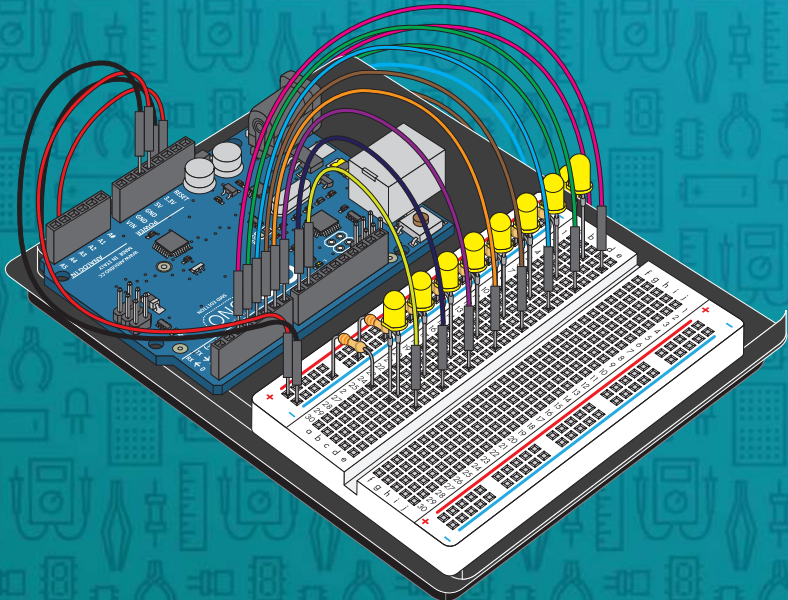


AKADEMİKPORT

Arduino Başlangıç Projeleri



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 kitapçığı 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 İlişkiler Koordinatörü

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 Projelerielektronik kitabı yazar isimleri belirtilip, lisans şartlarına uyularak paylaşılabilir fakat ticari bir faaliyette kullanılamaz.)

To view a copy of this license visit:

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

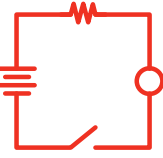
www.akademikport.com



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 Sensörü	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

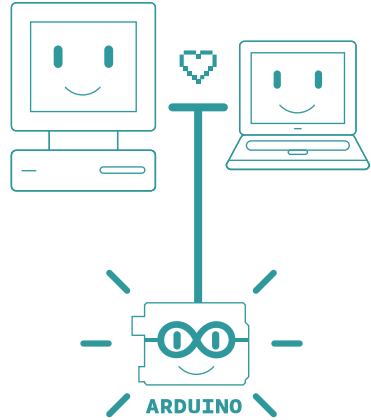
Arduino kolay bir şekilde çevresiyle etkileşime girebilen sistemler tasarlayabileceğiniz, hem acemi hem de ileri düzeydeki elektronik meraklılarına hitap eden, kolayca programlanabilen ve üzerine elektronik devre kurulabilen açık kaynaklı bir geliştirme platformudur.

arduino.cc

Fiziksel Dünya için Bir Bilgisayar

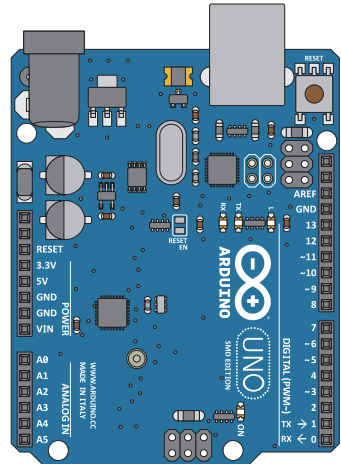
Bu elinizde tuttuğunuz dost canlısı mavi kartı masaüstü bilgisayarlarınızın ve laptoplarınızın çocuğu gibi taşınabilir küçük bir bilgisayar olarak düşünebilirsiniz. Arduino kütüphaneleri ile mikrodenetleyicileri kolaylıkla programlayabilirsiniz. Analog ve dijital girişleri sayesinde analog ve dijital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

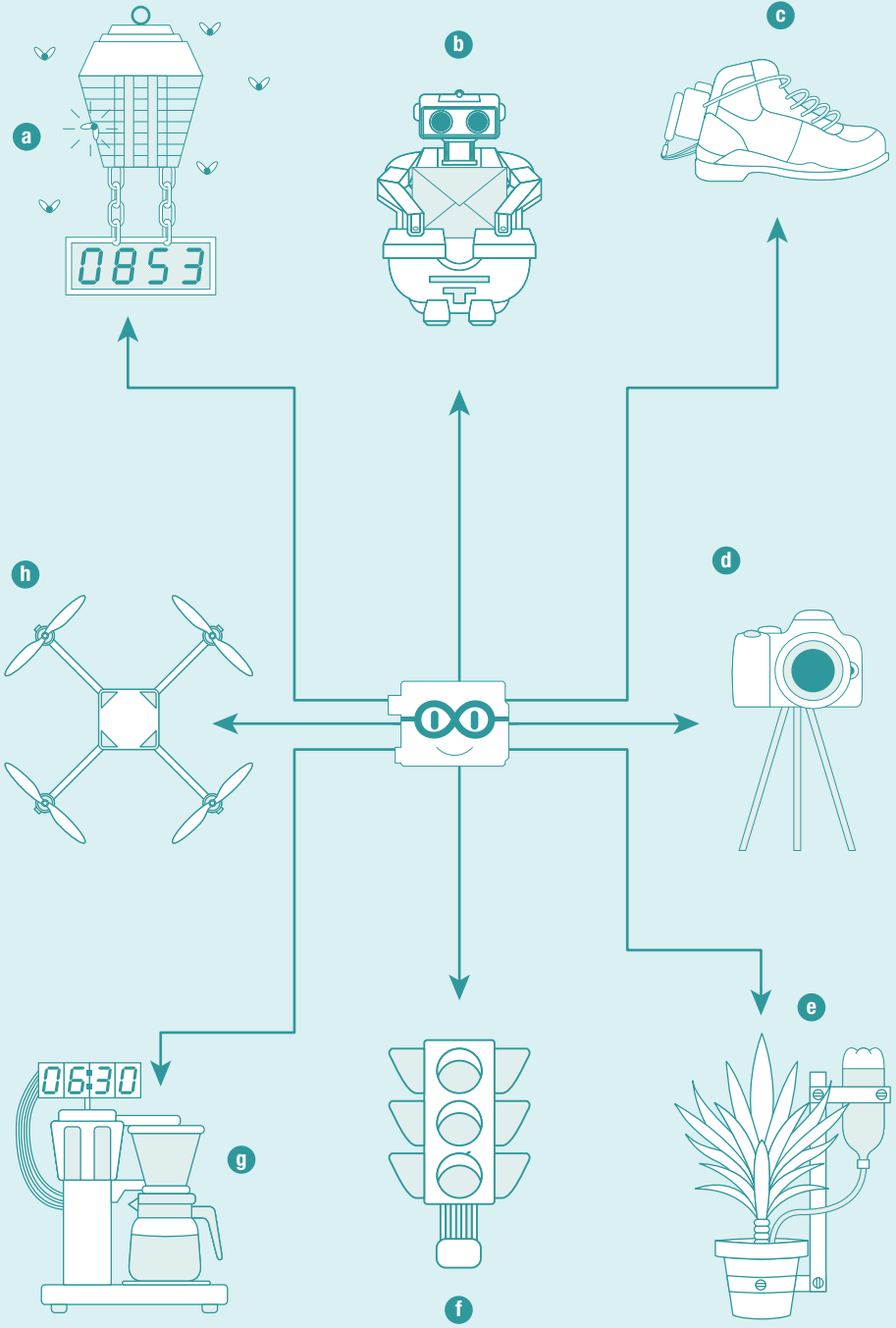
Arduino kartları bir Atmel AVR mikrodenetleyici programlama ve diğer devrelere bağlantı için gerekli yan elemanlardan oluşur. Her kartta en azından bir 5 voltluk regüle entegresi ve bir 16MHz kristal osilatör (bazılarında seramik rezonatör) bulunur. Mikrodenetleyiciye önceden bir bootloader programı 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 çıkışı), 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 değinilecektir. Şimdilik yüzeysel olarak bilinmesi yeterlidir.





a Savaşlı Böcek Yakalama

d Kamera Zamanlama Uygulaması

g Kahve Makinesi

b Oyuncaktan Email Habercisi

e Otomatik Çiçek Sulama

h Quadcopter

c Otomatik Ayakkabı Bağcığı

f Programlanabilir Trafik Işıkları

Arduino IDE'yi İndirme (Entegre Geliştirme Ortamı)



İnternet Erişimi

Arduino ile yazılım geliştirmek ve çalıştırmak için öncelikle 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 görcektir. İnternet tarayıcınızdan aşağıdaki adrese tıklayarak size uygun Arduino IDE 'yi indirebilirsiniz.

arduino.cc/en/Main/Software

1

Yükleme

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

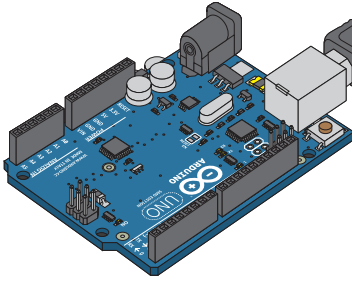
- + Windows
- + Mac OS X
- + Linux: 32 bit, 64 bit
- + source



Bilgisayarınız için uygun işletim sistemine göre yükleme paketi seçmeyi unutmayınız.

// Arduino Uno'yu Bilgisayara Baęlamak

Arduino ve Bilgisayarı USB çıkışlarından USB Kablosu ile baęlıyoruz.



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

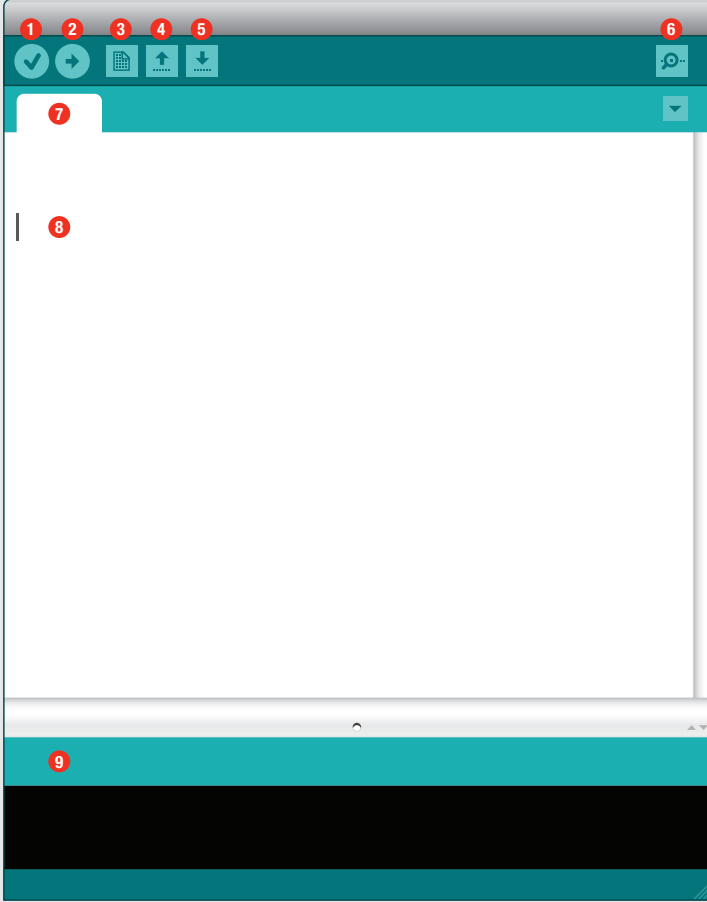
Aşağıdaki adreste Linux sürücü yüklemesi ile ilgili talimatları bulabilirsiniz.

<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 hatalarınızı 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):** Yazdığımız sketch'i kaydeder.
- 6 Serial Monitor(Seri monitör):** Seri bilgi görüntüleyen bir pencere açacaktır. Özellikle hata ayıklama için çok yararlıdır.
- 7 Sketch Name:** Şu an da üzerinde çalıştığımız sketch in adını gösterir.
- 8 Code Area(Kod Alanı):** Sketch için kod oluşturacağımız 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/ ATmega328J

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 menüsüne tıklarız. Genelde com3 veya üzeridir (COM1 ve COM2 genellikle donanım seri portu olarak kullanılmaktadır). Kullanacağımız arduino portunu bulmak için Arduino ile bilgisayar arasındaki bağlantıyı kesin ve menüye tekrar gelin ,bağlantıyı 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

Programmer

Burn Bootloader

com 1

com 12



Aygıtımız için uygun olan seri portu seçmek için Tools >Serial Port menüsüne tıklarız.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

Programmer

Burn Bootloader

/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;



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

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



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

Contents



Resources

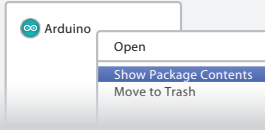


Java



examples

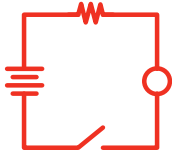
Adreste verdiğimiz sıkıştırılmış dosyayı indirip, içerisindeki "Arduino Kod" dosyasını dışarı çıkartıyoruz.



"Arduino Kod" dosyasını kopyalayıp examples klasörünün içerisine yapıyoruz.



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



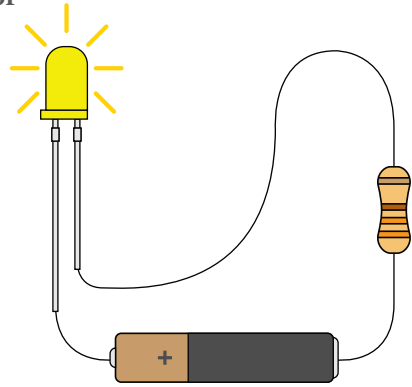
Elektrik Devresi Nedir?

Elektrik Devresi basitçe başlangıç ve bitiş noktası olan ve içerisinde bir çok devre elemanı buluduran dögüdür. Devreler her ölçüde ve şekilde yüzlerce ; direnç,diyot, 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 gösterilecektir.

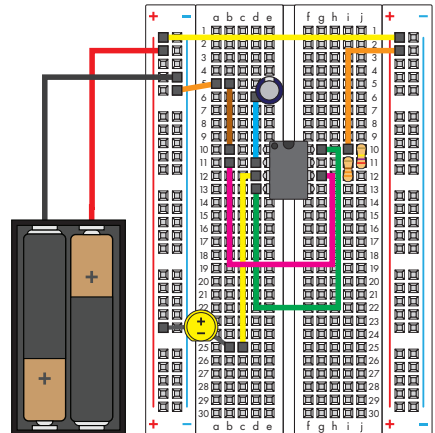
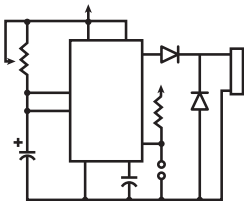
Dünya Bu Devreler Üzerinde Dönüyor

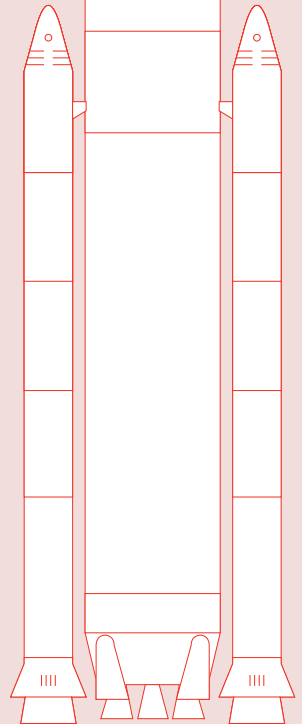
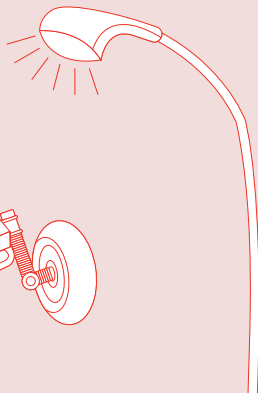
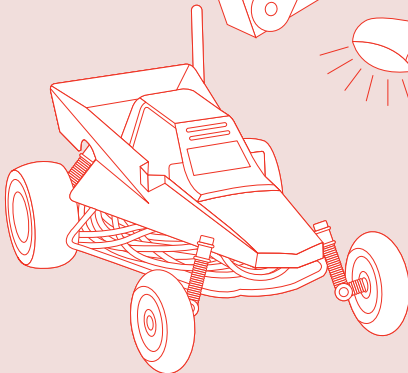
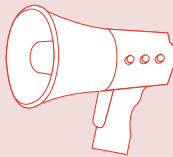
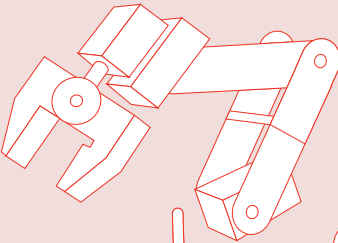
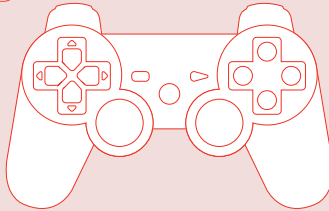
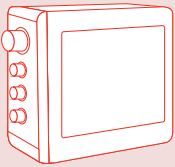
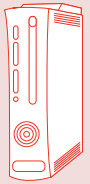
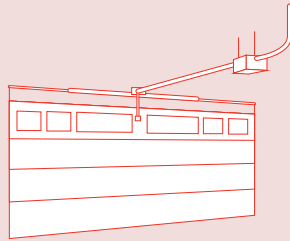
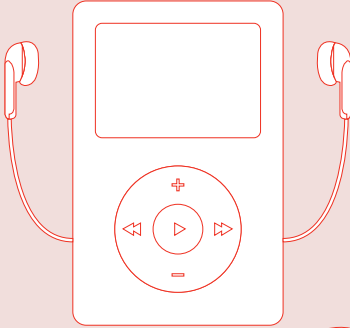
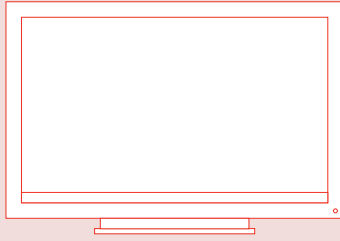
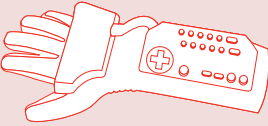
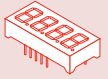
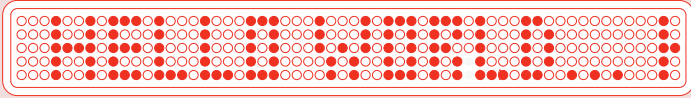
Nereye bakarsanız bakın devreler ile karşılaşacaksınız. Cebinizdeki telefon, aracınızın emisyon sistemini kontrol eden bilgisayar,oyun konsollarımız,elektrikli mutfak aletleriniz , bunların hepsi elektrik devreleriyle dolu. Bu kitapçıkta bazı basit devreleri deneyecek ve gömülü elektronik sistemlerinin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



// Basit ve Karmaşık Devreler

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

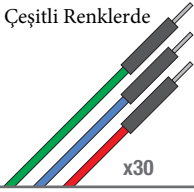




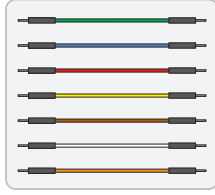
Devre Elemanları

Atlama Kablosu

Çeşitli Renklerde

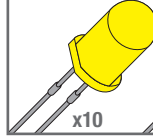


x30

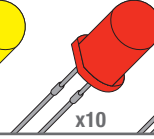


LED (5mm)

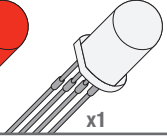
(Light Emitting Diode)



x10



x10



x1

330Ω Direnç



x25



*Gerçek Görünüm

10KΩ Direnç

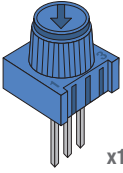


x25



*Gerçek Görünüm

Potansiyometre



x1

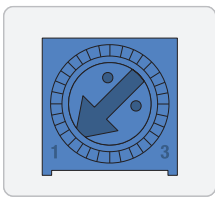
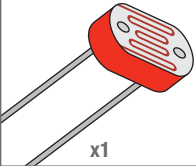
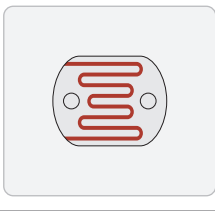


Foto Direnç (LDR)

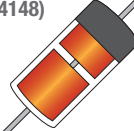


x1



Diyot

(1N4148)

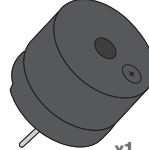


x2

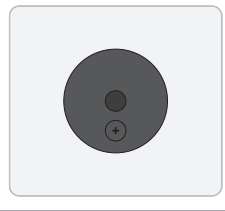


*Gerçek Görünüm

Buzzer

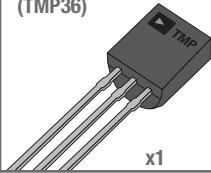


x1

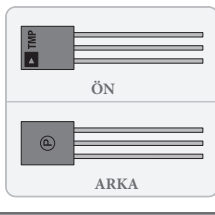


Sıcaklık Sensörü

(TMP36)

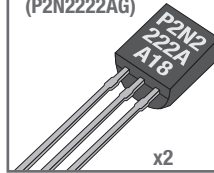


x1

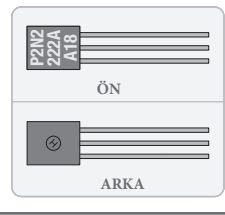


Transistor

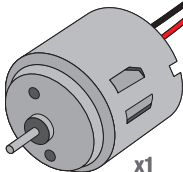
(P2N2222AG)



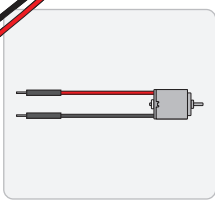
x2



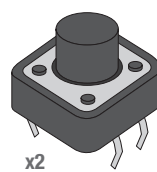
DC Motor



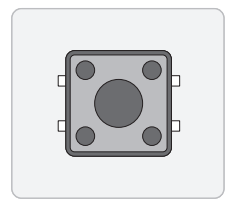
x1



Push Buton



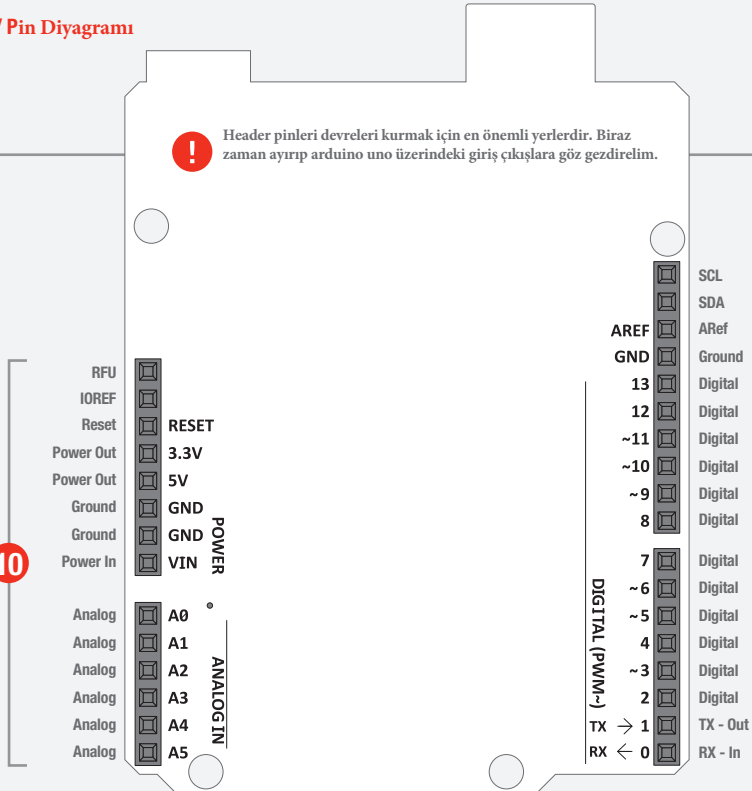
x2



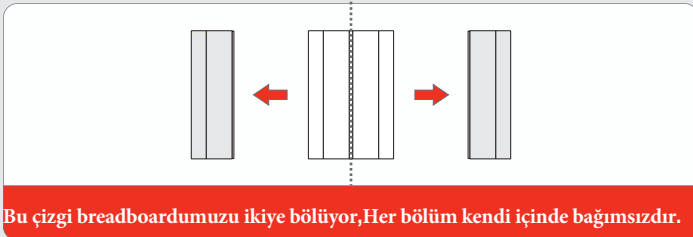
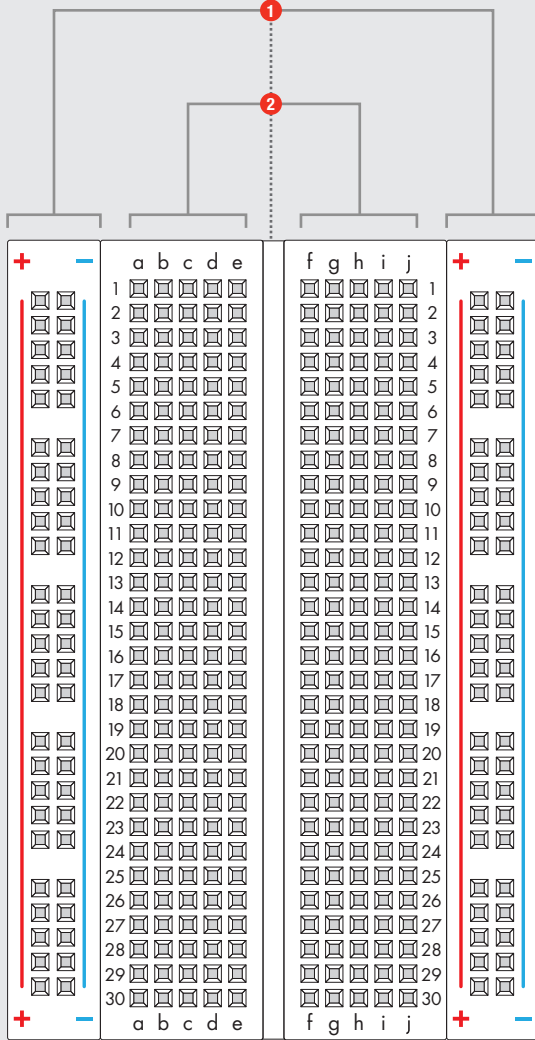
Arduino Uno

- 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 alışverişi olduğunda bu led yanar.
- 4 LED(TX: Transmitting) - Seri haberleşme için kullanılan TX pininin durumunu gösteren LED'dir. Veri alışverişi 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 işaretli bulunan pinler aynı zamanda analog çıkış(PWM) almak içinde kullanılıyor.
- 7 LED (Güç LED'i)-Kartımızın güç göstergesi LED 'idir. Devrelerinizi kurarken kısa devre yapıp yapmadığımız bu LED 'e bakarak anlayabilirsiniz.
- 8 Reset Butonu -Arduinoyu yeniden başlatır.Programlarımızı setup() fonksiyonundan itibaren yeniden başlar.
- 9 Kart Üzerinde Programlama(ICSP) Pinleri - Atmega microdenetleyicisi 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)



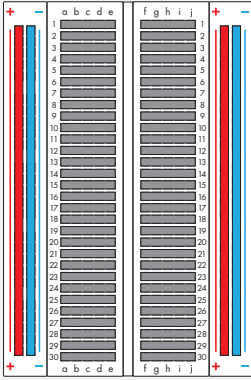
Bu çizgi breadboardumuzu ikiye bölüyor, Her bölüm kendi içinde bağımsızdır.

Breadboard

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

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

Bağlantılar Nasıldır?



+ Güç:

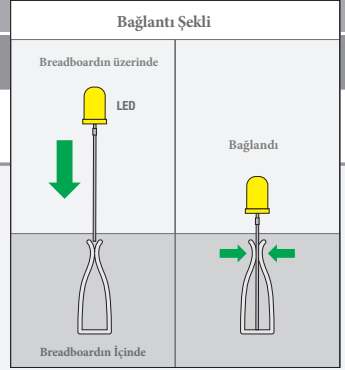
Her + işareti dikey sütundan güç geçişi sağlar.

- Toprak:

Her - işareti dikey sütunda her yerde toprak görevi görür.

Yatay Satırlar:

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



Breadboardın İç Görünümü>>>

Devreler #1 - İlk Devreniz

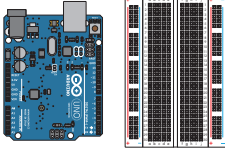
Nasıl Çalıştıracamız;

1 Parçaları Birleştir

2 Kodu Yaz

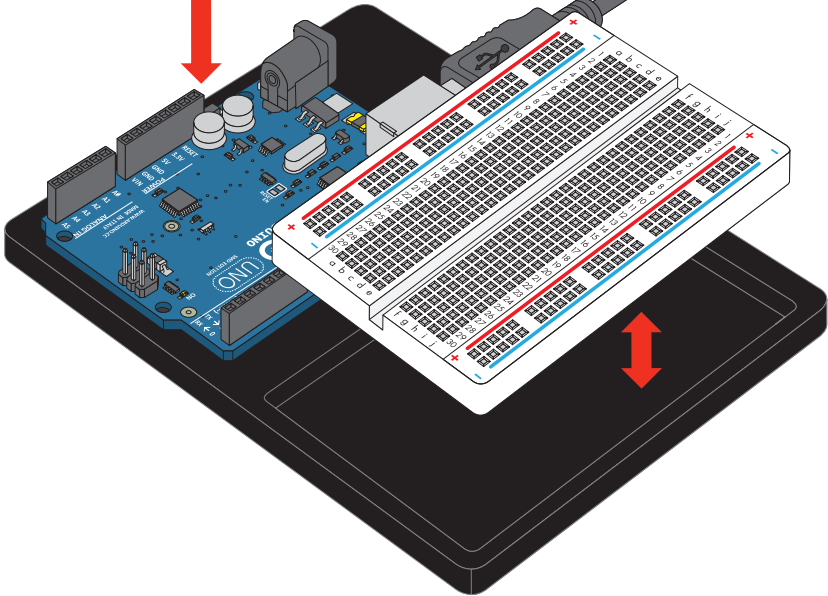
3 Arduino'ya Yükle

+ Breadboard hakkında genel bilgiler verildi, şimdi arduino ile breadboardu yan yana yerleştirelim ve talimatlara dikkat edelim.



+ Arduino ve Breadboardı yan yana yerleştir

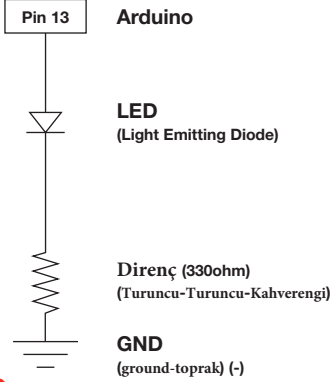
+ USB Kablosunu Bağla



5V Akım Arduino 5V da çalışır. Arduino'yu bilgisayara bağladığımızda aldığımız güç kurduğunuz devreleri çalıştıracaktır. Bilgisayarımızı Arduino'ya bağlayarak doğru gerilimi alabilirsiniz. 5V size zarar vermez bu yüzden devre içerisinde dokunmaktan korkmayın.

LED YAKMAK

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



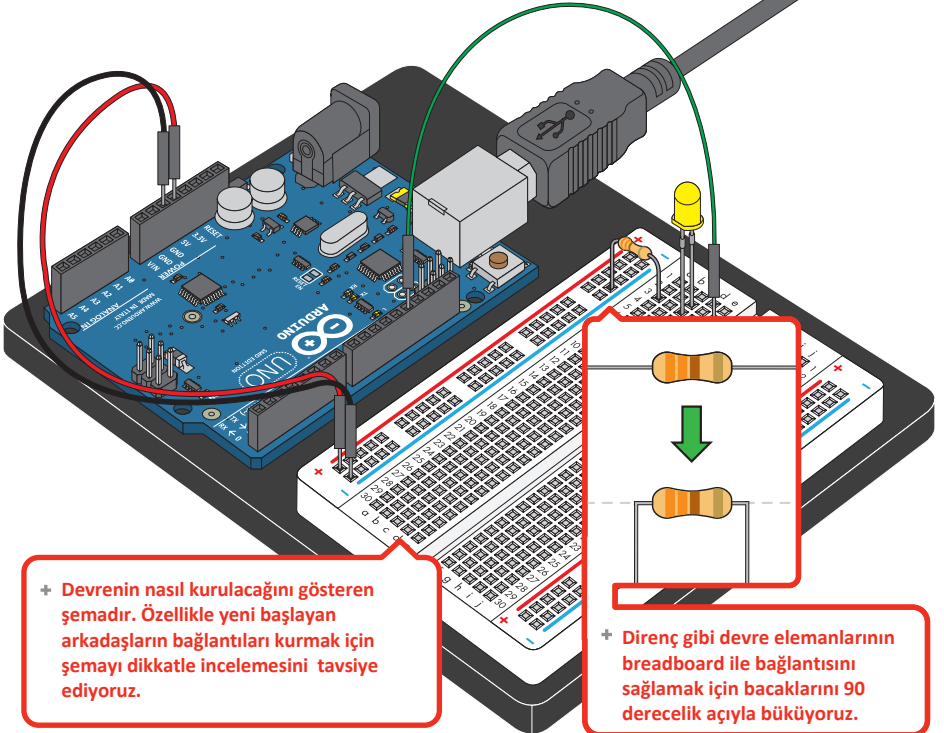
+ Bu LED yakma devremizin şemasıdır.

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

Elemanlar



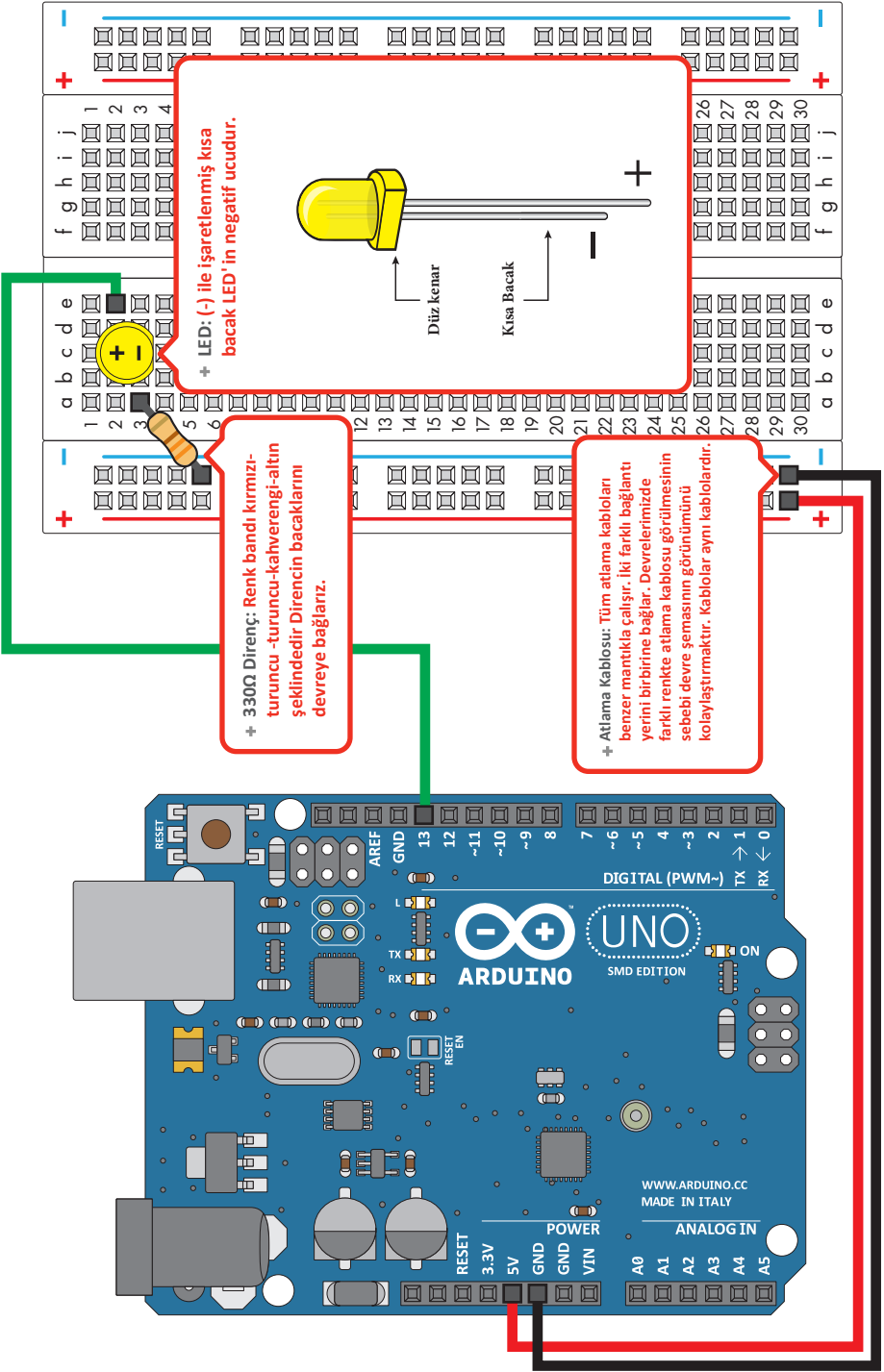
+ Bu bölümde, devreyi tamamlamak için gereken parçalar listelenir.













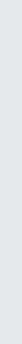




+ Devrenin nasıl kurulacağını gösteren şemadır. Özellikle yeni başlayan arkadaşların bağlantıları kurmak için şemayı dikkatle incelemesini tavsiye ediyoruz.

+ Direnç gibi devre elemanlarını breadboard ile bağlantısını sağlamak için bacaklarını 90 derecelik açıyla büküyoruz.

Devre 1: Led Yakmak

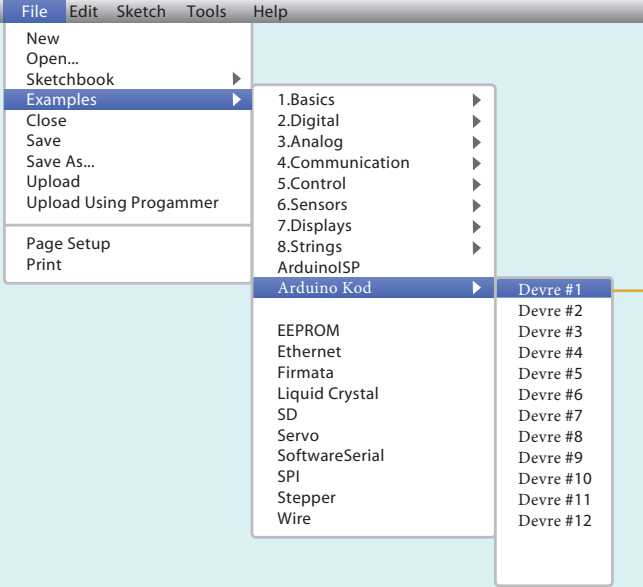


Elemanlar:	Gerçek Görünümleri:			
LED (5mm)				+ LED'İ c2 (uzun bacak) c3(kısa bacak) şeklinde breadboard'un soketlerine takıyoruz.
330Ω Direnç				+ Dirençler yalnızca Breadboard üzerindeki soketlere yerleştirilir. "-" işaretli satıra ve LED'in bacağına bağlıyoruz.
Atlama Kablosu				+ Arduino üzerindeki "GND" pinini breadboard üzerindeki "-" işaretli satıra bağlıyoruz.
Atlama Kablosu				+ Arduino üzerindeki "5V" pinini breadboard üzerindeki "+" işaretli satıra bağlıyoruz.
Atlama Kablosu				+ Arduino üzerindeki "Pin 13" pinini breadboard üzerindeki "e2" soketine bağlıyoruz.
				+ Breadboard: Beyaz şerit breadboard soket bağlantılarını temsil eder
				+ Arduino: Mavi şerit Arduino header pinlerini temsil eder.



İlk Kodumuzu Açalım:

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



//Devre #1

```
Circuit #1
/*
Led Yakmak (Blink=Göz Kırpma)
LED' i bir saniye yakıp bir saniye söndüren program
*/

int LEDcikis = 13;
void setup() {

// Dijital Pini 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

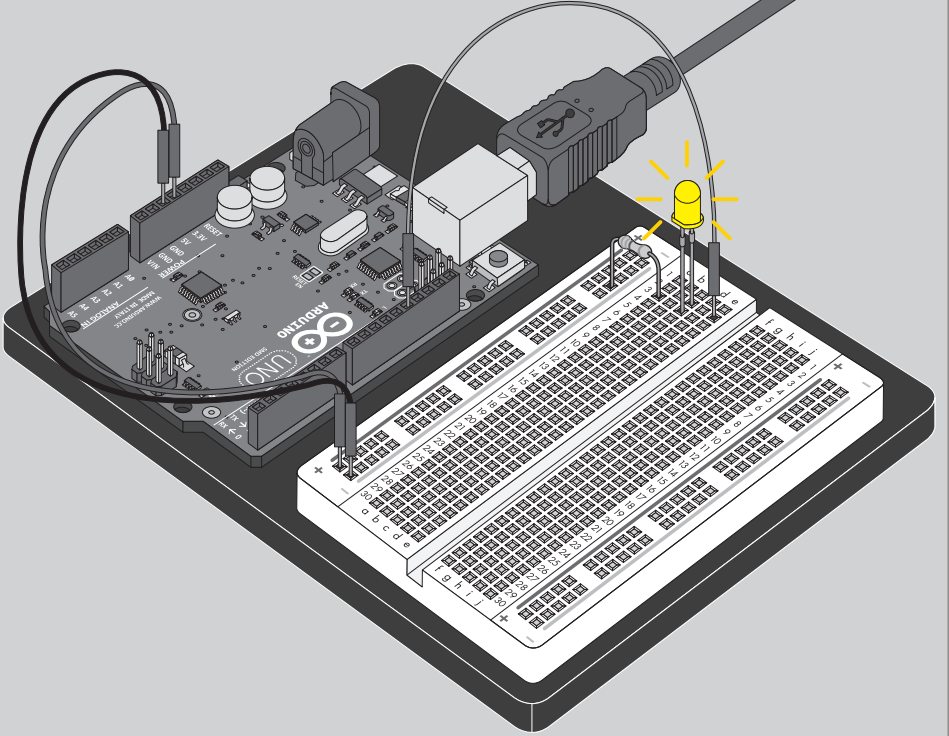
Kodu derliyoruz. Yazılım kodu arduino üzerindeki entegrenin anlayabileceği dile çeviriyor.



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

+ Tüm devrelerde kodların ne olduğunu açıklayan bölümdür.



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

Kod notları:

+ Kodun nasıl çalıştığını anlamak için aşağıdaki açıklamaları inceleyin.

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



pinMode(13, OUTPUT);



Arduinoki pinleri kullanmadan önce pini INPUT (giriş) yada OUTPUT (çıkış) olarak tanımlamanız gerekmektedir . Bunu yapmak için *pinMode()* yerleşik fonksiyonunu kullanacağız.

digitalWrite(13, HIGH);

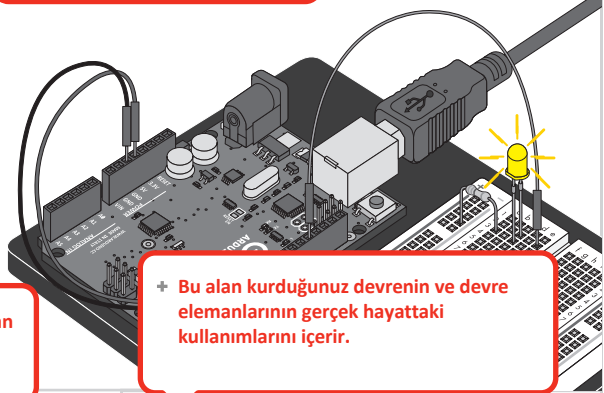


Bir pini OUTPUT (çıkış) olarak tanımladığımızda, o pinin HIGH (output 5 Volts-5 Volts çıkış), veya LOW (output 0 Volts-0 Volt çıkış) olarak davranmasını sağlayabilirsiniz .

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 hayattaki kullanımlarını içerir.

Sorun Giderme:

LED Yanmıyor

LED sadece tek yönlü çalışır. Bu tür sorunlar genelde LED in yanlış bacağına 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 adresine mail yollayabilirsiniz.

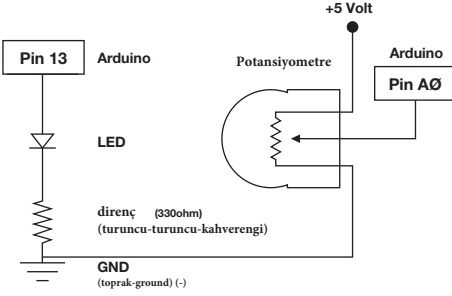
Gerçek Hayatta Uygulamaları:

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



Potansiyometre

Bu devrede potansiyometrenin ne işe 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 potansiyometriyi nasıl kullanacağınızı öğreneceksiniz.



Elemanlar:

Potansiyometre



x1

LED



x1

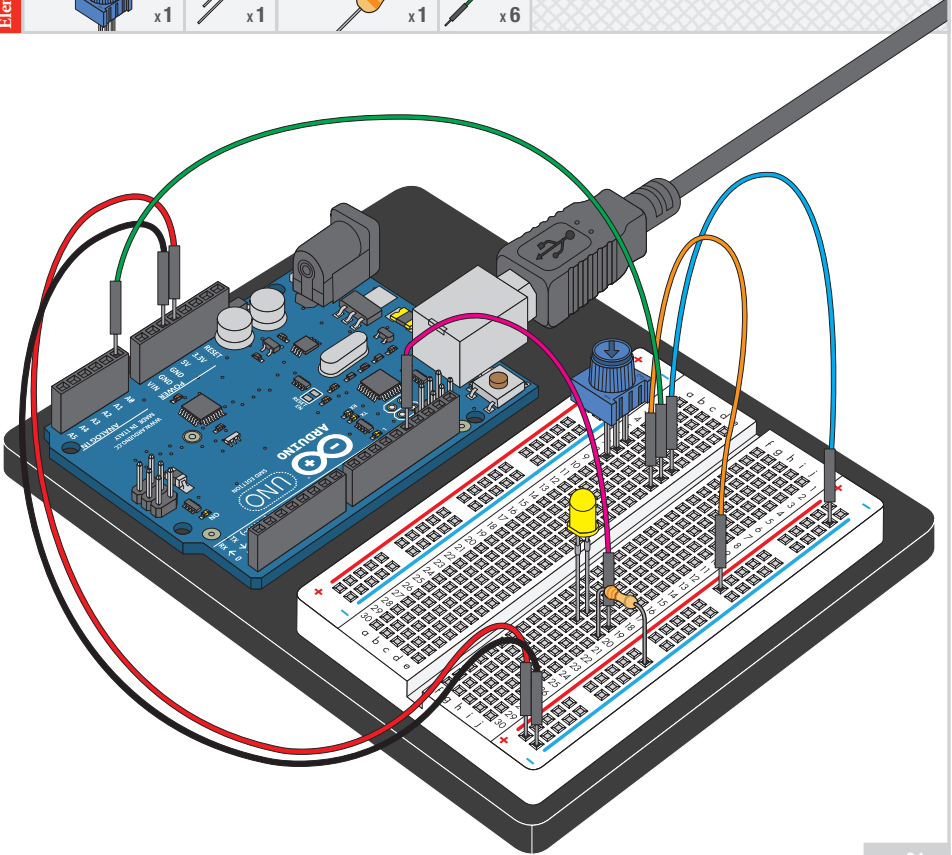
330Ω
Direnç

x1

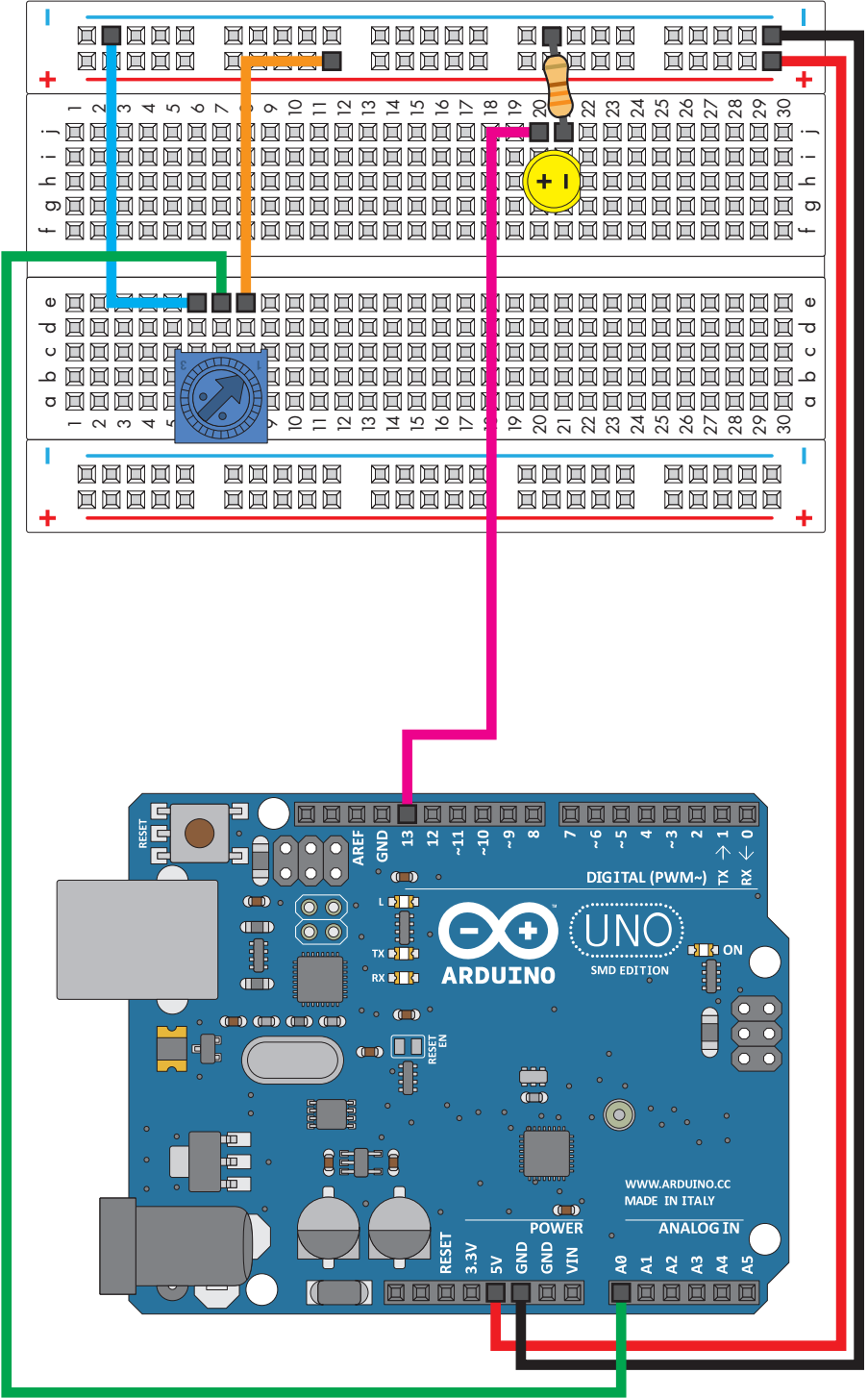
Kablo



x6



Devre 2: Potansiyometre

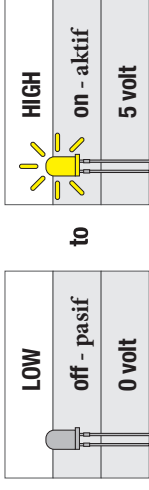


Dijital V Analog:

Eğer Arduinoya yakından bakarsanız bazı pinlerin "Dijital" bazı pinlerin ise "Analog" olduğunu göreceksiniz. Peki ama ne farkı var bu pinlerin?

Arduino üzerindeki dijital pinler ile dış dünyadan sinyal almak ve dış dünyaya sinyal vermek çok kolaydır. Hatta LED leri çok hızlı yakıp söndürerek dim sergilemek gibi numaralar ve seri iletişim de "HIGH" ve "LOW" gibi kalıplar ile başka cihazlara veri aktarımında kolaylıkla kullanılabilir.

Dijital



Ama sadece on ya da off olarak tanımlamayacağımız birçok şey vardır. Sıcaklık seviyeleri, kontrol düğmeleri gibi birçok şeyde pasif ve aktif (HIGH ve LOW) arasında sürekli değişebilen değerler alabilirler. Bu durumlarda Arduino bir giriş voltajını 0 (0 volt) ile 1023 (5 volt) arasında değişebilen sayılara dönüştürebilen altı analog giriş sunar. Analog girişler tüm bu gerçek değerleri ölçmekte çok iyidirler ve bu tarz değerleri Arduino'ya aktarmanızı sağlarlar.

Analog



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



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

Kod Notları:



int sensorValue;



"Değişken" sizin isim verdiğiniz bir rakamdır. Değişkeni kullanmadan önce onu tanımlamalısınız 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") dönüştürür, bu rakam 0 (0 Volt) ile 1023 (5 Volt) arasındadır.

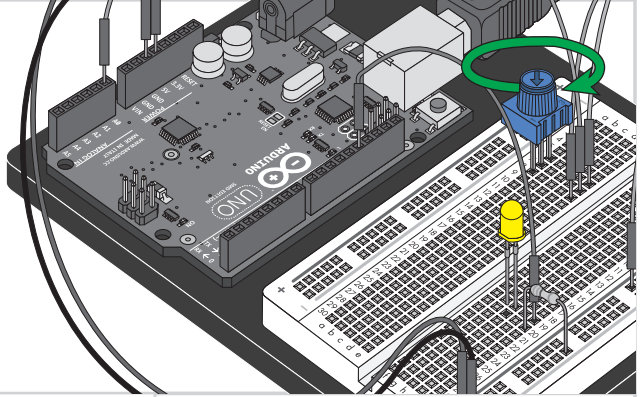
delay(sensorValue);



Arduino her saniye binlerce satır kodu çalıştıracak kapasiteye sahip, çok hızlı bir yapıdır. Onu yavaşlatıp neler yaptığını gözlemlemek 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 hızlı ve yavaş bi şekilde yanıp söndüğünü göreceksiniz. Eğer çalışmıyorsa 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 oturmasından kaynaklanabilir. Potansiyometreyi bastırarak bu sorunu çözebilirsiniz.

Çalışmama Durumu

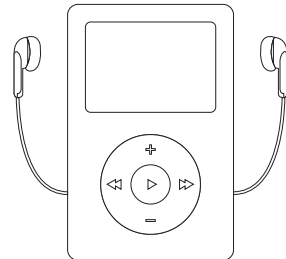
Potansiyometrenin 2. pinini analog pin yerine dijital pine bağlamadığınızdan emin olunuz.

Hâlâ Çalışmıyor mu?

Tamamlayamadığınız her devrenin hakkında yardım almak için emrearslan@elektrikport.com adresine mail yollayabilirsiniz.

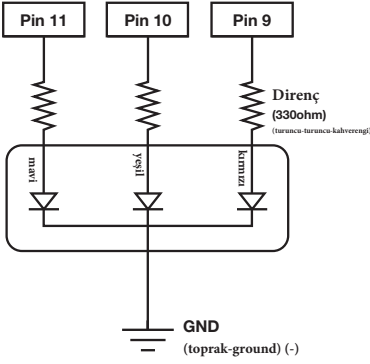
Gerçek Hayatta Uygulamaları:

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 barındıran ve her tür renk için kombin edilebilen bir diyet türüdür. Bu devrede Bir RGB LED kullanarak nasıl renk kombinasyonları oluşturabileceğinizi öğreneceksiniz. Her bir diyetün parlaklığına bağlı olarak neredeyse bütün renkleri elde etmeniz mümkün.



Elemenlar:

LED



x1

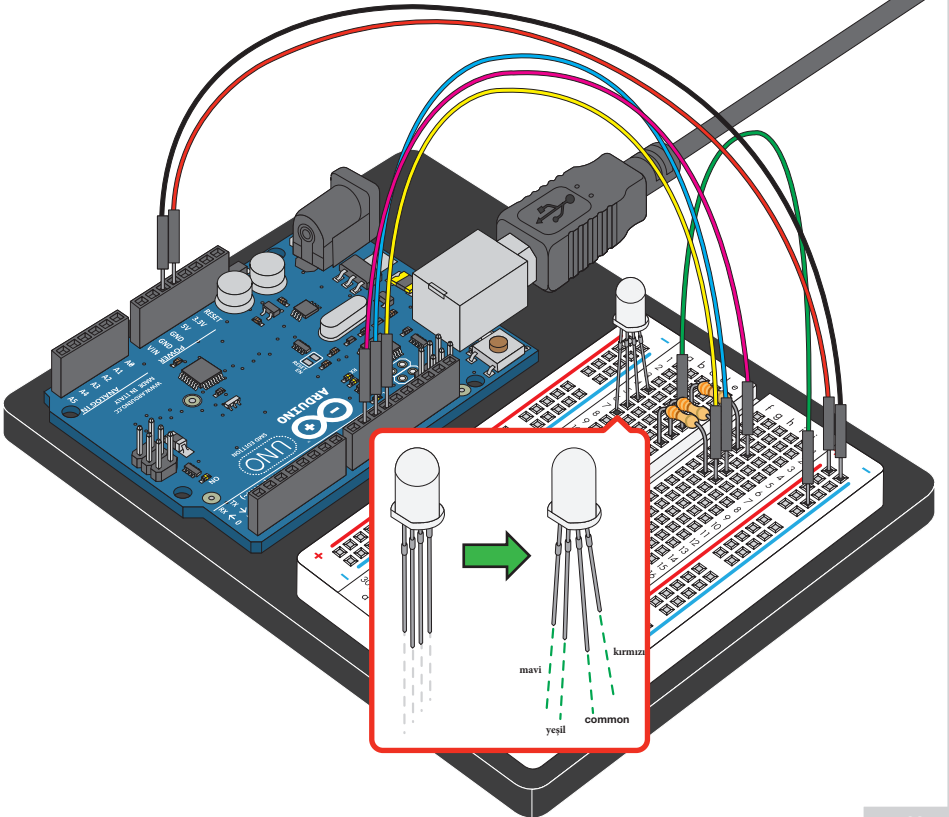
330Ω
Direnç

x3

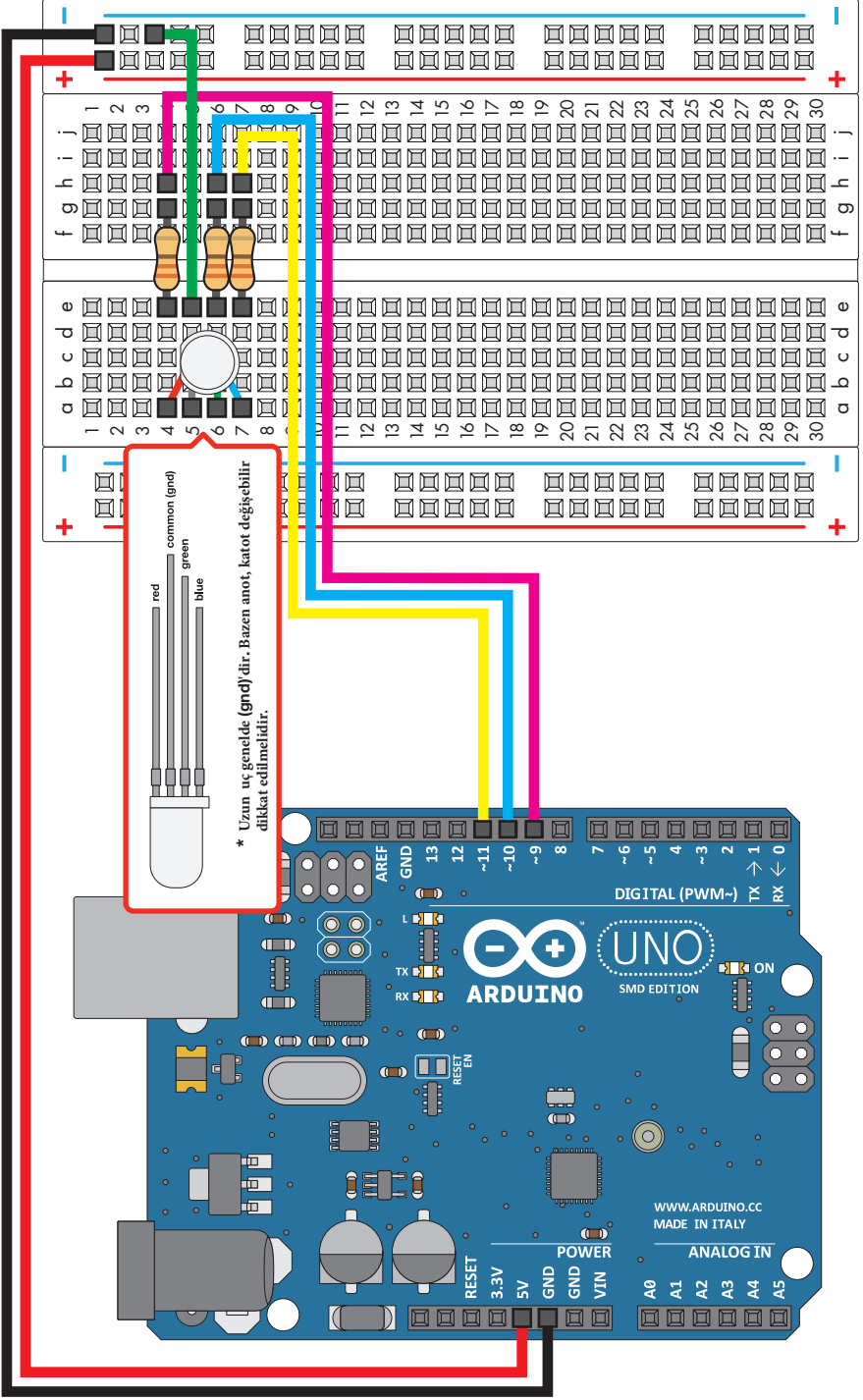
Kablo



x6



Devre 3: RGB LED

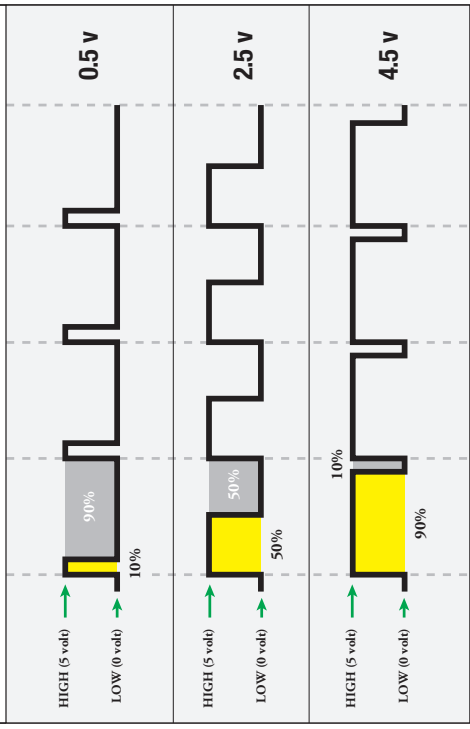


Elemanlar:	Gerçek Görünüm:		
RGB LED (5mm)			
330Ω Direnç			
330Ω Direnç			
330Ω Direnç			
Atlama Kablosu			
Atlama Kablosu			
Atlama Kablosu			
Atlama Kablosu			
Atlama Kablosu			
Atlama Kablosu			

analogWrite() Arkasındaki Şok Edici Gerçek:

Şimdiye kadar Arduino'nun analog voltajları (0 ve 5 voltaj arasında ki değerleri) analogRead() fonksiyonu kullanarak okuyabildiğini gördük. Peki Arduino'nun analog voltaj çıktısı vermek gibi bir imkanı var mı diye soracak olursak, cevabımız hem hayır hem evet olacaktır. Arduino gerçek bir analog voltaj çıkıtısına sahip değildir ama Arduino'nun çok hızlı olmasından dolayı PWM ("Pulse-Width Modulation") kullanılarak bu çıktıyı uyandırabilir*

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 vakit harcamıyorsa bu pine bağlı olan LED parlak yanacaktır. Fakat LOW konumunda daha fazla vakit harcamıyorsa LED daha sönük kalacaktır çünkü pin gözün görebileceğinden daha hızlı açıp kapama yapacaktır. İşte bu Arduino'nun "gerçek" analog çıktısı için yaptığı bir illüzyondur.





Arduino IDE 'yi Aç// File > Examples > Arduino Kod > Devre # 3

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 brackest {} içerisindeki ifade yürütülür.

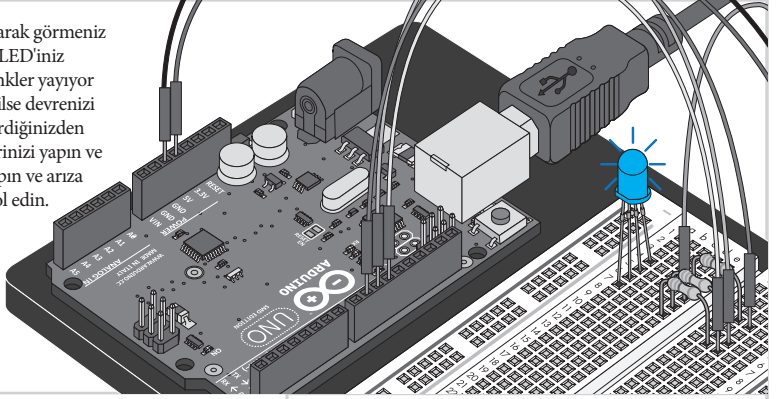
delay(sensorValue);



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

Ne göreceğiz?

LED'inizi çalışıyor olarak görmemiz lazım, fakat bu sefer LED'iniz sırayla farklı farklı renkleri yayıyor olmalı. Eğer öyle değilse devrenizi doğru şekilde birleştirdiğinizden emin olun, kontrollerinizi yapın ve kartınıza yüklemeyi yapın ve arıza tespiti kısmını kontrol edin.



Sorun Giderme:

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

Kırmızı Renk Görmek

RGB LED içerisindeki kırmızı diyot diğerlerinden daha parlak olabilir. Renklerinizi daha dengeli hale getirmek için daha yüksek bir direnç kullanın veya kodunuzda ayarlamalar yapın.

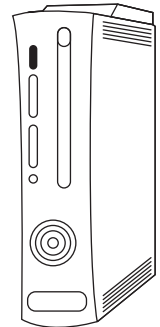
```
analogWrite(RED_PIN, redIntensity);
```

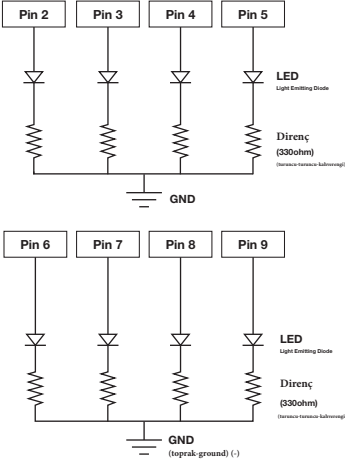
```
to
```

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

Gerçek Hayatta Uygulamaları:

Videogame konsolları gibi bir çok elektrikte 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 çıtayı 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 öğreneceksiniz.

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

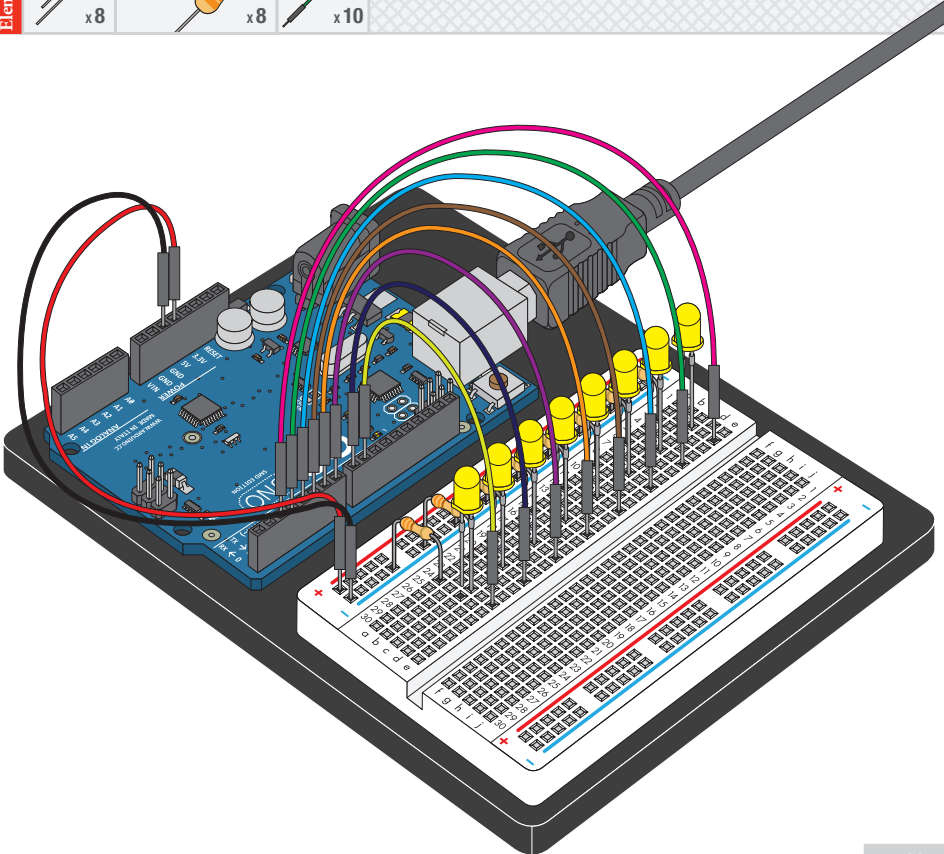
arrays[] - Birden fazla değişkeni grup haline getirerek yönetilmelerini kolaylaştırır.

Elemanlar:

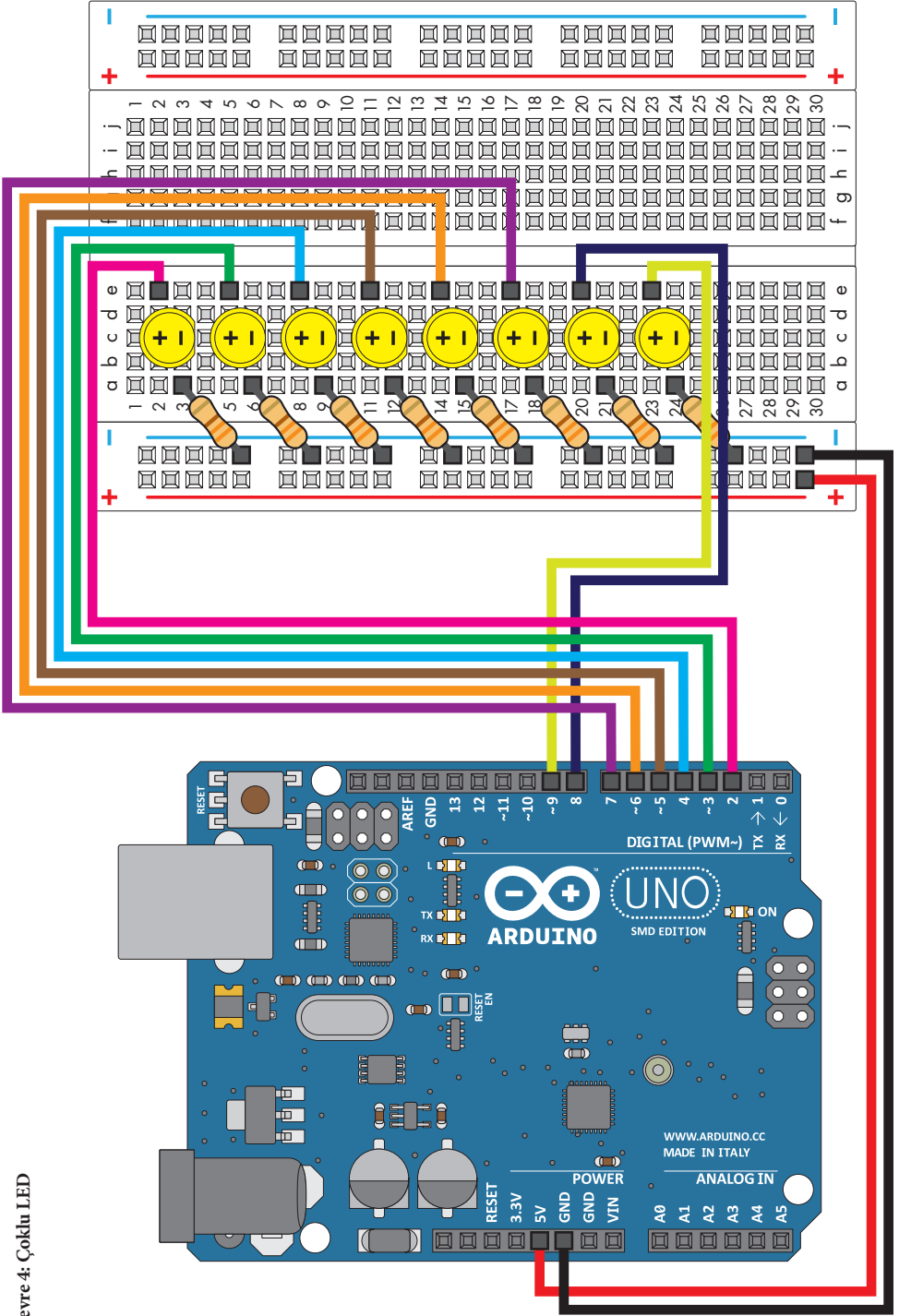
LED
x 8


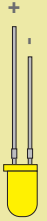








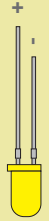












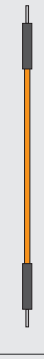

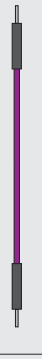





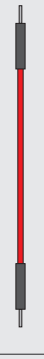


330Ω
Direnç
x 8

Kablo
x 10



Devre 4: Çoklu LED



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



Arduino IDE 'yi Aç// 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 buldukları adres yardımı ile ulaşırsınız. İlk elemanın adresi 0, ikinci elemanın adresi 1, vs. Bir elemana ulaşmak için “ledPins[x]” komutunu kullanarak x yerine o elemanın adresini yazarsınız. Burada dijital pin 2'yi HIGH yapıyoruz.

`index = random(8);`

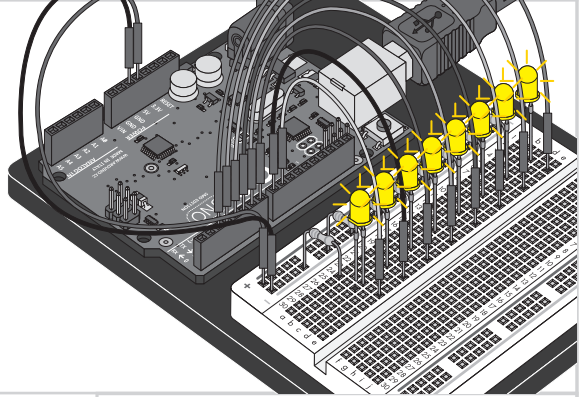


Bilgisayarlar çalışma sırasında aynı işlemleri gerçekleştirir fakat bazen bir şeylerin rastgele olmasını istersiniz, örneğin bir zar atma işleminde. Random() fonksiyonu bunu yapmak için en iyi yoldur. Daha fazla bilgi için adresi ziyaret ediniz.

<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 Vermeyebilir

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

Sıralama Düzeni

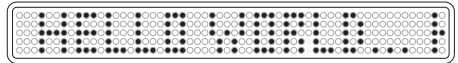
Sekiz adet bağlantı ile çalıştığınız için karışıklıklar olması gayet doğal. İlk LED'i pin 2'ye yerleştirin ve diğer LED'leri devam edecek şekilde yerleştirin ve tekrar kontrol edin.

Yeniden Başlayın

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

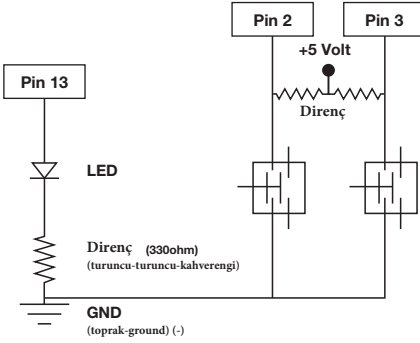
Gerçek Hayatta Uygulamaları:

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

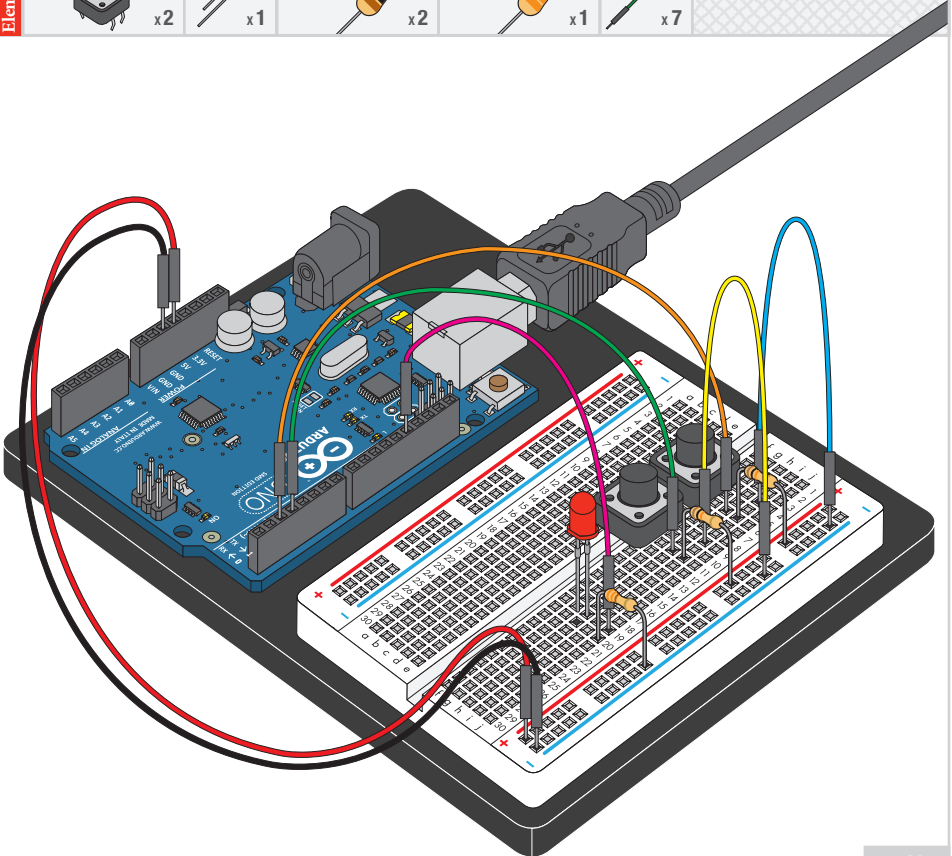


Buton Kullanımı

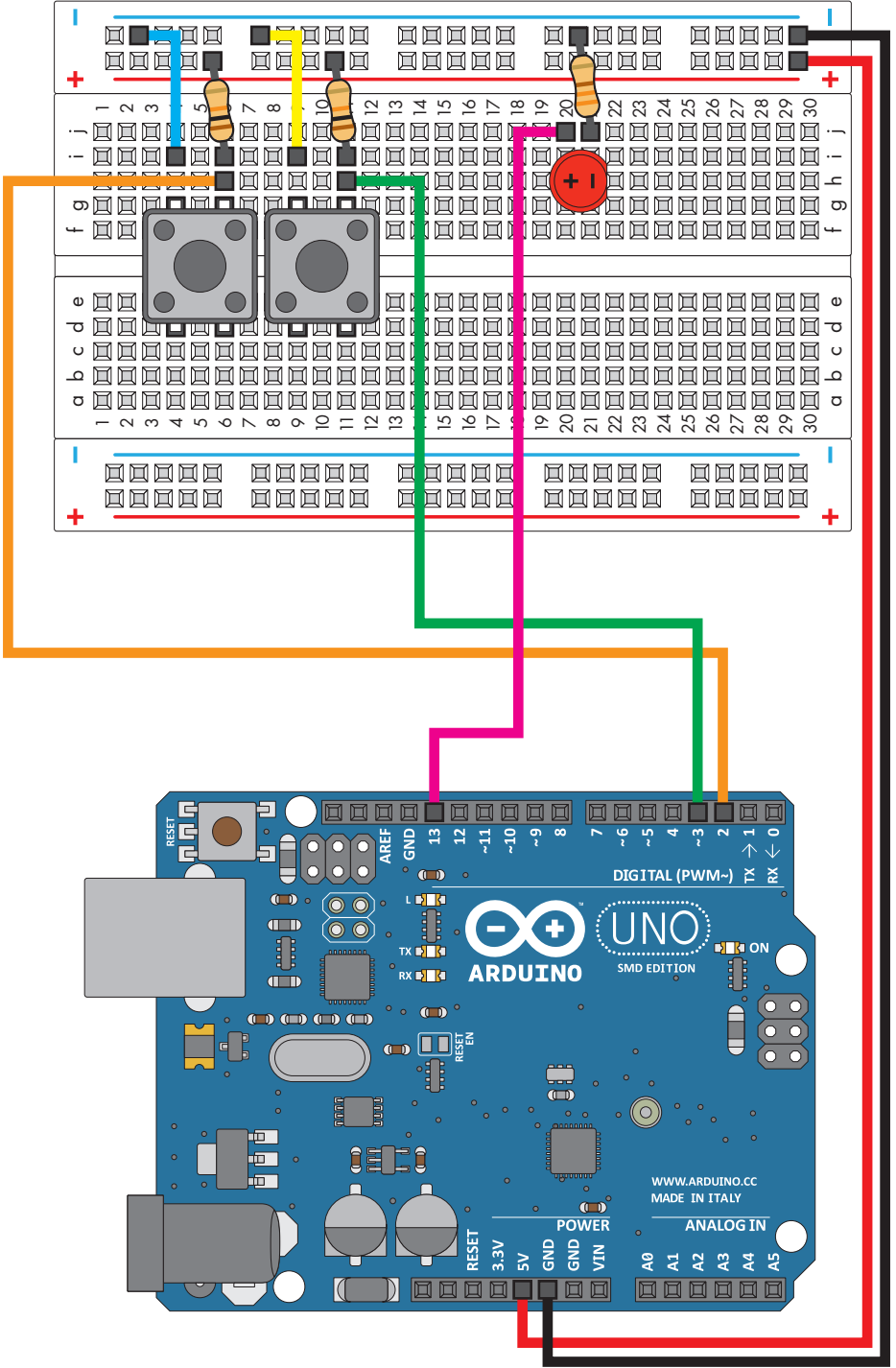
Şu ana kadar çıktı (output) üzerine yoğunlaşmıştık. Bundan sonraki projelerimize girdiler (inputs) ile devam ediyoruz. Bu devrede çok yaygın olan bir girdiye bakacağız push buton(buton). Bir butonun Arduino üzerindeki çalışma şekli şöyledir; ne zaman ki butona bastınız, voltaj LOW seviyesine geçer. Arduino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



Elemenlar:	Push Button	LED	10KΩ Direnç	330Ω Direnç	Kablo
	 x2	 x1	 x2	 x1	 x7



Devre 5: Buton Kullanımı



Iron Man Olmak için Arduino:

Arduino'yu kullanmış kulan şeylerden biri de, gündüzdeki bilgilere dayanan karmaşık eylemleri gerçekleştirebilir. Örneğin hava fazla soğuk olduğunda ısıtıcıyı çalıştırabilir, sıcak olduğunda vantilatörü açabilir, bitkiler kurumaya başladığında onları sulayabilirsiniz. Bu eylemleri yerine getirebilmesi için, Arduino "if" ile kurdüğunuz karmaşık yapıları mantıklı işlemlerle gerçekleştirebilir.















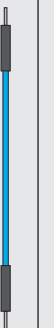







==	Eşitlik	A == B doğru ise A ve B aynıdır.
!=	Farklılık	A != B doğru ise A ve B aynı değildir.
&&	Ve	A && B doğru ise A ve B'nin ikisi de doğrudur.
 	Veya	A B doğru ise A veya B doğrudur.
!	Değil	!A doğru ise A yanlıştr. Yanlış ise A doğrudur.

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

Örneğin:

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

Eğer ısıtma modundaysanız ve sıcaklık düşerse bu kod ısıtıcının çalışacaktır. Bu tarz işlemlerle Arduinoyu akıllı eylemleri gerçekleştirebilecek şekilde kodlayabilir ve dış çevreyi kontrol altına alabilirsiniz.

Elemanlar:	Gerçek Görünüm:		
Push Buton			d4 g4 d6 g6
Push Buton			d9 g9 d11 g11
LED (5mm)			h20 h21 + -
10KΩ Direnç			i6 + i11 +
10KΩ Direnç			i11 +
330Ω Direnç			i21 - i4 - i9 -
Atlama Kablosu			i4 - i9 -
Atlama Kablosu			i9 -
Atlama Kablosu			h6 Pin 2
Atlama Kablosu			h11 Pin 3
Atlama Kablosu			h20 Pin 13



Arduino IDE 'yi Aç// File > Examples > Arduino Kod > Devre # 5

Kod Notları:



pinMode(button2Pin, INPUT);



Dijital pinler çıktılar gibi girdi olarak da kullanılabilir. Fakat bu işlemi yapmadan önce, Arduinoya kullandığımız yolu söylemeniz gerek.

button1State = digitalRead(button1Pin);



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

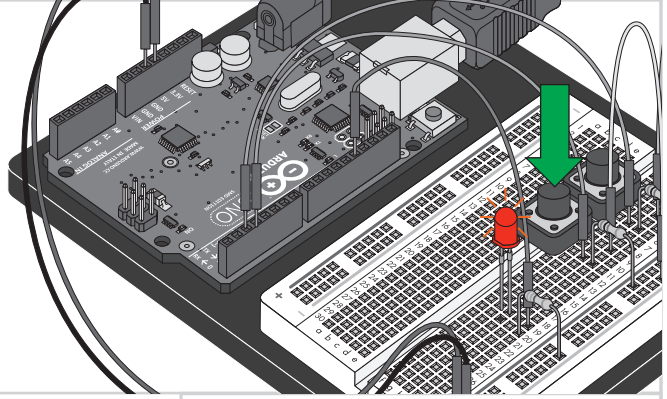
if (button1State == LOW)



Çünkü butonumuzu GND'ye bağlıyoruz ve butona bastığımızda LOW olarak okunuyor. Butonun basılmış durumda olup olmadığını görmek için ("==") operatörünü kullanıyoruz.

Ne göreceğiz?

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



Sorun Giderme:

Işık yanmıyor

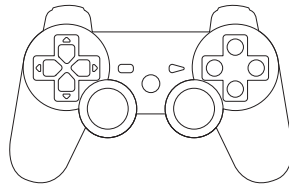
Butonuzum kare olduğu için yanlış yerleştirilmiş olabilir. 90 derece çevirin ve çalışıyor mu tekrar kontrol edin.

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.



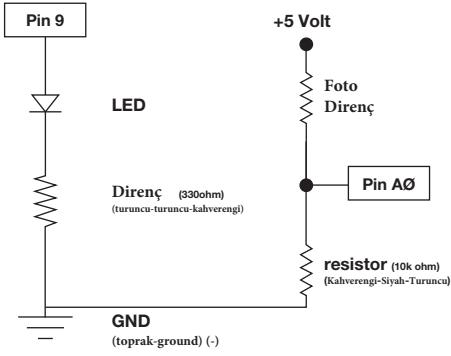
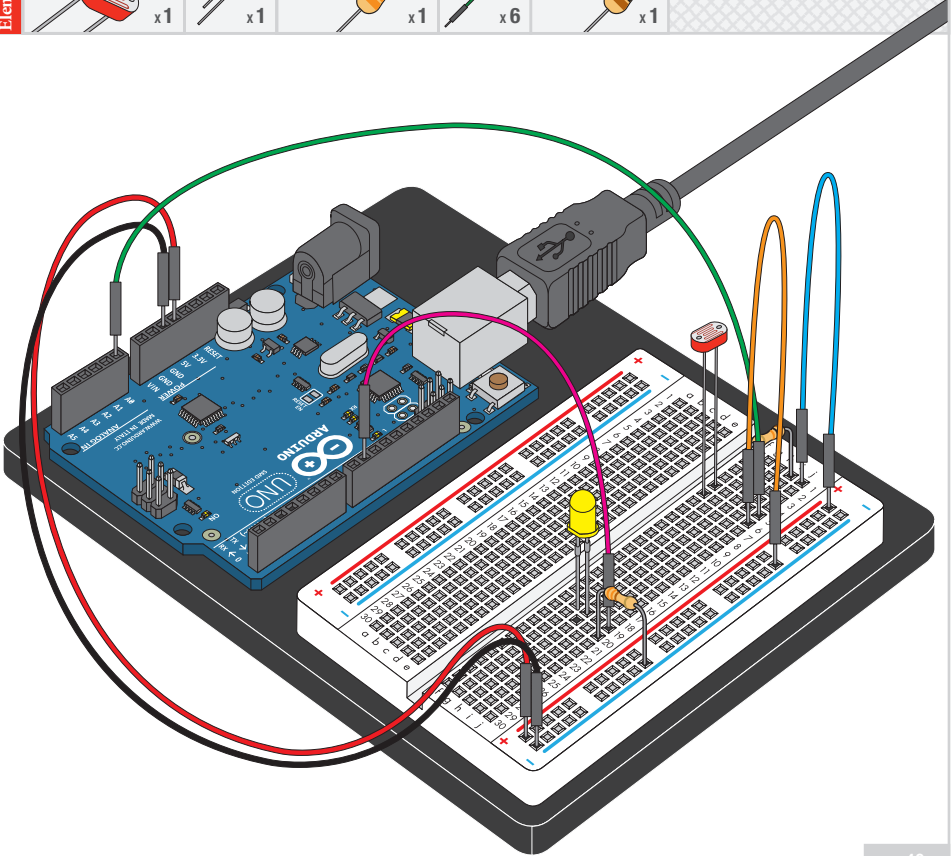


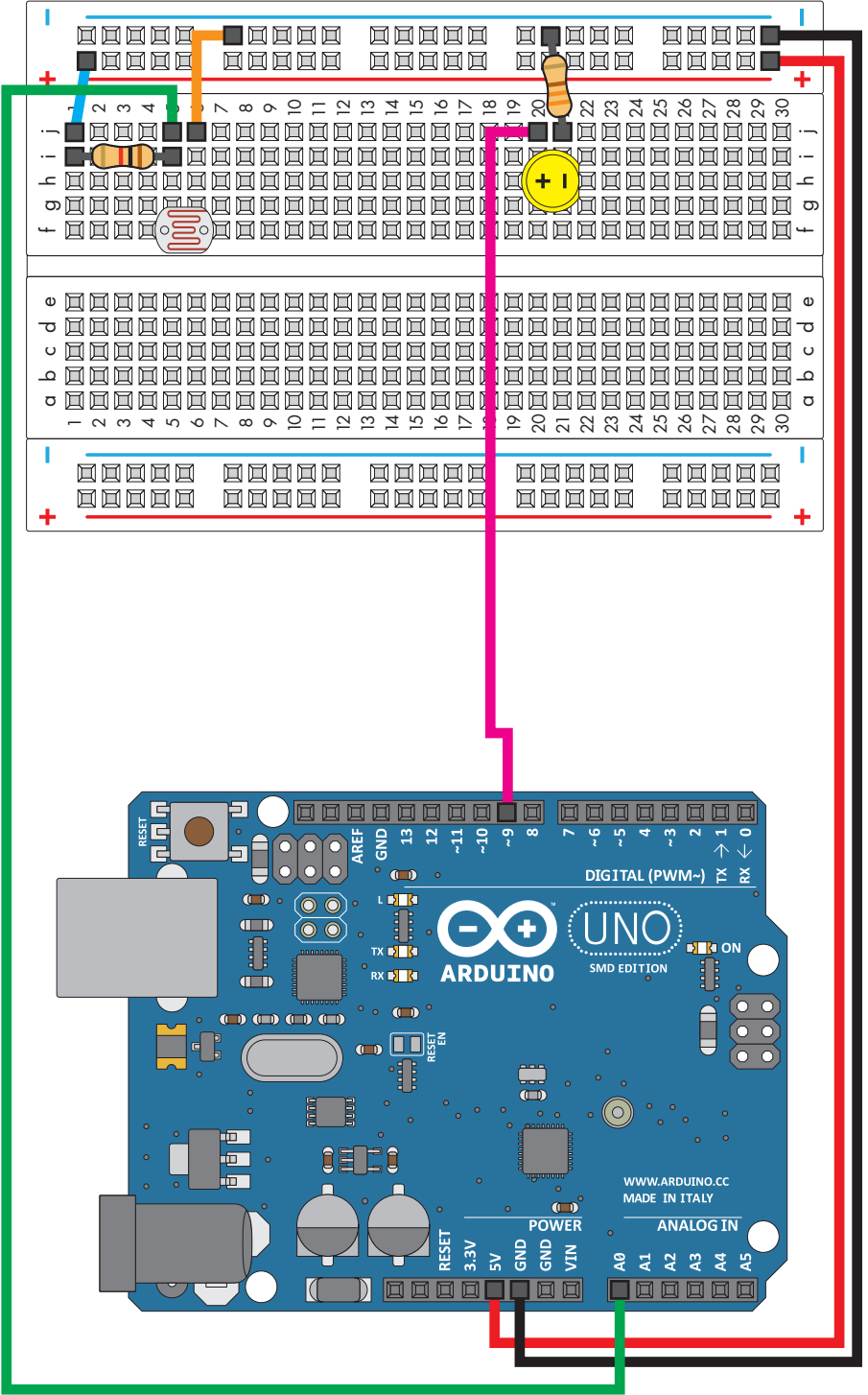
Foto Direnç

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

Elemanlar:



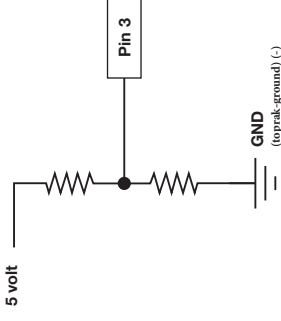
Devre 6: Foto Direç



Direnç Ölçüm Sensörleri:

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

Arduino analog giriş(input) pini direnci değil voltajı ölçer. Ama biz Arduinomuz ile birlikte "voltaj bölücü" olarak kullanılan dirençli sensörler kullanacağız.



Bir voltaj bölücü iki adet dirençten meydana gelir. Üst direnci 5 volta ve alt direnci toprağa(GND) bağladığımızda, orta kısım iki direnç değerine uygun olan bir volt çıkışı verir. Dirençlerden birinin değeri değiştiğinde (Algıladıkları birimin değeri değiştiğinde) direnç değeri de değişecektir ve bu yüzden çıkış voltajı da buna bağlı olarak değişecektir!

Elemanlar:	Gerçek Görünüm:		
Foto Direnç			f5 f6
LED (5mm)			h20-h21 +
330Ω Direnç (sensör)			i21 -
10KΩ Direnç			i1 i5
Atlama Kablosu			i1 +
Atlama Kablosu			A0 i5
Atlama Kablosu			i6 -
Atlama Kablosu			Pin 9 i20
Atlama Kablosu			5V +
Atlama Kablosu			GND -



Kod Notları:



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

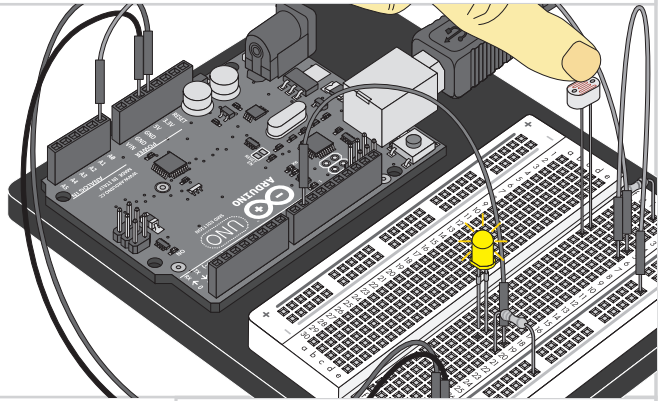
⇒ analogRead() kullanarak okuduğumuz bir analog sinyal, 0-1023 arasında bir değer olacaktır. Fakat analogWrite() kullanarak bir PWM çalıştırmak istediğimizde, 0-255 arasında bir değer isteyecektir. Bu durumda map() fonksiyonunu kullanarak geniş alanları daha dar alanlar olacak şekilde sıkıştırabiliriz.

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

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

Ne göreceğiz?

Fotodirençinizin algıladığı ışık miktarına göre LED'inizi 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.

Işığın Değişimine Tepki Vermiyor

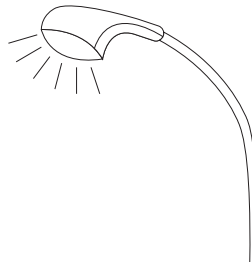
Fotodirenç üzerinde bağlantılar standart değildir, bu yüzden bağlantıları karıştırmış olmanız mümkün. Doğru yerleştirildiğinden emin olmak için iki kez kontrol edin.

Hala Çalışmıyor

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

Gerçek Hayatta Uygulamaları:

Sokak lambaları geceleri aydınlatmak için bu tür sensörler kullanır.

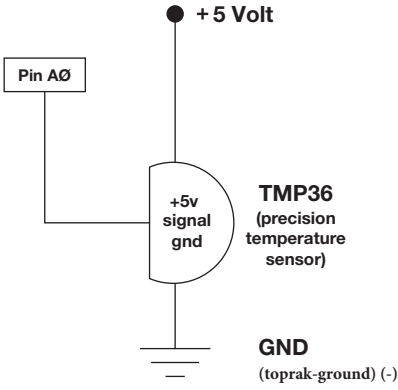


Sıcaklık Sensörü

Sıcaklık sensörü adından da anlaşıldığı 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.

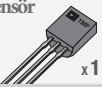


Transistör ve Sıcaklık Sensörü birbirleri çok benzeyen devre elemanlarıdır. Karıştırılmamaya dikkat edilmelidir.

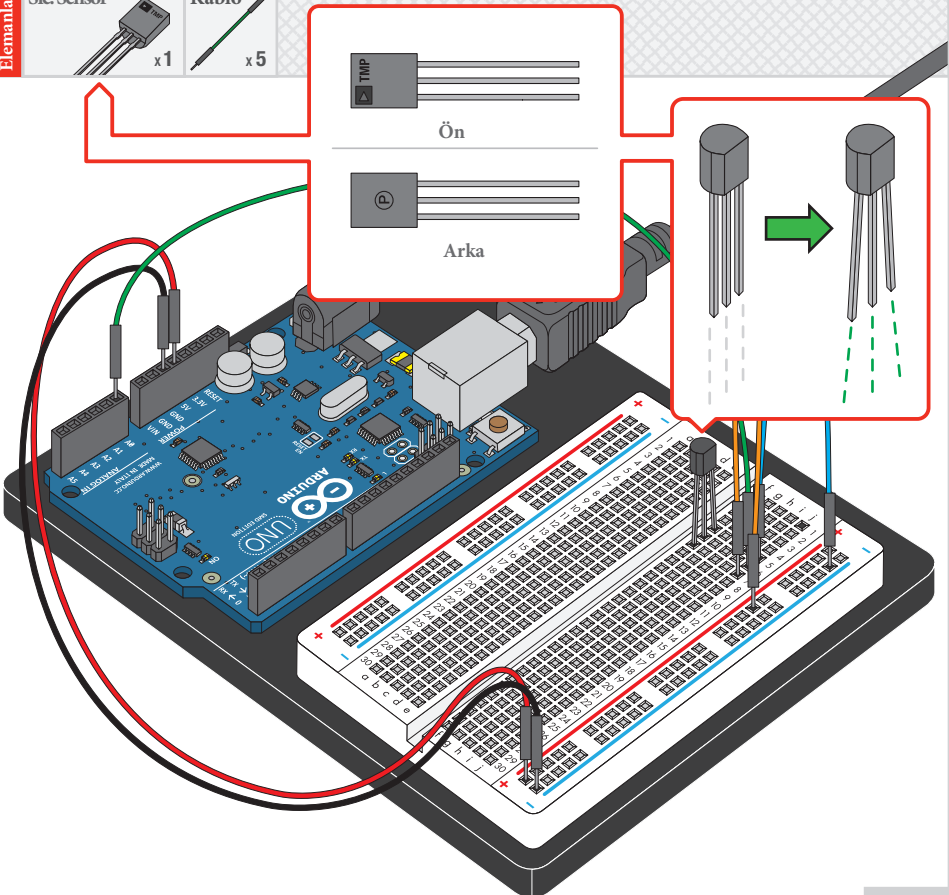
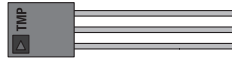


Elemanlar:

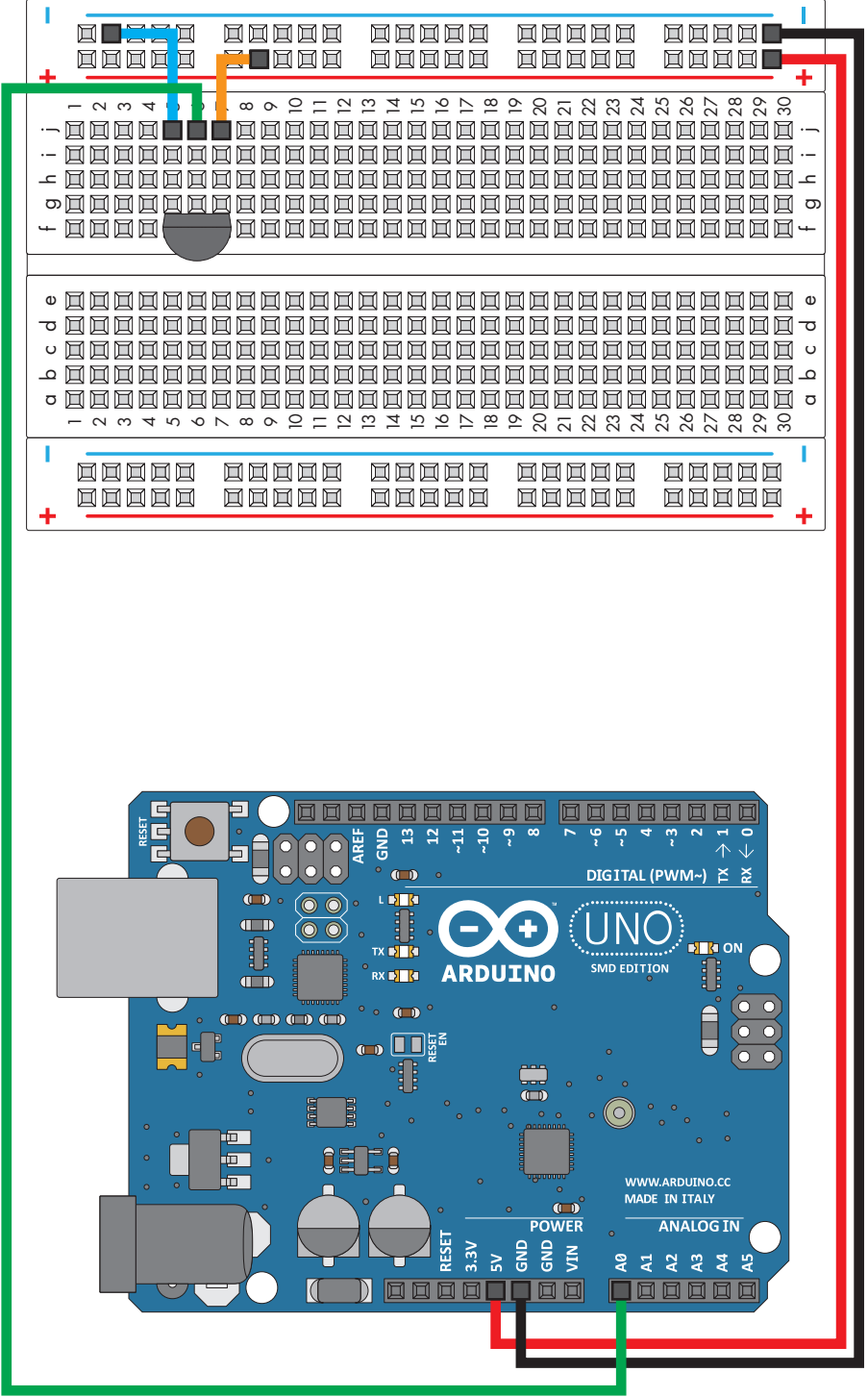
Sıc. Sensör



Kablo



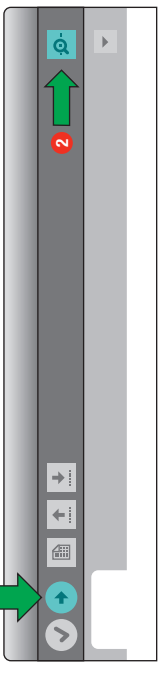
Devre 7: Sıcaklık Sensörü



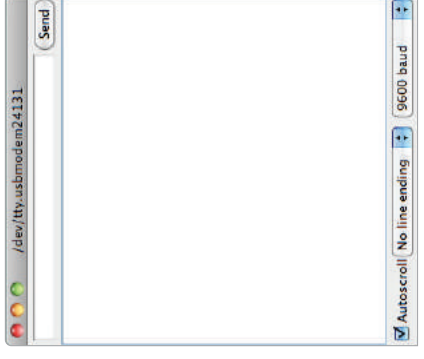
Serial Monitörü Açıyoruz

Bu devre Arduino IDE'nin seri monitörünü kullanıyor. Bunu açmak için öncelikle programı upload edip kare kutu içerisinde büyüteç gibi görünen butona basıyoruz.

1



3



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



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

Kod Notları:



Serial.begin(9600);



Seri monitörü kullanmadan önce, başlatmak için Serial.begin() komutunu çağırmanız gerek. Bağlantı hızı veya "Baud Hızı" 9600'dür. İki cihaz birbiriyle bağlantı kurduğunda, ikisi de aynı hıza 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>

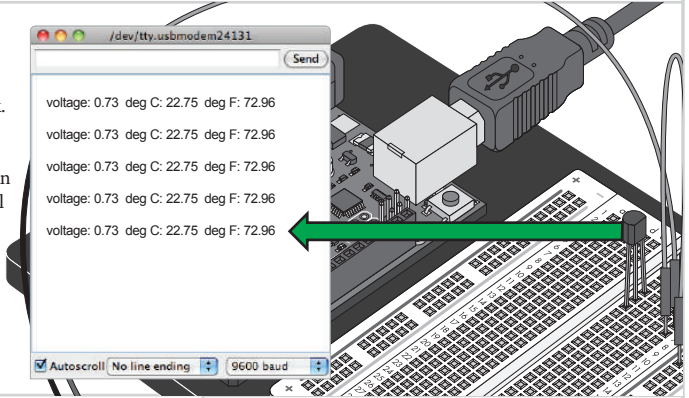
Serial.println(degreesF);



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 gerek. 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 gerek. (Talimatlar önceki sayfada mevcut.)

Anlamsız Ekran

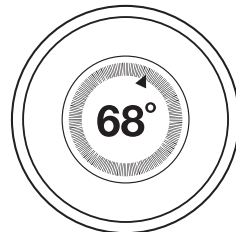
Bu tür şeylerin meydana gelmesinin sebebi seri monitörün beklenenden farklı hızda data almasından kaynaklanıyor. Düzeltmek için ise pull-down kutucuğuna tıklayıp "baud" yazan yeri "9600 baud" olarak düzeltmeniz gereklidir.

Sıcaklık Değeri Değişmiyor

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

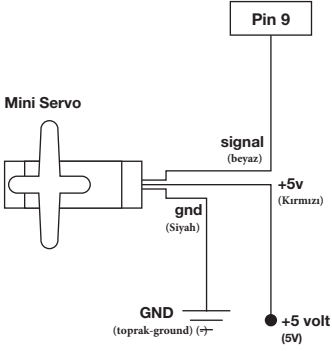
Gerçek Hayatta Uygulamaları:

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



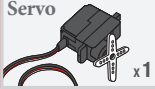
Servo Motor

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



Elemanlar:

Servo

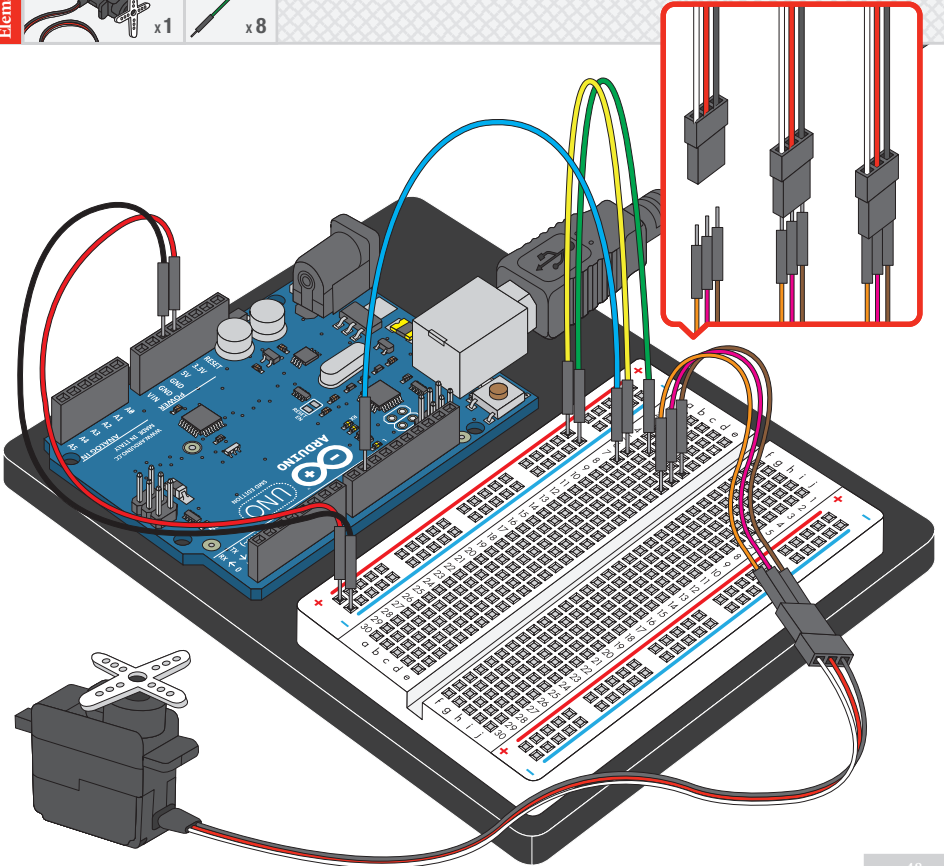


x 1

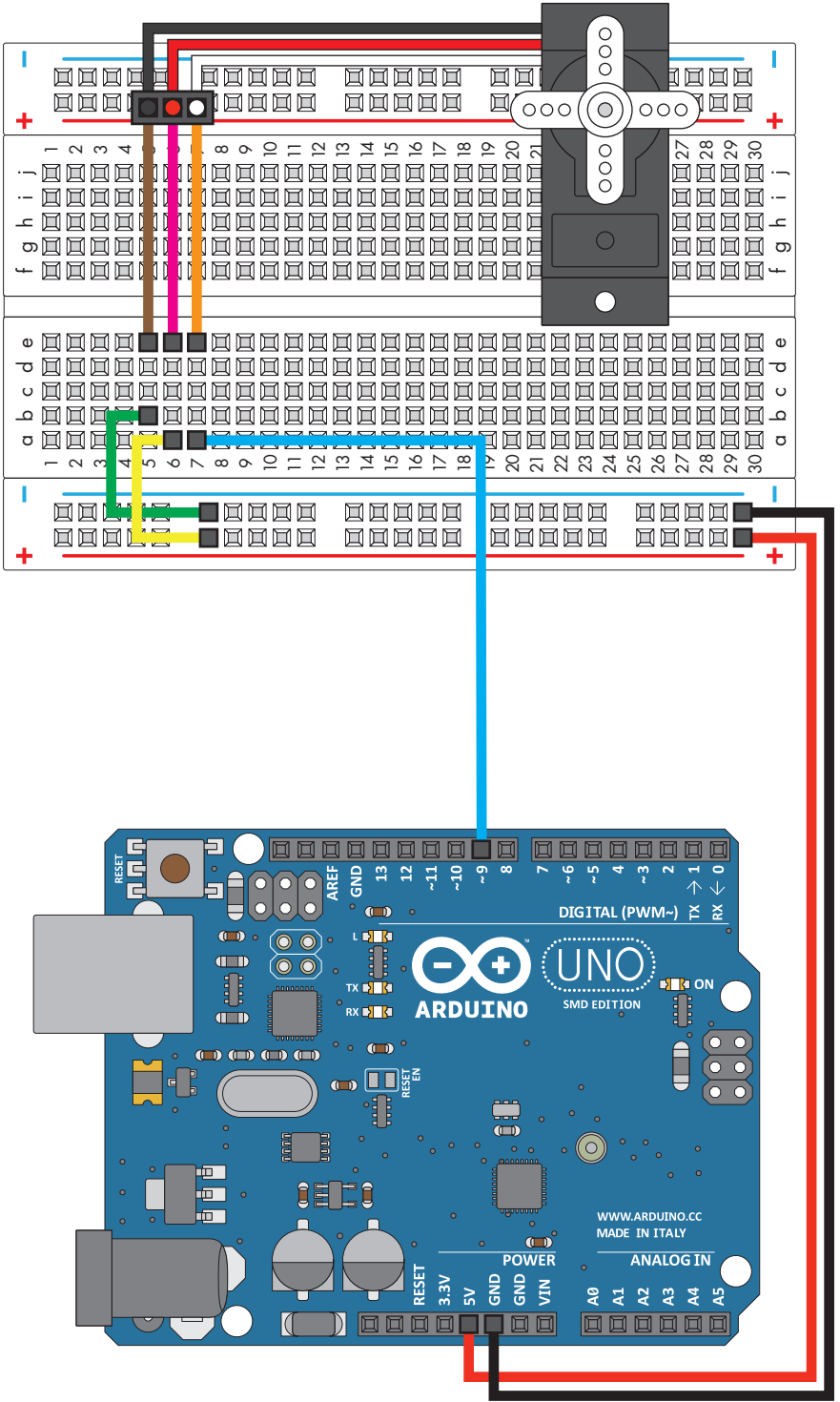
Kablo



x 8



Devre 8: Servo Motor



Kütüphaneler Kullanarak Ufkunuzu Genişletin

Arduino sağladığı kullanışlı dahili komutlarla; basit giriş(input) ve çıkış(output) işlemleri yapmanızı, mantık kullanarak karar vermenizi, matematik problemleri çözmenizi sağlar. Arduino'nun asıl gücü ise bu platformu kullanan devasa toplulukların yaptıkları çalışmalarını paylaşmalarıdır. Arduino bir çok kullanışlı kütüphaneye sahiptir. Bu örnekte kullandığımız servo kütüphanesi bunlardan biridir. Standart kütüphaneler ve kullanım kılavuzları için siteyi ziyaret edebilirsiniz.

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

Herkes kendi kütüphanesini oluşturabilir, eğer yeni bir sensör veya çıktı(output) cihazı kullanacaksanız yeni kütüphane yazabilirsiniz, tabii şansınıza 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'nuzu aldığımızda yeni bir cihazla kullanacaksınız, kütüphanenizi oluşturun ve bunu bütün dünyayla paylaşın! Kütüphaneyi taslaқта (sketch) kullanmak için **Sketch > Import Library** kısmından seçiniz.
















File Edit Sketch Tools Help

Verify / Compile

Show Sketch Folder
Add File...

Import Library

EEPROM
Ethernet
Firmata
LiquidCrystal
SD
Servo
SoftwareSerial
SPI
Stepper
Wire

Elemanlar:	Gerçek Görünüm:	İkon	İkon
Servo Motor			<input type="checkbox"/> e5 <input type="checkbox"/> e6 <input type="checkbox"/> e7
Atlama Kablosu			<input type="checkbox"/> e5
Atlama Kablosu			<input type="checkbox"/> e6
Atlama Kablosu			<input type="checkbox"/> e7
Atlama Kablosu		<input type="checkbox"/> Pin 9	<input type="checkbox"/> a7
Atlama Kablosu			<input type="checkbox"/> b5
Atlama Kablosu			<input type="checkbox"/> a6 <input type="checkbox"/> +
Atlama Kablosu		<input type="checkbox"/> 5V	<input type="checkbox"/> +
Atlama Kablosu		<input type="checkbox"/> GND	<input type="checkbox"/> -



Arduino IDE 'yi Aç// File > Examples > Arduino Kod > Devre # 8

Kod Notları:



#include <Servo.h>



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

Servo servo1;



Servo kütüphanesi servoyu kontrol etmenizi sağlayan yeni komut imkanı 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.attach(9);

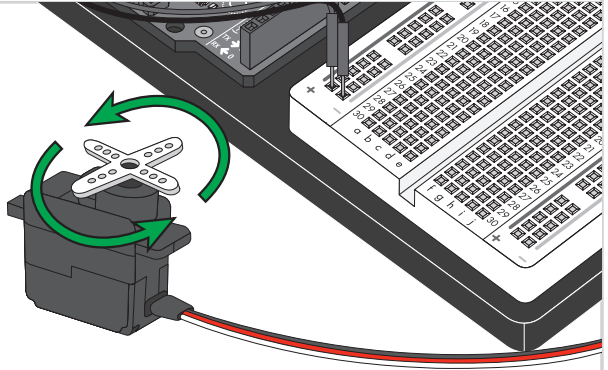
servo1.write(180);



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

Ne göreceğiz?

Servo motorunuzun farklı hızlarda değişik konumlarda dönebildiğini görebilirsiniz. Şayet motorunuz hareket etmiyorsa bağlantılarınızı kontrol ettikten 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 şaşırtıcı şekilde motorun ters bağlanmış olması mümkün olabilir. Muhtemelen sorunuz bundan kaynaklıdır.

Çalışmamakta Israr Ediyor:

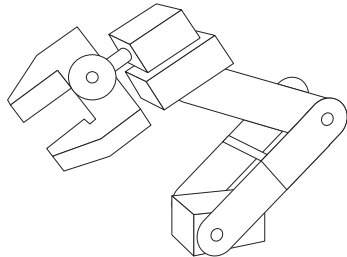
Muhtemelen (kırmızı ve kahverengi kablolar) 5 Volt ve toprak(gnd) bağlantısı yanlış yapılmıştır.

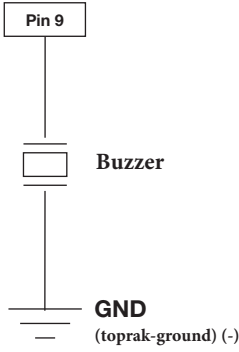
Oldu Bitti :

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

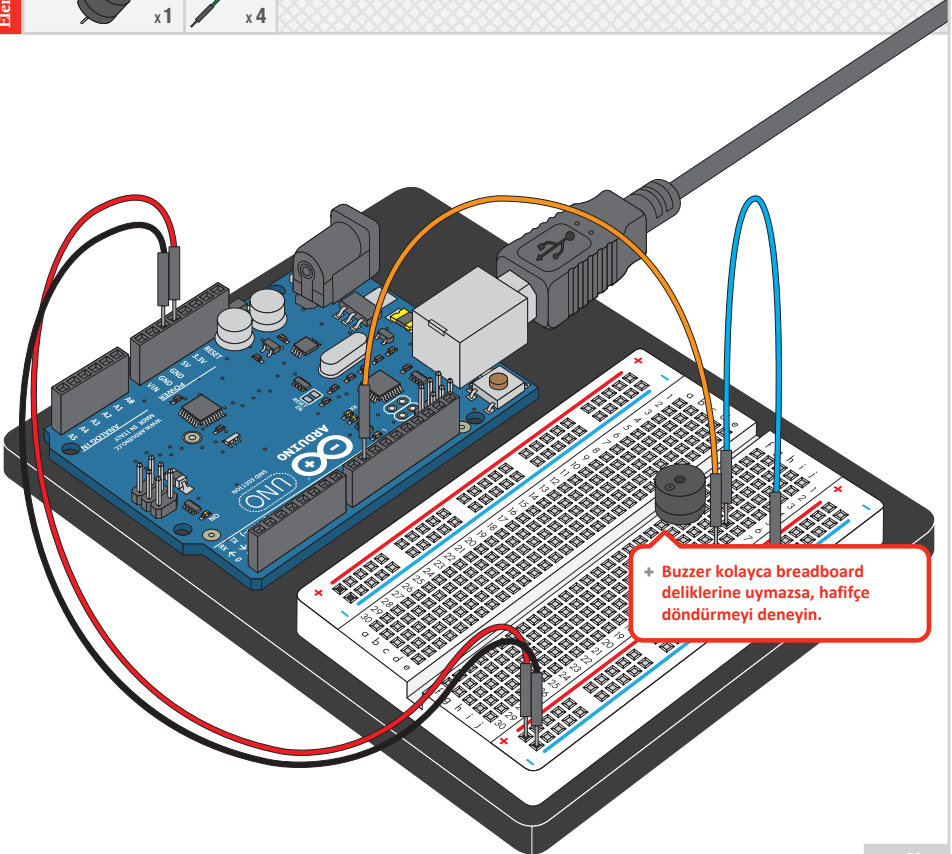


x1

Kablo

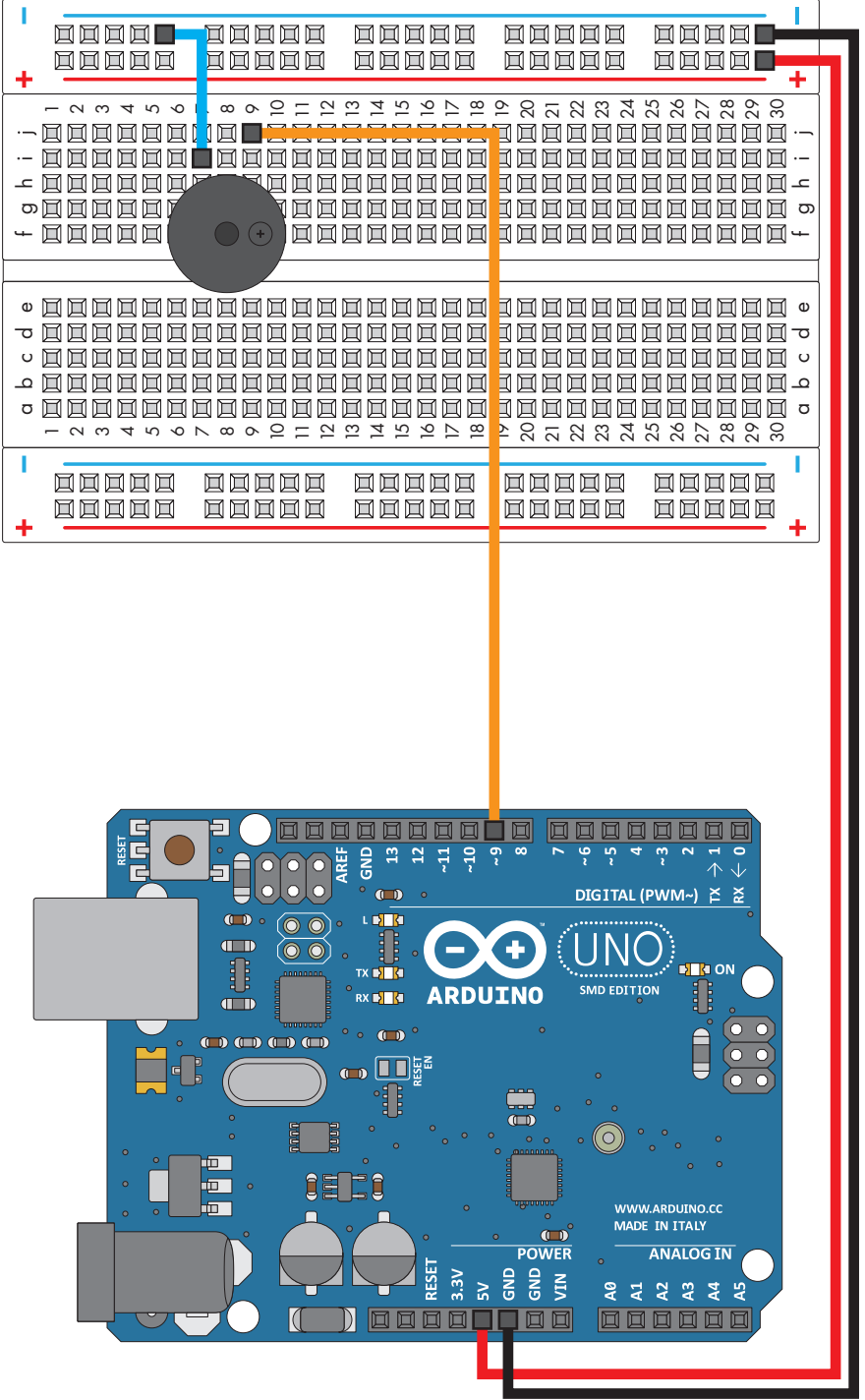


x4



✦ Buzzer kolayca breadboard deliklerine uymazsa, hafifçe döndürmeyi deneyin.

Devre 9: Buzzer













Kendi Fonksiyonları Oluşturma

Arduino içerisinde her türlü şey için kullanılabilcek kocaman bir servet barındırıyor. (Bknz: <http://arduino.cc/en/Reference>) Bunun yanı sıra kendi fonksiyonlarımızı oluşturmak da oldukça basit. Örnek verecek olursak “add” iki sayıyı bir toplayıp size sonucu verir.

```
int add(int parameter1, int parameter2)
{
    int x;
    x = parameter1 + parameter2;
    return(x);
}
```

Fonksiyonunuz bir değeri alabilir. (“parametre” ve başka bir değere dönüştürülebilir. Eğer fonksiyonuza bir parametre atamak isterseniz, fonksiyondan sonra gelen parantez içerisinde bu değeri yazabilirsiniz. Eğer parametre vermeyecekseniz parantez içeriğini boş bırakabilirsiniz. Eğer fonksiyonunuzdan bir değer çekecekseniz değer tipini fonksiyon isminizin önüne yazın. Değeri çekmeye hazırsanız **return()** komutu ekleyin. Eğer değer döndürmeyecekseniz fonksiyon isminizin önüne “**void**” ekleyin. (**setup()** ve **loop()** fonksiyonlarında görmüş olduğunuz gibi) Kendi fonksiyonunuzu yazdığımızda kodunuzu düzenli ve kullanımı kolay bir hale getirin.

Elemanlar:	Gerçek Görünüm:	
Buzzer		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		
Atlama Kablosu		



Kod notları:



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



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

Şimdiye kadar sadece sayısal verilerle çalıştık ama Arduino aynı zamanda metinlerle çalışabilir. Karakterlerin (tek karekterler, harfler, numaralar, semboller) kendilerine ait tipleri mevcuttur.

Bu tipe “char” diyoruz. Bir karakter dizinin varsa çift tırnak içerisinde tanımlanabilir. “string” olarak tanımlarsanız tek tırnak içerisinde belirtilmelidir.

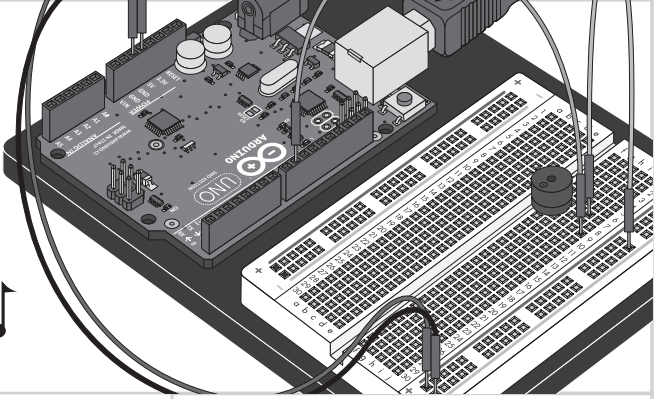
```
tone(pin, frequency, duration);
```



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

Ne göreceğiz?

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



Sorun Giderme:

Ses Yok

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

Müzik Çalarken Düşünemiyorum

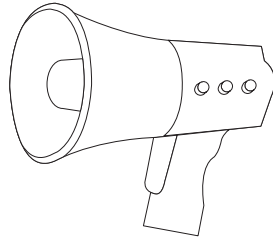
O halde düşünürken cihazınızı sökün, kullanacağınız zaman programı upload edip tekrar monte edin.

Twinkle Twinkle Little Stars Şarkısından Sıkıldım

Nasıl yapacağınızı öğrendiğimize göre artık kendi şarkılarınızı upload edebilirsiniz.

Gerçek Hayatta Uygulamaları:

Modern megafonlarda güçlendirilmiş buzzerlar kullanılıyor. Gerçekten fazla gürültülü olmalarına rağmen insanların ilgisini çekmekte yeterince etkili.

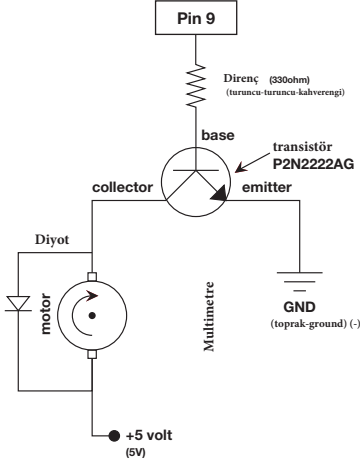


Motor Döndürmek

Servo motorlarla oynadığınız zamanlardan öncesini hatırlayın. Şimdi biraz da motorun dönüşüyle uğraşacağız. Bunun için transistöre ihtiyacımız var çünkü transistörler Arduino'nun yapabildiğinden daha büyük miktarlarda akım dönüştürebiliyor. Transistör kullandığınızda maximum seviyesine bakarak kullanımınız için yeterince uygun olup olmadığına karar verin. Bu devre için 40V ve 200 mA'lık transistör işimizi görür. Oyuncak motorumuz için yeter de artar!



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



Elemanlar:

Transistör
P2N2222AGDiodyot
1N4148

DC Motor

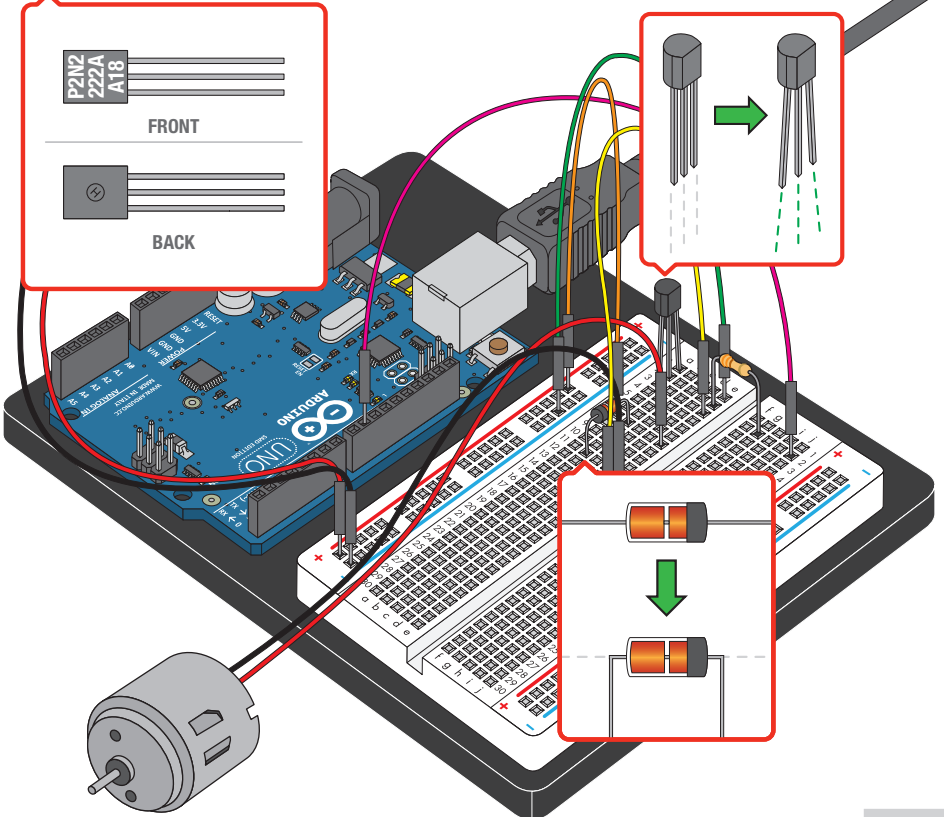
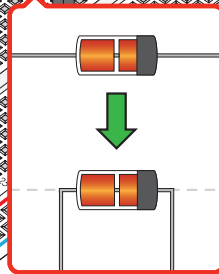
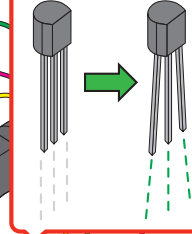


Kablo

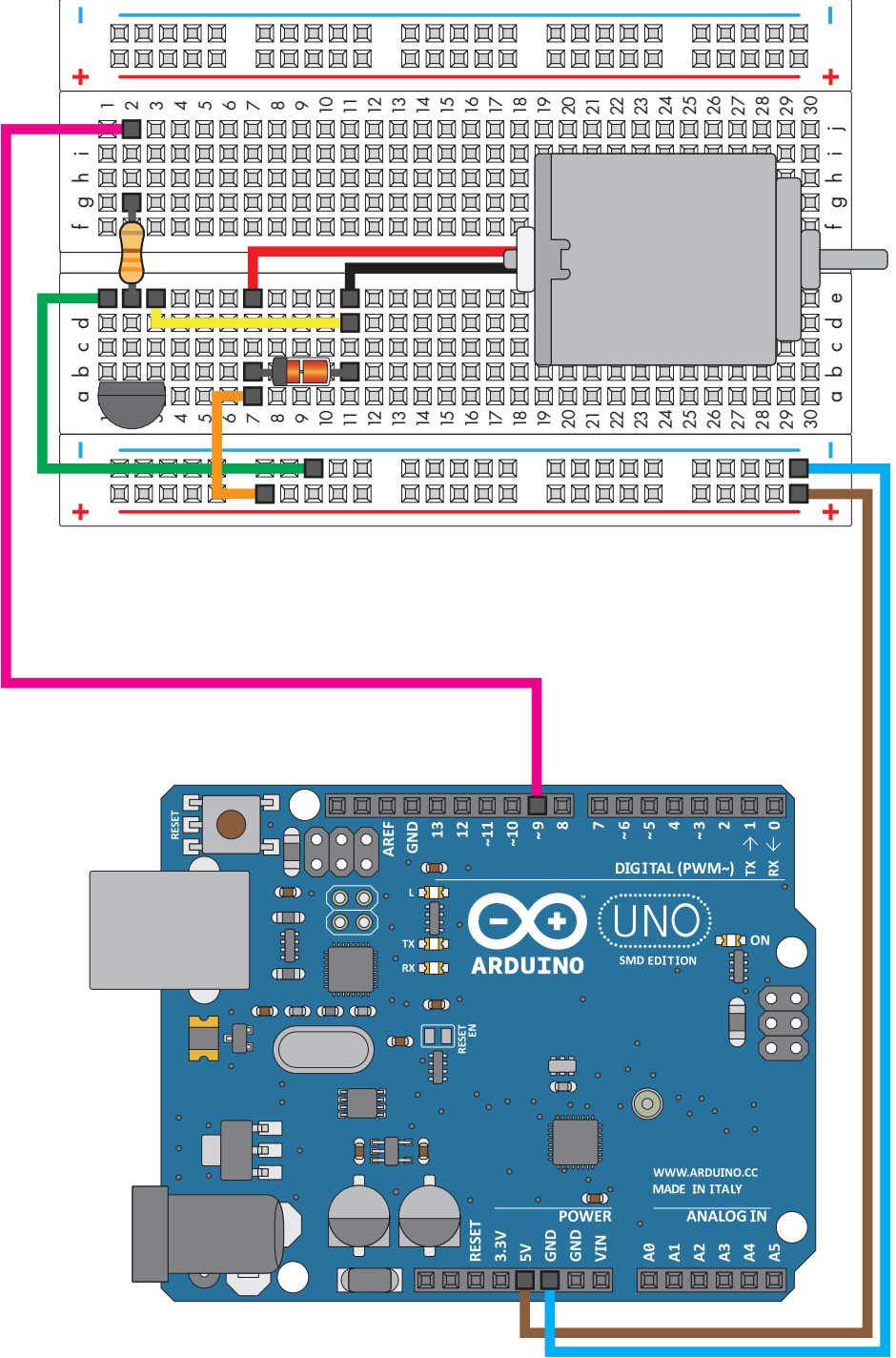
330Ω
DirençP2N2
222A
A18


















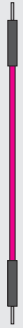


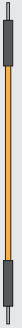











FRONT

BACK



Devre 10: Motor Döndürmek



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

Hepsini bir araya getirirsek:

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

Yazdığımız bir çok taslak (sketch) aşağıdakilerin bir kaçını veya hepsini içerecek:

1. **input denemesi yapın.**
2. **Bazı hesaplamalar yapın ve karar verin.**
3. **Output almaya çalışın.**
4. **Tekrar edin!(Veya etmeyin size kalmış,)**

Birçok input sensörü ve output cihazının nasıl çalıştığı hakkında epey bilgi verdik. (Bir kaç cihazımız ve sensörümüz daha var tabii) Artık öğrendiklerinizi kendi taslağınızda özgürce kullanabilirsiniz. İşte bu "Açık Kaynak" akımının ardında yatan fikrin ta kendisidir. Farklı tasarlardan(sketch) parçaları toplamak ve bunları birleştirip yeni şeyler üretmek fazlasıyla kolay, tek yapmanız gereken iki adet pencere açmak ve birinden diğerine kopyalayıp yapıştırmak. Bu yüzden "iyi programlama alışkanlığı" fikrimizi aşıyoruz. Aynı pin numaralarını kullanmak ve taslağınızı fonksiyonlara ayırmak, kodunuzun yeniden kullanılabilirliğini kolaylaştırır. Örneğin bir kodun iki bölümü için aynı pini kullanıyorsanız kolaylıkla ikisinden birinin pinini değiştirebilirsiniz.(Şunu unutmayın ki bütün pinler analogWrite() desteğine sahip değil, uyumlu olanlar devre kartımızda işaretli bulunuyor.)

Eğer yardıma 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. İşleri ilerletmeye hazır olduğumuzda daha ileri seviye konular için arduino.cc/en/Tutorial sayfasından Arduino Tutorial sayfasına uğrayabilirsiniz. Sonunda, havalı bir şeyler ürettiğiniz vakit, bunu bütün dünyaya paylaşım ki bütün dünya dahilliğimizden faydalanabilsin. (Ve bizim bundan haberdar olmamızı sağalayın ki bunu anasayfamızda paylaşabilelim!)



Arduino IDE 'yi Aç// File > Examples > Arduino Kod > Devre # 10

Kod Notları:



while (Serial.available() > 0)



Arduino seri portları data gönderimi için olduğu gibi data alımı için de kullanılabilir çünkü herhangi bir zaman da data aktarımı olabilir. Arduino siz bu bilgiyi kullanana kadar depolar ve korur. The Serial.available() komutu portunuza ulaşmış fakat taslağınızda (sketch) henüz kullanılmamış olan karakter numaralarını geri getirir. Sıfır, ulaşmış data yok anlamına gelir.

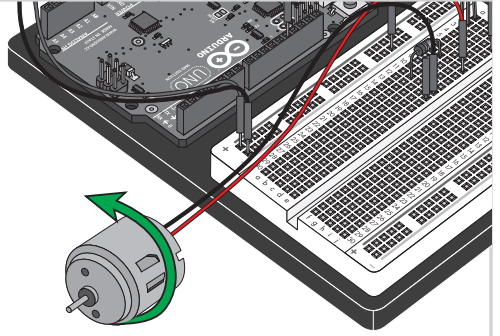
speed = Serial.parseInt();



Portunuzda bekleyen hali hazırda datanız mevcut ise, kullanabileceğiniz bir kaç yöntem var. Port içerisine 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 gerek. Eğer çalışmıyorsa arıza tespit kısmını kontrol edin.



Sorun Giderme:

Motor Dönmüyor

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

Çalışmamaya Devam Ediyor

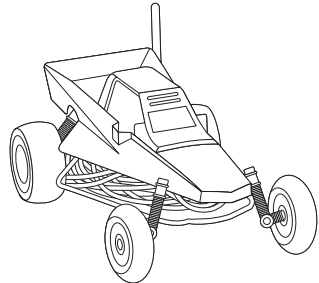
Kendi motorunuzu kullanıyorsanız motorunuzu 5V ile çalıştırmayı deneyin ve daha fazla güç çekip çekmediğini kontrol edin.

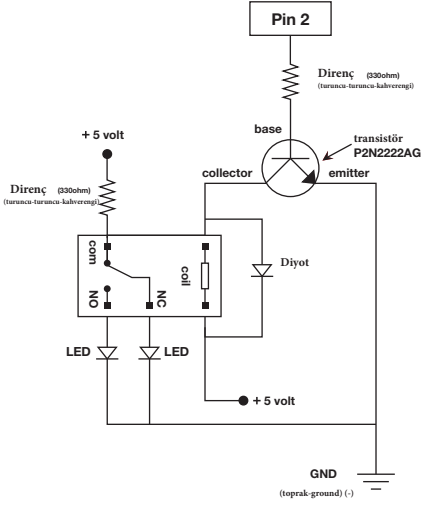
Çalışmamakta Israrlı

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

Gerçek Hayatta Uygulamaları:

Radio Kontrollü RC arabalar, DC motor kullanılır.





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 anahtar açıyor. Bu devrede Arduino'nuzda daha güçlü yetenekler kazandırmayı ve bir röleyi profesyonelce kullanmayı öğreneceksiniz.



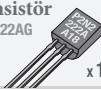
Röle kapalıyken, COM(common) pini NC (Normally Closed) pinine bağlanır.
Röle açıkken, COM(common) pini NO (Normally Open) pinine bağlanır.

Elemanlar:

Röle



x1

Transistör
P2N2222AG

x1

Diodyot
1N4148

x1

330Ω
Direnç

x2

LED

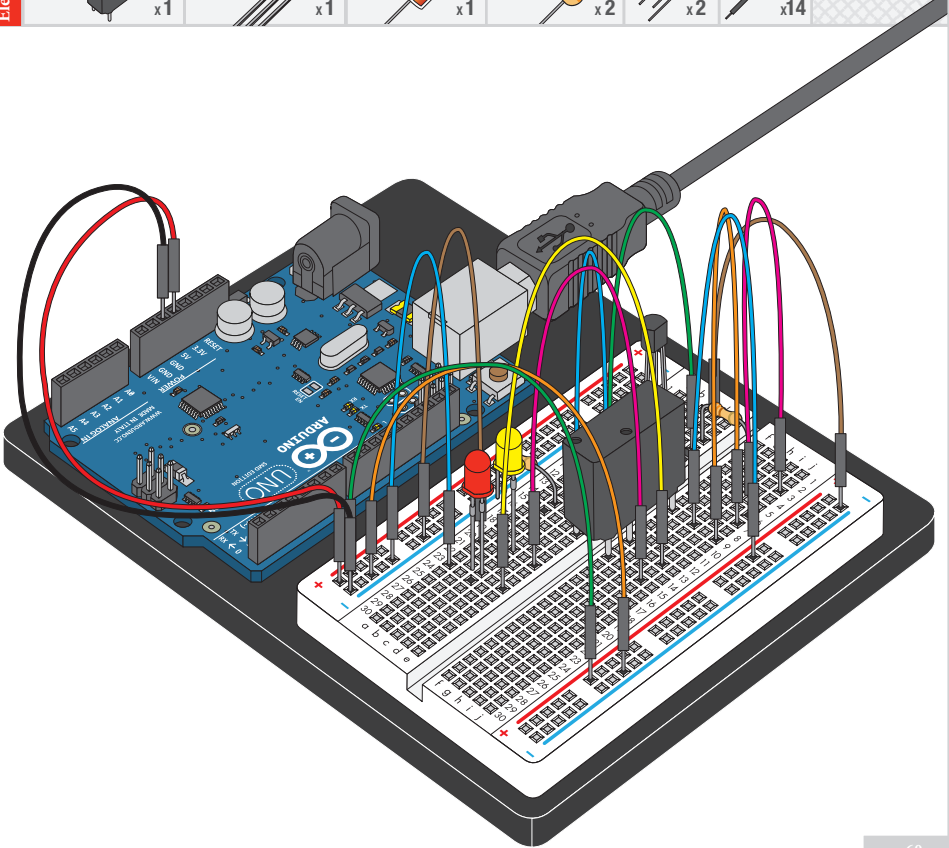


x2

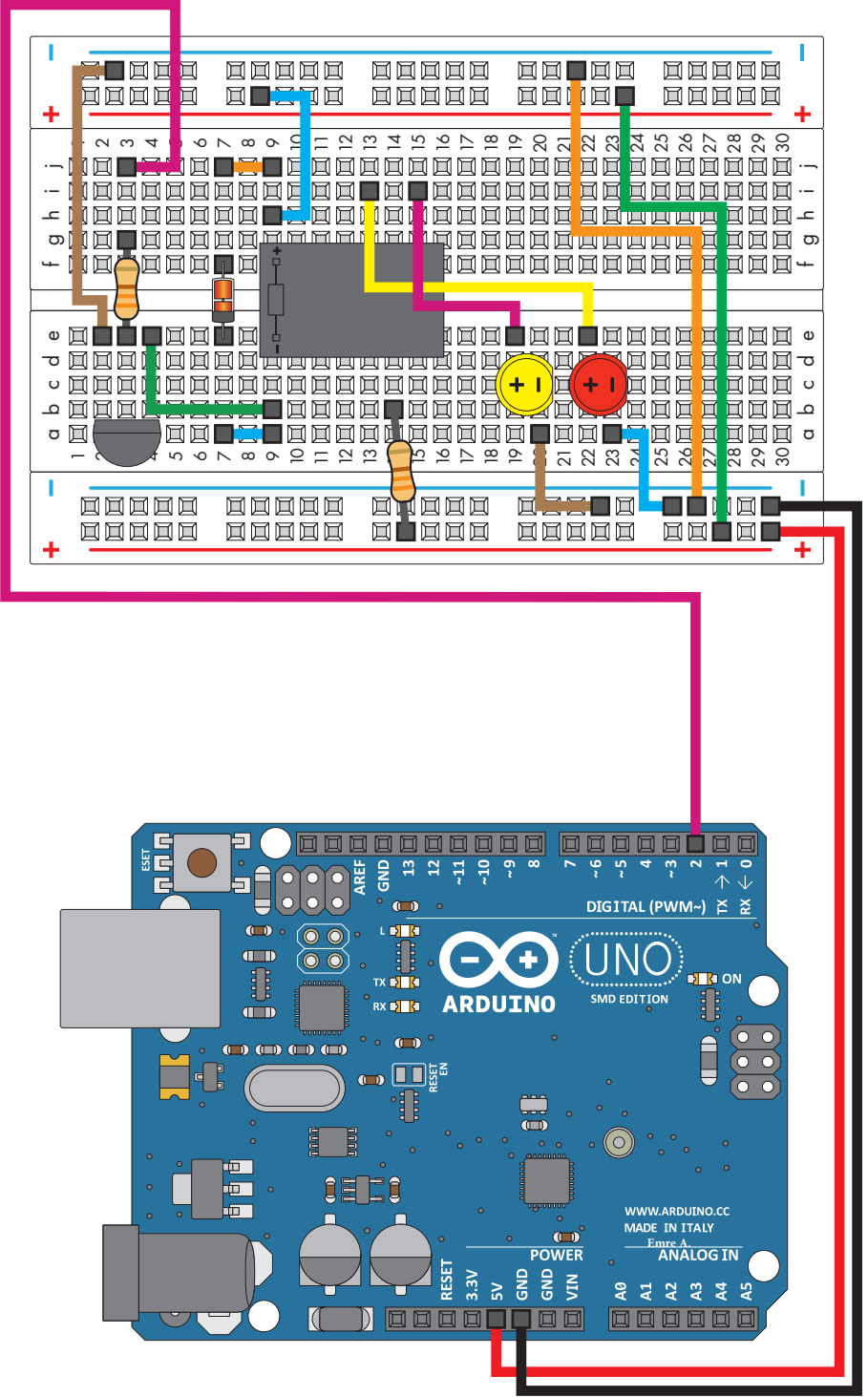
Kablo

































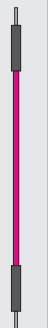

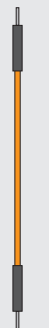





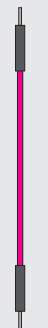



x14



Devre 11: Rôle



Elemanlar:	Gerçek Görünüm:	Pin 2	Elemanlar:	Gerçek Görünüm:	5V	GND
Röle			Atlama Kablosu			
Transistör P2N2222AG			Atlama Kablosu			
LED (5mm)			Atlama Kablosu			
LED (5mm)			Atlama Kablosu			
Diyot 1N4148			Atlama Kablosu			
330Ω Direnç			Atlama Kablosu			
330Ω Direnç			Atlama Kablosu			
Atlama Kablosu			Atlama Kablosu			
Atlama Kablosu						
Atlama Kablosu						
Atlama Kablosu						
Atlama Kablosu						
Atlama Kablosu						



Arduino IDE'yi Aç// File > Examples > Arduino Kod > Devre # 11

Kod Notları:



digitalWrite(relayPin, HIGH);



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

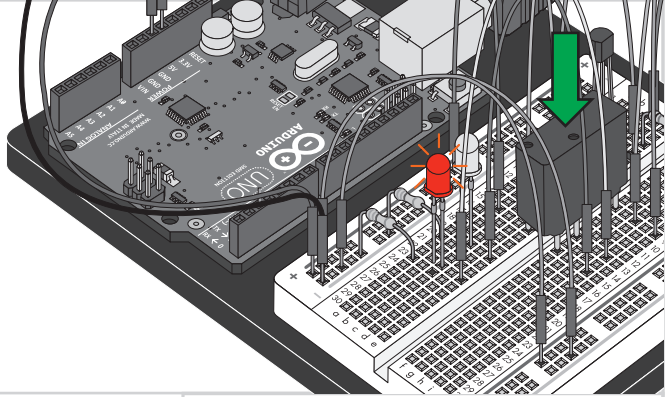
digitalWrite(relayPin, LOW);



Röle NC denen ilave bir bağlantıya sahiptir. Röle kapalı olduğunda NC pini COM pini ile bağlantı kurar. Rölenin açık veya kapalı olmasına bağlı olarak iki pinden birini kullanabilirsiniz. Aynı zamanda bu iki pini iki cihazın enerjisini değiştirmek için de kullanabilirsiniz.

Ne göreceğiz?

Röle bağlantı klik sesini duymuş olmanız gerek ve ardından 1 saniyelik aralıkta LED'leri aydınlıklarını kendi aralarında değiştirdiklerini görebilirsiniz. Eğer devreniz çalışmıyorsa devre kurulumunuzu kontrol edin, kodlarınızı tekrar upload edin.



Sorun Giderme:

LED Işık Vermiyor

LED'i doğru taktığınızdan emin olun. Uzun olan uç pozitif uçtur.

Klik Sesi Gelmiyor

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

Çalışmamakta İsrarcı

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

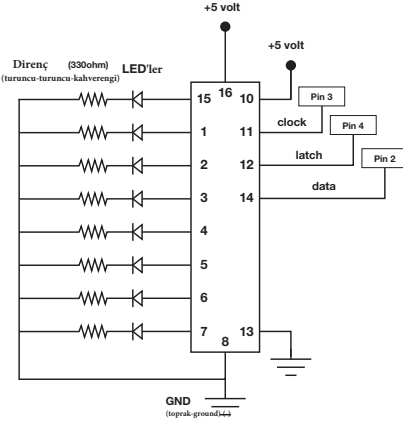
Gerçek Hayatta Uygulamaları:

Garaj kapılarını açmak için röle 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 Arduino'nuzaya ilaveten 8 output verir ve Arduino'nun sadece 3 pini kullanılır. Bu devrede 8 adet LED'i kontrol etmek için shift register kullanacağız.

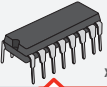


Q_b	1	16	V_{cc}
Q_c	2	15	Q_a
Q_d	3	14	SER
Q_e	4	13	OE
Q_f	5	12	RCLK
Q_g	6	11	SRCLK
Q_h	7	10	SRCLR
GND	8	9	Q_i

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

Elemanlar:

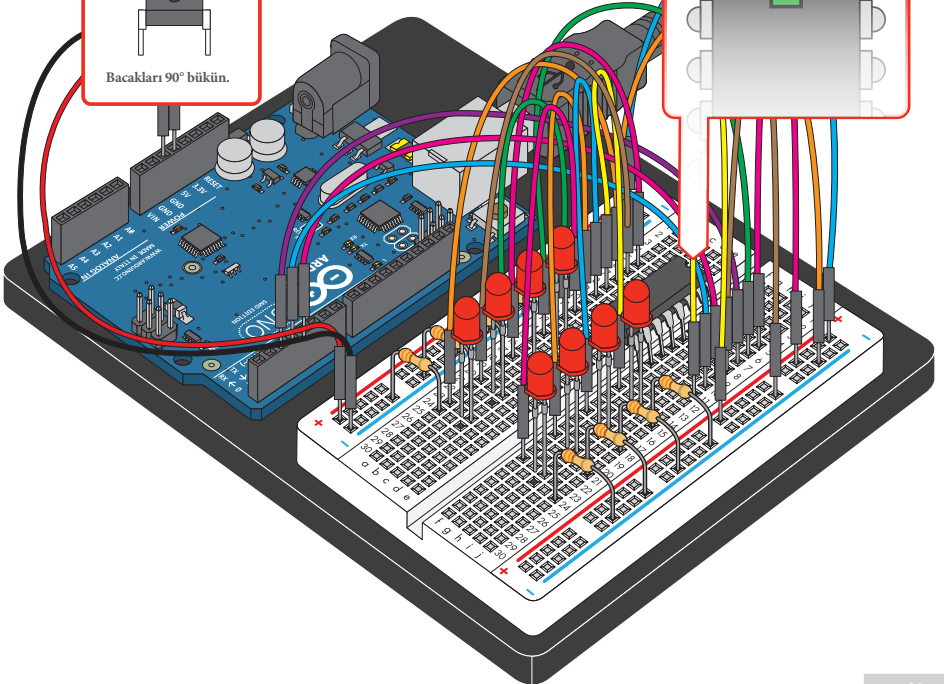
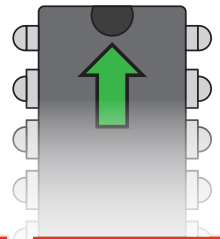
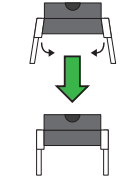
IC



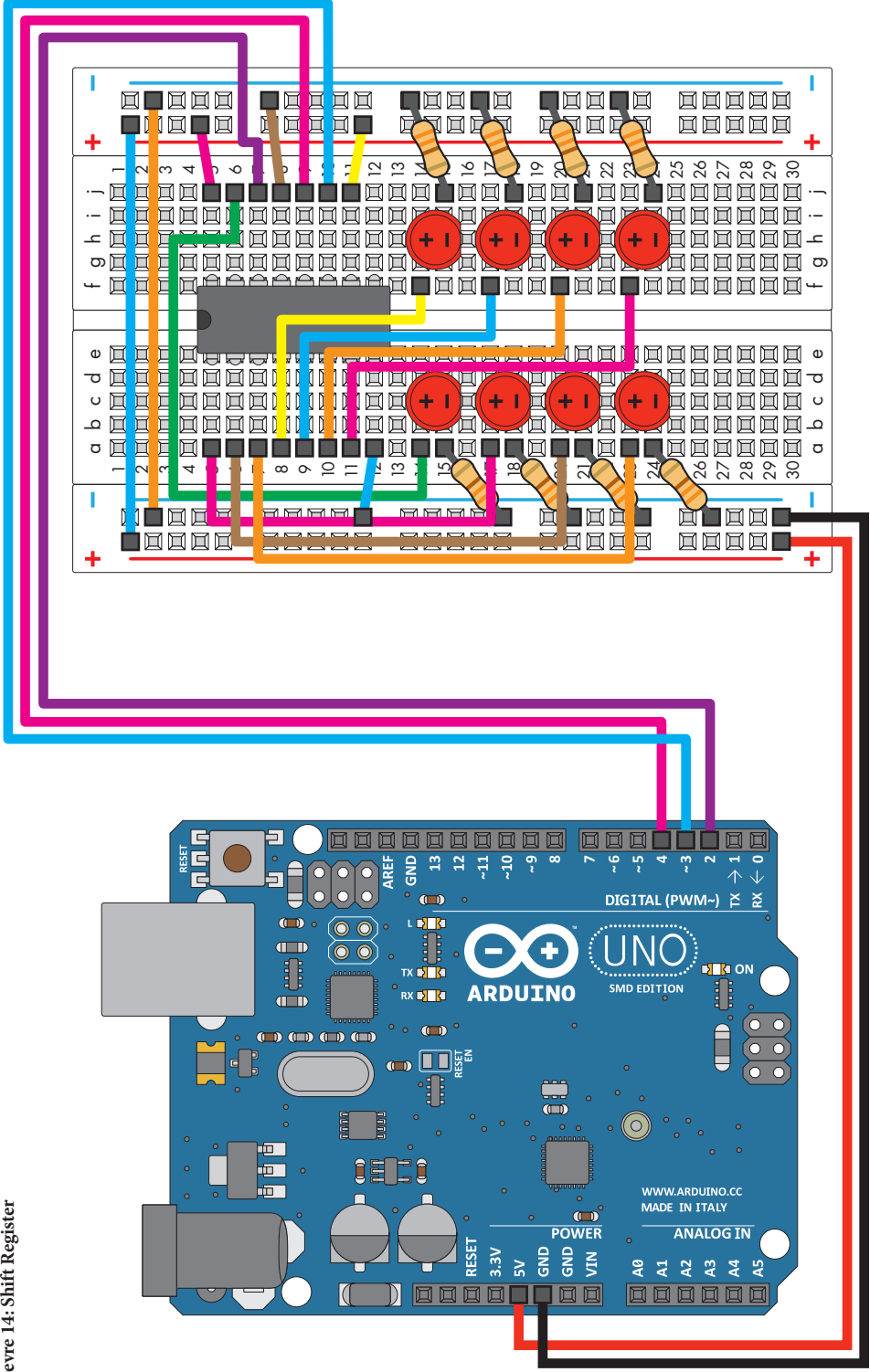
LED

330Ω
Direnç

Kablo



Devre 14: Shift Register



Elemanlar:	Gerçek Görünüm:	Gerçek Görünüm:	Elemanlar:	Gerçek Görünüm:	Gerçek Görünüm:
IC			Atlama Kablosu		
LED (5mm)			Atlama Kablosu		
LED (5mm)			Atlama Kablosu		
LED (5mm)			Atlama Kablosu		
LED (5mm)			Atlama Kablosu		
LED (5mm)			Atlama Kablosu		
LED (5mm)			Atlama Kablosu		
LED (5mm)			Atlama Kablosu		
330Ω Direnç			Atlama Kablosu		
330Ω Direnç			Atlama Kablosu		
330Ω Direnç			Atlama Kablosu		
330Ω Direnç			Atlama Kablosu		
330Ω Direnç			Atlama Kablosu		
330Ω Direnç			Atlama Kablosu		
330Ω Direnç			Atlama Kablosu		
330Ω Direnç			Atlama Kablosu		
Atlama Kablosu			Atlama Kablosu		



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

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);`

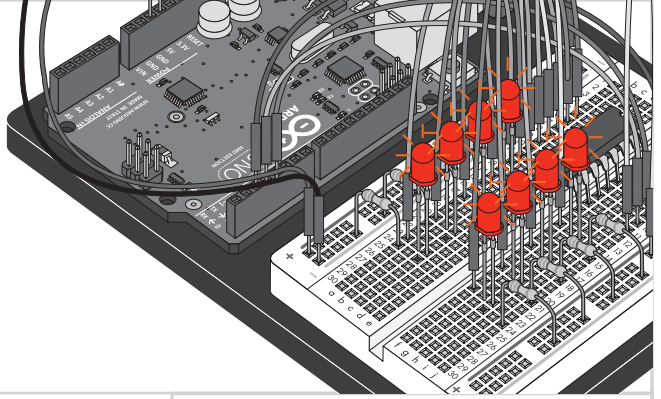


Bitler bilgisayar hafızasının en küçük yapı taşlarıdır, her bit 1 veya 0 depolayabilir. Daha büyük sayılar bitlerden oluşan dizilerle depolanır. Bazen bu bitler üzerinde oynama yapmak isteriz. Örneğin 8 bitli shift register'a gönderiyoruz ve LED'i açıp kapatması için 1 veya 0 yapmasını istiyoruz. Arduino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

`bitWrite(data,desiredPin,desiredState);`

Ne göreceğiz?

Devre 4' te olduğu gibi LED 'in yandığını görmemiz gerek (Tek fark shift register kullandık.) Eğer LED ışık vermiyorsa bağlantılarımızı kontrol edin ve kodunuzu tekrar upload edin.



Sorun Giderme:

LED Patladı

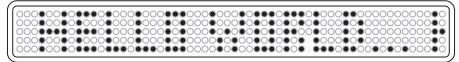
Bu bir çok kez başımıza geldi. Sebebi entegrenin ters takılmasından kaynaklanıyor. Eğer hemen düzeltirseniz hiçbir şeyi bozmadan durumu kurtarabilirsiniz.

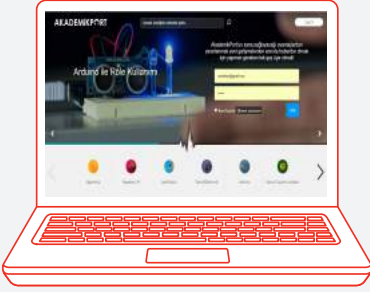
Çalışmamaya Devam Ediyor

Sürekli aynı şeyi söylüyoruz ama muhtelemen 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 adresini ziyaret edebilirsiniz. Bir sonraki eğitimde görüşmek üzere.

Emre ARSLAN

AkademikPort Kurumsal İlişkiler Koordinatörü

Kaynak

Sparkfun SIK GUIDE

Adafruit.com

Arduino - Coşkun Taşdemir

Instructables.com

NOTLAR



AKADEMIKPORT

www.akademikport.com



[/akademikport](https://www.facebook.com/akademikport)



[/akademikport](https://twitter.com/akademikport)



[/company/akademikport](https://www.linkedin.com/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 adresini ziyaret edebilirsiniz. Bir sonraki eğitimde görüşmek üzere.

Emre ARSLAN

AkademikPort Kurumsal İlişkiler Koordinatörü

Kaynak

Sparkfun SIK GUIDE

Adafruit.com

Arduino - Coşkun Taşdemir

Instructables.com

NOTLAR



AKADEMIKPORT

www.akademikport.com



[/akademikport](https://www.facebook.com/akademikport)



[/akademikport](https://twitter.com/akademikport)



[/company/akademikport](https://www.linkedin.com/company/akademikport)