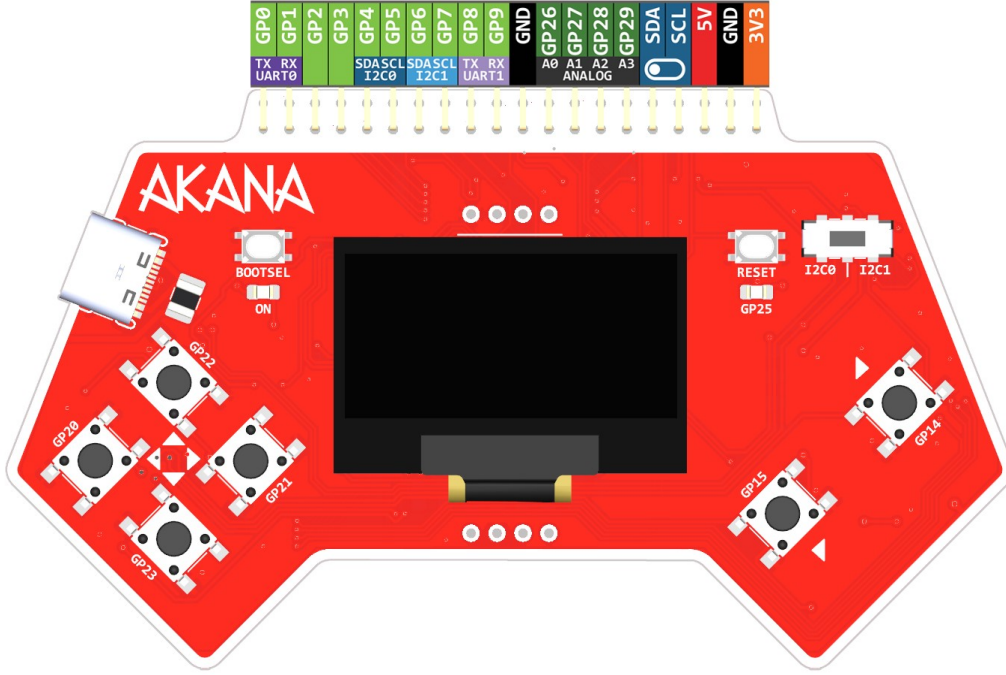


AKANA R1 KODLAMA KILAVUZU

ARDUINO IDE (MBED OS / PICO SDK) | MICROPYTHON | CIRCUITPYTHON



İÇİNDEKİLER

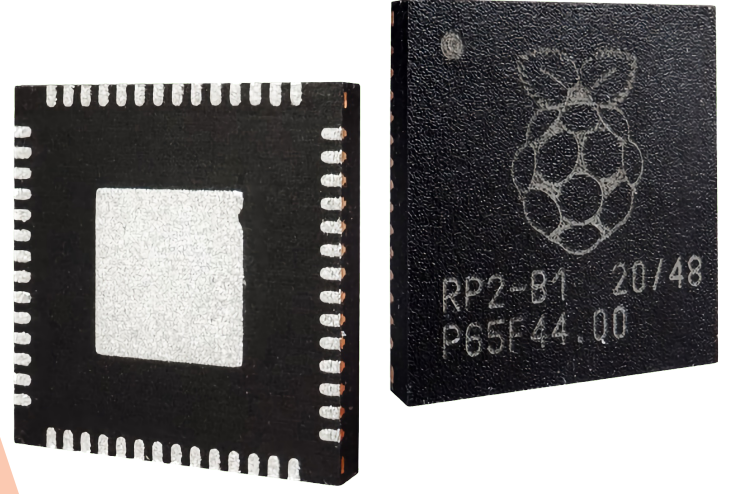
| | |
|---|----|
| Neden RP2040 | 2 |
| UF2 Bootloader | 3 |
| Akana R1'e Arduino Mbed OS Yükleme | 4 |
| Akana R1'i Arduino Mbed OS ile Kodlamak | 5 |
| Akana R1'e Arduino-Pico Yükleme | 6 |
| Akana R1'i Arduino-Pico ile Kodlamak | 7 |
| Akana R1'e MicroPython Yükleme | 8 |
| Akana R1'i MicroPython ile Kodlamak | 9 |
| Akana R1'e CircuitPython Yükleme | 10 |
| Akana R1'i CircuitPython ile Kodlamak | 11 |
| Sürümler | 12 |

NEDEN RP2040?

METE HOCA Akana R1, Raspberry Pi ekibinin geliştirdiği **RP2040** mikrokontrolcüsü üzerine kurulu olduğu için kullanımı ve yazılım desteği de bu mikrokontrolcüyü kullanan **Raspberry Pi Pico** ile aynı.

Arduino Uno R3'ün kısıtlı **32 KB** flash hafızasının aksine RP2040 üzerine kurulu Raspberry Pi Pico'da 2 MB, Akana R1'de ise **2 MB**, isteğe bağlı olarak 8 MB veya 16 MB flash hafıza yer alıyor.

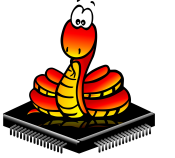
Öte yandan Uno R3'te 16 MHz hızında çalışan tek çekirdekli bir işlemci yer alırken RP2040 133 MHz hızında çalışan **çift çekirdekli** bir işlemciye sahip. Bu özellikleri RP2040'ı oldukça esnek kılıyor.



Arduino ekibi de bu yetenekli mikrokontrolcüyü görmezden gelmedi ve RP2040'ı kullanan **Arduino Nano RP2040 Connect** adında bir model üretti. RP2040'ın **Arduino IDE**'de **Mbed OS** ile kodlanabilmesini sağlayan Arduino ekibi bu kartın yanında doğrudan Raspberry Pi Pico'ya da destek veren yazılım hazırladı.

Öte yandan RP2040'ı **Arduino IDE** ile kodlamak için bir yöntem daha var. **Earle F. Philhower** tarafından RP2040'ın **Pico SDK** yazılım kiti kullanılarak hazırlanan yazılım RP2040 üzerine kurulu kartları çok daha verimli bir şekilde kullanabilmeye imkân veriyor.

Güçlü **Python** programlama dilinin mikrokontrolcülere indirgenmiş hafif sürümü olan **MicroPython** da RP2040'ı destekliyor. Bu mikrokontrolcüyü MicroPython'da kullanabilmek için çok sayıda kütüphane ve destek bulmak mümkün. **Robotistan** ekibinin geliştirdiği **PicoBricks** eğitim seti de blok kodlamanın yanında MicroPython üzerine kurulu.

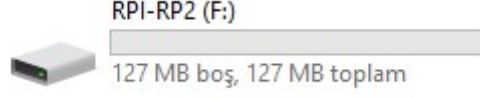


RP2040'ın en büyük destekçilerinden biri de **Adafruit**. Bu maker üretici devi, MicroPython temeli üzerine kurduğu ancak yaptığı eklemelerle birlikte çok farklı bir yere getirdiği **CircuitPython** adında kendi kodlama altyapısını geliştiriyor ve kütüphane konusunda da çok iyi destek veriyor.

Özetle; Akana R1'i Arduino IDE üzerinde 2 farklı yöntemle, Python dilinde de 2 farklı yöntemle kodlamak mümkün.

UF2 BOOTLOADER

Akana R1, üzerine kurulu olduğu RP2040'ın sahip olduğu **UF2 Bootloader** sayesinde kolayca kodlanabiliyor. UF2 Bootloader sayesinde Akana R1, bilgisayarda bir USB depolama aygıtı olarak görünüyor. Üstelik Windows 10, Mac ve hatta Linux'ta herhangi bir sürücü yüklemeye ihtiyaç duymuyor.



Peki, nedir bu UF2?

UF2'nin açılımı **USB Flashing Format**. Microsoft tarafından geliştirilen bu yapı kodların mikrokontrolcülere çıkarılabilir USB diske kopyalanması şeklinde kolaylıkla yüklenebilmesini sağlıyor. Böylece UF2 dosyasına dönüştürülmüş kodlar basit bir sürükle-bırak ile kolayca yüklenebiliyor.

Akana R1'i UF2 Bootloader moduna geçirmek için **BOOTSEL** butonunu basılı tutarken USB kablosunu takmak gerekiyor. Kart güç kesilene, bir UF2 dosyası kopyalanana veya RESET butonuna basılana kadar bu modda kalıyor. Akana R1'in UF2 Bootloader moduna geçtiğini bilgisayarda **RPI-RP2** adında bir USB depolama aygıtı olarak görünmesiyle anlamak mümkün.



Akana R1'i UF2 moduna geçirmek için bir diğer yöntem de USB kablosu takılıken BOOTSEL butonunu basılı tutarken bir kez **RESET** butonuna basmaktır. Bu yöntem belki biraz el marifeti gerektirebilir.

UYARI: Uzun USB kablolar veya USB uzatma kabloları sinyal bütünlüğünü bozabileceğinden dolayı Akana R1'in UF2 Bootloader moduna geçmesini engelleyebilir.

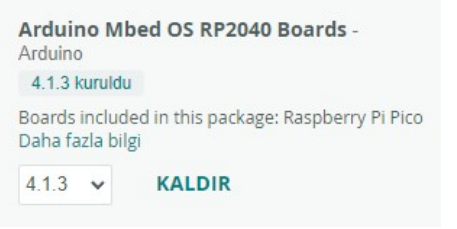
Mbed OS, Pico SDK, MicroPython veya CircuitPython kodlama yöntemleri arasında geçiş yaparken Akana R1'in doğru şekilde algılanmasını sağlamak için UF2 Bootloader moduna geçirilmesi gerekir.

AKANA R1'e ARDUINO MBED OS YÜKLEMEK

Yeni ürünlerinde giderek daha güçlü mikrokontrolcüler kullanmaya başlayan Arduino ekibi, **ARM** altyapısına sahip yeni nesil mikrokontrolcülerde kullanmak için hazırlanmış olan **Mbed OS**'u temel alarak kendi yazılımını oluşturdu. Mbed OS, mikrokontrolcüler gibi görece zayıf ARM işlemcilerde kullanılabilen hafif ama güçlü bir **RTOS** (Real Time Operating System / Gerçek Zamanlı İşletim Sistemi) altyapısı.

Bu sayede **32-bit ARM Cortex** serisi işlemcilere sahip yeni nesil Arduino kartları eşzamanlı işlem yapabilmeye gibi daha önce hayal edilemeyen yeteneklere kavuştu. Altyapının Mbed OS üzerinde ortak olması farklı model ARM işlemcilere sahip kartların aynı kodu rahatça çalıştırabilmesinin önünü açıyor.

Arduino Mbed OS yazılımı RP2040 için de uyarlandı ve Raspberry Pi Pico'ya özel bir sürümü de yayınlandı. Bu sürümü Arduino IDE'nin kart yöneticisi üzerinden **Arduino Mbed OS RP2040 Boards** seçerek yüklemek mümkün.



Kart olarak Raspberry Pi Pico seçtikten sonra Akana R1 kodlamaya hazır olur.

Öncelikle Akana R1'e Mbed OS'u yüklemek gerekiyor. Bunu Arduino IDE otomatik olarak yapıyor. Tek yapılması gereken Akana R1'i **BOOTSEL** tuşu ile **UF2 Bootloader** moduna almak ve basit bir eskiz yüklemek. Örneğin Arduino'nun meşhur **Blink** eskizi iyi bir başlangıç olabilir. GP25 LED'i yanıp sönmeye başladıysa yükleme başarıyla tamamlanmış demektir.

UF2 Bootloader aracılığıyla yüklenen ilk eskizden sonra Akana R1'in Mbed OS yazılımının sağladığı bir **seri (COM) portu** da olacak. Sonraki eskizleri yükleyebilmek için doğru seri portu seçmiş olmak gerekiyor. Bunun için Arduino IDE menüsünde **Araçlar > Port** altından ilgili seri portu seçmek yeterli.

Artık Arduino IDE her eskiz yüklemesi öncesinde bu seri portu kullanarak Akana R1'i otomatik olarak UF2 Bootloader moduna alıyor ve yeni eskizi yüklüyor.

AKANA R1'İ ARDUINO MBED OS İLE KODLAMAK

Mbed OS'un halihazırda oldukça yetenekli olan RP2040 mikrokontrolcüsünün gücüne güç kattığından bahsetmiştik. Ancak Arduino ekibi, RP2040'ın diğer Arduino modellerinin çoğunda olmayan bazı özelliklerini Mbed OS üzerine kurduğu yazılımına eklemeyi tercih etmemiş.



arm MBED OS

Akana R1'in önemli özelliklerinden biri **I2C0** ve **I2C1** kanalları arasında seçim yapılabilmesini sağlayan **I2C kanalı** anahtarı. Böylece bağlı tüm bileşenleri tek seferde istenen I2C kanalına aktarmak mümkün olabiliyor. Ancak Arduino ekibi RP2040'ın ikinci I2C kanalını Mbed OS yazılımına dahil etmemiş. Bu yüzden I2C1 adındaki ikinci kanal bu yazılım ile **kullanılmıyor**.

Arduino Mbed OS'un sağladığı I2C0 kanalı Akana R1'in tasarımındaki GP4 ve GP5 pinlerinde kullanılabilir. I2C1 iletişim kanalını kullanmak gerekmiyorsa Akana R1 ile **Arduino Mbed OS** yazılımını kullanmanın bir sakıncası bulunmuyor.

Öte yandan Akana R1 üzerinde yer alan **SSD1306** çipli I2C OLED ekran için kullanılacak en yaygın Arduino kütüphanesi olan **Adafruit SSD1306**'nın da Mbed OS yazılımı ile sorunları var.

Bu iki sorun da yazılım ve kütüphane dosyaları üzerinde bir takım düzenlemeler yapılarak aşılabiliyor, ancak bu değişiklikler her güncellemede silindiği için her seferine tekrar yapmak gerekiyor.



Adafruit SSD1306 kütüphanesi yerine tercih edilebilecek bir diğer ekran kütüphanesi de **u8g2**. Topluluk tarafından geliştirilen bu kütüphane sadece Akana R1 üzerindeki SSD1306 çipli ekranları değil, diğer pek çok **tek renkli OLED ekranı** çalıştırmak için de kullanılabilir.

U8g2 kütüphanesine Arduino'nun dahili **kütüphane yöneticisi** üzerinden ulaşılabilir.

AKANA R1'e ARDUINO-PICO YÜKLEMEK

Earle Philhower'ın Raspberry Pi ekibinin yayınladığı **Pico SDK**'yı kullanarak hazırladığı **Arduino-Pico** isimli yazılım Arduino IDE'ye harici olarak eklenebiliyor. Bu yazılım temelde arkasında herhangi bir üretici bulunmayan bir topluluk projesi. Bu sayede çok destek görüyor ve hızlı adımlarla geliştirilmeye devam ediliyor.

Maker devi **Adafruit** de RP2040 temelinde geliştirdiği kartları için bu yazılımı öneriyor.

Bu yazılım, Arduino'nun tercihi olan **Mbed OS** yerine Arduino IDE'de kullandığımız **C++** programlama dili temelindeki **Pico SDK** üzerine kurulu olması nedeniyle çok az yer kaplayan kodlar üretebilmenin yanında çoğu zaman daha hızlı da olabiliyor.

Arduino-Pico yazılımı Arduino IDE'ye entegre olmadığı için farklı bir yöntemle yüklemek gerekiyor. Arduino IDE ana menüsünden **Dosya > Tercihler** seçerek açılan Tercihler penceresinde **Ek Kart Yöneticisi URL'leri** bölümünün sağındaki butona tıklayarak açılan pencerede **Yeni bir satıra geçerek** yazılımın kaynak bağlantısını yapıştırdıktan sonra **Tamam** tıklayarak Arduino IDE'nin kart yazılımı listesine Arduino-Pico'yu eklemek mümkün.

Kaynak bağlantısı;

https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json

Ardından yine Arduino IDE ana menüsünden **Araçlar > Kart > Kart Yöneticisi** seçerek açılan pencerede RP2040 aratarak çıkan listede Earle Philhower tarafından hazırlanmış **Raspberry Pi Pico/RP2040** yazılımını kurmak yeterli. Yazılımın **3.9.3** sürümü ile birlikte **Akana R1** desteği eklenmiştir.

Son olarak **Araçlar > Kart** bölümünden Raspberry Pi Pico/RP2040 altından **METE HOCA Akana R1** seçerek kart seçimi tamamlanır.

AKANA R1'İ ARDUINO-PICO İLE KODLAMAK

Arduino-Pico yazılımı RP2040'ın tüm özelliklerinin sorunsuz bir şekilde kullanabilmesini sağlıyor ve yaygın kullanılan donanım ve kütüphanelerle bir sorunu bulunmuyor.

Arduino-Pico'yu yükledikten sonra ilk eskizi yüklemek için Akana R1'i **UF2 Bootloader** moduna geçirmek gerek. Böylece RP2040 yeni yazılımın isteklerine cevap verebilir. İlk eskiz yüklendikten sonra geriye Arduino IDE'nin **Araçlar** menüsünde yeni oluşan **seri portu** seçmek kalıyor. Artık Akana R1 Arduino-Pico ile kodlanmaya hazır.

Arduino-Pico yazılımı RP2040'ın iki I2C kanalının da rahatlıkla kullanabilmesini sağlıyor. Varsayılan ayarlarda I2C0 kanalı GP4 (SDA) ve GP5 (SCL) olarak ayarlıyken, I2C1 kanalı ise GP26 (SDA) ve GP27 (SCL) olarak ayarlanmış durumda. Elbette analog girişler olan GP26 ve GP27 pinlerini I2C1 kanalı için harcamak mantıklı olmadığı için Akana R1 üzerinde bu kanal için **GP6 (SDA)** ve **GP7 (SCL)** pinleri kullanılıyor.

Neyse ki Arduino-Pico'da I2C kanalı pinlerini değiştirmek çok kolay. **setSDA** ve **setSCL** komutları ile istenen kanalın pinleri değiştirilebilir. Bunun için eskizin `void setup()` fonksiyonu içine en üste aşağıdaki satırları yerleştirmek yeterli;

```
Wire.setSDA(4);  
Wire.setSCL(5);  
Wire1.setSDA(6);  
Wire1.setSCL(7);
```

Bu şekilde I2C0 kanalı **Wire** ile, I2C1 kanalı da **Wire1** ile kullanılabilir. Yazılımın 3.9.3 versiyonu ile gelen Akana R1 desteği ile birlikte bu kodları eklemeye **gerek kalmamıştır**. Yeni yazılım versiyonu ile Akana R1 üzerindeki butonları pin numaraları ile uğraşmadan **BTN_UP**, **BTN_DOWN**, **BTN_LEFT**, **BTN_RIGHT**, **BTN_ENTER** ve **BTN_BACK** sabitleri ile kullanmak mümkün hale geldi.

Arduino-Pico yazılımının Akana R1 üzerindeki I2C OLED ekranı çalıştırmak için tercih edilebilecek **Adafruit SSD1306** kütüphanesi ile de herhangi bir sorunu bulunmuyor. Aşağıda kütüphanenin I2C1 kanalı ile (**Wire1**) kullanımındaki kurulum satırı görülüyor. I2C0 kanalı için ise sadece **Wire** kullanmak yeterli.

```
Adafruit_SSD1306 ekran(128, 64, &Wire1, -1);
```

Bir diğer ekran kütüphanesi olan **u8g2**'de ise her iki I2C kanalı için farklı kurulum satırları kullanmak gerekiyor. Aşağıda ilk satırda I2C0 kanalı için, ikinci satırda da I2C1 kanalı için kurulum satırı görülüyor. İkinci satırda görülen **2ND** kodu I2C1 kanalını, yani ikinci kanalı ifade ediyor.

```
U8G2_SSD1306_128X64_NONAME_F_HW_I2C ekran(U8G2_R0, U8X8_PIN_NONE);  
U8G2_SSD1306_128X64_NONAME_F_2ND_HW_I2C ekran(U8G2_R0, U8X8_PIN_NONE);
```

Arduino-Pico yazılımının bir takım ilginç özellikleri bulunuyor. Örneğin aşağıdaki kod satırını `void setup()` fonksiyonu içine yerleştirdikten sonra Akana R1'i **RESET** butonuna **çift** tıklayarak **UF2 Bootloader** moduna geçirmek mümkün;

```
rp2040.enableDoubleResetBootloader();
```

Akana R1'i kod üzerinden resetlemek veya UF2 Bootloader moduna geçirmek de mümkün. Bunun için de aşağıdaki satırdaki komutları istenen yere yerleştirmek yeterli.

```
rp2040.reboot(); // RESET  
rp2040.rebootToBootloader(); // UF2 Bootloader
```

AKANA R1'e MICROPYTHON YÜKLEMEK

MicroPython programlama dili Arduino'nun kodlama yöntemine göre çok farklı bir mantıkla çalışıyor. Öncelikle bu mantığı kavramak gerek.

Arduino IDE'de kullanılan kod yükleme ve çalıştırma sistemi, temelde kodun RP2040'ın anlayacağı dile, yani makine koduna dönüştürülmesi, bir UF2 dosyası haline getirilmesi ve ardından Akana R1'in seri port üzerinden gönderilen bir mesajla otomatik olarak UF2 Bootloader moduna alınarak UF2 dosyasının USB diske kopyalanması yöntemine dayanıyor.

MicroPython ise çok farklı bir yapıya sahip. Bu tür diller kodu derlemek yerine cihaz üzerine yerleştirilmiş ve **yorumlayıcı** adı verilen yazılımın aktarılan kodu adım adım yorumlayarak çalıştırması üzerine kurulu. Yani yazılan kod her değiştirildiğinde uzun süren derleme ve yükleme süreçlerine gerek yok. Kod Akana R1'e gönderildiği anda içinde bulunan MicroPython yorumlayıcı yazılımı ilk satırdan başlayarak kodu çalıştırmaya başlıyor.

MicroPython ile kodlamaya başlamadan önce Akana R1'e bu yorumlayıcı yazılımı yüklemek gerekiyor. Bunun iki yöntemi var. Birincisi Akana R1'i UF2 Bootloader moduna geçirdikten sonra **MicroPython web sitesinden Raspberry Pi Pico** için olan UF2 dosyasını indirip yüklemek.

MicroPython indirme linki;

https://micropython.org/download/RPI_PICO/

İkincisi ise daha sonra kodlama için de tercih edilebilecek **Thonny** adlı açık kaynak kodlu yazılımı kullanmak. Thonny ayar ekranında gerekli sürümü indirip yükleyebiliyor. Thonny uygulamasını farklı işletim sistemleri için aşağıdaki linkten indirmek mümkün. Yazılımın yükleme gerektirmeyen **portable** sürümü de bulunuyor.

Thonny indirme linki;

<https://github.com/thonny/thonny/releases/>

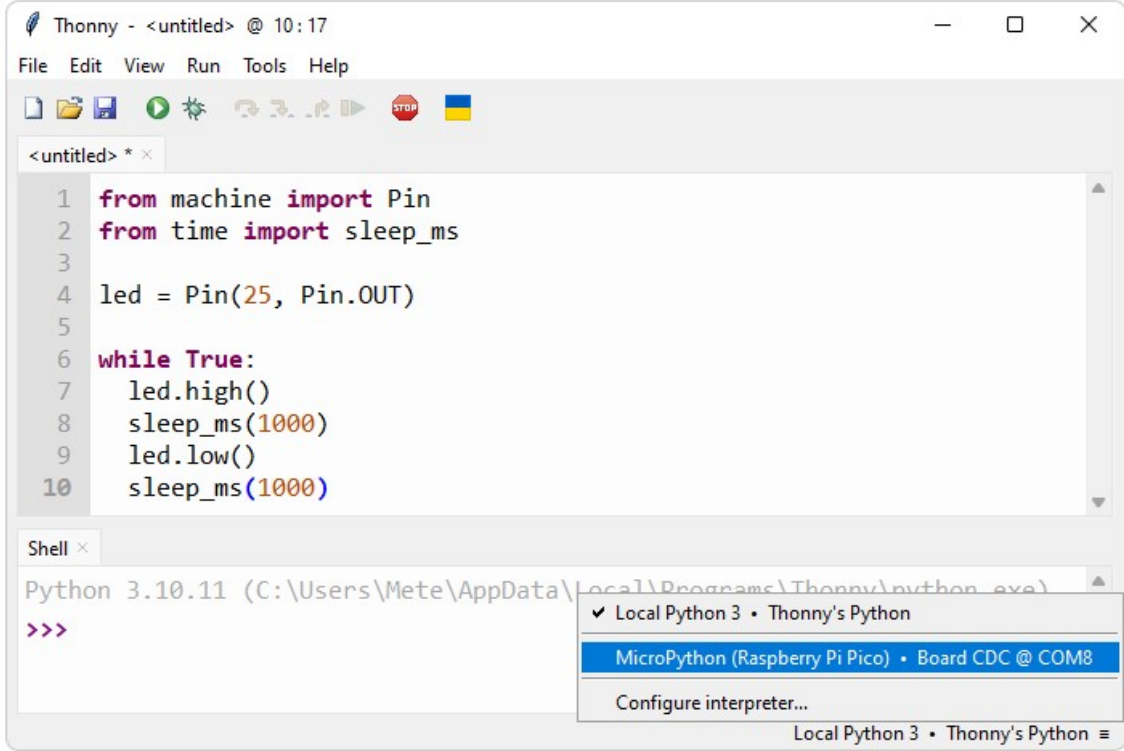
MicroPython yorumlayıcısını Thonny aracılığıyla yüklemek için Akana R1'i **UF2 Bootloader** moduna geçirdikten sonra Thonny ana menüsünde **Araçlar > Seçenekler** seçerek ilgili pencere açılır. **Yorumlayıcı** sekmesinde üst kısımdan **MicroPython (RP2040)** ve alt kısımdan **<Bağlantı noktasını otomatik olarak algılamayı dene>** seçerek mikrokontrolcü ve kodlanacak dil seçilir.

Ardından pencerenin sağ alt bölümünde yer alan **Install or Update MicroPython** linkine tıklayıp açılan yükleme penceresindeki **Target volume** alanında UF2 Bootloader modundaki Akana R1'in oluşturduğu **RPI-RP2** klasörünü seçtikten sonra alt bölümdeki **MicroPython family** bölümünde **RP2** seçili olduğundan emin olunmalı.

Variant olarak **Raspberry Pi Pico** seçtikten sonra alt bölümde preview uzantılı olmayan en yüksek sürümü seçip **Yükle** butonuna tıklayarak işlem başlatılır. Thonny seçilen MicroPython yorumlayıcısını indirip Akana R1'e yükleyecektir.

AKANA R1'İ MICROPYTHON İLE KODLAMAK

Akana R1'i MicroPython ile kodlamak için birkaç alternatif yazılım bulunuyor. Bunların en kullanıcı dostu olanları **Thonny**, **Mu Editor** ve Arduino'nun geliştirmeye devam ettiği **Arduino Lab for MicroPython**.



Yeni başlayanlar için Thonny ve Mu Editor önerilir. Arduino Lab for MicroPython ise henüz emekleme aşamasında, ileride çok kullanışlı olabilir. **Thonny**'nin başlangıç veya ileri seviye için en uygun seçenek olduğu söylenebilir.

Akana R1'e MicroPython yüklendiğinden emin olduktan ve Thonny'yi ilk kez açtıktan sonra sağ alt bölümdeki yazıya tıklayıp otomatik olarak bulunan donanımı seçmek ayarları tamamlamak için yeterli.

Yukarıdaki fotoğrafta Akana R1 üzerindeki GP25 LED'ini yakıp söndüren MicroPython kodu görülüyor. Kodu yazıp menüdeki yeşil renkteki çalıştırma butonuna tıklayınca kod Akana R1'e gönderilir ve LED yanıp sönmeye başlar. Python dilinde kod ayırımı için {parantezler} bulunmadığı için alt bölümdeki kodları **Tab** tuşuyla sağa kaydırarak belirtmek gerektiğini unutmamak gerek.

```
from machine import Pin
from time import sleep_ms
```

```
led = Pin(25, Pin.OUT)
```

```
while True:
    led.high()
    sleep_ms(1000)
    led.low()
    sleep_ms(1000)
```

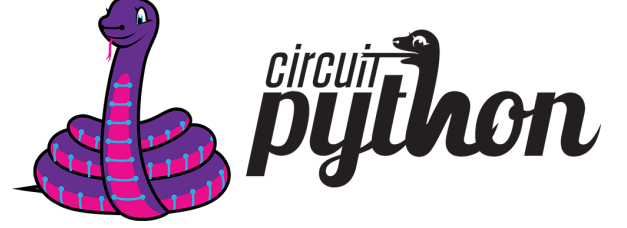
AKANA R1'e CIRCUITPYTHON YÜKLEMEK

CircuitPython, maker üretici devi **Adafruit** tarafından geliştirilen ve büyük oranda MicroPython üzerine kurulu bir yazılım. Ancak kod yapısı, çalışma mantığı ve kütüphane uyumluluğundaki farklarla MicroPython'dan oldukça ayrılıyor.

MicroPython'da kodlama, dosya aktarımı ve çalıştırma işlemleri UF2 yazılımının sağladığı seri port üzerinden yapılırken, CircuitPython yüklü geliştirme kartları bilgisayarda sürekli olarak bir **USB depolama diski** gibi görünür.

Kartlara yüklenen kodlar ve kütüphaneler de bilgisayarın dosya yöneticisi ile rahatlıkla ulaşılabilecek şekilde bu USB diskin içinde duruyor.

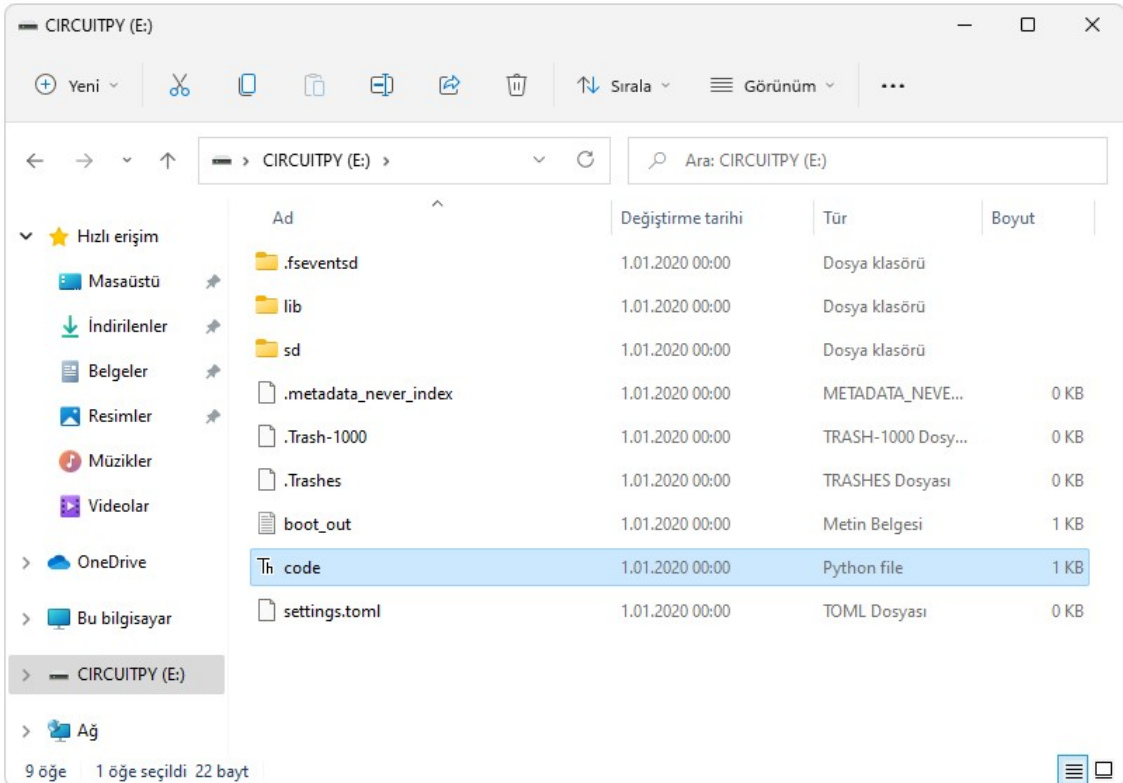
CircuitPython ile kodlamaya başlamadan önce Akana R1'e yorumlayıcı yazılımı yüklemek gerekiyor. Akana R1'i UF2 Bootloader moduna geçirdikten sonra **CircuitPython web sitesinden Raspberry Pi Pico** için olan UF2 dosyasını indirip yüklemek yeterli.



CircuitPython indirme linki;

https://circuitpython.org/board/raspberry_pi_pico/

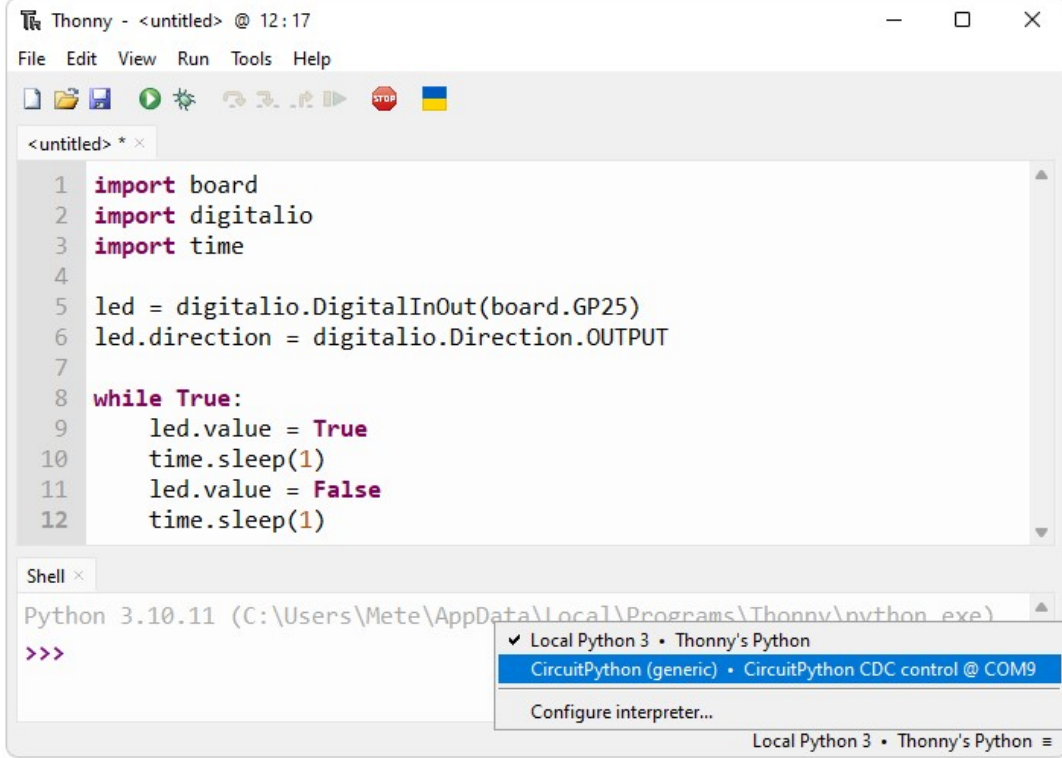
Yazılım yüklendikten sonra bilgisayarda **CIRCUITPY** adında bir USB disk sürücüsü görünecek. Her şey bu sürücünün içinde. Sürücü içindeki **code.py** dosyası Akana R1'e güç verildiğinde otomatik olarak çalışacak olan kod dosyası. Bu dosyayı herhangi bir metin düzenleyici veya IDE ile açıp düzenlemek mümkün. Dosya kaydedildiğinde Akana R1 otomatik olarak yeniden başlar ve kaydedilmiş kodu çalıştırır.



AKANA R1'i CIRCUITPYTHON İLE KODLAMAK

CircuitPython yorumlayıcısı yüklenmiş Akana R1'i kodlamak, oluşturduğu disk sürücüsü içindeki **code.py** dosyasını düzenleyip kaydetmek kadar kolay. Elbette bir IDE yazılımı kullanmak yararlı olacaktır.

Adafruit, CircuitPython için başlangıç seviyesinde Mu Editor'u öneriyor. Ancak bu yazılım kodlamada ilerledikçe yetersiz kalmaya başlayacaktır. Bu yüzden MicroPython için de önerilen **Thonny** kullanmak daha iyi bir tercih olabilir.



Thonny'yi CircuitPython moduna ayarlamak için yukarıdaki resimde sağ alt bölümde de görülen bölüme tıklayıp otomatik olarak bulunan donanımı seçmek yeterli.

Görüldüğü gibi CircuitPython'un kod yapısı MicroPython'a göre biraz değişik. machine yerine **board** ve **digitalio** kullanılıyor ve pin ayarlama komutları da farklı.

```
import board
import digitalio
import time
```

```
led = digitalio.DigitalInOut(board.GP25)
led.direction = digitalio.Direction.OUTPUT
```

```
while True:
    led.value = 1
    time.sleep(1)
    led.value = 0
    time.sleep(1)
```

SÜRÜMLER

Belge Sürümleri

| Tarih | Değişiklik |
|-----------------|--|
| 20 Haziran 2024 | Arduino-pico paketine eklenen Akana R1 desteği ile ilgili güncellemeler yapıldı. |
| 16 Mayıs 2024 | İlk sürüm |

