

5 dépôts sont à effectuer

TP noté sur /10**PARTIE 1 : PRÉSENTATION : CARTE BBC MICRO:BIT****> 1. HISTORIQUE**

La carte **BBC Micro:bit** est un « micro ordinateur de poche » (ou carte microcontrôleur) réalisé par la BBC en 2015. Initialement conçue pour permettre aux élèves du Royaume-Uni de s'initier dès l'âge de sept ans à l'algorithmique et à la programmation, elle est désormais accessible à tous. Ce projet prévoit d'offrir gratuitement un exemplaire à chaque écolier britannique de douze ans.

Cette carte peut être programmée à partir d'un ordinateur, d'un smartphone ou d'une tablette. Elle permet de s'initier à l'informatique embarquée, disposant nativement de nombreux capteurs et broches d'entrée-sortie, et possède la dernière technologie qui équipe les appareils modernes : téléphones mobiles, réfrigérateurs, montres intelligentes, alarmes antivol, robots, etc...



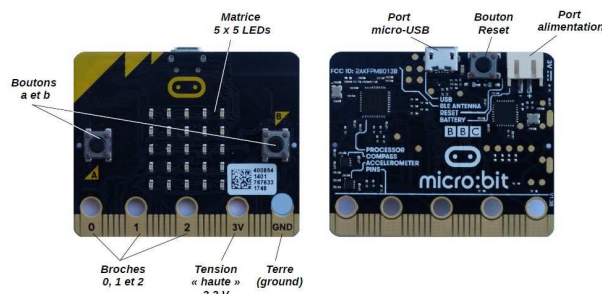
Ainsi, elle s'apparente à ce que l'on nomme l'**Internet des objets** : Internet of Things, abrégé IoT.

? 2. FONCTIONNEMENT

La carte BBC Micro:bit peut se programmer en utilisant plusieurs langages : Javascript (blocs ou texte), Scratch3, LISP, C++ avec l'environnement Arduino mais aussi **MicroPython**. Nous nous intéresserons dans cette séquence uniquement à la programmation de la carte sous MicroPython.

La carte Micro:bit dispose des spécificités techniques suivantes :

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

**! 3. A RETENIR**

Pour apprendre à programmer sur la carte Micro:bit, on utilisera :

- en cours, un exemplaire de la carte Micro:bit et le logiciel *Mu* pour programmer en MicroPython
- à la maison, l'*émulateur* en ligne (pour programmer et visualiser) : <https://create.withcode.uk>

Sur le logiciel *Mu*, une fois le programme écrit, il faut soit déposer le programme microPython (format .hex) sur la carte, soit plus simplement **Flasher** le programme sur la carte.

1. Chercher sur le web le prix moyen de vente d'un kit Micro:bit.

prix :

2. Ouvrir le logiciel *Mu*.

Au démarrage, choisir le mode BBC micro:bit.

Brancher la carte Micro:bit sur un port USB.

APPEL

→ Appeler le professeur pour vérification

! 1. À RETENIR

Il y a une matrice de 25 LED (diodes électroluminescentes) sur la face avant de la carte. Plusieurs commandes d'affichage sont disponibles en microPython :

- `display.show()` : affiche le caractère ou l'image choisi(e) entre parenthèses
- `display.scroll(string, delay=400)` : affiche une chaîne de caractères (*string*, du texte) en défilement avec une certaine vitesse (*delay*, plus le délai est grand, moins le texte défilera rapidement)
- `display.set_pixel(x, y, val)` : allume le pixel de coordonnées *x* et *y* (de 0 à 4 en abscisse et ordonnée) d'une intensité *val* (entre 0 et 9)
- `sleep()` : provoque la pause de la carte pendant un nombre défini de millisecondes choisi entre parenthèses
- `display.clear()` : efface l'affichage en cours

E 2. EXERCICE 1 : LISTE DÉROULANTE

```
from microbit import *
display.scroll("BIENVENUE")
```

1. Dans le logiciel *Mu*, écrire le programme ci-dessus puis flasher le programme sur la carte.
2. Modifier le programme pour que défile le texte OBJETS CONNECTES assez lentement.

E 3. EXERCICE 2 : IMAGES

```
from microbit import *
display.show(Image.SAD)
```

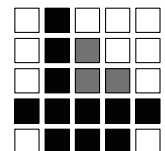
1. Dans le logiciel *Mu*, écrire le programme ci-dessus puis flasher le programme sur la carte. Observer. Il s'agit d'une des images intégrées contenues dans la bibliothèque MicroPython.
2. On constate que la carte est un peu triste ... En s'aidant de la liste des images intégrées disponible sur <https://snt.entraide-ella.fr/> dans le chapitre **Objets Connectés**, modifier le programme précédent pour lui redonner de la joie !

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 (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 intermédiaires.

3. Le programme suivant crée et affiche l'image d'un bateau.

```
1 from microbit import *
2
3 bateau=Image("09000:"
4             "09500:"
5             "09550:"
6             "99999:"
7             "09990:")
8
9 display.show(bateau)
```



Tester ce programme sur votre carte.

4. En s'inspirant du programme précédent, créer un programme qui affiche une image de sapin.
5. Enregistrer ce programme sous le nom **sapin.py**

DEPÔT 1

déposer sapin.py sur <https://entraide-ella.fr>

E 4. EXERCICE 3 : CŒUR CLIGNOTANT

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

1. Tester le programme ci-dessus. Que fait-il? **réponse :**
2. Le modifier pour allumer la LED au centre de la matrice.
3. En s'aidant de la liste d'images intégrées, écrire un programme qui fait clignoter un coeur indéfiniment.

E 5. EXERCICE 4 : COMPTE À REBOURS

1. Saisir le programme suivant.

```
1 from microbit import *
2
3 chiffre = 0
4 for k in range(4):
5     display.show(chiffre)
6     chiffre = chiffre+2
7     sleep(500)
```

2. Modifier le programme précédent afin qu'il effectue le compte à rebours de 9 jusqu'à 0.
3. Enregistrer ce programme sous le nom **chrono.py**

DEPÔT 2

déposer chrono.py sur <https://entraide-ella.fr>

E 6. EXERCICE 5 : UN PEU D'ALÉATOIRE

```
from microbit import *
import random

x=random.randint(0,4)
y=random.randint(0,4)
display.set_pixel(x, y, 9)
```

1. Tester le programme précédent plusieurs fois de suite.
Indication : Pour cela, il faut redémarrer le programme en appuyant sur le bouton RESET situé à l'arrière de la carte.
2. Écrire un programme qui allume successivement et indéfiniment 10 pixels au hasard à l'écran.
Indication : On pourra utiliser une boucle **for** comme à l'exercice 4.
3. Enregistrer ce programme sous le nom **alea.py**

DEPÔT 3

déposer alea.py sur <https://entraide-ella.fr>

E 7. EXERCICE 6 : COURSE DE PIXELS

```
1 from microbit import *
2
3 while True:
4     for i in range(0,5):
5         display.set_pixel(i,0,9)
6         sleep(200)
7     display.clear()
```

1. Tester ce programme.
2. Que fait-il? **réponse :**
3. **[bonus]** En s'inspirant du programme précédent, écrire un programme qui fait défiler un pixel sur tout l'écran de bas en haut.



1. A RETENIR

Il y a **deux boutons A et B** sur la face avant de la carte Micro:bit.

On peut détecter quand ces boutons sont pressés, ce qui permet de déclencher des instructions sur l'appareil.

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



2. EXERCICE 7 : BADGE

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

1. Tester ce programme. Que fait-il? **Réponse :**
2. En s'inspirant du programme précédent, réaliser votre badge numérique :
 - l'appui sur le bouton A affiche votre nom
 - l'appui sur le bouton B affiche votre prénom
 - l'appui simultané sur les bouton A et B affiche votre classe
3. Enregistrer ce programme sous le nom **badge.py**



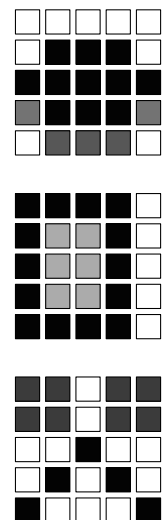
déposer badge.py sur <https://entraide-ella.fr>



3. EXERCICE 8 : PIERRE, FEUILLE, CISEAUX

```
from microbit import *

pierre = ...
feuille = ...
ciseaux = ...
while True:
    if ...:
        display.show(ciseaux)
        ...
    elif ...:
        display.show(pierre)
        ...
    elif ...:
        display.show(feuille)
        ...
    display.clear()
    sleep(100)
```



1. Écrire ce programme sur *Mu* puis créer les images pour la pierre, la feuille et les ciseaux.
2. Compléter ce programme afin qu'il affiche pendant 1 seconde des ciseaux si A et B sont pressés simultanément, une pierre si A est pressé et une feuille si B est pressé.
3. Enregistrer ce programme sous le nom **pfc.py**



déposer pfc.py sur <https://entraide-ella.fr>

