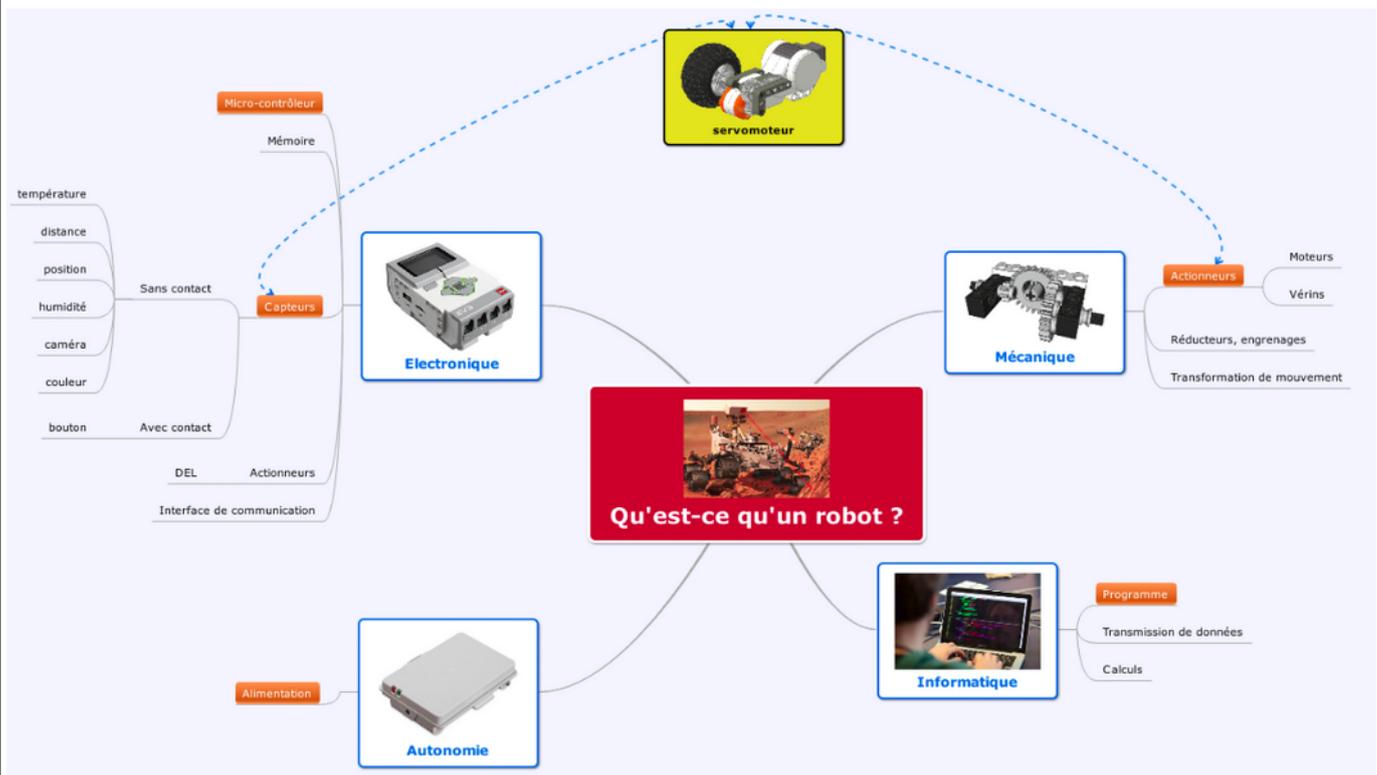


# Chap 13 – Informatique embarquée

## Objectifs

1. Périphériques d'entrée et de sortie Interface Homme-Machine (IHM) (capteurs et actionneurs systèmes embarqués ou robots)

## I. Rappel – Notions liées à la robotique



Un système informatique embarqué collecte des informations du monde réel à l'aide de capteurs, les traite dans un microprocesseur puis agit sur le monde réel par le biais d'actionneurs. Le traitement des informations est contrôlé est un programme qui peut interagir avec l'homme à travers une Interface Homme Machine.

### A faire vous-même 1

Trouvez 3-4 exemples de capteurs :

.....

Trouvez 3-4 exemples d'actionneurs :

.....

## II. Généralités sur le microcontrôleur

### A. Où sont utilisés les microcontrôleurs ?

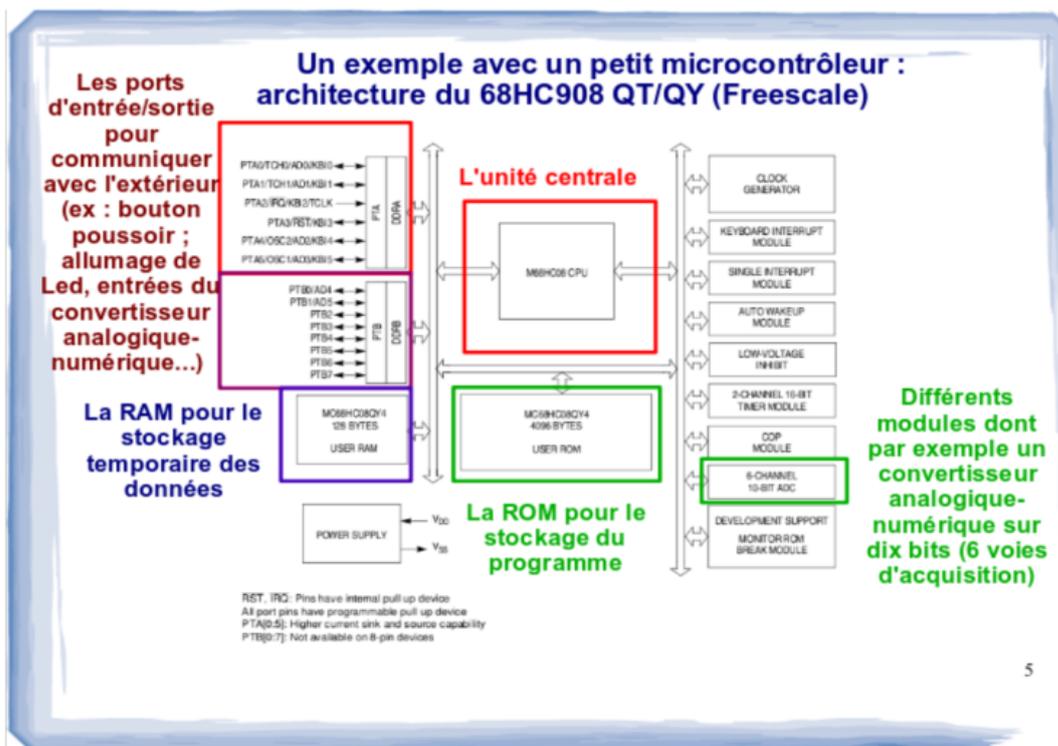


## B. Aperçu technique

Dans un ordinateur on trouve :

- un microprocesseur (Intel, AMD)
- de la mémoire vive (RAM) pour du stockage temporaire (quelques Go)
- de la mémoire pour du stockage permanent (disque dur : plusieurs centaines de Go)
- des ports de communication pour dialoguer avec l'extérieur (USB, Ethernet, WiFi, carte son, carte graphique...)

Le microcontrôleur, c'est un peu tout cela dans un format réduit (de l'ordre du cm<sup>2</sup>) avec quelques nuances cependant.



Le schéma ci-dessus représente sous forme de blocs fonctionnels le contenu d'un ancien « tout petit » microcontrôleur (dont une version réduite tient dans

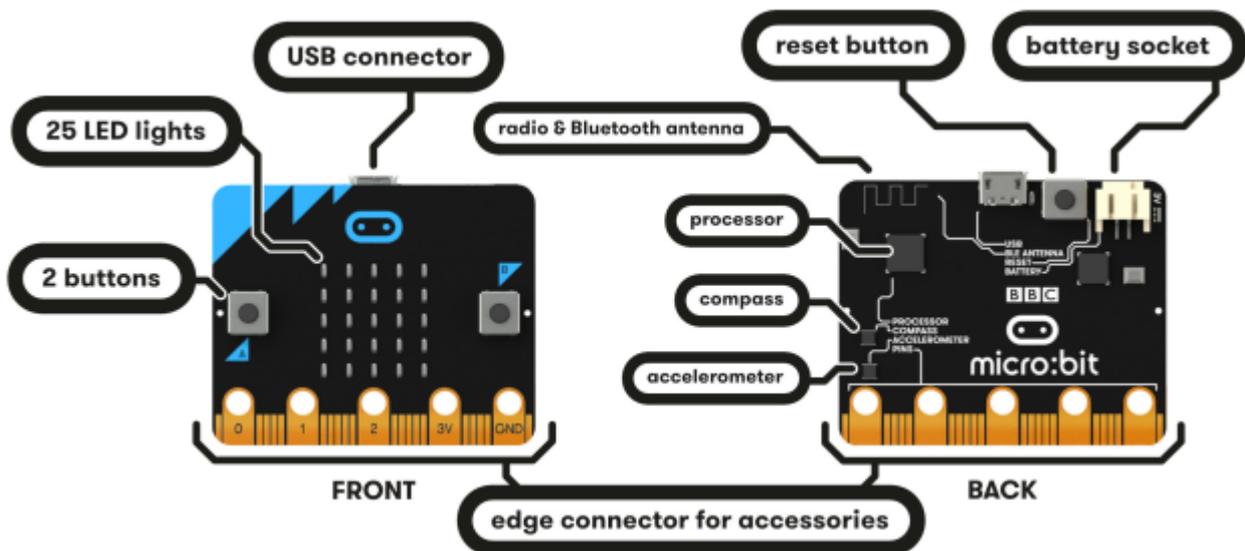
un boîtier de 8 broches seulement...)

On y trouve :

- un microprocesseur (CPU : Central Processor Unit)
- de la mémoire de type RAM (ici 128 octets seulement)
- de la mémoire non volatile pour stocker le programme (c'est l'équivalent du disque dur pour un ordinateur). Cette mémoire est maintenant reprogrammable ce qui permet de mettre à jour le programme ("firmware") (ici 4096 octets = 4 ko ... !!!)
- des ports d'entrées / sorties pour communiquer avec l'extérieur
- un convertisseur analogique-numériques (CAN) pour faire de la mesure

## C. La carte Micro:Bit

La carte micro:bit éditée par la BBC , est un nano-ordinateur qui peut équiper un système informatique embarqué.



Source : <https://microbit.org/fr/guide/features/>

Comparatif Arduino vs. Micro:Bit :

	ARDUINO UNO	BBC:MICROBIT
<b>Microcontrôleur</b>	ATmega328P	ARM Cortex M0
Architecture	8 bits	<b>32 bits</b>
Fréquence	16 MHz	16 MHz
Mémoire Flash	32 KB	<b>256 KB</b>
RAM	2 KB	<b>16 KB</b>
EEPROM	1 KB	
Alimentation	<b>5 V</b>	3,3 V
Entrées/Sorties	14	19
CAN	10 bits	10 bits
Bus série	I2C, SPI,UART,USB	I2C, SPI,UART,USB
<b>Capteurs/Actuateurs</b>		
Boutons poussoirs	-	<b>2</b>
LEDs	1	<b>25 (matrice 5x5)</b>
Accéléromètre	-	<b>1</b>
Boussole	-	<b>1</b>
Capteur de lumière	-	<b>(avec les Leds)</b>
Capteur de T°	-	<b>T° du chip</b>
Antenne radio	-	<b>1</b>
<b>Programmation</b>	C	<b>Python</b>
Documentation	<b>Importante</b>	Faible
<b>Cartes additionnelles</b>	<b>Très variées</b>	Beaucoup moins
<b>Prix</b>	~ 20 €	~ 20 €

### III. Programmons

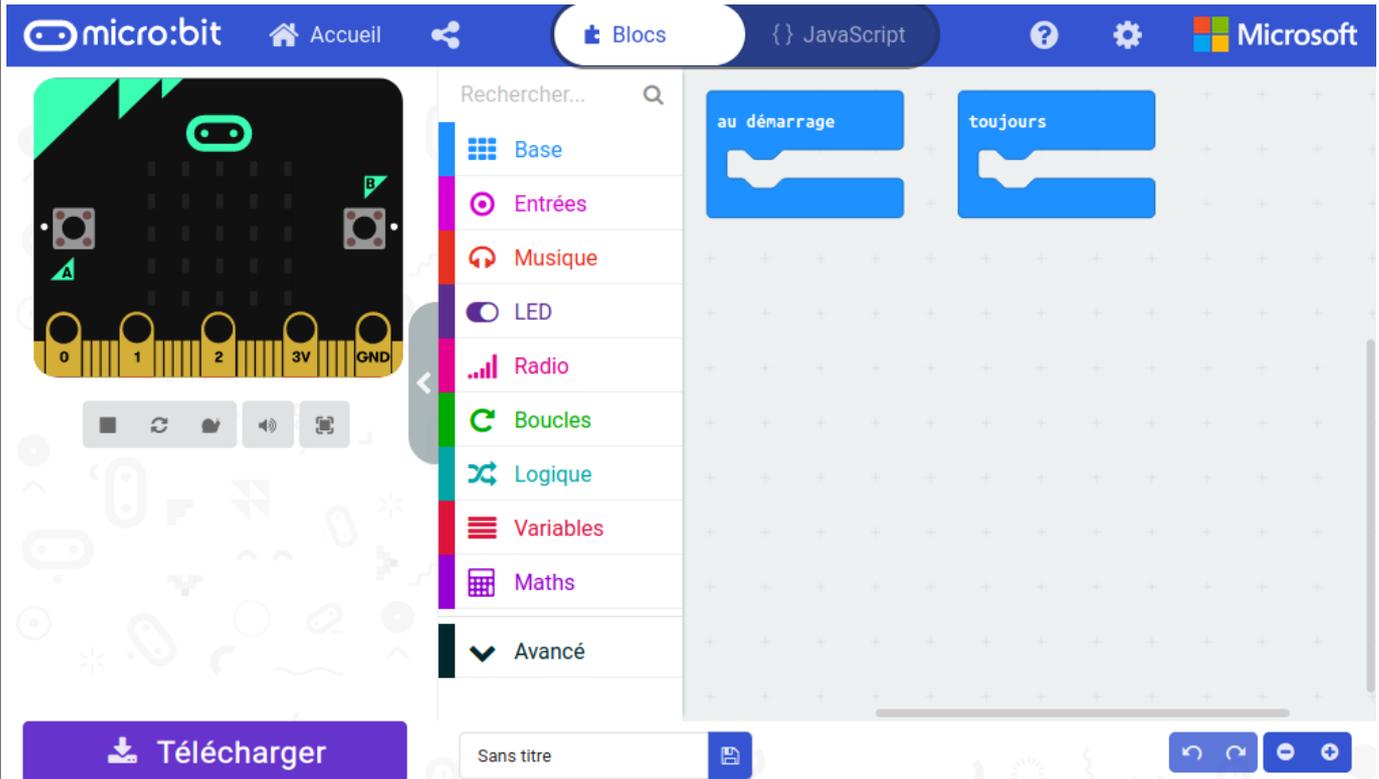
#### A. L'éditeur Makecode

Nous allons, dans un premier temps, utiliser l'éditeur MakeCode afin de réaliser ce programme interprétable par notre carte Micro:bit.

##### A faire vous-même 2

1. Pour commencer ouvrez la page Web suivante : <https://microbit.org/fr/code/>
2. Cliquez sur 
3. Une nouvelle page Web s'ouvre, cliquez alors sur l'icône : « Nouveau Projet ».
4. Vous arrivez sur un outil qui vous permet de concevoir des programmes

pour Micro:bit en utilisant des « blocs de programmation » (le modèle de conception est similaire à celui de Scratch)



### A faire vous-même 3

Pour une première prise en main nous allons créer un programme afin de pouvoir jouer au jeu : Pierre, Papier, Ciseau avec la carte Micro:bit.

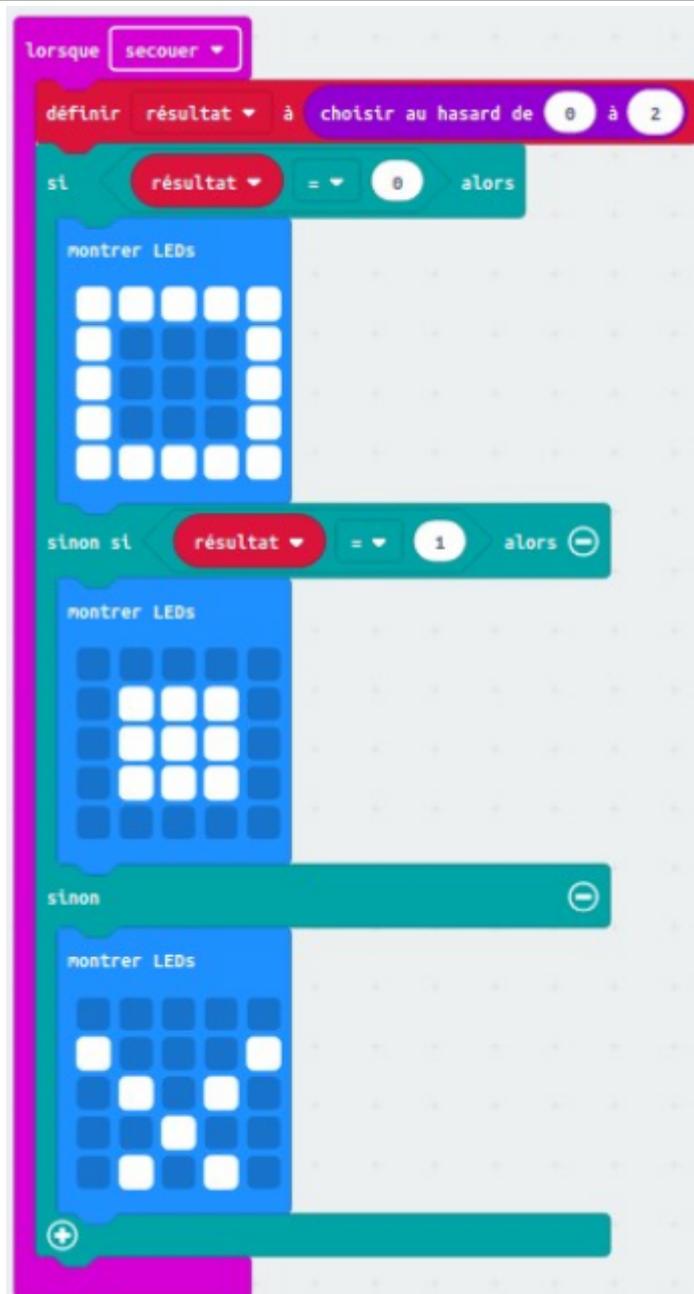
Nous allons utiliser les capteurs de mouvements de la carte Micro:bit afin de le mettre en action de jeu.

Lorsque l'utilisateur secoue la carte cette dernière doit afficher aléatoirement un symbole associé à une des trois solutions possibles. Il faut également pouvoir éteindre totalement l'écran après une partie.

1. Créez une variable « résultat » qui sera associée aléatoirement à Papier, Pierre ou Ciseau que nous symboliserons par trois chiffres, à savoir respectivement : 0, 1 et 2.
2. Codez :



3. Mettez en place une structure conditionnelle définissant la marche à suivre en fonction du résultat obtenu et utilisez des comparateurs logiques (ici « égal à ») afin d'établir les conditions.



Ici le premier symbole correspond au Papier, le deuxième à la Pierre et le troisième au Ciseau.

- Il faut maintenant pouvoir effacer l'écran une fois la manche terminée. Pour ce faire nous allons utiliser un autre capteur de la carte Micro:bit, à savoir la pression de l'un des boutons de la carte.



- Vous pouvez le tester en appuyant sur les boutons « shake » et « A » sur le simulateur de carte Micro:bit situé sur la gauche de votre écran.
- Téléchargez le programme dans la carte Micro:Bit en cliquant sur l'icône correspondant (bas de page) et en ayant pris soin de bien le nommer. La

dernière étape consiste à envoyer le fichier obtenu (.hex) sur votre carte Micro:bit par transfert USB.

7. Vous pouvez maintenant jouer à Pierre, Papier, Ciseaux.

## B. Programmez en Python

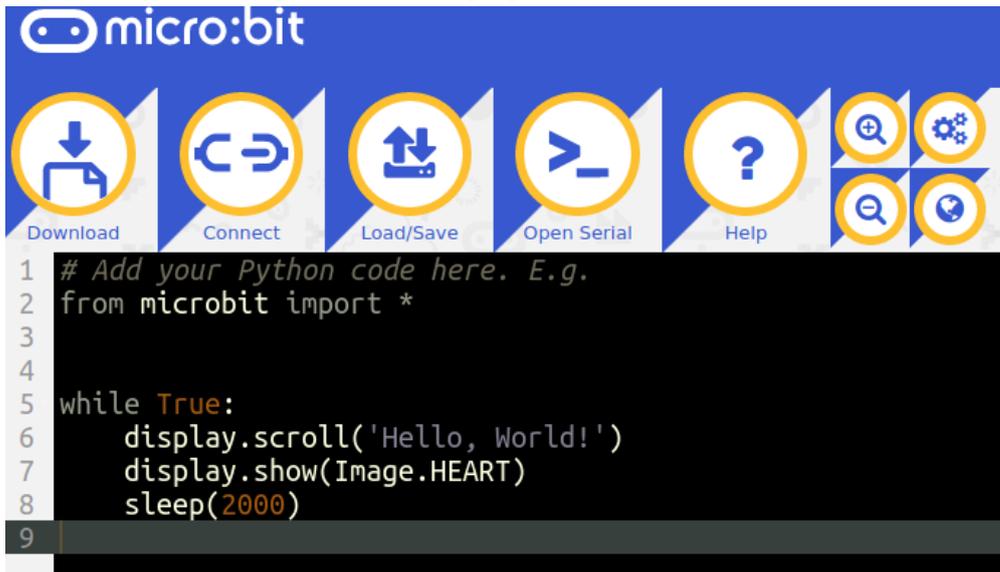


### A faire vous-même 4

1. Pour commencer ouvrez la page Web suivante : <https://microbit.org/fr/code/>

2. Cliquez sur 

3. Vous arrivez sur un outil qui vous permet de concevoir des programmes pour Micro:bit en utilisant le langage Python



### A faire vous-même 5



1. Recopiez le code Python suivant :

```
from microbit import *  
display.scroll("Hello World")
```

2. Téléchargez-le dans la carte en cliquant sur le bouton Download

3. Que fait ce programme ?

.....

### A faire vous-même 6

1. Recopiez le code Python suivant :

```
from microbit import *  
#boucle infinie  
while True:  
    #Structure conditionnelle avec 3 choix  
    if button_a.is_pressed():  
        display.show(Image.HAPPY)  
    elif button_b.is_pressed():  
        display.show(Image.ANGRY)  
    else:  
        display.show(Image.SAD)
```

2. Téléchargez-le dans la carte en cliquant sur le bouton Download
3. Que fait ce programme ?

.....  
4. Identifiez capteur et actionneur.  
.....

#### A faire vous-même 7

1. Recopiez le code Python suivant :

```
1 # Add your Python code here. E.g.
2 from microbit import *
3 from random import randint
4
5
6 while True:
7     gesture = accelerometer.current_gesture()
8     if gesture == "shake":
9         resultat=randint(0,2)
10        if resultat == 0:
11            display.show(Image("99999:90009:90009:90009:99999")) #Papier
12        elif resultat == 1:
13            display.show(Image("00000:09990:09990:09990:00000")) #Pierre
14        else:
15            display.show(Image("00000:90009:09090:00900:09090")) #Ciseau
16    if button_a.was_pressed():
17        display.clear()
```

2. Téléchargez-le dans la carte en cliquant sur le bouton Download
  3. Actionnez la carte
  4. Que fait ce programme ?
- .....

#### A faire vous-même 8

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.

1. Recopiez le code Python suivant :

```
from microbit import *
#Calibrage du compas
```

```

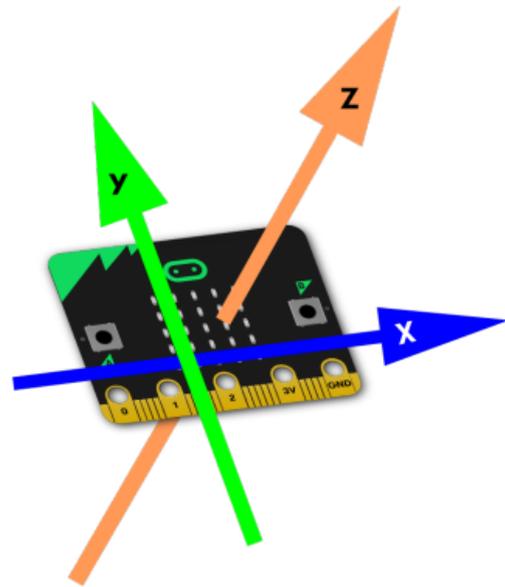
compass.calibrate()
#boucle
while True:
    angle = compass.heading()
    if 315 <= angle or angle <= 45:
        display.show('N')
    elif 45 < angle and angle <= 135:
        display.show('E')
#attente d'une seconde
sleep(1000)

```

2. Téléchargez-le dans la carte en cliquant sur le bouton Download
3. Actionnez la carte

## A faire vous-même 9

L'accéléromètre est un capteur mesurant l'accélération de la carte micro:bit . Il détecte ses mouvements et son inclinaison. La mesure fournie comporte trois composantes suivant les axes d'un repère de l'espace, chacune prend des valeurs entières entre  $-2000$  et  $2000$  g où g est l'unité d'accélération représentant approximativement la pesanteur terrestre.



`accelerometer.get_x()` , `accelerometer.get_y()` , et `accelerometer.get_z()` donnent les valeurs des composantes de l'accélération suivant chaque axe. `accelerometer.get_values()` retourne le triplet de composantes.

1. Créez un programme qui doit répondre aux spécifications suivantes :

Dans une boucle :

- capturer la valeur absolue de la composante en x de l'accélération dans une variable avec `gx = accelerometer.get_x()` ;
- faire défiler la température capturée avec `display.scroll(temperature())` si `gx > 100`
- ou sinon effacer l'écran avec `display.clear()` .
- Sortir de la boucle si le bouton A est pressé puis effacer l'écran.
- La température capturée par `temperature()` est celle du processeur mais comme il chauffe peu c'est une bonne approximation de celle de l'environnement.