

# Introduction à BBC micro:bit

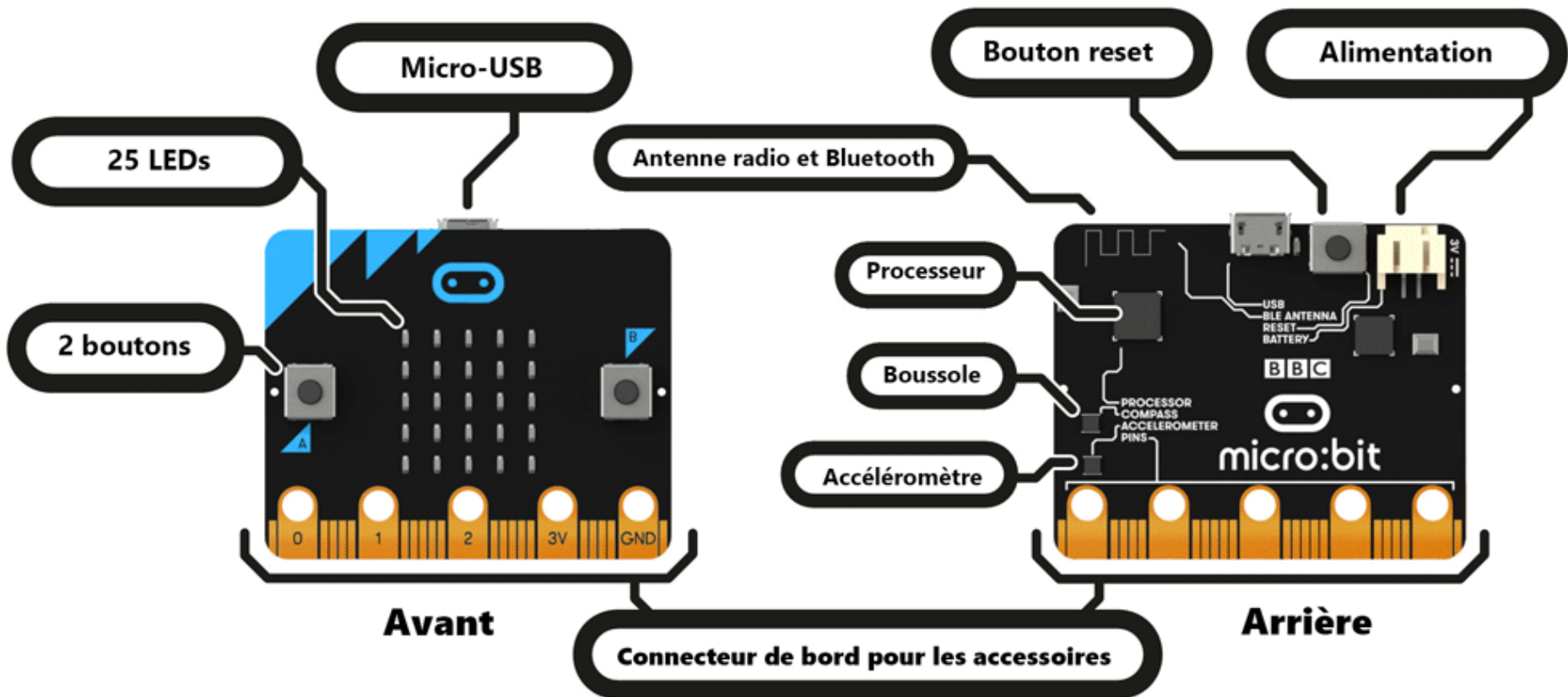
<https://microbit.org>

# Kezaco ?

- **micro:bit** est un micro-contrôleur avec plein de capteurs.
- 

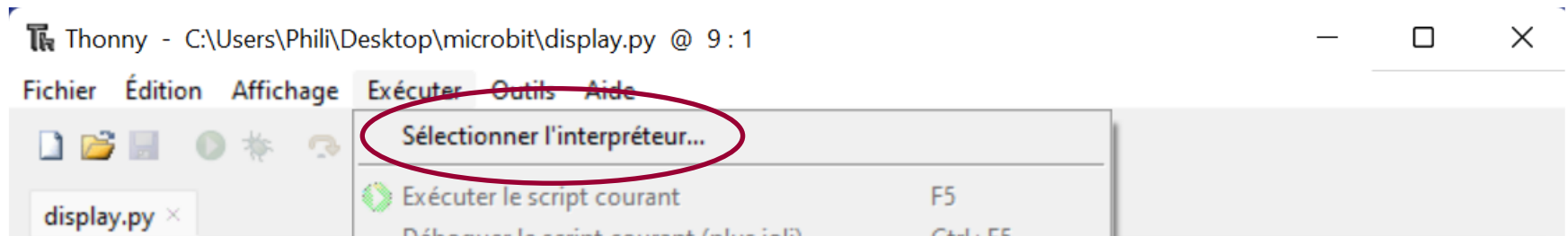


# des capteurs

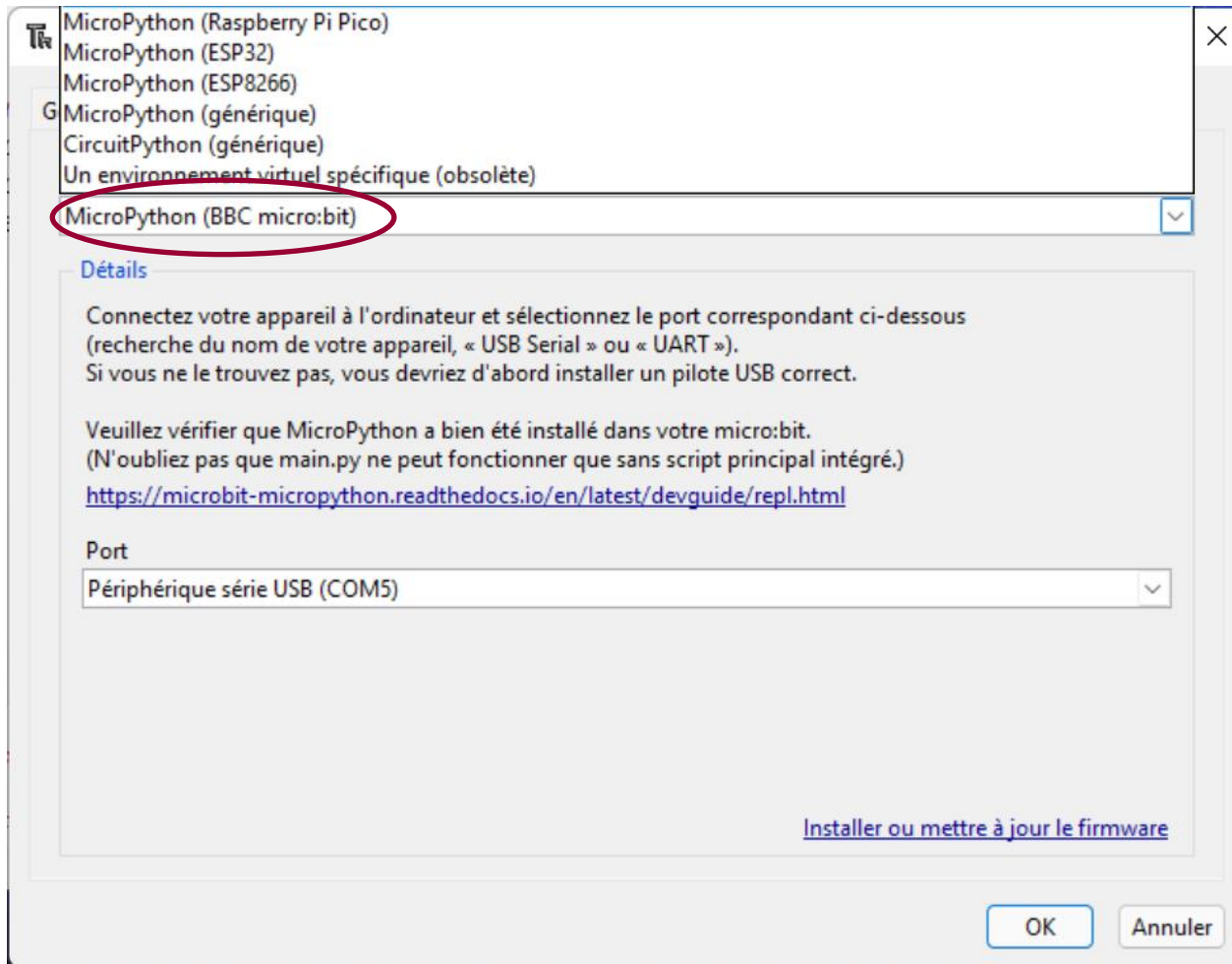


# Configurer l'interpréteur

- Avec Thonny (<https://thonny.org>)



# Configurer l'interpréteur



# Configurer l'interpréteur

Options de Thonny

Général Interpréteur Éditeur Thème et police Exécuter et déboguer Terminal Console Assistant

☐ Autoriser qu'une seule instance de Thonny

☐ Journaliser les événements d'utilisation du programme

☐ Réouvrir tous les fichiers de la session précédente

☐ Désactiver les sons de notification

☐ Mode de débogage (fournit des journaux de diagnostic plus détaillés)

Langue Français

Type d'interface regular

Facteur d'échelle de l'interface default

Mode de mise à l'échelle de la police default

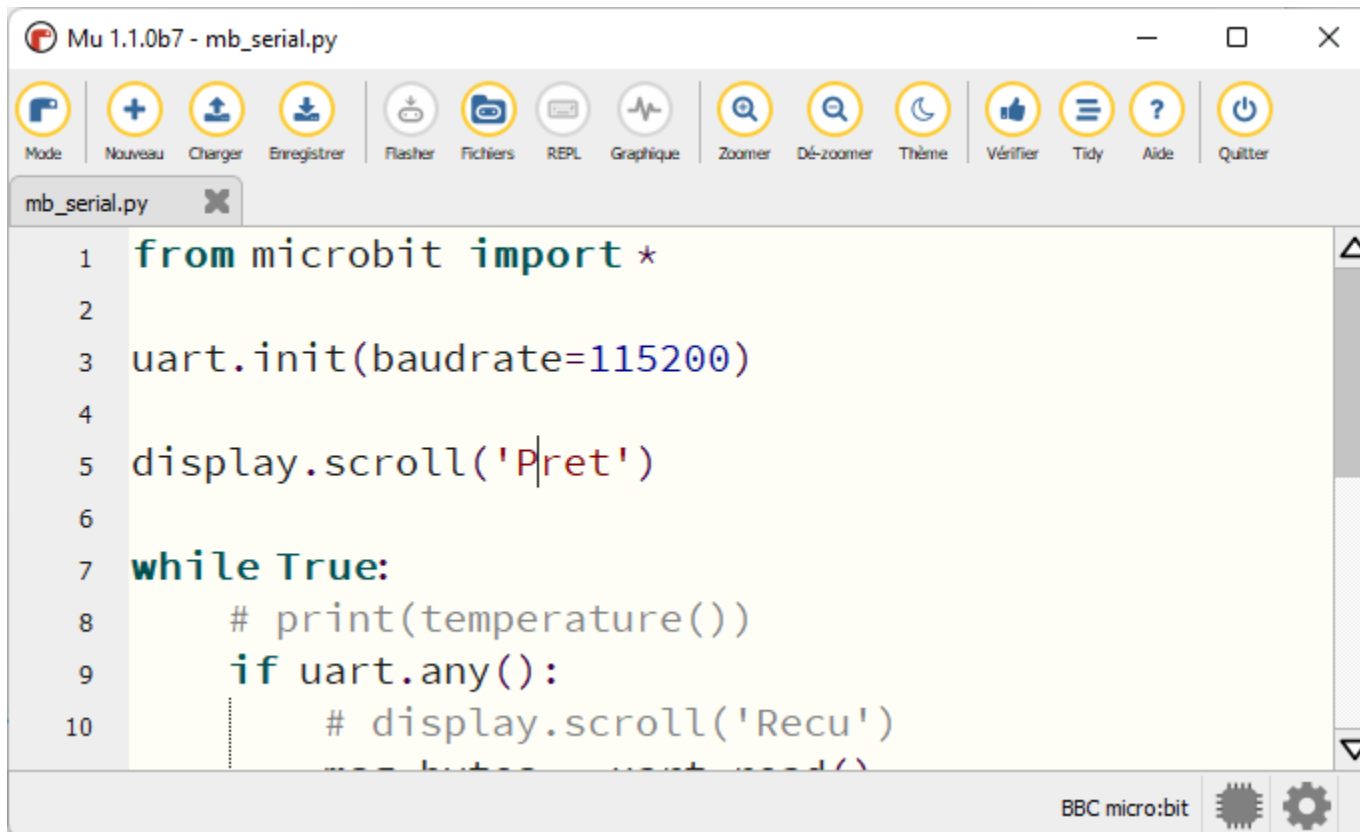
Variables d'environnement (une paire clé=valeur par ligne)

Remarque : Redémarrez Thonny après avoir modifié ces options !

OK Annuler

# Configurer l'interpréteur

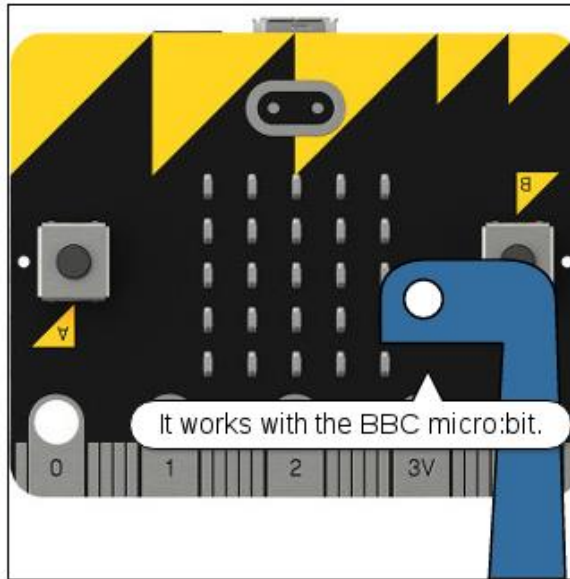
- Avec Mu: (<https://codewith.mu/en/download>)



# Premiers pas !

## First Steps with MicroPython by Mike Rowbitt

MicroPython was created by Damien...



```
from microbit import *  
# Edit your code here!  
display.scroll("Hello, world!")
```



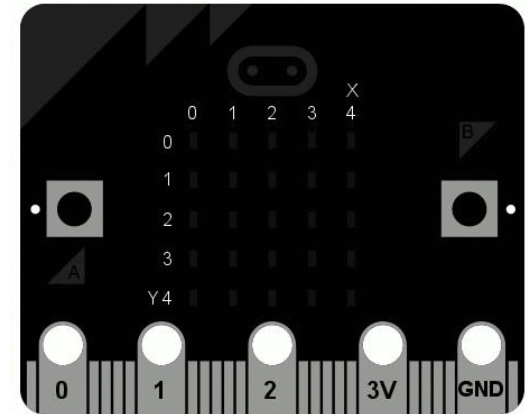
Everything you need to know about MicroPython on the BBC micro:bit is found in this documentation.

Generated by Python Comics. [MAKE YOUR OWN](#)



# Images

- `display.show(Image.XXX)`
- `display.scroll("texte")`
- `display.set_pixel(x,y,valeur)` # entre 0 et 9
- `display.read_light_detect()` # entre 0 et 100



<https://microbit-micropython.readthedocs.io/en/v2-docs/display.html>

# Images



Thonny - C:\Users\Phili\Desktop\microbit\display.py @ 9 : 1

Fichier Édition Affichage Exécuter Outils Aide

display.py ×

```
1 # display
2 from microbit import *
3
4
5 while True:
6     display.scroll('Hello, World!')
7     display.show(Image.HEART)
8     sleep(2000)
9
```

Console ×

MicroPython v1.9.2-34-gd64154c73 on 2017-09-01; micro:bit v1.0.1 with nRF51822  
Type "help()" for more information.  
>>>

MicroPython (BBC micro:bit)

# A vous !

- Afficher un texte et une image de personne heureuse
- Afficher une image « *faite maison* »

# button

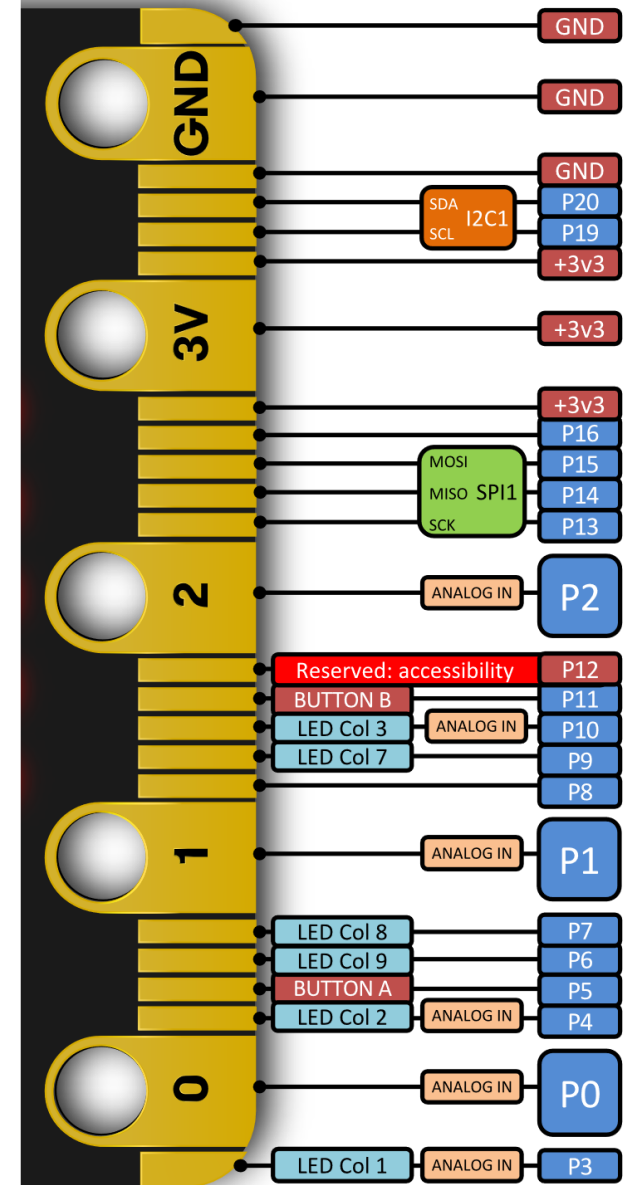
- `button_a`, `button_b`
- `button_x.is_pressed()`
- `button_x.was_pressed()`
- `button_x.get_presses()` # total de press

# A vous !

- Afficher un message particulier suivant l'appui sur les boutons

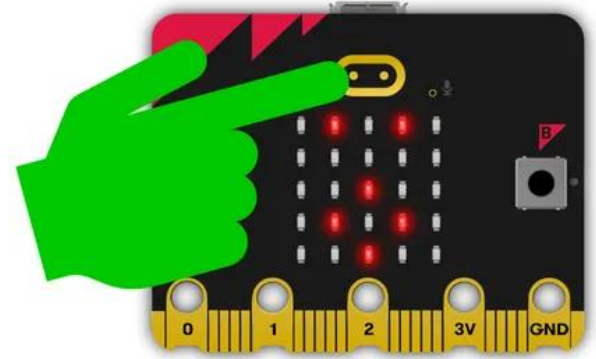
# pin(s)

- `pinx.read_digital()`



# pin\_logo

- `pin_logo.is_touched()`



# temperature

micro:bit possède un capteur de température

- `temperature()` #renvoie la temperature en °C

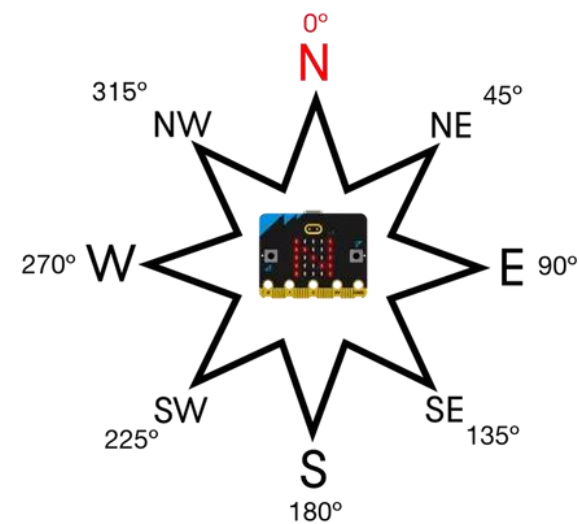


# A vous !

- Afficher la température courante en Fahrenheit sur la matrice courante

# compass

- `compass.calibrate()`
- `compass.heading()`  $\#0^\circ \rightarrow$  NORD

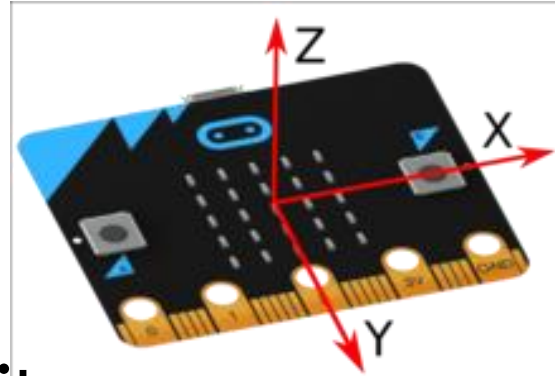


<https://microbit-micropython.readthedocs.io/en/v2-docs/compass.html>

# A vous !

- Afficher l'angle de la boussole (sans et avec calibration initiale)

# accelerometer



- `accelerometer.get_x()` # gauche/droite
- `accelerometer.get_y()` # avant / arrière
- `accelerometer.get_z()` # haut / bas
- `accelerometer.current_gesture()`
- `accelerometer.was_gesture()`

<https://microbit-micropython.readthedocs.io/en/v2-docs/tutorials/gestures.html>

# A vous !

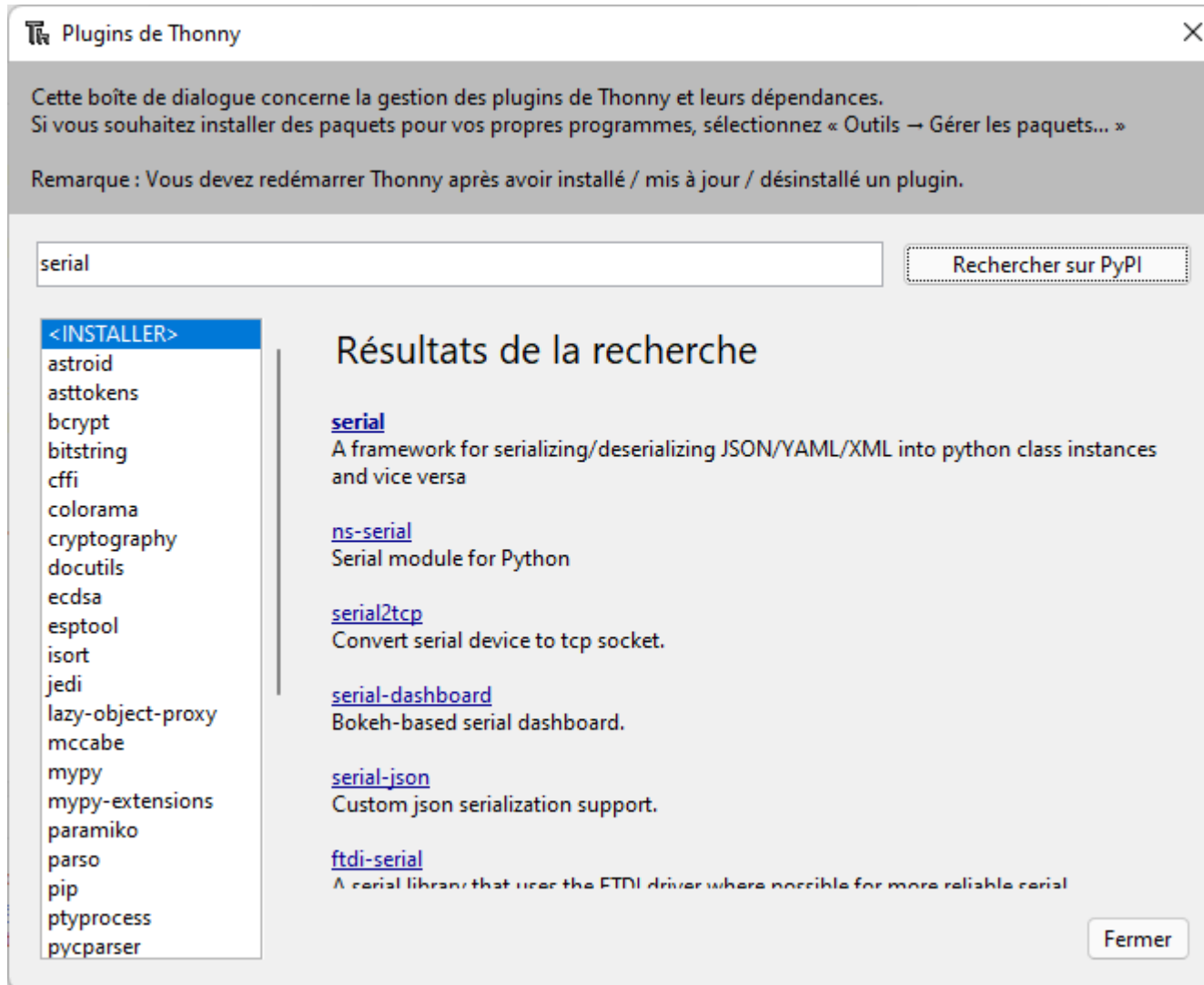
- Suivant le geste réalisé (exemple : un SHAKE), afficher :
  - soit la température courante
  - soit la direction magnétique

# liaison série

L'objectif est ici d'échanger des données (avec un PC, un autre micro:bit, ...)

- `uart.init(...), ...`
- `print("texte")`
- **Rapidité de modulation** = 115200 bauds
- **Données** = 8 bits
- **Parité** = none
- **Bit Stop** = 1 bit

# liaison série



# Exercice

- Récupérer la température depuis micro:bit et envoyer des messages

