

# Uygulamalı Projeler İle Arduinio Eğitimi-Robotistan

2 Ekim 2020 Cuma 19:30

- [https://www.robotistan.com/arduino-proje-seti?utm\\_source=youtube&utm\\_medium=aciklama](https://www.robotistan.com/arduino-proje-seti?utm_source=youtube&utm_medium=aciklama)

- ✓ 01 Arduino Nedir? Nasıl Kurulur ve Neler Yapılabilir ?
- ✓ 02 Arduino İle LED Yakma Blink Uygulaması
- ✓ 03 Buton İle Led Yakma Blink Uygulaması
- ✓ 04 Arduino İle Analog Okuma ve Seri Haberleşme
- ✓ 05 Potansiyometre İle Led Yakma
- ✓ 06 Arduino İle Karaşımşek Uygulaması
- ✓ 07 LDR İle Otomatik Lamba Uygulaması
- ✓ 08 Arduino İle RGB LED Uygulaması
- ✓ 09 NTC İle Sıcaklık Ölçümü
- ✓ 10 LM35 ile Sıcaklık Ölçümü
- ✓ 11 Ultrasonik Sensör İle Park Sensörü Yapımı
- ✓ 12 Ses İle Motor Kontrolü
- ✓ 13 Joystick İle Servo Motor Kontrolü
- ✓ 14 IR Kumanda İle LED Kontrolü
- ✓ 15 Arduino İle Dijital Metre Yapımı
- ✓ 16 Hareket Sensörü (PIR) İle Servo Motor Kontrolü
- ✓ 17 Bluetooth İle RGB LED Kontrolü
- ✓ 18 Arduino İle Dijital Saat Yapımı
- ✓ 19 Arduino İle Toprak Nem Sensörü Kullanımı
- ✓ 20 Arduino İle Yağmur Sensörü Kullanımı
- ✓ 21 Arduino İle Gaz Sensörü Kullanımı
- ✓ 22 Arduino İle RFID Sensörü Kullanımı
- ✓ 23 ESP8266 İle Sıcaklık ve Nem Ölçümü
- ✓ 24 ESP8266 İle Step Motor Kontrolü

## 01 Arduino Nedir? Nasıl Kurulur ve Neler Yapılabilir ?

28 Aralık 2020 Pazartesi 20:41

### Arduino Uno Rev 3 (CH340 Klon)

Ayrıntılar:

<https://www.robotistan.com/arduino-uno-r3-klon-usb-kablo-hediyeli-usb-chip-ch340>



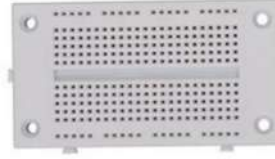
### A'dan B'ye USB Kablo

Standart USB 2.0 kablodur. Yazıcı kablosu olarak da bilinir. Arduino ve PIC Programlayıcılarla uyumludur.



### Breadboard

Bütün pinler arasındaki mesafe 0.1 inçtir (2.54mm). 20-29 AWG kalınlığındaki kablolar ile kullanılabilir.



### Erkek-Erkek ve Erkek-Dişi Jumper Kablo

Bu bağlantı kabloları 2,54 mm'lik standart pinlere göre dizayn edilmiştir. Bu 26 awg'lik bağlantı kablolarının boyu 20 cm'dir.



### 9V Pil



### 12V 500mA Adaptör

12V gerilimde maksimum 0.5A akım sağlayabilen DC adaptördür. 2.5mm barrel jack tipinde konnektöre sahiptir.



### 9v Pil Barrel Dönüştürücü Kablo

Basit bir kablo olarak görünse de bu ürün 9V'luk pilleri DC Adaptör çıkışına dönüştürür. Pozitif merkezli olup 10 cm uzunluğa sahiptir.



### 5V Röle Kartı

Ayrıntılar:[https://www.robotistan.com/1-way-5v-relay-module-tekli-5v-role-karti?language=tr&h=62226c45&gclid=Cj0KCQjw6ar4BRDnARIsAITGzIAy6h79y6hbM\\_jGgoiKU4kVpD5ZPIAFR94mDaVJJTweybbhLSv8iRgaAlKIEALw\\_wcB](https://www.robotistan.com/1-way-5v-relay-module-tekli-5v-role-karti?language=tr&h=62226c45&gclid=Cj0KCQjw6ar4BRDnARIsAITGzIAy6h79y6hbM_jGgoiKU4kVpD5ZPIAFR94mDaVJJTweybbhLSv8iRgaAlKIEALw_wcB)



## 01 Arduino Nedir? Nasıl Kurulur ve Neler Yapılabilir ?

### Arduino Nedir?

- Arduino, interaktif projeler geliştirmek için tasarlanan, elektronik donanım ve yazılım temelli bir geliştirme platformudur.
- Arduino, **wiring** tabanlı programlama dili ile programlanır ve **processing** tabanlı Arduino yazılım geliştirme ortamı olan Arduino IDE ile karta aktarılır.
- Yazılım tamamlandıktan sonra kod karta bir USB kablosu vasıtası ile kolayca aktarılır.

### Arduino ile Neler Yapılabilir?

- Arduino'nun analog ve dijital pinleri sayesinde analog ve dijital sinyalleri işlemek mümkündür. Arduino'ya bağlanan sensörler vasıtasıyla ortam etkileşimli giriş sinyalleri elde edip bu sinyalleri incelemek ve bu sinyaller ile algoritmalar kurarak çıkış sinyalleri üretmek mümkündür. Bu sayede çeşitli robotlar ve elektronik sistemler tasarlanabilir.

### Arduino Programın Bilgisayarımızda İlk Çalıştırılması

- Programda **void setup()** kısmına yazacağımız fonksiyonlar, kart ilk enerji alıp çalıştığında sadece bir kere çalışır. Kullanacağımız giriş/çıkış pinlerini, seri port konfigürasyonunu vb. ayarları bu kısımda yapıyoruz. **void loop()** kısmında ise, setup fonksiyonundaki komutlar çalıştıktan sonra kartın enerjisi kesilene kadar sürekli çalışacak olan fonksiyonları barındırır.

## 02 Arduino İle LED Yakma Blink Uygulaması

28 Aralık 2020 Pazartesi 21:03

### 330, 1K, 10K Ohm Direnç

0.25W değerine kadar enerjiye dayanıklı dirençlerdir.

Direnç değerini hesaplamak için;

<https://devreokulu.com/DirençHesaplama.html> |

Ohm çevirme işlemleri yapmak için;

<https://www.convertworld.com/tr/elektrik-direnci-direnc/ohm.html>

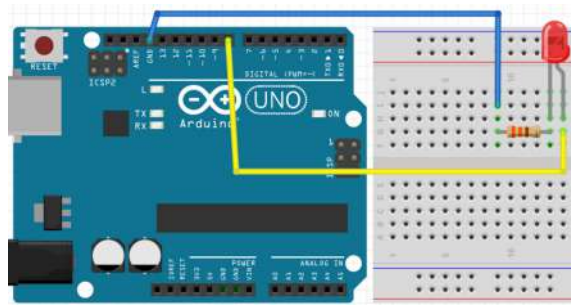
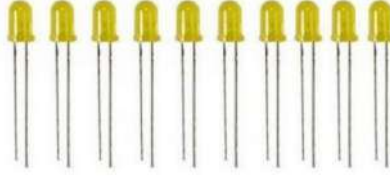
Ayrıntılar: <https://maker.robotistan.com/direnc/>



### 5mm Kırmızı, Yeşil, Sarı LED

1.5-3V arası gerilimde çalışır. 5V ve üstü voltaj değerleri için gerekli dirençlerle kullanılması gerekir.

Anot ve katot olmak üzere iki farklı bacağı vardır. Bunlardan anodu uzun olan bacağı pozitif gerilime yani + uca, katot kısa olan bacak ise negatif gerilime yani – uca ya da toprak hattına (GND, Ground) bağlanmalıdır.

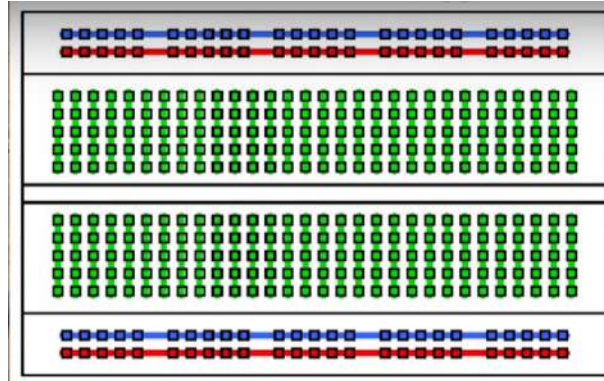


```
1 //define Led 8
2
3 void setup()
4 {
5   pinMode(8, OUTPUT);
6   //pinMode(Led, OUTPUT);
7 }
8
9 void loop()
10 {
11   digitalWrite(8, HIGH);
12   //digitalWrite(Led, HIGH);
13   delay(500);
14   digitalWrite(8, LOW);
15   //digitalWrite(Led, LOW);
16   delay(500);
17 }
```

### 02 Arduino İle LED Yakma Blink Uygulaması

#### Teorik Bilgi

- İşaretli olan kısımlar kendilerince kısa devredir.



- Arduino kartımız 5V gerilimle çalışmaktadır. LED'in üzerinden geçecek maksimum akımın 15 mA değerini geçmemesi gereklidir. Bunun için akım sınırlayıcı bir direnci LED'imize seri olarak bağlamamız gerekmektedir.

$$5V = 0,015A \times R \rightarrow R=333,33$$

#### Devre Kısmı

- Devrede LED'e seri olarak bir direnç bağlanır. Böylelikle LED üzerinden yüksek akım geçmesi ve LED'in zarar görmesi engellenir. LED'in (+) bacağı Arduino'nun 8.pinine bağlıdır. LED'in (-) bacağı dirence seri bağlanır, direncin diğer bacağına da Arduino'nun GND pinine bağlantı yapılmıştır.

#### Kod Kısmı

- Kullanacağımız pin çıkış veya giriş olarak belirlenmez ise programın devamında yazacağımız giriş veya çıkış fonksiyonları o pini kullanamaz.
- Loop fonksiyonu ile öncelikle 8 numaralı pine HIGH lojik seviyesine, yani 5V'a ayarlıyor, 500 milisaniye (1 saniyenin yarısına eşittir) hiçbir işlem yapmadan bekliyor ve bu sefer 8 numaralı pini lojik LOW yani 0V veya toprak hattı seviyesine ayarlıyor. Bu işlemi yaptıktan sonra mikokontrolcü, delay fonksiyonu sayesinde tekrardan 500 milisaniye hiçbir işlem yapmadan bekliyor.

## 03 Buton İle Led Yakma Blink Uygulaması

28 Aralık 2020 Pazartesi 21:03

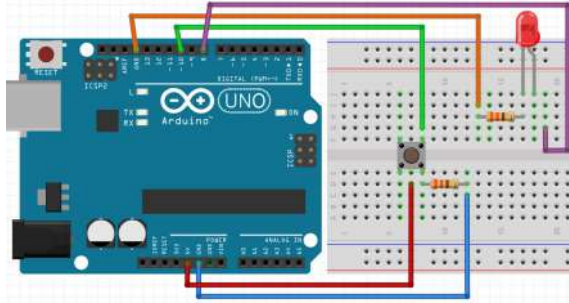
### 4 Pinli Push Buton

Push butonlar sadece basıldığında on olan bırakıldığında off olan devre elemanlarıdır.

Bir push buton iki anahtarın işini tek başına görebilir.

Pinlerin olduğu yüzlerin kısa devre olması sadece butonun basılmasına bağlıdır.

4 pinli push butonların ikiye bacakları paralel bağlıdır. Yani buton 4 pinli de olsa 2 pinli mantığı ile bağlanmaktadır Butona basıldığında 1 ve 3 nolu kırmızı işaretli pinler ile 2 ve 4 nolu mavi işaretli pinler birbirine paralel bağlıdır.

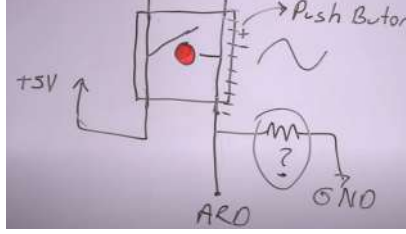


```
1 #define Buton 10
2 #define Led 8
3
4 int buton_durumu = 0;
5
6 void setup() {
7   pinMode(Buton, INPUT);
8   pinMode(Led, OUTPUT);
9 }
10
11 void loop() {
12   buton_durumu = digitalRead(Buton);
13   if(buton_durumu == 1) {
14     digitalWrite(Led, HIGH);
15   }
16   else{
17     digitalWrite(Led, LOW);
18   }
19 }
20
21 /*
22 #define Buton 10
23 #define Led 8
24
25 void setup() {
26   pinMode(Buton, INPUT);
27   pinMode(Led, OUTPUT);
28 }
29
30 void loop() {
31   if(digitalRead(Buton) == 1)
32     digitalWrite(Led, HIGH);
33   else
34     digitalWrite(Led, LOW);
35 }
36 */
37
```

### 03 Buton İle Led Yakma Blink Uygulaması

#### Devre Kısmı

- Butonu bağlarken 10kΩ direnç kullandık. Bu direncin ismi **pull-down direncidir**. Pull-down direnci, dijital pinleri giriş olarak kullandığımızda sinyalin bozulmamasını sağlar. Buton basılı değilken dijital pinden okunan değer 0V yani lojik LOW seviyesidir. Pull-down direnci, buton basılıp değer HIGH'a çekilmediği sürece bu pindeki gerilimin 0V'ta sabit kalmasını sağlar.



#### Kod Kısmı

- Giriş-çıkış ayarlarırken giriş yapmak istediğimiz butonlara "INPUT", çıkış yapmak istediğimiz pinlerde "OUTPUT" yazarız.

## 04 Arduino İle Analog Okuma ve Seri Haberleşme

28 Aralık 2020 Pazartesi 21:03

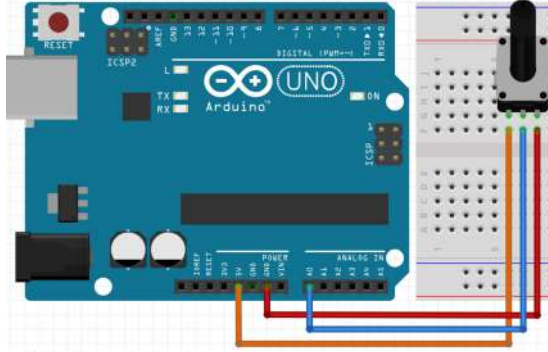
### 10K Potansiyometre

Potu çevirdikçe direnci değişir.

Ara direnç değerlerine ihtiyaç duyulduğu zaman veya Analogdan Dijitale Dönüştürücü (ADC - analog to digital converter) işlemlerinde kullanılabilir.

Potansiyometre 3 bacaklı devre elemanıdır. Orta bacak sinyali alacağımız kısımdır. Sağ ve sol bacaklar + ve - voltaj bağlayacağımız kısımdır. Dirençlerin voltaja göre yönü olmadığından sağ ve sol bacaklar için kaç volt olduğun bir önemi yoktur. Mikrokontrolcü uygulamalarında ise genellikle gerilim bölücü olarak kullanılır.

Ayrıntılar: <https://maker.robotistan.com/potansiyometre/>



```
1 #define potpin A0
2
3 int deger=0;
4
5 void setup() {
6   Serial.begin(9600);
7   Serial.println("Pot Değer Okuma");
8 }
9
10 void loop() {
11   deger = analogRead(potpin);
12   //float gerilim = (5.00/1024.00)*deger;
13   //Serial.println(gerilim);
14   Serial.println(deger);
15   delay(300);
16 }
```

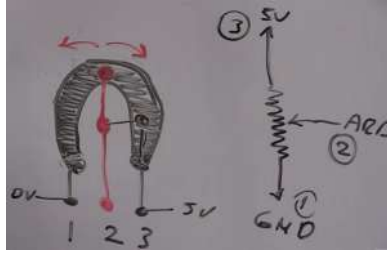
### 04 Arduino İle Analog Okuma ve Seri Haberleşme

#### Teorik Bilgi

- Arduino UNO kartımızdaki işlemcide, 10-bit çözünürlüğe sahip analogdan dijitale dönüştürücü (ADC – analog to digital converter) mevcuttur. Arduino mikrokontrolcüsü 5V gerilimle çalışmakta. Bu mikrokontrolcüde sahip olduğuna söylediğimiz 10-bit ADC, 0V ile 5V arası gerilimleri  $2^{10}=1024$  adım hassasiyet ile okuyabilir. Yani analog input pinlerinden birine vereceğimiz 0V gerilim bize 0 değerini; aynı şekilde 5V gerilim ise 1023 değerine denk düşüyor. Bu pine herhangi bir gerilim uygulamadığımız takdirde ADC'miz 0-5V arasında çalışacaktır.

#### Devre Kısım

- Potansiyometre, analog kontrol sunduğundan dolayı Arduino'da **Analog INPUT** olarak bağlanır. Böylelikle potansiyometreden gelen analog giriş sinyalleri ile çıkıştaki uyarıcılar kontrol edilebilir veya giriş sinyallerinin takibi yapılabilir. Potansiyometrenin orta bacağı **Data** olarak kullanılmıştır. Giriş sinyalleri bu baktan kontrol edilmektedir. Soldaki bacak **GND**, sağdaki bacak da **5V** pinine bağlanmıştır. Böylelikle potansiyometre sağa doğru çevrildikçe Arduino'ya giden veri artacak, sola doğru çevrildikçe veri azalacaktır. Sağdaki ve soldaki bacakların yerleri değiştirilseydi bu sefer sola çevrildikçe veri artıyor, sağa çevrildikçe veri azalıyor olurdu.

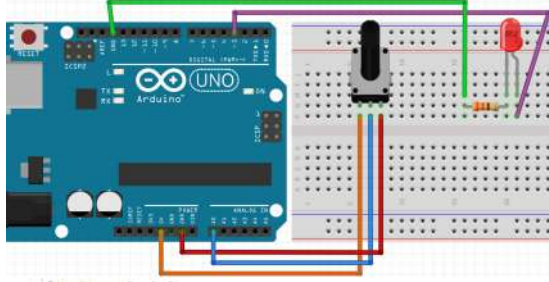


#### Kod Kısım

- "setup" kısmında önceki kod kısımlarında dijital giriş-çıkış kullandığımızdan dolayı, pinleri ne olarak kullanacağımıza göre ayarlıyorduk. Analog okuma yaparken giriş çıkış tanımlamamıza gerek yok bu sebepten dolayı bu kodda "pinMode" komutunu kullanmıyoruz.
- 6.satır ile 9600 Baund bir seri haberleşme başlatıyoruz. Arduino kodu çalışmaya başladığında ilk olarak bilgisayar ile haberleşmeyi başlatacaktır. Haberleşme başladıktan sonra 7.satırı okuyarak Pot Değer Okuma yazısı bilgisayarda seri monitöre yazdırılacaktır. Aynı şekilde 14.satırda okunan değeri yazdırır.
- 12.satırda 5V gerilim, 1024 bite bölünür ve potansiyometreden gelen değer ile çarpılır; değerın eşiti olan gerilim hesaplanır.

## 05 Potansiyometre İle Led Yakma

28 Aralık 2020 Pazartesi 21:03

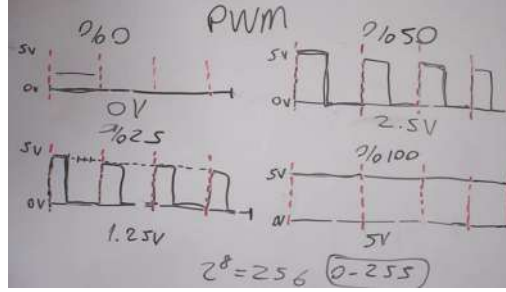


```
1 #define led 3
2 #define pot A0
3
4 void setup() {
5 }
6
7 void loop() {
8   int deger = analogRead(pot);
9   //deger=map(deger,0,1023,0,255);
10  deger=deger/4;
11  analogWrite(led, deger);
12 }
```

### 05 Potansiyometre İle Led Yakma

#### Devre Kısmı

- Bu uygulamaya kadar dijital giriş-çıkış ve analog girişi gördük. Bu uygulamayla analog çıkış yani PWM özelliğini öğreneceğiz.
- PWM (Pulse with Modulation), sinyal genişlik modülasyonun kısaltmasıdır. Bu özellik Arduino Uno üzerinde 6 pinde mevcut yaklaşık işaretleri bulunan 3,5,6,9,10 ve 11.pinlerdir.
- PWM özelliği ile istediğimiz voltajı vererek ledin parlaklığını ayarlayabiliriz. Biz bu özelliği genellikle robotlarda motor hızlarında kullanıyoruz.



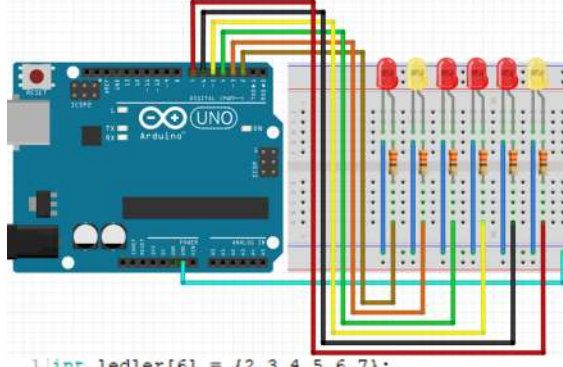
#### Kod Kısmı

- Dijital giriş-çıkış kullanmadığımızdan "setup" kısmı boş bırakıyoruz.
- Döngümüzde Pot'dan veriyi Led'e göndermek için "analogRead" komutu ile potansiyometreden veriyi okuyoruz. Okuduğumuz değeri "deger" değişkenine yazıyoruz. İkinci satırda ise "map" komutunu kullanarak 0 ile 1023 arasında gelen değeri 0 ile 255 arasında oranlıyoruz.
- Map kullanımında 4 parametre var. İlk olarak değişkenimiz yazıyoruz. Oranlanacak sayımızın min ve max değerleri yazılır sonra oranlanacak sayının min ve max değeri yazılarak kullanılır.
- Analog okumayı 10 bit ( $2^{10}=1024$ ) çözünürlükte yaparken, analog yazmayı 8 bit ( $2^8=256$ ) çözünürlükte yapabiliyoruz. Okuduğumuz veriyi, çıkışa göre oranlamamız gerekiyor.
- "map" komutu yerine direkt olarak 4'e bölebiliriz.
- Oranlama işleminden sonra "AnalogWrite" komutu ile PWM pinlerinden çıkış verebiliriz.



## 06 Arduino İle Karaşımşek Uygulaması

28 Aralık 2020 Pazartesi 21:09



```
1 int ledler[6] = {2,3,4,5,6,7};
2
3 void setup(){
4   for(int i=0; i<6; i++){
5     pinMode(ledler[i],OUTPUT);
6   }
7 }
8
9 void loop() {
10  for(int i=0; i<6; i++){
11    digitalWrite(ledler[i], HIGH);
12    delay(20);
13    digitalWrite(ledler[i], LOW);
14  }
15  for(int j=5; j>=0; j--){
16    digitalWrite(ledler[j], HIGH);
17    delay(20);
18    digitalWrite(ledler[j], LOW);
19  }
20 }
```

### 06 Arduino İle Karaşımşek Uygulaması

#### Devre Kısmı

- Diğer Led uygulamamızda direnci Led'in Katot yani (-) uca seri bağlamıştık. Bu uygulamada Anato yani (+) uca seri bağladık. Yani direnci Led'in hangi bacağına bağladığımızın bir önemi yok.

#### Kod Kısmı

- Dizi tanımlarken "ledler[]" ifasinin içerisinde dizinin elaman sayısını yazabiliriz.

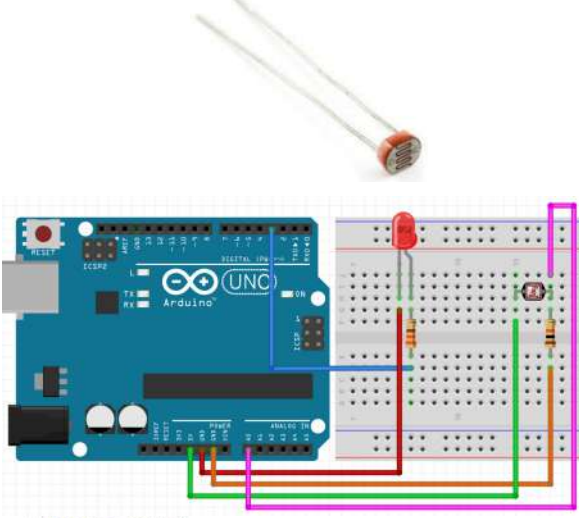
## 07 LDR İle Otomatik Lamba Uygulaması

28 Aralık 2020 Pazartesi 21:09

### 5mm LDR

Üzerine düşen ışığa bağlı olarak iki ucu arasındaki direnç değeri değişir.

Ayrıntılar: <https://maker.robotistan.com/ldr/>



```
1 #define led 3
2
3 void setup() {
4   pinMode(led, OUTPUT);
5   Serial.begin(9600);
6 }
7
8 void loop() {
9   int isik=analogRead(A0);
10  Serial.println(isik);
11  delay(50);
12  if(isik>580) {
13    digitalWrite(led, LOW);
14  }
15  if(isik<530) {
16    digitalWrite(led, HIGH);
17  }
18 }
```

### 07 LDR İle Otomatik Lamba Uygulaması

#### Devre Kısmı

- LDR'nin bir bacağından (-) hattına diğer bacağından (+) hattına gideriz. Direnci LDR'nin (-) hattına bağladık ve direncin diğer bacağından GND'ye bağladık. LDR'nin (+) bacağından 5V'a gittik. LDR ile direncin birbirleriyle ortak bacağından Analog Giriş olan A0'gideriz.



## 08 Arduino İle RGB LED Uygulaması

28 Aralık 2020 Pazartesi 21:09

### 5mm RGB LED

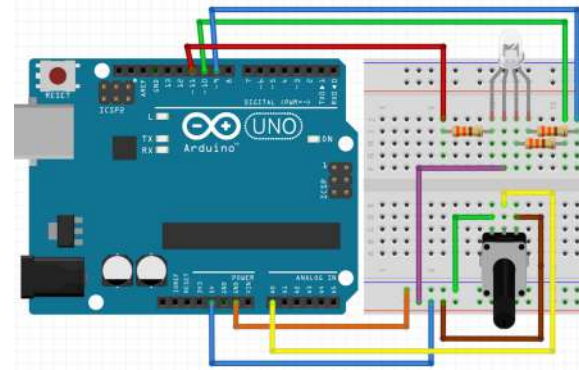
İçerisinde Kırmızı, Yeşil ve Mavi olmak üzere üç farklı renkte LED barındırmaktadır.

Uzun ucu artı uçtur, uzun uca artı voltajı verdikten sonra toprağa çektiğiniz bacağın rengi yanacaktır.

1.5-3V arası gerilimle çalışmaktadır.

5V ve üstü voltaj değerleri için gerekli dirençlerle kullanılması gerekir.

Ayrıntılar: <https://maker.robotistan.com/rgb-led-nedir-arduino-kontrolu/>



```
1 #define redpin 11
2 #define greenpin 10
3 #define bluepin 9
4
5 #define potpin A0
6
7 int potdeger=0;
8
9 int reddeger=0;
10 int greendeger=0;
11 int bluedeger=0;
12
13 void deger_oku() {
14     potdeger = analogRead(potpin);
15
16     if (potdeger < 341)
17     {
18         potdeger=(potdeger*3)/4;
19         reddeger=255-potdeger;
20         greendeger=potdeger;
21         bluedeger=0;
22     }
23     else if (potdeger < 682)
24     {
25         potdeger=((potdeger-341)*3)/4;
26         reddeger=0;
27         greendeger=255 - potdeger;
28         bluedeger=potdeger;
29     }
30     else
31     {
32         potdeger=((potdeger-683)*3)/4;
33         reddeger=potdeger;
34         greendeger=0;
35         bluedeger=255 - potdeger;
36     }
37 }
38
39 void deger_yaz() {
40     analogWrite(redpin,255 - reddeger);
41     analogWrite(greenpin,255 - greendeger);
42     analogWrite(bluepin,255 - bluedeger);
43 }
44
45 void setup()
46 {
47     pinMode(redpin,OUTPUT);
48     pinMode(greenpin,OUTPUT);
49     pinMode(bluepin,OUTPUT);
50
51     //digitalWrite(redpin,HIGH);
52     //digitalWrite(greenpin,LOW);
53     //digitalWrite(bluepin,HIGH);
54 }
55
```

### 08 Arduino İle RGB LED Uygulaması

#### Devre Kısmı

- RGB LED'imizin kırmızı, mavi ve yeşil bacakların hepsine aynı anda elektrik verdiğimizde sadece kırmızı ve yeşil yanacak fakat mavi yanmayacaktır çünkü iç dirençleri farklıdır. Akımda en düşük dirençli olan yoldan geçeceğinden önce kırmızıyı sonra yeşili tercih edecektir. Mavi ışık bu yüzden yanmayabilir. Bu yüzden RGB LED'imizi daha sağlıklı kullanabilmek için Mavi, Kırmızı ve Yeşil bacakların hepsine ayrı ayrı 330 Ohm direnç bağlıyoruz.



#### Kod Kısmı

- Loop içerisine yazdıklarımızı deger\_oku ve deger\_yaz olarak fonksiyon olarak ayrı şekilde belirttik.
- Loop kısmı olmadan setup kısmını çalıştırsak beyaz ışık yanar. Yorum kısmını kaldırıp çalıştırsak mavi yanar. Bu rengi digitalWrite kısmındaki HIGH ve LOW ile değiştirirsek başka renklerde elde ederiz.
- LED'i istediğimiz parlaklık seviyesinde yakabilmek için PWM özelliğini kullanıyoruz. PWM bacaklarda (3,5,6,9,10,11) bulunduğu için bağlantılarımız bunlarla yapıyoruz.
- Analog pin üzerinden 0 ile 1023 arasında okuma yapabiliyoruz. Bu değeri 3 adet LED olduğu için 3 farklı bölgeye böldük. 0-1023 arasında bu bölgeler 0-341, 342-681, 682-1023 olarak belirledik. Gelen değer bu bölgelerden hangisinde olduğunu belirlemek için "if-else" yapısını kullandık.
- Her basamak içerisinde benzer komutlar uygulanıyor. Sadece atanan değer farklı renklerde oluyor. "if" içerisinde öncelikle gelen değer 0 ile 255 arasında oranlanıyor. LED'i PWM ile kontrol ederken 255 verdiğimizde tam parlaklıkta yanar, Ancak buradaki devrede LED'in artı bacağı değil eksi bacağı PWM pin'ine bağladığımız için ters ekti olacaktır. PWM çıkışından 0 (sıfır) verdiğimizde LED tam parlaklıkta yanacak, 255 verdiğimizde sönecektir. Bu problemi ters durumu aşmak için çıkan değerleri LED'lere yollamadan önce 255'ten çıkararak yollayacağız. Bu durumda "if" koşulları içerisinde sanki normal LED bağlamış gibi kodumuzu yazabileceğiz. İlk "if" içerisinde kırmızı LED'e göndermek üzere 2595'ten "potDeger"i çıkararak gönderiyoruz. Yeşil LED'e ise direk olarak potDeger'ini gönderiyoruz. Mavi LED'i söndürmek için (Sıfır) değerini atıyoruz. Renkler arasında geçiş için 3 basamağın her birinde bir LED'i tamamen söndürüp diğer LED'lere gönderilen değerlerin toplamının 255 olmasını sağlıyoruz. Bu yöntem ile potansiyometreyi çevirdiğimizde bir rengin parlaklığı artarken diğeri azalıyor ve renk geçişleri meydana geliyor.

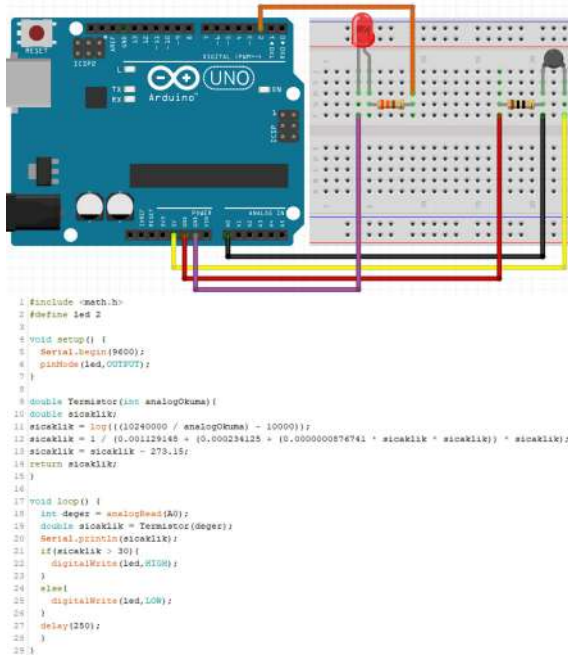
```
51 //digitalWrite(redpin,HIGH);
52 //digitalWrite(greenpin,LOW);
53 //digitalWrite(bluepin,HIGH);
54 }
55
56 void loop()
57 {
58   deger_oku();
59   deger_yaz();
60 }
```

## 09 NTC İle Sıcaklık Ölçümü

28 Aralık 2020 Pazartesi 21:09

### NTC Sıcaklık Sensörü

Bulunduğu ortamın veya temas ettiği yüzeyin sıcaklığı arttıkça elektriksel direnci azalan devre elemanıdır. – 300 C° ile +50 C° arasındaki sıcaklıklarda kararlı bir şekilde çalışabilirler.



### 09 NTC İle Sıcaklık Ölçümü

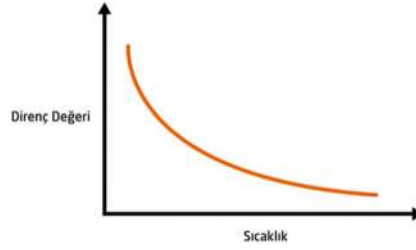
#### Teorik Bilgi

- LDR uygulamamızda yaptığımız gibi sıcaklık okurken de analog sinyali okuyup yorumlayarak istediğimiz şeyleri yaptıracağız. NTC (Negative Temperature Coefficient), sıcaklık artışı karşısında iç direncini düşüren bir elemandır. NTC yerine genellikle kullanılan elemanlardan bir tanesi de PTC'dir. PTC ise sıcaklık artışı karşısında iç direncini arttırarak tepki verir. NTC'den okuduğumuz veriyi sıcaklık birimine dönüştürmek için bir takım işlemlerden geçirmemiz gerekiyor. Aynı zamanda NTC sensörünün sıcaklık artısına göre değişken direnç değeri sabit olmadığı için logaritmik fonksiyonlardan geçirmek gerekiyor.

#### Devre Kısım

- NTC'nin bir bacağından (-) hattına diğer bacağından (+) hattına gideriz. Direnci NTC'nin (-) hattına bağladık ve direncin diğer bacağından GND'ye bağladık. LDR'nin (+) bacağından 5V'a gittik. NTC ile direncin birbirleriyle ortak bacağından Analog Giriş olan A0'gideriz.

#### Kod Kısım



28 Aralık 2020 Pazartesi 21:09

TO-92 kılıftadır. İçinde bulunan sıcaklık sensör devresi sayesinde bulunduğu ortamın sıcaklığını ölçmektedir. 4V - 30V arası gerilimde çalışmaktadır. Analog çıkış verir. -55 ile 150 derece arasında ölçüm yapabilir. 10mV/derce hassasiyete sahiptir.



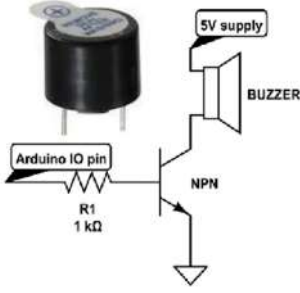
- LM35'in üç bacağı bulunmaktadır. Sensörün üzerindeki kesim kısım aynı zamanda yazılı kısma önden baktığınızda sağda kalan bacak negatif(-), solda kalan bacak pozitif(+) ve ortadaki bacak ise sinyal çıkışıdır. Sensörün pozitif bacağına +5V ve negatif bacağına GND'ye bağlıyoruz. Buzzer'ı kullanırken, LED kullanırkenki gibi direnç koyarak buzzer'ın zarar görmesini engelliyoruz. Buzzer'ın pozitif bacağına ise direnç üzerinden Arduino kartımızın 8 numaralı pinine bağlıyoruz. Buzzer'ın negatif bacağına ise GND hattına bağlıyoruz. LED'i ise pozitif bacağına direnç üzerinden Arduino kartımızın 9 numaralı pinine negatif bacağına ise GND hattına bağlıyoruz.

## 11 Ultrasonik Sensör İle Park Sensörü Yapımı

28 Aralık 2020 Pazartesi 21:14

### Buzzer

Mikrokontrolcü projelerinizde ses çıkışı almak için kullanabileceğiniz buzzer'dır.

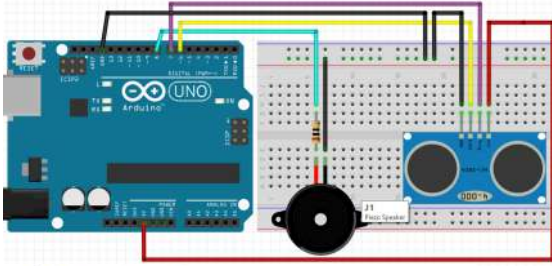


### HC-SR04 Ultrasonik Mesafe Sensörü

2cm'den 400cm'ye kadar 3mm hassasiyetle ölçüm yapabilen bu ultrasonik sensör çeşididir.

Uzaklık okuma, radar ve robot uygulamalarında kullanılabilir.

Ayrıntılar: <https://www.robotistan.com/hc-sr04-ultrasonik-mesafe-sensoru>



```
1 #define echopin 6
2 #define trigpin 7
3 #define buzzerpin 8
4
5 int maxRange=50;
6 int minRange=0;
7
8 void setup()
9 {
10   pinMode(echopin, INPUT);
11   pinMode(trigpin, OUTPUT);
12   pinMode(buzzerpin, OUTPUT);
13 }
14
15 void loop()
16 {
17   int olcum=mesafe(maxRange,minRange);
18   melodi(olcum*10);
19 }
20
21 int mesafe(int maxRange,int minRange)
22 {
23   long duration, distance;
24
25   digitalWrite(trigpin,LOW);
26   delayMicroseconds(2);
27   digitalWrite(trigpin,HIGH);
28   delayMicroseconds(10);
29   digitalWrite(trigpin,LOW);
30
31   duration=pulseIn(echopin,HIGH);
32   distance=duration/58.2;
33   delay(50);
34
35   if (distance>=maxRange || distance <=minRange)
36     return 0;
37   return distance;
38 }
39
40 int melodi(int dly)
41 {
42   tone(buzzerpin,440);
43   delay(dly);
44   noTone(buzzerpin);
45   delay(dly);
46 }
```

### 11 Ultrasonik Sensör İle Park Sensörü Yapımı

#### Teorik Bilgi

- HC-SR04 Sensörün Trig pininden uygulanan sinyal 40 kHz frekansında ultrasonik bir ses yayılmasını sağlar. Bu ses dalgası herhangi bir cisme çarpıp sensöre geri döndüğünde, Echo pini aktif hale gelir. Biz ise bu iki sinyal arasındaki süreyi ölçerek yani sesin yankısını algılayarak cismin sensörden uzaklığını tespit edebiliriz.

#### Kod Kısmı

- Long, int değişkenine göre 2 kat fazla hafıza kullanır.
- Trig pinini yüksek ve alçak yaparak sensörün fiziksel ortama ses dalgası yollamasını sağlıyoruz. Ses dalgası yollandıktan sonra "pulseIn(echoPin,HIGH)" komutu ile yolladığımız ses dalgasının cisimden yansıyıp geri gelmesini bekliyoruz. Bu beklediğimiz zamanı "pulseIn" komutu ile ölçüyoruz. Ölçtüğümüz bu değer "duration" değişkenine yazdırıyoruz. Ölçtüğümüz süreyi sesin hızına göre mesafeye çevirmek için, "58.2"ye bölüyoruz. Mesafe değerine ulaştıkça, bu değer sensörün ölçebildiği minimum (2 cm) ve maksimum (400 cm) arasında değilse 0(sıfır) değeri ile dönüş yap diyoruz. İstedğimiz aralıkta ise tekrar ana fonksiyona dönerek "olcum" değişkeni içerisine veri yazılıyor.
- Ana fonksiyonumuzda "melodi" fonksiyonuna olcum değişkeninin içindeki değer 10 ile çarpılıp gönderiliyor. Bu değer "melodi" içerisindeki bekleme sürelerinde kullanılarak 2 dıt sesi arasındaki süreyi belirleyecek. Eğer sensör az mesafe ölçüyor ise kısa aralıklarla, eğer sensör uzun mesafe algılıyor ise uzun aralıklarla ile ötecek.

## 12 Ses İle Motor Kontrolü

28 Aralık 2020 Pazartesi 21:14

### Ses Sensör Kartı

Üzerinde mikrofon bulunan ve ortamdaki ses seviyesine göre dijital çıkış veren bir karttır. Bir el çırpmasını veya bir kornayı; ani ve yüksek şekilde çıkan bir sesi veya ortam gürültüsünü sürekli olarak okuyabileceğiniz kart üzerinde potansiyometre yer almaktadır.

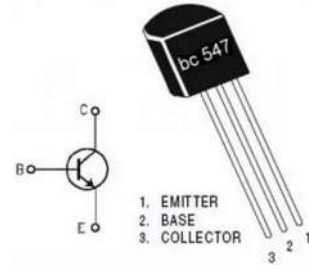
Bu sayede ortam gürültüsünden sıyrılarak anlık gelen ses çıkışını elde edebilirsiniz.

Ayrıntılar: [https://www.robotistan.com/ses-sensor-karti-mikrofon-sensoru?language=tr&h=7454c50c&gclid=Cj0KCQjw6ar4BRDnARIsAItGzICOSTsdH64uWSEs-JN422LO6N\\_RLokTm\\_KYwCX14C1IWnPOrlh1434aAi-zEALw\\_wcB](https://www.robotistan.com/ses-sensor-karti-mikrofon-sensoru?language=tr&h=7454c50c&gclid=Cj0KCQjw6ar4BRDnARIsAItGzICOSTsdH64uWSEs-JN422LO6N_RLokTm_KYwCX14C1IWnPOrlh1434aAi-zEALw_wcB)

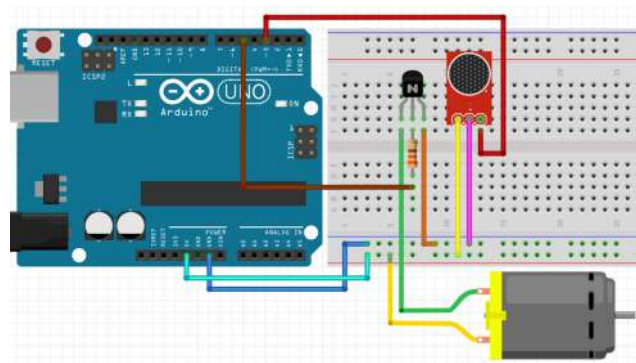


### BC 547 NPN Transistör

TO-92 kılıfta NPN transistördür



### DC Motor



```
1 #define SensorPin 3
2 #define MotorPin 5
3 int MotorDurum = LOW;
4
5 void setup() {
6   pinMode(SensorPin, INPUT);
7   pinMode(MotorPin, OUTPUT);
8 }
9
10 void loop() {
11   if( digitalRead(SensorPin) ){
12     if(MotorDurum == LOW){
13       MotorDurum = HIGH;
14     }
15   }
16   else{
17     MotorDurum = LOW;
18   }
19   digitalWrite(MotorPin, MotorDurum);
20   delay(50);
21 }
```

### 12 Ses İle Motor Kontrolü

#### Teorik Bilgi

- Ses sensörü, üzerinde bulunan mikrofon aracılığıyla ortamdaki ses seviyesini ölçerek dijital bir çıkış vermektedir. Sensör devresi, mikrofondan alınan ses sinyalini yükseltir ve analog ses sinyalini eşik seviyesine göre dijital sinyale çevirir.
- Motor, fazla akım çektiği için bu tür devrelerde motor sürücü kartı kullanılır. Motor sürücü, Arduino'dan aldığı sinyale göre motora güç verir. Bu sayede Arduino'ya zarar vermeden güvenli bir şekilde motoru kontrol edebiliriz.
- Ses sensörü üzerinde bulunan potansiyometre ile ses seviyesinin eşik değeri ayarlanabilmektedir. Bir tornavida yardımıyla bu ayar yapılabilir.



## 13 Joystick İle Servo Motor Kontrolü

28 Aralık 2020 Pazartesi 21:14

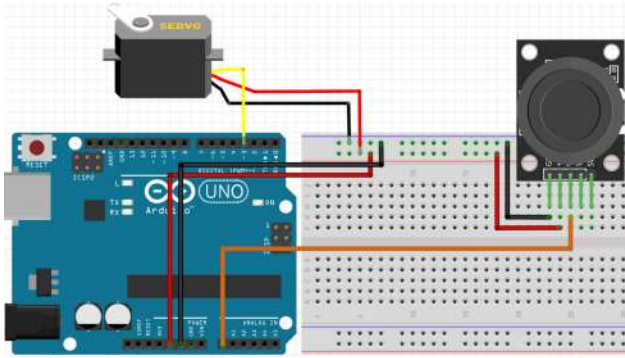
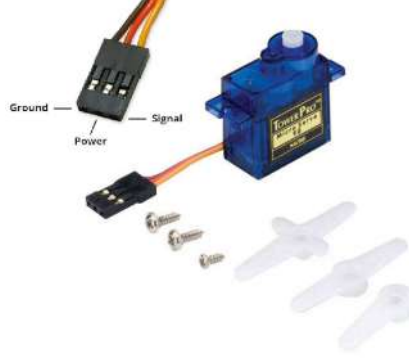
### 2 Eksenli Joystick Kartı

X ve Y eksenli olmak üzere iki eksenli analog çıkış verir. Bununla birlikte joystick'in ortasında bir adet de buton bulunmaktadır. Kart üzerinde güç ledi bulunmaktadır.



### SG90 Servo Motor

Ayrıntılar: <https://www.robotistan.com/tower-pro-sg90-rc-mini-servo-motor>  
<https://maker.robotistan.com/rc-servo-motor-nedir/>



```
1 #include <Servo.h>
2
3 Servo motor;
4 int deger;
5 int derece;
6
7 void setup() {
8   motor.attach(3);
9 }
10
11 void loop() {
12   deger = analogRead(A0);
13   derece = map(deger, 0, 1023, 0, 180);
14   motor.write(derece);
15 }
```

### 13 Joystick İle Servo Motor Kontrolü

#### Teorik Bilgi

- Joystick, üzerindeki kolun ileri-geri veya sağ-sol hareketi yaptırılmasına göre analog sinyal olarak pozisyon çıkışı verir.
- Servo, data pininden aldığı sinyale göre belli bir açıda kendini konumlandıran bir motordur. Joystick üzerindeki kolu ileri-geri hareket ettirdiğimizde VRX pinindeki, sağa-sola hareket ettirdiğimizde VRY pinindeki gerilim değerleri değişir. Joystick üzerine tıkladığımızda ise SW pini 5V çıkış verir. Bu örnekte sadece bir adet servo kullanacağımızdan VRX pinini kullanacağız. VRX pininden alacağımız veri 0 ile 5V arasında analog bir veri olduğundan bu pini Arduino üzerindeki A0 pinine bağlıyoruz. Servo motorun data pinini ise analog çıkış verebilen 3 numaralı pine bağlıyoruz. Örnek kodumuz joystickten alınan veri ile servo motorun 0 ile 180 derece arasında dönmesini sağlar.

#### Kod Kısmı

- Arduino Uno, analog okuma pinleri üzerinden 0 ile 1023 arasında veri vermektedir. Ancak servo motorumuz 0 ile 180 derece arasında hareket edebilmektedir. Bu yüzden, A0 pininden okuduğumuz değer ile servo motorumuzu kontrol edebilmek için "map()" fonksiyonunu kullanıyoruz. "map()" fonksiyonu girdi olarak verilen değişkenin istenilen aralığa oranlanması sağlar.
- "map()" fonksiyonu ile oranlanan okuma değeri, derece değerine eşitlenir. Son olarak derece değeri servo motora yazdırılarak servonun istenilen dereceye gelmesi sağlanır.

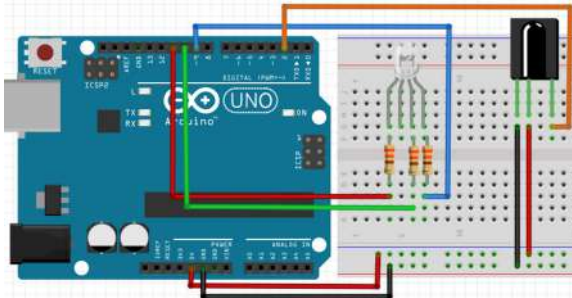
## 14 IR Kumanda İle LED Kontrolü

28 Aralık 2020 Pazartesi 21:14

### IR Alıcı Verici Kumanda Seti

Alıcı kısmında 38 KHz'lik 1838B kızılötesi alıcı göz bulunmaktadır. Kumanda ise 17 butonludur. Devre bağlantısını sağlamanız için iki ucu da dışı jumper kablolar da paket dahilindedir ve işlerinizi kolaylaştıracaktır. Açık alanda 8 metreye kadar haberleşme sağlanabilmektedir. Alıcı 60°'lik bir görme açısına sahiptir.

Ayrıntılar:<https://www.robotistan.com/ir-alici-verici-kumanda-seti-ir-receiver-module-wireless-remote-control-k>



```
1 #include <IRremote.h>
2
3 int RECV_PIN = 2;
4 int kirmiziled= 11;
5 int yesilled= 10;
6 int maviled= 9;
7 IRrecv irrecv(RECV_PIN);
8 decode_results results;
9
10 #define CH1 0xFFA25D
11 #define CH 0xFF629D
12 #define CH2 0xFFE21D
13 #define PREV 0xFF22DD
14 #define NEXT 0xFF02FD
15 #define PLAYPAUSE 0xFFC23D
16 #define VOL1 0xFFE01F
17 #define VOL2 0xFFA857
18 #define EQ 0xFF906F
19 #define BUTON0 0xFF6897
20 #define BUTON100 0xFF9867
21 #define BUTON200 0xFFB04F
22 #define BUTON1 0xFF30CF
23 #define BUTON2 0xFF18E7
24 #define BUTON3 0xFF7A85
25 #define BUTON4 0xFF10EF
26 #define BUTON5 0xFF38C7
27 #define BUTON6 0xFF5AA5
28 #define BUTON7 0xFF42BD
29 #define BUTON8 0xFF4AB5
30 #define BUTON9 0xFF52AD
31
32 void setup()
33 {
34   pinMode(kirmiziled, OUTPUT);
35   pinMode(yesilled, OUTPUT);
36   pinMode(maviled, OUTPUT);
37   Serial.begin(9600);
38   irrecv.enableIRIn();
39 }
40 void loop() {
41
42   if (irrecv.decode(&results)){
43     if (results.value == BUTON1){
44       digitalWrite(kirmiziled, !digitalRead(kirmiziled));
45       if (digitalRead(kirmiziled) == HIGH){
46         Serial.println("Kirmizi yandı");
47       }
48       else{
49         Serial.println("Kirmizi sondu");
50       }
51     }
52     if (results.value == BUTON2)
53     {
54       digitalWrite(yesilled, !digitalRead(yesilled));
55       if (digitalRead(yesilled) == HIGH)
56       {
57         Serial.println("Yesil yandı");
58       }
59       else
60       {
61         Serial.println("Yesil sondu");
62       }
63     }
64     if (results.value == BUTON3)
65     {
66       digitalWrite(maviled, !digitalRead(maviled));
67       if (digitalRead(maviled) == HIGH)
68       {
69         Serial.println("Mavi yandı");
70       }
71       else
72       {
73         Serial.println("Mavi sondu");
74       }
75     }
76     if (results.value == BUTON4)
77     {
78       digitalWrite(kirmiziled, LOW);
79       digitalWrite(yesilled, LOW);
80       digitalWrite(maviled, LOW);
81       Serial.println("Tüm LED'ler söndü");
82     }
83   }
84 }
```

### 14 IR Kumanda İle LED Kontrolü

#### Teorik Bilgi

- IR kumandanın üzerinde bir adet kızılötesi led bulunur. Kızılötesi LED bastığımız butonun adresine göre belli bir frekansta yanıp söner ve IR alıcıya sinyal gönderir. Genellikle bu kızılötesi LED'in yanıp sönmeye frekansı 38kHz'dir.

#### Kod Kısmı

- "loop" kısmında, "irrecv.decode(&results)" fonksiyonu ile kumandanın gelen verileri okuyoruz ve "results" değişkenine atıyoruz. "results" değişkenini tanımladığımız adreslerle karşılaştırıyoruz. "results" değişkeni "BUTON1" e eşit ise kırmızı, "BUTON2"ye eşit ise yeşil, "BUTON3"e eşik ise mavi LED'i yakıyoruz. "BUTON4" a eşit ise tüm LED'leri söndürüyoruz ve "BUTON4"e eşit ise bütün LED'leri yakıp RGB LED'in beyaz yanmasını sağlıyoruz.

```

56 {
57     Serial.println("Yesil yandi");
58 }
59 else
60 {
61     Serial.println("Yesil sondu");
62 }
63 }
64 if (results.value == BUTON3)
65 {
66     digitalWrite(maviled, !digitalRead(maviled));
67     if (digitalRead(maviled) == HIGH)
68     {
69         Serial.println("Mavi yandi");
70     }
71     else
72     {
73         Serial.println("Mavi sondu");
74     }
75 }
76 if (results.value == BUTON0)
77 {
78     digitalWrite(kirmiziled, LOW);
79     digitalWrite(yesilled, LOW);
80     digitalWrite(maviled, LOW);
81     Serial.println("Tum LED'ler sondu");
82 }
83 if (results.value == BUTON4)
84 {
85     digitalWrite(kirmiziled, HIGH);
86     digitalWrite(yesilled, HIGH);
87     digitalWrite(maviled, HIGH);
88     Serial.println("Tum LED'ler yandi");
89 }
90 irrecv.resume();
91 }
92 }

```

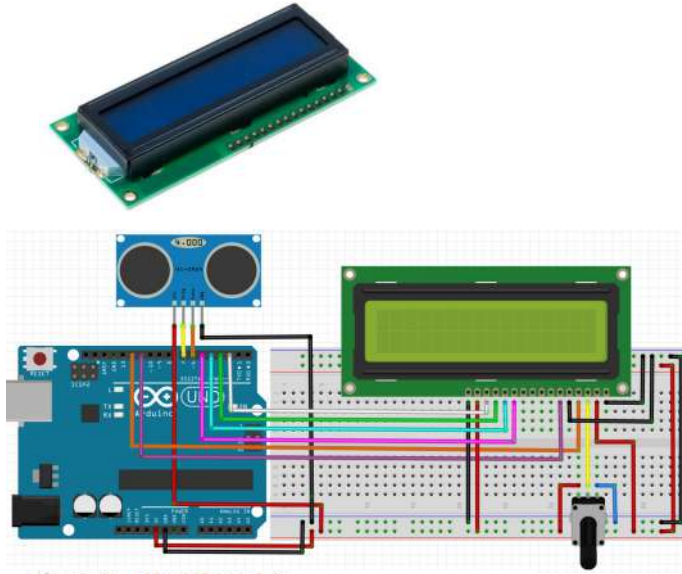
## 15 Arduino İle Dijital Metre Yapımı

28 Aralık 2020 Pazartesi 21:22

### 2x16 LCD Ekran

16x2 (16 sütun, 2 satır) LCD ekrandır. Mavi zemin üzerine beyaz yazı rengine sahiptir. Arka LED aydınlatması mevcuttur. 5V gerilim ile çalışır.

Ayrıntılar:<https://www.robotistan.com/2x16-lcd-ekran-mavi-uzerine-beyaz>



```
1 #include <LiquidCrystal.h>
2
3 int trigPin = 7;
4 int echoPin = 6;
5 int sure;
6 int uzaklik;
7 int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
8 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
9
10 void setup() {
11   pinMode(trigPin, OUTPUT);
12   pinMode(echoPin, INPUT);
13   lcd.begin(16, 2);
14 }
15
16 void loop() {
17   digitalWrite(trigPin, LOW);
18   delayMicroseconds(5);
19   digitalWrite(trigPin, HIGH);
20   delayMicroseconds(10);
21   digitalWrite(trigPin, LOW);
22
23   sure = pulseIn(echoPin, HIGH, 11600);
24   uzaklik= sure*0.0345/2;
25   delay(250);
26
27   lcd.clear();
28   lcd.setCursor(0, 0);
29   lcd.print("Uzaklik:");
30   lcd.setCursor(0, 1);
31   lcd.print(uzaklik);
32   lcd.print("cm");
33 }
```

### 15 Arduino İle Dijital Metre Yapımı

#### Teorik Bilgi

- Ultrasonik mesafe sensörü, bir ses dalgası gönderebilen ve yansıyan ses dalgasını tespit edebilen bir cihazdır. LCD, verdiğimiz verilere göre ekrana karakter yazan bir elemandır. LCD ekran, 2 satırdan oluşmaktadır ve her satıra 16 adet karakter yazılabilmektedir. Bir karakter 5x7 tane pixelden oluşmaktadır.
- Potansiyometre, ayarlanabilir bir dirençtir. Bu devrede potansiyometreyi gerilim bölücü olarak kullandık. Seri bağlı iki farklı dirence gerilim uyguladığımızda, dirençler üzerinde direnç değerleri ile doğru orantılı gerilimler elde ederiz. Potansiyometre döndürüldüğünde orta pininin gerilimi değişmektedir. Bu değişen gerilimde LCD ekranın Kontrastını ayarlamamızı sağlayacak.

#### Devre Kısmı

##### Bacak Bağlantıları:

LCD Pin	Sembol	Bağlantı
1	Vss	GND
2	Vdd	+5V
3	Vo	Kontrast ayarı
4	RS	Register seçme pini
5	R/W	Data read/write pini.
6	E	Enable pini
7	DB0	Data Bus 0
8	DB1	Data Bus 1
9	DB2	Data Bus 2
10	DB3	Data Bus 3
11	DB4	Data Bus 4
12	DB5	Data Bus 5
13	DB6	Data Bus 6
14	DB7	Data Bus 7
15	A	LED arka ışık anot (+5V)
16	K	LED arka ışık katot (GND)

#### Kod Kısmı

- 7. ve 8.satırda LCD'nin pin bağlantılarını ayarlıyoruz.
- 13.satırda "lcd.begin()" fonksiyonu ile LCD satır-sütun uzunluk ayarı yapılır.
- "loop" kısmında, önce trig pini LOW seviyesine getirilerek ultrasonik sensör ölçüm için hazır duruma getirilir. Daha sonra trig pini önce HIGH daha sonra LOW seviyesine çekilerek ses dalgası gönderilir.
- "pulseIn()" fonksiyonu ile ses dalgasının toplam gidiş-geliş süresi ölçülür. Uzaklığı hesaplamak için gidiş-geliş süresi 0.0345 sayısı ile çarpılır. 0.0345 sayısı, ses dalgasının 1 mikrosaniyede aldığı mesafedir. Ses dalgası gidip geldiği için hesaplanan uzaklık değerinin ikiye bölünmesi gerekir. Park sensörü yapımı uygulamamızda ölçtüğümüz değeri "58.2" sayısına bölmüştük. Bu iki işlem arasında bir fark bulunmuyor.
- "lcd.clear()" fonksiyonu ile daha önceden kalan yazılar ekrandan silinir.
- "lcd.setCursor()"fonksiyonu ile LCD ekrana yazacağımız yazının hangi satır ve sütuna yazdırılacağı ayarlanır.
- uzaklık değeri LCD ekrana "lcd.print()" fonksiyonu ile yazdırılır.

## 16 Hareket Sensörü (PIR) İle Servo Motor Kontrolü

28 Aralık 2020 Pazartesi 21:22

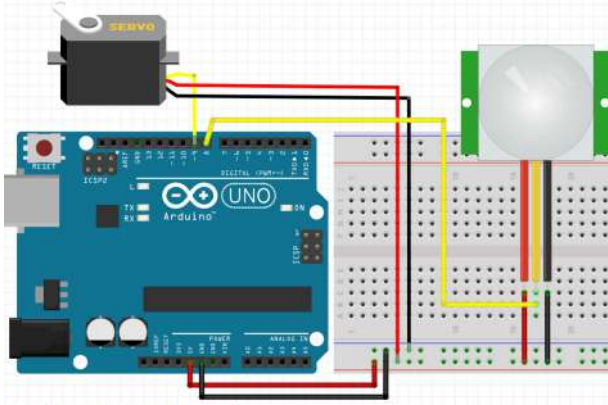
### PIR Sensörü (HC-SR501 Ayarlanabilir IR Hareket Algılama Sensörü

PIR sensörleri, bir ortamda oluşan canlı hareketini algılamak için kullanılan sensörlerdir.

Dijital çıkışlı olan bu modül, ortamda hareket algılamadığı zaman lojik 0, hareket algıladığı zaman ise lojik 1 çıkışı vermektedir.

Sensör üzerinde Sx ve Tx olmak üzere iki adet potansiyometre bulunmaktadır. Sx potansiyometresi sensörün görme mesafesini 3 ile 5 metre arasında değiştirmektedir. Tx potu ise sensör gördükten sonra ne kadar süre daha çıkış pininden lojik 1(3.3V) çıkışını vereceğini ayarlamaktadır.

Ayrıntılar:[https://www.robotistan.com/hc-sr501-ayarlanabilir-ir-hareket-algilama-sensoru-pir?](https://www.robotistan.com/hc-sr501-ayarlanabilir-ir-hareket-algilama-sensoru-pir?language=tr&h=b078908f&gclid=Cj0KCQjw6ar4BRDnARIsAITGzIBggU1Y5AmPDJiOJ-xE0LSzejvCLb78XH5uzAro2msE7qe0NaoCyQaAvK0EALw_wcB)  
[language=tr&h=b078908f&gclid=Cj0KCQjw6ar4BRDnARIsAITGzIBggU1Y5AmPDJiOJ-xE0LSzejvCLb78XH5uzAro2msE7qe0NaoCyQaAvK0EALw\\_wcB](https://www.robotistan.com/hc-sr501-ayarlanabilir-ir-hareket-algilama-sensoru-pir?language=tr&h=b078908f&gclid=Cj0KCQjw6ar4BRDnARIsAITGzIBggU1Y5AmPDJiOJ-xE0LSzejvCLb78XH5uzAro2msE7qe0NaoCyQaAvK0EALw_wcB)



```
1 #include <Servo.h>
2
3 int pirPin = 8;
4 int servoPin = 9;
5 int hareket;
6 Servo motor;
7
8 void setup() {
9   motor.attach(servoPin);
10  pinMode(pirPin, INPUT);
11
12 }
13
14 void loop() {
15   hareket = digitalRead(pirPin);
16   if(hareket == HIGH){
17     motor.write(150);
18     delay(250);
19     motor.write(30);
20     delay(250);
21     motor.write(150);
22     delay(250);
23     motor.write(30);
24     delay(250);
25     motor.write(150);
26     delay(250);
27     motor.write(30);
28     delay(250);
29     motor.write(90);
30   }
31   else{
32     motor.write(90);
33   }
34 }
```

### 16 Hareket Sensörü (PIR) İle Servo Motor Kontrolü

#### Teorik Bilgi

- PIR ismi Passive Infra-Red kelimelerinin baş harflerinden gelmektedir. Bu da bu sensörün kızılötesi dalgalarla çalıştığı anlamına gelir. Hareket sensörü ortamdaki sıcaklık ve kızılötesi dalga değişimlerini algılamaya yarayan sensörlerdir. Yapılarında bir fresnel lens bulunur. Bu lens sayesinde ortamdaki nesnelerden gelen ışınlar sensörün odaklanmasını sağlar. Ortamda bir dalga değişimi olduğunda sensör algılama işlemi gerçekleştirir.

#### Devre Kısmı

#### Kod Kısmı

- 9.satır ile servomuzu 9. pin ile ilişkilendiriyoruz.



## 17 Bluetooth İle RGB LED Kontrolü

28 Aralık 2020 Pazartesi 21:22

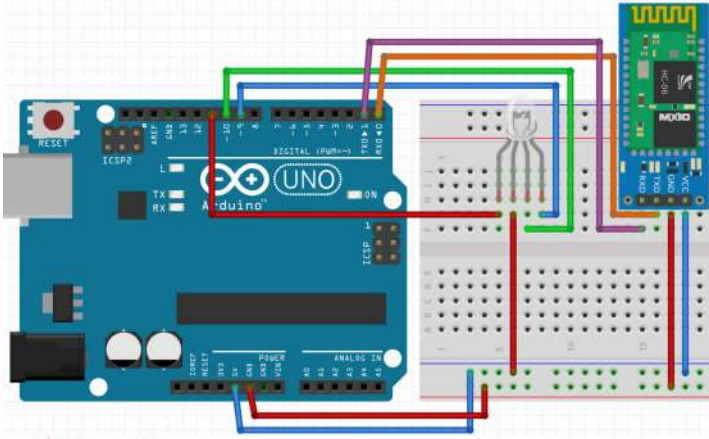
### HC06 Bluetooth Modül

HC06 Bluetooth-Serial Modül Kartı, Bluetooth SSP(Serial Port Standart) kullanımı ve kablosuz seri haberleşme uygulamaları için tasarlanmıştır.

Bluetooth 2.0'ı destekleyen bu kart, 2.4GHz frekansında haberleşme yapılmasına imkan sağlayıp açık alanda yaklaşık 10 metrelik bir haberleşme mesafesine sahiptir.

Bir çok hobi, robotik ve akademik projede kullanılabilir.

Ayrıntılar:<https://www.robotistan.com/kablolu-hc06-bluetooth-serial-modul-karti-hc06-bluetooth-to-serial-port-m>



```
1 int veri;
2 int kirmiziPin = 11;
3 int yesilPin = 10;
4 int maviPin = 9;
5
6 void setup() {
7   Serial.begin(9600);
8   pinMode(kirmiziPin, OUTPUT);
9   pinMode(yesilPin, OUTPUT);
10  pinMode(maviPin, OUTPUT);
11 }
12 void loop() {
13   if(Serial.available() > 0) {
14     veri = Serial.read();
15   }
16   if(veri == 'k') {
17     digitalWrite(kirmiziPin, LOW);
18     digitalWrite(yesilPin, HIGH);
19     digitalWrite(maviPin, HIGH);
20   }
21   else if(veri == 'y') {
22     digitalWrite(kirmiziPin, HIGH);
23     digitalWrite(yesilPin, LOW);
24     digitalWrite(maviPin, HIGH);
25   }
26   else if(veri == 'm') {
27     digitalWrite(kirmiziPin, HIGH);
28     digitalWrite(yesilPin, HIGH);
29     digitalWrite(maviPin, LOW);
30   }
31   else {
32     digitalWrite(kirmiziPin, HIGH);
33     digitalWrite(yesilPin, HIGH);
```

### 17 Bluetooth İle RGB LED Kontrolü

#### Teorik Bilgi

- Bluetooth modülü Arduino ile TX/RX protokolü ile birbirine bağlanır. RX "Receive" yani almak, TX ise "Transmit" yani vermek kelimelerinden gelmektedir. Yani kısaca bir alıcı/verici protokolüdür.
- Arduino, Bluetooth modülü ile haberleşmek için UART(Universal Asynchronous Receiver Transmitter) protokolünü kullanır. Bu bir seri haberleşme protokolüdür. UART protokolünü kullanabilmemiz için iki taraftaki Baud Rate(Haberleşme hızları) 'nın aynı olması gerekmektedir. 4800, 9600, 57600, 115200 en çok kullanılan haberleşme hızlarıdır. Bluetooth modülü öntanımlı olarak bu haberleşme hızını kullanmaktadır.

#### Devre Kısmı

- Bağlantıda Arduino'nun TX pini Bluetooth modülünün RX pinine bağlanmalı, Arduino'nun RX pini de Bluetooth modülünün TX pinine çapraz şekilde bağlanmalıdır.

#### Kod Kısmı

- Bilgisayardan Arduino'ya yükleme yaparken 0 nolu pindeki jumper kablo çıkarılması gerekir.
- 12.satır Seri haberleşmeden veri gelmesini bekler.
- 13.satır Seri haberleşmeden gelen veriyi okur.



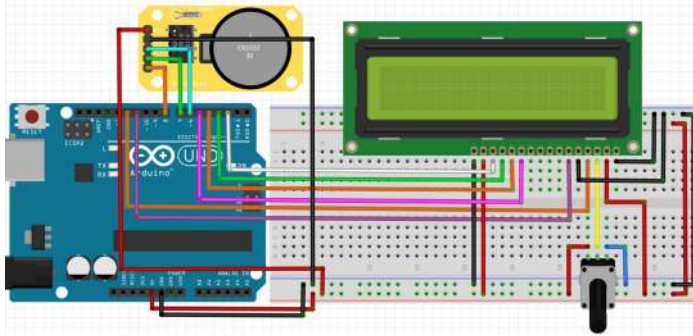
```
30 |  
31 | else(  
32 |     digitalWrite(kirmiziPin,HIGH);  
33 |     digitalWrite(yesilPin,HIGH);  
34 |     digitalWrite(maviPin,HIGH);  
35 | )  
36 | }
```

# 18 Arduino İle Dijital Saat Yapımı

28 Aralık 2020 Pazartesi 21:22

## DS1302 RTC Modülü

Gerçek zamanlı saat devresi entegresi olup saat bilgisini anlık ve sürekli olarak okuyabileceğiniz çok kullanışlı bir modüldür. Kart üzerinde CR2032 pil konnektörü bulunur. DS1302 entegresi saniye, dakika, saat, gün, ay ve yıl bilgisini sürekli olarak içinde tutabilen bir RTC entegresidir. Seri olarak SCLK pini üzerinden sürekli olarak çıkış verebilen karttır.



```
1 #include <LiquidCrystal.h>
2 #include <virtuabotixRTC.h>
3
4 int CLK_PIN = 6;
5 int DAT_PIN = 7;
6 int RST_PIN = 8;
7 virtuabotixRTC myRTC(CLK_PIN, DAT_PIN, RST_PIN);
8
9 int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
10 LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
11
12 void setup() {
13   Serial.begin(9600);
14   //myRTC.setDS1302Time(10, 11, 01, 4, 31, 12, 2020);
15   lcd.begin(16, 2);
16 }
17
18 void loop() {
19   myRTC.updateTime();
20
21   Serial.print("Tarih / Saat: ");
22   Serial.print(myRTC.dayofmonth);
23   Serial.print("/");
24   Serial.print(myRTC.month);
25   Serial.print("/");
26   Serial.print(myRTC.year);
27   Serial.print(" ");
28   Serial.print(myRTC.hours);
29   Serial.print(":");
30   Serial.print(myRTC.minutes);
31   Serial.print(":");
32   Serial.println(myRTC.seconds);
33
34   lcd.clear();
35   lcd.setCursor(0,0);
36   lcd.print(myRTC.dayofmonth);
37   lcd.print("/");
38   lcd.print(myRTC.month);
39   lcd.print("/");
40   lcd.print(myRTC.year);
41   lcd.setCursor(0, 1);
42   lcd.print(myRTC.hours);
43   lcd.print(":");
44   lcd.print(myRTC.minutes);
45   lcd.print(":");
46   lcd.print(myRTC.seconds);
47
48   delay(1000);
49 }
50
```

## 18 Arduino İle Dijital Saat Yapımı

### Teorik Bilgi

RTC modülleri zamanı sürekli senkron tutması için üretilmiştir. Çok az güçle bile çalışabilen bu modül üzerinde bulunan pil ile uzun yıllar boyunca zamanda sapmaya yol açmadan üzerinde bulunan kristal sayesinde sayım yapar. Bu kristal saniyede 32000 sinyal üretmektedir. RTC bu sinyalleri okur ve her 32000 adımda bir saniye ileri saymaktadır.

Saati kullanabilmemiz için öncelikle zamanı şu anki zamana göre ayarlamamız lazım. Onun için ilk önce ayar kodumuzu yüklememiz gerekmektedir.

### Devre Kısmı

#### Kod Kısmı

- 6. pini clock, 7. pini data, 8. pini reset pini olarak tanımladık.
- 14.satırda saati saniye, dakika, saat, haftanın günü, ayın günü, ay, yıl olarak ayarlıyoruz. Bu işlem zaman hatalı ise düzeltmek için kullanıyoruz yoksa her yüklemeye bu belirli tarihlerden saymaya başlar
- 19.satırda RTC'den zamanı okuyoruz
- loop() kısmında ilk önce RTC den zaman verisini okuyoruz. Ardında LCD'nin ekranındaki karakterleri temizliyoruz. Bunu yapmaz isek kullandığımız karakterler üst üste binebilir ve yanlış veri elde etmemize yol açar.
- İlk satır ilk sütundan itibaren; gün, ay ve yıl olacak şekilde tarihi bastırıyoruz. İkinci satır ilk sütundan itibaren ise saat, dakika, saniye olacak şekilde zamanı bastırıyoruz ve 1 saniyelik bir bekleme koyuyoruz.

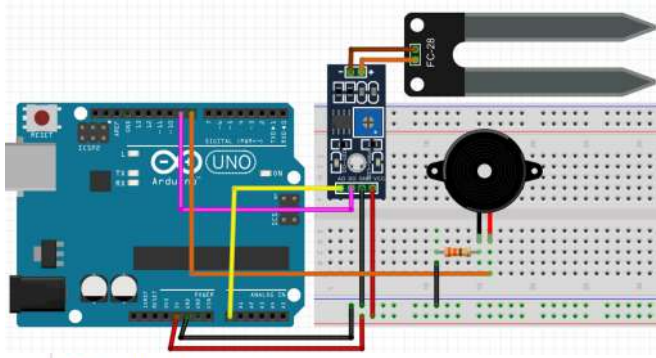
# 19 Arduino İle Toprak Nem Sensörü Kullanımı

28 Aralık 2020 Pazartesi 21:22

## Toprak Nem Algılama Sensörü

Nem ölçer problar ölçüm yapılacak ortama batırılarak kullanılır. Toprağın veya içine batırılan sıvının meydana getirdiği dirençten dolayı, prob uçları arasında bir gerilim farkı oluşur. Bu gerilim farkının büyüklüğüne göre de nem miktarı ölçülebilir. Topraktaki nem oranı arttıkça iletkenliği de artmaktadır. Kart üzerinde yer alan trimpot sayesinde hassasiyet ayarı yapılabilir.

Ayrıntılar:[https://www.robotistan.com/toprak-nemi-algilama-sensoru?language=tr&h=74d31b1a&gclid=Cj0KCQjw6ar4BRDnARIsAITGzIC5sSraCJYBAfv9vL3OX5DojohZzgsCFLPmEUfibh6zLdNnXYRlgaAhV7EALw\\_wcB](https://www.robotistan.com/toprak-nemi-algilama-sensoru?language=tr&h=74d31b1a&gclid=Cj0KCQjw6ar4BRDnARIsAITGzIC5sSraCJYBAfv9vL3OX5DojohZzgsCFLPmEUfibh6zLdNnXYRlgaAhV7EALw_wcB)



```
1 int sensorPin1 = 9;
2 int sensorPin2 = A0;
3 int buzzerPin = 8;
4
5 int veri;
6 int olcum;
7
8 void setup() {
9   Serial.begin(9600);
10  pinMode(sensorPin1, INPUT);
11  pinMode(buzzerPin, OUTPUT);
12 }
13 void loop() {
14   veri = digitalRead(sensorPin1);
15   if(veri == true){
16     digitalWrite(buzzerPin, HIGH);
17     delay(100);
18     digitalWrite(buzzerPin, LOW);
19     delay(100);
20   }
21   else{
22     digitalWrite(buzzerPin, LOW);
23   }
24   olcum=analogRead(sensorPin2);
25   Serial.print("Islaklik= ");
26   Serial.println(olcum);
27   delay(500);
28 }
```

## 19 Arduino İle Toprak Nem Sensörü Kullanımı

### Teorik Bilgi

- Toprak nem sensörü iki elektrottan oluşur. Bu elektrotlar arasındaki iletkenliği ölçerek bize toprağın ne kadar nemli olduğuna dair bilgi verir. Toprak nemli ise elektrotlar arasındaki iletkenlik artar. İletkenlik arttığında direnç azalacağından sensörün içindeki gerilim bölücünden daha az voltaj gelmeye başlar. Toprak kuru ise elektrotlar arasındaki direnç artacağından daha yüksek bir voltaj elde etmiş oluruz ve bu veriyi analog olarak işleyerek topraktaki nem miktarını öğrenmiş oluruz.
- Toprak Nem ölçer ile DHT11 ısı ve nem sensörü arasındaki farklardan ilki toprak nem ölçerin toprağa veya sıvıya batırılması gerektiği. DHT11 ısı ve nem sensörü ise her türlü sıcaklık ölçümü yapabiliyor ve herhangi bir alana batırılması gerekmiyor. Bir diğer fark ise toprak nem ölçer sensöründen hem analog hem de dijital bir çıkış almak mümkün. DHT11 sensörü ise sadece dijital çıkış üretebiliyor.
- Buzzer'ın aktif ya da pasif olup olmadığını anlamak için buzzerın alt tarafına bakılır. Eğer altı epoksi kaplı ise aktif değilse pasiftir.

### Devre Kısmı

- Topraktaki nem miktarı belli bir eşik değerini geçtiğinde bize bir sinyal gönderecek ve devre üzerindeki buzzer ses çıkarmaya başlayacak. Bu devrede eşik değerini sensörün üzerinde bulunan potansiyometre ile ayarladık. Yani eşik değer ayarladığımız değerin üzerine çıkarsa D0 pininden bize 0V'luk bir çıkış verecektir. Aynı şekilde A0 pininden 0V-5V aralığında analog olarak aldığımız veriyi kod içinde belli bir eşik değeri geçtiğinde çalışacak şekilde hazırlayabiliriz.

### Kod Kısmı

- True, HIGH ya da 1 aynı anlamdadır.
- "loop" kısmında, "digitalRead()" fonksiyonu ile sensörden alınan değer veri değişkenine eşitlenir. Daha sonra veri değişkeninin durumu "if-else" yapısı ile kontrol edilir.
- Toprak nem sensörümüz su algıladığında dijital çıkış pininden 0V sinyal vermektedir. Veri değişkeninin LOW değerinde olması, toprak nem sensörü tarafından suyun algılandığını göstermektedir. Bu durumda veri değişkeni HIGH değerini alır ve else yapısının içindeki komutlar çalıştırılır.
- Toprak nem sensörü tarafından su algılanmadığında, sensörün dijital çıkış pininden 5V çıkış verilmektedir. Bu durumda buzzer ötmeye başlar.

## 20 Arduino İle Yağmur Sensörü Kullanımı

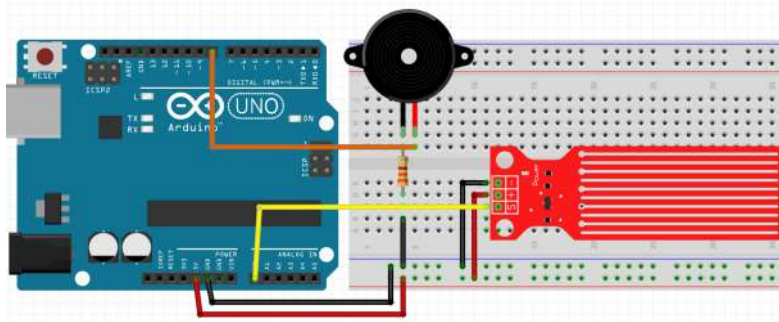
28 Aralık 2020 Pazartesi 21:26

### Su/Yağmur Sensörü

Hem sığ sular için su seviyesi(40mm'ye kadar) ölçüm sensörü hem de yağmur sensörü olarak kullanılabilir.

Birbirine paralel olarak çekilmiş iletken hatların su ile teması sonucu sensör çıkış pininde analog bir değer okunabilmektedir.

Besleme voltajı ve toprak bağlantısı yapılarak, sensör çıkış bacağından okuma yapılabilir.



```
1 int sensorPin = A0;
2 int esikDegeri = 100;
3 int buzzerPin = 8;
4 int veri;
5
6 void setup() {
7   Serial.begin(9600);
8   pinMode(buzzerPin, OUTPUT);
9 }
10 void loop() {
11   veri = analogRead(sensorPin);
12   if(veri > esikDegeri){
13     digitalWrite(buzzerPin, HIGH);
14     delay(100);
15     digitalWrite(buzzerPin, LOW);
16     delay(100);
17   }
18   else{
19     digitalWrite(buzzerPin, LOW);
20   }
21 }
```

### 20 Arduino İle Yağmur Sensörü Kullanımı

#### Teorik Bilgi

- Yağmur sensörü sığ su seviyelerinde ve yağmurlu ortamda kullanılan bir sensör çeşididir. Sensör 40 mm su seviyesine kadar ölçüm yapabilmektedir. Yağmur sensörünün yapısında birbirine paralel olarak bağlanmış iletken hatlar bulunmaktadır. Bu hatlar su ile temas ettiğinde Arduino'ya analog bir sinyal gönderir.

#### Devre Kısmı

- Yağmur sensörü 5V gerilim ile çalışmaktadır. 3 pini bulunmaktadır. + pinine besleme gerilimi olan 5V, – pinine GND bağlantısı yapılır. S pini ise sinyal (data) pini olduğundan dolayı Arduino'nun Analog pinlerinden biri bağlanır.

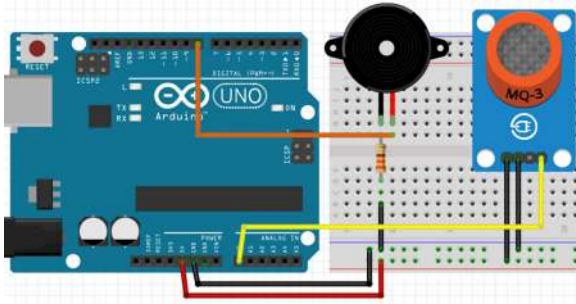
#### Kod Kısmı

## 21 Arduino İle Gaz Sensörü Kullanımı

28 Aralık 2020 Pazartesi 21:26

### MQ-2 Gaz Sensörü

300 ile 10.000 ppm konsantrasyonlarda yanıcı gaz ve sigara dumanını algılayacak türden bir sensördür. Sensör, taşıyıcı kartı ile birlikte gelir. 5V gerilim ile çalışır ve dijital (TTL seviyesinde) ve analog çıkışlara sahiptir.



```
1 int esikDegeri = 400;
2 int buzzerPin = 8;
3 int deger;
4
5 void setup() {
6   pinMode(buzzerPin, OUTPUT);
7 }
8
9 void loop() {
10  deger = analogRead(A0);
11  if(deger > esikDegeri){
12    digitalWrite(buzzerPin, HIGH);
13    delay(50);
14    digitalWrite(buzzerPin, LOW);
15    delay(50);
16  }
17  else{
18    digitalWrite(buzzerPin, LOW);
19  }
20 }
```

### 21 Arduino İle Gaz Sensörü Kullanımı

#### Teorik Bilgi

- Gaz sensörünün içinde bir dizi direnç bulunmaktadır. Ortamdaki gaz, sensörün içindeki dirençlerle etkileşime girerek direnç değerlerini değiştirir. Sensör, içinde bulunan bir gerilim bölücü sayesinde gaz yoğunluğuna göre farklı gerilim değerleri verir ve bu gerilim değerlerinin analog bir şekilde okunmasıyla ortamdaki gaz miktarı ölçülür.

#### Devre Kısmı

#### Kod Kısmı



## 22 Arduino İle RFID Sensörü Kullanımı

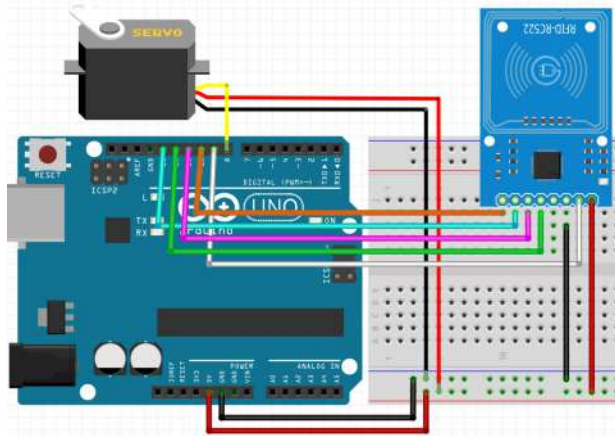
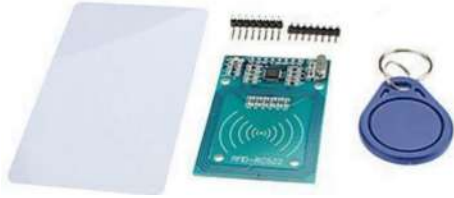
28 Aralık 2020 Pazartesi 21:26

### RC522 RFID NFC Kit

RC522 RFID kartı, NFC frekansı olan 13,56 MHz frekansında çalışan tagler üzerinde okuma ve yazma işlemi yapabilen, düşük güç tüketimli, ufak boyutlu bir karttır.

424 kbit/s haberleşme hızına sahiptir. RFID üzerinde farklı şifreleme türlerini desteklemektedir. Desteklediği kart türleri mifare1 S50, mifare1 S70 mifare ultralight, mifare pro ve mifare desfire'dir.

Ayrıntılar:[https://www.robotistan.com/rc522-rfid-nfc-kiti-rc522-rfid-nfc-modulu-kart-ve-anahtarlik-kiti-1356?language=tr&h=b5d0e742&gclid=Cj0KCQjw6ar4BRDnARIsAITGzIA1YcZgnZ3cGUCV1L0lwWUDpwhNnJQ92BNrnGMpBcBWozqBaTAhoQaAkRJEALw\\_wcB](https://www.robotistan.com/rc522-rfid-nfc-kiti-rc522-rfid-nfc-modulu-kart-ve-anahtarlik-kiti-1356?language=tr&h=b5d0e742&gclid=Cj0KCQjw6ar4BRDnARIsAITGzIA1YcZgnZ3cGUCV1L0lwWUDpwhNnJQ92BNrnGMpBcBWozqBaTAhoQaAkRJEALw_wcB)



```
1 #include <MFRC522.h>
2 #include <Servo.h>
3 #include <SPI.h>
4
5 int RST_PIN = 9;
6 int SS_PIN = 10;
7 int servoPin = 8;
8
9 Servo motor;
10 MFRC522 rfid(SS_PIN, RST_PIN);
11 byte ID[4] = {218, 5, 185, 37};
12
13 void setup() {
14   motor.attach(servoPin);
15   Serial.begin(9600);
16   SPI.begin();
17   rfid.PCD_Init();
18 }
19
20 void loop() {
21
22   if ( ! rfid.PICC_IsNewCardPresent() )
23     return;
24
25   if ( ! rfid.PICC_ReadCardSerial() )
26     return;
27
28   if ( rfid.uid.uidByte[0] == ID[0] &&
29         rfid.uid.uidByte[1] == ID[1] &&
30         rfid.uid.uidByte[2] == ID[2] &&
31         rfid.uid.uidByte[3] == ID[3] ) {
32     Serial.println("Kart okundu");
33     ekranaYazdir();
34     motor.write(90);
35     delay(1000);
36   }
```

### 22 Arduino İle RFID Sensörü Kullanımı

#### Teorik Bilgi

- SPI Master-Slave mantığına dayanan bir seri haberleşme protokolüdür. Yani birbirleriyle senkronize olarak çalışması için aralarında bir clock(saat) sinyaliye ihtiyaç duyarlar. Bu şekilde UART gibi asenkron protokollere göre daha güvenilir bir haberleşme sağlamış oluruz.
- RFID'nin açılımı **Radio Frequency Identification** yani radyo frekansı ile tanımlamadır. RFID teknolojisi nesnelerin radyo dalgaları kullanılarak tanınması için kullanılan teknolojidir.
- Kullandığımız kartların kendilerine ait UID isimli bir numarası vardır. Bu numara, her kart için farklıdır. Okuyucumuza kartımızı veya anahtarlığımızı yaklaştırdığımızda bu numara okunarak işlem yapılır.

#### Devre Kısmı

- SPI için en az 4 adet pine ihtiyaç duyarız. SCK clock sinyalini için kullanılır. MOSI(Master Out Slave In), master olan cihazdan slave olan cihaza veri göndermek için kullanılır. MISO (Master In Slave Out), slave olan cihazdan master olan cihaza veri göndermek için kullanılır. SS(Slave Select) pini, master cihazın hangi cihaz ile haberleşeceğini belirler

#### Kod Kısmı

- 5.satırda RC522 modülü reset pinini, 6.satırda RC522 modülü chip select pinini tanımlıyoruz.
- 9.satırda Servo motor için değişken oluşturuyoruz.
- 10.satırda RC522 modülü ayarlarını yapıyoruz.
- 11.satırda Yetkili kart ID'sini tanımlıyoruz.
- 14.satırda Servo motor pinini motor değişkeni ile ilişkilendiriyoruz.
- 16.satırda SPI iletişimini başlatıyoruz.
- 17.satırda RC522 modülünü başlatıyoruz.
- 22. Ve 23.satırda Yeni kartın okunmasını bekliyoruz.
- 25. Ve 26.satırda Kart okunmadığı zaman bekliyoruz.
- 28.satırdan 31.satıra kadar Okunan kart ID'si ile ID değişkenini karşılaştırıyoruz.



```
32     Serial.println("Kapi acildi");
33     ekranaYazdir();
34     motor.write(90);
35     delay(1000);
36     motor.write(0);
37     delay(500);
38 }
39 else{
40     Serial.println("Yetkisiz Kart");
41     ekranaYazdir();
42 }
43 rfid.PICC_HaltA();
44 }
45 void ekranaYazdir(){
46     Serial.print("ID Numarasi: ");
47     for(int sayac = 0; sayac<4; sayac++){
48         Serial.print(rfid.uid.uidByte[sayac]);
49         Serial.print(" ");
50     }
51     Serial.println(" ");
52 }
53 }
```

## 23 ESP8266 İle Sıcaklık ve Nem Ölçümü

28 Aralık 2020 Pazartesi 21:26

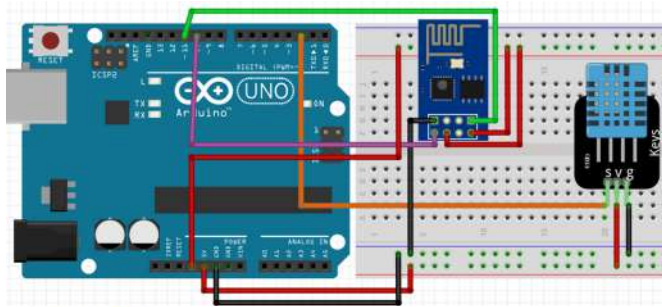
### ESP8266 WiFi Modülü

ESP8266 üzerinde dahili anten bulunmaktadır. Bu sayede ortamdaki Wifi ağına rahatlıkla bağlanabilmekte, veri paketleri alıp gönderebilmektedir.

Ayrıntılar:[https://www.robotistan.com/esp8266-ekonomik-wifi-serial-transceiver-module?language=tr&h=bb3946ee&gclid=Cj0KCQiw6ar4BRDnARIsAITGzICUcsGomYx6NUNjkDizN83\\_9BlqQB4Ez1tnNbzfOTCeDpgl\\_s7gPIaAmg1EALw\\_wcB](https://www.robotistan.com/esp8266-ekonomik-wifi-serial-transceiver-module?language=tr&h=bb3946ee&gclid=Cj0KCQiw6ar4BRDnARIsAITGzICUcsGomYx6NUNjkDizN83_9BlqQB4Ez1tnNbzfOTCeDpgl_s7gPIaAmg1EALw_wcB)

### DHT11 Sıcaklık ve Nem Sensörü

DHT11 sıcaklık ve nem algılayıcı kalibre edilmiş dijital sinyal çıkışı veren gelişmiş bir algılayıcı birimdir. Yüksek güvenilirliktedir ve uzun dönem çalışmalarda dengelidir. 8 bit mikroişlemci içerir, hızlı ve kaliteli tepki verir. 0 ile 50°C arasında 2°C hata payı ile sıcaklık ölçen birim, 20-90% RH arasında 5% RH hata payı ile nem ölçer. Pin GND, 5 pini dijital sinyal çıkış ve ortadaki pin ise 5V gerilim pinidir.



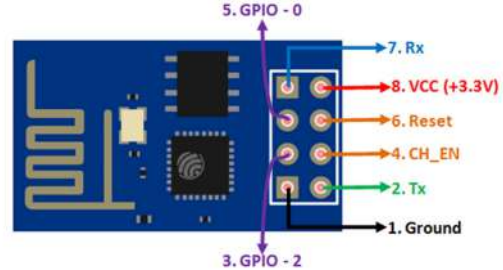
```
1 #include <dht11.h>
2 #include <SoftwareSerial.h>
3
4 String agAdi = "BAYBARS-2019";
5 String agSifresi = " ";
6
7 int rxPin = 10;
8 int txPin = 11;
9 int dht11Pin = 2;
10
11 String ip = "184.106.153.149";
12 float sicaklik, nem;
13
14 dht11 DHT11;
15 SoftwareSerial esp(rxPin, txPin);
16
17 void setup() {
18
19   Serial.begin(9600);
20   Serial.println("Started");
21   esp.begin(115200);
22   esp.println("AT");
23   Serial.println("AT Yollandi");
24   while(!esp.find("OK")){
25     esp.println("AT");
26     Serial.println("ESP8266 Bulunamadi.");
27   }
28   Serial.println("OK Komutu Alindi");
29   esp.println("AT+CMODE=1");
30   while(!esp.find("OK")){
31     esp.println("AT+CMODE=1");
32     Serial.println("Ayar Yapilmiyor...");
33   }
34   Serial.println("Client olarak ayarlandi");
35   Serial.println("Aga Baglaniliyor...");
36   esp.println("AT+CWJAP=\""+agAdi+"\", \""+agSifresi+"\"");
37   while(!esp.find("OK")){
38     Serial.println("Aga Baglandi.");
39     delay(1000);
40   }
41
42   void loop() {
43     esp.println("AT+CIPSTART=\"TCP\", \""+ip+"\", 80");
44     if(esp.find("Error")){
45       Serial.println("AT+CIPSTART Error");
46     }
47     DHT11.read(dht11Pin);
48     sicaklik = (float)DHT11.temperature;
49     nem = (float)DHT11.humidity;
50     String veri = "GET https://api.thingspeak.com/update?api_key=UFWBDRGHSOZ34QHP";
51     veri += "&field1=";
52     veri += String(sicaklik);
53     veri += "&field2=";
54     veri += String(nem);
55     veri += "\n\n";
56     esp.print("AT+CIPSEND=");
57     esp.println(veri.length()+2);
58     delay(2000);
59     if(esp.find(">")){
60       esp.print(veri);
61       Serial.println(veri);
62       Serial.println("Veri gonderildi.");
63       delay(1000);
64     }
65   }
66 }
```

## 23 ESP8266 İle Sıcaklık ve Nem Ölçümü

### Teorik Bilgi

- Thingspeak, açık kaynaklı bir IoT(Internet of Things) uygulamasıdır. Kullanıcılar HTTP üzerinden siteye veri gönderip kendi oluşturdukları uygulamaları site üzerinde bulunan grafik arayüzleri sayesinde daha görsel ve anlaşılması kolay bir hale getirir.
- ESP8266, üzerinde bulunan kablosuz haberleşme devresi sayesinde ethernet protokolü ile kablosuz internete bağlanmamızı sağlar. Ethernet protokolünü basitleştirip UART protokolüne dönüştüren bir tercüman görevi görmektedir.
- DHT11, nem ve sıcaklık verilerini okumamıza yarayan bir sensördür. Sensörün içinde bir NTC(Negative Temperature Coefficient) bulunmaktadır. NTC, bir çeşit dirençtir ve ortamın ısısı arttıkça iletkenliği artar ve direnç değeri düşer. Sensörün içinde 2 adet elektrot ve elektrotların arasında ise havadaki nemi tutan bir yüzey bulunmaktadır. Bu yüzey havadaki nem miktarı arttıkça elektrotlar arasındaki iletkenliği değiştirir. Bu şekilde havadaki nem düzeyini ölçmüş oluruz.

### Devre Kısmı



- ESP8266 modülü, Arduino ile seri haberleşmeyle iletişim kurar. Bu iletişim Arduino Uno'nun 0. ve 1. pinleriyle donanımsal olarak yapılabilir ancak SoftwareSerial kütüphanesi ile diğer pinlerden yazılımsal olarak yapılabilir.
- ESP8266 bazı durumlarda arduino'dan enerji aldığı anda eğer voltaj yetersiz geldiğinde sensörde bazı bağlantı ya da haberleşme problemleri olabilir. Bu problemleri engellemek için sensörün artı ve eksi bacakları arasında 100uF kondansatör bağlanarak voltaj dalgalanmalarını önüne geçebilir.

### Kod Kısmı

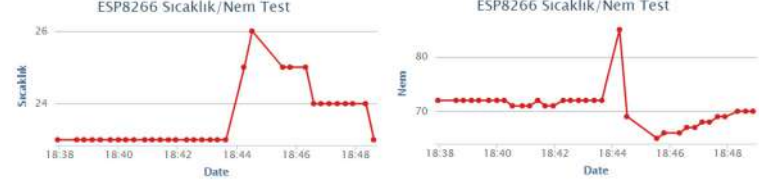
- 11.satırda Thingspeak ip adresini yazıyoruz.
- 15.satırda Seri haberleşme pin ayarlarını yapıyoruz.
- 21.satırda ESP8266 ile seri haberleşmeyi başlatıyoruz.
- 22.satırda AT komutu ile modül kontrolünü yapıyoruz.
- 24.satır ile Modül hazır olana kadar bekliyoruz.
- 29.satırda ESP8266 modülünü client olarak ayarlıyoruz.
- 30.satırda Ayar yapıldıya kadar bekliyoruz.
- 36.satır ile Ağımıza bağlanıyoruz.
- 43.satır ile Thingspeak'e bağlanıyoruz.
- 50.satırda Key kısmına kendi api keyimizi yazıyoruz.
- 52. ve 54.satırda Göndereceğimiz sıcaklık ve nem değişkeni
- 56.satırda ESP'ye göndereceğimiz veri uzunluğunu veriyoruz.
- 59.satırda ESP8266 hazır olduğunda içindeki komutlar çalışıyor.
- 66.satırda Bağlantıyı kapatıyoruz
- Verilerin internet üzerinde görüntülenmesi için Thingspeak platformunu kullanıyoruz. Thingspeak platformuna <https://www.thingspeak.com> adresinden üye oluyoruz. Üye olduktan sonra "Channels" sekmesinden "My Channels" bölümüne giriyoruz. Ardından "New Channel" butonuna tıklıyoruz. Gelen sayfada gerekli ayarları yukarıdaki şekilde yapıyoruz. Daha sonra sayfanın en altında bulunan "Save Channel" butonuna tıklayarak yeni kanalımızı oluşturuyoruz. Arduino'nun Thingspeak ile

```

59 if(esp.find(">")){
60   esp.print(veri);
61   Serial.println(veri);
62   Serial.println("Veri gonderildi.");
63   delay(1000);
64 }
65 Serial.println("Baglanti Kapatildi.");
66 esp.println("AT+CIPCLOSE");
67 delay(1000);
68 }

```

Oluyoruz. Öye olduktan sonra Channels sekmesinden My Channels bölümüne giriyoruz. Ardından “New Channel” butonuna tıklıyoruz. Gelen sayfada gerekli ayarları yukarıdaki şekilde yapıyoruz. Daha sonra sayfanın en altında bulunan “Save Channel” butonuna tıklayarak yeni kanalımızı oluşturuyoruz. Arduino'nun Thingspeak ile haberleşmesi için “api key”e ihtiyaç vardır. Arduino, “api key” ile Thingspeak hesabınıza bağlanarak verileri kanalınıza kaydeder. “Api key”e ulaşmak için “Api Keys” sekmesine tıklıyoruz. Gelen sayfadaki “Write API Key” bölümündeki “Key”i kopyalayarak Arduino kodumuzdaki gerekli yere yapıştırıyoruz. Kodumuzu Arduino'ya yükledikten sonra Arduino IDE üzerinden seri portu açıyoruz. Seri portta veri gönderildi yazısını gördükten sonra thingspeak üzerinden verilerimizi görüntülüyoruz. “Private View” sekmesine tıklayarak Arduino tarafından gönderilen verileri grafikler üzerinde görüntüleyebiliyoruz.



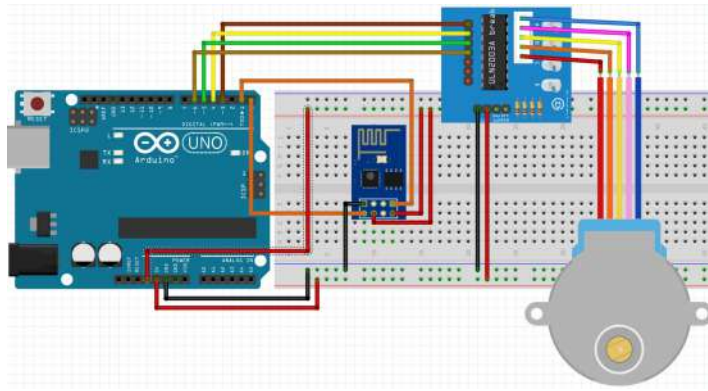
## 24 ESP8266 İle Step Motor Kontrolü

28 Aralık 2020 Pazartesi 21:26

### 28 BYJ Redüktörlü Step Motor ve ULN2003A Step Motor Sürücü Kartı

Kart üzerinde 4 led ve 4 kontrol pini bulunmaktadır. Bu pinler sayesinde step motorun sürülmesini, Arduino veya herhangi bir mikrodenetleyici ile gerçekleştirebilirsiniz.

Ayrıntılar:[https://www.robotistan.com/28-byj-48-reduktorlu-step-motor-ve-uln2003a-step-motor-surucu-karti?language=tr&h=75b29baa&gclid=Cj0KCQjw6ar4BRDnARIsAITGzIC2hDIQwUgs3ulNeNv3nfvTW15ZBy8Ta0Zn-bbGC1iv0jwx2LLt7QaAIIQeALw\\_wcB](https://www.robotistan.com/28-byj-48-reduktorlu-step-motor-ve-uln2003a-step-motor-surucu-karti?language=tr&h=75b29baa&gclid=Cj0KCQjw6ar4BRDnARIsAITGzIC2hDIQwUgs3ulNeNv3nfvTW15ZBy8Ta0Zn-bbGC1iv0jwx2LLt7QaAIIQeALw_wcB)



```
1 String agAdi = "BAYBARS-2019";
2 String agSifresi = " ";
3
4 int motorPin1 = 3, motorPin2 = 4, motorPin3 = 5, motorPin4 = 6;
5
6 void setup() {
7   pinMode(motorPin1, OUTPUT);
8   pinMode(motorPin2, OUTPUT);
9   pinMode(motorPin3, OUTPUT);
10  pinMode(motorPin4, OUTPUT);
11  Serial.begin(115200);
12  Serial.println("AT");
13  while(!Serial.find("OK")) {
14    Serial.println("AT");
15  }
16  delay(1000);
17  Serial.println("AT+RST");
18  delay(1000);
19  while(!Serial.find("OK")) {
20    delay(1000);
21    Serial.println("AT+CMODE=1");
22    while(!Serial.find("OK")) {
23      Serial.println("AT+CWJAP=\"" + agAdi + "\",\"" + agSifresi + "\"");
24      while(!Serial.find("OK")) {
25        Serial.print("AT+CIFSR\r\n");
26        Serial.print(espOkuma(1000));
27        serialTemizle(2000);
28        Serial.print("AT+CIPMUX=1\r\n");
29        serialTemizle(2000);
30        Serial.print("AT+CIPSERVER=1,80\r\n");
31        serialTemizle(2000);
32      }
33    }
34  }
35  void loop() {
36    if(Serial.available()) {
37      if(Serial.find("+IPD,") {
38        delay(200);
39        int connectionId = Serial.read() - 48;
40        String komut = espOkuma(1000);
41        if(komut.indexOf("step=ileri") != -1) {
42          for(int adim = 0; adim < 50; adim++) {
43            stepIleri(50);
44          }
45        } else if(komut.indexOf("step=geri") != -1) {
46          for(int adim = 0; adim < 50; adim++) {
47            stepGeri(50);
48          }
49        }
50        String sayfa = "<h1>Step Motor Kontrol</h1><br>";
51        sayfa+="  
<a href='\"?step=ileri\"'><button><h1>İleri</h1></button></a>";
52        sayfa+="  
<a href='\"?step=geri\"'><button><h1>Geri</h1></button></a>";
53        komut = "AT+CIPSEND=";
54        komut += connectionId;
55        komut += ",";
56        komut += sayfa.length();
```

### 24 ESP8266 İle Step Motor Kontrolü

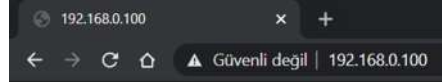
#### Teorik Bilgi

- Step motorlar, elektrik enerjisini dönme hareketi ile fiziksel enerjiye çeviren elektromekanik aygıtlardır. Adım adım hareket eden motorlardır. İçinde bulunan bobinlerin sırayla açılıp kapanması ile hareketlerini sürdürürler. Step motor 4 fazlı olup içinde 4 adet bobin bulunmaktadır. Step motorlar, çok yüksek hızlı anahtarlayabilen motor sürücülerle ve motor kontrol kartları ile kontrol edilirler.

#### Devre Kısım

#### Kod Kısım

- 17.satırda ESP8266'yı resetliyoruz.
- 30.satırda Server oluşturuyoruz ve 80. porttan dinlemeye başlıyoruz.
- 70. ve 81.satırlar arası ESP'den gelen dönütleri okur.
- 83. ve 90.satırlar arası Seri haberleşmede kullanmadığımız byteleri temizler.
- Kodlarımızı Arduino'ya atmadan önce ESP8266 modülünün TX ve RX pinlerinin Arduino'dan sökülmesi gerekmektedir. Arduino programlanırken ESP8266 modülünün TX ve RX pinlerinin bağlı olması, Arduino ile Bilgisayar arasındaki iletişimi engelleyerek Arduino'nun programlanmasını engellemektedir.
- Arduino IDE üzerinden seri portumuzu açarak veri hızını 115200 olarak ayarlıyoruz. ESP8266 modülü ile iletişimi buradan görebiliriz. Ayarlarlar yapıldığında seri porttaki IP adresine bir web tarayıcı ile bağlanıyoruz.



## Step Motor Kontrol

İleri

Geri

```

52     sayfa+="  
<a href='\"?step=geri\"><button><hl>Geri</hl></button></a>";
53     komut = "AT+CIPSEND=";
54     komut += connectionId;
55     komut += ",";
56     komut +=sayfa.length();
57     komut += "\r\n";
58     Serial.print(komut);
59     delay(1000);
60     Serial.print(sayfa);
61     delay(1000);
62     komut = "AT+CIPCLOSE=";
63     komut+=connectionId;
64     komut+="\r\n";
65     //Serial.print(komut);
66 }
67 }
68 }
69
70 String espOkuma(long int zamanAsimi){
71     long int baslangic = millis();
72     String gelen;
73     while(millis() - baslangic < zamanAsimi){
74         if(Serial.available()>0){
75             char c = Serial.read();
76             gelen += c;
77         }
78     }
79     gelen.replace("AT+", "");
80     return gelen;
81 }
82
83 void serialTemizle(long int zamanAsimi){
84     long int baslangic = millis();
85     while(millis() - baslangic < zamanAsimi){
86         if(Serial.available()>0){
87             Serial.read();
88         }
89     }
90 }
91
92 void stepIleri(int beklemeSuresi){
93     digitalWrite(motorPin1, HIGH);
94     digitalWrite(motorPin2, LOW);
95     digitalWrite(motorPin3, LOW);
96     digitalWrite(motorPin4, LOW);
97     delay(beklemeSuresi);
98     digitalWrite(motorPin1, LOW);
99     digitalWrite(motorPin2, HIGH);
100    digitalWrite(motorPin3, LOW);
101    digitalWrite(motorPin4, LOW);
102    delay(beklemeSuresi);
103    digitalWrite(motorPin1, LOW);
104    digitalWrite(motorPin2, LOW);
105    digitalWrite(motorPin3, HIGH);
106    digitalWrite(motorPin4, LOW);
107    delay(beklemeSuresi);
108    digitalWrite(motorPin1, LOW);
109    digitalWrite(motorPin2, LOW);
110    digitalWrite(motorPin3, LOW);
111    digitalWrite(motorPin4, HIGH);
112    delay(beklemeSuresi);
113 }
114
115 void stepGeri(int beklemeSuresi){
116     digitalWrite(motorPin1, LOW);
117     digitalWrite(motorPin2, LOW);
118     digitalWrite(motorPin3, LOW);
119     digitalWrite(motorPin4, HIGH);
120     delay(beklemeSuresi);
121     digitalWrite(motorPin1, LOW);
122     digitalWrite(motorPin2, LOW);
123     digitalWrite(motorPin3, HIGH);
124     digitalWrite(motorPin4, LOW);
125     delay(beklemeSuresi);
126     digitalWrite(motorPin1, LOW);
127     digitalWrite(motorPin2, HIGH);
128     digitalWrite(motorPin3, LOW);
129     digitalWrite(motorPin4, LOW);
130     delay(beklemeSuresi);
131     digitalWrite(motorPin1, HIGH);
132     digitalWrite(motorPin2, LOW);
133     digitalWrite(motorPin3, LOW);
134     digitalWrite(motorPin4, LOW);
135     delay(beklemeSuresi);
136 }

```