**Arduino RADAR Model using Ultrasonic Sensor for Detection & Ranging**

## ****Introduction:****

In this project we have designed Arduino RADAR Model using Ultrasonic Sensor for Detection & Ranging. RADAR is an object detection system that uses radio waves to identify the range, altitude, direction and speed of the objects. The radar antenna transmits radio wave pulses that bounce off any object in their path. The object returns a portion of the wave received by the receiver which is in line of sight with the transmitter.

This Arduino RADAR project aims to achieve a radar system prototype based on an Arduino board, capable of  
detecting stationary and moving objects.

**THE TEAM:**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Mentor** | **Qualification** |
| **1.** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Student name** | **Qualification** | **Other details** |
| **1.** |  |  |  |
| **2.** |  |  |  |
| **3.** |  |  |  |
| **4.** |  |  |  |

**Discussion :**

**Participation in entrepreneurship:**

**THE SUPPORT SYSTEM:**

**The host institution (peri institute of technology) is capable of supporting the student needs to develop the project.**

**The institution consist of hardware and software equipment that are helpful to test the project and develop the flaws.**

**THE FINANCIAL REQUIREMENT:**

For designing Arduino RADAR Model using Ultrasonic Sensor, we need following components.

1. Arduino UNO Board (Rs. 500)

2. Servo Motor SG90 (Rs. 120)

3. Ultrasonic Sensor HC-SR04 (Rs. 210)

4. LED - 2 no’s (Rs.10)

5. Buzzer (Rs.30)

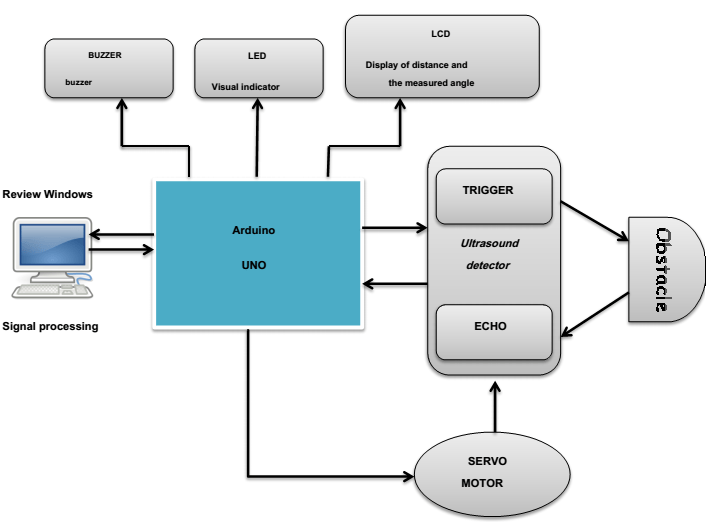
6. 16 x 2 LCD Display (Rs.150)

7. Bread Board (Rs. 90)

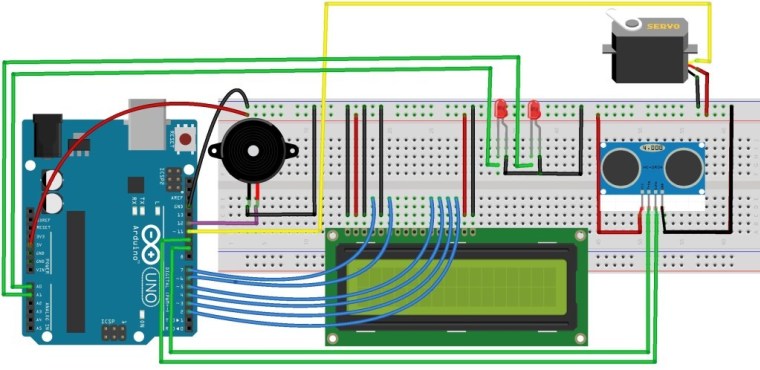
8. Connecting wires (Rs.30)

## ****Arduino RADAR Model using Ultrasonic Sensor:****

### ****Block Diagram:****



### ****Circuit Diagram:****



## ****Program/Source Code:****

#include <Servo.h>

#include <LiquidCrystal.h>

Servo myservo;

LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // Creates an LCD object. Parameters: (rs, enable, d4, d5, d6, d7)

int pos = 0;

const int trigPin = 9;

const int echoPin = 10;

const int moteur = 11;

const int buzzer = 12;

const int ledPin1 = 14;

const int ledPin2 = 15;

float distanceCm, DistanceSec,duration;

void setup() {

myservo.attach(moteur);

lcd.begin(16,2);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(buzzer, OUTPUT);

pinMode(ledPin1, OUTPUT);

pinMode(ledPin2, OUTPUT);

DistanceSec=20;

}

void loop() {

for (pos = 0; pos <= 180; pos += 1) {

// in steps of 1 degree

myservo.write(pos);

position (pos)

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distanceCm= duration\*0.034/2;

if (distanceCm <= DistanceSec)

{

if(distanceCm <= DistanceSec/2)

{

tone(buzzer, 10); // Send 1KHz sound signal...

digitalWrite(ledPin1, LOW);

digitalWrite(ledPin2, HIGH);

delay(700);

noTone(buzzer); // Stop sound...

lcd.setCursor(0,0);

lcd.print("Distance: ");

lcd.print(distanceCm);

lcd.print(" cm ");

delay(10);

lcd.setCursor(0,1);

lcd.print("Angle : ");

lcd.print(pos);

lcd.print(" deg ");

delay(2000);

}

else

{

digitalWrite(buzzer, HIGH);

digitalWrite(ledPin2, LOW);

digitalWrite(ledPin1, HIGH);

delay(100);

digitalWrite(buzzer, LOW);

lcd.setCursor(0,0);

lcd.print("Distance: ");

lcd.print(distanceCm);

lcd.print(" cm ");

delay(10);

lcd.setCursor(0,1);

lcd.print("Angle : ");

lcd.print(pos);

lcd.print(" deg ");

delay(2000);

}

}

else{

digitalWrite(buzzer, LOW);

digitalWrite(ledPin1, LOW);

digitalWrite(ledPin2, LOW);

}

lcd.setCursor(0,0);

lcd.print("Distance: ");

lcd.print(distanceCm);

lcd.print(" cm ");

delay(10);

lcd.setCursor(0,1);

lcd.print("Angle : ");

lcd.print(pos);

lcd.print(" deg ");

delay(80);

}

for (pos = 180; pos >= 0; pos -= 1) { //

myservo.write(pos); //

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distanceCm= duration\*0.034/2;

if (distanceCm <= DistanceSec){

if(distanceCm <= DistanceSec/2)

{

tone(buzzer, 10); // Send 1KHz sound signal...

digitalWrite(ledPin1, LOW);

digitalWrite(ledPin2, HIGH);

delay(700);

noTone(buzzer); // Stop sound...

lcd.setCursor(0,0);

lcd.print("Distance: ");

lcd.print(distanceCm);

lcd.print(" cm ");

delay(10);

lcd.setCursor(0,1);

lcd.print("Angle : ");

lcd.print(pos);

lcd.print(" deg ");

delay(2000);

}

else

{

digitalWrite(buzzer, HIGH);

digitalWrite(ledPin2, LOW);

digitalWrite(ledPin1, HIGH);

delay(100);

digitalWrite(buzzer, LOW);

lcd.setCursor(0,0);

lcd.print("Distance: ");

lcd.print(distanceCm);

lcd.print(" cm ");

delay(10);

lcd.setCursor(0,1);

lcd.print("Angle : ");

lcd.print(pos);

lcd.print(" deg ");

delay(2000);

}

}

else{

digitalWrite(buzzer, LOW);

digitalWrite(ledPin1, LOW);

digitalWrite(ledPin2, LOW);

}

lcd.setCursor(0,0); //

lcd.print("Distance: "); //

lcd.print(distanceCm); //

lcd.print(" cm ");

delay(10);

lcd.setCursor(0,1);

lcd.print("Angle : ");

lcd.print(pos);

lcd.print(" deg ");

delay(80);

}

}

**Project Description:**

This project is helpful in measuring the distance from one ship to another ship and also even in aircrafts so that accidents can be prevented and controlled.

This project can be done is both hardware as well as software programs.