

Radar System Using Arduino and Ultrasonic Sensor

PRACHI GUPTA¹, AMAN SINGH², AVIRAL KUMAR³, PROF. V. V. MEHTRE⁴

^{1, 2, 3, 4} *Electrical Engineering Department, Bharati Vidyapeeth (Deemed to Be) University College of Engineering, Pune, Maharashtra India*

Abstract- *The utilization of radio identification and running in better places, for example, army base, business use is finished with the assistance of RADAR SYSTEM which involves electromagnetic waves for discovery of various actual parts, for example, distance, speed, position, range, course, size and so on which can be either fixed or be moving. Utilization of radar framework has been grown extraordinarily exceptionally in field of route. In this exploration we learn about existing route advances and proposed an Arduino based radar framework. It enjoys upper hand over other radar framework as unit diminishes power utilization and interface software engineer to wide reach or Arduino developers and open-source code. The framework comprise a fundamental ultrasonic sensor put upon a servo engine which pivots at a specific point and speed. This ultrasonic sensor is associated with Arduino computerized input yield pins and servo engine additionally associated with advanced input yield pins.*

Indexed Terms- *Arduino, Ultrasonic sensor, Servo motor, Simulation,*

I. INTRODUCTION

We know all that produces sound wave just by presence and impact stream of air around them with their normal recurrence. These frequencies are past hearing scope of people. Wave of recurrence scope of 20000 hz and somewhere around there are called supersonic wave and these waves can be distinguished by a ultrasonic sensor which assists us with getting different information.

A Ultrasonic finder normally has a transducer which convert sound energy into electrical energy and electrical energy into sound energy. They are utilized for estimating object position and direction, crash evasion framework, observation framework and so on.

Ultrasonic innovation give help from issue like straight estimation issue, as it permits client to get non-contact estimations in this manner distance among object and its speed and so forth can me effectively estimated.

Speed of movement of sound wave relies on square foundation of proportion between medium thickness and solidness. Additionally, property of speed of sound can likewise be changed by regular habitat condition like temperature.

So fundamentally, a ultrasonic sensor sends ultrasonic waves which goes in air and gets reflected subsequent to striking any item. By concentrating on the property of reflected wave, we can get information about objects distance, position, and speed and so on.

A handling programming and an Arduino programming is utilized with equipment framework for discovery of items different boundaries.

One of the most well-known use of supersonic sensor is range finding. It is likewise called as sonar which is same as radar in which ultrasonic sound is aimed at a specific course and on the off chance that there is any item in its way it strikes it and gets reflected back and after estimation opportunity taken to arrive back we can decide distance of item. In actuality, this technique is utilized by bats.

II. DESIGN IMPLEMENTATION OF RADAR SYSTEM

The figure displayed underneath shows the advancement life pattern of Radar project which includes different step like plan of various parts, their testing, their execution and execution of whole framework and their testing. These means can be specified 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.

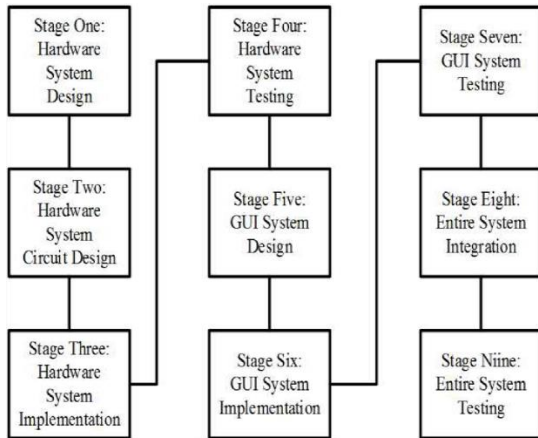


Figure 1. Development life cycle of Radar System.

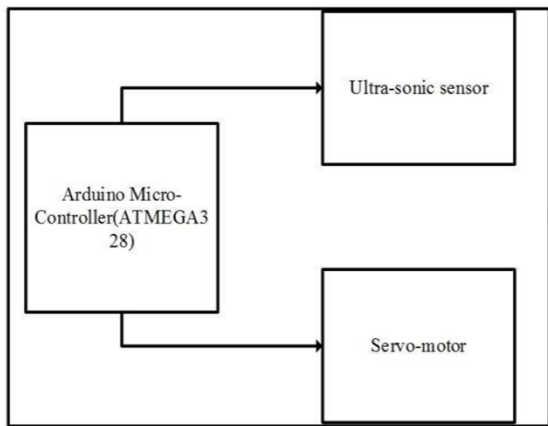


Figure 2. Hardware System Design of Radar System.

(A) Hardware system design for Arduino- Equipment framework comprise of fundamentally 3 parts named as Arduino, servo-engine, and supersonic sensor. Ultrasonic sensor is mounded upon a servo engine which assists it with moving and give it a turning system. Both ultrasonic sensor and servo engine are controlled and fueled by Arduino. As given in above figure 2 we can see both ultrasonic sensor and servo engine is fueled by Arduino.

(B)System circuit design-

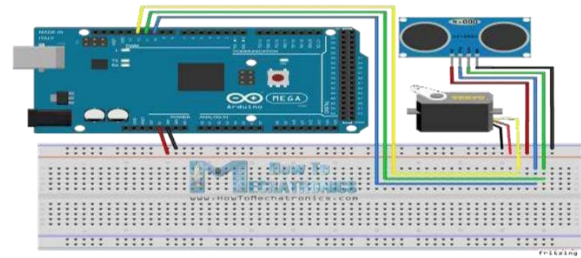


Figure3

Figure shows equipment framework plan which was planned utilizing fritzing climate. It shows the association of various gadgets parts. In the figure setting off pins of ultrasonic sensor is associated with D8 pin of Arduino; control line of servo engine is associated with D6 pin of Arduino and D7 pin of Arduino is associated with reverberation pin. VCC pins of servo engine and ultrasonic sensor is associated with 5V pin of Arduino while ground pin of Arduino is associated with ground pin of both servo engine and supersonic sensor.

(C)System circuit implementation on bread board –

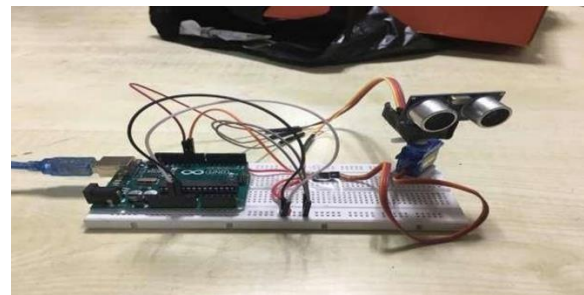


Figure 4. Breadboard of the hardware system implementation.

Above figure 4 shows total execution of equipment framework. It tends to be seen that ultrasonic servomotor is set upon a servo engine and it is set above bread board. Arduino is set in breadboard in opposite side of the breadboard and whole association is made between them. Arduino and servo engine are stick to breadboard to prevent it from stumbling over when servo engine moves.

Arduino IDE was utilized to compose code and transfer it in Arduino. Arduino code peruses position of servo engine and ascertain distance of closest item in the way.

(D) Hardware system testing -

A link was utilized for interfacing Arduino to foster creating machine. From Arduino IDE assisted us with getting bring about chronic screen.

(E) GUI system design and implementation -

GUI was build in JAVA program language and it has 2 classes that are given in below diagram in figure 5.

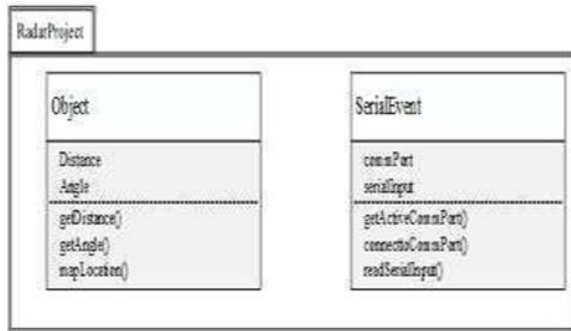


Figure 5. UML Class Diagram for GUI sub-system.

Object class of radar project address object that it experiences like distance, target/reach and point/course of position of article. Distance () strategy (), point () technique, area () technique takes the expected worth like distance, point and makes them on GUI to do recreation. Figure 6 shows line clear from one-course to other and a smear is made is GUI where ultrasonic sensor sense hindrances.

Hardware description

- Ultrasonic sensor

A ultrasonic sensor works comparative as of sonar. It can quantify distance of article by sending sound waves. Sound waves are send at a particular recurrence at a particular course and tune in for sound wave to return. time taken by sound wave to return assists us with deciding distance of item.



- Servo motor

A servomotor is a revolving actuator that takes into consideration exact control of precise position, speed and speed increase. It comprises of a reasonable engine coupled to a sensor for position criticism. It likewise requires a moderately refined regulator, frequently a devoted module planned explicitly for use with servomotors. Servomotors are not an alternate class of engine, based on major working rule, yet utilizes servomechanism to accomplish shut circle control with a conventional open circle engine. Servomotors are utilized in applications like mechanical technology, CNC hardware or robotized fabricating.



- Arduino

The Arduino is an open source gadgets stage in light of simple to utilize equipment and programming. The open source Arduino programming makes it simple to compose code and transfer it to the board. It runs on Windows, Mac OS X and Linux. The climate is written in java and in view of handling and other open source programming. This product can be utilized with any Arduino board. The Arduino programming IDE contains a word processor for composing code, a message region, a message console, a toolbar with buttons for normal capability. It interfaces with Arduino and Genuino equipment t+o transfer programs and speak with them. Program composed utilizing Arduino programming are called draws



III. WORKING

The point of this undertaking is to compute the distance position and speed of the article set at some separation from the sensor.

Ultrasonic sensor sends the ultrasonic wave every which way by turning with assistance of servo engine. This wave goes in air and gets reflected back subsequent to striking some item. This wave is again detected by the sensor and its attributes is broke down and result is shown in screen showing boundaries like distance and position of item.

Arduino IDE is utilized to compose code and transfer coding in Arduino and assists us with detecting position of servo engine and presenting it on the sequential port alongside the distance of the closest article in its way. The result of sensor is shown with the assistance of handling programming to give last result in show screen.

Arduino board conveys a message of +5V to the trig pin of Ultrasonic Sensor HC-SR04 which sets off the sensor.

Then, at that point, it gives rotational activity at the servo engine precisely fitted alongside ultrasonic Sensor HC-SR04 so it can recognize the moving articles and situate inside 180 degrees.

The Arduino sends a HIGH heartbeat width of (10 S) on the TRIGGER pin of the sensor to recover a progression of ultrasonic waves that engender through the air until it contacts a hindrance and returns the other way towards the sensor pin ECHO. The sensor recognizes the width of the beat to ascertain the distance.

The sign on pin ECHO the sensor stays at the HIGH situation during transmission, in this manner estimating the term of the full circle of ultrasound and consequently decide the distance.

The LCD show shows the determined distance and the point of turn. The signal is an extra part, it rings when there is a location (Tone1 and Tone2) alongside LEDs. The two LEDs alongside the signal decide the field where the article is situated (close or far off).

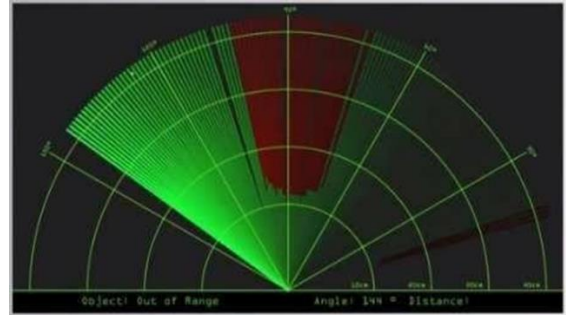


Figure 6. GUI Implementation for the mapping interface.

The Arduino code for the system is highlighted below;

```
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 (1366, 768); //Screen Resolution
surface.setResizable(true); 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(98,245,41); // 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 that
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 coordinats to new location
noFill(); strokeWeight(2); stroke(245,245,245); //
draws the arc lines   arc(0,0,(width-
width*0.0625),(widthwidth*0.0625),PI,TWO_PI);
arc(0,0,(width-
width*0.17),(widthwidth*0.17),PI,TWO_PI);
arc(0,0,(width-
width*0.270),(widthwidth*0.270),PI,TWO_PI);
arc(0,0,(width-
width*0.377),(widthwidth*0.377),PI,TWO_PI);
arc(0,0,(width-
width*0.479),(widthwidth*0.479),PI,TWO_PI);
arc(0,0,(width-
width*0.587),(widthwidth*0.587),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(12
0)));   line(0,0,(-
width/2)*cos(radians(150)),(width/2)*sin(radians(15
```

```
0)));   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(12);
stroke(255,0,0); // red color
```

```
pixsDistance =
iDistance*((heightheight*0.7830)*0.025); //
covers the distance from the sensor from cm to pixels
// limiting the range to 40 cms   if(iDistance<165){
// draws the object according to the angle and the
distance
line(pixsDistance*cos(radians(iAngle)),pixsDistance
*sin(radians(iAngle)),(widthwidth*0.505)*cos(radian
s(iAngle)),-
(widthwidth*0.505)*sin(radians(iAngle)));
}
```

```
popMatrix(); } void drawLine() {   pushMatrix();
strokeWeight(9);   stroke(143,188,143);
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>166) {   noObject = "Out
of Range";
}
else {   noObject = "In Range";
}   fill(50,60,70); noStroke();
rect(0,   height-height*0.0648,   width,   height);
fill(155,245,255);   textSize(15);
```

```
text("20cm",width-
width*0.4755,heightheight*0.0833);
text("40cm",width-
width*0.4260,heightheight*0.0833);
text("55cm",width-
width*0.3754,heightheight*0.0833);
text("75cm",width-
width*0.3270,heightheight*0.0833);
text("95cm",width-width*0.272,height-
height*0.0833);   text("110cm",width-
width*0.227,heightheight*0.0833);
text("130cm",width-
width*0.173,heightheight*0.0833);
text("150cm",width-
width*0.123,heightheight*0.0833);
```

```

text("165cm",width-
width*0.0708,height*0.0833);

textSize(30); text("Object: " + noObject, width-
width*0.875, height*0.0245); text("Angle: " +
iAngle + " °", width-width*0.58,
height*0.0245); text("Distance: ",
width-width*0.26,
height*0.0245); if(iDistance<166) {
text("... " + iDistance + "cm", width-width*0.225,
height-height*0.0245);
}
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();
}
    
```

IV. 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

V. APPLICATIONS

1. Through the wall sensors: It registers movement inside shut spaces.

2. Meteorological Applications microwave radars used to give data about dispersion, type and force of precipitation in climate.
3. UAV Navigations: Unmanned ethereal vehicles or robots usually involved ultrasonic sensors for checking any articles in the UAV's way and distance from the beginning
4. Military Application
5. Air Traffic Control
6. Aircraft Safety
7. Navigation

ACKNOWLEDGEMENT

I might want to thank my teacher V.V. MEHTRE for his master counsel and consolation through this troublesome task. I might likewise want to thank my companions who generally supported me all through this period.

CONCLUSION

In this paper a framework radar framework was planned with the assistance of Arduino, servomotor and ultrasonic sensor which can identify the position, distance of deterrent which comes in its direction and converts it into outwardly representable structure.

This framework can be utilized in advanced mechanics for object identification and aversion framework or can likewise be utilized for interruption recognition for area sizes. Scope of the framework relies on sort of supersonic sensor utilized. We utilized HC-SR04 sensor which range from 2 to 40 cm.

REFERENCES

- [1] G. Bhor, P. Bhandari, R. Ghodekar and S. Deshmukh, "Mini Radar," *International Journal of Technical Research and Applications*, pp. 68-71, 2016.
- [2] D. B. Kadam, Y. B. Patil, K. V. Chougale and. S. S. Perdeshi, "Arduino Based Moving Radar System," *International Journal of Innovative Studies in Sciences and Engineering Technology (IJISSET)*, vol. 3, no. 4, pp. 23-27,2017.
- [3] T. P. Rajan, K. K. Jithin, K. S. Hareesh, C. A. Habeeburahman and. A. Jithin, "Range Detection based on Ultrasonic Principle,"

- International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, no. 2, pp. 7638-7643, 2014.
- [4] P. S. Abhay, S. K. Akhilesh, P. Amrit and Kriti, "A Review on Ultrasonic Radar Sensor for Security system," *Journal of Emerging Technologies and Innovative Research (JETIR)*, pp.137-140, 2016.
- [5] P. P. Arun, M. A. Sudhakar, P. MeghaSunil and S. S. Balaji, "Ultrasonic Distance Meter," *SVERIAN Scientific*, pp. 1-4, 2015. [6] O. V. Amondi, "Collision Avoidance System," The University Of Nairobi, 2019.
- [6] Shamsul A., Tajrian M., "Design of an Ultrasonic Distance Meter", *International Journal of Scientific & Engineering Research*, pp. 1- 10, March 2020.
- [7] U. Papa, G. D. Core, "Design of sonar sensor model for safe landing of an UAV", *Proc. IEEE Metrol. Aerosp.*, pp. 346-350, Jun. 2015.
- [8] Abbay P., Akhilesh S., Amrit P., and Prof Kriti, "A Review on Ultrasonic Radar Sensor for Security system", *Journal of Emerging Technologies and Innovative Research (JETIR)*, April 2016.
- [9] Babu Varghese, "Collision Avoidance System in Heavy Traffic & Blind spot Assist Using Ultrasonic Sensor", *International Journal of Computer Science and Engineering Communications-IJCSEC*. Vol. 2, Issue 1, February, 2014 ISSN: 2347-8586.
- [10] S. Bharambe, R. Thakker, H. Patil, K. M. Bhurchandi, "Substitute eyes for blind with navigator using android", *Proc. Texas Instrum. India Edu. Conf. (TIIEC)*, pp. 38-43, Apr. 2018.
- [11] D. Sunehra, A. Bano, S. Yandrathi, "Remote monitoring and control of a mobile robot system with obstacle avoidance capability", *Proc. Int. Conf Adv. Comput. Commun. Inf. (ICACCI)*, pp. 1803-1809, Aug. 2015.
- [12] <http://www.instructables.com/id/ATMega328-using-Arduino/>
- [13] Hauptmann, Peter, Niels Hoppe, and Alf Püttmer. "Application of inaudible sensors within the method trade." *activity Science and Technology* thirteen.8 (2012): R73.
- [14] Idris, M. Y. I., et al. "Parking system utilizing wireless detector network and inaudible detector." *data Technology Journal* eight.2 (2019): 138-146.
- [11] <http://www.radomes.org/museum/photos/equip/ANSPS17.jpg>
- [16] <http://www.wired.com/dangerroom/2011/07/suicidebombers-from-100-yards/>
- [17] <http://upload.wikimedia.org/wikipedia/commons/Radaraccumulationseng.png>
- [15] <http://arduino.cc/en/Tutorial/BarGraph>.