

Unit 6: micro:bit

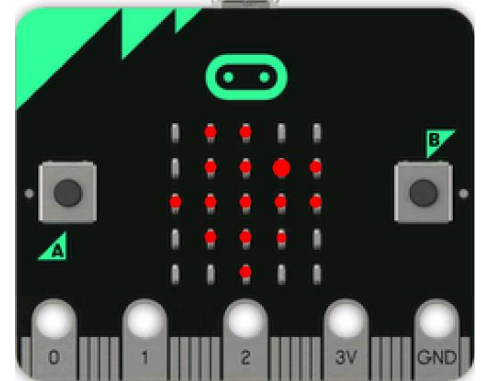
Oefenblad 3: Lichtmeting

In deze les gebruiken we de lichtsensor van de micro:bit en slaan de data op in een TI-84 Plus CE-T Python Edition-lijst voor verder onderzoek.

Doelen:

- Het aflezen en weergeven van de lichtsensor op de micro:bit;
- Exporteren van de data naar de TI-84 Plus CE-T Python Edition;
- Meetgegevens van de micro:bit analyseren.

1. Met de leds van het display van de micro:bit kun je de lichtsterkte van de omgeving meten.
Deze leds kun je dus niet alleen voor uitvoer gebruiken maar ook voor invoer.



2. Start een nieuw programma met de naam LICHT en begin weer met de gebruikelijke import-regels en de **while**-lus.

```
EDITOR: LICHT
PROGRAM LINE 0005
from ti_system import *
from microbit import *

while not escape():
  ..
```

3. Voeg nu de **from mb_disp import ***-regel toe om de display-functies te kunnen gebruiken.

In de **while**-lus plaatsen we de regel:

s = display.read_light_level()
(= **.read_light_level()** kun je vinden met **[math]** display...)

Voeg de regel **print("s =", s)** toe om te zien wat de resultaten zijn.

```
EDITOR: LICHT
PROGRAM LINE 0008
from ti_system import *
from microbit import *
from mb_disp import *

while not escape():
  ..s=display.read_light_level()
  ..print("s =",s)
```



4. **<Run>** het programma richt de voorkant van de micro:bit op een lichtbron. Beweeg de micro:bit van en naar de lichtbron en kijk hoe de waarden op het TI-84-scherm veranderen. Deze waarden variëren tussen 0 en 255.

*Opmerking: Als de waarden niet variëren zoals hiernaast, kun je in de while-lus een **sleep(100)** commando toevoegen om het proces te vertragen.*

Zoals verwacht zijn de uitkomsten kleiner als er minder licht gemeten wordt.

We breiden het programma uit zodat we de meetwaarden kunnen verzamelen en er een grafiek van kunnen maken.

Druk op **[clear]** om het programma stoppen en ga terug naar de **<Editor>**.

5. Creëer twee lege lijsten vóór de **while**-lus:
`tijd = []` en `lichtsterkte = []`
 (Je kunt natuurlijk ook een kortere naam voor deze lijsten gebruiken (bijv. `ls` voor `lichtsterkte`).
 Voeg ook de regel `t = 0` toe. De variabele `t` telt het aantal metingen.

6. Voeg in de while-lus de regel `t = t+1` toe om de tijd telkens met 1 te verhogen.

Opmerking: in Python kun je dit ook schrijven als: `t += 1`

```

PYTHON SHELL
s = 184
s = 182
s = 176
s = 171
s = 163
s = 157
s = 121
s = 98
s = 51
s = 0
>>> |
  
```

```

EDITOR: LICHT
PROGRAM LINE 0011
from ti_system import *
from microbit import *
from mb_disp import *

tijd=[]
lichtsterkte=[]
t=0
while not escape():
    s=display.read_light_level()
    print("s =",s)
  
```

```

EDITOR: LICHT
PROGRAM LINE 0011
from ti_system import *
from microbit import *
from mb_disp import *

tijd=[]
lichtsterkte=[]
t=0
while not escape():
    s=display.read_light_level()
    print("s =",s)
    t=t+1
  
```



7. Met de opdrachten **tijd.append(t)** en **lichtsterkte.append(s)** voeg je de waardes van t en s toe aan de lijsten tijd en lichtsterke.

.append() kun je vinden met <Fns...> List.

```

EDITOR: LICHT
PROGRAM LINE 0014

tijd=[]
lichtsterkte=[]
t=0
while not escape():
  s=display.read_light_level()
  print("s =",s)
  t=t+1
  tijd.append(t)
  lichtsterkte.append(s)

```

8. We willen de metingen elke seconde doen dus moeten we tussen twee opeenvolgende metingen 1 seconde wachten. Dit kan met **sleep(1000)**.

sleep() vind je met [math] micro:bit... Commands en gebruikt milliseconden.

```

EDITOR: LICHT
PROGRAM LINE 0015

tijd=[]
lichtsterkte=[]
t=0
while not escape():
  s=display.read_light_level()
  print("s =",s)
  t=t+1
  tijd.append(t)
  lichtsterkte.append(s)
  sleep(1000)

```

9. Aan het eind van het programma, nadat de while-lus verlaten wordt, komen nog twee regels om de lijsten op te slaan als TI-84-lijsten.

We gebruiken weer **store_list()**. (dit kun je vinden met [math] ti_system...)

De namen tussen de aanhalingstekens zijn de exacte namen die in de TI-84 omgeving gebruikt worden. Deze moeten in hoofdletters zijn en maximaal 5 letters bevatten.

```

EDITOR: LICHT
PROGRAM LINE 0017

while not escape():
  s=display.read_light_level()
  print("s =",s)
  t=t+1
  tijd.append(t)
  lichtsterkte.append(s)
  sleep(1000)
store_list("TIJD",tijd)
store_list("LICHT",lichtsterkte)

```



10. **<Run>** het programma. Start met de micro:bit dichtbij een lichtbron en beweeg de micro:bit langzaam verder weg van de lichtbron totdat de lichtsterkte lager is dan 10.

Druk op **[clear]** om het programma te stoppen.

11. Sluit de Python-app en ga naar met **[stat] edit** naar de list-editor van de TI-84.

Voeg de twee lijsten (TIJD en LICHT) toe en bekijk de data.

L4	L5	L6	TIJD	LICHT	7
-----	-----	-----	1	229	
			2	225	
			3	206	
			4	169	
			5	129	
			6	89	
			7	52	
			8	47	
			9	38	
			10	15	
			11	0	

TIJD(1)= 1

Met **stat-plot** kun je nu bijvoorbeeld een scatterplot maken.

