**BHARATI VIDYAPEETH COLLEGE OF ENGINEERING, NAVI MUMBAI**

A Report on

# ULTRASONIC RADAR USING ARDUINO

For

Mini Project 1-B (REV-2019 ‘C’ Scheme) of Second Year, (SE Sem-IV)

In

**Electronics & Telecommunication Engineering**

By

**1.Dhairyashil Jadhav-39 2. Anish Jadhav-38**
**3.Sushant Edake-26 4. Yash Gorad-31**

Under the Guidance of

**Prof. Rupali Patil**

UNIVERSITY OF MUMBAI

AY 2022-2023

# CERTIFICATE

This is to certify that the project entitled ULTRASONIC RADAR USING ARDUINO is a bonafide work of

**1.Dhairyashil Jadhav-39**

**2.Anish Jadhav-38**

**3.Sushant Edake-26**

**4.Yash Gorad-31**

submitted to the University of Mumbai in partial fulfilment of the requirement for the award of Mini Project 1-b (REV- 2019 ‘C’ Scheme) of Second Year, (SE Sem-IV) in Electronics & Telecommunication Engineering as laid down by University of Mumbai during academic year 2022-2023

(____________________) (____________________)

Internal Examiner External Examiner

Guide Head of Department Principal

.

# INDEX

# ABSTRACT

Radar is an object detection system which uses radio waves to determine the range, altitude, direction, or speed of objects. It can be used to detect aircraft, ships, spacecraft, guided missiles, motor vehicles, weather formations, and terrain. The radar dish or antenna transmits pulses of radio waves or micro waves which bounce off any object in their path.

The modern uses of radar are highly diverse, including air traffic control, radar astronomy, air-defense systems, antimissile systems ;marine radar start locate landmarks and other ships; aircraft anti-collision systems; ocean surveillance systems, outer space surveillance and rendezvous systems; meteorological precipitation monitoring; altimetry and flight control systems; guided missile target locating systems; and ground penetrating radar for geological observations.

The Arduino based project requires a ultrasonic sensor, the sensor released the waves which we want to measure the distance of a object. The microcontrollers of the Arduino board can be programmed using C and C++ languages. When a code is written in Arduino UNO IDE software and connected to the board through a USB cable, Arduino boards have lot of applications in the present day scenario, so we have decided to do a small project on them.

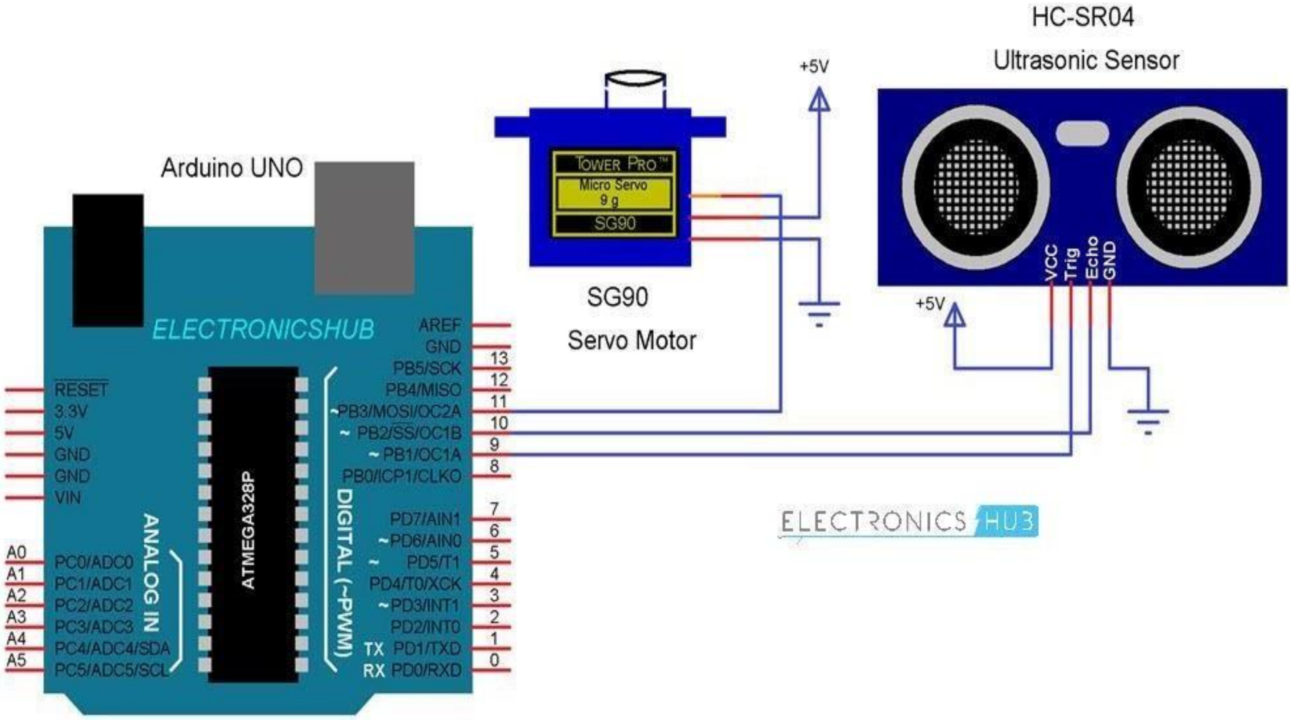
# INTRODUCTION

In this project, I will show you how to design a simple Radar Application using Arduino and Processing. This Arduino Radar Project is implemented with the help of Processing Application.

Radar is a long-range object detection system that uses radio waves to establish certain parameters of an object like its range, speed and position. Radar technology is used in aircrafts, missiles, marine, weather predictions and automobiles.

Even though the title says Arduino Radar Project, technically the project is based on Sonar technology as I will be using an Ultrasonic Sensor to determine the presence of any object in a particular range.

# CIRCUIT DIARGRAM



# COMPONENTS / TOOL

## Hardware

1. Arduino UNO

2. HC-SR04 Ultrasonic Sensor

3. SG90 Servo Motor

4. Connecting Wires

5. Jumper Cables

6. USB Cable (for Arduino)

7. Bread board

## Software

- Arduino IDE
- Processing Application

# CODE

## I. ARDUINO 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; return distance;

}
```

==================================================================

## II. PROCESSING 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,(widthwidth*0.479),(width
width*0.479),PI,TWO_PI); arc(0,0,(width-
width*0.687),(widthwidth*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)),(widthwidth*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 ("Indian
Lifehacker

", 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)wi
dth/2*sin(radians(30))); rotate(-radians(-60)); text("30°",0,0);

resetMatrix(); translate((width-
width*0.503)+width/2*cos(radians(60)),(heightheight*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(radian
s(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();
```

We create a variable analog and assign it to 0.This is because the voltage value we are going to read is connected to the analog pin is A0. This voltage represents the voltage value falls across the resistor value we are measuring. Next we create a variable name raw, which we will use to read in the analog voltage value. This later is our code get assigned to the analogue read () function.

## WORKING

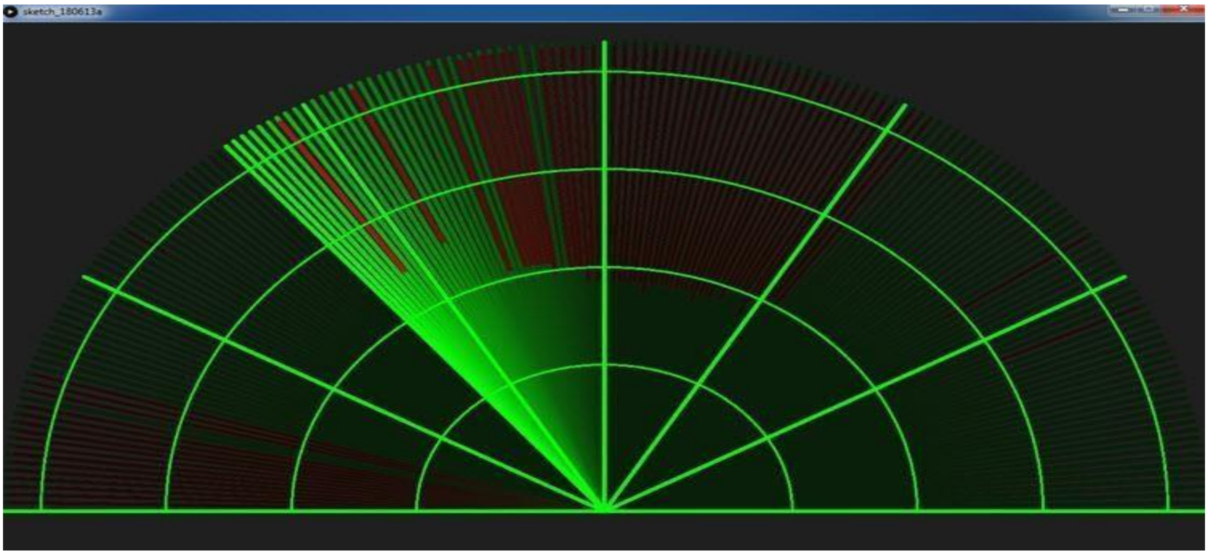
Initially, upload the code to Arduino after making the connections. You can observe the servo sweeping from 00 to 1800 and again back to 00. Since the Ultrasonic Sensor is mounted over the Servo, it will also participate in the sweeping action.

Now, open the processing application and paste the above given sketch. In the Processing Sketch, make necessary changes in the COM Port selection and replace it with the COM Port number to which your Arduino is connected to.

If you note the Processing Sketch, I have used the output display size as 1280×720 (assuming almost all computers now-a-days have a minimum resolution of 1366×768) and made calculation with respect to this resolution.

In the future, I will upload a new Processing sketch where you can enter the desired resolution (like 1920×1080) and all the calculations will be automatically adjusted to this resolution.

Now, run the sketch in the Processing and if everything goes well, a new Processing window opens up like the one shown below.



A Graphical representation of the data from the Ultrasonic Sensor is represented in a Radar type display. If the Ultrasonic Sensor detects any object within its range, the same will be displayed graphically on the screen.

## CONCLUSIONS:-

This project aims on the use of Ultrasonic Sensor by connected to the Arduino UNO R3 board and the signal from the sensor further provided to the screen formed on the laptop to measure the presence of any obstacle in front of the sensor as well as determine the range and angle at which the obstacle is detected by the sensor.

## REFERENCES

[1] http://www.arduino.cc/

[2] http://www.arduinoproducts .cc/
[3] http://www.atmel.com/atmega328/
[4] http://en.wikipedia.org/wiki/File:16MHZ_Crystal.jpg
[5]
http://www.google.co.in/imgres?imgurl=http://www.electrosome.com/
wp content/uploads/2012/06/ServoMotor.gif&imgrefurl=http://www.el
ectrosome.com/tag/se rvo
motor/&h=405&w=458&sz=67&tbnid=rcdlwDVt_x0DdM:&tbnh=100&tbn
w=113&zoo m=1 &usg=
6J2h0ZocdoSMrS1qgK1I2qpTQSI=&docid=lEfbDrEzDBfzbM&sa=X&ei=a_
OKU vTbD8O5rgeYv4DoDQ&ved=0CDwQ9QE

[6]
http//:www.sproboticworks.com/ic%20pin%20configuration/7805/Pinout.jpg/
[7] http://www.sproboticworks.com/ic%ultrasonicsensor%20pinout.jpg
[8] http://www.instructables.com/id/ ATMega328-using-Arduino-/
[9] http://www.motherjones.com/files/blog_google_driverless_car.jpg
[10]
http://www.google.co.in/imgres/Radar_antenna.jpg&w=546&h=697&ei=wuuK
[11] http://www.radomes.org/museum/photos/equip/ANSPS17.jpg [12]
http://www.wired.com/dangerroom/2011/07/ suicide- bombers-from100-yards/