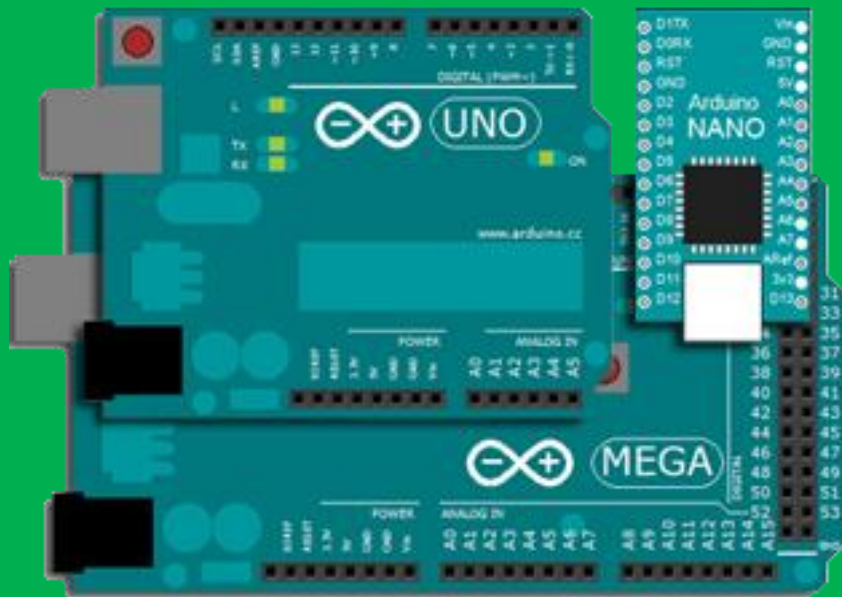




SCRATCH İLE ARDUİNO PROGRAMLAMA

Mahmut BORA KARAKUŞ
borakarakus@yandex.com



Katkılarından dolayı robotistan.com'a teşekkür ederim.

robotistan.com



"Türkiye'nin En Büyük Robotik Marketi"

Robotistan Satış Ofisi:

Perpa Ticaret Merkezi B Blok Kat:8 No:1101
(Halk Bankası Yanı) Şişli / İstanbul

Tel: 0850 766 0 425

Niçin Programlama Öğrenmeliyiz?

Programlama: Bilgisayara veya elektronik cihazlara ne yapması gerektiğinin adım adım söylenmesidir. Kullandığımız bütün elektronik cihazları (Bilgisayar,Çamaşır makinesi, Televizyon) kontrol eden programlar vardır. Bu programlar cihazların ne yapması gerektiğini anlatan kılavuzlardır.

Programlar, programlama dilleri kullanılarak yazılır. İnsanların günlük hayatta iletişim kurmak için nasıl bir dil kullanmaları gerekiyorsa bizim de bilgisayarlar ile haberleşebilmemiz için bir dil kullanmamız gerekir. Bu diller yazılı dillerdir . Üst seviye(kolay) ve alt seviye(zor) diller vardır. Mesela bugün okullarımızda okutulan bilişim teknolojileri ve yazılım derslerinde öğrendiğimiz Scratch programlama dili üst seviye(Kolay) bir programlama dilidir.

Programlama ilk başta çok karmaşık ve zor gözükebilir, bu sizi korkutmasın! İlk araba sürmeyi öğrendiğim günü hatırlıyorum, çok korkuyordum, öyle ki bu araba sokağın köşesinden nasıl dönecek diyordum ama zamanla anladım ki çok kolaymış. Programlama zor diyenlere inanmayın, ilerledikçe siz de çok kolay olduğunu göreceksiniz.

“Bu ülkedeki herkes programlamayı öğrenmeli, çünkü insana nasıl düşünmesi gerektiğini öğretiyor.”

Steve Jobs

Bilgisayara bir işi yaptırırken yani programlarken düşünmeyi öğreniriz. Analiz yapabilme ve problem çözebilme yeteneğimiz artar. Bir sorunla karşılaştığımızda daha iyi düşünebiliriz ve o sorun için bulduğumuz çözümlerin sayısı artar. Olaylara daha ayrıntılı bakabiliriz. Bilgisayarda program yazarken algoritmalar geliştiririz. Algoritma, problemin adım adım çözülme sürecidir. Bu da bize olayları detaylı düşünebilme ve çözebilme yeteneği kazandırır.

Ek olarak programlama çok eğlencelidir.



Yazar Hakkında



SCRATCH



1985 Mersin doğumluyum.
İlköğretim ve lise hayatım Mersin'de geçti.

Çanakkale 18 Mart Üniversitesi
Bilgisayar ve Öğretim Teknolojileri
Öğretmenliği mezunuyum.

Ahmet Yesevi Üniversitesi Bilgisayar
Mühendisliği 2. Sınıf öğrencisiyim.
Robotik ve Arduino özel ilgi alanımdır.
Ayrıca spor yapmayı, müzik dinlemeyi ve
paten kaymayı çok severim. Bir devlet
okulunda Bilişim Teknolojileri ve
Yazılım Dersi öğretmenliği
yapmaktayım.

Kitap Hakkında

Her geçen gün teknoloji hızla ilerlemekte, bu teknoloji çağında doğmuş ve büyüyor olan sizlerin hiç şüphesiz bu teknolojiye ayak uydurması gerekmektedir. Siz öğrenim hayatınızı bitirip iş hayatına atılıncaya kadar geçecek olan sürede, dünya çok değişmiş olacak ve bu değişime ayak uydurabilen , teknolojiyi etkin kullanan, problem çözme yeteneği üst seviyede olan, eleştirel düşünebilen girişimci bireyler her zaman bir adım önde olacaklar.

Bu kitap sizin teknolojiyi etkin kullanırken, programlama ile problem çözebilme yeteneğinizi eğlenceli ve zevkli bir yoldan arttırmayı hedeflemektedir. Bu amaç için dünyada çokça kullanılan programlanabilen Arduino kartlar kullanılmıştır. Arduino kartı programlamak için ilk aşamada Bilişim Teknolojileri ve Yazılım Derslerinde öğrendiğiniz Scratch programlama dili kullanılmıştır. İkinci aşamada ise Arduino'nun kendi yazılım geliştirme aracı kullanılmıştır.

Hepinize faydalı olması ümidiyle, kolay gelsin.

İÇİNDEKİLER

Scratch ile Arduino programlama

Arduino Nedir?	Sayfa 6
Arduino Rehberi	Sayfa 7
Klon Arduino'nun Driverını Yükleme	Sayfa 9
Arduino'muzu Tanıyalım	Sayfa 10
Scratch Nedir?	Sayfa 11
S4A Program Dosyalarının Yüklmesi	Sayfa 12
Arduino Üzerindeki Led'in Kontrolü	Sayfa 14
Breadboard Üzerinde Çalışmak	Sayfa 15

Elektronik Hakkında Temel Bilgiler

Elektrik Akımı	Sayfa 17
Elektronik Devre Elemanları	Sayfa 18
Dirençler	Sayfa 19

Uygulamalar

Breadboard Üzerindeki Led'i Yakıp Söndürme	Sayfa 22
Yürüyen Işık	Sayfa 23
Buton ile Led Yakma	Sayfa 25
Rgb (Red Green Blue) Led	Sayfa 26
Potansiyometre	Sayfa 28

Sensörler

Hava Kararınca Yanan Led	Sayfa 30
Mesafe Sensörü	Sayfa 31
Civalı Sensörle Oyun Kolu Yapımı	Sayfa 34
Muzdan Piyano	Sayfa 35

Android Telefonla Scratch Kontrolü

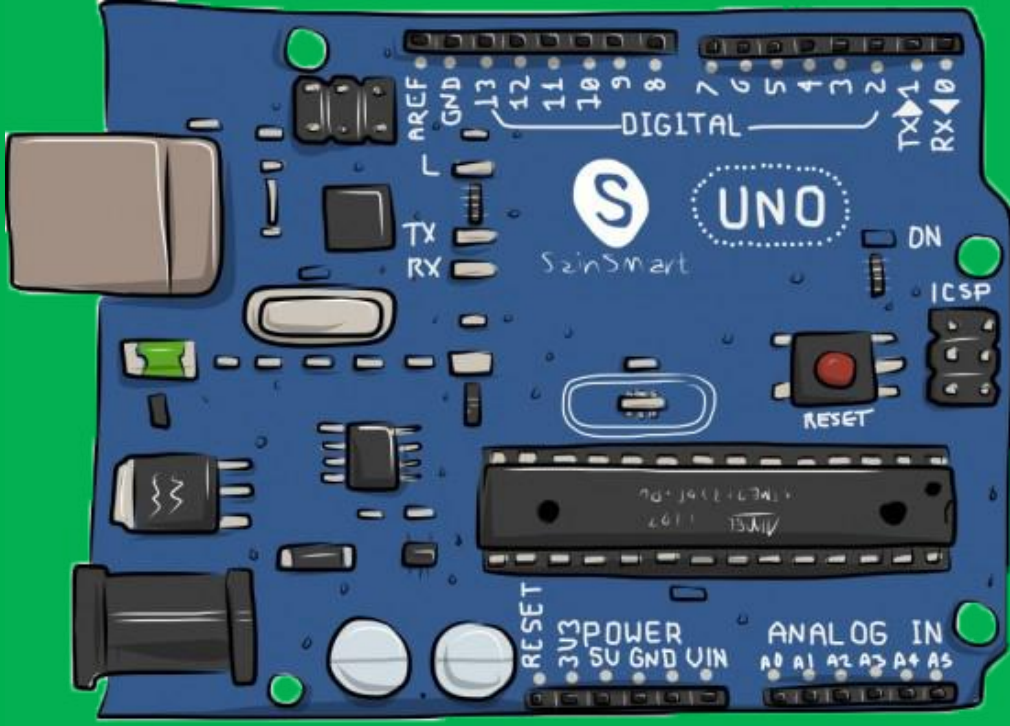
Sayfa 37

Arduino Yazılım Geliştirme Aracı

Sayfa 39

Led Yakıp Söndürme (Blink)	Sayfa 40
Değişken Kullanımı	Sayfa 42
Analog Pinden Değer Okuma	Sayfa 43
İf / Else (Eğer ise/ değilse)	Sayfa 44
Potansiyometreden Okunan Değere Göre Led'i Yakma	Sayfa 46
Döngüler	Sayfa 47

Arduino Nedir?



Arduino, bir mikro kontrol ünitesidir. Bilgisayarımızda bulunan işlemciye benzer. Üzerinde elektronik cihazlarımızı kontrol etmemize yarayan bir mikrodenetleyici vardır. Biz bu mikrodenetleyiciyi programlayarak istediğimiz işleri yapmasını sağlarız. Arduino'muz ile yapabileceğimiz hayli geniştir ve sizin hayal gücünüze kalmıştır. Örneğin; engelleri algılayıp aşan bir robottan, toprağın kurduğunu anlayıp çiçeğinizi sulayan bir sisteme kadar geniş bir yelpazede çalışmalar yapabilirsiniz. Arduino'nun bir çok modeli mevcut. Biz kitabımızda en çok kullanılan model olan uno r3 modelini kullanacağız.



[Youtube videosu için tıklayınız.](#)

Arduino'muz ile projelerimizin büyük bir bölümünde hiçbir şekilde lehimleme veya vidalama gibi karışık işler yapmayacağımız için gayet kolay bir şekilde adımları takip edebileceksiniz.



Engel Aşan
Robot

Arduino Rehberi

Arduino'nun piyasada orijinal ve klon versiyonları mevcut.

© Orijinal Arduino (Usb kablosuz)

<http://www.robotistan.com/Orijinal-Arduino-UNO-R3-Yeni-Versiyon,PR-903.html>

© Kutulu Klon(Çok benzer kopyası) Arduino

<http://www.robotistan.com/Arduino-UNO-R3-Kutulu-Klon,PR-2106.html>

Orijinal Arduino ile klon Arduino arasında pek bir fark yoktur. Orijinal Arduino İtalya'da üretilirken klon Arduinolar Çin'de üretilir. Orijinal Arduino daha kalitelidir.

Projelerimizi uygularken bazı devre elemanlarına ihtiyacımız olacak bu elemanları elektronikçilerden temin edebilirsiniz ya da size vereceğim, kitapta kullanmış olduğum malzemelerin olduğu alışveriş listesindekileri satın alabilirsiniz.

<http://www.robotistan.com//Breadboard,PR-694.html>

<http://www.robotistan.com/Mano-K-35-Cekmeceli-Kutu-110x170x65mm,PR-1556.html>

<http://www.robotistan.com/14W-10K-Direnc-Paketi-10-Adet-,PR-335.html>

<http://www.robotistan.com/14W-22K-Direnc-Paketi-10-Adet-,PR-338.html>

<http://www.robotistan.com/14W-220R-Direnc-Paketi-10-Adet-,PR-319.html>

<http://www.robotistan.com/10K-Potansiyometre-Ayarli-Direnc,PR-281.html>

<http://www.robotistan.com/5mm-Seffaf-Kirmizi-LED-Paketi-10-Adet-,PR-136.html>

<http://www.robotistan.com/Jumper-Kablo-E-E-M-M-65li-Karisik-Paket,PR-847.html>

<http://www.robotistan.com/2-Pinli-Uzun-Push-Buton,PR-819.html> (2 adet)

<http://www.robotistan.com/5mm-Seffaf-RGB-LED,PR-160.html>

<http://www.robotistan.com//Buzzer,PR-585.html>

<http://www.robotistan.com/5mm-LDR,PR-460.html>

<http://www.robotistan.com/Civali-Sensor,PR-1387.html>

<http://www.robotistan.com/HC-SR04-Ultrasonik-Mesafe-Sensoru,PR-1473.html>

Önemli hatırlatma:

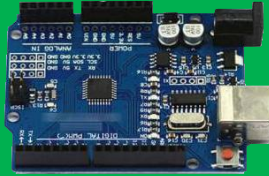
Burada paylaşılan linkler sadece tavsiye niteliğindedir. Yaşanacak problemlerden yazar sorumlu tutulamaz.

Ben, orijinal Arduino kullanmanızı tavsiye ediyorum.

© Orijinal Arduino:

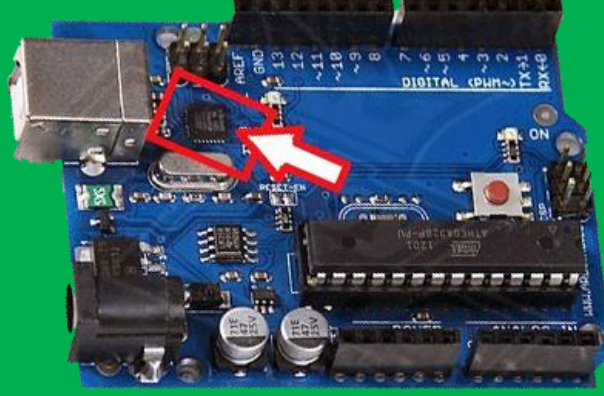


© Klon Arduino:

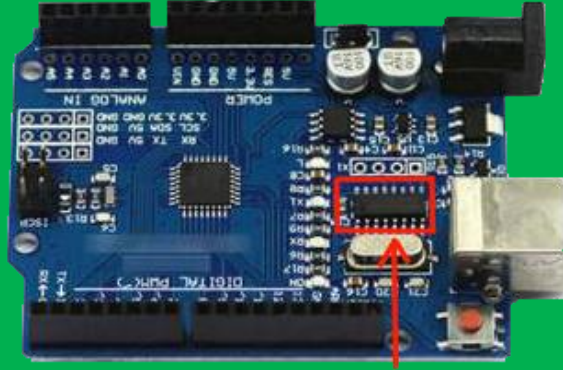


Eğer elinizdeki Arduino bir klon Arduino ise bilgisayarınıza usb ile bağlarken problem yaşayabilirsiniz. Bilgisayara bağladığınız her donanımın nasıl çalışacağını tanıtan bazı dosyalar vardır. Biz bunlara driver(sürücü) dosyaları deriz. Orijinal Arduino için Arduino programını kurduğunuzda tüm sürücüler yüklenir ancak klon Arduino farklı bir usb haberleşme çipi kullanıyorsa sorun yaşarsınız. Usb haberleşme çipi yazdığımız programları mikrodenetleyiciye yüklemek için kullanılır. Bu da Arduino'muzun kullanımını çok kolay yapar. Programımızı yazarız ve Arduino'muza yükleriz. Bu kadar basit.

Elinizdeki klon Arduino'yu bilgisayarınıza tanıtamadıysanız hemen telaşlanmayın Usb çipine bakın ve onun sürücüsünü internetten bulup indirin. Benim elimdeki klon Arduino'nun üzerinde CH340G adlı bir Usb çipi vardı muhtemelen sizdeki klon Arduinoda da aynı çip vardır.



Atmega8u2 kullanan klon Nano (Bu kurulum gerektirmez)



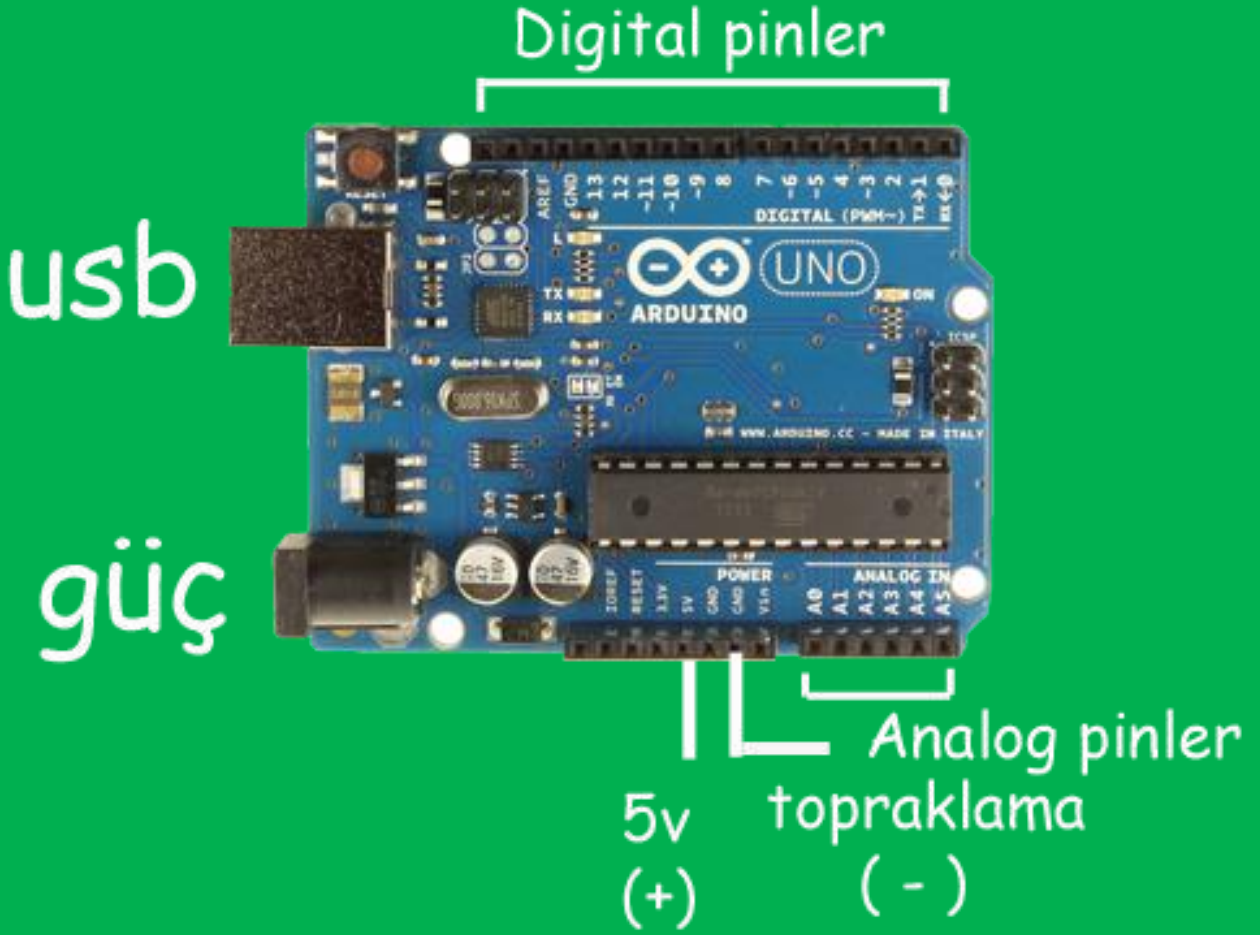
Bu klon CH340G kullanıyor.

Eğer elinizdeki klon Usb çipi olarak CH340G kullanıyorsa linkteki sürücüyü indirip içindeki CH341SER klasörünü bulun "setup.exe" programını bilgisayarınıza kurun.

Eğer 64 bit Windows kullanıyorsanız kurulumu garantilemek adına "DRVSETUP64" klasöründen "DRVSETUP64.exe" programını bir üst klasör olan "CH341SER" klasörüne kopyalayın.

<http://www.arduinoed.eu/files/CH341SER.zip>

Arduinomuzu Tanıyalım



Arduino'muzun üzerinde;

- ⊗ Bilgisayar bağlantısını yapabilmemiz için bir Usb girişi.
- ⊗ Usb bağlı değilken güç sağlaması için bir güç girişi .
- ⊗ 5Volt ve 3.3Volt (+) güç çıkışı.
- ⊗ Topraklama (-) çıkışları. Üzerinde Gnd yazan 3 çıkış.
- ⊗ 5 adet analog pin
- ⊗ 14 adet digital pin

mevcuttur.

Scratch Nedir?



Scratch MIT (Massachusetts Institute of Technology) üniversitesi tarafından geliştirilen bir görsel programlama dilidir. 9-16 yaş grubu öğrenciler bu programlama dilini kullanarak çok kolay bir şekilde program yazıyor ve birbirleri ile paylaşıyorlar. Scratch’da program yazmak son derece basittir. Her komut bir bloğu temsil eder, lego yapar gibi blokları birleştirirsiniz ve programı yazarsınız.

Algoritma hakkında doküman:

<https://drive.google.com/file/d/0B6wdhABIsqQRMHdQSVI1cGhKMkU/view?usp=sharing>

Scratch Site:

<https://scratch.mit.edu/>

Scratch Görsel Eğitimler:

<http://scratch.eba.gov.tr>

Scratch E-doküman:

<http://uekae.bilgem.tubitak.gov.tr/sites/images/scratch-kitabi.pdf>

Scratch ile Arduino Kontrolü

Arduino'muzu kontrol etmek için. Biraz programlama bilgisine ihtiyacımız var. İlerleyen derslerde bu konuya değineceğiz ancak şimdilik Bilişim Teknolojileri Ve Yazılım Derslerinde öğrendiğimiz Scratch programı ile Arduino'muzu nasıl kontrol ederiz onu öğreneceğiz.

Linkteki dosyaları indirip bilgisayarımıza kuralım

<https://drive.google.com/file/d/0B6wdhABIsqQRcG5UZFd0dm5OeFE/view?usp=sharing>

Arduinokurulumdosyaları.rar içindeki

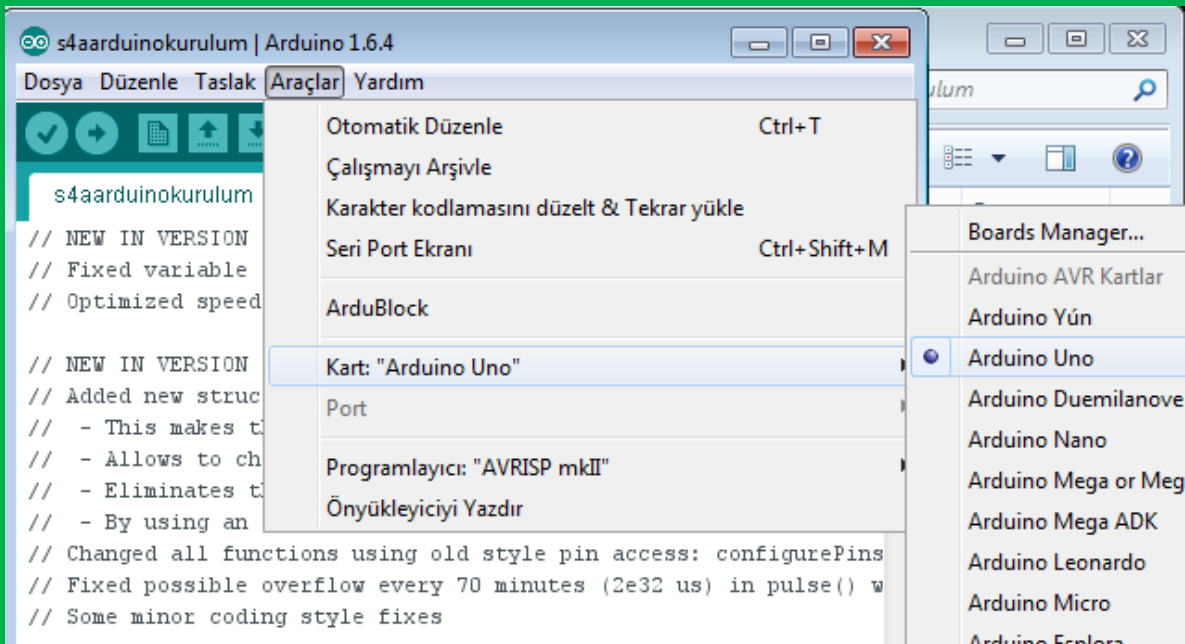
- Arduino-1.6.5-r2-windows.exe
- S4A16.exe
- ScratchKurulum.exe

dosyalarını bilgisayarınıza kurunuz.

Arduino'yu usb kablosu ile bilgisayara bağlayınız.

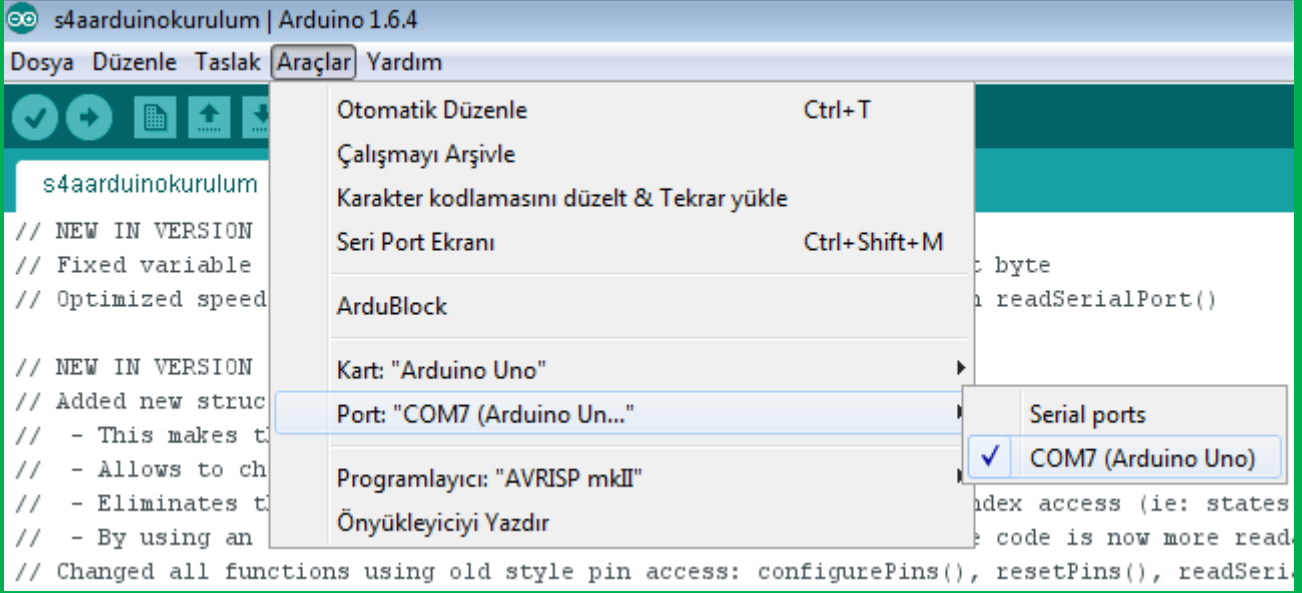
Daha sonra s4aArduinokurulum klasöründeki s4aArduinokurulum.ino dosyasını açınız.

Araçlar menüsünden kart olarak "Arduino Uno" seçiniz.



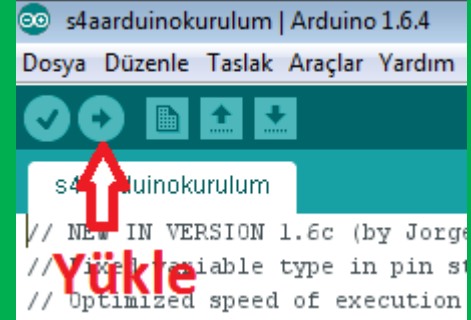
Daha sonra yine Araçlar menüsünden port olarak COM (Arduino Uno) seçiniz.

Burada bende COM7 olarak gözükmekte sizde farklı bir COM gözükabilir.



```
s4aarduinokurulum | Arduino 1.6.4
Dosya Düzenle Taslak Araçlar Yardım
Otomatik Düzenle Ctrl+T
Çalışmayı Arşivle
Karakter kodlamasını düzelt & Tekrar yükle
Seri Port Ekranı Ctrl+Shift+M
ArduBlock
Kart: "Arduino Uno"
Port: "COM7 (Arduino Uno...)"
Programlayıcı: "AVRISP mkII"
Önyükleyiciyi Yazdır
Serial ports
COM7 (Arduino Uno)
// NEW IN VERSION
// Fixed variable
// Optimized speed
// NEW IN VERSION
// Added new struc
// - This makes t
// - Allows to ch
// - Eliminates t
// - By using an
// Changed all functions using old style pin access: configurePins(), resetPins(), readSerialPort()
```

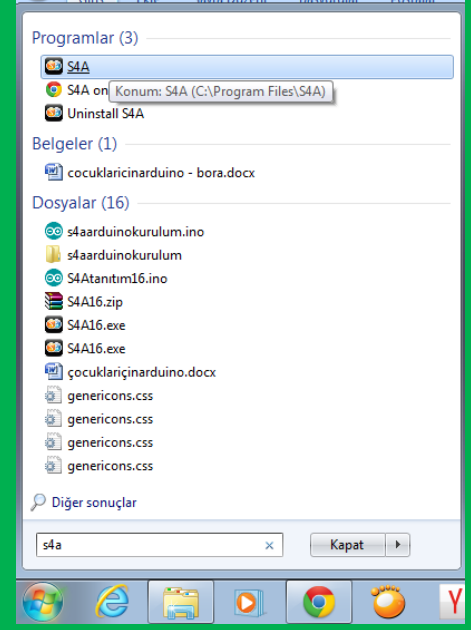
Bir sonraki işlemimiz programı Arduino'ya yüklemek. Yükle butonuna basıyoruz ve programı yüklüyoruz. Scratch ile Arduino'ya programlamak için bu programın Arduino'ya yüklü olması gerekir.



```
s4aarduinokurulum | Arduino 1.6.4
Dosya Düzenle Taslak Araçlar Yardım
Yükle
// NEW IN VERSION 1.6c (by Jorge
// Fixed variable type in pin st
// Optimized speed of execution
```



S4A programımızı çalıştırıyoruz.

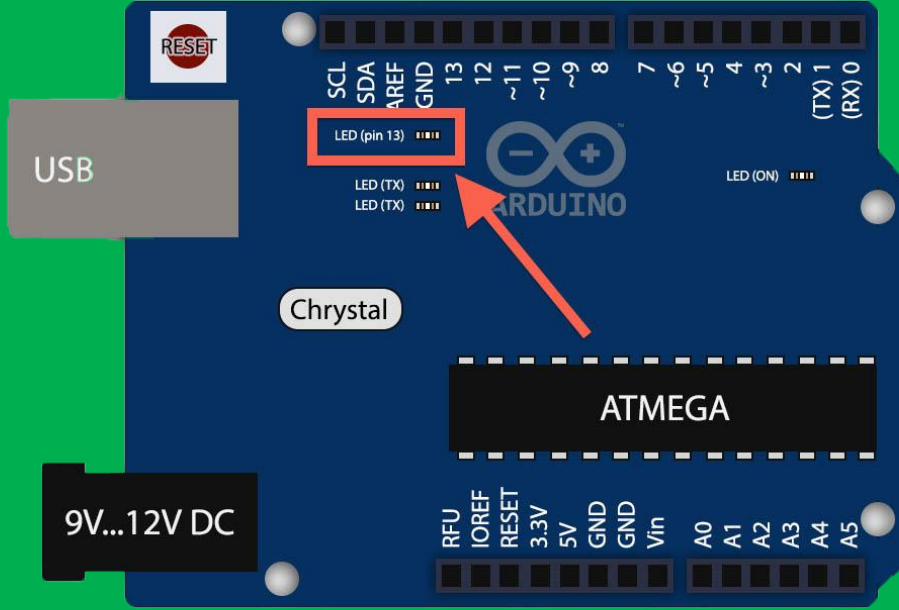


Arduino 1 port: COM7	
Analog0	558
Analog1	549
Analog2	541
Analog3	532
Analog4	559
Analog5	558
Digital2	false
Digital3	false

Eğer her şeyi doğru yapmışsak, analog pinlerden değerler okuyacaktır. Bundan sonra artık Scratch'da yazdığımız programları Arduino'ya yükleyebiliriz.

Scratch ile Arduino Üzerindeki Ledin Kontrolü

LED'ler elektrik enerjisini ışığa dönüştüren yarı iletken devre elemanlarıdır. Arduino'muz üzerinde sabit 13 nolu pine bağlı bir adet Led mevcut.



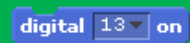
Arduino'muzu bilgisayarımıza bağlıyoruz S4A programımızı çalıştırıyoruz. Yandaki kodları Arduino karakterimize yazıyoruz ve yeşil bayrağa basıyoruz. Bu kod Arduino üzerindeki led'in yanıp 1 saniye sonra sönmesini ve 1 saniye sonra tekrar yanmasını sağlar.



Yeşil bayrak tıklandığında programı başlatır.



Programın sürekli tekrarlanmasını sağlar.



13 nolu pine güç verip led'in yanmasını sağlar.

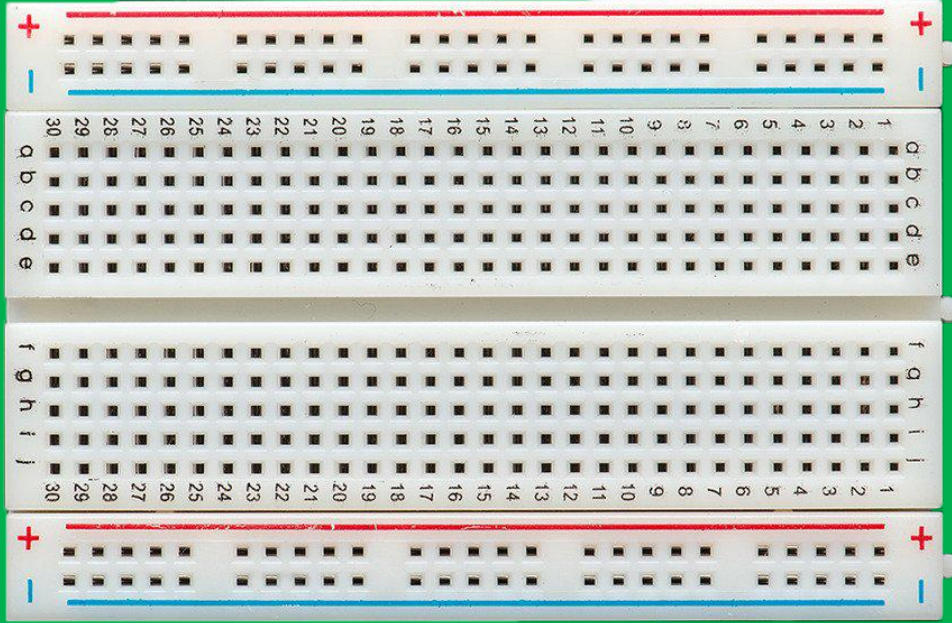


13 nolu pinden gücü kesip led'in sönmesini sağlar.

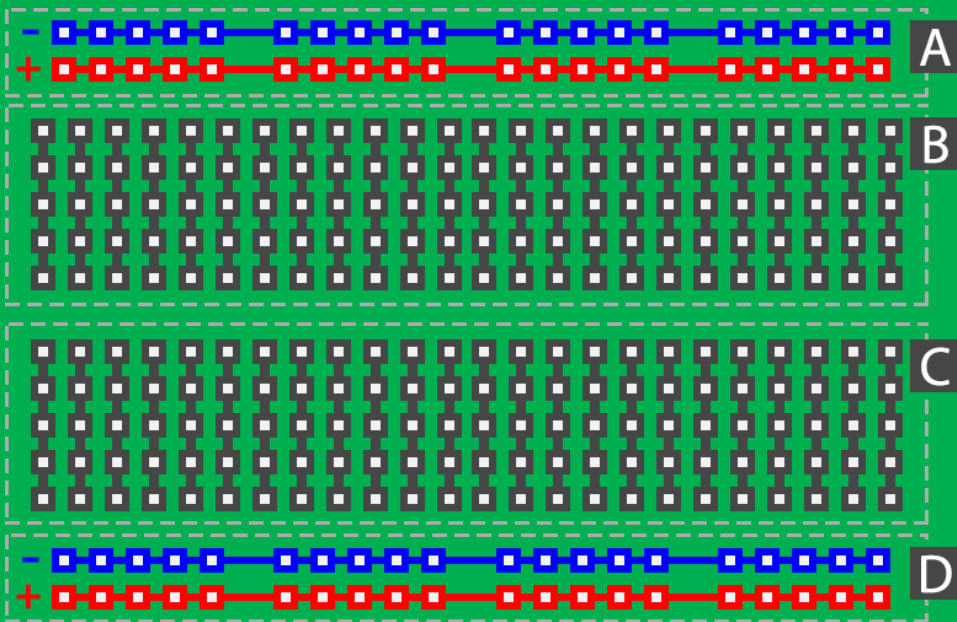


Led'lerin yanıp sönmesi arasında 1 saniye bekleme koyar.

Breadboard üzerinde çalışmak

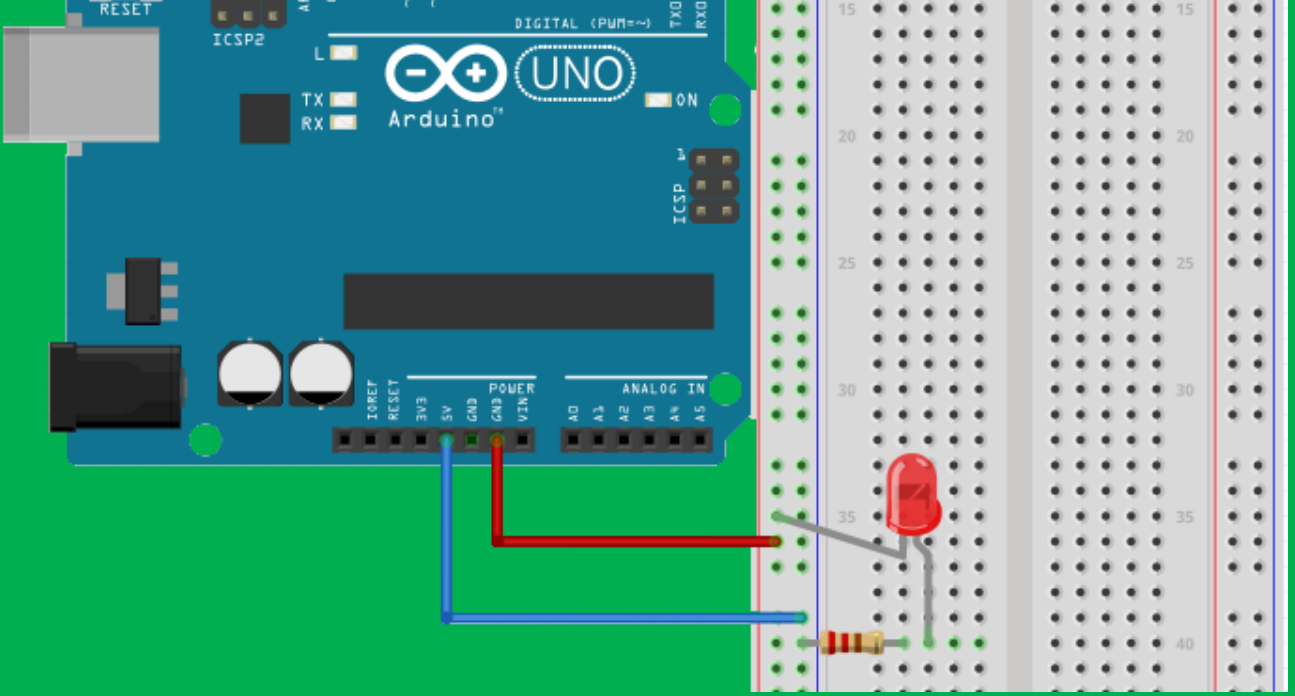


Breadboard devre elemanlarını birbirine bağlamak için kullanılır. Tekrar tekrar kullanabilirsiniz.



A ve D kısımları yatay olarak birbirine bağlıdır. B ve C kısımları ise dikey olarak birbirine bağlıdır.

Uygulama



Gerekli Malzemeler

1 adet Arduino Uno

1 adet breadboard

1 adet LED

1 adet 220 Ω (ohm) direnç

2 adet erkek-erkek jumper kablo

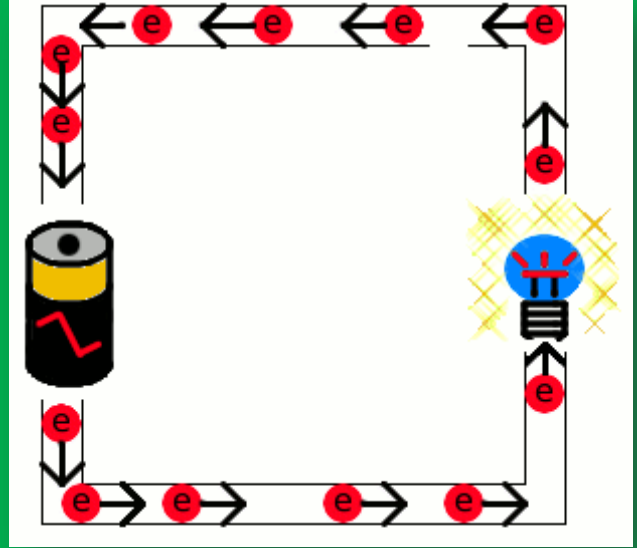
Arduino ile programlama uygulamalarımıza Led yakma ve söndürme ile başlayacağız. Nasıl ki okuma yazma öğrenirken önce harfleri öğreniyorsunuz, Arduino öğrenirken de önce Led yakıp söndürmeyi öğrenmelisiniz. Bu uygulamayı yaparken herhangi bir program yazmayacağız. Sadece Arduino'yu bir güç kaynağı gibi kullanarak Led'imizi yakacağız. Uygulamamıza geçmeden önce isterseniz kullanacağımız devre elemanlarını yakından tanıyalım ve temel elektronik hakkında bilgi edinelim.

Elektronik

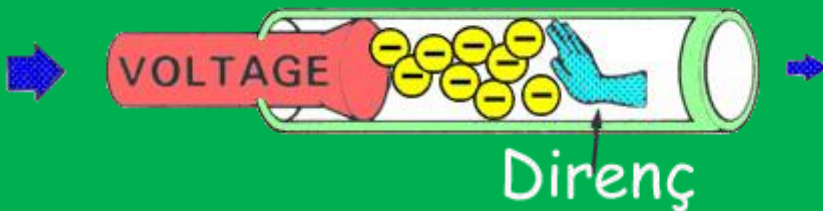
Elektronik, elektronik aygıtları çalıştırmak için küçük elektrik akımlarının nasıl kumanda edilmesi gerektiğini inceleyen bilim dalıdır. Kullandığımız bilgisayar,televizyon, akıllı telefonlar hepsi elektronik bilimi sayesinde hayatımıza girdi. Bizde kitabımızda bazı elektronik kavramlarını ve elektronik devre elemanlarını inceleyeceğiz. İnceleyeceğimiz devre elemanları, kullandığımız bütün elektronik aletlerde bulunan parçalardır.

Elektrik Akımı

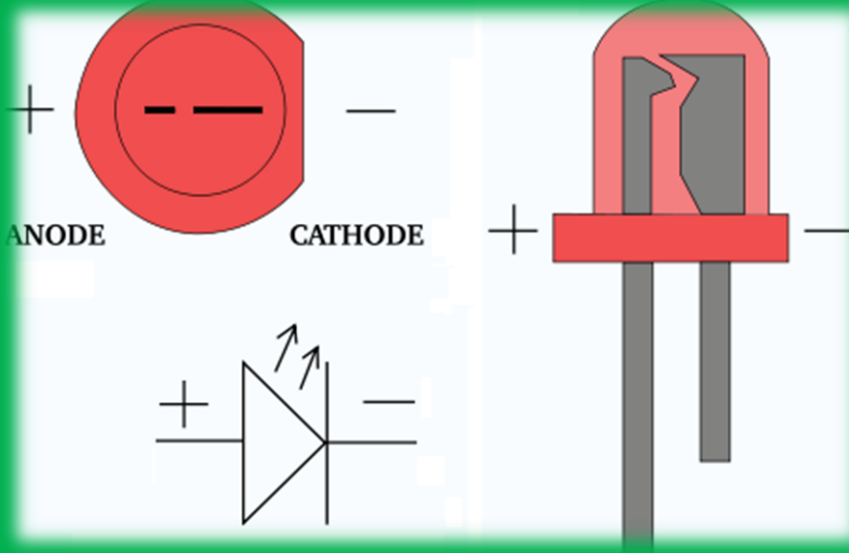
Elektrik akımı, elektron adı verilen çok küçük parçacıkların bir tel boyunca akışıdır. Bütün maddeler elektron içerir. Elektronlar her şeyi oluşturan atomların bir parçasıdır.



Elektronlar pilin (+) ucundan (-) ucuna doğru hareket ederler. Bu hareket sonucu bir gerilim oluşur, biz bu gerilime Voltaj diyeceğiz ve V ile göstereceğiz. Elektronların (+) uctan (-) uca doğru hareketine akım diyeceğiz ve I ile göstereceğiz. Akım (I) ve gerilim (V) bir devrenin temel bileşenlerinden ikisidir. Üçüncü temel bileşen dirençtir. Direnç, kimi maddelerin elektron akışına yani akıma karşı gösterdiği zorluktur.



Elektronik devre elemanları

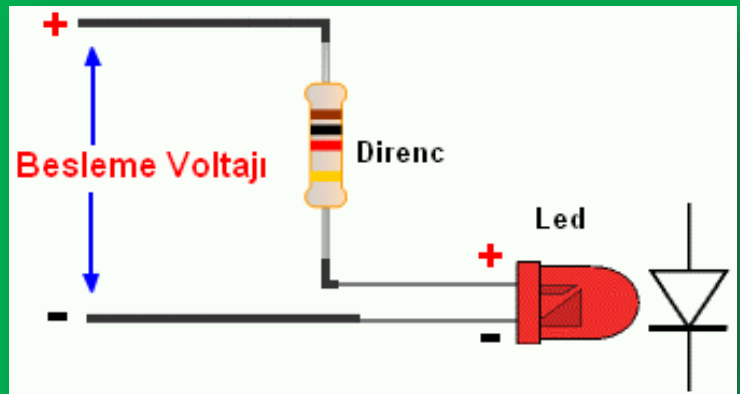


LED

LED'ler, üzerinden akım geçtiğinde ışık yayan devre elemanlarıdır. Uzun ucu (+) kısa ucu (-). Yalnız dikkat etmemiz gereken husus: Asla LED'imizi direnç olmadan devremize bağlamayacağız. Çünkü LED'ler hassas devre elemanlarıdır, uzun süre yüksek akıma maruz kaldığında yanabilir ve bir gaz açığa çıkabilir! Bu gaz tehlikelidir ve asla solumayınız. Direnç akıma zorluk göstererek akımı azaltacak ve LED'imize uygun hale getirecektir.

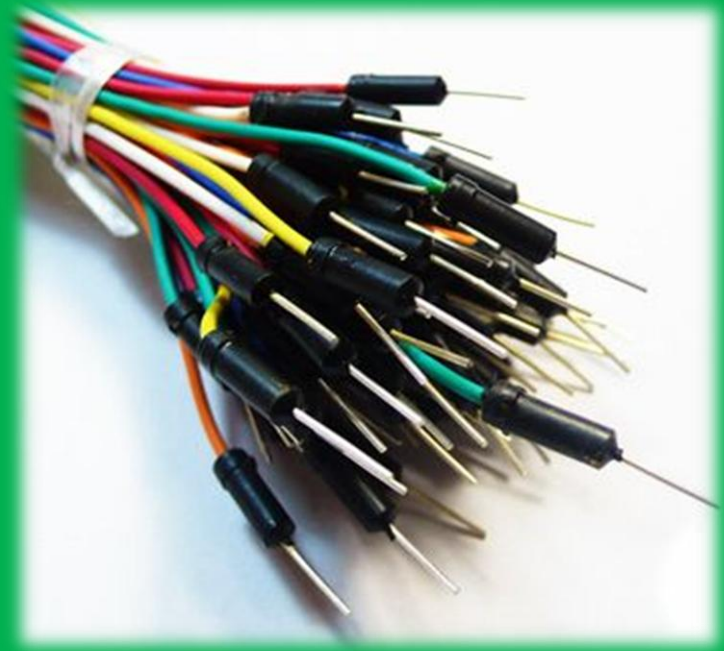
Direnç:

Dirençler akıma zorluk gösteren devre elemanlarıdır. Devremizde LED'in (+) ucuna her zaman bir direnç bağlayacağız

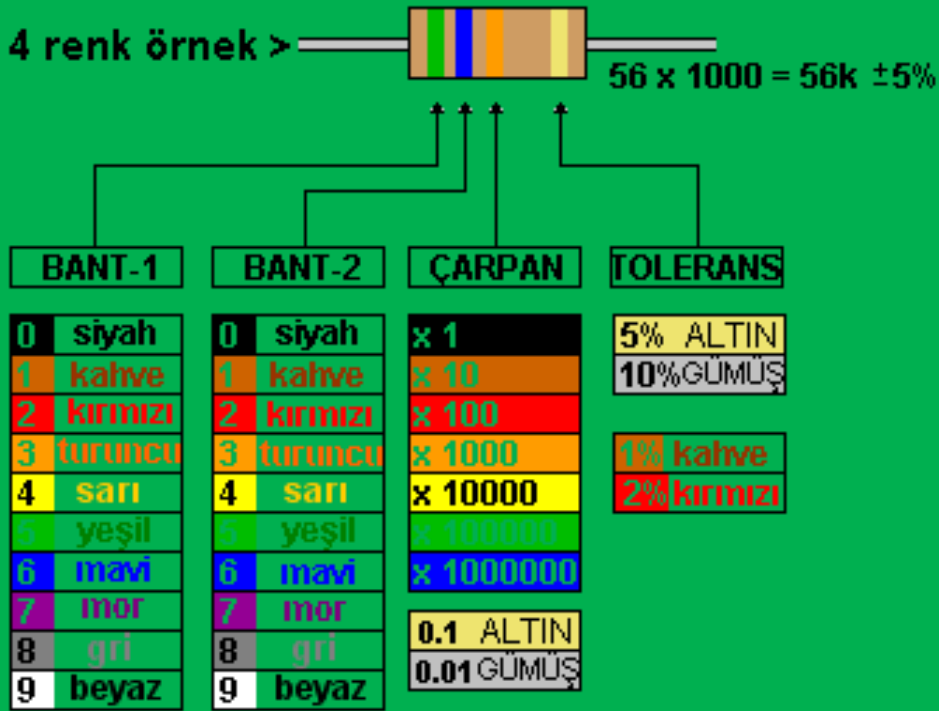


Jumper kablo:

Devre elemanlarını birbirine bağlamak için kullanılır. Bir ucunu Arduino'muza diğer ucunu da breadboarda bağlayacağız.



Dirençlerin akıma gösterdikleri zorlukların dereceleri vardır. Bunlar renklerle sembolize edilir ve birimi OHM'dur simgesi Ω dur. Şimdi bu renklere göre dirençlerimizin değerlerini nasıl hesaplayacağız görelim.

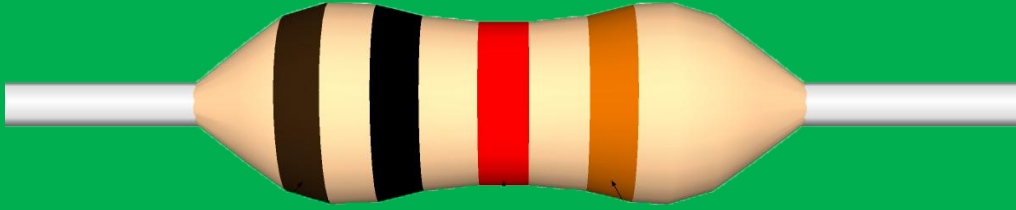


Dirençlerin üstünde renkler vardır. Bunlardan ilk ikisi katsayı, üçüncü çarpan, sondaki de toleransdır.



$$2 \quad 2 \quad 10^1=10 \quad = \quad 220\Omega$$

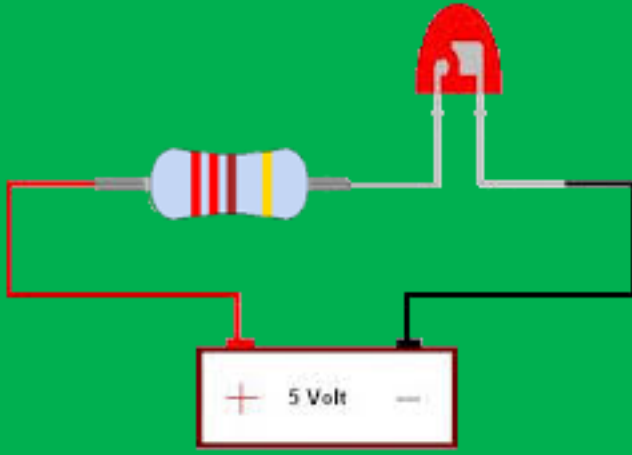
Yukarıdaki direnci ele alırsak ilk iki renk katsayı rengidir. Kırmızı renginin rakamsal karşılığı 2'dir. İki tane katsayı rengi olduğundan kırmızı kırmızı 22 yapar. Üçüncü renk yani çarpan rengi kaç tane sıfır ekleyeceğimizi söyler. Burada çarpan kahverengidir. Kahverenginin rakamsal karşılığı 1'dir yani 1 sıfır gelir. 22'nin yanına 1 sıfır eklenince 220 yapar. Son renk, altın rengi toleransı ifade eder.



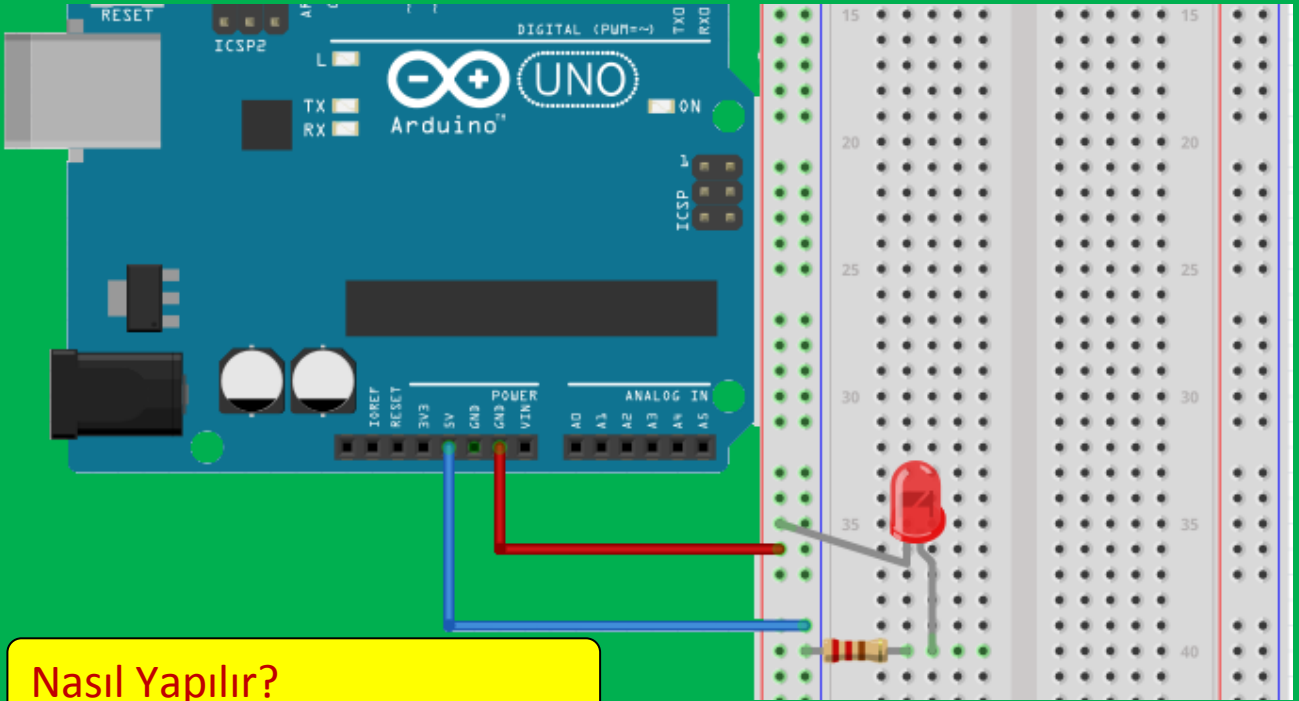
$$1 \quad 0 \quad 10^2=100 \quad = \quad 1000\Omega=1K\Omega$$



$$3 \quad 3 \quad 10^1=10 \quad = \quad 330\Omega$$



Burada devre 5V bir güç kaynağı ile beslenmiş. Arduino da 5V çıkış sağlar. Direnç olarak kırmızı, kırmızı, kahverengi yani 220Ω direnç kullanılmış. Led'in (+) ucu dirence bağlı. Biz bu devreyi Arduino ile yapacağız.

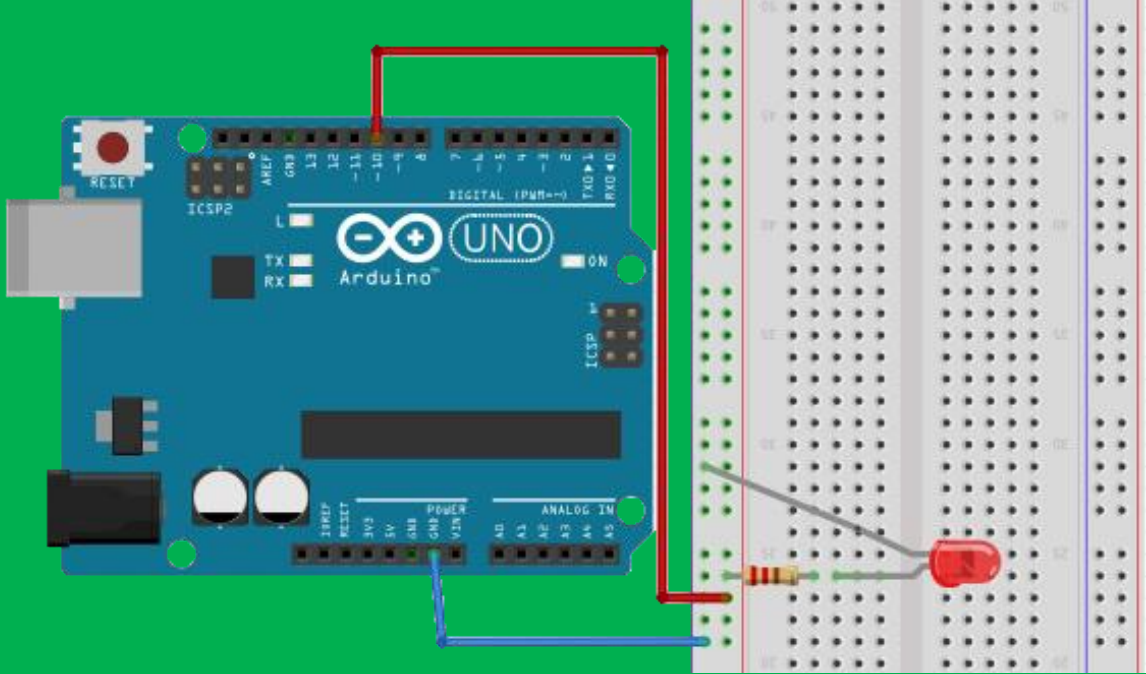


Nasıl Yapılır?

Breadboardun yan tarafında bulunan kısım paralel olarak birbirine bağlıdır. Kablonun bir ucunu Arduino'nun 5V pinine, diğer ucunu paralel olan kısımlardan birine (+ kısma) takınız. Artık o paralelde her yerde 5V var. Direncimizi de bu bölgede bir yere bağlıyoruz. Yani direncimiz Arduino'nun 5V girişine bağlanmış oluyor. Breadboardun orta kısmı ise dikey olarak birbirine bağlıydı, dikey kısımda direncin bağlı olduğu yere led'imizin uzun ucunu (+) takıyoruz. Diğer ucu da Arduino'nun GND (-) kısmını bağladığımız paralel olan kısma takıyoruz. Bu şekilde akım tamamlanmış oluyor. Bağlantıları yaptıktan sonra Arduino'ya usb girişini veya güç girişini bağlayınız. Led yanacaktır.

Breadboardun çalışma yapısına sayfa 13'ten bakabilirsiniz.

Breadboard üzerindeki ledi yakıp söndürme



Bu örnekte ise 5V pini yerine digital pinlerden olan 10 numaralı pini kullandık. 10 numaralı pini güç çıkışı olarak kullanacağız. 10 numaralı pinden çıkan akım dirençten geçerek Led'in (+) ucuna oradan da Arduino'nun topraklama (GND (-)) kısmına gelerek devreyi tamamlayacak.



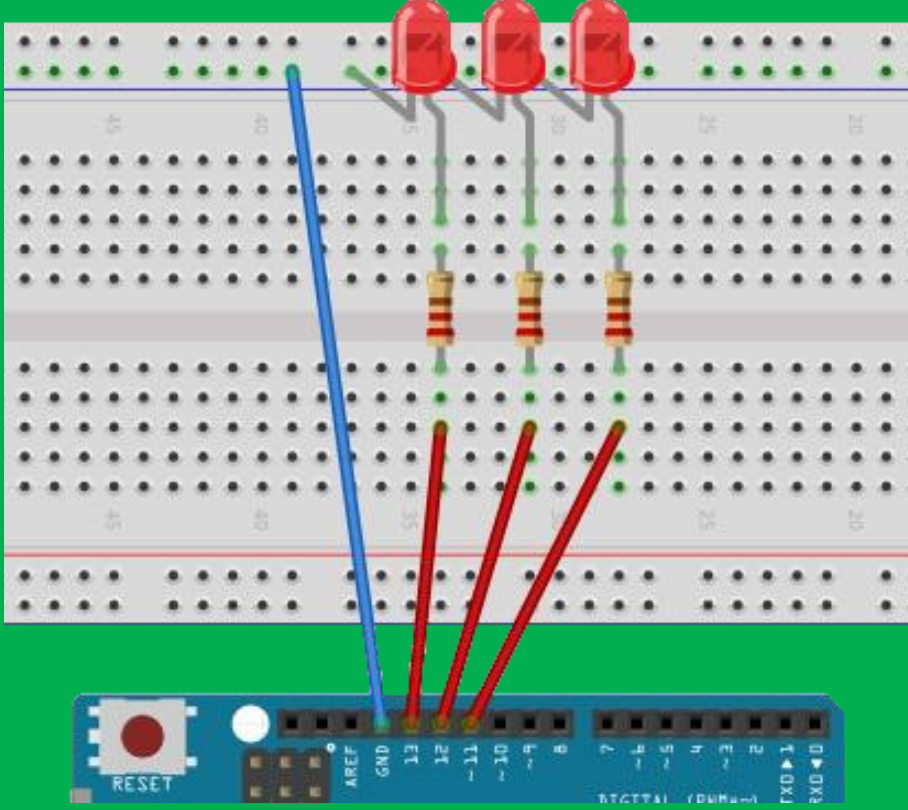
[Youtube videosu için tıklayınız.](#)

Daha önce s4aArduinokurulum.ino dosyasını Arduino'muza yüklemiştik. Bu program Scratch ile Arduino'nun haberleşmesi için gerekmektedir.

Bu örnekte ise S4A programımızı açıyoruz ve yandaki kodları yazıp yeşil bayrağa tıklıyoruz. Led'imiz 1 saniye yanıp 1 saniye sönecektir ve bu durum sürekli tekrarlayacaktır. Burada digital 10 on komutu: 10 nolu pine akım verir, led yanar.

Digital 10 off komutu ise: 10 nolu pinden akımı keser, led söner.

Yürüyen Işık Uygulaması



Gerekli Malzemeler

- 1 adet Arduino Uno
- 1 adet Breadboard
- 3 adet 220Ω direnç
- 3 adet led
- 4 adet erkek-erkek jumper kablo

Bu devrede 13,12,11 no.lu pinlerimizi dirençlerimize bağladık. Her dirence de 1 adet Led bağladık ve Led'lerin (-) ucunu Arduino'muzun GND(-) kısmının bağlandığı paralel kısma bağladık.

```
tklandığında
Sürekli
digital 13 on
0.2 saniye bekleyin
digital 13 off
digital 12 on
0.2 saniye bekleyin
digital 12 off
digital 11 on
0.2 saniye bekleyin
digital 11 off
```

Bu uygulamamızda digital 13,12,11 no.lu pinlere bağlı Led'leri sırasıyla yakıp söndüreceğiz.

Oluşturduğumuz kodun yapacağı işler sırasıyla:

Digital 13'e bağlı Led'i yakar, 0.2 saniye bekler, söndürür. O sönerken digital 12'ye bağlı Led'i yakar, 0.2 saniye bekler, söndürür. O sönerken digital 11'e bağlı Led'i yakar, 0.2 saniye bekler söndürür. Sonra tekrar başa döner, digital 11'e bağlı Led sönerken digital 13'e bağlı Led yanar. Bu sürekli olarak tekrar eder.

Biz burada 3 adet Led kullandık. Siz digital 10 ve digital 9'a aynı şekilde birer adet Led bağlayıp Led sayısını arttırabilirsiniz. Dikkat edin direnç bağlamayı unutmayın!

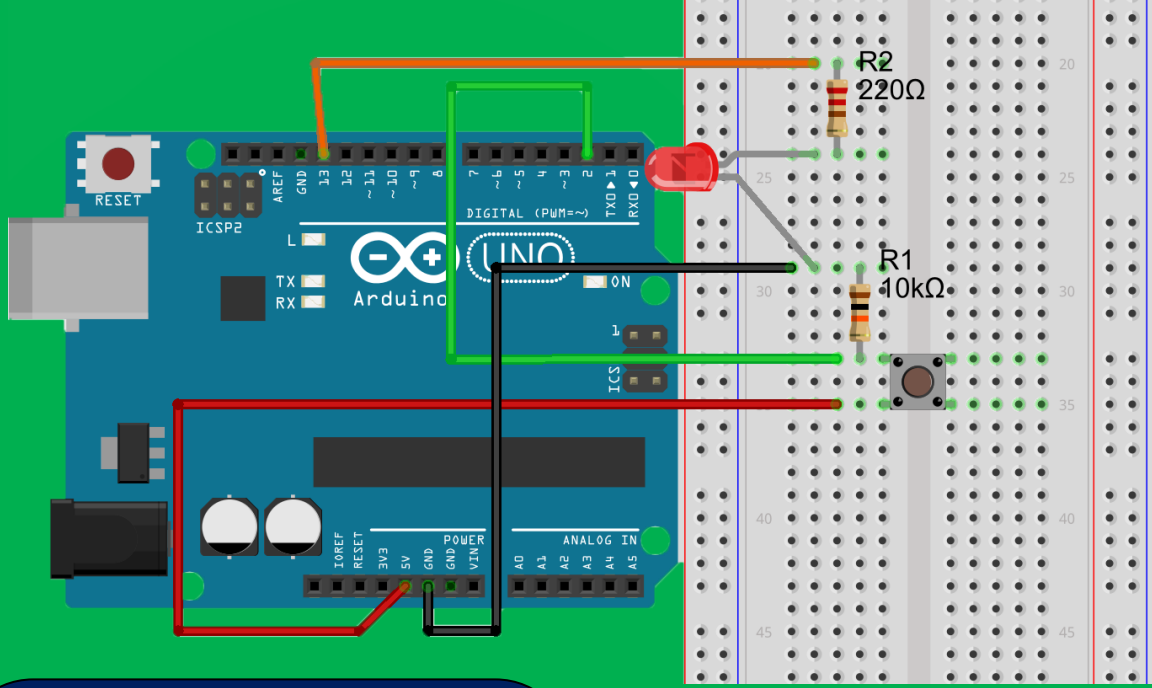
Devre bağlantısını yaptıktan sonra S4A program kodunuza ekleme yapmalısınız ki digital 10 ve digital 9'a güç verip Led'leri yakabilesiniz. Bu kodları ekleme işini size bırakıyorum. Biraz düşünün ve deneme yanılma yolu ile doğru blokları ekleyin.

Biz burada 220Ω direnç kullandık. Peki başka direnç kullanabilir miyiz? 220Ω altında direnç kullanmayın ancak 220Ω üstünde direnç kullanabilirsiniz. 220Ω üzerinde direnç kullandığınızda akım düşecektir. Sizde 220Ω dirençleri 1000Ω ($1K\Omega$) dirençlerle değiştirin. Aradaki farkı gözlemleyin.

Bir devrenin temel bileşenleri gerilim(V), akım(I) ve direnç(R) demiştik. Bunlar arasında şöyle bir ilişki vardır.

$V=IxR$, gerilim=akımxdirenç tir. Sabit gerilim altında direnci arttırırsanız akım azalır, bu azalma da Led'in parlaklığının düşmesine neden olur.

Buton ile Led Yakma



Gerekli Malzemeler

- 1 adet Arduino Uno
- 1 adet Breadboard
- 1 adet Push buton
- 1 adet Led
- 1 adet 220Ω direnç
- 1 adet 10KΩ direnç
- 4 adet Jumper kablo

Butunun bir ucu digital 2 pinine bağlıdır. Butona basıldığında digital 2 pininin değeri true (doğru ,basıldı) olur.

Kullandığımız yapı bir eğer ise değilse yapısı.

eğer ;

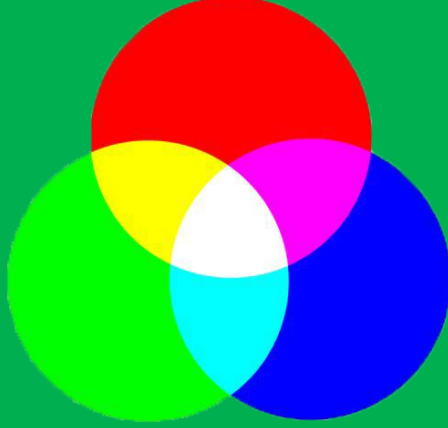
butona basılmış ise Led'in bağlı olduğu digital 10 pinine güç ver, Led'i yak.

Değilse;

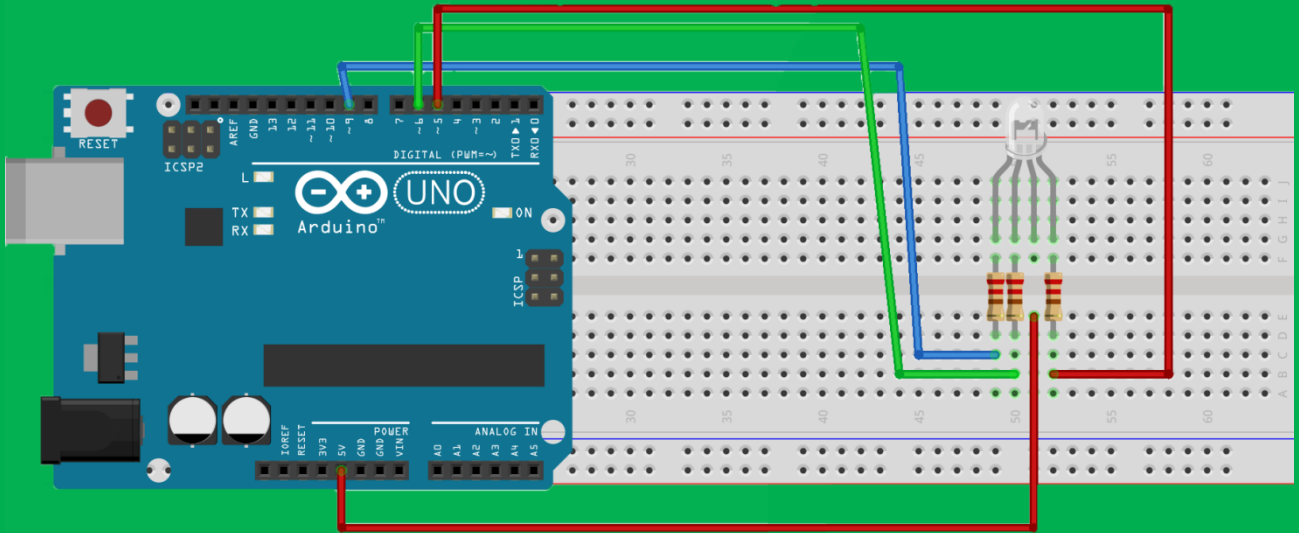
Digital 10 pininden gücü kes, Led'i söndür



RGB (Red, Green, Blue) Led



Üç ana renk vardır: Kırmızı, yeşil ve mavi. Diğer renkler bu renklerin belli oranda karışımından meydana gelir. RGB Led'ler de bu mantıkla çalışır. Üzerindeki 3 adet bacak renklerle ilgilidir. Eğer sadece kırmızı bacağından tam akım geçerse Led kırmızı yanar. Sadece mavi bacağından tam akım geçerse mavi yanar. Hem kırmızı hem de mavi bacağından tam güç geçerse magenta renginde yanar.



Gerekli Malzemeler

- | | |
|------------------------|--------------------|
| 1 adet Arduino Uno | 1 adet Breadboard |
| 1 adet RGB Anot(+) Led | 3 adet 220Ω direnç |
| 4 adet Jumper kablo | |

İki tür RGB Led var. Anot(+) ve Katot(-). Biz devremizde 5v'ye bağlanan anot Led kullandık. Led'in uzun ucu 5v'ye bağlanacak. Türkiye'de genellikle anot Led satılır.

```
m tuşu basıldığında
analog 9 value 0
analog 6 value 255
analog 5 value 255
```

```
y tuşu basıldığında
analog 9 value 255
analog 6 value 0
analog 5 value 255
```

```
k tuşu basıldığında
analog 9 value 255
analog 6 value 255
analog 5 value 0
```

M tuşuna basıldığında (Analog 9 value 0) komutu 9 nolu pinden gönderilen akımı 0 yapar ve mavi led'den akım geçer. (Analog 6 value 255 ve Analog 5 value 255) komutları 6 ve 5 pinlerine tam güç vererek akımın geçmesini engellerler, böylece sadece mavi renk yanar

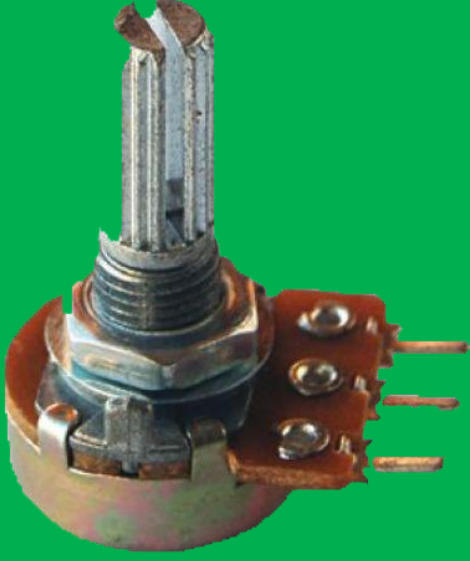
Y tuşuna basıldığında ise (Analog 9 value 255 ve analog 5 value 255) 9 ve 5 pinlerine tam güç verir, (analog 6 value 0) 6 no.lu pinden gücü keser ve yeşil yanar.

K tuşuna basıldığında ise (Analog 9 value 255 ve analog 6 value 255) 9 ve 6 pinlerine tam güç verir, (Analog 5 value 0) 5 no.lu pinden gücü keser ve kırmızı yanar.

Bütün uygulamaların .sb uzantılı dosyalarının ve devre şemalarının olduğu bir klasör hazırlandı. Siz bu klasördeki dosyaları kullanarak uygulamalarınızı geliştirebilirsiniz. Tek yapmanız gereken devre şemasını incelemek, breadboard üzerinde şemayı tamamlamak ve gerekli Scratch dosyasını S4A programında açarak uygulamanızı yapmak.

[Kaynak Dosyaları İndirmek için Tıklayınız!](#)

Potansiyometre



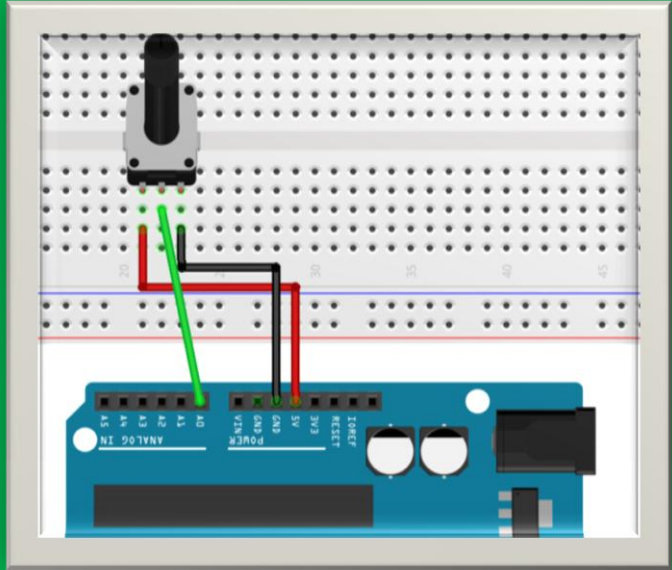
Potansiyometreler, ayarlanabilir dirençlerdir. Kullandığımız cihazlardaki sesi arttırıp azaltmaya yarayan düğmeler potansiyometre ile yapılır. Potansiyometreyi ayarlayarak direnci azaltırız ses artar, direnci arttırırız ses azalır.

Potansiyometrenin üç bacağı vardır. En soldaki bacağı (+) ya , orta bacağı analog 0'a, en sağdaki bacağı ise (-) ye bağlayarak uygulamalarımızı yapacağız.



[Youtube videosu için tıklayınız.](#)

Bu uygulamamızda değişkenleri kullanacağız, isterseniz önce değişken kavramından bahsedelim.



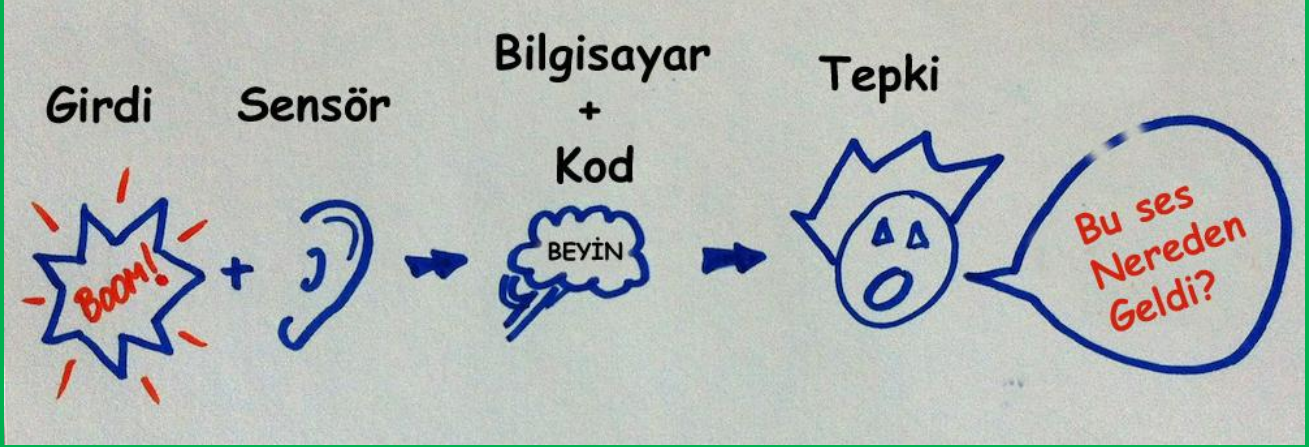
Biz programlamada değişkenleri, bazı değerleri bilgisayar hafızasında tutmak için kullanırız. Bu örnekte potansiyometre ile analog 0 pininden değerler okuyacağız. 0-1023 arası okuyacağımız değerlerle işlem yapabilmek için bu değeri bir değişkene aktaracağız. Potansiyometreden okunacak değer değiştikçe bizim değişkenimizin değeri de değişecek.

[Video](#)

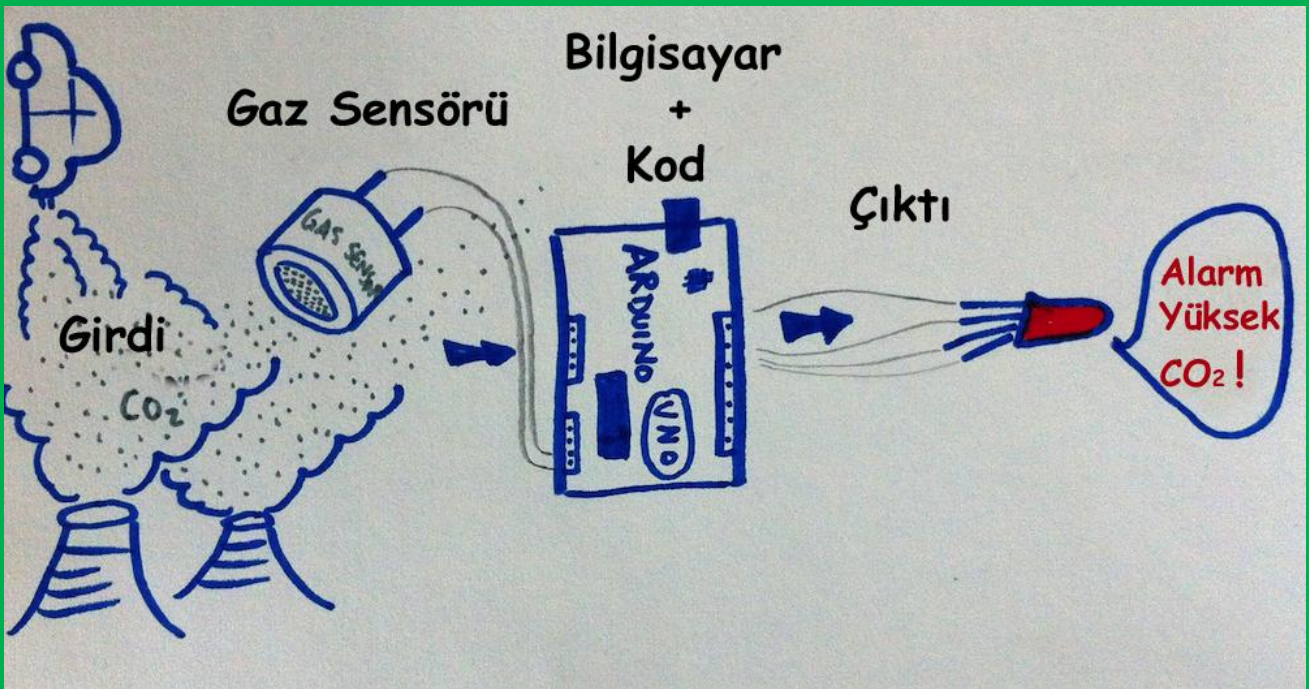


Elektronik devre elemanları: Sensörler

İnsanlar çevrelerindeki dünyayı ve çevrelerinde olup bitenleri algılamak için duyu organlarını kullanır.



Makineler (Bilgisayarlar, Robotlar) ise çevrelerinde olup bitenleri algılamak için sensörleri kullanır.

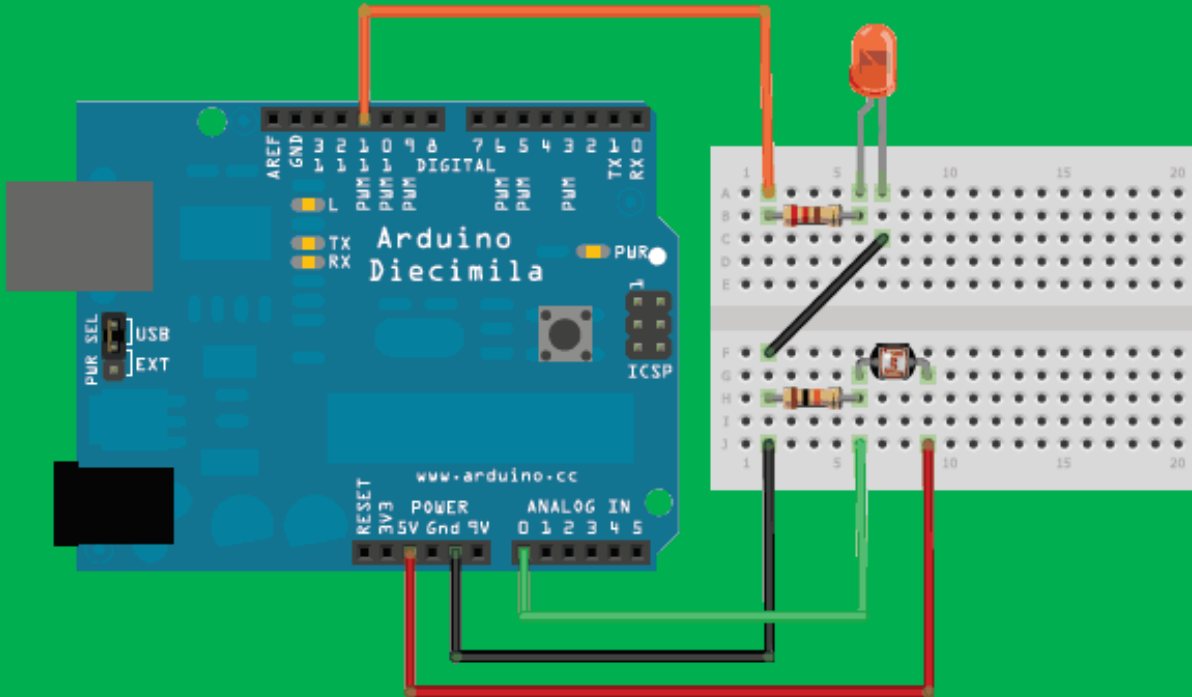


Biz de Arduino'muzun çevresi ile etkileşim kurduğu projeleri yapmak için sensörleri kullanacağız

Hava Kararınca Yanan Led

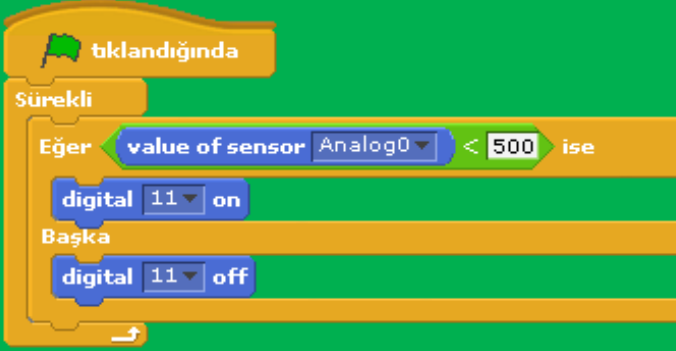
LDR (Foto Direnç)

LDR'ler ışığa duyarlı dirençlerdir. Aydınlıkta dirençleri minimum, karanlıkta ise maksimumdur. Biz bu özelliğini kullanarak gece olduğunda yani karanlıkta yanan Led yapacağız. Hava kararınca Ldr'nin direnci maksimum olacak ve yazacağımız kod ile bu durum gerçekleştiğinde Led yanacak.



Gerekli Malzemeler

- | | |
|---------------------------------|---|
| 1 adet Arduino uno | 1 adet 220Ω direnç (Led'e bağlanacak) |
| 1 adet breadboard | 1 adet 10000Ω (10KΩ) direnç (Ldr'ye bağlanacak) |
| 1 adet Ldr(foto direnç) | 1 adet Led |
| 5 adet erkek-erkek jumper kablo | |



Eğer sensör değeri 500'den küçük ise, Led'imizin bağlı olduğu digital 11 pini aktif olacak yani Led yanacak. Sensör değeri 500'den küçük değilse, digital 11 pini pasif olacak, giden akım kesilecek, Led sönecek.

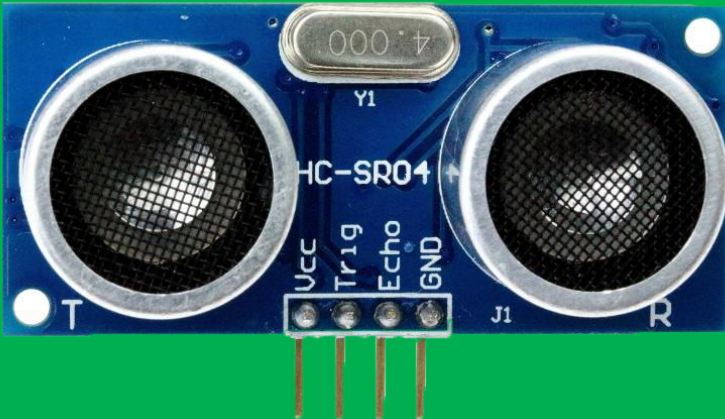
Bu uygulamada LDR'nin bir ucu Arduinonun analog 0 pinine bağlıdır ve buradan değeri okumaktadır. Bu değeri "value of sensor analog 0 (analog 0 pinine bağlı sensörün değeri)" komutu ile kontrol ederiz.

Bu pinden 0 ile 1023 arası değerler okunur. 1023 aydınlık, 0 karanlık. Işığın değerine göre de 0-1023 arası değer okunur.

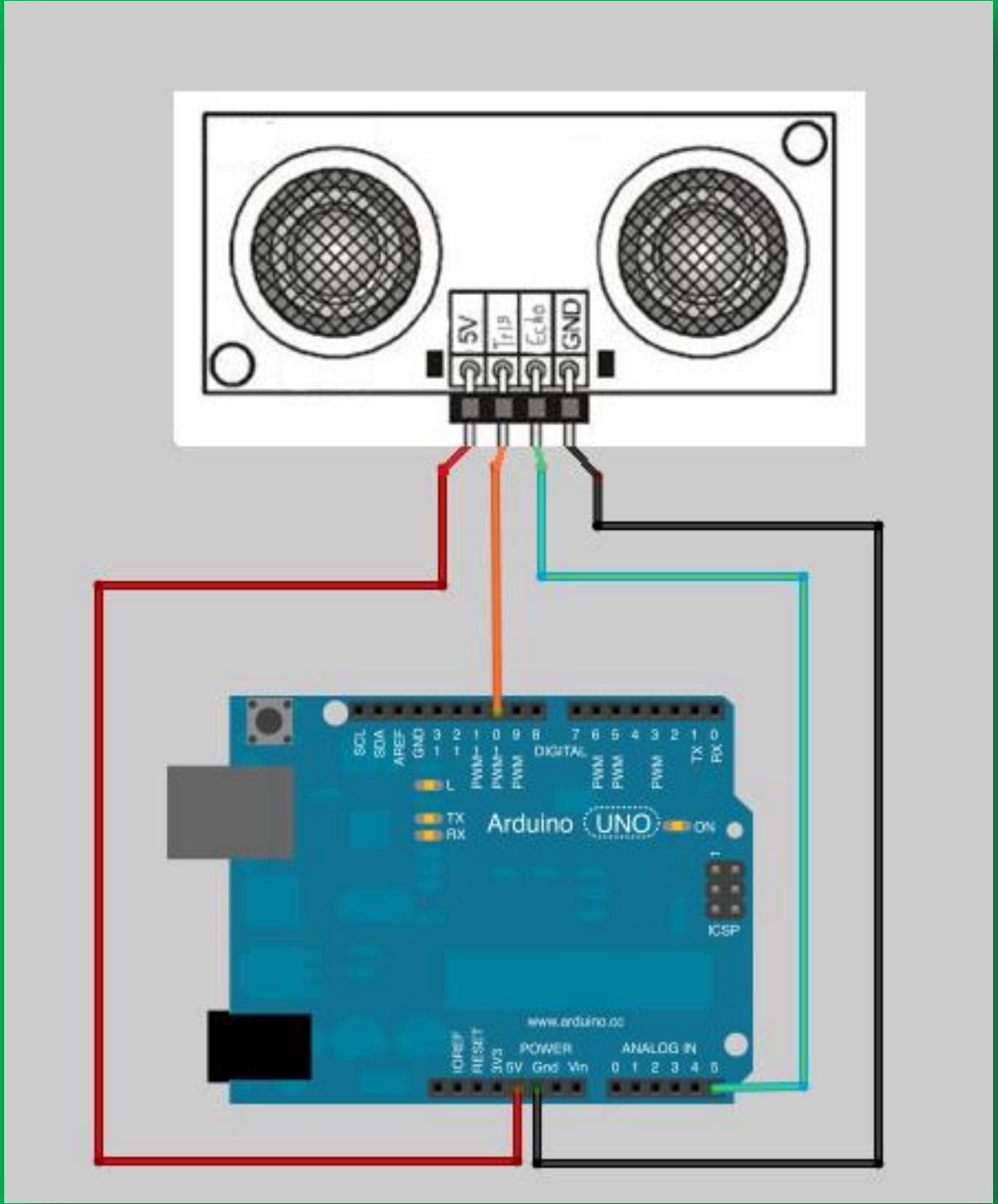
Burada bir sürekli bloğunun içerisinde bir eğer ise değilse yapısı kullandık. Bir koşul belirledik. Sensör değeri 500'ün altında mı, değil mi?

Mesafe Sensörü

Ultrasonik mesafe sensörünü projelerimizde cisimlerle olan mesafeleri ölçmek



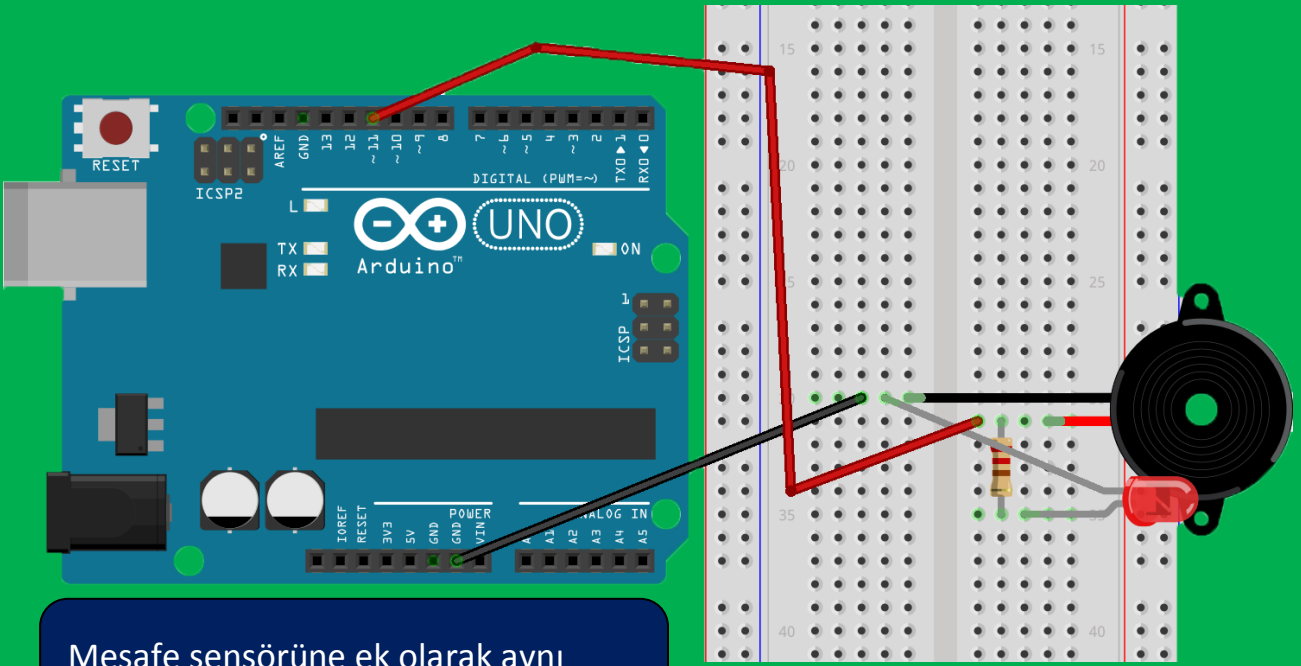
için kullanırız. Engelleri algılayıp aşan robotlarda sıkça kullanılır. Biz arabalardaki sistemlere benzer bir mesafe sensörü devresi yapacağız.



Mesafe sensörünün bağlantı şeması

Gerekli Malzemeler

1 adet Arduino Uno, 1 adet Breadboard, 1 adet HC-SR04 ultrasonik sensör
1 adet Buzzer, 1 adet Led, 1 adet 220Ω direnç, 6 adet Jumper kablo



Mesafe sensörüne ek olarak aynı board üzerine buzzer devresini kuruyoruz.

Buzzerlar devlerde bip seslerini çıkarmak için kullanılan devre elemanlarıdır.

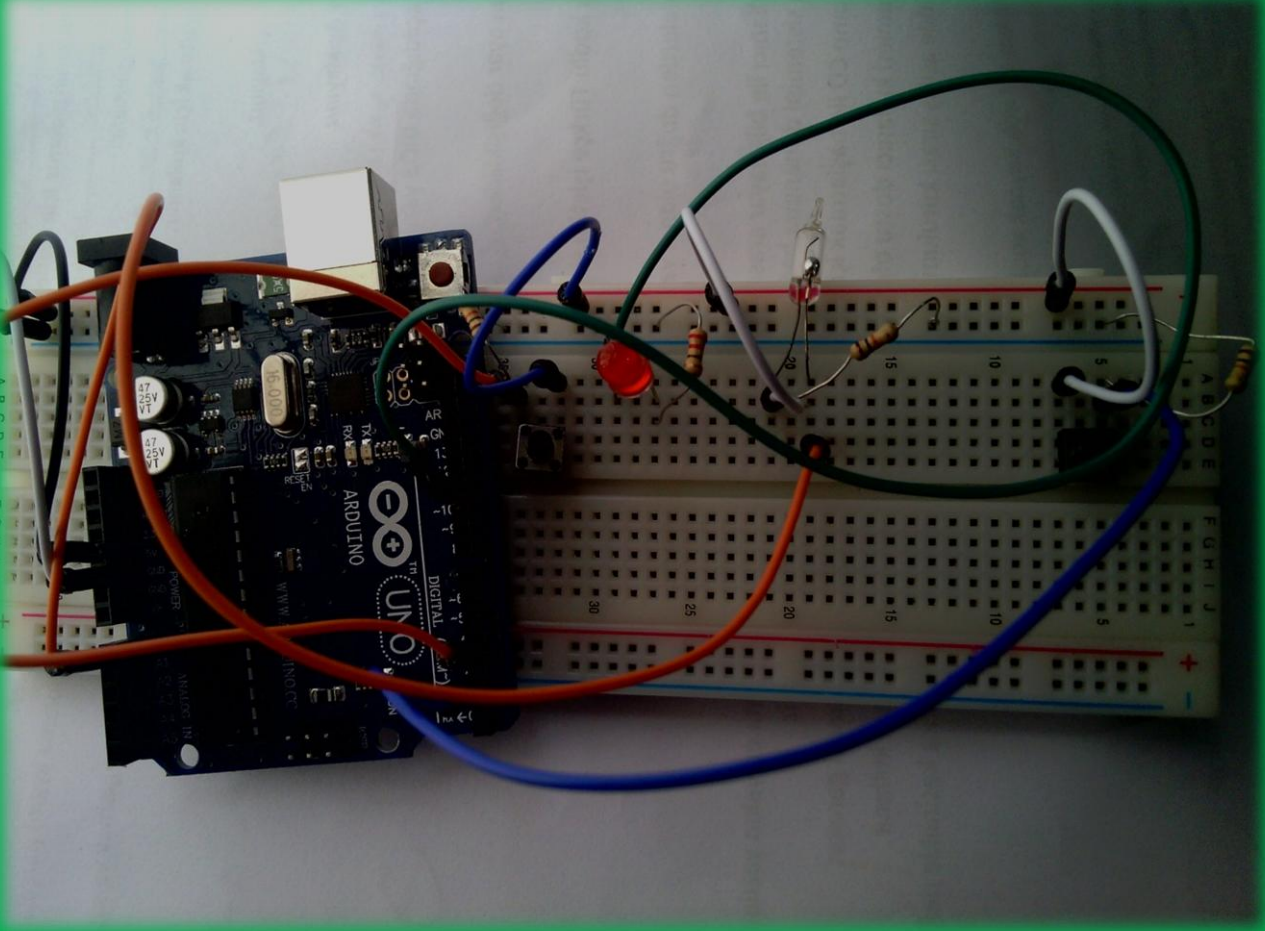


[Video](#)

```
when flag clicked
  set distance to 0.3
  loop
    if value of sensor Analog5 < 20
      set digital 11 to on
      set distance to saniye bekleyin
      set digital 11 to off
      set distance to saniye bekleyin
    if value of sensor Analog5 < 10
      set distance to 0.4
    else
      set distance to 1
    if value of sensor Analog5 < 5
      set distance to 0.1
```

Mesafe sensörünün çalışabilmesi için Arduino karta bir program yüklememiz lazım. Bu program kaynak kodların olduğu dosyada s4amesafeSensörüYa pımı klasöründe

Civalı Sensörle Oyun Kolu Yapımı



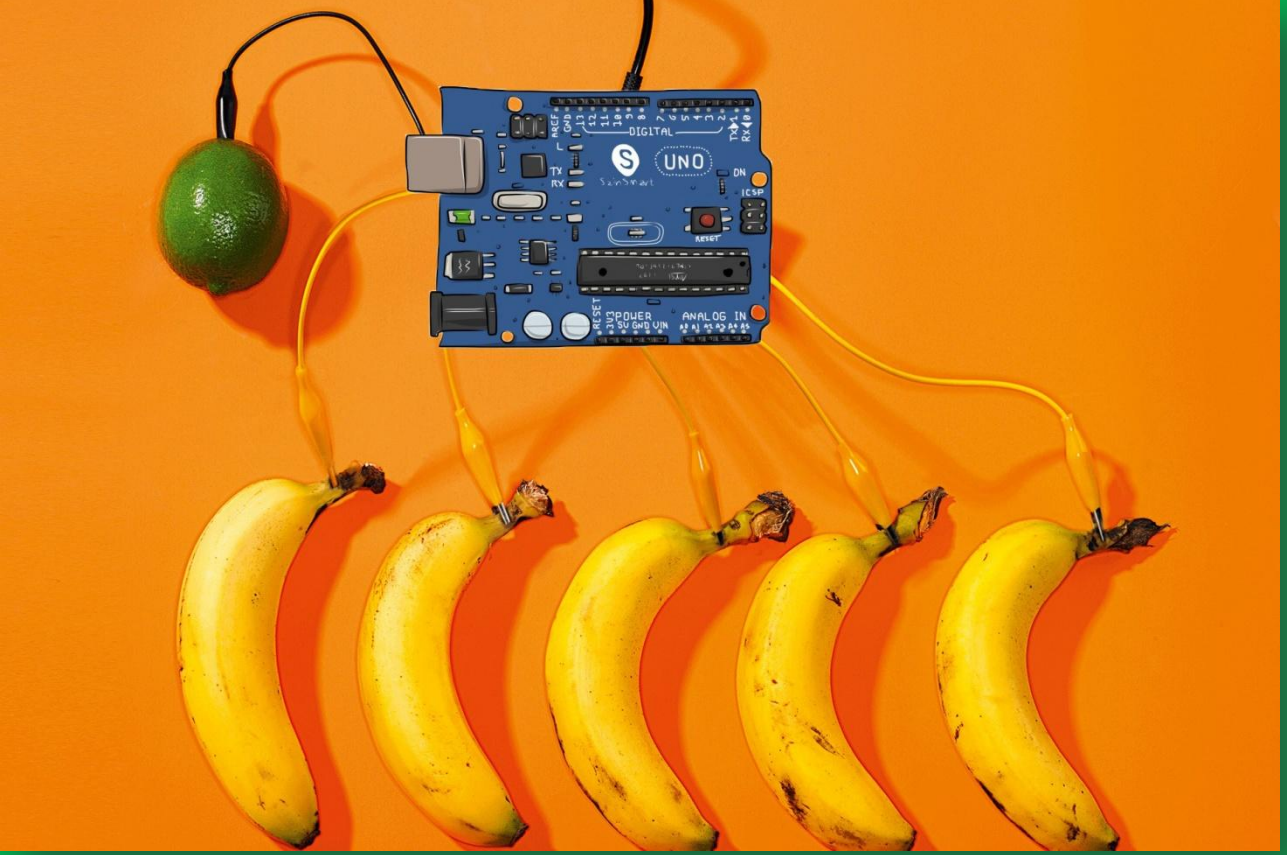
Gerekli Malzemeler

- | | |
|----------------------|-------------------------------------|
| 1 adet Arduino Uno | 3 adet 10K Ω direnç(Sensöre) |
| 1 adet Breadboard | 1 adet 220 Ω direnç(Led'e) |
| 1 adet Civalı sensör | 1 adet Led |
| 2 adet Push buton | Yeteri kadar Jumper kablo |

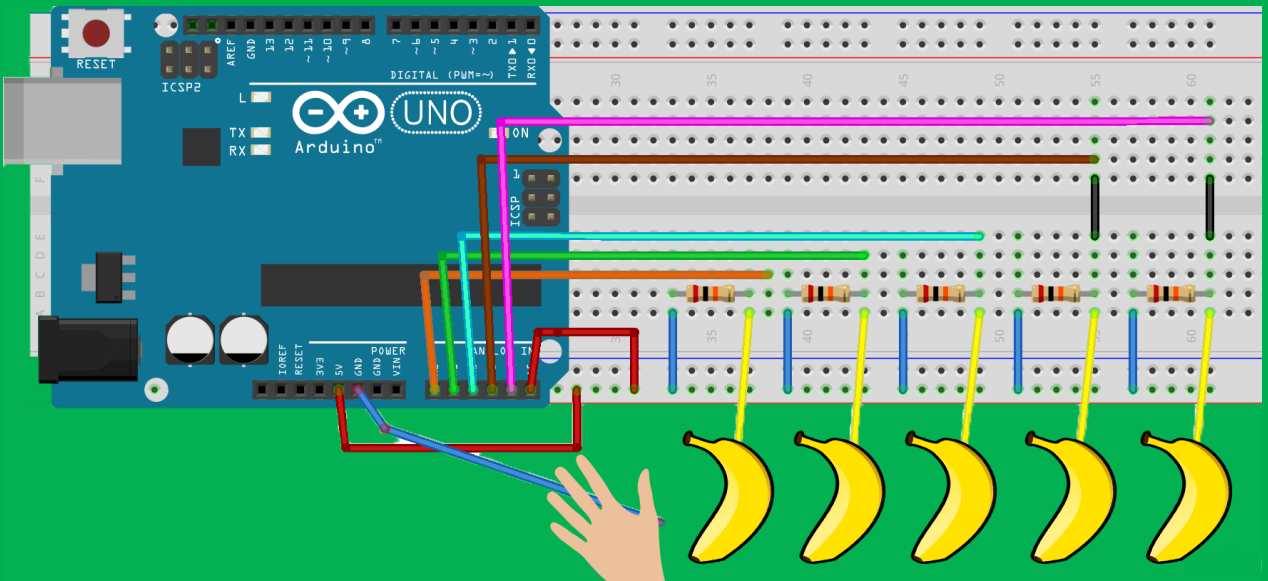


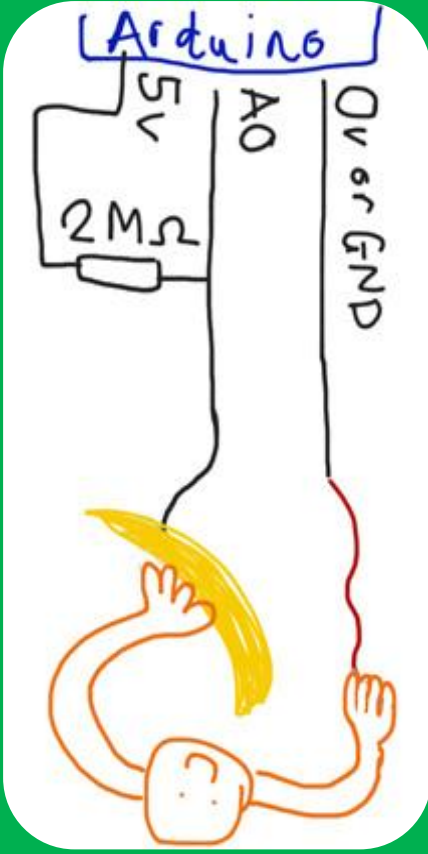
[Youtube videosu için tıklayınız.](#)

Muzdan Piyano Yapımı



Arduino'ya muz bağlayarak piyano tuşu haline getirebilirsiniz. Sadece muzları değil başka diğer meyveleri, oyun hamurlarını hatta kağıtları bile birer klavye tuşu haline getirebilirsiniz.





Bu devrede, biz de devrenin bir parçası oluruz. Bizim görevimiz bir nevi anahtar gibi çalışıp, devrenin GND pini ile bağlantı kurup kurmamasına karar vermektir.

Devre GND'ye bağlı değilken bir dirençle 5V'a bağlıdır. Bu devremizi elektriksel gürültüden korur. Karmaşık değerler okumasını engeller ve sabit 1023 değeri okunur. Biz bir elimizle GND'den gelen kabloya diğer elimizle de muza temas ettiğimizde devre tamamlanır ve analog 0 pininden okunan değer aşağıya düşer.



Yeşil bayrak tıkladığında program çalışır. İlk adımda enstrüman seçer. 1 no.lu enstrüman akustik piyanodur.

Sürekli bloğunun içinde bir eğer ise yapısı sensörden okunan değer 1023'ün altına düşüp düşmediğini kontrol eder. Değer 1023'ün

altına biz muza dokunduğumuzda düşer. Değer 1023'ün altına düşerse içerisine yazdığımız komutları yapar. Kostümü değiştirir. 60 notasını çalar ve değer 1021'in üstüne çıkmasını bekler. Değer tekrar 1023'e biz elimizi muzdan çektiğimizde gelir. Eski kostümüne geri gelir.

Bu eğer ise bloğunu diğer analog pinler için de yazarız.



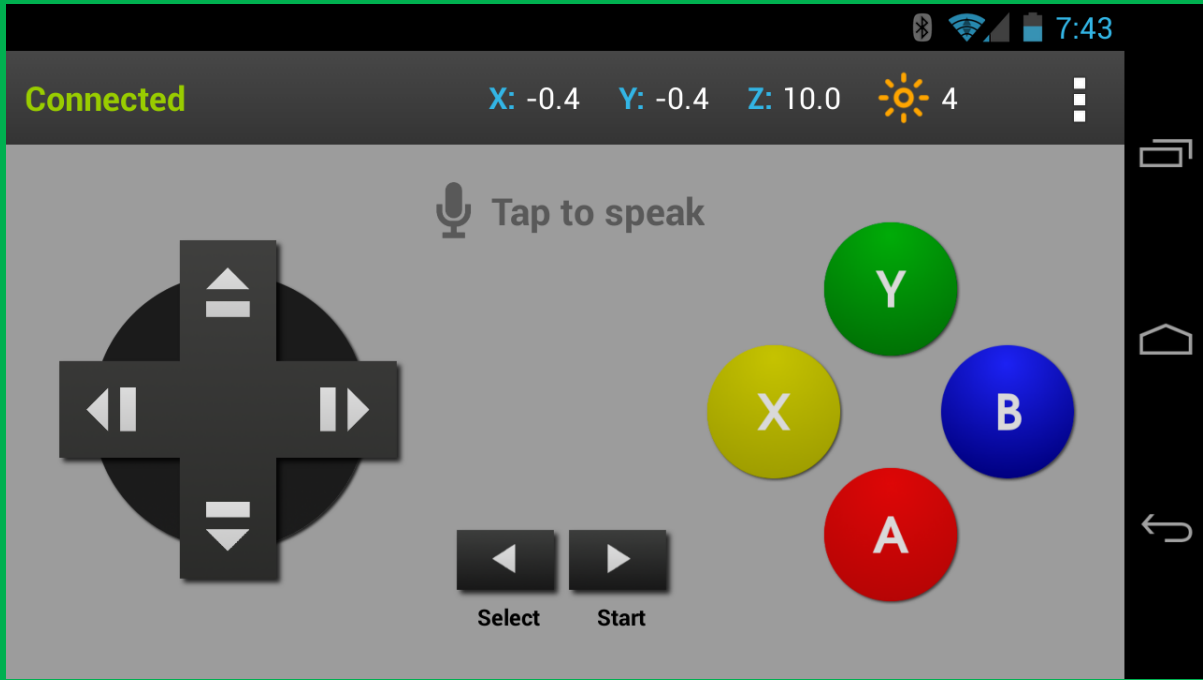
[Youtube videosu için tıklayınız.](#)

Android Telefonla Scratch Kontrolü

Hazır sensörlerden ve bu sensörleri kullanarak Scratch kontrolünden bahsetmişken, atlamak istemediğim bir konu daha var. Android telefonumuza indireceğimiz bir uygulama ile telefonun ivmeölçer ve ışık sensörlerini kullanarak Scratch karakterimizi programlayabiliyoruz.

Uygulama linki:

<https://play.google.com/store/apps/details?id=com.khanning.scratchercontrol>



Uygulamayı S4A programı ile değil, Scratch programının 1.4 versiyonu ile kullanacaksınız. Arduino kurulum dosyalarının içinde Scratch kurulum dosyası mevcut. Ya da aşağıdaki linkten indirip kurabilirsiniz.

<http://scratch.eba.gov.tr/ScratchKurulum.exe>

- Scratch programını kurun ve çalıştırın. Algılama menüsü altındaki sensör değeri bloğuna sağ tıklayın ve uzak sensör bağlantılarını sağlayı seçin.

Sensör değeri sürgü

yardım
çoğalt
sil
ScratchBoard izleyicisini göster
uzak sensör bağlantılarını sağla

- Uygulamayı kurup çalıştırın
- Menüden Connection (Bağlantı) kısmını seçin ve Scan (Tara) tuşuna basın
- 192.168.x.x olan ip numaranızı bulacaktır, ip numaranıza tıklayın ve Connect (Bağlan) tuşuna basın. Sol üst köşede yeşil **Connected** (Bağlandı) yazacaktır. Artık bağlantı kuruldu.

Android cihaz üzerinden kontrol edebileceğiniz sensörler, sensör değeri bloğuna eklenecek.

The image shows a Scratch code block for a 'when clicked' event. It contains a 'forever' loop with four 'if' blocks. Each 'if' block checks for a specific button press: 'button-right-pressed', 'button-left-pressed', 'button-up-pressed', and 'button-down-pressed'. If the condition is true, it triggers a 'change x by 5' or 'change y by 5' block. A dropdown menu for 'Sensör değeri' is open, showing a list of sensors including 'sürgü', 'ışık', 'Ses', 'direnç-A', 'direnç-B', 'direnç-C', 'direnç-D', 'eğim', 'mesafe', 'accelerometer-x', 'accelerometer-y', 'accelerometer-z', 'button-a-pressed', 'button-b-pressed', 'button-down-pressed', 'button-left-pressed', 'button-right-pressed', 'button-up-pressed', 'button-x-pressed', 'button-y-pressed', and 'light-level'. The 'accelerometer-x' sensor is currently selected.

Önemli Not: Android cihazınız ve bilgisayarınız aynı ağa yani aynı modeme bağlı olmalıdır.

The image shows a Scratch code block for a 'when clicked' event. It contains a 'forever' loop with two 'change x by' and 'change y by' blocks. The x-axis is updated by 'Sensör değeri accelerometer-x / 2' and the y-axis is updated by 'Sensör değeri accelerometer-y / 2'.

Arduino Yazılım Geliştirme Aracı

S4A programı ile Arduino'ya program yazmak her ne kadar kolay ve zevkli olsa da bazı olumsuz yönleri mevcuttur. S4A programını kapattığınızda ya da usb kablosunu çıkardığınızda program çalışmayı durdurmaktadır. Arduino'nun kendi yazılım geliştirme aracında ise yazdığımız ve yüklediğimiz programlar kalıcı olmaktadır.

Kitabın bu bölümünde bazı örnekleri aynı devre şemasını kullanarak Arduino'nun kendi yazılım geliştirme aracında yazacağız.

Arduino yazılım geliştirme aracında C++ programlama dilinin basitleştirilmiş bir versiyonu kullanılmaktadır.

Kitabın başında Scratch programlama dili için yüksek seviye (Kolay) bir programlama dili demiştik. C++ ise orta seviye bir programlama dilidir. Scratch'a göre biraz daha zordur ancak basit uygulamaları rahatlıkla geliştirebilirsiniz ve bu dilde program yazmaya alışabilirsiniz.

C++ dilinde kod yazarken Scratch'da olduğu gibi bloklardan oluşan bir dil değil, yazılı bir dil kullanacağız.

```
void setup() { // void setup() ve void loop() Arduino için temel kodlardır.
```

```
// Bu işaretten sonra yazılan yazılar yorum olarak algılanır ve işleme alınmaz.
```

```
/* void setup() kısmı program ilk çalıştığında yapılacak işlemleri tanımlayacağımız kısım. Scratch'da olduğu gibi yeşil bayrak yok. Programı yükledikten sonra program hemen çalışmaya başlar. */
```

```
}
```

```
void loop() {
```

```
// void loop() kısmına yazdığımız programlar sürekli olarak tekrarlanır.  
//Scratch'da kullandığımız sürekli komutu gibi. Loop döngü anlamına gelir.
```

```
}
```

[Kaynak Dosyaları İndirmek İçin Tıklayınız!](#)

Led Yakıp Söndürme (Blink)

```
void setup() {  
  pinMode(10, OUTPUT);  
}  
void loop() {  
  digitalWrite(10, HIGH);  
  delay(1000);  
  digitalWrite(10, LOW);  
  delay(1000);  
}
```



void setup()

{

/*void setup() kısmı program başladığında çalışır demiştik. Bu örnekte kullandığımız pinMode komutu ile ilgili pini çıkış pini (OUTPUT) veya giriş pini (INPUT) olarak tanımlarız. Devreden akım verilecekse, ilgili pini çıkış pini, akım değeri ölçülecekse, giriş pini olarak ayarlarız.

Scratch'da bu komutu kullanmamıştık. Çünkü S4A kullanmadan önce Arduino'ya bir program yüklemiştik. İşte o programda bunlar bizim için otomatik olarak tanımlanmıştı. */

pinMode(10, OUTPUT);

// Kullanım şekli: pinMode(ilgili pin numarası, giriş veya çıkış);

/* Kodları yazarken süslü parantezlere ve noktalama işaretlerine çok dikkat etmelisiniz. */

Programlama dillerinde bizim kullandığımız konuşma dillerinde olduğu gibi yazım kuralları vardır ve bu kurallar çok katıdır. Eğer bu yazım kurallarına uymazsanız program hata verir.

Örnek: Yap çıkış 10 pini dediğim zaman siz bunun 10 nolu pini çıkış olarak ayarla anlamına geldiğini anlayabilirsiniz ama bilgisayarlar bunu anlayamaz. O yüzden yazım kuralları çok önemlidir.

```
void loop() //void loop() içine yazdığımız kodlar sürekli tekrarlanır.
```

```
{
```

```
/* Kodları yazarken satır satır yazarız. Kodlar en üst satırdan aşağıya doğru çalışmaya başlar.*/
```

```
digitalWrite(10, HIGH);
```

```
/* digitalWrite komutu ilgili pine güç vermek veya gücü kesmek için kullanılır. Parantez içine işlem yapılacak pin numarasını yazarız, virgülden sonra ise güç verilecekse "HIGH" , güç kesilecekse "LOW" yazarız. Komutlardan sonra ilgili komutun bittiğini belirtmek için noktalı virgül koyarız. */
```

```
// Burada "HIGH" yazdığımız için 10 nolu pine güç verir.
```

```
delay(1000);
```

```
/* delay komutu araya bekleme koymak için kullanılır. Parantez içine yazdığımız değer kadar bekler ve birimi milisaniyedir. 1000 milisaniye 1 saniyedir. */
```

```
digitalWrite(10, LOW);
```

```
// Burada duruma "LOW" yazdığımız için 10 nolu pinden gücü keser.
```

```
delay(1000);
```

```
}
```

[Video](#)



Değişken Kullanımı

Değişkenleri bir sayıyı veya değeri bilgisayar hafızasında tutmak için kullanıyorduk. Scratch'da değişkenleri kullanmak çok kolaydı ancak Arduino yazılım geliştirme aracında dikkat etmemiz gereken bazı özellikler var. Mesela oluşturacağımız değişkenin türü ne olacak. Nasıl bir değişken kullanacaksak, değişken oluşturmadan önce türünü belirliyoruz. Mesela tamsayı, ondalıklı sayı, metin. Scratch'dan farklı olarak bir de Türkçe'ye has harfleri (ş, ç, ğ, ı, ö, ü) değişken isimlerinde kullanamıyoruz. Örneğin: "sayaç" isminde bir değişken oluşturamayız. Değişkenin isminde Türkçe'ye has harfler olmamalı. Değişkenin ismini "sayac" olarak tanımlayabiliriz.

Örnek: Led isminde bir değişken oluşturalım ve o değişkene 13 değerini atayalım. 13 değeri tamsayı olduğu için "int" komutunu kullanacağız, "int" komutu integer kelimesinden gelir. Integer İngilizce'de tamsayı demektir.

`int Led = 13;` //Dikkat etmemiz gereken nokta "=" işareti burada eşitliği temsil etmez. Atama işlemi için kullanılır. Yani 13 değerini Led değişkenine ata demektir. Eşittir işareti ise "==" dir.

```
int Led = 13; // Led isminde bir değişken tanımladık ve 13 değerini atadık.
```

```
void setup() {
```

```
  pinMode(Led, OUTPUT); /* 13 yazmak yerine Led yazdık. Led değişkeni 13 olduğu için 13 nolu pini çıkış pini olarak ayarlayacaktır. */
```

```
}
```

```
void loop() {
```

```
  digitalWrite(Led, HIGH); // Burada da değişkeni yazdık. 13 nolu pine güç verir.
```

```
  delay(1000);
```

```
  digitalWrite(Led, LOW); // Led değişkeni 13, 13 nolu pinden gücü keser.
```

```
  delay(1000);
```

```
}
```

Değişken türü

Atama Operatörü

```
int Led = 13;
```

Değişkenin adı

Değişkenin değeri

Değişken tamsayı değerler alacaksa int komutunu, ondalıklı değerler alacaksa float komutunu kullanıyoruz.

Örnek:

```
float pi = 3.14;
```

Analog Pinden Değer Okuma

Hatırlıyorsanız S4A programında Analog pinlerden değerler okuyorduk ve bu değerleri kullanarak projelerimizi gerçekleştiriyorduk. Burada da Analog 0 pinine bir potansiyometre bağlayacağız ve değerini okuyacağız.

```
analogRead(A0); // bu komut ile analog 0 pininin değerini okuruz
```

value of sensor Analog0

Bu okuduğumuz değeri bir değişkene aktaralım aynı S4A programında yaptığımız gibi.

```
int sensorOku = analogRead(A0);
```

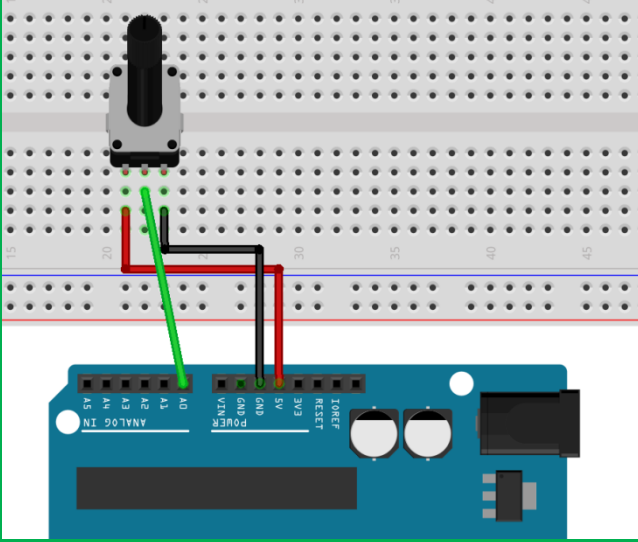
```
/* Analog 0 pininden okunan değeri "sensorOku" isimli bir değişkene aktardık. */
```

sensorOku 'i value of sensor Analog0 yapın

Bu şekilde oluşturduğumuz "sensorOku" isimli değişkene Analog 0 pininden okunan değeri atamış olduk. Daha sonra sensorOku değişkenini kullanarak projelerimizi gerçekleştireceğiz. Bu okunan değerler S4A programında sürekli ekranda gözüküyordu ancak Arduino'nun kendi yazılım geliştirme aracında bu değerleri görebilmek için bir kod yazmamız ve araçlar menüsündeki "Seri Port Ekranı" kısmına tıklamamız gerekir.

```
Serial.println(sensorOku);
```

```
/* Bu komut ile Analog 0 pininden okunan değerin aktarılmış olduğu  
sensorOku değişkenini seri port ekranına yazdırırız. Bu değerleri  
görmek isterseniz, Araçlar>Seri Port Ekranı menüsüne tıklayınız. */
```



Potansiyometrenin bir ucunu 5V Diğer ucunu GND(0V) ye bağlıyoruz. Orta bacağını ise Analog 0 pinine bağladıktan sonra yan taraftaki kodu Arduino karta yüklüyoruz. Araçlar menüsünden Seri Port Ekranını seçiyoruz ve değerleri okuyoruz.

```
void setup() {  
  Serial.begin(9600);  
  
  /* Bu kod bilgisayar ile Arduino'nun  
  haberleşmesi için bu kodu mutlaka yazın*/  
}  
  
void loop()  
{  
  int sensorOku = analogRead(A0);  
  Serial.println(sensorOku);  
  
  delay(1); /* Araya 1 milisaniye  
  bekleme yapıyoruz. Bu şekilde  
  düzenli bir şekilde okumayı yapması için koyduk. */  
}
```

Kontrol Yapıları/ Eğer ise Değilse

Biz nasıl günlük hayatımızda bazı kararlar almak zorunda kalıyorsak, bilgisayarlarında bazı işleri yapabilmeleri için kararlar almaları gerekir. Örneğin biz havaya bakıyoruz ve hava soğuksa üzerimize mont alıyoruz. Burada koşul havanın soğuk olması. Eğer koşul sağlanmışsa yani hava soğuksa bir eylem yapıyoruz yani dışarıya çıkarken montumuzu alıyoruz.

Program yazarken belli şartlar gerekleŖtiėinde bazı iŖlerin yapılmasını, bazı iŖlerinde yapılmamasını isteyebiliriz. Bu gibi durumlarda program yazarken kontrol yapılarını kullanırız. Bu yapılar if / else (eėer ise / deėilse) yapılarıdır.

Kullanım Ŗekli:

```
if (koŖul)
```

```
{
```

```
Komutlar; // KoŖul gerekleŖtiėinde yapılacaklar.
```

```
}
```

Sadece if yapısını kullanabileceėimiz gibi if / else yapısını birlikte de kullanabiliriz. if yapısında sadece koŖul gerekleŖtiėinde yapılacak iŖler vardır, if/else yapısında ise hem koŖul gerekleŖtiėinde yapılacak iŖler hem de koŖul gerekleŖmediėinde yapılacak iŖler vardır.

```
if (hava soėuk mu?) //koŖulumuz
```

```
{
```

```
Mont giy; //KoŖul gerekleŖtiėinde yapılacaklar
```

```
}
```

```
Else
```

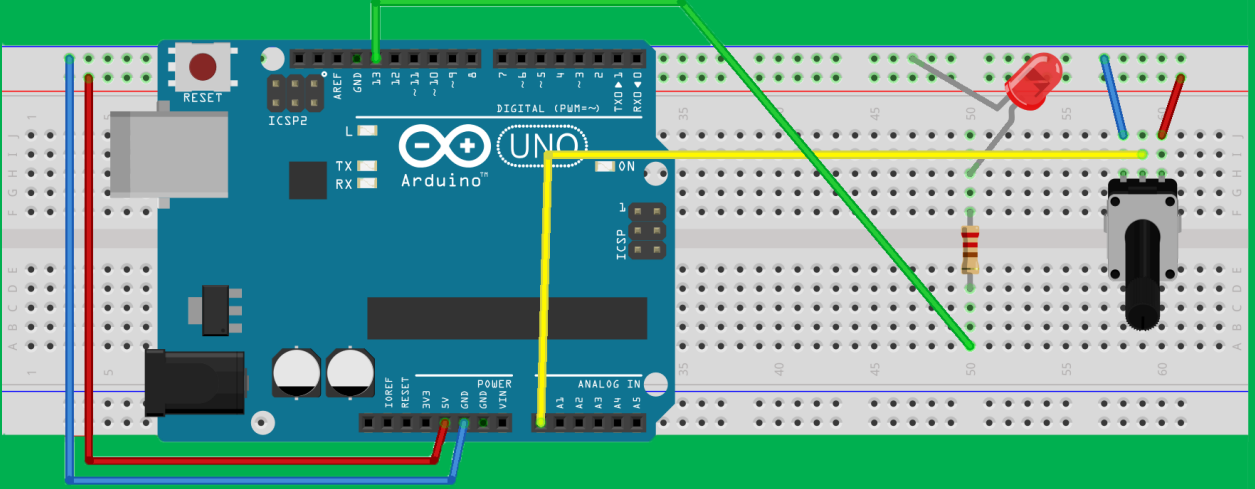
```
{
```

```
Gömlek giy; // KoŖul gerekleŖmediėinde yapılacaklar
```

```
}
```

Potansiyometreden Okunan Değere Göre Led'i Yakma

Bu uygulamada, potansiyometreden okunan değerlere göre Led yanacak veya sönecek. Bunun için potansiyometreden gelen değeri okuyarak bir değişkene aktaracağız. Daha sonra if/else yapısı kurarak koşulumuz gerçekleştiğinde Led'in yanmasını, gerçekleşmediğinde ise sönmesini sağlayacağız.



```
void setup() {  
  Serial.begin(9600);  
}  
void loop() { // Loop başlangıç.  
  int sensorOku = analogRead(A0);  
  Serial.println(sensorOku);  
  delay(1);  
  if (sensorOku > 500) // Koşul.  
  { // Koşul gerçekleştiğinde yapılacak komutlar için başlangıç.  
    digitalWrite(13, HIGH);  
  } // Koşul gerçekleştiğinde yapılacak komutlar için bitiş.  
  else  
  { // Koşul gerçekleşmediğinde yapılacak komutlar için başlangıç.  
    digitalWrite(13, LOW);  
  } // Koşul gerçekleşmediğinde yapılacak komutlar için bitiş.  
} // Loop bitiş.
```

```
if (sensorOku > 500)
{
  digitalWrite(13, HIGH);
}
else
{
  digitalWrite(13, LOW);
}
```



Potansiyometreden okunan değer sürekli sensorOku değişkenine aktarılıyor. Biz de koşul olarak bu değerın 500'den büyük olmasını belirledik. Yazdığımız kod, değişkenin değeri 500'den büyük olursa ilk süslü parantez içindeki komutları uygulayacak ve 13 nolu pine güç verecek aksi takdirde else kısmında bulunan süslü parantez içindeki komutları uygulayacak ve 13 nolu pinden gücü kesecek.

Döngüler

Tekrar eden olaylara döngü denir. Bilgisayarların başarılı olduğu konulardan birisi de döngülerdir. Biz aynı işleri tekrar tekrar yaparken hem yoruluruz hem de sıkılırız. Bilgisayar ise aynı işleri hem çok hızlı hem de hiç sıkılmadan yaparlar.

Program yazarken bazı iş ve işlemlerin tekrarlanması gerekebilir. Tekrar etmesi gereken komutlar için döngüleri kullanırız. Kitapta kullanacağımız döngü, for döngüsü olacak.

```
For (i=başlangıç değeri ; i< bitiş değeri; Artış miktarı)
```

```
{
```

```
Komutlar;
```

```
}
```

For döngüsünü bir tür sayaca benzetebiliriz. Bir değişken belirleyip (burada "i" belirlenmiş) başlangıç değeri atıyoruz sonra ";" işareti koyuyoruz ki bu kısmın bittiğini belirtiyoruz. Diğer kısımda döngü i değeri kaç oluncaya kadar devam edecek onu belirliyoruz ve ";" ile bitişini belirtiyoruz. Son kısımda da artış miktarını belirtiyoruz. Eğer her adımda değişkenin birer birer artmasını istiyorsak "i++" yazıyoruz.

```
void setup() {
```

```
  pinMode(9, OUTPUT); // 9 nolu pini çıkış pini yapıyoruz.
```

```
}
```

```
void loop()
```

```
{
```

```
  int deger = 0; // Deger değişkeni oluşturup 0 değerini atıyoruz.
```

```
    for (int i=0 ; i<255 ; i++) // Döngüyü tanımlıyoruz.
```

```
    {
```

```
      analogWrite(9, deger); //9 nolu pinden verilecek akımı belirliyoruz.
```

```
      /* DigitalWrite komutunda sadece HIGH ve LOW vardır. AnalogWrite komutu ile ise 0-255 arasında bir değerde akım yollayabiliriz. AnalogWrite komutunu sadece 11,10,9,6,5, ve 3 nolu digital pinlere uygulayabiliriz. */
```

```
      deger++; // Deger degiskenini bir arttırıyoruz.
```

```
      delay(20); // 20 milisaniye bekleme koyuyoruz.
```

```
    }
```

```
}
```



```
void loop()
```

```
{
```

```
  int deger = 0;
```

```
  for (int i=0 ; i<=255 ; i++)
```

```
  /* Döngü her adımda bir artarak, 0 dan 255'e kadar tekrarlayacak yani 255 defa  
  tekrarlayacak */
```

```
  {
```

```
  /* 9 nolu pin 0-255 arası değerler alır. Biz değer olarak "deger" değişkenini  
  kullandık. Değer değişkeni her basamakta bir artacak döngü 255 defa tekrarlayacağı  
  için 0'dan 255'e kadar değerler alacak. Pinden önce 0 değerinde, sonra artarak 255  
  değerinde akım gönderilecek */
```

```
    analogWrite(9, deger);
```

```
    deger++;
```

```
    delay(20);
```

```
  }
```

```
}
```



Kitabımızda programlama ile ilgili temel kavramları Arduino ile zevkli kılmaya çalıştık. Artık bilgisayar programlama ve elektronik cihazların nasıl çalıştığı ile ilgili bazı bilgiler edindiniz. Bu bilgileri ilerletmek sizin elinizde. Önünüzde sonsuz bilgi kaynağı olarak internet mevcut. Bu kaynağı kullanarak bilgilerinizi ilerletebilirsiniz.

Her şey gönlünüzce olsun, hayatınızda başarılar dilerim.