

 pico
bricks

IDE

 PicoBricks

 On IR Receiving

 Read Potentiometer

 Set Relay

 Read Humidity

 Read Button

Project Book

 IR Button is Pressed

 Read Distance Trig Echo

 Play Buzzer Freq

 Read Light Sensor

 Read Temperature (C)

 Set Led



Copyright © 2022 Robotistan

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

İçerikler: Mustafa Kemal Avcı, Abdullah Kaya, Selim Gayretli

Grafik Tasarımcı: Ahmet Gürsu, Elanur Tokalak

Editör: Naze Gizem Özer

PicoBricks AR-GE EKİBİ

Yasir Çiçek - CEO

Yusuf Gündoğdu - Yazılım Mühendisi

Mehmet Suat Morkan - Proje Yöneticisi

Mehmet Ali Dağ - Donanım Mühendisi





İçindekiler

Kodlama Platformları	9
PicoBricks IDE	9
Pico JR	10
PicoBlockly	11
Pico Py	11
PicoBricks Simulator	11
PROJELER	17
Blink	18
Proje Detayları ve Algoritma	18
Bağlantı Şeması	18
Proje Görselleri	19
Proje Önerisi	19
PicoBricks IDE Kodlama	19
Action - Reaction	21
Proje Detayları ve Algoritma	21
Bağlantı Şeması	21
Proje Görselleri	22
Proje Önerisi	22
PicoBricks IDE Kodlama	22
Autonomous Lighting	24
Proje Detayları ve Algoritma	24
Bağlantı Şeması	24
Proje Görselleri	25
Proje Önerisi	25
PicoBricks IDE Kodlama	25
Thermometer	28
Proje Detayları ve Algoritma	28
Bağlantı Şeması	28
Projenin Kurulumu	29
Proje Önerisi	29
PicoBricks IDE Kodlama	29
Graphic Monitor	31
Project Details and Algorithm	31
Bağlantı Şeması	31
Proje Görselleri	32
Proje Önerisi	32
PicoBricks IDE Kodlama	32
Dominate the Rhythm	34
Proje Detayları ve Algoritma	34
Bağlantı Şeması	35
Proje Görselleri	35
Proje Önerisi	35
PicoBricks IDE Kodlama	35

Show Your Reaction	39
Proje Detayları ve Algoritma	39
Bağlantı Şeması	39
Proje Görselleri	40
Proje Önerisi	40
PicoBricks IDE Kodlama	40
My Timer	43
Proje Detayları ve Algoritma	43
Bağlantı Şeması	43
Projenin Kurulum Adımları	44
Proje Önerisi	44
PicoBricks IDE Kodlama	44
Alarm Clock	48
Proje Detayları ve Algoritma	48
Bağlantı Şeması	48
Proje Görselleri	49
Proje Önerisi	49
PicoBricks IDE Kodlama	49
Know Your Color	51
Proje Detayları ve Algoritma	51
Bağlantı Şeması	51
Proje Görselleri	52
Proje Önerisi	52
PicoBricks IDE Kodlama	52
Magic Lamp	55
Proje Detayları ve Algoritma	55
Bağlantı Şeması	55
Projenin Kurulum Adımları	56
Proje Önerisi	56
PicoBricks IDE Kodlama	56
Smart Cooler	58
Proje Detayları ve Algoritma	58
Bağlantı Şeması	58
Proje Görselleri	59
PicoBricks IDE Kodlama	59
Buzz Wire Game	61
Proje Detayları ve Algoritma	61
Bağlantı Şeması	62
Projenin Kurulum Adımları	62
Proje Görselleri	62
PicoBricks IDE Kodlama	62
Dinosaur Game	64
Proje Detayları ve Algoritma	64
Bağlantı Şeması	65
Proje Görselleri	65

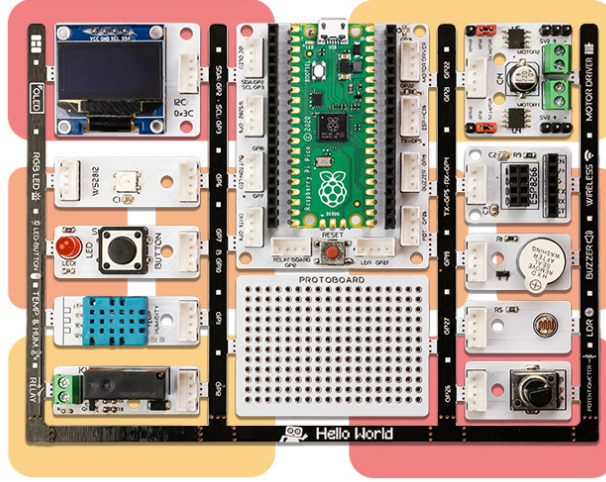
Projenin Kurulum Adımları	66
Proje Önerisi	66
PicoBricks IDE Kodlama	66
Night and Day	67
Proje Detayları ve Algoritma	67
Bağlantı Şeması	68
Proje Görselleri	68
Projenin Kurulum Adımları	69
PicoBricks IDE Kodlama	69
Voice Controlled Robot Car	71
Proje Detayları ve Algoritma	71
Projenin Kurulum Adımları	71
Projenin Kurulum Adımları	72
PicoBricks IDE Kodlama	72
Two Axis Robot Arm	74
Proje Detayları ve Algoritma	74
Bağlantı Şeması	75
Projenin Kurulum Adımları	75
Projenin Önerisi	75
PicoBricks IDE ile Kodlanması	80
Smart House	82
Proje Detayları ve Algoritma	82
Bağlantı Şeması	82
Projenin Kurulum Adımları	83
Projenin Önerisi	83
PicoBricks IDE ile Kodlanması	85
Hungry Piggy Bank	86
Proje Detayları ve Algoritma	86
Bağlantı Şeması	86
Projenin Kurulum Adımları	87
Projenin Önerisi	87
PicoBricks IDE ile Kodlanması	90
Automatic Trash Bin	91
Proje Detayları ve Algoritma	91
Bağlantı Şeması	91
PicoBricks IDE ile Kodlanması	92
Digital Ruler	93
Proje Detayları ve Algoritma	93
Bağlantı Şeması	93
Projenin Önerisi	93
PicoBricks IDE ile Kodlanması	94



Air Piano	95
Proje Detayları ve Algoritma	95
Bağlantı Şeması	95
Projenin Önerisi	95
PicoBricks IDE ile Kodlanması	96
Maze Navigating Robot	97
Proje Detayları ve Algoritma	97
Bağlantı Şeması	98
Projenin Önerisi	98
PicoBricks IDE ile Kodlanması	98
Smart Greenhouse	100
Proje Detayları ve Algoritma	100
Bağlantı Şeması	101
Projenin Önerisi	101
PicoBricks IDE ile Kodlanması	101
Kaynaklar	
Robot Araba Kurulum Kılavuzu	106
Akıllı Sera Kurulum Kılavuzu	112
3D Modeller	115

PicoBricks Nedir?

PicoBricks, maker projelerinde kullanılmak üzere tasarlanmış bir elektronik geliştirme kartı ve yazılımıdır. On adet ayrılabilir modülü olan PicoBricks ile çeşitli projeler yaratabilirsiniz. Ayrıca kendi modüllerinizi eklemek için kullanabileceğiniz bir alan da bulunur!



PicoBricks elektronik için , PicoBricks'in m

ve simülatörü sayesinde başlangıç yapmak oldukça kolay olacaktır. Tecrübesi olanlar elektroniği daha derinlemesine inceleyebilir veya Python'da kodlama yapmayı geliştirebilir. Uzman makerlar bile PicoBricks ile oldukça hızlı prototipler yapılmasını beğeneceklerdir.

Diğer geliştirme boardlarından farklı olarak PicoBricks, her seviyedeki makerlar için inanılmaz bir esnekliğe sahiptir! Bricks IDE'de farklı projeler ve senaryolar için örnek kodlar vardır.

MicroBlocks ve PicoBricks'in sürükle-bırak özelliği ile sıfırdan en iyiye kodlamayı öğrenin. MicroBlocks şimdiye kadar yaratılmış ve maker endüstrisinde yaygın olarak bilinen en kolay kodlama deneyimini sunar.

Bir sorun mu var? support@robotistan.com adresine yazabilirsin.



KODLAMA PLATFORMLARI

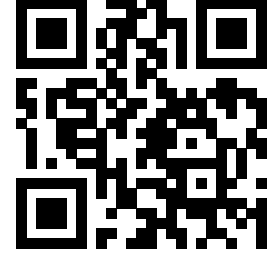
1. Kodlama Platformları

Bilgisayarda kod yazmak için programcılar genellikle bir yazılıma ihtiyaç duyarlar. İyi bir kodlama ortamı, kod yazan kişiye öneriler sunmalıdır. Ayrıca kodunu çalıştırdıktan sonra bir hata varsa bunu programcıya göstermeli ve hata hakkında ayrıntılı bilgiler sunmalıdır. Picobricks'i hem metin tabanlı hem de blok tabanlı editörler kullanarak kodlayabilirsin. PicoBricks IDE, python kodlama diline geçmeden önce bloklarla algortima kurmayı ve kod yazmanın detaylarını öğrenebileceğiniz bir editördür. PicoBricks IDE ile oluşturduğunuz bütün blok kodların python dilindeki karşılığını anında alabilirsiniz. MicroBlocks, Picobricks'i bloklarla kodlayabileceğin güçlü özellikleri olan bir editördür. Micropython öğrenmeye yeni başlayanlar için Thonny editörü en doğru seçimlerden biridir. İster sadece MicroPython kodla istersen bir elektronik devre kartını kodla Thonny birçok desteği sana sağlamaktadır. Arduino C diline hakimsen ya da C dilini öğrenmek istiyorsan Picobricks ve Arduino IDE senin için çok iyi seçimler olacaktır.



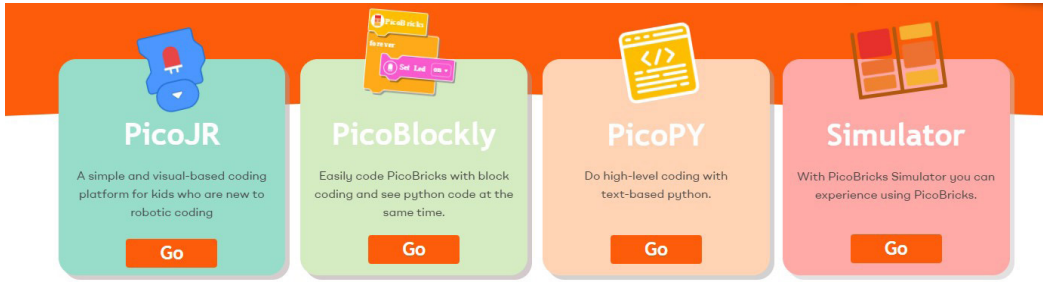
PicoBricks IDE

PicoBricks IDE, PicoBricks ile beraber MicroPython programlama diline, geçmeden önce bloklar ile kodlar yazarak projeler gerçekleştirebileceğin programlama editörüdür. PicoBricks IDE de blokları sürükleyerek oluşturduğunuz her kodun, MicroPython karşılığını alabilirsiniz. Ayrıca offline editörlere bağlı kalmadan PicoBricks IDE'nin içerisinde Python editörüne girerek python dilinde projeler yapabilirsiniz. PicoBricks IDE online ortamda hem blok tabanlı hem de metin tabanlı programlama yaparak kodlar yazabileceğiniz bir platformdur. PicoBricks IDE gerçek zamanlı çalışan online programlama editördür. Projeniz için oluşturduğunuz kod bloklarını "Run" butonuna basarak çalıştırabilir. "Stop" butonu ile durdurabilirsiniz. PicoBricks IDE ile hazırladığınız proje kodlarını Raspberry Pi Pico 'nun içerisine yükleyerek harici bir güç kaynağıyla PicoBricks'i çalıştırabilirsin. PicoBricks IDE'de de küçük yaş grubundaki kullanıcılar için yatay blok yapısında bulunmaktadır. Ayrıca PicoBricks IDE ile elinizde hiç PicoBricks olmasada PicoBricks Simülatör'ü kullanarak blok kodlama yaparak projeler oluşturabilirsiniz.



PicoBricks'i, PicoBricks IDE ile kodlamak için QR kodu taratabilirsiniz veya tarayıcıda <http://rbt.ist/ide> açabilirsiniz.

Bu sayfada, PicoBricks IDE'de kodlama yapabileceğiniz PicoJR, PicoBlockly, PicoPY ve PicoBricks Simülör Editörleri bulunur.

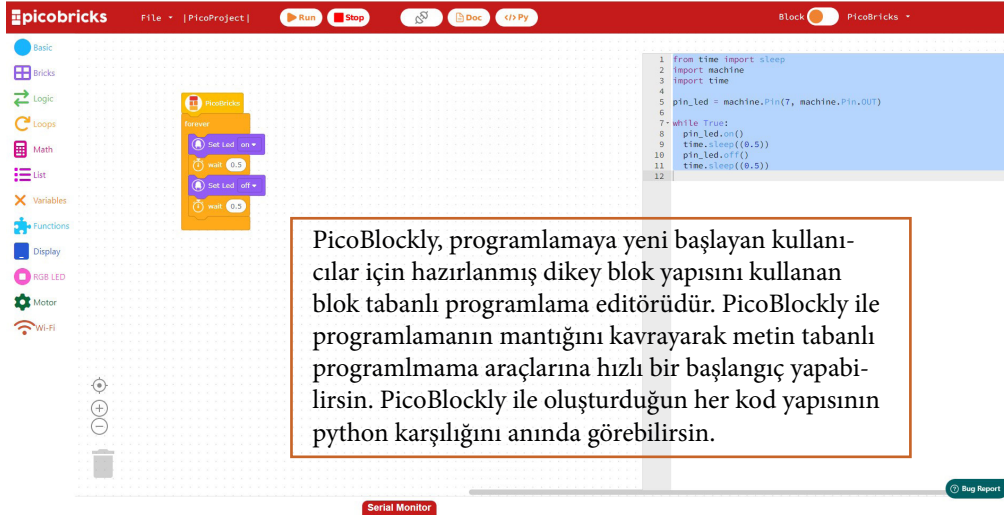


Hadi editörü tanıyalım!

PicoJR



PicoBlockly



```

1 from time import sleep
2 import machine
3 import time
4
5 pin_led = machine.Pin(7, machine.Pin.OUT)
6
7 while True:
8     pin_led.on()
9     time.sleep((0.5))
10    pin_led.off()
11    time.sleep((0.5))
12
  
```

PicoBlockly, programlamaya yeni başlayan kullanıcılar için hazırlanmış dikey blok yapısını kullanan blok tabanlı programlama editörüdür. PicoBlockly ile programlamanın mantığını kavrayarak metin tabanlı programlamaya araçlarına hızlı bir başlangıç yapabilirsiniz. PicoBlockly ile oluşturduğunuz her kod yapısının python karşılığını anında görebilirsiniz.

PicoPY



```

1 from time import sleep
2 import machine
3 import time
4
5 pin_led = machine.Pin(7, machine.Pin.OUT)
6
7 while True:
8     pin_led.on()
9     time.sleep((0.5))
10    pin_led.off()
11    time.sleep((0.5))
  
```

PicoPY, MicroPython diliyle kodlar yazabileceğiniz metin tabanlı bir programlama editörüdür. Herhangi bir offline programlama editörüne gerek kalmadan PicoPY ile MicroPython dilinde kod yazabilirsiniz.

PicoBricks Simülator

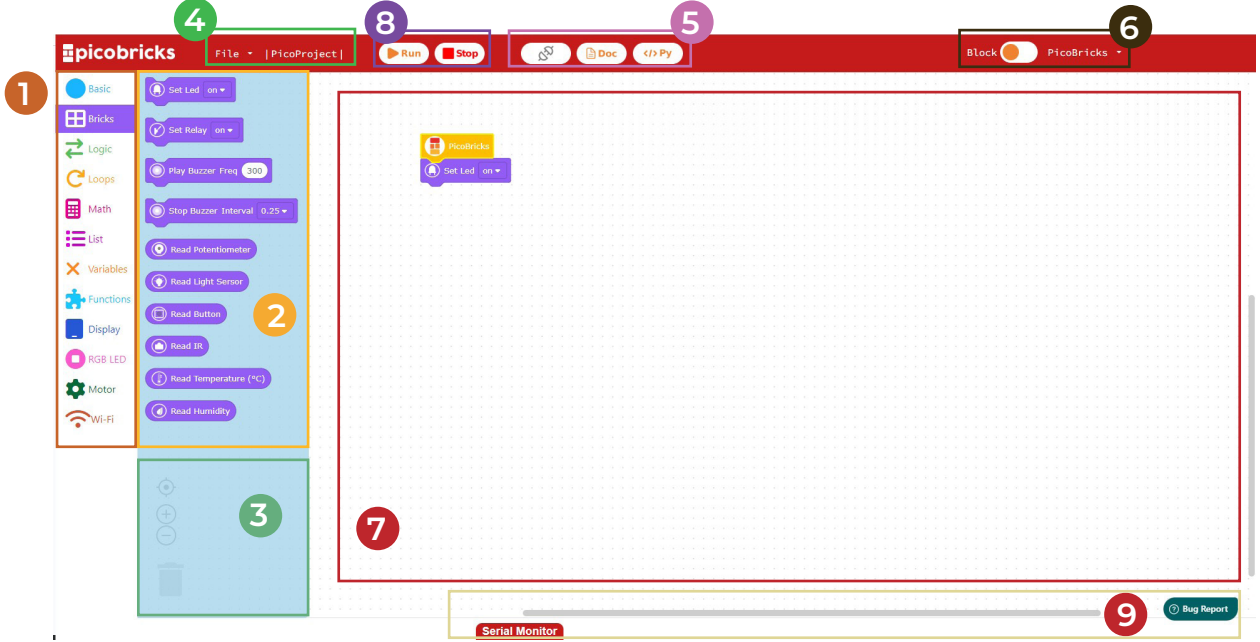


PicoBricks Simulator, fiziksel olarak PicoBricks kullanmadan simülasyon ortamında blok tabanlı programlama yaparak proje geliştirebileceğiniz PicoBricks Editörüdür.

PicoBlockly'yi Tanıyalım

PicoBlockly, dikey blok sistemine sahip blok tabanlı PicoBricks programlama editörüdür. Proje kitabında yapılan örnekler, PicoBlockly editörü kullanılarak hazırlanmıştır.

Arayüz Tanıtımı



1. Blok Kategorileri : PicoBricks'i programlamak için kullanılan blokların kategorilere ayrılarak listelendiği alandır. Bu alanda, PicoBricks sensörlerini kodlamak için gerekli olan sensör blokları, koşul blokları, döğü blokları vb. programlama yapılarının bloklarına ilgili yapının üstüne tıklayarak ulaşabilirsiniz. Bu alanda her kategori farklı bir renk ve ikonla birbirinden ayrılmıştır.. Kategorilerdeki bloklar, 2 numaralı blok paleti alanında listelenecektir.

2. Blok Paleti: Seçilen blok kategorisine göre blokların listelendiği alandır. Buradan seçilen kategoriye göre PicoBricks sensör blokları, if-else vb. koşul yapıları, döğü yapıları vb. programlama fonksiyonların blok karşılıkları bulunmaktadır.

3. Araç Paneli: 7 numaralı kodlama alanındaki kod bloklarını yakınlaştırma, uzaklaştırma, ortalama ve silmeyi sağlayan araçların bulunduğu alandır.

4. Dosya İşlemleri Paneli: Bu alanı kullanarak, yeni bir proje sayfası ve daha önceden PicoBricks IDE ile yaptığınız projeleri açabilir, blokları kullanarak oluşturduğunuz proje kodlarını bilgisayara ya da Raspberry Pi Pico W içine kaydedebilirsiniz.

5. Bağlantı ve Dokümantasyon Paneli: Bu alanı kullanarak PicoBricks IDE ile PicoBricks'in bağlantısını yapabilir, geliştirilen blok kodların MicroPython çıktısına ve PicoBricks projelerinin dökümanlarına ulaşabilirsiniz.

6. Yazılım Yükleme ve IDE Değişim Alanı: Bu alanda, PicoBricks 'in kullandığı Raspberry Pi Pico mikro denetleyicilerinin firmware dosyalarını ve MicroPython programlama dili ile hazırlanmış PicoBricks kütüphanesini yükleyebilir, PicoBlockly ve PicoPy editörleri arasında hızlıca geçiş yapabilirsiniz.

7. Kodlama Alanı: Burası, projelerinizin blok kodlarını sürükle-bırak yaparak hazırlayacağınız alandır. Projeniz için gerekli olan algoritmayı PicoBricks IDE blokları ile hazırlayarak bu alana sürükleyiniz.

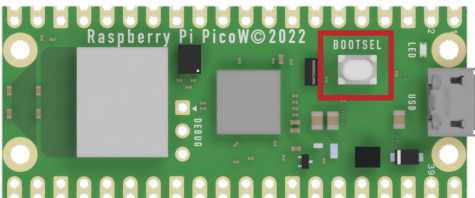
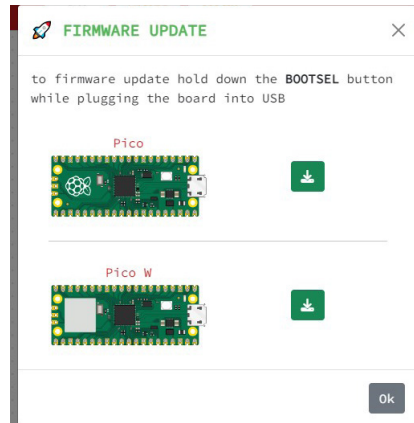
8. Start & Stop Alanı: Hazırladığınız kod bloklarını buradaki butonları kullanarak çalıştırabilir ya da durdurabilirsiniz.

9. Serial Port: Bu alanda serial ekrana yazdırmak istediğiniz değerleri görebilirsiniz.

PicoBricks IDE - PicoBricks Bağlantısı ve Çalıştırma



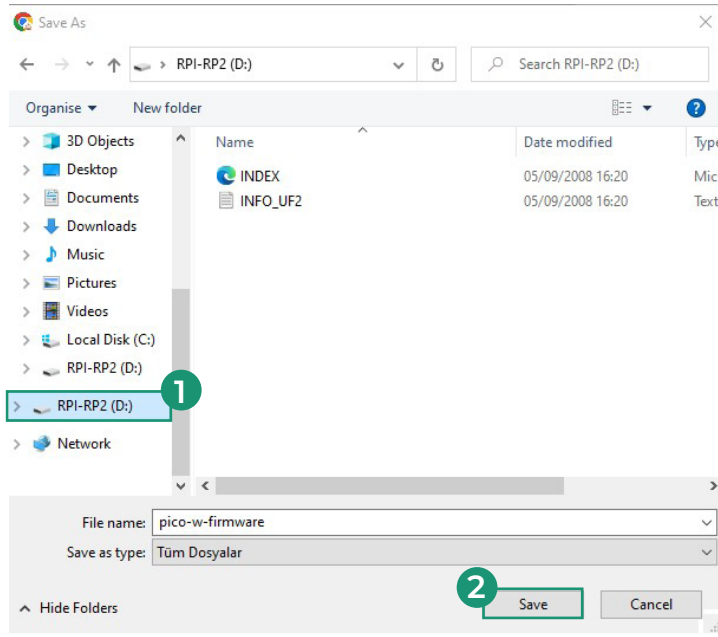
PicoBricks IDE'yi açtıktan sonra PicoBricks menüsüne tıklayınız. PicoBricks menüsünden "Upload Firmware" seçeneğine tıklayınız. Açılan pencerede PicoBricks'e Bootsel işlemi yapmanız istenecektir.



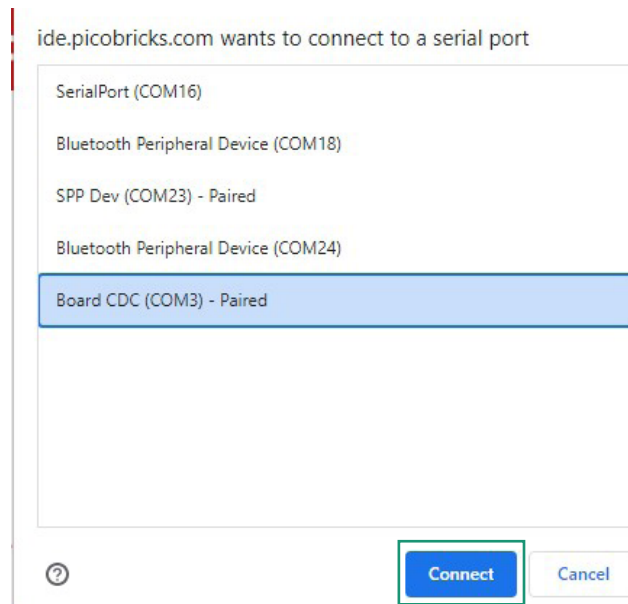
PicoBricks'i PicoBricks IDE'ye bağlamak için Raspberry Pi Pico üzerindeki beyaz BOOTSEL düğmesini basılı tutarken kartı USB kablo ile bilgisayarınıza bağlamalısınız (Farklı bir IDE'ye geçiş yapmayacaksanız bu işlemi bir kere yapmanız yeterli.)

PicoBricks'e Bootsel yaptıktan sonra açılan ilk pencereyi kapatabilirsiniz. PicoBricks üzerinde takılı olan Raspberry Pi Pico modülünü seçerek doğru Firmware dosyasını yükleyiniz.

Firmware dosyasını aşağıdaki adımları takip ederek “RPI-RP(D:)” adlı sürücüye yükleyiniz.

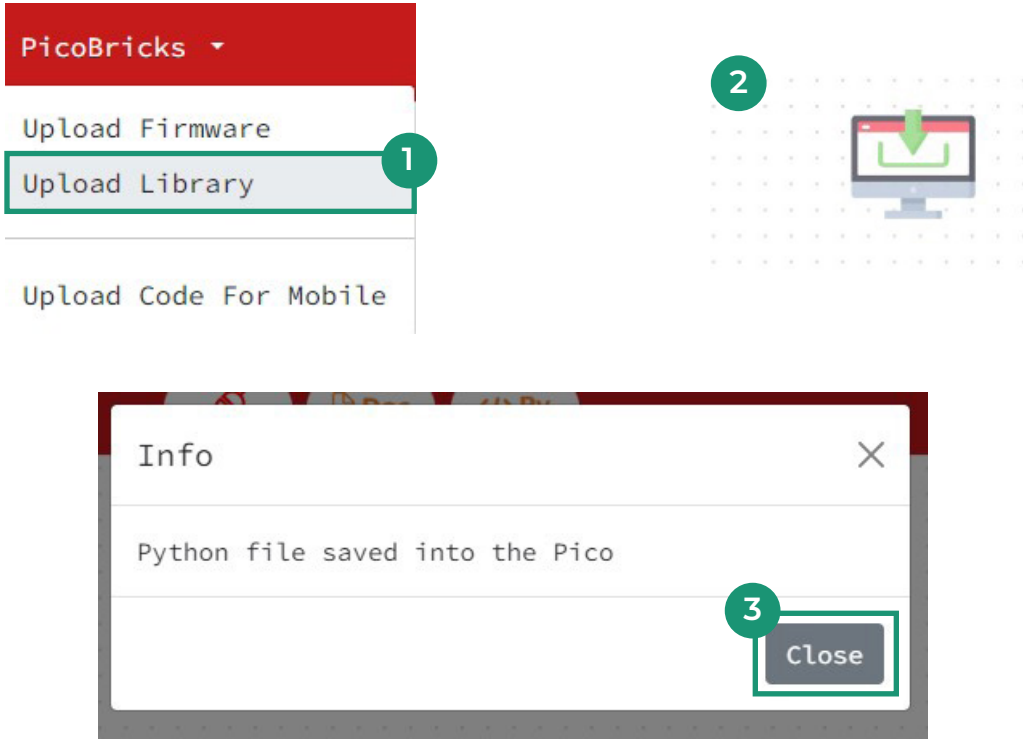


Firmware dosyasını kaydettikten sonra PicoBricks'i bağlamak için  butonuna tıklayınız. Açılan pencereden doğru portu seçerek PicoBricks'inizin bağlantısını yapabilirsiniz. Bağlantı yaptıktan sonra  değişecektir.



Bağlantımızı yaptığımız göre Raspberry Pi Pico mikro denetleyici kartımıza PicoBricks kütüphanemizi yükleyelim. (Eğer daha önce kütüphane yüklediyseniz ve farklı bir IDE'de çalışmadıysanız bu işlemi bir kere yapmanız yeterlidir.)

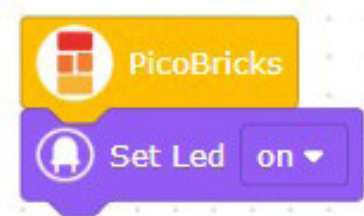
Aşağıdaki adımları izleyerek kütüphaneyi Raspberry Pi Pico içerisine yükleyiniz.



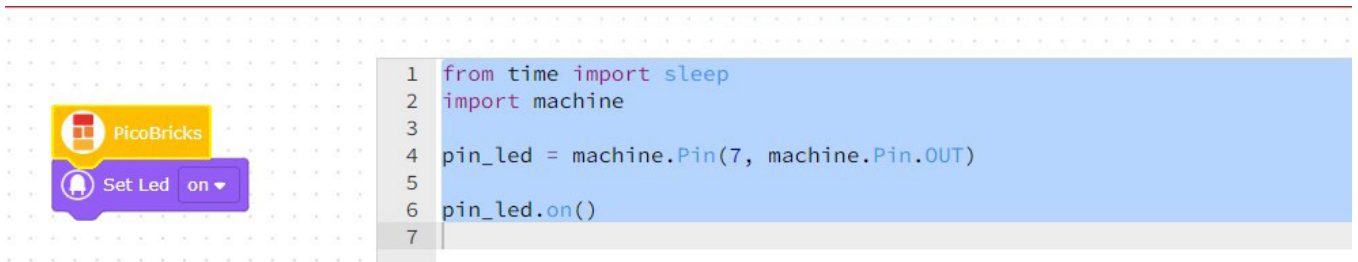
PicoBricks kütüphanesini de yüklediğimize göre PicoBricks LED modülünü çalıştırarak yaptığımız işlemleri test edelim.

Aşağıdaki işlemleri takip ederek kod bloklarını hazırlayalım.

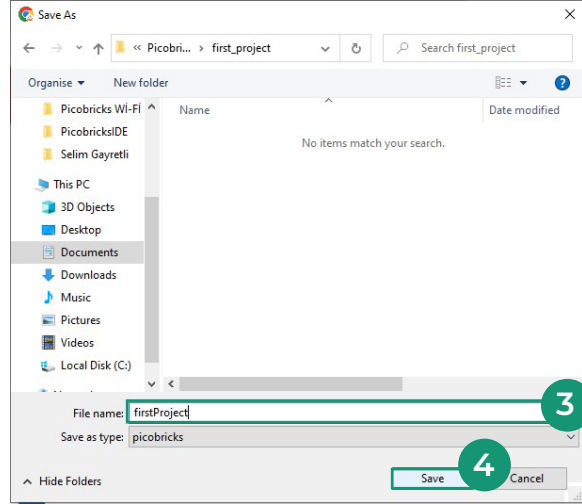
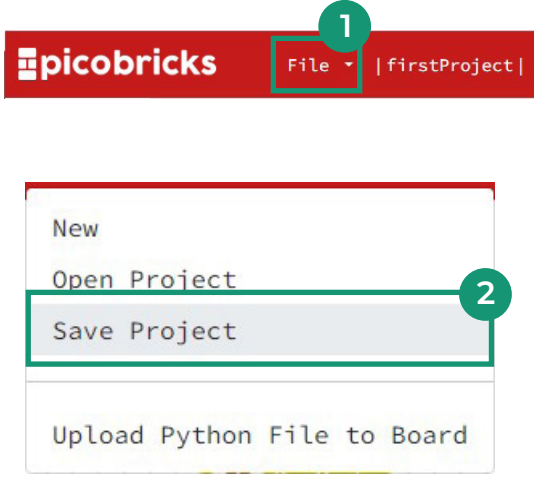
1. "Basic" bloklarından "PicoBricks" ve LED bloğunu alt alta gelecek şekilde sürükle-yelim.



2. Kod bloklarının python çıktısını  butonuna tıklayarak görebilirsiniz.



3. Kodumuzu hazırladıktan sonra “Start” butonuna basarak projemizi çalıştıralım.
4. Aşağıdaki adımları takip ederek projenizi kaydedebilirsiniz.



PROJELER

Blink

Gerçek hayatta işi yeni öğrenmeye başlayan çalışan, önce en temel görevi üstlenir. Temizlik görevlisi ise süpürgenin kullanımını, aşçı ise mutfak araç gereçlerini, garson ise tepsi taşımayı... Bu örnekleri arttırabiliriz. Yazılım geliştirmeye yeni başlayanların ilk yazdıkları kod "Hello World" olarak bilinir. Kullandıkları dil de ekrana ya da konsol penceresine program başlar başlamaz "Hello World" yazdırmak, programlamaya atılan ilk adımdır. Bir bebeğin emeklemeye başlaması gibi... Robotik kodlamaya diğer adıyla fiziksel programlamaya atılan ilk adım ise Blink uygulamasıdır. Robotik kodlamaya göz kırpmak anlamını taşır. Basit olarak bir LED'in bağlantısını devre kartına yaparak yapılan kodlama ile LED'in sürekli yanıp sönmelerini sağlar. Robotik kodlama alanında kendini geliştirmiş kişilere bu seviyeye nasıl geldiklerini sorun. Size verecekleri cevap şöyle başlar; her şey bir LED yakmak ile başladı!

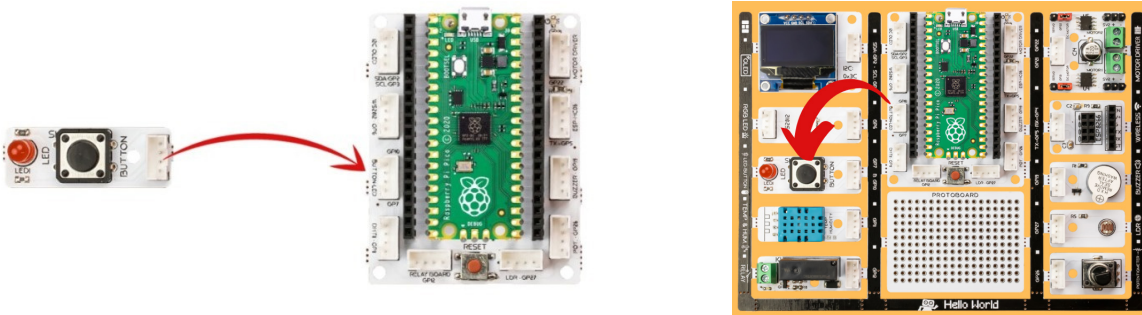
LED'ler elektronik cihazların dilidir. LED'ler sayesinde programcı cihazın görevin hangi aşamasında olduğunu, varsa sorunun ne olduğunu, hangi seçeneklerin aktif olduğunu kullanıcılara ifade eder. Bu projede Picobricks ile üzerinde yer alan LED'lerin çeşitlerini öğrenip onları nasıl yakıp söndüreceğini öğreneceksin.

Proje Detayları ve Algoritma

Picobricks üzerinde 1 adet 5mm kırmızı LED, 1 adet WS2812B RGB LED bulunmaktadır. Normal LED'ler tek renk yanabilirken RGB renkler hem ana hem ara renkler olmak üzere farklı renklerde yanabilmektedir. Bu projede Picobricks üzerindeki kırmızı LED kullanacağız.

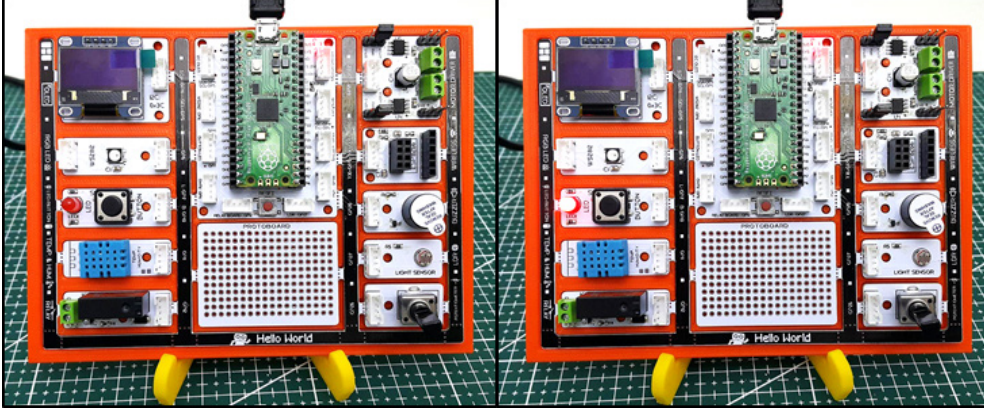
Projede Picobricks üzerindeki kırmızı LED yakılması, belirli bir süre geçtikten sonra söndürülmesi, tekrar belirli bir süre geçtikten sonra yakılması ve bu işlemlerin sürekli tekrarlanması için gerekli kodlar yazacağız.

Bağlantı Şeması



Picobricks'in modüllerini kablo bağlantısı yapmadan kodlayabilir ve çalıştırabilirsin. Modülleri board'dan ayırarak kullanacaksan grove kablolar ile modül bağlantılarını yapmalısın.

Proje Görseli



Proje Önerisi

LED'i farklı zaman aralıklarında yakabilir miyiz? Örneğin; LED'in yanıp sönmesi saniyede birkaç kez, her yarım saniyede birkaç kez.

Projeyi PicoBricks IDE ile Kodlama

PicoBricks IDE - Picobricks bağlantısı ve kütüphane yükleme işlemlerini yaptıysan ilk proje için takip etmen gereken adımlar aşağıdaki tabloda ayrıntılı olarak belirtilmiştir..

1	Öncelikle, "Start" butonuna tıkladığınızda projenin çalışması için "Basic" bloklarından "PicoBricks" bloğunu sürükleyin.	
2	Kırmızı LED'in devamlı yanıp sönmesi için "PicoBricks" bloğunun altına "Forever" bloğunu sürükleyin.	
3	Kırmızı LED'i yakmak için "Set LED on" bloğunu "Forever" bloğunun içine sürükleyin.	



4	<p>LED'i yakan blok olan "Set LED on" seçeneğinin altındaki "wait" bloğunu sürükleyin,böylece kırmızı LED belirli aralıklarla yanıp sönecektir. Bekleme için "wait" bloğunu 0,5'e ayarlayın.</p>	
5	<p>"Set LED" bloğunun açık veya kapalı olmasını belirlemek için, "ok" butonuna tıklayabilirsiniz.</p>	
6	<p>Kırmızı LED'i kapatan ve yarım saniye beklemeyi sağlayan kod blokları</p> <p>Projenin kod blokları hazır!</p>	

Action - Reaction

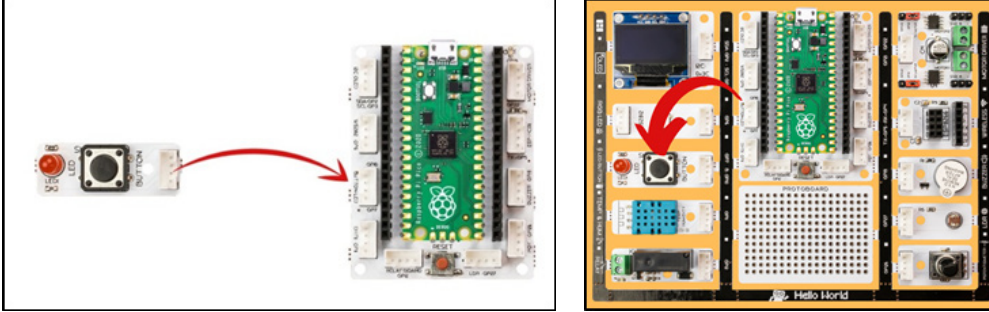
Newton hareket yasalarında açıkladığı gibi her etkiye karşı bir tepki oluşur. Elektronik sistemler kullanıcılardan komutlar alırlar ve görevlerini yerine getirirler. Genellikle bu iş için bir tuş takımı, dokunmatik ekran veya bir buton kullanılır. Elektronik cihazlar görevlerinin sona erdiğini ve görev 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 ve Algoritma

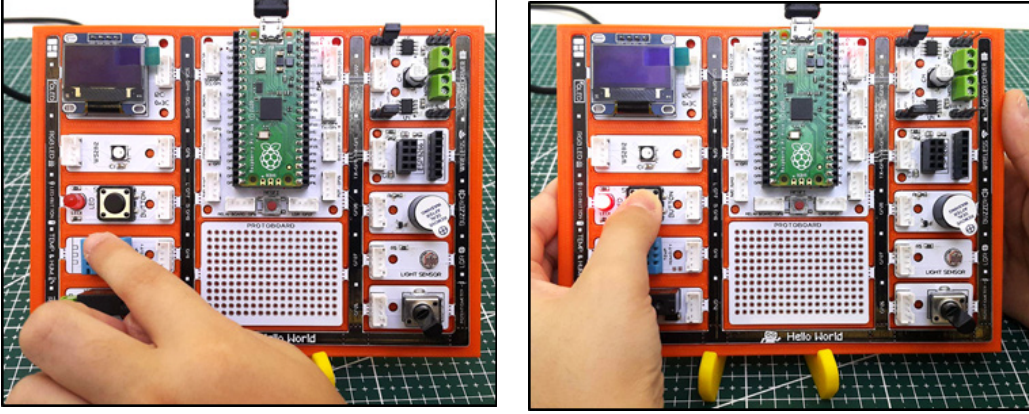
Elektronik sistemlerde farklı türlerde butonlar kullanılmaktadır. Kilitli butonlar, push butonlar, anahtarlmalı butonlar... Picobricks üzerinde 1 adet push buton bulunmaktadır. Anahtar gibi çalışmaktadırlar, basıldığında akımı iletirler bırakıldığında akımı iletmezler. Projede butonun akım iletip iletmediğini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

Bağlantı Şeması

Picobricks'in modüllerini kablo bağlantısı yapmadan kodlayabilir ve çalıştırabilirsin. Modülleri board'dan ayırarak kullanacaksan grove kablolar ile modül bağlantılarını yapmalısın.



2.2.3. Proje Görseli






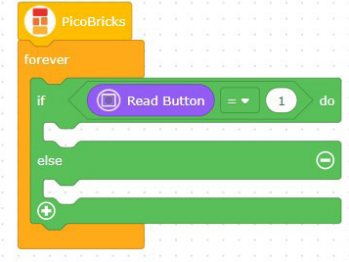
2.2.4. Proje Önerisi

Bu projede Picobricks'in buton-LED modülünü kodlayarak projelerinde kullanıcından bir komutun nasıl alınacağını ve bu komuta nasıl tepki verileceğini öğreneceksin.

2.2.5. Projenin PicoBricks IDE ile Kodlanması

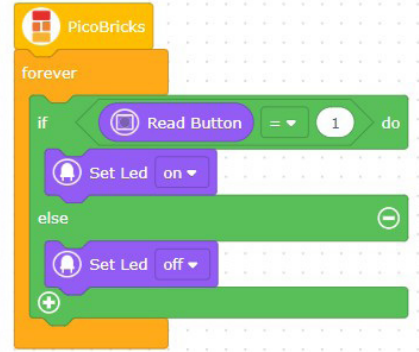
1	<p>Öncelikle, "Start" butonuna tıkladığınızda projenin çalışması için "Basic" bloklarından "PicoBricks" bloğunu sürükleyin.</p>	
2	<p>Projenin devamlı çalışması için "Forever" bloğunu sürükleyin.</p>	
3	<p>Proje için gerekli durum yapısını oluşturmak için, "Forever" bloğunun içine "Logic" bloklarından "if-else" yapısını sürükleyin.</p>	

4 Logic bloklarından  sürükleyerek ve “Bricks” bloklarından  sürükleyerek durum yapısını oluşturun. Daha sonra  yapısını “if” bloğunun içine sürükleyin.



5 Kod çalıştığında ve istenen durum sağlandığında LED'in yanması için “if” yapısının içine “Set LED on” bloğunu ve “else” yapısının içine “Set LED Off” bloğunu sürükleyin.

Projenin kodları hazır, kodunu kaydedebilirsin.



2.3. Autonomous Lighting

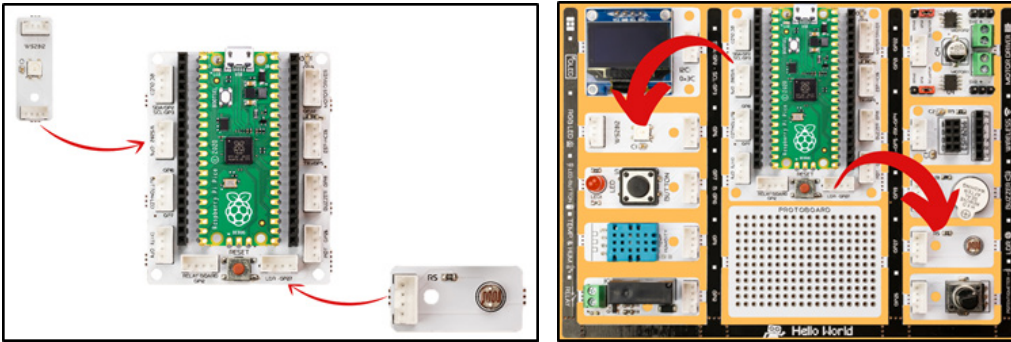
Elektronik sistemlerin verilen görevi, topladığı veriler doğrultusunda karar vererek otomatik olarak yapmasına otonom olma durumu denir. Elektronik sistemlerin çevresinden veri toplamasını sağlayan bileşenlerine sensör (algılayıcı) denir. Ortamdaki ışık seviyesinin hangi düzeyde olduğu, hava sıcaklığının kaç derece olduğu, su akış hızının kaç lt/dk olduğu, ses şiddetinin ne kadar olduğu gibi birçok veri sensörler tarafından toplanarak elektrik sinyalleri olarak yani veri olarak PicoBricks'e iletilir. Picobricks'te birçok sensör yer almaktadır. Sensörlerden verinin nasıl alındığını ve bu verilerin nasıl yorumlanıp kullanılacağını bilmek, kitap okumanın kelime dağarcığını geliştirmesi gibi proje fikirlerini geliştirecektir.

2.3.1. Proje Detayları ve Algoritma

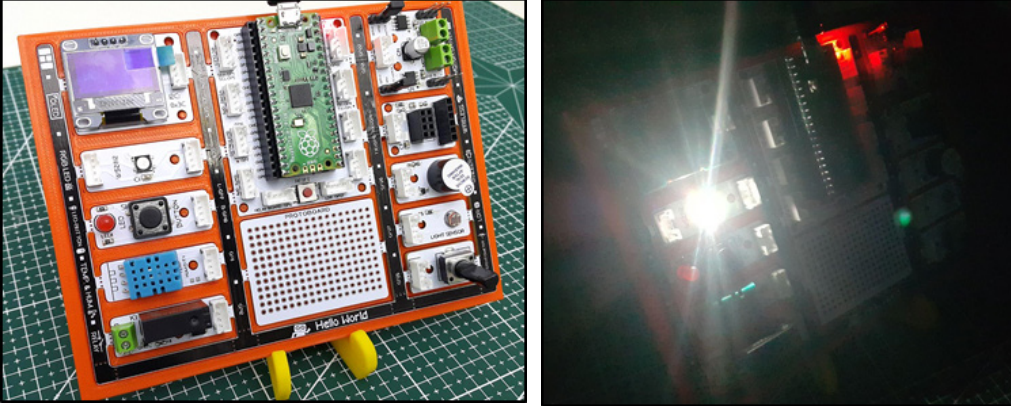
Sensörler dış ortamlardaki verileri algılayarak mikrokontrolcülere veriler gönderen elektronik bileşenlerdir. LDR sensör de ortamdaki ışık miktarını algılayarak analog değerler gönderir. Projemizde önce LDR sensör değerlerini okuyarak ortam aydınlık ve karanlık olduğunda gelen verileri kontrol edecek, sonra bu verilere göre bir sınır belirleyerek ışık miktarı bu sınırın altındaysa Picobricks'in RGB LED'ini yakacak değilse LED'i söndüreceğiz.

2.3.2. Bağlantı Şeması

Picobricks'in modüllerini kablo bağlantısı yapmadan kodlayabilir ve çalıştırabilirsin. Modülleri board'dan ayırarak kullanacaksan grove kablolar ile modül bağlantılarını yapmalısın.



2.3.3. Proje Görseli

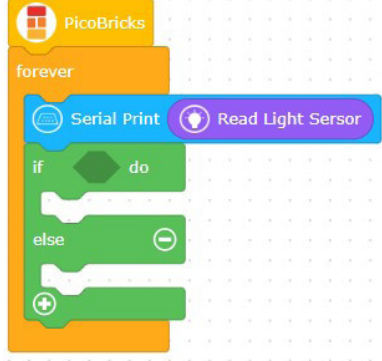



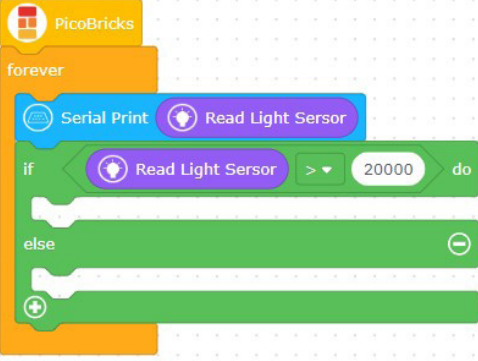
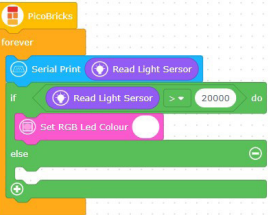


2.3.4. Proje Önerisi

Bu projede PicoBricks ile hava karardığında aydınlatmanın otomatik yandığı sistemlerin çalışma sistemlerini anlamak için ışık miktarı düştüğünde LED'in yanmasını sağlayacağız.

2.3.5. Projenin PicoBricks IDE ile Kodlanması

1	Öncelikle, "Start" butonuna tıkladığınızda projenin çalışması için "Basic" bloklarından "PicoBricks" bloğunu sürükleyin.	
2	Durdurana kadar projenin çalışması için "Forever" bloğunu sürükleyin. Diğer kod blokları bu bloğun içine sürüklenecek.	
3	LDR sensörünün değerini Seri Monitöre yazdırarak, karanlığı kolayca tespit edebilirsiniz. Bunun için ""Basic" menüsünden "Serial Print" bloğunu ve "Bricks" menüsünden "Read Light Sensor" bloğunu oluşturun ve "Forever" bloğunun için sürükleyin.	

4	<p>Projede gerekli durumu belirlemek için, “if-else” bloğunu “Forever” bloğunun içine sürükleyin.</p>	
5	<p>Durum yapısını oluşturduktan sonra “if” bloğunun içine sürükleyelim. Durum yapısını oluşturmak için, 3. adımda tespit edilen ışık sensörünün değerini kullanabilirsiniz. “Logic” bloklarından  bloğunu alarak, seçeneklerden “>” operatörünü seçiniz.</p> <p>“Bricks” bloklarından  bloğunu eşitliğin sol tarafına sürükleyin. LDR sensörünün değerini eşitliğin sağ tarafına yazalım.</p> <p>Karanlık tespit edildiğinde durum yapısının içindeki  blokları çalışacak.</p>	
6	<p>Karanlık tespit edildiğinde LDR sensörünün çevreyi aydınlatması için, RGB LED Color bloğunun rengini beyaz olarak ayarlayın.</p>	

7

Işık seviyesi 20000'den az olursa RGB LED'in kapanması için, "Clear RGB LED Color" bloğunu sürükleyin.



2.4. Thermometer

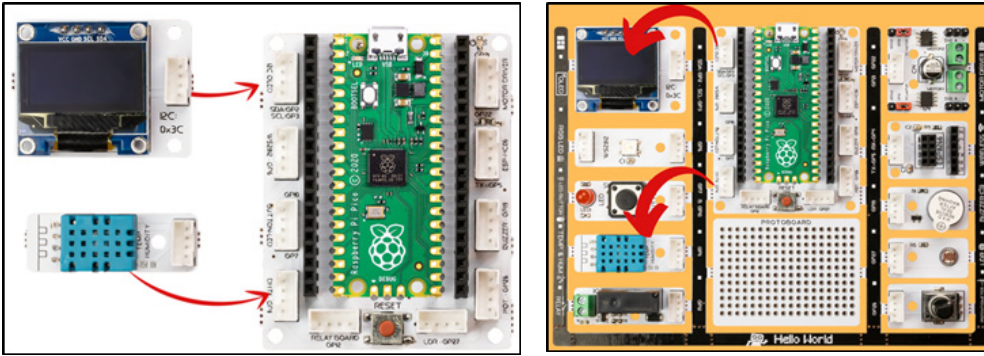
Sensörler elektronik sistemlerin duyu organlarıdır. Hissetmek için derimizi, görmek için gözümüzü, duymak için kulağımızı, tatmak için dilimizi ve koklamak için burnumuzu kullanırız. Picobricks'te hali hazırda bir çok duyu organı(sensör) vardır. Ayrıca yenileri de eklenebilir. Nem, sıcaklık, ışık ve daha birçok sensörü kullanarak çevreyle etkileşim sağlayabilirsiniz. Picobricks ortam sıcaklığını başka hiçbir çevre bileşenine ihtiyaç duymadan ölçebilir.

Ortam sıcaklığı seralarda, kuluçka makinelerinde, ilaçların taşınmasında kullanılan ortamlarda kısaca sıcaklık değişiminin sürekli takip edilmesi gereken durumlarda kullanılmaktadır. Projelerinde sıcaklık değişimi üzerine bir işlem yapacaksan ortam sıcaklığını nasıl ölçeceğini bilmelisin. Bu projede Picobricks ile ortam sıcaklığını OLED ekranda gösterecek bir termometre hazırlayacaksın.

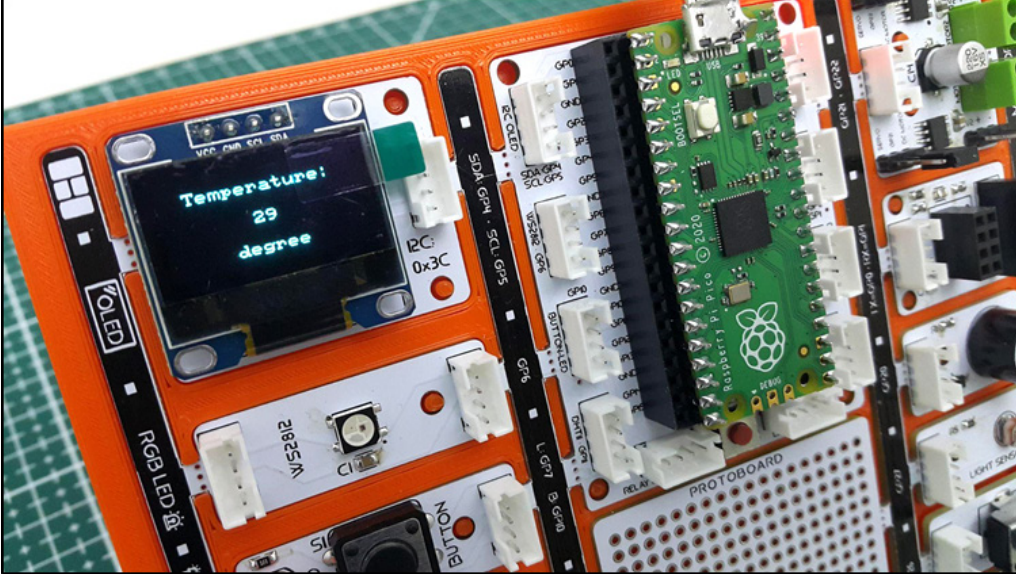
2.4.1. Proje Detayları ve Algoritma

Picobricks'te DHT11 modülü bulunmaktadır. Bu modül ortamdaki sıcaklık ve nem miktarını algılayarak mikrokontrolcüye veriler gönderebilmektedir. Bu bu projede Picobricks üzerindeki DHT11 sıcaklık ve nem sensörünün ölçtüğü sıcaklık değerlerini Picobricks OLED ekrana yazdırmak için gerekli kodları yazacağız.

2.4.2. Bağlantı Şeması



2.4.3. Projenin Kurulum Adımları

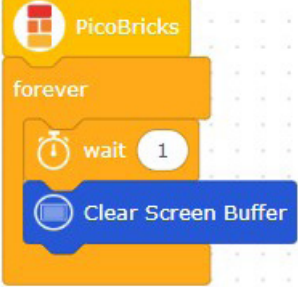
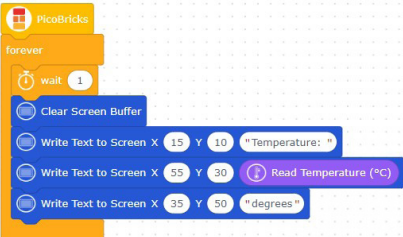


2.4.4. Proje Önerisi

Projenizi geliřtirmek için ortam sıcaklığı 30 derecenin üzerine çıktığında kırmızı LED'in yanmasını ve ekranda uyarı ibaresinin görünmesini sağlayabilirsiniz.

2.4.5. Projenin PicoBricks IDE ile Kodlanması

1	<p>Öncelikle, "Start" butonuna tıkladığınızda projenin çalışması için "Basic" bloklarından "PicoBricks" bloğunu sürükleyin.</p>	
2	<p>Projenin durdurana kadar çalışması için "Forever" bloğunu sürükleyin. Diğer kod blokları bu bloğun içine sürüklenecek.</p>	

3	Döngünün 1'er saniye aralıklarla tekrarlaması için, "Wait" bloğunu ilk sıradaki "forever" bloğunun içine sürükleyin.	
4	Döngü her tekrarladığında ekranın temizlenmesi için, "Display" bloklarından "Clear Screen Buffer" bloğunu "forever" bloğunun içine sürükleyin.	
5	OLED ekranın X=15 ve Y=10 koordinatlarına "Sıcaklık" yazısını yazmak için "Display" bloklarından "Write Text to Screen" bloğunu sürükleyin ve yazma alanına "Temperature:" yazın.	
6	DHT11 sensörünün aldığı değeri OLED ekranının X=55 ve Y=30 koordinatlarına yazdırmak için, "Bricks" bloklarından "Read Temperature" bloğunu "Write Text to Screen" bloğunun içine sürükleyin.	
7	X=35 ve Y=50 koordinatlarına "degrees" yazmak için, "Display" bloklarından "Write Text to Screen" bloğunu sürükleyerek yazma alanına "degrees" yazın.	
8	OLED ekrana yazmak istenen yazıları görmek için, OLED ekranı başlatmanız gerekir. Bunun için "Display" bloklarından "Show Screen Buffer" bloğunu sürükleyin.	

2.5. Graphic Monitor

Çevremizdeki elektronik eşyalara baktığımızda değiştirilebilir birçok özelliklerinin olduğunu mühendisler tarafından kullanıcının en çok işine yarayacak şekilde tasarlandıklarını fark edersin. Aydınlatma sistemleri, pişirme sistemleri, ses sistemleri, temizlik sistemleri gibi. bir çok sistem kullanıcısı tarafından çalışma şekli , miktarı, yöntemi vb. özellikleri değiştirilebilir şekilde programlanır.

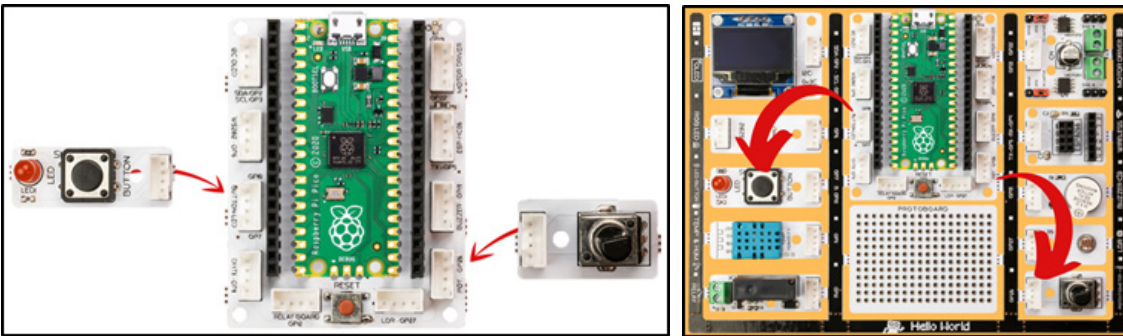
Robotik projelerde ses seviyesini değiştirme, motor hızını değiştirme , ışığın parlaklığını değiştirme işlemlerinde elektrik geriliminin daha düşük veya yüksek etki yaratacak şekilde gönderilmesi sağlanır. Bileşene giden elektrik sinyalinin sıklığı azaltılarak daha düşük seviyede çalışması giden elektrik sinyallerinin sıklığı artırılarak yüksek seviyede çalışması sağlanabilir.

Ekranı olmayan sistemlerde bazı sensörleri ve sistemin çalışmasında görev alan değişkenleri takip etmek için gerçek zamanlı grafik monitörler kullanılır. Arızanın tespit edilmesi için grafik monitörler oldukça kolaylık sağlamaktadır..

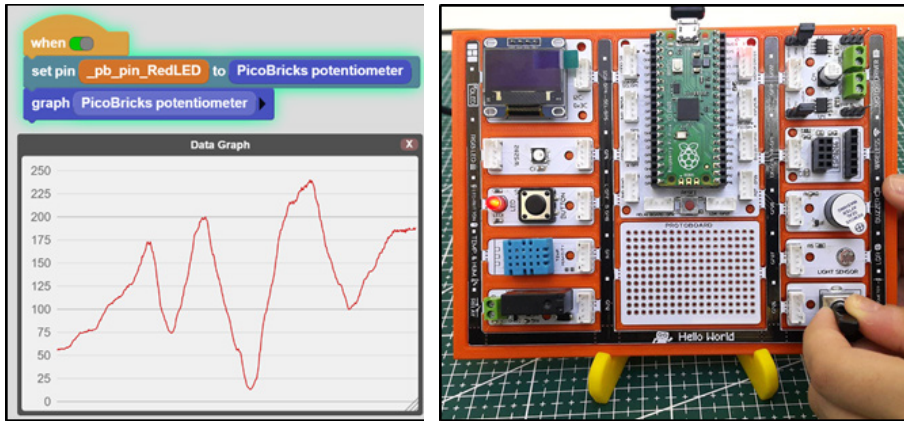
2.5.1. Proje Detayları ve Algoritma

Bu projede potansiyometre ile kırmızı LED'in parlaklığı artırıp azaltığımız bir proje hazırlayacağız. Ayrıca bu işlem sırasında meydana gelen elektriksel değişimi Microblocks grafik monitöründen eş zamanlı olarak takip edeceğiz. Picobricks başladığında potansiyometre değeri sürekli okunarak LED'in parlaklık değerini ayarlanacaktır. Elektrik sinyalinin frekansının değiştirilerek etkisinin azaltıldığı uygulamalara PWM denmektedir. Potansiyometreden okuduğumuz analog değerleri PWM sinyalleri olarak kırmızı LED'e göndereceğiz ve aydınlatma şiddetini ayarlayabileceğiz.

2.5.2. Bağlantı Şeması




2.5.3. Proje Görseli



2.5.4. Proje Önerisi

Potansiyometreyi çevirdikçe, buzzer'dan gelen ses seviyesi değişen ve grafik monitörde veri akışını gösteren bir proje geliştirebilirsin.

2.5.5. Projeyi PicoBricks IDE ile Kodlama

1	İlk olarak, "Start" butonuna tıkladığında veya PicoBricks'i çalıştırdığında projenin çalışması için "Basic" bloklarından "PicoBricks" bloğunu sürükleyin.	
2	Durdurana kadar projenin çalışması için "Forever" bloğunu sürükleyin. Diğer kod bloklarını bu bloğun içine sürükleyin.	

3	<p>Aşağıdaki adımları takip ederek, “color” adında bir değişken oluşturalım. “color” değişkenini oluşturduğunda, “Variables” bloklarında üç blok oluşturulur.</p> 	
4	<p>Potansiyometrenin değerini 1-255 aralığında ayarlamak için, “Math” bloklarından “operation” bloğu ile matematiksel ifade oluşturalım ve “color” değişkenine atayalım.</p> 	
5	<p>Potansiyometre ile RGB LED'in parlaklığını değiştirmek için, ilk olarak RGB LED'in rengini manuel olarak değiştiren “Set RGB LED Color” bloğunu RGB LED bloklarına sürükleyelim.</p>	
6	<p>RGB LED'in parlaklığını potansiyometre değerine göre değiştirmek için, oluşturduğun “color” değişkenini “Red”, “Green” ve “Blue” değerlerinin içine sürüküle.</p>	



2.6. 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.

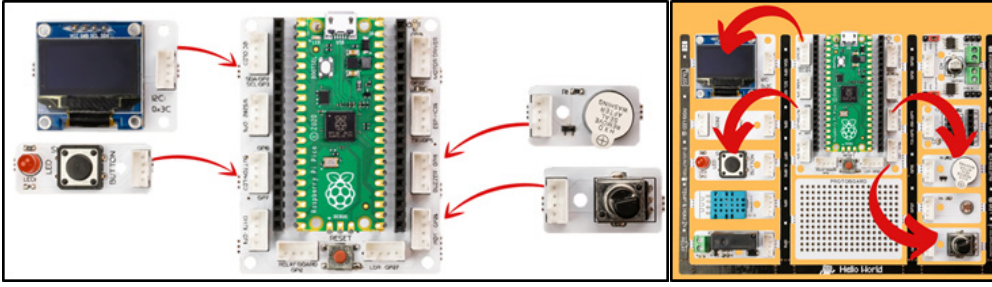
2.6.1. Proje Detayları ve Algoritma

Notalarını bildiğimiz her şarkıyı Picobricks ile çalabilirsin. Şarkıyı başlatmak için buton-LED modülünü, şarkının hızını ayarlayabilmek için potansiyometre modülünü notaları çalabilmek için buzzer modülünü kullanacağız.

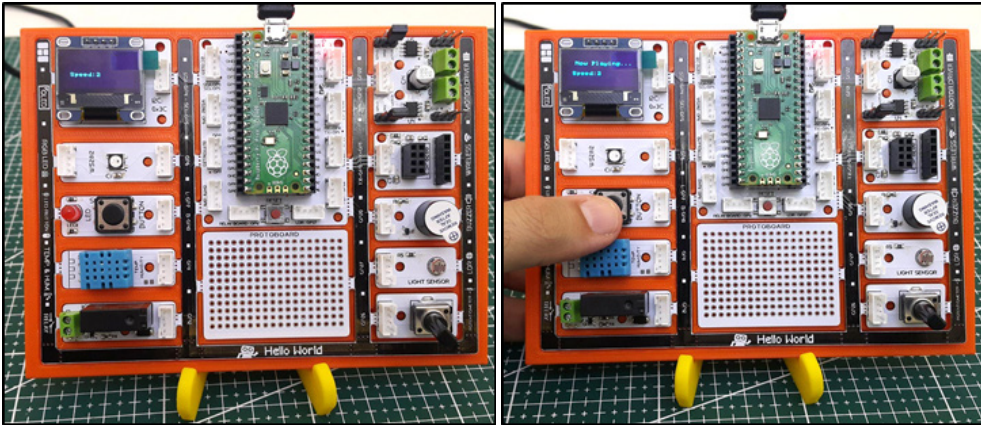
Potansiyometre analog giriş modülüdür. Değişken dirençtir. Üzerinden geçen akım miktarı çevrildikçe bir musluğun açılıp kapatılması gibi artar ve azalır. Bu akım miktarını kodlarla kontrol ederek şarkının hızını ayarlayacağız. Buzzer'lar üzerlerinden geçen akımın şiddetine göre ses seviyelerini, gerilim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

Projede butona basılma durumunu kontrol edeceğiz. Butona basıldığında melodinin çalmaya başlamasını sağlayacağız. Melodinin çalması sırasında notaların çalınma sürelerini aynı oranda artırıp azaltabilmek için rithm adında bir değişken kullanacağız. Picobricks başladıktan sonra kullanıcının potansiyometre ile ister melodi çalarken ister çalmadan önce rithm değişkenini ayarlayabilmesini sağlayacağız. Picobricks açık olduğu sürece potansiyometre değerini (0-1023) 128'e bölüp rithm değişkene atayacağız. Değişkenler, kullanıcı tarafından ya da sensörler tarafından değiştirilebilecek değerleri kodlarımızda kullanmak istediğimizde kullandığımız veri yapılarıdır. Kullanıcı şarkıyı başlatmak için butona bastığında ise notaların süreleri rithm değişkenine göre hesaplanan süre boyunca çalmasını sağlayacak nota kodlarını hazırlayacağız.

2.6.2. Bağlantı Şeması



2.6.3. Proje Görseli



2.6.4. Proje Önerisi

Çalan notaya göre farklı renk yakarak ve notanın adını OLED ekrana yazdırarak, projeni görsel açıdan geliştirebilirsin.

2.6.5. Projenin PicoBricks IDE ile Kodlanması

İlk olarak, kodumuz için gerekli fonksiyonları oluşturarak başlayalım.

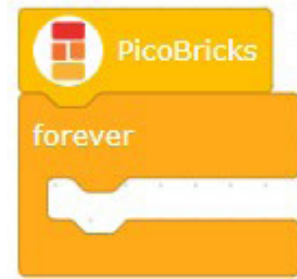
- “music” adında bir fonksiyon tanımlayalım.



- Belirli bir aralıkta çalan melodiyi, müzik fonksiyonundaki buzzer bloklarını kullanarak oluşturalım. Notaları oluşturmak için, “Play Buzzer Freq” bloğunun frekans değerini değiştirebilirsin. Dominate the rhythm projesinde kullanılan notaların frekansları aşağıdaki müzik fonksiyonunda gösteriliyor.

Şimdi, diğer kod bloklarını oluşturarak projeye devam edelim. Bu adımda yapılacaklar:

- “Start” butonuna tıkladıktan sonra, sürekli çalışan işlemleri oluşturmak için PicoBricks ve forever bloklarını sürükleyin.

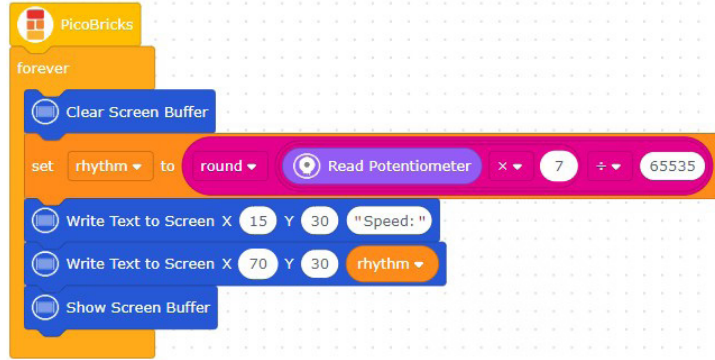


- “rhythm” isimli bir değişken oluşturalım ve “Read Potentiometer” bloğundaki 0-65535 değerini 0-7 aralığı ile değiştirelim.

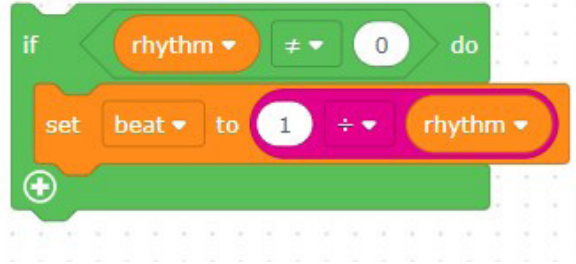
- Potansiyometrenin 0-7 aralığındaki değerini değiştirmek için aşağıdaki işlemi kullanabilirsin.



- OLED ekrandaki X=15, Y=30 ve X=70, Y=30 koordinatlarına “Speed:” yazısını ve “Rhythm” değişkenini yazdıralım.



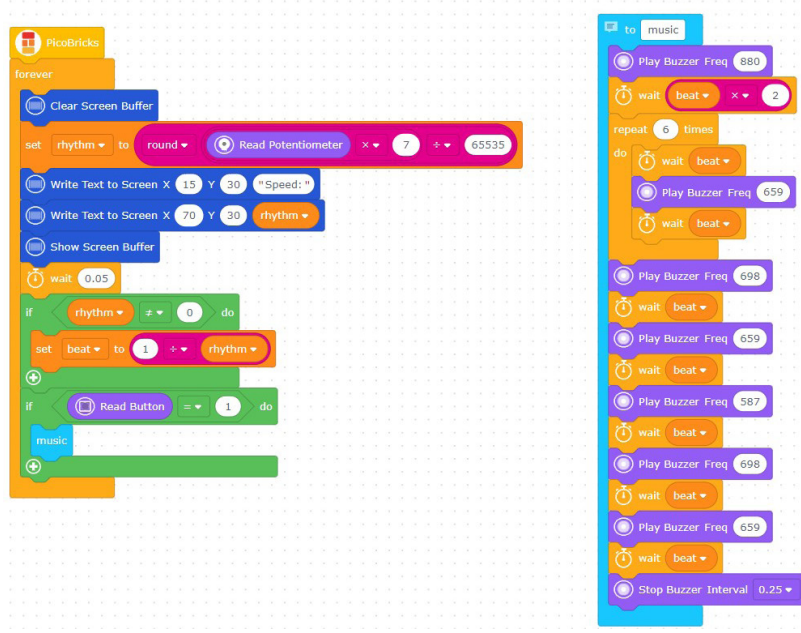
- Şimdi, durum yapısını oluşturalım. “rhythm” değişkeni 0’a eşit değilse, “beat” adında farklı bir değişken oluşturalım ve 1’i rhythm değerine bölelim. “music” fonksiyonundaki notaların çalma süresini belirlemek için “beat” değişkenini kullanabilirsiniz.



- Butona basıldığında, “music” fonksiyonunu “if” bloğu ile çağırmak için durum yapısını oluşturalım.



■ Kod blokları hazır!



2.7 Show Your Reaction

Şimdi de dikkat ve refleks geliştiren bir oyun hazırlayacağız. Hızlı hareket etmek ve dikkatin uzun süre sağlanabilmesi çocukların önemli gelişimsel özelliklerindedir. Okul öncesi ve ilköğretim dönemindeki çocukların dikkat sürelerini ve reflekslerini artırıcı etkinlikler yapması çocukların hoşlarına gittiği gibi ebeveynlerinin ve öğretmenlerinin de istediği bir durumdur. Hazırlayacağımız elektronik sistemi dikkat artırıcı ve refleks geliştirici bir oyun olacak. Projeyi bitirdikten sonra siz de arkadaşlarınızla yarışabilirsiniz. :)

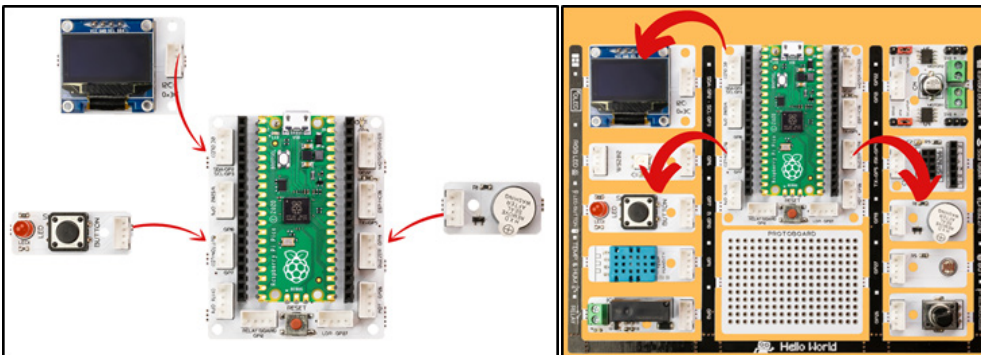
Bu projede her programlama dilinde kullanılan rastgelelik durumunu öğreneceksin. Picobricks ile OLED ekran, Buton-LED ve Buzzer modülünü kullanarak elektronik bir sistem geliştireceğiz.

2.7.1. Proje Detayları ve Algoritma

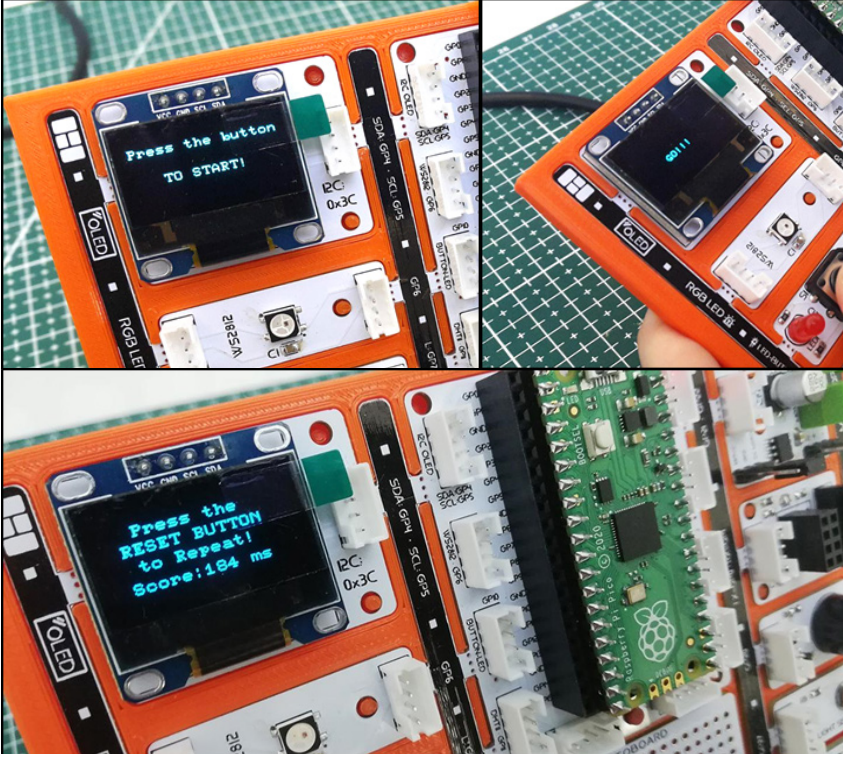
Picobricks açılır açılmaz bir zamanlayıcı çalışmaya başlar. Bu zamanlayıcı ile saniyenin binde 1'ini ölçebiliriz. Saniyenin binde birine milisaniye denir. Zamanlayıcılar günlük yaşamda birçok elektronik sistemde kullanılırlar. Zaman ayarlı aydınlatmalar, fırınlar, ütüler, mutfak robotları...

Projemiz çalışmaya başladığında OLED ekranda karşılama mesajı görüntüleyeceğiz. Ardından kullanıcıya oyuna başlaması için yapması gerekeni ekrana yazdıracağız. Oyuna başlayabilmek için butona basılacak butona basıldıktan sonra 3'ten geriye doğru ekranda sayılarak oyuncunun hazırlanmasını isteyeceğiz. Geri sayımın bitiminden sonra 2-10 saniye arasında rastgele bir süre içinde kırmızı LED yanacak. Kırmızı LED yandıktan sonra hemen zamanlayıcıyı sıfırlayacağız. Tekrar butona basılır basılmaz zamanlayıcıyı ölçeceğiz. Elde ettiğimiz bu değer milisaniye cinsinden olacak. Oyuncunun tepki süresi olarak bu değeri ekranda göstereceğiz.

2.7.2. Bağlantı Şeması



2.7.3. Proje Görseli



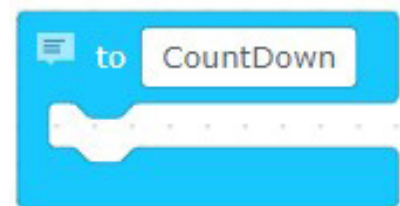
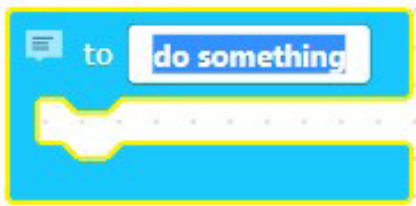
2.7.4. Proje Önerisi

Oyunu yeniden başlanması için, PicoBricks'in sıfırlanması gerekir. Oyuna yeniden başlanması için butona basılmasını isteyerek projeyi geliştirebilirsin. OLED ekrana en yüksek skoru ve son skoru yazdırabilirsin.

2.7.5. Projenin PicoBricks IDE ile Kodlanması

Öncelikle kodumuz için gereken fonksiyonları oluşturarak projemize başlayalım.

OLED ekranda 3'ten geriye saymayı sağlayan "CountDown" fonksiyonunu oluşturalım.



- Fonksiyon bloklarından “CountDown” bloğunun içine yazarak fonksiyonu tanımlayalım.

```

to Countdown
  set counter to 3
  while counter > 0
    Clear Screen Buffer
    Write Text to Screen X 60 Y 30 counter
    Show Screen Buffer
    wait 1
    change counter by -1
  end while
end Countdown

```

Fonksiyonu çağırdığımız zaman,

OLED ekrandaki X=60, Y=30 koordinatlarına 3-2-1 yazdırabilmesi için fonksiyon bloğuna sürükleyerek belirleyelim.

- İlk fonksiyonumuz hazır!
- İkinci fonksiyon olan “TimerFunc” fonksiyonunu oluşturalım.

```

to TimerFunc
  set TimeF to 0
  while Read Button = 0
    change TimeF by 1
    wait 1
    Set Led off
    Serial Print TimeF
    Clear Screen Buffer
    Write Text to Screen X 0 Y 10 "Press the button"
    Write Text to Screen X 25 Y 25 "to repeat"
    Show Screen Buffer
    Write Text to Screen X 25 Y 40 "Score:"
    Write Text to Screen X 70 Y 40 TimeF
    Write Text to Screen X 80 Y 40 "s"
  end while
  wait 0.3
  Show Screen Buffer
end TimerFunc

```

- “PicoBricks” bloğunu sürükleyerek, projeye başlayalım ve diğer kod bloklarını oluşturarak devam edelim. Proje her çalıştığında LED’i kapatan ve ekrana “Press the button” ve “TO START!” yazılarını yazan aşağıdaki kod bloklarını oluşturalım.

- “PicoBricks” bloğunu sürükleyerek projeye başlayalım ve diğer kod bloklarını oluşturalım. Proje her başladığında, LED’i kapatan ve ekrana “Press the button” ile “TO START!” yazılarını yazan aşağıdaki kod bloklarını oluşturalım.

```

PicoBricks
Set Led off
Write Text to Screen X 0 Y 10 "Press the button"
Write Text to Screen X 25 Y 35 "TO START!"
Show Screen Buffer

```

- Butona basıldıktan sonra OLED ekranı temizleyen
- “CountDown” fonksiyonunu çağırarak gerisayımı başlatan
- Gerisayımdan sonra, ekrana “GO!” yazarak rastgele LED’i yakan
- “TimerFunc” fonksiyonunu çağırarak “Score” yazdıran kod bloklarını oluşturalım.

```

forever
  while Read Button == 1
    Clear Screen Buffer
    Countdown
    Clear Screen Buffer
    Write Text to Screen X 60 Y 30 "GO!!!"
    Show Screen Buffer
    wait random integer from 1 to 5
    Set Led on
    TimerFunc

```

- Kod bloklarımız hazır!

```

PicoBricks
Set Led off
Write Text to Screen X 0 Y 10 "Press the button"
Write Text to Screen X 25 Y 35 "TO START!"
Show Screen Buffer

forever
  while Read Button == 1
    Clear Screen Buffer
    Countdown
    Clear Screen Buffer
    Write Text to Screen X 60 Y 30 "GO!!!"
    Show Screen Buffer
    wait random integer from 1 to 5
    Set Led on
    TimerFunc

to TimerFunc
  set TimeF to 0
  while Read Button == 0
    change TimeF by 1
    wait 1
  Set Led off
  Serial Print TimeF
  Clear Screen Buffer
  Write Text to Screen X 0 Y 10 "Press the button"
  Write Text to Screen X 25 Y 25 "to repeat"
  Show Screen Buffer
  Write Text to Screen X 25 Y 40 "Score:"
  Write Text to Screen X 70 Y 40 TimeF
  Write Text to Screen X 80 Y 40 "s"
  wait 0.3
  Show Screen Buffer

to Countdown
  set counter to 3
  while counter > 0
    Clear Screen Buffer
    Write Text to Screen X 60 Y 30 counter
    Show Screen Buffer
    wait 1
    change counter by -1

```

2.8. My Timer

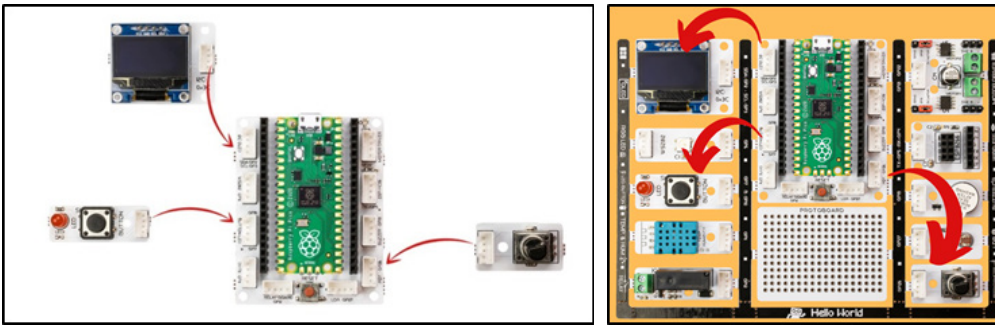
Zaman ölçmek günlük hayatımızda farkına varmadan yaptığımız basit ama önemli bir iştir. Ameliyattaki cerrah, toplantısına yetişmeye çalışan bir iş insanı, kazanmaya çalışan sporcu, sınavı bitirmeye çalışan bir öğrenci ya da satranç müsabakası... Zaman ölçmek için akıllı kol saatleri, telefonlar hatta profesyonel kronometreler kullanılmaktadır. Elektronik sistemler içinde zaman oldukça doğru kullanılması gereken bir değişkendir. Örneğin bir Çamaşır makinesi; tamburun ne kadar süre saat yönünde ne kadar saat yönü tersine döneceği, deterjanı eritip alabilmek için kaç sn su akması gerektiği hep zaman ölçerek yapılan görevlerdir. Zamanın önemli olduğu projeler geliştirmek için onu nasıl kullanacağını bilmelisin.

Bu projede Picobricks ile OLED ekran, buton ve potansiyometre modüllerini kullanarak kendi zaman ölçme aygıtını yapacaksın. Bir Timer...

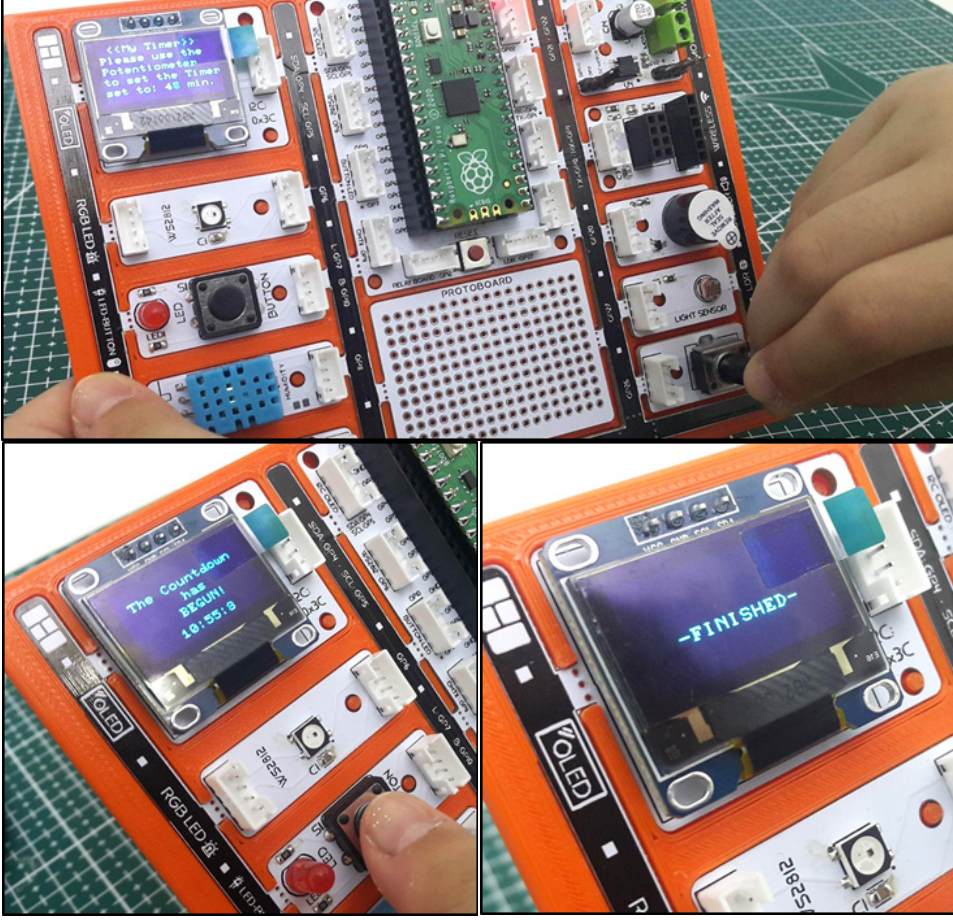
2.8.1. Proje Detayları ve Algoritma

Picobricks başladığında ekrana projeyi tanıtan ve yönerge içeren bir ifade yerleştirelim. Kullanıcı potansiyometreyi çevirdikçe 0-60 dakika aralığında bir süre belirleyecek. Kullanıcı potansiyometre ile süreye karar verdikten sonra Picobricks'in butonuna bastığında dakika saniye ve salise ekranda geri doğru saymaya başlayacak. Eğer zaman geriye doğru akarken butona basılırsa Timer duracak ve kalan süreyi ekranda gösterecek. Butona basılmadan dakika, saniye ve salise sıfır değerine ulaşırsa ekrana sürenin dolduğunu ifade eden bildirim gösterilecek ve program durdurulacak.

2.8.2. Bağlantı Şeması



2.8.3. Projenin Görself



2.8.4. Proje Önerisi

Zamanlayıcıyı başlatmak için buzzer ekleyebilirsin. Zaman sıfırlandığında, buzzer ile farklı ve yüksek tonda bir uyarı verebilirsin.

2.8.5. Projeyi PicoBricks IDE ile Kodlama

Öncelikle, kod için gereken fonksiyonları oluşturarak projeye başlayalım.

- Projeyi başlattığında ekrana yazılacak ifadeleri belirlemek için “starterFunction” isimli bir fonksiyon tanımla.

```

to starterFunction
  while Read Button = 0
    Write Text to Screen X 10 Y 0 "<< MY Timer >>"
    Write Text to Screen X 0 Y 15 "Press The Button"
    Write Text to Screen X 20 Y 25 "To START!"
    Write Text to Screen X 15 Y 45 "Use the POT"
    Write Text to Screen X 10 Y 55 "to set the Timer"
    Show Screen Buffer
  end

```

- Potansiyometre ile saati ayarlamak için “hour” adında bir fonksiyon belirleyelim.

```

to hour
  while Read Button = 0
    Clear Screen Buffer
    Write Text to Screen X 35 Y 30 hourv
    Show Screen Buffer
  end
  set hourv to round Read Potentiometer x 24 ÷ 65535
  set clue to 1

```

- Potansiyometre ile dakikayı ayarlamak için “minute” adında bir fonksiyon belirleyelim.

```

to minute
  while Read Button = 0
    Clear Screen Buffer
    Write Text to Screen X 55 Y 30 minutev
    Show Screen Buffer
  end
  set minutev to round Read Potentiometer x 60 ÷ 65535
  set clue to 2

```

- Potansiyometre ile saniyeyi ayarlamak için “second” adında bir fonksiyon belirleyelim.

```

to second
  while Read Button = 0
    Clear Screen Buffer
    Write Text to Screen X 75 Y 30 secondv
    Show Screen Buffer
  set secondv to round Read Potentiometer x 60 ÷ 65535
  set clue to 3

```

- Potansiyometre ile ayarlanan tüm saat, dakika ve saniye sıfır olana kadar gerisayımı sağlayan “timerFunction” adında bir fonksiyon tanımlayalım ve bunları OLED ekrana yazdıralım.

```

to timerFunction
  control
  while hourv >= 0 and minutev >= 0 and secondv >= 0
    repeat secondv Times
      do
        change secondv by -1
        wait 1
        Serial Print secondv
      if secondv = 0 do
        change minutev by -1
        Serial Print minutev
        clock
        set secondv to 60
      if minutev = 0 do
        clock
        set minutev to 60
        change hourv by -1
        Serial Print hourv

```

- OLED ekrana saat, dakika ve saniyeyi yazdırmak için “clock” adında bir fonksiyon oluşturalım.

```

to clock
  Clear Screen Buffer
  Write Text to Screen X 35 Y 30 hourv
  Write Text to Screen X 50 Y 30 ":"
  Write Text to Screen X 55 Y 30 minutev
  Write Text to Screen X 70 Y 30 ":"
  Write Text to Screen X 75 Y 30 secondv
  Show Screen Buffer

```

- “control” adında bir fonksiyon belirleyerek kontrol yapısını belirle.

```

to control
  if (hourv < 0) or (minutev < 0) or (secondv < 0) do
    set hourv to 0
    set minutev to 0
    set secondv to 0
    set clue to 4
  clock
  Set Led on
  Play Buzzer Freq 800
  Set RGB Led Colour

```

- “PicoBricks” bloğunu sürükleyerek kodu oluşturmaya devam edelim.

```

PicoBricks
starterFunction
  Clear Screen Buffer
  set hourv to 0
  set minutev to 0
  set secondv to 0
  set clue to 0
  forever
    if (clue = 0) do
      hour
    else if (clue = 1) do
      minute
    else if (clue = 2) do
      second
    else if (clue = 3) do
      clock
    timerFunction
  else
    control
  if (Read Button = 1) do
    Set Led off
    Stop Buzzer Interval 0.25
    Clear RGB Led Colour

```


2.9. Alarm Clock

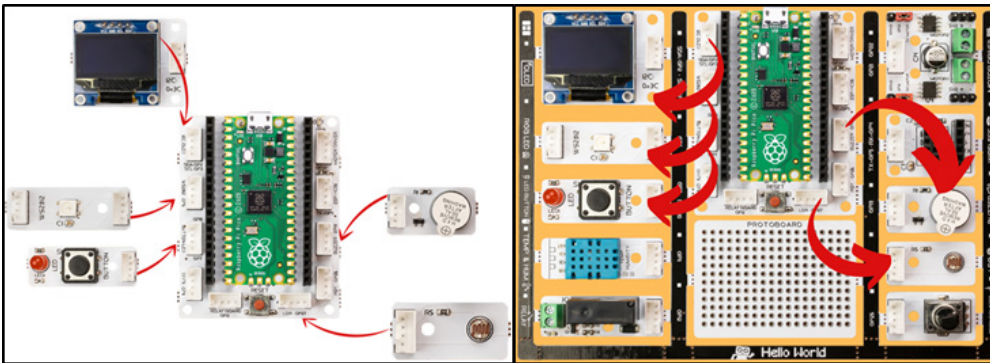
Küresel ısınma dünyamızın iklimini her geçen gün daha kötü etkiliyor. Ülkeler küresel ısınmanın etkilerini azaltmak için birçok tedbiri devreye sokuyor ve anlaşmalar imzalıyorlar. Yenilenebilir enerji kaynaklarının kullanılması ve enerjinin verimli kullanılması fabrikalardan evimizin odalarına kadar 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.

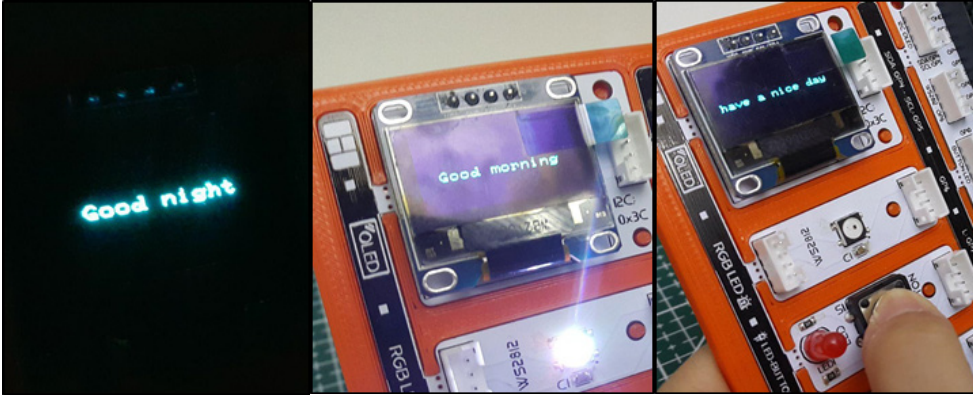
2.9.1. Proje Detayları ve Algoritma

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.

2.9.2. Bağlantı Şeması



2.9.3. Proje Görseli

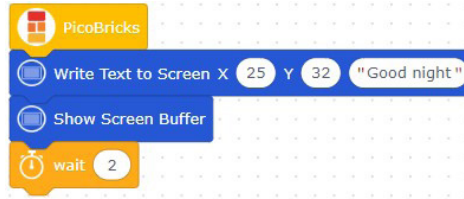


2.9.4. Proje Önerisi

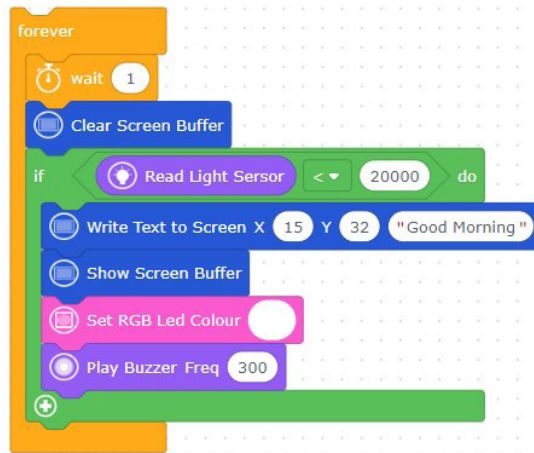
Alarm sesi olarak bir melodi ekleyerek projeni geliştirebilirsin. Gün ışığına göre alarm çaldırmak yerine belirlenen bir zaman için alarmı çaldırarak da projeni geliştirebilirsin.

2.9.5. Projenin PicoBricks IDE ile Kodlanması

PicoBricks başlatıldığında ekrana "Good night" yazan ve 2 saniye bekleyen kod bloklarını oluşturalım.

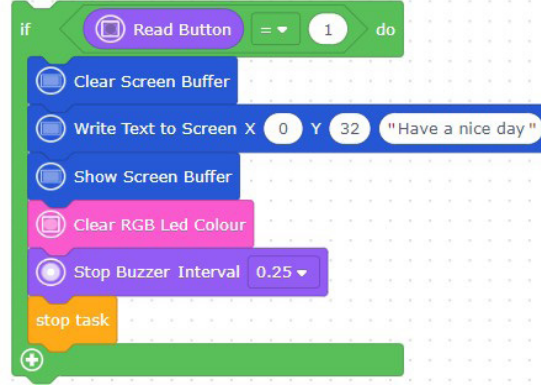


Proje için gerekli olan koşul ve döngü yapılarını oluşturalım. Eğer "Read Light Sensor" bloğunun değeri 2000'den az olursa, "Good Morning" yazısını yazan, buzzer'ı çalıştıran ve RGB LED'i yakan kod bloklarını oluşturalım.

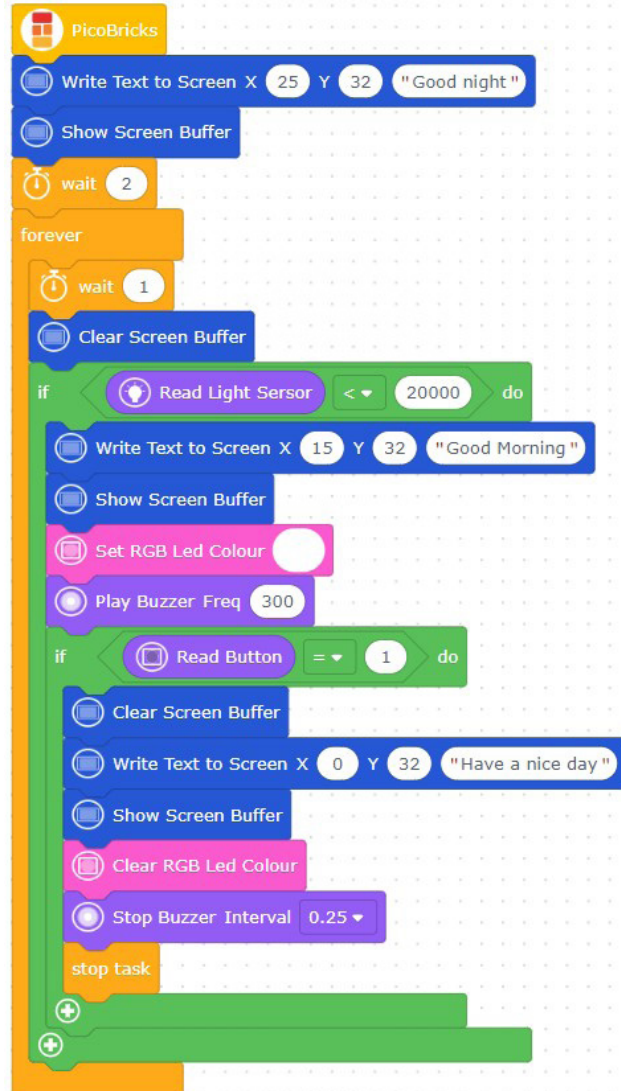


Butona basıldığında, ekranı temizleyen, “Have a nice day” yazısını ekrana yazan, RGB LED’i kapatan ve buzzer’ı durduran kod bloklarını oluşturalım.

İlk oluşturduğun “if” bloğunun içine bu bloğu sürükle.



Kod blokları hazır.



2.10. Know Your Color

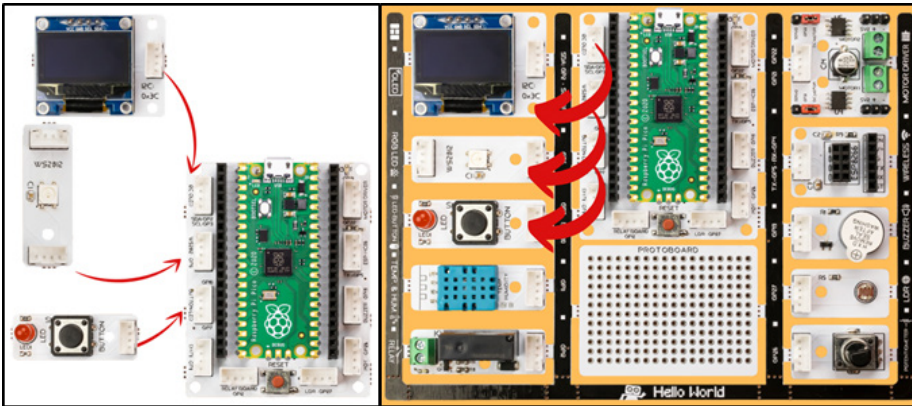
Elektronik sistemler üzerinde LED'ler sıklıkla kullanılır. Her butonun her seçeneğin yanında küçük LED'ler bulunabilmektedir. Tek bir LED'i değişik renklerde yanmasını sağlayarak birden fazla LED'in yaptığı işi tek bir LED ile yapılması sağlanabilmektedir. Bu türde çalışan LED'lere RGB LED denir. Adını 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.

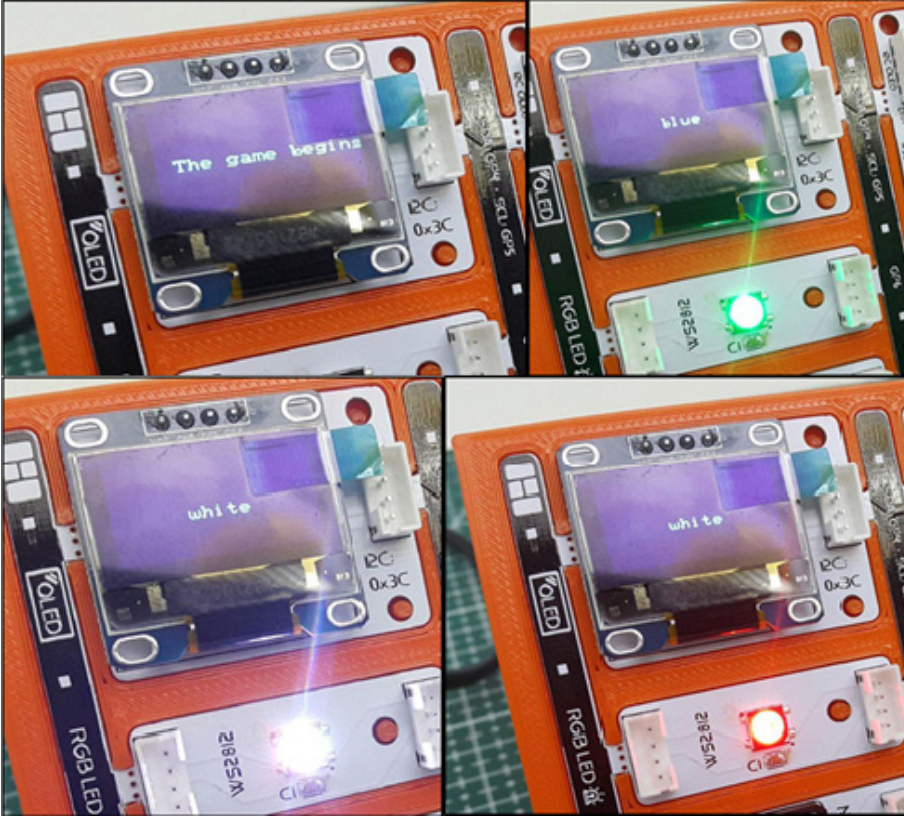
2.10.1. Proje Detayları ve Algoritma

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.

2.10.2. Bağlantı Şeması



2.10.3. Proje Görseli



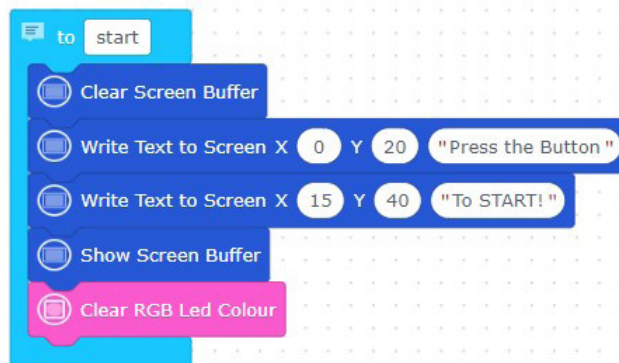
2.10.4. Proje Önerisi

Projeyi biraz daha zorlaştırarak daha eğlenceli bir hale getirebiliriz. Örneğin, renklerin tekrar süresini azaltarak oyunu hızlandırabilirsin.

2.10.5. Projenin PicoBricks IDE ile Kodlanması

Öncelikle, kod için gerekli olan fonksiyonları oluşturarak projemize başlayalım.

“start” adında bir fonksiyon oluşturarak, proje başladığında ekran yazılacak yazıyı belirleyelim ve RGB LED’i kapatalım.



Projede kullanılan her rengin adını OLED ekrana yazdırmak için renk fonksiyonlarını tanımlayalım.

```

to blue
  Clear Screen Buffer
  Write Text to Screen X 45 Y 30 "BLUE"
  Show Screen Buffer
end

to green
  Clear Screen Buffer
  Write Text to Screen X 45 Y 30 "GREEN"
  Show Screen Buffer
end

to red
  Clear Screen Buffer
  Write Text to Screen X 45 Y 30 "RED"
  Show Screen Buffer
end

to white
  Clear Screen Buffer
  Write Text to Screen X 45 Y 30 "WHITE"
  Show Screen Buffer
end

```

“colors” adında bir fonksiyon belirleyerek belirlediğin 4 renkten birini rastgele olarak yakalım.

```

to colors
  if randomColor = 1 do
    Set RGB Led Colour
  else if randomColor = 2 do
    Set RGB Led Colour
  else if randomColor = 3 do
    Set RGB Led Colour
  else
    Set RGB Led Colour
  end
end

```

Belirlediğin 4 rengi rastgele OLED ekrana yazdırmak için “oledcolor” adında bir fonksiyon tanımlayalım.

```

to oledColor
  if randomOLED = 1 do
    red
  else if randomOLED = 2 do
    green
  else if randomOLED = 3 do
    blue
  else
    white
  end
end

```

“check” isimli bir fonksiyon tanımlayarak, butona basıldığında OLED ekrandaki renk ile RGB LED’deki renk aynı ise, “score” adlı değişkeni 10’ar 10’ar arttıran kod bloklarını oluşturalım.

The screenshot shows a function definition block for 'check'. It starts with a 'to check' block. Inside, there is an 'if' block with two conditions: 'Read Button' equals '1' and 'randomOLED' equals 'randomColor'. If both conditions are true, the 'do' block contains a 'change score by 10' block.

Döngü ve koşul yapılarını kullanarak oluşturduğumuz fonksiyonları tanımlayalım ve çağıralım.

The screenshot shows a main program starting with 'PicoBricks' and 'set score to 0'. A 'start' block is followed by a 'forever' loop. Inside the loop, there is an 'if' block for 'Read Button' equals '1'. If true, a 'repeat 10 times' block is executed. Inside the repeat block, 'randomColor' and 'randomOLED' are set to 'random integer from 1 to 4'. Then, 'oledColor' is set to 'randomColor', a 'wait 1' block is added, and the 'check' function is called. After the repeat block, 'Serial Print score' is used. A 'start' block is followed by 'Write Text to Screen' blocks at X=0, Y=10 with text 'Score: ' and at X=50, Y=10 with the 'score' variable. 'Show Screen Buffer' is called, and 'set score to 0' is used to reset the score.

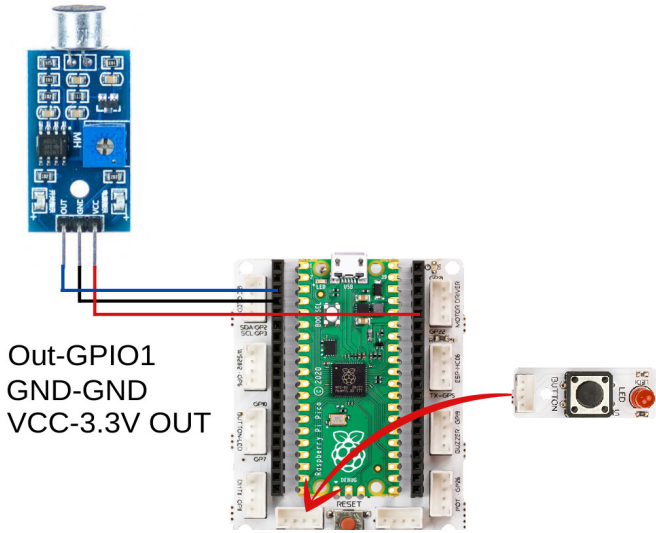
2.11. Magic Lamp

Çoğumuz, filmlerde alkış sesiyle sihirliymişçesine yanıp sönen lambaları veya açılıp kapanan kapıları görmüştür. Çekimlerde bu kapıları kapatan , lambaları söndüren set yardımcıları bulunmaktadır. Peki ya biz bunu otomatik olarak gerçekleştiresek nasıl olur? Ortamda meydana gelmesini beklediğimiz ses şiddeti değişikliğini elektrik sinyaline dönüştüren sensörler vardır. Bunlara ses sensörü denmektedir.

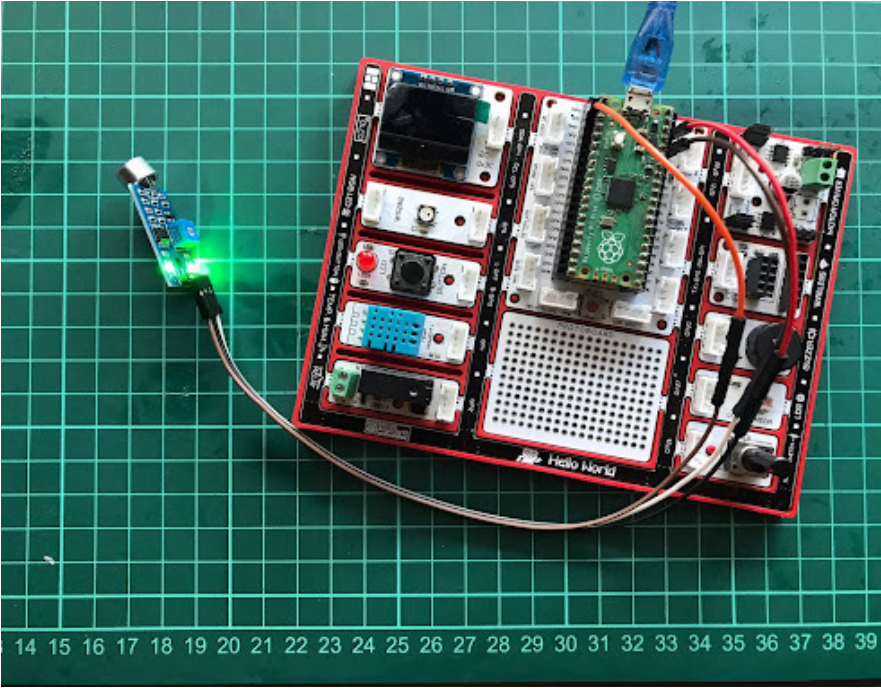
2.12.1. Proje Detayları ve Algoritma

Bu projede picobricks kartı üzerindeki LED modülünü sesle açıp kapatacağız. Picobricks ses seviyesi sensörü kullanarak inşa edeceğimiz projemizde alkış sesi çıkararak açma-kapama işlemlerini gerçekleştireceğiz. Önceki projelerde olduğu gibi sensörlerin kullanıldığı projelerde kodları yazmaya başlamadan önce sadece sensörü çalıştırarak yapmak istediğimiz işlemlerde sensörün hangi değerleri gönderdiğini görmek daha sonra bu değerleri baz alarak projenin kodlarını yazmak ilerlemenizi kolaylaştıracaktır.

2.12.2. Bağlantı Şeması



2.11.3. Projenin Görseli

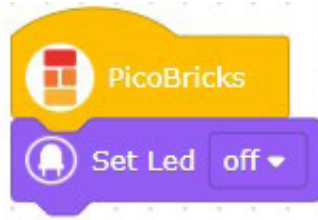


2.11.4. Proje Önerisi

Sihirli lamba projesi ile evindeki masa lambasını sesle kontrol edebilirsin. Röleye lamba bağlayabileceğin gibi evdeki diğer elektrikli aletleri de bağlayarak sesle ya da bluetooth üzerinden cihazları açıp kapatarak kontrol edebilirsin.

2.11.5. Projenin PicoBricks IDE ile Kodlanması

Döngü ve koşul yapılarını oluşturmadan önce, LED'i kapatan kod bloklarını sürükleyelim.



Projede tekrar edecek koşul yapısını ve işlemleri belirleyelim. Koşul ve döngü yapılarında ses sensöründen alınan dijital veriyi kullanmak için "basic" bloklarından "read digital pin" bloğunu kullanacağız. Ses sensörünü bağladığımız pini aşağıdaki gibi yazalım.



Eğer ses sensöründen gelen dijital veri 1 ise ve LED kapalıysa, LED'i açan aksi halde LED'i kapatan döngü ve koşul yapılarını oluşturalım.

```

forever
  set counter to 0
  while Read Digital Pin 1 = 1
    if counter = 1 do
      Set Led off
      set counter to 0
    else
      Set Led on
      set counter to 1
  
```

Kod blokları hazır!

```

PicoBricks
Set Led off
forever
  set counter to 0
  while Read Digital Pin 1 = 1
    if counter = 1 do
      Set Led off
      set counter to 0
    else
      Set Led on
      set counter to 1
  
```


2.12. Smart Cooler

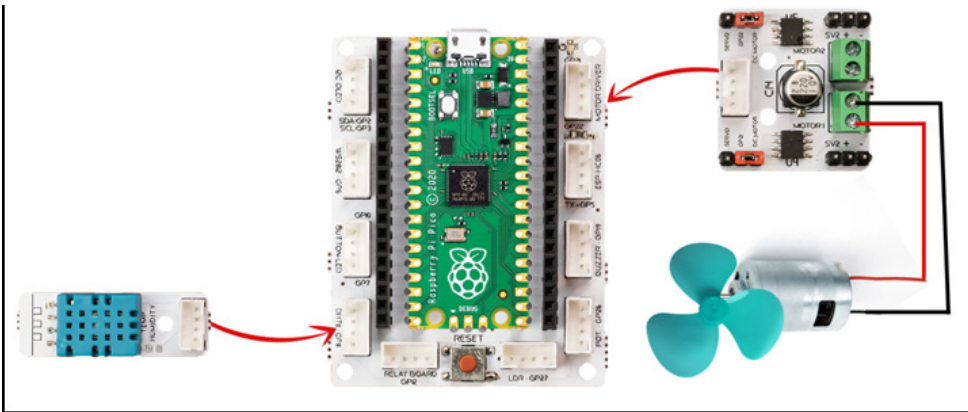
Yaz aylarında serinlemek için kış aylarında ısınmak için klimalar kullanılır. Klimalar ısıtma ve soğutma derecesini bulunduğu ortamın sıcaklığına göre ayarlamaktadır. Fırınlar yemeği pişirirken kullanıcının ayarladığı sıcaklık değerine çıkmaya ve o sıcaklığı korumaya çalışırlar. Bu iki elektronik cihazda sıcaklığı kontrol etmek için özel sıcaklık sensörleri kullanılmaktadır. Ayrıca seralarda sıcaklık ve nem birlikte ölçülür. Bu iki değer istenen düzeyde dengede kalabilmesi için fan ile hava akımını sağlanmaya çalışılır.

Picobricks'te sıcaklığı ve nemi ayrı ayrı ölçebilir ve bu ölçümler ile çevreyle etkileşime girebilirsiniz. Bu projede Picobricks ile sıcaklığa göre fan hızını otomatik ayarlayan bir serinletme sistemi hazırlayacağız. Böylelikle DC motor çalışma sistemini ve motor hız ayarı yapmayı öğreneceksin.

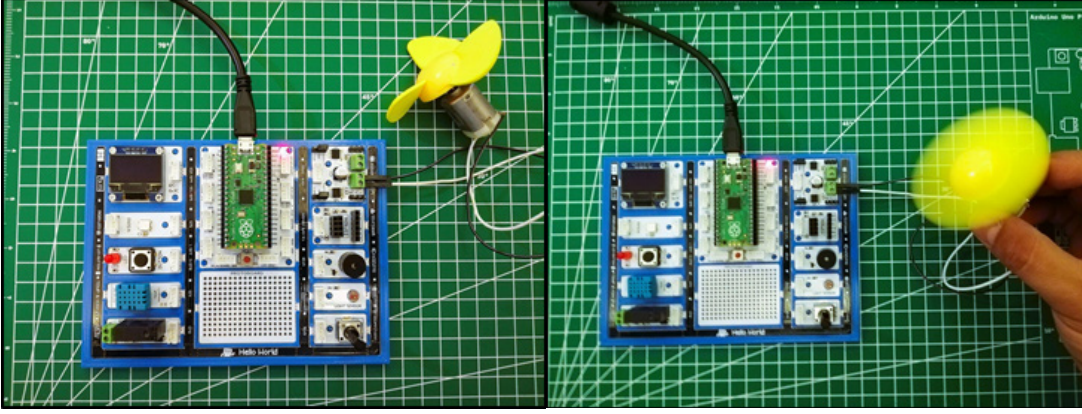
2.12.1. Proje Detayları ve Algoritma

Projemizde öncelikle Picobricks üzerindeki DHT11 sıcaklık ve nem sensörünün ölçtüğü sıcaklık değerlerini görüntüleyeceğiz. Daha sonra sonra bir sıcaklık sınırı belirleyerek DHT11 modülünden gelen sıcaklık değeri bu sınıra ulaştığında Picobricks'e bağlı DC motorun dönmeye başlaması, sıcaklık değeri belirlediğimiz sınırın altına indiğinde ise DC motorun durması için gerekli kodları yazacağız..

2.12.2. Bağlantı Şeması



2.12.3. Proje Görseli



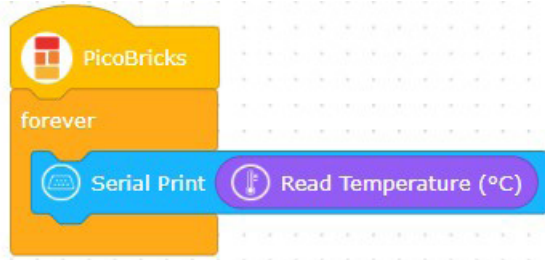
2.12.4. Proje Önerisi

Picobricks üzerindeki OLED ekranın kullanarak sıcaklığı ekrana yazdırabilir ve fanın devreye girdiği sıcaklığı takip edebilirsiniz.

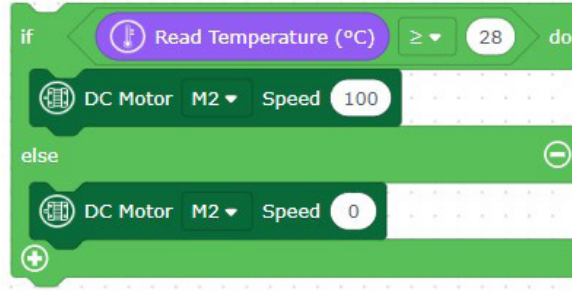
Picobricks modüler yapıdadır, modüller kırılarak ayrılabilir ve grove kablolar ile Pico boarda bağlanarak kullanılabilir. Projemizde yaptığımız akıllı serinletici devresini, robot araba şasesine montajlayarak bulunduğu ortamda otonom olarak gezen ve aynı anda ortamı serinleten bir proje geliştirebilirsiniz.

2.12.5. Projenin PicoBricks IDE ile Kodlanması

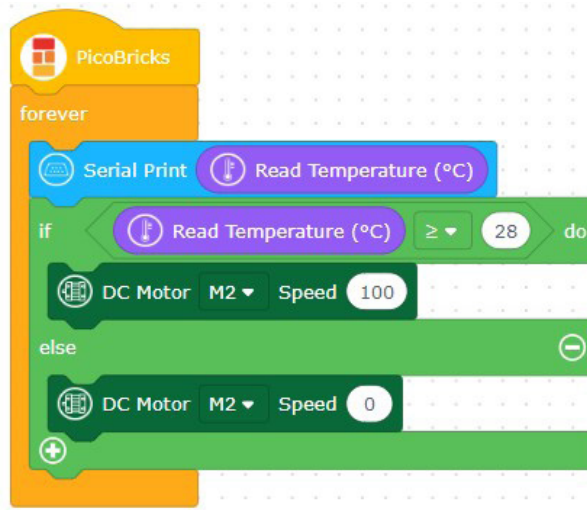
Döngü ve koşul yapısında, PicoBricks DHT modülünden gelen analog datayı kullanmak için, DHTmodülünden gelen değeri seri monitöre yazdırarak ortam sıcaklığını belirleyelim.



Eğer “Read Temperature” bloğunun değeri 28 veya daha fazla ise DC motoru çalıştıran değilse motoru durduran kod bloklarını oluşturalım.



Kod blokları hazır!





2.13. Buzz Wire Game

Projeler her zaman sorun çözmek ve işleri kolaylaştırmak için olmak zorunda değil. Eğlenmek ve gelişmek için de projeler hazırlayabilirsin. Dikkat ve konsantrasyon bir çok insanın geliştirmek istediği özellikleridir. Bunu yapabileceği uygulamalar oldukça ilgi çekicidir. Buzz Wire Game'i Picobricks ile yapmaya ne dersin?

Bilgisayarlar 0 ve 1 ler ile çalışır ifadesini mutlaka duymuşsunuzdur. 0 elektriğin yokluğunu 1 ise varlığını temsil eder. 0 ve 1 ler belirli sayıda ve sıralamadaki kombinasyonlarla bir araya gelerek anlamlı verileri oluştururlar. Elektronik sistemlerde 0 ve 1 ler doğrudan bir durumu kontrol etmek için kullanılabilir. Kapı kapalı mı değil mi? Işık açık mı kapalı mı? Sulama sistemi devrede mi değil mi? bunun gibi bilgiler elde etmek için durum kontrolü yapılmaktadır.

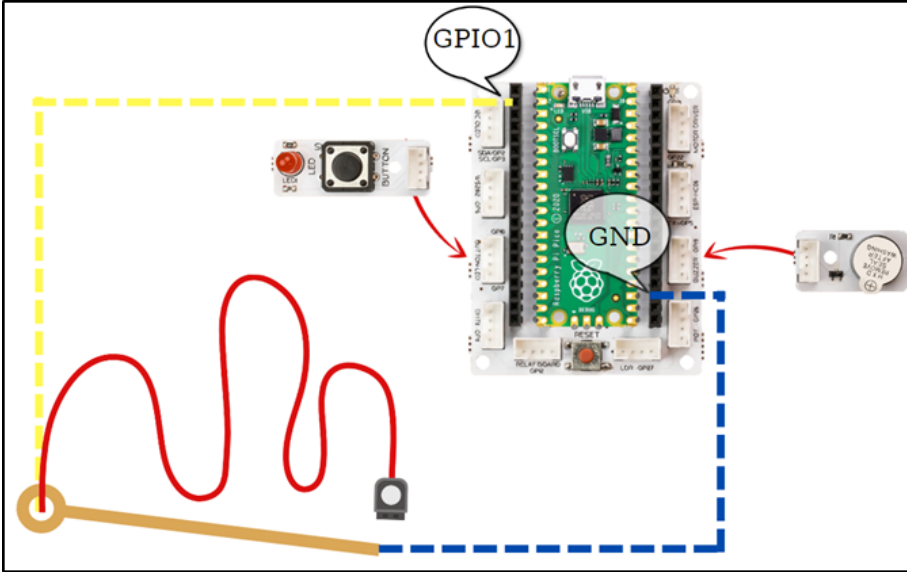
Bu projede Picobricks ile buzzer ve LED modülünü kullanarak iletken tel yardımı ile dikkat ve konsantrasyon geliştirici Buzz Wire Game oyununu elektronik olarak hazırlayacağız. Bu projeyi hazırlarken buton olmayan ancak buton gibi kullanılacak bir giriş tekniği de öğrenmiş olacaksın.

2.13.1. Proje Detayları ve Algoritma

Projeyi hazırlamak için 2 adet erkek-erkek jumper kablo ve 15 cm uzunluğunda iletken bükülebilir tele ihtiyacın var. Oyuncu hazır olduğunda oyunu başlatmak için butona basması istenecek. Butona basıldığında oyuncunun elinde jumper kablo iletken tele değerse Picobricks bunu algılayıp sesli ve yazılı uyarı verecek. Oyun başladıktan bitene kadar geçen sürede OLED ekranda gösterilecek.

Kullanıcı butona bastıktan sonra timer'ı resetliyoruz. Daha sonra Picobricks'in GPIO1 nolu pinine bağlı iletken tel'e 3.3V'luk gerilim vereceğiz. Oyuncunun elinde tuttuğu kablonun bir ucu Picobricks üzerinde GND pinine bağlı olacak. Eğer oyuncu elindeki jumper kabloyu iletken tele dokundurursa GPIO1 nolu pin Pasif/Kapalı/0 konumuna düşecektir. Daha sonra Oyunun bittiğini bildirir ışıklı, yazılı ve sesli geri bildirimde bulunulacak ardından OLED ekran üzerinde geçen süre milisaniye cinsinden gösterilecek. 5 saniye sonunda oyuncunun yeniden başlamak için butona basması istenecek.

2.13.2. Bağlantı Şeması



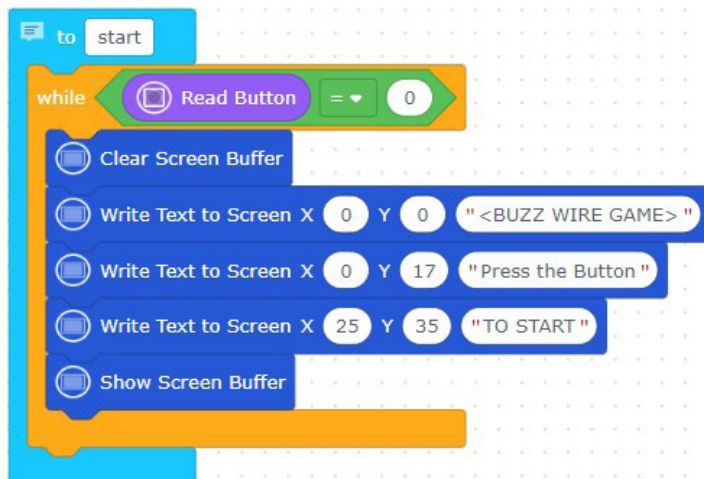
2.13.3. Proje Önerisi

Projeye fiziksel ve yazılımsal geliştirmeler yapabilirsin. Başlangıç ve bitiş noktalarını yalıtkan bant ile kapatarak oyuncunun oyuna başlarken ve bitirirken sorun yaşamasını engelleyebilirsin. Yazılımsal olarak ise oyuncu elindeki kabloyu tele değdirmeden diğer uca getirip bıraktığında butona basar ve skorunu OLED ekranda görmesini sağlayabilirsin.

2.13.4. Projenin PicoBricks IDE ile Kodlanması

İlk olarak, kod için gereken fonksiyonları oluşturarak başlayalım.

- Proje başladığında, ekranda yazacak olan ifadeleri belirlemek için "start" adında bir fonksiyon tanımlayalım. Bu fonksiyonda, butona basılana kadar ekrandaki yazılar silinmeyecek.



Proje tamamlandığında, OLED ekranda yazacak olan ifadeleri belirleyen “finish” adında bir fonksiyon tanımlayalım. Bu fonksiyonda, butona basıldıktan sonra “start” fonksiyonu tekrar çalışacaktır.

```

to Finish
  while Read Button = 0
    Clear Screen Buffer
    Write Text to Screen X 0 Y 10 "GAME OVER!"
    Write Text to Screen X 0 Y 27 "Press the Button"
    Write Text to Screen X 25 Y 45 "TO RESTART"
    Show Screen Buffer
  Stop Buzzer Interval 0.25
  Set Led off
  start

```

Gerekli durum ve döngü yapılarını kullanarak oluşturulan fonksiyonları çağıralım. GPIO pininin değeri “0” ise buzzer’ı çalan, LED’i yakan ve “finish” fonksiyonunu çağıran kod bloklarını oluşturalım.

```

while Read Button = 0
  start
  forever
    Clear Screen Buffer
    Write Text to Screen X 20 Y 30 "GAME STARED"
    Show Screen Buffer
  Serial Print Read Digital Pin 1
  if Read Digital Pin 1 = 0 do
    Play Buzzer Freq 300
    Set Led on
  Finish
  +

```




2.14. Dinosaur Game

Geliştirilecek elektronik sistemler görevini itme, çekme, döndürme, kaldırma, indirme gibi hareketle sonuçlanan işlerle yerine getireceklerse projede aktuatör olarak pnomatik sistemler ya da elektrik motorlu sistemler kullanılır. Picobricks, projelerinde yazdığı kodları harekete geçirebilecek sistemler üretebilmen için iki farklı motor tipini desteklemektedir. DC motor ve DC motorların elektronik olarak hareketlerinin düzenlendiği Servo motorlar. Servo motorlar dönüş açısı değeri verildiğinde o açığa dönen motorlardır. Model uçakların kanatlarında, uçağa yön verebilmek için kanat uçlarındaki yaprakları servo motorlar ile hareket ettirilmektedir. RC teknelerde aracın yönünü değiştirmek için de aynı mantıkla servo motorlar kullanılmaktadır. Ayrıca tam tur dönebilen, akıllı sürekli servo olarak bilinen gelişmiş servo motorlar, evlerimizde kullandığımız akıllı süpürgelerin tekerleklerinde de kullanılmaktadır.

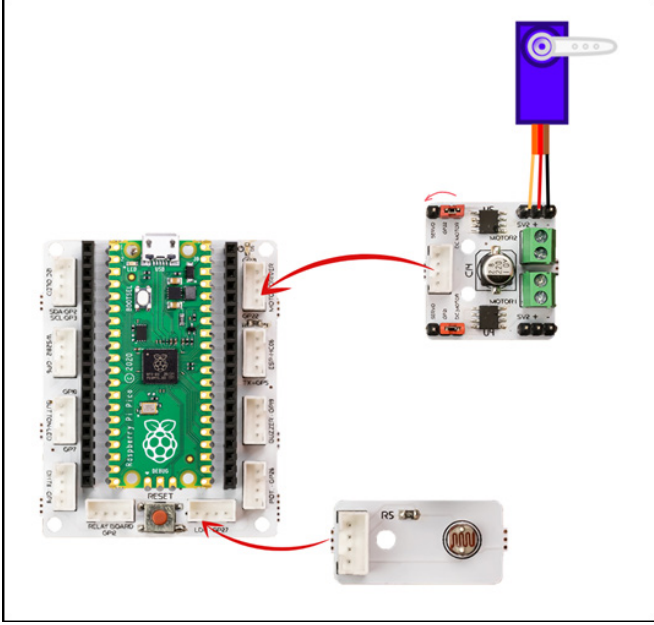
Bu projede PicoBricks ile Servo motorların nasıl kontrol edilebildiğini öğreneceksin.

2.14.1. Proje Detayları ve Algoritma

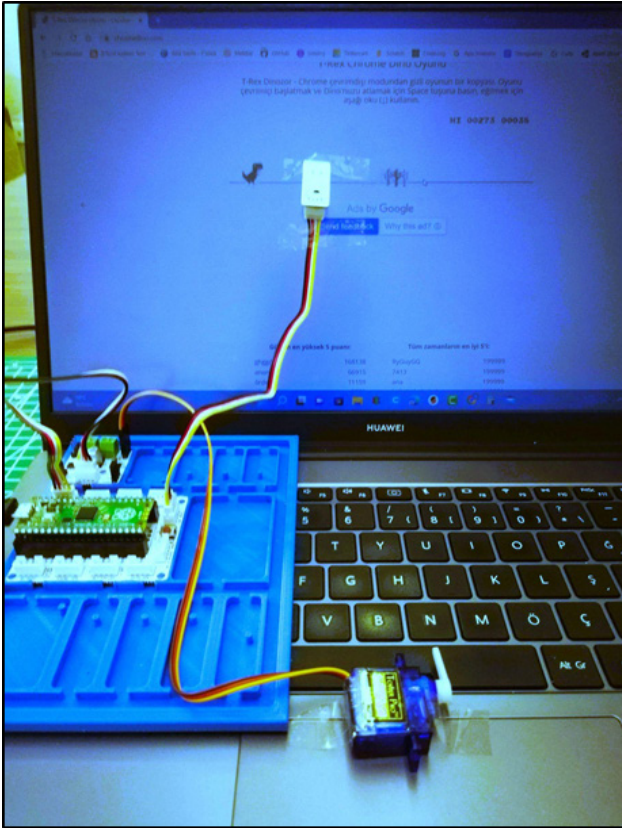
Bu projede Google Chrome offline dinasour game'i otomatik olarak Picobricks'e oynatacağız. Oyunda Picobricks engelleri algılayarak otomatik olarak dinazorun hareketlerini kontrol edecek. Oyun esnasında dinazorun karşısına çıkan engelleri algılamak için picobricks LDR sensör kullanacağız. LDR sensör yüzeyine temas eden ışık miktarını ölçerek analog sinyaller gönderebilmektedir. Sensörü bilgisayar ekranına sabitleyerek beyaz ve siyah renkler arasındaki ışık miktarı farkından yararlanarak dinazorun önüne engel gelip gelmediğini algılayabiliriz. Engel algılandığında ise servo motor kullanarak klavyedeki boşluk tuşuna otomatik olarak basılmasını sağlayabiliriz. Bu sayede dinazor engelleri kolaylıkla aşacaktır. Proje kodlarını yazarken öncelikle LDR sensörü bilgisayar ekranına sabitleyerek beyaz ve siyah zemindeki sensör verilerine okuyacak daha sonra bu verilere göre servo motorun hareket etmesi için gerekli kodları yazacağız.

2.14.2. Bağlantı Şeması

Not: Motor sürücü grove kablo girişinin sağ ve sol tarafında üçlü pinler bulunmakta ve bu pinler 2'li jumper ile kısa devre yapılmaktadır. DC motor kullanırken DC motor tarafında takılı olması gereken jumper, servo motor kullanırken çıkarılarak servo tarafına takılmalıdır.



2.14.3. Proje Görseli



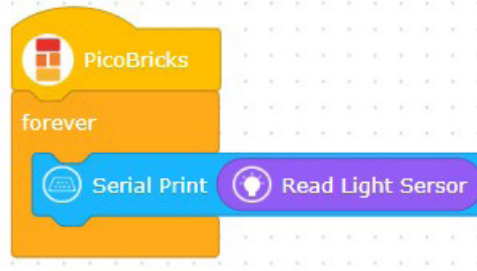
2.14.4. Proje Önerisi

Oyunda ilk başta zemin rengi beyaz figürler siyah renklidir. Belirli bir aşamadan sonra ise renkler tersine dönmektedir. Bu sebeple LDR sensör verileri değişmektedir. Bu problemi çözmek için değişken ve fonksiyon kullanarak oyun beyaz zemindeyken bir kod grubu, siyah zemindeyken diğer kod grubunu çalıştırabilir ya da bu farkı algılaması için ikinci bir LDR sensör takabilirsin.

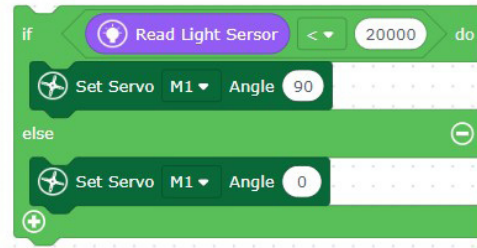
Picobricks ve modülleri basitten karmaşığa birçok proje geliştirmemize imkan sunmaktadır. Günlük hayatta oynadığımız bir bilgisayar oyununu otomatik olarak Picobricks'e oynattığımız bu projeyi geliştirerek minecraft gibi farklı oyunlarda da kullanabilirsin.

2.14.5. Projenin PicoBricks IDE ile Kodlanması

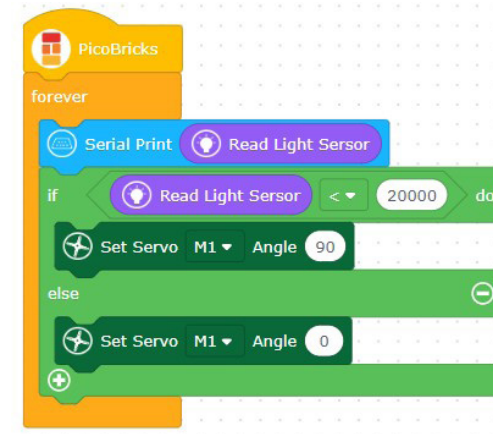
Ortamın ışık değerini seri monitöre yazdırarak kullanacağımız ışık değerini belirleyelim.



Koşul ve döngü yapılarında ışık değerini kullanarak servo motorun açılarını kontrol edeceğiz.



Kod blokları hazır!



2.15. Night and Day

Okulda oynadığınız Gece Gündüz oyununu elektronik olarak oynamaya ne dersin? Öğretmen gece dediğinde kafamızı öne eğip masanın üzerindeki kolumuza yasladığımız, gündüz dediğinde başımızı kaldırdığımız bir oyundur gece-gündüz oyunu. Bu oyun dikkatini ve refleksini kullanacağın bir oyun olacak. Bu projede 0,96" 128x64 piksel I2C OLED ekranı kullanacağız. OLED ekranlar yapay ışık kaynağı olarak kullanılabilirler için ekran üzerindeki karakterleri mercek ve ayna kullanarak büyütebilir ve istediğiniz düzleme yansıtabilirsin. Akıllı gözlükler ve otomobil camlarına bilgilendirme, yol ve trafik bilgisi yansıtabilen sistemler OLED ekranlar kullanılarak yapılabilmektedir.

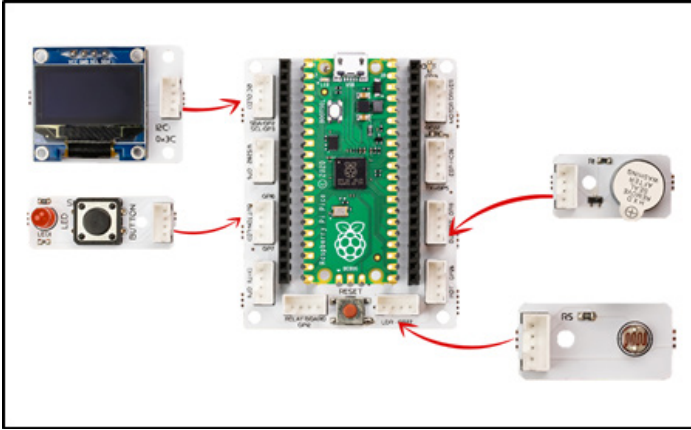
Işık sensörleri buldukları ortamın ışık seviyelerini ölçebilen, fotodiyot da denilen sensörlerdir. Işığa maruz kalan sensörün elektrik geçirgenliği değişmektedir. Biz de kodlayarak ışık sensörünü kontrol edip, ışık miktarının etkilediği elektronik sistemler geliştirebilmekteyiz.



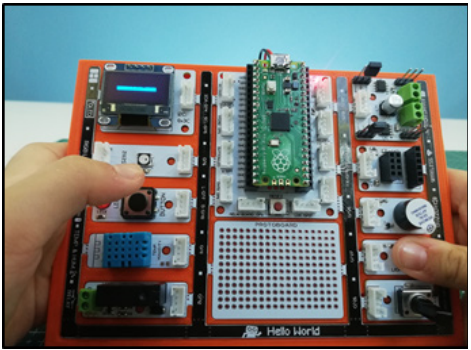
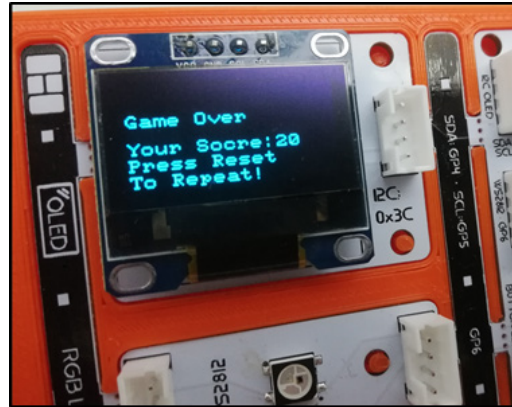
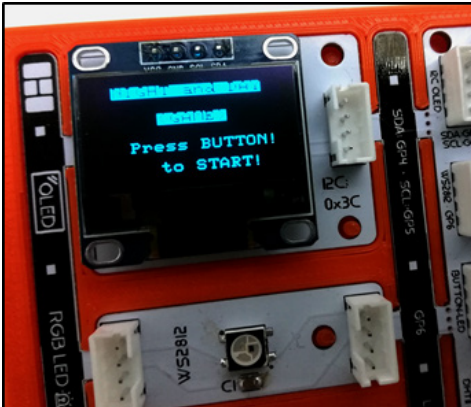
2.15.1. Proje Detayları ve Algoritma

Önce oyuncunun oyuna başlaması için butona basmasını isteyeceğiz. Ardından PicoBricks'in OLED ekranında NIGHT ve DAY ifadelerini rastgele olarak 2'şer saniye boyunca gösterilmesini sağlayacağız. Oyuncu, eğer OLED ekranda yazan kelime NIGHT ise 2 saniye içinde LDR sensörünün üzerini eliyle kapatmalı, eğer OLED ekranda GÜNDÜZ kelimesi yazıyorsa LDR sensörünün üzerinden elini kaldırmalı. Oyuncunun her doğru tepkisi 10 puan kazanmasını sağlayacak, yanlış tepkide ise oyun bitecek ve ekranda oyunun bitmesini bildiren yazılı ifade yer alacak, buzzerdan farklı tonda bir ses çalacak ve OLED ekranda puan bilgisi yer alacaktır. Eğer oyuncu toplam 10 doğru tepki verip 100 puan elde ederse "Congratulation" yazısı ile oyuna yeniden başlayabilmek için RESET butonuna basılmasını bildiren ifade OLED ekranda gösterilip buzzerdan farklı tonda notalar çalacaktır.

2.15.2. Bağlantı Şeması



2.15.3. Proje Görseli



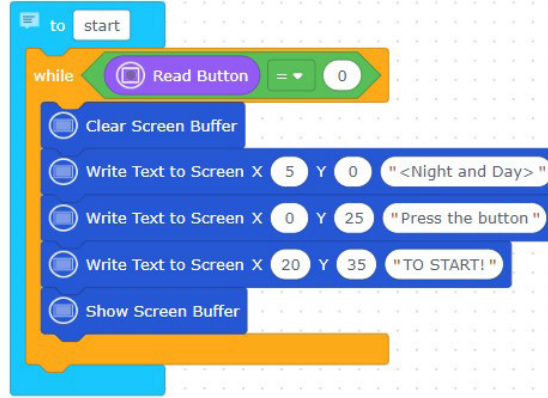
2.15.4. Proje Önerisi

Projeye LDR sensörün bulunduğu ortama göre gönderdiği değerleri alarak oyun içerisinde sensör değerine göre işlem yapılacak sınırın otomatik belirlenmesi, yani LDR sensör değeri kalibrasyonu kodları ekleyerek projeyi geliştirebilirsin. Oyuna zorluk seviyesi ekleyebilirsin. Potansiyometre ile zorluk seviyesi kolay orta ve zor seçimi yaptırılabilir. Kolay seçiliyken kelimelerin değişme süresi 2 saniye orta seçiliyken 1.5 saniye zor seçiliyken 1 saniye olabilir.

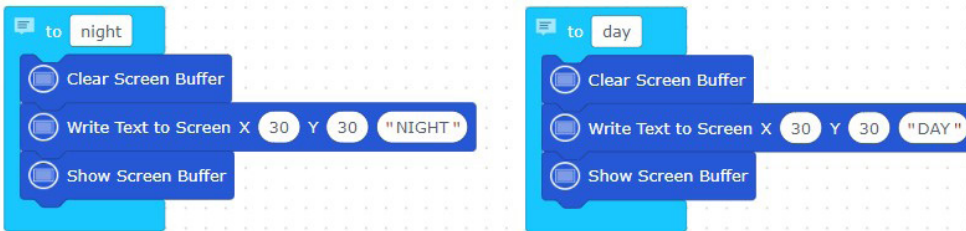
2.15.5. Projenin PicoBricks IDE ile Kodlanması

Öncelikle, kod için gerekli fonksiyonları oluşturarak projeye başlayalım.

- “start” adında bir fonksiyon belirleyerek, butona basılana kadar OLED ekranda yazacak olan ifadeleri belirleyelim.



- Ekranı rastgele olarak “NIGHT” ve “DAY” ifadelerini yazdırmak için “night” ve “day” isimli iki fonksiyonu tanımlayalım.



- Oyun bittiğinde, butona basılana kadar ekrana yazılacak ifadeleri belirlemek için “finish” adında bir fonksiyon tanımlayalım.

```

to finish
  while Read Button = 0
    Clear Screen Buffer
    Write Text to Screen X 5 Y 0 "<Night and Day>"
    Write Text to Screen X 20 Y 20 "Score:"
    Write Text to Screen X 70 Y 20 score
    Write Text to Screen X 20 Y 30 "Time (s):"
    Write Text to Screen X 90 Y 30 counter
    Write Text to Screen X 0 Y 40 "Press the button"
    Write Text to Screen X 20 Y 50 "TO RESTART!"
  Show Screen Buffer

```

- Fonksiyonları çağıran döngü ve koşul yapılarını kullanarak kod bloklarını oluşturalım. Oyucunun belirlediği değer göre 1 ile 2 arasında rastgele sayılar üreten, numara 1 olduğunda ekrana “NIGHT” yazdıran, numara 2 olduğunda ise ekrana “DAY” yazdıran ve LDR değerini ölçen blok yapısını oluşturalım.

```

start
forever
  set counter to 0
  set score to 0
  while score < 100
    set r to random integer from 1 to 2
    if r = 1 do
      night
      wait 1
      change counter by 1
      if Read Light Sensor > 20000 do
        change score by 10
        +
      else
        day
        wait 1
        change counter by 1
        if Read Light Sensor < 10000 do
          change score by 10
          +
        Serial Print Read Light Sensor
  finish

```

2.16. Voice Controlled Robot Car

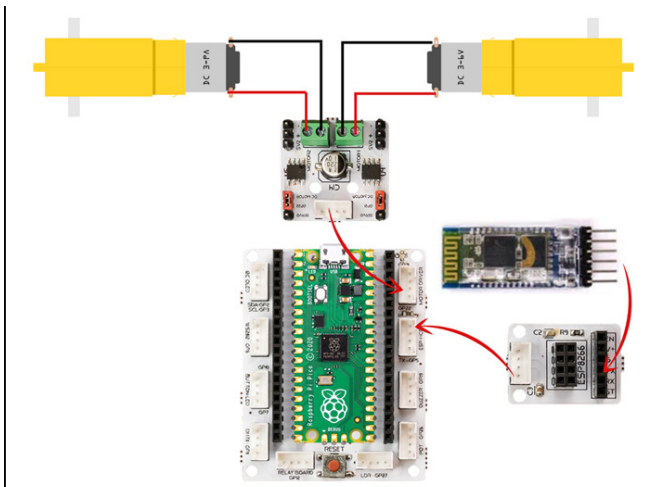
Gelişen ve gelişmeye devam eden yapay zeka uygulamaları insan özelliklerini tanımakta, öğrenmekte ve onun gibi davranmaya çalışmaktadır. Yapay zekayı en kısa haliyle öğrenebilen yazılımlar olarak ifade edebiliriz. Bazen görüntüyü bazen sesi bazen ise sensörlerden topladığı verileri kullanarak öğrenirler. Geliştiricilerinin belirlediği algoritmalar sayesinde bunu yaparlar ve elde ettikleri sonuçlara göre kullandıkları alanlarda karar verme süreçlerinde yardımcı olurlar. Kısaca karar verme sürecinin hızlı ve hatasız yapılması gereken durumlarda artık yapay zeka uygulamaları kullanılmaktadır. Pazarlama alanından savunma sanayisine, eğitimden sağlığa, ekonomiden eğlenceye her alanda yapay zeka verimi arttırıp maliyetleri düşürmektedir.

PicoBricks ile yapacağımız bu projede konuşarak kontrol edebileceğiniz 2WD araba yapacağız. PicoBricks 2 adet 6V DC motoru ve bluetooth ile kablosuz iletişim kurmanızı sağlamaktadır.

2.16.1. Proje Detayları ve Algoritma

Projede setin içinden çıkan robot araba kiti montajlanarak cep telefonu üzerinden kontrol edilecektir. HC05 bluetooth modülü PicoBricks ile cep telefonu arasında kablosuz olarak iletişim kurabilmemizi sağlayan bir modüldür. Projede cep telefonuna yüklenen mobil uygulama sayesinde telefonda gönderilen komutlar HC05 modülü üzerinden PicoBrickse iletilecek ve bu verilere göre de robot araba hareket edecektir. Cep telefonundan ileri, geri, sağ, sol butonları ile robot arabayı yönlendirebileceğimiz gibi sesli komutla da PicoBrickse veriler gönderebiliriz. Projede robot arabanın hareketlerini kontrol etmek için sesli komutlar vereceğiz.

2.16.2. Bağlantı Şeması



Projenin kurulum adımları için kaynaklar bölümüne gidiniz.

2.16.3. Proje Önerisi

Bu projede cep telefonuna yüklediğimiz mobil uygulama üzerinden sesli komutlar vererek robot arabayı hareket ettirdik. Sen de iki eksen robot kol projesindeki pan-tilt mekanizmasına HC05 bluetooth modülü bağlayarak sesli komutlar ile ya da butonlar ile mekanizmayı kontrol edebilirsin. Aynı şekilde bu projedeki robot arabayı sesli komutlar yerine butonlar kullanarak kontrol edebileceğin bir mobil uygulama deneyebilir ya da MIT Appinventor editörü ile projene özel mobil uygulama geliştirebilirsin.

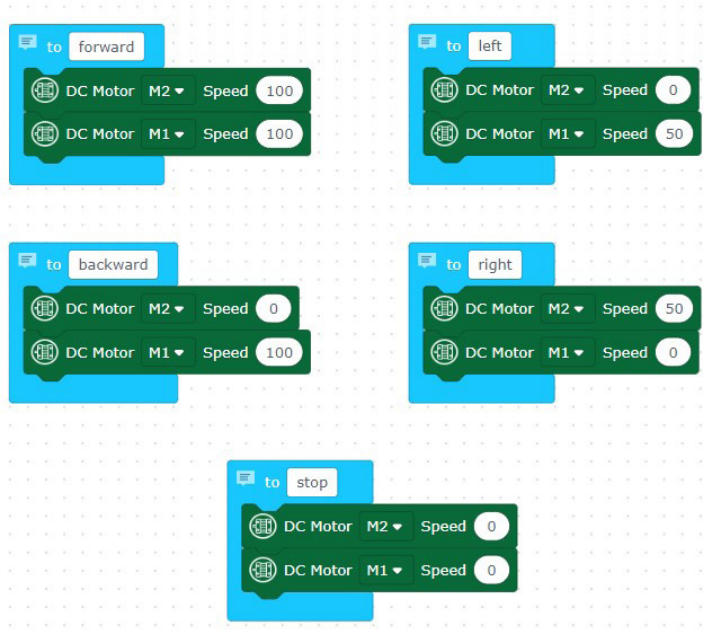
HC05 Bluetooth modülü ile sadece motor sürücü ve motor değil PicoBricks üzerindeki diğer modülleri de çalıştırabilirsin. Örneğin, mobil uygulama üzerinden RGB ledi dilediğin renkte yakabilir, DHT11 modülünden sıcaklık ve nem değerlerini, LDR sensör üzerinden ışık değerlerini okuyabilir, OLED ekrana yazılar yazdırabilirsin. MIT Appinventor editörü ile bu işlemler için özel olarak yazılmış bir mobil uygulama ve uygulamadan gelen verilerin otomatik olarak çalışması için Microblocks ile yazılmış hazır kodlar var. Aşağıdaki linkten Microblocks dosyasını indirip çalıştırarak ve android apk dosyasını indirip telefonuna kurarak tüm bu özellikleri çalıştırabilirsin.

[Download Link](#)

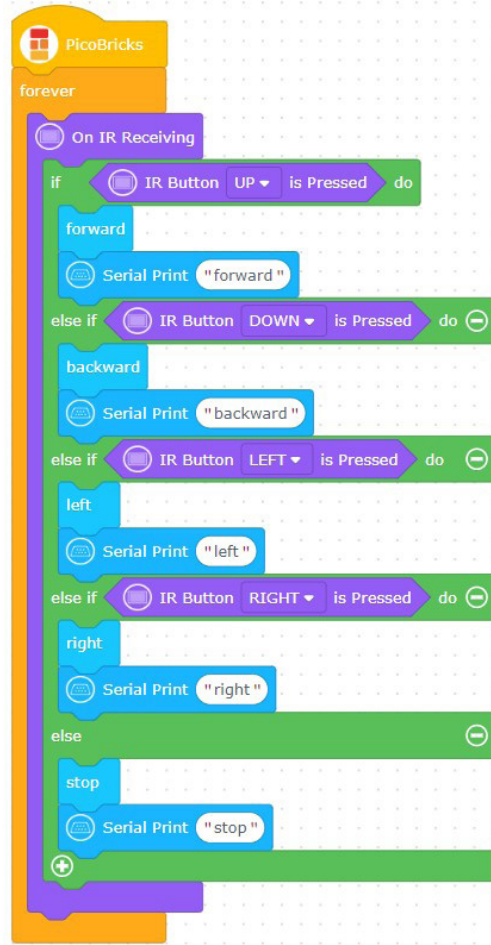
2.16.4. Projenin PicoBricks IDE ile Kodlanması (V1.2 IR)

İlk olarak kod için gerekli olan fonksiyonları oluşturarak projeye başlayalım.

- IR kumandasının butonlarına basılınca, arabanın hareketlerini belirlemek için “forward”, “left”, “backward”, “right” ve “stop” adında beş fonksiyon tanımlayalım.



- IR kumandaya basıldığında, basılan tuşa göre bu beş fonksiyonu çağırarak robot arabayı hareket ettiren kod bloklarını oluşturalım.





2.17. Two Axis Robot Arm

Robot kollar endüstriyel alanda insan gücünün yerini almıştır. Bir insanın taşıyamayacağı ağırlık ve büyüklükteki yükleri taşıma ve döndürme işlerini fabrikalarda robot kollar üstlenmektedir. Milimetrenin binde biri hassasiyetinde konumlandırılabilmesi de insan elinin sergileyebileceği hassasiyetin üzerindedir. Otomobil fabrikalarının üretim videolarını izlediğinde robot kolların ne kadar hayati bir öneme sahip olduğunu göreceksin. Robot denmesinin sebebi programlanarak sonsuz tekrarla aynı işi yapabilmelerinden kaynaklanmaktadır. Kol denmesinin sebebi ise bizim kollarımız gibi eklemlili bir yapıya sahip olmasından kaynaklanmaktadır. Bir robot kolun kaç farklı doğrultuda dönme ve hareket etme kabiliyeti varsa o kadar eksenli olarak ifade edilmektedir. Robot kollar alüminyum ve çeşitli metalleri oyma ve şekil verme işlerinde de kullanılmaktadır. 7 eksenli CNC Router olarak geçen bu cihazlar bir heykeltraşın çamura şekil vermesi gibi metallere şekil verebilmektedirler.

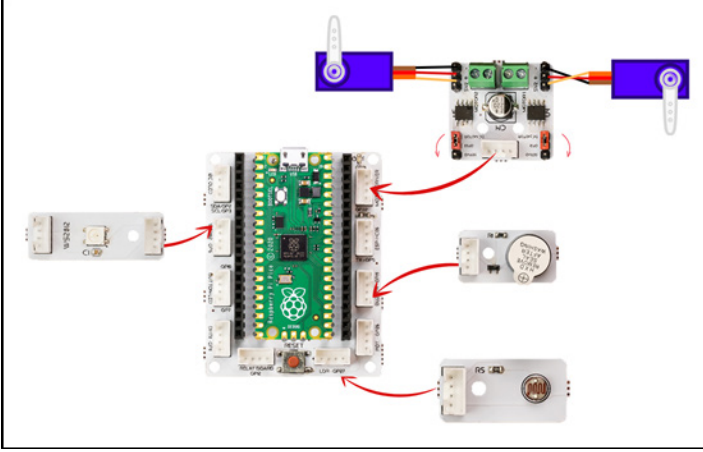
Robot kollarda kullanılma amacına göre bir tür elektrik motoru olan step motor ve servo motorlar kullanılmaktadır. PicoBricks servo motorlarla projeler yapmanıza olanak sağlamaktadır.

2.17.1. Proje Detayları ve Algoritma

Kuruluma hazırlık amacıyla öncelikle servo motorları 0 dereceye ayarlamak için kodlarını yazıp yükleyeceğiz. LDR sensörünün üzerine bir cisim konulduğunda robot kol aşağı eğilecek ve açık olan kaskacını kapatacak. Kaskaç kapandıktan sonra robot kol tekrar yukarı kalkacak. Robot kolun her hareketinin sonucunda buzzer dan kısa bir bip sesi çıkacak. LDR sensörünün üzerine cisim yerleştirildiğinde RGB LED kırmızı renkte yanacak. Cisim robot kol tarafından tutulup havaya kaldırıldığında ise RGB LED yeşil renkte yanacak.

Servo motorun hareketleri çok hızlıdır. Hareketi yavaşlatmak için 30 milisaniye aralıklarla 2 şer derece toplamda 90 derece hareket ile servo motorları kodlayacağız. Kaskacın kapanması için bunu yapmayacağız.

2.17.2. Bağlantı Şeması

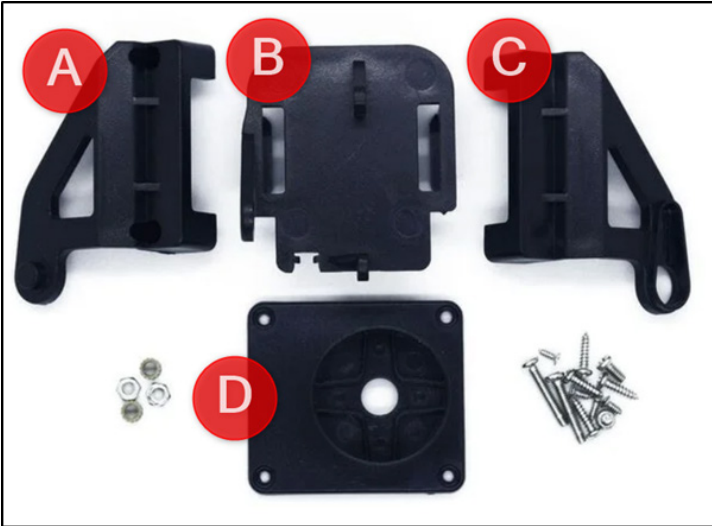


2.17.3. Proje Önerisi

2 eksen robot kol projesine HC05 modülü ekleyerek mobil uygulama ile cep telefonundan kontrol ederek geliştirebilirsin.

2.17.4. Projenin Kurulum Adımları

Projeyi hazırlamak için Pan-Tilt kitinin parçalarını hazırla. 3B yazıcıda basılmış parçaları, Atık karton parçaları , sıcak silikon yapıştırıcı ve makası da yanında bulundur.



1. Öncelikle robot kolun sabit kolunu hazırlayacağız. D parçasının yuvarlak kısmına 8 cm yüksekliğinde kartondan silindir yapın. D parçasına yerleştirip silikon ile yapıştırın.



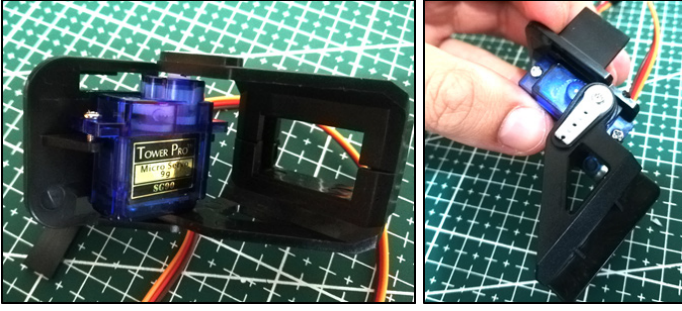
2. C parçasına servo motor paketinden çıkan başlığı biraz kısaltarak yerleştir. Pan Tilt kitinden çıkan en küçük vidalar ile sabitle.



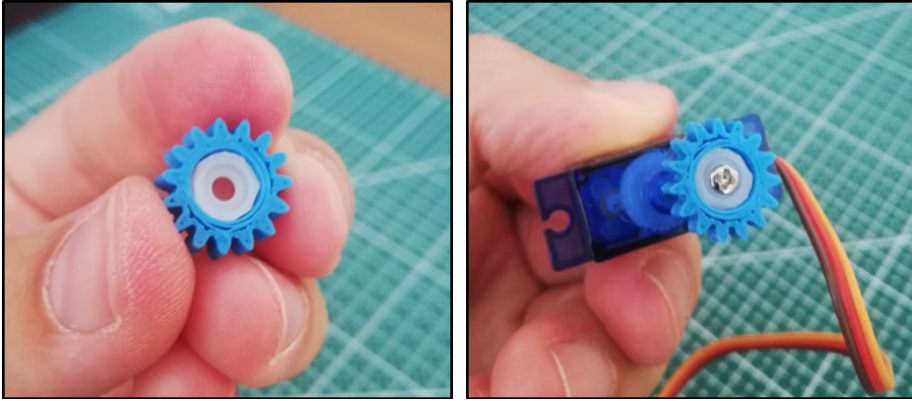
3. A ve C parçalarını sivri uçlu 2 adet vida ile birbirine sabitle.



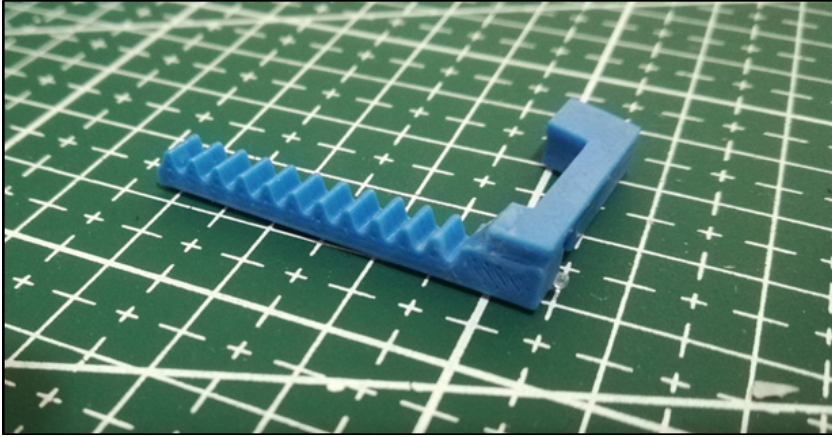
4. Servo motoru C parçasına içten tak. Ardından B parçasına servo motoru yerleştirip vidala.



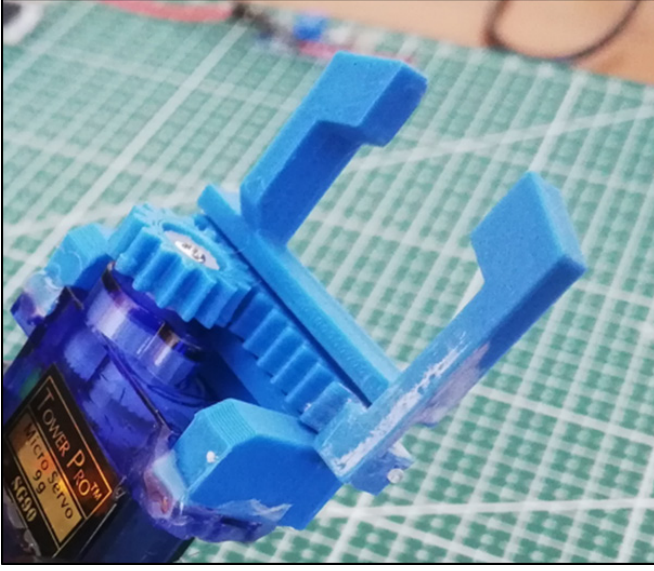
5. Tutucu için 3B yazıcıda bastığın dişli parçasını ortasına servo motor başlıklarından birini keserek dişlinin içine yerleştir. Ardından servo motora vidala.



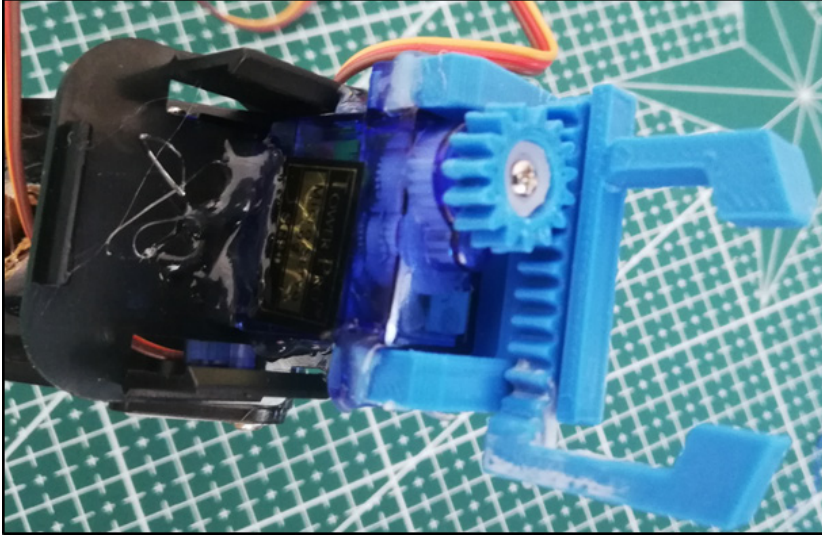
6. 3B yazıcıda basılmış Linear dişli ile tutma kolunu birbirine güçlü yapıştırıcı ile yapıştır.



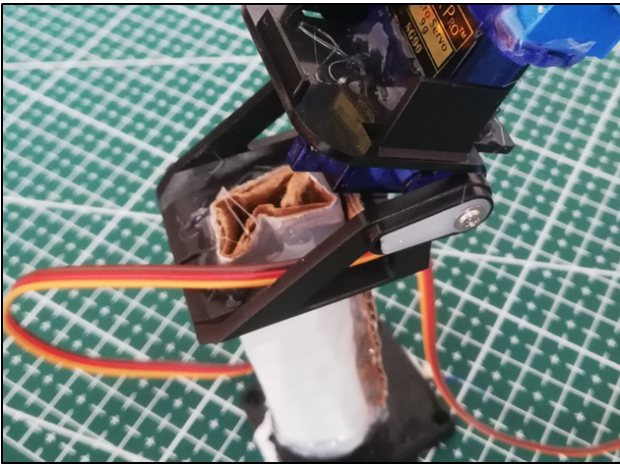
7. Servoyu 3B baskı tutucuya yerleştirip sabitle. İster sıcak silikon ile istersen vidalayarak bunu yapabilirsin. Linear dişliye servo dişlisini yerleştirirken tam açık halde olmasına dikkat et.



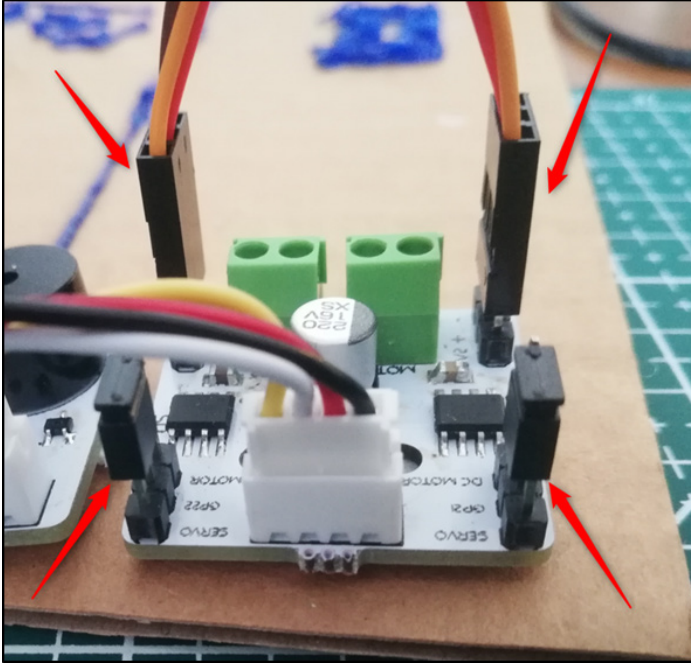
8. Tutma servosu sistemini B parçasına silikon ile yapıştır.



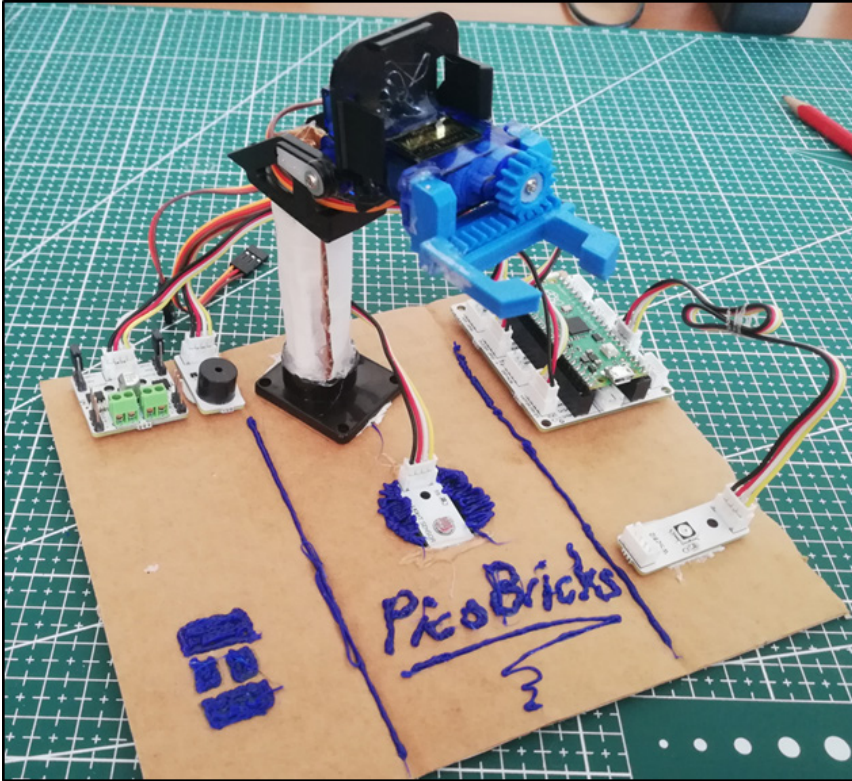
9. 3. adımda hazırladığın parçayı ilk adımda kartondan hazırladığın silindirin üstünden geçirip silikon ile sabitle.



10. Motor sürünün jumperlarını Servo pinlerine tak. Tutucu servonun kablosunu GPIO21'e eğilme servosunun kablosunu GPIO22 ye tak.



11. Motor sürücü, buzzer, LDR ve RGB LED modülünü bir platforma yerleştirip robot kolu uygun şekilde platforma yerleştir. 3B Kalem yazıcı ile projeni dilediğin gibi özelleştirebilirsin.



12. Picobricks'i USB ile ya da 3'lü kalem pil ile Picoboard'da bulunan power jak'ından beslersen Robot kolu çalıştırabilirsin.

2.17.5. Projenin PicoBricks IDE ile Kodlanması

Öncelikle kod için gerekli olan fonksiyonları oluşturarak projeye başlayalım.

- “open” fonksiyonunun içinde robot kolun kısıkaçlarını açmak için servo motorun pozisyonunu ayarlayan kod bloklarını tanımlayalım.

```

to open
  Set Servo M1 Angle 90
  Play Buzzer Freq 587
  Stop Buzzer Interval 0.5
  wait 1

```

- “close” fonksiyonunun içinde robot kolun kısıkaçlarını kapatmak için servo motorun pozisyonunu ayarlayan kod bloklarını tanımlayalım.

```

to close
  Set Servo M1 Angle 160
  Play Buzzer Freq 587
  Stop Buzzer Interval 0.5
  wait 1

```

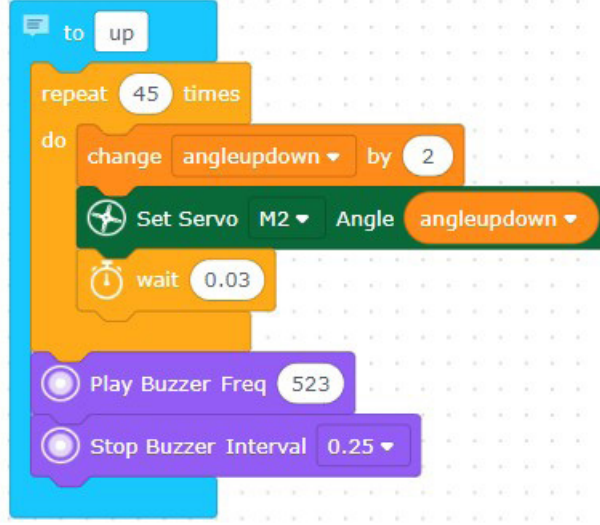
- “down” fonksiyonunun içinde robot kolu aşağı eğmek için servo motorun pozisyonunu ayarlayan kod bloklarını tanımlayalım.

```

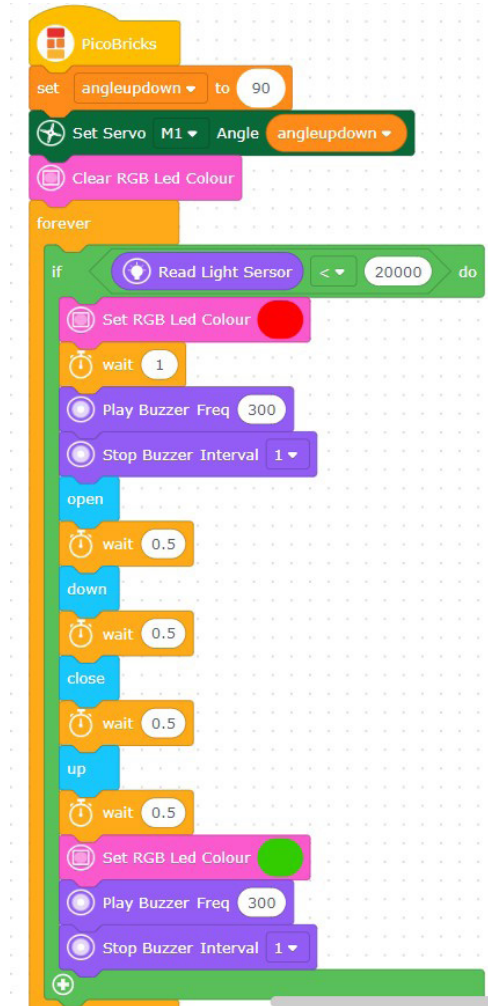
to down
  repeat 45 times
    do
      change angleupdown by -2
      Set Servo M2 Angle angleupdown
      wait 0.03
  Play Buzzer Freq 523
  Stop Buzzer Interval 0.25

```

- “up” fonksiyonunun içinde robot kolu yukarı kaldırmak için servo motorun pozisyonunu ayarlayan kod bloklarını tanımlayalım.



Döngü ve koşul yapılarını kullanarak oluşturduğumuz fonksiyonları çağıralım. LDR sensörü 20000'den az değer algıladığında, 0.5 saniye aralıklarla “open”, “close”, “down” ve “up” fonksiyonlarını çağırarak kod bloklarını oluşturalım.



2.18. Smart House

İşyerleri, fabrikalar, evlerimiz hatta hayvan barınakları... Yaşam alanlarımızı istemediğimiz davetsiz misafirlere karşı korumak için kullanılacak farklı elektronik sistemler bulunmaktadır. Bu sistemler ev ve iş yeri güvenlik sistemleri olarak üretilmekte ve pazarlanmaktadır. Güvenlik kameralarının ürettiği görüntülerin işlenip yorumlandığı sistemler bulunduğu gibi insan bedenini ve hareketlerinin sensörler ile tespit edip harekete geçen güvenlik sistemleri de mevcuttur. Güvenlik sistemleri bir tür alarmlı saat gibi kurulur ve belirlenen zaman diliminde tanımlanmayan bir hareketlilik algılandığında sesli ve ışıklı uyarılar verir. İşletme veya ev sahibine bildirimde bulunur ayrıca güvenlik birimlerine de otomatik bildirimlerde bulunabilmektedir.

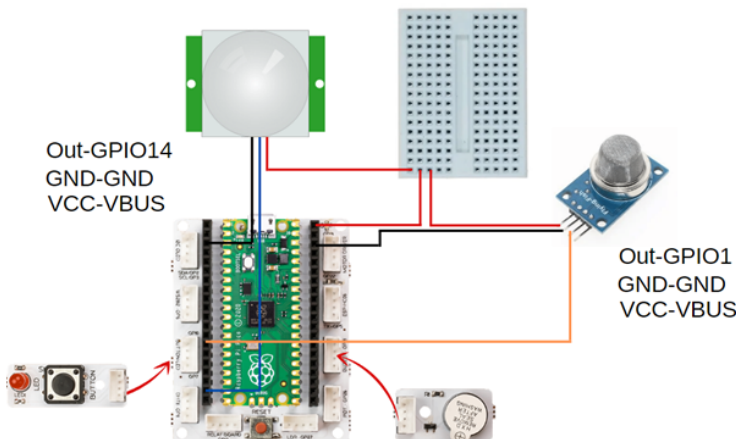
Gaz kaçağı yangın vb. durumlarda zehirlenmeleri önlemek için ev ve işyerlerinde gaz sensörleri kullanılır. Olumsuz bir durumda yüksek sesle alarm vererek ortamda yaşayan insanlar uyarılır.

PicoBricks ile HC-SR501 ve MQ-2 gaz sensörü kullanarak maket bir akıllı ev projesi hazırlayacağız. Bu sensör HC-SR501 , PIR sensörü olarak da ifade edilmekte, insan vücudunun yansıttığı kızılötesi dalgaların değişimlerini yakalayarak hareketi tespit etmektedirler.

2.18.1. Proje Detayları ve Algoritma

HC-SR501 PIR sensörü hareket algıladığında 3 saniye boyunca dijital çıkış vermektedir. Proje maketinde Picoboard, buzzer ve buton LED modülünü kullanacağız. Tüm parçalar maketin içinde olmalı. Picobricks başladığında alarm sisteminin devreye girmesi için butona basılması gerekmektedir. Butona basıldıktan sonra elin maketi içinden çekilmesi için 3 saniye bekleme koymalıyız. 3 saniyenin sonunda kırmızı LED yanar ve alarm sistemi devreye girer. Alarm sistemi devredeyken bir hareket algıladığında kırmızı LED yanıp sönmeye başlar ve buzzer'dan alarm sesi çalınır. Susturmak için Picobricks'in yeniden başlatılması gerekmektedir. MQ-2 sensörü ise sürekli devrededir. Zehirli bir gaz algıladığında ise buzzer ve kırmızı LED ile bunu bildirecektir.

2.18.2. Bağlantı Şeması

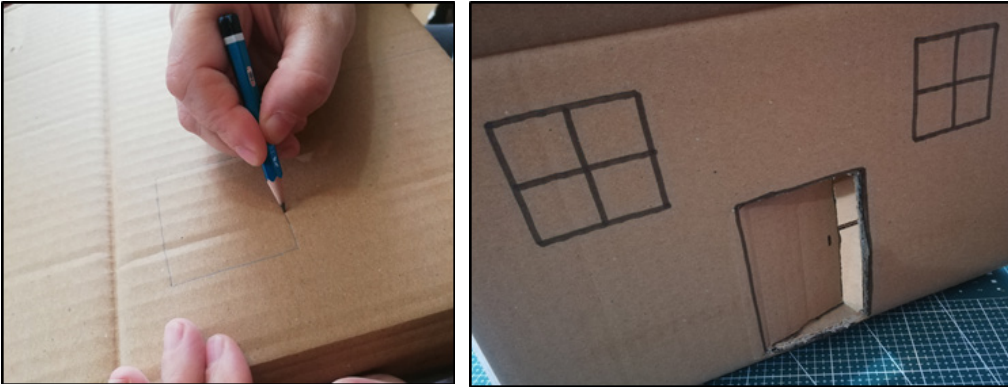


2.18.3. Proje Önerisi

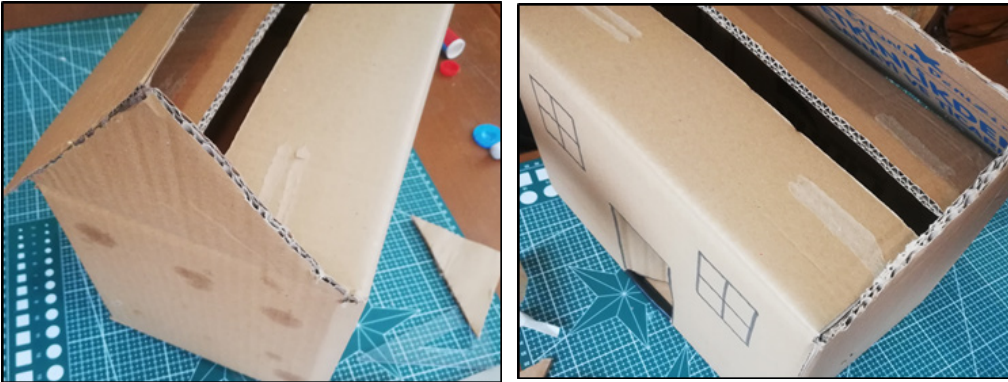
25. proje olan akıllı serayı yaptıktan sonra akıllı ev projesine ESP8266 modünü ekleyerek eve hırsız girdiğinde ev sahibinin telefonuna internet üzerinden bildirim gönderebilir ve projeyi IOT projesine dönüştürebilirsin. Dalgıç pompa ile ev tavanına yangın söndürme boruları çekebilir, bu sayede evde yangın çıktığında otomatik olarak söndürülmesi sağlayabilirsin.

2.18.4. Projenin Kurulum Adımları

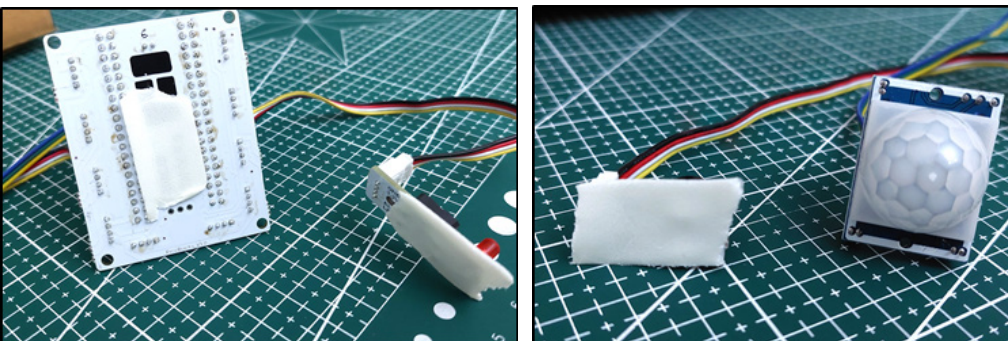
Projeyi çalıştırmak için karton bir kutuyu bir maket eve çevirmen gerekiyor. Bir makas, birkaç kalem, bant, yapıştırıcı ve bir maket bıçağına ihtiyacınız olacak. Kutunun üstüne bir kalemle pencereleri ve kapıları çizin. Kapı bölümlerini maket bıçağı ile kesin.



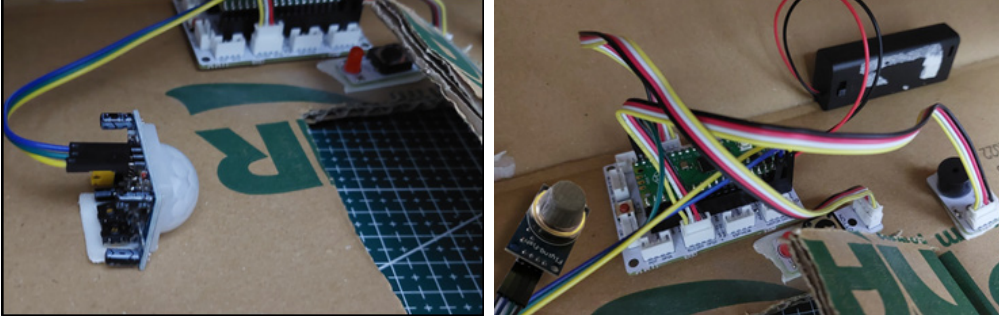
Çatı bölümünü yapmak için başka bir karton kutuyu kullanabilirsiniz.



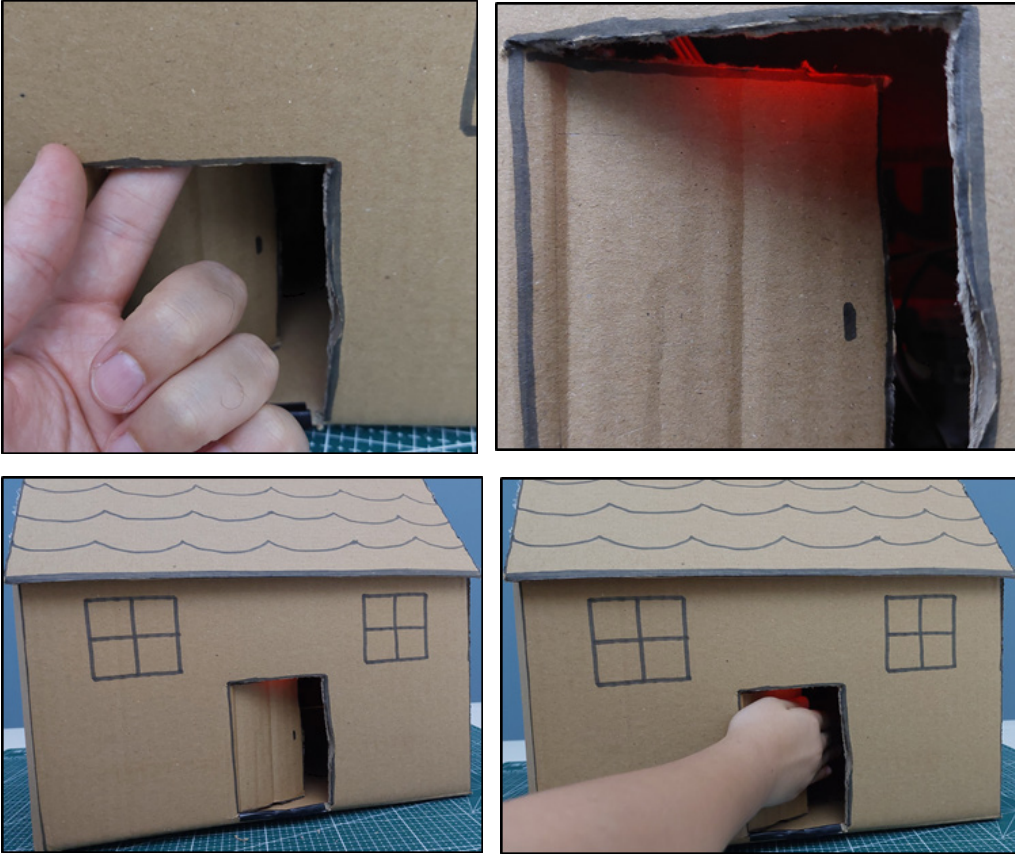
Çift taraflı bant ile PicoBricks modüllerini yapıştırın.



Picobricks'in modüllerini maket evinin içine yerleştirir. PIR sensörünü içeriden kapıyı doğrudan görecek şekilde konumlandırın. Buton modülünü ise içeriden kapının hemen üstüne yapıştırır.



Pil kutusunu PicoBricks'e bağlayıp açtığında kodlar çalışmaya başlayacaktır. Butona bastıktan 3 saniye sonra alarm sistemi devreye girecek ve kırmızı LED yanacak. Elini kapıdan içeri sokar sokmaz buzzer modülü alarmı çalmaya başlayacaktır.

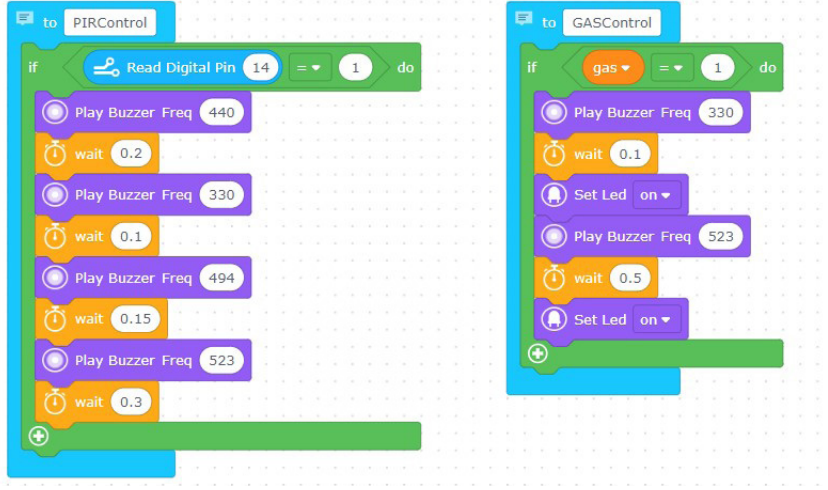


Çakmak gazını evin içine tuttuğunuzda alarm sistemi tekrardan devreye girecektir.

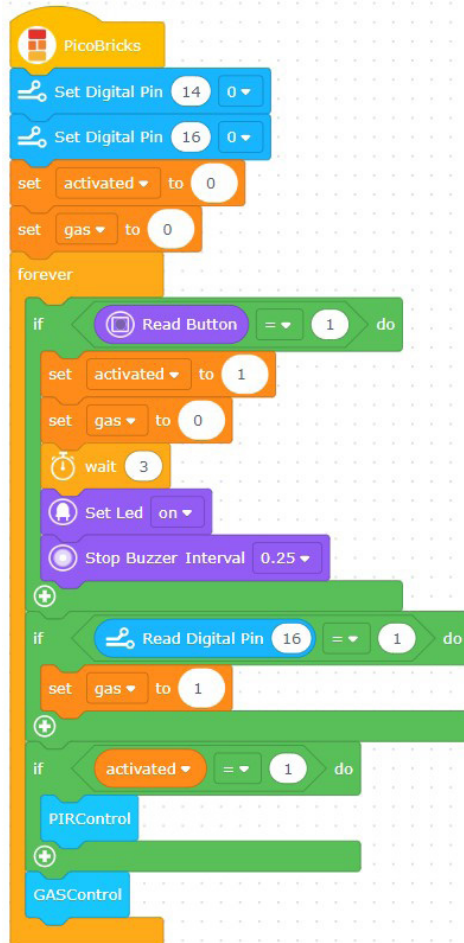
2.18.5. Projenin PicoBricks IDE ile Kodlanması

İlk olarak, kod için gerekli olan fonksiyonları oluşturarak projeye başlayalım.

- PIR sensörünün ve Gas sensörünün değerlerini kontrol ederek uyarı veren “PIRControl” ve “GASControl” adında iki fonksiyon oluşturalım.



- Koşul ve döngü yapılarını kullanarak bu fonksiyonları çağırın. PIR sensörünü GPIO 14 pinine, Gas sensörünü ise GPIO 16 pinine bağlayın. Eğer Gas ve PIR sensörlerinin değeri “1” ise, “GASControl” ve “PIRControl” fonksiyonlarını çağırın kod bloklarını oluşturalım.



2.19. Piggy Bank

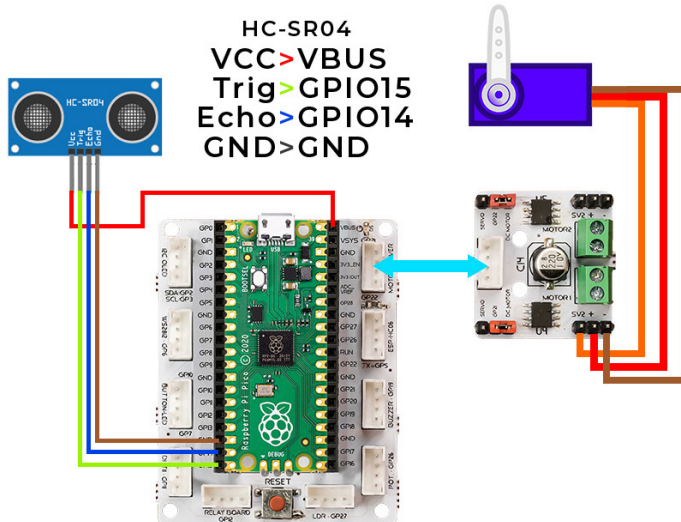
Ultrasonik sensörler ses dalgalarından etkilenerken elektriksel değişim gösteren sensörlerdir. Bu sensörler kulağımızın algılayamayacağı frekansta ses dalgaları gönderirler ve yansıyan ses dalgalarının geri geliş süresinin hesaplanması ile mesafe bilgisi üretirler. Biz programcılar ise ölçülen mesafeyi ve mesafedeki değişimleri anlamlandırarak projeler geliştiririz. Otomobillerin ön ve arkasında bulunan park sensörleri ultrasonik sensörlerin günlük yaşamda en çok karşımıza çıktığı yerlerdir. Doğada bu yöntemle yönünü bulan canlıyı biliyor musun? Yarasalar kör oldukları için çıkardıkları seslerin yansımalarından yollarını bulurlar.

Bir çoğumuz para biriktirmeyi severiz. Azar azar biriktirdiğimiz paraların ihtiyaç halinde işe yaraması çok güzel bir duygudur. Bu projede kendine çok eğlenceli ve sevecen bir kumbara yapacaksın. Kumbarayı yaparken servo motor ve ultrasonik mesafe sensörünü kullanacaksın.

2.19.1. Proje Detayları ve Algoritma

Bu projede HC-SR04 ultrasonik mesafe sensörü ve SG90 servo motor kullanılacaktır. Kumbaranın haznesine kullanıcı para bıraktığında mesafe sensörü yakınlığı algılayacak ve PicoBrick'se gönderecektir. PicoBricks de bu bilgiye göre servo motoru çalıştırarak kolu yukarı kaldırarak, parayı kumbaranın içine atacak ve kol tekrar aşağı inecektir.

2.19.2. Bağlantı Şeması



2.19.3. Proje Önerisi

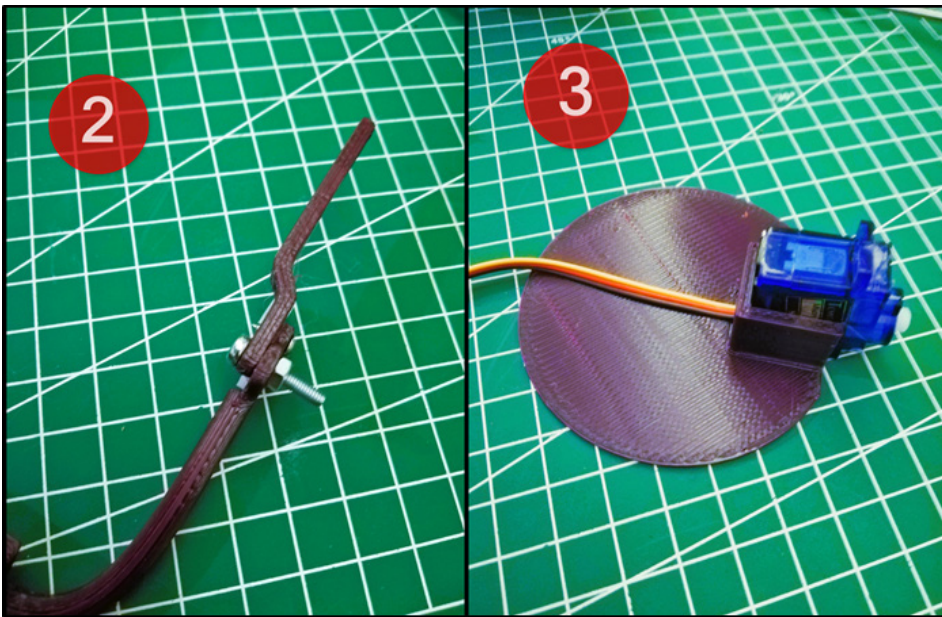
Obur kumbara projesine RGB led modülü ekleyerek her para atıldığında dilediğin renkte ışık yanmasını sağlayabilir, buzzer ekleyerek her para atıldığında ses çıkmasını sağlayabilirsin. Ayrıca OLED ekran ekleyerek kaç kere para atıldığını ekrana yazdırabilirsin.

2.19.4. Projenin Kurulum Adımları

Projenin orijinal dosyaları ve yapım aşamaları sayfasına kaynaklardan ulaşabilirsin. Bu linkteki çalışmadan farklı olarak biz HC-SR04 ultrasonik mesafe sensörü kullanacağız. HC-SR04 ultrasonik mesafe sensörüne göre güncellenmiş 3D çizim dosyaları bu linkten indirebilir ve 3D baskı alabilirsin.

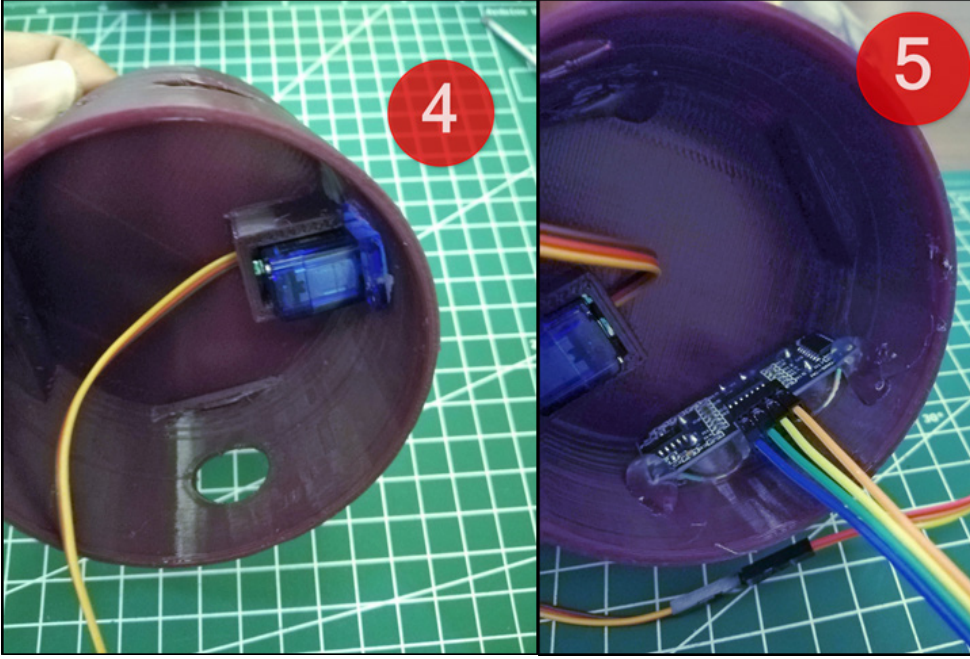


1: 2 vida ile servo motorun plastik aparatlarını yerleştirin.



2: Kumbara kolunun ikinci parçasını M3 vida ve somun ile haznenin bulunduğu birinci parçaya sabitle.

3: Servo motorun kablosunu geçirerek yuvasına yerleştir.



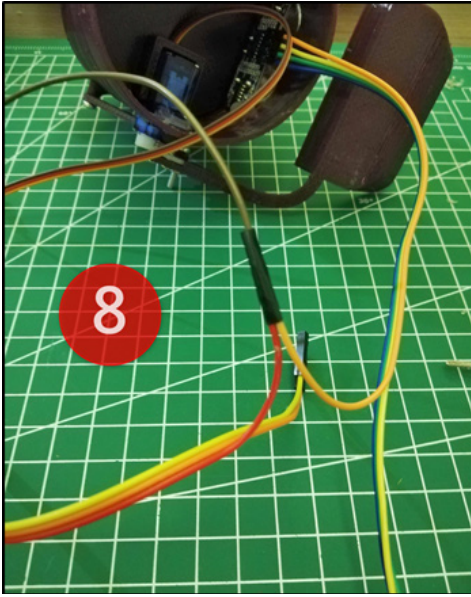
4: Servo motor ve yuvasını kumbaranın gövdesine yerleştir. Burada sıcak silikon kullanabilirsin.

5: Ultrasonik mesafe sensörünü kumbara gövdesine yerleştir ve sıcak silikon ile sabitle.



6: Kumbara kolunu servo motora tak ve üst kapağa M3 vida ile sabitle.

7: Kumbara kolunu gövdeye M2 vida ile sabitle.



8: Servo motor ve ultrasonik mesafe sensörün kablolarını tak ve güç kablolarını birleştir.

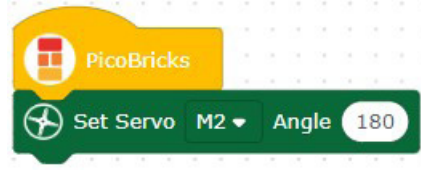
9: Devre şemasına göre servo motor ve ultrasonik mesafe sensörün kablolarını PicoBricks'e tak.



10: PicoBricks'in USB kablosunu tak ve kabloları toparlayarak alt kapağı tak. Hepsi bu kadar.

2.19.5. Projenin PicoBricks IDE ile Kodlanması

Servo motoru başlangıç pozisyonuna ayarlayarak projeye başlayalım.

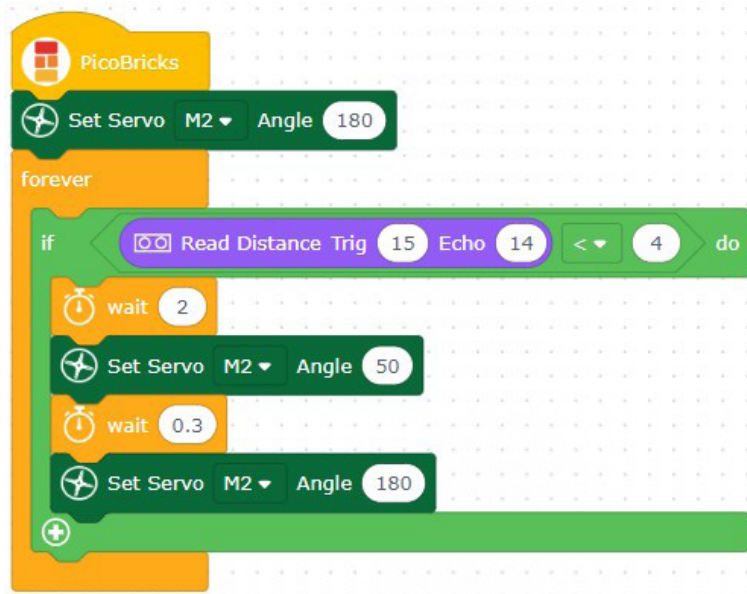


Proje başladığında sürekli çalışan koşul yapısını belirleyelim.

- 2 saniye bekleyen,
- 2. servo motorun açısını 50 derece olarak ayarlayan,
- son olarak eğer mesafe sensörünün değeri 4'ten az ise, 300 milisaniye bekledikten sonra servo motoru ilk pozisyona getiren kod bloklarını oluşturalım.



Kod blokları hazır!



2.20. Automatic Trash Bin

Covid-19 pandemisi insanların günlük yaşamındaki rutinlerini bir çok alanda deđiřtirdi. Temizlik, alıřma, alışveriř, sosyal hayat gibi bir ok alanda insanlar uymak zorunda olduđu bir dizi yeni kurallarla tanıştı. Covid-19 yeni iş alanlarının dođmasına ve gelişmesine zemin oluřturduđu gibi bazı ürünlerinde ön plana çıkmasını sađlamıştır. El hijyenin ok önemsendiđi bir dönemde kimse öpünü atmak için öp kovasının kapađına dokunmak istemezdi.

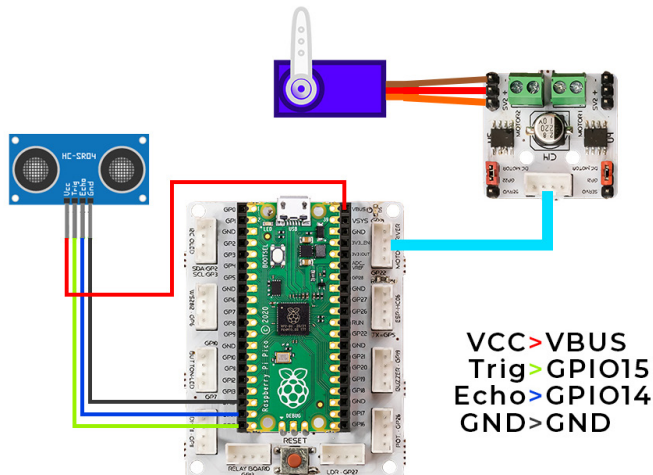
Yanına yaklařıldıđında kapađı otomatik açılan olduđunda ise tek bir hareketle içindeki pořeti büzüp ıkarıp atmaya hazır hale getiren öp kovaları maliyetinin ok üstünde fiyatlara alıcı buldu. Ayrıca otomatik dezenfektan makineleri elimizi altına tuttuđumuzda belirli bir miktar sıvıyı avucumuzun içine boşaltmakla temassız bir hijyen sađlıyordu. Otomatik dezenfektan sıkıcılarında maliyetinin ok üzerindeki fiyatlara raflarda yer aldı. Bu iki üründe alıřma sistemi aısından benzerlikler yer almaktadır. Otomatik dezenfektan sıkıcılarında dođrudan elektrik motorlu bir pompa sıvıyı dıřarı aktardıđı gibi bazı modellerde de servo motorun gücü ile pompalanma sistemine dayanan cihazlar bulunmaktaydı. Otomatik öp kovalarında ise kapađı açan servo motor kullanılmakta el hareketini algılamak için ise kızılötesi ya da ultrasonic sensörler kullanılmaktaydı.

Bu projede PicoBricks ile ultrasonic sensör ve servo motor kullanarak odan için hareketli ve otomatik řık bir öp kovası yapacaksın..

2.20.1. Projenin Detayları ve Algoritma

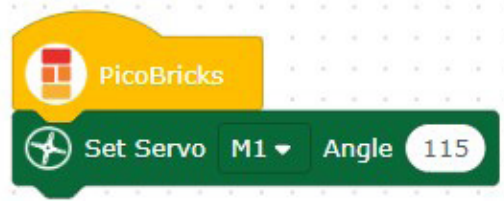
Bu projede HC-SR04 ultrasonic mesafe sensörü ve SG90 servo motor kullanılacaktır. öp kutusunun kapađının önüne kullanıcı elini yaklařtırdıđında mesafe sensörü yakınlıđı algılayacak ve PicoBricks'e gönderecektir. PicoBricks de bu bilgiye göre servo motor alıřtırarak öp kovasının kapađını açacak kısa bir süre sonra tekrar ařađı indirecektir.

2.20.2. Bađlantı řeması



2.20.3. Projenin PicoBricks IDE ile Kodlanması

Servo motoru başlangıç pozisyonuna alarak projeye başlayalım.

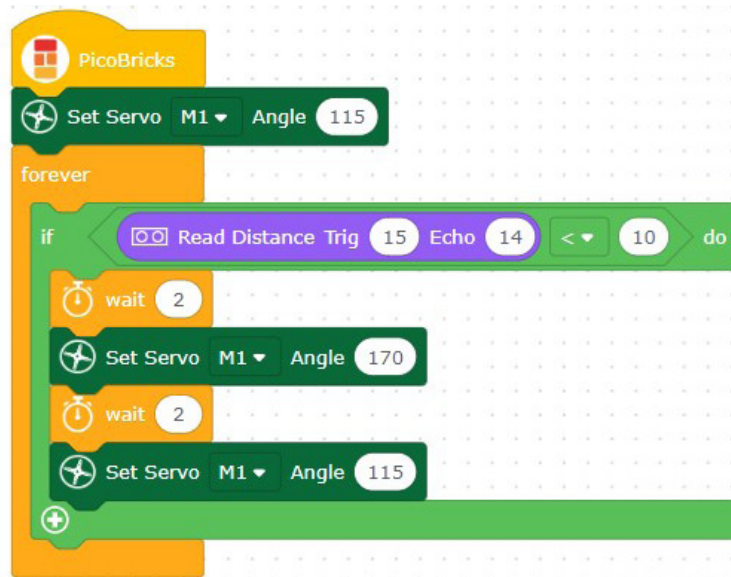


Proje başladığında, sürekli çalışan koşul yapısını belirleyelim.

- 2 saniye bekleyen
- 1. servo motorun derecesini 170
- eğer mesafe sensörünün değeri 10'dan az ise 2 saniye bekledikten sonra servo motoru ilk pozisyona getiren kod bloklarını oluşturalım.



Kod blokları hazır!



Digital Ruler

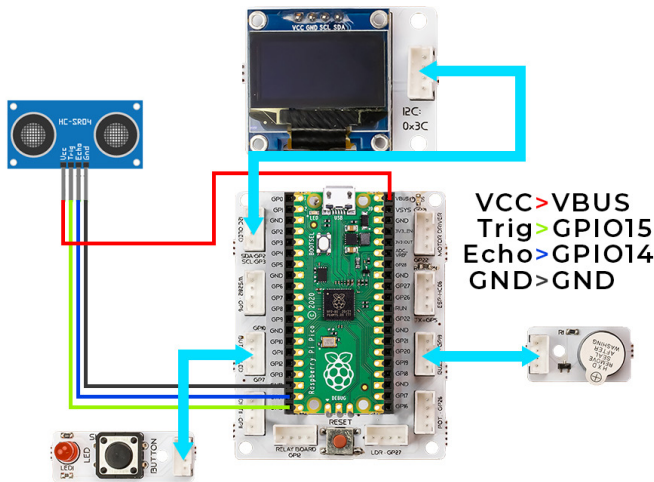
Uzunluk ölçmek için birçok araç kullanılmaktadır. Bu araçların başında cetveller gelmektedir. Ölçeğimiz yere ve büyüklüğüne göre ölçü aletimiz farklılaşmaktadır. İnşaat ve mimaride şerit metreler , küçük ve milimetrik hassasiyet gerektiren nesnelere için kumpaslar kullanılmaktadır. Ayrıca hem büyük hem de hassas ölçüm yapılması gereken bir alan ölçülmek isteniyorsa lazer ve kızılötesi sistemler ile çalışan mesafe ölçerler kullanılmaktadır. Sağlık sektöründe kullanılan ultrasonografi cihazları da benzer mantıkla çalışmakta ancak ölçümlerini görsellere dönüştürmektedir

Projemizde PicoBricks ile ultrasonik sensör kullanarak mesafe değerini butona basıldığında OLED ekranda göstereceğimiz dijital bir cetvel hazırlayacağız. Ultrasonik sensörler yaydıkları ses dalgalarının geri dönüş sürelerine göre mesafe tespiti yaparlar.

2.21.1. Proje Detayları ve Algoritma

Picobricks başladığında OLED ekranda yönerge görüntülenir. Kullanıcı butona bastıktan sonra 1 saniye boyunca 50'şer milisaniye aralıklarla 20 ölçüm yapılır ve ortalaması alınır. Ölçüm sırasında kırmızı LED açık kalır ölçüm tamamlandığında kırmızı LED söndürülür. Ortalama değere sensörün ucu ile kutunun arkasındaki mesafe eklenir. Son mesafe değeri OLED ekranda gösterilir.

2.21.2. Bağlantı Şeması



2.21.3. Projenin PicoBricks IDE ile Kodlanması

İlk olarak, kod için gerekli olan fonksiyonları oluşturarak projeye başlayalım.

- “start” adında bir fonksiyon tanımlayarak, proje başladığında ekranda yazacak olan ifadeleri belirleyelim.

```

to start
  Write Text to Screen X 5 Y 0 "<DIGITAL RULER>"
  Write Text to Screen X 0 Y 25 "Press the BUTTON"
  Write Text to Screen X 25 Y 35 "to start"
  Write Text to Screen X 25 Y 45 "measuring"
  Show Screen Buffer

```

- “finish” adında bir fonksiyon tanımlayarak ölçüm yapıldıktan sonra gereken işlemleri belirleyelim.

```

to finish
  Clear Screen Buffer
  Write Text to Screen X 5 Y 0 "<DIGITAL RULER>"
  Write Text to Screen X 0 Y 25 "Distance: "
  Write Text to Screen X 70 Y 25 measure
  Write Text to Screen X 100 Y 25 "cm"
  Show Screen Buffer
  Play Buzzer Freq 300
  Stop Buzzer Interval 0.5

```

- Döngü ve koşul yapılarını kullanarak yaptığımız fonksiyonları çağıralım. Butona her basıldığında ölçümü yap ve ölçümün değerini OLED ekrana yazdır.

```

PicoBricks
start
forever
  if Read Button = 1 do
    Set Led on
    Play Buzzer Freq 300
    Stop Buzzer Interval 0.25
    set measure to round Read Distance Trig 15 Echo 14
    Serial Print measure
    wait 0.2
    Set Led off
  finish

```


2.22. Air Piano

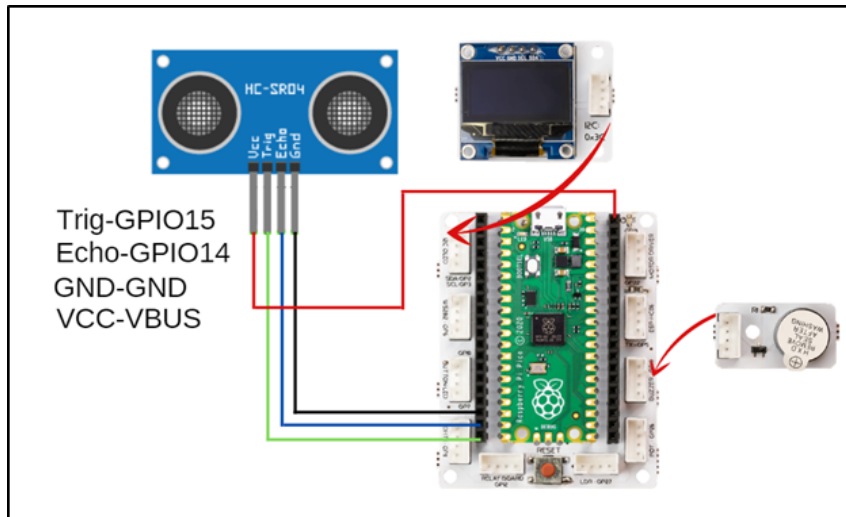
Elektronik teknolojisinin geliřimi ile üretimi zor, pahalı, yüksek kaliteli ses üreten müzik aletleri dijitalleşmiştir. Piyanolar bu enstrümanların başında gelmektedir. Dijital piyanoların her bir tuşu farklı frekansta elektrik sinyalleri üretir. Böylelikle hoparlörlerinden 88 farklı notayı çalabilmektedir. Dijital enstrümanların tuşlarının gecikme süresi, hoparlörün kalitesi, sesin çözünürlüğü gibi faktörler kaliteyi etkileyen faktörler olarak ortaya çıkmıştır. Elektro gitarlarda tuşlar yerine tellerdeki titreşimler dijitalleştirilir. Üflelemeli enstrümanlarda ise ses çıkışına takılan yüksek çözünürlüklü mikrofonlar sayesinde çalınan notalar elektrik sinyallerine dönüştürülüp kayıt edilebilmektedir. Elektronik teknolojisindeki bu gelişim yüksek maliyetli müzik aletlerine ulaşım kolaylaştırmış, müzik eğitimi daha geniş çeşitliliğe kavuşmuş ve daha geniş kitleye yayılmıştır.

Bu projede PicoBricks ile 8 nota çalabilen basit bir piyano yapacağız. Bu piyanonun hoparlörü buzzer'ın olacak. Piyanonun tuşlarının görevini ise ultrasonik sensör üstlenecek.

2.22.1. Proje Detayları ve Algoritma

Bu projede HC-SR04 Ultrasonik mesafe sensörü ve PicoBricks üzerindeki buzzer modülünü kullanarak piyano uygulaması yapacağız. Mesafe sensöründen gelen değerlere göre buzzer ın farklı notalar çalmasını sağlayacak ve sensöre elimizi yaklaştırıp uzaklaştırarak melodiler oluşturacağız. Ayrıca OLED ekrana anlık olarak mesafe çalınan nota bilgilerini yazdıracağız

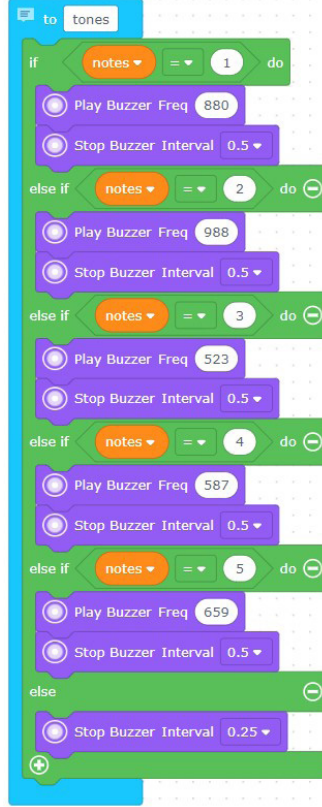
2.22.2. Bağlantı Şeması



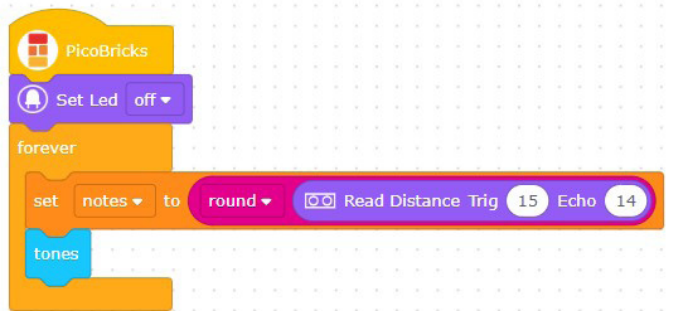
2.22.3. Projenin PicoBricks IDE ile Kodlanması

İlk olarak, gerekli fonksiyonları oluşturarak projeye başlayalım.

- “tones” adında bir fonksiyon oluşturarak piyano projesinde olacak notaları belirleyelim.



- “Forever” döngü bloğu kullanılarak mesafe değerine göre buzzer ile notalar çaldırılalım.





2.23. Maze Solver Robot

Kodlama eğitimi programlama dillerinin tarihi kadar eskidir. Günümüzde kodlama eğitimini yaygınlaştırmak, heyecanlı ve eğlenceli hale getirmek için farklı ürünler kullanılmaktadır. Bunların başında eğitsel robotlar gelmektedir. Robotları hazırlamak ve kodlamak çocukların mühendislik ve kodlama becerilerini geliştirmektedir. Kodlama eğitimini yaygınlaştırmak, öğretmen ve öğrencileri teşvik etmek için kurum ve kuruluşlar tarafından robotik yarışmalar düzenlenmektedir. Bu yarışmalardan biri de Labirent Çözen Robot yarışmalarıdır. Bu robotlar önce labirentte dolanarak varış noktasını öğrenir ve başlangıç noktasına geri dönerler. Daha sonra labirente tekrar başladıklarında en kısa yoldan en hızlı şekilde varış noktasına ulaşmaya çalışırlar. Robotlar labirenti öğrenirken mesafe sensörlerinden faydalanırlar. Kızılötesi ya da ultrasonik sensörler bu robotlarda görev almaktadır.

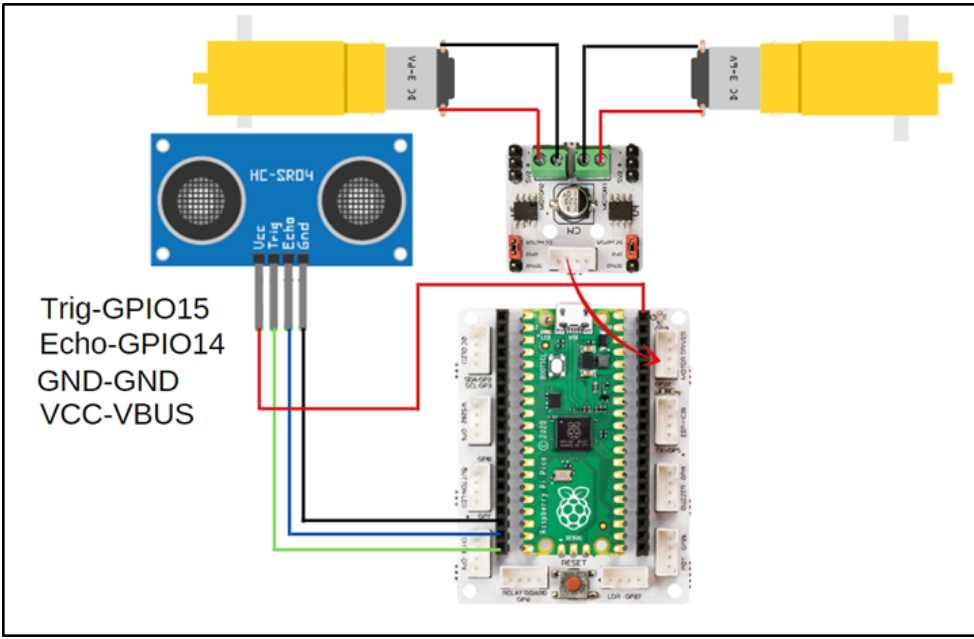
Ev ve işyerlerinde kullanılan akıllı robot süpürgeler de labirent çözen robotların algoritmalarına yakın mantıkla çalışmaktadır. Engelleri sürekli kontrol edip haritalayan algoritmaları sayesinde süpürme işini eksiksiz ve çarpmadan yapmaya çalışırlar. Akıllı süpürgelerin çoğunda mesafe ölçme ve engel algılamak için lazer ile yüksek hassasiyetli ölçüm yapan LİDAR ve kızılötesi sensörler görev almaktadır.

Bu projede PicoBricks ile labirent çözen robot yarışmalarına hazırlanabileceğiniz basit bir robot yapacağız.

2.23.1. Proje Detayları ve Algoritma

Labirent çözen robot projesinde setin içerisinden çıkan 2WD robot araba kitini kullanacağız. Robotun önündeki mesafeyi algılayarak hareketlerine kendi kendine karar verebilmesi için HC-SR04 ultrasonik mesafe sensörü kullanacağız. Labirent içerisinde robot araba önündeki mesafeyi algılayarak önü boş ise ilerleyecek. Eğer mesafe 5 cm den küçük ise araba sağa dönecek, tekrar mesafeyi ölçecek eğer sağ taraftaki mesafe 5 cm den büyük ise ilerleyerek yoluna devam edecek, küçükse sola dönerek ilerleyecek. Bu şekilde sağa ve sola dönerek labirent içerisinde boş olan yollardan aracın ilerlemesini ve labirentten çıkmasını sağlayacağız.

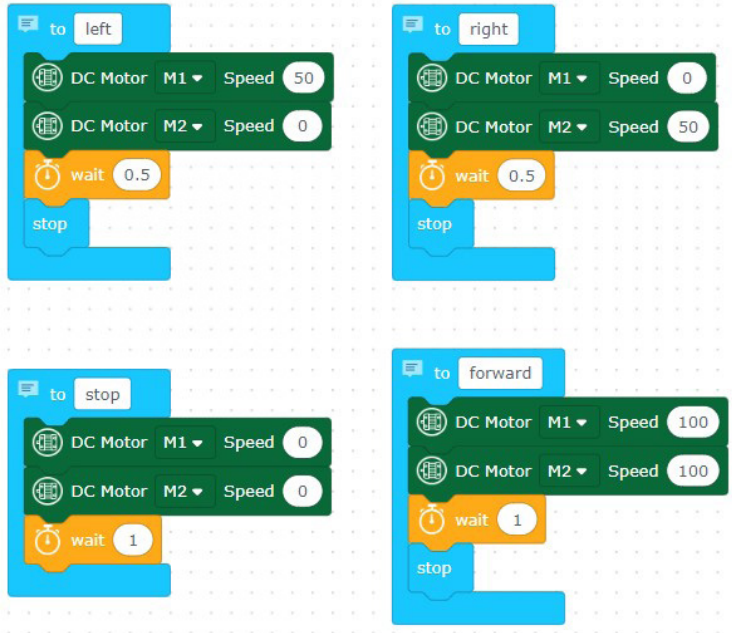
2.23.2. Bağlantı Şeması



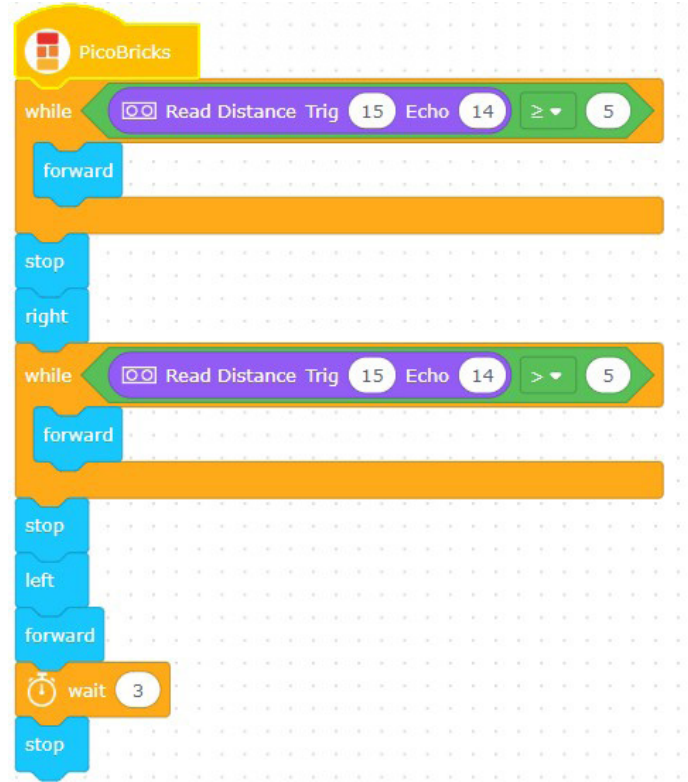
2.23.3. Projenin PicoBricks IDE ile Kodlanması

Gerekli fonksiyonları oluşturarak projeye başlayalım.

- “left”, “right”, “forward” ve “stop” adında dört adet fonksiyon tanımlayarak robotun hareketlerini belirleyelim.



- Labirentin rotasına göre belirlenen döngü ve koşul yapılarının içinde belirlediğimiz fonksiyonları çalıştıralım.





2.24. Smart Greenhouse

Küresel ısınmanın etkisiyle iklimlerde meydana gelen hızlı değişimler tarım alanındaki faaliyetlerde verimin düşmesine sebep olmaktadır. 1500'li yıllarda Daniel Barbaro tarihte bilinen ilk serayı yaptı. Seralar kontrol edilebilen hava, su, ısı ve ışık koşulları sunabilen bitki yetiştirmeye elverişli ortamlardır. Seralarda ısıyı dengelemek için ısıtıcılar , sulama yapmak için elektrikli su motorları, nem oranını ayarlamak ve tozlaşmayı sağlayabilmek için fanlar kullanılmaktadır. Teknolojinin de gelişmesiyle üretici seranın durumunu istediği yerden telefonuyla takip edebilmekte ve yapılması gereken işleri yapabilmektedir. Bu teknolojinin genel adı Nesnelerin İnterneti (IOT) dir.

Seralarda sıcaklık, nem ve oksijen miktarını ölçmek için özel sensörler kullanılır. Ayrıca sulamaya karar vermek için toprak nemini ölçen özel sensörler kullanılmaktadır. Sulama verimliliğini artırmak için ise elektronik kontrollü damla sulama sistemleri kullanılmaktadır.

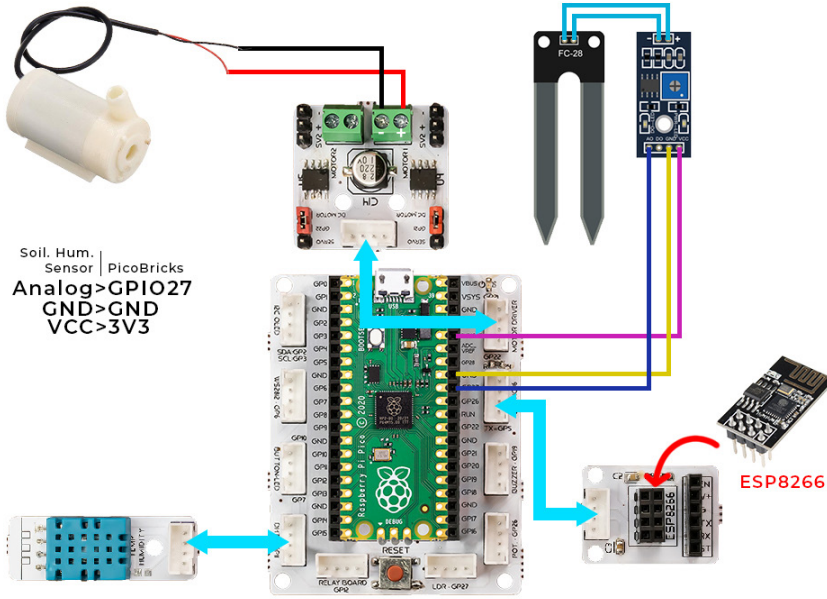
Bu projede PicoBricks ile IOT teknolojisini barındıran basit bir sera hazırlayacağız. PicoBricks'i ESP8266 wifi modülü ile bu serada kullanacağız. Bu sayede serayı internet üzerinden takip edebildiğimiz bir nesne haline getireceğiz.

2.25.1. Proje Detayları ve Algoritma

Hazırlayacağın sera maketinin içinde toprak nemi sensörü, üstten asılı olarak DHT11 sıcaklık ve nem sensörü bulunacaktır. Maketin dışındaki su tankının içine dalgıç pompa konulacak, pompanın ucundan çıkan hortum ise seradaki toprağa gidecektir. Picoboard sera maketinin dışında uygun bir yere yerleştirilecektir.

Picobricks başladığında ESP 8266 wifi modülü sayesinde wifi yayını yapmaya başlar. Aynı ağa bağlı akıllı telefondan Esp8266'nın IP adresini girdiğimizde Serayı kontrol edeceğimiz web sayfası ile karşılaşırız. Burada sıcaklık ve nem değerlerini görebiliriz. Dilersek sulama komutu vererek sulama işlemini başlatabiliriz.

2.24.2. Bağlantı Şeması



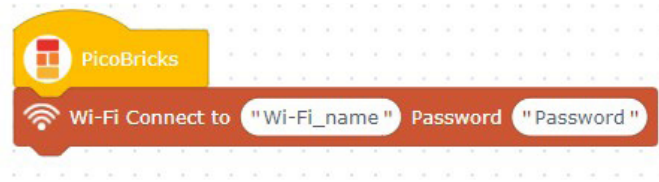
2.24.3. Projenin PicoBricks IDE ile Kodlanması

Öncelikle, gerekli fonksiyonları oluşturarak projeye başlayalım.

- Projedeki PicoBricks modüllerinin değerlerini yazdıran "controlPanel" adında bir fonksiyon oluşturalım.



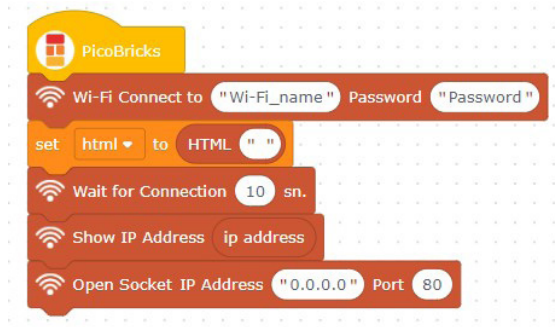
- PicoBricks IDE'deki WiFi blokları kullanılarak projeyi oluşturmaya devam edelim. WiFi ağı üzerinden Raspberry Pi Pico ile haberleşmeyi sağlamak için WiFi ağınızın şifresini ve kullanıcı adını yazın.



- WiFi bloklarını kullanarak, IP adresi üzerinden haberleşmeye sürecine devam edelim. Bu adımda, PicoBricks IDE'de bir HTML sayfası oluşturarak aşağıdaki HTML kodunu HTML bloğuna yapıştıralım.

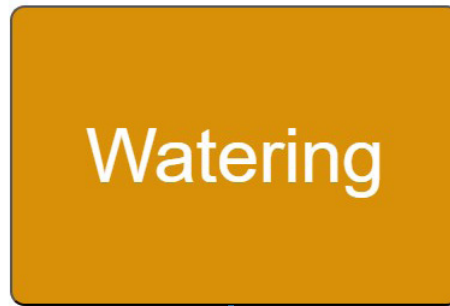
HTML Kodu:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" href="data:,">
    <title>Smart Green House</title>
  </head>
  <style>
    body{margin:15%;}
    h1{color:#7EA03E}
    .buttonw{width:300px; height:200px; background-color:#D89008; font-size:50px; color:white; border-radius:10px;cursor: pointer;}
  </style>
  <body>
    <center>
      <h1>Smart Green House Watering Panel</h1>
      <form><button class="buttonw" name="watering" value="on" type="submit">Watering</button></form>
    </center>
  </body>
</html>
```



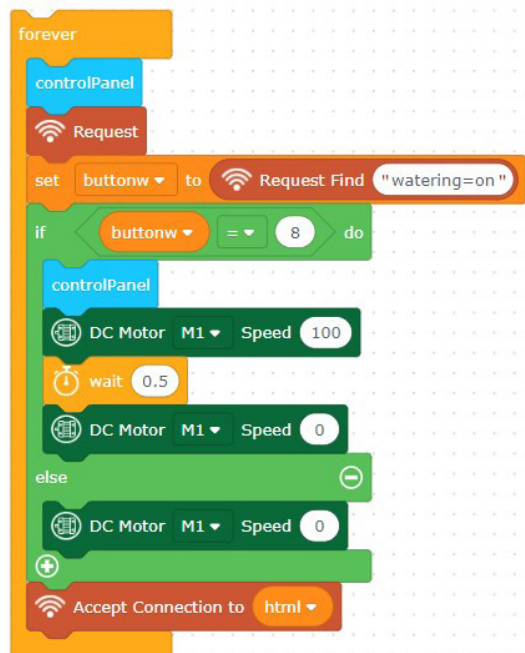
Verilen HTML kodunu bu alana yapıştırın.

Smart Green House Watering Panel



Butona tıklayarak seranızı sulayabilirsiniz.

- Koşul ve döngü yapılarında gerekli süreçleri yaparak projeyi tamamlayalım. Bu adımda, WEB sayfasındaki butona tıklayarak serayı sulamayı sağlayan kod bloklarını oluşturalım.



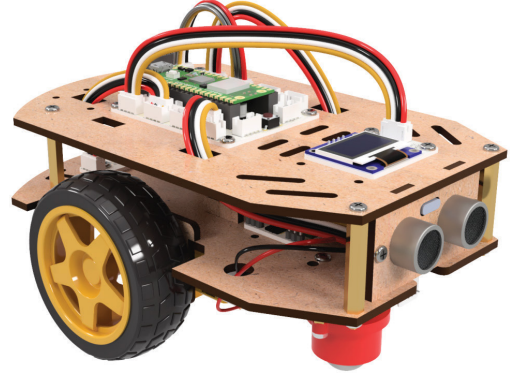
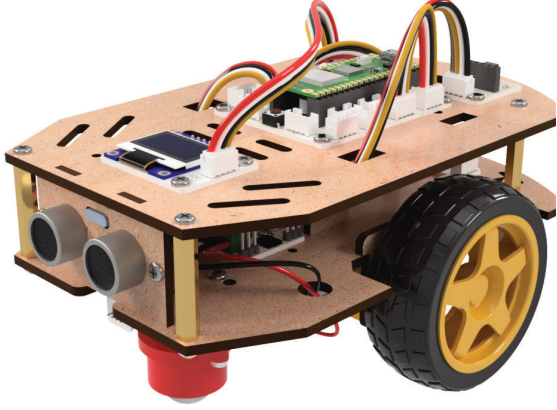
Proje tamamlandı!

```

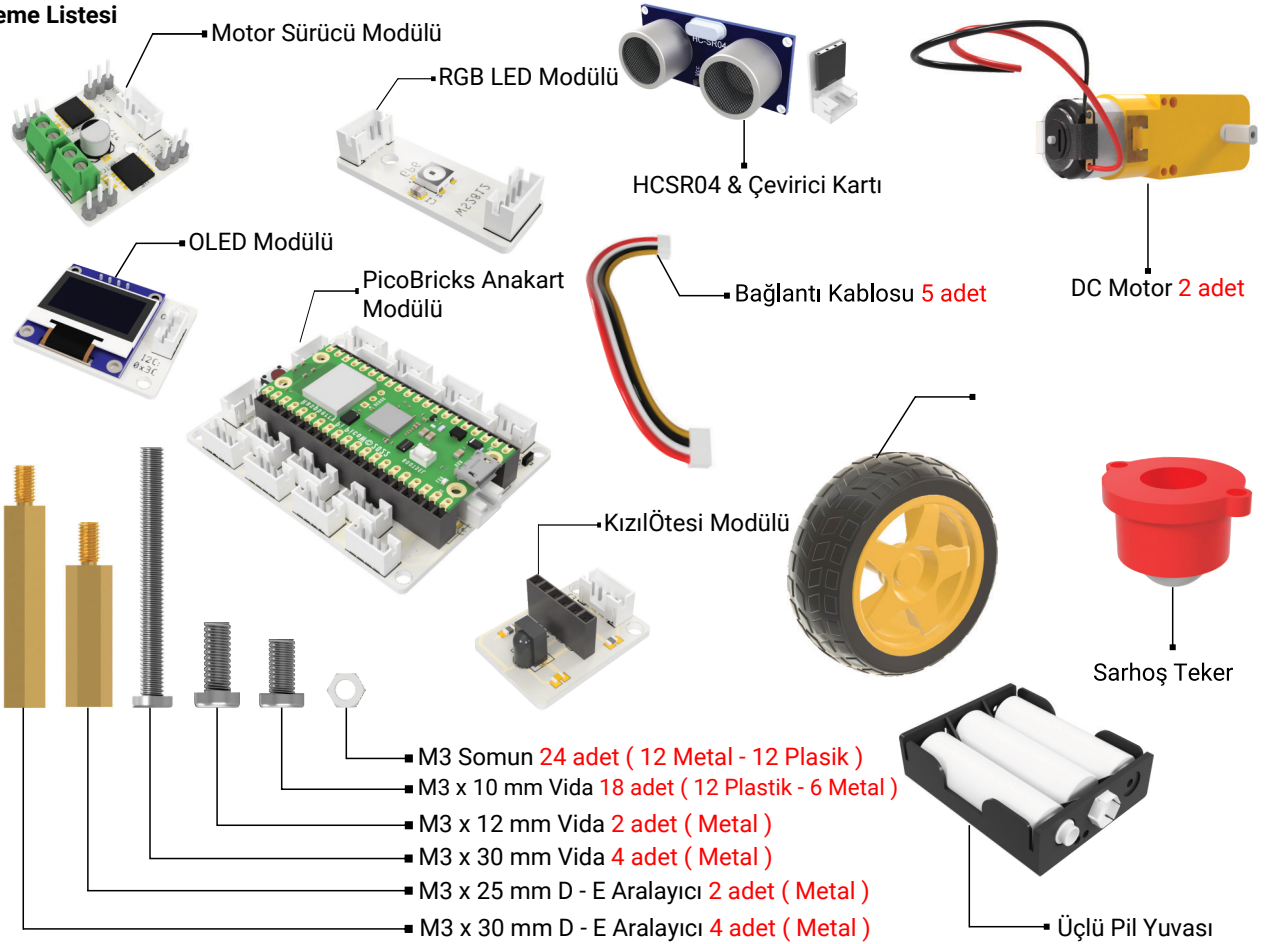
PicoBricks
Wi-Fi Connect to "MY_SSID" Password "MYpassword"
set html to HTML ""
Wait for Connection 10 sn.
Show IP Address ip address
Open Socket IP Address "0.0.0.0" Port 80
forever
  controlPanel
  Request
  set buttonw to Request Find "watering=on"
  if buttonw = 8 do
    controlPanel
    DC Motor M1 Speed 100
    wait 0.5
    DC Motor M1 Speed 0
  else
    DC Motor M1 Speed 0
  Accept Connection to html
  to controlPanel
    set smo to round Read Analog Pin 26 x 100 ÷ 65535
    Clear Screen Buffer
    Write Text to Screen X 10 Y 0 "Control Panel"
    Write Text to Screen X 40 Y 10 "Smart "
    Write Text to Screen X 10 Y 20 "Green House "
    Write Text to Screen X 0 Y 40 "SMO:"
    Write Text to Screen X 65 Y 40 smo
    Write Text to Screen X 0 Y 50 "TEMP:"
    Write Text to Screen X 65 Y 50 Read Temperature (°C)
    Show Screen Buffer
  
```


3. KAYNAKLAR

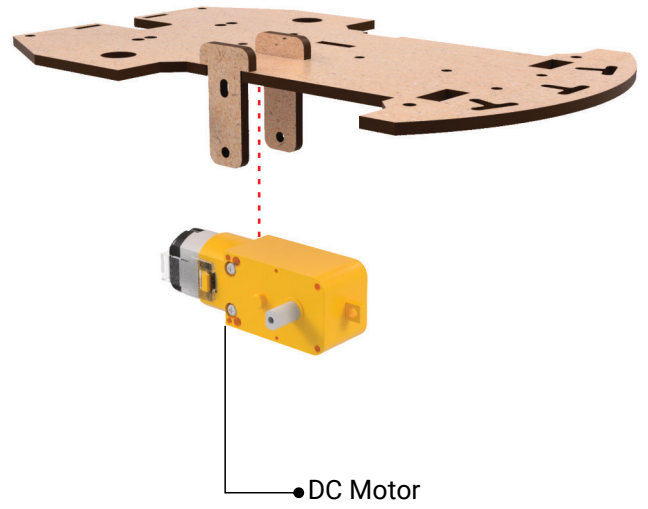
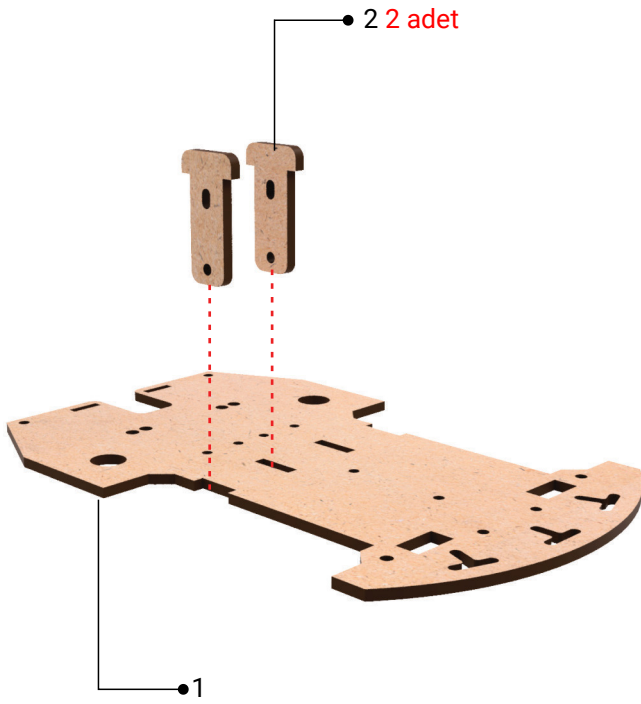
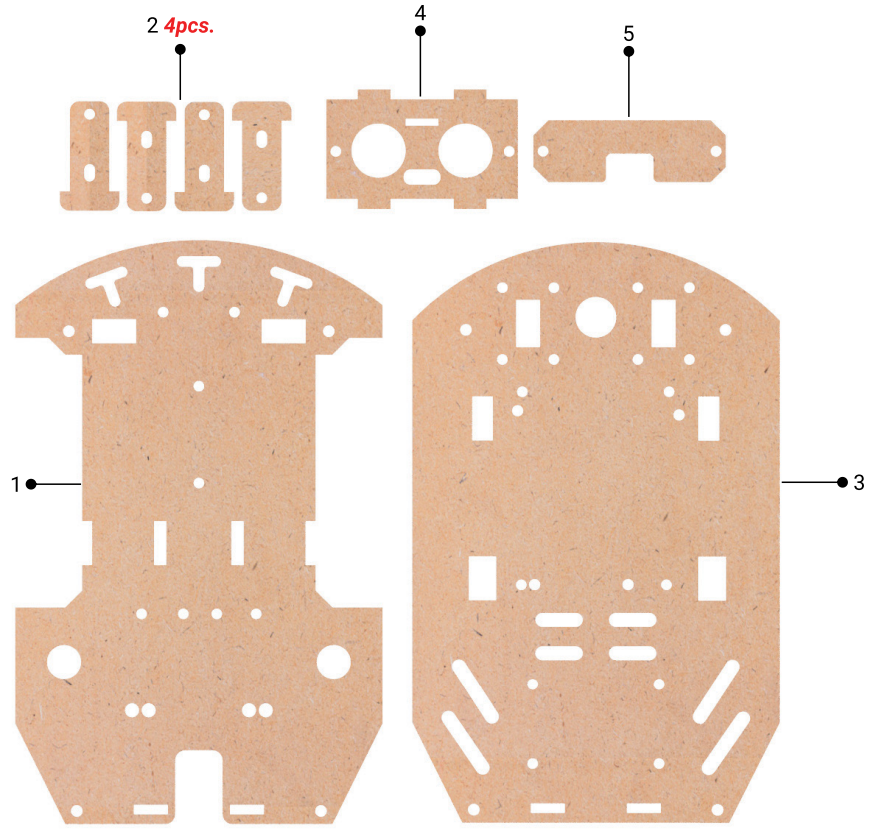
PicoBricks Robot Araba Kurulum Kılavuzu

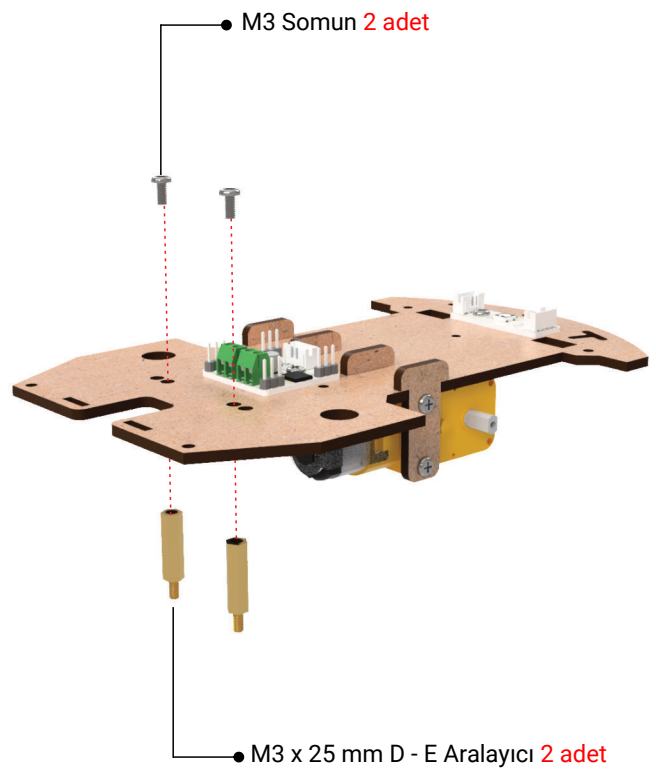
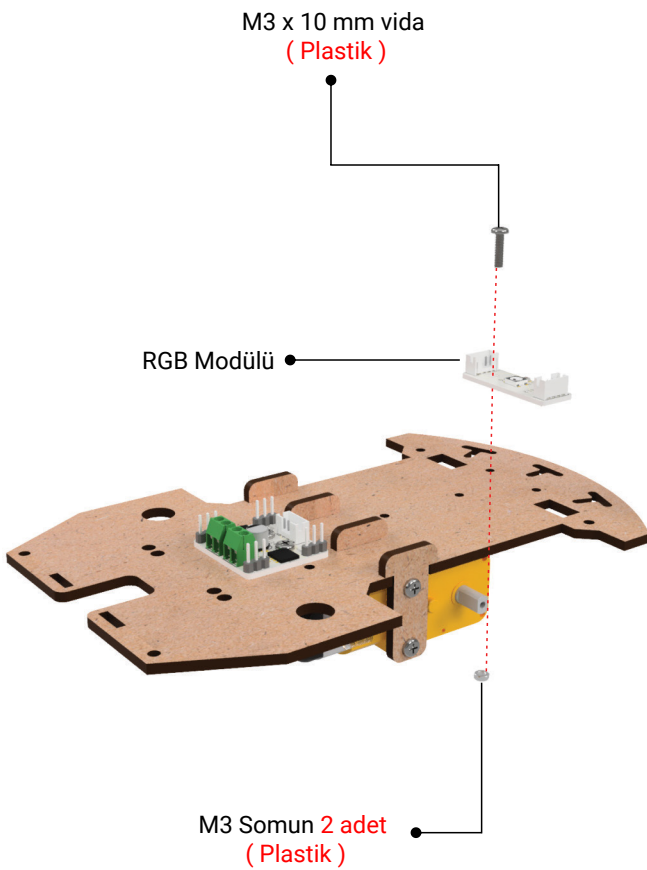
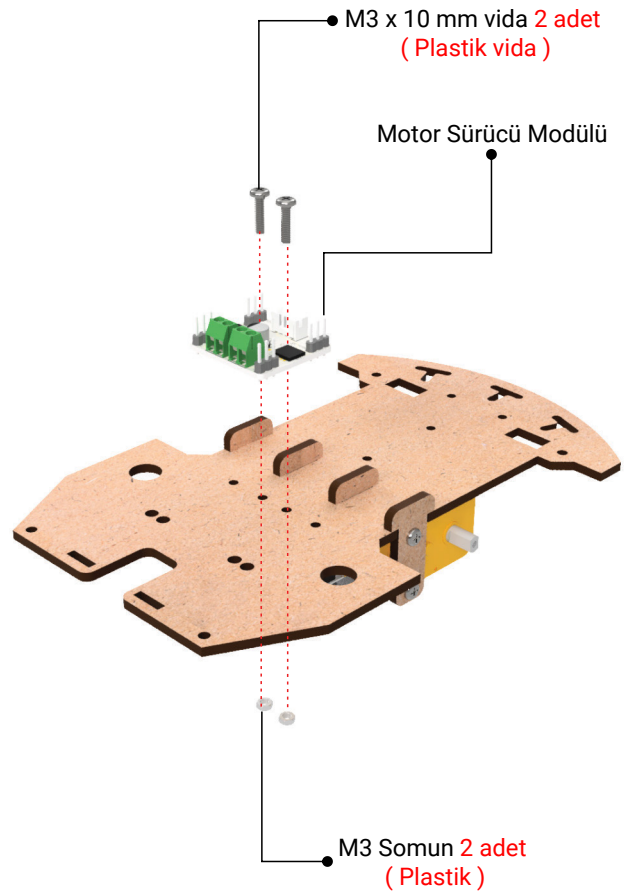
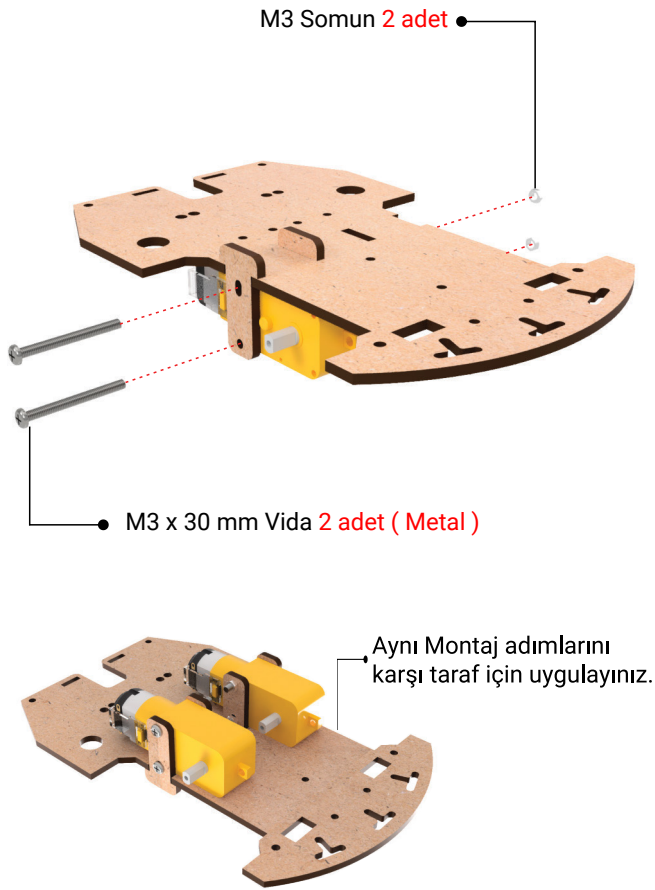


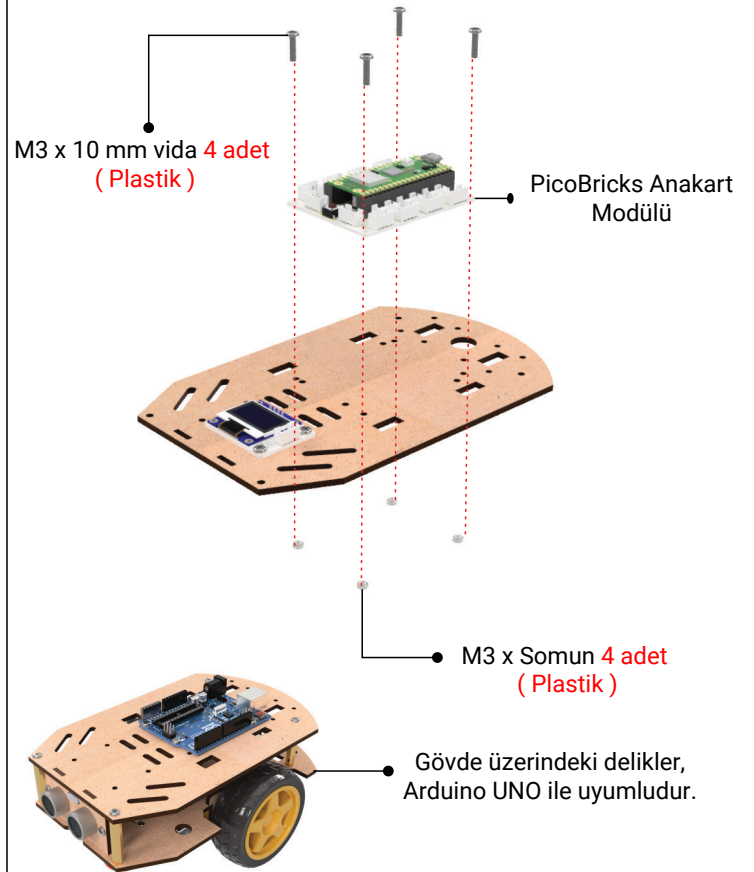
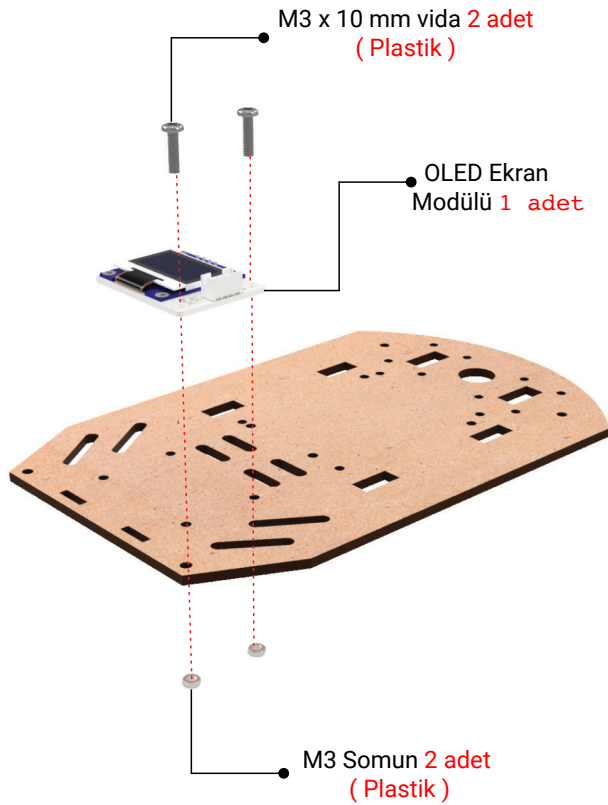
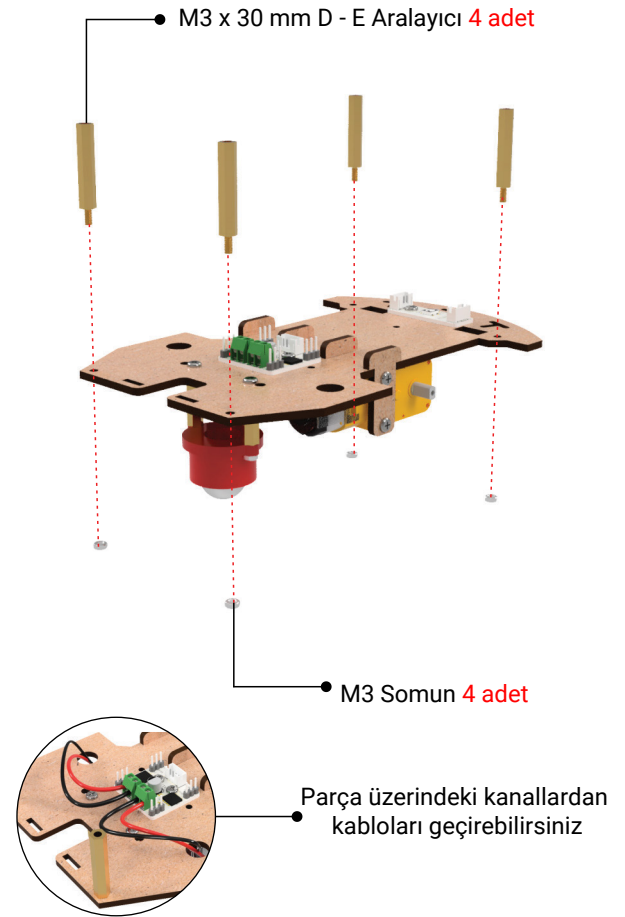
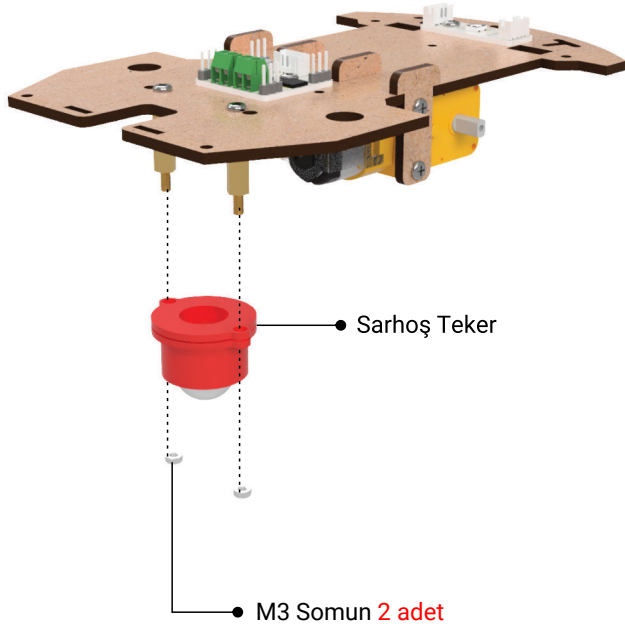
Malzeme Listesi

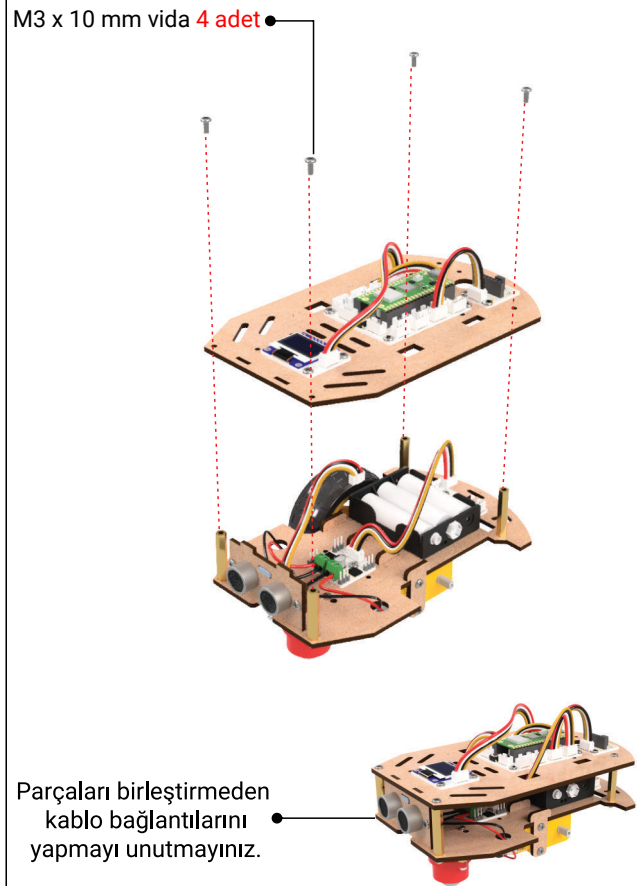
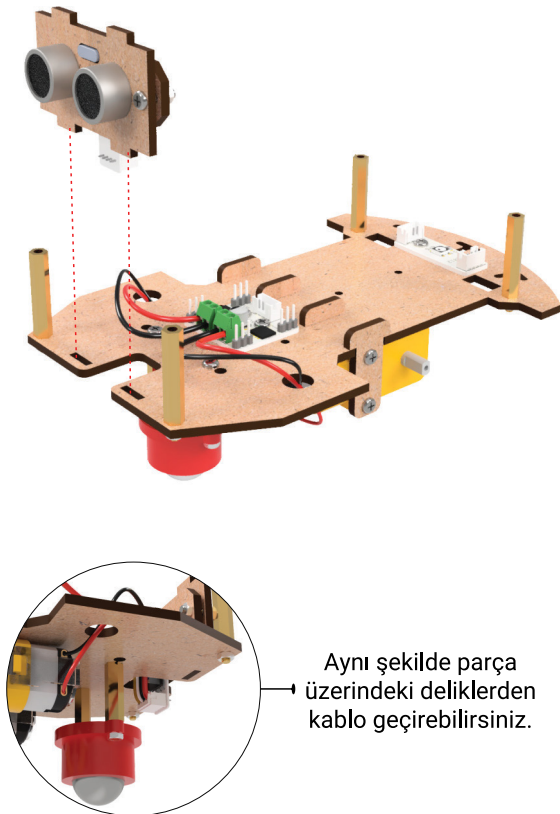
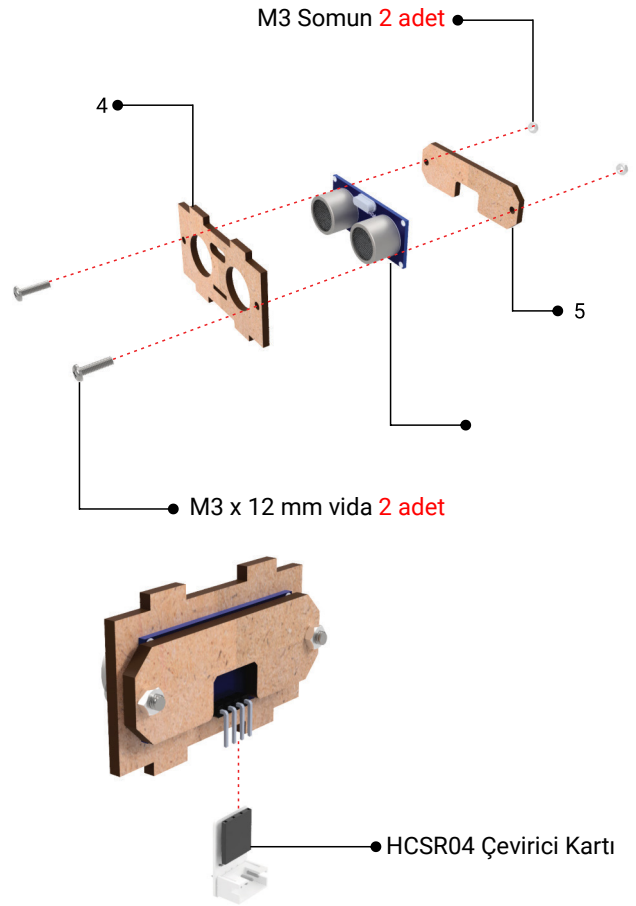
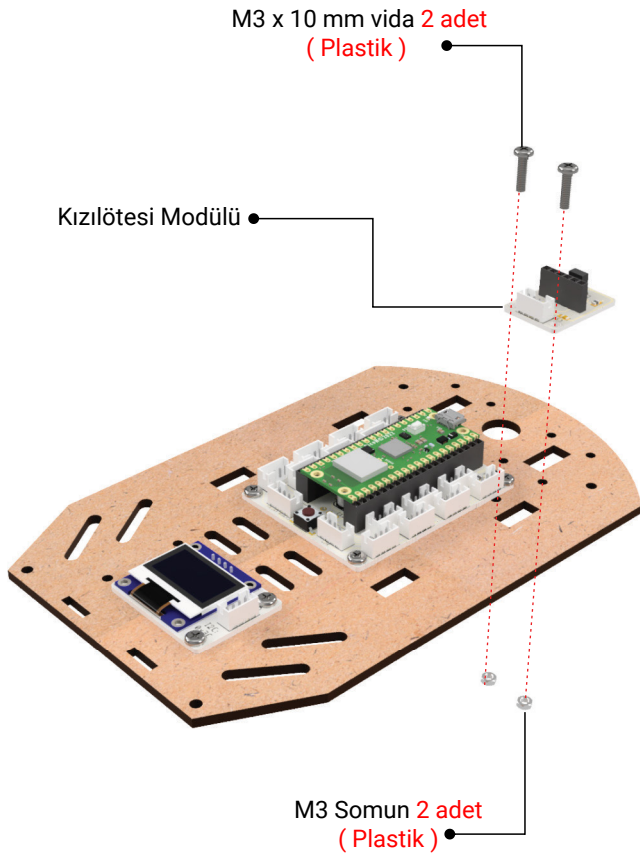


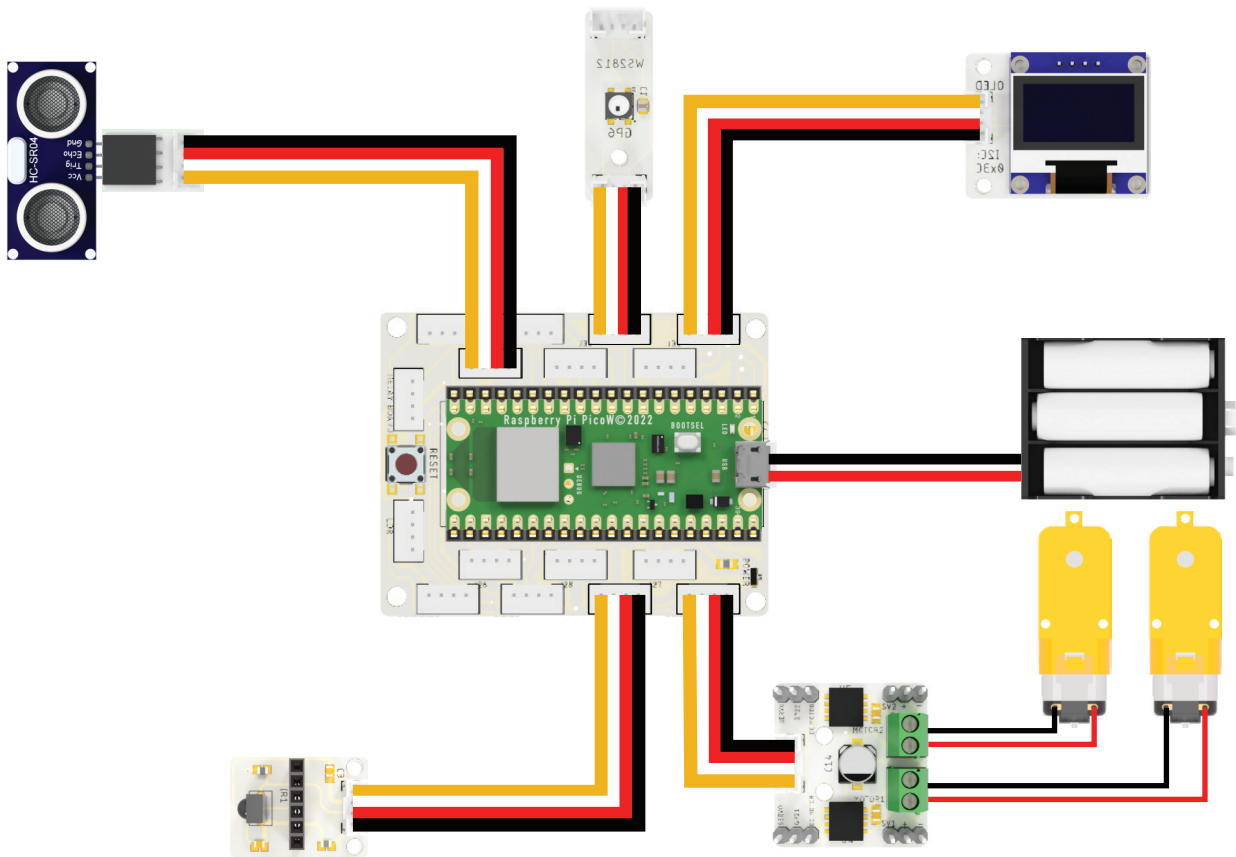
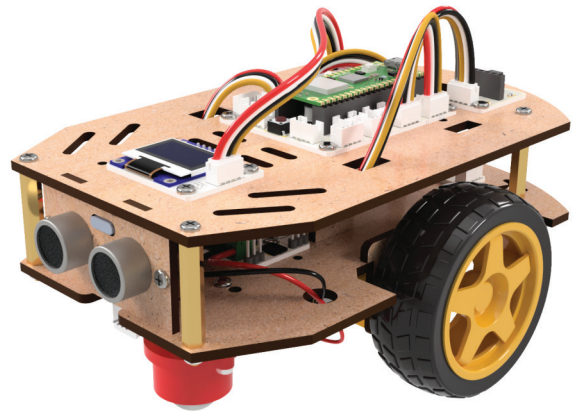
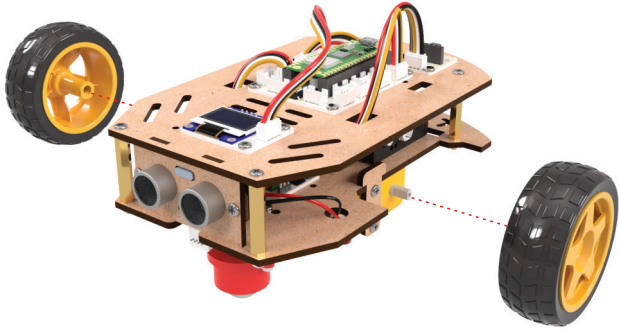
Ahşap Parçalar





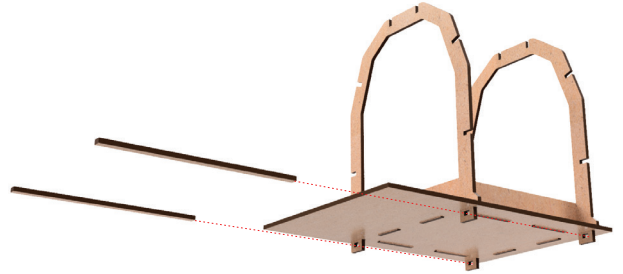
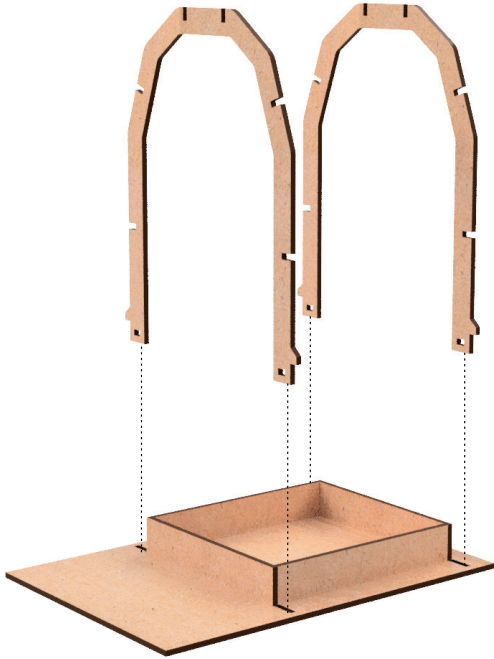
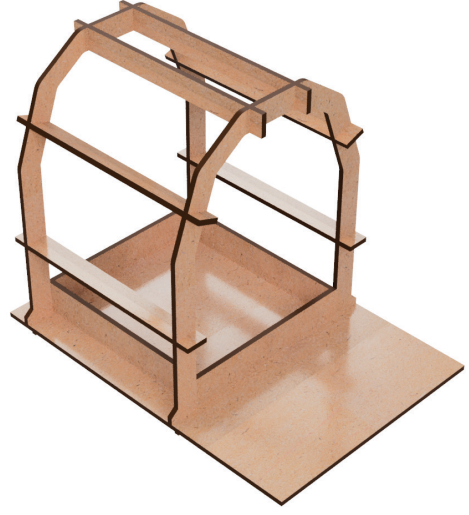


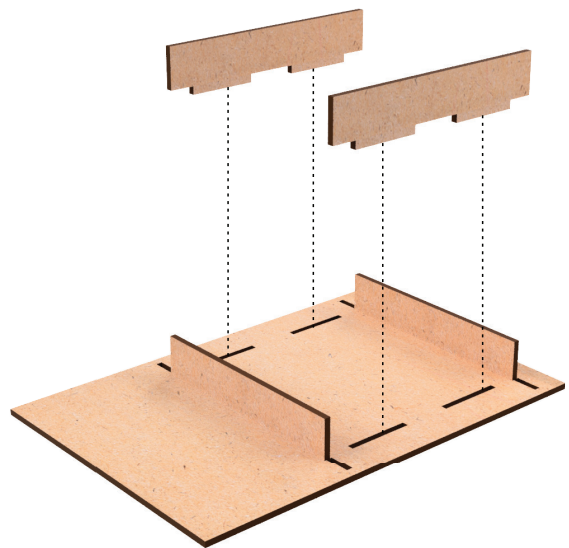
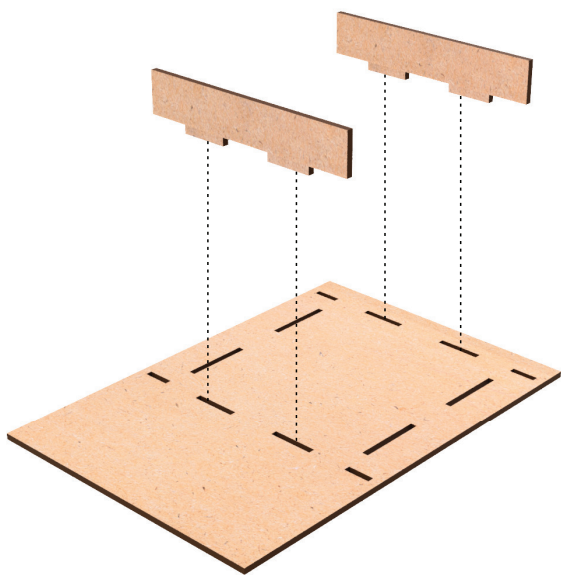
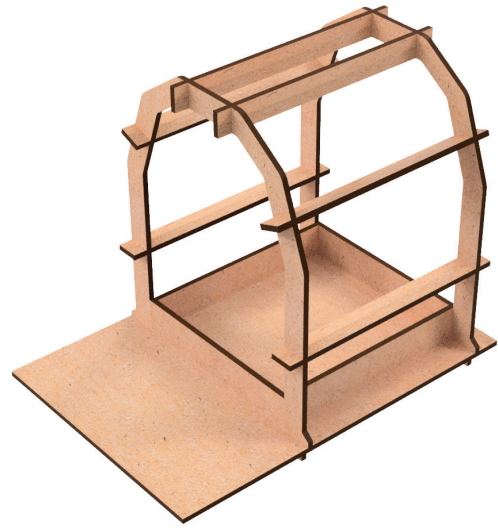
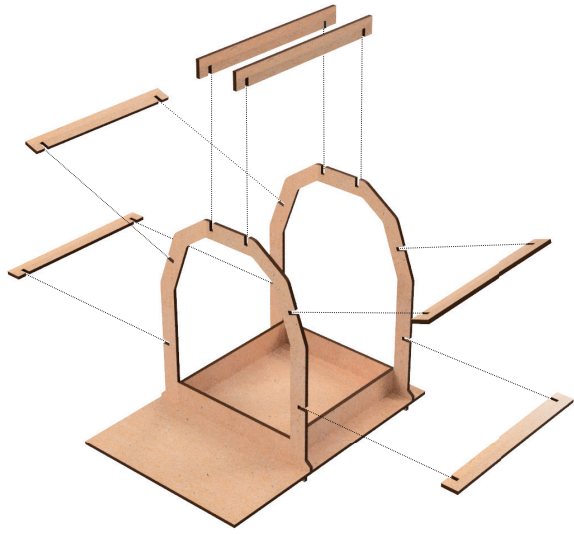


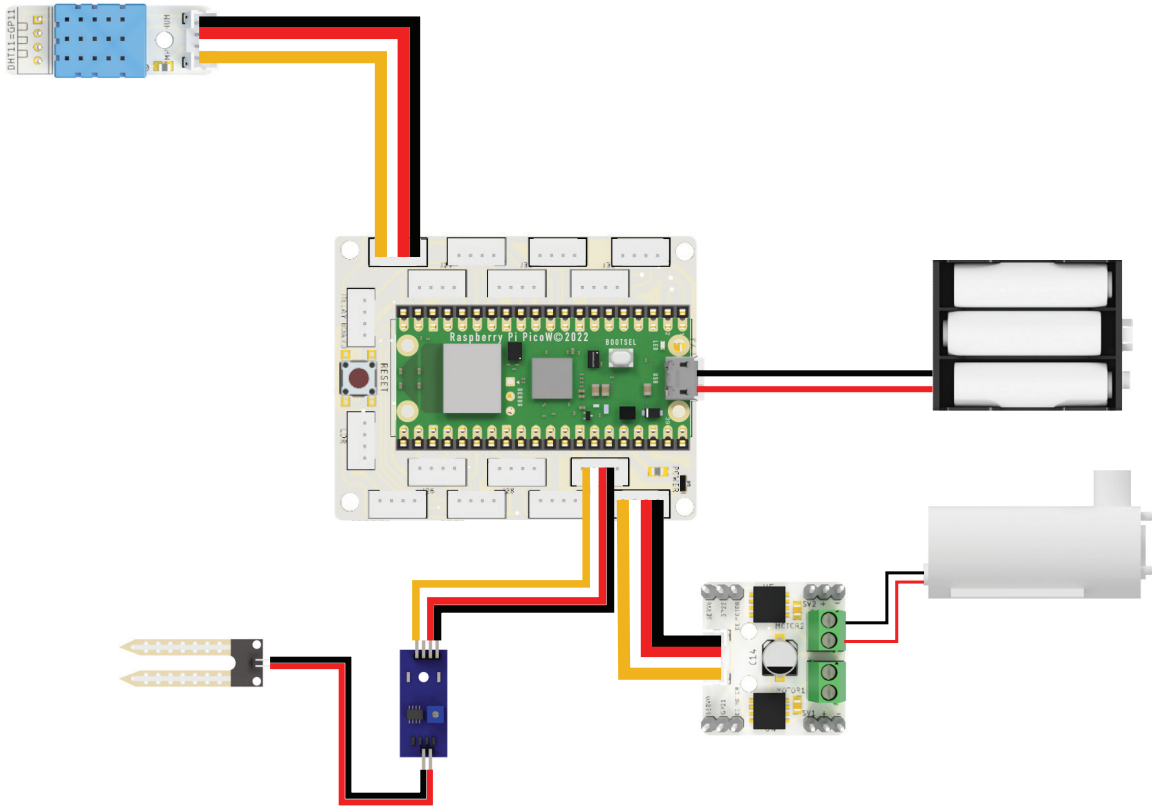




PicoBricks Akıllı Sera Kurulum Kılavuzu









3 Boyutlu Modeller

Two Axis Robot Arm Project 3D Parts

Piggy Bank Project 3D Parts

Maze Solving Robot Project 3D Parts

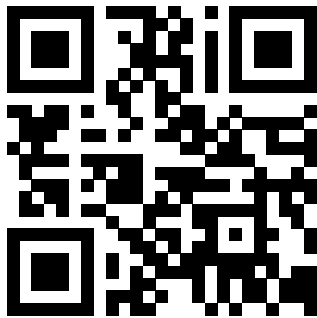
Greenhouse Control Android App(.apk)

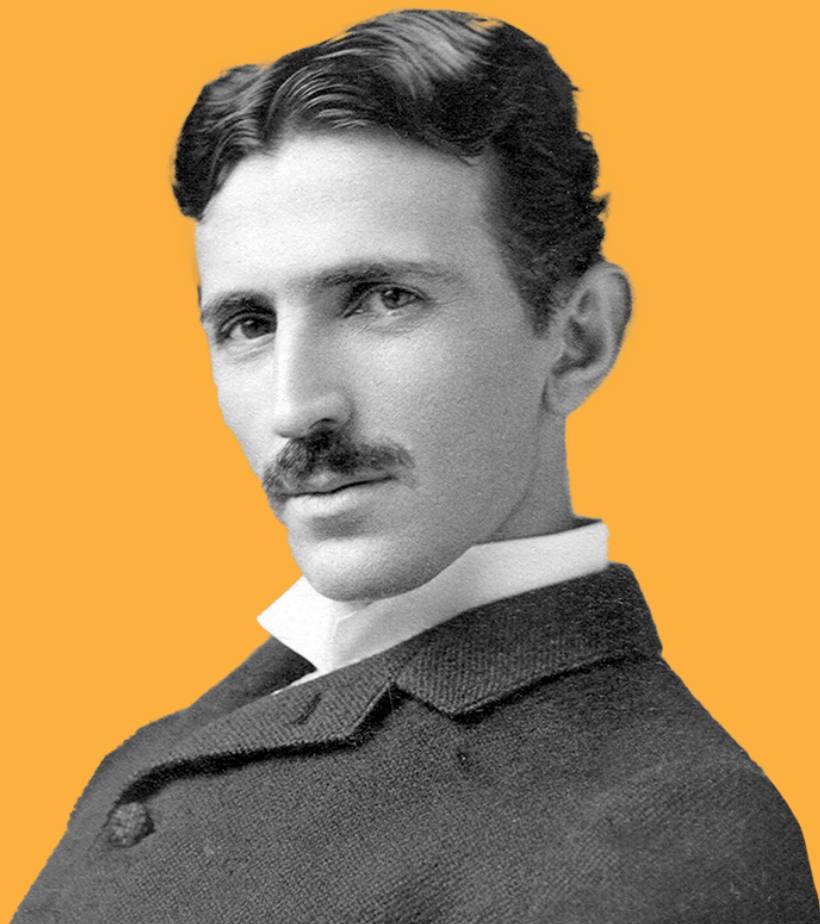
Sera Control MITAppInventor 2 Project File

Voice Controlled Robot Car Project Android App (.apk)

Piggy Bank 1: <https://www.thingiverse.com/thing:2824451>

Piggy Bank 2: https://drive.google.com/file/d/1_gjo-hEDvhdewtPoorDOv1YhhsNd74ZJ/view?usp=sharing





"If you want to find the secrets of the universe, think in terms of energy, frequency and vibration."

Nikola Tesla



Join the community



community.robotistan.com

PicoBricks GitHub



github.com/Robotistan/PicoBricks

picobricks.com