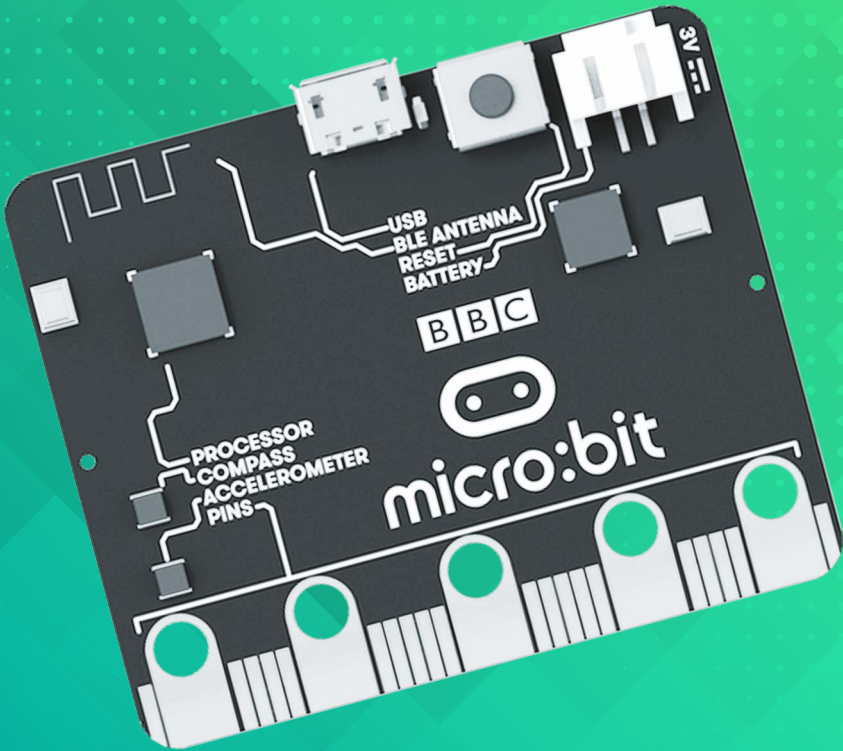


BBC



micro:bit

Uygulama Kitabı

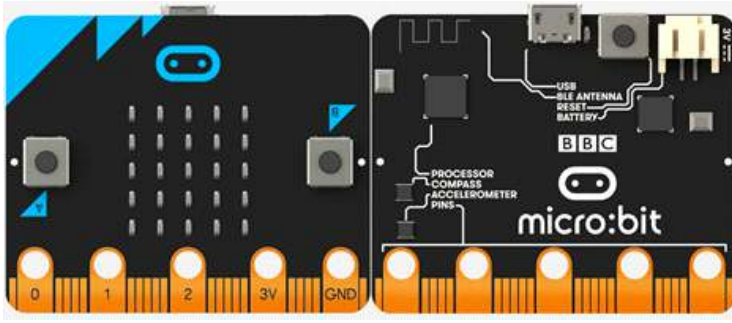


robotistan



İçindekiler

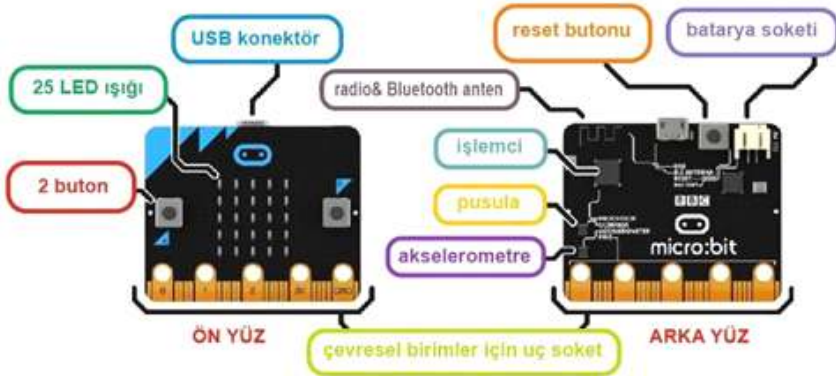
Micro:Bit Nedir.....	3
Micro:Bit Özellikleri.....	3
A.Makecode Editörü.....	4
B.Python Editörü.....	5
C.mBlock 5 Programı İle Kodlamak.....	5
Micro:Bit Pinleri.....	6
Proje 1: Adım Sayar.....	10
Proje 2: Termometre.....	12
Proje 3: Zar.....	14
Proje 4: Led Kullanımı.....	15
Proje 5: Buton Kullanımı.....	16
Proje 6: Pong Oyunu.....	17
Proje 7: Biletleme Sistemi.....	19
Proje 8: Trafik Lambası.....	21
Proje 9: Buzzer ile müzik seslendirme.....	24
Proje 10: Mesafe Ölçümü.....	25
Proje 11: Servo Motor kontrolü.....	27
Proje 12: Potansiyometre ile led parlaklığı.....	29
Proje 13: Pusula ile Motor kontrolü.....	30
Proje 14: RGB led kullanımı.....	31
Proje 15: Buton ile LED kullanımı.....	32



micro:bit, üzerinde kod çalıştırmanıza ve tüm donanıma erişmenize izin veren açık kaynaklı bir geliştirme kartıdır.

BBC micro:bit, robotlardan müzik aletlerine kadar harika tasarımlar yapmanıza olanak tanıyan sonsuz bir proje hazinesine sahiptir. micro:bit, mesajları 25 kırmızı LED ışığı ile kolayca gösterilebileceği gibi çok sayıda özelliğe sahiptir. Oyunları kontrol etmek veya bir çalma listesindeki şarkıları duraklatmak ve atlamak için kullanılabilen iki adet programlanabilir düğmesi mevcuttur. BBC micro:bit hareketi algılayabilir ve hangi yöne doğru ilerlediğinizi söyleyebilir. Diğer cihazlarla ve internet ile etkileşime girmek için düşük enerjili bir Bluetooth bağlantısı kullanabilir.

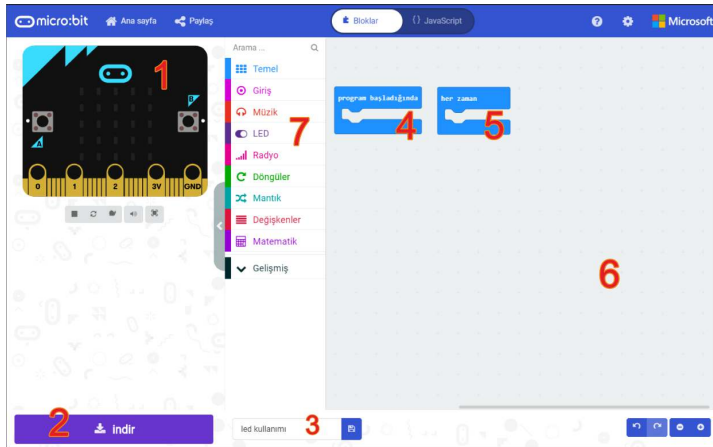
Micro:Bit Özellikleri



- 5x5 diziliminde matris LED
- 2 adet programlanabilir buton
- Sıcaklık sensörü
- 3-eksenli ivme ölçer
- 3-eksen pusula
- Bluetooth Low Energy (BLE) bağlantısı

(<https://makecode.microbit.org/>)

Blok sistemi kullanılarak kolaylıkla kodlama yapabilmenize olanak tanıyan bir editördür.



1.micro:bit Simülatörü

Simülâtör, programınızda bir değışiklik yaptıktan sonra otomatik olarak çalışır. Simülâtörün altındaki düğmeler, program yürütmeyi durdurmanıza / başlatmanıza ve simülâtörün diğér yönlerini kontrol etmenize olanak tanır.

2.İndirme Butonu

Programı bilgisayarınıza veya micro:bit'inize indirmek için kullanılır.

3.İsim Yazma Kutusu

Projenizi yeniden adlandırarak dilediğiniz ismi verebilirsiniz.

4.Program Başladığında Bloğu

Projeyi micro:bit'e yükleyip çalıştırdığınızda başlangıçta gerçekleşmesi gereken komutların eklendiği bloktur.

5. Her Zaman Bloğu

Projenin sürekli olarak çalıştırılmak istenen kısmıdır. Çoğunlukla proje bu bloğun içerisine kurulur.

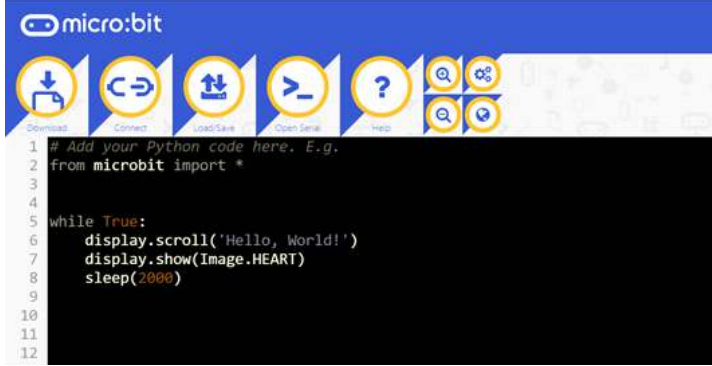
6. Programlama Alanı

Projelerinizi oluşturun / düzenleyin.

7.Araç Kutusu

Burada farklı kategorilerdeki programlama bloklarını bulabilirsiniz.

(<https://python.microbit.org/v/2.0>)



Sayfadaki download seçeneği ile karta yükleyeceğimiz hex uzantılı dosyayı bilgisayara indirebilir karta yükleyebiliriz.

Save seçeneği ile oluşturduğumuz .py uzantılı Python dosyalarını bilgisayarımızda muhafaza edebiliriz.

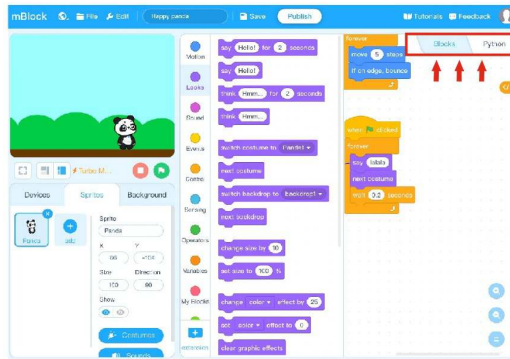
Load seçeneği ile daha önceden bilgisayarımıza indirdiğimiz .py uzantılı dosyaları micro:bit Python editörüne yükleyebilir üzerinde değişiklik yapabiliriz.

C.mBlock 5 Programı İle Kodlamak

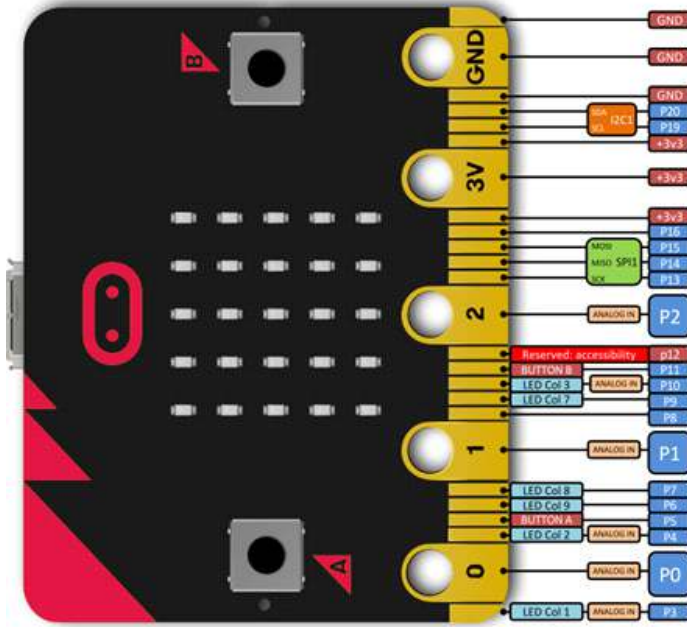
Kartımızı online olarak micro:bit kartının kendi sitesinden kodlayabileceğimiz gibi farklı araçlarda kullanabiliriz.

Örneğin, mBlock programının 5. Sürümünü yükleyerek internet olmadan offline olarak kartımızı programlayabiliriz.

Sol tarafta bulunun Ekle seçeneği ile kartımızı yükleyebiliriz.



Sağ üst tarafta bulunan Bloklar/Python seçeneği ile ister blok tabanlı, ister python kodları ile kartımızı kodlayabiliriz. Kart seçme işlemi için ekle seçeneği seçilerek, açılan menüden micro:bit kartı seçilir.



micro:bit kartı üzerinde 19 adet pin bulunur. Bu pinlerden bazıları led matris display için, bazıları butonlar için, bazıları pwm özelliği için, bazıları ise farklı veri yolu özellikleri için ayrılmıştır. Micropython ile çalışırken bu pinlerden ihtiyacımız olanlara pinlerin özelliklerine göre erişim sağlayacağız.

Display: Pin3, pin4, pin6, pin7, pin9 ve pin10 led matris display için kullanılmaktadır. Bu pinlerden herhangi birini kullanmanız gerektiğinde; **display.off()** komutu kullanarak display özelliğini kapatabiliriz. Bu sayede belirtilen pinleri projemizde genel amaçlı olarak kullanabiliriz. Display modu açıkken, 7 numaralı pine dijital 1 değerini veya pwm değeri vermek istediğimizde;

ValueError: Pin 7 in display mode şeklinde bir hata alırız. Pinlerin display modunda mı değil mi kontrolünü sağlamak için **display.is_on** komutunu kullanırız. Bu komut display özelliği açık ise **True**, kapalı ise **False** değeri döndürecektir.

Butonlar: pin5 ve pin11 butonların kullanımı için ayrılmıştır. Bu pinler 10K değerinde dirençler ile pull up özelliğine sahiptir. Buton pinlerine genel amaçlı kullanmak üzere ulaşmaya çalıştığımızda

ValueError: Pin 5 in button mode şeklinde bir hata almaktayız. Micropython ile buton özelliğini kapatma imkanımız bulunmamaktadır.

I2C: pin19 ve pin20 I2C özelliğine sahip pinlerdir. Pin19 SCL, pin20 ise SDA pinidir. I2C veri yolu ile lcd ekran, çeşitli sensörler aynı veri yolunu kullanarak iki hat üzerinden haberleşebilmektedirler. I2C veri yoluna bağlanacak cihazın olan harici güç kaynağı ile çalışması durumunda GND pini micro:bit ile ortak bağlanmalıdır.

SPI: Spi özelliği de bu veri yolunu kullanan cihazlar ile iletişim sağlamak amacıyla kullanılır. SPI veri yolunu kullanan pinler; pin13-SCLK, pin14-MISO pin15-MOSI.

PWM: pin0, pin1, pin2, pin3, pin4, pin6, pin7, pin8, pin9, pin10, pin12, pin13 pin14, pin15, pin16 nolu pinlerdir. Pulse width modulation-darbe genişlikli modülasyon ile led parlaklığı, motor hız kontrolü, hızlı anahtarlama gibi özellikleri için kullanılmaktadır.

UART: UART Özelliğine sahip cihazlar ile haberleşmek için kullanılır. Tx ve Rx pinleri pin0 ve pin1 olarak belirlenmiştir.

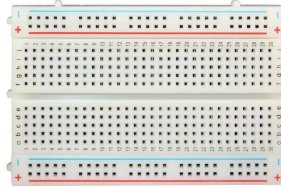
Analog Veri Okuma: pin0, pin1, pin2, pin3, pin4 ve pin10 analog veri okuma için kullanılabilir.

Set İçeriği



Micro:Bit Genişletme Kart

BBC micro:bit kartınızın Edge konektörü üzerinde yer alan bağlantıları kolay bir şekilde breadboard üzerine taşımanızı sağlayacak olan genişleme kartıdır.



Breadboard

Breadboard; 2 güç yoluna, 30 sütun ve 10 satıra sahip toplamda 400 pin deliklerine sahiptir. Bütün pinler arasındaki mesafe 1 inç'tir, Satırlar beşer olarak iki takım şeklinde 0.3 inc aralıkla ayrılmıştır. Breadboard'unuzu jumper kablo ile kolaylıkla kullanabilirsiniz.



Direnç

Elektrik devrelerinde direnç, bir iletken üzerinden geçen elektrik akımının karşılaştığı zorlanmadır. Mekanik sistemlerdeki sürtünmeye benzer özellikler gösterir. Direncin birimi Ohm (Ω)'dur. Denklemlerde R harfi ile gösterilir.



NTC

Bulunduğu ortamın veya temas ettiği yüzeyin sıcaklığı arttıkça elektriksel direnci azalan devre elemanıdır. Bulunduğu ortamın veya temas ettiği yüzeyin sıcaklığı arttıkça elektriksel direnci azalan devre elemanıdır.



Transistör

Geirge veya transistör giriřine uygulanan sinyali yükselterek gerilim ve akım kazancı sađlayan, gerektiđinde anahtarlama elemanı olarak kullanılan yarı iletken bir elektronik devre elemanıdır.



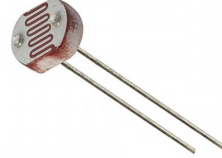
Kondansatör

Elektronların kutuplanıp elektriksel yükü elektrik alanının ierisinde depolayabilme özelliklerinden faydalanılarak bir yalıtkan malzemenin iki metal tabaka arasına yerleştirilmesiyle oluşturulan temel elektrik ve elektronik devre elemanıdır.



Pil Yatađı

Piller ile micro:bit'inizin bađlantısını yaparak, micro:bit'inize güç verir.



LDR

Foto direnler, üzerlerine düşen ışık şiddetiyle ters orantılı olarak direnleri deđişen elemanlardır. Foto diren, üzerine düşen ışık arttıka diren deđeri lineer olmayan bir şekilde azalır. LDR'nin aydınlıkta direnci minimum, karanlıkta maksimumdur.



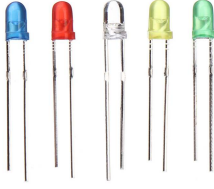
Buzzer

Buzzerlar, kullanım alanlarına da bađlı olarak alarm, zamanlayıcı, onaylama cevap ikazı gibi işlevlerde kullanılabilmektedirler.



Buton

Sistemin kontrolünü basma gibi tepkilerle kontrol edilmesini sađlayan elektriksel cihazdır.



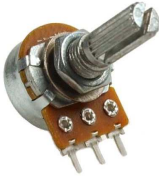
Ledler

LED, yarı-iletken, diyot temelli, ışık yayan bir elektronik devre elemanıdır.



RGB Led

RGB – Red (kırmızı) , Green (yeşil) , Blue (mavi) renklerin baş harfleri birleştirilerek oluşmuş bir terimdir. Genel çalışma prensibi; bu üç rengi kullanarak, farklı kombinasyonlarda, çok fazla renk verebilir.



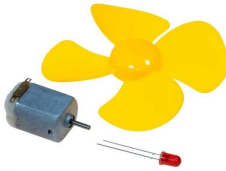
Potansiyometre

Potansiyometre, dışarıdan fiziksel müdahaleler ile değeri değiştirilebilen dirençlerdir.



Jumper Kablo

Breadboard ile micro:bit arasında bağlantı kurabilmek için kullanılan kablodur.



DC Motor ve Pervane (3-6 V)

DC motor, düz akım elektrik enerjisini mekanik enerjiye dönüştüren makinedir. Motorun içinde yer alan sargılara elektrik akımı uygulandığında, yine motorun içerisinde bulunan sabit mıknatıslara zıt yönde oluşan manyetik kuvvetin etkisi ile hareket etme prensibine dayanır.



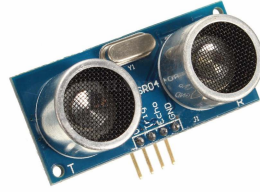
Klemens

İki ve daha çok parçaya ayrılmış elektrik iletim kablolarının, birbirlerine bağlanmasını sağlayan parçaya klemens denir. Klemenslerin yalıtkan kısımları kullanılacak yere göre plastikten, porselenden, metalden veya bakalitten yapılmaktadır.



Servo Motor

0-180 derece arasında kullanılabilen bu motorlar projelerinizde derece ile kontrol imkanı sunuyor.



Ultrasonik Mesafe Sensörü

2cm'den 400cm'ye kadar 3mm hassasiyetle ölçüm yapabilen bu ultrasonik sensör çeşididir.

Proje 1: Adım Sayar

- Micro:bit
- 2 adet AA pil
- 2'li Pil yataĐı

Projemizin kısaca nasıl çalışacaĐından bahsedelim. Attığımız her adımda micro:bit ayaĐımızı kaldırdığımız ve yere koyduğumuz andaki sarsıntıyı algılayacak ve üzerindeki ledler ile bize adım sayısını gösterecek.

Hemen kodlamaya başlamak için web sitesine gidiyoruz. (<https://makecode.microbit.org/>)



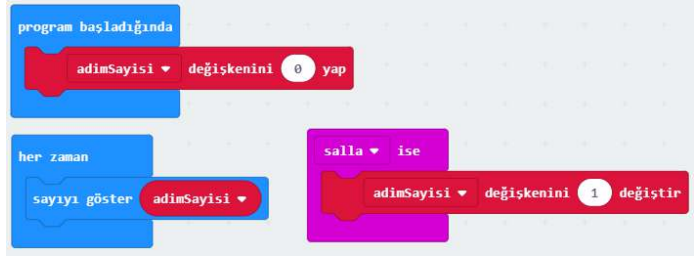
DeĐişkenimize adimSayisi ismini verip devamında başlangıç deĐerini **0** yapıyoruz. Her sallandığında ise deĐişkenimiz bir artacak.

Proje 1: Adım Sayar

“Salla” kod bloğu micro:bit sallandığında tetiklenen kod bloğudur.

“AdımSayisi 1 değiştir” kod bloğu ise adımSayisi değişkenini bir artıracak. Eğer değişkeni iki ya da farklı değerde artırmak istiyorsanız “1” yerine artırmak istediğiniz değeri girebilirsiniz.

Devamında adımSayisi'ni ledler üzerinde göstermemiz gerek. Bunu da “sayıyı göster” kod bloğuyla yapacağız. Her adım attığımızda adım sayacı artacağı ve micro:bit de bize bunu sürekli göstereceği için adımSayisi değişkenimizi sürekli tekrarlar kod bloğu içine yerleştiriyoruz.

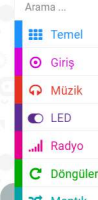
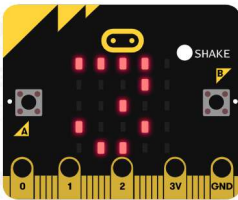


Adım Sayarı sıfırlamak istediğimizde ise herhangi bir butona tıklandığında AdımSayisi=0; diyerek sıfırlama işlemini yapıyoruz.

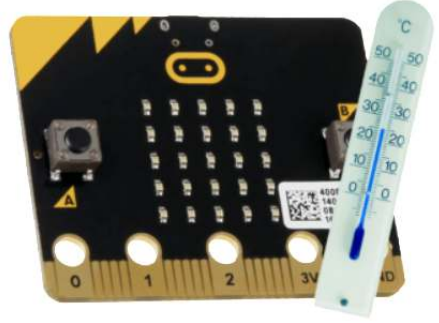
Projeye isim verip daha sonra indirip micro:bit'e yüklüyoruz. Ardından lastik veya bant ile micro:bit'i bacağımıza takıyoruz. Yürüdüğünüzde micro:bit size kaç adım attığınızı gösterecek.



İpucu: Adım sayınızı gösterirken micro:bit'in bazen geri kaldığını farkettiler mi? Micro:bit atılan bir adım için adım sayısını gösterdiği esnada yeni bir adım atmış olabilirsiniz. Bu durumda “Animasyonu durdur” kod bloğunu ekleyerek gecikmenin önüne geçmiş oluyoruz ! Kodlarımız artık daha sağlıklı çalışıyor!



- micro:bit
- Soğuk Nesne (Soğuk su şişesi veya soğuk zemin)
- Sıcak Nesne (Kalorifer veya micro:bit'i avuç içinde bir süre tutma)



micro:bit' i her salladığımızda bize ortam sıcaklığını söyleyecek. Farklı değerler alabilmek için micro:bit' i hem soğuk bir nesne hem de sıcak nesneye yaklaştırarak sıcaklığın nasıl değiştiğini gözlemleyeceğiz.

micro:bit üzerinde sıcaklık sensörünü bulmaya çalışırsanız göremeyeceksiniz, çünkü üzerinde harici olarak bir sıcaklık sensörü mevcut değil. Peki sıcaklığı nasıl ölçeceğiz dersiniz, aslında üstündeki işlemcinin sıcaklığını ölçeceğiz. Bu da bize yaklaşık olarak ortam sıcaklık değerini verebilmektedir.

Hemen “ <https://makecode.microbit.org/> ” adresine giderek, yeni bir proje oluşturup projeye “ **Termometre** ” adını veriyoruz.

Sıcaklık değerini almak için Input sekmesi altındaki “**sıcaklık (°C)**” kod bloğunu kullanacağız. Değişkenler sekmesi altında Yeni Değişken Yarat'ı tıklayarak “sıcaklık” isimli yeni bir değişken oluşturalım.

New variable name:

sıcaklık

Ok ✓

Cancel ✕

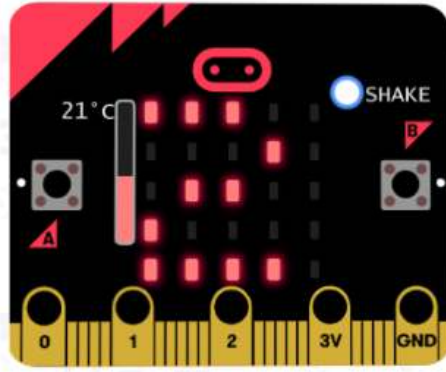
Proje 2: Termometre

Ardından oluşturduğumuz bu değişkene temperature'dan gelen değeri atayalım. micro:bit'i her salladığımızda bize değeri söylemesi için “salla ise” kod bloğunu kullanıyoruz. Değeri micro:bit üzerindeki ledler üzerinde gösterecek.



Ardından kodu indirip micro:bit'e yüklüyoruz. Her salladığımızda micro:bit bize sıcaklık değerini gösterecek. micro:bit'i avcunuzun içine alıp biraz beklettiğinizde sıcaklık değerinin arttığını göreceksiniz!

Projeyi micro:bit simülasyonunda çalıştıralım:



Projenin Javascript kodlarına bakmak için:

```
1 let sıcaklikDegeri = 0
2 girdi . onGesture (Gesture.Shake, function () {
3   sıcaklikDegeri = giriş . sıcaklık ()
4   temel . showNumber (sıcaklikDegeri)
5 })
6
```

Proje 3: Zar

- BBC Micro:Bit Kartı

Elinizde bir zar olduğunu düşünün. Sallayıp bıraktığımızda zar üzerinde 1 ile 6 arasında herhangi bir rakam gelecek. Aynı şeyi bu projemizde micro:bit ile yapacağız. micro:bit'i her salladığımızda üzerindeki ledler 1 ile 6 arasında rastgele gelen rakamı gösterecek.

Hemen “<https://makecode.microbit.org/>” adresine giderek, yeni bir proje oluşturup projeye “Zar” adını veriyoruz.

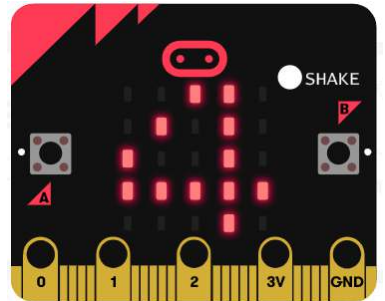
Micro:bit'i her salladığımızda bize rastgele bir rakam üretmesini isteyeceğiz. Dolayısıyla Giriş sekmesi altından “Salla ise” kod bloğunu alıp kod alanımıza sürüklüyoruz.

Ardından Matematik sekmesi altından “rastgele değer 1 ve 8 arası” kod bloğunu alıyoruz. Bu kod bloğu yazdığımız iki değer arasında rastgele sayı üretir. Ürettiği sayıları micro:bit üzerindeki ledler üzerinde göstereceğiz, dolayısıyla “sayıyı göster” kod bloğunun içerisine random kod bloğunu yerleştiriyoruz.

Son olarak projemize isim verip bilgisayarımıza indirelim. Ardından micro:bit'e yükleyelim. micro:bit'i her salladığımızda bize 1 ile 6 arasında farklı değerler üretecek.

Projenin Javascript kodlarına bakmak için:

```
1 girdi . onGesture (Gesture.Shake, function () {  
2     temel . showNumber ( Matematik . randomRange ( 1 , 6 ))  
3 })  
4 temel . sonsuza dek ( function () {  
5  
6 })  
7
```

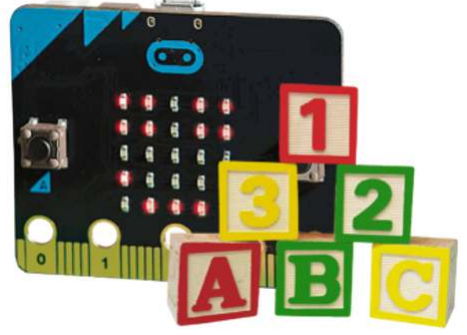


Proje 4: Led Kullanımı

- BBC Micro:Bit Kartı

Bu projemizde micro:bit üzerinde 5x5 ledlerin kullanımını göreceğiz.

Metin yazdırmak için “**dizgi göster**” kod bloğunu Kullanacağız.



Görseldeki gibi “**dizgi göster**” kod bloğunu alıp “**program başladığında**” kod bloğunun içine bırakalım. Ardından “**Hello!**” yazan yeri silip “**Merhaba Dünya <3**” yazalım.



Son olarak projemize isim verip bilgisayarımıza indirelim. Ardından micro:bit'e yükleyelim.

micro:bit'te kayan yazı ile “**Merhaba Dünya <3**” yazdığını göreceksiniz.

“**Merhaba Dünya <3**” yerine dilediğiniz yazıyı yazarak kullanabilirsiniz.

JavaScript kodları aşağıdaki gibidir.

```
1 temel . sonsuza dek ( function () {  
2     temel . showLeds ( `  
3         . . . . .  
4         . #. #.  
5         . . . . .  
6         #. . . #  
7         . # # #.  
8         ` )  
9 })  
10
```

Proje 5: Buton Kullanımı

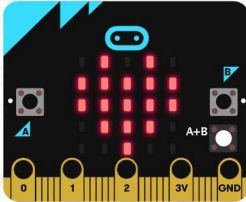
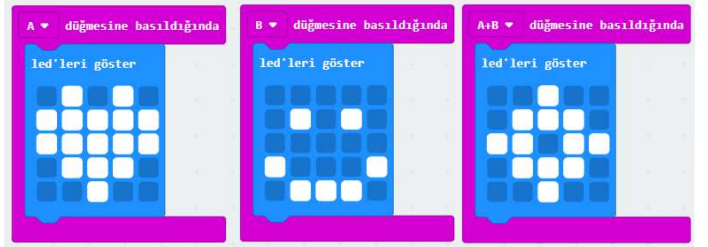
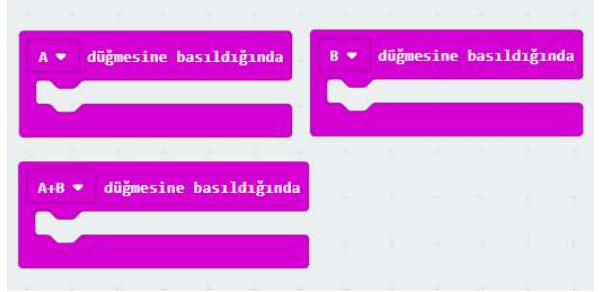
- BBC Micro:Bit Kartı

Micro:bit üzerindeki butonların her birini ayrı olarak programlayabiliriz. micro:bit, her iki butona aynı anda tıkladığımızda üçüncü bir buton varmış gibi davranmaktadır. Dolayısıyla hayali üçüncü butonu da programlayabiliriz.

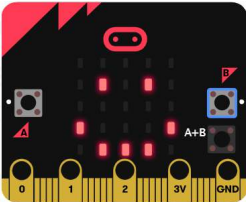
Butonlarla ilgili kullanabileceğimiz kod blokları yandaki gibi:

Butona tıkladığımızda micro:bit'in ne yapmasını istiyorsak kod bloklarını ona göre yerleştiriyoruz. Örnek olarak A butonuna tıkladığımızda "Kalp", B butonuna tıkladığımızda "Gülen Surat" ve her ikisine aynı anda tıkladığımızda(A+B) ise şekildeki gibi kendi ekran çıktınızı veren bir uygulama yapabilirsiniz.

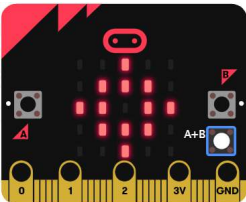
Kodumuzu yazıp indirdikten sonra micro:bit'e yükleyelim ve butonlara basarak nasıl çalıştığını görelim !



A Butonuna basıldığında



B Butonuna basıldığında



A+B Butonuna basıldığında

Projemizin java tarafındaki görüntüsü aşağıdaki gibidir;

```
1 girdi . onPressed (Button.A, function () {
2     temel . showLeds ( `
3         . #. #.
4         # # # # #
5         # # # # #
6         . # # #.
7         . #. .
8     ` )
9 })
10 girdi . onPressed (Button.B, function () {
11     temel . showLeds ( `
12         . . . . .
13         . #. #.
14         . . . . .
15         #. . #
16         . # # #.
17     ` )
18 })
19 girdi . onPressed (Button.AB, function () {
20     temel . showLeds ( `
21         . . #. .
22         . # # #.
23         # #. # #
24         . # # #.
25         . #. .
26     ` )
27 })
28
```

Proje 6: Pong Oyunu

- BBC Micro:Bit Kart

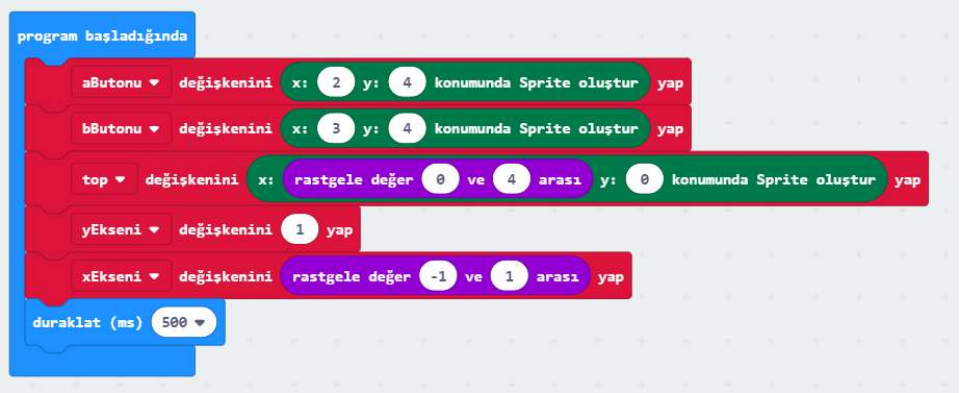
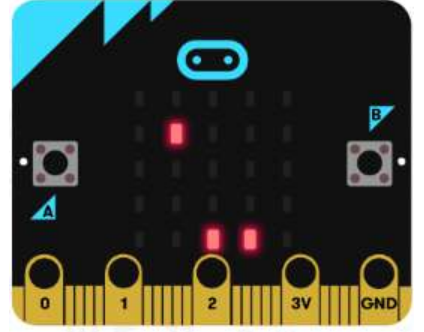
Bu öğreticide, 1 oyuncunun micro:bit kullanarak Oynaması için Pong'un basitleştirilmiş bir sürümünü oluşturacağız.

Zorluk, micro:bit ekranının 25 LED'lik bir 2D ızgaradan (5 x 5 ızgara) oluştuğunu düşünerek oyunu oluşturmaktır.

Gerçek Pong oyununun aksine, ekranın altında bulunan sadece bir raketi kullanacağız. Oynatıcı, A düğmesini (raketi sola hareket ettirmek için) ve B düğmesini (raketi sağa hareket ettirmek için) kullanarak raketi kontrol edecektir.

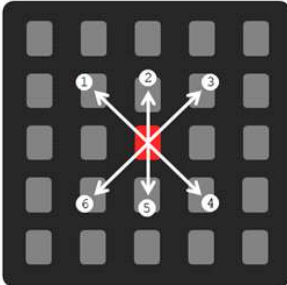
Ekranın üst, sol ve sağ kenarlarına ve küreklere sıçrayan topu temsil etmek için bir hareketli grafik kullanılacaktır. Kullanım topu yakalayamazsa, top ekranın alt kenarına ulaşır ve bu da oyun üzerinden mesaj gönderir.

İlk önce ana hareketli grafikleri başlatacağız:



Ana hareketli grafikler:

Top hareketli grafiği üst sıradan ($y = 0$) rastgele bir x konumunda (0 ile 4 arasında) başlayacaktır. Topa ayrıca iki değişken kullanılarak bir yön verilecektir: $yEkseni$ ve $xEkseni$. Aşağıdaki şema 2D ızgaradaki top hareketli grafiğin 6 farklı yönünü göstermektedir.

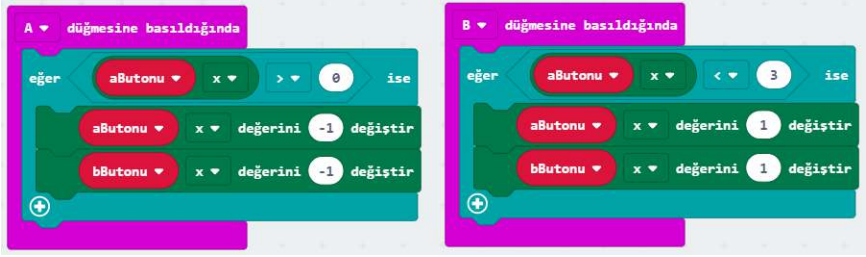


- 1 { directionX = -1
directionY = -1
- 2 { directionX = 0
directionY = -1
- 3 { directionX = 1
directionY = -1
- 4 { directionX = 1
directionY = 1
- 5 { directionX = 0
directionY = 1
- 6 { directionX = -1
directionY = 1

Topu ilk başlattığımızda ve ekranın üst kısmına yerleştirdiğimizde, topun aşağıya doğru yönlendireceği bir yön de belirleriz (yukarıdaki şemadan 4, 5 veya 6 yönü).

Proje 6: Pong Oyunu

- Raket 2 sprite, paddleA ve paddleB'den oluşur, sprite sadece 1 LED'den oluşur, daha geniş bir raket oluşturmak için yan yana iki sprite kullandık (2 LED kadar geniş). Raket alt sıradadır (Y = 4) ve aşağıdaki kod kullanılarak micro:bit'in sol ve sağ düğmelerine basılarak yatay olarak hareket ettirilebilir:

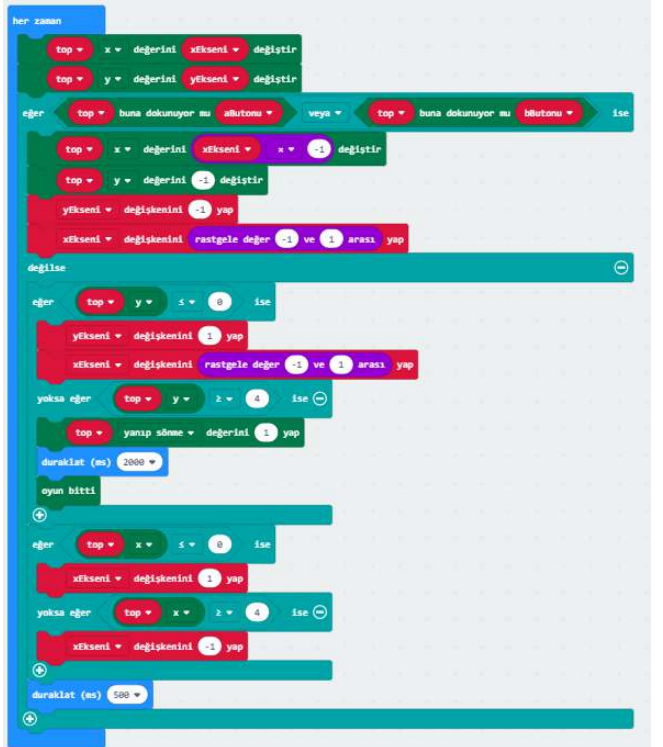


Zıplayan top algoritması:

Oyunun en karmaşık kısmı, top sprite hareketini kontrol etmektir. Her 500ms'de bir top, (directionX, directionY) vektörü tarafından belirlenen yönde hareket eder. Yeni konumuna bağlı olarak, aşağıda verilen kod farklı olası senaryoları algılar:

- Top, ekranın sol, üst ve sağ kenarlarına zıplamalıdır. Her sekme yönünü etkiler.
- Top rakete karşı zıplamalıdır. Bir kez daha sıçrama, topun yönünü etkileyecektir.
- Top ekranın alt kenarına ulaşırsa, oyun bitti mesajı görüntülenmelidir.

Kod şimdi tamamlandı. Bir puanlama sistemi ekleyerek veya topun rakete her vuruşunda topun daha hızlı hareket etmesini (daha kısa bir gecikme / duraklatma kullanarak) yeniden oluşturabilir ve daha sonra özelleştirebilirsiniz.



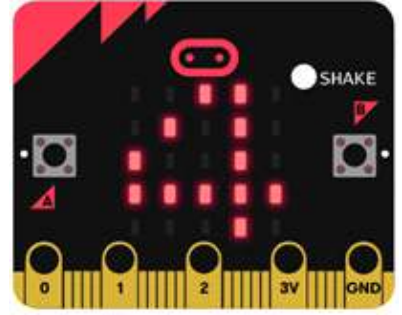
Proje 7: Biletleme Sistemi

- BBC Micro:Bit Kart

Biletleme sistemleri, bir dükkanda(banka, fatura ödeme Merkezi, noter vb.) müşteri sırasını kontrol etmek için kullanılır.

Hemen "<https://makecode.microbit.org/>" adresine giderek, yeni bir proje oluşturup projeye "**BiletlemeSistemi**" adını veriyoruz.

Biletli kişi sayısını sayacağımıza göre "**Değişkenler**" sekmesine tıklayarak "**yeni değişken yarat**" diyoruz "**BiletliKisi**" adında bir değişken tanımlıyoruz.



Bilet sırası, dükkanımızın başlangıçta boş olduğunu varsaydığımız için 0'dan başlamalı. Dolayısıyla Program başladığında bloğumuzun içine değişkenimizi "0" yapan ilgili bloğu ve değişkenimizi gösteren bloğu ekliyoruz.



Görevli kişi müşteriye çağırmak için "A" butonuna basar, değişkenimiz "1" artar ve müşterimiz işlem yapmaya başlar.

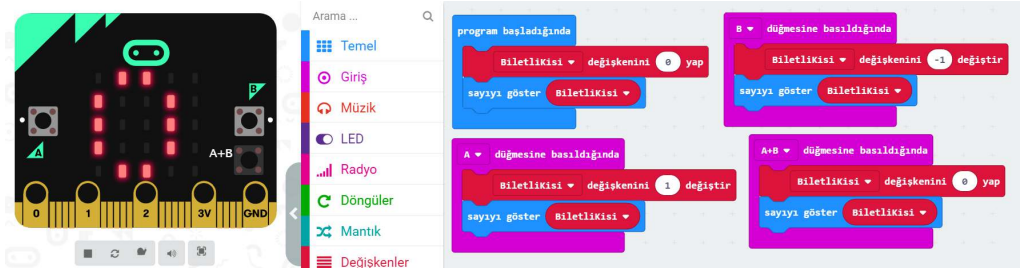


Proje 7: Biletleme Sistemi

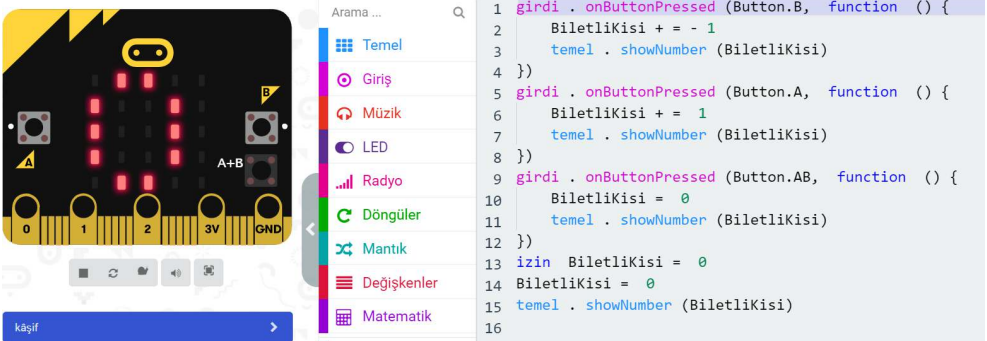
Görevli herhangi bir olumsuz duruma karşı “B” düğmesine basarak eylemlerini iptal edebilmelidir. Bu durumda sıra numarası 1 azalmalıdır.



Biletleme sistemini sıfırlamak için görevli kişi hem A hem de B düğmelerine aynı anda basmalıdır. Kuyruk numarası sıfırlanacaktır.



Projemizin Java kodları da görseldeki gibidir;



Proje 8: Trafik Lambası

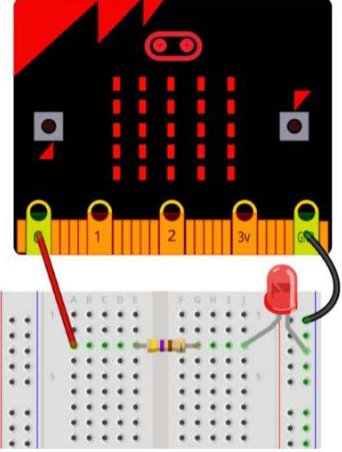
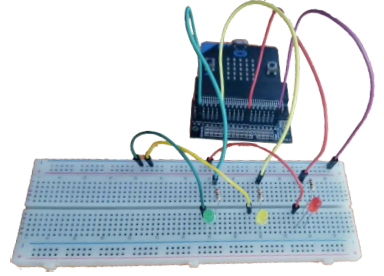
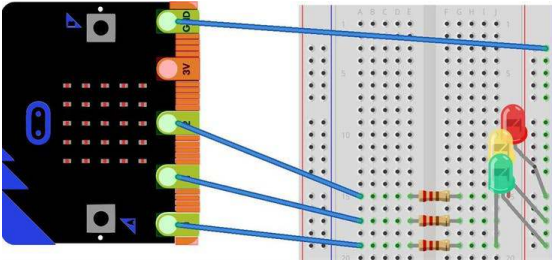
- BBC Micro:Bit Kart
- BBC Micro:Bit Geniřletme kartı
- Breadboard
- Jumper kablo
- Direnç (220 ohm)
- LED (Sarı, Kırmızı, Yeřil)

Bu projemize micro:bit kullanarak bir trafik ışığı sistemi yapacağız. Temel olarak, lehimsiz bir breadboard üzerinde micro:bit kullanarak LED'in nasıl kontrol edileceğini göreceğiz.

LED'lerin GND'sini bağlayarak GND olarak kullanabilmemiz için GND'yi breadboard'un GND terminaline bağlamamız gerekir.

3V ile sınırlayabilmemiz için dirençleri kullanmamız gerekiyor. 470 ohm en iyisidir ancak 220 ohm da kullanabilirsiniz.

Bunun için, direncin sol tarafına PIN0 bağlandı, bu yüzden direncin sağ tarafını LED'in + ve 'sine bağlamamız gerekiyor ve LED'in veve LED'i GND'ye bağlanıyor breadboard'un son satırı. Aşağıdaki resim yalnızca bir LED'in bağlantısını gösterir.

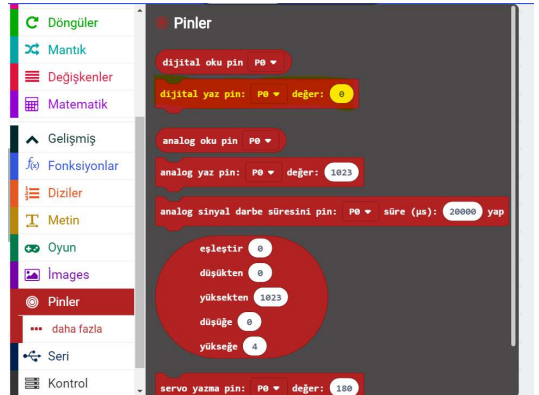


Benzer şekilde, yanda gösterildiği gibi iki LED daha bağlamamız gerekiyor.

řimdi makecode kullanarak bu projenin kodlama kısmını görelim.

Hemen “ <https://makecode.microbit.org/> ” adresine giderek, yeni bir proje oluşturup projeye “Trafik Lambası” adını veriyoruz.

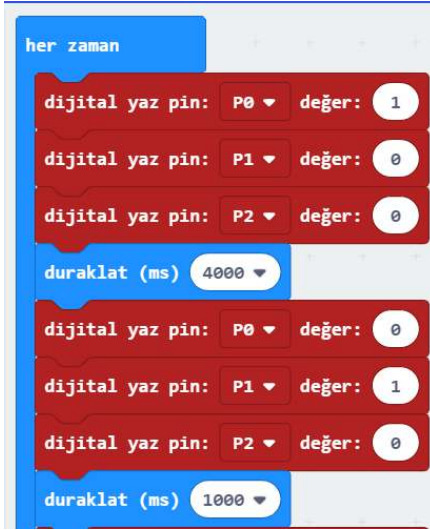
Geliřmiř'i tıklayın ve Pinler'e gidin. Sonra “**dijital yazma pin bloęu**” alın ve “**sonsuz kadar bloęun içine yerleřtirin**” ardından ledimizin bir süre yanması ve sönmesi için kullanacağımız “**duraklat**” bloęunu ekleyin.



Proje 8: Trafik Lambası

0'ın kapalı ve 1'in açık olduğunu biliyoruz. Herhangi bir PIN'i "0" olarak ayarladığımızda ışığı kapatır ve PIN'i "1" olarak ayarladığımızda LED'ler yanar. Bu yalnızca bir LED'in kodudur; benzer şekilde micro:bit'in pinlerini değiştirerek kodlayabiliriz.

"Dijital yaz" kod bloğuna P0 pinine bağladığımız yeşil ledimizin yanması için "1" değerini veriyoruz. Sadece yeşil ışığımızın yanmasını istediğimiz için "p1" ve "p2" pinlerine "0" değerini veriyoruz ve 4 saniye milisaniye cinsinden (4000 ms) duraklama ekliyoruz.



Yeşil ışığımızı 4sn yaktık daha sonrasında sarı ışığı 1 sn yaktık şimdi sırada kırmızı ışık var. Kırmızı ışığımızı 4 sn yanması için kırmızı ledimizin bağlı olduğu pini yani "p2" pinine 1 değeri veriyoruz. Geri kalan pinlere "0" değeri veriyoruz ki yanık olan varsa sönsün. Duraklama ekleyerek 4000 ms olmasına dikkat edelim. Böylece kırmızı ledimiz yanıyor ve 4 sn bekliyor.

Daha sonra sadece sarı ledimizin ilgili pinini yani "p1" pininin yanmasını istediğimizden dolayı "dijital yaz" bloğunu "p1" pinine ayarlayarak, 1 saniye yanmasını istediğimiz için milisaniye cinsinden 1000 ms "duraklama" bloğu ekliyoruz.



Proje 8: Trafik Lambası

Şimdi kırmızı ışık 4 sn yanıyor ve sonrası için kırmızı ile sarı 1 sn yanıp tekrar yeşil yanması gerekiyor. Çünkü kırmızı “dur”, kırmızı ile sarı “hazırlan ve dur” anlamı taşımaktadır. Bunun için “dijital yaz” bloğuna “p1” ve “p2” için değerlerini “1” yaparak ekliyoruz. 1 sn boyunca yanmasını istediğimiz için 1000 ms’ lik duraklama ekliyoruz ve bütün kod bloğu şu şekilde oluyor;

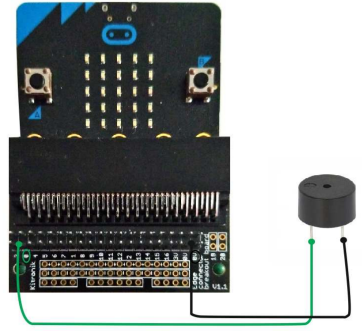
Projeniz hazır şimdi micro:bit'inize yükleyip çalıştırabilirsiniz. Dilerseniz “JavaScript” butonuna tıklayarak jascasript kodlarına erişebilirsiniz.

```
basic.forever(function () {  
  pins.digitalWritePin(DigitalPin.P0, 1)  
  pins.digitalWritePin(DigitalPin.P1, 0)  
  pins.digitalWritePin(DigitalPin.P2, 0)  
  basic.pause(4000)  
  pins.digitalWritePin(DigitalPin.P0, 0)  
  pins.digitalWritePin(DigitalPin.P1, 1)  
  pins.digitalWritePin(DigitalPin.P2, 0)  
  basic.pause(1000)  
  pins.digitalWritePin(DigitalPin.P0, 0)  
  pins.digitalWritePin(DigitalPin.P1, 0)  
  pins.digitalWritePin(DigitalPin.P2, 1)  
  basic.pause(4000)  
  pins.digitalWritePin(DigitalPin.P0, 0)  
  pins.digitalWritePin(DigitalPin.P1, 1)  
  pins.digitalWritePin(DigitalPin.P2, 1)  
  basic.pause(1000)  
})
```

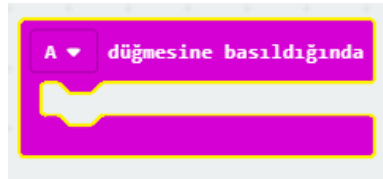


Proje 9: Buzzer ile müzik seslendirme

- BBC Micro:Bit Kartı
- BBC Micro:Bit Genişletme kartı
- Breadboard
- Jumper kablo
- Buzzer



Giriş sekmesinin içinden “A düğmesine basıldığında” seçeneğini seçiyoruz.



Neden bunu kullanıyoruz çünkü başlat veya sonsuza kadar kullanırsak , zil sesi sonsuza kadar çalacaktır. “A” düğmesine her bastığınızda müzik tekrar tekrar çalacaktır.

Müzik sekmesinin altından “**start melody dadadum repeating bir kez**” yazan bloğu seçiyoruz.



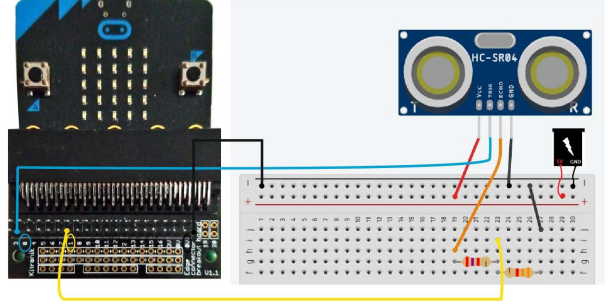
Ardından “A düğmesine basıldığında” bloğunun içine yerleştiriyoruz. Dilersek araya duraklama süresi ekleyerek farklı bir melodi daha ekleyebiliriz.



İşte bu kadar, şimdi micro:bit' inize yükleyerek çalıştırabilirsiniz.

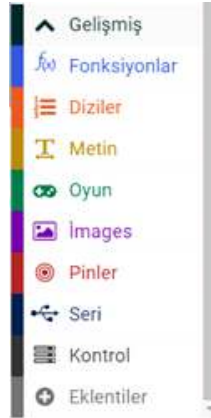
Proje 10: Mesafe Ölçümü

- BBC Micro:Bit Kart
- BBC Micro:Bit Genişletme kartı
- Breadboard
- Jumper kablo
- HCSR04 sensörü
- Direnç (2k7 ve 3k3)

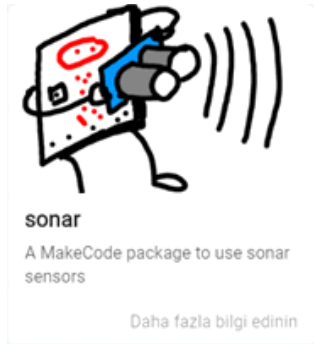


Not: Bu projede harici 5v bağlantısı kullanılmıştır. Dilerseniz arduino'nuz varsa “5v” ve “GND” bağlantılarını kullanabilirsiniz. 5v adaptör yardımı ile de çalıştırmanız mümkündür.

Uygulamanın makecode ile çalışabilmesi için eklentiler kategorisinin altındaki sonar eklentisini editöre dahil etmemiz gerekir. Böylelikle HC-SR04 ultrasonik sensörü kod blokları ile kullanabileceğiz. Blok editöründe gelişmiş sekmesinin en altında eklentiler bölümünün olduğu görülecektir. Buraya tıklayarak eklentiler sekmesini açalım.

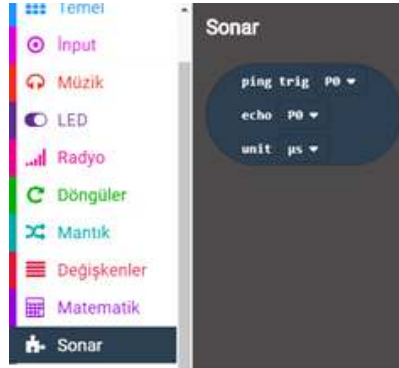


Buradan sonar eklentisine tıklayarak editöre gerekli blokları ekleyelim.

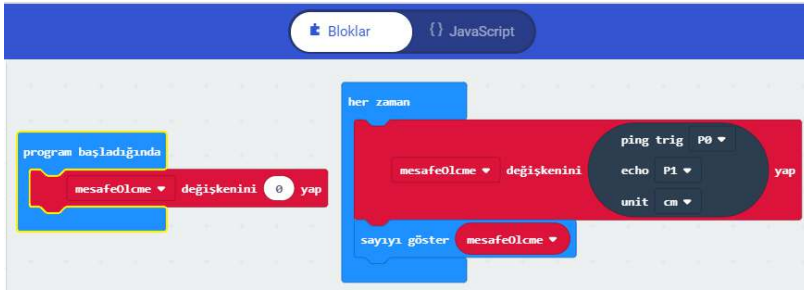


Proje 10: Mesafe Ölçümü

Eklentiye editöre eklediğimizde Sonar adında yeni bir kategorinin eklendiğini göreceğiz. Bu kategori altında Trig ve Echo pinlerini bağlayacağımız micro:bit bağlantı uçlarını seçebileceğimiz bir kontrol karşımıza çıkacaktır. Bu kontrol ile ölçülen mesafenin us cinsinden süresi, ölçülen mesafenin cm mi yoksa inch mi olacağı seçilebilmektedir.



Uygulamada ölçülen değeri mesafe adında bir değişkene aktardık. Bu değişkenin değeri micro:bit kartında sayısal bir değer olarak gösterilmiştir. Şekilde uygulamanın kod blokları gösterilmektedir.

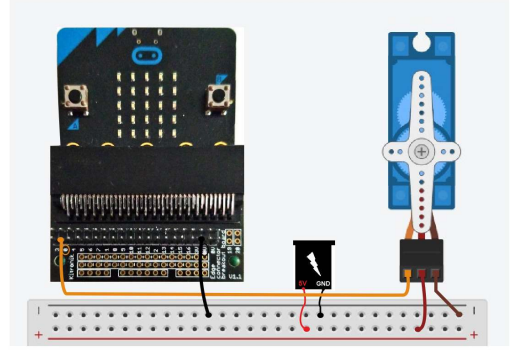


JavaScript kodu aşağıdaki gibidir.

```
let mesafeOlcm = 0
basic.forever(function () {
  mesafeOlcm = sonar.ping(
    DigitalPin.P0,
    DigitalPin.P0,
    PingUnit.MicroSeconds
  )
  basic.showNumber(mesafeOlcm)
})
```

Proje 11: Servo Motor kontrolü

- BBC Micro:Bit Kartı
- BBC Micro:Bit Genişletme kartı
- Breadboard
- Jumper kablo
- SG90 Servo motor
- 5V adaptör ve dişi barrel jack



Not: Bu projede harici 5v bağlantısı kullanılmıştır. Dilerseniz Arduino’nuz varsa “ 5v” ve “GND” bağlantılarını kullanabilirsiniz. 5v adaptör yardımı ile de çalıştırmanız mümkündür.

Servo motorlar belirli bir açı aralığında hareket edebilen motorlardır. 0 – 180 derece arasında hareket edebildikleri gibi 360 derece dönebilen çeşitleri de vardır. 5V ile çalışabilecek şekilde tasarlanmıştır. Micro:bit 3V ile çalıştığı için 5V dışarıdan harici olarak bağlamamız gerekmektedir.

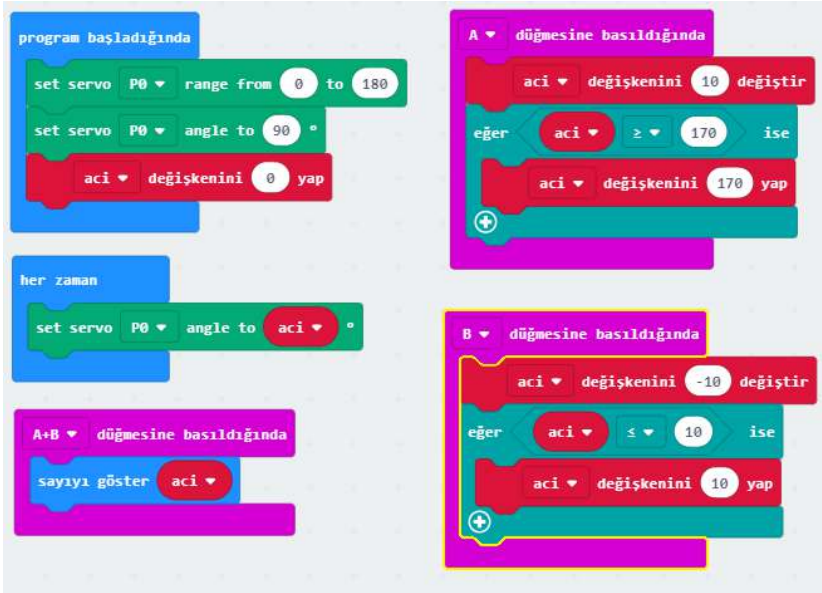
Yukarıdaki devre şemasında breadboard yardımı ile harici olarak 5v güç bağlıyoruz.

Servo motorumuzun kırmızı kablosunu breadboarddaki artı yuvaların bir tanesine, kahverengi kablosunu ise breadboard’daki eksi yuvaların bir tanesine bağlıyoruz. Servo motorumuzun turuncu kablosu sinyal kablosudur yani micro:bit’imizden gelen sinyalleri motorumuza iletmekle görevlidir dolayısıyla microbitin p0 pinine bağlanır. Son olarak microbit’imizin gnd pinini breadboard’daki eksi yuvalara bağlamamız gerekir. Devremiz hazır, şimdi sırada bloklar ile kodlarımızı hazırlamak var.

Uygulamayı bloklar ile çalıştırabilmek için servo isimli eklentiye projeye dahil etmemiz gerekmektedir.

Uygulamada B butonuna basıldığında Servo motorun açı değeri artırılmakta, A butonuna basıldığında azaltılmaktadır. A ve B butonuna beraber bastığınızda micro:bit bunu 3. buton gibi algılayacağından dolayı bu butona basarak açı değerini ledler aracılığı ile görebilirsiniz. A butonuna 10 ar derece azaltma, B butonuna 10 ar derece arttırma görevi vereceğiz. 0 – 180 derece arasında işlem yapabileceğimiz için açı değerimiz 10 derecenin altında ve 170 derecenin üzerinde bir değere ulaşmamalı. Çünkü micro:bit ledler ile 190 dereceyi gösterebilir fakat servo motor bu dereceyi gösteremez. Dolayısıyla projemizde hata olmuş olur. Şimdi görseldeki gibi bloklarımızı hazırlayalım.

Proje 11: Servo Motor kontrolü



Eğer uygulamanızı javascript ile kodlamak isterseniz aşağıdaki görseli inceleyebilirsiniz.

```
Bloklar {} JavaScript

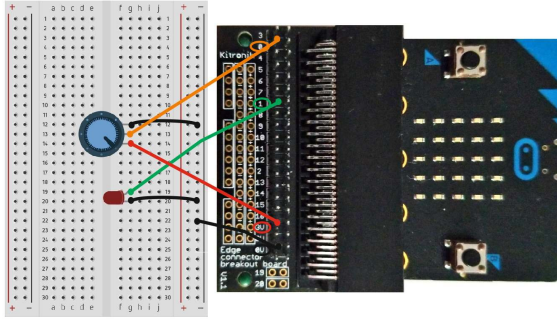
1 input.onButtonPressed(Button.A, function () {
2   aci += 10
3   if (aci >= 170) {
4     aci = 170
5   }
6 })
7 input.onButtonPressed(Button.B, function () {
8   aci += -10
9   if (aci <= 10) {
10    aci = 10
11  }
12 })
13 input.onButtonPressed(Button.AB, function () {
14   basic.showNumber(aci)
15 })
16 let aci = 0
17 servos.P0.setRange(0, 180)
18 servos.P0.setAngle(90)
19 aci = 0
20 basic.forever(function () {
21   servos.P0.setAngle(aci)
22 })
23
24
```

Proje 12: Potansiyometre ile led parlaklığı

- BBC Micro:Bit Kartı
- BBC Micro:Bit Genişletme kartı
- Breadboard
- Jumper kablo
- Potansiyometre
- Led

Potansiyometre bağladığımız ledin parlaklığını ayarlayabilmemiz gibi bir çok görevde kullanılabilir. Bu projeden ledin parlaklığını ayalamada kullanacağız.

Micro:bit kartımızı genişletme kartına takarak projemize başlıyoruz. Breadboard aracılığı ile potansiyometre ve led bağlantılarını aşağıdaki şemada belirtildiği gibi yapıyoruz.



Daha sonra bloklar ile kodlayacağımızdan dolayı ilgili sayfaya gidiyoruz. Tek yapmamız gereken potansiyometredeki değeri okumak yani P0 dan okuduğumuz değeri P1 de göstereceğiz. Bunun için **“her zaman”** kod bloğunun içerisine aşağıdaki resimde gösterilen pinler sekmesinin altında yer alan **“analog yaz pin”** ve değer belirttiğimiz kısımda **“analog oku pin P0”** eklemeliyiz. İşte bu kadar, indir seçeneğine basarak micro:bit kartına yükleme işlemi yapıyoruz. Yükleme işlemi bittikten sonra otomatik olarak çalışacaktır.

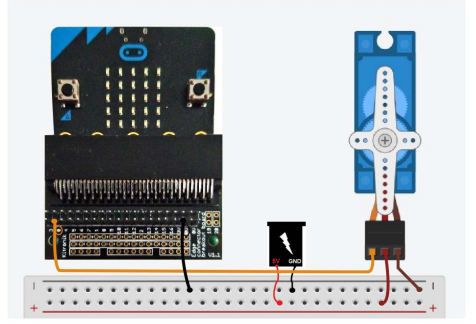


Projenizin python kodları yandaki gibidir;



Proje 13: Pusula ile Motor kontrolü

- BBC Micro:Bit Kartı
- BBC Micro:Bit Genişletme kartı
- Breadboard
- Jumper kablo
- SG90 Servo motor
- 5V adaptör vedışı barrel jack



Not: Bu projede harici 5v bağlantısı kullanılmıştır. Dilerseniz arduinonuz varsa “5v” ve “GND” bağlantılarını kullanabilirsiniz. 5v adaptör yardımı ile de çalıştırmanız mümkündür.

Micro:Bit kartının pusula sensörünün açış deęerini kullanarak P0' a baęlı olan servo motorumuzu kontrol edeceęiz. Yukarıdaki görselde devre şeması görölmektedir. Göröldüęü üzere servo motorumuzu 5V'luk bir güç kaynaęı ile besliyoruz. Servo motorun turuncu kablolu pini micro:bit kartının P0 isimli pinine baęlıyoruz.

Bloklar ile kodlama yapacaęımız için ilgili sayfayı açıyoruz ve yeni proje oluşturunuz. Yeni bir deęişken oluşturalım adı “aciDeęiskeni” olsun.

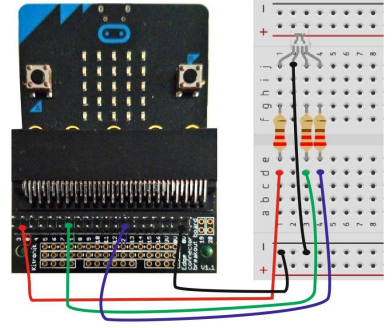
Program baęladıęında pusula sensörünü kalibre etmemiz gerekir. Kalibrasyon işlemi için led matrisin ortasında bir led yanıp sönecektir. Daha sonra bütün ledler yanacak şekilde microbit kartını saęa, sola, yukarıya ve aşıęıya doęru hareket ettiriyoruz. Bütün ledler yandıktan sonra gülen bir yüz belirecektir. Kalibrasyon işlemi tamamlandı.



Blokları şekilde gösterilen biçimde renkler yardımı ile kodluyoruz. Uygulama başladıęında pusula kalibre edilecek ve servo motorun aciDeęeri 0'a eşitlenecektir. Ardından servo motorun aciDeęeri pusulanın yönlendirmesine göre deęişecektir. Eęer aciDeęeri 0-180 derecenin dıřında bir deęer alırsa üęün surat ifadesi gözökecektir. Projeyi micro:bit kartına yükleyerek kullanmaya bařlayabilirsiniz.

Proje 14: RGB led kullanımı

- BBC Micro:Bit Kartı
- BBC Micro:Bit Genişletme kartı
- Breadboard
- Jumper kablo
- 220 ohm direnç (3 adet)
- RGB Led

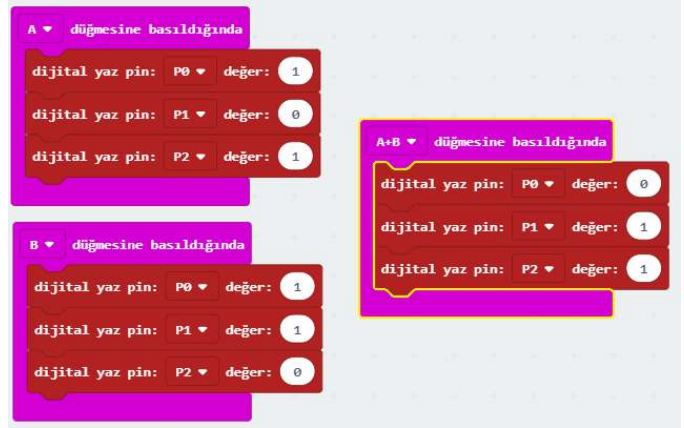


RGB Led'ler, içerisinde 3 farklı renk barındırır. R harfi kırmızıyı, G harfi yeşili, B harfi maviyi temsil etmektedir. RGB Led'lerde 4 adet bacak bulunur. Her bir bacak bir rengi temsil ederken kalan bir bacak ise artı(Anot) veya eksi(Katot) görev üstlenir. Devreyi şekildeki gibi kuruyoruz ardından bloklar ile kodlama yapacağımızdan dolayı ilgili sayfayı açıyoruz ve yeni proje oluşturuyoruz.

Not: Eğer eksi bacağı şekildeki devredeki gibi eksi pinine bağladığınızda çalışmıyorsa Rgb ledinizi artı pinine bağlayınız.

Projemizin adını yazdıktan sonra aşağıdaki blokları renklerden yardım alarak kodluyoruz.

A butonuna basıldığında yeşil, B butonuna basıldığında mavi ve A+B butonuna basıldığında kırmızı yandığını göreceksiniz.

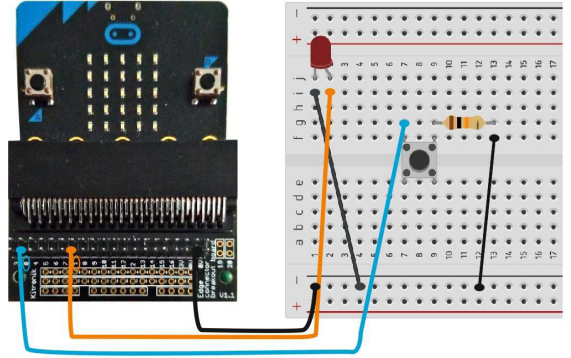


Dilerseniz javascript kodlarıyla da kodlama yapabilirsiniz.

```
input.onButtonPressed(Button.A, function () {
  pins.digitalWritePin(DigitalPin.P0, 1)
  pins.digitalWritePin(DigitalPin.P1, 0)
  pins.digitalWritePin(DigitalPin.P2, 1)
})
input.onButtonPressed(Button.AB, function () {
  pins.digitalWritePin(DigitalPin.P0, 0)
  pins.digitalWritePin(DigitalPin.P1, 1)
  pins.digitalWritePin(DigitalPin.P2, 1)
})
input.onButtonPressed(Button.B, function () {
  pins.digitalWritePin(DigitalPin.P0, 1)
  pins.digitalWritePin(DigitalPin.P1, 1)
  pins.digitalWritePin(DigitalPin.P2, 0)
})
```

Proje 15: Buton ile LED kullanımı

- BBC Micro:Bit Kartı
- BBC Micro:Bit Genişletme kartı
- Breadboard
- Jumper kablo
- 10K ohm direnç
- Buton
- Led diyot



micro:bit kartına bazı durumlarda harici buton bağlamak isteyebilirsiniz. Bunun için aşağıdaki bağlantılarımızı yapalım.

Butonun bir bacağı P0 pinine, diğer bacağına 10k ohm direnç bağlayalım ve direnci breadboardın eksi pinine bağlayalım. Ledin eksi bacağı P1 pinine bağlayalım. Ledin diğer bacağı P1 pinine bağlayalım.

Projemizin blokları aşağıdaki gibidir.



Projemizin JavaScript kodları aşağıdaki gibidir.

```
input.onPinPressed(TouchPin.P0, function () {  
  pins.digitalWritePin(DigitalPin.P1, 1)  
  basic.pause(500)  
  pins.digitalWritePin(DigitalPin.P1, 0)  
  basic.pause(500)  
})
```





youtube.com/robotistan

BLOG

robotistan



maker.robotistan.com

Robotistan Elektronik Ticaret A.Ş.

info@robotistan.com - www.robotistan.com

Tel: 0850 766 0 425