

pico
bricks

Öğretmen Kitabı

when

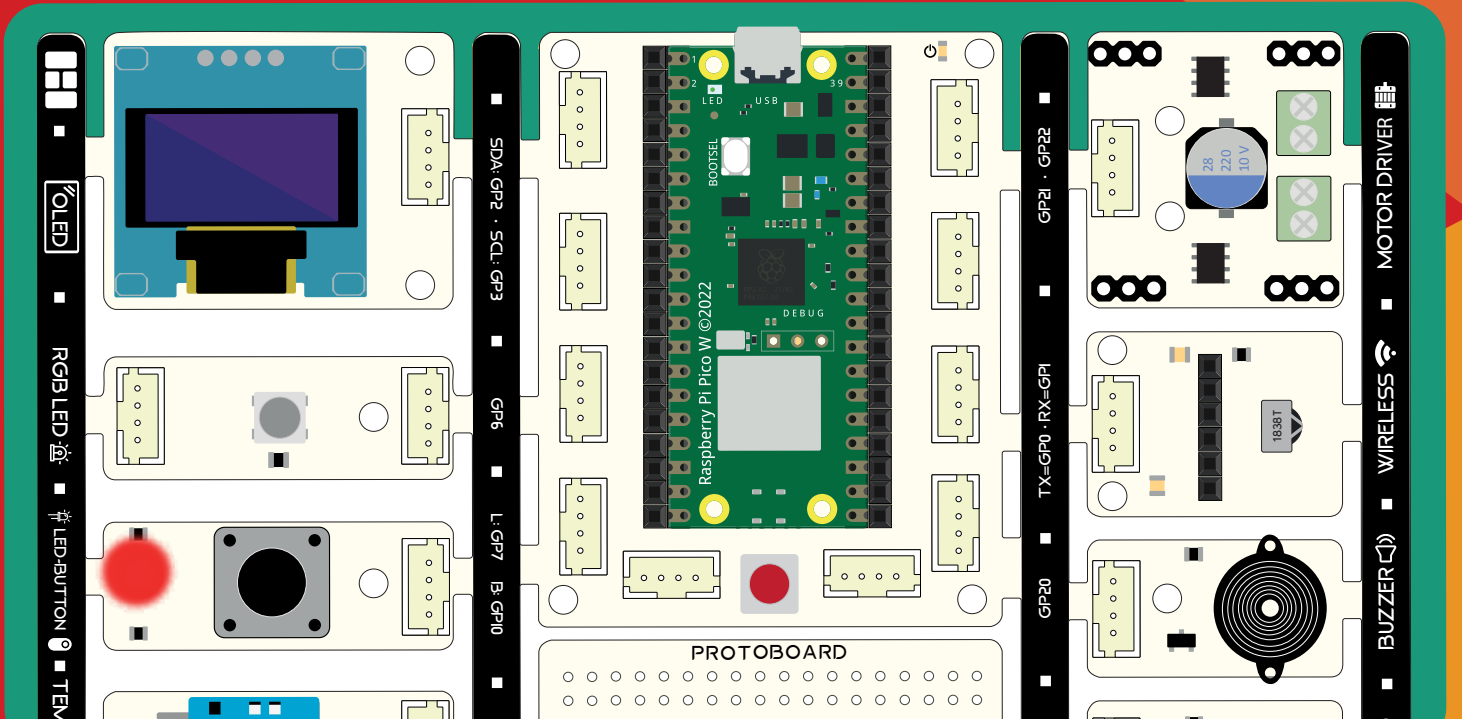
PicoBricks

Kablo Karmaşasını Kapat

Programlama Zorluğunu Kapat

forever

PicoBricks ile Yaratıcılığınızı Geliştirin





Copyright © 2022 Robotistan

Ticari kullanım dışında, bu kitaptaki fotoğraf ve içerikleri kaynak göstererek kopyalayabilir, çoğaltabilir ve düzenleyebilirsiniz.

Yazarlar: Selim Gayretli, Hakan Ataş
Grafik Tasarım: Elanur Tokalak
Editör: Naze Gizem Özer

Powered by



Bu kitaptaki aktiviteler, PicoBricks'in tüm versiyonlarıyla uyumludur.



[Education GitHub](#)



[Education Wiki](#)

iÇİNDEKİLER

İÇİNDEKİLER

1. Giriş	08
2. Eğitim Planı	09
3. Eğitimde PicoBricks'i Nasıl Kullanırız?	10
3.1 PicoBricks Nedir?	10
3.2 PicoBricks'le Kullanılması Önerilen Materyeller	11-12
4. Ders İçeriğinde Kullanılabilen Materyeller	13

PART 1

6. Giriş	15
6. PicoBricks'in Ortadan Kaldırdığı Sorunlar	15
7. Ders içeriğinde kullanılacak materyaller	16
8.1 Bilgisayar Nedir?	17
8.2 İşlemci	17
8. Depolama Ünitesi	17
9.1 Kısa Süreli Bellek	17
9.2 Uzun Süreli Bellek	17
9. Girdi Aygıtları	18
10. Çıktı Aygıtları	18
11. Mikroişlemci Nedir?	18
12. Raspberry Pi Pico Nedir?	18
13. PicoBricks'i İnceleyelim	19
14. 1. Aktivite - PicoBricks Modüllerini Tanıyalım	20
15. 2. Aktivite - Bilgisayarsız Aktivite	24
16. PicoBricks Modüllerini Test Edelim	25
17.1 MicroBlocks ile PicoBricks'e Kod Yükleyelim	31
17.2 Aktivitenin MicroBlocks Kod Blokları	31
17. Dersin Hedefleri	33

Part 2

18. MicroBlocks Bloklarını İnceleyelim	35
19. PicoBricks ile Basit İşlemler	36
19.1 MicroBlocks IDE'ye PicoBricks Kütüphanesini Yükleyelim	36
19.2 Proje: Renkli Notalar	47
19.3 Projemizi Bilgisayara Kaydedelim	48

Part 3

20. Bir Değişken Tanımlayalım ve Döngüye Giriş Yapalım	50
20.1 Değişken Nedir?	50
20.1.1 MicroBlocks Platformunda Değişkenler Oluşturalım	52
20.2 Döngü Nedir?	52
20.2.1 Tekrar Döngüsünde Ayarladığımız Değerde LED'i Yakalım	52
21. Counter Aktivitesi	57

Part 4

22. Kaplumbağa ile Döngüler	62
23. Turtle Kütüphanesini MicroBlocks ' a Yükleyelim	62
22.1 Aşağıdaki kaplumbağa projesini yorumlayalım	63
23.2 Kaplumbağa ile Bazı Şekiller	64

Part 5

24. Bilgisayarsız Aktivite: Amiral Battı Oynayalım	70
24.1 Amiral Battı Çalışma Kağıdı	73
25. OLED Ekrandaki Koordinatlar	74
26. Yukarıdan Aşağıya Kayan Yazı	75
27. Proje	77

Part 6

28. Koşul Yapılarına Giriş	80
29. IF-ELSE	80
30. Bilgisayarsız Aktivite	81
31. Skor Aktivitesi	82
32. Ekstra Proje: 2 Oyunculu Skor Aktivitesi	85
32.1 Projenin Algoritması	85
32.2 Projenin MicroBlocks Kodu	85
32.3 Taş-Kağıt-Makas MicroBlocks Projesi	87

Part7

33. For Döngüsü	93
33.1 For Döngüsü ile Sayaç (1-10)	94
33.2 For Döngüsü ile Sayaç (10-1)	95
33.3 For Döngüsü ile (2-10) 2'şer 2'şer Artan Sayaç	95
33.4 For Döngüsü ile (2-10) 2'şer 2'şer Azalan Sayaç	96
34. 7. Bölümün Projesi: Show Your Reaction	96

Bonus Proje: Numara Tahmin Etme Oyununun Detayları

35. Projenin Algoritması	104
36. Projenin Bağlantı Şeması (PicoBricks v1.0)	105
37. Projenin Görselleri (PicoBricks v1.0)	105
38. Projenin Görselleri (PicoBricks v1.2)	105
39. Projenin MicroBlocks Kodları	106

Part 8

40. Listeler	110
40.1 Bir Liste Tanımla	110
40.2 Listeden Bir Eleman Çağırma	111
40.3 Listenin Uzunluğunu Bulma	112
40.4 Listeye Eleman Ekleme	112
40.5 Listedeki Bir Elemanı Değiştirme	113
40.6 Listeden Bir Elemanı Silme	113
40.7 Listedeki Bir elemanı Bulma	114
40.8 Değer Aralığını Belirlediğimiz Kadar Elemanı Çağırılım	114
41. 8. Bölümün Projesi: Pi Sayısının İlk 100 Basamağı	115

Part 9

42. Tanımla - Kendi Kod Bloğunu Oluştur	119
59.1 Command ve Reporter Arasındaki Fark	120
59.2 Define ile Şekiller	121
43. Define(Reporter) ile Sayıyı İkiye Katlayın	124

1. STEM Projesi - Action Reaction

2. STEM Projesi - Dominate The Rhythm

3. STEM Projesi - Alarm Clock

4. STEM Projesi - Know Your Color

Fasikül ve Çalışma Kağıdı

Giriş

Günümüzde programlama eğitimi, gelişen yeni teknolojiyle beraber hızla önem kazanmaktadır. İlkokul, ortaokul, lise ve üniversite dahil öğretim ortamlarının hemen hemen hepsini kapsayan eğitim içerikleri hızla geliştirmeye devam etmektedir. Bu eğitim içerikleri, gelişen teknolojiyle paralel olarak sürekli kendini güncellemelidir.

Fiziksel programlama eğitimi de her eğitim süreci gibi beraberinde bir kaçtane zorluk getirmektedir. Bu zorluklar fiziksel programlamada kullanılan araçlar ile çeşitlilik gösterebilir. Fiziksel programlamada yaşanan başlıca sorunlar; kablolama ve programlama kaynaklı sorunlar olarak ikiye ayrılır. Başlıca karşılaşılan kablolama sorunları; lehimleme, pin bağlantı noktalarının fazlalığından dolayı yaşanan kablo karışıklığı vb. sorunlardır. Programlama alanında yaşanan başlıca sorunlar ise noktalama işaretleri (Syntax), programlama dillerinin çeşitliliğinden dolayı yaşanan ezber sorunlar, kapsamlı projelerde; yazılan kodların fazlalığından dolayı kod okumakta zorlanması, vb. bir çok sorunla karşılaşılabilir. Fiziksel programlama eğitimi için öğretim ortamlarında kullanılması amacıyla geliştirilen bu yeni araçlar yaşanan bu zorlukları ortadan kaldırmayı amaçlamaktadır. Bu zorluklar ortadan kaldırılırken eğitimin kalitesini düşürmemek ana hedefdir.

PicoBricks ekibi olarak hazırladığımız bu eğitim kitabında, PicoBricks geliştirme kartını kullanarak fiziksel programlama eğitimi vermeyi amaçladık. PicoBricks, fiziksel programlamada yaşanan sorunları ortadan kaldırmak amacıyla hazırlanmış bir geliştirme kartıdır. Modüler yapısı sayesinde kablolama, üzerinde bulundurduğu mikrodenetleyici kartı Raspberry Pi Pico'nun sağladığı programlama esnekliği sayesinde ise programlama da yaşanan sorunları ortadan kaldırmaktadır.

PicoBricks hakkında daha detaylı bilgi edinmek için picobricks.com web sitesini ziyaret edebilirsiniz.

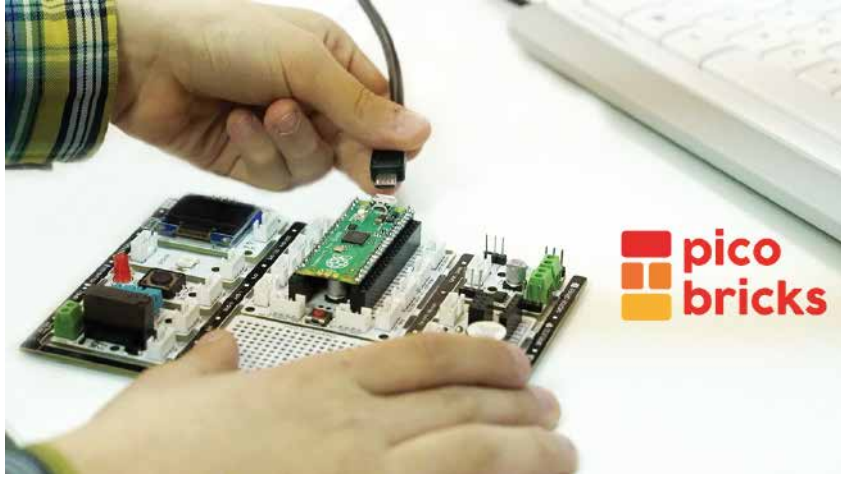


➤ Eğitim Planı

PicoBricks eğitim planı, 9 Bölüm + 4 STEM etkinliği olarak tasarlanmıştır. Öğretim süreci içerisinde PicoBricks proje geliştirme kartı ile beraber MicroBlocks IDE programlama aracı kullanılmaktadır. PicoBricks eğitim kitabı bölümlerinin detayları şu şekildedir;

<ul style="list-style-type: none">▪ Part 1• Unplugged Etkinlikler• PicoBricks Tanıtım?	<ul style="list-style-type: none">▪ Part 8• Listeler
<ul style="list-style-type: none">▪ Part 2• MicroBlocks tanıtım• Picobricks ile temel işlemler	<ul style="list-style-type: none">▪ Part 9• Define ile fonksiyonlar
<ul style="list-style-type: none">▪ Part 3• Değişkenler• Döngüye giriş	<ul style="list-style-type: none">▪ STEM Projeleri
<ul style="list-style-type: none">▪ Part 4• Döngüler• Turtle kütüphanesi	<ul style="list-style-type: none">▪ 1. STEM Projeleri• Action-Reaction
<ul style="list-style-type: none">▪ Part 5• Unplugged - Amiral battı etkinliği• OLED ekran üzerindeki koordinat noktaları	<ul style="list-style-type: none">▪ 2. STEM Projeleri• Dominate The Rhythm
<ul style="list-style-type: none">▪ Part 6• Koşul ifadelerine giriş (if- else)• Unplugged - Taş - Kağıt - Makas etkinliği• Picobricks ile Taş - Kağıt - Makas etkinliği• Skor etkinlikleri	<ul style="list-style-type: none">▪ 3. STEM Projeleri• Alarm Clock
<ul style="list-style-type: none">▪ Part 7• For döngüsü• Bonus Proje- Sayı tahmin oyunu• Unplugged- Akınatör etkinliği	<ul style="list-style-type: none">▪ 4. STEM Projeleri• Know Your Color

➤ Picobricks'i Eğitimde Nasıl Kullanırız?



■ PicoBricks Nedir?

PicoBricks, fiziksel programlamada yaşanan zorlukları ortadan kaldırıp. Bu zorluklarla ortaya çıkan zaman kaybını, daha yaratıcı projeler ortaya çıkarmak için harcamayı hedefleyen bir proje geliştirme kartıdır.

Picobircks, modüler yapısı sayesinde fiziksel programlamada yaşanan lehimleme, kablo karışıklığı, pin noktalarının yanlış bağlanması vb. zorlukları ortadan kaldırır. Aynı zamanda üzerinde bulundurduğu mikro denetleyici kartı Raspberry Pi Pico 'nun kolay kodlanabilir olması ve bir çok kodlama platformunu destekliyor olmasından dolayı programlama aşamasında yaşanan zorluğuda ortadan kaldırmaktadır.

PicoBricks, hem blok tabanlı, hem de metin tabanlı programlama araçlarını desteklemektedir. MicroBlocks IDE sayesinde blok tabanlı kodlama yaparak projelerimizi kısa süre içerisinde kodlayabiliriz. Blok tabanlı programlama araçları kod yazarken yaşanan noktalama işaretleri, ezber gerektiren fonksiyonlar vb. bir çok zorluğu ortadan kaldırmaktadır. Bu sayede küçük yaş gruplarında ya da programlamaya yeni başlayan kişilerde programlama eğitimi için gerekli olan algoritma becerisini geliştirmede oldukça etkili bir yöntemdir. PicBricks ile proje geliştirirken MicroBlocks programını kullanarak zorlu projeleri sadece bir kaç kod bloğunu proje sayfamıza sürükleterek oluşturabiliriz. Ayrıca PicoBricks Arduino IDE de c programlama dili, Thonny IDE de MicroPython programlama dilini desteklemektedir. Arduino IDE ve Thonny IDE programlama araçları fiziksel programlama eğitiminde metin tabanlı programlama araçları arasından en yaygın kullanımı olan programlama araçlarıdır. Thonny IDE MicroPython dilini desteklemesinden dolayı metin tabanlı programlama dillerinde sıklıkla yaşanan noktalama (Syntax) hatalarını ortadan kaldırmaktadır.

Bütün bu özelliklere bakıldığında PicoBricks, fiziksel programlama eğitiminde yaşanan sıkıntıları ortadan kaldırarak fiziksel programlama eğitiminde, eğitimin kalitesini arttırmayı ve yaratıcılığı ön plana çıkartmayı hedeflemektedir.

■ PicoBricks ile Beraber Kullanılması Önerilen Ekipmanlar

- **Dişi-Erkek Jumper Kablo:** Sensörlerin Raspberry Pi Pico bağlantısını sağlar.



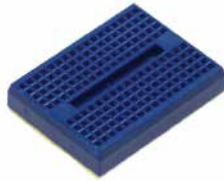
- **Servo Motor:** Açısal hareket gerektiren mekanik projelerde hareketi sağlamak için kullanılır.



- **Sarı DC Motor:** Dönme hareketi gerektiren mekanik projelerde dönme hareketini sağlar.



- **Mini Breadboard:** Birden fazla bağlantı gerektiren projelerde ara bağlantı elemanı olarak kullanılır.



- **HC-SR04 Ultrasonik Mesafe Sensörü:** Mesafe ölçümü gerektiren projelerde mesafeyi algılamaya yarayan devre elemanıdır.



- **RFID NFC Modülü, Kart ve Anahtarlık Kiti:** RFID ve NFC teknolojisi kullanarak devre elemanlarının kontrolünü sağlar.



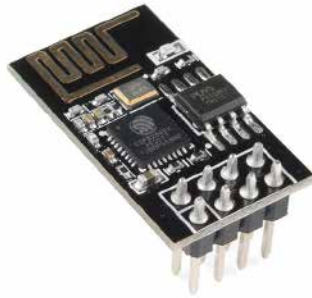
- **Bluetooth Modülü:** Bluetooth teknolojisini kullanarak devre ile bluetooth teknolojisi kullanan başka bir cihazın iletişime geçmesini sağlayan devre elemanıdır.



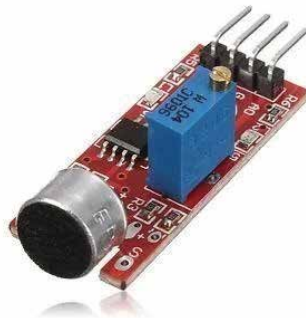
- **Toprak Nem Sensörü:** Topraktaki nem miktarını ölçmeyi sağlayan devre elemanıdır.



- **ESP8266 Wi-Fi Sensörü:** Kablosuz bağlantı gerektiren projelerde kurulan devrenin wi-fi ile başka cihazlar ile iletişime geçmesini sağlar.



- **Ses Sensörü:** Ortamdaki ses seviyesini algılayan devre elemanıdır.



- **PIR Sensörü:** Ortamdaki hareketi algılamayı sağlayan devre elemanıdır.



Ders içeriğinde kullanılacak bazı materyaller;

- PicoBricks kutusu
- Elişi kağıdı
- Atık karton kutu
- Makas
- Koli bandı
- Renkli kalem vb.

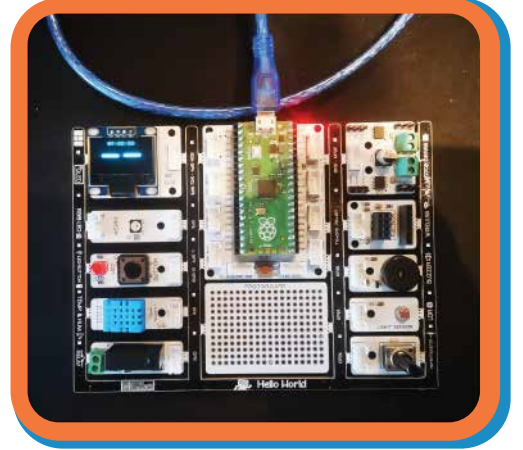
PICOBLOCKS'İ TANIYALIM ETKİNLİKLERİ

Bu bölümde,

- Öğrenci PicoBricks üzerindeki modülleri tanır,
- PicoBricks üzerindeki modüllerin çalışma mantığını kavrar,
- PicoBricks üzerindeki girdi-çıkış modüllerini ayırt eder,
- Bilgisayarlar ile mikroişlemci arasındaki ilişkiyi kavrar.

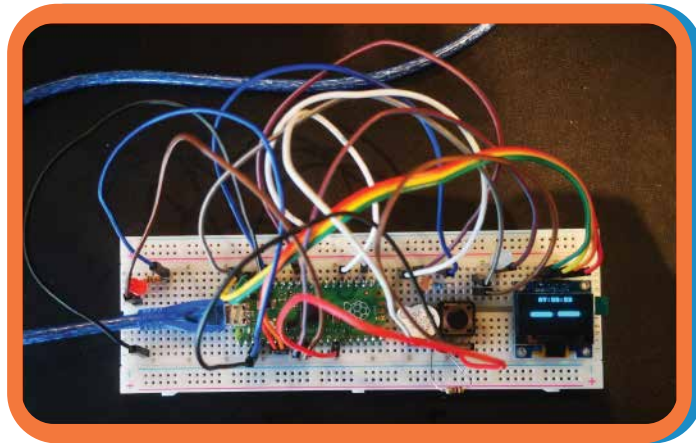
Giriş:

Günümüzde kodlama eğitimi ilkokuldan üniversite seviyesindeki öğrencilere kadar her seviyede öğrenciye hitap edecek şekilde hazırlanmaktadır. PicoBricks, modüler yapısı ve kodlama platformları ile beraber sunduğu esneklik sayesinde her seviye için robotik kodlama eğitimi veren ortamlarda rahatlıkla kullanılacak bir robotik kodlama aracıdır.



PICOBLOCKS'İN ORTADAN KALDIRDIĞI PROBLEMLER

Kablo Karmaşası Yok!

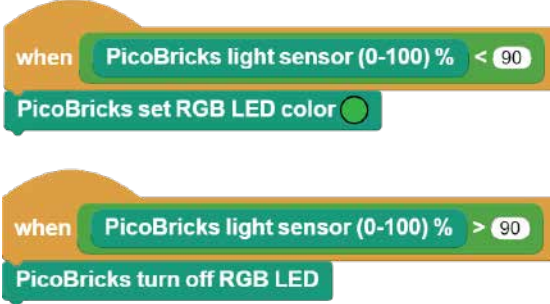


PicoBricks ile kablo karmaşası olmadan devrelerinizi kurabilirsiniz. Benzer şekilde Microbloks programı ile kodlama yaparak kod karmaşasından da kurtulabilirsiniz. Bu özellik ile robotik kodlama eğitimi dersine başlarken karşılaşılabileceğimiz birçok problemi ortadan kaldırmış oluruz. Robotik kodlama eğitimine giriş yaptıktan sonra PicoBricks üzerindeki modülleri gerekli yerlerden kırarak daha fazla proje yapma imkanına sahip oluruz.

Programlama Kolaylığı

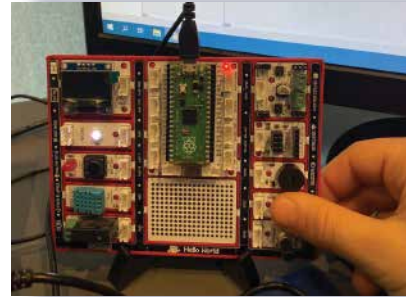
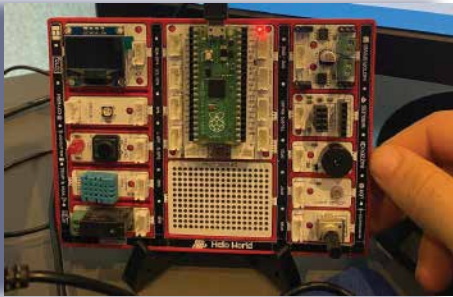
Bir örnek paylaşarak PicoBricks ve MicroBlocks ile kodlamaya başlamanın ne kadar kolay olduğu göstermek istiyoruz. PicoBricks'in karanlıkta otomatik olarak yanmasını sağlayacak bir proje yapacağız. Projemiz için PicoBricks üzerinde bulunan ışık sensörünü ve RGB Led'i kullanacağız. Projeye ait kod bloklarını aşağıda bulabilirsiniz.

Microbloks
ile
yazılmış hali



```
1 from machine import Pin, ADC
2 from picobricks import WS2812
3 define the library
4
5 Idr = ADC(Pin(27))
6 ws = WS2812(6, brightness=0.4)
7 define the input and output pins
8
9
10 define colors
11 RED = (255, 0, 0)
12 GREEN = (0, 255, 0)
13 BLUE = (0, 0, 255)
14
15 COLORS = (RED, GREEN, BLUE)
16 when solar code
17
18
19 while True:
20     while loop
21     print(Idr.read_u16()) # print the value of the IDR sensor to the screen.
22     if (Idr.read_u16() > 100000): # let's check the Idr sensor
23         for color in COLORS:
24             # turn on the LED
25             ws.pixels_fill(color)
26             ws.pixels_show()
27
28         # else:
29             ws.pixels_fill([0,0,0]) # turn off the RGB
30             ws.pixels_show()
```

Thonny IDE
ile
Micropython
dilinde yazılmış hali



Bu sayede kodlamaya yeni başlayan kullanıcılara MicroBlocks ile hızlı bir başlangıç yaptırdıktan sonra metin tabanlı Thonny IDE platformu ile kodlama dillerinin mantığını öğretebiliriz.

Ders içeriğinde kullanılacak materyaller;

- PicoBricks kutusu
- Elişi kağıdı
- Atık karton kutu
- Makas
- Koli bandı
- Renkli kalem vb.

Öğretmene Tavsiye

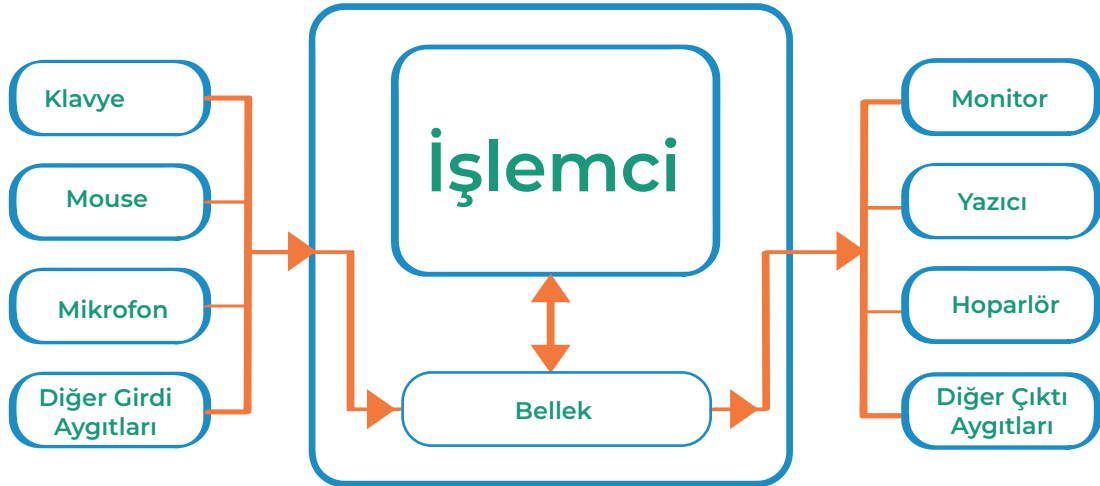
Ders planı hakkında öğrencilere aşağıdaki bilgiler kısa bir şekilde verilmelidir;

- PicoBricks aktivitelerini tanıyalım
- Bilgisayarsız bilgisayar bilimi aktiviteleri
- PicoBricks kodlama platformlarını tanıyalım
- PicoBricks ile proje geliştirmeye başlayalım

Bilgisayar Nedir?

Bilgisayarı oluşturan 4 ana bileşen vardır . Bu bileşenler;

- İşlemci
- Depolama birimleri
- Giriş Aygıtları
- Çıkış Aygıtları



1. **İşlemci:** Anakart üzerinden bulunan küçük çiplere işlemci denir. İşlemciyi bilgisayarın beyni olarak düşünebiliriz. İşlemci ne kadar hızlı ise bilgisayarınızda o kadar hızlıdır.

2. **Depolama Birimleri:** Bilgisayarlar depolama birimlerini kullanarak bilgileri saklar. Depolama birimleri kısa süreli ve uzun süreli olmak üzere ikiye ayrılır.

a) **Kısa Süreli Bellek:** RAM (Rastgele Erişimli Hafıza) bilgisayara iletilen bilgilerin geçici olarak saklandığı yerdir.

b) **Uzun Süreli Bellek:** LTM (Long-Term Hafıza) Bilgilerin silinmediği kalıcı olarak depo edildiği hafıza birimidir.

3. **Girdi Aygıtları:** Bilgisayarın dışarıdan bilgi almasını sağlayan aygıtlardır. Örneğin kulaklarımız vücudun girdi organlarından biridir işittiğimiz sesleri beynimize yollarlar. Bilgisayarın girdi elemanlarına ise klavye, mouse, mikrofon dokunmatik ekran vb. aygıtlar dışarıdan aldığı bilgileri bilgisayarın işlemcisine yollamızı sağlar.

4. **Çıktı Aygıtları:** Bilgisayarın bilgiyi ilettiği aygıtlardır. Örneğin ağızımız çıktı organlarımızdandır konuşarak ses yardımıyla dışarıya bilgi aktarır. Bilgisayarın çıktı aygıtları Monitör,hoparlör vb. aygıtlardır. Monitör görsel çıktıları ,hoparlör ise işitsel çıktıları oluşturur.

Mikrodenetleyici Nedir?

İçerisinde minik bir bilgisayar bulunduran programlanabilir kartlara mikrodenetleyici denir. Mikrodenetleyiciler ile beraber girdi ve çıktı sensörleri kullanılarak projeler geliştirilebilir.

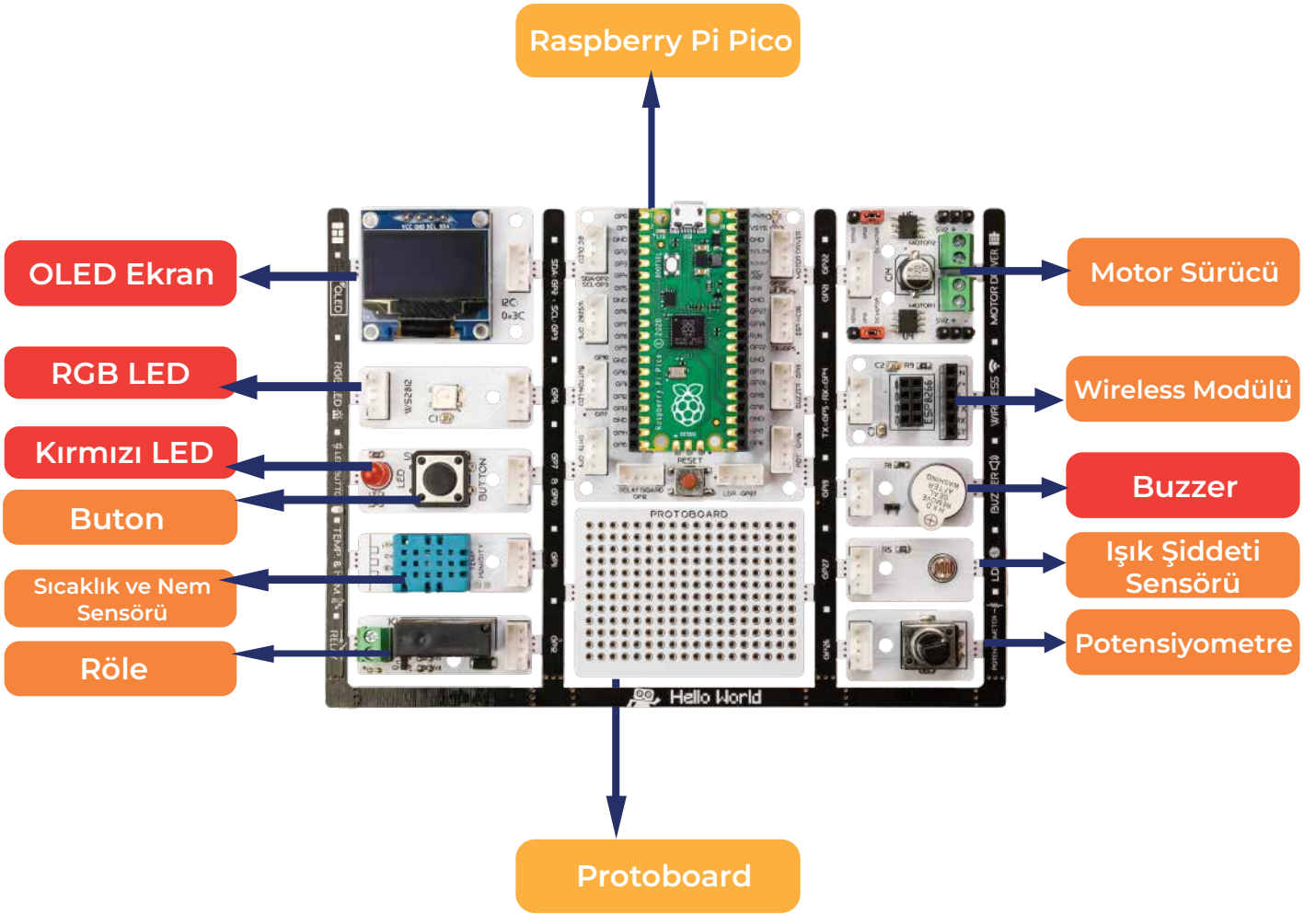
Raspberry Pi Pico Nedir?

Raspberry Pi Pico, PicoBricks üzerinde kullanılan bir mikrodenetleyici karttır.

Raspberry Pi'nin küçük bir bilgisayar olduğunu biliyor muydunuz?

Raspberry Pi , birleşik krallıkta Raspberry Pi vakfı tarafından geliştirilen ve 2012 yılında kullanıma sunulan tek kart bilgisayardır (Single Board Computer). Raspberry Pi Pico ise aynı şirket tarafından daha küçük boyutlarda üretilen bir mikrodenetleyici karttır.

Hadi PicoBricks'i İnceleyelim

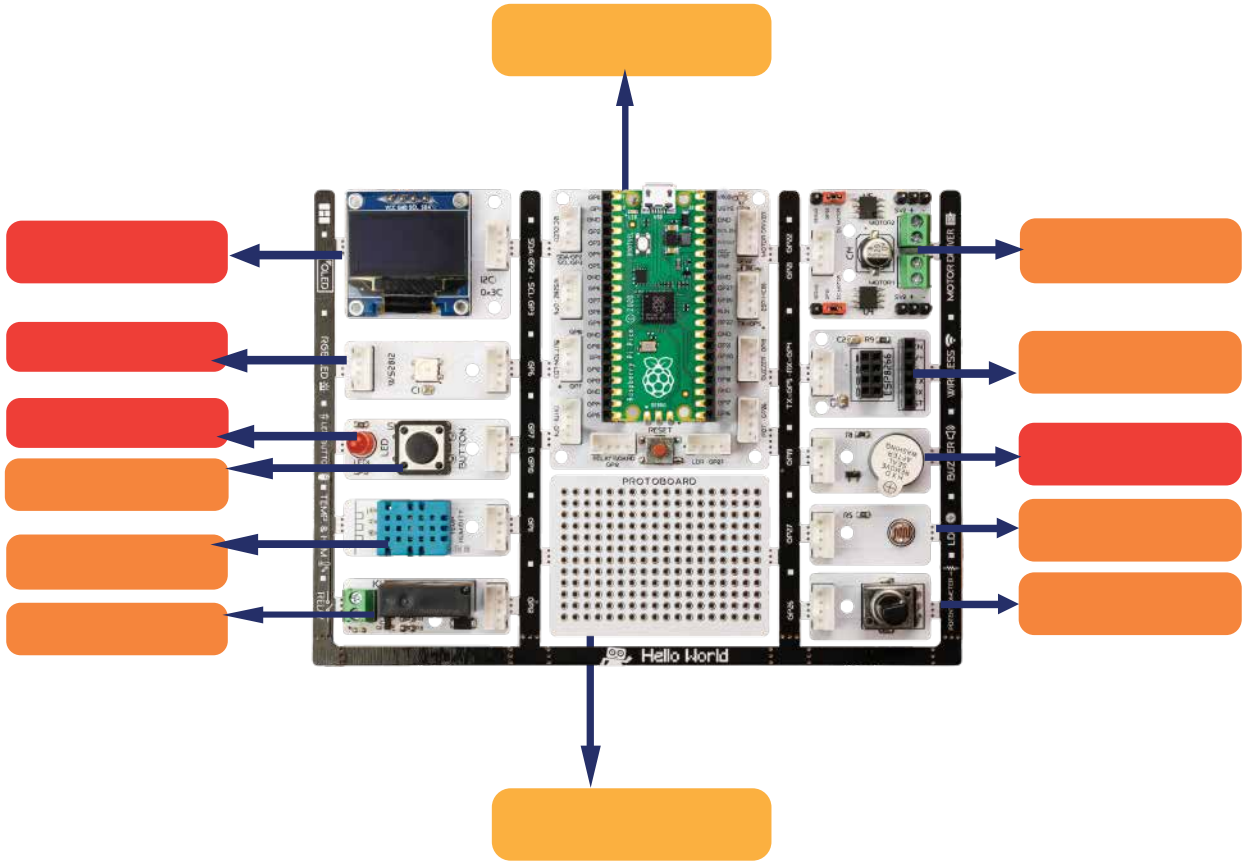


Öğretmene Öneri

Öğrenciler ile beraber aşağıdaki sorulara cevap verilerek PicoBricks tanıtılabilir. Etkinlikler ile bu bölümde öğrenilen bilgiler pekiştirilecektir.

- PicoBricks üzerinde kaç adet sensör vardır?
- PicoBricks üzerindeki sensörlerden hangileri girdi sensörüdür?
- PicoBricks üzerinde hangi sensör/sensörlerden görsel çıktı alınır?
- PicoBricks üzerindeki mikro denetleyici kart nerededir?
- PicoBricks üzerindeki protoboard ne işe yarar?

Etkinlik 1-PicoBricks Modüllerini Tanıyalım



Etkinliğe başlamadan önce öğretmen PicoBricks üzerindeki sensörleri öğrencilere tanıtır. Bununla beraber girdi-çıkıttı aygıtlarından bahseder.

OLED Ekran: Metinsel, görsel veya animasyon türünde çıktılar alabildiğimiz modüldür.

RGB (Red-Green-Blue) LED: Kırmızı-Yeşil- Mavi renklerini belirli oranlard karıştırılarak birçok renk çıktısı alabileceğimiz devre elemanıdır.

LED & Buton: Kırmızı LED modülü yalnızca kırmızı renkte ışık çıktısı almamızı sağlayan devre elemanıdır. Buton ise üzerine basılıp basılmadığını anladığımız bir giriş elemanıdır.

DHT Isı ve Nem Sensörü: Bulunduğu ortamın ısını ve nemini ölçen modüldür.

Röle: Üzerinden akım geçtiği zaman çalışan devre elemanıdır. Bunu lambanızı çalıştıran bir anahtar gibi düşünebilirsiniz.

Raspberry Pi Pico: PicoBricks kartımız ile geliştirdiğimiz projelerde sensörler ile iletişim sağlayan mikrodenetleyici karttır. Ayrıca üzerinde bulundurduğu pin bağlantı noktaları ile devreye başka sensörleri bağlamamızı sağlar.

Protoboard: PicoBricks kartı üzerinde bulunan üzerinde bulundurduğu iletken yollar ve jumper kablolar aracılığıyla kart üzerinde bulunmayan herhangi bir sensör veya iletken bir maddenin kart ile iletişime geçmesini sağlayan modüldür.

Motor Sürücü: Motorlar vb. devre elemanlarının hızını ve frekansını ayarlayan devre elemanıdır.

Wireless Modül: Ortamda bulunan WiFi ağına bağlanarak internete çıkışı sağlayan modüldür. Bu modül sayesinde projelerimizle uzaktan iletişim kurabiliriz.

Buzzer: Ses elde etmeyi sağlayan modüldür.

Işık Şiddeti Sensörü(LDR): Üzerine düşen ışık miktarına göre direnç değerini değiştiren ve bu sayede ortamın ışık miktarını algılayan modüldür. Işık sensörü olarak da adlandırılır.

Potansiyometre: Düğmeyi çevirdikçe değeri değişebilen dirençtir.



Sensörlerin ilk kim tarafından ve hangi amaçla kullanıldığını biliyor musunuz?

Sensörler ilk olarak, Steinel tarafından 1987 yılında aydınlatma amacıyla kullanılmıştır.

↓ Girdi elemanları: Bağlandığı devrede dışardan aldığı değerleri bilgisayara aktaran elemanlara girdi elemanları denir. Örnek vermek gerekirse DHT ısı ve nem sensörü bir girdi elemanıdır. Çünkü dışarıdan aldığı ısı ve nem değerlerini bilgisayar ortamına taşır.

↑ Çıktı elemanları: Devre yardımıyla dışarıdan aldığı bilgilerin görüntülü veya sesli bir şekilde çıktısını verir. Örnek vermek gerekirse OLED ekran bir çıktı elemanıdır. Dışarıdan farklı sensörler yardımıyla aldığı bilgilerin görüntülü bir şekilde çıktısını verir.

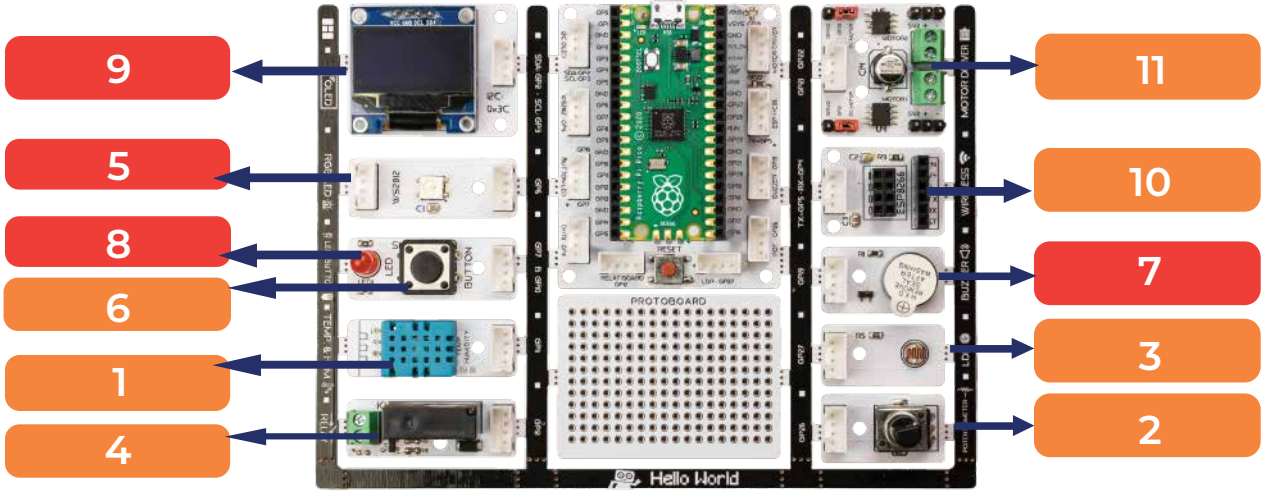
Yukarıdaki bilgiler öğrencilere aktarıldıktan sonra etkinlik öğrencilere tanıtılır.

Yukarıda verilen etkinlik tasarımı sınıf tahtası üzerine çizilebilir, projeksiyon makinesiyle yansıtılabilir, çıktısı alınıp her öğrenciye dağıtılabilir ya da her öğrencinin rahat bir şekilde görebileceği bir noktaya konumlandırılabilir. Öğretmen öğrencilere modüller ile ilgili aşağıdaki soruları ileterek öğrencilerden sorunun cevabı olan modülü doğru kutucuğa yazmalarını ister. Bu sorular ders esnasında teker teker okunur ve öğrencilerin doğru kutucuğa cevabı yerleştirmesi beklenir.

Örnek Sorular:

- PicoBricks üzerinde metin, resim vb. görüntü çıktısı elde etmemizi sağlayan devre elemanı hangisidir ya da adını yazın? PicoBricks üzerindeki ilgili kutucuğa yazınız. (OLED Screen)
- Kırmızı, yeşil ve mavi renklerini belirli değerlerle karıştırarak renk elde temeyi sağlayan çıktı elemanı hangisidir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (RGB LED)
- Sadece kırmızı renkte ışık çıktısı almamızı sağlayan devre elemanı hangisidir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (Kırmızı LED)
- Üzerine baskı uygulayarak bir sürecin başlamasını ve bitmesini sağlayan devre elemanı nedir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (Buton)
- Ortamdaki ısı ve nem miktarını ölçmeyi sağlayan devre elemanı nedir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (DHT Sensörü)
- Devreden geçen akımı kesmeyi sağlayan ya da akımın geçmesine izin veren devre elemanı nedir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (Role Modülü)
- PicoBricks üzerindeki modüllerin kodlanmasına olanak sağlayan üzerinde bulundurduğu pin noktalarıyla PicoBricks üzerinde bulunmayan farklı modüllerinde bağlantısına olanak sağlayan modül nedir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (Mikroişlemci Modülü)
- Devreye bağlanan DC motorların hız ve frekans kontrolünü yapan devre elemanı nedir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (Motor Sürücü)
- PicoBricks ile hazırlanan bir projenin kablosuz olarak kontrolüne olanak sağlayan devre elemanı nedir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (ESP Modülü)
- PicoBricks üzerinde sesli çıktı oluşturan devre elemanı nedir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (Buzzer)
- Ortamdaki ışık miktarını algılayan ve bunu dijital bir değere dönüştüren devre elemanı nedir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (LDR sensörü)
- Dışarıdan yapılan fiziki müdahaleler ile devreye uygulanan direnç değerini değiştiren devre elemanı nedir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (Potensiyometre)
- PicoBricks üzerinde bulundurduğu iletken noktalar sayesinde PicoBricks üzerine farklı sensör veya iletken malzemelerin bağlanmasını sağlayan modül nedir? PicoBricks üzerindeki ilgili kutucuğa yazınız. (protoboard)

Bu etkinliğin hemen ardından girdi ve çıktı modülleri daha iyi anlaşılması için aynı etkinlik tasarımı kullanılarak modülleri ilgilendiren kutucuklar numaralandırılır ve etkinlik tasarımı her öğrencinin görebileceği bir noktaya konumlandırılır.



Daha sonra yine her öğrencinin görebileceği bir noktaya aşağıdaki etkinlik tasarımı konumlandırılır.

Girdi ve Çıktı Modüllerini Tanıyalım



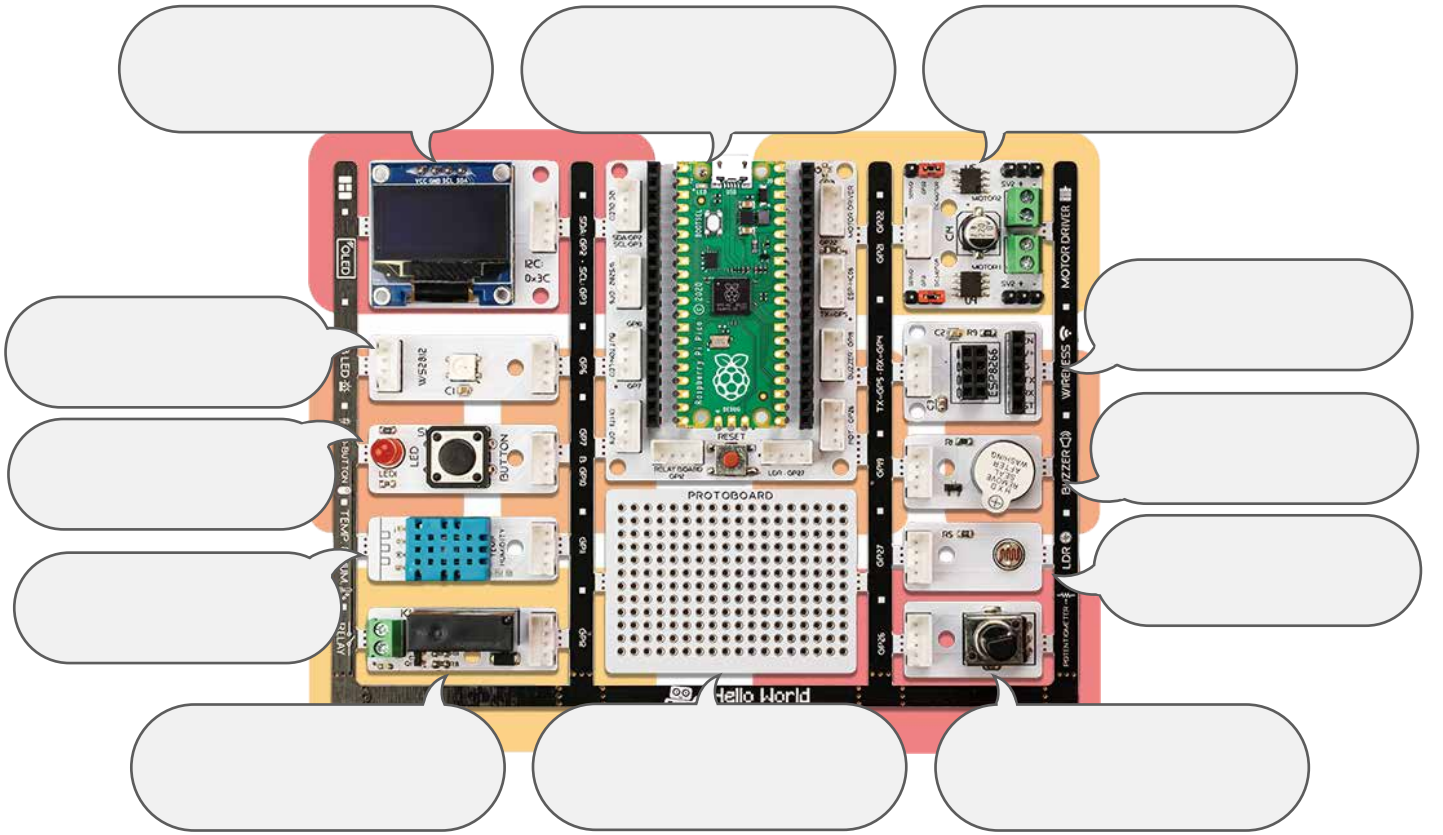
(ipucu: Rakamları farklı en büyük tek sayı)

- Kırmızı ışık çıktısı veren modül
- Metin, resim veya basit animasyon çıktısı veren modül.
- Ses çıktısı veren modül.
- Kırmızı, yeşil, mavi renklerini karıştırarak renkli ışık çıktısı veren modül.

Bu etkinlik tasarımının çıktısı alınarak öğrencilere dağıtılabilir, öğrenciler defterlerine çizilebilir ya da öğretmen tahtada her öğrencinin görebileceği bir noktaya konumlandırılabilir.

- 1) Öğrencilerin 2.etkinlik tasarımında verilen kutucuklara 1.etkinlik tasarımındaki modül numaralarını doğru bir şekilde yerleştirilmesini ister. Bu etkinlik tahta üzerinde soru cevap tekniğiyle yapılabilir aynı zamanda öğrencilerin kendi defterlerine yapmasında beklenebilir.
- 2) Daha sonra 2.etkinlik tasarımında verilen 4 adet boşluğa doğru rakamların doğru sırayla yazılması için öğretmen gerekli ipuçlarını öğrencilere açıklar;
 - Boşlukların altında bulunan 4 ipucunu ve 1. etkinlik tasarımındaki modül numaralarını kullanarak rakamları tespit ediniz.
 - Tespit ettiğiniz rakamlar yan yana getirilince rakamları birbirinden farklı en büyük tek sayıyı elde eder.

Etkinlik-2 (Bilgisayarsız Activite)



Modülleri Tanıyalım - PicoBricks Konuşma Balonları Etkinliği

Bu etkinliğe öğretmen aşağıdaki resimde olduğu gibi PicoBricks kartı üzerinde her modüle ait bir konuşma balonu çizerek başlar. Etkinliğin amacı modülleri ilgilendiren konuşma balonlarına doğru cümlelerin yazılmasıdır. Örnek cümleler aşağıdaki gibi olmalıdır;



Buzzer: Projenizde sesli uyarı vermek istiyorsanız beni kullanmanız lazım.



OLED Screen: Projelerinizde görüntü çıktısı almanızı sağlarım

LED: Projelerinizde sadece kırmızı ışıkla bir çıktı almak istiyorsanız beni kullanmanız gerekmektedir.



Buton: Projelerinizde başlatma ya da bitirme olaylarını benim üzerime baskı yaparak tetikleyebilirsiniz.



RGB LED: Projelerinizde kırmızı, yeşil ve mavi renklerini belirli oranlarda karıştırarak istediğiniz renk çıktısını oluşturmayı sağlarım.



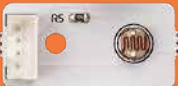
DHT: Projelerinizde bulunduğunuz ortamın ısı ve nem değerini kullanmak istiyorsanız beni kullanmanız gerekmektedir.



Röle Modülü: Projelerinizde elektrik akımını keserek ya da akımın devrede dolaşmasına izin vererek devreyi açıp kapatmanızı sağlarım.



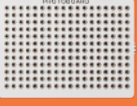
Potansiyometre: Benimle beraber fiziksel bir müdahale ile devre elemanlarına giden akıma farklı büyüklüklerde direçn uygulayabilirsiniz.



LDR Sensör: Projelerinizde ortamdaki ışık miktarını algılamak istiyorsanız beni kullanmanız gerekir.



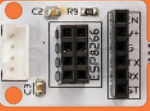
PicoBricks Modülü): Projelerinizi kodlamak için beni kullanırsınız ayrıca kullandığınız modüllerin ve sensörlerin pin bağlantılarını benim üzerimdeki pinlerden gerçekleştirirsiniz. Bi benzetme yapacak olursak PicoBricksin beyni benim.



Protoboard: Üzerimde bulunan iletken yollar ile devreye herhangi bir iletken maddeyi bağlayıp kod içerisinde bu maddeye değer atama işlemini benim sayemde yaparsınız.



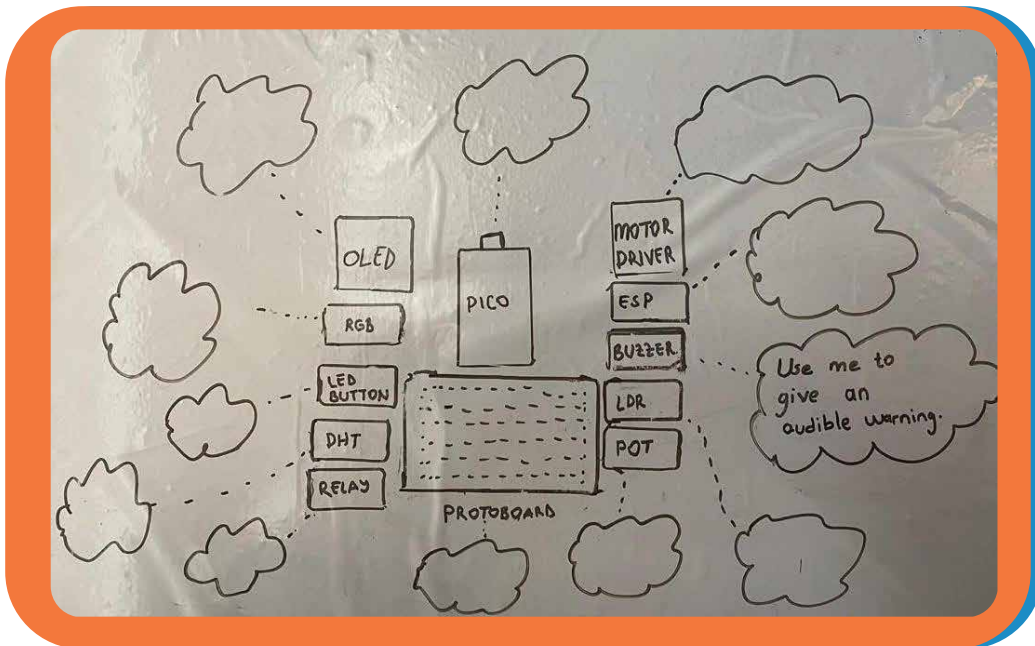
Motor Sürücü: Projelerinizde DC motor veya servo motor kullanmak istiyorsanız servo motor ve dc motorun hızını, frekansını ayarlamak için devrede benide kullanmanız gerekir.



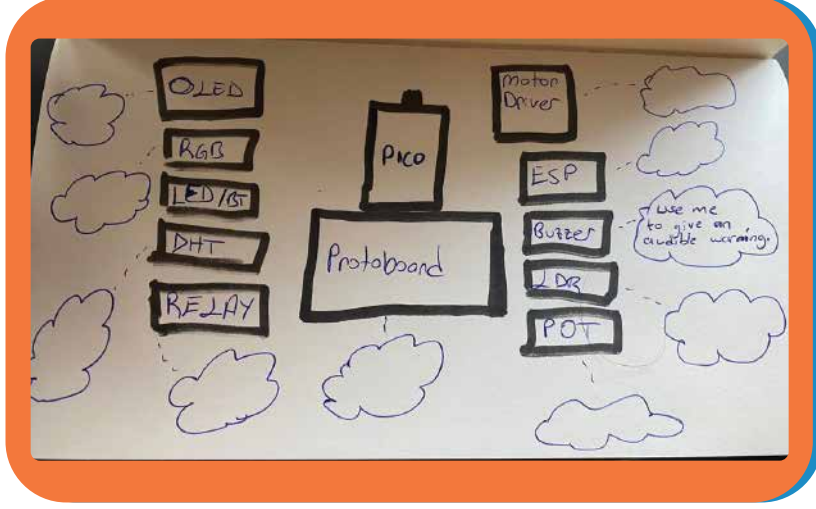
Wireless Modül: Projelerinizde bluetooth veya wi-fi ile haberleşme sağlama istiyorsanız beni kullanmanız gerekir.

Etkinlik öğretmen tarafından farklı şekillerde sınıfa uygulanabilir bu uygulamalardan bir kaçını aşağıda belirtilmiştir.

- Örnek etkinlik tasarımı sınıf tahtası üzerine çizilir veya ilgili tasarımın çıktısı alınarak tahta üzerine yapıştırılabilir. Hazırlanan örnek cümleler sırayla okunarak okunan cümlenin doğru konuşma baloncuğuna yazılması için öğrenciler tahtaya kaldırılır. Her öğrencinin tahtaya kaldırılmasına dikkat edilmelidir.



- Örnek etkinlik tasarımını her öğrenci kendi defterine çizer (Çok detaylı bir çizim beklenmemesi gerekir.) ya da örnek tasarımın çıktısı alınarak her öğrenciye dağıtılır. Sonrasında öğretmen sınıf içerisinde herkesin duyacağı şekilde hazırladığı örnek cümleleri sırayla okuyarak öğrencilerin okunan cümleleri doğru modülün üzerindeki konuşma baloncuğuna yazmasını bekler.



Bu etkinlikteki temel hedef öğrencinin PicoBricksi tanımasıdır. Öğrenci picobrick üzerindeki modülleri ve PicoBricksin bu modüller ile olan iletişimini kavrar ve ileride yapacağı projeler için fikir sahibi olur.

Etkinlik Kazanımları:

- Öğrenci PicoBricksi tanır.
- Öğrenci PicoBricks üzerindeki modüllerin çalışma mantığını kavrar.
- Öğrenci sensör teknolojilerini tanır.

Öğretmene Öneri

Etkinlik sonunda öğrenilen bilgilerin öğrenilip öğrenilmediğini anlamak ve dersin özetini yapmak için aşağıdaki sorular öğrencilere sorulabilir.

Değerlendirme Soruları:

PicoBricks üzerindeki modülleri kullanarak bir proje yapmak isteseydiniz bu nasıl bir proje olurdu ve hangi modülleri kullanırdınız? Açıklayınız.

PicoBricks üzerindeki girdi elemanları nelerdir? Sıralayınız.

PicoBricks üzerinde ışık çıktısı veren modüller nelerdir? Sıralayınız.

PicoBricks'i Tanıtma ve Test Etme

PicoBricks modüllerinin çalıştığını test etmek ve nasıl çalıştıklarını gözlemlemek için MicroBlocks ile yazılan test kodunu PicoBricks içerisine yükleyelim ve çalıştıralım.



Niçin test kodu yazarız?

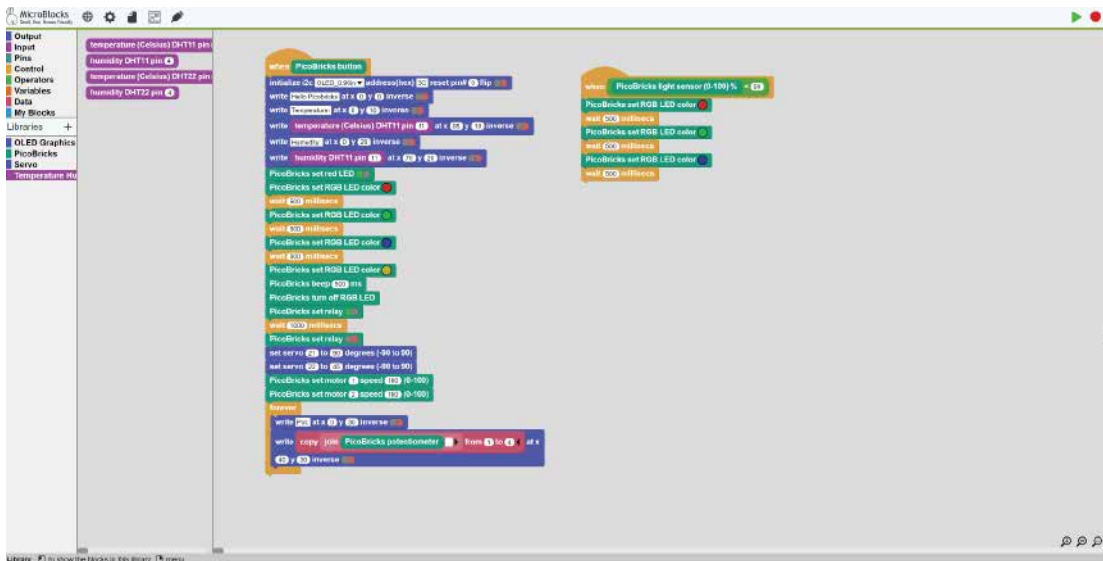
Tasarladığımız projede her aşama bittikten sonra test kodu yazmak ve o aşamayı test etmek gerekebilir. Eğer o aşamada sorun varsa ve test edilmediyse proje tamamlandıktan sonra o aşamaya geri dönmek zaman kaybına yol açacaktır.



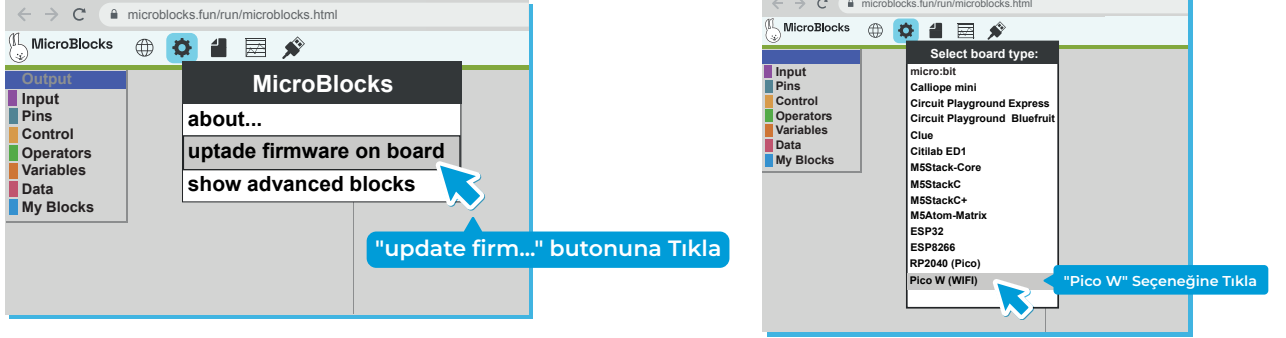
! !
Dijital kitaptaki linkten ya da PicoBricks GitHub sayfasından test kodunu açınız. Test koduna gitmek için tıklayınız.

MicroBlocks ile PicoBricks'e Kod Yükleyelim

1. Test kodunu MicroBlocks online platformunda açınız.

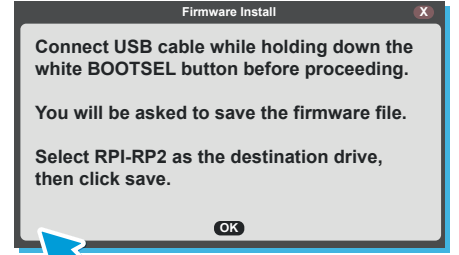


2. Şimdi PicoBricks'i bilgisayara bağlamak için aşağıdaki adımları takip edelim. İlk olarak ayarlar simgesine tıkladıktan sonra açılan pencereden "update firmware on board" seçeneğine tıklayınız. Açılan diğer pencereden ise "RP2042 (Pico)" seçeneğini seçiniz.



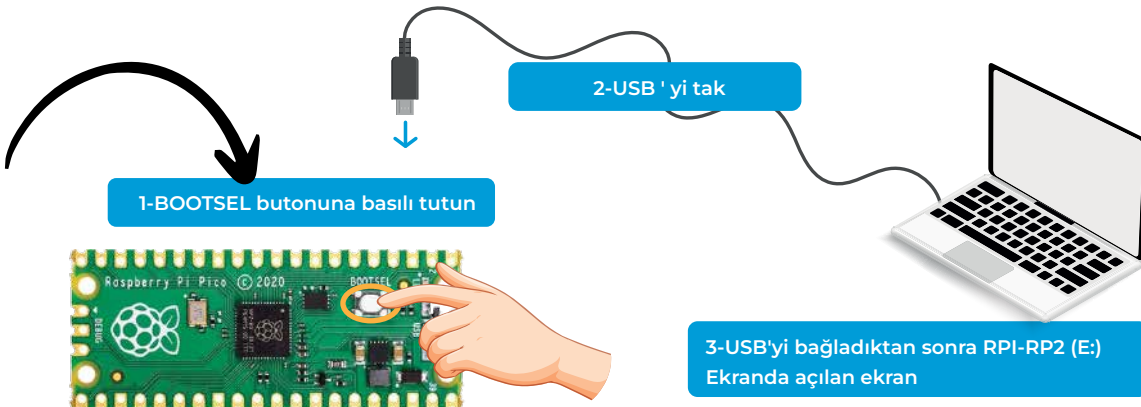
3. Açılan BOOTSEL penceresinde "OK" butonuna basın. Daha sonra açılan kaydetme penceresini kapatmayınız.

Bu pencere karşınıza çıktığında aşağıdaki adımları takip ediniz.
Pencereyi kapatmayın)

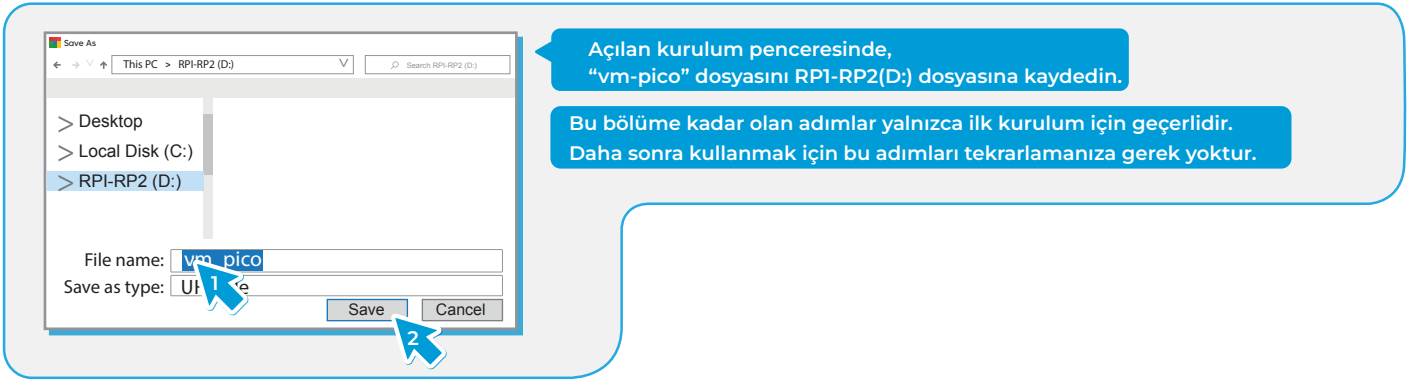


4-Daha sonra "OK" butonuna basın.

4. PicoBricks'i online editöre bağlamak için Raspberry Pi Pico üzerindeki beyaz BOOTSEL düğmesini basılı tutarken kartı USB kablo ile bilgisayarınıza bağlamalısınız.




5. 3.adımda açılan kaydetme penceresinde “vm_pico” dosyasını RPI-RP2 (D:) ‘ nin içine kaydediniz.

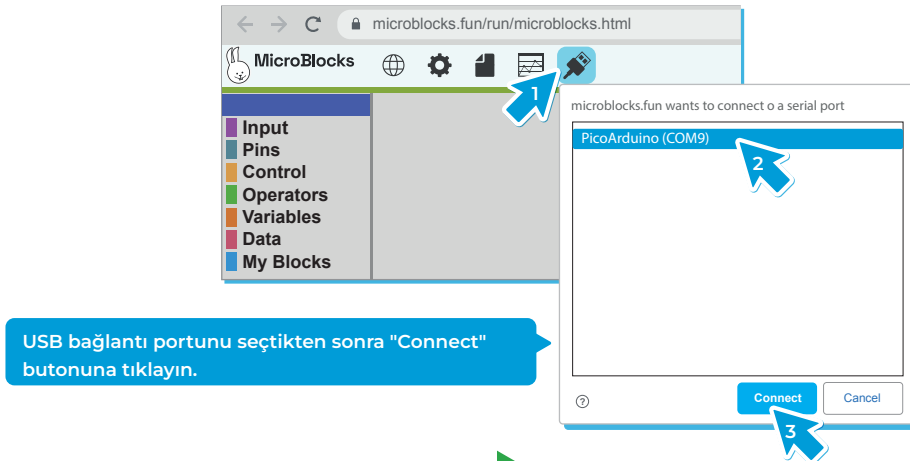


Açılan kurulum penceresinde, “vm_pico” dosyasını RPI-RP2(D:) dosyasına kaydedin.

Bu bölüme kadar olan adımlar yalnızca ilk kurulum için geçerlidir. Daha sonra kullanmak için bu adımları tekrarlamana gerek yoktur.

Eğer ikon yeşilse bağlantı sağlanmıştır.

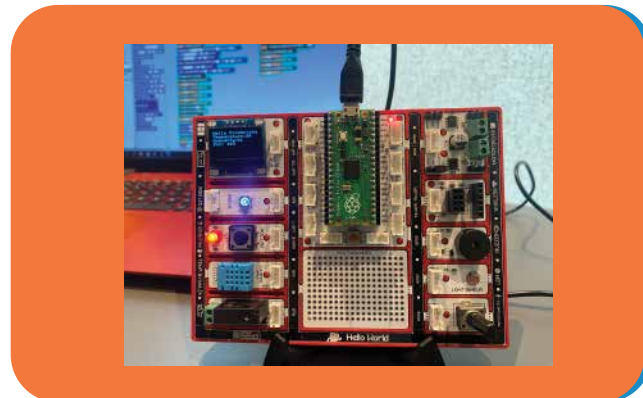
6.  Bağlan butonuna tıkladığında mikro aygıtların takılı olduğu sistem USB bağlantı noktalarını görüntülenir. Bu pencerede önce Pico aygıtını seçip sonra Bağlan butonlarına tıklayarak PicoBricks’i MicroBlocks’a bağlayabilirsiniz. Bağlantı başarılı olduğunda, USB simgesinin arkasında yeşil daire belirecektir.



USB bağlantı portunu seçtikten sonra “Connect” butonuna tıklayın.

7. PicoBricks test kodunu çalıştırmak için  butonuna tıklayabilirsiniz.  için butonuna tıklayınız. Kodu durdurmak için butonuna tıklayınız.

Etkinlik Görseli



Etkinliğin MicroBlocks Kod Blokları

```

when PicoBricks button
  Initialize I2C OLED_0.96in address(hex) 3C reset pin# 0 flip
  write Hello PicoBricks at x 0 y 0 Inverse
  write Temperature at x 0 y 10 Inverse
  write temperature (Celsius) DHT11 pin 11 at x 95 y 10 Inverse
  write Humidity at x 0 y 20 Inverse
  write humidity DHT11 pin 11 at x 70 y 20 Inverse
  PicoBricks set red LED
  PicoBricks set RGB LED color
  wait 500 millisecs
  PicoBricks set RGB LED color
  wait 500 millisecs
  PicoBricks set RGB LED color
  wait 500 millisecs
  PicoBricks set RGB LED color
  PicoBricks beep 500 ms
  PicoBricks turn off RGB LED
  PicoBricks set relay
  wait 1000 millisecs
  PicoBricks set relay
  set servo 21 to 90 degrees (-90 to 90)
  set servo 22 to 45 degrees (-90 to 90)
  PicoBricks set motor 1 speed 100 (0-100)
  PicoBricks set motor 2 speed 100 (0-100)
  forever
  write Acc at x 0 y 30 Inverse
  write copy join PicoBricks potentiometer from 1 to 4 at
  x 40 y 30 Inverse
  
```

```

when PicoBricks light sensor (0-100) % < 90
  PicoBricks set RGB LED color
  wait 500 millisecs
  PicoBricks set RGB LED color
  wait 500 millisecs
  PicoBricks set RGB LED color
  wait 500 millisecs
  
```

Etkinlik Detayları

PicoBricks test kodu ile PicoBricks üzerindeki Protoboard Modülü hariç tüm modülleri test edebiliriz.

Butona bastığımızda Oled ekran üzerinde "Hello Picobricks" yazar.

DHT ısı ve nem modülünü kullanarak ısı ve nem değerini yazdırıyoruz.

Kırmızı led modülü ve RGB led modülünü çalıştırıp renk çıktıları elde ediyoruz.

Röleyi açıp 1 saniye bekledikten sonra kapatıyoruz rölenin açılıp kapandığını röle modülü üzerindeki küçük led ışıktan ve açılıp kapanma sesinden anlayabilirsiniz.

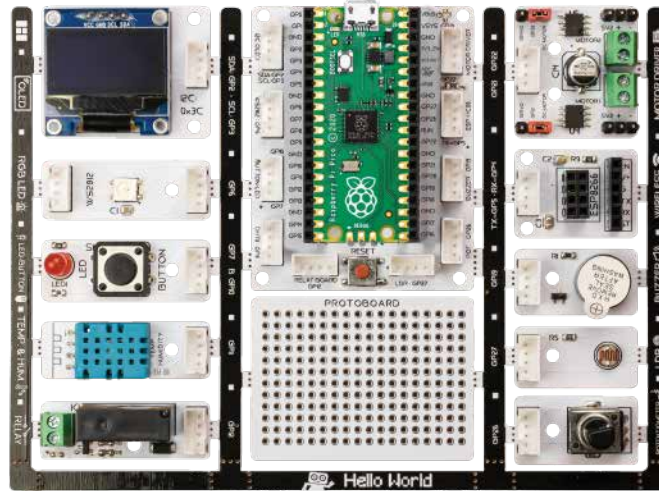
Motor sürücü modülü üzerine takılabilecek servo ve dc motor noktalarını aktif hale getiriyoruz motor sürücü pin noktalarına dc motor ve servo motor takarsak çalıştığını gözlemleyebiliriz.

Ayrıca picobricks ESP modülü üzerinde bluetooth sensörü için verilen pin noktasına bluetooth sensörü takılarak sensör üzerindeki ışığın yanıp yanmadığı test edilebilir eğer yanıyor ise modül çalışıyor demektir.

Buzzerdan ses çıktısı alıyoruz.

LDR sensörü modülünün üzerini elimizle kapatarak RGB ledi çalıştırıyoruz.

Potansiyometrenin ayarıyla oynayarak potansiyometre değerindeki değişmeyi oled ekran üzerinde görebiliriz.



Öğretmene Öneri

1. Yukarıdaki açıklamalardan yararlanarak test kodunun nasıl çalıştığını çalışma sırasını öğrencilere tanıtınız.
2. Test Kodunu her öğrenciyle paylaştıktan sonra öğrencilerle beraber kodu çalıştırınız.
3. Picobricks üzerindeki modülleri kod çalışırken anlatınız ve çalışma mantıkları hakkında ilk derste öğrenilen bilgileri tekrar ediniz.
4. Motor sürücünün pin noktalarına ilk önce sarı (DC) Motor takarak çalıştığını kontrol ediniz.
5. Motor sürücünün pin noktalarına servo motor takarak servo motorun çalıştığını gözlemleyiniz.
6. Servo motor ve Sarı (DC) motor çalışma mantığı arasındaki farka dikkat çekerek öğrencilere bu farkı anlatınız.
7. Potansiyometre ile oynayarak oled ekran üzerindeki potansiyometre değerinin değişimini gözlemleyiniz.
8. LDR sensörü üzerini parmağınızla kapatarak RGB ledin çalıştığını gösteriniz ve LDR sensörünün çalışma mantığını anlatınız.

Ders Hedefleri:

- Öğrencinin, PicoBricks'i tanınması hedeflenmektedir.
- Öğrencinin PicoBricks ile proje geliştirme hakkında bilgi sahibi olması hedeflenmektedir.
- Öğrencinin bilgisayar ile mikroişlemci arasındaki ilişkiyi kavraması hedeflenmektedir.
- Öğrencinin tasarım yaparken el-kol koordinasyonunun gelişmesi hedeflenmektedir.
- Öğrencinin bilgi işlemsel düşünme becerisi basamakları olan;
 - Problem çözme becerisi,
 - Algoritmik düşünme becerisi,
 - Ayırıştırma becerisi,
 - Soyutlama becerisi,
 - Modelleme Becerisi,
 - Değerlendirme becerisi, kazanması hedeflenmektedir.

K-12 Uluslararası Bilgisayar Bilimi Standartları

• 1B-AP-08 - 3-5 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.

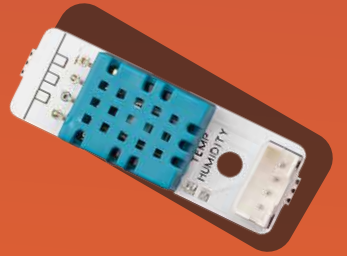
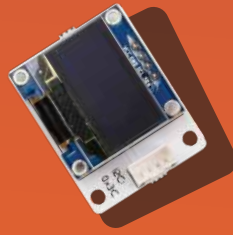
• 1B-DA-07 - 3-5 Organize and present collected data visually to highlight relationships and support a claim.

• 1B-DA-07 - 3-5 Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea

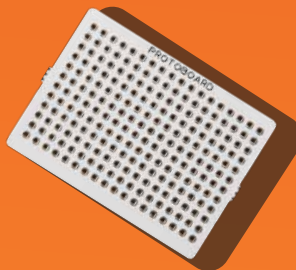
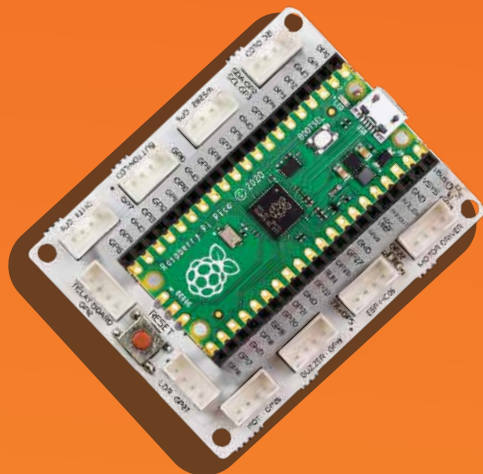
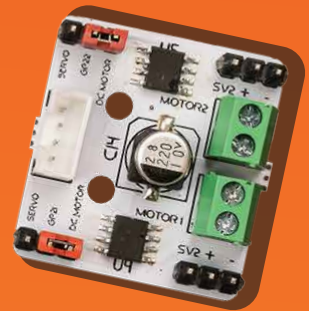
• 1B-AP-08 - 3-5 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.

• 1B-AP-11 - 3-5 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

• 1B-AP-13 - 3-5 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.



PART 2



Bu Bölümde Neler Öğreneceğiz

Bu bölümde öğrenci;

- MicroBlocks blok yapısını öğrenir.
- PicoBricks Kütüphanesini MicroBlocks IDE proje sayfasına yükleyebilir.
- Proje sayfasına kod yükleyebilir.
- Proje sayfasından kod silebilir.
- PicoBricks kütüphanesini kullanarak, PicoBricks modüllerini çalıştırabilir.

MicroBlocks Bloklarını İnceleyelim

Çıktı Blokları: Sesli ve görüntülü çıktı aygıtlarını yönetebildiğimiz bloklar bulunur.

Pin Blokları: Picobricks pinlerine sensör bağladığımızda buradaki blokları kullanabiliriz.

İşlem Blokları: Matematiksel işlemlerin bulunduğu kod blokları.

Veri Blokları: Projeye tanımladığımız verileri kullanmamızı sağlayan kod blokları bulunur.



Output
Input
Pins
Control
Operators
Variables
Data
My Blocks
Libraries +
OLED Graphics
PicoBricks
Temperature Hu

Girdi Blokları: Girdi aygıtlarını yönetebildiğimiz kod blokları bulunur.

Kontrol Blokları: Başlama, bitirme, koşul, döngü vb. olayları kontrol edebildiğimiz kod blokları bulunur.

Değişken Blokları: Değişken tanımlama, değişkene değer atama vb. işlemlerin yapıldığı kod blokları bulunur.

Bloklarım: Kendi kod bloklarımızı hazırlayıp görev tanımlayabileceğimiz kod blokları bulunur.

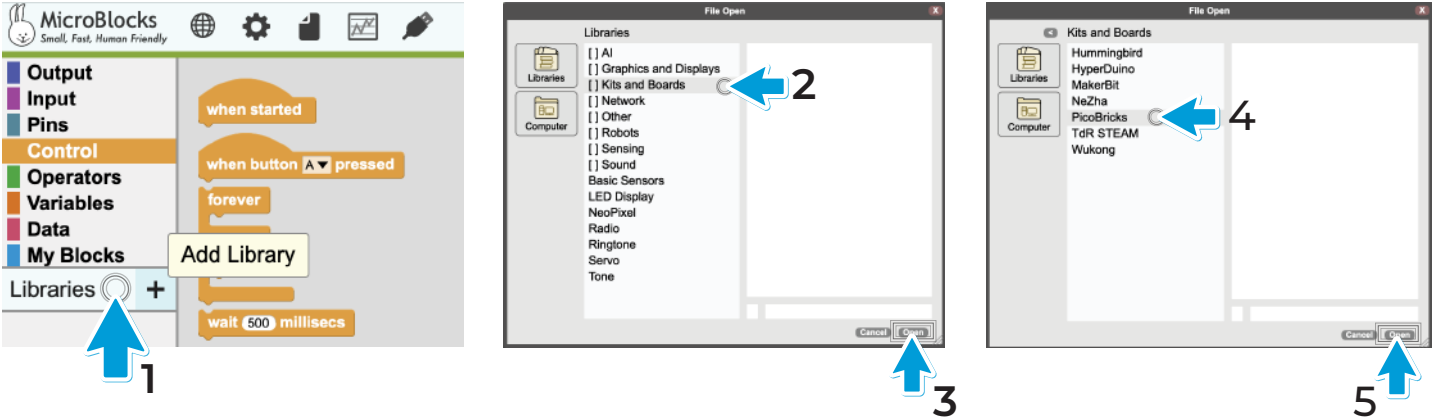
Kütüphaneler: Kütüphaneleri Microblocks IDE editörüne yükledikten sonra içerisindeki kod bloklarına erişebildiğimiz bölüm.

PicoBricks ile Temel İşlemler

MicroBlocks online editörü açıp PicoBricks'i bağlayalım (ikonun yeşil olduğunu görelim).   Ardından PicoBricks kütüphanesini editöre yükleyelim.

PicoBricks Kütüphanesini MicroBlocks IDE'ye Yükleyelim

Aşağıdaki adımları takip ederek PicoBricks kütüphanesini projemize yükleyelim.



Kırmızı LED Modülünü Yakalım ve Söndürelim.

PicoBricks set red LED 

PicoBricks set red LED 

comment

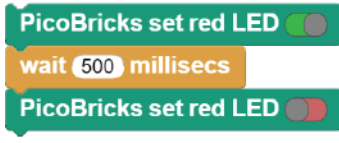
İkinci blokta anahtarı kapatalım. Önce yukarıdaki bloğa tıklayalım daha sonra aşağıdaki bloğa tıklayalım. Üstteki bloğa tıklandığında PicoBricks'in üzerindeki LED'in yandığını, alttaki bloğa tıklandığında LED'in kapandığını göreceksiniz.

Bu iki bloğu birleştirin ve üzerine tıklayın.

PicoBricks üzerindeki kırmızı LED'in yanmadığını gözlemleyeceksiniz. Aslında kırmızı led yanıyor ve hızlı bir şekilde sönyüyor fakat bu işlem çok hızlı gerçekleştiği için LED'in yandığını göremiyoruz

MicroBlocksta sadece çalıştırmak istediğiniz kod bloğu/blokları üzerine tıklayarak ilgili bloğu/blokları çalıştırabilirsiniz.

Bloklar arasında yarım saniye beklemek için “wait” bloğunu kullanalım.



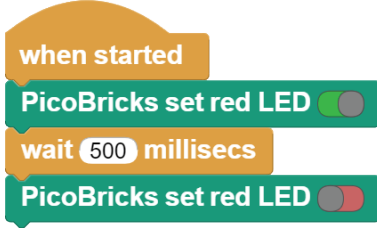
comment

"PicoBricks set red LED" bloklarının arasına control bloklarından "Wait 500 millisecs" bloğunu sürükleyin. Bu üç bloğu birleştirdikten sonra üzerlerine tıklayarak çalıştırınız.

Microblocks editöründe sağ üst köşede bulunan start butonuna tıklayınız. ▶

Start butonuna tıkladığınızda kodun çalışmadığını gözlemleyeceksiniz. Bunun sebebi herhangi bir başlangıç komut bloğu kullanmamamız.

Şimdi “When started” bloğunu kullanarak blokları birbirine bağlayalım ve start butonuna basalım. ▶



comment

Kod bloklarının en üstüne Control bloklarından "when started" bloğunu sürükleyelim. Start butonuna tıklayarak çalıştırın.

PicoBricks üzerindeki kırmızı LED'in bir kere yanıp söndüğünü gözlemleyeceksiniz.

RGB LED Modülünü Kullanalım



comment

Control blokları içerisinde "when started" bloğunu, PicoBricks kütüphanesi içerisinde "PicoBricks set RGB LED color" bloğu ile birleştirip start butonu ile projeyi çalıştırınız.

Az önce RGB LED modülünü yeşil renkte yaktık bu renk PicoBricks kütüphanesinden RGB LED bloğunu sürüklediğimizde varsayılan renk olarak gelir. Bu rengi değiştirmek için yeşil renkli noktanın üzerine tıklayın ve çıkan renk şemasından istediğiniz rengi seçerek programı çalıştırınız.



when started

PicoBricks set RGB LED color

PicoBricks set RGB LED color

PicoBricks color r 0 g 0 b 0 (0-255)

Daha önceden RGB LED'in çalışma mantığını öğrenmiştik şimdi kısa bir şekilde tekrar edelim. RGB LED'ler kırmızı, yeşil ve mavi renklerini belirli bir oranda karıştırarak farklı renkler elde etmeyi sağlar. PicoBricks kütüphanesindeki Picobricks color RGB bloğunu kullanarak renk değerlerini kendimiz ayarlayabiliriz. (Renk değeri Maksimum 250 olabilir.)



RGB LED ile kaç farklı renk elde edebileceğinizi biliyor musunuz?

RGB LED'ler ile $256 \times 256 \times 256 = 16,777,216$ farklı renk oluşturabiliriz.

PicoBricks set RGB LED color PicoBricks color r 0 g 0 b 0 (0-255)

Şimdi renk değerlerini (0-255) kendimiz belirlediğimiz bir rengi RGB LED modülü ile yakalım.

when started

PicoBricks set RGB LED color

PicoBricks color r 250 g 90 b 158 (0-255)

comment

"PicoBricks set RGB LED color" bloğunun sağ köşesinde bulunan renkli noktaya "PicoBricks color r_g_b" bloğunu sürükleyiniz. Daha sonra r (Red), g (Green), b (Blue) değerlerini giriniz."

Şimdi kırmızı LED'i yakalım yarım saniye bekledikten sonra kapatıp RGB LED'i Mavi renk yakalım. Bu şekilde PicoBricks kütüphanesinde bulunan farklı modüllere ait kod bloklarını beraber kullanmış olacağız.

when started

PicoBricks set red LED

wait 500 millisecs

PicoBricks set red LED

PicoBricks set RGB LED color



Şimdi Sıra Sizde!

RGB LED modülünü yarım saniye aralıklarla Kırmızı Yeşil ve Mavi renklerde yanmasını sağlayın.

Buzzer Modülünü Kullanarak Ses Çıkaralım

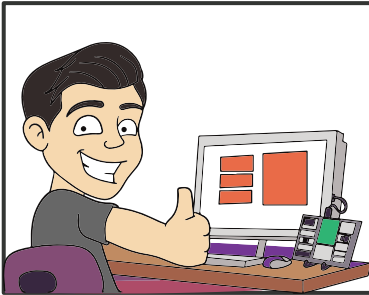
Hadi buzzer ile basit bir uyarı sesi çıkaralım.

when started

PicoBricks beep 500 ms

comment

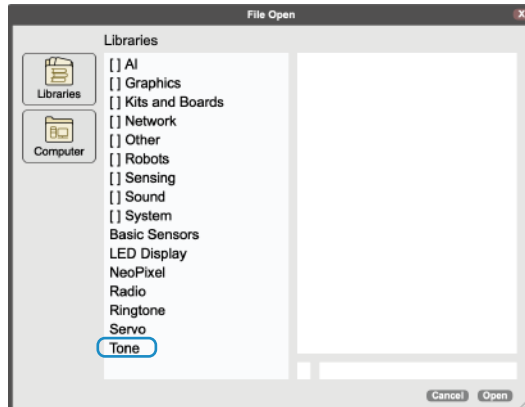
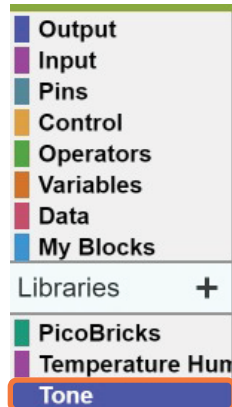
PicoBricks kütüphanesi içerisinde "PicoBricks beep " bloğunu proje sayfamıza sürükleyelim. Ve control bloklarından "when started" bloğuyla birleştirip start butonuna basalım.



Şimdi Sıra Sizde!

PicoBricks üzerindeki kırmızı LED ve Buzzer modülünü kullanarak bir uyarı projesi yapınız. Microblocks Tone kütüphanesini kullanarak buzzer ile farklı tonlarda ses çıkaralım.

Kütüphaneler sekmesinden Tone kütüphanesini MicroBlocks IDE online editörüne yükleyin.



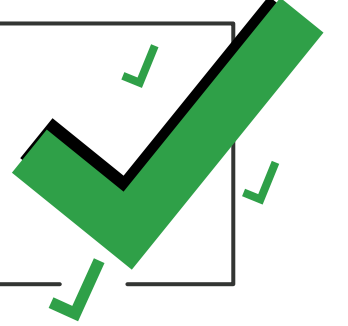
when started

play note C octave 0 for 500 ms

comment

Tone kütüphanesinden "play note" bloğunu "when started" bloğunun altına sürükleyerek start butonuna basalım.

Play Tone kütüphanesinden sürüklediğimiz bloğun içerisindeki değerleri değiştirerek buzzerdan çıkan sesin değişimini gözlemleyiniz.



Tone kütüphanesinden sürüklediğimiz note bloğunun içerisindeki değerleri değiştirerek üç farklı ses tonu üretelim ve aralarına yarım saniye bekle bloğunu kullanarak sesler arasındaki farkı gözlemleyelim.

when started

play note A octave 0 for 500 ms

wait 500 millisecs

play note F octave 0 for 500 ms

wait 500 millisecs

play note C octave 0 for 500 ms

Blokların içerisindeki değerleri istediğiniz gibi değiştirebilirsiniz.

Şimdi tone kütüphanesindeki diğer blokları keşfedelim.

when started

play note C octave 0 for 500 ms

wait 500 millisecs

play midi key 60 for 500 ms

wait 500 millisecs

play frequency 261 for 500 ms

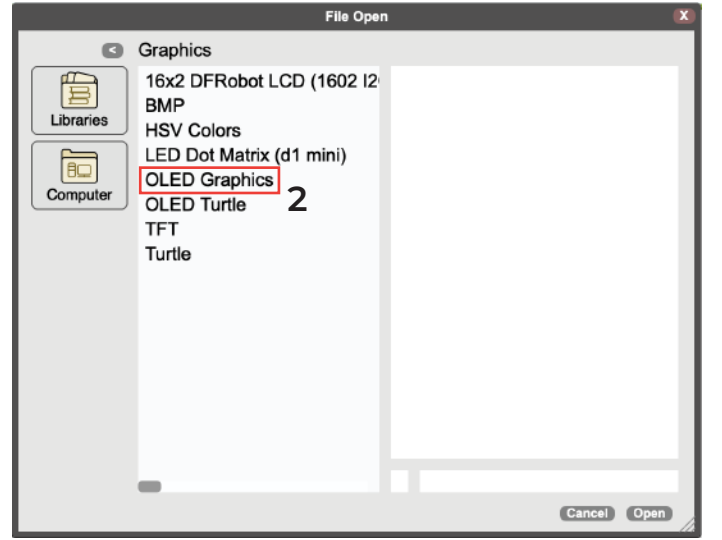
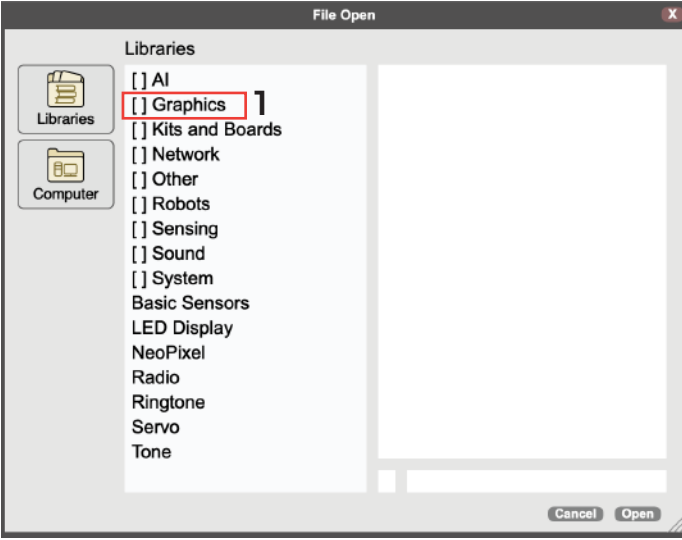
comment

Play tone kütüphanesinde, buzzer ile farklı ses tonlarını çalmak için üç tane blok bulunmaktadır. Bu blokları projemize sürükleyelim ve aralarına yarım saniye bekle bloğunu sürükleyelim. Blokların içerisindeki değerleri değiştirerek farklı ses tonları elde edebilirsiniz.

Buzzer Modülünü Kullanarak Ses Çıkartalım

OLED ekran modülünü kullanarak ekrana yazı yazalım

Ekran üzerinde yazı yazabilmek en güzel çıktı alma yöntemlerinden biridir. Bu bölümde OLED ekranı nasıl kullanacağımızı göreceğiz. MicroBlocks'ta OLED ekranı kullanmak için kütüphanelerden OLED Graphic kütüphanesini MicroBlocks IDE editörüne yüklememiz gerekmektedir. Aşağıdaki adımları takip ederek kütüphaneyi yükleyiniz.



PicoBricks üzerindeki OLED ekranı kullanabilmemiz için aşağıdaki başlangıç bloğunu projemize sürüklememiz lazım bu blok OLED ekranda herhangi bir şey yazdırmadan önce kullanılmalıdır.

initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip

"This is the default address for PicoBricks, we should not change it"

Hadi şimdi OLED ekrana ismimizi yazdıralım.

when started

initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip

write PicoBricks at x 0 y 0 inverse

comment

"When started" bloğundan sonra PicoBricks üzerindeki OLED ekranı çalıştırmamızı sağlayan bloğu sürükleyelim. Daha sonra ekrana yazı yazmak için gerekli olan kod bloğunu sürükleyelim. Bu kod bloğu üzerinden ekranda yazacak yazıyı ve yazının konumunu belirleyebiliriz. PicoBricks üzerindeki OLED ekran 128x64 boyutlarındadır. X ve Y koordinatlarına değer girilerek yazının konumu ayarlanabilir.



Şimdi X ve Y koordinatlarını kullanarak isminizi ekranın ortasına yazdırın.



Buton kullanarak projemizi başlatalım

Bu zamana kadar yaptığımız bütün etkinliklerde MicroBlocks IDE platformundaki start butonunu kullanarak projelerimizi başlattık. Artık PicoBricks üzerinde bulunan butonu kullanarak projelerimizi başlatabiliriz

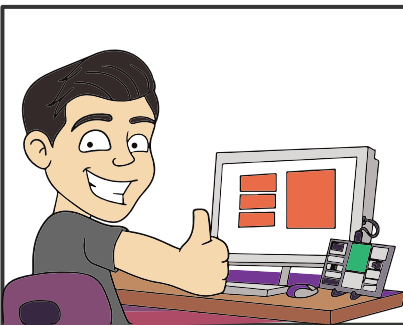
PicoBricks üzerindeki butonu kullanarak projemizi başlatabilmek için;

“Controls” bloklarından  bloğunu proje sayfanıza sürükleyiniz.

PicoBricks kütüphanesinden  bloğunu  bloğunun üzerindeki anahtar işaretinin içerisine sürükleyiniz.

Artık  butonuna bastıktan sonra  bloğu sayesinde PicoBricks üzerindeki butona basarak kod bloğunun altına sürüklediğimiz blokları çalıştırabiliriz.

Şimdi butona bastıktan sonra Kırmızı LED'i ve Buzzeri çalıştıran kod bloklarını proje sayfamıza sürükleyelim.

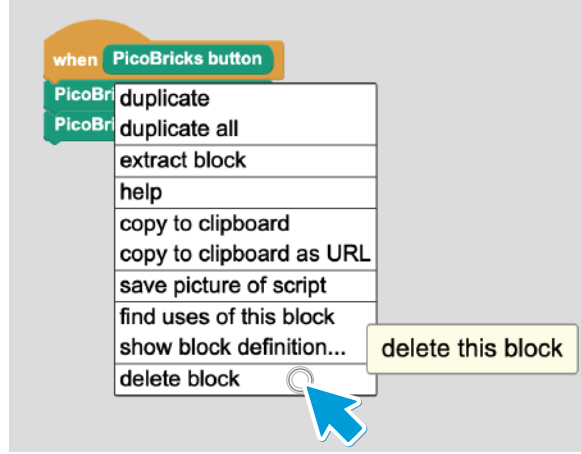


Şimdi Sıra Sizde!

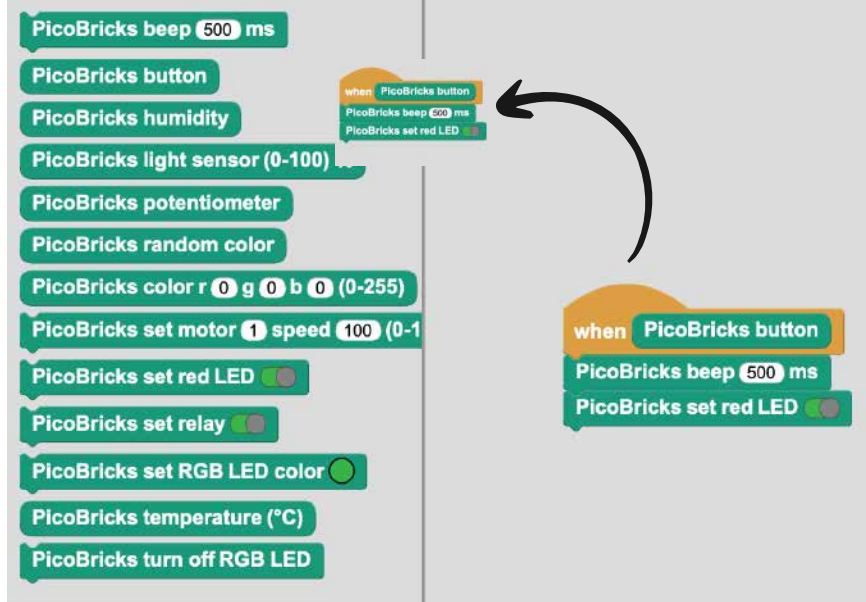
Picobricks ile butona bastıktan sonra RGB LED'i yarım saniye aralıklarla kırmızı, yeşil ve mavi renkte yakan hemen ardından buzzeri çalıştırıp OLED ekrana “picobricks” yazdıran kod bloklarını proje sayfanıza sürükleyiniz.


Proje Sayfasından Kod Bloğu Silelim

Proje sayfamızdaki bir kod bloğunu silmek istediğimizde o bloğun üzerine sağ tıkladıktan sonra “delete block” butonuna basarak silebiliriz.



Proje sayfamızda bulunan bir kod bloğunu / bloklarını sol tarafta bulunan kod blokları alanına tekrar sürükleyerek de silebiliriz. Hadi şimdi proje sayfamızda bulunan kod bloklarını sol tarafta bulunan bloklar sekmesine sürükleyerek silelim.



En son yaptığımız uygulama olan butona bastığında buzzer ve kırmızı LED’i aynı anda çalıştıran kodu program başlatıldığında yani proje sayfasının sağ üstünde bulunan  butonuna basıldığında çalışan kod ile değiştirelim. Bunun için yeni bir proje açmamıza gerek yok proje sayfamızda bulunan kod bloklarından en üstte bulunan bloğu silip yerine “When Started” bloğunu getirelim.

Sıcaklık ve Nem Sensörü Modülünü Kullanalım

Şimdi sıcaklık ve nem sensörü modülünü kullanarak ekrana bulunduğumuz ortamın sıcaklık değerini yazdıran kod bloklarını proje sayfamıza sürükleyelim.

Projeyi ve ekranı başlatmak için aşağıdaki iki kod bloğunu proje sayfamıza sürükleyelim.

when started

initialize i2c OLED_0.96in ▼ address(hex) 3C reset pin# 0 flip

Şimdi ekrana bir değer yazdırmak için kullandığımız kod bloğunu OLED kütüphanesinden proje sayfamıza sürükleyelim. Sonrasında PicoBricks kütüphanesinden sıcaklık değerini almamızı sağlayan kod bloğunu bu bloğun üzerine aşağıdaki resimde gösterildiği gibi sürükleyin..

PicoBricks temperature (°C) at x 0 y 0 inverse

write PicoBricks temperature (°C) at x 0 y 0 inverse

Bu kod bloğunu diğer iki kod bloğu ile birleştirdikten sonra projemizi çalıştırabiliriz.

when started

initialize i2c OLED_0.96in ▼ address(hex) 3C reset pin# 0 flip

write PicoBricks temperature (°C) at x 0 y 0 inverse

Hadi şimdi aynı şekilde **PicoBricks humidity** bloğunu kullanarak ekrana ortamın nem değerini yazdıralım.

Sadece sıcaklık değerini yazdıran bloğu proje sayfamızdan silip nem değerini aldığımız bloğu sürükleyerek bu işlemi yapabiliriz.

when started

initialize i2c OLED_0.96in ▼ address(hex) 3C reset pin# 0 flip

write PicoBricks humidity at x 0 y 0 inverse

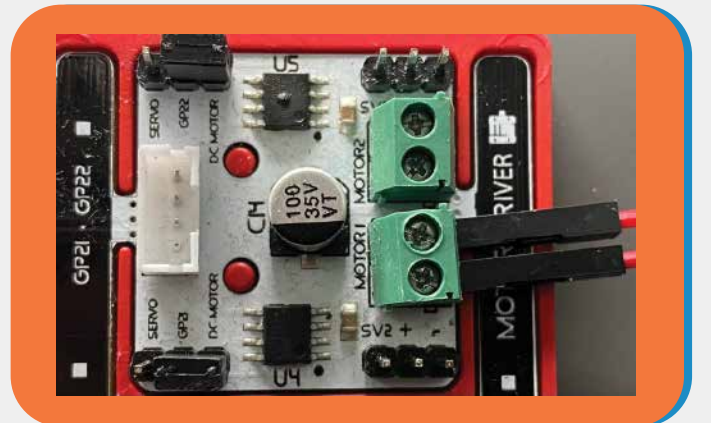
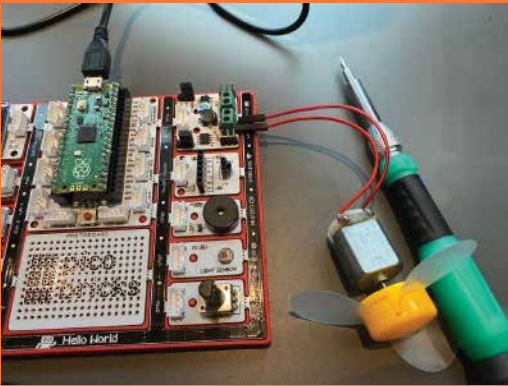
Motor Sürücü Modülünü Kullanarak ve Pervaneyi Çalıştıralım

PicoBricks kütüphanesinde bulunan **PicoBricks set motor 1 speed 100 (0-100)** bloğunu kullanarak set içeriğinde bulunan DC motoru çalıştıralım ve kendimize butonla çalışan basit bir pervane yapalım.

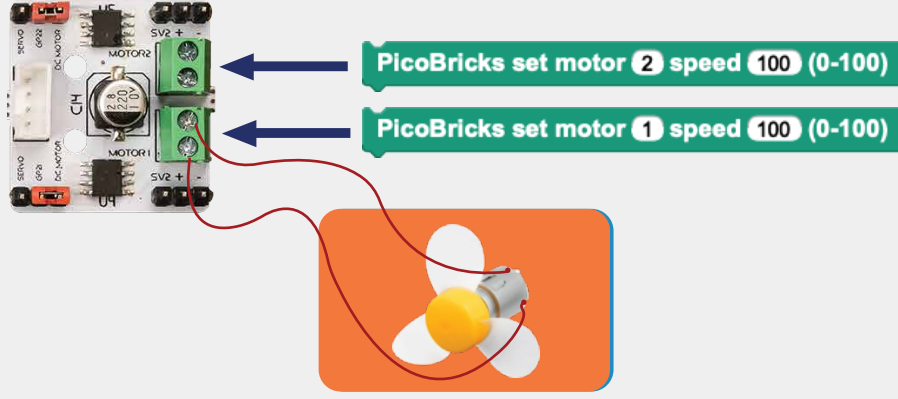


Bu etkinlik için Classroom kitinin içerisindeki Dc motor ve pervane setini kullanacağız.

1. Motor sürücü modülüne pervaneyi bağlayalım.
 - a. Motor sürücü modülünde DC motor kullanacağımız için resimde gösterildiği gibi siyah jumperı DC motor kısmı kapanacak şekilde takınız.
 - b. Daha sonra pervanenin motorunda bulunan iki kabloyu Dc motor 1 için ayrılan alana küçük bir tornavida yardımıyla takınız. **(Tornavida kullanmak istemiyorsanız jumper kablo uçlarını kullanarak bu alanlara karıştırabilirsiniz.)**



2. Şimdi kod bloklarını sürüklemeye başlayabiliriz."When Started" bloğunu ve motor sürücüyü kontrol etmemizi sağlayan **PicoBricks set motor 1 speed 100 (0-100)** bloğunu proje sayfamıza sürükleyelim.Motor sürücü modülünün üzerinde 2 adet DCmotor alanı bulunmaktadır. DC motoru 1. veya 2. alana taktığımızı kod bloğunda aşağıdaki resimde anlatıldığı gibi belirtmemiz gerekir. Biz pervaneyi 1. alana taktığımız için kod bloğunda ilgili alana 1 değerini yazalım.



when started

PicoBricks set motor 1 speed 100 (0-100)

3. Butona bastığımızda pervaneyi durdurmak için aşağıdaki kod bloklarını proje sayfamıza sürükleyelim.DC motoru durdurmak için **PicoBricks set motor 1 speed 100 (0-100)** bloğundaki "speed" değerini "0" girmemiz gerekmektedir.

when PicoBricks button

PicoBricks set motor 1 speed 0 (0-100)

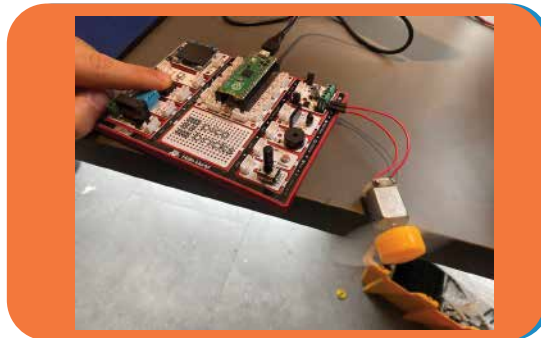
4. Kodumuz hazır şimdi start butonuna basarak pervanemizi çalıştırabiliriz. Durdurmak için PicoBricks üzerinde bulunan butonu kullanmalısınız.

when PicoBricks button

PicoBricks set motor 1 speed 100 (0-100)

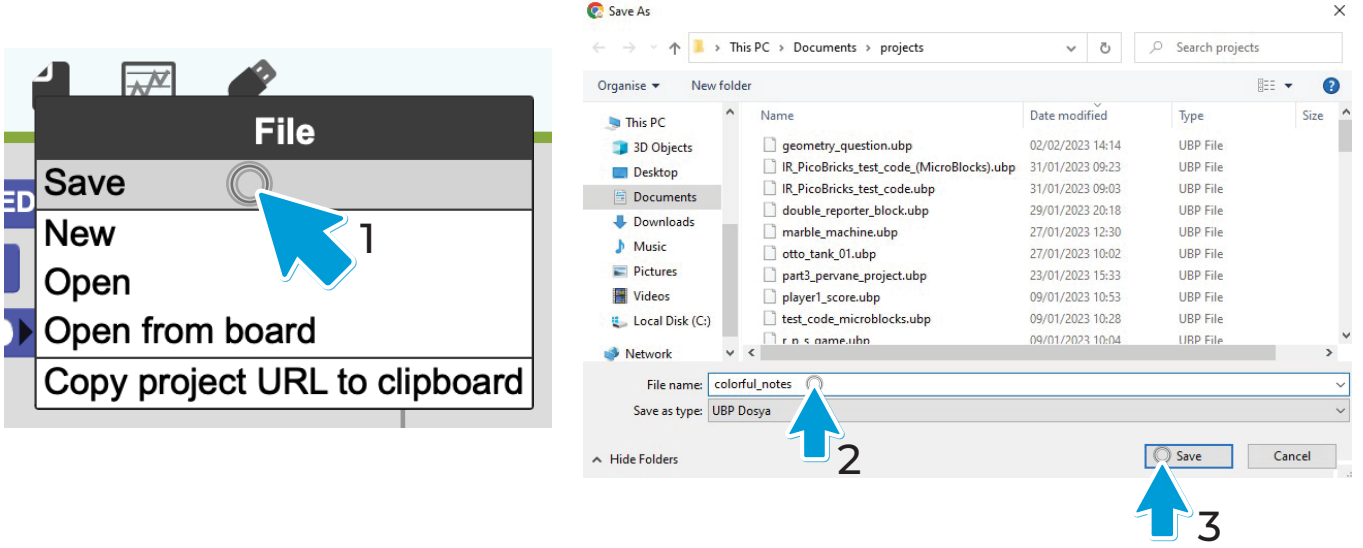
when PicoBricks button

PicoBricks set motor 1 speed 100 (0-100)



Projemizi Bilgisayara Kaydedelim

Aşağıdaki adımları takip ederek projemizi bilgisayara kaydedelim.



K-12 Uluslararası Bilgisayar Bilimi Standartları

- 1B-AP-09-3-5 Create programs that use variables to store and modify data.
- 1B-AP-16-3-5 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.

set picobricks ▼ to 0

set picobricks ▼ to 0

picobricks

PART 3

picobricks

picobricks

set picobricks ▼ to 0

Bu Bölümde Öğreneceklerimiz:

- Bu bölümde öğrenci ;
- Değişkenin tanımını,
- MicroBlocks platformunda değişken oluşturmayı,
- Döngünün tanımını ve çalışma mantığını ve
- MicroBlocks platformu ile repeat döngüsünü kullanmayı öğrenir.

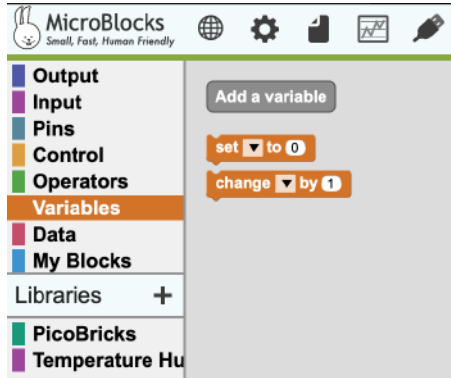
Değişken Tanımlayalım ve Döngüye Giriş Yapalım

(x) Değişken Nedir?

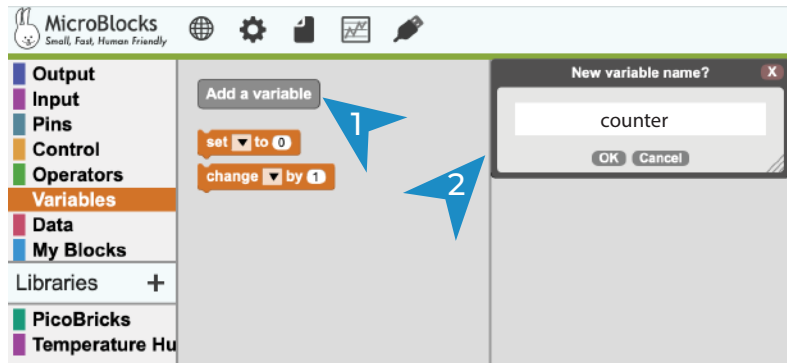
Değişkenler bilgisayara programında değiştirilebilecek bilgilerin depolandığı yapılardır. Değişkenleri bilgileri tutan kaplar olarak düşünebiliriz.

MicroBlocks Platformunda Değişken Oluşturalım

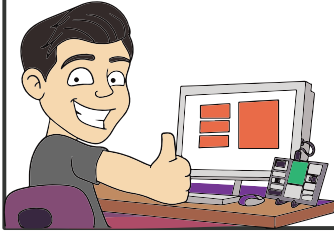
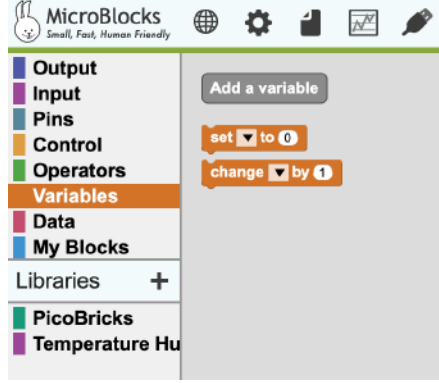
- “Variables” kod sekmesine tıklayalım.



- “Add a variable” seçeneğine tıklayarak değişkenimizin ismini belirleyelim. Biz burada örnek olması için değişken ismini “counter” olarak tanımladık.



3. Değişken ismini belirleyip “ok” butonuna bastıktan sonra oluşturduğumuz değişken bloğu “Variables” blokları arasına gelir. Artık oluşturduğumuz “counter” değişkenini projemizde kullanabiliriz.



Şimdi Sıra Sizde!

Şimdi yukarıdaki adımları takip ederek “PicoBricks” adında bir değişkeni oluşturalım.

PicoBricks Değişkeninin Başlangıç Değerini Belirleyelim

- PicoBricks değişkenin başlangıç değerini “0” olarak belirleyelim. Projemizi başlattığımızda PicoBricks değişkeninin başlangıç değerini belirlemek için “Variables” bloklarından **set OLEDReady ▼ to 0** bloğunu kullanacağız. Bu bloğun içindeki değeri PicoBricks olarak değiştirelim ve başlangıç değerini “0” olarak tanımlayalım.

set picobricks ▼ to 0

- Oluşturduğumuz bu kod bloğunu “When Started” bloğuyla birleştirdikten sonra PicoBricks değişkeninin başlangıç değerini “0” yapmış olacağız.

when started

set picobricks ▼ to 0

initialize i2c OLED_0.96in ▼ address(hex) 3C reset pin# 0 flip

when started

set picobricks ▼ to 0

initialize i2c OLED_0.96in ▼ address(hex) 3C reset pin# 0 flip

PicoBricks Değişkenini Butona Basarak Arttıralım

Butona her basıldığında PicoBricks değişkeninin değerini arttırmak için ekrana yazdıralım. Bunun için butona her basıldığında altındaki kod bloklarını çalıştıran kod bloklarını oluşturalım.

when PicoBricks button

1. Döngü her tekrar ettiğinde “picobricks” değişkenini birer birer arttırmak için

change picobricks ▼ by 1

bloğunu sürükleyelim. OLED ekranda “picobricks”

değişkeninin değerini yazdırmak için OLED ekranı başlatalım ve

write Hello! at x 0 y 0 inverse

bloğunun içine “Variables” bloklarından picobricks bloğunu sürükleyelim. Elde ettiğimiz bu kod bloğu ekrana picobricks değişkenini yazdırmamızı sağlayacak. Bu adımdan sonra tepkime süresini azaltmak için

wait 500 millisecs

wait 500 millisecs bloğunu projemize sürükleyelim. Kod bloklarını oluşturduğuna göre projemizi çalıştırabiliriz. Butona her bastığımızda önce PicoBricks değişkeni ekrana yazılacak daha sonra “1” artacaktır.

when started

initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip

set picobricks ▼ to 0

when PicoBricks button

change picobricks ▼ by 1

write picobricks at x 0 y 0 inverse

wait 500 millisecs

↻ Döngü Nedir?

Belirli bir koşul sağlanana kadar sürekli tekrar eden işlemlere döngü denir. Örneğin Çay demlemek, ocağa su koydunuz ve kaynamasını bekliyorsunuz belirli aralıklarla suyun kaynayıp kaynamadığını kontrol etmeniz gerekmektedir. İşte bu bir döngüdür.

MicroBlocks’ da içerisine sürüklediğimiz kod bloklarını belirlediğimiz değer kadar tekrar

repeat 10 ettirmek için bloğunu kullanırız.

Repeat döngüsünü kullanarak belirlediğimiz değer kadar Kırmızı LED’i yakalım

Projeyi sayfasından projeyi çalıştırdığımızda kod bloklarının çalışması için “When started” bloğunu proje sayfamıza taşıyalım.

• Proje başlatıldıktan hemen sonra döngünün başlaması için repeat bloğunu proje sayfamıza sürükleyelim. Repeat bloğunun üzerinde yazan sayı repeat bloğunun içerisindeki kod bloklarının kaç kere tekrar edeceğini belirler.



Burada repeat bloğunun içerisindeki kod blokları 10 kere tekrar edecek şekilde ayarlanmıştır.

Şimdi repeat bloğunun içerisinde 10 kere tekrar edecek kod bloklarını belirleyelim. PicoBricks üzerinde bulunan kırmızı LED'i yaktıktan sonra yarım saniye bekleyelim ve kapatalım daha sonra döngünün tekrar ettiğini daha iyi anlamak için sonuna bir yarım saniye bekleme bloğu daha koyarak bu blokları repeat bloğunun içerisinde sürükleyelim ve projemizi çalıştıralım.



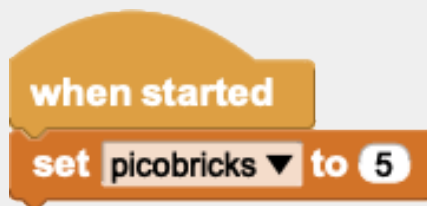
Şimdi Sıra Sizde!

Oluşturduğumuz "Picobricks" adındaki değişkenin başlangıç değerini 5 olarak belirledikten sonra repeat döngüsünü kullanarak 5 kere yarım saniye aralıklara kırmızı LED'i yakıp söndürelim.

Projenin Algoritması

1. Başla.
2. PicoBricks değişkeninin başlangıç değerini 5 olarak belirle
3. PicoBricks değişkenin değeri kadar aşağıdaki adımları takip et.
 - a) Kırmızı LED'i Aç
 - b) Yarım saniye bekle
 - c) Kırmızı LED'i Kapat
 - d) Yarım Saniye Tekrarla
4. İlk adıma geri dön

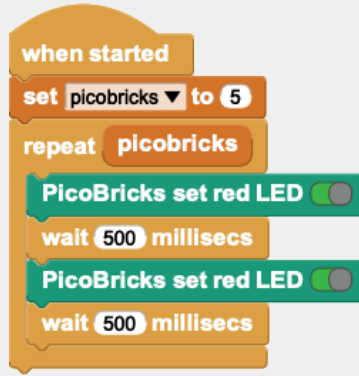
- Kontrol bloklarından "When started" bloğunu proje sayfanıza sürükleyin. Daha sonra önceden oluşturduğumuz PicoBricks değişkenine başlangıç değerini yazınız.



- Repeat bloğunu proje sayfasına sürükleyin ve repeat değerinun ierisine PicoBricks deęişkenini sürükleyin.



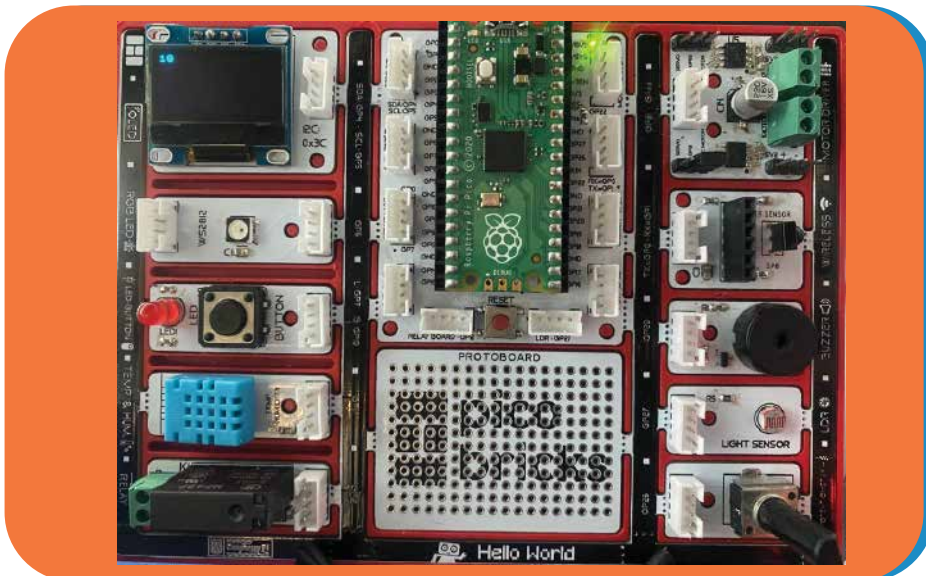
- Kırmızı LED'i yarım saniye aralıklarla yakıp söndürmek için PicoBricks kütüphanesinden ve kontrol bloklarından gerekli olan kod bloklarını sürükleyin.



Saya Etkinlięi

Saya etkinlięi, deęişken ve döngü kullanarak ekrana sırayla 1'den 10'a kadar olan sayıları yazdırmayı hedeflemektedir.

Etkinlik Görseli



Etkinliđi MicroBlocks Kodu

OLED kütüphanesini yüklemeyi unutma!

1. İlk olarak Controls bloklarından “When Started” blođunu proje sayfamıza sürükleyelim. Bu sayede programı bařlattıktan sonra bu blokla birleřtirdiđimiz diđer kod blokları çalıřmıř olacak.

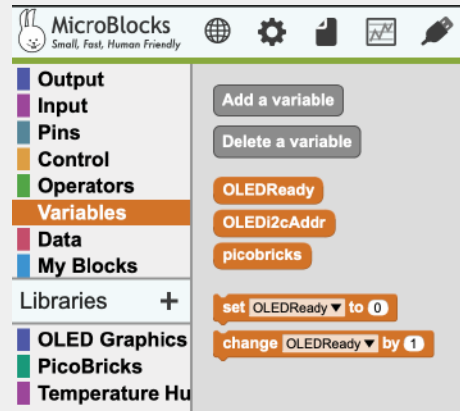
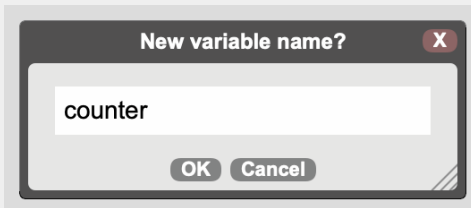
when started

2. Etkinlikte OLED ekran kullanacađımız için OLED ekranı bařlat blođunu projemize sürükleyelim.

when started

initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip

3. Sayaç deđiřkeni oluřturalım ve projemize sürükleyip bařlangıç deđerini 0 olarak belirleyelim.



when started

initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip

set counter to 0

Öğretmene Öneri

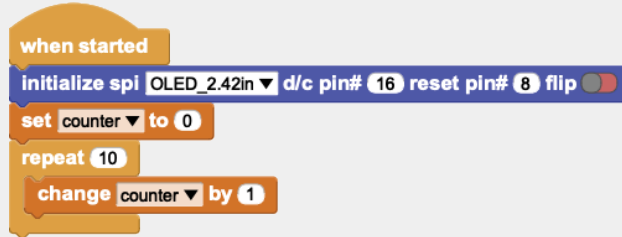
Bu adıma kadar yaptıklarımız öğrendiğimiz bilgilerden oluşmaktadır. Adımlar öğrencilere anlatıldıktan sonra öğrencilerin yardım almadan adımları gerçekleştirilmesi beklenir. Gerekli durumlarda yardım edilebilir.

4. Control blokları içerisinde repeat döngü bloğunu projemize sürükleyelim ve tekrar sayısını 10 olarak belirleyelim. Bu sayede repeat döngüsünün içerisinde sürüklediğimiz bütün bloklar 10 kere tekrar eder.



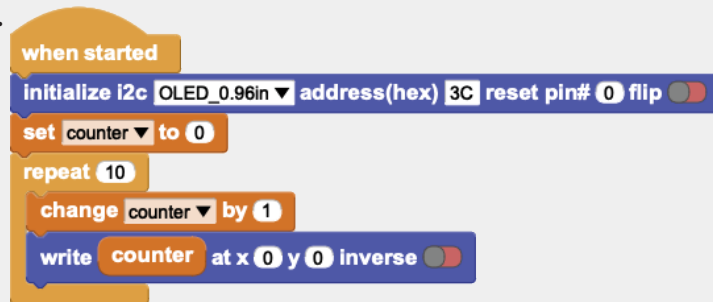
```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set counter to 0
  repeat 10
```

5. Başlangıç değerini 0 olarak belirlediğimiz sayaç değişkenini 1'er 1'er arttırmak için **change counter by 1** bloğunu kullanacağız. Change bloğunun içerisinde counter değişkenini seçtikten sonra artış miktarını "1" olarak belirleyiniz.



```
when started
  initialize spi OLED_2.42in d/c pin# 16 reset pin# 8 flip
  set counter to 0
  repeat 10
    change counter by 1
```

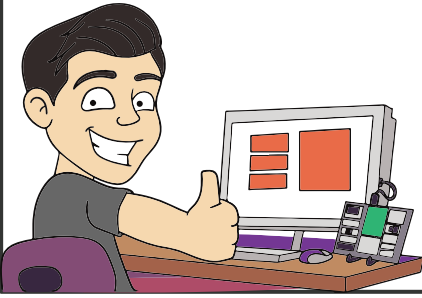
6. Sayaç değişkenini OLED ekrana yazdırmak için OLED Graphic kütüphanesinden **write Hello! at x 0 y 0 inverse** bloğunu sürükleyelim. Ekrana "counter" değişkenini yazdırmak için "Variables" bloklarından "counter" değişkeni bloğunu bu bloğun içerisine sürükleyiniz.



```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set counter to 0
  repeat 10
    change counter by 1
    write counter at x 0 y 0 inverse
```


7. Sayaç deęişkeninin, artışını OLED ekranda daha rahat gözlemleyebilmek için 0.5 saniye bekle bloęunu projemize sürükleyelim. Bu sayede sayaç deęişkeni yarım saniye aralıklarla deęişecektir.

```
when started
  initialize I2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set counter to 0
  repeat 10
    change counter by 1
    write counter at x 0 y 0 inverse
    wait 500 millisecs
```



Şimdi Sıra Sizde!

“counter” deęişkenini 2’şer 2’şer arttırarak ekrana yazdıralım. Ekrana yazan son sayının “10” olması gerekmektedir.



İPUCU

İpucu: Daha önce yaptığımız etkinlięin kod bloklarının içerisindeki iki deęeri deęiştirerek bu projeyi gerçekleştirebiliriz. Ekranda yazan son sayısının 10 olması gerektięini unutmayın !

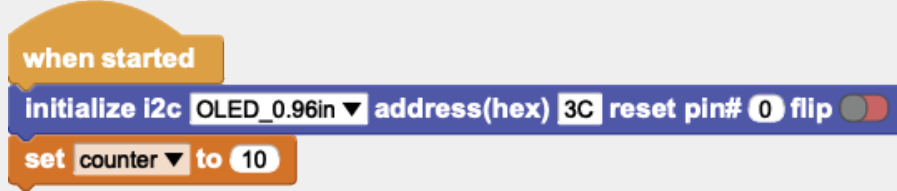
2.Sayaç Etkinlięi

Hadi şimdi bu etkinlięin tersini yapalım.10 ile 0 arasındaki sayıları 1’ er 1’ er azaltarak ekrana yazdıran MicroBlocks kod bloklarını proje sayfamıza sürükleyelim. **(10 ile 1 dahil deęil.)** Bu etkinlięi yapmak için kodu tekrar yazmamıza gerek yok ilk etkinlikteki birkaç noktayı deęiştirdikten sonra etkinlik için gerekli kod blokları hazır olacaktır. Gelin şimdi bu kod bloklarını inceleyelim.Aşağıdaki resimde bi önceki kod bloklarının üzeridne deęişmesi gereken noktalar işaretlenmiştir.

```
when started
  initialize I2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set counter to 0
  repeat 10
    change counter by 1
    write counter at x 0 y 0 inverse
    wait 500 millisecs
```

ilk iki kod bloğu proje ve OLED ekranın başlaması için gerekli kod blokları olduğu için onları aynı şekilde proje sayfamıza sürüklüyoruz.

“counter”, değişkenini tanımladıktan sonra 10’dan geriye doğru gideceğimiz için başlangıç değişkenini 10 olarak belirleyelim.





```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set counter to 10
```

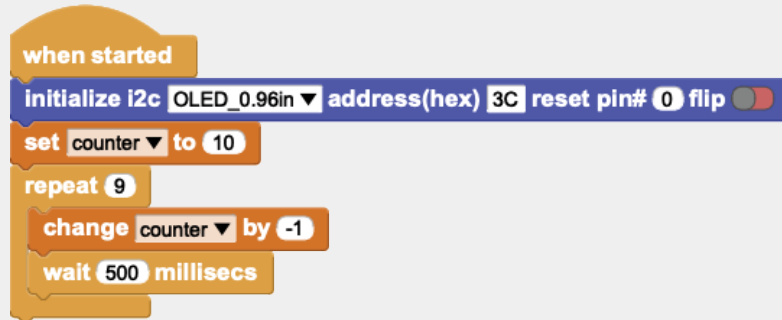
10 ile 0’ı dahil etmediğimiz için repeat döngüsünü 9 olarak tanımlayalım.



```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set counter to 10
  repeat 9
```

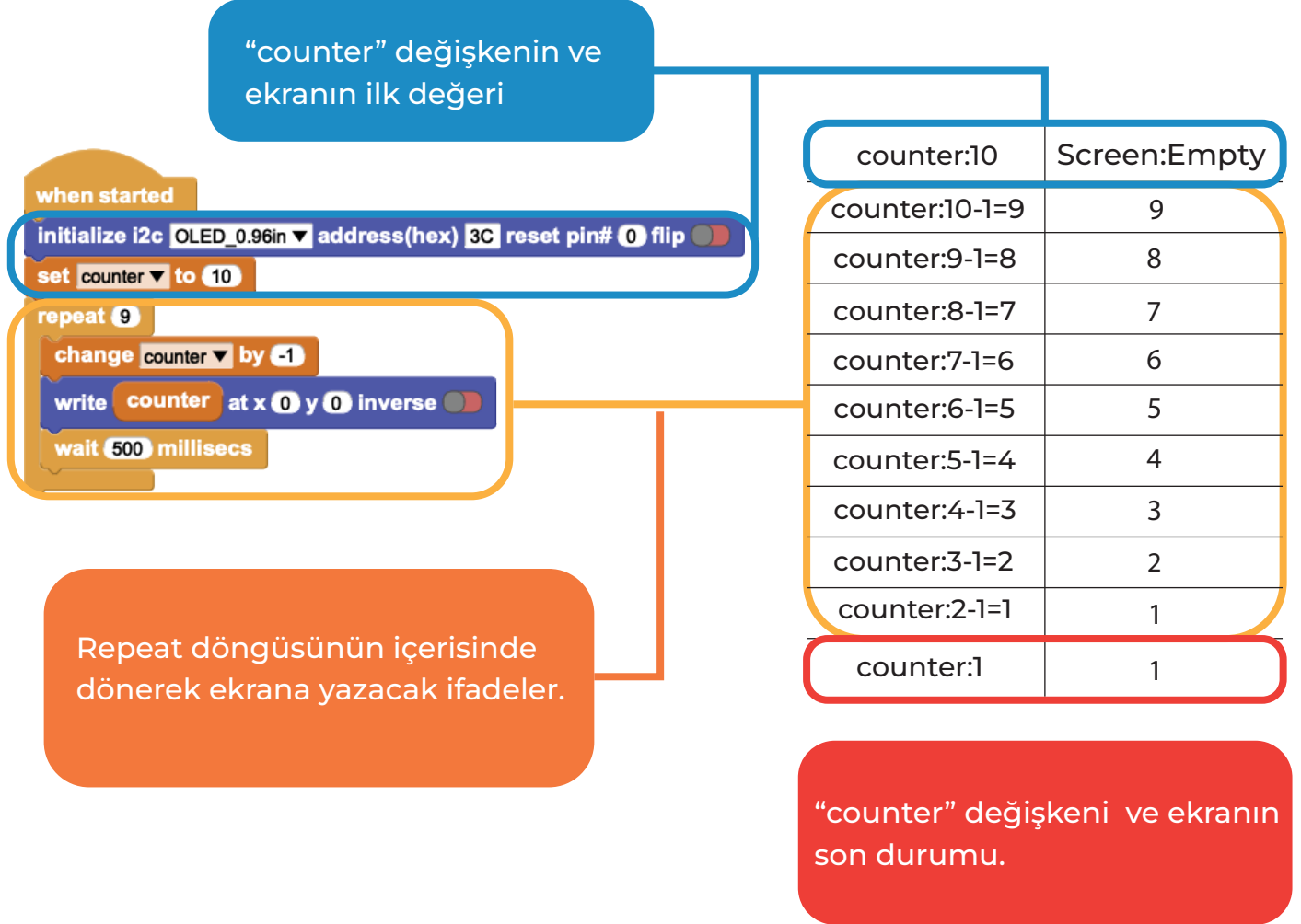
Repeat döngüsü her tekrar ettiğinde “counter” değişkenin 1 azalması için .

 bloğunda artış miktarını “-1” olarak belirleyelim. 



```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set counter to 10
  repeat 9
    change counter by -1
    wait 500 millisecs
```

Şimdi yazdığımız bu kod bloğunun nasıl çalıştığını aşağıdaki şema üzerinden inceleyelim.



2.Bölümde yaptığımız pervane etkinliğinde butona bastığımızda pervane duruyor basmadığımız zamanlarda çalışıyordu. Şimdi butona bastığımızda pervanenin çalışmasını basmadığımız zamanlarda ise durmasını sağlayalım.

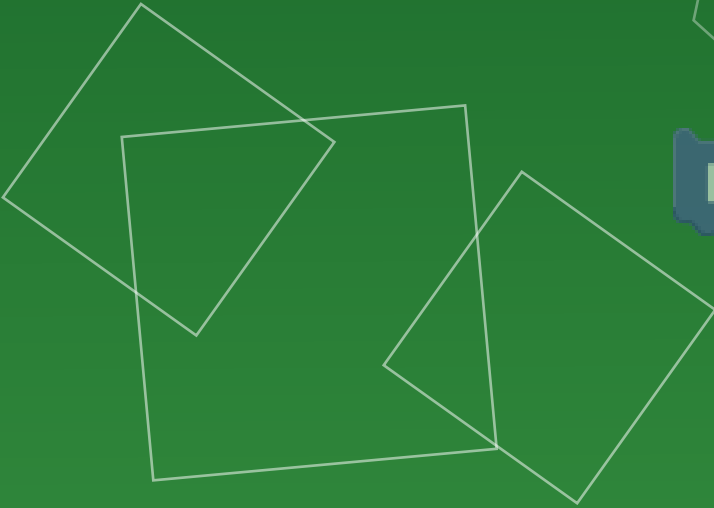
1. Çalışma kağıdındaki soruları çözünüz!
2. Çalışma kağıdındaki soruları çözünüz!

K-12 Uluslararası Bilgisayar Bilimi Standartları

- 2-AP-16 -6-8 - Incorporate existing code, media, and libraries into original programs, and give attribution.
- 2-AP-11 6-8 Create clearly named variables that represent different data types and perform operations on their values.
- 3A-AP-13- 9-10- Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
- 3A-AP-21- 9-10 - Evaluate and refine computational artifacts to make them more usable and accessible.



pen down



PART 4

move 10



turn 90 degrees



pen up



Bu Bölümde Öğreneceklerimiz

Bu bölümde öğrenciler;

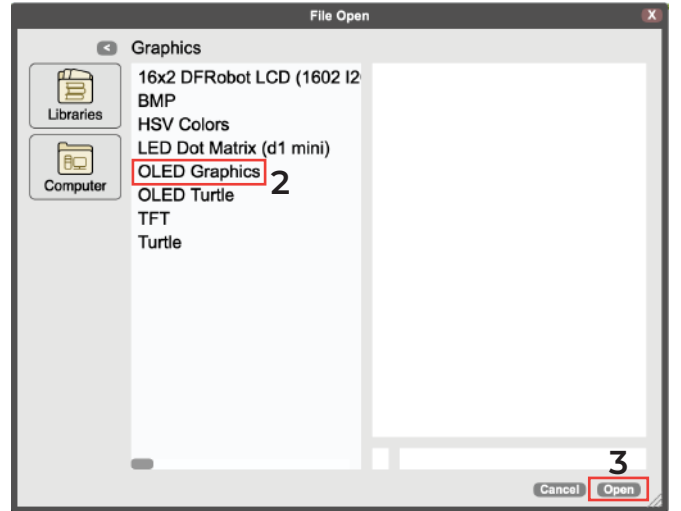
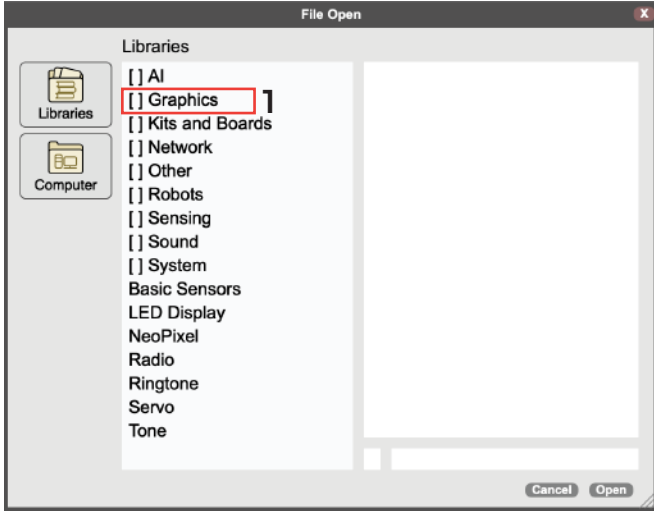
- Döngüleri blok tabanlı programlama araçlarında kullanmayı,
- Geometrik şekilleri döngü kullanarak oluşturmayı ve
- MicroBlocks platformunda turtle kütüphanesi ile proje geliştirmeyi öğrenir.

Turtle ile Döngüler

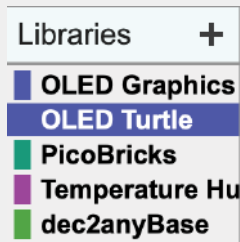
Kaplumbağa geometrisi programlama eğitimlerinde sıklıkla tercih edilen bir yöntemdir. PicoBricks modüllerinden OLED ekran üzerinde MicroBlocks turtle kütüphanesini kullanarak turtle projeleri gerçekleştirebiliriz.

**Hadi şimdi Turtle kütüphanesini
MicroBlocks editörü içerisine yükleyelim.**

Aşağıdaki şemada verilen adımları takip ederek Turtle kütüphanesini projemize yükleyelim.



Turtle kütüphanesini OLED ekranla beraber kullanacağımız için proje sayfamıza OLED Graphics kütüphanesini de yüklememiz gerekir.



Turtle projelerini OLED ekran üzerinden gözlemleyeceğimiz için OLED ekranı başlatan `initialize i2c OLED_0.96in` address(hex) `3C` reset pin# `0` flip bloğunu projemizin başlangıcında mutlaka kullanmalıyız. Şimdi Turtle kütüphanesinde bu bölüm için kullanacağımız kod bloklarının görevlerini öğrenelim.

<code>initialize turtle large</code>	Bloğu Turtle boyutunu belirlememize olanak sağlar.
<code>pen down</code>	Bloğu turtle imlecinin gittiği yol üzerine çizgi çizmemizi sağlar.
<code>move 10</code>	Bloğu Turtle imlecinin tek hamlede kaç adım ileri gideceğini belirler.
<code>turn 90 degrees</code>	Bloğu Turtle imlecinin kaç derece döneceğini belirler.

Aşağıdaki turtle projesini yorumlayalım

```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  pen down
  move 10
  turn 90 degrees
  move 10
  turn 90 degrees
```

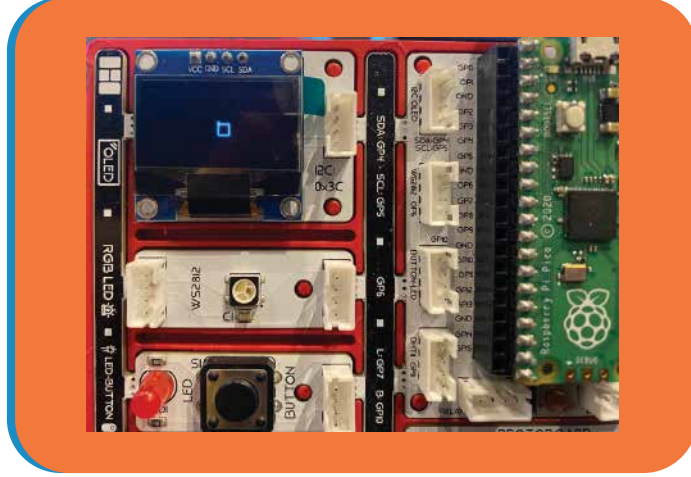
- OLED ekranı başlat.
- Turtle imlecinin gittiği yol üzerinde çizgi çiz.
- Turtle imlecini 10 adım ilerlet.
- Turtle imlecini 90 derece döndür.
- Turtle imlecini 10 adım ilerlet.
- Turtle imlecini 90 derece döndür.



Turtle ile Şekiller

OLED ekran üzerinde Turtle kütüphanesindeki `turn 90 degrees` bloğunun derece değerini değiştirerek çeşitli geometrik şekiller oluşturabiliriz. Hadi şimdi bu geometrik şekilleri nasıl oluşturabileceğimizi biraz inceleyelim;

KARE



Turtle imlecinin sadece tek bir yöne hareket ettiğini düşünelim. (Bu yön sağ olsun.) Sürekli sağ tarafa 90 derece dönen bir imlec 4 kere bu hareketi tekrarladığında kare şeklinde hareket etmiş olacaktır. Bu süreçte izlediği yolları kalemle çizerek olursak kare şeklini gözlemlemiş oluruz.



Turtle imlecinin hareketlerini takip edebilmek için yukarıdaki kod bloklarını proje sayfamıza sürüklememiz gerekmektedir. Bu kod bloklarını sırasıyla açıklayacak olursak;

- Projemizi başlattıktan sonra altına gelen kod bloklarını çalıştırmamızı sağlar.
- Turtle imlecini OLED ekran ile kullanacağımız için OLED ekranı başlamamızı sağlayan kod bloğudur.
- Turtle imlecinin boyutunu belirleyen kod bloğudur.
- Turtle imlecinin izlediği yolu takip etmemizi sağlayan kod bloğudur.

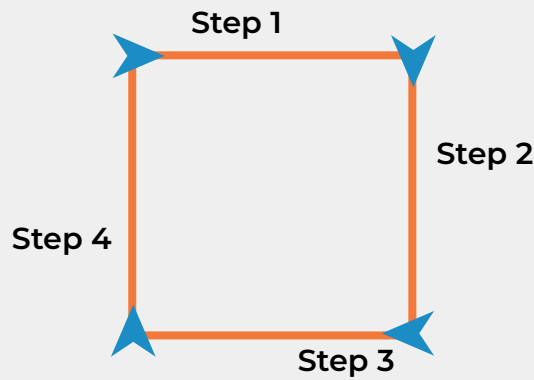
Şimdi Turtle imlecini 90 derece döndürerek kare bir yol izlemesini sağlayalım. Adım sayısını değiştirerek kenar uzunluğunu arttırabilir/azaltabilirsiniz.

```
move 10
turn 90 degrees
```

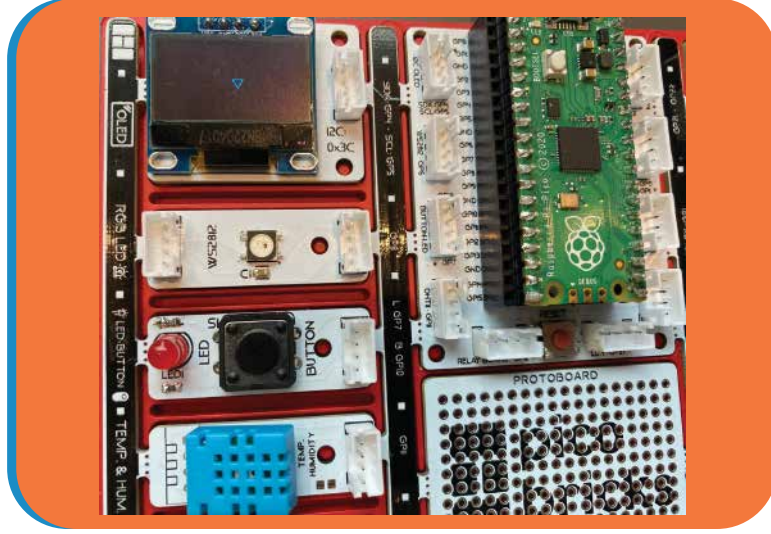
Yukarıdaki kod bloğunu 4 kere arka arkaya birleştirirsek ekranda 10 adım kenar uzunluğunda bir kare gözlemlemiş olacağız.

```
when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
initialize turtle large
pen down
move 10
turn 90 degrees
move 10
turn 90 degrees
move 10
turn 90 degrees
move 10
turn 90 degrees
```

Aşağıdaki şemada turtle imlecinin takip ettiği adımlar belirtilmiştir.



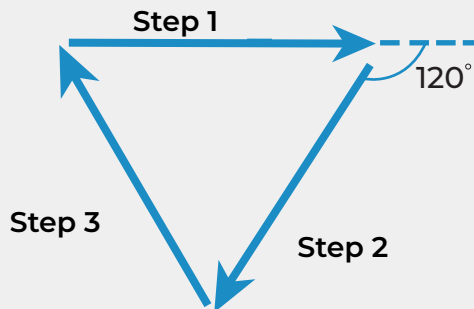
ÜÇGEN



Eğer turtle imleci 3 kere 120 derecelik bir açıyla hareket ederse üçgen şeklinde hareket etmiş olur. Hadi şimdi üçgen oluşturalım.

```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  initialize turtle large
  pen down
  move 10
  turn 120 degrees
  move 10
  turn 120 degrees
  move 10
  turn 120 degrees
```

Şimdi yukarıdaki kod blokları ile turtle imlecinin hareketlerini inceleyelim.



Yukarıda Turtle ile oluşturduğumuz şekillerde adım sayısı kadar aynı blokları tekrar ederek kare ve üçgen şeklini OLED ekran üzerinde oluşturduk. Aynı şekilleri Repeat bloğu kullanarak da oluşturabiliriz. Repeat bloğunu kullanarak kare şeklini OLED ekran üzerinde oluşturalım;

Daha önce kare şeklini oluşturduğumuz kod bloklarında aşağıdaki iki kod bloğunu dört kere arka arkaya getirerek şekli oluşturduk.

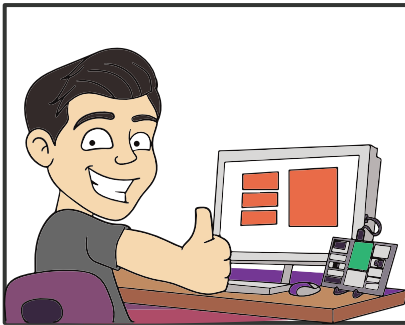
```
move 10
turn 90 degrees
```

Şimdi bu kod bloğunu repeat bloğunun içerisine sürükleyerek 4 kere tekrar etmesini sağlayalım.

```
repeat 4
  move 10
  turn 90 degrees
```

Repeat bloğunun değerini 4 yaparak içerisindeki kod bloklarının 4 kere tekrar etmesini sağladık..

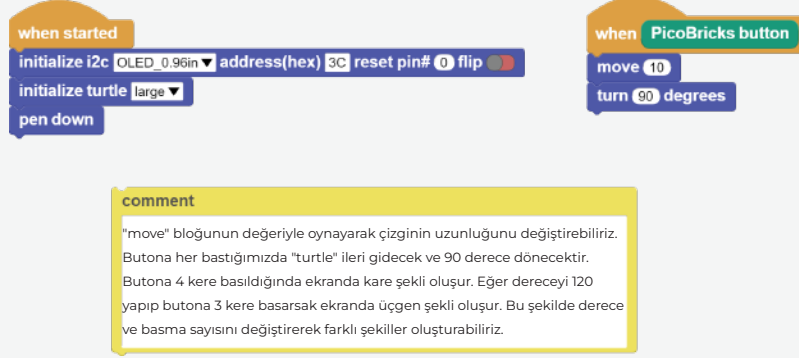
```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  initialize turtle large
  pen down
  repeat 4
    move 10
    turn 90 degrees
```



Şimdi Sıra Sizde!

Repeat bloğunu kullanarak turtle imleci ile üçgen şeklini oluşturunuz.

Öğretmene Öneri



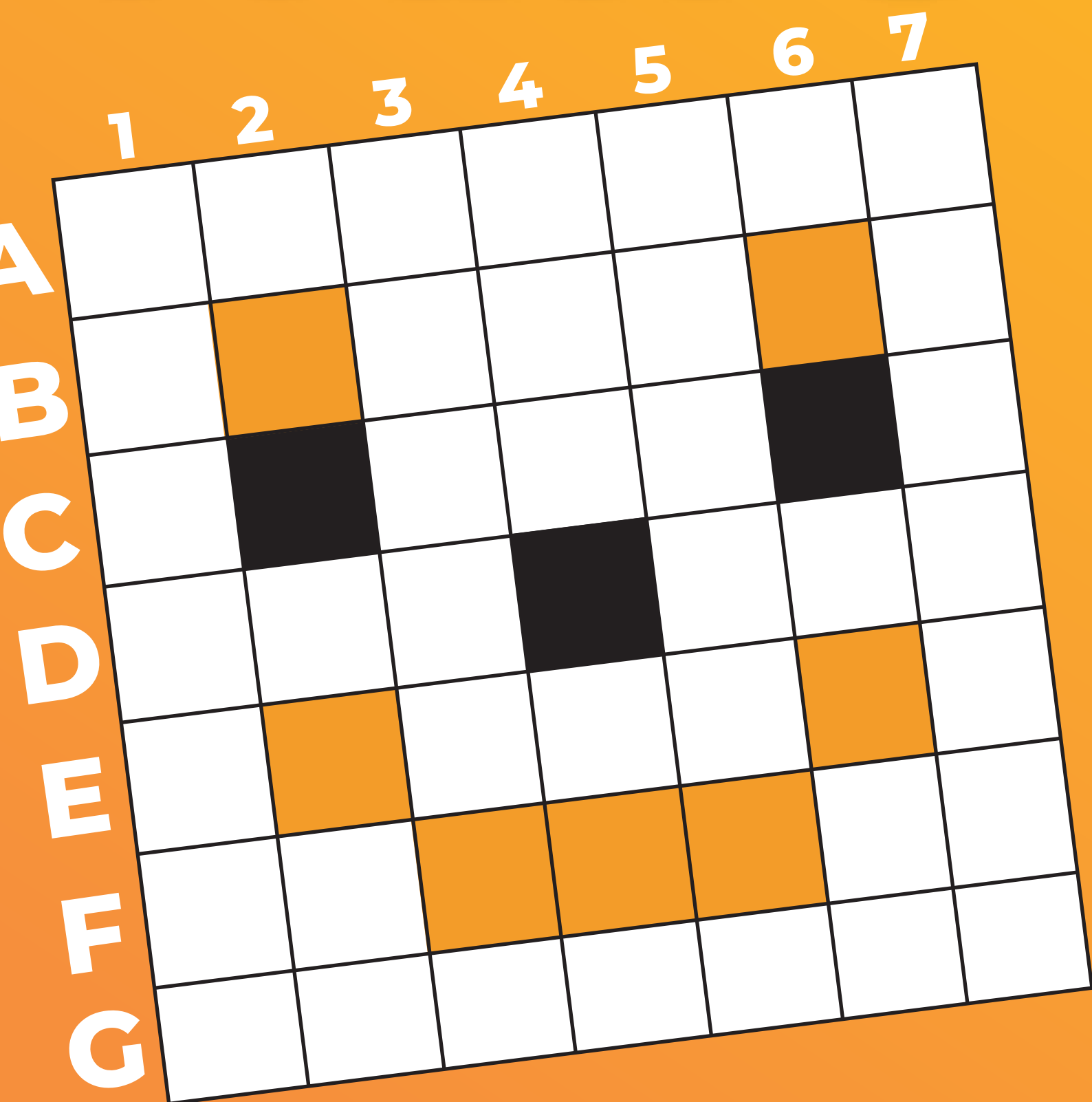
Yukarıdaki kod bloklarını bu etkinliğe başlamadan önce, hareketlerin tekrar sayısını ve döngü mantığını anlatmak için kullanabilirsiniz. Örneğin dereceyi 90 yaptıktan sonra butona 4 kere butona basarak turtle imlecini ile ekranda kare şekli oluşturulabilir. Dereceyi 120 yaparak öğrencilere "Butona kaç kere basarsam ekranda üçgen şekli oluşur?," "Butona üç kere bastığımda hangi şekil oluşur?" vb. sorular yöneltilerek döngülerin çalışma mantığı anlatılabilir.

Çalışma kağıdındaki soruları çözün!

K-12 Uluslararası Bilgisayar Bilimi Standartları

- 1B-AP-09 3-5 Create programs that use variables to store and modify data.
- 1B-AP-10 3-5 Create programs that include sequences, events, loops, and conditionals.
- 3A-AP-13 9-10 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

PART 5



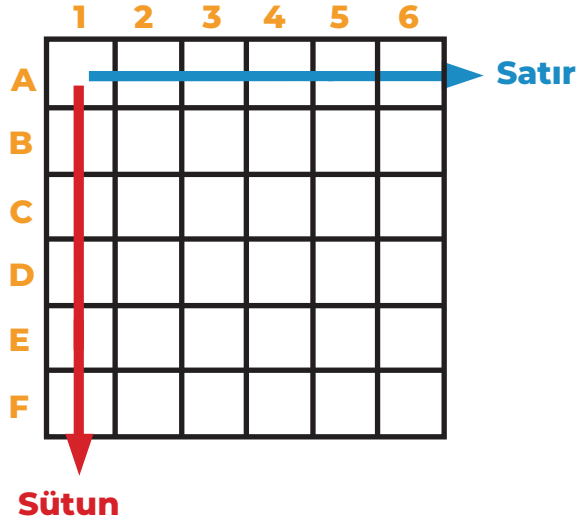
Bu Bölümde Öğreneceklerimiz

Bu bölümde öğrenci;

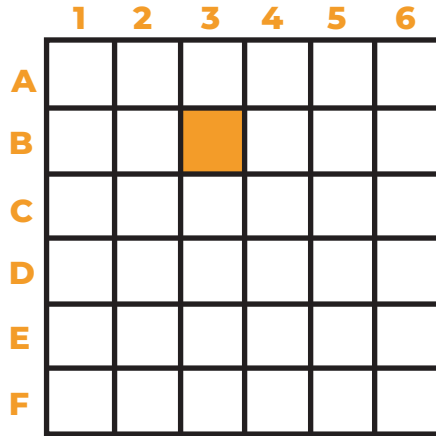
- Satır ve sütunları kullanarak koordinatları tespit etmeyi,
- Picobricks OLED ekran modülünün konumlandırma mantığını kavrar.

Bilgisayarsız: Amiral Battı Oynayalım

6x6 kareli oyun alanında amiral battı oynayarak koordinat sistemini öğrenelim.



Gerçek bir amiral battı oyununa başlamadan önce 6x6 oyun alanını tanıyalım. Oyun alanındaki her bir karenin kendine ait bir Koordinat noktası vardır. Örneğin; Aşağıdaki 6x6 oyun alanında turuncu renkle işaretlenmiş noktanın koordinatı **B3'dür**.



Oyun alanında harf ile belirtilen alanlar satırları, numara ile belirtilen alanlar ise sütunları temsil eder. Yukarıdaki oyun alanında işaretlenen nokta B satırında ve 3 numaralı sütundadır. Bu nedenle koordinat noktası **B3**.

- Aşağıdaki çalışma kağıdındaki soruları çözünüz.

	1	2	3	4	5	6
A						
B						
C						
D						
E						
F						

Yandaki tabloya göre aşağıdaki koordinat noktalarından hangisi turuncu renkle boyanmamıştır. Yanında verilen daireyi boyayarak belirtiniz.

- | | | | |
|-----------------------|-----------|-----------------------|-----------|
| <input type="radio"/> | A5 | <input type="radio"/> | D4 |
| <input type="radio"/> | B3 | <input type="radio"/> | A2 |
| <input type="radio"/> | C3 | <input type="radio"/> | D2 |
| <input type="radio"/> | B4 | <input type="radio"/> | C4 |

	1	2	3	4	5	6
A						
B						
C						
D						
E						
F						

Yandaki tabloya göre aşağıda verilen koordinat noktalarında boş bırakılan alanları doldurunuz.

- | | | | |
|-----------------------|-----------|-----------------------|-----------|
| <input type="radio"/> | D_ | <input type="radio"/> | E_ |
| <input type="radio"/> | C_ | <input type="radio"/> | D3 |
| <input type="radio"/> | C5 | <input type="radio"/> | E3 |
| <input type="radio"/> | B_ | <input type="radio"/> | F_ |

	1	2	3	4	5	6
A						
B						
C						
D						
E						
F						





Aşağıda verilen koordinat noktalarını yandaki oyun tablosunda doğru yerleri boyayarak belirtiniz.

- | | | | |
|-----------------------|-----------|-----------------------|-----------|
| <input type="radio"/> | A2 | <input type="radio"/> | D2 |
| <input type="radio"/> | B2 | <input type="radio"/> | D5 |
| <input type="radio"/> | E3 | <input type="radio"/> | A5 |
| <input type="radio"/> | E4 | <input type="radio"/> | B5 |

• Şimdi 6x6 oyun alanında amiral battı oynayalım.

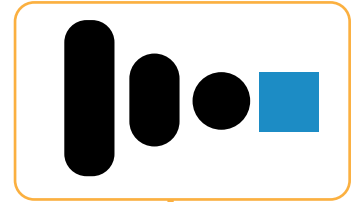
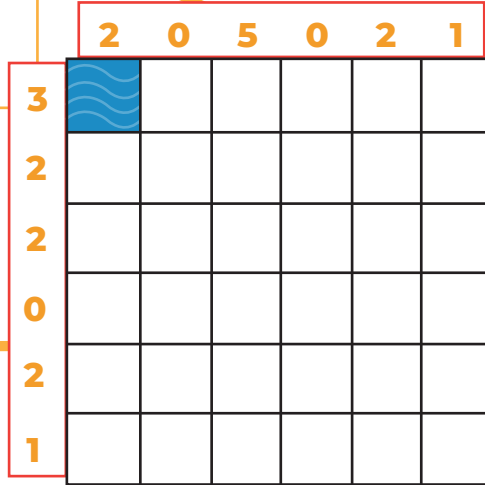
Oyun alanını tanıdığımızı göre şimdi amiral battı oyununun nasıl oynandığına bakalım.

Aşağıda verilen örnek üzerinden amiral battı oyununu inceleyelim.

• Denizi temsil eder oyun alanında bu mavi kutucuğu gördüğümüzde o koordinatlarda gemi yok demektir.	
• Oynayacağımız amiral battı oyununda 1,2 ve 3 birimlik üç tür gemi bulunmaktadır;	
• Bir birimlik gemiyi temsil eder.	
• İki birimlik gemiyi temsil eder.	
• Üç birimlik gemiyi temsil eder.	

Sütunların başındaki numaralar, bulunduğu sütunda kaç birim karenin dolu olduğunu belirtir.

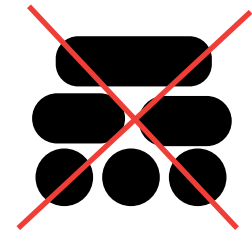
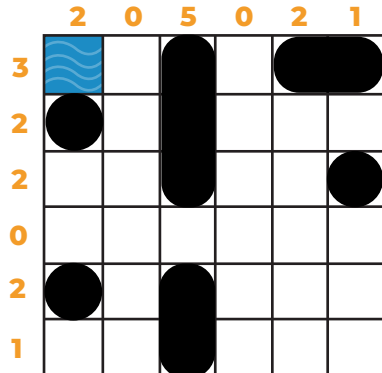
Satırların başındaki numaralar, bulunduğu satırda kaç birim karenin dolu olduğunu belirtir.



Oyun alanına yerleştirilmesi gereken gemileri temsil eder. Bu oyunda bir tane 3 birim, iki tane 2 birim, üç tane 1 birim gemi var demektir.

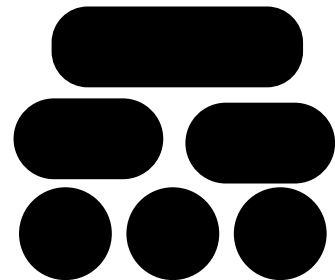
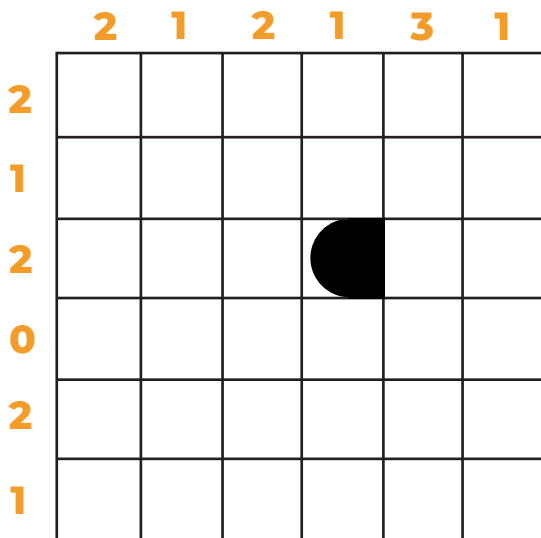
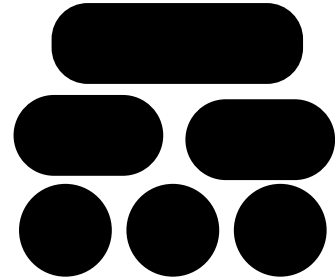
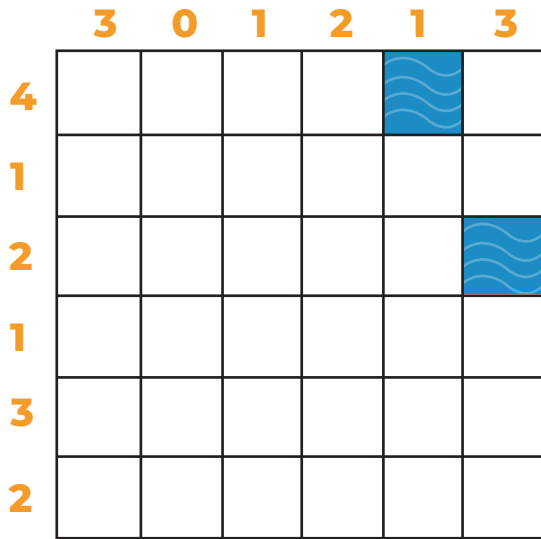
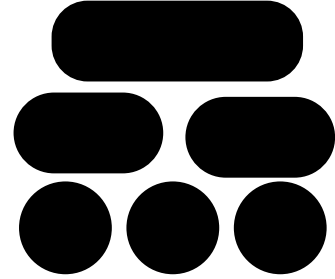
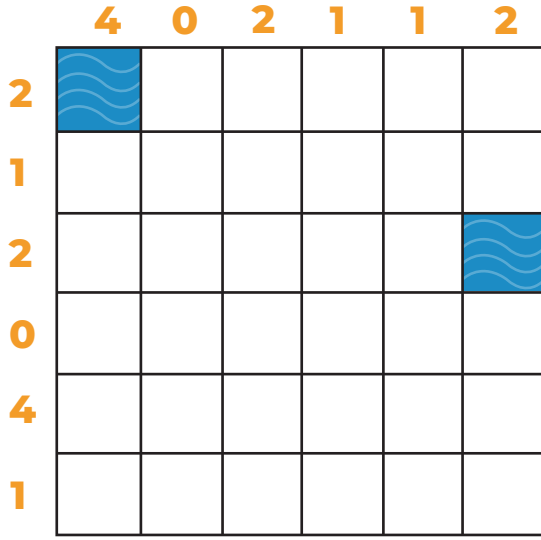
• Şimdi yukarıda verilen örnek amiral battı oyununu oynayalım.

Deniz olduğu için bu alanı boş bıraktık.



Bütün gemileri satır ve sütunların başındaki ipuçlarına göre oyun alanına yerleştirdik.

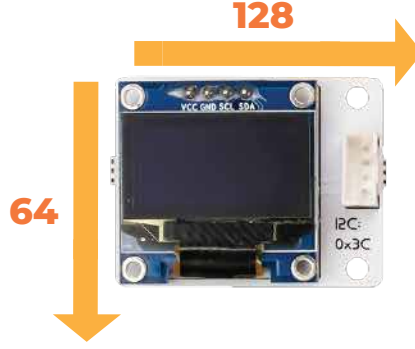
Amiral Battı Çalışma Kağıdı



(Bazı oyunlarda gemilerin bir kısmı ipucu olarak verilebilir.)

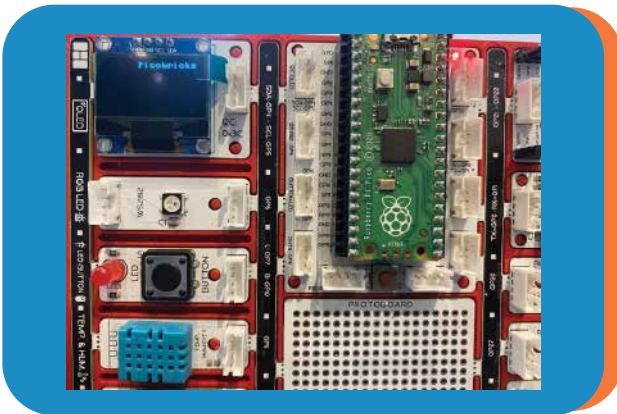
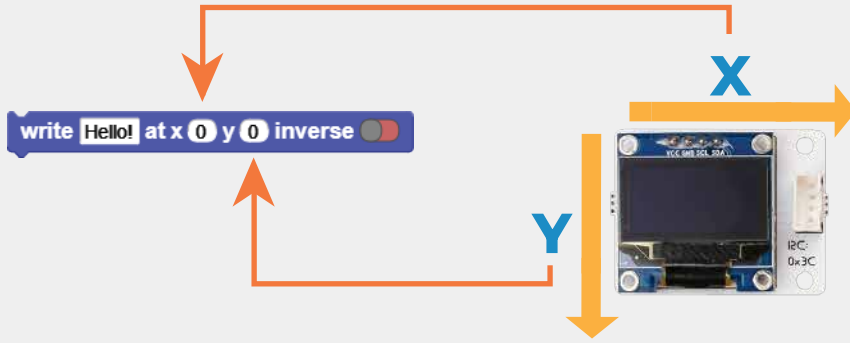
OLED Ekrandaki Koordinat Noktaları

Amiral battı oynayarak koordinatların nasıl isimlendirildiğini ve ne amaçlarla kullanıldığını öğrendiğimize göre artık picobricks üzerinde bulunan OLED ekran modülünün üzerindeki koordinat noktalarını daha rahat kullanabiliriz.

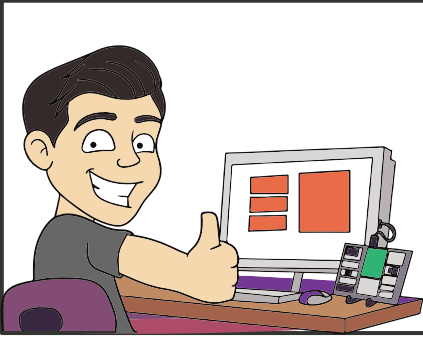


OLED ekran modülü üzerine yazılan yazının konumunu ayarlayalım.

Daha önceden öğrendiğimiz bilgilere göre OLED ekrana metinsel bir ifade yazmak için `write Hello! at x 0 y 0 inverse` bloğunu kullanıyoruz. Bu bloğun üzerinde bulunan x ve y değerleri ile oynayarak ekranda yazdırmak istediğimiz ifadenin konumunu değiştirebiliriz. X değeri OLED ekranın enini, Y değeri ise boyunu temsil eder.



Hadi şimdi X=45 , Y=0 koordinatlarına "Picobricks" yazdıralım.



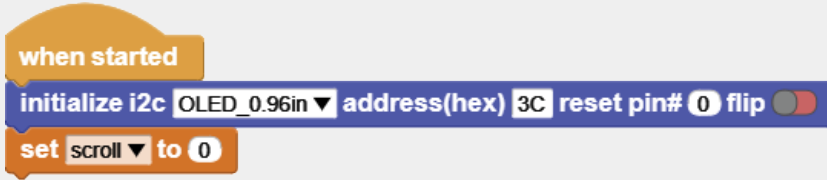
Şimdi Sıra Sizde!

Şimdi X=0, Y=25 koordinatlarına isminizi yazdırın.

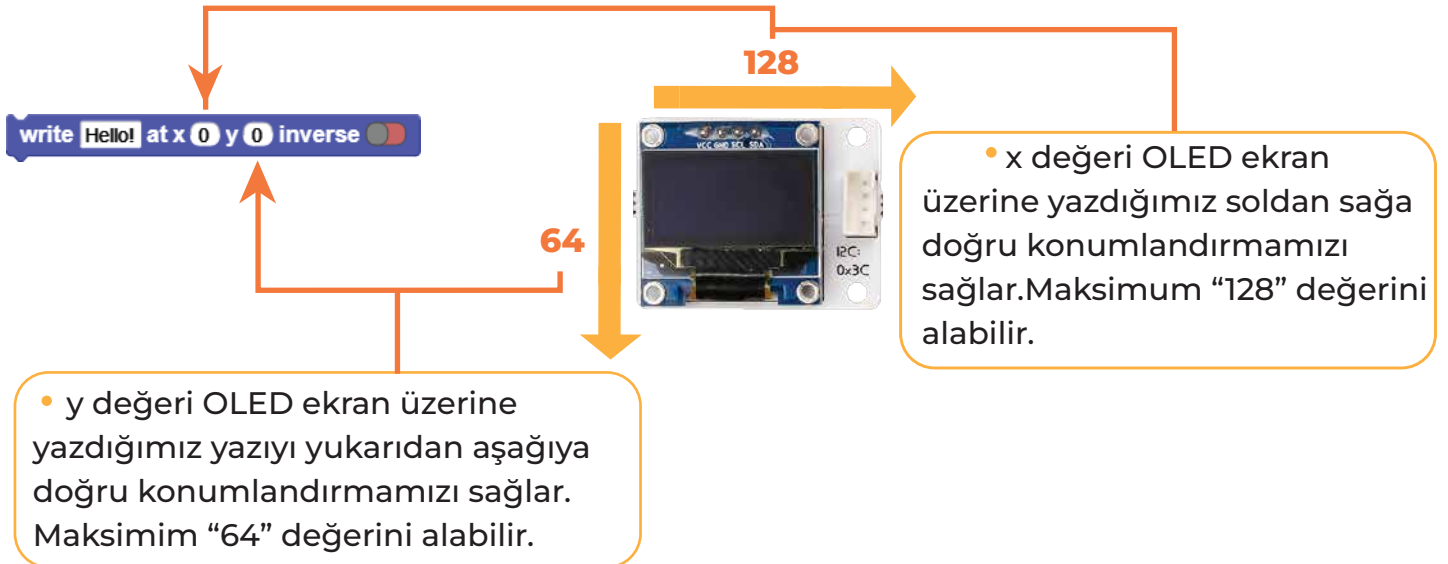
Yukarıdan Aşağıya Kayan PicoBricks Yazısı

Hadi şimdi "Picobricks" yazısını OLED ekranın en üstünden en altına kadar kaydıralım.

Projeyi ve OLED ekranı başlatmayı sağlayan blokları proje sayfamıza sürükledikten sonra "scroll" adında bir değişken tanımlayalım ve başlangıç değerini "0" yapalım.



Bundan sonraki kod bloklarını yazmadan önce OLED ekran özelliklerini aşağıdaki görsel şema üzerinden tekrar edelim.



“PicoBricks” yazısını, OLED ekran üzerinde 10’ ar birim aşağıya doğru kaydırmamızı sağlayan kod bloklarını yazalım.

Repeat döngü bloğunu proje sayfamıza sürükleyelim ve değerini “7” olarak belirleyelim. Ekranın yukarıdan aşağıya doğru maksimum alabileceği değer 64 olduğu için ve bizim döngümüzde `write Hello! at x 0 y 0 inverse` bloğundaki “y” değerinin alabileceği maksimum değer 60 olduğundan dolayı döngüyü yedi kere tekrar ettirdik. “Y” değerinin bulunduğu noktaya “scroll” değişkenini sürükleyelim ve ekranda yazacak ifadeyi “Picobricks” olarak değiştirelim.

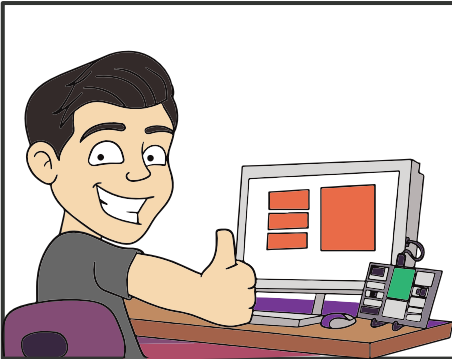
`write Picobricks at x 0 y scroll inverse`

50 ms bekleddikten sonra x değişkenine 10 ekleyelim ve ekranı her tekrarda temizleyelim.

```
repeat 7
  write Picobricks at x 0 y scroll inverse
  wait 50 millisecs
  change scroll by 10
  clear
```

Projemizin Kodları

```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set scroll to 0
  repeat 7
    write Picobricks at x 0 y scroll inverse
    wait 50 millisecs
    change scroll by 10
    clear
```



Şimdi Sıra Sizde!

“x” harfinin soldan sağa doğru kaymasını sağlayan kod bloklarını oluşturunuz.

Etkinlik

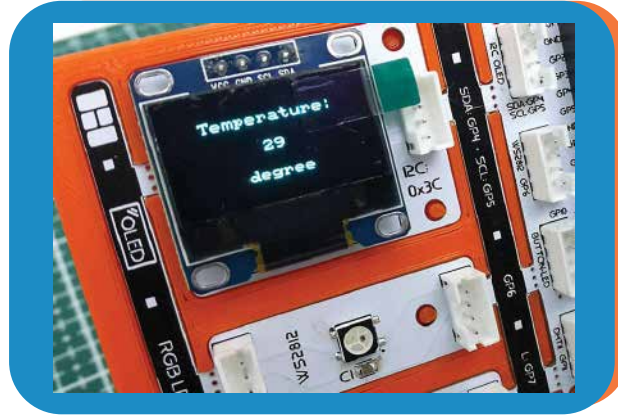
Aşağıda verilen kod bloklarının OLED ekran üzerinde nasıl bir çıktı oluşturacağını sınıfta tartışınız. Sonrasında kod bloklarını proje sayfanıza sürükleyerek PicoBricks üzerinde bulunan OLED ekrandaki değişimi gözlemleyiniz.

```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  repeat 10
    set scroll to 0
    repeat 7
      write Picobricks at x 0 y scroll inverse
    wait 50 milliseconds
    change scroll by 10
  clear
```

Proje

Hadi daha önceden yaptığımız termometre etkinliğini tekrardan yapalım ama bu sefer OLED ekran modülünde yazan ifadeleri biraz değiştirelim.

Daha önce yaptığımız termometre etkinliğinde sıcaklık ve nem sensöründen aldığımız değeri OLED ekran üzerinde X=0, Y=0 koordinatlarına yazdırıyorduk bu sefer “Temperature:”, Sıcaklık değeri ve “degree” ifadelerini de OLED ekranda yazdırarak aşağıdaki resimde verilen picobrickteki gibi bir çıktı elde edelim.

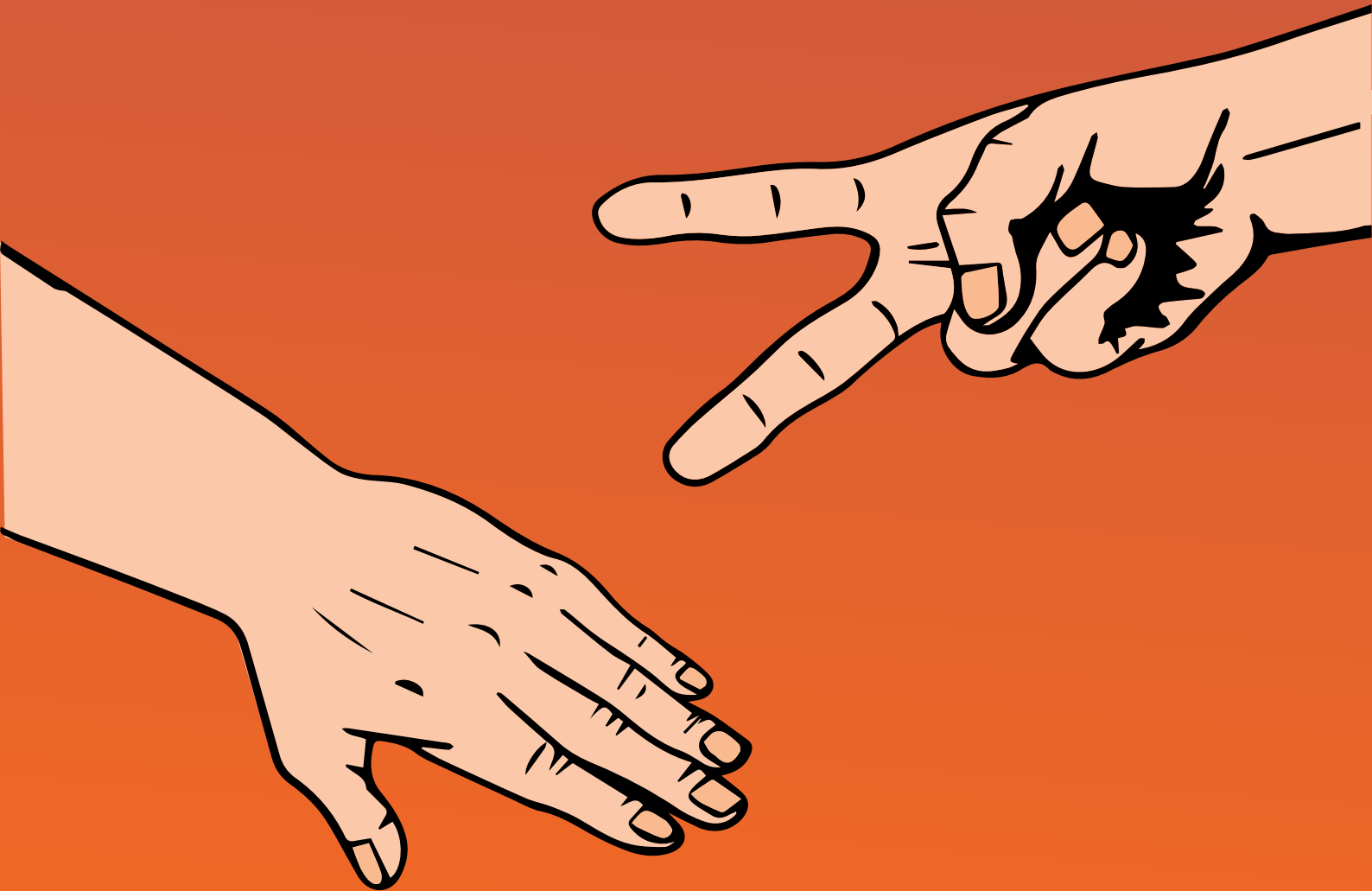


“Temperature” ifadesini X=15, Y=10, Sıcaklık ve nem sensöründen alınan sıcaklık değerini X=55, Y=30 ve “degree” ifadesini X=40, Y=50 koordinatlarına yazdıracağız.

```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  write Temperature: at x 15 y 10 inverse
  write PicoBricks temperature (°C) at x 55 y 30 inverse
  write degree at x 40 y 50 inverse
```

K-12 Uluslararası Bilgisayar Bilimi Standartları

- 1B-AP-12 3- 5 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.



PART 6



Bu Bölümde Öğreneceklerimiz


Bu bölümde öğrenci;

- Koşul ifadelerinden if - else yapısını
- Döngü ifadelerinden forever'ın yapısını ve
- Random değer oluşturmayı öğrenir.

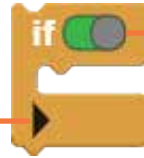
Koşul Yapılarına Giriş

IF-ELSE

If- else bir koşul yapısıdır. İstediğimiz bir durumun belirli bir koşul sonucunda gerçekleşmesini istiyorsak bu yapıyı kullanırız. Günlük yaşantımızda da bu ifadeleri sıklıkla kullanırız. Örneğin, çay demlemek için su kaynatıyoruz, su kaynadıktan sonra ocağın altını kapatmak, koşul gerektiren bir iştir. Suyun sıcaklığı 100 dereceyi geçerse ocağı kapatmamız gerekecektir. Programlama dillerinde de koşul gerektiren işlemleri gerçekleştirmek için bu yapıları kullanırız. Bu bölümde if-else ifadesini kullanarak

projeler geliştireceğiz. MicroBlocks'ta if - else için  bloğu kullanılır. Bu kod bloğunun kullanım amacı ve kullanımı aşağıdaki şemada anlatılmıştır.

Bu imlece tıkladığında else bloğu açılır. Else bloğu if bloğunun içerisindeki ifade gerçekleşmediği durumlarda yapılacak işleri belirler.



İstenilen koşul ifadesi bu alana getirilir.



Bu imlece tıkladığında if-else bloğu bir önceki haline geri döner.

İmlece tekrar tıkladığımız else if bloğu if ve else bloğunun arasında açılır. Else if bloğunun amacı if bloğundan farklı koşul durumlarını ve belirlenen koşullar gerçekleştikten sonra yapılacak işleri gerçekleştirmektir. Bu imlece ne kadar tıklarsak o kadar else if bloğu açarız.



IF-ELSE: Taş - Kağıt - Makas Etkinliği

Taş-kağıt-makas aktivitesi el ile oynanan bir oyundur. Taş-kağıt-makas seçimlerimize göre puan toplarız ve bu puanların sonucunda rakibimizi yenmeye çalışırız.

- Elimizi tam açtığımız zaman kağıt.
- Yumruk yaptığımız zaman taş.
- İşaret ve yüzük parmakları hariç diğer parmaklarımızı kapatarak oluşturduğumuz şekil ise makası temsil etmektedir.



Bu oyunda kazanan şu şekilde belirlenir;

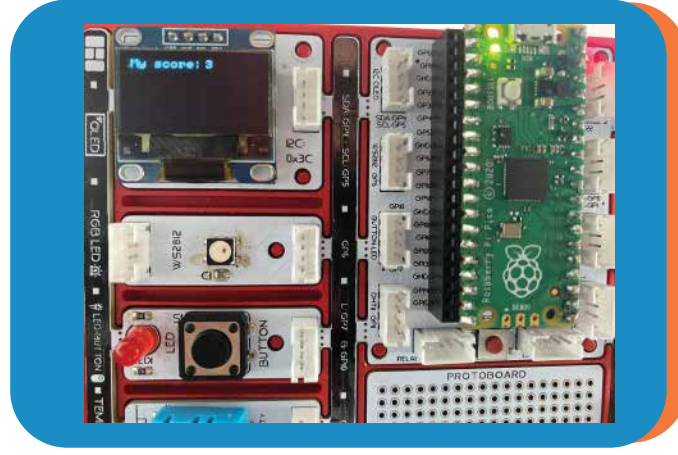
- Taş, makası kırar. Bu durumda taş hareketi yapan kişi makas hareketi yapan kişiyi yener.
- Makas, kağıdı keser. Bu durumda makas hareketi yapan bir kişi kağıt hareketi yapan kişiyi yener.
- Kağıt, taşı sarar. Bu durumda kağıt hareketi yapan bir kişi taş hareketi yapan bir kişiyi yener.

Hadi Şimdi 5 Kere Arkadaşımızla Taş-Kağıt-Makas Oynayalım

- Taş-kağıt-makas oyununu arkadaş / arkadaşlarınızla oynayalım ve skorlarımızı kağıda not edelim.
- Şimdi kendi skor tablomuzu picobricks ile oluşturalım ve tekrardan arkadaşınızla taş kağıt makas oynayarak skorunuzu picobricks'e not ediniz.
- Şimdi herkes kendi skorunu, picobricksine yazdırsın. Skorunuzu picobrickse kayıt etmek için aşağıdaki proje adımlarını takip ediniz.

Tek Kişilik Skor Etkinliği

Bu etkinliğin amacı öğrencinin picobricksi kullanarak oyunda elde ettiği puanları kayıt etmesidir. Öğrenci butona her bastığında kendi skor değeri bir artarak oled ekranda yazacaktır. Proje algoritmasını inceleyelim ve kod bloklarını proje sayfamıza sürükleyelim.



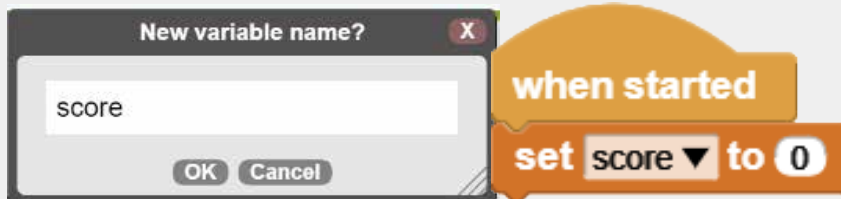
Proje Algoritması

1. Başla.
2. Skor değerinin başlangıç değerini "0" yap.
3. Eğer butona basıldıysa skor değerini 1 arttır ve OLED ekrana yaz.
4. Değilse skor değerinin o anki halini OLED ekrana yaz.
5. 3. maddeye dön.


Projenin MicroBlocks Kodları

Şimdi kod bloklarını sürüklemeye başlayalım.

"When started" bloğundan sonra "score" adında bir değişken tanımlayarak bu değişkenin başlangıç değerini "0" yapalım.



OLED ekranı kullanacağımız için OLED ekranı başlatan bloğunu sürükleyelim.

Bu adımdan sonra yapacağımız bütün işlemlerin program sonlanana kadar tekrar etmesi için bir döngü bloğu kullanacağız. Daha önce döngü ifadelerinden repeat bloğunu öğrenmiştik. Bu bölümle beraber  bloğunu öğreneceğiz. Forever bloğu


içerisine sürüklenen bütün kod bloklarının program sonlandırılana kadar devam etmesini sağlar.






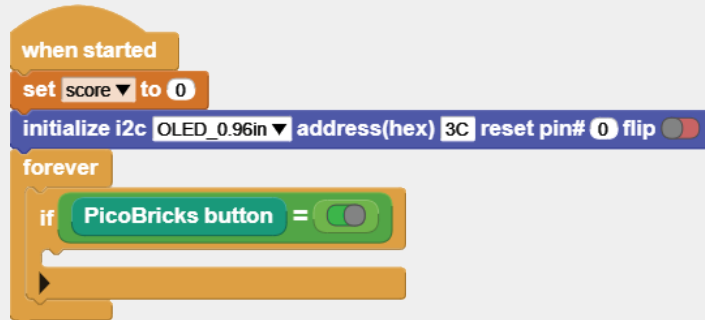
Artık forever bloğunun içerisinde program sonlandırılana kadar devam edecek ifadeleri belirleyelim.

Koşul ifadelerini proje sayfamıza sürüklemeye başlayalım. if bloğunu proje sayfamıza sürükleyelim. If bloğu koşul gerçekleşirse içerisindeki kod bloklarının çalıştırılmasını sağlar.

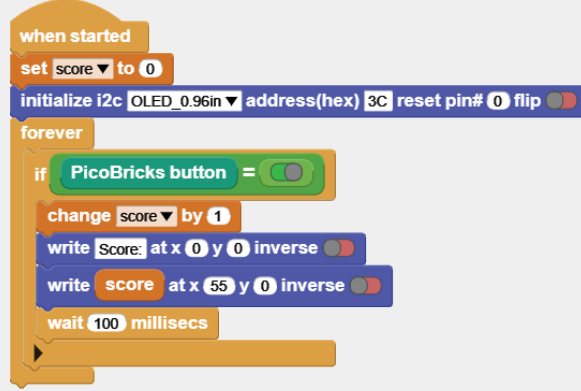


If bloğunun içerisindeki koşul ifadesini belirleyelim. Butona basıldığında score değişkenini bir kez artırıp ekrana yazdırmak için butona basıldığını anlamamızı sağlayan kod bloğunu oluşturalım ve if bloğunun içerisine sürükleyelim. 

Eşittir operatöründe eşitliğin sol tarafına PicoBricks bloklarından  bloğunu, sağ tarafına ise  bloğunu sürükleyelim  kod bloğunu oluşturduktan sonra if bloğunun içerisine sürükleyelim.

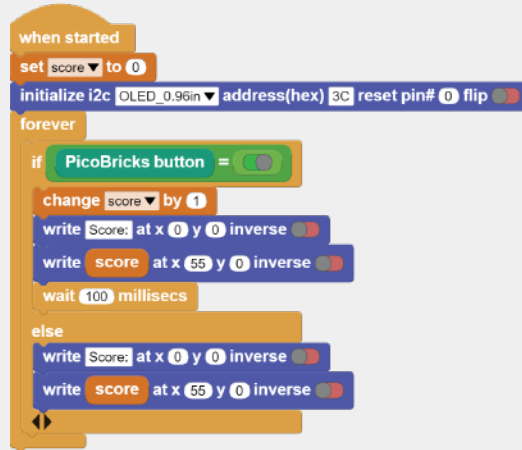


Koşul gerçekleşirse yani butona basıldığı durumlarda gerçekleşecek olayları belirleyelim . Butona basıldığında “score” değişkeninin bir artırıp daha sonra “Myscore” yazısıyla beraber ekrana yazılmasını istiyoruz bunu sağlayan kod bloklarını if bloğunun içerisine sürükleyeli sürükleyelim. Ekranda yazacak ifadeleri X ve Y koordinatlarıyla istediğimiz koordinat noktalarına yazdıralım.



```
when started
set score to 0
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
forever
if PicoBricks button =
change score by 1
write Score: at x 0 y 0 inverse
write score at x 55 y 0 inverse
wait 100 millisecs
```

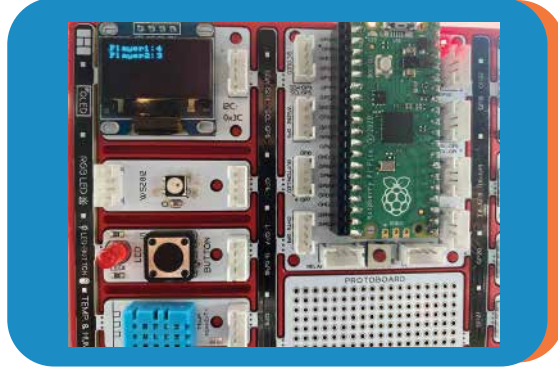
İf bloğunun altındaki imlece tıklayarak else bloğunu açalım. if bloğunun içerisindeki ifade gerçekleşmediği zamanlarda yani butona basılmadığı zamanlarda gerçekleşecek ifadeleri else bloğunun içerisine sürükleyelim. Butona basılmadığında ekrana o anki score değişkeni ve “Myscore” yazısının yazmasını istiyoruz.



```
when started
set score to 0
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
forever
if PicoBricks button =
change score by 1
write Score: at x 0 y 0 inverse
write score at x 55 y 0 inverse
wait 100 millisecs
else
write Score: at x 0 y 0 inverse
write score at x 55 y 0 inverse
```

Projemizi çalıştırıp taş- kağıt- makas oynamaya devam edebiliriz. Artık not defteri yerine Picobricks ‘i kullanabiliriz.

Ekstra Proje: İki Kişilik Skor Etkinliği



Daha önce yaptığımız skor etkinliğinde her öğrenci kendi Picobricksine kendi skorunu yazdırıyordu, şimdi tek picobricks kullanarak iki oyuncununda skorunu artırebileceğiz. Daha önceki etkinlikte “score” değişkenini arttırmak için buton modülünü kullanıyorduk. Bu yaptığımız etkinlikte iki oyuncuda aynı picobricksi kullanacağı için skoru arttırmak için iki farklı modül kullanmanız gerekecek. Bu etkinlikte 1.oyuncu LDR sensör modülünü 2. oyuncu ise buton modülünü kullanarak kendi skorlarını arttıracaklar.

Projenin Algoritması

1. Başla
2. player 1 değişkeninin başlangıç değerini “0” yap.
3. player 2 değişkeninin başlangıç değerini “0” yap.
4. Eğer Ortamın ışık miktarı küçükse 50’den player1 değerini 1 arttır ve ekrana yaz.
5. Eğer Butona basıldıysa player 2 değerini 1 arttır ve ekrana yaz.
6. 4. ve 5. maddedeki olasılıklar gerçekleşmediyse player1 ve player 2 değişkenlerinin o anki değerini ekrana yaz.
7. 4. maddeye geri dön.

Projenin MicroBlocks Kodları

Proje için gerekli kod bloklarını proje sayfamıza sürüklemeye başlayalım. Projede iki öğrenci tek picobricks kullanacağı için her öğrenci yani oyuncu için “player1” ve “player 2” değişkenlerini tanımlıyoruz ve bu değişkenlerin başlangıç değerlerini “0” olarak giriyoruz. Aynı şekilde yaptığımız diğer işlemlerin program sonlandırılana kadar devam etmesi için forever bloğunu da bu blokların hemen altına sürüleyelim. OLED ekranı kullanacağımız için OLED ekranı başlatma bloğunu da değişkenlerin hemen altında tanımlayabiliriz.



Şimdi koşul ifadelerimizi belirleyelim ve gerekli kod bloklarını sürükleyelim.

İki oyuncu iki farklı koşul ile skorlarını arttıracığı için bu projede if - else if ve else blokları kullanılacak. 1.oyuncu LDR sensörünün üstünü parmağıyla kapatarak değişken skor değişkenini arttıracak bunun için “eğer LDR sensörünün değeri küçükse 50’ den” cümlesini ifade eden kod bloğunu yazalım ve if değerinin içerisine sürükleyelim küçüktür operatöründe sol tarafa LDR sensörü bloğunu sürükleyelim sağ tarafa ise “50” değerini yazalım.

PicoBricks light sensor (0-100) % < 15

Sonrasında if bloğunun içerisinde “player1” değişkeninin değerini bir arttırarak OLED ekranda belirlenen koordinatlara “player 1:” ve player1 değişkenin değerlerini yazdırın ve bu işlem her gerçekleştiğinden sonra 250 millisaniye beklemek için “wait” bloğunu ekleyelim.

```

when started
  set player1 to 0
  set player2 to 0
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  forever
    if PicoBricks light sensor (0-100) % < 15
      change player1 by 1
      write Player1: at x 0 y 0 inverse
      write player1 at x 65 y 0 inverse
      wait 250 millisecs
    else if
    else
  
```

2.oyuncu ise butona basıldığında skor değişkenini arttıracığı için bu durumu ifade eden kod bloğunu oluşturarak else if bloğunun içerisine sürüklüyoruz.

PicoBricks button =

Sonrasında “player 2” değişkenini bir arttırarak OLED ekranda belirlenen koordinatlara player 2 değişkenini ve “Player 2:” yazısını yazdıralım ve 250 milisaniye bekleyelim . Else bloğunun içerisine ise bu iki koşul ifadesinin gerçekleşmediği durumlarda ekrana gelecek ifadeyi yazalım. Yani butona basılmadığı ve LDR sensörünün üstünün kapatılmadığı durumlarda OLED ekran üzerine player1 ve player 2 değişkenlerinin 0 anki değerlerini yazdıran kod bloklarını getirelim.

Artık projemizi çalıştırıp aynı PicoBricks ile skorlarımızı not edebiliriz.

```

when started
  set player1 to 0
  set player2 to 0
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  forever
    if PicoBricks light sensor (0-100) % < 15
      change player1 by 1
      write Player1: at x 0 y 0 inverse
      write player1 at x 65 y 0 inverse
      wait 250 millisecs
    else if PicoBricks button =
      change player2 by 1
      write Player2: at x 0 y 40 inverse
      write player2 at x 65 y 40 inverse
      wait 250 millisecs
    else
      write Player1: at x 0 y 0 inverse
      write player1 at x 65 y 0 inverse
      write Player2: at x 0 y 40 inverse
      write player2 at x 65 y 40 inverse
  
```

Taş - Kağıt - Makas MicroBlocks Projesi

Bu etkinliğin amacı öğrencinin PicoBricks'e karşı taş-kağıt-makas oyununu oynamasıdır. Ekran rastgele Taş-kağıt-makas değerlerinden bir tanesi gelir. Öğrenci ekrana gelen taş-kağıt-makas değerini yenmeye çalışır.

Taş kağıt makas oyununu PicoBricks'e karşı oynayalım.

1. Butona basın ve 3 saniye bekleyin.



2. Taş-Kağıt-Makas dan bir tanesi rastgele ekrana gelir.



3. Daha sonra oyunu tekrar başlatmak için ekrana tekrar "Butona basınız" uyarısı gelir.



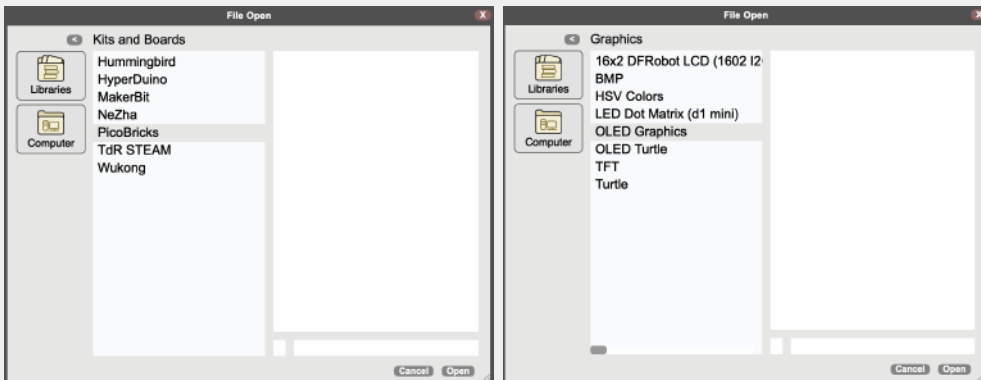
Proje Algoritması

1. Başla.
2. 100 ms aralıklarla ekrana "Rock","paper","scissors" yazdır.
3. Ekranı temizle ve ekrana "press", "the button","Wait the 3 secs" yazdır.
4. Butona basılınca
5. Ekranı temizle ve 3 saniye bekle
6. r_p_s adında bir değişken tanımla ve 1 ile 3 arasında random bir sayıyı bu değişkene ata.
7. Eğer sayı 1 olursa ekrana "Rock" yaz.
8. Eğer sayı 2 olursa ekrana paper yaz.
9. Eğer sayı 1 veya 2 olmazsa ekrana "scissors" yazdır.
10. Daha sonra r_p_s değişkenini sıfırla.
11. 3 saniye bekle ve butona basılırsa tekrar 1. maddeye dön.

Etkinliğin Microblocks Kodu

Projenin microblocks kodunun tamamı aşağıdaki resimde verilmiştir. Şimdi adım adım microblocks kodunu inceleyim.

1. Taş - Kağıt - Makas etkinliği için picobricks kütüphanesi ve OLED graphic kütüphanesini projemize yükleyelim.



2. İlk olarak program başlatıldığını ekranda belirtilen x - y koordinatlarına 100 ms aralıklarla Taş - Kağıt -Makas yazısı gelir ve daha sonra “Butona Basınız” ve “3 sn bekleyiniz” uyarısı gelir.

```

when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  wait 100 millisecs
  write Rock at x 40 y 30 inverse
  wait 100 millisecs
  clear
  wait 100 millisecs
  write paper at x 40 y 30 inverse
  wait 100 millisecs
  clear
  wait 100 millisecs
  write scissors at x 40 y 30 inverse
  wait 300 millisecs
  clear
  write Press at x 40 y 20 inverse
  write the button at x 26 y 30 inverse
  write Wait the 3 sec at x 10 y 60 inverse
  
```

3. Daha sonra picobricks üzerindeki butona tıkladıktan sonra ekrana rastgele Taş-Kağıt-Makas yazılarından birini getirmeyi sağlayan ve daha sonra programı yeniden başlatmak için butona tıklayınız uyarısını ekrana getiren kod bloklarını inceleyelim.

```

when PicoBricks button
  clear
  wait 3000 millisecs
  set r_p_s to random 1 to 3
  if r_p_s = 1
    write Rocks at x 10 y 20 inverse
  else if r_p_s = 2
    write paper at x 10 y 20 inverse
  else
    write scissors at x 10 y 20 inverse
  set r_p_s to 0
  wait 3000 millisecs
  write Press at x 40 y 30 inverse
  write the button at x 26 y 40 inverse
  
```

Şimdi bu kod bloklarını inceleyelim;

PicoBricks üzerinde bulunan butona tıkladığında ekranda daha önceden yazan ifadeleri silmek ve ekranı tekrardan başlatmak için aşağıdaki kod bloklarını oluşturunuz.

```

when PicoBricks button
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  clear
  
```

Şimdi daha önce öğrendiğimiz gibi `r_p_s` adında bir değişken tanımlayalım. Daha sonra bu değişkeni ekrana rastgele Taş-Kağıt-Makas yazılarından birini yazdırmak için kullanacağız. `r_p_s` değişkeninin ilk değerini 1 ile 3 arasında rastgele bir sayı yapmak için operatör bloklarından `random 1 to 10` bloğunun `set r_p_s to 0` içerisine sürükleyip “10” değerini “3” yapalım. `set r_p_s to random 1 to 10` Oluşturduğumuz bu blok sayesinde `r_p_s` değişkeni her butona basıldığında 1 ile 3 arasında rastgele birini alacaktır.

```

when PicoBricks button
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  clear
  wait 3000 millisecs
  set r_p_s to random 1 to 10
  
```

Şimdi koşul ifadelerine geçebiliriz. `r_p_s` değişkeni eğer “1” değerini alırsa ekrana “Roks”, 2 değerini alırsa “paper”, 1 veya 2 değerlerini almazsa yani 3 değerini alırsa “scissors” değerini yazıdran kod bloklarını oluşturmak için if, else if ve else bloklarını oluşturalım ve aşağıdaki resimde gösterildiği gibi koşul ifadelerinin içerisini dolduralım.

```

when PicoBricks button
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  clear
  wait 3000 millisecs
  set r_p_s to random 1 to 3
  if r_p_s = 1
    write Rocks at x 0 y 0 inverse
  else if r_p_s = 2
    write paper at x 10 y 20 inverse
  else
    write scissors at x 10 y 20 inverse
  
```

Bu bloklardan hemen sonra oyunu yeniden oynamak için `r_p_s` değişkeninin değerini tekrar “0” apalım. Üç saniye bekledikten sonra ekrana tekrar “press” ve “the button” ifadelerini yazdıralım.

Projemizi çalıştırıp PicoBricks’e karşı taş-kağıt-makas oynayabiliriz.

```

when PicoBricks button
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  clear
  wait 3000 millisecs
  set r_p_s to random 1 to 3
  if r_p_s = 1
    write Rocks at x 0 y 0 inverse
  else if r_p_s = 2
    write paper at x 10 y 20 inverse
  else
    write scissors at x 10 y 20 inverse
  set r_p_s to 0
  wait 3000 millisecs
  write Press at x 40 y 30 inverse
  write the button at x 26 y 40 inverse
  
```

Öğretmene Öneri

- Yukarıda verilen adımlar öğrencilerle ile beraber gerçekleştirilmelidir.
- Değişken tanımlama aşaması anlatılırken daha önce öğrenilen bilgilerden yararlanarak öğrencilerin değişkeni yardım almadan tanımlaması istemelidir. Bu süreçte zorlanan öğrencilere yardım edilebilir.
- If-else if ve else yapısı öğrencilere aktarılır.
- Diğer adımlar öğretmenle beraber yapılmalı ve öğretmenin her adımın mantığını öğrencilere aktarılması gerekmektedir.
- Öğrenciler projeyi tamamladıktan sonra picobricksle taş - kağıt - makas oyununu kısa bir süre oynayabilir.

K-12 Uluslararası Bilgisayar Bilimi Standartları

- 2-AP-10 6 - 8 Use flowcharts and/or pseudocode to address complex problems as algorithms.
- 3A-AP-13 9-10 Create prototypes that use algorithms to solve computational problems 3B-AP-10 11-12 by leveraging prior student knowledge and personal interests.
- 3B-AP-13 11-12 Use and adapt classic algorithms to solve computational problems.
- 3B-AP-13 11-12 Illustrate the flow of execution of a recursive algorithm.

wait until PicoBricks button

reset timer

for i in 10

PART 7

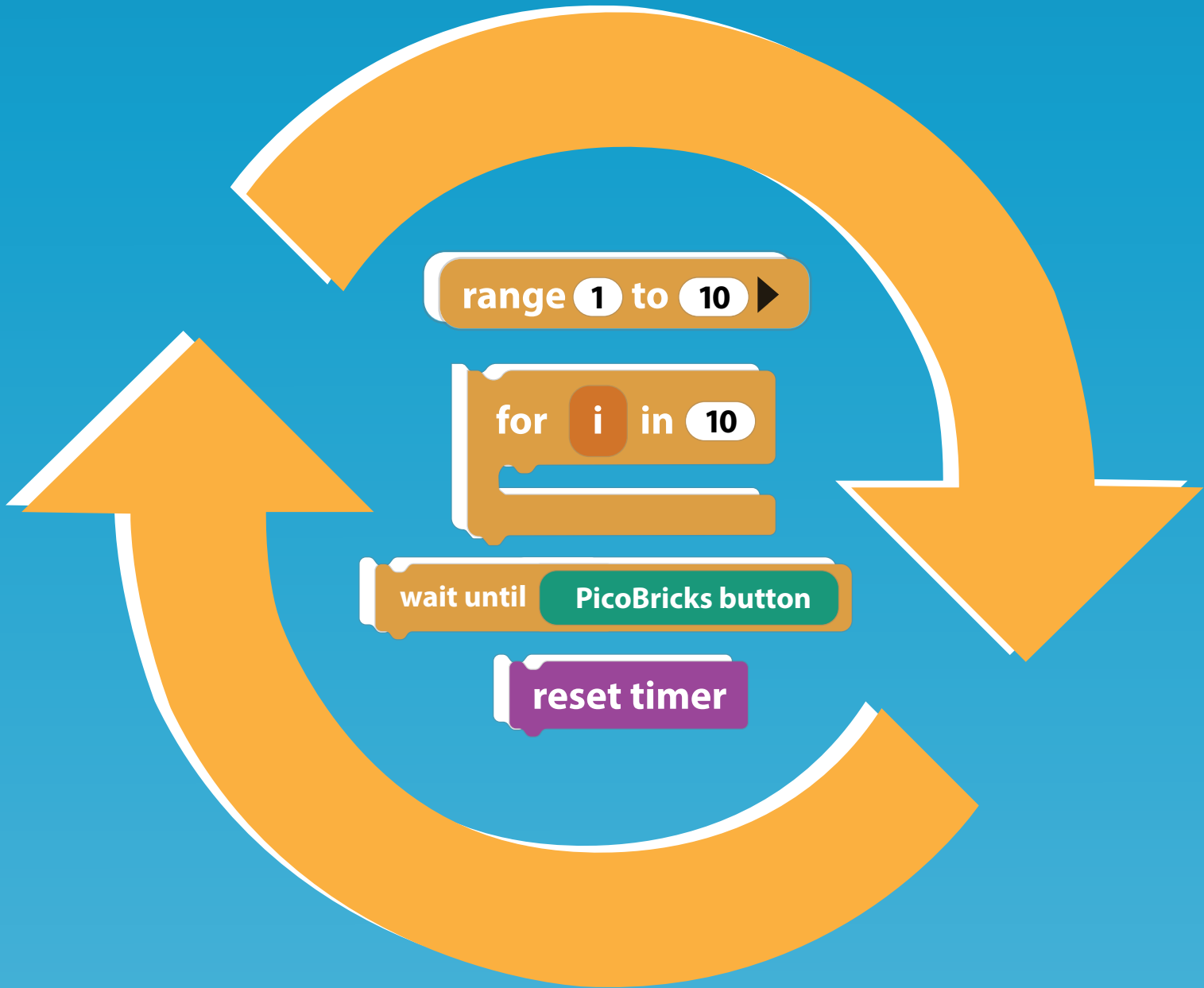
range 1 to 10

range 1 to 10

for i in 10

wait until PicoBricks button

reset timer



Bu Bölümde Öğreneceklerimiz:

Bu bölümde öğrenci;

- For döngüsünün kullanım amacını
- Range ifadesinin kullanımını
- Join bloğunu kullanmayı
- Timer ve reset timer bloklarının kullanmayı öğrenir.

For Döngüsü

Bu bölümde döngü türlerinden biri olan for döngüsünü kullanacağız. Daha önceki bölümlerde döngülerin ne olduğunu, hangi amaçla kullanıldığını öğrendik ve Repeat döngüsünü kullanarak projeler geliştirdik. For döngüsü ve repeat döngüsü arasındaki farkı anlayarak başlayalım.

Repeat döngüsünde döngünün başına kaç kere tekrar edeceğini, içerisinde ise döngüde hangi olayların gerçekleşeceğini belirliyoruz.

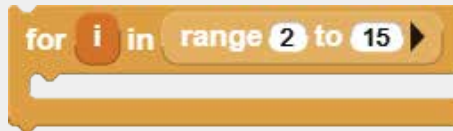
For döngüsünde, artık başlangıç değerimizi döngünün dışında tanımlamamıza gerek kalmadan, Döngü içerisinde, döngünün bitiş değerini de for bloğunun içerisinde tanımlayabiliyoruz.

For döngüsü içerisinde “i” değişkeni tanımlı olarak gelir bu i değişkeninin başlangıç değeri “0”dır. Döngünün bitiş değeri ise “10” olarak tanımlanmıştır fakat döngünün bitiş değerini istediğimiz gibi değiştirebiliriz.

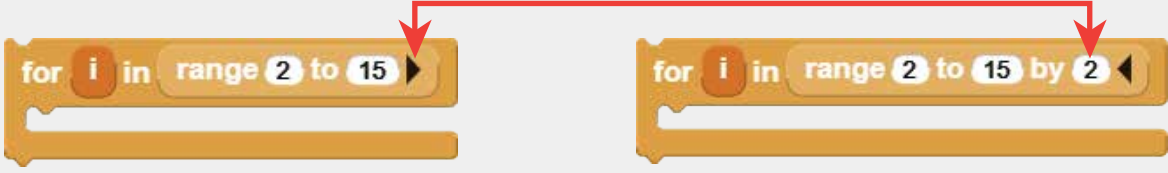


Şimdi for döngüsüyle beraber kullanılan `range 1 to 10` bloğunu tanıyalım.

“Range” bloğu sayesinde for döngüsünün başlangıç, bitiş ve artış miktarı değerlerini belirleyebiliriz. Aşağıda verilen for döngüsünü başlangıç değeri “2” ve bitiş değeri “15”dir. Artış miktarı ise “1”dir.



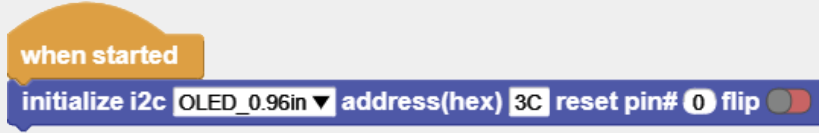
Range bloğunun sağındaki ok işaretine tıklayarak artış miktarını değiştirebiliriz. Oku tıkladıktan sonra açılan pencereye artış miktarını girmeliyiz. Aşağıda verilen for döngüsünün, başlangıç değeri 2, bitiş değeri 15, artış miktarı ise 2 dir.



For Döngüsü İle Sayaç (1-10)

Şimdi for döngüsünü kullanarak OLED ekrana 1'den 10'a kadar olan sayıları yazdıran kod bloklarını oluşturalım.

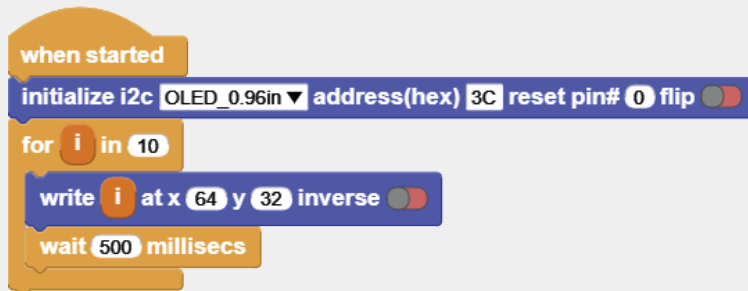
OLED ekranı kullanacağımız için "When started" bloğunu ve OLED ekranı başlatma bloğunu proje sayfamıza sürükleyelim.



For döngüsünü proje sayfamıza sürükleyelim. Başlangıç değerimiz "1" bitiş değerimiz ise 10 olduğu için bu değerleri değiştirmemize gerek yok çünkü "i" değişkeninin başlangıç değeri "1" dir for döngüsüyle beraber otomatik olarak bitiş değeri de "10" olarak belirlenmiştir.



Ekrana yazdıracağımız değer "i" değişkeni olduğu için "write" bloğunun içerisine i değerini girelim x ve y koordinatlarını ekranın ortası olarak belirleyelim (x=64 Y=32) Daha sonra ekrandaki sayı değişimlerini gözlemleyebilmek için yarım saniye bekleme bloğunu ekleyelim. "i" değişkenini for döngüsünün içerisinden sürükleyerek alabiliriz.



For Döngüsü İle Sayaç (10-1)

Şimdi 10'dan 1'e kadar olan sayıları 1'er 1'er azaltarak ekrana yazdıralım. Proje sayfamızda bulunan daha önceki etkinlikte değişiklikler yaparak bu etkinliği hazırlayabiliriz.

İlk olarak 10'dan geriye doğru yazdıracağımız için başlangıç değerimizi "10" olarak belirlememiz gerekiyor fakat "i" değişkeninin başlangıç miktarını değiştiremediğimiz için **range 1 to 10** bloğunu for döngüsünün içerisine sürükleyelim başlangıç değerini "10" bitiş değerini ise "1" yapalım. Daha sonra OLED ekrandaki değişimi daha iyi takip edebilmek için yarım saniye bekleyelim ve her seferinde ekranı temizleyelim. Ekranı **clear** temizlemek için bloğunu kullanacağız.

```

when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  for i in range 10 to 1
    write i at x 64 y 32 inverse
    wait 500 millisecs
    clear
  
```

SORU

Proje sonlandırıldığında ekranda "1" değerini görmek için **clear** bloğunu nereye koymamız gerekir ?

CEVAP

For Döngüsü İle 2'şer 2'şer artan sayaç (2-10)

2 ile 10 arasındaki sayıları 2'şer 2'şer arttırmak için başlangıç değerini 2 bitiş değerini 10 artış miktarını ise 2 olarak belirlemeliyiz. Bu değerleri **range 2 to 10 by 2** bloğunu for döngüsünün içerisine girerek sağlayabiliriz. Daha sonra ekrana yazacak ifadeleri belirleyerek kodumuzu tamamlayalım

```

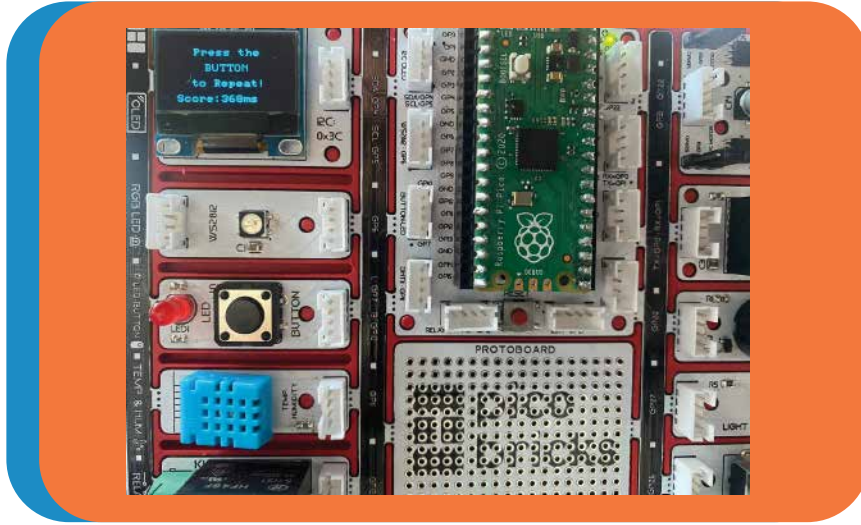
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  for i in range 2 to 10 by 2
    clear
    write i at x 64 y 32 inverse
    wait 500 millisecs
  
```

For Döngüsü ile 2'şer 2'şer azalan sayaç (10-2)

10 ile 2 arasındaki sayıları ekrana yazdırmak için sadece range bloğunda değişiklik yaparak tamamlayabiliriz. Başlangıç değerimiz "2" bitiş değerimiz "10" artış miktarımız ise "2" olacak. **range 10 to 2 by 2** bloğunu oluşturduktan sonra for bloğunun içerisine sürükleyin ve OLED ekranda yazdırmak için gerekli kod bloklarını oluşturun. (Bu kod bloklarını bir önceki etkinlikten kopyalayabilirsiniz.)

```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  for i in range 10 to 2 by 2
    clear
    write i at x 64 y 32 inverse
    wait 500 millisecs
```

Bölüm Projesi (Show Your Reaction)

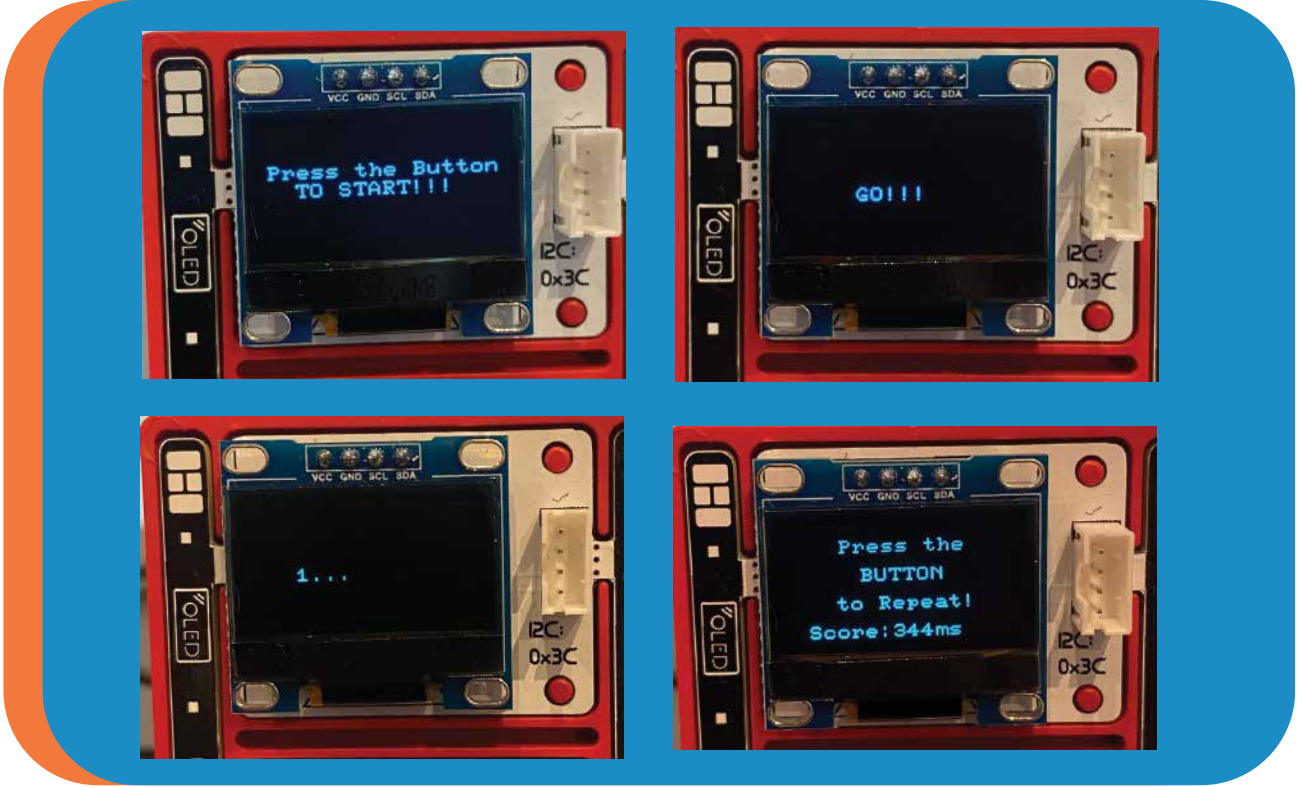


Tepkini göster (Show Your Reaction) projesi, kullanıcının refleksini ölçmeyi amaçlamaktadır. Kullanıcı butona bastığında 3 saniye bekleme süresinden sonra 1 ile 5 saniye aralığında rastgele bir zamanda kırmızı LEDi çalıştırarak bir uyarı verir. Kullanıcının bu uyarıyı aldıktan sonra en kısa zamanda butona basarak tepki göstermesi istenir. Butona bastıktan sonra tepki süresi OLED ekran üzerinde gösterilir ve buzzer çalışır.

Proje Algoritması

1. Başla
2. Butona basılana kadar bekle
3. Üç saniye bekle
4. 1-5 arasında rastgele bir saniyede kırmızı LED'i yak ve zamanı sıfırla
5. Tekrar butona basıldığında
6. Zamanı skora eşitle.
7. Skoru ekrana yaz ve buzzer'ı çalıştır.
8. Butona tekrar basıldığında 1.maddeye dön.

Proje Görselleri



Proje Videosu

Projenin videolu anlatımı için Picobricks.com ve Robotistan INC YouTube kanalını ziyaret edebilirsiniz.



Projenin MicroBlocks Kodu

Hadi şimdi projenin microblocks kodlarını inceleyelim.

Proje başladığında OLED ekran üzerinde “Press the button” ve “TO START!” ifadelerini yazdırmak için aşağıdaki kod bloklarını proje sayfanızda oluşturunuz.

```

when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  write Press the Button at x 0 y 20 inverse
  write TO START!!! at x 15 y 30 inverse
  
```

Butona bastıkta sonra çalışacak kod bloklarını yazmak için **when PicoBricks button** bloğunu oluşturun.

Butona basıldıktan sonra oyunun başlaması için gereken üç saniyeyi ekranda 3 den geriye sayarak gösteren kod bloğunu oluşturalım;

Bu bölümde öğrendiğimiz bilgileri kullanarak ekranda 3 den geriye doğru 1'er 1'er azalan for bloğunu oluşturacağız. Range bloğunu kullanarak for döngüsünün başlangıç değerini 3, bitiş değerini 1 artış miktarını da 1 olarak belirleyelim. **range 3 to 1** bloğunu oluşturmamız gerekiyor. Oluşturduğumuz bu range bloğunu for döngüsünün içerisine sürükleyiniz.

```

for i in range 3 to 1
  
```

For döngüsünü kullanarak ekranda 1 saniye aralıklarla sırasıyla “3...”, “2...”, “1...” yazdıracağımız için ilk önce ekran o anki yazan bütün ifadeler temizlememiz gerekmektedir. **clear** bloğunu for döngüsünün içerisine sürükleyiniz. Ekranda “i”değişkeni ve “...” ifadesini yazdırmak için OLED Graphic kütüphanesinden “Write” bloğunu kullanacağız. Bu zamana kadar OLED ekranda “Write” bloğunu kullanarak hep bir ifade yazdırmıştık ama bu projede tek seferde hem “i” değişkeni hemde “...” ifadelerini yazdırmak istiyoruz. Bunun için “Data” bloklarından **join micro blocks** (Sağ taraftaki ok tuşuna basarak ekranda yazacak ifade sayısını arttırabiliriz.) bloğunu kullanacağız. Bu blok üzerinde ilk alana i değişkenini sürükleyelim ve ikinci alana “...” yazalım daha sonra “write bloğunun içerisine sürükleyiniz.”

bloğunu oluşturduktan sonra for döngünün içerisine sürükleyin ve bir saniye beklemek için “wait” bloğunu hemen bu bloğun arkasına sürükleyin.

```

when PicoBricks button
  for i in range 3 to 1
    clear
    write join i ... at x 25 y 35 inverse
    wait 1000 millisecs
  
```

Ekranı 3'ten geriye saydırıp 3 saniye bekledikten sonra tekrardan "clear" bloğu ile ekranı temizliyoruz ve ekranda "GO!!!" ifadesini yazdırıyoruz.

clear
write GO!!! at x 35 y 35 inverse

bloklarını oluşturup for döngüsünün altına sürükleyelim.

Ekranı 1 ile 5 saniye aralığında rastgele bir zamanda bekledikten sonra kırmızı LED'i yakmak için

wait random 1000 to 5000 millisecs
PicoBricks set red LED

```

when PicoBricks button
  for i in range 3 to 1
    clear
    write join i at x 25 y 35 inverse
    wait 1000 millisecs
  clear
  write GO!!! at x 35 y 35 inverse
  wait random 1000 to 5000 millisecs
  PicoBricks set red LED
  
```

Bu bloklardan sonra butona basılmasını ve basıldıktan sonra Kırmızı LED'in yandığı andan söndüğü ana kadar geçen zaman dilimini ekrana yazdırmasını istiyoruz.

Microblocks'da bir programı başladığımızdan itibaren projede durduğumuz anı kaydeden bir zamanlayıcı çalışmaktadır ve bizim bu zaman dilimini projelerimizde kullanmamız için "input" blokları içerisinde **timer** bloğu bulunur. Bu timer bloğunun değerini ise **reset timer** bloğuyla sıfırlayabiliriz.

Şimdi **reset timer** bloğunu kullanarak o zamana kadar geçen zamanı sıfırlayalım ve butona basılana kadar beklememizi sağlayan kod bloklarını oluşturalım.

wait until bloğu bir koşullu bekleme bloğudur. İçerisine sürüklediğimiz değer ile o koşul gerçekleşene kadar programın durmasını sağlar. Biz butona basılana kadar beklemek istediğimiz için içerisine Picobricks kütüphanesinden buton bloğunu sürükleyelim. **wait until PicoBricks button** bloğunu oluşturduktan sonra "reset timer" bloğunun altına sürükleyin.

```

reset timer
wait until PicoBricks button
  
```

Şimdi bir score değişkeni oluşturalım ve zamanlayıcının yeni değerini bu “score” değişkenine atayalım.”score” değişkenini daha sonra ekrana yazdırmak için kullanacağız.

set score to timer bloğunu oluşturarak “score” değişkenine o anki zamanı tanımlayalım. Kırmızı LED’i kapattıktan sonra buzzer’ı çalıştıralım. Daha sonra ekrana aşağıdaki resimde verilen ifadeleri yazdıralım.

```

when PicoBricks button
for i in range 3 to 1
clear
write join i at x 25 y 35 inverse
wait 1000 millisecs
clear
write GO!!! at x 35 y 35 inverse
wait random 1000 to 5000 millisecs
PicoBricks set red LED
reset timer
wait until PicoBricks button
set score to timer
PicoBricks set red LED
PicoBricks beep 200 ms
clear
write Press the at x 28 y 8 inverse
write BUTTON at x 40 y 24 inverse
write to Repeat! at x 26 y 40 inverse

```

Kodumuzu tamamlamak için “Score: ” yazısını, “score” değişkenini ve “ms” yazısını Join bloğunu kullanarak ekrana yazdıralım.

```

write join counter th digit at x 25 y 0 inverse

```

Kod bloklarımız hazır. Çalıştırabiliriz.

```

when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
write Press the button at x 0 y 10 inverse
write TO START! at x 25 y 35 inverse

```

```

when PicoBricks button
for i in range 3 to 1
clear
write join i at x 25 y 35 inverse
wait 1000 millisecs
clear
write GO!!! at x 35 y 35 inverse
wait random 1000 to 5000 millisecs
PicoBricks set red LED
reset timer
wait until PicoBricks button
set score to timer
PicoBricks set red LED
PicoBricks beep 200 ms
clear
write Press the at x 28 y 8 inverse
write BUTTON at x 40 y 24 inverse
write to Repeat! at x 26 y 40 inverse
write join Score: score ms at x 10 y 56 inverse

```

K-12 Uluslararası Bilgisayar Bilimi Standartları

- 2-AP-10 6-8 Use flowcharts and/or pseudocode to address complex problems as algorithms.
- 3A-AP-13 9 - 10 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
- 2-AP-16 6-8 Incorporate existing code, media, and libraries into original programs, and give attribution.
- 3A-AP-23 9-10 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.
- 3B-AP-22 11 - 12 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
- 3B-AP-24 11-12 Compare multiple programming languages and discuss how their features make them suitable for solving different types of problems.

BONUS PROJE



Akinatör Oynayalım



Akinatör, sorduğu sorular ile aklımızdaki gerçek ya da hayali bir karakteri tespit etmeyi sağlayan bir oyundur. Bu oyun sorduğu sorular ile veri tabanında sakladığı olasılıkları birer birer eleyerek düşündüğünüz karakteri tespit eder. Akinator oyunu sorduğu sorulara aldığı cevaplar ile veritabanını sürekli güncellenmektedir. Bu şekilde oyun yapay zekanın temellerini atmaktadır.

Hadi Akinator Oynayalım

Akinator oyununu hem mobil uygulamasından hemde web sitesinden oynayabilirsiniz.

Sınıftaki her öğrenci akinator oyunun kendi bilgisayarından bir kaç kez oynamasını isteyin. Oyunu oynadıktan sonra Öğretmen öğrencilere oyunla ilgili aşağıdaki sorulardan bir kaçını sorarak, oyunun mantığının öğrenciler tarafından daha iyi kavranması sağlayabilir.



Öğretmene Öneri

- Sizce Akinator karakterinizi nasıl tahmin edebiliyor?
- Akinatör sorduğu soruları nasıl bir mantığa göre seçiyor olabilir?
- Akinatörün yerinde olsaydın kendine nasıl bir soru sorardın?

Sayı Tahmin Oyunu

Sayı tahmin oyunu projesi, kızılötesi sensör ve kumanda yardımıyla PicoBricks'i yönlendirerek picobricksin, 1-128 aralığında tuttuğunuz bir sayıyı bulmasını sağlar.

Butona bastıktan sonra Picobricks ilk tahmini yapar ve oyuncunun, kumandanın "yukarı", "aşağı" ve "ok" tuşunu kullanarak yönlendirmesini bekler.

PicoBricks'in tahmini;

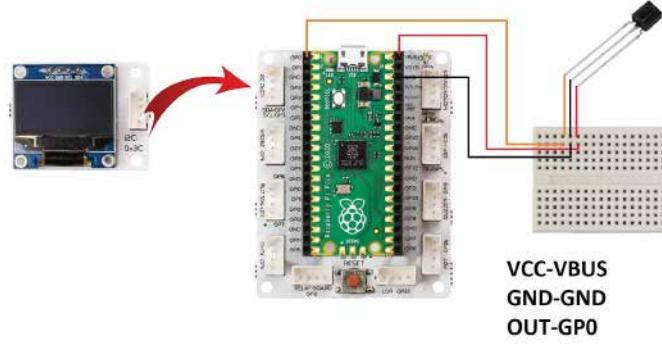


- Tuttuğunuz sayıdan büyükse,
- Tuttuğunuz sayıdan küçükse,
- PicoBricks tuttuğunuz sayıyı doğru tahmin ettiyse basınız.

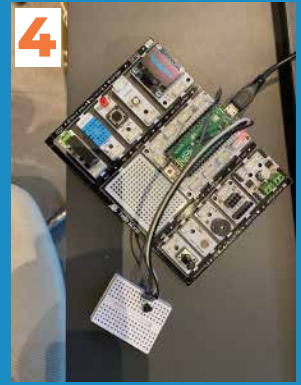
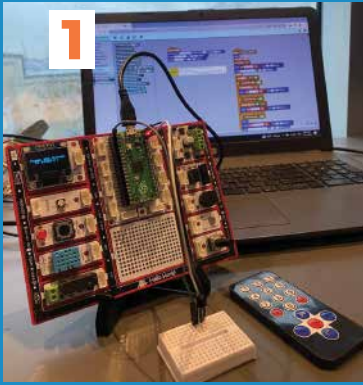
Projenin Algoritması

1. Butona basıldığında başla
2. Max tahmin değerini 2'ye böl.
3. Oyuncu yukarı ok tuşuna basarsa yani PicoBricks'in tahmini, oyuncunun tuttuğu sayıdan küçükse tahmin aralığını tekrar 2'ye böl ve o anki PicoBricks tahminine ekle.
4. Oyuncu aşağı ok tuşuna basarsa yani PicoBricks'in tahmini, oyuncunun tuttuğu sayıdan büyükse tahmin aralığını tekrar 2'ye böl ve o anki picobricks tahmininden çıkart.
5. 2. maddeye dön.
6. Oyuncu eğer "ok" tuşuna basarsa oyunu bitir ve PicoBricks'i tebrik et.

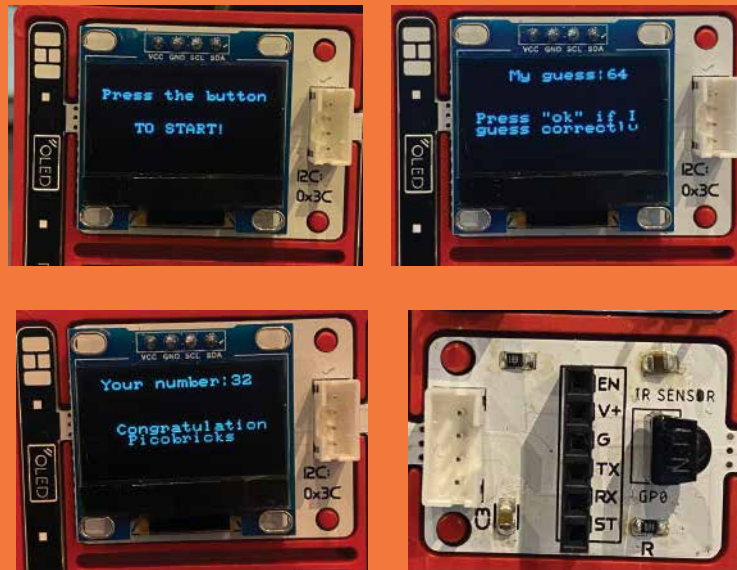
Projenin Pin Diyagramı (PicoBricks v1.0)



Projenin Görselleri (PicoBricks v1.0)

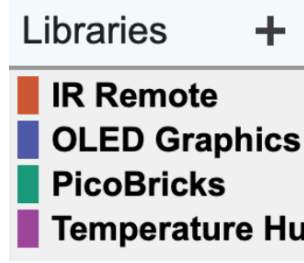


Projenin Görselleri (PicoBricks v1.2)

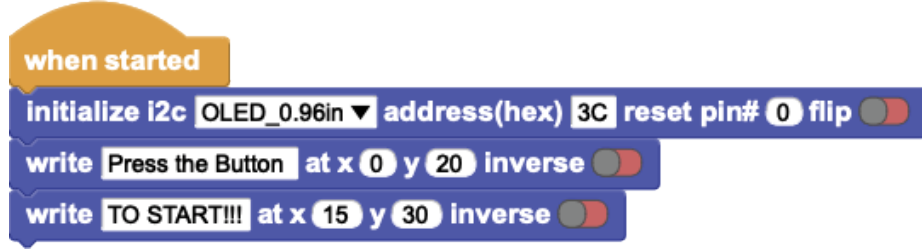


Projenin MicroBlocks Kodları

MicroBlocks online editörüne “IR Remote” kütüphanesini yükleyiniz.

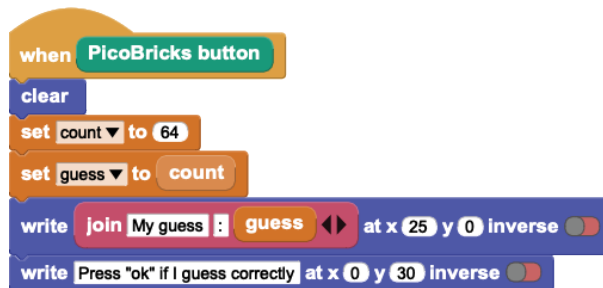


İlk olarak proje başlatıldığında “Press the button” ve “TO START!!!” ifadelerini OLED ekran üzerinde belirtilen konumlara yazdırınız.



Butona basıldığında;

- OLED ekranda daha önceden bulunan bütün ifadeleri silin.”count” adında bir değişken tanımlayarak başlangıç değerini 64 olarak belirleyin.
- “count” değişkeni sayı aralığını belirler.
- sonrasında “guess” adında bir değişken tanımlayarak bu değişkenin başlangıç değerini “count” değişkeninin ilk değerine eşitleyelim.
- Join bloğunu kullanarak “My guess”,“:” ve “guess” değişkenini ekranda belirtilen x ve y koordinatlarına yazdırınız.
- Ekranda belirtilen x ve y koordinatlarına “Press "ok" if I guess correctly” ifadesini yazdıralım.



Bu maddeye kadar kod bloklarını proje sayfamıza sürüklediğimize göre artık döngülere geçebiliriz.

Forever döngüsünü proje sayfanıza sürükleyiniz. Bundan sonra yaptığımız bütün işlemler sonsuza kadar devam edecektir.

Forever bloğunun içerisinde koşul ifadelerini belirleyeceğiz. Bunun için ilk önce kızılötesi sensör ile kumanda arasındaki iletişimi anlamaya çalışalım. Bunun için aşağıdaki tabloyu inceleyiniz.

Projemizde kumanda da bulunan 3 farklı tuşu kullanacağız. Bu tuşların her biri kızılötesi sensöre farklı değerler göndermektedir bu değerleri kullanarak koşul ifadelerini belirleyeceğiz.

Aşağıdaki üç farklı kumanda tuşu **receive IR code** bloğuna 3 farklı değer gönderir.



- Aşağı ok tuşu 82 değerini gönderir.
- Yukarı ok tuşu 24 değerini gönderir.
- OK tuşu 28 değerini gönderir.

Projemizde kullanacağımız üç tuş için 3 farklı if bloğu kullanacağız. Bu if bloklarından ilkinin koşulu kızıl ötesi sensöre kumanda yardımıyla “24” değeri gelirse olacak bu koşul ifadesini **receive IR code = 24** bloğunu oluşturarak sağlayacağız. Bu koşul sağlandığında,

“count” değişkeninin 2 ye bölme ve guess değişkenini count değişkeni kadar arttıralım. Sonra guess değişkenini ekrana yazdıralım.

```
if receive IR code = 24
  set count to count / 2
  change guess by count
  write join My guess : guess at x 25 y 0 inverse
```

İkinci if bloğunda ki koşul ifademiz, kızıl ötesi sensöre kumanda yardımıyla “82” değeri gelirse yani oyuncu aşağı ok tuşuna basarsa olacak bu koşul ifadesini **receive IR code = 82** bloğunu oluşturarak sağlayacağız. Bu koşul sağlandığında,

“count” değişkenini yarıya böleceğiz ve “guess” değişkeninden “count” değişkenini çıkaracağız. Daha sonra guess değişkenini ekrana yazdıralım.

```

if receive IR code = 82
  set count to count / 2
  change guess by 0 - count
  write join My guess : guess at x 25 y 0 inverse

```

Üçüncü koşul ifademiz, kızıl ötesi sensöre kumanda yardımıyla “28” değeri gelirse yani “OK” tuşuna basıldıysa olacak. Bu koşul sağlandığında, Ekranda yazan bütün ifadeleri temizleyeceğiz ve “Your number”, “:”, “PicoBricks’in en son tahmini, ve “Congratulations PicoBricks” ifadesini ekrana yazdıracacağız.

```

if receive IR code = 28
  clear
  write join Your number : guess at x 5 y 0 inverse
  write Congratulations PicoBricks at x 15 y 32 inverse

```

Projemiz Hazır

```

when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  write Press the Button at x 0 y 20 inverse
  write TO START!!! at x 15 y 30 inverse
  comment Direct PicoBricks to the arrow keys on the remote.
  Prediction of PicoBricks;
  If it is greater than the number you hold, press the "down" button.
  If it is less than the number you are holding, press the "up arrow" key.

```

```

when PicoBricks button
  clear
  set count to 64
  set guess to count
  write join My guess : guess at x 25 y 0 inverse
  write Press "ok" if I guess correctly at x 0 y 30 inverse
  forever
    if receive IR code = 24
      set count to count / 2
      change guess by count
      write join My guess : guess at x 25 y 0 inverse
    if receive IR code = 82
      set count to count / 2
      change guess by 0 - count
      write join My guess : guess at x 25 y 0 inverse
    if receive IR code = 28
      clear
      write join Your number : guess at x 5 y 0 inverse
      write Congratulations PicoBricks at x 15 y 32 inverse

```

PART 8



Bu Bölümde Öğreneceklerimiz

Bu bölümde öğrenciler;

- Listelerin kullanım amacını,
- PicoBricks programlama araçlarından microblocks ile değişkene bir liste atamayı ve liste bloklarını kullanmayı öğrenir.

Listeler

Listeler, verileri daha düzenli bir şekilde kullanmamızı sağlayan yapılardır. Listeleri kitaplığa benzetebiliriz. Kitaplıkta bulunan sıra sayısı ve kitap sayısı o kitaplığın büyüklüğünü belirler. Aşağıda verilen örnek kitaplığı inceleyiniz.



Kitaplıkta 6 kitap bulunmaktadır. Bu kitaplığı bir liste olarak düşünürsek bu liste 6 elemanlı bir listedir.

Liste Tanımlayalım

Hadi şimdi MicroBlocks ile bir liste tanımlayalım. Bunun için Veri (Data) bloklarından **list cat** bloğunu kullanacağız. Bloğun en sağında bulunan ok tuşuna basarak liste boyutunu ayarlayabiliriz. Oluşturduğunuz listeye bir isim vermek için öncelikle bir değişken tanımlamanız gerekmektedir. “list” adında bir değişken oluşturalım.

5 elemanlı bir liste oluşturalım, bu listenin elemanlarını [a,b,c,d,e] olarak belirleyelim ve bu listeyi daha önce oluşturduğumuz “list” değişkenine tanımlayalım.

set list ▼ to list a b c d e

Listeden Eleman Çağırılım

item 1 of Rosa bloğunu kullanarak oluşturduğumuz listedeki bir elemanı çağırabiliriz. Bu bloğu proje sayfamıza ilk sürüklediğimizde "Rosa" metnini otomatik olarak harflerine ayırır ve bu diziyi bir liste olarak algılar ve [R, o, s, a] listesindeki 1. elemanı çağırır. Bu bloğun çıktısı "R" harfidir.

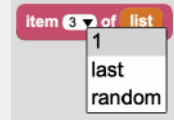
Şimdi "list" adını verdiğimiz listedeki 3. elemanı çağıralım ve OLED ekrana yazdıralım.

item 1 of Rosa bloğunda "Rosa" yazan alana değişkenlerden list değişkenini sürükleyelim ve eleman sayısını 3 yapalım. Eleman sayısını değiştirdiğimiz yerden ok tuşuna tıklayarak son elemanı ve rastgele bir elemanı çağırabiliriz.

Biz 3. elemanı

çağıracağımız için bu alana klavye ile "3" değerini yazalım.

value "3" in this field with the keyboard.



item 3 of list

Bu değeri OLED ekranda görebilmek için aşağıdaki kod bloklarını oluşturalım.

```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set list to list a b c d e
  write item 3 of list at x 0 y 0 inverse
```



Listenin Uzunluğunu Bulalım

Oluşturduğumuz bir listenin uzunluğunu bulmak için **length of Rosa** bloğunu kullanırız. Bu bloğu proje sayfamıza sürükleyip üstüne bastığımızda bloğun üzerinde 4 değerini yazdığını görürüz çünkü [R,o,s,a] bloğu 4 elemanlıdır.

Hadi şimdi OLED ekrana “list” adında oluşturduğumuz listenin eleman sayısını ekrana yazdıralım.”Rosa” yazısının bulunduğu alana **length of Rosa** bloğunu sürükleyerek **length of list** bloğunu oluşturalım ve OLED ekranda yazdıralım. Bizim oluşturduğumuz liste 5 elemandan oluştuğu için ekranda 5 yazacaktır.

```

when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set list to list a b c d e
  write length of list at x 0 y 0 inverse
  
```



Listeye Eleman Ekleyelim

Oluşturduğumuz bir listeye eleman eklemek için **add fish to list** bloğunu kullanırız. Şimdi “list” adında oluşturduğumuz listeye “f” elemanını ekleyelim ve OLED ekranda yazdıralım. Bunun için **add f to list list** bloğunu oluşturalım ve OLED ekranda yazdıralım.

```

when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set list to list a b c d e
  add f to list list
  write item last of list at x 0 y 0 inverse
  
```



Listedeki Bir Elemanı Değiştirelim

Oluşturduğumuz listedeki bir elemanı **replace item 1 of list** bloğunu kullanarak değiştirebiliriz. Şimdi “list” adında oluşturduğumuz listenin 1. elemanını “picobricks” olarak değiştirelim ve OLED ekranda yazdıralım. Bunun için ilk olarak **replace item 1 of list** bloğunu oluşturalım daha sonra aşağıdaki kod bloklarını oluşturarak OLED ekranda 1. elemanı yazdıralım.

```
when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
set list to list a b c d e
replace item 1 of list list with picobricks
write item 1 of list at x 0 y 0 inverse
```



Listeden Eleman Silme

Oluşturduğumuz bir listeden istediğimiz elemanı silmek için **delete item 1 of list** bloğunu kullanırız. Şimdi oluşturduğumuz listeden 1. eleman olan “a” ‘yı silelim ve tekrar 1. elemanı ekrana yazdıralım. **delete item 1 of list** bloğunu kullanıp 1. elemanı sildikten sonra artık 1. elemanımız “b” olacaktır. Bundan dolayı OLED ekran üzerinde “b” harfi yazar.

```
when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
set list to list a b c d e
delete item 1 of list list
write item 1 of list at x 0 y 0 inverse
```

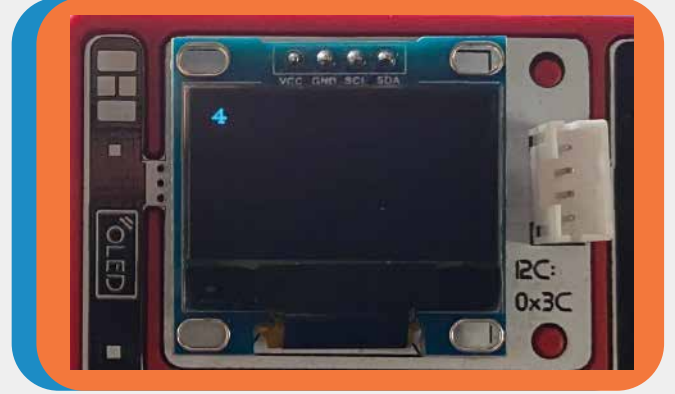


Listenin İçerisindeki Bir Elemanı Bulalım

Oluşturduğumuz listede ki bir elemanın, listenin kaçınıcı elemanı olduğunu bulmak için **find a in cat** bloğunu kullanırız. “list” adında tanımladığımız listeninin içerisinde “d” elemanını **find d in list** bloğuyla arayalım. Listede kaçınıcı eleman olduğunu OLED ekran yazdıralım.

```

when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  set list to list a b c d e
  write find d in list at x 0 y 0 inverse
  
```



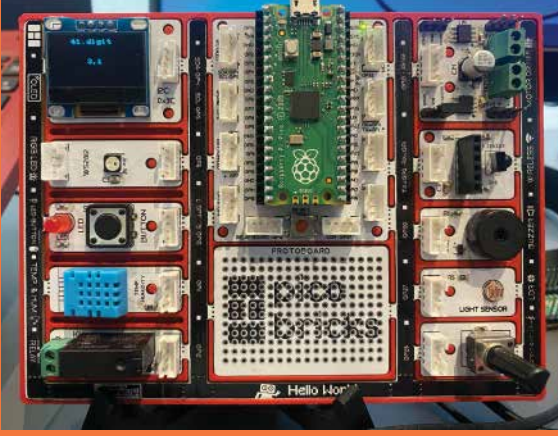
Belirlediğimiz Değer Aralığı Kadar Eleman Çağıralım

Oluşturduğumuz bir listede istediğimiz değer aralığını çağırmak için **copy smiles from 2** bloğunu kullanırız. Örneğin bu bloğu ilk olarak proje sayfamıza sürüklediğinde “smiles” kelimesini bir liste olarak algılar ve 2. elemanından başlayarak sonuncu elemanına kadar çağırır yani üzerine tıkladığımızda “miles” değerini görürüz. “smiles” listesinde 2. elemanda başlayarak 5. elemana kadar olan değerleri çağırmak için bloğun sonundaki ok işaretine tıklayarak değer aralığını girebiliriz.

copy smiles from 2 to 5 bu blokta 2. elemandan başlayarak 5. elemana kadar olan elemanları çağırır ve çıktısı “mile” ’dir. Şimdi daha önceden oluşturduğumuz “list” adındaki listeden yalnızca “b,c,d” elemanlarını çağıralım bunun için **copy list from 2 to 4** bloğunu kullanacağız.



Bölüm Projesi: Pi Sayısını İlk 100 Basamağı



Bu projede Pi sayısının ilk 100 basamağını sırayla ekrana yazdıracağız. Sayı basamakları ekrana yazarken aynı zamanda ekrandaki sayının kaçınıcı basamak olduğu ekranda belirtilecektir. Bu proje pi sayısının basamaklarını ezberlemek için tasarlanmış küçük bir egzersiz projesi olarak kullanılabilir.

Proje Algoritması

1. Başla
2. Pi sayısının ilk basamağı haric yani virgülden sonraki diğer 99 basamağı pi adında bir değişken oluştur ve pi değişkenine tanımla.
3. Kaçınıcı basamakta olduğunu belirlemek için bir "counter" değişkeni tanımla ve ilk değerini "1" olarak tanımla.
4. Pi sayısının basamak sayısı kadar yani "Pi" değişkeni kadar dönecek bir for döngüsü tanımla.
5. For döngüsü her tekrar edişinde "Counter" değişkenini 1 arttır ve "counter" değişkenini "th" ve "digit" ifadeleriyle beraber OLED ekrana yazdır.
6. Pi sayısını basamaklarını yani pi değişkenini, "3" ve "," ifadeleriyle beraber ekrana yazdıralım.
7. Basamakları yarım saniye aralıklarla yazdırmak için yarım saniye bekleyelim.
8. İlk adıma geri dön.

Projenin MicroBlocks Kodlarını Oluşturalım

"when started" bloğunu proje sayfamıza sürükleyelim ve basamakları OLED ekranda yazdıracağımız için OLED ekranı başlatalım.

when started

initialize I2c OLED_0.96in address(hex) 3C reset pin# 0 flip

“pi” adında bir değişken oluşturalım. Bu değişkenin içerisinde pi sayısının ilk basamak hariç diğer 99 basamağı bulunan listeyi tanımlayacağız. Pi sayısının basamaklarını teker teker listeye girmek yerine **split A,B,C by ,** bloğunu kullanarak her basamağın arasına virgül koyarak sağlayacağız. **split A,B,C by ,** bloğu içerisinde girdiğimiz değerlerin arasına virgül koyduğumuzda iki virgül arasındaki değeri listenin elemanı olarak algılar. Bloğun içerisine virgülden sonra gelen 99 elemanın arasına virgül yazarak

```
split
1,4,1,5,9,2,6,5,3,5,8,9,7,9,3,2,3,8,4,6,2,6,4,3,3,8,3,2,7,9,5,0,2,8,8,4,1,9,7,1,6,9,3,9,9,3,7,5,1,0,5,8,2,0,9,7,4,9,4,4,5,9,2,3,0,7,8,1,6,4,0,6,2,8,6,2,0,8,9,9,8,6,2,8,0,3,4,8,2,5,3,4,2,1,1,7,0,6,7
by ,
```

bloğunu oluşturalım ve pi değişkenine tanımlayalım.

Sonrasında counter adında bir değişken daha oluşturalım ve başlangıç değerini “1” yapalım.

```
when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
set pi to
split
1,4,1,5,9,2,6,5,3,5,8,9,7,9,3,2,3,8,4,6,2,6,4,3,3,8,3,2,7,9,5,0,2,8,8,4,1,9,7,1,6,9,3,9,9,3,7,5,1,0,5,8,2,0,9,7,4,9,4,4,5,9,2,3,0,7,8,1,6,4,0,6,2,8,6,2,0,8,9,9,8,6,2,8,0,3,4,8,2,5,3,4,2,1,1,7,0,6,7
by ,
set counter to 1
```

Şimdi projemizin içerisinde kullandığımız döngü aşamalarına geçebiliriz. For döngü bloğunu projemize sürükledikten sonra döngünün tekrar sayısını belirlemek için **length of pi** bloğunu for döngüsü bloğunun içerisine sürükleyiniz ve for bloğunu oluşturunuz.

```
when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
set pi to
split
1,4,1,5,9,2,6,5,3,5,8,9,7,9,3,2,3,8,4,6,2,6,4,3,3,8,3,2,7,9,5,0,2,8,8,4,1,9,7,1,6,9,3,9,9,3,7,5,1,0,5,8,2,0,9,7,4,9,4,4,5,9,2,3,0,7,8,1,6,4,0,6,2,8,6,2,0,8,9,9,8,6,2,8,0,3,4,8,2,5,3,4,2,1,1,7,0,6,7
by ,
set counter to 1
for i in 10
```

Döngünün içerisinde tekrar edecek ifadeleri oluşturalım. Döngü her tekrar ettiğinde basamak değerini arttırmak için tanımladığımız “counter” değişkenini “1” arttıralım bunun için **change counter by 1** bloğunu oluşturun. Basamak değerini ekrana yazdırmak için join bloğunu kullanarak “counter” değişkeni , “th” ve “digit” ifadelerini oled ekranda yazdırın. **write join counter digit at x 25 y 0 inverse** basamak değerinden sonra pi sayısının o an ki basamağını ekrana yazdırmak için **write join 3 item i of pi at x 50 y 30 inverse** bloğunu oluşturun. Bu blok sayesinde döngü her tekrar ettiğinde sırayla pi listesindeki elemanlar ekran gelecektir. Her basampı yarım saniye aralıklarla ekrana yazdırmak için yarım saniye bekleme bloğunu sürükleyelim ve projemizi çalıştıralım.

```
write join 3 item i of pi at x 50 y 30 inverse
```

Proje Kodlarını Çalıştırınız

```
when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
set pi to
split
1,4,1,5,9,2,6,5,3,5,8,9,7,9,3,2,3,8,4,6,2,6,4,3,3,8,3,2,7,9,5,0,2,8,8,4,1,9,7,1,6,9,3,9,9,3,7,5,1,0,5,8,2,0,9,7,4,9,4,4,5,9,2,3,0,7,8,1,6,4,0,6,2,8,6,2,0,8,9,9,8,6,2,8,0,3,4,8,2,5,3,4,2,1,1,7,0,6,7
by ,
set counter to 1
for i in length of pi
change counter by 1
write join counter digit at x 25 y 0 inverse
write join 3 item i of pi at x 50 y 30 inverse
wait 500 millisecs
```

K-12 Uluslararası Bilgisayar Bilimi Standartları

- 3B-AP-20 11 -12 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
- 3A-AP-14 9-10 Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
- 3A-AP-13 9-10 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
- 2-AP-11 6- 8 Create clearly named variables that represent different data types and perform operations on their values.
- 1B-AP-10 3-5 Create programs that include sequences, events, loops, and conditionals.
- 1B-AP-09 3-5 Create programs that use variables to store and modify data.

command ▶

reporter ▶

command ▶

reporter ▶

reporter ▶

PART 9

define command ▶

define reporter ▶



Bu Bölümde Öğreneceklerimiz

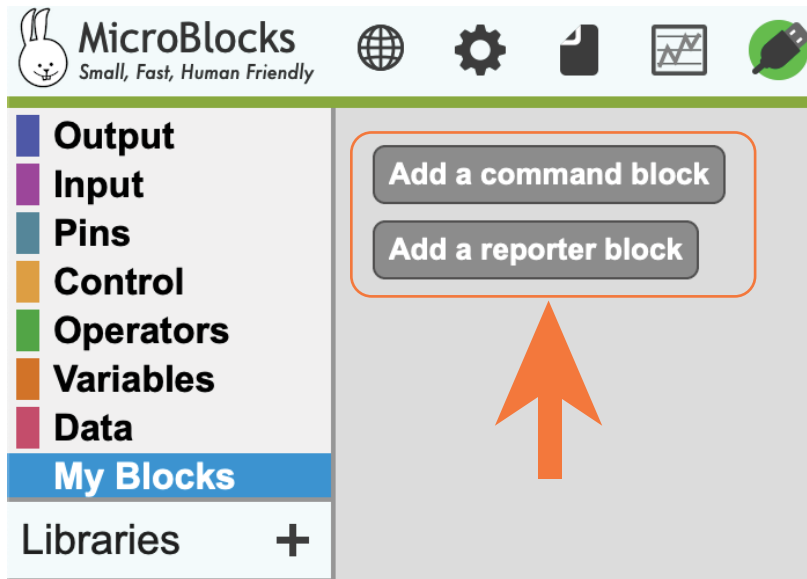
Bu bölümde öğrenciler;

- “My blocks” bloklarının hangi amaçla kullanıldığını kavrar,
- Define bloğunu kullanarak kendi bloklarını oluşturabilir,
- Command ve Reporter blokları arasındaki farkı açıklar.

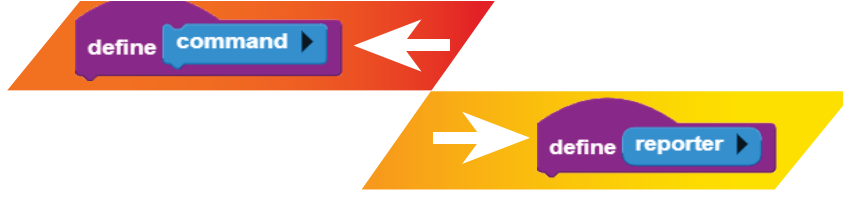
Define - Kendi Bloğunu Oluşturmak

“define”, sayesinde programlama ortamlarında kendimize ait komutlar oluşturabiliriz. Örneğin bir projede sürekli olarak asal sayıları tespit etmemiz ve o asal sayılara işlem yapmamız isteniyor bunun için sürekli olarak asal sayıları tespit etmek yerine asal sayıları bulduğumuz bir fonksiyon tanımlayarak sadece o fonksiyonu çağırabiliriz. Bu uygulama yazdığımız kodun daha sade ve anlaşılır olmasını da sağlamaktadır.

MicroBlocks IDE de projelerimizi geliştirirken kendimize ait kod blokları oluşturabilir ve bu bloklara görevler atayabiliriz. Bu işlemi define bloklarını kullanarak yaparız. MicroBlocks define bloklarına ulaşmak için “My Blocks” menüsü kullanılır. “My Blocks” menüsünde kendi blogunuzu tanımlamak için command block ve reporter block adında iki seçenek bulunur. Bu seçenekleri kullanarak kendi bloklarımızı oluşturabilir ve bu bloklara görevler atayabiliriz.



Command ve Reporter Arasındaki Fark

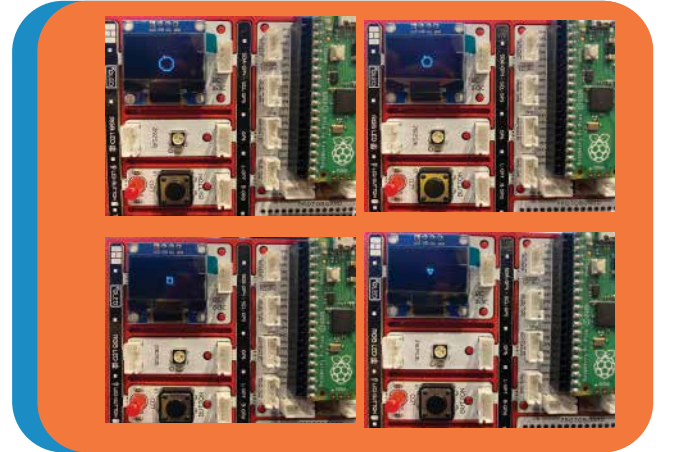


Command ve reporter blokları farklı amaçlarla kullandığımız blok türlerindedir bu farklılığı blokların yapısından da rahatlıkla anlayabiliriz. **command** bloğunun yapısına baktığımızda yapboz parçası gibi olduğunu görebiliyoruz yani oluşturduğumuz command bloklarını diğer klod bloklarının altına sürükleyebiliriz demektir. **reporter** bloğunun yapısına baktığımızda oval bir yapıda olduğunu bu sayede koşul ifadeleri vb. durumlarda kullanabileceğimiz kod bloklarından olduğunu anlayabiliyoruz.

Şimdi bir kaç tane etkinlik yaparak bu iki blok arasındaki farkı daha iyi kavrayabiliriz.

Define ile Şekiller (Command)

Bu projede turtle kütüphanesini kullanarak oluşturduğumuz kare, üçgen, altıgen ve sekizgen şekillerini "my blocks" sekmesindeki define(command) bloğunu kullanarak tek bir blok haline getireceğiz ve artık oluşturduğumuz o bloğu çağırığımızda istediğimiz şekili OLED ekranda yazdıracağız.



Projenin Algoritması

1. Başla
2. OLED ekranı başlat
3. Turtle'ı başlat
4. Sekizgeni tanımla
5. Altıgeni tanımla
6. Kareyi tanımla
7. Üçgeni tanımla
8. Sekizgeni çağır
9. Altıgeni çağır
10. Kareyi çağır
11. Üçgeni çağır
12. İlk adıma geri dön

Projenin Microblocks Kodları

Projemizde kullanacağımız OLED ve Turtle kütüphanlerini MicroBlocks IDE'ye yükleyelim. Daha sonra projemize OLED ekran ve turtle imlecimizi çağıralım. Daha sonra turtle imlecinin OLED ekran üzerinde izlediği yolu takip edebilmek için **pen down** bloğunu proje sayfamıza sürükleyelim.



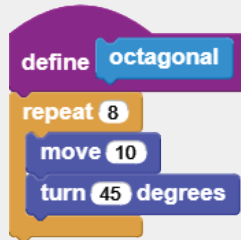
Şimdi "My Blocks" sekmesini kullanarak sekizgen,altıgen,kare ve üçgen şekillerini oluşturan kod bloklarını oluşturalım.



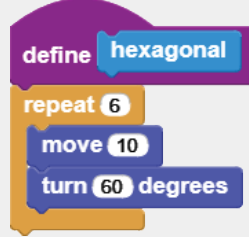
Yukarıdaki adımları takip ederek "octagonal" adında bir command bloğu oluşturalım. Bu blok sayesinde turtle imleci ile OLED ekran üzerinde sekizgen şeklini oluşturacağız.

Yukarıdaki adımları takip ettiğimizde proje sayfamızda **define octagonal** bloğunun oluştuğunu gözlemleyeceksiniz.

Turtle kütüphanesi ile sekizgen çizebilmek için turtle imlecini 8 kere 45 derecelik açılarla 10 adım ilerlemesini sağlayacağız. Bunun için aşağıdaki blok yapısını oluşturalım.

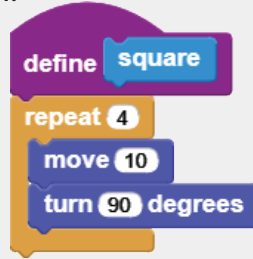


Altıgen bloğunu oluşturalım. Turtle imlecinin OLED ekran üzerinde altıgen şeklini oluşturması için 6 kere 60 derecelik açılarla dönerek 10 adım ileri gitmesi gerekir. Bu işlemleri aşağıdaki blok yapısını oluşturarak “hexagonal” adında oluşturduğumuz define bloğuna tanımlayalım.



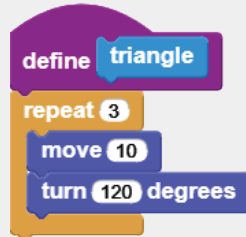
```
define hexagonal
repeat 6
move 10
turn 60 degrees
```

Kare bloğunu oluşturalım. Turtle imleci ile OLED ekran üzerinde kare şeklini oluşturabilmek için 4 kere 90 derecelik açılarla döndükten sonra 10 adım ileri giderek oluşturuz. “square” adında oluşturduğumuz command bloğuna bu adımları aşağıdaki kod bloklarındaki gibi tanımlayalım.



```
define square
repeat 4
move 10
turn 90 degrees
```

Üçgen bloğunu oluşturalım. Turtle imleci ile OLED ekran üzerinde üçgen şeklini oluşturabilmek için imlecini 3 kere 120 derecelik açılarla döndükten sonra 10 adım ileri gitmesi gerekmektedir. Bunun için “triangle” adında oluşturduğumuz command bloğuna bu adımları aşağıdaki kod bloklarında olduğu gibi sıralayalım.



```
define triangle
repeat 3
move 10
turn 120 degrees
```

Bu oluşturduğumuz kod bloklarını önce teker teker projemize çağıralım ve çalıştıralım. Ekranda oluşan şekilleri gözlemledikten sonra bütün şekilleri arka arkaya çağırarak projemizi tamamlayalım.

```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  initialize turtle large
  pen down
  octagonal
  hexagonal
  square
  triangle
```

Projemizin MicroBlocks kodları hazır

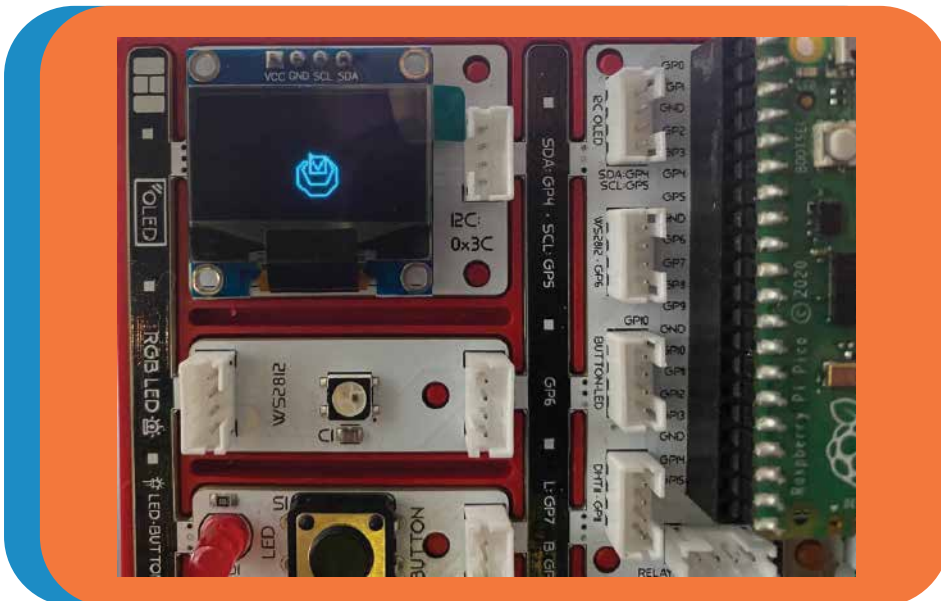
```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  initialize turtle large
  pen down
  octagonal
  hexagonal
  square
  triangle
```

```
define octagonal
  repeat 8
    move 10
    turn 45 degrees
```

```
define square
  repeat 4
    move 10
    turn 90 degrees
```

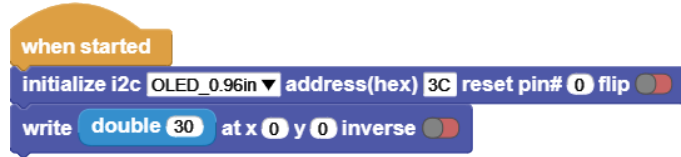
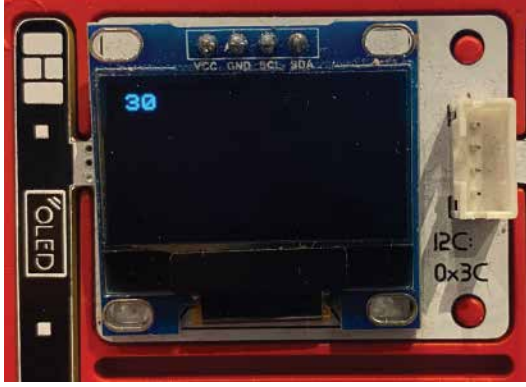
```
define hexagonal
  repeat 6
    move 10
    turn 60 degrees
```

```
define triangle
  repeat 3
    move 10
    turn 120 degrees
```



Define ile Sayının İki Katı (Reporter)

Bu projede, define bloklarından “reporter” yapısını kullanarak girilen sayının iki katını OLED ekran üzerinde yazdıracağız.



Proje Algoritması

1. Başla
2. “double” adında bir fonksiyon tanımla
3. “double” fonksiyonu içerisine “number” adında bir değişken tanımla.
4. “number” değişkenini sürekli 2 ile çarparak döndür.
5. OLED ekranda double fonksiyonunu belirlenen değer için çağır.
6. İlk maddeye dön

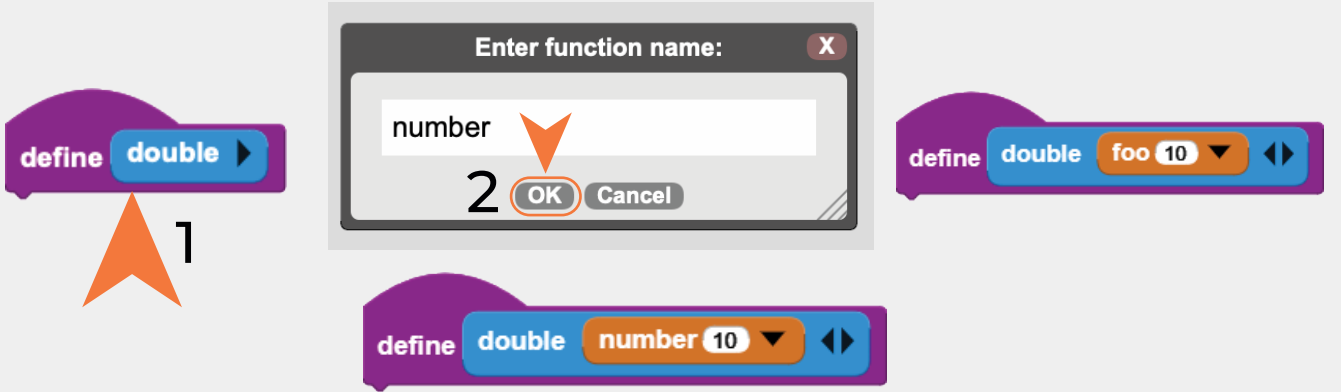
Projenin Microblocks Kodları

İlk olarak define fonksiyonumuzu hazırlayalım. “My Blocks” sekmesine girerek “double” adında bir reporter bloğu oluşturmak için aşağıdaki adımları takip edelim.

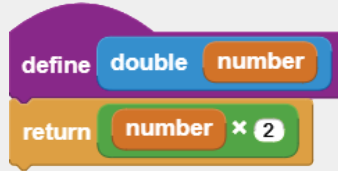


Yukarıdaki adımları takip ederek oluşturduğumuz double bloğunun içerisinde bir number değişkeni oluşturmak için aşağıdaki adımları takip edelim.

MicroBlocks IDE proje sayfamızda oluşan “double” fonksiyonunda “number” değişkeni oluşturmak için aşağıdaki gibi “double” bloğunun üzerindeki ok ikonuna tıklayalım.



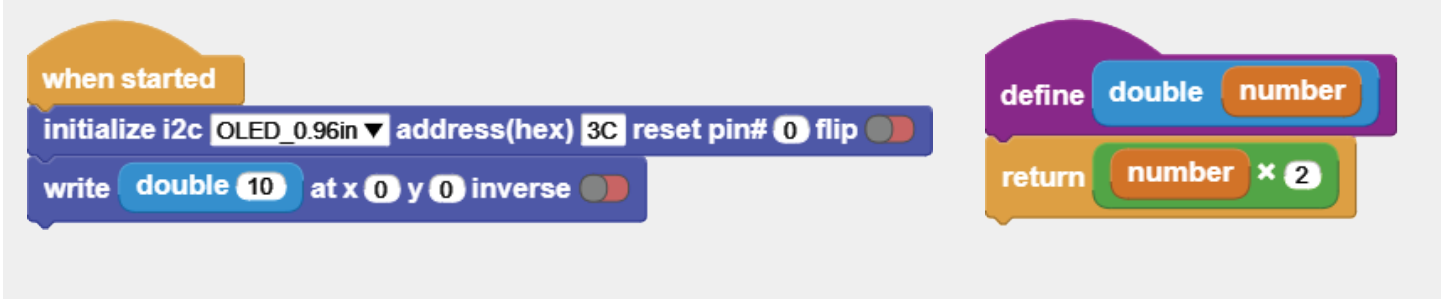
Yukarıda anlatıldığı gibi “number” değişkenini tanımlayalım. “Control” blokları içinden **return 0** bloğunu kullanarak, “double” bloğunu her çağırdığımızda “number” değişkeninin iki katını döndürelim. Bu işlemi yapmak için **return number * 2** bloğunu oluşturalım ve “double” fonksiyonunun altına sürükleyelim.



Şimdi “double” fonksiyonun döndürdüğü değeri OLED ekran modülü üzerinde yazdıralım.”My Blocks” sekmesi içersinden **double 10** bloğunun döndürdüğü değeri OLED ekran üzerinde yazdırmak için aşağıdaki kod bloklarını oluşturalım.



Projemizin MicroBlocks kodları hazır



K-12 Uluslararası Bilgisayar Bilimi Standartları

- 3A-AP-13 - 9-10 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
- 2-AP-19 -6-8 Document programs to make them easier to follow, test, and debug.
- 3A-AP-15-9-10 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
- 3A-AP-21 - 9-10 Evaluate and refine computational artifacts to make them more usable and accessible.
- 2-AP-16- 6-8 Incorporate existing code, media, and libraries into original programs, and give attribution.
- Use flowcharts and/or pseudocode to address complex problems as algorithms.



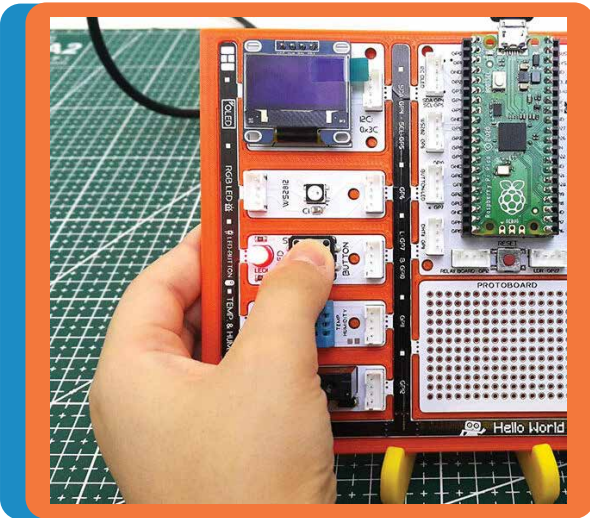
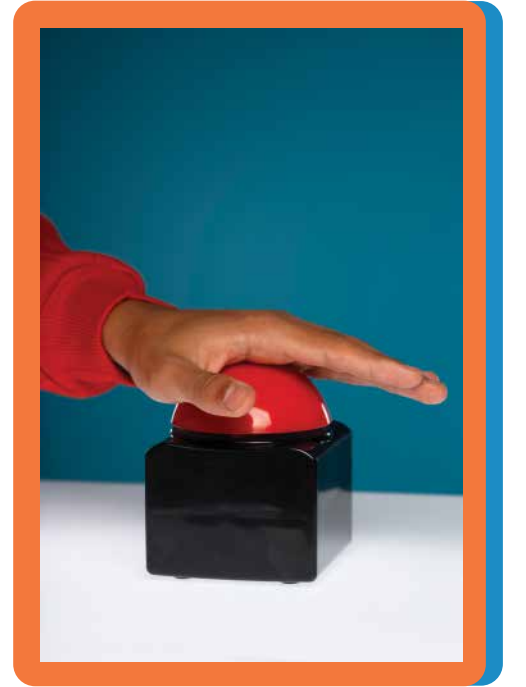
STEM 1

STEM Proje 1 - Action Reaction

Newton hareket yasalarında açıkladığı gibi her etkiye karşı bir tepki oluşur. Elektronik sistemler kullanıcılardan komutlar alırlar ve görevlerini yerine getirirler. Genellikle bu iş için bir tuş takımı, dokunmatik ekran veya bir buton kullanılır. Elektronik cihazlar görevlerinin sona erdiğini ve görev süresince olan bitenden kullanıcıyı haberdar etmek için sesli, yazılı veya görsel olarak tepki verirler. Bu tepkilerin kullanıcıyı haberdar etmenin yanı sıra olası arızada hatanın nerede olabileceğinin anlaşılmasında yardımcı olabilmektedir.

Proje Detayları

Elektronik sistemlerde farklı türlerde butonlar kullanılmaktadır. Kilitli butonlar, push butonlar, anahtarlama butonlar... Picobricks üzerinde 1 adet push buton bulunmaktadır. Anahtar gibi çalışmaktadırlar, basıldığında akımı iletirler bırakıldığında akımı iletmezler. Projede butonun akım iletip iletmediğini kontrol ederek basılma durumunu anlayacağız. Basılmışsa LED'i yakacak, basılmamışsa LED'i söndüreceğiz.






Proje Algoritması

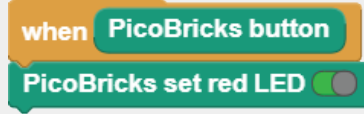
1. Başla
2. Butona basıldıysa LED'modülünü yak.
3. Butona basılmadığı durumlarda LED'i söndür.
4. İlk adıma geri dön





Projenin MicroBlocks Kodları


Proje sayfamıza PicoBricks kütüphanemizi yükleyelim. Sonrasında “control” bloklarından  bloğunu proje sayfamıza sürükleyelim ve içerisine PicoBricks

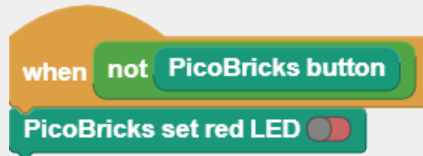
kütüphanesinden “PicoBricks button” bloğunu sürükleyelim.  Oluşturduğumuz bu blok sayesinde altına sürüklediğimiz bütün kod blokları PicoBricks üzerindeki butona bastığımızda gerçekleşecektir. Butona bastığımızda LED’in çalışmasını istediğimiz için  bloğunu proje sayfamıza sürükleyelim ve aşağıdaki kod bloğunu oluşturalım.



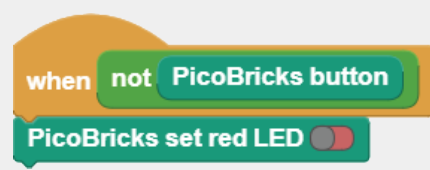
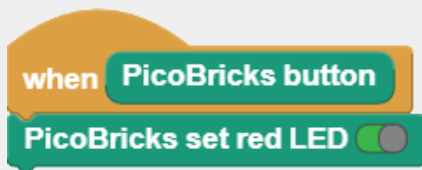
Şimdi PicoBricks üzerindeki butona basılı olmadığı durumlarda gerçekleşecek ifadeleri oluşturmak için gerekli olan kod bloklarını oluşturalım. İlk olarak  bloğunu

proje sayfamıza sürükleyelim. Daha sonra operatör bloklarından  bloğunu bu bloğun içerisine sürükleyelim. “not” bloğunun içerisine sürüklediğimiz ifade o bloğun olumsuz olduğu durumlara karşılık gelir. “not” bloğunun içerisine “PicoBricks button” bloğunu sürüklediğimizde butona basılmadığı durumları algılamamızı sağlayan kod bloğu hazır. demektir bu bloklarını when bloğunun içerisine sürükleyelim.

 bu bloğun altına sürüklediğimiz her blok butona basılmadığı durumlarda çalışır. Butona basmadığımız durumlarda LED’in sönmesini istediğimiz için aşağıdaki kod bloklarını proje sayfamızda oluşturalım.



Proje Kodlarımız Hazır



STEM 2



STEM Project 2 - Dominate The Rhythm



Hayatımızdaki birçok olgu dijitalleştirilmiştir. Bunlardan biri de seslerdir. Sesin tonu ve şiddeti elektriksel olarak işlenebilmektedir. Yani notaları elektronik olarak çıkarabiliriz. Müziği oluşturan seslerin en küçük birimine nota denir. Her notanın bir frekansı ve şiddeti vardır. Yazacağımız kodlarla frekans ve şiddet uygulayarak hangi notanın çalınmasını ve ne kadar sürmesi gerektiğini ayarlayabiliriz.

Bu projede Picobricks ile bir şarkının melodisini buzzer modülünü kullanarak çalacak, potansiyometre modülü ile ritmi ayarlayabilecek bir müzik sistemi hazırlayacağız. Programlama terminolojisinde önemli bir yere sahip değişken kullanımını da bu projede öğreneceksin.

Proje Detayları

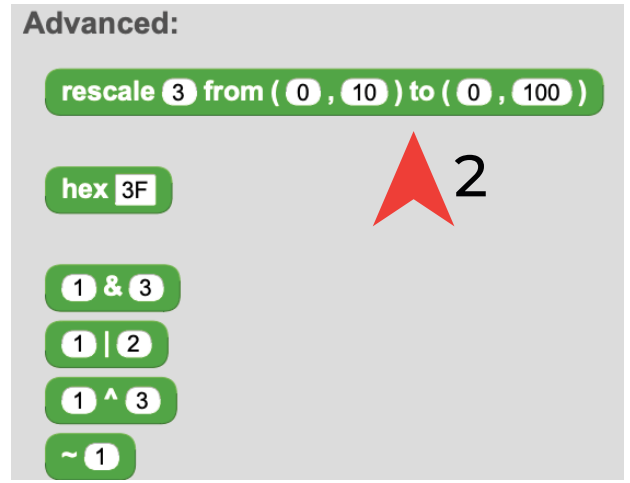
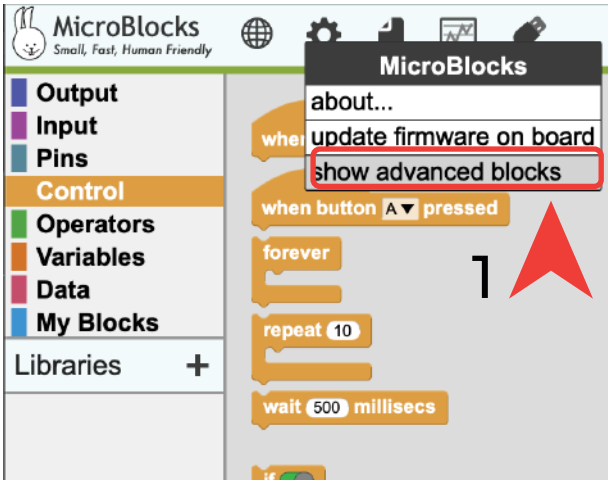
Potansiyometre analog giriş modülüdür. Değişken dirençtir. Üzerinden geçen akım miktarı çevrildikçe bir musluğun açılıp kapatılması gibi artar ve azalır. Bu akım miktarını kodlarla kontrol ederek şarkının hızını ayarlayacağız. Buzzer'lar üzerlerinden geçen akımın şiddetine göre ses seviyelerini, gerilim frekansına göre ses tonlarını değiştirmektedirler. Microblocks ile kolayca buzzer modülünden istediğimiz notaları, tonlarını ve sürelerini ayarlayarak kodlayabiliriz.

Proje Algoritması

1. Başla
2. Potansiyometrenin değerini bir değişkene ata ve ekrana yaz.
3. Butona basılana kadar bekle.
4. Butona basıldıysa belirlenen tonları potansiyometrenin değerine göre belirlenen hız ile çal.
5. 4.maddeyi iki kere tekrar et.
6. Tekrar butona basılana kadar. 2.maddeyi uygula

Projenin MicroBlocks Kodları

- 1) Proje başladığında OLED ekranda potansiyometreden gelen hız değerini yazdıracağımız için OLED ekranı başlatalım ve ekrana "Speed: " yazdıralım.
- 2) Potansiyometreden gelen değeri "rhythm" adında değişkene atayalım. Potansiyometrenin değeri 0-1023 arasındadır. Bu değer aralığını değiştirmek için 1-7 olarak değiştirmek için bloğunu kullanacağız. Bu blok operatör blokları içerisinde bulunur fakat gelişmiş bir bloktur. gelişmiş blokları göster seçeneğine tıkladıktan sonra operatör blokları arasında gözükecektir.



- 3) Potansiyometrenin değerini 1-7 arasına düşürüp "rhythm" adında bir değişkene tanımlamak için aşağıdaki kod bloklarını oluşturalım.

```
set rhythm to
rescale PicoBricks potentiometer from ( 0 , 1023 ) to ( 1 , 7 )
```

- 4) Bu değişkeni OLED ekranda yazdıralım. Daha sonra "beat" adında bir değişken daha oluşturalım ve bu değeri 1000 / "rhythm" değişkenini olarak tanımlayalım. Bu işlem tonların kaç saniye çalacağını belirler. Bu işlemden sonra 50 milisaniye bekleyelim.

```
set rhythm to
rescale PicoBricks potentiometer from ( 0 , 1023 ) to ( 1 , 7 )
write rhythm at x 56 y 36 inverse
set beat to 1000 / rhythm
wait 50 millisecs
```

5) Bu adımların sürekli devam etmesini istediğim için “forever” bloğunun içerisine sürükleyelim.

```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  write Speed: at x 8 y 36 inverse
  forever
    set rhythm to
      rescale PicoBricks potentiometer from ( 0 , 1023 ) to ( 1 , 7 )
    write rhythm at x 56 y 36 inverse
    set beat to 1000 / rhythm
    wait 50 millisecs
```

Butona bastıktan sonra tonların, belirlediğimiz hızda çalması için aşağıdaki adımları gerçekleştirelim.

```
when PicoBricks button
  write Now Playing... at x 8 y 16 inverse
```

5) Şimdi “Tone” kütüphanemizi yükleyelim ve melodimiz için gerekli olan tonları ayaralayalım. Daha sonra “beat” değişkeni ile bu tonların kaç milisaniye çalacağını belirleyelim.

```
play note A octave 0 for 2 * beat ms
play note E octave 0 for beat ms
play note E octave 0 for beat ms
play note E octave 0 for beat ms
play note E octave 0 for beat ms
play note E octave 0 for beat ms
play note E octave 0 for beat ms
play note E octave 0 for beat ms
play note F octave 0 for beat ms
play note E octave 0 for beat ms
play note D octave 0 for beat ms
play note F octave 0 for beat ms
play note E octave 0 for 2 * beat ms
```

- 4) Bu tonları iki kere tekrar ettirerek bir melodi oluşturmak istediğimiz için “repeat” bloğunun içerisine alalım ve tekrar değerini 2 olarak belirleyelim.

```

when PicoBricks button
  write Now Playing... at x 8 y 16 inverse
  repeat 2
    play note A octave 0 for 2 * beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note F octave 0 for beat ms
    play note E octave 0 for beat ms
    play note D octave 0 for beat ms
    play note F octave 0 for beat ms
    play note E octave 0 for 2 * beat ms
  
```

Proje Kodlarımız Hazır

```

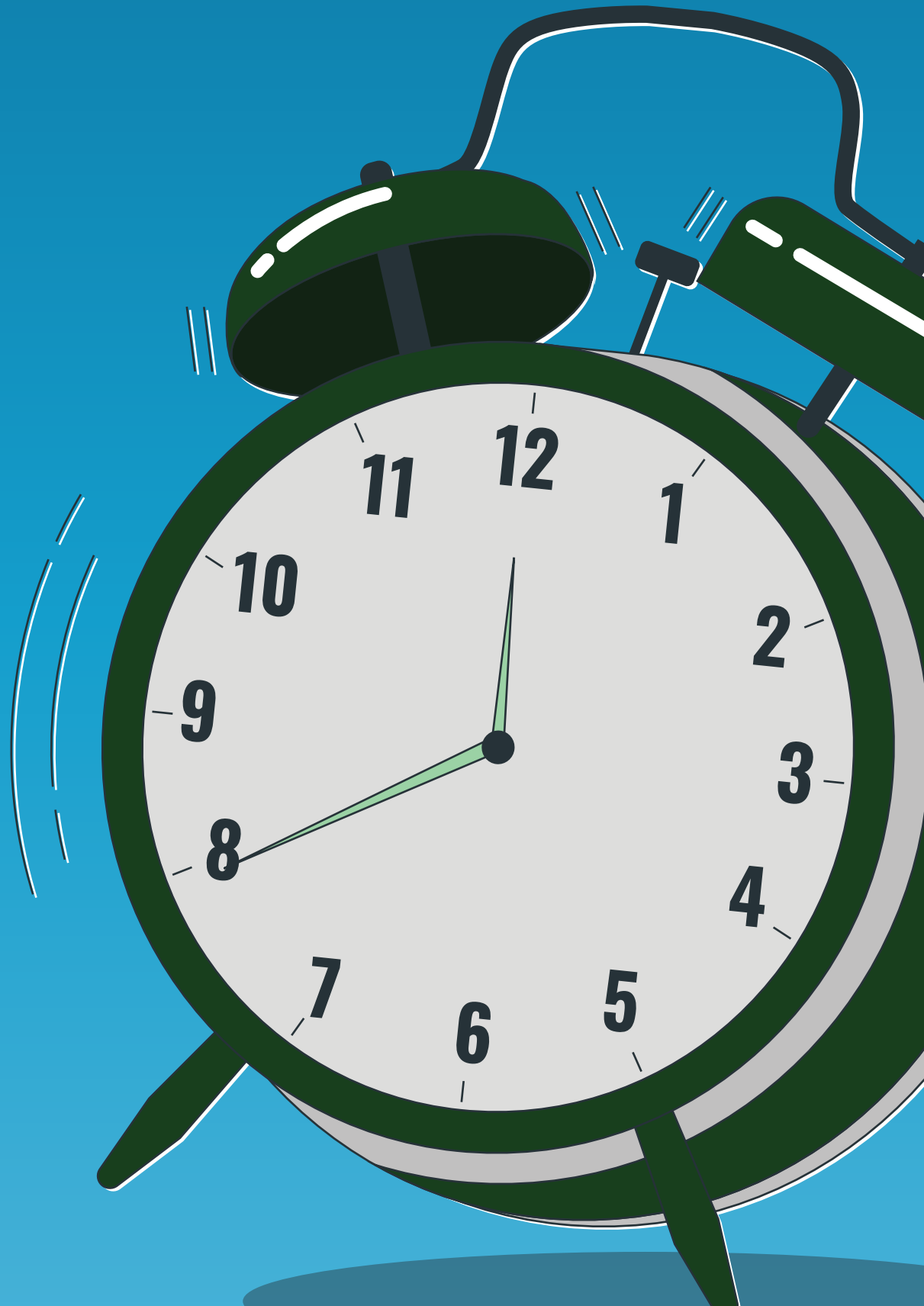
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  write Speed: at x 8 y 36 inverse
  forever
    set rhythm to
      rescale PicoBricks potentiometer from ( 0 , 1023 ) to ( 1 , 7 )
    write rhythm at x 56 y 36 inverse
    set beat to 1000 / rhythm
    wait 50 millisecs
  
```



```

when PicoBricks button
  write Now Playing... at x 8 y 16 inverse
  repeat 2
    play note A octave 0 for 2 * beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note E octave 0 for beat ms
    play note F octave 0 for beat ms
    play note E octave 0 for beat ms
    play note D octave 0 for beat ms
    play note F octave 0 for beat ms
    play note E octave 0 for 2 * beat ms
  
```

STEM 3



STEM Proje 3 - Alarm Clock



Küresel ısınma dünyamızın iklimini her geçen gün daha kötü etkiliyor. Ülkeler küresel ısınmanın etkilerini azaltmak için birçok tedbiri devreye sokuyor ve anlaşmalar imzalıyorlar. Yenilenebilir enerji kaynaklarının kullanılması ve enerjinin verimli kullanılması fabrikalardan evimizin odalarına kadar her yerde dikkat edilmesi gereken bir konudur. Şehirlerde yol ve park aydınlatmalarının insan hatasından dolayı açık kalması, yüksek enerji tüketen aydınlatma araçlarının kullanılması gibi birçok sebep enerji verimliliğini düşürmektedir. Ortamın ışık, sıcaklık ve nem değerlerini ölçerek sadece gerek duyulduğunda ve doğru miktarlarda kullanılmasını sağlayan birçok elektronik ve dijital sistem mühendisler tarafından geliştirilmekte ve programlanmaktadır.

Bu projede Picobricks'teki ışık sensörünü kullanarak gün ışığına göre ayarlanan bir saat alarmı hazırlayacağız.

Proje Detayları

Bu projede basit bir alarm uygulaması yapacağız. Tasarlayacağımız alarm sistemi sabah olduğunda otomatik olarak çalacak şekilde kurgulanmıştır. Bunun için projede LDR sensör kullanacağız. Gece olduğunda OLED ekranda kullanıcıya iyi geceler mesajı görüntülenecek, sabah olduğunda ise buzzer sesi ile alarm çalacak, ekranda kullanıcıya günaydın mesajı gösterilecek ve ışıklı bildirim amacıyla RGB LED beyaz renkte yanacak.

Kullanıcının alarmı durdurması için ise PicoBricks'in butonuna basması gerekecek. Alarm durdurulana kadar devam eden bu işlemlerden sonra butona basıldığında buzzer ve RGB LED kapanacak ve OLED ekranda kullanıcıya iyi günler mesajı gösterilecek.




Proje Algoritması

1. Başla
2. OLED ekrana “Good night” yaz.
3. PicoBricks LDR sensörünün değeri 90’dan büyükse (Sensör gün ışığını algıladığında) 3 saniye bekle
4. Butona basılana kadar OLED ekranda “Good morning” yaz, RGB LED’i ve buzzer’ı aç.
5. Butona basıldıysa OLED ekrana “Have a nice day” yaz ve RGB LED’i kapat.
6. 1.maddeye dön

Projenin MicroBlocks Kodları

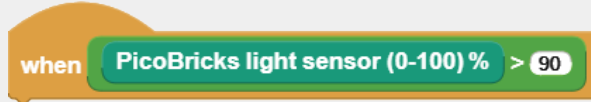
Proje kodlarımız başladığında, OLED ekranda “Good night” yazdıktan sonra iki saniye beklemek için aşağıdaki kod bloklarını oluşturalım.



```
when started
  initialize i2c OLED 0.96in address(hex) 3C reset pin# 0 flip
  write Good night at x 25 y 32 inverse
  wait 2000 millisecs
```

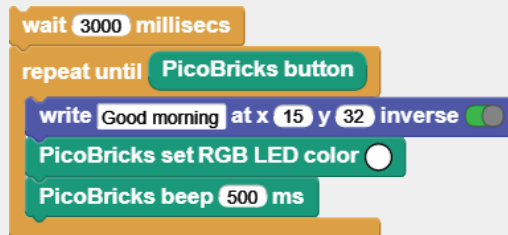
LDR sensörü gün ışığını algıladığında, çalışmasını istediğimiz kod bloklarını başlatmak için gerekli bloğu oluşturalım.

Gün aydınlandığında LDR sensörünün değerini 90’dan büyük kabul edelim (Bu değer ortamdan ortama farklılık göstermektedir.) Bu koşul gerçekleştiğinde istenilen kod bloklarını çalıştırmak için aşağıdaki kod bloklarını oluşturalım.



```
when PicoBricks light sensor (0-100) % > 90
```

Gün aydınlandığında yani LDR sensörünün değeri 90’dan büyük olduğunda, 3 saniye bekleyelim ve butona basılana kadar OLED ekranda “Good morning” yazdıralım, RGB LED’i açalım ve buzzer’ı çalıştıralım. Bu işlemleri gerçekleştirmek için aşağıdaki kod bloklarını sürükleyelim.



```
wait 3000 millisecs
repeat until PicoBricks button
  write Good morning at x 15 y 32 inverse
  PicoBricks set RGB LED color
  PicoBricks beep 500 ms
```

Butona basıldıktan sonra OLED ekranda “Have a nice day” yazdıralım, RGB LED’i kapatalım ve çalışan blokları durduralım. Bu işlemleri gerçekleştirmek için aşağıdaki kod bloklarını sürükleyelim.

```
write Have a nice day at x 0 y 32 inverse 
PicoBricks turn off RGB LED
stop this task
```

Projenin Kodları Hazır!

```
when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip 
write Good night at x 25 y 32 inverse 
wait 2000 millisecs
```

```
when PicoBricks light sensor (0-100) % > 90
wait 3000 millisecs
repeat until PicoBricks button
write Good morning at x 15 y 32 inverse 
PicoBricks set RGB LED color 
PicoBricks beep 500 ms
write Have a nice day at x 0 y 32 inverse 
PicoBricks turn off RGB LED
stop this task
```



The background is a solid green color with a subtle gradient. It is decorated with several horizontal strings of colorful lights. Each string consists of small, glowing circles in red, yellow, cyan, and white, arranged in a repeating pattern. The strings are curved, creating a festive and celebratory atmosphere.

STEM 4

STEM Project 4 - Know Your Color

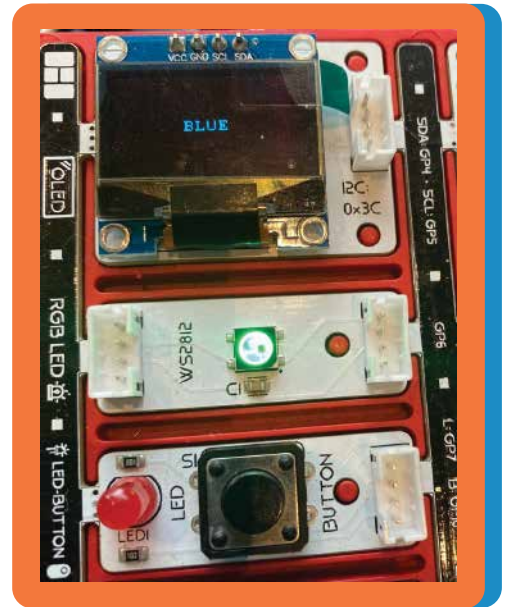


Elektronik sistemler üzerinde LED'ler sıklıkla kullanılır. Her butonun her seçeneğin yanında küçük LED'ler bulunabilmektedir. Tek bir LED'i değişik renklerde yanmasını sağlayarak birden fazla LED'in yaptığı işi tek bir LED ile yapılması sağlanabilmektedir. Bu türde çalışan LED'lere RGB LED denir. Adını Red, Green, Blue renk isimlerinin baş harflerinden alır. Bu LED'in diğer avantajı da 3 ana rengin karışımlarında da yanabilmesidir. Mor, turkuaz, turuncu...

Bu projede her programlama dilinde kullanılan rastgelelik durumunu öğreneceksin. Picobricks'in RGB LED, OLED ekran ve buton modülü ile eğlenceli bir oyun hazırlayacağız.

Proje Detayları

Projede inşa edeceğimiz oyun kullanıcının renkleri doğru veya yanlış bilmesi üzerine kurgulanacaktır. PicoBricks üzerindeki RGB LED'de kırmızı, yeşil, mavi ve beyaz renklerden birisi rastgele olarak yanacak, aynı anda OLED ekranda yine bu dört renkten birisinin adı rastgele olarak yazılacaktır. Kullanıcı cevap hakkını kullanmak için 1,5 saniye içerisinde PicoBricks'in butonuna basmalıdır. Oyun 10 kere tekrarlanacak, her tekrarda renkler eşleştiğinde kullanıcı butona basarsa ya da eşleşmediğinde kullanıcı butona basmazsa 10 puan alacaktır. Renkler eşleşmediği halde kullanıcı butona basarsa 10 puan kaybedecektir. On tekrar sonunda kullanıcının aldığı puan OLED ekranda gösterilecektir. Kullanıcı dilerse butona basmayarak cevap hakkını kullanmayabilir.



Projenin Algoritması

1. Başla
2. Yeşil-mavi-kırmızı-beyaz renkleri arasından rast gele bir rengi OLED ekran
3. üzerinde yazdır.
4. Yeşil - mavi- kırmızı - beyaz renkleri arasından rastgele bir rengi RGB LED de çalıştır.
5. OLED ekranda yazan renk ve RGB LED modülünde yanan renk aynıysa ve butona basıldıysa skoru 10 arttır.
6. OLED ekranda yazan renk ile RGB LED aynı fakat butona basılmadıysa skoru 10 azalt
7. OLED ekranda yazan renk ve RGB LED aynı değilse skoru aynı bırak
8. Bu işlemler 10 kere tekrar ettikten sonra skoru ekrana yazdır.

- 1) Projemizde OLED ekran modlünü kullanacağımız için OLED ekranı başlatarak başlayalım.
- 2) RGB LED modülünü proje her başladığında sıfırlamak için projenin başlangıcında RGB LED'i kapatalım.
- 3) Oyun boyunca skorumuzu kayıt etmesi için "score" adında bir değişken oluşturalım.
- 4) OLED ekrana "The game begins " yazısını yazdırdıktan sonra 2 saniye bekleyelim ve OLED ekranı temizleyelim.

Yukarıda belirtilen maddeleri sağlamak için aşağıdaki kod bloklarınıproje sayfamızda oluşturalım.

```
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  PicoBricks turn off RGB LED
  set score to 0
  write The game begins at x 8 y 32 inverse
  wait 2000 millisecs
  clear
```

- 5) Bu blokların hemen altında oyunumuz için gerekli olan asıl işlemler yapılacaktır. Oyun 10 kere tekrar edeceği için repeat döngüsünü sürükleyelim ve değerini 10 yapalım. Repeat döngüsünün içerisinde rastgele bir şekilde RGB LED modülünde de renk yakan ve aynı şekilde 4 farklı renkten rastgele bir tanesin OLED ekrana yazdıran "define" bloğu bulunacaktır.Bu bloğun adını "random_color" yapalım ve "repeat" bloğunun içerisine sürkleyelim.

```
repeat 10
  random_color
```

Aşağıdaki Bloklar üzerinden “random_color” bloklarını inceleyelim.

```
define random_color
  repeat 10
    set randomColorIdx to random 1 to 4
    set randomColorNameIdx to random 1 to 4
    set colorName to
      item randomColorNameIdx of list RED GREEN BLUE WHITE
    set RGB_color to
      item randomColorIdx of
        list PicoBricks color r 255 g 0 b 0 (0-255)
          PicoBricks color r 0 g 255 b 0 (0-255)
          PicoBricks color r 0 g 0 b 255 (0-255)
          PicoBricks color r 255 g 255 b 255 (0-255)
    PicoBricks set RGB LED color RGB_color
    write colorName at x 45 y 32 inverse
    wait 50 millisecs
```

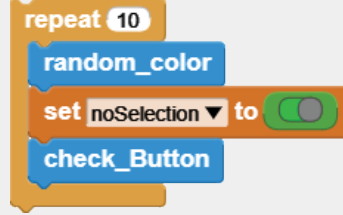
Bu bloğun içerisinde 10 kere tekrar eden bir repeat döngüsü bulunacaktır. Bu Döngünün içerisinde; RGB LED modülünde rastgele 4 renkten bir tanesini yakacağız ve aynı şekilde bu 4 renkten birini rastgele OLED ekran üzerinde yazdıracacağız bunun için;

- “randomColorIdx” adında bir değişken tanımlayalım ve bu değişken RGB LED modülünde 4 renkten bir tanesini rast gee yakmamızı sağlayacak. Bu değişkenin değerini 1 ile 4 arasında rastgele bir sayı yapan bloğu oluşturalım.
- “randomColorNameIdx” adında bir değişken daha oluşturalım. Bu değişken OLED ekran üzerinde belirlenen 4 renkten 1 tanesini rastgele bir şekilde yazdırmamızı sağlayacaktır. Bu değişkene de 1 ile 4 arasında rastgele bir sayı atayalım.
- “colorName” adında bir değişken daha oluşturalım bu değişkenin değerini 4 elemanlı bir liste oluşturup listenin elemanlarını belirlediğimiz 4 renk seçeneğinin adı yapalım. Daha sonra bir öneki maddede oluşturduğumuz değişkenin değerine göre bu listenin bir elemanını “colorName” isimli değişkene atayalım.
- Aynı şekilde “RGB_color” adında bir değişken daha oluşturalım bu değişkenin değerini tanımlayacağımız liste belirleyecek. 4 elemanlı bir liste tanımlayalım ve listenin elemanlarını belirlediğimiz 4 RGB renk değeri yapalım. Daha önce tanımladığımız “randomColorIdx” değişkenin değerine göre “RGB_color” değişkeninin değerini belirleyelim.
- PicoBricks RGB LED Modülünün değerini “RGB_color” değişkeni yapalım.
- OLED ekranda yazacak ifadeye ise “colorName” değişkeni yapalım.
- Bütün bu işlemlerden sonra 50 milisaniye bekleme bloğu ekleyelim.

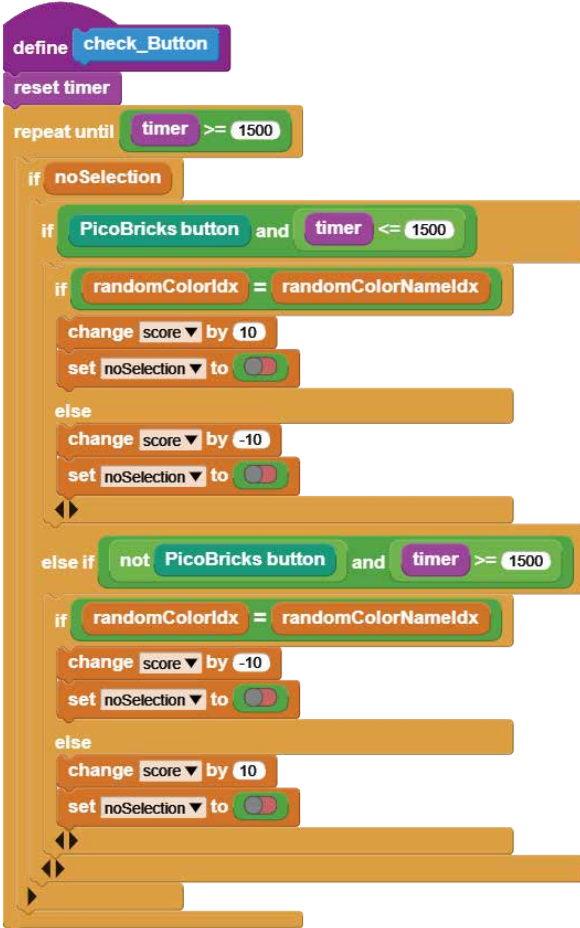
- 7) “noSelection” adında bir değişken daha tanımlayalım e bu değişkenin değerini operatör bloklarından anahtar bloğunu sürükleyelim. Bu anahtarın değerini açık olarak belirleyelim.Bu değişken butona basılıp basılmadığını tespit etmemizi sağlayacak.



- 8) Butona basılıp basılmadığını tespit etmek için “check_Button” adında bir “define” bloğu daha oluşturalım.



Aşağıdaki Bloklar üzerinden “check_Button” bloklarını inceleyelim.



- Oyunda oyuncunun tepki vermesi için 1.5 saniye süre sınırı bulunmaktadır.Bu süreyi başlatmak için ilk önce “timer” ı resetlemek gerekir. Bunun için “reset timer” bloğunu sürükleyelim.

- Oyuncunun butona 1.5 saniye içinde bastığını anlamak için döngüyü timer’ın 1.5 saniyeden büyük olana kadar tekrar ettirmesi gerekir.

- Bu “repeat until” bloğu içerisinde eğer oyuncunun butona basmadığını “noSelection” değişkeni ile kontrol ederiz. Eğer “no selection” ilk tanımladığımız daki gibi açık anahtar değerinde ise;

- 1.5 saniye içerisinde butona basılıp basılmadığını kontrol ederiz. Eğer butona 1.5 saniye içerisinde basıldıysa ve ekranda yazan renk ile RGB de yanan renk aynı ise “score” değişkenini 10 arttırıp “no selection” anahtarını kapatırız. Değilse “score” değişkenini 10 azaltıp “no selection” anahtarını kapatırız.

- Eğer butona basılmadıysa ve süre 1.5 saniyeyi geçtiyse aynı zamanda ekranda yazan renk ile RGB de yanan renk aynı ise “score” değişkenini 10 azaltırız ve “no selection” anahtarını kapatırız.Değilse “score” değişkenini 10 arttırıp.”no selection” anahtarını kapatırız.

9) “check_Button” bloğu çalıştıktan sonra ekranı temizleriz ve RGB LED modülünü kapatırız.

10) Son olarak “score” değişkenini ekrana yazdırırız.

```

when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  PicoBricks turn off RGB LED
  set score to 0
  write The game begins at x 8 y 32 inverse
  wait 2000 milliseconds
  clear
  repeat 10
    random_color
    set noSelection to
    check_Button
    clear
    PicoBricks turn off RGB LED
  wait 1500 milliseconds
  write Your Total Score at x 0 y 24 inverse
  write join score points at x 30 y 40 inverse
  
```

Proje Kodlarımız Hazır

```

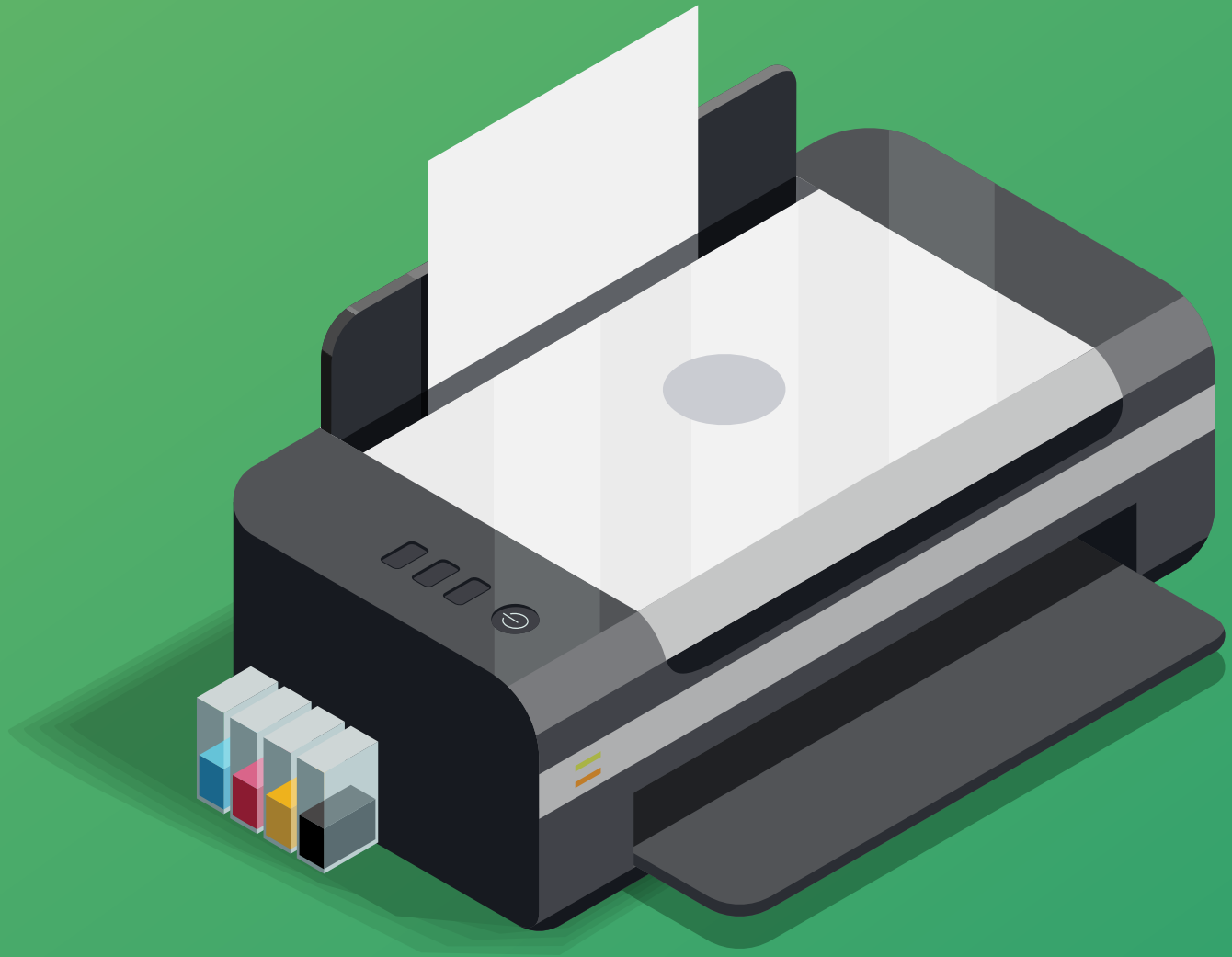
when started
  initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
  PicoBricks turn off RGB LED
  set score to 0
  write The game begins at x 8 y 32 inverse
  wait 2000 milliseconds
  clear
  repeat 10
    random_color
    set noSelection to
    check_Button
    clear
    PicoBricks turn off RGB LED
  wait 1500 milliseconds
  write Your Total Score at x 0 y 24 inverse
  write join score points at x 30 y 40 inverse

define random_color
  repeat 10
    set randomColorIdx to random 1 to 3
    set randomColorNameIdx to random 1 to 4
    set colorName to
    item randomColorNameIdx of list RED GREEN BLUE WHITE
    set RGB_color to
    item randomColorIdx of
    list PicoBricks color r 255 g 0 b 0 (0-255)
    PicoBricks color r 0 g 255 b 0 (0-255)
    PicoBricks color r 0 g 0 b 255 (0-255)
    PicoBricks color r 255 g 255 b 255 (0-255)
  PicoBricks set RGB LED color RGB_color
  write colorName at x 45 y 32 inverse
  wait 50 milliseconds

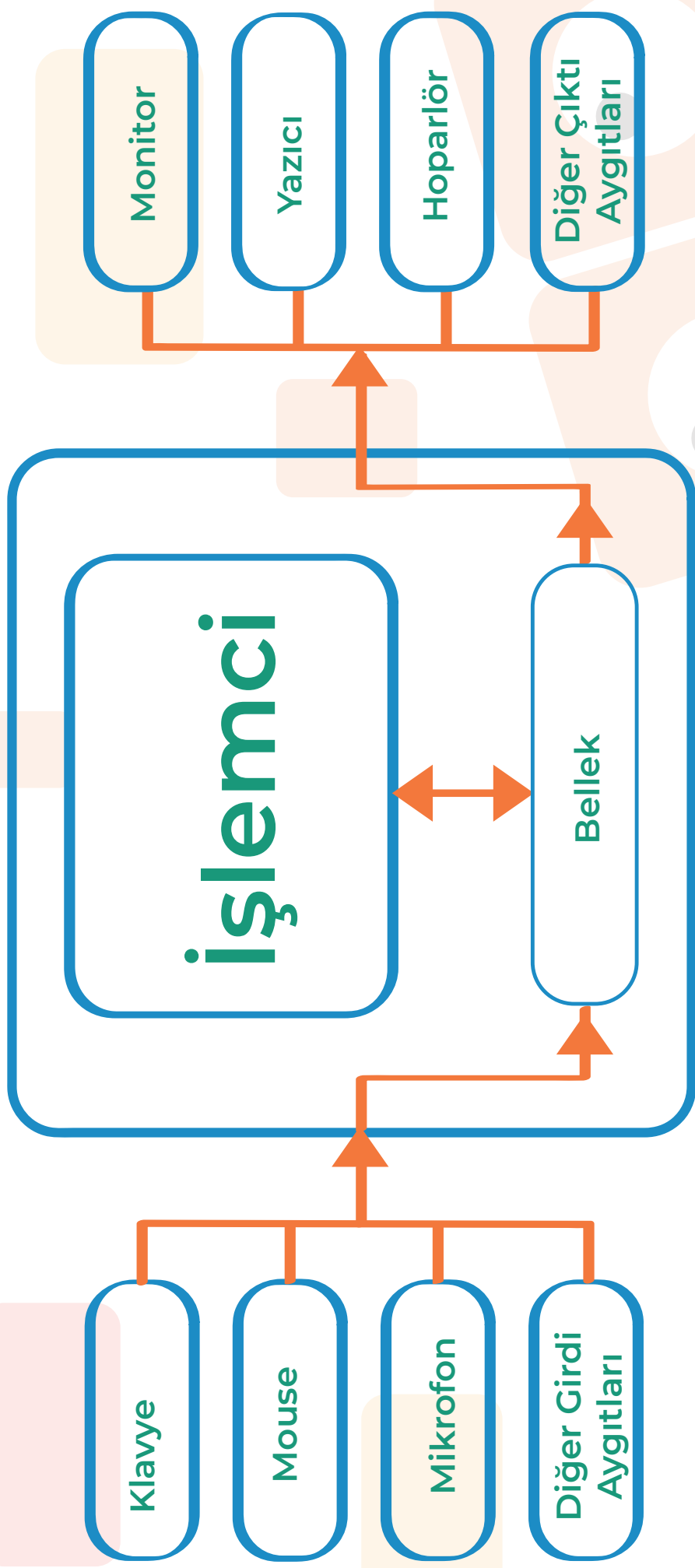
define check_Button
  reset timer
  repeat until timer >= 1500
  if noSelection
    if PicoBricks button and timer <= 1500
      if randomColorIdx = randomColorNameIdx
        change score by 10
        set noSelection to
      else
        change score by -10
        set noSelection to
    else if not PicoBricks button and timer >= 1500
      if randomColorIdx = randomColorNameIdx
        change score by -10
        set noSelection to
      else
        change score by 10
        set noSelection to
  
```



Yazdır ve Sınıfa Dağıt



Fasikül ve Çalışma Kağıdı



1. **İşlemci:** Anakart üzerinden bulunan küçük çiplere işlemci denir. İşlemciyi bilgisayarın beyni olarak düşünebiliriz. İşlemci ne kadar hızlı ise bilgisayarınızda o kadar hızlıdır.

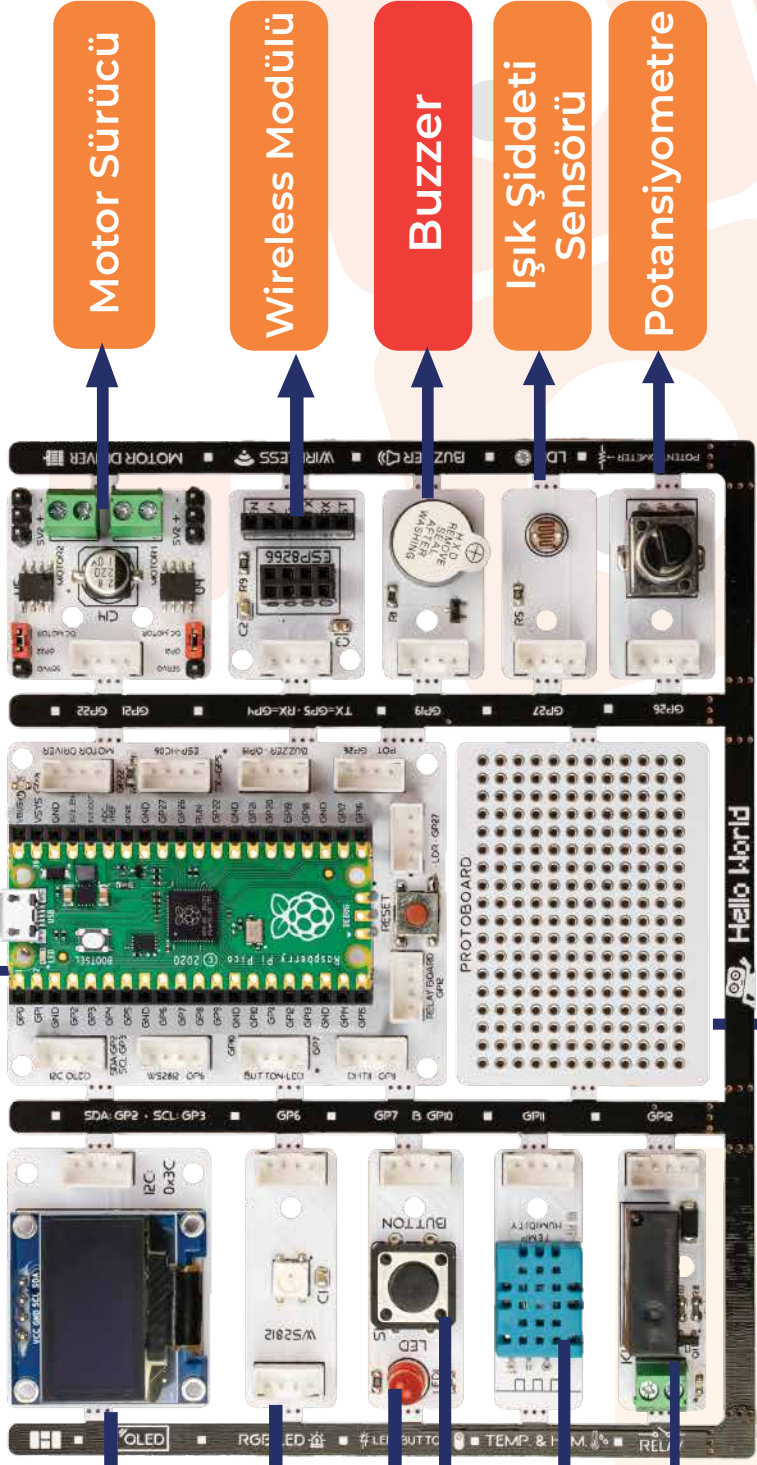
2. **Depolama Birimleri:** Bilgisayarlar depolama birimlerini kullanarak bilgileri saklar. Depolama birimleri kısa süreli ve uzun süreli olmak üzere ikiye ayrılır.

- a) **Kısa Süreli Bellek:** RAM (Rastgele Erişimli Hafıza) bilgisayara iletilen bilgilerin geçici olarak saklandığı yerdir.
- b) **Uzun Süreli Bellek:** LTM (Long-Term Hafıza) Bilgilerin silinmediği kalıcı olarak depo edildiği hafıza birimidir.

3. **Girdi Aygıtları:** Bilgisayarın dışarıdan bilgi almasını sağlayan aygıtlardır. Örneğin kulaklarımız vücudun girdi organlarından biridir. İşittığımız sesleri beynimize yollarlar. Bilgisayarın girdi elemanlarına ise klavye, mouse, mikrofon dokunmatik ekran vb. aygıtlar dışarıdan aldığı bilgileri bilgisayarın işlemcisine yollamızı sağlar.

4. **Çıktı Aygıtları:** Bilgisayarın bilgiyi ilettiği aygıtlardır. Örneğin ağızımız çıktı organlarımızdandır konuşarak ses yardımıyla dışarıya bilgi aktarır. Bilgisayarın çıktı aygıtları Monitör, hoparlör vb. aygıtlardır. Monitör görsel çıktıları, hoparlör ise işitsel çıktıları oluşturur.

Raspberry Pi Pico



Protoboard

OLED Ekran

RGB LED

Kırmızı LED

Buton

Sıcaklık ve Nem Sensörü

Röle

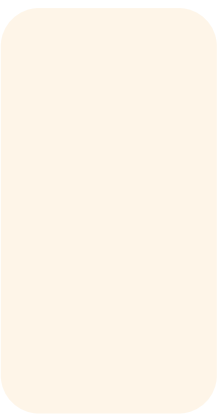
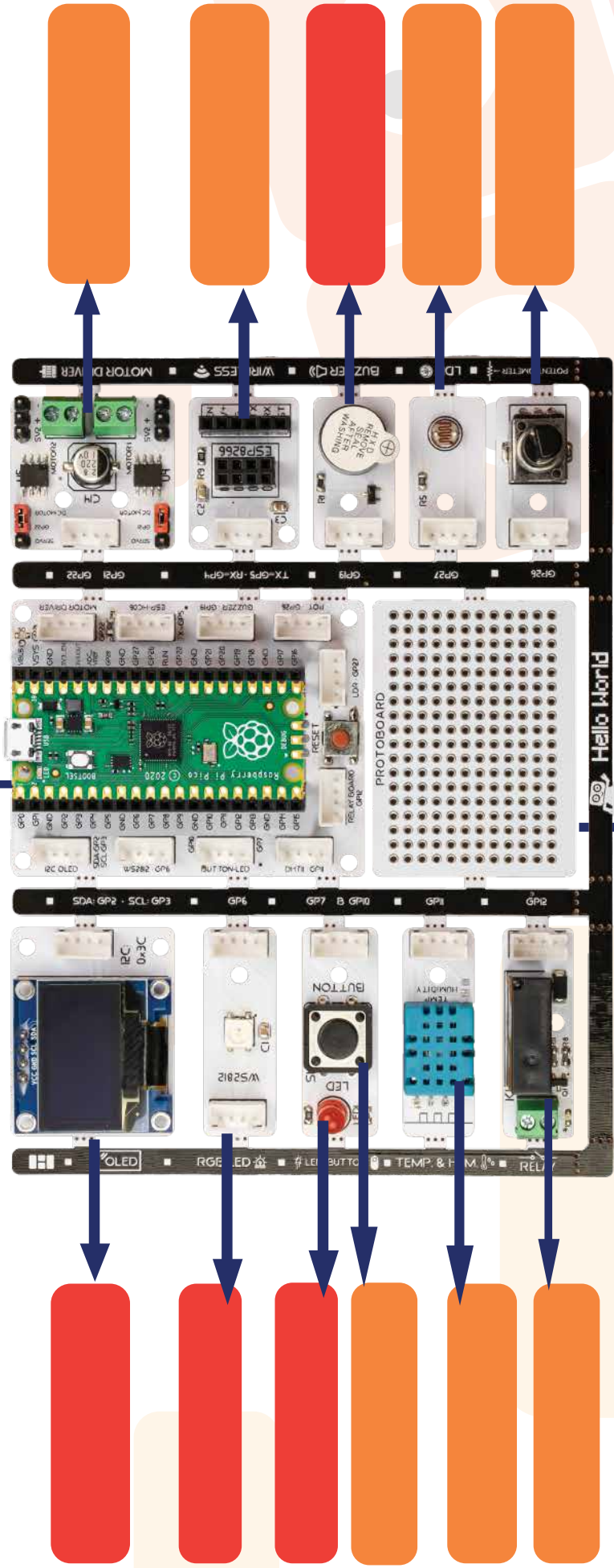
Motor Sürücü

Wireless Modülü

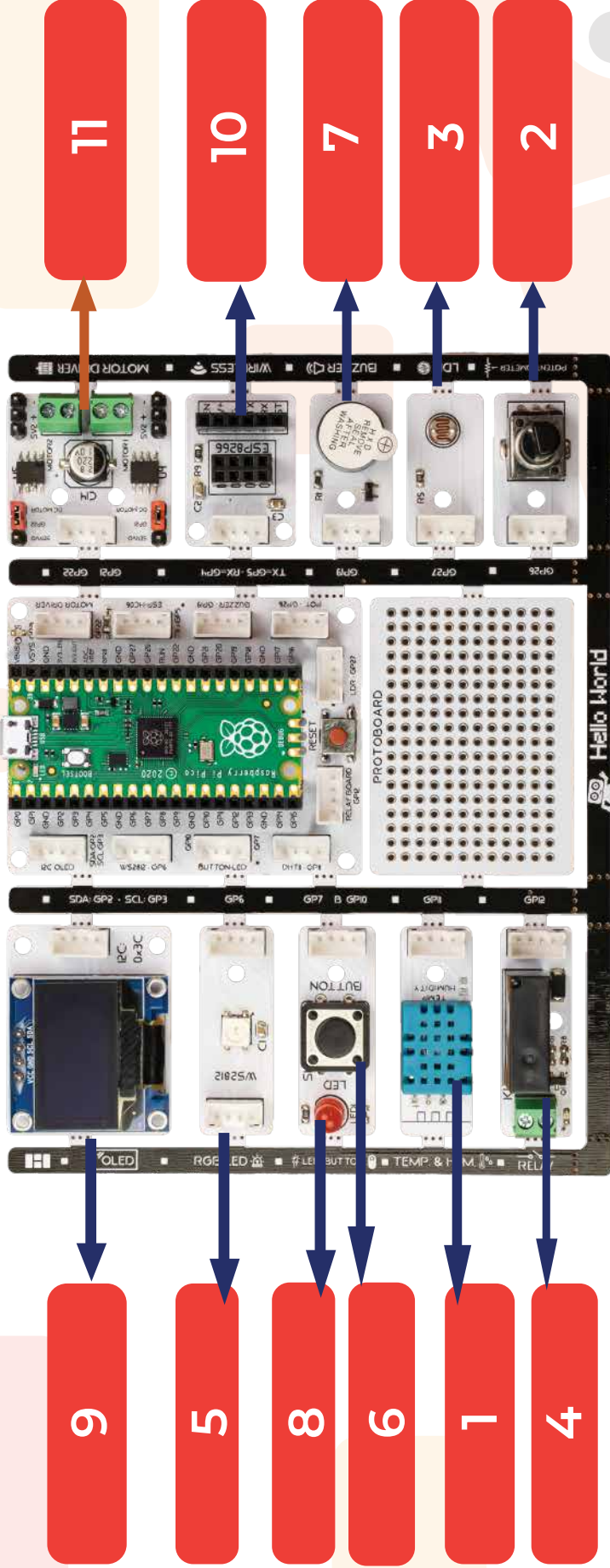
Buzzer

Işık Şiddeti Sensörü

Potansiyometre



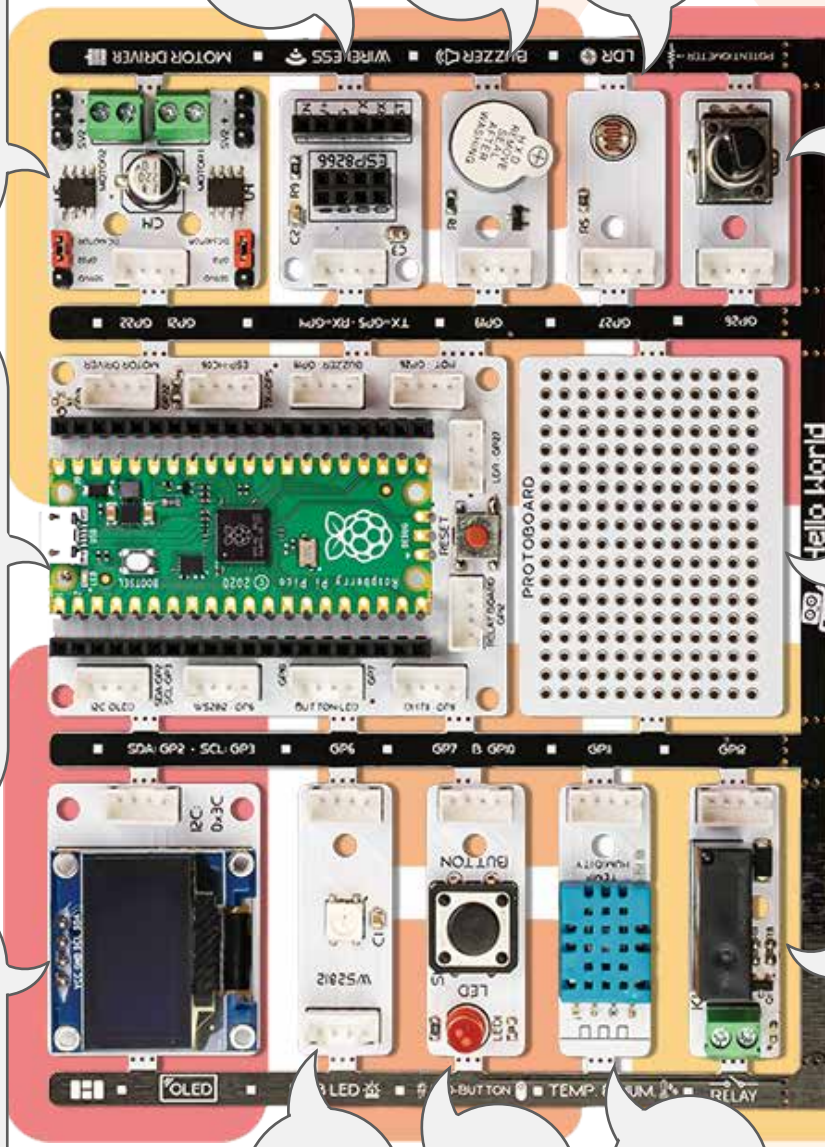
(ipucu: Rakamları farklı en büyük tek sayı)



Çıktı Modülleri

Girdi Modülleri

1. Kırmızı ışık çıkıtsı veren modül
2. Metin, resim veya basit animasyon çıkıtsı veren modül.
3. Ses çıkıtsı veren modül.
4. Kırmızıyeşil, mavi renklerini karıştırarak renkli ışık çıkıtsı ve



Empty speech bubble

Empty speech bubble

Empty speech bubble

Empty speech bubble

Empty speech bubble

Empty speech bubble

Empty speech bubble

Empty speech bubble

Empty speech bubble

Empty speech bubble

Empty speech bubble

Empty speech bubble



Buzzer: Projenizde sesli uyarı vermek istiyorsanız beni kullanmanız lazımdır.



OLED Screen: Projelerinizde görüntü çıktısı almanızı sağlarım



LED: Projelerinizde sadece kırmızı ışıkla bir çıktı almak istiyorsanız beni kullanmanız gerekmektedir.
Buton: Projelerinizde başlatma ya da bitirme olaylarını benim üzerime basarak tetikleyebilirsiniz.



RGB LED: Projelerinizde kırmızı, yeşil ve mavi renklerini belirli oranlarda karıştırarak istediğiniz renk çıktısını oluşturmayı sağlarım.



DHT: Projelerinizde bulunduğunuz ortamın ısı ve nem değerini kullanmak istiyorsanız beni kullanmanız gerekmektedir.



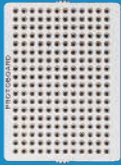
Role Modülü: Projelerinizde elektrik akımını keserek ya da akımın devrede dolaşmasına izin vererek devreyi açıp kapatmanızı sağlarım



LDR Sensör: Projelerinizde ortamdaki ışık miktarını algılamak istiyorsanız beni kullanmanız gerekir.



PicoBricks Modülü: Projelerinizi kodlamak için beni kullanırsınız ayrıca kullandığınız modüllerin ve sensörlerin pin bağlantılarını benim üzerimdeki pinlerden gerçekleştirebilirsiniz. Bi benzetme yapacak olursak PicoBricksin beyni benim.



Protoboard: Üzerimde bulunan iletken yollar ile devreye herhangi biriletken maddeyi bağlayıp kod içerisinde bu maddeye değer atama işlemini benim sayemde yaparsınız.



Motor Sürücü: Projelerinizde DC motor veya servo motor kullanmak istiyorsanız servo motor ve dc motorun hızını, frekansını ayarlamak için devrede benide kullanmanız gerekir.



Wireless Modülü: Projelerinizde bluetooth veya wi-fi ile haberleşme sağlamka istiyorsanız beni kullanmanız gerekir.



Potansiyometre: Benimle beraber fiziksel bir müdahale ile devre elemanlarına giden akıma farklı büyüklüklerde direçn uygulayabilirsiniz.

Çıktı Blokları: Sesli ve görüntülü çıktı aygıtlarını yönetebildiğimiz bloklar bulunur.

Pin Blokları: Picobricks pinlerine sensör bağladığımızda buradaki blokları kullanabiliriz.

İşlem Blokları: Matematiksel işlemlerin bulunduğu kod blokları.

Veri Blokları: Projeye tanımladığımız verileri kullanmamızı sağlayan kod blokları bulunur.



Output

Input

Pins

Control

Operators

Variables

Data

My Blocks

Libraries +

OLED Graphics

PicoBricks

Temperature Hu

Girdi Blokları: Girdi aygıtlarını yönetebildiğimiz kod blokları bulunur.

Kontrol Blokları: Başlama, bitirme, koşul, döngü vb. olayları kontrol edebildiğimiz kod blokları bulunur.

Değişken Blokları: Değişken tanımlama, değışkene değer atama vb. işlemlerin yapıldığı kod blokları bulunur.

Bloklarım: Kendi kod bloklarımızı hazırlayıp görev tanımlayabileceğimiz kod blokları bulunur.

Kütüphaneler: Kütüphaneleri Microblocks IDE editörüne yükledikten sonra içerisindeki kod bloklarına erişebildiğimiz bölüm.

when PicoBricks button

initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip

play note A octave 0 for 1000 ms

write A at x 0 y 0 inverse

PicoBricks set RGB LED color

play note E octave 0 for 500 ms

write E at x 0 y 0 inverse

PicoBricks set RGB LED color

play note E octave 0 for 500 ms

PicoBricks set RGB LED color

play note E octave 0 for 500 ms

PicoBricks set RGB LED color

play note E octave 0 for 500 ms

PicoBricks set RGB LED color

play note E octave 0 for 500 ms

PicoBricks set RGB LED color

play note E octave 0 for 500 ms

PicoBricks set RGB LED color

play note D octave 0 for 500 ms

write D at x 0 y 0 inverse

PicoBricks set RGB LED color

play note F octave 0 for 500 ms

write F at x 0 y 0 inverse

PicoBricks set RGB LED color

play note E octave 0 for 1000 ms

write E at x 0 y 0 inverse

PicoBricks set RGB LED color

play note F octave 0 for 500 ms

write F at x 0 y 0 inverse

PicoBricks set RGB LED color

play note F octave 0 for 500 ms

write F at x 0 y 0 inverse

PicoBricks set RGB LED color

play note F octave 0 for 500 ms

write F at x 0 y 0 inverse

PicoBricks set RGB LED color

Butona bastıktan sonra ekranı ve diğerkod bloklarını çalıştırmamız için bu iki kod bloğunu proje sayfamıza sürüklememiz gerekmektedir.

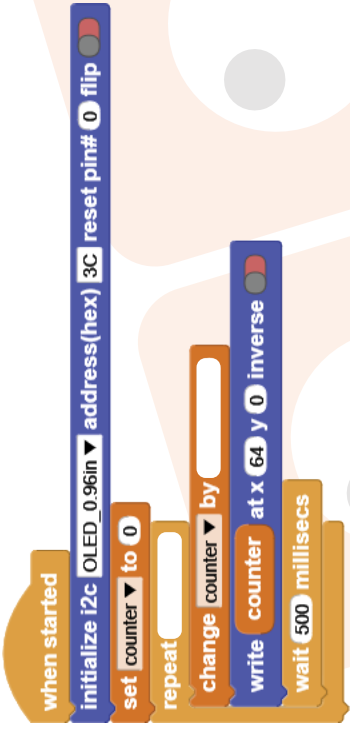
Melodi için gerekli tonların, buzzerdan çalmasını, ton değerinin ekranda yazmasını ve RGB LED modülünden renk çıktısını almayı sağlayan kod bloklarıdır. Bu blokları kopyalayıp içerisindeki değerleri değiştirerek projemizi tamamlayabiliriz.

1.Çalışma Kağıdı (Döngüler)

1. Aşağıdaki kod bloğunu çalıştırdığımızda ekranda 0 ile 5 arasında sayıların yazılmasını istiyoruz. Bunu sağlayacak şekilde boş bırakılan alanları doldurunuz.



2. 0 ile 10 arasındaki sayıları 2'şer arttıran Microblocks kodu için boş bırakılan alanları doldurunuz.



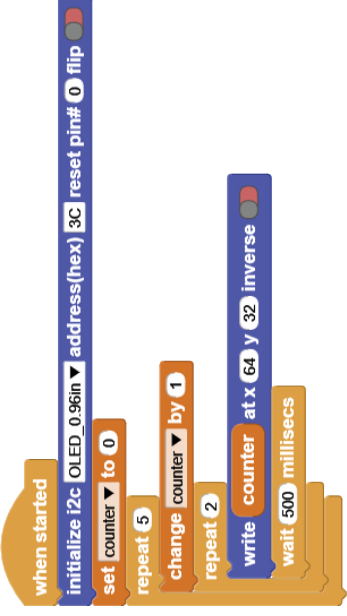
3. Aşağıdaki kod bloklarını Microblocks IDE editörüne yazdıktan sonra ekranda sırasıyla ilk ve son yazan rakam, aşağıdaki şıklardan hangisinde doğru verilmiştir.



- A) 8 - 2 B) 8 - 0 C) 10 - 0 D) 4 - 2

2.Çalışma Kağıdı

1. TAşağıda iki ayrı kod bloğu verilmiştir. Aşağıdaki şıklardan hangisinde bu kod bloklarının OLED ekranda yazdığı son sayı sırası ile doğru bir şekilde verilmiştir?



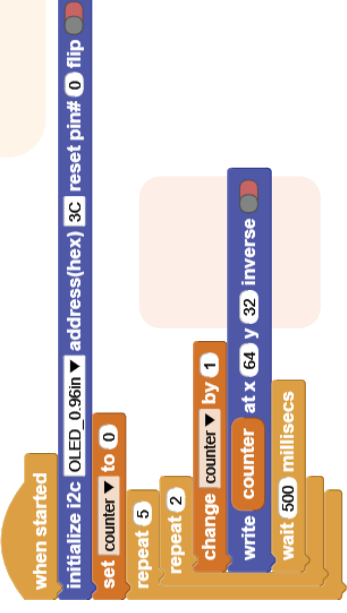
```
when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
set counter to 0
repeat 5
change counter by 1
repeat 2
write counter at x 64 y 32 inverse
wait 500 milliseconds
```

A) 5-1

B) 2-5

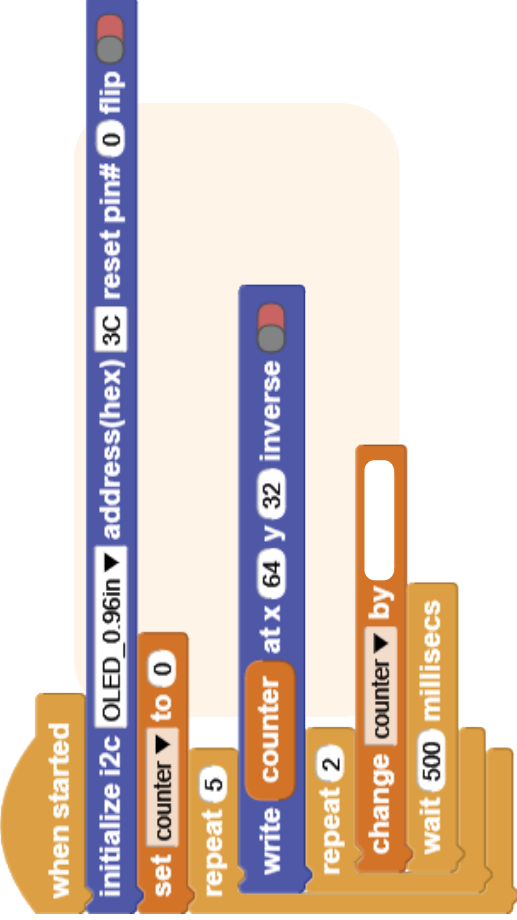
C) 5-10

D) 10-5



```
when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
set counter to 0
repeat 5
repeat 2
change counter by 1
write counter at x 64 y 32 inverse
wait 500 milliseconds
```

2. Aşağıdaki kod bloğunda verilen boşluğa hangi sayı gelirse OLED ekranda 0 ile 10 arasındaki sayıları (10 dahil değil) 2 şer 2 şer artarak yazar. Boşluğu doldurunuz.



```
when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
set counter to 0
repeat 5
write counter at x 64 y 32 inverse
repeat 2
change counter by
wait 500 milliseconds
```

3. Yanda verilen kod bloğu çalıştırıldığında OLED ekranda yazan ifade aşağıdakilerden hangisidir?

A) $10 \times 2 = 20$

B) $2 \times 20 = 40$

C) $20 \times 10 = 200$

D) $10 \times 20 = 200$



```
when started
initialize i2c OLED_0.96in address(hex) 3C reset pin# 0 flip
set counter to 0
set counter2 to 0
repeat 10
change counter by 1
repeat 2
change counter2 by 1
write counter at x 0 y 25 inverse
write X at x 20 y 25 inverse
write counter2 at x 40 y 25 inverse
write = at x 60 y 25 inverse
write counter * counter2 at x 70 y 25 inverse
```

- Aşağıdaki çalışma kağıdındaki soruları çözünüz.

	1	2	3	4	5	6
A						
B						
C						
D						
E						
F						

Yandaki tabloya göre aşağıdaki koordinat noktalarından hangisi turuncu renkle boyanmamıştır. Yanında verilen daireyi boyayarak belirtiniz.

- A5
- B3
- C3
- B4
- D4
- A2
- D2
- C4

	1	2	3	4	5	6
A						
B						
C						
D						
E						
F						

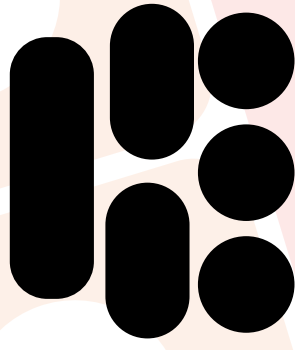
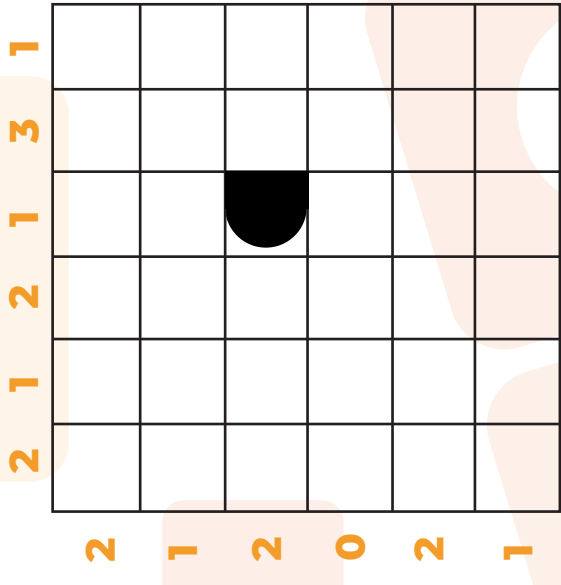
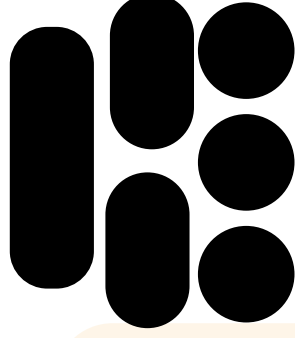
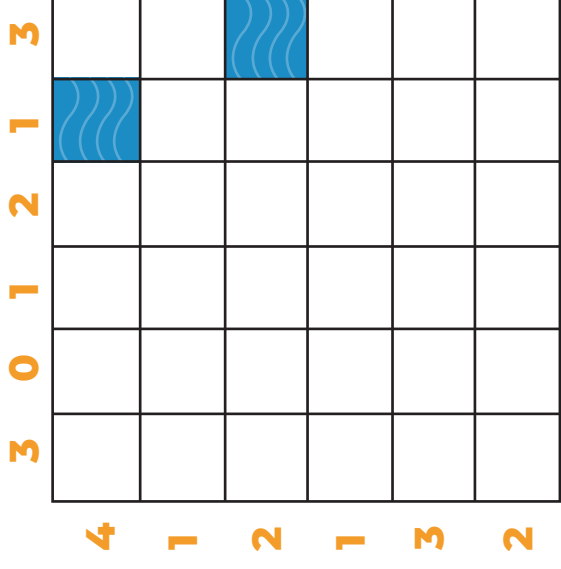
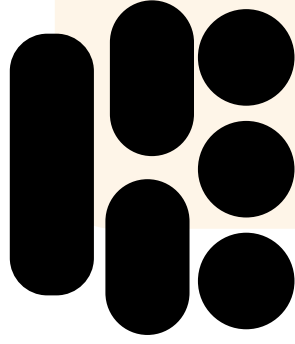
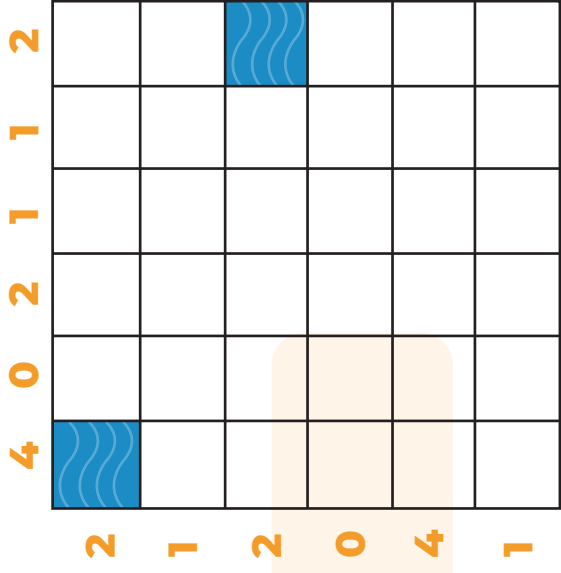
Yandaki tabloya göre aşağıda verilen koordinat noktalarında boş bırakılan alanları doldurunuz.

- D_
- C_
- C5
- B_
- E_
- D3
- E3
- F_

	1	2	3	4	5	6
A						
B						
C						
D						
E						
F						

Aşağıda verilen koordinat noktalarını yandaki oyun tablosunda doğru yerleri boyayarak belirtiniz.

- A2
- B2
- E3
- E4
- D2
- D5
- A5
- B5





GALILEO GALILEI

"We cannot teach people anything; we can only help them discover it within themselves."