

# Digitale IO

**update** 14 september 2018: figuur 6 en figuur 3.

## Hardware

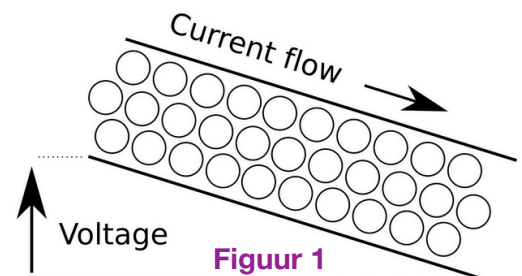
**Digitale IO**, *digital Input/Output*, is een van de meest eenvoudige manieren om met hardware te communiceren: je hebt te maken met slechts twee toestanden: AAN of UIT. **Digitaal** betekent dit dat er ofwel (positieve)spanning op een pin staat (AAN), ofwel géén spanning (UIT). Hiermee kan je een LED<sup>1</sup> aanzetten of uitzetten of een knop inlezen (**ingedrukt** of **niet-ingedrukt**).

Met slechts enkele regels micropython zet je een LED aan, of, uit en als je dat achter elkaar doet, heb je een knipperend LED.

### Basis electronica

Bij het maken van een schakeling heb je te maken met **spanning** (notaties: **U**, of **V**), stroom (notatie: **I**) en weerstand (notatie **R**).

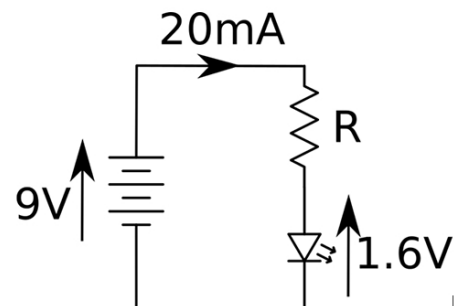
Een analogie, om het verband tussen deze parameters in te zien, is je een pijp met water voor te stellen dat schuin staat (zie figuur 1). De hoogte stelt de spanning voor, de watersnelheid de stroom, en vernauwing van de pijp hoe hard het water stroomt de weerstand. Hoe schuiner de pijp staat (hoge spanning), hoe harder het water stroomt. Hoe nauwer de pijp, hoe meer weerstand het water ondervindt.



Figuur 1

### Wet van Ohm

In een gesloten elektronisch circuit zorgt een batterij voor een spanningsval tussen de "+" en "-" aansluitpunten, waardoor er stroom loopt in het circuit (zie figuur 2). Dit betekent dat op elk punt in het circuit er een spanningsverschil is met een referentie punt. Dat referentiepunt noemt men 'grond' (Engels: *ground*, notatie **GND**). Het circuit moet **gesloten** zijn, zodat de stroom van "+" naar "-" kan gaan.



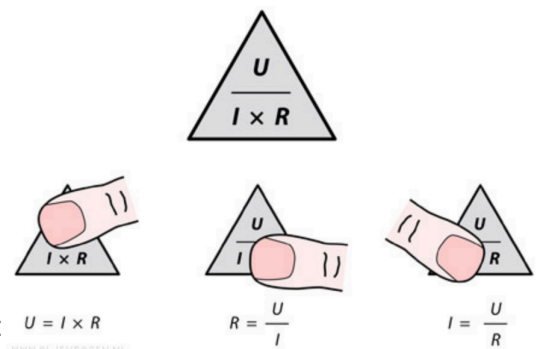
Figuur 2

Het volgende schema van een elektronisch circuit<sup>2</sup> toont een batterij, een LED en een weerstand (zie figuur 2). Het hele circuit is gesloten, dat betekent dat de positieve pool van de batterij via draden, LED en weerstand is verbonden met de negatieve pool van de batterij. De negatieve pool van de batterij is het referentiepunt, ofwel GND.

Het verband tussen de spanning(sverschil tussen uiteinde), stroom en weerstand wordt gegeven door de **wet van Ohm**. Deze luidt:

$$U = I \cdot R$$

**U** = spanning(sverschil met GND! in eenheden van **Volt**), **I** = stroom (in eenheden van **Ampère**), en **R** = weerstand (in eenheden van Ohm (notatie:  $\Omega$ )). Voor 'U' wordt nog veel ook het symbool 'V' gebruikt.



<sup>1</sup> LED = **L**ight **E**mitting **D**iode - zie <https://www.instructables.com/id/LEDs-for-Beginners/>

<sup>2</sup> **Bouw dit schema NIET**, want de spanning van 9V is teveel voor de LoPy4, waar je **maximaal 3.3 V** mag aansluiten.

Als je twee van de drie parameters weet, dan kan je de 3de uitrekenen met deze wet.

Bijvoorbeeld zoals in schema (figuur 2):

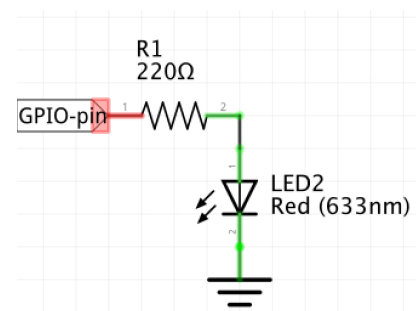
$U = 9V$ , het voltage over de LED is  $1.6V$  (dit haal je uit de datasheet van een LED onder de naam **forward voltage**), de stroom is  $20\text{ mA}$ , dan is de weerstand  $R$ :

$$R = U / I = (9\text{ V} - 1.6\text{ V}) / 20\text{ mA} = (9 - 1.6) / 0.020 = 370\ \Omega\ (\Omega: \text{spreek uit als 'Ohm'}).$$

Het betekent verder dat als je de **stroom** **maximaal**  $20\text{ mA}$  wilt houden (bijvoorbeeld omdat bij grotere stroom er anders iets doorbrandt), de **weerstand minimaal**  $370\ \Omega$  moet zijn. Dit gebruik je om te berekenen wat de weerstand je moet hebben bij **aansluiten** van een LED op een GPIO<sup>3</sup>-pin van de microcontroller. Kortom, **sluit nooit een LED zonder weerstand aan** (de term *short circuit* vergeet je voorlopig niet meer).

In **figuur 3** staat een schakeling van een LED (**LED1**) dat met een weerstand **R1** verbonden is met een GPIO-pin van de microcontroller, en anderzijds met **GND**. Intern de microcontroller is de GPIO-pin verbonden met **GND**, zodat er sprake is van een gesloten circuit.

De spanning op de GPIO-pin ga je regelen met micropython code en is ofwel  $3.3\text{ V}$  (**HIGH**-state) ofwel  $0\text{ V}$  (=GND, **LOW**-state). Als verder gegeven is dat de stroom door de GPIO-pin, en door het circuit, maximaal  $12\text{ mA}$  mag zijn ([LoPy4 datasheet](#), pagina 26), de forward voltage van de LED  $1.6\text{ V}$  is, hoe groot moet dan de weerstand minimaal zijn?



**Figuur 3**

Wet van Ohm:  $R = U / I$

$$I \leq 12\text{ mA}, U = (3.3 - 1.6)\text{ V}$$

$$\Rightarrow R \geq (3.3 - 1.6) / 0.012 \approx 142\ \Omega.$$

In de praktijk heb je te maken met een weerstanden reeks zoals '**E12**' en in het pakket zijn er weerstanden van  $220\ \Omega$  aanwezig en dat dan een veilige waarde om te gebruiken.

### Wetten van Kirchhoff

Met deze wetten kan je uitspraken doen over de stromen en spanningen op elk punt in een elektrisch circuit. Lees Wikipedia: [electriciteitswetten van Kirchhoff](#).

Ik geef slechts een voorbeeld van de 1ste (stroomwet) en 2de (spanningswet) wet van Kirchhoff.

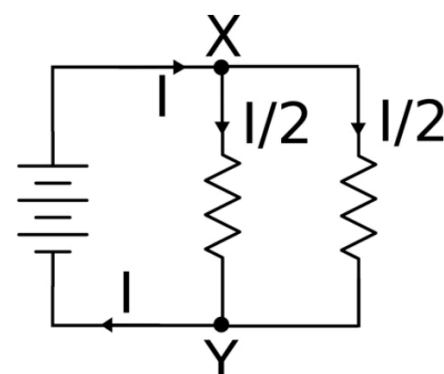
#### Kirchhoff's Stroomwet

Gegeven het circuit van figuur 4 met twee parallel geschakelde weerstanden. Wat is de stroom op punt X en op punt Y?

**Kirchhoff's Stroomwet** zegt dat op elk punt in het circuit, de ingaande stroom gelijk moet zijn aan de uitgaande stroom.

**punt X:** ingaande stroom  $I$  is gelijk aan de uitgaande stroom door ene EN andere weerstand. als de weerstanden gelijk zijn, betekent dat de helft van de stroom  $I$  door elk van de weerstand gaat.

**punt Y:** De ingaande stromen zijn  $1/2 I$  door ene weerstand en  $1/2 I$  door andere weerstand, en, volgens Kirchhoff is de som ervan is gelijk aan de uitgaande stroom  $I$ .



**Figuur 4**

<sup>3</sup> GPIO = General Input/Output pin.

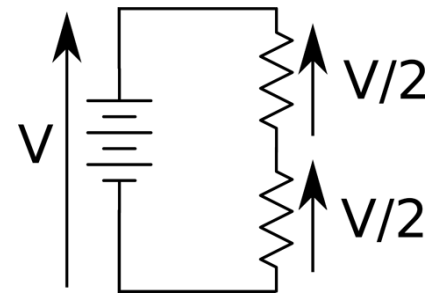
## Kirchhoff's Spanningswet

Gegeven het circuit van twee weerstanden in serie geschakeld (figuur 5).

**Kirchhoff's Spanningswet** zegt dat alle spanningen/voltages tussen de verschillende punten in een circuit opgeteld 0 oplevert.

De spanning  $V = V/2 + V/2$ , ofwel  $V - (V/2 + V/2) = 0$

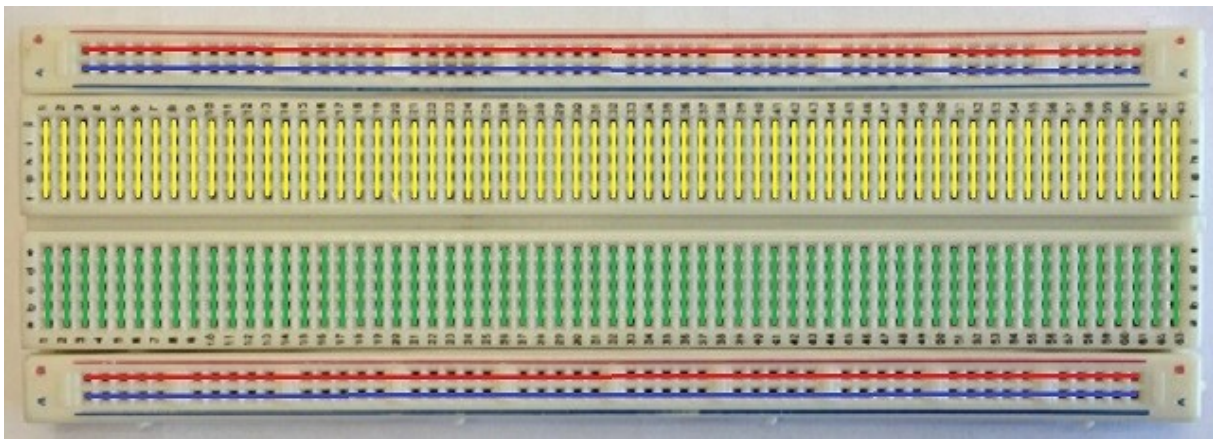
**Let op:** het teken van  $V$  is tegengesteld aan dat van  $V/2$  en  $V/2$ !



Figuur 5

## Breadboard

**Breadboard** is een plastic bord met een aantal kleine gaatjes en wordt gebruikt voor het bouwen en testen van circuits. Het heeft gaten op hen die intern zijn verbonden in een bepaald patroon zoals getoond in de onderstaande afbeelding. De gaten die via een groene lijn zijn verbonden, vertegenwoordigen dat ze intern zijn verbonden. De **rode** lijn geeft de voedingslijn (spanning) aan, die normaal is aangesloten op de "+" pool van batterij of voedingsbron. De **blauwe** lijn geeft Ground aan, die normaal verbonden is met de aarde van het circuit. Breadboards zijn er in diverse maten, meest bekende zijn [full-size](#), [half-size](#) en [tiny-size](#).

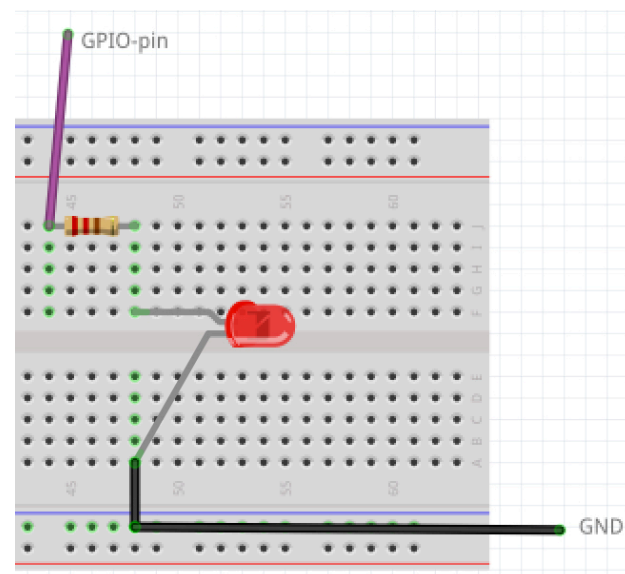


Om een breadboard voor jouw circuit te gebruiken, volgt je het schakelschema en sluit je een component op een lijn aan. **Sluit de batterij/voeding altijd op het laatst aan nadat je alle verbindingen hebt gecontroleerd.** Houd veelvoorkomende fouten in de gaten, zoals het mixen van GND en toevoer, aansluiten op een verkeerde rail, buttons, sensors of leds niet goed geplaatst, etc.

In het volgende diagram (figuur 6) maakten we een circuit voor een LED op een breadboard. In feite bouw je het schema van figuur 3.

Het diagram is gemaakt met een programma genaamd [Fritzing](#), waarmee je ook schakelschema's kan maken. Het programma is Open Source - donaties zijn welkom- en te downloaden [hier](#).

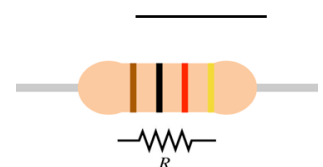
Lees ook het uitgebreide Sparfun's artikel over breadboards: [How to use a breadboard](#).



Figuur 6

## Weerstanden

digital IO



Een goede bron is de volgende website: <https://components101.com/resistor>

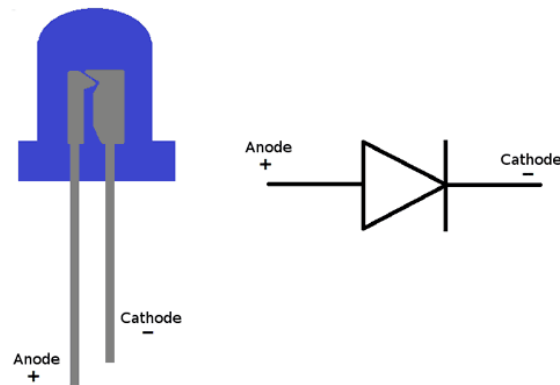
Online weerstandscalculators: [Resistorreader](#), [Resistor color code calculator](#), en vele anderen op het web.



## LED

Een LED is een halfgeleider-lichtbron met twee elektroden, die licht uitstraalt wanneer geactiveerd. Wanneer een geschikt voltage op de LED-terminal wordt toegepast, kunnen de elektronen recombineren met de elektronengaten in de inrichting en energie in de vorm van fotonen vrijgeven. Dit effect staat bekend als elektroluminescentie. De kleur van de LED wordt bepaald door de energiebandafstand van de halfgeleider.

Zie verder de beschrijving van de 5mm ronde LEDs, w.o. waarden voor **forward voltage** de volgende website: <https://components101.com/diodes/5mm-round-led>



Er zijn nog vele andere basis componenten, zoals condensator, transistor en diode, en zijn een [cursus/module](#) op zich. Dit geldt ook voor solderen van de componenten tot een elektrisch circuit op PCB-boarden, etc. Voorlopig vallen deze buiten de scope van de module.

# Opdrachten - digital IO

Maak de volgende opdrachten. De opdrachten hoeven niet gemaakt te worden in de opgegeven volgorde, maar kies er uit die je interessant en leerzaam vindt.

1. [Hello World](#)
2. [Button en LED](#)
3. [Spaceship](#)
4. [Cylon / Knight-rider lichtpatronen](#)
5. [Binaire LED teller](#)
6. [Binaire Up/Down LED teller](#)
7. [LED dobbelsteen](#)

## 1. Hello World schakeling

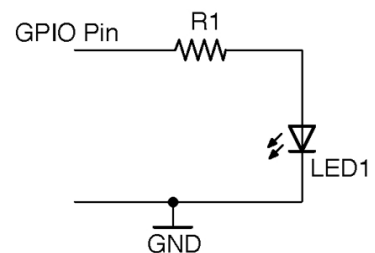
### Materiaal

- 1x LED, kleur maakt niet
- 1x 220  $\Omega$  weerstand

De 'Hello world' van de microcontrollers is een knipperende LED. Er zit een RGB<sup>4</sup>-LED op de LoPy4 (Micropython code: Pycom documentatie sectie '[All Pycom Device examples](#)' - RGB LED), maar je gaat nu een externe LED aansluiten.

Verwijder de spanning op de LoPy4 door de micro-USB kabel uit de computer te halen. **Het bouwen van een elektronisch circuit doe je ALTIJD zonder dat er spanning aanwezig is. Dit om te voorkomen dat je kortsluiting maakt en het development board onherstelbaar beschadigt.**

Bouw het schema om een LED en weerstand aan te sluiten op een GPIO-pin. Sluit de weerstand aan op een GPIO-pin van het Expansion Board, bijvoorbeeld 'G28', en op de breadboard de LED en een draad naar de GND-pin van het Expansion board. Uiteraard kan je ook draden vanuit de Expansion board op de breadboard aansluiten en op de breadboard de LED en weerstand monteren. Denk eraan dat je de LED op een correcte manier aansluit (kathode, ofwel de korte poot, sluit je aan op GND, de anode, ofwel de lange poot, sluit je aan de weerstand (positive spanning). Een verkeerd aangesloten LED geeft geen licht.



Let op de layout van een breadboard - [pagina 3-6](#).

Calculator voor berekenen weerstand bij een gegeven LED: <http://led.linear1.org/1led.wiz>

Check de aansluiting minimaal 2x voordat je spanning op het circuit zet middels de micro-USB kabel.

Zodra je het schema hebt gebouwd en gecheckt dat het correct is, zet je spanning op het circuit door de micro-USB kabel in de computer te steken. Start Atom op, als je dat al niet had gedaan en maak een verbinding met de LoPy4. Je ziet de REPL-prompt `>>>` en je bent in seriële REPL-mode. Klik in Pymakr-panel, zodat je Micropython code achter de REPL-prompt kan intikken. Deze code wordt op de microcontroller uitgevoerd.

Je gaat nu met Micropython code de LED aan- en uitzetten!

<sup>4</sup> RGB = Red-Green-Blue. Code voorbeeld micropython: Pycom documentatie sectie 4.2.11 [RGB LED](#).

In de volgende code-statements ga ik ervan uit dat de LED is aangesloten op de GPIO-pin 'G28' op het Expansion board. Verander dat als je een andere GPIO-pin gebruikt.

Tik de volgende coderegels in achter de REPL, regel voor regel en na elke regel de RETURN/ENTER toets.

```
import machine
LED = machine.Pin('G28', mode=machine.Pin.OUT)
LED.value(1)
```

Als goed is, is de LED aangegaan en geeft licht.

Tik in: `LED.value(0)`

en RETURN-toets, en de LED gaat uit!

Als je nu handmatig snel de twee coderegels `LED.value(1)` en `LED.value(0)` intikt, knippert de LED. Uiteraard hebben we daar een betere, geautomatiseerde, oplossing voor: maak een *loop* waarin je de LED aan- en uitzet met een wachttijd ertussen. In (micro)Python code:

```
import time
while True:
    LED.value(1) # LED aan
    time.sleep(1)
    LED.value(0) # LED uit
    time.sleep(1)
```

Waarom de wachttijden? Zie wat er gebeurt als je ze weglaat. Dit is een kenmerk van embedded programmeren en je zult een wachttijd (Engels: *delay*) wel vaker tegenkomen.

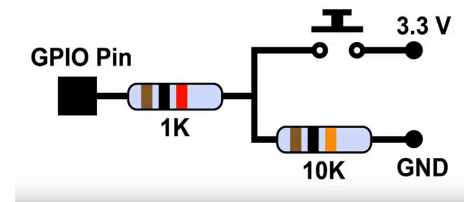
Gefeliciteerd, je hebt de **Hello World** van de embedded systemen gemaakt.



## 2. Button en LED schakeling

### Benodigheden:

- 1 rode/groene/gele LED,
- weerstanden (220Ω, 1kΩ, 10kΩ),
- 1x drukknop
- breadboard



**Figuur 7**

➡ Verwijder de spanning op de LoPy4 door de micro-USB kabel uit de computer te halen.

Breid het 'Hello World' circuit uit met een button/knop (schakelschema figuur 7). Verbind ene poot van de button met 3V3, en de andere, tegenoverstaande poot met een weerstand van 1kΩ, die, op zijn beurt, is verbonden met een GPIO-pin op Expansion board, bijvoorbeeld 'G10'. Op Verbind met GND via een 10kΩ weerstand (zie figuur 7). De weerstand van 1kΩ is bedoeld om de stroom te beperken als je de button indrukt.

De bedoeling is dat als je de button indrukt, de LED aangaat, en als je de button loslaat, de LED weer uitgaat. Er is nu sprake van géén gesloten circuit tussen button en LED. Er zal een micropython programma gemaakt moeten worden, dat de button moeten lezen (*ingedrukt* of *niet*) en dat vervolgens de LED aan- of uitzet.

Gegeven de volgende code, die leest of de button is ingedrukt of niet, aangenomen dat de button op GPIO-pin 'G10' van het Expansion board is aangesloten. Let op: de schakeling zo gemaakt dat GPIO-pin bij open button (nominal) LOW is en als de button ingedrukt wordt, de GPIO-pin HIGH is.

```
# volgende is een alternatief voor import machine, maar wat efficiënter...
from machine import Pin
button = Pin(Pin.exp_board.G10, mode=Pin.IN)
# Pin.exp_board.G** is een manier om direct met G-namen te werken
while(True):
    if(button.value() == 1):
        print("presssss !!!!")
```

Je merkt dat met bovenstaande code een button indrukken meerdere output tekst geeft. Dat heeft te maken met het feit dat de button, elektronisch bekeken, zeer veel keren ingedrukt is. Om dit effect te verhelpen moet je inbouwen een button-'**debounce**', zoals dat genoemd wordt.

Maak het programma zodanig dat één keer indrukken ook één tekstmelding geeft. Maak gebruik van het artikel van Tony DiCola dat een eenvoudige oplossing met goede uitleg geeft. Kijk of je dat met de LoPy4 en bovenstaande schakeling voor elkaar krijgt.

Maak het programma af, zodat de LED aan (bij indrukken) of uitgezet wordt (button losgelaten).

Bestudeer ook de volgende bronnen:

1. Forum PyCom: [user button configuration](#). Hierbij wordt interrupt handling gedaan. Zie ook Pycom documentatie - sectie [class Pin - Control I/O Pins](#).
2. Tony DiCola's artikel: [digitale IO - digitale Input](#). **Let op:** button is connected to GND instead of 3V3.

### 3. Spaceship

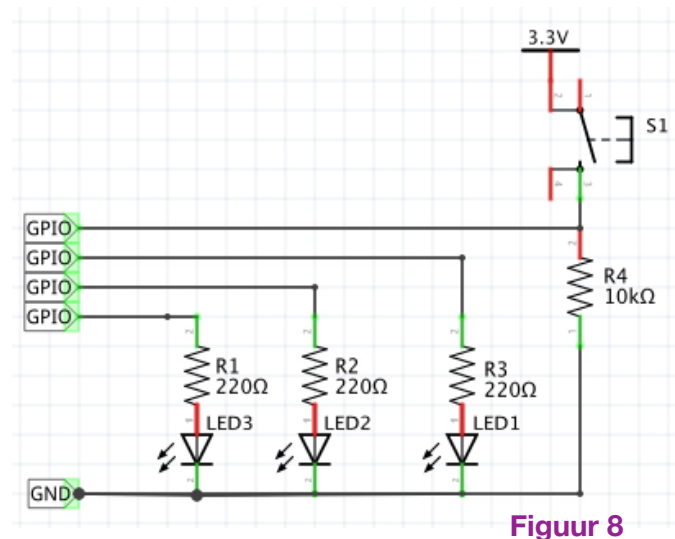
#### Benodigheden:

- 2 rode LEDs, 1 groene LED,
- weerstanden (220  $\Omega$ , 1x 10 k $\Omega$ ),
- drukknop
- breadboard

☞ Verwijder de spanning op de LoPy4 door de micro-USB kabel uit de computer te halen.

In deze opdracht is het de bedoeling om met de LEDs een console voor een ruimteschip te maken. De LEDs gaan aan bij het indrukken van de drukknop. Jij bepaalt wat elk lichtje betekent, en kan bijvoorbeeld zijn *engage hyperdrive*, of, *fire photonlaser*, etc.

Het scenario is als volgt: een groen LED staat aan, totdat je de drukknop indrukt. Vervolgens gaat de groene LED uit en beginnen de twee rode LEDs te knipperen.



Maak het circuit en een micropython programma dat het scenario realiseert.

**Creatief:** maak van dun karton een console met gaten waarin de LEDs en de drukknop passen. Je hebt dan een 'echt' spaceship console gemaakt.

### 4. Cylon / Knight-rider lichtpatronen

#### Benodigheden

- 5 LEDs,
- 5 weerstanden 220  $\Omega$ ,
- breadboard

In mijn Github repository [smart-devices-code-samples](#) staan voorbeelden van een lichtpatronen generator op basis van 3 LEDs. Micropython code. Download de voorbeelden in jouw LoPy4-projectfolder.

☞ Verwijder de spanning op de LoPy4 door de micro-USB kabel uit de computer te halen.

- Maak een elektronisch circuit waarin plm. 5 LEDs zijn aangesloten volgens het 'Hello World' schakelschema aan het Expansion Board. Gebruik het breadboard voor de weerstanden en de LED's. Let op dat elke LED via een weerstand verbonden is met een GPIO-pin.
- Gebruik de gegeven cylon patroon generator (zie [Github repository 'lib'](#)) om 5 LEDs aan te sturen in plaats van 3.

**Tip:** werk modulair d.w.z. zet alle code in subfolder en maak een aanroep vanuit [main.py](#).

- **Uitdaging:** Maak een *knightrider* lichtpatroon functie `knightrider(leds, seq, n):` de LEDs gaan van links naar rechts en van rechts naar links aan en uit. Zie [Youtube](#), tijdlijn plm. 14 seconden na begin video.



## 5. Binaire LED teller

**Inspiratie:** [Binary LED counter](#)

Bouw een schakeling met 5 LEDs en weerstanden (220 Ohm), een knop en bijbehorende weerstand (limiting resistor van  $1k\Omega$ ). Maak een Micropython programma waarin binair optellen wordt uitgevoerd en gevisualiseerd met de LEDs.

---

## 6. Binaire Up/Down LED teller

**Inspiratie:** [Binary Up/Down LED counter](#)

Breidt de schakeling van de vorige opdracht (binaire teller) uit met nog een knop en weerstand, en maak een micropython programma waarmee je binair op- en aftrekken kan, gevisualiseerd door de LEDs.

---

## 7. LED dobbelsteen

**Inspiratie:** [LED Dice](#)

LED dobbelsteen: maak een schakeling met 6 LEDs (en weerstanden) waarmee je een dobbelsteen kan maken.