

**ARDUİNO NEDİR?**

Arduino, mikrodenetleyici ve yazılım geliştirme ortamından oluşan fiziksel programlama platformudur.

**ARDUİNO İLE NE YAPILIR? NE YAPILMAZ?**

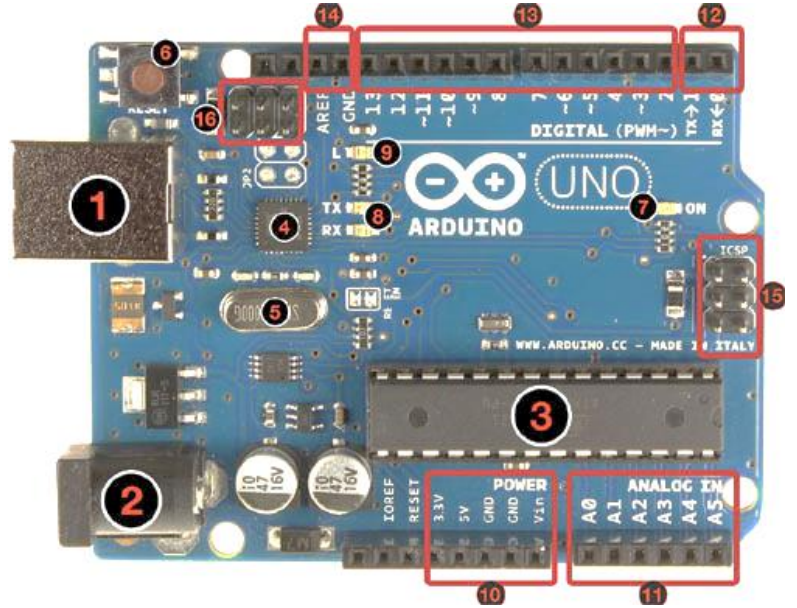
- ▶ Kolay bir şekilde çevresiyle etkileşime girebilen sistemler tasarlayabilirsiniz
- ▶ Arduino kütüphaneleri ile mikrodenetleyicileri kolaylıkla programlayabilirsiniz
- ▶ Analog ve dijital girişleri sayesinde analog ve dijital verileri işleyebilirsiniz
- ▶ Sensörlerden gelen verileri kullanabilirsiniz
- ▶ Dış dünyaya çıktılar (ses, ışık, hareket vs...) üretebilirsiniz
- ▶ Sıfır programlama bilgisiyle Arduino ile projeler yapmanız zordur. Ama aynı zamanda Arduino programlama öğrenmek için de güzel bir platformdur
- ▶ Gerçek zamanlı sinyal işleme, kamera görüntüsü aktarma gibi ağır işleri yapamazsınız (Arduino Due ile bu kısmen mümkün hale geldi)
- ▶ Üzerinde Android, Windows CE, Linux gibi işletim sistemleri çalıştıramazsınız. Bu tür çalışmalar yapmak için Rasperry Pi, Beagle Bone vs., gibi kartlara bakabilirsiniz.

**ARDUİNO UNO (I/O) GİRİŞLER/ÇI KİŞLAR**

Arduino üzerinde;

- ▶ Bilgisayar bağlantısını yapabilmemiz için bir USB girişi
- ▶ USB bağlı değilken güç sağlaması için bir güç girişi
- ▶ 5Volt ve 3.3Volt (+) güç çıkışı
- ▶ Topraklama (-) çıkışları. (Üzerinde GND yazan 3 çıkış)
- ▶ 5 adet analog pin
- ▶ 14 adet dijital pin mevcuttur.

1 : USB jaki	28
2 : Power jaki (7-12 V DC)	29
3 : Mikrodenetleyici ATmega328	30
4 : Haberleşme çipi	31
5 : 16 MHz kristal	32
6 : Reset butonu	33
7 : Power ledi	34
8 : TX / NX ledleri	
9 : Led	
10 : Power pinleri	
11 : Analog girişler	
12 : TX / RX pinleri	
13 : Dijital giriş / çıkış pinleri (yanında ~ işareti olan pinler PWM çıkışı olarak kullanılabilir.)	
14 : Ground ve AREF pinleri	
15 : ATmega328 için ICSP	
16 : USB aravüzü için ICSP	



Arduino kartı kenarlarında giriş/çıkış pinleri kullanıma uygun olarak hazırlanmıştır. Kod geliştirme yazılımı üzerinden yapılan yükleme işlemi sonrası doğrudan kodlarınız fiziksel ortamla etkileşimli çalışmaya başlar. Arduino, USB kablo, bilgisayar ve kod geliştirme yazılımı kullanarak projeler oluşturabilirsiniz

## Güç

Arduino Uno bir USB kablosu ile bilgisayar bağlanarak çalıştırılabilir ya da harici bir güç kaynağından beslenebilir. Harici güç kaynağı bir AC-DC adaptör ya da bir pil / batarya olabilir. Adaptörün 2.1 mm jaklı ucunun merkezi pozitif olmalıdır ve Arduino Uno 'nun power girişine takılmalıdır. Pil veya bataryanın uçları ise power konektörünün GND ve Vin pinlerine bağlanmalıdır.

- **VIN** : Arduino Uno kartına harici bir güç kaynağı bağlandığında kullanılan voltaj girişidir.
- **5V** : Bu pin Arduino kartındaki regülatörden 5 V çıkış sağlar. Kart DC power jakından (2 numaralı kısım) 7-12 V adaptör ile, USB jakından (1 numaralı kısım) 5 V ile ya da VIN pininden 7-12 V ile beslenebilir. 5V ve 3.3V pininden voltaj beslemesi regülatörü bertaraf eder ve karta zarar verir.
- **3.3V** : Arduino kart üzerindeki regülatörden sağlanan 3,3V çıkışıdır. Maksimum 50 mA dir.
- **GND** : Toprak pinidir.
- **IOREF** : Arduino kartlar üzerindeki bu pin, mikrodenetleyicinin çalıştığı voltaj referansını sağlar. Uygun yapılandırılmış bir shield IOREF pin voltajını okuyabilir ve uygun güç kaynaklarını seçebilir ya da 3.3 V ve 5 V ile çalışmak için çıkışlarında gerilim dönüştürücülerini etkinleştirebilir.

## Giriş ve Çıkışlar

Arduino Uno 'da bulunan 14 tane dijital giriş / çıkış pininin tamamı, pinMode(), digitalWrite() ve digitalRead() fonksiyonları ile giriş ya da çıkış olarak kullanılabilir. Bu pinler 5 V ile çalışır. Her pin maksimum 40 mA çekebilir ya da sağlayabilir ve 20-50 KOhm dahili pull - up dirençleri vardır. Ayrıca bazı pinlerin özel fonksiyonları vardır:

- **Serial 0 (RX) ve 1 (TX)** : Bu pinler TTL seri data almak (receive - RX) ve yaymak (transmit - TX) içindir.
- **Harici kesmeler (2 ve 3)** : Bu pinler bir kesmeyi tetiklemek için kullanılabilir.
- **PWM: 3, 5, 6, 9, 10, ve 11** : Bu pinler analogWrite () fonksiyonu ile 8-bit PWM sinyali sağlar.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)** : Bu pinler SPI kütüphanesi ile SPI haberleşmeyi sağlar.
- **LED 13** : Dijital pin 13 e bağlı bir leddir. Pinin değeri High olduğunda yanar, Low olduğunda söner.

Arduino Uno 'nun A0 dan A5 e kadar etiketlenmiş 6 adet analog girişi bulunur, her biri 10 bitlik çözünürlük destekler. Varsayılan ayarlarda topraktan 5 V a kadar ölçerler. Ancak, AREF pini ve analogReference() fonksiyonu kullanılarak üst limit ayarlanabilir.

- **TWI** : A4 ya da SDA pini ve A5 ya da SCL pini Wire kütüphanesini kullanarak TWI haberleşmesini destekler.
- **AREF** : Analog girişler için referans voltajdır. analogReference() fonksiyonu ile kullanılır.
- **RESET** : Mikrodenetleyiciyi resetlemek içindir. Genellikle shield üzerine reset butonu eklemek için kullanılır.

## SORULAR ( SINAVLAR İÇİN KAYNAK )

1. Arduino üzerinde kaç adet analog pin vardır.?
2. Arduino üzerinde kaç adet dijital pin vardır.?
3. Arduino üzerinde kaç çeşit çıkış gerilim desteği vardır ?
4. Arduino basit tanımını yapınız.

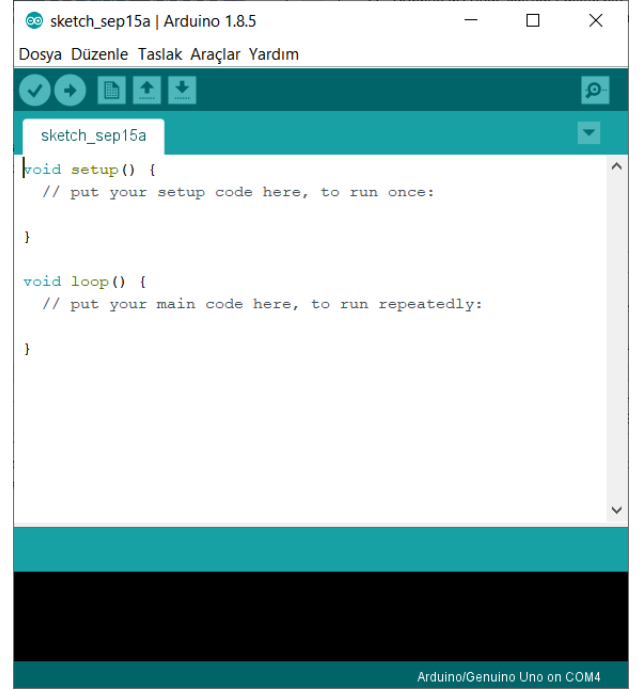
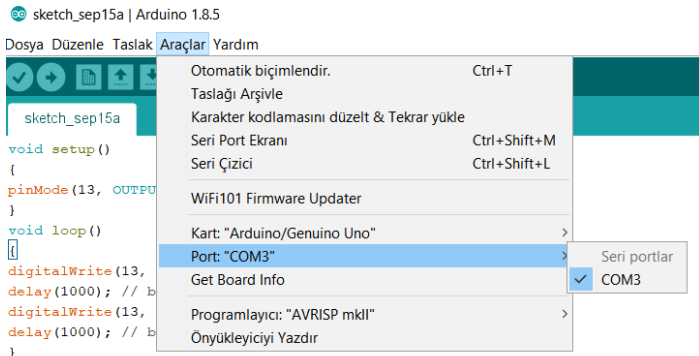
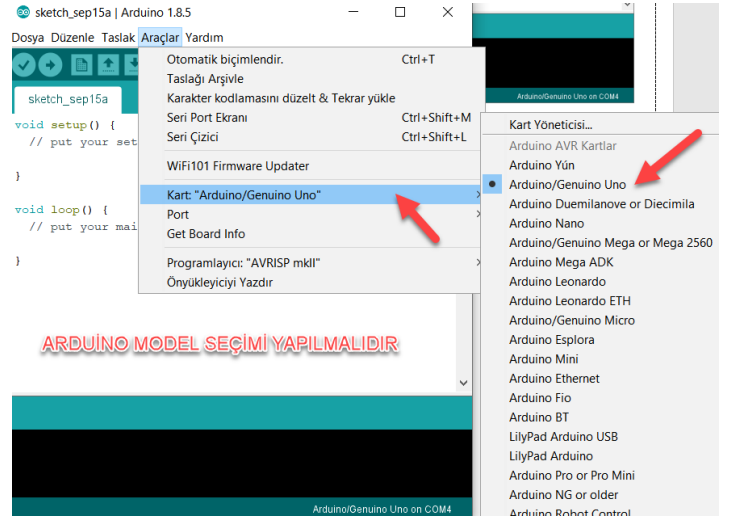
İŞE BAŞLAMA ve BİTİRME			PUANLAMA					KONTROL	
TARİH	SAAT	SÜRE	Teknoloji	İşlem Bas	Yazı/Çiz	İş Al.	Süre	TOPLAM	DERS ÖĞRETMENİ
...../...../20..	..... : .....	..... DERS SAATİ	%25	%25	%25	%15	%10	%100	
ÖĞRENCİNİN ADI SOYADI :		SINIF NO							

**Arduino Programlama**

Arduino yazılım geliştirme ortamında kod yazmak için '**sketch**' adı verilen proje dosyası oluşturulur. Arduino, en az iki fonksiyondan oluşan programı yürütür.

```
void setup(){
  /*
   * Burası sadece bir kere çalışır
   * Genel ayarlar buradan yapılır
   */
}

void loop(){
  /*
   * Bu fonksiyon sonsuza kadar çalışır
   * Arduino'nun yapması gereken işlemler buraya yazılır
   */
}
```

**GENEL AYAR KONTROLLERİ****PORT SEÇİMİ YAPILMALIDIR****ARDUINO MODEL SEÇİMİ YAPILMALIDIR****SORULAR ( SINAVLAR İÇİN KAYNAK )**

1. Arduino üzerine yazılım yüklemek için hangi program kullanılmalıdır ?
2. Arduino üzerine program yüklemeyen önce hangi ayar kontrolleri yapılmalıdır?
3. Arduino yazılım penceresini ilk olarak açtığınızda ekranda gözükken void setup () ve void loop () kavramlarını açıklayınız.

**İŞE BAŞLAMA ve BİTİRME**

İŞE BAŞLAMA ve BİTİRME			PUANLAMA						KONTROL
TARİH	SAAT	SÜRE	Teknoloji	İşlem Bas	Yazı/Çiz	İş Al.	Süre	TOPLAM	DERS ÖĞRETMENİ
...../...../20..	..... : .....	..... DERS SAATİ	%25	%25	%25	%15	%10	<b>%100</b>	
ÖĞRENCİNİN ADI SOYADI :		SINIF NO							

## AÇIKLAMALAR

## /\* ... \*/ Blok açıklama

Bölme-Yıldız ve Yıldız-Bölme işaretleri arasında yer alan ifadeler açıklama olarak kabul edilir.

Derleyici bu blok arasında yer alan komutları kontrol etmez. Bu blok arasına programcı tarafından yazılım geliştirmede görülmesi istenen, fakat programın işletilmesi esnasında geçersiz olan ifadeler yazılır.

```
/* açıklama metin grubu  
Bu alana yazılır.  
*/
```

## // ... Satır Açıklama

Bölme-Bölme işaretlerinden itibaren satırın sonuna kadar yer alan ifadeler açıklama olarak kabul edilir. Derleyici bu işaret sonrası bulunan ifadeyi satırın sonuna kadar kontrol etmez. Satırın başında olabileceği gibi satır içerisinde bir komuttan sonra da yer alabilir. Yine işaretin bulunduğu yerden satırın sonuna kadar olan ifadeleri açıklama olarak kabul eder.

```
// satır açıklama metni  
int led = 13; // led 13 ayarlanmıştır.
```

## Değer Atama

İfade ile belirtilen değer hesaplandıktan sonra, elde edilen sonucun değişkenin yeni değeri olacağı anlamına gelir. İfadeden sonraki noktalı virgüle dikkat edin. Bu noktalı virgül, deyimi bitirir. Bir satır içerisinde birden fazla komut bulunabilir.

Basit bir atama deyimi şu şekildedir:  
değişken = ifade;  
y = 5; // y değişkenine 5 değerini yükler.  
x = 3; z = 7;

**PinMode:** Arduino kartı üzerinde bulunan pinleri çıkış veya giriş olarak tanımlayabilirsiniz. Giriş olarak

tanımlanmış pinler düğme gibi elemanlar için kullanılırken, çıkış olarak tanımlanmış pinler ise LED gibi

elemanlar için kullanılır. Programın en başında (Setup içerisinde), kullanılacak pinler ayarlanmalıdır. Bunun

için aşağıdaki kod kullanılır: Böylece **13. Pin çıkış olarak** tanımlanır. Giriş olarak tanımlanması için **OUTPUT** yerine **INPUT** yazılması gerekir.

```
pinMode(13,OUTPUT);
```

**Digitalwrite:** Çıkış olarak Tanımlanmış pinlerden enerji vermek için veya enerji vermeyi durdurmak için

kullanılır. Böylece, daha önce çıkış olarak tanımlanmış 13 numaralı pinden enerji çıkışı sağlanır. Enerji çıkışı durdurulmak istenildiğinde HIGH yerine LOW yazılması yeterlidir.

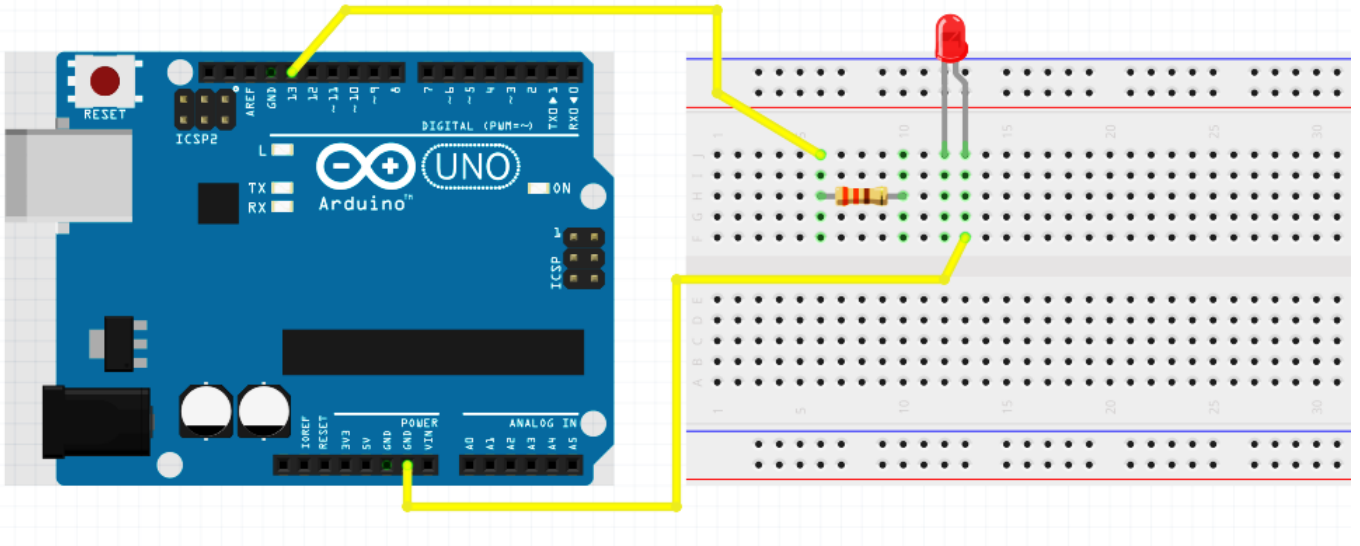
```
digitalwrite(13,HIGH);
```

**Delay:** İki kod arasında bir süre beklenmesi istenirse, delay fonksiyonu kullanılır. Örneğin yanıp sönen lamba

uygulaması yapılmak istenirse, lamba yakıp söndürme kodları arasında bekleme amacıyla delay fonksiyonu kullanılır. Delay fonksiyonunun içerisinde yazılan değer milisaniye düzeyindedir. Bir saniyelik bir bekleme için fonksiyona 1000 değeri verilmelidir.

```
delay(500);
```

Örneğin yarım saniyelik bir bekleme kodu yazalım;



#### UYGULAMA KODU 1:

```
void setup()
{
  pinMode(13, OUTPUT); // D13 pinini çıkış yap
}

void loop()
{
  digitalWrite(13, HIGH); // ledi yak
  delay(1000);           // bir saniye bekle
  digitalWrite(13, LOW); // ledi söndür
  delay(1000);           // bir saniye bekle
}
```

#### UYGULAMA KODU 2:

```
const int LED = 13;

void setup()
{
  pinMode(LED, OUTPUT); // D13 pinini çıkış yap
}

void loop()
{
  digitalWrite(LED, HIGH); // ledi yak
  delay(1000);           // bir saniye bekle
  digitalWrite(LED, LOW); // ledi söndür
  delay(1000);           // bir saniye bekle
}
```

#### SORULAR:

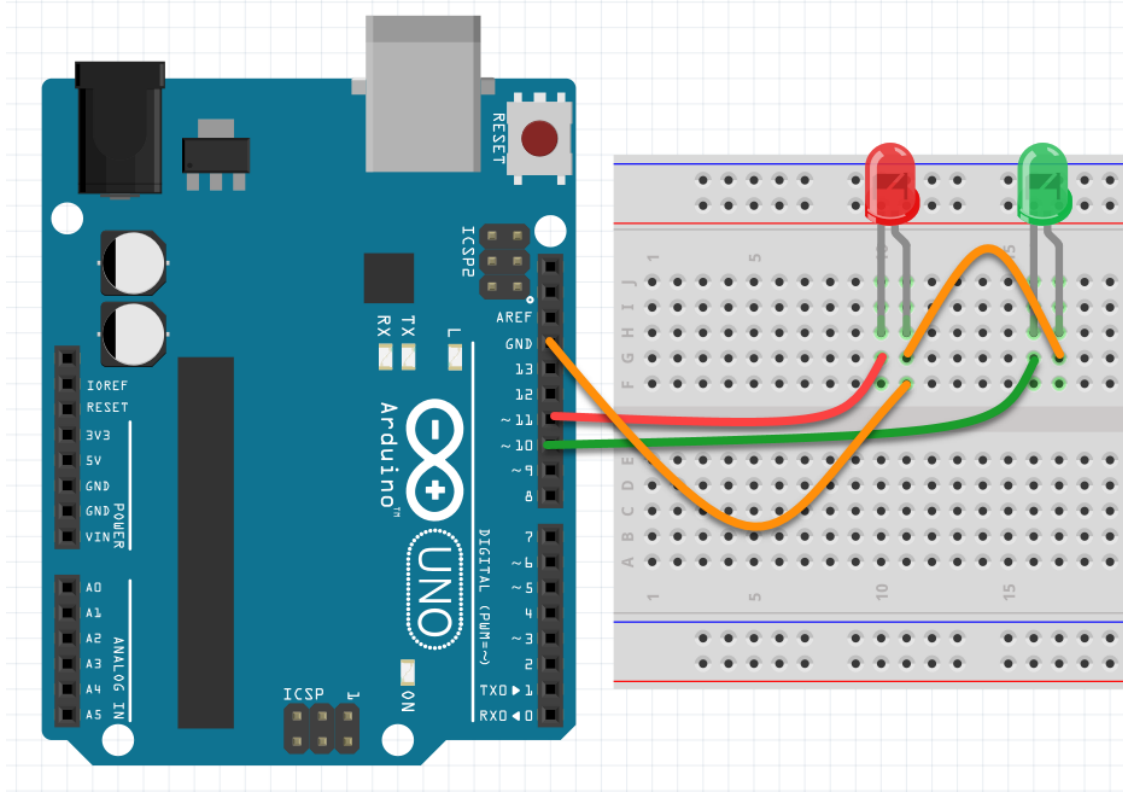
- Ledin 850 ms yanık, 200ms sönmük kalmasını sağlayacak programı yazınız.
- Ledin arka arkaya 200ms yanık, 200ms sönmük, 200ms yanık, 200ms sönmük 1sn yanık, 1sn sönmük kalmasını sağlayacak programı yazınız.
- Pinmode(8,output) komutunu açıklayınız.
- Delay(900), komutunu açıklayınız.
- A=5; C=12 ; komutlarının anlamını açıklayınız.
- Digitalwrite(9,low); komutunu açıklayınız.

İŞE BAŞLAMA ve BİTİRME			PUANLAMA						KONTROL
TARİH	SAAT	SÜRE	Teknoloji	İşlem Bas	Yazı/Çiz	İş Al.	Süre	TOPLAM	DERS ÖĞRETMENİ
...../...../20..	..... : .....	..... DERS SAATİ	%25	%25	%25	%15	%10	%100	
ÖĞRENCİNİN ADI SOYADI :		SINIF NO							

**Deneğin Amacı** : Arduino uno kullanarak temel elektronik uygulamalardan olan flaşör yani flip flop uygulamasını arduino kullanarak kodlarla gerçekleştirmek.

**Deneğin Malzemeleri** : Arduino UNO , 2 adet 330R direnç , 2 Adet Led diyot , Bread board , Gerekli miktar Jumper

**Deneğin Bağlantı Şeması** :



**Devre Kodları** :

```
int KirmiziLed = 11;
```

```
int SariLed = 10;
```

```
void setup() {
```

```
pinMode(KirmiziLed, OUTPUT); // Hangi dijital pinleri output olarak seçeceğimizi belirtiyoruz.
```

```
pinMode(SariLed, OUTPUT); // Hangi dijital pinleri output olarak seçeceğimizi belirtiyoruz.
```

```
}
```

```
void loop() {
```

```
digitalWrite(KirmiziLed, HIGH); // LED Yak (HIGH seçeneği ile KirmiziLed(pin 13) 5 Volt yapmış oluyoruz )
```

```
digitalWrite(SariLed, LOW); // LED Sondur (LOW seçeneği ile SariLed(pin 9) 0 Volt yapmış oluyoruz )
```

```
delay(1000); // 1000 mikrosaniye bekle
```

```
digitalWrite(KirmiziLed, LOW); // LED Yak (HIGH seçeneği ile KirmiziLed(pin 13) 5 Volt yapmış oluyoruz )
```

```
digitalWrite(SariLed, HIGH); // LED Sondur (LOW seçeneği ile SariLed(pin 9) 0 Volt yapmış oluyoruz )
```

```
delay(1000); // 1000 mikrosaniye bekle
```

```
}
```

**SORULAR:**

1- flip flop süresini 300ms olarak ayarlayarak kodu düzenleyiniz.

İŞE BAŞLAMA ve BİTİRME			PUANLAMA						KONTROL
TARİH	SAAT	SÜRE	Teknoloji	İşlem Bas	Yazı/Çiz	İş Al.	Süre	TOPLAM	DERS ÖĞRETMENİ
...../...../20..	..... : .....	..... DERS SAATİ	%25	%25	%25	%15	%10	%100	
ÖĞRENCİNİN ADI SOYADI :		SINIF NO							

**Deneyin Amacı** : Arduino uno kullanarak temel elektronik uygulamalardan olan trafik ışığı uygulamasını arduino kullanarak kodlarla gerçekleştirmek.

**Deney Malzemeleri** : Arduino UNO , 3 adet 330R direnç , 3 Adet Led diyot , Bread board , Gerekli miktar Jumper

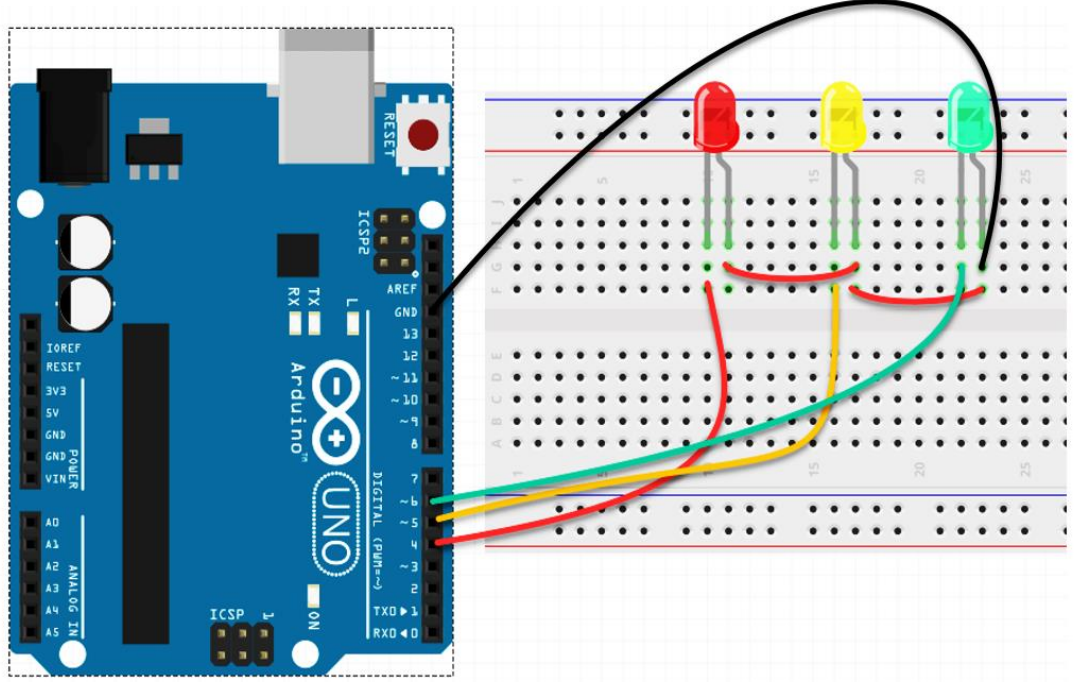
**Deney Bağlantı Şeması** :

**Devre Kodları :**

```
int kirmiziled = 4;
int sariled = 5;
int yesilled = 6;
```

```
void setup() {
  pinMode(kirmiziled, OUTPUT);
  pinMode(sariled, OUTPUT);
  pinMode(yesilled, OUTPUT);
}
```

```
void loop() {
  digitalWrite(kirmiziled, HIGH);
  delay(5000);
  digitalWrite(kirmiziled, LOW);
  digitalWrite(sariled, HIGH);
  delay(3000);
  digitalWrite(sariled, LOW);
  digitalWrite(yesilled, HIGH);
  delay(3000);
  digitalWrite(yesilled, LOW);
}
```



**SORULAR:**

1- Arduino çıkışlarının 4, 5,6,7 nolu çıkışları kullanılacaktır. Dört adet lede sırasıyla Led1 , Led2 , Led3 ve Led4 isim atamalarını yapınız. Sonra Led1 ve Led2 1 saniye yanık , sonra Led3 ve Led4 1 sn yanık olacak kod dizimini yazınız.

**İŞE BAŞLAMA ve BİTİRME**

**TARİH** : ..../.../20..

**SAAT** : ..... : .....

**ÖĞRENCİNİN ADI SOYADI :**

**SÜRE** : ..... DERS SAATİ

**SINIF NO**

**PUANLAMA**

Teknoloji

İşlem Bas

Yazı/Çiz

İş Al.

Süre

TOPLAM

%25

%25

%25

%15

%10

%100

**KONTROL**

**DERS ÖĞRETMENİ**

## Arduino Programlamada Veri Tipleri

Değişken, program içerisinde kullanılan nümerik değerlerin saklanmasında veya isimlendirilmesinde kullanılan bir yöntemdir.

```
int potdegeri = 0;           // potdegeri isminde ve değeri 0 olan değişken tanımlama
potdegeri = analogRead(2); // 2. analog kanaldan okunan veriyi potdegeri isimli değişkene yükle.
```

Örnekte yer alan 'int' ifadesi kullanılan değişkenin türünü gösterir. 'integer' kelimesinin kısaltmasıdır, Tam sayı türünde değişken tanımlar. 'potdegeri' değişkene verilmesi istenen isimdir.

- Rasgele isimler seçilebilir. İsmin ilk harfi rakam olmamalıdır.
- Anahtar kelimeler değişken ismi olarak kullanılamaz. (if, switch, for, true, high vb.)
- Değişken ismi tek kelimedenden oluşmalıdır. İki kelime kullanılacaksa arada '\_' kullanılmalıdır.
- 'Ç, ç, Ğ, ğ, İ, İ, Ö, ö, Ş, ş, Ü, ü' Türkçe karakterleri değişken isimlendirmede kullanılamaz.

Değişkenler lokal ve global olmak üzere iki yerde bulunur. Lokal değişkenler fonksiyon içerisinde yer alır ve sadece o fonksiyon içerisinde kullanılabilir. Global değişkenler kod dosyası içerisindeki tüm fonksiyonlarda geçerlidir. Global değişkenler setup fonksiyonundan önce yazılır.

Kullanılan sayısal ifadelerde gereksiz veri boyutu tercih edilmemelidir. Bütün mikrodenetleyiciler sınırlı bir veri boyutu alanı vardır. Ayrıca gereksiz veri boyutu işlem yükünü artırır. Arduino programlamada kullanılan veri tipleri, boyutu ve veri aralıkları tabloda verildiği gibidir

## TRUE / FALSE Tanımlamaları

- Lojik seviyeleri tanımlayan Boolean sabitleridir.
- TRUE ifadesi sıfırdan farklı herhangi bir değer için kullanılabilir.
- 3, 5, -7, -500 gibi ifadeler TRUE olarak tanımlanır.
- FALSE ifadesi ise yalnızca sıfır '0' olarak tanımlıdır.

Veri Tipi	Boyutu	Veri Aralığı
boolean	1 bit	TRUE veya FALSE
byte	8 bit	0 ... 255
unsigned char	8 bit	0 ... 255
char	8 bit	-128 ... 127
signed char	8 bit	-128 ... 127
unsigned int	16 bit	0 ... 65535
int	16 bit	-32768 ... 32767
unsigned long	32 bit	0 ... 4294967296
long	32 bit	-2147483648 ... 2147483647
float	32 bit	-3.4028235E+38 ... 3.4028235E+38

## HIGH / LOW Tanımlamaları

- Arduino pinlerindeki lojik seviyeleri tanımlamada kullanılır.
- HIGH, Lojik 1 anlamına gelir, pinde 5V gerilimi ifade eder.
- LOW, Lojik 0 anlamına gelir, pinde 0V gerilimi ifade eder.

## SORULAR

1. 0 ile 255 arasındaki tam sayılar hangi veri tipi aralığını ifade eder ?
2. İnt veri aralığını yazınız.
3. Arduino da bir digitalWrite komutu ile verilen "high" komutu neyi ifade etmektedir?
4. Arduino kodlama işlemlerinde true ve false kavramını açıklayınız.
5. "23led1" değişkeni arduino değişken tanımında hata vermektedir , nedeni hakkında bilgi veriniz.
6. "Led 1" değişkeni arduino değişken tanımında hata vermektedir , nedeni hakkında bilgi veriniz.

## İŞE BAŞLAMA ve BİTİRME

İŞE BAŞLAMA ve BİTİRME			PUANLAMA						KONTROL
TARİH	SAAT	SÜRE	Teknoloji	İşlem Bas	Yazı/Çiz	İş Al.	Süre	TOPLAM	DERS ÖĞRETMENİ
...../...../20..	..... : .....	..... DERS SAATİ	%25	%25	%25	%15	%10	<b>%100</b>	
ÖĞRENCİNİN ADI SOYADI :		SINIF NO							



Aritmetik işlemler, toplama, çıkarma, çarpma ve bölme ifadelerini içerir. Değişken-değişken, değişken-sayı veya sayı-sayı biçiminde kullanılabilir.

```
y = x + 3;
x = t - 5;
k = 2 * m;
r = s / 5;
y = x + z;
z = 3 + 5;
```

Tam sayılarla bölme işlem yapıldığında sonuç bölümün tam kısmıdır. 9 / 4 işleminin sonucu 2,25 olmasına rağmen tam sayı sonucu 2 alınır. Yuvarlama yoktur. Kısaltılmış aritmetik işlemler program yazımında kolaylık sağlar.

```
x ++; // x = x + 1; ile aynı anlama gelir. x değerini 1 artırır. "Bu işlem çok kullanılan bir uygulamadır"
x --; // x = x - 1; ile aynı anlama gelir. x değerini 1 azaltır. "Bu işlem çok kullanılan bir uygulamadır"
x += y; // x = x + y; ile aynı anlama gelir. x ve y değerini topla, sonucu x'e yükle.
x -= y; // x = x - y; ile aynı anlama gelir. x'den y değerini çıkar, sonucu x'e yükle.
x *= y; // x = x * y; ile aynı anlama gelir. x ile y değerini çarp, sonucu x'e yükle.
x /= y; // x = x / y; ile aynı anlama gelir. x'i y değerine böl, sonucu x'e yükle.
```

Örneklerde y değişkeni yerine sayısal değer de kullanılabilir.

```
x *= 3; // x = x * 3; ile aynı anlama gelir. x ile 3 değerini çarp, sonucu x'e yükle.
```

### Karşılaştırma İfadeleri

Karşılaştırma ifadeleri iki değişkeni birbiri ile karşılaştırmak veya değişkeni sayı ile karşılaştırmak için kullanılır. Sonucu lojik olarak TRUE ya da FALSE olur. Bu sonuç çoğunlukla karar yapılarında kullanılır.

```
x == y // x değeri, y'ye eşit mi?
x != y // x değeri, y'ye eşit değil mi?
x < y // x değeri, y'den küçük mü?
x > y // x değeri, y'den büyük mü?
x <= y // x değeri, y'den küçük veya eşit mi?
x >= y // x değeri, y'den büyük veya eşit mi?
```

### SORULAR

x=2 ve y=3 için aşağıda verilen soruları cevaplandırınız

1. X++; "işlemi ile ortaya çıkan sonuç x=...?... "
2. X--; "işlemi ile ortaya çıkan sonuç x=...?... "
3. X += y; "işlemi ile ortaya çıkan sonuç x=...?... "
4. X \*= y; "işlemi ile ortaya çıkan sonuç x=...?... "
5. Y=x+3; "işlemi ile ortaya çıkan sonuç y=...?... "

### İŞE BAŞLAMA ve BİTİRME

İŞE BAŞLAMA ve BİTİRME			PUANLAMA						KONTROL
TARİH	SAAT	SÜRE	Teknoloji	İşlem Bas	Yazı/Çiz	İş Al.	Süre	TOPLAM	DERS ÖĞRETMENİ
...../...../20..	..... : .....	..... DERS SAATİ	%25	%25	%25	%15	%10	<b>%100</b>	
ÖĞRENCİNİN ADI SOYADI :		SINIF NO							

**For Döngüsü nedir ?**

For döngüsü, programlama dillerinde en sık kullanılan döngülerden biridir. Arduino programlamada da sıkça kullanılır ve uygulamaları daha pratik bir yapıda kodlamamızı sağlar. **For döngüsü uygulanması istenen işlemin kaç kez tekrar edileceğinin belli olduğu durumlarda kullanılır.** Daha net bir ifade ile for döngüsü sayesinde işlemin kaç kez tekrar edileceğini, başlangıç/bitiş değerlerini ve değerler arasındaki artış/azalış miktarını kontrol edebiliriz. Aşağıdaki kod parçacığındaki gibi özetleyebiliriz.

```
for (başlangıç değeri; döngünün devam etme koşulu; değerdeki değişim)
{
işlemler...
}
```

```
int ledler[] = {2,3,4,5,6,7}; //Değişkenleri belirttiğimiz bir dizi tanımlıyoruz.
//Dizilerde ilk eleman 0. ( sıfırıncı elemandır )
void setup()
{
for(int i=0; i<6; i++) { //For döngüsünün 6 kez tekrarlanmasını istiyoruz.
pinMode(ledler[i], OUTPUT); //i değişkenlerini "çıkış" olarak tanımlıyoruz.
}
}

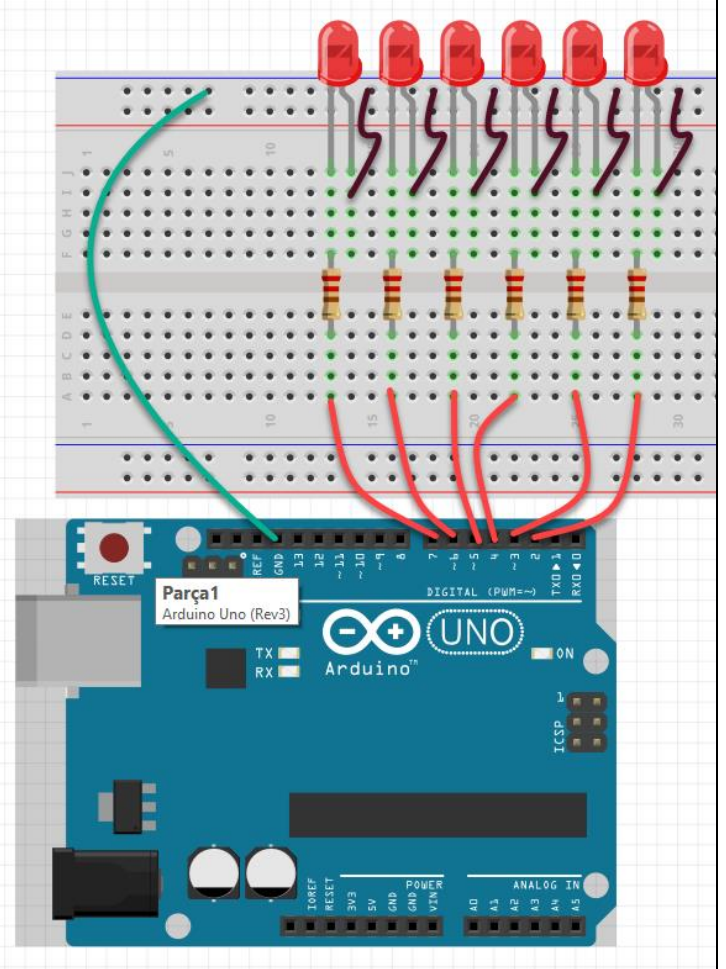
void loop()
{
for(int i=0; i<6; i++) {
digitalWrite(ledler[i], HIGH);
delay(100);
digitalWrite(ledler[i], LOW);
}
//İlk ledten beşinci lede kadar ledler sırasıyla yanıyor. Yani döngüde ilk led
yanar, söner ve ikinci led yanar. İkinci led söndüğünde ise üçüncü led yanar
ve döngü bu şekilde devam eder.

for(int j=5; j>-1; j--) {
digitalWrite(ledler[j], HIGH);
delay(100);
digitalWrite(ledler[j], LOW);
}

//Bu kez ters döngü yapıyoruz. Yani dizinin 5. elemanı olan 7 pinine bağlı led
yanar, söner ve bu kez bir önceki led yanar. İşlem bu şekilde devam eder.

}

//Bu iki döngü sırası birbirinin tersidir. Bu şekilde ledler sırasıyla ileri ve geri
yanıp sönerler.
```

**SORULAR**

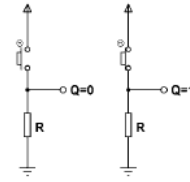
Yukarıda yaptığınız 6 ledli yürüyen ışık “ karaşimşek “ devresini 4 ledli düzeneğini kurunuz. Kod yapısını düzeltiniz.

İŞE BAŞLAMA ve BİTİRME			PUANLAMA						KONTROL
TARİH	SAAT	SÜRE	Teknoloji	İşlem Bas	Yazı/Çiz	İş Al.	Süre	TOPLAM	DERS ÖĞRETMENİ
...../...../20..	..... : .....	..... DERS SAATİ	%25	%25	%25	%15	%10	%100	
ÖĞRENCİNİN ADI SOYADI :		SINIF NO							

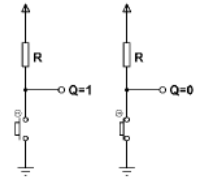
**İŞİN ADI****Buton Kontrol Uygulaması****MİKROKONTROL DEVRELERİ****TEMRİN**

AMAÇ: Arduino ile giriş-çıkış işlemlerini kavramak

GİRİŞ: Arduino pinlerinin giriş olarak mı yoksa çıkış olarak mı kullanılacağına ayarlanmasında pinMode() fonksiyonu kullanılır. Arduino programlamada bir pinin giriş olarak kullanılabilmesi için pinMode komutu ile INPUT veya INPUT\_PULLUP modu kullanılmalıdır. INPUT\_PULLUP modunun diğerinden farkı mikrodenetleyici içerisinde yer alan dahili PULL-UP direcinin kullanıma almasıdır. Eğer giriş elemanı pull-up durumunda kullanılacaksa, harici bir direnç bağlantısına gerek kalmaz.



**Pull-down Bağlantı**  
Buton basılı=1  
Buton basılı değil=0



**Pull-up Bağlantı**  
Buton basılı=0  
Buton basılı değil=1

```
pinMode (2, INPUT); // 2 nolu pin dijital giriş modunda
```

```
pinMode(2, INPUT_PULLUP); // 2 nolu pin Lojik 1'e çekmeli giriş modunda
```

Buton bağlantısında PULL-UP ve PULL-DOWN olmak üzere iki farklı yapı vardır.

### Değişkenlerin isimlendirilmesinde bazı kurallara dikkat edilmelidir Hatırlatma !

```
int durum = 0;
```

```
// durum isminde ve değeri 0 olan değişken tanımlama
```

```
durum = digitalRead(3);
```

```
// dijital 3 kanalından okunan değeri okunan değeri durum isimli değişkene yükle.
```

```
const int ledPin = 8;
```

```
// 8 nolu pin ledPin değişkeni
```

Değişken tanımlamada kullanılan **"const"** ifadesi, değişkenin değerinin program içerisinde değiştirilemeyeceğini gösterir. Bu ifadeye göre belirtilen değişken program hafızada saklanır.

### İF ve SWITCH CASE KULLANIMI

Arduino programlamada koşula dayalı yapılması istenen işlemler **'if'** koşul yapısı veya **'switch case'** koşul yapısı ile gerçekleştirilir. Çoktan seçmeli koşullarda **'switch case'** kullanımı yaygındır. **'if'** koşul yapısı tek yönlü, çift yönlü ve çok yönlü olmak üzere üç farklı kullanımı vardır. Koşula dayalı yapılması istenen işlem birden fazla ise bu komutlar tırnak parantez { } arasına alınarak gruplanmalıdır. Koşula dayalı tek işlem yapılacaksa tırnak parantez ihtiyacı yoktur.

✓ Tek yönlü koşul yapısı

```
if (koşul)
{
.....
}
```

✓ Çift yönlü koşul yapısı

```
if (koşul)
{
.....
}
else
{
.....
}
```

✓ Çok yönlü koşul yapısı

```
if (koşul1)
{
.....
}
else if (koşul2)
{
.....
}
else
{
.....
}
```

**Uygulamada** Arduino UNO kartının **3 nolu** pinine **pull-down** yapısında bağlı butona basıldığında, **8 nolu pinine** bağlı ledin yanmasını sağlanacaktır.

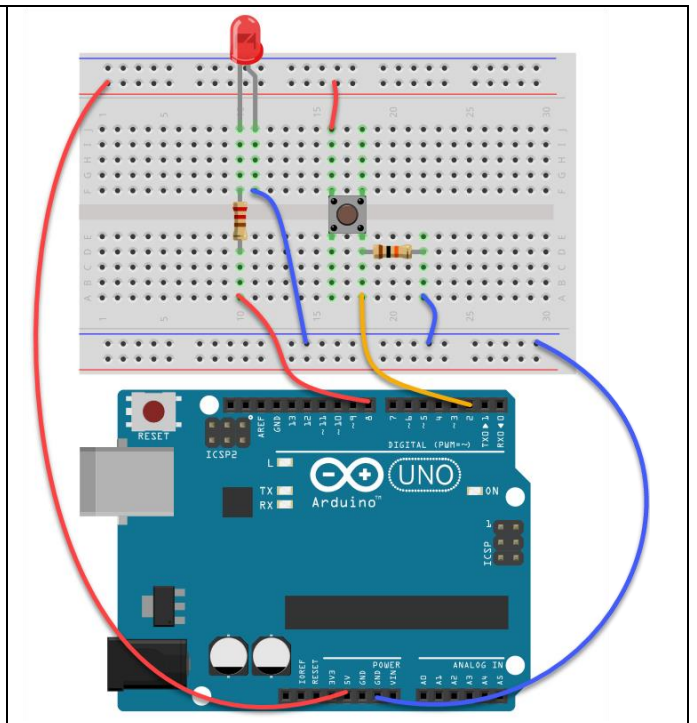
Bir pinden giriş değeri okuyabilmek için **"digitalRead"** komutu kullanılır. Bu komut sonucu elde edilen değer LOW (Lojik 0) veya HIGH (Lojik 1) olacaktır.

```
durum = digitalRead (3);
```

```
//Dijital 3 kanalından giriş değeri oku
```

```
const int buton = 3; // buton için dijital giriş numarası
const int led = 8; // led için dijital çıkış numarası
int butondurum = 0; // buton durumunu gösteren degisken
void setup()
{
pinMode(led, OUTPUT);
pinMode(buton, INPUT);
}
void loop()
{
butondurum = digitalRead(buton); // butonun bağlı olduğu girişi oku

if (butondurum == HIGH) // butondurum Lojik 1 ise
{
digitalWrite(led, HIGH); // ledi yak
}
else
{
digitalWrite(led, LOW); // ledi söndür
}
}
```



### SORULAR

Uygulamada verilen devre şemasına göre butona basılması durumunda ledin 3sn yanık kaldıktan sonra sönmelerini sağlayacak, kod düzenlemesini yapınız.

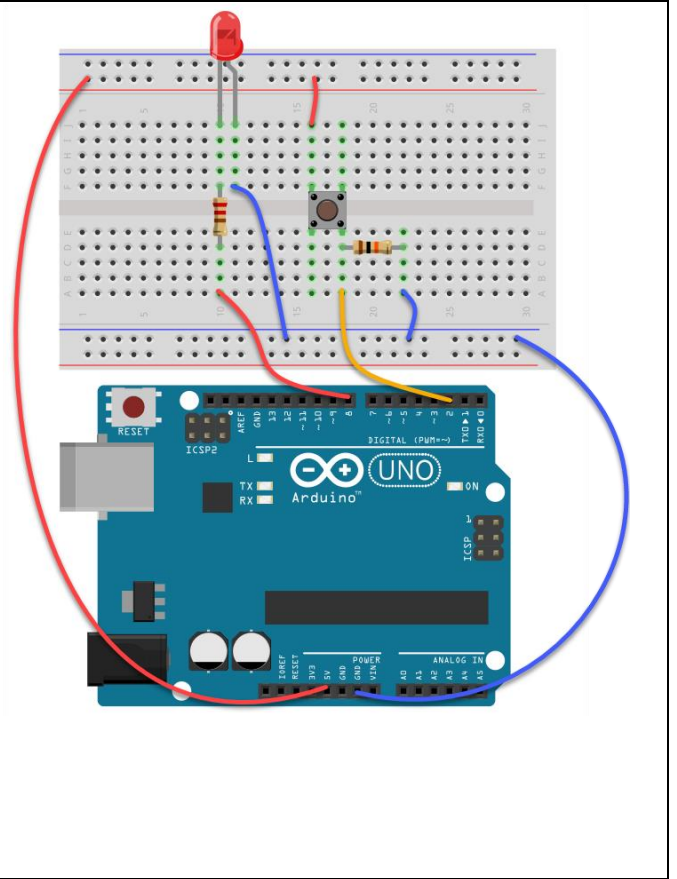
**Uygulamada** Arduino UNO kartının **3 nolu** pinine **pull-down** yapısında bağlı butona basıldığında, **8 nolu** pinine bağlı ledin sürekli yanması, bir daha basıldığında sönmesini sağlayacak arduino programını uygulayınız.

```
int led=8;
int buton=3;
int butonDurum;
int ledDurum=0;
int x=0;

void setup(){
  pinMode(led, OUTPUT);
  pinMode(buton, INPUT);
}

void loop(){
  butonDurum=digitalRead(buton);
  if(butonDurum==HIGH && x==0){
    x=1;
    if(ledDurum==0) ledDurum=1;
    else if(ledDurum==1) ledDurum=0;
  }
  else if(butonDurum==LOW && x==1){
    x=0;
  }

  if (ledDurum==1){
    digitalWrite(led, HIGH);
  }
  else{
    digitalWrite(led, LOW);
  }
}
```



## İŞE BAŞLAMA ve BİTİRME

TARİH ...../...../20..

SAAT ..... : .....

SÜRE

..... DERS SAATİ

ÖĞRENCİNİN  
ADI SOYADI :

SINIF  
NO

## PUANLAMA

Teknoloji	İşlem Bas	Yazı/Çiz	İş Al.	Süre	TOPLAM
%25	%25	%25	%15	%10	%100

## KONTROL

DERS  
ÖĞRETMENİ

**LDR Nedir?**

LDR (Light Dependent Resistor) yani Türkçe'de "Işığa Bağımlı Direnç" anlamına gelmektedir. LDR ışığa duyarlı bir devre elemanıdır. Ters orantılı bir çalışma prensibine sahiptir. Yani üzerine düşen ışık değeri arttıkça sahip olduğu direnç değeri azalırken, üzerine düşen ışık değeri arttıkça sahip olduğu direnç artar. Bu özelliği sebebiyle LDR devrede bir ON-OFF şeklinde anahtarlama görevinde bulunur. Günlük hayatta LDR sensörü alarmlarda, anahtarlama cihazlarında, sokak aydınlatmalarında kullanılabilirler. Devremiz de kullandığımız LDR sensörümüz bir Analog Girişli Sensördür.

**Analog Giriş Nedir?**

Arduino da ADC (Analog To Digital Converter) özelliği mevcuttur. Bu özellik sayesinde dış dünyadan gelen verileri sayısal değerlere dönüştürebiliriz. Arduino UNO kartı üzerinde A0-A5 olmak üzere 6 adet analog giriş bulunur. Analog giriş sayısı modele göre değişebilir ama işlevi aynıdır.

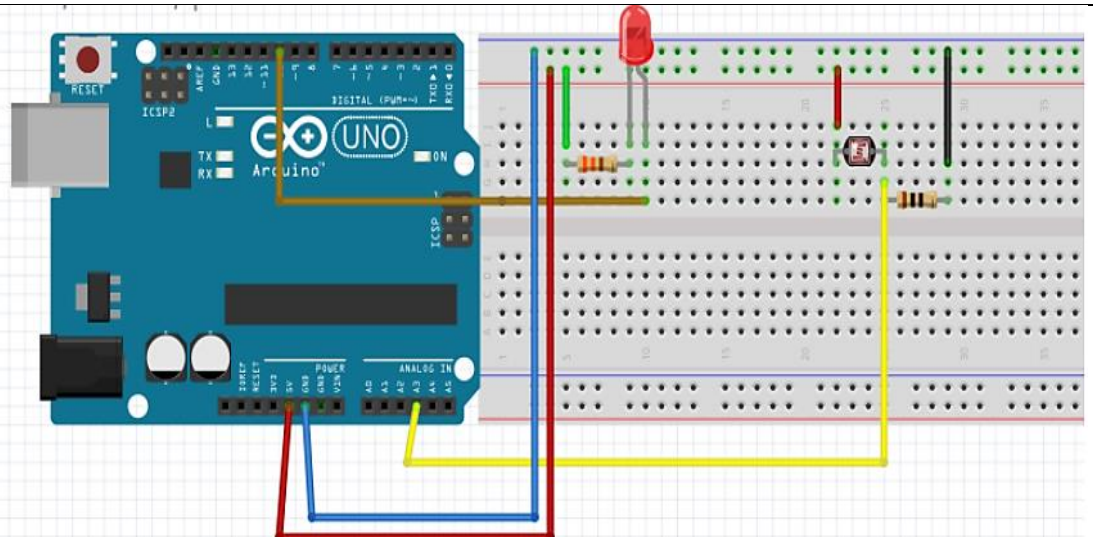
```
int ledPin = 10;
int LDRPin = A3;
void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  int isik_degeri = analogRead(LDRPin);
  Serial.println(isik_degeri);
  delay(500);
  if(isik_degeri<10)
  {
    digitalWrite(ledPin, HIGH);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }
}
```

```
int ledPin = 10; // İlk olarak LED'imizin ve LDR
sensörümüzün pinlerini belirliyoruz.
int LDRPin = A3;
pinMode(ledPin, OUTPUT); // LED'imizin bir OUTPUT
olduğunu belirtiyoruz.
Serial.begin(9600); // Seri iletişimi başlatıyoruz.
Loop Kısımına indiğimiz zaman ise;
int isik_degeri = analogRead(LDRPin); // LDR sensörümüzün
ölçmüş olduğu ışık değerini int tipinde bir değişkene
atıyoruz.
Serial.println(isik_degeri); // Gelen değeri ise yeni
değişkenimizde okumasını istiyoruz.
delay(100); // Saniyenin 1/10 kadarını beklemesini
söylüyoruz.
if(isik_degeri<200) // if döngüsü başlatıyoruz gelen ölçülen
değerin bizim belirlediğimiz değerin altında veya üstünde
olma durumunu kontrol ediyoruz.
{
  digitalWrite(ledPin, HIGH); // Eğer ki gelen ışık düzeyi 200
de fazla ise LED'imizi aktif ediyoruz.
}
else
{
  digitalWrite(ledPin, LOW); // Eğer ki gelen ışık düzeyi 200
den fazla ise LED'imizi pasif durumuna çekiyoruz.
}
```

**MALZEME LİSTESİ**

- Arduino Uno
- Breadbord
- LDR Işık Sensörü
- 1 Adet LED
- 330 Ω Direnç
- 100 Ω Direnç
- Jumper Kablolar



## SORULAR

1. Arduino ADC özelliđi hakkında kısaca bilgi veriniz.
2. Arduino UNO kartının 2 ve 3 nolu pinlerine pull-down yapısında buton bađlayınız. Ayrıca 13, 12, 11 nolu pinlere led bađlayınız. Bu bađlantılara göre;
  - A ve B butonuna basıldıysa 13 nolu pine bađlı ledin,
  - A veya B butonuna basıldıysa 12 nolu pine bađlı ledin,
  - A butonu - B basılı deđil veya A basılı deđil B basılı ise 11 nolu pine bađlı ledin yanmasını sađlayan programı yazınız.

İŐE BAŐLAMA ve BİTİRME			PUANLAMA					KONTROL	
TARİH	SAAT	SÜRE	Teknoloji	İŐlem Bas	Yazı/Çiz	İŐ Al.	Süre	TOPLAM	DERS ÖĐRETMENİ
...../...../20..	..... : .....	..... DERS SAATİ	%25	%25	%25	%15	%10	<b>%100</b>	
ÖĐRENCİNİN ADI SOYADI :		SINIF NO							