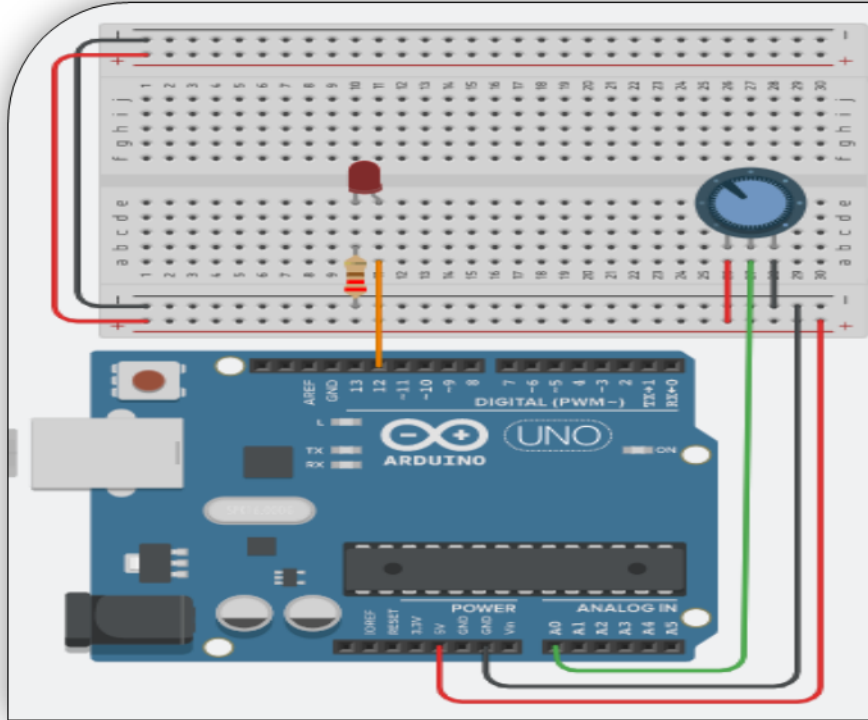
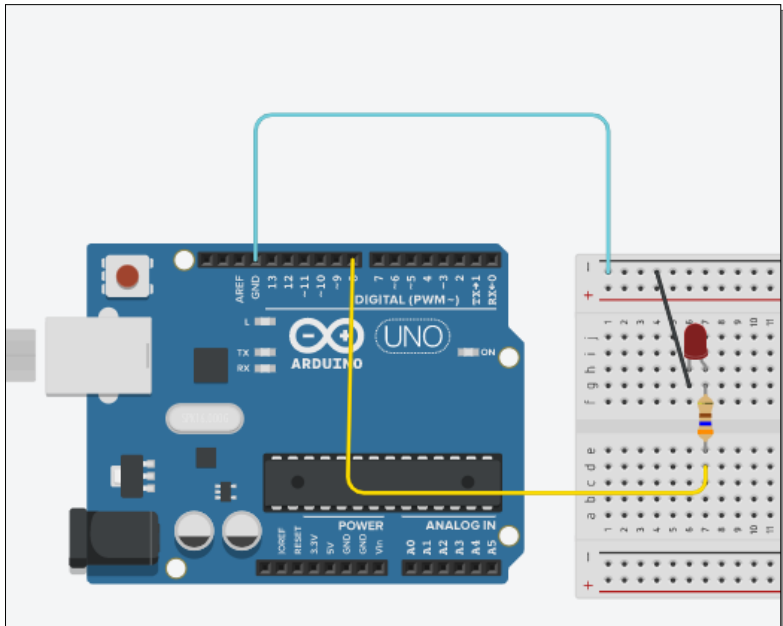


POTANSİYOMETRE İLE İSTENİLEN DEĞERDE LEDİN YANMASINI SAĞLAMAK



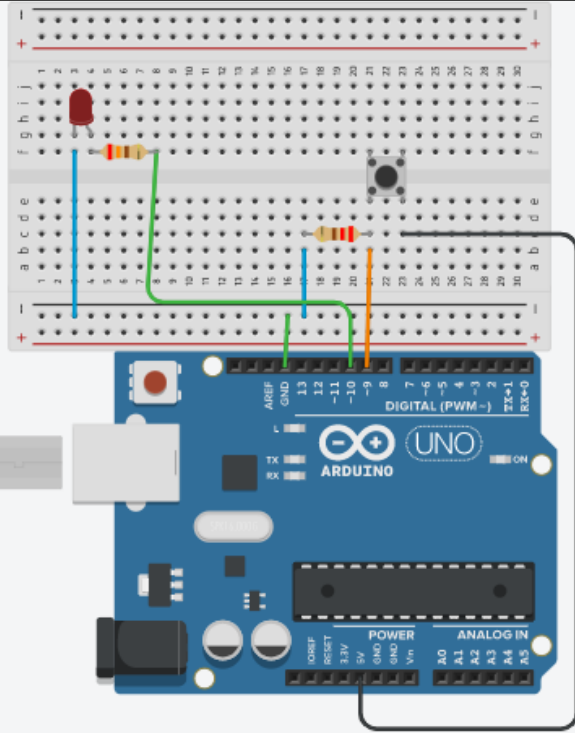
```
Metin
1  int pot=A0;
2  void setup()
3
4  {
5  pinMode(12, OUTPUT); }
6
7  void loop()
8
9  {
10
11     if (analogRead(pot)>=512)
12     {
13         digitalWrite(12, HIGH);
14     }
15     else
16     {
17         digitalWrite(12, LOW);
18     }
19 }
```

BİR SANİYE ARALIKLARLA LED YANSIN SÖNSÜN



```
1  void setup()
2  {
3     pinMode(8, OUTPUT);
4  }
5
6  void loop()
7  {
8     digitalWrite(8, HIGH);
9     delay(1000);
10    digitalWrite(8, LOW);
11    delay(1000);
```

BUTONA BASINCA LED YANSIN



```
1 #define Buton 9
2 #define Led 10
3 void setup()
4 {
5   pinMode(Buton, INPUT);
6   pinMode(Led, OUTPUT);
7 }
8
9 void loop()
10 {
11   if (digitalRead(Buton) == 1)
12     digitalWrite(Led, HIGH);
13   else
14     digitalWrite(Led, LOW);
15 }
```

digitalRead=Belirtilen dijital pindeki değerin yüksek(1) mi alçak(0) mı olduğunu okur.

digitalWrite= Belirtilen dijital pindeki değerin yüksek(1) ya da alçak(0) değer gönderir.

pinMode=Digital pinlerin mutlaka setup kısmında giriş ya da çıkış olduğu tanımların

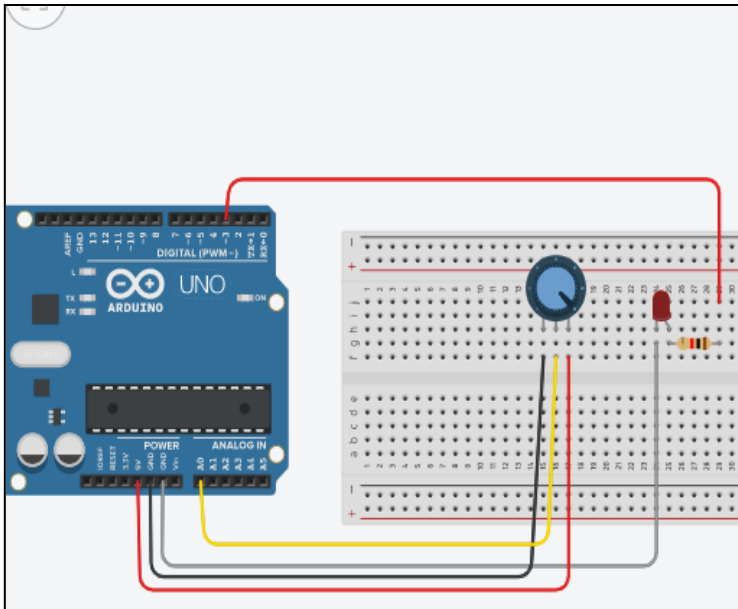
#define=Sabit değişken değeri tanımlar.

const= Sabit değişken değeri tanımlar.

analogRead=Analog pinlerdeki sensörlere gelen değeri okur.

map komutu=Analog bir değeri istenilen pwn değerine çevirir.

POTANSİYOMETRE İLE LEDİN PARLAKLIĞINI DEĞİŞTİRME



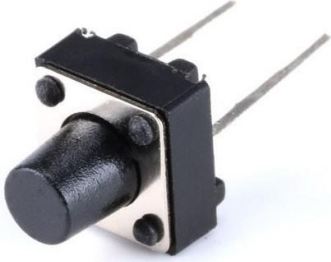
```
1 #define led 3
2 #define pot A0
3
4 void setup() {
5 }
6
7 void loop() {
8   int deger = analogRead(pot);
9   deger = map(deger, 0, 1023, 0, 255);
10  analogWrite(led, deger);
11 }
```

```
int led=3;
int potansiyometre=0;
int parlaklik;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  parlaklik=analogRead(potansiyometre) / 4;
  analogWrite(led, parlaklik);
}
```

Deney 6: Push Buton ile LED Yakmak



Malzeme Listesi:

- 1) Arduino UNO
- 2) Push Buton
- 3) 10K direnç (Buton için)
- 4) LED Diyot
- 5) 220 / 270 / 330 ohm direnç (LED için)
- 6) Yeter sayıda jumper kablo

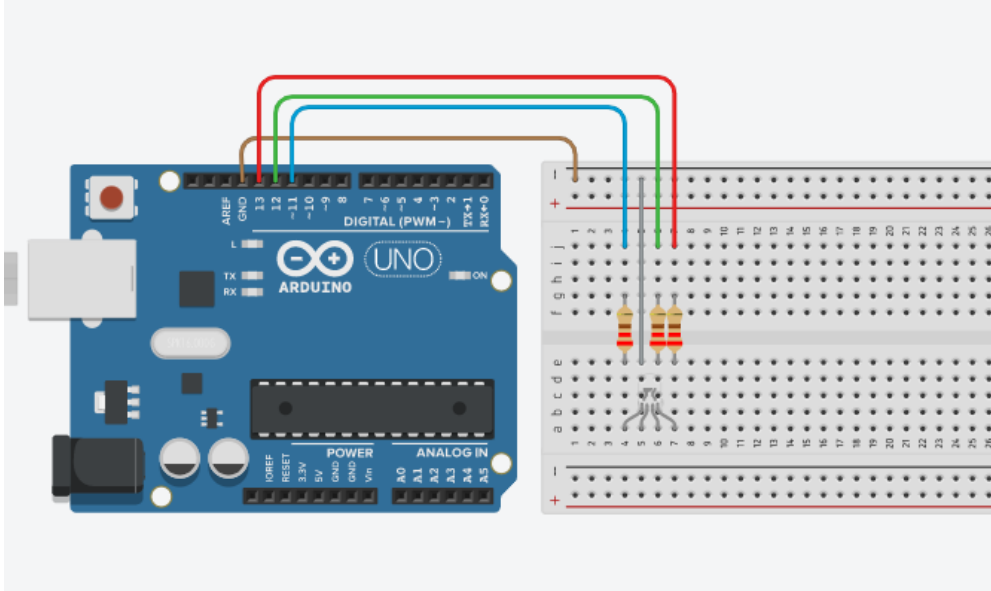
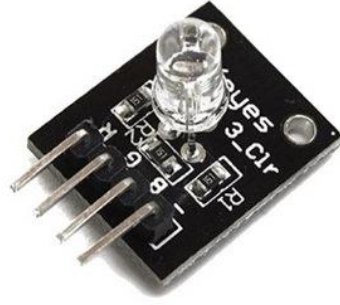
Örnek Kodumuz:

```
int ledPin=3;
int butonPin=2;
int butonDurum=0;
void setup(){
    pinMode(ledPin,OUTPUT); //led pinimizi çıkıs ayarladık
    pinMode(butonPin,INPUT); // buton pinimizi giriş ayarladık
}
void loop(){
    butonDurum=digitalRead(butonPin); // dijital olarak okuduk
    if(butonDurum==HIGH){ // Eğer butona basılmış ise LED i yakalım
        digitalWrite(ledPin,HIGH);
    }else
        digitalWrite(ledPin,LOW);
}
```

Deney 7: RGB LED Kontrolü

Malzeme Listesi:

- 1) Arduino UNO
- 2) RGB LED
- 3) Yeter sayıda jumper kablo



Bilgi:

RGB LED, Red(Kırmızı), Green(Yeşil) ve Blue(Mavi) olmak üzere 3 ana rengi ve bu renklerin birleşim ara renkleri görüntüleyebilen bir LED çeşididir. RGB led 4 bacaklı led olarak veya modül olarak bulabileceğiniz oldukça uygun fiyatlı bir üründür.

RGB LED Arduino'ya bağlanırken R, G ve B bacakları ile kırmızı, yeşil ve mavi renkleri kontrol edeceğimizden Arduino üzerindeki PWM (~ işaretli pinler) pinlerinden herhangi birine bağlanmalıdır. Eksi (-) ile gösterilen pin ise her hangi bir GND yani toprak pinine bağlanabilir.

RGB LED içerisindeki her bir renk için 0-255 arasında PWM sinyali ile renk ayarı yapılabilir. Bu nedenle RGB led arduino bağlantısı yapılırken arduinodaki ~ işareti ile gösterilen PWM özellikli dijital

pinler tercih edilmelidir. Bu sayede 3 farklı pinden ledlere farklı PWM değerleri göndererek RGB led ile farklı renkler elde etmek mümkün olacaktır. Şimdi dişi-erkek jumper kullanarak RGB LED inizi Arduino'nuza bağlayıp gerekli kodu yazınız.

Örnek Kodumuz:

```
int r=3; // 3 numaralı PWM pinine bağlanmış
int g=5; // 5 numaralı PWM pinine bağlanmış
int b=6; // 6 numaralı PWM pinine bağlanmış

void setup() {
  pinMode(r,OUTPUT); // r=3 numaralı pin çıkış olarak ayarlandı
  pinMode(g,OUTPUT); // g=5 numaralı pin çıkış olarak ayarlandı
  pinMode(b,OUTPUT); // b=6 numaralı pin çıkış olarak ayarlandı
```

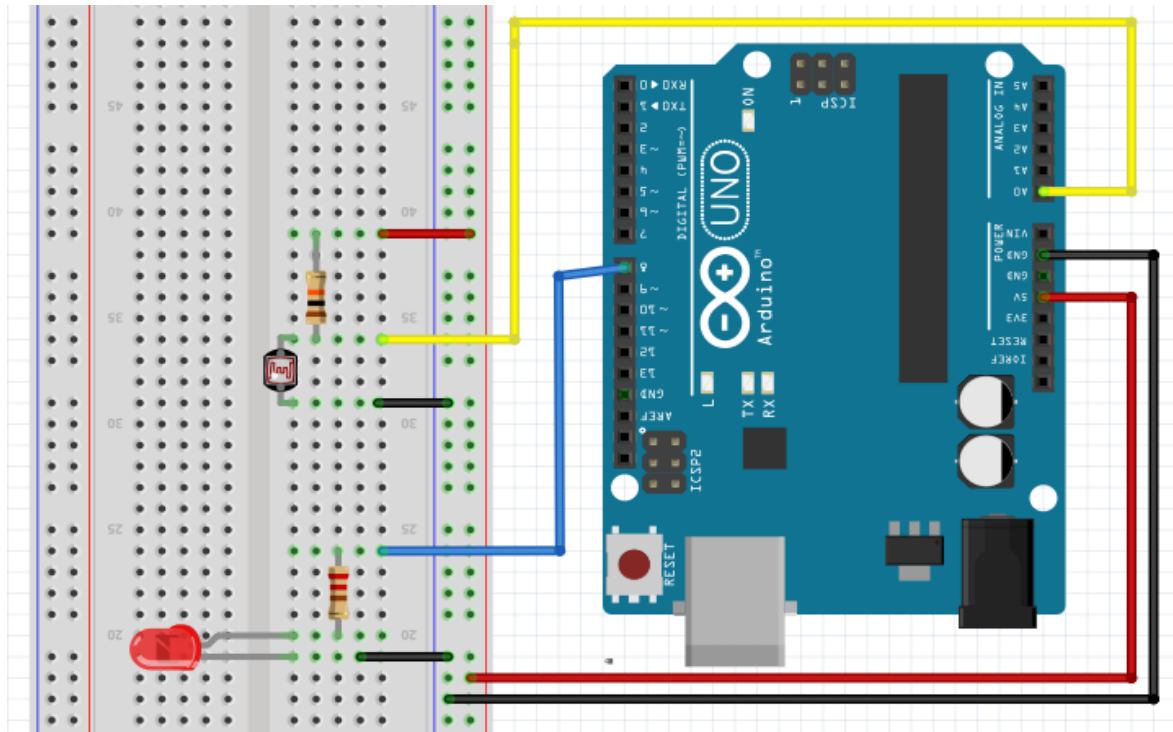
```
}  
  
void loop() {  
  analogWrite(r,0); // r pinine Analog 0 değeri gönderildi  
  analogWrite(g,255); // g pinine Analog 255 değeri gönderildi  
  analogWrite(b,0); // b pinine Analog 180 değeri gönderildi  
  delay(500); // 500 milisaniye bekletiliyor  
}
```

sketch_oct25a

```
void setup()  
{  
  Serial.begin(9600);  
  Serial.println("Pot Deger Okuma"); }  
void loop()  
  
{  
  
  Serial.println(analogRead(A0));  
  delay(1000);  
}
```

```
Pot Deger Okuma  
568  
584  
584  
585  
585
```

LDR İLE GECE LAMBASI



```
#define led 8
void setup() {
    pinMode(led, OUTPUT);
    Serial.begin(9600);
}
void loop() {
    int isik = analogRead(A0);
    Serial.println(isik);
    delay(50);
    if(isik > 500){
        digitalWrite(led, LOW);
    }
    if(isik < 300){
        digitalWrite(led, HIGH);
    }
}
```

ULTRASONİK SENSÖR KULLANIMI

Ultrasonik Mesafe Sensörü Nedir Ne İşe Yarar?

Hc-sr04 Ultrasonik sensör sonar (Sound Navigation and Ranging) iletişim kullanarak karşısındaki nesneye olan mesafeyi hesaplayan bir kaynaktır. Sonar dediğimiz sistem ses dalgalarını kullanarak cismin uzaklığını hesaplamamıza yardımcı olur. Bu tür sensörlerin esin kaynağı yunuslar ve yarasalardır. Yunuslar ve yarasalarda ses dalgası göndererek karşısına çıkabilecek engellerin mesafelerini hesaplayabilmektedirler.

Hc-sr04 sensörümüzün 4 adet bacağı bulunmaktadır, bunlar:

- Vcc = 5v kaynağı.
- Gnd = Topraklama bacağı.
- Trig = Sensörün ses dalgası gönderen kısmı.(output)
- Echo = Gönderilen ses dalgasını alan kısmı(.input)

Hc-sr04 sensörümüz 5v elektrik akımı ile çalışmaktadır. En verimli ölçüm yaptığı mesafe 2-200 cm arasındadır. 200 cm'den fazla mesafelerde verimli bir şekilde ölçüm yapmamaktadır.

Arduino Ultrasonik Mesafe Sensörü Kullanımı Kodları

```
int trigPin = 2;  
int echoPin = 7;  
long sure;  
long uzaklık;
```

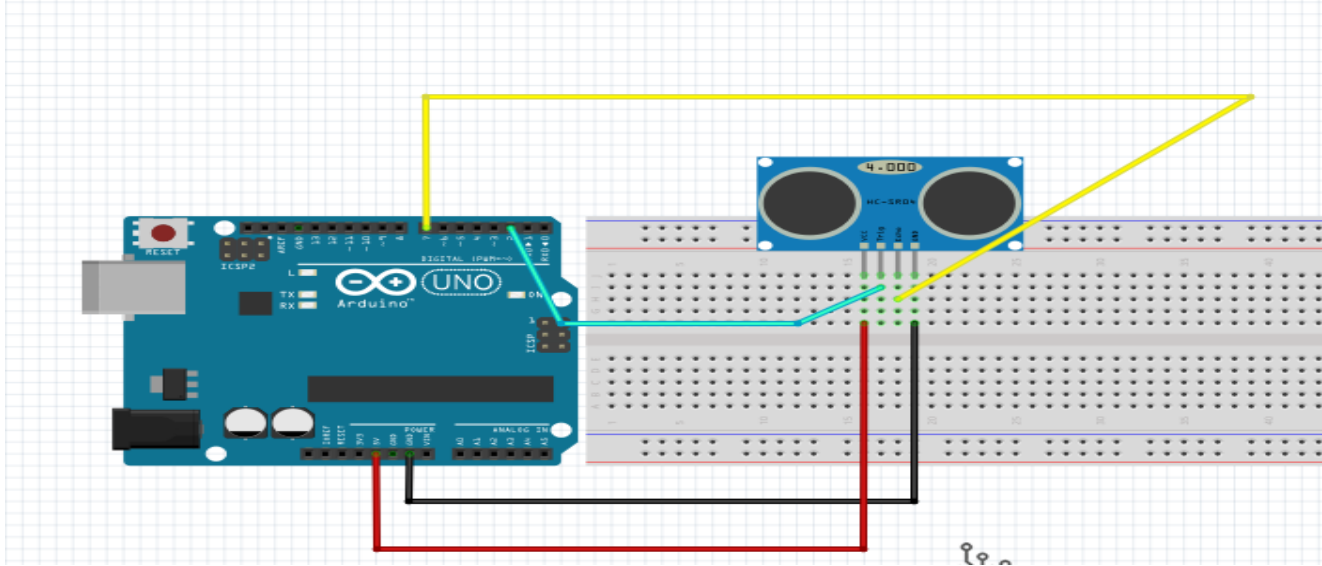
Burada öncelikle bizim ses dalgası gönderen ve ses dalgasını alan pinlerimizin Arduino kartımız üzerindeki pinlerimizi belirliyoruz. Daha sonra süre ve uzaklık adında iki tane değişken atıyoruz.

```
pinMode(trigPin, OUTPUT);  
pinMode(echoPin,INPUT);  
Serial.begin(9600);
```

Burada ise trig pinimizi OUTPUT olarak belirliyoruz. Bunun sebebi ise ses dalgasını gönderen kısmımız trig pinimiz olmasından dolayı. Echo pinimiz ise gönderilen ses dalgasını aldığı için INPUT olarak belirtiyoruz ve son olarak Serial.begin ile de seri haberleşmemizi başlatıyoruz.

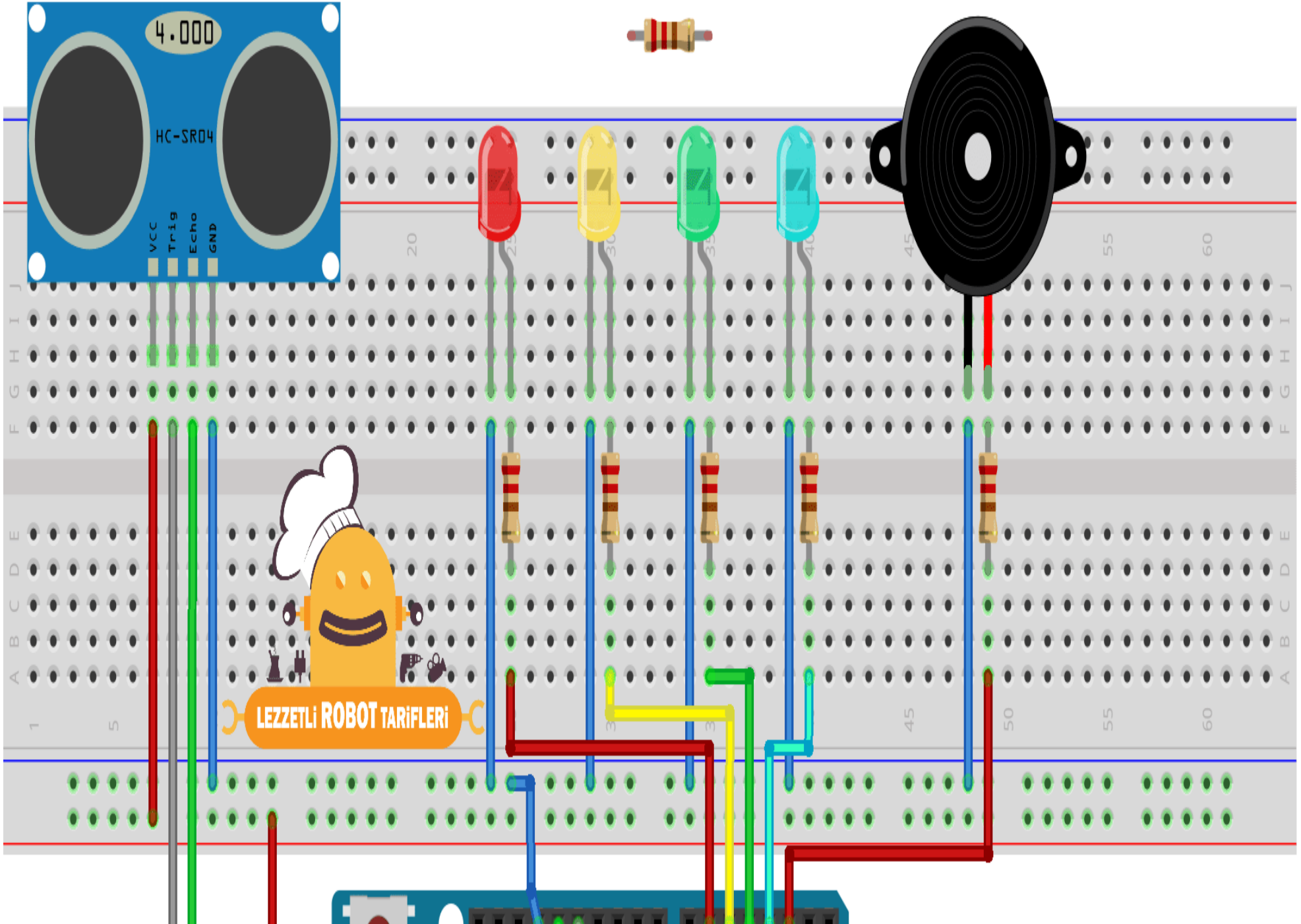
```
digitalWrite(trigPin, LOW);İlk olarak trip pinimizi low durumunda başlatıyoruz.
```

```
delayMicroseconds(5); 5 Mikrosaniye(saniyenin milyonda biri) beklemesini belirtiyoruz.  
digitalWrite(trigPin, HIGH); Daha sonra pinimizin ses dalgası göndermesi için emir veriyoruz.  
delayMicroseconds(10);  
digitalWrite(trigPin, LOW); Yeni ses dalgası üretebilmesi için trig pinimizi pasif durumuna getiriyoruz.  
sure = pulseIn(echoPin, HIGH); Gönderilen ses dalgasının geri dönmesindeki süre ölçülüyor. gidiş-dönüş  
uzaklik= sure /29.1/2; Ölçtüğümüz süre uzaklığa çevriliyor.  
if(uzaklik <20) if komutu ile 200 cm ve üzeri bütün uzaklıklar 200 cm olarak yazılacak.  
uzaklik = 200;  
Serial.print("Uzaklik ");  
Serial.print(uzaklik); Ölçtüğümüz uzaklığımız bilgisayarımıza yani Arduino programımızda Araçlar>Seri Port Ekranı kısmına yazılıyor.  
Serial.println(" CM ");  
delay(100); }
```



BU DEVREYE LED VE BUZZER EKLEYEREK DEVAM EDELİM.


```
int trigPin = 2;
int echoPin = 7;
long sure;// long tam sayıdan daha büyük sayılar için kullanılır
long uzaklik;
void setup(){
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  sure = pulseIn(echoPin, HIGH);
  uzaklik= sure /29.1/2;
  if(uzaklik > 200)
    uzaklik = 200;// buraya eğer belli bir mesafeden sonra led yan buzzer öt komutları ekleyebiliriz
  Serial.print("Uzaklik =");  Serial.print(uzaklik);  Serial.println(" CM ");  delay(100);
}
```



220Ω

4.000

HC-SR04

VCC
Trig
Echo
GND

LEZZETLİ ROBOT TARIFLERİ

```

const int trigger_pin = 12;
const int echo_pin = 13;
const int buzzer = 2;
const int mled = 3;
const int yled = 4;
const int sled = 5;
const int kled = 6;
int sure;
int mesafe;
void setup() {
    pinMode(kled, OUTPUT); // kırmızı led
    pinMode(sled, OUTPUT); // sarı led
    pinMode(yled, OUTPUT); // yeşil led
    pinMode(mled, OUTPUT); // mavimsi yeşil led
    pinMode(buzzer, OUTPUT); // ses
    pinMode(trigger_pin, OUTPUT); // ses dalgası yollar
    pinMode(echo_pin, INPUT); // ses dalgası bir nesneye çarpıp buraya döner
    Serial.begin(9600); // seriport ekranı başlatma
}

void loop() {
    digitalWrite(trigger_pin, HIGH); // ses dalgası salınımı yapılır ara ara
    delayMicroseconds(1000);
    digitalWrite(trigger_pin, LOW);
    sure = pulseIn(echo_pin, HIGH); // ses dalgası trig pininden çıkıp echoya kadar giden süre
    hesaplanır.*/
    mesafe = (sure / 2) / 28.5; // bu süre 2 ye bölünür çünkü gidiş-dönüş yolu mesafe cm
    dönüştürülür. 28.5 sabit bir değer (sıcaklığa göre değişebiliyor. Bazı yerlerde 29.1 dalga hızı
    hesaplamaları ile bulunuyor.*/

    if (mesafe <= 10) // mesafe 10 cm den küçük ve eşitse kırmızı led yak buzzer 50 ms delay olması
    sık sık ses çıkarması
    {
        digitalWrite(kled, HIGH);
        digitalWrite(buzzer, HIGH);
    }
}

```

```
    delay(50);
    digitalWrite(kled, LOW);
    digitalWrite(buzzer, LOW);
    delay(250);
}

else if (mesafe <= 25)
{
    digitalWrite(sled, HIGH);
    digitalWrite(buzzer, HIGH);
    delay(50);
    digitalWrite(sled, LOW);
    digitalWrite(buzzer, LOW);
    delay(500);
}

else if (mesafe <= 50)
{
    digitalWrite(yled, HIGH);
    digitalWrite(buzzer, HIGH);
    delay(50);
    digitalWrite(yled, LOW);
    digitalWrite(buzzer, LOW);
    delay(750);
}

else
{
    digitalWrite(mled, HIGH);
    delay(1000);
    digitalWrite(mled, LOW);
    delay(1000);
}
}
```

ARDUİNO PIR HAREKET SENSÖRÜ KULLANIMI

Yazar : MURAT DURAN22 AĞUSTOS 2015

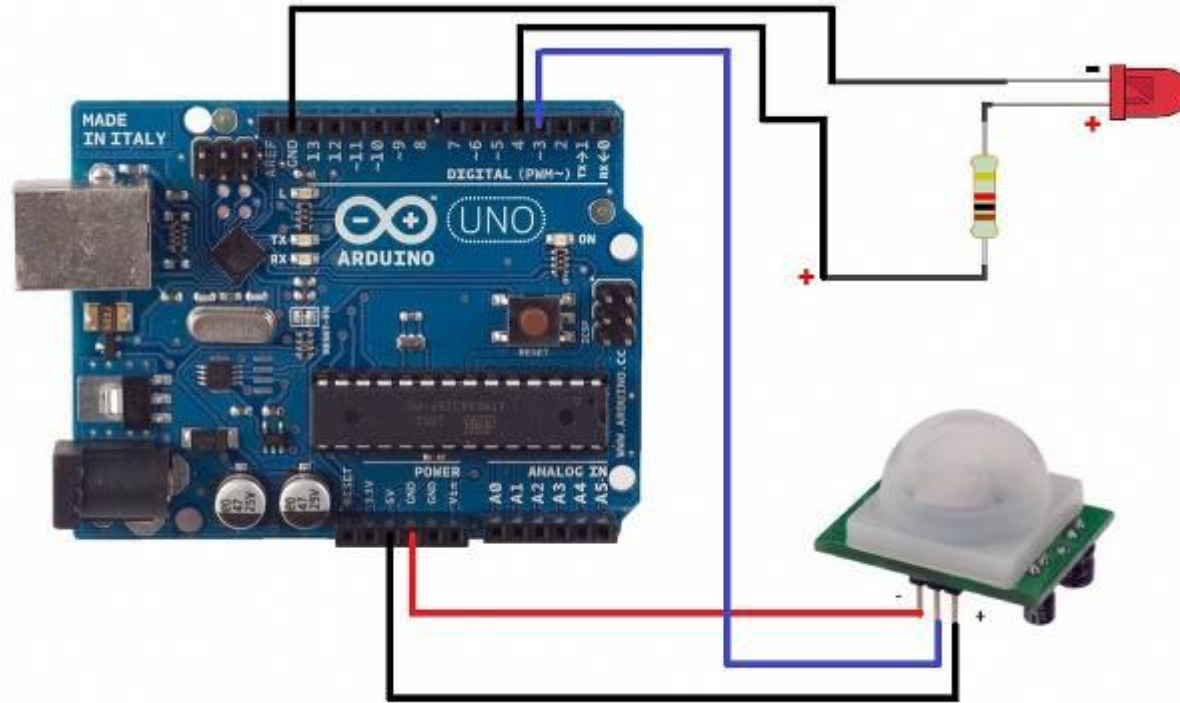
PIR hareket sensörü birçok alanda kullanılan sensör çeşitlerindedir.En sık karşılaştığımız yerler otomatik lambalardır.Ayrıca bazı uyulamalar da otomatik kapılarda ve yürüyen merdivenlerde de kullanılmaktadır.Kullanım alanı sadece bunlarla sınırlı değildir.Sizin hayal gücünüze kalmıştır.PIR (Passive Infrared sensor) , türkçesi pasif kızılötesi sensör demektir.

Projemiz gayet basit ve açıktık.Malzemelerimiz

- 1- Arduino
- 2- PIR sensörü
- 3- Led + direnç

PIR sensörünün 3 adet pin girişi vardır. VCC , GND ve Sinyal pinleri.3.3 ile 5 V arası çalışmaktadır.Enerjiyi arduino üzerinden almayı tercih ediyorum.Cihaz yaklaşık 3-5 metre arası aktif haldedir. Hareket algıladığı an DATA çıkışını HIGH yapmaktadır. Yaklaşık 15sn HIGH'ta bulunmaktadır. Daha sonra sabit değere yani LOW'a geri dönmektedir. Uygulamamızda gayet basit olacaktır.

Arduino da herhangi bir dijital pin'i input yaptıktan sonra anlık olarak DATA girdisini okuyacağız ardından bir led'i yakıp söndürme işlemi yapacağız.



```
1 int pirPin = 3; // PIR pin
2 int ledPin = 4; // LED pin
3 int deger = 0;
4
5 void setup() {
6   pinMode(pirPin, INPUT); // PIR Pin'i giriş yapılıyor
7   pinMode(ledPin, OUTPUT); // LED Pin'i çıkış yapılıyor
   Serial.begin(9600); //Serial Porttan veri göndermek için baudrate ayarlanıyor.
```

```
8   }
9
10  void loop(){
11  deger = digitalRead(pirPin); // Dijital pin okunuyor
12  Serial.println(deger); // Okunan deęer seri porttan okunuyor.
13  if (deger == HIGH) {
14  digitalWrite(ledPin, HIGH); // Eęer okunan deęer 1 ise LED yakılıyor.
15  }
16  else{
17  digitalWrite(ledPin,LOW); // Eęer okunan deęer 0 ise LED söndürülüyor.
18  }
19
20
```