



west-vlaanderen
de gedreven provincie

Python met Micro:Bit

Docenten Nele Moens, Michiel Van Iede en Wim Dejonghe

studiegebied **Onderzoek**

campus **Kortrijk**

academiejaar **2022-2023**



Inhoudsopgave

1.	Toon cijfers op LED display:	Fout! Bladwijzer niet gedefinieerd.
2.	Voer 1 keer uit / herhaal altijd (oneindige-iteratie):	Fout! Bladwijzer niet gedefinieerd.
3.	Tel af (en schrijf tekst):	Fout! Bladwijzer niet gedefinieerd.
4.	Tel danswedstrijd af :	Fout! Bladwijzer niet gedefinieerd.
5.	Communicatie met de computer via USB kabel	4
6.	Gebeurtenis op basis van drukknoppen knopA	4
7.	Event op knop A en B	Fout! Bladwijzer niet gedefinieerd.
8.	Tellen en weergeven van het aantal keren gedrukt op knop A.	Fout! Bladwijzer niet gedefinieerd.
9.	Geluidje afspelen wanneer op Knop A wordt gedrukt	Fout! Bladwijzer niet gedefinieerd.
10.	Parking versie1	4
11.	Lopende LED	5
1.	Python	Fout! Bladwijzer niet gedefinieerd.
12.	Parkeerplaatsen versie2	6
13.	Touch logo	8
14.	Externe pinnen	9
15.	Capacitieve Touch pin P0	11
16.	Dobbelsteen (random integer)	12
17.	Functies/Methoden met een parameter	14
18.	Gebeurtenis (event) bij schudden	15
19.	Schudden : Lovemeter :	16
20.	Analyse sensoren	17
21.	Automatische verlichting	18
22.	Geluidsensor	20
23.	Temperatuursensor	20
24.	Gyrosensor	20
25.	Magnetischesensor	21
26.	Data logging	22
27.	Starten en stoppen van Data logging en clear de data	25

1.Inleiding

13.Parking versie1

Hier maken we een parking-display met aanduiding van het aantal bezette plaatsen.

Gegeven:

Er zijn 10 plaatsen.

Drukknop A => registratie aan de ingang

Drukknop B => registratie aan de uitgang

Gevraagd:

Maak een teller zorg ervoor dat de teller niet boven de *10 plaatsen* gaat en ook niet onder de *0 plaatsen* gaat.

```
1  # Imports go at the top
2  from microbit import *
3
4  #declaratie variabelen met toekenning van een waarde
5  max_plaatsen = 10
6  plaatsenbezet = 0
7
8
9  # Code in a 'while True!' loop repeats forever
10 while True:
11     if button_a.was_pressed():
12         if plaatsenbezet < max_plaatsen:
13             plaatsenbezet = plaatsenbezet + 1
14     if button_b.was_pressed():
15         if plaatsenbezet > 0:
16             plaatsenbezet = plaatsenbezet - 1
17     display.scroll(plaatsenbezet)
```

14.Lopende LED

`display.set_pixel(x,y,helderheid)`

Met dit Micro:Bit statement kan een bepaalde pixel binnen de LED matrix worden aangestuurd.

Er wordt hier gebruik gemaakt van twee geneste FOR-loops om een loop-sequentie te programmeren.

```
1  # Imports go at the top
2  from microbit import *
3
4
5  # Code in a 'while True:' loop repeats forever
6  while True:
7      for y in range(5):
8          for x in range(5):
9              display.set_pixel(x, y, 9)
10             sleep(50)
11         sleep(1000)
12         display.clear()
13         sleep(1000)
```

Opdracht : Analyseer de werking en programmeer nog andere sequenties.

15.Parkeerplaatsen versie2

De bezette plaatsen / vrije plaatsen kunnen we ook grafisch aanduiden op de LED-matrix. Er zijn 5*5 LED's, zo kunnen we een parkeergarage van 25 plaatsen visualiseren.

Rekenkundige bewerkingen: de normale bewerkingen spreken voor zich: + - * die kunnen zowel met integer- als met float waarden werken. Let wel : een computer kijkt steeds om een bewerking zo eenvoudig mogelijk uit te voeren.

Dwz, als de parameters integer waarden zijn , dan zal de computer de bewerking ook uitvoeren met integer waarden en het resultaat zal ook een integer zijn.

Is echter een parameter een float dan zal de computer een complexere berekening uitvoeren en zal het resultaat ook een float zijn.

Met deze redenering is er echter een probleem bij de deling / . Twee integer waarden delen zal steeds een integer opleveren. Dit kan dus tot een onvolledige deling leiden. Vb:

$$10/5 = 2$$

$$5/2=2 \text{ !!!! volledig behandeld met integers}$$

$$5.0/2=2.5$$

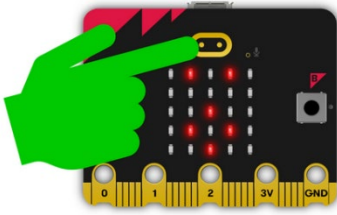
Met integers kan ook een restdeling worden uitgevoerd (modulo deling). Vb:

$$11\%4 = 11 \text{ mod } 4 = 3$$

Hiermee kunnen we de coördinaten van een LED x, y bepalen uit een groter getal

```
1  # Imports go at the top
2  from microbit import *
3
4  max_plaatsen = 25
5  plaatsenbezet = 0
6  yPos = 0
7  xPos = 0
8  ledWaarde = 0
9
10 # Code in a 'while True:' loop repeats forever
11 while True:
12     if button_a.was_pressed():
13         if plaatsenbezet < max_plaatsen:
14             plaatsenbezet = plaatsenbezet + 1
15             ledWaarde = plaatsenbezet - 1
16             xPos = ledWaarde % 5
17             yPos = int(ledWaarde / 5)
18             display.set_pixel(xPos, yPos, 9)
19     if button_b.was_pressed():
20         if plaatsenbezet > 0:
21             plaatsenbezet = plaatsenbezet - 1
22             ledWaarde = plaatsenbezet
23             xPos = ledWaarde % 5
24             yPos = int(ledWaarde / 5)
25             display.set_pixel(xPos, yPos, 0)
```

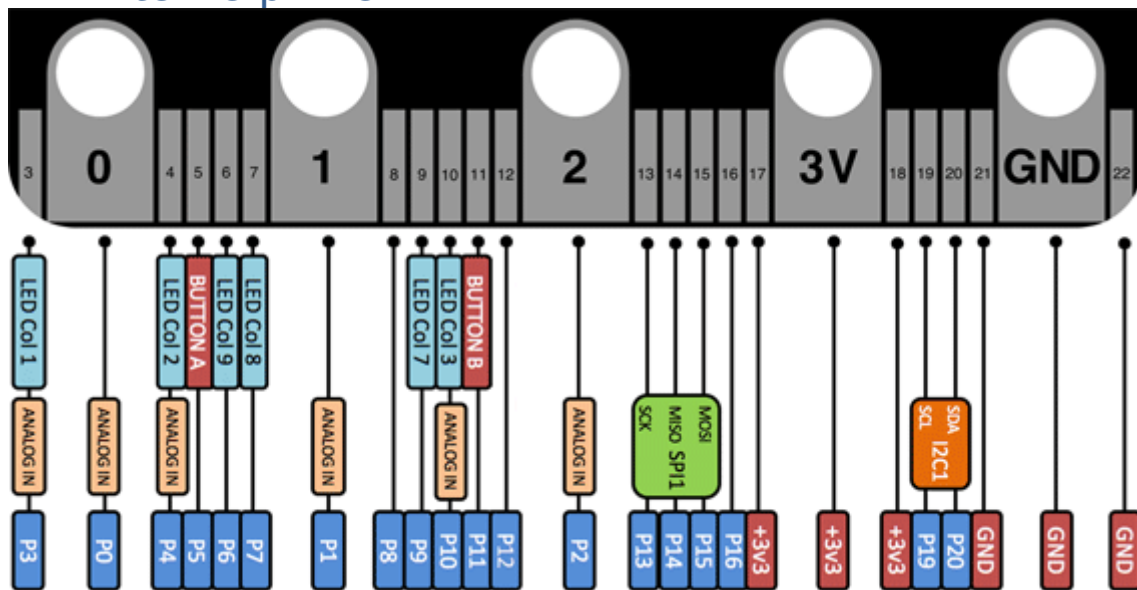
16.Touch logo



De Micro:Bit bevat ook een touch-logo (sensor). Dit is een soort drukknop die werkt op capacitieve verandering. Het volstaat om met de vinger hier dicht in de buurt te komen om de drukknop te activeren (analogie met een touch-screen)

```
1 # Imports go at the top
2 from microbit import *
3
4
5 # Code in a 'while True:' loop repeats forever
6 while True:
7     if pin_logo.is_touched():
8         display.show(Image.HAPPY)
9         sleep(100)
10        display.clear()
```


17. Externe pinnen



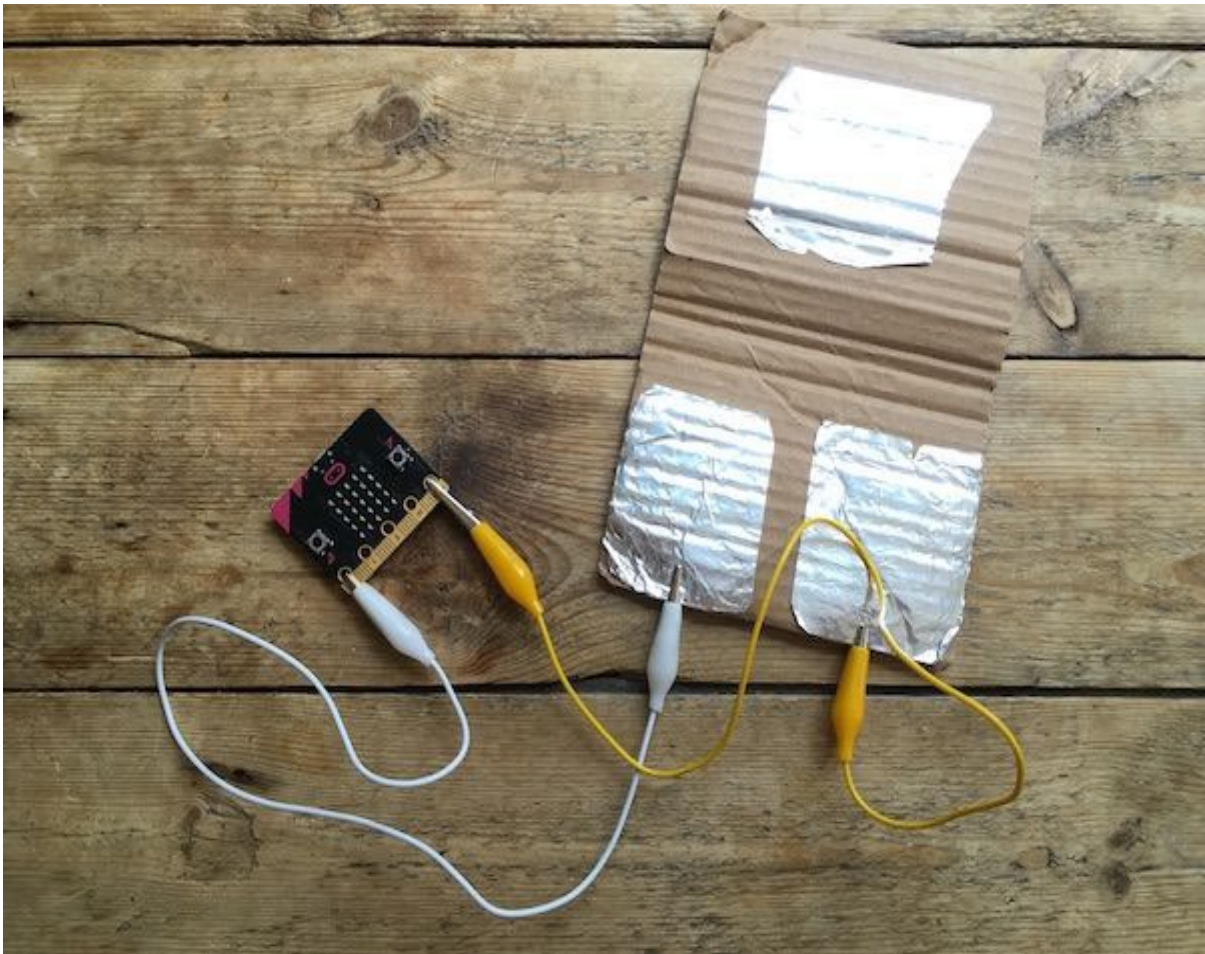
De meeste pinnen kunnen gebruikt worden als digitale in/uitgang. Sommige pinnen kunnen ook als een analoge ingang worden gebruikt (om bv een analoge sensor die een analoge elektrische spanning afgeeft ifv een te meten grootte als druk, temperatuur, geluid, licht, CO₂,). Nog andere pinnen kunnen worden gebruikt binnen een bus-protocol standaard (SPI, I2C). Zelf een drukknop maken kan eenvoudig door een elektrische verbinding te maken tussen de 0-pin en de GND. Dit met een elektrisch geleidende stof en niet met een isolator.

```

1  # Imports go at the top
2  from microbit import *
3
4  # Code in a 'while True:' loop repeats forever
5  while True:
6      if pin0.is_touched():
7          display.show(Image.HAPPY)
8      else:
9          display.show(Image.SAD)

```

Opdracht: maak een knop met aluminiumfoliestrookjes en krokodil-klemmen. Is een menselijk lichaam een geleider of een isolator?



18.Capacitieve Touch pin P0

Een pin kan ook veel gevoeliger worden gemaakt door die als een touch-sensor te laten werken (capacitief). Hierbij is geen verbinding nodig naar de GND als je met je vinger in de buurt komt van P0. Dit kan door éénmalig de touch_mode op die pin als capacitive in te stellen.:

```
1  # Imports go at the top
2  from microbit import *
3
4
5  pin0.set_touch_mode(pin0.CAPACITIVE)
6
7
8  # Code in a 'while True:' loop repeats forever
9  while True:
10     if pin0.is_touched():
11         display.show(Image.HAPPY)
12     else:
13         display.show(Image.SAD)
```

19.Dobbelsteen (random integer)

De Micro:Bit kan een random getal genereren tussen bepaalde grenzen. Hiervoor moet echter wel een aparte bibliotheek worden geïmporteerd. Met `random.randint(ondergrens, bovengrens)` kan dus een getal worden gegenereerd.

```
1  # Imports go at the top
2  from microbit import *
3  import random
4
5  # Code in a 'while True:' loop repeats forever
6  while True:
7      if button_a.was_pressed():
8          display.show(random.randint(1,6))
```

Opdracht: maak een reactie spel. Na een random tijd wordt een lichtje aangestuurd. Dan is het de bedoeling dat twee spelers (knop A – knop B) om ter snelst de knop indrukken. De Micro:Bit gaat na wie er snelst heeft gedrukt en displayed de winnaar.

Uitbreiding : De Micro:Bit kan ook bijhouden hoeveel keer A en B heeft gewonnen (stand) en dit weergeven als tekst op de Console (terminal).

Uitbreiding dobbelsteen: maak op de LED matrix de visualisatie van een echte dobbelsteen.

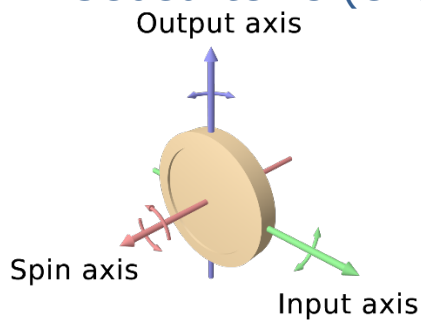
```
1  # Imports go at the top
2  from microbit import *
3  import random
4
5  # Code in a 'while True:' loop repeats forever
6  while True:
7      if button_a.was_pressed():
8          waarde = random.randint(1,6)
9          #display.show(waarde)
10         if waarde == 1:
11             display.clear()
12             display.set_pixel(2,2,9)
13         elif waarde == 2:
14             display.clear()
15             display.set_pixel(0,0,9)
16             display.set_pixel(4,4,9)
17         elif waarde == 3:
18             display.clear()
19             display.set_pixel(0,0,9)
20             display.set_pixel(2,2,9)
21             display.set_pixel(4,4,9)
22         elif waarde == 4:
23             display.clear()
24             display.set_pixel(0,0,9)
25             display.set_pixel(0,4,9)
26             display.set_pixel(4,0,9)
27             display.set_pixel(4,4,9)
28         elif waarde == 5:
29             display.clear()
30             display.set_pixel(2,2,9)
31             display.set_pixel(0,0,9)
32             display.set_pixel(0,4,9)
33             display.set_pixel(4,0,9)
34             display.set_pixel(4,4,9)
35         else:
36             display.clear()
37             display.set_pixel(0,2,9)
38             display.set_pixel(4,2,9)
39             display.set_pixel(0,0,9)
40             display.set_pixel(0,4,9)
41             display.set_pixel(4,0,9)
42             display.set_pixel(4,4,9)
```

20.Functies/Methoden met een parameter

Je kan ook een aparte functie maken om de Leds aan te sturen in python. Functies worden binnen programmeertalen ook als methoden benoemd of subroutines.

```
1  # Imports go at the top
2  from microbit import *
3  import random
4  def toonLeds(getal):
5      if getal == 1:
6          display.clear()
7          display.set_pixel(2,2,9)
8      elif getal == 2:
9          display.clear()
10         display.set_pixel(0,0,9)
11         display.set_pixel(4,4,9)
12     elif getal == 3:
13         display.clear()
14         display.set_pixel(0,0,9)
15         display.set_pixel(2,2,9)
16         display.set_pixel(4,4,9)
17     elif getal == 4:
18         display.clear()
19         display.set_pixel(0,0,9)
20         display.set_pixel(0,4,9)
21         display.set_pixel(4,0,9)
22         display.set_pixel(4,4,9)
23     elif getal == 5:
24         display.clear()
25         display.set_pixel(2,2,9)
26         display.set_pixel(0,0,9)
27         display.set_pixel(0,4,9)
28         display.set_pixel(4,0,9)
29         display.set_pixel(4,4,9)
30     else:
31         display.clear()
32         display.set_pixel(0,2,9)
33         display.set_pixel(4,2,9)
34         display.set_pixel(0,0,9)
35         display.set_pixel(0,4,9)
36         display.set_pixel(4,0,9)
37         display.set_pixel(4,4,9)
38     # Code in a 'while True:' loop repeats forever
39     while True:
40         if button_a.was_pressed():
41             waarde = random.randint(1,6)
42             toonLeds(waarde)
```

21.Gebeurtenis (event) bij schudden



De Micro:Bit bezit een gyrosensor die op de in de drie dimensies een versnelling kan waarnemen. Schudden is een actie die in de drie dimensies een versnelling kan veroorzaken. De Micro:Bit kan dit waarnemen en dit als voorwaarde worden gebruikt om een handeling uit te voeren. Is vergelijkbaar met een event op een knop maar nu niet drukken maar schudden.

Opdracht : laat de Micro:Bit een random getal genereren (0-100) bij het schudden van de Micro:Bit en geef dit getal weer op het LED-display.

```
1 # Imports go at the top
2 from microbit import *
3 import random
4
5 # Code in a 'while True:' loop repeats forever
6 while True:
7
8     if accelerometer.was_gesture('shake'):
9         waarde = random.randint(1,100)
10        display.show(waarde)
```


22.Schudden maar!!! Lovemeter :

Idem als vorige opdracht maar nu interpreteren we de random waarde naar een symbool.

- Getal tussen 0-30 : `display.show(Image.SAD)`
- Getal tussen 30-60 : `display.show(Image.SMILE)`
- Getal tussen 60-100 : `display.show(Image.HAPPY)`

Untitled project

```
1 # Imports go at the top
2 from microbit import *
3 import random
4
5 # Code in a 'while True:' loop repeats forever
6 while True:
7
8     if accelerometer.was_gesture('shake'):
9         waarde = random.randint(1,100)
10        #display.show(waarde)
11        if waarde < 30:
12            display.show(Image.SAD)
13        elif waarde < 60:
14            display.show(Image.SMILE)
15        else:
16            display.show(Image.HAPPY)
```


23.Analyse sensoren

Op de Micro:Bit zitten al een aantal sensoren die in python kunnen worden uitgelezen. Onboard:

- Gyro sensor : 3-dimensionele versnelling (g-force)
- Magnetisme : 3-dimensionele magnetische veldsterkte (μ -tesla)
- Geluidsensor
- Temperatuursensor
- Lichtsensor
- Voel-sensoren (drukknoppen, en touch sensor, externe: pinnen)

Om een analyse (studie) te maken over de werking en verwerking van data binnen de Micro:Bit kunnen we gebruik maken van het print() python statement. De Micro:Bit kan zeker uitgebreid worden met externe sensoren door die juist aan te sluiten op de beschikbare pinnen.

Analyse van lichtsensor:

```
1 # Imports go at the top
2 from microbit import *
3
4 # Code in a 'while True:' loop repeats forever
5 while True:
6     tekst = 'De hoeveelheid licht = ' + str(display.read_light_level())
7     print(tekst)
8     sleep(500)
```

Met het print commando stuur je data van de Micro:Bit naar de computer (console – terminal). Op de computer kan via het venster **Show Serial**

```
micro:bit flashed ✓  
De hoeveelheid licht = 101  
De hoeveelheid licht = 71  
De hoeveelheid licht = 33  
De hoeveelheid licht = 127  
De hoeveelheid licht = 83  
De hoeveelheid licht = 16  
De hoeveelheid licht = 148
```

In de code zie je ook een type casting (conversie/omzetting) van een integer waarde naar een string :
`str(integer)`

Je ziet ook een concatenatie van twee strings (twee strings aan elkaar kleven).

24. Automatische verlichting

Opdracht : Zorg dat de LED matrix in drie helderheden wordt aangestuurd afhankelijk van het invallend licht. Analogie met de automatische verlichting bij een wagen die een tunnel binnen komt. Wanneer geen lichten? Wanneer kleine lichten? Wanneer grote lichten? Bepaal eerst aan de hand van uw testen de twee grenswaarden voor de lichtsensoren.

Teken hier eerst een flowchart van!!!

```
1 # Imports go at the top
2 from microbit import *
3
4 # Code in a 'while True:' loop repeats forever
5 while True:
6     if (display.read_light_level() < 100):
7         display.show(Image('30903:'
8                             '06960:'
9                             '99999:'
10                            '06960:'
11                            '30903'))
12
13     elif (display.read_light_level() < 150):
14         display.show(Image('00300:'
15                             '03630:'
16                             '36963:'
17                             '03630:'
18                             '00300'))
19
20     else:
21         display.clear()
```

Uitbreiding: Je kan gerust meerdere niveaus gebruiken met meerdere helderheden.

25.Geluidsensor

Gelijkaardig kan je een analyse doen met de geluidsensor. Doe dit eerst door de waarde van het geluid te PRINTEN naar de Serial console. Bepaal hiermee welke waarden vertegenwoordigen stil/luid? Wat is stil? Wat is luid? Bepaal een grenswaarde en schrijf dan code om een alarm te maken wanneer er teveel lawaai is in de klas.

Meet het geluid in Python met het statement :

microphone.sound_level()

26.Temperatuursensor

Opdracht: Meet de temperatuur in de klas en display de temperatuur op de LEDS als getal of als figuur of met pixels. Print ook die waarde eens realtime naar de Serial Console.

Lees de temperatuur in Python met: ***temperature()***

27.Gyrosensor

Opnieuw gelijkaardig kan de gyrosensor worden geanalyseerd. Echter bezit de gyrosensor niet één enkele meetwaarde maar drie meetwaarden volgens de versnelling in de drie dimensies (X, Y en Z as).

accelerometer.get_x()

Bepaal zelf via printen naar de Serial console wat X, Y en Z is. De versnelling wordt uitgedrukt in milli-G-kracht (versnelling 1G = 9,81m/s²). Op die manier kan zwaartekracht berekend worden uit massa en versnelling $F=m.a$ (vb $m=80\text{kg}$ en $a=9,81\text{m/s}^2 \Rightarrow F=78,48\text{N}$).

Bepaal zelf wat en waar is in de x-richting de versnelling = 1000mG, waar en hoe 0, waar en hoe max positief, max negatief?

Idem voor de y-richting en de z-richting?

28.Magnetischesensor

Met de Micro:Bit kan ook een magnetisch veld worden gemeten. Opnieuw kan dit in de 3-dimensionele ruimte adhv de 3 assen (X, Y en Z). Hiermee kan een kompas worden gemaakt die volgens een as van de micro:bit (meestal X) het aardmagnetisme kan meten. Maar het kan ook magnetische velden van losse magneten meten en bepalen waar zich een NOORD of een ZUIDPOOL ligt.

Echter is het wel zo dat wanneer je deze sensor wenst te gebruiken de Micro:Bit een kalibratie procedure zal vragen die moet uitgevoerd worden op de Micro:Bit vooraleer deze kan worden gebruikt.

Experimenteer met deze sensor door gebruik te maken van het Python commando : ***compass.get_x()***

Stuur opnieuw waarden naar de Serial console na de kalibratie procedure gevolgd te hebben (zie instructies op de Micro:Bit zelf).

29.Data logging

Nu we wat ervaring hebben met de verschillende sensoren willen we veelal waarden van die sensoren gaan loggen (data wegschrijven op de Micro:Bit zelf en die bijhouden) gedurende een bepaalde tijd of een bepaalde omstandigheid om er dan nadien een analyse op uit te voeren eventueel met Excel.

We wensen dus bijvoorbeeld de lichtsterkte (zonlicht) te monitoren gedurende een volledige dag en dit door iedere minuut een lichtmeting uit te voeren.

Dit kan met de Micro:Bit in Python als volgt gebeuren.

Logging_Light

```
1 # Imports go at the top
2 from microbit import *
3 import log
4
5 log.set_labels('light')
6
7 @run_every(s=1)
8 def log_data():
9     my_effect = audio.SoundEffect(duration=10)
10    audio.play(my_effect)
11    log.add({'light': display.read_light_level()})
12
13 while True:
14    sleep(100000)
```

- Vooreerst zie je dat een extra import moet uitvoeren van **log**
- Daarna moet 1 keer **log.set_labels()** worden uitgevoerd die aangeeft wat de hoofding (gewoon tekst) van die reeks getallen zal inhouden. Stel dat je meerdere verschillende waarden wil loggen op het zelfde moment, dan kunnen die waarden gescheiden worden van elkaar door ze in aparte kolommen op te nemen en de kolom te voorzien van een hoofding (titel). Dit kan achteraf eenvoudig in Excel worden gelezen via CSV.
- We zien dat er in de code een oneindige lus (While True:) is opgenomen maar dat we daar niets in doen. We laten daar de Micro:Bit slapen gedurende 100000 seconden. We kunnen gerust de Micro:Bit iets anders laten doen in de oneindige lus (zie later)
- Er is een aparte functie (zie eerder) opgenomen in de code met de naam: **log_data()** zonder parameter. Deze methode of functie wordt aangeroepen (uitgevoerd) op basis van een timer: `@run_every(h=1, min=20, s=30, ms=50)`.
- In die timer functie wordt een klein beepje afgespeeld zodat je de werking van de timer kan horen. De werking kan natuurlijk worden weggelaten.

Als je nu de gelogde data wenst te lezen dan kan je als volgt doen:

- Ontkoppel de Micro:Bit van de USB-kabel en wacht 10 seconden en koppel die terug aan (let wel de

logging zal opnieuw starten, maar de oude data is bereikbaar, nieuwe logging data zal maar bereikbaar zijn na een volledige reset van de Micro:Bit => ontkoppel van USB / wacht 10sec / en koppel terug aan de computer)

- Ga naar de Verkenner van de computer en zoek naar de extra harddrive die de Micro:Bit heeft gemaakt tijdens het koppelen:

MICROBIT (D:)

Netwerk

- Klik op die drive en zoek het bestand :
MY_DATA.HTM

DETAILS.TXT	22/03/2016 15:30	Tekstdocument	1 kB
MICROBIT.HTM	22/03/2016 15:30	Chrome HTML Do...	1 kB
MY_DATA.HTM	22/03/2016 15:30	Chrome HTML Do...	124 kB

- Open dit bestand door dubbelklik:

 micro:bit

micro:bit data log

Download

Copy

Update data...

Clear log...

Visual preview

This is the data on your micro:bit. To analyse it and create your own graphs, transfer it to your computer. You can copy and paste your data, or download it as a CSV file which you can import into a spreadsheet or graphing tool. [Learn more about micro:bit data logging.](#)

Time (seconds)	light
2.12	135
3.09	134
4.09	134
5.08	135
6.09	134
7.09	136

Met copy kan je de lijst exporteren naar Excel (CSV).
Met Visual preview kan je een grafische weergave bekijken van de data.

30.Starten en stoppen van Data logging en clear de data

Soms is het handig dat je het loggen van sensordata kan starten en stoppen. Dat het loggen niet onmiddellijk start van het moment de code in de Micro:Bit wordt gestart. Dit kan als volgt:

```
1 # Imports go at the top
2 from microbit import *
3 import log
4
5 log.set_labels('X','Y','Z', timestamp=log.MILLISECONDS)
6 logging = False
7
8 @run_every(ms=50)
9 def log_data():
10     if logging == True:
11         log.add({'X': accelerometer.get_x(), 'Y': accelerometer.get_y(), 'Z': accelerometer.get_z()})
12
13 # Code in a 'while True:' loop repeats forever
14 while True:
15     if button_a.was_pressed():
16         logging = not logging
17         if logging == True:
18             display.set_pixel(2,2,9)
19         else:
20             display.clear()
21     if button_b.was_pressed():
22         log.delete()
23         display.show(Image.SQUARE_SMALL)
24         sleep(500)
25         display.clear()
26         log.set_labels('X','Y','Z', timestamp=log.MILLISECONDS)
```

Veel succes