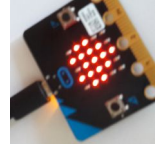


# Micro:bit

## programmeren met MicroPython



### Installeren van Mu-editor

Wil je meer mogelijkheden om de micro:bit te programmeren in Python, download en installeer dan Mu: <https://codewith.mu/> (zie ook <http://inigem.ugent.be/microbit.html>)

Met deze editor kan je de micro:bit aansturen met Python-code, werken met rijen,.. maar ook werken met bestanden. Zo kan je metingen uitvoeren, de resultaten opslaan in een bestand op de micro:bit (beperkte grootte wegens beperkt geheugen op de microcontroller) en deze gegevens nadien verwerken.

### Inleiding tot programmeren in python met de micro:bit:

Hieronder volgen een paar korte programma-fragmenten om kennis te maken met de syntax van MicroPython.

#### Tellen

Op de display van de micro:bit tellen we van 1 t.e.m. 9. → gebruik for-lus en sleep(), tonen op display.

```
from microbit import *

# vooraleer de lus te starten wordt eerst 2 seconden gepauseerd
sleep(2000)

# er wordt geteld van 1 t.e.m. 9 en de uitvoer komt telkens op de display
for teller in range(1,10):
    display.show(str(teller))      # met show kan je enkel stringwaarden tonen
    sleep(500)
```

Je kan ook vooruit tellen per twee (of meer) of achteruit tellen (de spronggrootte is dan negatief). Een voorbeeld zie je hieronder:

```
from microbit import *

# Wacht tot de A-knop is ingedrukt
while not button_a.is_pressed():
    sleep (20)                # wacht 20 ms

# Toon de oneven getallen 1, 3, 5, 7, 9 met telkens een seconde tussen

for teller in range (1,11,2): # van 1 tot 11 in stappen van 2
    display.show(str(teller)) # display kan enkel strings tonen,
    sleep(1000)              # we zetten het getal dus om naar een string met
    'str'

display.clear()              # Display leegmaken
# De micro:bit is klaar. Je moet op de reset-knop drukken (achterkant) om
# het programma opnieuw te laten lopen
```

## Werken met knoppen

(zie ook online oefening)

```
from microbit import *

while True:
    if button_a.is_pressed(): # als
        if button_b.is_pressed(): # als
            display.show (Image.ARROW_N)
        else: # anders
            display.show (Image.ARROW_W)
    elif button_b.is_pressed(): # anders als
        display.show (Image.ARROW_E)
    else:
        display.clear()

    sleep (20)                # ALTIJD even wachten...
```

## Gebruik van de accelerometer- simuleren waterpas

### Waterpas - met MicroPython

Het programma dat eerst grafisch online werd geprogrammeerd, zullen we nu programmeren in MicroPython.

```
from microbit import *

while True:          # voortdurend herhalen
    a_x = accelerometer.get_x() # we slaan de waarde op in een variabele
    if a_x < -20:
        display.show (Image.ARROW_E)
    elif a_x > 20:
        display.show (Image.ARROW_W)
    else:
        display.show (Image.SQUARE_SMALL)

    sleep (20)        # telkens even wachten tussen twee metingen!
```

## Werken met bestanden

```
from microbit import *

# Illustreert het gebruik van micro:bit bestanden
# Schrijft 10 keer A naar en bestand, gevolgd door 10 keer B


with open('ab.txt','w') as bestand: # Het bestand heet 'ab.txt'
    for i in range (0,10):
        bestand.write ('A' + '\r\n') # \r\n neemt een nieuwe lijn
                                     (\r op macOS en Linux)

    for i in range (0,10):
        bestand.write ('B' + '\r\n') # \r\n neemt een nieuwe lijn
                                     (\r op macOS en Linux)

    bestand.close()

# Na afloop moet je het bestand overbrengen van de micro:bit naar je
# computer om het te kunnen bekijken, bijvoorbeeld met 'kladblok'
```



Opm: Het bestand *ab.txt* staat op de micro:bit. Via  kan je dit bestand naar een map op jouw PC slepen en daarna bekijken met kladblok.

Opgelet: je kan dit bestand niet rechtstreeks zien in de verkenner, alleen in de mu-editor.

## Experiment



Wij zullen in deze sessie een fysica-proef uitvoeren waarbij we de micro:bit monteren op een fietswiel en tijdens het draaien van het wiel metingen registreren met de accelerometer.

(zie ook presentatie op <http://inigem.ugent.be/microbit.html>)

Werk je met Windows, dan heb je voor de seriële communicatie een extra driver nodig (niet voor Mac en Linux). Waar je die vindt en hoe te configureren vind je op <http://inigem.ugent.be/microbit.html>

```
from microbit import *

# wacht totdat er op de A-knop gedrukt wordt


while not button_a.is_pressed():
    sleep (20)

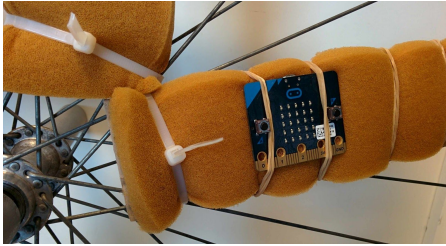
# tel af
for teller in range (3,0,-1):
    display.show(str(teller))
    sleep (1000)
display.clear()

# doe 1000 metingen (100 per seconde) en schrijf deze naar het bestand
'metingen.csv'
with open('metingen.csv', 'w') as bestand:
    for meting in range (0,1000):
        x = accelerometer.get_x()
        bestand.write (str(x) + '\r\n')      # \n op macOS en Linux
        sleep (10)
    bestand.close()

# toon dat je klaar bent met meten
display.show(Image.YES)
```



Na het meten staat het bestand 'meting.csv' op de microbit. Via de knop  in Mu sleep je dit bestand naar jouw PC. Dit bestand kan je nu eenvoudig openen in een rekenblad zoals Excel.

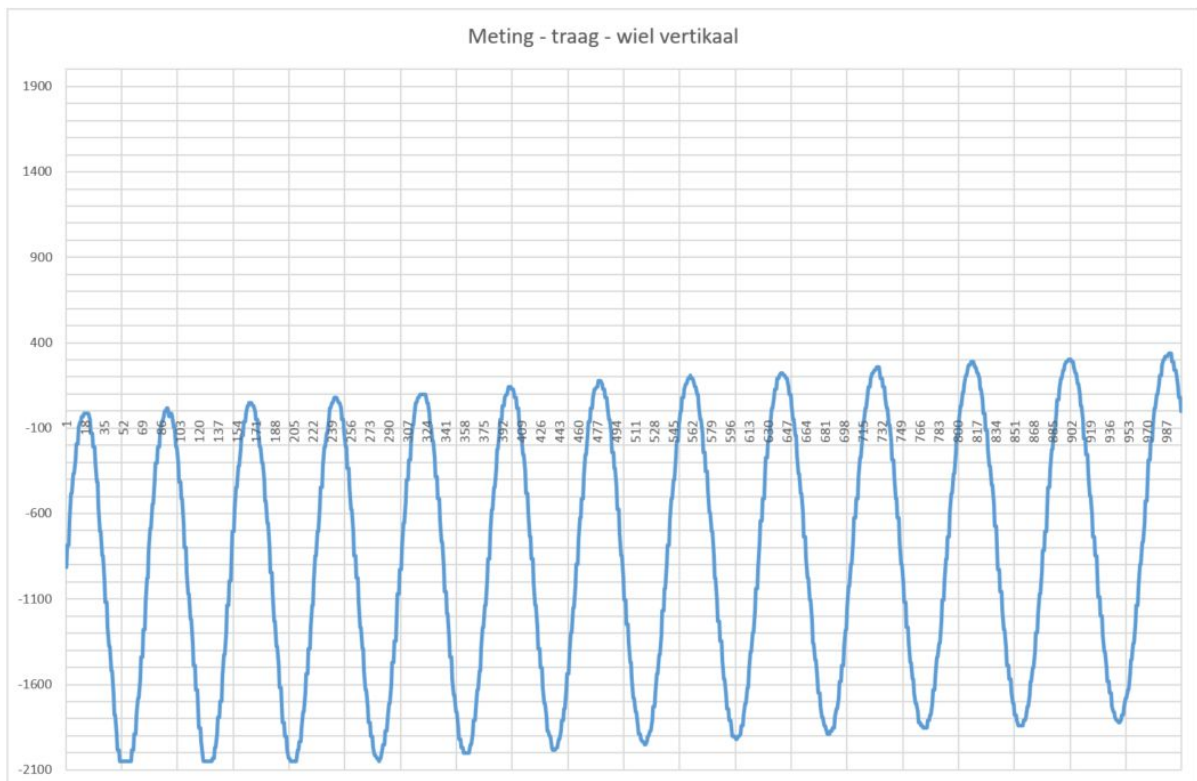


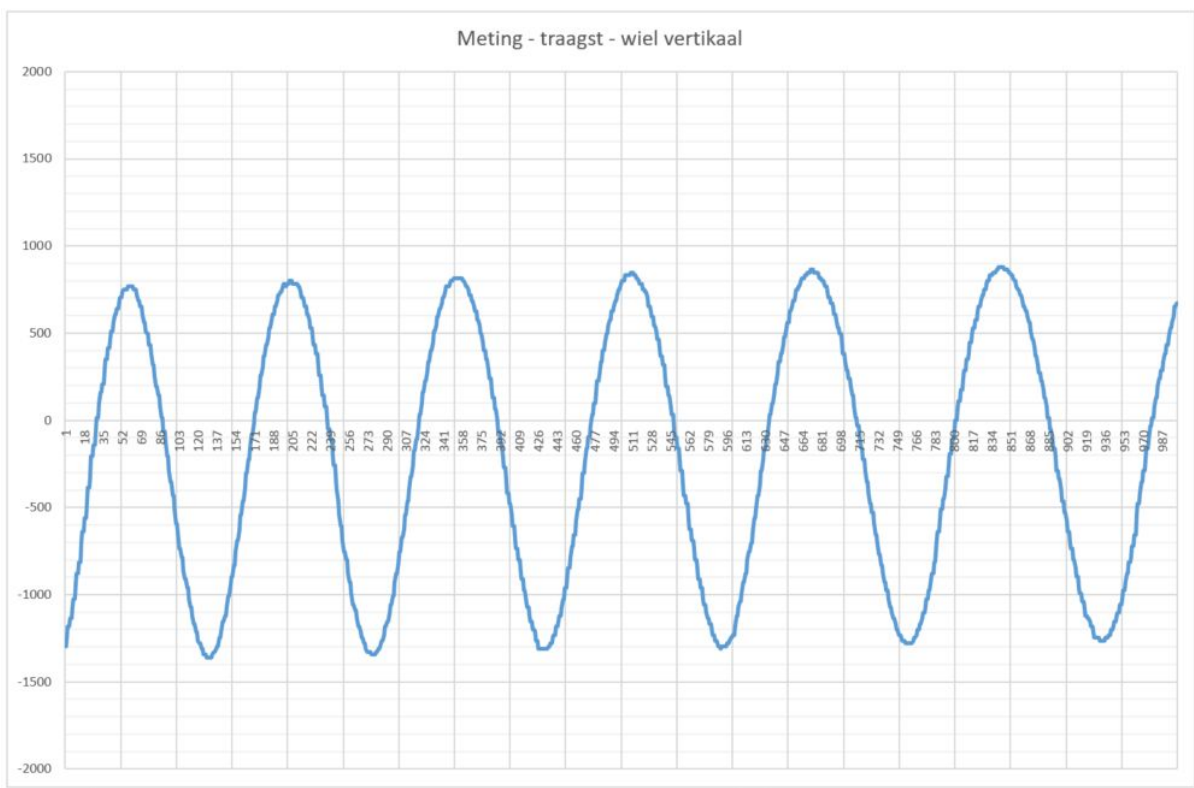
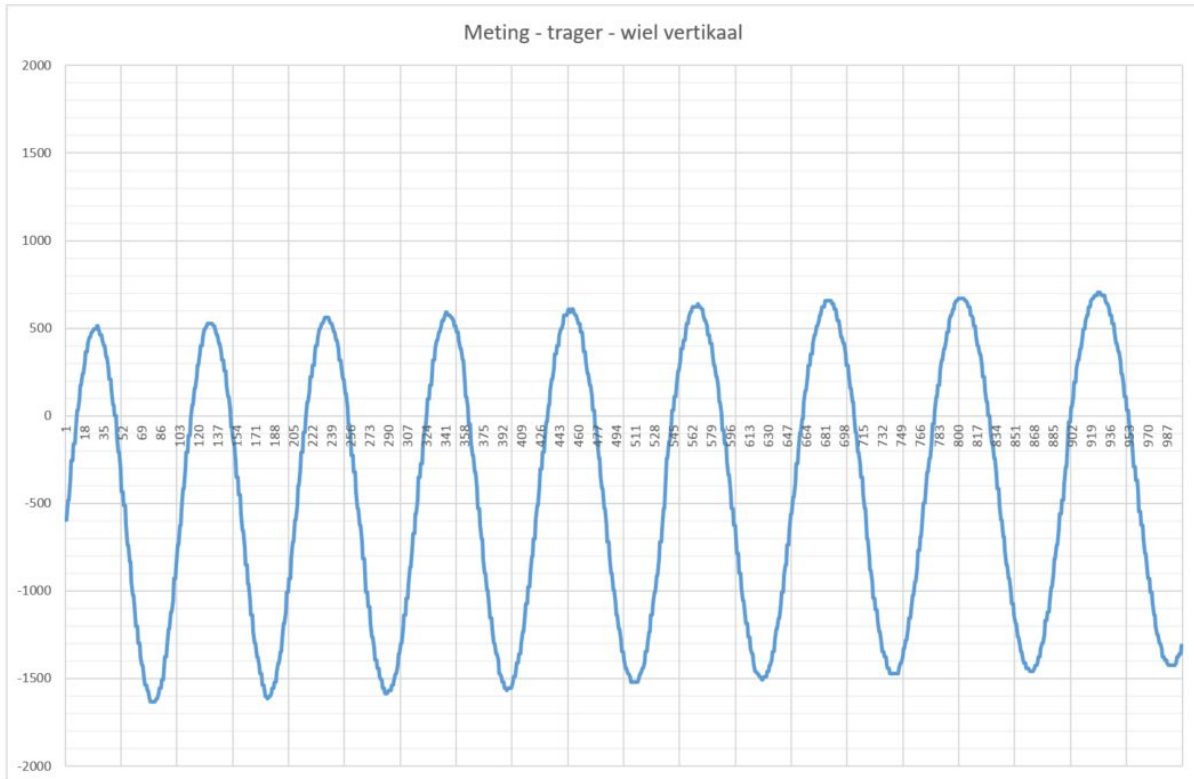
Aangezien de microbit zo gemonteerd is dat de X-as gericht is naar het centrum van de fietsas is de centripetale versnelling steeds gericht volgens de x-as en verwerken we de x- component van de metingen van de accelerometer.

Hieronder vind je grafieken van een aantal metingen uit bovenstaand experiment.

Het wiel draait telkens iets trager.

Bekijk het resultaat en vergelijk met wat jij hebt bekomen.





## En verder....

Wens je nog te experimenteren met de micro:bit in MicroPython dan vind je bijkomende documentatie op de website van de BBC microbitMicroPython

<https://microbit-micropython.readthedocs.io/en/latest/>