



Lycée Joliot Curie à 7	<i>SNT – Séance n°5</i>	Classe de Seconde
Informatique embarquée et objets connectés		Classe 2° ... Nom : Nom :

Que sont les objets connectés ...” Vidéo 1

Résumé de la vidéo 1 du chapitre

.....

.....

.....

.....

.....

.....

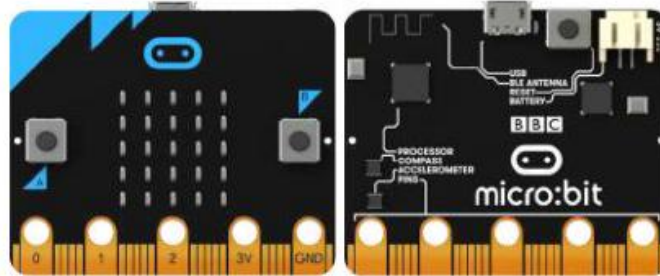
.....

.....

.....

Présentation de la carte BBC micro:bit

BBC micro:bit est une carte à microcontrôleur conçue en 2015 au Royaume-Uni pour développer l'apprentissage de l'algorithmique et de la programmation. Pourvu de capteurs et d'actionneurs, ce petit ordinateur possède la dernière technologie qui équipe les appareils modernes : téléphones mobiles, réfrigérateurs, montres intelligentes, alarmes antivols, robots, etc...
Ainsi, il s'apparente à ce que l'on nomme l' **Internet des objets** : **Internet of Things**, abrégé **IoT**.



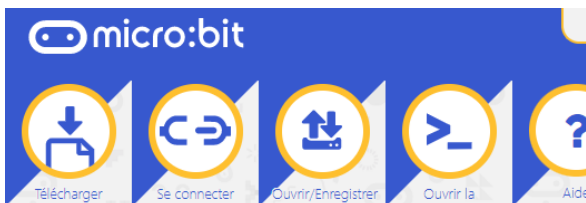
La carte micro:bit dispose des spécificités techniques (<https://microbit.org/fr/guide/features/>) suivantes :

- 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

Partie A : Découverte des fonctionnalités de la carte BBC micro:bit

1-Toute 1ère connexion:

- Brancher la carte sur un port USB.
- Sur Capneuronal chapitre 5, ouvrir l'éditeur "Python - Micro:bit V2"



- Cliquez sur “se connecter” et sélectionner la carte
- Cliquez sur “flash”

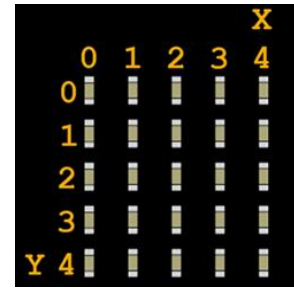
Le programme écrit en Python se “téléverse” sur la carte et s'exécute.

A chaque fois que vous changerez le programme, il faudra le téléverser sur la carte.

2- Commandes de base de l'afficheur, matrice de 5x5 LEDs

- Copier le programme ci-contre dans l'éditeur

```
from microbit import *
while True:
    display.scroll("Hello, World!")
    display.show(Image.HEART)
    sleep(2000)
```



Matrice 5 x 5

While : tant que ...

a- Afficher un texte "défilant" `display.scroll(string, delay=400)`

Nous allons commencer par afficher quelques informations sur l'afficheur.

- La première ligne de ce programme importe la bibliothèque de fonctions `micro:bit`.

- La deuxième ligne fait défiler un message à l'écran. **Cela n'arrive qu'une seule fois.**

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

Exercice 1: Modifier le programme précédent pour qu'il affiche le texte de ton choix.

La vitesse de défilement peut être ralentie ou accélérée à l'aide du paramètre `delay` . Plus le nombre est grand, plus le défilement est lent.

```
from microbit import *
display.scroll("          ", delay=... )
```

Exercice 2 : Modifier votre programme pour que l'affichage ne s'arrête jamais !

```
from microbit import *
...
display.scroll("          ", delay=... )
```

b- Afficher une "image" `display.show(image)`

Exercice 3 : On constate que la carte est un peu triste. Modifier le programme précédent pour lui redonner de la joie.

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

Aide: ci-dessous la liste 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, Image.ARROW_SE,
Image.ARROW_S, Image.ARROW_SW, Image.ARROW_W, Image.ARROW_NW Image.TRIANGLE
Image.TRIANGLE_LEFT Image.CHESSBOARD Image.DIAMOND Image.DIAMOND_SMALL Image.SQUARE
Image.SQUARE_SMALL Image.RABBIT Image.COW Image.MUSIC_CROCHET Image.MUSIC_QUAVER
Image.MUSIC_QUAVERS Image.PITCHFORK Image.XMAS Image.PACMAN Image.TARGET Image.TSHIRT
Image.ROLLERSKATE Image.DUCK Image.HOUSE Image.TORTOISE Image.BUTTERFLY
Image.STICKFIGURE Image.GHOST Image.SWORD Image.GIRAFFE Image.SKULL Image.UMBRELLA
Image.SNAKE

Essayer plusieurs images intégrées.

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

c- 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).

```
from microbit import *
bateau = Image( "05050:"
                "05050:"
                "05050:"
                "99999:"
                "09990" )
display.show(bateau)
```


Dans le programme suivant que vous exécuterez, on importe le module **random** de MicroPython et on l'utilise pour afficher un pixel au hasard sur la matrice.

Exécuter le programme ci-contre:

Tester le programme précédent plusieurs fois de suite. Pour cela, redémarrer la micro:bit en appuyant sur le bouton RESET situé à l'arrière de la carte.

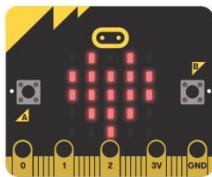
```
from microbit import *
import random
n=random.randint( 0 , 4 )
p=random.randint( 0 , 4 )
display.set_pixel(n, p, 9 )
```

Exercice 7: Ecrire un programme ci-dessous qui allume successivement et indéfiniment des pixels au hasard à l'écran (aide: utiliser une boucle infinie).

```
from microbit import *
import random
```

Exercice 8:

Ecrire ci-dessous un programme qui fait clignoter un cœur indéfiniment (voir illustration ci-dessous).



```
from microbit import *
import random
```

4- Boucle for

Le programme suivant utilise une boucle for pour faire défiler un pixel sur une ligne. Exécutez-le.

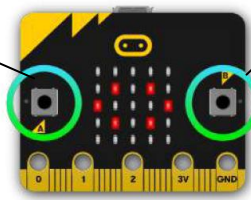
```
from microbit import *
while True :
    for i in range ( 5 ):
        display.set_pixel(i, 0 , 9 )
        sleep( 200 )
    display.clear()
```

Exercice 10 (la double boucle) : S'inspirer du programme précédent pour réaliser un programme ci-dessous qui fait défiler un pixel sur tout l'écran.

```
from microbit import *
while True :
```

5- Les entrées boutons A, B et A+B – programmation

Bouton A



Bouton B

Il y a deux boutons sur la face avant du micro:bit (étiquetés A et B). On peut détecter quand ces boutons sont pressés, ce qui permet de déclencher des instructions sur l'appareil.

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.

Exemple : Essayer le programme suivant qui fait défiler le texte "SNT" indéfiniment. On introduit l'instruction conditionnelle *if* qui va tester si le bouton A a été pressé (pendant le défilement du texte ou pendant la pause), auquel cas le programme s'arrête en exécutant la commande `break`.

La touche *Reset* permet de relancer le programme.

```
from microbit import *
while True :
    display.scroll( "SNT" )
    sleep( 200 )
    if button_a.was_pressed():
        break
display.show(Image.SAD)
```

Exercice 11: Pierre feuille ciseaux!

Compléter le programme suivant dans lequel

- une pression simultanée sur les boutons A et B affichera une image de ciseaux.
- Sinon si, une pression sur le bouton A affichera une image de pierre.
- Sinon si, une pression sur le bouton B affichera une image de feuille.

Il faudra créer vous-même l'image de la pierre et de la feuille avec un temps d'affichage de 0,5 seconde.

6- Capteur de lumière

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.

Plus l'entier est élevé et l'intensité lumineuse est élevée.

```
from microbit import *
pierre = .....

feuille = .....

ciseaux = Image( "99009:"
    "99090:"
    "00900:"
    "99090:"
    "99009:")

while True:
    if button_a.is_pressed() and button_b.is_pressed():
        display.show(ciseaux)
        sleep( ... )
    elif button_a.is_pressed():
        display.show(.....)
        sleep( ... )
    elif button_..._is_pressed():
        display.show(.....)
        sleep( ... )
    .....
    sleep( 100 )
```

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

Exercice 12: Compléter le programme ci-contre qui affiche une image de lune si on baisse la luminosité (en recouvrant la carte avec sa main par exemple) et un soleil sinon.

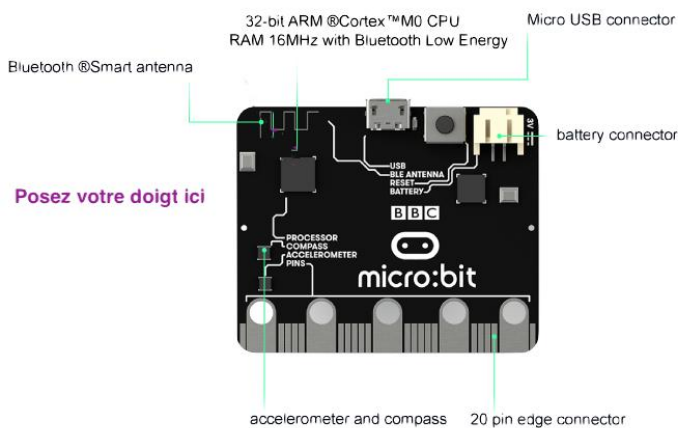
Exercice 13: créer un programme qui affiche le niveau de luminosité et le tester avec la LED d'un téléphone portable ou une lampe-torche par exemple. Plus la luminosité sera élevée, plus il y aura de LEDs affichées sur la matrice.

1					15				
2					16				
3					18				
4					19				
5					20				
6									
8									
9									
10									
11									
12									
13									
14									

6- Capteur en température

Le micro:bit n'a pas un capteur de température dédié. Au lieu de cela, la température fournie est en fait la température de la puce de silicium du processeur principal. Comme le processeur chauffe peu en fonctionnement, sa température est une bonne approximation de la température ambiante. L'instruction *temperature()* renvoie la température de la carte micro:bit en degrés Celsius.

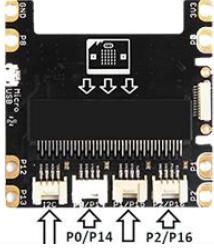
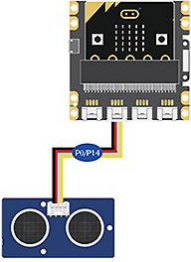
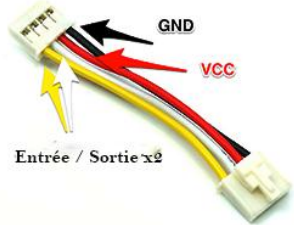
Exercice 13: Ecrire un programme qui affiche la température (aide: on pourra utiliser l'instruction `display.scroll()` ;



1				
2				
3				
4				
5				

Les défis !

Le Shield (bouclier) est parmi nous !

		
<p>A quoi sert le Shield Grove pour microbit BBC ?</p> <p>C'est une interface qui permet de connecter des capteurs et actionneurs grove à la carte Microbit</p>	<p>Les capteurs et actionneurs disponibles se connectent sur la broche P0/P14</p> <ul style="list-style-type: none"> - Ruban led - Afficheur digital - servomoteur – Haut parleur 	<p>Les cordons GROVE sont toujours constitués de 4 fils</p>

Défi 1 : le ruban à Leds

```

from microbit import *
import neopixel
from random import randint

np = neopixel.NeoPixel(pin0, 8)
# 30 LEDS connectées sur la pin0

while True:
    # Pour chaque NeoPixel du stick, de l'adresse_pixel = 0 à 7
    for adresse_pixel in range(0, 8):
        # intensité aléatoire entre 0-60 par canaux RVB
        # une intensité aléatoire entre 0-255 est trop forte
        r = randint(0, 60)
        v = randint(0, 60)
        b = randint(0, 60)

        # Affecte le tuple (r, v, b) au NeoPixel
        np[adresse_pixel] = (r, v, b)

        # Envoi des données sur le stick Neopixel
        np.show()
        sleep(100)
    
```

Après avoir connecté votre Micro :bit V2 sur un Shield Grove, brancher le ruban de 30 LEDs en P0/P14.

- Faire fonctionner le programme ci-contre.
- Modifier le programme pour allumer toutes les Leds du ruban.

Défis :

- Modifier le programme afin d'allumer les 2 Leds puis faites les se rejoindre au centre ...

- A vous de donner libre court à votre imagination.

1				15			
2				16			
3				18			
4				19			
5				20			
6							
8							
9							
10							
11							
12							
13							
14							

Défi 1 : Haut parleur



```
from microbit import *
import music

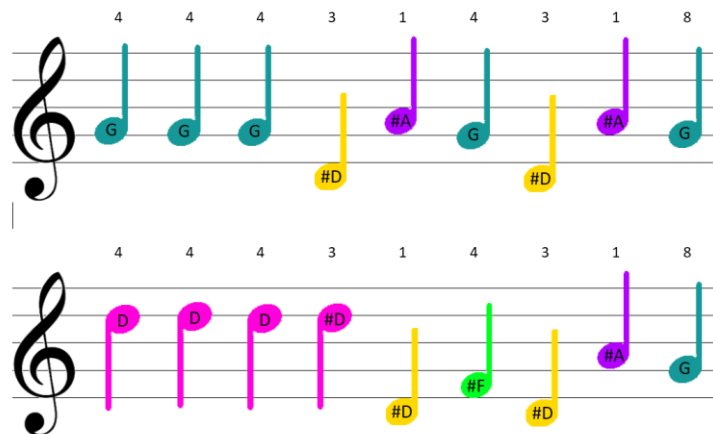
while True:
    if button_a.is_pressed():
        tune = ["g4:4", "g4:4", "g4:4", "d#4:3", ...
            music.play(tune)
```

Après avoir connecté votre Micro :bit V2 sur un Shield Grove, brancher le haut parleur

Une **note de musique** est décrite par sa hauteur et sa durée. Nous utiliserons le système anglo-saxon pour décrire la hauteur d'une note. La description de la hauteur contient un caractère entre A et G, et un numéro d'octave.



- Modifier le programme pour jouer la musique de star War
Les premières notes de Star Wars



Défis :

- Jouer « Au claire de lune »

1				15			
2				16			
3				18			
4				19			
5				20			
6							
8							
9							
10							
11							
12							
13							
14							