

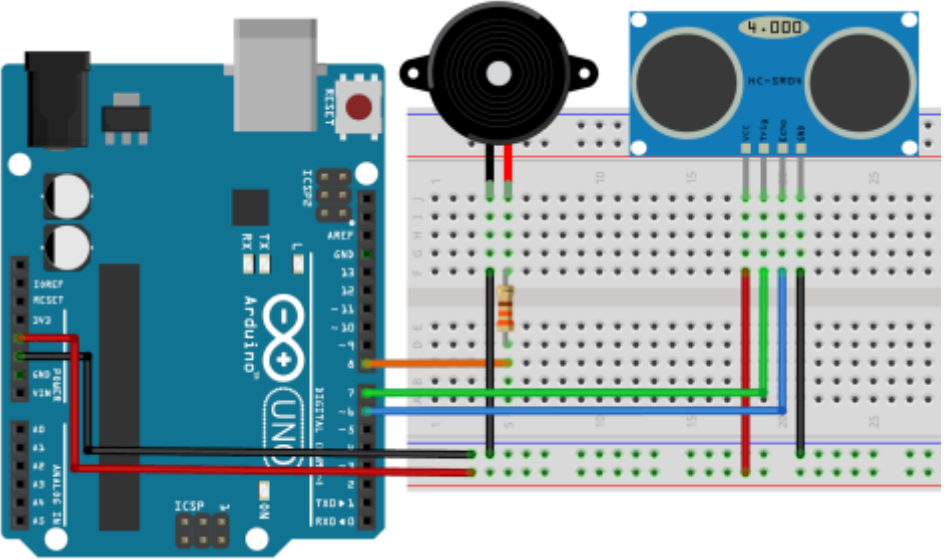
Ultrasonik Sensör ile Park Sensörü Yapımı

Gerekli malzemeler:

- Arduino Uno
- Breadboard
- 8 Adet Erkek-Erkek Jumper Kablo
- Buzzer
- 330 Ohm Direnç (Turuncu-Turuncu-Kahverengi)
- HC-SR04 Ultrasonik Sensör

Ultrasonik sensör uygulamamızda yeni bir kullanım örneği göreceğiz. HC-SR04 ultrasonik sensörü üzerinde bir tane ses gönderebilen, bir tane de ses algılayabilen metal kısımlar bulunuyor. Sensörden ses gönderildikten sonra eğer önünde cisim veya engelden yansıyor tekrar sensöre geliyor. Alıcı kısım yansıyan sinyali algılayarak ölçüm yapılabiliyor.

Mesafeyi ölçmek için sinyal gönderdikten itibaren geri gelene kadarki süreyi ölçüyoruz. Sesin havadaki hızını bildiğimiz için süre ve hızdan toplam mesafeyi hesaplayabiliyoruz. Hemen devremizi kurarak devam edelim.



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

Yazılım kısmında "#define" komutları ile kullanacağımız pinlere isimler veriyoruz. "maximumRange" ve "minimumRange" isminde "integer"(tamsayı) tipinde değişkenler tanımlıyoruz. "setup" kısmında giriş ve çıkış olacak pinleri ayarlıyoruz.

Ana program döngümüz çok kısa görünüyor. Bu kısımda ilk olarak mesafe fonksiyonuna gidiyoruz."long" türünde "duration" ve "distance" değişkenleri tanımlanıyor."long" Önceden kullandığımız "integer" gibi bir değişken. İçerisinde "integer" değişkenine göre çok daha büyük sayılar tutabilir. +2,147,483,647'den, -2,147,483,647'ye kadar içerisine atanabilmektedir. tutabilmektedir. Tutabildiği sayı hacminden dolayı tanımlandığında "integer" değişkenine göre 2 kat fazla hafıza kullanır. Değişken tanımladıktan sonra sensörün "trig" pinini yüksek ve alçak yaparak sensörün fiziksel ortama ses dalgası yollamasını sağlıyoruz. Ses dalgası yollandıktan sonra "pulseln(echoPin,HIGH)" komutu ile yolladığımız ses dalgasının cisimden yansiyıp geri gelmesini bekliyoruz. Bu beklediğimiz zamanıda "pulseln" komutu ile ölçebiliyoruz. Ölçtüğümüz bu değer "duration" değişkenine yazdırıyoruz. Süre ölçüldüğüne göre şimdi mesafeyi hesaplamaya geldi. Ölçtüğümüz süreyi sesin hızına göre mesafeye çevirmek için "58.2"ye bölüyoruz. Mesafe değerine ulaşıncaya, bu değer sensörün ölçebildiği minimum (2 cm) ve maksimum (400 cm) arasında değilse 0(sıfır) değeri ile dönüş yap diyoruz. İsteddiğimiz aralıktta 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 aralıklar ile, eğer sensör uzun mesafe algılıyor ise uzun aralıklar ile ötecek.