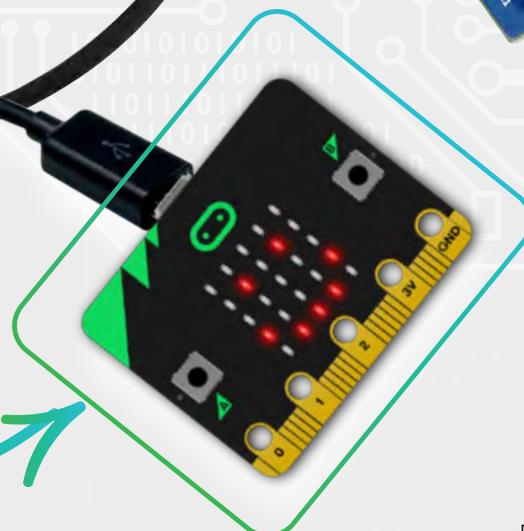
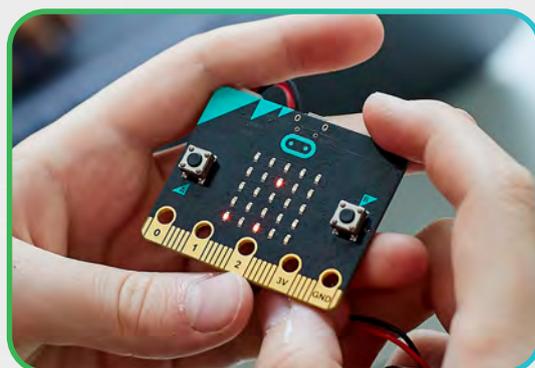
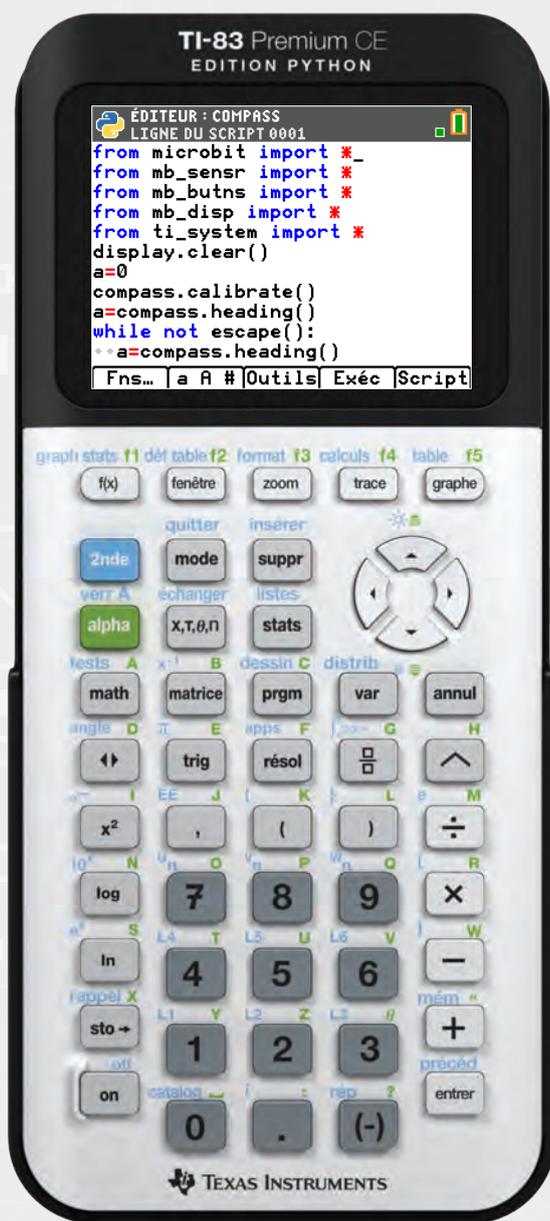


Activités BBC micro:bit avec la TI-83 Premium CE Edition Python

Enseignement de la physique-chimie au lycée



Sommaire

Utiliser la carte BBC micro:bit	Présentation et installation	J.L&A.Y	3
Mesurer une tension	Concevoir un testeur de piles	J.L	8
Mesurer une luminosité	Adapter un niveau de luminosité	J.L	15
Mesurer une pression	Loi de Boyle-Mariotte	A.Y	22
Acide-Base	Mesurer le pH d'une solution	A.Y	29
Objets connectés	Fabriquer un niveau numérique	J.L	36
Utiliser un capteur d'humidité	Mesurer le taux d'humidité	A.Y	40
Modes de transfert thermique	Se protéger de la chaleur	J.L	48
Propagation d'une onde sonore	Alarme de tiroir	A.Y	54
Acoustique	Mesure de la vitesse du son dans l'air	J.L	60
Concevoir un télémètre	Mesurer une distance	A.Y	64
Phénomènes magnétiques	Fabriquer une boussole	J.L	71
Etudier un mouvement	Accélération d'un ascenseur	J.L	77
Mesure de la conductance	Détecter une émotion	A.Y	81
Acquisition de données	Mesurer une accélération	J.L	85

Présentation et installation

La carte BBC Microbit



BBC micro:bit (que nous appellerons dans la suite micro:bit) est une carte micro-contrôleur qui a été conçue par la BBC, accompagnée de 29 partenaires dont Samsung, ARM et Microsoft.

Elle permet d'aborder la robotique avec divers langages et elle autorise la conception de systèmes automatisés en technologie et sciences.

Elle mesure seulement 52 x 42 mm et elle est conçue pour les jeux et les applications connectées. Elle est programmable depuis les PC, les appareils Android et IOS et maintenant pour la calculatrice graphique TI-83 Premium Edition Python.

La carte Micro:bit dispose des spécificités techniques suivantes :

Un bouton de réinitialisation, libellé « R » pour « Reset ».

Deux boutons programmables, libellés « A » et « B » sur la carte.

Un accéléromètre pour détecter les variations d'accélération de l'appareil (qui permet par exemple de détecter des actions comme secouer, pencher et la chute libre).

Une boussole magnétique 3D pour détecter les champs magnétiques.

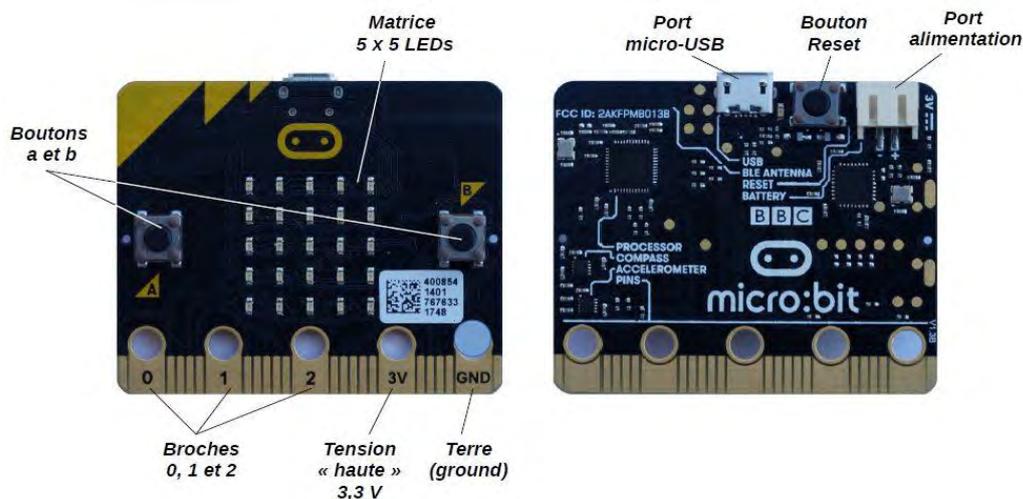
Un capteur de température (sur le processeur).

Un capteur de luminosité lié aux diodes.

Une connectique Bluetooth 4.0 basse énergie/2.4 GHz maître/esclave .

Elle contient également :

Un afficheur digital carré de 25 LED (5 x 5) rouges programmables pouvant servir d'affichage, notamment pour des motifs animés, ou encore pour afficher un texte alphanumérique défilant.



Présentation et installation

Installation des modules dans la calculatrice

Les modules nécessaires au fonctionnement de la carte BBC micro:bit connectée à la calculatrice TI-83 Premium CE Edition python sont les suivants :

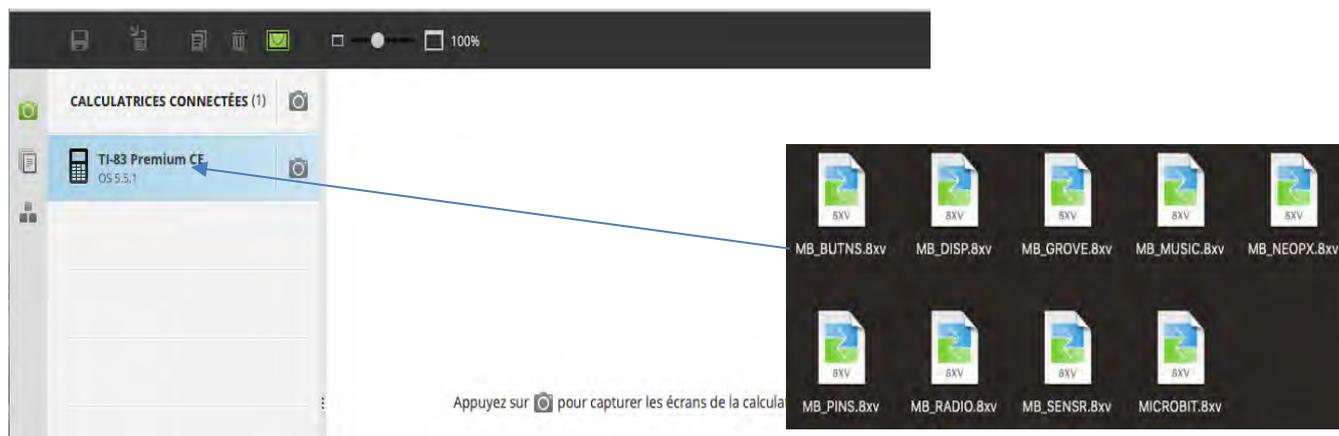
MICROBIT.8xv	Module général de gestion de la carte micro:bit.
MB_GROVE.8xv	Module de gestion des capteurs de type « grove »
MB_SENSR.8xv	Module de gestion des capteurs intégrés à la carte.
MB_RADIO.8xv	Module permettant les communications radio entre plusieurs cartes.
MB_PINS.8xv	Module de communication avec les ports P0 à P2.
MB_BUTNS.8xv	Module permettant d'utiliser les boutons A et B de la carte.
MB_DISP.8xv	Module pour gérer l'affichage des DEL.
MB_MUSIC.8xv	Module pour faire de la musique avec la carte.
MB_NEOPX.8xv	Module pour utiliser une carte Neopixel.

L'installation des modules s'effectue à l'aide du logiciel TI Connect™ CE : <https://education.ti.com/fr>

Connecter la calculatrice à un ordinateur en utilisant le câble Usb A – Usb mini.

Ouvrir le dossier contenant les fichiers relatif à la carte micro:bit (disponibles sur le site précédemment cité).

Sélectionner l'ensemble des fichiers, puis les glisser sur l'icône de la calculatrice.



Présentation et installation

J-L Balas & A Yazı

Appeler un module dans un script

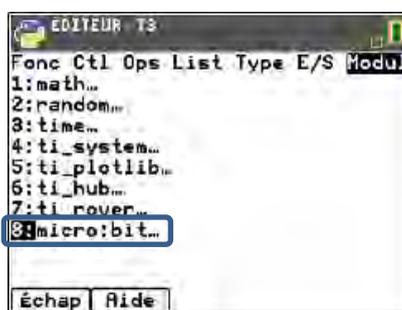
Dans l'application Python, créer un nouveau script.

Appuyer sur **[2nde]** **[0]** puis **[alpha]**™ afin d'accéder au catalogue, puis à la page des instructions commençant par la lettre F.

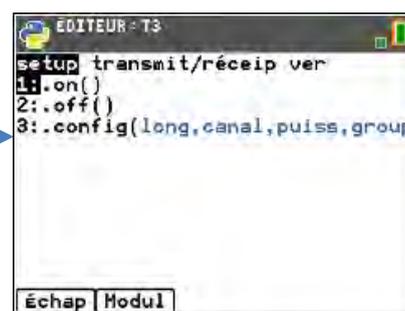
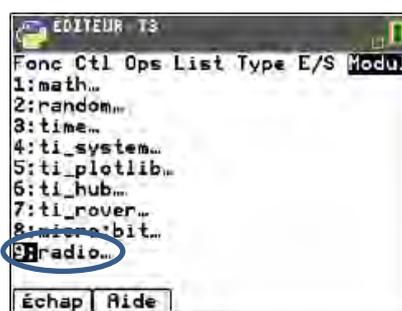
puis **[v]** afin de choisir l'instruction « from **SCRIPT** import* » .

A l'emplacement prévu (le curseur clignote). Ecrire le nom du module « **microbit** » à la place du mot **SCRIPT**.

Les autres modules relatifs à la carte micro:bit seront ensuite appelés à partir du menu « microbit » qui vient d'être rajouté dans les modules.



Lorsqu'un script nécessitera un module inclus dans le module global du micro:bit, par exemple, le module « **radio** », il suffira de le choisir dans cette liste, pour bénéficier ensuite des fonctions qui le concerne. Ainsi le module **radio** sera rajouté à la liste des modules.



Préparer la carte BBC micro:bit

Afin de rendre compatible le fonctionnement de la calculatrice graphique avec la carte micro:bit, un fichier dont l'extension est **.hex** (TI_Runtime_for_Microbit.hex) doit être chargé dans la carte BBC micro:bit.

Relier la carte micro:bit à l'ordinateur par l'intermédiaire d'un câble USB A – micro USB. La carte est vue comme un périphérique.

Glisser déposer le fichier **.hex** sur l'icône du périphérique.

Lors du chargement du fichier dans la carte, la diode située à côté du bouton « reset » clignote.

Ensuite, votre carte BBC micro:bit est prête à être utilisée avec la calculatrice TI-83 Premium CE Edition Python.



Présentation et installation

Tester la connexion de la carte BBC micro:bit

Pour tester la connexion de la carte BBC micro:bit, se munir d'un câble mini USB – micro USB.

Relier le câble micro USB à la carte micro:bit.

Relier le câble mini USB à la calculatrice.

La calculatrice se met automatiquement sous tension.

Appuyer éventuellement sur le bouton **reset** de la carte.

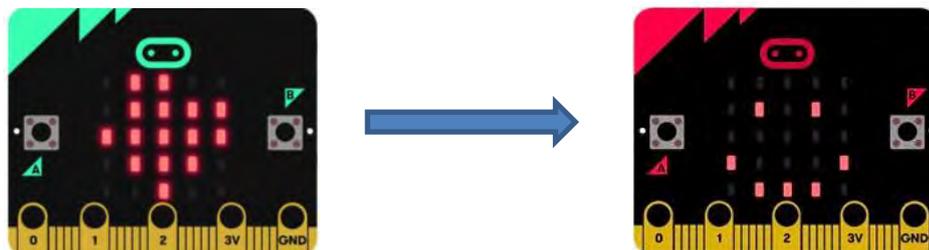
La carte doit afficher le logo de Texas Instruments sur sa matrice de DEL.

Créer un script appelé TEST comme montré ci-contre.

Exécuter votre script.

Si votre calculatrice communique correctement avec votre carte, un smiley doit être affiché à la place du logo de Texas Instruments.

```
ÉDITEUR : TEST
LIGNE DU SCRIPT 0001
from microbit import *
from mb_disp import *
display.clear()
display.show("Image.HAPPY",delay
            =400,wait=True)
```



Remarque :

Pour un adaptateur USB mini (femelle) – USB micro (Mâle) et le câble unité-unité livré avec la calculatrice, connecter le côté marqué A vers la calculatrice.



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



Concevoir un testeur de piles

Référentiel, compétences

Lycée :

- Mesurer une grandeur physique à l'aide d'un capteur.
- Utiliser un dispositif comportant un microcontrôleur.
- Variabilité de la mesure d'une grandeur physique.
- Capteurs électriques, loi d'Ohm.

Lycée Professionnel :

- Identifier les grandeurs d'entrée et de sortie (avec leur unité) d'un capteur.
- Lire et représenter un schéma électrique.
- Réaliser un montage à partir d'un schéma.

Compétences :

- **S'approprier** : Rechercher et organiser l'information. Représenter une situation par un schéma.
- **Analyser Raisonner** : Évaluer des ordres de grandeur.
- **Réaliser** : Effectuer des procédures courantes.
- **Valider** : Identifier des sources d'erreurs, estimer une incertitude, comparer à une valeur de référence.
- **Communiquer** : Utiliser un vocabulaire adapté et choisir des modes de représentation appropriés.

Situation déclenchante

La carte BBC micro:bit communique avec le monde extérieur par l'intermédiaire de capteurs intégrés (température, accéléromètre, lumière, magnétomètre). Il est aussi relativement facile d'associer à la carte d'autres types de capteurs. Ceux-ci peuvent être de type « grove » et nécessiter l'insertion du capteur et de la carte micro:bit au sein d'une carte d'extension (shield). Cependant certains capteurs intégrés à la carte sont relativement peu précis. On peut également lors de la réalisation de certains projets, être confronté à la nécessité de connaître certains niveaux de tension électrique (Pour alimenter un servomoteur par exemple). Plus précisément, il peut également s'agir d'adapter la tension de sortie de la carte, à un autre composant.



Problématique

Comment fonctionne une mesure de signal sur la carte BBC micro:bit ?

Quels sont les niveaux de tension admissible ?

Quelle est la résolution de la carte ?

Utiliser la carte micro:bit comme un voltmètre.



Concevoir un testeur de piles

Matériel nécessaire

- Des piles (neuves et usagées).
- Des câbles de connexion micro:bit avec pince crocodile.
- Un potentiomètre « grove ».
- Un Grove Shield pour micro:bit v2.0. Une calculatrice TI-83 Premium CE Edition Python.
- Un câble mini USB - micro USB.

Mesurer une tension avec un microcontrôleur

Lorsque l'on utilise un transducteur (LDR ; photorésistance ; thermistance...), la grandeur mesurée est une tension.

Afin de tenir compte des caractéristiques de ce composant, celui-ci est généralement associé à une résistance au sein d'un montage potentiométrique.

Ainsi la tension mesurée U_s aura pour valeur :
$$U_s = \frac{R_2}{R_1 + R_2} \times U_e$$

Rappel : On aboutit à cette relation par application de la loi des mailles dans le circuit ci-dessus. $U_e = (R_1 + R_2) \times I$ et $U_s = R_2 \times I$

Ensuite on exprime la tension de sortie U_s en fonction de U_e . Il est donc nécessaire de remplacer I de l'une de ces deux équations.

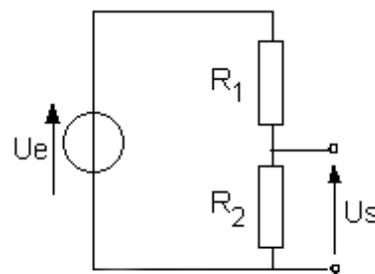
On utilise un capteur « grove » $R = R_1 + R_2 = 10 \text{ k}\Omega$ linéaire Plage angulaire : 300° .

U_e : tension d'alimentation, 3,3V.

U_s : tension de sortie, varie proportionnellement à la position du potentiomètre entre 0-3,3V.

La tension analogique issue du potentiomètre est convertie en une valeur numérique par le CAN de la carte micro:bit (voir ci-dessous).

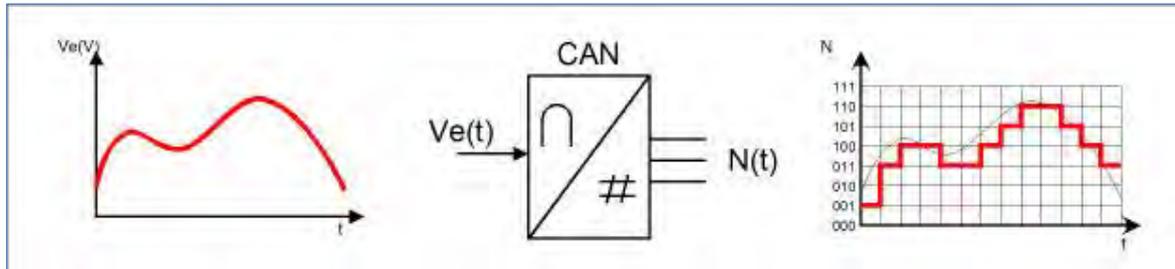
Un CAN converti la grandeur analogique en grandeur numérique.



Concevoir un testeur de piles

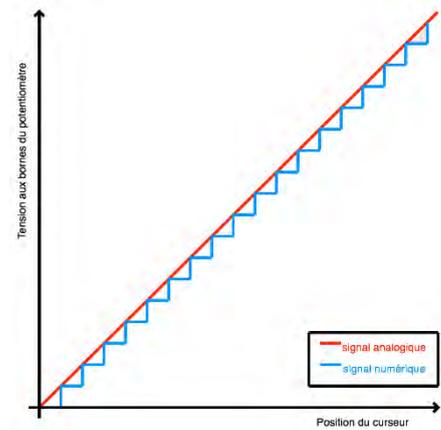
Résolution numérique :

La résolution de la carte est de 10 bits, cela signifie que la tension d'entrée entre 0 et 3,3 V est convertie en une valeur numérique entière comprise entre 0 et 1023.



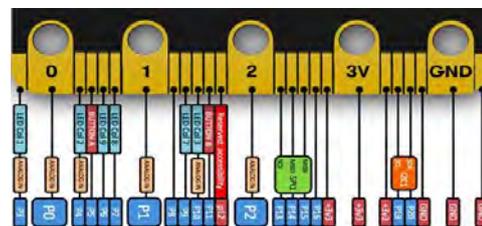
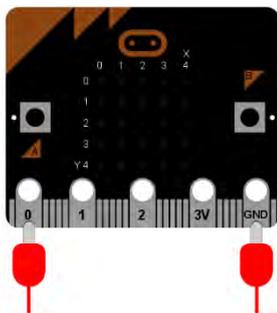
Résolution analogique :

Il en résulte une résolution analogique ou quantum de 3,225 mV (= 3,3 / 1023).



Remarques

La carte micro:bit comporte trois sorties analogiques (0 à 2) et permet d'alimenter un circuit en 3,3 V. Lorsque certains actionneurs (servomoteur, relais) seront connectés aux ports $P_0 \dots P_2$ de la carte, une alimentation extérieure peut être nécessaire.



Concevoir un testeur de piles

Expérimentation

Le potentiomètre "grove" est connecté au port P₀ de la carte BBC micro:bit.

- Chargement des modules nécessaires.
- **mb_grove** gestion des capteurs grove.
- **mb_pins** (connexion aux ports).
- **microbit**.
- **mb_disp** (affichage DEL).

Création d'une fonction **tension**.

- Lecture de l'information sur le port **pin0**.
- Conversion de la valeur lue en tension à partir de la mesure effectuée sur le port pin0 et affecté à la variable **can**.

$$u = \frac{3.3 \times can}{1023}$$

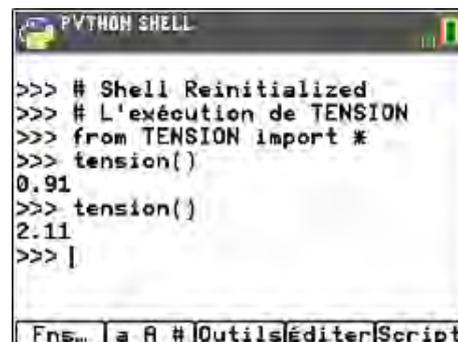
Affichage de la tension u en volt avec une décimale.

Création d'une fonction **aff**. Affichage continu de la tension.

- **sleep(1)** délai de 1s.
- Appel de la fonction **tension()** et affectation de la mesure à la variable u .
- **plt.cls** pour effacer l'écran, puis affichage de la mesure.



```
EDITEUR : TENSION
LIGNE DU SCRIPT 0001
from ti_system import *
import ti_plotlib as plt
from mb_grove import *
from mb_pins import *
from microbit import *
from mb_pins import *
def tension():
    can=pin0.read_analog()
    u=3.3*can/(2**10-1)
    return round(u,2)
def aff():
    sleep(1)
    u=tension()
    msg="Tension en volt=%.2f"%u
    plt.cls()
    plt.text_at(7,msg,"center")
-
```

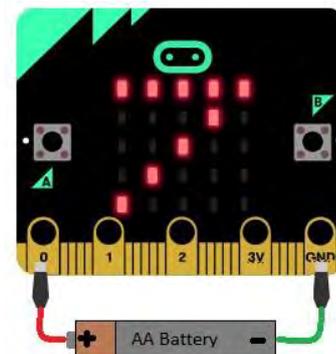


```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de TENSION
>>> from TENSION import *
>>> tension()
0.91
>>> tension()
2.11
>>> |
```

Concevoir un testeur de piles

Utiliser la carte BBC micro:bit comme un voltmètre

- Réaliser le montage ci-dessous.
- Connecter les fils au micro:bit avec les connexions à la broche 0 et la broche de la « terre ». La broche 0 se connectera à l'extrémité positive (+) de la batterie. La terre se connecte à l'extrémité négative (-) de la batterie.
- Collecter un échantillon de données en utilisant une pile avec un voltmètre, permet de « calibrer » le programme du micro:bit.



Lorsque la tension est lue entre les bornes (3,3V – 0V) les informations renvoyées sont autour de 1020. La lecture de la tension analogique est convertie en valeur numérique avec 3,3 volts approchant la limite supérieure de 1023. Une tension de 1,5 volt devrait renvoyer une lecture d'environ 512 sur le micro:bit lors de sa conversion analogique-numérique. Si aucun fil n'est connecté, on peut obtenir une lecture d'environ 25.

Les mesures obtenues peuvent être tabulées comme suit :

Tension (Voltmètre)	Lecture Micro:bit (0 – 1023)
0	276
0	286
1.27	441
1.47	509
1	391
1.45	506

- Préparer les documents de collecte de données.
- Programmer le micro:bit.
- Lorsque la pile est connectée au micro:bit. Le bouton **A** donne une lecture. La lecture est ensuite convertie en millivolts à partir de la lecture numérique sur la broche 0.
- Expérimenter avec différentes piles. Utiliser des piles neuves et des piles plus anciennes.
- Analyser les résultats et observations des expériences.

Concevoir un testeur de piles

Le code est conçu pour renvoyer une valeur convertie de la tension en millivolts (1,5 volt = 1500 millivolts). Au début de la première instruction, une variable « lecture » est créée et une valeur de lecture est donnée à partir de la lecture analogique de la broche 0.

La variable **lecture** est ensuite multipliée par 1000 et divisée par 340 et stockée en tension.

Si 3,3 volts donnent une lecture d'environ 1023, alors 1 volt devrait correspondre à environ 340 (340 = 1024/3). L'utilisation de ce rapport et la multiplication du nombre par 1000 devraient convertir le nombre en millivolts (les micro:bits ne travaillent qu'avec des nombres entiers donc la tension est multipliée par 1000 avant de faire la division par 340).

Mise en œuvre

lecture ← Valeur analogique broche 0

$u1 \leftarrow \frac{\text{lecture} \times 3.3}{1023}$: Tension en volt

$u2 \leftarrow \frac{\text{lecture} \times 1000}{340}$: Tension en millivolt

Si $u1 > u0$

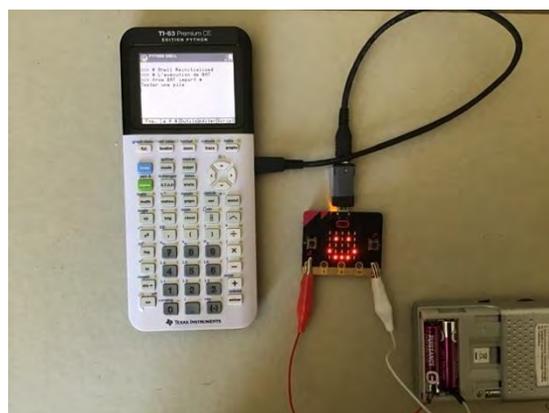
Afficher un smiley "Happy"

Sinon Si $0,9 < u1 < u0$

Afficher un smiley "Normal"

Sinon

Afficher un smiley "Triste"



- Chargement des modules nécessaires, **tiplotlib** (représentations graphiques) et **ti_system**.
- **microbit**.
- **mb_pins** (connexion aux ports).
- **mb_disp** (affichage DEL).

```
EDITEUR : PILE
LIGNE DU SCRIPT 0002
from microbit import *
from mb_pins import *
from mb_disp import *
#Mesure de la tension
def tension(u0):
    lu=1
    sleep(1000)
    lu=pin0.read_analog()
    u1=lu*3.3/1023
    u2=int(u1*1000)
    return u2
Fns... | a A # | Outils | Exéc | Script
```

Concevoir un testeur de piles

- Création d'une fonction `tension` mesurant et renvoyant la tension aux bornes d'une pile de valeur nominale `u0`, l'argument de la fonction.
- `pin0.read_analog` lecture de la valeur sur cette broche.
- Conversion de la valeur lue en tension en volt et conversion en mv.
- Affichage d'une image `display.show` selon le niveau de tension `u2` de la batterie.

```
EDITEUR : PILE
LIGNE DU SCRIPT 0022
#Affichage
def affichage(u0):
    display.clear()
    if tension(u0)>u0:
        msg="Image.HAPPY"
    elif tension(u0)<=u0:
        msg="Image.SMILE"
    else:
        msg="Image.SAD"
    display.show(msg)_
```

Création d'une fonction `affichage` qui prend comme argument `u0`. Cette fonction complète la fonction `tension` et permet d'afficher un smiley sur l'écran de la carte en comparant la valeur mesurée à la tension nominale de la batterie.

```
PYTHON SHELL
Tension en volt 2.9
Tension en millivolt 2891.2
>>> |
```

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

Adapter un niveau de luminosité

Référentiel, compétences

Lycée :

- Mesurer une grandeur physique à l'aide d'un capteur.
- Utiliser un dispositif comportant un microcontrôleur.
- Variabilité de la mesure d'une grandeur physique.
- Capteurs électriques, loi d'Ohm.

Lycée Professionnel :

- Identifier les grandeurs d'entrée et de sortie (avec leur unité) d'un capteur.
- Lire et représenter un schéma électrique.
- Réaliser un montage à partir d'un schéma.

Compétences :

- **S'approprier** : Rechercher et organiser l'information. Représenter une situation par un schéma.
- **Analyser Raisonner** : Évaluer des ordres de grandeur.
- **Réaliser** : Effectuer des procédures courantes.
- **Valider** : Identifier des sources d'erreurs, estimer une incertitude, comparer à une valeur de référence.
- **Communiquer** : Utiliser un vocabulaire adapté et choisir des modes de représentation appropriés.

Situation déclenchante

Le confort de lecture sur un écran de téléphone portable ou celui d'un ordinateur associé au souci d'économie de la batterie nécessitent une adaptation en temps réel de l'intensité lumineuse en fonction de l'environnement lumineux.

Dans l'obscurité, l'écran est éclairé avec une intensité lumineuse minimale afin de limiter l'éblouissement et d'économiser de la batterie.

À l'extérieur, en plein soleil, l'intensité lumineuse de l'écran est à son maximum, assurant une lecture claire de l'affichage. Cette adaptation est la plupart du temps automatique. L'éclairage de l'écran sera représenté ici par le panneau de DEL de la carte BBC micro:bit dont on fera varier l'intensité lumineuse.



Problématique

1. Comment faire varier l'intensité lumineuse de la carte BBC micro:bit à l'aide d'un potentiomètre.
2. Comment mesurer une intensité lumineuse ?
3. Utilisation d'une photorésistance.
4. Capteur intégré à la carte BBC micro :bit.
5. Utilisation d'un capteur « grove ».

Comment adapter l'intensité lumineuse d'un écran ?



Adapter un niveau de luminosité

Matériel nécessaire

- Des fils de connexion.
- Une carte BBC micro:bit.
- Une platine de connexion sans soudures.
- Une photorésistance (LDR) et une résistance de $10k\Omega$.
- Une photorésistance "grove".
- Une lampe de poche.
- Un multimètre.
- Un câble mini USB – micro USB

Mise en œuvre

A) Utiliser une photorésistance pour moduler l'intensité lumineuse d'une DEL.



Une photorésistance est un composant électronique dont la valeur de la résistance, en ohms dépend de l'intensité lumineuse qui l'éclaire.

Vous disposez d'un multimètre permettant de mesurer une résistance s'il est utilisé avec les bornes et COM.

1. En utilisant le matériel à votre disposition répondez à la question suivante : quand la luminosité augmente, la résistance augmente-t-elle ou diminue-t-elle ?
2. Entre quelles valeurs limites semble varier la valeur de la résistance de votre composant ?
3. Compléter le tableau.

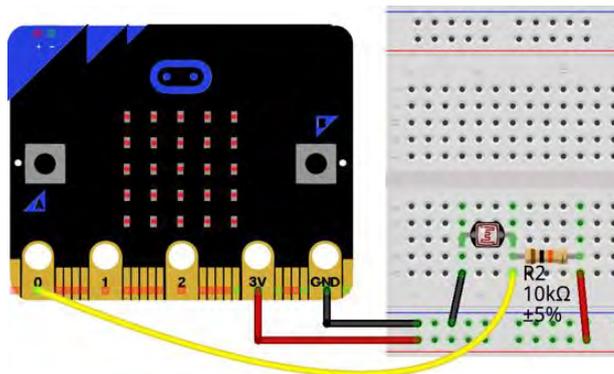
Éclairement de la photo résistance	Valeur de la résistance en Ω
Obscurité :	
Pleine lumière :	
Lumière ambiante de la salle :	

Adapter un niveau de luminosité

a) Algorithme proposé.

```
lum ← lecture de la grandeur analogique sur  
le port 0  
l ← Affectation d'un niveau entre 0 et 9  
Allumer la matrice de DEL selon le niveau de  
tension (0 à 3.3V) reçu  
L'intensité lumineuse des DEL varie de 0 à 9
```

b) Réalisation du montage.



- Chargement des modules nécessaires : **tiplotlib** (représentation graphiques) ; **microbit** (fonctionnement de la carte) **mb_disp** (affichage des DEL) **mb_pins** (accès aux ports P0 à P2).
- Création d'une fonction **lum** réalisant l'acquisition de la tension sur le port P0.
- Conversion de cette tension en une valeur de 0 à 9.
- La fonction renvoie la valeur du niveau de luminosité.
- Création d'une fonction **affiche** destinée à allumer la matrice de DEL selon le niveau mesuré.

```
EDITEUR : LUM0  
LIGNE DU SCRIPT 0001  
from microbit import *  
from mb_disp import *  
from mb_pins import *  
def lum():  
    lum=pin0.read_analog()  
    l=round(lum/1023.*9.)  
    return l  
#allumage de la matrice  
def affiche(l):  
    display.clear()  
    for x in range(5):  
        for y in range(5):  
            display.set_pixel(x,y,l)
```

Adapter un niveau de luminosité

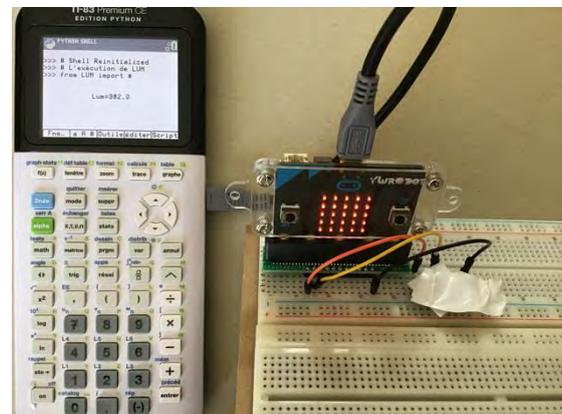
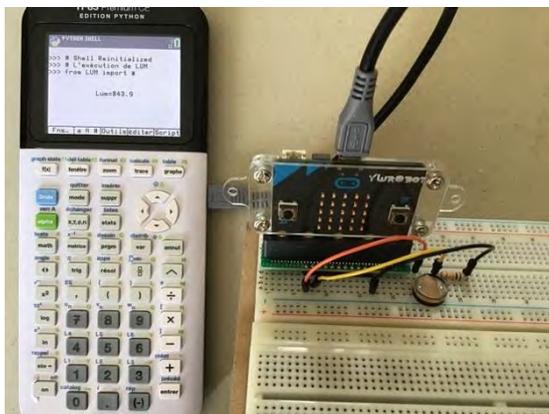
En fonction de la valeur lue, on commande l'affichage de la DEL avec un niveau d'intensité compris entre 0 et 9.

1. Entre quelles limites peut varier la variable lum ? (Masquer totalement la photorésistance, puis la saturer avec une lampe.)
2. Justifier que `round(lum/1023.*9.)` renvoie une valeur comprise entre 0 et 9.
3. Que vaut la variable I si la tension appliquée sur la borne 0 vaut 3,3 V ? puis 1,7 V ?

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de LUM0
>>> from LUM0 import *
>>> lum()
2
>>> I=lum()
>>> affiche(I)
>>> |
```

Utilisation du script : Exécuter dans une console les instructions appelant la fonction lum. Sauvegarder la valeur retournée dans une variable I, puis appeler la fonction affiche.

La matrice de DEL s'allumera avec une intensité lumineuse selon un niveau compris entre 0 et 9 (voir photos ci-dessous)



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de LUM
>>> from LUM import *

Lum=741.0

Fns... a A # Outils Éditer Script
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de LUM
>>> from LUM import *

Lum=310.9

Fns... a A # Outils Éditer Script
```

Adapter un niveau de luminosité

B) Une autre possibilité si l'on ne dispose pas de photorésistance.

Exploiter la fonction de mesure de luminosité intégrée à la carte BBC micro:bit.

Le panneau de DEL peut être utilisé comme capteur de luminosité basique en inversant la polarisation des DEL. Cela ne requiert aucun câblage additionnel.

La mesure du niveau de luminosité passe par la commande ci-dessous.

display.read_light_level()

Elle renvoie une valeur entière comprise entre 0 et 255 : 0 pour l'obscurité et 255 pour le niveau maximal de luminosité.

Ensuite, le programme **lum3** est à adapter sur la base du précédent, afin de pouvoir allumer les DEL selon un niveau d'intensité compris entre 0 et 9.

Afficher le niveau de luminosité sur la console Python :

Le programme simple ci-dessous se contente de renvoyer la valeur de luminosité captée par la carte micro:bit.

- Chargement des modules nécessaires au fonctionnement de la carte micro:bit, à l'affichage sur les DEL **mb_disp** et accès aux ports P0 à P2 **mb_pins**.
- **display.clear()** éteindre la matrice de DEL.
- Création d'une fonction effectuant la mesure de la luminosité sur le capteur intégré à la carte micro:bit.



```
ÉDITEUR : LUM3
LIGNE DU SCRIPT 0009
from microbit import *
from mb_disp import *
from mb_pins import *
from ti_system import *
display.clear()
def niv():
    lum=display.read_light_level()
    return lum
```

C) Déclenchement d'un éclairage à distance :

Avec deux cartes micro:bit, on peut envisager de déclencher l'allumage du panneau de DEL d'une carte à partir de la mesure de luminosité effectuée par une autre carte en utilisant la communication radio.

Remarque

Définir un canal est indispensable en classe pour éviter les brouillages si les cartes communiquent sur le même canal non précisé par défaut.

Fonctionnement : Lorsque la luminosité sera inférieure à une valeur seuil sur la carte émettrice (ambiance sombre), le panneau de DEL de la carte réceptrice sera totalement éclairé.

Il est aussi possible de moduler l'intensité de l'éclairage comme dans les exemples précédents.



Adapter un niveau de luminosité

J-L Balas

Ci-dessous le programme pour la carte émettrice qui effectue la mesure de luminosité.

```

ÉDITEUR : LIGHTLVE
LIGNE DU SCRIPT 0003
from ti_system import *
from mb_disp import *
from mb_radio import *

radio.on()
display.clear()
radio.config(length=32,channel=7
             ,power=6,group=0)
while not escape():
    ll=display.read_light_level()
    print("Luminosité=%.2f" %ll)
    if ll<10:
        radio.send("sombre")
    else:
        radio.send("clair")

```

Ci-dessous le programme pour la carte réceptrice.

```

ÉDITEUR : LIGHTVR
LIGNE DU SCRIPT 0001
from ti_system import *
from microbit import *
from mb_disp import *
from mb_radio import *

radio.on()
display.clear()
radio.config(length=32,channel=7
             ,power=6,group=0)
while not escape():
    ll=radio.receive()
    if ll=="sombre":
        for i in range(5):
            for j in range(5):
                display.set_pixel(i,j,9)
    if ll=="clair":
        for i in range(5):
            for j in range(5):
                display.set_pixel(i,j,0)

```

- Chargement des modules en particulier **mb_radio** (communication radio entre deux cartes ou plus).
- **radio.on()** activer la communication radio sur la carte.
- **display.clear()** éteindre la matrice de DEL.
- **radio.config()** configuration du canal et de la chaîne (évite les conflits entre cartes).
- Boucle Tant que

Sur la carte émettrice : A la variable ll est affecté la mesure du niveau de luminosité. Si ce niveau est inférieur à 10, on envoie le message « sombre », « clair » sinon.

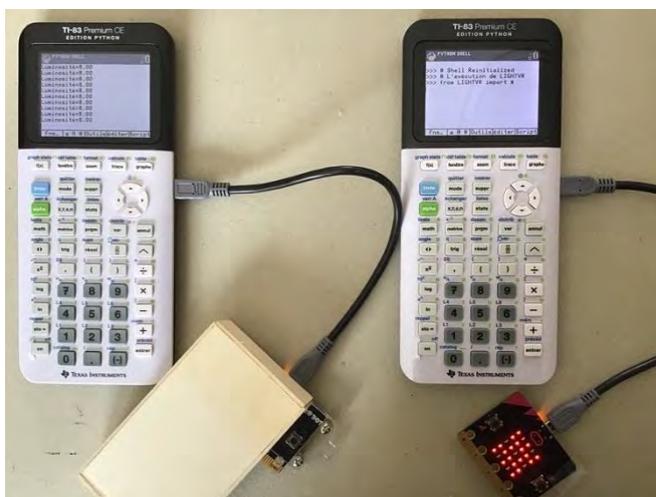
Sur la carte réceptrice : Si le message reçu est sombre, la matrice de DEL est allumée au niveau maximal

Adapter un niveau de luminosité



J-L Balas

Exécuter le script LIGHTVE sur la calculatrice émettrice et LIGHTVR sur la réceptrice.



Remarque



Exemple d'adaptateur de carte micro:bit à une platine de connexions sans soudures.

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



Loi de Boyle-Mariotte

Référentiel, Compétences

Lycée :

- Mesurer une grandeur physique à l'aide d'un capteur.
- Utiliser un dispositif comportant un microcontrôleur.

Lycée Professionnel :

- Mesurer la pression en un point d'un fluide.
- Vérifier expérimentalement la loi de Boyle-Mariotte.

Compétences :

- **S'approprier** : Représenter une situation par un schéma.
- **Analyser Raisonner** : Choisir un modèle ou des lois pertinentes.
- **Réaliser** : Utiliser un modèle.
- **Valider** : Confronter un modèle à des résultats expérimentaux.
- **Communiquer** : Utiliser un vocabulaire adapté.

Situation déclenchante

En 1648, **Blaise Pascal** publie son fameux traité sur la pression atmosphérique et confirme la pensée de **Jean Baptiste Baliani** qui remet en cause l'inexistence du vide. Ensuite, **Boyle** et **Mariotte** établissent leur relation entre pression et volume. La connaissance et la maîtrise de la pression est utilisée dans plusieurs domaines : Météo (cyclone), médecine (tension artérielle), aviation (altimètre avant le GPS), automobile (pneumatiques, suspension), fermeture des portes (TGV, RER), même les chaussures de sport...



La carte micro:bit peut-elle se transformer en appareil de mesure de pression ?

Peut-on reproduire l'expérience de Boyle-Mariotte en utilisant la carte BBC micro:bit en association avec quelques capteurs ? L'air peut-il nous servir aussi de balance ?

Problématique

Comment évolue la pression dans les milieux ?
Comment peut-t-on la mesurer avec la carte BBC micro:bit et un capteur de pression ?
Etalonner un capteur de pression (annexe).
Concevoir une balance grâce au principe fondamental de l'hydrostatique.

Principe fondamental de l'hydrostatique

La pression est une grandeur scalaire proportionnelle à l'intensité de la force F et inversement proportionnelle à la surface S sur laquelle s'exerce la force :

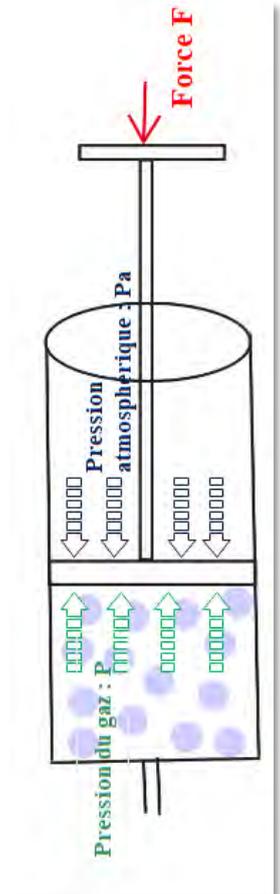
$$p = \frac{F}{S}$$

en pascal (Pa) ← en Newton (N) en m²

Soit un fluide en équilibre qui subit une force F :

La différence de pression est donnée par la relation suivante : $P - P_a = \frac{F}{S}$

La pression atmosphérique $P_a = 101\,325\text{ Pa} = 101,3\text{ kPa}$



Matériel nécessaire

- Une calculatrice TI-83 Premium CE Edition Python.
- Une carte BBC micro:bit.
- Un câble USB mini-USB micro:bit.
- Un capteur grove « pression ».
- Une carte d'extension (Shield grove).

Mesurer une pression

Réaliser le montage suivant :

Le capteur utilisé dans cette expérience est un capteur Grove **MPX5700AP** capable de mesurer la pression de l'air dans une plage de 15 kPa à 700 kPa, il inclut une seringue de 60 ml et un tuyau translucide souple et qui possède une sensibilité de 6,4 mV/kPa.

Une proposition d'étalonnage de ce capteur se trouve en annexe de cette activité.



Le script permet d'afficher **la pression mesurée en kPa**.

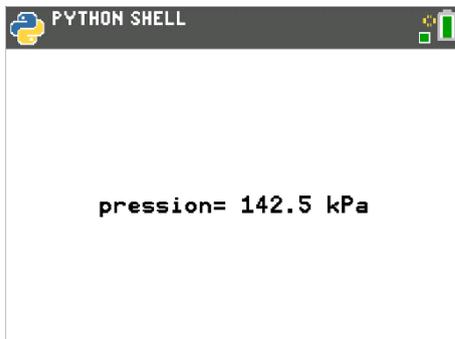
La disponibilité de la commande du module Grove pour la pression rend la réalisation de ce script très facile.

Chargement des modules nécessaires :

- `micobit` pour le fonctionnement de la carte.
- `ti_system`.
- `mb_ti_ptotlib` pour les représentations graphiques.
- `mb_grove` pour la gestion des capteurs grove.
- Création d'une fonction `press`.
 - Boucle **Tant que** (la touche **annul** n'est pas pressée)
 - Lecture de la valeur de la distance et affectation à la variable `p`.
 - Affichage de la valeur lue avec une décimale au

```
ÉDITEUR : PRESS
LIGNE DU SCRIPT 0005
import ti_plotlib as plt
from microbit import *
from ti_system import *
from mb_grove import *
#Mesure de pression
def press():
    while not escape():
        p=grove.read_pressure(pin0)
        msg="pression= %.1f kPa"%p
        plt.cls()
        plt.text_at(7,msg,"center")
        sleep(10)
```

Compression $p > p_a$



Dépression $p < p_a$



Remarque

Au niveau microscopique, La pression est due à l'agitation incessante des atomes ou des molécules qui se déplacent à grande vitesse. De ce fait une pression nulle correspond à un vide parfait.



Application

Balance à base d'air

On se propose de déterminer la masse inconnue posée sur le piston. Ce dernier va effectuer une course allant de P_1 à une pression P_2 (qui sera mesurée avec notre dispositif).

Loi fondamentale de l'hydrostatique donne :

$$P_2 - P_1 = \frac{F}{S} = \frac{mg}{\pi r^2} \quad m = \frac{(P_2 - P_1) \times \pi \times r^2}{g}$$

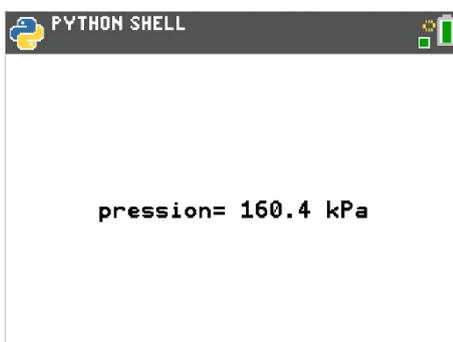
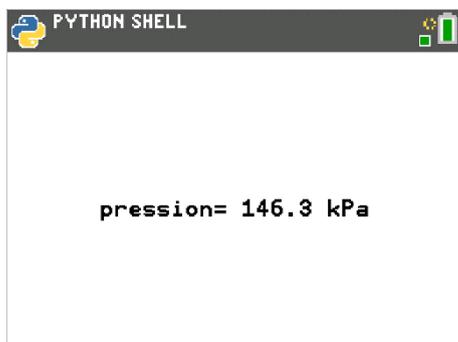
d'où

- $r = 1,5 \times 10^{-2} \text{ m}$
- $g = 9,81 \text{ kg.m.s}^{-2}$
- $m = 7,2 \times 10^{-5} \times (P_2 - P_1)$

Expérience :

P_1

P_2

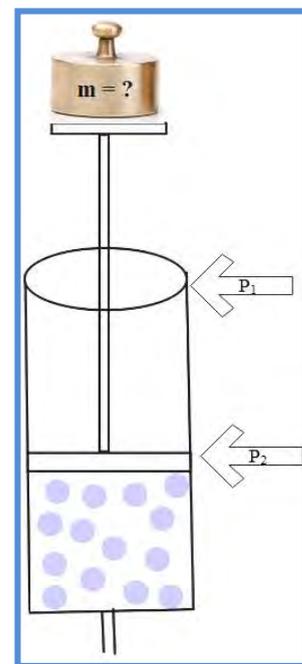


Calcul de la masse m :

$$P_2 - P_1 = 14,1 \text{ kPa} = 14\,100 \text{ Pa.}$$

$$m = 7,2 \times 10^{-5} \times 14\,100 = 1,015 \text{ kg.}$$

Effectivement, il s'agit d'un paquet d'un kilogramme de riz.



Loi de Boyle-Mariotte

Enoncé de la loi :

A température constante, le volume d'une masse gazeuse est inversement proportionnel à la pression.

Mathématiquement : $P = \frac{k}{V}$ ou $P \times V = k$

A l'aide d'une seringue de 60mL, un tuyau souple, du montage comportant le capteur de pression et la carte micro:bit, on se propose de vérifier la loi de Boyle-Mariotte.

Facultativement, on a ajouté un capteur de température pour confirmer la constance de la température.

Chargement des modules nécessaires : `tiplotlib` (représentation graphique), `ti_system` (export des données), `microbit` (gestion de la carte), `mb_pins` (gestion des ports) et `mb_grove` (capteurs grove)

- Création d'une fonction `mesure`.
- Lecture de la valeur de la pression et affectation à la variable `p`.
- Lecture de la valeur de la température et affectation à la variable `T`.
- `plt.cls()` : effacer l'écran.
- `plt.auto.window` : réglage automatique de la fenêtre graphique.
- `plt.axes` : affichage des axes.
- `plt.labels` : étiquettes sur les axes.
- `plt.title` : Titre du graphique.
- `plt.color` : couleur rouge pour la représentation des données.
- `plt.scatter()` : motif de représentation.
- `plt.show_plot()` : Met en pause l'affichage.

On peut réaliser la même expérience en exprimant la pression P en fonction de l'inverse du volume, ainsi on obtient la constante k qu'on peut expliciter en adoptant le modèle du gaz parfait : $PV = K = nRT$, avec :

T : la température (°K).

R : la constante des gaz parfaits = $8,314 \text{ J} \cdot \text{K}^{-1} \cdot \text{mol}^{-1}$.

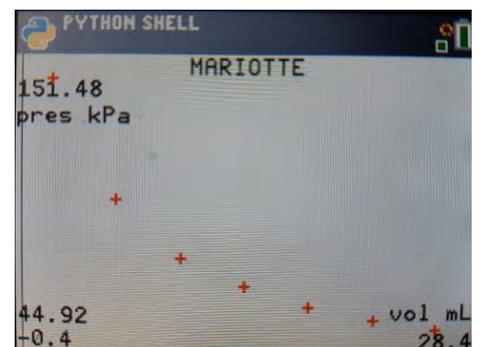
n : nombre de moles contenus dans la seringue.

```
ÉDITEUR : BOYLEMAR
LIGNE DU SCRIPT 0081
import tiplotlib as plt
from microbit import *
from mb_pins import *
from ti_system import *
from mb_grove import *
def mesure():
    p=grove.read_pressure(pin0)
    T=grove.read_temperature(pin1)

    vol=[2,6,10,14,18,22,26]
    pres=[]
    for i in range(len(vol)):
        msg1="Volume fixe a= %.1f"%
            vol[i]
        msg2="Temperature = %.1f"%T
        plt.cls()
        plt.text_at(5,msg1,"center")

        plt.text_at(8,msg2,"center")

        sleep(3000)
        p=grove.read_pressure(pin0)
        pres.append(p)
# representation graphique
vol=[2,6,10,14,18,22,26]
plt.cls()
plt.auto_window(vol,pres)
plt.axes("on")
plt.labels("vol mL","pres kPa",
            ,11,3)
plt.title("MARIOTTE")
plt.color(255,0,0)
plt.scatter(vol,pres,"+")
plt.show_plot()
Fns... a A # Outils Exéc Script
```



ANNEXE :

Etalonnage d'un capteur de pression

Deux méthodes peuvent être proposées à partir d'une recherche personnelle :

- **A partir des caractéristiques du capteur :**

Plage de 15 kPa à 700 kPa.

Sensibilité de 6,4 mV/kPa.

A une tension, on va lui associer une pression : $1kPa = 1000Pa \longrightarrow 6,4mv = 0,0064 V$
 $P(Pa) \longrightarrow U(V)$

Donc : $U(V) = 6,4 \times 10^{-6} \times P$

La valeur analogique associée est : $A = \frac{U \times 1023}{3,3} = 0,001984 \times P$

Exemple

Si $P = 1atm = 101\,325 Pa$, on doit obtenir une tension $U = 0,65V$ délivrée par le micro:bit et une valeur analogique $A = 201$.

Expérience

On est à $1atm \approx 101 kPa$, or la valeur analogique $A = 192$ et la pression trouvée est de $97 kPa$.



```
PYTHON SHELL  
  
valeur analog A=191.9  
press analog=96.8
```

- **En utilisant la fonction de transfert et les données réels :**

La fonction du transfert fourni par le constructeur pour une plage de **0 à 700 kPa** est :

Transfert Function :

$$V_{out} = V_s \times (0,0012858 \times P + 0,04) \pm Error$$

$$V_s = 5,0 Vdc$$

Temperature = 0 to 85°C

Donc $P = \frac{V_{out} \times 777,73}{V_s} - 31,11 kPa.$



Exemple :

Cette fois-ci, on mesurera la pression atmosphérique :

$$V_{out} = 0,71 V$$



$$V_s = 3,91 V$$



En appliquant l'expression ci-dessus, la pression trouvée est $P = 110 \text{ kPa}$.

Remarque

Pour avoir des pressions faibles en dessous de P_{atm} , ($P < 101 \text{ kPa}$), on peut soit :

- Aspirer la seringue (tirer vers l'arrière).
- Placer l'ensemble dans une cloche sous vide.



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



Référentiel, Compétences

Lycée :

- Réaliser une mesure de pH en s'aidant d'une notice.
- Tracer une courbe d'étalonnage.

Lycée Professionnel :

- Reconnaître expérimentalement le caractère acide, basique ou neutre d'une solution.
- Mesurer un pH.

Compétences :

- **S'approprier** : Énoncer une problématique.
- **Analyser Raisonner** : Choisir, justifier, élaborer un protocole.
- **Réaliser** : Mettre en œuvre les étapes d'une démarche.
- **Valider** : Proposer d'éventuelles améliorations d'une démarche.
- **Communiquer** : Expliquer une démarche.

Situation déclenchante

Le pH ou **potentiel hydrogène** désigne l'une des caractéristiques principales d'une solution aqueuse : acide, neutre ou basique. Le pH est un paramètre très important en cosmétique. Pour laver un cheveu normal et sain il est recommandé d'utiliser un pH acide aux environs de 5.

Les cheveux gras qui souffrent d'une hyper-sécrétion de sébum se caractérisent d'une acidité importante.

Son médecin lui a conseillé un shampoing neutre c'est-à-dire avec un pH de 7.

La carte micro:bit peut-elle nous servir pour mesurer le pH d'un shampoing ?

Problématique



Qu'est-ce que le pH et comment le mesurer ?
Comment étalonner un capteur de pH ?
Comment peut-on mesurer le pH d'un shampoing avec la carte BBC micro:bit et un capteur de pH ?



Mesurer le pH d'une solution

Matériel nécessaire

- Une calculatrice TI-83 Premium CE Edition Python.
- Une carte BBC micro:bit.
- Un câble miniUSB-microUSB.
- Une alimentation de 5V.
- Un capteur de pH (SEN0161).
- Solutions tampons de pH=4 et de pH= 9.
- Eau distillée.
- Shampoing neutre.

pH-métrie

En 1909, le chimiste Lauritz Sørensen introduit la notion du potentiel hydrogène qui mesure l'activité chimique des ions hydrogène (protons) en utilisant des électrodes.

Plus souvent, le pH mesure l'acidité ou la basicité d'une solution aqueuse. Ainsi, dans un milieu aqueux à 25 °C :

- une solution de pH = 7 est dite neutre ;
- une solution de pH < 7 est dite acide ; plus son pH diminue, plus elle est acide ;
- une solution de pH > 7 est dite basique ; plus son pH augmente, plus elle est basique.

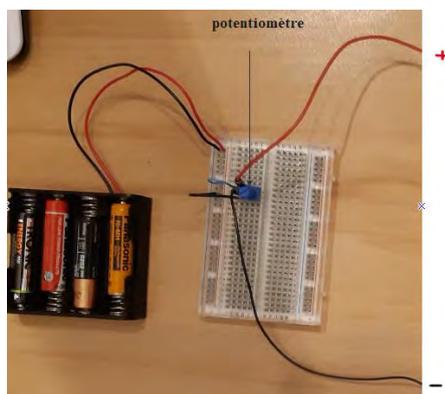
Pour mesurer le pH, on utilisera un capteur de pH (SEN0161). Il existe plusieurs variétés de capteurs pour la même utilisation.

Avant son utilisation, il faut l'étalonner pour qu'il renvoie la valeur attendue de pH de la solution.

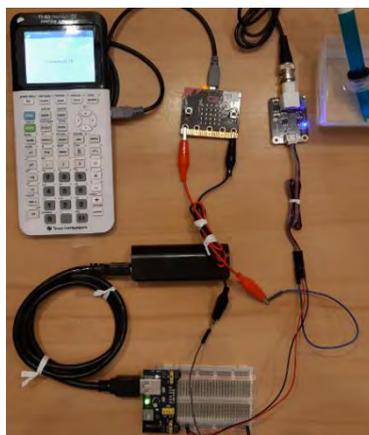
Etalonnage du capteur

Alimentation du capteur

Ce capteur doit être alimenté avec 5V. Pour cela, vous pouvez utiliser une alimentation de ce type :



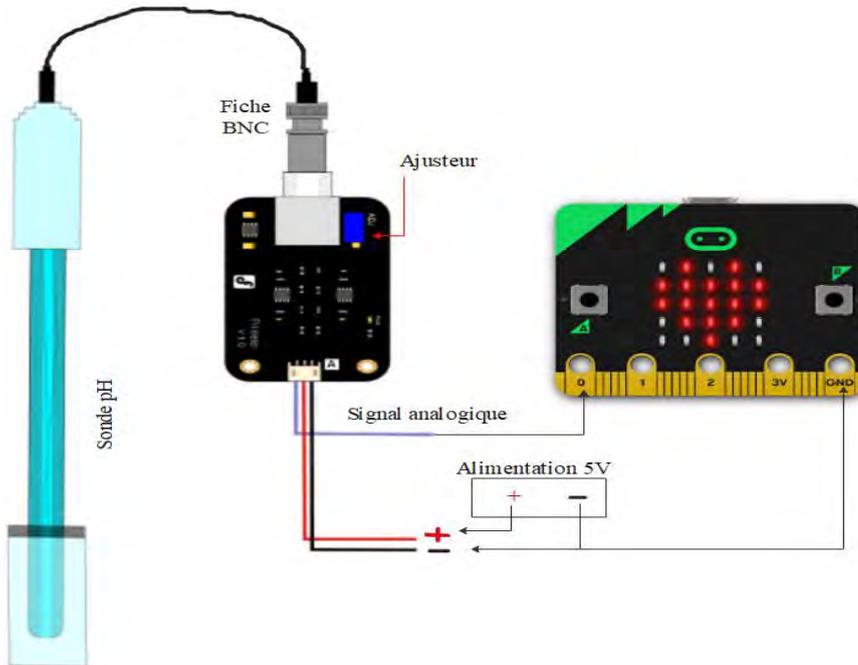
ou



Mesurer le pH d'une solution

Plus la tension (5V) est précise, plus la précision des mesures est élevée.

Réaliser le montage suivant :



Script pour le calibrage

Calibrer le pH-mètre revient à établir une relation « linéaire » entre la valeur de pH et la tension mesurée (issue d'une valeur analogique).

Le script permet d'afficher la tension mesurée entre la broche GND et la broche (pin0).

Chargement des modules nécessaires : `ti_plotlib` (représentation graphique), `ti_system` (export des données), `microbit` (gestion de la carte), `mb_pins` (gestion des ports) et `mb_disp` (affichage).

- Création d'une fonction `calibr`.
- Lecture de la valeur analogique et affectation à la variable `a`.
- Conversion de la valeur analogique `a` en tension et affectation à la variable `u`.

```

ÉDITEUR : CALIBRAG
LIGNE DU SCRIPT 0001
import ti_plotlib as plt
from microbit import *
from ti_system import *
from mb_pins import *
from mb_disp import *
display.clear()
def calibr():
    while not escape():
        a=pin0.read_analog()
        u=a*(5/1024)
        msg="tension = %.2f"%u
        plt.cls()
        plt.text_at(7,msg,"center")
        sleep(2000)

```

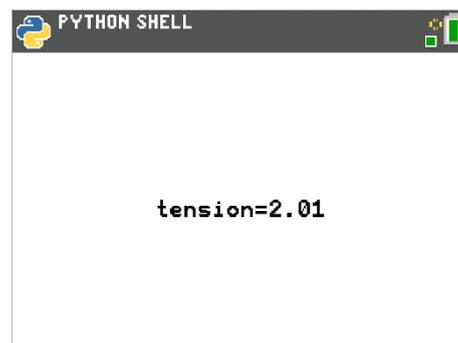
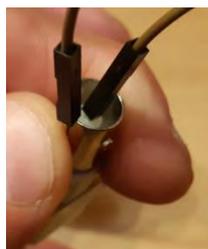
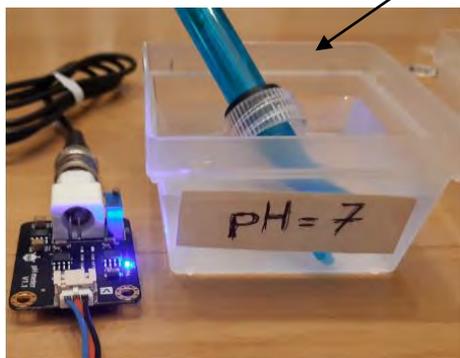
Remarque : Mesurer un pH revient à mesurer une tension.

Mesurer le pH d'une solution

Calibrage de la sonde

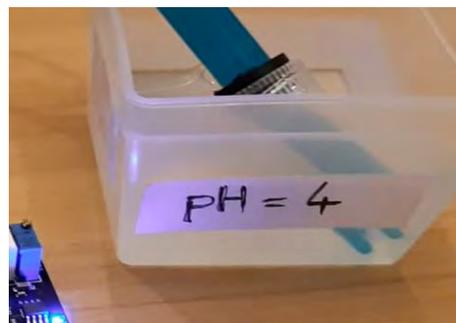
Il se fait en trois étapes à 25 °C :

Tremper la sonde dans une solution de pH=7 ou court-circuiter la fiche BNC pour avoir une tension voisine de 2V.



On a la possibilité d'ajuster cette valeur en utilisant l'ajusteur (bouton bleu sur la sonde).

On refait la même opération avec 2 solutions tampons pH =9 et pH=4, après avoir rincer la sonde. On obtient alors :



Mesurer le pH d'une solution

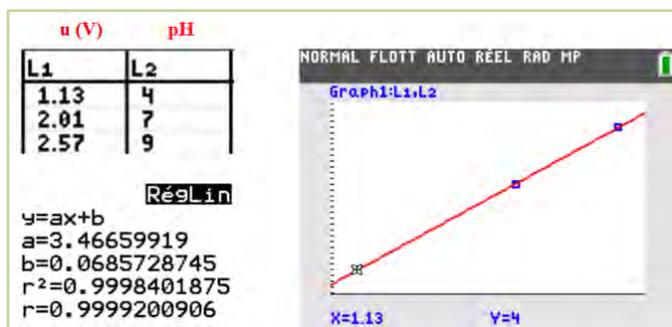
Pour établir la relation de pH en fonction de la tension, on peut utiliser

- Soit les fonctionnalités du module statistique de la calculatrice TI-83 Premium CE :

Finalement la relation proposée est :

$$pH = 3,47 \times u + 0,07$$

Remarquons que r^2 est proche de 1, ce qui montre que les trois points sont quasi-alignés.



- Soit écrire un script qui contient une fonction qui aura pour but de calculer l'équation de la droite de régression :

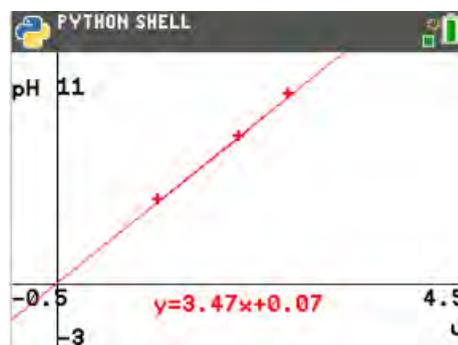
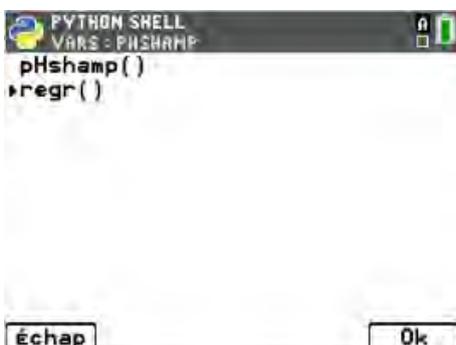
Chargement des modules nécessaires : `tiplotlib` (représentation graphique), `ti_system` (export des données)
 Création d'une fonction `regr`.

- Enregistrer les données du calibrage dans deux listes `u` et `pH`.
- Préparer la représentation graphique :
 - Effacer l'écran : `plt.cls()`.
 - Régler les paramètres de la fenêtre graphique : `plt.window(xmin, xmax, ymin, ymax)`.
 - Afficher les axes : `plt.axes()`.
 - Afficher la régression linéaire à partir des listes : `plt.lin_reg`.
 - Mettre en pause représentation graphique : `plt.show_plot()`.

```

ÉDITEUR : PHSAMP
LIGNE DU SCRIPT 0001
import tiplotlib as plt
from microbit import *
from ti_system import *
from mb_pins import *
from mb_disp import *

#Regression lineaire
def regr():
    u=[1.13,2.01,2.57]
    pH=[4,7,9]
    plt.cls()
    plt.window(-0.5,4.5,-3,11)
    plt.color(0,0,0)
    plt.axes("on")
    plt.labels("u","pH")
    plt.color(255,0,0)
    plt.plot(u,pH,"+")
    plt.lin_reg(u,pH,"center",11)
    plt.show_plot()
    
```



Application

pH d'un shampoing

- Réaliser le même montage que le précédent en prenant comme solution le shampoing neutre :
- Pour mesurer le pH de cette solution, il suffit de mesurer la tension.
 - Le script global comporte deux fonctions :
 - La fonction `regr` qui nous fournit `a` et `b`.
 - La fonction `pHshamp` qui mesure la tension `u` et par conséquent nous affichera la tension `u` et le `pH` de la solution quand on saisit les arguments `a` et `b`.



Coefficients de la droite de régression
obtenus lors du calibrage

$$\text{pH} = a \times u + b$$

tension

Chargement des modules nécessaires.

- Création d'une fonction `pHshamp`.
- Boucle **Tant que** (la touche **annul** n'est pas pressée).
 - Lecture de la valeur analogique et affectation à la variable `c`.
 - Conversion de la valeur analogique `c` en tension et affectation à la variable `u`.
 - Affectation à la variable `pH`.
 - `plt.cls()` : effacer l'écran.
 - `plt.text_at` : afficher un message et sa position.

Effectivement le shampoing choisi est au pH neutre et qui a pour but de préserver le cheveu qui souffre d'une sécrétion de sébum.

```
PYTHON SHELL

tension=1.98

pH = 6.9
```

```
ÉDITEUR : PHSAMP
LIGNE DU SCRIPT 0001
import ti_plotlib as plt
from microbit import *
from ti_system import *
from mb_pins import *
from mb_disp import *

#regression lineaire
def regr():
    u=[1.13,2.01,2.57]
    pH=[4,7,9]
    plt.cls()
    plt.window(-0.5,4.5,-3,11)
    plt.color(0,0,0)
    plt.axes("on")
    plt.labels("u","pH")
    plt.color(255,0,0)
    plt.plot(u,pH,"+")
    plt.lin_reg(u,pH,"center",11)
    plt.show_plot()

#mesure de pH
def pHshamp(a,b):
    while not escape():
        c=pin0.read_analog()
        u=c*(5/1024)
        pH=a*u+b
        msg1="tension=%.2f"%u
        msg2="pH = %.1f"%pH
        plt.cls()
        plt.text_at(7,msg1,"center")
        plt.text_at(9,msg2,"center")
    sleep(1000)
```

Prolongement

Réaliser un dosage

Il est possible d'effectuer un suivi de pH :

- Dilution.
- Dosage.

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/3.0/fr/>
© Texas Instruments 2020 / Photocopie autorisée



Construire un niveau numérique

Référentiel, Compétences

Lycée :

- Mesurer une grandeur physique à l'aide d'un capteur.
- Utiliser un dispositif comportant un microcontrôleur.
- Variabilité de la mesure d'une grandeur physique.

Lycée Professionnel :

- Echelles caractéristiques d'un système.
- Etudier expérimentalement le basculement d'un solide posé sur un plan.
- Calculs trigonométriques (lien avec les mathématiques)

Compétences :

- **S'approprier** : Représenter une situation par un schéma.
- **Analyser Raisonner** : Procéder à des analogies.
- **Réaliser** : Mettre en œuvre les étapes d'une démarche.
- **Valider** : Proposer d'éventuelles améliorations d'une démarche.
- **Communiquer** : Présenter une démarche de manière argumentée. Echanger entre pairs.

Situation déclenchante

Comment peut-on s'assurer de l'horizontalité ou de la verticalité d'un tableau, d'une conduite de câbles électriques ?

Nous allons voir dans cette activité, qu'avec l'aide d'une carte BBC micro:bit, on peut aisément construire un niveau, qui permettra de déterminer la verticale et l'horizontale d'un lieu.



Problématique

Comment fonctionne l'accéléromètre de la carte BBC micro:bit ?

Comment utiliser la carte BBC micro:bit afin de fabriquer un niveau à bulle numérique ?

Construire un niveau numérique

L'accéléromètre de la carte BBC micro:bit

La carte BBC micro:bit est équipée d'un circuit accéléromètre MMA8652 produit par la société NXP/Freescale. Il permet d'obtenir une mesure de l'accélération suivant les trois axes X, Y, Z, positionnés comme sur la figure suivante :

La connaissance de ces trois composantes de l'accélération permet de calculer l'angle d'inclinaison du dispositif.

Attention, le sens de la carte à une importance. Respecter cette configuration.



Fonctionnement du capteur

Chaque capteur fournit une information sur l'accélération enregistrée sur l'axe donné. Pour simplifier, l'accélération est une information sur la variation de la vitesse.

Les valeurs sont accessibles dans le code en utilisant les fonctions Python suivantes :

acceleration.get_x() : pour obtenir la valeur de l'accélération suivant l'axe X.

acceleration.get_y() : pour obtenir la valeur de l'accélération suivant l'axe Y.

acceleration.get_z() : pour obtenir la valeur de l'accélération suivant l'axe Z.

acceleration.get_values() : pour obtenir les 3 valeurs dans un tuple (**accX** , **accY**, **accZ**)

La valeur renvoyée par ces 3 fonctions est l'accélération en mg (milli-g), sachant que :

$$1g = 1000 \text{ mg.}$$

Le '*g*' est l'unité d'accélération utilisée dans l'industrie de l'aéronautique, ou l'automobile par exemple. En physique l'unité du système international pour l'accélération est le mètre par seconde carré ($m.s^{-2}$).

On a la correspondance suivante :

$$1g = 9,81 \text{ m.s}^{-2}, \text{ il correspond à l'accélération de la pesanteur utilisée dans la célèbre formule}$$

$$P = m.g !$$

Matériel nécessaire

- Une carte BBC microbit.
- Un fil à plomb (éventuellement) ou un niveau à bulle pour contrôler.
- Une calculatrice graphique TI-83 Premium CE Edition Python.
- Un câble USB mini – USB micro.



Construire un niveau numérique

Réalisation d'un niveau électronique

On souhaite réaliser un système embarqué qui fonctionne comme un niveau. Ce type de système est particulièrement utile dans les domaines du bâtiment, ou encore de l'aménagement intérieur.

Rechercher l'inclinaison revient à mesurer l'angle que fait un plan du système avec l'axe vertical.

Un élément de masse m , exprimée en kilogrammes, subit une force d'attraction vers la Terre. Cette force est donnée par la relation : $P = m \times g$.

Lorsqu'elle est statique, la carte micro:bit mesure l'accélération du champ de pesanteur, soit $1g$. En effet, l'accélération est la même quelle que soit la masse de l'élément qui subit la force de gravitation. Cette mesure est donnée en millième de g .

Si on positionne un des axes de l'accéléromètre vers le bas, nous devons trouver $1000 \text{ mg} = 1g$. C'est-à-dire le champ de pesanteur terrestre classique.

On vérifiera qu'en positionnant la carte micro:bit à plat et que l'on affiche l'accélération suivant l'axe Z, on obtient la valeur $z = -1000$.

Cette valeur est négative car l'axe Z est orienté vers le haut, alors que le champ de pesanteur est dirigé quant à lui vers le bas (centre de la Terre).



Remarque

Il est possible de tester les trois axes et de vérifier ainsi que l'on peut mesurer le champ de pesanteur en positionnant correctement l'axe de l'un des trois capteurs verticalement.

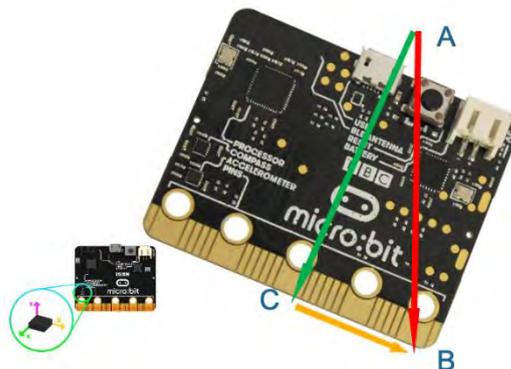
Mise en œuvre

Le triangle ABC est rectangle en C . L'angle α est donc défini par les deux mesures de l'accélération sur les axes X et Y.

$$\tan(\alpha) = \frac{\text{accélération sur l'axe Y}}{\text{accélération sur l'axe X}}$$

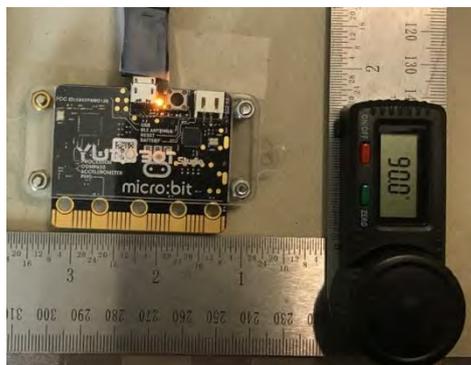
L'algorithme du script est le suivant :

```
Acc_x ← capteur accélération axe X
Acc_y ← capteur accélération axe Y
Inclinaison ←  $\tan^{-1} \alpha$ 
Afficher inclinaison en degrés
```



Construire un niveau numérique

- Chargement des modules de la carte microbit.
- La fonction `niv` déclenche la mesure de l'accélération selon X et Y, puis renvoie le calcul de l'inclinaison.
- Affichage de l'inclinaison en degrés.



```
ÉDITEUR : NIVEAU
LIGNE DU SCRIPT 0001
from math import *
import ti_plotlib as plt
from microbit import *
from mb_sensr import *
from ti_system import *
def niv():
    acc_x=accelerometer.get_x()
    acc_y=accelerometer.get_y()
    inclinaison=int(atan2(acc_y,ac
c_x)*180/pi)
    incl=round(inclinaison,1)
    return incl
Fns... a A # Outils Exéc Script
```

```
PYTHON SHELL
>>>
>>> niv()
90
>>> |
Fns... a A # Outils Éditer Script
```

Affichage en continu de l'inclinaison.

Créer une fonction `aff`.

- Insérer une boucle `Tant que`.
- Utiliser `sleep` pour avoir un délai d'affichage (en seconde).
- Affecter la valeur de l'inclinaison à la variable `l` en utilisant la fonction `niv`.
- `plt.cls` effacer l'écran.
- Afficher le message "Inclinaison" à la ligne 7 de l'écran et au centre.

```
ÉDITEUR : NIVEAU
LIGNE DU SCRIPT 0022
def aff():
    while not escape():
        sleep(1)
        l=niv()
        plt.cls()
        msg="Inclinaison deg =%.1f"%
l
        plt.text_at(7,msg,"center")
Fns... a A # Outils Exéc Script
```

```
PYTHON SHELL
Inclinaison =90.0 deg
```

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



Référentiel, Compétences

Lycée :

- Citer des exemples de capteurs présents dans les objets de la vie quotidienne.
- Mesurer la conductivité électrique d'un milieu.

Lycée Professionnel :

- Exploiter une représentation graphique.
- Mesurer une grandeur physique à l'aide d'un capteur électrique résistif.

Compétences :

- **S'approprier** : Rechercher et organiser l'information en lien avec la problématique étudiée.
- **Analyser Raisonner** : Formuler des hypothèses.
- **Réaliser** : Mettre en œuvre les étapes d'une démarche.
- **Valider** : Proposer d'éventuelles améliorations d'une démarche.
- **Communiquer** : Expliquer une démarche.

Situation déclenchante

L'air qui nous entoure contient de l'eau sous forme de vapeur. Le taux d'humidité est la mesure de la quantité de vapeur d'eau. Elle joue un rôle dans le cycle de l'eau, peut affecter notre santé physique et mentale, et peut aussi provoquer la propagation et la multiplication des micro-organismes (bactéries, moisissures...).

Le taux d'humidité est un paramètre important à prendre en compte dans la fabrication, production, stockage des aliments, médicaments. Le taux d'humidité est un indice de confort dans nos habitations. Par exemple, **contrôler** ce taux dans une SSI (Station Spatiale Internationale) représente un grand défi technologique pour les astronautes qui y habitent. Les services météorologiques **mesurent** quotidiennement l'humidité de l'air.



La carte BBC micro:bit peut-elle nous aider à mesurer de l'humidité d'une plante, et de l'air d'une habitation ?

Problématique

Comment peut-t-on mesurer l'humidité à l'aide d'une carte BBC micro:bit ?
Quels sont les paramètres physico-chimiques qui peuvent **influencer** le taux d'humidité ?
Comment peut-t-on **contrôler** l'humidité en utilisant des capteurs appropriés ?

Matériel nécessaire

- Une calculatrice TI-83 Premium CE Edition Python.
- Une carte BBC micro:bit et un câble miniUSB-microUSB.
- Câbles avec pinces crocodiles.
- Du terreau.
- Une carte d'extension (Shield grove).
- Un capteur d'humidité.
- Un capteur de température et humidité (sht35).

Mesurer le taux d'humidité d'un terreau

La conductance électrique G traduit la capacité d'un milieu à conduire le courant électrique. La résistance électrique est reliée à la conductance par la relation suivante : $R = \frac{1}{G}$.

Le terreau de nos plantes possède une conductance électrique qui dépend de la quantité d'eau et des éléments qu'il contient.

Cette conductance dépend du taux d'humidité du terreau.

Les qualités d'un bon terreau :

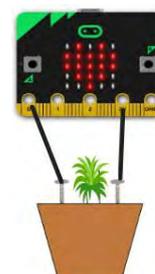
Sa conductivité électrique doit être comprise entre 45 et 55 mS/m (millisiemens par mètre). Elle reflète en partie la présence d'engrais minéraux.

Protocole n°1

Pour mettre en évidence cette dépendance, on réalise le montage ci-contre. Relions la carte à deux clous plantés dans le terreau via des câbles avec pinces crocodiles.

La résolution de la carte est de 10 bits, cela signifie que la tension d'entrée entre 0 et 3.3 V est convertie en une valeur numérique entière comprise entre 0 et 1023.

Plus il y a d'eau, plus le taux d'humidité du terreau est élevé et on aura un maximum de courant électrique, et le résultat mesuré sera proche de 1023.



Mesurer le taux d'humidité

Chargement des modules nécessaires :

- `microbit` pour le fonctionnement de la carte.
- `ti_system`.
- `mb_pins` : connexion aux ports P0 à P2.
- `mb_plotlib` pour les représentations graphiques.
- `mb_disp` affichage des DEL de la matrice.
- Création d'une fonction `humid1` qui prend comme paramètre `s` le seuil.
- Boucle Tant que (la touche `annul` n'est pas pressée)
 - `pin0.read_analog()` : lecture de la valeur sur cette broche et affectation à la variable `h`.
 - Affichage de la valeur lue avec une décimale.
 - `display.show` : affichage d'une image selon la valeur seuil de `h`.
 - `display.clear()` effacement de la matrice de DEL.

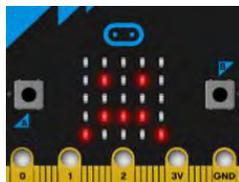
```
ÉDITEUR : HUMID1
LIGNE DU SCRIPT 0004
import ti_plotlib as plt
from microbit import *
from ti_system import *
from mb_pins import *
from mb_disp import *
display.clear()
def humid1(s):
    while not escape():
        h=pin0.read_analog()
        msg="humidite=%.1f"%h
        plt.cls()
        plt.text_at(7,msg,"center")
        sleep(5)
        if h>s:
            display.show("Image.HAPPY",delay=6000,wait=True)
            display.clear()
        else:
            display.show("Image.SAD",delay=6000,wait=True)
            display.clear()
```

```
PYTHON SHELL
humidite=882.0
```

On refait la même expérience, cette fois-ci pour une terre sèche, afin d'observer la différence de valeur affichée.

Une plante a besoin d'un taux d'humidité plus élevée (70 à 90 % dans leur milieu d'origine). Dans le script ci-dessus, nous avons fixé un seuil de 800 pour juger un bon ou un mauvais taux d'humidité du terreau.

```
PYTHON SHELL
humidite=262.0
```



Mesurer le taux d'humidité

Protocole n°2 (utilisation d'un capteur grove)

Toutes les plantes ne demandent pas les mêmes conditions d'humidité. Par exemple, une humidité de 30% à 60% convient parfaitement aux plantes **xérophytes**, alors que ce taux est largement insuffisant pour les **épiphytes**.

On se propose de tester des terreaux destinés à accueillir des plantes. On refait la même expérience en utilisant un capteur d'humidité compatible Grove.

- Connecter le capteur à la carte Sheild.
- Il délivre une valeur analogique en fonction de la teneur en eau. Il est basé sur des mesures de résistivité électrique.
- A pour avantage la reproductibilité des mesures puisque l'écartement entre les deux électrodes est fixe, alors qu'avec la première expérience, cet écartement dépend de la position des clous.



Remarque : La partie supérieure du capteur ne doit pas être en contact avec l'eau.

Chargement des modules nécessaires :

- `microbit` pour le fonctionnement de la carte.
- `ti_system`.
- `ti_plotlib` pour les représentations graphiques.
- `mb_pins` connexion aux ports P0 à P2.
- `mb_grove` pour la gestion des capteurs grove .
- Création d'une fonction `humid1(s)`.
- Boucle **Tant que** (la touche **annul** n'est pas pressée)
 - Lecture de la valeur sur le `pin0` et affectation à la variable `h`.
 - `plt.cls()` effacement de l'écran de la calculatrice.
 - Affichage de la valeur lue avec une décimale.
 - Impression d'un message selon la valeur seuil `s`.

```
ÉDITEUR : HUMID2
LIGNE DU SCRIPT 0001
import ti_plotlib as plt
from ti_system import *
from microbit import *
from mb_grove import *
from mb_pins import *
def humid2(s):
    while not escape():
        h=grove.read_moisture(pin0)
        msg="humidite=%.1f"%h
        plt.cls()
        plt.text_at(7,msg,"center")
        plt.cls()
        plt.text_at(7,msg,"center")
        if h>s:
            print("Bon terreau")
        else:
            print("Mauvais terreau")
Fns... | a A # |Outils| Exéc |Script
```

```
PYTHON SHELL
Mauvais terreau

humidite=52.5
```

On obtient un taux d'humidité de 52.5% inférieur au taux conseillé de 60% pour notre plante.

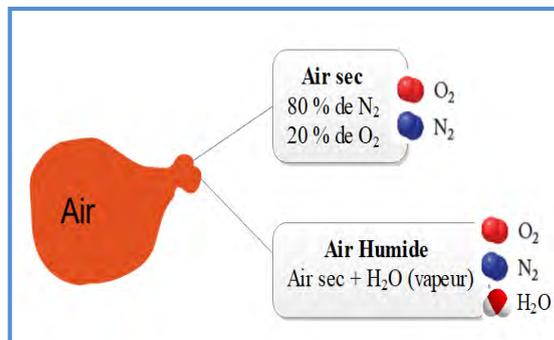
Mesurer le taux d'humidité

Mesurer le taux d'humidité de l'air

Composition de l'air

La figure ci-contre illustre la composition de l'air sec et de l'air humide. Comme exemple, nous considérons 1 kg d'air sec à 20°C auquel on rajoutera de la vapeur d'eau.

On s'intéressera à la pression et à la masse de l'air sec et de l'air humide.



	Pression	Masse
Air sec	$P_{\text{air-sec}} = 1\text{bar}$	$m_{\text{air-sec}}=1\text{kg}$
Air Humide	$P_{\text{air-humide}}=P_{\text{air-sec}} + P_{\text{vapeur}}$	$m_{\text{air-humide}}=m_{\text{air-sec}} + m_{\text{vapeur}}$

La quantité maximale de vapeur d'eau que peut absorber 1 kg d'air sec est 13,5 g (saturation) et qui correspond à une pression de vapeur de 24 mbar :

Pour un air saturé : $P_{\text{air-humide}} = 1 + 0,024 = 1,024\text{ bar}$ et $m_{\text{air-humide}}=1013,5\text{ g}$.

Humidité spécifique et humidité relative (HR)

Exemple : Si on injecte 5 grammes de vapeur d'eau dans 1 kg d'air sec, on dit que l'air possède une humidité spécifique de 5 g/ kg d'air sec.

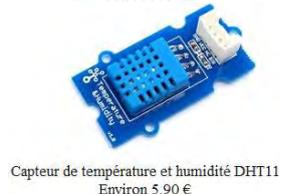
On définit l'humidité relative (HR) par la quantité d'eau dans l'air par rapport à la saturation.

Pour le même exemple $HR = 5/13,5 \times 100 = 37\%$ (humidité sur une station météo).

Ce capteur compatible Grove mesure l'humidité relative (HR) et la température de l'air en sortie digitale.

- Tension de fonctionnement : 3.3V / 5V.
- Température : -40°C à 125°C (±0.1°C).
- Humidité : 0 à 100% HR (±5%).

Il existe d'autres capteurs grove qui remplissent les mêmes fonctions mais qui ont l'inconvénient de nécessiter une alimentation de 5V.



Mesurer le taux d'humidité

Mesure de l'humidité relative (HR) dans une habitation

Réaliser ce montage en connectant le capteur SHT 35 au port I2C pour lire les deux grandeurs à la fois (température et taux d'humidité).

Le script proposé est le suivant :

Chargement des modules nécessaires :

- `microbit` pour le fonctionnement de la carte.
- `ti_system`.
- `ti_plotlib` pour les représentations graphiques.
- `mb_pins` connexion aux ports.
- `mb_grove` pour la gestion des capteurs grove .
- `mb_disp` pour l'affichage.

- Boucle **Tant que** (la touche **annul** n'est pas pressée)
 - Lecture de la valeur de l'humidité et de la valeur de température et affectation respectivement à la variable `h` et à la variable `T`.
 - Affichage de la valeur lue avec une décimale.

La valeur trouvée est correcte au niveau d'indice de confort d'une habitation puisque de nombreuses études scientifiques ont prouvé que le taux idéal d'humidité relative de l'air dans une habitation doit être compris entre 40% et 60% pour une température de 20°C.

Ensuite on confronte ce montage à des situations de sous pression et à un ajout de vapeur d'eau pour observer l'évolution de l'humidité.

- Le taux d'humidité baisse quand on introduit le dispositif dans une cloche sous vide.



```
EDITEUR : HUMID3
LIGNE DU SCRIPT 0001
import ti_plotlib as plt
from ti_system import *
from mb_disp import *
from microbit import *
from mb_grove import *
from mb_pins import *
display.clear()
while not escape():
    t,h=grove.read_sht35()
    msg1="HUMIDITE relative=%.1f"%
        h+" %"
    msg2="TEMPERATURE=%.1f"%t+" °C
    plt.text_at(7,msg1,"left")
    plt.text_at(9,msg2,"left")
    sleep(2000)
```



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de HUMID3
>>> from HUMID3 import *

HUMIDITE relative=54.6 %
TEMPERATURE=26.1 °C
```



Application

Suivi du taux d'humidité relative de l'air (ajout de vapeur d'eau)

- On reprend le montage précédent, on exécute le script afin de relever l'humidité de l'air (par exemple : 20 secondes), ensuite on plonge le tout dans une étuve saturée de vapeur d'eau pendant (40 secondes).
- Durée de l'expérience : 60 secondes.



Avant l'ajout



Après l'ajout

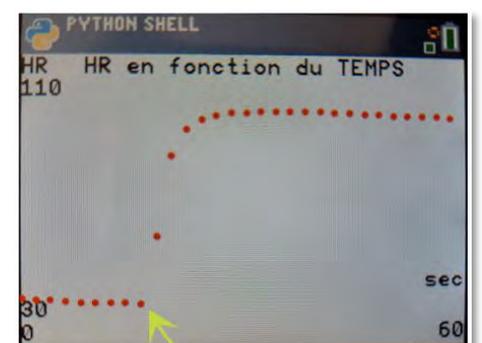
Chargement des modules nécessaires : `ti_plotlib` (représentation graphique), `ti_system` (export des données), `microbit` (gestion de la carte), et `mb_grove` (capteurs grove)

Création d'une fonction `humid_temp` : d' argument `n` qui influencera la suite de l'expérience (voir explication supplémentaire ci-dessous).

- `plt.cls()` : effacer l'écran.
- `plt.window` : réglage de la fenêtre graphique.
- `plt.axes` : affichage des axes.
- `plt.labels` : étiquettes sur les axes.
- `plt.title` : Titre du graphique.
- `plt.color` : couleur rouge pour la représentation des données.
- Boucle `for` ; acquisition des données (à modifier selon les conditions expérimentales).
- Lecture de la valeur de la pression et température affectation respectivement à la variable `p` et à la variable `T`.
- `plt.plot` : Affichage des points expérimentaux.
- `plt.scatter` : motif de représentation.
- `plt.show_plot()` : Met en pause l'affichage.

```

ÉDITEUR : HUMIDAIR
LIGNE DU SCRIPT 0001
import ti_plotlib as plt
from ti_system import *
from microbit import *
from mb_grove import *
def humid_temp(n):
    plt.cls()
    plt.window(0,2*n,30,110)
    plt.axes("on")
    plt.labels("sec", "HR",10,1)
    plt.title("HR en fonction du T
    EMPS")
    plt.color(255,0,0)
    for i in range(n):
        t=i*2
        T,h=grove.read_sht35()
        sleep(500)
        plt.plot(t,h,"o")
    plt.show_plot()
    
```



Introduction de la vapeur d'eau



Détails du script :

- $n=30$ donc $x_{max}= 2*30 =60$
- i allant de 0 à 29 et on aura 60 points (mesures) espacées de 0.5 secondes, donc la durée de l'expérience est de 30 secondes

```
def humid_temp(n):  
    plt.window(0,2*n,30,110)  
    for i in range(n):  
        t=i*2  
        T,h=grove.read_sht35()  
        sleep(500)
```

Cette expérience montre bien que l'humidité est liée directement à la présence de la vapeur d'eau dans l'air sec. On assiste à une brutale augmentation du taux d'humidité relative instantanément après introduction de la vapeur d'eau.

On remarque que l'air dans le vase ne peut plus absorber la vapeur d'eau, il s'agit de la saturation. Ensuite c'est la condensation qui s'installe.

Prolongement

Calcul de la masse de vapeur d'eau

Exercice : Calculer la masse de vapeur d'eau introduite qui fait passer le taux d'humidité relative de 40 % à 90%, dans un vase cylindrique de diamètre 12 cm et de hauteur 21 cm.

Aide : On définit l'humidité relative (HR) par la quantité d'eau dans l'air par rapport à la saturation.

Pour le même exemple $HR = 5/13,5 \times 100 = 37\%$.

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

Se protéger de la chaleur

Référentiel, compétences

Lycée :

- Mesurer une grandeur physique à l'aide d'un capteur.
- Utiliser un dispositif comportant un microcontrôleur.
- Prévoir le sens d'un transfert thermique.
- Caractériser qualitativement les trois modes de transfert thermique : conduction, convection, rayonnement.

Lycée Professionnel :

- Mettre en évidence les modes de transfert thermique.
- Exploiter une représentation graphique.
- Comparer expérimentalement de façon qualitative les propriétés de plusieurs matériaux vis-à-vis de la conduction thermique.

Compétences :

- **S'approprier** : Rechercher et organiser l'information en lien avec une problématique.
- **Analyser Raisonner** : Planifier des tâches, évaluer des ordres de grandeur.
- **Réaliser** : Mettre en œuvre un protocole expérimental en respectant des règles de sécurité.
- **Valider** : Procéder à des tests de vraisemblance. Identifier des sources d'erreur, estimer une incertitude.
- **Communiquer** : Echanger entre pairs.

Situation déclenchante

L'été, la température à l'intérieur d'un véhicule peut atteindre des valeurs très élevées. L'ouverture des vitres du véhicule est insuffisante. Peut-on favoriser l'abaissement de la température à l'intérieur du véhicule et à l'aide de quels dispositifs ?



Problématique

- Comment mesurer une température en utilisant le capteur de température intégré à la carte BBC micro:bit ?
- Communiquer la mesure à un autre dispositif et l'afficher.
- Représenter graphiquement deux séries de mesures et évaluer leurs écarts.

Matériel à utiliser

- Papier aluminium et Canson noir mat.
- Plaque de verre ou plexiglass ou pochette plastique ou bocal.
- 2 cartes BBC micro:bit et câbles de liaison USB mini-micro USB.
- 2 calculatrices TI-83 Premium CE™ Edition Python.
- Un câble mini USB – micro USB.

Proposition de protocole

On souhaite vérifier l'efficacité d'un pare soleil afin de diminuer la température dans l'habitacle d'un véhicule.

L'habitacle des véhicules est très souvent de couleur noire mat. Ce type de décor peut poser des problèmes de confort et de sécurité en été.

L'expérience se réalise en deux temps et utilise le capteur de température intégré à la carte micro:bit.

1. On tapisse l'intérieur d'une boîte en plastique transparente ou d'un bocal d'une feuille de Canson noir. On dépose la carte micro:bit sur cette feuille, la boîte est ensuite fermée afin de simuler l'habitacle du véhicule et on relève la température à intervalle régulier pendant 10 minutes.
2. Tout en conservant la feuille de canson noir, on dépose une feuille d'aluminium sur laquelle on place la carte micro:bit. Les mesures sont enregistrées également pendant 10 minutes.

Le script doit prévoir la possibilité de communiquer par liaison radio un avertissement à une autre carte, dès lors que la température atteint une valeur critique (30°C par exemple). Les mesures seront sauvegardées dans des listes, puis représentées graphiquement, afin de pouvoir évaluer les écarts et proposer une réponse au problème.



Remarque

Pour exporter les listes de données expérimentales on utilise l'instruction `store_list("nom", var)` disponible dans le module `ti_system`. Un tutoriel est disponible à cette adresse :

<https://education.ti.com/fr/mises-a-jour-et-logiciels/ti-codes>



Se protéger de la chaleur

On réalisera donc deux scripts à partir des algorithmes ci-dessous.

Celui de la machine émettrice.

Celui de la machine réceptrice.

```
t ← 0
Pour i variant de 1 à t par
pas de Δt
    T ← mesure de la
    température
    Si T > Tmax alors
        Envoi du message
        "danger"
    Sinon
        Envoi du message
        "calme"
    Fin Si
    t ← t + Δt
Fin Pour
```

```
Tant que (on n'appuie pas
sur la touche "annul")
    msg ← reception
    message
    Si msg = "danger"
alors
        allumer la matrice
        DEL
    Fin Si
    Si msg = "calme"
Alors
        ne pas allumer la
        matrice DEL
    Fin Si
Fin Tant que
```

Mise en œuvre

La carte BBC micro:bit intègre un capteur de température. Il n'est pas très précis car il est soudé sur la carte elle-même à proximité du processeur ; il est donc perturbé par l'échauffement provoqué par le fonctionnement de la carte. Le phénomène est encore amplifié lorsque la carte est insérée dans un étui de protection.

Selon les souhaits de réalisation, il pourra être judicieux d'utiliser un capteur de température « grove » associé à la carte micro:bit par l'intermédiaire d'une carte d'extension (shield). Le même script fonctionne après une légère adaptation et en chargeant le module grove.





Calculatrice émettrice.

Chargement des modules nécessaires :

- **microbit** pour le fonctionnement de la carte et des modules associés (sensors, radio, affichage).
- **radio.config** configure une chaîne et un groupe de communication lorsque de nombreuses cartes sont dans la même pièce.
- **radio.on** active la fonction de communication radio de la carte.

```
ÉDITEUR : TEMP1
LIGNE DU SCRIPT 0010
from microbit import *
from mb_sensr import *
from mb_radio import *
from mb_disp import *
radio.config(length=32, channel=7
            , power=6, group=0)
radio.on()
```

- Création d'une fonction **mes** qui prend comme argument un entier n et qui recueille n mesures de la température puis renvoie la température moyenne pendant l'intervalle de temps $\Delta t = 3n$.
- Création d'une boucle **for** pour stocker les mesures effectuées par la fonction **temperature** dans la liste **temper**.
- Les données sont acquises toutes les 3 secondes (**sleep(3000)**).
- Intervalle de temps $\Delta t = 3n$.

```
ÉDITEUR : TEMP1
LIGNE DU SCRIPT 0007
#acquisition des données
def mes(n):
    temper=[]
    for i in range(n+1):
        T=temperature()
        temper.append(T)
        sleep(3000)
    Tmoy=sum(temper)/len(temper)
    return round(Tmoy,2)
```



Se protéger de la chaleur

J-L Balas

- Création d'une fonction **alarme** qui prend comme arguments **c** (la température maximale T_{\max} tolérée) et la température **T** et renvoie un message si la température est supérieure à la température de consigne **c** fixée.
- Création d'une fonction **exp** qui prend comme arguments **n** le nombre de mesures et **c** la température maximale tolérée et qui permet de gérer l'expérience complète (acquisition des mesures et alarme).



```
EDITEUR : TEMP1
LIGNE DU SCRIPT 0027
#alarme
def alarme(c,T):
    if T>=c:
        radio.send("danger")
    else:
        radio.send("calme")
#Experience
def exp(n,c):
    T=mes(n)
    alarme(c,T)
```

Calculatrice réceptrice.

- Chargement des modules comme sur la machine émettrice.
- Configuration du groupe et du canal de communication radio.
- **radio.on** active la fonction de communication radio de la carte.
- Boucle **Tant que** le mode radio est actif en réception de message.
- Si le message reçu est "danger" alors on allume la matrice de DEL avec l'intensité maximale (9).
- Si le message reçu est "calme" alors la matrice de DEL reste éteinte.



```
EDITEUR : TEMP2
LIGNE DU SCRIPT 0089
from ti_system import *
from microbit import *
from mb_disp import *
from mb_radio import *
radio.config(length=32,channel=7
            ,power=6,group=0)
radio.on()
display.clear()
while True:
    msg= radio.receive()
    if msg=="danger":
        for i in range(5):
            for j in range(5):
                display.set_pixel(i,j,9)
    if msg=="calme":
        for i in range(5):
            for j in range(5):
                display.set_pixel(i,j,0)
```

Remarque

La calculatrice recevant le message doit avoir le script en fonctionnement afin de recevoir le message et la consigne correspondant à l'alarme.

Pour réaliser l'expérience en utilisant la communication radio avec deux cartes micro:bit, exécuter les scripts :

TEMP1 sur la machine émettrice (**exp(n, c)** où **n** est le nombre de mesures et **c** la température au-delà de laquelle l'alarme se déclenche).

TEMP 2 sur la machine réceptrice.



Mise en œuvre expérimentale

- Placer la carte micro:bit à température ambiante.
- Sur la feuille de Canson noir et sous la feuille d'aluminium, placer la carte micro:bit, puis la couvrir du plexiglass.
- Réaliser les mesures de température en utilisant la fonction `mes` selon que l'on souhaite utiliser une seule carte afin de mesurer la température ou `exp` si l'on désire utiliser une carte pour effectuer la mesure et une autre pour signaler une alarme.
- Remettre la carte à température ambiante et attendre quelques minutes.
- Sur la feuille de Canson noir, placer la carte puis la couvrir du plexiglass.
- Mesurer l'écart de température ΔT entre les deux expériences.
- Conclure.

Remarque

Dans la mesure du possible, il sera intéressant de réaliser cette expérience en été à l'intérieur d'un véhicule.



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de TEMP1
>>> from TEMP1 import *
>>> mes(10)
30.55
>>> mes(10)
38.91
>>> |
Fns... | a A # |Outils|Éditer|Script
```

On peut observer : La mesure avec un pare soleil est de 30.55°C. Lorsque l'on enlève la protection, la température est de 38.91°C. L'écart est donc de 8.36°C en une minute. Le capteur a été posé sur le tableau de bord.

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



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.

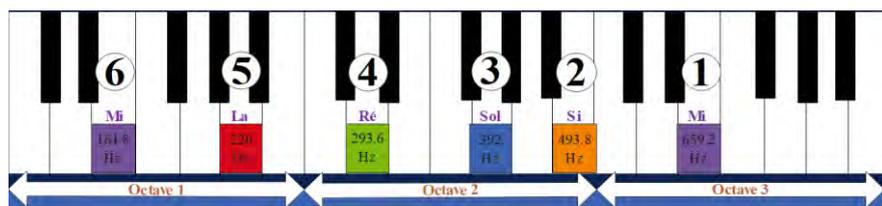
Proposition de protocole

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

```

ÉDITEUR : ACORGUIT
LIGNE DU SCRIPT 0001
from microbit import *
from mb_music import *
from mb_butns import *
def accor():
    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"

```

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.



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



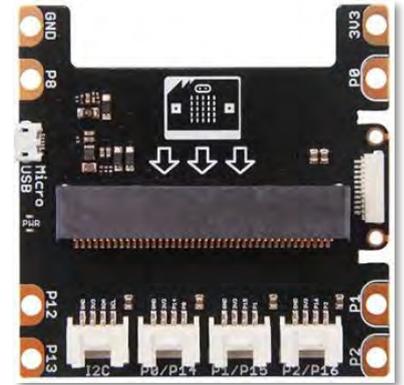
A. Yazı

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.

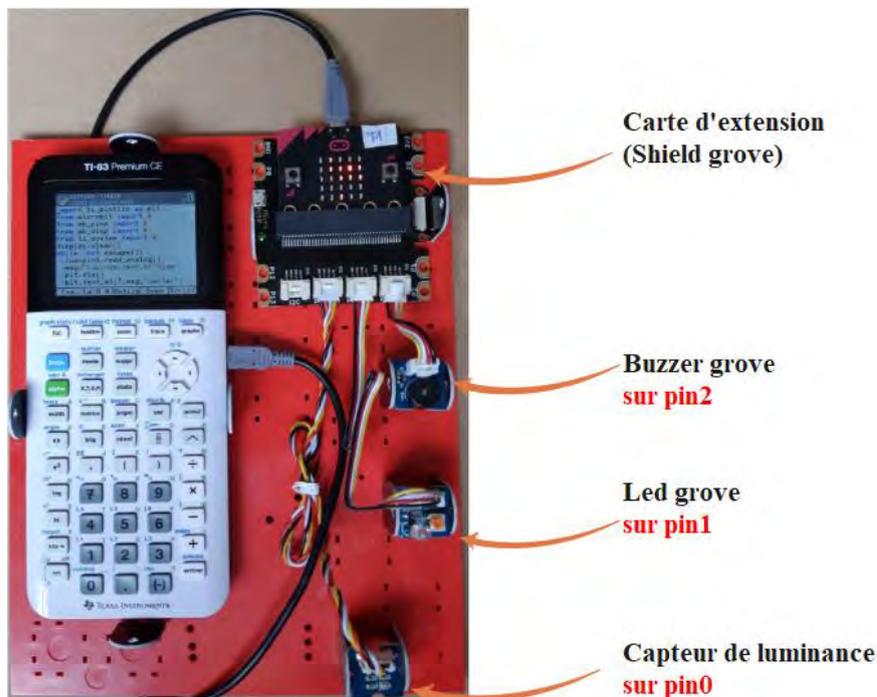
Ce type de connexion sera à privilégier si l'on souhaite se concentrer sur le fonctionnement par bloc des composants électroniques, et surtout pour une mise en œuvre rapide.



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).



Algorithme

Tant que

$l \leftarrow$ 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



Mesure de la Vitesse du son dans l'air

Référentiel, compétences

Lycée :

- Mesurer une grandeur physique à l'aide d'un capteur.
- Utiliser un dispositif comportant un microcontrôleur.
- Variabilité de la mesure d'une grandeur physique.
- Célérité d'une onde.

Lycée Professionnel :

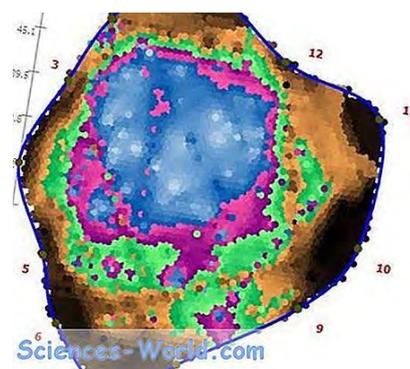
- Identifier les grandeurs d'entrée et de sortie (avec leur unité) d'un capteur.
- Mettre en œuvre un capteur.
- Mettre en œuvre une chaîne de transmission.

Compétences :

- **S'approprier** : Énoncer une problématique.
- **Analyser Raisonner** : Choisir, élaborer, justifier un protocole.
- **Réaliser** : Mettre en œuvre un protocole expérimental.
- **Valider** : Confronter un modèle à des résultats expérimentaux.
- **Communiquer** : Présenter une démarche argumentée de manière synthétique et cohérente.

Situation déclenchante

Afin d'évaluer la santé des arbres, les forestiers sont amenés à mesurer par tomographie, l'état du tronc. En effet la pourriture du bois dans les arbres vivants peut entraîner une surestimation des réserves mondiales de carbone, la perte de bois dans le secteur forestier et la mauvaise santé des arbres. Mais en quoi consiste la technique de tomographie afin d'évaluer l'état de décomposition intérieure d'un tronc de bois ?



Problématique

La tomographie par ultrasons consiste à envoyer des ondes sonores à travers les arbres. Plus il faut de temps à une onde sonore pour traverser le tronc, plus le bois est décomposé.

Comment mesurer la Vitesse du son dans l'air ?



Mesure de la vitesse du son dans l'air

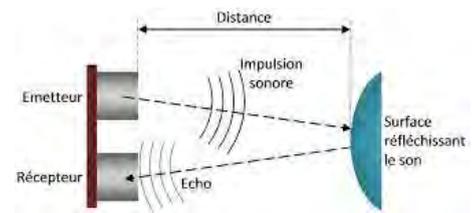
Matériel nécessaire

- Une carte BBC micro:bit.
- Une calculatrice TI-83 Premium CE Edition Python.
- Un câble mini USB-micro USB.
- Un capteur grove « Ranger ».
- Un Grove Shield pour micro:bit v2.0.



Principe de la mesure

En utilisant un capteur ultrason type HC_SR04, il est possible de chronométrer le temps qu'un ultrason met pour effectuer un trajet aller-retour entre le capteur et un obstacle.



Fonctionnement du capteur

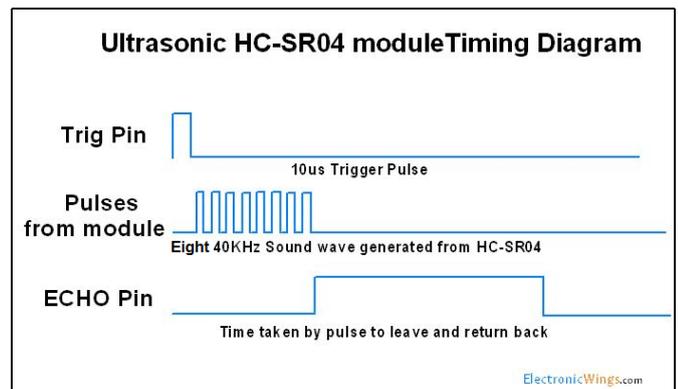
Le principe de ce capteur est le suivant : On envoie sur la broche **trig** une impulsion de 10 μ s provoquant le déclenchement de la mesure. La broche **echo** passe alors au niveau haut le temps que le son parte et revienne du capteur vers l'obstacle.

Il s'agit de connecter l'émetteur/récepteur d'ultrasons comme indiqué précédemment puis de placer un obstacle permettant la réflexion de l'onde.

On mesure le plus précisément possible la distance qui sépare le composant de l'obstacle en prenant comme référence les capsules protégées par la grille du capteur.

Cette valeur pourrait être entrée dans un programme en tant que variable, mais les instructions du module Python de la calculatrice permettent de le faire. Le programme mesure le temps mis par l'onde pour effectuer un aller-retour et renvoie la valeur en microsecondes. Il est alors possible de calculer la vitesse du son et de l'afficher.

On affiche le résultat de la mesure sur l'écran de la calculatrice, mais d'autres options sont possibles avec la carte micro:bit (utilisation des diodes).



Mesure de la vitesse du son dans l'air

Remarques

Le capteur HC_SR04 utilisé fonctionne en 5V. Si la broche **trig** accepte le 3,3V provenant de la carte, la broche **echo** renvoie 5V à la micro:bit qui ne va pas du tout apprécier, celle-ci fonctionnant à 3,3V.

Pour cette expérience, on utilise le « Grove Shield pour micro:bit v2.0 » qui propose une protection contre les surtensions et permet de protéger la carte. Si vous n'avez pas ce shield, utilisez pour la broche **echo** un pont diviseur de tension avec 2 résistances afin de ramener la tension à 3V (voir activité « Mesurer une tension »).

Le capteur doit posséder une alimentation dédiée de 5V, car la carte micro:bit ne peut fournir ce niveau de tension. Pensez toutefois à relier les masses GND de la micro:bit et de l'alimentation 5V.

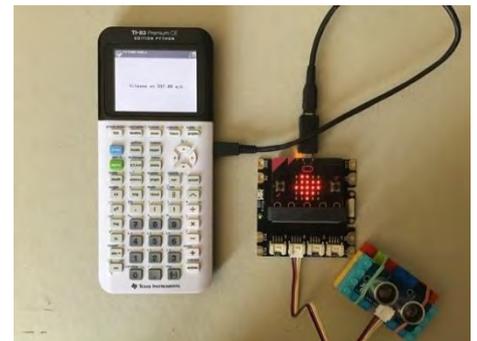
Mise en œuvre

Écrire un script Python permettant de mettre en œuvre l'algorithme ci-dessous.

$d \leftarrow$ mesurer la distance (capteur-objet)
 $echo \leftarrow$ mesurer l'intervalle de temps (Δt)

$$v \leftarrow \frac{2 \times d}{\Delta t}$$

Afficher v



- Import des modules nécessaires au fonctionnement du script.
- Connexion du capteur à la broche **pin0** et affectation de la lecture à la variable **init**.

- Création d'une fonction **vit** permettant de mesurer :
- La distance **d** en cm séparant le capteur d'un objet.
- Le temps **echo** mis par l'onde pour effectuer un aller-retour.
- Calcul de la vitesse de propagation du son dans l'air en m/s.
- Résultat arrondi à 2 décimales.

```

EDITEUR : VITSON
LIGNE DU SCRIPT 0001
from microbit import *
from mb_pins import *
from mb_grove import *
def vit():
    init=grove.read_ranger_cm(pin0
    )
    d=grove.read_ranger_cm(pin0)
    echo=grove.read_ranger_time(pi
n0)
    v=2*d/(echo*100)
    return round(v,2)

```

Fns... | a A # |Outils| Exéc |Script

Mesure de la vitesse du son dans l'air

Prolongements possibles

- Déclencher l'acquisition par appui sur le bouton **A** de la carte micro:bit et sauvegarder la mesure dans une liste en appuyant sur le bouton **B**. Effectuer ensuite une étude statistique sur la série de mesures (moyenne, écart type, diagramme de dispersion etc..).
- Utiliser le capteur de température incorporé à la carte micro:bit (ou un capteur grove (DS18B20)) afin d'observer l'influence de la température sur la mesure.
- Évaluer la précision de la mesure en effectuant plusieurs essais sur des distances courtes et plus grandes (dans les limites de fonctionnement du capteur HSCR04).
- Utiliser les DEL de la carte pour afficher le résultat d'une mesure.



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de VITSON
>>> from VITSON import #
>>> vit()
340.0
>>> |
```

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

Mesurer une distance



A. Yazı

Référentiel, Compétences

Lycée :

- Mesurer une grandeur physique à l'aide d'un capteur.
- Utiliser un dispositif comportant un microcontrôleur.

Lycée Professionnel :

- Variabilité de la mesure d'une grandeur physique.
- Vitesse de propagation.

Compétences :

- **S'approprier** : Représenter une situation par un schéma.
- **Analyser Raisonner** : Proposer une stratégie de résolution.
- **Réaliser** : Mettre en œuvre des algorithmes.
- **Valider** : Proposer d'éventuelles améliorations d'une démarche.
- **Communiquer** : Expliquer une démarche.

Situation déclenchante

La consigne de l'OMS sur les gestes barrières semble issue d'un postulat historique, en place depuis 1930 par le médecin William F. Wells.

Il estime que la charge infectieuse des gouttelettes infectées atteint avec une grande difficulté la distance d'un mètre. Lors de la pandémie du Covid19, une entreprise française (société de Sophia-Antipolis) a mis au point un bracelet au poignet qui va vous avertir et clignoter si la distance barrière n'est pas respectée.



Comment fonctionne ce dispositif ?

Peut-on reproduire le même dispositif en utilisant la carte BBC micro:bit et la calculatrice TI-83 Premium CE Edition Python, en association avec quelques capteurs ?

Problématique

Quel est le principe d'un télémètre ?
Comment mesure-t-on une distance avec la carte BBC micro:bit ?
Quelle est la nature de l'onde utilisée dans un tel dispositif ?
Comment générer une alerte sonore ou lumineuse en cas de proximité avec un individu ?



Mesurer une distance

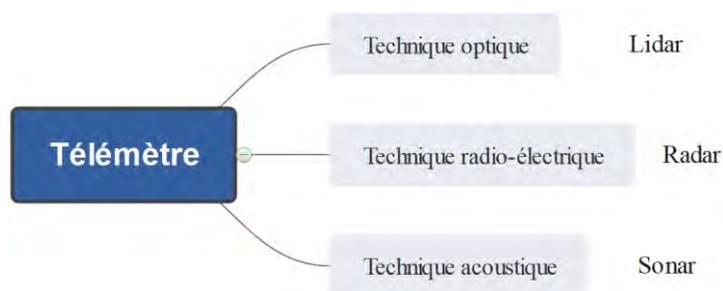
Matériel nécessaire

- Une calculatrice TI-83 Premium CE Edition Python.
- Une carte BBC micro:bit.
- Un câble microUsb-miniUsb.
- 3 capteurs Grove « Ranger ».
- Une carte d'extension (Shield Grove).
- Un buzzer Grove.

Principe d'un télémètre

Outil d'une grande simplicité mais très performant, le télémètre est l'accessoire de mesure indispensable pour chaque professionnel ayant besoin de mesurer une distance.

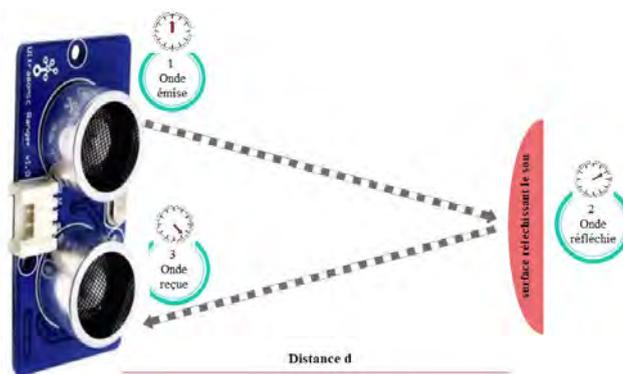
Il existe plusieurs types de télémètres selon la technique utilisée :



Notion d'ultrasons et principe d'écholocalisation

Le capteur utilisé dans cette fiche est à base d'ultrasons. Il mesure la distance le séparant de l'obstacle. Il fonctionne par émission d'une salve d'ultrasons (paquet d'onde de courte durée, émis à intervalles égaux de temps) et détection de l'écho en retour.

Le principe utilisé est celui de la propagation d'ondes ultrasonores d'une fréquence de 40 kHz dans l'air. La vitesse v du son dans l'air dépend de la température. Elle est de l'ordre de 340 m/s à 20°C. Une description plus détaillée est abordée dans la fiche « vitesse du son ».



Mesurer une distance



A. Yazı

Mesure de distance

Réaliser le montage suivant (en connectant le ranger au pin0 de la carte Shield).



Le script permet d'afficher **la distance mesurée en cm** par le capteur à ultrasons.

La disponibilité de quelques commandes du module **grove** rend la réalisation de ce script très facile.

Chargement des modules nécessaires :

- **microbit** pour le fonctionnement de la carte.
- **ti_system**
- **mb_ti_ptotlib** pour les représentations graphiques.
- **mb_grove** pour la gestion des capteurs grove.
- Création d'une fonction **dist**.
 - Boucle **Tant que** (la touche **annul** n'est pas pressée)
 - Lecture de la valeur de la distance et affectation à la variable **d**.
 - Affichage de la valeur lue avec une décimale au centre de l'écran.

```
ÉDITEUR : TELEMET1
LIGNE DU SCRIPT 0001
from ti_system import *
import ti_plotlib as plt
from microbit import *
from mb_grove import *
def dist():
    while not escape():
        d=grove.read_ranger_cm(pin0)
        msg="DISTANCE=%1f"%d+"cm"
        plt.cls()
        plt.text_at(7,msg,"center")
        sleep(2500)
Fns... | a A # | Outils | Exéc | Script

PYTHON SHELL
DISTANCE=99.0cm
```

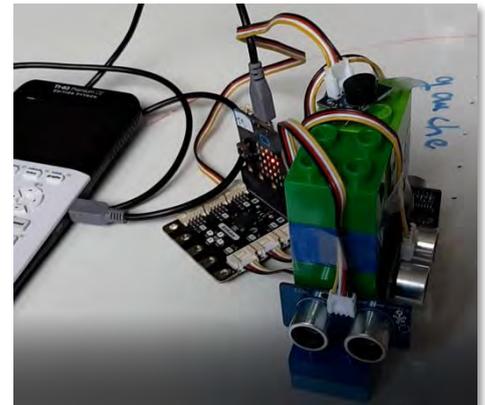
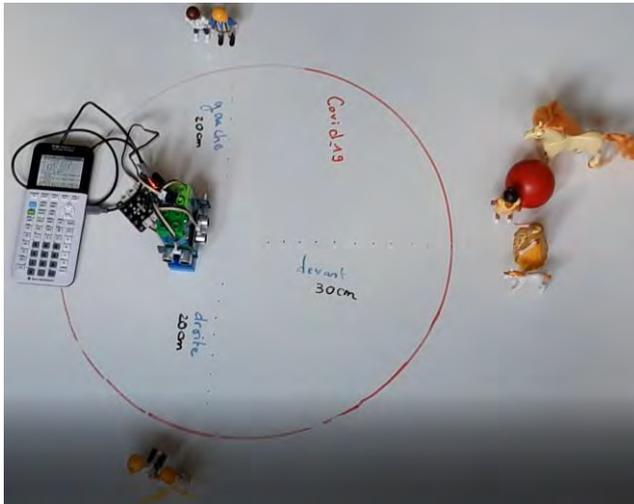
Application

Distance barrière –Covid19

Le dispositif qui simulera un bouclier virtuel pour respecter la distance barrière d'un mètre, est constitué de 3 capteurs « ranger » et d'un buzzer afin d'avertir du non-respect de la distanciation, quand la distance mesurée est inférieure à une distance « seuil » fixée à l'avance.



Mesurer une distance



Algorithme

```

valeur seuil 1 ← valeur #Seuil à choisir
valeur seuil 2 ← valeur #Seuil à choisir
valeur seuil 3 ← valeur #Seuil à choisir
Tant que
  dd ← distance lue par le ranger sur le port 0 #distance à droite
  dg ← distance lue par le ranger sur le port 1 #distance à gauche
  df ← distance lue par le ranger sur le port 2 #distance en face

  Si dd < valeur seuil 1 ou dg < valeur seuil 2 ou df < valeur seuil 3 alors
    Envoi d'un signal sonore (pin8)
  Fin si
Fin Tant que
```

Mesurer une distance



A. Yazi

Chargement des modules nécessaires :

- `microbit` pour le fonctionnement de la carte.
- `ti_plotlib` pour les représentations graphiques.
- `mb_pins` connexion aux ports.
- `mb_grove` pour gérer les capteurs grove.
- **Création de la fonction** `alarme_covid`.
- Boucle **Tant que** (la touche **annul** n'est pas pressée)
 - Lecture des 3 distances et leurs affectations à la variable `dd`, `dg` et `df`.
 - Affichage des valeurs lues avec une décimale.
 - Si l'une des 3 distances lues est inférieure à la distance seuil « de votre choix ».
 - Alors envoyer un signal sonore à partir du port 8.
 - `plt.cls()` : effacer l'écran.
 - `plt.text_at` : afficher un message et sa position.

```

ÉDITEUR : COVID19
LIGNE DU SCRIPT 0001
from ti_system import *
import ti_plotlib as plt
from microbit import *
from mb_grove import *
from mb_pins import *
#dd:droite dg:gauche df:en face
def alarme_covid(a,b,c):
    while not escape():
        dd=grove.read_ranger_cm(pin0
        )
        dg=grove.read_ranger_cm(pin1
        )
        df=grove.read_ranger_cm(pin2
        )
        if dd<a or dg<b or df<c :
            pin8.write_digital(1)
            msg1="DROITE=%.1f"%dd+"cm"
            msg2="GAUCHE=%.1f"%dg+"cm"
            msg3="FACE=%.1f"%df+"cm"
            plt.cls()
            plt.text_at(7,msg1,"center")

            plt.text_at(9,msg2,"center")

            plt.text_at(11,msg3,"center"
            )
            sleep(2500)

```

Fns... a A # |Outils| Exéc |Script
 PYTHON SHELL

```

DROITE=155.4cm
GAUCHE=167.1cm
FACE=81.4cm

```

Prolongement 1

Effet de la température

Un télémètre à ultrasons n'est pas un outil précis en le comparant à un télémètre laser.

Cette mesure a été réalisée en supposant que la vitesse du son est de $343 \text{ m} \cdot \text{s}^{-1}$.

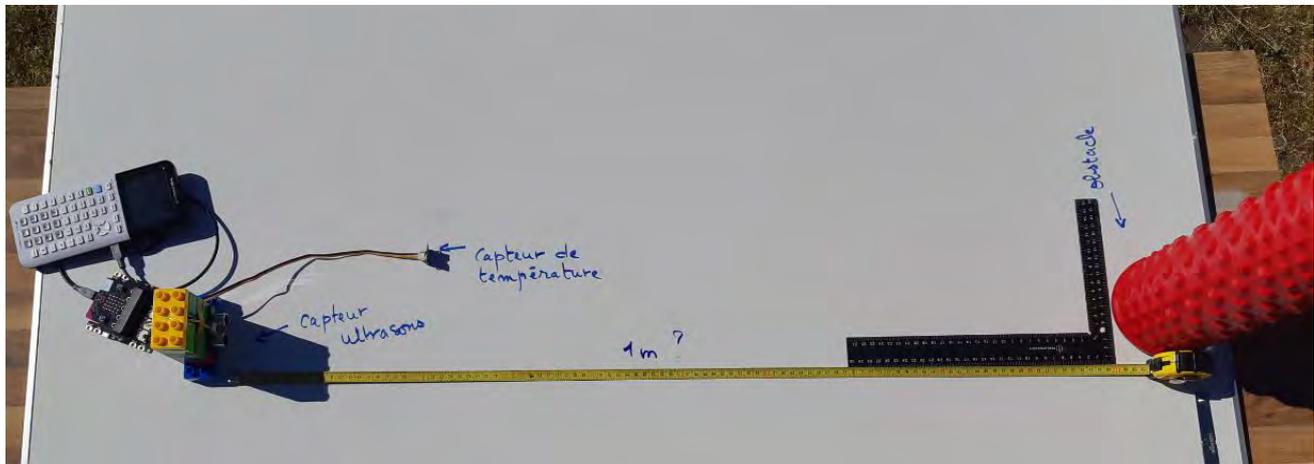
En réalité la vitesse d'une onde sonore est aussi influencée par la température du milieu traversé. La célérité a la propriété d'augmenter avec la température du milieu. Si T désigne la température du milieu, la célérité des ondes sonores dans ce milieu est approximativement égale à vingt fois la racine carrée de la température.

$$v = 20 \cdot \sqrt{T + 273,15} \quad T \text{ en } (^\circ\text{C})$$



Mesurer une distance

On refait la même expérience en utilisant cette approximation et en insérant un capteur de température au montage précédent (au pin1 par exemple) :



Cette expérience est réalisée au mois de juillet (une journée où il fait 29°C)

Remarque : L'objectif de cette expérience est de sensibiliser les élèves sur le fait que la mesure d'une grandeur physique peut être influencée par d'autres paramètres (ici la température).

La température du milieu est mesurée grâce au capteur ci-contre



Chargement des modules nécessaires :

- `microbit` pour le fonctionnement de la carte.
- `ti_system`.
- `mb_ti_plotlib` pour les représentations graphiques.
- `mb_grove` pour la gestion des capteurs grove.

• Création d'une fonction `temp`.

- Lecture de la valeur de la température et affectation à la variable `T`.
- Renvoie la valeur lue.

• Création d'une fonction `dist`.

- Boucle **Tant que** (la touche **annul** n'est pas pressée)
 - Lecture de la valeur de la distance et affectation à la variable `d`.
 - Correction de la distance `d` en tenant compte du paramètre température `T` (argument de la fonction et fourni par la fonction `temp`) et affectation à la variable `dT`.
 - Affichage des valeurs des 3 variables avec une décimale.

```

ÉDITEUR : TELEMET2
LIGNE DU SCRIPT 0001
from math import *
from ti_system import *
import ti_plotlib as plt
from microbit import *
from mb_grove import *
from mb_pins import *
#Mesure de temperature
def temp():
    T=grove.read_temperature(pin1)
    return T
#Mesure de la distance
def dist(T):
    while not escape():
        d=grove.read_ranger_cm(pin0)
        dT=d*20/342*sqrt(T+273.15)
        msg1="DISTANCE=%.1f"%d
        msg2="TEMPERAT=%.1f"%T
        msg3="DISTANCE corrig=%.1f"%dT
        plt.cls()
        plt.text_at(7,msg1,"center")
        plt.text_at(9,msg2,"center")
        plt.text_at(11,msg3,"center")
        sleep(2500)
    
```

Mesurer une distance



A. Yazici

La distance mesurée en prenant compte du paramètre température apporte la correction attendue.

$$v = 20 \cdot \sqrt{T + 273,15}$$

Aide pour le calcul de la distance corrigée :

$$d_{\text{corrigé}} = d_{\text{mesurée}} \times \frac{20 \times \sqrt{T + 273.15}}{342}$$

```
PYTHON SHELL

DISTANCE=98.4
TEMPERAT=28.5
DISTANCE corrig=99.9
```

Prolongement 2

Radars de recul

Le dispositif peut être adapté à un radar de recul, en remplaçant le buzzer avec un haut-parleur et en sollicitant le module music afin de générer différents signaux sonores en fonction du rapprochement de l'obstacle.



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



Construire une boussole

Référentiel, compétences

Lycée :

- Mesurer une grandeur physique à l'aide d'un capteur.
- Utiliser un dispositif comportant un microcontrôleur.

Lycée Professionnel :

- Identifier des sources et détecteurs d'ondes électromagnétiques dans les objets de la vie courante.
- Connaître différents dispositifs permettant de créer un champ magnétique.
- Connaître l'unité de champ magnétique dans le système international et quelques ordres de grandeur de champs magnétiques usuels.
- Représenter un vecteur (lien avec les mathématiques).

Compétences :

- **S'approprier** : Rechercher et organiser l'information en lien avec une problématique.
- **Analyser Raisonner** : Proposer une stratégie de résolution, planifier des tâches.
- **Réaliser** : Effectuer des procédures courantes.
- **Valider** : Procéder à des tests de vraisemblance.
- **Communiquer** : Echanger entre pairs.

Situation déclenchante

Lors d'une randonnée en montagne, deux amis passionnés de sciences du numérique, souhaitent utiliser le magnétomètre intégré à une carte BBC micro:bit afin de fabriquer une boussole.

Ils souhaitent utiliser la carte afin que celle-ci leur indique la position du nord magnétique terrestre et l'associer à leur calculatrice graphique pour dessiner une représentation vectorielle du champ magnétique terrestre.



Problématique

Comment utiliser la carte BBC micro:bit comme une boussole ?
Mesurer le champ magnétique terrestre avec une carte micro:bit.

Matériel nécessaire

- Une calculatrice TI-83 Premium CE Edition Python.
- Un câble USB min-USB micro.
- Une carte BBC micro:bit.



Principe de fonctionnement

Un magnétomètre mesure l'intensité du champ magnétique selon chacun des trois axes. Il peut être utilisé pour créer une boussole numérique ou pour explorer des champs magnétiques, tels que ceux générés par un aimant permanent ou autour d'une bobine traversée par un courant.

L'interprétation de l'intensité du champ magnétique n'est pas facile. Chaque magnétomètre est différent et nécessitera un étalonnage préalable à toute utilisation, un peu comme un GPS du commerce.

Cette étape s'appelle la calibration de la carte.

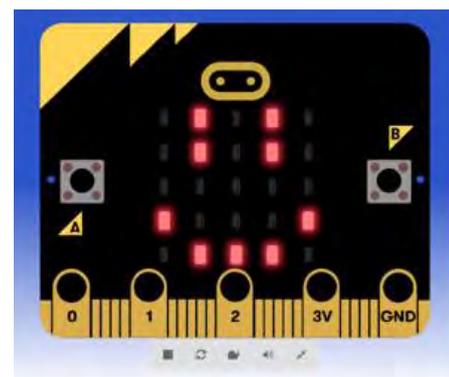
Remarques

L'étalonnage de la boussole entraînera une pause de votre programme jusqu'à ce que l'étalonnage soit terminé. L'étalonnage consiste en un petit jeu pour dessiner un cercle sur l'écran DEL en tournant l'appareil.

Le jeu consiste à tourner la carte BBC micro:bit afin en suivant le mouvement du symbole ∞ jusqu'à ce que tous les pixels de la carte soient allumés. Lorsque l'étalonnage est terminé, la carte affiche un smiley.

Les mesures effectuées par le micro:bit sont en nano tesla et seront converties en micro tesla.

Le facteur d'échelle de représentation graphique des vecteurs sera à ajuster manuellement dans le script.



Mise en œuvre

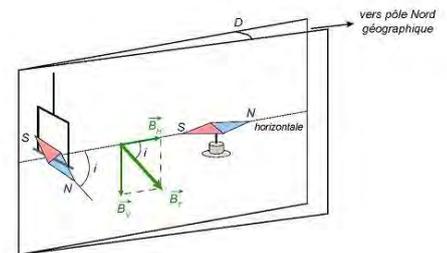
Les commandes du magnétomètre ressemblent beaucoup à celles de l'accéléromètre. Nous utilisons uniquement les valeurs x et y pour déterminer la direction du champ magnétique. Ne pas oublier : avant d'utiliser la boussole, de l'étalonner, sinon les lectures peuvent être fausses.

Ce script effectue une mesure du champ magnétique selon deux dimensions et renvoie les valeurs. Plus le champ est fort, plus le nombre est grand.

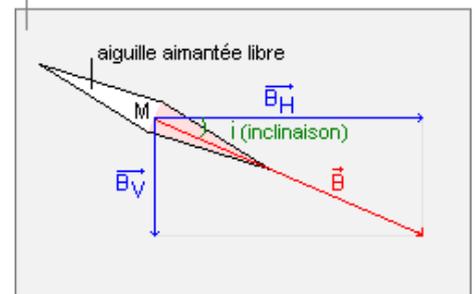
Le cap de la boussole est donné sous la forme d'un entier compris entre 0 et 360, représentant l'angle en degrés, dans le sens des aiguilles d'une montre, avec le nord à 360.

Le script à réaliser doit permettre à la carte micro:bit d'afficher par l'intermédiaire de la matrice de DEL, l'orientation magnétique vers laquelle on se situe.

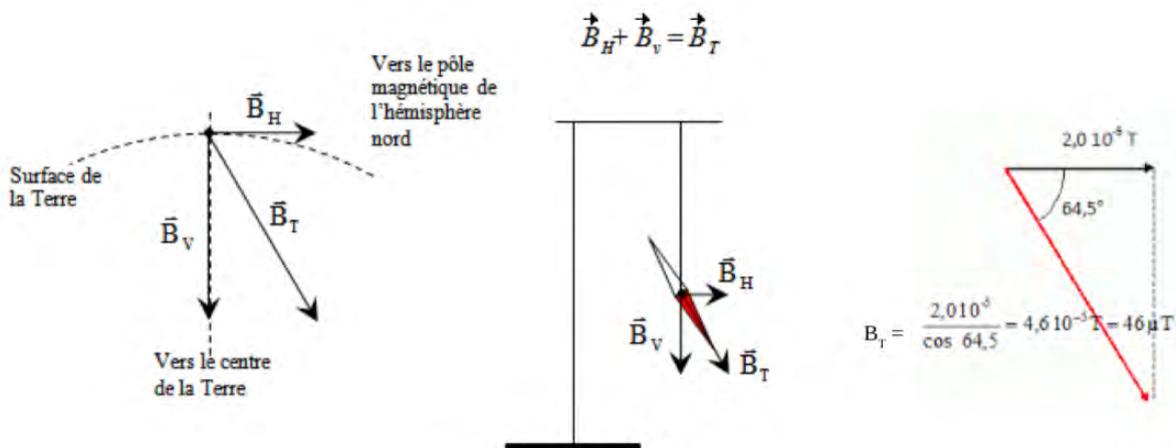
L'écran graphique de la calculatrice affichera les mesures du champ magnétique terrestre (dans le plan xOy), ainsi que sa décomposition vectorielle comme indiqué ci-contre.



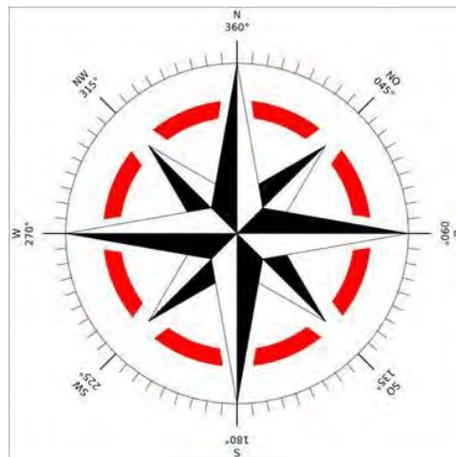
plan du méridien magnétique (plan contenant l'axe sud-nord de l'aiguille et le point M)



Exemple théorique de calcul du champ magnétique terrestre total.



La réalisation du script est effectuée en prenant l'orientation du nord à 360°. La rotation s'effectue dans le sens trigonométrique inverse.



A) Création de la boussole.

- Chargement des modules de la carte micro:bit et **ti_system**.
- **display.clear** éteint les DEL de la carte micro:bit.
- Création de la fonction **az** permettant d'afficher un élément de la chaîne de caractères en fonction de la valeur renvoyée par l'instruction **compass.heading()**. Cette fonction donne directement la direction en degrés par rapport au nord magnétique, quelle que soit la position de la carte micro:bit. Elle utilise les trois composantes sur les trois axes X,Y,Z pour en calculer l'angle cap.
- **compass.calibrate()** étalonnage du magnétomètre.
- Boucle Tant que (touche annul non pressée)
 - **compass.heading()** lecture de la position.
 - Affichage de la position par appel de la fonction **az**.

```
ÉDITEUR : BOUSSOLE
LIGNE DU SCRIPT 0001
from mb_sensr import *
from mb_disp import *
from ti_system import *

def az(a):
    r="NEESSOON"
    display.show(r[a],delay=400,wait=True)

display.clear()
compass.calibrate()
while not escape():
    a=int(compass.heading()/45)
    az(a)
```

Remarque

Lors de la réalisation des tests de fonctionnement du script ; éloigner la carte de votre ordinateur afin de limiter les perturbations du champ magnétique terrestre par le rayonnement électromagnétique de l'ordinateur.

B) Représentation du vecteur champ magnétique.

- Chargement des modules nécessaires aux calculs (maths), représentation graphique (**tiplotlib**), fonctionnement de la carte (microbit) et du magnétomètre (**mb_sensr**) et affichage des DEL.
- **compass.calibrate()** étalonner le magnétomètre.
- **display.clear()** éteindre les DEL de la carte.
- **compass.heading()** mesurer la position et l'affecter à la variable **angle**.
- **compass.get_field_strength()** mesure du champ magnétique et décomposition selon ($0x$) et ($0y$) **compass.get_x()**.
- Calcul des intensités en microtesla et de l'angle α (direction entre \vec{B} et l'horizontale).
- Représentation graphique selon un facteur d'échelle **e** à fixer et **plt.cls()** pour effacer l'écran.
- Représentation graphique des différents vecteurs selon une couleur spécifiée **plt.color(r,v,b)**.
- **sleep(5000)** attente de 5 sec.
- Affichage des composantes horizontale, verticale et totale du champ magnétique terrestre.

```
ÉDITEUR : COMPASS1
LIGNE DU SCRIPT 0001
from math import *
import tiplotlib as plt
from ti_system import *
from microbit import *
from mb_sensr import *
from mb_disp import *
compass.calibrate()
display.clear()
angle=compass.heading()
bt=compass.get_field_strength()
mx=compass.get_x()
my=compass.get_y()
mx=mx/1000
my=my/1000
bt=bt/1000
alph_yx=atan2(my,mx)*180/pi
alph_yx=round(alph_yx,0)
#Graphe
e=2
plt.cls()
plt.grid(1,1,"dash")
plt.title("Champ magnétique terr")
plt.pen("medium","solid")
plt.color(255,0,0)
plt.line(0,-5,mx/e,-5,"arrow")
plt.color(0,0,255)
plt.line(0,-5,0,my/e,"arrow")
plt.color(255,0,255)
plt.line(0,-5,mx/e,my/e,"arrow")

sleep(5000)
plt.cls()
plt.c_lor(0,0,0)
msg="direction/Nord Geo=%.2f deg"
"%angle
plt.text_at(5,msg,"center")
msg="Bx=%.2f muT"%mx
plt.text_at(6,msg,"center")
msg="By=%.2f muT"%my
plt.text_at(7,msg,"center")
msg="Champ mag=%.2f muT"%bt
plt.text_at(8,msg,"center")
msg="direction=%.1f deg"%alph_yx
msg="Bx=%.2f muT"%mx
plt.text_at(6,msg,"center")
msg="By=%.2f muT"%my
plt.text_at(7,msg,"center")
msg="Champ mag=%.2f muT"%bt
plt.text_at(8,msg,"center")
msg="direction=%.1f deg"%alph_yx

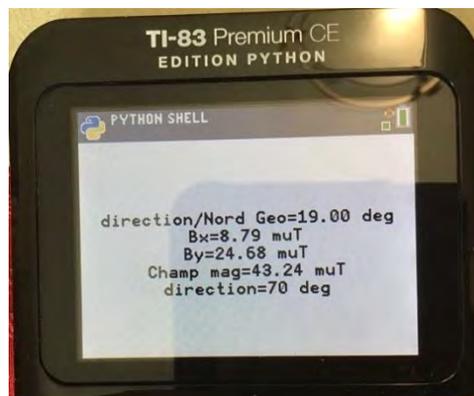
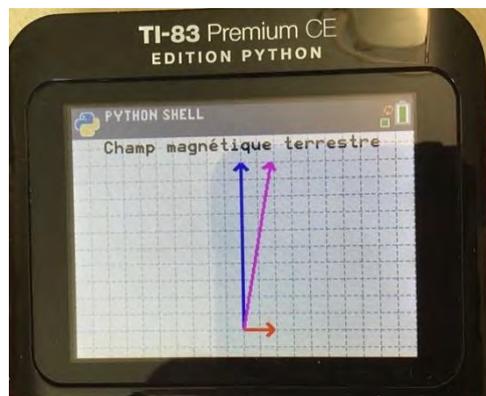
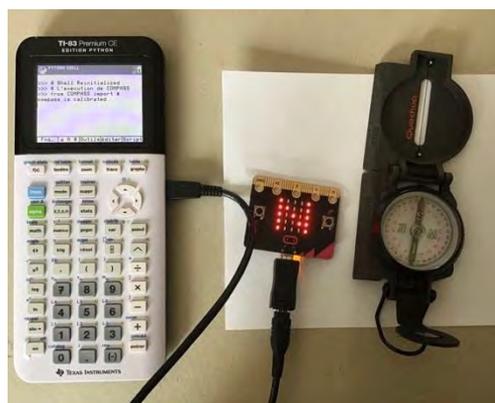
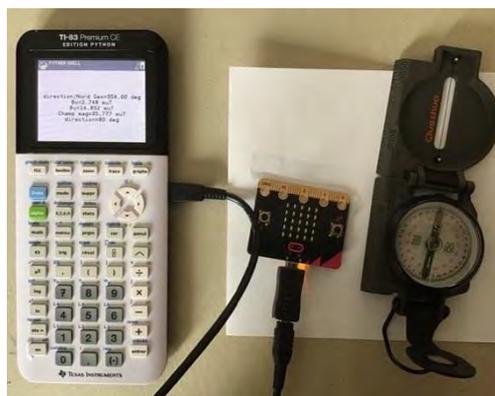
plt.text_at(9,msg,"center")
plt.show_plot()
```



Quelques résultats

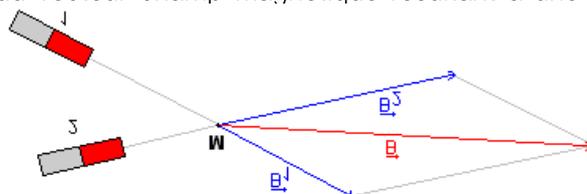
Transformer le micro:bit en une boussole qui illumine la DEL la plus proche de l'endroit où se trouve le nord.

Calibrer le magnétomètre. Observer si l'étalonnage reste (à peu près) identique au fil du temps et s'il est le même à l'intérieur ou à l'extérieur d'un bâtiment ou à proximité d'une structure qui contient beaucoup d'acier (par exemple un ascenseur).



Prolongement possible

- Utiliser le module « `ce_quivr` » pour représenter une somme vectorielle.
- Rechercher la valeur de l'intensité et les composantes du vecteur champ magnétique résultant à une expérience produite par des aimants.
- Utiliser pour la mesure du champ magnétique, l'instruction `compass.get_field_strength()`.
- Fabriquer une girouette.



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



Accélération d'un ascenseur

Référentiel, compétences

Lycée :

- Mesurer une grandeur physique à l'aide d'un capteur.
- Utiliser un dispositif comportant un microcontrôleur.
- Variabilité de la mesure d'une grandeur physique.
- Référentiel et relativité d'un mouvement. Description du mouvement d'un système.

Lycée Professionnel :

- Identifier les grandeurs d'entrée et de sortie (avec leur unité) d'un capteur.
- Mettre en œuvre un capteur.
- Décrire un mouvement.

Compétences :

- **S'approprier** : Énoncer une problématique.
- **Analyser Raisonner** : Choisir, élaborer, justifier un protocole.
- **Réaliser** : Mettre en œuvre un protocole expérimental.
- **Valider** : Confronter un modèle à des résultats expérimentaux.
- **Communiquer** : Présenter une démarche argumentée de manière synthétique et cohérente.

Situation déclenchante

Vous avez souvent remarqué lorsque vous êtes amenés à prendre un ascenseur de taille importante (hôtel, hôpital...) que celui-ci fournit lors de son mouvement des accélérations, décélérations importantes. Celles-ci peuvent parfois même causer une gêne pour certains passagers.



Problématique

Comment évaluer la variation de vitesse d'un ascenseur ?

Représenter graphiquement ces variations en fonction du temps

Exploiter la courbe obtenue (calcul de l'accélération, vitesse, temps de montée...)

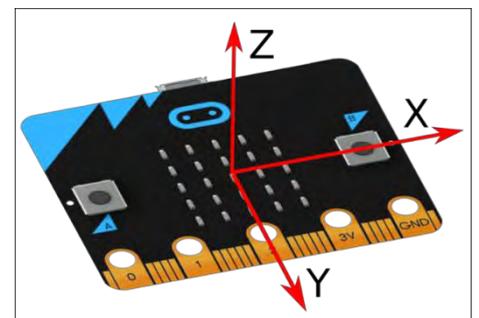
Matériel

- une calculatrice TI-83 Premium CE Edition Python.
- un câble USB Mini – USB micro.
- une carte BBC micro:bit.

Mesurer une accélération

L'accéléromètre de la carte BBC **micro:bit** mesure constamment aussi bien :

1. l'accélération de la pesanteur à la surface de la terre (environ 9.81 m/s^2), dirigée verticalement vers le bas.
2. que des accélérations dues aux changements de vitesse dans la direction des 3 axes **X****Y****Z**.



L'accéléromètre est bien visible sur la carte **micro:bit**.

Il est tout-à-fait identique à ceux qui se trouvent dans les smartphones et qui permettent de déterminer leur orientation ou leurs mouvements.

Remarque

Afin d'obtenir des mesures de qualité, on évitera lors de la réalisation de l'expérience de porter sur soi la calculatrice et la carte BBC **micro:bit**. L'ensemble pourra par exemple être contenu et fixé ou calé dans une boîte, puis déposé sur le sol de l'ascenseur avant le départ de celui-ci.

La carte peut être fixée dans la boîte (à $t = 0 \quad z > 0$) ou le contraire si on le souhaite, mais attention à l'interprétations des résultats. N'oublions pas que toute étude de mouvement nécessite le choix d'un référentiel à préciser.



Mise en œuvre

Chargement des modules nécessaires : **ti_plotlib** (représentation graphique), **ti_system** (export des données), **microbit** (gestion de la carte) et **mb_sensr** (accéléromètre).

Création d'une fonction **graphe()** pour représenter les données.

plt.cls() : effacer l'écran.

plt.window() : réglages de la fenêtre graphique.

plt.grid() : affichage et réglage de la grille ;

plt.axes() : affichage des axes.

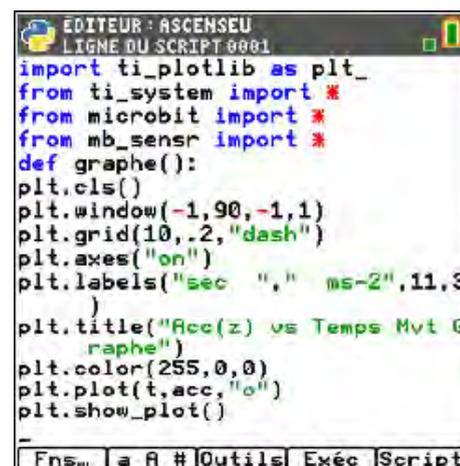
plt.labels : étiquettes sur les axes ; **plt.title()** : Titre du graphique.

plt.color() : couleur rouge pour la représentation des données.

plt.plot() : Représentation expérimentale.

plt.show() : Affichage de la représentation (hors de la boucle).

- Création d'une fonction **mvt()** acquisition des mesures de l'accélération et export des données vers les listes de la calculatrice.
- Initialisation des listes de données et de l'accéléromètre.
- Boucle FOR ; acquisition des données (à modifier selon les conditions expérimentales).
 - **store_list(" nom », var)** : Export vers les listes de la calculatrice , L₁ pour le temps et L₂ pour l'accélération.



```
EDITEUR : ASCENSEU
LIGNE DU SCRIPT 0001
import ti_plotlib as plt_
from ti_system import *
from microbit import *
from mb_sensr import *
def graphe():
plt.cls()
plt.window(-1,90,-1,1)
plt.grid(10,.2,"dash")
plt.axes("on")
plt.labels("sec ", " ms-2",11,3
)
plt.title("Acc(z) vs Temps Mvt G
raphe")
plt.color(255,0,0)
plt.plot(t,acc,"o")
plt.show_plot()
_
Fns... | a A # | Outils | Exéc | Script
```



```
EDITEUR : ASCENSEU
LIGNE DU SCRIPT 0015
#Mouvement
def mvt(D,dt):
temps=[]
acc_z=[]
for i in range(360):
acc=accelerometer.get_z()
t=i*dt
sleep(1000*dt)
temps.append(t)
acc_z.append(acc)
#Export des données
store_list("1",temps)
store_list("2",acc_z)
_
Fns... | a A # | Outils | Exéc | Script
```

Pour réaliser un exemple d'acquisition de données, exécuter le script en appelant la fonction **mvt(D, dt)**. D représente la durée totale du trajet en ascenseur et **dt**, l'intervalle de temps entre deux mesures.

Les mesures sont sauvegardées dans les listes L₁ (temps) et L₂ (accélération) de la calculatrice.

La fonction **graphe()** effectue la représentation graphique des données.



Remarques

Fonctionnement de l'accéléromètre.

Sensibilité : l'accéléromètre implanté dispose de plusieurs gammes de mesures, mais sur la carte Micro:Bit, seule la plage $-2g$ à $+2g$ est disponible (g étant bien sûr la valeur de l'accélération de la pesanteur). Cette mesure se fait avec une résolution de 10 bits (1024 valeurs différentes) de façon individuelle sur un axe (par exemple l'axe Z) avec la fonction :

`accelerometer.get_z()`

qui retourne un entier compris entre ± 2000 (en $10^{-3}g$)

Acquisition des données.

On nomme D la durée du mouvement et dt le pas d'acquisition. Le nombre d'itérations de la boucle est donc `int(D/dt)`.

L'accélération s'exprime en $m.s^{-2}$. La conversion est effectuée `acc = acc*1000/9.81` à partir de la mesure effectuée en « mg » par la carte BBC micro:bit.

($mg = 10^{-3}g$: g étant l'accélération de la pesanteur et non le gramme dans ce contexte).

Un exemple

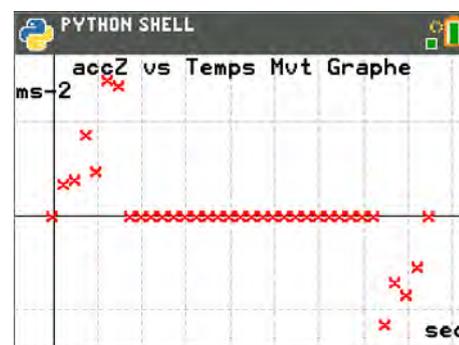
Dans l'exemple ci-contre, on réalise l'acquisition de l'accélération de l'ascenseur selon l'axe Z en fonction du temps. Les mesures sont effectuées toutes les 0,25 s sur une durée totale de 1min30s. Soit 90 s et 4 mesures par seconde.

Les réglages de la fenêtre graphique sont effectués avec l'instruction `plt.window(-1,90,-1.5,1.5)` pour laquelle on peut préciser les paramètres de la façon suivante.

`plt.window(-1, Nbre mesures, Acc(min), Acc(max))`

A partir de la représentation graphique du mouvement, il est intéressant de rechercher la nature de celui-ci.

Parmi quelques directions possibles, pouvant favoriser la recherche d'une réflexion par groupe d'élèves, dès lors que les données sont exportées dans les listes de la calculatrice, afin d'être partagées.



- Nombre de phases du mouvement.
- Lorsque `accZ < 0`, l'ascenseur monte-t-il vers l'étage supérieur ?
- Lorsque `accZ = 0`, est-il possible de déduire que l'ascenseur est à l'arrêt ? Justifier votre réponse.
- Déterminer le temps de descente ou de montée entre deux étages.

Prolongements possibles

- Donner une valeur numérique à g puis déterminer l'accélération de l'ascenseur a_z .
- Calculer numériquement la vitesse de l'ascenseur v_z à partir de son accélération a_z . (Numériquement, les intégrations peuvent être approchées par la méthode d'Euler.)
- Représenter graphiquement à l'aide d'un script Python, les vecteurs vitesse et accélération (on peut également utiliser le module `ce_quivr`).



Détecter une émotion



A. Yazici

Référentiel, Compétences

Lycée :

- Utiliser un dispositif comportant un microcontrôleur.
- Mesurer la conductivité électrique d'un milieu.

Lycée Professionnel :

- Réaliser un montage à partir d'un schéma.
- Mesurer une grandeur physique à l'aide d'un capteur électrique résistif.

Compétences :

- **S'approprier** : Rechercher et organiser l'information en lien avec la problématique étudiée.
- **Analyser Raisonner** : Formuler des hypothèses.
- **Réaliser** : Mettre en œuvre les étapes d'une démarche.
- **Valider** : Proposer d'éventuelles améliorations d'une démarche.
- **Communiquer** : Expliquer une démarche.

Situation déclenchante

La start-up française Neotrope a mis au point un bracelet « **Affect tag** » qui mesure les émotions de son utilisateur. Cette technologie se base sur la physiologie du corps humain à l'inverse de celle de Apple qui se base sur la reconnaissance faciale.

Les grandes marques de marketing s'intéressent à ces méthodes pour connaître la réponse émotionnelle et le niveau d'engagement de leurs clients.

Nous utiliserons la technique de **la réponse galvanique de la peau** pour suivre, l'état émotionnel (réponse au stress galvanique) d'un sujet. Il s'agit de mesurer la conductance électrique de sa main à l'aide d'un capteur approprié.



La carte micro:bit peut-elle nous aider à détecter le changement émotionnel chez ce sujet ?

Problématique

Qu'est-ce que la technique GSR ?
Comment relier une émotion à une grandeur physique ?
Comment générer une alerte lumineuse lors de l'apparition d'une émotion ?



Détecter une émotion



A. Yazi

Principe de la réponse galvanique de la peau

Le système Galvanic Skin Response (GSR) utilisé en biomédical, indique les niveaux de conductance électrique du corps humain. Il se comporte comme un ohmmètre si précis qu'il est capable, lorsqu'elle est calibrée, de créer un profil de santé (équilibre de tous les systèmes du corps).

Ce module GSR permet de mesurer la conductivité de la peau à travers deux électrodes. La conductivité varie en fonction des émotions et le capteur délivre une valeur analogique en fonction de celle-ci.

Lorsque la tension est lue entre les bornes (0V–3,3V), les informations renvoyées sont autour de 1020. La lecture de la tension analogique est convertie en valeur numérique avec 3,3 volts approchant la limite supérieure de 1023. Une tension de 1,5 volt devrait renvoyer une lecture d'environ 512 sur le micro:bit lors de sa conversion analogique-numérique.



La valeur mesurée dépendra des individus (adultes, enfants, mains moites...).

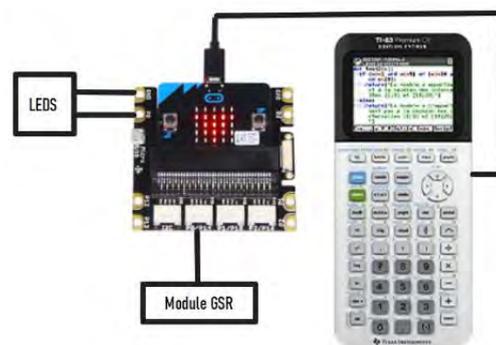
Matériel nécessaire

- Une calculatrice TI-83 Premium CE Edition Python.
- Une carte BBC micro:bit.
- Une carte d'extension Shield (Grove).
- Un câble miniUSB-microUSB.
- Module GSR (Galvanic Skin Reponse).
- Une LED ou une ceinture de LED.
- Câbles avec pinces crocodiles.

Mise en œuvre

A partir du schéma ci-contre, réaliser le montage :

- Le module GSR est connecté au pin0.
- La ceinture des Led est connectée au pin8 via des câbles avec pinces crocodiles. On peut substituer cette ceinture de Led par un capteur Led grove qu'on peut le connecter au pin1 (par exemple).



Détecter une émotion

- On commence par collecter la valeur analogique dans un état dit « normal », ensuite lors de l'apparition d'une émotion « qui sera simulée par des contractions des mains », on assistera à une baisse de la valeur analogique et de la tension aux bornes des électrodes du GSR.
- Ensuite, on peut définir une valeur seuil, à partir de laquelle, la carte micro:bit envoie un signal lumineux (ou sonore) pour nous avertir d'une détection d'émotion, comme ce qui est illustré dans la photo ci-contre.
- Le script suivant tient compte de ces spécificités.



Chargement des modules nécessaires :

- `microbit` pour le fonctionnement de la carte.
- `ti_system`.
- `ti_plotlib` pour les représentations graphiques.
- `mb_pins` connexion aux ports P0 à P2.
- `mb_grove` pour la gestion des capteurs grove .
- Création d'une fonction `emotion` qui prend `n` comme argument.
- Boucle **Tant que** (la touche **annul** n'est pas pressée)
 - Lecture de la valeur analogique sur le pin0 et affectation à la variable `a`.
 - Conversion de la valeur analogique `a` en tension et affectation à la variable `u`.
 - `plt.cls()` effacement de l'écran de la calculatrice.
 - Affichage de la valeur lue avec une décimale.
 - Envoi d'un signal lumineux via le port 8 si la valeur analogique est inférieure au seuil `n`.

```
ÉDITEUR : EMOTION
LIGNE DU SCRIPT 0001
import ti_plotlib as plt
from microbit import *
from ti_system import *
from mb_pins import *
from mb_grove import *
from mb_disp import *
display.clear()
def emotion(n):
    while not escape():
        a=pin0.read_analog()
        u=a*(3.3/1023)
        msg1="analog =%.1f"%a
        msg2="tension =%.1f"%u+"V"
        plt.cls()
        plt.text_at(5,msg1,"center")
        plt.text_at(7,msg2,"center")
        sleep(2000)
        if a<n:
            pin8.write_digital(1)
Fns... | a A # | Outils | Exéc | Script
```

Lors de l'exécution du script, on a fixé la valeur seuil `s` à 600 soit 1.94V :

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de EMOTION
>>> from EMOTION import *
>>> emotion(600)
Fns... | a A # | Outils | Éditer | Script
```



```
PYTHON SHELL
analog =670.8
tension =2.2V
```

Détecter une émotion

Si la valeur analogique lue est en-dessous de 600, la ceinture Led (témoin lumineux) se déclenche pour alerter du changement de l'état émotionnel.



Prolongement

Détermination de la résistance électrique du corps humain

- Faites une recherche concernant la résistance électrique du corps humain.
- Se référer au site du constructeur du capteur : https://wiki.seeedstudio.com/Grove-GSR_Sensor/
- Relever la formule qui donne la résistance électrique du corps humain. (Attention aux unités !).
- Elaborer un script qui affiche la résistance électrique du corps humain.

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



Mesurer une accélération

Référentiel, compétences

Lycée :

- Mesurer une grandeur physique à l'aide d'un capteur.
- Utiliser un dispositif comportant un microcontrôleur.
- Variabilité de la mesure d'une grandeur physique.
- Référentiel et relativité d'un mouvement. Description du mouvement d'un système.
- Collecter des données sur un mouvement (capacité expérimentale).

Lycée Professionnel :

- Identifier les grandeurs d'entrée et de sortie (avec leur unité) d'un capteur.
- Mettre en œuvre un capteur.
- Décrire un mouvement.

Compétences :

- **S'approprier** : Énoncer une problématique.
- **Analyser Raisonner** : Choisir, élaborer, justifier un protocole.
- **Réaliser** : Mettre en œuvre un protocole expérimental.
- **Valider** : Confronter un modèle à des résultats expérimentaux.
- **Communiquer** : Présenter une démarche argumentée de manière synthétique et cohérente.

Situation déclenchante

La carte BBC micro:bit communique avec le monde extérieur par l'intermédiaire de capteurs intégrés (température, accéléromètre, lumière, magnétomètre). L'utilisation de ces capteurs peut donner l'occasion de répliquer avec des moyens moderne une expérience qui passionna les scientifiques d'une autre époque. L'apport des nouvelles technologies facilite un regard vers l'histoire, tout en adoptant une posture de recherche similaire à celle qui anima les grands scientifiques. Peut-on utiliser la carte BBC micro:bit afin de retrouver une valeur de l'accélération de la pesanteur.



L'accélération de pesanteur g est l'objet d'étude de la gravimétrie. Elle n'est pas constante à la surface de la Terre, variant entre autres, avec l'altitude mais aussi la latitude en diminuant du pôle ($9,83 \text{ m/s}^2$) à l'équateur ($9,78 \text{ m/s}^2$) en raison de l'aplatissement de la Terre aux pôles et de la force centrifuge perçue dans le référentiel terrestre due à la rotation de la Terre autour de son axe.

Dans cette activité, on propose de marcher sur les pas de Pierre BOUGUER afin de retrouver à l'aide d'un pendule une valeur de l'accélération de la pesanteur.

Problématique

Comment fonctionne une mesure de signal sur la carte BBC micro:bit ? Comment représenter des données ? Utiliser les données pour effectuer un calcul ?



Mesurer une accélération

Un peu d'histoire

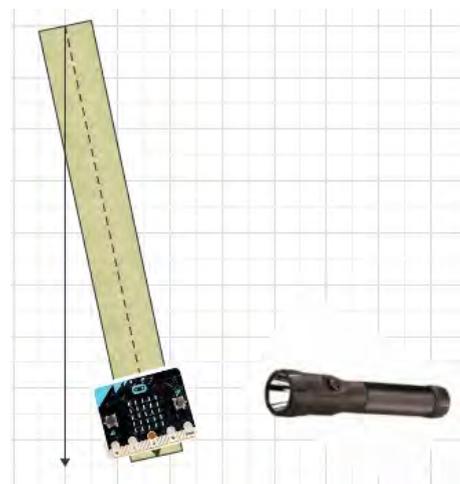
C'est au cours de l'expédition géodésique au Pérou entre 1735 et 1744, que Pierre BOUGUER réalise ses mesures pendulaires et ses mesures de la déviation de la verticale consignées dans son livre *La Figure de la Terre*, paru en 1749. BOUGUER a été envoyé au Pérou par l'Académie des Sciences, en compagnie de Godin, La Condamine et Jussieu pour réaliser des mesures astronomiques et des mesures géodésiques. Le but est de trouver quelle est la forme exacte de la Terre : sphérique, aplatie aux pôles ou allongée selon l'axe de rotation. Ce voyage, qui dure 9 ans, est épique. Les académiciens subissent un tremblement de terre, une épidémie, une révolte de la population locale et un climat montagneux parfois rude. Coupés de la France, ils doivent également subvenir à leurs besoins sur place, et pour ce faire l'un d'entre eux développe un trafic d'or ! Les personnes de l'expédition se divisent, certaines sont mortes, d'autres ont sombré dans la folie... Ce voyage est cependant d'une très grande richesse d'un point de vue scientifique : mesure d'un degré du méridien, mesure de la vitesse du son, mesure de la réfraction atmosphérique, détermination des altitudes grâce au baromètre et, nous allons le voir, mesure de la diminution de la pesanteur avec l'altitude ainsi que mesure de la déviation de la verticale par l'attraction des montagnes.



(Extrait de Culture Sciences Physiques [en ligne] - Ens Lyon - 2013

Matériel nécessaire

- Une règle plate.
- Du ruban adhésif ou élastiques.
- Un roulement à bille à monter sur une tige métallique.
- Une lampe de poche.
- Une carte BBC micro:bit.
- Un câble USB A mini USB micro.
- Une calculatrice graphique TI-83 Premium CE Edition Python.
- Des potences et noix de serrage.
- Il est aussi possible d'utiliser un pendule pesant prêt à l'emploi sur lequel on fixera la carte micro:bit.



Mesurer une accélération

Mise en œuvre

L'expérience consiste à faire osciller une règle autour d'un axe fixe, sur lequel on a fixé la carte BBC micro:bit.

Une lampe de poche fixée sur la règle éclaire un plan horizontal sur lequel est placée la carte micro:bit.

La règle est écartée de sa position d'équilibre.

On appuie sur le bouton **A** de la carte micro:bit afin de déclencher l'exécution du script.

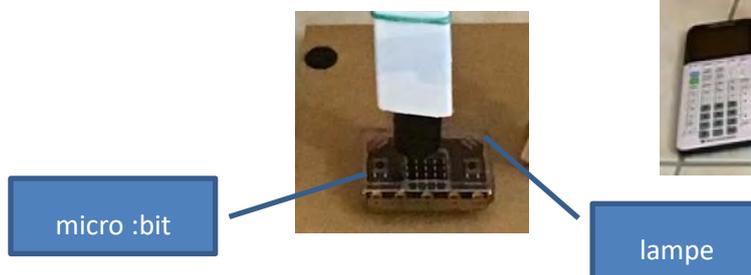
Lorsque la lampe passe devant la carte, l'intensité lumineuse captée par le capteur augmente brutalement.

On incrémente alors un compteur de 1 unité.

Le pendule oscille librement, le programme enregistre le nombre de passage devant la lampe et la durée totale de l'expérience.

La période T du pendule correspond à la durée d'une oscillation. Pour un écart $\alpha < 30^\circ$ et pour un pendule simple, la période est donnée par la relation :

$$T = 2\pi \sqrt{\frac{l}{g}}$$



Remarque

Le pendule est monté sur une tige métallique équipée d'un roulement à billes afin de limiter les frottements.

Fixer la lampe sur la règle à la place de la carte micro:bit évite d'avoir à utiliser un long câble USB et également d'être confronté à des dysfonctionnements (incompatibilités de certains adaptateurs USB).

Mesurer une accélération

Algorithme

Le script à réaliser peut correspondre à la proposition de l'algorithme suivant.

```
t0 ← mesure du temps           # Valeur initiale du temps
n ← 0                          # Nombre de passage devant la lampe
n_max ← 10
    Si le bouton A a été pressé Alors
        Tant que n_max < 10 : #cela fera 5 aller-retours pour n_max =10
            lumi ← mesure du niveau de luminosité #indentation
            Si lumi > l0    alors #Seuil de luminosité ambiante (appel de fonction ajuste)
                n ← n + 1
                Allumer le pixel numéro n//2 de la ligne 1 (on aura donc une progression
                visuelle du pixel allumé)
            Fin Si
        Fin Tant que
    tf ← mesure du temps #mesure de tf en sortie de boucle tant que
    duree ← tf - t0
    période ←  $\frac{2 \times \textit{duree}}{n}$ 
    Afficher période
```

Remarque

La lampe étant située sur la verticale passant par la position d'équilibre, une oscillation enregistre deux passages.

Ainsi si l'on souhaite enregistrer le temps mis par le pendule pour effectuer 10 oscillations, on fixera le test d'arrêt à une valeur ***n*=20**.

La valeur ***lumi*** de luminosité de la lampe pourra préalablement être mesuré en utilisant un script destiné à cet effet.

Le calcul de la période du pendule est trivial.

Le pendule effectue $\frac{n}{2}$ oscillations en une durée de « time »

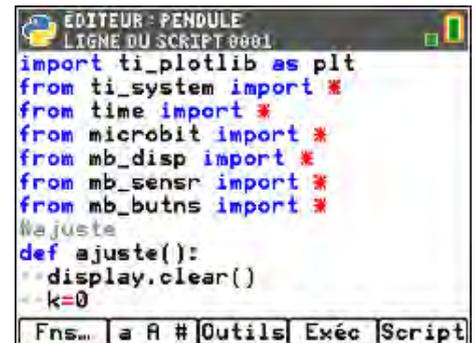
Le pendule effectue 1 oscillation en une durée de T (période)

Les grandeurs étant proportionnelles, on obtient : $T = \frac{2 \times \textit{time}}{n}$



Étapes de résolution

- Chargement des modules nécessaires au fonctionnement du script.
- Le module **microbit** nécessite les modules **mb_disp** (affichage); **mb_sensr** (capteurs) et **mb_butns** (boutons).
- Création de la fonction **ajuste** afin de mesurer le niveau de luminosité ambiante. Pendant cette étape, la carte affiche la valeur du numéro de mesure **k**.



```
EDITEUR : PENDULE
LIGNE DU SCRIPT 0001
import ti_plotlib as plt
from ti_system import *
from time import *
from microbit import *
from mb_disp import *
from mb_sensr import *
from mb_butns import *
#ajuste
def ajuste():
    display.clear()
    k=0
```

- Création de la fonction **pixel**. Lors du passage du pendule devant la lampe, le pixel de rang $n//2$ sera allumé. La ligne centrale (n^2) est choisie.
- Création d'une fonction **periode**.
- Initialisation des variables n ; t_0 ; t_f et *periode*.
- On stocke dans **l0**, le résultat de la luminosité ambiante mesuré par la fonction **ajuste**.

- Si le bouton A à été pressé.
- **display.clear()** éteindre les DEL de la carte micro:bit.

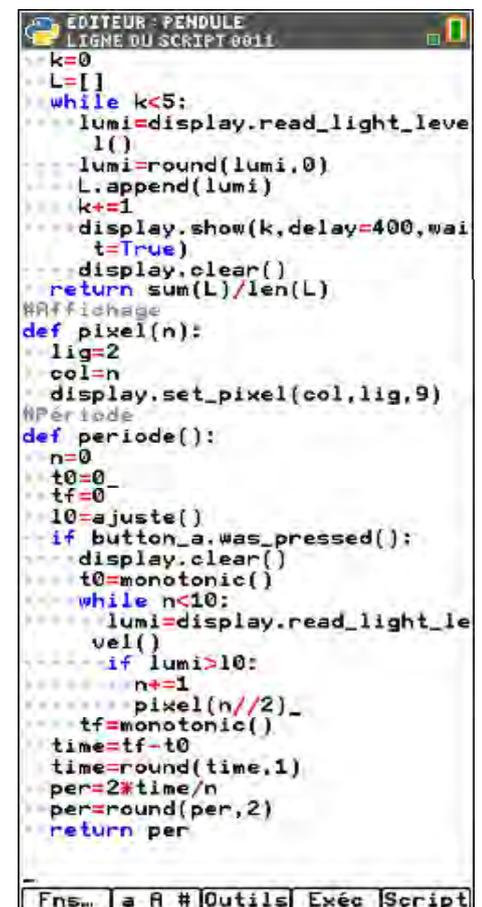
- début de la mesure du temps.

Boucle Tant que :

- Mesure du niveau de luminosité.
 - Si le niveau $> l_0$.
 - Incrémenter la valeur de n .
 - Afficher le pixel de rang $n//2$.
- Mesure de la durée t_f .

Fin de la boucle Tant que

Calcul de la période T.



```
EDITEUR : PENDULE
LIGNE DU SCRIPT 0011
k=0
L=[]
while k<5:
    lumi=display.read_light_level()
    lumi=round(lumi,0)
    L.append(lumi)
    k+=1
    display.show(k,delay=400,wait=True)
    display.clear()
return sum(L)/len(L)
#Affichage
def pixel(n):
    lig=2
    col=n
    display.set_pixel(col,lig,9)
#Période
def periode():
    n=0
    t0=0
    tf=0
    l0=ajuste()
    if button_a.was_pressed():
        display.clear()
        t0=monotonic()
        while n<10:
            lumi=display.read_light_level()
            if lumi>l0:
                n+=1
                pixel(n//2)
            tf=monotonic()
        time=tf-t0
        time=round(time,1)
        per=2*time/n
        per=round(per,2)
    return per
```

Mesurer une accélération

Calcul de la valeur de g

On enregistrera plusieurs essais, puis on effectuera un calcul de la moyenne des périodes de chaque essai. Les résultats pourront être présentés dans un tableau.

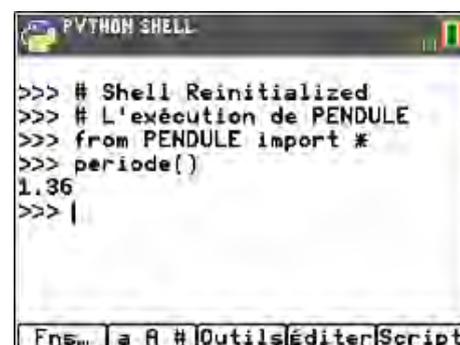
Après réalisation du script, la valeur de la période peut être obtenue dans la console en appuyant sur la touche VAR. A partir de la mesure de T et de la relation $T = 2\pi \sqrt{\frac{l}{g}}$, montrer que : $g = \frac{4\pi^2 \times l}{T^2}$

Mesurer la longueur du pendule, puis en déduire une mesure de g .

Exemple

Exécuter le script, appeler la fonction **periode**.

- Pendant la phase de mesure du niveau moyen de luminosité ambiante (sur 5 mesures), La carte affiche le n° de mesure.
- Appuyer sur le bouton **A** de la carte micro:bit.
- Après 10 oscillations, la valeur de la période en s est affichée.
- Le pendule utilisé à une longueur de 45 cm.
- Une valeur expérimentale de g sera donc : $g = \frac{4 \times \pi^2 \times 0.45}{1.36^2}$
soit $g = 9.6 \text{ m} \cdot \text{s}^{-2}$



```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de PENDULE
>>> from PENDULE import *
>>> periode()
1.36
>>> |
```

Afin d'affiner cette mesure de la valeur de g , d'autres expérimentations sont nécessaires. On pourra en particulier les réaliser pour différentes valeurs de la longueur du pendule.

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





education.ti.france



@TIEducationFR



TledtechFR



Contactez notre service client TI-Cares
education.ti.com/fr/csc
01 41 04 60 40

education.ti.com/fr