Bansilal Ramnath Agarwal Charitable Trust's

VISHWAKARMA INSTITUTE OF TECHNOLOGY

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

(2018-2019)

Project Report
On

Radar Application Using Ultrasonic Sensor and Processing IDE

Master of Technology In
Computer Science and Engineering

Submitted By:

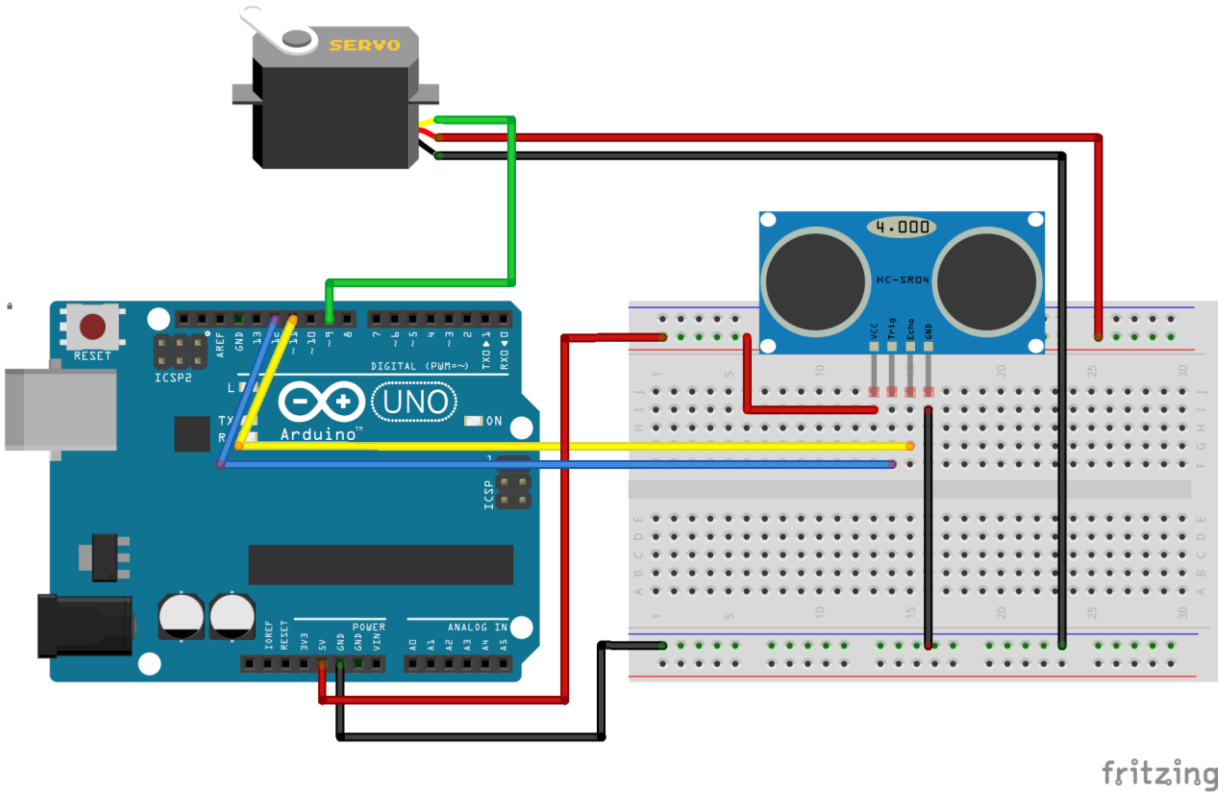
Yashowardhan Dole (Gr. No: 118M0102) Roll No: 4
Girish Karadas (Gr. No: 118M0091) Roll No: 10
Arshadhafeez Khan (Gr. No: 118M0113) Roll No: 12

Under the Guidance of:
Asst. Prof. Sangeeta Kumari

PROBLEM STATEMENT

Development of a Radar Application using Arduino Uno, Ultrasonic Sensor and Processing IDE by sensing objects in the surrounding and determining distance to it using sonar.

BLOCK DIAGRAM



COMPONENTS

Arduino Uno:

Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

Ultrasonic Sensor Module HC-SR04:

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. It comes complete with ultrasonic transmitter and receiver module.

Servomotor:

A Servomotor is a rotatory actuator or linear actuator that allows for precise control of angular or linear positon, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback.

CODE:

1) servo_sonar.ino

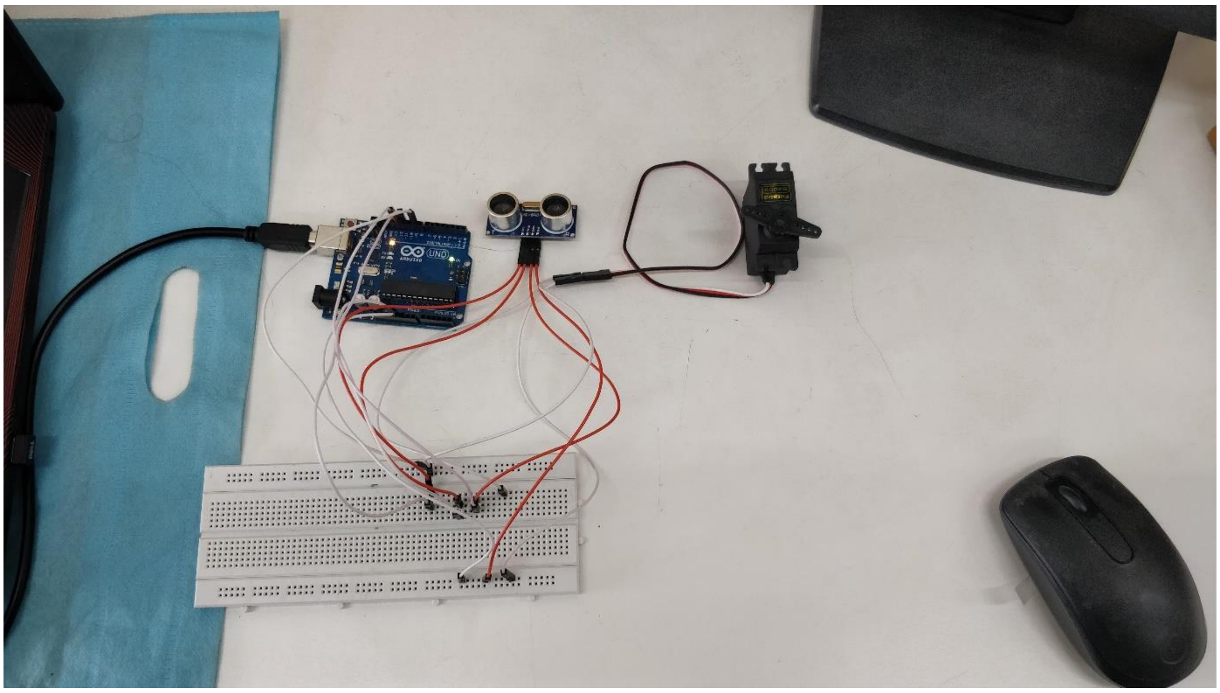
```
#include<Servo.h>
int trigPin=12;
int echoPin=11;

long duration;
int distance;
Servo servo;

void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
  servo.attach(9);
}

void loop()
{
  for(int i=15;i<=165;i++)
  {
    servo.write(i);
    delay(100);
    distance=calculateDistance();

    Serial.print(i);
    Serial.print(",");
    Serial.print(distance);
    Serial.print(".");
  }
  for(int i=165;i>15;i--)
  {
    servo.write(i);
    delay(100);
    distance=calculateDistance();

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

int calculateDistance()
{
  digitalWrite(trigPin,LOW);
  delayMicroseconds(2);
  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;
}
```

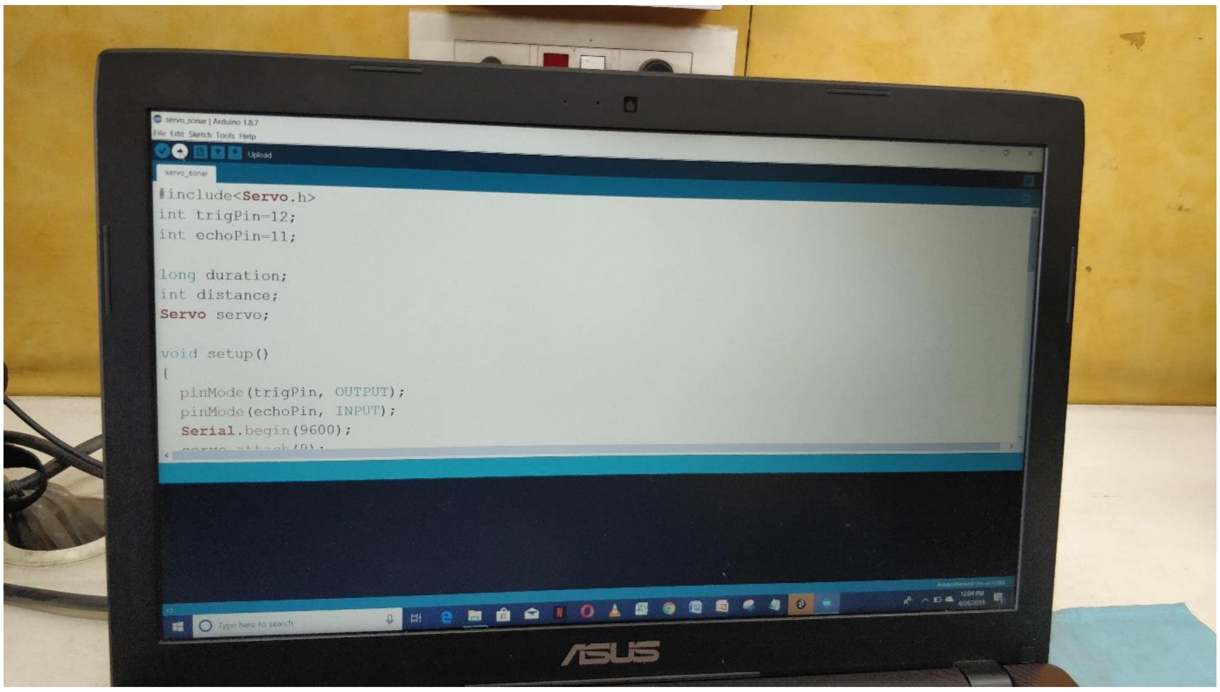
2) sonar.pde

```
import processing.serial.*;
import java.awt.event.KeyEvent;
import java.io.IOException;


Serial myPort;
String angle="";
String distance="";
String data="";
String noObject;
float pixelsDistance;
int iAngle,iDistance;
int index1=0,index2=0;
```

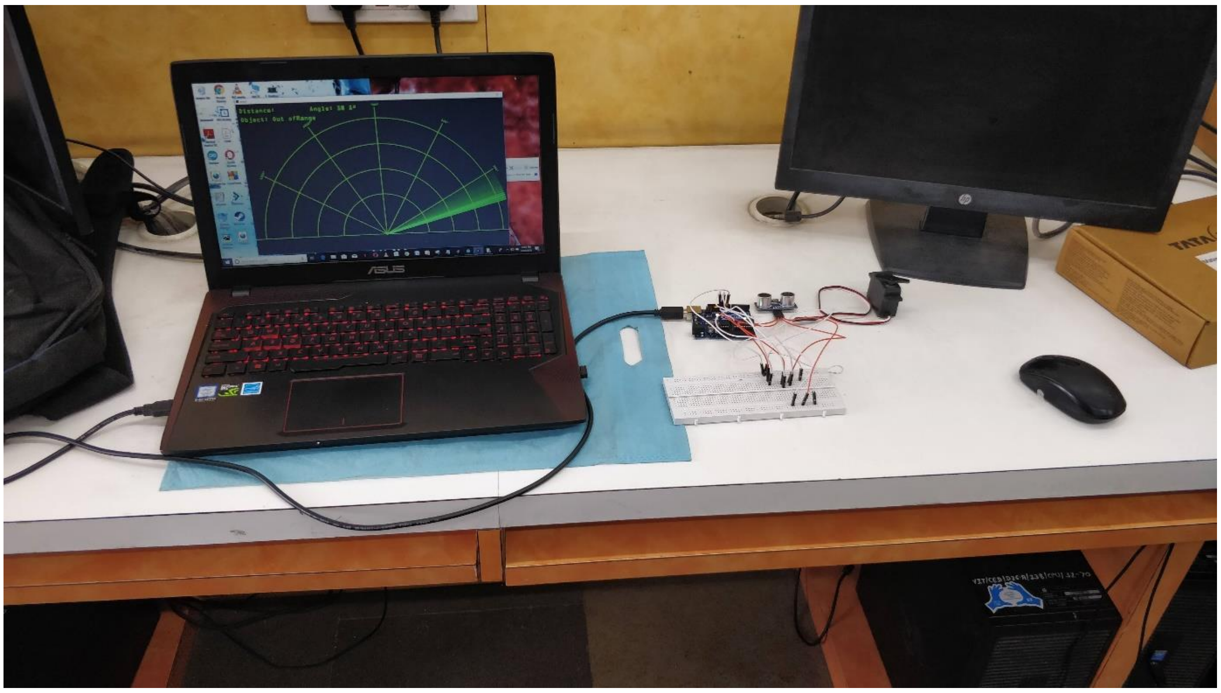
```
PFont orcFont;

void setup()
{
  size(1496,900);
  smooth();
  myPort=new Serial(this,"COM4",9600);
  myPort.bufferUntil('.');
  orcFont=loadFont("OCRAExtended-30.vlw");
}

void draw()
{
  fill(98,245,31);
  textFont(orcFont);
  noStroke();
  fill(0,4);
  rect(0,0,width,1010);

  fill(98,245,31);

  drawRadar();
  drawLine();
  drawObject();
  drawText();
}

void serialEvent(Serial myPort)
{
  data=myPort.readStringUntil('.');
  data=data.substring(0,data.length()-1);

  index1=data.indexOf(",");
  angle=data.substring(0,index1);
  distance=data.substring(index1+1,data.length());

  iAngle=int(angle);
  iDistance=int(distance);
}

void drawRadar(){
  pushMatrix();
  translate(760,800);
  noFill();
  strokeWeight(2);
  stroke(98,245,31);
  arc(0,0,1400,1400,PI,TWO_PI);
  arc(0,0,1100,1100,PI,TWO_PI);
  arc(0,0,800,800,PI,TWO_PI);
  arc(0,0,500,500,PI,TWO_PI);
  line(-760,0,760,0);
  line(0,0,-760*cos(radians(30)),-760*sin(radians(30)));
  line(0,0,-760*cos(radians(60)),-760*sin(radians(60)));
  line(0,0,-760*cos(radians(90)),-760*sin(radians(90)));
  line(0,0,-760*cos(radians(120)),-760*sin(radians(120)));
  line(0,0,-760*cos(radians(150)),-760*sin(radians(150)));
  line(-760*cos(radians(30)),0,760,0);
  popMatrix();
}


void drawObject() {
  pushMatrix();
  translate(760,800); // moves the starting coordinats to new location
  strokeWeight(9);
  stroke(255,10,10); // red color
  pixelsDistance = iDistance*22.5; // 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(pixelsDistance*cos(radians(iAngle)),-
pixelsDistance*sin(radians(iAngle)),750*cos(radians(iAngle)),-750*sin(radians(iAngle)));
  }
  popMatrix();
}

void drawLine()
{
  pushMatrix();
  strokeWeight(9);
  stroke(30,250,60);
  translate(760,800);
  line(0,0,750*cos(radians(iAngle)),-750*sin(radians(iAngle)));
  popMatrix();
}

void drawText()
{
  pushMatrix();
  if(iDistance>40)
  {
    noObject="Out ofRange";
  }
  else{
    noObject="In Range";
  }
  fill(0,0,0);
  noStroke();
  rect(0, 1010, width, 1080);
  fill(98,245,31);
  textSize(25);
  text("10cm",1180,990);
  text("20cm",1380,990);
  text("30cm",1580,990);
  text("40cm",1780,990);
  textSize(30);
  text("Object: " + noObject,15,100);
  text("Angle: " + iAngle +" Â°",400, 50);
  text("Distance: ", 13,50);
  if(iDistance<40) {
  text("        " + iDistance +" cm",30,50);
  }
  textSize(15);
  fill(98,245,60);
  translate(761+760*cos(radians(30)),782-760*sin(radians(30)));
  rotate(-radians(-60));
  text("30Â°",0,0);
  resetMatrix();
  translate(754+760*cos(radians(60)),784-760*sin(radians(60)));
  rotate(-radians(-30));
  text("60Â°",0,0);
  resetMatrix();
  translate(745+760*cos(radians(90)),790-760*sin(radians(90)));
  rotate(radians(0));
  text("90Â°",0,0);
  resetMatrix();
  translate(735+760*cos(radians(120)),803-760*sin(radians(120)));
  rotate(radians(-30));
  text("120Â°",0,0);
  resetMatrix();
  translate(740+760*cos(radians(150)),818-760*sin(radians(150)));
  rotate(radians(-60));
  text("150Â°",0,0);
  popMatrix();
}
```

Results and Monitor Outputs:





```
#include<Servo.h>
int trigPin=12;
int echoPin=11;

long duration;
int distance;
Servo servo;

void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
```


Distance:
Angle: 18 °
Object: Out ofRange
ASUS
TATA