

Alarme de tiroir

A. Yazı

Référentiel, Compétences

Lycée :

- Mesurer et traiter un signal au moyen d'une interface ou d'un microcontrôleur.
- Commander la production d'un signal grâce à un microcontrôleur.
- Emission et propagation d'un signal sonore.

Lycée Professionnel :

- Mettre en œuvre une chaîne de transmission par canal sonore.
- Mettre en œuvre des émetteurs et des capteurs piézoélectriques.

Compétences :

- **S'approprier** : Traduire des informations, des codages.
- **Analyser Raisonner** : Elaborer un algorithme.
- **Réaliser** : Expérimenter – en particulier à l'aide d'outils numériques.
- **Valider** : Proposer d'éventuelles améliorations d'une démarche.
- **Communiquer** : Expliquer une démarche.

Situation déclenchante

Situation 1 : Vincent aime jouer avec sa guitare, malheureusement il n'arrive pas à accorder son instrument quand il se désaccorde. Son professeur de musique lui a conseillé cette méthode d'accordage illustrée ci-dessous :



En utilisant la carte micro:bit, pouvez-vous l'aider à accorder sa guitare ?

Situation 2 : A l'occasion de l'anniversaire du professeur de physique-chimie, les élèves de sa classe souhaitent lui offrir l'incontournable « Happy Birthday » et une autre chanson « Bella Ciao ». Grâce au module « music » de la carte micro:bit, arriveront-ils à réaliser leur projet ?

Situation 3 : Dans le tiroir de Boris plusieurs objets disparaissent mystérieusement, alors il décide de créer une alarme de tiroir. Pourriez-vous l'assister à le concevoir ?



Problématique

Comment générer une note musicale à l'aide de la carte micro:bit ?
 Comment jouer une mélodie ou composer une partition musicale ?
 Comment créer une alarme de tiroir ?

Alarme de tiroir

A. Yazı

Matériel nécessaire

- Une calculatrice TI-83 Premium CE Edition Python.
- Une carte BBC micro:bit.
- Un câble miniUsb-microUsb.
- Des fils de connexion.
- Un haut-parleur ou écouteurs.
- Une carte d'extension (Shield Grove).
- Un capteur de lumière Grove.
- Un buzzer Grove.
- Une Led Grove.

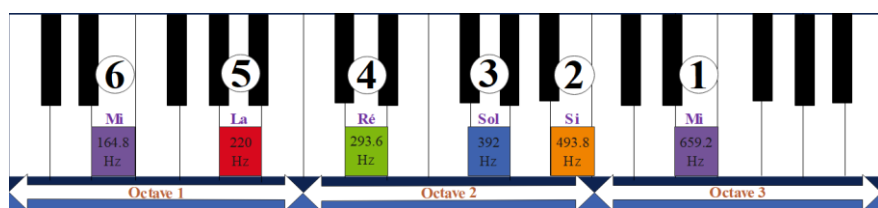
Proposition de protocole

Situation 1 :

Le son produit par une corde vibrante, pour une tension donnée, est lié à sa longueur d'onde, dont dépend sa fréquence de vibration.

En musique, une note est symbolisée par une lettre permettant de représenter un fragment de musique par une convention d'écriture de la hauteur et de la durée d'un son.

Lorsqu'on fait varier la longueur d'onde, et donc la fréquence de la corde, dans un rapport 2, on obtient des sons séparés d'une octave.



Mi₂ devient Mi₃ lorsqu'on divise la longueur de la corde par 2, donc lorsqu'on multiplie sa fréquence de vibration par 2. Si on multiplie encore la fréquence par 2, on obtient Mi₄, et ainsi de suite.

La carte micro:bit possède un module de musique très intéressant. En connectant un haut-parleur à la carte, il est possible de jouer de réelles partitions de musique, ou encore des mélodies préenregistrées. Chaque note de musique est définie par : **note + octave + durée**. La note dans une octave donnée est définie par une fréquence de vibration acoustique.

La correspondance de nos jours, s'établit comme suit :

la	si	do	ré	mi	fa	sol
A	B	C	D	E	F	G

Exemple : le code "**R4:2**" ou "**R4:2**" correspond à la note **la** (440 Hz), **4e** octave et de durée **2 crochets**. Pour jouer cette note, il suffit de mettre la série des notes à jouer les unes à la suite des autres dans **une liste**.

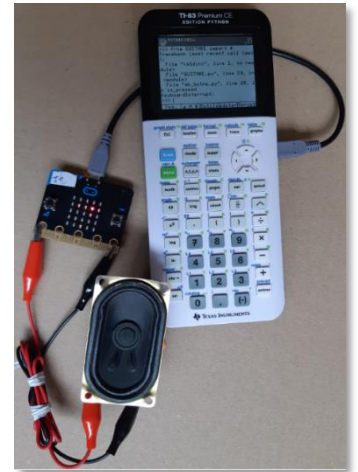
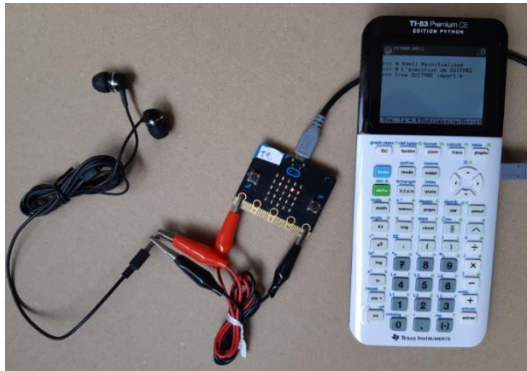


Alarme de tiroir

A. Yazı

Proposition de solution

Réaliser le branchement en utilisant un haut-parleur de récupération ou des écouteurs (pour éviter le bruit en classe).



Le script prévoit le fait de maintenir une note générée par la carte le temps nécessaire pour accorder la note sur la guitare. Pour générer la deuxième note il suffit d'appuyer sur le bouton **a** de la carte, ensuite refaire de même pour les autres notes.

Chargement des modules nécessaires :

- **microbit** pour le fonctionnement de la carte.
- **mb_music** pour la gestion du son.
- **mb_butns** pour pouvoir utiliser les 2 boutons programmables, libellés « A » et « B » sur la carte

Création d'une fonction **accord**.

- Créer une liste **L** dont les éléments sont les six notes consignées dans des listes.
- Créer une boucle bornée qui permettra de lire les éléments de la liste **L**.
- **music.play** permet de jouer la partition composée dans la liste.
- Boucle **Tant que** (le bouton A ou B de la carte micro:bit n'est pas pressée).
 - Jouer la note consignée dans la liste en permanence
 - Une fois le bouton A est pressé, on passe à la deuxième note.

```

ÉDITEUR : ACORGUIT
LIGNE DU SCRIPT 0001
from microbit import *
from mb_music import *
from mb_butns import *
def accord():
    L=[['E5:S'], ['B4:S'], ['G4:S'],
       ['D4:S'], ['A3:S'], ['E3:S']]

    for i in range(0,6):
        while not button_a.is_pressed():
            music.play(L[i], wait=True)
    return L, "instrument accordé"
  
```

Alarme de tiroir

A. Yazı

Situation 2 :

Quant aux élèves de la classe de seconde qui souhaitent offrir l'incontournable « Happy Birthday » à l'occasion de l'anniversaire de leur professeur, la carte intègre un certain nombre de musiques préenregistrées pour différentes ambiances.

Au niveau des branchements on peut procéder comme dans la situation 1.

Il existe plusieurs modules de musique sur laquelle la carte micro:bit s'enfiche, qui intègrent un haut-parleur. La carte peut alors émettre des sons sans branchements supplémentaires. C'est le cas de **Base:bit V0.2**.



Module **base:bit** permettant l'alimentation de votre carte micro:bit. Ce module comporte également un inverseur marche-arrêt, une led d'indication bleue et un buzzer.

Chargement des modules nécessaires :

- **microbit** pour le fonctionnement de la carte.
- **mb_music** pour la gestion du son.
- **music.play** permet de jouer :
 - La mélodie préenregistrée sur la carte.
 - La partition personnelle dans la liste.

```

ÉDITEUR : ANIVPROF
LIGNE DU SCRIPT 0003
from microbit import *
from mb_music import *
music.play("music.BIRTHDAY",wait
=True)
partition=["A4:2","B4:2","C5:2",
"A4:2","A4:2","B4:2","C5:2",
"A4:2","A4:2","B4:2","C5:4",
"B4:2","A4:2","C5:4","B4:
2","A4:2","E5:2","E5:2","E5
:2","E5:2","D5:2","E5:2","F
5:2","F5:2","A4:2","G5:2","
F5:2","E5:2","D5:2","C5:2",
"B4:2","E5:2","C5:2","B4:2",
,"A4:2"]
music.play(partition,wait=True)
    
```

Situation 3 :

Pour protéger les biens de Boris, on va simuler une alarme, en utilisant un capteur de luminance (sensation visuelle de luminosité d'une surface) qui se trouve à l'intérieur du tiroir. Tant que le tiroir est fermé, il ne se passe rien. A l'ouverture, le capteur détecte la lumière qui pénètre dans le tiroir, une DEL et un buzzer se déclenchent pour alerter une intrusion.

On propose de fixer le seuil de luminance le plus bas par exemple à 2 (il fait sombre quand le tiroir est fermé).

(Plage de luminance : 0 à 1000 lux).



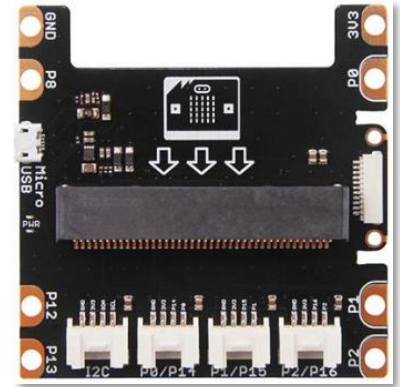
Alarme de tiroir

Utilisation d'un capteur « Grove »

Connecter la carte micro:bit sur une carte d'extension (Shield Grove).

Cette carte permet de brancher 4 modules à l'aide de 4 connexions : une I2C (pour, entre autres, le capteur de mouvement, l'afficheur 7 segments et le capteur de température), ainsi qu'une P0/14, une P1/15 et une P2/16 pour les autres composants.

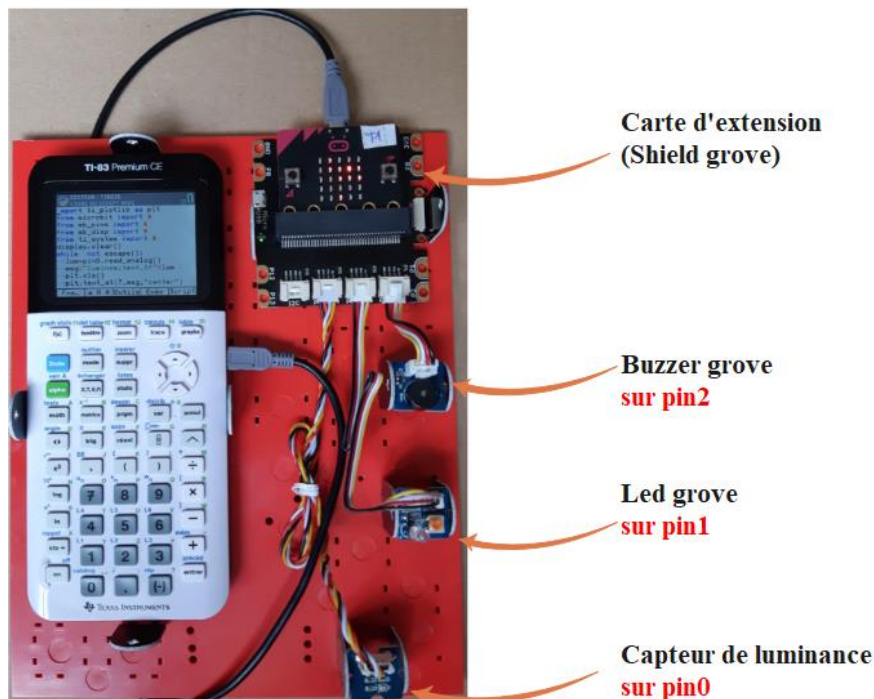
Ce type de connexion sera à privilégier si l'on souhaite se concentrer sur le fonctionnement par bloc des composants électroniques, et surtout pour une mise en œuvre rapide.



Il est toujours possible de connecter ces composants sur une platine de connexions sans soudures.

Montage

Connecter les 3 capteurs sur la carte d'extension (Shield Grove).



Alarme de tiroir

A. Yazı

Algorithme

Tant que

l ← mesure de la luminance (pin0).

Si l > s #Seuil à ajuster, la fonction lum() donnera l'ambiance de la pièce.

Envoi d'un signal sonore (Pin2).

Envoi d'un signal lumineux (Pin1).

Fin si

Fin Tant que :

Chargement des modules nécessaires :

- **microbit** pour le fonctionnement de la carte.
- **ti_plotlib** pour les représentations graphiques.
- **mb_pins** connexion aux ports P0 à P2.
- Boucle **Tant que** (la touche **annul** n'est pas pressée)
 - Lecture de la luminance et affectation à la variable **l**.
 - **plt.cls()** effacement de l'écran de la calculatrice.
 - **plt.text** affichage de la valeur lue avec une décimale.
 - Si la luminance dépasse le seuil fixé, déclenchement d'un signal sonore et d'un signal lumineux.

```

ÉDITEUR : ALARMTIR
LIGNE DU SCRIPT 0004
import ti_plotlib as plt
from microbit import *
from mb_pins import *
from ti_system import *
#ambiance lumineuse de la piece
def lum():
    while not escape():
        lum=pin0.read_analog()
        plt.cls()
        msg="luminance=%.1f lux"%lum
        plt.text_at(7,msg,"center")
#declenchement alarme
def alarme(s):
    while not escape():
        l=pin0.read_analog()
        plt.cls()
        msg="luminance=%.1f lux"%l
        plt.text_at(7,msg,"center")
        if l>s:
            pin2.write_digital(1)
            pin1.write_digital(1)

```

Remarque

Un capteur de niveau sonore peut remplacer le capteur de luminance au cas où l'intrus ouvre le tiroir en l'absence de lumière.

On peut de la même manière, fixer un seuil sonore à partir duquel le déclenchement du buzzer aura lieu.



Capteur de niveau sonore

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



Ce document est mis à disposition sous licence Creative Commons
<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>

© Texas Instruments 2020 / Photocopie autorisée