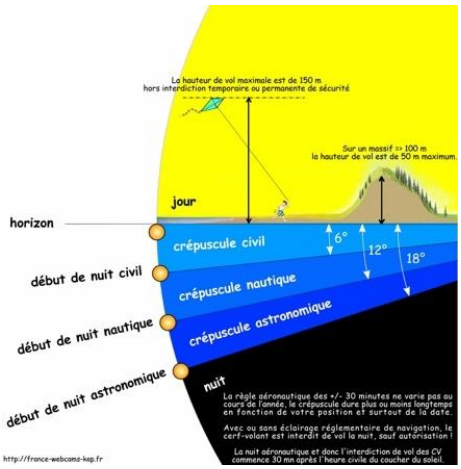
Lektion 6 : Verwendung der Module ti_hub & ti_rover | **Übung 1 : Die eingebauten Sensoren des Hub**

In dieser ersten Übung von Lektion 6 wird gezeigt, wie das Modul **ti_hub** zur Steuerung der in den **TI-Innovator™- Hub** integrierten Geräte verwenden wird.

Lernziele :

- Entdeckung des Moduls **ti-hub.**
- Erstellen eines Programmes, das die eingebauten Sensoren des **TI-Innovators™** nutzt.

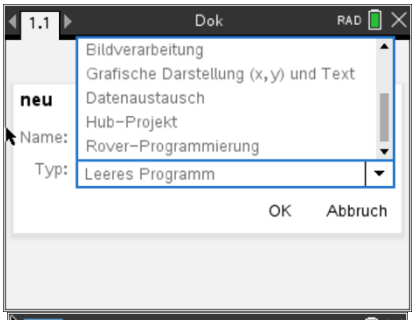
In dieser Übung wird die **ti_Innovator**-Bibliothek verwendet, um Helligkeitsänderungen zu messen, so dass man einen Dämmerungsschalter simulieren kann oder um eine Messreihe bei Sonnenaufgang oder Dämmerung aufzuzeichnen.
Anschließend wird diese Bibliothek mit den bereits bekannten Modulen (**ti_plotlib** & **ti_system**) kombiniert, um die Messungen auch grafisch darzustellen.



Das Programm entspricht dem folgenden einfachen Algorithmus:

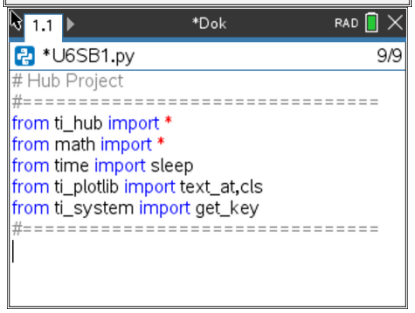
```
Messung der momentanen Helligkeit:
Lum0 ← Messung ± (Abweichung ?)
Veränderung der Helligkeit (hellere Lampe,
Abschattung des Lichtsensors) :
Lum1 ← Messung
   Falls Lum1 > Lum 0 :
   |  Die RGB LED für 2s rot einschalten
   Oder falls Lum0 < Lum1 :
   |  Die RGB LED für 2s grün einschalten
    Oder : nichts tun
```

Anlegen eines neuen Programmes mit dem Namen **U6SB1**.

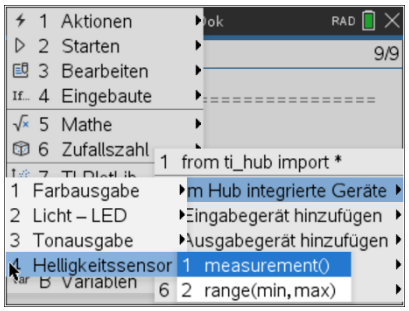


Dieses Programm muss das Modul **ti_hub** enthalten. Dafür gibt es mehrere Möglichkeiten :

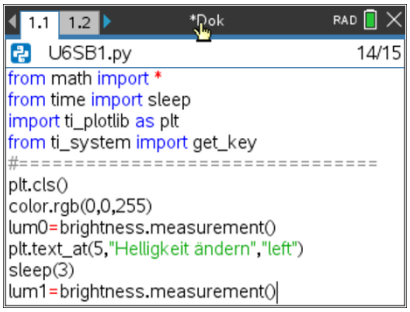
a) Man wählt von vorneherein unter Typ das **Hub Project**. Dadurch werden mehrere Module bzw. auch nur einzelne Anweisungen aus diesen Modulen in das Programm mit eingebunden. Ob das immer ausreicht, ist fraglich. Allerdings kann man die Modul-Liste nachträglich verändern.
b) Es ergibt sich das untere Bild.
c) Oder man wählt – wie bisher – ein Programm ohne Vorauswahl. Das ist die Voreinstellung, wenn man ein neues Python-Programm anlegt. Allerdings muss dann das Modul **hub** extra geladen werden.

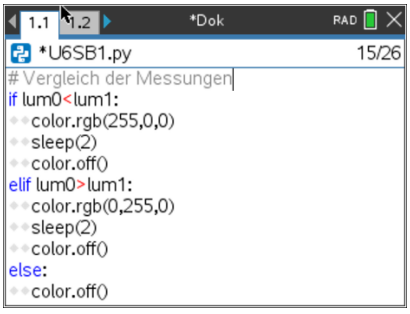


Für das Programm wird der im **TI-Innovator** eingebaute Helligkeitssensor **Brightness** verwendet ebenso wie die eingebaute **RGB-LED** (Farbausgabe). Über das Menü **ti-hub** gelangt man zu diesen im Hub integrierten Geräten mit ihren entsprechenden Untermenüs, hier am Beispiel des Helligkeitssensors.



Das Programm :





- Um Schwierigkeiten bei den Grafikbefehlen zu umgehen, wurde das Modul **ti_plotlib** komplett eingefügt (vergl. mit dem letzten Bild auf der ersten Seite).
- Nun kannn man den Bildschirm mit **plt.cls()** aus dem Modul **ti_plotlib** löschen.
- Dann wird die LED mit blauem Licht eingeschaltet.
- Anlegen einer Variablen **lum0**, die den anfänglichen Helligkeitswert enthalten wird. Mit dem Befehl **brightness.measurement()** wird eine Helligkeitsmessung ausgelöst und in **lum0** gespeichert.
- Nun wird der Benutzer durch einen Text(**disp_at()**) aufgefordert, die Helligkeit zu ändern. Der Helligkeitssensor befindet sich an der unten abgebildeten Stelle.

- Für diese Änderung hat er 3s Zeit. Dann läuft das Programm weiter.
- Anlegen einer neuen Variablen **lum1** , in der der neue Helligkeitswert abgespeichert wird.
- Jetzt werden die Messungen miteinander verglichen. Je nachdem leuchtet die LED jetzt für 2s rot oder grün oder sie geht aus

Eine mögliche Erweiterung:

Mit einem neuen Programm soll während einer frei wählbaren Anzahl von Minuten im Abstand von 1 Minute die Helligkeit gemessen und in einer Liste abgespeichert werden.

Hinweis 1: Der Helligkeitssensor ist nicht in Lux geeicht. Das spielt in diesem Fall aber auch keine Rolle, da nur die Veränderung der Helligkeit erfasst werden soll, nicht der tatsächliche Messwert.
Hinweis 2: Will man die Veränderung der Helligkeit z.B. beim Sonnenuntergang messen, so darf während der Messung die Position des Sensors nicht verändert werden !

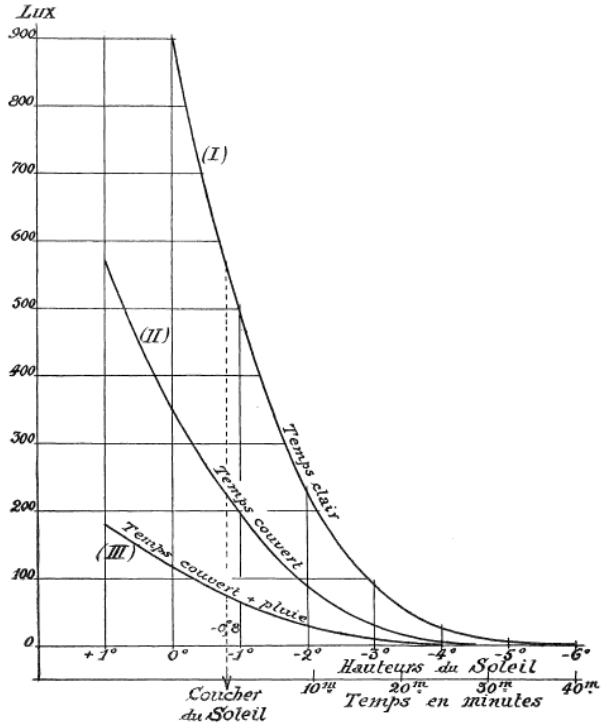


FIG. 2. — Décroissement de la lumière naturelle pendant le crépuscule civil.

Die Funktion **bri(n)** ermittelt während der Messzeit von **n** Minuten die Messwerte für die Helligkeit und speichert sie in der Liste **h[ ]**. Zugleich wird die Liste **t[ ]** der Messzeiten erzeugt.
Anschließend wird die Messung grafisch dargestellt und die Messwerte werden in die Listen **tt** und **hh** des Taschenrechners exportiert.
Darunter ein mögliches Bild.