Thème : informatique embarquée et objets connectés

TI-83 Premium CE
EDITION PYTHON

Classe enseignée : 2nde SNT

Lampe frontale

F. GIROD

Compétences visées

Un des objectifs de l'enseignement de SNT est de développer et de coder des scripts PYTHON afin d'apporter une réponse à une problématique précise. A travers le thème "**informatique embarquée et objets connectés**", nous pouvons notamment travailler les compétences suivantes dans l'activité proposée :

- Coder des scripts simples d'acquisition de données.
- Gérer des entrées/sorties à travers les ports utilisés par le système.
- Écrire et développer des algorithmes pour résoudre une problématique.
- Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs.

Situation déclenchante

Une lampe frontale propose en général plusieurs modes :

- éteinte ;
- lumière faible ;
- lumière plus soutenue ;
- lumière rouge ;
- lumière rouge clignotante.

Comment reproduire un fonctionnement de ce type ?

Ici, une **vidéo** illustrant le projet.

Problématique

Comment reproduire le fonctionnement d'une lampe frontale ?

1. Identifier un capteur possible ;
2. Identifier un actionneur possible ;
3. Élaborer un algorithme modélisant le fonctionnement d'une lampe frontale ;
4. Mettre cet algorithme en application.

Matériel disponible

Les élèves disposent, en plus de leur TI-83 Premium CE :

- d'un TI-Innovator HUB ;
- d'un capteur de distance à ultrasons.

Thème : informatique embarquée et objets connectés

TI-83 Premium CE
EDITION PYTHON

Classe enseignée : 2nde SNT

Fiche méthode

F. GIROD

Déroulement possible du projet

En amont du projet :

Les propositions suivantes permettent de préparer le projet et de le rendre possible sur 2 séances d'1h30. Elles peuvent aussi être préparées par l'enseignant sous forme de fiches et/ou d'un exposé au tableau.

- Un groupe d'élèves peut préparer une fiche ou présenter un rapide exposé illustrant les principaux points pour maîtriser la programmation en PYTHON spécifique au **TI-Innovator HUB**, en s'appuyant sur les **'10 mn de code'** (Unité 6)
- Un groupe d'élèves peut expliquer le **mode de codage RGB** pour les couleurs.
- Un groupe d'élèves peut décrire le **TI-Innovator HUB**.

Différentes évolutions possibles pour le projet :

Les propositions suivantes donnent des pistes pour gérer les différences de vitesse d'exécution des élèves.
On peut aussi considérer ces différentes étapes comme une manière de séquencer le projet pour ceux qui en auraient besoin.

- 1ère étape : on allume une lumière en passant sa main devant le capteur à ultrasons.
- 2ème étape : on allume et on éteint une lumière en passant sa main devant le capteur à ultrasons : création d'un compteur et discrimination selon sa parité.
- 3ème étape : on actionne plusieurs modes (au choix des élèves):
    - On passe sa main devant le capteur : une lumière peu intense s'allume ;
    - On passe à nouveau sa main : une lumière plus intense s'allume ;
    - On passe à nouveau sa main : une lumière rouge s'allume ;
    - On passe à nouveau sa main : la lumière s'éteint ;
    - On passe à nouveau sa main : le cycle recommence.
- 4ème étape : on met en place un mode "clignotement".

A la suite du projet :

Synthétiser ce travail par rapport au cours de SNT :

- le(s) capteur(s) et le(s) actionneur(s) mis en jeu ;
- l'algorithme qui gère les données d'entrée ;
- le langage de programmation utilisé.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

Thème : informatique embarquée et objets connectés

TI-83 Premium CE
EDITION PYTHON

Classe enseignée : 2nde SNT

Fiche méthode

F. GIROD

Proposition de résolutions

Choix du capteur :

- Capteur de distance à ultrasons : en passant sa main à proximité, la distance mesurée passe en dessous d'un seuil qui entraîne le passage d'un mode à l'autre. C'est ce capteur qui a été choisi ici.
- Capteur de luminosité (brightness) : en cachant ce capteur, la luminosité passe en dessous d'un seuil qui entraîne le passage d'un mode à l'autre. On peut l'utiliser si on ne dispose pas de capteur à ultrasons ou si on veut proposer une alternative.

Choix de l'actionneur :

- LED RGB du HUB : elle pourra délivrer une lumière blanche plus ou moins intense ou une lumière colorée.

Etapes de résolution

L'importation de ces bibliothèques est explicitée dans la remarque qui suit cette partie.

On crée les variables **c** et **n** qui seront des compteurs. Elles sont initialisées à 0.

- **c** comptera le nombre de fois où l'on passe près du capteur de distance.
- **n** donnera le reste de la division euclidienne de **c** par 5 (cela s'écrit en python **c%5**) : cela permet d'activer une des cinq modalités de la lampe.

La variable **m** représente l'entrée du ranger (capteur de distance) relié au port d'entrée **IN 1** ; les instructions se trouvent dans le module **ranger** importé précédemment.

```
ÉDITEUR : LAMPEF
LIGNE DU SCRIPT 0001
# Projets STEM Hub
from ti_system import *
from time import *

import color
from ranger import *

c=0
n=0

m=ranger("IN 1")
Fns… | a A # | Outils | Exéc | Script
```

- `while not escape():` crée une boucle 'infinie' qui sera interrompue par l'appui sur la touche **on** ; l'instruction se trouve dans le module *ti_system*.
- `d=m.measurement()` crée la variable **d** qui est la mesure donnée par **m** (le ranger branché à l'entrée 1).
- Une mesure est réalisée chaque demi-seconde.
- Si la distance mesurée par le ranger est inférieure à 0,1 m = 10 cm (en passant la main à proximité), le compteur **c** s'incrémente d'une unité.
- Rappel : `c+=1` est un raccourci pour `c=c+1`
- Selon la valeur de **n** (0/1/2/3/4), on aura diverses situations :
  - lumière éteinte avec **color.rgb(0,0,0)** (cette fonction se trouve dans la bibliothèque *color* importée en préambule) ;
  - lumière blanche peu lumineuse avec **color.rgb(100,100,100)** ;
  - lumière blanche lumineuse avec **color.rgb(255,255,255)** ;
  - lumière rouge avec **color.rgb(255,0,0)** ;
  - lumière rouge clignotante (temps de clignotement de 0,1 s).

```
ÉDITEUR : LAMPEF
LIGNE DU SCRIPT 0025
while not escape():
  print("n=",n)
  d=m.measurement()
  sleep(0.5)
  print("d=",d)
  if d<0.1:
    c+=1
  n=c%5
  if n==0:
    color.rgb(0,0,0)
  elif n==1:
    color.rgb(100,100,100)
  elif n==2:
    color.rgb(255,255,255)
  elif n==3:
    color.rgb(255,0,0)
  else:
    color.rgb(255,0,0)
    sleep(0.1)
    color.rgb(0,0,0)
    sleep(0.1)
Fns… | a A # | Outils | Exéc | Script
```

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus

Thème : informatique embarquée et objets connectés

TI-83 Premium CE
EDITION PYTHON

Classe enseignée : 2nde SNT

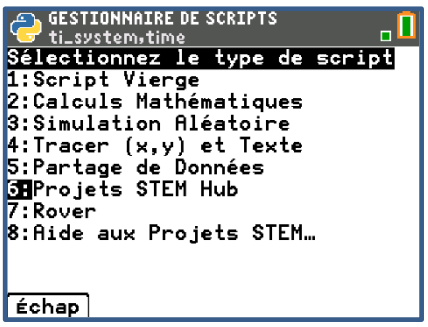
Fiche méthode

F. GIROD

Remarques

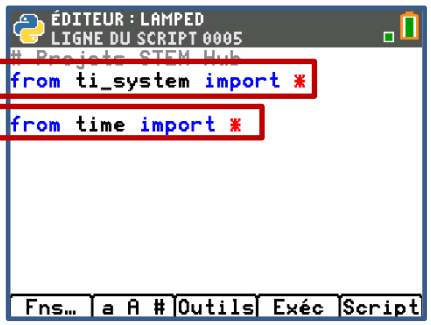
En préambule du code, on trouve l'importation de bibliothèques spécifiques au projet :

- elles peuvent être importées une à une en allant les chercher dans le menu **Modul** ;
- elles peuvent s'implémenter en choisissant le type de programme « **Projets STEM Hub** » (choix 6 des types de programmes proposés) à la création du nouveau programme.

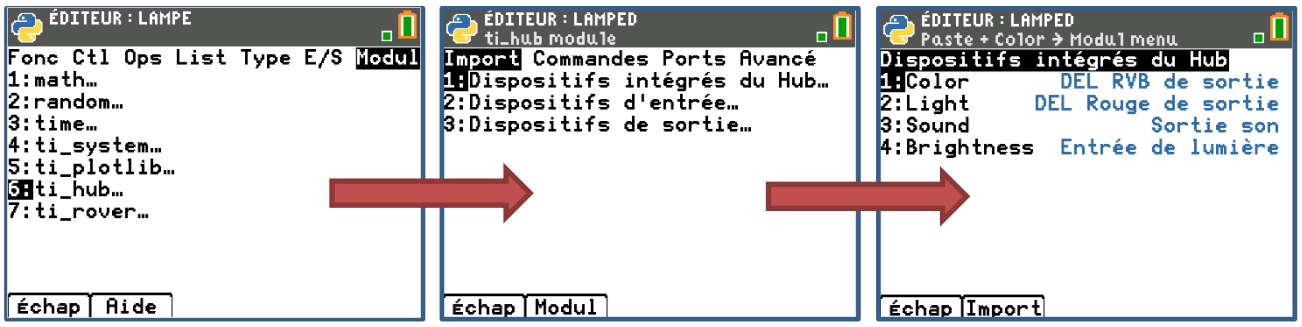


La bibliothèque *ti_system* permet d'importer ou d'exporter des données dans les listes de la calculatrice.
Pour ce projet, elle contient l'instruction **while not escape( ) :** qui permet de lancer une boucle qui ne s'interrompt que lorsqu'on appuie sur la touche **on**.

La bibliothèque *time* permet en particulier d'utiliser la fonction **sleep** qui prend comme argument un temps de pause donné en secondes.



On va ensuite importer une bibliothèque spécifique au projet, à savoir la bibliothèque *color* en suivant les instructions suivantes :



Au fur et à mesure que les bibliothèques sont importées, elles s'ajoutent aux bibliothèques déjà présentes (*math*, *random*, …etc.) et permettent d'accéder aux instructions qu'elles contiennent.

On procède de même pour importer la bibliothèque *Ranger* qui s'obtiendra en choisissant '**dispositifs d'entrée**' à l'étape 2 de la procédure précédente.

Pour profiter de tutoriels vidéos, Flasher le QRCode ou cliquer dessus