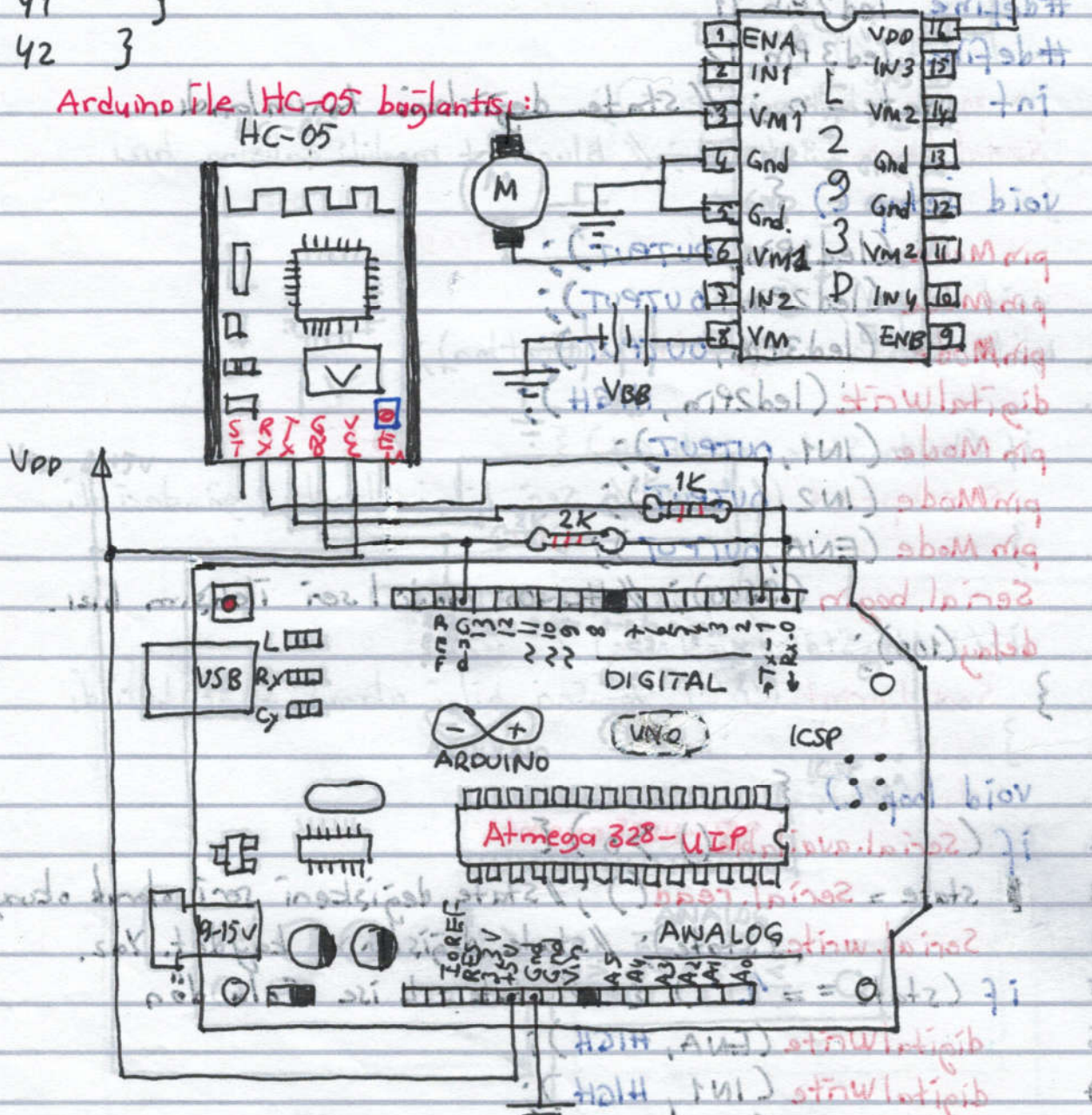


```

30 if (state == 'R') { // state = R ise Sağa dön
31   digitalWrite(ENA, HIGH);
32   digitalWrite(IN1, LOW);
33   digitalWrite(IN2, HIGH); }
34 state = 0; }
35
36 if (state == 'S') { // state = S ise Dur
37   digitalWrite(ENA, LOW);
38   digitalWrite(IN1, LOW);
39   digitalWrite(IN2, LOW);
40   state = 0; }
41 }
42 }

```

Arduino ile HC-05 bağlantısı:



Arduino ile HC-05 'yardımıyla Analog okuma BT analog RX.ino

Arduino ve HC-05 Bluetooth modülü kullanarak RobotX adını vermiş olduğumuz Bluetooth ile Cep Telefonumuzdan kurduğumuz Arduino Bluetooth uygulamasıyla eşleştirme yaparak (Pin kodu {1234} girilerek) bağlantı kurduğumuz program kodu yardımıyla bir led veya motor sürücüyü kontrol edebiliriz. Böylece analog bilgi gönderi leret motorun hızını uzaktan kontrolüne sağlayabiliriz.

```
1 #include <SoftwareSerial.h>
2 #include <Wire.h>
3 #define IN1 2
4 #define IN2 3
5 #define ENA 5
6 #define led1 9
7 int deger;
8
9 void setup () {
10   Serial.begin(9600); // Seri iletişimi başlat
11   Wire.begin(); // WiFi iletişimi başlatır. 2.4GHz.
12   pinMode(led1, OUTPUT);
13   pinMode(IN1, OUTPUT);
14   pinMode(IN2, OUTPUT);
15   pinMode(ENA, OUTPUT);
16   digitalWrite(IN1, HIGH); // L293D motor sürücü ile ayarlandı.
17   digitalWrite(IN2, LOW); }
18
19 void loop() {
20   if (Serial.available() > 0) {
21     while (Serial.available()) {
22       deger = Serial.read(); // Seri okunanları deger'e ata.
23       int ValueMapped = map(deger, 48, 57, 0, 255); // Analog okunan
24       analogWrite(led1, ValueMapped); // degeri (48-57) 0 ile 255
25       analogWrite(ENA, ValueMapped); // arasındaki değere çevirip
26     } // ValueMapped (0-255) olarak led1 ve ENA pinine uygular.
```

```

28 Serial.println(deger);
29 }

```

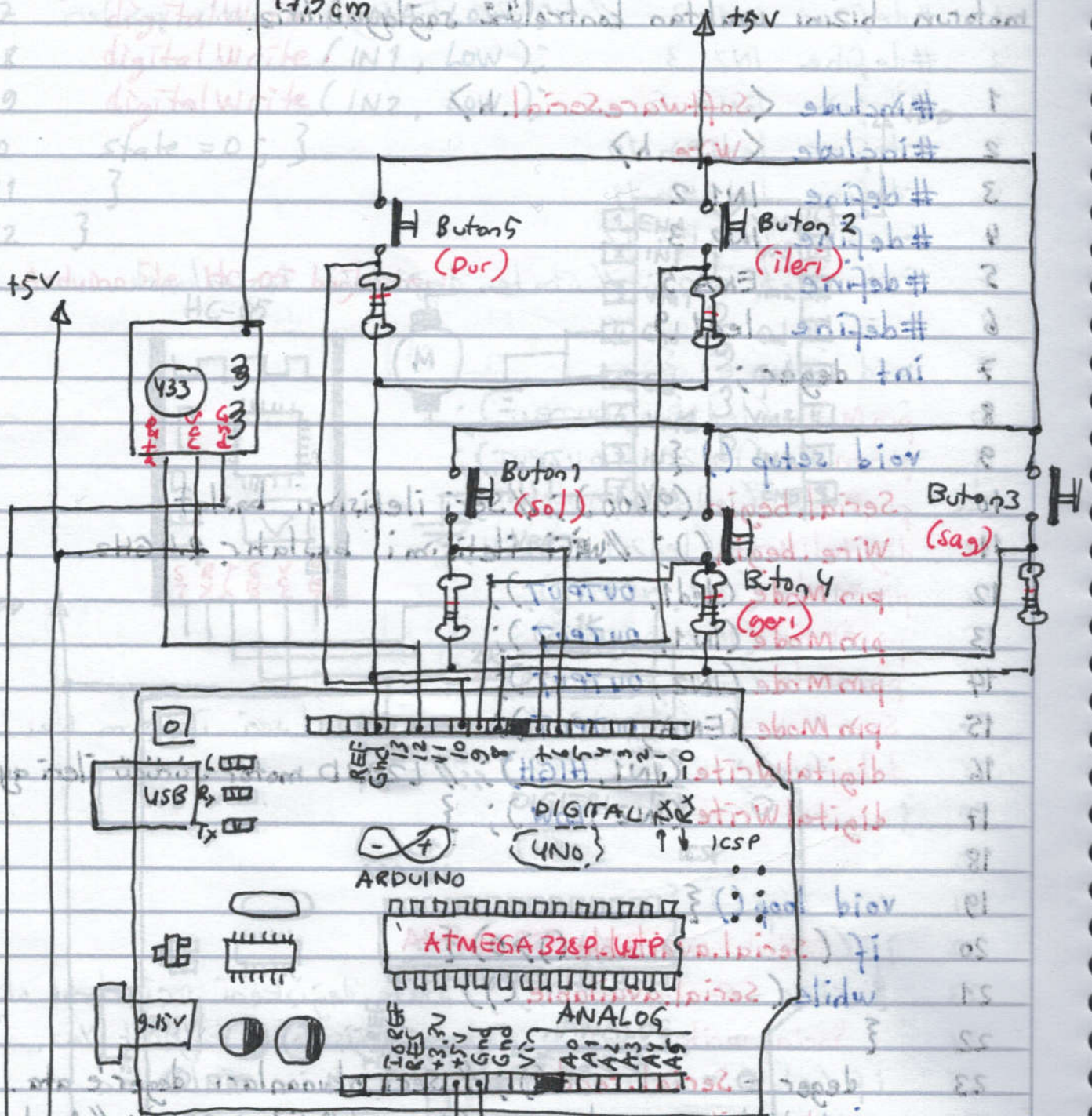
```

30 delay(10);
31 }

```

RF verici
Arduino ile 433 MHz RF alıcı bağlantısı

17,5 cm



Arduino ile RF (Radio Freqans) modüllerinin kullanımı için hem alıcı (RX) hem de verici (TX) için ayrı ayrı devreler ve iki tane Arduino kullanılarak birbirleriyle haberleşme sağlanabilir. Bu iletişimin için kullanılan frekans 433 MHz'dir.

Wire.h ve RH-ASK.h libraryleri Arduino programında program files/Arduino/library klasörü içine kurulmalıdır. Aşağıdaki program verici (TX) için kullanılabılır. Transmitter için data ucunu R12 pinine bağlanacaktır.

```

1 #include <Wire.h>
2 #include <RH-ASK.h>
3 RH-ASK driver;
4 int button1 = 6; // sol buton
5 int button2 = 7; // ileri buton
6 int button3 = 8; // sag buton
7 int button4 = 9; // geri buton
8 int button5 = 10; // dur buton
9 const char *b1 = "buton1";
10 const char *b2 = "buton2";
11 const char *b3 = "buton3";
12 const char *b4 = "buton4";
13 const char *b5 = "buton5";
14 uint8_t mesaj = 0;
15
16 void setup() {
17   Serial.begin(9600);
18   if (!driver.init()) Serial.println("RF433 hatasiz");
19   pinMode(button1, INPUT);
20   pinMode(button2, INPUT);
21   pinMode(button3, INPUT);
22   pinMode(button4, INPUT);
23   pinMode(button5, INPUT);
24   driver.init();
25   Serial.println((char *)buf);
26   rf_gelen = ((char *)buf);

```

Serial

```
27 void loop() { //153 gönder
28   if (digitalRead(button1)) { //162 gönder
29     driver.send((uint8_t*)b1, strlen(b1));
30     driver.waitPacketSend();
31     mesaj = "buton1";
32     delay(300); // Buton sıfırlaması için bekle
33   }
34   if (digitalRead(button2)) { //160 gönder
35     driver.send((uint8_t*)b2, strlen(b2));
36     driver.waitPacketSend();
37     mesaj = "buton2";
38     delay(300); // Buton sıfırlaması için bekle
39   }
40   if (digitalRead(button3)) { //167 gönder
41     driver.send((uint8_t*)b3, strlen(b3));
42     driver.waitPacketSend();
43     mesaj = "buton3";
44     delay(300); // Buton sıfırlaması için bekle
45   }
46   if (digitalRead(button4)) { //174 gönder
47     driver.send((uint8_t*)b4, strlen(b4));
48     driver.waitPacketSend();
49     mesaj = "buton4";
50     delay(300); // Buton sıfırlaması için bekle
51   }
52   if (digitalRead(button5)) { //180 gönder
53     driver.send((uint8_t*)b5, strlen(b5));
54     driver.waitPacketSend();
55     mesaj = "buton5";
56     delay(300); // Buton sıfırlaması için bekle
57   }
58   Serial.print("Gönderilen mesaj: ");
59   Serial.println(mesaj);
60 }
```

Aşağıdaki program ile arduino ile RF alıcı (Rx) modülü kullanarak 433MHz frekans üzerinden haberleşme sağlanmıştır. 4 tane farklı Led diyot yakılıp söndürülmüştür. Bu led diyotlar sinyal gelmedikçe konum değiştirmezler. Alıcı modülün Data ucu Arduino uno'nun D11 pinine bağlanmalıdır.

```

1 #include <Wire.h>
2 #include <RF_ASK.h>
3 RF_ASK driver;
4 String rf_gelen = "";
5
6 int led1 = 2; // sol butonu
7 int led2 = 3; // ileri butonu
8 int led3 = 4; // sağ butonu
9 int led4 = 5; // geri butonu
10 int led5 = 6; // dur butonu
11
12 void setup() {
13   Serial.begin(9600);
14   driver.init();
15   if (driver.init()) Serial.println("RF433 hatasız");
16   pinMode(led1, OUTPUT);
17   pinMode(led2, OUTPUT);
18   pinMode(led3, OUTPUT);
19   pinMode(led4, OUTPUT);
20   pinMode(led5, OUTPUT);
21 }
22 void loop() {
23   uint8_t buf[12];
24   uint8_t buflen = sizeof(buf);
25   if (driver.recv(buf, &buflen)) {
26     Serial.println("Gelen mesaj:");
27     Serial.println((char*)buf);
28     rf_gelen = ((char*)buf);

```

29 }

30 if (rf-gelen.indexOf("buton1") >= 0)

31 digitalWrite(led1, !digitalRead(led1));

32

33 if (rf-gelen.indexOf("buton2") >= 0)

34 digitalWrite(led2, !digitalRead(led2));

35

36 if (rf-gelen.indexOf("buton3") >= 0)

37 digitalWrite(led3, !digitalRead(led3));

38

39 if (rf-gelen.indexOf("buton4") >= 0)

40 digitalWrite(led4, !digitalRead(led4));

41

42 if (rf-gelen.indexOf("buton5") >= 0)

43 digitalWrite(led5, !digitalRead(led5));

44

45 rf-gelen = "";

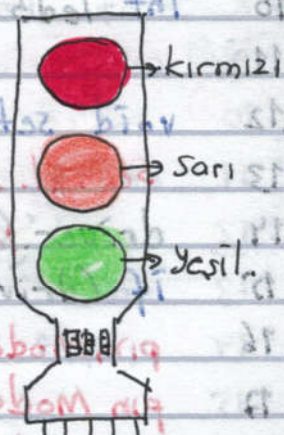
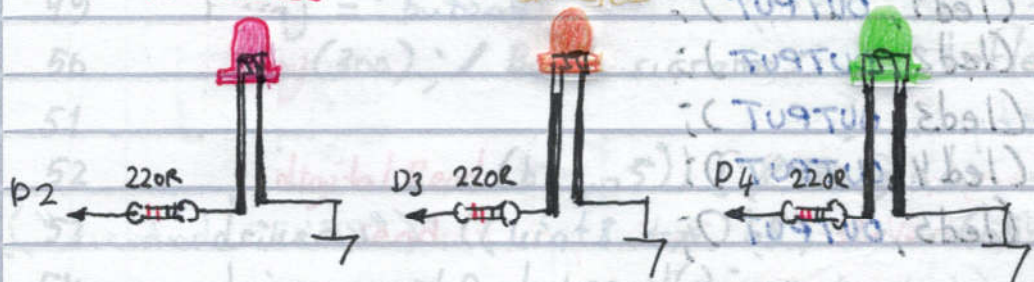
46

47 }

Trafik ışığı lamba bağlantısı:

kırmızı-led

Sarı-led



```

1  const int kirmizi_led = 2; // kirmizi_led 2.pine bağı
2  const int sarı_led = 3; // sarı_led 3.pine bağı
3  const int yesil_led = 4; // yesil_led 4.pine bağı
4  int Led-4 = 1; // Led-4 D11 " "
5  void setup() { // Led-5 D12 " "
6      pinMode(kirmizi_led, OUTPUT); // Ledleri çıkış olarak tanımla
7      pinMode(sarı_led, OUTPUT);
8      pinMode(yesil_led, OUTPUT); // Led-1 çıkış yapıldı
9  } // Led-2 çıkış yapıldı, kirmizi ledi yakarak 2. ledi yakıyor
10 // Led-3 çıkış yapıldı
11 void loop() { // Led-4 " "
12     digitalWrite(kirmizi_led, 1); // kirmizi ledi yakar
13     delay(9000); // 9 sn bekle
14     digitalWrite(sarı_led, 1); // sarı ledi yakar.
15     delay(2000); // 2 sn bekle
16     digitalWrite(kirmizi_led, 0); // kirmizi ledi söndür.
17     digitalWrite(sarı_led, 0); // sarı ledi söndür.
18     digitalWrite(yesil_led, 1); // yesil ledi yakar.
19     delay(7000); // 7 sn bekle
20     digitalWrite(sarı_led, 1); // sarı ledi yakar.
21     digitalWrite(yesil_led, 0); // yesil ledi söndür.
22     delay(3000); // 3 sn bekle
23     digitalWrite(sarı_led, 0); // sarı ledi söndür.
24 } // Led-5 " "
    digitalWrite(Led-4, 1); // Led-4'i yakar.
    delay(250); // 250 ms bekle
    digitalWrite(Led-4, 0); // Led-4'i söndür.
    digitalWrite(Led-5, 1); // Led-5'i yakar.
    delay(250); // 250 ms bekle
    digitalWrite(Led-5, 0); // Led-5'i söndür.
    delay(500); // 500 ms bekle
}

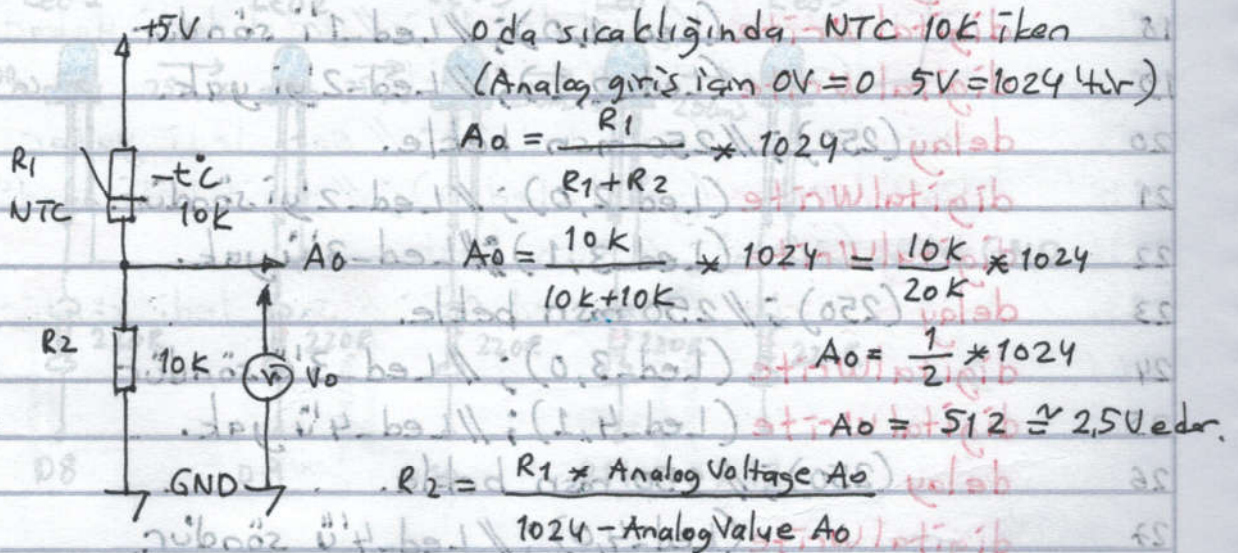
```


5 led ile yürüyen ışık

Yuryen.Ino

```
1 int Led-1=8; // Led-1 D8 pinine bağlı.
2 int Led-2=9; // Led-2 D9 " "
3 int Led-3=10; // Led-3 D10 " "
4 int Led-4=11; // Led-4 D11 " "
5 int Led-5=12; // Led-5 D12 " "
6 Serial.begin(9600);
7 void setup() {
8   pinMode(Led-1, OUTPUT); // Led-1 çıkış yapıldı.
9   pinMode(Led-2, OUTPUT); // Led-2 " "
10  pinMode(Led-3, OUTPUT); // Led-3 " "
11  pinMode(Led-4, OUTPUT); // Led-4 " "
12  pinMode(Led-5, OUTPUT); // Led-5 " "
13 }
14
15 void loop() {
16   digitalWrite(Led-1, 1); // Led-1'i yak.
17   delay(250); // 250 msn bekle.
18   digitalWrite(Led-1, 0); // Led-1'i söndür.
19   digitalWrite(Led-2, 1); // Led-2'yi yak.
20   delay(250); // 250 msn bekle.
21   digitalWrite(Led-2, 0); // Led-2'yi söndür.
22   digitalWrite(Led-3, 1); // Led-3'ü yak.
23   delay(250); // 250 msn bekle.
24   digitalWrite(Led-3, 0); // Led-3'ü söndür.
25   digitalWrite(Led-4, 1); // Led-4'ü yak.
26   delay(250); // 250 msn bekle.
27   digitalWrite(Led-4, 0); // Led-4'ü söndür.
28   digitalWrite(Led-5, 1); // Led-5'i yak.
29   delay(250); // 250 msn bekle.
30   digitalWrite(Led-5, 0); // Led-5'i söndür.
31   delay(500); // 500 msn bekle.
32
33 }
```

10K NTC ile sıcaklık hesaplama Ao pin bağlantısı.



NTC ısıtıldığında direnci düşer.

Yeni değer R olsun. $V_0 = R / (R + 10K) \times V_{CC} (5V)$

Örneğin $R = 8K$ olsun $V_0 = 8 / (8 + 10k) \times 5V$

$$V_0 = 8 / 18 \times 5V = (0,444) \times 5 = 2,22V \text{ eder.}$$

Yani direnç küçüldükçe gerilim düşümünde azalır.

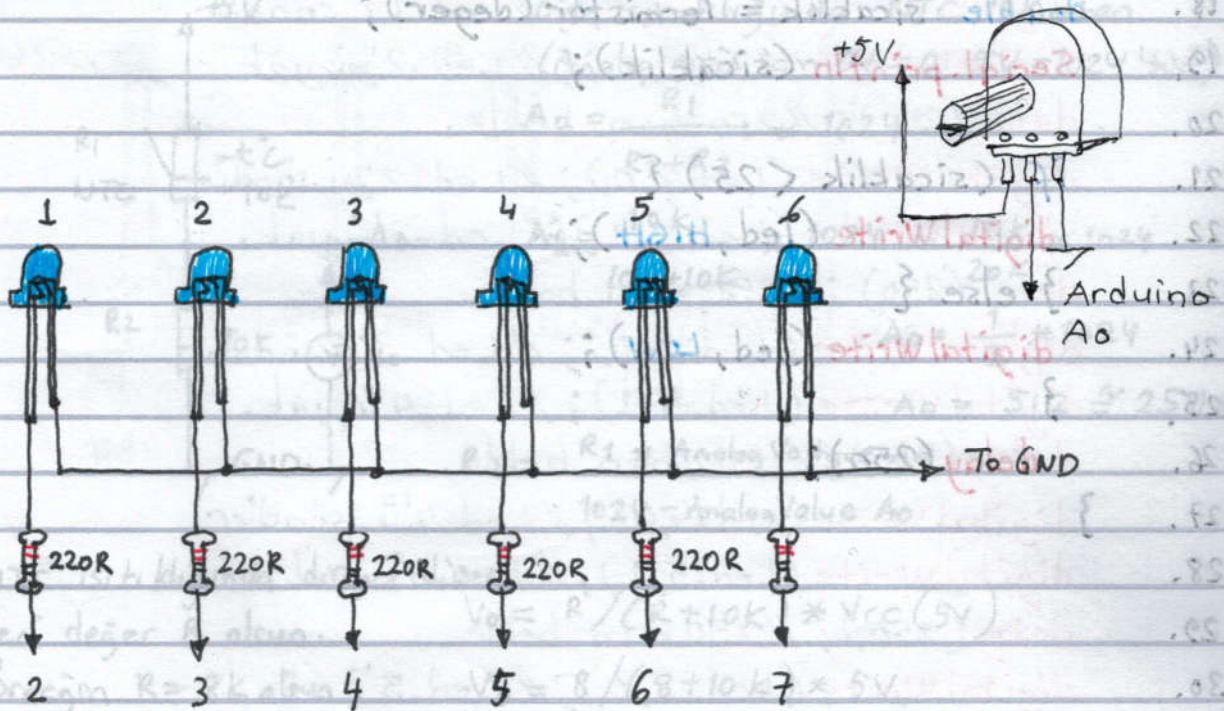
NTC 10K ile Sıcaklık hesaplama NTC.ino

```
1. #include <math.h> // Matematik formülleri eklendi.
2. #define led 2
3.
4. void setup() {
5.     pinMode(led, OUTPUT);
6.     Serial.begin(9600);
7. }
8. double Termistor(int analogOkuma) {
9.     double sıcaklik;
10.    sıcaklik = log(((10240000/analogOkuma) - 10000));
11.    sıcaklik = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 *
12.        sıcaklik * sıcaklik))) * sıcaklik);
13.    sıcaklik = sıcaklik - 273,15 ;
14.    return sıcaklik ;
15. }
16. void loop() {
17.     int deger = analogRead(A0);
18.     double sıcaklik = Termistor(deger);
19.     Serial.println(sıcaklik);
20.
21.     if (sıcaklik < 25) {
22.         digitalWrite(led, HIGH);
23.     } else {
24.         digitalWrite(led, LOW);
25.     }
26.     delay(250);
27. }
28.
29.
30.
31.
32.
33.
34.
```

NTC 10K ile sıcaklık hesaplaması

```
#include <math.h> // matematik fonksiyonları ekleniyor
#define led 2
void setup() {
  pinMode(led, OUTPUT);
  Serial.begin(9600);
}
double Termiztor(int analogOkuma) {
  double sıcaklik;
  sıcaklik = log(((1024 - analogOkuma) / 1024));
  sıcaklik = 1 / (0.00112618 + (0.000284765 * (0.000000854144 *
    sıcaklik * sıcaklik) * sıcaklik));
  sıcaklik = sıcaklik - 273.15;
  return sıcaklik;
}
```

6 Led POT'a bağlı hız ayarlı yürüyen ışık (pin bağlantısı)

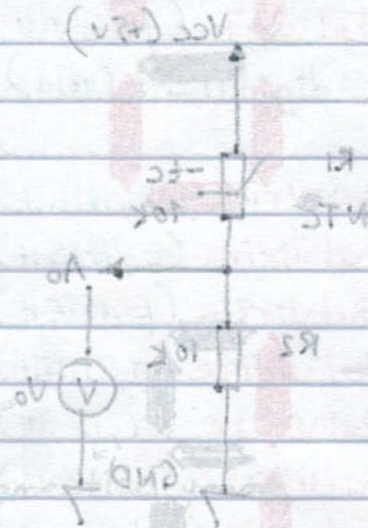


(6 LED) POT 'a bağılı Hız ayarlı Yürüyentedi 8 led.ino

```

1  int sensorValue = 0;
2
3  void setup()
4  {
5      pinMode(2, OUTPUT);
6      pinMode(3, OUTPUT);
7      pinMode(4, OUTPUT);
8      pinMode(5, OUTPUT);
9      pinMode(6, OUTPUT);
10     pinMode(7, OUTPUT);
11 }
12
13 void loop()
14 {
15     digitalWrite(2, HIGH);
16     sensorValue = analogRead(0);
17     delay(sensorValue + 25);
18     digitalWrite(2, LOW);
19     delay(15);
20     digitalWrite(3, HIGH);
21     sensorValue = analogRead(0);
22     delay(sensorValue + 25);
23     digitalWrite(3, LOW);
24     delay(15);
25     digitalWrite(4, HIGH);
26     sensorValue = analogRead(0);
27     delay(sensorValue + 25);
28     digitalWrite(4, LOW);
29     delay(15);
30     digitalWrite(5, HIGH);
31     sensorValue = analogRead(0);
32     delay(sensorValue + 25);
33     digitalWrite(5, LOW);
34     delay(15);

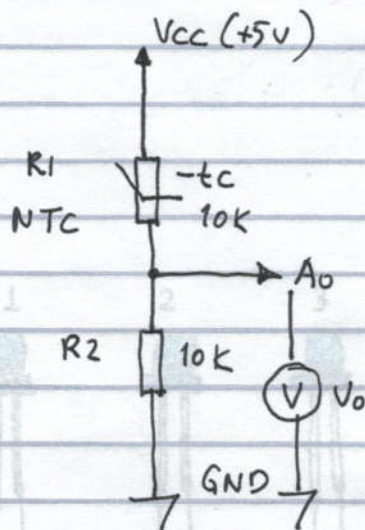
```



```

35 digitalWrite(6, HIGH);
36 sensorValue = analogRead(0);
37 delay(sensorValue + 25);
38 digitalWrite(6, LOW);
39 delay(15);
40 digitalWrite(7, HIGH);
41 sensorValue = analogRead(0);
42 delay(sensorValue + 25);
43 digitalWrite(7, LOW);
44 delay(15);
45 }

```



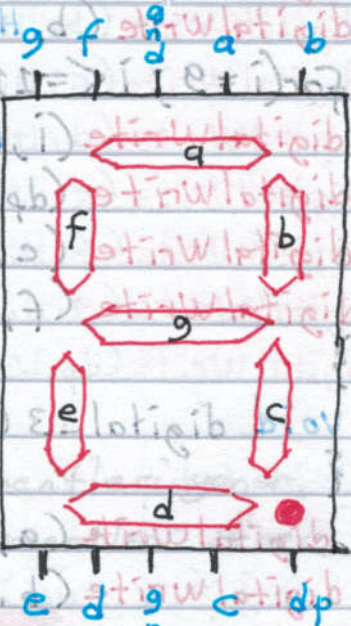
$$A_0 = \frac{R_1}{R_1 + R_2} \times 1024$$

$$R_2 = \frac{R_1 \times \text{AnalogVoltage } A_0}{1024 - \text{AnalogVoltage } A_0}$$

$$V_0 = \frac{R}{R + 1k} \times V_{CC}(5V)$$

7 Segment 0-9 Sayıcı.

7_SEGMENT_OK.ino



```

1. // pin kontrol hepsi Digital I/O
2. int a = 7; // A segmenti
3. int b = 6; // B segmenti
4. int c = 5; // C segmenti
5. int d = 10; // D segmenti
6. int e = 11; // E segmenti
7. int f = 8; // F segmenti
8. int g = 9; // G segmenti
9. int dp = 4; // DP segmenti
10.
11. void digital_0(void)
12. {
13.     unsigned char i; // a, b, c, d, e ve f segmentleri yanar.
14.     digitalWrite(a, HIGH);
15.     digitalWrite(b, HIGH);
16.     digitalWrite(c, HIGH);
17.     digitalWrite(d, HIGH); // 0
18.     digitalWrite(e, HIGH);
19.     digitalWrite(f, HIGH);
20.     digitalWrite(g, LOW);
21.     digitalWrite(dp, HIGH);
22. }
23. void digital_1(void)
24. {
25.     unsigned char i; // b ve c segmentleri yanar.
26.     digitalWrite(b, HIGH);
27.     digitalWrite(c, HIGH);
28.     for(i = 7; i <= 11; i++)
29.         digitalWrite(i, LOW); // sırasıyla a, f, g, d ve e segmentleri söner.
30.     digitalWrite(dp, LOW); // dp segmenti söner.
31. }
32. void digital_2(void) // 1
33. {
34.     unsigned int i; // a, b, g, e ve d segmentleri yanar

```

```

35. digitalWrite(a, HIGH); // a, b, g, e ve d pinleri yanar
36. digitalWrite(b, HIGH);
37. for(i=9; i<=11; i++) // c, f ve dp segmentleri söner.
38. digitalWrite(i, HIGH);
39. digitalWrite(dp, LOW);
40. digitalWrite(c, LOW); // 2
41. digitalWrite(f, LOW);
42. }
43. void digital-3(void)
44. { // a, b, c, d, e ve g segmentleri yanar
45. digitalWrite(a, HIGH); // dp, e ve f segmentleri söner.
46. digitalWrite(b, HIGH);
47. digitalWrite(c, HIGH);
48. digitalWrite(d, HIGH); // 3
49. digitalWrite(e, LOW);
50. digitalWrite(g, HIGH);
51. digitalWrite(dp, LOW);
52. digitalWrite(f, LOW);
53. }
54. void digital-4(void) // b, c, f ve g segmentleri yanar.
55. { // a, e, d ve dp segmentleri söner.
56. digitalWrite(b, HIGH);
57. digitalWrite(c, HIGH);
58. digitalWrite(f, HIGH);
59. digitalWrite(g, HIGH); // 4
60. digitalWrite(a, LOW);
61. digitalWrite(e, LOW);
62. digitalWrite(d, LOW);
63. digitalWrite(dp, LOW);
64. }
65. void digital-5(void) // a, d, c, f ve g segmentleri yanar.
66. { // b, e ve dp segmentleri söner.
67. unsigned char i;
68. digitalWrite(a, HIGH);
69. digitalWrite(c, HIGH); // 5
70. digitalWrite(d, HIGH);

```

```

102. digitalWrite (f, HIGH); // a, f, g ve dp segmentleri yanar
103. digitalWrite (g, HIGH); // b, c ve dp segmentleri yanar
104. digitalWrite (b, LOW); // e ve dp segmentleri yanar
105. digitalWrite (c, LOW); // a, f, g ve dp segmentleri yanar
106. digitalWrite (dp, LOW); // a, b, c, d, e, f ve g segmentleri yanar
107. }

void digital-6 (void)
{
    // a, f, g, d ve dp segmentleri yanar
    unsigned char i; // b, c ve dp segmentleri yanar
    for (i=7; i<=11; i++)
    {
        digitalWrite (i, HIGH); // a, f, g ve dp
        digitalWrite (c, HIGH); // b, c ve dp
        digitalWrite (dp, LOW); // 6
        digitalWrite (b, LOW);
    }
}

void digital-7 (void)
{
    unsigned char i;
    for (i=5; i<=7; i++) // a, b, c segmentleri yanar
    {
        digitalWrite (i, HIGH);
        digitalWrite (dp, LOW); // dp segmenti yanar
    }
    for (i=8; i<=11; i++) // e, d, f ve g
    {
        digitalWrite (i, LOW); // segmentleri
    }
    // sönür
}

void digital-8 (void) // 7
{
    unsigned char i;
    for (i=5; i<=11; i++) // a, b, c, d, e, f ve g segmentleri yanar
    {
        digitalWrite (i, HIGH);
        digitalWrite (dp, LOW);
    }
}

void digital-9 (void) // 8
{
    unsigned char i;

```

```

105. digitalWrite(a, HIGH); // a, b, c, d, f, ve g segmentleri yazar
106. digitalWrite(b, HIGH); // e ve dp segmentleri söner
107. digitalWrite(c, HIGH);
108. digitalWrite(d, HIGH);
109. digitalWrite(e, LOW); // g
110. digitalWrite(f, HIGH);
111. digitalWrite(g, HIGH);
112. digitalWrite(dp, LOW);
113. }
114. void setup()
115. {
116.   int j; // j değişkeni tanımlandı
117.   for (j=4; j<=11; j++)
118.     pinMode(j, OUTPUT); // 4'den 11'e kadar pinler ÇIKIŞ oldu
119. }
120. void loop()
121. {
122.   while (1) // sürekli döngü
123.   {
124.     digital_0(); // 0 yazar
125.     delay(1000); // 1 sn bekler
126.     digital_1(); // 1 yazar
127.     delay(1000); // 1 sn bekler
128.     digital_2(); // 2 yazar
129.     delay(1000); // 1 sn bekler
130.     digital_3(); // 3 yazar
131.     delay(1000); // 1 sn bekler
132.     digital_4(); // 4 yazar
133.     delay(1000); // 1 sn bekler
134.     digital_5(); // 5 yazar
135.     delay(1000); // 1 sn bekler
136.     digital_6(); // 6 yazar
137.     delay(1000); // 1 sn bekler
138.     digital_7(); // 7 yazar
139.     delay(1000); // 1 sn bekler
140.     digital_8(); // 8 yazar

```



141. **delay**(1000); // 1 sn bekle

142. **digitalWrite**(9, 0); // 9 yazar

143. **delay**(1000); // 1 sn bekle

144. } // while (1) döngüsü

145. } // loop döngüsü.

6 // DP GFEDCBA 0 1 2 3 4 5 6 7 8 9 DP GFEDCBA Ortak katot

7 // 0; 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 = 0x3F

8 // 1; 1 1 1 1 1 0 0 1 0 0 0 0 0 0 1 0 1 0 = 0x06

9 // 2; 1 0 1 0 0 1 0 0 1 0 1 0 1 0 1 1 0 1 1 = 0x5B

10 // 3; 1 0 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 = 0x4F

11 // 4; 1 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 1 0 = 0x66

12 // 5; 1 0 0 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 = 0x6D

13 // 6; 1 0 0 0 0 0 1 0 0 1 1 1 0 1 1 0 1 1 = 0x7D

14 // 7; 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 = 0x07

15 // 8; 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 = 0x7F

16 // 9; 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 = 0x6F

17 **unsigned char** table[] = { 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F };

18 { 0xC0, 0xF9, 0xA4, 0x80, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90 };

19 **void** setup() { // 1. pin → Serial Data (TX) 2. pin → Serial RX (RX) 3. pin → 5V 4. pin → GND

20 **Serial**.begin(115200); // Serial hızı 115200 baud hızında başlat

21 **pinMode**(latch, OUTPUT); // latch pin'i çıkış yapıldı.

22 **pinMode**(cs, OUTPUT); // chip select pin'i çıkış yapıldı.

23 **pinMode**(12, INPUT); // 12. pin → 5V

24 **for** (int i = 0; i < 10; i++) { // 10. pin → 5V

25 **pinMode**(13, INPUT); // 13. pin → 5V

26 } // 13. pin → 5V

27 **pinMode**(14, INPUT); // 14. pin → 5V

28 **pinMode**(15, INPUT); // 15. pin → 5V

29 **pinMode**(16, INPUT); // 16. pin → 5V

30 **pinMode**(17, INPUT); // 17. pin → 5V

31 **pinMode**(18, INPUT); // 18. pin → 5V

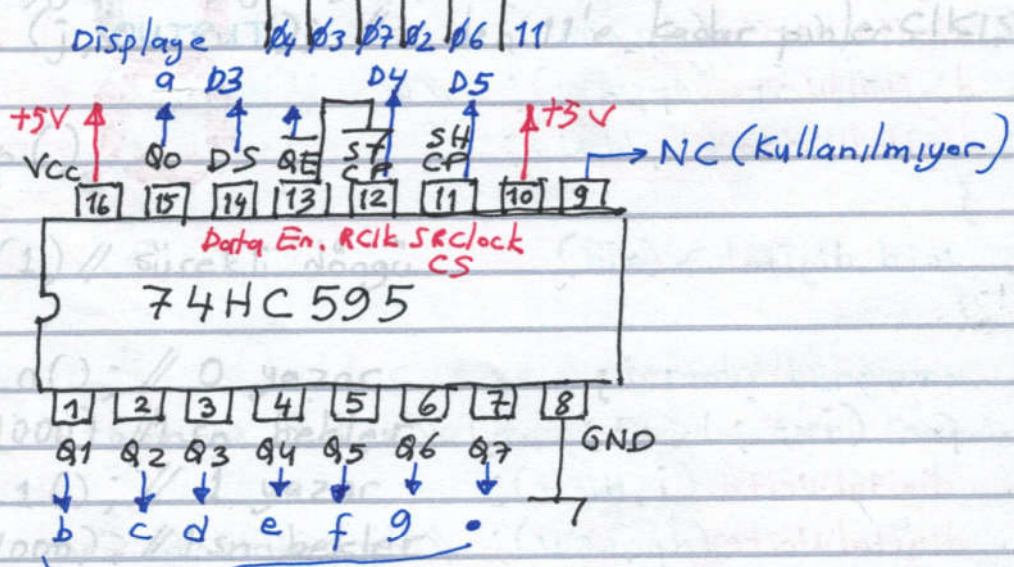
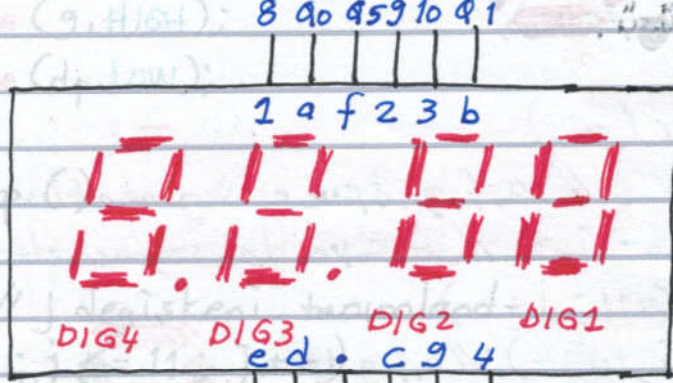
32 **Serial**.println(i); // 0'dan 9'a kadar seri iletiliyor

33 }

34 **delay**(500); // 500 ms bekle

105. digitalWrite(a, HIGH); // a, b, c, d, f, ve 9 segmentleri yandır
 106. digitalWrite(b, HIGH); // e ve dp segmentleri söndür
 107. digitalWrite(c, HIGH); // added a21 // : (0001) 11111111
 108. digitalWrite(d, HIGH); // added c // : (0001) 11111111
 109. **4 Digit 7 Segment Display (Ortak Anot)**

Arduino pinleri D3=DS, D4=STCP, D5=SH_CP



Q0 - Q7 : 15-1, 2, 3, 4, 5, 6 ve 7. pinler : Displayın a, b, c, d, e, f ve g'ne

DS : 14. pin → Serial Data Girişi

SH_CP : 11. pin → Shift Register (Clock) Saat girişi

ST_CP : 12. pin → Latch (Kilitleme)

$\overline{OE}$ 13. pin → Çıkış Enable etkinleştirme (Gnd)'ye bağlanır.

Kullanımı : Veri girişi yapılmak istendiğinde latch (ST_CP)

12. pini lojik 0 (0Volt) seviyesine çekilir. Data girişi 14. pin-

den yapılır ve Clock pulsı SH_CP 11. pine uygulanır, Sonra

bilgi kaydırılmış olur. Tekrar 12. pini lojik 1 (+5V) seviyesine

çıkartılır.

```

1. #define NUM_OFF_DIGITS 4
2. int latch = 4; // 74HC595 pin 12 ST_CP
3. int cs = 5; // 74HC595 pin 10 SH_CP
4. int data = 3; // 74HC595 pin 8 DS
5. int digitPins[4] = {11, 10, 9, 8}; // digitPins 4 elemanlı dizi.
6. // DP GFEDCBA Ortak Anot DPGFEDCBA Ortak Katot
7. // 0; 11000000 = 0xC0 0; 00111111 = 0x3F
8. // 1; 11111001 = 0xF9 1; 00000110 = 0x06
9. // 2; 10100100 = 0xA4 2; 01011011 = 0x5B
10. // 3; 10110000 = 0xB0 3; 01001111 = 0x4F
11. // 4; 10011001 = 0x99 4; 01100110 = 0x66
12. // 5; 10010010 = 0x92 5; 01101101 = 0x6D
13. // 6; 10000010 = 0x82 6; 01111101 = 0x7D
14. // 7; 11111000 = 0xF8 7; 00000111 = 0x07
15. // 8; 10000000 = 0x80 8; 01111111 = 0x7F
16. // 9; 10010000 = 0x90 9; 01101111 = 0x6F
17. unsigned char table[] =
18. {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90};
19. void setup() { // Bu kodlarda 0 led'i yakar, 1 ise led'i söndürür.
20.   Serial.begin(115200); // Seri portu 115200 baud hızında başlat.
21.   pinMode(latch, OUTPUT); // latch pin'i çıkış seçildi.
22.   pinMode(cs, OUTPUT); // clock pin'i cs çıkış seçildi.
23.   pinMode(data, OUTPUT); // data pin'i ds çıkış seçildi.
24.   for (int j = 0; j < NUM_OFF_DIGITS; j++)
25.     pinMode(digitPins[j], OUTPUT); // digit seçme pinleri çıkış
26. }
27. void loop() { for (int i = 0; i < NUM_OFF_DIGITS; i++) {
28.   // 0'dan 9'a kadar segmentleri say ve 500 ms seri ekranda göster.
29.   for (int j = 0; j < 10; j++) {
30.     Display(i, j);
31.     delay(500); // 500 msn bekle
32.     Serial.println(j); // 0'dan 9'a kadar seri ekrana yazdır.
33.   }
34.   delay(500); // 500 msn bekle

```



```
35. }
36. {
37. void Display (int id, unsigned char num)
38. { // Display için 74HC595 shift register entegrasyonu hazırlanıyor.
39.   digitalWrite (latch, LOW);
40.   shiftOut (data, cs, MSBFIRST, table[num]);
41.   digitalWrite (latch, HIGH); // Kaydırmadan sonra latch high yapılır.
42.   for (int j = 0; j < NUM_OFF_DIGITS; j++) digitalWrite (digitPins[j], LOW);
43.   digitalWrite (digitPins[id], HIGH);
44. }
```

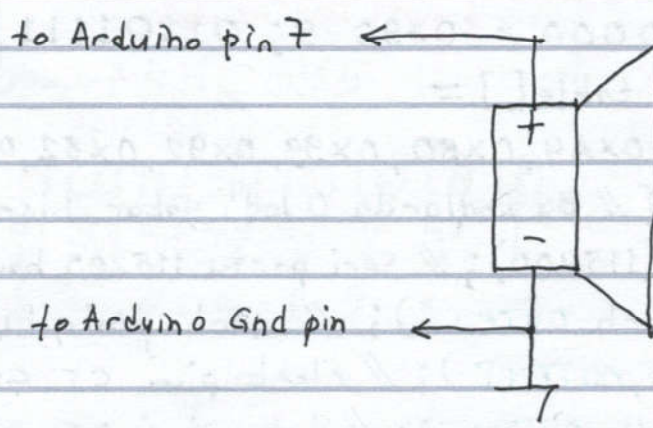
*g1 = 830 ; a2 = 880 ; a1 = 932 ;
a = 988 ; c2 = 1046 ; c1 = 1108 ; d2 = 1174 ; d1 = 1244 ; e2 = 1318 ; f2 = 1396 ;
f1 = 1480 ; g2 = 1568 ; g1 = 1660 ; a3 = 1760 ;*

tone (buzzer, a); a notasına karşılık gelen 440 frekansı çıkar.
tone (buzzer, c); // Korkma Sürmez bu sifak
delay (800); // do
tone (buzzer, f); // fa → f işi okunur A ot
delay (800);
tone (buzzer, g); // sol #
delay (800);
tone (buzzer, e); // mi → mi işi okunur A ot
delay (400);
tone (buzzer, g); // sol
delay (200);
tone (buzzer, f); // fa
delay (1600);
int a = (buzzer);
delay (300);
// larda 4'üzün a' sence
tone (buzzer, f); // fa
delay (800);
tone (buzzer, a2); // la işi
delay (800);

```

35. }
36. }
37. void Display (int id, unsigned char num)
38. { // Display in 54HC595 shift register integrated peripheral
39. digitalWrite (dataPin, LOW); // 54HC595 pin 1 to GND
40. shiftOut (dataPin, 2, MSBFIRST, num); // 54HC595 pin 2 to GND
41. digitalWrite (latchPin, HIGH); // 54HC595 pin 12 to HIGH
42. for (int j = 0; j < NUM_OF_DIGITS; j++) digitalWrite (digitalPin[j], num[j]);
43. digitalWrite (digitalPin[19], HIGH);
44. }

```



İstiklal marşı notaları (Performans 1)

mars.ino

```

1. const int buzzer = 7; // Buzzer pini
2. int a, ad, b, c, cd, d, dd, e, f, fd, g, gd, a2, ad2, b2, c2, cd2,
3. d2, dd2, e2, f2, fd2, g2, gd2, a3;
4. delay(800);
5. void setup()
6. {
7.   pinMode(buzzer, OUTPUT);
8.   a = 440; ad = 466; b = 494; c = 523; cd = 554; d = 587; dd = 622;
9.   e = 659; f = 698; fd = 740; g = 784; gd = 830; a2 = 880; ad2 = 932;
10.  b2 = 988; c2 = 1046; cd2 = 1108; d2 = 1174; dd2 = 1244; e2 = 1318; f2 = 1396;
11.  fd2 = 1480; g2 = 1568; gd2 = 1660; a3 = 1760;
12.  tone(buzzer, c2); // do##
13. //tone(buzzer, a); a notasına karşılık gelen 440 tonunu çalar;
14. tone(buzzer, c); // Korkma Sönmez bu şafak
15. delay(800); //do
16. tone(buzzer, f); //fa
17. delay(800);
18. tone(buzzer, g); //sol
19. delay(800);
20. tone(buzzer, gd); //sol#
21. delay(800);
22. tone(buzzer, e); //mi
23. delay(400);
24. tone(buzzer, g); //sol#
25. delay(200);
26. tone(buzzer, f); //fa
27. delay(1600);
28. notone(buzzer); //mi
29. delay(300);
30. // larda yüzen al sancak
31. tone(buzzer, f); //fa
32. delay(800);
33. tone(buzzer, ad2); //la##
34. delay(800);

```