Projede kullanılan malzemelerin amacı:

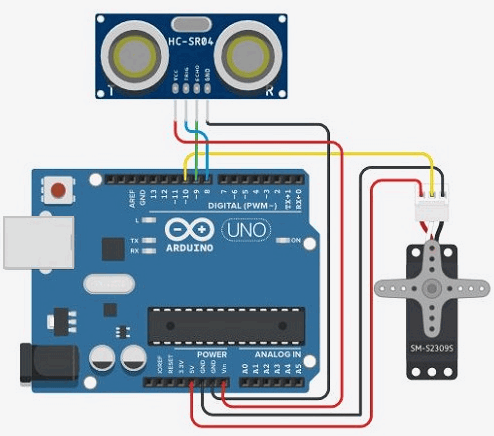
Jumper Dupont Kablo: Veri aktarımı ve güç aktarımı.

Arduino Uno R3:Komut yollama veri veri aktarım köprüsü olarak kulanıır beyi.

SG90 Servo Motor:Ultra sonik sensörün dönmedine etrafı taramasına yardımcı olur

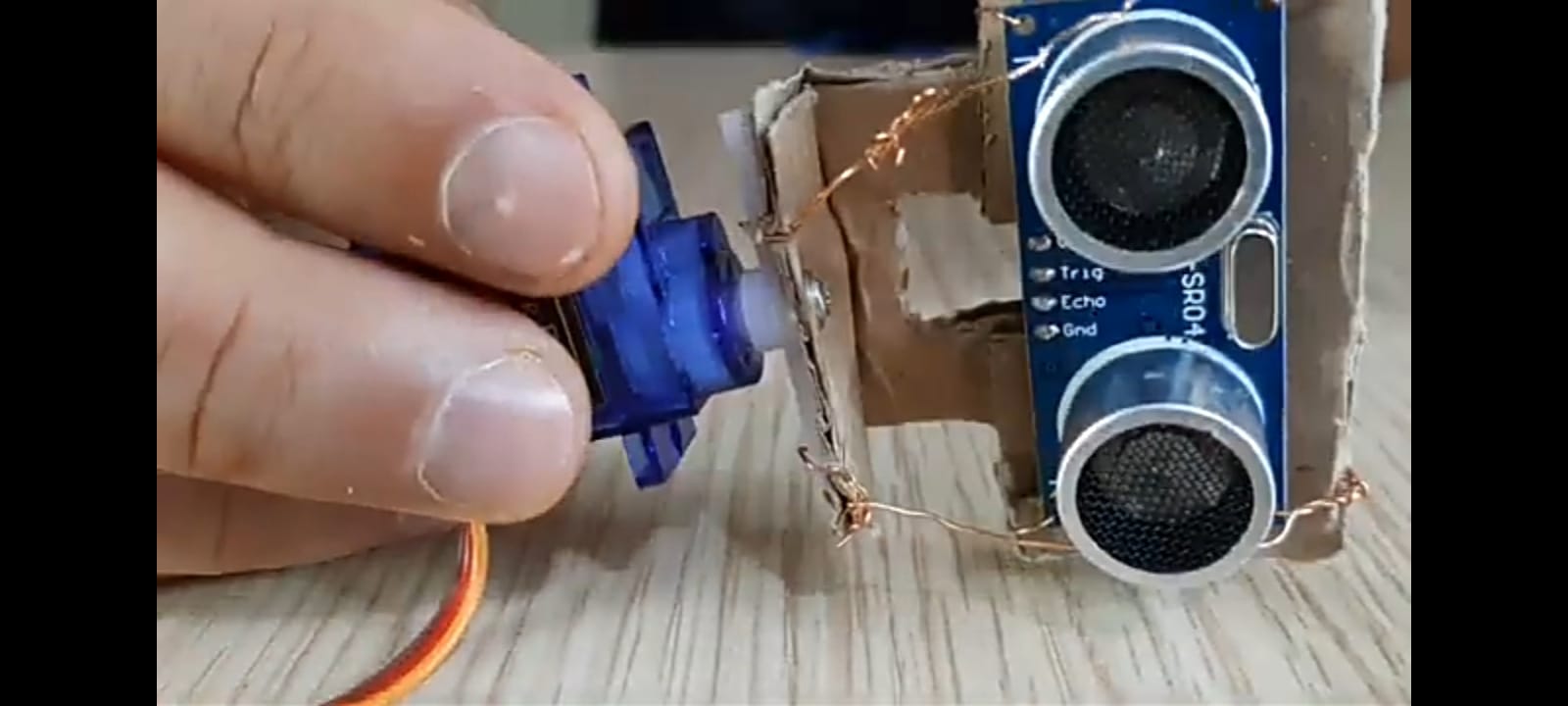
Ultrasonik Sensör HC SR04: etrafa radyo dalgalrı yollayarak cisme çarpıp geri gelen dalgalardan cismin yerini bulur

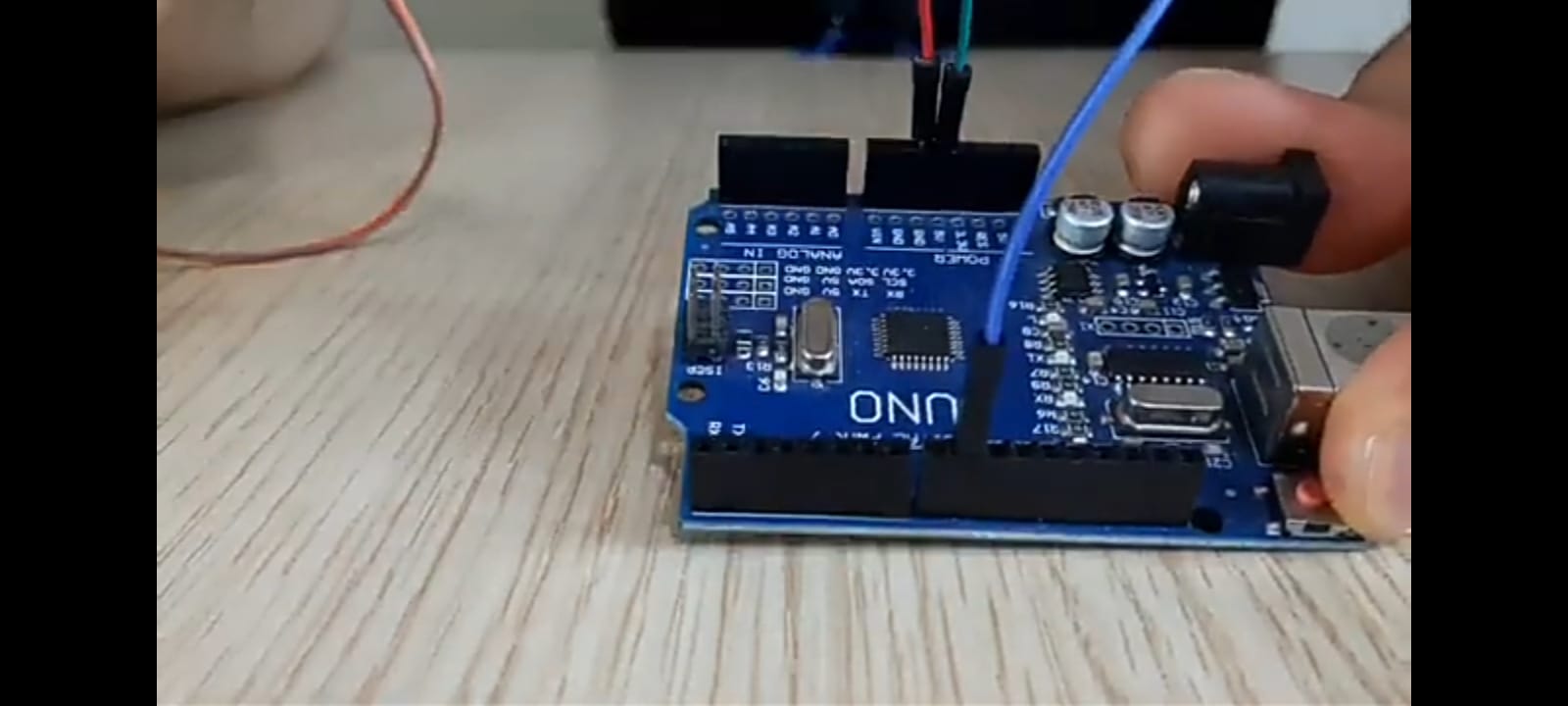
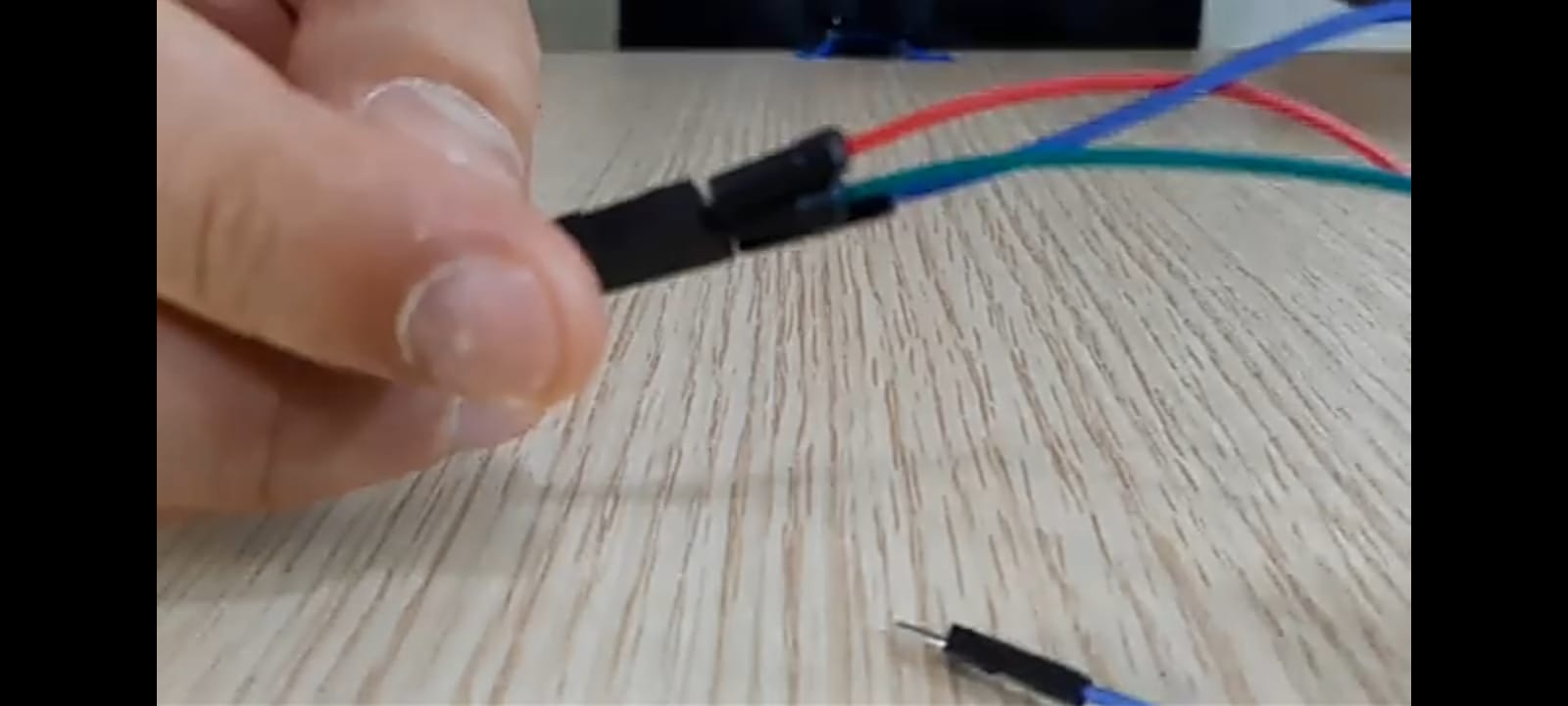
ARDUİNO RADAR DEVRE ŞEMASI



1.Adım

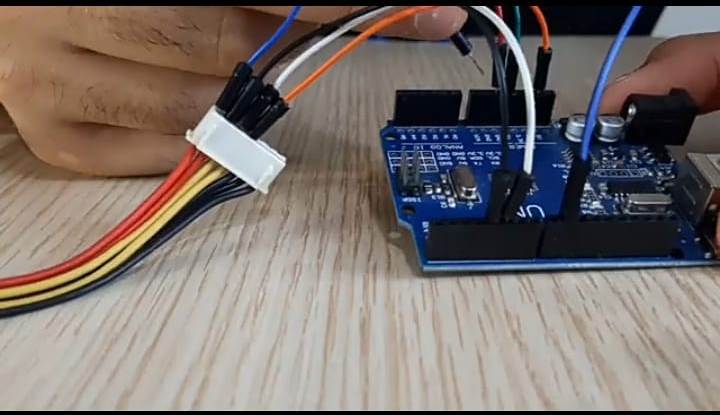
Sensörünüzü , Servomotora takmakiçin karton bir stand yapın.

2.Adım

Servo motra erkek kablolarımızı bağlıyoruz sarı uc sensor (9P),yeşil güç (5V) ye,kırmızı topraklama (GND).

3.Adım

Turuncu vcc (5V) ye takın,beyaz trig pin’i 4.pin’e takın,siyah eco pinini 3.pin’e takın,mavi gnd yani 2.toprak pinin’e takın.



|  |  |
| --- | --- |
|  | |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | |  |

ARDUİNO CODE:

#include <Servo.h>.

const int trigPin = 4;

const int echoPin = 3;

long duration;

int distance;

Servo myServo;

void setup() {

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

Serial.begin(9600);

myServo.attach(9);

}

void loop() {

for(int i=15;i<=165;i++){

myServo.write(i);

delay(30);

distance = calculateDistance();

Serial.print(i);

Serial.print(",");

Serial.print(distance);

Serial.print(".");

}

for(int i=165;i>15;i--){

myServo.write(i);

delay(30);

distance = calculateDistance();

Serial.print(i);

Serial.print(",");

Serial.print(distance);

Serial.print(".");

}

ARDUİNO KOD AÇIKLAMA

#include <Servo.h>

Bu hat, servo motorları kontrol etmek için işlevler sağlayan Servo kitaplığını içerir.

const int trigPin = 4;

const int echoPin = 3;

Bu satırlar iki sabit bildirir: trigPin ve echoPin. Bu pinler, trigPin'in tetikleme sinyalini göndermek için kullanıldığı ve echoPin'in yankı sinyalini almak için kullanıldığı ultrasonik sensöre bağlanır.

Servo myServo;

Bu satır, Servo sınıfının bir myServo nesnesini bildirir. Servo motoru kontrol etmek için kullanılacaktır.

void setup() {

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

Serial.begin(9600);

myServo.attach(9);

}

setup() işlevi başlangıçta bir kez yürütülür. Pinleri ve Seri iletişimi başlatır. pinMode(), trigPin'i çıkış pini olarak ve echoPin'i giriş pini olarak ayarlar. Serial.begin(9600), seri iletişimi 9600 baud hızında başlatır. myServo.attach(9), servo motoru pin 9'a bağlar.

void loop() {

// Code for sweeping the servo from 15 to 165 degrees

for(int i=15; i<=165; i++) {

myServo.write(i);

delay(30);

distance = calculateDistance();

Serial.print(i);

Serial.print(",");

Serial.print(distance);

Serial.print(".");

}

// Code for sweeping the servo 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(".");

}

}

loop() işlevi, setup() işlevinden sonra art arda yürütülür. İki for döngüsü içerir. İlk döngü, servo motoru 1 derecelik artışlarla 15 ila 165 derece arasında kaydırır. Döngü içinde, myServo.write(i) kullanarak servo açısını ayarlar, delay(30) kullanarak 30 milisaniye bekler, hesaplaMesafe() işlevini kullanarak mesafeyi hesaplar ve açı ve mesafe değerlerini Seri monitöre yazdırır.

Birinci döngüden sonra, servo motoru 1 derecelik artışlarla 165 dereceden 15 dereceye çıkaran başka bir döngü vardır. Bu döngü içindeki kodun geri kalanı, ilk döngüye benzer.

int calculateDistance() {

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = duration \* 0.034 / 2;

return distance;

}

measureDistance() işlevi, ultrasonik sensörü kullanarak mesafeyi hesaplar. TrigPin'i DÜŞÜK, ardından YÜKSEK ve ardından tekrar DÜŞÜK olarak ayarlayarak sensörü tetikler. pulseIn(echoPin, HIGH) kullanarak yankı darbesinin süresini ölçer. Süre daha sonra şuna dönüştürülür:

PROCESSİNG CODE AÇIKLAMA:

import processing.serial.\*;

import java.awt.event.KeyEvent;

import java.io.IOException;

Serial myPort;

String angle = "";

String distance = "";

String data = "";

String noObject;

float pixsDistance;

int iAngle, iDistance;

int index1 = 0;

int index2 = 0;

PFont orcFont;

Bu satırlar, gerekli kitaplıkları içe aktarır ve kodda kullanılan değişkenleri bildirir. Seri kitaplığı, bir seri bağlantı noktasıyla iletişim için kullanılır. Diğer değişkenler, seri bağlantı noktasından alınan verileri depolamak ve değiştirmek için kullanılı.

void setup() {

size(1366, 700);

smooth();

myPort = new Serial(this, "COM3", 9600); // Change the COM port

myPort.bufferUntil('.');

}

setup() işlevi başlangıçta bir kez yürütülür. Çizim tuvalinin boyutunu ayarlar, smooth() ile kenar yumuşatmayı etkinleştirir, Serial(this, "COM3", 9600) kullanarak seri bağlantı noktası iletişimini başlatır ve seri bağlantı noktasını gelen verileri belirli bir süre (' .') alınır.

void draw() {

// Drawing the radar visualization

fill(98, 245, 31);

noStroke();

fill(0, 4);

rect(0, 0, width, 1010);

fill(98, 245, 31);

drawRadar();

drawLine();

drawObject();

drawText();

}

Draw() işlevi, setup() işlevinden sonra art arda yürütülür. Birkaç yardımcı işlevi (drawRadar(),drawLine(),drawObject() vedrawText()) çağırarak radar görselleştirmesini çizer. Arka planı yeşil renkle doldurur, radar yay çizgilerini, açı çizgilerini, algılanan nesneyi ve metni çizer.

void serialEvent(Serial myPort) {

data = myPort.readStringUntil('.');

data = data.substring(0, data.length() - 1);

index1 = data.indexOf(",");

angle = data.substring(0, index1);

distance = data.substring(index1 + 1, data.length());

iAngle = int(angle);

iDistance = int(distance);

}

Seri bağlantı noktasından veri alındığında serialEvent() işlevi otomatik olarak çağrılır. Alınan verileri okur ve açı ve mesafe değerlerini çıkarır. Bu değerler daha sonra tamsayılara dönüştürülür ve iAngle ve iDistance değişkenlerinde saklanır.

void drawRadar() {

// Drawing the radar arcs and angle lines

pushMatrix();

translate(683, 700);

noFill();

strokeWeight(2);

stroke(98, 245, 31);

// draws the arc lines

arc(0, 0, 1300, 1300, PI, TWO\_PI);

arc(0, 0, 1000, 1000, PI, TWO\_PI);

arc(0, 0, 700, 700, PI, TWO\_PI);

arc(0, 0, 400, 400, PI, TWO\_PI);

// draws the angle lines

line(-700, 0, 700, 0);

line(0, 0, -700 \* cos(radians(30)), -700 \* sin(radians(30)));

line(0, 0, -700 \* cos(radians(60)), -700 \* sin(radians(60)));int calculateDistance(){

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance= duration\*0.034/2;

return distance;

}

PROCESSİNG CODE:

import processing.serial.\*; // kütüphane entegresi

import java.awt.event.KeyEvent;

import java.io.IOException;

Serial myPort;

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, 700);

smooth();

myPort = new Serial(this,"COM3", 9600); // Com portunu seçin

myPort.bufferUntil('.');

}

void draw() {

fill(98,245,31);

noStroke();

fill(0,4);

rect(0, 0, width, 1010);

fill(98,245,31); // yeþil renk

drawRadar();

drawLine();

drawObject();

drawText();

}

void serialEvent (Serial myPort) {

data = myPort.readStringUntil('.');

data = data.substring(0,data.length()-1);

index1 = data.indexOf(",");

angle= data.substring(0, index1);

distance= data.substring(index1+1, data.length());

iAngle = int(angle);

iDistance = int(distance);

}

void drawRadar() {

pushMatrix();

translate(683,700);

noFill();

strokeWeight(2);

stroke(98,245,31);

// draws the arc lines

arc(0,0,1300,1300,PI,TWO\_PI);

arc(0,0,1000,1000,PI,TWO\_PI);

arc(0,0,700,700,PI,TWO\_PI);

arc(0,0,400,400,PI,TWO\_PI);

// draws the angle lines

line(-700,0,700,0);

line(0,0,-700\*cos(radians(30)),-700\*sin(radians(30)));

line(0,0,-700\*cos(radians(60)),-700\*sin(radians(60)));

line(0,0,-700\*cos(radians(90)),-700\*sin(radians(90)));

line(0,0,-700\*cos(radians(120)),-700\*sin(radians(120)));

line(0,0,-700\*cos(radians(150)),-700\*sin(radians(150)));

line(-700\*cos(radians(30)),0,700,0);

popMatrix();

}

void drawObject() {

pushMatrix();

translate(683,700);

strokeWeight(9);

stroke(255,10,10); // kýrmýzý renk

pixsDistance = iDistance\*22.5;

// 40 cm ye kadar ölçer

if(iDistance<40){

line(pixsDistance\*cos(radians(iAngle)),-pixsDistance\*sin(radians(iAngle)),700\*cos(radians(iAngle)),-700\*sin(radians(iAngle)));

}

popMatrix();

}

void drawLine() {

pushMatrix();

strokeWeight(9);

stroke(30,250,60);

translate(683,700);

line(0,0,700\*cos(radians(iAngle)),-700\*sin(radians(iAngle)));

popMatrix();

}

void drawText() {

pushMatrix();

if(iDistance>40) {

noObject = "Out of Range";

}

else {

noObject = "In Range";

}

fill(0,0,0);

noStroke();

rect(0, 1010, width, 1080);

fill(98,245,31);

textSize(25);

text("10cm",800,690);

text("20cm",950,690);

text("30cm",1100,690);

text("40cm",1250,690);

textSize(40);

text("Object: " + noObject, 240, 1050);

text("Angle: " + iAngle +" °", 1050, 1050);

text("Distance: ", 1380, 1050);

if(iDistance<40) {

text(" " + iDistance +" cm", 1400, 1050);

}

textSize(25);

fill(98,245,60);

translate(390+960\*cos(radians(30)),780-960\*sin(radians(30)));

rotate(-radians(-60));

text("30°",0,0);

resetMatrix();

translate(490+960\*cos(radians(60)),920-960\*sin(radians(60)));

rotate(-radians(-30));

text("60°",0,0);

resetMatrix();

translate(630+960\*cos(radians(90)),990-960\*sin(radians(90)));

rotate(radians(0));

text("90°",0,0);

resetMatrix();

translate(760+960\*cos(radians(120)),1000-960\*sin(radians(120)));

rotate(radians(-38));

text("120°",0,0);

resetMatrix();

translate(840+900\*cos(radians(150)),920-960\*sin(radians(150)));

rotate(radians(-60));

text("150°",0,0);

popMatrix();

}

Arduino Ultrasonik Radar Projesi Malzemeleri:

Ultrasonik Sensör HC SR04

SG90 Servo Motor

Arduino Uno R3

Jumper Dupont Kablo

Amacı:

bir cismin varlığını tayin etme, bulunduğu yön ve uzaklığı ölçme işlevlerini yerine getirir.

Tanmı:

radyo dalgalarının yansıması yardımıyla uzaktaki nesneleri ve bu nesnelerin hız, kerteriz ve mesafesini tespit eden cihazdır.

PROJE RAPORU

YAZAN VE DÜZENLYEN:

Yunus Emre Ortakaya.

PROJE KONUSU:

Arduino Radar.

Tarih:21/05/2023

Saat:16.33