



SCRAPY Priručnik



Co-funded by
the European Union

Potpore Europske komisije za izradu ove publikacije ne predstavljaju potvrdu sadržaja koji odražava samo stajališta autora te se Komisija ne može smatrati odgovornom za bilo kakvu uporabu informacija sadržanih u njoj.

UPOZORENJE

Prilikom korištenja ovog priručnika i SCRAPY KIT-a, molimo da obratite pozornost na sljedeće točke upozorenja:

1. SCRAPY KIT sadrži mnogo sitnih dijelova. Gutanje ili nepravilno rukovanje može uzrokovati ozbiljne infekcije i smrt. U slučaju nesreće odmah potražite liječničku pomoć.
2. Strogo zabranjena uporaba proizvoda i njegovih dijelova u blizini izlaza za izmjeničnu struju ili drugih strujnih krugova u slučaju mogućeg rizika od strujnog udara.
3. Strogo zabranjena uporaba ovog proizvoda u blizini tekućina ili vatre.
4. Vodljive materijale držati podalje od ovog proizvoda.
5. Ne dopuštajte djeci mlađoj od 3 godine da koriste ovaj proizvod bez nadzora odrasle osobe. Molimo da ovaj proizvod držite na mjestu koje nije dostupno djeci mlađoj od 3 godine.
6. Ne skladištiti niti upotrebljavati ovaj proizvod u ekstremnim uvjetima kao što su ekstremna hladnoća ili vrućina, visoka vlažnost, pod izravnom sunčevom svjetlošću itd.
7. Ne zaboravite prekinuti strujni krug kada nije u upotrebi.
8. Neki dijelovi ovog proizvoda mogu postati topli na dodir kada se upotrebljavaju u određenim dizajnima strujnih krugova, što je uobičajeno.
9. Nepravilna upotreba može uzrokovati pregrijavanje.
10. Korištenje komponenta koje nisu u skladu sa specifikacijom može uzrokovati oštećenje proizvoda.

Uvod

„SCRAPY KIT“, komplet je izrađen na temelju upotrebe mikrokontrolera Raspberry Pi Pico. „SCRAPY komplet uspostavlja se u okviru projekta sufinanciranog od Erasmus+ programa.

Novi trend edukacije na daljinu, koja je posljedica pandemije bolesti COVID-19, predstavlja jaz u poučavanju i prakticiranju aktivnosti fizičkog računalstva do učenika koji su izloženi riziku od slabih rezultata u takvim STEM predmetima.

Cilj kompleta je da podupire i promiče praktično učenje u računalstvu i programiranju u slučajevima poučavanja na daljinu i u učionicama. Komplet pruža kombinaciju fizičkih računalnih načela i programiranja s hardverom i softverom, što dovodi do inovativnog iskustva učenja.

Opseg ovog priručnika je:

- Obavijestiti vas o ključnim komponentama i upotrebi glavnih elektroničkih elemenata.
- Voditi vas korak po korak kako biste učinkovito sastavili komplet, uzimajući u obzir odgovarajuće mjere opreza.
- Pružiti upute za uporabu komponenta i vezu s Pico mikrokontrolerom.
- Pružiti upute s funkcijama i opsegom svake elektroničke komponente.

Uživajte u čitanju i zabavite se kroz praktičnu praksu i eksperimentiranje sa „SCRAPY“ kompletom

Tablica sadržaja

Uvod.....	1
Tablica sadržaja.....	2
Uključeno u „SCRAPY“ kompletu	4
Objašnjenje sastavnih dijelova	6
1. Što je breadboard testna ploča?.....	6
2. Što je otpornik?	7
3. Što je električni kondenzator?.....	10
4. Što je dioda?	10
5. Što su prenosni kabovi?	11
Priprema projekta	12
Montaža kompleta	19
Osnovne upute.....	24
0. “Pozdrav „SCRAPY“ ljudi!”	24
4. Kontrola LED diode.....	26
5. Tipkalo	28
6. Zujalica.....	30
7. Potencijometar	32
8. RGB LED	34
Napredne upute	36
9. LDR Fotootpornik.....	37
10. Servo motor	39
11. OLED I2C SSD1306 zaslon.....	41
12. „Joystick“ modul	45
Lekcije sa senzorima	47

13.	Senzor kapi kiše.....	47
14.	HC-SR04 Ultrazvučni senzor	49
15.	PIR Detektor pokreta	52
16.	DHT11 senzor.....	54
17.	SW-420 Senzor vibracije	56
18.	Detektor plamena.....	58
19.	Senzor za otkrivanje zvuka	60
20.	Detektor vlage tla	62
21.	Infracrveni IR senzor.....	64
DODATAK: MicroPython tablica sažetka		66

Uključeno u „SCRAPY“ kompletu

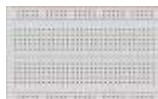
GPIO Tiskana pločica
Raspberry Pi Pico
(2kom.)



MB-102 Napajački modul
(1kom.)



Bijela breadboard testna
ploča 830 rupica(1kom.)
+ 400 rupica (1kom.)



Tipkalo (1kom.) i
Poklopac za gumb
(1kom.)



Zujalica (1kom.)



Otpornik 220 Ohm
(5kom.) + 1000 Ohm
(5kom.)



LDR fotootpornik
(1kom.)



100uF električni
kondenzator (1kom.)



LED diode 3mm Plava
(1kom.), Zelena (1kom.),
Crvena(1kom.) Žuta
(1kom.)



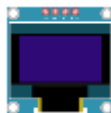
RGB LED 5mm (1kom.)



SG90 servo motor
(1kom.)



OLED I2C ICC (1kom.)



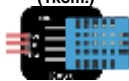
HC-SR04 ultrazvučni
senzor (1kom.)



PIR Detektor pokreta HC-
SR501 (1kom.)



DHT11 Senzor digitalne
temperature i vlažnosti
(1kom.)



Senzor kapi kiše (1kom.)


 Rotacijski potencijometar
linearni B1k Ohm
(1kom.)

 SW-420 senzor vibracija
(1pc)

 Detektor plamena
(1kom.)

 Senzor otkrivanja zvuka
visoke osjetljivosti
(1kom.)

 Higrometar tla / detektor
vlage tla (1 kom.)

 Modul infracrveni IR
senzora KY-032 (1kom.)

 "Joystick" modul
(1kom.)

 USB na micro-USB kabel
1m (1kom.)

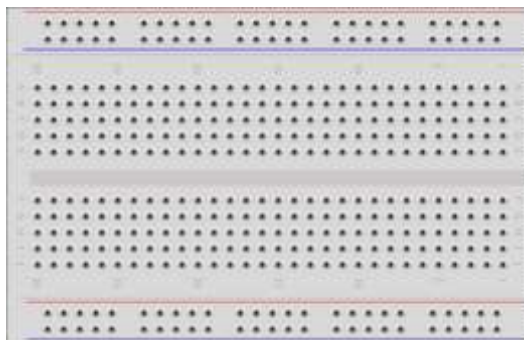
 6 x AA 1.5V baterija
(1kom.) + držač (1kom.)


Premosni kabel (6 kom.)


 Plastični vijci (10 kom.) ,
plastične matice (6
kom.), plastični stupići
(10 kom.)


Objašnjenje sastavnih dijelova

1. Što je breadboard testna ploča?



Breadboard testna ploča je plastična ploča sa sitnim rupama koje omogućavaju jednostavno umetanje elektroničkih komponenti (prijenosnika, otpornika, čipova itd.) kako bi se izradio prototip elektroničkog strujnog kruga. Unutrašnjost se sastoji od nizova sitnih metalnih kopči za držanje vodova koje treba spojiti.

Većina breadboard testnih ploča ima nizove brojeva, slova te plus/minus znakova koji su napisani na njima. Svrha oznaka je da vam pomognu locirati određene rupe na breadboard testnoj ploči kako biste mogli slijediti upute pri izgradnji strujnog kruga.

Duge trake na dvije strane breadboard testne ploče su obično označene crvenom i plavom bojom ili crvenom i crnom bojom te znakovima plus (+) i minus (-). Ti se redovi nazivaju „busovima“ ili vodovima i obično se upotrebljavaju za napajanje strujnog kruga električnom energijom kad je priključen na napajanje (baterija ili vanjski izvor napajanja).

Pozitivan “bus” označen je crvenom bojom, ima znak plus (+) i osigurava struju.

Negativni “bus” je označen plavom ili crnom bojom, ima znak minus (-) i osigurava uzemljenje.

Prednosti uporabe breadboard testne ploče:

- Olakšava brzu provjeru jednostavnih i složenih strujnih krugova te laku provjeru strujnih krugova u njihovoj početnoj fazi.
- Lako prilagodljiva.
- Fleksibilna.
- Nije potrebno bušenje rupa.
- Nije potrebno lemljenje.
- Jednostavno otklanjanje neispravnosti strujnih krugova i programa.

2. Što je otpornik?



Otpornik je malo pakiranje otpora. Njegovom upotrebom u strujnom krugu, struja se smanjuje za točno određen iznos. Za određivanje otpora otpornika postoji uzorak obojenih traka.

Color	1st Band	2nd Band	3rd Band (5-Band Only)	Multiplier (3rd or 4th Band)	Tolerance (Last Band)
Black	0	0	0	1	
Brown	1	1	1	10	± 1%
Red	2	2	2	100	± 2%
Orange	3	3	3	1000	
Yellow	4	4	4	10000	
Green	5	5	5	100000	± 0.5%
Blue	6	6	6	1000000	± 0.25%
Violet	7	7	7	10000000	± 0.1%
Grey	8	8	8		± 0.05%
White	9	9	9		
Gold				0.1	± 5%
Silver				0.01	± 10%
None					± 1%

(Image credit: Future Owns) available at <https://www.tomshardware.com/how-to/resistor-color-codes>

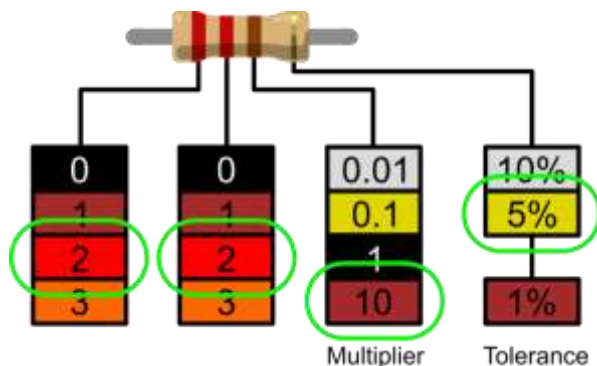
Česte boje kodova otpornika i njihova uporaba:

Vrsta otpornika	Boja koda s 4 vrpce	Boja koda s 5 vrpca	Česta uporaba
220Ohm	Crvena-Crvena-Smeđa-Zlatna	Crvena-Crvena-Crna-Crna-Zlatna	Zaštita LED svjetlosti
1000 Ohm (1Kiloohm)	Smeđa-Crna-Crvena-Zlatna	Smeđa-Crna-Crna-Smeđa-Zlatna	Zaštita LED svjetlosti, Razdjelnik napora

Otpornici nemaju polarnost kako bi se mogli koristiti u bilo kojoj orijentaciji u strujnom krugu. Međutim, kako bismo utvrdili točne vrijednosti kodova u boji, moramo razumjeti značenje obojenih traka na otporniku.

Na tipičnom amaterskom otporniku s 4 vrpce, u skupini se nalaze tri boje. Ovo su prva i druga značajna brojka te množitelj. Konačna vrpca je tolerancija otpornika, ona služi kao granica pogreške. Za većinu amatera, tolerancija od 5% (zlatna) je savršena i uobičajena.

Otpornik 200 Ohm (4 vrpce)

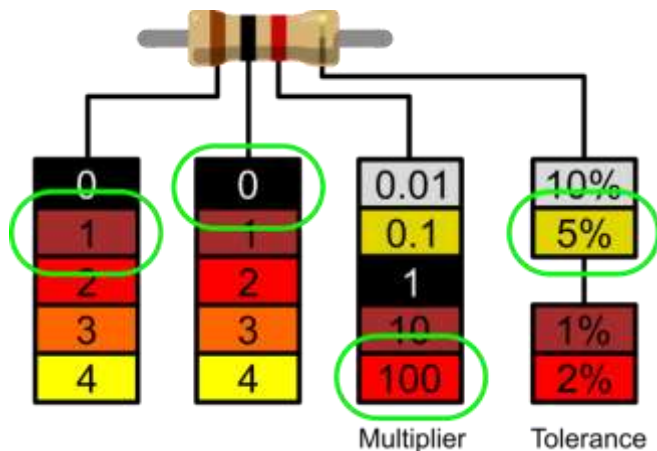


(Image credit: Future) available at <https://www.tomshardware.com/how-to/resistor-color-codes>

1. Prva je značajna brojka crvena, a upotrebom dekodera možemo vidjeti da njena vrijednost iznosi 2.
2. Druga značajna brojka je također crvena, što nam daje 22.
3. Množitelj je smeđe boje te se dekodira u 10. Ako pomnožimo 22 s 10, dobivamo 220.
4. Zadnja vrpca, tolerancija, je zlatna. Zlatna vrpca ima vrijednost 5%, što znači da možemo prihvatiti otpor koji odstupa 5%.

Za izrađivače kojima je potrebna veća preciznost postoje i otpornici s pet vrpca koji imaju i treću značajnu brojku. Ta dodatna brojka pruža jasnoću koja može biti ključna u strujnim krugovima koji su osjetljivi na otpornost, primjerice u znanstvenim i inženjerskim instrumentima.

Otpornik 1000 Ohm (4 vrpce)



(Image credit: Tom's Hardware) available at <https://www.tomshardware.com/how-to/resistor-color-codes>

1. Prva linija je smeđa, a koristeći dekoder može vidjeti da je njena vrijednost 1.
2. Druga linija je crna, što nam daje 10.

3. Množitelj je crvene boje te se dekodira u 10. Ako pomnožimo 10 s 100, dobivamo 1000.
4. Zadnja vrpca, tolerancija, je zlatna. Zlatna vrpca ima vrijednost 5%, što znači da možemo prihvatiti otpor koji odstupa 5%.

3. Što je električni kondenzator?



Električni kondenzator je naprava koja pohranjuje električnu energiju u električnom polju. Riječ je o pasivnoj elektroničkoj komponenti sastavljenoj od dva terminala. Sastoji se od dva odvojena električna vodiča. Prostor između tih vodiča može biti ispunjen vakuumom ili dielektričnim izolacijskim materijalom. (Wikipedia)

Kondenzator od 100uF je elektrolitski kondenzator za odvajanje. Ovi kondenzatori su izvrsni prigušivači prenapona i uporabom jednog između energije i uzemljenja strujnog kruga osigurava se nesmetani dovod energije.

4. Što je dioda?



Dioda je dvoterminalna elektronička komponenta koja provodi struju prvenstveno u jednom smjeru (asimetrična vodljivost); ima nizak (idealno nikakav) otpor u jednom smjeru i visok (idealno beskonačan) otpor u drugom.

Najčešća je funkcija diode omogućiti prolaz električne struje u jednom smjeru (smjer naprijed) dok ga blokira u drugom smjeru (obrnuti smjer). Kao takva, dioda se može smatrati električnom

verzijom nepovratnog ventila. Ovo jednosmjerno ponašanje naziva se rektifikacijom i upotrebljava se za pretvaranje izmjenične struje (ac) u istosmjernu struju (dc). Kao rektifikatori, diode se mogu koristiti za zadatke poput izvlačenja modulacije iz radio signala u radijskim prijammnicima. (Wikipedia <https://en.wikipedia.org/wiki/Diode>)

5. Što su prenosni kabovi?



Prenosni kabovi su, jednostavno rečeno, kabovi koji imaju priključke na svakom kraju što im omogućuje da se koriste za spajanje dviju točaka bez lemljenja. Premosni kabovi su tipično korišteni s breadboard testnim pločama i drugim alatima za izradu prototipa kako bi se prema potrebi olakšala izmjena strujnih krugova. Razlika u boji kablova može služiti kao indikator različitih vrsta priključaka, kao što su uzemljenje ili energija.

Premosni kabel obično dolaze u tri verzije: muški, muški-prema-ženskom i ženski. Razlikuju se po krajnjoj točki kabla. „Muški“ krajevi kablova imaju iglu te se mogu spojiti u nešto, dok „ženski“ krajevi nemaju iglu i koriste se da se spoje u nešto. Muški kablovi su najčešći. Pri povezivanju dva priključaka na breadboard testnu ploču, većinom nam je potrebna muška žica. (<https://blog.sparkfuneducation.com/what-is-jumper-wire>)

Priprema projekta

1. Pročitaj prije korištenja

OBAVIJEST: Budući da sljedeći eksperimenti svi koriste strujni krug, pogrešno spajanje ili kratki spoj mogu oštetiti vašu Pico razvojnu pločicu. Molimo da prije priključenja napajanja uvijek ponovno provjerite strujni krug.

2. Raspberry Pi Pico

Ovo je Raspberry Pi Pico:

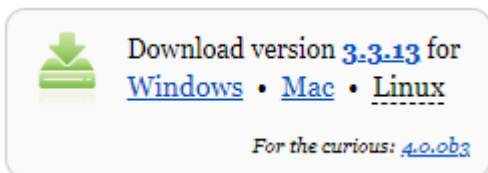


Priključite micro-USB kabel u otvor na lijevoj strani ploče.



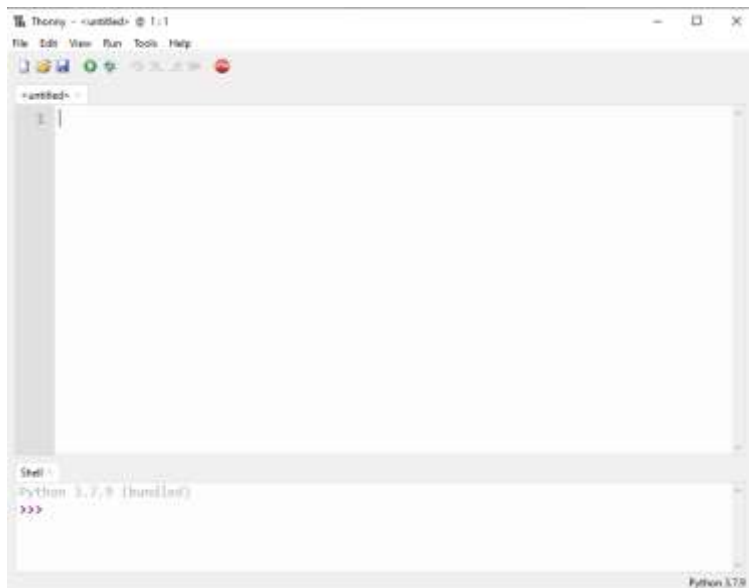
3. Instalirajte Thonny IDE

Posjetite web-stranicu <https://thonny.org> i odaberite odgovarajući operativni sustav. Slijedite upute za dovršetak instalacije.



U ovom priručniku, sve upute su programirane u Windows 10, koristeći Raspberry Pi Pico i odgovarajući firmware.

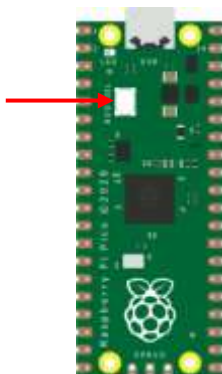
Nakon završetka instalacije, otvorite Thonny na svojem računalu.



4. Instalacija firmwarea

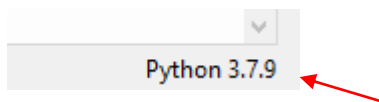
Raspberry Pi Pico može se programirati pomoću varijante Pythona koja se zove MicroPython. Da biste mogli koristiti MicroPython na Pico-u, prvo morate instalirati njegov firmware.

Korak 1: Pronađite BOOTSEL tipku na svojem Raspberry Pi Pico uređaju.

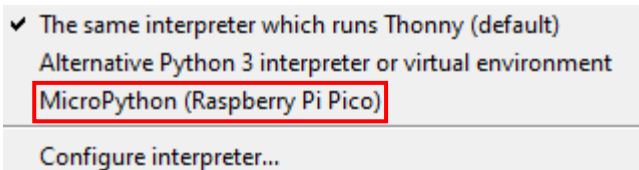


Korak 2: Pritisnite gumb BOOTSEL i držite ga dok drugi kraj micro-USB kabela povežete sa svojim računalom.

Korak 3: U donjem desnom kutu Thonnya vidjet ćete verziju Pythona koju trenutno koristite.

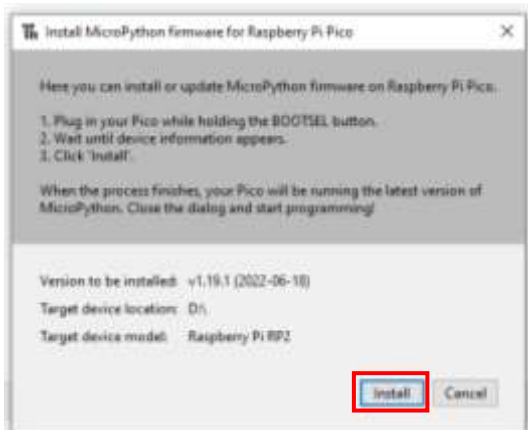


Kliknite na Python verziju i odaberite MicroPython (Raspberry Pi Pico)

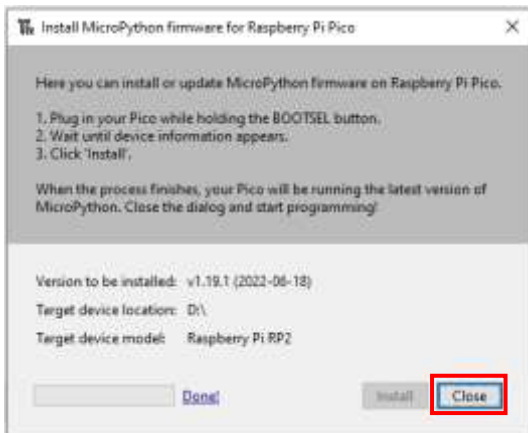


Ako ne vidite ovu opciju, provjerite ako je kabel ispravno spojen na Pico i/ili na vaše računalo.

Korak 4: Pojavit će se dijaloški okvir u kojem će se od vas tražiti da instalirate najnoviju firmware verziju Pico-a. Kliknite na „**Install**“ kako biste kopirali firmware u vaš Pico.



Korak 5: Pričekajte da se instalacija dovrši i kliknite „Close“.



Ne morate ponavljati ovaj postupak svaki put kada spajate Raspberry Pi Pico na svoje računalo, stoga ga sljedeći put samo priključite i možete krenuti s radom.

5. Uvod u MicroPython programiranje

Sada ćete upotrijebiti Thonny IDE kako biste pokrenuli jednostavni Python kod te kako biste se upoznali s Thonny's Shell i MicroPythonom.

Prvo se pobrinite da je vaš Raspberry Pi Pico spojen na vaše računalo te da ste odabrali MicroPython funkciju kako je objašnjeno u prethodnom odjeljku.

Shell okvir na dnu Thonny uređivača trebao bi izgledati ovako:



```

Shell

MicroPython v1.19.1 on 2022-06-10; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> |
    
```

Thonny je spreman komunicirati s vašim Pico koristeći REPL (čitaj-procijeni-ispisaj petlja) framework, koji vam omogućuje da napišete kod direktno u Shell i dobijete output.

Upišite sljedeću naredbu:

```
print("Hello!")
```

Zatim pritisnite tipku Enter i vidjet ćete sljedeći output.



```

Shell

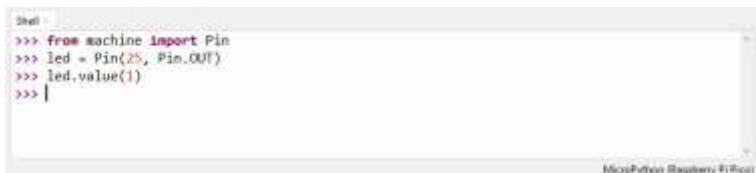
MicroPython v1.19.1 on 2022-06-10; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> print("Hello!")
Hello!
>>>
    
```

MicroPython vam omogućuje dodavanje modula specifičnih za hardver kao što su `machine`, koje možete koristiti u svojem programu Pico. U sljedećem primjeru upotrebljavat ćete `machine` modul kako biste uključili LED diode Pico-a.

Napišite sljedeći kod u Thonny's Shell:

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

```
led.value(1)
```



```

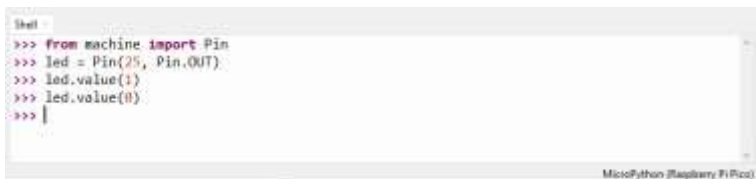
Shell
>>> from machine import Pin
>>> led = Pin(25, Pin.OUT)
>>> led.value(1)
>>> |
    
```

Pritisnite Enter tipku i LED diode Pico-a će se odmah uključiti.



Kako biste isključili LED diode, upišite sljedeći kod:

```
led.value(0)
```



```

Shell
>>> from machine import Pin
>>> led = Pin(25, Pin.OUT)
>>> led.value(1)
>>> led.value(0)
>>> |
    
```

U ostatku ovog odjeljka, napisati ćete svoj prvi „pravi“ program kojim će LED dioda treptati svaki put kada pokrenete svoj program.

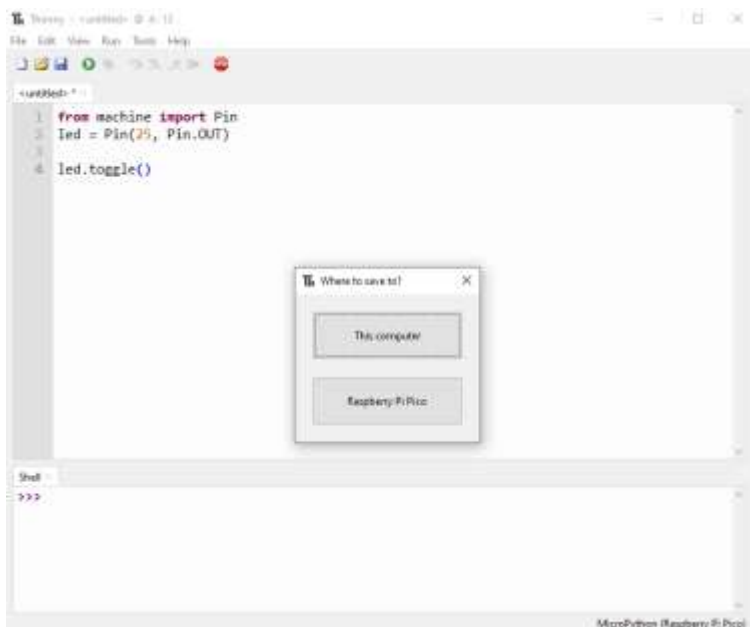
Thonny Shell koristan je za izvođenje brzih naredbi i osigurava da sve radi ispravno. Međutim, dulji programi trebali bi se spremiti u .py datoteku. Koristeći Thonny, možete spremiti programe direktno u Raspberry Pi Pico i zatim ih pokrenuti.

Otvorite Thonny Python i na glavnom oknu uređivača upišite sljedeći kod:

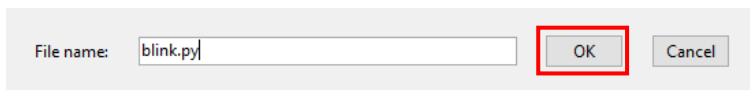
```
from machine import Pin
```

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

Sada, spremite svoj program klikom na „Save“ na gornjoj lijevoj strani ili pritisnite Ctrl+S na vašoj tipkovnici.



Thonny će vas upitati gdje želite spremiti vaš program. Odaberite **Raspberry Pi Pico**. Spremite datoteku kao `blink.py` i kliknite **OK**. Uvijek morate dodati `.py` proširenje tako da Thonny prepozna datoteku kao Python datoteku.



Svaki put kada pritisnete na „Play“, trebali biste vidjeti LED diodu kako se uključuje i isključuje.

Da vaš kod odvedete jedan korak dalje, možete učiniti da LED diode trepću određenom brzinom.

Upišite sljedeći kod i spremite program koristeći isto ime kako je prethodno navedeno.

```
from machine import Pin
from time import sleep
led = Pin(25, Pin.OUT)
while True:
    led.toggle()
    sleep(1)
```

Sada, kada pokrenete program, LED dioda će treptati svaku sekundu dok ne zaustavimo program. Da zaustavite program možete kliknuti „STOP“ ili pritisnuti Ctrl+C na vašoj tipkovnici.

U budućim uputama, naučit ćemo kako dodati i kontrolirati ostalu elektroniku i senzore kako bismo mogli stvoriti programe koji mogu međusobno komunicirati.

Montaža kompleta

SCRAPY komplet sastoji se od sljedećih komponenti:

- 1x 3D printani komad
- 1 x komad pleksiglasa
- 1 x Raspberry Pi Pico
- 1 x GPIO tiskana pločica
- 1 x Breadboard testna ploča (830 rupica)
- 10 x Plastičnih vijaka
- 6 x 12mm plastičnih stupića
- 4 x 6mm plastičnih stupića
- 6 x plastičnih matica



Postupak sastavljanja je jednostavan i može se dovršiti u 6 koraka:

Korak 1: Montirajte plastične stupiće veličine 6 x 12mm koristeći 6 plastičnih matica na komad pleksiglasa.

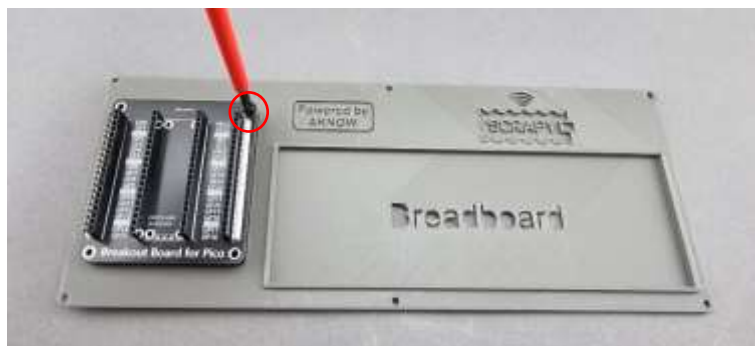




Korak 2: Na 3D isprintani komad, postavite 4 x 6mm plastične stupiće na “GPIO Breakout Board” dio.



Korak 3: Montirajte GPIO tiskanu pločicu na 4 stupića koristeći 4 plastična vijka.



Korak 4: Montirajte 3D printani komad s komadom pleksiglasa tako da stavite 6 plastičnih vijaka na 6 plastičnih stupića koje ste montirali u prvom koraku.





Korak 5: Odljepite zaštitnu vrpcu koja se nalazi ispod breadboard testne ploče s 830 rupica i postavite ju na "Breadboard" dio kompleta. Molimo pobrinite se da je pozitivna (+) strana ujedno i gornja strana, kako pokazuje sljedeća slika.



Korak 6: Postavite Raspberry Pi Pico na GPIO tiskanu pločicu i pritisnite sve dok sve GPIO igle nisu ispravno umetnute. Pobrinite se da je micro-USB utor na gornjoj strani kako slika pokazuje.



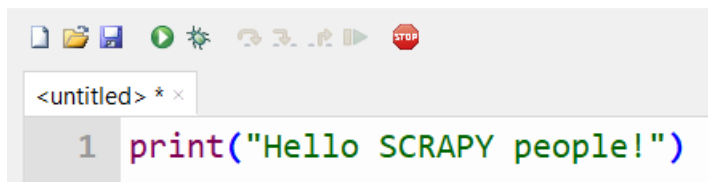
Osnovne upute

0. “Pozdrav „SCRAPY“ Ijudi!”

U ovoj osnovnoj lekciji, naučit ćemo kako printati jednostavnu poruku u Thonny-u koristeći Python programiranje.

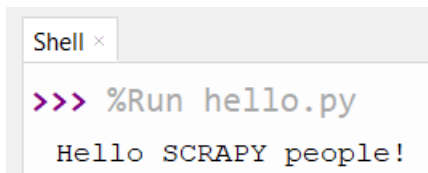
Print() funkcija

1. Uđite u Thonny uređivač, napišite sljedeći kod i pritisnite „play“. Thonny će vas upitati da prvo spremite program. Spremite ga pod nazivom hello.py.



```
<untitled> * x
1 print("Hello SCRAPY people!")
```

2. Provjerite shell prozor.



```
Shell x
>>> %Run hello.py
Hello SCRAPY people!
```

3. Dobar posao! Upravo ste kreirali vaš prvi Python program.

Print funkcija je ugrađena Python funkcija koja nam dopušta da printamo tekst u shell. Također može uzeti parametre. Kreirajte novi program i kopirajte sljedeći kod, zatim pritisnite „play“ i gledajte kako se tekst pojavljuje u shellu.

```

hello.py
1 print(1,2,3,4,5) #This is a comment!
2 print("I am ",2,"awesome") #1 line
3 print("Python is")      #1 line
4 print("amazing")       #1 line
5 print("I cant wait.....\n to learn more") # 2 lines of output!
    
```

Kao što možete vidjeti u shell prozoru, svaka print funkcija printa tekst u zasebnom retku. Međutim, ako koristite „\n“ (znak novog retka) možete promijeniti redak u istoj ispisnoj izjavi.

```

Shell x
>>> %Run hello.py

1 2 3 4 5
I am 2 awesome
Python is
amazing
I cant wait.....
  to learn more
    
```

Vježba

Iskoristite print funkciju da printate 3 zasebna retka „Goodbye SCRAPY people!“ koristeći samo jednu ispisnu izjavu.

4. Kontrola LED diode

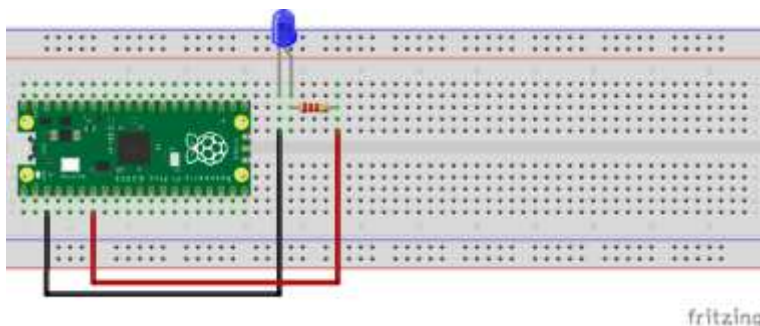
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati LED svjetlo. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `led.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x LED (bilo koje boje)
- 1 x Pico komplet za breadboard testnu ploču
- 2 x Muški prenosni kabel
- 1 x 220 Ohm otpornik

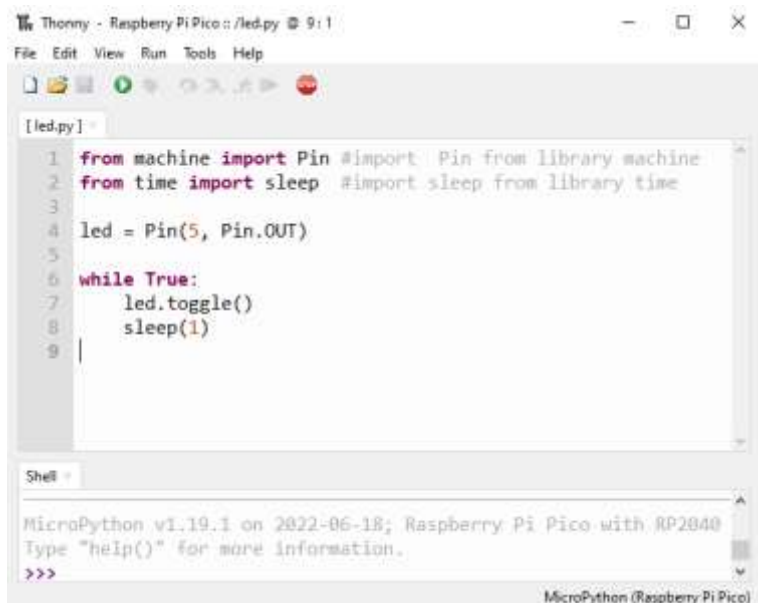
Dijagram ožičenja



- spojite duži kraj (+) LED diode na 220Ohm otpornik
- spojite otpornik na GPIO5 (crveni kabel)
- spojite kraći (-) LED diode na GND iglu

Kod

MicroPython kod za lekciju:



```

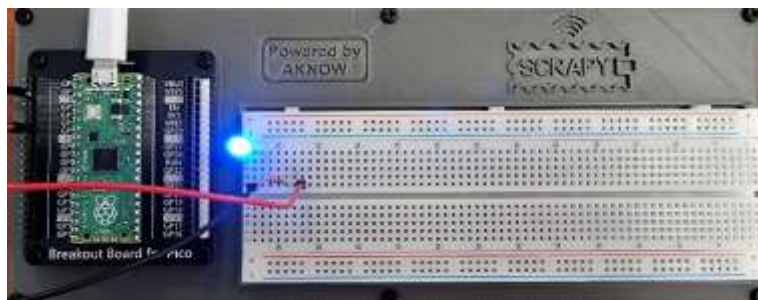
Thonny - Raspberry Pi Pico :: /led.py @ 9:1
File Edit View Run Tools Help

[led.py]
1 from machine import Pin #import Pin from library machine
2 from time import sleep #import sleep from library time
3
4 led = Pin(5, Pin.OUT)
5
6 while True:
7     led.toggle()
8     sleep(1)
9

Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
MicroPython (Raspberry Pi Pico)
    
```

Primjer

Slika kako strujni krug izgleda koristeći priloženi hardver:



5. Tipkalo

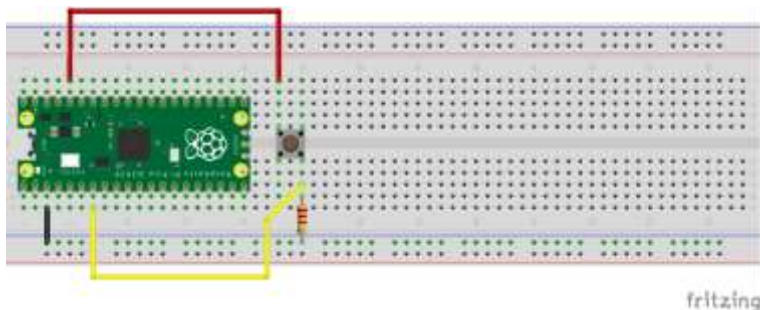
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati tipkalo. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `button.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Tipkalo (bilo koje boje)
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x 220 Ohm otpornik
- 1 x Poklopac za gumb

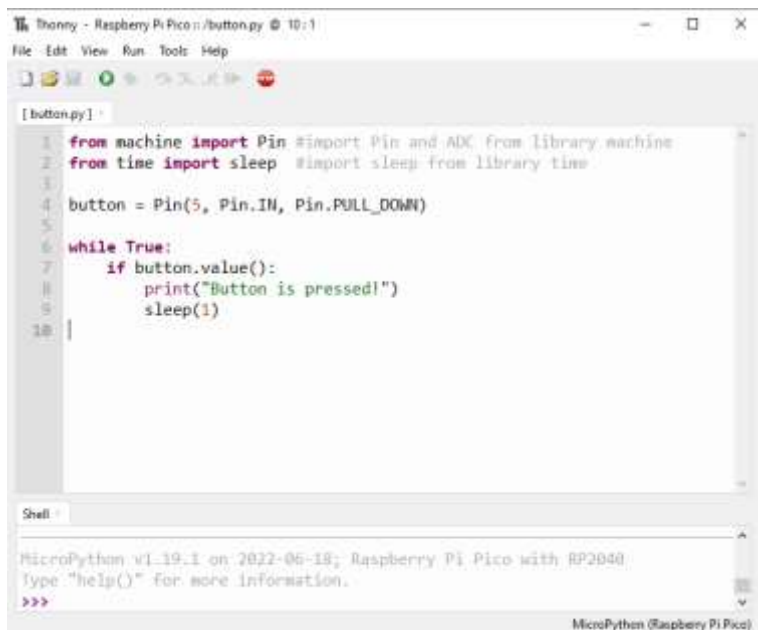
Dijagram ožičenja



- spojite gornju lijevu stranu tipkala na 3v3 iglu (crveni kabel)
- spojite donju desnu stranu tipkala na GPIO5 (žuti kabel)
- spojite GND iglu na (-) vod (crni kabel)
- spojite 220 Ohm otpornik na (-) vod i desnu gornju stranu tipkala

Kod

MicroPython kod za lekciju:



```

Thonny - Raspberry Pi Pico ::/button.py @ 10:1
File Edit View Run Tools Help

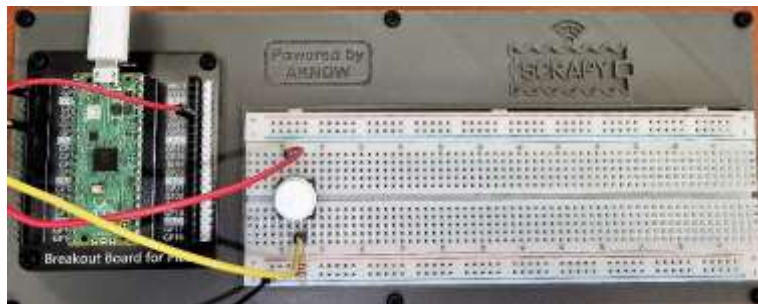
[button.py]
1 from machine import Pin #import Pin and ADC from library machine
2 from time import sleep #import sleep from library time
3
4 button = Pin(5, Pin.IN, Pin.PULL_DOWN)
5
6 while True:
7     if button.value():
8         print("Button is pressed!")
9         sleep(1)
10

Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>

MicroPython (Raspberry Pi Pico)
    
```

Primjer

Slika kako strujni krug izgleda koristeći priloženi hardver:



6. Zujalica

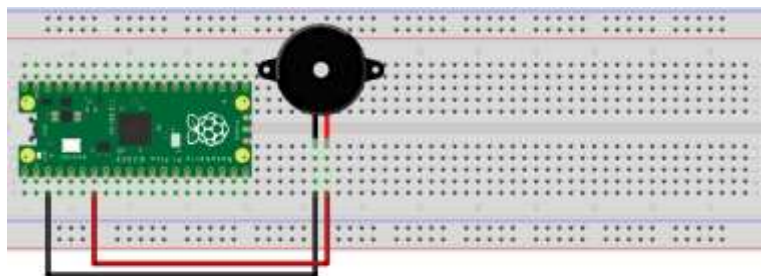
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati zujalicu. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `buzzer.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 2 x Muški prenosni kabel
- 1 x Zujalica

Dijagram ožičenja

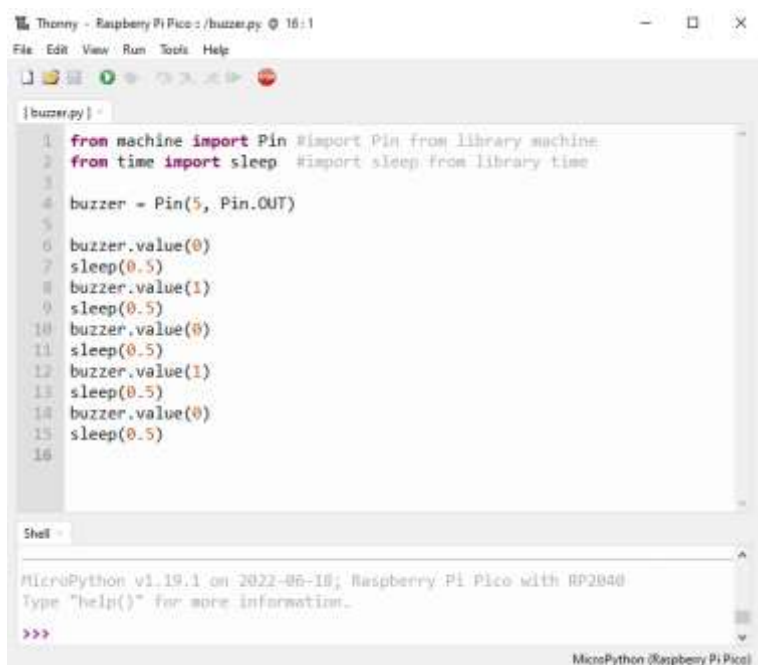


fritzing

- spojite duži kraj (+) zujalice na GPIO5 iglu
- spojite kraći kraj (-) zujalice na GND iglu

Kod

MicroPython kod za lekciju:



```

1 from machine import Pin #import Pin from library machine
2 from time import sleep #import sleep from library time
3
4 buzzer = Pin(5, Pin.OUT)
5
6 buzzer.value(0)
7 sleep(0.5)
8 buzzer.value(1)
9 sleep(0.5)
10 buzzer.value(0)
11 sleep(0.5)
12 buzzer.value(1)
13 sleep(0.5)
14 buzzer.value(0)
15 sleep(0.5)
16
    
```

Shell

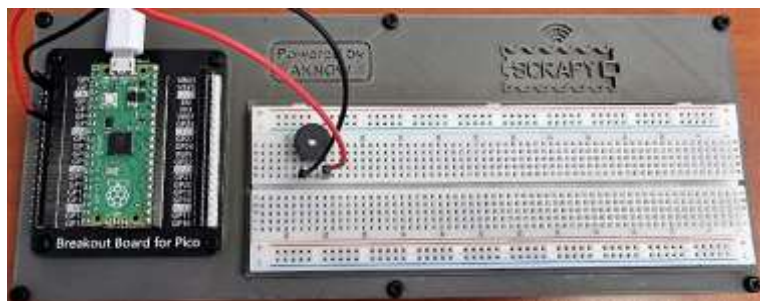
```

MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more informatiz...
>>>
    
```

MicroPython (Raspberry Pi Pico)

Primjer

Slika kako strujni krug izgleda koristeći priloženi hardver:



7. Potenciometar

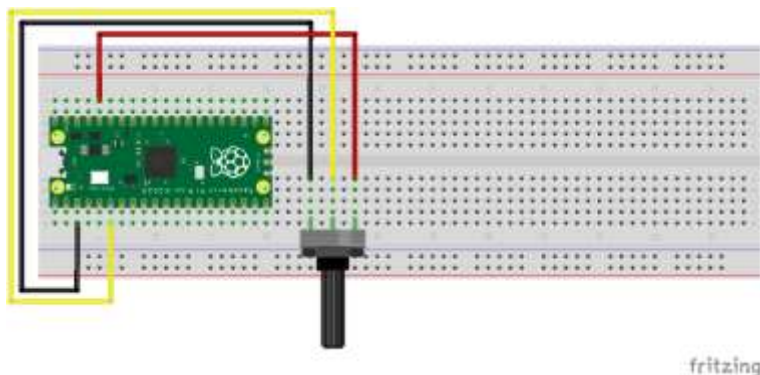
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati potenciometar. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `pot.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x Potenciometar

Dijagram ožičenja



- crni kabel bi trebao biti spojen na GND (Igla 3) iglu
- žuti kabel bi trebao biti spojen na GPIO6 iglu
- crveni kabel bi trebao biti spojen na 3V3 iglu
- pomaknite potenciometar ulijevo tako da bude isključen

Kod

MicroPython kod za lekciju:



```

1 from machine import Pin, ADC #import Pin from library machine
2 from time import sleep #import sleep from library time
3
4 pot = ADC(Pin(26))
5
6 while True:
7     print(pot.read_u16())
8     sleep(0.5)
    
```

Shell

```

MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
    
```

MicroPython (Raspberry Pi Pico)

Primjer

Slika kako strujni krug izgleda koristeći priloženi hardver:



8. RGB LED

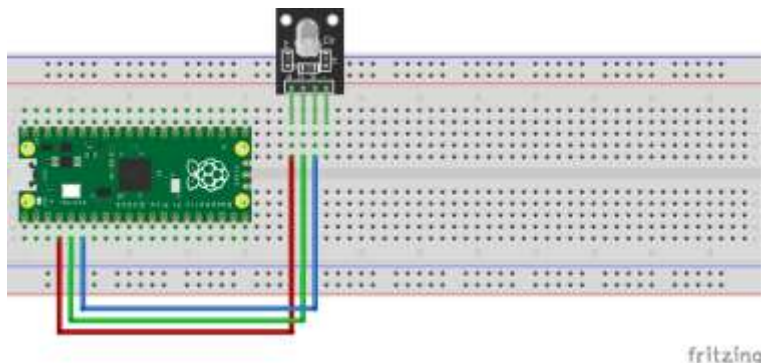
Description

U ovoj lekciji naučit ćete kako spojiti i kontrolirati RGB LED svjetla. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `rgb.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potrebna materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 4 x Muški prenosni kabel
- 1 x RGB LED modul

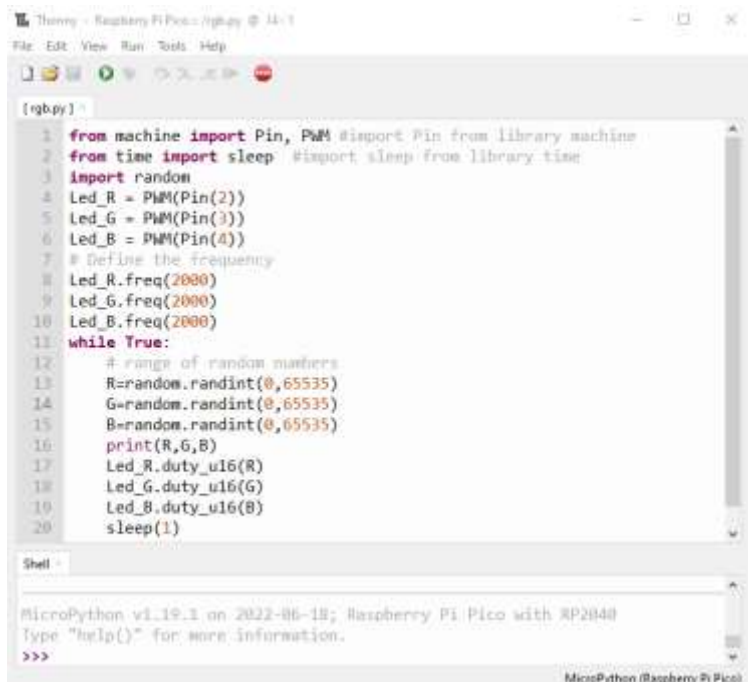
Dijagram ožičenja



- crveni kabel bi trebao biti spojen na GPIO2
- zeleni kabel bi trebao biti spojen na GPIO3
- plavi kabel bi trebao biti spojen na GPIO4
- nije potrebna GND konekcija

Kod

MicroPython kod za lekciju:



```

1  from machine import Pin, PWM #import Pin from library machine
2  from time import sleep #import sleep from library time
3  import random
4  led_R = PWM(Pin(2))
5  led_G = PWM(Pin(3))
6  led_B = PWM(Pin(4))
7  # Define the frequency
8  led_R.freq(2000)
9  led_G.freq(2000)
10 led_B.freq(2000)
11 while True:
12     # range of random numbers
13     R=random.randint(0,65535)
14     G=random.randint(0,65535)
15     B=random.randint(0,65535)
16     print(R,G,B)
17     led_R.duty_u16(R)
18     led_G.duty_u16(G)
19     led_B.duty_u16(B)
20     sleep(1)
    
```

Shell

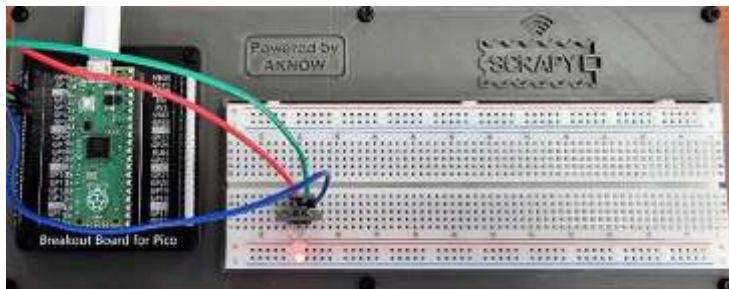
```

MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
    
```

MicroPython (Raspberry Pi Pico)

Primjer

Slika kako strujni krug izgleda koristeći priloženi hardver:



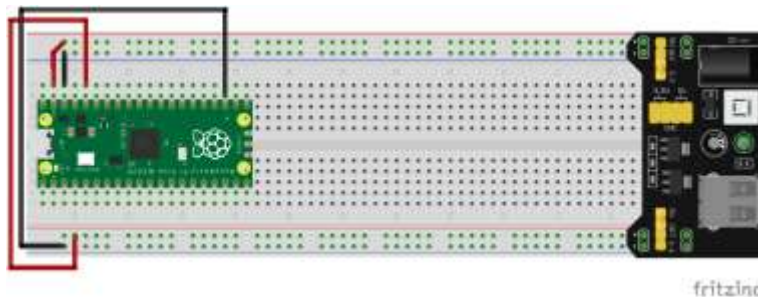
Napredne upute

Prije nego započnemo s ovim i sljedećim odjeljkom, moramo napraviti nekoliko modifikacija na našem SCRAPY kompletu. To uključuje nekoliko dodatnih komponenti i njihovu povezanost.

Komponente

- 1 x 6-AA baterija
- 1 x MB-102 modul napajanja
- 4 x Muški prenosni kabel

Molimo da slijedite shemu za povezivanje novih komponenti:



Strujni krug



- ova će se instalacija koristiti za preostale lekcije
- priključci na gornjoj strani: VSYS 5V ((+) crvena) i GND ((-) crna)

– priključci na donjoj strani: 3V3 ((+) crvena) i GND ((-) crna)

9. LDR Fotootpornik

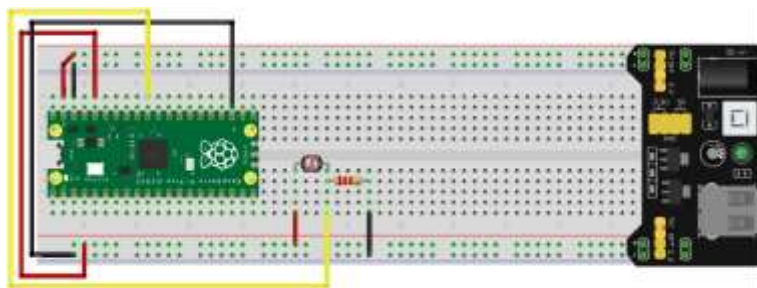
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati LDR fotootpornik. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `ldr.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x 220Ohm otpornik
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x LDR fotootpornik

Dijagram ožičenja



fritzing

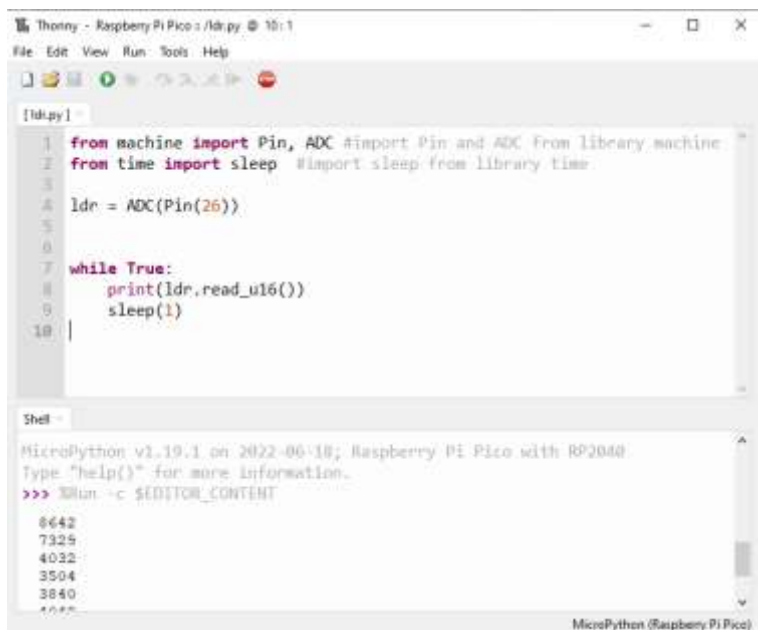
– lijeva strana LDR fotootpornika je spojena na 3V vod ((+) crvena)

– desna strana LDR fotootpornika je spojena na 220 Ohm otpornik i GPIO26 ADC (žuti kabel) iglu

– desna strana otpornika je spojena na GND vod ((-) crna)

Kod

MicroPython kod za lekciju:



```

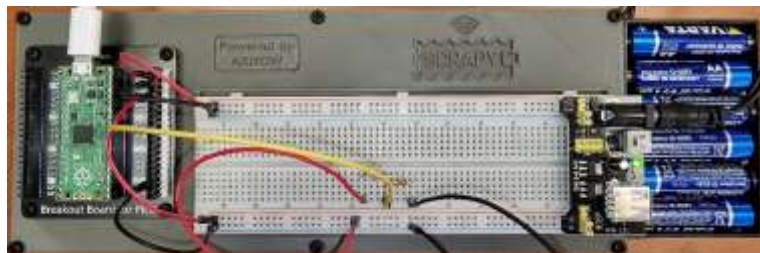
Thonny - Raspberry Pi Pico - /ldr.py @ 10:1
File Edit View Run Tools Help

[ldr.py]
1 from machine import Pin, ADC #import Pin and ADC from library.machine
2 from time import sleep #import sleep from library.time
3
4 ldr = ADC(Pin(26))
5
6
7 while True:
8     print(ldr.read_u16())
9     sleep(1)
10 |

Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> Run -c $EDITOR_CONTENT
8642
7325
4032
3504
3840
4000
    
```

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



10. Servo motor

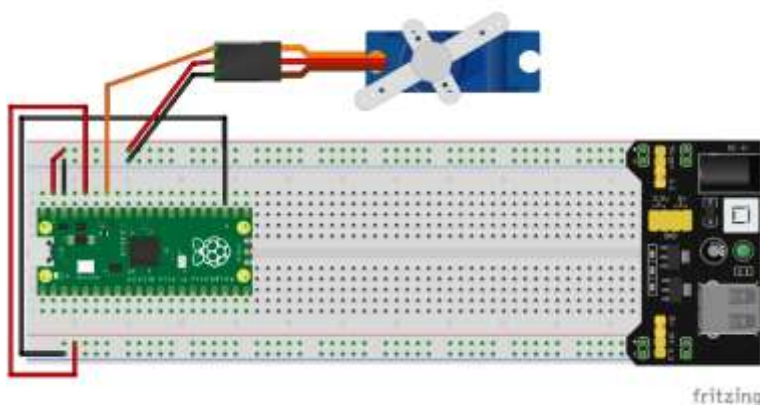
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati servo motor. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `servo.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal)

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x SG-90 servo motor

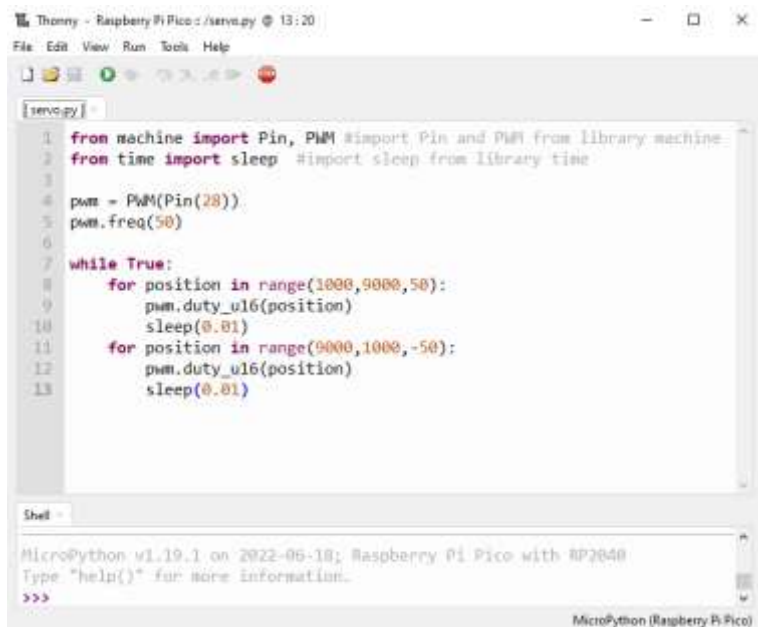
Dijagram ožičenja



- crveni kabel je spojen na 5V vod (+)
- crni/smeđi kabel je spojen na GND vod (-)
- narančasti kabel je spojen na GPIO28 ADC iglu

Kod

MicroPython kod za lekciju:



```

1 from machine import Pin, PWM #import Pin and PWM from library machine
2 from time import sleep #import sleep from library time
3
4 pwm = PWM(Pin(28))
5 pwm.freq(50)
6
7 while True:
8     for position in range(1000,9000,50):
9         pwm.duty_u16(position)
10        sleep(0.81)
11    for position in range(9000,1000,-50):
12        pwm.duty_u16(position)
13        sleep(0.81)
    
```

Shell

```

MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
    
```

MicroPython (Raspberry Pi Pico)

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



11. OLED I2C SSD1306 zaslon

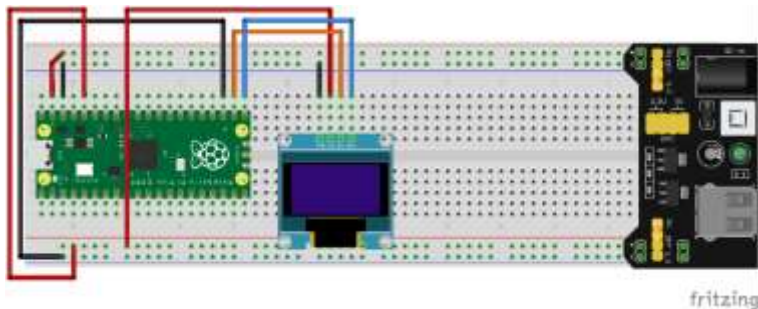
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati I2C ICC OLED zaslon. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `oled.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 4 x Muški prenosni kabel
- 1 x OLED I2C SSD1306 zaslon

Dijagram ožičenja

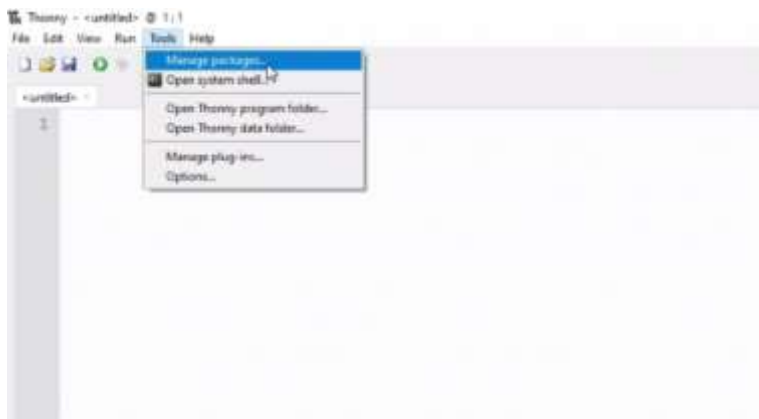


- crveni kabel je spojen na 3v3 vod (+)
- crni kabel je spojen na GND vod (-)
- narančasti kabel je spojen na GPIO17 I2C0 SCL iglu
- plavi kabel je spojen na GPIO26 I2C0 SDA iglu

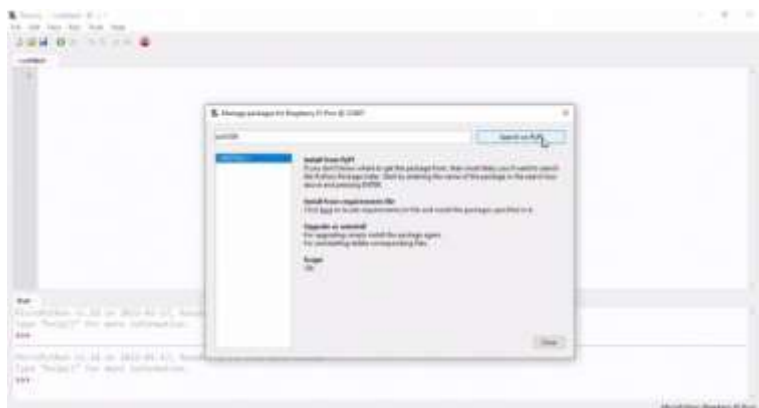
Kod

Prije početka programiranja OLED zaslona moramo dodati SSD1306 paket u naš RPi Pico. Kako biste to učinili, molimo slijedite sljedeće korake:

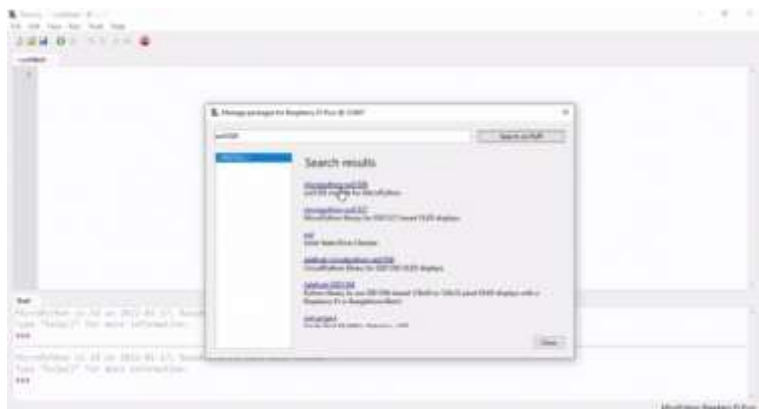
1. Otvorite Thonny i idite u **Tools** → **Manage packages...**



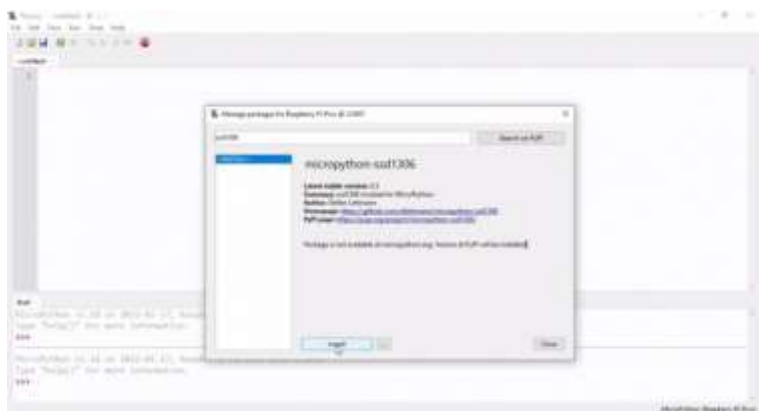
2. Na prozoru upravljanja paketa, upišite **SSD1306** i kliknite *Search*.



3. Nakon što je pretraga gotova, kliknite na *micropython-ssd1306*.



4. Na sljedećem prozoru, kliknite *Install*.



5. Pričekajte instalaciju paketa, i kliknite close.

Sada smo spremni nastaviti s programiranjem OLED zaslona.

MicroPython kod za lekciju:

```

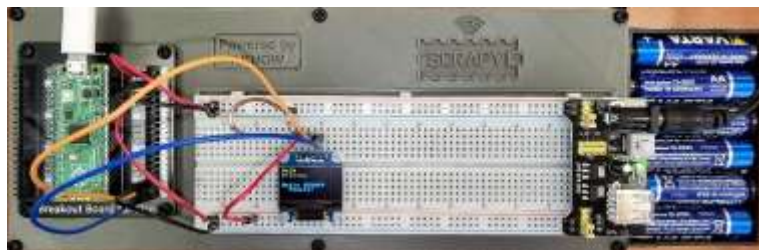
Thierry - Raspberry Pi Pico - /olek.py @ 20:11
File Edit View Run Tools Help

[olek.py]
1 from machine import Pin, I2C
2 import ssd1306
3
4 WIDTH = 128
5 HEIGHT = 64
6
7 PIN_SCL = 17
8 PIN_SDA = 16
9
10 i2c = I2C(0, scl=Pin(PIN_SCL), sda=Pin(PIN_SDA))
11 oled = ssd1306.SSD1306_I2C(WIDTH, HEIGHT, i2c)
12 oled.fill(0)
13
14 oled.text("OLED", 0, 0)
15 oled.text("Display", 0, 10)
16 oled.text("Hello SCRAPY", 0, 30)
17 oled.text(" People!", 0, 40)
18
19 oled.show()
20
21 |

Shell -
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
type "help()" for more information.
>>> $!run -c $EDITOR_CONTENT
>>>
MicroPython (Raspberry Pi Pico)
    
```

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



12. „Joystick“ modul

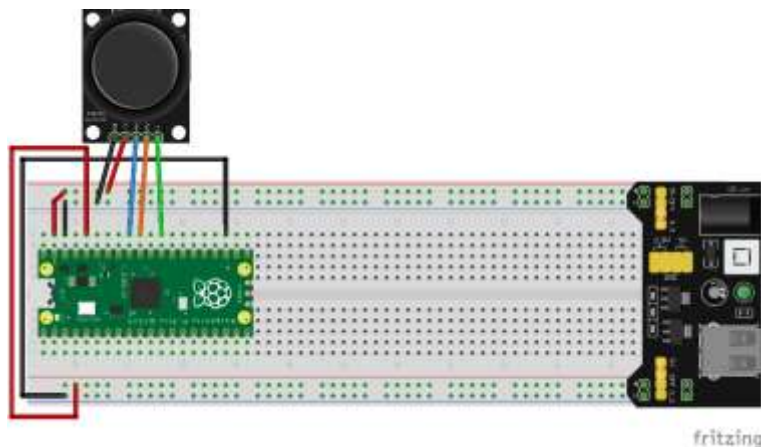
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati „joystick“ modul. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `joystick.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Pico komplet za breadboard testnu ploču
- 1 x Breadboard testna ploča pune veličine
- 5 x Muški prenosni kabel
- 1 x Micro-USB kabel
- 1 x „Joystick“ modul

Dijagram ožičenja



- +5V (crveni kabel) je spojen na 5V vod (+)
- GND (crni kabel) je spojen na GND vod (-)

- VRx (plavi kabel) je spojen na GPIO27 ADC1 iglu
- VRy (narančasti kabel) je spojen na GPIO26 ADC0 iglu
- SW (zeleni) kabel je spojen na GPIO22 iglu

Kod

MicroPython kod za lekciju:



```

1 from machine import Pin, ADC
2 from time import sleep
3
4 VRX = ADC(Pin(27))
5 VRy = ADC(Pin(26))
6 SW = Pin(22, Pin.IN, Pin.PULL_UP)
7
8 while True:
9     xAxis = VRX.read_u16()
10    yAxis = VRy.read_u16()
11    switch = SW.value()
12
13    print("X-axis: " + str(xAxis) + ", Y-axis: " + str(yAxis) + ", Switch " + str(switch))
14    if switch == 0:
15        print("Push button pressed!")
16    print("")
17    sleep(1)
    
```

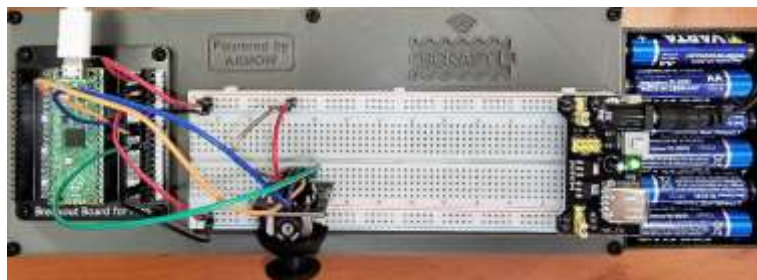
Shell

```

1 [000] "000000" 100 0000 100000000
2 >>> Shell = 00000000 00000000
3 X-axis: 32750, Y-axis: 34616, Switch 1
4 X-axis: 32471, Y-axis: 34520, Switch 1
5 X-axis: 32453, Y-axis: 34470, Switch 1
    
```

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



Lekcije sa senzorima

13. Senzor kapi kiše

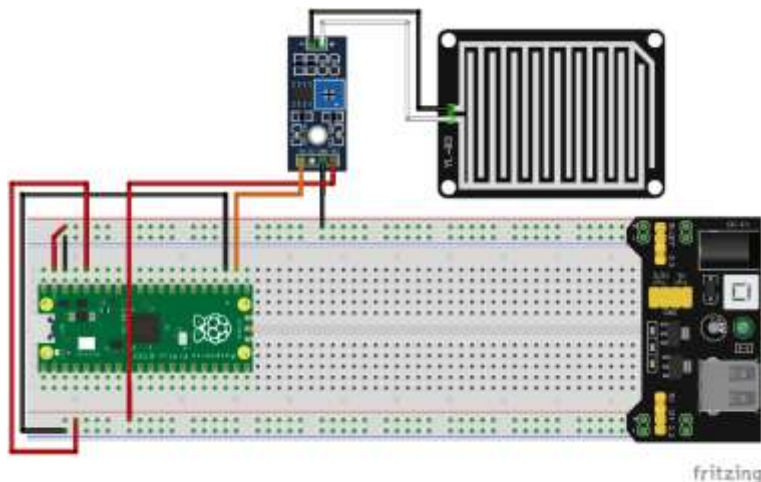
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati senzor kapi kiše. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `raindrop.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x Senzor kapi kiše

Dijagram ožičenja

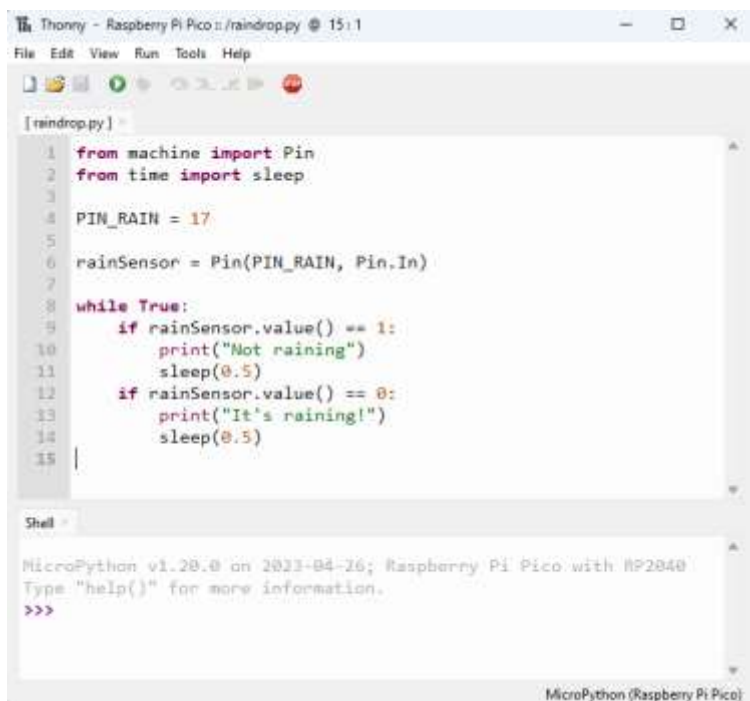


- 3v3V (crveni kabel) je spojen na 3v3V vod (+)
- GND (crni kabel) je spojen na GND vod (-)

– DO (narančasti kabel) je spojen na GPIO17 iglu

Kod

MicroPython kod za lekciju:

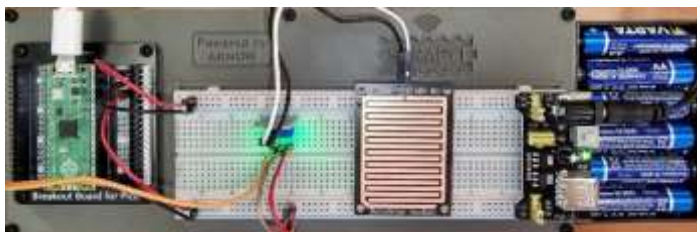


```

Thonny - Raspberry Pi Pico ::raindrop.py @ 15:1
File Edit View Run Tools Help
[raindrop.py]
1 from machine import Pin
2 from time import sleep
3
4 PIN_RAIN = 17
5
6 rainSensor = Pin(PIN_RAIN, Pin.In)
7
8 while True:
9     if rainSensor.value() == 1:
10        print("Not raining")
11        sleep(0.5)
12    if rainSensor.value() == 0:
13        print("It's raining!")
14        sleep(0.5)
15
Shell
MicroPython v1.20.0 on 2023-04-26; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
MicroPython (Raspberry Pi Pico)
    
```

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



14. HC-SR04 Ultrazvučni senzor

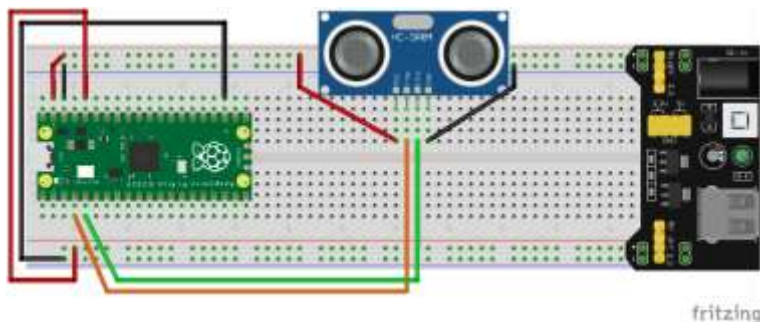
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati HC-SR04 ultrazvučni senzor. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `ultrasonic.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x HC-SR04 Ultrazvučni senzor

Dijagram ožičenja

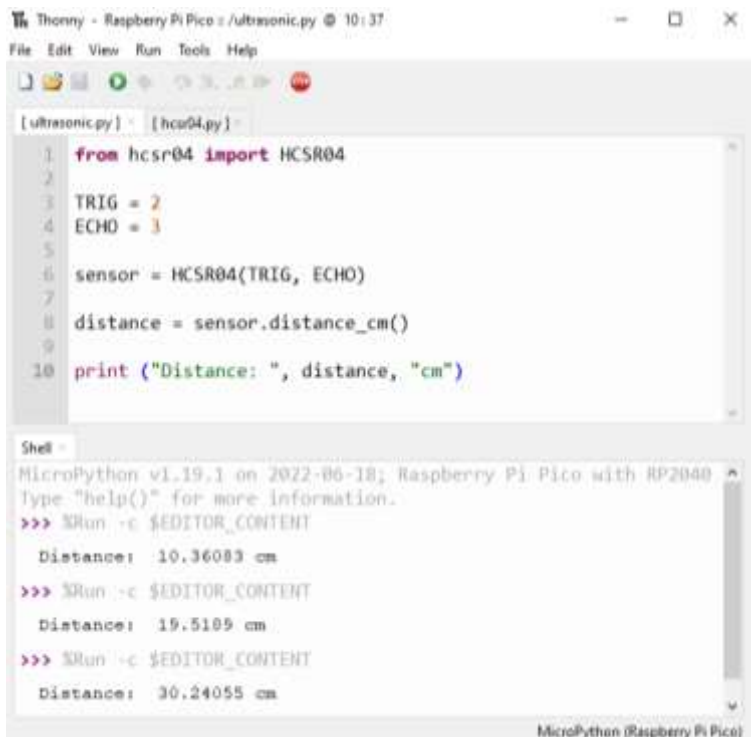


- VCC (crveni kabel) je spojen na 5V vod (+)
- GND (crni kabel) je spojen na GND vod (-)
- TRIG (narančasti kabel) je spojen na GPIO2 iglu
- ECHO (zeleni kabel) je spojen na GPIO3 iglu

Kod

Za upotrebu HC-SR04 ultrazvučnog senzora možemo razviti vlastiti program, ili koristiti jednu od dostupnih programskih biblioteka na internetu s primjerice [Githuba](#). Ako odlučite preuzeti *hcsr04.py* programsku biblioteku, trebali biste je spremiti u svoj Pico-u pod istim imenom.

MicroPython kod koristeći postojeću programsku biblioteku:



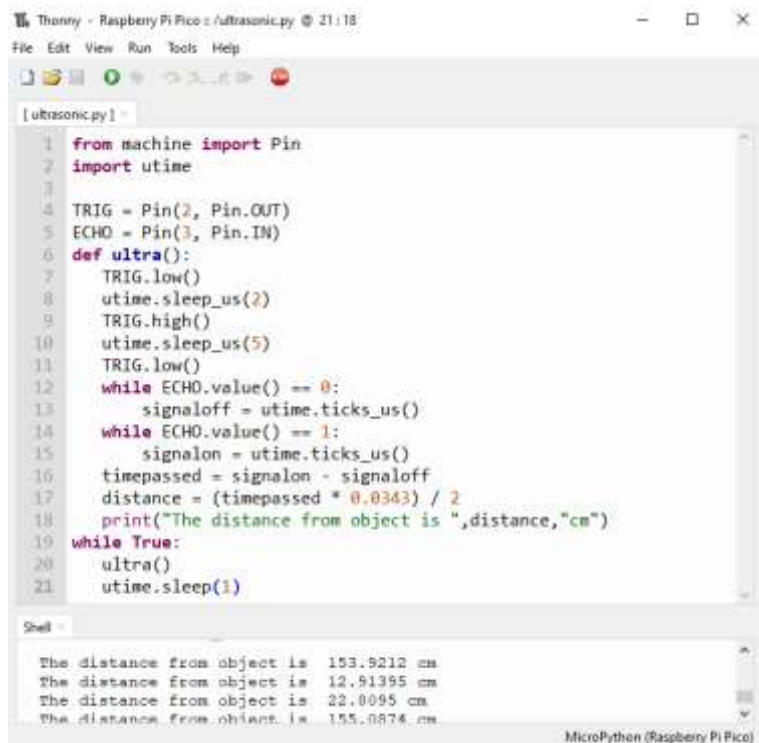
```

Thonny - Raspberry Pi Pico : /ultrasonic.py @ 10:37
File Edit View Run Tools Help

[ultrasonic.py] - [hcsr04.py]
1 from hcsr04 import HCSR04
2
3 TRIG = 2
4 ECHO = 3
5
6 sensor = HCSR04(TRIG, ECHO)
7
8 distance = sensor.distance_cm()
9
10 print ("Distance: ", distance, "cm")

Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> $Run -c $EDITOR_CONTENT
    Distance: 10.36083 cm
>>> $Run -c $EDITOR_CONTENT
    Distance: 19.5109 cm
>>> $Run -c $EDITOR_CONTENT
    Distance: 30.24055 cm
    
```

MicroPython kod bez postojeće programske biblioteke:



```

1 from machine import Pin
2 import utime
3
4 TRIG = Pin(2, Pin.OUT)
5 ECHO = Pin(3, Pin.IN)
6 def ultra():
7     TRIG.low()
8     utime.sleep_us(2)
9     TRIG.high()
10    utime.sleep_us(5)
11    TRIG.low()
12    while ECHO.value() == 0:
13        signaloff = utime.ticks_us()
14    while ECHO.value() == 1:
15        signalon = utime.ticks_us()
16    timepassed = signalon - signaloff
17    distance = (timepassed * 0.0343) / 2
18    print("The distance from object is ",distance,"cm")
19 while True:
20     ultra()
21     utime.sleep(1)
    
```

Shell

```

The distance from object is 153.9212 cm
The distance from object is 12.91395 cm
The distance from object is 22.0095 cm
The distance from object is 155.0874 cm
    
```

MicroPython (Raspberry Pi Pico)

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



15. PIR Detektor pokreta

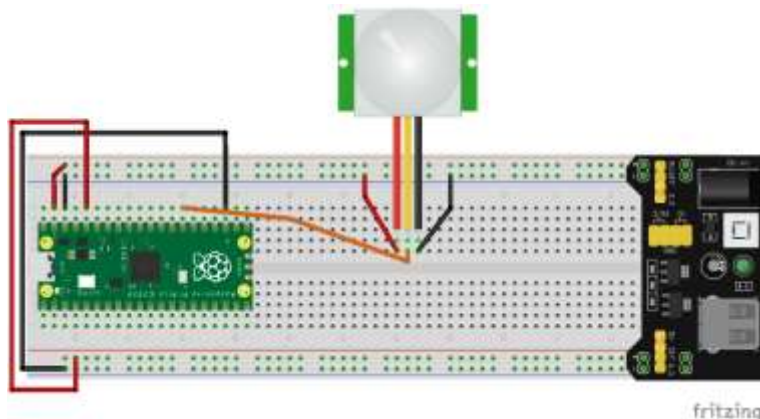
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati PIR detektor pokreta. Otvorite Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `motion.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Full size breadboard
- 1 x Micro-USB cable
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x PIR detektor pokreta

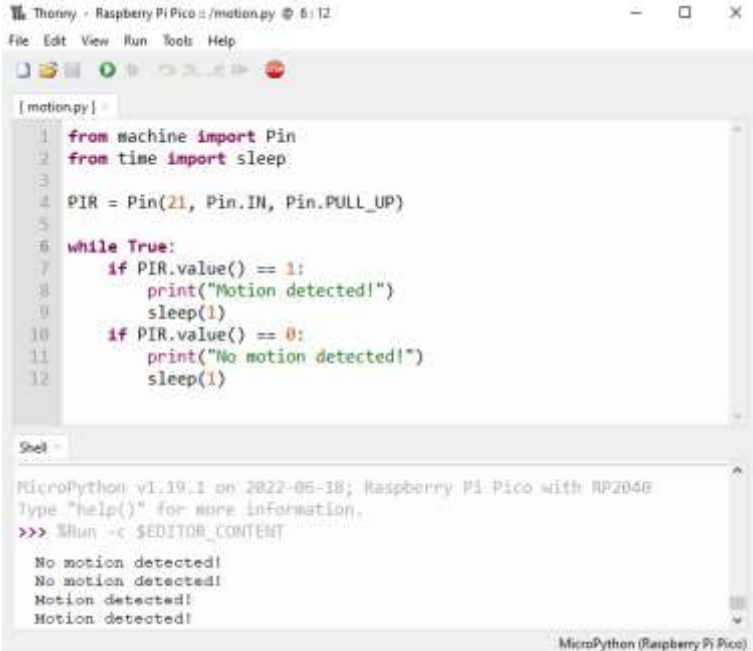
Dijagram ožičenja



- VCC (crveni kabel) je spojen na 5V vod (+)
- GND (crni kabel) je spojen na GND vod (-)
- OUT (narančasti kabel) je spojen na GPIO21 iglu

Kod

MicroPython kod za lekciju:



```
Thonny - Raspberry Pi Pico :: motion.py @ 5:12
File Edit View Run Tools Help

[motion.py]
1 from machine import Pin
2 from time import sleep
3
4 PIR = Pin(21, Pin.IN, Pin.PULL_UP)
5
6 while True:
7     if PIR.value() == 1:
8         print("Motion detected!")
9         sleep(1)
10    if PIR.value() == 0:
11        print("No motion detected!")
12        sleep(1)

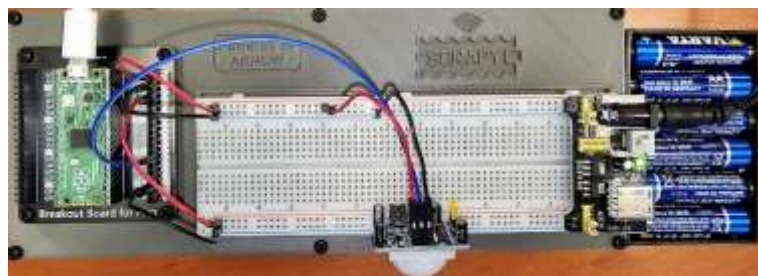
Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %run -c $EDITOR_CONTENT

No motion detected!
No motion detected!
Motion detected!
Motion detected!

MicroPython (Raspberry Pi Pico)
```

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



16. DHT11 senzor

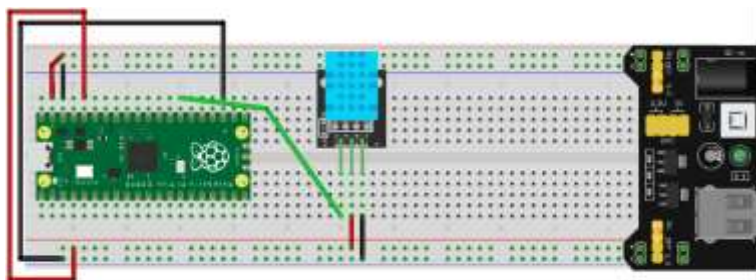
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati DHT11 senzor digitalne temperature i vlažnosti. Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `dht11.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x DHT11 senzor temperature

Dijagram ožičenja

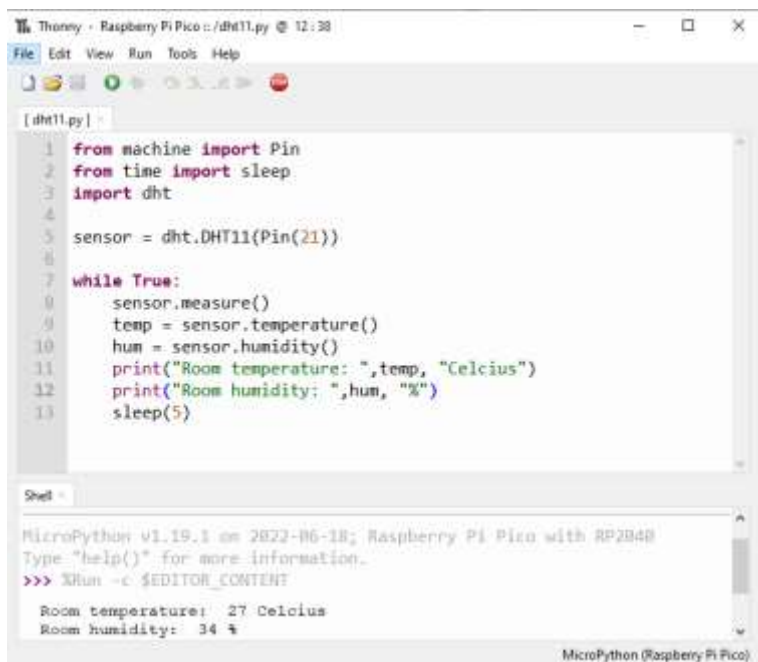


fritzing

- VCC (red cable) is connected to 3v3 rail (+)
- GND (black cable) is connected to GND rail (-)
- S (green cable) is connected to GPIO21 pin

Kod

MicroPython kod za lekciju:



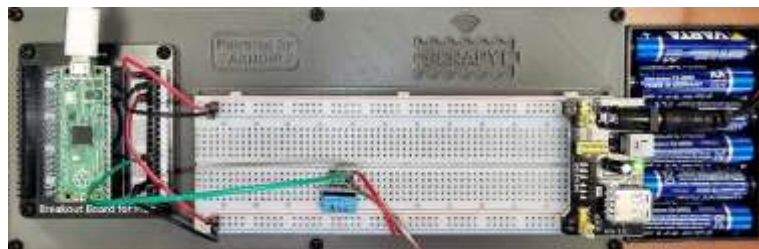
```

Thonny - Raspberry Pi Pico -:/dht11.py @ 12:38
File Edit View Run Tools Help
[dht11.py]
1 from machine import Pin
2 from time import sleep
3 import dht
4
5 sensor = dht.DHT11(Pin(21))
6
7 while True:
8     sensor.measure()
9     temp = sensor.temperature()
10    hum = sensor.humidity()
11    print("Room temperature: ",temp, "Celcius")
12    print("Room humidity: ",hum, "%")
13    sleep(5)

Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> !Run -c $EDITOR_CONTENT
Room temperature: 27 Celcius
Room humidity: 34 %
MicroPython (Raspberry Pi Pico)
    
```

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



17. SW-420 Senzor vibracije

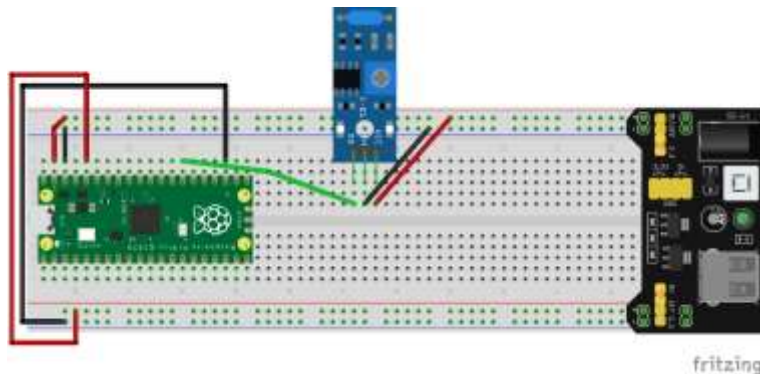
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati SW-420 senzor vibracije. Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `vibration.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x SW-420 senzor vibracije

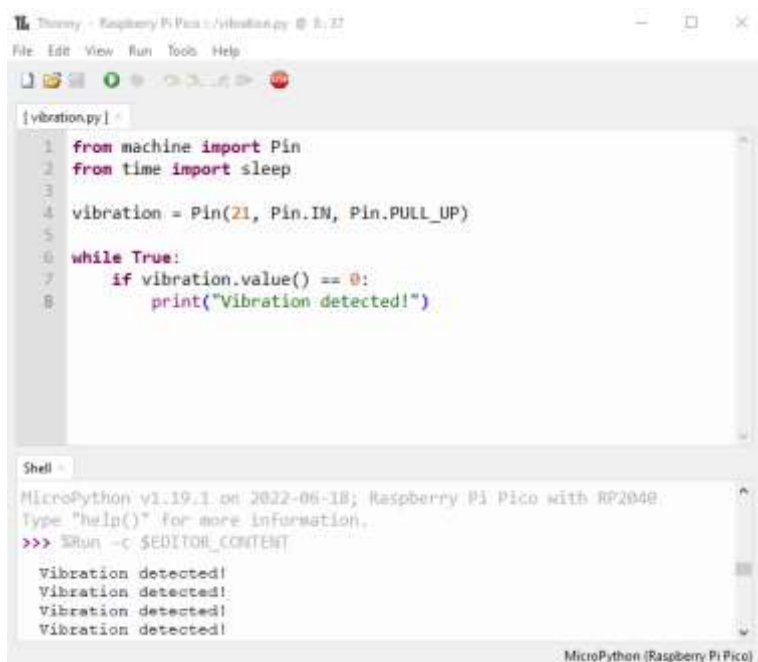
Dijagram ožičenja



- VCC (crveni kabel) je spojen na 5V vod (+)
- GND (crni kabel) je spojen na GND vod (-)
- DO (zeleni kabel) je spojen na GPIO21 iglu

Kod

MicroPython kod za lekciju:



```

1  from machine import Pin
2  from time import sleep
3
4  vibration = Pin(21, Pin.IN, Pin.PULL_UP)
5
6  while True:
7      if vibration.value() == 0:
8          print("Vibration detected!")
    
```

Shell

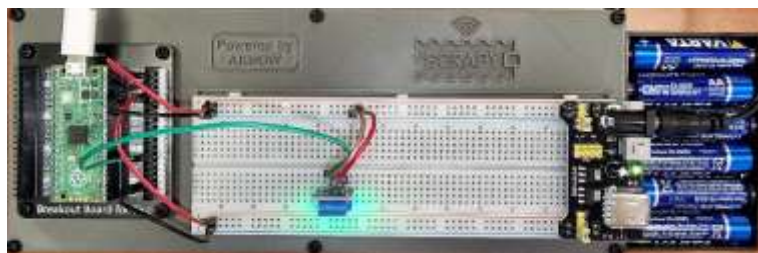
```

MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> $RUN -c $EDITOR_CONTENT
Vibration detected!
Vibration detected!
Vibration detected!
Vibration detected!
    
```

MicroPython (Raspberry Pi Pico)

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



18. Detektor plamena

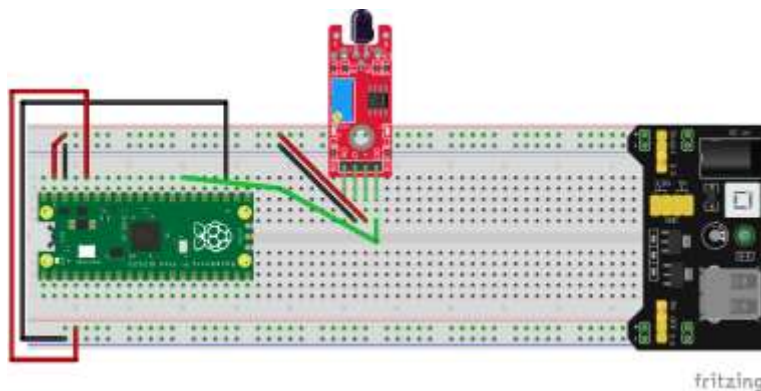
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati KY-026 detektor plamena. Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `flame.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x KY-026 detektor plamena

Dijagram ožičenja



- VCC (crveni kabel) je spojen na 5V vod (+)
- GND (crni kabel) je spojen na GND vod (-)
- DO (zeleni kabel) je spojen na GPIO21 iglu

Kod

MicroPython kod za lekciju:



```

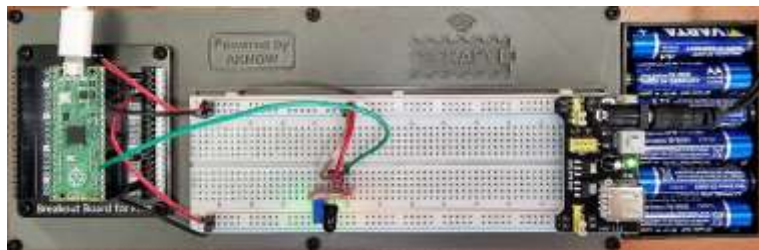
Thonny - Raspberry Pi Pico ::/flame.py @ 10:18
File Edit View Run Tools Help
[flame.py]
1 from machine import Pin
2 from time import sleep
3
4 flame = Pin(21, Pin.IN)
5 sleep(2)
6
7 while True:
8     if flame.value() == 0:
9         print("Flame Detected")
10        sleep(3)

Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> Run -c $EDITOR_CONTENT

MicroPython (Raspberry Pi Pico)
    
```

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



19. Senzor za otkrivanje zvuka

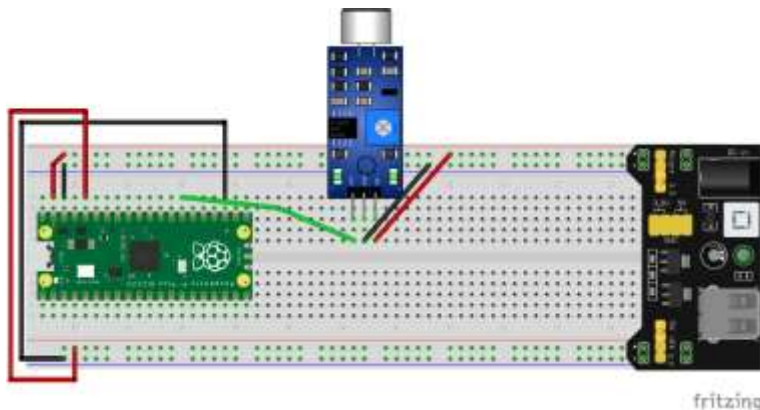
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati KY-037 senzor za otkrivanje zvuka. Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `sound.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x KY-037 senzor za otkrivanje zvuka

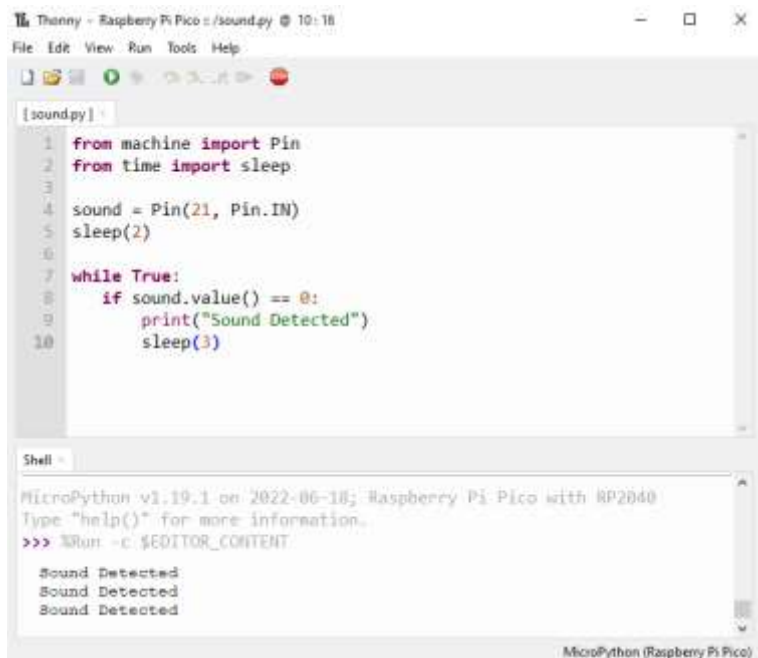
Dijagram ožičenja



- VCC (crveni kabel) je spojen na 5V vod (+)
- GND (crni kabel) je spojen na GND vod (-)
- DO/OUT (zeleni kabel) je spojen na GPIO21 iglu

Kod

MicroPython kod za lekciju:



```

Thonny - Raspberry Pi Pico - /sound.py @ 10: 18
File Edit View Run Tools Help

[sound.py]
1 from machine import Pin
2 from time import sleep
3
4 sound = Pin(21, Pin.IN)
5 sleep(2)
6
7 while True:
8     if sound.value() == 0:
9         print("Sound Detected")
10        sleep(3)

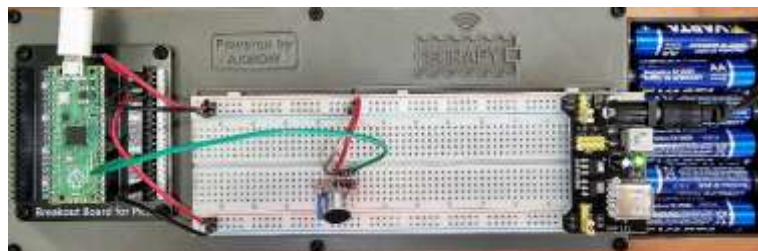
Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> \Run -c: $EDITOR_CONTENT

Sound Detected
Sound Detected
Sound Detected

MicroPython (Raspberry Pi Pico)
    
```

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



20. Detektor vlage tla

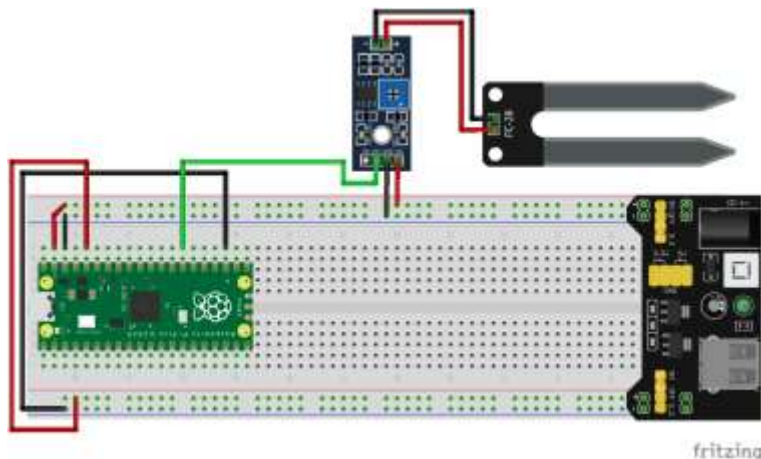
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati detektor vlage tla. Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `soil.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x Detektor vlage tla

Dijagram ožičenja

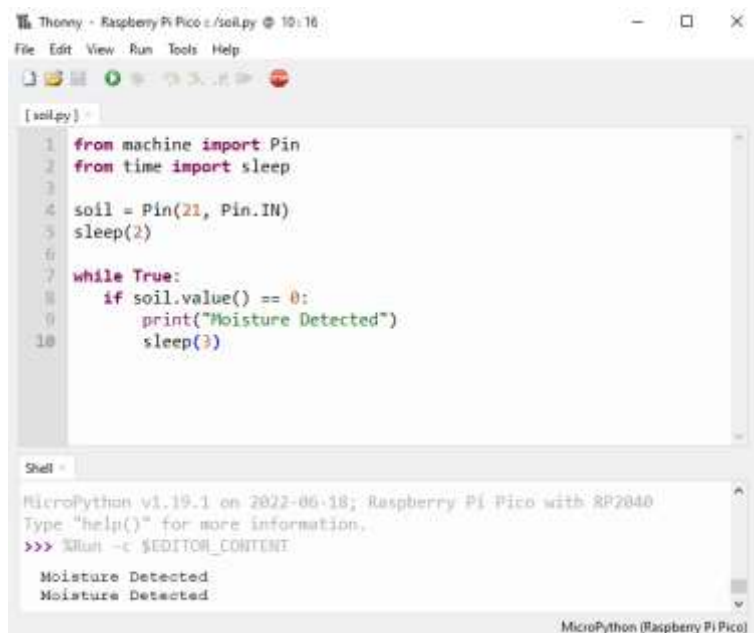


- VCC (crveni kabel) je spojen na 5V vod (+)
- GND (crni kabel) je spojen na GND vod (-)

– DO/OUT (zeleni kabel) je spojen na GPIO21 iglu

Kod

MicroPython kod za lekciju:



```

1  from machine import Pin
2  from time import sleep
3
4  soil = Pin(21, Pin.IN)
5  sleep(2)
6
7  while True:
8      if soil.value() == 0:
9          print("Moisture Detected")
10         sleep(3)
    
```

Shell

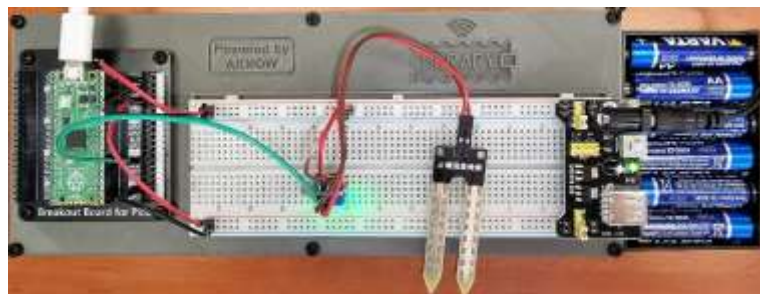
```

MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> Run -c $EDITOR_CONTENT
Moisture Detected
Moisture Detected
    
```

MicroPython (Raspberry Pi Pico)

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



21. Infracrveni IR senzor

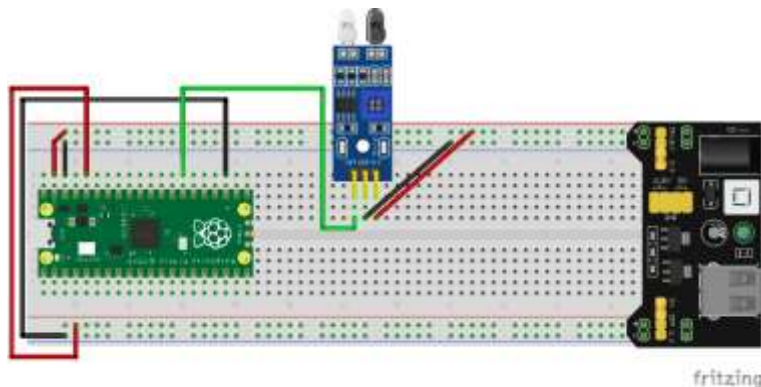
Opis

U ovoj lekciji naučit ćete kako spojiti i kontrolirati infracrveni IR senzor. Thonny Python, zatim idite u File → Save as..., odaberite Raspberry Pi Pico, i spremite datoteku pod imenom `ir.py`. Zatim je vrijeme da spojite elektroniku i napišite svoj program. Molimo pratite instrukcije u nastavku.

Potreban materijal

- 1 x Raspberry Pi Pico
- 1 x Breadboard testna ploča pune veličine
- 1 x Micro-USB kabel
- 1 x Pico komplet za breadboard testnu ploču
- 3 x Muški prenosni kabel
- 1 x Infracrveni IR senzor

Dijagram ožičenja



- VCC (crveni kabel) je spojen na 5V vod (+)
- GND (crni kabel) je spojen na GND vod (-)
- OUT (zeleni kabel) je spojen na GPIO21 iglu

Kod

MicroPython kod za lekciju:

```

Thierry - Raspberry Pi Pico - /ngy @ 13:17
File Edit View Run Tools Help

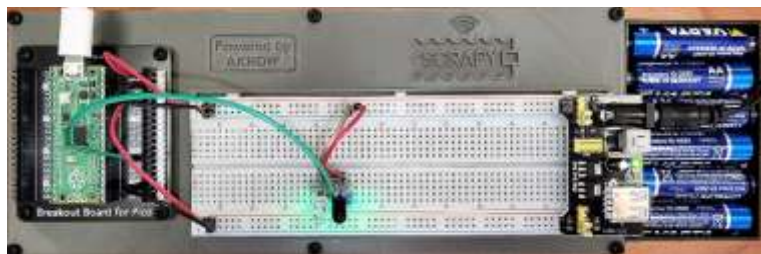
[ngy]
1 from machine import Pin
2 from time import sleep
3
4 ir = Pin(21, Pin.IN)
5 sleep(2)
6
7 while True:
8     if ir.value() == 1:
9         print("No obstacles")
10        sleep(3)
11    else:
12        print("Obstacle detected!")
13        sleep(3)
14

Shell
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> !Run -c $EDITOR_CONTENT
Obstacle detected!
Obstacle detected!

MicroPython (Raspberry Pi Pico)
    
```

Primjer

Slika kako lekcija izgleda koristeći priloženi hardver:



DODATAK: MicroPython tablica sažetka

Digitalni izlaz		
Pozivanje na vrijednost pina	<code>from machine import Pin</code>	
Inicijalizacija digitalnog izlaznog objekta	<code>led = Pin(pin_value, Pin.OUT)</code>	<code>pin_value</code> od 0 do 40
Otvori digitalni izlaz (3.3V izlaz)	<code>led.value(1)</code>	UKLJUČENO
Zatvori digitalni izlaz (0V izlaz)	<code>led.value(0)</code>	ISKLJUČENO

Digitalni ulaz		
Pozivanje na vrijednost pina	<code>from machine import Pin</code>	
Inicijalizacija digitalnog izlaznog objekta	<code>button = Pin(pin_value, Pin.IN)</code>	<code>pin_value</code> od 0 do 40
	<code>button = Pin(pin_value), Pin.IN, Pin.PULL_UP)</code>	Aktivacija PULL UP otpora
	<code>button = Pin(pin_value), Pin.IN, Pin.PULL_DOWN)</code>	Aktivacija PULL DOWN otpora
Očitavanje unosa	<code>value = button.value(1)</code>	Povratna vrijednost može biti 0 ako je igla na 0V, ili 1 ako je igla na 3.3V

Analogni izlaz (Modulacija širine pulsa - PWM)		
Pozivanje na vrijednost PWM-a	<code>from machine import PWM</code>	
Inicijalizacija analognog izlaznog objekta	<code>led = PWM(Pin(pin_value), frequency)</code>	<code>pin_value</code> od 0 do 40 <code>frequency</code> u HZ, od 0 do 78125
Očitavanje unosa	<code>led.duty(duty_cycle)</code>	<code>duty_cycle</code> od 0 do 1023 (0V izlaz do 3.3V izlaz)

Digitalni Ulaz		
Pozivanje na vrijednost ADC-a	<code>from machine import ADC</code>	
Inicijalizacija analognog ulaznog objekta	<code>pot = ADC(Pin(pin_value))</code>	<code>pin_value</code> može biti GPIO26, GPIO27 and GPIO28
Deklaracija pri kojem će naponu ulaz dati najveću vrijednost (u ESP32 obično 3.3V)	<code>pot.atten(ADC.ATTN_11DB)</code>	ADC.ATTN_0DB: napon punog raspona: 1.2 V ADC.ATTN_2_5DB: napon punog raspona: 1.5 V ADC.ATTN_6DB: napon punog raspona: 2.0 V ADC.ATTN_11DB: napon punog raspona: 3.3 V
Deklaracija raspona ulazne vrijednosti (zadano 12bit)	<code>pot.width(ADC.WIDTH_10BIT)</code>	ADC.WIDTH_9BIT: raspon 0 do 511 ADC.WIDTH_10BIT: raspon 0 do 1023 ADC.WIDTH_11BIT: raspon 0 do 2047 ADC.WIDTH_12BIT: raspon 0 do 4095
Očitavanje unosa	<code>value = pot.read1()</code>	<code>value</code> je cijeli broj od 0 do maksimuma raspona određenog naredbom <code>ADC.WIDTH_#BIT</code> (pogledajte prethodno)

Varijabla vremena		
Pozivanje vrijednosti spavanja	<code>from time import sleep</code>	
Korištenje funkcije spavanja	<code>sleep(sec)</code>	<code>sec</code> je broj sekundi za koje će program biti odgođen
Pozivanje vrijednosti vremena	<code>import time</code>	
Korištenje funkcije vremena	<code>current_time = time()</code>	The <code>current_time</code> varijabla će imati numeričku vrijednost jednaku broju sekundi od zadnjeg resetiranja na ploči

Struktura varijable ako (if)	
<pre>if <expr1>: <statement1> elif <expr2>: <statement2> elif <expr3>: <statement3> (...) else: <statementn></pre>	<p><expr#>: kontrolni uvjet koji mora ispasti istinit ili lažan</p> <p><statement#>: skup naredbi koje treba izvršiti kada je susjedni uvjet zadovoljen</p> <p><expr#> (npr. skup <statement2> se izvršava kada je <expr2> zadovoljen)</p> <p><statementn>: skup instrukcija koje se izvršavaju kada niti jedan od <expr#> uvjeta nije zadovoljen</p>

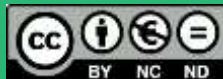
Struktura „while“ petlje	
<pre>while <expr>: <statement(s)></pre>	<p><expr>: kontrolni uvjet koji mora ispasti istinit ili lažan</p> <p><statement#>: skup naredbi koje treba izvršiti sve dok je <expr> uvjet zadovoljen</p>

Struktura „for“ petlje	
<pre>for <var> in <iterable>: <statement(s)></pre>	<p><iterable>: kolekcija objekta, npr. popis koji sadrži brojeve, alfanumeričke znakove itd.</p> <p><var>: varijabla kojoj se dodjeljuje vrijednost sljedeće stavke u kolekciji <iterable></p> <p><statement(s)>: skup instrukcija koje se izvršavaju pri svakoj iteraciji</p>
<pre>for <var> in range(<start>, <end>, <step>): <statement(s)></pre>	<p>range(<start>, <end>, <step>): funkcija koja vraća niz brojeva od <start> do <end>-1, s <step> razlikom između dva uzastopna broja (<start> i <step> parametri nisu obavezni).</p> <p><var>: varijabla kojoj se dodjeljuje vrijednost sljedećeg elementa niza proizvedenog rasponom</p>

		<statement(s)>: skup instrukcija koje se izvode u svakoj iteraciji
Razno		
DHT11	<code>import dht</code>	Uvoz DHT biblioteke
	<code>sensor = dht.DHT11(Pin(pin_number))</code>	Inicijalizacija varijable senzora s pripadajućim <code>pin_number</code> .
	<code>sensor.measure()</code>	Ažuriranje vrijednosti senzora
	<code>temp = sensor.temperature()</code>	Spremanje trenutne vrijednosti temperature
	<code>hum = sensor.humidity()</code>	Spremanje trenutne vrijednosti vlažnosti
OLED ZASLON	<code>from machine import I2C</code>	Uvoz I2C biblioteke
	<code>import ssd1306</code>	Uvoz ssd1306 biblioteke
	<code>i2c = I2C(-1, scl=Pin(1), sda=Pin(0))</code>	Inicijalizacija varijable <code>i2c</code> na SCL & SDA iglama Pico-a
	<code>oled_width = 128</code> <code>oled_height = 64</code> <code>oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)</code>	Inicijalizacija zaslona
	<code>oled.text('Hello, World 1!', 0, 0)</code> <code>oled.text('Hello, World 2!', 0, 10)</code> <code>oled.text('Hello, World 3!', 0, 20)</code>	Pohranjivanje poruka u međuspremnik zaslona
	<code>oled.show()</code>	Prikazivanje poruka (potrebno za prikaz poruka pohranjenih u međuspremniku zaslona)
	<code>display.pixel(3, 4, 1)</code>	Postavite piksel koji je lociran na poziciji (x,y) na ekranu, s x=3 & y=4, na stanje 1 (tj. zaslon)



Co-funded by
the European Union



2021-1-FR01-KA220-SCH-000031617

Potpora Europske komisije za izradu ove publikacije ne predstavlja potvrdu sadržaja koji odražava samo stajališta autora te se Komisija ne može smatrati odgovornom za bilo kakvu uporabu informacija sadržanih u njoj.

Raspberry Pi Pico Pinout

