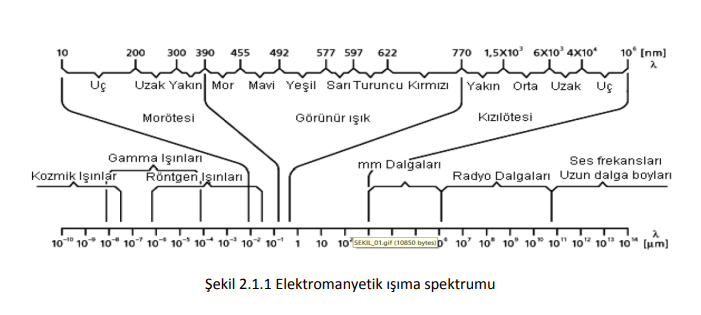
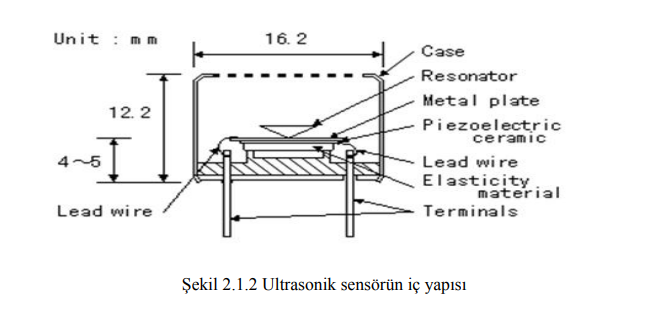
1. **ULTRASONİK RADAR PARÇALARI**

**7.1. ULTRASONİK SENSÖR**

Ses dalgalarının sınıflandırılmasında 20KHz ile 1GHz arasındaki ses sinyalleri ultrasonik dalgalar olarak tanımlanır. Ultrasonik cihazlar, aralarındaki mesafenin veya sıvı seviyesinin tahribatsız ölçülmesini gerektiren birçok endüstriyel alanda, ayrıca güvenlik alarm sistemleri ve robotik sistemlerde özellikle mobil robotlarda kullanılmaktadır. Elektromanyetik radyasyon spektrumu aşağıda verilmiştir.

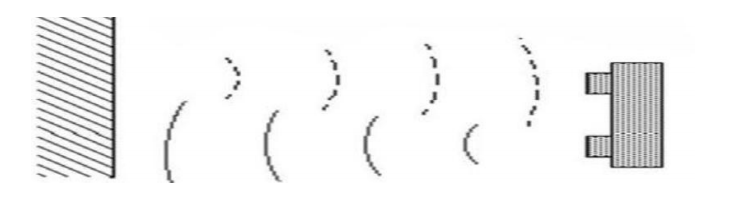


Şekil 7.1.1. Elektromanyetik ışıma spektrumu



Şekil 7.1.2. Ultrasonik sensörün iç yapısı

Ultrasonik dalgaları ileten verici parçanın ve ultrasonik dalgaları algılayan alıcı parçanın yapısında ince piezoelektrik seramikler bulunmaktadır. Ultrasonik mesafe sensörünün çalışma prensibi, bir piezoelektrik dönüştürücüden (hoparlör) kısa 40 kHz ultrasonik darbeler yaymaktır. Ses enerjisinin küçük bir kısmı sensörün önündeki nesneden yansır ve başka bir piezoelektrik seramiğe (mikrofon) ulaşır. Ultrasonik hoparlörden ses dalgalarının yayılması ile bu ses dalgalarının ultrasonik mikrofon tarafından engel tarafından algılanması arasındaki süre ölçülür.



Şekil 7.1.3. Ses dalgalarının engele çarpıp yansıması

Uygulama alanları;

* Yaklaşım uygulaması
* Boyutlandırma
* Seviye ölçümü
* Sınıflandırma , seçme

HCSR-04 Ultrasonik Sensör

Yapılan bu projede HC-SR04 Ultrasonik Mesafe Sensörü kullanılmış olup, bu ultrasonik sensör 2cm'den 400cm'ye kadar 3mm hassasiyetle ölçüm yapabilir ve çeşitli menzil, radar ve robotik uygulamalarında kullanılabilir. Maksimum izleme mesafesi: 4m Minimum izleme mesafesi: 2cm



Şekil 7.1.4. HC\_SR04 Ultrasonik Mesafe Sensörü

Kullanılan sensörün dört bacağı var :

Vcc:Sensor için 5 volt.

Trig:Sensörün çalışması için tetik sinyali üretme girişi

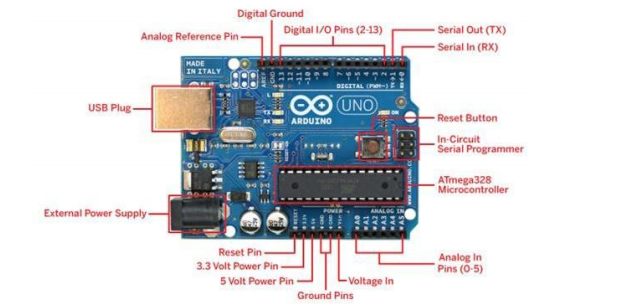
Echo:Sinyal çıkışı

Gnd:Toprak

|  |  |
| --- | --- |
| Çalışma Voltajı | DC 5V |
| Çektiği Akım | 15 mA |
| Çalışma Frekansı | 40 Hz |
| Maksimum Görme Menzili | 4m |
| Minimum Görme Menzili | 2cm |
| Görme Açısı | 15° |
| Tetik Bacağı Giriş Sinyali | 10 us TTL Darbesi |
| Echo Çıkış Sinyali | Giriş TTL sinyali ve Mesafe Oranı |

Tablo 7.1.1 HC\_SR04 Sensörünün Özellikleri

**7.2. ARDUİNO**

Arduino, temel olarak çeşitli sistemlerin tasarlanabileceği açık kaynaklı bir sistemdir. platformudur. Arduino kartında Atmega'dan 8 bit ve 32 bit mikrodenetleyiciler var. Bu mikrodenetleyiciler, PIC ile aynı kategoriye aittir. Piyasada en çok kullanılan PIC ve ARM gibi gömülü sistem yazılımlarına alternatif olarak doğmuştur. Daha kolay programlanabilir ve geniş bir kütüphaneye sahip olduğu için Kısa kodlar kullanarak karmaşık işlemleri gerçekleştirmenizi sağlayan bir platformdur. Mikrodenetleyici kendi kütüphanesi üzerinden kolaylıkla programlanabilir. Bu da kullanılır Pratiklik açısından. Analog ve dijital verileri işleyebilen girişlere sahiptir. Bir bilgisayardan veya başka bir ekipmandan veri alabilir ve ayrıca ses ve ışık gibi veriler de üretebilir. Arduino'nun Arduino Uno, Arduino Mega, Arduino Nano, Arduino Leonardo ve diğer çeşitleri vardır.

Şekil 7.2.1 Arduino Uno Üzerindeki Elemanlar

**7.3 Tower Pro Servo Motor 9g**

Tower Pro SG90, küçük kuruluşlar için ideal bir servo motordur. Her markadan olan uzaktan kumanda ile tam uyumludur, uzaktan kumandalı aracınızda kullanabilirsiniz. Ayrıca birçok mikro denetleyiciden elde edilen PWM sinyalleri sayesinde kendi robotik projelerinizde rahatlıkla kullanabilirsiniz.



Şekil 7.3.1. Tower Pro Servo Motor 9g

|  |  |
| --- | --- |
| Boyutlar: | 23.1 x 12.2 x 29 mm |
| Ağırlık | 9 g |
| Çalışma gerilimi: | 4.8 - 6.0 VDC |
| Hız | 0.06 - 0.12 sn/60° |
| Zorlanma Torku | 0-5 kg/cm |
| Dönüş açısı | 0-180° |
| Çalışma PWM sinyali | 500-2400 μs |
| Dijital/Analog | Analog |
| Dişli Tipi | Plastik |
| Kablo Uzunluğu | 15 cm |

## Tablo 7.3.1 SG90 Servo Motor Özellikleri

1. **Arduino Ultrasonik Radar Projesi**

Bu projede ultrasonik radar yapacağım. Bildiğiniz gibi askeri radarlar radyo frekanslarını yansıtarak ölçülür. Bu radarlar geniş bir tarama aralığına sahiptir ve uçakları tespit etmek için kullanılır. Bu projede benzer ama daha küçük bir proje yapacağım. Radarımız radyo dalgalarını değil, ses dalgalarını ölçer. Yapacağım radar 150º'lik bir alanı tarayabilir. Projem arduino ve işleme üzerine kurulu. Arduino bölümünü C ++ ile yazacağım.

Proje Malzemelerim:

•Arduino UNO

•Tower Pro Servo Motor 9g

•HCSR-04 Ultrasonik Sensör

•Jumper Kablo

Sonra işlem kısmını oluşturacağım. Arduino UNO Arduino, çevre birimleriyle temel giriş ve çıkış uygulamalarını gerçekleştirmek için işleme / kablolama dilini kullanan açık kaynaklı bir fiziksel programlama platformudur. Arduino'yu kullanarak etkileşimli uygulamaları bağımsız olarak uygulayabilirsiniz. Aynı zamanda Arduino'yu Flash, Processing, MaxMSP, C Sharp gibi çeşitli yazılımlar veya kendi yazdığınız yazılımlar aracılığıyla bilgisayarla iletişim kurmak için kullanabilirsiniz.

Windows, Mac OS X ve Linux platformları için açık kaynaklı arayüz yazılımını bu web sitesinden indirebilirsiniz. Arduino UNO R3, Arduino Uno'nun en son modelidir. Önceki modeldeki tüm işlevleri destekler. UNO R3 modeli için önceki versiyondaki 8U2 modeli yerine 16U2 modeli kullanıldı. Bu sayede daha az bellek kullanarak daha hızlı veri aktarımı sağlanabilir. Arduino'yu Linux ve Mac bilgisayarlardaki bilgisayarlara bağlamak için sürücü gerekmez. Windows bilgisayarda Arduino IDE yazılımının içerisinde bulunan inf dosyasını bilgisayarınıza aktarmanız yeterlidir. Bu sayede Arduino'yu bilgisayara klavye, fare, joystick ve benzeri aksesuarları bağlayabilir ve kullanılabilir hale getirebilirsiniz. UNO R3, kart düzeninde AREF pininin yanında bulunan ek SDA ve SCL pinlerine sahiptir. Ancak önceki versiyondan farklı olarak kart üzerindeki reset pininin yanına iki yeni pin eklendi.

Biri devre kartı üzerindeki blendaja güç sağlayan IOREF pimi, diğeri ise ileride kullanılmak üzere ayrılmış boşta bir bağlantı kesme pimidir. UNO R3, piyasadaki tüm koruyucu kapaklarla uyumludur ve yeni pimleri, bundan sonra üretilen yeni koruyucu kapaklarla da uyumludur.

* 1. **PROJE KODLARI**
     1. **Arduino Kodları**

#include <Servo.h>.

const int trigPin = 10;

const int echoPin = 11;

long duration;

int distance;

Servo myServo;

void setup() {

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

Serial.begin(9600);

myServo.attach(12); // Servo motor sinyal pini }

void loop() {

// 15 derece ile 165 derece arasında dön

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(".");

}

}

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;

}

* + 1. **Processing Kodları:**

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;

void setup() {

size (1366, 700);

smooth();

myPort = new Serial(this,"COM5", 9600);

myPort.bufferUntil('.'); }

void draw() {

fill(98,245,31);

noStroke();

fill(0,4);

rect(0, 0, width, 1010);

fill(98,245,31);

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); // kırmızı renk

pixsDistance = iDistance\*22.5;

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();

}

* + 1. **Ultrasonik Sensörün Değişmesi İle Mesafenin Artması İçin Oluşturulan Yeni Processing Kodu**

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;

void setup() {

size (1366,700);

smooth();

myPort = new Serial(this,"COM5", 9600);

myPort.bufferUntil('.');

}

void draw() {

fill(98,245,31);

noStroke();

fill(0,4);

rect(0, 0, width, 1010);

fill(98,245,31);

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 arc lines

arc(0,0,2200,2200,PI,TWO\_PI);

arc(0,0,1900,1900,PI,TWO\_PI);

arc(0,0,1600,1600,PI,TWO\_PI);

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); // kırmızı renk

pixsDistance = iDistance\*22.5;

if(iDistance<120){

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>120) {

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);

text("50cm",1400,690);

text("60cm",1550,690);

text("70cm",1700,690);

text("80cm",1850,690);

text("90cm",2000,690);

text("100cm",2150,690);

text("110cm",2300,690);

text("120cm",2450,690);

textSize(120);

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

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

text("Distance: ", 2450, 1050);

if(iDistance<120) {

text(" " + iDistance +" cm", 2450, 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();

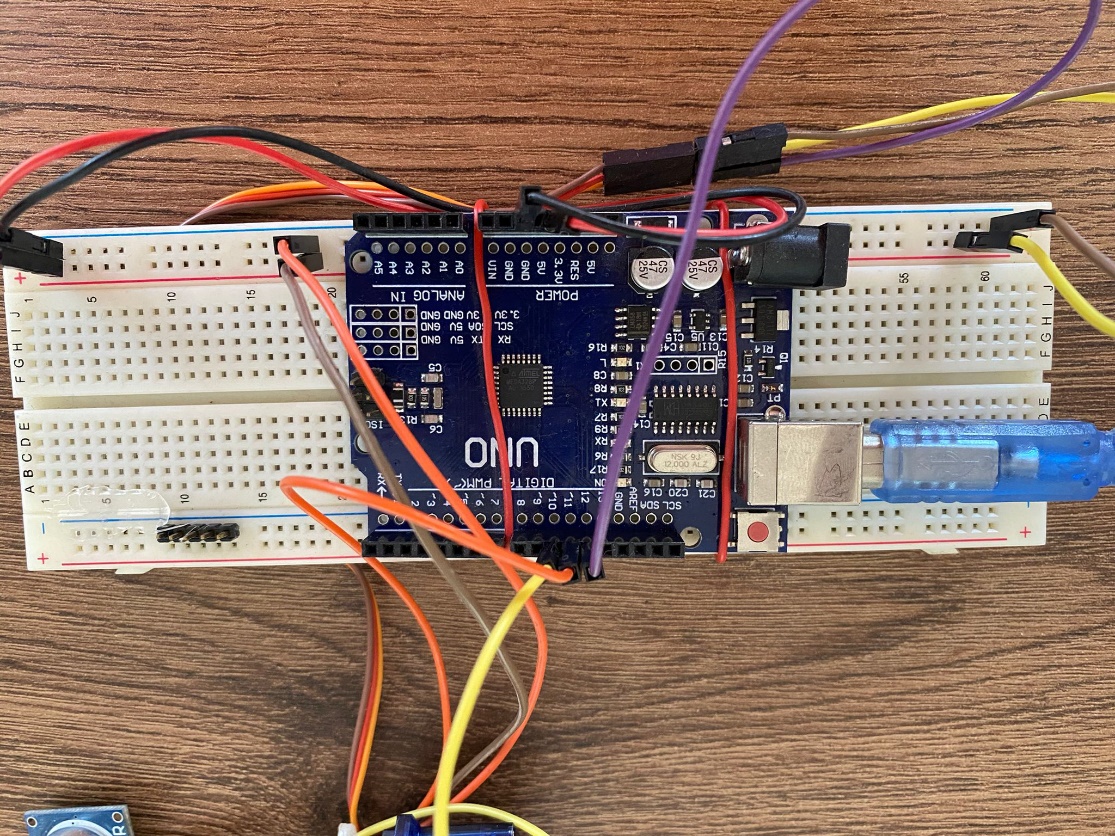
translate(840+900\*cos(radians(150)),920-960\*sin(radians(150)));

rotate(radians(-60));

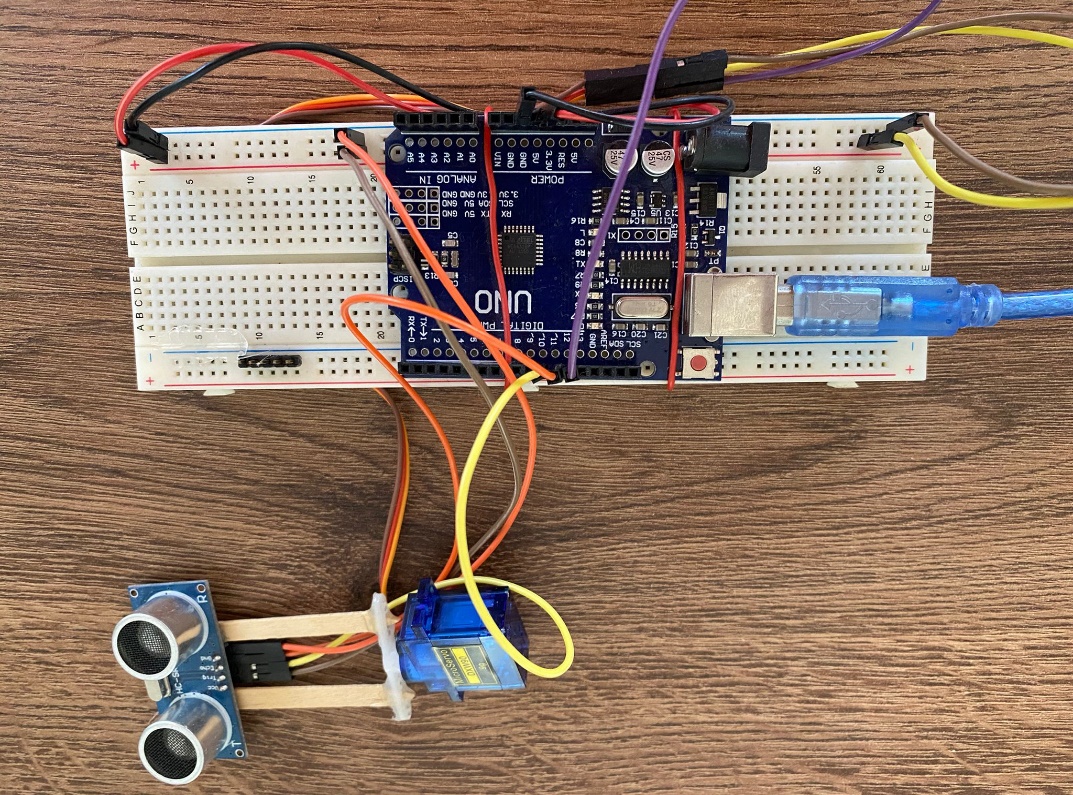
text("150°",0,0);

popMatrix();

}

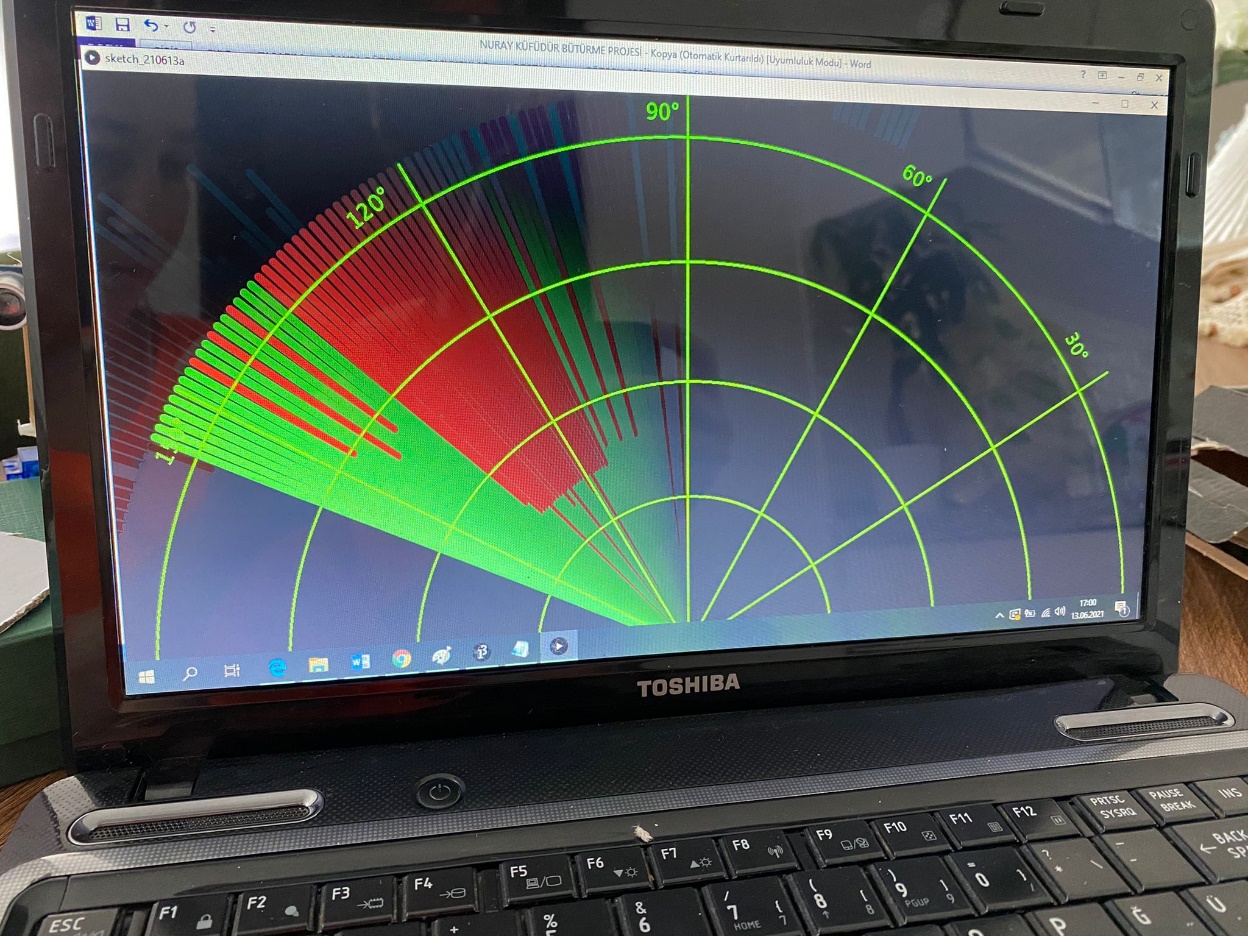


Şekil 8.1 ARDUİONO



Şekil 8.2 Arduino ve Ultrasonik Sensör



****