CENTRO PAULA SOUZA

ETEC PROF. MARCOS UCHÔAS DOS SANTOS PENCHEL

Técnico em Eletrônica Integrado ao Ensino Médio – Novotec Integrado

Sensor Ultrassônico Para Detectar os Objetos

Ultrasonic Sensor To Detect The Objects

Amanda Ribeiro Villela[1]

José Guilherme Fonseca Lopes[2]

Orientadores:

Prof. Felipe Lopes Cavalcanti[3]

Prof. Marcílio Marques Monteiro de Azevedo[4]

Resumo: O sensor ultrassônico para detectar objetos e uma imitação de radar, porem menor. Um radar no qual iremos utilizar o servo motor para girar em um ângulo de 180°, um sensor ultrassônico para detectar os objetos e o arduino para controlar os dois de acordo com os códigos que vamos colocar nele.

Palavras-chave: Sensor; Ângulo; Arduino.

Abstract: The ultrasonic sensor for detecting objects is an imitation of radar, but smaller. A radar in which we will use the servo motor to rotate at an angle of 180°, an ultrasonic sensor to detect the objects and the Arduino to control both according to the codes that we are going to put on it.

Keywords: Sensor; Angle; Arduino.

[1] Aluna do 3º ano do Técnico em Eletrônica Integrado ao Ensino Médio – NOVOTEC Integrado
[2] Aluno do 3º ano do Técnico em Eletrônica Integrado ao Ensino Médio – NOVOTEC Integrado
[3] Professor do Ensino Médio e Técnico da ETEC Prof. Marcos Uchôas dos Santos Penchel.
[4] Professor do Ensino Médio e Técnico da ETEC Prof. Marcos Uchôas dos Santos Penchel.

INTRODUÇÃO

O projeto do detector de radar Arduino é popular hoje em dia e agora vamos entender o que realmente é. É como um mini-radar que reconhece o objeto no caminho e cria uma área afetada vermelha perto do objeto. Então, isso é chamado de radar e funciona na frequência ultrassônica. Consiste em sensor ultrassônico montado sobre o servo motor e conectado a um software que mostra o resultado na tela do computador. A interface deste software é praticamente a mesma da interface do Radar.

Objetivo

O projeto sensor para identificar objetos vem com o intuito de futuramente servir para colocar nas propriedades das pessoas que querem melhorar a segurança da sua empresa, condomínio, casa, fazenda, hotel e lojas, assim atuando junto com as câmeras de segurança, alarmes e cadeados, funcionaria como um sensor para identificar as pessoas, utilizando como uma proteção quando não estiver ninguém no local.

Cronograma

Meses	Atividades	Entregas
Fevereiro	Definição dos grupos.	20/02/2023
Março	Definição do projeto.	17/03/2023
Abril	Procura e compra dos materiais.	15/04/2023
Maio	Montagem do projeto.	10/05/2023
Junho	Apresentação Pré TCC com Power point, começando artigo.	29/06/2023
Agosto	Artigo pronto.	21/08/2023
Setembro	Toques finais no projeto.	20/09/2023
Novembro	Revisão de tudo e entrega de TCC.	13/11/2023
Dezembro	TCC concluído.	04/12/2023

Desenvolvimento

Sensor ultrassônico gira com o servo motor e transmite as ondas ultrassônicas durante esse tempo. Precisando o tempo todo de uma interface gráfica feita no software de simulação. Se tiver algum objeto sob o alcance do sensor ultrassônico, ele começará a detectar o objeto. Nesse momento, a interface do gráfico dentro do software torna-se vermelha na área do objeto. O sensor ultrassônico funciona como um detector de objetos neste projeto**.** Sensor ultrassônico funciona em software e faz a reação de acordo com as ondas recebidas.

Sensores ultrassônicos têm dois terminais, um é um transmissor e outro é o receptor. O terminal do transmissor é conhecido como Trigger e o terminal do receptor é conhecido como echo. O Arduino continuamente dá um comando para o servo motor girar. E o transmissor transmite o sinal paralelamente da mesma forma que o software também faz o gráfico. O sensor ultrassônico deu um sinal diferente para o Arduino se alguma coisa aparecer no caminho.

Então, o Arduino notifica o software da região afetada, o projeto depende do funcionamento do sensor ultrassônico. Radar usando Arduino, sensor ultrassônico e conteúdo do servo motor sem outros componentes principais divide-se em seções e subseções, que variam em função da abordagem do tema e do método. É a parte principal do artigo, e inclui a metodologia, fundamentação teórica, dados obtidos por meio de pesquisas, resultados alcançados e discussão.

Componentes

Sensor ultrassônico HC-SR04

O sensor ultrassônico é um dispositivo que emite ondas sonoras de alta frequência.

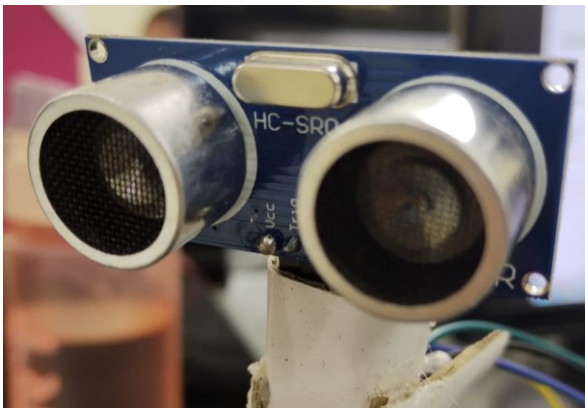

FIGURA 1: SENSOR ULTRASSÔNICO HC-SR04

Servo motos Sg9

Os servos motores são usados em várias aplicações quando se deseja movimentar algo de forma precisa e controlada.

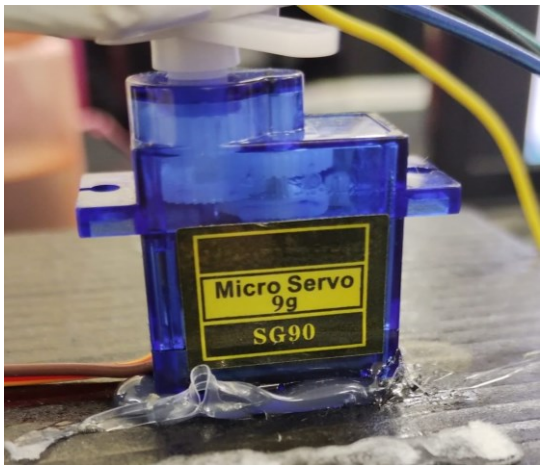

FIGURA 2: SERVO MOTOS SG9

Placa Arduino

O Arduino serve para facilitar o aprendizado de programação, ensinando as pessoas a desenvolverem projetos de eletrônica e de robótica.

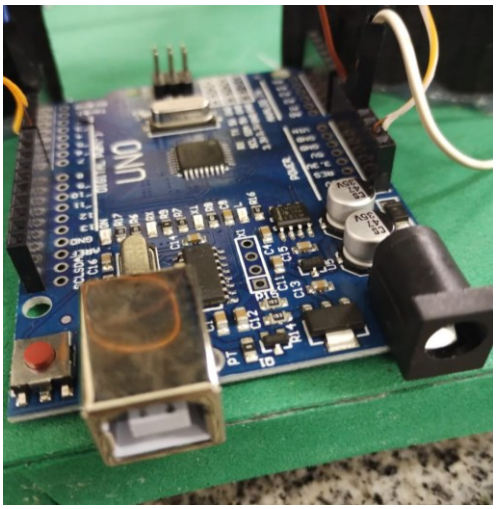

FIGURA 3: PLACA ARDUINO

Jumpers Macho e fêmea

Fios para conectar os componentes.

FIGURA 4: JUMPERS MACHO E FÊMEA

Circuito

O que é um circuito

Circuito elétrico é uma ligação de dispositivos, como geradores, resistores, receptores, capacitores, indutores, etc., feita por meio de um fio condutor, que permite a passagem de cargas elétricas pelos elementos do circuito.

Circuito do projeto

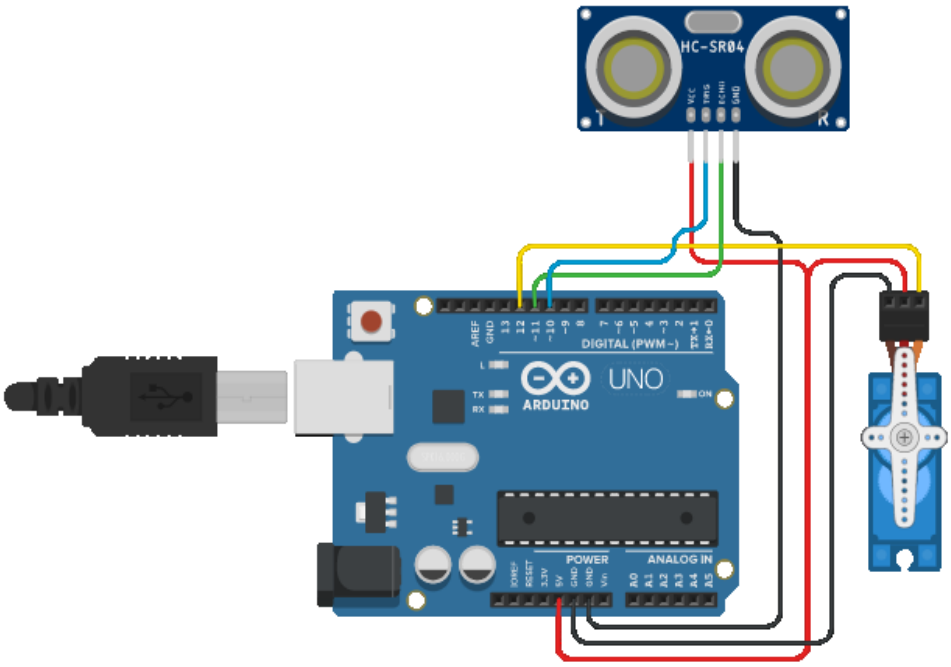

FIGURA 5: CIRCUITO DO PROJETO

- Porta do servo motor (PWR) no pino 12 do Arduino.
- Porta do sensor ultrassónico (TRIG) no pino 10 do Arduino.
- Porta do sensor ultrassónico (ECHO) no pino 11 do Arduino.
- Porta do positivo Arduino (5V) conectado no sensor ultrassônico (VCC) e servo motor (VCC).
- Porta do negativo Arduino (GND) conectado no sensor ultrassônico (GND) e servo motor (GND).

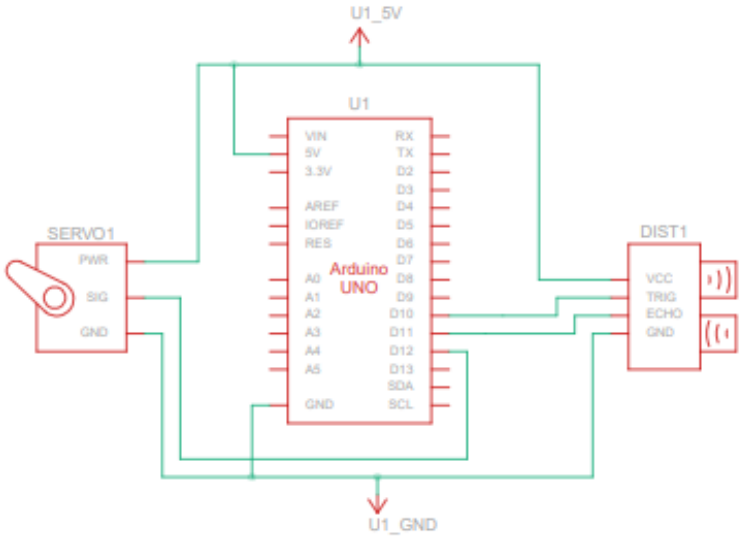

FIGURA 6: DIAGRAMA DO CIRCUITO

Códigos do Projeto

Programação do Arduino

Linguagem de programação c++

// Inclui a biblioteca servo

#include <Servo.h>.

// Definir Tirg and Echo pinos do sensor Ultrasonic

const int trigPin = 10;

const int echoPin = 11;

// Variavel para a duração e a distancia

long duração;

int distancia;

Servo myServo; // Cria um objeto servo para controlar o servo motor

void setup() {

pinMode(trigPin, OUTPUT); // define trigPin como uma saida

pinMode(echoPin, INPUT); // define echoPin como uma entrada

Serial.begin(9600);

myServo.attach(12); // Define em qual pino o servo ta conectado

}

```
void loop() {
// gira o servo de  15 a 165 graus
for(int i=15;i<=165;i++){
myServo.write(i);
delay(30);
distance = calculateDistancia();// Chama uma função para calcular distancia medido pelo sensor ultrassônico para cada grau
Serial.print(i); // envia o grau atual
Serial.print(","); // envia o caractere da edição ao lado do valor anterior necessário posteriormente no IDE de processamento para indexação
Serial.print(distance); // envia o valor da distancia para o serial
Serial.print("."); // envia o caractere de adição ao lado do valor anterior necessário posteriormente no IDE de processamento para indexação
}
// Repete as linhas anteriores de 165 a 15 graus
for(int i=165;i>15;i--){
myServo.write(i);
delay(30);
distance = calculateDistance();
Serial.print(i);
Serial.print(",");
Serial.print(distance);
Serial.print(".");
}
} // função para calcular a distancia medida pelo sensor ultrassônico

int calculateDistance(){
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// define o trigPin no estado alto por 10 microssegundos
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
```

```
// lê o echoPin, retorna o tempo da viagem da onda Sonora em microssegundos
distance= duration*0.034/2;
return distance;
}
```

Programação do Aplicativo Prossesing

Processing é utilizada para desenvolvimento de artes visuais.

Linguagem de programação java.

```
import processing.serial.*; // biblioteca de importações para comunicação serial
import java.awt.event.KeyEvent; // biblioteca de importações para ler os dados da porta serial
import java.io.IOException;
Serial myPort; // define objeto Serial
// desfaz variáveis
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);
smooth();
myPort = new Serial(this,"COM4", 9600); // inicia comunicação serial
myPort.bufferUntil('.'); // lê os dados para porta serial ate o caractere'.'. Então na verdade e assim: Angulo, distancia.
}
void draw() {
fill(98,245,31);
```

```
// simulando desfoque de movimento e desvanecimento lento da linha em movimento
noStroke();
fill(0,4);
rect(0, 0, width, height-height*0.065);
fill(98,245,31); // cor verde
// chama as funções para desenhar o radar
drawRadar();
drawLine();
drawObject();
drawText();
}
void serialEvent (Serial myPort) { // inicia a leitura de dados da porta serial
// lê os dados da Porta serial ate o caractere'.' e o coloca na variavel string "data".
data = myPort.readStringUntil('.');
data = data.substring(0,data.length()-1);
index1 = data.indexOf(","); // encontra o caractere ',' e o coloca na variavel "index1"
angle= data.substring(0, index1); // leia os dados da posição"0" ate a posição da variavel index1 ou esse e o valor do anguloque a placa Arduino enviou para a porta serial
distance= data.substring(index1+1, data.length()); // leia os dados da posição "index1" ate o final dos dados pr esse e o valor da distancia
// converte a variavel String em variavel Integer
iAngle = int(angulo);
iDistance = int(distancia);
}
void drawRadar() {
pushMatrix();
translate(width/2,height-height*0.074); // move as coordenadas iniciais para um novo local
noFill();
strokeWeight(2);
stroke(98,245,31);
// desenha as linhas do arco
```

```
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);
// desenha as linhas do ângulo
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();
}
```

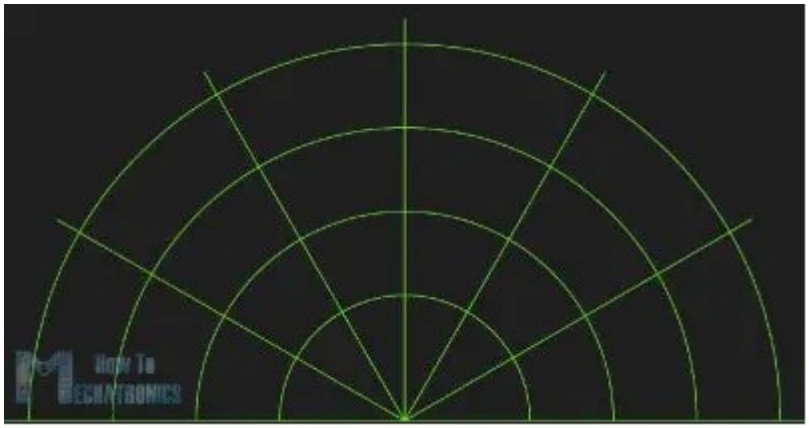

FIGURA 7: FORMANDO LINHA DO RADAR

```
void drawObject() {
pushMatrix();
translate(width/2,height-height*0.074);
 // move as coordenadas iniciais para um novo local
strokeWeight(9);
stroke(255,10,10); // cor vermelha
pixsDistance = iDistance*((height-height*0.1666)*0.025);
 // cobre a distancia do sensor de cm a pixels
// limitando o alcance para 40 cms
```

```
if(iDistance<40){
// desenha o objeto de acordo com o ângulo e a distância
line(pixsDistance*cos(radians(iAngle)),-
pixsDistance*sin(radians(iAngle)),(width-width*0.505)*cos(radians(iAngle)),-
(width-width*0.505)*sin(radians(iAngle)));
}
```

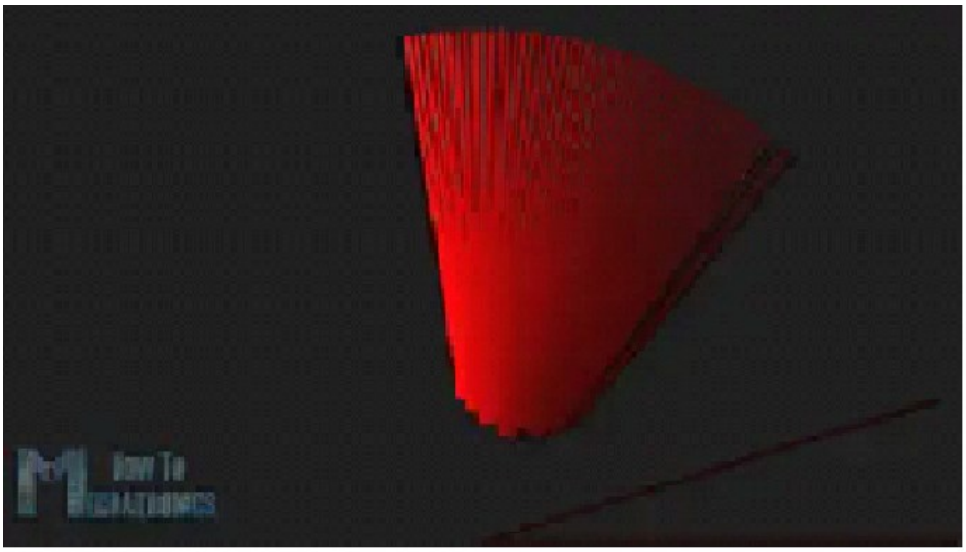

FIGURA 8: LINHA PARA IDENTIFICAR OBJETO

```
void drawLine() {
pushMatrix();
strokeWeight(9);
stroke(30,250,60);
translate(width/2,height-height*0.074);
// move as coordenadas iniciais para um novo local
line(0,0,(height-height*0.12)*cos(radians(iAngle)),-(height-
height*0.12)*sin(radians(iAngle)));
 //desenha a linha de acordo com o ângulo
popMatrix();
}
```

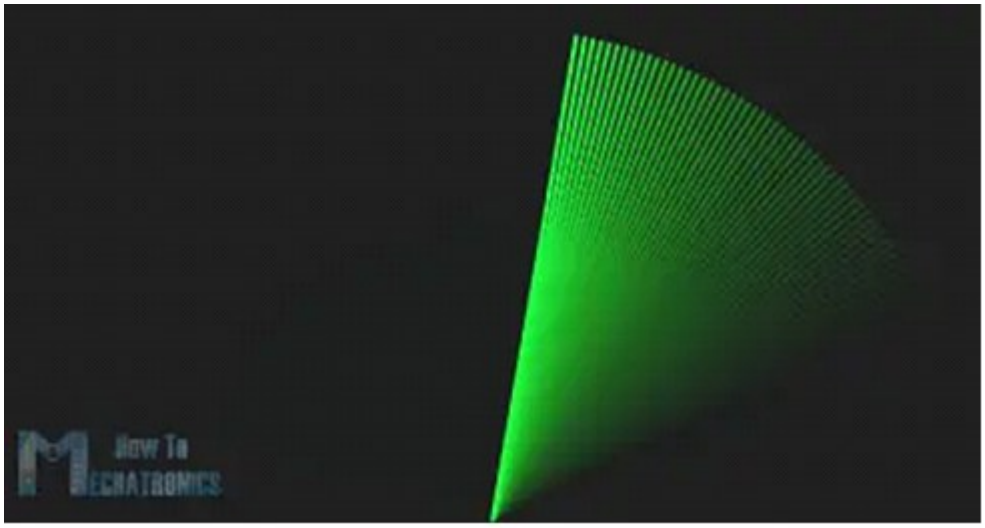

FIGURA 9: LINHA PARA IDENTIFICAR SEM NENHUM OBJETO

```
void drawText() { // desenha os textos na tela
pushMatrix();
if(iDistance>40) {
noObject = “fora do alcance";
}
else {
noObject = “no intervalo";
}
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("Indian Lifehacker ", 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();
}
```

CONSIDERAÇÕES FINAIS

Como base que foi apresentado, o sensor ultrassónico foi projetado para ser utilizado como uma forma de segurança no intuito de trabalhar junto das câmeras e alarmes. Temos como prioridade a facilidade, utilidade e inovação.

Nosso objetivo é evoluir ele conforme o tempo, criando nosso próprio sistemas de monitoramento, alarmes e sensores.

REFERÊNCIAS

https://howtomechatronics.com/projects/arduino-radar-project/
https://www.makerhero.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/

E-mail de contato dos autores:

[1] E-mail de contato: amandavillela23@gmail.com
[2] E-mail de contato: joselopes.cp17@gmail.com

E-mail de contato dos orientadores:

[3] E-mail de contato: felipe.cavalcante12@etec.sp.gov.br
[4] E-mail de contato: marcilio.azevedo@etec.sp.gov.br