

## TP prise en main du Hub TI

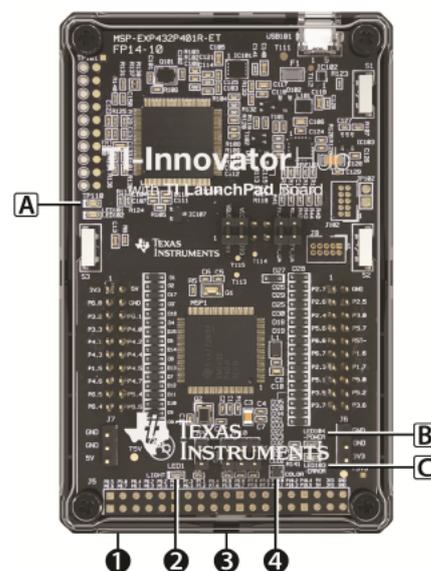
Nous allons voir ici comment prendre en main le Hub Texas Instruments, puis dans un autre TP, nous verrons la prise en main du Rover TI.

Le Hub est composé de :

- 1 : Un capteur de luminosité sur la partie inférieure du Hub qui peut être lu comme « BRIGHTNESS » dans les chaînes de commande ;
- 2 : Une DEL rouge qui est adressable en tant que « LIGHT » dans les chaînes de commande ;
- 3 : Un haut-parleur (à l'arrière du Hub) qui est adressable en tant que « SOUND » dans les chaînes de commande ;
- 4 : Une DEL rouge-vert-bleu qui est adressable en tant que « COLOR » dans les chaînes de commande.

Les éléments suivants sont également visibles en face du hub :

- A DEL d'alimentation auxiliaire verte ;
- B DEL d'alimentation verte ;
- C DEL d'erreur rouge.



Il est équipé aussi d'un certain nombre d'entrées et sorties :

- Trois ports IN permettant de recueillir des données ou l'état des modules d'entrée ;
- Trois ports OUT permettant de contrôler les modules de sortie (OUT 1 et 2 en 3V et OUT 3 en 5V).
- Un port I2C pour gérer les périphériques qui utilisent le protocole de communication I2C.
- Un port mini-USB B pour se connecter à un ordinateur ou à une calculatrice graphique compatible pour les données et l'alimentation.
- Un connecteur de la platine d'essais comprenant 20 broches étiquetées pour une communication avec les composants connectés.

Pour la connexion à une calculatrice TI 83 Python Edition CE, utilisé le câble data B, reliez le connecteur B au port B du Hub et le connecteur A à la calculatrice.

La Hub TI réagit aux commandes de programmation en TI Basic ou Python. Nous utiliserons Python pour l'ensemble des TP.

Quelques informations avant de commencer pour voir comment rentrer un programme à l'aide de la calculatrice :

1. Sur l'écran Accueil, appuyez sur **prgm**, puis Python APP (2). Appuyez sur **enter**.
2. Choisissez **Nouv** (en appuyant sur **zoom**).
3. Rentrer le nom de votre programme, par exemple "SOUNDTST", appuyez sur **Types**. Sélectionnez Projets STEM Hub (6) (pour les projets avec le Rover vous choisirez Rover (7)). Appuyer sur **Ok**.

L'éditeur de programmes s'ouvre en affichant un modèle du code de votre programme.

Vous remarquez que 2 lignes imports de modules sont déjà là.

L'image montre l'écran d'un éditeur de code sur une calculatrice TI-83 Python Edition CE. Le titre de la fenêtre est 'ÉDITEUR : SOUNDTST'. Le contenu du script est :

```
LIGNE DU SCRIPT 0004
# Projets STEM Hub
from ti_system import *
from time import *
```

Le menu de la calculatrice est visible en bas de l'écran, avec les options 'Fns...', 'a A #', 'Outils', 'Exéc' et 'Script'.

3. Pour rentrer les lignes de codes, nous utiliseront beaucoup les onglets en bas de l'écran qui permettent d'avoir accès à des parties de scripts déjà tapées.

Au début il ne faudra pas hésiter à chercher un peu...

Je vous propose de taper un court programme qui fera du bruit.

Dans votre programme nommé "SOUNDTST", tapez :

Pour la ligne 3 :

**Fns...** puis **Modul** puis 6 : ti\_hub... puis 1 : Dispositifs intégrés du Hub puis 3 :Sound et **Import**.

Le fait d'importer le module Sound vous permet d'avoir accès à la bibliothèque de commande de ce module.

Pour la ligne 4 :

**Fns...** puis **Modul** puis 8 : Sound... puis 1 : tone(freq,time)

La fréquence est en Hz (ici 440Hz correspond à la note La) et time en s.



```
ÉDITEUR : SOUNDTST
LIGNE DU SCRIPT 0003
# Projets STEM Hub
from ti_system import *
from time import *
import sound
sound.tone(440,2)

Fns... a A # Outils Exéc Script
```

Pour exécuter votre script, vérifiez que le hub est bien connecté à la calculatrice puis sélectionnez **Exéc**.

Vous venez de voir le fonctionnement de base du Hub.

Nous allons voir quelques exemples de codes à tester :

Remarque : les lignes de codes sont données en partant du principe que les 2 modules ti\_system et time sont déjà importées.

Voici quelques exemples de codes à tester :

Action souhaitée	Code du programme
Allumer le voyant LED rouge intégré ("LIGHT").	<code>Import light</code> <code>Light.on()</code>
Produire une tonalité de 440 Hz sur le haut-parleur intégré ("SOUND") pendant 2 secondes.	<code>Import sound</code> <code>Sound.tone(440,2)</code>
Allumer la Led RGB ("COLOR") avec les paramètres RGB 255, 255,255.	<code>Import color</code> <code>Color.RGB(255,255,255)</code> Les 3 paramètres peuvent être choisis entre 0 et 255 (à vous de tester...)
Lire et afficher la valeur actuelle qui apparaît sur le capteur de niveau de lumière intégré ("BRIGHTNESS"). La plage varie de 0 % à 100 %.	<code>Import brightns</code> <code>a=Brightns.measurement()</code> <code>print(a)</code>

Pour faire clignoter une LED, vous pouvez rentrer ce programme :



```
ÉDITEUR : SOUNDTST
LIGNE DU SCRIPT 0001
# Projets STEM Hub
from ti_system import *
from time import *
import light
for i in range(0,10):
  light.on()
  sleep(1)
  light.off()
  sleep(1)

Fns... a A # Outils Exéc Script
```

Remarques :

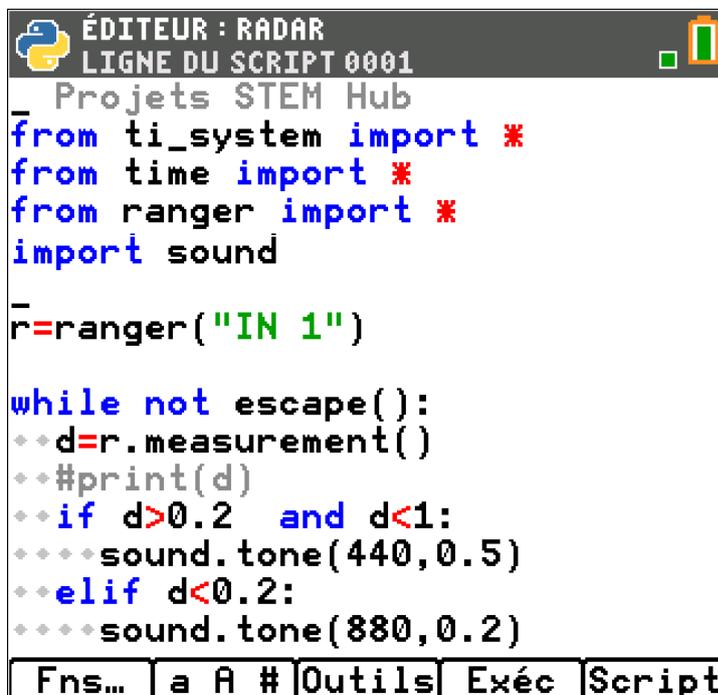
- la commande `sleep()` se trouve dans `Fns...` puis `Modul` puis 4 : `ti_system`.
- Il y a une commande pour le clignotement que vous avez dû croiser : `blink(freq,time) ;`

### Connexion d'un capteur à ultrasons :

Brancher le dans l'entrée IN 1.

Nous allons faire un programme de type radar de recul.

Voici le code :



```

ÉDITEUR : RADAR
LIGNE DU SCRIPT 0001
_ Projets STEM Hub
from ti_system import *
from time import *
from ranger import *
import sound

r=ranger("IN 1")

while not escape():
  *d=r.measurement()
  *#print(d)
  *if d>0.2 and d<1:
  *   *sound.tone(440,0.5)
  *elif d<0.2:
  *   *sound.tone(880,0.2)
  
```

- Remarque: la commande `while not escape()` : se trouve dans `Fns...` puis `Modul` puis 4 : `ti_system`.

Plusieurs remarques pour ce programme :

`while not escape()` : permet de créer un boucle infinie qui se termine lorsque l'on appuie sur ON.

`#print(d)` : Si vous enlevez le `#` qui place cette ligne en commentaire, vous affichez la distance.

Prenez le temps de bien comprendre chaque ligne du programme et de modifier certaines valeurs pour comprendre son fonctionnement.

### Connexion de la LED blanche :

Connectez la Led sur le Port OUT3.

Reproduire le code ci-contre :

Modifier le code pour que la LED clignote 10 avec un intervalle de 2 secondes.

Remarque :

La commande `sleep` se trouve dans `Fns...` puis `Modul` puis 3 : `time...`



```

ÉDITEUR : TESTPROF
LIGNE DU SCRIPT 0010
# Projets STEM Hub
from ti_system import *
from time import *
from led import *
l1=led("OUT 3")
for i in range(5):
  *l1.on()
  *sleep(1)
  *l1.off()
  *sleep(1)
  
```