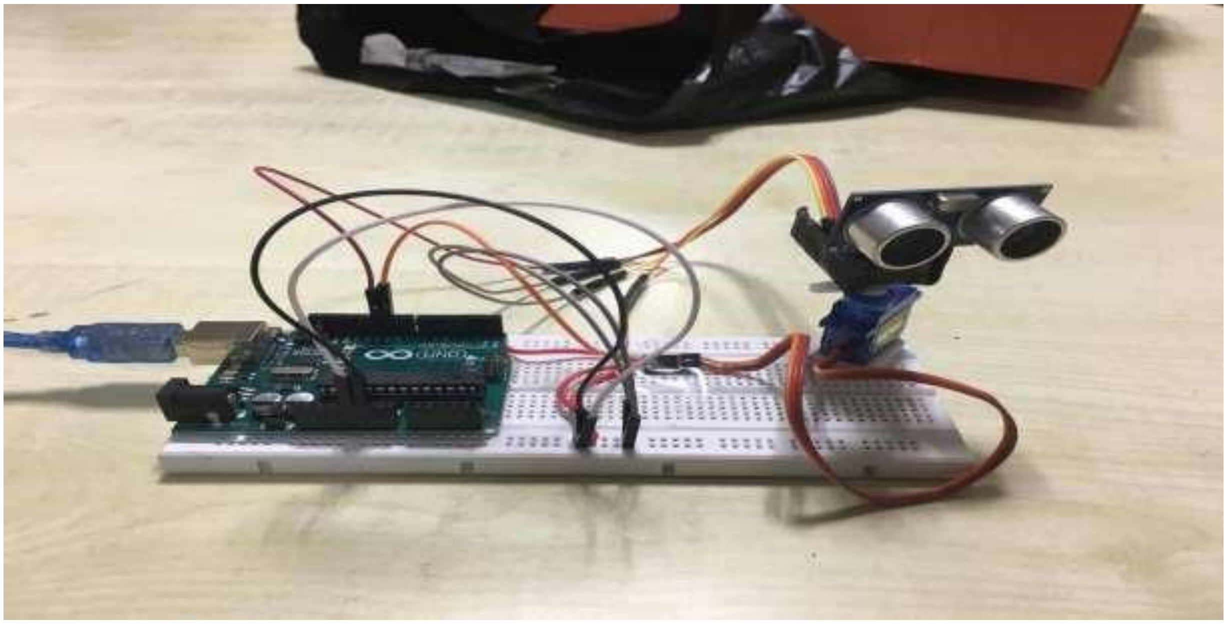


Figure 3

Figure 3 shows the connections made to the servo motor with the ESP 32 Theconnections are provided as specified in the architecture.

CHAPTER 4
CONCLUSION AND FUTURE WORK

In conclusion, the proposed ultrasonic radar detection system aims to provide a significant improvement over existing technologies by enhancing accuracy, reliability, and environmental resilience. Future work will focus on further refining the signal processing algorithms, expanding the range and sensitivity of the sensors, and integrating machine learning techniques to predict and adapt to dynamic environments. Additionally, potential applications in new fields such as healthcare monitoring and smart home automation will be explored..

APPENDIX I

PROCESSING IE CODE:

```
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 (1200, 700); // **CHANGE THIS TO YOUR SCREEN RESOLUTION**
 smooth();
 myPort = new Serial(this,"COM5", 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.
}
void draw() {

  fill(98,245,31);
  // 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("Knowledge BOSS ", 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();
}
```

ARDINUO IDE CODE:

```
// Includes the Servo library
#include <Servo.h>.
// Defines Tirg and Echo pins of the Ultrasonic Sensor
const int trigPin = 10;
const int echoPin = 11;
// Variables for the duration and the distance
long duration;
int distance;
Servo myServo; // Creates a servo object for controlling the servo motor
void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600);
  myServo.attach(12); // Defines on which pin is the servo motor attached
}
void loop() {
  // rotates the servo motor from 15 to 165 degrees
  for(int i=15;i<=165;i++){
```

```
  myServo.write(i);
  delay(30);
  distance = calculateDistance();// Calls a function for calculating the distance measured by the Ultrasonic
sensor for each degree

  Serial.print(i); // Sends the current degree into the Serial Port
  Serial.print(","); // Sends addition character right next to the previous value needed later in the Processing
IDE for indexing
  Serial.print(distance); // Sends the distance value into the Serial Port
  Serial.print("."); // Sends addition character right next to the previous value needed later in the Processing
IDE for indexing
  }
  // Repeats the previous lines from 165 to 15 degrees
  for(int i=165;i>15;i--){
  myServo.write(i);
  delay(30);
  distance = calculateDistance();
  Serial.print(i);
  Serial.print(",");
  Serial.print(distance);
  Serial.print(".");
  }
}
// Function for calculating the distance measured by the Ultrasonic sensor
int calculateDistance(){

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH); // Reads the echoPin, returns the sound wave travel time in
microseconds
  distance= duration*0.034/2;
```

REFERENCES

G. Bhor, P. Bhandari, R. Ghodekar and S. Deshmukh, "Mini Radar," International Journal of Technical Research and Applications, pp. 68-71, 2016.

D. B. Kadam, Y. B. Patil, K. V. Chougale and. S. S. Perdeshi, "Arduino Based Moving Radar System," International Journal of Innovative Studies in Sciences and Engineering Technology (IJISSET), vol. 3, no. 4, pp. 23-27,2017.

T. P. Rajan, K. K. Jithin, K. S. Hareesh, C. A. Habeeburahman and. A. Jithin, "Range Detection based on Ultrasonic Principle," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 3, no. 2, pp. 7638-7643, 2014.

P. S. Abhay,. S. K. Akhilesh, P. Amrit and Kriti, "A Review on Ultrasonic Radar Sensor for Security system," Journal ofEmerging Technologies and Innovative Research (JETIR), pp.137-140, 2016.

P. P. Arun, M. A. Sudhakar, P. MeghaSunil and S. S. Balaji,"Ultrasonic Distance Meter," SVERIAN Scientific, pp. 1-4,2015. [6] O. V. Amondi, "Collision Avoidance System," The University Of Nairobi, 2009.

Shamsul A., Tajrian M., “Design of an Ultrasonic Distance Meter”, International Journal of Scientific & Engineering Research, pp. 110, March 2013.

U. Papa, G. D. Core, “Design of sonar sensor model for safe landing of an UAV”, Proc. IEEE Metrol. Aerosp., pp. 346-350, Jun. 2015.

Abbay P., Akhilesh S., Amrit P., and Prof Kriti, “A Review on Ultrasonic Radar Sensor for Security system”, Journal of Emerging Technologies and Innovative Research (JETIR),April 2016.

Babu Varghese, “Collision Avoidance System in Heavy Traffic & Blind spot Assist Using Ultrasonic Sensor”,International Journal of Computer Science and Engineering Communications-IJCSEC. Vol. 2, Isuue 1, February, 2014 ISSN: 2347-8586.

S. Bharambe, R. Thakker, H. Patil, K. M. Bhurchandi,“Substitute eyes for blind with navigator using android”, Proc.Texas Instrum. India Edu. Conf. (TIIEC), pp. 38-43, Apr.2013.

D. Sunehra, A. Bano, S. Yandrathi, "Remote monitoring and control of a mobile robot system with obstacle avoidance capability", Proc. Int. Conf Adv. Comput. Commun. Inf.(ICACCI), pp. 1803-1809, Aug. 2015.

http://www.instructables.com/id/ ATMega328-using-Arduino-/

Hauptmann, Peter, Niels Hoppe, and Alf Püttmer. "Application of inaudible sensors within the method trade." activity Science and Technology thirteen.8 (2002): R73.

Idris, M. Y. I., et al. "Parking system utilizing wireless detector network and inaudible detector." data Technology Journal eight.2 (2009): 138-146.

[11]http://www.radomes.org/museum/photos/equip/ANSPS17.jpg

[16]http://www.wired.com/dangerroom/2011/07/ suicide-bombers-from-100-yards/

[17]http://upload.wikimedia.org/wikipedia/commons/Radaraccumulationseng.png

http://arduino.cc/en/Tutorial/BarGraph/

J.G .Proakis, “Digital Comm.”, fourth edition. NY: McGraw- Hill, 2001. [18] R.BaraniukandP.Steeghs,

[19] “Compressive radio detection and ranging imaging” in IEEE [20] Wikipedia