

Unit 3 : Luminosité, IF, et WHILE

Application : Harpe lumineuse

Dans cette application, vous allez écrire un programme pour contrôler le son provenant du haut-parleur en fonction de l'évolution de la luminosité détectée par un capteur de lumière et de votre propre mouvement de main d'où, une harpe lumineuse !

**Objectifs :**

- Ecrire un programme qui convertit la luminosité en son
- Revoir les notes de musique, leur fréquence ainsi que la progression de celles-ci selon la racine douzième de 2.

Ecrire un programme qui lit la LUMINOSITE reçue par un capteur de lumière et joue un son différent en fonction de la luminosité. Il y a deux options possibles pour le son :

- Jouer une fréquence contenue dans une portion du spectre audible (peut-être 100Hz – 1000Hz)
- Jouer une note de musique (un des sons harmonieux spécifiques trouvé sur un piano ou un autre instrument de musique)

La première option consisterait à jouer simplement un bruit et la seconde à réellement faire de la musique, mais cela fait appel à des théories mathématiques plus complexes

Ce programme rend la TI-Innovator™ Hub se comporte comme un « theremin ».

**Commençons**

1. Débuter un nouveau programme et le nommer APPLIC3.
2. Ajouter les commandes **Effécran** et **Disp** afin d'afficher le titre comme montré sur l'écran de droite.
3. Initialiser la variable variable **b** en ajoutant l'instruction **2→B**.
4. Ajouter une boucle **While** ainsi qu'une instruction **Read** pour lire la valeur de la luminosité (**BRIGHTNESS**) issue du capteur et obtenir sa valeur **Get** dans la variable **B**.
5. Ajouter la commande pour jouer le son.
  - Noter que nous utiliserons la variable **B** pour lire la LUMINOSITE et la variable **F** pour jouer le SON.

Votre travail consiste à compléter le code manquant qui convertit la LUMINOSITE en un *son* audible ou une *note de musique*.

Pour le son, utiliser une fréquence comprise dans l'intervalle 100 à 1000 (ou deux fréquences au choix).

Pour les notes de musique, essayer avec le LA 1 (A1) (55Hz) et en remontant la gamme de 50 notes. (Vous devriez vous référer à l'activité de l'Unité 2, Compétence 3 – prgmSON2— qui joue les 12 notes sur une octave.

Pour les notes de musique, vous devrez convertir votre valeur en un nombre entier de sorte qu'une note « nombre » soit correctement représentée.

Vous pouvez utiliser la fonction **ent( )** ou la fonction **arrondir( ,0)**.

**ent(X)→X** donne le plus grand nombre entier inférieur à X.

**arrondir(X,0)→X** arrondi X à l'entier le plus proche.

```
NORMAL FIXE4 AUTO REEL RAD MP
PROGRAM:APPLIC3
:2→B
:While B>1
:Send("READ BRIGHTNESS ")
:Get(B)
:
:Send("SET SOUND eval(F)")
:Wait 0.2
:End
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
MUSIQUE A LA MAIN
B=24.024904
N=11
F=103.8261744
```

Ce document est mis à disposition sous licence Creative Commons



<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>



**Conseil à l'enseignant :**

Pour jouer juste un son, les étudiants doivent convertir B depuis (0,100) en F dans (100,1000).

B	F
0	100
100	1000

Calculer la pente de la droite passant par ces deux points, puis écrire l'équation de F en fonction de B.

$$M = (1000-100)/(100-0) = 9$$

soit

$$F = 9*B+100$$

Pour jouer les notes de musique  $F*2^{(1/12)}$  donne la propriété des intervalles de notes. Si on donne  $A1=55\text{Hz}$  et que nous voulions 50 notes, la première d'entre elles est #0 et la dernière est #49 (les programmeurs commencent toujours à compter à partir de 0). Aussi souvenez-vous que le nombre de notes doit être entier, ainsi nous devons utiliser `ent()` ou `arrondir()`. Nous utilisons deux lignes de commandes afin de rendre cette partie compréhensible.

$$N=\text{ent}(49B/100) \quad \text{-- nombre de notes}$$

$$F=55*2^{(N/12)} \quad \text{-- fréquence de la note}$$

Ce document est mis à disposition sous licence Creative Commons



<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>