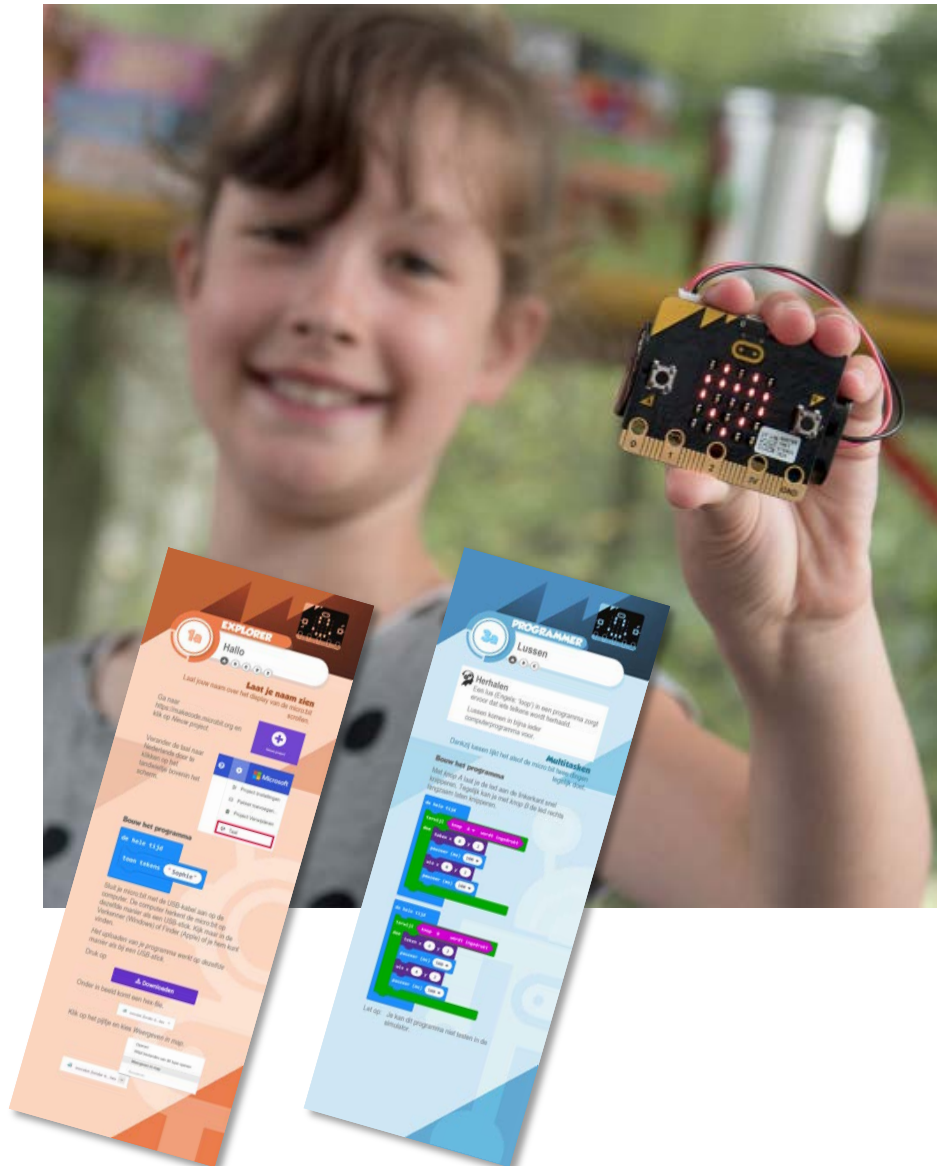


micro:bit kaarten

Explorer Programmer



Handleiding voor docenten

Inhoud

Vooraf	4
De micro:bit kaarten	5
Onderdelen van iedere kaart	5
Werken met de micro:bit kaarten	7
Werken met de micro:bit	10
Explorer kaarten	12
Programmer kaarten	12
Meer weten?	14
Explorer 1: Hallo	16
Laptop met Windows	16
Chromebook (Explorer kaart 1d)	17
iPad (Explorer kaart 1e)	17
Kaart 1a - Opdracht 1: Laat je naam zien	17
Kaart 1b - Opdracht 2: Klaar voor de start	17
Kaart 1c – micro:bit pairen	18
Kaart 1d – Ik heb een Chromebook	18
Kaart 1e – Ik heb een tablet	19
Explorer 2: Knoppen	20
Kaart 2a - Opdracht 1: Laat mijn naam zien als ik op knop A druk	20
Kaart 2a - Opdracht 2: Druk op alle knoppen	20
Explorer 3: Tellen	22
Kaart 3a - Opdracht 1: Tellen met de micro:bit	22
Kaart 3b - Opdracht 2: Stappen tellen	23
Explorer 4: Opslaan, importeren en delen	25
Over deze kaarten	25
Kaart 4A - Opdracht 1: Geheime boodschap opslaan	25
Kaart 4a - Opdracht 2: Geheime boodschap openen	25
Kaart 4b - Opdracht 3: Geheime boodschap op internet opslaan	26
Explorer 5: Herrie maken	27
Kaart 5a - Opdracht 1: Speel één toon	27
Kaart 5b - Opdracht: Speel een liedje	28
Explorer 6: Sensoren	30
Kaart 6a - Opdracht 1: Maak muziek met licht	30
Kaart 6a - Opdracht 2: Toontje hoger	30
Kaart 6b - Opdracht 3: Horen hoe warm het is	31
Explorer 7: Beweging	32
Kaart 7a - Opdracht 1: Kantel het licht	32
Kaart 7b - Opdracht 2: Van links naar rechts	34
Kaart 7b - Opdracht 3: Van links naar rechts en van voor naar achter	36
Explorer 8: Pinnen	37
Over het aansluiten van dingen op de micro:bit	37
Kaart 8a - Opdracht 1: Raak me (niet) aan	37
Kaart 8b - Opdracht 2: Knipperled	38
Explorer 9: Getallen	40
Kaart 9a - Opdracht 1: Speel papier-steen-schaar	40
Programmer 1: Variabelen en rekenen	41
Kaart 1a - Opdracht 1: Secondeteller	41
Kaart 1b - Opdracht 2: Rekenmachine	42
Programmer 2: Debuggen en algoritmes	44
Kaart 2a - Opdracht 1: Foutje	44

Kaart 2b - Opdracht 2: Plot erop los.....	45
Programmer 3a: Lussen	48
Kaart 3a - Opdracht 1: Multi-taken.....	49
Kaart 3b - Opdracht 2: Knippen (maar niet te vaak)	50
Kaart 3c - Opdracht 3: Playlist	51
Programmer 4: Voorwaarden	53
Tip Computational Thinking	53
Kaart 4a - Opdracht 1: Toonladder	53
Kaart 4b - Opdracht 2: EN en/of OF spelletje	55
Kaart 4c - Opdracht 1: Wie drukte als eerste?	58
Programmer 5: Meet het lichtniveau	60
Kaart 5a - Opdracht 1: Plot staafdiagram	60
Kaart 5a - Opdracht 2: Meet het lichtniveau	60
Programmer 6: Versnelling en beweging	62
Kaart 6a - Opdracht 1: X-moves	62
Kaart 6b - Opdracht 2: Waterpas	62
Kaart 6c - Opdracht 3: Maak een kompas	63
Programmer 7: Radio	65
Kaart 7a - Opdracht 1: Radioled	65
Kaart 7b - Opdracht 2: Speurneus	65
Programmer 8: Functies	67
Kaart 8a - Opdracht 1: Van links naar rechts	67
Programmer 9: Pins	69
Kaart 9a - Krijgt de plant genoeg water?.....	69
Kaart 9b – Schuifmuziek	70
Bijlage 1: Kaarten afdrukken.....	73

Vooraf

Hartstikke leuk dat je de micro:bit kaarten gebruikt. Op dit moment zijn er drie soorten kaarten:

- Explorer
- Programmer
- Maker

Deze handleiding beschrijft het gebruik van Explorer en Programmer kaarten, voor de Maker kaarten zijn er aparte docentenbladen.

Als je opmerkingen en aanvullingen hebt dan horen we die graag!

Bedankt!

Begin 2017 hebben ongeveer 15 mensen meegewerkt in het Docent Ontwikkel Team en de eerste input geleverd voor deze kaarten. Zonder deze input en de gedeelde ervaringen uit de praktijk hadden de kaarten en deze handleiding niet tot stand kunnen komen.

Dit Docent Ontwikkel Team is opgezet en wordt ondersteund door:

Stichting DevLab Academy



NXP



Fontys Leraren Opleiding Tilburg



JetNet



CodeKids.nl



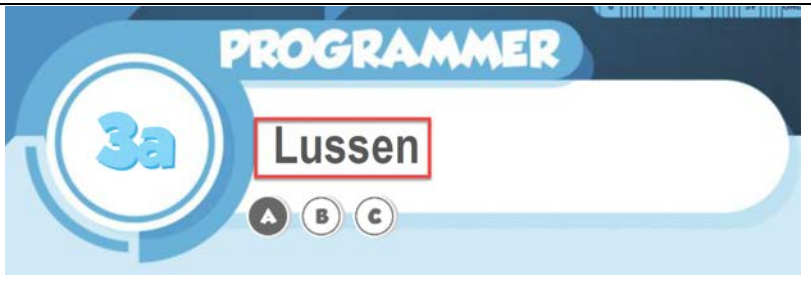
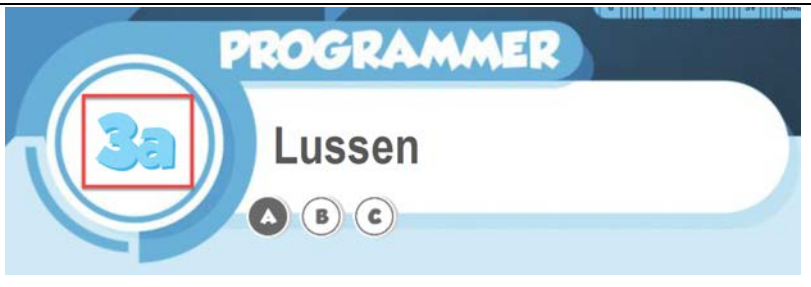
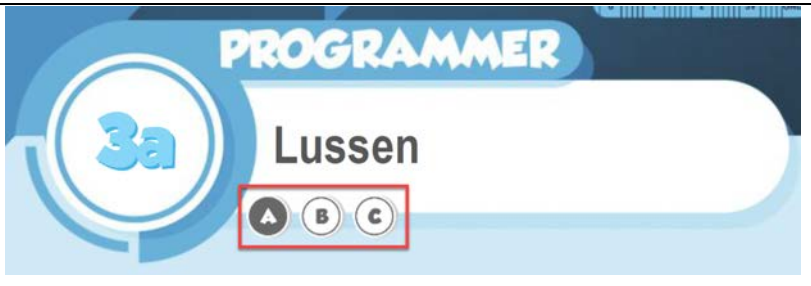
Correspondentie over de kaarten kan gericht worden aan:

- Lex van Gijssel, Stichting DevLab Academy, lex.van.gijssel@devlab.nl
- Chris Dorna, CodeKids.nl, chris@codekids.nl

Tip	In Bijlage 1 staat hoe je de kaarten zelf het beste kan afdrukken en afwerken. De kaarten zijn ook te downloaden en bestellen via de webshop van de micro:bit NL community: www.micro-bit.nl .
-----	--

De micro:bit kaarten

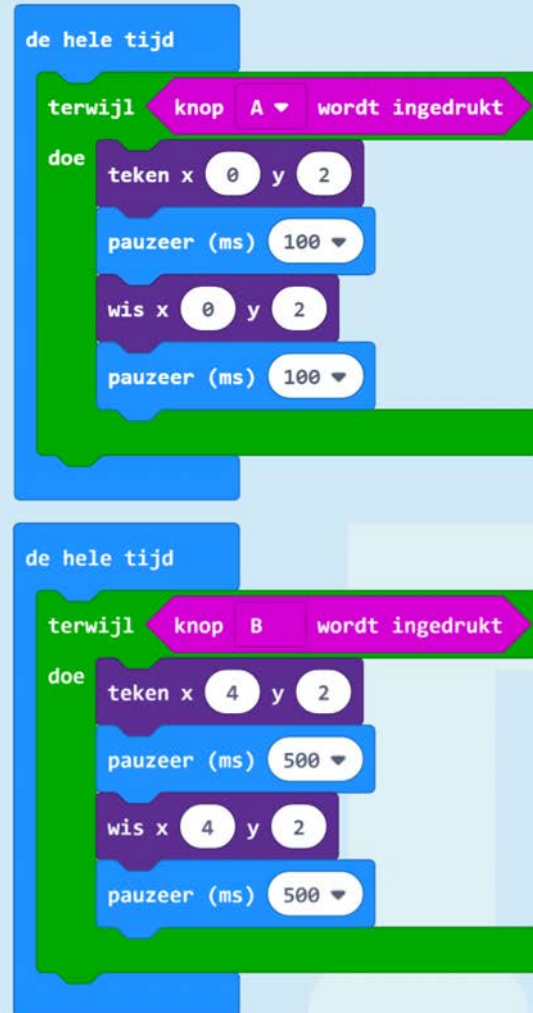
Onderdelen van iedere kaart

Titel van de kaart	
Nummer van de kaart	
Aantal kaarten over dit onderwerp	
Denk als een programmeur (Computational Thinking)	 Herhalen Een lus (loop) in een programma zorgt ervoor dat iets telkens wordt herhaald. Lussen komen in bijna ieder computerprogramma voor. Dit onderdeel komt alleen voor bij de Programmer kaarten en is gebaseerd op de Leerlijn Programmeren in het PO.
Titel van de opdracht	Multitasken Dankzij lussen lijkt het alsof de micro:bit twee dingen tegelijk doet.

Bouw het programma

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.

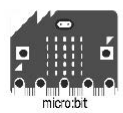
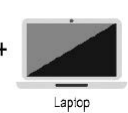


Ik snap het



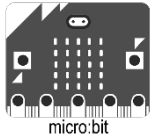
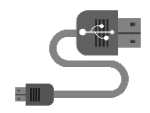
Dit zijn vragen/opdrachten die de leerling aan zichzelf kan stellen om te bepalen of hij/zij het snapt.

In deze handleiding vind je de antwoorden op de vragen.

Uitleg van blokjes	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 <i>waar (true)</i> als de knop is ingedrukt en <i>onwaar (false)</i> als de knop niet is ingedrukt. Als een blokje voor het eerst wordt gebruikt staat hier de uitleg van het blokje.
Dit heb je nodig	Dit heb je nodig  +  + 

Dit heb je nodig

Alhoewel je heel veel verschillende dingen op de micro:bit kunt aansluiten, hebben we ons voor de Explorer en de Programmer kaarten beperkt tot:

 micro:bit	 Koptelefoon met aansluitplug. Vaak kan je ook de oortjes van een mobiele telefoon gebruiken
 USB-snoertje	 Krokodilsnoertjes
 Laptop (met internetverbinding)	 USB-stick
 Led	

Werken met de micro:bit kaarten

Hieronder een paar overwegingen die wij hebben gemaakt bij het schrijven van de kaarten en wat praktische tips voor jouw micro:bit lessen.

Voor welke leeftijden?

De kaarten zijn voor leerlingen middenbouw PO, bovenbouw PO en onderbouw VO.

Taal

In de eerdere serie kaarten werden op de Programmer kaarten Engelstalige blokjes gebruikt. Naar aanleiding van de ervaringen en recent gepubliceerd onderzoek hebben we besloten om de Nederlandse blokjes te gebruiken op de Explorer én de Programmer kaarten.

JUNE 26, 2017 BY BENJAMIN MAKO HILL

Learning to Code in One's Own Language

I recently published a paper with [Sayamindu Dasgupta](#) that provides evidence in support of the idea that kids can learn to code more quickly when they are programming in their own language.

<https://mako.cc/copyrighteous/scratch-localization-and-learning>

Opstelling in de klas

- Door het lange formaat van de kaarten kunnen ze ook op relatief kleine schooltafels naast de laptop of het toetsenbord worden gelegd.
- Als je werkt met extra laptops zorg dan dat niemand over de aansluitsnoeren kan struikelen.
- Wij hebben gemerkt dat je als begeleider het prettigst werkt als de tafels in een U-vorm staan en je dus niet om de tafels hoeft te lopen om mee te kunnen kijken op het scherm van de leerlingen.

Meer leerlingen achter één computer

Het is goed mogelijk om twee leerlingen tegelijk met één laptop en één micro:bit te laten werken. Je hebt niet alleen minder apparatuur nodig, maar leerlingen leren ook van elkaar.

Een paar tips om het samenwerken nog beter te laten verlopen:

- Koppel leerlingen van gelijk niveau.
- Maak afspraken over het afwisselen van het bedienen van computer. Bijvoorbeeld na iedere micro:bit kaart wisselen.
 - Bij de weekendschool wordt ook gewerkt in koppels. Hierbij zijn op de computers twee muizen aangesloten en dit blijkt verrassend goed te werken.
- Laat leerlingen met elkaar overleggen over de vragen die op de kaarten staan.
- Zorg dat er ruimte genoeg is om te werken en er voldoende ruimte is tussen de verschillende koppels.

Laat leerlingen zo zelfstandig mogelijk werken

- Laat de leerlingen zelf de muis en het toetsenbord hanteren. Vertel wat ze moeten doen in plaats van het voor ze te doen. Ook als leerlingen elkaar helpen is dit een goede gewoonte.

Programmeerles geef je met de handen in de zakken of met je handen op de rug.

- Leren programmeren is vooral een kwestie van ontdekken en uitproberen. Het is geen enkel probleem (en zelfs de bedoeling) dat leerlingen afwijken van de opdrachten op de kaarten.
- Je zal merken dat sommige leerlingen heel erg snel gaan of met onverwachte oplossingen komen. Laat ze aan andere leerlingen hun ideeën en oplossingen vertellen.



Hoe lang duurt een les?

Dat is moeilijk te zeggen. Leerlingen kunnen de kaarten in hun eigen tempo doorwerken. Wij hebben gemerkt dat je rekening moet houden met een concentratieboog van 50 – 60 minuten, waarbij het goed is om iedere 20 minuten een breekpunt in te bouwen (bijvoorbeeld tussentijdse uitleg of elkaar laten vertellen hoe ver je bent).

Om ergonomische redenen is het belangrijk dat kinderen zich na ongeveer 45-60 minuten even kunnen bewegen.

Geluid aan of uit?

Één micro:bit die geluid maakt gaat nog wel, twee micro:bits gaat nog net, maar als het er meer worden dan ontstaat er een kakafonie die niet bijdraagt aan een goede sfeer om te ontdekken en te leren.

Op een deel van de kaarten staan opdrachten waarbij de micro:bit geluid maakt.

Net als met alle andere dingen in de klas waarbij de leerlingen 'eigen' geluid hebben, is het gebruik van hoofdtelefoons aan te bevelen. Deze hoeven niet duur te zijn. Gebruik bij voorkeur hoofdtelefoons met een volumeregelaar in het snoer.



Let op

De ene hoofdtelefoon klinkt harder dan de andere. Probeer het geluid daarom eerst zonder de hoofdtelefoon op de oren te zetten.

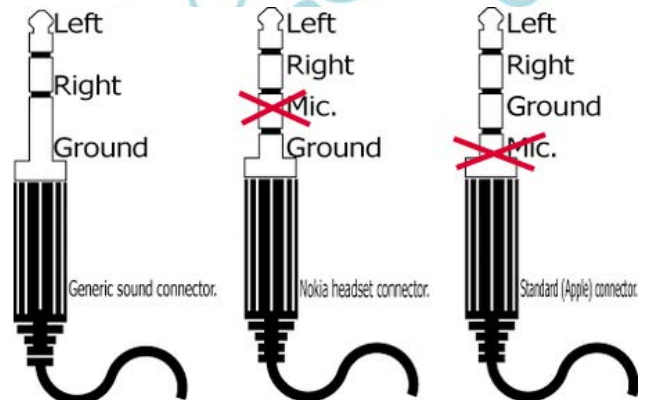




Oortjes mobiele telefoons

Ook de meeste oortjes van mobiele telefoons zijn bruikbaar.

De GND pin verbind je met de Ground van de telefoonplug en P0 met de Left of Right van de plug.



Nummering van de kaarten en de opdrachten

De volgorde van de kaarten is niet willekeurig en we adviseren om de kaarten in volgorde te doorlopen. Over het algemeen wordt een blokje uitgelegd op de eerste kaart waar het wordt gebruikt.

Werken met de micro:bit

Welke browser?

De kaarten zijn gemaakt op een Windows computer met Google Chrome als browser. De editor werkt ook in andere browsers en op andere computersystemen.

Als je werkt op een Apple computer dan ziet het downloaden en het kopiëren van bestanden er anders uit. Het pairen van een micro:bit met de browser browser Safari kan problemen opleveren.

Als je werkt met een Chromebook of een tablet (iPad) dan zie je op de kaarten Explorer 1d en 1e hoe je een programma overzet naar de micro:bit.

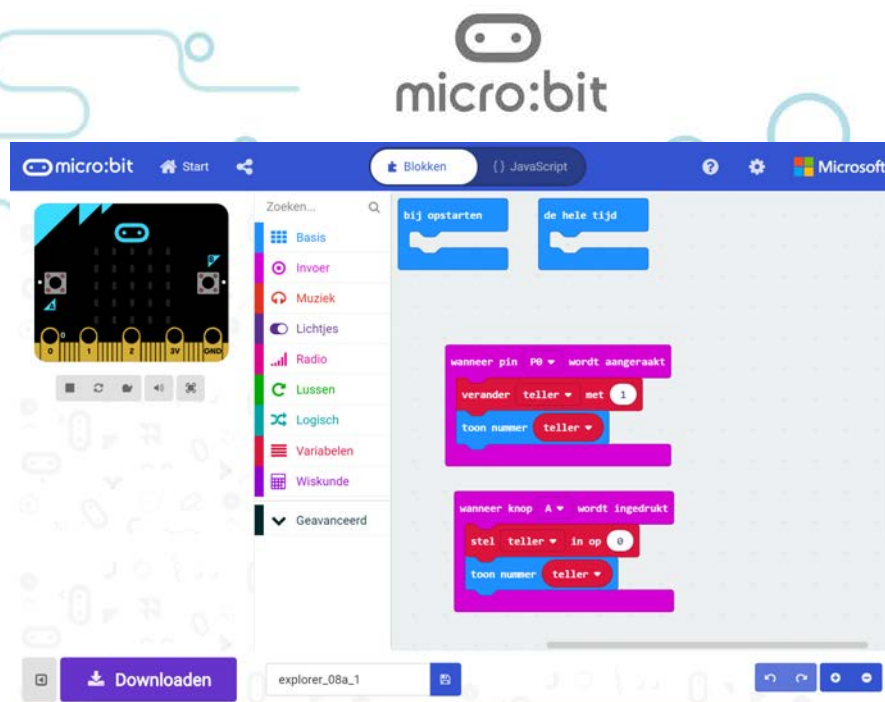
Op <https://makecode.microbit.org/browsers> vind je de eisen die aan het system worden gesteld.

Microsoft Makecode for micro:bit

De micro:bit kan met verschillende editors worden geprogrammeerd. De Explorer- en Programmerkaarten zijn gebaseerd op Microsoft MakeCode for micro:bit (versie 1): <https://makecode.microbit.org/>.

Eerdere versies

Op dit moment is versie 1 de meest actuele. Als je wilt werken met de eerdere versie (versie 0) dan kan dat via: <https://makecode.microbit.org/v0>

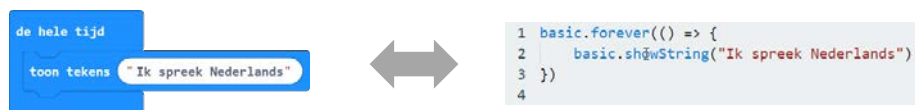


Microsoft?

MakeCode is volledig open source en de broncode wordt gedeeld via het Github platform. De Make Code ondersteunt naast de micro:bit ook verschillende andere systemen en programma's waaronder Minecraft en Lego® Mindstorms®.

Blokken / { } JavaScript

Je kan in de Editor eenvoudig omschakelen van het programmeren in blokken of in JavaScript.



Op de kaarten maken we gebruik van de blokeeditor. Op een enkele Programmer kaart maken de leerlingen voorzichtig kennis met de JavaScript editor.

USB-kabel

De standaard USB-kabel van de micro:bit is maar 15 cm lang. Zeker als je (voor de klas) iets wilt demonstreren is een langere kabel handig. Je kan hiervoor iedere micro usb-kabel gebruiken.

Let op: USB-kabels die worden meegeleverd met opladers of powerbanks zijn vaak alleen maar geschikt voor de voeding (stroom) en hebben niet de draadjes die nodig zijn om data te versturen.

Batterijen

De micro:bit kan zijn voeding halen uit de USB-aansluiting of uit een batterijhouder. Nu gaat het aansluiten van de batterijhouder niet echt makkelijk en heeft de houder die standaard wordt meegeleverd geen aan/uit schakelaar.

Bij een aantal opdrachten gaan de leerlingen met een geprogrammeerde micro:bit op pad. In dit geval gebruiken ze de (onhandige) batterijhouder of een USB-powerbank.



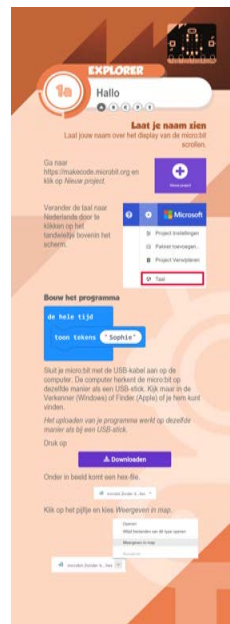


Let op: Sommige (grotere) powerbanks hebben een beveiliging die ervoor zorgt dat de stroomtoevoer wordt uitgeschakeld als de afgenomen stroom laag wordt (als de mobiele telefoon is opgeladen). Deze powerbanks zijn niet te gebruiken omdat de micro:bit heel weinig stroom gebruikt.

Explorer kaarten

De Explorer kaarten zijn bedoeld als eerste kennismaking met de micro:bit. Met deze kaarten maak je kennis met de mogelijkheden van de micro:bit. Het idee is dat leerlingen de voorbeelden nabouwen en hiermee verder spelen.

Bij het samenstellen van deze kaarten waren de verschillende mogelijkheden en de belangrijkste codeblokken het uitgangspunt.



Programmer kaarten

De Programmer kaarten hebben programmeerprincipes als uitgangspunt. We hebben hierbij gekeken naar het *Leerplankader Computational Thinking* van de SLO (voor PO en VO):

- <http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden/digitale-geletterdheid/computational-thinking/voorbeeldmatig-leerplankader/>

en de leerlijn *Programmeren in het PO*

- https://maken.wikiwijs.nl/74282/Programmeren_in_het_PO.

Deze kaarten beginnen met het onderdeel *Denken als een programmeur*. Hiermee kan een aansluiting worden gemaakt tussen het leren programmeren van de micro:bit en *Computational Thinking*.



Verband leerlijn Programmeren in het PO en de Programmer kaarten

Onderwerp Computational Thinking	Omschrijving	Expliciet genoemd op kaart
1. Algoritmes	Een reeks instructies , die wanneer deze stap voor stap in de juiste volgorde uitgevoerd worden, tot een vooraf vastgesteld doel leiden.	Programmer 2a
2. Decompositie	Een probleem onderzoeken en uitsplitsen in deelproblemen om het te kunnen oplossen. (Stappenplan met kleine deelproblemen en deeloplossingen)	Programmer 9a
3. Patronen	Patronen (model, vorm, kleur, voorbeeld, herhalingen) herkennen.	Programmer 6c
4. Herhaling	Een lus of loop (Engels) zorgt ervoor dat iets herhaald wordt.	Programmer 3a
5. Fouten	Fouten opsporen en fouten oplossen. Dit noemen we ook wel debuggen. (Probleemoplossend vermogen ondersteunen /ontwikkelen)	Programmer 2a
6. Voorwaarden	Soms moet aan een voorwaarde voldaan worden (of juist niet) voordat een stukje programma wordt uitgevoerd. Als...., dan	Programmer 4a
7. Abstractie	Het veralgemeniseren van verschillende problemen, door de verschillen weg te laten. Een oplossing vinden die bruikbaar is voor verschillende problemen door specifieke verschillen weg te laten.	Programmer 7a
8. Functie	Een "hulp"programma binnen het "hoofd"programma dat hergebruikt kan worden. Wordt ook wel <i>subroutine</i> genoemd. Bij de micro:bit wordt de naam <i>Functie</i> gebruikt.	Programmer 8a
9. Variabele	Een variabele is een waarde die varieert en moet worden opgeslagen om te kunnen vergelijken met andere waarden.	Programmer 1a
10. Representatie	Het weergeven en organiseren van gegevens in passende grafieken, lijsten, teksten of plaatjes.	Programmer 5a

Meer weten?

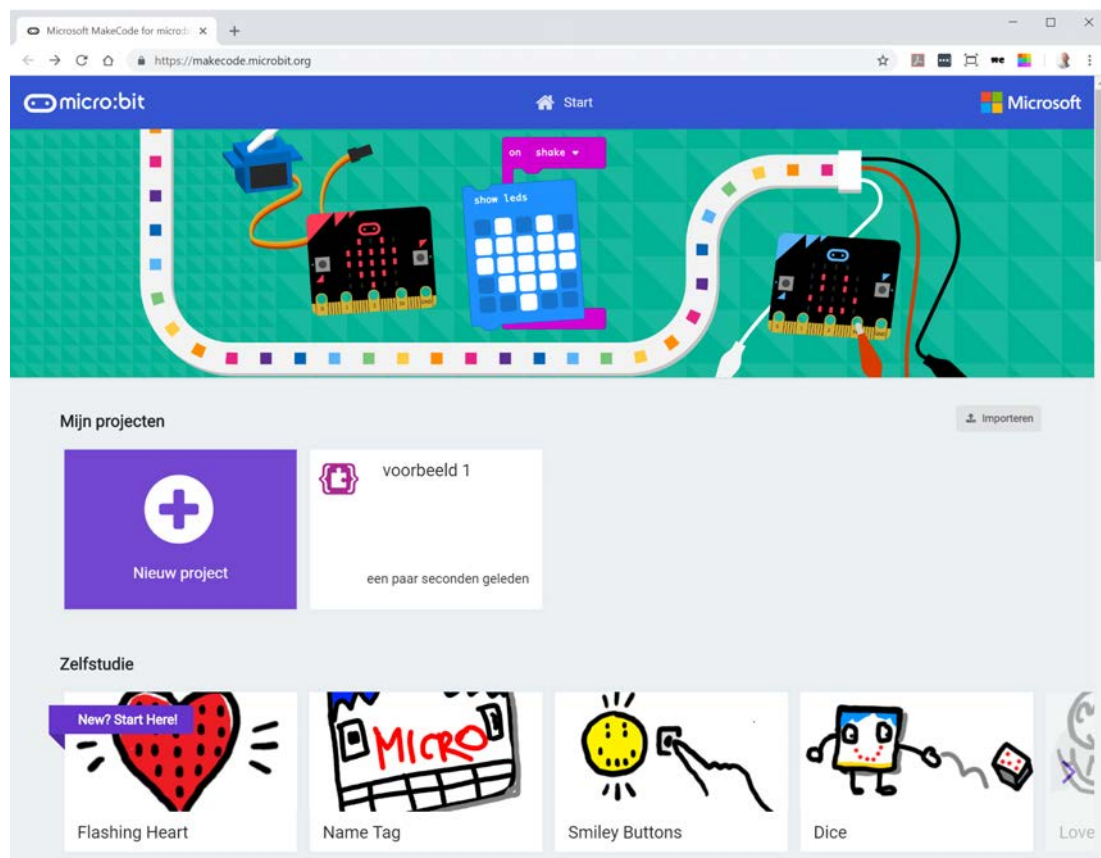
Voorbeelden

Via de knop Start heb je toegang tot tientallen projecten en voorbeelden.



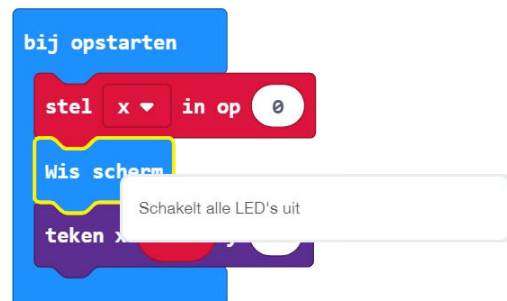
Startpagina

Op de startpagina staan de projecten die je zelf hebt gemaakt en verschillende voorbeelden en tutorials.



Helpfunctie

Als je met de muis even boven een blokje blijft hangen verschijnt er een korte uitleg over het blokje.



Als je meer wilt weten over de werking van een blokje klik er dan met je rechtermuisknop op en kies [Help](#).



Over DevLab en CodeKids

De Stichting DevLab Academy en CodeKids hebben als doel om techniek, en in dit geval programmeren en micro:bit op een duurzame manier te integreren in het onderwijs. Dit omdat we er in geloven dat deze manier van kennismaking met elektronica en programmeren van toegevoegde waarde is voor het onderwijs van vandaag en morgen.

In de toekomst zijn vaardigheden op het gebied van computational thinking niet meer weg te denken. DevLab en CodeKids vormen hiervoor samenwerkingsverbanden tussen bedrijven en instellingen die eenzelfde doel nastreven.

Zo werken we nauw samen met o.a. NXP, JetNet, Philips, ICT-group, Deloitte en Fontys Leraren Opleiding.

Door middel van die samenwerkingen worden activiteiten opgezet om het doel te realiseren. De kaartseries EXPLORER en PROGRAMMER zijn daar de concrete resultaten van, alsmede de organisatie van pilots met scholen, en de opzet van een train-de-trainer programma.

Explorer 1: Hallo

De eerste vijf kaarten gaan erover hoe je een programma op de micro:bit zet. Afhankelijk van de computer die je gebruikt werkt dit op verschillende manieren.

- Laptop met Windows
- Chromebook
- Tablet (iPad of Android)

Laptop met Windows

Mogelijkheid 1: hex-bestand naar micro:bit kopiëren (Explorer kaart 1a)

Vanuit de online editor (<https://makecode.microbit.org/>) wordt het programma met de knop Downloaden een hex-bestand gemaakt wat naar de map *Downloads* wordt geschreven. De micro:bit wordt via een USB-kabel aangesloten op de computer en het hex-bestand wordt vanuit de downloadmap naar de micro:bit gekopieerd.

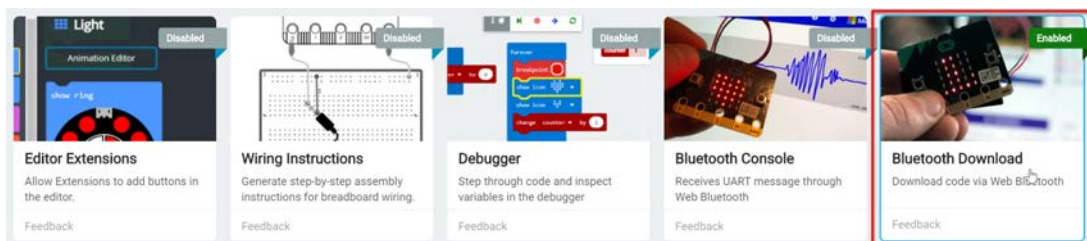
Mogelijkheid 2: direct via USB naar micro:bit schrijven (Explorer kaart 1c)

Het werken vanuit de downloadmap is voor veel mensen moeilijker dan je wellicht zou verwachten. Dit komt ook doordat het concept van bestanden, mappen etc. door het gebruik van tablets en smartphones minder bekend is.

Door de micro:bit via een USB-kabel aan te sluiten en vervolgens te koppelen (pairen) hoeft je alleen maar op de knop Downloaden te klikken. Het programma wordt daarna direct naar de micro:bit geschreven.

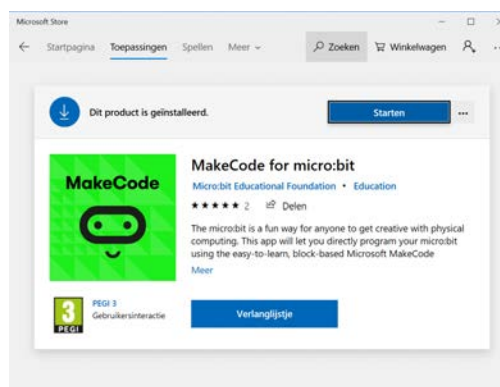
Mogelijkheid 3: via Bluetooth naar micro:bit schrijven (Nog niet beschikbaar)

Op het moment dat deze handleiding wordt geschreven kan je in de beta-versie van de editor experimenteren met Bluetooth Download waarbij je een programma via Bluetooth op de micro:bit zet:



Mogelijkheid 4: MakeCode app (geen kaart beschikbaar)

In de Microsoft Store kan je de gratis app *MakeCode for micro:bit* ophalen. Nadat deze is geïnstalleerd op de computer kan je ook zonder internetverbinding aan een programma werken. De editor is exact hetzelfde als de online versie en een programma wordt met één klik op de knop Downloaden op de micro:bit gezet.



De app werkt alleen onder Windows 10

Chromebook (Explorer kaart 1d)

Ook met een Chromebook kan je een micro:bit programmeren. Je doet dit vanuit de browser (<https://makecode.microbit.org/>). Het programmeren gaat op dezelfde manier als bij Windows en ook in de browser van Chromebook kan je een micro:bit via een USB-kabel pairen.

iPad (Explorer kaart 1e)

Voor de iPad (maar ook voor Android tablets) is er in de App Store een app beschikbaar waarmee je via Bluetooth een micro:bit kunt programmeren. De stappen voor het pairen en het overzetten van het programma blijken kinderen verrassend snel onder de knie te hebben.

Kaart 1a - Opdracht 1: Laat je naam zien

Het programma



https://makecode.microbit.org/_Jjc1VmPW4L54

Ik snap het

Ik kan uitleggen via welke stappen ik een programma maak en dit op de micro:bit zet.

Antwoord

1. Start de editor
2. Programma bouwen in de editor
3. Hex-file downloaden in de editor
4. Hex-file kopiëren naar micro:bit (zorg er voor dat micro:bit via een USB kabel op computer is aangesloten)

Kaart 1b - Opdracht 2: Klaar voor de start

Het programma



https://makecode.microbit.org/_Uqjb7JETpchD

Ik snap het nog steeds

Ik kan uitleggen waarom ik de tekst nu maar één keer zie.

Antwoord

Het blok *bij opstarten* wordt eenmalig uitgevoerd als het programma wordt gestart. Dit kan op de volgende momenten gebeuren:

1. Direct nadat het programma vanaf de computer naar de micro:bit is gekopieerd.
2. Als je op de resetknop achterop de micro:bit drukt.
3. Als de micro:bit op een spanningsbron (via USB-kabel of batterij) wordt aangesloten.

Ik snap het nog beter



Ik weet...

...nog een manier om de tekst nog een keer te laten zien.

Antwoord

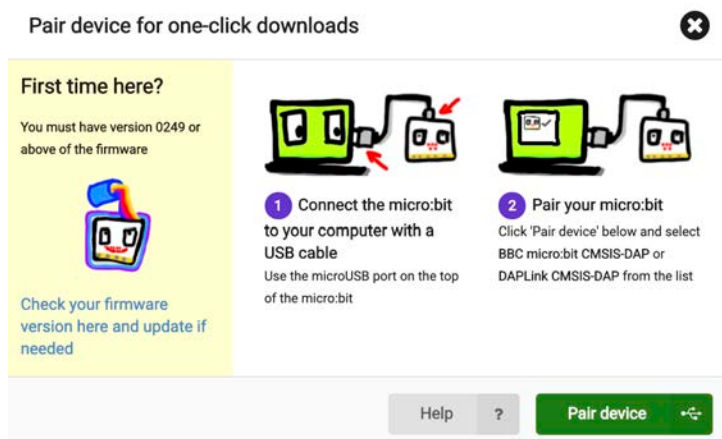
Je kan het blok *bij opstarten* dus opnieuw starten door de USB-kabel of de batterij even los te nemen en weer aan te sluiten.

Kaart 1c – micro:bit pairen

Door de micro:bit te pairen (koppelen) wordt het programma direct naar de micro:bit geschreven als je in de editor op de button *Downloaden* klikt.

Het is hiervoor wel noodzakelijk dat de juiste firmware op de micro:bit staat. Zeker als je les geeft aan meer kinderen is het handig om vooraf te controleren of de juiste firmware op de micro:bits staat.

Klik op bij *instellingen* op *pair device*. Je krijgt dan een pop-up scherm om de op de USB aangesloten micro:bit te kunnen pairen.



In de linkerkolom van de pop-up kun je klikken op *Check your firmware version here and update if necessary*. Je krijgt dan een (Engelstalige) handleiding die je uitlegt hoe je de firmware versie van je micro:bit kunt controleren, en hoe je die kunt updaten als dat nodig is. Het is een simpele actie die per micro:bit niet meer dan 1-2 minuten duurt.

Kaart 1d – Ik heb een Chromebook

Een programma



<https://makecode.microbit.org/KV7UL8W9HUf3>

Chromebooks worden steeds populairder, ook in het onderwijs. De ontwikkelaars van MakeCode (de omgeving waarin je het programma voor de micro:bit schrijft) hebben er voor gezorgd dat de micro:bit ook prima vanaf een Chromebook kan worden geprogrammeerd.

Ook op een Chromebook kan de micro:bit worden gepaired zodat leerlingen niet met hex-bestanden hoeven te slepen.

Kaart 1d is in feite overbodig. Alleen als het paren om de een of andere reden (beveiliging) niet mogelijk is zal het hex-bestand naar de micro:bit moeten worden gesleept. Op deze kaart staat hoe dat werkt.

Kaart 1e – Ik heb een tablet

Een programma



[https://makecode.microbit.org/ KV7UL8W9HUf3](https://makecode.microbit.org/KV7UL8W9HUf3)

Ook met een iPad of Android tablet kan je de micro:bit programmeren. Als je kaart 1e voor het eerst ziet dan schrikt het aantal knoppen en de stappen misschien even af. Als je de stappen een keertje hebt doorlopen dan blijkt het programmeren eigenlijk verrassend makkelijk te gaan.

Ook als een hele klas aan het werk is met verschillende micro:bits die via Bluetooth worden geprogrammeerd dan gaat dit prima en zitten ze elkaar niet in de weg.

Een voordeel van Bluetooth is dat er geen USB-kabel nodig is. Zeker als je bijvoorbeeld micro:bit autootjes gaat programmeren, is dit erg handig.

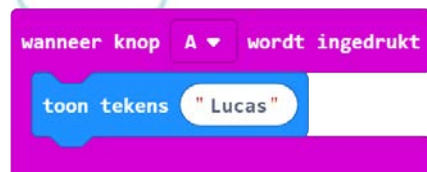
Voorbereiding van een les

- Controleer of de laatste versie van de firmware op de micro:bits staat.
- Zorg dat de micro:bit app op de tablets is geïnstalleerd. Mogelijk moet dit gedaan worden door de IT-afdeling van jouw school.
- Probeer het eerst zelf.
- Als je de micro:bit app in de klas wilt demonstreren, zorg er dan ook voor dat je het scherm van je tablet kunt delen via het digitale schoolbord.

Explorer 2: Knoppen

Kaart 2a - Opdracht 1: Laat mijn naam zien als ik op knop A druk

Het programma



https://makecode.microbit.org/_bRb3R7Y1q69L

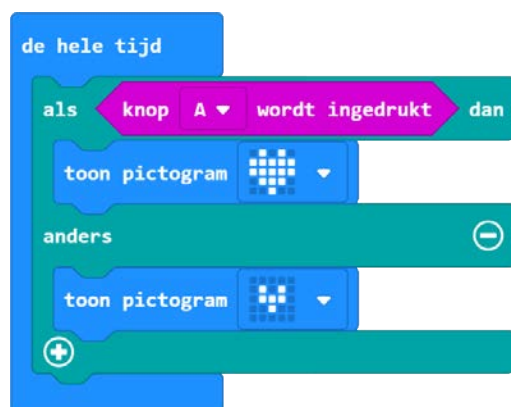
Ik snap het

Ik weet wat er gebeurt als ik nog een keer op de knop druk terwijl de tekst nog voorbij komt.

Antwoord

Bij het blok *wanneer knop A wordt ingedrukt* houdt de micro:bit bij hoe vaak je op de knop drukt. Dit noemen we ook wel *bufferen*. Als je op de knop drukt terwijl de tekst nog loopt dan wordt de tekst nog een keer getoond. In totaal net zo vaak als je op de knop hebt gedrukt.

Hieronder zie je een andere manier om te kijken of een knop is ingedrukt (*knop A wordt ingedrukt*). Hierbij wordt er niet gebufferd. Er wordt alleen op dat moment gecontroleerd of de knop is ingedrukt of niet.



https://makecode.microbit.org/_AgT4PpVriF2D

Kaart 2a - Opdracht 2: Druk op alle knoppen

Het programma

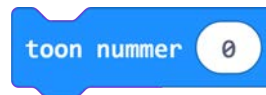


https://makecode.microbit.org/_gkFReckEt8jw



Ik snap het

Ik snap het verschil tussen deze blokken:



Antwoord

Voor een computer is het belangrijk om te weten met wat voor soort gegevens we te maken hebben. We noemen dit ook wel *typecasting*.

- Een groep tekens noemen we een string en staan altijd tussen dubbele aanhalingstekens "...".
- Een groep cijfers is een nummer. Met nummers kunnen we bijvoorbeeld rekenen of een teller bouwen. Je kan van nummers ook zeggen of het ene groter is dan een ander nummer.

In een string kunnen ook cijfertekens voorkomen ("Sofie heeft 2 broers"), maar in dat geval wordt het cijfer niet gezien als een nummer.

Leervraag

Met deze twee blokjes komt er een nul (0) op het display te staan. Wat is het verschil



Ik snap het

Ik snap waarom er (nog) niets gebeurt na het downloaden.

Antwoord

Het programma wacht tot er op een knop wordt gedrukt

Explorer 3: Tellen

Kaart 3a - Opdracht 1: Tellen met de micro:bit

Het programma

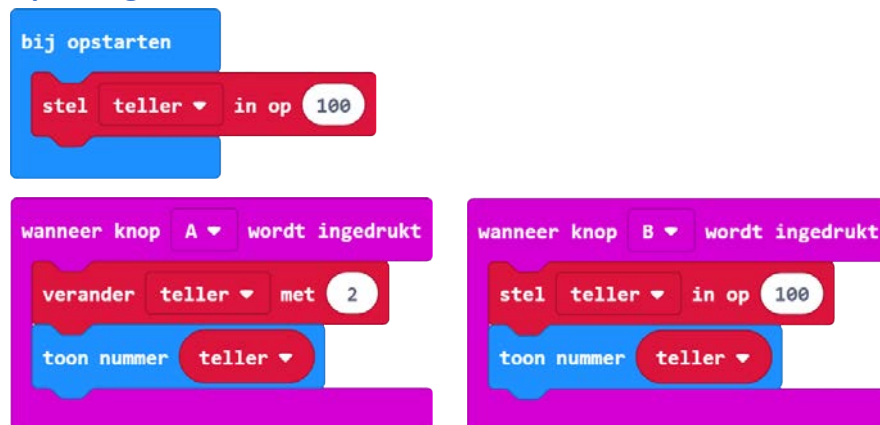


https://makecode.microbit.org/_gKa9LvX0s4WJ

Ik snap het

Ik kan het programma zo aanpassen dat de teller begint bij 100 en er iedere keer 2 bij wordt opgeteld.

Oplossing



https://makecode.microbit.org/_K2EeCmYbhe09

Ik snap het helemaal

Ik kan het programma zo aanpassen dat ik met knop B kan aftellen en de teller met de knoppen A+B weer op 0 kan zetten.

Oplossing



https://makecode.microbit.org/_bJOPiJ8TFJmc

Kaart 3b - Opdracht 2: Stappen tellen

Het programma



https://makecode.microbit.org/_9eVTrtifKg9e

Ik snap het

Ik kan het programma zo aanpassen dat als ik op knop B druk niet 0, maar een "x" op het scherm komt.

Antwoord

Hierbij moet in het 2^e deel van het programma het blokje *toon nummer* worden vervangen door een blokje *toon tekens*. Als je in het blokje toon nummer een "x" probeert in te vullen dan zal dit niet gaan omdat de letter x een ander type is dan het getal 0.



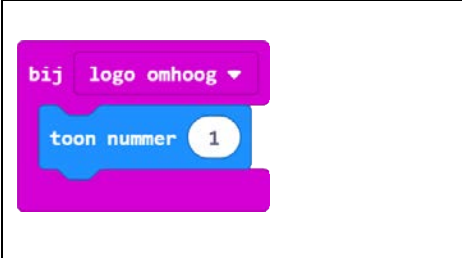
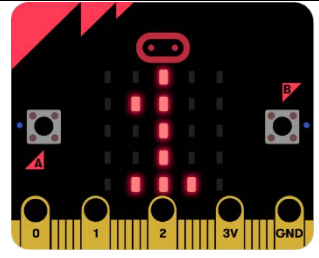
https://makecode.microbit.org/_1Vt70Xfai4sV

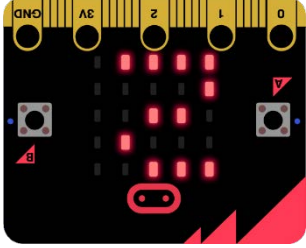
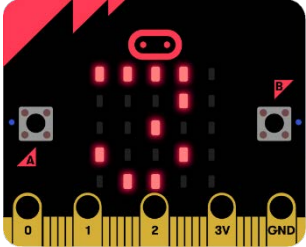
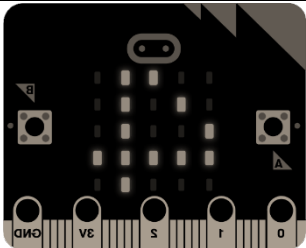
Ik snap nog meer

Ik weet het verschil tussen logo omhoog, logo omlaag, scherm naar beneden en scherm omhoog.

Antwoord

Dit kan je eigenlijk alleen maar goed uitleggen als je de verschillende mogelijkheden op de micro:bit laat zien. Dat kan met dit programma.

		Rechtop staand op tafel
---	--	--------------------------------

bij logo naar beneden ▾ toon nummer 2		Op zijn kop staand op tafel
bij scherm omhoog ▾ toon nummer 3		Met rug liggend op tafel
bij scherm naar beneden ▾ toon nummer 4		Op zijn kop liggend op tafel

<https://makecode.microbit.org/4ikCJkgibVfT>

Explorer 4: Opslaan, importeren en delen

Over deze kaarten

Deze kaarten gaan niet over het programmeren maar over de manier waarop je programmabestanden opslaat en deze kunt delen met anderen.

Bij de eerste twee opdrachten wordt een beroep gedaan op ICT basisvaardigheden. In de praktijk blijkt dat veel leerlingen niet goed weten wat precies een bestand is, hoe deze op een computer worden opgeslagen en hoe je bestanden kunt kopiëren.

Voor de derde opdracht (geheime boodschap op internet opslaan) geeft de mogelijkheid om met leerlingen internetveiligheid (mediawijsheid) te bespreken.

Deze handleiding gaat niet verder in op ICT basisvaardigheden en Mediawijsheid. De site van de Stichting Leerplan Ontwikkeling (SLO) is een goed startpunt als je hier meer over wilt weten: <http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden/digitale-geletterdheid>. Ook Kennisnet heeft een mooie startpagina: <https://www.kennisnet.nl/artikel/overzicht-leerlijnen-digitale-geletterdheid>.

Kaart 4A - Opdracht 1: Geheime boodschap opslaan

Het programma



https://makecode.microbit.org/_14aav1PD38F5

Over het HEX-bestand

De editor genereert een HEX-bestand. Hierin zit de code die naar de micro:bit wordt gedownload, maar ook het programma dat in de editor bewerkt kan worden.

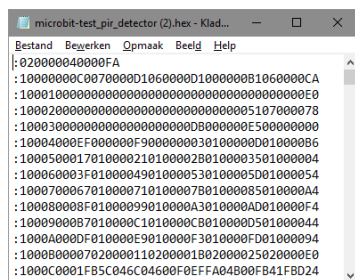
Kaart 4a - Opdracht 2: Geheime boodschap openen

Als je het echt wilt weten

De letters HEX staan voor hexadecimaal. Dit is een voorstelling van getallen die uit gaat van 16 verschillende karakters:

0 1 2 3 4 5 6 7 8 9 A B C D E F

Als je een hex-bestand (per ongeluk) opent met het Windows kladblok dan zie je dit:



Je kan een hex-bestand alleen bewerken met de JavaScript Blockeditor.

Kaart 4b - Opdracht 3: Geheime boodschap op internet opslaan

Ik snap het

Ik kan uitleggen wie er allemaal bij een programma kunnen dat ik via internet heb gedeeld.

Antwoord

Als je een programma hebt gedeeld dan kunnen alleen mensen die de hyperlink van jou hebben gekregen het programma openen.

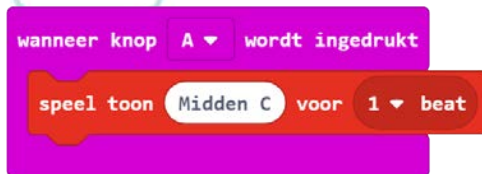
De hyperlink kan natuurlijk weer worden doorgestuurd zodat nog meer mensen jouw programma kunnen openen. Zoals ook de links in deze handleiding onder elk programma.

Als je een programma aanpast en je wilt dit weer delen dan moet je een nieuwe hyperlink maken.

Explorer 5: Herrie maken

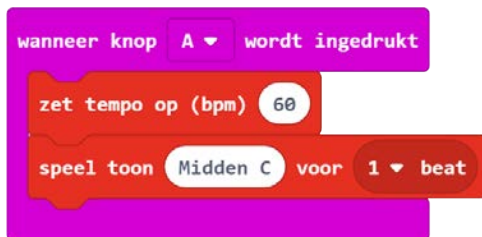
Kaart 5a - Opdracht 1: Speel één toon

Het programma



https://makecode.microbit.org/_f9hfjwHqiYz2

Probeer dit ook



https://makecode.microbit.org/_g9C2xcR76P6m

Ik snap het

Duurt de toon korter of langer als je een hogere waarde voor bpm invoert?

Antwoord

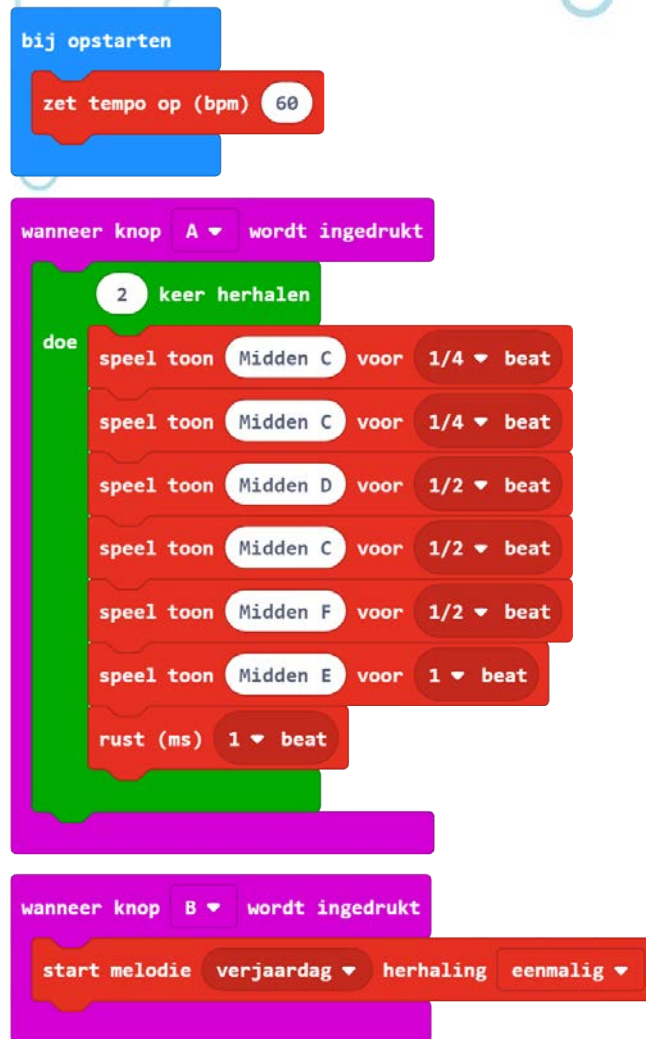
Hoe meer beats er in een minuut worden afgespeeld, hoe korter ze zijn. De toon klinkt dus korter.

Bijvoorbeeld:

BPM	Duur van één beat
60	1 seconde
120	0,5 seconde

Kaart 5b - Opdracht: Speel een liedje

Het programma



https://makecode.microbit.org/_dMAi1LYz74pX

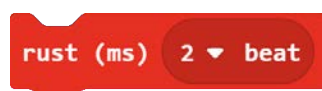
Ik snap het

Je hebt ook het blokje *pauzeer*. In dit blokje vul je in hoe lang het programma moet pauzeren.



Dit doe je in milliseconden. Eén seconde (s) bestaat uit 1000 milliseconden (ms).

Ik kan uitleggen wat het verschil is met het blokje



Antwoord

Het blokje *pauzeer* houdt geen rekening met de maat (beat) van de muziek. Voor het inbouwen van een pauze in de muziek is dit blokje dus minder geschikt.

Ik snap lussen



https://makecode.microbit.org/_gTJPT5id8MiV

Ik kan uitrekenen hoe vaak de toon wordt gespeeld als ik op knop A klik.

Antwoord

Hier is een herhaalblokje opgenomen in een ander herhaalblokje. Programmeurs noemen dat *nesten*.

De toon wordt $2 \times 4 = 8$ keer afgespeeld.

Explorer 6: Sensoren

Kaart 6a - Opdracht 1: Maak muziek met licht

Het programma



https://makecode.microbit.org/_3y9cpoayTHRp

Ik snap het

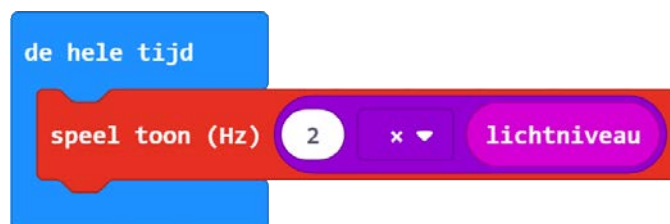
Ik kan uitleggen waarom de toon hoger wordt als er meer licht op de micro:bit valt.

Antwoord

Als er meer licht op de sensor valt dan wordt het gemeten lichtniveau ook groter. Hiermee wordt ook de frequentie (toon) hoger.

Kaart 6a - Opdracht 2: Toontje hoger

Het programma



https://makecode.microbit.org/_ehUJxH8wd74p

Vraag

Wat is de laagste toon en wat is de hoogste toon?

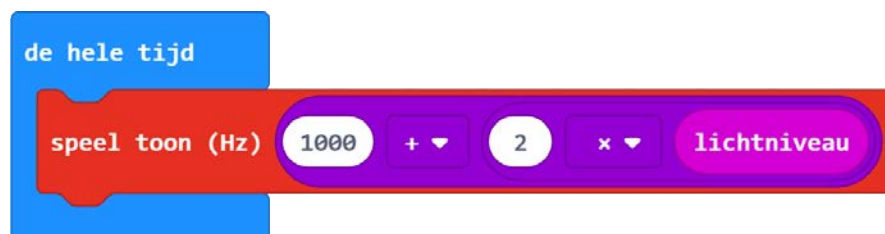
Antwoord

De laagste frequentie is 0 Hertz (het laagste lichtniveau is 0 en $2 \times 0 = 0$)

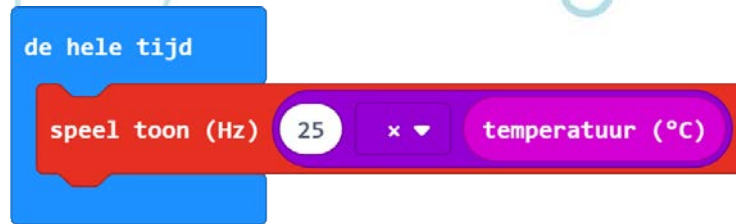
De hoogste toon is 2×255 (het hoogste lichtniveau) = 510 Hertz

Suggestie

Om de laagste toon te 'verleggen' naar 1000 Hertz kan je twee rekenblokjes combineren:



https://makecode.microbit.org/_P5aAcaJh7KpM

Kaart 6b - Opdracht 3: Horen hoe warm het is**Het programma**

https://makecode.microbit.org/_2ohfWeb5TVUL

Vraag

Waarom wordt de temperatuur vermenigvuldigd met 25?

Antwoord

Het temperatuurverschil is maar enkele graden en het hoorbare verschil in toonhoogte is hierdoor ook erg klein. Door de gemeten temperatuur met 25 te vermenigvuldigen wordt het verschil in toonhoogte ook groter.

Stel dat de omgevingstemperatuur verandert van 20 °C naar 23 °C dan is het verschil in frequentie maar 3 Hertz.

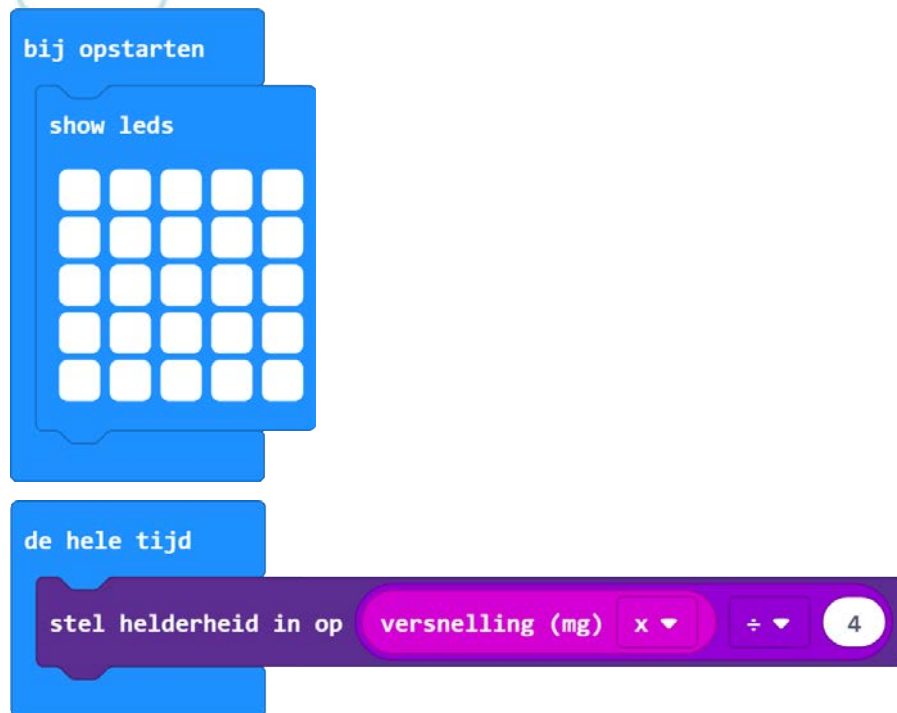
Door de temperatuur te vermenigvuldigen wordt het verschil 75 Hz:

$$(25 \times 23) - (25 \times 20) = 575 - 500 = 75 \text{ Hz}$$

Explorer 7: Beweging

Kaart 7a - Opdracht 1: Kantel het licht

Het programma



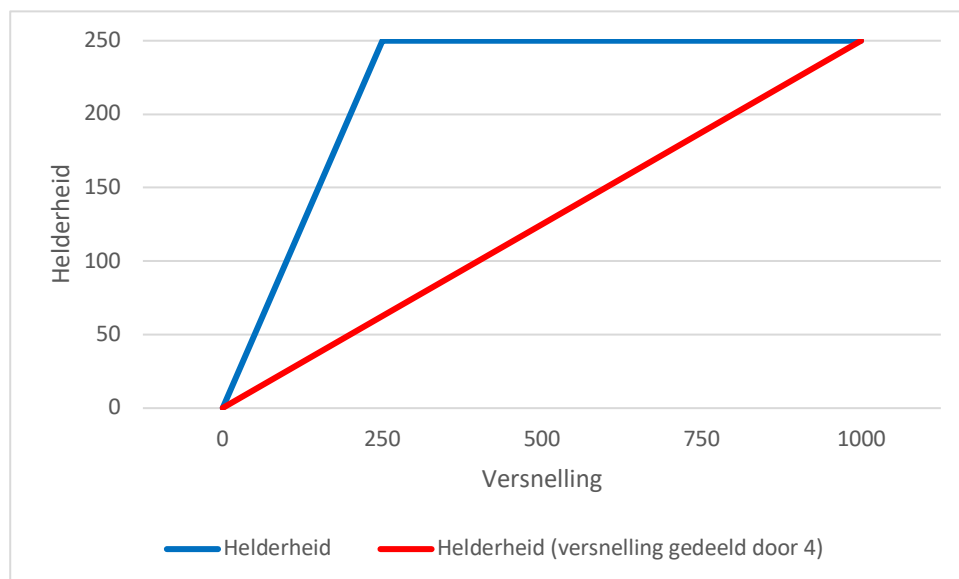
https://makecode.microbit.org/_hmmEfv0m8Rc6

Ik snap het

Waarom wordt de versnelling gedeeld door 4?

Antwoord

De maximale waarde van de versnelling is 1023 en de maximaal in te stellen helderheid is 255. Door de versnelling te delen door 4 zorgen we ervoor dat de helderheid niet alleen verandert tussen 0 en 255, maar over het hele bereik van 0 tot bijna 1023.



Probeer dit ook eens



https://makecode.microbit.org/_85efRbFJzix3

Reactie toevoegen

Als je met de rechtermuisknop op een blokje klikt dan zie je dit menu:



Met *Reactie toevoegen* kan je eigen commentaar aan een blokje toevoegen. Hiermee heb je als programmeur de mogelijkheid om jouw programma te documenteren (van uitleg en commentaar te voorzien).

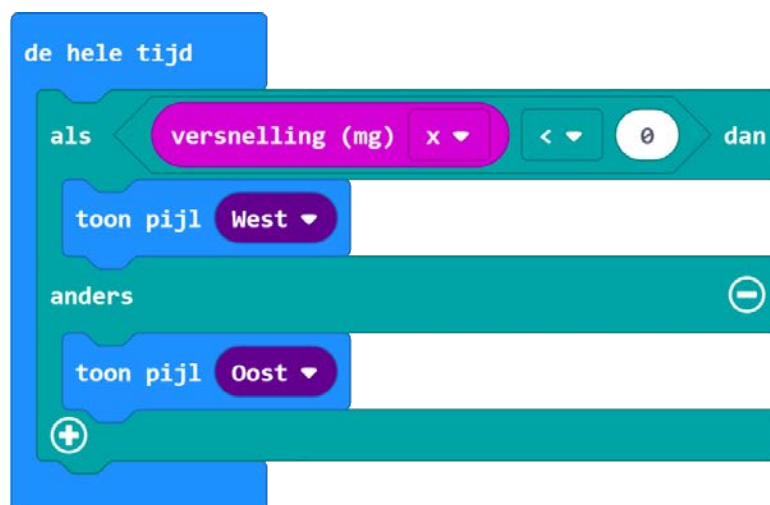
Aan een vierkant tekstballonnetje in het blokje kan je zien dat het blokje een reactie heeft.



https://makecode.microbit.org/_R9XYKwfEKVdd

Kaart 7b - Opdracht 2: Van links naar rechts

Het programma

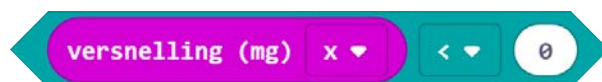


https://makecode.microbit.org/_iVDcT298kfHu

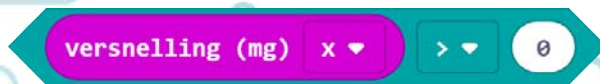
Vraag

Wat moet je veranderen aan het programma als je het < teken (kleiner dan) in de voorwaarde verandert in een > teken (groter dan)?

De voorwaarde is nu:

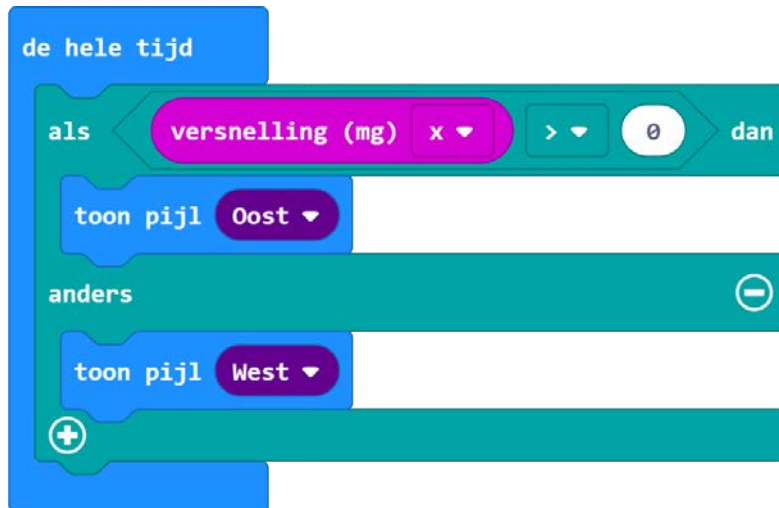


En dit wordt:



Antwoord

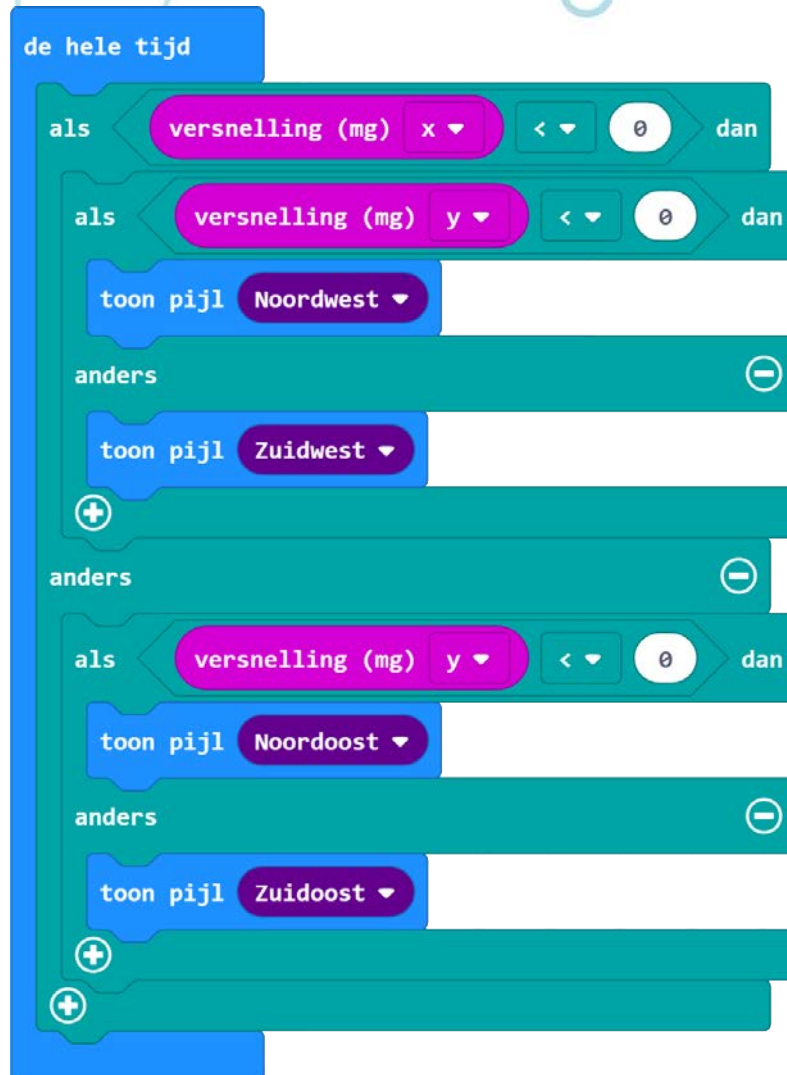
De blokjes tussen *als* en *anders* moeten worden verwisseld. Het programma ziet er dan zo uit.



https://makecode.microbit.org/_HoTWobl1PKLL

Kaart 7b - Opdracht 3: Van links naar rechts en van voor naar achter

Het programma



https://makecode.microbit.org/_EPa0fyVdqLYz

Achtergrondinformatie

Bij deze opdracht worden verschillende als-dan-anders blokken in elkaar gezet (genest).

Ik snap het

Dit programma is te groot om in één keer op het scherm te laten zien (tenzij je een heel groot beeldscherm hebt). Wat kan je doen om het toch helemaal te tonen?

Antwoord



Rechts onderin het scherm staan knoppen waarmee je kan in- en uitzoomen

Explorer 8: Pinnen

Over het aansluiten van dingen op de micro:bit

Bij de eerste opdracht heb je nog geen extra onderdelen nodig. Bij de tweede opdracht gaan de leerlingen een ledje aansluiten met krokodilsnoertjes.

Deze snoertjes zijn er in veel verschillende kwaliteiten. Op scholen treffen we helaas vaak de slechtste (goedkope) kwaliteit aan:

- Voor leerlingen onhandelbaar vanwege een 'glibberige' buitenmantel.
- Slecht of geen contact omdat de kern van het snoer slecht op de krokodilklem is geklemd.

Test de snoertjes daarom vooraf en gooi onbetrouwbare snoertjes weg.

Kaart 8a - Opdracht 1: Raak me (niet) aan

Het programma



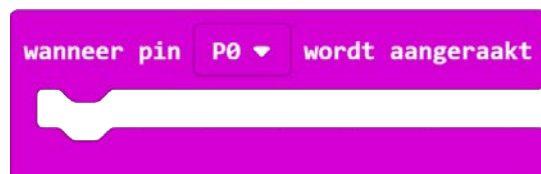
<https://makecode.microbit.org/UtdVyTeFdPpJ>

Ik snap het

Ik kan uitleggen of *P0* hier als input of als output wordt gebruikt.

Antwoord

Pin 0 wordt hier als input gebruikt. Het blokje *wanneer pin P0 wordt aangeraakt* is te vinden in de groep *Invoer*.



Let op

Pin 0 is ook de standaardaansluiting voor een hoofdtelefoon. Het volgende programma werkt daarom dus niet goed:



<https://makecode.microbit.org/JfRgFgTpAAqs>

De oplossing is om P1 te gebruiken voor het aanraken en P0 voor de hoofdtelefoon.

ICT Basisvaardigheden

Bij computers heb je altijd te maken met invoer- en uitvoerapparaten.

- Bekende invoerapparaten zijn een computermuis en toetsenbord.
- Bekende uitvoerapparaten zijn bijvoorbeeld beeldschermen en printers.

Er zijn ook steeds meer gecombineerde apparaten, denk aan een touchscreen of een printer die ook als scanner gebruikt kan worden.

Kaart 8b - Opdracht 2: Knipperled

Het programma



<https://makecode.microbit.org/6y6TmidME3Ck>

Over leds

Leds hebben een lange en een korte aansluiting. De lange aansluiting is de plus (Anode) en de korte aansluiting de min (Kathode).

Bij veel opdrachten wordt een weerstand gebruikt om de stroom door de led te beperken. Strikt genomen is dit nodig, maar in de praktijk gaat het bij de spanning die de micro:bit afgeeft (3 Volt) ook prima zonder deze weerstand.

Veel gemaakte fouten/problemen bij het aansluiten

- Min en plus verwisseld.
- Krokodilklem maakt niet goed contact met de led.



- Krokodilklem maakt niet goed contact met de micro:bit.
- Krokodilklemmen maken onderling kortsluiting.
- Led is stuk (zie [Led testen](#) hieronder).
- Krokodilsnoertje is stuk (zie begin van dit hoofdstuk). Je kan snoertjes testen met de opdracht hiervoor ([Raak me \(niet\) aan](#))

Led testen

Als je niet zeker weet of een led (nog) goed is kan je de lange draad tegen de **3V** pin en de korte draad tegen de **GND** pin houden terwijl de micro:bit voeding via een batterij of de USB-aansluiting.

Gewone lampjes?

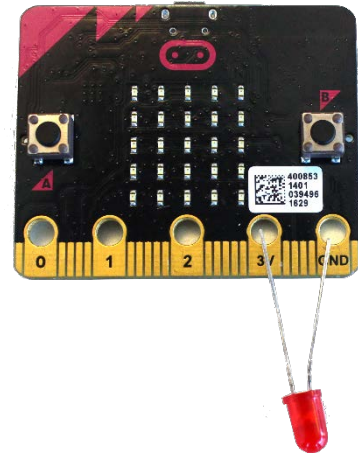
Veel scholen hebben een voorraad gloeilampjes. Omdat deze vrij veel stroom nodig hebben, kan je ze niet gebruiken op de micro:bit.

Ik snap het

Ik kan uitleggen of **P0** hier als input of als output wordt gebruikt.

Antwoord

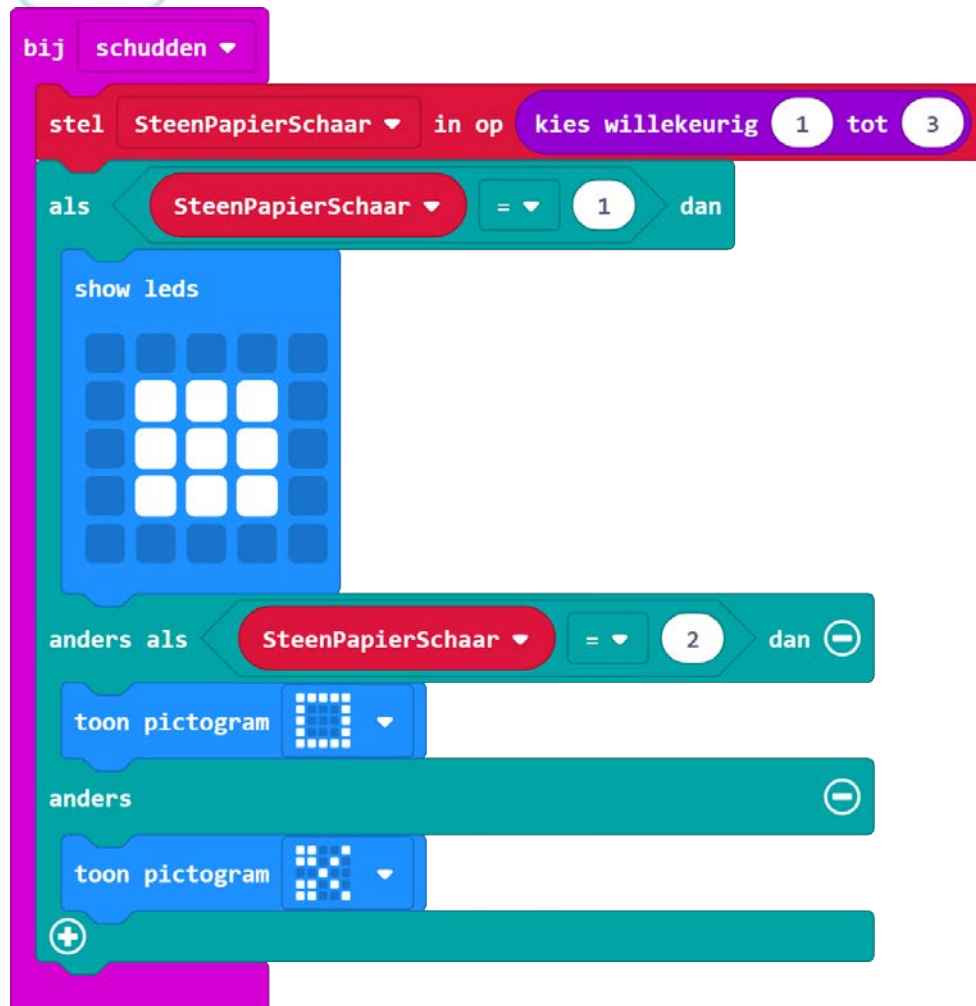
Als output



Explorer 9: Getallen

Kaart 9a - Opdracht 1: Speel papier-steen-schaar

Het programma



https://makecode.microbit.org/_Pvc4V2cEwMv5

Ik snap het

Ik weet welke waarden ik moet invullen als ik een dobbelsteen wil maken.

De laagste waarde is dan 1 en de hoogste 6.



Taalkundig dingetje

In het Nederlands is er een verschil tussen *tot* en *tot en met*. Strikt taalkundig gezien zou in het programma hierboven de waarde 6 niet voor kunnen komen.

Programmer 1: Variabelen en rekenen

Kaart 1a - Opdracht 1: Secondeteller

Opdracht

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

Oplossing



https://makecode.microbit.org/_Ji5bzg5bgA8d

Ik snap het

Ik kan dit programma zo aanpassen dat het begint te tellen bij 2 en daarna alleen even getallen laat zien (0, 2, 4 etc.)

Oplossing



https://makecode.microbit.org/_X4AFsT9xKH2A

Ik snap het

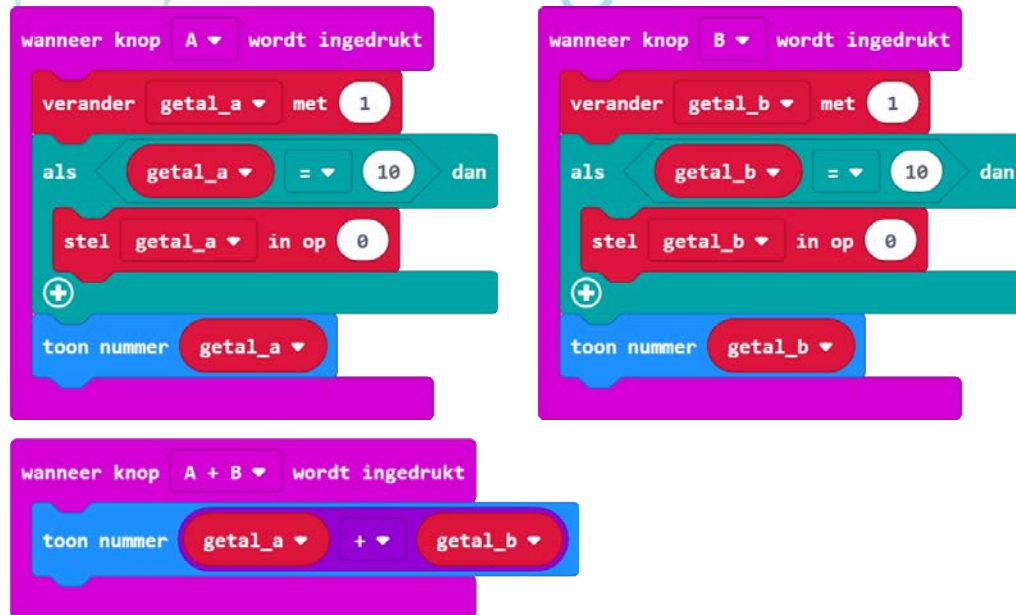
Ik kan vertellen wat er met de naam van de variabelen in het project gebeurt.

Antwoord

Als je de naam van een variabele verandert dan wordt deze verandering in het hele project doorgevoerd.

Kaart 1b - Opdracht 2: Rekenmachine

Het programma



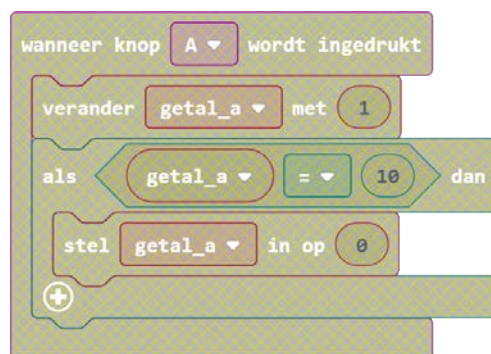
https://makecode.microbit.org/_XHkgqEX2pD5o

Achtergrond: Namen van variabelen

- Als in een programma veel verschillende variabelen worden gebruikt dan is het belangrijk dat leerlingen leren om variabelen een logische naam te geven.
- Voorkom spaties in de namen van variabelen. De JavaScript editor accepteert deze spaties niet.
- De naam van een variabele mag niet met een cijfer beginnen.
- Programmeurs gebruiken bij de namen van variabelen hoofdletters en kleine letters *getal_a*, zal door veel programmeurs als *getalA* worden geschreven.

Ik snap het 1

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



Antwoord

Er was al een blok *on button A pressed*. Stel dat je twee van deze blokken zou hebben dan weet de micro:bit niet welke code uitgevoerd moet worden als op knop A wordt gedrukt.



Het blok *de hele tijd* kan je wel vaker in één programma gebruiken.

Ik snap het 2

- In plaats van optellen kan ik de getallen ook vermenigvuldigen, aftrekken en delen.

Oplossing

Dit doe je door de operator in het rekenblokje aan te passen



Operator	Teken	Voorbeeld
Optellen	+	$6 + 2 = 8$
Aftrekken	-	$6 - 2 = 4$
Vermenigvuldigen	x	$6 \times 2 = 12$
Delen	÷	$6 / 3 = 2$
Machtsverheffen	**	$6 ** 2 = 6^2 = 36$

Over machtsverheffen

Bij machtsverheffen wordt een getal een opgegeven aantal keer met zichzelf vermenigvuldigd. Er zijn verschillende manieren om dit op te schrijven:

3 tot de macht 2	$3^2 = 9$	$3 \wedge 2 = 9$	$3 \times 3 = 9$	$3 ** 2 = 9$
3 tot de macht 3	$3^3 = 27$	$3 \wedge 3 = 27$	$3 \times 3 \times 3 = 27$	$3 ** 3 = 27$

Operator: [https://nl.wikipedia.org/wiki/Operator_\(wiskunde\)](https://nl.wikipedia.org/wiki/Operator_(wiskunde))

Ik snap het 3

- Ik kan uitleggen wat dit deel van het programma doet



Antwoord

Dit deel zorgt ervoor dat het hoogste getal dat kan worden ingevoerd 9 is. In dit voorbeeld hebben we hiervoor gekozen om te voorkomen dat de tekst tijdens het invoeren gaat scrollen.

De voorwaarde kan je ook op twee andere manieren opgeven:



Programmer 2: Debuggen en algoritmes

Kaart 2a - Opdracht 1: Foutje

Ik kan uitleggen ...

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

Antwoord



https://makecode.microbit.org/_gCHLTC4MWEmm

De variabele *mijnGetal* wordt eerst gevuld met een tekst (string), waarna wordt geprobeerd om deze tekst te laten zien met het blokje *show number*. Dit blokje kan echter alleen overweg met getallen.

Voorbeeld van een fout in het algoritme

Bij dit programma wordt de tekst "Hallo" niet getoond omdat de blokjes *show string* en *set mijnTekst to* in de verkeerde volgorde staan.



https://makecode.microbit.org/_gpb2P1LpihFu

Over het algemeen kan je het volgende zeggen:

- Bij een *syntaxfout* doet het programma het niet.
- Bij een *fout in het algoritme* doet het programma niet wat de programmeur wilde.

Ik snap het

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

Oplossing 1

Gebruik een het blok *toon tekens* om de inhoud van een variabele van het type String te laten zien



https://makecode.microbit.org/_Php8Wo2y1ev3

Oplossing 2

Vul de variabele met een waarde van het type Number (Getal)



<https://makecode.microbit.org/iwpWW2L2bU99>

Oplossing 3 (voor de specialisten)

Je kan van een *tekenreeks* een *getal* maken. Hieronder zie je hoe dat kan:



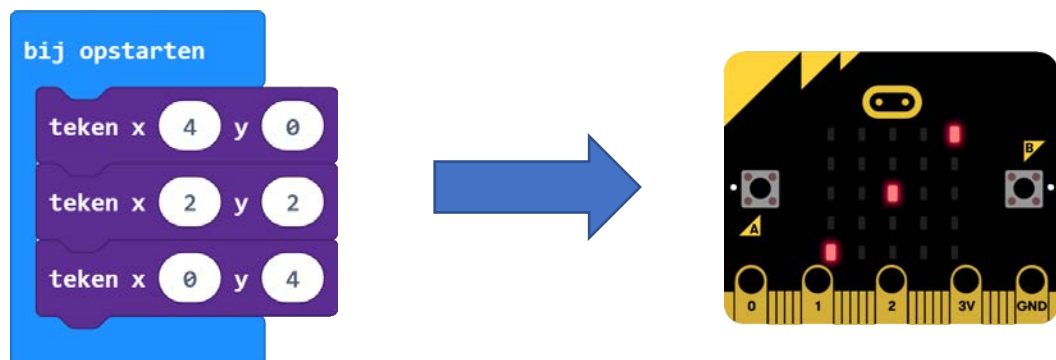
<https://makecode.microbit.org/7Df7ijHyXYo5>

Als een variabele eenmaal is gevuld met een string of met een getal dan kan je het type niet meer veranderen. Je kan wel een nieuwe variabele (*mijnEchteGetal*) maken en een parse (converteer) blok gebruiken om van de string een getal (zie hieronder) te maken.

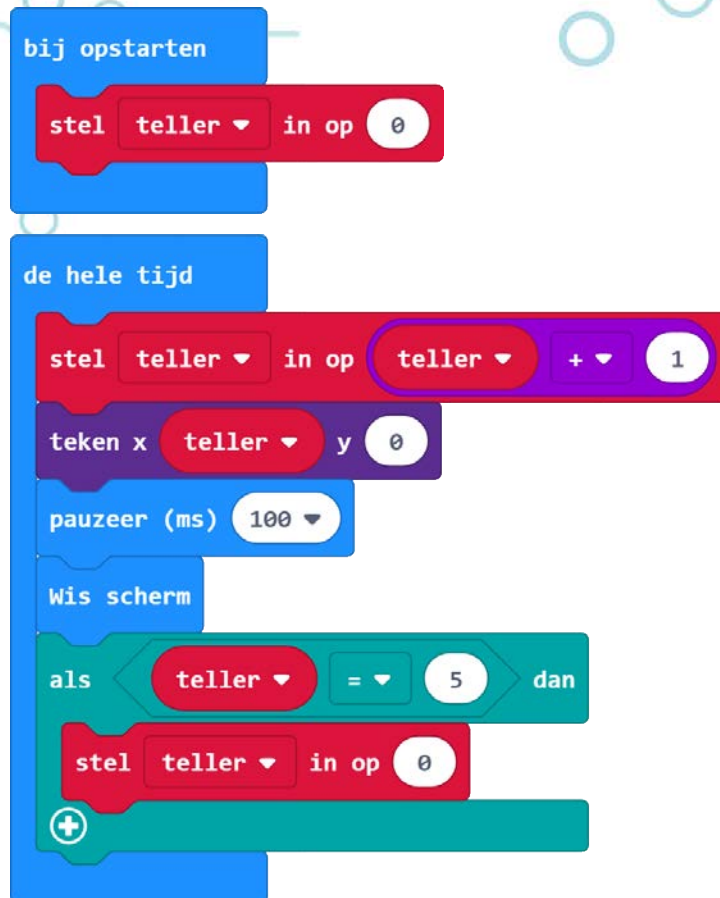
Kaart 2b - Opdracht 2: Plot erop los

Tip blokje plot x y

Laat de leerlingen eventueel eerst het blokje *teken x y* ontdekken met dit programma.



Bouw het programma



https://makecode.microbit.org/_OKY8gyCAYfYz

Ik kan vertellen ...

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

Antwoord

Het looplicht begint telkens bij het tweede ledje in plaats van bij het eerste.

Bij het zoeken naar fouten is het belangrijk dat je zo precies mogelijk beschrijft wat je ziet en wat er fout gaat.

Ik snap het

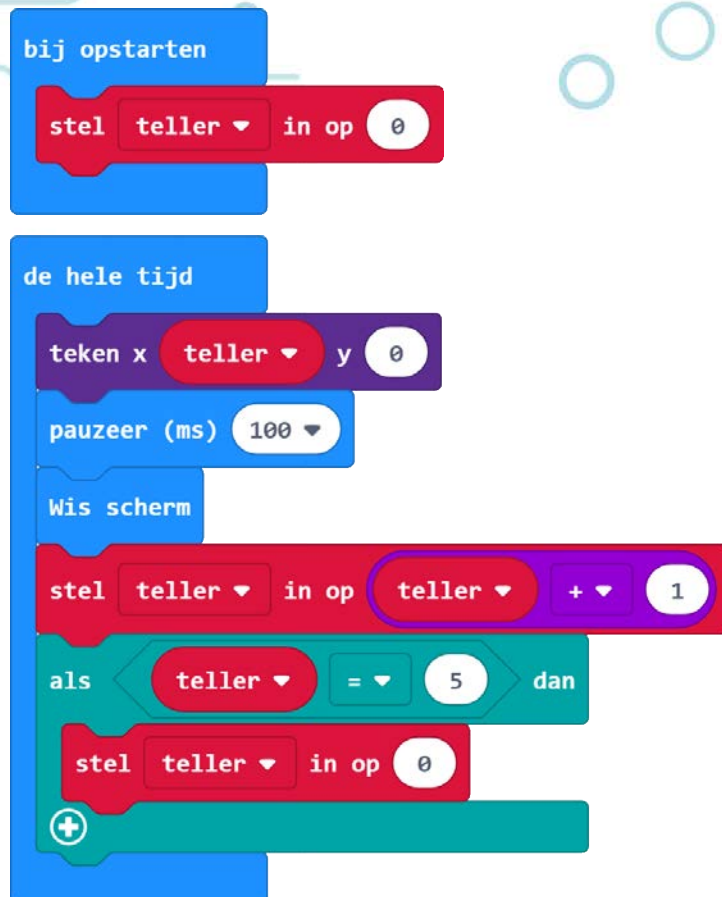
Ik heb het programma zo aangepast dat het wél doet wat de bedoeling is.

Wat gaat er fout?

Het gaat fout met het eerste blokje in de lus *de hele tijd*. Hier wordt de variabele *teller* al met één verhoogd terwijl het ledje op (0,0) nog moet worden aangezet. Hierna staan twee mogelijke oplossingen.

Oplossing 1

Heel veel programma's werken met dit soort tellers en het is gebruikelijk om de teller pas op het einde van een lus op te hogen. Dit is ook de 1^e oplossing

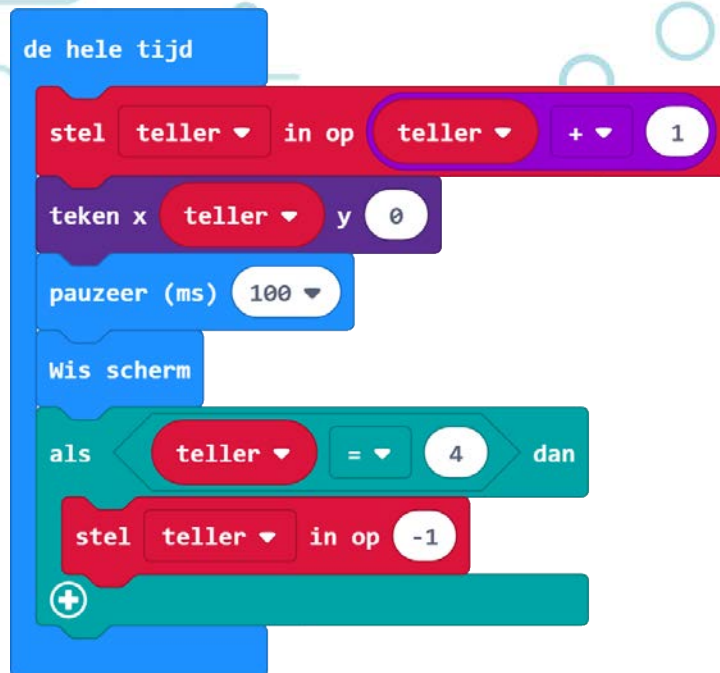


https://makecode.microbit.org/_hueP9ECDREhW

Oplossing 2

Hierbij blijft de volgorde van de blokken (syntax) hetzelfde, maar beginnen we niet te tellen bij 0, maar bij -1.

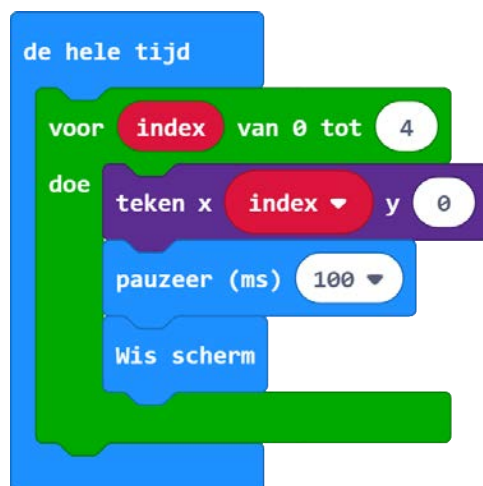




https://makecode.microbit.org/_XoqKR4c38U8b

Een ander algoritme

In plaats van het 'zelf' ophogen van de teller en telkens met een *als ... dan* blok controleren of deze de maximale waarde heeft bereikt kan je ook gebruik maken van het *voor ... van 0 tot ...* blok. (deze zogenaamde 'lus' wordt in kaart 3 uitgelegd)

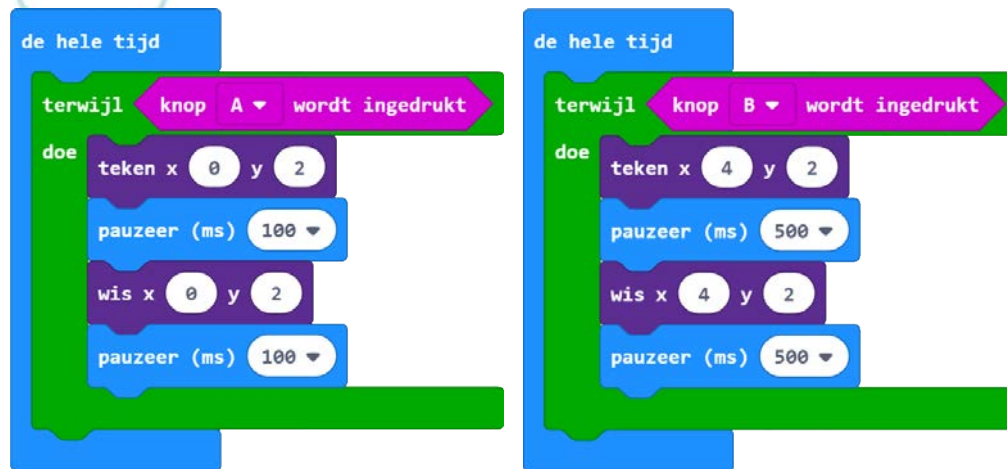


https://makecode.microbit.org/_Aff4veiqJcmY

Programmer 3a: Lussen

Kaart 3a - Opdracht 1: Multitasken

Het programma



https://makecode.microbit.org/_3Dod9UbJ9aqU

Met *multitasken* bedoelen we dat een computer of een mens verschillende taken tegelijk uitvoert. Voor wat betreft de computer moet worden opgemerkt dat het lijkt alsof dit zo is. In de praktijk worden de verschillende taken in kleine stukjes gesplitst die razendsnel achter elkaar worden uitgevoerd.



Doordat het blokje *de hele tijd* vaker gebruikt kan worden, is het eenvoudig om de micro:bit verschillende zaken tegelijk te laten doen. Alle andere 'hoofdblokken' mag je maar één keer in een programma gebruiken. In de editor zijn deze fouten gemarkeerd met een geel

raster.



https://makecode.microbit.org/_39cLcdadCHx8

Ik snap waarom...

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

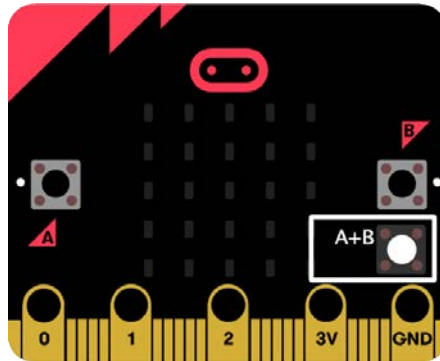
Antwoord

In de simulator is (nog) geen mogelijkheid om de twee knoppen tegelijk in te drukken (de muis zou dan op twee plaatsen tegelijk moeten zijn).

wanneer knop A + B wordt ingedrukt



Voor het blok *wanneer knop A + B wordt ingedrukt* verschijnt in de simulator een knop waarmee het indrukken van de twee toetsen tegelijk wordt gesimuleerd:

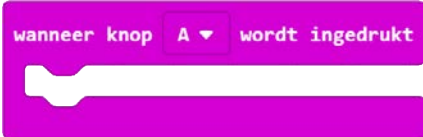


Dit werkt echter niet bij de blokjes *knop A wordt ingedrukt* en *knop B wordt ingedrukt*.

Ik snap het verschil ...

... tussen deze blokken:

Antwoord

	
Reageert als de knop wordt ingedrukt en daarna wordt losgelaten.	Reageert direct als dit blokje 'aan de beurt is' in het programma en de knop is ingedrukt.

Kaart 3b - Opdracht 2: Knippen (maar niet te vaak)

Het programma



https://makecode.microbit.org/_aLC3ribiqEjc

Ik snap het

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

Antwoord

In het blokje doet de vier ook mee. Samen met de nul wordt de lus dus vijf keer doorlopen. Taalkundig zou het dus *tot en met* moeten zijn.



Kaart 3c - Opdracht 3: Playlist

Het programma



https://makecode.microbit.org/_cTkdGW4fee25

Dit is een moeilijke kaart en we hebben hem opgenomen om alle soorten lussen te kunnen behandelen.

Het is ook voor het eerst dat we een uitstapje maken naar de JavaScript editor.

Ik snap het

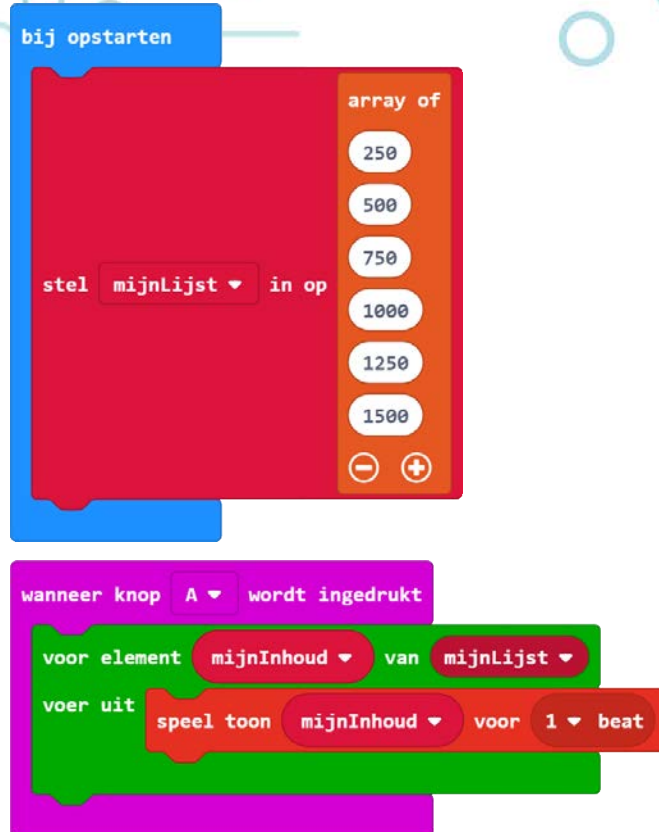
Ik kan het programma zo aanpassen dat de lijst met tonen 4 keer wordt afgespeeld als ik op *knop A* druk.

Oplossing



https://makecode.microbit.org/_e5pKTxKbhEg8

Voor de specialisten

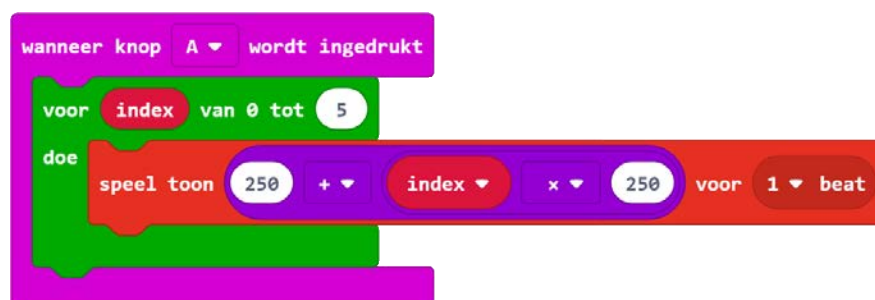


https://makecode.microbit.org/_dEaX6z7g0dMH

Let op Bij het werken met de JavaScript (tekst) versie van het programma mag geen tyfout gemaakt worden, zoals bijvoorbeeld `]` vervagen door `,` , of een punt gebruiken i.p.v. een komma. Dit leidt in de meeste gevallen tot syntaxfouten waardoor het programma niet meer (goed) kan worden uitgevoerd, of niet meer naar blokken kan worden teruggezet.

Dit kan ook

In dit voorbeeld zijn de tonen in de reeks steeds een veelvoud van 250. Hierom kan in dit voorbeeld hetzelfde resultaat worden bereikt met dit programma:



https://makecode.microbit.org/_eMWhjf3FV8xY

Programmer 4: Voorwaarden

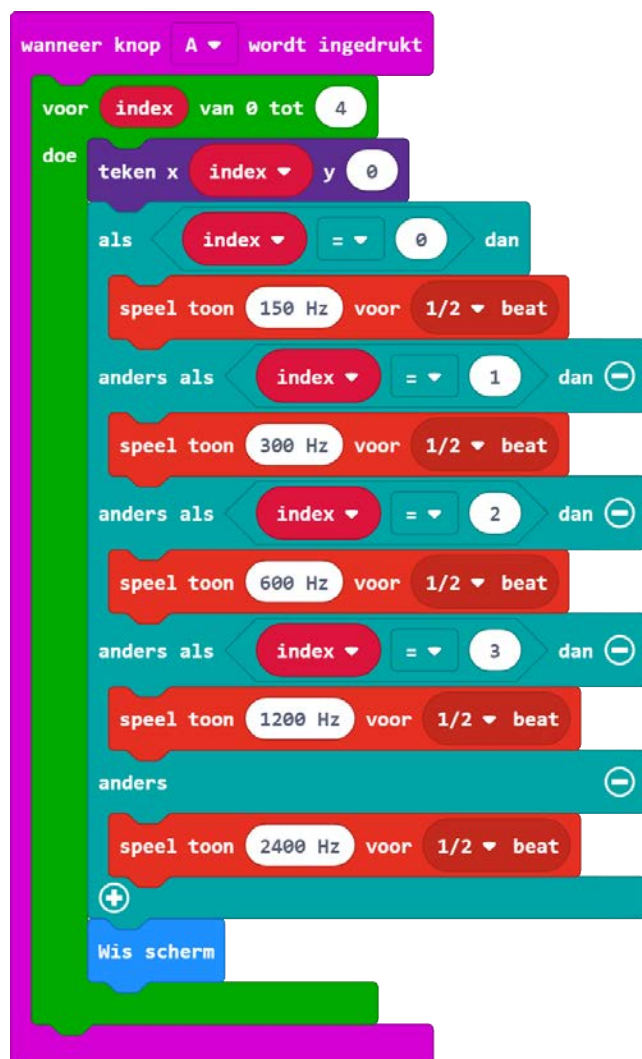
Tip Computational Thinking

In de leerlijn Programmeren in het VO staan leuke unplugged¹ oefeningen om kennis te maken met *als – dan – anders als – anders*.

- Middenbouw:
[https://maken.wikiwijs.nl/74282/Programmeren in het PO#!page-1838209](https://maken.wikiwijs.nl/74282/Programmeren%20in%20het%20PO#!page-1838209)
- Bovenbouw:
[https://maken.wikiwijs.nl/74282/Programmeren in het PO#!page-1838211](https://maken.wikiwijs.nl/74282/Programmeren%20in%20het%20PO#!page-1838211)

Kaart 4a - Opdracht 1: Toonladder

Het programma



https://makecode.microbit.org/_amFTefgxR3dm

Het blokje voor ... van 0 tot ...

Met dit programmaatje kan je de werking van het blokje laten zien

¹ Unlugged activiteiten zijn oefeningen over Computational Thinking waarbij geen computer nodig is

```

de hele tijd
  voor index van 0 tot 4
  doe
    toon nummer index
    pauzeer (ms) 100
  
```

https://makecode.microbit.org/_8FYfDET1wFew

Ik snap het

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

```

index = 4

```

Antwoord

Dit is niet nodig omdat dit bij de voorwaarde *anders* is afgevangen.

Door een extra *anders als* toe te voegen en *anders* te verwijderen zou je dit wel nodig hebben:

```

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 als index = 4 dan
      speel toon 2400 Hz voor 1/2 beat
  +
  wis scherm

```

https://makecode.microbit.org/_er5hY8eEyThE

Ander algoritme 1

Omdat er een rekenkundig verband is tussen de opeenvolgende tonen zijn er ook andere manieren (algoritmes) om hetzelfde resultaat te bereiken. Eén hiervan is deze:

```

wanneer knop A wordt ingedrukt
  stel mijnToon in op 150
  voor index van 0 tot 4
  doe
    teken x index y 0
    speel toon mijnToon voor 1/2 beat
  Wis scherm
  stel mijnToon in op 2 x mijnToon
  
```

https://makecode.microbit.org/_9qCJ6r5cmDXf

Kaart 4b - Opdracht 2: EN en/of OF spelletje

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 x y y
  
```

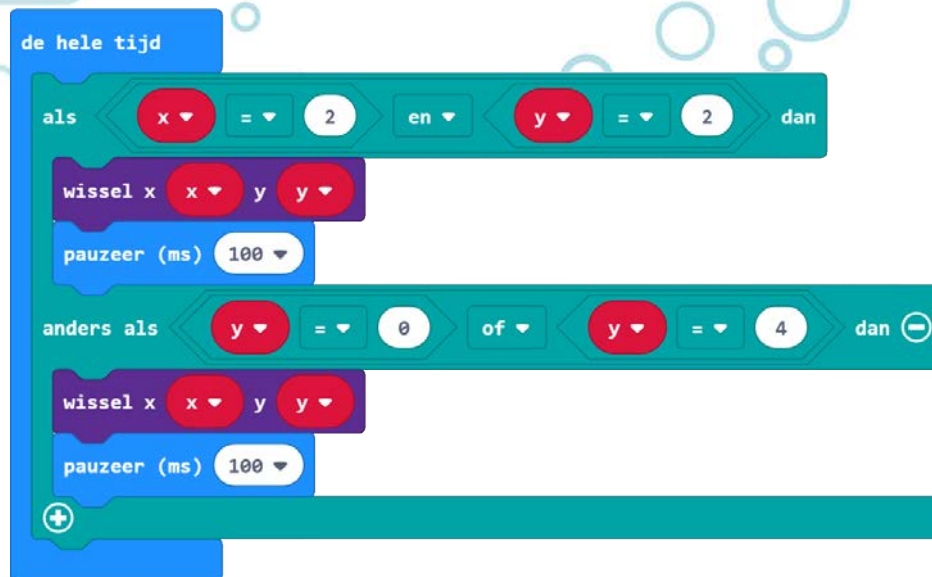
```

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 x y y
  
```

```

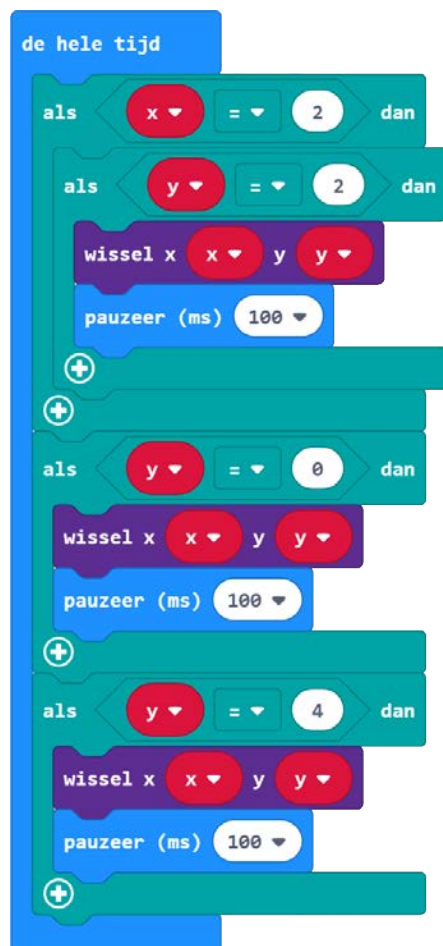
de hele tijd
  als x = 2 en y = 2 of y = 0 of y = 4 dan
    wissel x x y y
    pauzeer (ms) 100
    +
  
```

https://makecode.microbit.org/_gK4ghvVHa84g



https://makecode.microbit.org/_WJDfWUDt9EUo

Nog een manier



https://makecode.microbit.org/_RDAem4czKbvp

Ik snap het

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



Oude blokjes?

Inderdaad, hierboven en hieronder zie je nog de blokjes zoals die in de oude editor werden gebruikt. Deze had de mogelijkheid om lange samengestelde voorwaarden op een overzichtelijke manier weer te geven, onder elkaar in plaats van alles naast elkaar. In het Nederlands heette dit *Externe invoeren*. In het Engels was het *External Inputs*.

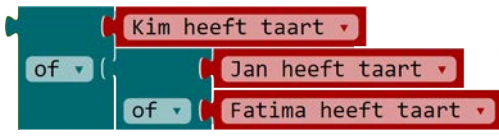
Als je ook vindt dat deze mogelijkheid weer terug moet komen, ga dan naar de Github-site van de ontwikkelaars en maak je wensen kenbaar:

<https://github.com/Microsoft/pxt-microbit/issues/1698>

We hebben de oude blokjes hier gebruikt omdat ze zo lekker duidelijk waren

Antwoord

Dit is een typisch voorbeeld waarbij de programmeur zich moet afvragen wat precies de gewenste logica is. Hieronder zie je hoe de zin *Kim of Jan en Fatima hebben taart* duidelijker kan worden geschreven.

Oude blokjes										
	<table border="1"> <tr> <td rowspan="2">EN</td><td>OF</td><td>Kim heeft taart</td></tr> <tr> <td></td><td>Jan heeft taart</td></tr> <tr> <td></td><td></td><td>Fatima heeft taart</td></tr> </table>	EN	OF	Kim heeft taart		Jan heeft taart			Fatima heeft taart	Met een komma: - Kim of Jan, en Fatima hebben taart Met haakjes: (Kim of Jan) en Fatima hebben taart
EN	OF		Kim heeft taart							
		Jan heeft taart								
		Fatima heeft taart								
	<table border="1"> <tr> <td rowspan="2">OF</td><td></td><td>Kim heeft taart</td></tr> <tr> <td>EN</td><td>Jan heeft taart</td></tr> <tr> <td></td><td></td><td>Fatima heeft taart</td></tr> </table>	OF		Kim heeft taart	EN	Jan heeft taart			Fatima heeft taart	Met een komma: - Kim of, Jan en Fatima hebben taart Met haakjes: - Kim of (Jan en Fatima) hebben taart
OF			Kim heeft taart							
	EN	Jan heeft taart								
		Fatima heeft taart								

Voor de specialisten

Dit probleem komt ook voor bij rekenen. Bij een som als $3 + 2 \times 4$ is echter vastgelegd in welke volgorde de berekening moet worden uitgevoerd (oude ezelsbruggetje Mijnheer Van Dale Wacht op Antwoord).

Bij logica geldt de regel NAO:

1. Not
2. And
3. Or

Dit speelt een rol bij tekstuele programmeertalen waar de regel zo geschreven wordt:

```
(Kim_heeft_taart || Jan_heeft_taart && Fatima_heeft_taart)
```

Operator	Betekenis
	OR
&&	AND

Kaart 4c - Opdracht 1: Wie drukte als eerste?

Het programma



https://makecode.microbit.org/_9iD909C8YAkx

Vraag

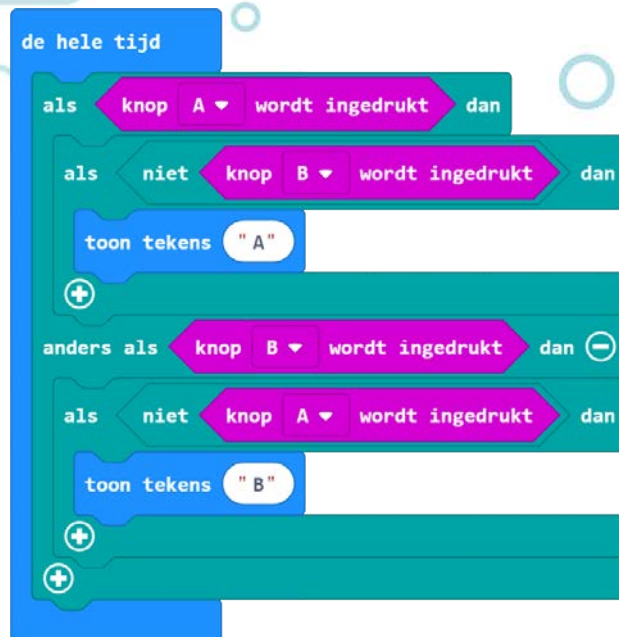
- Wat gebeurt als je knop A ingedrukt houdt en daarna op knop B drukt?
- Wat gebeurt als je knop B ingedrukt houdt en daarna op knop A drukt?

Antwoord

In het eerste geval blijft de *A* staan, het *anders dan* gedeelte van het blokje wordt niet meer uitgevoerd (wat ook de bedoeling is).

In het tweede geval verandert de *B* in een *A* (niet de bedoeling).

Oplossing



<https://makecode.microbit.org/diY6665Md3Tt>

Snap ik het?

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



Antwoord

Als op *knop A* wordt gedrukt komt er een *1* te staan.

Het blokje *knop A wordt ingedrukt* geeft *Waar (True)* als er op de knop wordt gedrukt.

Daarvoor staat twee keer een *niet* blokje die de waarde van *knop A wordt ingedrukt* telkens 'omdraaien'. Dit heet officieel *inverteren*.

Het werkt net als met een muntje: als je het twee keer omdraait dan ligt het weer hetzelfde. Drie keer omdraaien geeft hetzelfde effect als één keer omdraaien. De twee *niet* blokjes doen dus uiteindelijk niets.

Programmer 5: Meet het lichtniveau

Kaart 5a - Opdracht 1: Plot staafdiagram

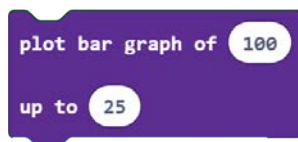
Het programma



https://makecode.microbit.org/_Y55Xi8aaq99d

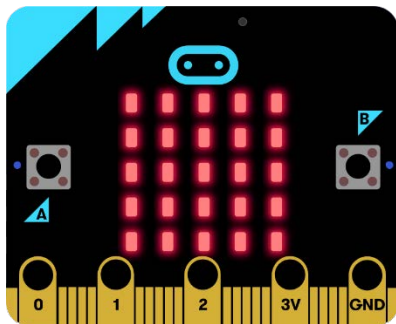
Ik kan voorspellen ...

... hoe het staafdiagram er uitziet als ik dit programmeer.



Antwoord

De maximale waarde die wordt verwacht is 25. Bij 100 gaan dus alle leds branden:



Kaart 5a - Opdracht 2: Meet het lichtniveau

Het programma



https://makecode.microbit.org/_LdiYHRhUs0yP



Hoe kan het dat het display tegelijk licht kan geven licht kan meten?

In het totaal worden er negen leds gebruikt om licht te meten. Dit doen ze in 4 milliseconde (vier duizendste van een seconde). Van iedere seconde geven deze leds dus 99,6% licht en zijn ze 0,4% van de tijd bezig met het meten van licht. Dit gaat zo snel dat je niet ziet dat de leds tijdens de lichtmeting even uit zijn.

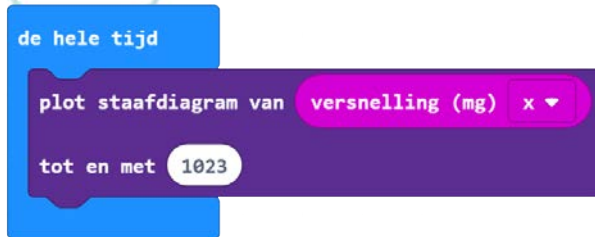
De lichtsensor is overigens het gevoeligst voor rood licht omdat de leds het gevoeligst zijn voor de kleur licht die ze kunnen geven.

1.1	1.2	1.3
2.2	2.3	2.1
3.3	3.1	3.2

Programmer 6: Versnelling en beweging

Kaart 6a - Opdracht 1: X-moves

Het programma



https://makecode.microbit.org/_MJ5MUw3pLLH8

Ik weet waarom ...

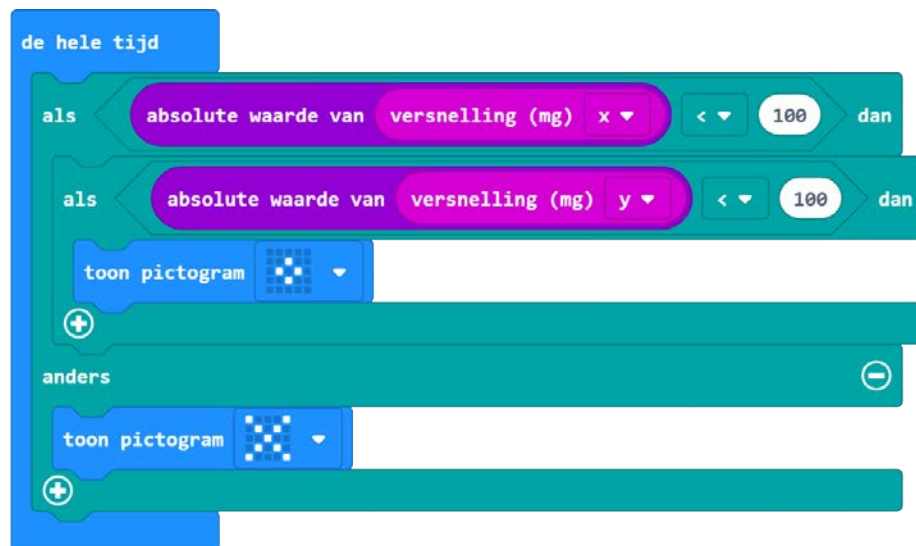
... in dit programma bij *up to* de waarde 1023 is ingevuld.

Antwoord

Dit is omdat 1023 de maximale waarde is die het blokje *versnelling (mg)* kan teruggeven.

Kaart 6b - Opdracht 2: Waterpas

Het programma



<https://makecode.microbit.org/fPsVq7332WKi>

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.

Antwoord

Dit is omdat de nauwkeurigheid anders te groot zou zijn. De waarde van de versnelling kan namelijk variëren van -1023 tot +1023.

Op deze manier is de werking van het waterpas duidelijk. Er is natuurlijk niets op tegen om de nauwkeurigheid groter te maken, bijvoorbeeld 50.

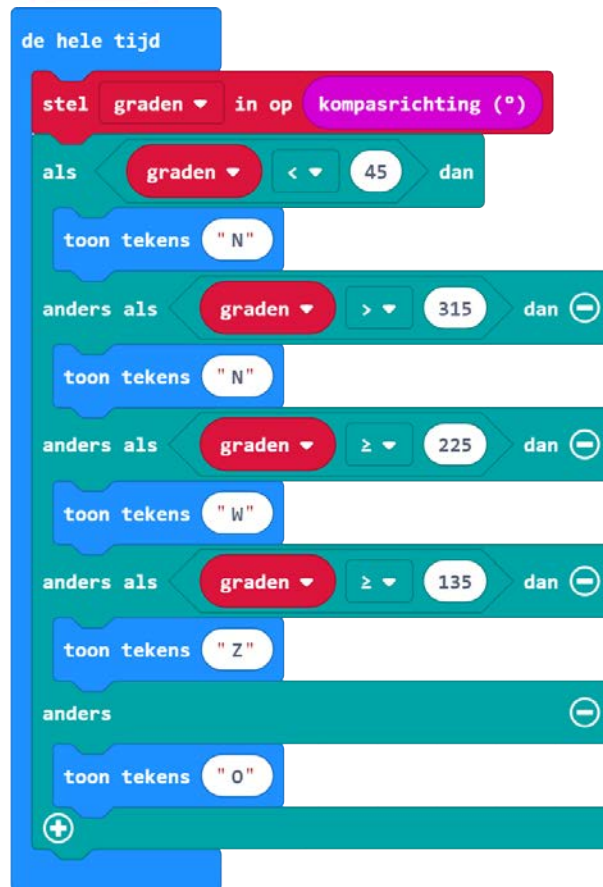
Tip: Maak er een spel van

Je kan de micro:bit waterpas gebruiken om er een spel mee te spelen waarbij één of meer kinderen de micro:bit moeten verplaatsen en daarbij horizontaal moeten houden. Het

wordt dan interessant om het programma zo aan te passen dat de leds gaan knipperen als de micro:bit te schuin wordt gehouden.

Kaart 6c - Opdracht 3: Maak een kompas

Het programma



<https://makecode.microbit.org/EeP2xoVajbUF>

Ik snap het

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

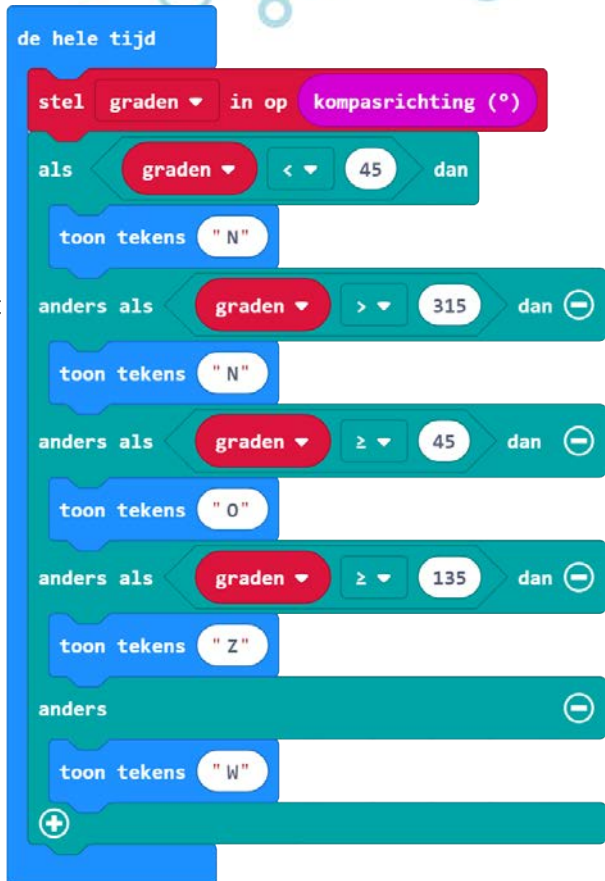
Waarom staan ze in het programma in de omgekeerde volgorde (Noord, West, Zuid, Noord)?

Antwoord

Stel dat het programma anders hadden opgebouwd.

In dit voorbeeld is de voorwaarde voor *Noord* nog steeds juist.

De voorwaarde voor *Oost* lijkt juist, maar stel dat het kompas richting *Zuid* (180°) wijst dan gaat het fout. 180 is namelijk ook groter dan 45 en op het display komt *Oost* te staan.

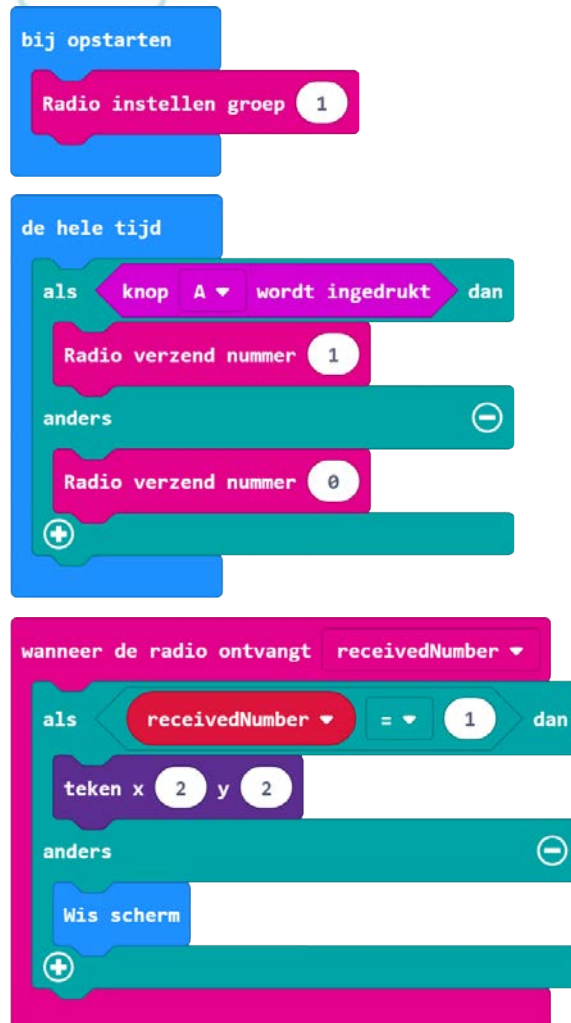


https://makecode.microbit.org/_42xPPpC8R3xW

Programmer 7: Radio

Kaart 7a - Opdracht 1: Radioled

Het programma



https://makecode.microbit.org/_M8i38q1va7vd

Kaart 7b - Opdracht 2: Speurneus

Het programma van de zender



```

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

<https://makecode.microbit.org/i2aPa8ficUf1>

Het programma van de ontvanger

```

bij opstarten
  Radio instellen groep 1

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

<https://makecode.microbit.org/H1zdqwLVK7zm>

Vraag

Waarom moet de micro:bit eerst in een envelop?

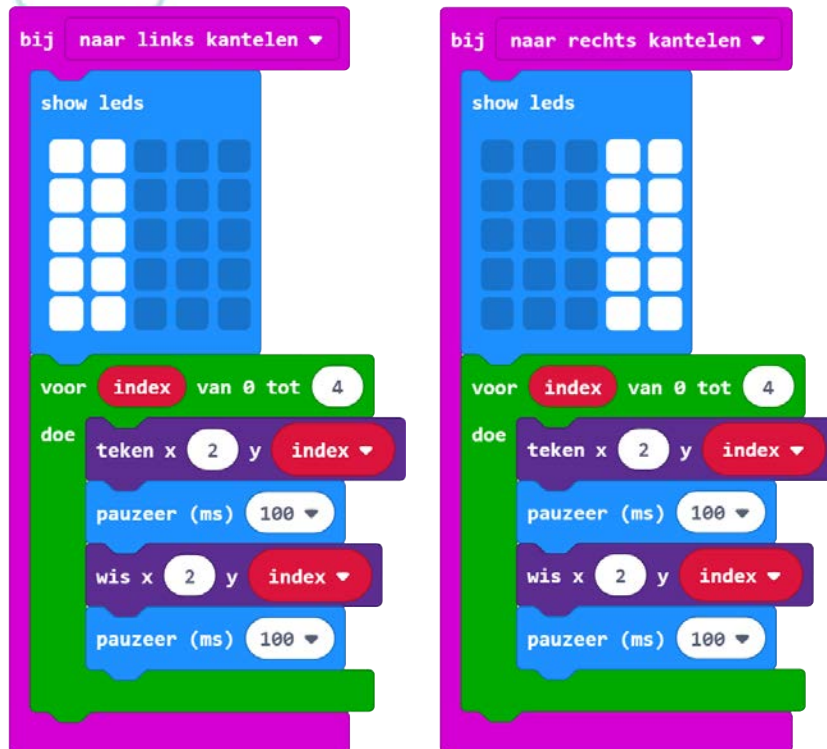
Antwoord

Hiermee voorkom je de kans op kortsluiting als je de micro:bit direct in aluminiumfolie zou wikkelen.

Programmer 8: Functies

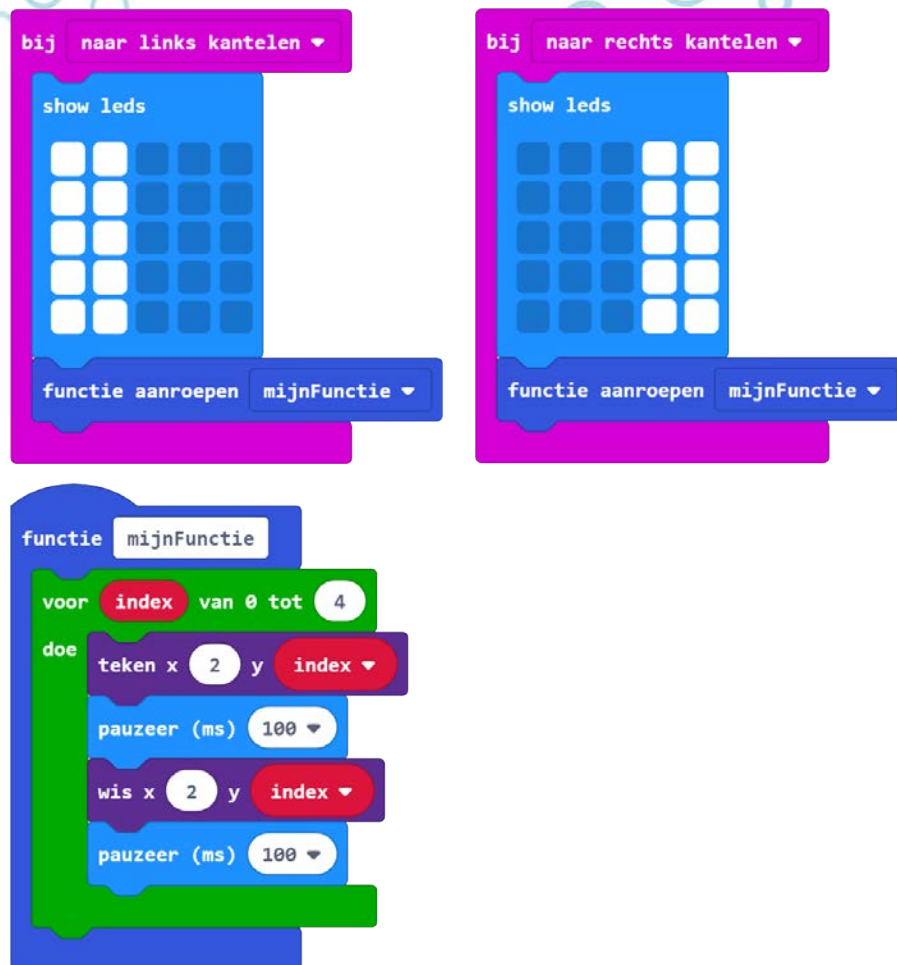
Kaart 8a - Opdracht 1: Van links naar rechts

Het programma



https://makecode.microbit.org/_P089fEMuxL5R

Met functie

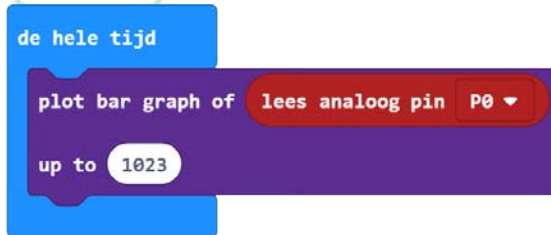


https://makecode.microbit.org/_chLMA7H83YvL

Programmer 9: Pins

Kaart 9a - Krijgt de plant genoeg water?

1^e versie van het programma



https://makecode.microbit.org/_MpW2cgYFv69P

2^e versie van het programma



https://makecode.microbit.org/_Fw61XbVWsbUE

Deze kaart is gebaseerd op één van de voorbeelden op de website van de editor:

<https://makecode.microbit.org/projects/soil-moisture/code>

Over analoog en digitaal

- Een *analoog* signaal kan alle waarden aannemen. Zo zit tussen 7 en 8 bijvoorbeeld 7,5, maar ook 7,55 en 7,5000067 etc.
- Een *digitale* waarde laat zich in een getal met een vast aantal digits (cijfers) beschrijven.
Met digitale techniek wordt over het algemeen bedoeld dat de signalen maar twee waarden (betekenissen) kunnen hebben:



Ja	Nee
----	-----

Waar	Onwaar
True	False
Eén	Nul
Aan	Uit

Zie ook: https://wikikids.nl/Analoog_en_digitaal

Om over na te denken

Zijn de cijfers die je leerlingen geeft analoog of digitaal? En wat is dan een *zes min* of een *zeven plus*?

Kaart 9b – Schuifmuziek

Het programma



https://makecode.microbit.org/_JPYLWUXpYi56

Over weerstanden

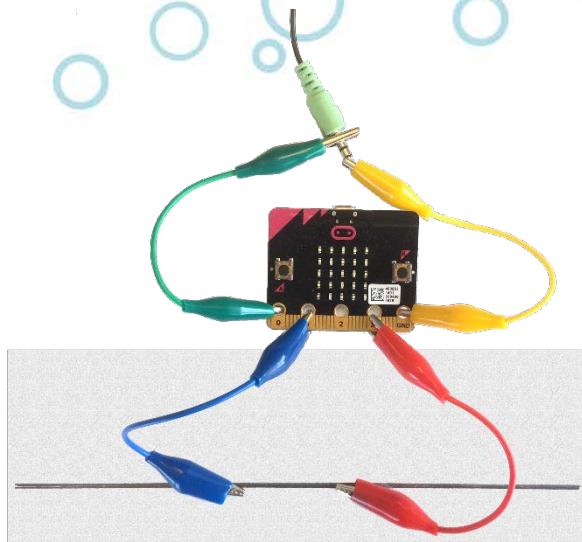
Weerstand is één van de drie elementen uit de Wet van Ohm. Deze wet beschrijft hoe elektrische spanning, stroom en weerstand zich tot elkaar verhouden:

$$Stroom = \frac{Spanning}{Weerstand}$$

De weerstand die bij deze opdracht wordt gebruikt bestaat uit het grafietpoeder (koolstof) uit een potlood.

Het is bij het 'tekenen' van de weerstand belangrijk dat er een 'vette' strook ontstaat.

De breedte van de streep moet ongeveer 2 mm zijn. Hoe breder je de streep maakt, hoe lager de weerstand en hoe kleiner het verschil in weerstand als je er met de klemmen over schuift.



Let op: Gebruik een 'echt' potlood. Er zijn ook potloden die geen grafiet bevatten.

Ik snap het

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



<https://makecode.microbit.org/TW44Evi1AFMy>

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

Antwoord

Ten opzichte van het basisprogramma heeft het uiteindelijke programma nog twee andere functionaliteiten:

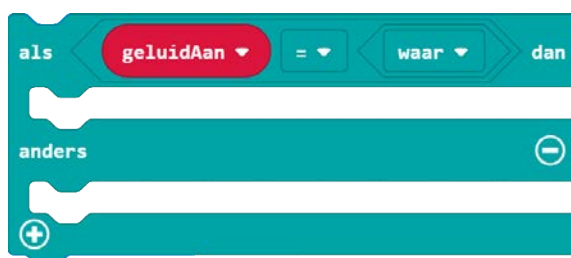
- Aan/uit functie voor het geluid
- Onderdrukking van geluid als er geen contact wordt gemaakt (drempel)

	Met de variabele <i>geluidAan</i> wordt geregeld of het geluid <i>Aan</i> of <i>Uit</i> staat. Bij de start van het programma wordt <i>geluidAan</i> gevuld met <i>True</i>.
	Met <i>knop A</i> wordt het geluid aan of uit gezet.

	Er is alleen geluid als de variabele <i>geluidAan</i> de waarde true heeft EN de gemeten waarde van de weerstand (<i>mijnWeerstand</i>) groter is dan 400. Het uitzetten van de toon wordt gerealiseerd met het blokje <i>speel toon</i>. Als <i>geluidAan</i> onwaar is, zorgt het blokje <i>rust (ms)</i> voor een stilte.
--	--

Verkorte versie als ... dan

In het eerste *als ... dan* blok hebben nu een blokje met een vergelijking gezet waarmee wordt gekeken of de variabele *geluidAan* de waarde *waar* heeft.



Omdat *geluidAan* een zogenaamde booleaanse variabele is en dus alleen maar de waarde waar of onwaar kan hebben, mag het blokje met de vergelijking worden weggelaten:



Aandachtspunten

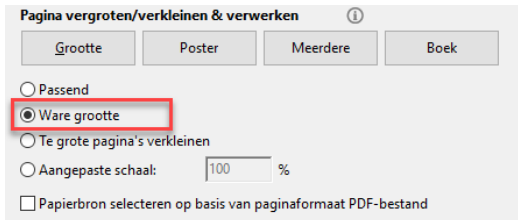
- Het kan zijn dat de drempel van 400 te hoog of te laag is. Experimenteer hiermee
- Als het programma niet werkt, controleer dan of de weerstand goed is aangesloten en ook werkt. Dit kan met het basisprogramma:



Bijlage 1: Kaarten afdrukken

De kaarten hebben een langwerpige vorm. Hierdoor kunnen ze ook op de (relatief) smalle Nederlandse schooltafeltjes naast de laptop of toetsenbord worden gelegd.

1. Druk de pdf in kleur af op wit papier. Kies er bij de printeropties voor dat het document op ware grootte wordt afgedrukt.



2. Snijd rondom 5 mm af
3. Vouw het papier dubbel
4. Lamineer de kaart. Gebruik als het even kan geen glanzend, maar mat lamineerplastic.

Als je meerdere kaarten moet afdrukken dan kan je ze dubbelzijdig afdrukken en na het afsnijden van de 5 mm doormidden snijden.



Dit is een uitgave van Devlab Academy en CodeKids.nl

Als er geen sprake is van winst oogmerk mag er vrijelijk uit deze uitgave worden gekopieerd, mits de bron wordt vermeld. Je mag dit document dan ook afdrukken en kopiëren.

Als je op de een of andere manier wel een winst oogmerk hebt, neem dan contact op met ons op:

This user guide is issued by DevLab Academy and CodeKids.nl

Reproduction for educational and non-commercial purposes is permitted provided that the source is acknowledged. Such acknowledgement must be included in each copy of the material.

In case of commercial use, please get in contact with us:



lex.van.gijssel@devlab.nl



chris@codekids.nl