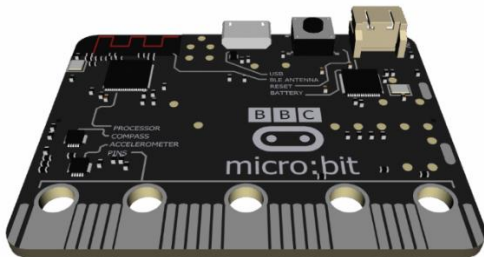


## Mise en œuvre de la carte Micro:Bit

### I - Présentation de la carte BBC Micro:bit

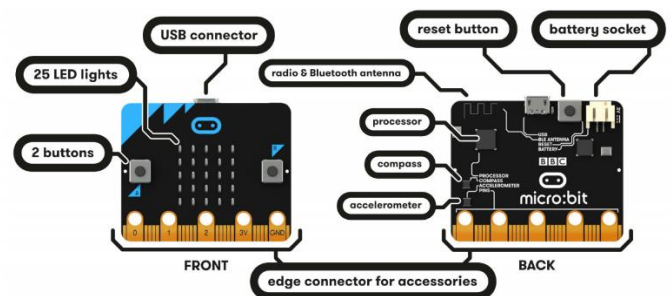
#### 1.1 Spécificités



Processeur : PU ARM® Cortex™ M0 32 bits  
Fréquence de l'horloge de processeur : 16 Mhz  
RAM : 16 Ko  
Tension des broches : 3,3 V  
6 broches analogiques + 19 broches digitales

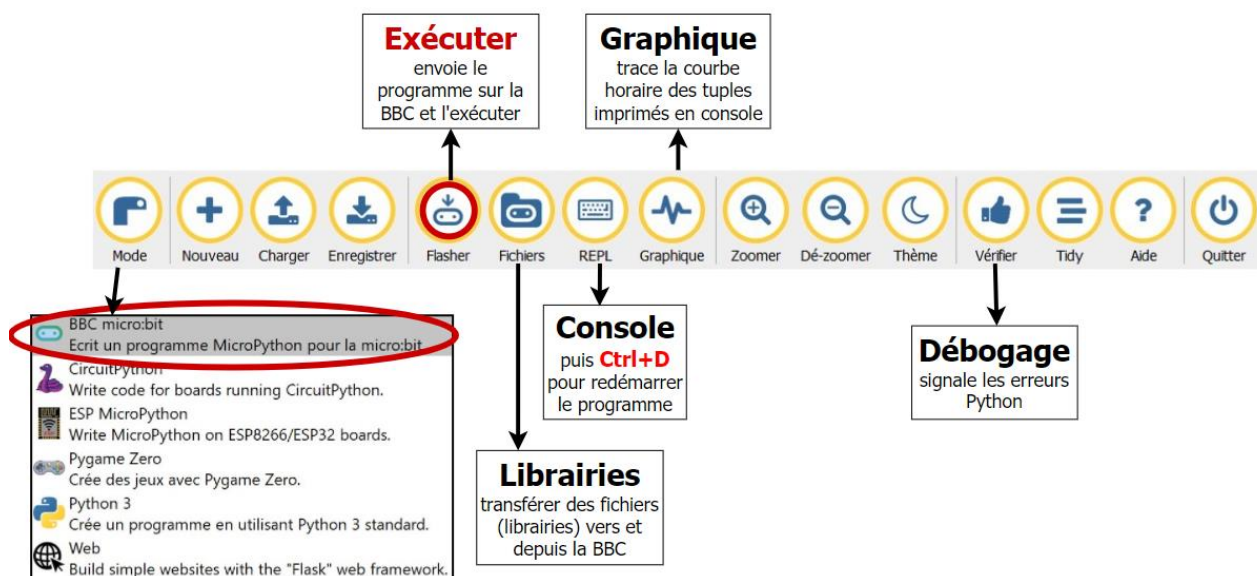
#### La carte intègre également :

- Un bouton de réinitialisation, libellé « R » pour « Reset »
- Deux boutons programmables, libellés « A » et « B » sur la carte
- Un accéléromètre 3D
- Une boussole magnétique 3D
- 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
- Un afficheur digital carré de 25 LED ( 5 × 5 ) rouges programmables



#### 1.2 Editeur Micro-Python

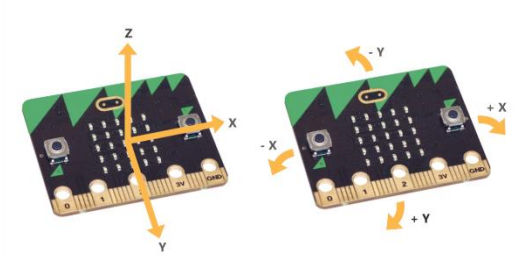
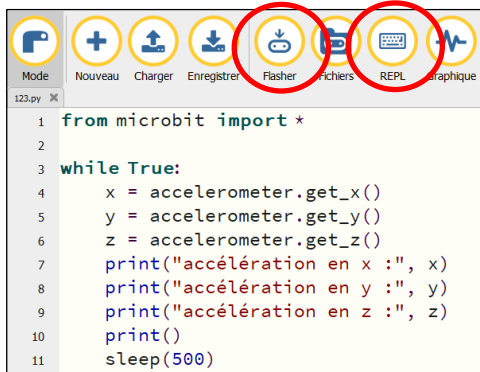
Le logiciel utilisé sera « Mu Editor » disponible sur votre ordinateur ou sur <https://codewith.mu>.



## II - Manipulations

### Manipulation 1 : Mise en œuvre de l'accéléromètre

- 1) Si vous ne savez pas ce qu'est un accéléromètre, veuillez consulter Google.
- 2) Copier le script suivant sur MU puis tester son fonctionnement en utilisant la console du mode REPL. *Un reset de la carte est nécessaire pour exécuter le script après son chargement.*



#### Documentation technique :

Mot clé Google : microbit, accéléromètre, python ou le lien suivant :

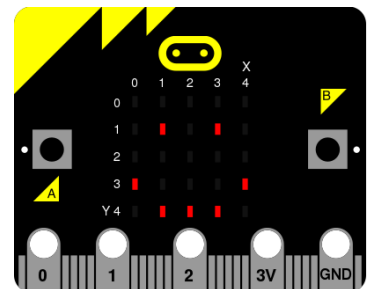
<https://microbit-challenges.readthedocs.io/en/latest/tutorials/accelerometer.html>

- 3) Placer la carte horizontalement puis verticalement pour valider la valeur de la mesure affichée. Quelle grandeur d'accélération mesurez-vous dans ces conditions ?
- 4) Modifier le programme afin d'afficher la résultante des trois vecteurs d'accélérations (la ligne *import math* doit figurer dans le script pour le calcul de la racine carrée). Montrez que l'accélération de la pesanteur est bien obtenue dans l'affichage de la mesure.

Ce lien GitHub va vous présenter la gestion de l'afficheur de la carte :

<https://github.com/bbcmicrobit/micropython/blob/v1.0.1/docs/tutorials/images.rst>

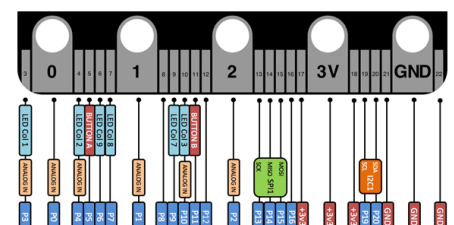
- 5) Modifier le programme afin d'afficher l'image HAPPY au démarrage du programme.
- 6) Faites-en sorte que l'image SAD apparaisse quand vous donnez un choc sur la carte. L'image Happy devra réapparaître au bout d'une seconde. *Utiliser une structure si-alors-sinon*
- 7) Afficher une barre de 5 pixels au milieu de l'afficheur. Faites-en sorte qu'elle puisse se déplacer de gauche à droite en tournant la carte micro:bit autour de l'axe Y.



### Mise en œuvre des broches d'Entrée/Sortie d'une carte Micro:bit

Les broches (pin) permettent à votre carte de communiquer avec les périphériques externes qui y seront connectés. Il y a 19 broches à votre disposition, numérotées de 0 à 16 et de 19 à 20. Les broches 17 et 18 ne sont pas disponibles.

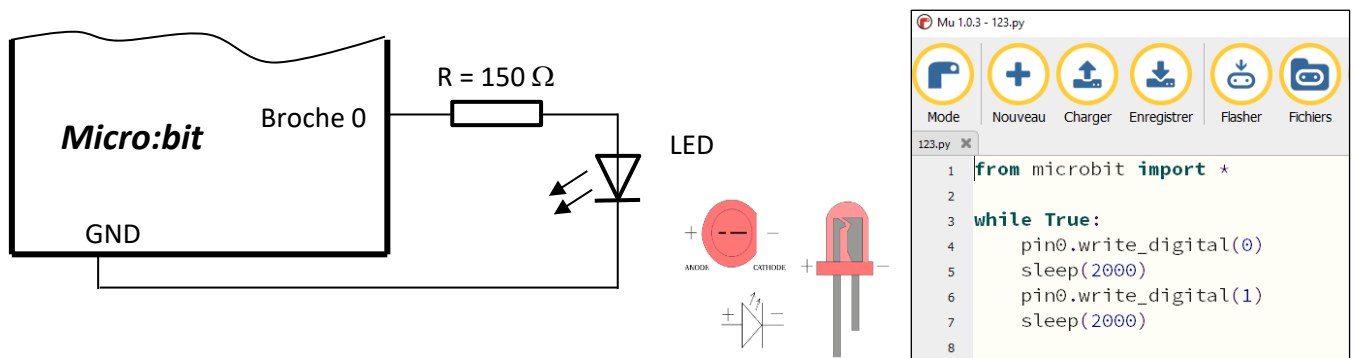
Connector Pin Layout



## Manipulation 2 : Clignotement d'une LED externe à partir de la carte Micro:Bit.

- 1) Charger le programme ci-dessous sous le logiciel Mu disponible sur votre ordinateur. Réaliser ce câblage afin de vérifier que la LED clignote bien.
- 2) Modifier le programme afin de faire clignoter la Led à une fréquence de 5 Hz s'il y a une action sur le bouton poussoir A et de 10 Hz s'il y a une action sur le bouton poussoir B. La LED restant allumée lorsque qu'aucun bouton n'est activé.

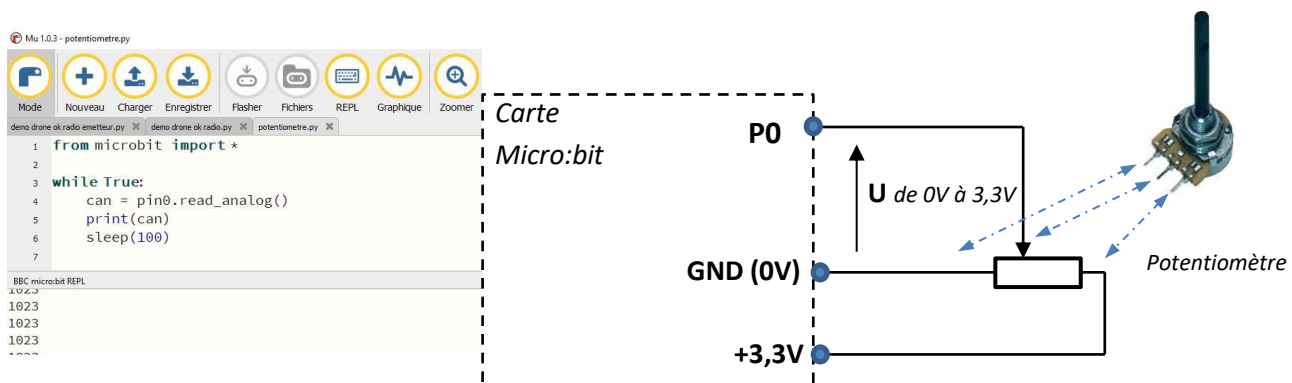
Se servir des boutons A et B : <https://microbit-challenges.readthedocs.io/en/latest/tutorials/buttons.html>



## Manipulation 3 : Convertisseur Analogique Numérique (CAN) sur une carte Micro:bit

La tension U qui est présentée sur la broche P0 évolue de 0 à 3,3V en fonction de la position de la tige du potentiomètre. La valeur de la tension sera convertie en une donnée numérique évoluant de 0 à 1023. Il s'agit d'une Conversion Analogique Numérique.

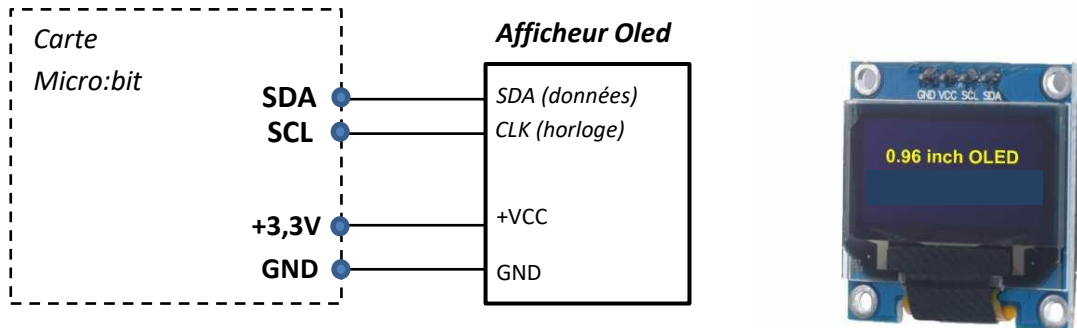
- 1) **Câbler** le circuit ci-dessous sur la platine d'essais puis saisir le programme sur le logiciel MU. Vérifier que l'on puisse faire varier la valeur des données affectées à la variable can.
- 2) **Câbler** la LED sur la broche P1. Modifier le programme afin de pouvoir gérer le clignotement de la LED en fonction de la position de la tige du potentiomètre **sur une plage de 20 Hz** tel que :  
Si can = 0 la LED est éteinte. Si can = 1023 la fréquence est de 20 Hz.



## Manipulation 4 : Mise en œuvre de l'afficheur Oled ssd1306.

Ce petit écran d'une diagonale de 0,96" est composé de 128x64 pixels. Il peut gérer des images ou du texte, la configuration proposée permet de gérer 4 lignes de 12 caractères.

Il sera relié à la carte micro:Bit en utilisant une liaison série I2C mettant en œuvre quatre fils. SDA (les données), CLK (l'horloge de synchronisation des données), VCC (3,3 V) et GND (0V)



### Programme de démonstration :

```
initialize()    #Initialisation de l'afficheur
clear_oled()

a="Bonjour" #variable de type String
b=2022     #variable de type Integer

#La fonction add_text(n° col, n°ligne, chaine de caractère) fonctionne ainsi :
add_text(2, 0, "Test Oled")
add_text(0, 1, a)
add_text(0, 2, str(b))
add_text(2, 3, "C'est ok ?" )
```

- 1) Faire le câblage de l'afficheur puis exécuter le script du fichier « Démon\_Afficheur\_Oled.py ».
- 2) Réaliser un programme capable de compter de 0 à 100 en boucle sur cet afficheur.
- 3) Utiliser le câblage de la manipulation 3 pour afficher la valeur de la fréquence de clignotement de la LED sur ce composant.
- 4) **Mise en œuvre du capteur de température** : Réaliser un programme capable de délivrer sur l'afficheur la valeur de la température ambiante exprimée en degrés Celsius. Vérifier la température à l'aide d'un thermomètre, si besoin étalonnez le capteur en modifiant votre programme.

<https://microbit-challenges.readthedocs.io/en/latest/tutorials/thermometer.html>

- 5) **Mise en œuvre du compas** : Réaliser un programme capable de délivrer sur l'afficheur la direction mesurée par le capteur que vous exprimerez en degré (n'oubliez pas d'insérer le module de calibration au début du programme).

<https://microbit-challenges.readthedocs.io/en/latest/tutorials/compass.html>