

1. Objectifs

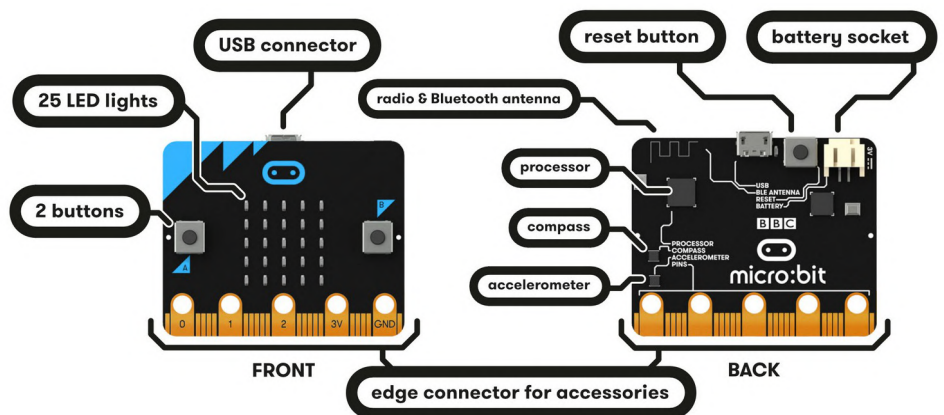
Écrire des programmes simples d'acquisition de données ou de commande d'un actionneur.

2. Contextualisation

La carte micro:bit, éditée par la BBC, est un microcontrôleur (microprocesseur avec mémoire et entrées/sorties). Elle est munie d'un processeur ARM et de plusieurs capteurs et interfaces de connexion. Il n'a donc pas de système d'exploitation mais permet d'exécuter des programmes.

Fonctionnalités incluses :

Capteurs de lumière, de température, broches de connexion, communication sans fil (Radio et Bluetooth), interface USB.



Programmation :

En Python (bibliothèque MicroPython), via l'éditeur en ligne : <https://python.microbit.org/v/3>

Nous utiliserons uniquement 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.

Prise en main :

```
1 from microbit import *
2
3 display.scroll("Hello World")
```

✂ À Faire 1 : Effectuer les actions suivantes :

1. Connecter la micro:bit à l'ordinateur via le câble USB,
2. Se rendre à l'adresse <https://python.microbit.org/v/3>
3. Copier le code ci-dessus dans l'interface Web et cliquer sur « Send to micro:bit ».
4. Décrire l'effet du programme sur la carte. Une interaction est-elle possible ?

5. Préciser le rôle de chaque instruction.

3. Manipulation de la matrice de LEDs

✂ À Faire 2 : Effectuer les actions suivantes :

```
1 from microbit import *
2
3 while True:
4     display.show("1")
5     sleep(500)
6     display.clear()
7     sleep(500)
```

1. Copier le code ci-dessus dans l'interface Web et cliquer sur « Send to micro:bit ».
2. Décrire l'effet du programme sur la carte. Une interaction est-elle possible ?

3. Préciser le rôle des instructions :

i. while True :

ii. display.show("1") :

iii. sleep(500) :

iv. display.clear() :

✂ À Faire 3 : Modifier le programme ci-dessous afin qu'il compte en boucle de 0 jusqu'à 9

```
1 from microbit import *
2
3 for i in range(10):
4     x = str(i)
5     display.show(x)
6     sleep(500)
7     display.clear()
8     sleep(500)
```

Indications :

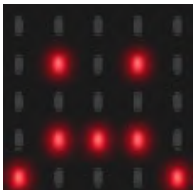
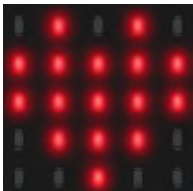
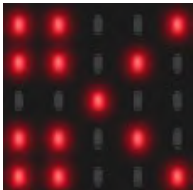
- La commande str(i) qui transforme le nombre i en texte.
- L'instruction for i in range(n): la variable i prend successivement la valeur de 0 à n - 1

✂ À Faire 4 : Effectuer les actions suivantes :

```
1 from microbit import *
2
3 emoji = Image("00000:09090:00000:90009:09990")
4 display.show(emoji)
```

1. Copier le code ci-dessus dans l'interface Web et cliquer sur « Send to micro:bit ».
2. A la lecture du code, comment allumer une led à une coordonnée x, y ?

✂ À Faire 5 : Modifier le programme ci-dessus afin d'afficher les images suivantes.

Image	Codage de l'image
 <p>A. Visage triste</p>	Image(" ")
 <p>B. Coeur</p>	Image(" ")
 <p>C. Ciseaux</p>	Image(" ")

✂ À Faire 6 : Modifier le programme ci-dessous afin qu'il allume les pixels de la ligne centrale

```
1 from microbit import *
2
3 for x in range(5):
4     display.set_pixel(x,0,9)
5     sleep(500)
```

Indications : La commande `display.set_pixel(x,y, z)` prend 3 entiers x, y et z en paramètre où :

- x : l'abscisse (colonne) du pixel concerné
- y : l'ordonnée (ligne) du pixel concerné
- z : l'intensité lumineuse du pixel concerné (0 : éteint, ..., 9 : allumé avec une intensité maximale)

✂ À Faire 7 : Modifier le programme du À Faire 4 afin qu'il allume tous les pixels, ligne par ligne.

```
1 from microbit import *
2
3
4
5
6
7
8
```

4. Synthèse

✂ À Faire 8 : Compléter le texte suivant avec les termes adéquats.

En considérant la carte micro:bit comme un système embarqué, la matrice de LEDs est un

Lorsque que la carte est branchée à un ordinateur, elle est alimentée en et il est possible de charger des qui seront exécutés tant que la carte est branchée.

Il existe 3 commandes pour manipuler la matrice de LED :

- `display.scroll(texte)` :
- `display.show(texte ou image)` :
- `display.set_pixel(x, y, z)` :