



T.C.

ONDOKUZ MAYIS ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ

RADAR TESPİT SİSTEMİ

Öğrenci numarası ve Ad Soyadı

17060056 Ezginaz YALÇIN

17060101 Muhammed Mert ÇAKIR

## **ÖZET**

Radarlar askeri veya sivil amaçlı kullanılmaktadır. Bunlar hava savunma, savaş alanı, hava trafik kontrol ve muhtelif radarlar olarak ayrılmaktadır. Radar sisteminin temel amacı cisimlerin veya nesnelerin algılanması ve uzaklığının tespit edilmesine yöneliktir. Tahmin edilen hedef üzerine gönderilen güçlü radyo dalgaları hedefe ulaştığında radar sistemine geri dönmektedir. Bu dönüş şekline göre hedefin ne kadar uzaklıkta olduğu ve hedefin büyüklüğü cihaz üzerinde görünmektedir. Bu sebeple bu projede radar tespit sistemi prototipi gerçekleştirildi.

Haziran, 2021

## İçindekiler

### 1.Giriş

Özet.....	2
İçindekiler .....	3
Önsöz .....	4
<b>2. Radarın Prototipinin Çalışma Prensibi.....</b>	<b>5</b>
<b>3. Projenin Yöntemi.....</b>	<b>6</b>
3.1.Radar Prototipinde Kullanılan Bileşenler.....	6
3.2.Prototip Devre Modeli.....	8
3.3.Radar Prototip Arduino Kodları.....	9
3.4.Radar Prototip Proccesing Arayüzü Kodları.....	11
<b>4. Radar Prototip Fotoğrafı ve Radar Sonucu.....</b>	<b>15</b>

## ÖNSÖZ

Çalışmamızı hazırlarken değerli vaktini bizlere ayırarak bilgi birikimini bizimle paylaşan ve bu ödevi bizlere vererek araştırıp bilgilerimizi güncellememize yardımcı olan Sayın hocamız Doç. Dr. Çetin KURNAZ'a teşekkürlerimizi sunarız.

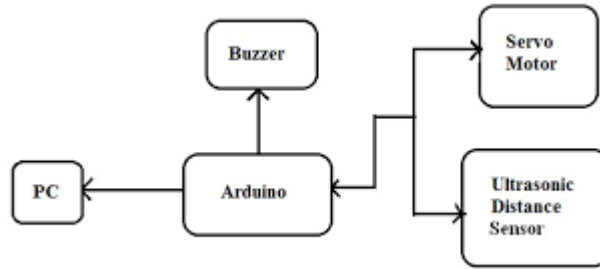
17600056 Ezginaz YALÇIN

17060101 Muhammed Mert ÇAKIR

## 2. Radar Prototipinin Çalışma Prensibi

Radar bir nesne algılama sistemidir. Nesnelerin menzilini, yüksekliğini, yönünü veya hızını belirlemek için mikrodalgaları kullanır. Radar, yoluna çıkan herhangi bir nesneden yansıyan radyo dalgalarını veya mikrodalgaları iletebilir. Böylece radar menzilineki herhangi bir nesneyi kolayca belirleyebiliriz. Arduino, elektroniği daha disiplinli hale getirmek için kullanılan tek kartlı bir mikro denetleyicidir. Radar sistemi farklı performans özelliklerine sahiptir ve aynı zamanda çok çeşitli boyutlarda gelir.

Ultrasonik sensör, Arduino yardımıyla nesne bilgilerini toplar ve Processing ara yüzüne iletir. İşleme ara yüzünde, bir radar ekranını taklit eden basit bir grafik uygulaması vardır.



Ultrasonik Sensör menzili içinde herhangi bir nesne tespit ettiğinde, nesnenin algılandığı yerde radarımızın rengi yeşilden kırmızıya döner. Videomuzda daha detaylı olarak gösterilmiştir.

Radar tespit sistemi uygulamamızda kullanılan malzemeler, devre şeması ve kodlar aşağıda açıklanmıştır.

### 3. Projenin Yöntemi

#### 3.1.Radar Prototipinde Kullanılan Bileşenler

##### Arduino Uno

- Mikrodenetleyici: ATmega328
- Çalışma Gerilimi: 5V
- Giriş Gerilimi (önerilen): 7-12V
- Giriş Gerilimi (limit): 6-20V
- Dijital I/O Pinleri: 14 (6 tanesi PWM çıkışı)



##### HC-SR04 Ultrasonik Mesafe Sensörü

- Çalışma Voltajı: DC 5V
- Çektiği Akım: 15 mA
- Çalışma Frekansı: 40 Hz
- Maksimum Görme Menzili: 4m
- Minimum Görme Menzili: 2cm



## Servo Motor

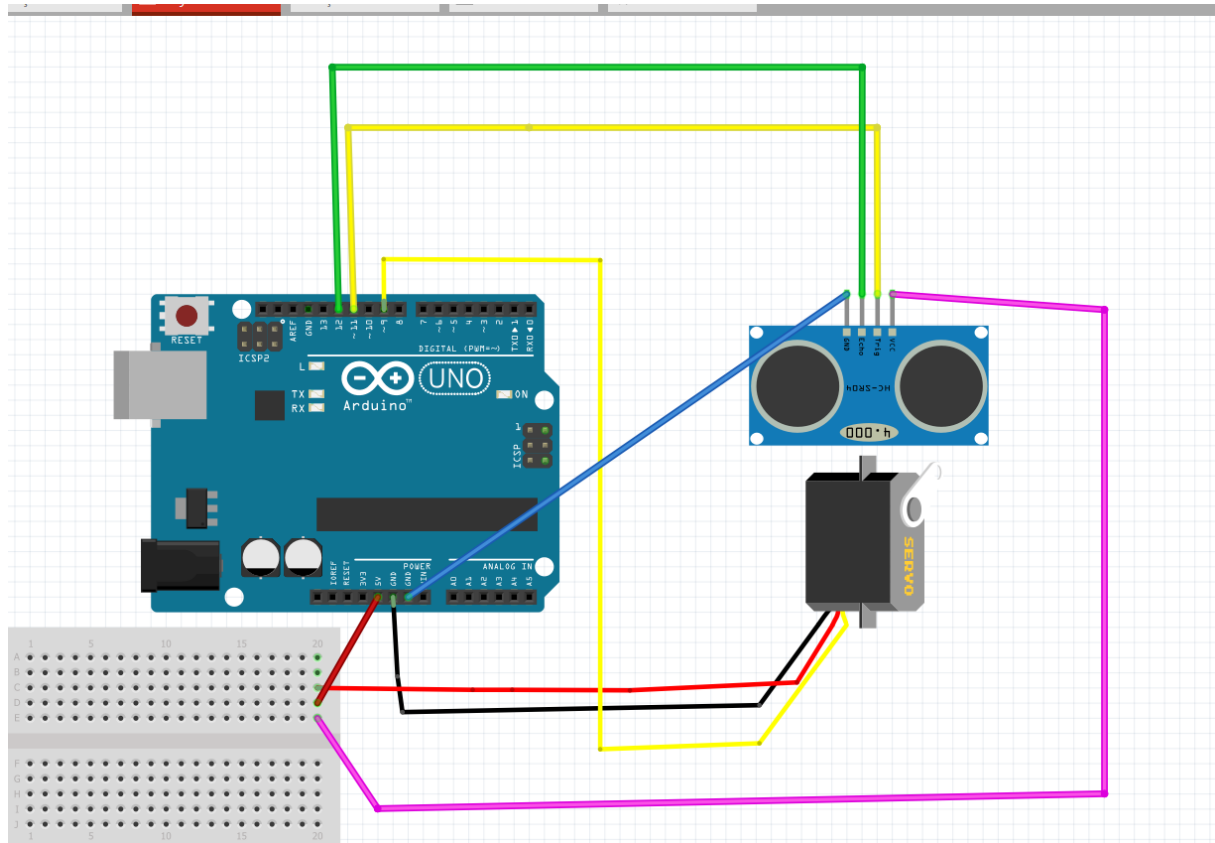
- Boyutlar: 23.1 x 12.2 x 29 mm
- Ağırlık: 9 g
- Çalışma gerilimi: 4.8 - 6.0 VDC
- Hız @4.8V: 0.1 sn/60°
- Zorlanma Torku @6V: 1.8 kg.cm
- Dişli kutusu: Plastik



## 3. Jumper Kablo

## 4. Breadboard

### 3.2. Prototip Devre Modeli



Bu devre modeline göre malzemelerimizi birleştirdik ve devremizi kurduk.



### 3.3. Radar Prototipi Arduino Kodları

```
radar | Arduino 1.8.13
Dosya Düzenle Taslak Araçlar Yardım

✓ → 📄 ⬆ ⬇

radar

#include<Servo.h>

// set ouptut pins
const int TriggerPin = 11;
const int EchoPin = 12;
const int motorSignalPin = 9;

// starting loaction
const int startAngle = 90;

// rotation limits
const int minimumAngle = 6;
const int maximumAngle = 145;

// speed
const int degreesPerCycle = 1;

// Library class instance
Servo motor;

void setup(void)
{
    pinMode(TriggerPin, OUTPUT);
    pinMode(EchoPin, INPUT);
    motor.attach(motorSignalPin);
    Serial.begin(9600);
}

void loop(void)
{
    static int currentAngle = startAngle;
    static int motorRotateAmount = degreesPerCycle;
```

radar | Arduino 1.8.13

Dosya Düzenle Taslak Araçlar Yardım

✓ ↻ 📄 ⬆ ⬇

radar\$

```
motor.write(currentAngle);
delay(10);

SerialOutput(currentAngle, CalculateDistance());

currentAngle += motorRotateAmount;

if(currentAngle <= minimumAngle || currentAngle >= maximumAngle)
{
    motorRotateAmount = -motorRotateAmount;
}
}

int CalculateDistance(void)
{
    |
    digitalWrite(TriigerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(TriigerPin, LOW);
    long duration = pulseIn(EchoPin, HIGH);

    float distance = duration * 0.017F;
    return int(distance);
}

void SerialOutput(const int angle, const int distance)
{
    Serial.println(String(angle) + "," + String(distance));
}
```

### 3.4.Radar Prototip Proccesing Arayüzü Kodları

```
radar ▼
1 import processing.serial.*;
2 import java.awt.event.KeyEvent;
3 import java.io.IOException;
4
5 Serial myPort;
6 PFont orcFont;
7 int iAngle;
8 int iDistance;
9
10 void setup() {
11     size(1300, 800);
12     smooth();
13
14     myPort = new Serial(this, "COM3", 9600);
15     myPort.clear();
16     myPort.bufferUntil('\n');
17     orcFont = loadFont("ACaslonPro-Semibold-48.vlw");
18 }
19 void draw()
20 {
21     fill(98, 245, 31);
22     textFont(orcFont);
23     noStroke();
24     fill(0, 4);
25     rect(0, 0, width, 0.935 * height);
26     fill(98, 245, 31);
27
28     DrawRadar();
29     DrawLine();
30     DrawObject();
31     DrawText();
32 }
33 void serialEvent (Serial myPort)
34 {
```

radar

```
35     try {
36         String data = myPort.readStringUntil('\n');
37         if (data == null) {
38             return;
39         }
40         int commaIndex = data.indexOf(",");
41         String angle = data.substring(0, commaIndex);
42         String distance = data.substring(commaIndex+1, data.length()-1);
43         iAngle = StringToInt(angle);
44         iDistance = StringToInt(distance);
45     } catch(RuntimeException e) {}
46 }
47 void DrawRadar()
48 {
49     pushMatrix();
50     translate(width/2, 0.926 * height);
51     noFill();
52     strokeWeight(2);
53     stroke(98, 245, 31);
54
55     // draws the arc lines
56     DrawRadarArcLine(0.9375);
57     DrawRadarArcLine(0.7300);
58     DrawRadarArcLine(0.5210);
59     DrawRadarArcLine(0.3130);
60
61     // draws the angle lines
62     final int halfWidth = width/2;
63     line(-halfWidth, 0, halfWidth, 0);
64     for(int angle = 30; angle <= 150; angle+=30) {
65         DrawRadarAngledLine(angle);
66     }
67     line(-halfWidth * cos(radians(30)), 0, halfWidth, 0);
68     popMatrix();
```

radar

```
68     popMatrix();
69 }
70 void DrawRadarArcLine(final float coefficient)
71 {
72     arc(0, 0, coefficient * width, coefficient * width, PI, TWO_PI);
73 }
74 void DrawRadarAngledLine(final int angle){
75     line(0, 0, (-width/2) * cos(radians(angle)), (-width/2) * sin(radians(angle)));
76 }
77 void DrawObject()
78 {
79     pushMatrix();
80     translate(width/2, 0.926 * height);
81     strokeWeight(9);
82     stroke(255,10,10);
83     int pixsDistance = int(iDistance * 0.020835 * height);
84     if(iDistance < 40 && iDistance != 0) {
85         float cos = cos(radians(iAngle));
86         float sin = sin(radians(iAngle));
87         int x1 = +int(pixsDistance * cos);
88         int y1 = -int(pixsDistance * sin);
89         int x2 = +int(0.495 * width * cos);
90         int y2 = -int(0.495 * width * sin);
91
92         line(x1, y1, x2, y2);
93     }
94     popMatrix();
95 }
96 void DrawLine()
97 {
98     pushMatrix();
99     strokeWeight(9);
100    stroke(30, 250, 60);
101    translate(width/2, 0.926 * height);
```

radar

```
101     translate(width/2, 0.926 * height);
102
103     float angle = radians(iAngle);
104     int x = int(+0.88 * height * cos(angle));
105     int y = int(-0.88 * height * sin(angle));
106     line(0, 0, x, y);
107     popMatrix();
108 }
109 void DrawText()
110 {
111     pushMatrix();
112     fill(0, 0, 0);
113     noStroke();
114     rect(0, 0.9352 * height, width, height);
115     fill(98, 245, 31);
116     textSize(25);
117     text("10cm", 0.6146 * width, 0.9167 * height);
118     text("20cm", 0.7190 * width, 0.9167 * height);
119     text("30cm", 0.8230 * width, 0.9167 * height);
120     text("40cm", 0.9271 * width, 0.9167 * height);
121
122     textSize(40);
123     text("Object: " + (iDistance > 40 ? "Out of Range" : "In Range"), 0.125 * width, 0.9723 * height);
124     text("Angle: " + iAngle + " °", 0.52 * width, 0.9723 * height);
125     text("Distance: ", 0.74 * width, 0.9723 * height);
126     if(iDistance < 40) {
127         text("      " + iDistance + " cm", 0.775 * width, 0.9723 * height);
128     }
129     textSize(25);
130     fill(98, 245, 60);
131     translate(0.5006 * width + width/2 * cos(radians(30)), 0.9093 * height - width/2 * sin(radians(30)));
132     rotate(-radians(-60));
133     text("30°",0,0);
134 }
```

radar

```
133     text("30°",0,0);
134
135     resetMatrix();
136
137     translate(0.497 * width + width/2 * cos(radians(60)), 0.9112 * height - width/2 * sin(radians(60)));
138     rotate(-radians(-30));
139     text("60°",0,0);
140     resetMatrix();
141     translate(0.493 * width + width/2 * cos(radians(90)), 0.9167 * height - width/2 * sin(radians(90)));
142     rotate(radians(0));
143     text("90°",0,0);
144     resetMatrix();
145
146     translate(0.487 * width + width/2 * cos(radians(120)), 0.92871 * height - width/2 * sin(radians(120)));
147     rotate(radians(-30));
148     text("120°",0,0);
149     resetMatrix();
150
151     translate(0.4896 * width + width/2 * cos(radians(150)), 0.9426 * height - width/2 * sin(radians(150)));
152     rotate(radians(-60));
153     text("150°",0,0);
154     popMatrix();
155 }
156
157 int StringToInt(String string)
158 {
159     int value = 0;
160     for(int i = 0; i < string.length(); ++i) {
161         if(string.charAt(i) >= '0' && string.charAt(i) <= '9') {
162             value *= 10;
163             value += (string.charAt(i) - '0');
164         }
165     }
166     return value;
}
```

#### 4. Radar Prototip Fotoğrafi ve Radar Sonucu

