

Fiche méthode

Référentiel, compétences

- **Capacité** : prendre en main les dispositifs intégrés du microcontrôleur TI-Innovator™ Hub (fig. 1), à l'aide du langage de programmation Python.
- Utiliser un dispositif avec microcontrôleur et capteur.
- Passer de l'algorithme au programme.



Fig. 1

Commentaires de l'auteur

- Une carte à microcontrôleur dispose d'entrées et de sorties programmables à l'aide d'un langage de programmation. Le Hub est une carte programmable en Python qui dispose de 4 dispositifs intégrés : un capteur de luminosité et trois actionneurs (DEL RGB, DEL rouge et haut-parleur).
- Nous allons découvrir dans cette fiche-méthode deux sketches destinés à faciliter la prise en main de la calculatrice TI-83 Premium CE Edition Python et du microcontrôleur TI-Innovator™ Hub.
- Le premier sketch propose de jouer un son d'autant plus aigu que la luminosité ambiante est élevée avec :
 - ✓ En entrée : le capteur de luminosité intégré au Hub.
 - ✓ En sortie : un actionneur, le haut-parleur intégré au Hub.
- Le deuxième sketch propose de simuler le contrôle de la luminosité d'un écran de smartphone. L'écran sera d'autant plus lumineux que la luminosité ambiante est élevée :
 - ✓ En entrée : le capteur de luminosité intégré au Hub.
 - ✓ En sortie : un actionneur, la diode RGB intégrée au Hub.

Matériel

- TI-83 Premium CE Edition Python.
- TI-Innovator™ Hub et le câble USB de liaison.

Prérequis

- Savoir utiliser les dispositifs d'entrée et de sortie intégrés au Hub. Consulter si besoin la fiche méthode intitulée « Découverte des dispositifs intégrés du TI-Innovator™ Hub ».

PREMIER SKETCH : Jouer un son d'autant plus aigu que la luminosité ambiante est élevée

- **OBJECTIF** : Grâce à la programmation, nous allons déclencher une action (modifier la fréquence d'un son) en fonction de la mesure de la luminosité. La note jouée sera dans une gamme comprise entre un La_3 et un La_4 :

Note jouée	% de luminosité mesuré
La_3 (220 Hz)	0 % de luminosité
La_4 (440 Hz)	100 % de luminosité

- **ALGORITHME**

Tant que vrai :

$lum \leftarrow$ Mesure de la luminosité ambiante

$f \leftarrow$ calcul de la fréquence du son à émettre en fonction de la valeur de lum

Émettre le son de fréquence f

Fin de tant que.

- **FONCTIONS PYTHON**

- 1) Créer un SCRIPT nommé SKETCH1.
- 2) L'éditer et écrire une fonction Python **lumi**, qui renvoie la valeur de la luminosité ambiante, mesurée avec le capteur de luminosité intégré au Hub.
- 3) Écrire une fonction Python **note**, qui joue le son de fréquence f pendant 0.2 secondes.
- 4) On souhaite que la fréquence du son joué soit une fonction affine de la luminosité. Trouver l'expression de la fréquence f en fonction de la luminosité lum mesurée.
- 5) À l'aide des fonctions précédentes, écrire une fonction Python **joue**, qui émet un son dont la fréquence est fonction de la luminosité ambiante mesurée.

- **SOLUTION PROPOSÉE**

$$f = 2,2 \times lumi() + 220$$

Télécharger le programme **SKETCH1** à l'adresse :

<https://education.ti.com/fr/physique-chimie>

```
ÉDITEUR : SKETCH1
LIGNE DU SCRIPT 0001
import sound
import brightns
def lumi():
    lum=brightns.measurement()
    return lum
def note(f):
    sound.tone(f,0.2)
def joue():
    while True:
        f=2.2*lumi()+220
        note(f)
```



DEUXIÈME SKETCH : Rendre l'écran d'un smartphone d'autant plus lumineux que la luminosité ambiante est élevée

- **OBJECTIF** : Grâce à la programmation, nous allons déclencher une action (modifier l'éclat d'une DEL qui modélisera l'écran lumineux) en fonction de la mesure d'une luminosité.
 - ✓ On choisit ici de travailler avec la DEL RGB, dont la couleur est déterminée par le triplet **(r, g, b)** avec **r**, **g** et **b** trois entiers compris entre 0 et 255. En faisant varier les valeurs du triplet entre **(0, 0, 0)** et **(255, 255, 255)**, il est possible de moduler l'éclat de la DEL.
- **ALGORITHME**

Tant que vrai :

$lum \leftarrow$ Mesure de la luminosité ambiante

$val \leftarrow$ calcul d'une valeur en fonction de la valeur de lum

Affecter la valeur val à la diode RGB pour qu'elle s'éclaire plus ou moins.

Fin tant que.

- **FONCTIONS PYTHON**

- 1) Créer un SCRIPT nommé SKETCH2.
- 2) L'éditer et écrire une fonction Python **couleur**, munie de l'argument **val**, qui éclaire la DEL RGB avec le triplet **(val, val, val)**.
- 3) Écrire une fonction Python **lumi**, qui renvoie la valeur de la luminosité ambiante, mesurée avec le capteur de luminosité intégré au Hub.
- 4) On souhaite que la valeur **val** soit une fonction linéaire de la luminosité renvoyée par la fonction **lumi**. Trouver l'expression de **val** en fonction de la luminosité mesurée.
- 5) À l'aide des fonctions précédentes, écrire une fonction Python **ecran**, qui éclaire la DEL en fonction de la luminosité ambiante mesurée.

- **SOLUTION PROPOSÉE**

$val = 2,55 \times lumi()$. Ainsi si **lumi** renvoie 0%, **val** vaudra 0 et si **lumi** renvoie 100%, **val** vaudra 255.

Remarque : en réalité, un écran de smartphone n'est jamais complètement éteint ! On peut alors définir une luminosité minimale « seuil » égale à 5 grâce à une fonction **max**. Pour cela remplacer la dernière ligne du script par :

```
.....couleur(max(val,5))
```

Télécharger le programme **SKETCH2** à l'adresse :

<https://education.ti.com/fr/physique-chimie>.

```
EDITEUR : SKETCH2
LIGNE DU SCRIPT 0001
import brightns
import color
def couleur(val):
    color.rgb(val,val,val)
def lumi():
    lum=brightns.measurement()
    return lum
def ecran():
    while True:
        val=2.55*lumi()
        couleur(val)
```

