

[LESSEN]

Figure 1: Boek 2: Outputs en inputs

Contents

Voorwoord	1
Les 5: 7-segment display	2
Les 6: RGB	9
Les 7: Potmeter	21

Voorwoord

Dit is een boek over mBot, geschreven voor jonge tieners. Een mBot is een machine die je kunt programmeren. Dit boek leert je hoe je elektronica op de mBot aansluit, en hoe je deze programmeert.

Over dit boek

Dit boek heeft een CC-BY-NC-SA licentie.



Figure 1: De licentie van dit boek

(C) Richèl Bilderbeek en alle docenten en alle leerlingen

Met dit boekje mag je alles doen wat je wilt, als je maar verwijst naar de oorspronkelijke versie op deze website: https://github.com/richelbilderbeek/mbot_voor_jonge_tieners. Dit boekje zal altijd gratis, vrij en open blijven.

Het is nog een beetje een slordig boek. Er zitten tiepvauten in en de opmaak is **niet altijd even mooi**. Omdat dit boek op een website staat, kan iedereen die dit boek te slordig vindt minder slordig maken.

Les 5: 7-segment display

In deze les gaan we met een 7-segment display gebruiken

5.1 7-segment display: extensiecentrum

Als je geen ‘Makersplatform’ blokken hebt, installeer deze:

- Klik op de ‘+’ onder de blokken

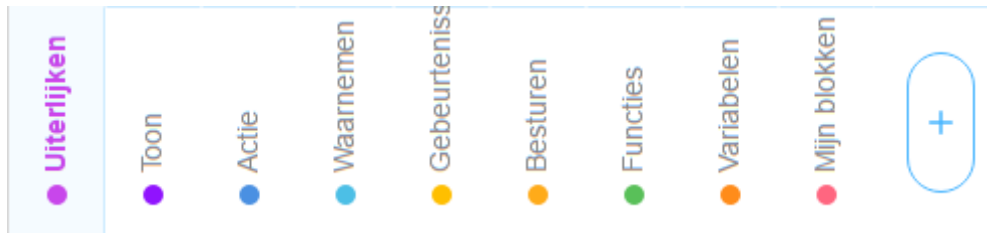


Figure 2: Geen ‘Makersplatform’ blokken. Klik op de ‘+’

- In het ‘Extensie Centrum’, zoek ‘Makerplatform’ en klik eronder op ‘Toevoegen’

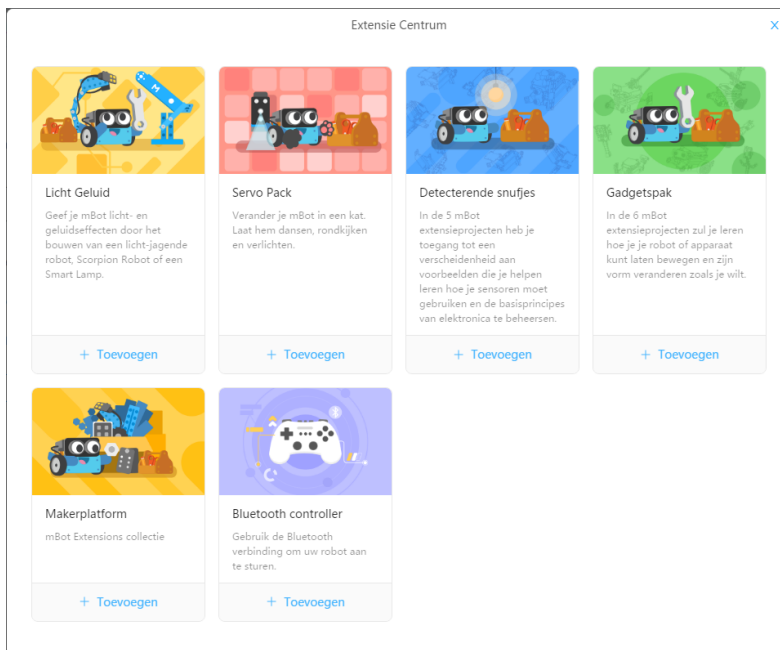


Figure 3: Zoek ‘Makerplatform’ en klik eronder op ‘Toevoegen’



Er zijn genoeg coole uitbreidingen voor mBot!

5.2 7-segment display: eerste getal, opdracht

- Stop een 7-segment display in poort 1

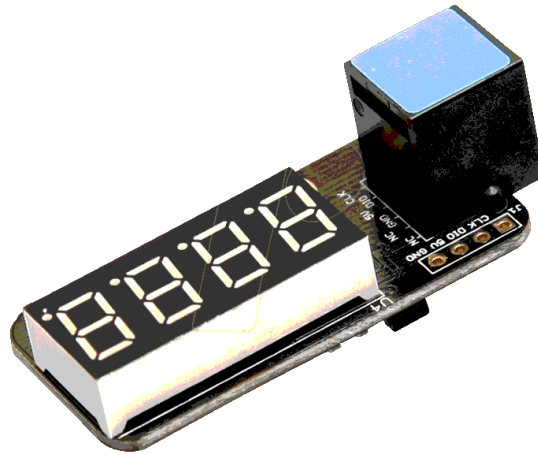


Figure 4: Een 7-segment display

- Upload dit programma:

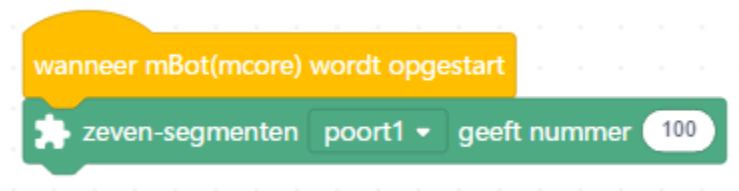


Figure 5:

- Wat zie je op het 7-segment display?



7-segmenten display, oftewel iets dat getallen laat zien met zeven LEDjes

5.2 7-segment display: eerste getal, oplossing

Je ziet 100.0, oftewel, in het Nederlands, ‘honderd komma nul’, oftewel honderd.



Engelsen gebruiken een punt voor kommagetallen!

5.3 7-segment display: tellen, opdracht

Maak een programma:

- Maak een variabele aan met de naam `teller`
- Herhaal eeuwig:
- De 7 segment display laat de waarde van `teller` zien
- Verander `teller` met 1

Hoe lang duurt het ongeveer voor de teller over de duizend komt?



Je mag best een andere naam dan `teller` kiezen



Ik gebruik graag de variabelenaam `ikbencool`



Programmeurs gebruiken graag de variabelenaam `i`

5.4 7-segment display: tellen, oplossing



Figure 6: 5.4 7-segment display: tellen, oplossing

Het duurt ongeveer 5 seconden voor de teller meer is dan duizend.



Het duurt ongeveer 5 seconden voor de teller meer is dan duizend

5.5 7-segment display: tot tien tellen, opdracht

Maak een programma:

- Maak een variabele aan met de naam **teller**
- Herhaal eeuwig:
- De 7 segment display laat de waarde van **teller** zien
- Verander **teller** met 1
- Wacht 1 seconde
- Als teller meer is dan 9, maak teller dan 1



Het is niet erg als je 1.000, 2.000, enzovoorts ziet

5.6 7-segment display: tot tien tellen, oplossing



Figure 7: 7-segment display: tot tien tellen, oplossing

5.7 7-segment display: blok verander_teller, opdracht

Maak een blok:

- Ga naar 'Mijn blokken' en klik op 'Maak een Blok'

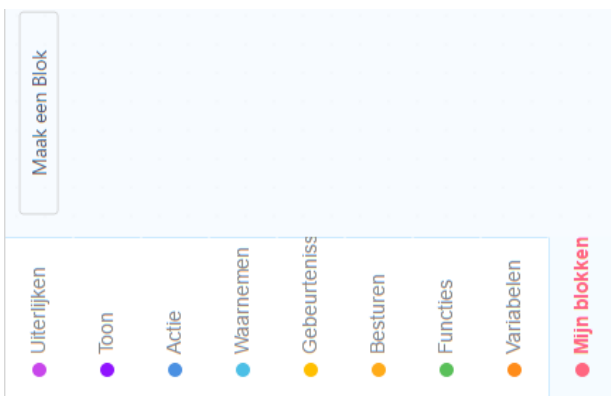


Figure 8: Ga naar 'Mijn blokken' en klik op 'Maak een Blok'

- Noem dit nieuwe blok **verander_teller**
- In het blok **verander_teller** wordt **teller** met 1 verhoogd. Als **teller** meer is dan 9, dan wordt **teller** 1 gemaakt
- Het hoofdprogramma blijft hetzelfde doen:
- De 7 segment display laat de waarde van **teller** zien
- Wacht 1 seconde
- **verander_teller** wordt gedaan

5.8 7-segment display: blok verander_teller, oplossing



Figure 9: 5.8 7-segment display: blok verander_teller, oplossing



Blokken maak je code leesbaarder



Met blokken kan iedereen gemakkelijker zien wat je programma doet

5.9 7-segment display: blok verander_teller, opdracht

- Maak een blok: laat_teller_zien
- In het blok laat_teller_zien wordt de waarde van teller op het 7-segments display getoond
- Het hoofdprogramma blijft eeuwig hetzelfde doen:
- laat_teller_zien wordt gedaan
- Wacht 1 seconde
- verander_teller wordt gedaan

5.10 7-segment display: blok verander_teller, oplossing



Figure 10: 5.10 7-segment display: blok verander_teller, oplossing

5.11 7-segment display: eindopdracht

- Maak een blok: `effe_wachten`. Hierin wacht het programma 1 seconde
- Maak een blok: `doe_eeuwig`. Hierin staat het programma: herhaal `laat_teller_zien`, dan `effe_wachten`, dan `verander_teller`



Figure 11: 5.11 7-segment display: eindopdracht blokken



Effe wachten ... pizza!



(je ouders snappen deze grap misschien wel)

Les 6: RGB

In deze les gaan we met RGB waarden werken.



Rainbow Dash is een pony met regenboog haar Een RGB module

6.1 RGB: Start

- Sluit een RGB module aan op poort 1
- Maak onderstaand programma na:



Figure 12: 6.1 RGB: Start



Het RGB blok staat onder 'Makersplatform'

- Welke kleur denk je dat dit wordt?

6.2 RGB: Kleurencirkel, opdracht

- Bekijk de kleurencirkel hieronder. Om geel te krijgen heb je rood en groen nodig.

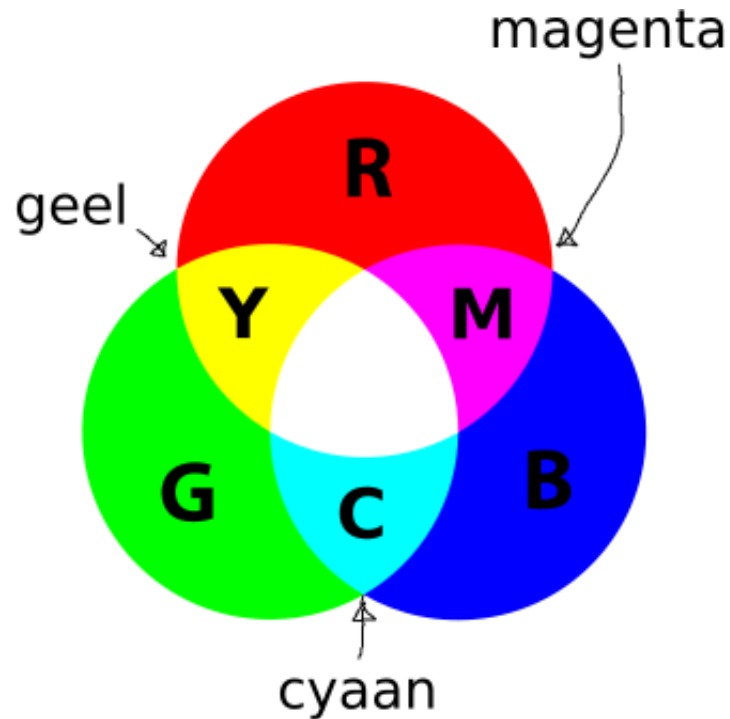


Figure 13: De kleurencirkel

- Verander `laat_zien`, zodat deze de kleur cyaan laat zien



Een rood, groen of blauwwaarde kan niet hoger zijn dan 255



Er zijn meer kleurencirkels! Dit is de kleurencirkel van licht, omdat pixels kleine lampjes zijn

6.3 RGB: Kleurencirkel, oplossing

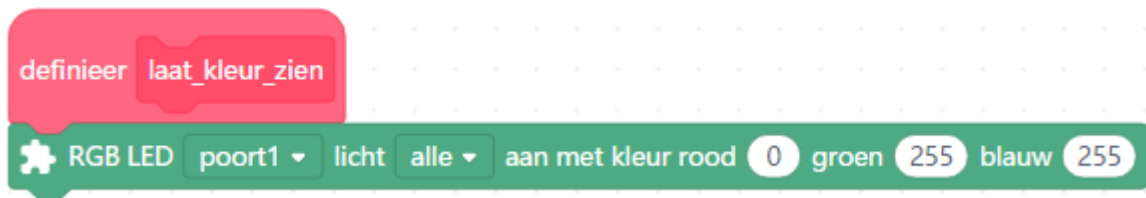


Figure 14: 6.3 RGB: Kleurencirkel, oplossing



Vanaf nu zie je alleen het stuk van het programma dat veranderd is



Ha, meer ruimte voor ons om grapjes te maken!

6.4 RGB: Groenerer, opdracht

- Maak een nieuwe variabele, **groen**
- In **volgende_kleur**, maak **groen** steeds 1 hoger
- In **laat_kleur_zien**, gebruik **groen** als de kleur groen (duh!). Gebruik het getal 0 voor de kleuren rood en blauw

Wat zie je? Het is misschien niet wat je verwacht!



Wat zie je? Het is misschien niet wat je verwacht!

6.5 RGB: Groenerer, oplossing



Figure 15: 6.5 RGB: Groenerer, oplossing



De LED wordt steeds groener, en gaat dan uit en begint overnieuw

6.6 RGB: Gelerer, opdracht

- Maak een nieuwe variabele, **rood**
- Wanneer de mBot wordt opgestart, zet **rood** op 255
- In **laat_kleur_zien**, gebruik **rood** als de kleur rood (duh!). Gebruik nog steeds de variabele **groen** voor de kleur groen. Gebruik nog steeds het getal 0 voor de kleur blauw

Wat zie je? Het is misschien niet wat je verwacht!



Wat zie je? Het is misschien niet wat je verwacht!

6.7 RGB: Gelerer, oplossing

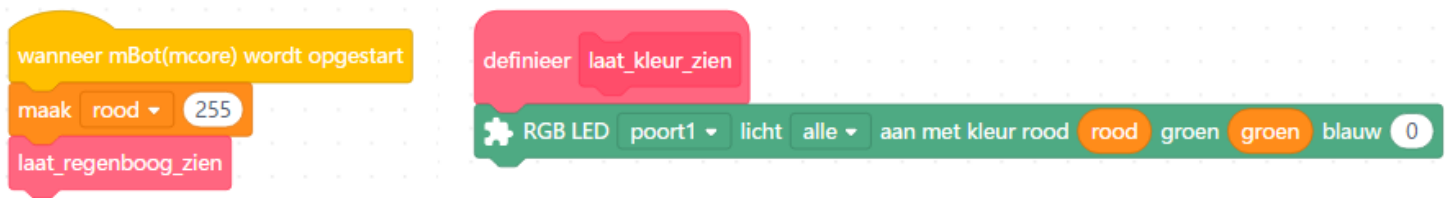


Figure 16: 6.7 RGB: Gelerer, oplossing



De LED gaat van rood naar geel, en begint dan overnieuw

6.8 RGB: RGB display, opdracht

- Maak een derde variabele: **blauw**. Deze **variabele** doet nog niks
- Verander **laat_kleur_zien**:
- De RGB LED licht 1 laat zien: roodwaarde **rood**, groenwaarde 0 en blauwwaarde 0
- De RGB LED licht 2 laat zien: roodwaarde 0, groenwaarde **groen** en blauwwaarde 0
- De RGB LED licht 3 laat zien: roodwaarde 0, groenwaarde 0 en blauwwaarde **blauw**
- De RGB LED licht 4 laat zien: roodwaarde **rood**, groenwaarde **groen** en blauwwaarde **blauw**

Wat zie je? Het is misschien niet wat je verwacht!



Wat zie je? Het is misschien niet wat je verwacht!

6.9 RGB: RGB display, oplossing



Figure 17: 6.9 RGB: RGB display, oplossing



Een LED toont de rood, de tweede groen, de derde blauw, de vierde allemaal samen



Vindt je de LEDs te fel? Gebruik dan het programma hieronder:

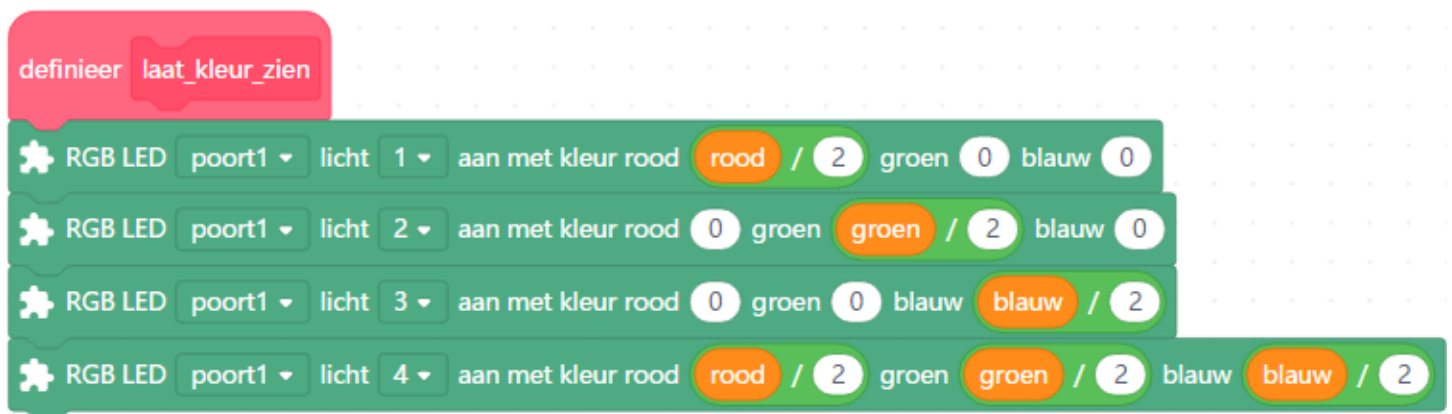


Figure 18: 6.9 RGB: RGB display, oplossing met minder felle LEDs



De / 2 lees je als: 'gedeeld door twee'.

6.10 RGB: regenboog stap 1, opdracht

- Maak een vierde variabele: `stap`.
- Als de mBot wordt opgestart, zet `stap` op 1
- In `volgende_kleur`: als `stap` gelijk is aan 1, doe dan twee dingen:
 - verander `groen` met 1
 - als `groen` gelijk is aan 255, zet `stap` op 2
- Maak een nieuw functieblok: `effe_wachten`. Hierin wordt 0,1 seconde gewacht
- In `laat_regenboog_zien`, herhaal eeuwig `laat_kleur_zien`, `volgende_kleur` en nu ook `effe_wachten`

Wat zie je? Het is misschien niet wat je verwacht!



Wat zie je? Het is misschien niet wat je verwacht!



Effe wachten ... nog effe wachten ... pizza!



Onze programma's zijn groot, dus stoer!

6.11 RGB: regenboog stap 1, oplossing



Figure 19: 6.11 RGB: regenboog stap 1, oplossing



De eerste LED brandt rood en blijft rood



De tweede LED brandt zwart en wordt groen



De vierde LED brandt rood en wordt geel

6.12 RGB: regenboog stap 2, opdracht

- In **volgende_kleur**: als **stap** gelijk is aan 2, doe dan twee dingen:
 - verander **rood** met -1
 - als **rood** gelijk is aan 0, zet **stap** op 3
- In **effe_wachten**: laat 0,03 seconden wachten, inplaats van 0,1

Wat zie je? Het is misschien niet wat je verwacht!



Wat zie je? Het is misschien niet wat je verwacht!

6.13 RGB: regenboog stap 2, oplossing



Figure 20: 6.13 RGB: regenboog stap 2, oplossing



De eerste LED brandt een tijd rood en gaat dan uit



De tweede LED brandt zwart en wordt groen en blijft groen



De vierde LED brandt rood, wordt geel, dan blijft dan groen

6.14 RGB: regenboog stap 3, opdracht

- Maak drie functieblokken: `volgende_kleur_1`, `volgende_kleur_2` en `volgende_kleur_3`
- In `volgende_kleur`:
 - als `stap` gelijk is aan 1, doe dan `volgende_kleur_1`
 - als `stap` gelijk is aan 2, doe dan `volgende_kleur_2`
 - als `stap` gelijk is aan 3, doe dan `volgende_kleur_3`
- Verplaats de volgende code naar `volgende_kleur_1`:
 - verander `groen` met 1
 - als `groen` gelijk is aan 255, zet `stap` op 2
- Verplaats de volgende code naar `volgende_kleur_2`:
 - verander `rood` met -1
 - als `rood` gelijk is aan 0, zet `stap` op 3
- Doe in `volgende_kleur_3`:
 - verander `blauw` met 1
 - als `blauw` gelijk is aan 255, zet `stap` op 4

Wat zie je? Het is misschien niet wat je verwacht!



Wat zie je? Het is misschien niet wat je verwacht!

6.15 RGB: regenboog stap 3, oplossing



Figure 21: 6.15 RGB: regenboog stap 3, oplossing



De eerste LED brandt een tijd rood en gaat dan uit



De tweede LED brandt zwart en wordt groen en blijft groen



De derde LED brandt zwart en wordt blauw en blijft blauw



De vierde LED brandt rood, wordt geel, dan groen, dan cyaan

6.16 RGB: eindopdracht

- Maak drie nieuwe functieblokken: `volgende_kleur_4`, `volgende_kleur_5` en `volgende_kleur_6`
- Voeg toe aan `volgende_kleur`:
 - als `stap` gelijk is aan 4, doe dan `volgende_kleur_4`
 - als `stap` gelijk is aan 5, doe dan `volgende_kleur_5`
 - als `stap` gelijk is aan 6, doe dan `volgende_kleur_6`
- Doe in `volgende_kleur_4`:
 - verander `groen` met -1
 - als `groen` gelijk is aan 0, zet `stap` op 5
- Doe in `volgende_kleur_5`:
 - verander `rood` met 1
 - als `rood` gelijk is aan 255, zet `stap` op 6
- Doe in `volgende_kleur_6`:
 - verander `blauw` met -1
 - als `blauw` gelijk is aan 0, zet `stap` op 1



Hieronder staat in een plaatje hoe een regenboog werkt

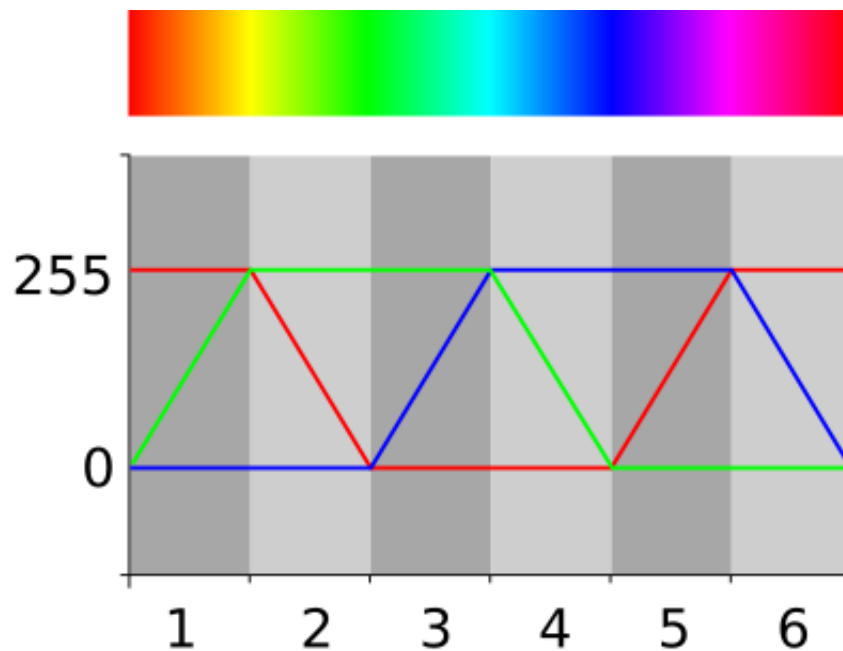


Figure 22: 6.16 RGB: eindopdracht

Les 7: Potmeter

In deze les gaan we met een potmeter werken.

7.1 Potmeter: Start, opdracht

- Sluit een potentiometer ('potmeter') module aan op poort 4
- Sluit een 7 segment display aan op poort 1

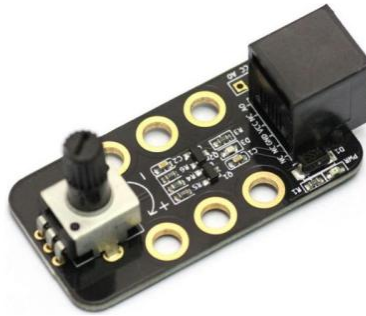


Figure 23: Een potentiometer ('potmeter') module



'potmeter' is een afkorting van 'potentiometer'

Schrijf het volgende programma:

- Als de mBot wordt opgestart, doe dan eeuwig `lees_potmeter`, `laat_zien` en `effe_wachten`
- Maak een variabele `waarde`
- Maak een blok `lees_potmeter`, waarin `waarde` de waarde krijgt van de potentiometer
- Maak een blok `laat_zien`, waarin het 7-segmenten display de waarde van `waarde` laat zien
- Maak een blok `effe_wachten`, waarin 0,1 seconde gewacht wordt

Upload het programma en draai aan de potmeter. Welke getallen laat de 7-segmenten display zien?

7.2 Potmeter: Start, oplossing

De 7-segmenten laat getallen tussen nul en 984 zien.



Figure 24: 7.2 Potmeter: Start, oplossing

7.3 Potmeter: Error, opdracht

Een LEDje heeft een RGB waarde van ten hoogste 255. We laten de mBot een error geven als **waarde** boven de 255 komt.

- In het blok **laat_zien**, voeg toe dat als **waarde** groter is dan 255, dat dan het blok **error** wordt aangeroepen
- Maak een blok **error**, deze zet de rechter LED op rood, doet dan **effe_wachten**, en zet dan de rechter LED weer op zwart



Draai aan de potmeter. Wat zie je gebeuren?

7.4 Potmeter: Error, oplossing

Het rechter rode LEDje gaat knipperen als de 7-segment display boven de 255 komt.

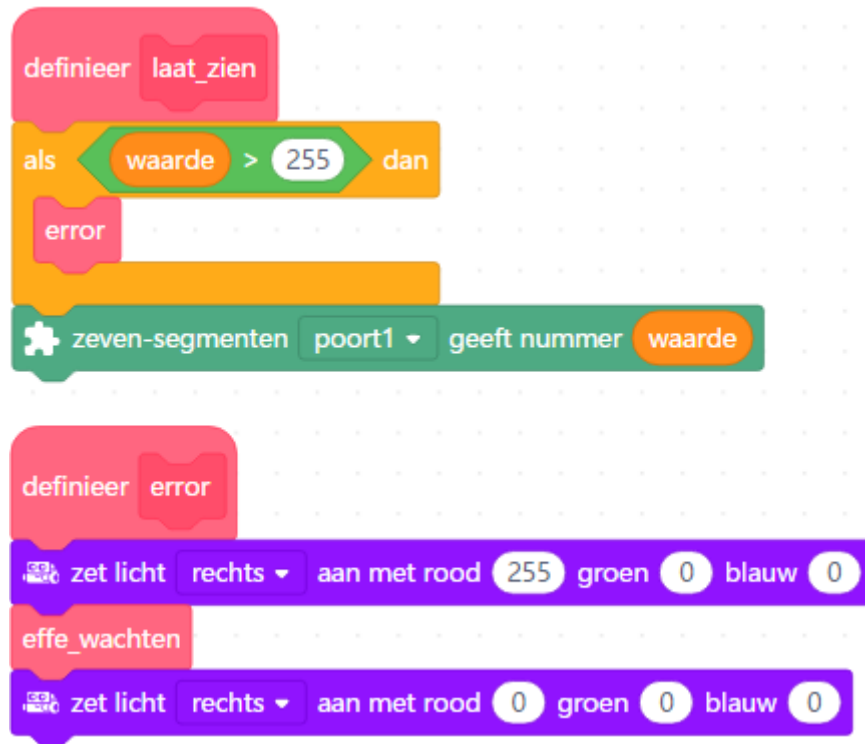


Figure 25: 7.4 Potmeter: Error, oplossing

7.5 Potmeter: Delen, opdracht

In `lees_potmeter`, wordt nu `waarde` gelijk gemaakt aan de waarde van de potmeter. De hoogste waarde van de potmeter is te hoog voor ons:

Verander in het programma:

- In `lees_potmeter`, zet `waarde` gelijk aan de waarde van de potmeter gedeeld door 4

Upload je code en kijk of er geen error meer is. Wat is de hoogste waarde die nu op de 7-segment display komt?



Het teken / lees je als 'gedeeld door'

7.6 Potmeter: Delen, oplossing

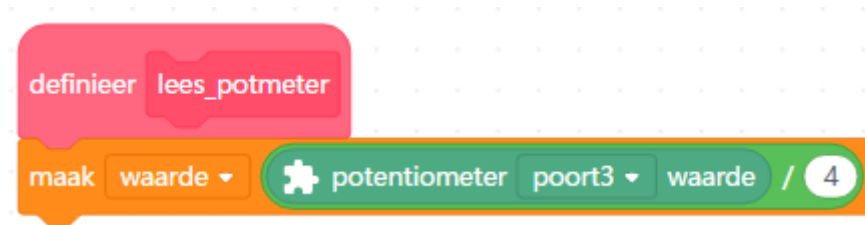


Figure 26: 7.6 Potmeter: Delen, oplossing



Huh? Was dat alles?



De '/' herken je misschien van breuken: $1/2$ is een half!

7.7 Potmeter: Blauwerer, opdracht

Maak een programma dat:

- In `laat_zien` de `waarde` laat zien als de blauwwaarde van de linker LED



Leuk! `laat_zien` laat meerdere dingen `waarde` zien!

7.8 Potmeter: Blauwerer, oplossing



Figure 27: 7.8 Potmeter: Blauwerer, oplossing



Op de potmeter zit een lampje dat precies hetzelfde doet!

7.9 Potmeter: eindopdracht met afstandssensor

Maak een programma dat:

- Sluit de ultrasone afstandssensor aan.



Figure 28: Een ultrasone afstandssensor

- Verander de naam `lees_potmeter` naar `lees_afstand`
- In `lees_afstand` maak `waarde` 25 keer de waarde van de ultrasone sensor
- In `lees_afstand`, als `waarde` meer is dan 255, zet `waarde` dan op 255



Het teken `*` lees je als 'keer'/'maal'/'vermenigvuldigd met'