

SmartHome4SENIORS Kit Handleiding



SmartHome
4SENIORS



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 vervat.

WAARSCHUWING

1. Houd bij het gebruik van dit product rekening met het volgende:
2. Dit product bevat veel kleine onderdelen. Het inslikken of verkeerd gebruiken van een van deze onderdelen kan leiden tot ernstige medische complicaties, inclusief overlijden. Zoek onmiddellijk medische hulp als er een ongeluk gebeurt.
3. Het gebruik van dit product en onderdelen ervan in de buurt van stopcontacten of andere circuits is ten strengste verboden vanwege het mogelijke risico op elektrische schokken.
4. Het gebruik van dit product in de buurt van vloeistoffen of vuur is ten strengste verboden.
5. Houd geleidende materialen uit de buurt van dit product.
6. Laat kleine kinderen dit product niet gebruiken zonder toezicht van een volwassene. Bewaar dit product buiten het bereik van kinderen en huisdieren.
7. Bewaar of gebruik dit product niet in extreme omgevingen zoals extreem warm of koud, hoge vochtigheid, onder direct zonlicht, enz.
8. Vergeet niet het circuit te verbreken als het niet nodig is.
9. Sommige onderdelen van dit product kunnen warm aanvoelen bij gebruik in bepaalde circuitontwerpen, wat normaal is.
10. Onjuist gebruik kan oververhitting veroorzaken.
11. Het gebruik van onderdelen die niet in overeenstemming zijn met de specificaties kan schade aan het product veroorzaken.

Inleiding

De SmartHome4SENIORS Kit is gebouwd op basis van de Raspberry Pi Pico (RPi Pico) microcontroller. De kit is gemaakt in het kader van het gelijknamige Erasmus+ medegefinancierde project met projectnummer 2021-1-DE02-KA220-ADU-000033587.

Smart-home automatisering is een bestaande trend die duizenden mensen helpt om processen in hun huishouden te vereenvoudigen. De SmartHome4SENIORS Kit is bedacht en ontwikkeld omdat senioren minder bereid zijn om technologische oplossingen voor slimme huizen te omarmen en te begrijpen.

De kit is bedoeld om een veilig en gezond leven voor senioren te bevorderen door middel van praktijkgericht educatief leren. De Kit biedt gebruikers een geweldige kans om een duik te nemen in de wereld van doe-het-zelf slimme domotica-oplossingen.

De kit bevat alle benodigde hardware (microcontroller, elektronica, sensoren, randapparatuur, etc.) met fysieke computer- en programmeerconcepten.

Het doel van deze handleiding is

- U te informeren over de onderdelen van de Kit en het gebruik van de belangrijkste elektronische elementen.
- U stap voor stap begeleiden bij het effectief in elkaar zetten van de kit, rekening houdend met de bijbehorende voorzorgsmaatregelen.
- Instructies geven voor het gebruik van de componenten en hun verbinding met de RPi Pico microcontroller.
- Instructies geven over de functionaliteit en reikwijdte van elk elektronisch onderdeel.

Veel leesplezier en plezier door praktische oefeningen en experimenten met de SmartHome4SENIORS Kit

Inhoudsopgave

Inleiding.....	1
Inhoudsopgave	2
Inbegrepen in de SmartHome4SENIORS-kit	4
Componenten Uitleg	6
1. Wat is een breadboard?.....	6
2. Wat is een weerstand?	7
3. Wat is een condensator?	10
4. Wat is een diode?	10
5. Wat is een startkabel?	11
Projectvoorbereiding	13
Montage SmartHome-kit.....	21
SmartHome-kit elektronicamontage	28
Basishandleidingen.....	34
0. “Hello SmartHome people!”	34
1. Een LED aansturen.....	36
2. Drukknop.....	38
3. Zoemer.....	40
4. Potentiometer	42
Handleidingen voor gevorderden.....	44
5. LED-verkeerslichtmodule.....	45
6. LDR-fotoresistor	47
7. DC Motor (kleine ventilator)	49
8. SG-90 servomotor.....	51
9. OLED I2C SSD1306 scherm	53

10. RFID-lezer RC522	57
Tutorials met sensoren	60
11. Regendruppelsensor.....	60
12. HC-SR04 Ultrasonische sensor	62
13. PIR-bewegingssensor	65
14. DHT11-sensor.....	67
15. Flame Sensor.....	69
16. MQ-135 gasdetectiesensor.....	71
APPENDIX: Samenvattende tabel voor MicroPython	73

Inbegrepen in de SmartHome4SENIORS-kit

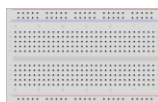
1 x Raspberry Pi Pico
Microcontroller



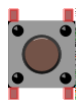
1 x MB-102
Voedingsmodule



1 x Witte Broodplank 830
stukks



1 x Drukknop &
1 x Knopdop



1 x Zoemer



5 x Weerstanden 220
Ohm &
2 x 1k Ohm



1 x LDR Fotoresistor



1 x 100uF Condensator



4 x LED 3mm
(blauw, groen, rood,
geel)



1 x RGB LED



1 x Stoplicht LED-
module



1 x Ventilator (DC-motor)



1 x TIP-120 Regelmodule



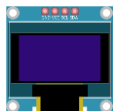
1 x Diode 1N4007



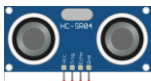
1 x SG90 Servomotor



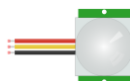
1 x OLED SSD1306 I2C
Weergave



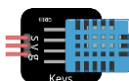
1 x HC-SR04 Ultrasonische
sensor



1 x PIR
bewegingsdetector
sensor HC-SR501



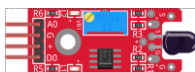
1 x DHT11 digitale
temperatuur- en
vochtigheidssensor



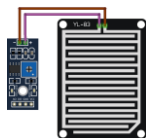
1 x MQ-135 Sensor
luchtkwaliteit



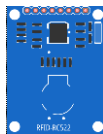
1 x Sensor voor
vlamdetectie



1 x Regendruppelsensor



1 x RFID-lezer RC522



1 x RFID Tag 3.56MHz



1 x Roterende
potentiometer Linear
B1k Ohm



1 x USB naar micro-USB-
kabel



6 x AA 1.5V Batterijen &
1 x batterijhouder



9 x Jumper Kabels
Pakken



19 x M2 Bouten & 19 x
M2 Moeren

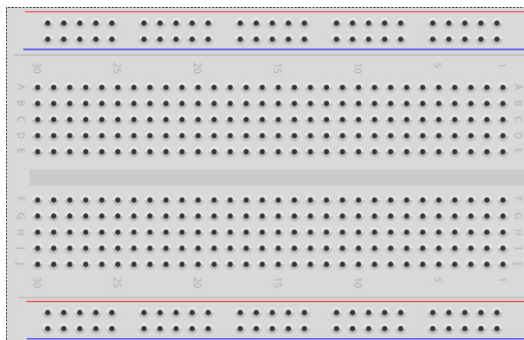


4 x M2 Metalen
Houtschroeven



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 schakeling.

De lange stroken aan de 2 zijkanten 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 stroom 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, heeft een minteken (-) en zorgt voor de aarde.

Voordelen van het gebruik van een Breadboard:

- Vereenvoudigt het snel controleren van eenvoudige en complexe schakelingen en het gemakkelijk controleren van schakelingen in hun beginstadium.
- 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>

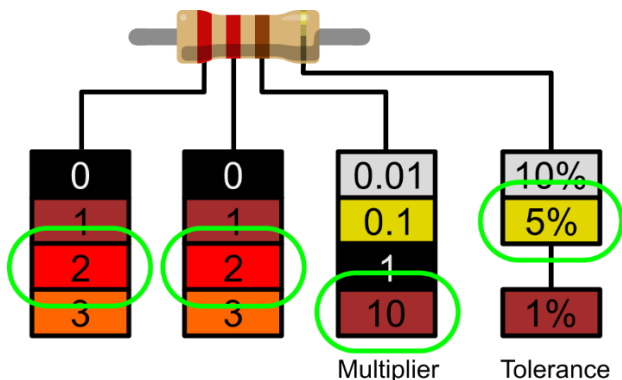
Algemene kleurcodes voor weerstanden en hun gebruik:

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 en kunnen dus in elke richting in een schakeling worden gebruikt. Maar om de juiste kleurcodewaarden van een weerstand 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)

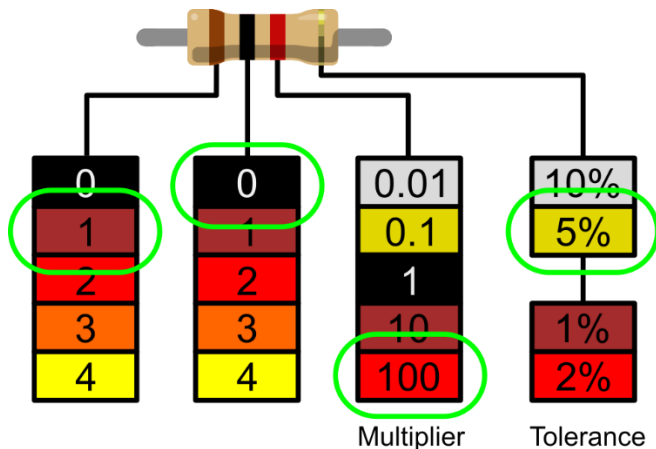


(Afbeelding credit: Future) beschikbaar op <https://www.tomshardware.com/how-to/resistor-color-codes>

1. Het eerste significante cijfer is rood en met behulp van de decoder kunnen we zien dat rood de waarde 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 vijfbandweerstand met een derde significant cijfer. Het extra cijfer zorgt voor duidelijkheid die essentieel kan zijn in schakelingen die gevoelig zijn voor weerstand, bijvoorbeeld 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 1e regel is bruin en met behulp van de decoder kunnen we zien dat de waarde 1 is.
2. De 2e regel 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. 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 diëlektricum wordt genoemd. (Wikipedia)

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

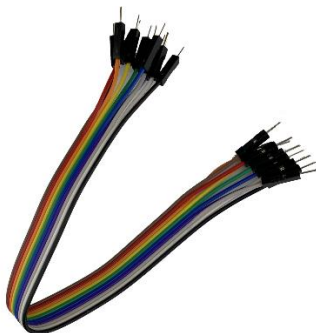
4. Wat is een diode?



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

De meest gebruikelijke 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 worden gebruikt als voordeel 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 tussen elke versie zit in het eindpunt 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>)

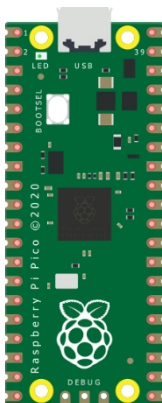
Projectvoorbereiding

1. Leesmij Voor gebruik

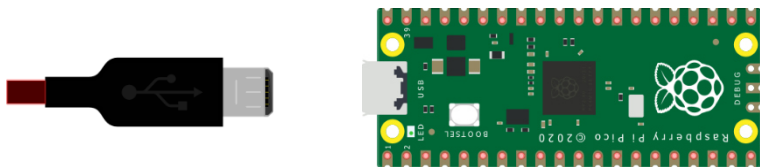
NOTE: Aangezien het hier om experimenten met circuits gaat, kan een verkeerde aansluiting of kortsluiting uw RPi Pico-ontwikkelford beschadigen. Controleer het circuit altijd opnieuw voordat u de voeding aansluit.

2. Raspberry Pi Pico microcontroller

Dit is de Raspberry Pi Pico:

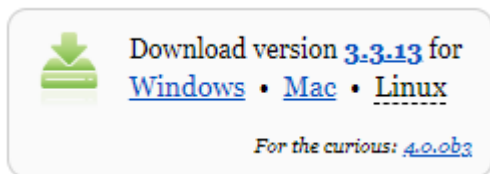


Sluit de micro-USB-kabel aan op de poort aan de linkerkant van het bord.



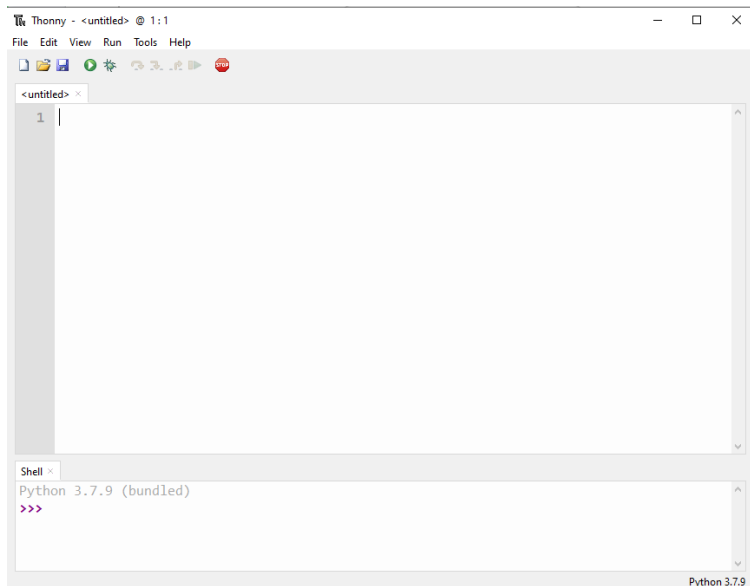
3. Installeer Thonny IDE

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



In deze handleiding worden alle tutorials geprogrammeerd in Windows 10, met behulp van een RPi Pico microcontroller en de juiste firmware.

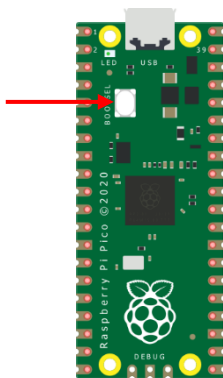
Nadat de installatie is voltooid, open je Thonny vanaf je computer.



4. Installatie firmware

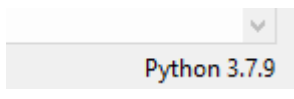
De RPi Pico kan worden geprogrammeerd met een Python-variant, MicroPython genaamd. Om MicroPython op de RPi Pico te gebruiken, moet je eerst de firmware installeren.

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

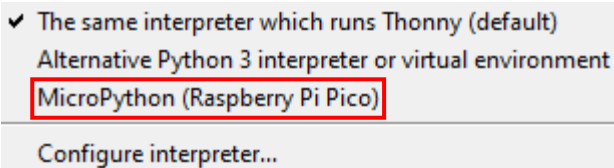


Stap 2: Druk op de BOOTSEL knop en houd deze ingedrukt terwijl je het andere uiteinde van de micro-USB kabel aansluit op je computer.

Stap 3: In de rechter benedenhoek van Thonny zie je de versie van Python die je momenteel gebruikt.

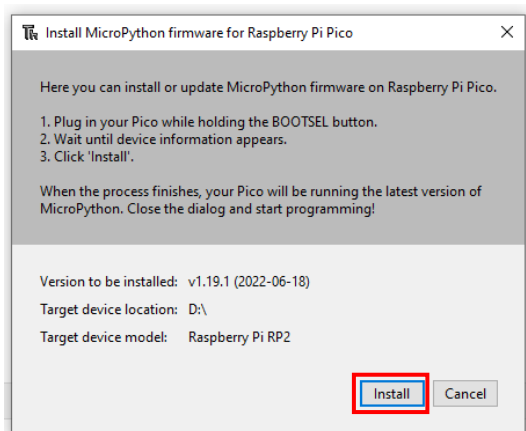


Klik op de Python-versie en kies MicroPython (Raspberry Pi Pico).

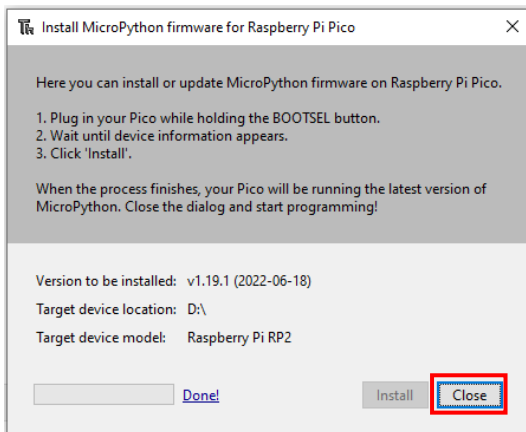


Als u deze optie niet ziet, controleer dan of de kabel goed is aangesloten op de Pico en/of uw computer.

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



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



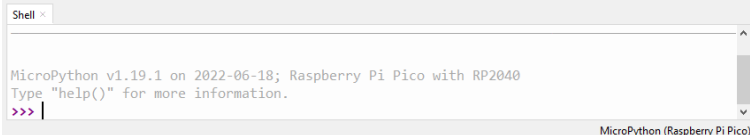
Je hoeft dit proces niet elke keer te herhalen als je de Raspberry Pi Pico aansluit op je computer, dus sluit hem de volgende keer gewoon aan en je kunt aan de slag.

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.
>>> |
```

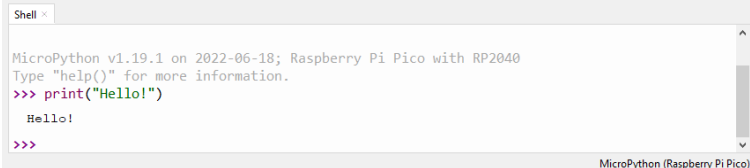
MicroPython (Raspberry Pi Pico)

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.

Type het volgende commando:

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

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("Hello!")
Hello!
>>>
```

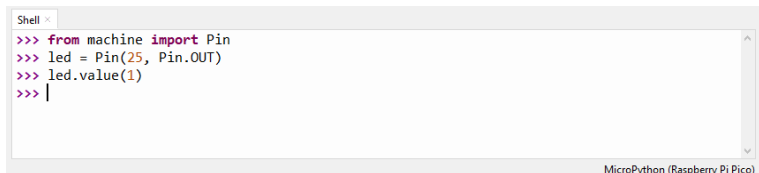
MicroPython (Raspberry Pi Pico)

Met MicroPython kun je hardware-specifieke 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)
```



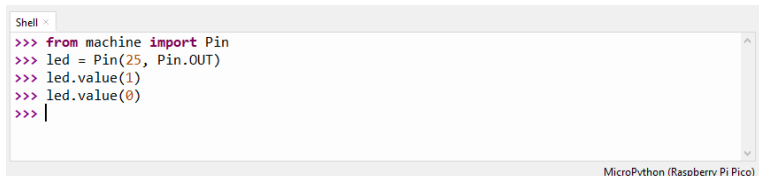
```
Shell
>>> from machine import Pin
>>> led = Pin(25, Pin.OUT)
>>> led.value(1)
>>> |
```

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)
>>> |
```

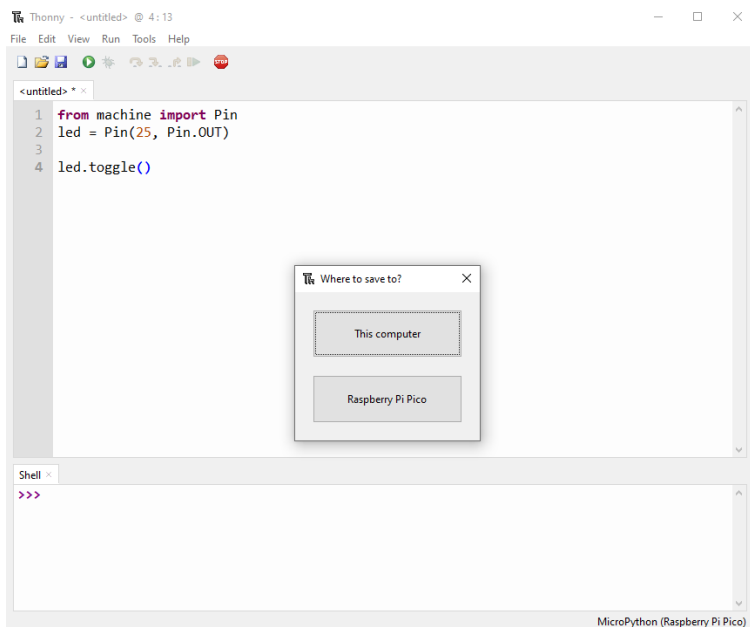
In de rest van dit onderdeel ga je je eerste "echte" programma schrijven dat de LED op het bord laat knipperen telkens als je je programma uitvoert.

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 kunt u programma's rechtstreeks 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 pictogram Afspelen klikt, zou je moeten zien dat de LED aan boord 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, klik je op het STOP-pictogram of druk je op Ctrl+C op je toetsenbord.

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

Montage SmartHome-kit

Voor de montage zijn de volgende onderdelen nodig:

- 6 x Multiplex stukken
- 10 x M3 Metalen bouten
- 10 x M3 Metalen moeren



De procedure is eenvoudig en kan in 7 stappen worden voltooid. Alle multiplex stukken zijn gemarkeerd op de rechter zijhoek met de volgende notatie:

BP: Basisstuk

B: Achterstuk

F: Voorkant stuk

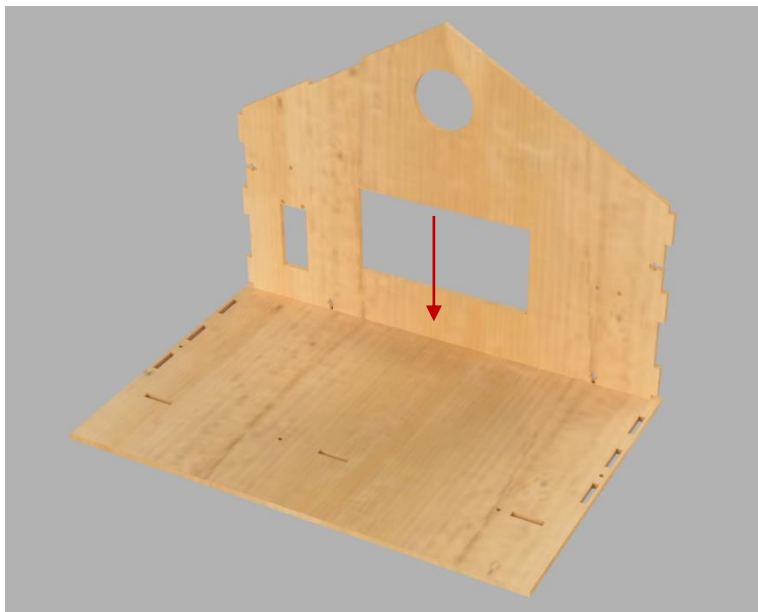
L: Linker zijstuk

R: Rechter zijstuk

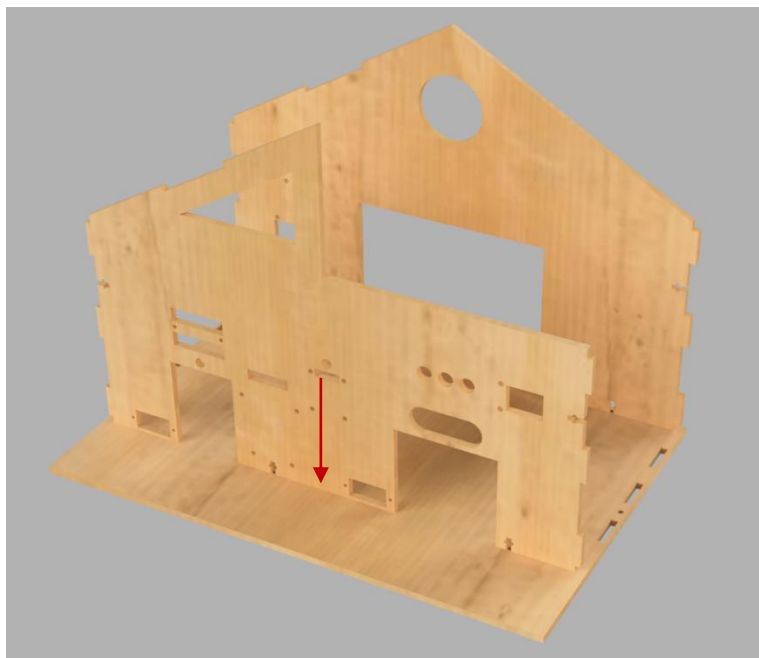
Stap 1: Plaats het multiplex basisstuk op een horizontaal oppervlak.



Stap 2: Plaats het stuk aan de achterkant zoals op de afbeelding.



Stap 3: Plaats het stuk aan de voorkant zoals aangegeven in de afbeelding.



Stap 4: Plaats het multiplex stuk aan de linkerkant zoals aangegeven in de afbeelding.

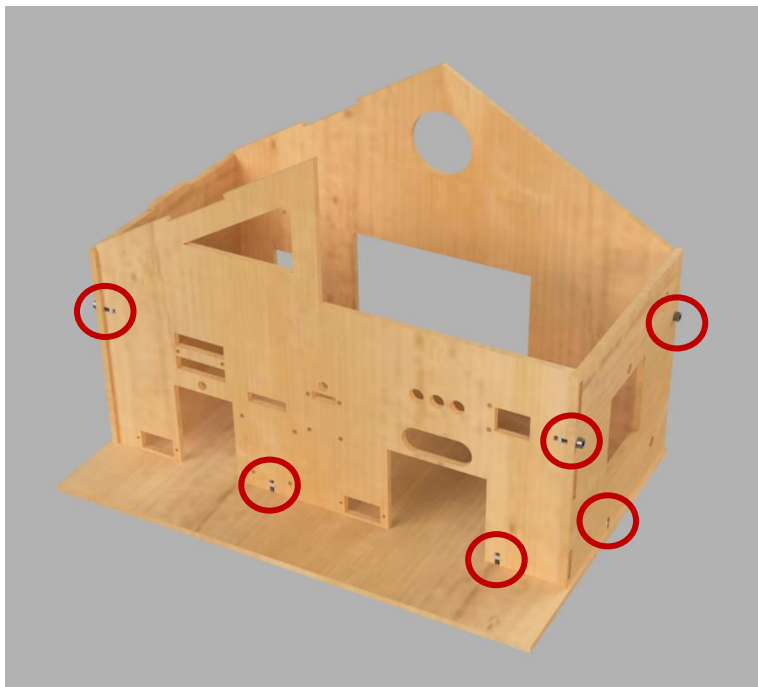


Stap 5: Plaats het multiplex stuk aan de rechterkant zoals aangegeven in de afbeelding.

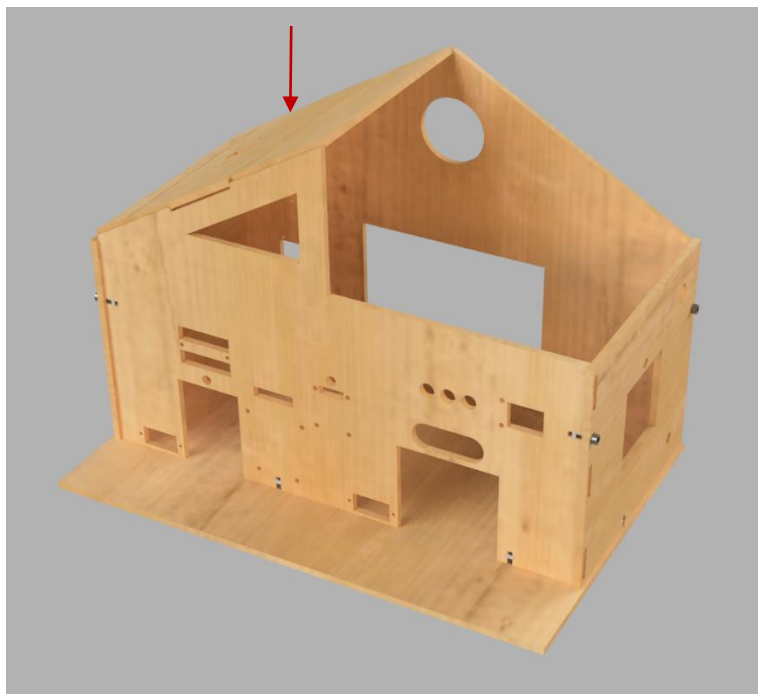


Stap 6: Zet de multiplexdelen aan elkaar vast met 10 bouten en 10 moeren.

- 6 M3 bouten en moeren worden onder het basisstuk geplaatst en zijn verantwoordelijk voor het bij elkaar houden van de muren.
- 4 M3 bouten en moeren worden aan de linker- en rechtermuur geplaatst en zorgen ervoor dat de constructie stabiel blijft.

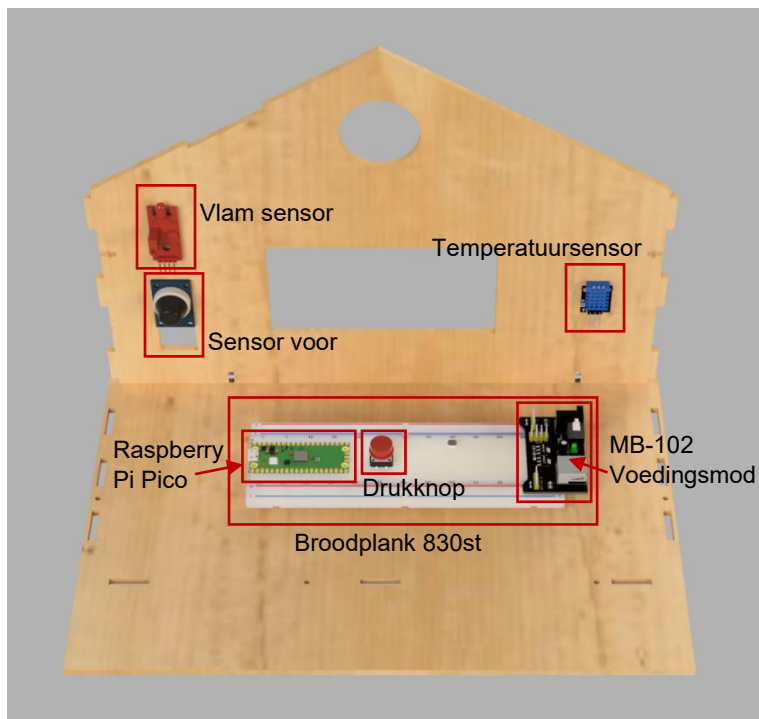


Stap 7: Plaats het multiplex dakstuk bovenop het huismodel zoals op de afbeelding:



SmartHome-kit elektronikamontage

Er zijn verschillende elektronicaonderdelen die op het SmartHome huismodel moeten worden gemonteerd. De volgende afbeeldingen laten in detail zien welke elektronica je moet monteren en hoe je ze op de verschillende sleuven monteert. Je hebt ook een kruiskopschroevendraaier en een tang nodig.



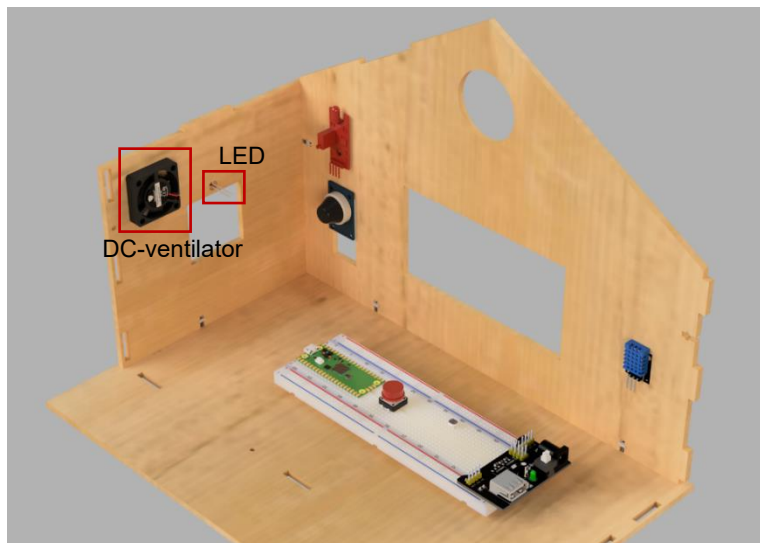
Plaats de Raspberry Pi Pico op het breadboard, samen met de voedingsmodule (MB-102) en de drukknoop. Het breadboard heeft een kleeflaag aan de onderkant, dus zorg ervoor dat je de

bescherm laag eraf trekt en plaats het dan op het basisstuk van het huis.

Er is wat elektronica die op de achterwand gemonteerd moet worden. Dit zijn de vlamsensor, de luchtkwaliteitssensor en de DHT11 temperatuur- en vochtigheidssensor. Je hebt de volgende onderdelen nodig:

- **Vlamvoeler:** 1 x bout en 1 x moer. Monteer deze door het middelste gedeelte van de sensor en zorg ervoor dat de sensor (zwarte gedeelte) omhoog kijkt.
- **Luchtkwaliteitssensor:** 2 x bout en 2 x moer. Monteer hem door de montagegaten linksboven en rechtsboven.
- **DHT11:** 2 x bout en 2 x moer. Je moet de sensor (blauwe gedeelte) naar voren buigen om bij de montagegaten te komen. Monteer de sensor vervolgens door de montagegaten linksboven en rechtsboven.

Aan de linkerkant moet je een **DC-ventilator** en een **LED-lampje** monteren. Voor de **DC-ventilator** gebruik je de bouten en moeren uit de verpakking (4 x bouten en 4 x moeren). Steek de **LED** in het montagegat en de wrijving houdt hem op zijn plaats.



Op het rechter stuk moet je een 5V zoemer, een potentiometer en een LED monteren.

Plaats de **zoemer** in de gleuf rechtsboven (zorg ervoor dat de pinnen aan de binnenkant van het huismodel zitten) en wrijving houdt hem op zijn plaats.

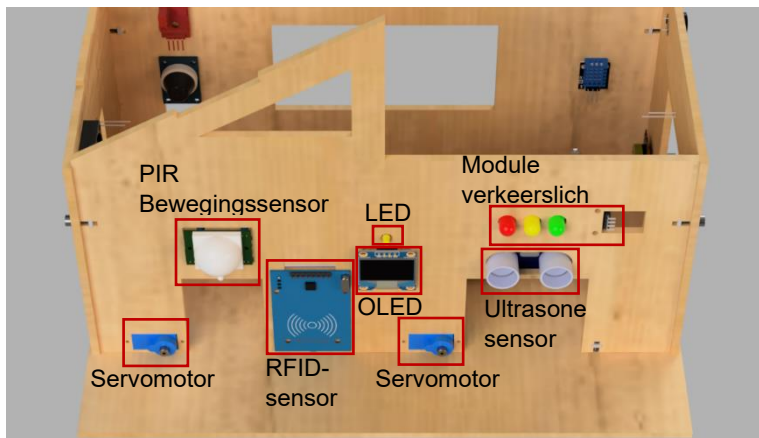
De **potentiometer** heeft al een moerring die je moet losdraaien. Plaats de potentiometer van de binnenkant van het huis naar de buitenkant, zodat de hendel eruitziet zoals op de afbeelding hieronder. Schroef vervolgens de moerring terug totdat deze vast genoeg zit om hem op zijn plaats te houden wanneer je aan de hendel draait.

Tot slot moet er een **LED** worden geïnstalleerd. Steek de LED in het montagegat en de wrijving houdt hem op zijn plaats.

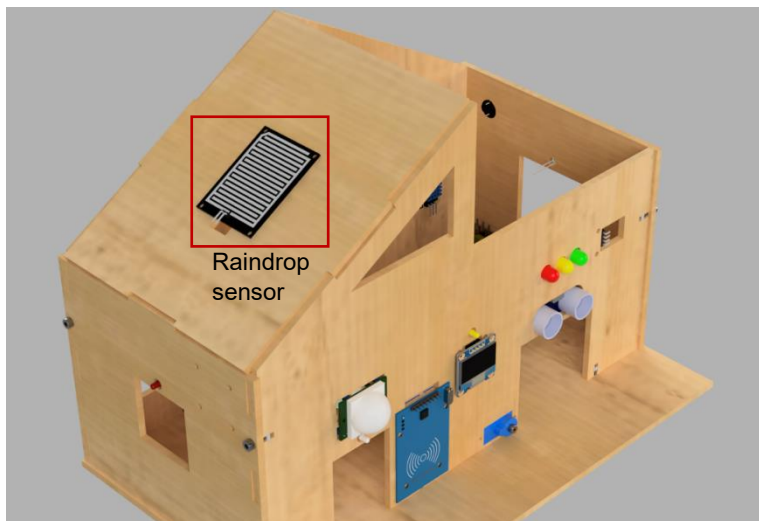


Er zijn verschillende sensoren en elektronica die aan de voorkant van het huismodel moeten worden gemonteerd.

- **Servomotoren:** 4 x schroeven. Plaats ze van de binnenkant van het huis naar de buitenkant en monteer ze met de bijgeleverde schroeven.
- **PIR Bewegingssensor:** 2 x bout en 2 x moer. Monteer deze door de montagegaten linksboven en rechtsonder.
- **RFID-sensor:** 2 x bout en 2 x moer. Monteer deze door de montagegaten linksboven en rechtsonder.
- **OLED-display:** 2 x bout en 2 x moer. Monteer het door de montagegaten linksboven en rechtsonder.
- **Module verkeerslichten:** 2 x bout en 2 x moer. Steek de module van binnen naar buiten. Monteer hem vervolgens door de twee montagegaten aan de rechterkant.
- **Ultrasonische sensor:** Steek hem van de binnenkant van het huis naar de buitenkant. Wrijving houdt hem op zijn plaats.
- **LED:** Steek de LED in het montagegat en wrijving houdt hem op zijn plaats.



Tot slot moet de **regendruppelsensor** op het dak worden geïnstalleerd. Gebruik 2 x bouten en 2 x moeren en monteer hem door de montagegaten linksboven en rechtsonder.



Nadat alle elektronica en sensoren op hun plaats zijn gemonteerd, moet je ze bedraden met behulp van de jumper

kabels die in het pakket zitten. Deze procedure wordt echter uitgelegd in de tutorials-sectie, waarin de verschillende GPIO-pinnen van de Raspberry Pi Pico en de pinnen van elke elektronische component of sensor worden uitgelegd.

In de verpakking vind je ook 6 AA-batterijen en een batterijhouder. Plaats de batterijen op de batterijhouder en houd deze voorlopig apart. In een latere tutorial leer je waar je het moet plaatsen en hoe het moet worden verbonden met de MB-102 voedingsmodule.

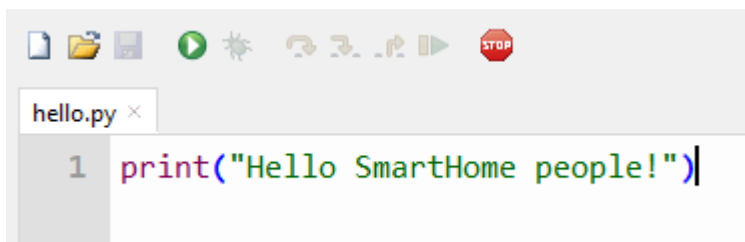
Basishandleidingen

0. “Hello SmartHome people!”

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

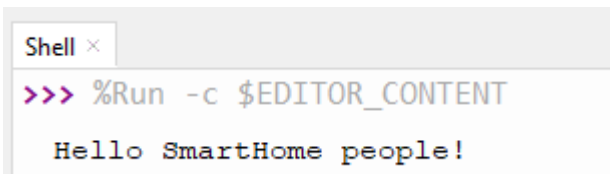
De functie print()

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.



```
hello.py x
1 print("Hello SmartHome people!")
```

2. Controleer het venster van de schelp.



```
Shell x
>>> %Run -c $EDITOR_CONTENT
Hello SmartHome 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.

```

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!
  
```

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

```

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 afdrukfunctie om in 3 afzonderlijke regels "Vaarwel SmartHome-mensen!" af te drukken met slechts één afdrupinstructie.

1. Een LED aansturen

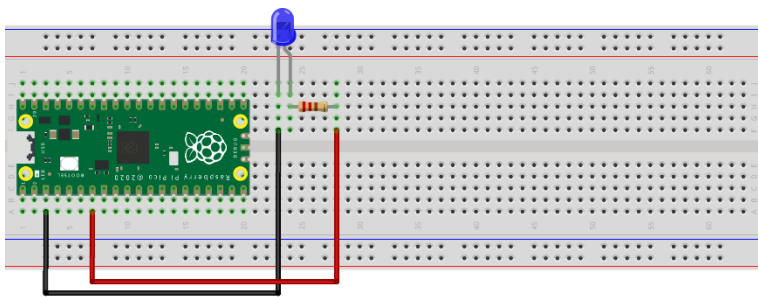
Beschrijving

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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 1 x Full size breadplank
- 1 x Micro-USB kabel
- 2 x Man-tot-man jumperdraden
- 1 x LED (elke kleur)
- 1 x weerstand van 220 Ohm

Bedradingschema



fritzing

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

Code

MicroPython-code voor de tutorial:



```

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

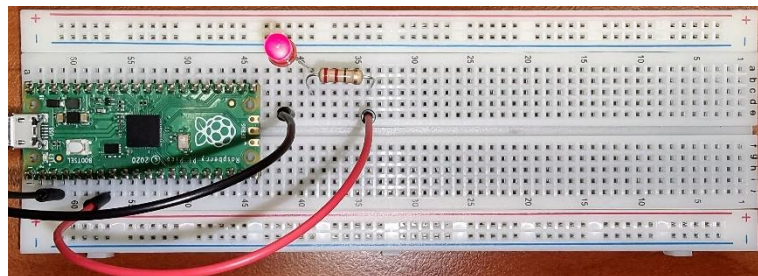
[ led.py ] x
1 from machine import Pin #import Pin from library machine
2 from time import sleep #import sleep from library time
3
4 led = Pin(5, Pin.OUT)
5
6 while True:
7     led.toggle()
8     sleep(1)
9 |

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 de tutorial eruit ziet met de meegeleverde hardware:



2. Drukknop

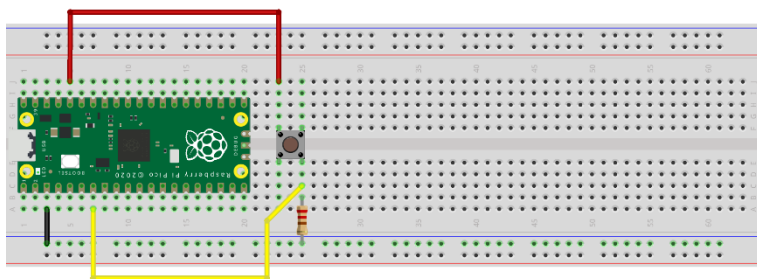
Beschrijving

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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 3 x mannetje naar mannetje jumperdraden
- 1 x Full size broodplank
- 1 x 220 Ohm weerstand
- 1 x Micro-USB kabel
- 1 x knoopkapje
- 1 x Drukknop (elke kleur)

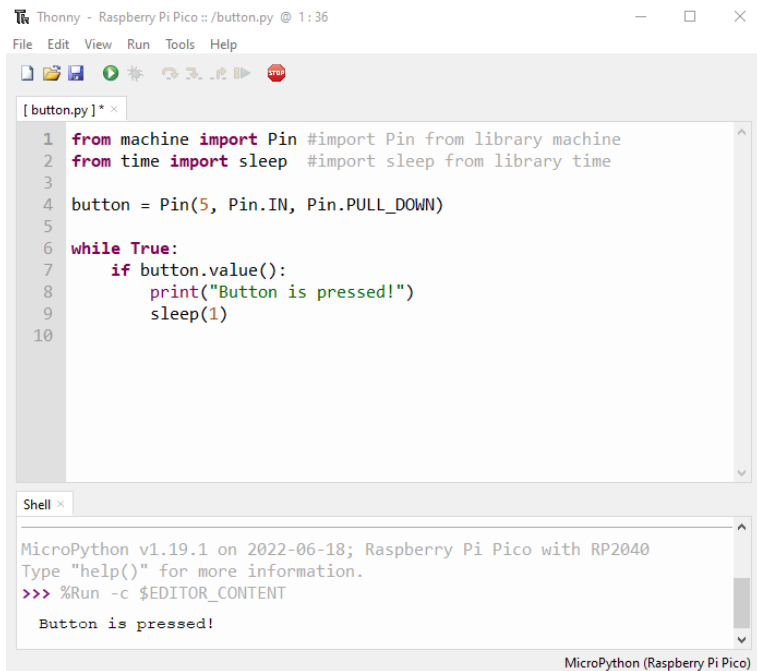
Bedradingschema



- sluit de linkerbovenknop aan op de 3v3-pin (rode kabel)
- sluit de rechteronderkant van de knop aan op GPIO5 (gele kabel)
- verbind een GND-pin met de (-) rail (zwarte kabel)
- sluit de weerstand van 220 Ohm aan op de (-) rail en de rechter onderkant van de knop

Code

MicroPython-code voor de tutorial:

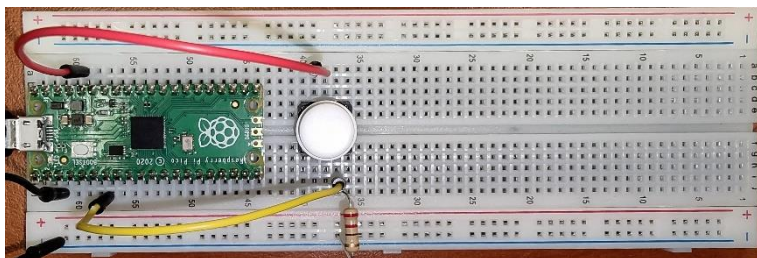


```

Thonny - Raspberry Pi Pico :: /button.py @ 1:36
File Edit View Run Tools Help
[button.py]* x
1 from machine import Pin #import Pin 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.
>>> %Run -c $EDITOR_CONTENT
Button is pressed!
MicroPython (Raspberry Pi Pico)
  
```

Voorbeeldafbeelding

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



3. Zoemer

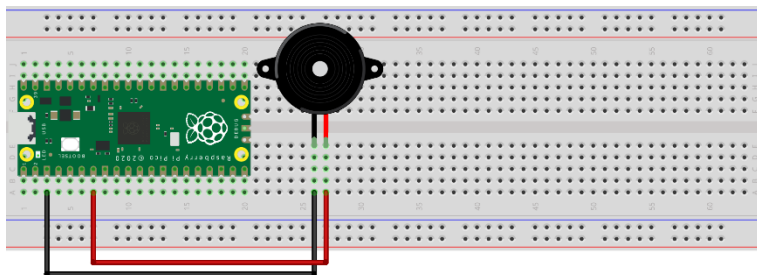
Beschrijving

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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 2 x Man-naar-man jumperdraden
- 1 x Full size broodplank
- 1 x Zoemer
- 1 x Micro-USB kabel

Bedradingsschema

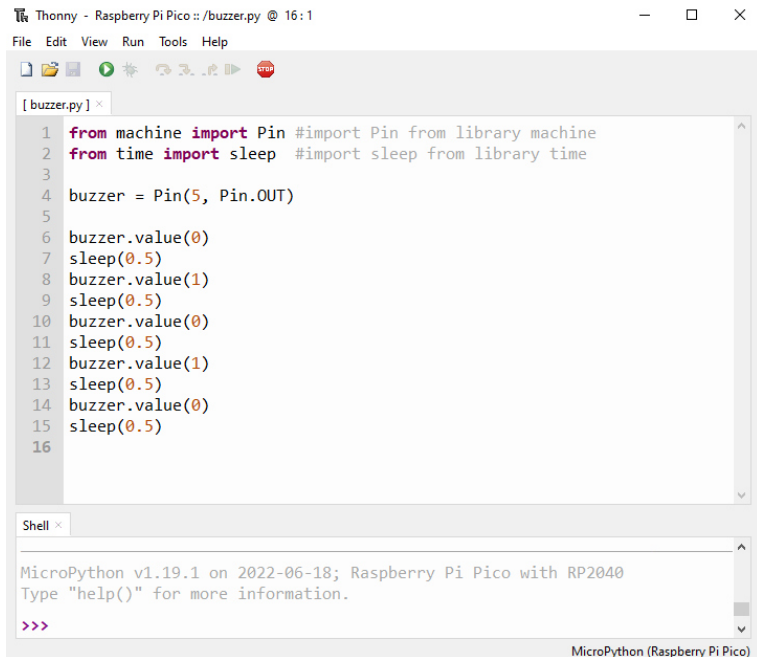


fritzing

- verbind het lange uiteinde (+) van de zoemer met pin GPIO5
- verbind het korte uiteinde (-) van de zoemer met 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
```

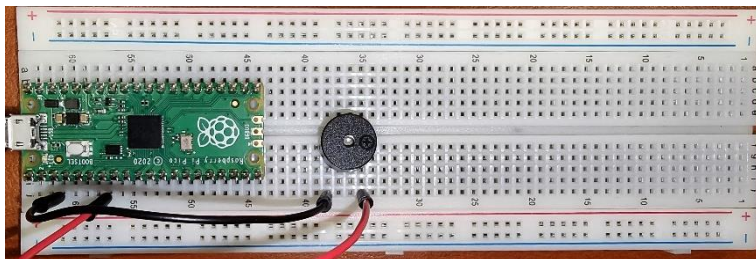
Shell x

```
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 de tutorial eruit ziet met de meegeleverde hardware:



4. Potentiometer

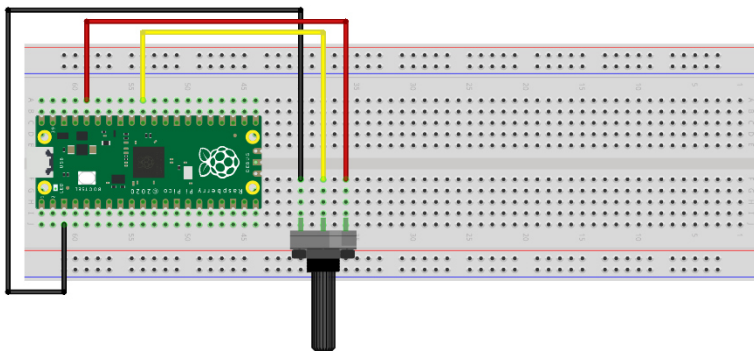
Beschrijving

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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 3 x Man-naar-man jumperdraden
- 1 x Full size broodplank
- 1 x potentiometer
- 1 x Micro-USB kabel

Bedradingsschema

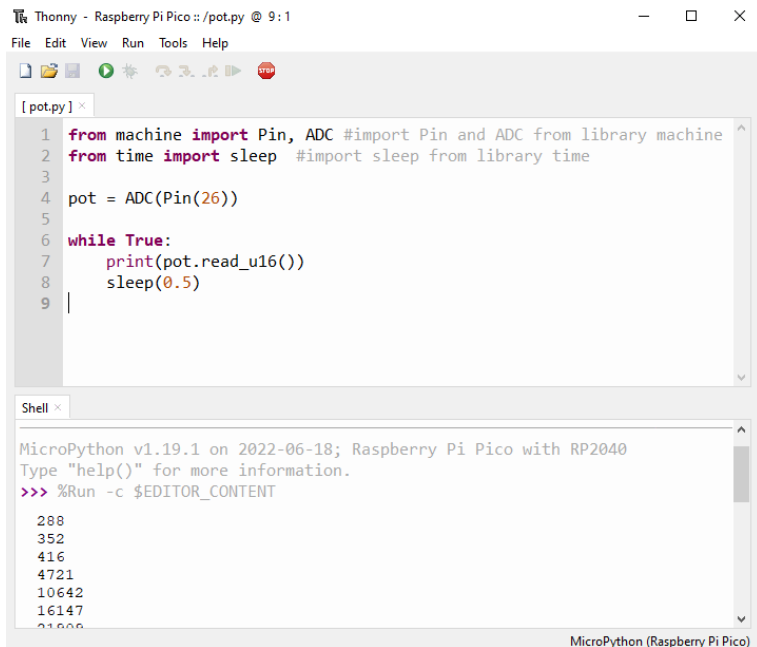


fritzing

- zwarte kabel moet worden verbonden met pin GND (pin 3)
- gele kabel moet worden verbonden met GPIO26 ADC-pin
- rode kabel moet worden verbonden met 3V3 voedingspin
- 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

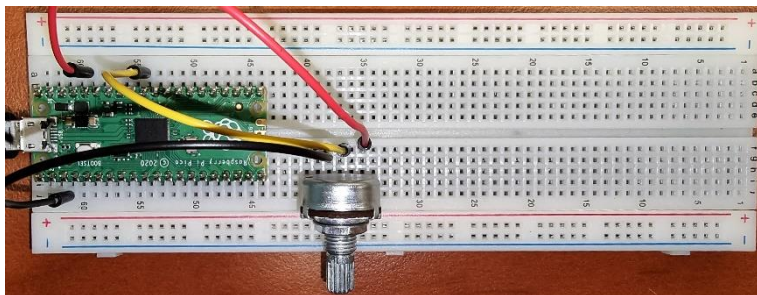
[ pot.py ] x
1 from machine import Pin, ADC #import Pin and ADC from library machine
2 from time import sleep #import sleep from library time
3
4 pot = ADC(Pin(26))
5
6 while True:
7     print(pot.read_u16())
8     sleep(0.5)
9

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
288
352
416
4721
10642
16147
21000

MicroPython (Raspberry Pi Pico)
  
```

Voorbeeldafbeelding

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



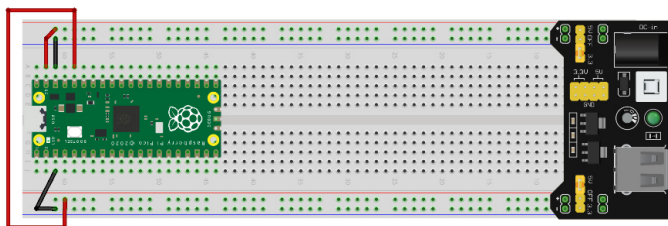
Handleidingen voor gevorderden

Voordat we verder gaan met dit onderdeel en het volgende, moeten we een paar wijzigingen aanbrengen aan ons ontwikkelbord. 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

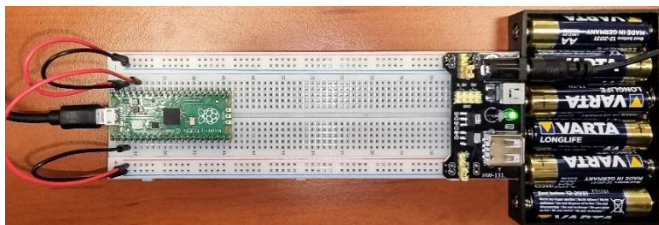
Volg het schema om de nieuwe componenten aan te sluiten:



fritzing

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

Voorbeeldafbeelding



5. LED-verkeerslichtmodule

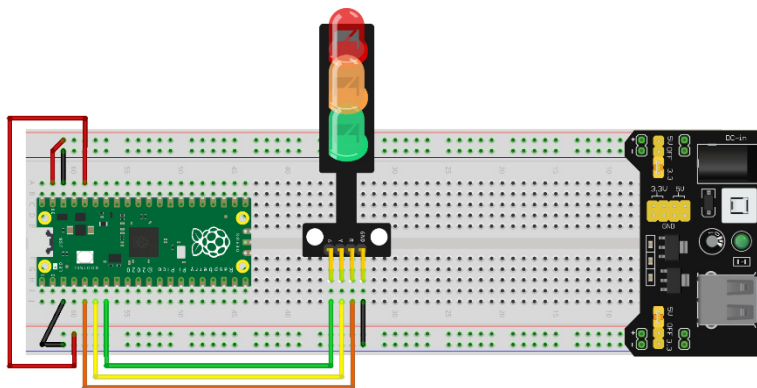
Beschrijving

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

Benodigde materialen

- 1 x Raspberry Pi Pico
- 1 x Full size broodplank
- 1 x Micro-USB kabel
- 4 x Male-to-male jumperdraden
- 1 x LED-verkeerslichtmodule

Bedradingsschema

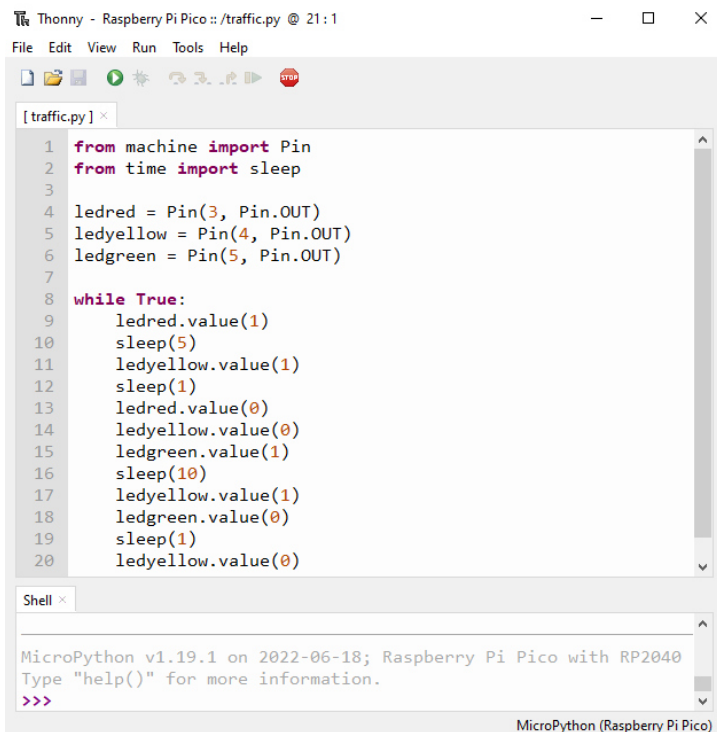


fritzing

- oranje kabel (R) is verbonden met pin GPIO3
- gele kabel (Y) is verbonden met GPIO4-pin
- groene kabel (G) is verbonden met GPIO5-pin
- zwarte kabel (GND) is verbonden met GND-rail ((-) zwart)

Code

MicroPython-code voor de tutorial:



```

Thonny - Raspberry Pi Pico :: /traffic.py @ 21: 1
File Edit View Run Tools Help

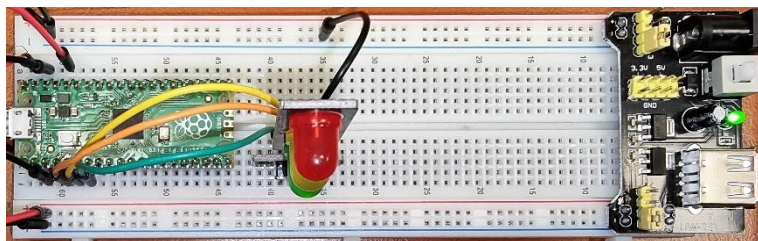
[ traffic.py ] x
1 from machine import Pin
2 from time import sleep
3
4 ledred = Pin(3, Pin.OUT)
5 ledyellow = Pin(4, Pin.OUT)
6 ledgreen = Pin(5, Pin.OUT)
7
8 while True:
9     ledred.value(1)
10    sleep(5)
11    ledyellow.value(1)
12    sleep(1)
13    ledred.value(0)
14    ledyellow.value(0)
15    ledgreen.value(1)
16    sleep(10)
17    ledyellow.value(1)
18    ledgreen.value(0)
19    sleep(1)
20    ledyellow.value(0)

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:



6. LDR-fotoresistor

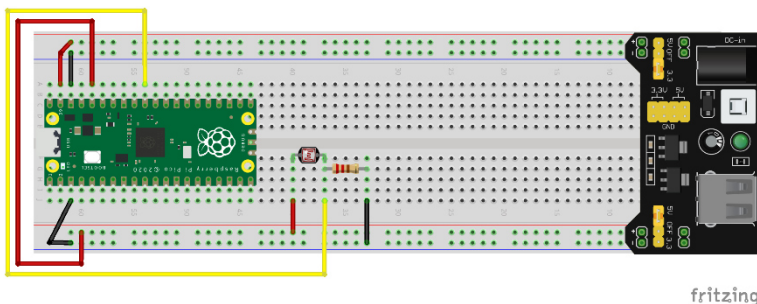
Beschrijving

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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 3 x Mannetje-naar-man jumperdraden
- 1 x Full size broodplank
- 1 x LDR fotoresistor
- 1 x Micro-USB kabel
- 1 x 220Ohm weerstand

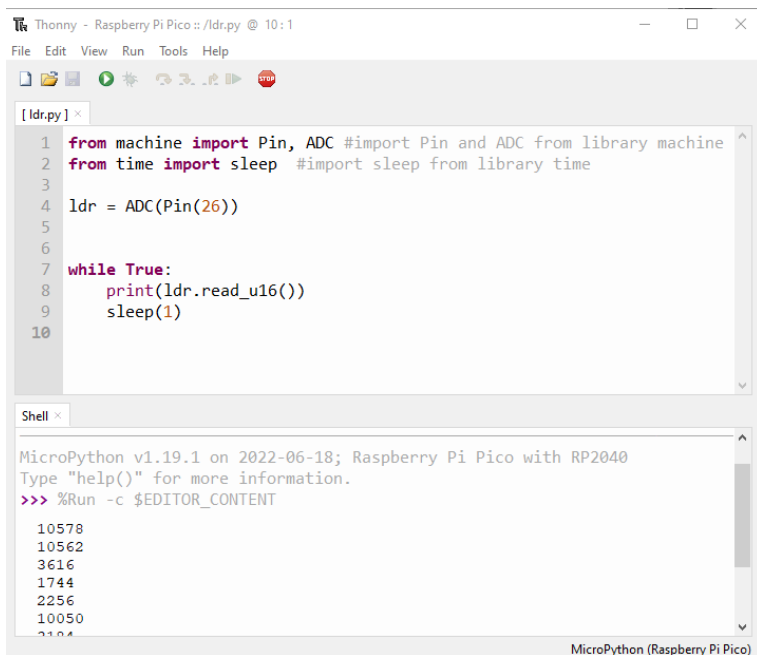
Bedradingsschema



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

Code

MicroPython-code voor de tutorial:



```

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

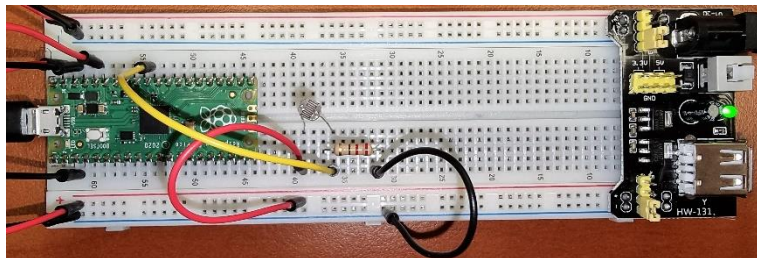
[ ldr.py ] x
1 from machine import Pin, ADC #import Pin and ADC from library machine
2 from time import sleep #import sleep from library time
3
4 ldr = ADC(Pin(26))
5
6
7 while True:
8     print(ldr.read_u16())
9     sleep(1)
10

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
10578
10562
3616
1744
2256
10050
2104

MicroPython (Raspberry Pi Pico)
  
```

Voorbeeldafbeelding

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



7. DC Motor (kleine ventilator)

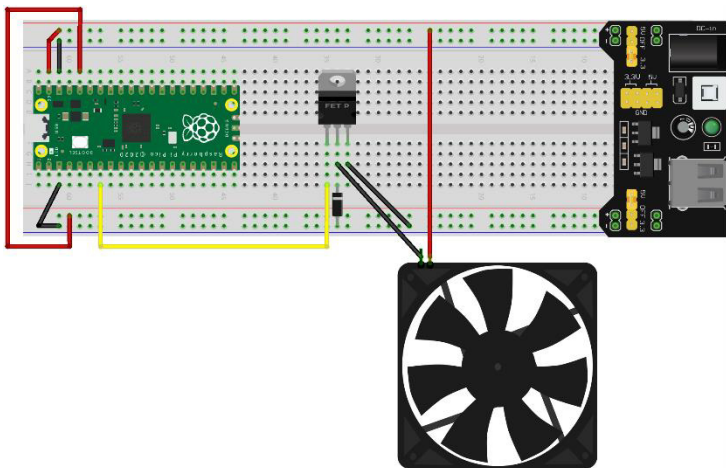
Beschrijving

In deze tutorial leer je hoe je een kleine DC (gelijkstroom) motor 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 `fan.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 1 x Full size broodplank
- 1 x Micro-USB kabel
- 1 x Diode
- 3 x mannetje naar mannetje jumperdraden
- 1 x DC-motor (kleine ventilator)
- 1 x TIP-120 transistor

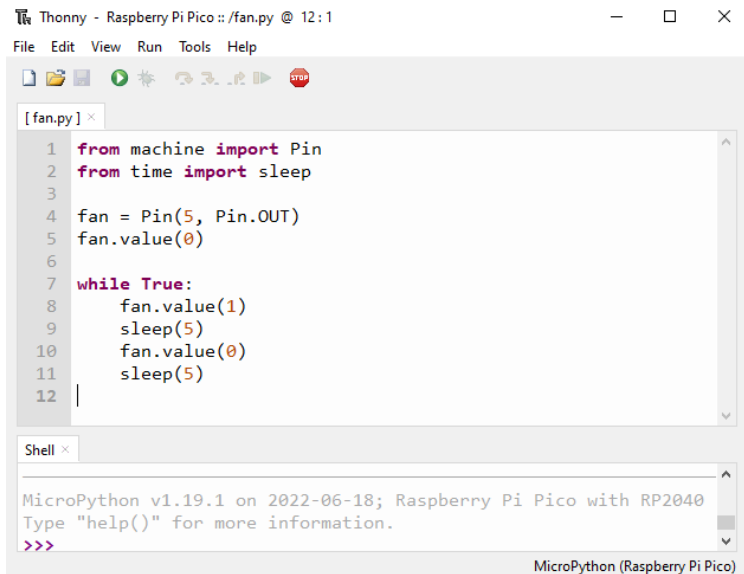
Bedradingsschema



fritzing

Code

MicroPython-code voor de tutorial:

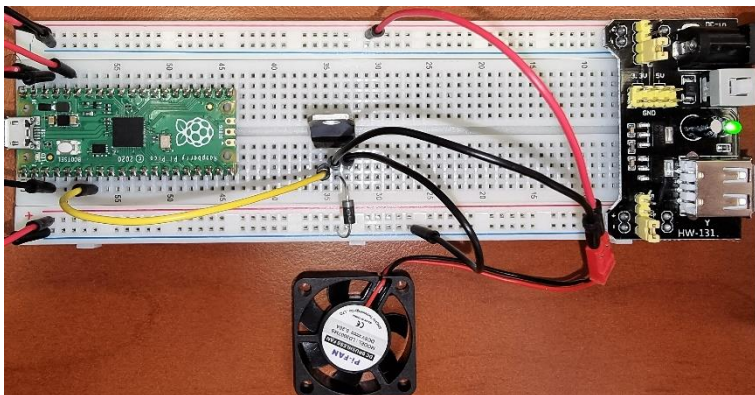


```

Thonny - Raspberry Pi Pico :: /fan.py @ 12: 1
File Edit View Run Tools Help
[ fan.py ] x
1 from machine import Pin
2 from time import sleep
3
4 fan = Pin(5, Pin.OUT)
5 fan.value(0)
6
7 while True:
8     fan.value(1)
9     sleep(5)
10    fan.value(0)
11    sleep(5)
12
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:



8. SG-90 servomotor

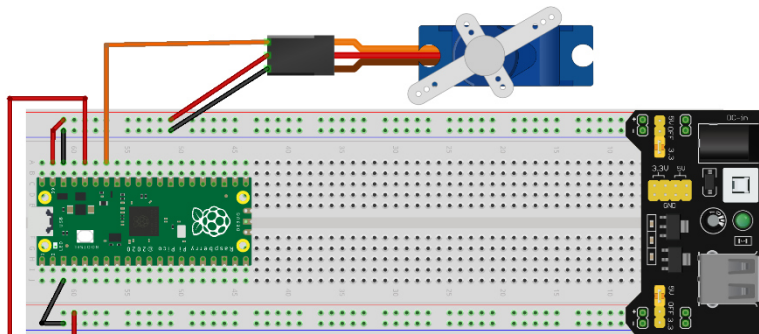
Beschrijving

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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 1 x Full size broodplank
- 1 x Micro-USB kabel
- 3 x mannetje naar mannetje jumperdraden
- 1 x SG-90 servomotor

Bedradingsschema

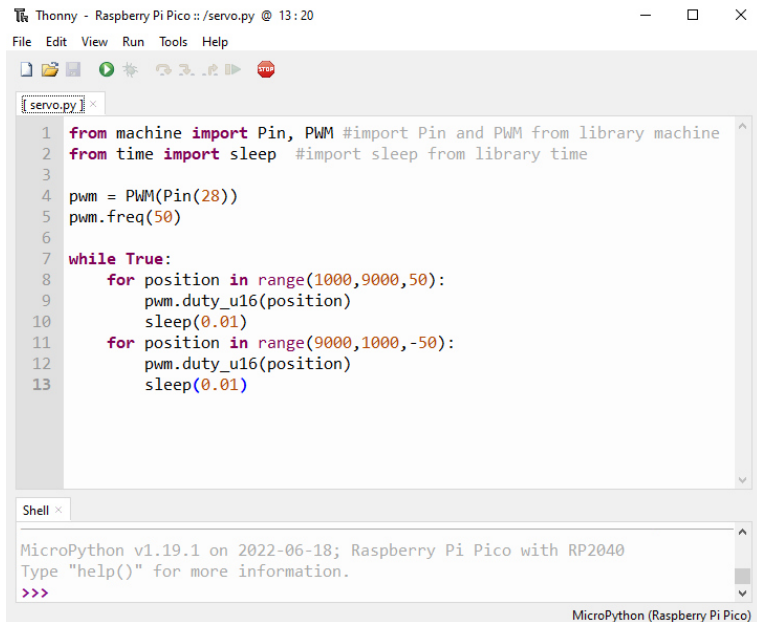


fritzing

- 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

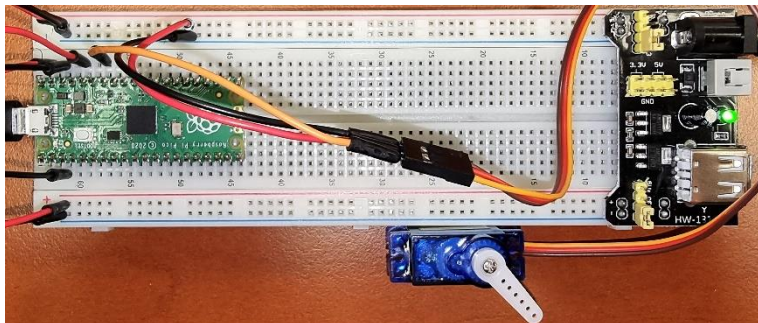
[servo.py]
1 from machine import Pin, PWM #import Pin and PWM from library machine
2 from time import sleep #import sleep from library time
3
4 pwm = PWM(Pin(28))
5 pwm.freq(50)
6
7 while True:
8     for position in range(1000,9000,50):
9         pwm.duty_u16(position)
10        sleep(0.01)
11    for position in range(9000,1000,-50):
12        pwm.duty_u16(position)
13        sleep(0.01)

Shell
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:



9. OLED I2C SSD1306 scherm

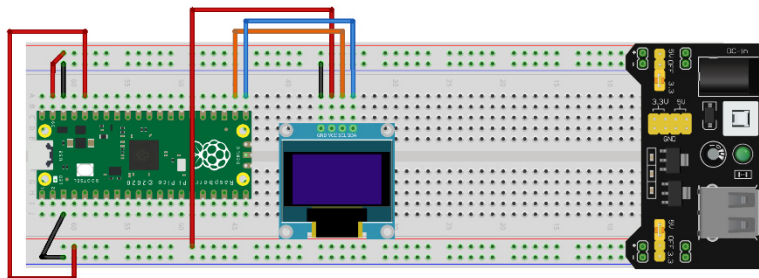
Beschrijving

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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 4 x Man-naar-man jumperdraden
- 1 x Full size broodplank
- 1 x OLED I2C SSD1306 display
- 1 x Micro-USB kabel

Bedradingsschema



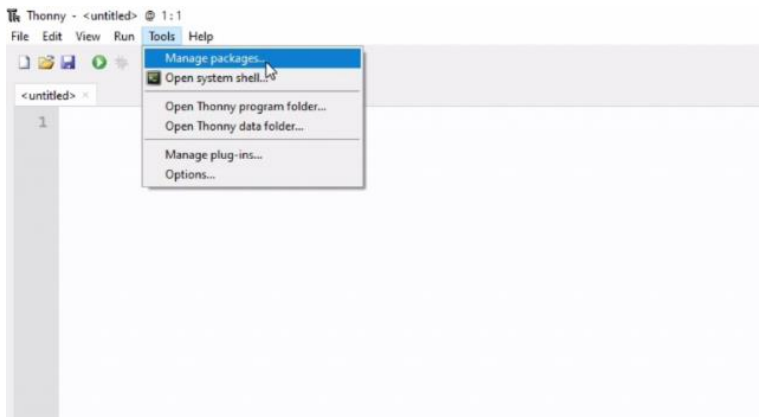
fritzing

- 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

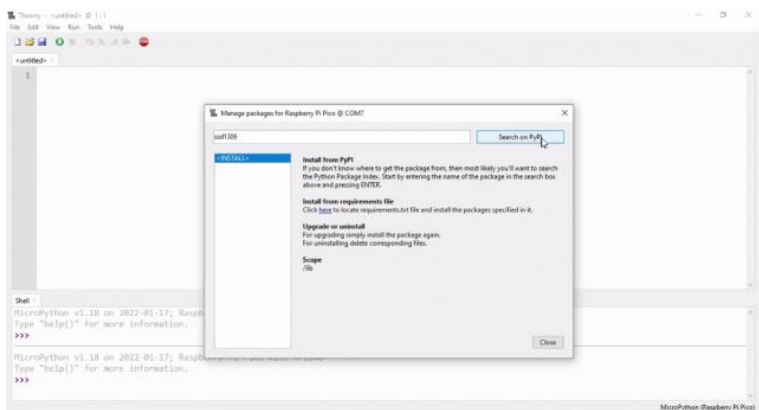
Code

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:

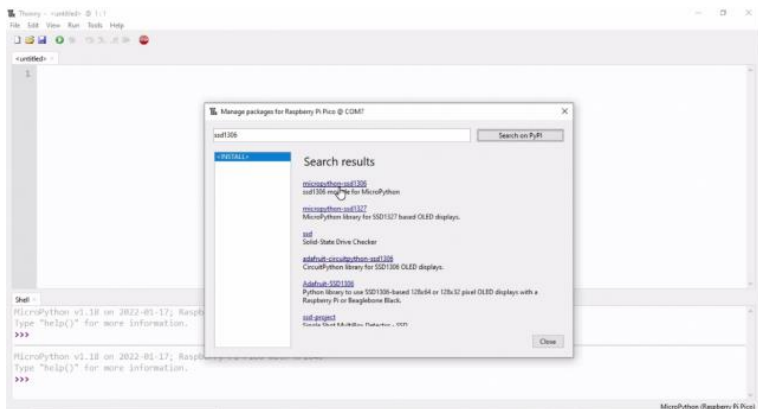
1. Open Thonny en ga naar Extra → Beheer pakketten...



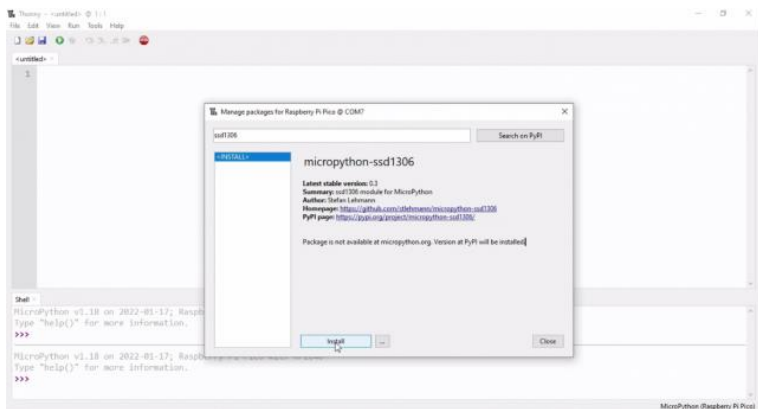
2. Typ SSD1306 in het venster Beheer pakketten en klik op Zoeken.



3. Klik op de micropython-ssd1306



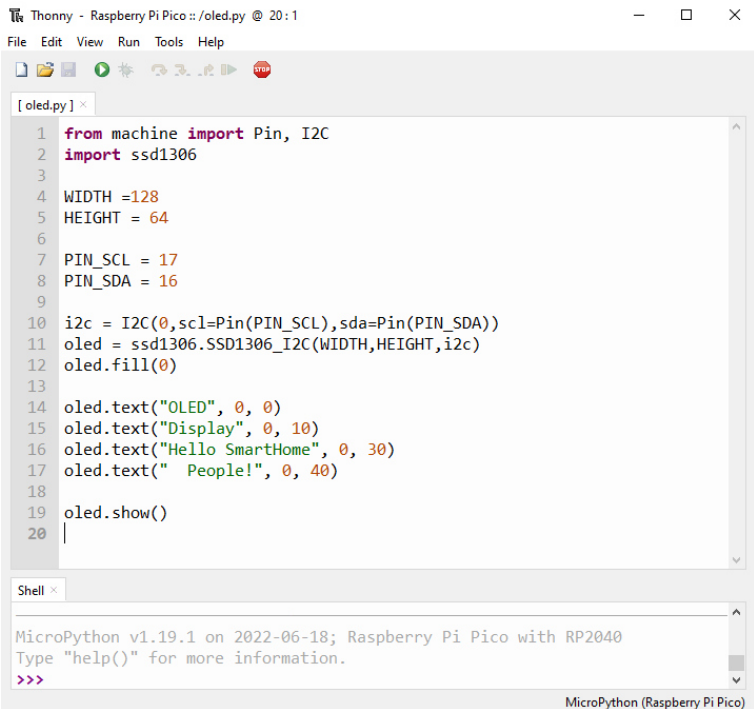
4. Klik in het volgende venster op Installeren.



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

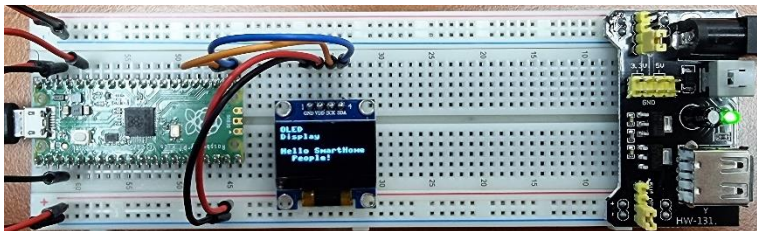
[oled.py] x
1 from machine import Pin, I2C
2 import ssd1306
3
4 WIDTH =128
5 HEIGHT = 64
6
7 PIN_SCL = 17
8 PIN_SDA = 16
9
10 i2c = I2C(0,scl=Pin(PIN_SCL),sda=Pin(PIN_SDA))
11 oled = ssd1306.SSD1306_I2C(WIDTH,HEIGHT,i2c)
12 oled.fill(0)
13
14 oled.text("OLED", 0, 0)
15 oled.text("Display", 0, 10)
16 oled.text("Hello SmartHome", 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.
>>>

MicroPython (Raspberry Pi Pico)
  
```

Voorbeeldafbeelding

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



10. RFID-lezer RC522

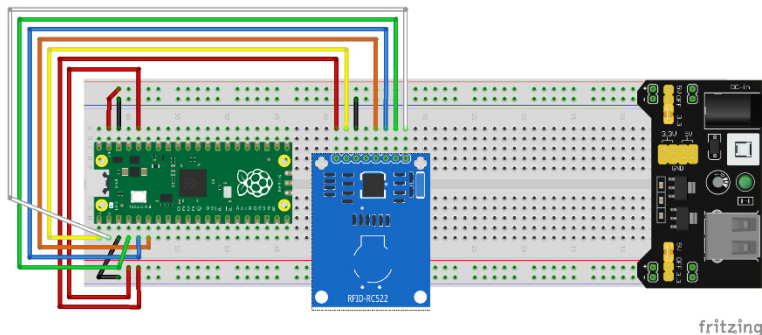
Beschrijving

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

Benodigde materialen

- 1 x Raspberry Pi Pico
- 7 x Man-naar-man jumperdraden
- 1 x Full size broodplank
- 1 x RFID RC522 module
- 1 x Micro-USB kabel

Bedradingschema



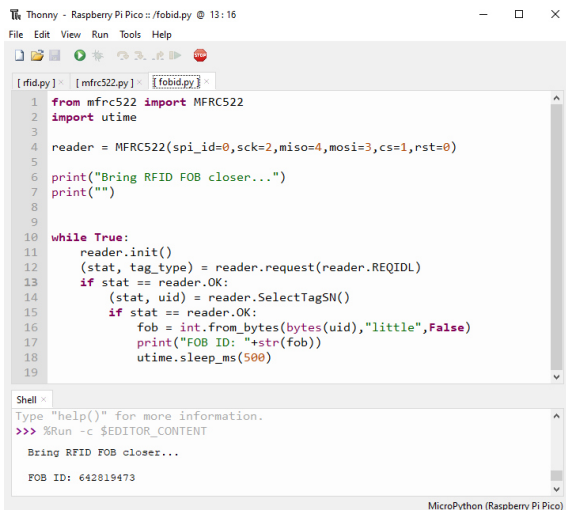
- 3v3 (rode kabel) wordt aangesloten op 3v3-rail (+)
- GND (zwarte kabel) wordt verbonden met GND-rail (-)
- RST (gele kabel) is verbonden met GPIO0 pin
- SDA (witte kabel) is verbonden met GPIO1-pin

- SCK (groene kabel) is verbonden met pin GPIO2
- MOSI (blauwe kabel) is verbonden met GPIO3 pin
- MISO (oranje kabel) is verbonden met pin GPIO4

Code

Net als voor het OLED-scherm hebben we een extra bibliotheek nodig die de RFID-module aanstuurt, namelijk de MFRC522-bibliotheek. We kunnen deze downloaden van <https://github.com/pimylifeup/MFRC522-python/blob/master/mfrc522/MFRC522.py>. Download het bestand en open het in Thonny Python. Klik vervolgens op Bestand → Opslaan als... kies Raspberry Pi Pico en sla je bestand op onder de naam `mfrc522.py`.

Vervolgens moet je de ID van de sleutelhanger identificeren om de RFID-lezer te laten werken. Daarvoor moet je een programma maken dat de RFID Fob leest en zijn ID geeft. Bekijk het onderstaande programma:



```

Thonny - Raspberry Pi Pico ~ /fobid.py @ 13:16
File Edit View Run Tools Help

[rfid.py] [mfrc522.py] [fobid.py]
1 from mfrc522 import MFRC522
2 import utime
3
4 reader = MFRC522(spi_id=0,sck=2,miso=4,mosi=3,cs=1,rst=0)
5
6 print("Bring RFID FOB closer...")
7 print("")
8
9
10 while True:
11     reader.init()
12     (stat, tag_type) = reader.request(reader.REQIDL)
13     if stat == reader.OK:
14         (stat, uid) = reader.SelectTagSN()
15         if stat == reader.OK:
16             fob = int.from_bytes(bytes(uid),"little",False)
17             print("FOB ID: "+str(fob))
18             utime.sleep_ms(500)
19
Shell
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
Bring RFID FOB closer...
FOB ID: 642819473
MicroPython (Raspberry Pi Pico)
  
```

Klik op Afspelen en scan de sleutelhanger. Dit geeft je zijn ID. Nu, terug naar het `rfid.py` programma, moet je het nummer in regel 20 veranderen zodat het overeenkomt met de ID van je fob. Klik dan op Opslaan en start je programma.

MicroPython-code voor de tutorial:

```

Thonny - Raspberry Pi Pico ~/rfid.py @ 11:1
File Edit View Run Tools Help

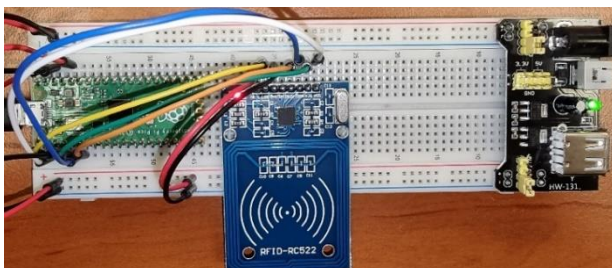
[rfid.py] x [mfr522.py] x [fobid.py] x
1 from machine import Pin
2 from mfr522 import MFRC522
3 import utime
4 import time
5
6 reader = MFRC522(spi_id=0,sck=2,miso=4,mosi=3,cs=1,rst=0)
7
8 print("Bring RFID FOB Closer...")
9 print("")
10
11 |
12 while True:
13     reader.init()
14     (stat, tag_type) = reader.request(reader.REQIDL)
15     if stat == reader.OK:
16         (stat, uid) = reader.SelectTagSN()
17         if stat == reader.OK:
18             fob = int.from_bytes(bytes(uid), "little", False)
19
20             if fob == 642819473:
21                 print("Fob ID: "+ str(fob))
22                 print("Fob is accepted")
23                 time.sleep(1)
24             else:
25                 print("Fob is not accepted")

Shell x
type _help() for more information.
>>> %Run -c $EDITOR_CONTENT
Bring RFID TAG Closer...
Fob ID: 642819473
Fob is accepted

MicroPython (Raspberry Pi Pico)
  
```

Voorbeeldafbeelding

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



Tutorials met sensoren

11. Regendruppelsensor

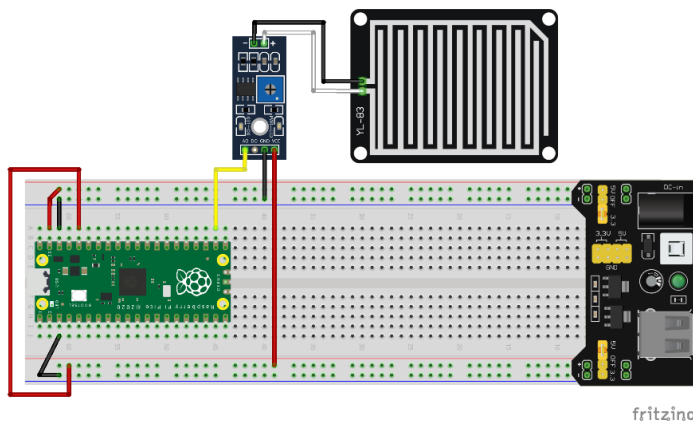
Beschrijving

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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 1 x Full size broodplank
- 1 x Micro-USB kabel
- 3 x Man-naar-man jumperdraden
- regendruppelsensor

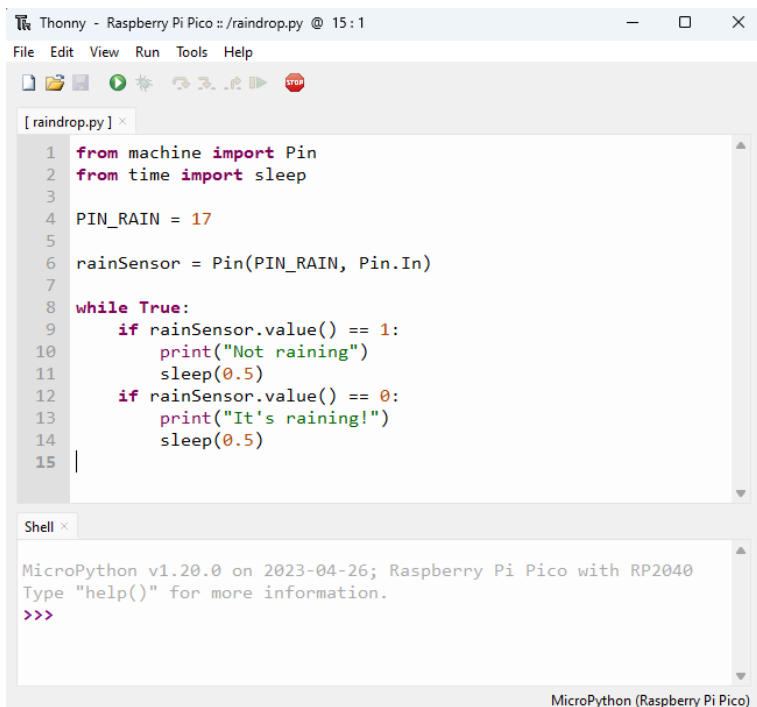
Bedradingsschema



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

Code

MicroPython-code voor de tutorial:



```

Thonny - Raspberry Pi Pico :: /raindrop.py @ 15:1
File Edit View Run Tools Help

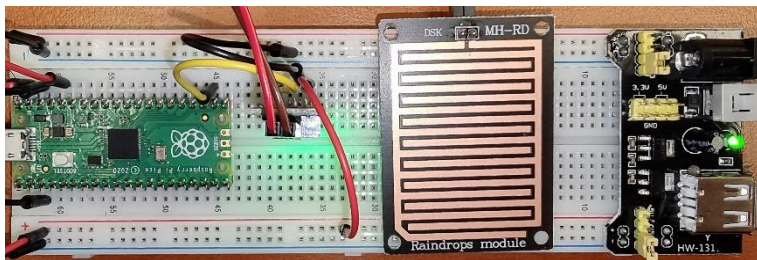
[raindrop.py] x
1 from machine import Pin
2 from time import sleep
3
4 PIN_RAIN = 17
5
6 rainSensor = Pin(PIN_RAIN, Pin.In)
7
8 while True:
9     if rainSensor.value() == 1:
10        print("Not raining")
11        sleep(0.5)
12     if rainSensor.value() == 0:
13        print("It's raining!")
14        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:



12. HC-SR04 Ultrasonic sensor

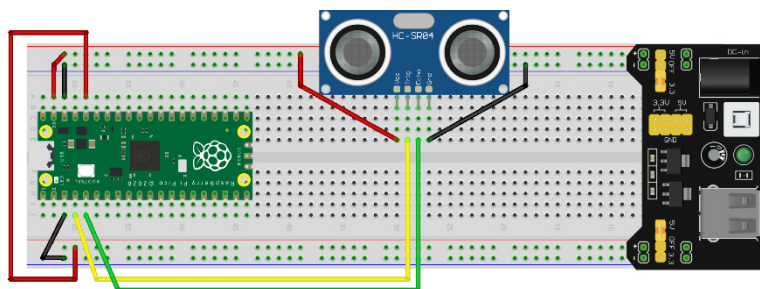
Beschrijving

In deze tutorial leert u hoe u de HC-SR04 ultrasonic sensor aansluit en bedient. Open Thonny Python, ga vervolgens naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla uw 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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 1 x Full size broodplank
- 1 x Micro-USB kabel
- 4 x man-man-jumperdraden
- 1 x HC-SR04 Ultrasonic sensor

Bedradingsschema



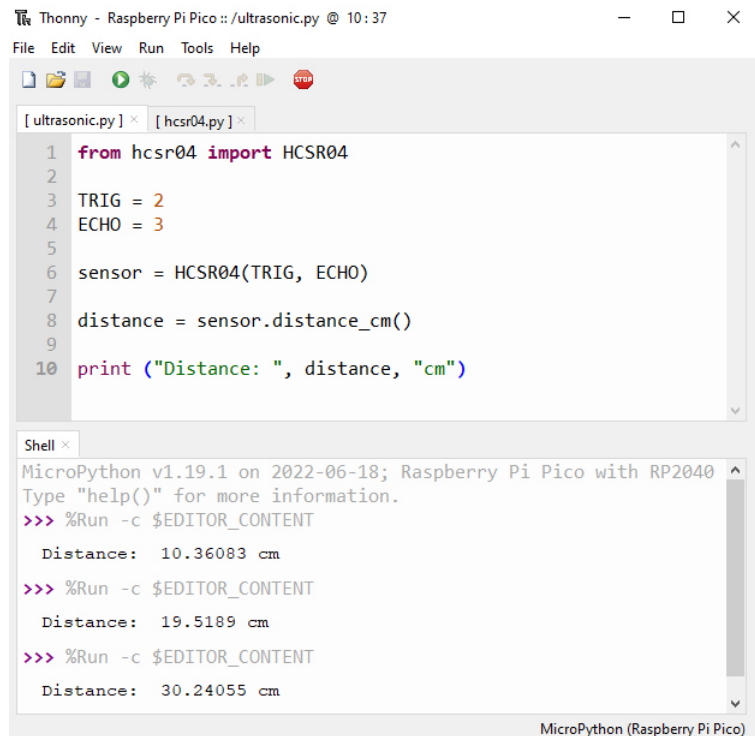
fritzing

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

Code

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

MicroPython-code met behulp van een bestaande bibliotheek:



```

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 (Raspberry Pi Pico)

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

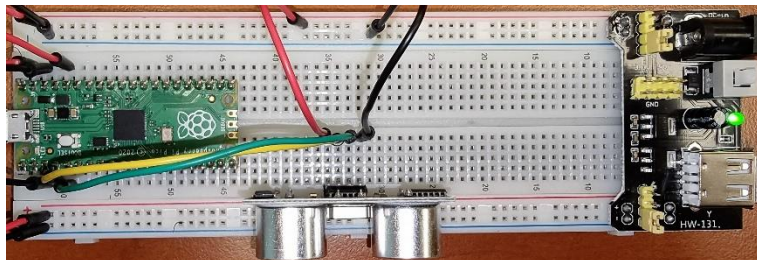
```

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:



13. PIR-bewegingssensor

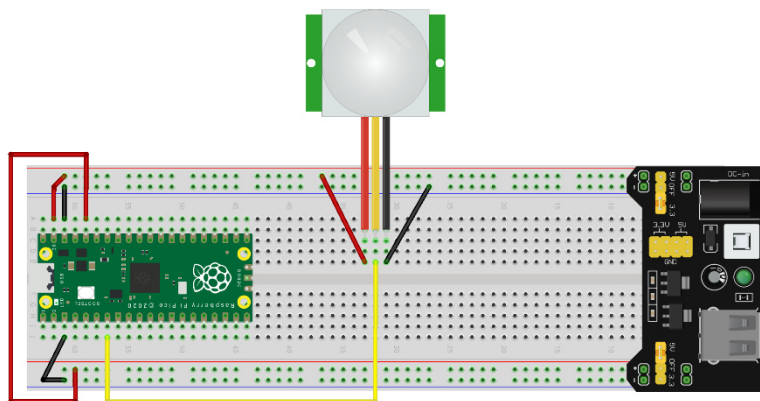
Beschrijving

In deze tutorial leert u hoe u de PIR-bewegingssensor aansluit en bestuurt. Open Thonny Python, ga vervolgens naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla uw 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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 1 x Full size broodplank
- 1 x Micro-USB kabel
- 3 x man-vrouw-jumperdraden
- 1 x PIR-bewegingssensor

Bedradingsschema

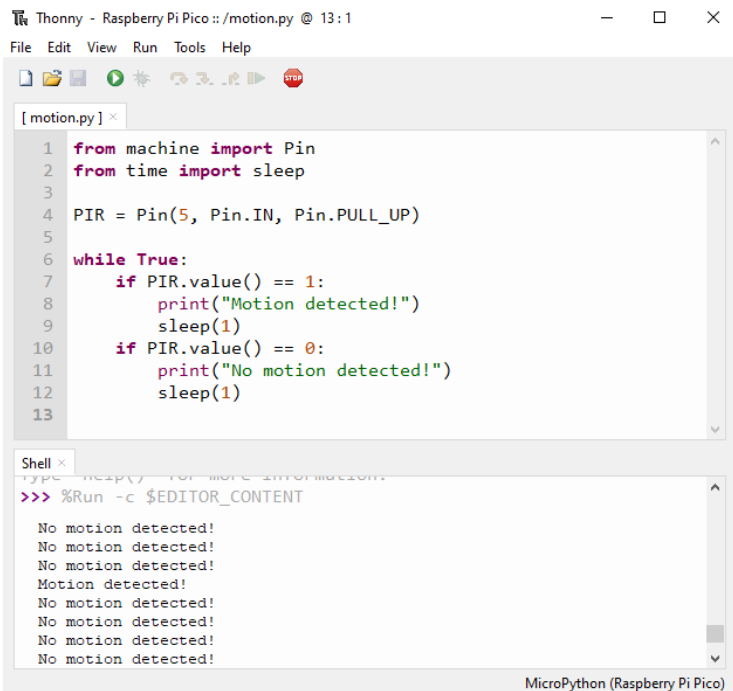


fritzing

- VCC (rode kabel) is aangesloten op 5V rail (+)
- GND (zwarte kabel) is aangesloten op GND-rail (-)
- OUT (oranje kabel) is aangesloten op GPIO5-pin

Code

MicroPython-code voor de tutorial:

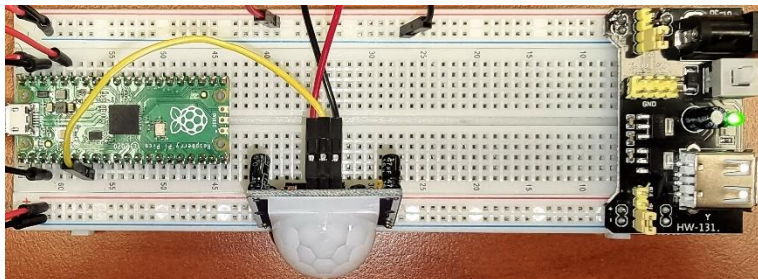


```

Thonny - Raspberry Pi Pico :: /motion.py @ 13:1
File Edit View Run Tools Help
[motion.py] x
1 from machine import Pin
2 from time import sleep
3
4 PIR = Pin(5, 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)
13
Shell x
Type help() for more information.
>>> %Run -c $EDITOR_CONTENT
No motion detected!
No motion detected!
No motion detected!
Motion detected!
No motion detected!
No motion detected!
No motion detected!
No motion detected!
No motion detected!
MicroPython (Raspberry Pi Pico)
  
```

Voorbeeldafbeelding

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



14. DHT11-sensor

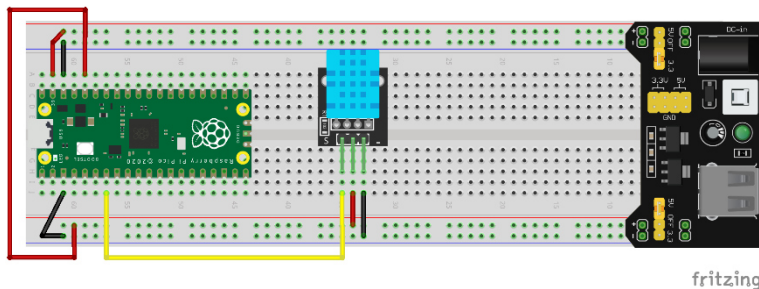
Beschrijving

In deze tutorial leert u hoe u de DHT11 temperatuur- en vochtigheidssensor aansluit en bestuurt. Open Thonny Python, ga vervolgens naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla uw 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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 1 x Full size broodplank
- 1 x Micro-USB kabel
- 3 x man-man-jumperdraden
- 1 x DHT11 temperatuursensor

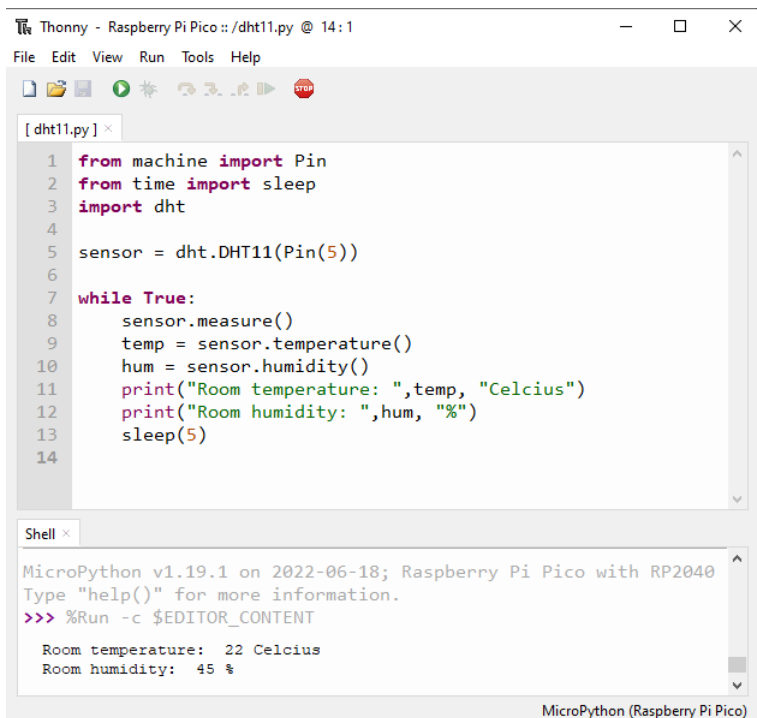
Bedradingsschema



- VCC (rode kabel) is aangesloten op 3v3 rail (+)
- GND (zwarte kabel) is aangesloten op GND-rail (-)
- S (groene kabel) is aangesloten op GPIO5-pin

Code

MicroPython-code voor de tutorial:



```

Thonny - Raspberry Pi Pico :: /dht11.py @ 14: 1
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(5))
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)
14

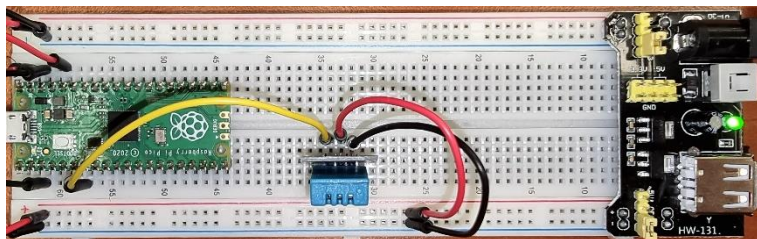
Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT

Room temperature: 22 Celcius
Room humidity: 45 %

MicroPython (Raspberry Pi Pico)
  
```

Voorbeeldafbeelding

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



15. Flame Sensor

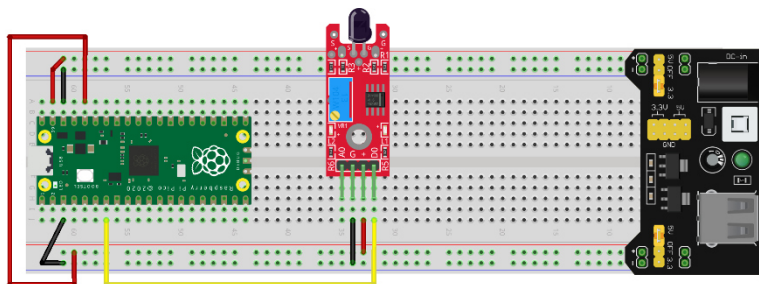
Beschrijving

In deze tutorial leert u hoe u de KY-026 vlamsensor aansluit en bestuurt. Open Thonny Python, ga vervolgens naar Bestand → Opslaan als..., kies Raspberry Pi Pico en sla uw 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.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 1 x Full size broodplank
- 1 x Micro-USB kabel
- 3 x man-man-jumperdraden
- 1 x KY-026 vlamsensor

Bedradingsschema

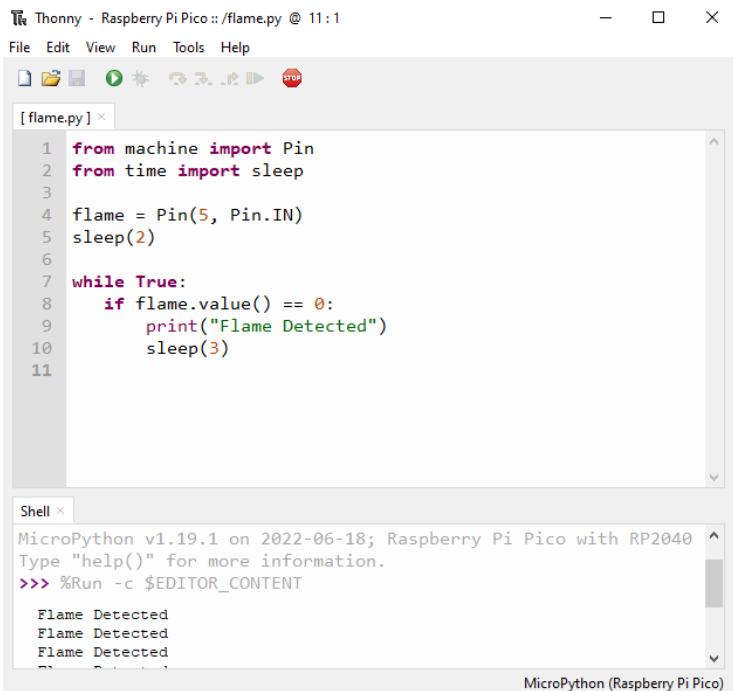


fritzing

- VCC (rode kabel) is aangesloten op 5V rail (+)
- GND (zwarte kabel) is aangesloten op GND-rail (-)
- DO (groene kabel) is aangesloten op GPIO5-pin

Code

MicroPython-code voor de tutorial:



Thonny - Raspberry Pi Pico :: /flame.py @ 11: 1

File Edit View Run Tools Help

```
[flame.py] x
1 from machine import Pin
2 from time import sleep
3
4 flame = Pin(5, Pin.IN)
5 sleep(2)
6
7 while True:
8     if flame.value() == 0:
9         print("Flame Detected")
10        sleep(3)
11
```

Shell x

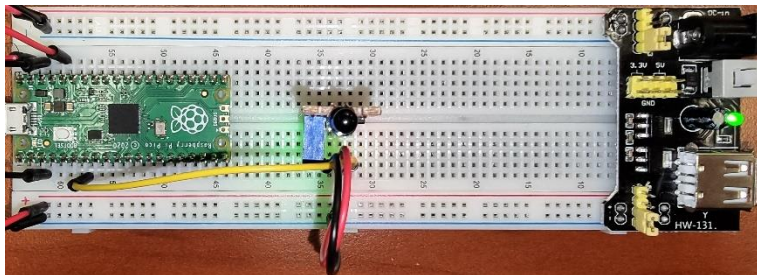
```
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT

Flame Detected
Flame Detected
Flame Detected
```

MicroPython (Raspberry Pi Pico)

Voorbeeldafbeelding

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



16. MQ-135 gasdetectiesensor

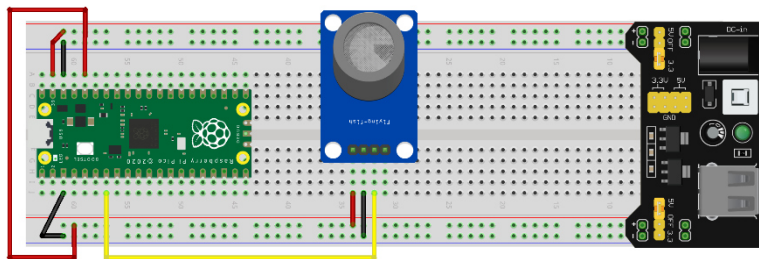
Beschrijving

In deze tutorial leer je hoe je de MQ-135 gasdetectiesensor 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 `gas.py`. Dan is het tijd om de elektronica aan te sluiten en je programma te schrijven. Volg de onderstaande instructies.

Benodigde materialen

- 1 x Raspberry Pi Pico
- 1 x Full size broodplank
- 1 x Micro-USB kabel
- 3 x Male-to-male jumper draden
- 1 x MQ-135 gasdetectie

Bedradingsschema

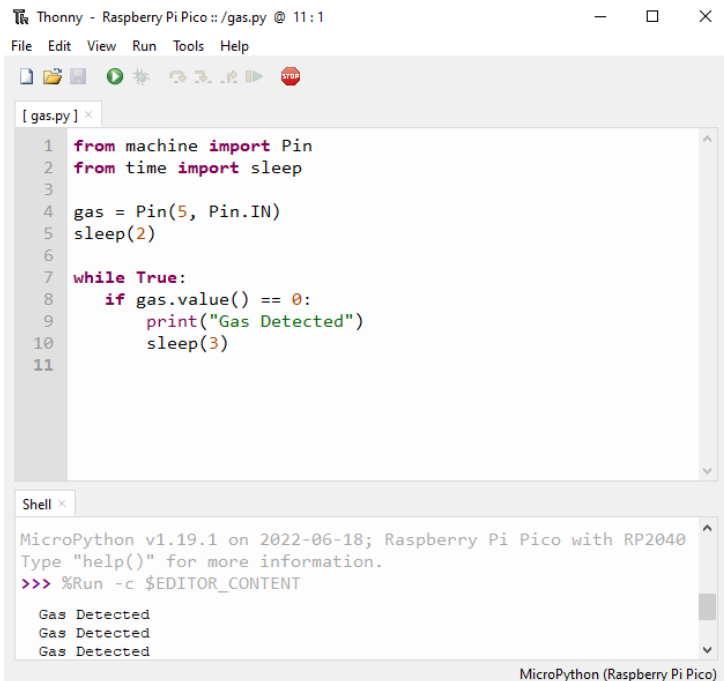


fritzing

- VCC (rode kabel) is verbonden met 3v3-rail (+)
- GND (zwarte kabel) is verbonden met GND-rail (-)
- DO/OUT (gele kabel) is verbonden met GPIO5-pin

Code

MicroPython-code voor de tutorial:



```

Thonny - Raspberry Pi Pico :: /gas.py @ 11:1
File Edit View Run Tools Help

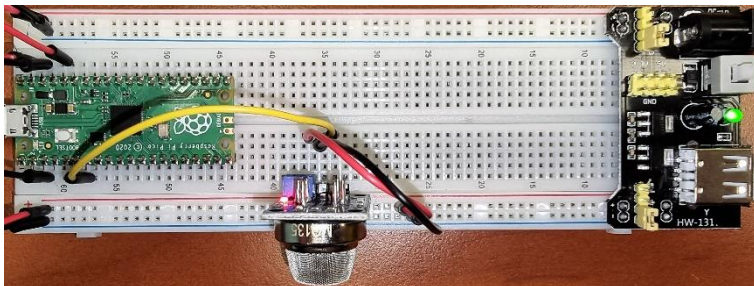
[ gas.py ] x
1 from machine import Pin
2 from time import sleep
3
4 gas = Pin(5, Pin.IN)
5 sleep(2)
6
7 while True:
8     if gas.value() == 0:
9         print("Gas Detected")
10        sleep(3)
11

Shell x
MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>> %Run -c $EDITOR_CONTENT
Gas Detected
Gas Detected
Gas 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 oproepen	<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
Digitale uitgang openen (3.3V uitgang)	<code>led.value(1)</code>	ON
Sluit digitale uitgang (0V uitgang)	<code>led.value(0)</code>	OFF

Digitale ingang		
De klasse Pin oproepen	<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 PULL UP weerstand
	<code>button = Pin(pin_value), Pin.IN, Pin.PULL_DOWN)</code>	Activering van de PULL DOWN weerstand
Invoer lezen	<code>value = button.value(1)</code>	Retourwaarde kan 0 zijn als pin op 0V staat, of 1 als 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
Uitlezen van de ingang	<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 (in 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
Uitlezen van de ingang	<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 sleep-functie	<pre>sleep(sec)</pre>	<code>sec</code> is het aantal seconden dat het programma wordt vertraagd
Aanroepen van de time klasse	<pre>import time</pre>	
Gebruik van de tijd-functie	<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 opdrachten die moeten worden uitgevoerd als aan de aangrenzende voorwaarde is voldaan</p> <p><expr#> (bijvoorbeeld de set <statement2>) wordt uitgevoerd wanneer <expr2> is voldaan</p> <p><statementn>: verzameling opdrachten die worden uitgevoerd wanneer aan geen van de voorwaarden <expr#> 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#>: reeks opdrachten die moeten worden uitgevoerd zolang aan de voorwaarde <expr> is voldaan</p>

Structuur van de For-lus	
<pre>for <var> in <iterable>: <statement(s)></pre>	<p><iterable>: a collection of objects, for example a list containing numbers, alphanumerics, etc.</p> <p><var>: a variable to which the value of the next item in the collection <iterable> is assigned.</p> <p><statement(s)>: a set of instructions executed at each iteration</p>
<pre>for <var> in range(<start>, <end>, <step>): <statement(s)></pre>	<p>range(<start>, <end>, <step>): function that returns a sequence of numbers from <start> to <end>-1, with a <step> difference between two consecutive numbers (<start> and <step> parameters are optional).</p> <p><var>: variable to which the value of the next element of the sequence produced by range is assigned.</p>

		<statement(s)>: a set of instructions executed in each iteration
Diverse		
DHT11	<code>import dht</code>	De DHT-bibliotheek importeren
	<code>sensor = dht.DHT11(Pin(pin_number))</code>	Initialisatie van de sensorvariabele met het bijbehorende <code>pin_number</code> .
	<code>sensor.measure()</code>	Sensorwaarden bijwerken
	<code>temp = sensor.temperature()</code>	Huidige temperatuurwaarde opslaan
	<code>hum = sensor.humidity()</code>	Huidige vochtigheidswaarde opslaan
OLED DISPLAY	<code>from machine import I2C</code>	Importeren van de I2C bibliotheek
	<code>import ssd1306</code>	Importeren van de <code>ssd1306</code> bibliotheek
	<code>i2c = I2C(-1, scl=Pin(1), sda=Pin(0))</code>	Initialisatie van de <code>i2c</code> variabele op de SCL & SDA pinnen van de Pico
	<code>oled_width = 128</code> <code>oled_height = 64</code> <code>oled = ssd1306.SSD1306_I2C(oled_width, oled_height, i2c)</code>	Initialiseren van het scherm
	<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>	Opslaan van berichten in de scherm buffer
	<code>oled.show()</code>	Het tonen van berichten (nodig om berichten te tonen die zijn opgeslagen in de scherm buffer)
	<code>display.pixel(3, 4, 1)</code>	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

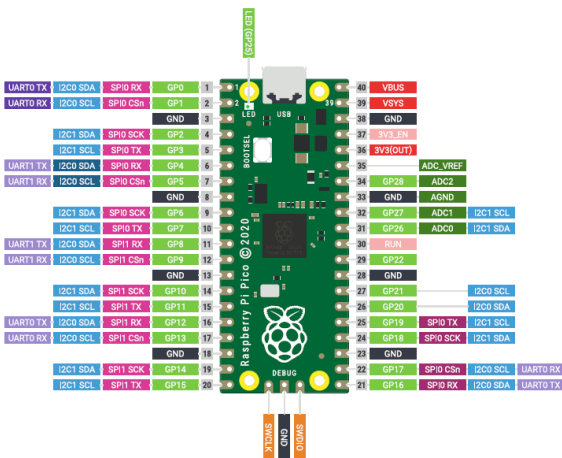


SmartHome
4SENIORS

2021-1-DE02-KA220-ADU-000033587

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



w: sh4seniors.erasmusplus.website
e: info@sh4seniors.erasmusplus.website