

## PROGRAMMER

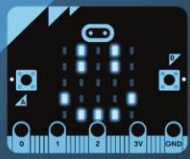
Leer programmeren

**Leer met de Programmer kaarten de principes van programmeren en word een kei in computational thinking**



***Kaarten voor een verdere kennismaking met de micro:bit en belangrijke programmeerprincipes***

***Word een kei in computational thinking***



## PROGRAMMER

### Overzicht kaarten

**Leer met de Programmer kaarten de principes van programmeren en word een kei in computational thinking**

#### **1 Variabelen en rekenen**

- 1a *Secondeteller* - Maak je eigen variabele om secondes te tellen.
- 1b *Rekenmachine* - Maak sommen met de micro:bit.

#### **2 Debuggen en algoritmes**

- 2a *Foutje* - Ontdek waarom een programma het niet doet.
- 2b *Plot er op los* - Kan jij het looplicht repareren?

#### **3 Lussen**

- 3a *Multitasken* - Dankzij lussen lijkt het alsof de micro:bit twee dingen tegelijk doet.
- 3b *Knippen (maar niet te vaak)* - Met lussen met een ingebouwde teller, tel je nooit teveel.
- 3c *Playlist* - Speel de tonen uit jouw lijst.

#### **4 Voorwaarden**

- 4a *Toonladder* - Bij deze toonladder beweegt de led mee met de muziek.
- 4b *EN en/of OF spelletje* - Bij dit spel verplaats je de led met de knoppen A en B.
- 4c *Wie drukte als eerste?* - De micro:bit als scheidsrechter.

#### **5 Meet het lichtniveau**

- 5a *Plot staafdiagram* - Zo maak je een staafdiagram.

#### **6 Versnelling en beweging**

- 6a *X-moves* - Waarheen beweegt de micro:bit?
- 6b *Waterpas* - Gebruik de micro:bit als waterpas.
- 6c *micro:bit kompas* - Dankzij de micro:bit nooit meer de weg kwijt.

#### **7 Radio**

- 7a *Radioled* - Laat twee micro:bits draadloos met elkaar communiceren.
- 7b *Speurneus* - Gebruik de radiofunctie van de micro:bit om een andere micro:bit (de zender) op te sporen.

**Lees verder aan de andere kant**



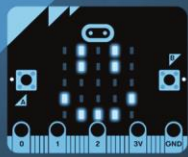
## **8 Functies**

**8a Van links naar rechts** - Bespaar programmablokjes met functies.

## **9 Pins**

**9a Krijgt de plant genoeg water?** - Meten via de pinnen van de micro:bit.

**9b Schuifmuziek** - Maak muziek met een potloodstreep.



## PROGRAMMER

1a

### Variabelen en rekenen

A B



#### Variabelen

In bijna ieder programma worden variabelen gebruikt. Een variabele is een stukje van het computergeheugen dat je zelf een naam kunt geven.

Vergelijk dit met een muur met fotolijstjes. Onder ieder fotolijstje hangt een kaartje waarop staat wat de persoon of de foto voor functie heeft. Bijvoorbeeld *Directeur*, *Conciërge* en *Mentor*.

Als er een nieuwe directeur komt dan wordt de foto in het lijstje *Directeur* vervangen. De naam van het lijstje blijft hetzelfde.

Zo werkt het ook met variabelen in een computerprogramma.

#### Secondeteller

Maak je eigen variabele om secondes te tellen

#### Bouw het programma

Maak met de knop

Maak een variabele...

een nieuwe variabele met de naam *teller* en gebruik deze in het blok *bij opstarten*.



Bouw een teller met deze blokken. Tussen het tonen van ieder getal moet één seconde worden gewacht.



Je kan het programma opnieuw starten door op de resetknop van de micro:bit te drukken



## Ik snap het

Ik kan dit programma zo aanpassen dat het begint te tellen bij 2 en daarna alleen even getallen laat zien (2, 4, 6 etc.)

### Probeer dit ook eens

Verander de naam van de variable *teller*. Dit doe je door in een blokje met de variabele op het ▼ te klikken en daarna te kiezen voor *Naam van variabele wijzigen...*

teller ▼

✓ teller

Naam van variabele wijzigen...

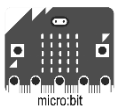
Verwijder de variabele "teller"



## Ik snap het

Ik kan vertellen wat er met de inhoud van variabelen gebeurt als je de naam wijzigt.

### Dit heb je nodig

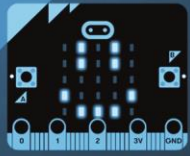


+



+





## PROGRAMMER

1b

# Variabelen en rekenen



## Rekenmachine

Maak sommen met de micro:bit

Dankzij variabelen kan je van je micro:bit een rekenmachine maken. Met dit programma kan je twee getallen bij elkaar optellen.

Het eerste getal komt in de variabele *getal\_a* en het tweede in *getal\_b*.

Door op *knop A* te drukken wordt *getal\_a* telkens met 1 verhoogd en met *knop B* verhoog je de variabele *getal\_b*. Door tegelijk op *knop A* en *knop B* te drukken zie je de uitkomst van de optelling. Het programma zorgt er ook voor dat het hoogste getal dat je kan invoeren 9 is.

### Bouw het programma

Maak als eerste de variabelen *getal\_a* en *getal\_b*. en bouw dan het eerste deel van de rekenmachine.

```
wanneer knop A wordt ingedrukt
  verander getal_a met 1
  als getal_a = 10 dan
    stel getal_a in op 0
  toon nummer getal_a
```

De code voor *knop B* is bijna hetzelfde. Daarom is het handiger om de code die je net hebt gemaakt te dupliceren en daarna aan te passen.

Klik met je rechtermuisknop op het blokje *wanneer knop A wordt ingedrukt* en kies *Dupliceren*.

- Dupliceren
- Reactie toevoegen
- Delete Blocks
- Help



Verander de kopie zodat deze er zo uitziet:

```
wanneer knop B wordt ingedrukt
  verander getal_b met 1
  als < getal_b = 10 > dan
    stel getal_b in op 0
  +
  toon nummer getal_b
```

Als laatste bouw je de code die wordt uitgevoerd als je tegelijk op knop A en knop B drukt.

```
wanneer knop A + B wordt ingedrukt
  toon nummer getal_a + getal_b
```



### Ik snap het 1

Toen je de code van knop A dupliceerde was de kopie vaag. Waarom?

```
wanneer knop A wordt ingedrukt
  verander getal_a met 1
  als < getal_a = 10 > dan
    stel getal_a in op 0
  +
  toon nummer getal_a
```



### Ik snap het 2

In plaats van het +-teken kan ik ook andere operatoren (-, x en ÷) gebruiken om getallen op af te trekken, te vermenigvuldigen en te delen.

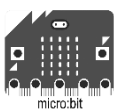


### Ik snap het 3

Ik kan uitleggen wat dit deel van het programma doet..

```
als < getal_a = 10 > dan
  stel getal_a in op 0
+
```

## Dit heb je nodig



micro:bit

+



USB-kabel

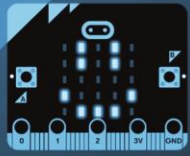
+



Laptop

[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER

2a

### Debuggen en algoritmes

A B



#### Debuggen

Vaak doet een programma niet precies wat jij had bedoeld. Dit kan omdat je:

- een code hebt ingevoerd die niet kan, dit heet een syntax-fout of een error,
- of omdat er een fout zit in het algoritme dat je hebt bedacht.



#### Algoritme

Een algoritme is een stap-voor-stap beschrijving om van een beginsituatie naar een doel te komen.

Voorbeelden van algoritmes zijn:

- Het recept van een appeltaart
- De bouwbeschrijving van een Ikea-kast

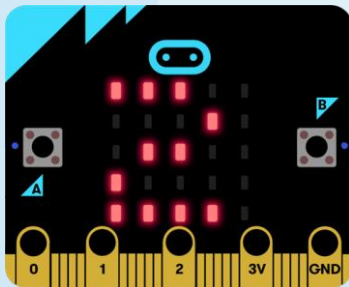
In algoritmes van computerprogramma's kom je onder andere herhalingen en voorwaarden tegen.

### Foutje

Ontdek waarom een programma het niet doet

#### Bouw het programma

Het is de bedoeling dat je dit ziet:

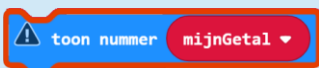


bij opstarten


stel mijnGetal in op "2"

toon nummer mijnGetal

Er verschijnt een rode rand en een waarschuwingsteken:





Je kan op het  klikken om meer te lezen over de waarschuwing.

Als je het programma in de simulator probeert te starten zie je dit:

Oeps, we konden dit project niet uitvoeren. Controleer uw code op fouten.



Hiermee kan je een tekst opgeven. Je vindt het onder *Geavanceerd* in de groep *Tekst*:

Geavanceerd

Tekst



### **Ik kan uitleggen ...**

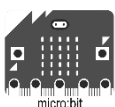
... of dit een fout is in het algoritme of dat het een syntaxfout is.



### **Ik snap het ...**

... want ik kan het programma op twee manieren aanpassen zodat er wel een 2 op het display komt.

### **Dit heb je nodig**



micro:bit

+

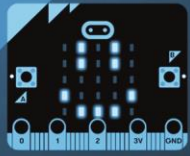


USB-kabel

+



Laptop



# PROGRAMMER

2b

## Debuggen en algoritmes

A B

### Plot er op los

Kan jij het looplicht repareren?

### Bouw het programma

```
teken x 0 y 0
```

Hiermee kan je aangeven welke led moet gaan branden. De 25 leds van het display hebben allemaal een eigen coördinaat.

(x,y)	⇒ X ⇒				
	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
↕	(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
↕	(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
	(0,4)	(1,4)	(2,4)	(3,4)	(4,4)

Iemand wilde met dit blokje een looplicht maken waarbij de vijf ledjes op de bovenste rij stuk voor stuk aan en uit gaan. Het programma werkt echter niet. Kan jij helpen?

Bouw het programma en kijk op de micro:bit of in de simulator wat het programma (niet) doet. Bedenk daarna een oplossing.



```

bij opstarten
  stel teller in op 0

de hele tijd
  stel teller in op teller + 1
  teken x teller y 0
  pauzeer (ms) 100
  Wis scherm
  als teller = 5 dan
    stel teller in op 0
  +

```

## TIP



Klik in de simulator op de slowmotion knop om beter te kunnen zien wat er precies gebeurt



### Ik kan vertellen ...

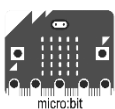
... wat het 'foute' programma wel doet en wat niet.



### Ik snap het

Ik heb het programma zo aangepast dat alle leds in de bovenste rij wel meedoen.

## Dit heb je nodig



+

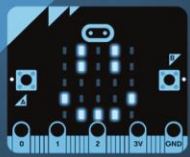


+



[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER

3a

### Lussen

A B C



#### Herhalen

Een lus (Engels: 'loop') in een programma zorgt ervoor dat iets telkens wordt herhaald.

Lussen komen in bijna ieder computerprogramma voor.

#### Multitasken

Dankzij lussen lijkt het alsof de micro:bit twee dingen tegelijk doet.

#### Bouw het programma

Met *knop A* laat je de led aan de linkerkant snel knipperen. Tegelijk kan je met *knop B* de led rechts langzaam laten knipperen.

```
de hele tijd
terwijl knop A wordt ingedrukt
doe
  teken x 0 y 2
  pauzeer (ms) 100
  wis x 0 y 2
  pauzeer (ms) 100
```

```
de hele tijd
terwijl knop B wordt ingedrukt
doe
  teken x 4 y 2
  pauzeer (ms) 500
  wis x 4 y 2
  pauzeer (ms) 500
```

Let op: Je kan dit programma niet testen in de simulator.

de hele tijd



Dit is de bekendste en meest gebruikte lus in micro:bit programma's. Je kan deze lus zo vaak gebruiken als je wilt en de micro:bit zo verschillende dingen tegelijk laten doen.

knop A wordt ingedrukt



Hiermee kan je bekijken of één van de toetsen op dit moment wordt ingedrukt.

Het blok is *waar (true)* als de knop is ingedrukt en *onwaar (false)* als de knop niet is ingedrukt.



### Ik snap waarom...

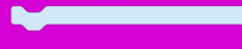
... dit programma niet (helemaal) werkt in de simulator.



### Ik snap het verschil ...

... tussen deze blokken:

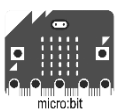
wanneer knop A wordt ingedrukt



knop A wordt ingedrukt



### Dit heb je nodig



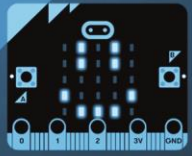
+



+



[www.micro-bit.nl](http://www.micro-bit.nl)



## PROGRAMMER

3b

### Lussen

A B C

#### Knipperen (maar niet te vaak)

Met lussen met een ingebouwde teller, tel je nooit teveel.

#### Bouw het programma

Als je de micro:bit naar links kantelt dan knippert de middelste led precies vier keer.

Als je naar rechts kantelt dan loopt de led van links naar rechts.

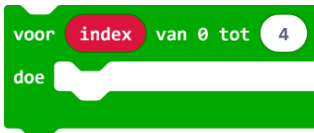
```
bij naar links kantelen ▼  
  4 keer herhalen  
doe  
  teken x 2 y 2  
  pauzeer (ms) 100 ▼  
  wis x 2 y 2  
  pauzeer (ms) 100 ▼
```

```
bij naar rechts kantelen ▼  
voor index van 0 tot 4  
doe  
  teken x index ▼ y 2  
  pauzeer (ms) 100 ▼  
  wis x index ▼ y 2  
  pauzeer (ms) 100 ▼
```



herhaald moeten worden.

Deze lus heeft een ingebouwde teller waarmee je kan aangeven hoe vaak de opdrachten in de lus



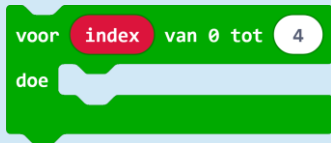
herhaald moeten worden. Ook deze lus heeft een ingebouwde teller. In een variabele (*index*) wordt de stand van de teller voor je bijgehouden. Op deze manier kan je de tellerstand in de lus gebruiken.

Ook deze lus heeft een ingebouwde teller. In een variabele (*index*) wordt de stand van de teller voor je bijgehouden. Op deze manier kan je de tellerstand in de lus gebruiken.

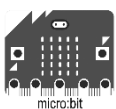


### Ik snap het

Wordt met het woordje *tot* in dit blok *tot of tot en met* bedoeld?



### Dit heb je nodig



micro:bit

+



USB-kabel

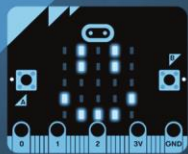
+



Laptop

[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER

3c

### Lussen

A B C

#### Playlist

Speel de tonen uit jouw lijst.

#### Bouw het programma

Als je op knop A drukt dan worden de tonen uit jouw lijst afgespeeld.

Maak eerst twee variabelen: *mijnLijst* en *mijnInhoud*.

In het eerste deel van het programma vul je de lijst (het array) met de tonen die je wilt afspelen.

bij opstarten

stel mijnLijst in op array of 250 500 750 - +

In het tweede deel van het programma staat de lus die de tonen uit de lijst ophaalt, opslaat in de variabele *mijnInhoud* en daarna afspeelt.

wanneer knop A wordt ingedrukt

voor element mijnInhoud van mijnLijst  
voer uit speel toon mijnInhoud voor 1 beat

array of 250 - +

Een lijst wordt door programmeurs een

*array* genoemd. Een array moet je net als een variabele een naam geven.

Om elementen aan het array toe te voegen klik je op het +.

250

De blokjes met de waarden (getallen) haal je uit de groep *Wiskunde (Math)*.

voor element waarde van list  
voer uit

Deze lus haalt iedere keer een element uit een lijst

(array) en zet deze in de variabele *waarde*.



#### Ik snap het

Ik kan het programma zo aanpassen dat de lijst met tonen 4 keer wordt afgespeeld als ik op knop A druk.





## Voor de specialisten

Het vullen van het array is nogal bewerkelijk. Het zou een stuk makkelijker zijn als je de waardes direct kon typen.

Klik bovenin de editor maar eens op de knop `{}` JavaScript:

`{}` JavaScript

Je ziet nu een tekstversie van het programma:

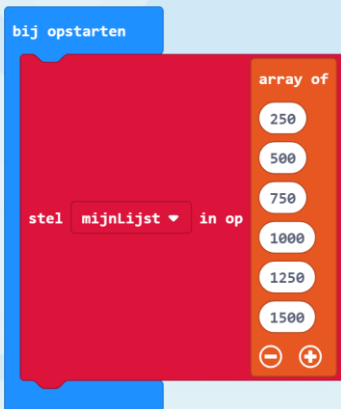
```
1 let mijnLijst: number[] = []
2 input.onButtonPressed(Button.A, () => {
3   for (let mijnInhoud of mijnLijst) {
4     music.playTone(mijnInhoud, music.beat(BeatFraction.Half))
5   }
6 })
7 mijnLijst = [250, 500, 750]
8
```

Op regel 7 kan je extra waardes typen.

```
7 mijnLijst = [250, 500, 750, 1000, 1250, 1500]
```

Klik nu op de knop *Blokken*:

Blokken



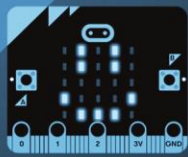
Je bent nu weer in de blockeditor en de extra waardes (elementen) zijn toegevoegd aan je lijst.

## Dit heb je nodig



[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER

4a

### Voorwaarden

A B C



#### Voorwaarden

In een computerprogramma worden keuzes gemaakt op basis van voorwaarden.

Voorwaarden kunnen op allerlei manieren worden geschreven. Lees dit maar eens:

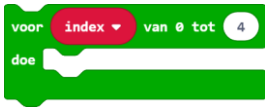
- ALS het regent DAN doe ik mijn regenjas aan
- ALS het NIET droog is DAN doe ik mijn regenjas aan.
- ALS het regent EN ik kan NIET met de auto DAN doe ik mijn regenjas aan.

#### Toonladder

Bij deze toonladder beweegt de led mee met de muziek.

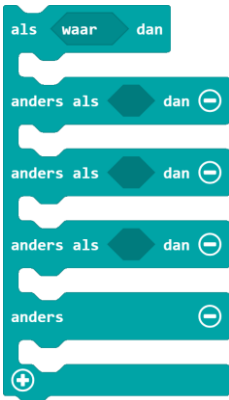
#### Bouw het programma

```
wanneer knop A wordt ingedrukt
  voor index van 0 tot 4
  doe
    teken x index y 0
    als index = 0 dan
      speel toon 150 Hz voor 1/2 beat
    anders als index = 1 dan
      speel toon 300 Hz voor 1/2 beat
    anders als index = 2 dan
      speel toon 600 Hz voor 1/2 beat
    anders als index = 3 dan
      speel toon 1200 Hz voor 1/2 beat
    anders
      speel toon 2400 Hz voor 1/2 beat
  Wis scherm
```



hoe vaak de lus is doorlopen.

Hiermee herhaal je de opdrachten die in deze lus (loop) staan en wordt in een variabele bijgehouden



Door op het **+** te klikken kan je extra voorwaarden toevoegen.

Als aan geen enkele voorwaarde wordt voldaan wordt dan worden de opdrachten bij *anders* uitgevoerd.



### Ik snap het

De variabele *index* telt van 0 t/m 4. Waarom staat dit dan niet in het programma?



### Dit heb je nodig

micro:bit  
USB-kabel  
Laptop

+



Koptelefoon

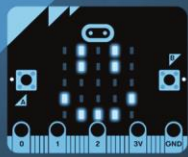
+



Krokodilsnoertje

[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER

4b

### Voorwaarden

A B C

#### EN en/of OF spelletje

Bij dit spel verplaats je de led met de knoppen A en B.

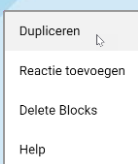
De led gaat knipperen als hij in het midden is OF als hij boven OF onder is.

#### Bouw het programma

```
wanneer knop A wordt ingedrukt
  wis x x y y
  verander x met 1
  als x = 5 dan
    stel x in op 0
  teken x y y
```

Het programma bestaat uit drie delen.

Het eerste deel zorgt ervoor dat de led van links naar rechts beweegt (x-richting):



Dupliceer dit deel door met rechts te klikken op het eerste blok.

Pas de kopie daarna zo aan:

```
wanneer knop B wordt ingedrukt
  wis x x y y
  verander y met 1
  als y = 5 dan
    stel y in op 0
  teken x y y
```

Dit tweede deel zorgt ervoor dat de led verticaal beweegt (y-richting).

Hieronder zie je het laatste deel. Hiermee controleer je of aan de voorwaarden wordt voldaan.

```
de hele tijd
  als x = 2 en y = 2 of y = 0 of y = 1 dan
    wissel x y y
    pauzeer (ms) 100
```

Dit deel is niet makkelijk te lezen(en dat komt niet alleen omdat deze kaart zo smal is). Door het *als ... dan* blokje anders in te delen, wordt de code overzichtelijker:



```
de hele tijd
als <x = 2 en y = 2> dan
  wissel x x y y
  pauzeer (ms) 100
anders als <y = 0 of y = 4> dan
  wissel x x y y
  pauzeer (ms) 100
```



Dit blokje uit de groep *Logisch* controleert of aan twee voorwaarden wordt voldaan.

*Voorwaarde 1* is waar EN *Voorwaarde 2* is waar



Dit blokje controleert of aan één van de voorwaarden wordt voldaan.

*Voorwaarde 1* is waar OF *Voorwaarde 2* is waar



Dit verandert de led op de opgegeven positie. Als de led *aan* is dan gaat hij *uit* en andersom.



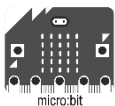
### Ik snap het

De zin *Kim of Jan en Fatima* hebben taart kan je op twee manieren uitleggen. Ik kan het verschil uitleggen tussen:

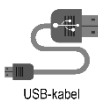
Kim of (Jan en Fatima) hebben taart

(Kim of Jan) en Fatima hebben taart

### Dit heb je nodig



+

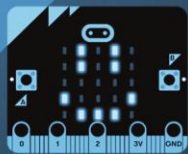


+



[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER



### Voorwaarden



#### Wie drukt als eerste?

De micro:bit als scheidsrechter.

#### Bouw het programma

Met dit programma kan je zien welke knop als eerste werd ingedrukt. Als één van de knoppen is ingedrukt dan blijft de letter van de eerst ingedrukte knop staan.

Bouw dit programma en zet het op de micro:bit.

```
de hele tijd
als knop A wordt ingedrukt dan
  toon tekens "A"
als knop B wordt ingedrukt dan
  toon tekens "B"
```

Dit programma kan NIET worden getest in de simulator.

Onderzoek wat gebeurt als je ...

- ... knop A ingedrukt houdt en daarna op knop B drukt?
- ...knop B ingedrukt houdt en daarna op knop A drukt?

#### Oplossing

Als op een knop wordt gedrukt, moet het programma controleren of de andere knop NIET is ingedrukt.

```
de hele tijd
als knop A wordt ingedrukt dan
  als niet knop B wordt ingedrukt dan
    toon tekens "A"
als knop B wordt ingedrukt dan
  als niet knop A wordt ingedrukt dan
    toon tekens "B"
```



knop A wordt ingedrukt

Hiermee kijk je of een knop is ingedrukt.

wanneer knop A wordt ingedrukt

Bij dit blokje gebeurt er pas iets als de knop wordt losgelaten.

niet

Hiermee keer je een logische waarde (*Waar* of *Onwaar*) om. *Waar* wordt *Onwaar* en andersom.

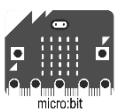


## Ik snap het

Ik kan voorspellen wat er op het display komt te staan als op de knop wordt gedrukt.



## Dit heb je nodig



micro:bit

+



USB-kabel

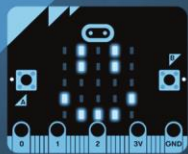
+



Laptop

[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



# PROGRAMMER

## 5a

### Meet het lichtniveau



#### Hoe laat je het zien? (Representatie)

Computers kunnen werken met erg veel gegevens en leveren veel informatie. Programmeurs bedenken allerlei manieren om deze gegevens zo duidelijk mogelijk te laten zien. Dit heet *representatie*.

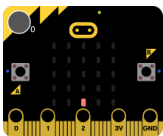
#### Plot staafdiagram

Zo maak je een staafdiagram.

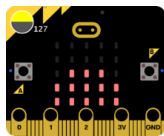


Met dit blok kan je in een staafdiagram (grafiek) een waarde laten zien. Deze waarde staat in de 1<sup>e</sup> regel van het blokje *..of (van)* dat is toegevoegd.

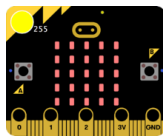
In de 2<sup>e</sup> regel van het blokje *up to* (tot en met) staat de hoogste waarde die je verwacht.



0%



50%



100%

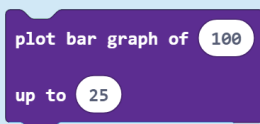
#### Bouw het programma

Als je meer wilt experimenteren met het blok plot staafdiagram dan is dit een leuk programma



#### Ik kan voorspellen ...

... hoe het staafdiagram eruit ziet als ik dit programmeer.







## Meet het lichtniveau

Laat in een staafdiagram zien hoeveel licht op de micro:bit valt.

### Bouw het programma

De 25 lampjes (leds) van de micro:bit kan je niet alleen gebruiken om licht te laten geven, maar ook om te meten

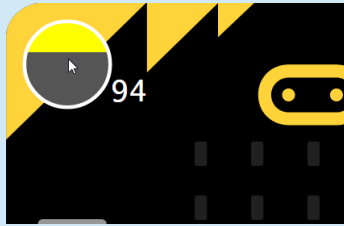
hoeveel licht erop valt. De leds zijn dan een sensor. Je kan dit vergelijken met onze zintuigen.

De micro:bit heeft ook 'zintuigen' voor het waarnemen van versnelling, kompasrichting, rotatie, magnetische kracht, temperatuur en dus ook licht.



### Simulator

Zodra je het blok lichtniveau gebruikt verschijnt links bovenin de simulator een icoontje waarmee je het licht kunt regelen.



**lichtniveau**

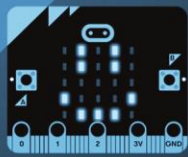
Hiermee meet je via de leds van het display hoeveel licht er op de micro:bit valt. De laagste waarde die dit blokje teruggeeft is 0 en de hoogst mogelijke waarde is 255.

### Dit heb je nodig



[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER

6a

### Versnelling en beweging

A B C

#### Versnelling (acceleration)

Versnelling is een verandering van snelheid. Als je bijvoorbeeld fietst met een snelheid van 10 kilometer per uur en je gaat 15 kilometer per uur fietsen dan moet je versnellen. De versnelling duurt totdat je 15 kilometer per uur fiets.

Als je in een auto zit die heel snel optrekt dan kan je de versnelling voelen doordat je met je rug in de autostoel wordt gedrukt.

Het tegenovergestelde van versnelling is vertraging. De natuurkundige eenheid van versnelling is de *g* (van *g*-kracht). Eén *g* is de versnelling die optreedt als iets valt.

#### X-moves

Waarheen beweegt de micro:bit?

#### Bouw dit programma

de hele tijd

```
plot staafdiagram van versnelling (mg) x v
tot en met 1023
```

Sluit een batterij aan op de micro:bit, download het programma en kopieer het naar de micro:bit.

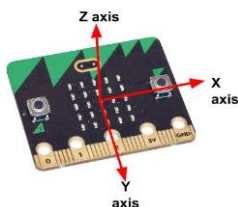
Schuif de micro:bit heen en weer in de X-richting en de Y-richting. Wat zie je op het staafdiagram?

versnelling (mg) x v

De micro:bit heeft een sensor om versnelling te meten. De gegeven waarde is de versnelling in mg ofwel milli-g (dit is één duizendste van één g).

Omdat de micro:bit in verschillende richtingen kan bewegen, kan je ook aangeven in welke richting de versnelling moet worden gemeten.

- x
- y
- z
- strength





## Ik weet waarom ...

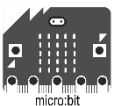
... in dit programma bij *up to* de waarde 1023 is ingevuld.

de hele tijd

plot staafdiagram van versnelling (mg) x ▼

tot en met 1023

## Dit heb je nodig



+



+

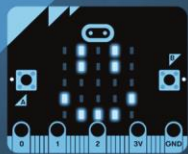


+



[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER

6b

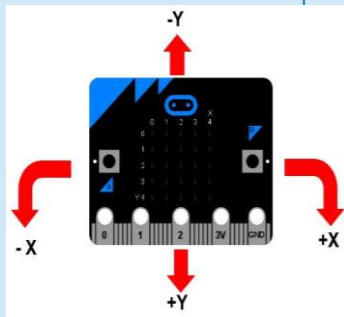
### Versnelling en beweging

A B C

#### Waterpas

Gebruik de micro:bit als waterpas.

De sensor die de versnelling meet, meet tegelijk ook de kanteling. Je kan dit zien als je met het programma van de opdracht hiervoor de micro:bit niet heen en weer beweegt (schuift), maar kantelt.



Als je de micro:bit helemaal vlak houdt dan is de versnelling 0. Kantel je de micro:bit naar rechts dan wordt de X-versnelling positief. Kantel je micro:bit naar links dan wordt de X-versnelling negatief.

#### Bouw dit programma

```
de hele tijd
als absolute waarde van versnelling (mg) x < 100 dan
als absolute waarde van versnelling (mg) y < 100 dan
  toon pictogram [grid icon]
+
anders
  toon pictogram [cross icon]
+
```



absolute waarde van  $e$

De absolute waarde van een getal is altijd positief. Van bijvoorbeeld  $-10$  is de absolute waarde  $10$ . Maar de absolute waarde van  $10$  is ook  $10$ .



### **Ik weet waarom ...**

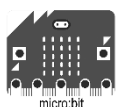
... voor de waterpasstand wordt gecontroleerd of de absolute waarde van de X-versnelling en die van de Y-versnelling kleiner zijn dan  $100$ .



### **Ik weet hoe ...**

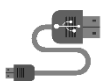
... ik de waterpas nauwkeuriger kan maken.

### **Dit heb je nodig**



micro:bit

+



USB-kabel

+



Laptop

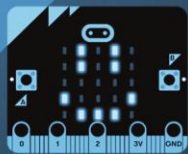
+



Batterij

[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER

6c

### Versnelling en beweging

A B C



#### Wat moet er precies gebeuren? (patronen)

Om een algoritme (oplossing) voor een probleem te kunnen bedenken, besteden programmeurs veel tijd aan het analyseren van wat er precies moet gebeuren. Een tekening of schema helpt dan vaak.

#### micro:bit kompas

Dankzij de micro:bit nooit meer de weg kwijt.

#### Bouw dit programma

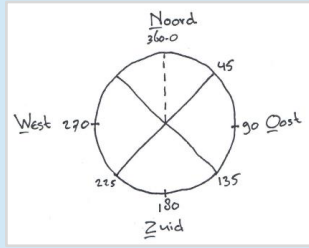
```
de hele tijd
  stel graden in op kompasrichting (°)
  als graden < 45 dan
    toon tekens "N"
  anders als graden > 315 dan
    toon tekens "N"
  anders als graden ≥ 225 dan
    toon tekens "W"
  anders als graden ≥ 135 dan
    toon tekens "Z"
  anders
    toon tekens "0"
```

Sluit een batterij aan op de micro:bit, download het programma en kopieer het naar de micro:bit.

Als het programma op de micro:bit wordt gestart dan zegt de micro:bit *TILT TO FILL SCREEN (BEWEEG OM HET SCHERM TE VULLEN)*. Dit doe je door de micro:bit in verschillende richtingen te kantelen. Als alle ledjes op het display aan zijn dan is het kompas gekalibreerd. Dit betekent dat de micro:bit dan 'weet' waar het Noorden (0°) is.

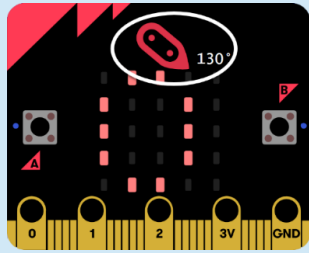


Schemaatje dat de programmeur gebruikte om dit programma te schrijven.



## Simulator

Zodra je het blok *compass heading* gebruikt dan kan je in de simulator aangeven in welke richting de micro:bit wordt gehouden.



### kompasrichting (°)

De aarde heeft een magnetisch veld. Hierdoor kan je met een kompas altijd bepalen waar Noord, Oost, Zuid en West zijn.

In plaats van de namen van de richtingen (Noord, Oost, Zuid en West), kan je een richting ook aangeven in graden. Deze lopen van 0 graden tot 360 graden. Net als bij het opschrijven van een temperatuur kan je het woordje *graden* ook als een kleine *o* schrijven: 90°.

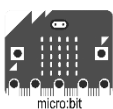


## Ik snap het

Als we de windrichtingen opnoemen dan doen we dit meestal in de volgorde Noord, Oost, Zuid, West (met de klok mee).

Waarom staan ze in het programma in de omgekeerde volgorde (Noord, West, Zuid, Noord)?

## Dit heb je nodig



+



+

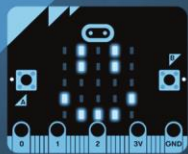


+



[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER

7a

### Radio



#### Abstract denken: Als je het ziet valt het mee.

Een goed programma begint bij een goede beschrijving van wat het programma allemaal moet doen.

Soms is het handiger om taken op te splitsen en een andere keer juist om ze samen te voegen. Een goede programmeur kan abstract denken. Dat betekent dat hij/zij belangrijke dingen van de minder belangrijke dingen kan onderscheiden.

Door abstract te denken worden problemen die heel ingewikkeld leken vaak een stuk eenvoudiger.

#### Radioled

Laat twee micro:bits draadloos met elkaar communiceren.

#### Bouw dit programma

Als je op knop A drukt, gaat de middelste led van de andere micro:bit aan. De micro:bits werken allebei als zender en als ontvanger.

bij opstarten

Radio instellen groep 1

de hele tijd

als knop A wordt ingedrukt dan

Radio verzend nummer 1

anders

Radio verzend nummer 0



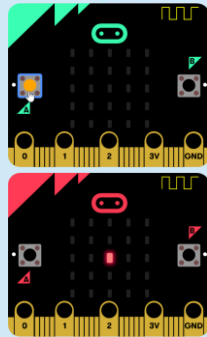


Om het te testen zet je dit programma op twee micro:bits.

## Simulator

Zodra je een radiofunctie gebruikt zie je in de simulator twee micro:bits.

De programmeur heeft bij het maken van het programma telkens moeten bedenken dat er nog een micro:bit is die hetzelfde doet.



Radio verzend nummer 0

Hiermee verstuur je een getal (number) of de inhoud van een variabele.

wanneer de radio ontvangt receivedNumber

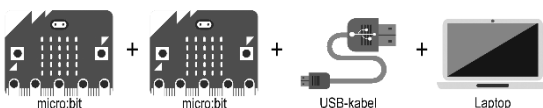
Zodra de micro:bit een getal ontvangt

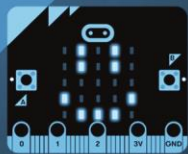
wordt het getal opgeslagen in de variabele *receivedNumber* en worden de opdrachten in dit blok uitgevoerd.

Radio instellen groep 1

Stelt het radiokanaal in waarop de micro:bit uitzendt en ontvangt. Je kan hiervoor de nummers 0 t/m 255 gebruiken. Op deze manier kan je ervoor zorgen dat micro:bits elkaar niet storen.

## Dit heb je nodig





## PROGRAMMER

7b

### Radio



#### Speurneus

Gebruik de radiofunctie van de micro:bit om een andere micro:bit (de zender) op te sporen.

Je hebt hiervoor twee micro:bits nodig.

#### Bouw het programma van de zender

```
bij opstarten
  Radio instellen groep 1
  radio stel uitzendkracht in 1
```

```
de hele tijd
  voor index van 0 tot 9
  doe
    Radio verzend nummer index
    toon nummer index
    pauzeer (ms) 100
```

Het eerste deel van het programma zorgt ervoor dat je zeker weet in welke groep je uitzendt.

Het tweede deel van het programma zorgt ervoor dat er continue een getal wordt uitgezonden.

Download dit programma en kopieer het naar de micro:bit die je als zender gaat gebruiken.

```
radio stel uitzendkracht in 1
```

Stelt het zendvermogen in. Je kan hier 0 t/m 7 invullen. Hoe groter het zendvermogen, hoe groter het zendbereik (tot wel 70 meter!).

#### Bouw het programma voor de ontvanger

```
bij opstarten
  Radio instellen groep 1
```

```
wanneer de radio ontvangt receivedNumber
  plot bar graph of received packet signaalsterkte + 128
  up to 86
```



received packet **signaalsterkte** ▾

Iedere keer als een getal wordt ontvangen, wordt de variabele *signaalsterkte* gevuld met de sterkte van het ontvangen signaal. Dit kan variëren van -128 (zeer zwak) t/m -42 (erg sterk).

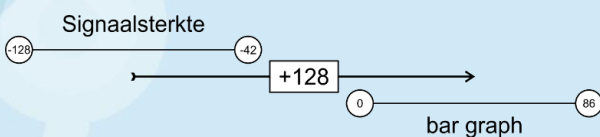
## Berekening om signaalsterkte in staafdiagram weer te geven

In een staafdiagram kan je alleen positieve waarden weergeven. Daarom tellen we 128 op bij de gemeten waarde:

$$-128 + 128 = 0$$

De laagst mogelijke waarde in de bargraph is nu 0. De hoogste waarde is dan:

$$-42 + 128 = 86$$



### Probeer dit

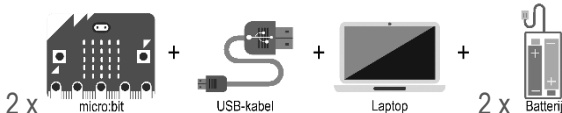
- Laat iemand de zender verstoppen en probeer hem dan met de ontvanger te vinden.
- Doe de zender in een envelop en doe er dan zilverpapier omheen.



### Ik snap waarom ...

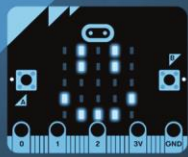
... de micro:bit met de zender eerst in een envelop moet.

### Dit heb je nodig



[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER

8a

### Functies



#### Functies

Soms moet op verschillende plaatsen in je programma hetzelfde worden gedaan. Het is dan handig zijn als je dit deel van het programma maar één keer hoeft te bouwen en telkens kan hergebruiken.

Programmeurs noemen dit een *functie* of *subroutine*.

#### Van links naar rechts

Bespaar programmablokjes met functies.

#### Bouw dit programma

Bij dit programma 'valt' er in de middelste kolom een ledje naar beneden als je de micro:bit naar links of rechts kantelt.

```
bij naar links kantelen v
show leds
voor index van 0 tot 4
doe teken x 2 y index
pauzeer (ms) 100
wis x 2 y index
pauzeer (ms) 100
```

```
bij naar rechts kantelen v
show leds
voor index van 0 tot 4
doe teken x 2 y index
pauzeer (ms) 100
wis x 2 y index
pauzeer (ms) 100
```

In allebei de delen van het project komt dezelfde code voor. We gaan hier een functie (subroutine) van maken zodat de code nog maar één keer in het project voorkomt.

Ga naar de groep *Functies*:

[f\(x\) Functies](#)

En klik hier op de knop *Maak een functie...*:

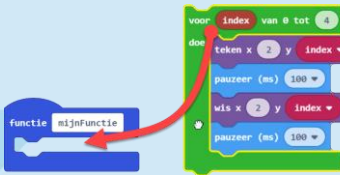
Maak een functie...

Geef de functie een naam en klik op de knop *Ok*

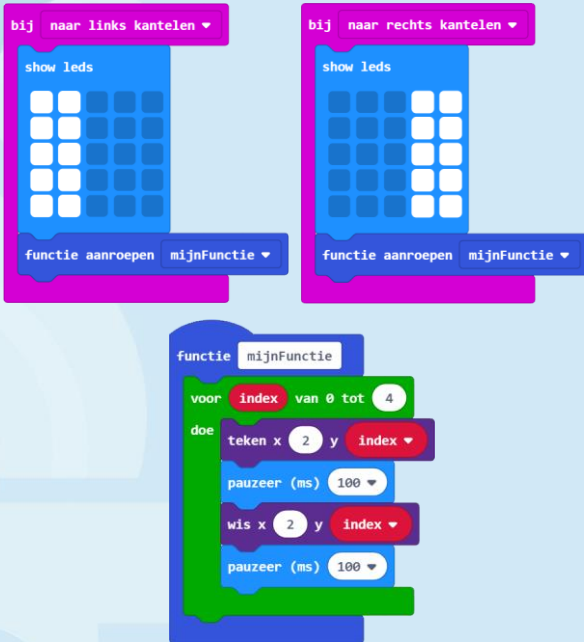
**Nieuwe functienaam:**

Ok ✓
Annuleren ✕

Sleep nu dit deel van de code naar het blokje *functie*.



Pas het programma nu zo aan



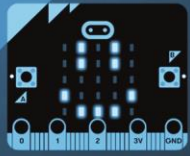
functie aanroepen doeWat

Hiermee roep je een functie aan die je eerder hebt gedefinieerd in het blokje *functies*.

functie doeWat

### Dit heb je nodig





## PROGRAMMER

9a

### Pins



#### Decompositie

Programmeurs zijn niet alleen goed in het in elkaar zetten (*componeren*) van programma's, maar ook in het opdelen van problemen (dat heet *decomponeren*). Door problemen en vragen te decomponeren kan je ze stap voor stap oplossen.

Bij het volgende voorbeeld moest de programmeur een energiezuinige nauwkeurige vochtsensor bouwen.

### Krijgt de plant genoeg water?

Meten via de pinnen van de micro:bit.

lees analoog pin P0 ▾

Hiermee meet je de spanning op pin 0. Het blokje geeft de een waarde tussen 0 en 1023. *Analoog* betekent dat er veel verschillende waarden mogelijk zijn. Het tegengestelde van analoog is digitaal.

Je vindt het blokje in de groep *Pinnen*:

📍 Pinnen

### Bouw de 1<sup>e</sup> versie van het programma

de hele tijd

plot bar graph of lees analoog pin P0 ▾

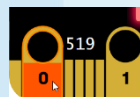
up to 1023

Sluit nu krokodil-snoertjes aan op *Pin 0* en op *Pin 3V*. De andere kant van de snoertjes klem je op twee spijkers aan. Deze spijkers steek je in de aarde van je plant.



### Simulator

Als je het blokje *lees analoog pin* gebruikt dan kan je met de pin op de simulator de waarde veranderen.





## Wat zijn de andere problemen?

De basis van het programma werkt goed, dat probleem is opgelost. Maar de sensor moet ook een nauwkeurigere waarde kunnen meten en zo min mogelijk stroom gebruiken.

## Bouw de 2e versie van het programma

```
de hele tijd
  schrijf analog pin P1 naar 1023
  stel mijnWaarde in op lees analog pin P0
  schrijf analog pin P1 naar 0
  plot bar graph of mijnWaarde
  up to 1023
  pauzeer (ms) 5000
```



In de 1<sup>e</sup> versie van het programma loopt er altijd stroom tussen de pinnen 3V en P0. In plaats van pin 3V gebruikt de programmeur nu pin P1. Op P1 wordt alleen spanning (3 Volt) gezet als er gemeten moet worden. Direct na de meting krijgt P0 de waarde 0 (0 Volt) en loopt er **geen stroom** meer.

```
wanneer knop A wordt ingedrukt
  toon nummer mijnWaarde
```

Het probleem dat er ook **nauwkeurig** gemeten moet worden heeft de programmeur hiermee opgelost. Als je op knop A drukt wordt de waarde van de vochtigheid op het display getoond.

```
schrijf analog pin P1 naar 1023
```

Zet een spanning van 0 tot 3 Volt op Pin 1. Voor 0 Volt vul je 0 in, voor 3 Volt de waarde 1023.

## Dit heb je nodig



+



Batterij

+



Krokodilsnoertje

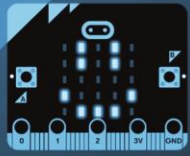
+



Spijkers

[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



## PROGRAMMER

9b

### Pins



### Schuifmuziek

Maak muziek met een potloodstreep.

Met het blokje *lees analoge pin* kan je de elektrische weerstand tussen twee pinnen meten. Bij dit muziekinstrument verander je de toonhoogte door over een zelfgemaakte weerstand te schuiven.

lees analoge pin P1 ▾

### Weerstand

Hoe makkelijk een elektrische stroom door materiaal gaat hangt af van de *elektrische weerstand* van dat materiaal. De weerstand van koper is laag en de weerstand van plastic hoog. Het woord *weerstand* gebruiken we voor de eigenschap, maar ook voor onderdelen die een bepaalde weerstand hebben.

De weerstand voor dit programma maken we met een papier, potlood en een liniaal.

Trek langs een liniaal een lijn van ongeveer 20 cm. Houdt de liniaal op zijn plaats en ga met het potlood nog een paar keer over de lijn heen en weer zodat er veel grafiet op het papier achterblijft.

Verplaats de liniaal een klein beetje naar beneden en maak nog een dikke lijn. Doe dit tot de streep ongeveer 2 mm breed is.

### Bouw dit programma

```
bij opstarten
  stel geluidAan ▾ in op waar ▾

wanneer knop A ▾ wordt ingedrukt
  stel geluidAan ▾ in op niet geluidAan ▾

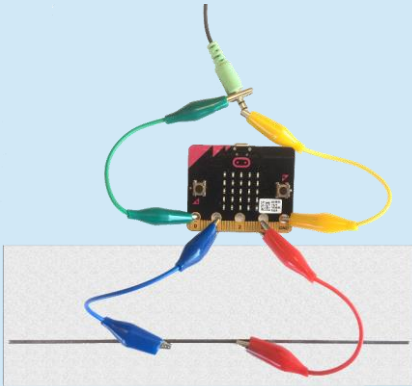
de hele tijd
  stel mijnWeerstand ▾ in op lees analoge pin P1 ▾
  als geluidAan ▾ = waar ▾ dan
    als mijnWeerstand ▾ > 400 dan
      speel toon (Hz) mijnWeerstand ▾
    anders
      rust (ms) 1/16 ▾ beat
```





Sluit nu als eerste de hoofdtelefoon aan op de pinnen *P0* en *GND*.

De snoertjes die naar de weerstand gaan sluit je aan op de pinnen *P1* en *3V*.



Je maakt muziek door de onderste krokodilklemmen op verschillende plaatsen op de potloodstreep (de weerstand) te drukken. Door de afstand tussen te klemmen te veranderen, varieer je de weerstand.



### Ik snap het

De basis van het programma waarmee de programmeur het idee heeft getest zag er zo uit:

de hele tijd

speel toon (Hz)

lees analogoog pin

P1

Uiteindelijk zit er veel meer in het programma. Kan jij uitleggen wat het programma nog meer doet en welke blokjes hiervoor zorgen?

### Dit heb je nodig

micro:bit  
USB-kabel  
Laptop

+ 2x



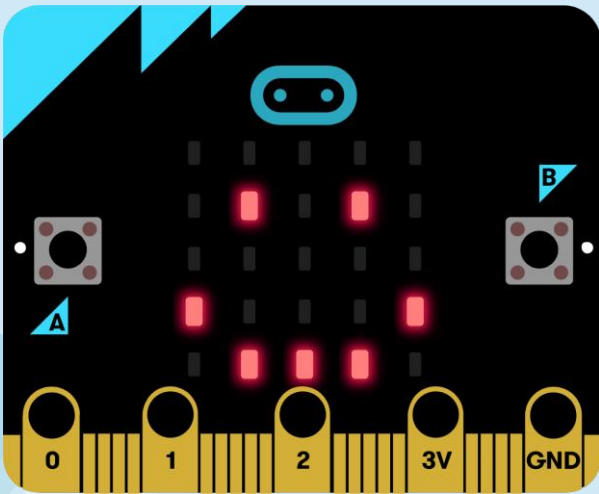
Krokodilsnoertje

+

Potlood  
Linaal  
Papier

[www.micro-bit.nl](http://www.micro-bit.nl)

Versie 1.1



Deze micro:bit PROGRAMMER  
kaartenserie is tot stand gekomen in  
een samenwerking tussen Stichting  
DevLab Academy en CodeKids.nl

DEVLAB | ACADEMY

CODEKIDS

Ga voor meer informatie over micro:bit naar de  
website van *micro:bit community NL*

[www.micro-bit.nl](http://www.micro-bit.nl)



© 2019