



SCRAPY-handleiding



**Co-funded by
the European Union**

De steun van de Europese Commissie voor de productie van deze publicatie houdt geen goedkeuring in van de inhoud, die uitsluitend de standpunten van de auteurs weergeeft, en de Commissie kan niet verantwoordelijk worden gehouden voor het gebruik van de informatie die erin is vevat.

WAARSCHUWING

Let bij het gebruik van dit product op de volgende waarschuwingen:

1. Dit product bevat veel kleine onderdelen. Inslikken of onjuist gebruik kan ernstige infecties en de dood tot gevolg hebben. Zoek onmiddellijk medische hulp bij een ongeval.
2. Gebruik van dit product en onderdelen ervan in de buurt van stopcontacten of andere circuits is strikt verboden vanwege het mogelijke risico op elektrische schokken.
3. Gebruik dit product nooit in de buurt van vloeistoffen of vuur.
4. Houd geleidende materialen uit de buurt van dit product.
5. Laat kinderen jonger dan 3 jaar dit product niet gebruiken zonder toezicht van een volwassene. Plaats dit product op een plek waar kinderen jonger dan 3 jaar niet bij kunnen.
6. Bewaar of gebruik dit product niet in extreme omgevingen, zoals extreme hitte of kou, hoge vochtigheid, direct zonlicht, enz.
7. Vergeet niet het circuit te verbreken als het niet nodig is.
8. Sommige onderdelen van dit product kunnen warm aanvoelen bij gebruik in bepaalde circuitontwerpen, wat normaal is.
9. Onjuist gebruik kan oververhitting veroorzaken.
10. Het gebruik van onderdelen die niet voldoen aan de specificatie kan leiden tot schade aan het product.

Inleiding

De SCRAPY KIT is gebaseerd op het gebruik van de Raspberry Pi Pico Microcontroller. De "SCRAPY KIT" is gemaakt in het kader van een door Erasmus+ medegefinancierd project.

De nieuwe trend in onderwijs op afstand, veroorzaakt door de COVID-19 pandemie, zorgt voor een leemte in het onderwijzen en oefenen van activiteiten met fysieke informatica om leerlingen te bereiken die het risico lopen slecht te presteren in dergelijke bèta/technische vakken.

Het doel van het KIT is het ondersteunen en bevorderen van praktijkgericht onderwijs in fysieke computertechniek en programmeren, zowel in het onderwijs op afstand als in de klas. Het KIT biedt een combinatie van fysieke computerprincipes en programmeren met hardware en software, wat leidt tot een innovatieve leerervaring.

Het doel van deze handleiding is:

- Je informeren over de onderdelen van de Kit en het gebruik van de belangrijkste elektronische elementen.
- Begeleid je stap voor stap bij het effectief in elkaar zetten van de KIT, rekening houdend met de bijbehorende voorzorgsmaatregelen.
- Zorg voor tutorials voor het gebruik van de componenten en de verbinding met de Pico microcontroller.
- Geef tutorials met de functionaliteiten en reikwijdte van elk elektronisch onderdeel.

**Veel leesplezier en plezier door
praktijkoefeningen en experimenten met de
SCRAPY KIT**

Inhoudsopgave

Inleiding.....	1
Inhoudsopgave	2
Inbegrepen in de SCRAPY KIT	4
Componenten Uitleg	6
1. Wat is een breadboard?	6
2. Wat is een weerstand?	7
3. Wat is een condensator?	11
4. Wat is een diode?	11
5. Wat is een startkabel?	12
Projectvoorbereiding	14
Kit montage.....	22
Basishandleidingen.....	26
0. "Hallo SCRAPY mensen!"	26
4. Een LED aansturen.....	28
5. Drukknop.....	30
6. Zoemer.....	32
7. Potentiometer.....	34
8. RGB-LED	37
Tutorials voor gevorderden.....	39
9. LDR-fotoresistor	40
10. Servomotor	43
11. OLED I2C SSD1306 beeldscherm	45
12. Joystickmodule	49
Tutorials met sensoren	51

13.	Regendruppelsensor	51
14.	HC-SR04 Ultrasonische sensor	54
15.	PIR-bewegingssensor.....	57
16.	DHT11 Sensor	60
17.	SW-420 Trillingssensor.....	62
18.	Vlam Sensor	64
19.	Sensor voor geluidsdetectie	66
20.	Bodemvochtigheidssensor.....	68
21.	Infrarood IR-sensor	70
APPENDIX: Samenvattende tabel voor MicroPython		72

Inbegrepen in de SCRAPY KIT

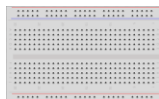
**GPIO Breakout Board
Raspberry Pi Pico (2st)**



**MB-102 voedingsmodule
(1 stuks)**



**Witte breadboard 830
stuks (1pc) + 400 stuks
(1pc)**



**Drukknop (1st) &
Knopdop (1st)**



Zoemer (1pc)



**Weerstanden 220 Ohm (5
stuks) + 1k Ohm (5
stuks)**



LDR-fotoresistor (1pc)



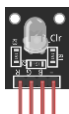
100uF condensator (1pc)



**LED 3mm blauw (1pc),
groen (1pc), rood (1pc)
geel (1pc)**



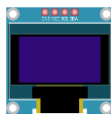
RGB LED 5mm (1pc)



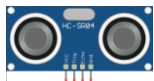
SG90 servomotor (1pc)



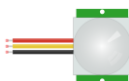
OLED I2C ICC (1pc)



**HC-SR04 ultrasone
sensor (1pc)**

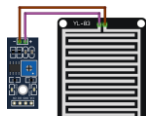


**PIR bewegingsmelder
sensor HC-SR501 (1pc)**

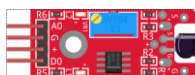


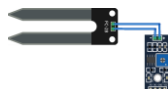
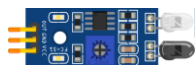
**DHT11 digitale
temperatuur en
vochtigheidssensor
(1pc)**



**Regendruppelsensor
(1pc)**

**Roterende potentiometer
lineair B1k Ohm (1pc)**

**SW-420 trillingssensor
(1pc)**

Vlam sensor (1pc)

**Hoge gevoeligheid
Sound Detectie Sensor
(1pc)**

**Bodemhygrometer /
vochtdetectiesensor (1
st)**

**Infrarood IR-
sensormodule KY-032
(1pc)**

Joystickmodule (1 stuks)

**USB naar micro-USB
kabel 1m (1pc)**

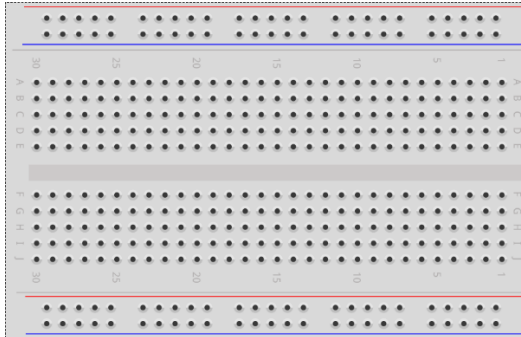
**6 x AA 1,5V batterijen
(1pc) + houder (1pc)**

Startkabels (6 stuks)

**Kunststof schroeven (10
stuks), kunststof moeren
(6 stuks), kunststof
pilaren (10 stuks)**


Componenten Uitleg

1. Wat is een breadboard?



Een breadboard is een plastic bord met kleine gaatjes waarin je gemakkelijk elektronische componenten (transistors, weerstanden, chips, enz.) kunt plaatsen om een prototype te maken (bouwen en testen) van een elektronisch circuit. De binnenkant bestaat uit rijen kleine metalen clips om de aan te sluiten draden vast te houden.

Op de meeste breadboards staan rijen cijfers, letters en plus- en mintekens geschreven. Het doel van de labels is om je te helpen bepaalde gaten op het breadboard te vinden, zodat je aanwijzingen kunt volgen bij het bouwen van een circuit.

De lange stroken aan de 2 zijden van het breadboard zijn meestal gemarkeerd met rood en blauw of rood en zwart en met respectievelijk plus (+) en min (-) tekens. Deze rijen worden de bussen of rails genoemd en worden meestal gebruikt om elektrische voeding aan de schakeling te leveren wanneer deze is aangesloten op een voeding (batterij of externe voeding).

De positieve "bus" is rood gemarkeerd, heeft het plusteken (+) en levert de voeding.

De negatieve "bus" is blauw of zwart gemarkeerd, heeft het minteken (-) en vormt de aarde.

Voordelen van het gebruik van een Breadboard:

- Maakt het eenvoudiger om eenvoudige en complexe schakelingen snel te controleren en schakelingen in hun beginstadium gemakkelijk te verifiëren.
- Gemakkelijk aan te passen.
- Flexibel.
- Geen gaten boren.
- Solderen is niet nodig.
- Eenvoudige debugging van circuits en programma's.

2. Wat is een weerstand?



Een weerstand is een klein pakketje weerstand. Door het in een circuit te gebruiken, wordt de stroom met een bepaalde hoeveelheid verminderd. Om de weerstand van een weerstand te bepalen is er een patroon van gekleurde banden.

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%



(Afbeelding credit: Future Owns) beschikbaar op
<https://www.tomshardware.com/how-to/resistor-color-codes>

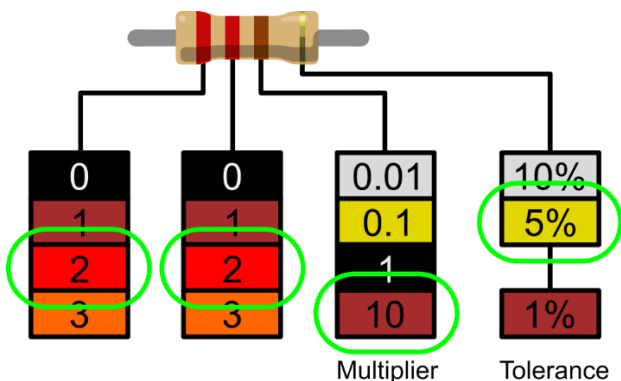
Algemene kleurcodes voor weerstanden en hun gebruik:

Type weerstand	4-bands kleurencode	5-bands kleurencode	Algemeen gebruik
220Ohm	Rood-Rood-Bruin-Goud	Rood-Rood-Zwart-Goud	LED Licht Bescherming
1K Ohm (1Kilohm)	Bruin-Zwart-Rood-Goud	Bruin-Zwart-Zwart-Bruin-Goud	LED-bescherming, spanningsdeler

Weerstanden hebben geen polariteit, dus ze kunnen in elke richting in een schakeling worden gebruikt. Maar om de juiste kleurcodewaarden voor weerstanden te bepalen, moeten we de gekleurde banden op de weerstand begrijpen.

Op een typische hobby weerstand met vier banden zijn er drie kleuren in een groep. Dit zijn de eerste, tweede significante cijfers en de vermenigvuldiger. De laatste band is de tolerantie van de weerstand, een foutmarge zo je wilt. Voor de meeste hobbyisten is een tolerantie van 5% (goud) perfect en gebruikelijk.

Weerstand van 220 Ohm (4-band)

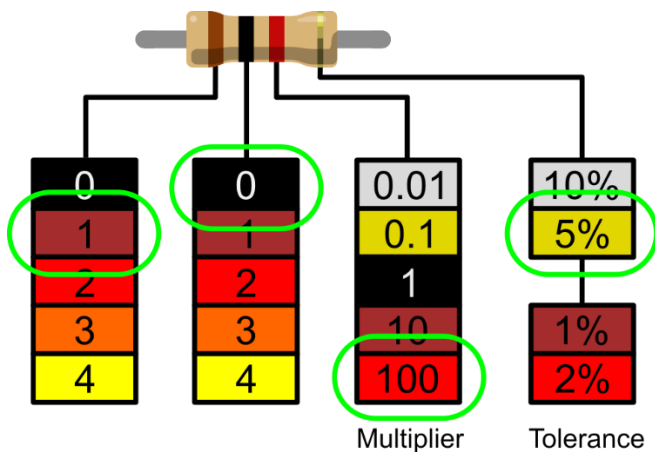


(Afbeelding credit: Future) beschikbaar op <https://www.tomshardware.com/how-to/resistor-color-codes>

1. Het eerste significante cijfer is rood en met behulp van de decoder kunnen we zien dat rood een waarde van 2 heeft.
2. Het tweede significante cijfer is ook rood, dus dat geeft ons 22.
3. De vermenigvuldiger is bruin en dit decodeert naar 10. Als we 22 vermenigvuldigen met 10, krijgen we 220.
4. De laatste band, tolerantie, is goud. Goud is 5%, wat betekent dat we een weerstand kunnen accepteren met een foutmarge van 5%.

Voor makers die meer precisie nodig hebben, zijn er ook weerstanden met vijf banden die een derde significant cijfer hebben. Het extra cijfer zorgt voor duidelijkheid die essentieel kan zijn in schakelingen die gevoelig zijn voor weerstand, zoals wetenschappelijke en technische instrumenten.

Weerstand van 1K Ohm (4-band)



(Afbeelding credit: Tom's Hardware) beschikbaar op <https://www.tomshardware.com/how-to/resistor-color-codes>

1. De lijn 1st is bruin en met behulp van de decoder kunnen we zien dat de waarde 1 is.

2. De lijn 2nd is zwart, dus dat geeft ons 10.
3. De vermenigvuldiger is rood en dit decodeert naar 100. Als we 10 vermenigvuldigen met 100 krijgen we 1000.
4. De laatste band, tolerantie, is goud. Goud is 5%, wat betekent dat we een weerstand kunnen accepteren met een foutmarge van 5%.

3. Wat is een condensator?



Een condensator is een apparaat dat elektrische energie opslaat in een elektrisch veld. Het is een passieve elektronische component met twee aansluitingen. Hij bestaat uit twee elektrische geleiders die van elkaar gescheiden zijn door een afstand. De ruimte tussen de geleiders kan gevuld zijn met vacuüm of met een isolerend materiaal dat diëlectricum wordt genoemd. (Wikipedia)

De 100uF condensator is een elektrolytische ontkoppelingscondensator. Deze condensatoren zijn geweldige transiënt/spanningsonderdrukkers en het gebruik van een condensator tussen de voeding en de massa van het circuit zorgt voor een soepele stroomtoevoer.

4. Wat is een diode?

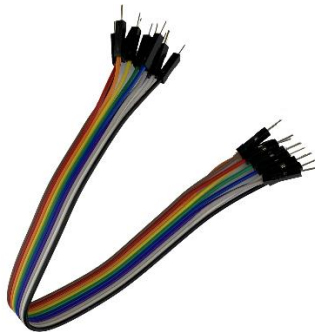


Een diode is een elektronische component met twee uitgangen die hoofdzakelijk in één richting stroom geleidt (asymmetrische geleiding); hij heeft een lage (idealiter nul) weerstand in de ene

richting en een hoge (ideaaliter oneindige) weerstand in de andere richting.

De meest voorkomende functie van een diode is het doorlaten van een elektrische stroom in één richting (de voorwaartse richting van de diode genoemd), terwijl de stroom in de tegenovergestelde richting wordt geblokkeerd (de omgekeerde richting). Als zodanig kan de diode worden gezien als een elektronische versie van een terugslagklep. Dit eenrichtingsgedrag wordt gelijkrichting genoemd en wordt gebruikt om wisselstroom (ac) om te zetten in gelijkstroom (dc). Als gelijkrichters kunnen diodes worden gebruikt voor taken als het extraheren van modulatie uit radiosignalen in radio-ontvangers. (Wikipedia <https://en.wikipedia.org/wiki/Diode>)

5. Wat is een startkabel?



Jumper kabels/draden zijn gewoon draden met connectorpinnen aan elk uiteinde, waardoor ze gebruikt kunnen worden om twee punten met elkaar te verbinden zonder te solderen. Jumper draden worden meestal gebruikt met breadboards en andere prototyping tools om het makkelijk te maken om een circuit te veranderen als dat nodig is. De kleurvariatie van de draden kan als voordeel worden gebruikt om onderscheid te maken tussen soorten verbindingen, zoals massa of stroom.

Doorverbindingsdraden zijn er meestal in drie versies: mannelijk-naar-man, mannelijk-naar-vrouwelijk en vrouwelijk-naar-vrouwelijk. Het verschil zit hem in het uiteinde van de draad. Mannelijke uiteinden hebben een uitstekende pin en kunnen in dingen gestoken worden, terwijl vrouwelijke uiteinden dat niet hebben en gebruikt worden om dingen in te steken. Doorverbindingsdraden van mannelijk naar mannelijk komen het meest voor. Bij het verbinden van twee poorten op een breadboard is meestal een male-to-male draad nodig. (<https://blog.sparkfuneducation.com/what-is-jumper-wire>)

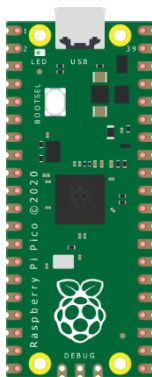
Projectvoorbereiding

1. Leesmij voor gebruik

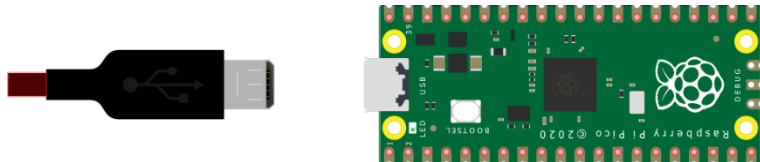
OPMERKING: Aangezien het hier om experimenten met circuits gaat, kan een verkeerde aansluiting of kortsluiting uw Pico-ontwikkelbord beschadigen. Controleer het circuit altijd opnieuw voordat u de voeding aansluit.

2. Raspberry Pi Pico

Dit is de Raspberry Pi Pico:

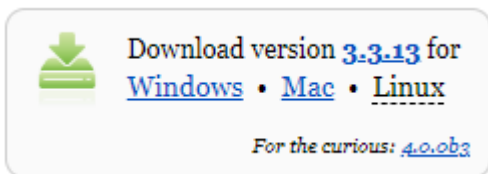


Sluit de micro-USB-kabel aan op de poort aan de linkerkant van het bord.



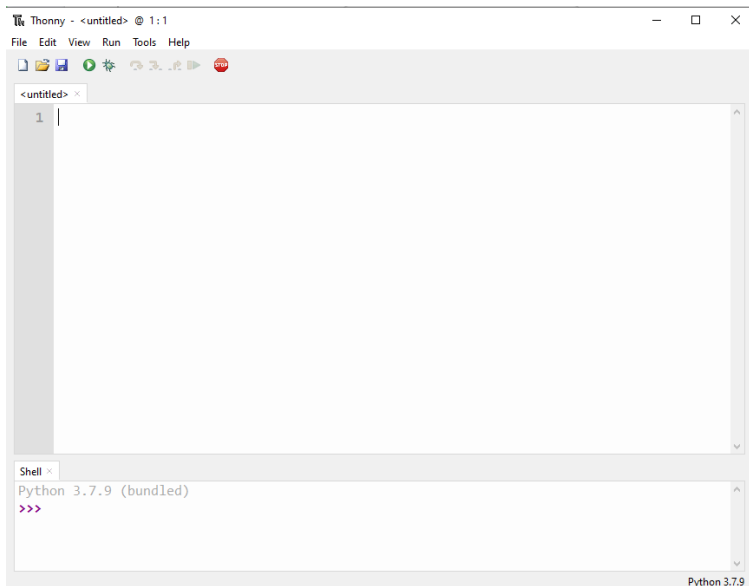
3. Installeer Thonny IDE

Ga naar <https://thonny.org> en kies het juiste besturingssysteem. Volg de instructies om de installatie te voltooien.



In deze handleiding zijn alle tutorials geprogrammeerd in Windows 10, met behulp van een Raspberry Pi Pico en de juiste firmware.

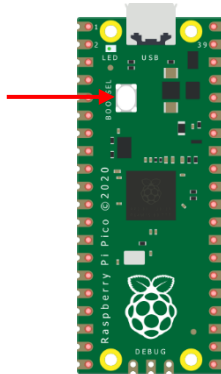
Nadat de installatie is voltooid, open je Thonny vanaf je computer.



4. Installatie firmware

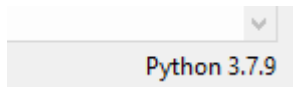
De Raspberry Pi Pico kan geprogrammeerd worden met een Python-variant, MicroPython genaamd. Om MicroPython op de Pico te gebruiken, moet je eerst de firmware installeren.

Stap 1: Zoek de BOOTSEL-knop op je Raspberry Pi Pico.

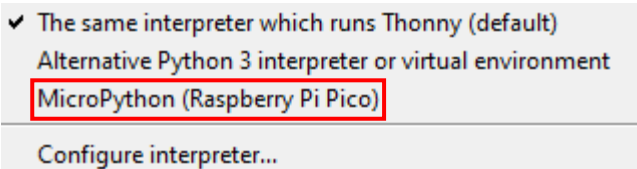


Stap 2: Druk op de knop BOOTSEL en houd deze ingedrukt terwijl je het andere uiteinde van de micro-USB-kabel aansluit op je computer.

Stap3: Rechtsonder in Thonny zie je de versie van Python die je momenteel gebruikt.

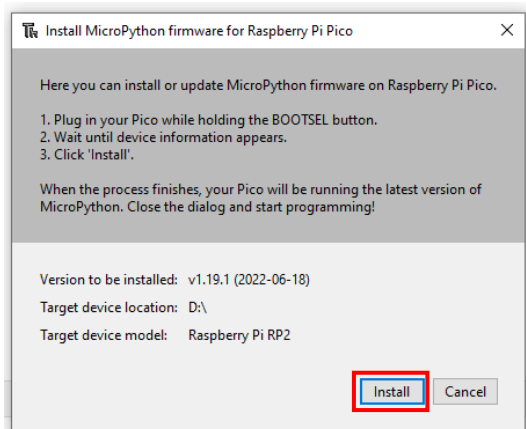


Klik op de Python-versie en kies MicroPython (Raspberry Pi Pico).

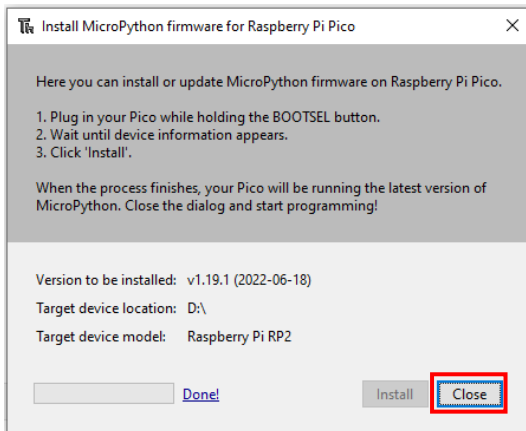


Als je deze optie niet ziet, controleer dan of de kabel goed is aangesloten op de Pico en/of je computer.

Stap 4: Er verschijnt een dialoogvenster waarin u wordt gevraagd de nieuwste firmwareversie op uw Pico te installeren. Klik op de knop **Installeren** om de firmware naar je Pico te kopiëren.



Stap 5: Wacht tot de installatie is voltooid en klik op **Sluiten**.



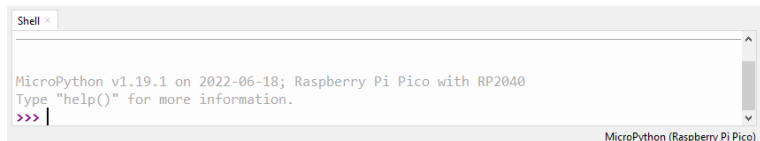
Je hoeft dit proces niet elke keer te herhalen als je de Raspberry Pi Pico aansluit op je computer, dus sluit hem de volgende keer gewoon aan en je kunt aan de slag.

5. Inleiding tot MicroPython programmeren

Je gaat nu Thonny IDE gebruiken om wat eenvoudige Python code te draaien om kennis te maken met Thonny's Shell en MicroPython.

Zorg er eerst voor dat je Raspberry Pi Pico is aangesloten op je computer en dat je de MicroPython-interpreter hebt geselecteerd zoals uitgelegd in de vorige sectie.

Het Shell-paneel onderaan de Thonny-editor zou er zo uit moeten zien:



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

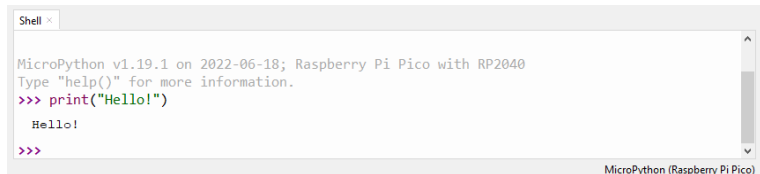
MicroPython (Raspberry Pi Pico)

Thonny is klaar om te communiceren met je Pico met behulp van REPL (read-eval-print loop), waarmee je direct code kunt schrijven op de Shell en output kunt krijgen.

Typ het volgende commando:

```
print("Hallo!")
```

Druk vervolgens op de Enter-toets en zie de volgende uitvoer:



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

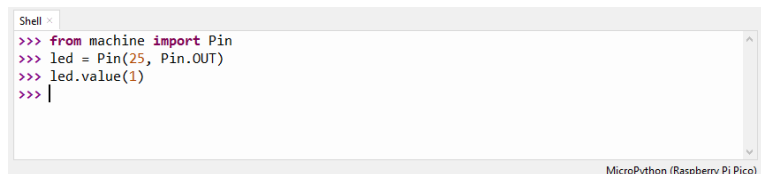
MicroPython (Raspberry Pi Pico)

Met MicroPython kun je hardwarespecifieke modules toevoegen, zoals `machine`, die je kunt gebruiken om je Pico te

programmeren. In het volgende voorbeeld gebruik je de machine module om de LED van de Pico aan te zetten.

Schrijf de volgende code in Thonny's Shell:

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



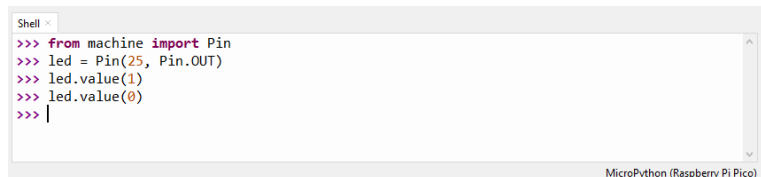
```
Shell
>>> from machine import Pin
>>> led = Pin(25, Pin.OUT)
>>> led.value(1)
>>> |
```

Druk op de Enter-toets en onmiddellijk gaat de LED van de Pico aan.



Schrijf de volgende code om de LED uit te schakelen:

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



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

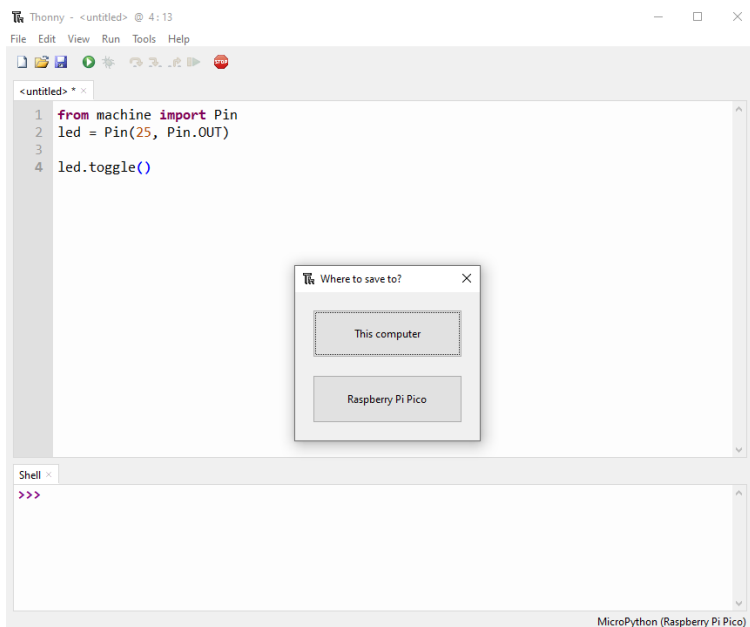
In de rest van dit onderdeel ga je je eerste "echte" programma schrijven dat de LED op het bord laat knipperen telkens als je je programma uitvoert.

Thonny Shell is handig om snelle commando's uit te proberen en te controleren of alles goed werkt. Langere programma's moeten echter worden opgeslagen in een .py-bestand. Met Thonny kun je programma's direct op de Raspberry Pi Pico opslaan en vervolgens uitvoeren.

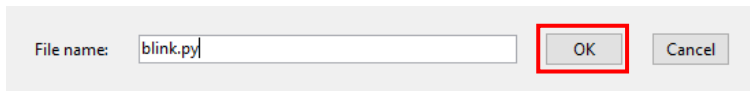
Open Thonny Python en schrijf in het hoofdvenster van de editor de volgende code:

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

Sla nu je programma op door op het pictogram Opslaan linksboven te klikken of door op Ctrl+S op je toetsenbord te drukken.



Thonny zal je vragen waar je je programma wilt opslaan. Kies de **Raspberry Pi Pico**. Sla het bestand op als *blink.py* en klik op **OK**. Je moet altijd de *.py* extensie toevoegen zodat Thonny het bestand herkent als een Python bestand.



Elke keer dat je nu op het Afspeel-pictogram klikt, zou je moeten zien dat het LED-lampje aan en uit gaat.

Als je nog een stap verder gaat met je code, kun je de LED aan boord in een bepaald tempo laten knipperen.

Schrijf de volgende code en sla je programma op met dezelfde naam als hierboven.

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

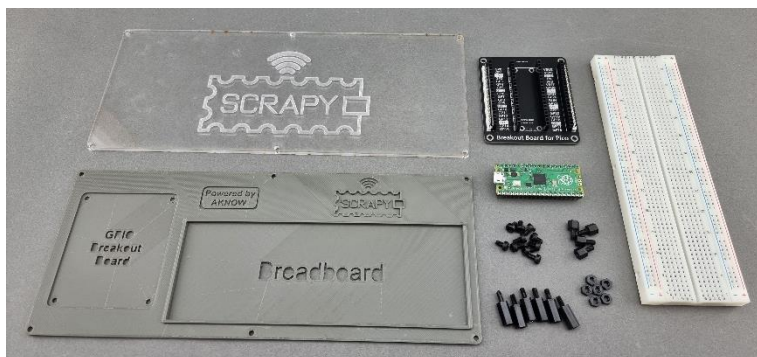
Wanneer je nu het programma uitvoert, zal de LED op het bord elke 1 seconde knipperen totdat we het programma stoppen. Om het programma te stoppen kun je op het STOP-pictogram klikken of op Ctrl+C op je toetsenbord drukken.

In toekomstige tutorials leren we hoe we andere elektronica en sensoren kunnen toevoegen en aansturen en hoe we programma's kunnen maken die met elkaar kunnen communiceren.

Kit montage

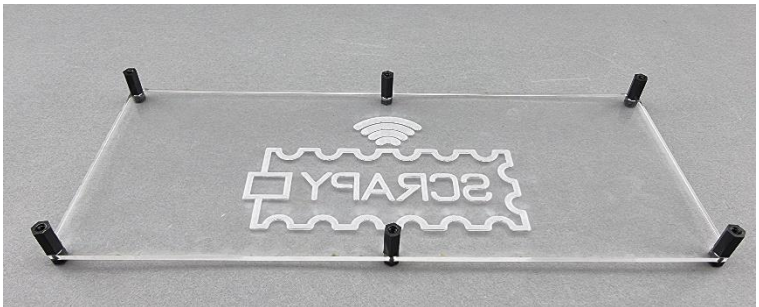
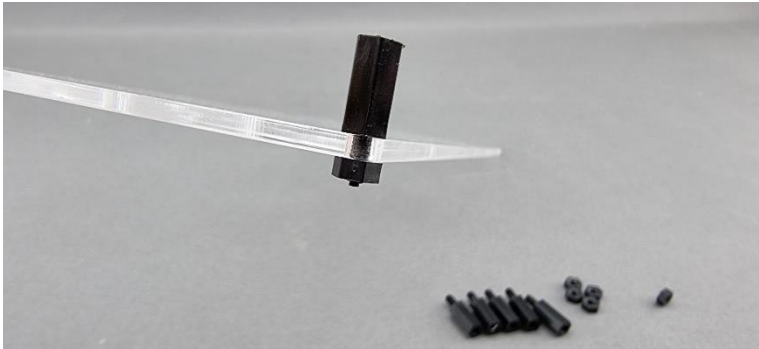
De SCRAPY Kit bestaat uit de volgende onderdelen:

- 1x 3D geprint stuk
- 1 x stuk plexiglas
- 1 x Raspberry Pi Pico
- 1 x GPIO-uitbreidingskaart
- 1 x Broodplank (830st)
- 10 x plastic schroeven
- 6 x 12mm plastic pilaren
- 4 x 6mm plastic pilaren
- 6 x Plastic moeren

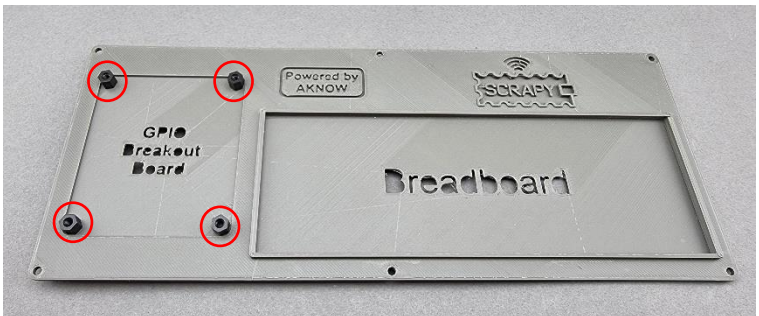


De montageprocedure is eenvoudig en kan in 6 stappen worden voltooid:

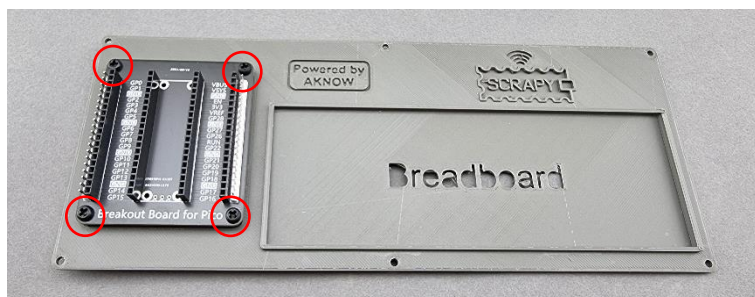
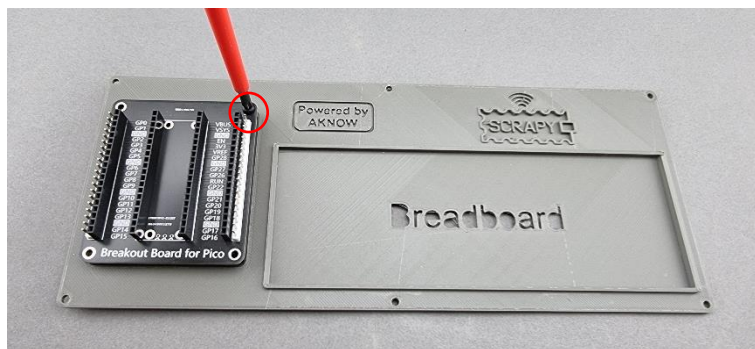
Stap 1: Monteer de 6 x 12mm plastic pilaren met de 6 x plastic moeren op het plexiglas stuk.



Stap 2: Plaats op het 3D-geprinte stuk 4 x 6 mm plastic pillars bij het "GPIO Breakout Board" gedeelte.

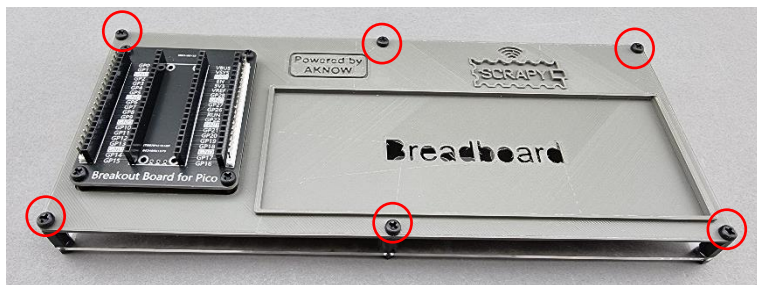


Stap 3: Monteer het GPIO breakout bord op de 4 pilaren met 4 plastic schroeven.

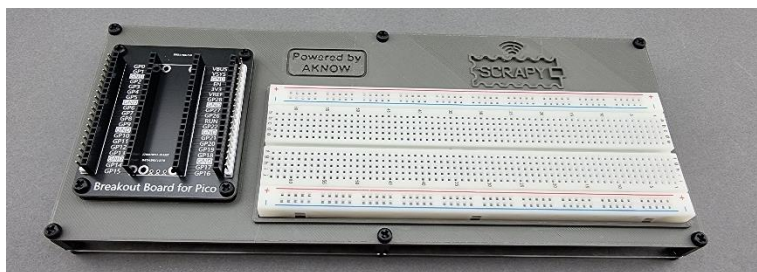


Stap 4: Monteer het 3D geprinte object met het plexiglas object met 6 plastic schroeven op de 6 plastic pilaren die je in Stap 1 hebt gemonteerd.

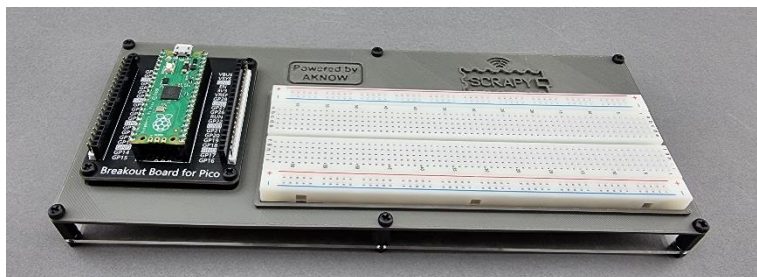




Stap 5: Verwijder de beschermende tape onder de 830 stuks breadboard en plaats deze op het "Breadboard"-gedeelte van de Kit. Zorg ervoor dat de positieve (+) kant van het breadboard bovenop zit, zoals de volgende afbeelding aangeeft.



Stap 6: Plaats de Raspberry Pi Pico bovenop het GPIO Breakout Board en druk ertegen tot alle GPIO-pinnen correct zijn geplaatst. Zorg ervoor dat het micro-USB slot aan de bovenkant zit zoals de afbeelding aangeeft.



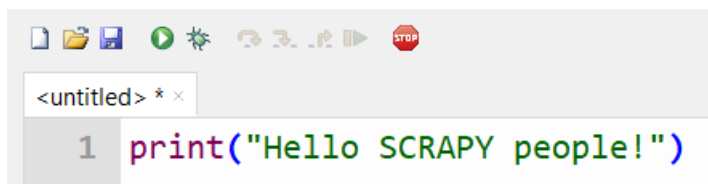
Basishandleidingen

0. "Hallo SCRAPY mensen!"

In deze basistutorial leren we hoe we een eenvoudig bericht kunnen afdrucken in Thonny met behulp van Python.

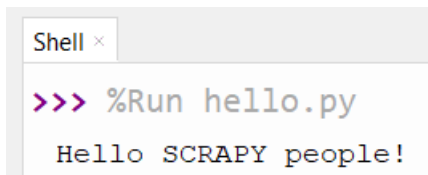
De functie print()

1. Ga naar de Thonny editor, schrijf de volgende code en druk op het play icoontje. Thonny zal je vragen om je programma eerst op te slaan. Sla het op onder de naam hello.py.



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

2. Controleer het shell-venster.



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

3. Goed gedaan! Je hebt zojuist je eerste Python-programma gemaakt.

De printfunctie is een ingebouwde Python-functie waarmee we tekst kunnen afdrucken in de shell. De functie kan ook parameters aannemen. Maak een nieuw programma en kopieer de volgende code, druk dan op play en kijk hoe de tekst in de shell verschijnt.

```

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!
    
```

Zoals je kunt zien in het shell venster, drukt elke afdrufunctie tekst af op een aparte regel. Als je echter het "\n" (newline karakter) gebruikt, kun je van regel wisselen in dezelfde afdrufopdracht.

```

Shell x
>>> %Run hello.py

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

Oefening

Gebruik de printfunctie om in 3 afzonderlijke regels "Vaarwel SCRAPY mensen!" af te drukken met slechts één printopdracht.

4. Een LED aansturen

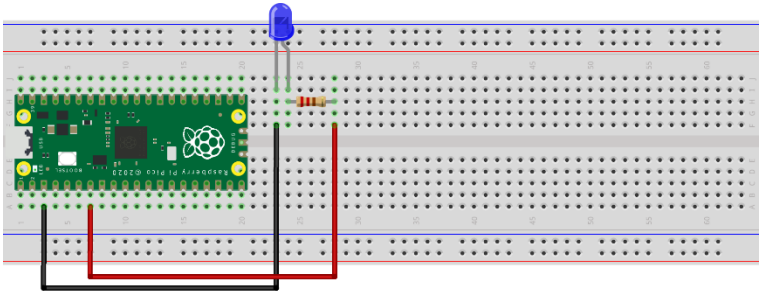
Beschrijving

In deze tutorial leer je hoe je een LED-lampje kunt aansluiten en bedienen. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `led.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x LED (elke kleur)
- 1 x Pico-breadbordkit
- 2 x mannelijke doorverbindingen
- 1 x 220 Ohm weerstand

Bedradingsschema



fritzing

- sluit het lange uiteinde (+) van de LED aan op een weerstand van 220 ohm
- sluit de weerstand aan op GPIO5 (rode kabel)
- sluit het korte uiteinde (-) van de LED aan op een GND-pin

Code

MicroPython-code voor de tutorial:

```

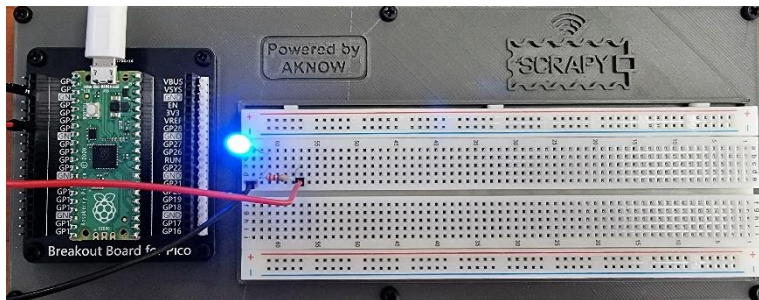
Thonny - Raspberry Pi Pico :: /led.py @ 9:1
File Edit View Run Tools Help

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

Voorbeeldafbeelding

Afbeelding van hoe het circuit eruit ziet met de meegeleverde hardware:



5. Druknop

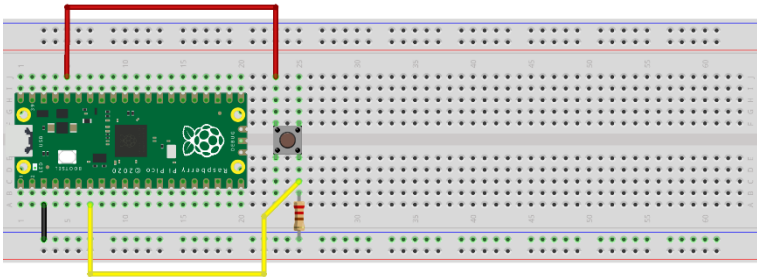
Beschrijving

In deze tutorial leer je hoe je een drukknop kunt aansluiten en bedienen. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `button.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Druknop (elke kleur)
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x 220 Ohm weerstand
- 1 x Knooppod

Bedradingsschema



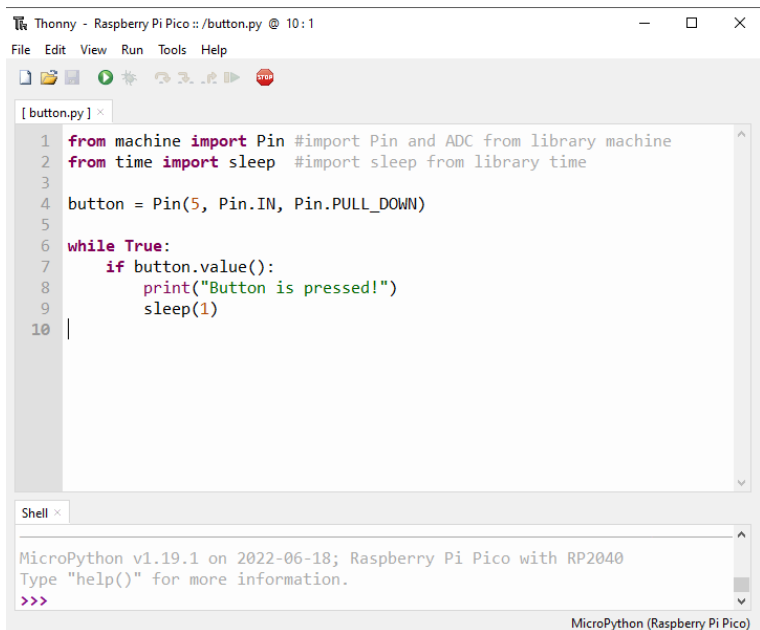
fritzing

- sluit de knop linksboven aan op de 3v3-pin (rode kabel)
- sluit de knop rechtsonder aan op GPIO5 (gele kabel)
- sluit een GND-pin aan op de (-) rail (zwarte kabel)

- sluit de weerstand van 220 Ohm aan op de (-) rail en de knop rechtsonder

Code

MicroPython-code voor de tutorial:

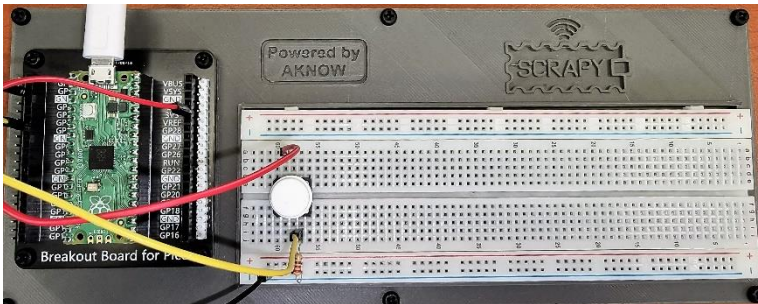


```

Thonny - Raspberry Pi Pico ::/button.py @ 10:1
File Edit View Run Tools Help
[button.py] x
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 x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
MicroPython (Raspberry Pi Pico)
    
```

Voorbeeldafbeelding

Afbeelding van hoe het circuit eruit ziet met de meegeleverde hardware:



6. Zoemer

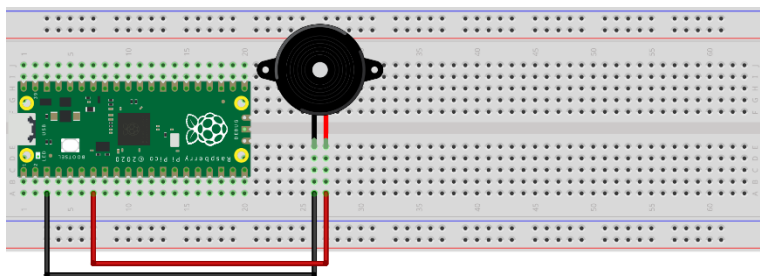
Beschrijving

In deze tutorial leer je hoe je een buzzer aansluit en aanstuurt. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `buzzer.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 2 x man-naar-man doorverbindingsdraden
- 1 x Zoemer

Bedradingsschema

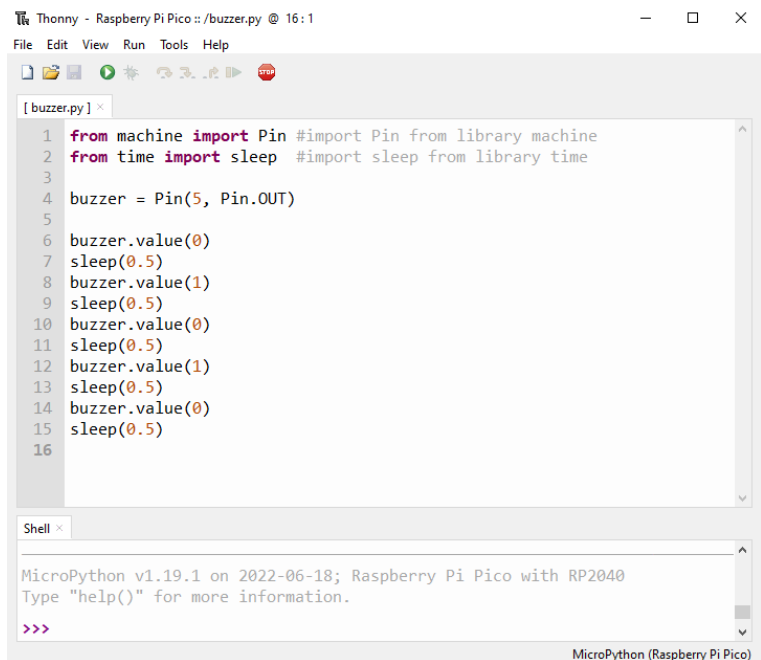


fritzing

- sluit het lange uiteinde (+) van de zoemer aan op pin GPIO5
- sluit het korte uiteinde (-) van de zoemer aan op een GND-pin

Code

MicroPython-code voor de tutorial:



```

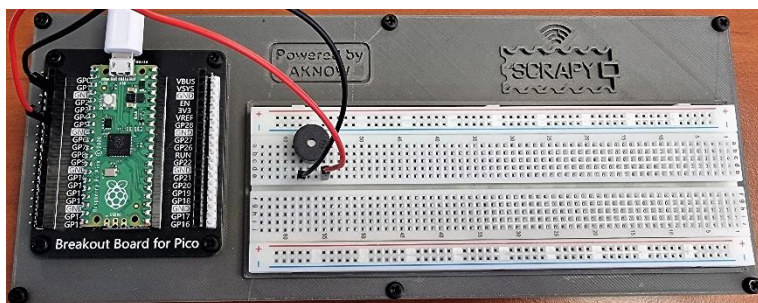
Thonny - Raspberry Pi Pico :: /buzzer.py @ 16: 1
File Edit View Run Tools Help

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

Voorbeeldafbeelding

Afbeelding van hoe het circuit eruit ziet met de meegeleverde hardware:



7. Potentiometer

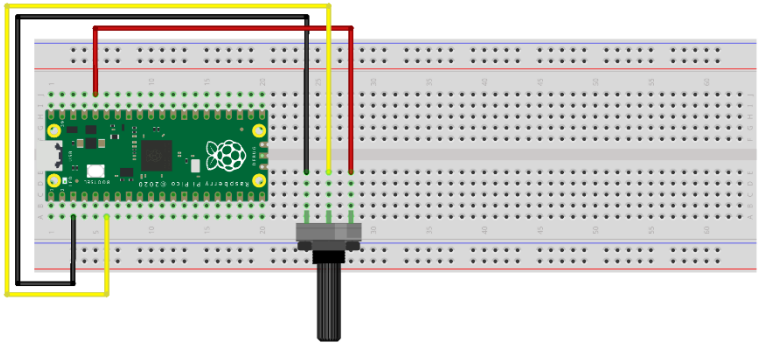
Beschrijving

In deze tutorial leer je hoe je een potentiometer aansluit en aanstuurt. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `pot.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x potentiometer

Bedradingsschema



fritzing

- zwarte kabel moet worden aangesloten op pin GND (pin 3)
- gele kabel moet worden aangesloten op pin GPIO5
- rode kabel moet worden aangesloten op voedingspin 3V3
- draai de potentiometer naar links zodat deze uit staat

Code

MicroPython-code voor de tutorial:

Thonny - Raspberry Pi Pico :: /pot.py @ 9:1

File Edit View Run Tools Help

```
[ pot.py ] x
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)
9
```

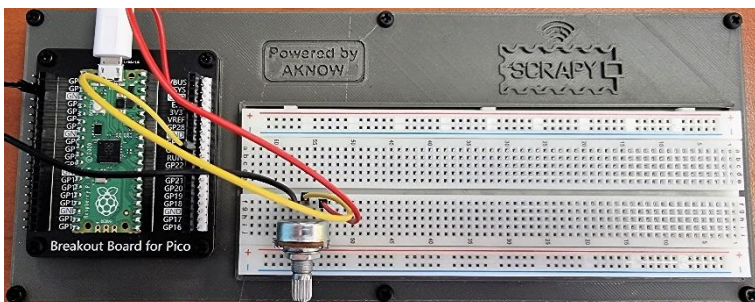
Shell x

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

MicroPython (Raspberry Pi Pico)

Voorbeeldfoto

Afbeelding van hoe het circuit eruit ziet met de meegeleverde hardware:



8. RGB-LED

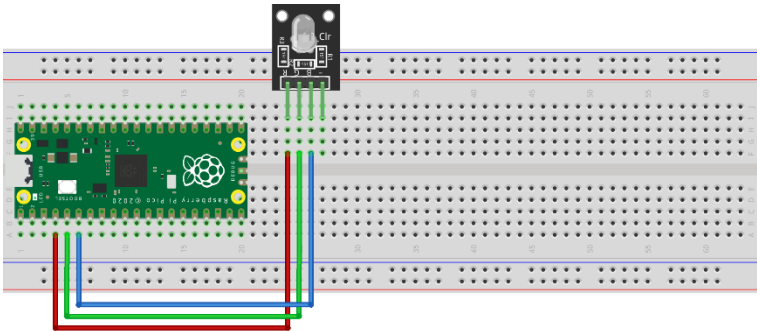
Beschrijving

In deze tutorial leer je hoe je een RGB LED-module kunt aansluiten en aansturen. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `rgb.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 4 x man-naar-man doorverbindingsdraden
- 1 x RGB LED-module

Bedradingsschema



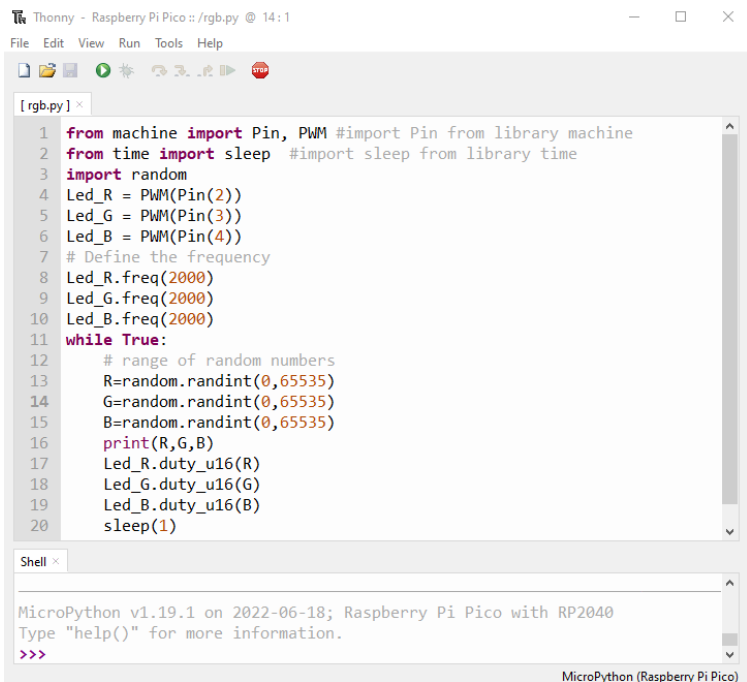
fritzing

- rode kabel moet worden aangesloten op GPIO2
- groene kabel moet worden aangesloten op GPIO3
- blauwe kabel moet worden aangesloten op GPIO4

- er is geen GND-aansluiting nodig

Code

MicroPython-code voor de tutorial:



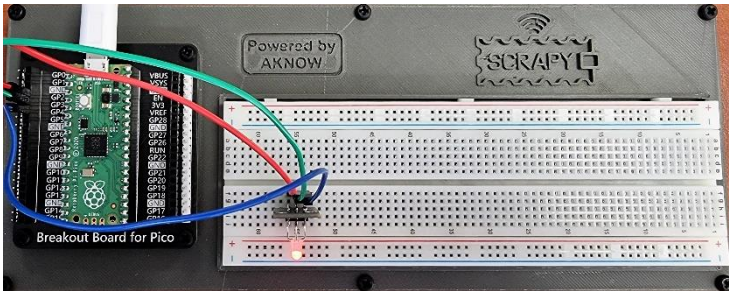
```

Thonny - Raspberry Pi Pico :: /rgb.py @ 14:1
File Edit View Run Tools Help
[rgb.py] x
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 x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
MicroPython (Raspberry Pi Pico)
    
```

Voorbeeldafbeelding

Afbeelding van hoe het circuit eruit ziet met de meegeleverde hardware:



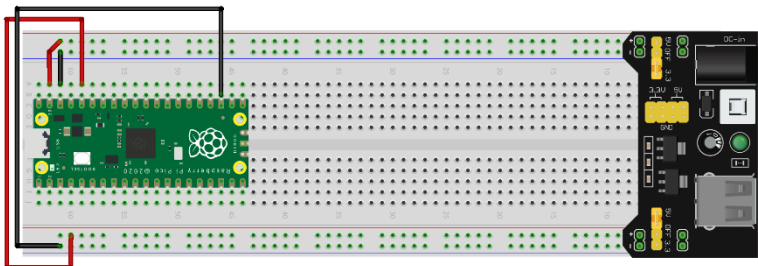
Handleidingen voor gevorderden

Voordat we verder gaan met dit en het volgende onderdeel, moeten we een paar wijzigingen aanbrengen in onze SCRAPY Kit. Het gaat om de toevoeging van een paar extra componenten en hun connectiviteit.

Onderdelen

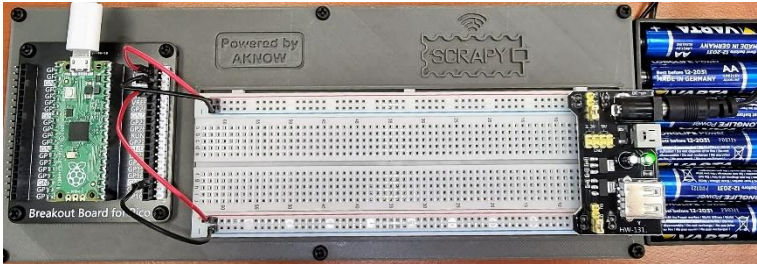
- 1 x 6-AA batterij
- 1 x MB-102 voedingsmodule
- 4 x man-naar-man doorverbindingsdraden

Volg het schema om de nieuwe componenten aan te sluiten:



fritzing

Circuit



- deze opstelling wordt gebruikt voor de resterende tutorials
- aansluitingen bovenkant: VSYS 5V ((+) rood) en GND ((-) zwart)
- aansluitingen onderkant: 3V3 ((+) rood) en GND ((-) zwart)

9. LDR-fotoresistor

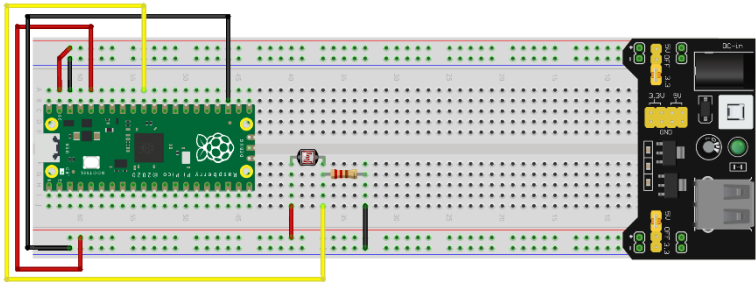
Beschrijving

In deze tutorial leer je hoe je een LDR-fotoresistor aansluit en aanstuurt. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `ldr.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x 220 Ohm weerstand
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x LDR-fotoresistor

Bedradingsschema

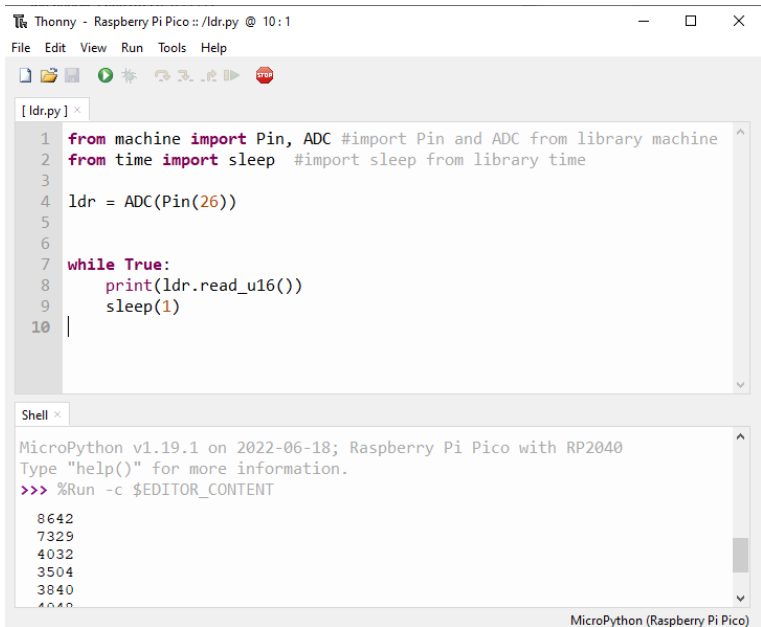


fritzing

- linkerkant van LDR is verbonden met 3V rail ((+) rood)
- rechterkant van LDR is verbonden met een weerstand van 220 Ohm en met GPIO26 ADC (gele kabel) pin
- rechterkant van weerstand is verbonden met GND-rail ((-) zwart)

Code

MicroPython-code voor de tutorial:



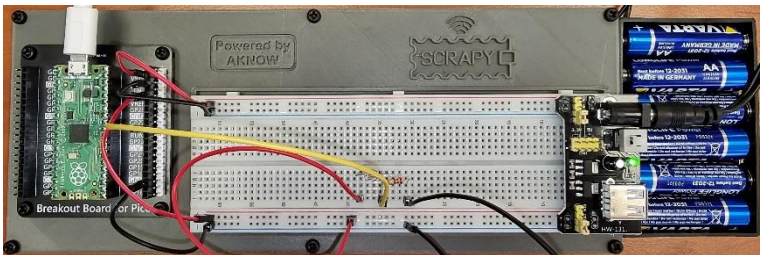
```
Thonny - Raspberry Pi Pico :: /ldr.py @ 10:1
File Edit View Run Tools Help

[ ldr.py ] x
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 x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
8642
7329
4032
3504
3840
4040
MicroPython (Raspberry Pi Pico)
```

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



10. Servomotor

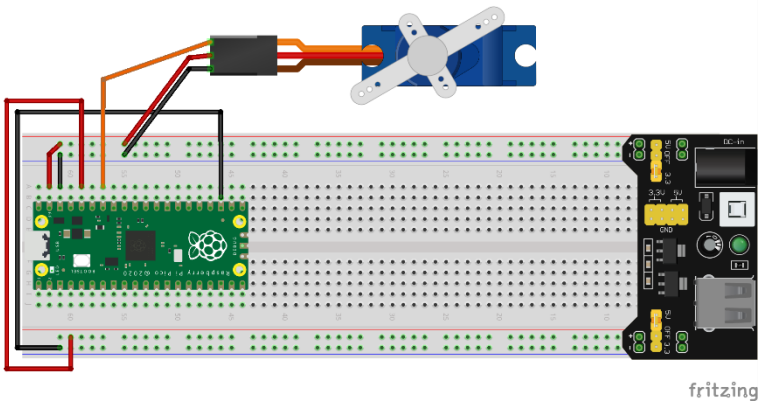
Beschrijving

In deze tutorial leer je hoe je een servomotor aansluit en aanstuurt. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `servo.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x SG-90 servomotor

Bedradingsschema

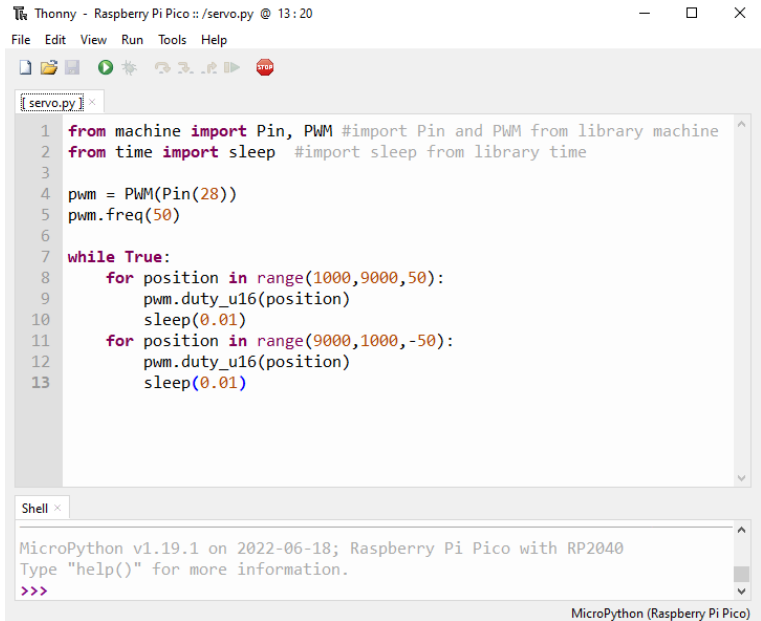


- rode kabel is verbonden met 5V rail (+)
- zwart/bruine kabel is verbonden met GND-rail (-)

- oranje kabel is verbonden met GPIO28 ADC-pin

Code

MicroPython-code voor de tutorial:



```

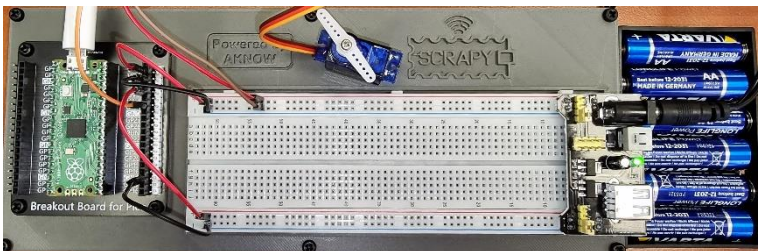
Thonny - Raspberry Pi Pico ::/servo.py @ 13: 20
File Edit View Run Tools Help

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(0.01)
11    for position in range(9000,1000,-50):
12        pwm.duty_u16(position)
13        sleep(0.01)

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

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



11. OLED I2C SSD1306 beeldscherm

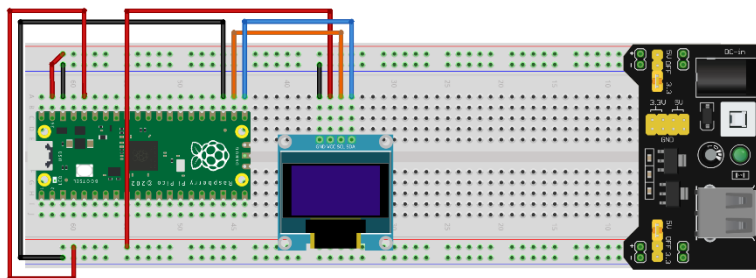
Beschrijving

In deze tutorial leer je hoe je het I2C ICC OLED-display kunt aansluiten en bedienen. Open Thonny Python, ga dan naar Bestand→ Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `oled.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 4 x man-naar-man doorverbindingsdraden
- 1 x OLED I2C SSD1306 beeldscherm

Bedradingsschema



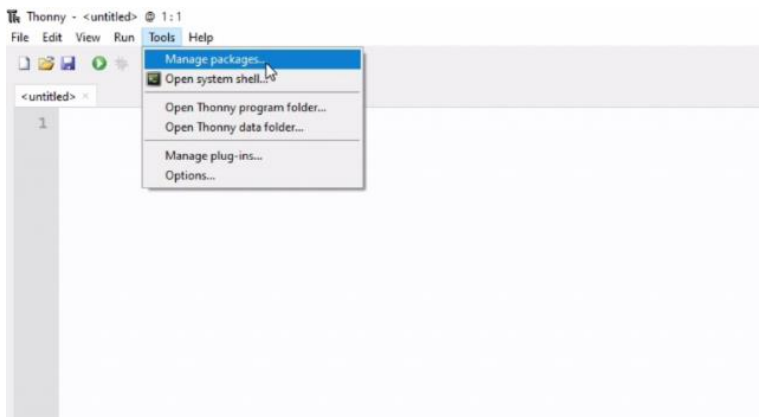
fritzing

- rode kabel is verbonden met 3v3 rail (+)
- zwarte kabel is verbonden met GND-rail (-)
- oranje kabel is verbonden met GPIO17 I2C0 SCL pin
- blauwe kabel is verbonden met GPIO26 I2C0 SDA pin

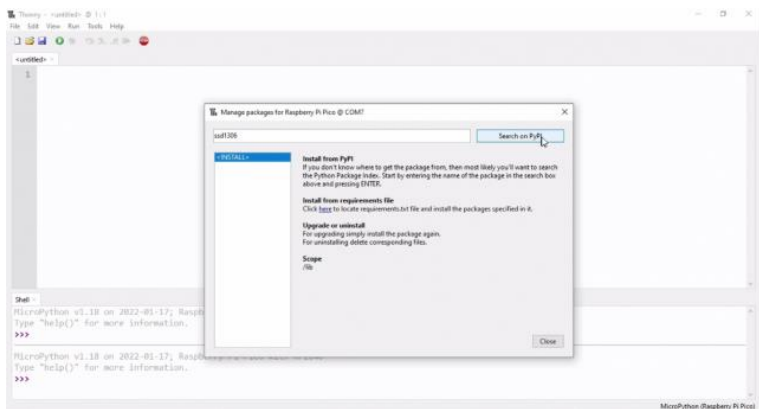
Code

Voordat we beginnen met het programmeren van het OLED display, moeten we eerst het SSD1306 pakket toevoegen aan onze RPi Pico. Volg daarvoor de volgende stappen:

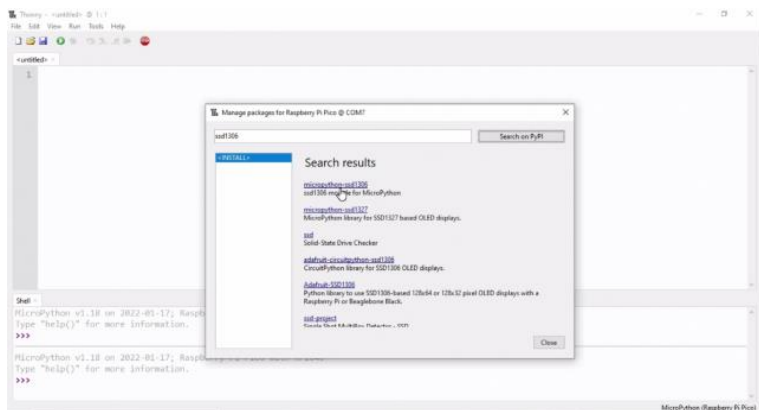
1. Open Thonny en ga naar **Extra** → **Beheer pakketten...**



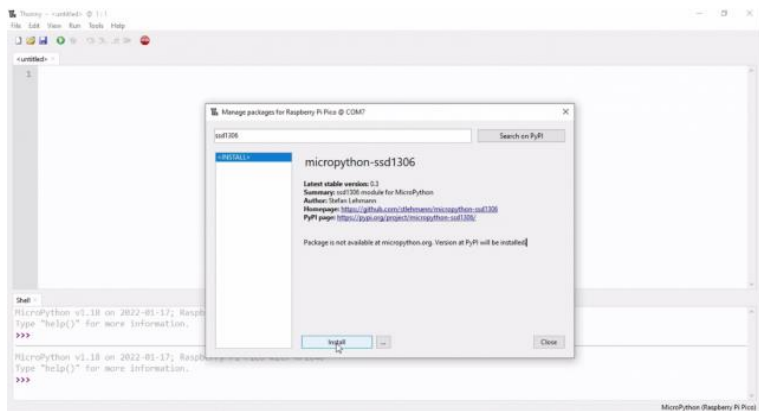
2. Typ **SSD1306** in het venster Beheer pakketten en klik op **Zoeken**.



3. Zodra het zoeken voorbij is, klik je op *micropython-ssd1306*.



4. Klik in het volgende venster op *Installeren*.



5. Wacht op de installatie van het pakket en klik op Sluiten.

Nu zijn we klaar om verder te gaan met het programmeren van het OLED display.

MicroPython-code voor de tutorial:

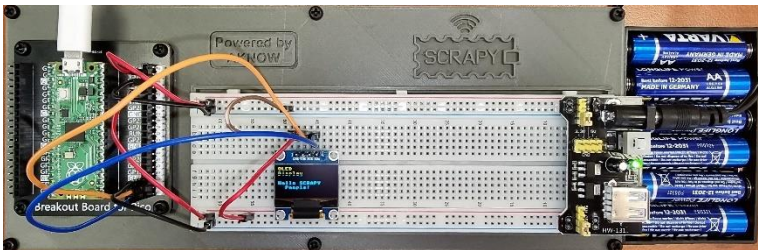
```
Thonny - Raspberry Pi Pico :: /oled.py @ 20: 1
File Edit View Run Tools Help
[ oled.py ] x
1 from machine import Pin, I2C
2 import ssd1306
3
4 WIDTH =128
5 HEIGHT = 64
6
7 PIN_SCL = 17
8 PIN_SDA = 16
9
10 i2c = I2C(0,scl=Pin(PIN_SCL),sda=Pin(PIN_SDA))
11 oled = ssd1306.SSD1306_I2C(WIDTH,HEIGHT,i2c)
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 |

Shell x
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)

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



12. Joystickmodule

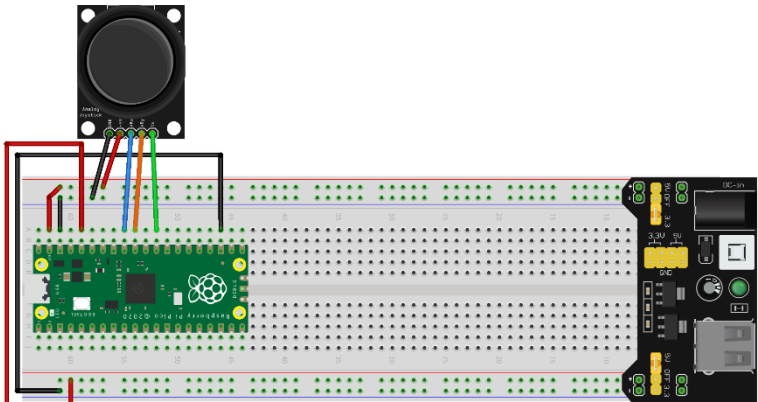
Beschrijving

In deze tutorial leer je hoe je een joystickmodule kunt aansluiten en besturen. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `joystick.py`. Dan is het tijd om de elektronica aan te sluiten en uw programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 5 x man-naar-man doorverbindingsdraden
- 1 x joystickmodule

Bedradingsschema



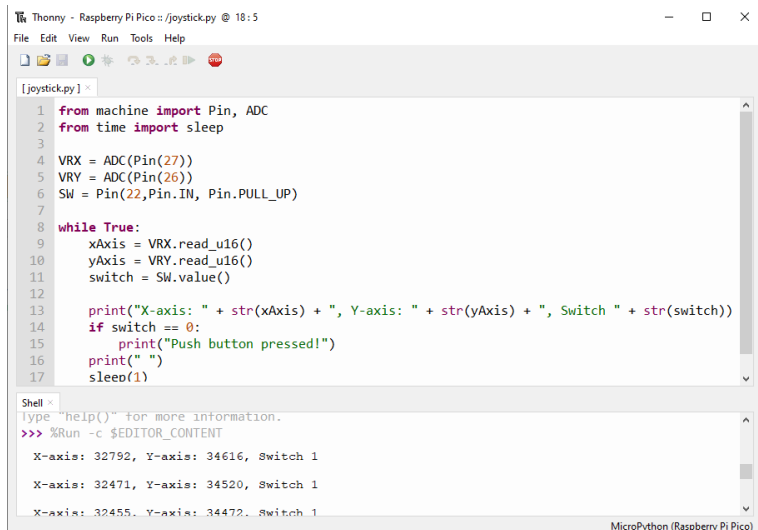
fritzing

- +5V (rode kabel) is aangesloten op 5V rail (+)

- GND (zwarte kabel) is verbonden met GND-rail (-)
- VRx (blauwe kabel) is verbonden met GPIO27 ADC1-pin
- VRy (oranje kabel) is verbonden met GPIO26 ADC0-pin
- SW-kabel (groen) is aangesloten op pin GPIO22

Code

MicroPython-code voor de tutorial:



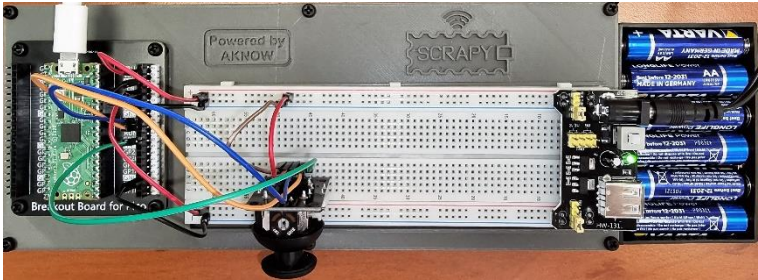
```
[joystick.py]
1 from machine import Pin, ADC
2 from time import sleep
3
4 VRX = ADC(Pin(27))
5 VRY = ADC(Pin(26))
6 SW = Pin(22, Pin.IN, Pin.PULL_UP)
7
8 while True:
9     xAxis = VRX.read_u16()
10    yAxis = VRY.read_u16()
11    switch = SW.value()
12
13    print("X-axis: " + str(xAxis) + ", Y-axis: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

Shell
type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
X-axis: 32792, Y-axis: 34616, Switch 1
X-axis: 32471, Y-axis: 34520, Switch 1
X-axis: 32455, Y-axis: 34472, Switch 1

MicroPython (Raspberry Pi Pico)
```

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



Tutorials met sensoren

13. Regendruppelsensor

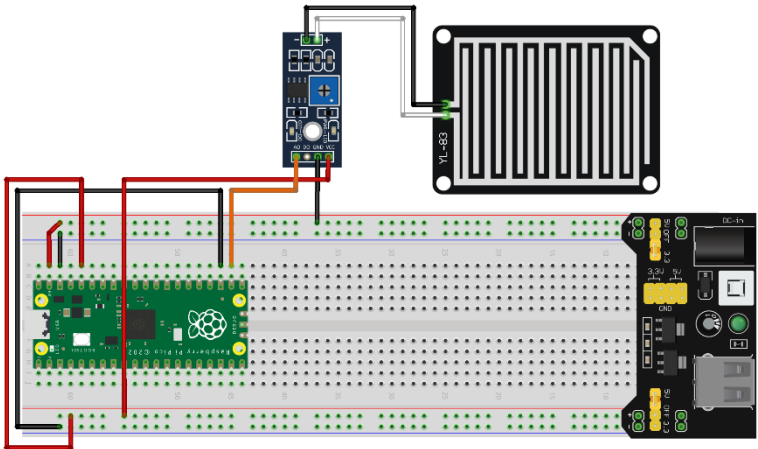
Beschrijving

In deze tutorial leer je hoe je de regendruppelsensor kunt aansluiten en besturen. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `raindrop.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x regendruppelsensor

Bedradingsschema



fritzing

- 3v3V (rode kabel) is verbonden met 3v3V rail (+)
- GND (zwarte kabel) is verbonden met GND-rail (-)
- DO (oranje kabel) is verbonden met pin GPIO17

Code

MicroPython-code voor de tutorial:

Thonny - Raspberry Pi Pico :: /raindrop.py @ 15:1

File Edit View Run Tools Help

```
[ raindrop.py ] x
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 |
```

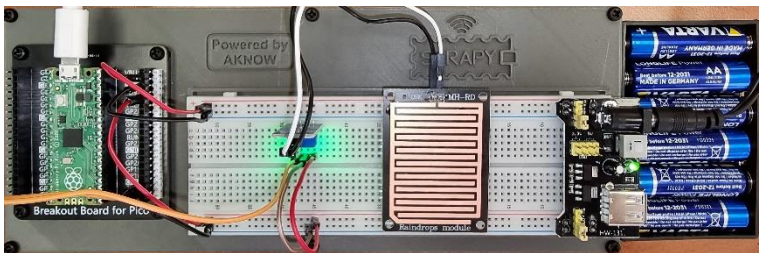
Shell x

```
MicroPython v1.20.0 on 2023-04-26; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
```

MicroPython (Raspberry Pi Pico)

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



14. HC-SR04 Ultrasonic Sensor

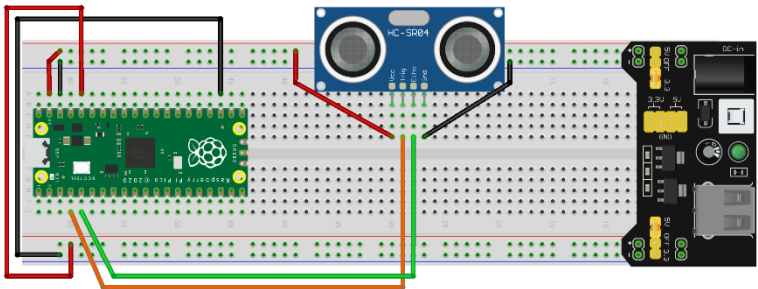
Beschrijving

In deze tutorial leer je hoe je de HC-SR04 ultrasonic sensor kunt aansluiten en besturen. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `ultrasonic.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x HC-SR04 ultrasonic sensor

Bedradingsschema



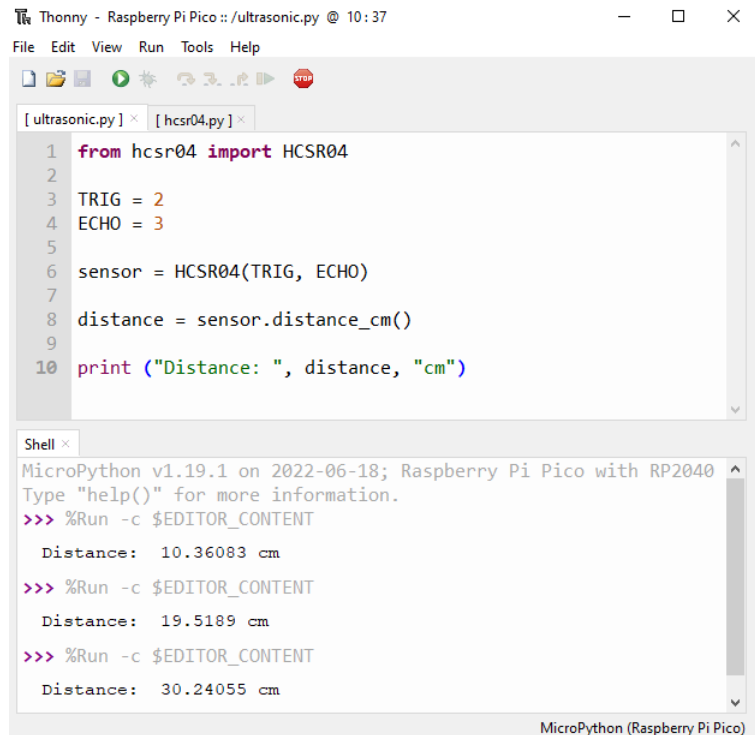
fritzing

- VCC (rode kabel) is verbonden met 5V rail (+)
- GND (zwarte kabel) is verbonden met GND-rail (-)
- TRIG (oranje kabel) is verbonden met GPIO2 pin
- ECHO (groene kabel) is verbonden met GPIO3 pin

Code

Om de HC-SR04 ultrasone sensor te gebruiken, kunnen we ons eigen programma ontwikkelen of een van de bibliotheken gebruiken die online beschikbaar zijn, bijvoorbeeld op [Github](#). Als je ervoor kiest om de *hcsr04.py* bibliotheek te downloaden, moet je deze opslaan in je Pico onder dezelfde naam.

MicroPython-code die een bestaande bibliotheek gebruikt:

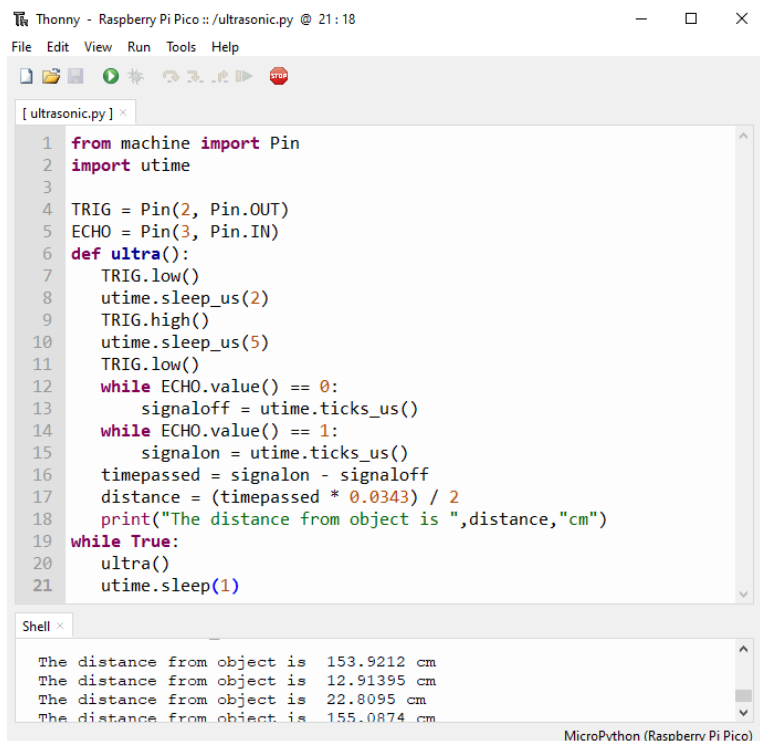


```

Thonny - Raspberry Pi Pico :: /ultrasonic.py @ 10:37
File Edit View Run Tools Help
[ ultrasonic.py ] x [ hcsr04.py ] x
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 x
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:  19.5189 cm
>>> %Run -c $EDITOR_CONTENT
    Distance:  30.24055 cm
    
```

MicroPython-code zonder bestaande bibliotheek:



```

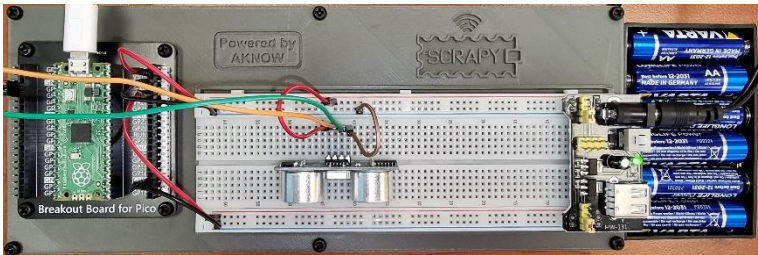
Thonny - Raspberry Pi Pico :: /ultrasonic.py @ 21:18
File Edit View Run Tools Help
[ultrasonic.py] x
1 from machine import Pin
2 import utime
3
4 TRIG = Pin(2, Pin.OUT)
5 ECHO = Pin(3, Pin.IN)
6 def ultra():
7     TRIG.low()
8     utime.sleep_us(2)
9     TRIG.high()
10    utime.sleep_us(5)
11    TRIG.low()
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 x
The distance from object is 153.9212 cm
The distance from object is 12.91395 cm
The distance from object is 22.8095 cm
The distance from object is 155.0874 cm

MicroPython (Raspberry Pi Pico)
    
```

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



15. PIR Bewegingssensor

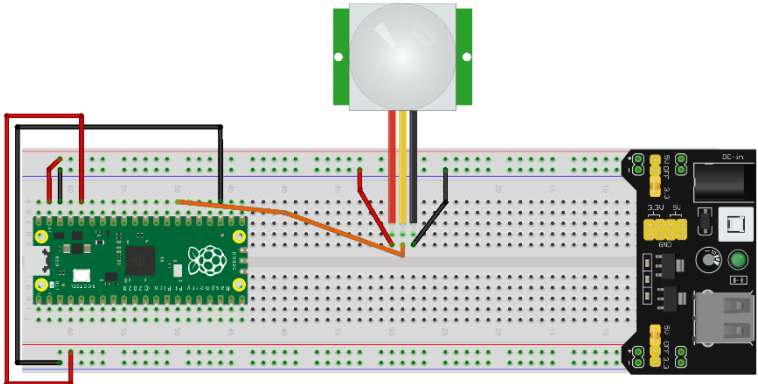
Beschrijving

In deze tutorial leer je hoe je de PIR bewegingssensor aansluit en aanstuurt. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `motion.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadboardkit
- 3 x verbindingsdraden van man naar vrouw
- 1 x PIR-bewegingssensor

Bedradingsschema

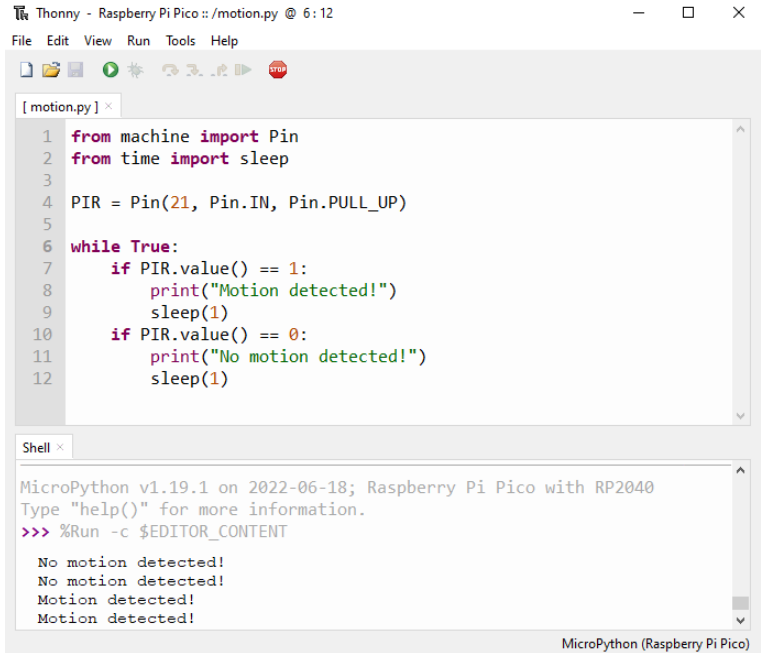


fritzing

- VCC (rode kabel) is verbonden met 5V rail (+)
- GND (zwarte kabel) is verbonden met GND-rail (-)
- OUT (oranje kabel) is verbonden met pin GPIO21

Code

MicroPython-code voor de tutorial:



Thonny - Raspberry Pi Pico :: /motion.py @ 6:12

File Edit View Run Tools Help

```
[ motion.py ] x
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)
```

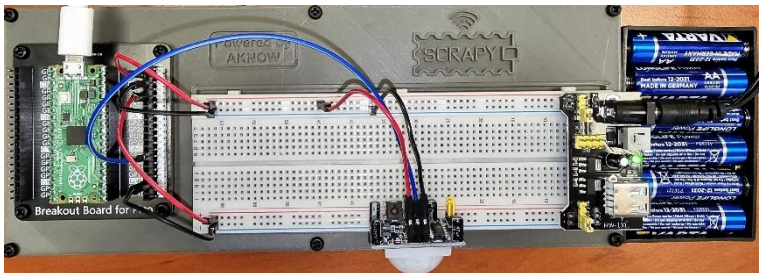
Shell x

```
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)

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



16. DHT11 Sensor

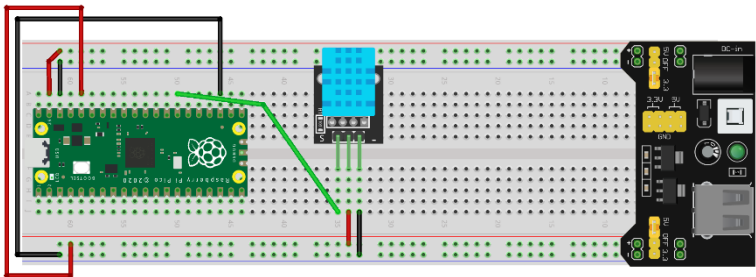
Beschrijving

In deze tutorial leer je hoe je de DHT11 temperatuur- en vochtigheidssensor kunt aansluiten en besturen. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `dht11.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x DHT11 temperatuursensor

Bedradingschema



fritzing

- VCC (rode kabel) is verbonden met 3v3 rail (+)
- GND (zwarte kabel) is verbonden met GND-rail (-)
- S (groene kabel) is verbonden met GPIO21 pin

Code

MicroPython-code voor de tutorial:

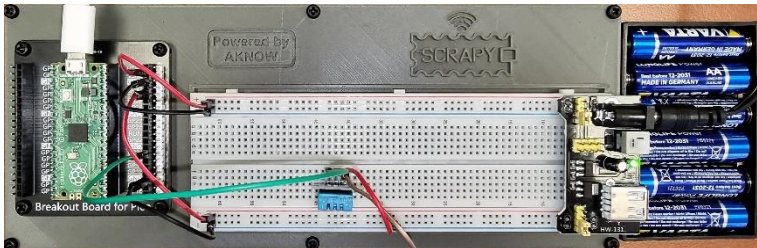
```
Thonny - Raspberry Pi Pico :: /dht11.py @ 12:38
File Edit View Run Tools Help
[dht11.py] x
1 from machine import Pin
2 from time import sleep
3 import dht
4
5 sensor = dht.DHT11(Pin(21))
6
7 while True:
8     sensor.measure()
9     temp = sensor.temperature()
10    hum = sensor.humidity()
11    print("Room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

Shell x
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)
```

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



17. SW-420 Trillingssensor

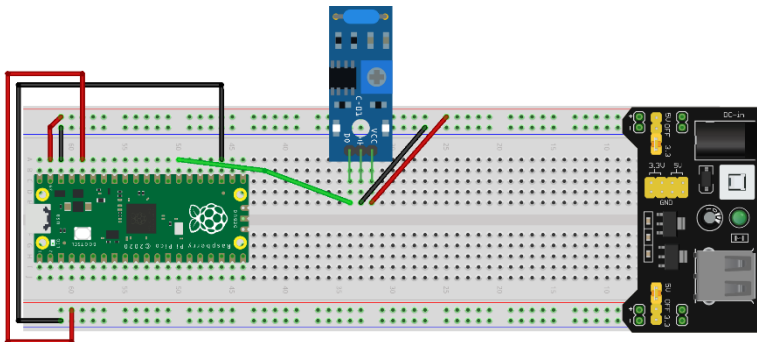
Beschrijving

In deze tutorial leer je hoe je de SW-420 vibratiesensor kunt aansluiten en besturen. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `vibration.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x SW-420 trillingssensor

Bedradingsschema

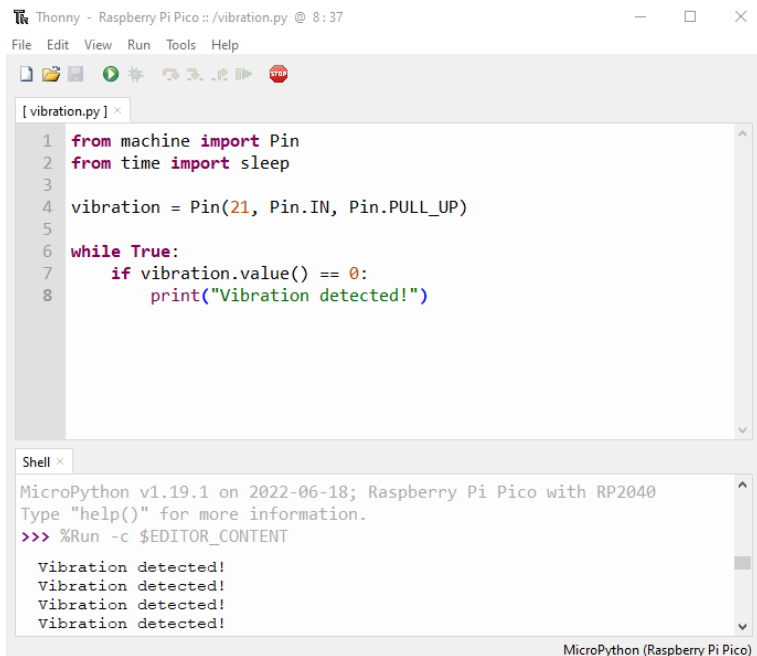


fritzing

- VCC (rode kabel) is verbonden met 5V rail (+)
- GND (zwarte kabel) is verbonden met GND-rail (-)
- DO (groene kabel) is verbonden met pin GPIO21

Code

MicroPython-code voor de tutorial:



```

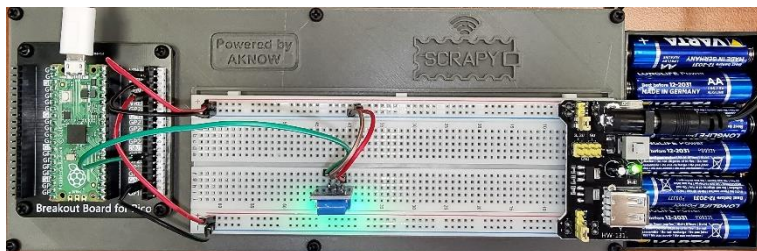
Thonny - Raspberry Pi Pico :: /vibration.py @ 8:37
File Edit View Run Tools Help

[vibration.py] x
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 x
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!
    
```

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



18. Vlam Sensor

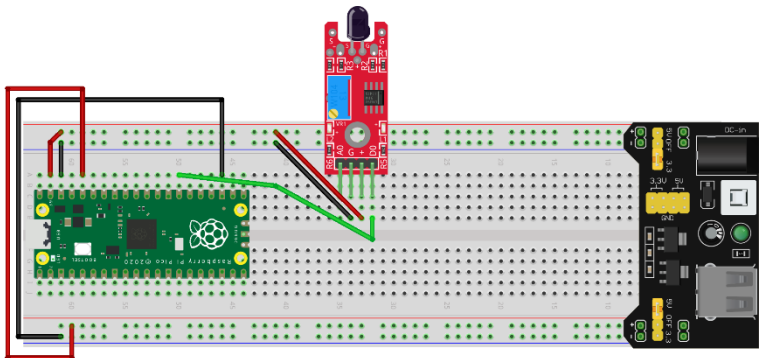
Beschrijving

In deze tutorial leer je hoe je de KY-026 vlamsensor aansluit en aanstuurt. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `flame.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x KY-026 vlamsensor

Bedradingschema

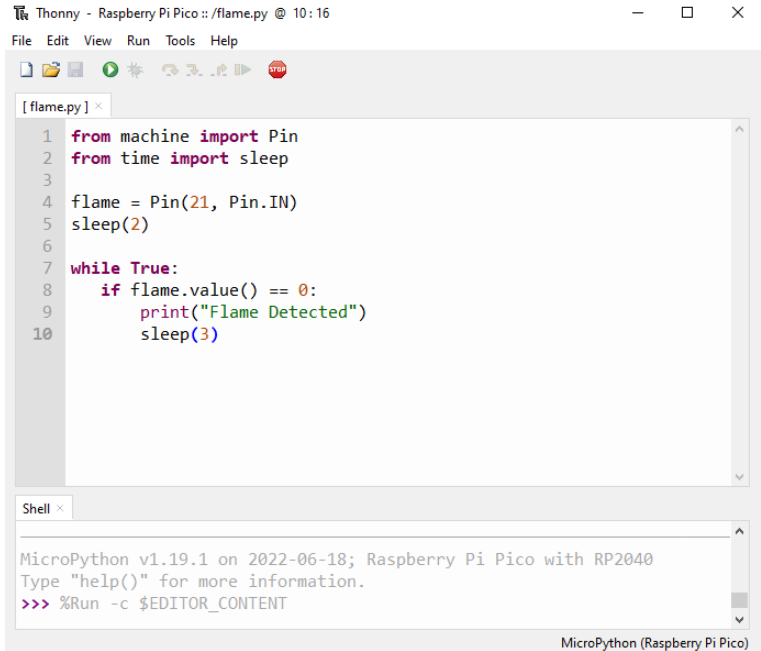


fritzing

- VCC (rode kabel) is verbonden met 5V rail (+)
- GND (zwarte kabel) is verbonden met GND-rail (-)
- DO (groene kabel) is verbonden met pin GPIO21

Code

MicroPython-code voor de tutorial:



```
Thonny - Raspberry Pi Pico :: /flame.py @ 10:16
File Edit View Run Tools Help

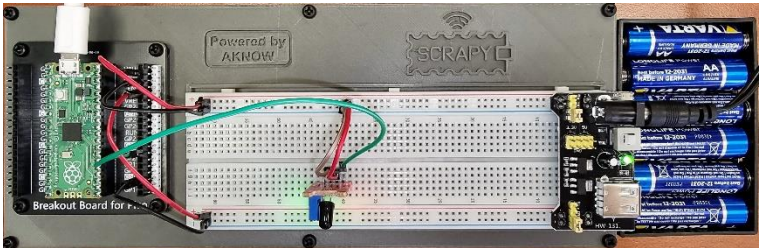
[flame.py] x
1 from machine import Pin
2 from time import sleep
3
4 flame = Pin(21, Pin.IN)
5 sleep(2)
6
7 while True:
8     if flame.value() == 0:
9         print("Flame Detected")
10        sleep(3)

Shell x
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)
```

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



19. Sensor voor geluidsdetectie

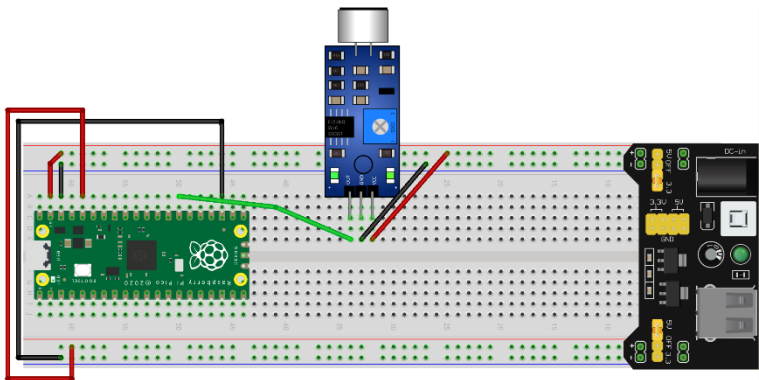
Beschrijving

In deze tutorial leer je hoe je de KY-037 geluidsdetectiesensor kunt aansluiten en besturen. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `sound.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x KY-037 geluidssensor

Bedradingsschema

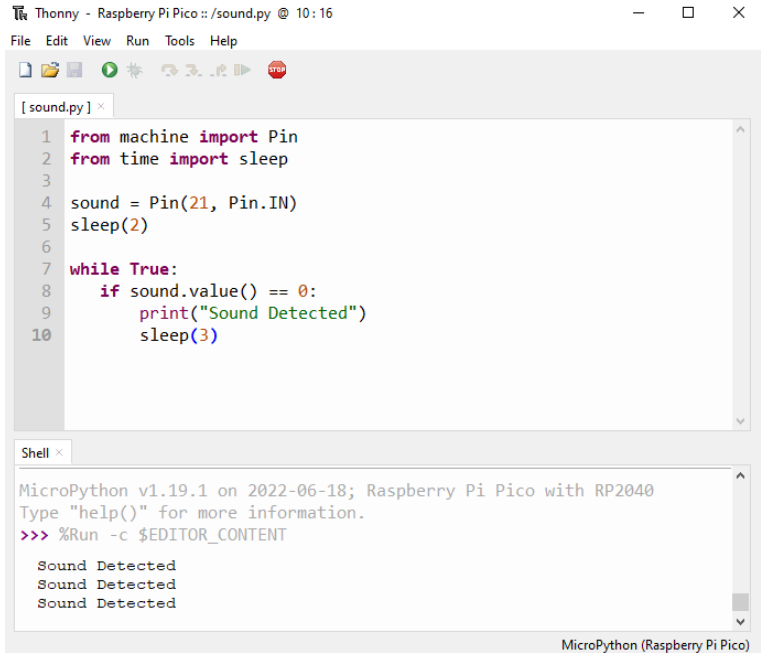


fritzing

- VCC (rode kabel) is verbonden met 5V rail (+)
- GND (zwarte kabel) is verbonden met GND-rail (-)
- DO/OUT (groene kabel) is verbonden met pin GPIO21

Code

MicroPython-code voor de tutorial:



```

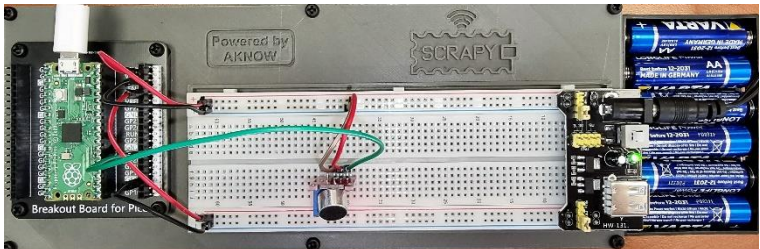
1  from machine import Pin
2  from time import sleep
3
4  sound = Pin(21, Pin.IN)
5  sleep(2)
6
7  while True:
8      if sound.value() == 0:
9          print("Sound Detected")
10         sleep(3)
    
```

```

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
    
```

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



20. Bodemvochtigheidssensor

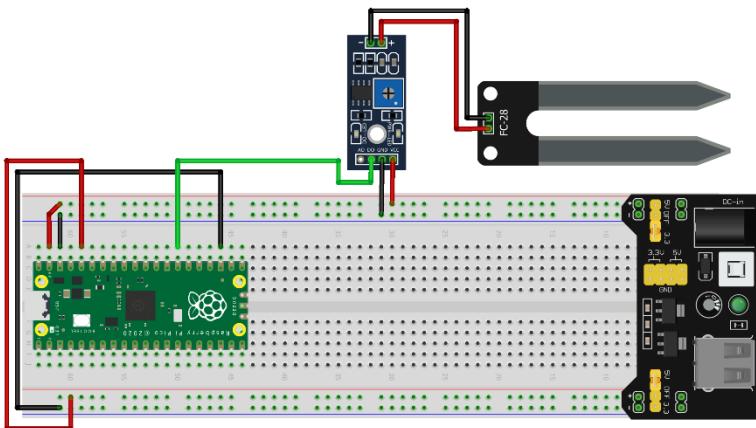
Beschrijving

In deze tutorial leer je hoe je de bodemvochtigheidssensor aansluit en aanstuurt. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `soil.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x KY-037 geluidssensor

Bedradingsschema

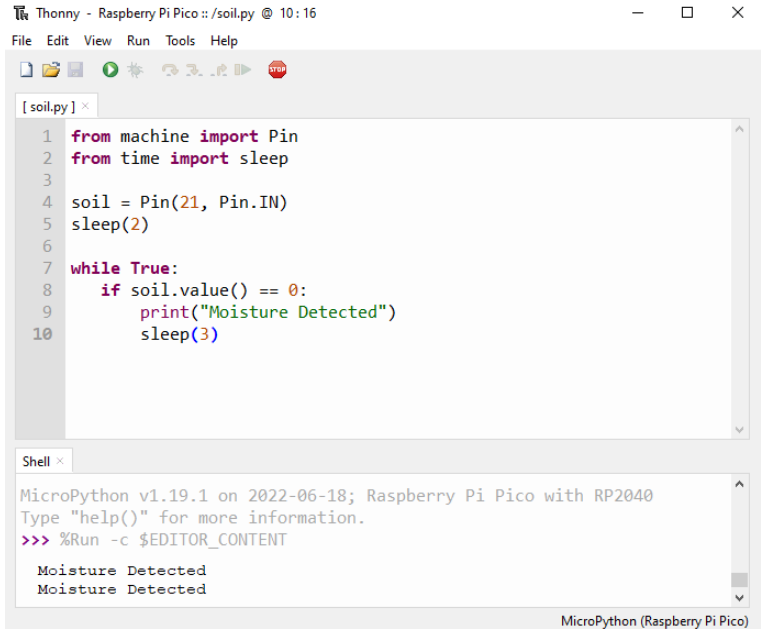


fritzing

- VCC (rode kabel) is verbonden met 5V rail (+)
- GND (zwarte kabel) is verbonden met GND-rail (-)
- DO/OUT (groene kabel) is verbonden met pin GPIO21

Code

MicroPython-code voor de tutorial:



```

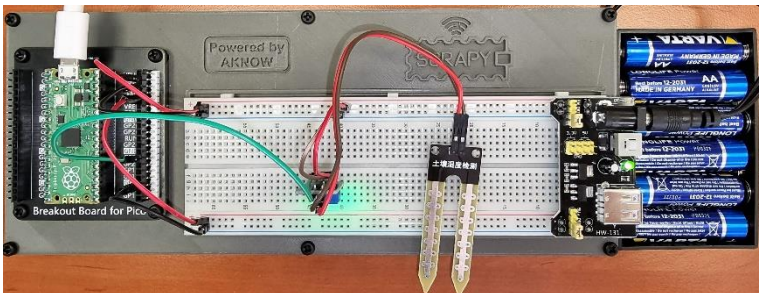
Thonny - Raspberry Pi Pico :: /soil.py @ 10: 16
File Edit View Run Tools Help
[soil.py] x
1 from machine import Pin
2 from time import sleep
3
4 soil = Pin(21, Pin.IN)
5 sleep(2)
6
7 while True:
8     if soil.value() == 0:
9         print("Moisture Detected")
10        sleep(3)

Shell x
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)
    
```

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



21. Infrarood IR-sensor

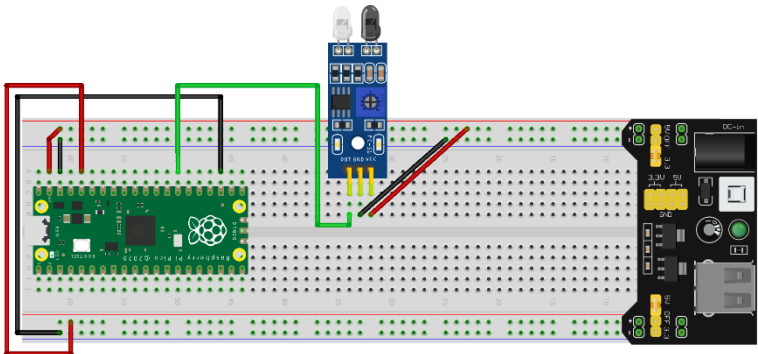
Beschrijving

In deze tutorial leer je hoe je de infrarood IR-sensor aansluit en aanstuurt. Open Thonny Python, ga dan naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla je bestand op onder de naam `ir.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Vereist materiaal

- 1 x Raspberry Pi Pico
- 1 x volledig breadboard
- 1 x micro-USB-kabel
- 1 x Pico-breadbordkit
- 3 x man-naar-man doorverbindingsdraden
- 1 x infrarood IR-sensor

Bedradingsschema

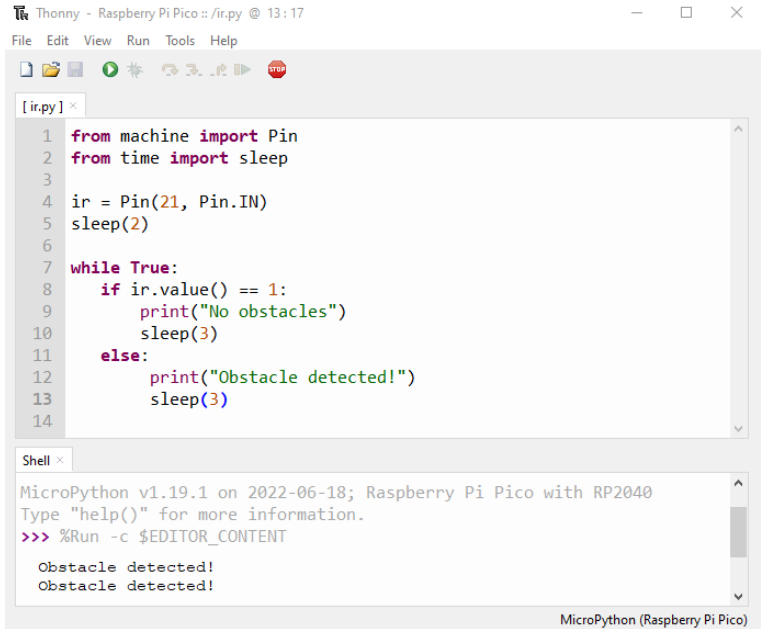


fritzing

- VCC (rode kabel) is verbonden met 5V rail (+)
- GND (zwarte kabel) is verbonden met GND-rail (-)
- OUT (groene kabel) is verbonden met pin GPIO21

Code

MicroPython-code voor de tutorial:



```

Thonny - Raspberry Pi Pico :: /ir.py @ 13:17
File Edit View Run Tools Help

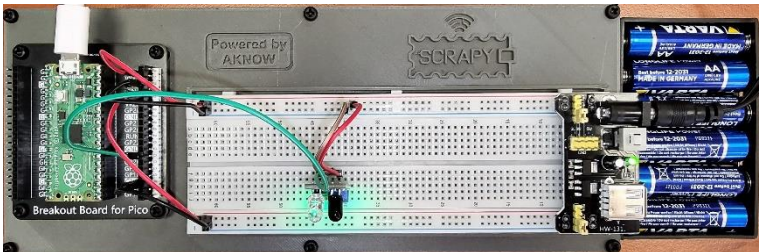
[ir.py] x
1 from machine import Pin
2 from time import sleep
3
4 ir = Pin(21, Pin.IN)
5 sleep(2)
6
7 while True:
8     if ir.value() == 1:
9         print("No obstacles")
10        sleep(3)
11    else:
12        print("Obstacle detected!")
13        sleep(3)
14

Shell x
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)
    
```

Voorbeeldafbeelding

Afbeelding van hoe de tutorial eruit ziet met de meegeleverde hardware:



APPENDIX: Samenvattende tabel voor MicroPython

Digitale uitgang		
De klasse Pin aanroepen	<code>from machine import Pin</code>	
Initialisatie van het digitale uitgangsobject	<code>led = Pin(pin_value, Pin.OUT)</code>	<code>pin_value</code> van 0 tot 40
Open digitale uitgang (3,3V uitgang)	<code>led.value(1)</code>	OP
Sluit digitale uitgang (0V uitgang)	<code>led.value(0)</code>	UIT

Digitale ingang		
De klasse Pin aanroepen	<code>from machine import Pin</code>	
Initialisatie van het digitale uitgangsobject	<code>button = Pin(pin_value, Pin.IN)</code>	<code>pin_value</code> van 0 tot 40
	<code>button = Pin(pin_value), Pin.IN, Pin.PULL_UP)</code>	Activering van de weerstand PULL UP
	<code>button = Pin(pin_value), Pin.IN, Pin.PULL_DOWN)</code>	Activering van de weerstand PULL DOWN
Invoer uitlezing	<code>value = button.value(1)</code>	De retourwaarde kan 0 zijn als de pin op 0V staat, of 1 als de pin op 3,3V staat.

Analoge uitgang (Pulsbreedtemodulatie - PWM)		
De PWM-klasse aanroepen	<code>from machine import PWM</code>	
Initialisatie van de analoge uitgang	<code>led = PWM(Pin(pin_value), frequency)</code>	<code>pin_value</code> van 0 tot 40 <code>frequency</code> in Hz, van 0 tot 78125
Invoer uitlezing	<code>led.duty(duty_cycle)</code>	<code>duty_cycle</code> van 0 tot 1023 (respectievelijk 0V-uitgang tot 3,3V-uitgang)

Analoge ingang		
De ADC-klasse aanroepen	<pre>from machine import ADC</pre>	
Initialisatie van de analoge ingang	<pre>pot = ADC(Pin(pin_value))</pre>	<code>pin_value</code> kan GPIO26, GPIO27 en GPIO28 zijn.
Verklaring bij welke spanning de ingang zijn maximale waarde geeft (bij ESP32 meestal 3,3V)	<pre>pot.atten(ADC.ATTN_11DB)</pre>	ADC.ATTN_0DB: spanning volledig bereik: 1,2 V ADC.ATTN_2_5DB: spanning volledig bereik: 1,5 V ADC.ATTN_6DB: spanning volledig bereik: 2,0 V ADC.ATTN_11DB: spanning volledig bereik: 3,3 V
Verklaring van het bereik van de ingangswaarde (standaard 12bit)	<pre>pot.width(ADC.WIDTH_10BIT)</pre>	ADC.WIDTH_9BIT: bereik 0 tot 511 ADC.WIDTH_10BIT: bereik 0 tot 1023 ADC.WIDTH_11BIT: bereik 0 tot 2047 ADC.WIDTH_12BIT: bereik 0 tot 4095
Invoer uitlezing	<pre>value = pot.readl()</pre>	<code>value</code> is een geheel getal van 0 tot het maximum van het bereik dat is opgegeven in de <code>ADC.WIDTH_#BIT</code> -instructie (zie vorige).

De tijdbibliotheek		
De slaapklasse aanroepen	<pre>from time import sleep</pre>	
Gebruik van de slaapfunctie	<pre>sleep(sec)</pre>	<code>sec</code> is het aantal seconden dat het programma wordt vertraagd
De tijdklasse oproepen	<pre>import time</pre>	
Gebruik van de tijdfunctie	<pre>current_time = time()</pre>	De variabele <code>current_time</code> neemt een numerieke waarde aan, gelijk aan het aantal seconden sinds de laatste reset op het bord.

Als statementstructuur	
<pre>if <expr1>: <statement1> elif <expr2>: <statement2> elif <expr3>: <statement3> (...) else: <statementn></pre>	<p><expr#>: de controlevoorwaarde die Waar of Onwaar moet opleveren</p> <p><statement#>: verzameling commando's die moeten worden uitgevoerd als aan de aangrenzende voorwaarde is voldaan</p> <p><expr#> (bijv. de set <statement2>) wordt uitgevoerd als <expr2> is voldaan)</p> <p><statementn>: verzameling instructies die worden uitgevoerd als aan geen van de <expr#> voorwaarden is voldaan</p>

Structuur while-lus	
<pre>while <expr>: <statement(s)></pre>	<p><expr>: de controlevoorwaarde die Waar of Onwaar moet opleveren</p> <p><statement#>: verzameling commando's die worden uitgevoerd zolang aan de voorwaarde <expr> wordt voldaan</p>

Structuur van de For-lus	
<pre>for <var> in <iterable>: <statement(s)></pre>	<p><iterable>: een verzameling objecten, bijvoorbeeld een lijst met getallen, alfanumerieke tekens, enzovoort.</p> <p><var>: een variabele waaraan de waarde van het volgende item in de verzameling <iterable> wordt toegewezen.</p> <p><statement(en)>: een set instructies die bij elke iteratie wordt uitgevoerd</p>
<pre>for <var> in range(<start>, <end>, <step>): <statement(s)></pre>	<p>range(<start>, <end>, <step>): functie die een reeks getallen teruggeeft van <start> tot <einde>-1, met een <step> verschil tussen twee opeenvolgende getallen (<start> en <step> parameters zijn optioneel).</p>

		<p><var>: variabele waaraan de waarde van het volgende element van de reeks geproduceerd door bereik wordt toegewezen.</p> <p><statement(en)>: een set instructies die in elke iteratie wordt uitgevoerd</p>
Diverse		
DHT11	<pre>import dht</pre>	De DHT-bibliotheek importeren
	<pre>sensor = dht.DHT11(Pin(pin_number))</pre>	Initialisatie van de sensorvariabele met het bijbehorende <code>pin_number</code> .
	<pre>sensor.measure()</pre>	Sensorwaarden bijwerken
	<pre>temp = sensor.temperature()</pre>	Huidige temperatuurwaarde opslaan
	<pre>hum = sensor.humidity()</pre>	Huidige vochtigheidswaarde opslaan
OLED-DISPLAY	<pre>from machine import I2C</pre>	De I2C-bibliotheek importeren
	<pre>import ssd1306</pre>	Importeren van de <code>ssd1306</code> bibliotheek
	<pre>i2c = I2C(-1, scl=Pin(1), sda=Pin(0))</pre>	Initialisatie van de <code>i2c</code> -variabele op de SCL- & SDA-pinnen van de Pico
	<pre>oled_width = 128 oled_height = 64 oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)</pre>	Het scherm initialiseren
	<pre>oled.text('Hello, World 1!', 0, 0) oled.text('Hello, World 2!', 0, 10) oled.text('Hello, World 3!', 0, 20)</pre>	Berichten opslaan in de schermbuffer
	<pre>oled.show()</pre>	Berichten weergeven (nodig om berichten weer te geven die zijn opgeslagen in de schermbuffer)
	<pre>display.pixel(3, 4, 1)</pre>	Zet de pixel op positie (x,y) op het scherm, met <code>x=3</code> & <code>y=4</code> , in status 1 (d.w.z. weergave)



Co-funded by
the European Union



2021-1-FR01-KA220-SCH-000031617

De steun van de Europese Commissie voor de productie van deze publicatie houdt geen goedkeuring in van de inhoud, die uitsluitend de standpunten van de auteurs weergeeft, en de Commissie kan niet verantwoordelijk worden gehouden voor het gebruik van de informatie die erin is vervat.

Raspberry Pi Pico Pinout

