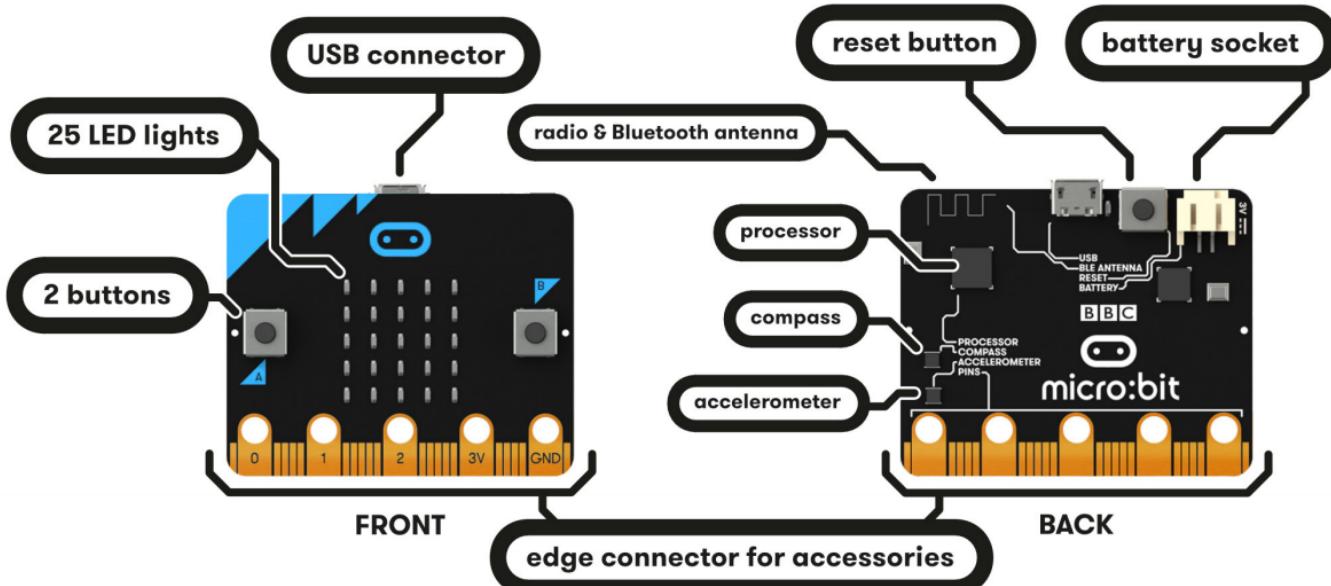


# TP : Découverte de la carte micro:bit

NOM :  
Prénom :

## Présentation de la carte

La carte micro:bit éditée par la BBC, est un nano-ordinateur qui peut équiper un système informatique embarqué. Elle est munie d'un processeur ARM et de plusieurs capteurs et interfaces de connexion. Le guide de présentation en ligne est disponible sur <https://microbit.org/fr/guide/>.



Nous utiliserons dans un premier temps la carte en la connectant à un ordinateur avec le câble USB fourni qui assure la liaison de communication et l'alimentation.

Ensuite, lorsque l'on voudra intégrer la carte dans un système embarqué, il sera possible de la connecter à une alimentation externe par piles.

Lorsque la communication entre l'ordinateur et la carte échoue, on peut essayer de la redémarrer avec le bouton reset situé au verso.

## *Le premier programme*

Les manipulations suivantes sont à effectuer en ligne via le site web :  
<https://python.microbit.org/>

### Exercice 1 :

Recopier le code ci-dessous dans l'environnement de programmation micro-bit.

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

Ensuite utilisez le bouton

Send to micro:bit



pour transférer ce programme sur la carte.

### Décrire l'effet du programme :

.....  
.....

### Quel est le rôle de la fonction display.show ?

.....

On pourra tester et remplacer HAPPY par HEART, DIAMOND, DUCK, etc...

## *Une première boucle*

Lorsque l'on veut faire tourner un programme avec un effet qui se répète, on doit faire une boucle qui ne s'arrête pas à l'aide de la syntaxe : `while True :` (on parle de boucle infinie)

### Exercice 2 :

Recopier le code ci-dessous dans l'environnement de programmation micro-bit.

```
from microbit import *\n\nwhile True:\n    if button_a.is_pressed():\n        display.show(Image.HAPPY)\n    else:\n        display.show(Image.SAD)
```

Ensuite utilisez le bouton  pour transférer ce programme sur la carte.

### Décrire l'effet du programme :

.....  
.....  
.....

### Que signifie l'instruction : `if button_a.is_pressed() :` ?

.....  
.....

### Que signifie l'instruction : `else :` ?

.....  
.....

### Quel est le capteur ?

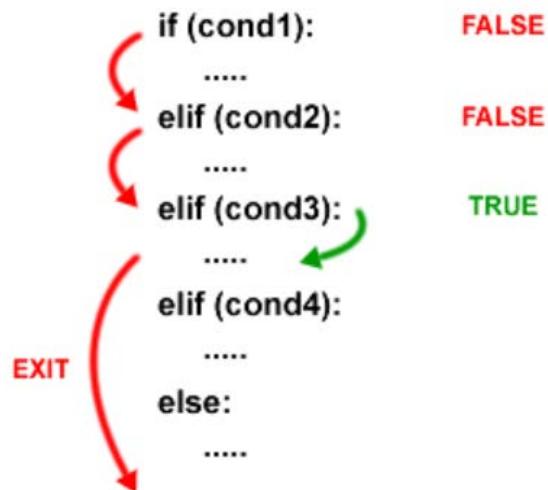
.....  
.....

### Quel sont les actionneurs ?

## *Les instructions conditionnelles*

On a vu dans la partie précédente qu'il était possible de déclencher différents effets suivants les résultats des capteurs à l'aide de l'instruction conditionnelles (if / else).

De façon générale, cette instruction est la suivante :



### Exercice 3 :

Recopier en remplaçant les mots rouges le code ci-dessous dans l'environnement de programmation micro-bit.

On souhaite qu'un cœur apparaisse si on appuis sur le bouton A, qu'un canard apparaisse si on appuie sur le bouton B et qu'un crane apparaisse si on n'appuie sur rien.

```
from microbit import *  
  
while True:  
    if condition 1:  
        instruction 1  
    elif condition 2 :  
        instruction 2  
    else :  
        instruction 3
```

Ensuite utilisez le bouton pour transférer ce programme sur la carte.

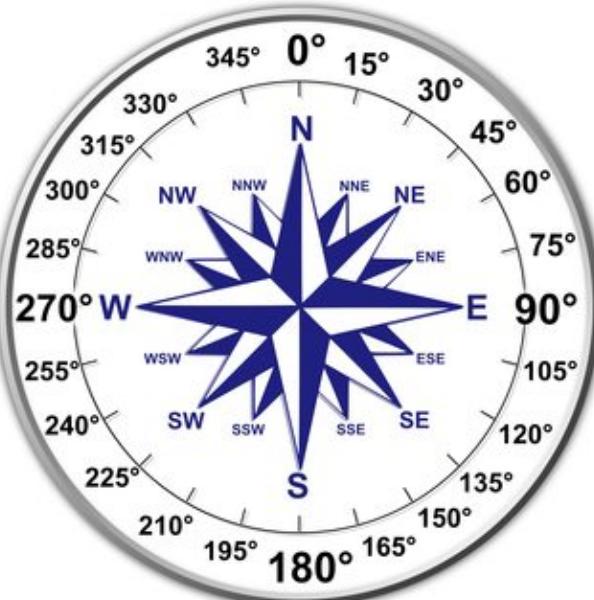
**Appelez un enseignant pour valider votre résultat.**

## La boussole

La carte micro:bit dispose d'un capteur de champ électromagnétique accessible par l'objet **compass**.

Avant toute utilisation du compass, il faut le calibrer avec `compass.calibrate()` : un message défilant sur l'écran nous invite à incliner la carte jusqu'à ce que les 25 diodes soient allumées.

`compass.heading()` retourne l'angle entre le vecteur du champ magnétique mesuré et le Nord magnétique comme 0. La mesure en degrés, est un entier compris entre 0 et 360. Par défaut, le champ magnétique terrestre est mesuré. Pour simuler une boussole indiquant les quatre point cardinaux avec la carte micro:bit, on découpe l'intervalle [0; 360] en 4 intervalles d'amplitude 90 degrés.



### Exercice 4 :

Recopier le code ci-dessous dans l'environnement de programmation micro-bit.

```
from microbit import *

#Calibrage du compas
compass.calibrate()

#boucle infinie
while True:
    angle = compass.heading()
    if 315 <= angle or angle <= 45:
        display.show("N")
    elif 45 < angle and angle <= 135:
        display.show("E")
```

Ensuite utilisez le bouton

Send to micro:bit

:

pour transférer ce programme sur la carte.

*On remarquera qu'au déclenchement du programme, le message « TILT TO FILL SCREEN » apparaît. C'est le message qui indique qu'il faut bouger la carte micro-bit dans tous les sens pour la calibrer (l'écran doit être totalement allumé pour le calibrage soit validé).*

### Préciser le rôle de :

- ```
if 315 <= angle or angle <= 45:  
    display.show('N')  
.....
```
- ```
elif 45 < angle and angle <= 135:  
    display.show('E')  
.....
```

### Identifier capteur et actionneur :

### Pour aller plus loin :

1. Compléter le programme pour qu'il affiche aussi les directions Sud et Ouest.  
Tester votre programme sur la carte micro:bit
2. Compléter le programme pour qu'un appui sur le bouton A provoque une sortie de la boucle suivie d'un effacement de l'écran.  
Tester votre programme sur la carte micro:bit
3. Modifier le programme pour qu'il affiche aussi les directions intermédiaires (SE pour Sud Est par exemple).  
Tester votre programme sur la carte micro:bit