



In deze les gaan we de lichtsensor van de Ti-Innovator Hub onderzoeken en gebruiken we de het **Disp** commando om de waarden weer te geven.

Doel:

- Uitlezen van de BRIGHTNESS sensor
- Introductie van de **While** loop

In de vorige lessen hebben we alleen opdrachten naar de TI-Innovator™ Hub gestuurd die de ingebouwde apparaten zoals LIGHT, SOUND en COLOR aanstuurden.

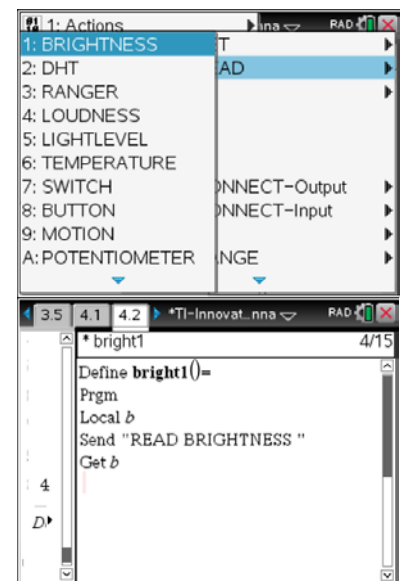
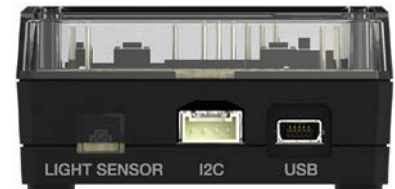
In deze unit gaan we werken met de ingebouwde lichtsensor, en gebruiken we de waarde in het programma om een lichtmeter te bouwen. De lichtsensor geeft een waarde tussen 0 en 100 (met decimalen).

Om de waarde van de lichtsensor te krijgen, hebben we twee commando's nodig:

- **Send "READ BRIGHTNESS"**
- **Get <var>**

Het programma:

1. Maak een nieuw programma en noem het BRIGHT1.
2. We gebruiken de variabele **b** voor de waarde van de lichtsensor. We declareren deze variabele lokaal (dit is niet noodzakelijk).
3. Selecteer via **menu > Hub > Send "READ... > BRIGHTNESS**. Druk op enter.
4. Selecteer vervolgens **menu > Hub > Get**.
5. Type de variabele **b** (zonder haakjes of aanhalingstekens).



Hoe werkt het:

- **READ BRIGHTNESS** geeft de opdracht aan de TI-Innovator Hub om de helderheid of wel de lichtsterkte (brightness) te meten. Deze waarde wordt in de innovator in een buffer opgeslagen.
- **Get b** zorgt er voor dat de waarde die in de buffer is opgeslagen, wordt doorgegeven naar de TI-Nspire en in de variabele **b** wordt gestopt.

Tip: Een 'buffer' is een geheugenplaats op de TI-Innovator Hub waar tijdelijk waarden worden opgeslagen. Deze geheugenplaats wordt bij elk READ commando aangepast. Daarom moet na elk READ commando een GET commando komen om de waarde door te geven aan de rekenmachine. Het is mogelijk om een reeks metingen uit te voeren met TI-Innovator Hub en deze in een lijst op te slaan en deze voor latere analyse te gebruiken. Dat gaat echter te ver voor deze introductie.



While Loop:

De **While...EndWhile** loop (menu > **Besturing** >) wordt gebruikt om een hoeveelheid code te herhalen zolang (while !) de conditie *waar* is. Een *conditie* is een logische expressie die de waarde *waar* (*true*) of *niet waar* (*false*) geeft. De relationele operatoren en logische operatoren kun je op de rekenmachine vinden met ctrl = .

De relationele operatoren zijn =, ≠, <, >, ≤, en ≥.

De logische operatoren zijn **and**, **or**, **not**, en **xor**.

Deze operatoren kunnen gezamenlijk gebruikt worden om condities te maken zoals : **x>0** and **y>0**.

We gaan een eenvoudige **While** loop gebruiken die stopt als de lichtsterkte (BRIGHTNESS) lager is dan 1. Om het programma te beëindigen hoef je dan alleen de lichtsensor op de TI-Innovator Hub te bedekken met je hand.

Een While Loop toevoegen:

6. Voeg voor het **Send** commando in je programma de volgende regels toe:

- **b:=2**
- **While b>1** (gebruik ctrl-= key om > te selecteren)

Deze regels starten de loop. Zolang de conditie (voorwaarde) **b>1** *waar* is, blijft de loop de lichtsensor uitlezen. Zodra de voorwaarde *onwaar* is (b.v. als er geen licht meer op de sensor valt omdat je hand het bedekt) wordt de loop beëindigd.

```

bright1 6/7
Define bright1()=
Prgm
Local b
b:=2
While b>1
Send "READ BRIGHTNESS "
Get b
EndWhile
EndPrgm

```

Als je **While...EndWhile** selecteert van het Control menu, dan wordt het **EndWhile** commando op de verkeerde plaats in je programma gezet!

Het moet na het **Get** commando zoals in de figuur hierboven. Je kunt het **EndWhile** commando selecteren, knippen (ctrl X) en plakken op de juiste plaats (ctrl V). Je kunt het ook wegehalen met delete en dan op de juiste plaats opnieuw intikken.

Als je **While** met de hand hebt ingetikt dan moet ook de **EndWhile** van de **While** loop worden ingevoerd met de hand. De **EndWhile** moet na het **Get** commando .

Tip: Het is aan te raden dat een programma altijd een mogelijkheid heeft om uit een loop te komen. Bedenk dat elke 'control structure' zijn eigen **End...** heeft. In grotere programma's zullen veel verschillende **End...** commando's zijn. De processor weet welk **End...** hoort bij welke structuur/loop. De programmeur moet de juiste code ontwerpen.

7. Voeg een **Disp** commando (menu > **I/O** > **Disp**) toe na het **Get** commando maar voor **EndWhile** (zie voorbeeld).
8. Toon de waarde van **b**.

```

bright1 6/7
Define bright1()=
Prgm
Local b
b:=2
While b>1
Send "READ BRIGHTNESS "
Get b
Disp b
EndWhile
EndPrgm

```



10 Minutes of Code

TI-NSPIRE CX MET DE TI-INNOVATOR™ HUB

9. Voer het programma uit met ctrl-R (zorg dat er een innovator is bevestigd aan de rekenmachine!).
 - Je zult een rij met waarden over het scherm zien rollen. Deze waarden veranderen met de helderheid van het licht zoals deze wordt gemeten door de sensor.
10. Om de loop (en het programma) te stoppen bedek je de lichtsensor met je hand zodat de waarde van de helderheid zakt onder de 1.

UNIT 3: OEFENBLAD 1

DOCENTENHANDLEIDING

```
2.30117  
1.97766  
1.45883  
1.22078  
1.07428  
0.665324  
Done
```

```
Define bright1()  
Prgm  
Local b  
b:=2  
While b>1  
Send "READ BRIGHTNESS"  
Get b  
Disp b  
EndWhile  
EndPrgm
```

Tip: Lokale variabelen bestaan in het programma dat uitgevoerd wordt. Ze zijn niet beschikbaar in het document. Als een variabele niet lokaal gedeclareerd wordt, dan wordt de variabele gemaakt in het document. Hij is dan ook beschikbaar buiten het programma. Dit kan soms prettig zijn en soms niet.