**GELİŞMİŞ BUTON KONTROLÜ**

**16**

Önceki derslerimizde, ledin buton ile kontrolünü uygulamasını gerçekleştirmiştik. Ancak mikrodenetleyicinin çok hızlı çalışması ve butonda oluşan elektriksel arktan dolayı stabil çalışan bir uygulama değildi.

**AMAÇ:**  Bu uygulamanın amacı, Buton yardımı ile bir ledin kontrolünü butona tek tıklama ile sağlayacağız. Böylece diğer sensör ve aygıtları kontrol ederken butona tıklandığında istenilen işlemin tam doğrulukta çalışmasını sağlayacağız.

### **16.1 Buton ile Ledi yakıp söndürme**

Bir Ledin buton yardımıyla istenildiği gibi yakıp söndürülmesi doğru gerçekleştirilebildiğinde güzel bir örnektir. Çünkü bir Ledin yakılıp söndürülmesi farklı cihazların açılıp kapatılması ile aynı işleme tabidir. Bir ledi yakıp söndürme komutu başka bir şeyi açıp kapatmak içinde kullanılabilecektir. Örneğin bluetooth veya wifi yardımı ile bir şeyi kontrol etmek için buton yardımı ile kontrol işlemi için aynı kodlar kullanılabilecektir.

### **16.2 Kesme veya Buton durumunu kontrol etme**

Ledi yakmanın birçok yolu var. Tek butonla, çift butonla veya tuş takımı ile kontrol sağlanabilir. Ancak bunlarında ötesinde kullanılacak kodların yöntem ve farklılıkları vardır. Genellikle bir buton ile kontrol işlemi sağlanırken Kesme veya butonun durumunu kontrol etme yöntemi kullanılmaktadır.

**Buton Durumunu Kontrol etme**

Bu yöntemde digitalRead komutu ile kontrol işlemi sağlanır. Loop() içerisinde her daim butonun bağlı olduğu pinin dijital durumu hakkında bilgi alınır. Bu yöntemde butonun durumu hakkında bilgiye ancak kontrol ettiğimizde ulaşıyoruz. Bu durum; bize gelecek bir postanın ulaşıp ulaşmadığını kontrol etmek için, kapıdaki posta kutusunu sıklıkla kontrol etmemize benzetilebilir.

**Kesme (Interrupt)**

Adından da anlaşılacağı gibi kesmeler, mevcut işlemin kesintiye uğradığı ve yeni bir işlemin gerçekleştirildiği yerdir.

nunla ilgili Kullanıcı butona tıkladığında butonun iki ucu kısa devre olur ve böylece akım akmaya başlar. Kullanıcı butondan elini çektiğinde devrenin önceki durumuna dönmesi için pull-up/pull-down direnç sistemi kullanılır. Burada direnç kullanmamızın sebebi, butona basıldığında 5V kaynağımızdan gelen gerilim buton üzerinden Arduino dijital pinine ulaşarak butonun aktif edildiğini yani HIGH değerini gönderiyor. Aynı şekilde butona basılmadığında ise DEAKTİF olduğunu, butonun basılmadığını yani LOW sinyalini göndermesi için butona toprak terminalini de bağlamamız gerekiyor ki buton basılmadığında 0V u Arduinomuza göndersin. Eğer bu toprak terminalini butonumuza bağlamazsak aktif edildiğinde 5 Vu iletmesine rağmen aktif edilmediğinde Arduinoya ne göndereceğini bilemediğinden, LEDimiz Arduinonun içerisindeki “Elektronik Ortam Gürültüsü” denilen sebepten ötürü 0-5V arasındaki herhangi bir değeri alarak LEDimiz üzerindeki kontrolümüzü sağlayamamıza neden olur.

Ancak tamda burada bir sorunumuz oluşuyor. Butonun bacaklarına hem 5v u hem de toprağı arada seri herhangi bir yük olmadan bağlarsak bu butona basıldığında kısa devreye neden olacaktır. Kısa devre durumunda ise kısa devre akımı teorik olarak sonsuza pratikte ise güç kaynağınızın verebileceği maksimum değere kadar aniden artarak muhtemelen birkaç saniye içerisinde eğer koruma devresi mevcut değilse güç kaynağınızın veya Arduinonuzun yanmasına sebep olabilir.

Bu nedenle butonun ayaklarından toprak hattına seri olarak Pull-Down (aşağı çekme) adı verilen bir direnç bağlamamız gerekiyor. Bu sayede butona basılmadığında toprak hattı direnç üzerinden Arduinoya bağlanmış olmasına rağmen buton tetiklendiğinde kısa devre olmadan 5V, daha az dirençli yolu seçerek toprak hattı yerine Arduino pinine HIGH sinyalini göndermekte ve kısa devre önlenmektedir.

Eğer kodumuzu tersten yazarsak yani dijital girişe buton basılmadığında 5v (HIGH) ,buton basıldığında ise 0v (LOW) sinyali verdirmek için ise Pull-Up (yukarı çek) isim verilen direnci 5v kaynağına seri olarak bağlamak gerekiyor. Bu durumda LED normalde yanacak. Butona basıldığında ise sönecektir.

### **08.2 Pull-Down Direnç**



Butona basıldığında 5 Volt arduinonun input ayağına ulaşır. Fakat butondan elimizi kaldırdığımızda arduinoun pininde 5 volt gerilimi kalmaktadır. Bu durumdan kurtulmak için genellikle 10K ohm değerinde bir direnç arduinoun input ayağından toprağa bağlanır.

### **08.3 Pull-Up Direnç**



Butona basılmadığı durumlarda arduinonun input ayağı 5V’dur. Butona basıldığında akım arduinonun input ayağı yerine toprağa ulaşmaktadır. Pull-Down direnç sisteminin tam tersi şeklinde çalışmaktadır. Direncin konulma nedeni butona basıldığında 5 Voltun doğrudan toprağa (- uça) ulaşmasını engellemektir. Genellikle 10K değerinde direnç kullanılmaktadır.

Pull-Up Direnç

### **08.4 digitalRead Komutu**

Dijital pinin aktif mi pasif mi olduğunu yani HIGH mı LOW mu olduğunu belirleriz.

Eğer bir kontrolörle işlem yapacaksak ona gelen bilgiyi denetleyebilmek gerekir. Gelen bilgiler yorumlanarak çıkış verilir. Peki, gelen bilgi nasıl okunur İki farklı türde okuma yapabiliriz. Bunlar digital ve analog okumadır.

Digital okuma işleminde Arduino nun bacağına logik 1(+5V) mi yoksa logik 0 (0V) mı geldiğini öğrenmemizi sağlar.

digitalRead(5); // 5 nolu pinde 0V veya 5V olup olmadığı

a=digitalRead(5); // Bir değişkene aktarmak isteyebilirsiniz;

if(digitalRead(buton) ==HIGH) // veya sorgulamak isterseniz;

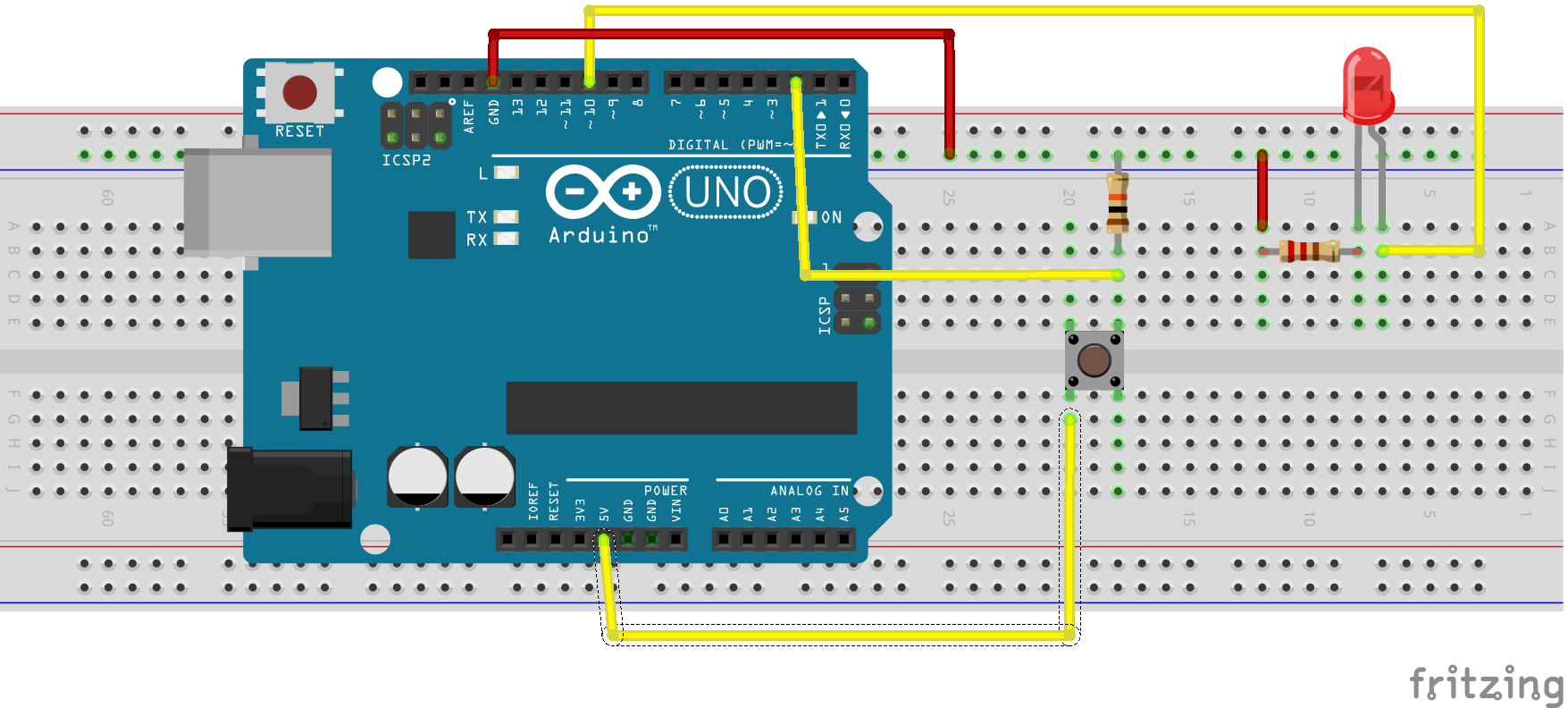
pinMode(butonPin, INPUT); // Butonun bağlı olduğu pini giriş olarak ayarlıyoruz.

Kaçırmamanız gereken kullanacağınız bu bacağı muhakkak pinMode komutu ile giriş olarak tanımlamanız gerekir.

### **08.4 Devrede Kullanılan Malzemeler:**

* 1 X Arduino UNO
* 1 X led
* 1 x 220 Ω (LED için)
* 1 x 10K Ω (Buton için)

### **08.5 Devrede Şeması:**



### **08.6 Kodlama:**

const int butonPin = 2; // Butonun pin numarası

const int ledPin = 10; // Ledin bağlandığı pin numarası

int butonDurum = 0; // Butonun durumu (basılıp basılmadığı)

void setup()

{

pinMode(ledPin, OUTPUT); // Ledin bağlı olduğu pini çıkış olarak ayarlıyoruz.

pinMode(butonPin, INPUT); // Butonun bağlı olduğu pini giriş olarak ayarlıyoruz.

}

void loop()

{

butonDurum = digitalRead(butonPin); //butonun durumunu butonDurum değişkenine atıyoruz

if (butonDurum == HIGH) { //eğer butona basılmışsa, butondan 5Vsinyali geliyorsa

digitalWrite(ledPin, HIGH); // Ledi yak

}

else { //eğer basılmamışsa

digitalWrite(ledPin, LOW); //ledi söndür

}

}

**Dahili Pull\_Up Direnç ile kodlama (2 Nolu pine bağlı buton ile 13 nolu pine bağlı ledin kontrolü)**

Arduino Digital pinleri dahili Pull-Up direncine sahiptir.

**Hatırlatma:** Pull\_Up direnç de butona basılmadığında Arduino’nun 2 nolu pinine 5V gerilim uygulanır.

void setup() {

pinMode(2, INPUT\_PULLUP); // Dahili PullUp direncini kullanmamızı sağlayan koddur.

pinMode(13, OUTPUT);

}

void loop() {

int durum = digitalRead(2); // 2 nolu pindeki butonun durumunu okur.

if (durum == HIGH) // Eğer buton basılı durumda ise

{

digitalWrite(13, LOW); // Ledi söndür  
}

else {  
digitalWrite(13, HIGH); //Ledi yak  
}

}

**Uygulama:**  Programın çalışmasını kısaca açıklayınız.

*int butonPin = 2; //2 nolu pini butonPin olarak isimlendirdik*

*int ledPin = 12; //12 nolu pini LedPin olarak isimlendirdik*

*int butonsayac = 0; //butona kaç kere basıldığını yazıcağımız değişkenimiz*

*int butondurum = 0; //butonun basılıp basılmadığını atadığımız değişkenimiz*

*int sonbutondurum = 0; //butonun en son basılıp basılmadığını atadığımız değişkenimiz*

*void setup() {*

*pinMode(butonPin, INPUT); //buton pinini giriş olarak atadık*

*pinMode(ledPin, OUTPUT); //ledpinini çıkış olarak atadık*

*}*

*void loop() {*

*butondurum = digitalRead(butonPin);*

*if (butondurum != sonbutondurum) { //eğer şuanki buton durumu son buton durumundan*

*//farklıysa yani pasiften aktif olmuşsa*

*//veya aktiften basif olmuşsa*

*if (butondurum == HIGH) { //ve eğer buton durumu aktifse*

*butonsayac++; //butonsayacını 1 arttır*

*}*

*delay(50); // çift basım hatasını gidermek için 50ms bekle*

*}*

*sonbutondurum = butondurum; //butondurumunu son buton durumuna kaydet*

*if (butonsayac % 2 == 0) { //eğer butona 2 kes basılmışsa aktif et ve 0 a eşitle*

*digitalWrite(ledPin, HIGH); //ledi yak*

*} else {*

*digitalWrite(ledPin, LOW); //4 kere basılmamışsa ledi yakma*

*}*

*}*

……………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………………………

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ÖĞRENCİNİN** | | **DEĞERLENDİRME** | | | | | | |
| Adı | : | **Bilgi (20)** | **Düzen (20)** | **İş Sırası (20)** | **Sonuç (20)** | **Süre (20)** | **Toplam** | |
| Soyadı | : | **Rakam** | **Yazı** |
| Sınıfı | : |  |  |  |  |  |  |  |
| No | : | **Öğretmenin Adı:** | | | | | **İmza:** | |