

In deze les maak je je eerste Python-programma's voor het op verschillende manieren aansturen van het micro:bit-display.

Deel 1: Alien encounter

Deel 2: Plaatjes weergeven

Doelen :

- Het display van de micro:bit gebruiken met de opdrachten `.show()`, `.scroll()` en `.show(image)`.

1. Controleer voordat je begint het volgende:

- Je gebruikt een **TI-84 Plus CE-T Python Edition met OS 5.7**.
- Je beheerst de beginselen van het programmeren in Python.
- De micro:bit is verbonden met de rekenmachine.
- Je hebt de instructies gevolgd voor het instellen van de micro:bit, zie:

<https://education.ti.com/nl/teachers/microbit>

Deze procedure hoef je maar één keer te volgen, maar let op eventuele updates.



2. Vanuit het rekenmachinescherm kun je controleren of de benodigde bestanden zijn geïnstalleerd:

Druk daarvoor op **[2nd] [+]** en kies dan

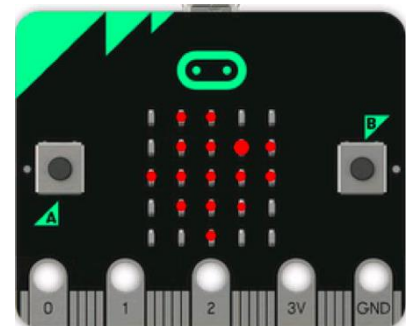
2: Mem Management > AppVars.

Hier kun je de bestanden vinden.

NORMAL FLOAT AUTO a+bI RADIAN MP	
RAM FREE	101702
ARC FREE	1250K
MB_DISP	4689
MB_GROVE	3947
MB_MUSIC	2452
MB_NEOPX	1860
MB_PINS	2144
MB_RADIO	2568
MB_SENSR	4460
▶ MICROBIT	1964

3. Als alles goed is gegaan ziet je micro:bit er zo uit, nadat je hem hebt aangesloten op je rekenmachine:

Op het display van de micro:bit zie je het TI-logo, de staat Texas met een heldere stip op de plek van Dallas.



4. **Deel 1: alien encounter**

Zoals gebruikelijk bij een eerste programma begin je met een bericht op het micro:bit-display.

Begin in de Python-editor met een nieuw programma (**F3<New>**). We noemen het GREETING.

Als je in OS 5.7 **<Fns...> Modul** kiest, verschijnt boven de F5-toets **<Add-On>** waarmee je extra modules kunt vinden. (Als dit niet op het scherm verschijnt moet je de rekenmachine updaten naar OS 5.7)

Tip: Als de mededeling 'micro:bit not connected' verschijnt, koppel dan de micro:bit los en sluit hem opnieuw aan.



5. Kies **<F4 : Add-On>** en kies dan de micro:bit-module om **from microbit import *** naar het programma te kopiëren.

6. Als je nu opnieuw naar het modules-menu gaat (met **<Fns...> Modul** of met **[math]**) zie je een extra optie: **micro:bit**. Telkens als je uit het micro:bit menu-opties kiest, kun je deze lijst uitbreiden.

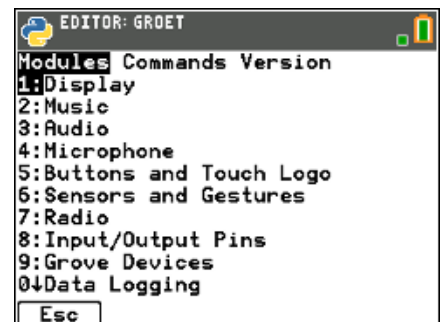
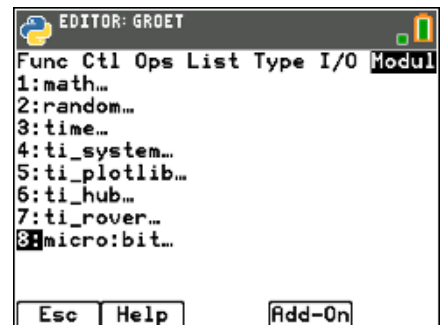
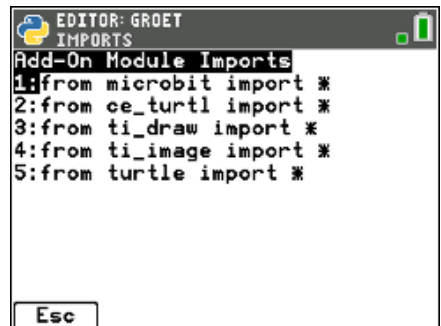
Opmerking: de [math] toets is een shortcut voor <Fns...> Modul, maar toont niet het Add-On-tabblad.

7. Het **micro:bit**-submenu bevat alle opdrachten voor het programmeren van de micro:bit in aparte submodules.

Elk van deze menu-opties voegt een importopdracht toe aan het Python-programma.

Dit komt omdat al deze opties in aparte modules zijn ondergebracht.

Selecteer nu het **Display**-menu ...





8. In het Python-programma is nu een nieuwe regel toegevoegd:
from mb_disp import *

```
EDITOR: GROET  
PROGRAM LINE 0003  
from microbit import *  
from mb_disp import *  
-  
Fns... | a A # | Tools | Run | Files
```

9. Open het **[math]**-menu opnieuw. Er is nu een **Display**-optie bijgekomen.
Selecteer **Display ...**

```
EDITOR: GROET  
Modul  
1:math...  
2:random...  
3:time...  
4:ti_system...  
5:ti_plotlib...  
6:ti_hub...  
7:ti_rover...  
8:micro:bit...  
9:display...  
Esc
```

10. Het **Display**-menu (zie hiernaast) bevat de opdrachten waarmee je het micro:bit-display kunt aansturen.
Dit display bestaat uit 5x5 rode leds in het midden van de micro:bit.

Selecteer nu **.show(val)**

```
EDITOR: GROET  
Display Images  
1:.show(val)  
2:.scroll(val)  
3:.clear()  
4:.set_pixel(x,y,val)  
5:var=Image(':::':':':':':':':')  
6:var=.read_light_level()  
Esc
```

11. De opdracht wordt ingevoegd als **display.show()**
Typ tussen de haakjes: "greetings earthlings" inclusief de aanhalingstekens.

Tip: Druk op [2nd] [alpha] om 'alpha lock' in te schakelen. De aanhalingstekens staan boven de [+]-toets.

Voer het programma uit. De tekst wordt letter voor letter op het op het display getoond, maar het is moeilijk leesbaar.



```
EDITOR: GROET
PROGRAM LINE 0006
from microbit import *
from mb_disp import *

display.show("greetings earthlings")
```

12. Een betere manier om tekst weer te geven is met **display.scroll(...)**. Ook deze opdracht staat in het display-menu.

Om het bericht sneller te laten zien kan een extra parameter worden toegevoegd:

display.scroll("greetings earthlings", 200).

Het getal 200 staat voor een wachttijd van 200 milliseconden tussen de letters. Standaard is dit 400, dus 200 laat het bericht tweemaal zo snel lopen.



```
EDITOR: GROET
PROGRAM LINE 0002
from microbit import *
from mb_disp import *

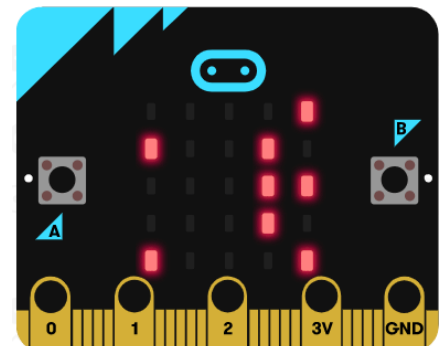
display.scroll("greetings earthlings")
```

13. Hiernaast zie je het resultaat met **display.scroll("greetings earthlings", 200).**

Opmerking: Je kunt nog een tweede parameter toevoegen:
display.scroll("greetings earthlings", 200, wait).

Als wait = True wacht het programma met de volgende opdracht totdat het bericht volledig is getoond.

Als wait = False wordt de volgende programmaregel direct uitgevoerd.



14. Deel 2: Hartslag

In dit gedeelte laten we zien hoe je een plaatje op het display kunt tonen.

Maak een nieuw programma met de naam HART.

(Kies hiervoor **Files – New**)

Begin net als in het vorige onderdeel met het importeren van twee modules:

```
from microbit import *  
from mb_disp import *
```



```
EDITOR: HART  
PROGRAM LINE 0003  
from microbit import *  
from mb_disp import *
```

15. Om een afbeelding van een hart op het display te tonen gebruiken we de opdracht **display.show(...)** die weer staat bij **[math] display...**

Als de cursor tussen de haakjes staat kun je uit het submenu **[math] display... Images... HEART** selecteren.



```
EDITOR: HART  
Display Images  
1: HEART  
2: HEART_SMALL  
3: HAPPY  
4: SMILE  
5: SAD  
6: CONFUSED  
7: ANGRY  
8: ASLEEP  
9: SURPRISED  
0: SILLY  
Esc
```

16. In het programma wordt dan de string **"Image.HEART"** ingevoegd.

*Opmerking: In dit menu staan 36 plaatjes, maar je kunt ook zelf een afbeelding maken met **var = Image(...)**.*

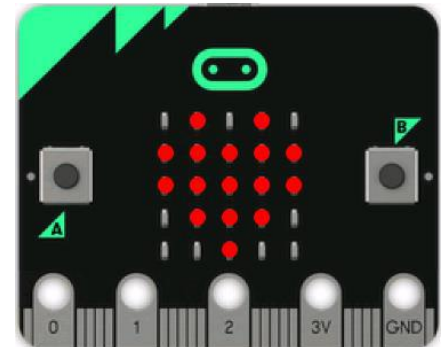
Zie hiervoor de micro:bit-documentatie die online beschikbaar is.



```
EDITOR: HART  
PROGRAM LINE 0005  
from microbit import *  
from mb_disp import *  
  
display.show("Image.HEART")
```

17. Voer het programma uit, dan zie je een hartvorm op het micro:bit-display verschijnen.
Het hartsymbool blijft op het display totdat je weer een andere opdracht geeft, ook als het programma gestopt is.

(Je kunt het verwijderen door op de reset-knop op de achterkant van de micro:bit te drukken.)



18. Ga terug naar de Python-editor en voeg de opdracht toe om het kleine hartsymbool weer te geven:
display.show("Image.HEART_SMALL").

```
EDITOR: HART
PROGRAM LINE 0006
from microbit import *
from mb_disp import *

display.show("Image.HEART")
display.show("Image.HEART_SMALL")
)
```

19. Voer het programma uit. Dan zie je eerst het grote hart en dan direct daarna het kleine.

20. We maken nu een lus om het hart te laten "kloppen".
Voeg vóór de twee opdrachten de volgende opdracht toe:
while not escape():

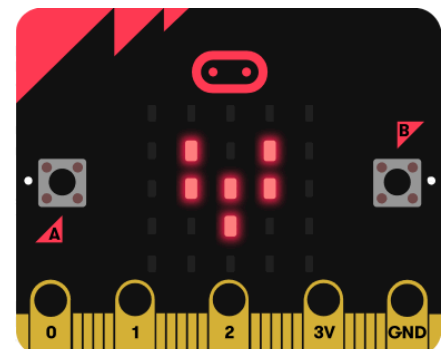
Je kunt deze vinden in **[math] ti_system...**

Zorg ervoor dat de twee display-opdrachten beide inspringen.
(Dit zorgt ervoor dat beide opdrachten in de lus worden uitgevoerd.)

```
EDITOR: HART
PROGRAM LINE 0007
from microbit import *
from mb_disp import *

while not escape():
    display.show("Image.HEART")
    display.show("Image.HEART_SMALL")
)
```

21. Voer het programma weer uit. Het hart klopt nu.
Je kunt het programma onderbreken met de **[clear]**-toets.



22. Je kunt de snelheid van de hartslag veranderen door een waarde voor de delay-parameter te kiezen.
Bijvoorbeeld zoals hiernaast

```
EDITOR: HART
PROGRAM LINE 0005
from microbit import *
from mb_disp import *

while not escape():
    display.show("Image.HEART", 400)
    display.show("Image.HEART_SMALL", 500)
```

Fns... | a A # | Tools | Run | Files

23. **Extra:** Probeer ook eens verschillende gezichten uit.
Gebruik dezelfde structuur als het hartslagprogramma, maar gebruik nu verschillende gezichtsuitdrukkingen.

