Lektion 3: Helligkeit, IF und WHILE — Übung 1: Helligkeitsmessungen

In dieser Übung untersuchen wir den eingebauten Lichtsensor BRIGHTNESS und verwenden die **Disp**-Anweisung um die Messwerte des Sensors anzuzeigen.

Lernziele:

- Die Messwerte des BRIGHTNESS-Sensors ablesen
- Die **While**-Schleife kennen lernen

In den vorigen Übungen haben wir Befehle an den TI-Innovator™ Hub gesendet, die seine eingebauten Geräte (LIGHT, COLOR und SOUND) ansprechen.

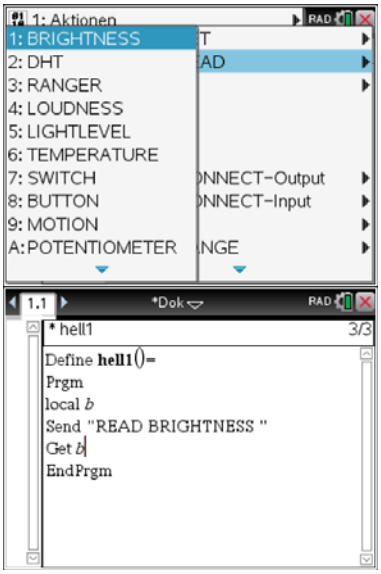
Jetzt werden wir mit dem eingebauten Lichtsensor arbeiten und die Werte unseres Programms verwenden um einen „Helligkeitsmesser“ zu erzeugen. Dieser Sensor erzeugt Werte in Dezimalform zwischen 0 und 100.

Den Helligkeitswert vom TI-Innovator™ erhalten wir über ZWEI Anweisungen:

- **Send “READ BRIGHTNESS”**
- **Get** <*Variable*>

Der Beginn des Programms:

1. Beginne ein neues Programm mit dem Namen *hell1*.
2. In der Variablen **b** werden wir die BRIGHTNESS-(Helligkeits-)Werte speichern. Wir deklarieren **b** mit **Local** als lokale Variable. (Das ist optional).
3. Wähle ***menu > Hub > Send “READ… > BRIGHTNESS***. Drücke enter.



4. Wähle nun **menu > *Hub* > Get**.
5. Tippe den Variablennamen **b** (hier ohne Klammern).

Wie das funktioniert:

- **READ BRIGHTNESS** sagt dem TI-Innovator™ Hub, dass er den Helligkeitswert messen und diesen Wert in einem eingebauten Pufferspeicher speichern soll.
- Mit der Anweisung **Get b** holt man den Wert aus dem Pufferspeicher des TI-Innovator™ Hub und überträgt ihn auf die Variable **b** des TI-Nspire™ CX. Dazu kann jeder legale Variablenname verwendet werden.

> **Hinweis:** Ein “Puffer” ist eine Speichereinheit im TI-Innovator™ Hub die einen Wert temporär speichert. Er wird immer dann aufgefrischt, wenn eine neue *READ*-Anweisung ausgeführt wird. Daher ist sehr zu empfehlen, dass gleich nach dem *READ* der Variablenwert vom TI Innovator™ Hub über das *Get* in einen Variablenwert auf dem Rechner übertragen wird. Man kann eine Menge Daten am TI-Innovator™ Hub sammeln und für spätere Analyse der Daten in einer Liste speichern. Aber das liegt außerhalb dieser Einführung.

Die While-Schleife:

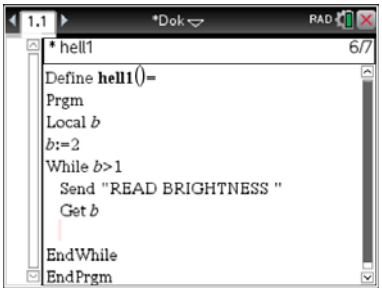
Mit der **While…EndWhile**-Schleife (***menu* > *Steuerung* >**) wird ein Block von Anweisungen so lange ausgeführt wie eine *Bedingung* wahr ist. Eine *Bedingung* ist ein logischer Ausdruck, der den Wert wahr oder falsch annehmen kann. Die Relationszeichen und die logischen Operatoren finden sich über die ⌜|≠≥▸⌝ -Taste (ctrl =).

Die Relationszeichen sind **=**, **≠** **<**, **>**, **≤**, and **≥**. Die logischen Operatoren sind: **and**, **or**, **not**, and **xor**.

Damit können Bedingungen zusammengesetzt werden wie z.B.: **x>0 and y>0.**

Wir werden eine einfache **While**-Schleife verwenden, die anhält, sobald der Helligkeitswert kleiner als 1 ist. Für den Abbruch des Programms muss man dann nur den Sensor am Ende des TI-Innovator™ Hub mit der Hand abdecken.

Einbau der While-Schleife:



6. Füge vor der **Send**-Anweisung die beiden folgenden Anweisungen ein:
   - **b:=2**
   - **While b>1** (mit ctrl = kommst du zum >-Zeichen)

   Diese Anweisungen initialisieren die Schleife. So lange als b>1 wahr ist, wird in der Schleife die Helligkeit über den Sensor gemessen. Sobald die Bedingung nicht erfüllt ist – also wenn kein Licht mehr auf den Sensor fällt, da er mit der Hand bedeckt wird – wird die Schleife und damit das Programm beendet.

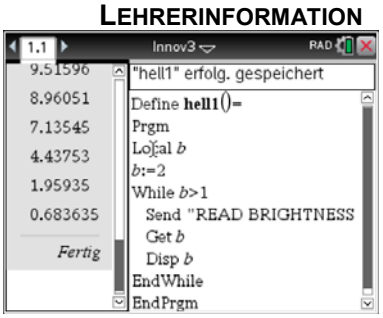
Wenn du **While…EndWhile** *über das Steuerungsmenü einfügst, dann wird das* **EndWhile** *in deinem Code an der falschen Stelle stehen!*

Es gehört <u>hinter</u> die **Get***-Anweisung wie oben gezeigt. Du kannst sie aktivieren, ausschneiden (ctrl X) und an der richtigen Stelle einfügen (ctrl V), oder auch löschen und an der richtigen Stelle hinschreiben.*

Das **EndWhile** der **While**-Schleife muss auch eingegeben werden. Füge das nach der **Get**-Anweisung als Ende der **While**-Schleife ein.

> **Hinweis:** Es macht immer Sinn, im Programm einen Ausstieg aus der Schleife vorzusehen. Jede Kontrollstruktur hat ihr *eigenes* **End…**. In umfangreicheren Programmen können da viele **End**-Anweisungen zusammenkommen. Der Rechner „weiß“, welches **End** zu welcher Konstruktion gehört, aber es liegt am Programmierer, den richtigen Code zu entwerfen.

7. Füge die **Disp**-Anweisung <u>nach</u> dem **Get** aber <u>vor</u> dem **EndWhile** über ***menu* > *E/A* > Disp** ein.
8. Damit wird der Wert der Variablen **b** auf den Schirm des *Calculators* gebracht.



9. Speichere das Programm (ctrl B) und führe es im *Calculator* mit angeschlossenem TI-Innovator™ Hub.
   - Du solltest eine Folge von Werten im *Calculator* entstehen sehen, die sich abhängig von der Helligkeit, die am Sensor empfangen wird, ändert.
10. Um die Schleife (und damit auch das Programm) zu beenden, decke den Lichtsensor auf der Seite des TI-Innovator™ Hub zu, so dass der angezeigte Helligkeitswert unter 1 liegt.

Hinweis: Lokale Variable existieren nur während einer Programmausführung. Sie werden nicht im aktuellen Problem erzeugt. Wenn eine Variable in einem Programm nicht als **local** deklariert wird, dann wird sie bei Programmausführung auch im Problem geschaffen. Das kann ein manchmal vorteilhaft, sehr oft aber auch problematisch sein.