



Unité 6 : Utiliser la librairie ti_hub & ti_rover

Compétence 1 : Les capteurs intégrés au hub

Dans cette première leçon de l'unité 6, vous allez découvrir comment utiliser la librairie ti_hub afin de commander les dispositifs intégrés au hub TI-Innovator™.

Objectifs :

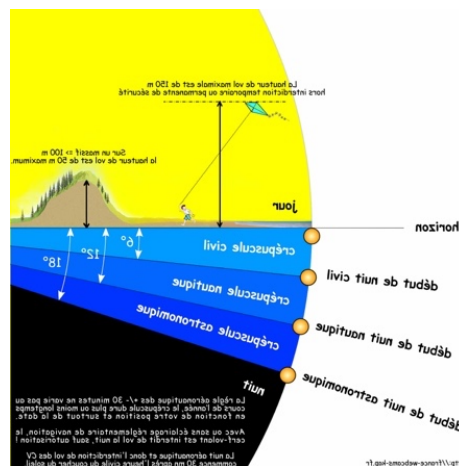
- Découvrir le module **ti-hub**.
- Écrire un script intégrant la librairie ti-hub pour les dispositifs intégrés.

Vous allez, dans cette leçon, utiliser la librairie **ti_Innovator** afin de noter visuellement un changement de luminosité pour, par la suite, simuler un interrupteur crépusculaire, ou bien enregistrer une série de mesures lors du lever du soleil ou du crépuscule.

Vous commencerez par ailleurs à envisager comment associer cette librairie à celle que vous connaissez déjà (**ti_plotlib & ti_system**) afin de créer un projet scientifique complet.

Le script que vous allez écrire correspond à l'algorithme simple suivant :

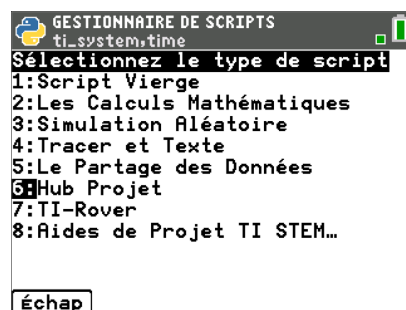
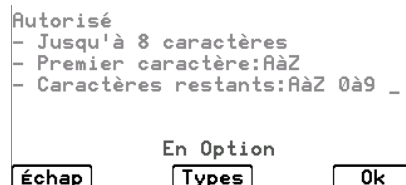
```
Mesurer l'intensité lumineuse ambiante :
Lum0 ← mesure ± (tolérance ?)
Modifier l'intensité lumineuse (lampe ; cache
devant le capteur)
Lum1 ← mesure
Si Lum1 > Lum 0 :
    | Alors allumer la DEL RVB en rouge (2s)
Sinon Si Lum0 < Lum1 :
    | Alors allumer la DEL RVB en vert (2s)
Sinon : Ne rien faire
```



Commencer un nouveau script et le nommer U6SB1

Ce script doit intégrer la librairie **ti_hub**. Pour cela, vous avez plusieurs possibilités.

- Appuyer sur la touche **F3** [Types] puis choisir le menu 6 : Hub Projet. Valider ensuite en appuyant sur **entrer** deux fois ou une seule fois puis OK.
- Vous pouvez également partir d'un script vierge, puis incorporer manuellement les librairies dont vous avez besoin.
- Enfin, vous pouvez aussi utiliser les librairies et variables qui ont été écrites lors des projets TI STEM. Celles-ci seront immédiatement recopiées comme « Objet du projet ». Cette dernière option n'est intéressante que pour créer un script analogue à ceux ayant été définis lors du projet STEM. <https://education.ti.com/en/resources/stem>





10 Minutes de Code

TI-83 PREMIUM CE & TI - PYTHON

En choisissant le type **6 : Hub Projet**, Vous devriez obtenir l'écran ci-contre.

Nous allons utiliser le capteur de luminosité intégré au **TI-Innovator**, ainsi que la diode RVB. Afin que le script soit en mesure de les gérer, intégrons les bibliothèques correspondantes. Pour cela choisir dans le menu **Modul** puis **6 : ti_hub...** et enfin dans le sous-menu **1 : Dispositifs intégrés du Hub**.

Conseil à l'enseignant : Les deux autres possibilités concerneront les capteurs et actionneurs que l'on connectera directement sur les ports d'entrée sortie IN... et OUT... du Hub ou éventuellement sur les ports BBx.

UNITE 6 : COMPETENCE 1

NOTES DU PROFESSEUR

```
ÉDITEUR : U6SB1
LIGNE DU SCRIPT 0004
# Hub Projet
from ti_system import *
from time import *
```

Fns... a A # Outils Exéc Script

```
ÉDITEUR : U6SB1
ti_hub module
Import Commandes Ports Avancé
1:Dispositifs intégrés du Hub...
2:Dispositifs d'entrée...
3:Dispositifs de sortie...
```

Échap Modul

```
ÉDITEUR : U6SB1
LIGNE DU SCRIPT 0006
# Hub Projet
from ti_system import *
from time import *
import color
import brightness
```

Fns... a A # Outils Exéc Script

```
ÉDITEUR : U6SB1
Fonc Ctl Ops List Type E/S Modul
1:math...
2:random...
3:time...
4:ti_system...
5:ti_plotlib...
6:ti_hub...
7:ti_rover...
8:Color...
9:Brightness...
«Hub Output»
«Hub Input»
```

Échap Aide





- Afin que le programme affiche des informations sur un écran « propre », nettoyer celui-ci, à l'aide l'instruction **disp_clr()** se trouvant dans la librairie **ti_system**.
- Créer une variable **lum0**, à laquelle on affecte une mesure de la luminosité. L'instruction **brightns.measurement()** se trouve dans le module **9 : Brightness**, chargé dès lors que le module correspondant est importé.
- Afficher un message invitant l'utilisateur à modifier l'intensité lumineuse dans le voisinage du capteur intégré **disp_at()**.

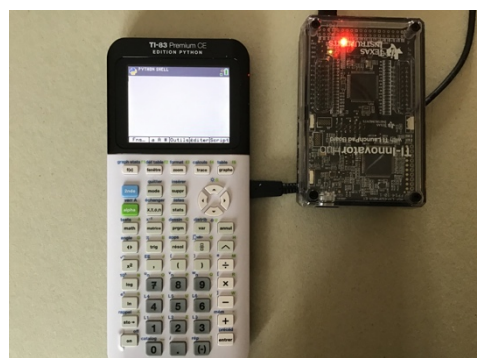


- Mettre le programme en attente **disp_wait()**, le temps d'effectuer cette modification.
- Créer une variable **lum1** à laquelle, on affecte la nouvelle mesure.
- Les mesures sont ensuite comparées. Selon le résultat de l'instruction conditionnelle, la DEL RVB s'allumera en vert ou rouge pendant un délai de 2 secondes.

```

ÉDITEUR : U6SB1
LIGNE DU SCRIPT 0002
# Hub Projet
from ti_system import *
from time import *
import color
import brightns
#mesure
disp_clr()
lum0=brightns.measurement()
disp_at(5,"modifier la luminosité", "left")
disp_wait()
lum1=brightns.measurement()
#comparaison
if lum0<lum1:
    color.rgb(255,0,0)
    sleep(2)
    color.off()
elif lum0>lum1:
    color.rgb(0,255,0)
    sleep(2)
    color.off()
if lum0<lum1:
    color.rgb(255,0,0)
    sleep(2)
    color.off()
elif lum0>lum1:
    color.rgb(0,255,0)
    sleep(2)
    color.off()
else:
    color.off()

```





10 Minutes de Code

TI-83 PREMIUM CE & TI - PYTHON

Prolongement de l'activité :

Modifier le script précédent ou en créer un autre U6SB11

Ce nouveau script doit sur 40 minutes enregistrer les mesures de la luminosité.

Conseil à l'enseignant : Attention le capteur brightness du TI-Innovator n'est pas étalonné en Lux, mais cela n'a pas d'importance dans la mesure où l'on ne s'intéresse qu'aux variations de la luminosité et non à leur mesure en Lux.

Les mesures sont sauvegardées dans une liste **r[]**.

Celles correspondant à la valeur du temps dans une liste **t[]**.

Le script proposé ci-dessous propose également la représentation graphique des mesures afin de le comparer à la représentation ci-contre.

Il est également proposé un export vers les listes de la calculatrice,

UNITE 6 : COMPETENCE 1

NOTES DU PROFESSEUR

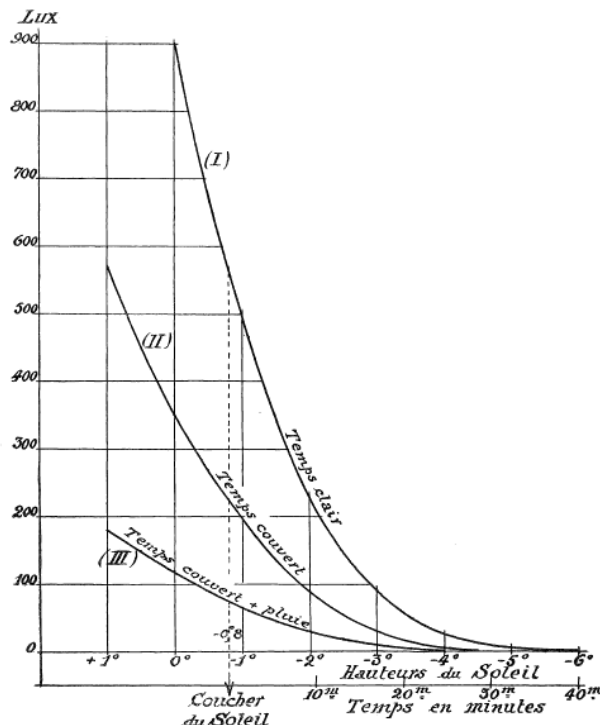


FIG. 2. — Décroissement de la lumière naturelle pendant le crépuscule civil.

a) Acquisition des données

```
ÉDITEUR : U6SB11
LIGNE DU SCRIPT 0001
# Hub Projet_
from ti_system import *
from time import *
import ti_plotlib as plt
import brightness
disp_clr()
def bri(n):
    r=[]
    t=[]
    for i in range(n):
        r.append(brightness.measuremen
            t())
        t.append(i)
        sleep(60)
    return t,r
#représentation graphique
Fns... a A # Outils Exéc Script
```

b) Représentation graphique

```
ÉDITEUR : U6SB11
LIGNE DU SCRIPT 0025
#représentation graphique
def graphe(t,r):
    plt.cls()
    plt.auto_window(t,r)
    plt.labels("t(min)", "r")
    plt.title("Crépuscule")
    plt.color(255,0,255)
    plt.scatter(t,r,"+")
    store_list("1",t)
    store_list("2",r)
    plt.show_plot()
Fns... a A # Outils Exéc Script
```

La fonction **bri(n)** réalise l'acquisition de données toutes les minutes pendant **n** minutes et renvoie les liste **t** et **r**

La fonction **graphe(t,r)** réalise la représentation graphique des données **t[]** et **r[]**, puis les exporte vers les listes L1 et L2 de la calculatrice.

Ce document est mis à disposition sous licence Creative Commons

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

