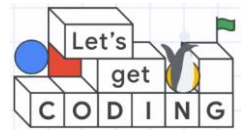


LYS-2019



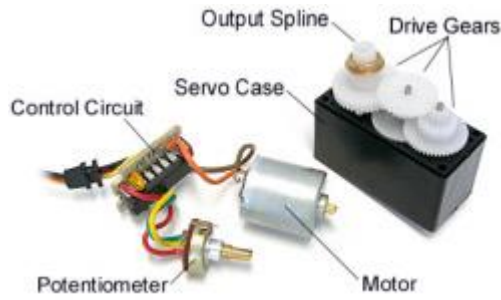
Arduino Uygulama Çalışması-6

Servo Sonar Radar

Servo Motor Nedir?

Servo, mekanizmalardaki açısal-doğrusal pozisyon, hız ve ivme kontrolünü hatasız bir şekilde yapan tahrik sistemi olarak tanımlanır. Yani hareket kontrolü yapılan bir düzendir. Servo motorlar, robot teknolojilerinde en çok kullanılan motor çeşidi olmakla birlikte, RC (Radio Control) uygulamalarda da kullanılmaktadırlar. RC Servo Motorlar ilk olarak uzaktan kumandalı model araçlarda kullanılmışlardır. Servolar, istenilen pozisyonu alması ve yeni bir komut gelmediği sürece bulunduğu pozisyonu değiştirmemesi amacıyla tasarlanmıştır.

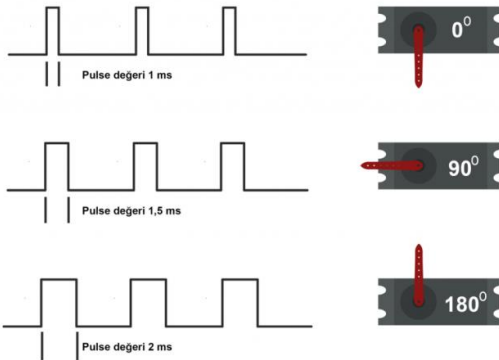
Servo Motor Çalışma Prensibi



Servo motorların içerisinde motorun hareketini sağlayan bir DC motor bulunmaktadır. Bu motorun dışında bir dişli mekanizması, potansiyometre ve bir motor sürücü devresi bulunmaktadır. Potansiyometre, motor milinin dönüş miktarını ölçmektedir. Servo içerisindeki DC motor hareket ettikçe potansiyometre döner ve kontrol devresi motorun bulunduğu pozisyon ile istenilen pozisyonu karşılaştırarak motor sürme işlemi yapar. Yani, servolar

diğer motorlar gibi harici bir motor sürücüyü ihtiyaç duymadan çalışmaktadırlar. Genellikle çalışma açıları 180 derece ile sınırlıdır fakat 360 derece çalışma açısına sahip özel amaçlı servo motorlar da vardır. Servolar genellikle 4.8-6V gerilim ile çalışmaktadırlar. 7.4V ve daha yüksek gerilimle çalışan servolar da bulunmaktadır.

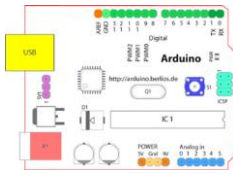
PWM (Pulse Width Modulation - Darbe Büyüklüğü Modülasyonu)



Darbe genişliği ile servo açısı kontrolü

Servo motorlar PWM (Sinyal Genişlik Modülasyonu) sinyal ile çalışmaktadırlar. Bu PWM sinyaller bir mikrokontrolörden veya uzaktan kumandanın sağlanabilmektedir. Servo, her 20 ms içerisinde bir puls değeri okumaktadır. Puls uzunluğu motorun dönüşünü belirlemektedir. Örnek olarak 1.5 ms'lik bir puls, motorun 90 derece pozisyonunu almasını sağlayacaktır (Nötr Pozisyon). Servolar hareket etmeleri için bir komut aldıklarında önce istenilen pozisyona hareket ederler, sonrasında ise o pozisyonda kalırlar. Servolar bulundukları pozisyonu korurken kendilerine dışarıdan bir güç uygulandığında bu güce direnirler.

Bulundukları konumu sonsuza kadar koruyamazlar, pozisyonlarını koruyabilmeleri için palsin tekrar edilmesi gerekebilir. Hareket etmeleri için gereken pals genişliklerinin minimumları ve maksimumları vardır ve bu değerler değişkendir. Fakat genellikle minimum pals genişliği 1 ms, maksimum pals genişliği ise 2 ms'dir.



LYS-2019



Arduino ile Servo Motor Kullanımı

```
#include <Servo.h> // Servo.h kütüphanemizi ekliyoruz.
Servo servomotor; // Servo objemiz.
int aci; // Servo mototrun açı değişkeni

void setup() {
  servomotor.attach(3); // servo motor arduinodaki pin tanımlaması.
}
void loop() {
  for(aci=0; aci<=180; aci++); // servo motorun açısı 0'dan 180'e
  {
    servomotor.write(aci); // Açıyı servo motora yaz.
    delay(10); //10 ms bekleme süresi
  }

  for(aci=180; aci>=0; aci--); // ters yöne döndür
  {
    servomotor.write(aci); // açıyı yaz
    delay(10); //10 ms bekleme süresi
  }
}
```

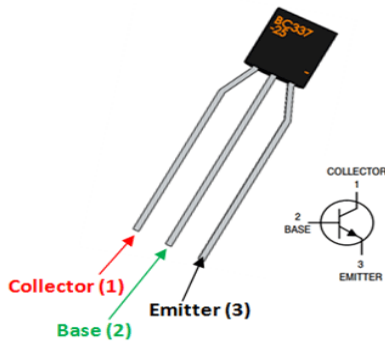
HC-SR04 Ultrasonik Mesafe Sensörü



Bu sensör, robotik projelerde Arduino ile kullanılan en popüler sensörlerden birisidir. Kullanımı oldukça kolaydır ve program kısmı doğru olduğu sürece 2cm – 400cm arası uzaklıkları düzgün bir şekilde ölçebilmektedir. Çalışma prensibi ise şu şekildedir: 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. Aşağıda verilen fonksiyonla basit olarak bu sensörden uzaklık bilgisi alabilirsiniz.

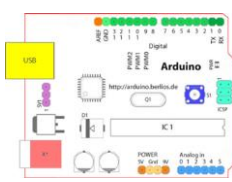
Arduino Sonar Radar Yapımı

Transistör küçük elektrik sinyalleri yükseltmek veya anahtarlama amacıyla

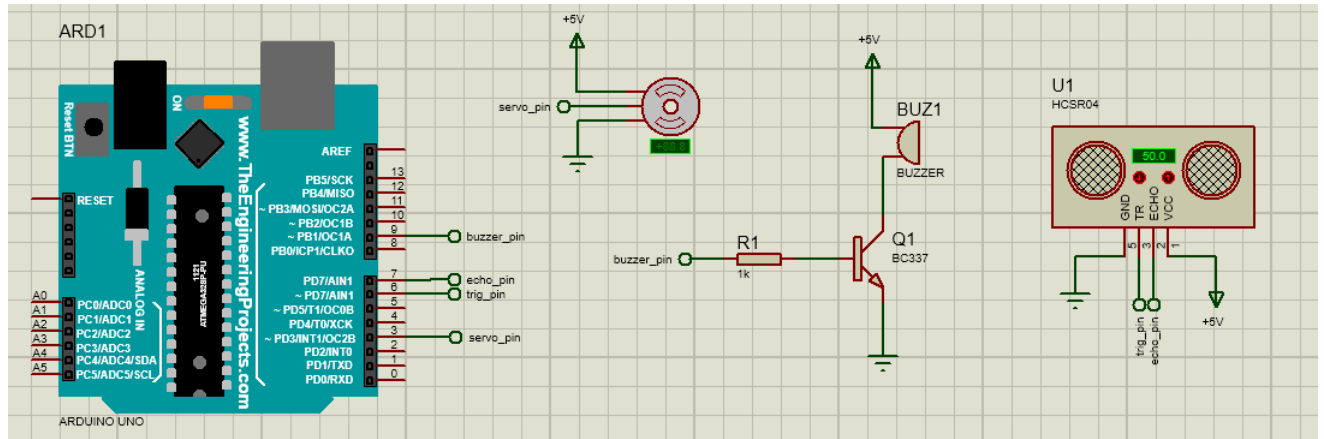


kullanabileceğimiz bir **yarı-iletken** devre elemanıdır. 3 veya daha fazla bacağı bulunan transistörün bacaklarından birisine uygulanan elektrik sinyali ile diğer bacakları arasındaki elektrik akımını kontrol edebiliriz. Yan tarafta devrede kullanacağımız BC337 isimli transistör şekli görülmektedir. Bu devre elemanının beyz(Base) ucuna sinyal uygulandığında kollektör-emiter arasından akım akışı olur ve kollektör ucuna bağlı

olan devre elemanı çalışır.



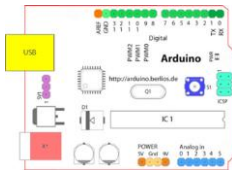
LYS-2019



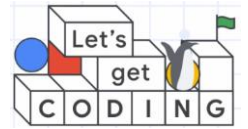
```

1 //Sonar Radar
2 //SG90 mikro Servo
3 //kırmızı +5V, kahve gnd, turuncu kontrol
4 #include <Servo.h>
5 Servo servo;
6
7 #define _sol 1
8 #define sag 0
9 #define _var 1
10 #define _yok 0
11
12 #define buzzer 9
13
14 //Notalarla ilgili tanımlamalar
15 const int Do = 262;
16 const int re = 294;
17 const int mi = 330;
18 const int fa = 349;
19 const int sol = 392;
20 const int la = 440;
21 const int si = 494;
22 const int doi = 523;
23 const int notalar[] = {Do, re, mi, fa, sol, la, si, doi};
24
25 int uzaklik = 0;
26
27 byte aci = 15;
28 bool yon = _sol;
29 boolean dur=false;
30 long zaman_print=0;
31
32 void setup() {
33
34     servo.attach(3);
35     Serial.begin(9600);
36     zaman_print=millis();
37
38 }

```

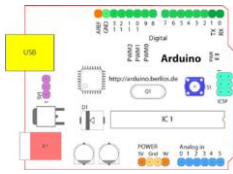


LYS-2019

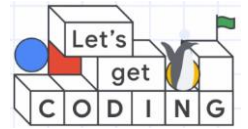


```
40 void loop() {
41
42     uzaklik = sonarSensOku();
43
44     if(!dur){
45         servo.write(aci);
46     }
47     if(uzaklik<150){
48         sonar_ses(uzaklik);
49         dur=true;
50     }else dur=false;
51
52     if (yon == _sol&&!dur){
53         aci += 5;
54     }
55     if (yon == sag&&!dur ) {
56         aci -= 5;
57     }
58     if (aci > 165) {
59         yon = sag;
60     }
61
62     if (aci < 15) {
63         yon = _sol;
64     }
65
66     if(millis()-zaman_print>1000){
67         Serial.println(uzaklik);
68         zaman_print=millis();
69     }
70
71 } //loop

73 int sonarSensOku() {
74
75     const int trigPin = 6;
76     const int echoPin = 7 ;
77
78     long duration = 0;
79
80     pinMode(trigPin, OUTPUT);
81     pinMode(echoPin, INPUT);
82
83     digitalWrite(trigPin, LOW);
84     delayMicroseconds(2);
85     digitalWrite(trigPin, HIGH);
86     delayMicroseconds(10);
87     digitalWrite(trigPin, LOW);
88
89     duration = pulseIn(echoPin, HIGH);
90     return (duration / 29 / 2); //cm
91 }
```



LYS-2019



```
106 void sonar_ses(int yap) {
107   if (yap < 50) {
108     Serial.println(yap);
109     tone(buzzer, Do);
110     delay(10);
111     noTone(buzzer);
112   }
113 }
114 if (yap > 50 && yap < 100) {
115   Serial.println(yap);
116   tone(buzzer, re);
117   delay(10);
118   noTone(buzzer);
119 }
120 }
121 if (yap > 100 && yap < 150) {
122   Serial.println(yap);
123   tone(buzzer, mi);
124   delay(10);
125   noTone(buzzer);
126 }
127 }
128 if (yap > 150 && yap < 200) {
129   tone(buzzer, fa);
130   delay(10);
131   noTone(buzzer);
132 }
133 if (yap > 200) noTone(buzzer);
134 }
135 }
```