

< HELLO, WORLD! />



SCIENCEMODULE OVER PROGRAMMEREN  
MET DE MICRO:BIT

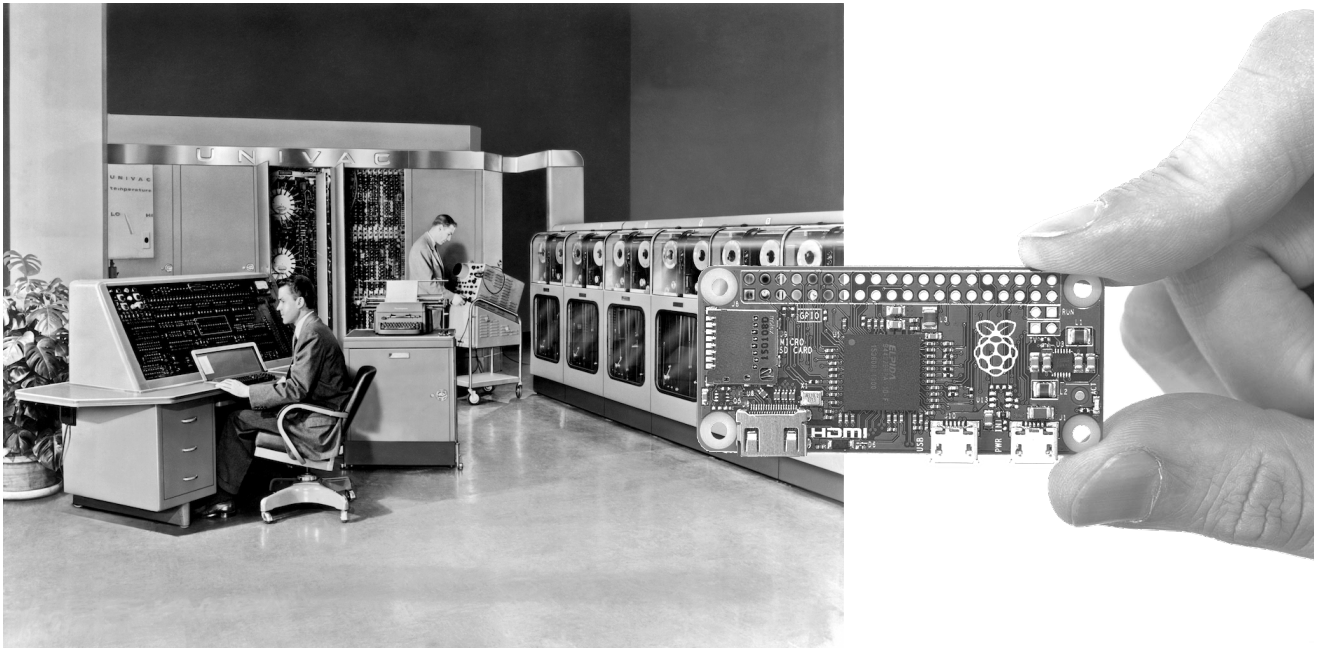
NAAM:

NUMMER:

(Les 4)

# LES 1 - INPUT & OUTPUT

Tegenwoordig kunnen wij bijna niet meer overleven zonder computer of telefoon, laat staan wifi. Robots nemen steeds meer werk van ons over en apparaten worden steeds slimmer. Er bestaat zelfs al een koelkast die online boodschappen bestelt zodra deze leeg is. Deze ontwikkeling is ook goed te zien aan de computer. De eerste computers waren zo groot dat ze niet eens in een klaslokaal pasten, terwijl er tegenwoordig computers zijn die kleiner zijn dan een post-it!



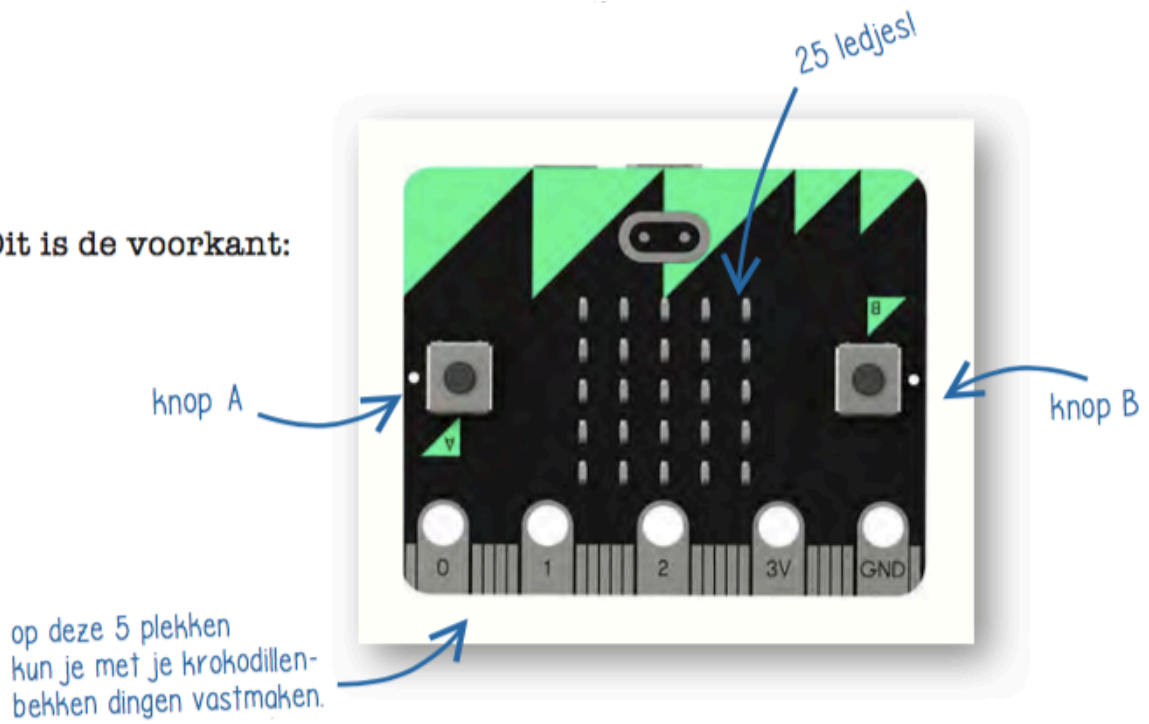
*Figuur 1: Oude computer (links) vs nieuwe computer (rechts)*

Een computer is een elektrisch apparaat dat niks kan als “hem” niks verteld wordt. Vertellen wat een elektrisch apparaat moet doen heet programmeren. De allereerste computers werden geprogrammeerd door schakelaars om te zetten. Daarna ontstonden er machinetalen die nauwelijks leesbaar zijn voor mensen (zoals Chinees voor ons). De eerste programmeertaal die leesbaar was voor ons, was ‘Plankalkül’. Tegenwoordig zijn er honderden programmeertalen bij gekomen. Misschien ken je er al een paar? Een aantal bekende programmeertalen zijn bijvoorbeeld: Javascript, PHP, Python en Scratch.

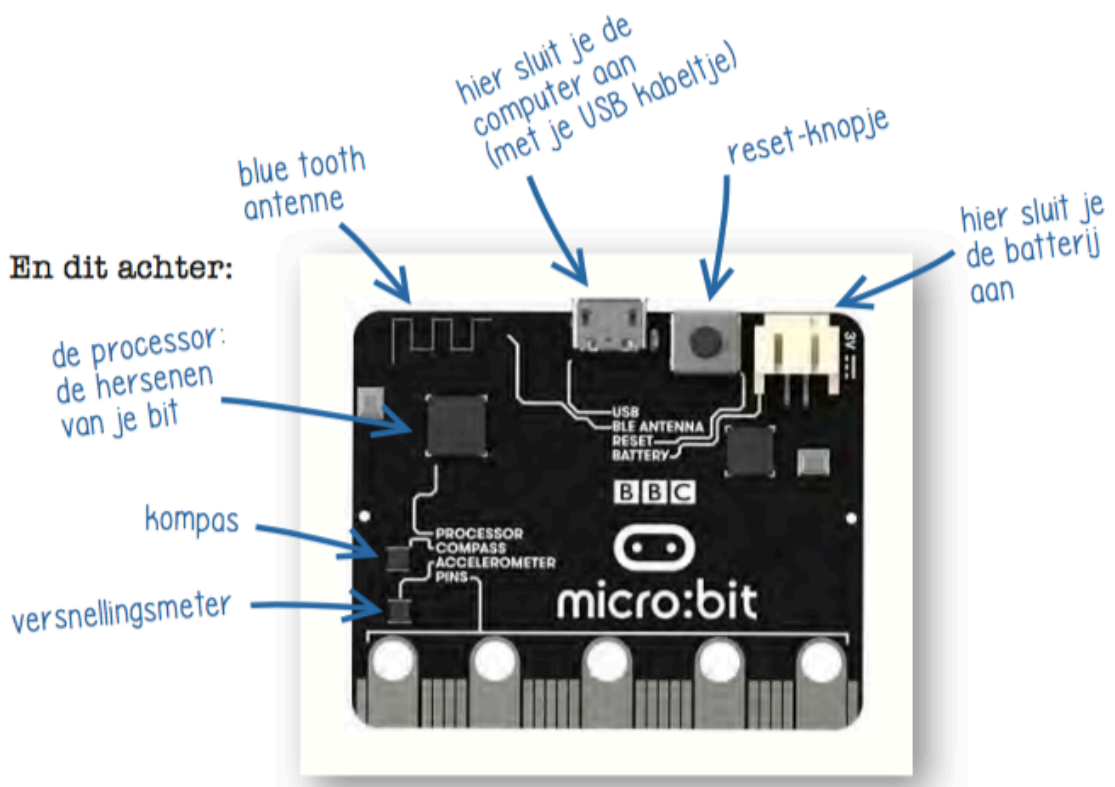
## **De micro:bit**

Wij gaan programmeren met de micro:bit. Daarvoor hoeven wij geen ingewikkelde taal uit ons hoofd te leren want wij gaan werken met de Block Editor van micro:bit, hierover later meer. De micro:bit is door de BBC gemaakt in samenwerking met onder andere Microsoft en Samsung. Pak je micro:bit er maar bij en bekijk hem even goed.

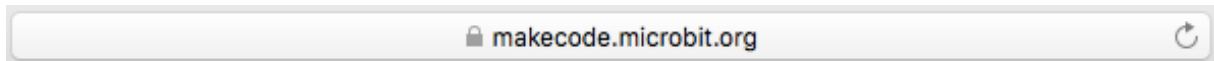
Dit is de voorkant:



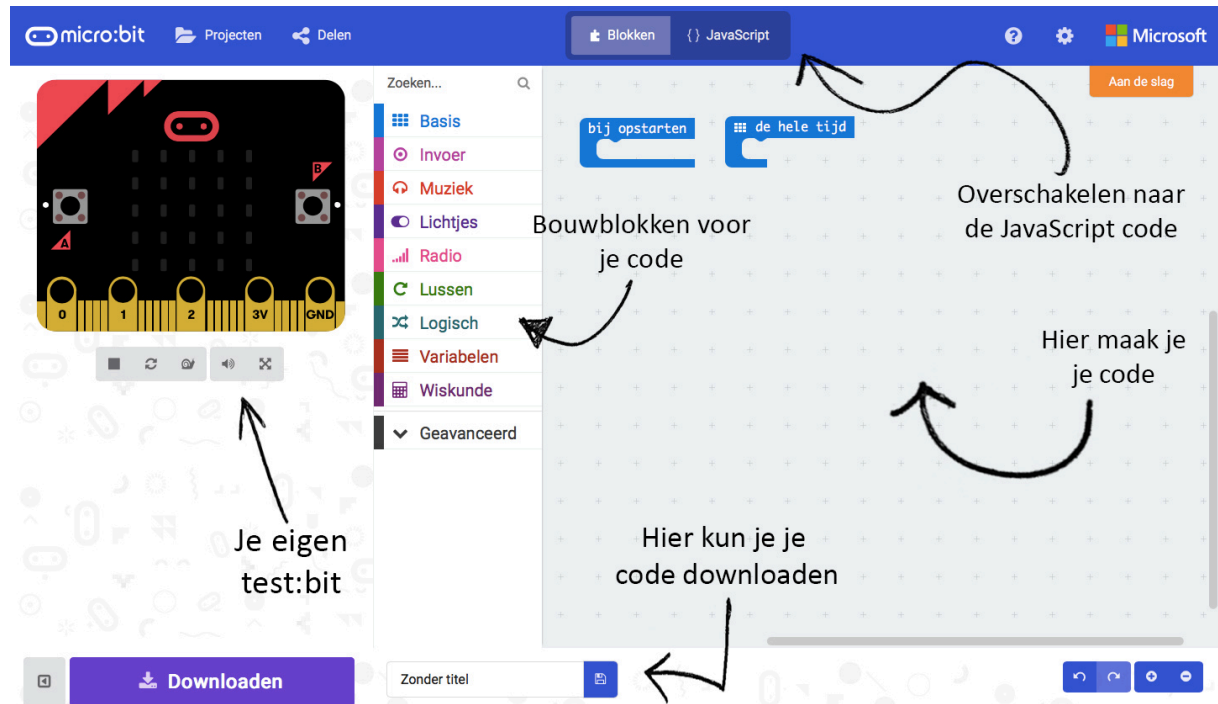
En dit achter:



Om de micro:bit te programmeren gaan we eerst op internet naar

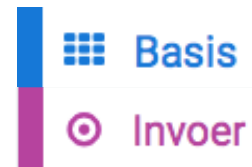


Druk dan op  en zet de taal op Nederlands. Als je dit gedaan hebt zie je onderstaand scherm.



Als je je micro:bit en de programmeer website goed hebt bekeken kunnen we samen gaan starten met het programmeren van onze micro:bit.

# TEST 1

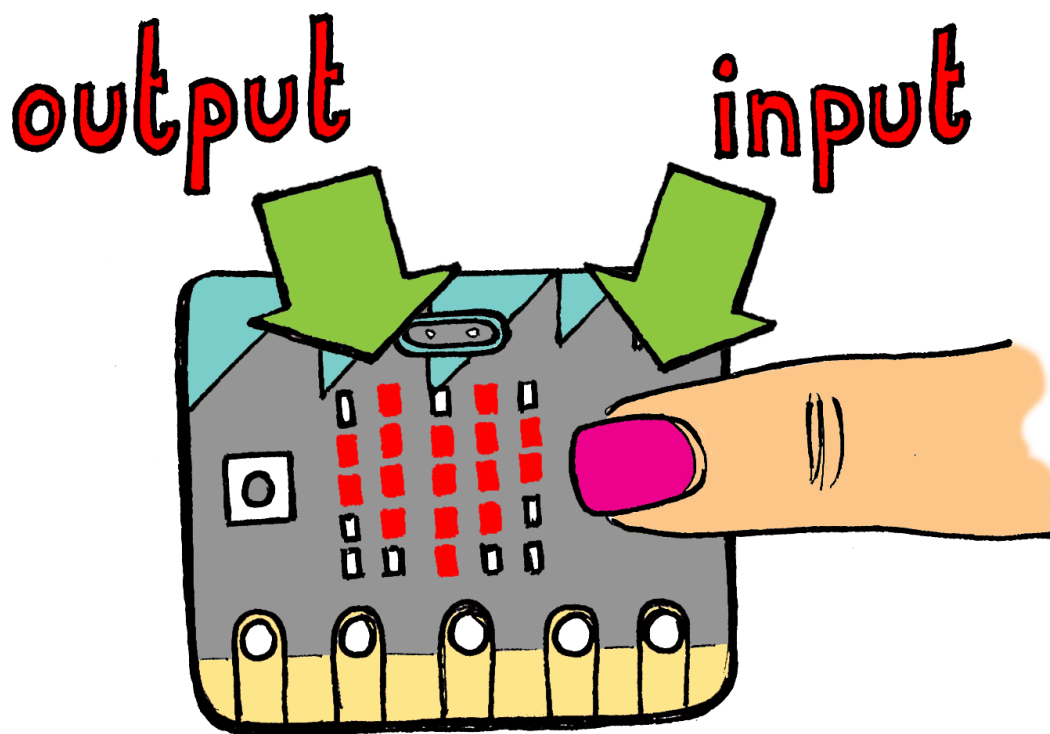


Als programmeurs een nieuw programma gaan maken, testen ze eerst altijd of dit programma wel werkt. Ze laten het programma vaak eerst 'hello, world!' op het beeldscherm schrijven en als dit werkt dan kunnen ze verder. Wij gaan dit ook doen.

Wij gaan een code maken die zorgt dat er op onze micro:bit 'hello, world!' komt te staan. Hiervoor moeten we de juiste blokken zoeken. Wij gaan vandaag vooral kijken bij de blokken van **basis en invoer**.

Bij **basis** staan vooral de 'uitvoer' blokken. Hier vind je de blokken om teksten, cijfers of pictogrammen te laten zien, maar je kunt ook zelf iets tekenen met de ledjes. Uitvoer is dus wat er uit de micro:bit komt. Wij noemen de uitvoer: **Output**.

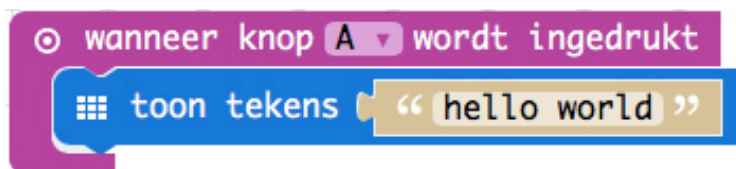
Bij **invoer** kun je blokken vinden die de micro:bit vertellen dat er iets gebeurt of dat hij iets moet doen. Bijvoorbeeld als je op knop A drukt of als je hem ondersteboven houdt. Invoer betekent dus wat je in de micro:bit stopt. Bij het begin van de lessen zie je steeds rechts bovenin over welke blokken je iets gaat leren. Wij noemen de invoer: **Input**.



Als je wat blokken bij elkaar zoekt dan zie je dat bepaalde blokken precies in elkaar passen. Hierdoor kun je altijd zien of iets wel of niet past en of je het dus wel of niet kunt programmeren. Kijk nu in het menu bij **basis** en **invoer** en zoek onderstaande blokken bij elkaar.



Zet de blokken in elkaar zodat het er zo uit komt te zien:



Je hebt nu je eerste programma gemaakt! Nu kun je hem even testen op je test:bit links op je scherm en daarna kunnen we hem downloaden op onze echte micro:bit. Geef je programma hiervoor eerst een naam en druk daarna op downloaden.

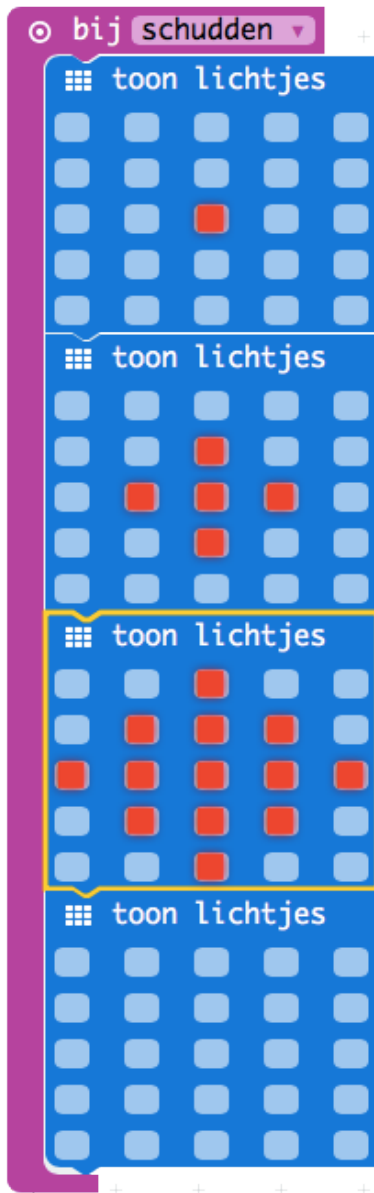


Nu is je programma gedownload en hoef je hem alleen nog op je micro:bit te uploaden. Hiervoor plug je de micro:bit met het kabeltje in je computer en verplaats je het bestand (.hex) uit de map downloads naar je micro:bit. Als dit is gelukt verdwijnt de micro:bit even van de computer en daarna verschijnt hij weer.

Haal je micro:bit uit je computer, doe de batterij in de micro:bit en test je programma.

## TEST 2

Je kunt ook een animatie maken met de micro:bit. Dit doe je door een aantal schermen achter elkaar te laten zien. Je kunt hierbij kiezen voor pictogrammen of je kunt zelf figuren tekenen. Dat laatste zie je terug in dit voorbeeld.



Als je goed kijkt zie je dat de input nu schudden is. Je kunt allerlei soorten input gebruiken en het ligt er dus maar net aan wat je wil of nodig hebt voor je programma.

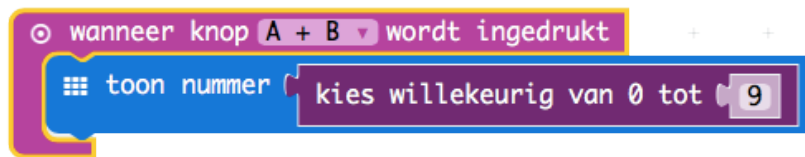
**Opdracht:** Druk op schudden en kijk welke opties je nog meer hebt. Schrijf er 3 op:

**Opdracht:** Maak bovenstaande animatie na en test dit op je micro:bit.

## TEST 3

Als jij en je broer allebei het laatste toetje willen, dan kiezen je ouders vaak de methode 'kies een getal onder de 10'. Je grote broer mag altijd als eerste raden en hij weet dat je vader bijna altijd voor het getal 9 kiest. En zo loop jij dus elke keer dat laatste toetje mis.

Hier ga jij iets aan doen. De micro:bit kan namelijk, in tegenstelling tot mensen, altijd willekeurig een nummer kiezen. Dit is terug te vinden bij 'Wiskunde'. Je vader drukt in dit geval op de A- en B-knop tegelijk en dan kiest de micro:bit willekeurig een getal onder de 10. Als je de micro:bit willekeurig iets wilt laten doen, kies je eigenlijk altijd voor deze optie. Later gaan we leren hoe we aan deze willekeurige getallen een betekenis kunnen geven.



**Opdracht:** Het nummer 0 kan nu ook als nummer gegeven worden. Hoe kun je er voor zorgen dat de micro:bit alleen uit de nummers 1 tot en met 9 er uit kunnen komen?

**Opdracht:** Maak dit na en test dit op je micro:bit.

## EIND VAN DE TESTFASE

Als alle bovenstaande tests gelukt zijn, kun je beginnen met de opdrachten. Als je een opdracht hebt gemaakt, kom je met je micro:bit en je boekje naar de docent voor een paraaf.





# LES 2 - VARIABELEN

Variabelen

Wiskunde

Bij programmeren wordt vaak gebruik gemaakt van **variabelen**.

Een variabele is een beetje te vergelijken met een wasknijper. Je kunt aan de wasknijper een getal hangen maar er kan maar 1 getal tegelijk in hangen want als je er nog een getal bij wilt hangen dan valt het vorige getal er uit. Daarnaast kun je ook een woord aan de wasknijper hangen. Een variabele werkt ongeveer hetzelfde en dit getal kun je op 2 manieren veranderen:

1. Je kunt de variabele veranderen met een bepaald aantal. Je kunt bijvoorbeeld zeggen dat je de variabele wilt veranderen met 1. Als aan je wasknijper dan het getal 3 hangt dan verandert dit naar 4.
2. Je kunt de variabele instellen op een bepaald getal. Je kunt bijvoorbeeld zeggen dat je de variabele wilt instellen op 9. Als aan de wasknijper dan het getal 3 hangt dan valt 3 er uit en wordt 9 aan de wasknijper gehangen.

Als je wilt kun je ook naar de wasknijper kijken om te zien welk getal erin zit en eerst hangt aan de wasknijper altijd het getal 0.



## TEST 1

Laten we bovenstaand voorbeeld eens vertalen naar onze micro:bit. De wasknijper is dan de variabele en we willen dat deze variabele iets telt. Hiervoor moeten we eerst een variabele aanmaken. Hiervoor ga je naar **Variabelen**. Druk op 'Maak een variabele' en noem deze 'Tellen' (of een andere naam, dit maakt niet uit). Als het goed is, staat hij er nu bij in de lijst:

tellen ▾

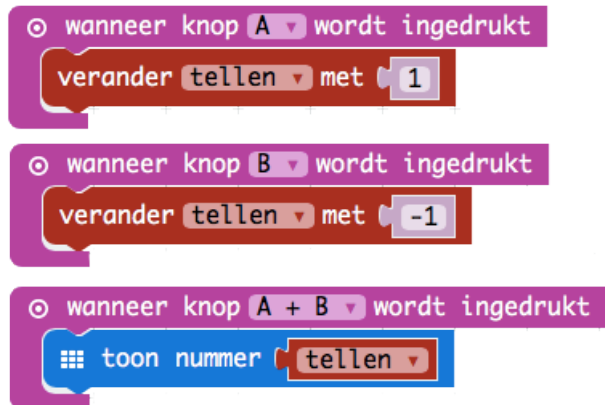
Deze variabele kunnen wij veranderen met een **input**. Hoe de variabele verandert kun je zelf bepalen. Je wil bijvoorbeeld dat als je op de A-knop drukt of als je de micro:bit schudt, dat de variabele verandert. Je kunt zeggen dat met een bepaalde input de variabele 'tellen' op 3 wordt gezet. Dan gebruik je:

stel tellen ▾ in op 3

Wij willen dat de variabele echt gaat tellen en dus moeten wij een andere manier gebruiken. Wij kunnen namelijk ook zeggen dat de variabele niet op een bepaald getal wordt ingesteld, maar dat hij wordt veranderd. Bijvoorbeeld: elke keer als wij op de A-knop drukken dan verandert 'tellen' met 1.

wanneer knop A ▾ wordt ingedrukt  
verander tellen ▾ met 1

Als wij nu op de A-knop drukken dan wordt de variabele 1 meer. Dus stel wij drukken 10 keer op de A-knop, dan staat de variabele 'tellen' op 10. We willen ook dat als we op de B-Knop drukken, de variabele 1 minder wordt en dat we kunnen zien welk getal de variabele is. Dit gebeurt met onderstaand voorbeeld.



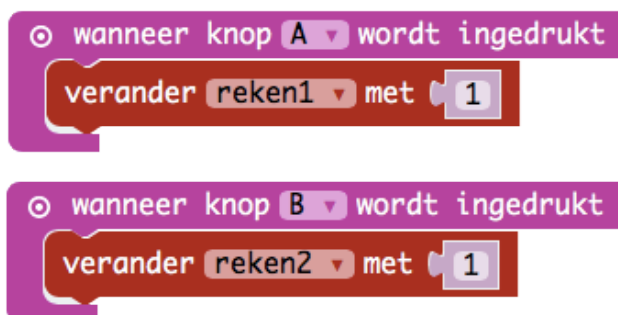
**Opdracht:** Maak dit na en test dit op je micro:bit en zorg dat de variabele constant in beeld is.

Een variabele is dus niks anders dan een soort wasknijper waarmee je je micro:bit iets kunt laten onthouden en bijhouden. Welke naam je op de wasknijper plakt maakt niet uit. Het is wel handig om het een naam te geven waaraan je kunt zien wat het doet. Je kunt namelijk oneindig veel variabele gebruiken in een programma.

## TEST 2

Naast optellen kunnen we ook vermenigvuldigen, delen en zelfs met machten rekenen met de micro:bit. Hiervoor hebben we meerdere variabele nodig, waarmee we gaan rekenen. Als je niet zo goed bent in vermenigvuldigen, kan de micro:bit je hierbij helpen. We gaan dus een rekenmachine maken van de micro:bit. Met de A-knop en B-knop willen we bepalen welke getallen we gaan vermenigvuldigen en als we de A- en B-knop tegelijk indrukken, willen we zien wat de uitkomst is.

Laten we bij het begin beginnen. We gaan twee variabelen aanmaken die optellen hoe vaak we op A-knop of op B-knop drukken. Dit hebben we bij de vorige test al gezien en bij dit voorbeeld zijn de variabelen 'reken1' en 'reken2' genoemd.



Nu krijgen we dus twee verschillende getallen als we willen en moeten we de micro:bit alleen nog vertellen wat hij met deze twee getallen moet doen.

Wij willen dat de micro:bit het getal van variabele 'reken1' vermenigvuldigt met het getal van variabele 'reken2'. Hiervoor maken we een nieuwe variabele aan en deze noemen we 'uitkomst'. Als wij nu op knop-A+B drukken moet 'uitkomst' worden ingesteld op 'reken1' keer 'reken2' want:

$$\text{Reken1} \times \text{Reken2} = \text{uitkomst}$$

Dit is te vinden bij 'Wiskunde' en zoals je daar ziet, passen de variabele precies in de puzzelstukjes. Daarna wil je dat de micro:bit de uitkomst laat zien. En uiteindelijk moet je weer opnieuw kunnen beginnen en moeten de variabelen weer op 0 worden gezet.

```
wanneer knop A wordt ingedrukt
  verander reken1 met 1

wanneer knop B wordt ingedrukt
  verander reken2 met 1

wanneer knop A + B wordt ingedrukt
  stel uitkomst in op reken1 x reken2
  toon nummer uitkomst
  pauzeer (ms) 500
  stel reken1 in op 0
  stel reken2 in op 0
  Wis scherm
```

**Opdracht: Waarom staat er een blok met 'Pauzeer' en een blok met 'Wis scherm'?**

**Opdracht: Maak dit na en test het op de micro:bit.**

## EIND VAN DE TESTFASE

Als alle bovenstaande tests zijn gelukt, kun je beginnen met de opdrachten. Als je een opdracht hebt gemaakt kom je met je micro:bit en je boekje naar de docent voor een paraaf.

# OPDRACHTEN LES 2

OPDRACHT	PUNTEN

Score les 2:

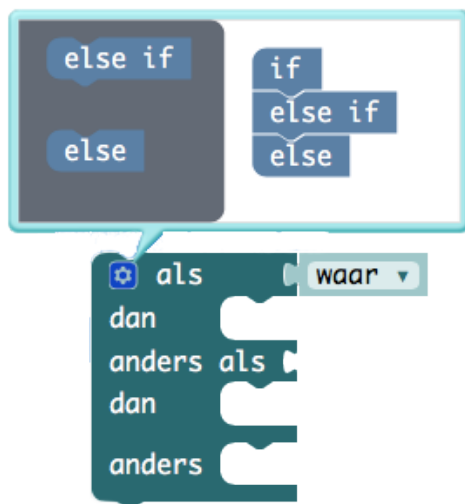
# LES 3 - ALS, DAN, ANDERS

We hebben vorige les geleerd hoe we variabelen kunnen instellen en hoe we deze kunnen veranderen. Deze les gaan we de micro:bit keuzes laten maken door hem te laten vergelijken. Het werkt een beetje hetzelfde als bij jou en mij. Als het bijvoorbeeld regent dan pak je je paraplu en anders niet. Je zou het ook kunnen vergelijken met het alarm van je wekker die je om 07:00 hebt gezet. Als het 07:00 is dan moet hij een geluid maken en anders moet hij niks doen.

Dit kun je ook doen met je micro:bit en dit heet in het Engels een 'if-then-else statement' en ziet er zo uit:



Op de plek van 'waar' kunnen we variabelen vergelijken en aan de hand van die vergelijking de micro:bit keuzes laten maken. Mocht je hier bij een opdracht niet uitkomen dan kun je eerst een flowchart maken om de keuzes zichtbaar te maken. Als we weer terugdenken aan de wekker dan vullen we in: **Als** tijd = 07:00 **dan** speel alarm af, **anders** doe niks. We kunnen deze keuze ook nog uitbreiden, hiervoor moeten we op het tandwieletje drukken en dan zie je dit:



**Opdracht:** Maak bovenstaand Als-Dan-Anders blok na.

Op de plek van 'waar' kun je een vergelijking invoegen. Met deze vergelijking kun je bijvoorbeeld zeggen: als mijn variabele het getal 3 is dan moet je dit plaatje laten zien. Zo'n vergelijkingsblok ziet er zo uit, maar kun je dus ook veranderen:



## TEST 1

Je wil een programma maken waarbij je, door op de A-knop te drukken, steeds wisselt tussen een blijde en een verdrietige smiley. Hiervoor heb je een variabele nodig die, door op de A-knop te drukken, steeds 1 groter wordt. Deze variabele kun je 'smile' noemen. Nu moeten we de micro:bit gaan vertellen wat hij moet doen. We hebben dus onderstaande vergelijking nodig:



Nu kunnen we de micro:bit vertellen dat, als dit waar is, hij een blijde smiley moet laten zien. Daarna doen we hetzelfde maar dan als 'smile' = 1 met een verdrietige smiley. Als wij hierna weer op de A-knop drukken wordt de variabele 2 en dan moet hij weer terug naar de blijde smiley.

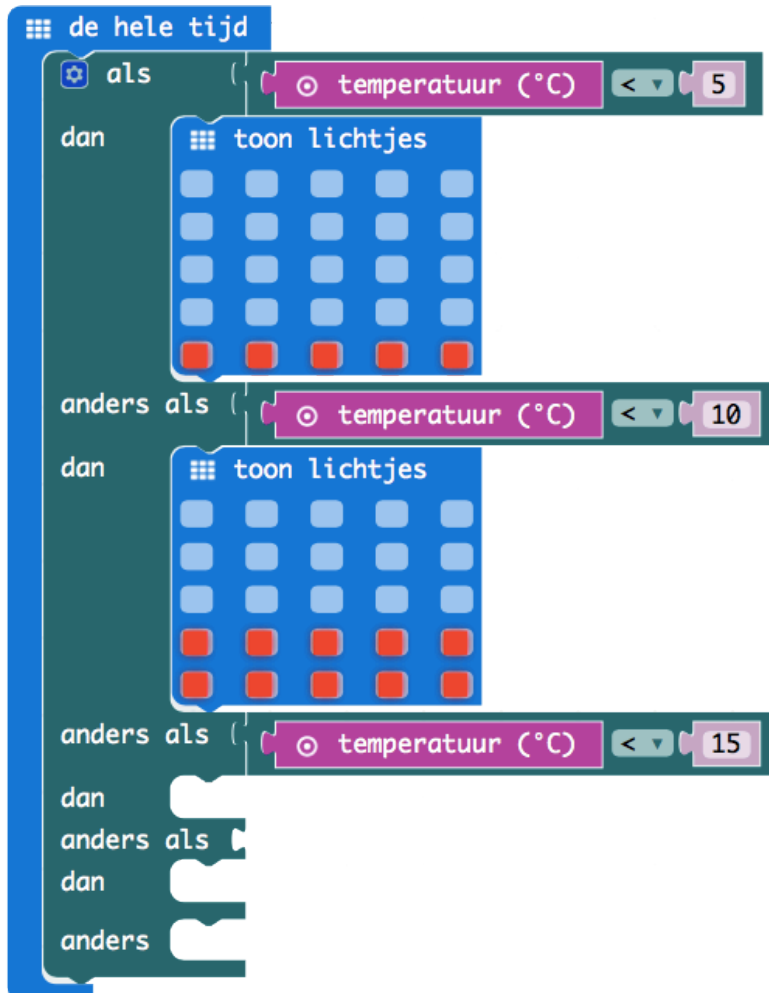
Omdat we deze opdracht al gemaakt hebben, hoeven we de variabele hier alleen weer terug te sturen. Dit doen wij door te zeggen dat als de variabele anders is dan 0 of 1, hij weer terug moet naar 0. Hiermee zorg je dat als de variabele van 1 naar 2 verandert, hij gelijk weer 0 wordt en dus weer de blijde smiley laat zien. Hieronder zie je de code uitgewerkt:



**Opdracht:** Maak bovenstaand programma na en test dit op je micro:bit. Maak er ook nog een derde functie bij.

## TEST 2

Naast alle variabelen die je zelf kunt maken zijn alle meters in de micro:bit ook variabelen. We kunnen bijvoorbeeld een thermometer maken van onze micro:bit. Onder de 5 graden Celsius laat de micro:bit 1 streep zien, onder de 10 graden Celsius laat de micro:bit 2 strepen zien en zo gaan we door tot en met 5 strepen. Als we dat programmeren ziet dit er zo uit:



**Opdracht:** maak bovenstaand blokkenschema na en maak het af.

## EIND VAN DE TESTFASE

Als alle bovenstaande tests zijn gelukt kun je beginnen met de opdrachten. Als je een opdracht hebt gemaakt, kom je met je micro:bit en je boekje naar de docent voor een paraaf.

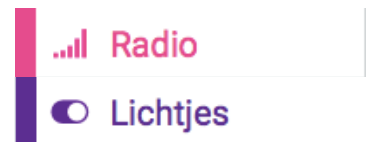


# OPDRACHTEN LES 3

OPDRACHT	PUNTEN

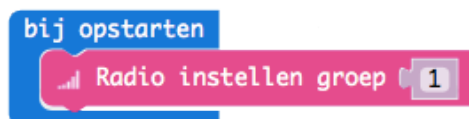
Score les 3:

# LES 4 - RADIO



Je micro:bit kan ook praten met andere micro:bits en zelfs met je telefoon. Hierdoor kun je de gegevens van alle meters uit je micro:bit verwerken maar je kunt ook vanaf de ene micro:bit iets versturen naar de andere micro:bit. Je micro:bit wil dan niet alleen weten wat je gaat verzenden, maar ook wat je gaat ontvangen. Als je micro:bit daarna het juiste signaal ontvangen heeft, kun je dit weer omzetten naar een actie.

Wij gaan met heel de klas in tweetallen nummers en woorden naar elkaar sturen. Om nou te zorgen dat dit goed verloopt en om te zorgen dat de micro:bit weet naar welk signaal hij wel en niet moet luisteren, krijgt iedereen zijn eigen groep. Als je in tweetallen aan de slag gaat kun je kiezen uit twee radiogroepen. Je moet dus bij elk programma dat je gaat schrijven, en waarbij je wilt dat de micro:bits met elkaar communiceren, als eerste dit in je code zetten:



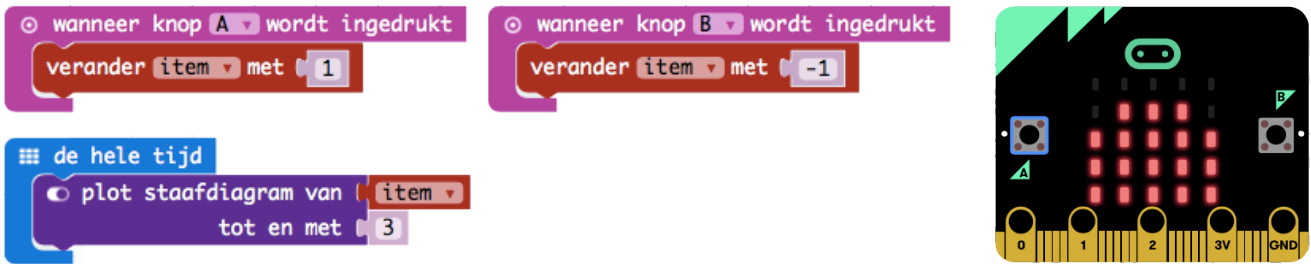
**Opdracht:** Zet nu op de voorkant van je boekje welk nummer jij hebt gekregen.

Als je daarna bijvoorbeeld een nummer gaat verzenden dan moet je zorgen dat de micro:bit die dit moet ontvangen, weet dat hij moet wachten op een nummer. Dit nummer komt binnen als variabele 'receivedNumber', wat ontvangen nummer betekent, en heeft dus de waarde die is verstuurd.

## TEST 1

Er liggen verspreid over dit lokaal drie micro:bits verstopt die in groep 1, groep 2 of groep 3 de hele tijd een signaal uitzenden. Aan jullie de taak om deze te zoeken. Dit gaan we doen door de sterkte van het signaal te onderzoeken. Hiervoor moeten we eerst kijken welke micro:bit, of in welke groep van het signaal, we gaan zoeken. Daarna moeten we dit signaal omzetten naar een manier die gelijk duidelijk maakt of we dichtbij of juist ver van de micro:bit zijn. Dit gaan we doen met een staafdiagram en deze kun je vinden bij 'Lichtjes'.

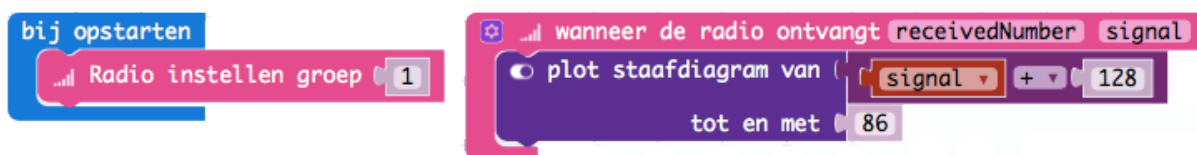
Als je onderstaande code namaakt dan laat het staafdiagram zien hoe groot de waarde van de variabele 'item' is. Door een staafdiagram te plotten van een variable kun je deze dus zichtbaar maken.



Wij willen nu het signaal van de verstopte micro:bit ontvangen en de sterkte daarvan plotten zodat we kunnen zien of we wel dichterbij komen. Hiervoor moeten we dus niet alleen het nummer ontvangen dat hij uitzendt maar ook de sterkte van het signaal. Om dit te doen moeten we dit toevoegen in de balk zoals hieronder is te zien.



Nu ontvangt de radio niet alleen het nummer dat gestuurd is maar ook hoe sterkt het signaal is. Om dit signaal daarna in een staafdiagram te kunnen plotten moeten we de variabele van het signaal verhogen met +128 om deze duidelijk te kunnen plotten.



**Opdracht:** Maak bovenstaande code na (met het nummer dat jij kiest) en ga op zoek naar de verstopte micro:bit

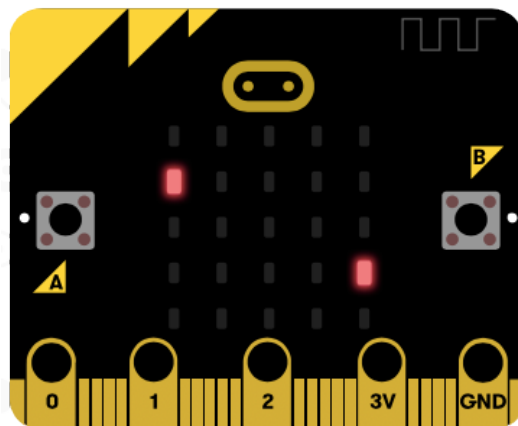
## TEST 2

Als we kijken naar de ledlampjes op de micro:bit dan heeft elk lampje zijn eigen coördinaat. Hierdoor kun je elk lampje apart aansturen als je dit op de juiste manier programmeert. Hoe deze coördinaten zijn verdeeld, zie je in onderstaande tabel.

(X,Y)	X				
Y	(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)	(3,4)
	(0,4)	(1,4)	(2,4)	(3,4)	(4,4)

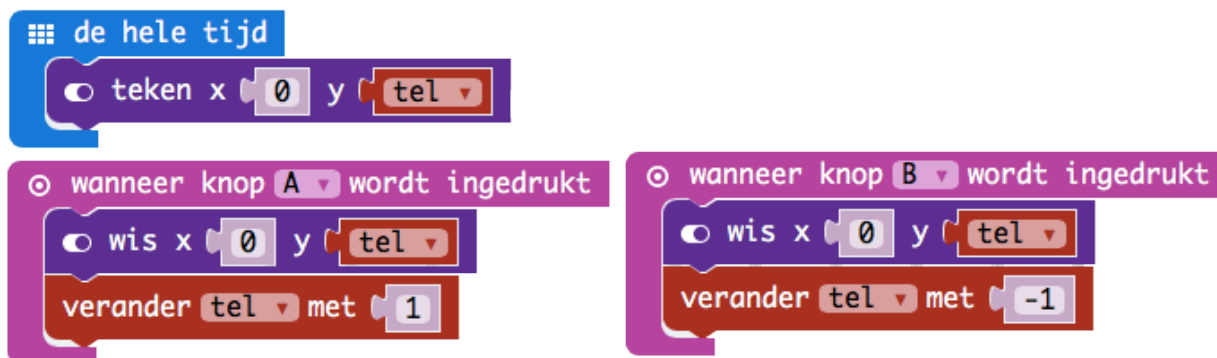
Tabel 1: coördinaten ledlampjes micro:bit

We gaan een code maken waarbij we niet alleen de ledlampjes op onze eigen micro:bit gaan laten bewegen maar ook die op de micro:bit van onze buurman of buurvrouw. Jij zelf kunt het ledlampje bewegen van (0,0) tot (0,4) en met de andere micro:bit kun je het ledlampje bewegen van (4,0) tot (4,4).



Als we eerst kijken naar ons eigen lampje dan willen we dus dat deze het punt tekent waar X nul is en Y de variabele is die wij kunnen veranderen, in dit geval de variabele 'tel'. Als de micro:bit eenmaal een punt heeft getekend, gaat deze pas weg als je hem wist. In de onderstaande code zie je dus ook dat wij de hele tijd het punt (0,tel) tekenen.

Als wij daarna op de A-knop drukken, wist de micro:bit eerst het vorige punt en verandert daarna gelijk tel met 1: het stipje springt een ledje naar beneden. Als wij op de B-knop drukken dan wist hij weer eerst het vorige punt en verandert daarna gelijk 'tel' met -1, het stipje springt een ledje omhoog.



Nu willen wij ook een ledje dat door de andere micro:bit wordt bestuurd. Hiervoor moet de micro:bit op het moment dat er een knop wordt ingedrukt, ook een signaal naar de andere micro:bit sturen en deze micro:bit moet het signaal van de ander ontvangen en tekenen. Dit zie je hieronder staan bij 'wanneer knop A wordt ingedrukt' en 'wanneer knop B wordt ingedrukt'.

Als jouw micro:bit een signaal ontvangt dan moet hij eerst het vorige punt wissen en daarna de variabele voor het ontvangen signaal, deze heb ik 'ontv' genoemd, veranderen. In dit geval sturen wij of 1 of -1 naar de andere micro:bit en dus willen wij dat de variabele 'ontv' verandert met het ontvangen signaal. Als de andere micro:bit nu 1 ontvangt, wordt die variabele 1 meer. En als de micro:bit -1 ontvangt, wordt de variabele 1 minder.

Als wij daarna kijken bij wat de micro:bit de hele tijd moet doen dan zien wij dat onze eigen variabele 'tel' nog steeds hetzelfde is maar dat de variabele 'ontv' daarbij is gekomen. Deze tekenen we aan de andere kant van ons scherm dus op punt (4,ontv). Aan het begin is 'ontv' 0 en als de andere micro:bit op de A-knop drukt wordt 'ontv' 1 meer en verplaatst het ledlampje 1 naar beneden en als de B-knop wordt ingedrukt wordt, wordt 'ontv' 1 minder en verplaatst het ledlampje 1 naar boven.

Als je deze code uitprobeert zal je echter zien dat er een fout in zit. Onderzoek waar deze fout zit en probeer deze eruit te halen.

```

    ☉ wanneer knop A wordt ingedrukt
      ☐ wis x 0 y tel
      ... Radio verzend nummer 1
      verander tel met 1

    ☉ wanneer knop B wordt ingedrukt
      ☐ wis x 0 y tel
      ... Radio verzend nummer -1
      verander tel met -1

    ⚙️ ... wanneer de radio ontvangt receivedNumber
      ☐ wis x 0 y ontv
      verander ontv met receivedNumber

    bij opstarten
      ... Radio instellen groep 1
      stel ontv in op 0
      stel tel in op 0

    de hele tijd
      ☐ teken x 0 y tel
      ☐ teken x 4 y ontv
  
```

**Opdracht:** Welke fout zat er in bovenstaand programma en hoe heb je dit opgelost?

**Opdracht:** Neem de goede code over en test deze op je micro:bit.

## TEST 3

Wij gaan een spelletje maken waarbij twee personen proberen op dezelfde knop te drukken, terwijl ze 10 meter uit elkaar staan en niet kunnen zien op welke knop de ander drukt.

Als ze allebei op dezelfde knop hebben gedrukt dan laat de micro:bit een vrolijke smiley zien. Als ze op een andere knop hebben gedrukt dan laat de micro:bit een verdrietige smiley zien.

Je wil natuurlijk dat de micro:bit pas gaat vergelijken of dezelfde knop is ingedrukt als jullie beiden een knop hebben ingedrukt. Om dit voor elkaar te krijgen moeten we zorgen dat een variabele bijhoudt of er al een radiosignaal ontvangen is en of er al gedrukt is. Ik heb daarvoor een variabele aangemaakt die 'druk' heet. Als er gedrukt is dan is deze variabele 1. Ik heb ook een variabele aangemaakt die 'ontvangen' heet. Als er een signaal is ontvangen dan is deze variabele 1 geworden. Als beide variabelen 1 zijn dan kan jouw micro:bit gaan vergelijken: je ziet hier dan een vinkje. Als dit gelukt is moeten ze weer allebei op 0 worden gezet zodat je weer opnieuw kunt beginnen. Dit zie je in onderstaande code:

```
Scratch code for a radio-based game on a micro:bit. The code is organized into three main sections: 'bij opstarten', 'wanneer knop B wordt ingedrukt', and 'wanneer knop A wordt ingedrukt'.  
1. 'bij opstarten':  
- 'Radio instellen groep' set to 1.  
- 'wanneer de radio ontvangt receivedNumber' block with 'stel ontvangen in op' set to 1.  
2. 'wanneer knop B wordt ingedrukt':  
- 'stel klik in op' set to 1.  
- 'Radio verzend nummer' set to 2.  
3. 'wanneer knop A wordt ingedrukt':  
- 'stel klik in op' set to 1.  
- 'Radio verzend nummer' set to 2.  
4. 'de hele tijd' loop:  
- 'als' block with condition: 'klik = 1 en ontvangen = 1'.  
- 'dan' block:  
  - 'toon pictogram' (happy smiley).  
  - 'stel ontvangen in op' set to 0.  
  - 'stel klik in op' set to 0.  
- 'anders' block:  
  - 'toon lichtjes' (sad smiley).
```

Als dit gelukt is hoeven we alleen nog de variabelen met elkaar te vergelijken. Ik heb hier de variabele 'persoon2' het nummer gegeven van het ontvangen nummer zodat ik deze kan vergelijken met de variabele 'persoon1'.

```

bij opstarten
  Radio instellen groep 1

wanneer de radio ontvangt receivedNumber
  stel ontvangen in op 1
  stel persoon2 in op receivedNumber

wanneer knop A wordt ingedrukt
  stel klik in op 1
  stel persoon1 in op 5
  Radio verzend nummer 5

wanneer knop B wordt ingedrukt
  stel klik in op 1
  stel persoon1 in op 10
  Radio verzend nummer 10

de hele tijd
  als (klik = 1) en (ontvangen = 1)
  dan
    als (persoon1 = persoon2)
    dan
      toon pictogram
      stel ontvangen in op 0
      stel klik in op 0
    anders
      toon pictogram
      stel ontvangen in op 0
      stel klik in op 0
  anders
    toon lichtjes
  
```

**Opdracht:** Neem deze code over en test dit op je micro:bit

## EIND VAN DE TESTFASE

Als alle bovenstaande tests zijn gelukt, kun je beginnen met de opdrachten. Als je een opdracht hebt gemaakt, kom je met je micro:bit en je boekje naar de docent voor een paraaf.



# OPDRACHTEN LES 4

OPDRACHT	PUNTEN

Score les 4:

# LES 5 - PINS

🎯 Pinnen

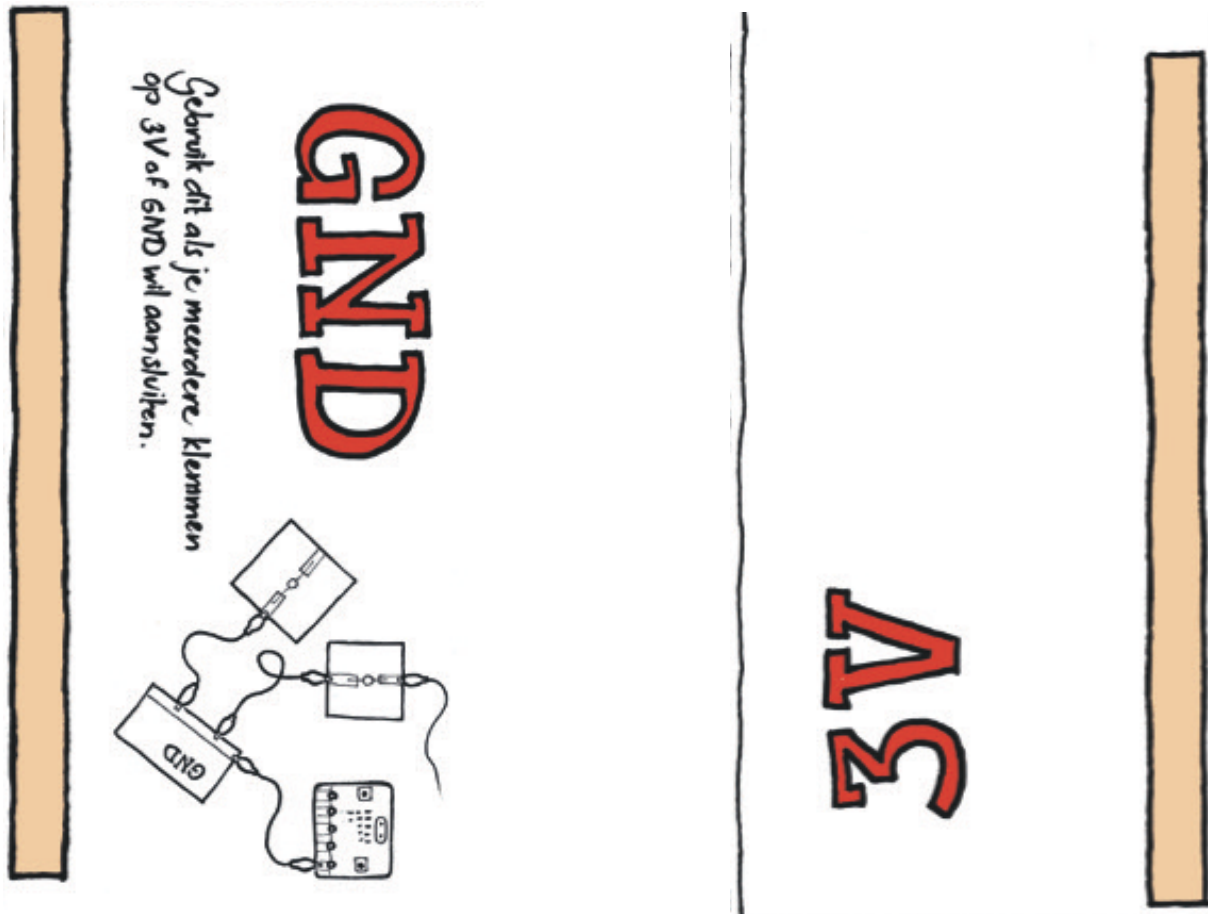
🎵 Muziek

Onze micro:bit bestaat al uit heel veel knoppen en lichtjes, maar wij kunnen ook nog allerlei dingen toevoegen aan de micro:bit. Dit kan via de **pinnen** die zich onder aan de micro:bit bevinden. Hiermee kun je andere ledlampjes, servo's en zelfs je koptelefoon aansluiten op je micro:bit. Als we naar onze micro:bit kijken dan zien we aan de onderkant 5 grote pinnen zitten. Deze pinnen gaan we deze les gebruiken om randapparatuur aan te sturen.

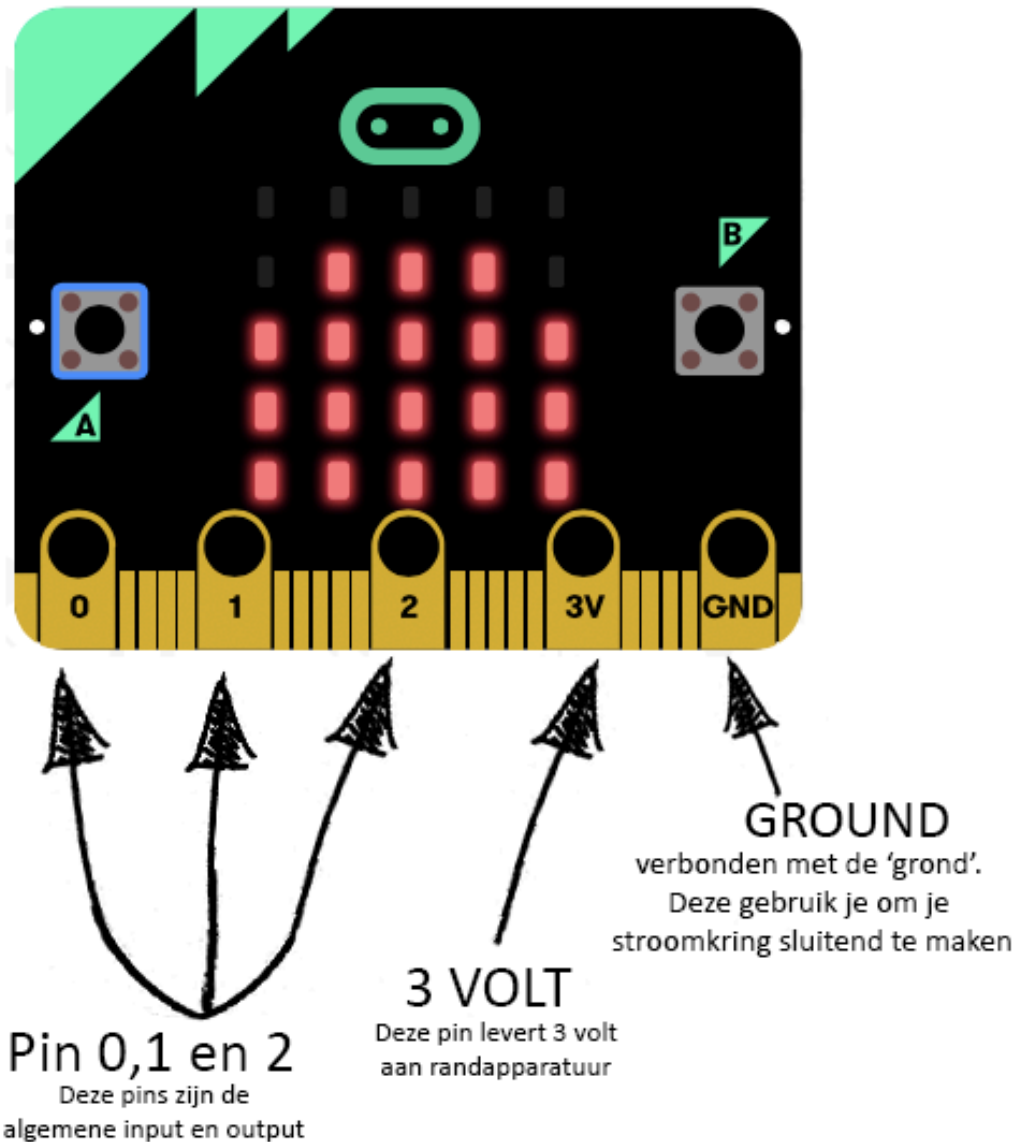
Als je goed naar de pin's kijkt zie je dat er drie verschillende pin's zijn. Eerst hebben we pin 0 tot en met pin 2. Deze kunnen we als input en als output gebruiken. Als je deze als output gebruikt komt er 3 Volt uit deze pin. Daarnaast zit de pin met 3V erop. Deze pin geeft constant 3 volt maar kan ook gebruikt worden als input voor spanning om de micro:bit te laten werken.

Als laatste staat er GND. Dit staat voor **ground** en deze pin heb je nodig om de stroomkring te sluiten en dus bijvoorbeeld een lampje te laten branden.

Als je meerdere krokodillenklemmen moet aansluiten op de ground of de 3 volt wordt dit lastig. Mocht je dit willen dan kun je een stukje kopertape op een papiertje plakken en daar alle krokodillenklemmen aan vastklemmen. Je kunt hiervoor onderstaande afbeelding uitknippen om de kopertape op te plakken

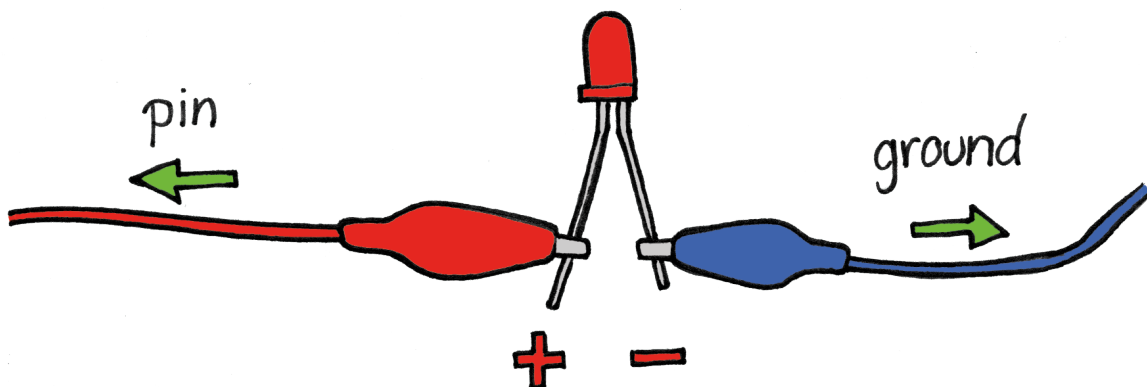


Daarnaast kan de micro:bit zo hele kleine stroompjes meten. Als jij een krokodillenbek vast hebt die met een pin verbonden is en je buurman of buurvrouw heeft een krokodillenbek vast die met de ground verbonden is, dan gaat er een heel klein stroompje lopen en dit kan de micro:bit dus meten. Hieronder zie je nogmaals aangegeven wat wat is op de micro:bit.



## TEST 1

We gaan beginnen met het aansluiten van een ledlampje en we gaan deze laten knipperen. We moeten hiervoor eerst het ledlampje aansluiten en dit gaan we doen met krokodillenklemmen. Een ledlampje heeft een lang en een kort pootje. Het lange pootje is de pluskant van het ledlampje en deze moet dus worden verbonden met pin 0, 1 of 2. Het korte pootje is de minkant van het LED lampje en deze moet worden verbonden met GND. In dit voorbeeld is het ledlampje verbonden met pin 0.



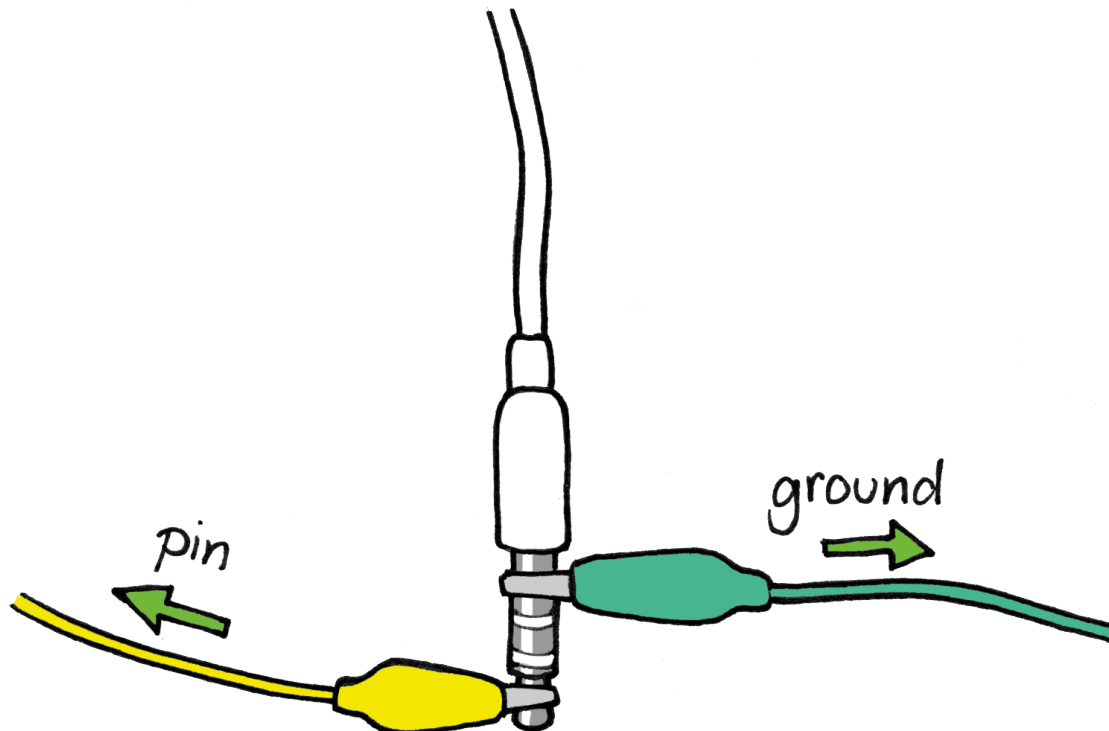
Daarna moeten we de code gaan maken. Als het lampje aan is dan moeten wij pin 0 digitaal naar 1 schrijven. 1 betekent dat het lampje aan staat en 0 betekent dat hij uit staat. Daar moet natuurlijk ook een pauze tussen om hem even aan en even uit te laten staan.

```
de hele tijd
  schrijf digitaal pin P0 naar 0
  pauzeer (ms) 500
  schrijf digitaal pin P0 naar 1
  pauzeer (ms) 500
```

**Opdracht:** Maak bovenstaande code na en test dit op je micro:bit.

## TEST 2

We kunnen onze micro:bit ook aansluiten op onze koptelefoon. Hiervoor moeten we de krokodillenklemmen gebruiken en deze op de juiste manier aansluiten op de stekker van onze koptelefoon. We moeten hiervoor een krokodillenklem van pin 0 naar het puntje van de stekker van je koptelefoon laten gaan en daarna van het bovenste deel naar de ground van de micro:bit. Als dit is gelukt kunnen we geluiden maken met onze micro:bit en misschien zelfs wel een liedje spelen.



Wij gaan de micro:bit een geluid laten maken dat wij kunnen veranderen door de lichtniveau aan te passen. De code hiervan zie je hieronder staan. Hiervoor moeten we eerst zorgen dat het lichtniveau wordt omgezet in een variabele. Deze variabele heb ik **'licht'** genoemd en deze wordt door de microbit steeds ingesteld op het lichtniveau.

Daarna moet deze variabele worden omgezet in een toon en dit doen we door in te stellen: Als de variabele 'licht' kleiner is dan 25 dan moet hij de toon 'Midden C' afspelen. Dit laatste is te vinden onder het kopje 'Muziek'. Als je daar een toon wilt kiezen, dan druk je op 'Midden C' en dan komt er een piano tevoorschijn. Kies nu de juiste toon en je kunt verder. Dit doen we daarna ook voor andere lichtsterktes.

Als je de code dan verder afmaakt, ziet het er zo uit:

```
de hele tijd
  rust (ms) 1 beat
  stel licht in op lichtniveau
  als
    licht < 25
  dan
    speel toon (Hz) Midden C
  anders als
    licht < 50
  dan
    speel toon (Hz) Midden D
  anders als
    licht < 100
  dan
    speel toon (Hz) Midden E
  anders als
    licht < 150
  dan
    speel toon (Hz) Midden F
  anders als
    licht < 180
  dan
    speel toon (Hz) Midden G
  anders
    speel toon (Hz) Midden A
```

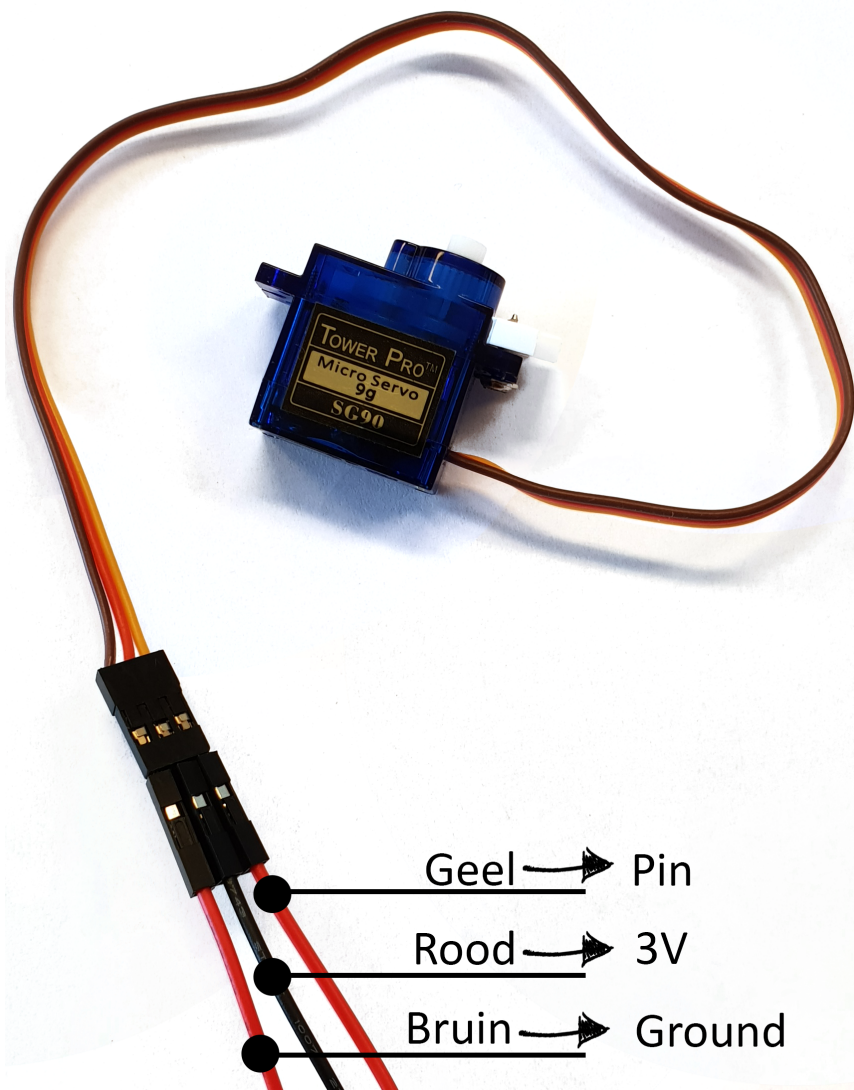
**Opdracht:** Maak bovenstaande code na, test het op je micro:bit en verbeter hem.

**Opdracht:** Ik heb de code verbeterd door:

## TEST 3

Met je micro:bit kun je ook een motortje besturen. Dit heet een servomotor. Deze motor wordt vaak gebruikt, bijvoorbeeld in de cruise control van een auto en in de vleugels van een vliegtuig. In kleine vorm gaan wij nu ook een servo gebruiken.

Een servo moet je op een bepaalde manier aansluiten, anders doet hij het niet. Hij moet natuurlijk verbonden zijn met een pin en met de ground, maar een servo heeft ook 3 volt nodig. Een servo heeft daardoor drie draadjes: geel, rood en bruin. Hiervan moet geel altijd naar de pin toe, rood moet altijd naar de 3 volt toe en bruin moet altijd naar de ground toe.



Nu willen wij graag weten of wij een korte broek aan kunnen en dit gaan we de micro:bit laten beslissen. In het beste geval zou je natuurlijk een micro:bit buiten kunnen hangen om de buitentemperatuur te meten maar wij doen het in dit voorbeeld met één micro:bit. Op de volgende bladzijde zie je een blad waar je je servo op kunt plakken om dit voorbeeld na te maken.

Laten we zeggen dat we een korte broek aan kunnen vanaf 20 °C. Onze micro:bit moet dan als het kouder is dan 20 °C naar 'Nee' wijzen en als het niet kouder is dan 20 °C naar 'Ja' wijzen. In de onderstaande code is dit het geval.

```
de hele tijd
  stel temperatuur in op temperatuur (°C)
  als temperatuur < 20
  dan
    schrijf servo op pin P0 naar waarde 315
  anders
    schrijf servo op pin P0 naar waarde 225
```

**Opdracht:** Maak deze code na en test dit met je micro:bit.

**Opdracht:** Je pijltjes wijzen niet gelijk de goede kant op. Probeer dit nu te debuggen waardoor de pijltjes wel de goede kant op wijzen.

Naast de randapparatuur die we nu getest hebben is er nog veel meer dat je aan kunt sluiten aan de micro:bit. Hieronder zie je een lijst met wat je nog meer aan zou kunnen sluiten:

**Piezo:** Een piezo kan een drukverschil registreren als input en trillen als output

**RGB-LED:** Een RGB led kan drie kleuren geven en de micro:bit kan dan hem vertellen welke kleur hij moet laten zien.

Daarnaast kun je ook analoge input aansluiten op de micro:bit. Als je deze wilt aansluiten moet je dit even vragen aan de docent. Voorbeelden hiervan zijn:

**LDR:** Een LDR is een lichtmeter en die meet dus het verschil in lichtsterkte

**Potmeter:** Een potmeter is een draaiknop. Deze meet hoever een draaiknop gedraaid is.

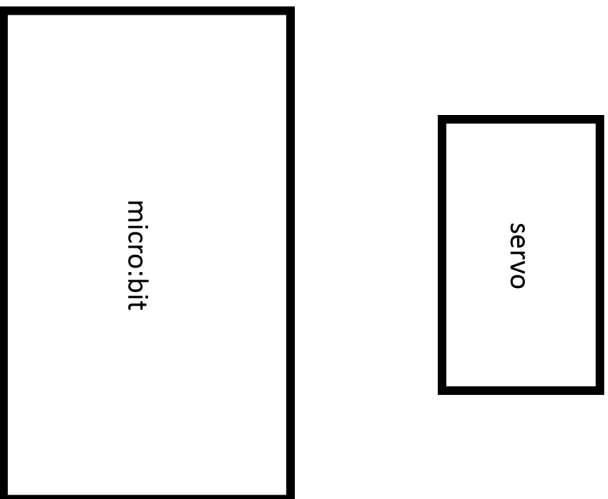
**Microfoon:** Een microfoon meet het geluidsniveau.

## EIND VAN DE TESTFASE

Als alle bovenstaande tests zijn gelukt, kun je beginnen met de opdrachten. Als je een opdracht hebt gemaakt, kom je met je micro:bit en je boekje naar de docent voor een paraaf.



# KAN IK VANDAAG EEN KORTE BROEK HAAN?





# EINDOPDRACHT

Jullie hebben nu al ontzettend veel geleerd over programmeren en hoe je dit moet doen op de micro:bit. Daarnaast hebben jullie gezien dat je met de micro:bit erg veel praktische oplossingen kunt verzinnen voor grote of kleine problemen in het dagelijks leven. Voor de eindopdracht is het de bedoeling dat je in tweetallen zelf iets gaat maken waarbij er minstens twee micro:bits met elkaar communiceren.

Je wordt voor de eindopdracht in een stage ingedeeld en je maakt de eindopdracht met iemand uit dezelfde stage. In welke stage je wordt ingedeeld is afhankelijk van je scores bij de opdrachten. Je hebt nu 5 (of 6) lessen gehad waarin je steeds 5 opdrachten hebt gemaakt.

Aantal punten behaald	Opdracht
25	Stage 1
20	Stage 2
15	Stage 3

## STAGE 1:

Jullie mogen zelf bedenken wat je wilt maken. Het is wel belangrijk dat er 2 micro:bits met elkaar communiceren en ik verwacht dat jullie het jezelf niet te makkelijk maken. Je idee werk je eerst even kort uit en pitch je aan de docent. Als je docent het goedgekeurd heeft, kun je beginnen met het uitwerken van je idee op de volgende bladzijde en daarna kun je aan de slag.

## STAGE 2:

Voor de eindopdracht mogen jullie uit de onderstaande lijst een onderwerp kiezen. Dit onderwerp is een schets van je product. Hoe het eindproduct er uiteindelijk uit gaat zien is aan jou. In de lijst staan enkele eisen waar het aan moet voldoen, maar je mag er dus altijd iets bij maken.

Kies een onderwerp uit onderstaande lijst:

- Hack je voordeur: De bel gaat af als er iemand op de deurmat staat, als er wordt aangebeld praat je micro:bit terug en de deur gaat vanzelf open met code of een bepaalde handeling met een magneet. Of hack je kamer: Als het te koud is, gaat de verwarming aan, als het te donker is gaat het licht aan en als de deur even open staat gaat hij vanzelf dicht.
- Maak een simulatie van een verkeerslicht langs een eenrichtingsweg waarbij er ook een mogelijkheid moet zijn voor voetgangers om over te steken.
- Maak een geheime doos die alleen opengaat met een speciale code. Wordt de doos geopend zonder code of is hij te ver bij je vandaan dan gaat er een alarm af van geluid en licht.

## STAGE 3:

Voor de eindopdracht mogen jullie uit onderstaande lijst een opdracht kiezen. Deze opdracht werk je uit en dit wordt je eindproduct. Als je de opdracht af hebt maar je wilt er nog iets bij verzinnen dan mag dat na goedkeuring van de docent.

- Maak een lovemeter en zorg dat hij er ook echt uit ziet als een lovemeter. Maak er iets omheen en zorg dat de twee personen tegenover elkaar op de knop moeten drukken. Daarnaast mag het percentage op de micro:bit worden getoond maar moet met een meter (servo) of ledlampjes worden verteld of het een match is of niet.
- Maak een simulatie van een verkeerslicht langs een weg. Het is een eenrichtingsweg waar een stoplicht staat. Dit stoplicht (micro:bit 1) staat steeds 20 seconde op groen en daarna gaat hij via oranje op rood waar hij 10 seconde op blijft staan. Als er een voetganger wil oversteken, drukt hij op de knop (micro:bit 2) waardoor het stoplicht van de auto via oranje op rood komt te staan en 30 seconde op rood blijft staan. Als het stoplicht eenmaal op rood staat, krijgt de voetganger 20 seconde om over te steken. Daarna gaat het licht 5 seconde knipperen en springt op rood.
- Maak een gitaar met je micro:bit en zorg dat je er liedjes op kunt spelen. Ga hiervoor naar <https://makecode.microbit.org/projects/guitar> en probeer dit na te maken. Als je gitaar eenmaal doet wat er in deze uitleg staat, probeer je hem zo te debuggen dat je er een liedje mee kunt spelen.

## EINDPRESENTATIE

Het laatste uur van de laatste les presenteer je je eindopdracht aan de klas. Deze presentatie mag 5 – 10 minuten duren en hierin moet worden uitgelegd wat het eindproduct is, hoe het werkt en hoe de code werkt. Wie van de twee uit het groepje de presentatie doet wordt bepaald op het moment van presenteren. Een micro:bit zal hier random één iemand uit het groepje kiezen en deze mag de presentatie geven.

In de presentatie zitten de volgende elementen:

- Het eindproduct wordt getoond samen met alle functies die in het eindproduct zitten.
- De code achter het eindproduct wordt uitgelegd. Wat doet wat en waarom.
- Wat zijn de problemen waar jullie tegen aan zijn gelopen en hoe hebben jullie dit opgelost?
- Zijn er vragen uit de klas?

# PLAN VAN AANPAK

Welke opdracht heb je gekozen? Leg in je eigen woorden uit wat je eindproduct wordt en aan welke eisen het moet voldoen.

Hoe ga je het vertalen naar je code? Hoe ga je welke eis oplossen?

Eis	Welke Input	Welke output	Code

Maak een schematische tekening van je eindproduct:

Gezien door docent

