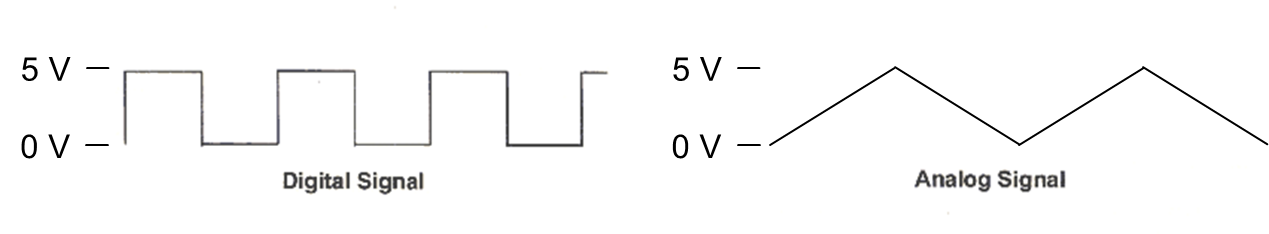
Arduino Uygulama Çalışması-4

### Dijital-Analog Okuma

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

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



**Arduino’ya Dışarıdan Verilen Bilgiyi Okuma İşlemi**

Arduino içersindeki mikrodenetleyicide bulunan dahili devreleriyle analog ve dijtal bilgileri işleyebilir. Dijtal bilgiler butonlar, reed röleler, anahtarlar ve dijtal sensörelerden 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 kullanıulabilir. Fakat dijital pinler analog giriş olarak kullanılamazlar.

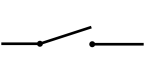
**Arduino da Dijital Veri Okuma**

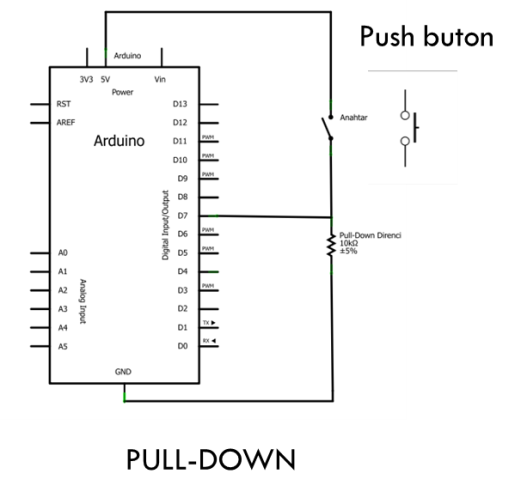
Dijital veriler sadece 1 ve 0 olabilen verilerdir. İkili sayı sistemine uygun olarak rahatlıkla kodlanabilirler. Dolayısıyla herhangi bir dijital pinden dışarıya çıkış olarak 1 ve 0 lar gönderilebileceği gibi, içeriye de giriş olarak alınabilirler. Dijital 1’in gerilim karşılığı Arduino için 5V , 0’ın ise 0 volttur. 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.  Buıton sembolü.

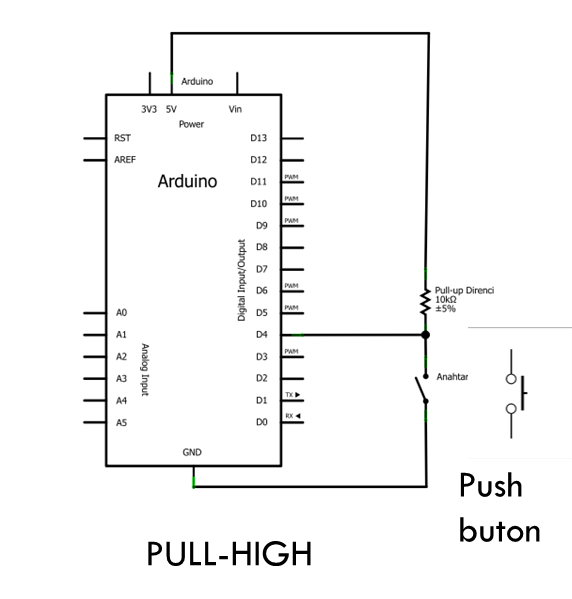
Anahtalar ise butonlardan farklı olarak bir kere basıldığında konum değiştirerek basıldıkları konumda kalırlar. Devreye butonlarla benzer şekilde bağlanırlar.

Anahtarın sembolü

**PULL-DOWN Bağlantı**

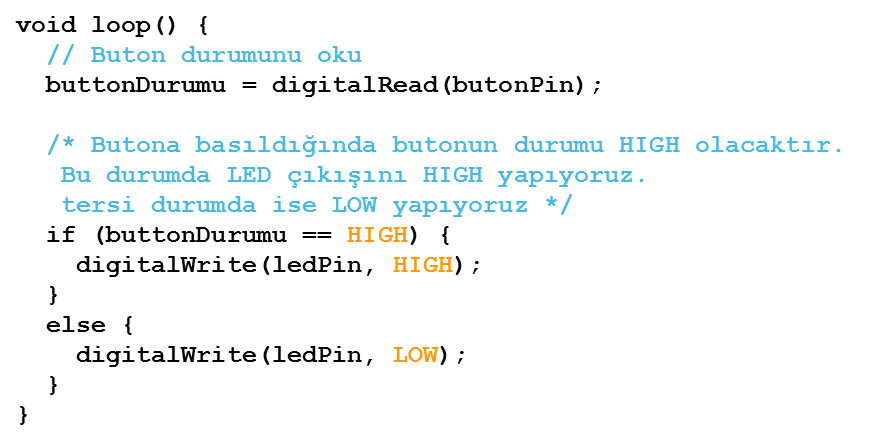
Bu bağlantı şaklinde anahtar yada butona normalde basılmazken bir direnç üzerinden giriş yaptığımız pin 0’ a (GND) çekilir. Böylece butona yada anahtara basılmazken giriş pininde sürekli 0 (LOW, false) olması garanti altına alınmış olur. Butona yada anahtara baslıdığında giriş pinine 5V uygulanır. Böylece 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**

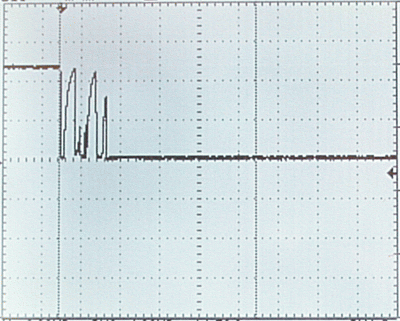
Pull-down bağlantının tersi bir bağlantı şeklidir. Direnç bu sefer GND ‘ye değil +5V ‘a bağlanmıştır. Dolayısıyla giriş pininde normalde 1 vardır. Butona basıldığında ise giriş pini 0 olur. Yasılımda basılı konumu bulmak için 0-LOW-false durumu aranır.

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

**digitalRead(pin\_no) Komutu**

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

Yandaki örnerk 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. Diğer durumlarda led sönük 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)) {**

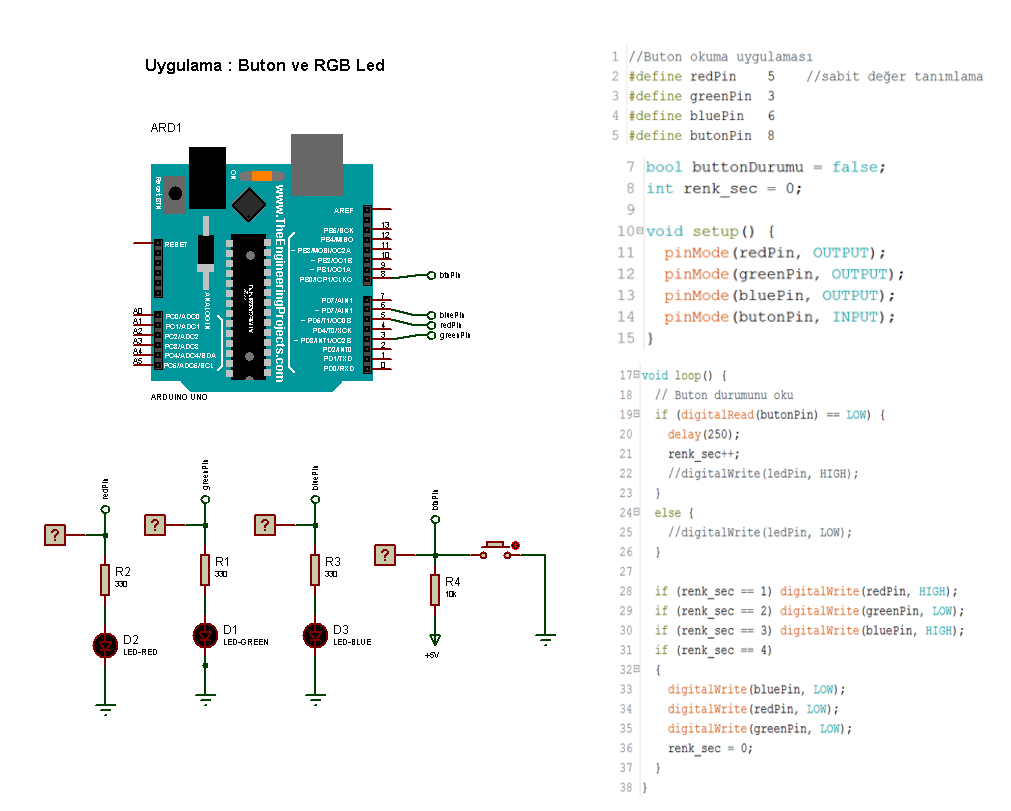
**delay(250); digitalWrite(ledPin, HIGH);**

**}**

**else {**

**digitalWrite(ledPin, LOW);**

**}**

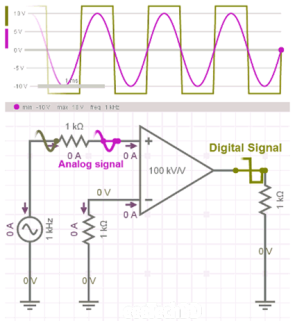
****

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.

**Analog Giriş Sinyali Okuma**

Analog sinyallerin mikrodenetleyiceler tarafından işlenebilmesi için dijitale dödüştürülmesi gerekmektedir. Bunun için ADC (Analog Dijital Çevirici) devreleri kullanılmaktadır. Arduino UNO üzerinde 6 adet aanlog 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 saysı olmak üzere;

-1 =-1=1023 tür.

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

= 0,00488V (4,8mV) yapar. Yani en küçük okuyabileceğimiz gerilim değeri 4,88 mV olur.

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

Gerilim =

Örn : ADC tarfından okunan değer 200 ise;

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

Bir diğer ifade ile;

Gerilim =4,88mV \* 200= 976mV 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.

