

# ARDUNIO DERS NOTLARI

**Arduino:** Arduino, interaktif projeler geliřtirmek için tasarlanan, elektronik donanım ve yazılım temelli bir geliřtirme platformudur. Arduino kolay bir şekilde çevresiyle etkileřime girebilen sistemler tasarlanabilir açık kaynaklı bir geliřtirme platformudur.

Arduino kütüphaneleri sayesinde kolaylıkla programlanabilir. Analog ve dijital giriřleri sayesinde analog ve dijital veriler işlenebilir.<sup>1</sup>



## ARDUNIO KOMUTLARI

### Void Setup ( )

Kullanmış olduğumuz pinler tanımlanır ve gerekli başlama komutları verilir. Bu kısım arduino güç aldığında ya da resetlendiğinde bir kere çalışmaktadır.

### Void Loop ( )

Void Loop kısmında ise devamlı olarak çalışmasını istediğimiz pinlerin aktif ya da pasif olması, sensörlerden gelen verileri okuma ve işleme gibi komutları yukarıdan aşağıya doğru döngüsel olarak işleyen kısımdır.

### PinMode(,)

Burada pinimizin modunu yani giriş pini mi çıkış pini mi olduğunu belirtiyoruz. Bunun için ilk parametre pini ikinci parametre de ise modu belirtiyoruz

( INPUT: GİRİŞ | OUTPUT: ÇIKIŞ )

**digitalWrite(,)** Bu komut ile digital OUTPUT (yani çıkış) olan pinimize güç verebilmekteyiz. Bunun için ilk parametre pini ikinci parametre de ise güç modunu belirtiyoruz (HIGH: Aktif | LOW:Pasif )

**Delay ( )** Bekleme için kullandığımız bir komuttur ve parantez içine yazılan değer milisaniye cinsinde olur.<sup>2</sup>

**digitalRead ( )** digitalRead fonksiyonu belirtilen pindeki gerilim değerine göre 1 ( true veya HIGH ) veya 0 ( false veya LOW ) değerlerini döndürür. Eğer belirtilen pinde gerilim varsa 1, yoksa 0 döndürür.<sup>3</sup>

**analogRead:** Analog pinden değeri okur. Kullanımı: analogRead(pin);

**analogWrite:** Analog değer yazmak için PWM pinleri kullanılır. PWM pinleri arduino da " ~ " işareti ile gösterilmektedir. 3,5,6,9,10,11 numaralı pinler Arduino Uno için PWM pinleridir. Örneğin bir motoru yavaş veya hızlı sürmek için PWM pinleri kullanılabilir. Işık şiddetini ayarlamak için de.<sup>4</sup>

**map:** Map komutu ile dönüřtürmek istediğimiz değışkeni ve dönüřecek iki aralığı aynı satırda belirtiyoruz.

map (Değişken Adı, Dönüřtürülecek en düşük değer, Dönüřtürülecek en yüksek değer, Dönüřtürülmüş en düşük değer, Dönüřtürülmüş en yüksek değer);

```
yeniDeger=map(değer, 0,1024,0,255);
```

*// burada değer değışkeninin 0-1024 arasındaki değerini oranlayarak 0-255 arasına dönüřtürür ve yeniDeger değışkenine aktarır.*

**FOR Döngüsü:** Bir değışkene belirlenen bir değerden, belirlenen başka bir değere kadar, belirlenmiş bir değışim miktarı ile yeni değerler atayan döngüdür.

```
for (int n=10;n>5;n--) // n değışkeni 5'ten büyük olduğu sürece, 10'dan başlayarak birer azalan değerler alır. Alacağı değerler: 10, 9, 8, 7, 6 olur
```

```
for (int n=6;n<11;n++) //n değışkeni 11'den küçük olduğu sürece, 6'dan başlayarak birer artan değerler alır. Alacağı değerler: 6,7,8,9,10 olur
```

**// işareti:** Kodların yanına açıklama (dahili dokümantasyon) yazmak için kullanılır.

<sup>1</sup> Wikipedi

<sup>2</sup> [https://ogrencibloglari.net/arduino-nedir-6-adimda-arduino-kodlama/#Void\\_Setup](https://ogrencibloglari.net/arduino-nedir-6-adimda-arduino-kodlama/#Void_Setup)

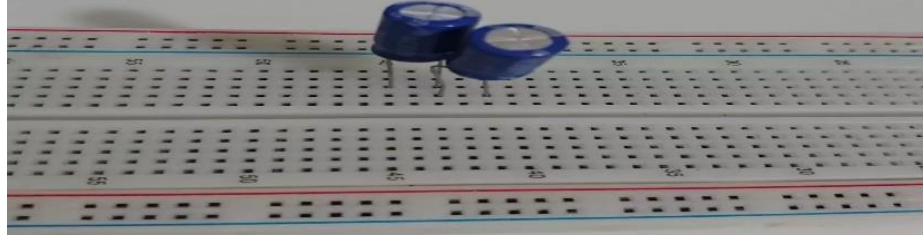
<sup>3</sup> <https://www.mobilhanem.com/arduino-dersleri-dijital-pinler-ve-dijital-io-fonksiyonlari/>

<sup>4</sup> <https://ctrlbizde.com/index.php/egitimler/arduino-programlama/item/620-analog-giris-ve-cikis-komutlari-analogread-analogwrite-arduino-programlama-6>

## DONANIMLAR:

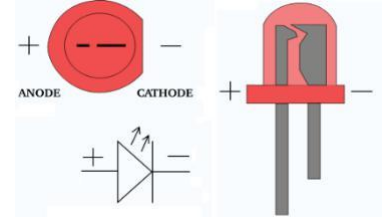
### Breadboard:

Breadboard üzerinde devrelerimizi test ettiğimiz araçtır. Kurduğumuz devreleri birbirlerine lehimlemeden kolaylıkla test etmemizi sağlar.<sup>5</sup>



### LED:

("Light Emitting Diode", Işık Yayan Diyot), yarı-iletken, diyot temelli, ışık yayan bir elektronik devre elemanıdır.<sup>6</sup>



### Direnç:

Ohm kanununa göre uçları arasında gerilim düşümüne sebep olan devre elemanıdır.<sup>7</sup>



### Potansiyometre:

Potansiyometre, dışarıdan fiziksel müdahaleler ile değeri değiştirilebilen dirençlerdir. Potansiyometrelerin daha güçlülerine ve daha yüksek akım değerine sahip devrelerde kullanılanlarına ise reosta denir.<sup>8</sup>



### Buton:

Devremizde on/off kontrolü için kullanılır. Yani biz butonumuza bastığımız zaman on durumuna geçerken elimizi buton üzerinden çektiğimiz zaman off durumuna geçer.<sup>9</sup>



**Buzzer:** Kullanım alanlarına da bağlı olarak alarm, zamanlayıcı, onaylama cevazı gibi işlevlerde kullanılabilirler. Nitekim tanımında da belirttiğimiz üzere, buzzerlar işitsel ikaz cihazı çeşitleridir.



**RGB LED:** RGB (Red-Green-Blue) LED içerisinde kırmızı, yeşil ve mavi renkleri barındıran bir LED çeşididir. Özellikle animasyon ve ışıklandırma sistemlerinde sıklıkla kullanılmaktadır. RGB LED'lerde her renk için belirli aralıklar mevcuttur. Bu aralıklar sayesinde birçok renk elde etmek mümkündür.



**ULTRASONİK Mesafe Sensörü:** Sonar iletişim kullanarak karşısındaki nesneye olan mesafeyi hesaplayan bir kaynaktır. Sonar dediğimiz sistem ses dalgalarını kullanarak cismin uzaklığı hesaplamamıza yardımcı olur.



<sup>5</sup> <https://maker.robotistan.com/breadboard/>

<sup>6</sup> <https://diyot.net/led/>

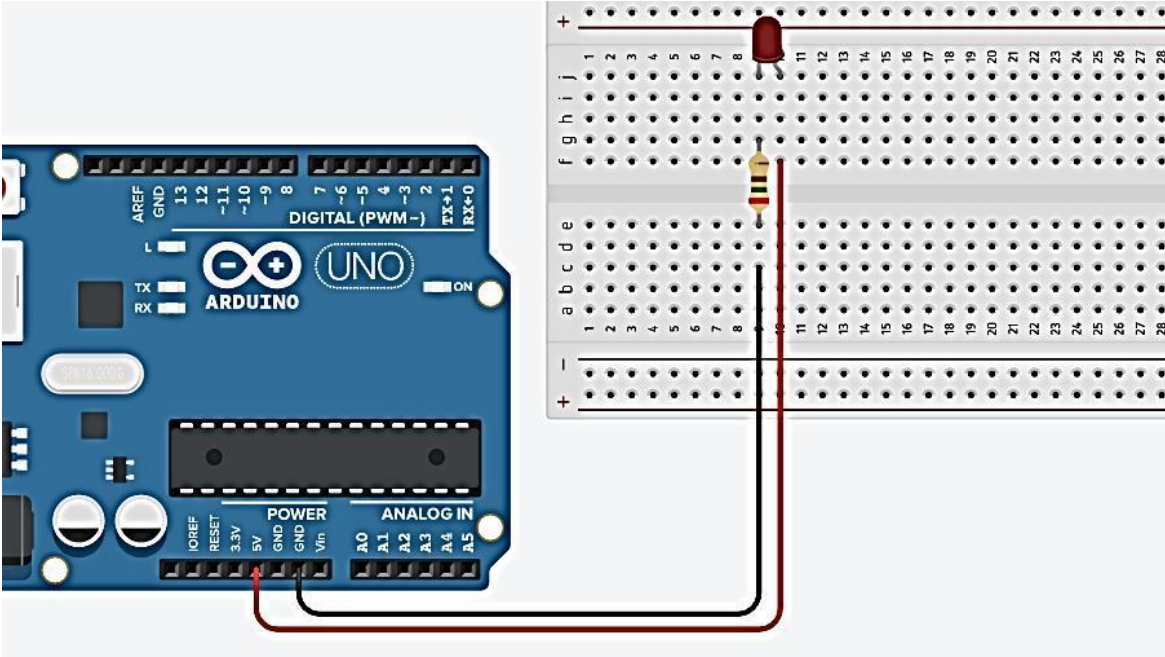
<sup>7</sup> [https://tr.wikipedia.org/wiki/Diren%C3%A7\\_\(devre\\_eleman%C4%B1\)](https://tr.wikipedia.org/wiki/Diren%C3%A7_(devre_eleman%C4%B1))

<sup>8</sup> <https://tr.wikipedia.org/wiki/Potansiyometre>

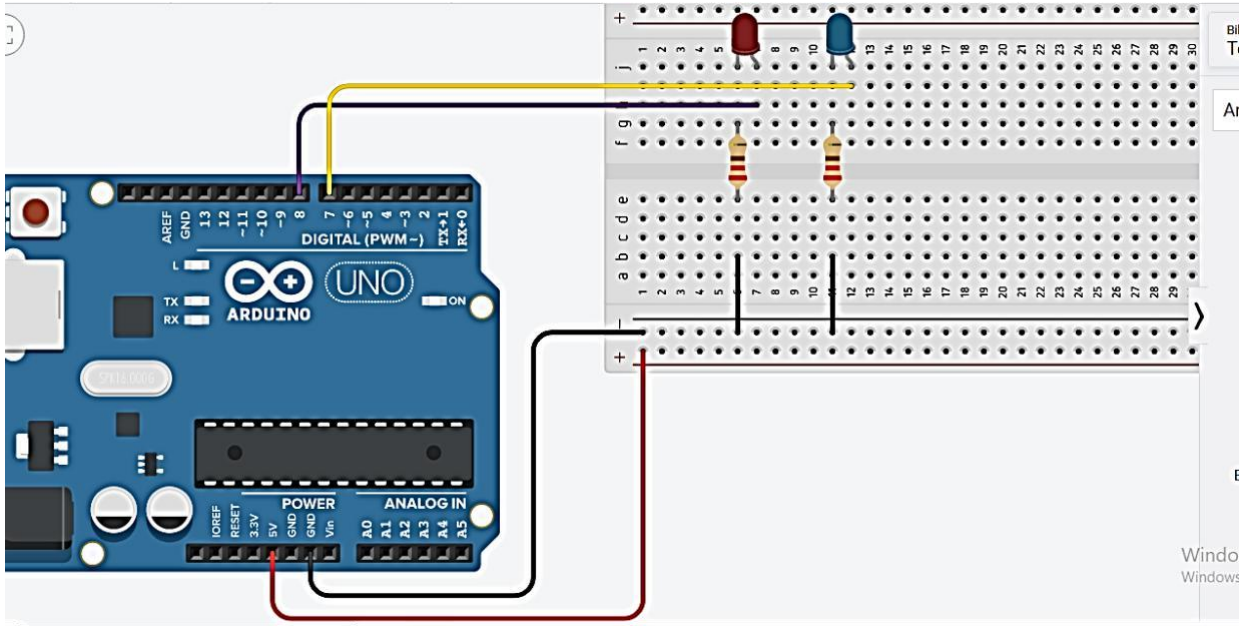
<sup>9</sup> <https://hayaletveyap.com/arduino-ile-push-buton-kullanimi/>

## ÖRNEKLER:

### LED YAKMA DEVRESİ (KODSUZ)



### 2 LED YAKMA DEVRESİ ve KODLARI



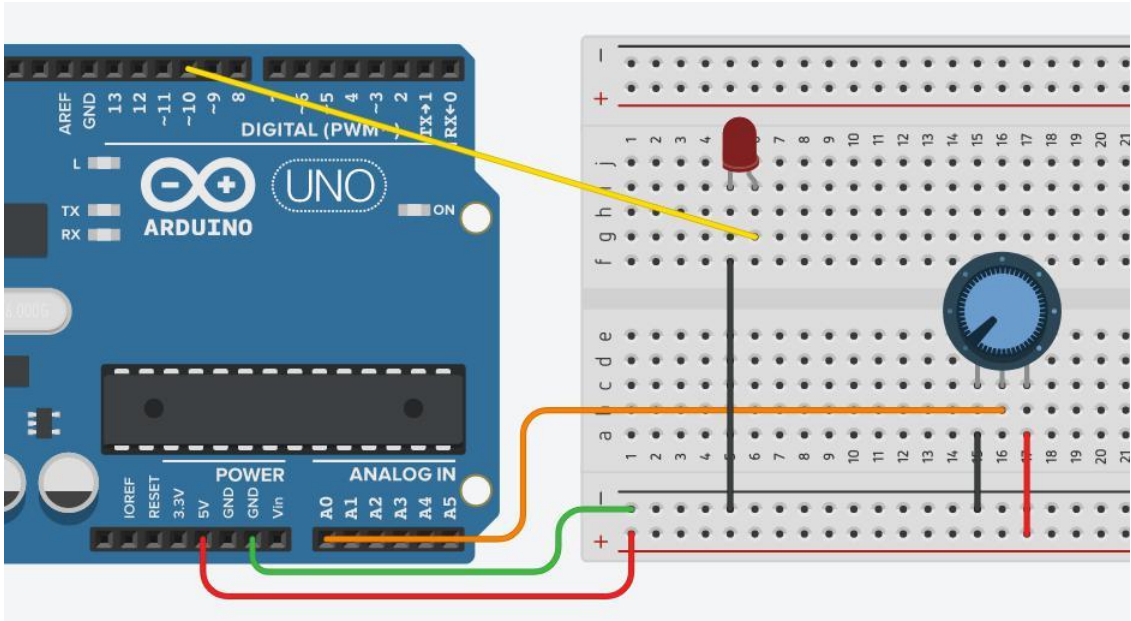
```
void setup()
{
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);
}
```

```
void loop()
{
  digitalWrite(8, HIGH);
  digitalWrite(7, LOW);
```

```
  delay(1000);
  digitalWrite(8, LOW);
  digitalWrite(7, HIGH);
  delay(1000);
}
```

Bu kodlarla ledlerin birisi yanarken diğeri söner, 1000 milisaniye bekledikten sonra diğeri yanar ve ilk led söner.

### 3 Potansiyometre Devresi



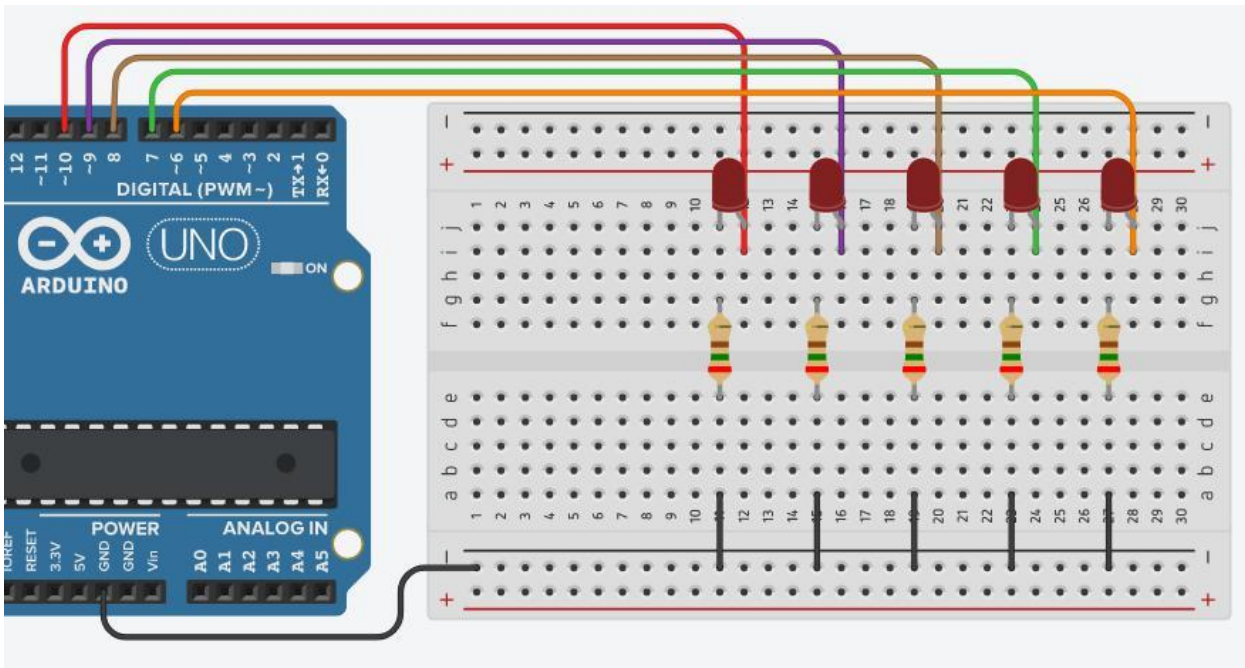
```
void setup()
```

```
{  
  pinMode(10, OUTPUT);  
  pinMode(A0, INPUT);  
}
```

```
void loop()
```

```
{  
  int deger_ilk=analogRead(A0);  
  int deger=map(deger_ilk,0,1024,0,255);//map komutu belirli degerleri bizim istedigimiz araliga  
  oranlar analogWrite(10, deger);  
}
```

### 4 Karaşımşek Devresi





```

void setup()
{
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
}

```

```

void loop()
{
  for (int n=10;n>5;n--){
    digitalWrite(n, HIGH);
    delay(100);
    digitalWrite(n,LOW);
  }
  delay(100);
  for (int n=6;n<11;n++){
    digitalWrite(n,HIGH);
    delay(100);
    digitalWrite(n,LOW);
  }
}

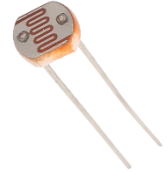
```

//for döngüsü, n değişkeni 10 dan başlayarak birer azalacak şekilde değerler alır, bunu n değeri 5 den büyük olduğu sürece yapar

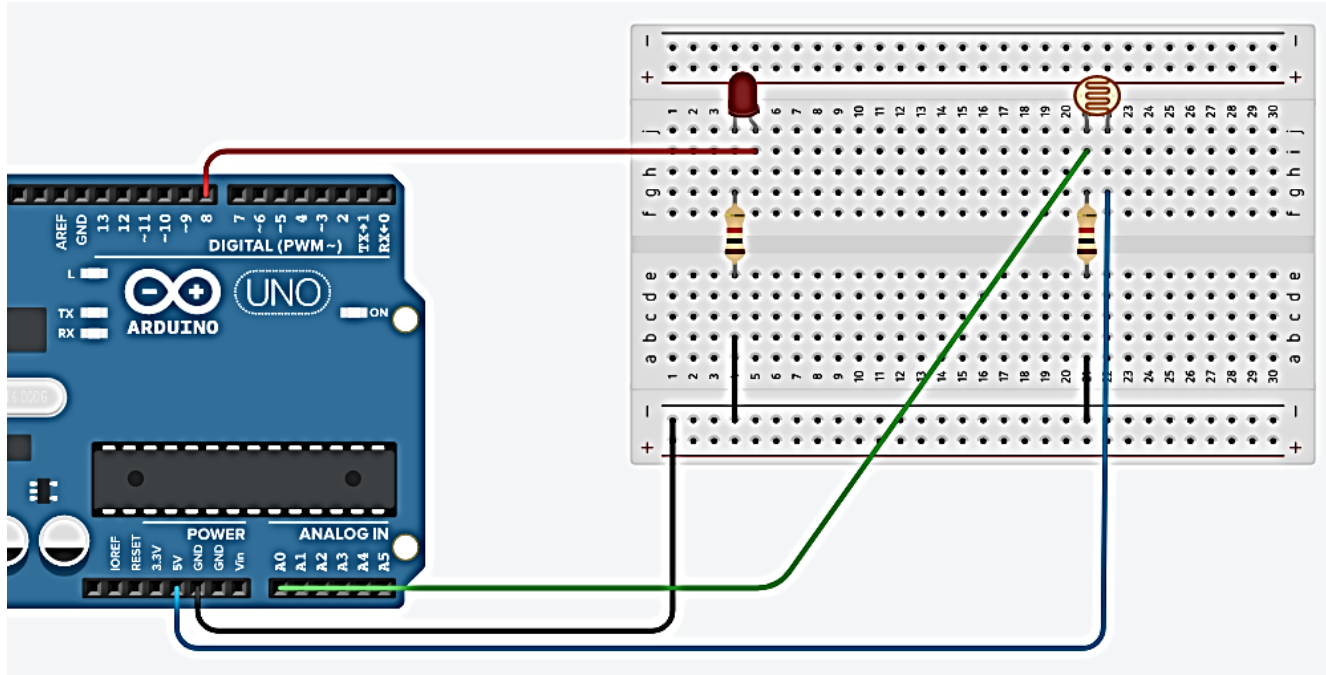
// n değişkeni 6 dan başlayarak 11 den küçük olduğu sürece //birer artarak değerler alır

## 2. DÖNEM KONULARI

**LDR Işık Sensörü:** Bir direnç çeşidi olmakla birlikte Ldr sensör aynı zamanda pasif bir sensör olarak nitelendirilmektedir. Işığın şiddetine göre bize sayısal bir değer verir.



### IŞIK SENSÖRÜ DEVRESİ ve KODLARI

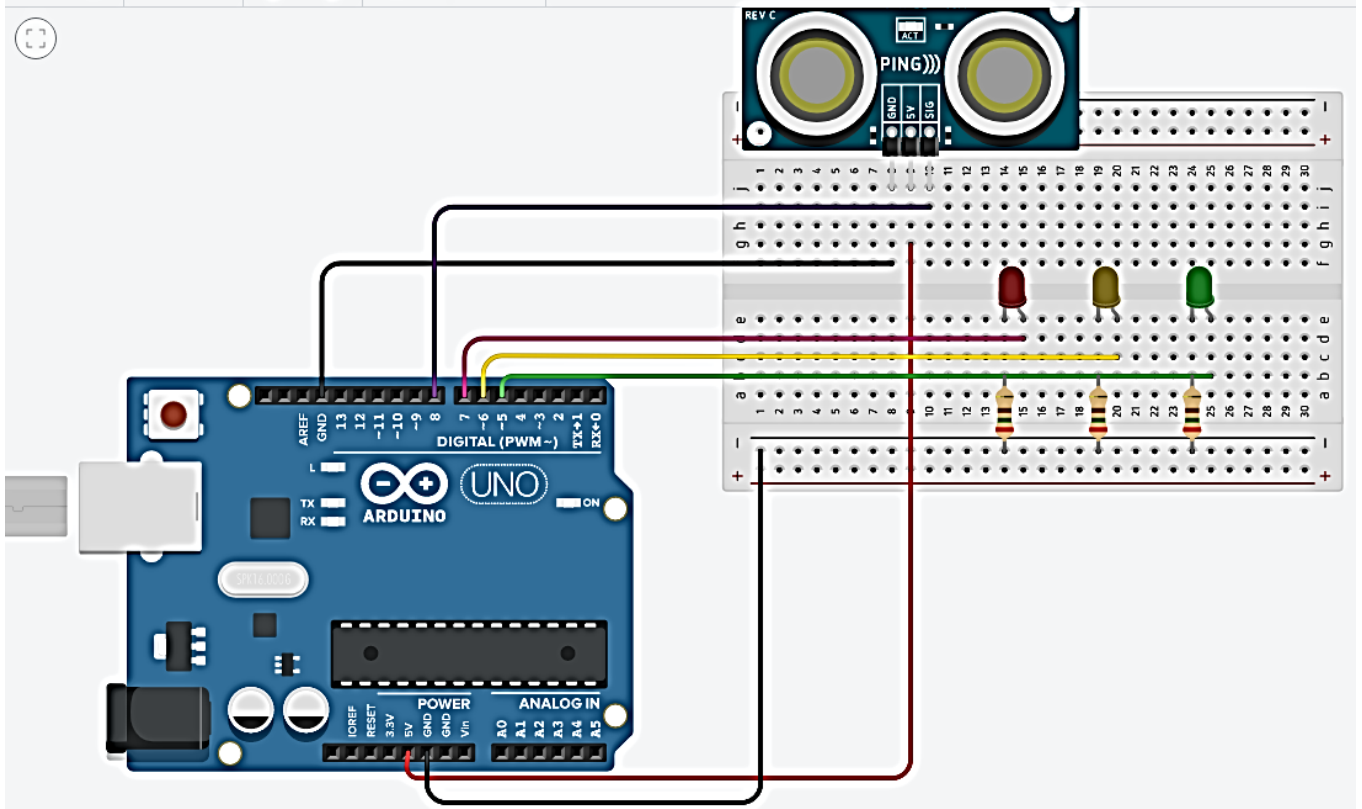


```

void setup()
{
  pinMode(8, OUTPUT);    //8 nolu pin veri çıkışı için tanımlanır
  pinMode(A0, INPUT);    // A0 pini veri girişi için tanımlanır
  Serial.begin(9600);    // seri port başlatılır,
                        //seri porttan sensör verileri okunabilir
}
void loop()
{
  int light=analogRead(A0); //A0 dan alınan veri light değişkenine aktarılır
  Serial.println(light);    //light değişkeninin değeri seri porta yazdırılır
  delay(100);
  if (light>200){          // eğer light değişkeni değeri 200 den büyükse
    digitalWrite(8,HIGH); // 8 nolu porta enerji ver
  }
  if (light<200){         // eğer light değişkeni değeri 200 den küçükse
    digitalWrite(8,LOW); // 8 nolu porta enerji verme
  }
}

```

## MESAFE SENSÖRÜ DEVRESİ ve KODLARI



## KODLAR;

```
int mesafe=0; // deęişkenler bu bölümde de tanımlanabilir, süre ve mesafe deęişkenleri tam sayı  
int süre=0; // olarak tanımlanmış ve ilk deęer olarak 0 girilmiş
```

```
void setup()
```

```
{  
  pinMode(5,OUTPUT);  
  pinMode(6,OUTPUT);  
  pinMode(7,OUTPUT);  
  Serial.begin(9600);  
}
```

```
void loop()
```

```
{  
  pinMode(8,OUTPUT); //mesafe sensöründen sinyal göndermek için tetikleme süreci başlıyor  
  digitalWrite(8,HIGH); // ses gönderiliyor  
  delayMicroseconds(1000);  
  digitalWrite(8, LOW); // ses gönderimi duruyor
```

```
  pinMode(8, INPUT); //sesin engele çarpıp gelmesini duymak için veri girişı moduna geçiliyor  
  süre=pulseIn(8,HIGH); //pulseIn komutu ile sesin gidip gelme süresi "süre" deęişkenine aktarılıyor  
  mesafe=(süre/2)/29,1; //sesin gidip gelme süresinden engelin mesafesi hesaplanıyor
```

```
  Serial.print("Mesafe= "); // süre bilgileri seri porta yazılıyor  
  Serial.print(mesafe);  
  Serial.println(" cm");
```

```
  if (mesafe<150){ //süre 150 cm den yakınsa 5 nolu pindeki led yanıyor  
    digitalWrite(5,HIGH);  
    digitalWrite(6,LOW);  
    digitalWrite(7,LOW);  
  }
```

```
  if (mesafe<100){ //süre 100 cm den yakınsa 6 nolu pindeki led yanıyor  
    digitalWrite(6,HIGH);  
    digitalWrite(5,LOW);  
    digitalWrite(7,LOW);  
  }
```

```
  if (mesafe<50){ //süre 50 cm den yakınsa 7 nolu pindeki led yanıyor  
    digitalWrite(7,HIGH);  
    digitalWrite(6,LOW);  
    digitalWrite(5,LOW);  
  }
```

```
  delay(100);
```

```
}
```

NOT: TINKERCAD deki mesafe sensörü 3 bacaklıdır. Trigger ve Echo aynı pinden kullanılır. Eđer 4 bacaklı mesafe sensörü kullanılıyorsa Echo pini ve Trigger pini ayrı deęerlendirilir.