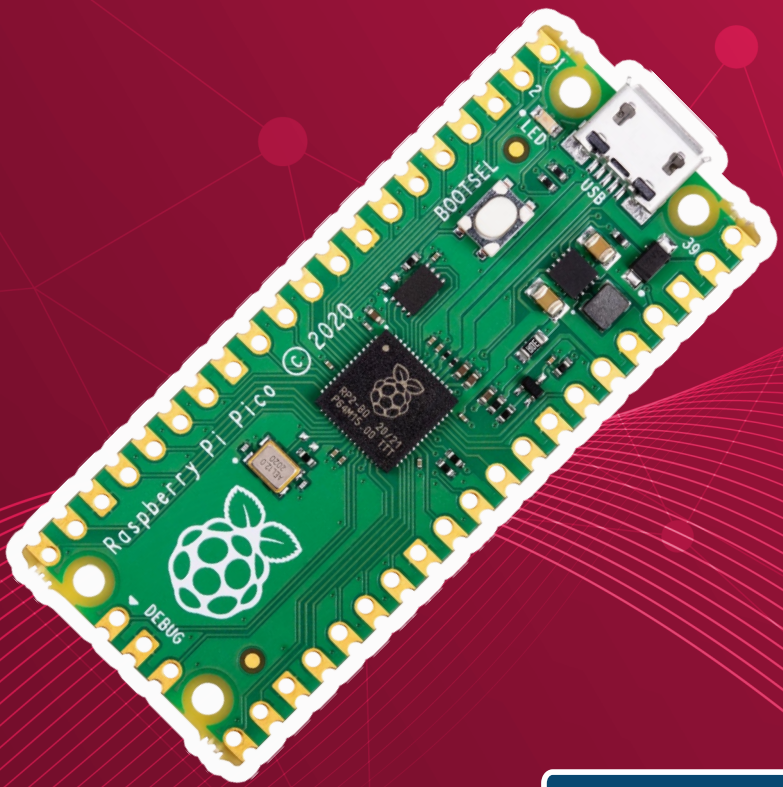
Raspberry Pi Pico Uygulama Kitabı



robotistan

Elektronik ve Kodlama dünyasına hoşgeldiniz. Bu kitabı açtığınıza göre siz de merak denizinde yüzüp, yeni şeyler öğrenmeye heveslisiniz demektir. Bu tür konularda yeni şeyler öğrenmek zor gibi düşünülse de adım adım ve doğru uygulamalar ile ilerlerseniz çok basit olduğunu fark edeceksiniz. İlk aşamalarda uygulamaları yaptıkça oturmayan anlamsız gelen yerler olacaktır. Bu sorunu uygulama yaptıkça aşacaksınız. Sadece biraz sabır gerekli...Kolay ve doğru yol haritası ile Arduino programlamayı öğrenebilmeniz için uygulamalar kolaydan başlayarak, daha komplekse doğru ilerlemektedir.

Uygulamaların daha detaylı videolu anlatımlarını izlemek isterseniz kitabın arka kısmındaki QR kodu taratarak YouTube kanalımıza gidebilirsiniz. Uygulamalara dijital ortamda erişmek isterseniz http://maker.robotistan.com blog sayfamızda da bulunmaktadır. Kitapçık içerisinde yazılan kodlara hem ilgili videoların açıklama kısmından hem de blog sayfamızdan ulaşabilirsiniz.

Bu kitap Robotistan Elektronik A.Ş bünyesinde yazılmıştır. Yazılış amacı ise Arduino'ya kolay ve doğru yoldan başlamak isteyenlere rehber olmasıdır. Umudumuz bu içeriklerin herkese faydalı olması ve sizlerin öğrenme sürecini kolaylaştırıp hızlı şekilde proje yapmanızı sağlamaktır.

Set içerikleri, uygulamalar, videolarımız ve aklınıza takılan ter türlü öneri ve sorularınız için info@robotistan.com e-mail adresinden bize iletebilirsiniz.

Robotistan Ekibi

İçindekiler

01

Raspberry Pi Pico'yu Tanıyalım

robotistan

BLOG

maker.robotistan.com

FORUM

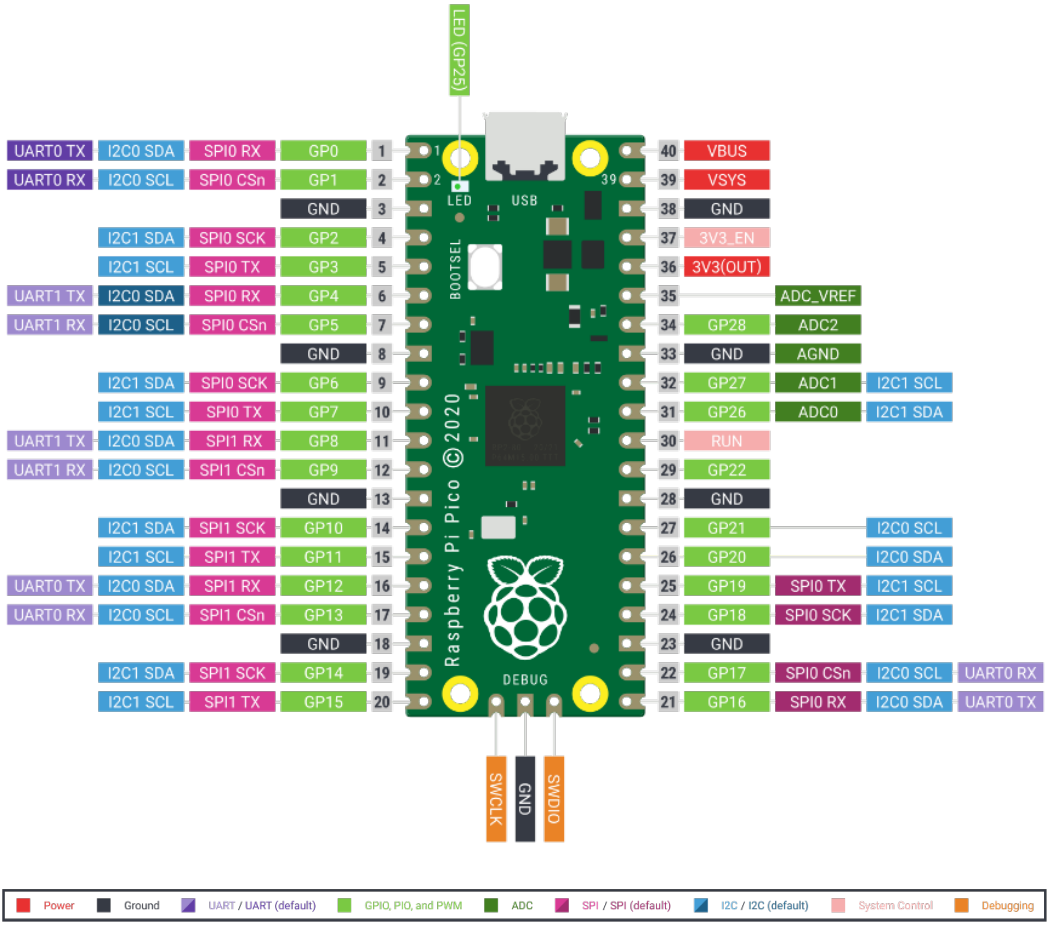
forum.robotistan.com

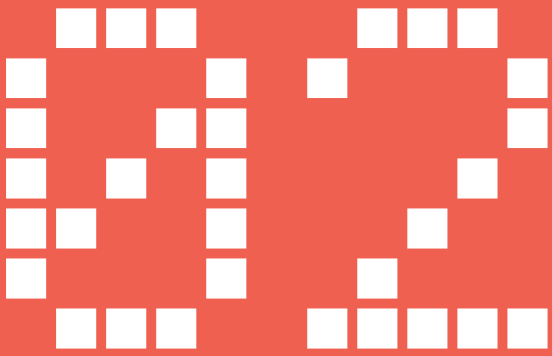
YouTube

youtube.com/robotistan

Raspberry Pi Vakfı tarafından piyasaya sunulan Raspberry Pi Pico gömülü sistemler projelerimizde, prototiplemede ya da bizlere sunduğu Micropython ile hem elektronik hem de kodlama öğrenmek için kullanabileceğimiz bir mikrodenetleyicidir. Arduino Nano'ya göre oldukça güçlü olan Raspberry Pi Pico 32 bit Arm Cortex M0+ 133 MHz işlemcisiyle öne çıkıyor. Pinlerini incelediğimizde bizlere geniş hareket alanı sağlıyor. Üzerinde bulunan "Gerçek Zamanlı Saat(RTC)" modülüyle bir veri kaydedici gibi de kullabiliriz.

Ayrıca üzerinde bulunan sıcaklık sensörü sayesinde hava durumu monitörü gibi projeleri yapmak oldukça kolay olacak bizim için. Lafı çok uzatmadan gelin Thonny IDE Kurulumuna geçelim. Eminin siz de benim gibi Micropython'ı çok seveceksiniz.





Raspberry Pi Pico'ya Thonny IDE Kurulumu

robotistan

BLOG

maker.robotistan.com

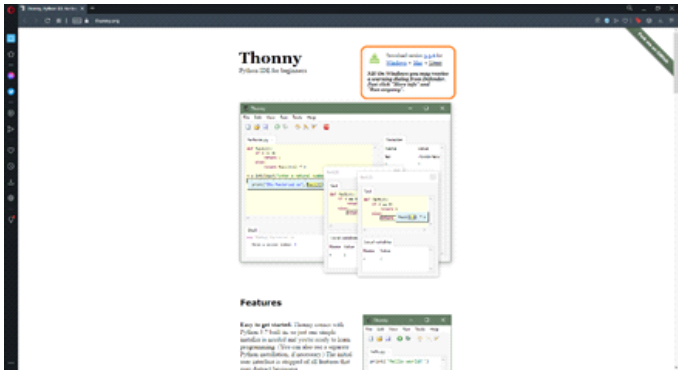
FORUM

forum.robotistan.com

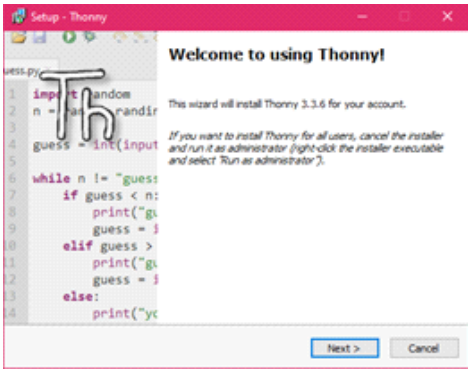
YouTube

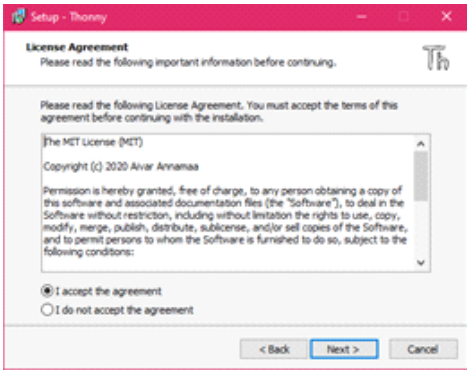
youtube.com/robotistan

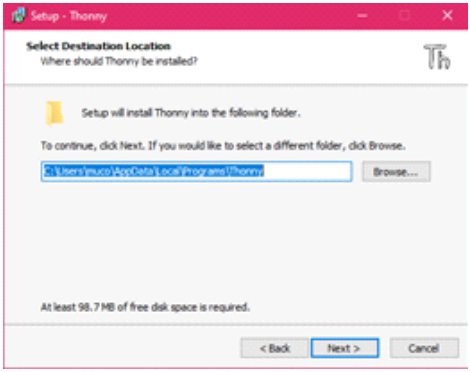
Öncelikle **https://thonny.org** adresinden işletim sistemimize uygun Thonny dağılımını indirelim. Ben Windows 10 kullanıyorum, o yüzden Windows'u seçeceğim.

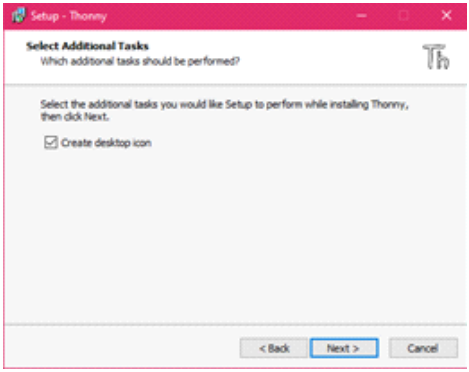


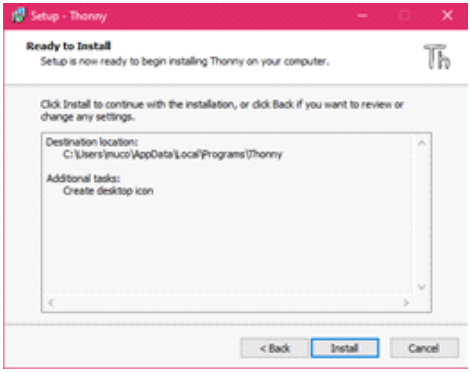
İndirdiğimiz kurulum dosyasını çalıştıralım ve aşağıdaki ekran görüntülerine göre sırayla kurulum adımlarını gerçekleştirelim.

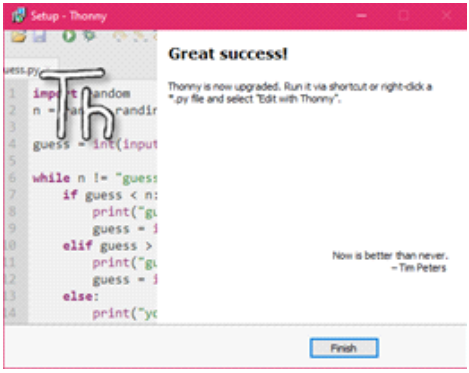












Son olarak "**Finish**" butonuna tıklayarak kurulumumuzu bitirelim.

Raspberry Pi Pico'ya MicroPython Firmware'i Yükleme

robotistan

BLOG

maker.robotistan.com

FORUM

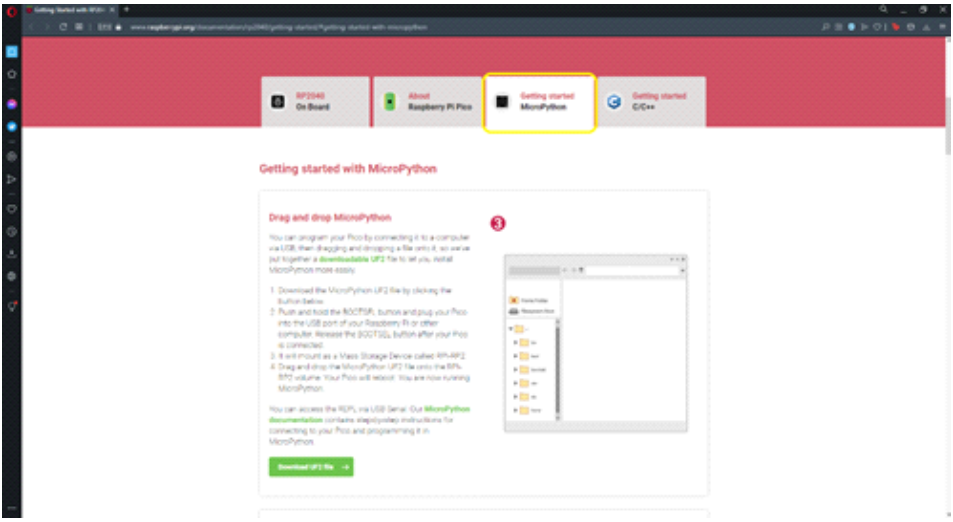
forum.robotistan.com

YouTube

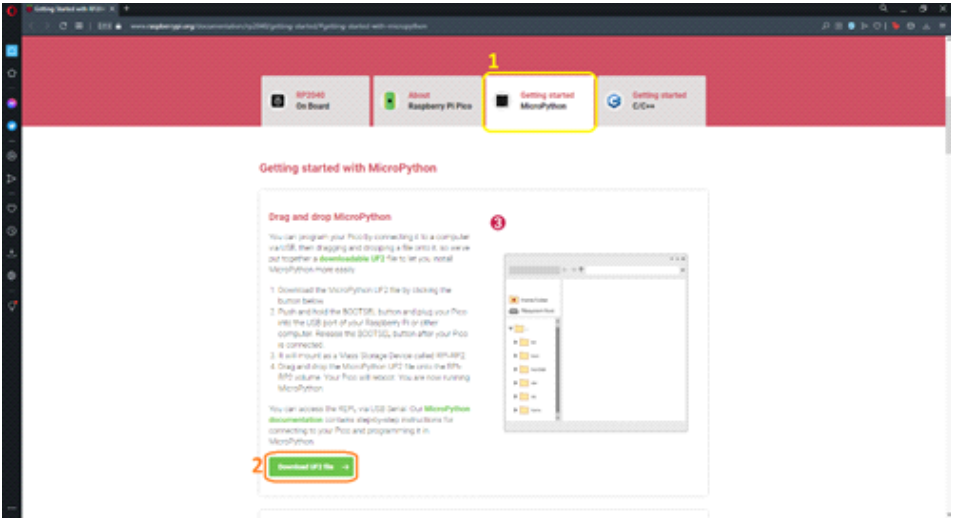
youtube.com/robotistan

Raspberry Pi Pico “MicroPython” ve “C/C++” olmak üzere 2 ayrı programlama dilinde programlanabiliyor. Biz de Pico'muzu programlayabilmemiz için uygun firmware dosyasını yüklememiz gerekiyor. Yükleme işlemi için aşağıdaki adımları izlemeniz yeterli olacaktır.

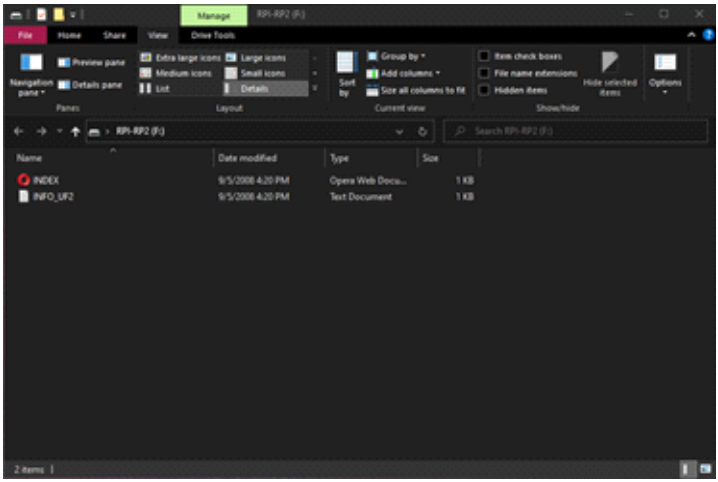
Https://www.raspberrypi.org/documentation/rp2040/getting-started/ adresini açalım ve sayfanın biraz aşağısında bulunan “**Getting Started MicroPython**” sekmesine tıklayalım.



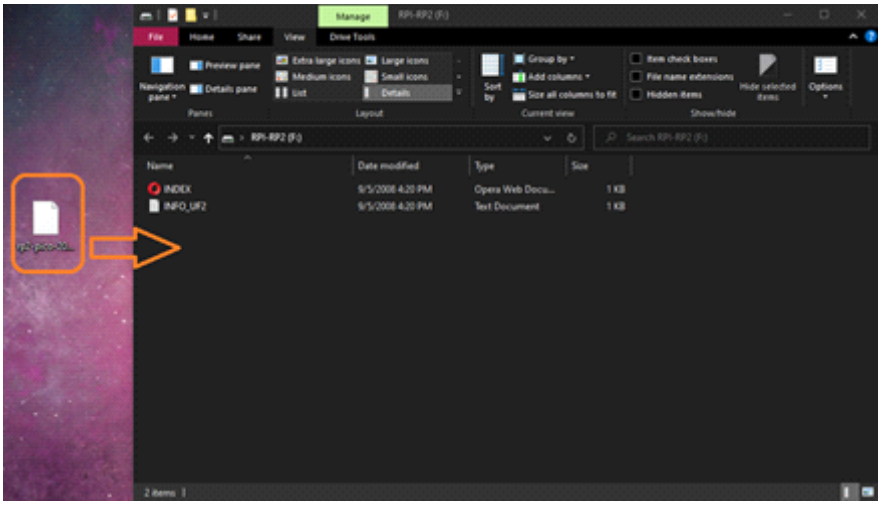
Altta bulunan “**Download UF2 file**” butonuna tıklayalım ve firmware'i indirelim.



Pico'muzun üzerinde bulunan “BOOTSEL” tuşuna basılı tutarak bilgisayarımıza bilgisayarımıza bağlayalım. Bilgisayarımıza bağladığımızda fotoğraftaki gibi bir klasör ekrana gelecektir.



İndirdiğimiz “***.UF2**” dosyasını açılan bu klasöre kopyalayalım.



Kopyalama işlemi bitince klasör otomatik olarak kapanacaktır. Şimdi de Thonny Arayüzünü tanıyalım.

04

Thonny IDE Arayüzünü Tanıyalım

robotistan

BLOG

maker.robotistan.com

FORUM

forum.robotistan.com

YouTube

youtube.com/robotistan

Raspberry Pi Pico Arduino'dan farklı olarak kendi hafızasına sahip ve biz bu hafızaya Thonny IDE aracılığıyla istediğimiz scriptleri ya da kütüphaneleri yükleyebiliyoruz. Gelin Thonny IDE'de bizim işlerimizi kolaylaştıracak neler varmış birlikte bakalım.
Thonny IDE'yi açtığımızda bizi fotoğraftaki gibi bir arayüz karşılayacak. En üst kısımda bulunan, turuncu kutunun içerisine aldığım kısımdaki sekmelerin işlevleri sırasıyla;

·Files: Türkçesi “Dosyalar” olan bu sekmeden yazdığımız scriptleri kaydedebilir, daha önce yazıp, kaydettiğimiz script varsa onları açabilir ya da yeni bir proje dosyası açabiliriz.

·Edit: Türkçesi “Düzenle” olan bu sekmeden yazdığımız scriptte geri alma, ileri alma, kopyalama, tümünü seçme gibi işlemleri yapabiliriz.

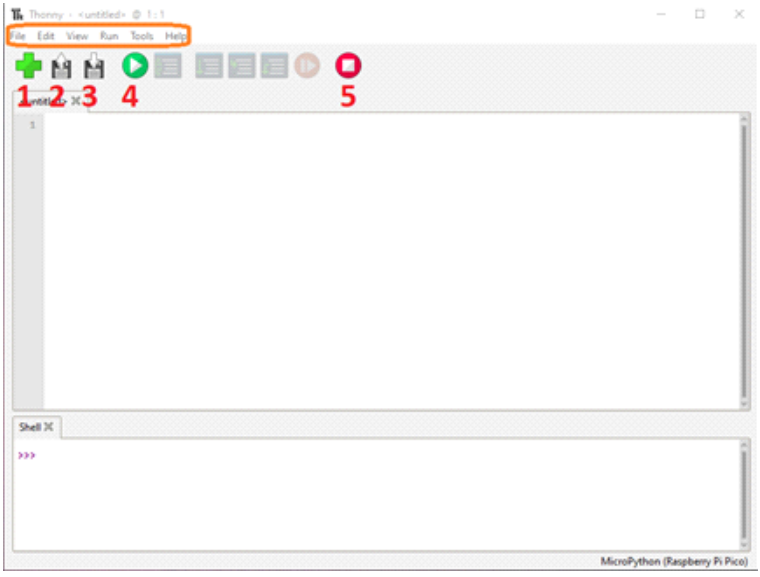
·View: Türkçesi “Görünüm” olan bu sekmeden arayüzümüzü özelleştirebiliriz.

·Run: Türkçesi “Çalıştır” olan bu sekmeden yazdığımız kodları çalıştırabilir, Raspberry Pi Pico'muz için “Yorumlayıcı(Interpreter)” seçebilir ya da kodda “Hata Ayıklama(Debug)” gibi işlemleri yapabiliriz.

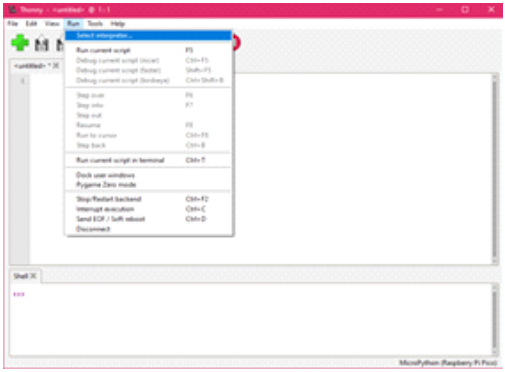
·Tools: Türkçesi “Araçlar” olan bu sekmeden paketleri yönetebiliriz ya da thonny ayarlarını yapabiliriz.

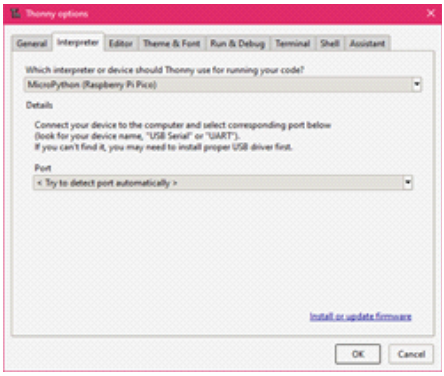
·Help: Türkçesi “Yardım” olan bu sekmeden thonny sürümü hakkında bilgi alabilir ya da karşılaştığımız bir sorunla ilgili yardım içeriklerine bakabiliriz.

1. “Artı” ikonuyla yeni bir dosya açabiliriz.
2. “Disket (Yukarı Ok)” ikonuyla daha önce kaydettiğimiz scripti açabiliriz
3. “Disket (Aşağı Ok)” ikonuyla yazdığımız scripti kaydedebiliriz.
4. “Çalıştır” ikonuyla yazdığımız scripti çalıştırabiliriz.
5. “Dur” ikonuyla çalışan scripti durdurabiliriz.

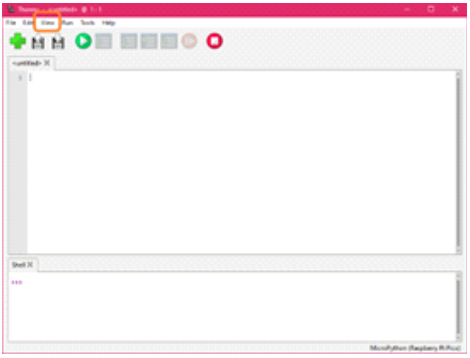


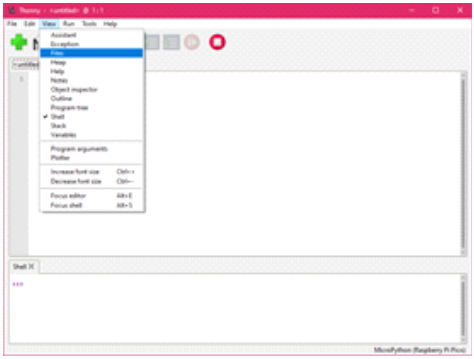
Şimdi Raspberry Pi Pico'muzu kullanabilmemiz Thonny IDE'de “Interpreter(Yorumlayıcı)” ayarlarını yapalım. Üst kısımda bulunan “Run” sekmesine tıklayalım, ardından “Select Interpreter”e tıklayalım. Açılan pencereden “Interpreter(Yorumlayıcı)” olarak “MicroPython(Raspberry Pi Pico)” ve Port olarak “Try to detect port automatically” seçelim.



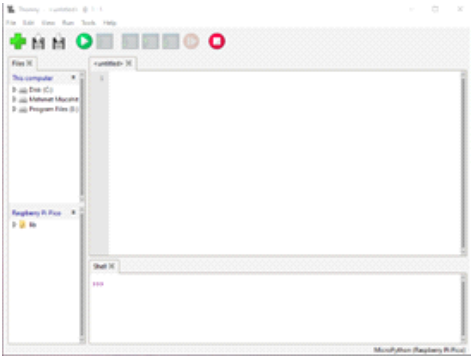


Şimdi arayüzde işimizi kolaylaştıracak olan “Files(Dosyalar)” sekmesini ekleyelim. Üst kısımda bulunan “View” sekmesine tıklayalım ve “Files”a tıklayalım.





Tıkladıktan sonra solda fotoğraftaki gibi bir sekme çıkacak. Buradan bilgisayarımızdaki ve Raspberry Pi Pico'muzdaki dosyalar görünecektir. Artık kolaylıkla Pico'muzdaki dosyaları kontrol edebiliriz.



05

Raspberry Pi Pico ile Dahili LED Yakma

robotistan

BLOG

maker.robotistan.com

FORUM

forum.robotistan.com

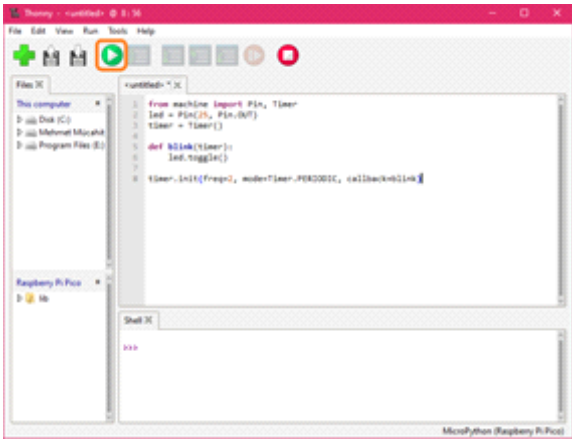
YouTube

youtube.com/robotistan

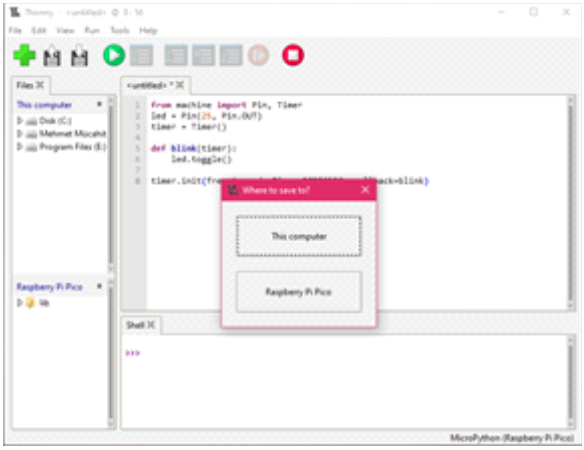
Her sette olduğu gibi bu sette de ilk projemiz LED yakmak olacak. Çünkü Maker kültürünün yazılı olmayan kuralı “Her şey LED yakmakla başlar”.
Aşağıdaki kodu Thonny IDE'ye yazalım. Bu kod dahili LED'in yarım saniyede bir yanıp, sönmesini sağlıyor. Bu kodu incelediğimizde Pico'nun üzerindeki donanıma erişebilmemiz için “from machine import Pin, Timer” kodunu yazdık. 2. Satırdaysa Pico'nun üzerinde bulunan dahili LED'in 25 numaralı pine bağlı olduğunu belirttik. 3. Satırda zamanlayıcıyı başlattık. 5. Satırda bir blink fonksiyonu tanımladık. Bu fonksiyon LED'in yanmasını sağlayan basit bir fonksiyon. 8. Satırdaysa zamanlayıcıyı çalıştırdık. Parantez içerisinde “freq” ile işlemin sıklığını, “mode” ile zamanlayıcının modunu ve “callback” ile hangi fonksiyonu çağıracağımızı belirttik.

```
from machine import Pin, Timer
led = Pin(25, Pin.OUT)
timer = Timer()

def blink(timer):
    led.toggle()

timer.init(freq=2, mode=Timer.PERIODIC, callback=blink)
```

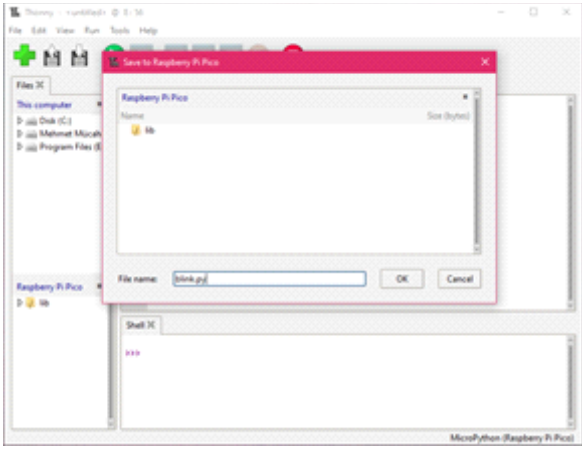
Şimdi bu kodu Pico'muza yükleyelim. Üstte bulunan yeşil “Çalıştır” ikonuna tıklayalım.



Kodu nereye kaydetmek istediğimizi soruyor. Eğer kodu bilgisayara kaydedip çalıştırırsak, kod sadece Pico bilgisayara takılıyken çalışır. Kodun sürekli çalışması için “Raspberry Pi Pico” seçeneğine tıklayıp, Pico'muza kaydetmeliyiz.



“Raspberry Pi Pico” yazısına tıkladıktan sonra fotoğraftaki gibi bir pencere açılacak. Bu pencerede bizden yazdığımız kodu Pico'da hangi konuma kaydetmek istediğimizi soruyor. Direkt Pico'nun içerisine kaydedebiliriz. Bunun için yazdığımız kodu kaydetmemiz için bir isim ve dosya uzantısı yazmamız gerekli. MicroPython ile çalıştığımız için dosya uzantılarımız hep ”*.py” olacaktır. Ben dosya adı olarak “blink.py” yazdım. Siz de istediğiniz bir dosya adı yazabilirsiniz. Dosya adını yazdıktan sonra “OK” tuşuna tıklayarak kodu kaydedip, çalıştırabiliriz.



Kod çalıştığında, Pico'muzun üzerinde bulunan dahili LED yarım saniye aralıklarla yanıp, sönecektir.

Raspberry Pi Pico ile Buton ile LED Yakma

robotistan

BLOG

maker.robotistan.com

FORUM

forum.robotistan.com

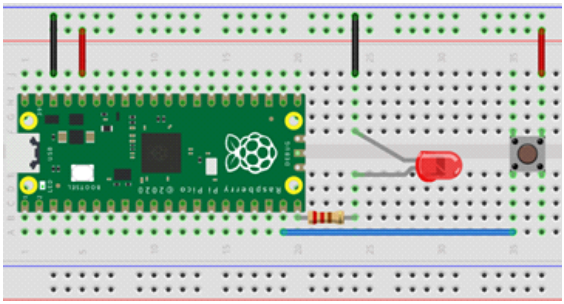
YouTube

youtube.com/robotistan

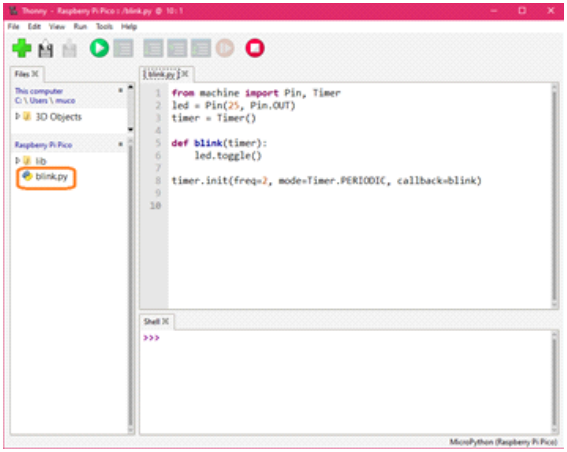
Bir kere LED yaktık ama bu sefer de Pico'muzun donanıma ve breadboard kullanımına ısınmamız için bu projeyi gerçekleştirmemiz gerekli.

Gerekli Malzemeler:
- Raspberry Pi Pico
- Breadboard
- Push Buton
- LED
- Erkek – Erkek Jumper Kablo
- 220 Ω ya da 330 Ω Direnç

Pico'muzun pin diyagramına baktığımız zaman 38. Pin “GND” yani toprak pini, 36. Pinse “3V3(OUT)” pini. Projelerimizde bu iki pini sıkça kullanacağız.
İlk olarak aşağıdaki şemaya göre devremizi breadboard üzerinde oluşturalım.



Ardından Thonny IDE'mizi açalım ve bir önceki projede yazdığımız “blink” kodunu düzenleyelim. Thonny IDE'de sol tarafta bulunan “blink.py” dosyasına çift tıklayarak açalım. Bir önceki projede yazdığımız kodlar açılacaktır. Şimdi bu kodları tamamen silip, yeni kod yazacağız.

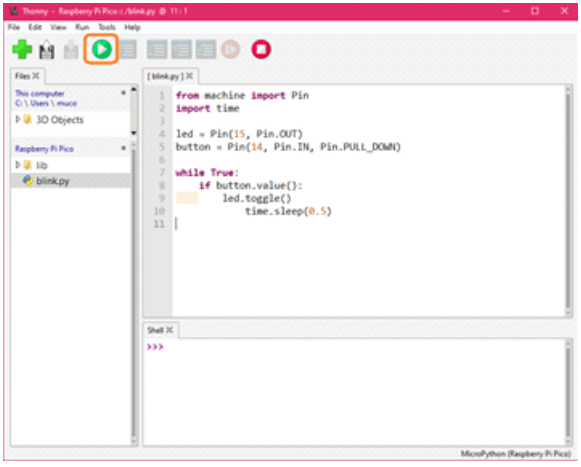


Aşağıdaki kodları incelediğimizde bir önceki projedeki gibi “from machine import Pin” koduyla Pico'nun donanımını tanımladık ve “time” kütüphanesini ekledik. “led = Pin(15,Pin.OUT)” koduyla 15 numaralı pini LED'i bağladığımız pin olarak atadık. “button = Pin(14, Pin.IN, Pin.PULL_DOWN)” koduyla butonu bağladığımız 14 numaralı pini giriş pini olarak tanımladık.

Ardından bir tane “while döngüsü” oluşturduk. Bu döngüde butondan okunan değere göre ledin durumunu değiştirecek. Mesela LED kapalıyken butona bir kere basarsak açılacaktır, eğer LED açıksa, butona bastığımızda kapanacaktır. “time.sleep(0.5)” koduyla da butona basılı tuttuğumuzda LED yarım saniye aralıklarla yanıp, sönecektir.

```
from machine import Pin
import time

led = Pin(15, Pin.OUT)
button = Pin(14, Pin.IN, Pin.PULL_DOWN)

while True:
    if button.value():
            led.toggle()
                time.sleep(0.5)
```

Kodları yazdıktan sonra üstte bulunan “Çalıştır” ikonuna basalım ve kodlarımızı Pico'muza kaydedip, çalıştıralım.



Evet artık butona basarak LED'i açıp kapatabileceğiz. Haydi sıradaki projeye geçelim.

07

Raspberry Pi Pico ile Sıcaklık Ölçümü

robotistan

BLOG

maker.robotistan.com

FORUM

forum.robotistan.com

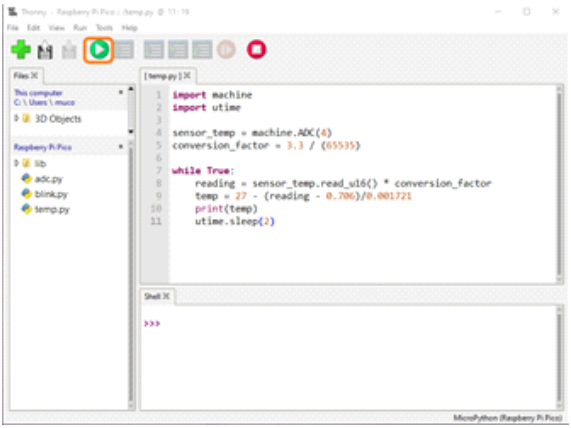
YouTube

youtube.com/robotistan

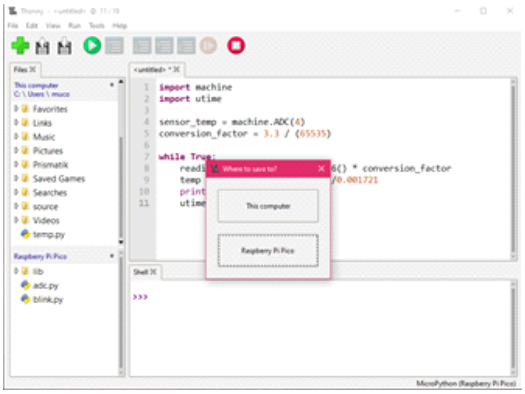
İlk başta da belirttiğim gibi Raspberry Pi Pico'da dahili sıcaklık sensörü bulunuyor. O zaman neden bir sıcaklık ölçümü yapmayalım.

Öncelikle Thonny IDE'yi açalım ve aşağıdaki kodları yeni bir dosya açıp, yazalım. Şimdi kodları incelediğimizde her zaman olduğu gibi gerekli olan “machine” ve “utime” kütüphanelerini ekledik. Sıcaklık sensörüyle analog okuma yaptığımız için “sensor_temp = machine.ADC(4)” koduyla 4 numaralı pine bağlı olduğunu tanımladık ardından “conversion_factor” koduyla sensörden gelen 16 bitlik veriyi bizim için anlamlı olabilecek gerilim verisine çevirdik. Pico'nun pini 3.3 volt verdiği için 3.3 voltu 216 – 1 = 65535'e böldük. Şimdi While döngüsünü incelediğimizde sensörden gelen veriyi okuma işlemi yapıyoruz. 8. Satırdaki kodda sıcaklık sensöründen gelen veriyi biraz önce yazdığımız “conversion_factor” ile çarparak sıcaklık verisini gerilim cinsinden yazıyoruz. Elde ettiğimiz veriyi 9. Satırda “Celcius”a çevirme işlemi yapıyoruz. Son olarak “print(temp)” koduyla sıcaklık verisini Shell'e yazdırıyoruz.
Şimdi bu kodları Pico'muza kaydedip, çalıştıralım. Üst kısımda bulunan “Çalıştır” ikonuna basalım.

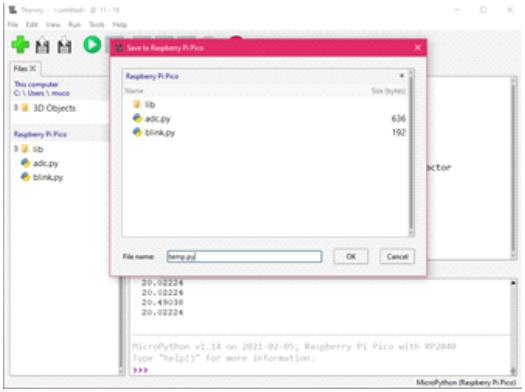
```
 1 import machine
 2 import utime
 3
 4 sensor_temp = machine.ADC(4)
 5 conversion_factor = 3.3 / (65535)
 6
 7 while True:
 8     reading = sensor_temp.read_u16() * conversion_factor
 9     temp = 27 - (reading - 0.706)/0.001721
10     print(temp)
11     utime.sleep(2)
```



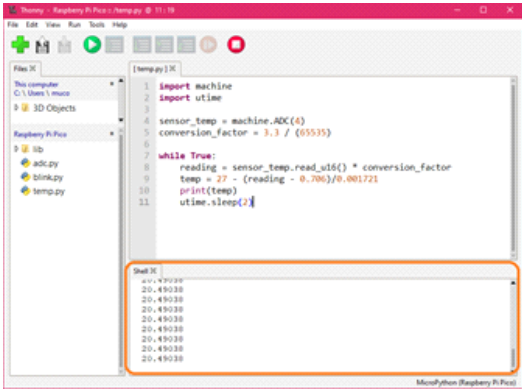
Kodu nereye kaydetmek istediğimizi soruyor. Kodun sürekli çalışması için “Raspberry Pi Pico” seçeneğine tıklayıp, Pico'muza kaydetmeliyiz.

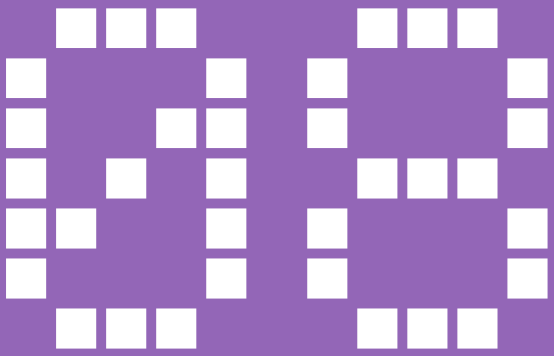


Dosya adı olarak “temp.py” yazıp, “OK” butonuna tıklayalım ve artık Shell'den sıcaklık verisini okuma işlemini yapabiliriz.



Alt kısımda bulunan Shell penceresinde 2 saniyede bir Pico'muzdan gelen sıcaklık verisini görebiliriz.





Raspberry Pi Pico ile Hırsız Alarmı

robotistan

BLOG

maker.robotistan.com

FORUM

forum.robotistan.com

YouTube

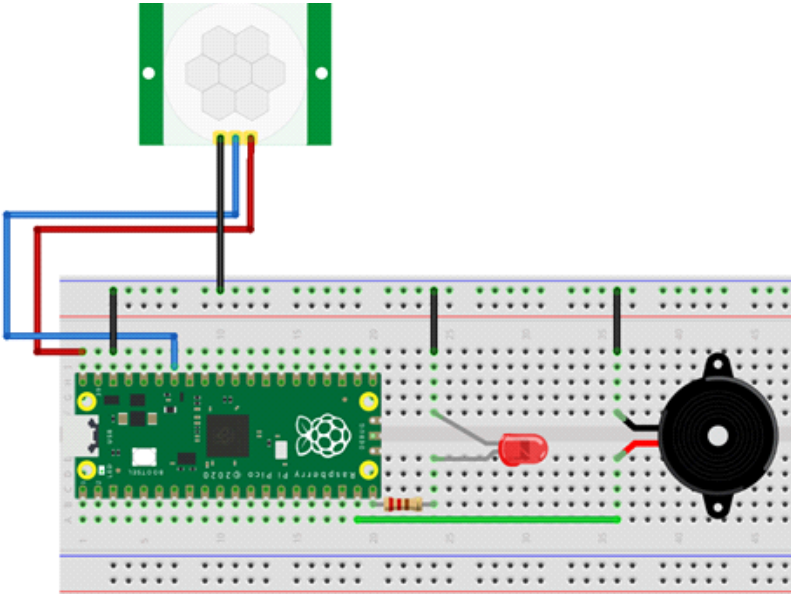
youtube.com/robotistan

Bu projemizde PIR sensörü, LED ve buzzer kullanarak bir hırsız alarmı yapacağız.

Gerekli Malzemeler:

- Raspberry Pi Pico
- Breadboard
- PIR Sensörü
- Buzzer
- LED
- Erkek – Erkek Jumper Kablo
- Erkek – Dişi Jumper Kablo
- 220 Ω ya da 330 Ω Direnç

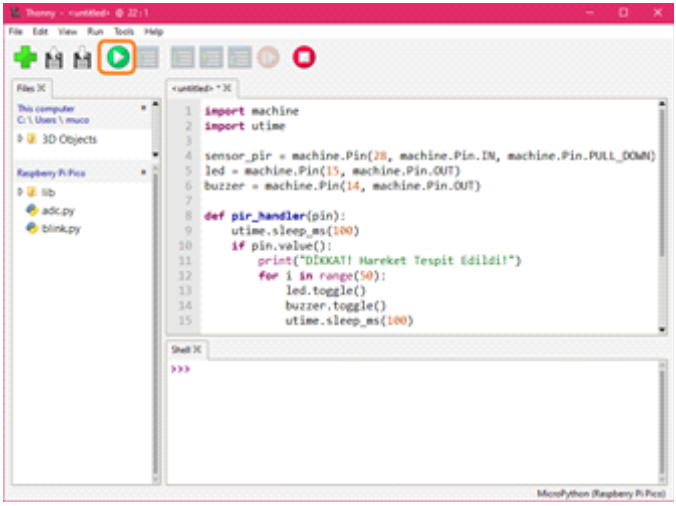
İlk olarak devremizi aşağıdaki şemaya göre oluşturalım.



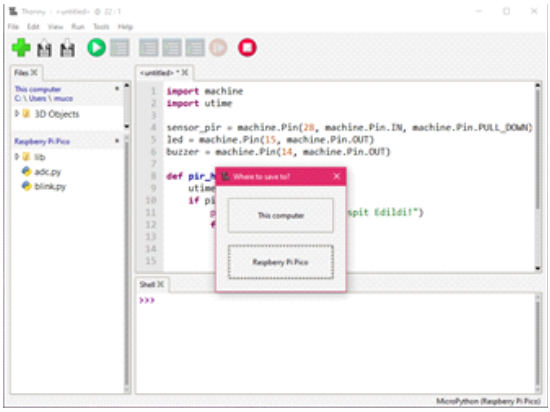
Şimdi de aşağıdaki kodları Thonny IDE'de yeni bir dosya açıp, yazalım. Kodları incelediğimizde ilk olarak kütüphanelerimizi ekledik ardından PIR sensörünü, Buzzer ve LED'i bağladığımız pinleri tanımladık. “def pir_handler(pin)” ile bir fonksiyon oluşturduk. Fonksiyonun içerisindeki if bloğu ile hareket tespit edildiğinde Shell ekranına “DİKKAT! Hareket Tespit Edildi!” yazdırıyor, buzzer ve LED'in çalışmasını sağladık.

```
import machine
import utime

sensor_pir = machine.Pin(28, machine.Pin.IN, machine.Pin.PULL_DOWN)
led = machine.Pin(15, machine.Pin.OUT)
buzzer = machine.Pin(14, machine.Pin.OUT)

def pir_handler(pin):
    utime.sleep_ms(100)
    if pin.value():
        print("DİKKAT! Hareket Tespit Edildi!")
        for i in range(50):
            led.toggle()
            buzzer.toggle()
            utime.sleep_ms(100)
sensor_pir.irq(trigger=machine.Pin.IRQ_RISING, handler=pir_handler)

while True:
    led.toggle()
    utime.sleep(5)
```

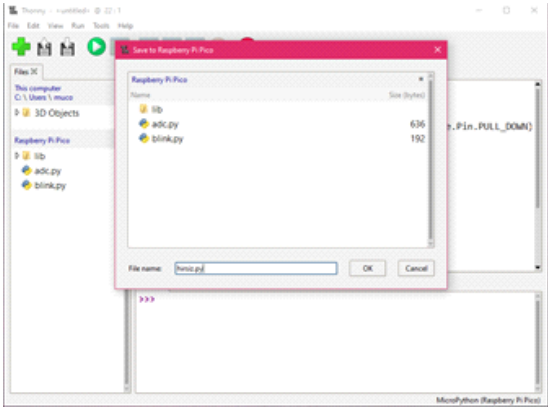
Şimdi bu kodları Pico'muza kaydedip, çalıştıralım. Üst kısımda bulunan “Çalıştır” ikonuna basalım.



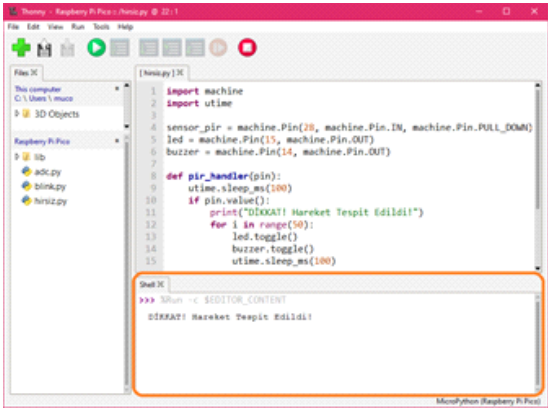
Kodu nereye kaydetmek istediğimizi soruyor. Kodun sürekli çalışması için “Raspberry Pi Pico” seçeneğine tıklayıp, Pico'muza kaydetmeliyiz.

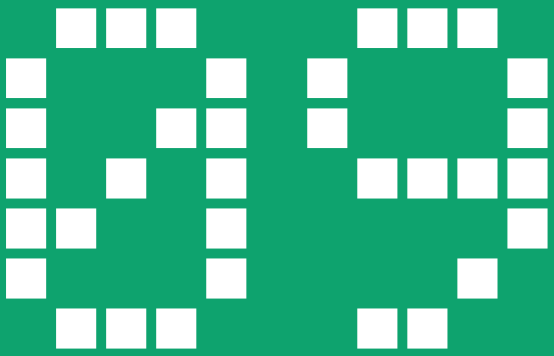


Dosya adı olarak “hirsiz.py” yazıp, “OK” butonuna tıklayalım ve artık hareket sensörümüzü tetikleyen herhangi bir durum söz konusu olduğunda buzzerdan ses gelecek ve LED'imiz yanacak.



Alt kısımda bulunan Shell penceresindeyse hareket tespit edildiğinde “DİKKAT! Hareket Tespit Edildi!” yazısı çıkacaktır.





Raspberry Pi Pico ile Hava Durumu Monitörü

robotistan

BLOG

maker.robotistan.com

FORUM

forum.robotistan.com

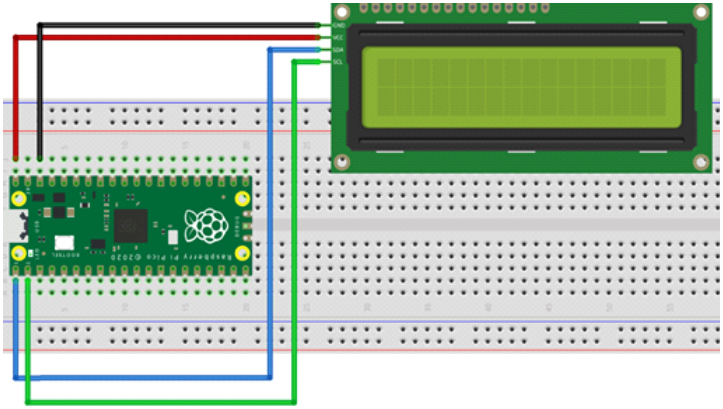
YouTube

youtube.com/robotistan

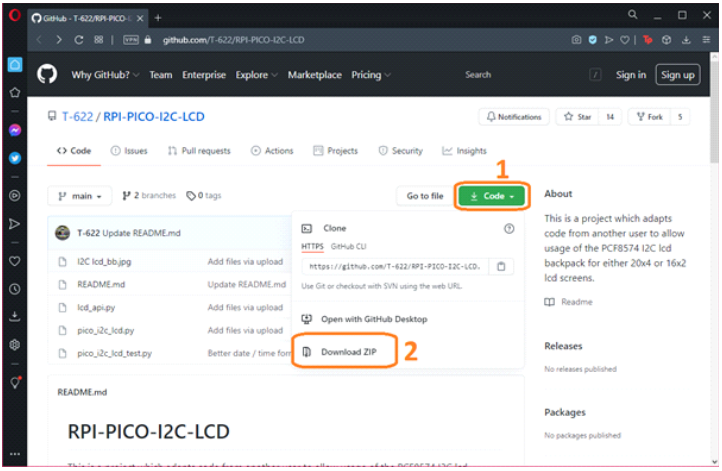
Bu projemizde 2X16 LCD ekran kullanarak bir hava durumu monitörü yapacağız. Ayrıca bu projemizde kütüphane eklemeyi ve ölçtüğümüz sıcaklık verilerini bir metin dosyasına otomatik olarak nasıl kaydedeceğiz öğreneceğiz.

Gerekli Malzemeler:
- Raspberry Pi Pico
- Breadboard
- 2X16 LCD Ekran
- Erkek – Dişi Jumper Kablo

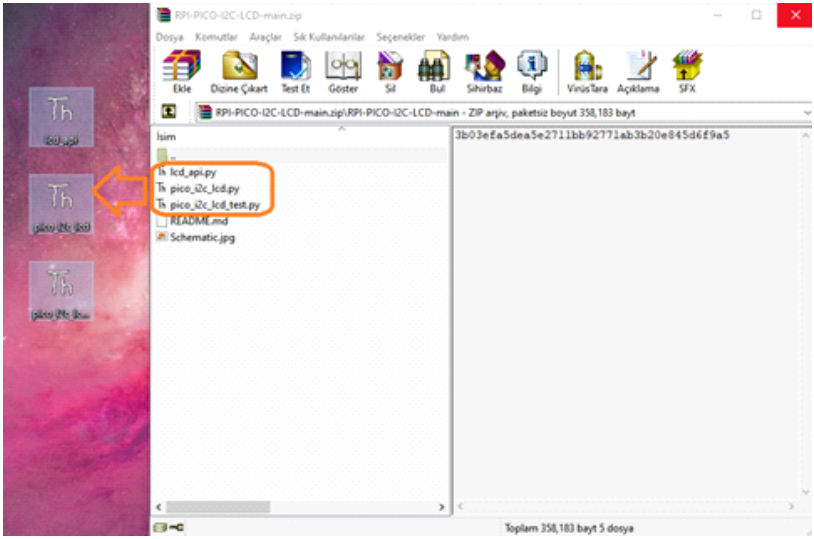
İlk olarak aşağıdaki devreye göre kendi devremizi breadboard üzerinde oluşturalım.



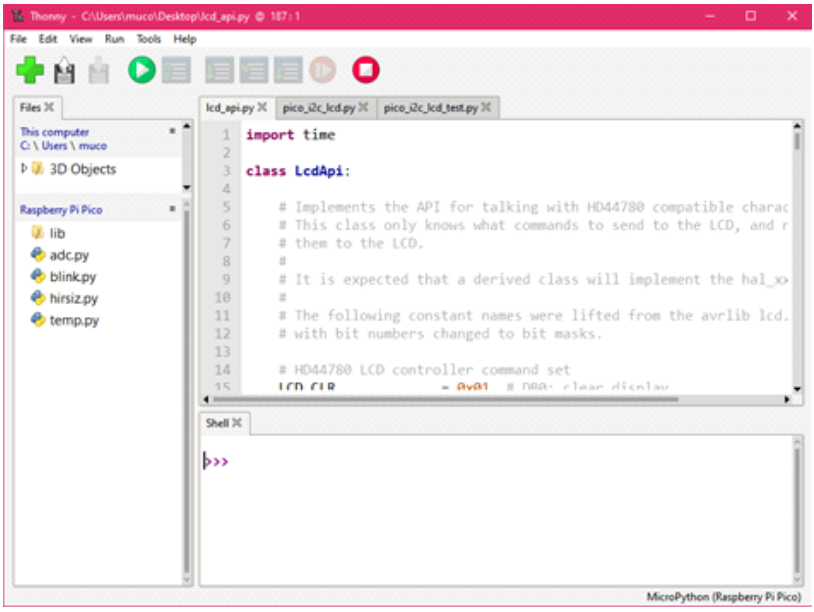
Şimdi **https://github.com/T-622/RPI-PICO-I2C-LCD** linkinden LCD ekranımızı kullanabilmemiz için gerekli kütüphaneyi indirelim. Linki açtığımızda "Code" yazan yeşil butona tıklayalım ve ardından "**Download ZIP**" butonuna tıklayarak kütüphaneyi indirelim.



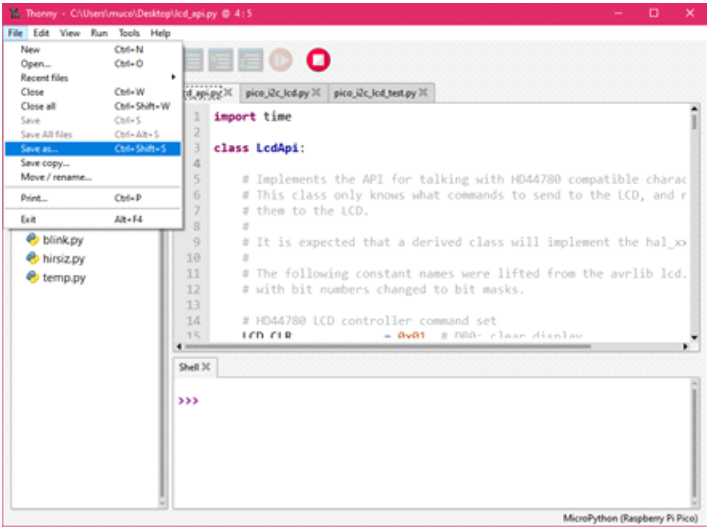
İndirdiğimiz “*.zip” dosyasını açalım ve içerisindeki “lcd_api.py”, “pico_i2c_lcd.py”, “pico_i2c_lcd_test.py” dosyalarını masaüstüne çıkaralım.



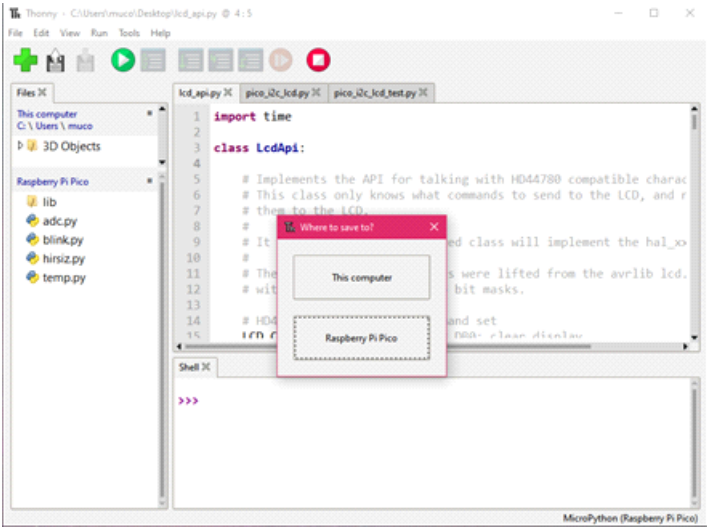
Masaüstüne çıkardığımız bu dosyalara sırayla çift tıklayarak Thonny IDE'de açalım.



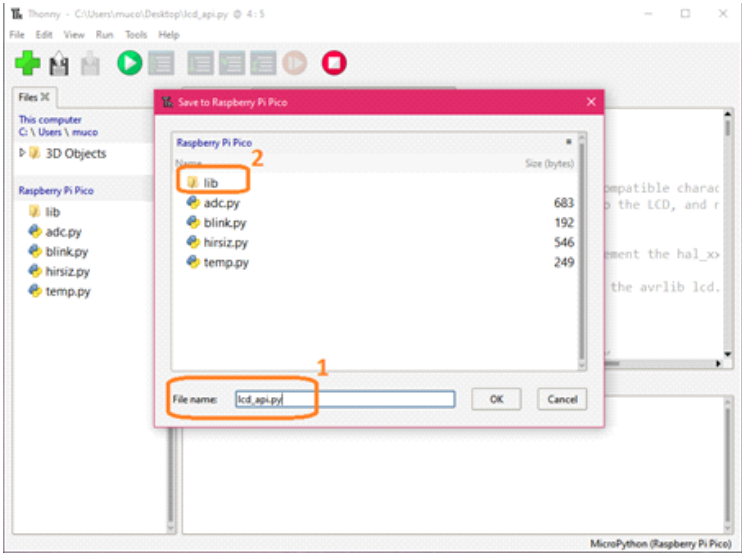
Şimdi bu dosyaları sırayla Pico'muzda bulunan “lib” klasörüne kaydedelim. Üstte bulunan “Files” sekmesine tıklayalım ve ardından “Save as...”e tıklayalım.



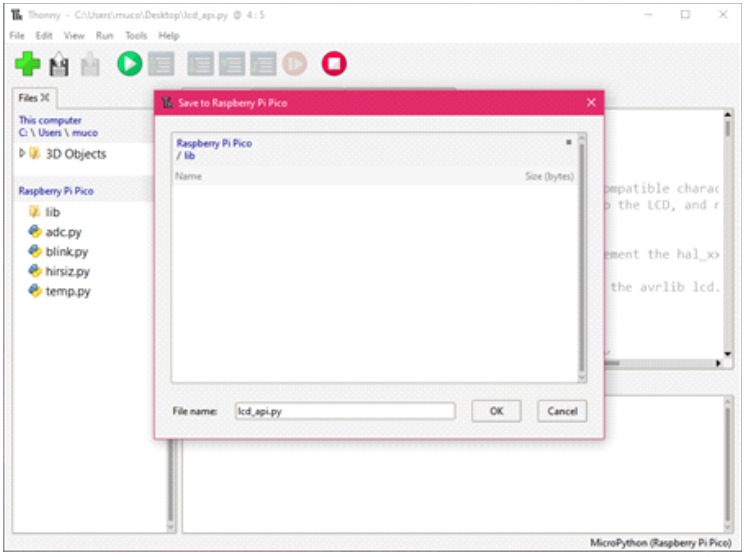
Kütüphane dosyasını nereye kaydetmek istediğimizi soruyor. Biz Raspberry Pi Pico'yu seçeceğiz.



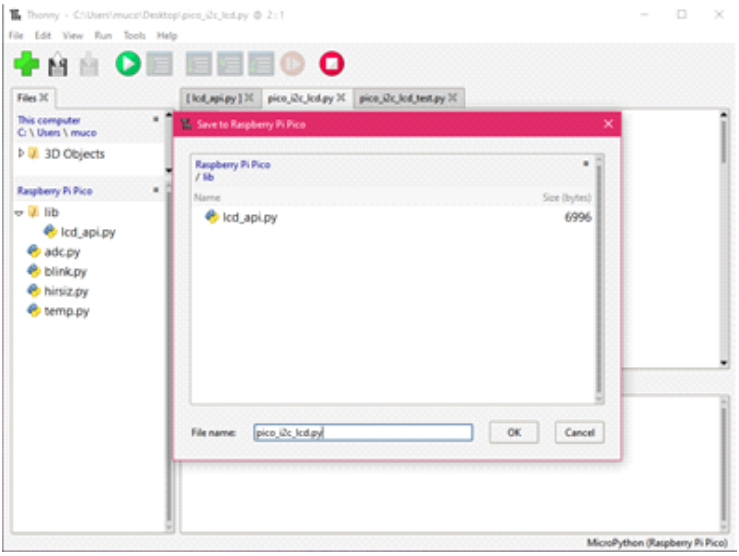
Bu kısımdaysa dosyayı kaydetmek istediğimiz konumu seçmemiz gerekiyor. İlk olarak dosya adını yazıp sonra “lib” klasörüne çift tıklayalım.

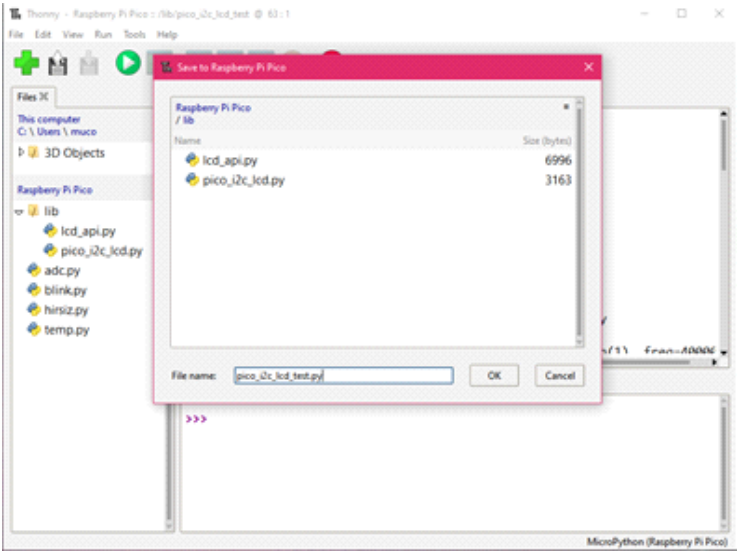


Ardından “OK” butonuna tıklayarak klasöre kaydedelim.



Aynı işlemi diğer iki dosya için de yapalım.

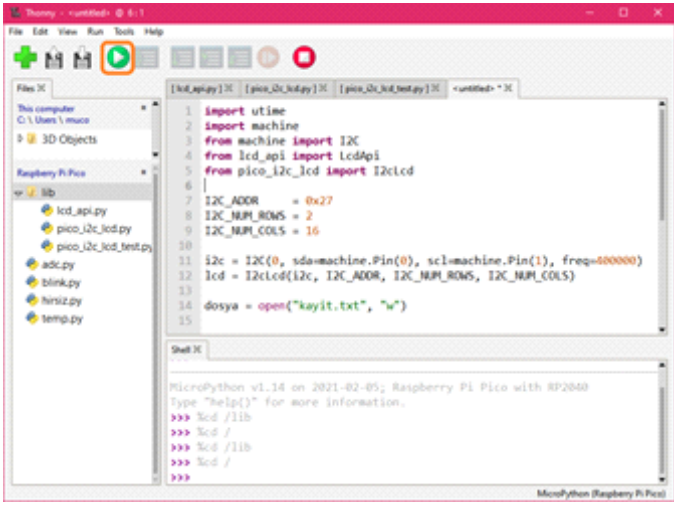




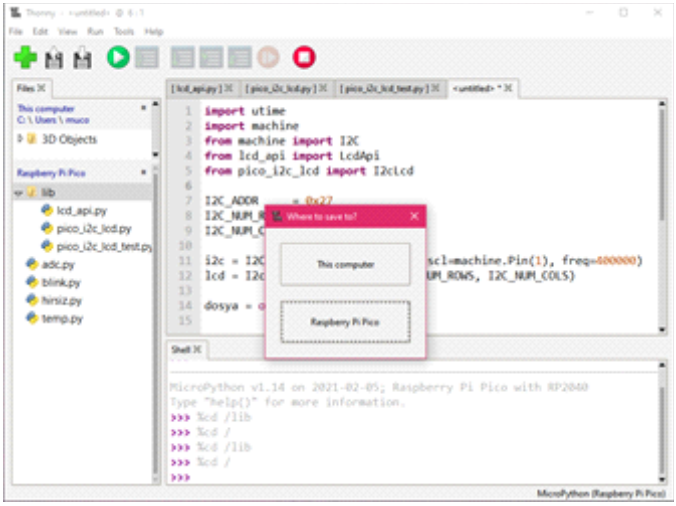
Kütüphaneleri eklediğimize göre artık yeni bir Thonny IDE'de üstte bulunan “Artı” ikonuna tıklayarak yeni bir dosya açıp, aşağıdaki kodları yazabiliriz. Kodları incelediğimizde diğer projelerde olduğu gibi ilk olarak kütüphaneleri ekledik. Ardından 2X16 LCD ekranımızın ayarlarını yaptık ve “dosya = open(“kayit.txt”,”w”)” koduyla “kayit” adında yeni bir metin belgesi oluşturduk.

```
import utime
import machine
from machine import I2C
from lcd_api import LcdApi
from pico_i2c_lcd import I2cLcd

I2C_ADDR     = 0x27
I2C_NUM_ROWS = 2
I2C_NUM_COLS = 16

i2c = I2C(0, sda=machine.Pin(0), scl=machine.Pin(1), freq=400000)
lcd = I2cLcd(i2c, I2C_ADDR, I2C_NUM_ROWS, I2C_NUM_COLS)

dosya = open("kayit.txt", "w")

sensor_temp = machine.ADC(4)
conversion_factor = 3.3 / (65535)

while True:
    lcd.clear()
    reading = sensor_temp.read_u16() * conversion_factor
    temp = 27 - (reading - 0.706)/0.001721
    lcd.putstr("Sicaklik: ")
    lcd.move_to(0,1)
    lcd.putstr(str(temp))
    dosya.write(str(temp) + "\n")
    dosya.flush()
    utime.sleep(2)
```

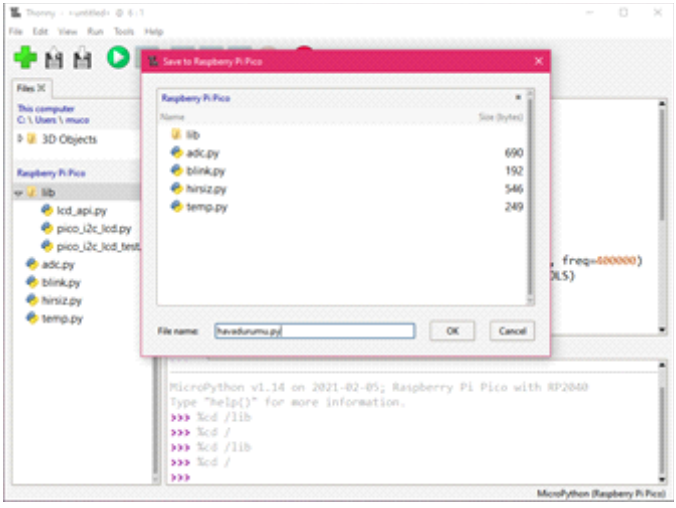
Şimdi bu kodları Pico'muza kaydedip, çalıştıralım. Üst kısımda bulunan “Çalıştır” ikonuna basalım.



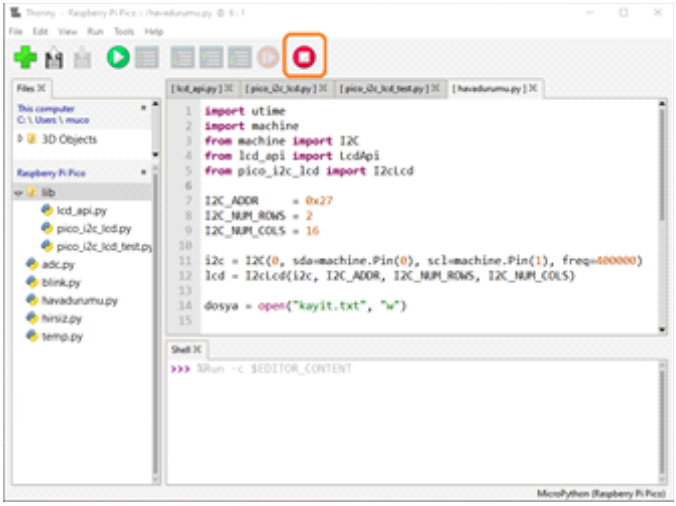
Kodu nereye kaydetmek istediğimizi soruyor. Kodun sürekli çalışması için “Raspberry Pi Pico” seçeneğine tıklayıp, Pico'muza kaydetmeliyiz.



Dosya adı olarak “havadurumu.py” yazıp, “OK” butonuna tıklayalım. Artık LCD ekranınımızda sıcaklık verisini görebileceğiz aynı zamanda bu veriler bir metin belgesinde kaydedilecek.

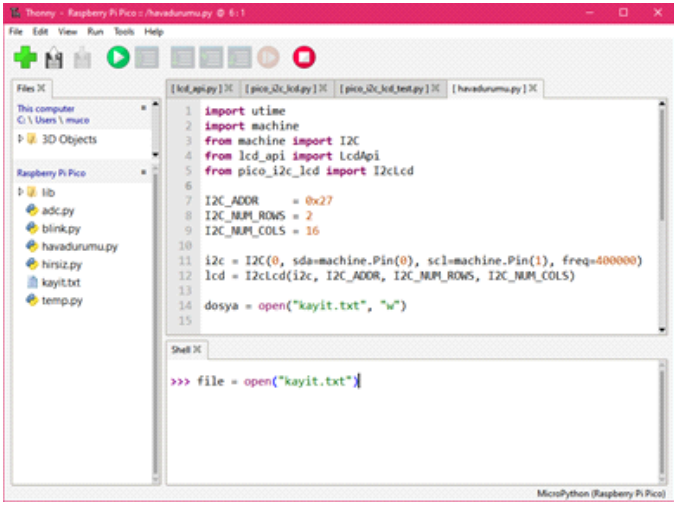


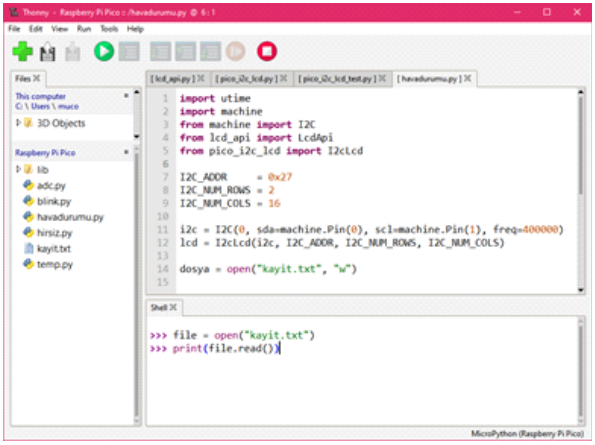
Şimdi biraz veri toplaması için bekleyelim ardından üstte bulunan “Dur” ikonuna tıklayarak kodun çalışmasını durduralım.



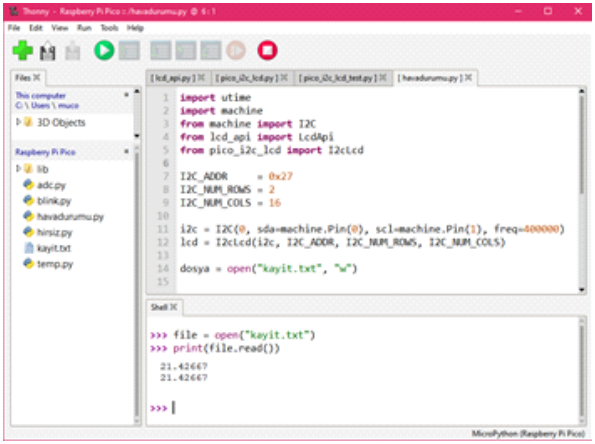
Kaydettiğimiz verileri görebilmemiz için alt kısımda bulunan Shell kısmına sırayla aşağıdaki kodları yazalım.

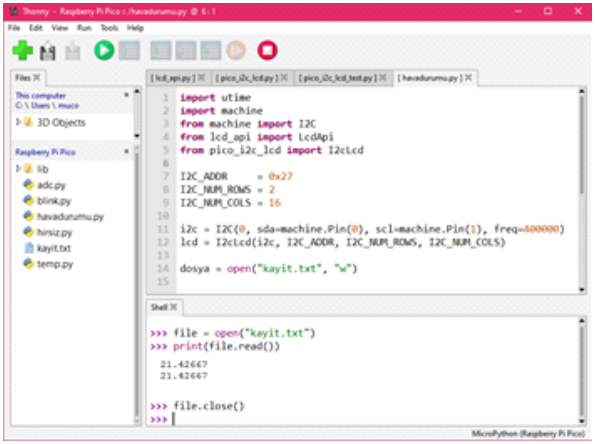
```
file = open("kayit.txt")
print(file.read())
file.close()
```





"Print(file.read())" kodunu yazdıktan sonra aşağıdaki fotoğraftaki gibi kaydedilen verileri görebiliyoruz.





10

Raspberry Pi Pico ile Mesafe Sensörü Kullanımı

robotistan

BLOG

maker.robotistan.com

FORUM

forum.robotistan.com

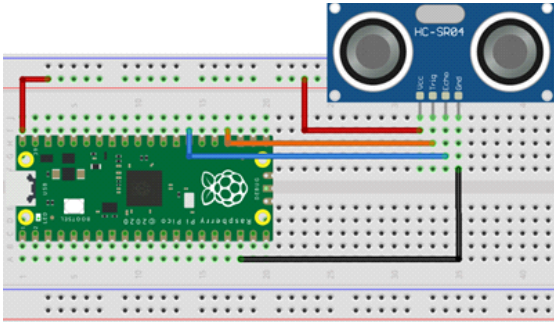
YouTube

youtube.com/robotistan

Bu projemizde HC-SR04 mesafe sensörüyle mesafe ölçümü yapacağız.

Gerekli Malzemeler:
- Raspberry Pi Pico
- Breadboard
- HC-SR04 Mesafe Sensörü
- Erkek – Erkek Jumper Kablo

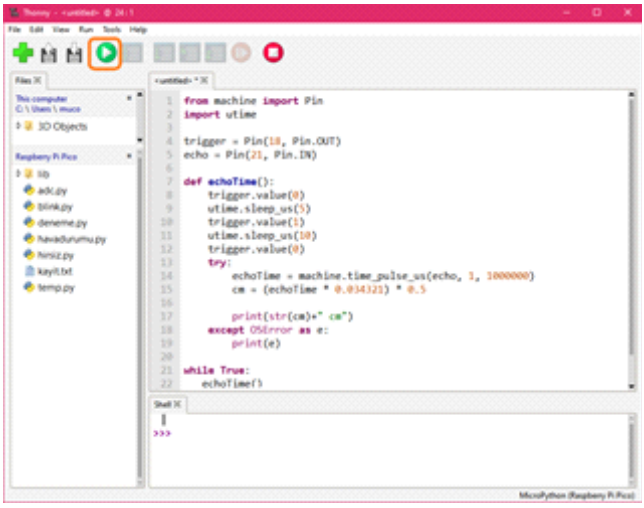
Öncelikle devremizi aşağıdaki şemaya göre breadboard üzerinde oluşturalım.



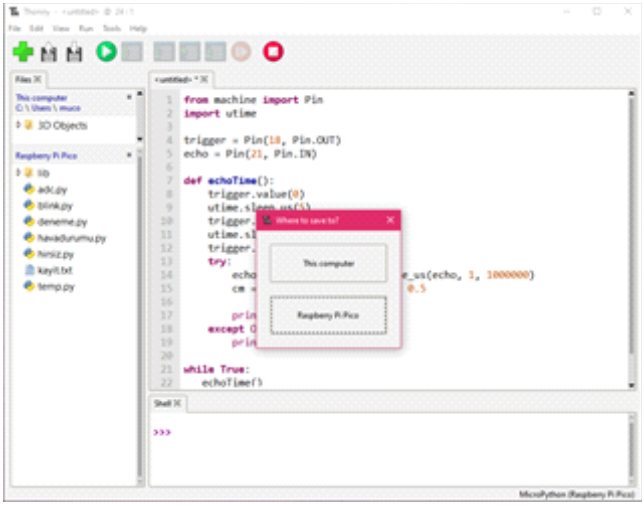
Şimdi de Thonny IDE'de yeni bir dosya açıp kodlarımızı yazalım. Kodları incelediğimizde ilk iki satırda kütüphaneleri tanımladık ardından mesafe sensörünün bağlı olduğu pinleri tanımladık. “def echoTime():” fonksiyonuyla da mesafe ölçümü fonksiyonunu yazdık. While döngüsüyle de tanımladığımız mesafe ölçüm fonksiyonunu 5 saniyede bir çalışmasını sağladık.

```
from machine import Pin
import utime

trigger = Pin(18, Pin.OUT)
echo = Pin(21, Pin.IN)

def echoTime():
    trigger.value(0)
    utime.sleep_us(5)
    trigger.value(1)
    utime.sleep_us(10)
    trigger.value(0)
    try:
        echoTime = machine.time_pulse_us(echo, 1, 1000000)
        cm = (echoTime * 0.034321) * 0.5

        print(str(cm)+" cm")
    except OSError as e:
        print(e)

while True:
   echoTime()
   utime.sleep(5)
```

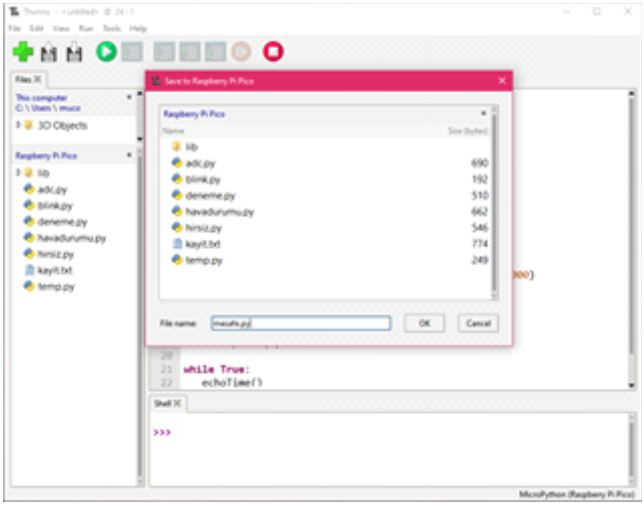
Şimdi bu kodları Pico'muza kaydedip, çalıştıralım. Üst kısımda bulunan “Çalıştır” ikonuna basalım.



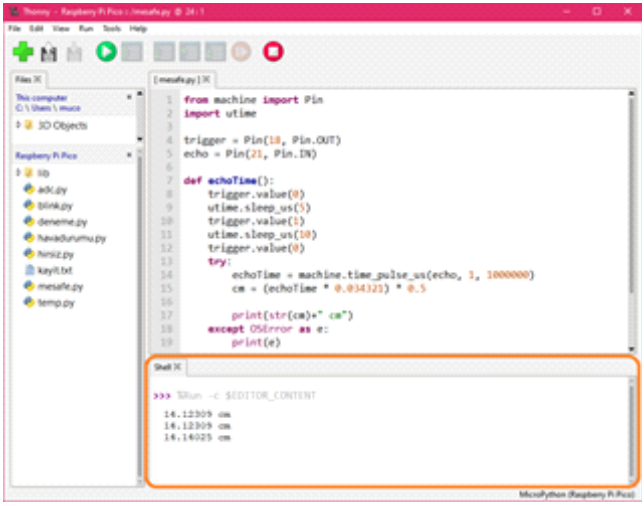
Kodu nereye kaydetmek istediğimizi soruyor. Kodun sürekli çalışması için “Raspberry Pi Pico” seçeneğine tıklayıp, Pico'muza kaydetmeliyiz.



Dosya adı olarak “mesafe.py” yazıp, “OK” butonuna tıklayalım.



Artık Thonny IDE'de bulunan Shell kısmında mesafeyi görebileceğiz.



YouTube

youtube.com/robotistan

FORUM

robotistan

forum.robotistan.com

BLOG

robotistan

maker.robotistan.com

Robotistan Elektronik Ticaret A.Ş.

Hazırlayanlar: Mehmet Mücahit KAYA(İçerik) - Mehmet AKÇALI - Yasin TAŞCIOĞLU (Editör) - Mehmet Nasır KARAER (Grafik)
info@robotistan.com - www.robotistan.com
Tel: 0850 766 0 425