SNT – ELLA

OBJETS CONNECTÉS

SÉANCE 1 - PREMIERS PAS AVEC LA CARTE MICRO:BIT
PRÉSENTATION ET AFFICHAGES LED

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...

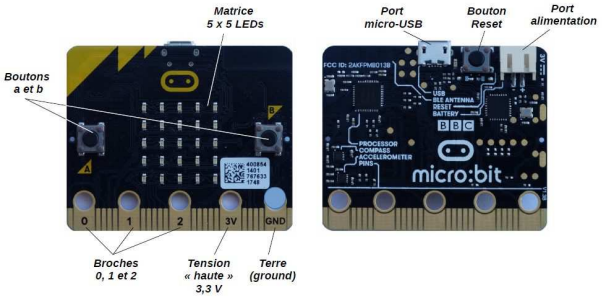
Ainsi, elle s'apparente à ce que l'on nomme l'**Internet des objets** : Internet of Things, abrégé IoT.

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.
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. A 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

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.

3. EXERCICE 2 : IMAGES

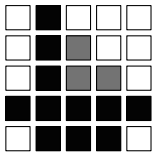
```
from microbit import *

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/ 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-mb.py**

DEPÔT **sapin-mb.py sur votre zone SNT de http://entraide-ella.fr**

4. Exercice 3 : Coeur clignotant

```
from microbit import *

while True:
    display.set_pixel(1, 4, 9)
    sleep(500)
    display.clear()
    sleep(500)
```

1. Tester le programme ci-dessus. Que fait-il ?
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.

5. Exercice 4 : Compte à rebours

1. Compléter le programme ci-dessous afin qu'il effectue le compte à rebours de 9 jusqu'à 1.
   Indication : Utiliser la commande `str(i)` qui transforme le nombre i en texte.
   Par exemple, str(5) renvoie "5".

```
from microbit import *

chiffre = ...
for i in range(1, ...  ):
  display.show( ..... )
  chiffre = ....
  sleep(500)
```

2. Modifier ce programme pour que défile "PARTEZ !" à la fin du compte à rebours.

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**

7. Exercice 6 : Course de pixels

```
from microbit import *

while True:
    for i in range(0,5):
        display.set_pixel(i,i,9)
        sleep(200)
        display.clear()
```

1. Tester ce programme. Que fait-il ?
2. En s'inspirant du programme précédent, écrire un programme qui fait rebondir un pixel comme une balle de tennis.
3. Sauvegarder ce programme sous le nom **balle-mb.py**

DEPÔT **balle-mb.py sur votre zone SNT de http://entraide-ella.fr**