





















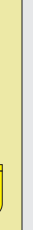













Elemanlar:	Gerçek Görünüm:	Elemanlar:	Gerçek Görünüm:
LED (5mm) 		330Ω Direnç	
LED (5mm) 		330Ω Direnç	
LED (5mm) 		330Ω Direnç	
LED (5mm) 		Atlama Kablosu	
LED (5mm) 		Atlama Kablosu	
LED (5mm) 		Atlama Kablosu	
LED (5mm) 		Atlama Kablosu	
LED (5mm) 		Atlama Kablosu	
330Ω Direnç		Atlama Kablosu	
330Ω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

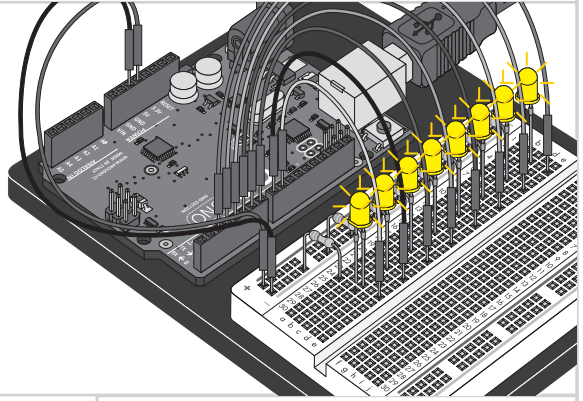


Bilgisayarlar çalışma sırasında aynı işlemleri gerçekleştirir fakat bazen bir şeylerin rastgele olmasını istersiniz, örneğin bir zar atma işleminde. Random() fonksiyonu bunu yapmak için en iyi yoldur. Daha fazla bilgi için adresi ziyaret ediniz.

<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

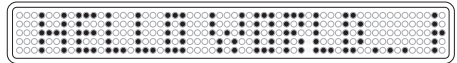
Sekiz adet bağlantı ile çalıştığınız için karışıklıklar olması gayet doğal. İlk LED'i pin 2'ye yerleştirin ve diğer LED'leri devam edecek şekilde yerleştirin ve tekrar kontrol edin.

Yeniden Başlayın

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

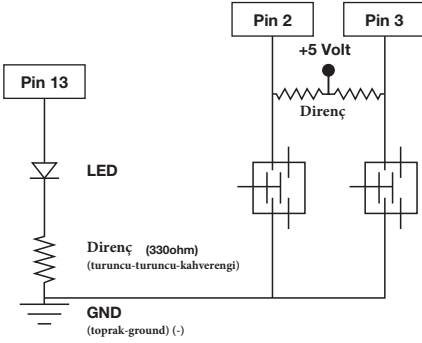
Gerçek Hayatta Uygulamaları:

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.



Elemanlar:

Push Button



x2

LED



x1

10KΩ



Direnç

x2

330Ω



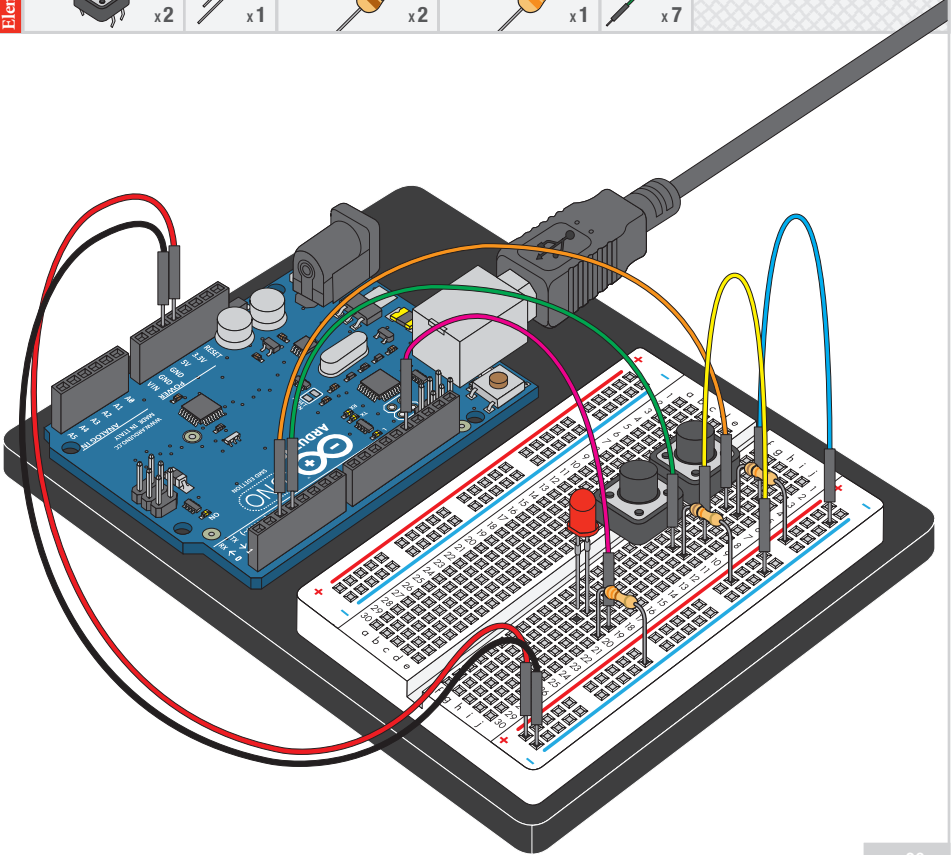
Direnç

x1

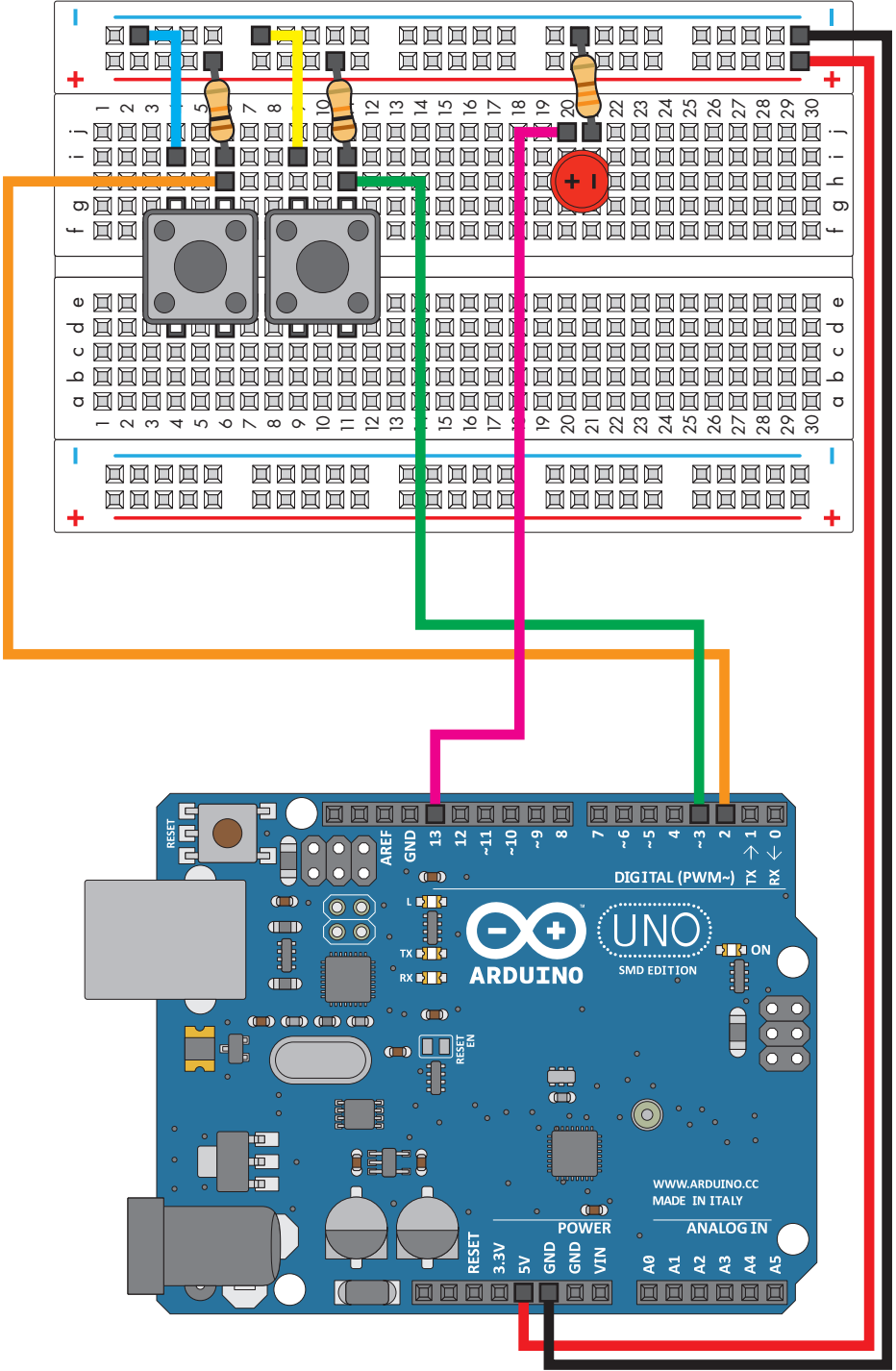
Kablo



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 kurdüğünüz 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 çalıştıracağıdır. Bu tarz işlemlerle Arduino'yu 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
Atlama Kablosu			h11
Atlama Kablosu			h20



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.

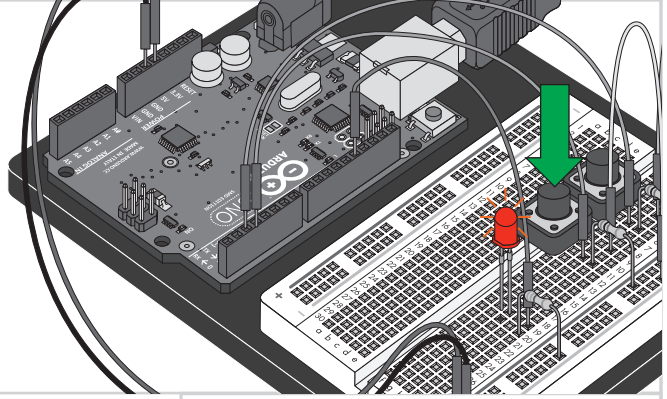
if (button1State == LOW)



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

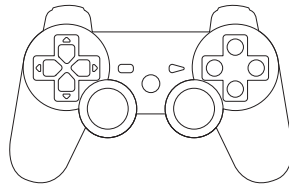
Butonuzum kare olduğu için yanlış yerleştirilmiş olabilir. 90 derece çevirin ve çalışıyor mu tekrar kontrol edin.

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.



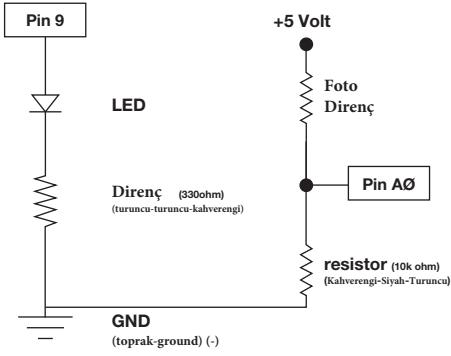
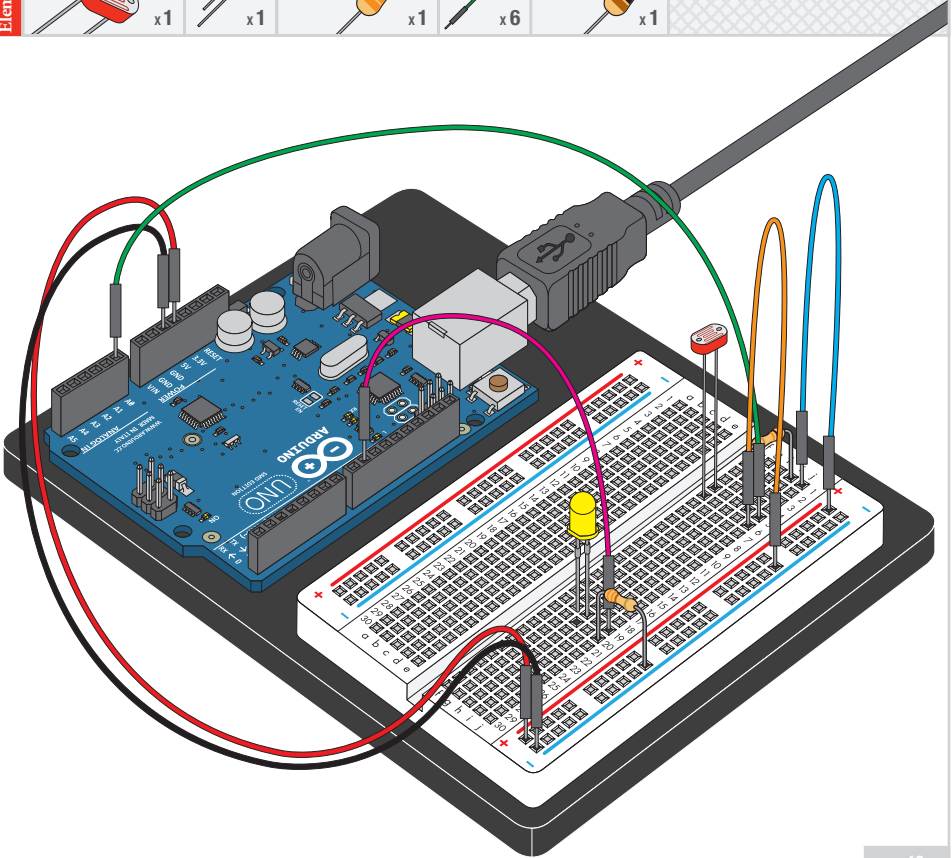


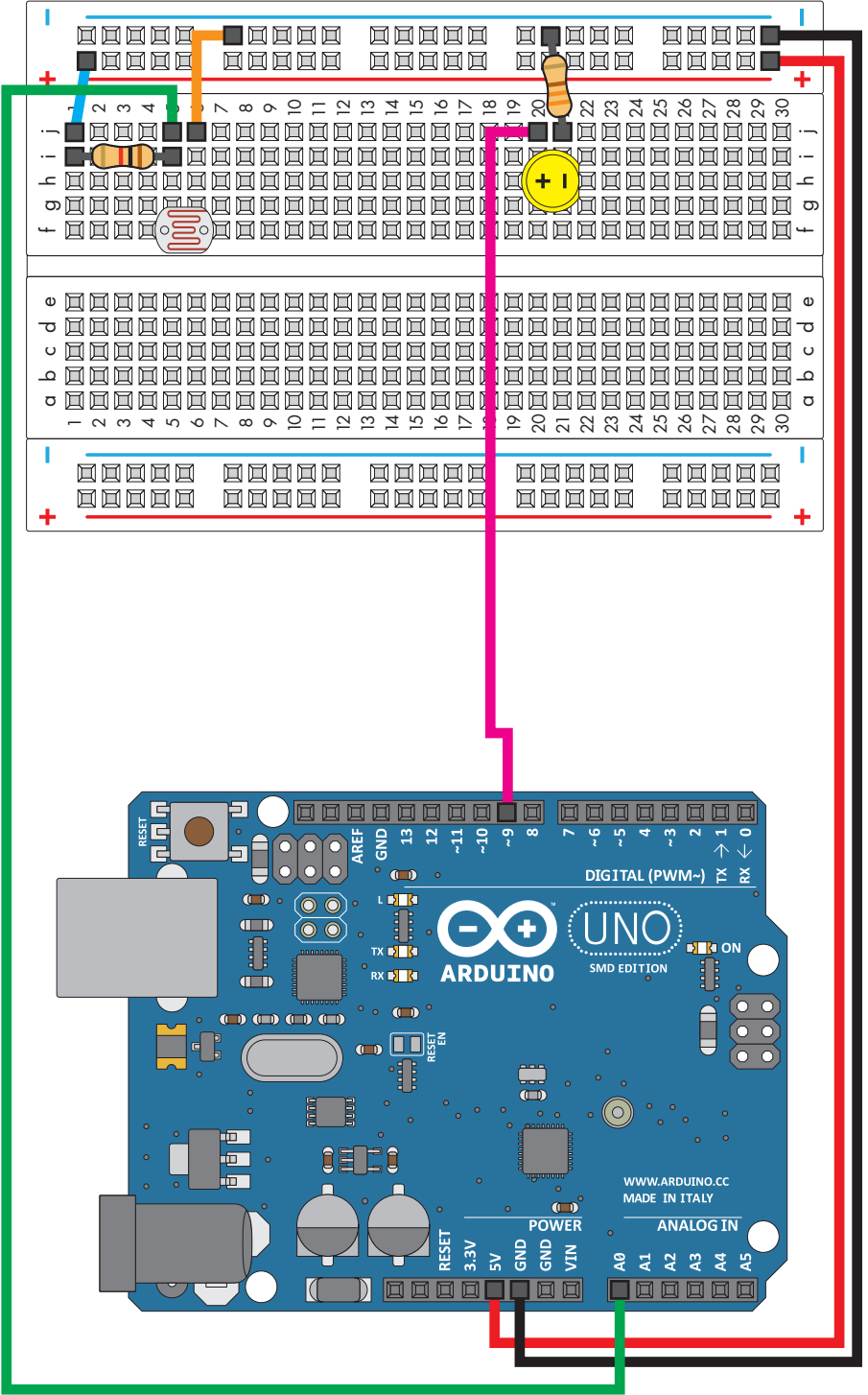
Foto Direnç

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:



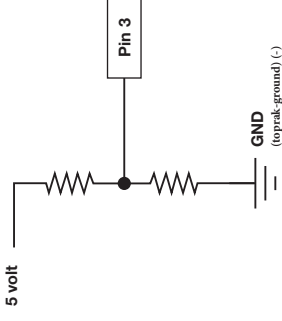
Devre 6: Foto Direç



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 Arduinomuza ile birlikte "voltaj bölücü" olarak kullandığımız 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
Atama Kablosu			i1 +
Atama Kablosu			A0 i5
Atama Kablosu			i6 -
Atama Kablosu			Pin 9 i20
Atama Kablosu			5V +
Atama Kablosu			GND -



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

Kod Notları:



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

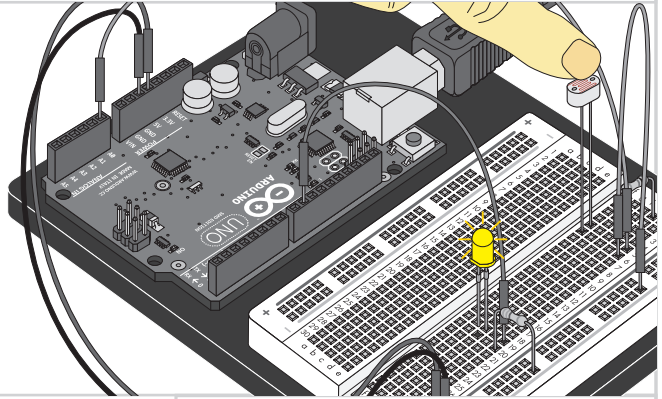
⇒ analogRead() kullanarak okuduğumuz bir analog sinyal, 0-1023 arasında bir değer olacaktır. Fakat analogWrite() kullanarak bir PWM çalıştırmak istediğimizde, 0-255 arasında bir değer isteyecektir. Bu durumda map() fonksiyonunu kullanarak geniş alanları daha dar alanlar olacak şekilde sıkıştırabiliriz.

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

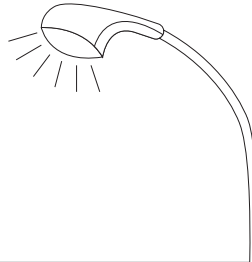
Fotodirenç üzerinde bağlantılar standart değildir, bu yüzden bağlantıları karıştırmış olmanız mümkün. Doğru yerleştirildiğinden emin olmak için iki kez kontrol edin.

Hala Çalışmıyor

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

Gerçek Hayatta Uygulamaları:

Sokak lambaları geceleri aydınlatmak için bu tür sensörler kullanır.

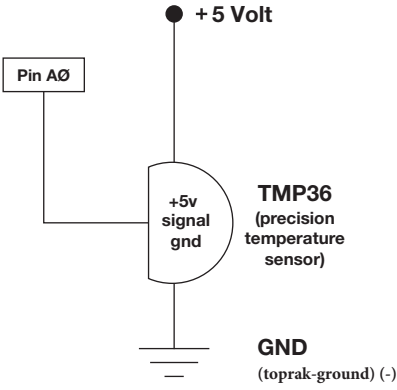


Sıcaklık Sensörü

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.

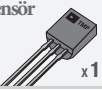


Transistör ve Sıcaklık Sensörü birbirleri çok benzeyen devre elemanlarıdır. Karıştırılmamaya dikkat edilmelidir.



Elemanlar:

Sıc. Sensör

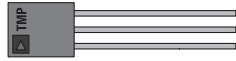


x1

Kablo



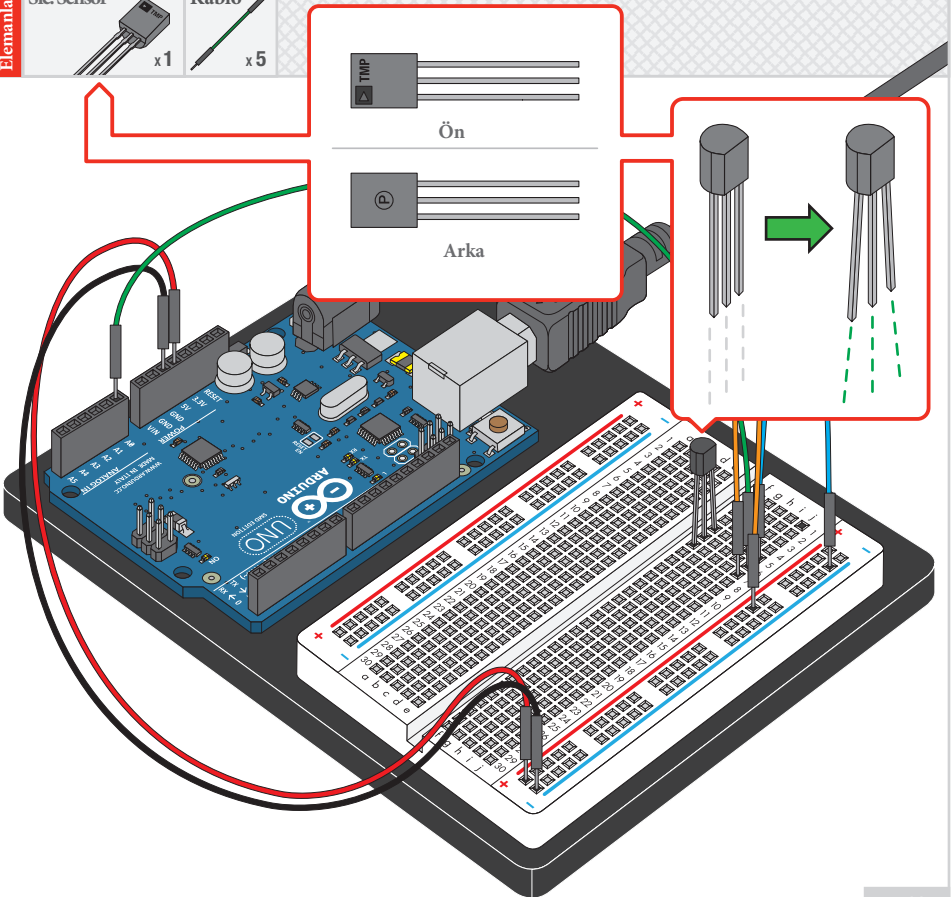
x5



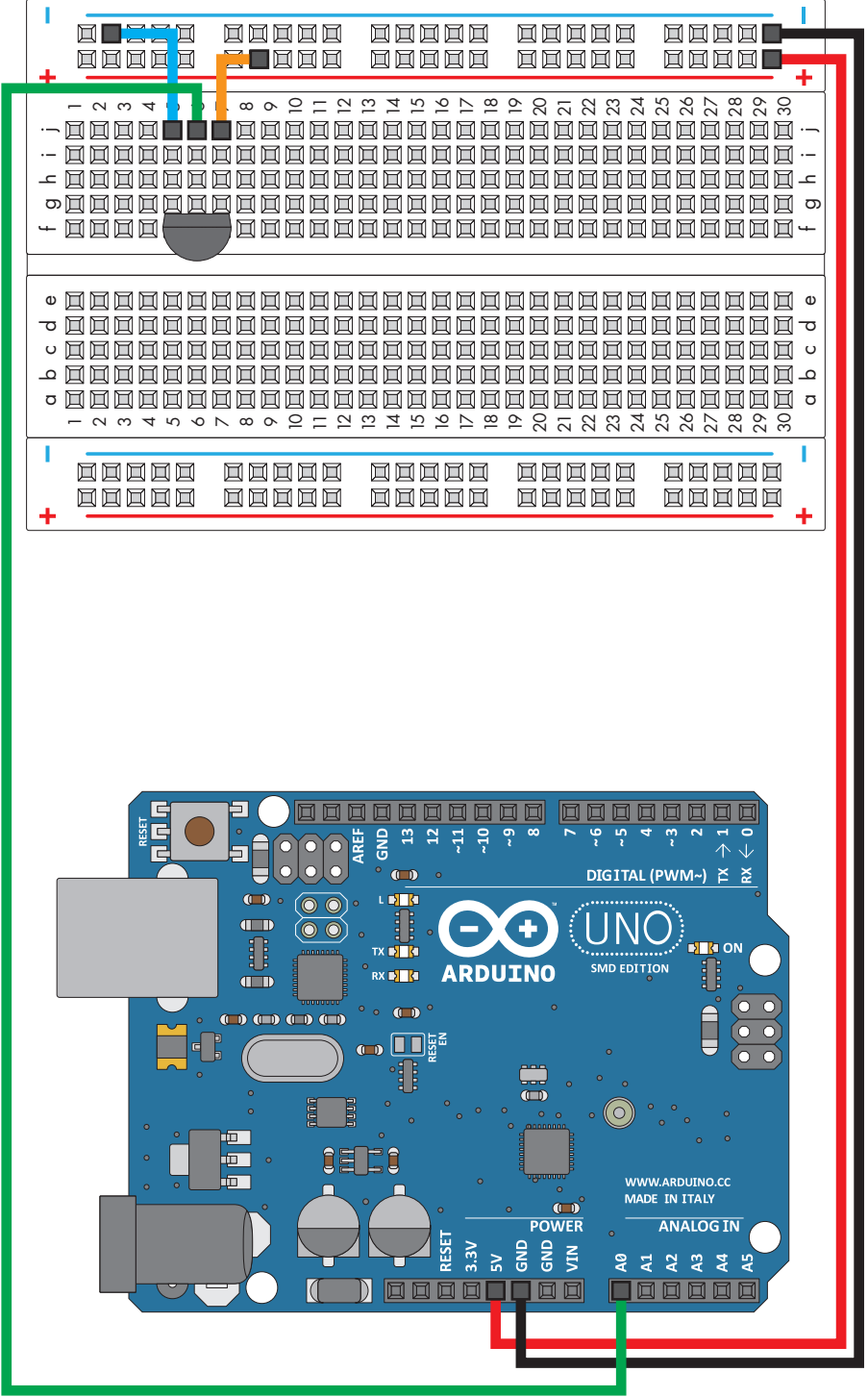
Ön



Arka

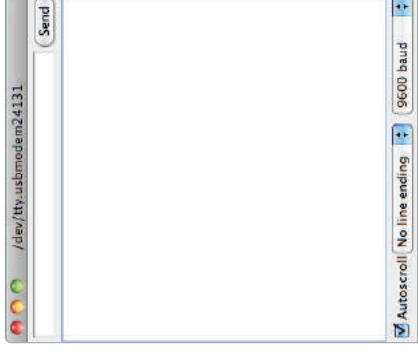














Devre 7: Sıcaklık Sensörü



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.



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>

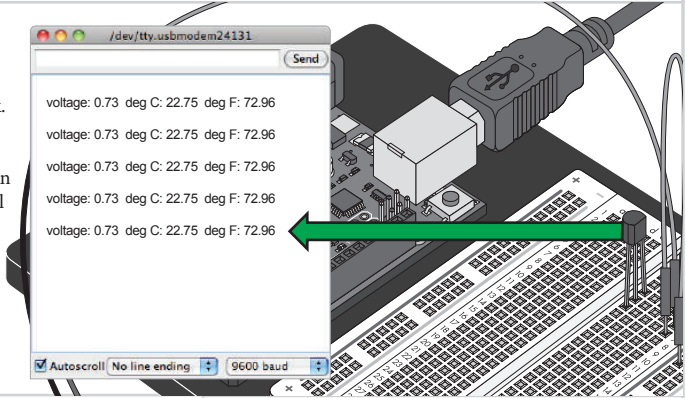
Serial.println(degreesF);



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

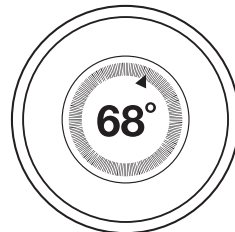
Bu tür şeylerin meydana gelmesinin sebebi seri monitörün beklenenden farklı hızda data almasından kaynaklanıyor. Düzeltmek için ise pull-down kutucuğuna tıklayıp "baud" yazan yeri "9600 baud" olarak düzeltmeniz gereklidir.

Sıcaklık Değeri Değişmiyor

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

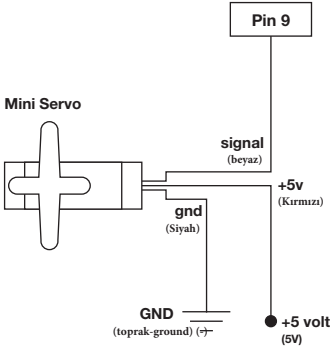
Gerçek Hayatta Uygulamaları:

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



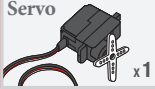
Servo Motor

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.

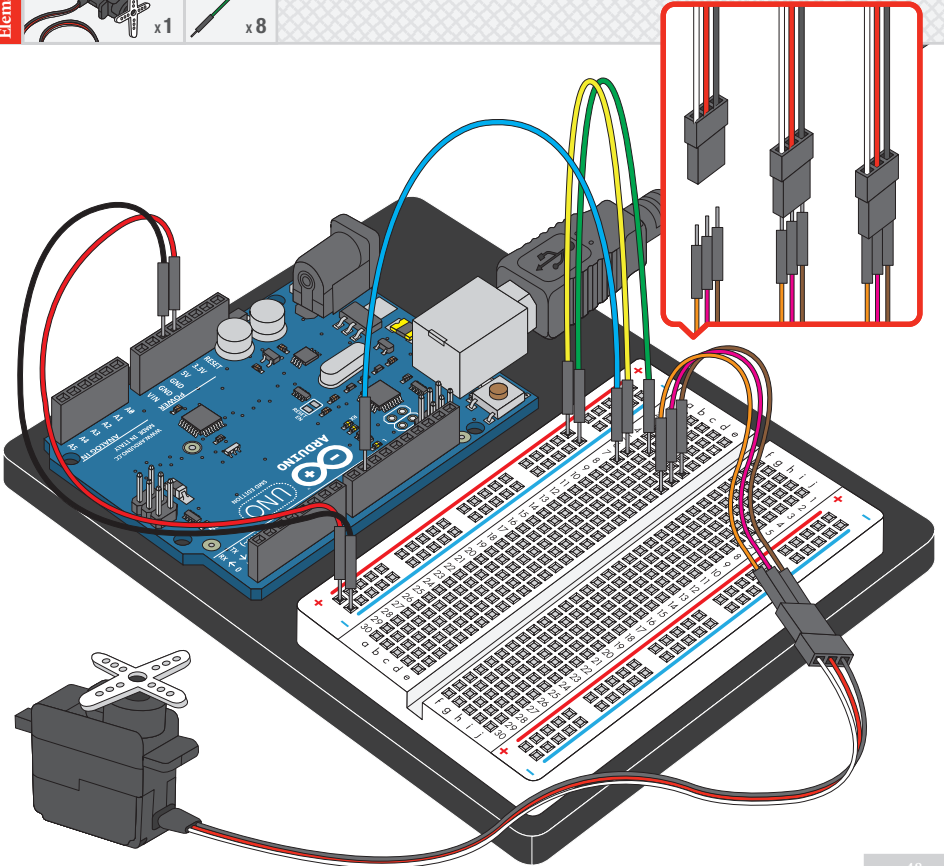


Elemenlar:

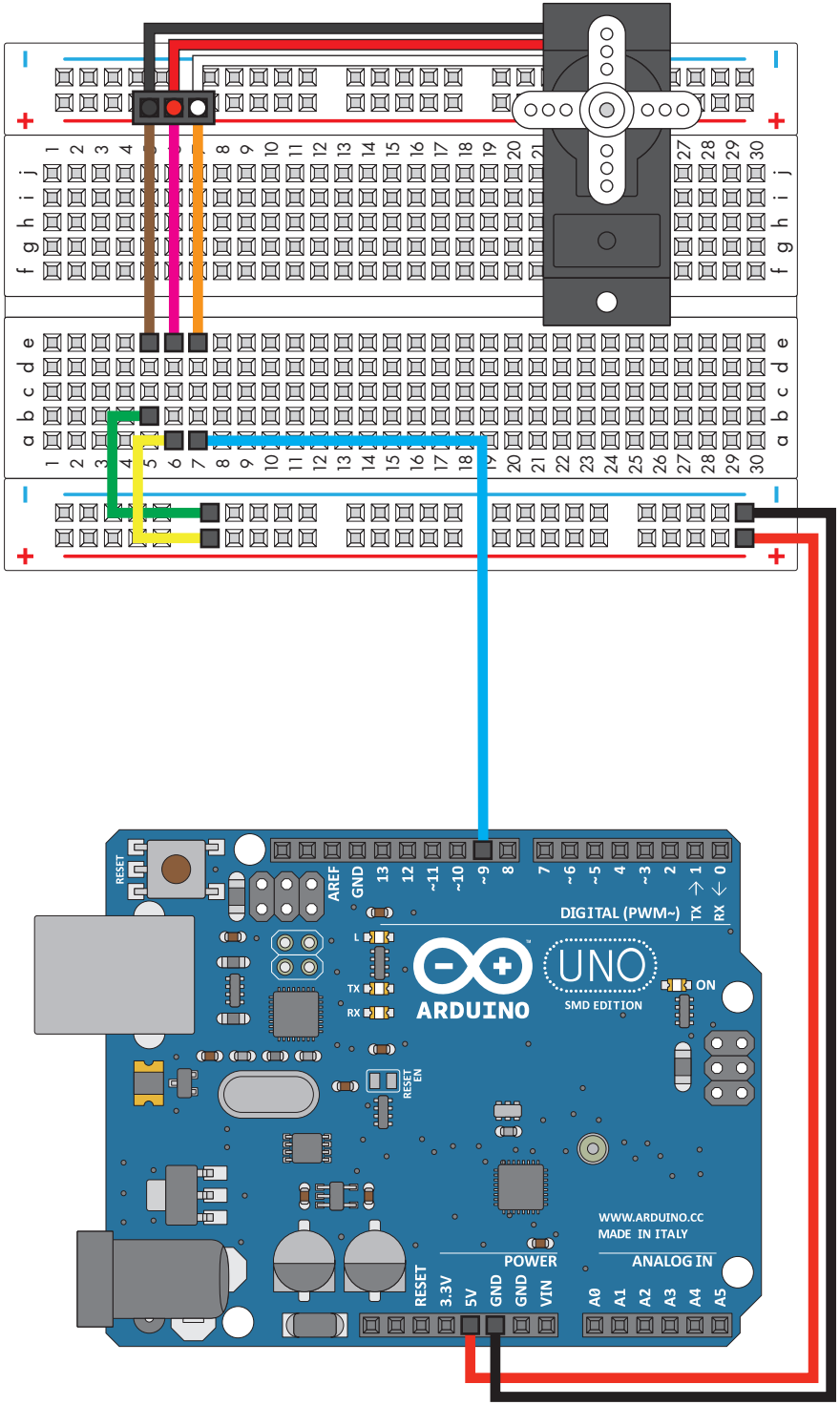
Servo



Kablo



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>

Herkese 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 kullanacaksanız, 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:	Simge	İkone
Servo Motor			
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 # 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);

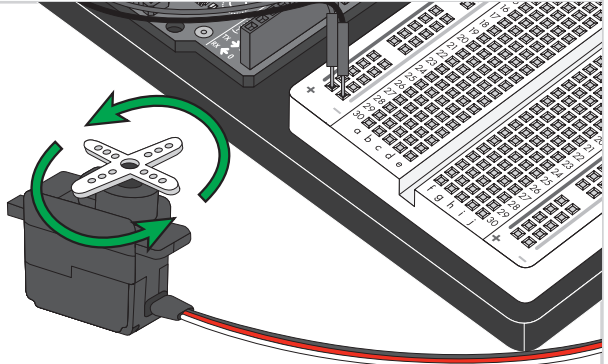
servo1.write(180);



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 sorununuz bundan kaynaklıdır.

Çalışmamakta Israr Ediyor:

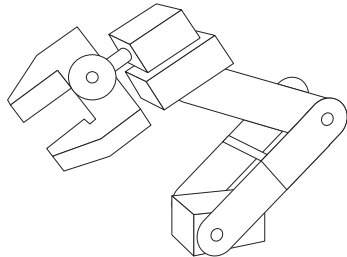
Muhtemelen (kırmızı ve kahverengi kablolar) 5 Volt ve toprak(gnd) bağlantısı yanlış yapılmıştır.

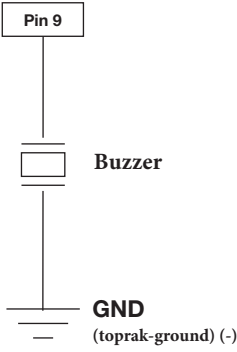
Oldu Bitti :

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

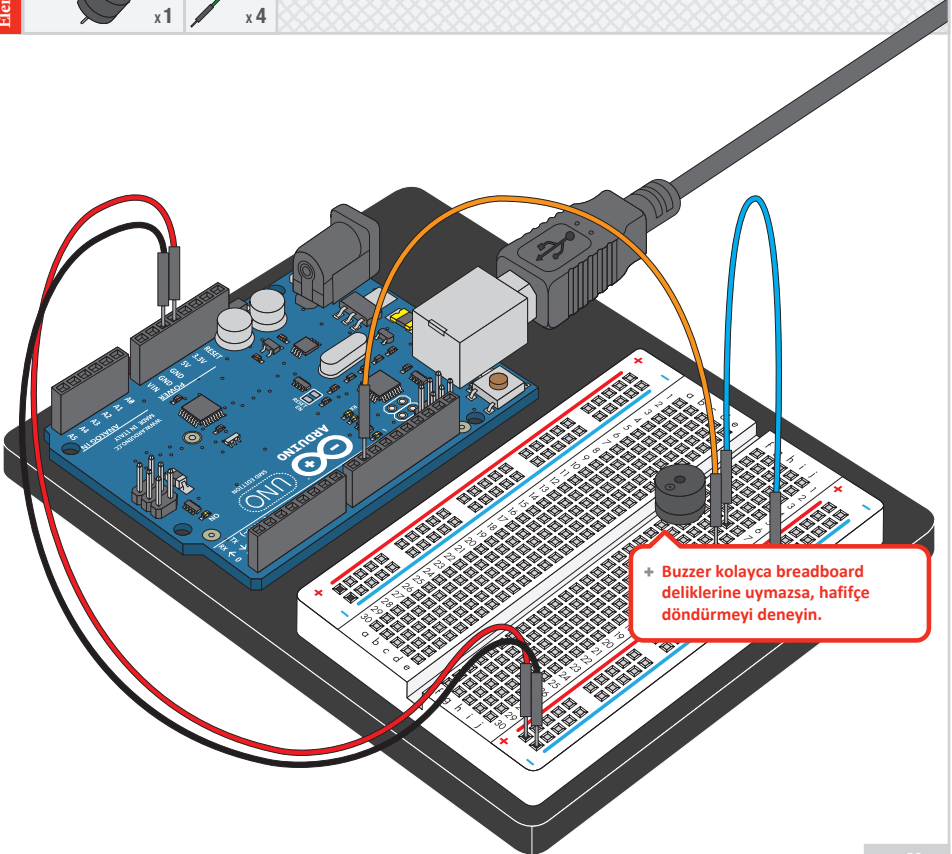


x1

Kablo

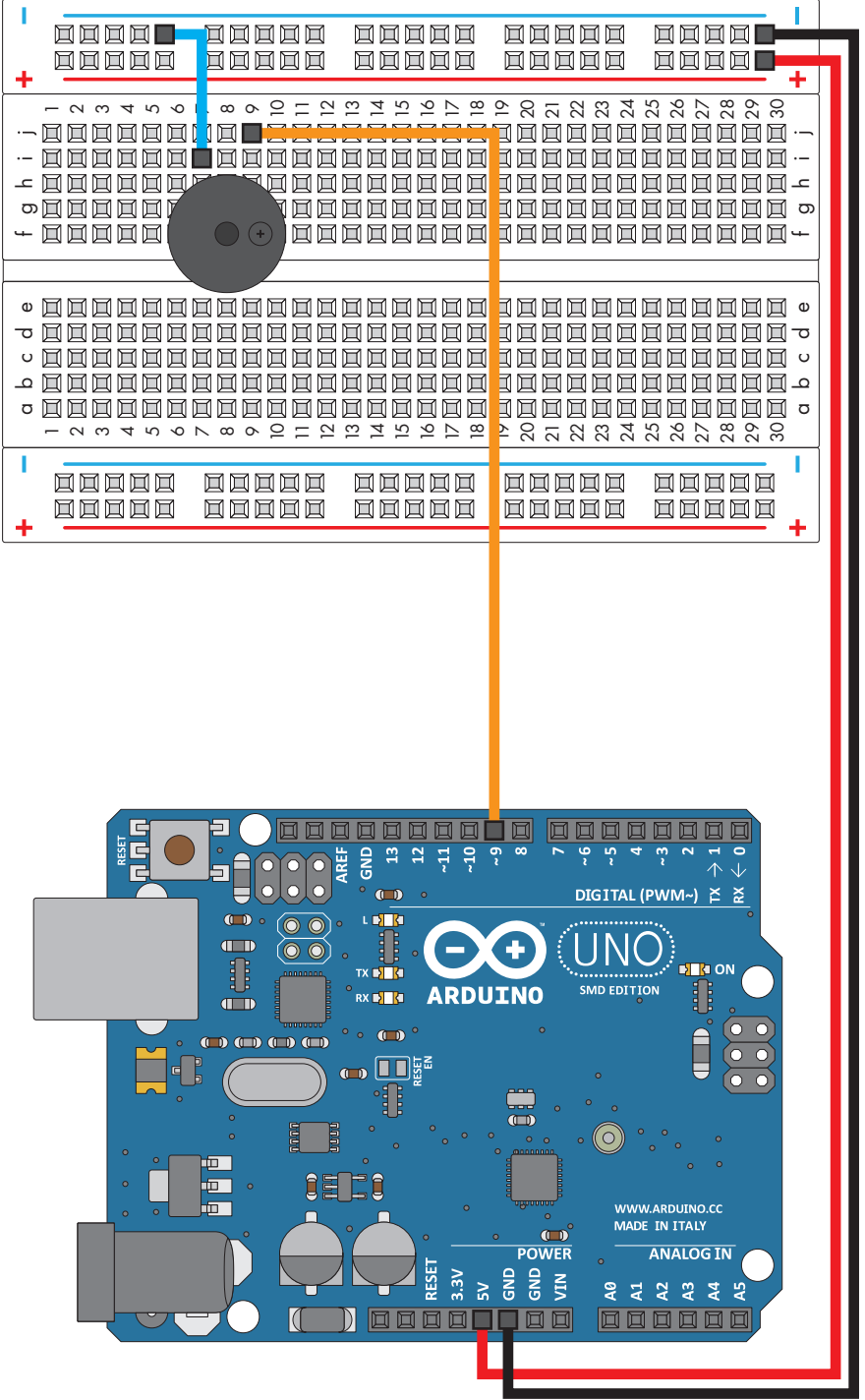


x4



✦ 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		
Atama Kablosu		
Atama Kablosu		
Atama Kablosu		
Atama 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.

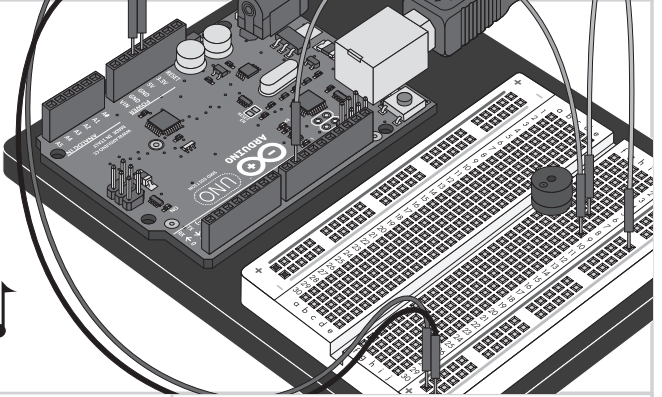
```
tone(pin, frequency, duration);
```



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

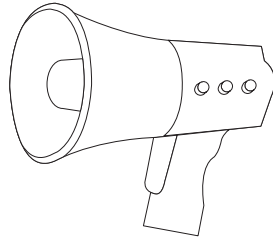
O halde düşünürken cihazınızı sökün, kullanacağınız zaman programı upload edip tekrar monte edin.

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.

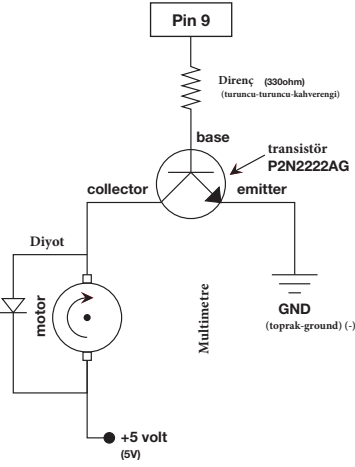


Motor Döndürmek

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 moturumuz için yeter de artar!



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



Elemanlar:

Transistör
P2N2222AGDiodyot
1N4148

DC Motor

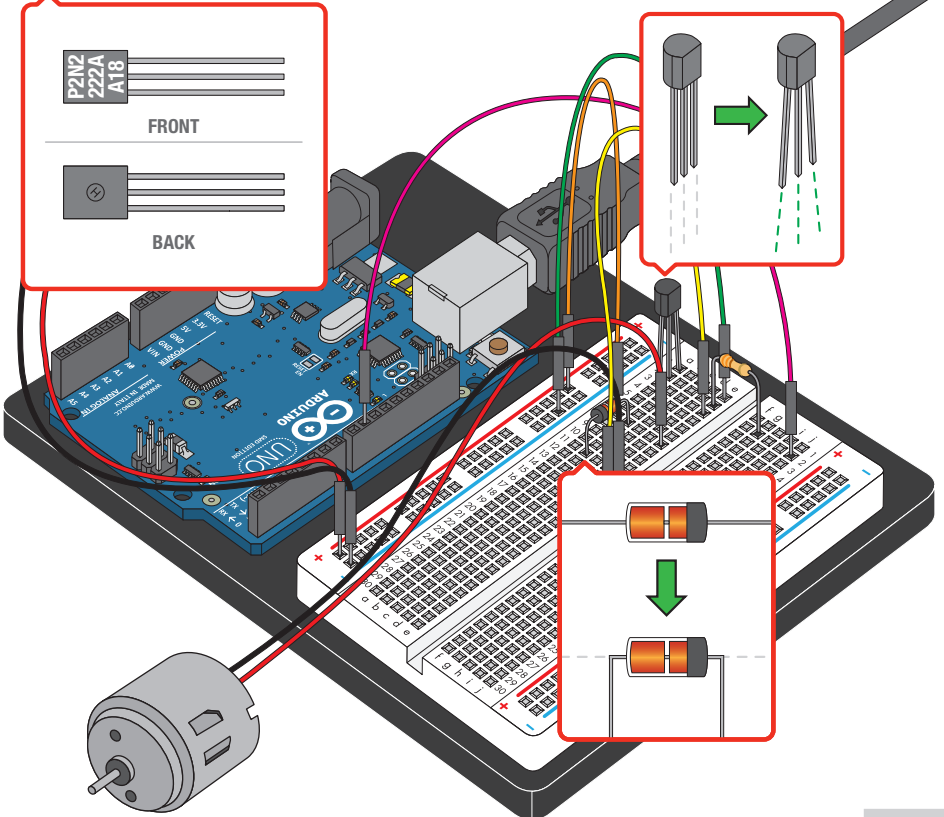
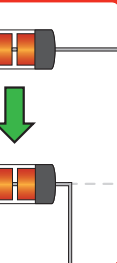
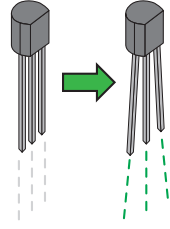


Kablo

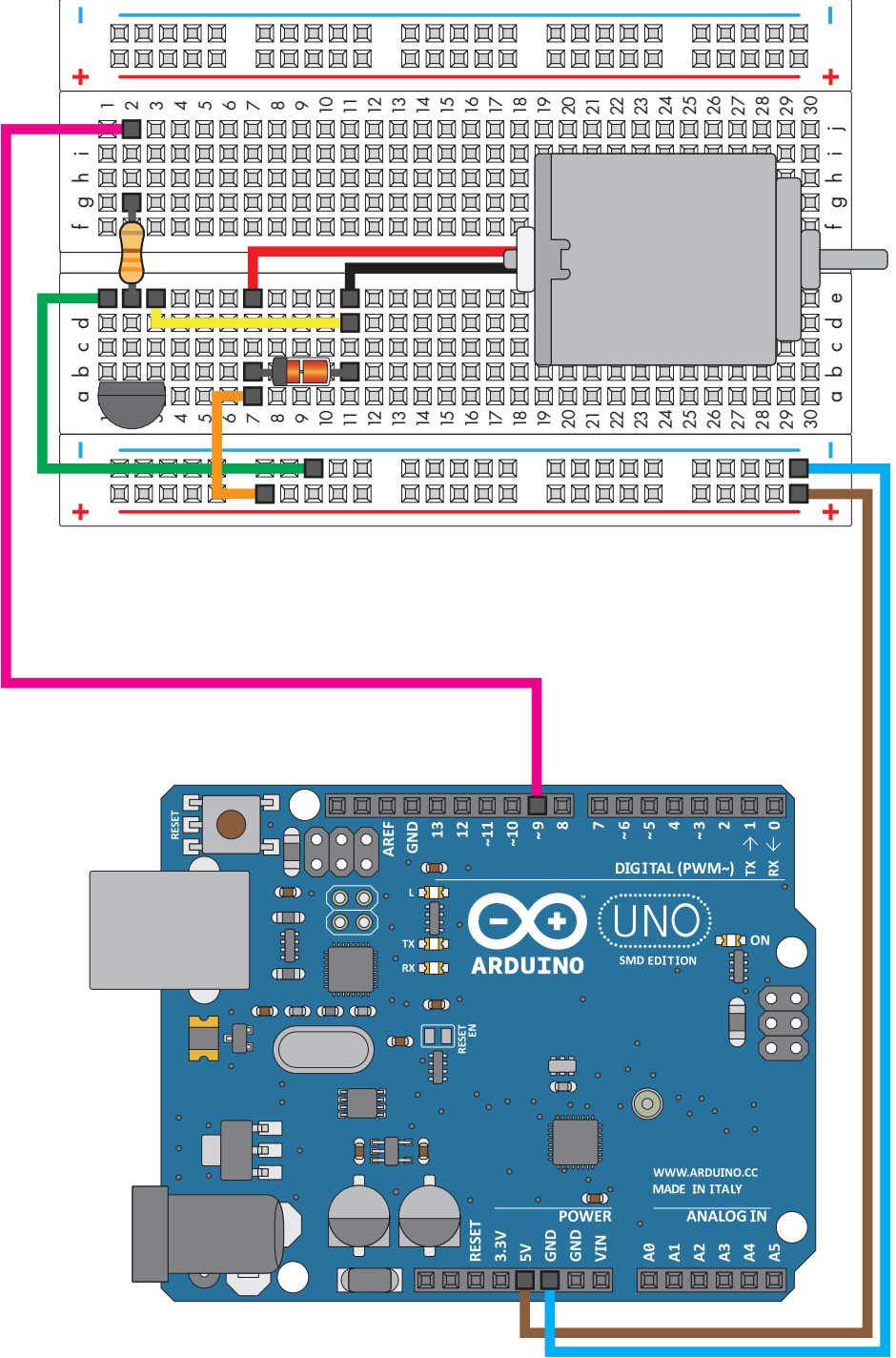
330Ω
DirençP2N2
222A
A18









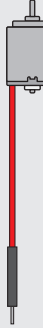























FRONT

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 öğrendiklerimizi kendi taslağımızda özgürce kullanabiliriz. İşte bu "Açık Kaynak" akımının ardında yatan fikrin ta kendisidir. Farklı tasarlardan(sketch) parçaları toplamak ve bunları birleştirip yeni şeyler üretmek fazlasıyla kolay, tek yapmamız gereken iki adet pencere açmak ve birinden diğerine kopyalayıp yapıştırmak. Bu yüzden "iyi programlama alışkanlığı" fikrimizi aşıyoruz. Aynı pin numaralarını kullanmak ve taslağımızı 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 uğrayabiliriz. Sonunda, havalı bir şeyler ürettiğiniz vakit, bunu bütün dünyayla paylaşım ki bütün dünya dahilliğimizden faydalanabilsin. (Ve bizim bundan haberdar olmamızı sağlayın ki bunu anasayfamızda paylaşabilelim!)



Kod Notları:



while (Serial.available() > 0)



Arduino seri portları data gönderimi için olduğu gibi data almı 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.

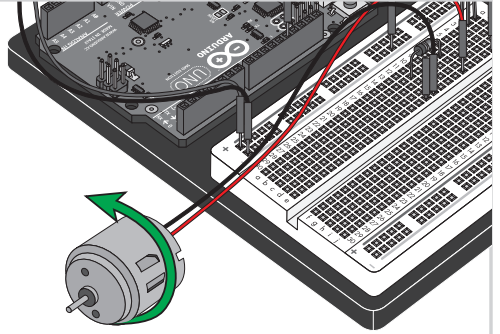
speed = Serial.parseInt();



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

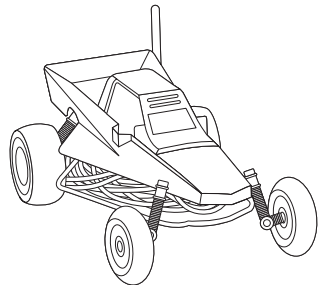
Kendi motorunuzu kullanıyorsanız motorunuzu 5V ile çalıştırmayı deneyin ve daha fazla güç çekip çekmediğini kontrol edin.

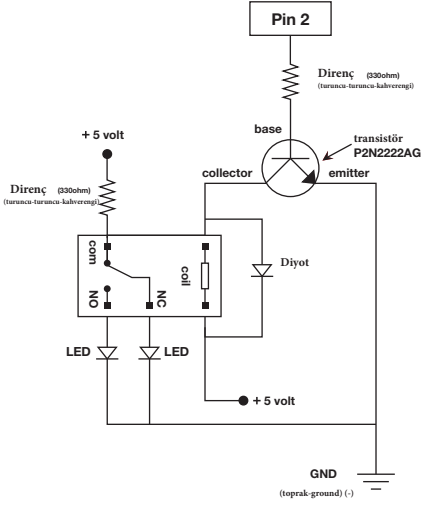
Çalışmamakta Israrlı

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.



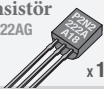
Röle kapalıyken, COM(common) pini NC (Normally Closed) pinine bağlanır.
Röle açıkken, COM(common) pini NO (Normally Open) pinine bağlanır.

Elemanlar:

Röle



x1

Transistör
P2N2222AG

x1

Diodyot
1N4148

x1

330Ω
Direnç

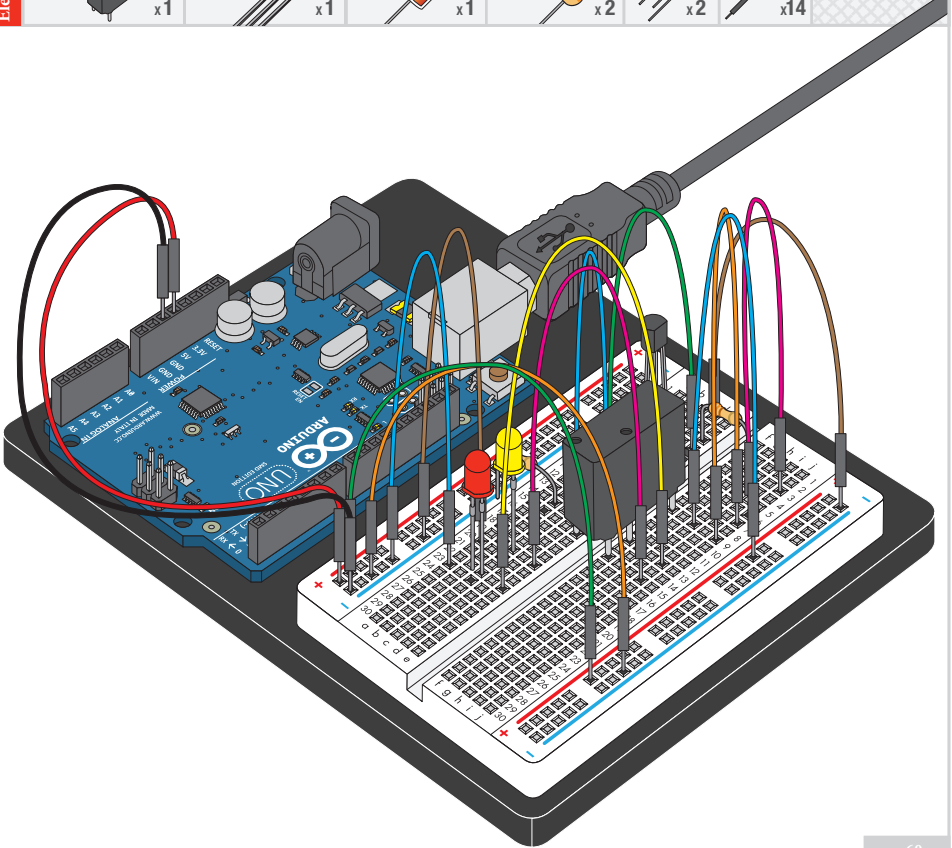
x2



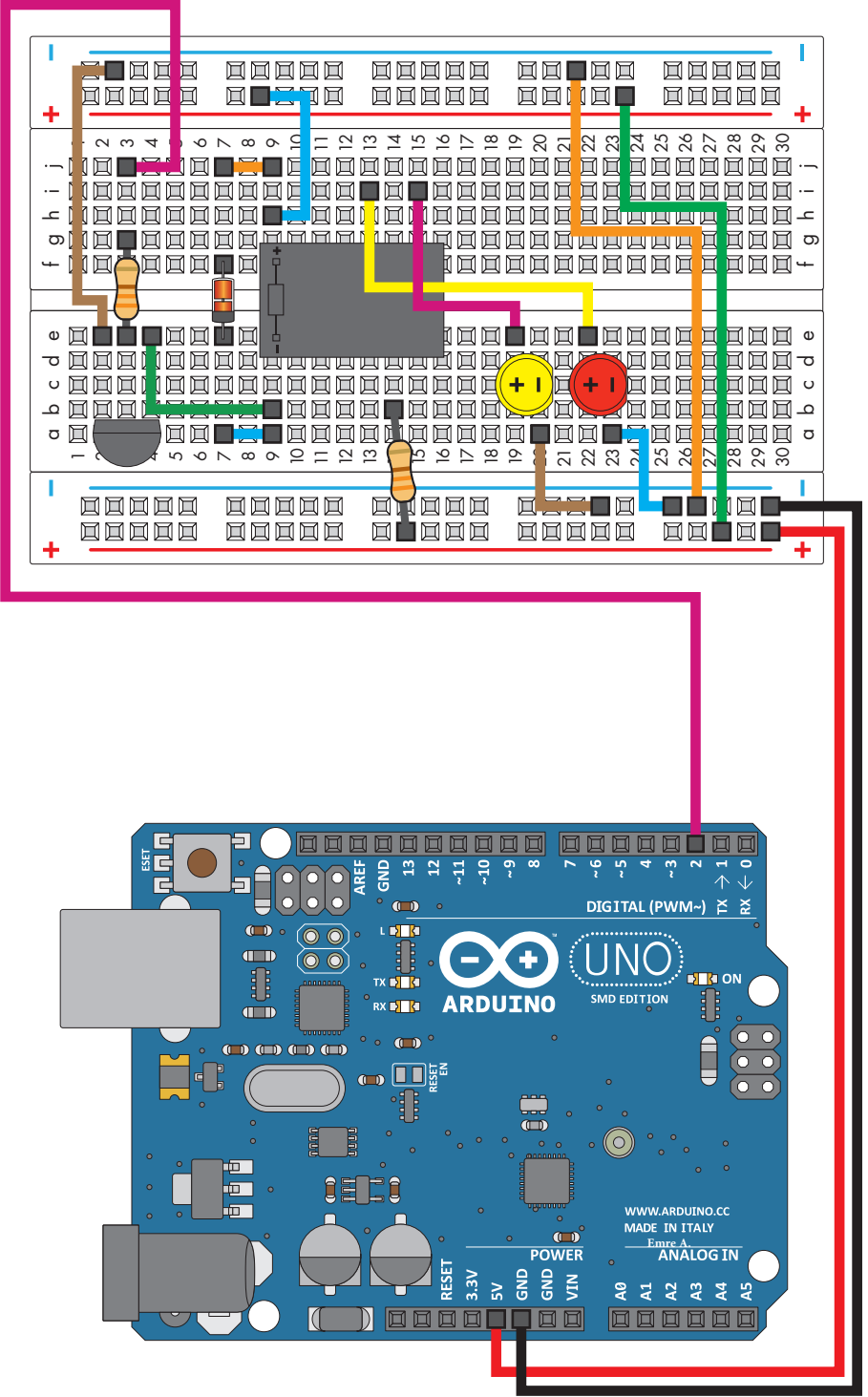
x2














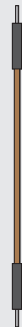














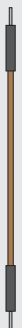






















x14



Devre 11: Rôle



Elemanlar:	Gerçek Görünüm:	Elemanlar:	Gerçek Görünüm:	Elemanlar:	Gerçek Görünüm:
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					
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 tabi ki.)

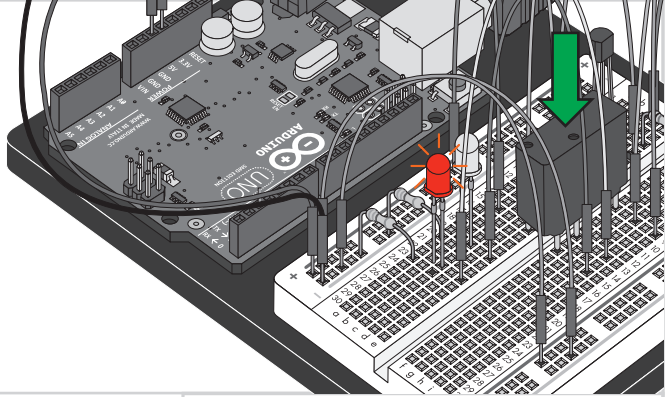
digitalWrite(relayPin, LOW);



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

LEDi 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.

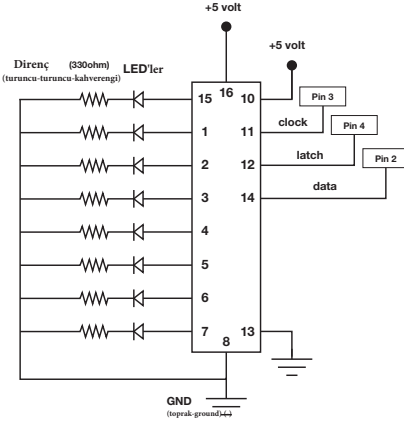
Gerçek Hayatta Uygulamaları:

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.

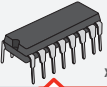


Q_b	1	16	V_{cc}
Q_c	2	15	Q_a
Q_d	3	14	SER
Q_e	4	13	OE
Q_f	5	12	RCLK
Q_g	6	11	SRCLK
Q_h	7	10	SRCLR
GND	8	9	Q_i

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

Elementler:

IC



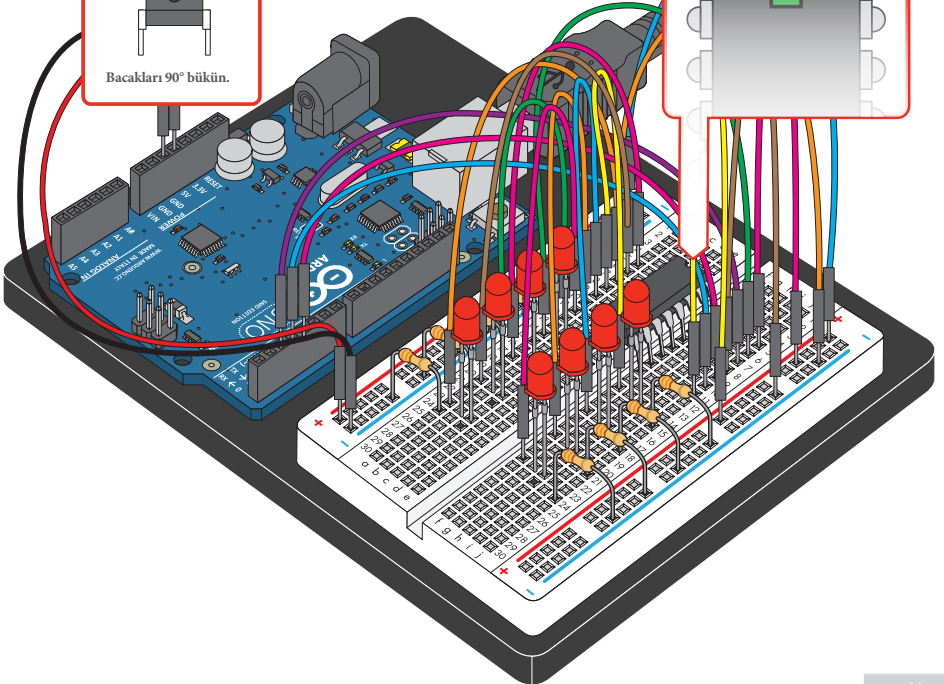
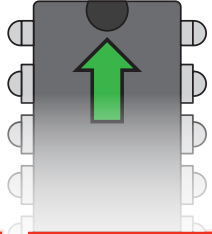
LED

330Ω
Direnç

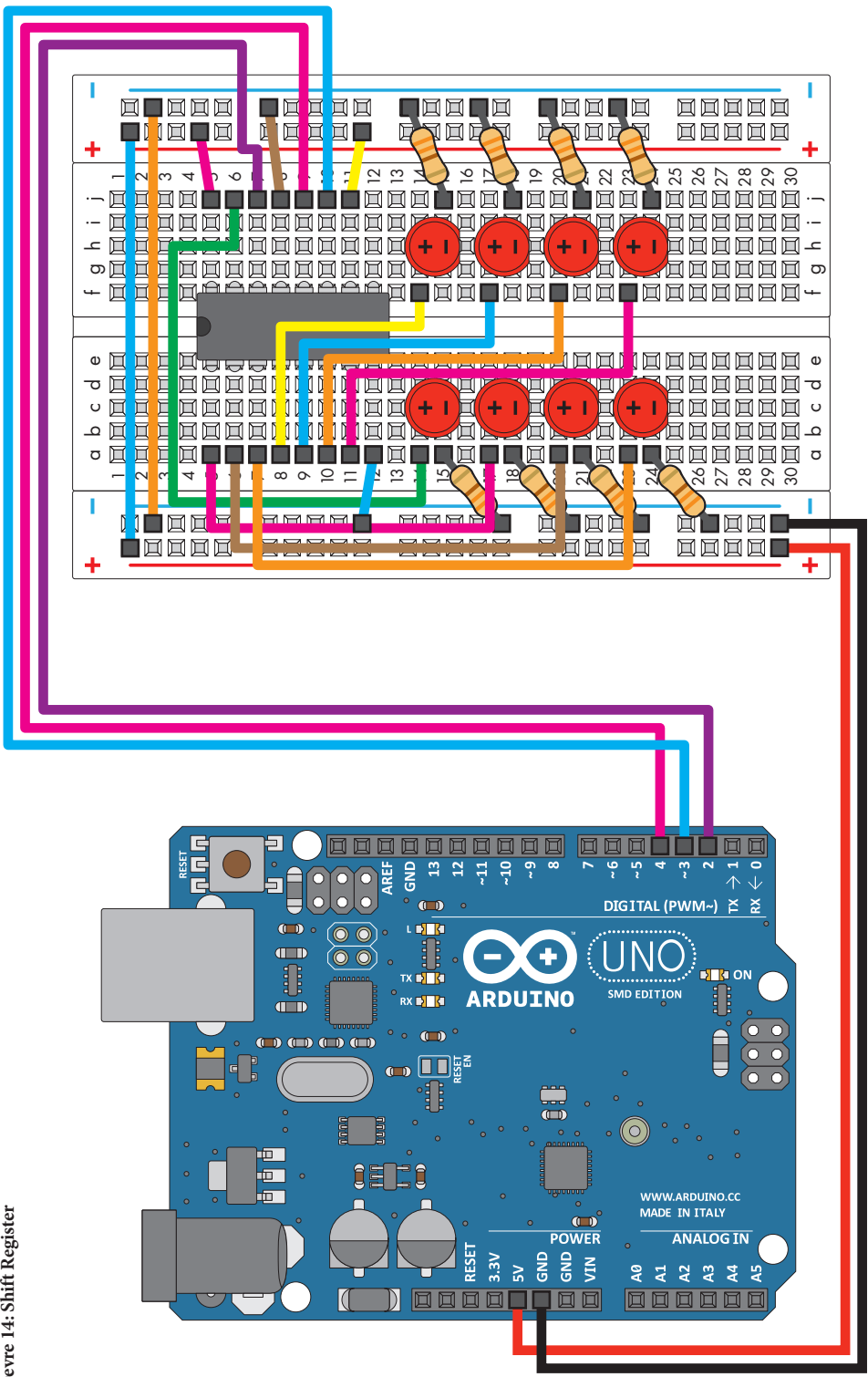
Kablo



Bacakları 90° bükün.



Devre 14: Shift Register



Elemanlar:	Gerçek Görünüm:	Elemanlar:	Gerçek Görünüm:	Elemanlar:	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			
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);
```

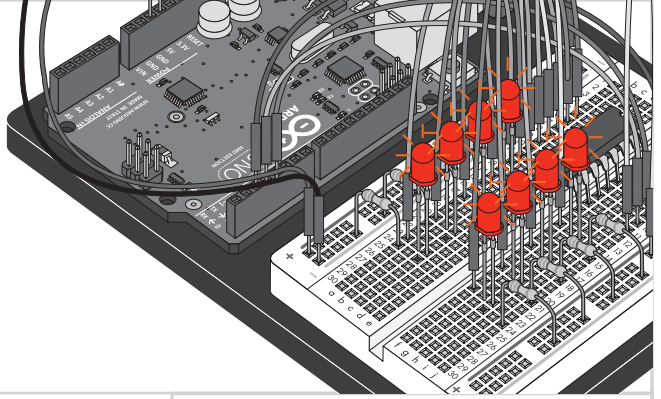


Bitler bilgisayar hafızasının en küçük yapı taşlarıdır, her bit 1 veya 0 depolayabilir. Daha büyük sayılar bitlerden oluşan dizilerle depolanır. Bazen bu bitler üzerinde oynama yapmak isteriz. Örneğin 8 bitli shift register'a gönderiyoruz ve LED'i açıp kapatması için 1 veya 0 yapmasını istiyoruz. Arduino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

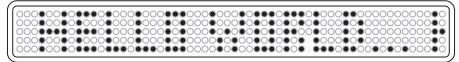
Bu bir çok kez başımıza geldi. Sebebi entegrenin ters takılmasından kaynaklanıyor. Eğer hemen düzeltirseniz hiçbir şeyi bozmadan durumu kurtarabilirsiniz.

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)