



Kayseri Üniversite

Teknik Bilimler Meslek Yüksekokulu

Bilgisayar Programcılığı

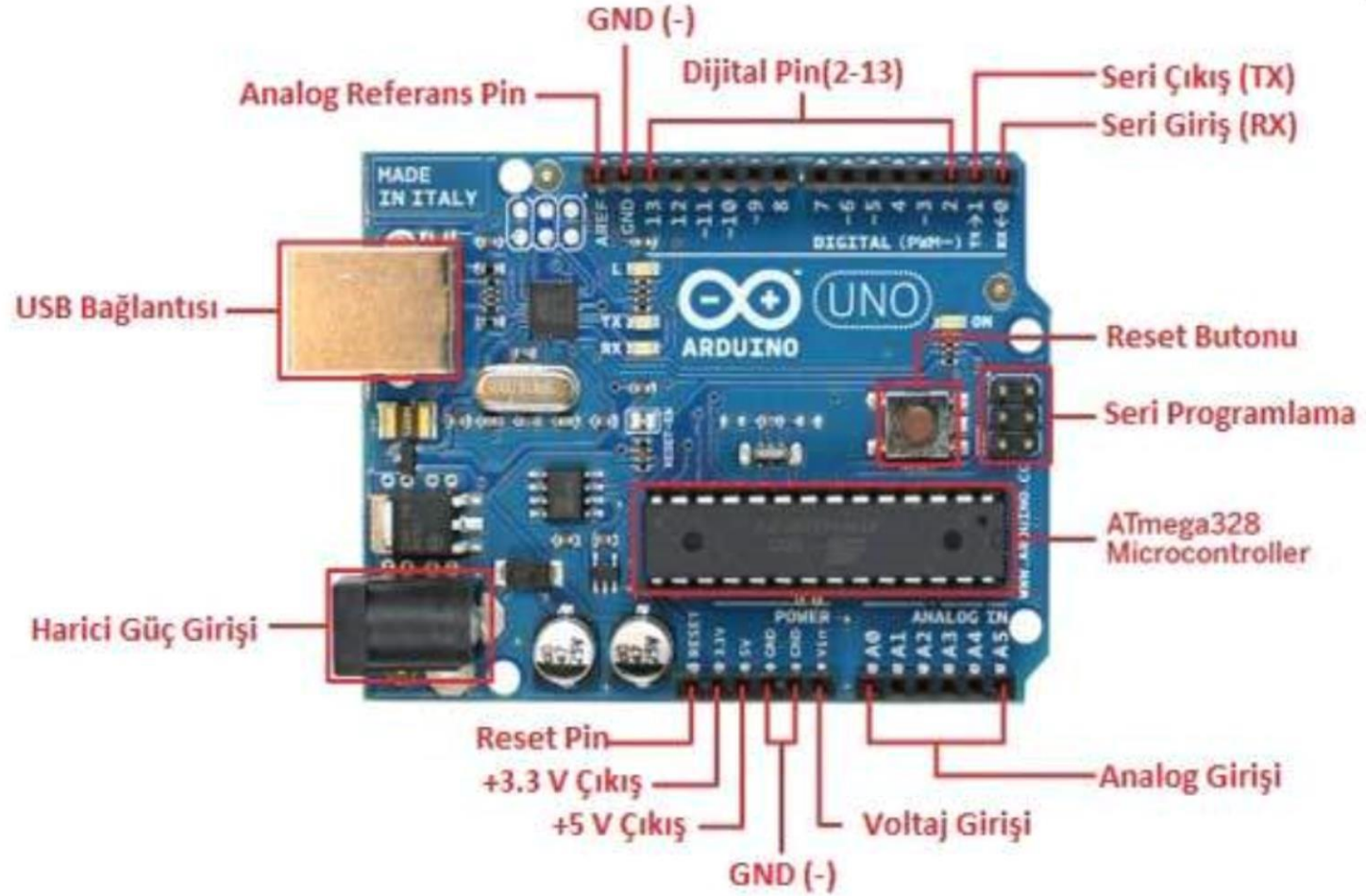
Hazırlayan : Mustafa Muhammedi

E-Mail : mustafa.muhammedi1396@gmail.com



Arduino UNO

Arduino Uno, ATmega328P mikrodeneetleyiciye sahip, 14 dijital ve 6 analog giriş/çıkış pini bulunan, USB bağlantılı bir mikrodeneetleyici geliştirme kartıdır. Kolay kullanımı, geniş kullanıcı topluluğu ve çeşitli projeler için bol miktarda kaynak ve kütüphane desteği ile bilinir.



RAdio Detection And Rangin

Radar : Elektromanyetik dalgaları kullanarak cisimlerin konumunu, mesafesine, hızını ve diğer özelliklerini belirlemek için kullanılan bir teknolojidir.

Bir Radar sistemi genellikle Verici, Alıcı, Anten, İşlem birimi Temel bileşenlerden oluşur.



Radar Ürünleri

- 1 Ardunio UNO
- 2 Data Kablu
- 3 Servo Motoru
- 4 Ultrasonik Sensör
- 5 Sensör Tutucu
- 6 Erkek ve Dişi kablo



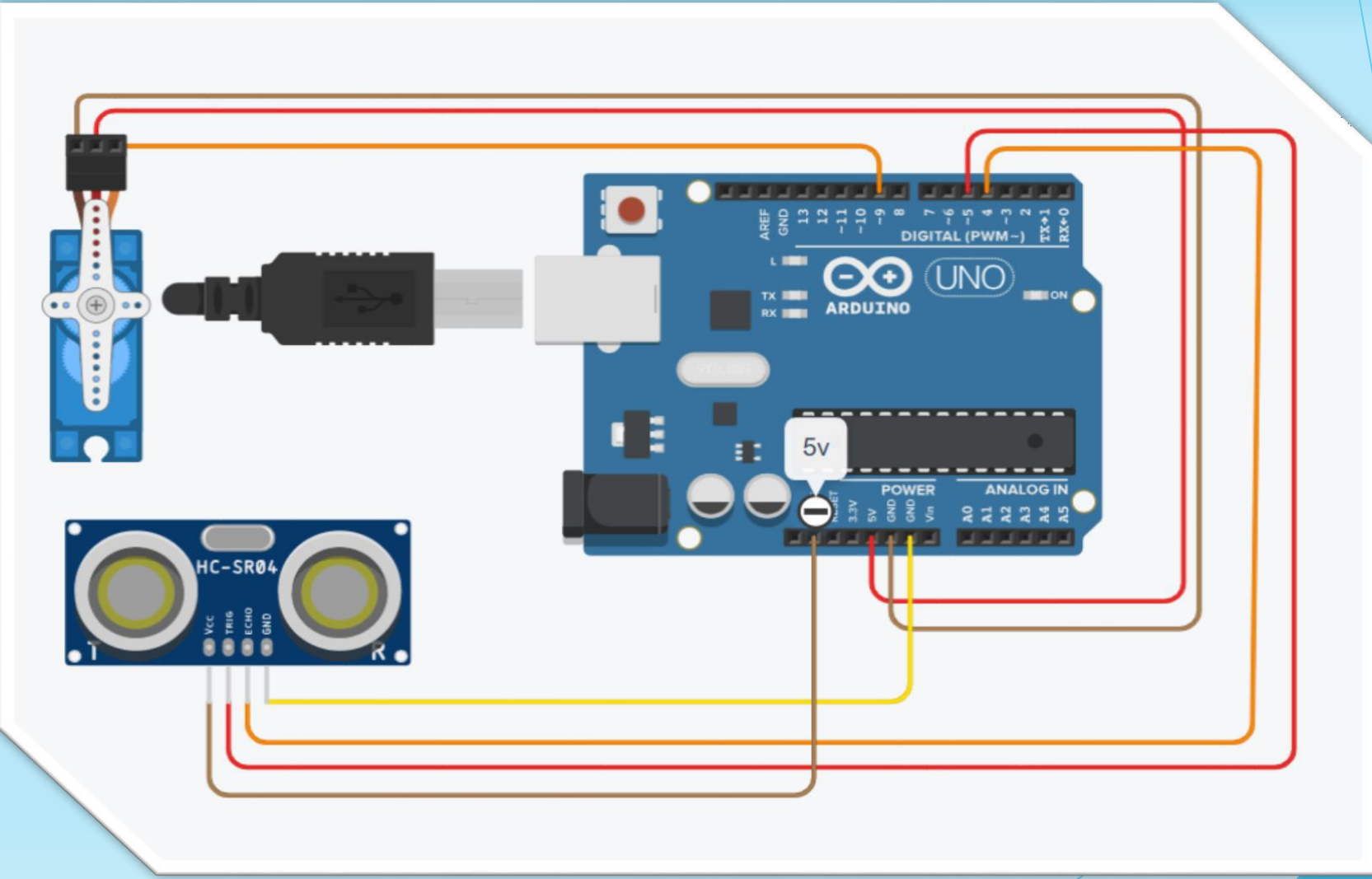
Servo Motor Bağlantılar

- ▶ Kahverengi (Toprak):GND bağlanır
- ▶ Kırmızı rengi(Güç) : 5v bağlanılır
- ▶ Turuncu rengi (Sinyal) : -9 bağlanılır

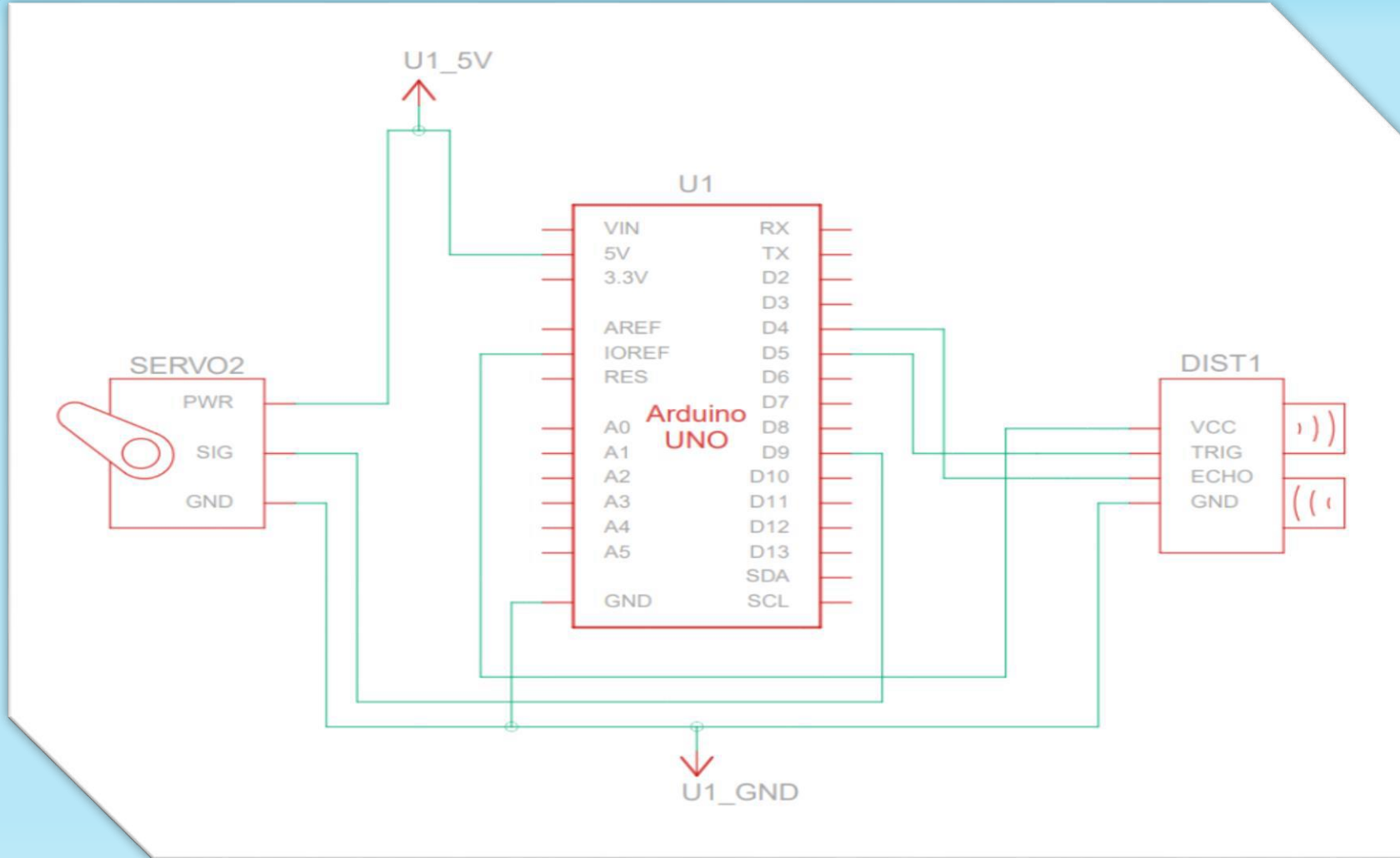


Ultrasonik Sensör Bağlantıları

- ▶ Kahve rengi(Vcc (Güç)) : 5V bağlanılır
- ▶ Kırmızı rengi(TRING (Tetikleyici)) : -5 bağlanılır
- ▶ Turuncu rengi(ECHO (Eko)) : 4 bağlanılır
- ▶ Sarı rengi(GND(Toprak)) : GND ye bağlanılır



Şematik Görünüm



//-----Arduino kodları-----

```
1. #include <Servo.h>
2. const int trigPin = 4;
3. const int echoPin = 3;
4. long duration;
5. int distance;
6. Servo myServo;

7. void setup() {
8.   pinMode(trigPin, OUTPUT);
9.   pinMode(echoPin, INPUT);
10. Serial.begin(9600);
11. myServo.attach(9);

12. }
```

```
13. void loop() {  
  
14.   for(int i=15;i<=165;i++){  
15.     myServo.write(i);  
16.     delay(30);  
17.     distance = calculateDistance();  
18.     Serial.print(i);  
19.     Serial.print(",");  
20.     Serial.print(distance);  
21.     Serial.print(".");  
22.   }
```

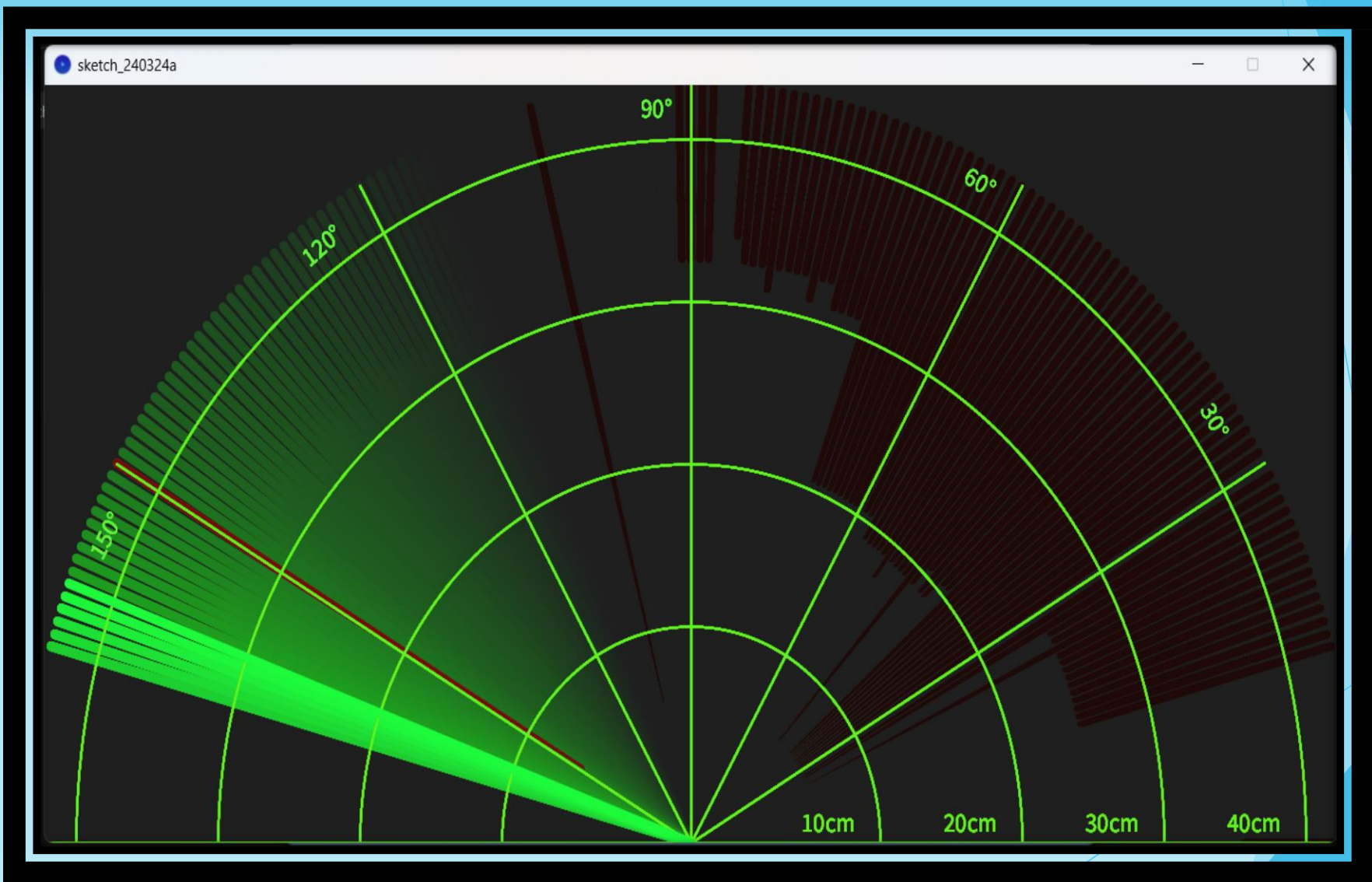
```
23.  for(int i=165;i>15;i--){  
  
24.    myServo.write(i);  
25.    delay(30);  
26.    distance = calculateDistance();  
27.    Serial.print(i);  
28.    Serial.print(",");  
29.    Serial.print(distance);  
30.    Serial.print(".");  
  
31.  }  
32. }
```

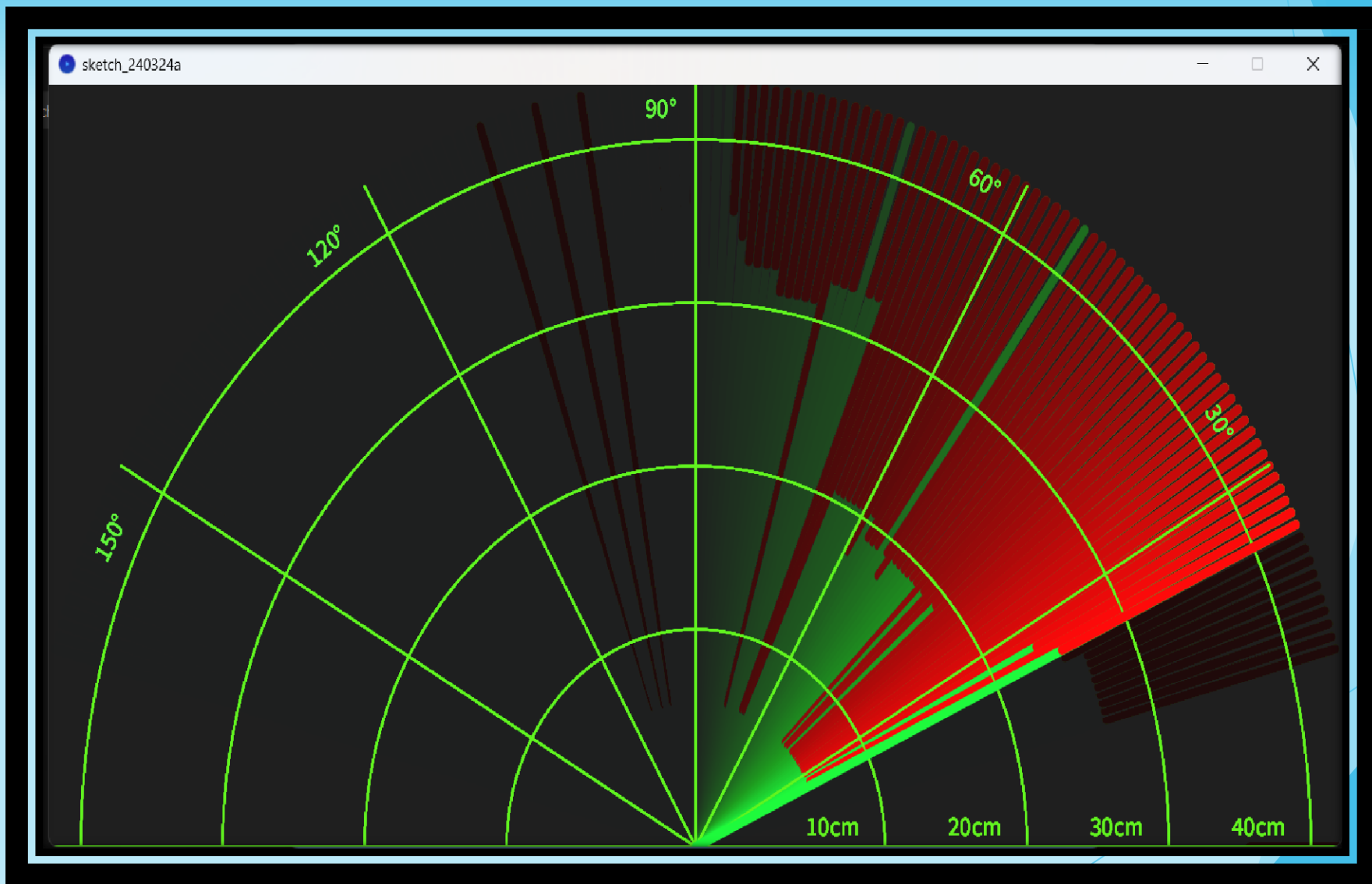
```
33. int calculateDistance(){
34.
35.   digitalWrite(trigPin, LOW);
36.   delayMicroseconds(2);
37.   digitalWrite(trigPin, HIGH);
38.   delayMicroseconds(10);
39.   digitalWrite(trigPin, LOW);
40.   duration = pulseIn(echoPin, HIGH);
41.   distance= duration*0.034/2;
42.   return distance;
43. }
```



Processing

genellikle görsel sanatlar, tasarım ve programlama alanlarında kullanılan bir açık kaynaklı programlama dilidir. Java tabanlıdır ve özellikle görsel sanatlar, animasyonlar, veri görselleştirmesi ve interaktif medya gibi uygulamalar için popülerdir.







Radar Tasarımı

Bu Processing dilinde yazılmıştır.bunun Kod, bir radar tarayıcısının simülasyonunu oluşturur. Aşağıdaki işlevleri yerine getirir



işlevleri

- ▶ 1. Seri porttan gelen verileri okur.
- ▶ 2. Verileri açı ve mesafe olarak ayırır.
- ▶ 3. Radar tarayıcısını ve algılanan nesnelerin konumunu çizer.
- ▶ 4. Nesne varsa açı ve mesafeyi görüntüler.
- ▶ 5. Nesne yoksa "Out of Range" mesajını gösterir.
- ▶ 6. Gerekli grafikleri çizer



Kodda önemli bazı bölümler

- **setup() Fonksiyonu:** İşlev, Processing penceresinin boyutunu ve seri bağlantıyı başlatmak için gerekli ayarları yapar.
- **draw() Fonksiyonu:** Bu fonksiyon, ekranda radar benzeri bir görüntü oluşturmak için diğer alt fonksiyonları çağırır.
- **serialEvent() Fonksiyonu:** Bu fonksiyon, Arduino'dan seri bağlantı yoluyla gelen veriyi işler. Veri, açı ve mesafe olmak üzere iki parçaya ayrılır ve işlenir.



- **drawRadar() Fonksiyonu:** Bu fonksiyon, radarın temel yapılarını çizer. Yani dairenin içine açıları ve bazı mesafe halkalarını çizer.
- **drawObject() Fonksiyonu:** Bu fonksiyon, radarın tespit ettiği nesneyi çizer. Nesne, kırmızı bir çizgi olarak gösterilir ve uzaklık ve açı bilgileri kullanılarak konumu belirlenir.



- **drawLine() Fonksiyonu:** Bu fonksiyon, radarın önündeki bir nesneye doğru çizgi çizer.
- **drawText() Fonksiyonu:** Bu fonksiyon, ekranda metin bilgilerini gösterir. Örneğin, algılanan nesnenin açısı ve mesafesi gibi bilgileri gösterir. Ayrıca, belirli açılara karşılık gelen metin etiketlerini çizer.



```
1  import processing.serial.*; // kütüphane entegresi
2  import java.awt.event.KeyEvent;
3  import java.io.IOException;
4
5  Serial myPort;
6
7  String angle="";
8  String distance="";
9  String data="";
10 String noObject;
11 float pixsDistance;
12 int iAngle, iDistance;
13 int index1=0;
14 int index2=0;
15 PFont orcFont;
16
17 void setup() {
18
19     size (1366, 700);
20     smooth();
21     myPort = new Serial(this,"COM3", 9600); // Com portunu seçin
22     myPort.bufferUntil('.');
23
24 }
```



```
25 |
26 void draw() {
27
28     fill(98,245,31);
29
30     noStroke();
31     fill(0,4);
32     rect(0, 0, width, 1010);
33
34     fill(98,245,31); // yepil renk
35
36     drawRadar();
37     drawLine();
38     drawObject();
39     drawText();
40 }
41
42 void serialEvent (Serial myPort) {
43
44     data = myPort.readStringUntil('.');
45     data = data.substring(0,data.length()-1);
46
47     index1 = data.indexOf(",");
48     angle= data.substring(0, index1);
49     distance= data.substring(index1+1, data.length());
50
51
52     iAngle = parseInt(angle);
53     iDistance = parseInt(distance);
54 }
```

```
55
56 void drawRadar() {
57     pushMatrix();
58     translate(683,700);
59     noFill();
60     strokeWeight(2);
61     stroke(98,245,31);
62     // draws the arc lines
63     arc(0,0,1300,1300,PI,TWO_PI);
64     arc(0,0,1000,1000,PI,TWO_PI);
65     arc(0,0,700,700,PI,TWO_PI);
66     arc(0,0,400,400,PI,TWO_PI);
67     // draws the angle lines
68     line(-700,0,700,0);
69     line(0,0,-700*cos(radians(30)),-700*sin(radians(30)));
70     line(0,0,-700*cos(radians(60)),-700*sin(radians(60)));
71     line(0,0,-700*cos(radians(90)),-700*sin(radians(90)));
72     line(0,0,-700*cos(radians(120)),-700*sin(radians(120)));
73     line(0,0,-700*cos(radians(150)),-700*sin(radians(150)));
74     line(-700*cos(radians(30)),0,700,0);
75     popMatrix();
76 }
```




```
77
78 void drawObject() {
79     pushMatrix();
80     translate(683,700);
81     strokeWeight(9);
82     stroke(255,10,10); // kırmızı renk
83     pixsDistance = iDistance*22.5;
84     // 40 cm ye kadar ölçer
85     if(iDistance<40){
86
87         line(pixsDistance*cos(radians(iAngle)),-pixsDistance*sin(radians(iAngle)),700*cos(radians(iAngle)),-700*sin(radians(iAngle)));
88     }
89     popMatrix();
90 }
91
92 void drawLine() {
93     pushMatrix();
94     strokeWeight(9);
95     stroke(30,250,60);
96     translate(683,700);
97     line(0,0,700*cos(radians(iAngle)),-700*sin(radians(iAngle)));
98     popMatrix();
99 }
```

```
100
101 void drawText() {
102
103     pushMatrix();
104     if(iDistance>40) {
105         noObject = "Out of Range";
106     }
107     else {
108         noObject = "In Range";
109     }
110     fill(0,0,0);
111     noStroke();
112     rect(0, 1010, width, 1080);
113     fill(98,245,31);
114     textSize(25);
115     text("10cm",800,690);
116     text("20cm",950,690);
117     text("30cm",1100,690);
118     text("40cm",1250,690);
119     textSize(40);
120     text("Object: " + noObject, 240, 1050);
121     text("Angle: " + iAngle + " °", 1050, 1050);
122     text("Distance: ", 1380, 1050);
123     if(iDistance<40) {
124         text("          " + iDistance + " cm", 1400, 1050);
125     }
```



```
126  textSize(25);
127  fill(98,245,60);
128  translate(390+960*cos(radians(30)),780-960*sin(radians(30)));
129  rotate(-radians(-60));
130  text("30°",0,0);
131  resetMatrix();
132  translate(490+960*cos(radians(60)),920-960*sin(radians(60)));
133  rotate(-radians(-30));
134  text("60°",0,0);
135  resetMatrix();
136  translate(630+960*cos(radians(90)),990-960*sin(radians(90)));
137  rotate(radians(0));
138  text("90°",0,0);
139  resetMatrix();
140  translate(760+960*cos(radians(120)),1000-960*sin(radians(120)));
141  rotate(radians(-38));
142  text("120°",0,0);
143  resetMatrix();
144  translate(840+900*cos(radians(150)),920-960*sin(radians(150)));
145  rotate(radians(-60));
146  text("150°",0,0);
147  popMatrix();
148 }
```