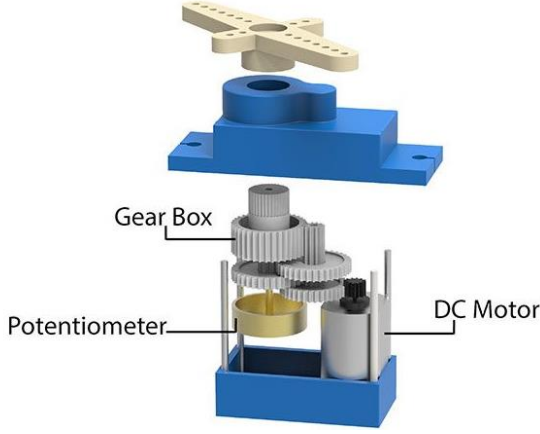


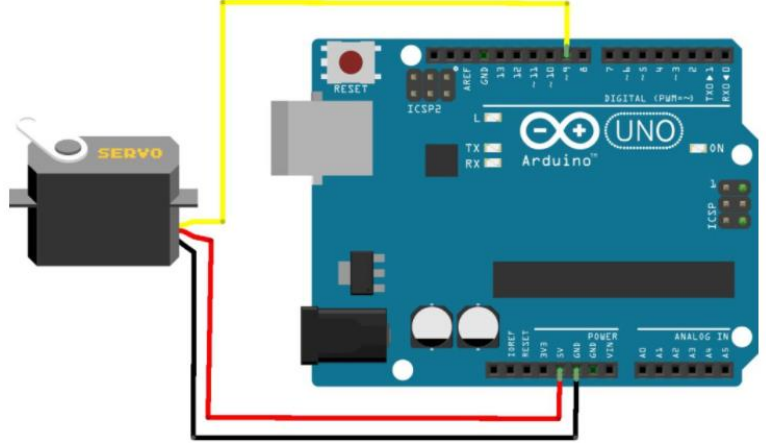
Servo Motorlar

Servo motorlar 0 ile 180 derece arasında bir açıda döndürebildiğimiz elektronik bileşenlerdir. İhtiyacımız olan güce göre çeşitli büyüklükte ve plastik ve metal dişlilere sahip modelleri bulunmaktadır.



Arduino servo motor kontrolü için PWM pinlerinden yararlanıyoruz. Bu pinlerin yanında dalga (~) işaret bulunmaktadır. Servo motoru bu pinlerle kullanabilirsiniz. Servo motorun 3 adet bağlantı kablosu bulunmaktadır. Projelerde ve prototiplerde kullanılan hem yaygın hem de ekonomik olan SG-90 servo motorunda renkleri kahverengi, kırmızı ve turuncu olmaktadır. Bu kabloların özellikleri kahverengi (GND), kırmızı (5v), Turuncu (Dijital PWM Pin) sinyal pini olarak belirlenmiştir.

Nasıl Bağlantı Sağlarız?



Genelde kullanılan SG-90 servo motorun kablo renkleriyle devremizdeki servo motor temsili ile karşılaştığımızda sarı renkli kablo sg-90 da Turuncu, kırmızı renk yine kırmızı, siyah renk ise kahverengi kablo ile eşleşmiştir. Bağlantımızı yukarıdaki devreye göre sağladıktan sonra kodumuzu yazmadan Servo kütüphanesini Arduino IDE'ye eklememiz gerekiyor.

Arduino IDE'ye Kütüphane Ekleme

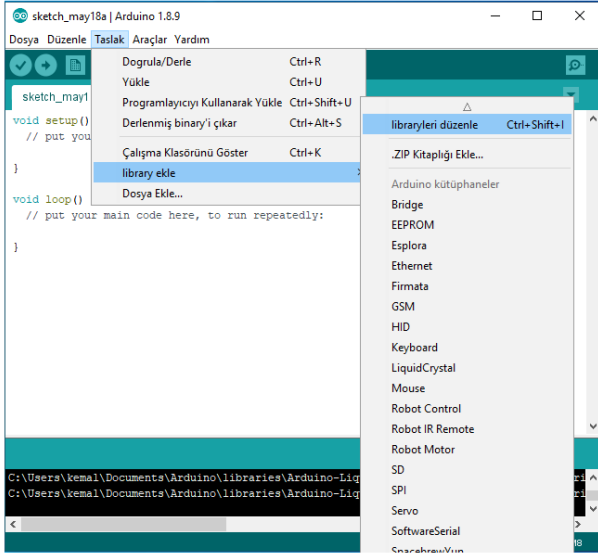
Arduino IDE'ye kütüphane ekleme işlemini 3 farklı yöntem ile gerçekleştirebiliriz. bu yöntemler;

Arduino IDE kütüphane yükleme yöneticisi ile eklemek

Arduino IDE içerisinde Zip dosyası olarak eklemek

Doğrudan kütüphane klasörüne eklemek

Biz Arduino IDE kütüphane yükleme yöneticisi üzerinden servo kütüphanesini ekleyeceğiz. Diğer yöntemleri de Arduino ile uğraştıkça farklı sensör ve modüllerle çalışırken ihtiyaç duyduğunuzda deneyimleme şansını bulacaksınız.



Arduino ile Servo Motor Kontrolü Uygulaması

```
#include <Servo.h> // Servo kütüphanesini projemize ekledik.
Servo servoM; // Servo nesnemizi oluşturduk, ismini servoM yaptık.
int pos = 0; // pos değişkenini oluşturup 0'a eşitledik.
void setup() {
  servoM.attach(9); // Servoyu 9nolu pin'e bağladığımızı belirttik.
}
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // For döngüsü ile 0 ile 180 derece
    // her bir adımda 1 derece artacak şekilde ayarladık.
    servoM.write(pos); // Servo açılış değeri olarak belirlediğimiz pos
    // değişkenini servoya yazdırdık.
    delay(15); // servonun hedeflenen açılış gidebilmesi için 15
    // ms bekleme ekledik.
  }
  for (pos = 180; pos >= 0; pos -= 1) { // for döngüsü ile 180 ile 0 derece
    // arası gitmesini sağladık.
  }
}
```

Arduino IDE menüsünden TASLAKLAR / Library Ekle / Libraryleri Düzenle seçeneğini seçiyoruz.

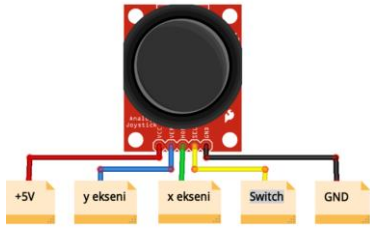
```
servoM.write(pos); // Servo açılış değeri olan pos değişkenini servo'ya yazdırdık.
delay(15); // Servonun açılış değerine gidebilmesi için 15 ms. bekleme ekledik.
}
```

Arduino ile Joystick Kullanımı

Joystick, iki eksenle hareket etmeyi sağlayan iki potansiyometre bulunduran ve dikey hareketinde ise buton görevi gören bir modüldür.

Joystick'leri genellikle oyun konsollarında ve uzaktan kumandalı araçlarda görürüz.

Arduino ile uyumlu joystick modülleri ile birçok farklı proje yapılma imkanı vardır. Motor kontrolü, LED kontrolü, robotik kontroller gibi birçok farklı uygulama görmeniz mümkündür.

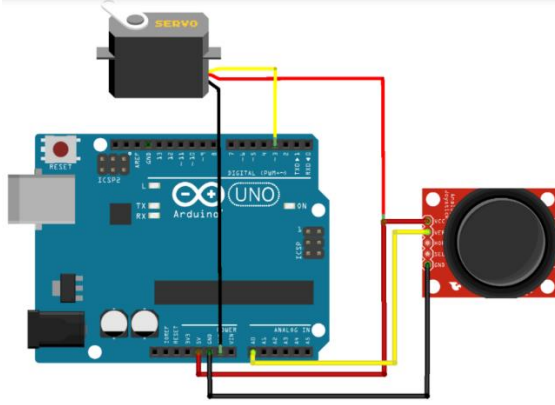


Arduino Joystick shield butonu, diğer butonların tersine çalışan bir karakteristiğe sahiptir. Yani bir pullup butondur. Butona basmadığımızda 1, bastığımızda ise 0 değerini okuruz.

Aşağıda Joystick modülünün pinout yapısı bulunmaktadır. Dikey, yatay ve buton hareketlerinin hepsi için ayrı birer data pini bulunmaktadır. Modülün çalışabilmesi için ise 5V ve GND beslemesinin yapılması gerekmektedir.

Arduino ile Servo Motor Kullanımı

Servo motorlar sayesinde çok hassas pozisyon kontrolü yapmak mümkündür. Çoğunlukla çalışma açıları 0-180 derece arasındadır. Arduino'da servo motor kontrol kodları bulunmaktadır. Bunlar için Arduino'da Dosya>Örnekler>Servo>Sweep ve Dosya>Örnekler>Servo>Knob komutlarını takip etmeniz gerekecektir. Arduino'da Servo Motorlar PWM sinyali ile çalışırlar. PWM'in Türkçe açılımı Sinyal Genişlik Modülasyonudur ve bu sinyaller mikrokontrolcülerden veya uzaktan kumandalardan sağlanırlar. PWM pinleri ile servonun tam dönüş açısı ve dönüş süresi kontrol edilebilmektedir. Arduino PWM Pinleri: 3, 5, 6, 9, 10, 11



Arduino Joystick ile Servo Motor Kontrolü

```
#include <Servo.h> //servo kütüphanesi ekleme
Servo motor;
int deger;
int derece;
void setup() {
  motor.attach(3); //servonun 3 nolu pine bağlı olduğunu gösterir.
}
void loop() {
  deger = analogRead(A0);
  derece = map(deger, 0, 1023, 0, 180);
  motor.write(derece);
}
```

Arduino ile LDR Kullanımı – (LDR ile LED Kontrolü)

Arduino dersleri serimizde bolca uygulama yapmaya devam ediyoruz. Bu yazımızda kurduğumuz devredeki INPUT değerine göre OUTPUT çıkışının otonom olarak nasıl değiştiğini göreceğiz.

Yani bir sensör-aktüatör algoritması yazacağız. LDR 'Light Dependent Resistor' tamlamasının kısaltmasından gelmektedir ve anlamı ışığa bağımlı dirençtir.

LDR ayrıca foto direnç olarak da adlandırılır. Bir direnç çeşidi olmasının yanında aynı zamanda bir pasif sensör olan LDR, ışık şiddetine bağlı olarak yapısında değişiklik gösterir. Bu ortamdan aldığı fiziksel değişimden faydalanarak çıkışında da bir değişiklik gösterebildiği için sensör görevi gördüğünü söyleyebiliriz.



Elektronik devrelerde LDR'yi tekli halde yukarıdaki şekilde görüldüğü gibi kullanabiliriz.

Arduino ile LDR Kullanımı

Bu yazımızda karanlıkta çalışan LDR devresi uygulaması yapacağız. Yani aslında LDR ile ışık kontrolü yapmış olacağız. Hemen gerekli malzeme listesini inceleyelim ve uygulamaya başlayalım.

Gerekli malzemeler:

Arduino Uno

Breadboard

1 adet 1KΩ Direnç

1 adet 220Ω Direnç

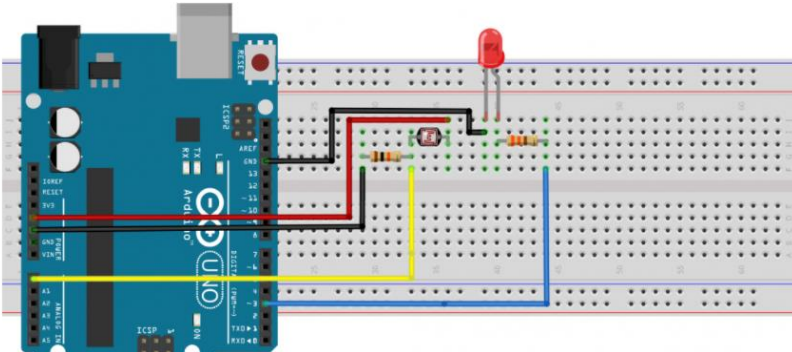
LDR

1 adet LED

Karanlıkta LED Yakan LDR Devresini Kuralım

LDR LED devresi için ihtiyacımız olan malzemeler hazır ise aşağıdaki gibi devremizi kuralım.

Devremiz de hazır ise Arduino LDR kodunu aşağıdaki gibi yazabiliriz.



```
#define led 3 //3.Pinde LED olduğunu tanımlıyoruz
```

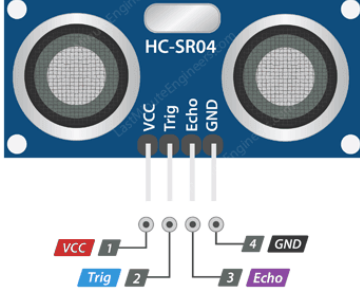
```
void setup() {
  pinMode(led, OUTPUT); //LED'in çıkış elemanı olduğunu belirtiyoruz
  Serial.begin(9600); //9600 Baundluk bir seri haberleşme başlatıyoruz
}
```

```

void loop() {
  int isik = analogRead(A0); //Işık değışkenini A0 pinindeki LDR ile okuyoruz
  Serial.println(isik); //Okunan değeri seri iletişim ekranına yansıtıyoruz
  delay(50);
  if (isik > 900) { //Okunan ışık değeri 900'den büyük ise
    digitalWrite(led, LOW); //LED yanmasın
  }
  if (isik < 850) { //Okunan ışık değeri 850'den küçük ise
    digitalWrite(led, HIGH); //LED yansın}}

```

Mesafe sensörü ne işe yarar?



Herhangi bir fiziksel temas söz konusu olmadan bir nesnenin yakınlığını belirlemede kullanılan sensörlere mesafe sensörleri adı verilir. Optik, akustik, kapasitif, endüktif ve fotoelektrik fizik dahil olmak üzere farklı fiziksel mekanizmalara dayalı olarak çalışan birkaç çeşit mesafe sensörü vardır. Uzaklık ölçümü için HC-SR04 ismiyle anılan ultrasonik sensörler çok kullanılır. Sensör üzerinde giriş ve çıkış olmak üzere iki kısım bulunmaktadır. Çıkış kısmından ortama salınan ses sinyali, nesnelere yansıyor giriş kısmında toplanır. Sesin çıkış kısmından çıkmasıyla, giriş kısmına ulaşması arasında geçen süre ölçülerek, cismin uzaklığı hesaplanır. Bu mantıkla çalışan sensör, 2 cm ile 400 cm arasındaki uzaklıkları ölçebilmektedir.

Sensör üzerinde VCC, Trig, Echo, GND olmak üzere 4 adet pin bulunmaktadır. Bunlardan VCC pini besleme (5 volt), GND pini toprak hattıdır. Trig pini (TETİK) ile çıkış kısmından sesin salınması sağlanır. Echo pini (OKUMA) ise giriş kısmına yansıyan sesin ulaştığını haber veren pindir.

Arduino ile Park Sensörü Yapımı

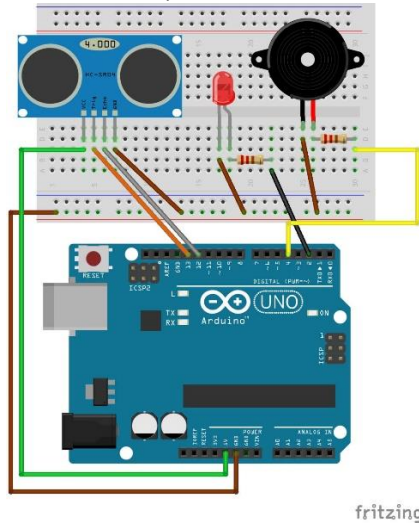
Bu yazımızda arduino öğrenirken vazgeçilmez uygulamalardan biri olan Park Sensörü yapımını anlatacağız. Bu eğlenceli ve bir okadarda bütçe dostu arduino projesi yeni başlayanlar için güzel bir başlangıç projesi olabilir. Arduino ile park sensörü yapımıyla HC-SR04 Ultrasonik Sensörünün kullanımını öğrenmiş olacaksınız. Proje içinde kullandığımız malzemeler ise [Arduino Uno](#), [HC-SR04 Ultrasonik Mesafe Sensörü](#), Buzzer, ve 5mm Led'dir. Eksik olduğunu düşündüğünüz malzemeleri ürün ismine tıklayarak sipariş verebilirsiniz. Dilerseniz [Arduino Megalos Başlangıç Seti](#) içerisindeki malzemelerle bu eğlenceli ve öğretici projeyi gerçekleştirebilirsiniz.

Bu Projede Neler Öğrenilecek?

Projenin detaylarına inmeden önce bu arduino uygulamasıyla buzzer kontrolü, led kontrolü, ultrasonik sensör kullanımı ve arduino ile if else() kullanımını öğrenmiş olacaksınız.

Arduino ile Park Sensörü Yapımı için Gerekli Malzemeler Nelerdir?

- Arduino Uno R3 Kartı
- HC-SR04 Ultrasonik Mesafe Sensörü
- 5mm LED
- Orta Boy Breadboard
- 220ohm Direnç
- Erkek-Erkek Jumper Kablo



Devrenizi yukarıdaki gibi oluşturabilirsiniz. Ardından arduino uno kartınızı bilgisayara usb kablo ile takıp kodlamaya geçebilirsiniz.

Arduino ile Park Sensörü Kodu Nasıl Yazılır?

Park sensörü uygulamasında if döngüsü kullanacağız. Hc-sr04 ultrasonik mesafe sensöründen gelen veriyi cm cinsine çevireceğiz. Sonrasında if döngüsü içerisinde ölçülen uzaklıklara göre belirli aralıklarla buzzerden uyarı sesi verip ledi yakıp söndüreceğiz. Projemizi kodlamaya geçmeden önce if-else () kullanımı nasıldır bunu aşağıdan inceleyebilirsiniz.

If Else Yapısı Nedir?

If İngilizcede eğer anlamında kullanılmakta olup else ise değil anlamındadır. If else yapısında bir koşul veya koşullar doğruysa yapılacak işlemler, koşul doğru değilse yapılacak işlemlerin yazıldığı bir çeşit sorgudur.

```

if( koşul1)
{
  Koşul1 Doğruysa Yapılacak işlemler
}
else if( koşul2)
{

```

```

  Koşul2 doğruysa yapılacak işlemler
}
else
{
  Hiçbir koşul doğru değilse yapılacak işlemler
}

```

Yukarıdaki örnekte çalışma mantığını daha kolay anlayabilirsiniz. Koşulları oluştururken karşılaştırma operatörlerini kullanınız. Projemizin yazılım aşamasında pekiştireceğinizi düşünüyorum.

pulseIn() fonksiyonu, bir pinin ne kadar süre HIGH veya LOW durumda kaldığını takip eder ve geçen süreyi mikrosaniye cinsinden döndürür.

Arduino ile Park Sensörü Kodları

```
const int trigger_pin = 12; //12. pini trigger pin olarak tanımlandı.  
const int echo_pin = 13; //13. pini echo pin olarak tanımlandı.  
int uyariled = 2; //2. pini uyariled olarak tanımlandı.  
int buzzer = 6; //6. pini buzzer olarak tanımlandı.
```

```
int sure ; //sure adlı bir değişken tanımlandı.
```

```
int mesafe ; //mesafe adlı bir değişken tanımlandı.
```

```
void setup() {
```

```
pinMode(uyariled , OUTPUT); //led'i çıkış olarak tanımladık.
```

```
pinMode(buzzer , OUTPUT); //buzzer'i çıkış olarak tanımladık.
```

```
pinMode(trigger_pin , OUTPUT); //trigger pin'i çıkış olarak tanımladık.
```

```
pinMode(echo_pin , INPUT); //echo pin'i giriş olarak tanımladık.
```

```
}
```

```
void loop()
```

```
{digitalWrite(trigger_pin , HIGH);
```

```
delayMicroseconds(1000);
```

```
digitalWrite(trigger_pin , LOW);
```

```
sure = pulseIn(echo_pin , HIGH); //echo_pin verisi sure değişkenine atandı.
```

```
mesafe = (sure / 2) / 29.1; //cm cinsine çevrildi.
```

```
if (mesafe <= 10) //mesafe 10 cm den kısaysa aşağıdaki işlemler gerçekleşir.
```

```
{digitalWrite(uyariled , HIGH);
```

```
digitalWrite(buzzer , HIGH);
```

```
}
```

```
else if(mesafe>10 && mesafe<=20) //Mesafe 10 cm den uzun 20cm de eşit veya kısaysa aşağıdaki işlemler gerçekleşir.
```

```
{digitalWrite(uyariled , HIGH);
```

```
digitalWrite(buzzer , HIGH);
```

```
delay(150); // ledin yanık kalma süresiyle buzzerin uyarı süresi standart bir süreye 150ms ye ayarlandı.
```

```
digitalWrite(uyariled , LOW);
```

```
digitalWrite(buzzer , LOW);
```

```
delay(250);
```

```
}
```

```
else if(mesafe>20 && mesafe<=30) //Mesafe 20 cm den uzun 30cm de eşit veya kısaysa aşağıdaki işlemler gerçekleşir.
```

```
{digitalWrite(uyariled , HIGH);
```

```
digitalWrite(buzzer , HIGH);
```

```
delay(150);
```

```
digitalWrite(uyariled , LOW);
```

```
digitalWrite(buzzer , LOW);
```

```
delay(450);}
```

```
else if(mesafe>30 && mesafe<=40) //Mesafe 30 cm den uzun 40cm de eşit veya kısaysa aşağıdaki işlemler gerçekleşir.
```

```
{digitalWrite(uyariled , HIGH);
```

```
digitalWrite(buzzer , HIGH);
```

```
delay(150);
```

```
digitalWrite(uyariled , LOW);
```

```
digitalWrite(buzzer , LOW);
```

```
delay(650);}
```

```
else if(mesafe>40 && mesafe<=50) //Mesafe 40 cm den uzun 50cm de eşit veya kısaysa aşağıdaki işlemler gerçekleşir.
```

```
{digitalWrite(uyariled , HIGH);
```

```
digitalWrite(buzzer , HIGH);
```

```
delay(150);
```

```
digitalWrite(uyariled , LOW);
```

```
digitalWrite(buzzer , LOW);
```

```
delay(850);}}
```

Devreyi kurup yazılımı arduinoya ekledikten sonra yükleyici butonuna basarak arduino kartınıza yazılım yükleyebilirsiniz.

Uygulama çalışmasında mesafe sensörüne cisim yaklaştıkça verilen uyarının miktarı artacak ve 10cm ve altında led ve buzzer sürekli açık kalacaktır.

Mesafe uzadıkça uyarılar azalacaktır.