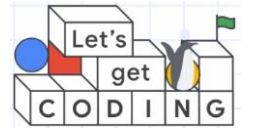


LYS-2019

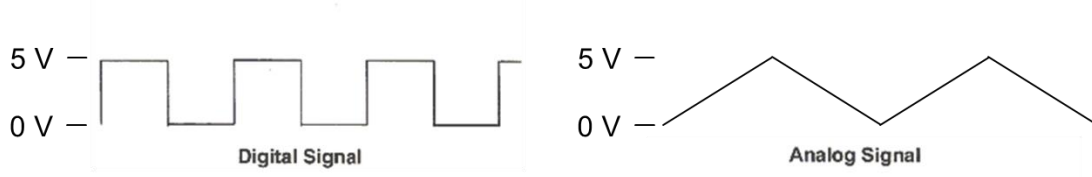


# Arduino Uygulama Çalışması-4

## Dijital-Analog Okuma

**Analog** : Zamana bağlı olarak sürekli devam eden sinyaller.

**Dijital** : Zamana bağlı olarak ayırık bir şekilde devam eden sinyaller. (1-0)



### Arduino'ya Dışarıdan Verilen Bilgiyi Okuma İşlemi

Arduino içersindeki mikrodenetleyicide bulunan dahili devreleriyle analog ve dijital bilgileri işleyebilir. Dijital bilgiler butonlar, reed röleler, anahtarlar ve dijital sensörlerden gelebilir. Analog veriler ise her türlü akım ve gerilim, analog çıkış veren sensörler vb. devre elemanları vasıtasıyla uygulanabilir. Arduino da 13 adet dijital ve 6 adet te analog giriş pini bulunmaktadır. Analog giriş pinleri de ayrıca dijital olarak kullanılabilir. Fakat dijital pinler analog giriş olarak kullanılamazlar.

### Arduino da Dijital Veri Okuma

Dijital veriler sadece 1 ve 0 olabilen verilerdir. İkili sayı sistemine uygun olarak rahatlıkla kodlanabilirler. Dolayısıyla herhangi bir dijital pinden dışarıya çıkış olarak 1 ve 0 lar gönderilebileceği gibi, içeriye de giriş olarak alınabilirler. Dijital 1'in gerilim karşılığı Arduino için 5V , 0'ın ise 0 voltur. Aynı zamanda 1, HIGH veya true ile ifade edilebilirken, 0 ise LOW veya false ile ifade edilebilir. Kodlama sırasında HIGH-true ve LOW-false aynı şekilde kullanılabilir. 5 Voltun sayısal karşılığı 1023 iken 0 volt, 0'dır (10 bit dijital veri).

### Buton-Anahtar Okuma



Butonlar dışarıdan Arduinoya dijital veri girişi yapabileceğimiz mekanik devre elemanlarıdır. İki uçları normalde açık iken butona basıldığında kısa devre, buton bırakıldığında ise tekrar açık devre olur. Bu özelliklerinden yararlanılarak basma anında iki ucu arasından sinyal geçişi sağlanacak şekilde devreye bağlanırlar. Bu

şekildeki butonların diğer adı push-buton dur.



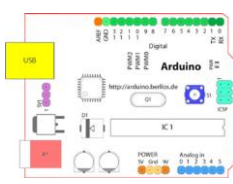
Buton sembolü.



Anahtarlar ise butonlardan farklı olarak bir kere basıldığında konum değiştirerek basıldıkları konumda kalırlar. Devreye butonlarla benzer şekilde bağlanırlar.



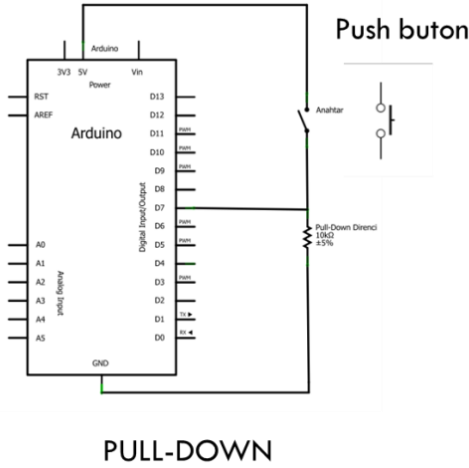
Anahtarın sembolü



LYS-2019

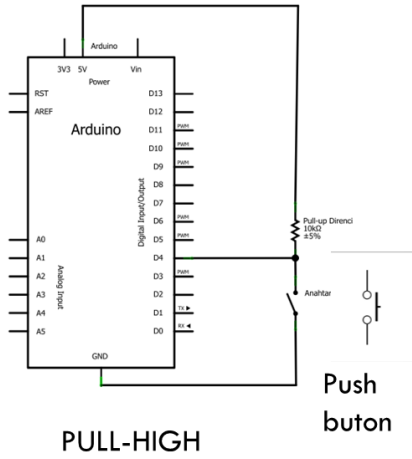


## PULL-DOWN Bağlantı



Bu bağlantı şeklinde anahtar yada butona normalde basılmazken bir direnç üzerinden giriş yaptığımız pin 0' a (GND) çekilir. Böylece butona yada anahtara basılmazken giriş pininde sürekli 0 (LOW, false) olması garanti altına alınmış olur. Butona yada anahtara basıldığında giriş pinine 5V uygulanır. Böylece giriş pininden 1 (HIGH-true) alınmış olur. Yazılımda basılı konumu bulmak için giriş pininin durumuna sürekli bakılarak, 1-HIGH-true olması durumu aranır.

## PULL-HIGH Bağlantı



Pull-down bağlantının tersi bir bağlantı şeklidir. Direnç bu sefer GND 'ye değil +5V 'a bağlanmıştır. Dolayısıyla giriş pininde normalde 1 vardır. Butona basıldığında ise giriş pini 0 olur. Yazılımda basılı konumu bulmak için 0-LOW-false durumu aranır.

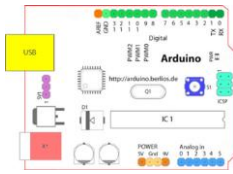
Dijital çıkış veren sensörlerde aynı şekilde Arduinoya bağlanabilirler. Bu sensörlerde sensörün normalde çıkış değeri ve algılama anındaki çıkış değerine göre yazılımda algılama bulunur.

## digitalRead(pin\_no) Komutu

Arduino da herhangi bir pinden dijital giriş okuma komutudur. Kullanılacak olan pin setup fonksiyonunda pinMode komutu ile giriş olarak yapılandırılmalıdır.

```
void loop() {  
  // Buton durumunu oku  
  buttonDurumu = digitalRead(butonPin);  
  
  /* Butona basıldığında butonun durumu HIGH olacaktır.  
   Bu durumda LED çıkışını HIGH yapıyoruz.  
   tersi durumda ise LOW yapıyoruz */  
  if (buttonDurumu == HIGH) {  
    digitalWrite(ledPin, HIGH);  
  }  
  else {  
    digitalWrite(ledPin, LOW);  
  }  
}
```

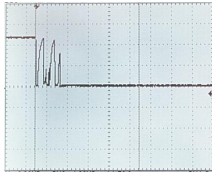
Yandaki örnekte komutlarda buton yada anahtar devreye pull-down olarak bağlanmıştır. Çünkü yazılımda if komutu ile giriş pininin HIGH olma anı aranmış ve HIGH olduğu anda butona basma algılanarak led yakılmıştır.



LYS-2019



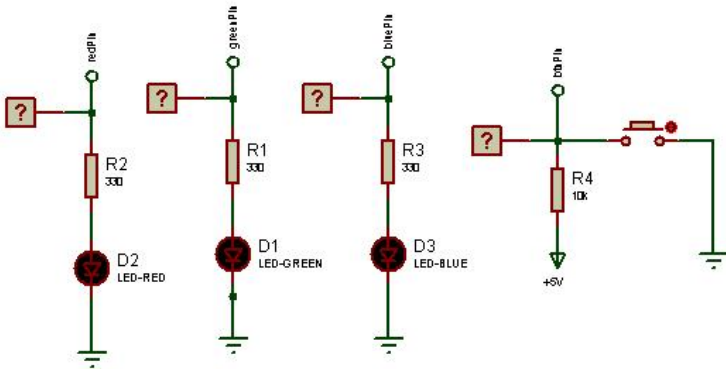
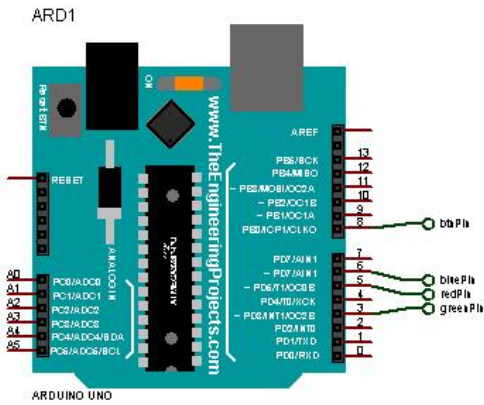
Diğer durumlarda led sönmük durumdadır.



Butonlarda sıçrama **sıçrama (bouncing)** adı verilen problemin önüne geçebilmek için de if komutunun hemen arkasından 250-350mS bir zaman gecikmesi delay komutu ile verilebilir.

```
if (digitalRead(butonPin)) {
    delay(250); digitalWrite(ledPin,
HIGH);
}
else {
    digitalWrite(ledPin, LOW);
}
```

Uygulama : Buton ve RGB Led



```
1 //Buton okuma uygulaması
2 #define redPin 5 //sabit değer tanımlama
3 #define greenPin 3
4 #define bluePin 6
5 #define butonPin 8

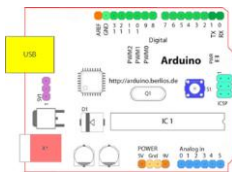
7 bool butonDurumu = false;
8 int renk_sec = 0;
9
10 void setup() {
11     pinMode(redPin, OUTPUT);
12     pinMode(greenPin, OUTPUT);
13     pinMode(bluePin, OUTPUT);
14     pinMode(butonPin, INPUT);
15 }

17 void loop() {
18     // Buton durumunu oku
19     if (digitalRead(butonPin) == LOW) {
20         delay(250);
21         renk_sec++;
22         //digitalWrite(ledPin, HIGH);
23     }
24     else {
25         //digitalWrite(ledPin, LOW);
26     }
27
28     if (renk_sec == 1) digitalWrite(redPin, HIGH);
29     if (renk_sec == 2) digitalWrite(greenPin, LOW);
30     if (renk_sec == 3) digitalWrite(bluePin, HIGH);
31     if (renk_sec == 4)
32     {
33         digitalWrite(bluePin, LOW);
34         digitalWrite(redPin, LOW);
35         digitalWrite(greenPin, LOW);
36         renk_sec = 0;
37     }
38 }
```

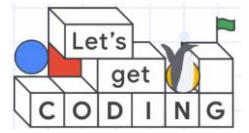
14. satırdada yer alan komut yerine ,

pinMode(butonPin, INPUT\_PULLUP);

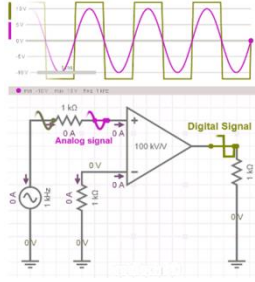
komutunu koyarak Arduinodaki buton pinini pull-up yapan 10K'lık direnci yerinden sökerek çalışmayı tekrar deneyiniz.



LYS-2019



## Analog Giriş Sinyali Okuma



Analog sinyallerin mikrodenetleyiceler tarafından işlenebilmesi için dijitale döndürülmesi gerekmektedir. Bunun için ADC (Analog Dijital Çevirici) devreleri kullanılmaktadır. Arduino UNO üzerinde 6 adet analog giriş (A0-A5) bulunmaktadır. UNO'nun ADC'si 10 bit çevrim yapmaktadır. Basit olarak 0-5V arasında olan gerilimleri 0-1023 arasındaki sayılara dönüştürmektedir. Burada 0, 0V karşılık gelirken 1023 değeri 5V karşılık gelir.

10Bitte maksimum elde edilebilecek sayı değeri, n bit sayısı olmak üzere;

$$2^n - 1 = 2^{10} - 1 = 1023 \text{ tür.}$$

Buradan anlaşılacağı üzere 5V sıfırda dahil olmak üzere 1024 eşit parçaya bölünmüştür. Yani sürekli olan analog sinyal ayırık bir şekilde getirilerek dijitale dönüştürülmüştür. Buna göre;

$$\frac{5V}{1024} = 0,00488V (4,8mV) \text{ yapar. Yani en küçük okuyabileceğimiz gerilim değeri 4,88 mV olur.}$$

ADC ile sayısal çevrilen gerilim değeri tekrar bulunmak istenirse aşağıdaki formül kullanılabilir.

$$\text{Gerilim} = \frac{\text{ADC Değer}}{1023} * 5V$$

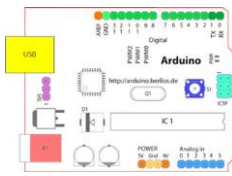
Örn : ADC tarafından okunan değer 200 ise;

$$\text{Gerilim} = (200/1023) * 5V = 0,977 V \text{ olur.}$$

Bir diğer ifade ile;

$$\text{Gerilim} = 4,88mV * 200 = 976mV \text{ olur.}$$

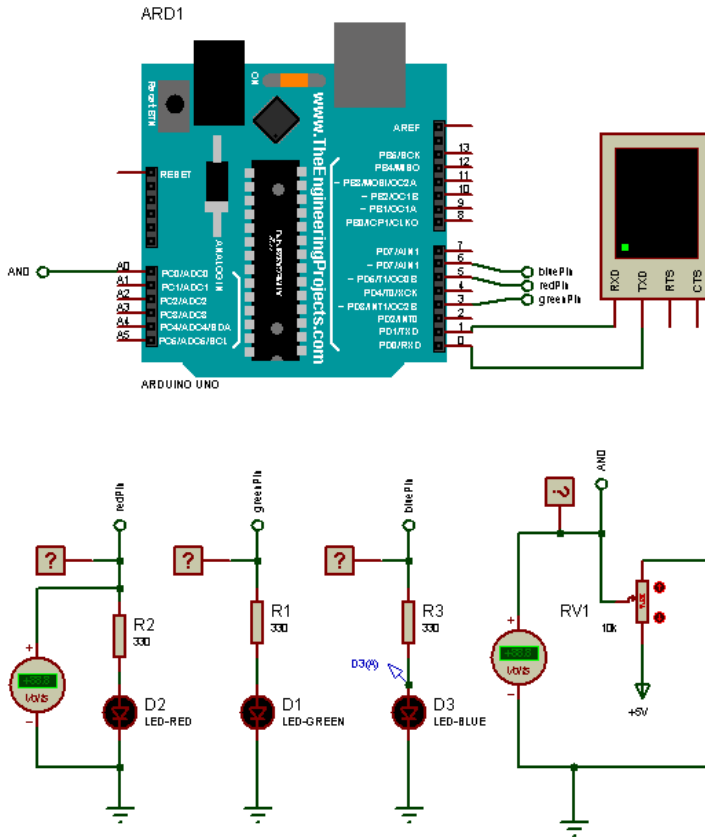
**Not:** Kodlama esnasında ondalıklı sayı kullanmak için ondalık kısım tam kısımdan nokta ile ayrılır. Aşağıdaki kodda 8. Satırda mantıksal bir hata yapılmıştır. Hatayı bulunuz.



LYS-2019



## Uygulama : Analog Değer Okuma



```

1 //Buton okuma uygulaması
2 #define redPin 5 //sabit değer tanımlama
3 #define greenPin 3
4 #define bluePin 6
5 #define butonPin 8

6
7 bool buttonDurumu = false;
8 int analogGiris = 0, gerilimDegeri=0;
9
10 void setup() {
11     // Seri Haberleşmeyi başlat
12     Serial.begin(9600);
13     Serial.println("Basla...");
14
15     pinMode(redPin, OUTPUT);
16     pinMode(greenPin, OUTPUT);
17     pinMode(bluePin, OUTPUT);
18 }

19
20 void loop() {
21     analogGiris = analogRead(A0);
22     gerilimDegeri = (analogGiris / 1023.0) * 5.0;
23     Serial.print("Analog deger:");
24     Serial.println(analogGiris);
25     Serial.print("Gerilim: ");
26     Serial.print(gerilimDegeri);
27     Serial.println(" V");
28     delay(1000); // Bir saniye bekle
29 }

```