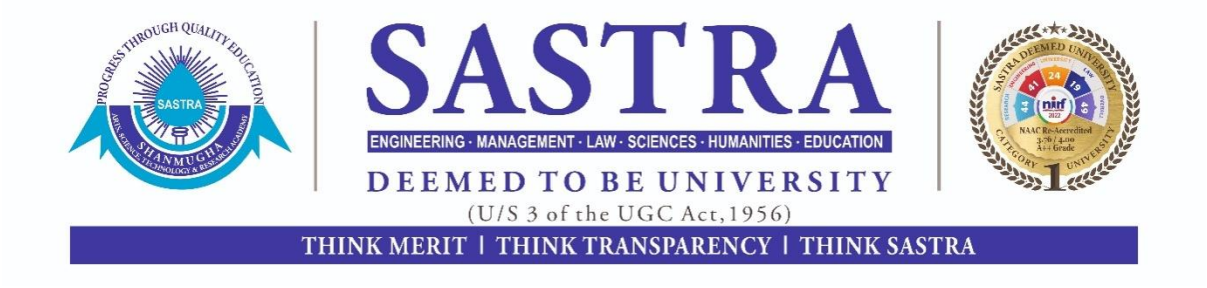


School of Electrical & Electronics Engineering

Project Title: ARDUINO BASED RADAR SYSTEM WITH OBJECT'S RANGE DETECTION

Reg No :124006059&124006058

TEAM MEMBER:

HARIHARAN

&

MANOJ.R. D

Course Code: EIE203

Sensors & Transducer

ABSTRACT- The application of radio detection and ranging in different places such as military installation, commercial use is done with the help of RADAR SYSTEM which uses electromagnetic waves for detection of different physical components such as distance, speed, position, range, direction, size etc which can be either fixed or be in motion. Use of radar system has been developed greatly specially in field of navigation. In this research we study about existing navigation technologies and proposed an Arduino based radar system. It has advantage over other radar system as kit reduces power consumption and connect programmer to wide range or Arduino programmers and open-source code. The system consist a basic ultrasonic sensor placed upon a servo motor which rotates at a certain angle and speed. This ultrasonic sensor is connected to Arduino digital input output pins and servo motor also connected to digital input output pins.

KEYWORDS*- Arduino, Ultrasonic sensor, Servo motor, Simulation*

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 back and after calculation time taken to come back we can determine distance of object. in real life this method is used by bats.

DESIGN IMPLEMENTATION OF RADAR SYSTEM

The figure shown below shows the development life cycle of Radar project which involves various step such as design of different components,

their testing, their implementation and implementation of entire system and their testing. These steps can be enumerated into following stages

a) Hardware System Design.

b) Hardware Circuit Design.

c) Hardware System implementation.

d) Hardware unit testing.

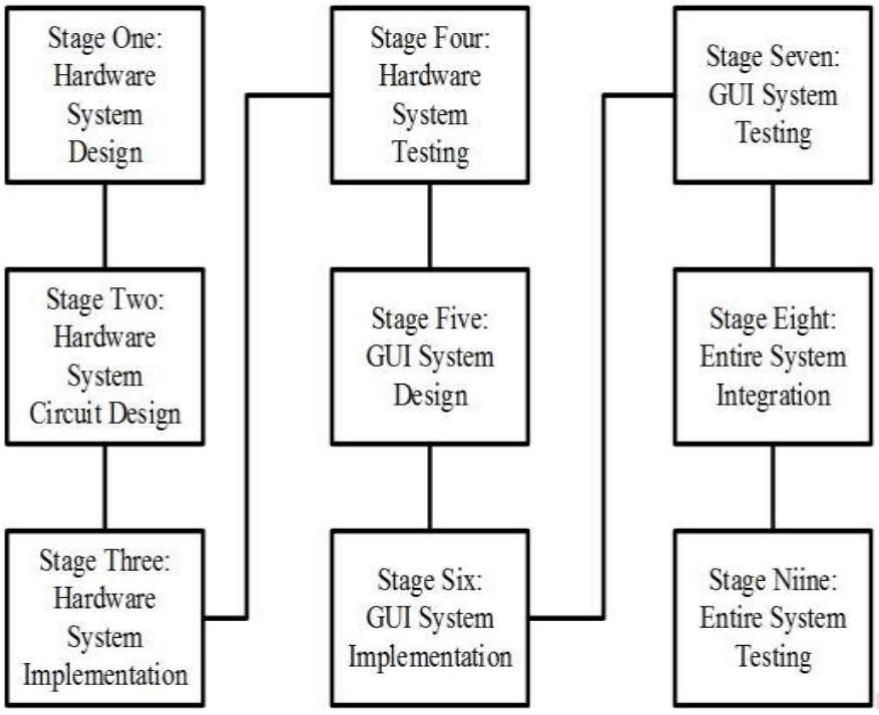
e) GUI System Design.

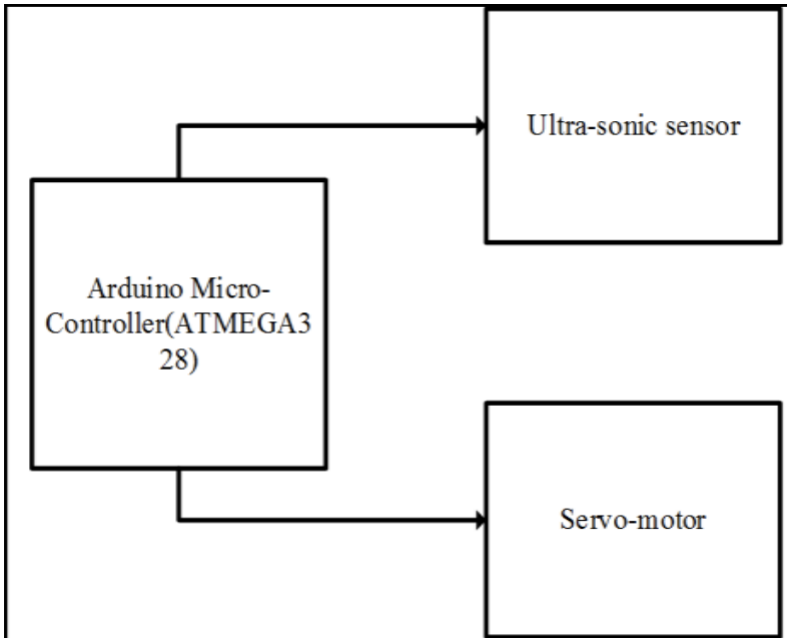
f) GUI System Implementation.

g) GUI unit testing.

h) Entire system integration.

i) Entire system testing





(A)**Hardware system design for Arduino**-

Hardware system consist of basically 3 components named as Arduino, servo-motor, and ultra-sonic sensor. Ultrasonic sensor is mounded upon a servo motor which helps it to move and provide it a turning mechanism. Both ultrasonic sensor and servo motor are controlled and powered by Arduino. As given in above figure 2 we can see both ultrasonic sensor and servo motor is powered by Arduino.

(B) **System circuit design**-

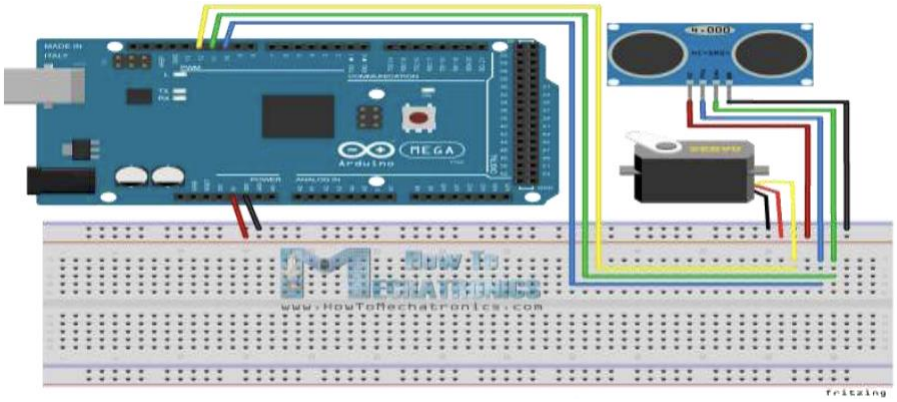
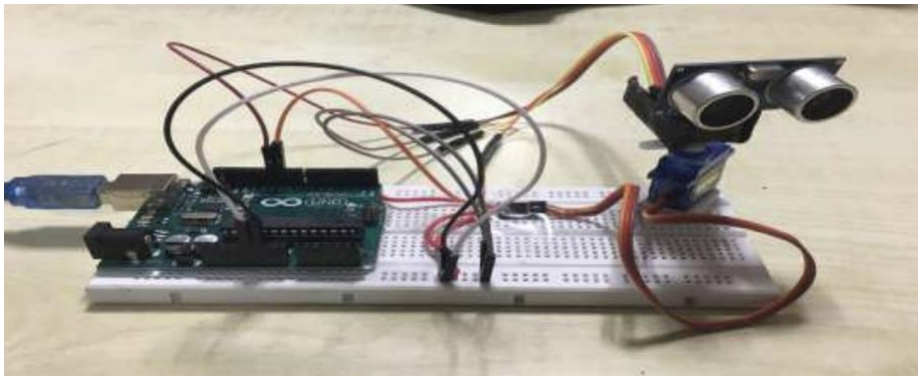


Figure shows hardware system design which was designed using fritzing environment. It shows the connection of different electronics components. In the figure triggering pins of ultrasonic sensor is connected to D8 pin of Arduino, control line of servo motor is connected to D6 pin of Arduino and D7 pin of Arduino is connected to echo pin. VCC pins of servo motor and ultrasonic sensor is connected to 5V pin of

Arduino while ground pin of Arduino is connected to ground pin of both servo motor and ultra-sonic sensor

(C) System circuit implementation on bread board –



Above figure 4 shows complete implementation of hardware system. It can be seen that ultrasonic servomotor is placed upon a servo motor andit is placed above bread board. Arduino is placed in breadboard in other side of the breadboard and entire connection is made between them. Arduino and servo motor are stick to breadboard to stop it from tripping over when servo motor moves. Arduino IDE was used to write code and upload it in Arduino. Arduino code reads position of servo motor and calculate distance of nearest object in the path.

The Arduino code for the system is highlighted below;

```
// Includes the Servo library
#include <Servo.h>.
// Defines Tirg and Echo pins of the Ultrasonic
Sensor
const int trigPin = 8;
const int echoPin = 7;
// 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(6); // 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();
Serial.print(i); // Sends the current degree into the
Serial Port
Serial.print(",");
Serial.print(distance); // Sends the distance value
into the Serial Port
Serial.print(".");
}
// 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;
}
```

(D)Hardware system testing – A cable was used for connecting Arduino to develop developing machine. From Arduino IDE helped us to obtain result in serial monitor.

(E) GUI system design and implementation-GUI was built in processing software.

The proccesing software 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);
```

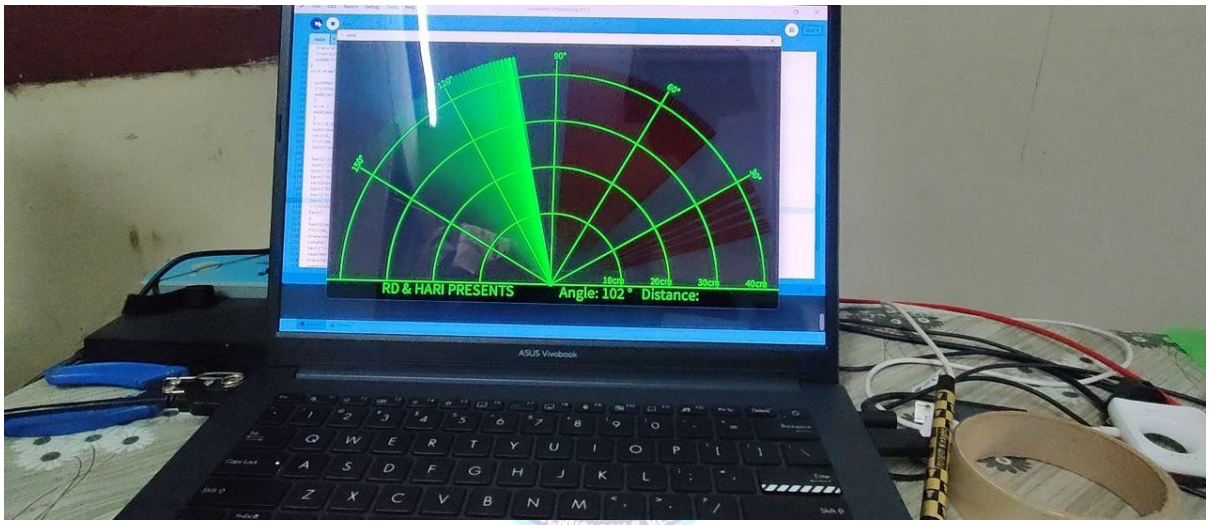
```
  stroke(30,250,60);
  translate(width/2,height-height*0.074); // moves the starting coordinats to new
location
  line(0,0,(height-height*0.12)*cos(radians(iAngle)),-(height-
height*0.12)*sin(radians(iAngle))); // draws the line according to the angle
  popMatrix();
}
void drawText() { // draws the 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("Nurlan Quluzade ", 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) {
```

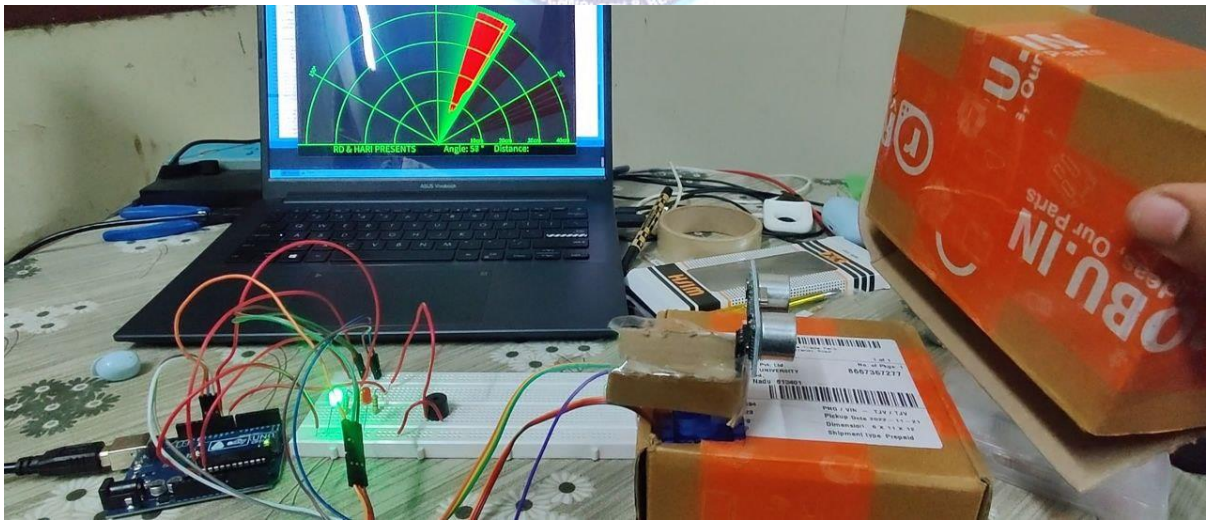
```
  text("        " + iDistance +" cm", width-width*0.225, height-height*0.0277);
  }
  textSize(25);
  fill(98,245,60);
  translate((width-width*0.4994)+width/2*cos(radians(30)),(height-height*0.0907)-width/2*sin(radians(30)));
  rotate(-radians(-60));
  text("30°",0,0);
  resetMatrix();
  translate((width-width*0.503)+width/2*cos(radians(60)),(height-height*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();
}
```

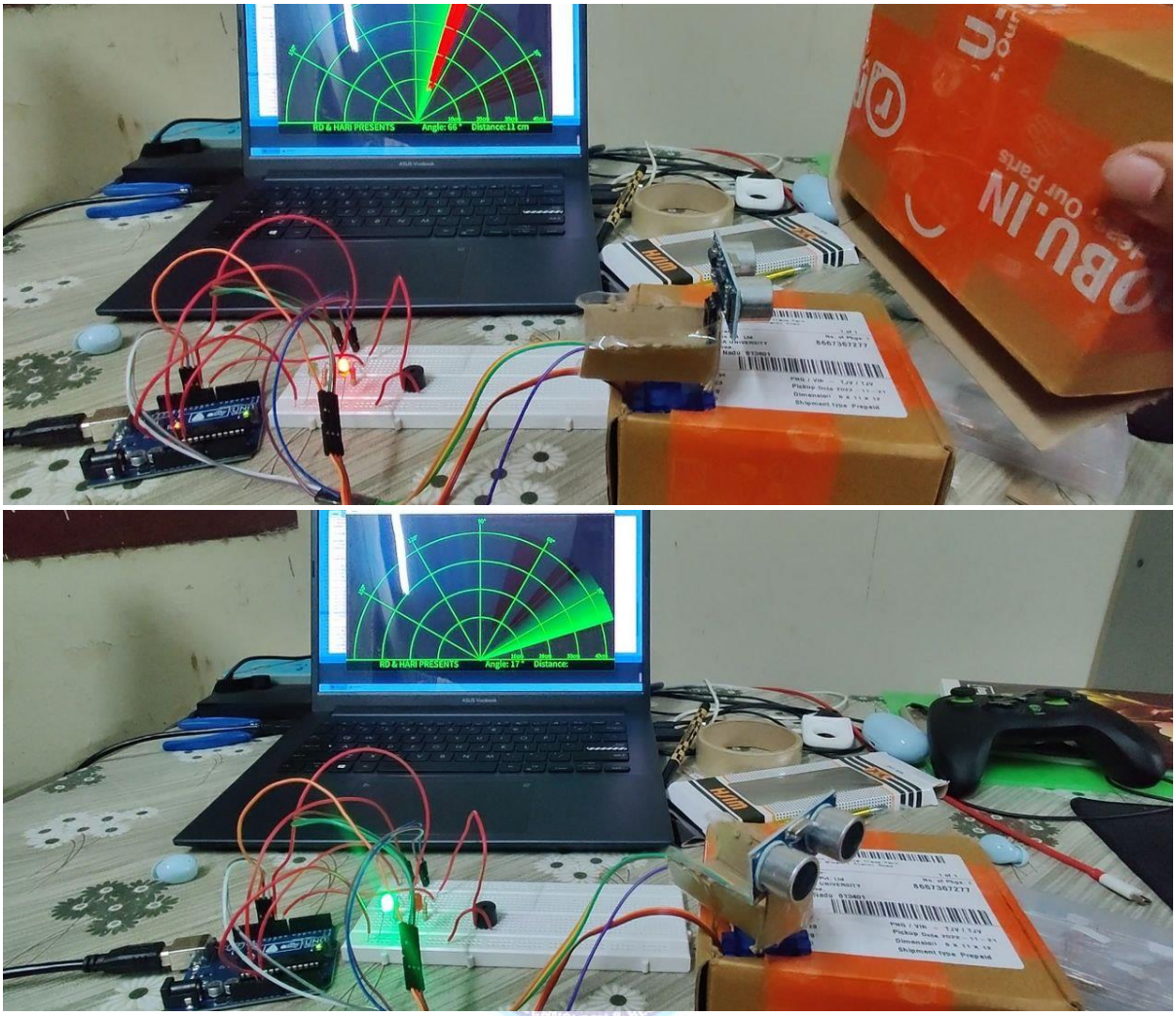
Result images:

WORKING :

The aim of this project is to calculate the distance position and speed of the object placed at some distance from the sensor. Ultrasonic sensor sends the ultrasonic wave in different directions by rotating with help of servo motor. This wave travels in air and gets reflected back after striking some object. This wave is again sensed by the sensor and its characteristics is analysed and output is displayed in screen showing parameters such as distance and position of object. Arduino IDE is used to write code and upload coding in Arduino and helps us to sense position of servo motor and posting it to the serial port along with the distance of the nearest object in its path. The output of sensor is displayed with the help of processing software to give final output in display screen.







Hardware description

Ultrasonic sensor

An ultrasonic sensor works similar as of sonar. It can measure distance of object by sending sound waves. Sound waves are sent at a specific frequency at a specific direction and listen for sound wave to come back. time taken by sound wave to come back helps us to determine distance of object.

Servo motor

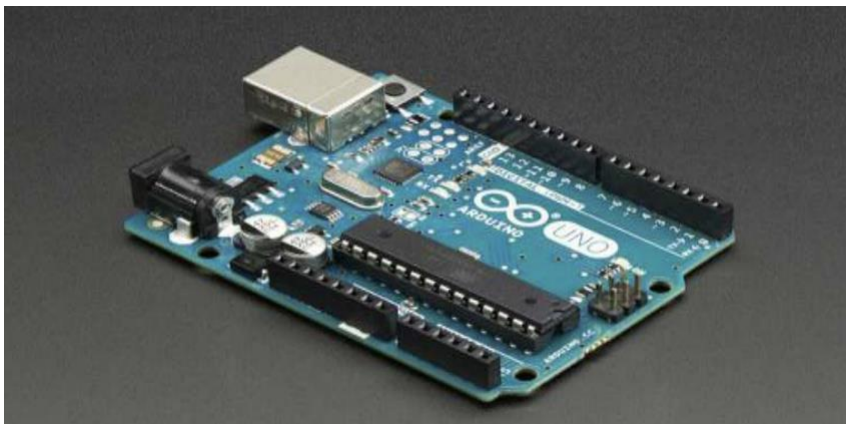
A servomotor is a rotary actuator that allows for precise control of angular position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors. Servomotors are not a different class of motor, on the basis of fundamental operating principle, but uses servomechanism to achieve closed loop control with a generic open loop motor. Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

Arduino

The Arduino is an open-source electronics platform based on

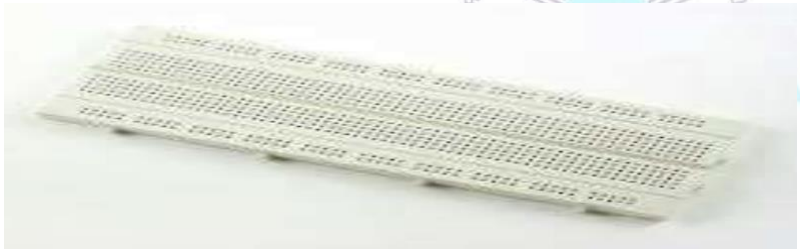
easy-to-use hardware and software. The open-source Arduino software makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X and Linux. The environment is written in java and based on processing and other open-source software. This software can be used with any Arduino board. The Arduino software IDE contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common function. It connects to Arduino

and Genuine hardware to upload programs and communicate with them. Program written using Arduino software are called sketches.



Bread board

Breadboards are one of the most fundamental pieces when learning how to build circuits. In this tutorial, you will learn a little bit about what breadboards are, why they are called breadboards, and how to use one. Once you are done you should have a basic understanding of how breadboards work and be able to build a basic circuit on a breadboard.



ADVANTAGES

1. Radar procurable value is very low
2. Working and maintenance value is low.
3. Distance active resolution is high
4. Radar's jam is troublesome
5. It can work in any place
6. NASA uses radio detection and ranging to map the world and alternative plants
7. Activity gets updated in conclusion

ACKNOWLEDGEMENT

I would like to thank my professor Senthil murugan for his expert advice and encouragement through this difficult project. I would also like

to thank my friends who always encouraged me throughout this period.

Conclusion

In this paper a system radar system was designed with the help of Arduino, servomotor and ultrasonic sensor which can detect the position,

distance of obstacle which comes in its way and converts it into visually representable form.

This system can be used in robotics for object detection and avoidance system or can also be used for intrusion detection for location sizes.

Range of the system depends upon type of ultra-sonic sensor used. We used HC-SR04 sensor which range from 2 to 40 cm.

REFERENCE

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/wcontent/uploads/2012/06/ServoMotor.gif&imgrefurl=http://www.electrosome.com/tag/servomotor/&h=405&w=458&sz=67&tbnid=rcdlwDVt_x0DdM:&tbnh=100&tbnw=113&zoom=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-from-100-yards/

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

[14] http://arduino.cc/en/Tutorial/BarGraph/

[15] http://arduino.cc/en/Tutorial/LiquidCrystal/

[16] http://fritzing.org