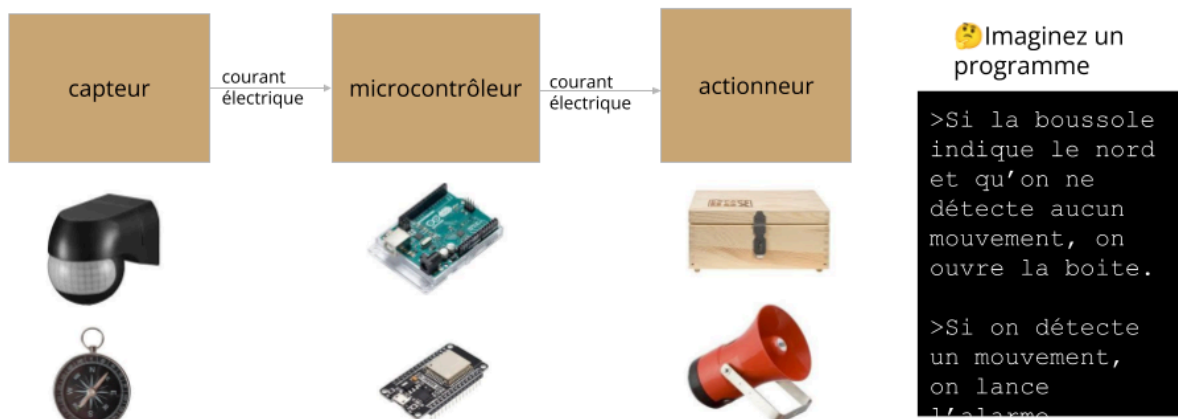


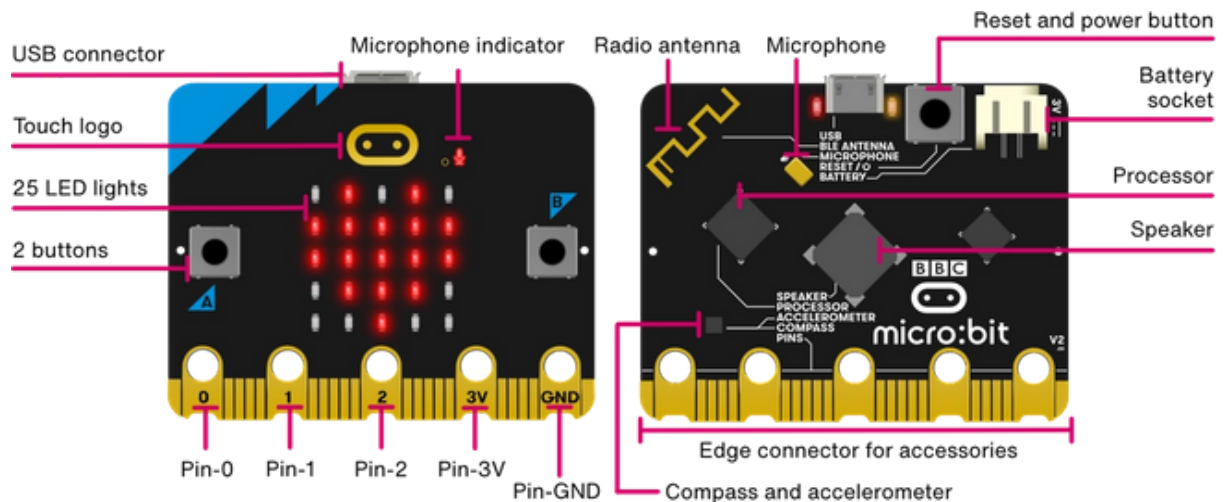
Série 0 - micro:bit

Du monde extérieur... vers le monde extérieur



- microcontrôleur programmable en microPython ou en blocs
- matériel pédagogique conçu par BBC learning
- capteurs et actionneurs intégrés.
 - capteurs intégrés:
 - 2 boutons
 - température
 - accéléromètre
 - chaîne humaine
 - luminosité
 - microphone
 - compas
 - récepteur radio micro:bit
 - actionneurs intégrés :
 - 25 leds (qui peuvent faire une sorte d'écran)
 - son (sur les versions 2)
 - émetteur radio micro:bit

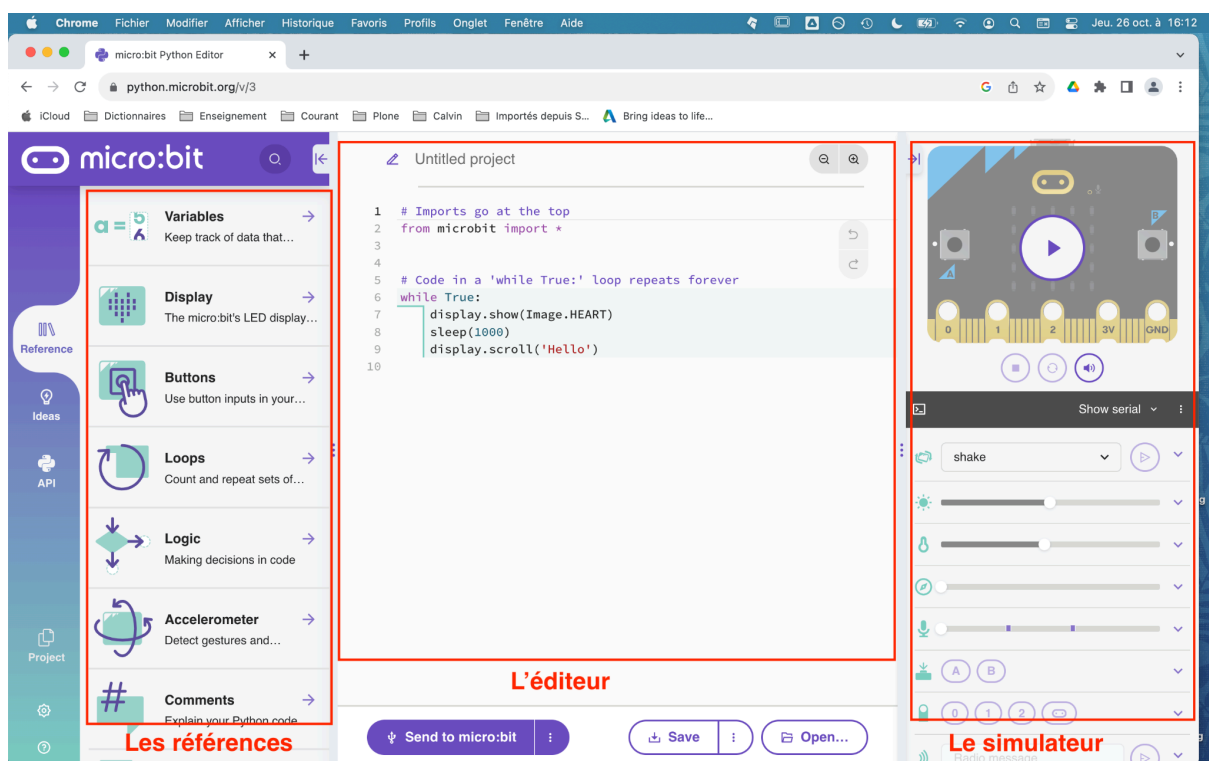




Il est possible de programmer la carte micro:bit avec le langage microPython avec le logiciel Thonny. Une documentation complète de ce langage avec cette carte se trouve sur : <https://microbit-micropython.readthedocs.io/fr/>

Il est aussi possible de passer par l'éditeur en ligne qui est très facile à prendre en main : <https://python.microbit.org/v/3>

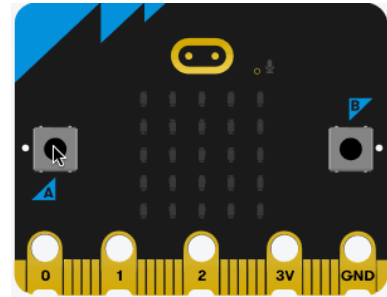
Pour téléverser vos programmes sur la carte, il vous est conseillé d'utiliser le navigateur Chrome. Cet éditeur en ligne contient trois espaces : les références, l'éditeur et le simulateur.



Exemple de code :

```
from microbit import *

while True:
    if button_a.was_pressed():
        #affichera un coeur
        display.show(Image.HEART)
        sleep(1000)
        display.clear()
```



Exercice 1 : (secoue_peu.py)

Programmez votre carte de sorte que lorsqu'elle est secouée, elle affiche un emoji 'énervé' pendant une seconde, sinon est affiché l'emoji 'content'.

Exercice 2 : (secoue_trop.py)

Programmez votre carte de sorte que lorsqu'elle est secouée pendant plus que 5 secondes, les leds s'éteignent toutes pendant 5 secondes. Sinon un emoji 'content' est affiché.

Exercice 3 : (jeu_decompte.py)

Créez un programme permettant d'afficher un décompte de 3 à 0.

Pendant ce temps, le joueur peut appuyer de manière répétitive sur le bouton A.

A la fin du décompte, si le joueur a appuyé plus que 10 fois sur le bouton, alors un cœur s'affiche, sinon un visage triste est affiché.

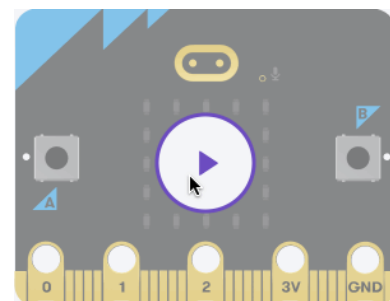
Exercice 4 : (jeu_plateau.py)

Programmez un jeu dans lequel un pixel se balade sur l'écran (sans en sortir) selon l'inclinaison de la carte. Si le pixel reste allumé pendant plus que deux secondes au centre de la carte (position = (2,2)), alors un cœur s'affiche.

Exercice 5 : (spaceInvaders.py)

Programmez le jeu du SpaceInvaders sur le micro:bit.

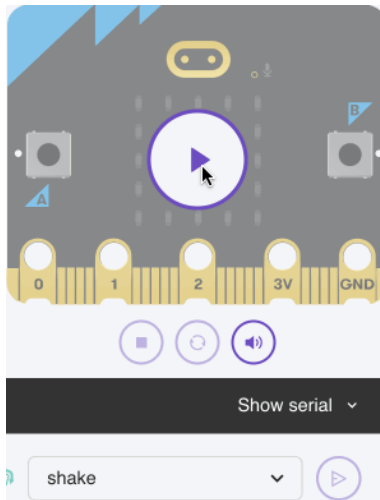
Des astéroïdes tombent à une position x aléatoire, vous devez les intercepter avec votre vaisseau situé sur la dernière ligne en vous déplaçant grâce aux boutons a et b. Lorsque 10 astéroïdes sont interceptées par le vaisseau le jeu s'arrête et l'emoji 'content' est affiché. Par contre, si 5 astéroïdes touchent le sol sans être interceptées, l'emoji 'triste' est affiché et le jeu s'arrête.



Exercice 6 : (perroquet.py)

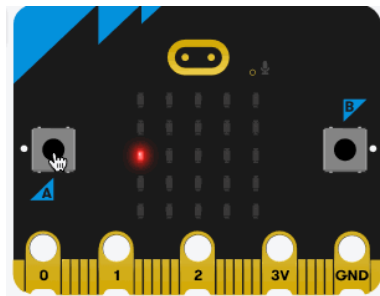
Lors d'une pression sur le bouton A (B), une led située à gauche (à droite) doit s'allumer brièvement. Lorsque la carte est secouée, la séquence doit être lue.

Exemple : (séquence : A,A,A,B,A,A)



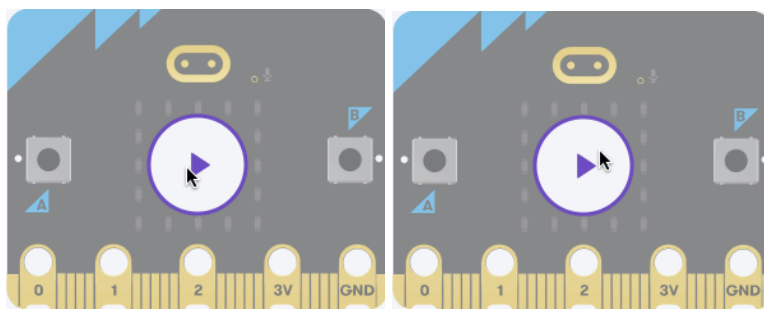
Exercice 7 : (digicode.py)

Lorsque l'on appuie successivement sur les boutons A-A-B-A-B, alors l'emoji 'content' est affiché.



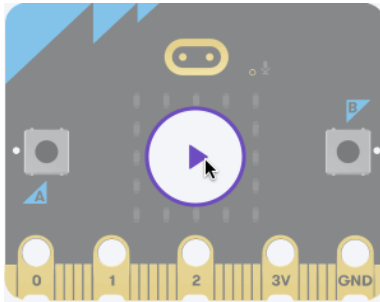
Exercice 8 : (jeu_Simon_v1.py + jeu_Simon_v2.py + jeu_Simon_v3.py)

version 1 : Au démarrage, la carte affiche une série de cinq clignotements aléatoires, soit du côté gauche, soit du côté droit. Cette série doit être rejouée par en appuyant sur les boutons A ou B. Si la série jouée correspond à celle donnée au départ, c'est gagné et l'emoji 'content' est affiché, sinon si la série est mal reproduite, l'emoji 'triste' est affiché à la 5ème pression.

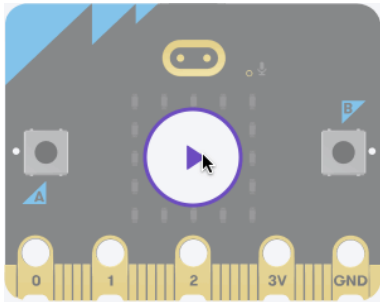


version 2 : Modifiez votre programme précédent de sorte que le jeu s'arrête dès qu'une erreur est commise.

Exemple :



version 3 : Programmez le jeu du Simon avec les deux boutons (A et B) et deux leds (situées à gauche et à droite). A chaque nouvelle étape du jeu, l'ancienne suite de leds est jouée avec une led supplémentaire (gauche ou droite). Le joueur doit rejouer cette suite en appuyant sur les boutons. Si la suite jouée est fausse le jeu s'arrête et le score est affiché, sinon la carte affiche un émoji 'content' et la suite à mémoriser augmente de 1.



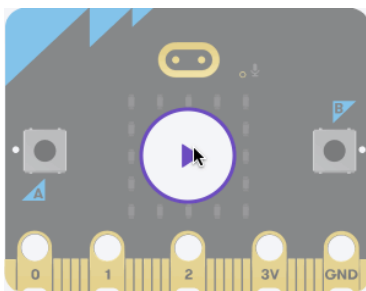
Exercice 9 : (allumettes_v1.py + allumettes_v2.py + allumettes_v3.py)

Les allumettes sont représentées par des leds allumées. Au départ, les 25 leds sont allumées.

version 1 : Créez un programme qui demande à l'utilisateur, via la console, le nombre d'allumettes qu'il souhaite enlever. Suite à sa réponse, le nombre restant d'allumettes est affiché via les leds.

version 2 : Modifiez votre programme pour que la question de la quantité enlevée soit posée à deux joueurs à tour de rôle, via les boutons A et B. Chaque appui sur le bouton enlève une allumette, un joueur ne peut enlever plus que trois allumettes. Celui qui prend la dernière allumette a gagné et voit sa lettre (A/B) affichée.

Exemple:



version 3 : Modifiez votre code pour que le joueur affronte l'ordinateur. Le passage du joueur à l'ordinateur se fait après 2 secondes d'attente après la dernière pression du joueur.

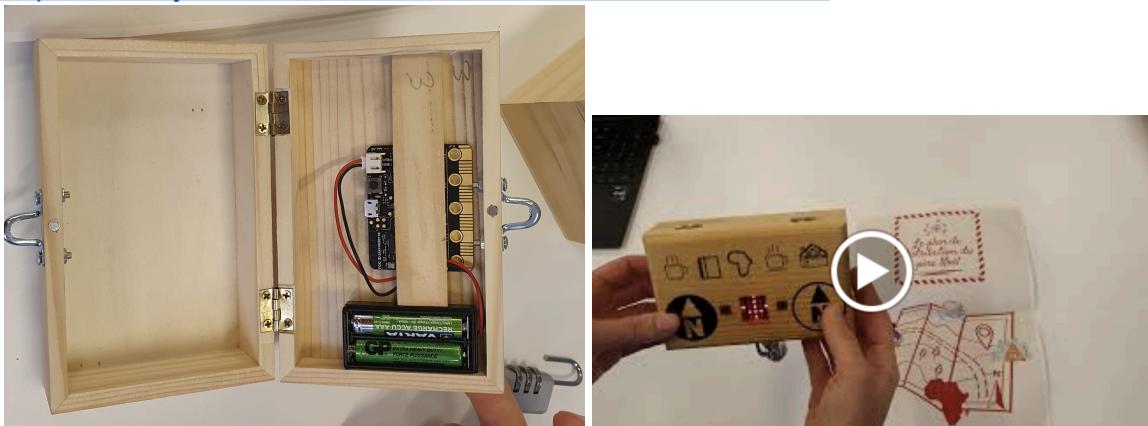
Exercice 10 : (capteur_alarme.py + recepteur_alarme.py)

Programmez un système d'alarme avec deux micro-bits. L'un est fixé sur une poignée de porte, de sorte que dès qu'un mouvement est détecté, une sonnerie retentit et un signal radio est envoyé. Le signal radio reçu par la deuxième carte enclenche également une sonnerie sur cette dernière. Pour stopper les deux sonneries, un appui sur le bouton de la deuxième carte est nécessaire.

Exercice 11 : (boite_orientation.py)

Programmez une Escape Box de sorte que le code du cadenas soit affiché une fois que la boîte ait été penchée successivement vers le haut, la droite, puis vers le bas.

<https://www.youtubeeducation.com/watch?v=dlhIS3zGndY>

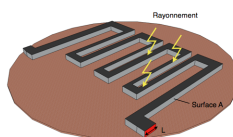
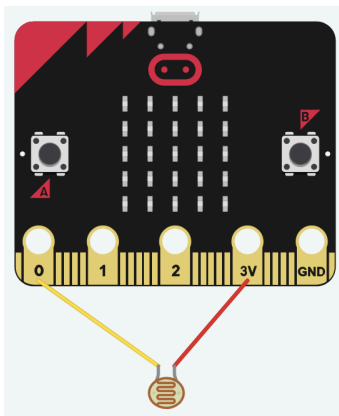


Exercice 12 : escape_lux.py

Programmez un système qui permette d'ouvrir automatiquement une boîte lorsque des capteurs sensibles à la lumière auront été correctement éclairés (par exemple, deux sous la lumière et un sans lumière).

Petite [démonstration ici](#).

Il faudra pour cela brancher trois [photorésistances](#) sur un micro-bit et avoir à proximité une [Eric'Scape](#) contrôlée par un message radio.



Note 1 : la pin sur laquelle est branchée la photorésistance devra être mise en 'pull-down'. Explications théoriques données en classe. La commande est :

```
pin0.set_pull(pin0.PULL_DOWN)  
print(pin0.read_analog())
```

Note 2 : au départ, vous pouvez commencer par approximer la valeur de la lumière ambiante (par exemple la médiane de 500 mesures). Cela permettra ensuite de comparer les valeurs lues des capteurs à cette valeur médiane pour déterminer s'ils sont significativement éclairés.

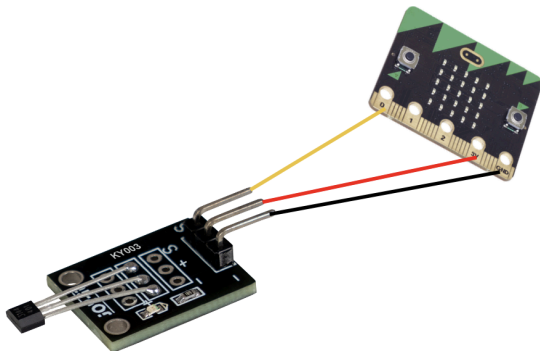
Exercice 13 : escape_magnet.py

Programmez un système qui permette d'ouvrir automatiquement une boîte lorsque des aimants auront été correctement placés selon un ordre défini sur des capteurs à effet Hall. (par exemple, d'abord l'aimant 1, puis l'aimant 2 et finalement le 3).

Petite [démonstration ici](#).

Il faudra pour cela brancher trois [capteurs](#) sur un micro-bit et avoir à proximité une [Eric'Scape](#) contrôlée par un message radio.

Note 1 : ces capteurs numériques étant "actifs", il faudra les brancher avec trois fils sur le micro-bit.



Note 2 : les plaquettes doivent être posées sur la boîte à des emplacements prévus. Il faut donc que les aimants se trouvant sous ces plaquettes soient décentrés et placer les capteurs alignés aux plaquettes souhaitées.

Note 3 : pour vérifier qu'un aimant vient d'être mis à proximité d'un capteur et enregistrer cette unique ponctuelle information, il faut tester deux mesures successives. Par exemple, l'information 00000011110000000000111111000 lue par le capteur, doit être interprétée comme une pièce posée, retirée puis remise à proximité du capteur.

Exercice 14 : chaine_humaine.py

Lorsque que le pin0 est relié au GND par une chaîne humaine, le micro-bit doit indiquer par des flashes lumineux un code. Par exemple, le code 1-2-3 est traduit par 1 flash, une pause, 2 flashes, une pause, 3 flashes, une pause.

Note 1 : commencer par traduire un flash en allumant les 25 leds du micro-bit un bref instant.

Note 2 : les flashes lumineux peuvent aussi être joués sur une guirlande grâce à un [transistor](#), un interrupteur électronique pour déclencher un circuit de maximum 12V. Cette guirlande peut-être contrôlée par un message radio.

