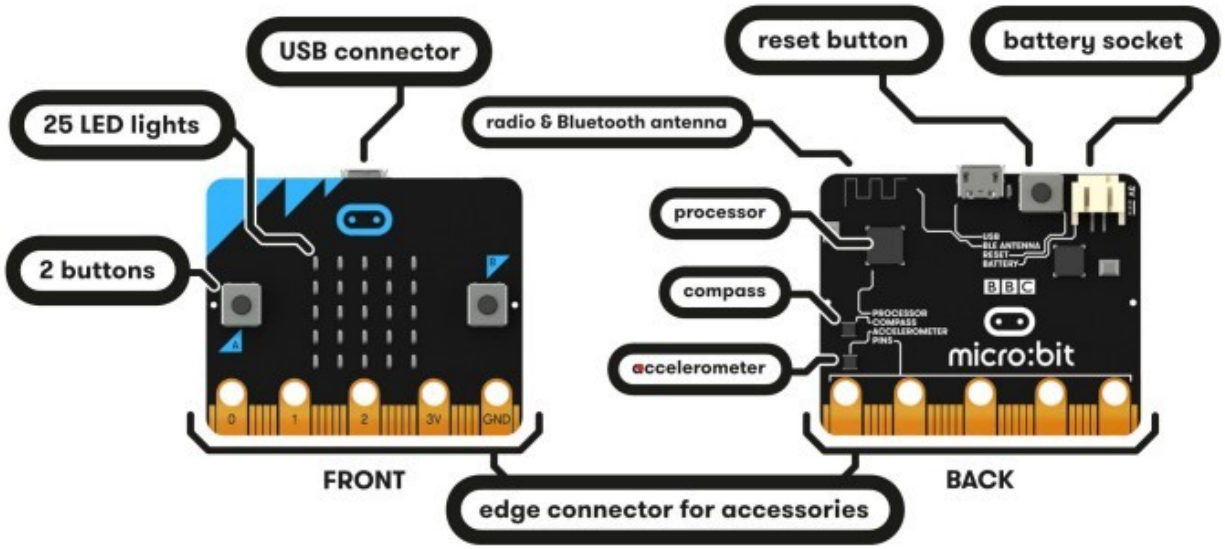
La carte Micro Bit

I. Qu'est ce que c'est ?

La carte micro:bit éditée par la BBC est un nano-ordinateur qui peut équiper un systéme informatique embarqué. Elle est munie d'un processeur ARM et de plusieurs capteurs et interfaces de connexion.
Le guide de présentation en ligne est disponible sur https://microbit.org/fr/guide/



Source : https://microbit.org/fr/guide/features/

La carte micro:bit dispose des spécificités techniques suivantes :

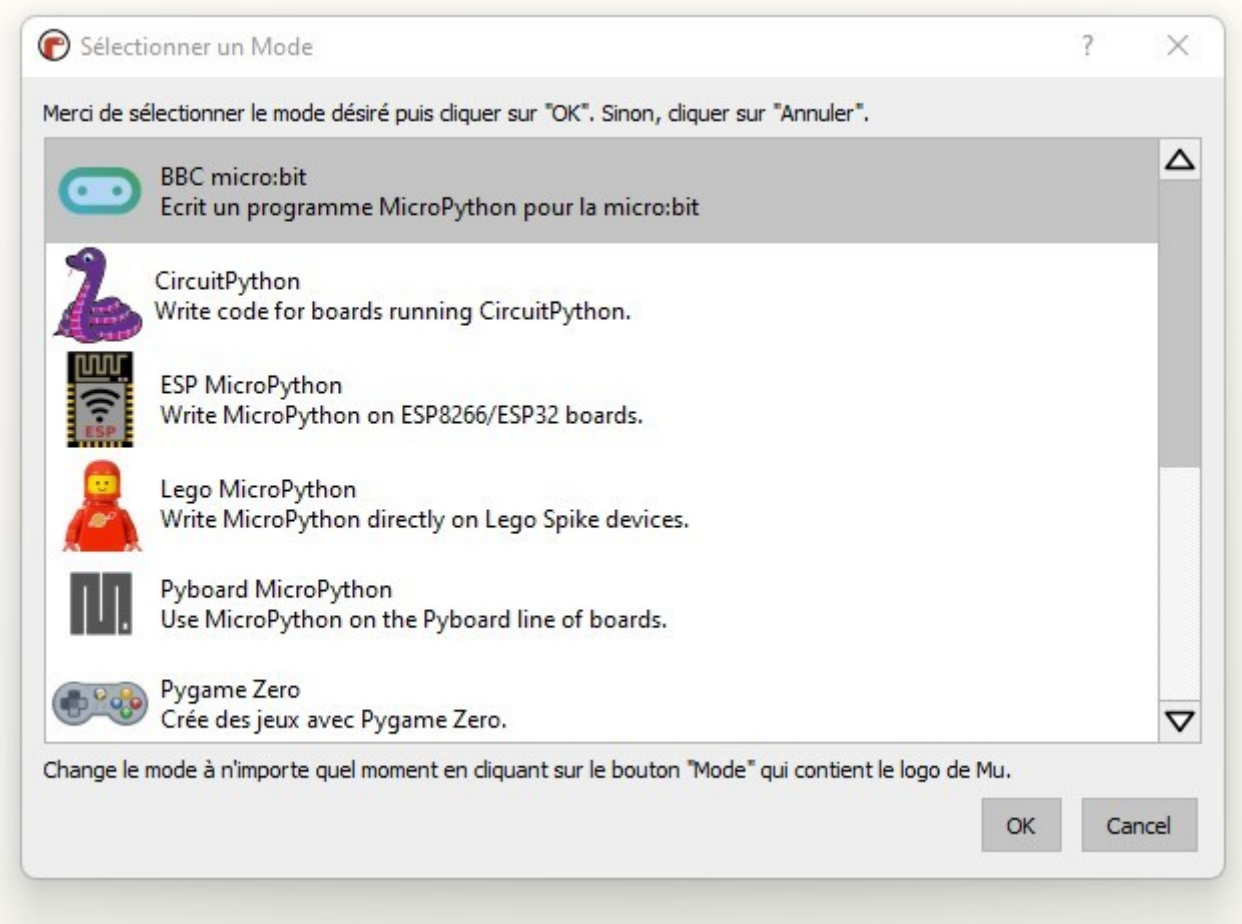
- 25 LEDs programmables individuellement
- 2 boutons programmables
- Broches de connexion
- Capteurs de lumière et de température
- Capteurs de mouvements (accéléromètre et boussole)
- Communication sans fil, via Radio et Bluetooth
- Interface USB

Nous utiliserons la carte en la connectant à un ordinateur avec le câble USB fourni qui assure la liaison de communication et l'alimentation.
Si on veut intégrer la carte dans un système embarqué, il est possible de la connecter à une alimentation externe par piles.
Lorsque la communication entre l'ordinateur et la carte échoue, on peut essayer de la redémarrer avec le bouton reset situé au verso.

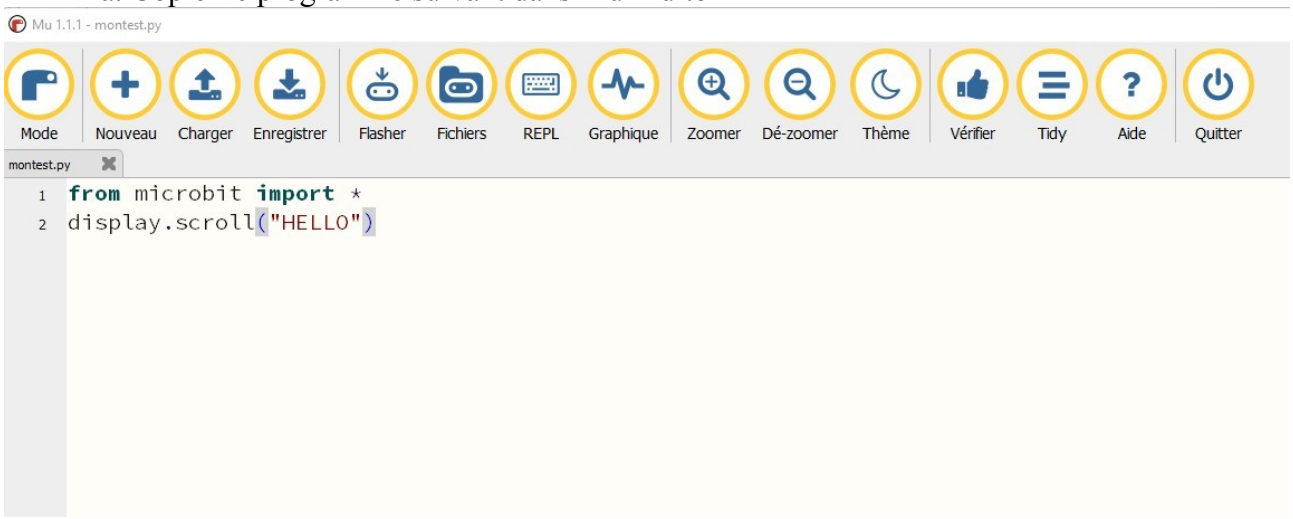
II. Pour afficher sur une carte micro:bit.

Brancher votre carte micro:bit sur un port USB.

1. Ouvrir Mu Editor et sélectionner le mode BBC micro:bit



2. Nous utiliserons la carte avec le langage Python et son module microbit
   a. Copier le programme suivant dans Mu Editor



   b. Dans votre répertoire personnel, créer un répertoire IHM et enregistrer le programme ci-dessus.
   Ensuite, copier le programme sur la carte (comme vous le feriez sur une clé USB).
   c. Pour transférer le programme sur la carte, cliquer sur Flasher. Lors de chaque téléversement, la mémoire Flash contenant le programme exécuté par la carte est réinitialisée.
   Si cela ne fonctionne pas, débrancher la carte microbit puis rebrancher là.
   d. Décrire l'effet du programme sur la carte. Une interaction est-elle possible ?

e. Modifier le programme afin qu'il affiche votre prénom.

3. a. Recopier le programme suivant :

```
1 from microbit import *
2 display.show(Image.HAPPY)
```

b. Exécutez le. Que se passe-t-il ?

c. Modifiez l'affichage sur la micro:bit.
Vous pouvez utiliser une des images intégrées :
Image.HEART Image.HEART_SMALL Image.HAPPY Image.SMILE Image.SAD Image.CONFUSED Image.ANGRY Image.ASLEEP Image.SURPRISED Image.SILLY Image.FABULOUS Image.MEH Image.YES Image.NO Image.CLOCK12, Image.CLOCK11, Image.CLOCK10, Image.CLOCK9, Image.CLOCK8, Image.CLOCK7, Image.CLOCK6, Image.CLOCK5, Image.CLOCK4, Image.CLOCK3, Image.CLOCK2, Image.CLOCK1 Image.ARROW_N, Image.ARROW_NE, Image.ARROW_E

4. Créer sa propre image.

Chaque pixel LED sur l'affichage physique peut prendre une parmi dix valeurs. Si un pixel prend la valeur 0 c'est qu'il est éteint. Littéralement, il a une luminosité de zéro.
En revanche, s'il prend la valeur 9 il est à la luminosité maximale. Les valeurs de 1 à 8 représentent des niveaux de luminosité entre éteint (0) et « au maximum » (9).

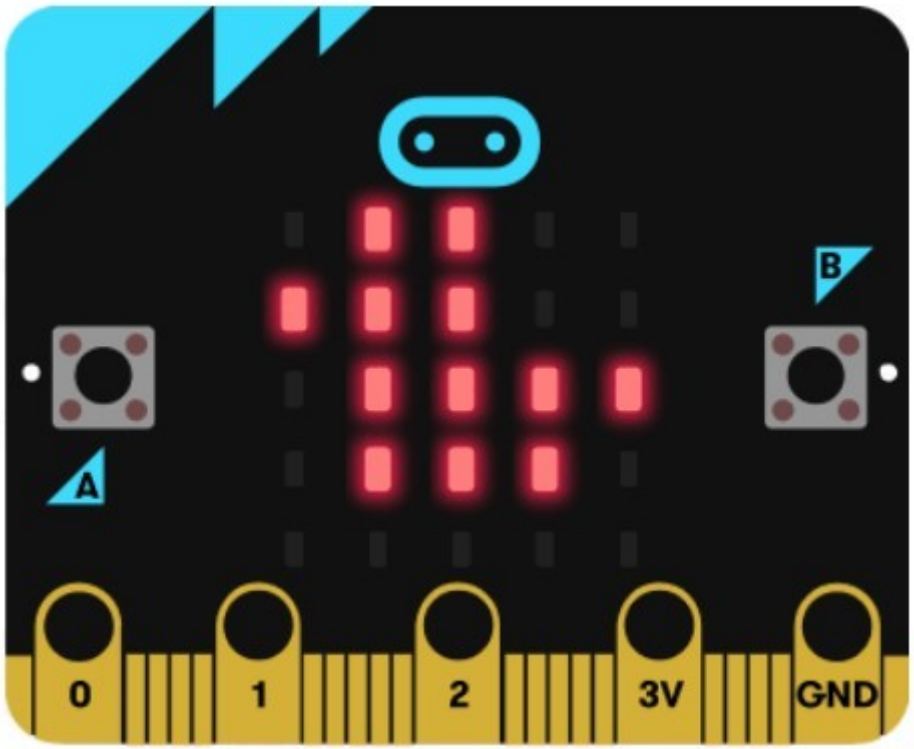
a. Recopier et exécuter le code suivant :

```
1 from microbit import *
2 bateau = Image("05050:"
3                "05050:"
4                "05050:"
5                "99999:"
6                "09990")
7
8 display.show(bateau)
```

Chaque ligne de l'affichage physique est représentée par une ligne de nombres se terminant par : et entourée de guillemets doubles " .
Chaque nombre indique une luminosité.
Il y a cinq lignes de cinq nombres donc il est possible de spécifier la luminosité individuelle de chacune des cinq LED sur chacune des cinq lignes sur l'affichage physique. C'est ainsi que l'on crée une image.

b. Modifier le code précédent afin d'afficher un bateau avec un seul mât.

c. Rédiger le programme qui affiche l'image suivante :



5. Afficher un pixel.
   a. Vous pouvez régler la luminosité des pixels de l'affichage individuellement de 0 (désactivé) à 9 (luminosité maximale). Pour des informations sur les coordonnées de l'affichage, voir le guide pour matrice à LED (https://microbit.org/guide/hardware/leds/) . Exécuter le programme suivant :

```
1 from microbit import *
2 display.set_pixel(1,4,9)
```

   b. Écrire le programme permettant d'allumer le pixel du centre de la matrice.

III. Boucle While.

1. Recopier et exécuter le programme suivant.

```
1 from microbit import *
2 while True:
3     display.set_pixel(2,2,9)
4     sleep(500)
5     display.clear()
6     sleep(500)
```

   a. Que se passe-t-il ?

   b. A quoi sert, à votre avis, la ligne 2 ?

   c. A quoi sert, à votre avis, la ligne 4 ?

d. A quoi sert, à votre avis, la ligne 6 ?

e. Modifier le programme pour que le pixel clignote plus rapidement.

2. Écrire un programme qui fait clignoter un cœur indéfiniment.
   Vous utiliserez image.HEART

3. Il est possible de combiner plusieurs images pour créer une animation.
Écrire un programme permettant d'afficher l'animation suivante : Microsoft MakeCode for micro:bit (microbit.org)

Remarque : il est évidement possible d'utiliser une boucle for ou un structure conditionnelle (avec il, elif, else).

IV. La programmation événementielle : utilisons les boutons.

La carte micro:bit possède deux boutons servant d'actionneur. C'est à dire qu'il est possible de modifier l'aspect de la carte en appuyant sur l'un des boutons et en programmant correctement la carte.

1. Recopier.

```
1 from microbit import *
2 while True:
3     display.scroll("SNT")
4     sleep(200)
5     if button_a.was_pressed():
6         break
```

2 .Exécuter le code sans appuyer sur le bouton A, puis en appuyant sur le bouton A. Que se passe-t-il ?

Exemples avec le bouton A:
button_a.is_pressed() : renvoie True si le bouton spécifié est actuellement enfoncé et False sinon.
button_a.was_pressed() : renvoie True ou False pour indiquer si le bouton a été appuyé depuis le démarrage de l'appareil ou la dernière fois que cette méthode a été appelée.

3. Écrire le programme permettant d'afficher l'image HAPPY si le bouton A est pressé, l'image SAD si le bouton est pressé, et l'image HEART sinon.

V. Capteur de luminosité.

En inversant les LEDs d'un écran pour devenir un point d'entrée, l'écran LED devient un capteur de lumière basique, permettant de détecter la luminosité ambiante.
La commande display.read_light_level() retourne un entier compris entre 0 et 255 représentant le niveau de lumière.

1. : Compléter le programme ci-dessous qui affiche une image de lune si on baisse la luminosité (en recouvrant la carte avec sa main par exemple) et un soleil sinon.

```
from microbit import *
soleil = Image("90909:"
 "09990:"
 "99999:"
 "09990:"
 "90909:")
lune = Image("00999:"
 "09990:"
 "09900:"
 "09990:"
 "00999:")
while True:
    if display.read_light_level()>"compléter ici":
        display.show(soleil)
        sleep(500)
    else:
        display.show("compléter ici")
        sleep(500)
```

2. créer un programme qui plus la luminosité sera élevée, plus il y aura de LEDs affichées sur la matrice.