Python et micro:bit - Plan de travail

1NSI – septembre 2022

Contenus

- Périphériques d'entrée et de sortie Interface Homme-Machine (IHM)
- Language et programmation : constructions élémentaires

Capacités attendues

- Identifier le rôle des capteurs et actionneurs.
- Réaliser par programmation une IHM répondant à un cahier des charges donné.
- Maitriser les bases du language Python

1 Prise en main de la carte micro:bit

☐ Exercice 1 : Découverte de la carte Micro:bit ..... ☆☆☆☆☆

2 Programmation linéaire

☐ Exercice 2 : Matrice Led ..... ☆☆☆☆☆
☐ Exercice 3 : Quelques capteurs ..... ☆☆☆☆☆

3 Boucle avec condition

☐ Exercice 4 : Boucles infinies ..... ☆☆☆☆☆
☐ Exercice 5 : Boucles finies avec condition ..... ☆☆☆☆☆

4 Boucle finie

☐ Exercice 6 : Boucle finie ..... ☆☆☆☆☆

5 Conditions

☐ Exercice 7 : Capteurs ..... ☆☆☆☆☆
☐ Exercice 8 : Plus ou moins ..... ☆☆☆☆☆

6 Mini projet

☐ Exercice 9 : Queulorior ..... ☆☆☆☆☆
☐ Exercice 10 : Snake ..... ☆☆☆☆☆

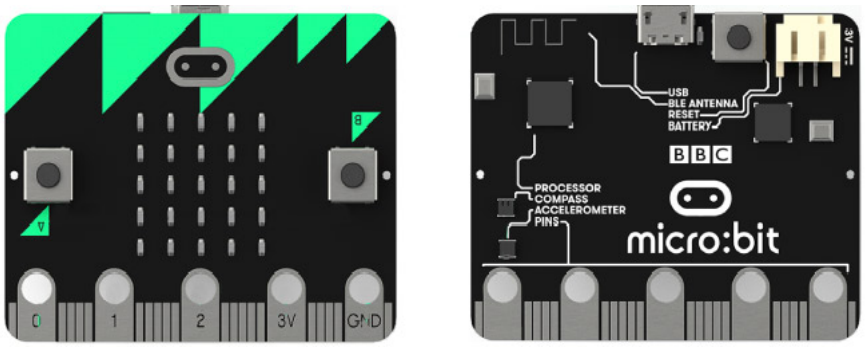
7 Pure Python

Cet exercice se fait sans utilisation du module micro:bit.

☐ Exercice 11 : Pure Python ..... ☆☆☆☆☆

Exercice 1 — Découverte de la carte Micro:bit

1. Observer la carte micro:bit puis identifier les éléments présents en les reportant sur le schéma ci-dessous



2. Classer ces éléments dans l'une des trois catégories suivantes : capteur, actionneur, autre. Imaginer une utilité possible de ces éléments.
3. Trouver la documentation officielle (en français) de la programmation de cette carte en python (micropython ici).

Exercice 2 — Matrice Led

Dans cette exercice, vous apprendrez à utiliser la matrice de led.

Page de la documentation officiel traitant des images : `https://microbit-micropython.readthedocs.io/fr/latest/tutorials/images.html`

La documentation nous invite à essayer ce premier programme

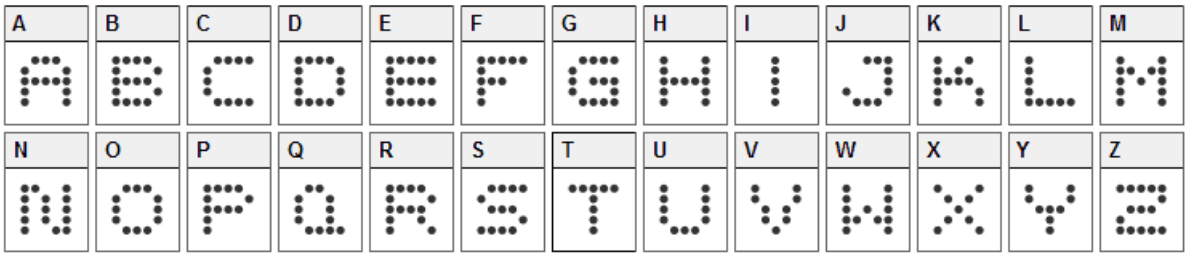
```
from microbit import *
display.show(Image.HAPPY)
```

1. Écrire le programme proposé et le déposer sur la carte pour voir le dessin affiché.
2. Affichage d'images pré-dessinées
   (a) En vous basant sur la documentation, modifier le programme précédent pour faire afficher d'autres images.
   (b) Pour afficher plusieurs image les unes à la suite des autres, il faut faire une pause entre.

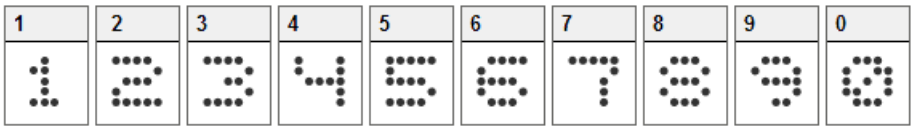
```
from microbit import *
import time

display.show(Image.HAPPY)
time.sleep(1)
display.show(Image.SAD)
```

   Modifier le programme précédent pour faire tourner une aiguille comme une montre.
3. Affichage d'images personnelles
   (a) Lire la suite de la documentation pour afficher des images personnelles.
   (b) Reproduire quelques caractères du tableau suivant

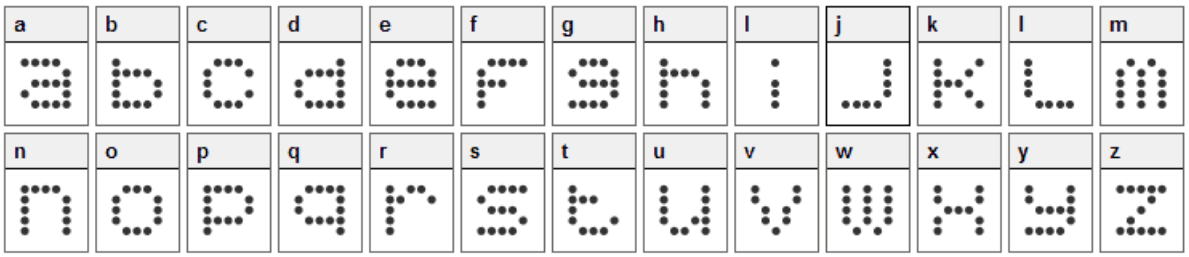
Uppercase Letters



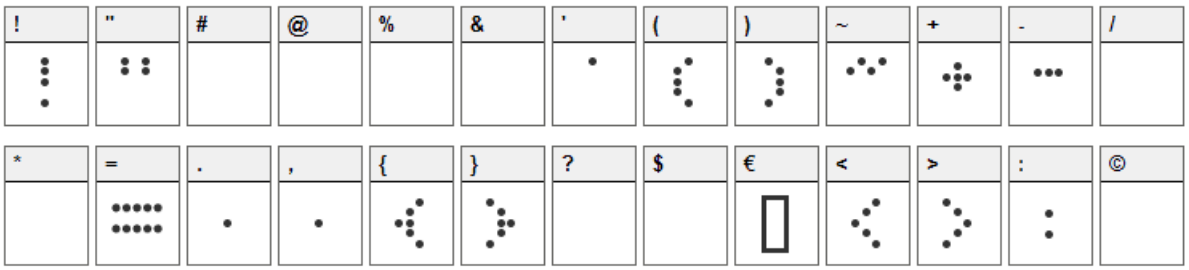
Numbers



Lowercase Letters



Special Characters



4. Affichage de texte : La méthode `display.scroll(....)` permet de faire défilé du texte. Faire défiler votre nom sur la matrice.

Exercice 3 — Quelques capteurs

Comme sur dans le premier exercice, la carte micro:bit est composé de quelques capteurs.

1. Pour accéder au capteur de température, il faut utiliser la commande `temperature()`. Écrire un programme qui fait défiler la température.
2. Pour accéder au capteur de la boussole, il faut utiliser la commande `compass.heading()`. Écrire une programme qui fait défiler la direction de la boussole.
3. Écrire un programme qui affiche une image, puis la température et enfin la direction de la boussole.
4. Critiquer les deux qui viennent d'être écrits. Pourquoi ne sont-ils pas satisfaisants ?

Exercice 4 — Boucles infinies

Pour mettre à jour de façon continue une valeur, il faut utiliser une **boucle infinie** pour mesurer la valeur puis éventuellement l'afficher.

```
from microbit import *
import time

while True:
    temp = temperature()
    display.scroll(temp)
    time.sleep(1)
```

1. Décrire ce que fait chaque ligne du programme.
2. Écrire un programme qui indique la direction dans laquelle pointe la carte à tout moment.
   Pour calibrer la boussole, vous pouvez utiliser `compass.calibrate()` avant la boucle infinie.
3. Dessiner deux coeurs, un petit et un gros, puis faire apparaitre un coeur qui bat sur la matrice.
4. Le planeur est le plus petit vaisseau qui peut apparaitre dans le Jeu de la vie de Conway. Faire défiler un planeur sur la matrice.

Exercice 5 — Boucles finies avec condition

La boucle `while` vue dans l'exercice précédent peut aussi s'arrêter à une certain condition.

```
from microbit import *
import time

limite = 10
tour = 0

while tour < limite:
    display.scroll(tour)
    tour = tour + 1
    time.sleep(0.5)

display.scroll("Fin")
```

1. Décrire ce que fait chaque ligne.
2. Écrire un programme qui fait un décompte en partant de 5 puis qui affiche un coeur.

Exercice 6 — Boucle finie

Il est possible aussi de faire des boucles sur des ensembles définis à l'avance.
Par exemple faire déplacer un point sur la matrice.

```
from microbit import *
import time

for i in range(5):
    display.set_led(i, 0, 9)
    # set_led(colonne, ligne, intensité)
    time.sleep(0.5)
```

1. Faire parcourir au point toute la matrice led.
2. Faire clignoter 5 fois chaque led de la première colonne de la matrice.
3. (*) Faire 5 relevés de température à une seconde d'intervalle, ajouter ces 5 valeurs, puis afficher la température moyenne sur les 5 secondes écoulées.

On peut aussi faire une boucle sur des éléments d'une liste

```
from microbit import *
import time

images = [
    Image.HEART,
    Image.HEART_SMALL,
    Image.HAPPY,
    Image.SAD,
    Image.CONFUSED
]

for img in images:
    display.show(img)
    time.sleep(1)
```

4. Changer les images affichées.
5. Afficher une aiguille qui tourne.

Exercice 7 — Capteurs

Il est aussi possible d'utiliser les deux boutons à côté de la matrice led.

```
from microbit import *
import time

while True:
    if button_a.is_pressed():
        display.show(Image.HAPPY)
        time.sleep(1)
        display.clear()
    elif button_b.is_pressed():
        display.show(Image.SAD)
        time.sleep(1)
        display.clear()
    else:
        display.show(Image.HEART)
```

1. Modifier le programme pour qu'il affiche la température si le bouton a est préssé et la direction de la boussole sinon.
2. (*) Écrire un programme qui montre un visage content sauf quand on secoue le microbit où le visage devient triste.
3. (*) Écrire une programme qui permet de faire circuler des images dans une liste avec les boutons a et b.

Exercice 8 — Plus ou moins

Écrire un programme qui compte le nombre de fois où vous avez appuyé sur le bouton A et sur le bouton B et qui affiche les symboles :

- "=" : si vous avez autant appuyé sur A que sur B.
- ">" : si vous avez plus appuyé sur A que sur B.
- "<" : si vous avez moins appuyé sur A que sur B.

Exercice 9 — Queulorior

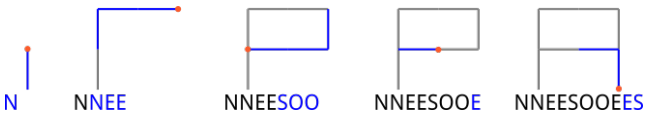
Le Queulorior a ceci de particulier que de la peinture s'écoule au bout de sa queue. Ainsi, lorsqu'il se déplace, il laisse une trace.

Sacha sait que vous avez un Queulorior et pour communiquer avec vous, il vous envoie des instructions à donner au Queulorior. Ces instructions forment une séquence assez longue composée de 4 lettres : N, S, O et E. La lettre N indique au Queulorior de faire un pas vers le nord, la lettre S lui indique de faire un pas vers le sud, la lettre E un pas vers l'est et la lettre O un pas vers l'ouest.

Par exemple, donnez ces instructions au Queulorior :

NNEESOOEES

Il va faire deux pas vers le nord, deux pas vers l'est, un pas vers le sud, deux pas vers l'ouest, deux pas vers l'est et un pas vers le sud, ce qui dessinera un A approximatif :



Écrire un programme où le message est stocké dans une variable et où le message décodé s'affiche sur l'écran.

Exercice 10 — Snake

On souhaite coder un prototype du jeu snake sur le micro:bit. Vous pouvez le commencer seul ou suivre les étapes suivantes

1. Pour le moment le serpent a une taille de 1 pixel. Le faire avancer en ligne droite sur la matrice. Quand il sort de la matrice, tout l'écran doit clignoter.
2. On veut pouvoir le faire tourner à droite ou à gauche avec les boutons A et B. Ajouter cette possibilité de changement de direction.
3. Ajouter une choses à manger (un pixel). La matrice doit afficher un cœur quand la chose est touché par le serpent puis la chose réapparait à un autre endroit.
4. (*) Faire en sorte que le serpent face 2 pixels de long.
5. (*) Le serpent doit s'allonger de 1 pixel à chaque fois qu'il mange un pixel.
6. (*) La partie doit être perdue quand la tête du serpent touche une partie de son corps.

Exercice 11 — Pure Python

Ci-dessous quelques exercices techniques pour se faire la main avec Python. À vous de choisir les bons outils dans le corpus pour les résoudre.

Les exercices peuvent être faire dans l'ordre de votre choix.

1. **Mot de passe :** Étant donné un mot de passe stocké dans une variable. Le programme demande à l'utilisateur de rentrer un mot de passe tant que ce dernier n'est pas identique à celui stocké. Quand le mot de passe est trouvé, on afficher "accès autorisé"
2. **Plus ou moins** : Étant donné un nombre stocké dans une variable. Le programme demande à l'utilisateur de le deviner. Tant qu'il ne la pas trouvé, le programme lui dira si la proposition est plus grande ou plus petite que la valeur stockée.
3. **Questionnaire à une question :** Étant données une question et la réponse associée stockées dans deux variables. Le programme demande à l'utilisateur la réponse à la question et affiche si la réponse est bonne ou non. *(On pourra amélioré ce programme en stockant plusieurs réponses possibles)*.

4. **QCM :** Étant donnée une liste de questions fermées (qui se répondent pas oui ou non). Le programme pose les questions successivement et à la fin donne le nombre de bonnes réponses.
5. **Table de ... :** Étant donné un nombre ainsi qu'une limite. Affiche tous les multiples de ce nombre juqu'à atteindre la limite.
6. **Table de multiplication :** Étant donné un nombre limite. Affiche la table de multiplication de tous les nombres de 0 à cette limite.
7. **Multiples de 3 ou 5 :** Étant donné un nombre limite. Afficher tous les multiples de 3 ou de 5 inférieur à cette limite.
8. **Pyramide de lettres :** Étant donnée une chaine de caractères. Afficher un fois la première lettre, puis deux fois la deuxième puis 3 fois la 3e...
9. **Moyenne interactive :** Le programme demande des nombres jusqu'à ce que l'utilisateur entre "s" (pour stop) puis affiche la moyenne des nombres entrés.
10. **Max interactif :** Le programme demande des nombres jusqu'à ce que l'utilisateur entre "s" (pour stop) puis affiche le max des nombres entrés.