A HOBBY Project report on

SHORT RANGE ULTRASONIC RADAR

BACHELOR OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Of

Sri Siddhartha Academy of Higher Education

Tumkur, Karnataka

By

NEHA ACHARYA : 20CS051

PRATHUSAHA K B : 20CS058

SRI SIDDHARTHA INSTITUTE OF TECHNOLOGY

Tumkur , Karnataka

2021-22

Department of Computer Science and Engineering

SRI SIDDHARTHA INSTITUTE OF TECHNOLOGY

Tumkur, Karnataka

[A constituent college of Sri Siddhartha Academy of Higher Education, Tumkur]

CERTIFICATE

This is to certify that the project work entitled

SHORT RANGE ULTRASONIC RADAR

Submitted in partial fulfilment of requirement for the award of degree of **BE in Computer Science and Engineering** of Sri Siddhartha Institute of Technology, Tumkur ,Karnataka is a result of the bonafide work carried out by

NEHA ACHARYA : 20CS051

PRATHUSAHA K B : 20CS058

During the academic year 2021-2022

Signature of the Guide

Dr.Renukalatha.S,M.S,PhD,MISTE

Professor,

Dept. of CSE

SSIT,Tumkur

ACKNOWLEDGEMENT

We would like to express our sincere thanks to honourable Founders **Dr.H.M.Gangadharaiah** and **Dr.G.Shivaprasad,Secretary** and Chancellor **Dr.G.Parameshwara** and Principal **Dr.M.S.Raviprakasha** of Sri Siddhartha Institute of Technology ,Tumakuru for providing us with all necessary facilities for the research.

We place on record, our sincere thanks to our **H.O.D.Dr.M.Siddappa** , Department of Computer Science and Engineering for continuous encouragement.

We are also grateful to **Dr.Renukalatha.S** Professor in the Department of Computer Science and Engineering and Asst.Placement Officer,Training & Placement Department. We are extremely thankful and indebted to her for expertise, sincere and valuable guidance and encouragement extended to us.

We take this opportunity to express gratitude to all of the Department faculty members for their help and support. We also thank our parents for unceasing encouragement, support and attention.

DECLARATION

I **Neha Acharya**, **Prathuasha K B**, students of III semester CSE, Sri Siddhartha Institute of Technology bearing CSE hereby declare that the project entitled **"SHORT RANGE ULTRASONIC RADAR"** has been carried out by us under the supervision of Guide **Dr.Renukalatha.S** Professor and submitted in partial fulfilment of the requirements for the hobby project of III semester of CSE by Sri Siddhartha Institute of Technology during the academic year 2021-22.This report has been submitted to any other organisations/university for any award of degree or certificate.

Neha Acharya(20CS051)

Prathuasha K B(20CS058)

INDEX

INDEX

INTRODUCTION

Chapter-1

INTRODUCTION

1.1 Abstract

A rangefinder is a device that measures the distance from the target to the observer, for the purposes of surveying, determining focus in photography, or accurately aiming a weapon. In this technical project, we make a simple radar using the ultrasonic sensor, this radar works by measuring a range from 3cm to 40 cm as non-contact distance, with angle range between 15˚ and 165˚.The movement of the sensor is controlled by using a small servo motor. Information received from the sensor will be used by "Processing Development Environment" software to illustrate the result on a PC screen.

1.2 Introduction

Radar is an object detection system that uses electromagnetic waves to identify range, altitude, direction, or speed of both moving and fixed objects such as aircraft, ships, vehicles, weather formations, and terrain Fig(1). When we use ultrasonic waves instead of electromagnetic waves, we call it ultrasonic radar .

The main components in any ultrasonic radar are the ultrasonic Sensors. Ultrasonic sensors work on a principle similar to radar or sonar which evaluates attributes of a target by interpreting the echoes from radio or sound waves respectively.

Radar's information will appear in different ways. Basic and old radar station used sound alarm or LED, modern radar uses LCD display to show detailed information of the targeted object. We use Computer screen to show the information (distance and angle).

Fig.1 Google Auto-drive car

LITERATURE SURVEY

Chapter-2

LITERATURE SURVEY

2.1 Tools and technologies used:

2.1.1 Arduino Board UNO Model

Arduino is a hardware and software company, project, and user community that designs and manufactures computer open-source hardware, open-source software, and microcontroller-based kits for building digital devices and interactive objects that can sense and control physical devices.

The project is based on microcontroller board designs. The board provides sets of digital and analog Input/output (I/O) pins that can interface to various expansion boards (termed shields) and other circuits Fig (2). The boards feature serial communication interfaces, including Universal Serial Bus (USB) on UNO model, for loading programs from personal computers.

For programming the microcontrollers, the Arduino project provides an integrated development environment (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board Fig (3).

Fig.2 Arduino UNO

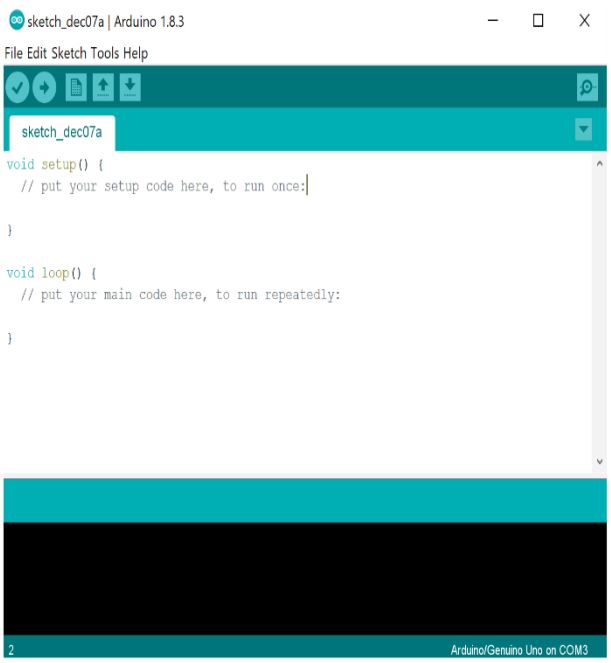


Fig.3 Arduino IDE Software

2.1.2 Processing Software

Processing is an open source computer programming language and integrated development environment (IDE) built for the electronic arts, new media art, and visual design communities with the purpose of teaching the fundamentals of computer programming in a visual context Fig (4).

Specifications:

- ♣ Free to download and open source
- ♣ Interactive programs with 2D, 3D or PDF output
- ♣ OpenGL integration for accelerated 2D and 3D
- ♣ For GNU/Linux, Mac OS X, and Windows
- ♣ Over 100 libraries extend the core software
- ♣ Well documented, with many books available

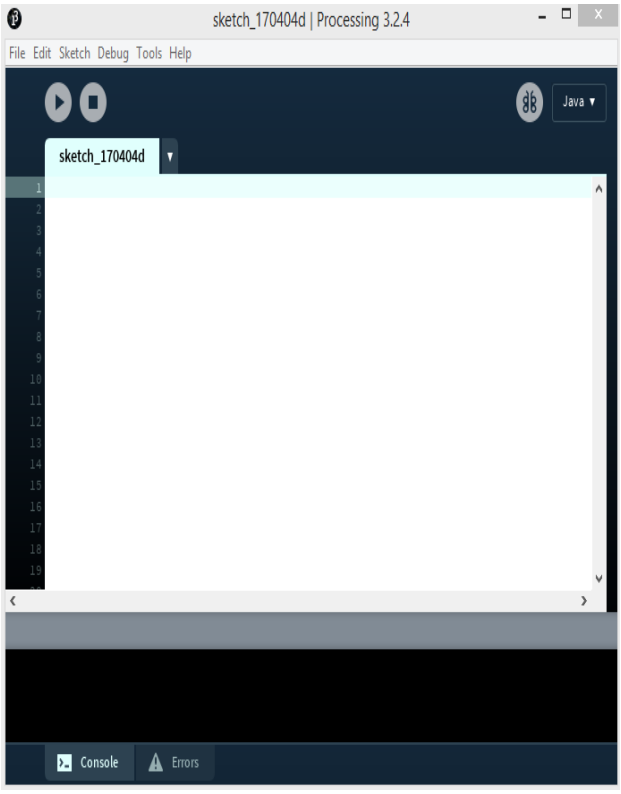


Fig.4 Processing IDE

2.1.3 Ultrasonic sensors HC- SR04

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules include ultrasonic transmitters, receiver, and control circuit, within measuring angle 15 degrees Fig (5).

The basic principle of work Fig (6)**:**

(1) Using IO trigger for at least 10us high-level signal.

(2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.

(3) IF the signal back, through high level, time of high output IO duration is the time from sending ultrasonic to returning. Test distance = (high level time × velocity of sound (340M/S) / 2.

Wire connecting directly as following:

♣ 5V Supply ♣ Trigger Pulse Input ♣ Echo Pulse Output ♣ 0V Ground

Fig.5 Ultrasonic Sensor

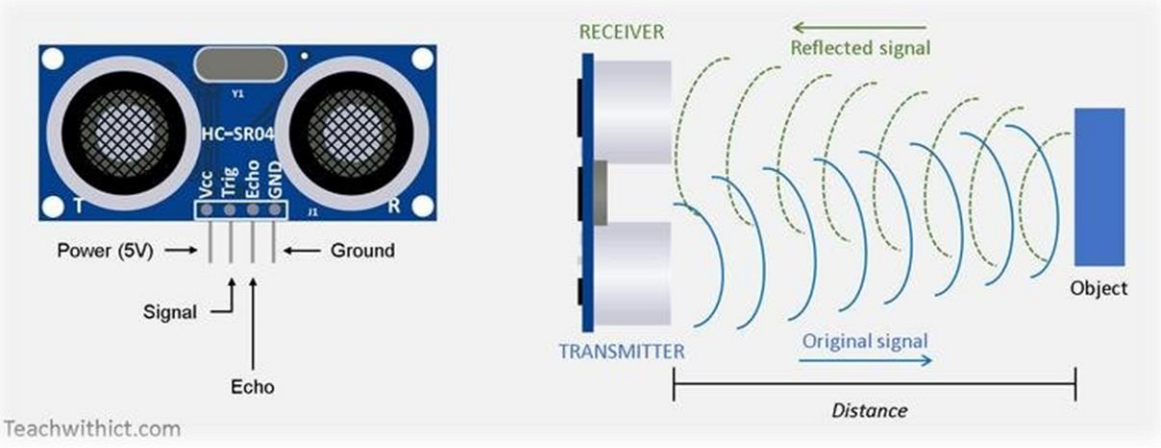


Fig.6 Working Principle

2.1.4 Servo Motor tower pro micro servo 9g

Tiny and lightweight with high output power. The servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller Fig (7). You can use any servo code, hardware or library to control these servos.

Specifications

♣ Weight: 9 g

♣ Dimension: 22.2 x 11.8 x 31 mm approx.

♣ Stall torque: 1.8 kg f cm

♣ Operating speed: 0.1 s/60 degree

♣ Operating voltage: 4.8 V (~5V)

♣ Temperature range: 0 ºC – 55 ºC

Fig.7 Servomotor. Tower Pro Micro Servo 9g and cable

2.2 Working procedure

1. Components needed for this Project :

- Arduino Board UNO Model.
- Processing software.
- Ultrasonic sensor HC- SR04.
- Servo Motor tower pro micro servo 9g.
- Jump Wires.

2.2.1 Circuit Diagram

We connected the Ultrasonic Sensor HC-SR04 to the pins number 10 and 11 on the Arduino Board.

Trig Pin = 8.

- ♣ Echo Pin = 9.

And the servo motor to the pin number 10 on the Arduino Board. Fig (9) shows circuit structure for the project.

- ♣ My Servo = 10.

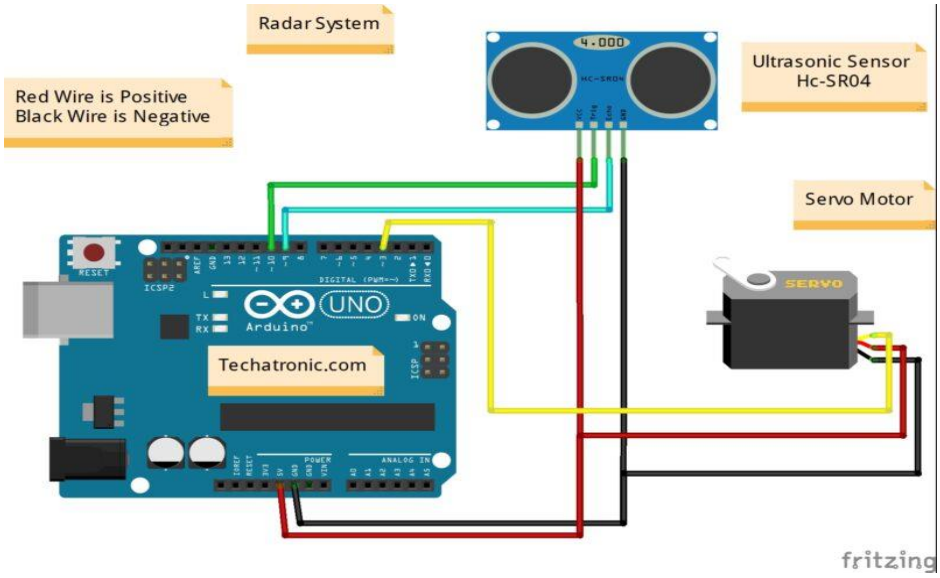


Fig.9 Circuit Diagram

2. Here's the final appearance of the project Fig(10):

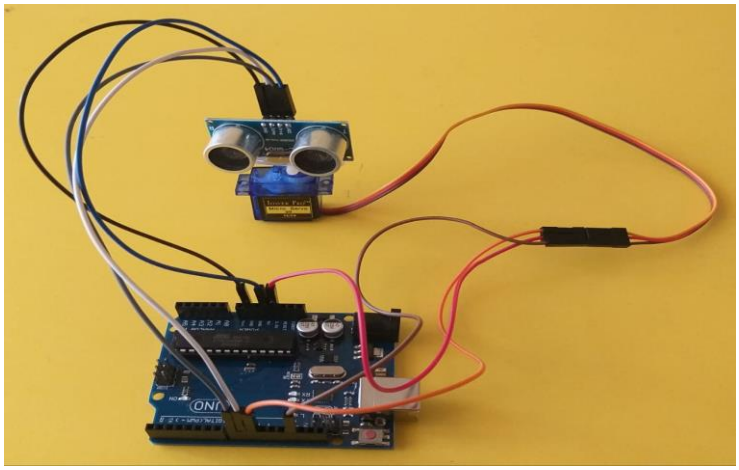


Fig.10 Final appearance

2.2.2 Write and upload sketch to Arduino

1. We wrote a sketch in IDE, for this project we need to include some libraries. We use (Serial.h) built-in library for transfer data through the serial port with processing software. Therefore, we add the last library for servo motor (Servo.h) and added NewPing library which includes the last update functions and features for the ultrasonic sensor. Fig (11).

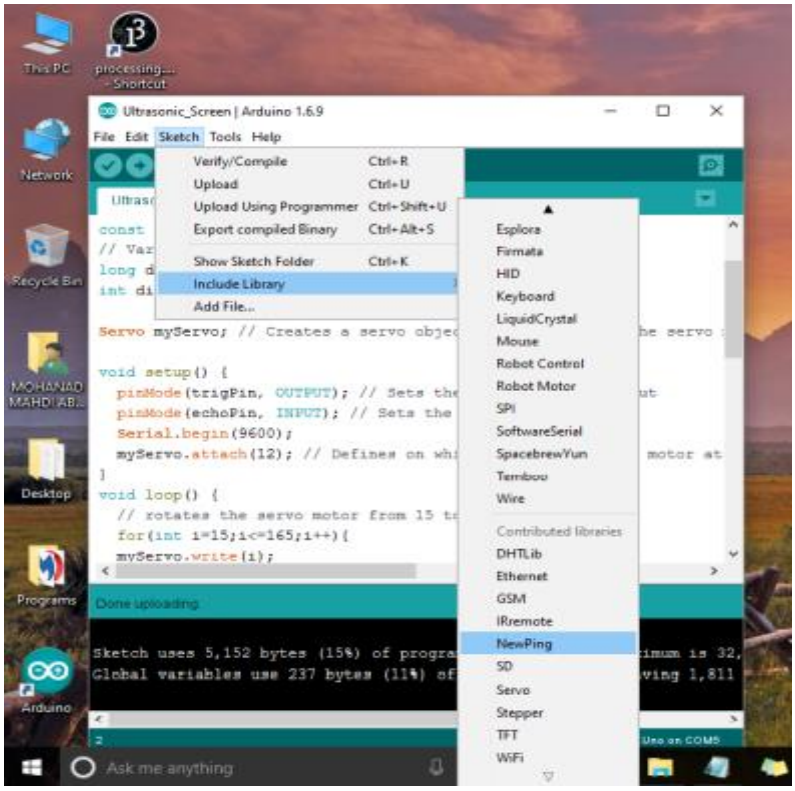
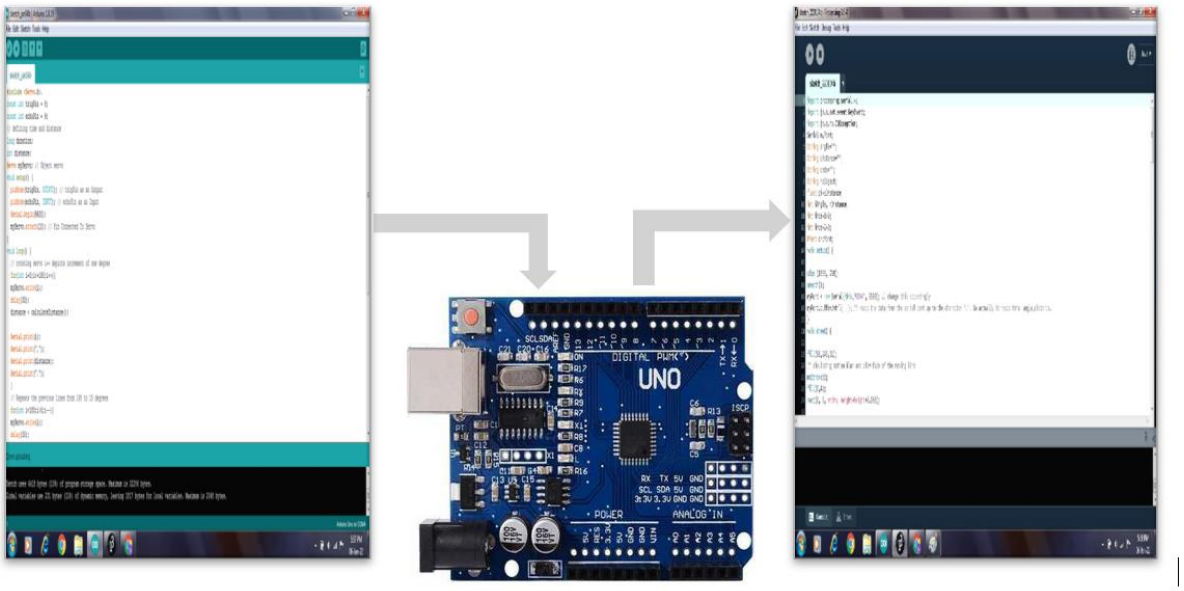


Fig.11.Add library to IDE

2. Then, we make a code and upload it to the Arduino board to enable the interaction between the Arduino and the Processing IDE Fig(12).



Upload code =>Arduino UNO => Serial Communication

Fig.12 Project Code

2.2.3 Write and upload sketch to Processing

1.The values for the angle and the distance measured by the sensor will be read from the Arduino board by the Processing IDE using the SerialEvent() function which reads the data from the Serial Port. These values will be used for drawing the lines, the detected objects and some texts.

2. For drawing the radar display we make this function drawRadar() which consist of arc() and line() functions Fig(13).

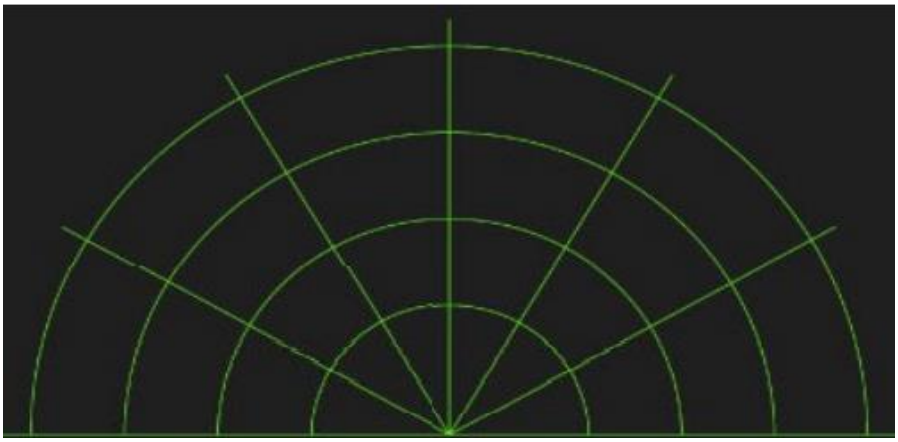


Fig.13 The Radar Workspace

3.For drawing the moving lines we make this function drawLine(). Its center of rotation is set with the translate() function and using the line() function in which the i Angle variable is used to redraw the line for each degree. Fig (14)

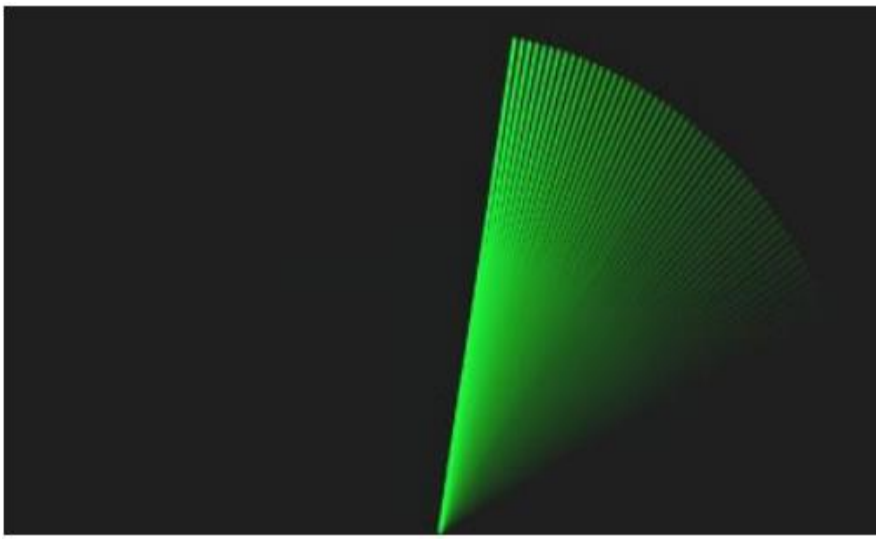


Fig.14 Radar Lines

4. For drawing the detected objects we made the drawObject() function. It receives the distance from the ultrasonic sensor, transforms it into pixels. Then, using the angle detected by the sensor it draws the object on the radar screen Fig(15).

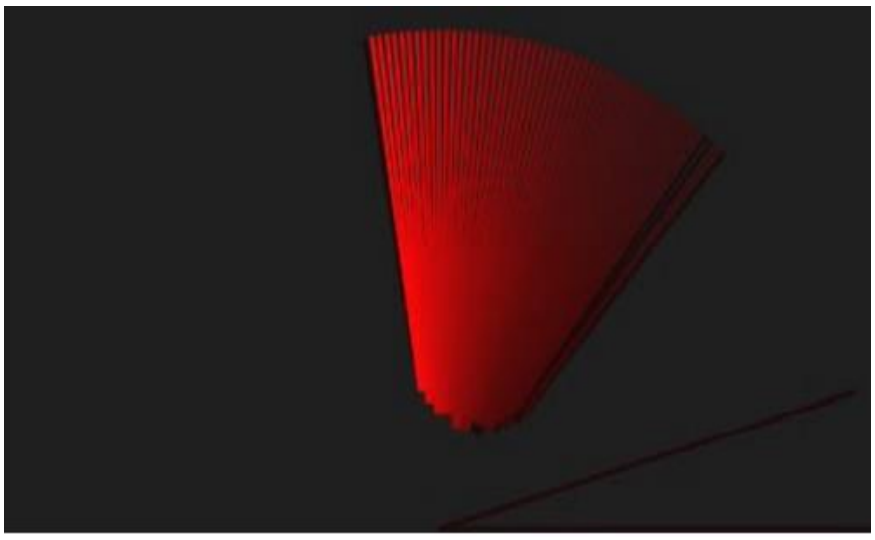


Fig.15 Radar detected lines

5.To illustrate the text on the screen, we make the drawText() function that draws texts on some particular locations. All of these functions are called in the main draw() function which is repeated in each iteration to draw the screen details.

6.We are using the fill() function with 2 parameters for simulating motion blur and slow fade of the moving line. Fig (16) shows the final appearance of the radar screen:

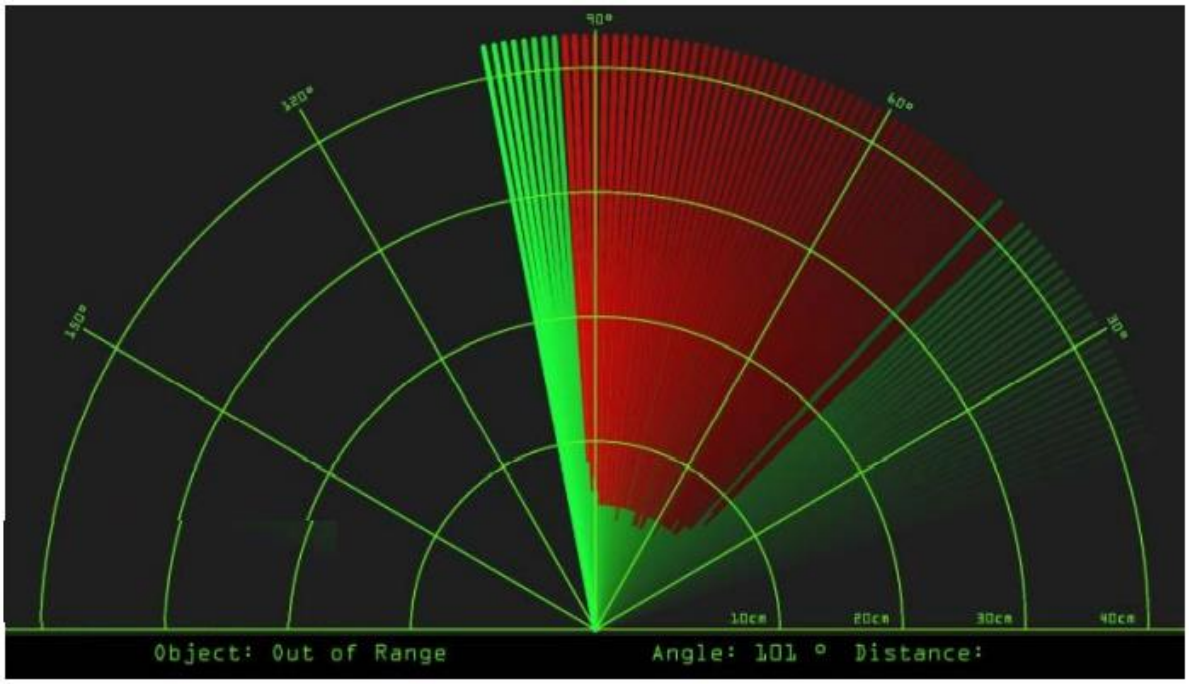


Fig.16 Radar Screen

CODE IMPLEMENTATION

Chapter-3

Code implementation

3.1 Arduino IDE Code

```
#include <Servo.h>.
const int trigPin = 8;
const int echoPin = 9;
// defining time and distance
long duration;
int distance;
Servo myServo; // Object servo
void setup() {
      pinMode(trigPin, OUTPUT); // trigPin as an Output
      pinMode(echoPin, INPUT); // echoPin as an Input
      Serial.begin(9600);
      myServo.attach(10); // Pin Connected To Servo
}
void loop() {
  // rotating servo i++ depicts increment of one degree
  for(int i=0;i<=180;i++){
        myServo.write(i);
        delay(30);
        distance = calculateDistance();

        Serial.print(i);
        Serial.print(",");
        Serial.print(distance);
        Serial.print(".");
  }
  // Repeats the previous lines from 165 to 15 degrees
```

```
  for(int i=180;i>0;i--){
          myServo.write(i);
          delay(30);
          distance = calculateDistance();
          Serial.print(i);
          Serial.print(",");
          Serial.print(distance);
          Serial.print(".");
  }
}
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);
          distance= duration*0.034/2;
          return distance;
}
```

3.2 Processing Software Code

```
import processing.serial.*;
import java.awt.event.KeyEvent;
import java.io.IOException;
Serial myPort;
String angle="";
String distance="";
String data="";
```

```
String noObject;
float pixsDistance;
int iAngle, iDistance;
int index1=0;
int index2=0;
PFont orcFont;
void setup() {

        size (1366, 768);
        smooth();
        myPort = new Serial(this,"COM4", 9600); // change this accordingly
        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 colour
// 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 that's 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 that's 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 coordinates 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 coordinates to
new location
        strokeWeight(9);
        stroke(255,10,10); // red colour
        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("Radar", 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();

}
```

ADVANTAGES AND APPLICATIONS

Chapter-4

Advantages and applications

4.1 Advantages

- ♣ RADAR can penetrate mediums such as clouds, fogs, mist, and snow
- ♣ RADAR signal can penetrate insulators.
- ♣ It is cheaper as compared to other systems.
- ♣ It is wireless and does not rely on wire connectivity.
- ♣ It covers a wider geographical area.
- ♣ It allows for repetitive coverage.
- ♣ It is fast if the area is not too large.
- ♣ Cheap and fast method of calculating base maps when no detailed survey is required.
- ♣ RADAR signals can target several objects simultaneously.
- ♣ It has several industrial applications.

4.2 Applications

This Radar System have various applications for security purposes and it is mainly used for mapping.

- ♣ APPLCATION IN AIR FORCE: It is used in airplanes or aircraft machines which have implemented radar system in it to detect the objects that comes in a way. It is also used to calculate height readings.

- ♣ APPLICATION IN MARINE: This radar system also used in ships or marine. It is implemented on big ships to calculate the distance of other boats or ships, with the help of this sea accidents can also be reduced by not colliding. It can also be implemented on ports to see the distance of other ships and to monitor or control the ship movements.

- ♣ APPLICATON IN METEROLOGY: Meteorologists also uses radar systems to track or monitor the wind. It has been become an important equipment for climate testing. For example to detect tornados, storms.

CONCLUSION

Chapter-5

Conclusion

Radar is normally used to determine velocity, range, and position of an object. In this technical project, we read the distance and angles of detected objects in order to convert these data into visual information. The performance of our project is so good. It works smoothly to detect objects within the designed range. The screen shows the information clearly with enough delay for the user to read it. As we have designed a short range radar therefore our research was specified and limited. This system can only detect objects from 0 to 180 degrees only because the servo motor that we have used can rotate only to this range. So, due to this limitation our design cannot be applied to places or areas for obstacle detection on a larger scale. Usage of a 360 degrees rotating servo motor can make the system more efficient.

This project could be helpful for object avoidance/ detection applications. This project could easily be extended and could be used in any systems may need it. This framework can also be developed or modified according to the rising needs and demand.

REFERENCES

1. Ultrasonic Radar and its applications:
   https://scholar.google.co.in/scholar?q=ultrasonic+radar+applications&hl=en&as_sdt=0&as_vis=1&oi=scholart

2. Arduino UNO:
   https://en.wikipedia.org/wiki/Arduino

3. Servo motor datasheet.

4. Ultrasonic Sensors:
   https://www.solidswiki.com/index.php?title=Ultrasonic_Sensors

5. Circuit Diagram:
   https://techatronic.com/radar-using-arduino-ultrasonic-sensor/

6. Arduino IDE and Processing Code:]
   http://howtomechatronics.com/projects/arduino-radar-project/