Thème : Découverte des dispositifs intégrés du Ti-Innovator™ Hub : **DEL RGB, DEL rouge, Haut-parleur, Capteur de luminosité.**

TI-83 Premium CE

Fiche méthode

Stéphane Bonnaud

Référentiel, compétences

- **Capacité :** prendre en main les dispositifs intégrés du microcontrôleur TI-Innovator™ Hub, à l'aide du langage de programmation Python.

Commentaires de l'auteur

Fig. 1

- Le TI-Innovator™ Hub de Texas Instruments (fig. 1) est une carte à microcontrôleur. Il peut être un outil privilégié pour aborder sereinement avec les élèves les capacités expérimentales numériques de physique-chimie.
- Le Hub est muni de 3 entrées (IN1, IN2 et IN3, fig.2, à gauche) et de 3 sorties (OUT1, OUT2, OUT3, fig.2, à droite), toutes programmables avec Python. Sur chaque port (IN ou OUT), il est possible de connecter un capteur (IN) ou un actionneur (OUT).
- Le Hub est également muni de **4 dispositifs déjà intégrés**, que nous allons détailler dans cette fiche méthode : un capteur de luminosité et les trois actionneurs (DEL RGB, DEL rouge et haut-parleur).

  Le Hub communique avec la calculatrice TI-83 premium CE grâce à un câble USB. Le langage de programmation utilisé dans cette fiche est Python.

Fig. 2

Matériel

- Calculatrice TI-83 premium CE Edition Python.
- TI-Innovator™ HUB et le câble USB de liaison.

Prérequis

- S'assurer que le Hub soit bien à jour. Pour cela suivre la procédure suivante :
  1. Télécharger le logiciel de mise à jour du TI-Innovator™ Hub
  2. L'installer sur votre ordinateur. Lancer le logiciel.
  3. À ce stade, si le Hub n'est pas relié à l'ordinateur le message ci-contre s'affiche (fig.3).
  4. Relier le HUB à l'ordinateur avec le câble USB approprié en utilisant l'entrée **POWER (PWR) du Hub.** Un message de confirmation apparaît (fig.4). Attention, il est recommandé d'utiliser le câble livré avec le Hub.
  5. Un menu propose alors de télécharger puis d'installer le fichier *sketch* le plus récent pour mettre à jour le Hub (1.4.0.28.Hub en avril 2020).

Fig. 3

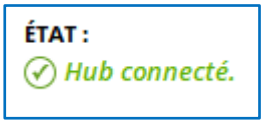
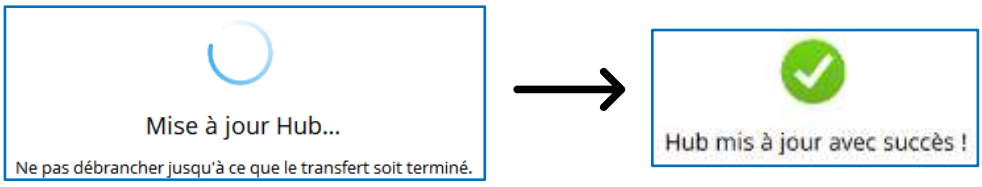


Fig. 4



Avertissement

Nous allons présenter en détail comment utiliser les 4 dispositifs intégrés au Hub.

Étape 1 : Prise en main *détaillée* de la diode RGB (actionneur).

Étape 2 : Prise en main de la diode rouge (actionneur), moins détaillée qu'à l'étape 1, car il existe des similitudes avec l'étape 1.

Étape 3 : Prise en main du haut-parleur (actionneur).

Étape 4 : Prise en main du capteur de luminosité (capteur).

Le SCRIPT **HUB1** rassemblera les **4** fonctions nécessaires à la découverte de ces **4** dispositifs intégrés.

Étape 1 : Découverte de la diode RGB : le module Color

- Une diode RGB programmable est intégrée au Hub (fig. 5). Elle permet d'obtenir une couleur au choix parmi 255^3 =16,7 millions de couleur. Chaque couleur est caractérisée par un triplet de valeurs (R, V, B).

  On souhaite faire briller la diode RGB selon la couleur de notre choix, par exemple en magenta, caractérisé par le triplet $(R, V, B) = (255,0,255)$. Cette diode est repérable sur le Hub par la mention *COLOR* (fig. 5).

Fig. 5 : La diode RGB a été ici recouverte d'un papier calque pour mieux diffuser la lumière.

  - Préparer un nouveau script Python nommé HUB1.
  - Dans le SCRIPT HUB1 :

1. Dans Modul, sélectionner le menu 6 : ti_Hub… (fig. 6)
2. Sélectionner 1 : Dispositifs intégrés du Hub… (fig. 7)
3. Sélectionner 1 : Color (fig. 8). Le module **`Color`** est ainsi importé au début du script (**`import color`**), voir fig. 11 ligne 2.
4. Dès lors, un nouveau module nommé **`Color`** apparaît dans la liste des modules disponibles (fig. 9). Toutes les fonctions du module **`Color`** sont alors accessibles dans le menu Modul avec 8 : Color…

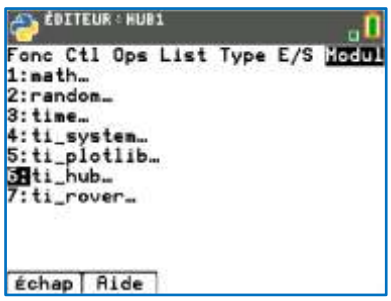
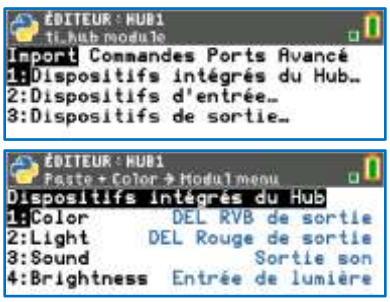


Fig. 6



< Fig. 7

< Fig. 8

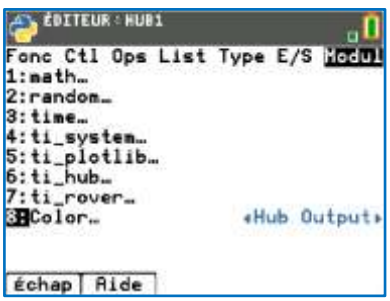


Fig. 9

5. Le module **Color** possède 3 fonctions : **rgb**, **blink** et **off**, (fig. 10).

Fig. 10

- La fonction **rgb** allume la DEL RGB avec la couleur correspondant au triplet **(r,g,b)**. Chaque variable **r**, **g** ou **b** a une valeur entière comprise entre 0 et 255.
- La fonction **blink** fait clignoter la DEL avec la fréquence **freq**, exprimée en Hz, dans l'intervalle [0.1 Hz ; 20 Hz] pendant une durée égale à **time**, exprimée en secondes.
- La fonction **off** éteint la DEL RGB. Attention, cette fonction éteint la DEL immédiatement, sans attendre par exemple qu'elle ait fini de clignoter. Si besoin, un appel à la méthode **sleep(secondes)** du module **time** permettrait de temporiser en mettant le programme sur pause quelques instants.

- Écrire une fonction Python **couleur**, munie des 5 arguments **r,g,b,f,dt**, qui fasse clignoter la DEL avec la couleur **(r,g,b)** souhaitée, à la fréquence **f,** pendant la durée **dt** (fig. 11).

Fig. 11

Étape 2 : Découverte de la diode Rouge : le module Light

- Sur le même modèle que le module **Color**, le module **Light** permet de piloter la DEL rouge du Hub (fig. 12). Cette DEL est toujours de couleur rouge.

Fig. 12 : La diode rouge a été ici recouverte d'un papier calque pour mieux diffuser la lumière.

- Dans le SCRIPT HUB1 :
  1. Importer le module **Light** : **import light** s'affiche au début du script (fig. 11, ligne 4).
  2. Le module **Light** possède 3 fonctions : **on, off** et **blink**, dont les rôles sont explicites (fig. 13).

  NB : **freq** appartient à l'intervalle [0.1 Hz ; 20 Hz].

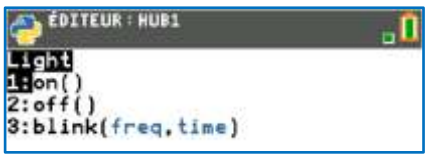


Fig. 13

- Écrire une fonction Python **rouge**, munie de 2 arguments **f** et **dt**, qui fasse clignoter la DEL rouge à la fréquence **f,** pendant la durée **dt** (fig. 14).

Fig. 14

```
def rouge(f,dt):
    light.blink(f,dt)
```

Étape 3 : Découverte du haut-parleur : le module Sound

- Sur le même modèle que le module **Color**, le module **Sound** permet de produire un son à l'aide du haut-parleur intégré au Hub (fig. 15).

Fig. 15

- Dans le SCRIPT HUB1 :
  1. Importer le module **Sound** : `import sound` s'affiche au début du script (fig. 11, ligne 3).
  2. Le module **Sound** possède 2 fonctions : **tone** et **note** dont les rôles sont explicites (fig. 16).

NB : L'argument `time,` de type *float*, est optionnel, compris entre 0.1 et 100.0, exprimé en secondes.

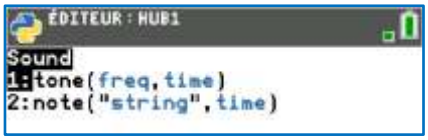


Fig. 16

- ❖ La fonction **tone** produit un son pur de fréquence `freq`, exprimée en Hz, dans la gamme [1Hz ; 8000 Hz] pendant la durée `time`, exprimée en secondes.
- ❖ La fonction **note** produit une note de musique à partir de son nom, codé selon la norme anglo-saxonne, qui utilise des notes de l'alphabet. Chaque note est suivie d'un chiffre correspondant à son octave. Ainsi la note « La 440 », qui est un La_3, s'écrit en notation anglo-saxonne "**A4**".

Notation Française	Notation Anglo-saxonne
Do	C
Ré	D
Mi	E
Fa	F
Sol	G
La	A
Si	B

- Écrire une fonction Python **joue1,** munie des arguments **f et dt,** qui joue une note de fréquence **f,** pendant la durée `dt` (fig. 17).

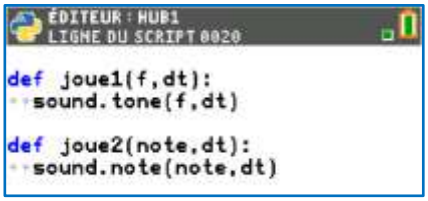


Fig. 17

- Écrire une fonction Python **joue2**, qui joue une note appelée `note` en notation anglo-saxonne (ne pas oublier les guillemets lors de l'appel de la fonction) pendant la durée `dt` (fig. 17). L'argument `note` est de type chaîne de caractère.

Étape 4 : Découverte du capteur de luminosité : le module Brightness

Fig. 18

- Sur le même modèle que le module **Color**, le module **Brightness** permet de mesurer une luminosité à l'aide du capteur de luminosité intégré (fig. 18).

- Dans le SCRIPT HUB1 :
  1. Importer le module **Brightness** : `import brightns` s'affiche au début du script (fig. 11, ligne 5).
  2. Le module **Brightness** possède 2 fonctions : **measurement** et **range**, fig. 19.

Fig. 19

  - ❖ La méthode **measurement** déclenche la mesure de la luminosité. Son unité n'est pas connue. La valeur renvoyée est un nombre flottant compris entre 0% (très sombre) et 100% (très lumineux).
  - ❖ La méthode **range** permet de définir si nécessaire une plage de mesure, comprise entre `min` et `max`.

- Écrire une fonction Python **lumi** qui renvoie la valeur de la luminosité mesurée par le capteur, comprise entre 0 et 1000. (fig. 20).

```
def lumi():
  brightns.range(0,1000)
  mes=brightns.measurement()
  return mes
```

Fns… | a A # | Outils | Exéc | Script

Fig. 20

Télécharger le script **HUB1** à l'adresse : https://education.ti.com/fr/physique-chimie.