Robotics & Automation

ULTRASOUND SCANNER

USER MANUAL

A fun project for students with robotic interests.

CERTIFIED ISO 9001:2015 COMPANY
ULTRASONIC SCANNER KIT
Build a classroom, build a future
CE
IP65
Features :-
1.Ultrasonic Sensor
2.Servo Motor
3.Microcontroller
SVR INFO TECH
MAKE IN INDIA
Skill India
कौशल भारत - कुशल भारत
Digital India
Power To Empower
Address: 301-302, 3rd Floor, Amber Plaza, Sinhgad Institute Rd,
Ambegaon BK,Pune, Maharashtra 411046
support@svrinfotech.net +91 9923444362 / 020 29840013

www.svrinfotech.net

©Copyrights of SVR INFOTECH

Table of Contents:

Introduction:

In Education Module, SVR InfoTech offers ULTRASONIC SCANNER. Environment scanners are widely used for various applications like in airplanes, ships, self-driving cars, autonomous robots, hurricane monitoring, etc. The purpose of the environment scanner is to see the surrounding environment and report it. For this purpose, various types of sensing are utilized which include light, ultrasonic sound, infrared light, radio waves, etc.
The distance of obstacle or discontinuities in metals is related to velocity of sound waves in a medium through which waves are passed and the time taken for echo reception. Hence the ultrasonic detection can be used for finding the distances between particles, for detecting the discontinuities in metals and for indicating the liquid level. The project is designed and develop distance measurement system using ultrasonic waves interfaced with Arduino.

General Precautions:

Caution: To avoid injury, damage to the robot or equipment, please follow the provided guidelines.

1. Keep away from pets and animals of any kind, animals may behave erratically in the presence of the robot.
2. If the robot is operating abnormally, there is an unusual sound, smell or smoke is detected:
    a. Turn off the robot immediately
3. Always follow the installation and service instructions closely. Keep manuals for future reference.
4. This guide does not cover all possible safety issues or conditions. Always use common sense and good judgment.

©Copyrights of SVR INFOTECH

5. Please take care of this unit and its accessories, keep them clean.
6. Please do not break, throw or trample the robot.
7. Avoid installation in extremely hot, rainy or water splashing, or being placed in high temperature or moist environment.

Components:

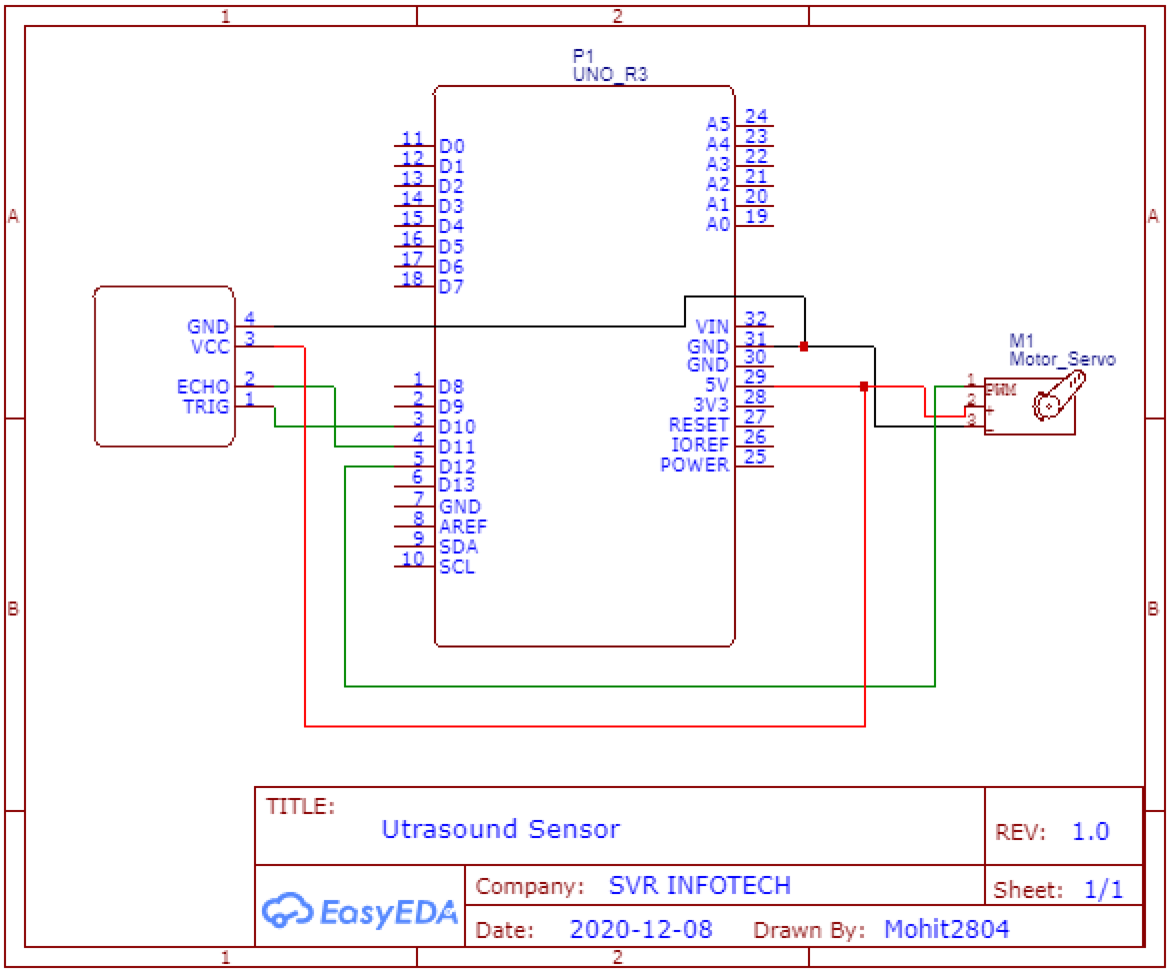
MECHANICAL COMPONENTS		
SR. NO.	PART NAME	QUANTITY
1.	Plastic Enclosure box (IP65)	1
2.	Upper Acrylic plate	1
3.	Bottom Acrylic Plate	1
4.	M3*45 (Spacers)	4
5.	M3*6 (Nut)	1

ELECTRICAL COMPONENTS		
SR NO.	PART NAME	QUANTITY
1	Servo Motor SG90	1
2	Arduino Uno	1
3	Ultrasound Sensor	1
4	Connecting Wires	

PREREQUISITES		
SR NO.	PART NAME	QUANTITY
1	Personal Computer with Arduino IDE	1
2	USB B to USB A cable	1

©Copyrights of SVR INFOTECH

Circuit Diagram



©Copyrights of SVR INFOTECH

Connections:

Arduino and Ultrasound Sensor		
SR NO.	Ultrasound Sensor Pins	Arduino Pins
1	TRIG	D10
2	ECHO	D11
3	GND	GND
4	VCC	5V

For Servo Motor		
SR NO.	Motor Pins	Arduino Pins
1	SIGNAL	D12
2	GND	GND
3	VCC	5V

Code Explanation:

The explanation of the code is presented in the program as the comment.

©Copyrights of SVR INFOTECH

Code:

Arduino Code:

```
// Includes the Servo library

#include <Servo.h>.

// Defines Tirg and Echo pins of the Ultrasonic Sensor

const int trigPin = 10;
const int echoPin = 11;
// 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(12); // 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();// Calls a function for calculating the
distance measured by the Ultrasonic sensor for each degree
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(int i=165;i>15;i--){
myServo.write(i);
delay(30);
distance = calculateDistance();

Serial.print(i);

Serial.print(",");
Serial.print(distance);
```

©Copyrights of SVR INFOTECH

```
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:

```
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,"COM4", 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.
orcFont = loadFont("OCRAExtended-30.vlw");
}
void draw() {
fill(98,245,31);
textFont(orcFont);
```

©Copyrights of SVR INFOTECH

```
// 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();
```

©Copyrights of SVR INFOTECH

```
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("Object: " + noObject, width-width*0.875, height-height*0.0277);
text("Angle: " + iAngle +" °", width-width*0.48, height-height*0.0277);
```

©Copyrights of SVR INFOTECH

```
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();
}
```

Note:

size (1920, 1080); >> Change the values to adjust to your screen resolution

myPort = new Serial(this,"COM4", 9600); >> Change COM port value as per the system

©Copyrights of SVR INFOTECH

Warranty Terms and Conditions:

Warranty Period: - 1 year for mechanical components & 90 Days for electronic components from the date of delivery.

What is covered: - Any Technical defect, malfunctioning.

What is not covered: - Physical Damage, Water Damage, Wear & Tear.

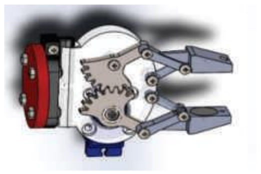
What will we do: - Repair or Replacement whichever will be applicable.

Other terms and conditions: -

- Warranty will be void if warranty seals are broken.
- The warranty will be void if the product is mishandled.
- Warranty will be void on installation to wrong voltage, overload, and wrong application.
- Use other than in accordance with handling instructions.
- The warranty will not extend after replacement.

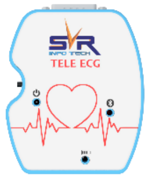
OUR OTHER PRODUCTS

Hast Articulated Robot


Miniature Geared Gripper

Miniature Cam Guided Gripper

Miniature Curvilinear Gripper


Tele ECG

SCARA Robot

Flexible Robotic System

Robotic leech

Self-Balancing Robot

Ultrasonic Scanner

Maze Robot

Digital Dice kit

Multi Gripper Robot

Conveyor Belt object counter

Line Follower Robot

Parallel Manipulator Gripper stem

Amber Plaza, 3rd Floor, Near Bank of Maharashtra, Sinhgad College Rd, Ambegaon Budruk, Pune-411046.

Website: www.svrinfotech.net E-mail: admin@svrinfotech.net / support@svrinfotech.net

+91 9923444362, +91 9975507755 , 020 2984 0013

©Copyrights of SVR INFOTECH