PaperPass.net

Title

report ioe plagcheck

14%	13%	3%
SIMILARITY INDEX	ACADEMIC	INTERNET

Date:	2022-04-28 15:51:02(-05:00 UTC)
Report ID:	626af064ec286cabc
Word count:	2628
Character count:	13295

Similar sources

1	● Design of Unmanned Aerial Systems ● Mohammad H. Sadraey Others	**0.9%**
2	● Microcontroller ● Design of Unmanned Aerial Systems,2020 Others	**0.7%**
3	● CONSTRUCTION AND TESTING OF THE MEASUREMENT SYSTEM FOR EXCAVATOR PRODUCTIVITY ● Mario Klanfar,Vjekoslav Herceg,Dalibor Kuhinek,Kristijan Sekulić ● Rudarsko-geološko-naftni zbornik,2019 Academic	**0.6%**
4	● Obstacle-avoiding robot with IR and PIR motion sensors ● R Ismail,Z Omar,S Suaibun ● IOP Conference Series: Materials Science and Engineering,2016 Academic	**0.5%**
5	● Book reviews [8 books reviewed] ● John Shea ● IEEE Electrical Insulation Magazine,2014 Academic	**0.5%**
6	● Sign language translator and gesture recognition ● Mohammed Elmahgiubi,Mohamed Ennajar,Nabil Drawil,Mohamed Samir Elbuni ● 2015 Global Summit on Computer & Information Technology (GSCIT),2015 Others	**0.5%**
7	● An Arduino board with ultrasonic sensor investigation of simple harmonic motion ● A Buachoom,A Thedsakhulwong,S Wuttiprom ● Journal of Physics: Conference Series,2019 Academic	**0.5%**
8	● Lights! speed! action! ● Erik Brunvand ● ACM SIGGRAPH 2013 Courses on - SIGGRAPH '13,2013 Others	**0.5%**
9	● Web-Based Design and Implementation of Smart Home Appliances Control System ● Samuel Bimenyimana,Aimable Ishimwe,Godwin Norense Osarumwense Asemota,Cecilia Messa Kemunt... ● IOP Conference Series: Earth and Environmental Science,2018 Academic	**0.5%**
10	● An Intelligent Walking Stick for the Visually-Impaired People ● Nadia Nowshin,Sakib Shadman,Sakib Shadman,Saha Joy,Saha Joy,Sarker Aninda,Sarker Aninda,Is... ● International Journal of Online Engineering (iJOE),2017 Academic	**0.5%**
11	● An Arduino Investigation of Simple Harmonic Motion ● Calin Galeriu,Scott Edwards,Geoffrey Esper ● The Physics Teacher,2014 Academic	**0.5%**
12	● Measurement of Wind Parameters ● Methods of Measuring Environmental Parameters,2014 Others	**0.4%**

13	• A Survey on Odometry for Autonomous Navigation Systems • Sherif A. S. Mohamed,Mohammad-Hashem Haghbayan,Tomi Westerlund,Jukka Heikkonen,Hannu Tenhu... • IEEE Access,2019 Academic	0.4%
14	• bin: The intelligent and entertaining bin • Mohammad Wasiul Haq,Nolok Noor Prem,Md. Khalilur Rhaman • 2014 17th International Conference on Computer and Information Technology (ICCIT),2014 Others	0.4%
15	• An enhancement in conventional potential field using a fuzzy system for navigation of a sumo robot • Carlos Erlan Olival Lima,Francisco Marcelino Almeida de Araujo,Mario Bibiano da Silva,Anto... • 2013 16th International Conference on Advanced Robotics (ICAR),2013 Others	0.4%
16	• radar system using arduino and ultrasonic sensor • https://www.researchgate.net/publication/354715482_RADAR_SYSTEM_USING_ARDUINO_AND_ULTRASONIC_SENSOR Internet	0.4%
17	• Design and Analysis of a Scale-Sized Electromechanical Actuator for Unsteady Condition Monitoring Applications • Jason R. Kolodziej,William S. Craig • Volume 2: Diagnostics and Detection; Drilling; Dynamics and Control of Wind Energy Systems; Energy Harvesting; Estimation and Identification; Flexible and Smart Structure Control; Fuels Cells/Energy Storage; Human Robot Interaction; HVAC Building Energy Management; Indu...,2015 Others	0.3%
18	• An Automation System for Controlling Streetlights and Monitoring Objects Using Arduino • Zain Mumtaz,Saleem Ullah,Zeeshan Ilyas,Naila Aslam,Shahid Iqbal,Shuo Liu,Jehangir Meo,Hamz... • Sensors,2018 Academic	0.3%
19	• Automatic Parking Vehicle System • Honghong Liu,Gene Liao,Chih-Ping Yeh,Jimmy Chen • 2016 ASEE Annual Conference & Exposition Proceedings Others	0.3%
20	• ATmega8 Based Obstacle Avoiding Helicopter • Suhas Divakar • Advanced Materials Research,2011 Academic	0.3%
21	• The Basics • John-David Warren,Josh Adams,Harald Molle • Arduino Robotics,2011 Others	0.3%
22	• An educational Arduino robot for visual Deep Learning experiments • He Hu • International Journal of Intelligent Robotics and Applications,2019 Academic	0.3%
23	• Exploring Arduino® • Jeremy Blum Others	0.3%
24	• Using Transistors and Driving DC Motors • Exploring Arduino®,2019 Others	0.3%

25	• Arduino Microcontroller Processing for Everyone! Third Edition • Steven F. Barrett • Synthesis Lectures on Digital Circuits and Systems,2013 Academic	**0.3%**
26	• What Are Jumper Wires: Know by Colour, Types and Uses • https://www.wiltronics.com.au/wiltronics-knowledge-base/what-are-jumper-wires/ Internet	**0.3%**
27	• Radar System Using Ultrasonic Sensor - Arduino Project Hub • https://create.arduino.cc/projecthub/rithikkaparthi/radar-system-using-ultrasonic-sensor-6c71fa?ref=part&ref_id=8233&offset=16 Internet	**0.3%**
28	• What is Servo Motor?- Definition, Working And Types • https://www.engineeringchoice.com/servo-motor/ Internet	**0.3%**
29	• Hexapod Performance • Priyanka Thirumalesh,P Malarvezhi,Ankita Mahato,Meghna Choudhury • International Journal of Engineering & Technology,2018 Academic	**0.2%**
30	• Kinematic Model Building and Servo Parameter Identification of 3-HSS Parallel Mechanism • Zhi-yong Yang,Jiang Wu,Tian Huang,Yan-bing Ni • Frontiers of Mechanical Engineering in China,2006 Academic	**0.2%**
31	• Underconstrained planar cable-direct-driven robots: A trajectory planning method ensuring positive and bounded cable tensions • Alberto Trevisani • Mechatronics,2010 Academic	**0.2%**
32	• Arduino Microcontroller: Processing for Everyone! Second Edition • Steven F. Barrett • Synthesis Lectures on Digital Circuits and Systems,2012 Academic	**0.2%**
33	• Sensor Fusion for Joint Kinematic Estimation in Serial Robots Using Encoder, Accelerometer and Gyroscope • Benigno Munoz-Barron,Jesus R. Rivera-Guillen,Roque A. Osornio-Rios,Rene J. Romero-Troncoso... • Journal of Intelligent & Robotic Systems,2014 Academic	**0.2%**
34	• Smart Walking Stick for Visually Impaired People • Shravan Mohite,Abhishek Patel,Milan Patel,Vaishali Gaikwad (Mohite) • SSRN Electronic Journal,2018 Academic	**0.2%**

1. Introduction:

RADAR is an object detection and tracking technology that uses radio waves to determine the direction, range, altitude and speed of an object. Radar frameworks or systems are available in a variety of forms, sizes, and performance levels. At airports, some radars are utilised by aviation authorities, while others are employed by law enforcement. Applications include long-range surveillance and early-warning systems [1].There are a few options. There are a variety of ways to present radar operational data. Radar systems that have been modified and have improved technology for handling the systems are also available. At a higher level, these modified systems are utilised to gather or extract useful or crucial information [2].

The components listed below are relevant to our recommended system's functioning philosophy : an ultrasonic sensor tied to the digital input and output pins of a microcontroller (we chose Arduino). The Digital output and input pins are also linked/connected to the servo motor. Our ultrasonic sensor and servo motor linked at the same time, therefore our servo motor rotates near its axis when it rotates from 0 to 180 degrees from extreme right to extreme left [3]. To show the data (distanceance and angle) on a computer screen, we utilise software called "Processing Development Environment" [1].

2. Review of Literature:

Following a review of some of the articles on the use of ultrasonic sensors and ARDUINO, it was discovered that this concept is widely researched and is a mainstream concept that is continuing in development. The techniques used were not only effective and reliable, but also cost-effective [5]. Not only that, but additional extremely beneficial ultrasonic sensor uses were also noticed. This paper discusses a monitoring system that uses an ultrasonic sensor and a microcontroller to measure the speed of waves and the high of a river (Arduino). If the river can't handle the volume of water, the water will overflow and flood the land, causing a phenomena known as a flood or surge. We can avoid flooding by identifying the high of the water and monitoring its speed earlier. If we discover an issue early enough, we can remedy it before it becomes a catastrophe. A basic water level was used to test the system, and it was revealed that ultrasonic had a 96.6 percent accuracy. However, when it is used in rivers, there are numerous inaccuracies due to the various types of water levels caused by heavy waves and water speed, as well as the floating of heavy things. Unlike previous testing results, the author focused this investigation on the rate of water improvement or modification, as well as the amount of water in floods. When the

Arduino was employed as the application's controller, the test was completed. For further study, data on the depth level and water speed of this system will be uploaded to an onlinee database server, which will be analysed on a regular basis [8]. The goal of the study is to offer a system for intelligent driver monitoring and vehicle control. The author discusses after analysing some of the most ordinary or customarily common causes of accidents nowadays. These include drunkenness, carelessness, exhaustion, or medical illness on the part of the driver. The framework's different components are checked and determined to be in excellent working condition, including motors, relays, the power unit, and the ESP8299 module. When another vehicle approaches his automobile, an ultrasonic sensor alerts the driver.The status of the driver can be determined with the help of sensors installed in the vehicle, and the proprietor can be updated on the subtle elements. This technology solves all of the factors that have caused earlier technologies created for this purpose to fail, making it more helpful, efficient, cost-effective, and time-consuming [7].

The authors of this research study describe how to detect radio waves and track or range them using a radar set made up of components such as an ultrasonic sensor, a servo motor, and an Arduino. The author explains how the creation of an ultrasonic distance measurer solved the lineear measurement problem, which made it difficult to quantify distance between specific objects. It enables noncontact measurements to be taken. This system is a highly helpful radar system, according to the author, since it can read or track the distanceance and angle of an obstruction and show it on a monitor screen. The ultrasonic sensor was put on top of the servomotor to detect impediments in the right and left directions from 0 to 180 degrees. This research demonstrates one of the most common of influential method for spotting barriers in a familiar area. This system is powered by a mobile camera that runs on Android. Visually impaired people have a comparatively harder time detecting obstacles and navigating through them while walking. They employ sticks to solve this problem, however this method, or the Arduino-based radar system strategy, is not the best way to go about it. Object indication or detectors can help individuals avoid accidents or collisions, or they can help with correct map reading. Indoor mapping is the major goal of the algorithm highly suggested in this thorough study. All distanceinctive floors are considered in the indoor environment, and a single picture is retained or saved for each distanceinctive level. These floor images will serve as a reference. This algorithm, according to its designer, is 96 percent accurate and functions in real time. To overcome the challenges, we not only use a SONAR sensor but also a laser camera in order to overcome our problems and optimise the project. This study [9] proposed a method for detecting hindrance in a known condition utilising an android-based flexible camera that scans a pre-selected region for impediment location before capturing the image.

3. Report on Present Investigation :

The development life cycle of the Radar project is depicted in the diagram below, which includes individual component design, testing, installation, and overall system testing are all tasks that must be completed. The phases indicated below can be used to break down these steps :

a) Hardware System Design.
b) Hardware Circuit Design.
c) Hardware System implementation.
d) Hardware unit testing.
e) GUI System Design.
f) GUI System Implementation.
g) GUI unit testing.
h) Entire system integration.
i) Entire system testing.

Hardware description :

Ultrasonic Sensor :

Fig 1. Ultrasonic Sensor

Ultrasonic and sonar sensors function the same way, it uses sound waves in order to measure the distanceance between objects.. The sender sends sound waves in a given frequency and direction, and the receiver waits for them to return. By measuring the time it takes for a sound wave to return, we may determine how far away something is.

Servo motor :

Fig 2. Servo Motor

A servomotor is a whirling motor with precision control over angular position, velocity, and acceleration. It is built using a suitable motor and a position feedback sensor. It also necessitates the use of a complex controller, which is usually a separate module dedicated to servomotors. Servomotors are not a distanceinct type of motor, but rather use servomechanism to achieve closed loop control with a standard open loop motor, according to its essential operating concept. Robotics, CNC machines, and automated manufacturing are just a few examples of where servomotors are used.

Arduino :

Fig 3. Arduino Uno

The Arduino platform is a free of charge and open source hardware and software electronics platform. Thus, making it simpler and easier to not only write code but also upload it to the Arduino board. Windows, Mac OS X, and Linux are all supported. The environment is written in Java and uses open-source tools such as Processing as a foundation. This application can be optimised and run on any Arduino board. The Arduino software IDE also includes a text editor for writing code, a message box, a text terminal, and a toolbar along with basic function buttons. By uploading and downloading code, it links to Arduino and Genuine devices. Sketches are programmes developed to be executed in the Arduino programming language.

Bread board:

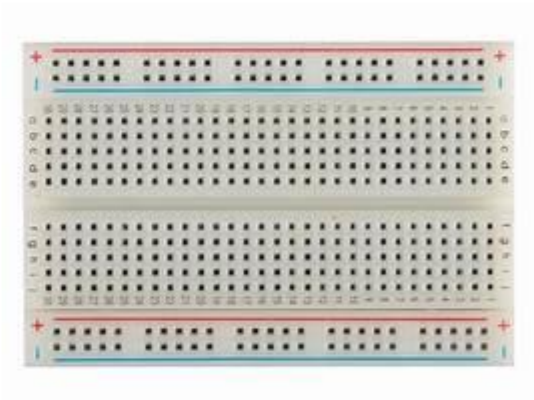


Fig 4. Bread Board

When learning how to create circuits, breadboards are one of the most important components. This lesson will teach you all you need to know about breadboards, including what they are, why they're called that, and how to use one. You should now be able to design a basic circuit on a breadboard and understand how they function.

Jumper wires:

Fig 5. Jumper wires

58%

Jumper wires are cables with connector pins on both ends that are used to connect two sites without soldering. With breadboards and other prototype tools, jumper wires are widely utilised to allow for fast circuit adjustments as needed.

(A) Hardware System Design

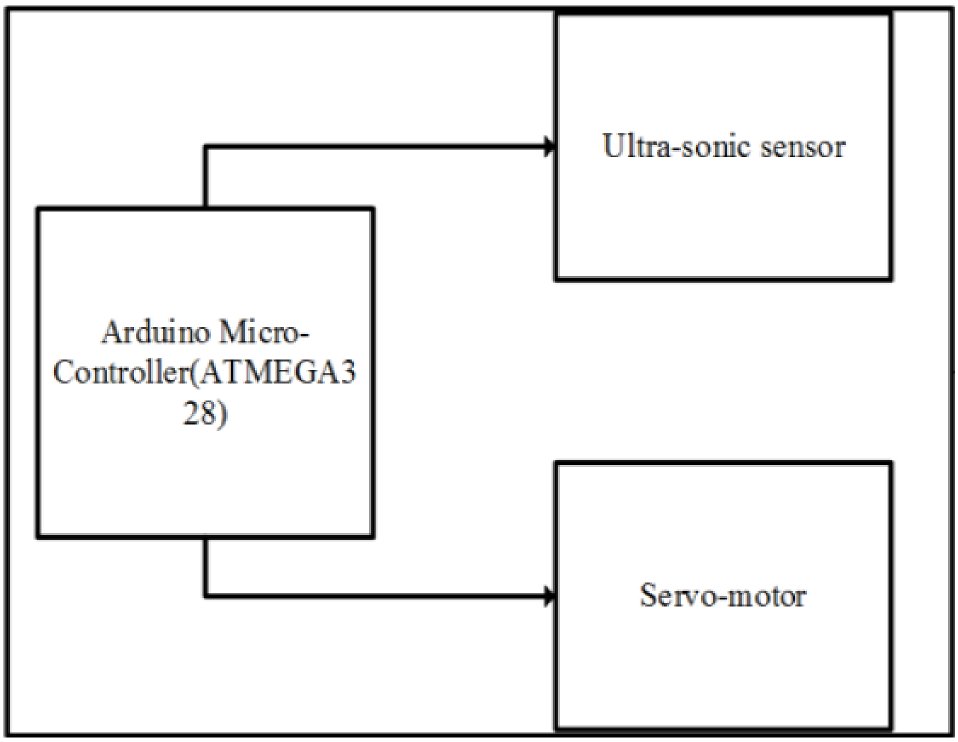


Fig 7. Hardware System Design of Radar System.

The hardware system comprises of three parts: an Arduino, a servo motor, and an ultrasonic sensor. A servo motor is mounded with an ultrasonic sensor, which aids in movement and provides a turning mechanism. The Arduino uno not only controls the Arduino Uno but it also powers it as illustrated in Figure

(B) System circuit design

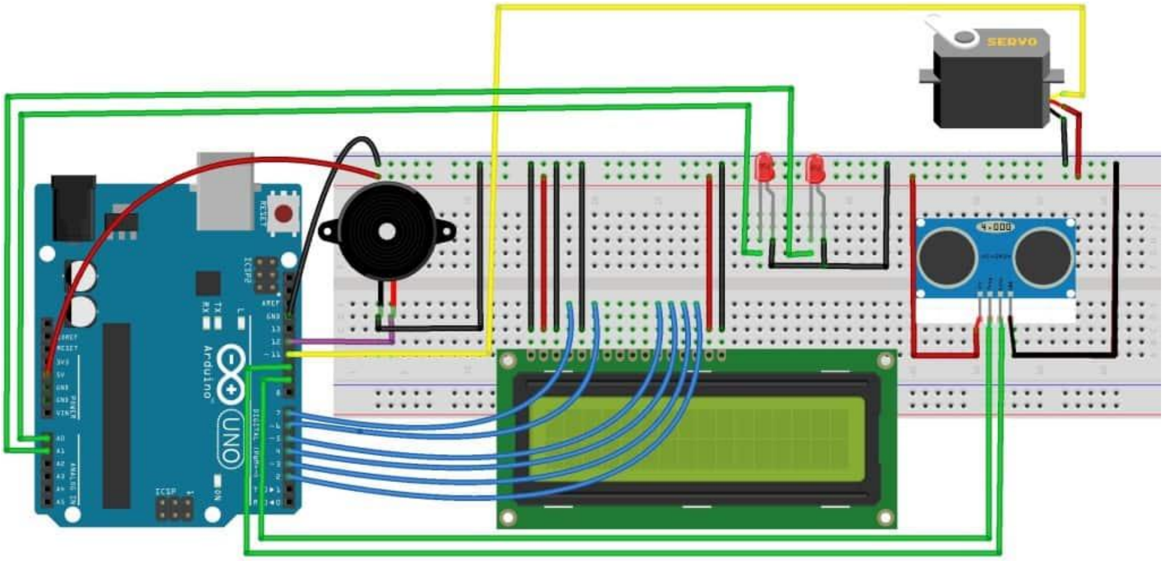


Fig 8. System circuit design

A hardware system design produced in the Fritzing environment is shown in Figure. It shows how several electrical components are linked together. The triggering pins of the ultrasonic sensor are connected to the Arduino's D10 pin, the servo motor's control linee to the Arduino's D8, and the echo pin to the Arduino's D11. The VCC pins of the servo motor and ultrasonic sensor are connected to the Arduino's 5V pin, while the Arduino's ground pin is connected to both the servo motor and ultrasonic sensor's ground pin.

(C) System circuit implementation on bread board –

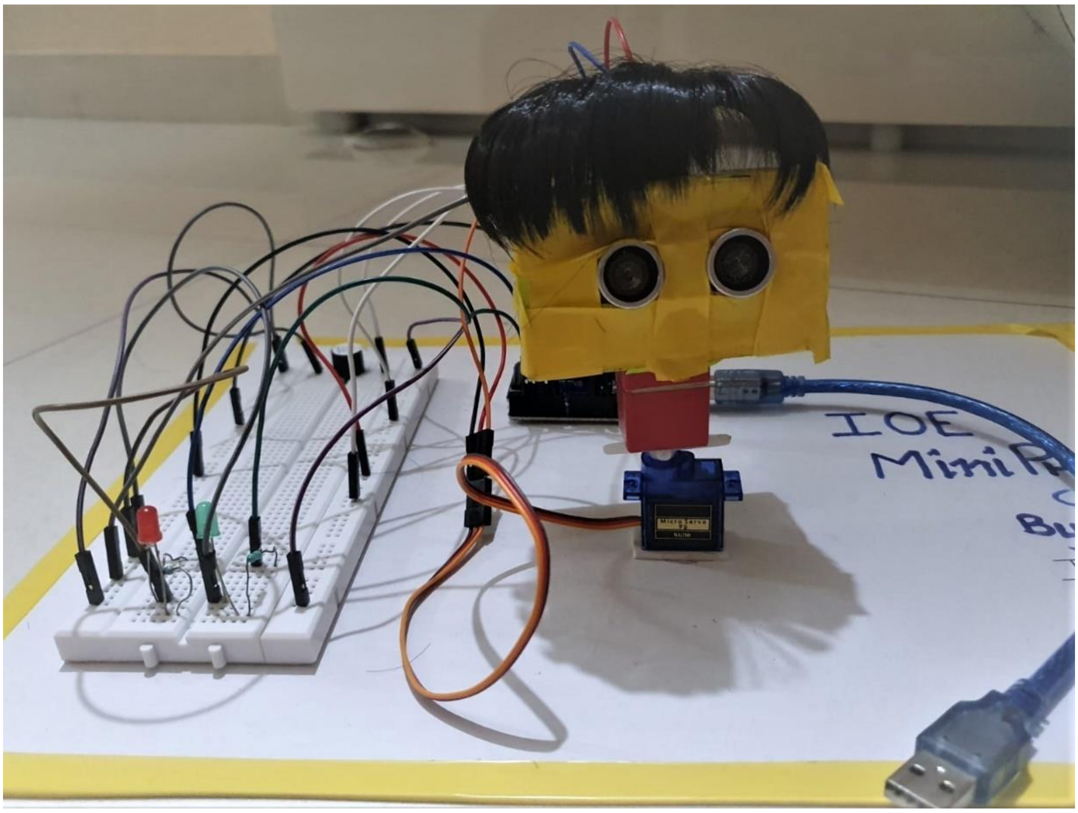


Fig 9. Implementation on bread board

The following diagram depicts the hardware system in its entirety. The ultrasonic servomotor may be seen mounted on a servo motor and positioned above the breadboard. On the other side of the breadboard, Arduino is installed, and the whole connection is formed between them. To keep the Arduino and servo motor from tripping over as the servo motor travels, they are stuck to the breadboard. To create code and upload it to Arduino, the Arduino IDE was utilised. The Arduino code reads the servo motor's location and calculates the distance to the closest item in the route.

The Arduino code is given below:

radar | Arduino 1.8.19

File Edit Sketch Tools Help

radar radar2

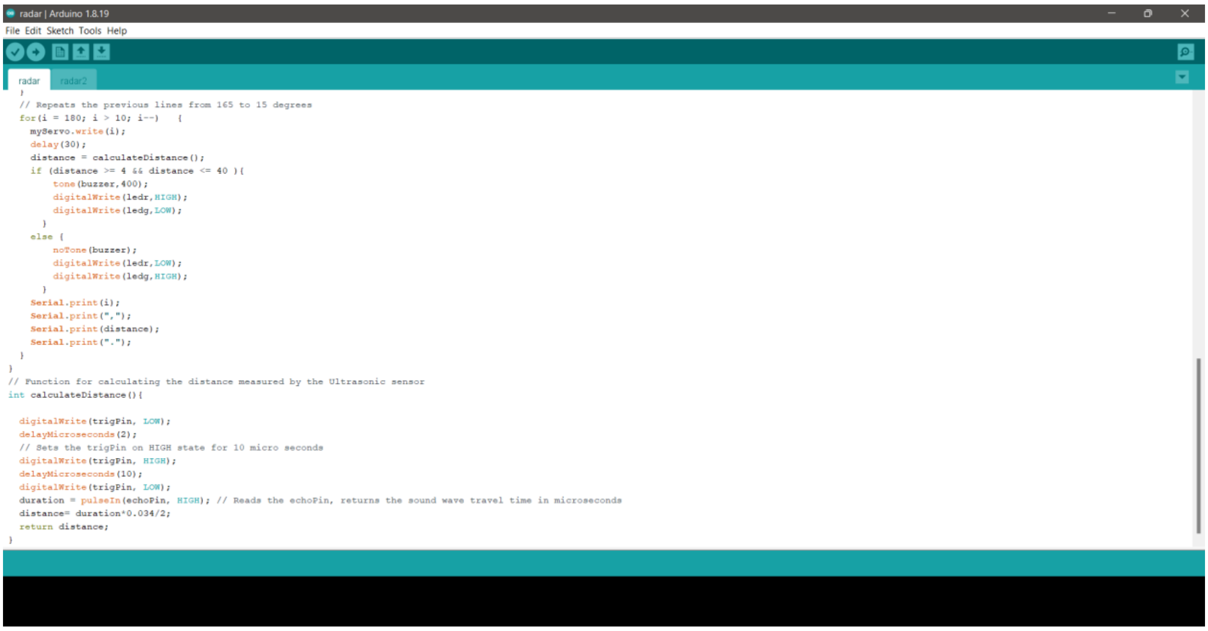
```
// Includes the Servo library
#include <Servo.h>
// Defines Tirg and Echo pins of the Ultrasonic Sensor
const int trigPin = 5;
const int echoPin = 6;

// Variables for the duration and the distance
long duration;
int distance;
int angle = 10;
int i = 10;
int ledr = 3;
int ledg = 2;
int buzzer = 13;
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
  pinMode(3,OUTPUT);
  pinMode(2,OUTPUT);
  pinMode(13,OUTPUT);
  Serial.begin(9600);
  myServo.attach(11); // Defines on which pin is the servo motor attached
}
void loop() {
  // rotates the servo motor from 15 to 165 degrees
//  tone(buzzer,400);
//  delay(300);
//  noTone(buzzer);
//  delay(300);
//  tone(buzzer,400);
//  delay(300);
  for(i = 10; i < 180; i++)  {
    myServo.write(i);
```

radar | Arduino 1.8.19

File Edit Sketch Tools Help

radar radar2

```
void loop() {
  // rotates the servo motor from 15 to 165 degrees
//  tone(buzzer,400);
//  delay(300);
//  noTone(buzzer);
//  delay(300);
//  tone(buzzer,400);
//  delay(300);
  for(i = 10; i < 180; i++)  {
    myServo.write(i);
//    delay(30);
    distance = calculateDistance();// Calls a function for calculating the distance measured by the Ultrasonic sensor for each degree
    if (distance >= 4 && distance <= 40 ){
        tone(buzzer,1000);
        digitalWrite(ledr,HIGH);
        digitalWrite(ledg,LOW);
      }
    else {
        noTone(buzzer);
        digitalWrite(ledr,LOW);
        digitalWrite(ledg,HIGH);
      }
    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(i = 180; i > 10; i--)   {
    myServo.write(i);
    delay(30);
    distance = calculateDistance();
    if (distance >= 4 && distance <= 40 ){
        tone(buzzer,400);
```



radar | Arduino 1.8.19

File Edit Sketch Tools Help

radar radar2

```
  }
  // Repeats the previous lines from 165 to 15 degrees
  for(i = 180; i > 10; i--)   {
    myServo.write(i);
    delay(30);
    distance = calculateDistance();
    if (distance >= 4 && distance <= 40 ){
        tone(buzzer,400);
        digitalWrite(ledr,HIGH);
        digitalWrite(ledg,LOW);
      }
    else {
        noTone(buzzer);
        digitalWrite(ledr,LOW);
        digitalWrite(ledg,HIGH);
      }
    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;
}
```

Processing code to display graph on laptop:

graph | Processing 3.5.4

File Edit Sketch Debug Tools Help

graph

```java
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 (1920, 1080); // ***CHANGE THIS TO YOUR SCREEN RESOLUTION***
 smooth();
 myPort = new Serial(this,"COM3", 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();
```

graph | Processing 3.5.4

File Edit Sketch Debug Tools Help

graph

```java
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);
```

graph | Processing 3.5.4

File Edit Sketch Debug Tools Help

graph

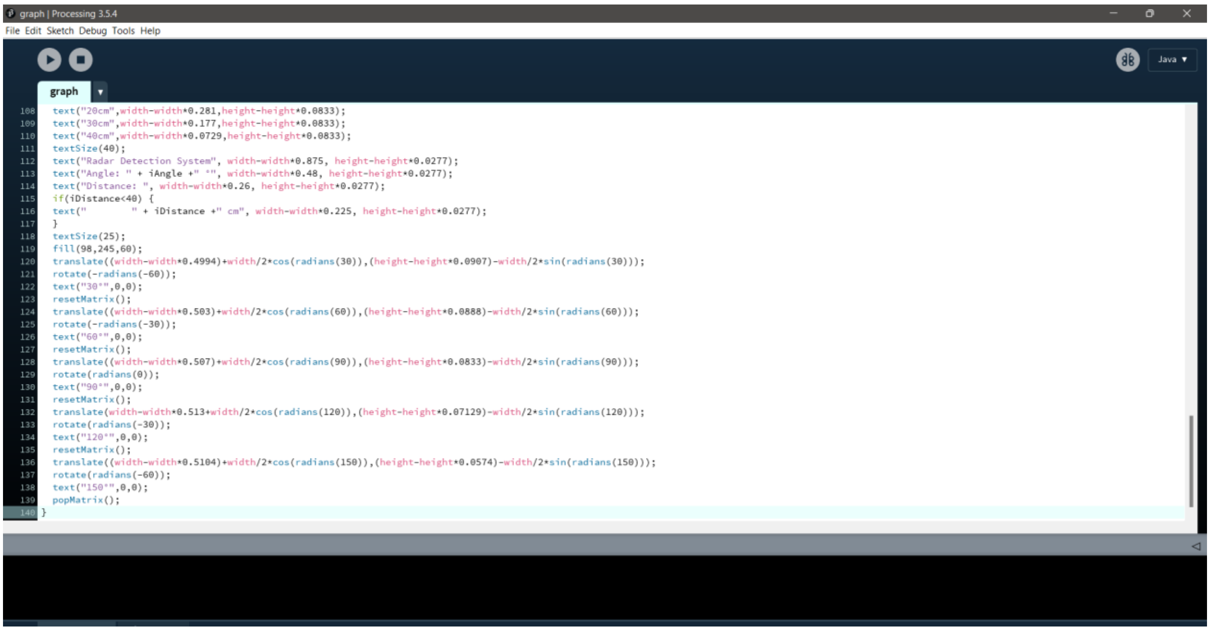
```
void drawRadar() {
  pushMatrix();
  translate(width/2,height-height*0.074); // moves the starting coordinats to new location
  noFill();
  strokeWeight(2);
  stroke(98,245,31);
  // draws the arc lines
  arc(0,0,(width-width*0.0625),(width-width*0.0625),PI,TWO_PI);
  arc(0,0,(width-width*0.27),(width-width*0.27),PI,TWO_PI);
  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();
```

graph | Processing 3.5.4

File Edit Sketch Debug Tools Help

graph

```
}
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 Detection System", 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) {
```



(D)Hardware system testing – The Arduino was connected to the development machine through a wire. Using the Arduino IDE, we were able to get a result in the serial monitor.

(E) Design and implementation of a graphical user interface system The GUI was created using the JAVA programming language and consists of two classes. The radar project's object class represents the objects it meets, such as distance, target/range, and angle/direction of position. The distance () method, angle () method, and location () method all accept needed values like distance and angle and display them on a GUI for simulation. Figures depict a linee sweep from one direction to the other, as well as a smudge in the GUI where ultrasonic sensors detect impediments.

WORKING :

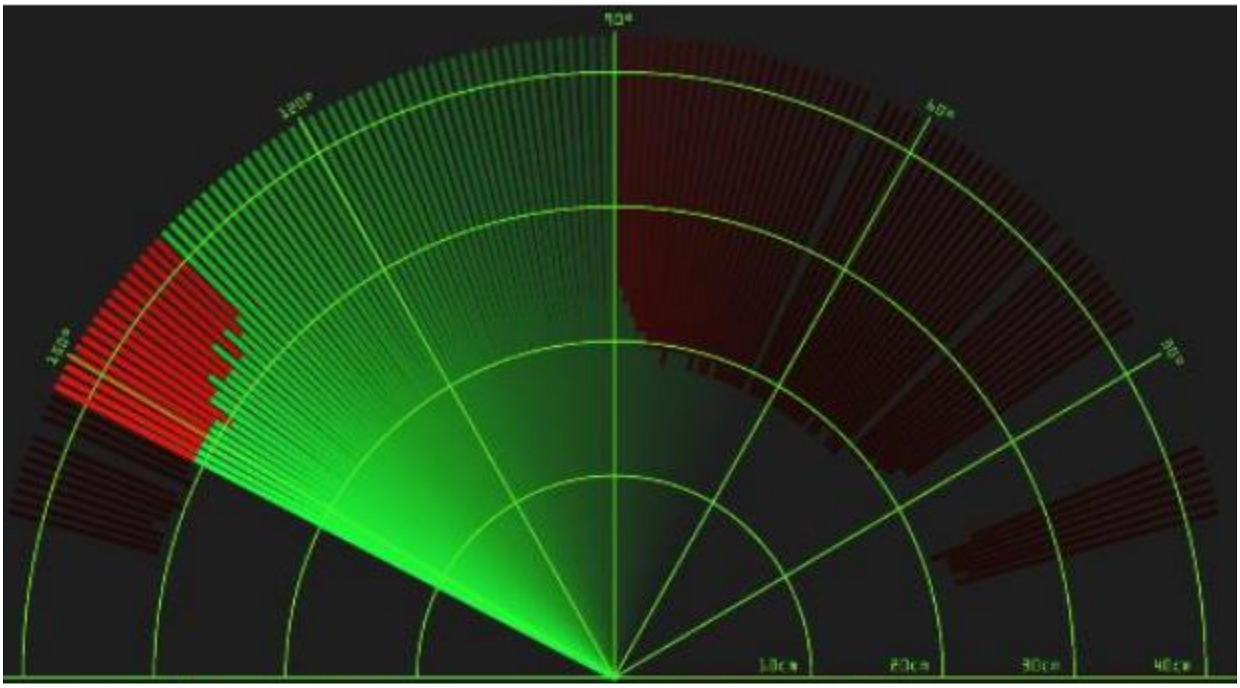
Our design's main goal is to determine the distance, location, and speed of a barrier placed at a specific distance from the sensor. The ultrasonic sensor enables the servo motor to rotate in order to deliver the ultrasonic wave in various directions. This wave travels through air and is reflected back after colliding with any item. This wave is detected by the sensor once again, and its characteristics are examined, with the results shown on the screen as metrics such as object distance and location. The Arduino IDE is used to write and transfer code in Arduino, allowing us to detect the position or angle of a servo motor, which is conveyed over the serial port together with the distance covered by the nearest item in its path. The outcome of all of this labour is displayed in the processing programme, which displays the object's input/output and range [4]. The sensors are implemented in such a manner that an ultrasonic sensor is coupled to the servo motor in order to detect the item and its distance. The ultrasonic sensor and servo motor will be controlled by an Arduino (microcontroller), and both will be powered by the microcontroller [3]. It displays a green light all of the time, indicating that object detection is active, and a buzzer and red light begin as soon as an item approaches it (0 degrees to 180 degrees) across a distance of 4cm to 40cm.

4. Result and Discussions :

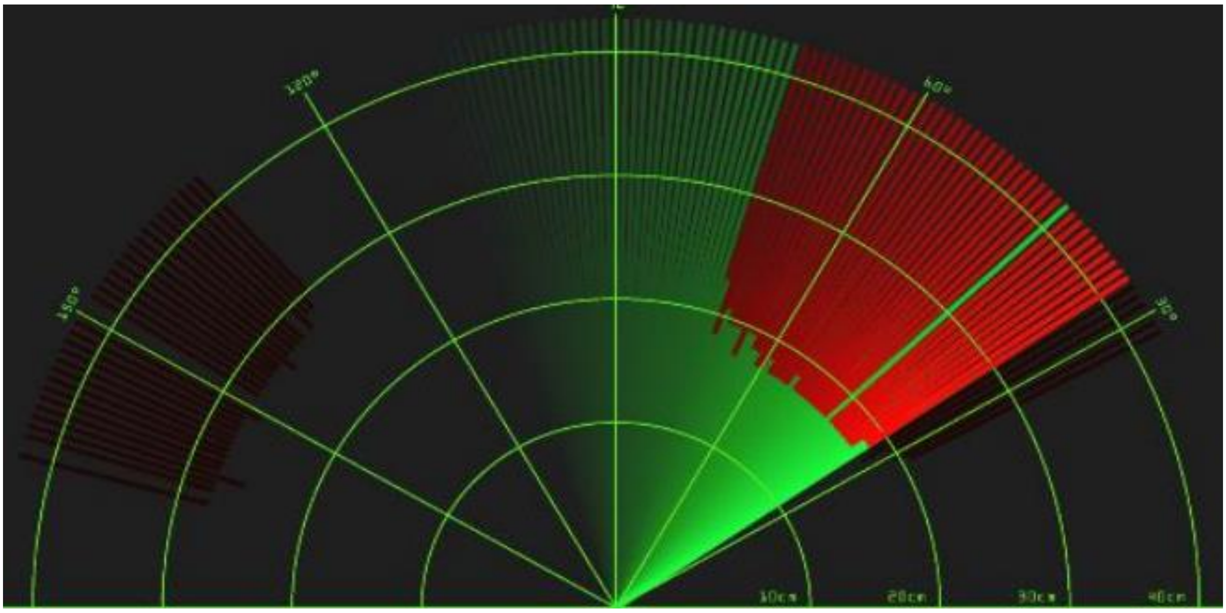
As we have previously stated in this study report , our system is made up of the following components : a servo-motor, an ultrasonic sensor, and a microcontroller (Arduino). The goal of the system is to track the object's distance and angle and to visually display this information. This implies that the system's output should be in graphical form, which will be rendered using processing software. We may get an estimate of the radar's effectiveness by testing items at various highs and seeing how quickly or smoothly it identifies an object and gives us a predicted range of the obstacle [3]. The outcomes of our design's monitor screen when the sensor rotates across the region and finds an impediment are shown in the following image. The existence of an obstruction is shown by the red region, and the angle of incident and distance are presented below.

Testing of the system

a) Object A is kept at 20.5 far from the radar, radar gives the distance 32 cm, so: o error =(32-20.5)/20.5)*100= 3.918% o efficiency 1 = 100-error =94.08%



b) object 2 placed at a distance of 19.3 cm, radar gives the distance 21 cm so: o error = ((21-19.3)/20.3)*100 = 2.44% o efficiency 2 =100-error= 95.55%



After observation and calculation we conclude that this system is 94.815% efficient.

5. Conclusion and future scope of project:

In this Project an Arduino, a servomotor and an ultrasonic sensor were used to create a radar system that can detect the location and distance of any obstacle in its path and transform it into a graphically representable form. This method may be utilised in robotics for object identification and avoidance, as well as intrusion detection for various sizes of locations. The system's range is determined by the type of ultrasonic sensor that is employed. The sensor we utilised was the HC-SR04, which has a range of 2 to 40 cm. Designers now have more control over a variety of sophisticated applications thanks to a variety of improved control mechanisms. The proposed whole-system mapping approach is evaluated in our study on modest principles or scale [9]. The field of "Radar System," which we have chosen for our design, is quite broad, and the technology's future potential is enormous. Radar systems have been deployed or employed in a variety of applications [3]. Because of its security capabilities, this design has a lot of future potential. It has a wide range of uses[4]. Our investigation was restricted and constrained since we constructed a short-range radar. Because the servo motor we used can only rotate to this range, our system can only detect objects from 0 to 180 degrees. As a result of this limitation, our method cannot be utilised to identify larger-scale obstacles in locations or areas. The system's efficiency can be improved by using a 360-degree rotating servo motor. By altering this system and adding a fully 360-degree rotating servo and a higher-ranged ultrasonic sensor, we intend to better our research. We may enhance this system by making it transportable and adding an alert system that activates when an impediment is identified. Additional improvements might include an obstacle-avoiding robot with a surveillance system.