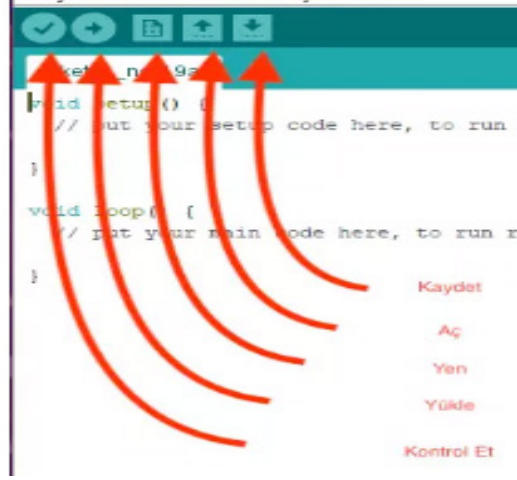


## 6.2. Arduino IDE Menüleri



Şekil 5.12. Ara Yüz Genel Görünüm

**Kaydet:** Projeyi kaydetmek için kullanılır.

**Aç:** Önceden kayıtlı olan projeyi açmak için kullanılır.

**Yeni:** Yeni proje oluşturmak için kullanılır.

**Yükle:** Yazılmış olan projeyi karta yüklemek için kullanılır.

**Kontrol et:** Proje içerisinde yazılan kodlarda hata olup olmadığını kontrol etmek için kullanılır.

Kodlamaya başlamadan önce projeyi kaydetmek gerekir. Kaydedilen projeye gerekli kodlar yazıldıktan sonra **Kontrol Et** butonu ile yazım hataları denetlenebilir. Hata yoksa **Yükle** butonu ile Arduino kartına yükleme işlemine geçilir.

## 6.3. Arduino Programlama Giriş

Geliştirme ortamı ile Arduino programları yazılıp, derlenerek kartlar üzerine yüklenebilir. Arduino programlamada C / C++ /Java temelli bir dil yapısı kullanılır. Kütüphaneler sayesinde donanım seviyesine inmeye gerek yoktur, kod yazımında küçük büyük harf hassasiyeti vardır.

Arduino IDE programları 3 bloktan oluşur.

```

//Tanımlamalar
Void setup() {
// Kurulum için gerekli komutlar yazılır. (Bir defa çalışır)
}
Void loop() {
/* Ana program kodları buraya yazılır.
(Sürekli çalışır) */
}

```

1. Blok  
2. Blok  
3. Blok

Şekil 6.12. Arduino IDE Program Blokları

**Tanımlamalar:** Bu kısımda değişkenler tanımlanabilir, kütüphaneler programa dâhil edilebilir veya pinlere isim verme gibi işlemler yapılabilir. Örneğin; program içerisinde sürekli olarak kullanılan bir pin için, 13. pin şeklinde kullanmak yerine; 13. pine LED bağlanmış ise “LED-13” kullanılarak istenildiğinde kolaylıkla değiştirilebilir.

**void setup():** Arduino’ya enerji verildiğinde veya yeniden başlatıldığında bu bölüme yazılan kodları program 1 kere okur. Mesela pinlerin giriş mi yoksa çıkış mı olacağı bir kere buraya yazılır ve hafızada tutulur bir daha o kodun yazılmasına gerek kalmaz

**void loop():** Bu bölüme yazılan kodlar sürekli okunur ve kodlara göre pinlere çıkış verir yada sensör değerleri okunur.

Süslü parantez ({} ) aralarında kullanılan “//” simge, program sayfası içerisinde herhangi bir yere yazılabilir ve yazıldığı satır açıklama satırıdır. Açıklama satırları program yazarken hatırlatma amaçlı bazı notlar alınmasına yarar. Derleyici, “// “ işaretinin olduğu satırı dikkate almaz.

Açıklama amacıyla kullanılan diğer bir yöntem ise “/\*” ve “\*/” işaretleridir. Program içerisinde istenilen yere “/\*” işareti koyulduktan sonra gerekli notlar ya da açıklamalar yazılır. Sonrasında “\*/” işareti konularak açıklama satırları bitirilir. Derleyici bu işaretler arasına yazılan hiçbir yazıyı dikkate almaz, sadece kullanıcıya ait satırlardır. Birden fazla açıklama satırı yazmak için bu yöntem tercih edilir.

## 6.4. Kodlama ile İlgili Bazı Kurallar

- Komutlar yan yana aynı satıra yazılabileceği gibi alt alta da yazılabilir. Ancak programın anlaşılabilirliği açısından alt alta yazmak faydalı olacaktır.
- Komutların sonuna (;) noktalı virgöl konulur.
- Programın başında tanımlamalar kısmında kullanılacak kütüphaneler varsa **#include** komutu ile programa dâhil edilir.
- Türkçe karakter kullanılmamalıdır. Fakat açıklama satırları içerisinde (derleme işlemine dâhil edilmediğinden) kullanılabilir.

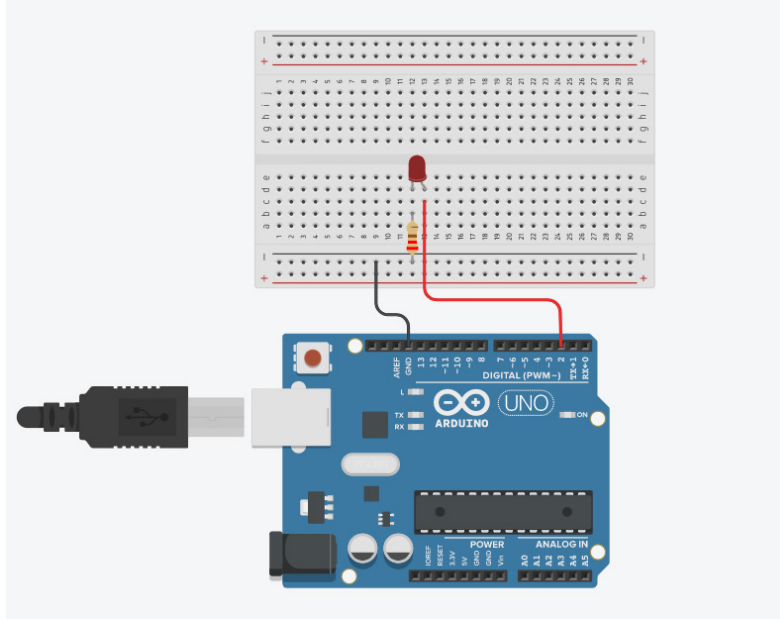
## 6.5. Arduino IDE Uygulamaları ile Komutlar

Aşağıda Arduino uygulamaları ile birlikte Arduino IDE komut satırları verilmiştir. Böylelikle kodların tanımlanması, kullanım şekilleri ve devre üzerindeki etkileri gösterilmiştir.

### 6.5.1. Dijital Çıkış (Led Yakma)

#### Uygulama 1

Arduino Uno ile 1 Led'i 1 saniye yakıp 1 saniye söndürme.



Şekil 6.13. Led Yakma Devre Şeması

Tablo 6.1. LED yakma devresinde kullanılan malzemeler

Ad	Miktar	Bileşen
U1	1	Arduino Uno R3
D1	1	Kırmızı LED
R1	1	330 $\Omega$ Rezistör

Ledin eksi (-) ucunu Arduino kartının GND pinine Ledin artı (+) ucunu direnç üzerinden Arduino kartının 2 numaralı dijital ucuna bağlanır. Direncin görevi; ledin üzerinden geçen akımı sınırlandırmak için kullanılır. Direnç kullanılmadığında Led'in bozulması olası bir durumdur.

```
void setup()
{
  pinMode(2, OUTPUT);
}
void loop()
{
  digitalWrite(2, HIGH);
  delay(1000);
  digitalWrite(2, LOW);
  delay(1000);
}
```

Şekil 6.14. Led Yakma Komut Satırları

## Kullanılan komutlar;

**void Setup():** Arduino programında donanımsal ayarların yapıldığı bölümdür. Giriş çıkışların ayarlanması, varsa haberleşme ayarları bu bölümde yazılır. Program çalıştırıldığında sadece bir defa çalışır.

**pinMode(pinNo,[INPUT,OUTPUT,INPUT\_PULLUP]):** pinNo ile numarası belirtilen dijital pinin giriş veya çıkış olarak ayarlanmasını sağlar. Pin numarasından sonra OUTPUT yazılırsa pin çıkış olarak ayarlanır. INPUT seçilirse pin giriş olarak ayarlanır.

**void loop():** Arduino programında ana program bloğudur. Bu blok sürekli tekrar eden bloktur. Arduino programında sürekli tekrarlanan görevler bu blok içerisinde tanımlanır.

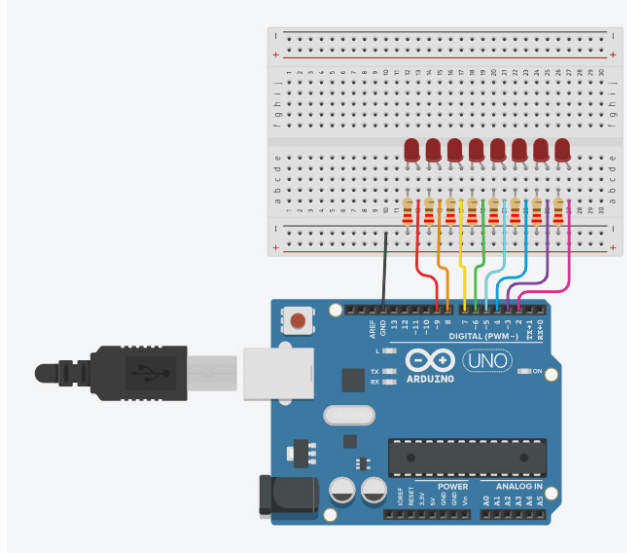
**digitalWrite(pinNo, [HIGH,LOW]):** Önceden çıkış olarak tanımlanan ve **pinNo** ile belirtilen dijital pini HIGH seçeneği ile 5 volt, LOW ile 0 Volt seviyesine çekilmiş olur. Bu programda HIGH ile LED ışık verir, LOW ile kırmızı led söner. Arduino dijital pinleri 5 volt ve 40 mili amper çıkış akımı verebilir. Bunun üzerindeki değerlerde zarar görebilir. Bu uygulamada devreye bağladığımız kırmızı LED 2 volt seviyesinde bir voltaj ile çalıştığı için arduino ile LED arasına veya LED ile Arduino GND bağlantı noktası arasına bir direnç (220 ohm) bağlanmaktadır.

**delay(süremilisaniye):** Verilen süre kadar programın beklemesini sağlar. Burada 1000 yazarak ledin 1 saniye yanıp, 1 saniye sönmesi sağlanır.

# BÖLÜM 6

## Uygulama 2

Arduino ile 8 adet ledi sıra ile yakıp hepsinin birden söndürülmesi. Bu işlem sürekli devam eder.



Şekil 6.15. 8 Adet Ledi Sıra ile Yakıp Hepsini Birden Söndürme Devresi

Tablo 6.2. 8 LED'i sırasıyla yakma devresinde kullanılan malzemeler

Ad	Miktar	Bileşen
U1	1	Arduino Uno R3
D1 D2 D3 D4 D5 D6 D7 D8	8	Kırmızı LED
R1	1	330 $\Omega$ Rezistör

Ledlerin eksi (-) uçları birleştirilerek direnç üzerinden GND pinine bağlanır, Ledlerin artı uçları Arduino'nun sırasıyla 9-8-7-6-5-4-3-2 numaralı dijital pinlerine bağlanılır.

```
void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
}
void loop()
{
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  delay(1000);
  digitalWrite(2, HIGH);
  delay(1000);
  digitalWrite(3, HIGH);
  delay(1000);
  digitalWrite(4, HIGH);
  delay(1000);
  digitalWrite(5, HIGH);
  delay(1000);
  digitalWrite(6, HIGH);
  delay(1000);
  digitalWrite(7, HIGH);
  delay(1000);
  digitalWrite(8, HIGH);
  delay(1000);
  digitalWrite(9, HIGH);
  delay(1000);
}
```

Şekil 6.16. 8 Adet Ledi Sıra ile Yakıp Hepsini Birden Söndürme Kod Satırları

## Uygulama 3

Uygulama 2 Şekil 6.17'deki program kodları, döngü kullanarak çok daha kısa yazabilir. Özellikle tekrarlama sayısı belli olan uygulamalarda FOR döngüsü tercih edilebilir.

### Kullanılan komutlar;

**For (döngü değişkeni = başlangıç değeri; koşul; döngü değişkeni artış miktarı):** Bu döngü verilen başlangıç değerinden başlayıp verilen artış miktarı ile artarak koşul sağlandığı sürece işlemi tekrar eder.

For döngüsünü kullanarak uygulama 2'deki program aşağıdaki gibi kısaltılabilir.

```
int i = 0;
void setup()
{
  for(i = 2;i<=9;i++){
    pinMode(i,OUTPUT);
  }
}
void loop()
{
  for(i = 2;i<=9;i++){
    digitalWrite(i,LOW);
  }
  delay(1000);
  for(i = 2;i<=9;i++){
    digitalWrite(i,HIGH);
    delay(1000);
  }
}
```

Şekil 6.18. 8 Adet Led Sıra ile Yakıp Hepsini Birden Söndürme Kodlarının For Döngüsü ile Yazılması

## Uygulama 4

Uygulama 2'deki devrede ledler sırasıyla sona kadar yanacak ve tekrar sondan başa doğru sırasıyla sönecek.

```
int i = 0;
void setup()
{
  for(i = 2;i<=9;i++){
    pinMode(i,OUTPUT);
  }
}
void loop()
{
  for(i = 2;i<=9;i++)
  {
    digitalWrite(i,HIGH);
    delay(500);
  }
  for(i = 9;i>=2;i--)
  {
    digitalWrite(i,LOW);
    delay(500);
  }
}
```

Şekil 6.19. Yürüyen Işık Kodları



# BÖLÜM 6

## Uygulama 5

Uygulama 2'deki devrede ledler dıştan içe doğru sırasıyla yanacak ve sonra tekrar içten dışa doğru sönecek.

```
int i = 0;
void setup()
{
  for(i = 2;i<=9;i++){
    pinMode(i,OUTPUT);
  }
}
void loop()
{
  for(i = 0;i<4;i++)
  {
    digitalWrite(2+i,HIGH);
    digitalWrite(9-i,HIGH);
    delay(500);
  }
  for(i = 3;i>=0;i--)
  {
    digitalWrite(2+i,LOW);
    digitalWrite(9-i,LOW);
    delay(500);
  }
}
```

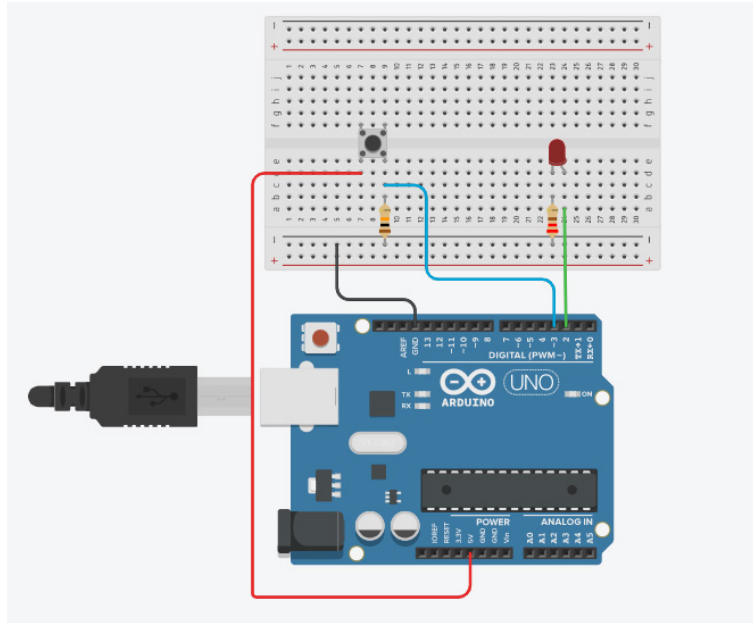
Şekil 6.20. Uygulama 5 Kod Satırları

### 6.5.2. Dijital Giriş Buton Uygulamalar

Buton kullanarak Arduino devrelerin kontrol edilmesi.

#### Uygulama 1

Bir butona bastığımızda Led yanacak, butonu bıraktığımızda Led sönecek. Devre şeması ve programı aşağıdaki gibidir.



Şekil 6.21. Buton Kontrollü Led Yakıp Söndürme Devresi

Tablo 6.3. Buton kontrollü LED kontrol devresinde kullanılan malzemeler

Ad	Miktar	Bileşen
U1	1	Arduino Uno R3
D1	1	Kırmızı LED
R1	1	330 $\Omega$ Rezistör
S1	1	Basma düğmesi
R2	1	10 k $\Omega$ Rezistör

# BÖLÜM 6

Ledin eksi (-) ucu direnç üzerinden Arduino kartın GND pinine, ledin artı (+) ucu 2 numaralı pine, buton çıkışı 3 numaralı pine, 5 volt ve GND pinleri de butona bağlanır. Bu sayede butona basılmadığında 3 numaralı pine GND gelmesi sağlanır.

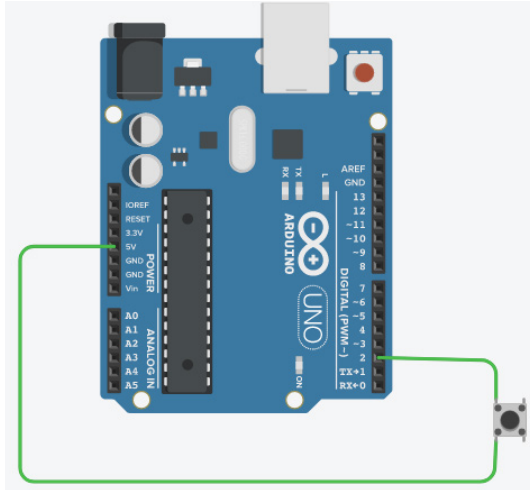
```
void setup()
{
  pinMode(2, OUTPUT);
  pinMode(3, INPUT);
}

void loop()
{
  int oku = digitalRead(3);
  digitalWrite(2, oku);
}
```

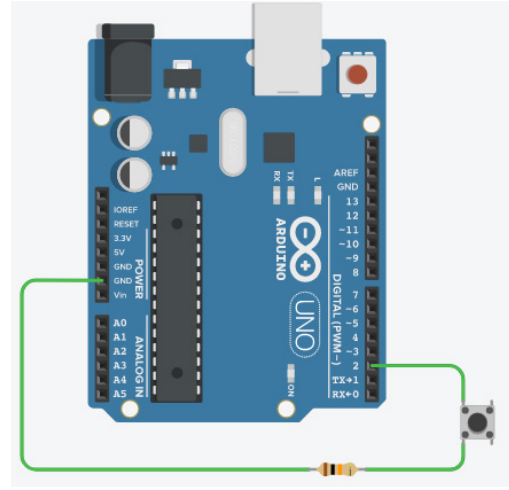
Şekil 6.22. Buton Kontrollü Led Yakıp Söndürme Kod Satırları

## Kullanılan komutlar;

**pinMode(pinNo, INPUT):** Pin numarası verilen bağlantı noktasını giriş olarak ayarlar. Bir bağlantı noktasını giriş olarak kullanmanın 2 yöntemi vardır. Birincisi INPUT, diğeri INPUT\_PULLUP seçeneğidir. INPUT seçildiğinde pin girişi 0 Volt'dur dışarıdan 5 Volt uygulayarak bu girişi aktif ederiz. INPUT\_PULLUP seçilirse pin girişi 5 Volt'dur, dışarıdan 0 Volt uygulayarak bu girişi pasif ederiz. Bu bağlantıda kısa devreyi engellemek için 10 kilo ohm luk bir direnç bağlanmalıdır. Bu iki yapının buton bağlantı şeması aşağıdaki gibidir.



Şekil 6.23. INPUT Bağlantı Şeması



Şekil 6.24. INPUT\_PULLUP Bağlantı Şeması