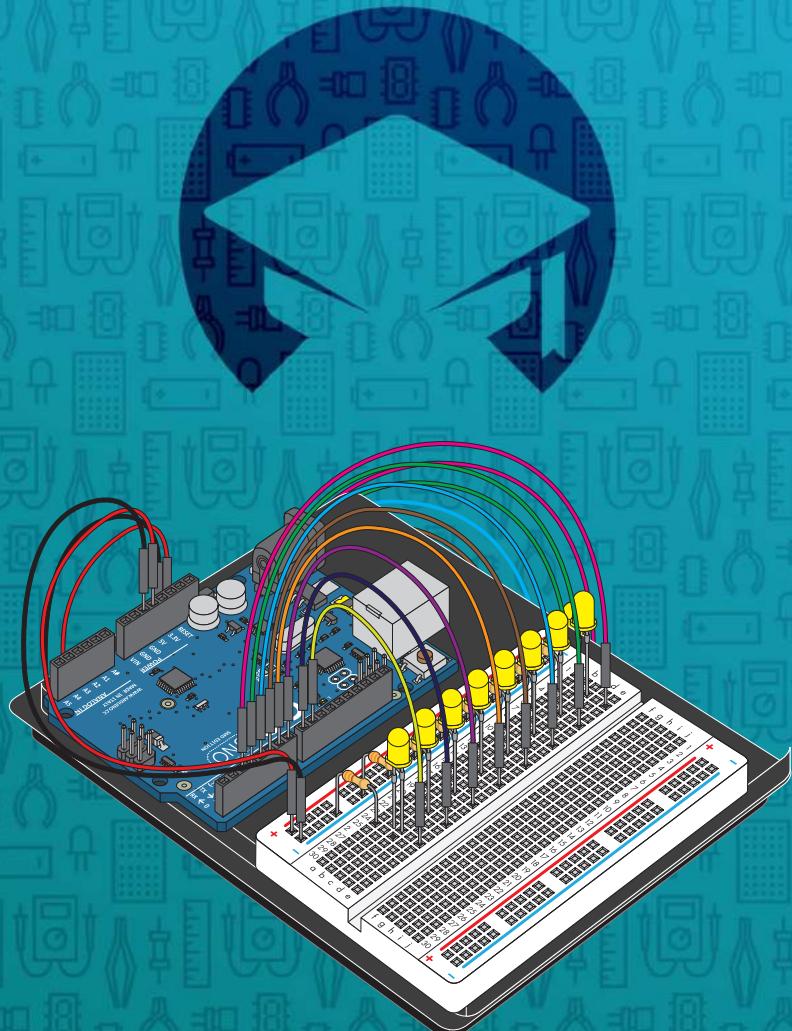


# AKADEMİKPORT

## Arduino Başlangıç Projeleri



## Kitapçık Hakkında

### AkademikPort "Arduino Başlangıç Projeleri" Eğitimine Hoşgeldiniz!

Bu kitapçık **SparkFun SIK Guide(retired)** ve bir çok yerli-yabancı kaynaktan yararlanılarak oluşturulmuş ve açık kaynaklı olarak ilk lisans şartı olan **Creative Commons CC BY SA 4.0** lisansı ile yayınlanmıştır. "**AkademikPort Arduino Başlangıç Projeleri**" Eğitimi **AkademikPort**'un açık kaynaklı elektronik kitap eğitimlerinin yedinci kitabı olarak yayınlanmıştır. Bu kitapçıkta sizler için hazırlanan 12 Arduino breadboard devresi sayesinde devre elemanları ve sensörlerle Arduino projeleri geliştirerek kısa süre içerisinde Arduino'yu kullanmayı ve programlamayı öğreneceksiniz.

Üretmekten korkmayan bir toplum olmamız dileğiyle...

Emre ARSLAN  
AkademikPort Kurumsal İlliskiler Koordinatörü

[www.akademikport.com](http://www.akademikport.com)

### Orjinal Kitap Editörleri

**SparkFun Education Department**

### Türkçe Kitap Hazırlayanlar

Emre Arslan

Hakan Çolakoğlu



Arduino Başlangıç Projeleri is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

(Arduino Başlangıç Projeleri elektronik kitabı yazar isimleri belirtilip, lisans şartlarına uyularak paylaşılmabilir fakat ticari bir faaliyette kullanılamaz.)

To view a copy of this license visit:

<http://creativecommons.org/licenses/by-sa/4.0/>

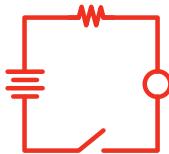
[www.akademikport.com](http://www.akademikport.com)



## 1. Bölüm

### Arduino'ya Başlamak

Arduino nedir?	1
Arduino IDE 'yi İndirmek	3
Sürücüyü Yüklemek	4
Identify your Arduino	7
"Elektrikport Arduino Kod" indirme	8



## 2.Bölüm

### Temel Elektronik ve Arduino Projeleri

Elektrik Devreleri	9
Devre Elemanları	11
Arduino Uno	13
Breadboard	15
Devre #1 - İlk Devreniz: LED Yakma	17
Devre #2 - Potansiyometre	24
Devre #3 - RGB LED	28
Devre #4 - Çoklu LED	32
Devre #5 - Buton Kullanımı	36
Devre #6 - Foto Direnç	40
Devre #7 - Sıcaklık Sensorü	44
Devre #8 - Servo Motor	48
Devre #9 - Buzzer	52
Devre #10 - Motor Döndürme	56
Devre #11 - Röle	60
Devre #12 - Shift Register	64
Notlar	68

# Arduino nedir?



## Arduino Çağı Başlıyor

Arduino kolay bir şekilde çevresiyle etkileşime girebilen sistemler tasarlayabileceğiniz, hem acemi hem de ileri düzeydeki elektronik meraklılarına hitap eden, kolayca programlanabilen ve üzerine elektronik devre kurulabilen açık kaynaklı bir geliştirme platformudur.

[arduino.cc](http://arduino.cc)

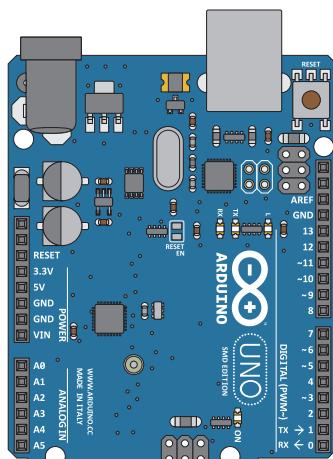
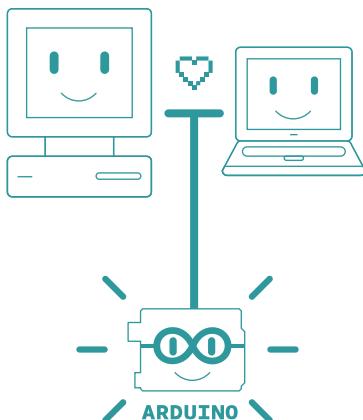
## Fiziksel Dünya için Bir Bilgisayar

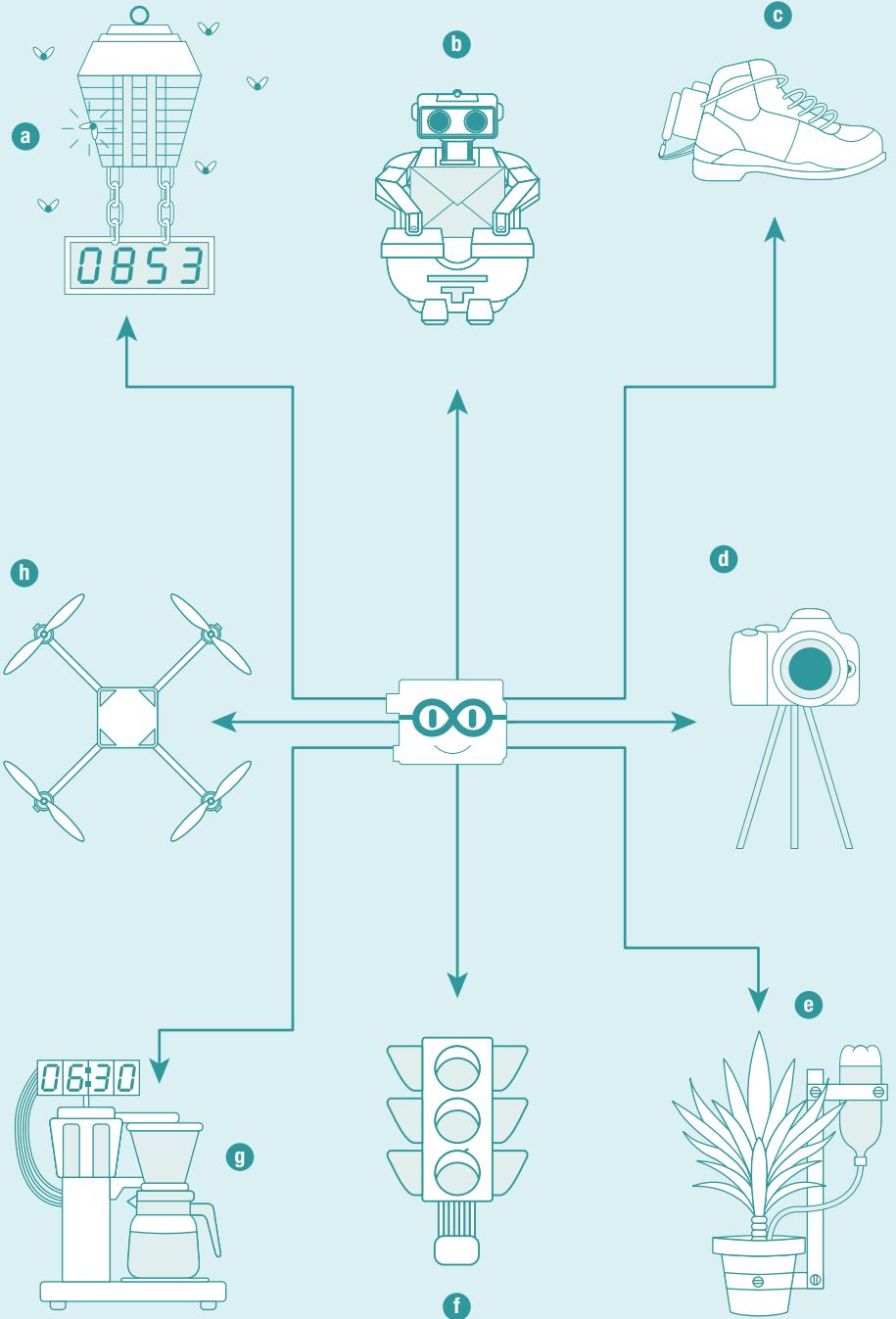
Bu elinizde tuttuğunuz dost canlısı mavi kartı masaüstü bilgisayarlarınızın ve laptoplarınızın çocuğu gibi taşınabilir küçük bir bilgisayar olarak düşünebilirsiniz. Arduino kütüphaneleri ile mikrodenetleyicileri kolaylıkla programlayabilirsiniz. Analog ve dijital girişleri sayesinde analog ve dijital verileri işleyebilir ve sensörlerden gelen verileri kullanarak dış dünyaya çıktılar (ses, ışık, hareket vs...) üretebilirsiniz.

Arduino kartları bir Atmel AVR mikrodenetleyici programlama ve diğer devrelere bağlantı için gerekli yan elemanlarından oluşur. Her kartta en azından bir 5 voltlu regüle entegresi ve bir 16MHz kristal osilatöre (bazlarında seramik rezonatör) bulunur. Mikrodenetleyiciye önceden bir bootloader programı yazılı olduğundan programlama için harici bir programlayıcıya ihtiyaç duyulmaz.

### // Arduino UNO SMD R3

Arduino Uno en yaygın kullanılan arduino kartıdır. 14 dijital giriş / çıkış'a (6 PWM çıkış), 6 analog girişe, bir 16 MHz kristal osilatöre, bir USB bağlantısına, bir güç girişine, bir ICSP başlığına ve bir reset düğmesine sahiptir. Heyecana kapılmaya gerek yok eğitim içerisinde tekrar bu konulara degeinilecektir. Şimdiyuk yüzeysel olarak bilinmesi yeterlidir.





- a** Sayaçlı Böcek Yakalama
- b** Oyuncaktan Email Habercisi
- c** Otomatik Ayakkabı Bağcığı

- d** Kamera Zamanlama Uygulaması
- e** Otomatik Çiçek Sulama
- f** Programlanabilir Trafik İşıkları

- g** Kahve Makinesi
- h** Quadcopter

# Arduino IDE'yi İndirme (Entegre Geliştirme Ortamı)



## İnternet Erişimi

Arduino ile yazılım geliştirmek ve çalıştmak için öncelikle [www.arduino.cc](http://www.arduino.cc) sitesinden bir Arduino Entegre Geliştirme Ortamı indirmeliyiz. Arduino IDE olarak bilinen bu yazılım arduino'yu programlamak için adeta bir kelime işlemci görevi görecektir. İnternet tarayıcınızdan aşağıdaki adrese tıklayarak size uygun Arduino IDE'yi indirebilirsiniz.



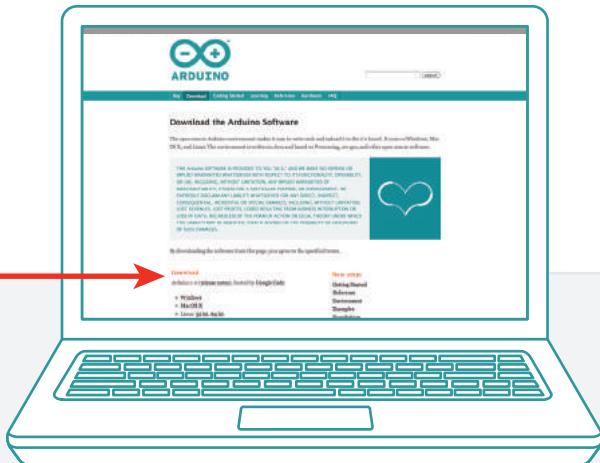
[arduino.cc/en/Main/Software](http://arduino.cc/en/Main/Software)

1

### Yükleme

Kullandığınız işletim sisteme göre size uygun olan Arduino programını seçmek için artı(+) işaretine basmanız yeterlidir.

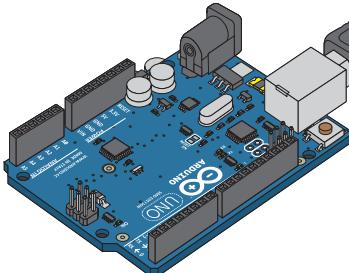
- + Windows
- + Mac OS X
- + Linux: 32 bit, 64 bit
- + source



! Bilgisayarınız için uygun işletim sisteme göre yükleme paketi seçmemi unutmayın.

## // Arduino Uno'yu Bilgisayara Bağlamak

Arduino ve Bilgisayarı USB çıkışlarından USB Kablosu ile bağlıyoruz.



2

## // Sürücü Yükleme

3

Bilgisayarınızın işletim sistemine göre ihtiyacınız olan sürücüyü indirip yükliyoruz.

### Windows Sürücü Yükleme

Aşağıdaki adreste Windows sürücü yüklemesi ile ilgili talimatları bulabilirsiniz.

<http://arduino.cc/en/Guide/Windows>



### Macintosh OS X Sürücü Yükleme

Aşağıdaki adreste MacOSX sürücü yüklemesi ile ilgili talimatları bulabilirsiniz.

<http://arduino.cc/en/Guide/MacOSX>



### Linux: 32 bit / 64 bit Sürücü Yükleme

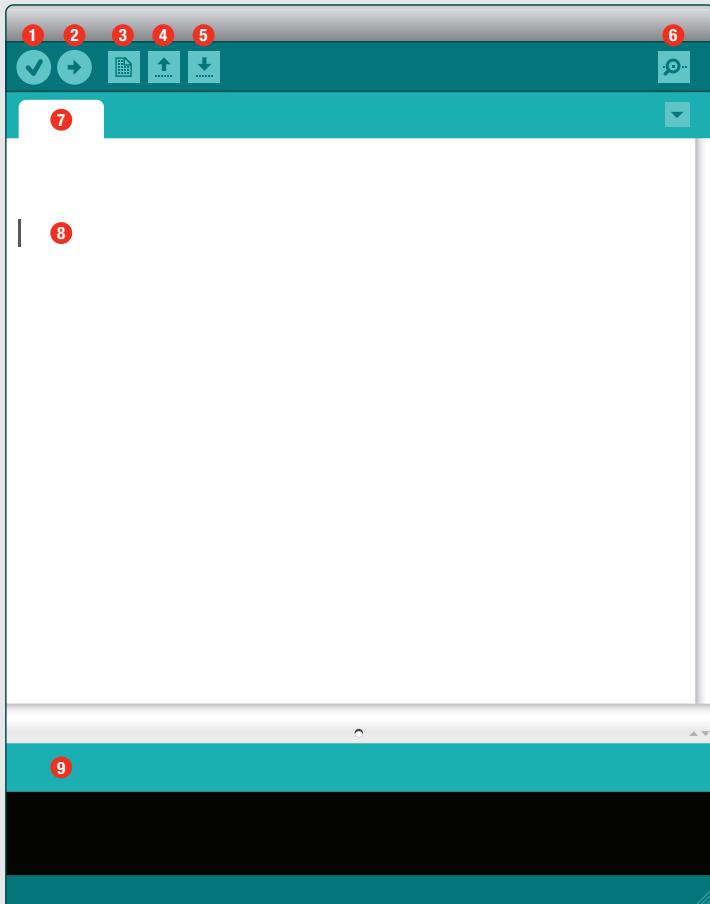
Aşağıdaki adreste Linux sürücü yüklemesi ile ilgili talimatları bulabilirsiniz.

<http://www.arduino.cc/playground/Learning/Linux>



## //Arduino IDE 'yi Açıyoruz:

Öncelikle bilgisayarımıza indirdiğimiz arduino geliştirme programını açalım. Programlamaya geçmeden önce biraz geliştirme ortamını kurcalayalım.



## GUI (Grafik Kullanıcı Arayüzü)

- 1 Verify(Derleme):** Yazmış olduğunuz kodu derler. Söz dizimi hatalarını bulur.
- 2 Upload(Yükleme):** Kodu arduino kartına gönderir.
- 3 New(Yeni):** Bu buton yeni kod penceresi açar.
- 4 Open(Açmak):** Bu buton var olan bir arduino dosyasını açmanızı sağlar.
- 5 Save(Kaydetmek):** Yazdığınız sketch'i kaydeder.
- 6 Serial Monitor(Seri monitör):** Seri bilgi görüntüleyen bir pencere açacaktır. Özellikle hata ayıklama için çok yararlıdır.
- 7 Sketch Name:** Şu anda üzerinde çalıştığımız sketch'in adını gösterir.
- 8 Code Area(Kod Alanı):** Sketch için kod oluşturacağınız alandır.
- 9 Message Area(Mesaj Alanı):** Kodunuzdaki hatayı gösteren alan .

// Bu klavuz için en önemli üç komut aşağıda görülmektedir:



*Open(Açmak)*



*Verify(Derleme)*



*Upload(Yükleme)*

# 4

## // Kartımızı Seçelim: Arduino Uno

File Edit Sketch Tools Help

Auto Format  
Archive Sketch  
Fix Encoding & Reload  
Serial Monitor

Board

Serial Port  
Programmer  
Burn Bootloader

Arduino Uno

Arduino Duemilanove w/ ATmega328]  
Arduino Diecimila or Duemilanove w/ ATmega168  
Arduino Nano w/ ATmega328  
Arduino Nano w/ ATmega168  
Arduino Mega 2560 or Mega ADK  
Arduino Mega (ATmega1280)  
Arduino Mini  
Arduino Mini w/ATmega168  
Arduino Ethernet  
Arduino Fio  
Arduino BT w/ ATmega328  
Arduino BT w/ATmega168  
LilyPad Arduino w/ ATmega328  
LilyPad Arduino w/ ATmega168  
Arduino Pro or Pro Mini (5V, 16 MHz) w/ATmega328  
Arduino Pro or Pro Mini (5V, 16 MHz) w/ATmega168  
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ATmega328  
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ATmega168  
Arduino NG or older w/ ATmega168  
Arduino NG or older w/ ATmega8



Aygıtımız için uygun olan seri portu seçmek için Tools > Serial Port menüsünü tıklyoruz. Genelde com3 veya üzerirdir (COM1 ve COM2 genellikle donanım seri portu olarak kullanılmaktadır). Kullanacağınız arduino portunu bulmak için Arduino ile bilgisayar arasındaki bağlantıya kesin ve menüye tekrar gelin, bağlantıya kestikten sonra menüde kaybolan port sizin Arduino portunuzdur.

O portu seçip arduino programlamaya başlayabiliriz.

Tools Help

Auto Format  
Archive Sketch  
Fix Encoding & Reload  
Serial Monitor

Board

Serial Port com 1  
com 12  
Programmer  
Burn Bootloader



Aygıtımız için uygun olan seri portu seçmek için Tools > Serial Port menüsünü tıklyoruz. Mac bilgisayarlarda Uno veya Mega 2560 için (**/dev/tty.usbmodem**) ve ya eski kartlar için (**/dev/tty.usbserial**) seçiyoruz.

Tools Help

Auto Format  
Archive Sketch  
Fix Encoding & Reload  
Serial Monitor

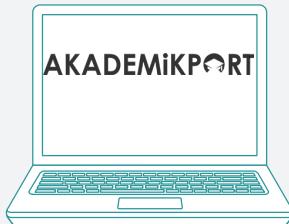
Board

Serial Port /dev/tty.usbmodem262471  
/dev/cu.usbmodem262471  
/dev/tty.Bluetooth-Modem  
/dev/cu.Bluetooth-Modem  
/dev/tty.FireFly-7256-SPP  
/dev/cu.FireFly-7256-SPP  
/dev/tty.tiPhone-WirelessIAP-1  
/dev/cu.tiPhone-WirelessIAP-1  
/dev/tty.Bluetooth-PDA-Sync  
/dev/cu.Bluetooth-PDA-Sync



<http://www.arduino.cc/playground/Learning/Linux>

5



İndirmek için aşağıdaki bağlantıyı takip edin;



Start → Programs → arduino → examples

Adreste verdığımız sıkıştırılmış dosyayı indirip, içerisindeki "Arduino Kod" dosyasını dışarı çıkartıyoruz.

"Arduino Kod" dosyasını kopyalayıp arduino kurulumu içerisindeki "examples" klasörünün içerisine yapıştırıyoruz.



Adreste verdığımız sıkıştırılmış dosyayı indirip, içerisindeki "Arduino Kod" dosyasını dışarı çıkartıyoruz.



Arduino 'yu uygulama dosyasından buluyoruz. Arduino 'ya sağ tıklıyoruz(ctrl + click) . "Show Package Contents" i seçiyoruz.

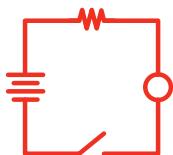


"Arduino Kod" dosyasını kopyalayıp examples klasörünün içerisine yapıştırıyoruz.



<http://www.arduino.cc/playground/Learning/Linux>

# Elektrik Devreleri



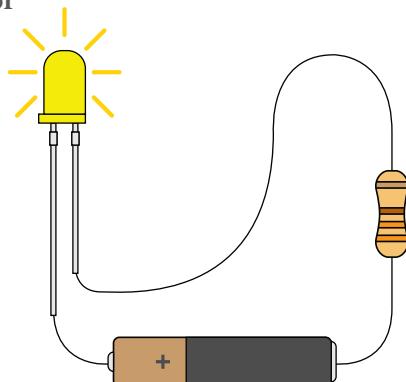
## Elektrik Devresi Nedir?

Elektrik Devresi basitçe başlangıç ve bitiş noktası olan ve içerisinde bir çok devre elemanı buluduran döngüdür. Devreler her ölçüde ve şekilde yüzlerce ; direnç, diyon, bobin, sensör, motor ve diğer devre elemanlarını barındırabilir.

Devreler genelde analog devreler, dijital devreler ve karışık sinyal devreleri olmak üzere üç kategoriye ayrılır. Bu kitapçıkta her üç devre çeşidi de gösterelecektir.

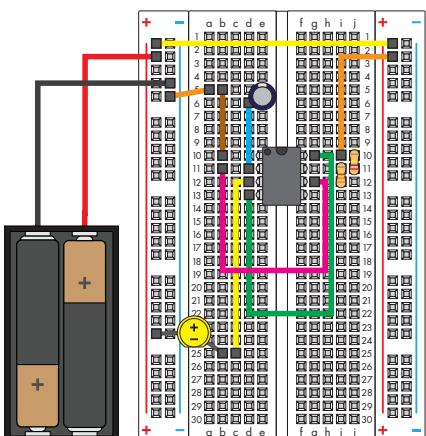
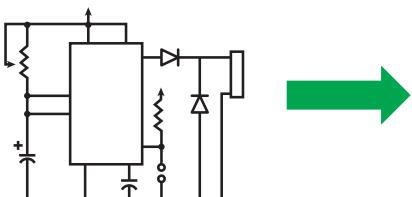
## Dünya Bu Devreler Üzerinde Dönüyor

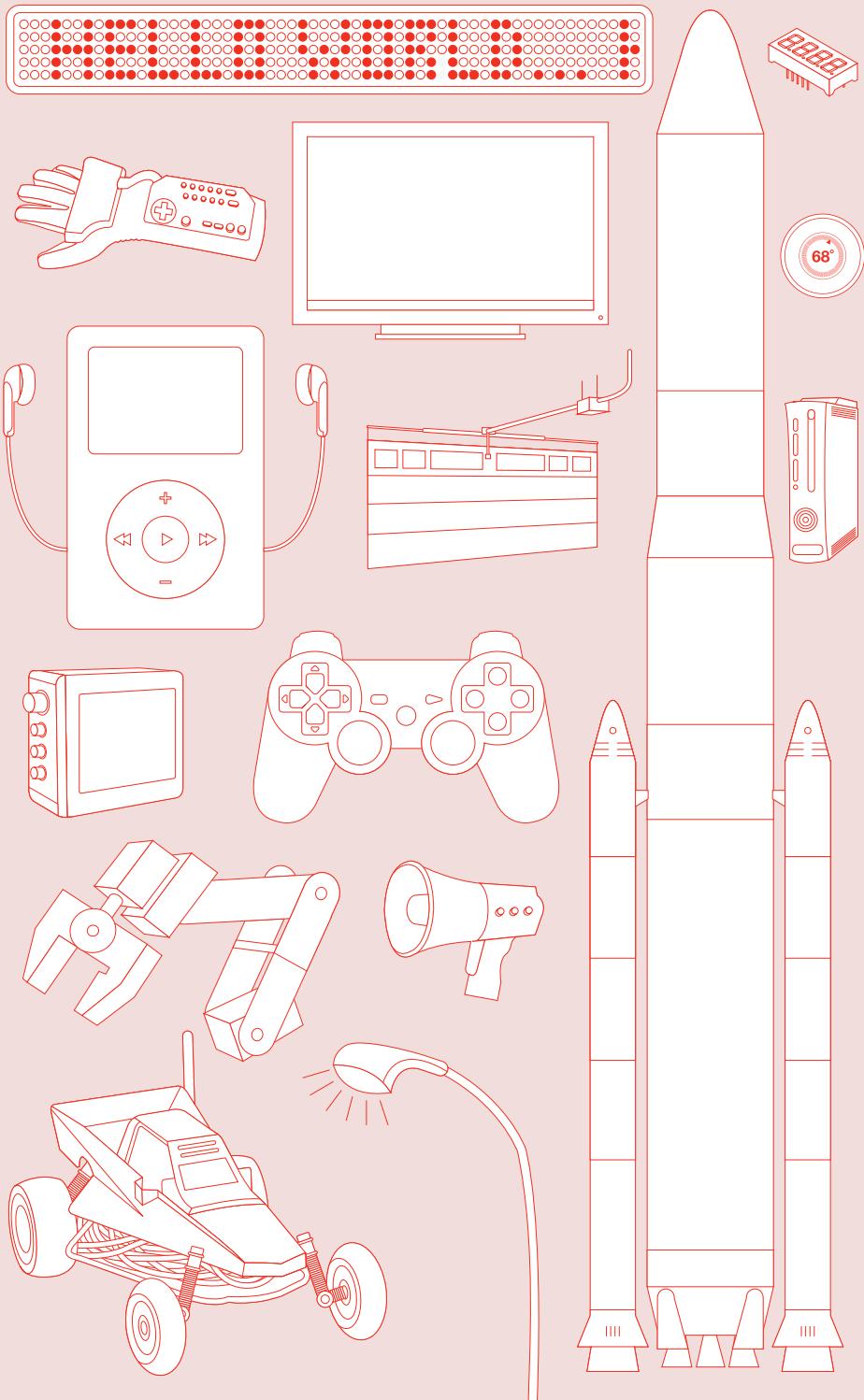
Nereye bakarsanız bakın devreler ile karşılaşacaksınız. Cebinizdeki telefon, aracınızın emisyon sistemini kontrol eden bilgisayar, oyun konsollarınız, elektrikli mutfak aletleriniz, bunların hepsi elektrik devreleriyle dolu. Bu kitapçıkta bazı basit devreleri deneyecek ve gömülü elektronik sistemlerinin temeline adım atacaksınız.



## // Basit ve Karmaşık Devreler

Bu kitapçıkta ilk olarak basit devreleri keşfedeceksiniz. Ama bu sizin basit araçlarla büyülüyici şeyler yapamayacağınız anlamına gelmiyor elbette. Bu eğitimi tamamladıktan sonra devreler hakkında öğrendiğiniz bilgiler hayal gücünüzde de bağlı olarak muhteşem projeler yapmanızı sağlayacak.



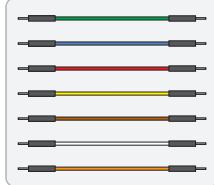


# Devre Elemanları

## Atlama Kablosu

Çeşitli Renklerde

x30



## 330Ω Direnç



x25

\*Gerçek Görünüm

## LED (5mm)

(Light Emitting Diode)



x10

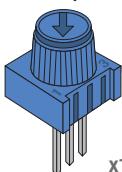


x10

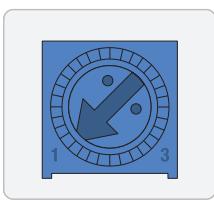


x1

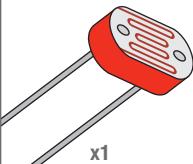
## Potansiyometre



x1



## Foto Direnç(LDR)

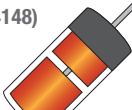


x1



## Diyot

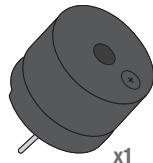
(1N4148)



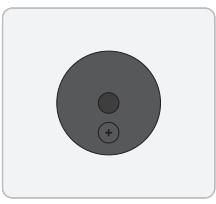
x2

\*Gerçek Görünüm

## Buzzer



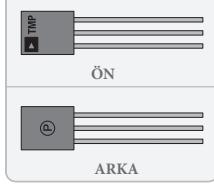
x1



## Sıcaklık Sensörü (TMP36)



x1



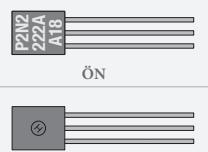
ON

ARKA

## Transistor (P2N2222AG)



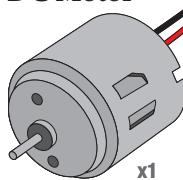
x2



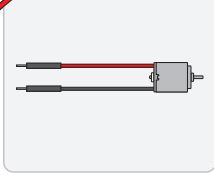
ON

ARKA

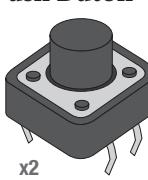
## DC Motor



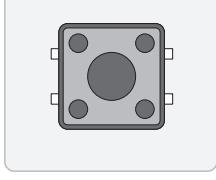
x1



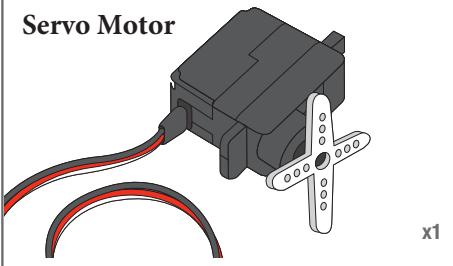
## Push Buton



x2

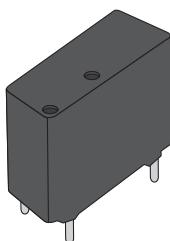


**Servo Motor**



x1

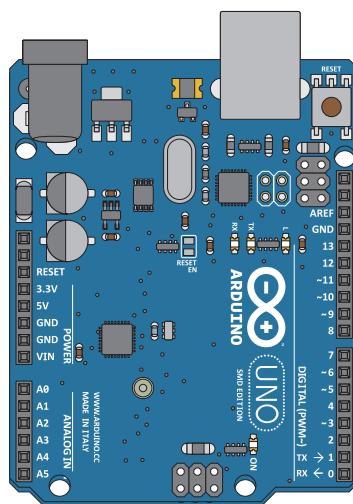
**Röle**



x1

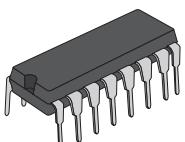
**Arduino Kartı**

UNO

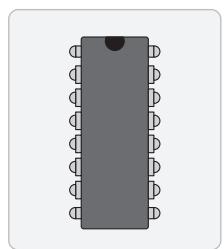


x1

**Entegre (IC)**

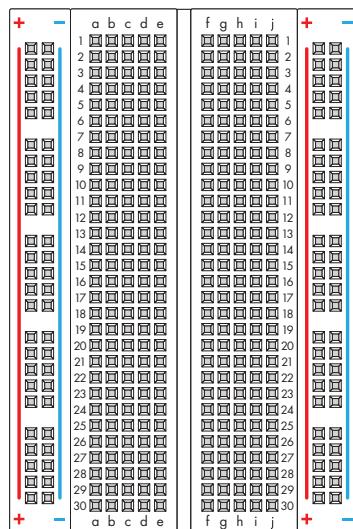


x1

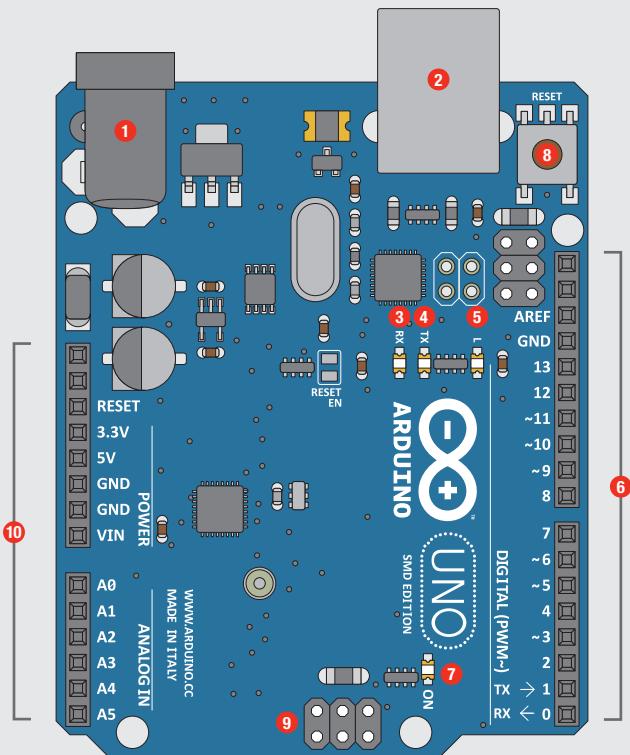


**Breadboard**

(Standart Lehimsiz)



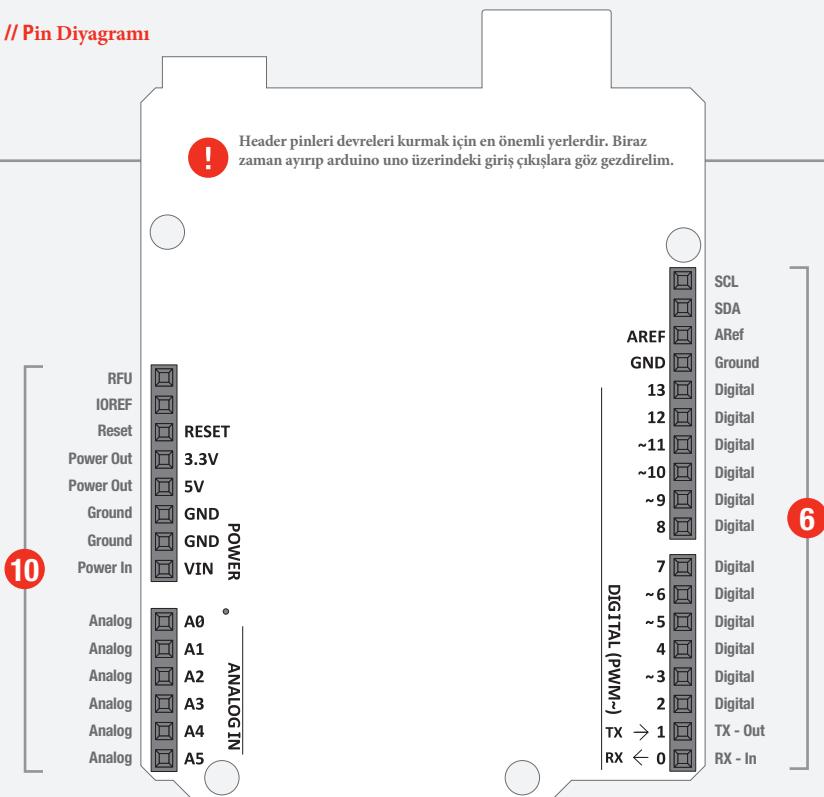
x1



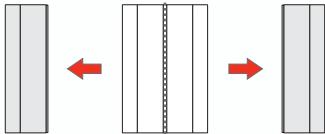
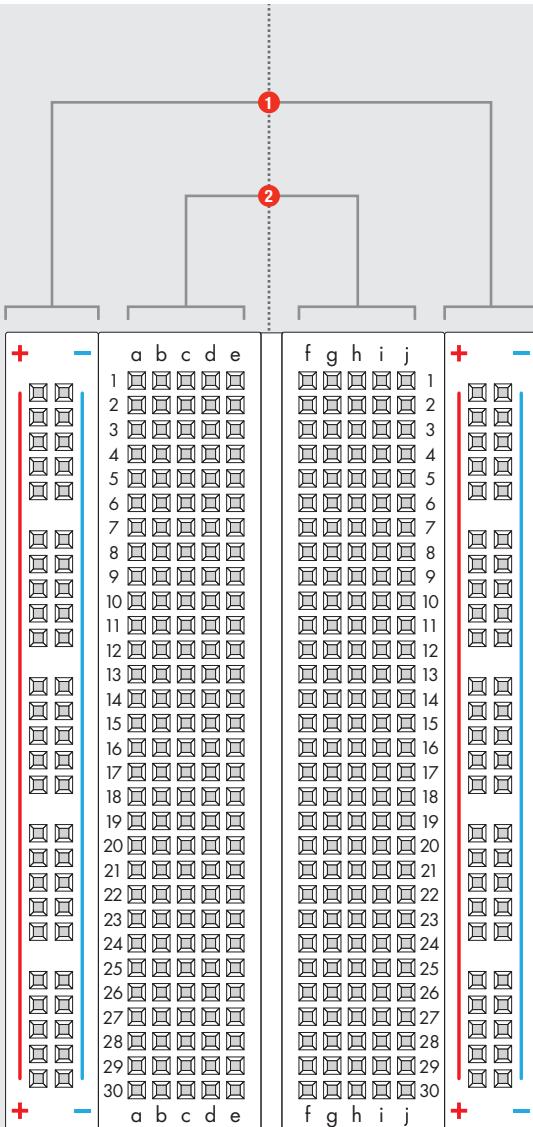
# Arduino Uno

- 1 9-12V DC Güç Girişi( Barrel Jack) - 9V veya 12V güç bağlanabilen güç girişidir .
- 2 USB Bağlantı Konnektörü(USB Port) - Arduino 'ya program yüklemek ve bilgisayar ile haberleşmek için kullanılan bağlantı konnektörüdür.
- 3 LED(RX: Receiving) - Seri haberleşme için kullanılan RX pininin durumunu gösteren LED'dir. Veri alışıverisi olduğunda bu led yanar.
- 4 LED(TX: Transmitting) - Seri haberleşme için kullanılan TX pininin durumunu gösteren LED'dir. Veri alışıverisi olduğunda bu led yanar.
- 5 LED (Pin 13: Troubleshooting) - 13 Nolu dijital pine bağlı olan LED'dir. Programları test etmek için kullanılabilir.
- 6 Dijital Giriş-Çıkış Pinleri- Dijital giriş-çıkış için kullanılan pinler burada bulunuyor.Yanında işaret bulunan pinler aynı zamanda analog çıkış(PWM) almak içinde kullanılıyor.
- 7 LED (Güç LED'i)-Kartımızın güç gösterge LED 'idir. Devrelerinizi kurarken kısa devre yapıp yapmadığınız bu LED 'e bakarak anlayabilirsiniz.
- 8 Reset Butonu -Arduinoyu yeniden başlatır.Programlarımız setup() fonksiyonundan itibaren yeniden başlar.
- 9 Kart Üzerinde Programlama(ICSP) Pinleri - Atmega microdenetleyiciyi harici bir programlayıcı ile programlama pinleri.
- 10 Analog Giriş Pinleri ve Güç Bağlantıları -Analog giriş ve gerilim bağlantıları bu bölümde bulunuyor.Ayrıca analog-dijital çeviricinin referans giriş pini ve seri iletişim pinleri de (RX ve TX) burada bulunuyor.

## // Pin Diyagramı



~~ = PWM/Analog çıkış uyumlu (i.e. ~3 )



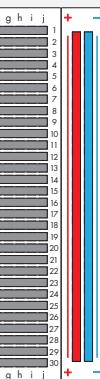
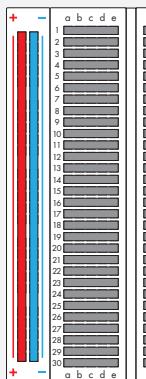
Bu çizgi breadboardumuzu ikiye böliyor, Her bölüm kendi içinde bağımsızdır.

# Breadboard

**1** Dikey Bağlantı (+ Güç ve - Toprak // Aşağıdaki diyagrama Bakınız)

**2** Yatay Bağlantı (a-e & f-j // Aşağıdaki diyagrama Bakınız)

## Bağlantılar Nasıldır?



### + Güç:

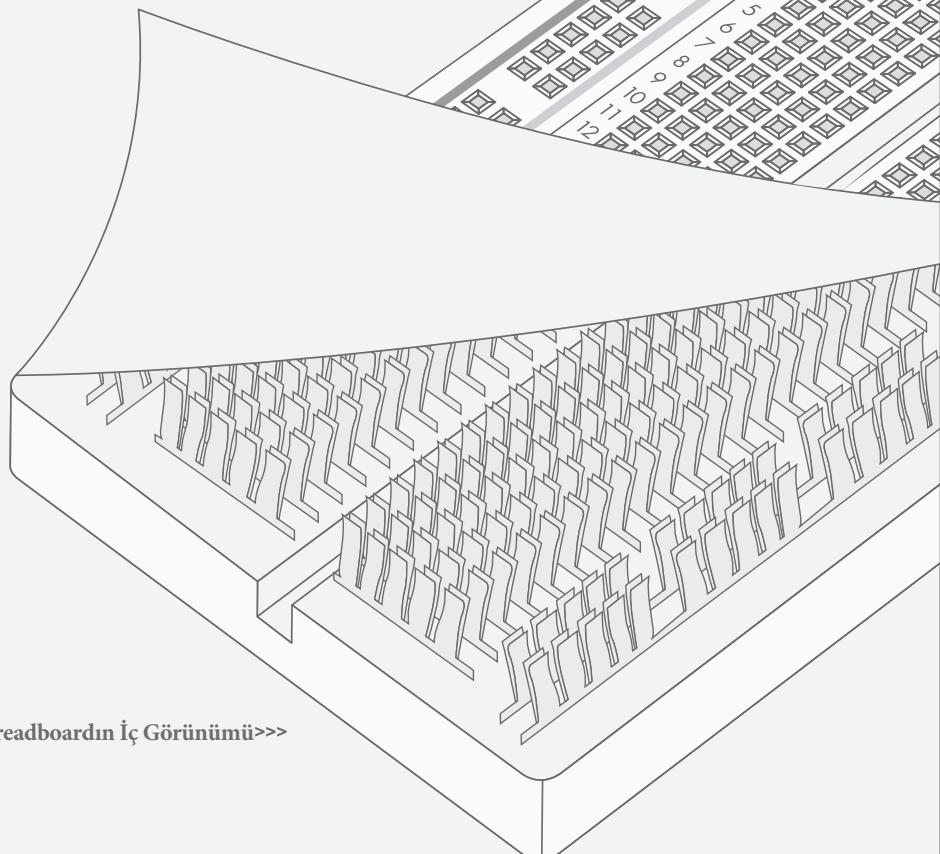
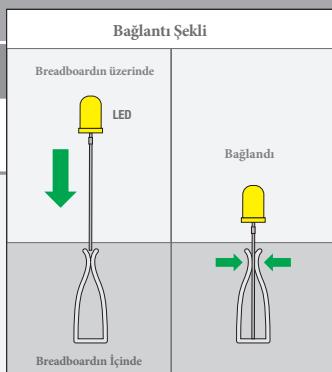
Her + işaretini dikey sütünden güç geçisi sağlar.

### - Toprak:

Her - işaretini dikey sütünden her yerde toprak görevi görür.

### Yatay Satırlar:

1 den 30 a kadar numaralı satırların herbiri kendi içerisinde (abcde ve ya fghij) güç iletimi sağlar.



[Breadboardin İç Görünümü>>](#)

# Devreler #1 - İlk Devreniz

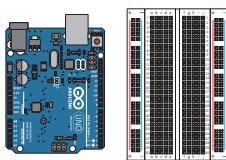
Nasıl Çalıştıracağınız;

## 1 Parçaları Birleştir

## 2 Kodu Yaz

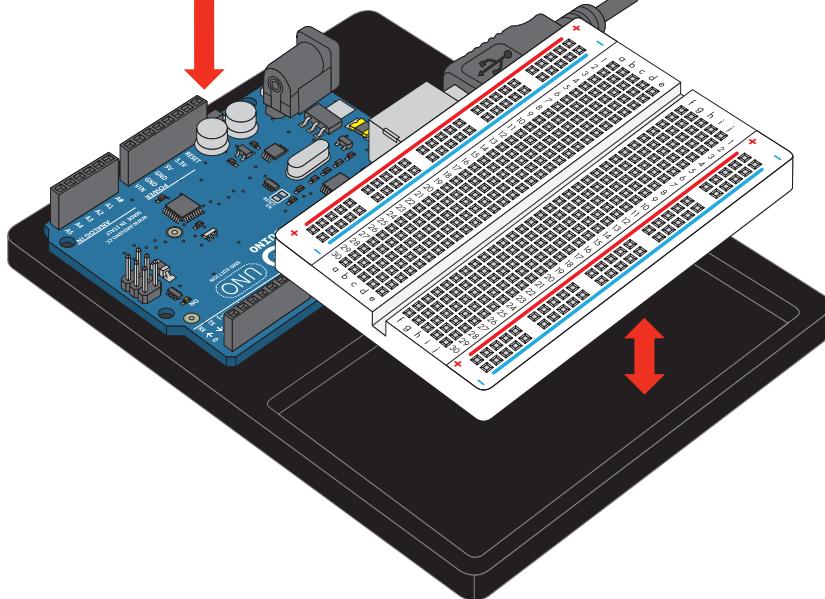
## 3 Arduino'ya Yükle

- + Breadboard hakkında genel bilgiler verildi,
- + Şimdi arduino ile breadboardu yanına yerlestirelim ve talimatlarla dikkat edelim.

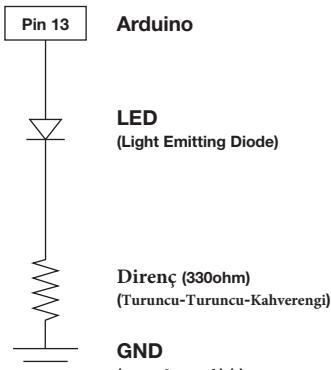


- + Arduino ve Breadboardı yanına yerleştir

- + USB Kablosunu Bağla



**5V Akım** Arduino 5V da çalışır. Arduino'yu bilgisayara bağladığınızda aldığınız güç kurdugunuz devreleri çalıştıracaktır. Bilgisayarınızı Arduino'ya bağlayarak doğru gerilimi alabilirsiniz. 5V size zarar vermez bu yüzden devre içerisinde dokunmamak.



+ Bu LED yakma devremizin şemasıdır.

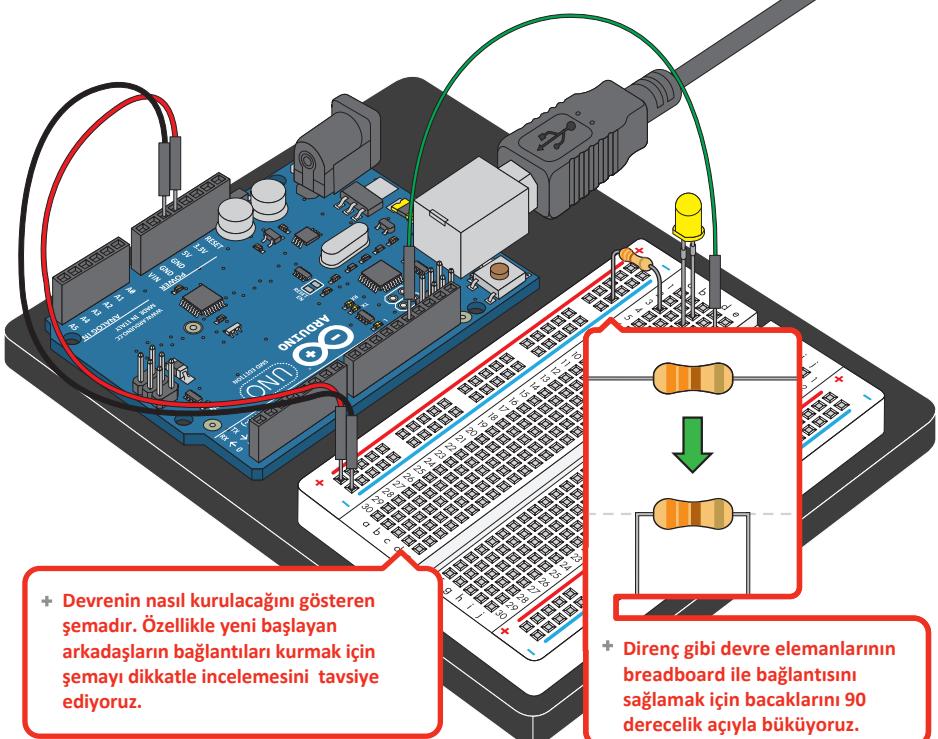
## LED YAKMAK

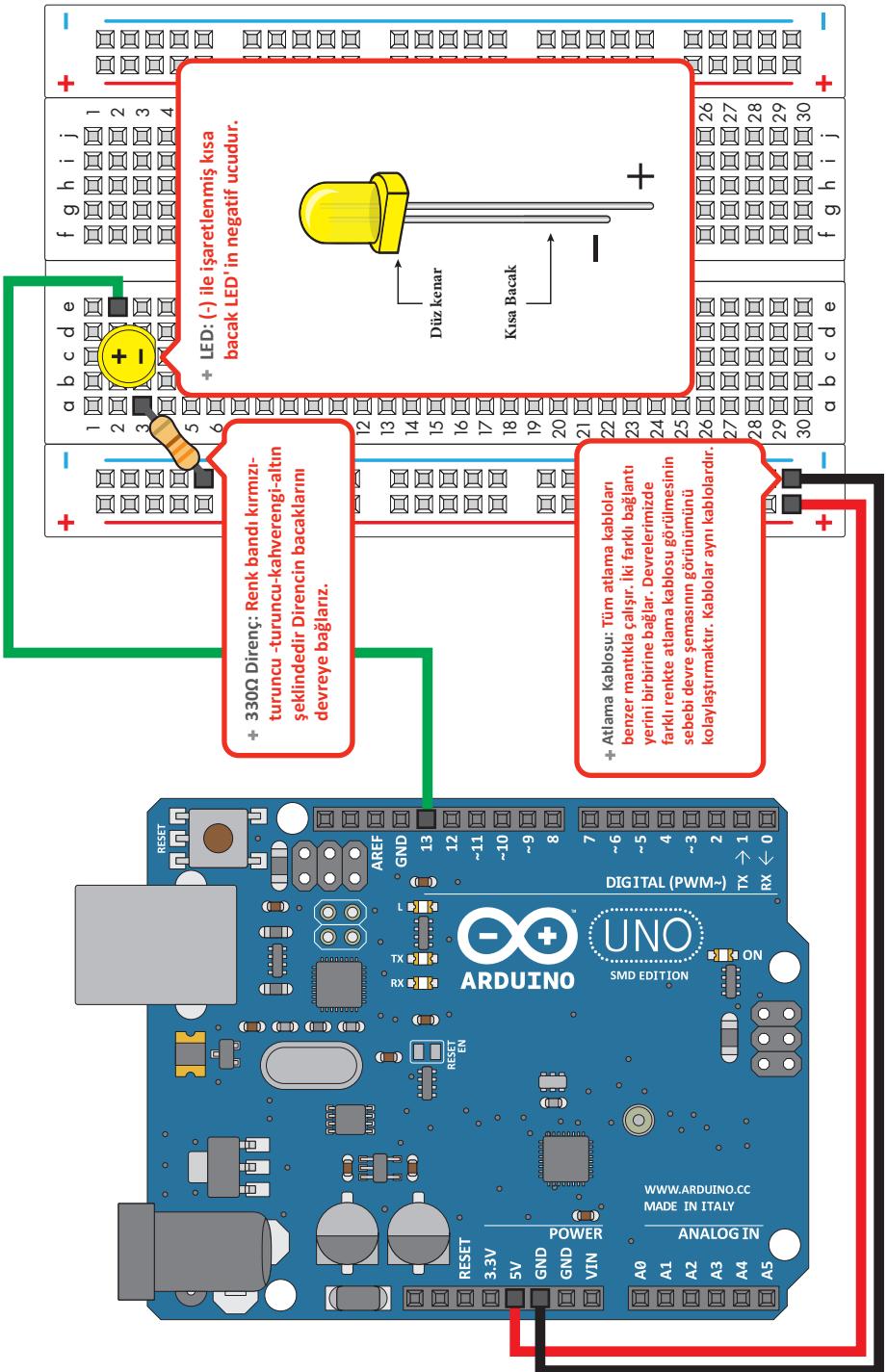
LED'ler (light-emitting diodes) bir çok elektronik devrede kullanılan güclü ışık kaynaklarıdır. Çalışmalarımıza ilki olarak her elektronik meraklısının yaptığı gibi led yakarak başlayacağız. Bir çoğunuz için çok basit olabilir ama hem devre kurma hem de programlama olarak bize katacakları şeyler daha komplex projeleri yapmak için bizlere birer basamak olacaktır.

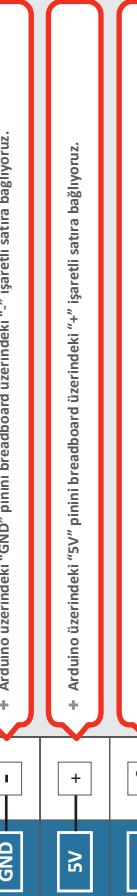
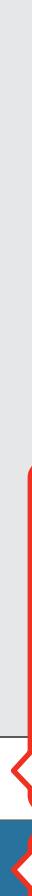
+ Bu bölümde devrelerimizi kurmadan önce kuracağımız devre ile ilgili kısa açıklamalar yer alır.



+ Bu bölümde, devreyi tamamlamak için gereken parçalar listelenir.



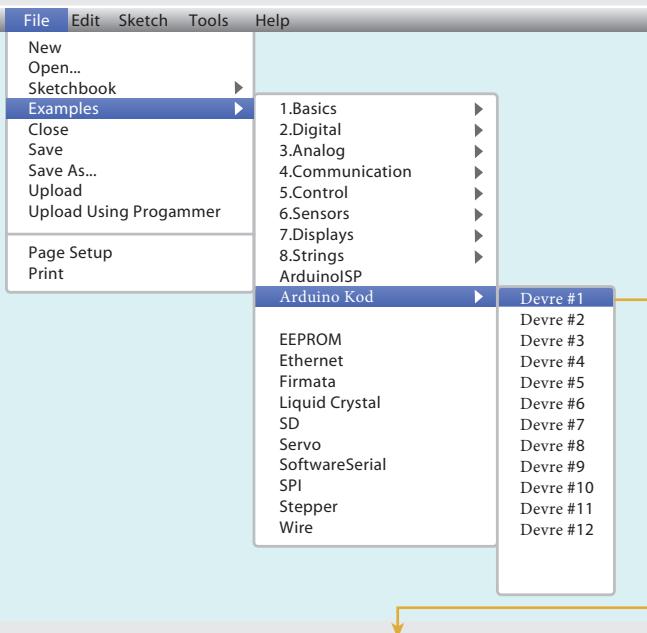


Elemanlar:	Gerçek Görünlümleri:
LED (5mm)	
330Ω Direnç	
Atlama Kablosu	
Atlama Kablosu	
Atlama Kablosu	
<p><b>Elemanlar:</b></p> <ul style="list-style-type: none"> <li>+ LED'i C2(uzun bacak) C3(kısa bacak) şekeiten breadboard'un sökötelerine taktıyoruz.</li> <li>+ Direnler yahmaz Breadboard üzerindeki sökötelerde yerleştirilir. " " işaretli satira ve LED'in bacağına bağlıyoruz.</li> <li>+ Arduino üzerindeki "GND" pininin breadboard üzerindeki " " işaretli satira bağlıyoruz.</li> <li>+ Arduino üzerindeki "5V" pininin breadboard üzerindeki "+" işaretli satira bağlıyoruz..</li> <li>+ Arduino üzerindeki "Pin 13" pinini breadboard üzerindeki "e2" sökötene bağlıyoruz.</li> </ul>	
<p><b>Breadboard:</b> Beyaz şerit breadboard sököt bağlantılarını temsil eder.</p>	
<p><b>Arduino:</b> Mavi şerit Arduino header pinlerini temsil eder.</p>	



## İlk Kodumuzu Açılm:

Bilgisayarımızdaki Arduino IDE yazılımını açıyoruz. Daha sonra aşağıdaki şemayı takip ederek "examples" klasörü içerisindeki "Arduino Kod" bölümünden "Devre #1" kodunu seçiyoruz.



// Devre #1

```
/*
Led Yakmak (Blink=Göz Kırpması)
LED' ı bir saniye yakıp bir saniye söndürün program
*/
int LEDcikis = 13;
void setup() {
    // Dijital Pinin çıkış olarak ayarlayacağız.
    pinMode(LEDcikis, OUTPUT);
}
void loop() {
    digitalWrite(LEDcikis, HIGH);           // LED'i yak
    delay(1000);                          // 1 saniye bekle
    digitalWrite(LEDcikis, LOW);           // LED'i söndür
    delay(1000);                          // 1 saniye bekle
}
```



### Derleme

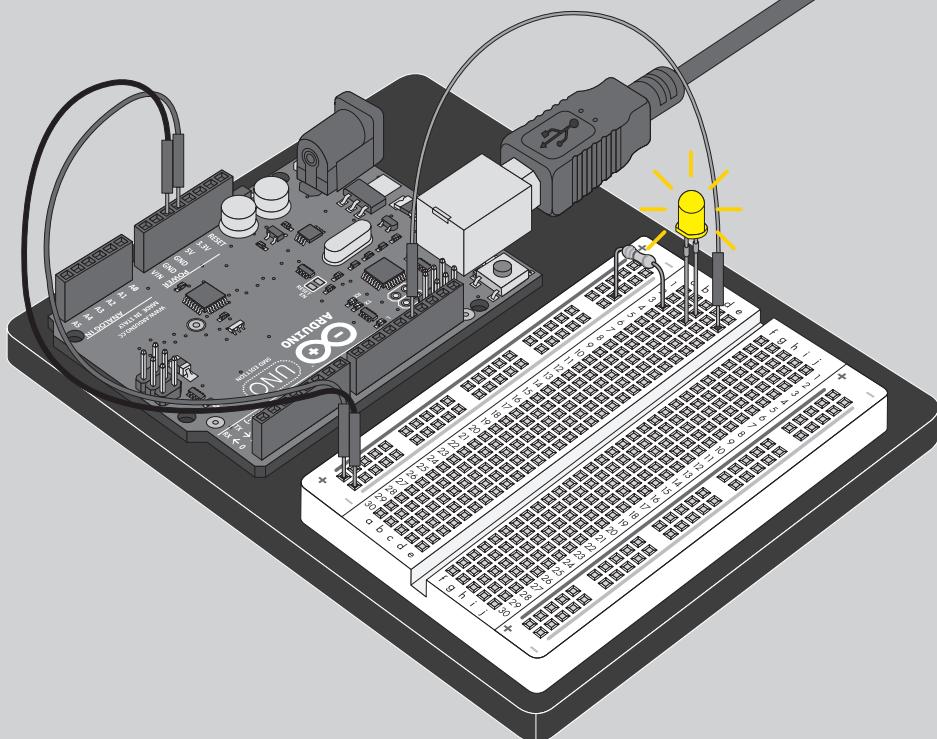
Kodu derliyoruz. Yazılım kodu arduino üzerindeki entegrenin anlayabileceği dile çeviriyor.



### Yükleme

USB Kablo aracılığıyla kodu Arduino üzerindeki entegreye gönderiyoruz. Daha sonra devre otomatik olarak çalışmaya başlayacaktır.

// Verilen kod ile derledikten ve yükledikten sonra breadboard üzerindeki LED yanıp sönmeye başlayacaktır



# 1



Arduino IDE 'yi Açı // File > Examples > Arduino Kod > Devre # 1

Kod notları:

- + Kodun nasıl çalıştığını anlamak için aşağıdaki açıklamaları inceleyin.

+ Tüm devrelerde kodların ne olduğunu açıklayan bölümdür.

- + Verilen Kodu derlemeyi(verify) ve yüklemeyi(upload) unutmayın



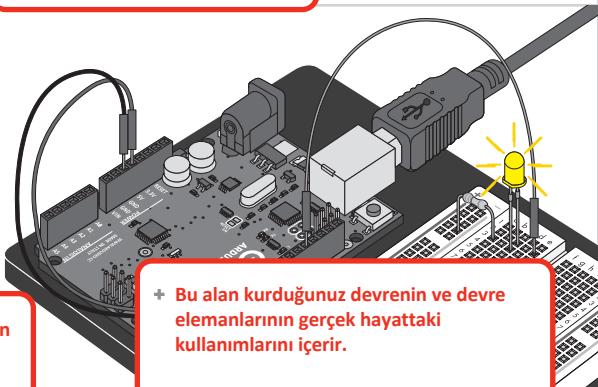
**pinMode(13, OUTPUT);** → Arduinokı pinleri kullanmadan önce pini INPUT (giriş) yada OUTPUT (çıkış) olarak tanımlamanız gerekmektedir . Bunu yapmak için **pinMode()** yerleşik fonksiyonunu kullanacağız.

**digitalWrite(13, HIGH);** → Bir pini OUTPUT (çıkış) olarak tanımladığınızda, o pinin HIGH (output 5 Volts-5 Volts çıkış), veya LOW (output 0 Volts-0 Volt çıkış) olarak davranışmasını sağlayabilirsiniz .

## Ne göreceğiz?

- + Devrenizi tamamladıysanız , aşağıdaki gibi LED yanıp sönecektir

Yanıp sönen bir LED göreceksiniz. Eğer çalışmıyorsa kodu kontrol edin ve tekrar derleyin sonra yükleyin ve devreyi tekrar kontrol edin veya aşağıdaki sorun giderme ipuçlarını bakın.



- + Bu bölüm devre montajı sırasında yapılan en yaygın hataları içerir.

- + Bu alan kurduğunuz devrenin ve devre elemanlarının gerçek hayatı kullanımlarını içerir.

## Sorun Giderme:

### LED Yanmıyor

LED sadece tek yönlü çalışır. Bu tür sorunlar genelde LED in yanlış bacagının bağlanmasından kaynaklanıyor. Endişeye gerek yok Led in bacaklarını çevirin ve yeniden bağlayın.

### Program Yüklenmiyor

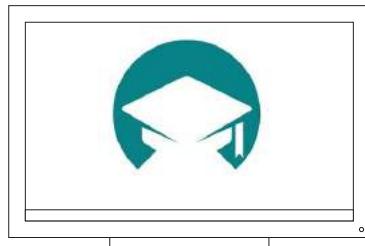
Yükleme ile ilgili sorunlar genelde seri port'dadır. tools>serial port> sekmesinden portunuzu değiştirip tekrar yüklemeye çalışın.

### Hâlâ Çalışmıyor mu?

Tamamlayamadığınız her devreniz hakkında yardım almak için [emrearslan@elektrikport.com](mailto:emrearslan@elektrikport.com) adresine mail yollayabilirsiniz.

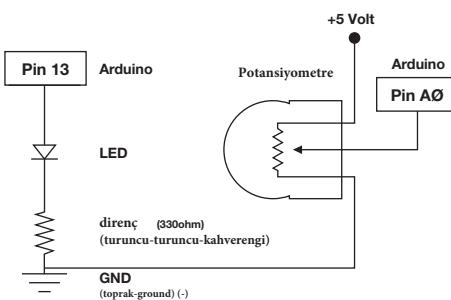
## Gerçek Hayatta Uygulamaları:

Hemen hemen tüm modern televizyon ve monitörlerde ürünün çalıştığını ya da standby durumunda olduğunu gösteren LED ler bulunur.



## Potansiyometre

Bu devrede potansiyometrenin ne iş yaradığını öğreneceğiz. Potansiyometre değişken bir direnç olarak bilinir. İki çıkış pini 5 Volta bağlandığında ortadaki pin potansiyometredeki kontrol düğmesinin konumuna göre 0V ile 5V arasında çıkış verir. Bu devrede, LED parlaklığını kontrol etmek için potansiyometreyi nasıl kullanacağınızı öğreneceksiniz.



Elemanlar:



x1

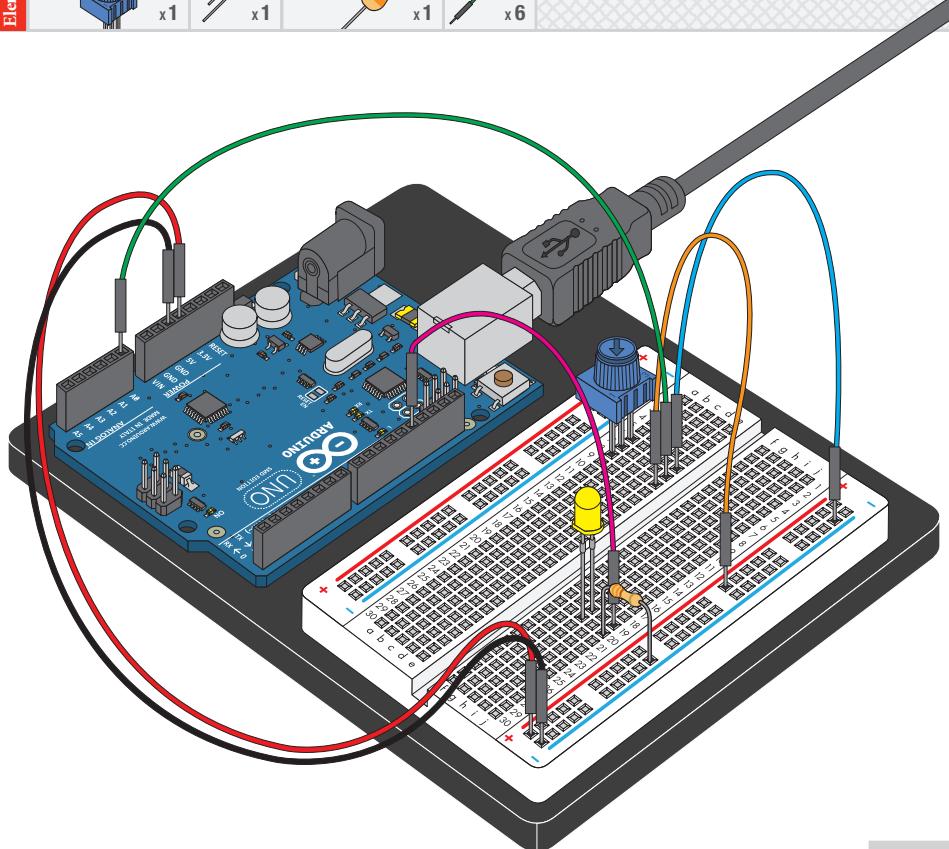


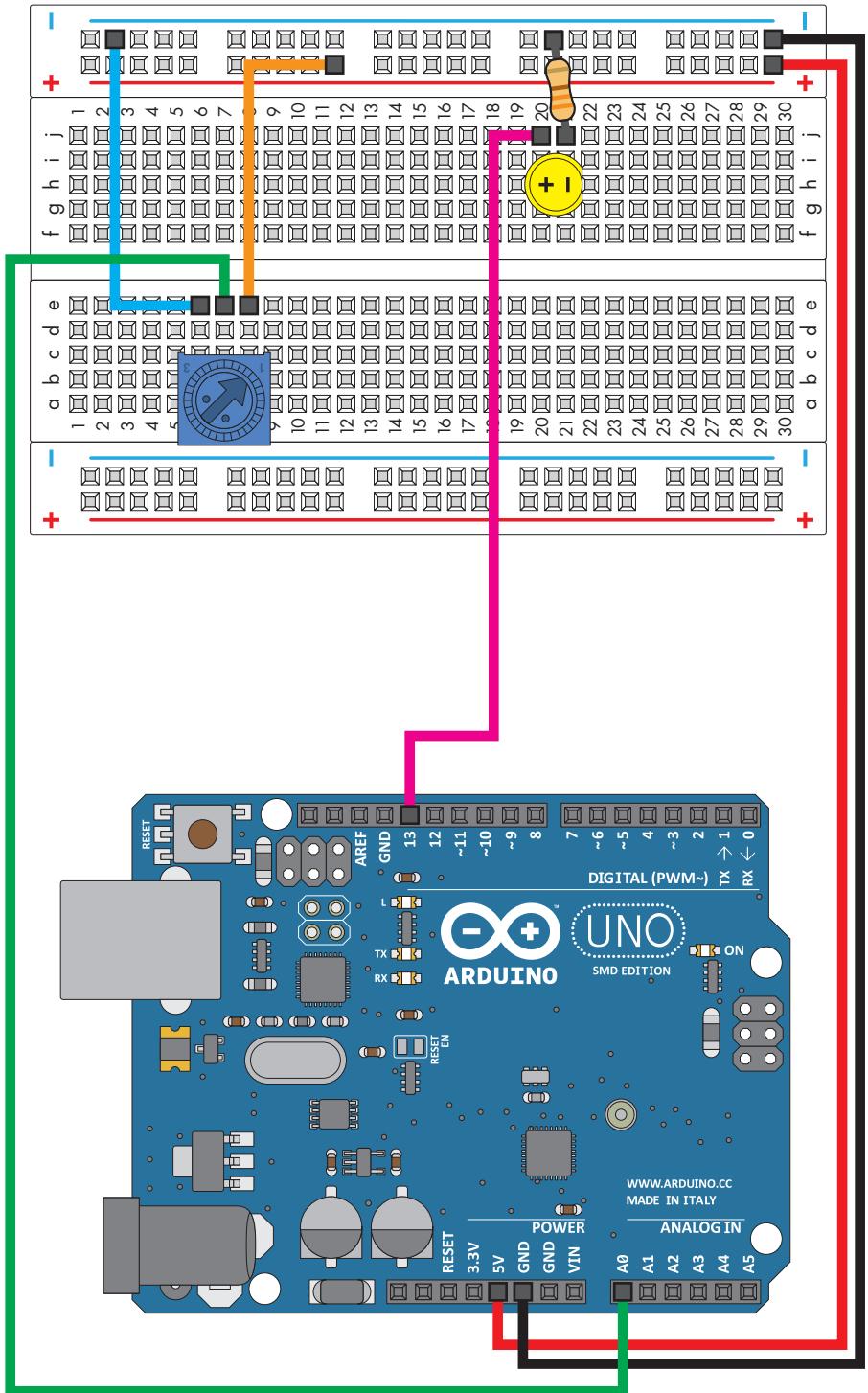
x1

330Ω Direnç



x1

Kablo  
x6



Devre 2: Potansiyometre

Digital V Analog:	
Elemenlar:	Gerçek Görünüm:
Potansiyometre	
LED (5mm)	
330Ω Direnç	
Atlama Kablosu	
Atlama Kablosu	
Atlama Kablosu	
Atlama Kablosu	
Atlama Kablosu	
Dijital	
Analog	

Elemanlar:	Gerçek Görünüm:	
Potansiyometre		
LED (5mm)		
330Ω Direnç		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		
Dijital		
Analog		



Arduino IDE 'yi Açı // File > Examples > Arduino Kod > Devre #2

### Kod Notları:



**int sensorValue;**



"Değişken" sizin isim verdığınız bir rakamdır. Değişkeni kullanmadan önce onu tanımlamalı veya bildirmelisiniz; burada sensorValue adında bir değişkeni "int" (integer-tamsayı) olarak bildiriyoruz. Bu değişken isimlerinin küçük-büyük harf duyarlı olduğunu unutmayın.

**sensorValue = analogRead(sensorPin);**



Analog pindeki değeri okumak için analogRead() fonksiyonunu kullanırız. analogRead() kullanmak istediğiniz analog pindeki ("sensorPin") değeri okur ve bir rakama("sensorValue") dönüştür, bu rakam 0 (0 Volt) ile 1023 (5 Volt) arasındaki.

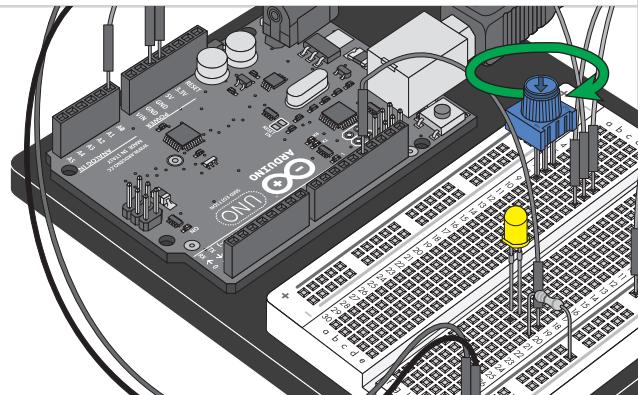
**delay(sensorValue);**



Arduino her saniye binlerce satır kodu çalıştıracak kapasiteye sahip, çok hızlı bir yapıdır. Onu yavaşlatıp neler yaptığı gözlemlmek için bazen kodlar arasına ("delay") gecikmeler ekleriz. Delay() fonksiyonu milisaniye ile sayar; yani 1 s gecikme için parantez içine 1000ms yazılmalıdır.

## Ne göreceğiz?

Potansiyometrenin ayarını değiştirdikçe LED'in hızı ve yavaş bi şekilde yanıp söndüğünü göreceksiniz. Eğer çalışmıyororsa kodu kontrol edin ve tekrar derleyin sonra yükleyin ve devreyi tekrar kontrol edin veya aşağıdaki sorun giderme ipuçlarını bakın.



### Sorun Giderme:

#### Düzensiz Çalışma Durumu

Büyük olasılıkla potansiyometrenin pinlerinin breadboarda tam oturmamasından kaynaklanabilir. Potansiyometreyi bastırarak bu sorunu çözebilirsiniz.

#### Çalışmama Durumu

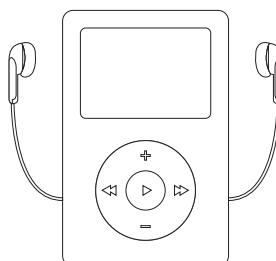
Potansiyometrenin 2. pinini analog pin yerine dijital pine bağlamadığınızdan emin olunuz.

#### Hâlâ Çalışmıyor mu?

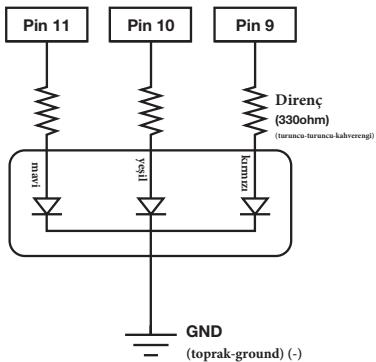
Tamamlayamadığınız her devreniz hakkında yardım almak için [emrearslan@elektrikport.com](mailto:emrearslan@elektrikport.com) adresine mail yollayabilirsiniz.

### Gerçek Hayatta Uygulamaları:

MP3 çalarlardaki ses kontrolü potansiyometre kullanımına bir örnektir.



## RGB LED



Yanıp sönen bir LED'den daha eğlenceli şey nedir sizce? Tabii ki renkli LED'ler! RGB, yani red-blue-green LED'ler 3 adet renk barındırın ve her tür renk için kombin edilebilen bir diyon türüdür. Bu devrede Bir RGB LED kullanarak nasıl renk kombinasyonları oluşturabileceğinizi öğreneceksiniz. Her bir diyonun parlaklığuna bağlı olarak neredeyse bütün renkleri elde etmeniz mümkün.

Elemanlar:



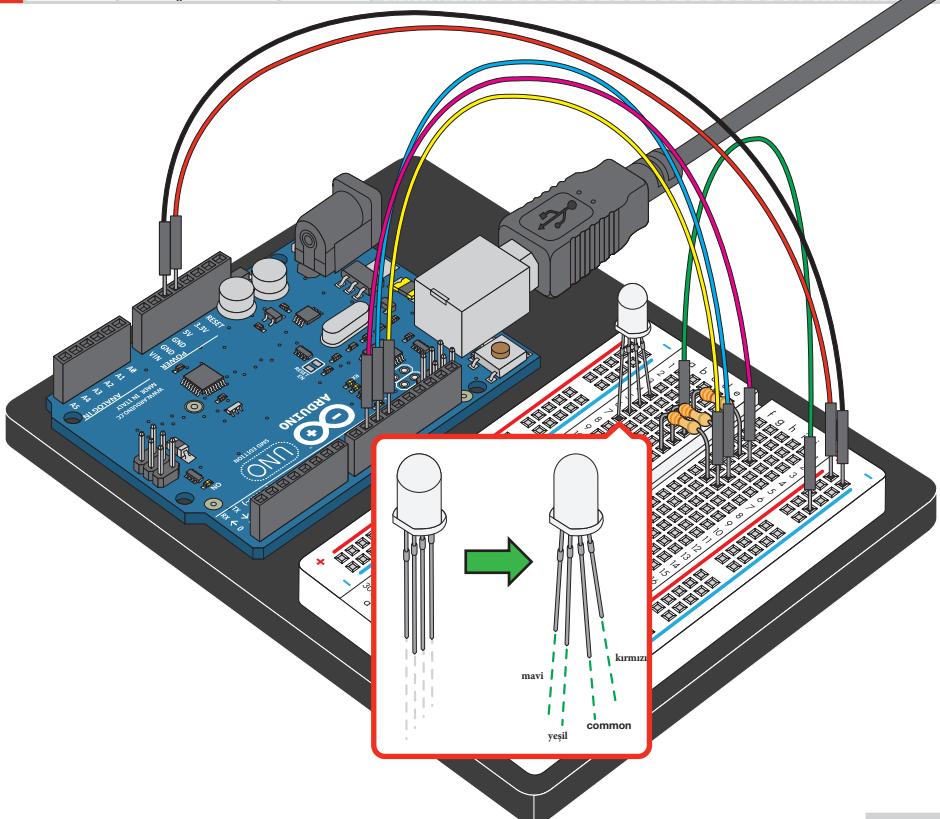
330Ω

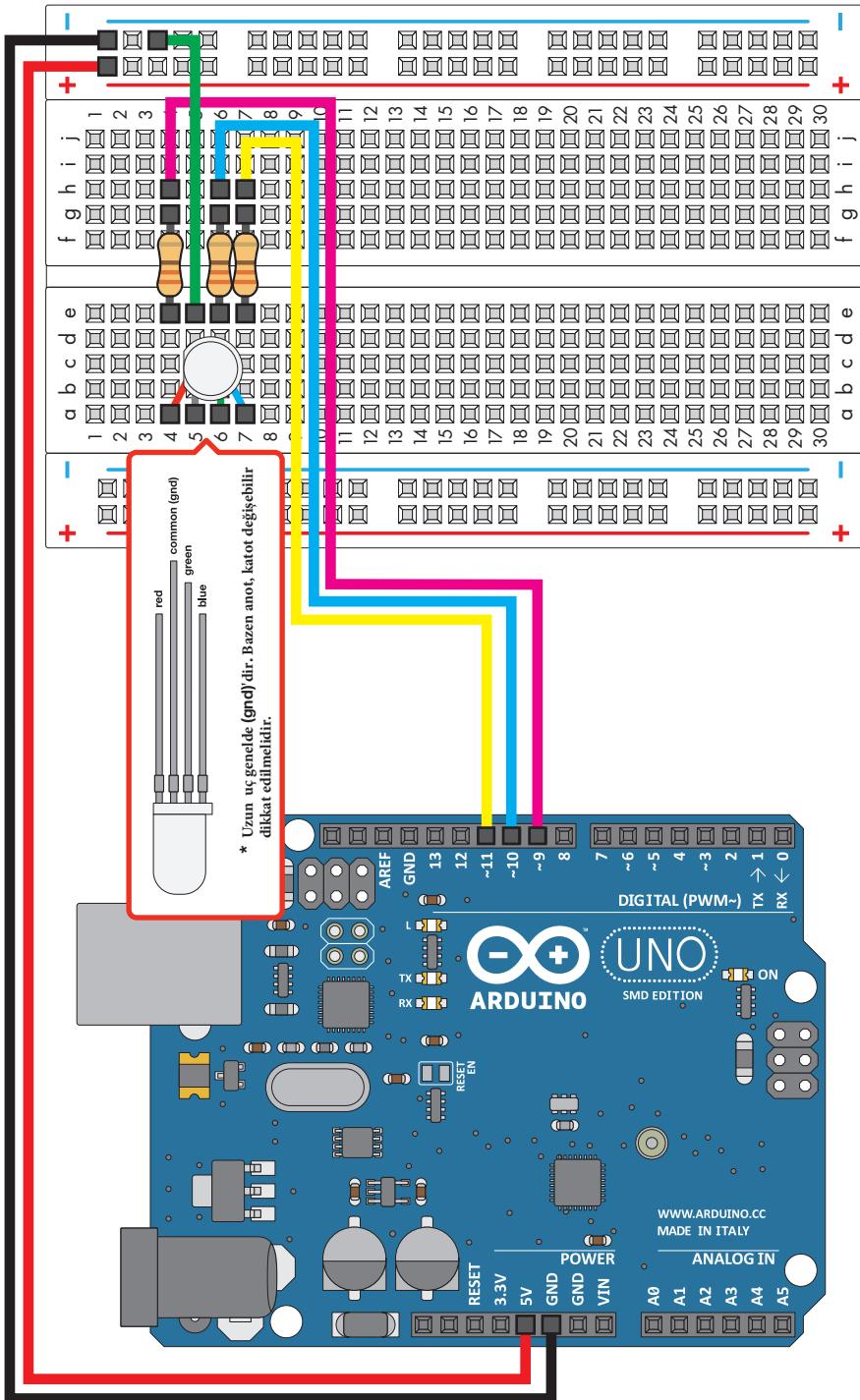


Direnç x3

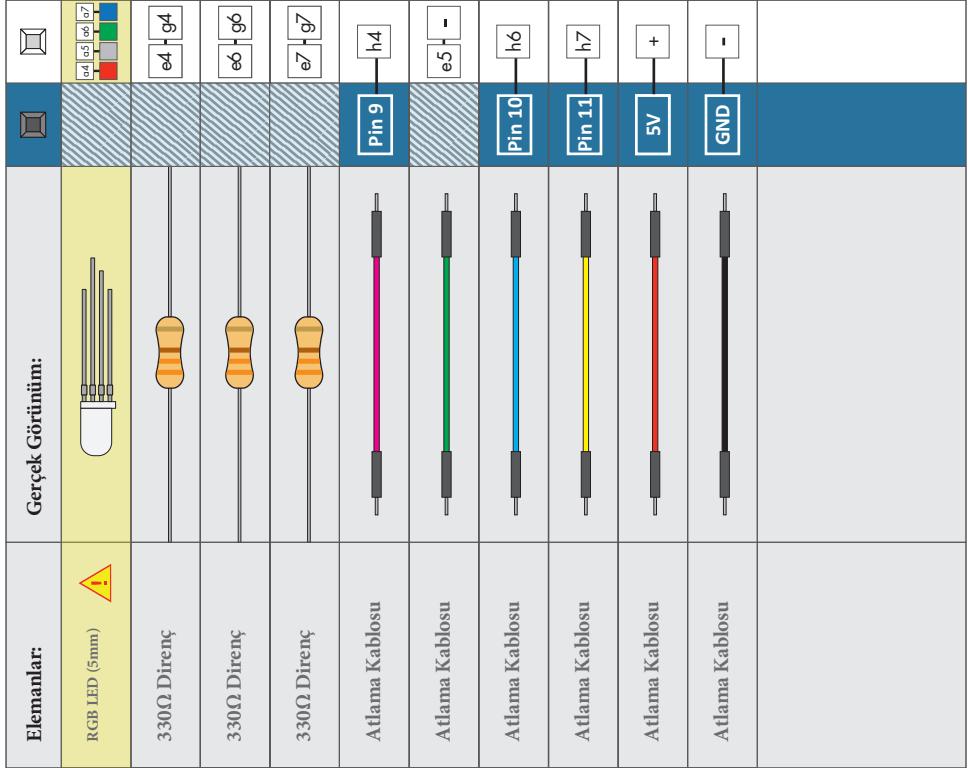


Kablo x6





analogWrite() Arkasındaki Şok Edici Gerçek																												
<p><b>Elementlar:</b></p> <ul style="list-style-type: none"> <li>RGB LED (5mm)</li> <li>330Ω Direnç</li> <li>330Ω Direnç</li> <li>330Ω Direnç</li> <li>Atlama Kablosu</li> <li>Atlama Kablosu</li> <li>Atlama Kablosu</li> <li>Atlama Kablosu</li> <li>Atlama Kablosu</li> <li>Atlama Kablosu</li> <li>Atlama Kablosu</li> <li>5V</li> <li>GND</li> </ul> <p><b>Gerçek Görünüm:</b></p> <p><b>Şözdışı:</b></p> <p>Şözdışı kadar Arduino'nun analog voltajları (0 ve 5 volt arasında ki değerleri) analogRead() fonksiyonu kullanarak okuyabildiğini gördük. Peki Arduino'nun analog voltaj çıkışını vermek gibi bir imkanı var mı diye soracak olursak, cevabımız hem hayır hem evet olacaktır. Arduino gerçek bir analog voltaj çıkışına sahip değildir ama Arduino'nun çok hızlı olmasına dolay <b>PWM ("Pulse-Width Modulation")</b> kullanarak bu çıkıştı uyutabilir*</p> <p><b>Anahtarlar:</b></p> <p>Arduino o kadar hızlı çalışır ki bir pini saniyede 1000 kez açıp kapatabilir. PWM HIGH ve LOW olarak harcanan zamanı kullanıyor. Eğer HIGH konumunda daha fazla vakt harcamıyorsa bu pini bağlı olan LED parlak yanacaktır. Fakat LOW konumunda daha fazla vakt harcamışsa LED daha sönük kalacaktır. İşte bu Arduinonun "gerçek" analog çıkış için yapıtı bir lüzyondur.</p> <table border="1"> <thead> <tr> <th>Voltaj (V)</th> <th>Pin 9 (LOW)</th> <th>Pin 9 (HIGH)</th> <th>Pin 10 (LOW)</th> <th>Pin 10 (HIGH)</th> <th>Pin 11 (LOW)</th> <th>Pin 11 (HIGH)</th> </tr> </thead> <tbody> <tr> <td>0.5</td> <td>10%</td> <td>90%</td> <td>10%</td> <td>90%</td> <td>10%</td> <td>90%</td> </tr> <tr> <td>2.5</td> <td>50%</td> <td>50%</td> <td>10%</td> <td>90%</td> <td>10%</td> <td>90%</td> </tr> <tr> <td>4.5</td> <td>10%</td> <td>90%</td> <td>10%</td> <td>90%</td> <td>10%</td> <td>90%</td> </tr> </tbody> </table>	Voltaj (V)	Pin 9 (LOW)	Pin 9 (HIGH)	Pin 10 (LOW)	Pin 10 (HIGH)	Pin 11 (LOW)	Pin 11 (HIGH)	0.5	10%	90%	10%	90%	10%	90%	2.5	50%	50%	10%	90%	10%	90%	4.5	10%	90%	10%	90%	10%	90%
Voltaj (V)	Pin 9 (LOW)	Pin 9 (HIGH)	Pin 10 (LOW)	Pin 10 (HIGH)	Pin 11 (LOW)	Pin 11 (HIGH)																						
0.5	10%	90%	10%	90%	10%	90%																						
2.5	50%	50%	10%	90%	10%	90%																						
4.5	10%	90%	10%	90%	10%	90%																						



# 3



Arduino Kodu:

Kod Notları:



**for (x = 0; x < 768; x++)  
{}**



Bir for() döngüsü bir aralıkta sayı yerleştirmek için kullanılır ve tekrarlayarak brackets{} içerisinde kodu çalıştırır. Değişken bir "x" 0 olarak başlıyor ve 767 sayısına kadar her adımda bir artarak ilerliyor ("x++")..

**if (x <= 255)  
{}  
else  
{}**



"If / else" ifadeleri programınızda seçim yapmak için kullanılır. Parantez () içerisindeki ifade değerlendirildiğinde, eğer ifadesiniz doğru ise ilk brackets{} içerisindeki ifade yürütülür. Şayet doğru değil ise ikinci bracket {} içerisindeki ifade yürütülür.

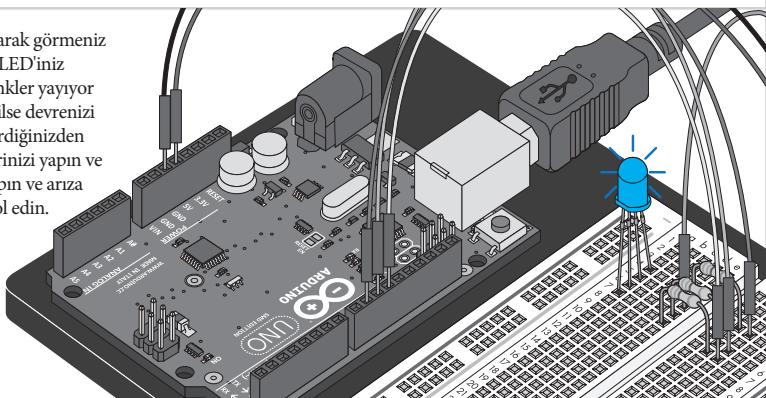
**delay(sensorValue);**



Arduino çok hızlı çalışmaktadır. Öyle ki her saniye binlerce kod satırını çalıştırabilecek kapasiteye sahiptir. Biz de cihazın ne yaptığına görebilmek için yavaşlatmak adına gecikmeler ekliyoruz. Delay() milisaniye olarka hesaplanıyor; 1 saniye 1000 milisaniye.

## Ne göreceğiz?

LED'inizi çalışıyor olarak görmeniz lazımlı, fakat bu sefer LED'iniz sırayla farklı farklı renkler yayıyor olmalı. Eğer öyle değilse devrenizi doğru şekilde birleştiridiğinizden emin olun, kontrollerinizi yapın ve kartınıza yüklemeye yapın ve arıza tespiti kısmını kontrol edin.



## Sorun Giderme:

**LED'in Işık Vermemesi veya Yanlış Renk Vermesi**  
LED'in dört pinin birbirine çok yakın bulunmakta, kolaylıkla yerlerini karıştırabilirsiniz. Pinlerin nerede olması gerektiğini iki kez kontrol edin. Ayrıca uzun uç bazen GND olmamayı, uzun ucu Arduino kartınızın 5v çıkışına bağlayıp tekrar deneyiniz.

### Kırmızı Rengi Görmek

RGB LED içerisindeki kırmızı diyon diperlerinden daha parlaklı olabilir. Renklerinizi daha dengeli hale getirmek için daha yüksek bir direnç kullanın veya kodunuza ayarlamalar yapın.

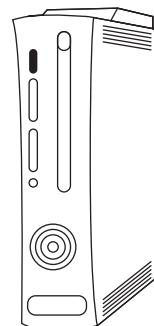
```
analogWrite(RED_PIN, redIntensity);
```

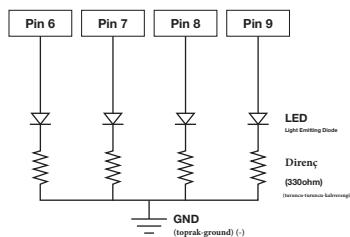
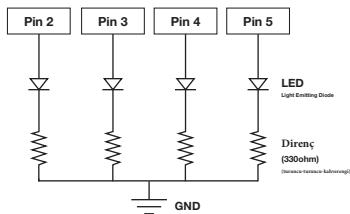
```
to
```

```
analogWrite(RED_PIN, redIntensity/3);
```

## Gerçek Hayatta Uygulamaları:

Videogame konsolları gibi bir çok elektronikte RGB LED kullanılmaktadır. Bu LED'ler aynı bölgede farklı renkleri göstermek için kullanılıyor. Sıklıkla farklı renkler farklı çalışma şartlarını ifade eder





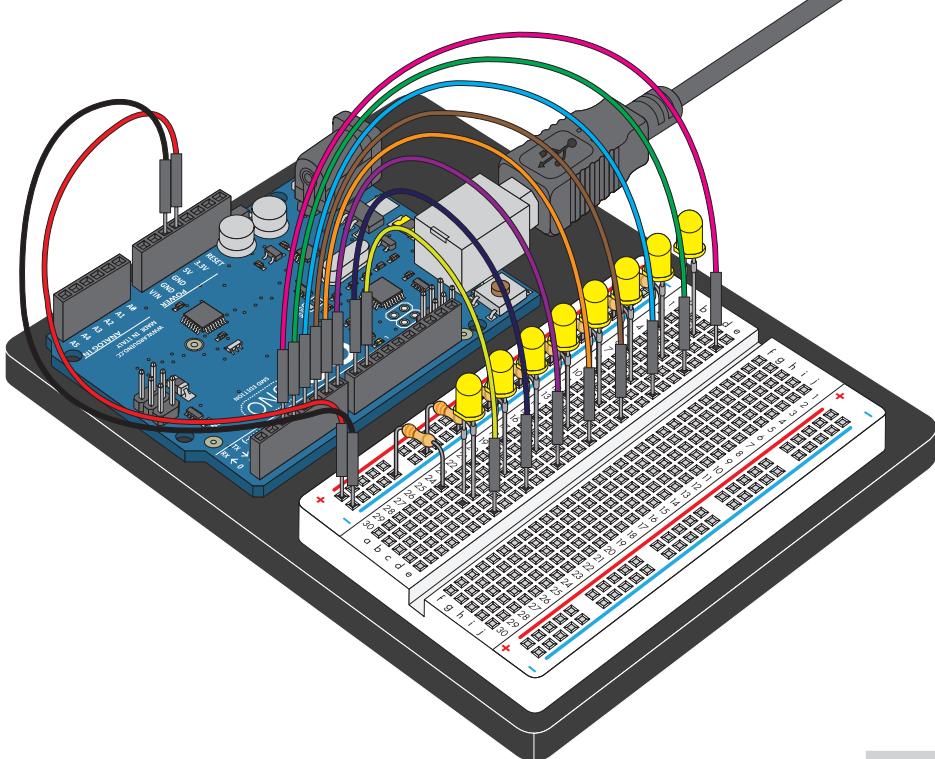
## Çoklu LED

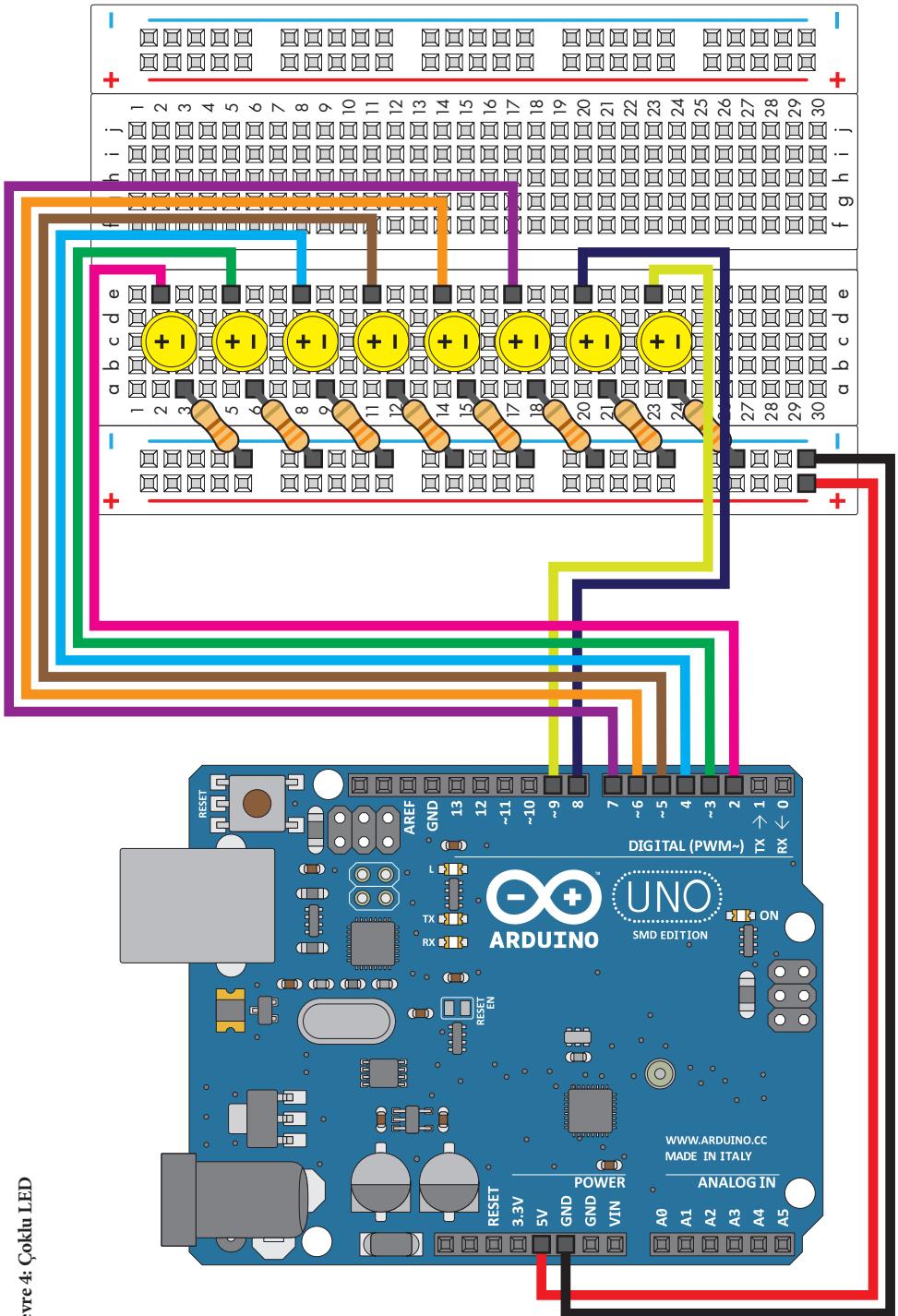
Elimizde yakıp söndürmek için bir LED var. Gelin şimdi **SEKİZ LED'i** anda bağlayarak çitayı biraz yükseltelim. Böylece çeşitli renkler oluşturma konusunda Arduinomuzu da ufak bir teste tabi tutmuş olacağız. Bu devre kendi programınızı yazma pratikleri için güzel bir başlangıç adımı olacak. LED'leri kontrol aşamasında yazdığımız programı düzenli tutmanızı sağlayacak bir kaç ipucu öğrencenkinsiniz.

**for() loops** - Bu döngü birden fazla kez aynı kodu kullanmanızı sağlar.

**arrays[ ]** - Birden fazla değişkeni grup haline getirerek yönetilmelerini kolaylaştırır.

Elemanlar:

330Ω  
Direnç



Elemanlar:	Gerçek Görünüm:	Elemanlar:	Gerçek Görünüm:
LED (5mm)		c2 c3 + -	330Ω Direnç
LED (5mm)		c5 c6 + -	330Ω Direnç
LED (5mm)		c8 c9 + -	330Ω Direnç
LED (5mm)		c11 c12 + -	Atlama Kablosu
LED (5mm)		c14 c15 + -	Atlama Kablosu
LED (5mm)		c17 c18 + -	Atlama Kablosu
LED (5mm)		c20 c21 + -	Atlama Kablosu
LED (5mm)		c23 c24 + -	Atlama Kablosu
330Ω Direnç		c23 c24 + -	Atlama Kablosu
330Ω Direnç		GND c6 + -	Atlama Kablosu
330Ω Direnç		GND c9 + -	Atlama Kablosu
330Ω Direnç		GND c12 + -	Atlama Kablosu
330Ω Direnç		GND c15 + -	Atlama Kablosu
		e2	
		e5	
		e8	
		e11	
		e14	
		Pin 2	
		Pin 3	
		Pin 4	
		Pin 5	
		Pin 6	
		Pin 7	
		Pin 8	
		Pin 9	
		Pin 3	+
			-
		GND	

# 4



Arduino IDE 'yi Açı // File > Examples > Arduino Kod > Devre # 4

Kod Notları:



**int ledPins[] = {2,3,4,5,6,7,8,9};**



“array” çok fazla değişkeni gruplar haline getirerek yönetilmesini kolaylaştıran en kullanışlı yöntemdir. Burada sekiz eleman içeren integer değerler için bir array oluşturuyoruz ve buna ledPins adını veriyoruz.

**digitalWrite(ledPins[0], HIGH);**



Array içerisindeki bir elemana bulundukları adres yardımı ile ulaşırınsız. İlk elemanın adresi 0, ikinci elemanın adresi 1, vs. Bir elemana ulaşmak için “ledPins[x]” komutunu kullanarak x yerine o elamanın adresini yazarsınız. Burada dijital pin 2'yi HIGH yapıyoruz.

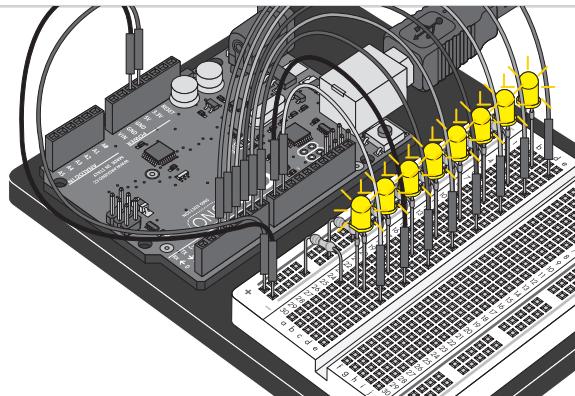
**index = random(8);**



Bilgisayarlar çalışma sırasında aynı işlemleri gerçekleştirir fakat bazen bir şeyin rastgele olmasını istersiniz, örneğin bir zar atma işleminde. Random() fonksiyonu bunu yapmak için en iyi yoldur. Daha fazla bilgi için adresi ziyaret ediniz.  
<http://arduino.cc/en/Reference/Random>

## Ne göreceğiz?

Tek LED yerine bütün LED'lerin yanıp söndüğünü görmeniz gerek. Eğer sorun varsa devreyi doğru şekilde kurduğunuzdan emin olduktan sonra kodunuzu kontrol edin ve arıza tespit kısmını inceleyin.



## Sorun Giderme:

### Bazı LED'ler Işık Vermeye Bilir

LED'inizi ters takmış olabilirsiniz, sıkça karşılaşılan bir sorundur. Çalışmayan LEDinizin doğru bağlandığından emin olun.

### Sıralama Düzeni

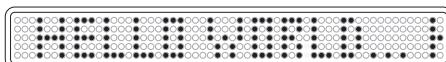
Sekiz adet bağlantı ile çalışığınız için karışıklıklar olması gayet doğal. İlk LED'i pin 2'ye yerleştirin ve diğer LED'leri devam edecek şekilde yerleştirin ve tekrar kontrol edin.

### Yeniden Başlayın

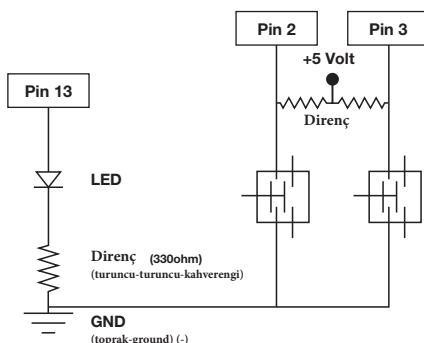
Farkında olmadan bağlantıyı yanlış yere kurabilirsiniz. Genelde her şeyi çıkarıp tekrar yerleştirmek, nerede hata yaptığınızı aramaktan daha kolaydır.

## Gerçek Hayatta Uygulamaları:

Kayan yazı ekranları genellikle önemli bilgilerin kısa parçalarının yayınlanması şeklinde kullanılır. Bu ekranlar çok sayıda LED kullanılarak üretilir.



## Buton Kullanımı



Şu an kadar çıktı (output) üzerine yoğunlaşmıştık. Bundan sonraki projelerimize girdiler (inputs) ile devam ediyoruz. Bu devrede çok yaygın olan bir girdiye bakacağınız push button (buton). Bir butonun Arduino üzerindeki çalışma şekli söyledir; ne zaman ki butona bastınız, voltaj LOW seviyesine geçer. Arduino bunu okur ve buna göre davranışır. Bu devrede, bir adet pull-up direnç görüyoruz. Bu direnç temiz bir voltaj olmasını ve butondan gelecek yanlış okumaların önüne geçilmesini sağlar.

Elemanları:



x2



x1

10KΩ  
Direnç

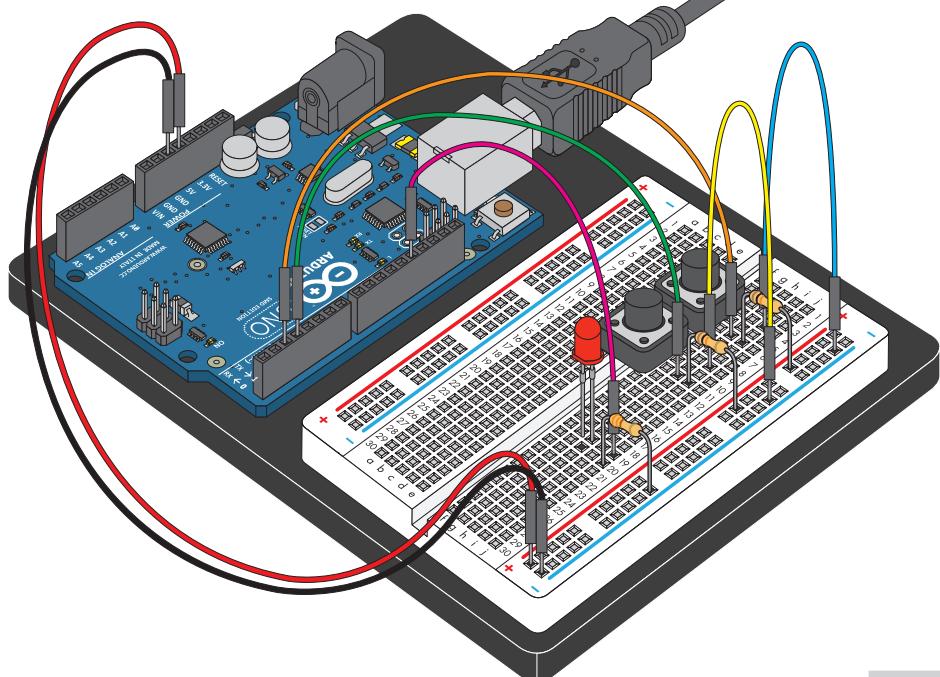
x2

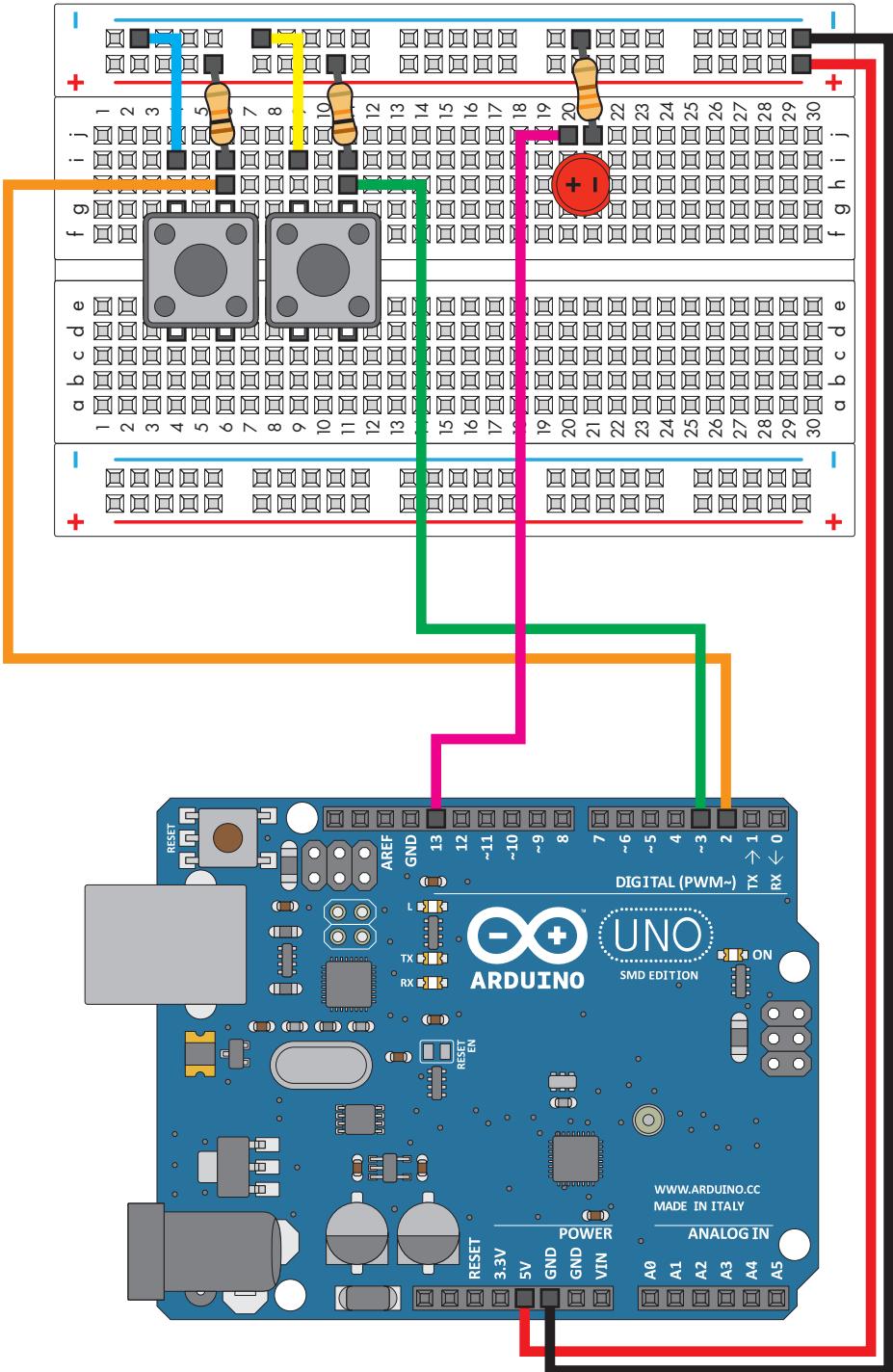
330Ω  
Direnç

x1



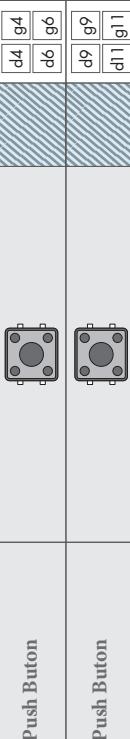
x7





Devre 5: Buton Kullanımı

### Elemanlar:



### Gerçek Görünüm:



Push Buton

Push Buton



LED (5mm)

10KΩ Direnç

10KΩ Direnç

330Ω Direnç

Atlama Kablosu

Atlama Kablosu

Atlama Kablosu

Atlama Kablosu

Atlama Kablosu



Arduino'yu kullanışlı kılan seylerden biri de, girdinizdeki bilgilere dayanan karmaşık eylemleri gerçekleştirebilir. Örneğin havanın soğuk olduğuunda istenilen çalıştırıcının açılabileceği, sıcak olduğunda vanitatorü açabilen, bitkilerin kurunmaya başladığında onları sulayabileceğiniz. Bu eylemleri yerine getirebilmesi için, Arduino "if" ile kurdığınız karmaşık yapıları mantıklı işlemleri gerçekleştirteğiniz.

<b>==</b>	<b>Eşitlik</b>	A == B doğru ise A ve Baynudur.
<b>!=</b>	<b>Farklılık</b>	A != B doğru ise A ve B aynı değildir.
<b>&amp;&amp;</b>	<b>Ve</b>	A && B doğru ise A ve B'nin ikisi de doğrudur.
<b>  </b>	<b>Veya</b>	A    B doğru ise A veya B doğrudur.
<b>!</b>	<b>Değil</b>	!A doğru ise A yanlış. Yanlış ise A doğrudur.

Karmaşık bir if() ifadesi oluşturmak için başka fonksiyonları kombinleyebilirsiniz.

Örnekler:

```
if ((mode == heat) && ((temperature < threshold) || (override == true)))
{
    digitalWrite(HEATER, HIGH);
}
```

Eğer istenilen modundaysanız ve sıcaklığı düşükse bu kod istenilen çalıştırıcıyı etkinleştirebilcek şekilde kodlayabilir ve dış çevreyi kontrol altına alabilirsiniz.

# 5



Arduino Kodu:

Kod Notları:



`pinMode(button2Pin, INPUT);`



Dijital pinler çıktılar gibi girdi olarak da kullanılabilir. Fakat bu işlemi yapmadan önce, Arduino'ya kullandığınız yolu söylemeniz gereklidir.

`button1State = digitalRead(button1Pin);`



Dijital bir girdiyi okumak için `digitalRead()` fonksiyonunu kullanırız. Eğer pinde 5V varsa HIGH, 0V ise LOW olacaktır.

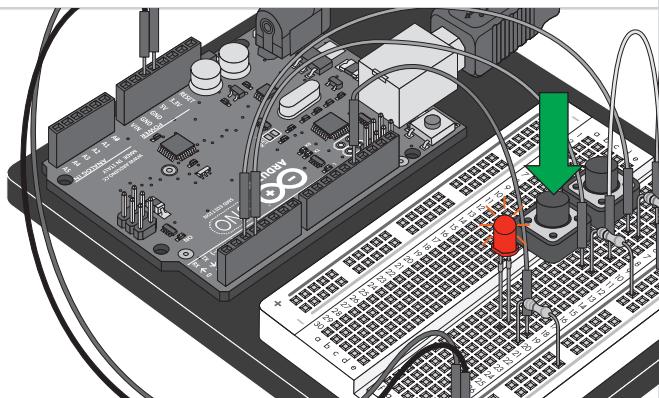
`if (button1State == LOW)`



Çünkü butonumuzu GND'ye bağlıyoruz ve butona bastığınızda LOW olarak okunuyor. Buttonun basılmış durumda olup olmadığı görmek için ("==") operatörünü kullanıyoruz.

## Ne göreceğiz?

Butona bastığınızda LED'in yanıp söndüğünü görebilirsiniz. Eğer sorun varsa devreyi doğru şekilde kurduğunuzdan emin olduktan sonra kodunuza kontrol edin ve arıza tespit kısmını inceleyin.



## Sorun Giderme:

### Işık yanmıyor

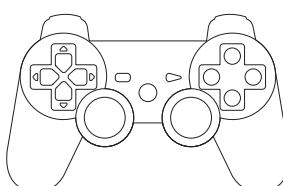
Butonuzum kare olduğu için yanlış yerleştirilmiş olabilir. 90 derece çevirin ve çalışıyor mu tekrar kontrol edin.

### Işık sönmüyor

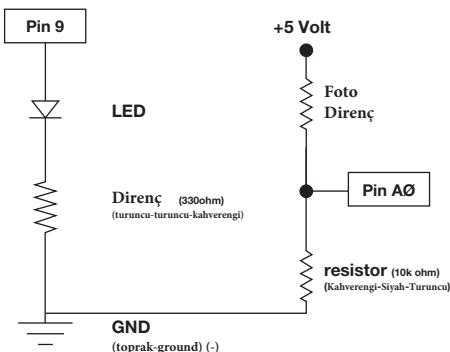
Sıklıkla yaptığımız bir hatadır. Işığı kapatırken LED bağlantınızı pin 13' den pin 9' a getirmeyi unutmayın.

## Gerçek Hayatta Uygulamaları:

Butonlar bir çok oyun konsolunda kontrolü sağlamak için kullanılır.

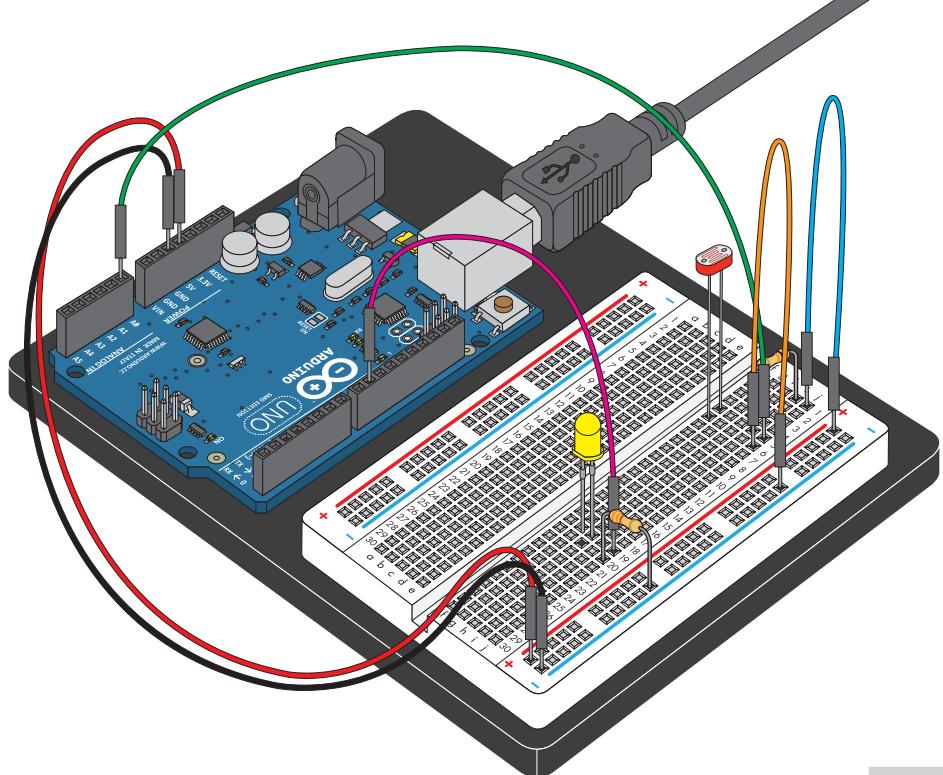


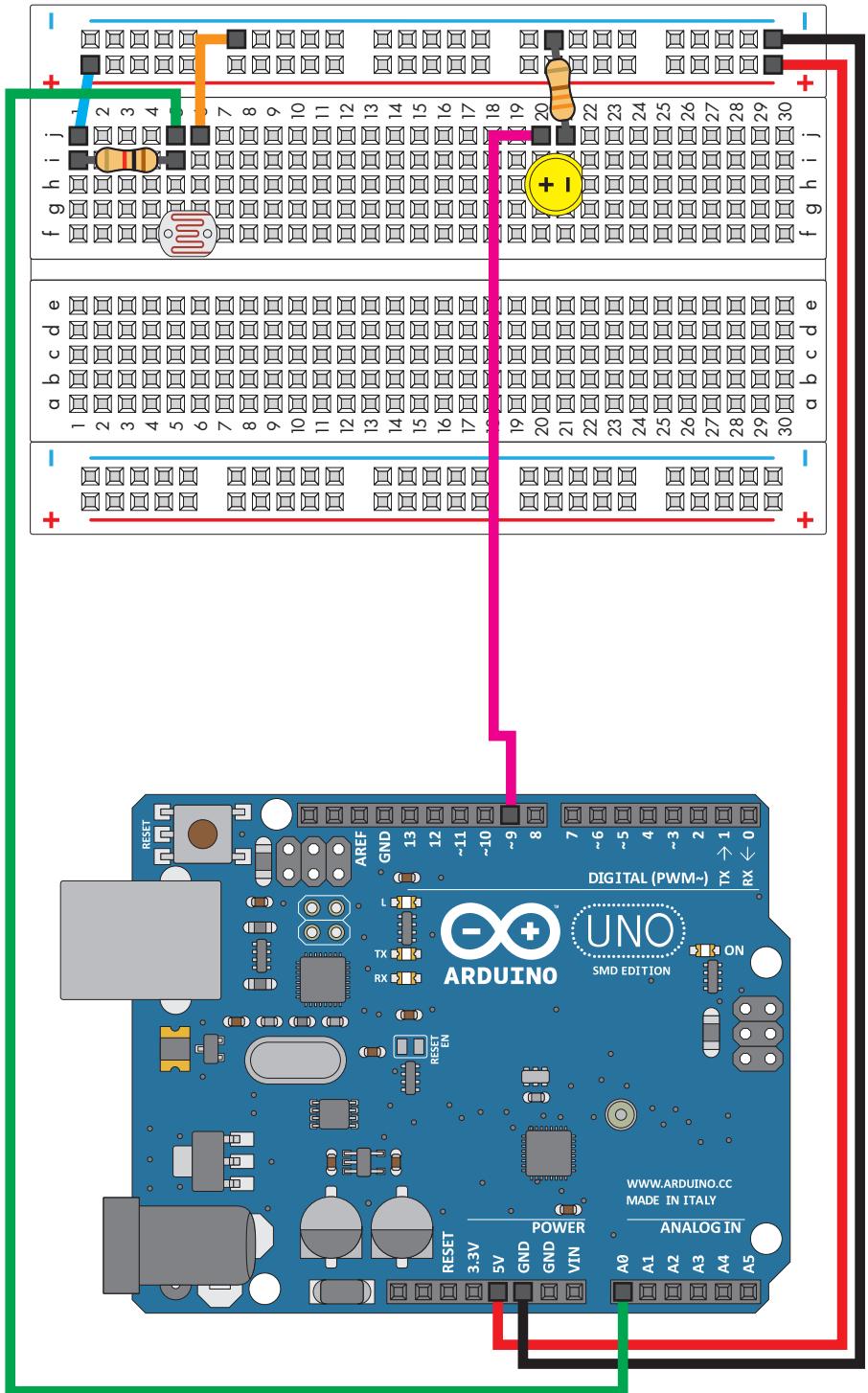
## Foto Direnç



Potansiyometreyi önceki projelerimizde direnci değiştirmek için kullanmıştık. Bu devrede sensöre ulaşan ışık miktarın göre direnci değişen foto dirençleri kullanacağınız. Arduino direnci direkt olarak değerlendiremediği için foto direncimizi kullanmak için voltaj bölücü kullanacağız. Voltaj bölücü fazla ışık altında yüksek voltaj çıkışı, fazla ışık almadığı zamanda da düşük voltaj çıkışı verecek.

Elementler:



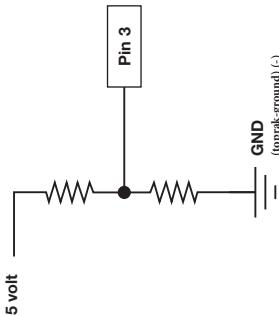


Devre 6: Foto Direç

Direnç Ölçüm Sensörleri:	
Elemenlar:	Gerçek Görünüm:
Foto Direnç	
LED (5mm)	
330Ω Direnç (sensör)	
10KΩ Direnç	
Atlama Kablosu	
Atlama Kablosu	
Atlama Kablosu	
Atlama Kablosu	
Atlama Kablosu	
Atlama Kablosu	

Gördüğünüz bütün bu sensörlerin (potansiyometreler, fotodirençler vs.) hepsi farklı şekillerde görünen dirençlerdir. Direnç değeri algadıkları şeyin değerine göre değişir. (İşik seviyesi vs.)

Arduino analog giriş(input) pini direnci değil voltaj ölçer. Ama biz Arduino'umuz ile birlikte "voltaj bölücü" olarak kullanılan dirençlər sensörler kullanacağız.



Bir voltaj bölücü ikili adet direncen meydana gelir. Üst direnci 5 volta ve alt direnci topraga(GND) bağlılığınızda, orta kism ılık direnc değerine uygun olan bir volt çıkışı verir. Dirençlerden birinin değeri değiştiğinde (Algılaklı birimin degeri değiştiğinde) direnç degeri de değişecektir ve bu yüzden çıkış voltagı da buna bağlı olarak değişecektir!



Kod Notları:

**lightLevel = map(lightLevel, 0, 1023, 0, 255);**

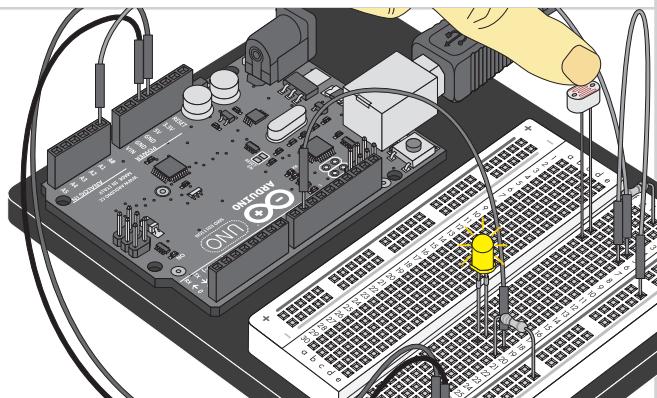
⇒ analogRead() kullanarak okuduğumuz bir analog sinyal, 0-1023 arasında bir değer olacaktır. Fakat analogWrite() kullanarak bir PWM çalıştmak istediğimizde, 0-255 arasında bir değer isteyecektir. Bu durumda map() fonksiyonunu kullanarak geniş alanları daha dar alanlar olacak şekilde sıkıştırabiliriz.

**lightLevel = constrain(lightLevel, 0, 255);**

⇒ Kullandığımız map() fonksiyonu sınırı daraltır fakat biz bunu yanında constrain() komutu kullanarak sayıları bu sınır içerisinde tutacağz. Eğer sayı bu sınırı aşarsa daha büyük bir sayıya dönüşecektir. Ama sınırın içerisindeyse aynı kalacak.

## Ne göreceğiz?

Fotodirencinizin algıladığı ışık miktarına göre LEDinizin daha parlak veya sönük yandığını görebilirsiniz. Eğer çalışmıyorsa devrenizin doğru kurulduğundan emin olun, konudunuzu yeniden yükleyin ve arıza tespit kısmını kontrol edin.



## Sorun Giderme:

### LED Işık Vermiyor

Foto Direnç ve LED in bağlantılarını tekrar kontrol edin.

### Işığı Değişimine Tepki Vermiyor

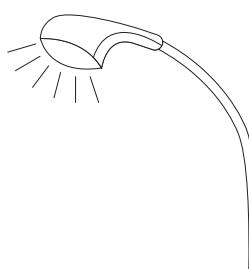
Fotodirenç üzerinde bağlantılar standart değildir, bu yüzden bağlantıları karıştırmış olmanız mümkün. Doğru yerleştirildiğinden emin olmak için iki kez kontrol edin.

### Hala Çalışmıyor

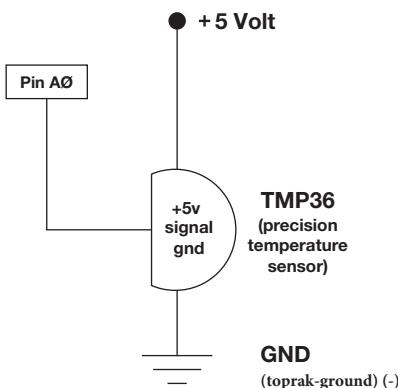
Aydınlatma olarak çok aydınlatık veya karanlık bir odada olabilirsiniz. Işıkları duruma göre açın veya kapatın. Eğer yakınımızda flaş varsa bir de onu deneyin.

## Gerçek Hayatta Uygulamaları:

Sokak lambaları geceleri aydınlatmak için bu tür sensörler kullanır.



## Sıcaklık Sensörü



Sıcaklık sensörü adından da anlaşılacağı gibi ortam sıcaklığını ölçmek için kullanılır. Bu sensörümüzde 3 adet pin bulunuyor. Pozitif, toprak(GND) ve sinyal pinleri. Bu devremizde, sıcaklık sensörünü Arduino ile nasıl entegre edileceğini öğreneceğiz ve Arduino IDE'deki serial monitörde sıcaklık değerini göreceğiz.



Transistör ve Sıcaklık Sensörü birbirlerle çok benzeyen devre elemanlarıdır. Karıştırılmamaya dikkat edilmelidir.

Elemanlar:

Sic. Sensör

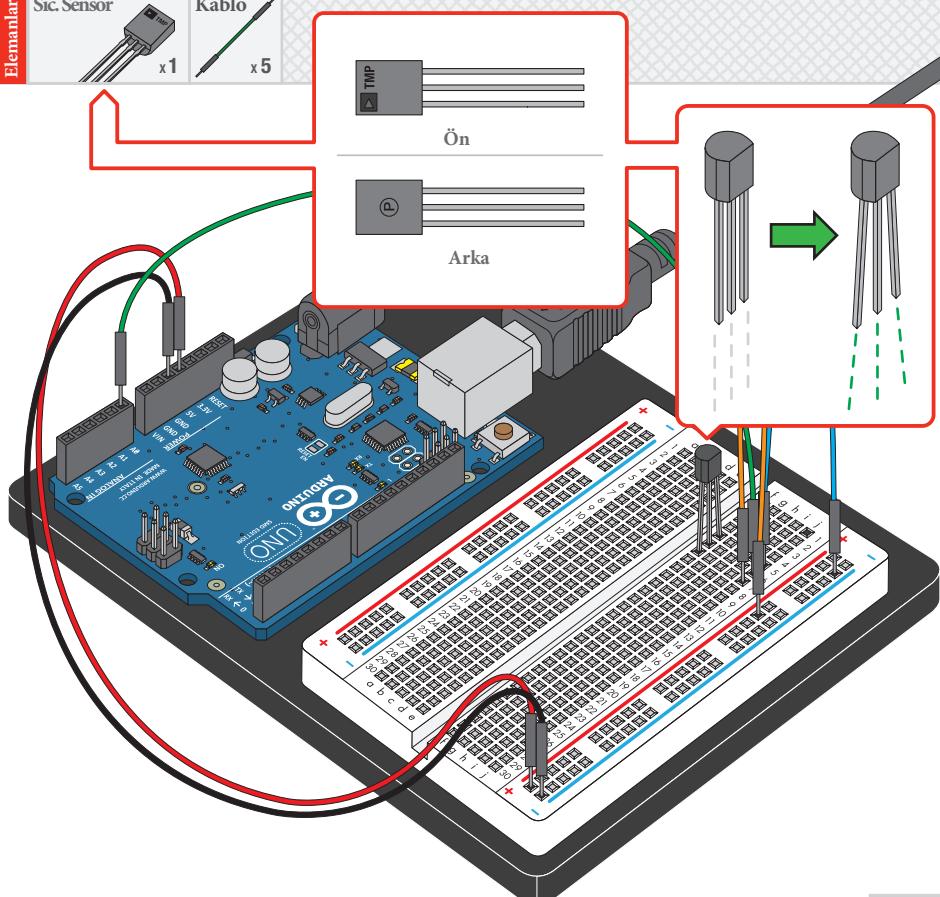


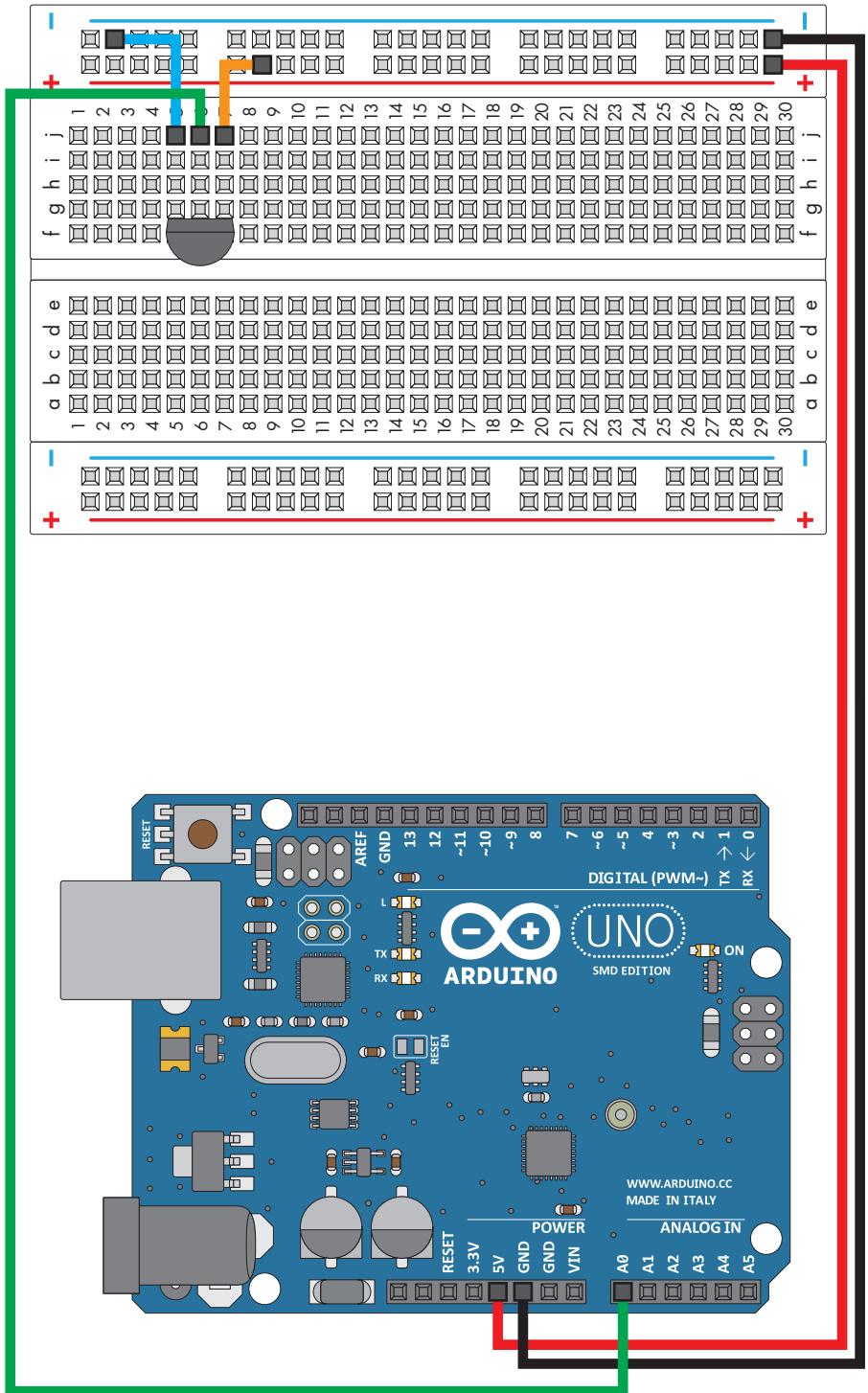
Kablo



x 1

x 5

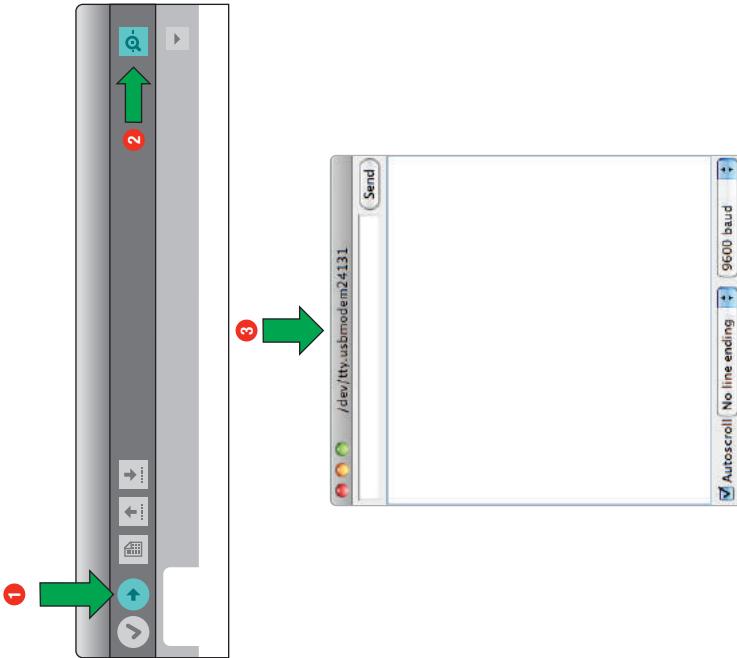




Devre 7: Sıcaklık Sensörü

## Serial Monitörü Açıyoruz

Bu devre Arduino IDE'nin seri monitörünü kullanıyor. Bunu açmak için öncelikle program upload edip kare kutu içerisindeki butona basıyoruz.



Elemanlar:	Gerçek Görünüm:	İşlem Görseli:
Sıcaklık Sensörü		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		



Arduino IDE 'yi Açı // File > Examples > Arduino Kod > Devre # 7

Kod Notları:



**Serial.begin(9600);**



Seri monitörü kullanmadan önce, başlatmak için Serial.begin() komutunu çağrımanız gereklidir. Bağlantı hızı veya "Baud Hizi\*" 9600'dür. İki cihaz birbirileyle bağlantı kurduğunda, ikisi de aynı hızda ayarlanmış olmalıdır.

**Serial.print(degreesC);**



Serial.print() komutu epey zekidir. İçine attığınız hemen hemen her şeyin çıktısını verebilir, buna her türlü değişkenler de dahildir. Baud: Veri iletiminde modülatör çıkışında bir saniyede meydana gelen semboldeğişikliğidir.

*Daha fazla bilgi için > <http://arduino.cc/en/Serial/Print>*

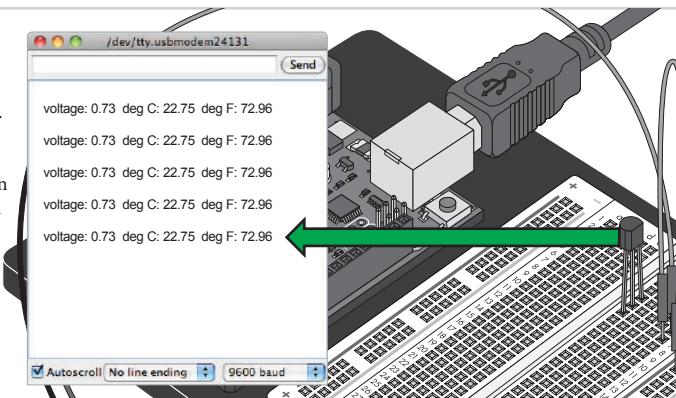
**Serial.println(degreesF);**



Serial.print() her şeyi aynı satır üzerinde yazdırır. Serial.println() diğer satıra geçiş yapar. Bu iki komutu birlikte kullanarak okunması kolay metin ve data çıktıları oluşturabilirsiniz.

## Ne göreceğiz?

Sıcaklık sensörünüzün algıladığı sıcaklık değerini Arduino IDE seri monitöründe okunabiliyor olarak görebiliyor olmanız gereklidir. Eğer çalışmıyorsa devrenizin doğru kurulduğundan emin olun, kodunuzu yeniden yükleyin ve sorun giderme kısmını kontrol edin.



## Sorun Giderme:

### Görünürde Çalışan Bir Şey Yok

Programın çalıştığına dair bir gösterge yok mu? Sonuçları görmek için Arduino IDE seri monitörünü açmanız gereklidir. (Talimatlar önceki sayfada mevcut.)

### Anlamsız Ekran

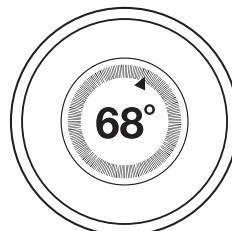
Bu tür şeylerin meydana gelmesinin sebebi seri monitörün beklenenden farklı hızda data almasından kaynaklanıyor. Düzeltmek için ise pull-down kutucuğuna tıklayıp "baud" yazan yeri "9600 baud" olarak düzeltmeniz gereklidir.

### Sıcaklık Değeri Değişmiyor

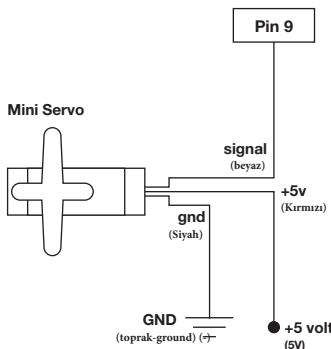
Sensörünüzü parmağınızın arasına sıkıştırarak ısısını yükseltin veya buz torbası kullanarak soğutmayı deneyin.

## Gerçek Hayatta Uygulamaları:

Klima sistemlerinde sıcaklığı takip etmek ve buna göre ayarları yapılandırmak için sıcaklık sensörleri kullanılır.

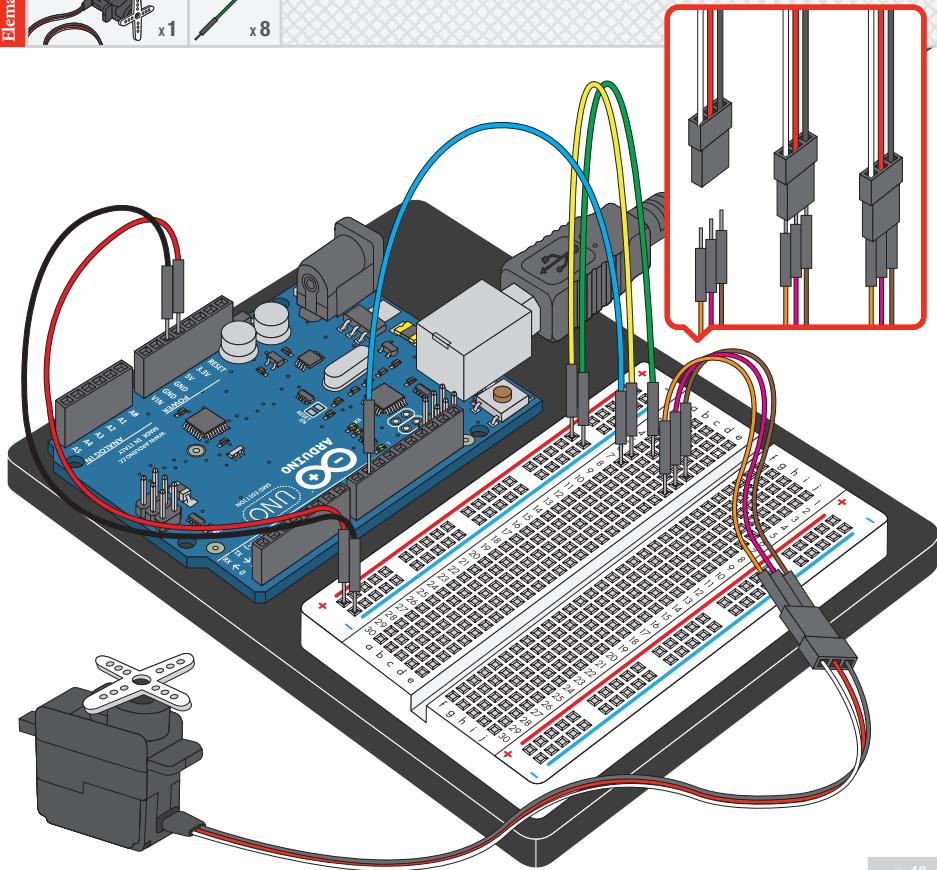


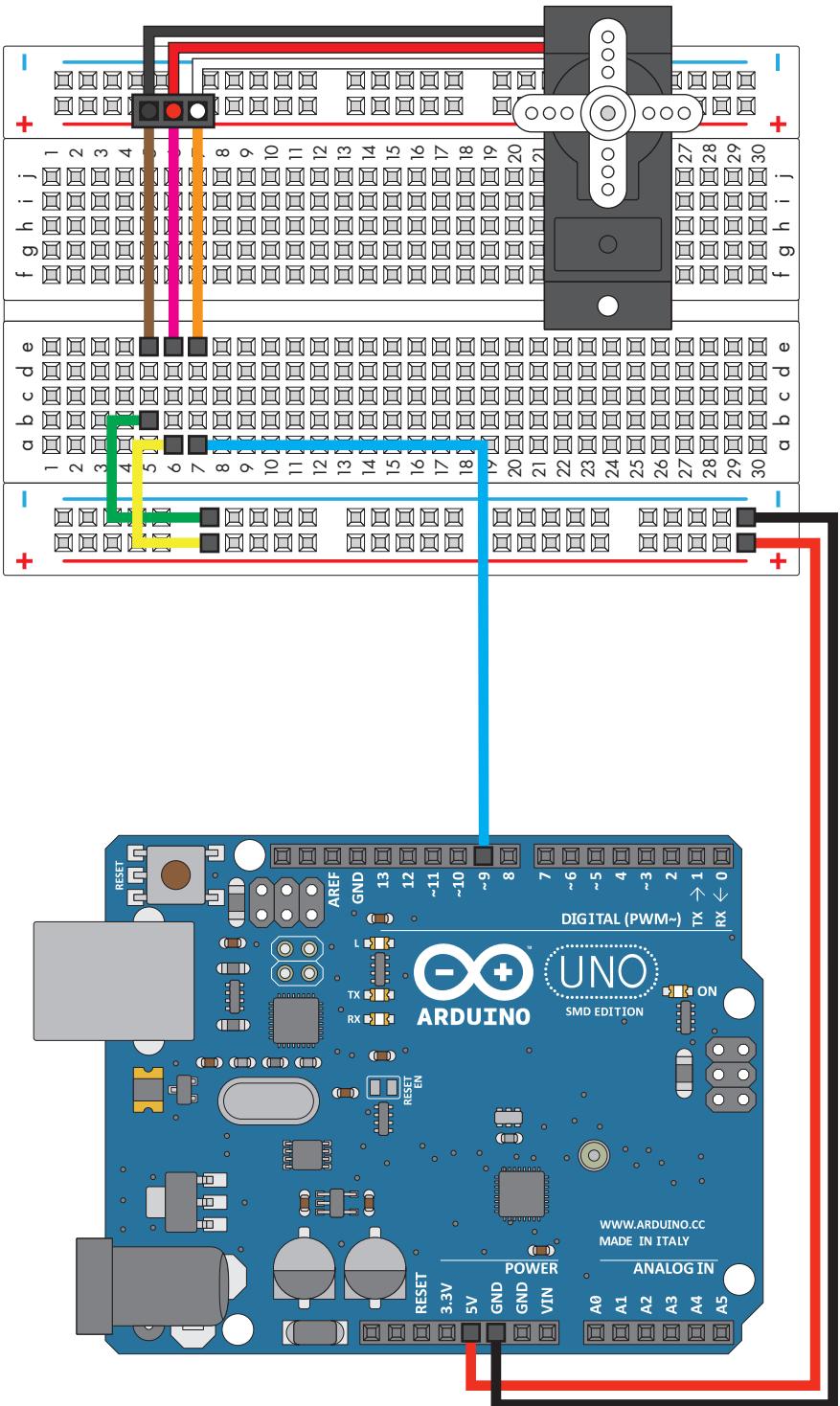
## Servo Motor



Servo motorlar gömülü elektronik uygulamalar için son derece idealdir çünkü dönən normal motorların aksine istenilen herhangi bir yönde dönebilir. Servoya ulaşan voltaj değişimine göre servoyu belirli bir pozisyonaya sokabilirsiniz. Örneğin 1.5 milisaniyelik bir değişim servoyu 90 derece hareket ettirebilir. Bu devrede, servoyu kontrol edip yönlendirmek için PWM kullanmayı öğreneceğiz.

## Elemanlar:





Devre 8: Servo Motor

## Kütüphaneler Kullanarak Ufkunuzu Genişletin

Elemanlar:	Gerçek Görünüm:	Diagram:	Pin Numarası:	Altıgen:
Servo Motor			e5   e6   e7	
Atlama Kablosu			e5	●
Atlama Kablosu			e6	●
			e7	○
Atlama Kablosu			Pin 9	a7
Atlama Kablosu			b5	-
Atlama Kablosu			d6	+
Atlama Kablosu			5V	+
Atlama Kablosu			GND	-
Atlama Kablosu				

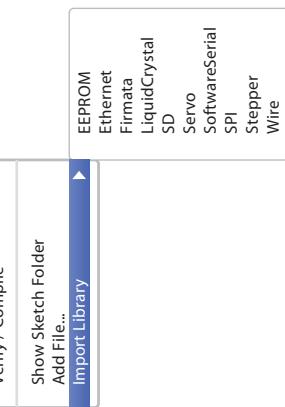
Arduino sağladığı kullanımış dâhil komutlarda; basit giriş(input) ve çıkış(output) işlemleri yapmanızı, mantık kullananak karar vermenizi, matematik problemleri çözmenizi sağlar. Arduino'nun asıl gücü ise bu platformu kullanan devasa toplulukların yaptıkları çalışmalarını paylaştırmaya istegidir. Arduino bir çok kullanışlı kütüphaneye sahiptir. Bu örnekte kulandığımız servo kütüphanesi bunlardan biridir. Standart kütüphaneler ve kütüphaneleri kullanın kılavuzları için siteyi ziyaret edebilirsiniz.

<http://arduino.cc/en/Reference/Libraries>

Herkes kendi kütüphanesini oluşturabilir, eğer yeni bir sensör veya çıktı(output) cihazı kullanacağınız, yeni kütüphane yazabilirsiniz, tabiki şansınızda bağlı olarak daha önce bunu birisi sizin için yapmış olabilir. Birçok Arduino kütüphanelerine ulaşmak için Google veya Arduino Playground'ı kullanabilirsiniz.

<http://arduino.cc/playground>

Arduino'nuza adığınızda yeni bir cihazla kullanacaksanız, kütüphanenizi oluşturan ve bunu bütün dünyaya paylaşın! Kütüphaneyi taskatta (sketch) kullanmak için Sketch > Import Library kısmından seçiniz.





## Kod Notları:



#include &lt;Servo.h&gt;



#include kütüphaneyi (veya başka bir dosya) taslağınızma (sketch) ekleyen özel bir öncülemcidir. Bu komutu kendiniz yazabilir veya hali hazırda yüklü olan bir kütüphaneyi "sketch / import library" menüsünden seçebilirsiniz.

Servo servo1;



servo1.attach(9);

Servo kütüphanesi servoyu kontrol etmenizi sağlayan yeni komut imkani sunar. Arduino'yu servo kontrolüne hazırlarken öncelikle her servo için Servo "object" oluşturmanız gereklidir. (Biz "servo1" olarak adlandırdık.) Ardından servoyu bir dijital pine eşleştirmeniz "attach" gereklidir. (Biz pin 9'u kullanıyoruz.)

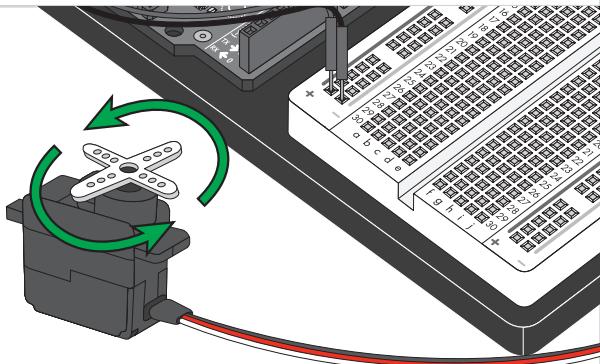
servo1.write(180);



Servo kütüphanesinde write() komutunu kullanarak servonun döneceği derece aralığını (0-180) belirleyebiliriz. Şunu unutmayın, servo hareket için zaman ihtiyaç duyar, ihtiyacımıza göre delay() komutu kullanarak kısa zaman aralıkları tanımlayabilirsiniz.

## Ne göreceğiz?

Servo motorunuzun farklı hızlarda değişik konumlarda dönebildiğini görebilirsiniz. Şayet motorunuz hareket etmiyorsa bağlantılarınıza kontrol ettiğten sonra kodunuzun doğru olduğundan emin olun ve upload edin. Hatanın kaynağını görmek için arıza tespit kısmına göz atın.



## Sorun Giderme:

## Servo Dönmüyor:

Farklı renkli kabloları olmasına rağmen şaşırıcı şekilde motorun ters bağlanmış olması mümkün olabiliyor. Muhtemelen sorunuz bundan kaynaklıdır.

## Çalışmamakta Israr Ediyor:

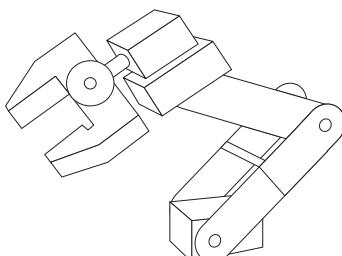
Muhtemelen (kırmızı ve kahverengi kablolar) 5 Volt ve toprak(gnd) bağlantısı yanlış yapılmıştır.

## Oldu Bitti :

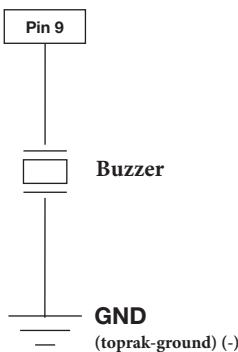
Servonuz çalışıyor fakat kesik hareketler gerçekleştiriyorsa ve Arduino üzerinde yanıp sönen bir ışık var ise muhtemelen enerjiniz yetersizdir. Usb yerine bir adaptör kullanmak bu sorunu çözecektir.

## Gerçek Hayatta Uygulamaları:

Karşılaştığınız robot kollarda servo motorlar kullanılmaktadır.



## Buzzer



Bu devremizde dijital dünya ve analog dünya arasında köprü kuracağız. Bunun için bir buzzer (speaker) kullanacağız. Tek başına çok heyecan verici bir şey olmasa da, bir saniyede yüzlerce kez voltajı açıp kapadığınızda buzzer ses üretecektir. Birden fazla üretilmiş sesi bir araya getirdiğinizde, kendi müziğinizi elde edebilirsiniz! Bu devre ve taslak klasik bir ses üretecektir. Sizi hayal kırıklığına uğratmayacağız.

Elemanlar:

Buzzer

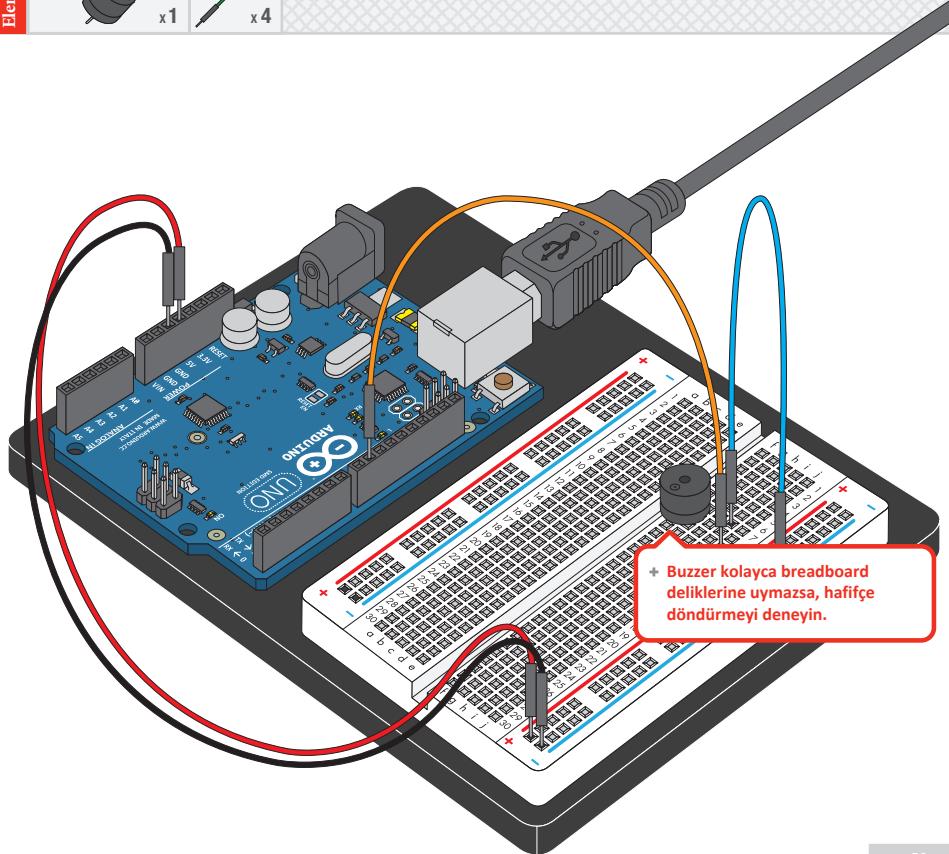


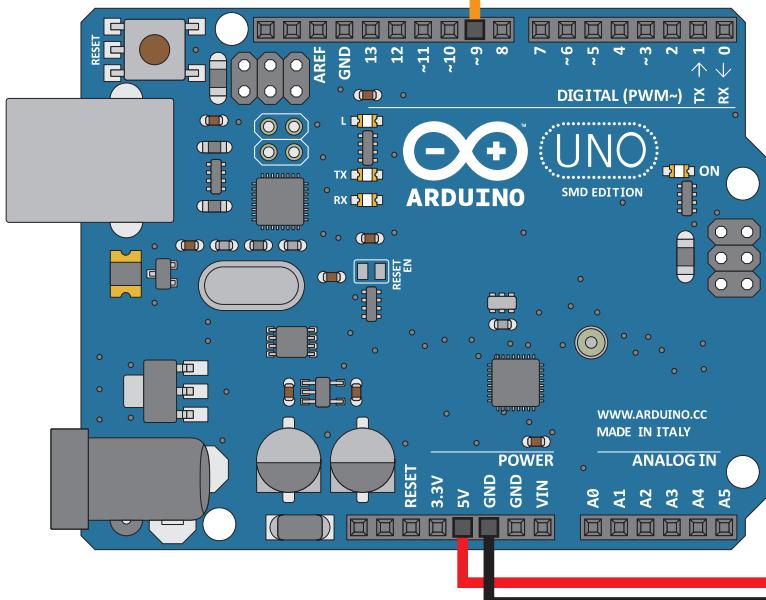
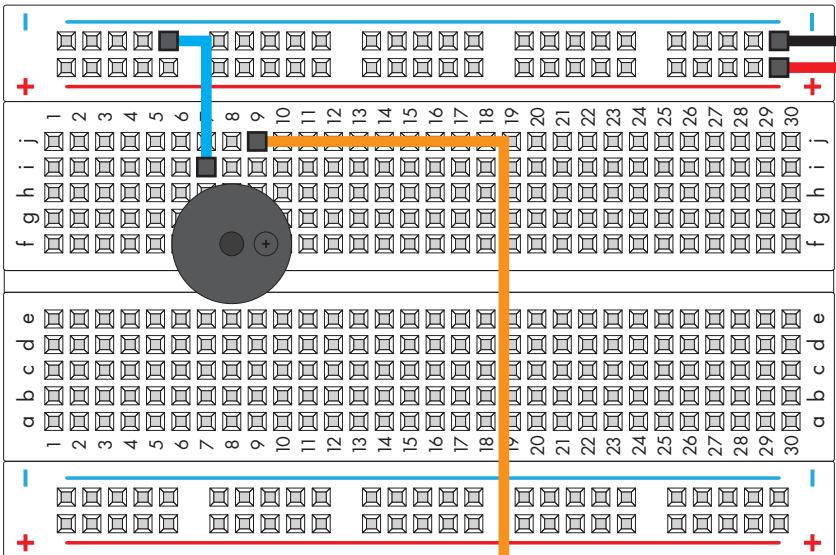
x1

Kablo



x4





Kendi Fonksiyonları Oluşturma													
<p>Elemenlar:</p> <table border="1"> <tr> <td>Buzzer</td> <td>Gerçek Görünüm:</td> <td></td> </tr> <tr> <td>Atlama Kablosu</td> <td></td> <td></td> </tr> <tr> <td>Atlama Kablosu</td> <td></td> <td></td> </tr> <tr> <td>Atlama Kablosu</td> <td></td> <td></td> </tr> </table>	Buzzer	Gerçek Görünüm:		Atlama Kablosu			Atlama Kablosu			Atlama Kablosu			<p>Arduino içerisinde her türlü şey için kullanılabilecek kocaman bir servet barındırıyor. (Bkz: <a href="http://arduino.cc/en/Reference">http://arduino.cc/en/Reference</a>) Bunun yanı sıra kendi fonksiyonlarınız oluşturmak da oldukça basit. Örnek verecek olursak “add” iki sayıyi bir toplayıp size sonucu verir.</p> <pre> int add(int parameter1, int parameter2) {     int x;     x = parameter1 + parameter2;     return(x); } </pre>
Buzzer	Gerçek Görünüm:												
Atlama Kablosu													
Atlama Kablosu													
Atlama Kablosu													

Elemenlar:	Gerçek Görünüm:			
Buzzer				
Atlama Kablosu				
Atlama Kablosu				

Fonksiyonunuz bir değer alabilir. (“parametre”) ve başka bir değere dönüştürbilir. Eğer fonksiyonuna bir parametre atamak isterseniz, fonksiyondan sonra gelen parantez içerişine bu değeri yazabilirsiniz. Eğer parametre vermeyecekseniz parantez içersini boş bırakabilirsiniz. Eğer fonksiyonundan bir değer çekerek seniz değer tipini fonksiyon isminizin önüne yazın. Değer çekmeye hazırlısanız `return()` komutu ekleyin. Eğer değer döndürmeyecekseniz fonksiyon isminizin önüne “void” ekleyin. (`setup()` ve `loop()` fonksiyonlarında göreviniz olduğunuz gibi.) Kendi fonksiyonunuuzu yazdığınızda kodunuzu düzeltmek kolay bir hale getirin.



Kod notları:

**char notes[] = "cdfda ag cdfdg gf";**

Şimdiden kadar sadece sayısal verilerle çalıştık ama Arduino aynı zamanda metinlerle de çalışabilir. Karakterlerin (tek karekterler, harfler, numaralar, semboller) kendilerine ait tipleri mevcuttur. Bu tipe "char" diyoruz. Bir karakter diziniz varsa çift tırnak içerisinde tanımlanabilir. "string" olarak tanımlarsamız tek tırnak içerisinde belirtilmelidir.

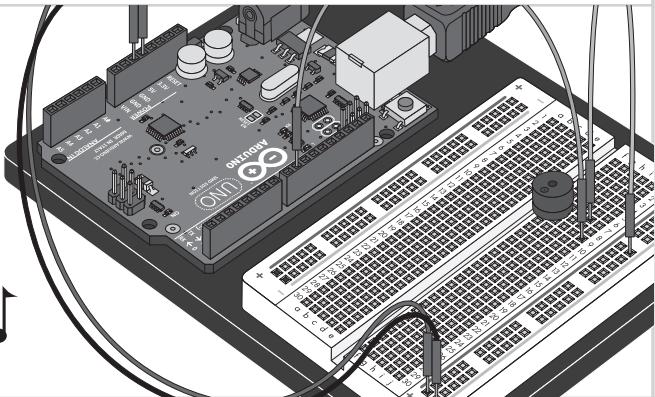
**char names[] = {'c','d','e','f','g','a','b','C';**

Arduino'nun kullanışlı komutlarından biri de **tone()** fonksiyonudur. Bu fonksiyon bir çıkış pinini belirli bir frekansa çalıştırıyor, böylece buzzer ve speakerlar için ideal hale getiyor. Eğer belirli bir süre tanımlayacak olursanız (milisaniye cinsinden) o süre zarfından ses oluşturacak ve ardından suracaktır. Eğer süre belirlememişseniz, sonsuza kadar ses üretebilir. (Tabiki **noTone()** komutu kullanarak bunu sonlandırabilirsiniz.)

**tone(pin, frequency, duration);**

## Ne göreceğiz?

Tabi bir şey duymuyor olmanızda mümkün. Eğer çalışmamışsa bağlantılarınıza kontrol ettikten ve kodunuzu gözen geçirdikten sonra tekrar upload edin ve hata devam ederse sorun giderme kısmına göz atın.



## Sorun Giderme:

### Ses Yok

Breadboard üzerindeki boşlukları karıştırılmış olmanız mümkün. Cihaz yerleşimini tekrar kontrol edin.

### Müzik Çalarken Düşünemiyorum

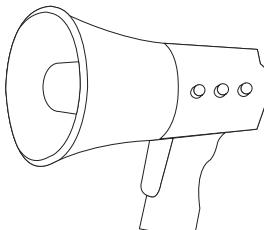
O halde düşünürken cihazınızı söküp, kullanacağınız zaman programı upload edip tekrar monte edin.

### Twinkle Twinkle Little Stars Şarkısından Sıkıldım

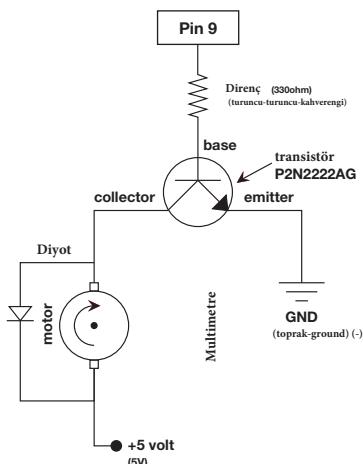
Nasıl yapacağınızı öğrendiğinize göre artık kendi şarkılarınızı upload edebilirsiniz.

## Gerçek Hayatta Uygulamaları:

Modern megafonlarda güçlendirilmiş buzzerler kullanılıyor. Gerçekten fazla gürültülü olmalarına rağmen insanların ilgisini çekmeye etkili.



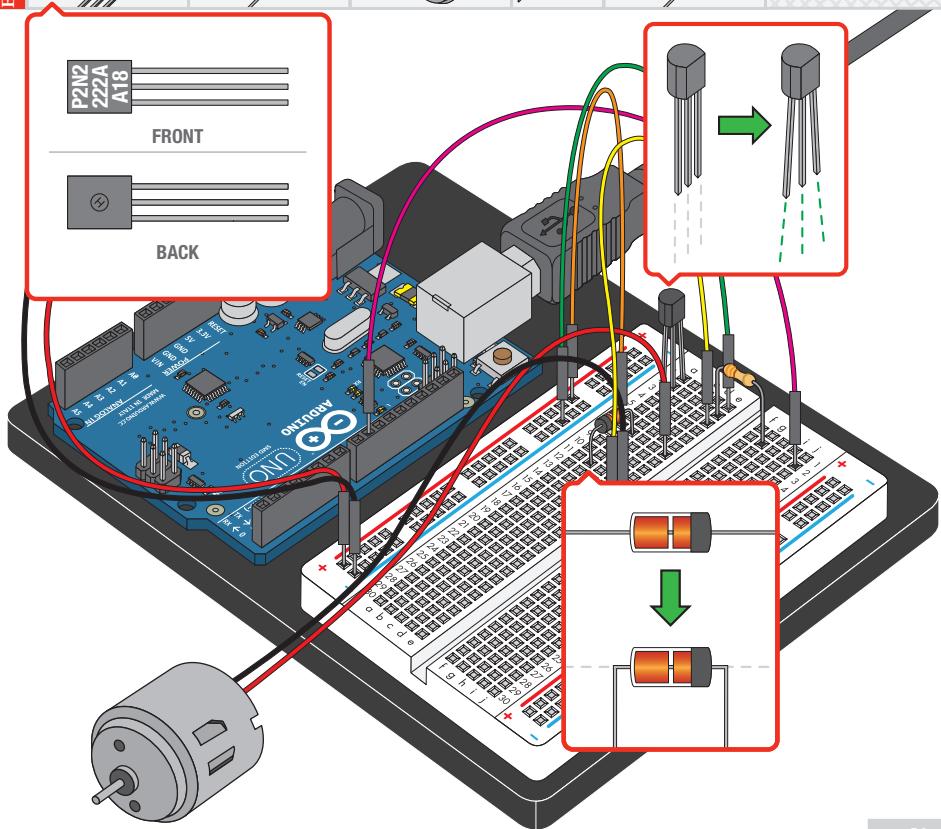
## Motor Döndürmek

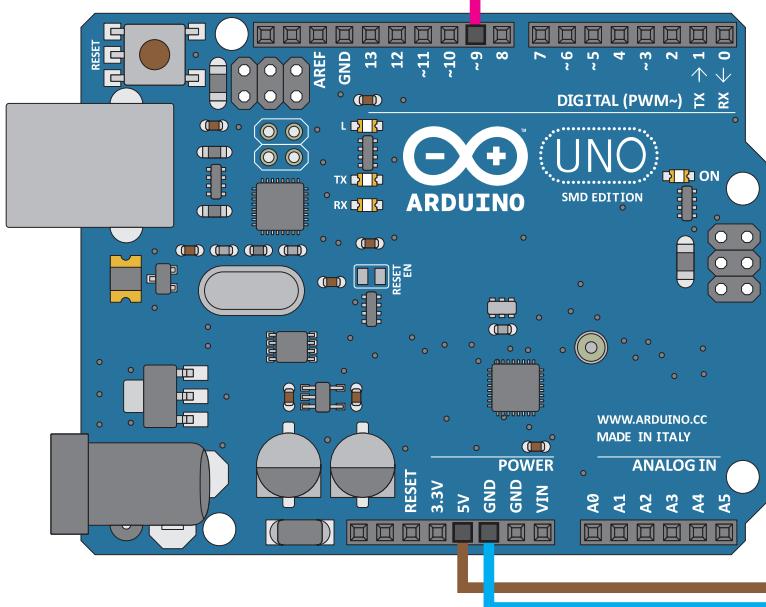
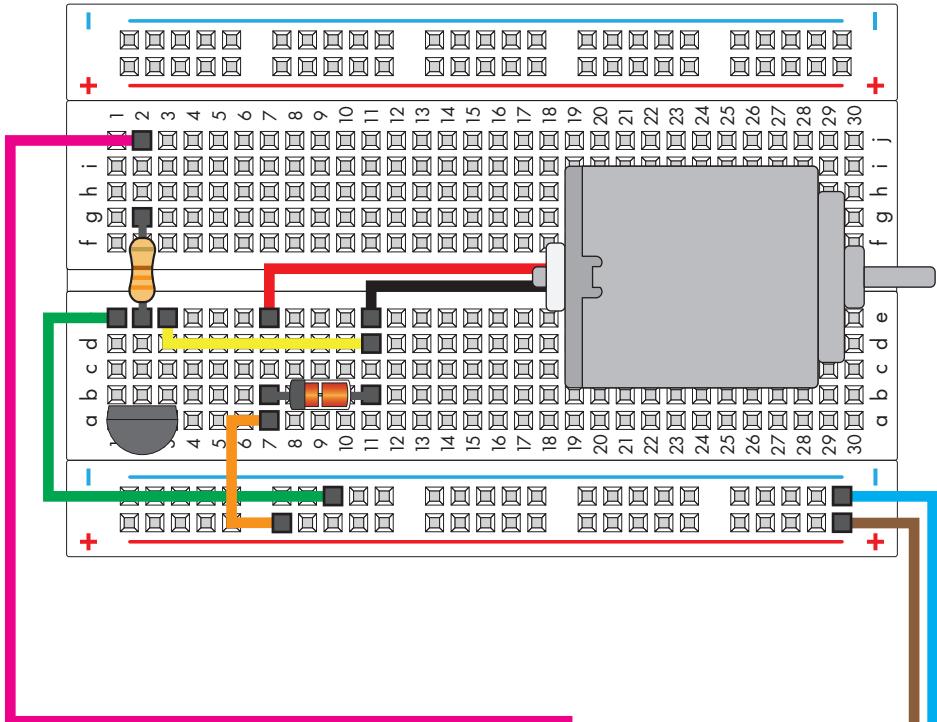


Servo motorlarla oynadığınız zamanlardan öncesini hatırlayın. Şimdi biraz da motorun dönüsünü uğraşacağız. Bunun için transistore ihtiyacımız var çünkü transistorler Arduino'nun yapabildiğinden daha büyük miktarlarda akım dönüştürebiliyor. Transistor kullandığınızda maximum seviyesine bakarak kullanımınız için yeterince uygun olup olmadığını karar verin. Bu devre için 40V ve 200 mA'lık transistor isimizi görür. Oyunca motorumuz için yeter de artar!



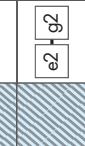
Transistör ve Sıcaklık Sensörü birbirlerle çok benzeyen devre elemanlarıdır. Karıştırılmamaya dikkat edilmelidir.





Devre 10: Motor Döndirmek

## Hepsini bir araya getirirsek:

Elemanlar:	Gerçek Görünüm:	
Transistor P2N2222AG		
Diode 1N4148		
DC Motor		
330Ω Direnç		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		

Muhtemelen bu noktada devreniz için eğlenceli bir fikriniz veya bir sorunu çözmeye yönelik çözüm öneriniz olabilir. Harika! O halde biz de size genel programlama konusunda bir kaç püf noktası önerelim.

Yazdığımız bir çok taslağı (sketch) aşağıdakilerin bir karışımı veya hepsini içerecek:

1. input denemesi yapın.
2. Bazı hesaplanalar yapın ve karar verin.
3. Output alımaya çalışın.
4. Tekrar edin!(Veya eteneyin size kalmış.)

Birçok input sensöri ve output cihazının nasıl çalıştığı hakkında epiy bilgi verdik. (Bir kaç cihazınızın ve sensörünüz daha var tabii) Artık öğrendiklerinizi kendi taslağınızda öğürice kullanabilirsiniz. İste bu “Açık Kaynak” akımının ardından yatan fikrin ta kendisidir. Farklı taslaklärden (sketch) parçaları toplayın ve bunları birleştirip yeni şeyler üretmek fazlasıyla kolay, tek yapmanız gereken iki adet pencere açmak ve birinden diğerine kopyalayıp yapıştmak. Bu yüzden “yi programlama alışkanlığı” fikrinizi aslıyorum. Ayni pin numaralarını kullanın ve taslağınızı fonksiyonla ayırmak, kodunuzun yeniden kullanılabilirliğini koruyacaktır. Örneğin bir kodun ikinci bölümünü için aynı pinı kullanırsanız kolaylıkla ikisinden birinin pinini değiştirebilirsiniz.(Şunu unutmayın ki büyük pinler analogWrite() destegine sahip değil, uyumlu olanlar devre kartlarında işaretli bulunuyor.)

Eğer yardımına ihtiyaç duyarsanız bunun için kullanabileceğiniz internet forumları var. arduino.cc/forum adresindeki Arduino forumunu deneyebilirsiniz. Bunun yanı sıra forum.sparkfun.com adresinden ve forum.arduinoturkiye.com da sizlere yardımcı olacaktır. İki ilerletmeye hazır olduğunuzda daha ileri seviye konular için arduino.cc/en/Tutorial sayfasından Arduino Tutorial sayfasına uguya bilirsiniz. Sonunda, havall bir şeyle ürettığınız valıt, buntu bütüt dünyaya paylaşın ki bütün dünya dahilinizden faydalana bilsin. (Ve bizim bundan haberدار olmamızı sağlayın ki bunu anasayıfımızda paylaşabilelim!)

# 10



[Arduino IDE 'yi Açı](#) // File > Examples > Arduino Kod > Devre # 10

Kod Notları:



**while (Serial.available() > 0)**



Arduino seri portları data gönderimi için olduğu gibi data alımı için kullanılabilir çünkü herhangi bir zaman da data aktarımı olabilir.

Arduino siz bu bilgiyi kullanana kadar depolar ve korur.

The Serial.available() komutu portunuzda ulaşmış fakat taslağınızda (sketch) henüz kullanılmamış olan karakter numaralarını geri getirir. Sıfır, ulaşmış data yok anlamına gelir.

**speed = Serial.parseInt();**

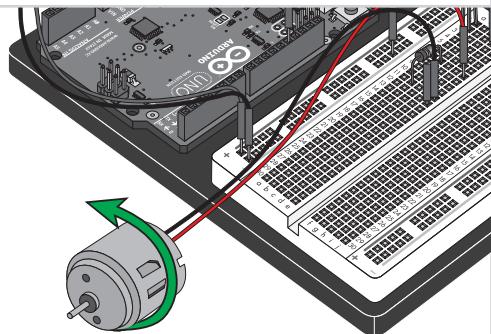


Portunuzda bekleyen hali hazırda datanız mevcut ise, kullanabileceğiz bir kaç yöntem var. Port içeresine sayıları girmeye başladığımızdan beri Serial.parseInt() komutu ile

ayırma ve integer numaraları kendisini oluşturan karakterlere ayırtmak için kullanıyoruz. Eğer portunuza "1" "0" "0" yazarsanız, bu fonksiyon bu numaraları 100 olarak çevirecektir.

## Ne göreceğiz?

Eğer bileşenleri doğru yerleştirmişseniz DC motorunuzun çalışması gereklidir. Eğer çalışmıyorsa arıza tespit kısmını kontrol edin.



## Sorun Giderme:

### Motor Dönmüyor

Kendi transistörünü kullanıyorsanız, data sheeti iki kez kontrol ederek pinout'un P2N2222AG ile uyumlu olup olmadığını tespit edin.

### Çalışmamaya Devam Ediyor

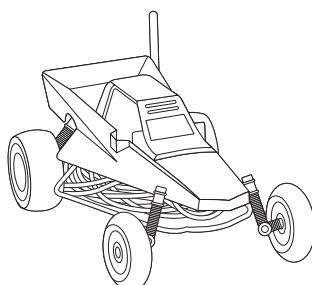
Kendi motorunuzu kullanıyorsanız motorunuzu 5V ile çalıştırmayı deneyin ve daha fazla güç çekip çekmediğini kontrol edin.

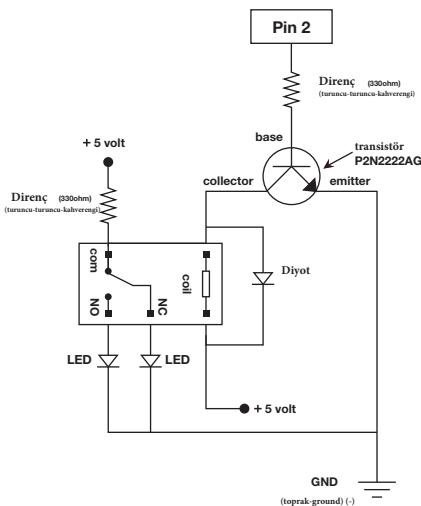
### Çalışmamakta Israrlı

Bazen Arduino bilgisayarla bağlantısını koparabilir. USB girişinizi çıkarıp tekrar takın.

## Gerçek Hayatta Uygulamaları:

Radio Kontrollü RC arabalar, DC motor kullanırlar.





## Röle

Bu devrede, röle kontrolü için Devre#10' da öğrendiklerimizi kullanacağız. Röle basitçe söyleyecek olursak elektriksel olarak kontrol edilen mekanik bir anahtardır. Bu zararsız görünen plastik kutunun içerisinde elektromagnet bulunuyor. Ne zaman ki fazla enerji ile yüklenirse anahtarı açıyor. Bu devrede Arduino'nuza daha güçlü yetenekler kazandırmayı ve bir röleyi profesyonelce kullanmayı öğreneceksiniz.



Röle kapalıken, COM(common) pini NC (Normally Closed) pinine bağlanır. Röle açıkken, COM(common) pini NO (Normally Open) pinine bağlanır.

Elementalar:

Röle



x1

Transistör  
P2N2222AG

x1

Diyot  
1N4148

x1

330Ω  
Direnç

x2

LED

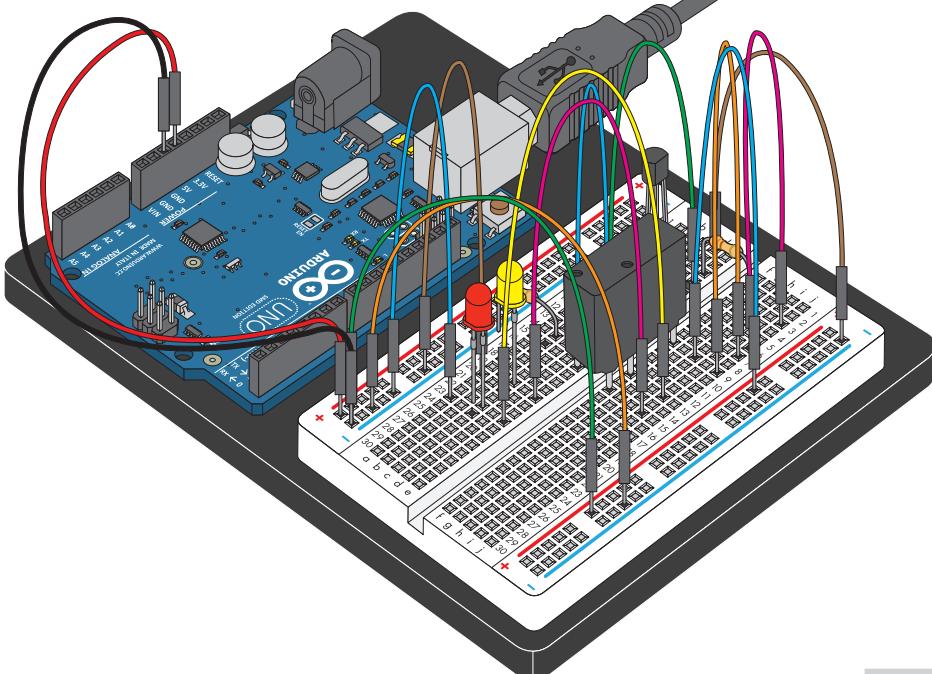


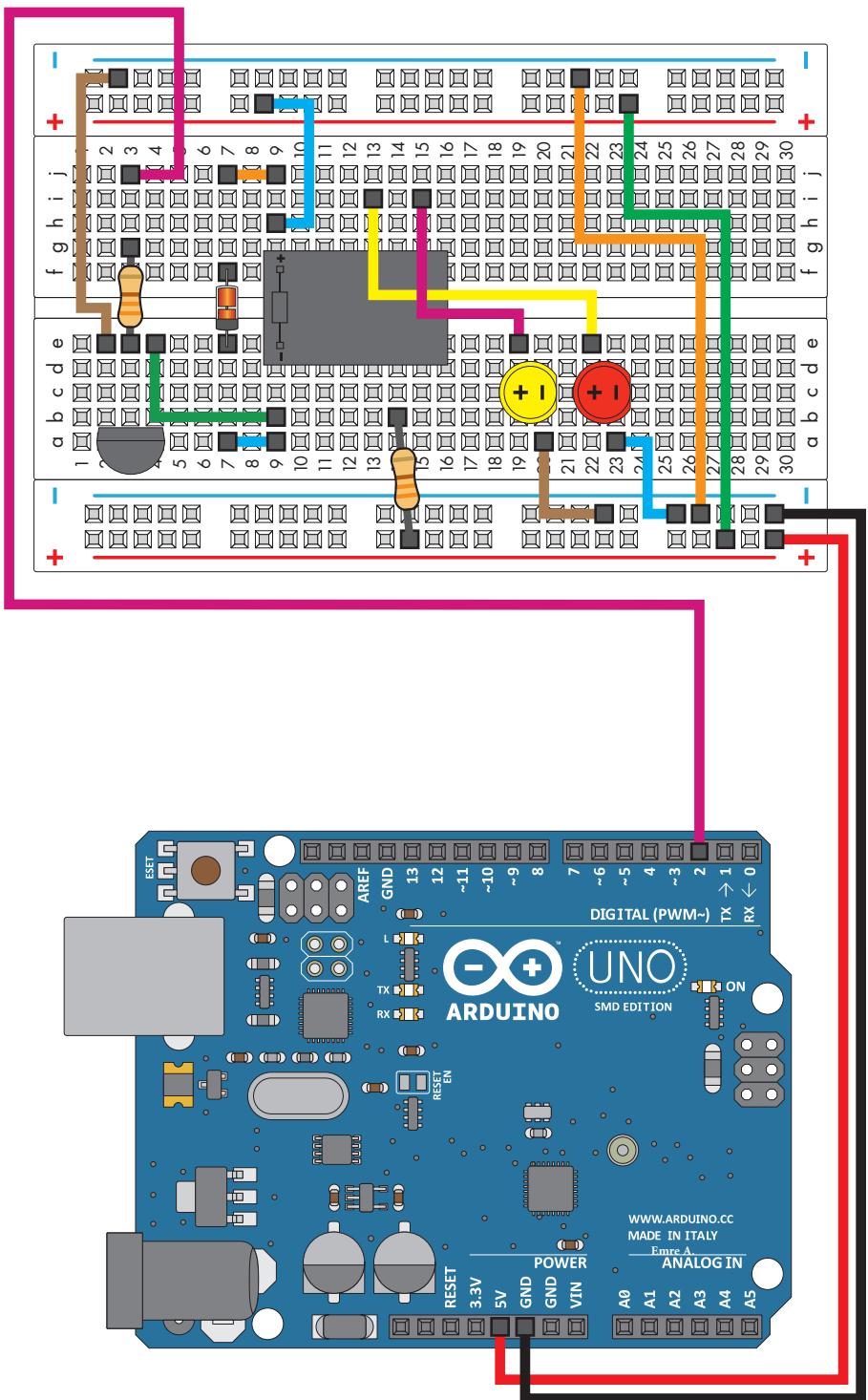
x2

Kablo



x14





Elemanlar:	Gerçek Görünüm:	Elemanlar:	Elemanlar:	Gerçek Görünüm:
Röle		e9   f9 e15   f15	Atama Kablosu	-   e19
Transistör P2N2222A		e2   e3   e4	Atama Kablosu	+   -
LED (5mm)		c19   c20 +   -	Atama Kablosu	d23   -
LED (5mm)		c22   c23 +   -	Atama Kablosu	d20   -
Diyot 1N4148		b7   b11	Atama Kablosu	c7   e9
330Ω Direnç		e3   g3	Atama Kablosu	e4   e9
330Ω Direnç		e2   g2	Atama Kablosu	+   5V
Atama Kablosu		e2   -	Atama Kablosu	GND
Atama Kablosu		Pin 2   i3		
Atama Kablosu		i7   i9		
Atama Kablosu		h9   +		
Atama Kablosu		i13   e22		
Atama Kablosu		i15   e19		



Kod Notları:

**digitalWrite(relayPin, HIGH);**

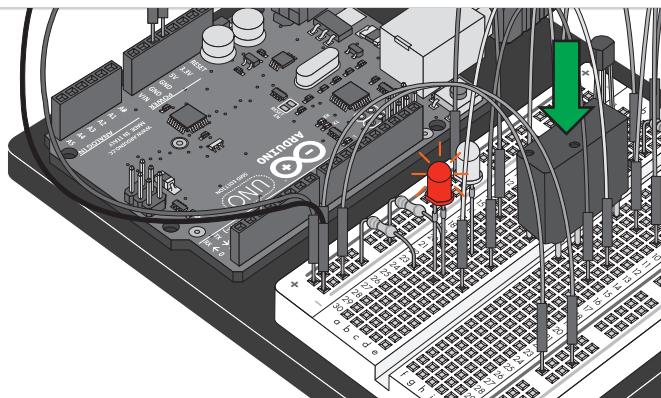
Transistor çalışmaya başladığında rôle bobinine enerji sağlar. Böylece rôle anahtarı(switch) kapanır. Bu kapanma ile rôlenin COM pinı NO(Normally Open) pinine bağlanmış olur. Bağlantı sağlandıktan sonra pinler çalışmaya başlayacaktır. (Biz çalıştığını görmeniz için LED kullandık ama başka bir şey de kullanılabilir tabii ki.)

**digitalWrite(relayPin, LOW);**

Rôle NC denen ilave bir bağlantı sahiptir. Rôle kapalı olduğunda NC pinı COM pin ile bağlantı kurar. Rôle'nin açık veya kapalı olmasına bağlı olarak iki pinden birini kullanabilirsiniz. Aynı zamanda bu iki pinı iki cihazın enerjisini değiştirmek için de kullanabilirsiniz.

## Ne göreceğiz?

Rôle bağlantı klik sesini duymuş olmanız gerek ve ardından 1 saniyelik aralıktaki LED'leri aydınlatmasını kendi aralarında değiştirdiklerini görebilirsiniz. Eğer devrenizi çalışmamışsa devre kurulumunuzu kontrol edin, kodlarınızı tekrar upload edin.



## Sorun Giderme:

### LED Işık Vermiyor

LEDi doğru taktığınızdan emin olun. Uzun olan uç pozitif uçtur.

### Klik Sesи Gelmiyor

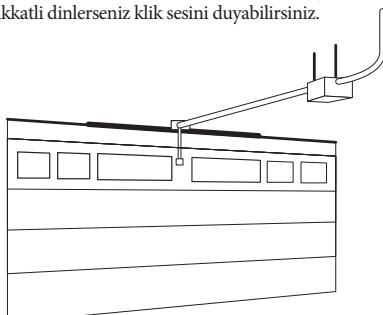
Transistor veya bobin çalışmıyorsa transistörün doğru şekilde takılı olduğundan emin olun.

### Çalışmamakta Israrlı

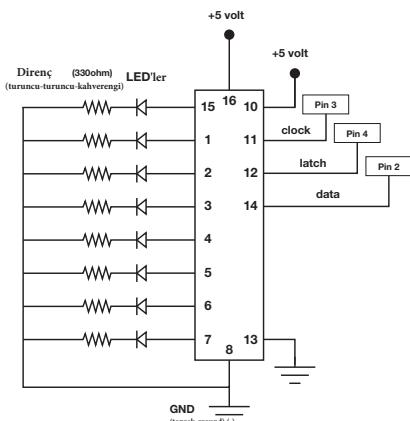
Rôle breadboard ile kullanılmak için değil lehimlenmek için dizayn edilmiş olabilir. Breadboard deliklerine tam oturması için gerektiği kadar ittirmeyi deneyin.

## Gerçek Hayatta Uygulamaları:

Garaj kapılarını açmak için rôle kullanılır. Eğer dikkatli dinlerseniz klik sesini duyabilirsiniz.

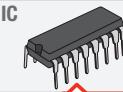


## Shift Register



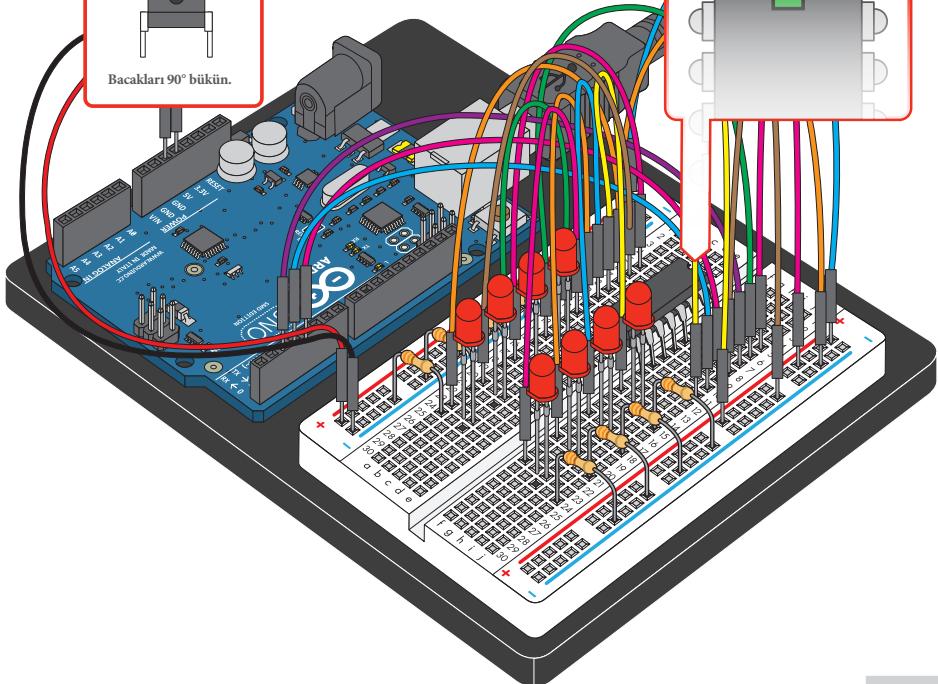
Şimdi de entegre devrelere adım atıyoruz. Bu devremizde shift register hakkında her şeyi öğreneceksiniz. Shift register Arduinonuzu ilaveten 8 output verir ve Arduino'nun sadece 3 pini kullanılır. Bu devrede 8 adet LED'i kontrol etmek için shift register kullanacağız.

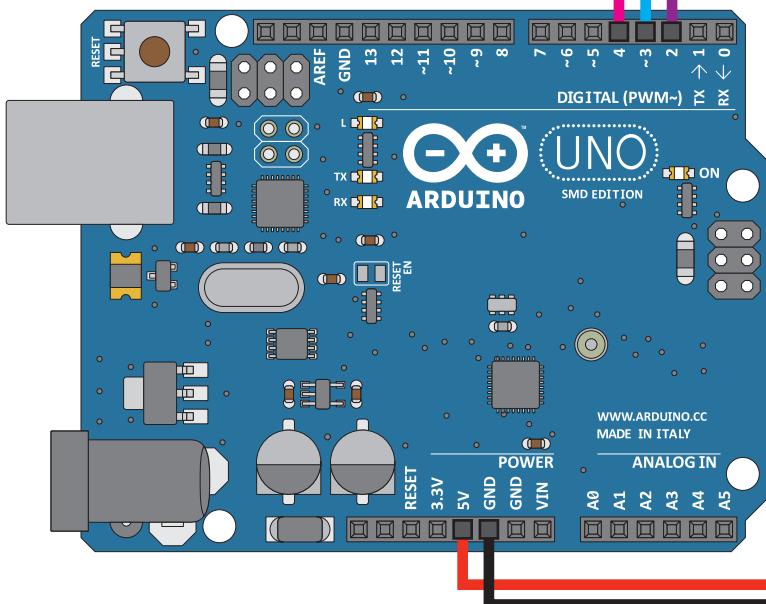
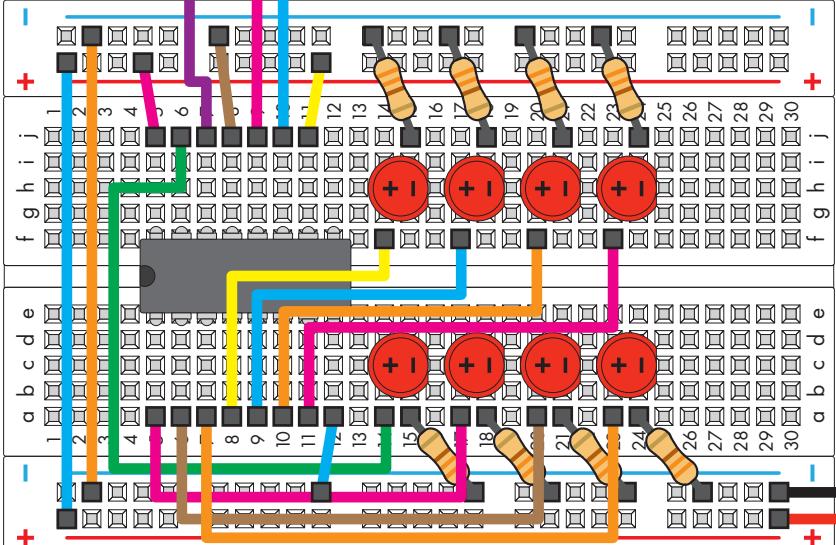
Elemanlar:

330Ω  
DirençKablo  
x19

$Q_0$	1	16	$V_{cc}$
$Q_c$	2	15	$Q_A$
$Q_b$	3	14	SER
$Q_e$	4	13	$\bar{OE}$
$Q_f$	5	12	RCLK
$Q_g$	6	11	SRCLK
$Q_h$	7	10	SRCLR
GND	8	9	$Q_H$

Breadboard üzerinde "e5" ve "f5" arasında çentigi hizalayın.





Devre 14: Shift Register

Elemanlar:	Gerçek Görünüm:	Elemanlar:	Gerçek Görünüm:
IC			
LED (5mm)		e5 e6 e7 e8 [5] [6] [7] [8] e9 e10 e11 e12 [9] [10] [11] [12]	Atlama Kablosu
LED (5mm)		c14 c15 + -	Atlama Kablosu
LED (5mm)		c17 c18 + -	Atlama Kablosu
LED (5mm)		c20 c21 + -	Atlama Kablosu
LED (5mm)		c23 c24 + -	Atlama Kablosu
LED (5mm)		c24 c25 + -	Atlama Kablosu
LED (5mm)		h4 h5 + -	Atlama Kablosu
LED (5mm)		h7 h8 + -	Atlama Kablosu
LED (5mm)		i20 i21 + -	Atlama Kablosu
LED (5mm)		i23 i24 + -	Atlama Kablosu
330Ω Direnç		- q15	Atlama Kablosu
330Ω Direnç		- q18	Atlama Kablosu
330Ω Direnç		- q21	Atlama Kablosu
330Ω Direnç		- q24	Atlama Kablosu
330Ω Direnç		i15 -	Atlama Kablosu
330Ω Direnç		i18 -	Atlama Kablosu
330Ω Direnç		[2] -	Atlama Kablosu
330Ω Direnç		[24] -	Atlama Kablosu
Atlama Kablosu		[+] +	Atlama Kablosu
		5V	
		GND	



Kod Notları:



SPI(Serial Peripheral Interface) arayüzünü kullanmak için shift register (ve birçok farklı parça) ile iletişim kuracaksınız. MSBFIRST parametresi bireysel bitlere gönderilen komutları belirler, bu durumda MSBFIRST gönderiyoruz.

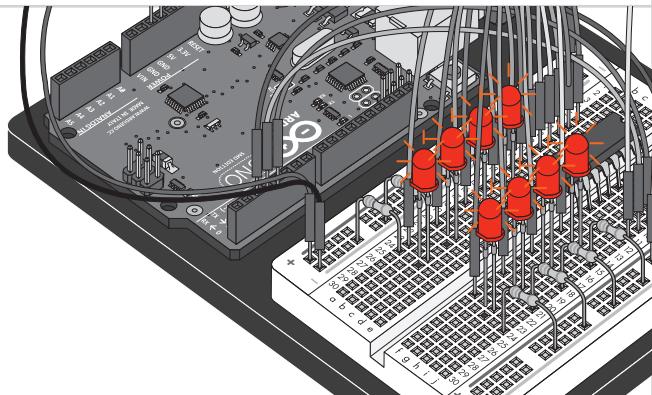
**shiftOut(datapin, clockpin, MSBFIRST, data);**

Bitler bilgisayar hafızasının en küçük yapı taşılardır, her bit 1 veya 0 depolayabilir. Daha büyük sayılar bitlerden oluşan dizilerle depolanır. Bazen bu bitler üzerinde oynamaları isteriz. Örneğin 8 biti shift register'a gönderiyoruz ve LED'i açıp kapatması için 1 veya 0 yapmasını istiyoruz. Arduino bitWrite() gibi basit komutlarıyla bu işlemi gerçekleştirebiliyor.

**bitWrite(data,desiredPin,desiredState);**

## Ne göreceğiz?

Devre 4' te olduğu gibi LED 'in yanğını göreniz gereklidir. (Tek fark shift register kullandık.) Eğer LED ışık vermiyorsa bağlantılarınızı kontrol edin ve kodunuzu tekrar upload edin.



## Sorun Giderme:

### LED Patladı

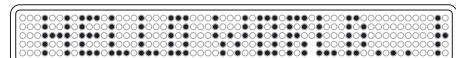
Bu bir çok kez başımıza geldi. Sebebi entegrenin ters takılmasından kaynaklanıyor. Eğer hemen düzeltirseniz hiçbir şeyi bozmadan durumu kurtarabilirsiniz.

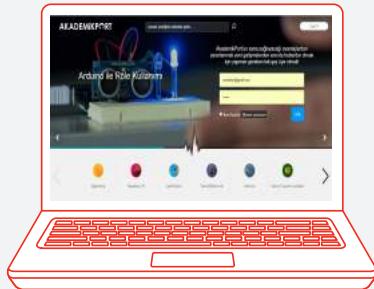
### Çalışmamaya Devam Ediyor

Sürekli aynı şeyi söylüyoruz ama muhtemelen kablolar yanlış takılmıştır.

## Gerçek Hayatta Uygulamaları:

Devre 4 mantığı ile, birden çok LED kullanarak kayan yazı ekranları yapabilirsiniz.





## Sitemizi ziyaret edebilirsiniz!

Mühendisler, Teknik Elemanlar, Maker'lar, Hacker'lar, hobi elektronik seven herkes için açık kaynaklı bir çok yerli ve yabancı kaynaktan yararlanarak sizler "**AkademikPort Arduino Başlangıç Projeleri**" eğitimini hazırladık. Daha çok eğitim ve projelere ulaşmak için [www.akademikport.com](http://www.akademikport.com) adresini ziyaret edebilirsiniz. Bir sonraki eğitimde görüşmek üzere.

**Emre ARSLAN**  
**AkademikPort Kurumsal İlliskiler Koordinatörü**

## Kaynak

***Sparkfun SIK GUIDE***

***Adafruit.com***

***Arduino - Coşkun Taşdemir***

***Instructables.com***

## NOTLAR



**AKADEMiKPORT**

[www.akademikport.com](http://www.akademikport.com)

/akademikport

/akademikport

in /company/akademikport



## Sitemizi ziyaret edebilirsiniz!

Mühendisler, Teknik Elemanlar, Maker'lar, Hacker'lar, hobi elektronik seven herkes için açık kaynaklı bir çok yerli ve yabancı kaynaktan yararlanarak sizler "**AkademikPort Arduino Başlangıç Projeleri**" eğitimini hazırladık. Daha çok eğitim ve projelere ulaşmak için [www.akademikport.com](http://www.akademikport.com) adresini ziyaret edebilirsiniz. Bir sonraki eğitimde görüşmek üzere.

**Emre ARSLAN**  
**AkademikPort Kurumsal İlliskiler Koordinatörü**

## Kaynak

***Sparkfun SIK GUIDE***

***Adafruit.com***

***Arduino - Coşkun Taşdemir***

***Instructables.com***

## NOTLAR



**AKADEMiKPORT**

[www.akademikport.com](http://www.akademikport.com)

/akademikport

/akademikport

in /company/akademikport