Programmeren met de Raspberry Pi





Krijn Hoogendorp

Inhoud

- 1. Introductie 'Programmeren met de Raspberry Pi'
- 2. Kennismaking met de Raspberry Pi
- 3. Raspbian Linux
- 4. Python 1
- 5. Python 2
- 6. LED laten branden
- 7. LED schakelen met button
- 8. LED laten knipperen
- 9. Toepassing van een lichtsensor
- **10.** Toepassing van een bewegingssensor

De rechten met betrekking tot de inhoud van deze uitgave liggen bij Ichabod Tekst. Deze uitgave valt onder een Creative Commons GelijkDelen 4.0 Internationaal-licentie.

Raspberry Pi and the Raspberry Pi logo are trademarks of the Raspberry Pi Foundation.

Deze uitgave is verspreid door Brinkman Uitgeverij.

1. Introductie 'Programmeren met de Raspberry Pi'

In deze uitgave ga je aan de slag met de Raspberry Pi (RPi). De RPi is een kleine computer waar je mee kunt internetten, minecraft spelen en een heleboel andere dingen kunt doen.

Het leuke is dat je ook sensoren op de RPi kunt aansluiten. Zo is het bijvoorbeeld mogelijk om een alarmsysteem te maken: met een bewegingssensor meet je of er iemand in de buurt komt en daarna laat je een geluid afspelen. Het is wel nodig om dit te programmeren. Dat doe je met een computertaal zoals Python.

In deze uitgave staat een aantal opdrachten. Terwijl je die uitvoert leer je meer over de Raspberry Pi.

2. Kennismaking met de Raspberry Pi



De Raspberry Pi ziet er als volgt uit:

bron: https://cyberpi.nl/les-1/

Met alleen de RPi kun je nog niets; er moeten wat dingen op aangesloten worden. Je hebt een scherm nodig, een toetsenbord plus muis om de RPi te bedienen en een adapter voor de stroomvoorziening. De GPIO pinnen (zie de afbeelding) worden gebruikt om sensoren en/of actuatoren op aan te sluiten.

In opdracht 1 ga je het besturingssysteem Raspbian installeren op een USB stick of SD kaart. Daarmee kun je de RPi opstarten en in gebruik nemen.

Computers (dus ook de Raspberry Pi) werken met een besturingssysteem. Bekende voorbeelden van besturingssystemen zijn Windows (Microsoft), iOS (voor Apple) en Android. Het meest gebruikte besturingssysteem voor de RPi is Raspbian (een Linux variant). Met behulp van dit besturingssysteem kun je de RPi opstarten en gebruiken. Het besturingssysteem staat op een geheugenmedium. Bij de eerste Raspberries was het geheugenmedium een SD kaart, maar vanaf versie 3 is de RPi ook voorzien van USBpoorten. Aangezien SD kaarten af-en-toe corrupt raken bij het aan- en uitzetten van de RPi gebruiken we bij voorkeur de USB stick voor het opstarten van de RPi. Niet alle merken USB sticks kunnen worden gebruikt. De volgende werken in ieder geval wel: 1

- Sandisk Cruzer Fit 16GB
- Sandisk Cruzer Blade 16Gb
- Samsung 32GB USB 3.0 drive
- MeCo 16GB USB 3.0

Opdracht 1

In deze opdracht ga je een USB stick gereedmaken om de RPi mee op te starten. Download en installeer het gewenste besturingssysteem. Voer daartoe de volgende stappen uit:

a) Open op een Windows pc de site <u>https://www.raspberrypi.org/downloads/</u>. Op deze site kun je verschillende besturingssystemen downloaden. Omdat we snel naar een hoger niveau toe willen kies je voor Raspbian.



- b) Kies hier voor Raspbian. Vervolgens kies je niet voor de 'Lite versie', maar voor de 'Raspbian with desktop' versie.
- c) Waarschijnlijk heb je een .zip file gedownload. Pak deze vervolgens uit (unzip). Dit kan bijvoorbeeld met *7-zip* voor Windows of *Unarchiver* voor Mac.
- d) Het uitgepakte image-bestand moet je daarna naar de USB stick of SD kaart *flashen*². Dit kun je doen met het programma *Etcher* dat je hier kunt downloaden: <u>https://etcher.io/</u>
- e) Steek je USB stick (of SD kaart) in je computer en start Etcher. Je krijgt dan dit scherm te zien:

¹ Zie ook: <u>https://www.raspberrypi.org/blog/pi-3-booting-part-i-usb-mass-storage-boot/</u>

² Zie ook: https://www.raspberrypi.org/documentation/installation/installing-images/README.md

| ••• | | | | | 0 ¢ |
|-----|--------------|--------------------------------|--------|---|------|
| | + | | | 4 | |
| | Select image | | | | |
| | | | | | |
| _ | | | | | |
| | ETCHE | R is an open source project by | sin.io | | .4.4 |

f) Kies het image-bestand, kies de juiste drive en klik op Flash. Na enige tijd is de SD kaart of USB stick klaar en zie je dit scherm:

| ••• | ÷ — | | | • | 8 | ٠ |
|-----|--------------------------------------|-----------------------------------|----------|------------|---|---|
| | 2018-06-2retch.img 4.82 GB | APPLE SDder Media 8.05 GB | 5% Fla | ETA: 6m49s | | |
| | ₩ETCH | ER is an open source project by 🧯 | resin.io | | | |

g) Zodra het flashen klaar is kun je de USB stick of SD kaart uit je PC of Macbook halen en doorgaan met opdracht 2.

Opdracht 2

Sluit alle hardware aan en start de RPi. LET OP: Voorkom dat de RPi in aanraking komt met metalen deeltjes (zoals een paperclip) die kortsluiting kunnen veroorzaken.

- a) Controleer of je alle onderdelen hebt:
 - ✓ Raspberry Pi (versie 3B+)
 - ✓ Monitor
 - ✓ Muis
 - ✓ Toetsenbord
 - ✓ Behuizing
 - ✓ USB-stick met besturingssysteem Raspbian Linux
 - ✓ HDMI-kabel voor aansluiten van beeldscherm
 - ✓ Eventueel: HDMI-adapter, afhankelijk van de monitoraansluiting
 - ✓ Eventueel: behuizing voor de RPi
 - ✓ Netadapter

- b) Verbind de RPi componenten. Sluit de voeding als laatste aan en schakel hem in.
- c) Als het goed is krijg je op de monitor een afbeelding te zien van een weg. Dat je dit beeld ziet is te danken aan het besturingssysteem.



Zie je dit beeld niet probeer er dan achter te komen wat er verkeerd is gegaan. De belangrijkste vaardigheid van een IT specialist is het onderzoeken van dingen die *niet* werken. Stel jezelf vragen als:

- Brandt er een LEDje op de RPi? Zo niet, dan is er misschien iets mis met de voeding.
- Zie je wel tekst op het scherm verschijnen maar stopt dit opeens? Dan is het besturingssysteem misschien niet goed op de USB stick geïnstalleerd.
- Zie je helemaal niets op het scherm? Misschien heb je de monitor niet ingeschakeld?

Als je denkt dat een van de componenten stuk is, leg dan uit aan je docent waarom je dat denkt. Hij/zij heeft misschien een vervangend exemplaar.

d) ** - NIET VERPLICHT- Kijk of je het internet werkend krijgt door rechtsboven te klikken op:



Selecteer de wifi-router waarvan je gebruik wilt maken en geef vervolgens je gebruikersnaam en wachtwoord.

e) ** - NIET VERPLICHT - Speel 10 minuten Minecraft. Kijk eventueel op https://cyberpi.nl/ voor uitleg over het spelen van Minecraft op de RPi.

3. Raspbian Linux

Een besturingssysteem zorgt ervoor dat een computer start en werkt. Zonder besturingssysteem start en werkt de Raspberry Pi niet. Het meest gebruikte besturingssysteem voor de RPi is Raspbian Linux. Linux is Open Source software³.

In deze paragraaf leer je enkele Linux opdrachten, die je vaak zult gebruiken in je RPi projecten. Er zijn twee manieren om Linux te gebruiken: via de GUI (klikken met de muis) of via de command line (opdrachten typen op het toetsenbord):

• GUI (Graphical User Interface)

De Graphical User Interface van Raspbian Linux lijkt wel een beetje op het Windows scherm of het scherm van Apple computers: je klikt op een icoontje om iets te starten. Klik je bijvoorbeeld op het Raspberry icoontje (de framboos links bovenaan het scherm) dan open je een menu met programma's, net zoals met de Windowsknop.

Je zet de RPi uit door te klikken op *Shutdown*. Probeer dat maar eens. Je zet de RPi vervolgens weer aan door de micro-USB stekker uit de RPi te trekken en er weer in te steken.



Net zoals Windows Verkenner werkt ook Raspbian met een structuur van mappen (Engels: folders) waarin documenten en/of andere mappen zitten. Welke map opent als je op de icoon *Bestandbeheerder* (zie afbeelding) klikt?

| 🖲 🕘 🗮 🗷 🌞 🔇 🗐 | i@raspberrypi: ~ | /D D pi | | | | | |
|--------------------------------|------------------|----------------|-----------|-------|----------|--------|---|
| Bestandbeheerder | | | | | | | |
| Bestand Bewerken Beeld Sort Ga | Gereedschappe | en | | | | | |
| 📷 👪 🏙 📰 🙆 🗇 | ⇒ ⊕ [/hom | e/pi | | | | | ~ |
| - 📜 / | Î | | | | | | |
| 🕀 🛄 bin | | | | 99 | | E | |
| 🕀 🚺 boot | Desktop | Documents | Downloads | Music | Pictures | Public | |
| 🗉 📜 dev | | | | | | | |
| 🗉 🛄 etc | python_gam | Templates | Videos | | | | |
| 🗆 🚺 home | es | | | | | | |
| = 🙆 pi | | | | | | | |
| 🗉 🔳 Desktop | | | | | | | |
| Documents | | | | | | | |
| Downloads | | | | | | | |

³ Open Source betekent dat de software vrij gebruikt en aangepast mag worden. Voor het gebruik van de software hoef je dus niet te betalen.

Command line

De command line biedt de mogelijkheid om via het toetsenbord opdrachten door te geven aan de RPi. Voordat GUI's bestonden werkten vrijwel alle computers met command lines en nog steeds geven veel mensen de voorkeur aan deze methode, omdat die vaak sneller en handiger werkt.

In deze uitgave werk je zowel met de GUI als met de command line. Zo leer je een aantal command line-opdrachten die je kunnen helpen bij het onderzoeken van problemen. De command line start je door op deze icoon te klikken:



Vervolgens krijg je dit scherm te zien:

| Bestand | Bewerken | Tabbladen | Hulp | | |
|----------|------------|-----------|------|--|---|
| i@raspbe | rrypi:~ \$ | | | | 1 |
| | | | | | 1 |
| | | | | | 1 |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

Linksboven zie je *pi@raspberrypi: \$*. Dit betekent dat je bent ingelogd als gebruiker *pi* op de computer genaamd *raspberrypi*. Het \$ teken wordt 'prompt' genoemd. Achter dit prompt-teken typ je je opdrachten. Typ maar eens *whoami* (wie ben ik?) en druk op de Enter-toets. Je ziet dan:

| Ş | whoami |
|----|--------|
| \$ | |
| | ŝ |

Met de opdracht *whoami* controleer je onder welke naam je bent ingelogd. Je krijgt in dit geval als antwoord *pi*. Andere Linux opdrachten zijn:

- *pwd* om te bepalen in welke map je zit; in een map worden documenten bewaard.
- *ls* om een lijst op te vragen van alle documenten en mappen in de huidige map.
- cd omnaareen andere maptegaan.

Bekijk de volgende afbeelding. Begrijp je wat er in de afbeelding gebeurt?



Eerst vind je met *pwd* de huidige map: */home/pi*.

Vervolgens vraag je met *ls* een lijst op van alle documenten (en submappen) in */home/pi*. Met *cd Downloads* open je de submap *Downloads*. Met *pwd* zie je dat de huidige map */home/pi/Downloads* is.

Opdracht 3

a) Open de Text Editor, dat is een simpele tekstverwerker. Zie de afbeelding hieronder hoe dit moet.



b) Noteer één zin in het document en sla het document op als een bestand met de naam *test.txt* in de map */home/pi* (zie afbeelding).

| Bestand Bewerken Zoeken Opties Hulp Nieuw Ctrl+N P Openen Ctrl+O Opslaan Ctrl+S Opslaan als Shift+Ctrl+S P Afdrukken Ctrl+P Afdrukken Ctrl+P P Afsluiten Ctrl+Q | Bestand Bewerken Zoeken Opties Hulp Nieuw Ctrl+N P Opslean Ctrl+S P Opslean als Shift+Ctrl+P Afdrukken Ctrl+P Afdrukken Ctrl+Q P P | | | | | | | |
|---|--|----------|-----------|---------|--------|------|--|---|
| Nieuw Ctrl+N Pomentum Openentum Ctrl+O Opslaan Ctrl+S Opslaan als Shift+Ctrl+P Afdrukken Ctrl+P Afdrukken Ctrl+P Afsluiten Ctrl+P | Nieuw Ctrl+N Openen Ctrl+O Opslaan als Shift-Ctrl+S Afdrukvoorbeeld Shift-Ctrl+P Afdrukken Ctrl+P Afsluiten Ctrl+Q | Bestand | Bewerken | Zoeken | Opties | Hulp | | |
| Openen Ctrl+O Opslaan Ctrl+S Opslaan als Shift+Ctrl+S Afdrukvoorbeeld Shift+Ctrl+P Afdrukken Ctrl+P Afsluiten Ctrl+Q | Openen Ctrl+O Opslaan Ctrl+S Copslaan als Shift+Ctrl+P Afdrukken Ctrl+P Afdrukken Ctrl+P | Nieuw | | C | trl+N | | | 2 |
| Opslaan Ctrl+S Opslaan als. Shift+Ctrl+S Afdrukvoorbeeld Shift+Ctrl+P Afdrukken Ctrl+P Afsluiten Ctrl+Q | Opslaan Ctrl+S Opslaan als Shift+Ctrl+S Afdrukvoorbeeld Shift+Ctrl+P Afdrukken Ctrl+P Afsluiten Ctrl+Q | Opener | n | C | trl+0 | | | |
| Opslaan als Shift-Ctrl+S Afdrukvoorbeeld Shift+Ctrl+P Afdrukken Ctrl+P Afsluiten Ctrl+Q | Opslaan als Shift-Ctrl+S Afdrukvoorbeeld Shift+Ctrl+P Afdrukken Ctrl+P Afsluiten Ctrl+Q | Opslaa | in | C | trl+S | | | |
| Afdrukvoorbeeld Shift+Ctrl+P Afdrukken Ctrl+P Afsluiten Ctrl+Q | Afdrukvoorbeeld Shift+Ctrl+P Afdrukken Ctrl+P Afsluiten Ctrl+Q | Opslaa | in als | Shift+C | trl+S | | | |
| Afdrukken Ctrl+P Afsluiten Ctrl+Q | Afdrukken Ctrl+P Afsluiten Ctrl+Q | Afdruk | voorbeeld | Shift+C | trl+P | | | |
| Afsluiten Ctrl+Q | Afsluiten Ctrl+Q | Afdruk | ken | C | trl+P | | | |
| | | Afsluite | en | C | trl+Q | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

- c) Kijk vervolgens weer in *Bestandbeheerder* of je het document kunt terugvinden.
- d) ** NIET VERPLICHT- Het is ook mogelijk om een bestand te lezen via de command line. Dit doe je met de opdracht *more <naam bestand>*. Hieronder zie je als voorbeeld de opdracht *more test.txt*. Lees het bestand dat je net hebt gemaakt.



 e) ** – NIET VERPLICHT- Je kunt een bestand ook aanmaken of aanpassen via de command line. Dit doe je met de opdracht *nano <naam bestand>*. Hieronder een voorbeeld. Open het bestand dat je hebt gemaakt en pas het aan. Je slaat het bestand op met de sneltoets Ctrl+X. Onder in de afbeelding is dit weergegeven als ^X (^ staat voor Ctrl of Control).

| GNU nano 2.7.4 | Bestand: test.txt | |
|--|-------------------------|---------------------------------------|
| Ik leer over de RPi | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | [1 regel gelezen] | |
| <pre>^G Hulp ^O Opslaan Afsluiten AB Inlezen</pre> | W Zoeken K Knippen | ^J Uitvullen ^C Positie |
| ^X Afsluiten ^R Inlezen | ∧\ Vervangen ^U Plakken | 🚹 Spelling 🦰 Naar regel |

4. Python 1

Elke applicatie of 'app' die je gebruikt is geprogrammeerd in een computertaal. Eén van de meest populaire computertalen is Python. Deze taal is eind jaren '90 ontworpen door de Nederlander Guido van Rossum.



Guido van Rossum (foto: Daniel Stroud)

Python is standaard aanwezig in de Raspberry Pi. Je opent Python door achtereenvolgens met je muis te klikken op de *RPi icoon* (links boven) > *Programmeren* > *Python 3 (IDLE).*



Je krijgt dan de Python command line met een ander type prompt: >>> . Achter deze prompt kun je Python- opdrachten typen. Typ bijvoorbeeld de volgende opdracht en druk op *Enter*:

3+8

Of typ:

varX=9

en druk op Enter. Hiermee maak je de variabele varX gelijk aan het getal 9. Als je dus

```
print(varX)
```

typt en op Enter drukt, dan krijg je het resultaat van de variabele:

9

Of typ:

```
varY="hallo wereld"
```

en druk op Enter. In dit geval wordt variabele varY gelijk aan de zin *hallo wereld*. Bekijk het resultaat met:

print(varY)

en druk op de Enter-toets. Zie de afbeelding.



Merk op: tekst staat tussen aanhalingstekens, terwijl getallen en namen van variabelen niet tussen aanhalingstekens staan. Enkele voorbeelden:

| print("hallo") | tekst |
|-----------------------------|------------------------|
| <pre>print("hallo!!")</pre> | tekst en leestekens |
| print(varY) | variabele |
| print(8) | getal |
| varY = 8 | variabele en een getal |
| varZ="hallo" | variabele en tekst |

Het afgebeelde voorbeeld is je eerste Python programma. Echter, op het moment dat je Python sluit ben je alles kwijt. Daarom moet je je programma bewaren. Dit doe je zo:

- Open het menu en kies achtereenvolgens: File > New File.
- Vervolgens kun je de programmaregels typen, bijvoorbeeld de regels:

varA="dit is mijn eerste programma"

en

print(varA)

- Daarna kun je dit programma uitvoeren door op de toets F5 te drukken, of door te klikken op *Run > Run Module*.
- Eerst vraagt Python nog om het programma een bestandsnaam te geven, bijvoorbeeld:

paragraaf4



Python voegt zelf de bestandsnaamextensie .py toe. Sluit Python en start Python weer. Je kunt je programma weer terughalen met *File > Open* of *File > Recent files*.

Opdracht 4

a) Maak een nieuwe file genaamd *opdracht4.py* met de volgende regels:

```
varA = 224
varB = 2
varC = varA+varB
print(varC)
```

- b) Run het programma.
- c) Voer de volgende wijzigingen door in opdracht 4:

wijs aan varD de waarde "hallo" toe wijs aan varE de waarde "lieve mensen" toe wijs aan varF de som toe van varD + varE

- d) Run het programma.
- e) ****** NIET VERPLICHT- Als je in het programma *opdracht4.py* de variabelen varC en varF bij elkaar wilt optellen krijg je een foutmelding. Leg uit waarom de foutmelding optreedt.

5. Python 2

In de vorige paragraaf heb je geleerd hoe je een Python programma maakt, bewaart en runt. Ook heb je leren werken met variabelen. Om de basis van programmeren te begrijpen moet je ook een aantal andere veelgebruikte constructies kennen. In deze paragraaf krijg je uitleg over *if-then* constructies en over *loops*, spreekt uit: *loeps*. In een loop wordt dezelfde opdracht een aantal malen uitgevoerd.

If-then-else (als-dan-anders)

Elke keer als je inlogt maak je gebruik van een *if-then* constructie: *als* je gebruikersnaam en wachtwoord kloppen *dan* mag je verder. Dit soort *if-then* constructies wordt zeer vaak gebruikt: *als* je een winkeldeur nadert, *dan* gaat de deur open; *als* je op de 1-toets drukt van de tv-afstandsbediening, *dan* schakelt de tv om naar Nederland 1; *als* iemand de deur van de auto openbreekt, *dan* gaat het alarm af. In Python ziet een *if-then* constructie eruit zoals in de afbeelding. Typ de regels over en voer ze uit met F5.



De betekenis is als volgt:

- Met *if(varX == 9)* wordt beoordeeld of varX de waarde 9 heeft. Het antwoord kan waar (True) of *niet waar* (False) zijn:
 - is het antwoord *True* dan worden de ingesprongen regel(s) uitgevoerd. In dit geval betekent dat het printen van de tekst "de variabele heeft de waarde negen".
 - is het antwoord *False* dan worden de *ingesprongen* overgeslagen en het normale verloop van het programma hervat. In dit geval is dat het printen van het woordje "klaar".
- Let op het dubbele == teken.
- De opdracht *then* hoeft niet te worden getypt, want de dubbele punt neemt deze functie over.
- Regels met opdrachten heten *statements*. Na de dubbele punt volgen dus altijd een (of meer) ingesprongen statements. Ook print("klaar") is een statement.

Wil je dat er ook iets gebeurt als het antwoord op if(var==9) *False* is, voeg dan een *else:*-opdracht toe met een of meer ingesprongen statements. Zie de afbeelding.



De betekenis is als volgt:

• Een statement met *else:* gebruik je als er ook iets moet gebeuren wanneer het antwoord op het if-statement False is. Na *else:* volgen een of meer ingesprongen regels. Deze worden alleen uitgevoerd als het antwoord op het if-statement False is. Regels die niet inspringen, horen bij het normale verloop van het programma.

Let op: Het tegenovergestelde van if(varX == 9) is if(varX != 9). Als varX gelijk is aan 9 dan levert (*varX != 9*) als antwoord *False*.

While (net zolang ...)

Een while-loop (lus) zorgt ervoor dat iets steeds blijft doorgaan, totdat er iets verandert waardoor de *loop* stopt. Een voorbeeld uit het echte leven: blijf hardlopen zolang je de finish nog niet hebt bereikt. Bekijk het Python programma in de volgende afbeelding. Het programma blijft een regel op het scherm printen, zolang varY kleiner is dan 7.



De betekenis is als volgt:

- In de eerste regel krijgt de variabele varY de waarde 1 toegewezen.
- In de regel met *while(varY < 7)* voert het programma de twee ingesprongen regels uit zolang varY kleiner is dan 7.
- De eerste ingesprongen regel *print(varY)* zet de waarde van varY op het scherm.
- De tweede regel *varY=varY+1* telt elke keer 1 op bij de waarde van varY; deze variabele hoogt dus steeds met 1 op: 2, 3, 4 etc.
- De twee ingesprongen regels worden herhaald zolang varY kleiner is dan 7.
- Is varY gelijk aan 7 dan vervolgt het programma weer zijn normale verloop met:

print("klaar")

Opdracht 5

a) Maak een Python programma van een while-loop die van 1 tot 10 telt en de getallen 1, 2, 3 tot 10 onder elkaar op het scherm print. Zodra het getal 5 is bereikt, moet het programma *vijf* printen in plaats van 5. Hint: gebruik voor het printen van *vijf* een if-statement:

| <pre>if(varZ == 5): print("vijf")</pre> | |
|--|---|
| Python 3.5.3 Shell 🗕 🗖 | × |
| <u>F</u> ile <u>E</u> dit She <u>l</u> l <u>D</u> ebug <u>O</u> ptions <u>W</u> indow <u>H</u> elp | |
| <pre>Python 3.5.3 (default, Jan 19 2017, 14:11:04) [GCC 6.3.0 20170124] on linux Type "copyright", "credits" or "license()" for more information. >>> =================================</pre> | |

 b) ** – NIET VERPLICHT – Er bestaat ook nog een *for-loop*. Maak het onderstaande programma. Zoek op het internet informatie over de werking van deze Python forloop.

> <u>File Edit Format Run Options Window Help</u> varDieren = ["kat", "hond", "vogel"] for x in varDieren: print(x)

Verander het programma zodanig, dat het programma kijkt of *hond* in de verzameling *varDieren* zit. Is dat het geval dan laat je deze zin op het scherm printen: "Er zit een hond in de lijst". Gebruik voor dit laatste een if-then constructie.

6. LED laten branden

In deze paragraaf gaan we een LED laten branden met behulp van de Raspberry Pi. Hiervoor heb je de volgende spullen nodig: een breadboard, een LED, enkele verbindingsdraadjes of 'jumper wires' en een weerstand. Je krijgt deze materialen van je docent plus enkele andere zaken die je nodig hebt in de paragrafen 6 t/m 13.

Uitleg componenten

Een **breadboard** is een experimenteerbordje waarop je elektronische componenten met elkaar kunt verbinden. Je zou de componenten ook aan elkaar kunnen solderen maar een breadboard werkt gemakkelijker en sneller.



bron: <u>https://cyberpi.nl/les-7/</u>

Vaak ziet een **LED** eruit zoals in de afbeelding hieronder. Het ene draadje is langer dan het andere. Een LED werkt alleen als stroom via het lange draadje *binnenkomt* en via het korte draadje weer *uitstroomt*.



Jumper wires zijn de verbindingsdraadjes die je o.a. gebruikt om het breadboard te verbinden met Raspberry Pi (via de pinnetjes).



Een **weerstand** wordt gebruikt om de stroom die door een circuit (stroomkring) loopt te verminderen.



In de opdrachten ga je een stroomkring maken. Elektrische stroom loopt altijd van een positieve spanning (Plus) naar een lagere spanning (Min). Spanning wordt aangeduid met volt. Hier zie je een voorbeeld van een stroomkring waarin de spanning wordt geleverd door een AA-batterij. De batterij heeft twee 'polen': op de pluspool van de batterij staat een spanning van +1,5 volt, op de minpool van de batterij staat 0 volt. In plaats van min of 0 volt spreken we ook wel van 'ground', afgekort GND. De stroomkring bestaat verder uit een lampje en twee draden:



De stroom loopt van de pluspool (+1,5 volt) door de lamp naar de minpool (0 volt, GND). Als in de stroomkring een onderbreking zit dan kan er geen stroom lopen en zal de lamp niet branden. Met de Raspberry Pi kun je een stroomkring maken omdat de RPi pinnetjes heeft die vergelijkbaar zijn met de Plus van de batterij, en pinnetjes die vergelijkbaar zijn met GND. De overige pinnetjes kun je zo programmeren, dat er soms wel en soms geen spanning (+3,3 Volt) op staat.

In dit overzicht zie je welke pinnetjes spanning (Plus) afgeven en welke pinnetjes ground (Min) zijn:



Plus: pinnen 1 en 17 (spanning +3,3 volt) en pinnen 2 en 4 (spanning +5 volt)Min: pinnen 6, 9, 14, 20, 25, 34 en 39 (ook wel ground, GND of aarde genoemd)

Opdracht 6

- a) Als voorbereiding op vraag 6 t/m 10 controleer je eerst of je de volgende materialen hebt (sommige daarvan zijn pas nodig in andere opdrachten):
 - ✓ Breadboard
 - ✓ LEDs
 - Weerstanden van 330 Ω
 - ✓ Jumper wires (verschillende soorten)



b) Je gaat met de RPi een stroomkring maken om een LED te laten branden. De kring loopt van pin 1 (+3,3 volt) naar pin 39 (ground). Daartussen zit een LED maar ook een weerstand van 330Ω (Ohm). De weerstand is nodig om de stroom door de LED te beperken. De korte draad van de LED zit rechts. Het ziet er zo uit:



De stroom loopt van pin 1 (+3,3 volt) door de rode draad naar het breadboard. Via het lange blauwe draadje loopt de stroom naar de weerstand en vervolgens door de LED. Via het korte blauwe draadje vloeit de stroom via het breadboard en de lange zwarte draad terug naar de *ground* pin van de RPi (pin 39).

Begrijp je dat dit een stroomkring is? Bekijk in de volgende afbeelding hoe de verbindingen in het breadboard lopen.



7. LED schakelen met button

Een van de materialen is een drukknopschakelaar of *button*. Je kunt deze ergens in de stroomkring plaatsen. Als de button niet ingedrukt is, dan is de stroomkring onderbroken. Zodra de button wordt ingedrukt is de stroomkring gesloten en zal het lampje branden.

Hieronder een afbeelding waaruit blijkt hoe het werkt. De button heeft vier pinnen A t/m D. A en C zijn altijd met elkaar verbonden, evenals B en D. De pinnen A en B (of D) zijn alleen met elkaar verbonden als de button wordt ingedrukt.



Let op: er zijn ook andere buttons verkrijgbaar. Sommige buttons werken precies andersom: als de knop is ingedrukt wordt de stroomkring onderbroken.

Opdracht 7

a) Plaats de button in de stroomkring. Je kunt onderstaand schema gebruiken:



8. LED laten knipperen

Bekijk de stroomkring van opdracht 6b: als de RPi ingeschakeld is, zal ook de LED branden. In de meeste gevallen wil je dat een LED alleen aan is als jij dat wilt, bijvoorbeeld als het donker wordt. Daartoe sluit je de LED (en de weerstand) aan op een van de andere pinnetjes. Op de meeste pinnetjes staat pas spanning, als een computerprogramma daarvoor de opdracht geeft. Als voorbeeld gebruik je pin GPIO18.

In opdracht 8 zet je met een Python programma 3,3 volt spanning op pin GPIO18⁴. Het Python programma ziet er zo uit:

```
#!/usr/bin/python
from gpiozero import LED
from time import sleep
varLed = LED(18)
while True:
    varLed.on()
    sleep(1)
    varLed.off()
    sleep(1)
```

De Python programmeertaal ondersteunt een aantal standaard commando's, waaronder *if..then* en *while..true*. Aanvullende commando's zitten in zogenaamde bibliotheken of *libraries*. Om die commando's te kunnen gebruiken moet je ze importeren. Dat gebeurt in het voorbeeldprogramma in regel 2 en 3: de libraries heten *gpiozero* en *time*, de benodigde commando's zitten in de code *LED* en *sleep*.

Betekenis van dit programma:

- #!/usr/bin/python \rightarrow Python bibliotheken zijn te vinden in de map /usr/bin/python.
- from gpiozero import LED → Uit de library gpiozero importeer je de code⁵ om een LED aan te sturen.
- Uit de library *time* importeer je de code *sleep*. De code *sleep(1)* zorgt voor een pauze van 1 seconde.
- varLed = LED(18) → aan de variabele varLed wordt de waarde toegekend van LED(18), dus van pin GPIO18.
- Via de regel *while True:* wordt de code die eronder staat voortdurend herhaald; dit heet een lus of *loop*.

Opdracht 8

a) Maak het circuit als volgt:

⁴ Let op: als je de library *gpiozero* gebruikt dan zijn de pinnen als GPIO genummerd. Kijk nog even goed naar de nummering in hoofdstuk 6.

⁵ Krijg je een foutmelding dan is dit bestand waarschijnlijk nog niet beschikbaar. Hier vind je uitleg: https://www.raspberrypi-spy.co.uk/2012/05/install-rpi-gpio-python-library/



- c) Open een nieuwe file in Python 3 zoals beschreven in paragraaf 4, dus klik op het Raspberry Pi icoontje bovenaan, en kies *Programmeren > Python 3*. Daarna kies je *File > New File*.
- d) Typ onderstaande code over. Zorg ervoor dat de regels onder *while True:* iets inspringen; gebruik daarvoor de Tab-toets. Voer het programma uit via *Run > Run Module*. Als het goed is gaat de LED nu knipperen.

```
#!/usr/bin/python
from gpiozero import LED
from time import sleep
varLed = LED(18)
while True:
    varLed.on()
    sleep(1)
    varLed.off()
    sleep(1)
```

Je kunt de loop afbreken met de sneltoets Ctrl+C of via menu-optie *Shell > Interrupt Execution:*



9. Toepassing van een lichtsensor

Bij het materiaal zit ook een lichtsensor of LDR (Light Dependent Resistor). Als deze wordt aangesloten op de Raspberry Pi dan komt er informatie beschikbaar over de hoeveelheid licht die op de LDR valt. Hiermee is het bijvoorbeeld mogelijk om licht in te schakelen zodra het donker wordt.

Om de LDR goed te laten werken is ook een *condensator* nodig (in het Engels: capacitor). Voor deze opdracht is het niet nodig om precies te weten wat een condensator doet, maar onderaan de pagina vind je wel een korte uitleg.⁶ Zorg dat je een condensator hebt van 1 microfarad of 1 μ F. Op de condensator staat meestal 1 μ F of 1 μ F.

De gegevens die de PRi over de hoeveelheid opvallend licht verzamelt, kun je zichtbaar maken op het beeldscherm. In paragraaf 2 en 3 heb je al geleerd dat je met de opdracht *Print()* iets op het scherm kunt laten weergeven.

Opdracht 9

a) Maak een stroomkring die eruit ziet zoals in de afbeelding. De condensator heeft een plus-draad en een min-draad. Let erop dat je de min-draad via de korte blauwe draad verbindt met GND:



b) Maak op de inmiddels bekende manier een Python programma met onderstaande code.
 Zorg ervoor dat de regels onder *while True:* weer een stukje inspringen; gebruik daarvoor de Tab-toets. Test vervolgens het programma.

```
#!/usr/bin/python
from gpiozero import LightSensor
from time import sleep
varSensor = LightSensor(18)
while True:
    varSensor.wait_for_light()
    print("licht is aan")
    varSensor.wait_for_dark()
```

⁶ De condensator slaat elektrische energie op, waardoor de spanning op de plus-draad van de condensator toeneemt. Hoe meer licht, des te sneller dit gebeurt. Het programma meet hoe lang het duurt eer de condensator een spanning van 1,8 volt heeft bereikt.

c) Bij stap b kun je niet aangeven hoe donker het moet zijn, voordat de LED inschakelt. Het programma werkt met een standaardinstelling. Wil je daar meer invloed op hebben, dan moet je eerst de waarde zien weer te geven die de lichtsensor meet. Dit doe je met de regel:

print(varSensor.value)

Pas het programma aan zodat de waarde op het scherm verschijnt.

- d) NIET VERPLICHT -- Verander de code zodanig, dat de melding "het is donker" op het scherm verschijnt, als er géén licht op de lichtsensor valt. Dit kun je doen met behulp van de *if-else* opdracht van paragraaf 5:
 - if (varSensor.value < 0.75):
 print("het is donker")</pre>
- e) NIET VERPLICHT-- Laat een LED branden als het donker is. Hint: Kijk goed naar het stroomcircuit en de code van opgave 8. De nieuwe code zet je in de *if-loop*.

10. Toepassing van een bewegingssensor

De volgende sensor waar we mee aan de slag gaan is de PIR bewegingssensor, kortweg PIR. PIR staat voor Passive InfraRed. Deze sensor 'ziet' de infrarood warmte die mensen en dieren uitstralen. In dat geval stuurt de PIR via zijn uitgang (pin 2, zie afbeelding) een 5 volt signaal. Dit signaal kan de Raspberry Pi opvangen en er vervolgens iets mee doen. In opdracht 10 wordt bijvoorbeeld een melding op het scherm getoond.



bron: https://maker.pro/raspberry-pi/tutorial/how-to-interface-a-pir-motion-sensor-with-raspberry-pi-gpio

Opdracht 10

a) Maak een stroomkring die er uitziet zoals in onderstaande afbeelding. Let op: de PIR werkt op 5 volt. Die spanning betrekt de PIR van pin 2 of 4 van de RPi.



b) Schrijf een Python programma dat de sensor leest. Hieronder een voorbeeld:

#!/usr/bin/python

```
from gpiozero import MotionSensor
varPir = MotionSensor(23)
while True:
    if varPir.motion_detected:
        print("er is beweging")
    else:
        print("geen beweging")
```

c) – NIET VERPLICHT—Heb je ook een zoemer bij de hand, pas dan de stroomkring en het programma zodanig aan, dat je een geluid hoort als er beweging is. De code om een

zoemer te laten werken is:

```
from gpiozero import Buzzer
varZoemer = Buzzer(24)
varZoemer.on()
```

Natuurlijk moet de zoemer ook worden aangesloten. De zoemer heeft twee draden, een rode en een zwarte. De rode moet met GPIO24 verbonden worden en de zwarte met een *ground* pin.

Het ROC van Amsterdam, House of Digital en PA Consulting organiseren in het schooljaar 2018/2019 een wedstrijd waarin de Raspberry Pi centraal staat. Leerlingen van VMBO-, MAVO-, HAVO- en MBO-scholen gaan aan de slag met het thema "Verbeter welzijn - doe een uitvinding!". Wie kan op basis van de Raspberry Pi een toepassing voor in huis ontwikkelen waar mensen echt iets aan hebben?

Meer informatie over deze wedstrijd vind je hier:

https://www.paconsulting.com/insights/2018/raspberry-pi-nederland/



Meer weten over de Raspberry Pi? In april 2019 verschijnt bij Brinkman Uitgeverij het boek *Raspberry Pi en het Internet of Things*.

Houd de website <u>www.brinkman-uitgeverij.nl/lesmateriaal</u> in de gaten.



Recente boeken van Brinkman Uitgeverij:









