Thème: SNT (Informatique embarquée)

Niveau: 2nde SNT

TI-83 Premium CE
EDITION PYTHON

python

Radar de recul

F. GIROD

Compétences visées

Un des objectifs de l'enseignement de SNT est d'écrire et développer des programmes pour répondre à des problèmes concrets. A travers le thème "**informatique embarquée et objets connectés**", nous pouvons notamment travailler les compétences suivantes dans l'activité proposée :

- Écrire des programmes simples d'acquisition de données.
- Gérer des entrées/sorties à travers les ports utilisés par le système.
- Écrire et développer des programmes pour répondre à des problèmes.
- Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs.

Situation déclenchante

Comment signaler au conducteur d'une voiture que son véhicule approche d'un obstacle lorsqu'il recule ?

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

Problématique

Comment reproduire le fonctionnement d'un radar de recul ?

1. Identifier un capteur possible ;
2. Identifier un actionneur ou des actionneurs possibles ;
3. Mettre en place un algorithme modélisant le fonctionnement d'un radar de recul ;
4. Le mettre 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: SNT (Informatique embarquée)

Niveau: 2nde SNT

TI-83 Premium CE
EDITION PYTHON

python™

F. GIROD

Fiche méthode

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 fiche 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 émet un bip (à une certaine fréquence) lorsque la distance détectée par le capteur est inférieure à une certaine valeur.
- 2ème étape : on émet différents types de 'bip' selon la distance à l'obstacle.
- 3ème étape : on émet un bip dont la fréquence dépend continument de la distance à l'obstacle.
- 4ème étape : on double ce bip par un clignotement lumineux.

A la suite du projet :

Synthétiser ce travail par rapport au cours de SNT :

le/les capteurs et le/les actionneurs mis en jeu / l'algorithme qui gère les données d'entrée / le langage de programmation utilisé.

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 l'émission d'un bip ou/et d'un clignotement lumineux.

Choix de / des actionneurs :

- LED RGB du HUB : elle pourra délivrer une lumière de la couleur de son choix.
- LED du HUB : elle émet une lumière rouge : il est choisi dans ce projet.
- Haut-parleur du HUB : il émet un son dont on définit la fréquence d'émission et la durée : il est choisi dans ce projet.

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

Thème: SNT (Informatique embarquée)

Niveau: 2nde SNT

TI-83 Premium CE
EDITION PYTHON

python

F. GIROD

Fiche méthode

Etapes de résolution

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

On crée la variable **r** qui représente l'entrée du ranger (capteur de distance) relié au port d'entrée **IN 1** ; les instructions se trouve dans le module **ranger** importé précédemment.

```
ÉDITEUR : RADAREC
LIGNE DU SCRIPT 0001
# Projets STEM Hub
from ti_system import *
from time import *
from ranger import *
import light
import sound

r=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 **r** (le ranger branché à l'entrée 1)
- On propose un affichage de **d** (pendant la réalisation du programme, pour s'assurer que tout fonctionne correctement), affichage que l'on peut commenter par la suite par la suite par # (qui s'obtient par exemple par la touche **2nde** suivie de la touche **3**)
- selon la valeur de **d** , on aura plusieurs cas de figure :
  - distance comprise entre 20 cm et 1 m : on fait clignoter la lumière rouge du hub (toutes les 0,5 seconde) et on émet un bip (son de 440 Hz toutes les 0,5 seconde)
  - distance inférieure à 20 cm : on fait clignoter la lumière rouge du hub (toutes les 0,2 seconde) et on émet un bip plus aigu (son de 880 Hz toutes les 0,5 seconde)
  - distance supérieure à 1 m : il ne se passe rien ; pas besoin d'écrire d'instructions.
- La fonction **sleep** se trouve dans la bibliothèque **time**.
- La fonction **sound.tone** se trouve dans la bibliothèque **sound**.
- Les fonctions **light.on** et **light.off** se trouvent dans la bibliothèque light.

```
ÉDITEUR : RADAREC
LIGNE DU SCRIPT 0020
while not escape():
  d=r.measurement()
  #print('d=',d)
  if d>0.2 and d<1:
    light.on()
    sleep(0.5)
    light.off()
    sound.tone(440,0.5)
  elif d<0.2:
    light.on()
    sleep(0.2)
    light.off()
    sound.tone(880,0.2)
```

Fns… | a A # | Outils | Exéc | Script

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

Thème: SNT (Informatique embarquée)

Niveau: 2nde SNT

TI-83 Premium CE
EDITION PYTHON

python

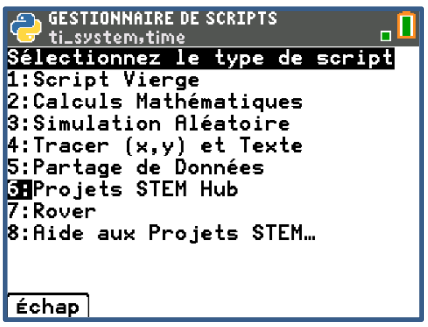
F. GIROD

Fiche méthode

Remarques

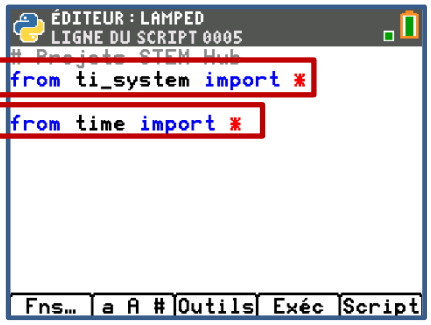
En préambule du code, importation de bibliothèques spécifiques au projet :

- 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.

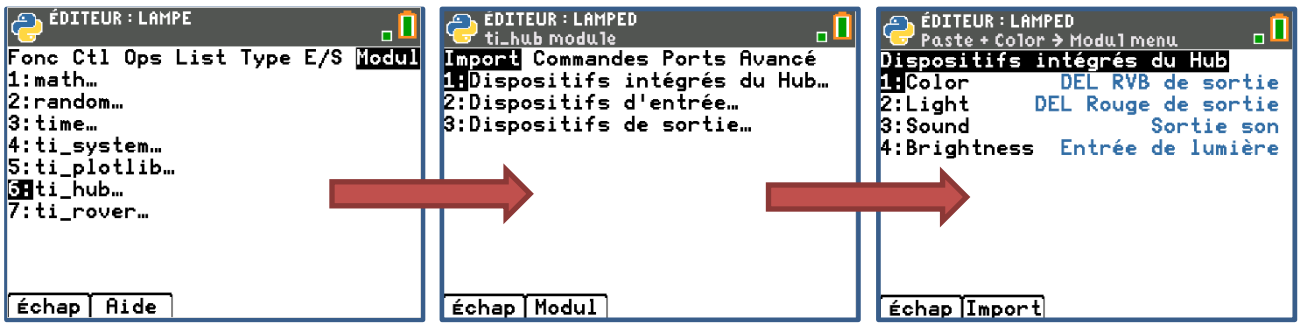


La bibliothèque « **ti_system** » permet d'importer ou d'exporter des données dans les listes de la calculatrice.
Pour ce projet, elle contient l'instruction **while not escape() :** qui permet de lancer une boucle qui ne s'interrompt que si on appuie sur la touche **on**.

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 « **light** » 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 **sound** ; la bibliothèque **ranger** 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

Fiche méthode

F. GIROD

Pour aller plus loin

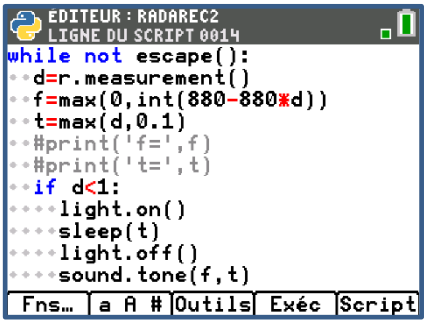
On propose sur la base de ce projet l'approfondissement suivant qui donne un rendu plus réaliste pour le radar de recul :

un 'bip' qui devient de plus en plus fréquent et de plus en plus aigu au fur et à mesure que l'on se rapproche de l'obstacle

L'objectif est double :

- Complexifier le projet.
- Le rendre plus proche de la réalité.

Pour cela, on peut créer deux variables : **f** et **t** qui dépendent de la variable **d** :



- On a choisi pour cette fréquence un maximum de 880 Hz avec un son qui commence à être audible à moins de 1 m.
  Il faut s'assurer que la valeur de **f** est positive, c'est pourquoi on a utilisé la fonction **max** ; par ailleurs, la valeur de **f** doit être un nombre entier, d'où l'utilisation de **int**.

- On a choisi un intervalle de temps entre deux bips égal à **d**.
  Il faut s'assurer que cette valeur soit au minimum égale à 0,1 pour que la fonction **sound.tone** fonctionne. C'est pourquoi on a utilisé la fonction **max** avec la valeur 0.1 pour être sûr de ne pas passer sous cet intervalle de temps.

On pourra bien sûr modifier les expressions de **f** et **t**. Il faudra être attentif à ce que :

- Les valeurs soient du bon format (entier pour **f**).
- Les valeurs soient dans des intervalles compatibles avec l'effet voulu (des fréquences trop aigues ou trop graves ne sont pas audibles).
- Les valeurs aient des valeurs compatibles avec le fonctionnement des fonctions associées (un temps inférieur à 0,1 s pour **sound.tone** ne convient pas).

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