Ce document a été inspiré entre autres par la lecture des articles :

http://tableauxmaths.fr/spip/spip.php?article232
Et
https://frederic-junier.org/SNT/Theme3_InformatiqueEmbarquee/ressources/Activite1-Microbit-2019V1.pdf

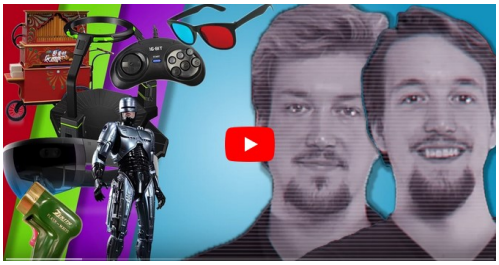
ETAPE 1 : Qu'est-ce qu'une IHM ? : (Interface Homme/Machine) :

_____

_____

Lire les vidéos :

https://www.youtube.com/watch?v=z10xT6rAF7I Et **https://www.youtube.com/watch?v=k-c04hAoN_Q**



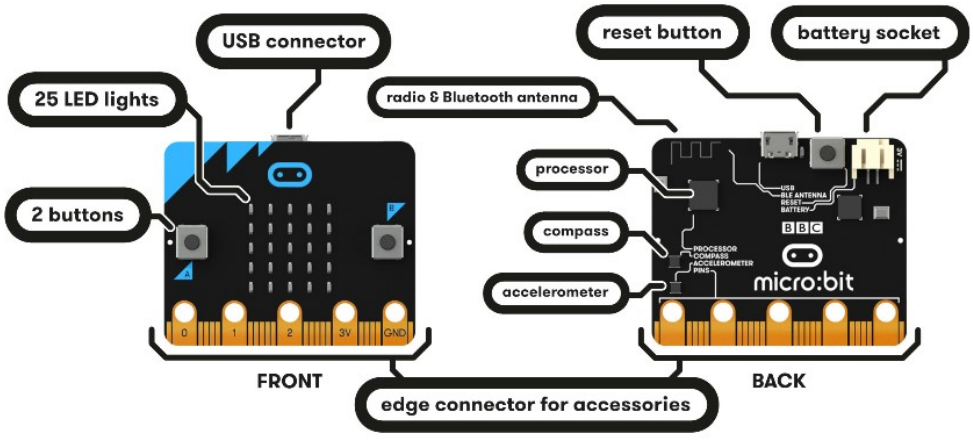
ETAPE 2 : Présentation de la carte BBC :

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/ Et **https://microbit-micropython.readthedocs.io/en/latest/**



https://microbit.org/fr/guide/features/

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

Ce document a été inspiré entre autres par la lecture des articles :

http://tableauxmaths.fr/spip/spip.php?article232
Et
https://frederic-junier.org/SNT/Theme3_InformatiqueEmbarquee/ressources/Activite1-Microbit-2019V1.pdf

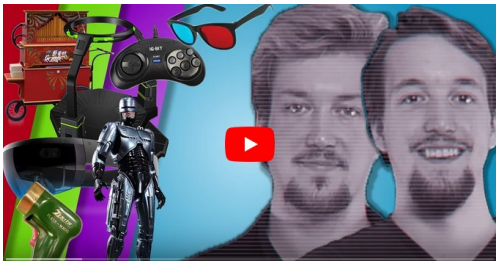
ETAPE 1 : Qu'est-ce qu'une IHM ? : (Interface Homme/Machine) :

Une Interface Homme Machine ou IHM est un ensemble de dispositifs physique (boutons, curseurs) et logiciels (interface graphique) permettant d'échanger des informations avec une machine.

Lire les vidéos :

CORRECTION

https://www.youtube.com/watch?v=z10xT6rAF7I Et **https://www.youtube.com/watch?v=k-c04hAoN_Q**



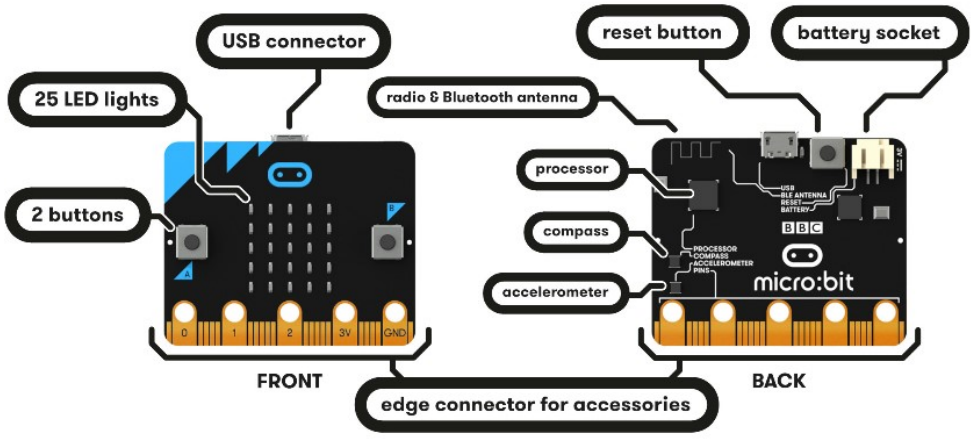
ETAPE 2 : Présentation de la carte BBC :

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/ Et **https://microbit-micropython.readthedocs.io/en/latest/**



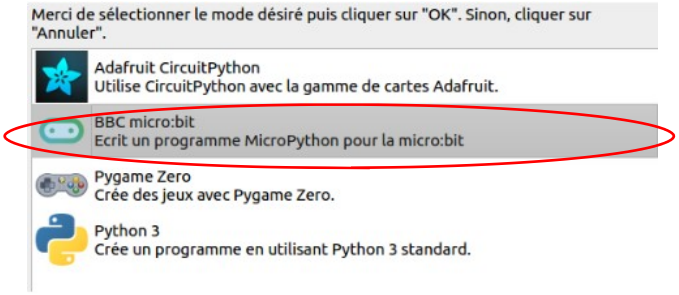
https://microbit.org/fr/guide/features/

Nous utiliserons uniquement la carte en la connectant à un ordinateur avec le câble USB fourni qui assure la liaison de communication et l'alimentation.
(Si on veut intégrer la carte dans un système embarqué, il est possible de la connecter à une alimentation externe **par piles ou batterie LIPO.)**

Mes premiers programmes d'IHM :

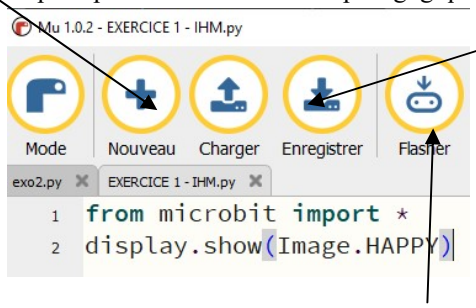
Exo n°1 : Afficher l'image « HAPPY » une seule fois (PAS D'INTERACTION POSSIBLE)

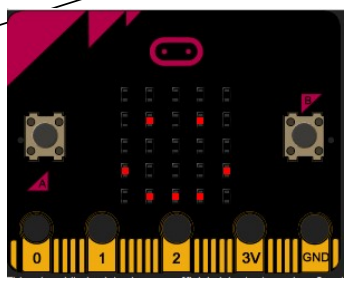
1. Lancer l'environnement de programmation **Mu** depuis le bureau et sélectionner le mode **BBC micro:bit.**



Nous programmerons la carte avec le langage Python et son module microbit

Sélectionner **Nouveau** dans la barre de menu pour créer un nouveau programme puis **enregistrer** le fichier sous le nom *exo1.py* dans un dossier pertinent de son espace personnel sur le réseau pédagogique.





https://create.withcode.uk/

Pour transférer le programme sur la carte, cliquer sur **Flasher**.
Lors de chaque téléversement la mémoire Flash contenant le programme exécuté par la carte est réinitialisée.

Décrire l'effet du programme sur la carte. Une interaction est-elle possible ?

. . . . . .

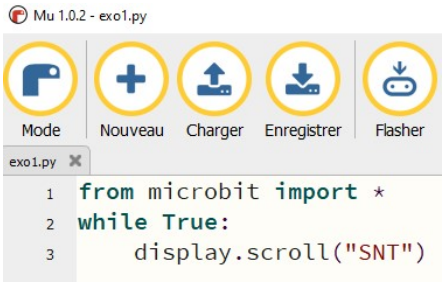
Préciser le rôle de chaque instruction

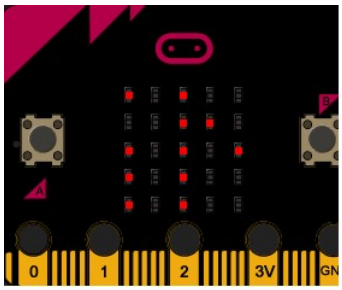
.. . . . .

Est-ce une interface Homme/Machine ?

.. . . . .

Exo n°2 : Afficher le texte « SNT » toutes les secondes (PAS D'INTERACTION POSSIBLE)





https://create.withcode.uk/

Décrire l'effet du programme sur la carte. Une interaction est-elle possible ?

. . . . . .

Préciser le rôle de chaque instruction

.. . . . .

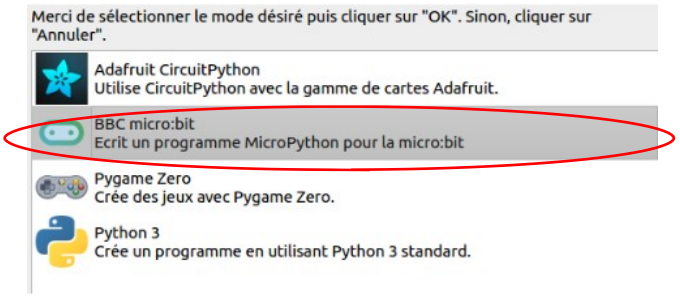
Est-ce une interface Homme/Machine ?

.. . . . .

ETAPE 3 - Mes premiers programmes réalisant des IHM :

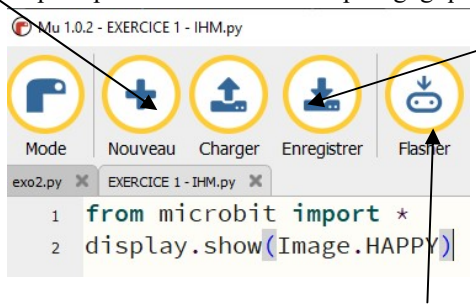
Exo n°1 : Afficher l'image « HAPPY » une seule fois (PAS D'INTERACTION POSSIBLE)

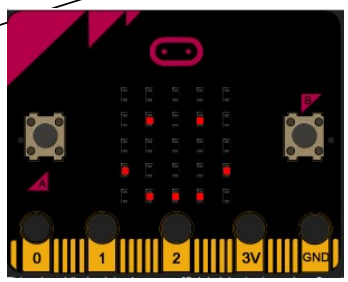
1. Lancer l'environnement de programmation **Mu** depuis le bureau et sélectionner le mode **BBC micro:bit.**



Nous programmerons la carte avec le langage Python et son module microbit

Sélectionner **Nouveau** dans la barre de menu pour créer un nouveau programme puis **enregistrer** le fichier sous le nom *exo1.py* dans un dossier pertinent de son espace personnel sur le réseau pédagogique.



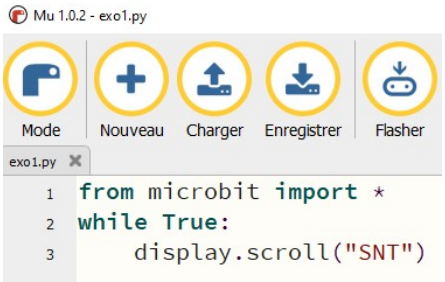


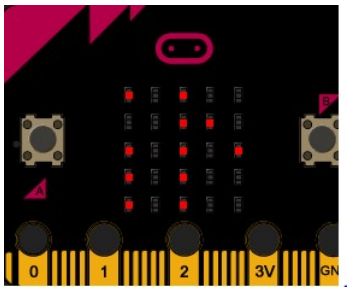
CORRECTION

https://create.withcode.uk/

Pour transférer le programme sur la carte, cliquer sur **Flasher**.
Lors de chaque téléversement la mémoire Flash contenant le programme exécuté par la carte est réinitialisée.
Décrire l'effet du programme sur la carte. Une interaction est-elle possible ?
Ce programme affiche un texte FIXE « CONTENT » - pas d'interaction possible
Préciser le rôle de chaque instruction
demande à MicroPython de récupérer tout ce dont il a besoin pour fonctionner avec le micro :bit
MicroPython est fourni avec beaucoup d'images intégrées à montrer sur l'affichage. « HAPPY »
Est-ce une interface Homme/Machine ?
OUI - C'est une IHM FIXE DE TYPE LED

Exo n°2 : Afficher le texte « SNT » toutes les secondes (PAS D'INTERACTION POSSIBLE)



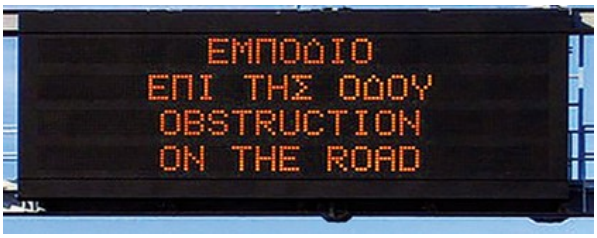


CORRECTION

https://create.withcode.uk/

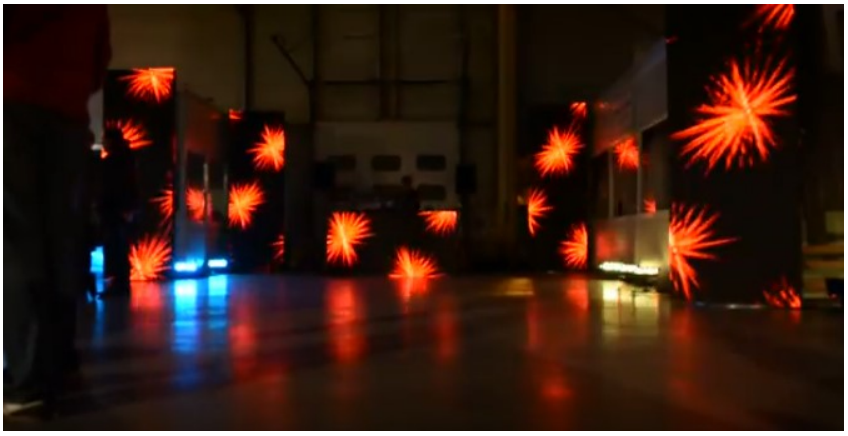
Décrire l'effet du programme sur la carte. Une interaction est-elle possible ?
Ce programme affiche un texte défilant « SNT » - pas d'interaction possible
Préciser le rôle de chaque instruction
demande à MicroPython de récupérer tout ce dont il a besoin pour fonctionner avec le micro :bit
Toujours faire défiler le texte « SNT » - voir https://microbit-micropython.readthedocs.io/en/latest/display.html
Est-ce une interface Homme/Machine ?
OUI - C'est une IHM A DEFILEMENT DE TEXTE

Exo n°1 : Afficher un texte fixe « panneaux LEDS » :



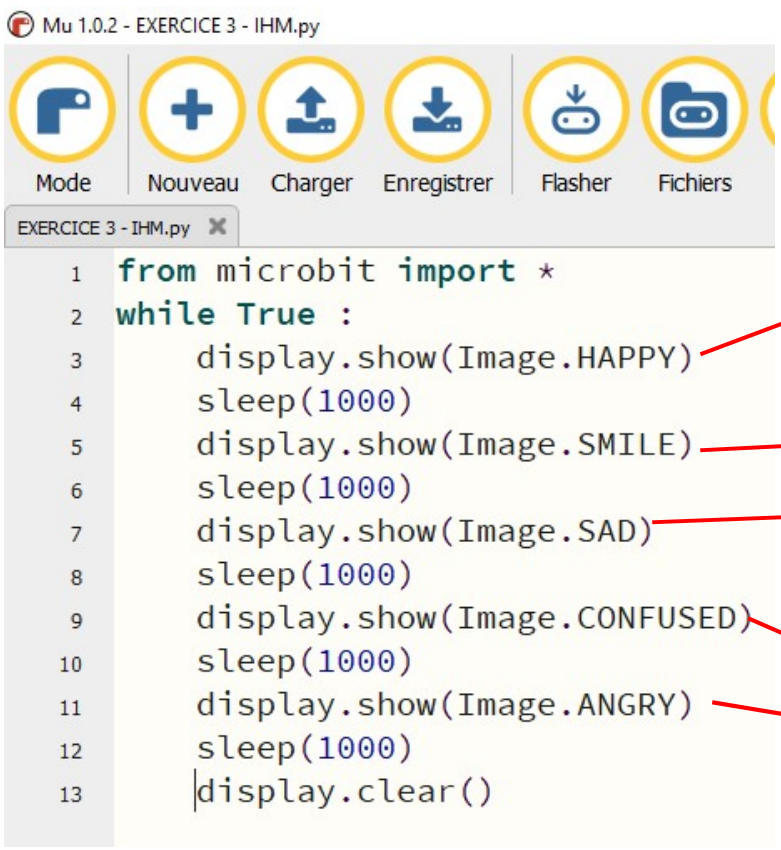
Exo n°2 : Afficher un texte défilant « panneaux LEDS » :

https://www.youtube.com/watch?v=H_y-FT0n_tU&feature=youtu.be

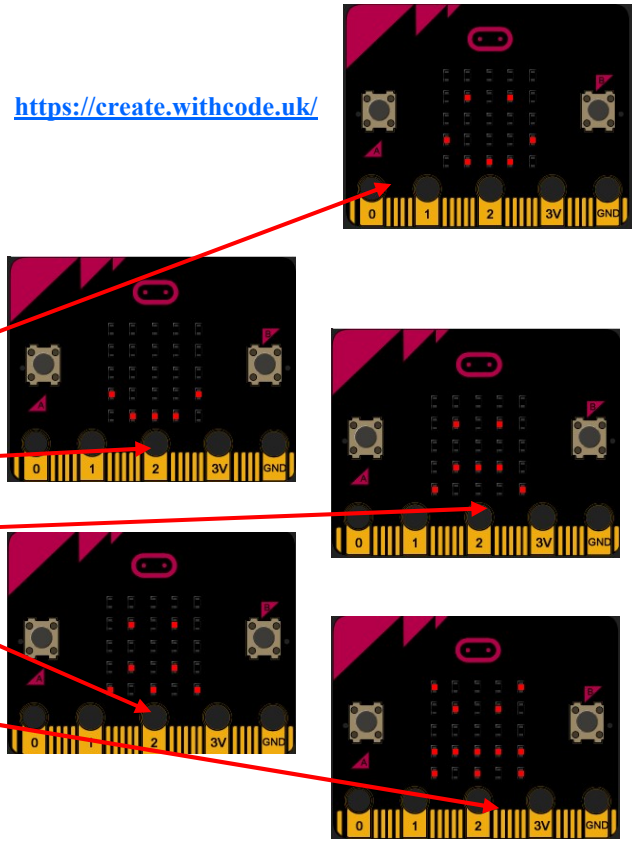


https://www.youtube.com/watch?v=L4QGh5Bm1TI

Exo n°3 : Afficher une image alternative (PAS D'INTERACTION POSSIBLE)



https://create.withcode.uk/



https://microbit-micropython.readthedocs.io/en/latest/tutorials/images.html

Décrire l'effet du programme sur la carte.

. . . . . .

Une interaction est-elle possible ?

. . . . . .

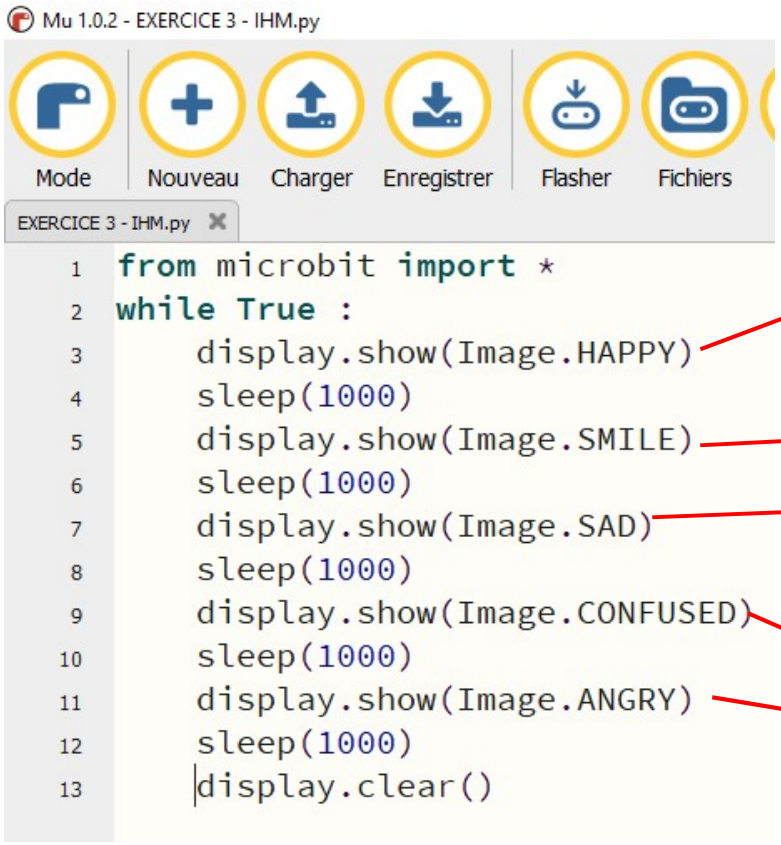
Préciser le rôle de chaque instruction

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

Est-ce une interface Homme/Machine ?

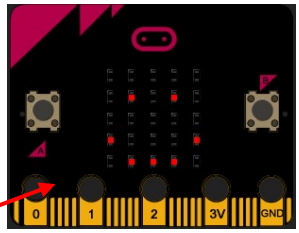
.. . . . .

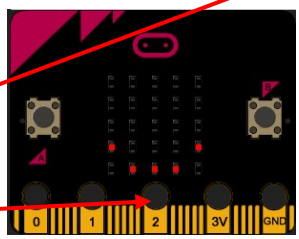
Exo n°3 : Afficher une image alternative (PAS D'INTERACTION POSSIBLE)

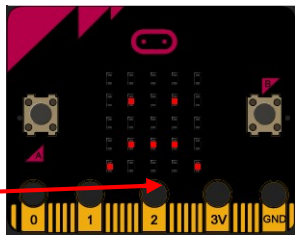


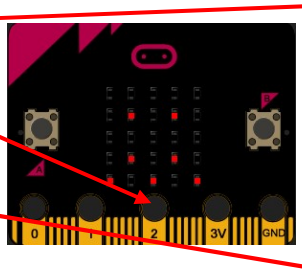
CORRECTION

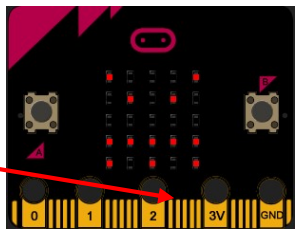
https://create.withcode.uk/











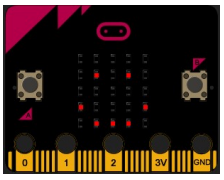
https://microbit-micropython.readthedocs.io/en/latest/tutorials/images.html

Décrire l'effet du programme sur la carte.
On observe en boucle l'affichage d'une série d'images—5 images

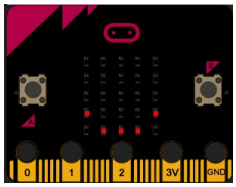
Une interaction est-elle possible ?
pas d'interaction possible

Préciser le rôle de chaque instruction

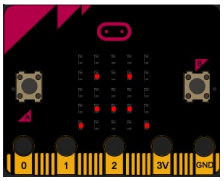
```
display.show(Image.HAPPY)
sleep(1000)
```



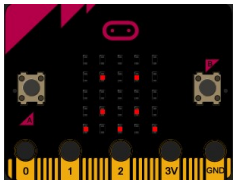
```
display.show(Image.SMILE)
sleep(1000)
```



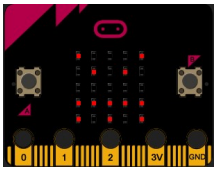
```
display.show(Image.SAD)
sleep(1000)
```



```
display.show(Image.CONFUSED)
sleep(1000)
```



```
display.show(Image.ANGRY)
sleep(1000)
```



https://create.withcode.uk/

CORRECTION

Est-ce une interface Homme/Machine ?
OUI - C'est une IHM A DEFILEMENT D'IMAGES

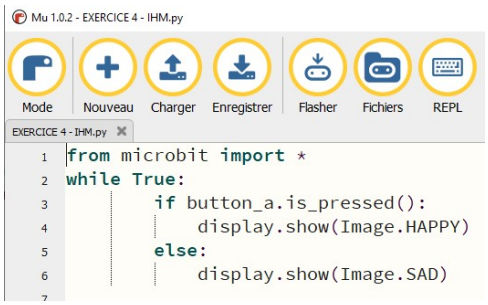
Exo n°3 : Afficher une image alternative sur un « panneaux LEDS » :

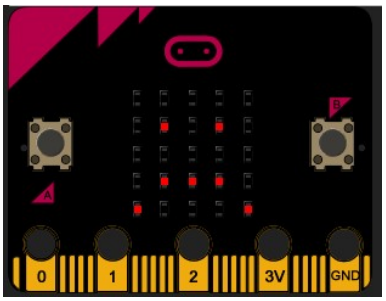
FAV néons

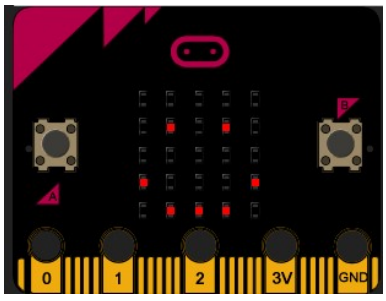
FAV à néons Silec en gare de péage sur l'A7.

FAV à néons à la gare de péage de Coutevroult (A4)

Exo n°4 : Afficher une image en fonction du bouton A : (AVEC INTERACTION POSSIBLE)







Décrire l'effet du programme sur la carte.

. . . . . .

Une interaction est-elle possible ?

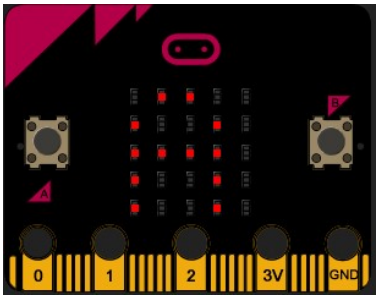
. . . . . .

Préciser le rôle de chaque instruction. Identifier capteur et actionneur.

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

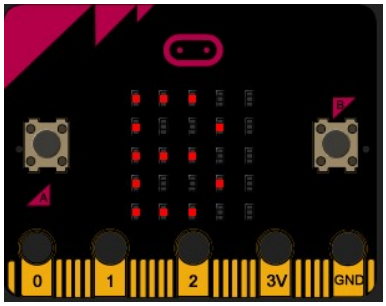
https://microbit-micropython.readthedocs.io/en/latest/tutorials/buttons.html

Exo n°5 : Afficher A OU B en fonction des boutons A OU B : (AVEC INTERACTION POSSIBLE)



EXERCICE 5 - IHM.py

```
from microbit import *
while True:
    if button_a.is_pressed():
        display.show('A')
    elif button_b.is_pressed():
        display.show('B')
    else:
        display.show('A OU B')
```



https://create.withcode.uk/

Décrire l'effet du programme sur la carte.

. . . . . .

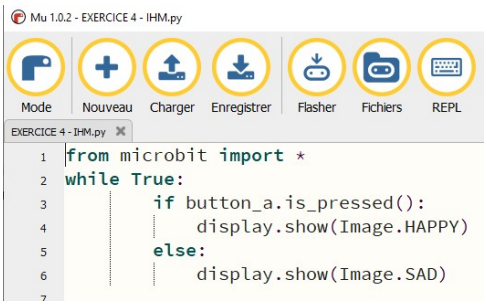
Une interaction est-elle possible ?

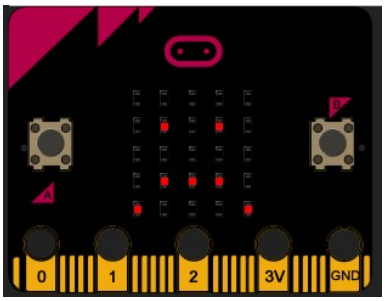
. . . . . .

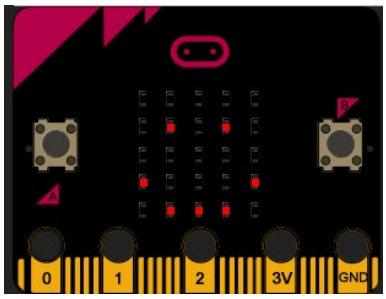
Identifier capteurs et actionneur.

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

Exo n°4 : Afficher une image en fonction du bouton A : (AVEC INTERACTION POSSIBLE)







Décrire l'effet du programme sur la carte.
Si on appuie sur le bouton A on obtient l'image CONTENT, sinon on affiche l'image PAS CONTENT

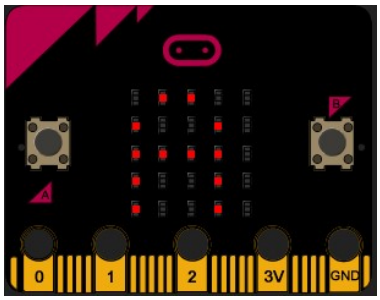
Une interaction est-elle possible ?
L'interaction se fait sur le bouton A

CORRECTION

Préciser le rôle de chaque instruction. Identifier capteur et actionneur.
On répète indéfiniment, si on appuie sur A on affiche l'image HAPPY, sinon on affiche l'image SAD
Le capteur est le bouton A - l'actionneur est l'afficheur 25 LEDS.

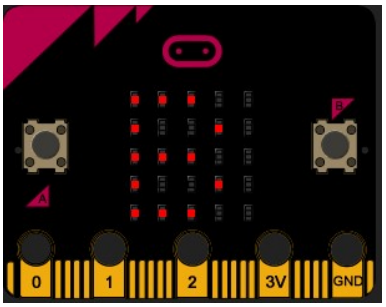
https://microbit-micropython.readthedocs.io/en/latest/tutorials/buttons.html

Exo n°5 : Afficher A OU B en fonction des boutons A OU B : (AVEC INTERACTION POSSIBLE)



EXERCICE 5 - IHM.py

```
from microbit import *
while True:
    if button_a.is_pressed():
        display.show('A')
    elif button_b.is_pressed():
        display.show('B')
    else:
        display.show('A OU B')
```

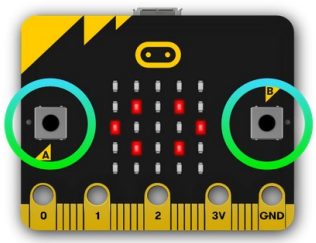


https://create.withcode.uk/

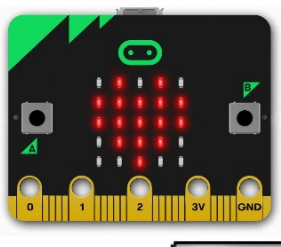
Décrire l'effet du programme sur la carte.
Si on appuie sur le bouton A on obtient la lettre A, Si on appuie sur le bouton B on obtient la lettre B
Sinon on obtient le texte A ou B

Une interaction est-elle possible ?
Les interactions se font sur le bouton A ou B

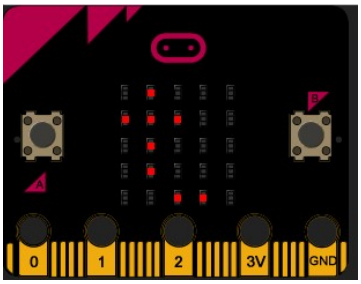
Identifier capteurs et actionneur.
Les capteurs sont les boutons A et B **l'actionneur est l'afficheur 25 LEDS.**



https://microbit.org/fr/guide/features/#light



Exo n°6 : Afficher la température :



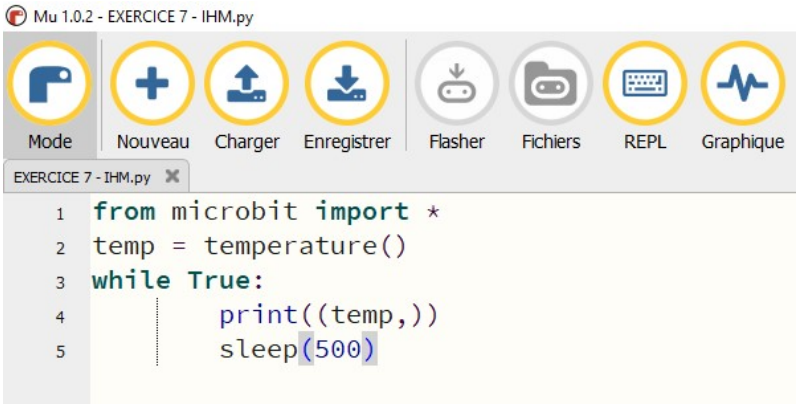

```
EXERCICE 6 - IHM.py
1 from microbit import *
2 temp = temperature()
3 while True:
4         display.show('t')
5         sleep(1000)
6         display.show(temperature())
7         sleep(1000)
```

https://create.withcode.uk/

Décrire l'effet du programme sur la carte.

. . . . . .

Exo n°7 : Afficher sur MU les valeurs de température :



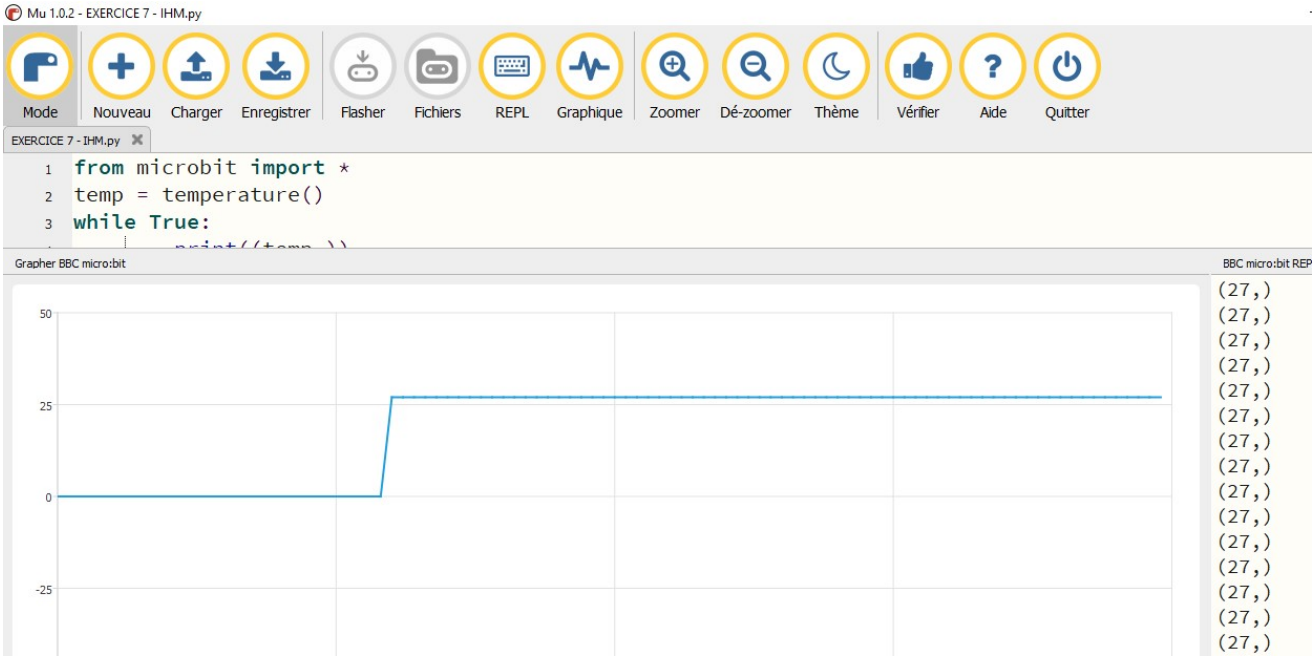
Flasher le programme

Puis Faire un RESET

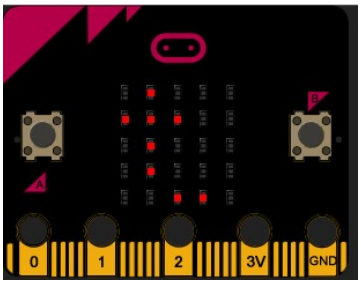
Cliquer sur REPL et GRAPHIQUE

Activer le graphique et le REPL : faire un commentaire

. . . . . .



Exo n°6 : Afficher la température :



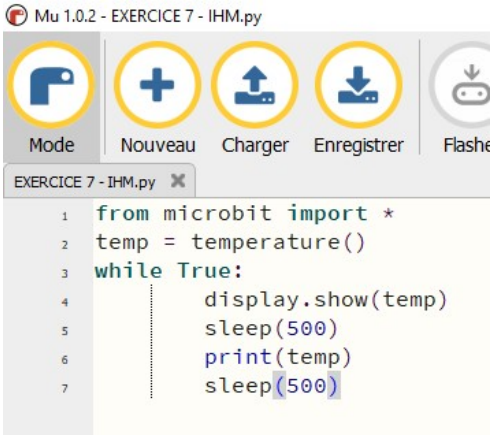

```
EXERCICE 6 - IHM.py
1 from microbit import *
2 temp = temperature()
3 while True:
4         display.show('t')
5         sleep(1000)
6         display.show(temperature())
7         sleep(1000)
```

CORRECTION

https://create.withcode.uk/

Décrire l'effet du programme sur la carte.
La matrice à Led affiche la lettre t puis la température en degrés toutes les secondes

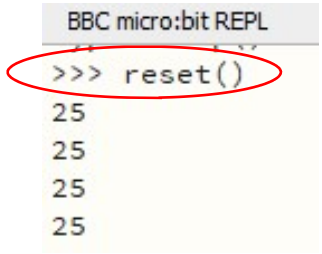
Exo n°7 : Afficher sur MU les valeurs de température :



Flasher le programme

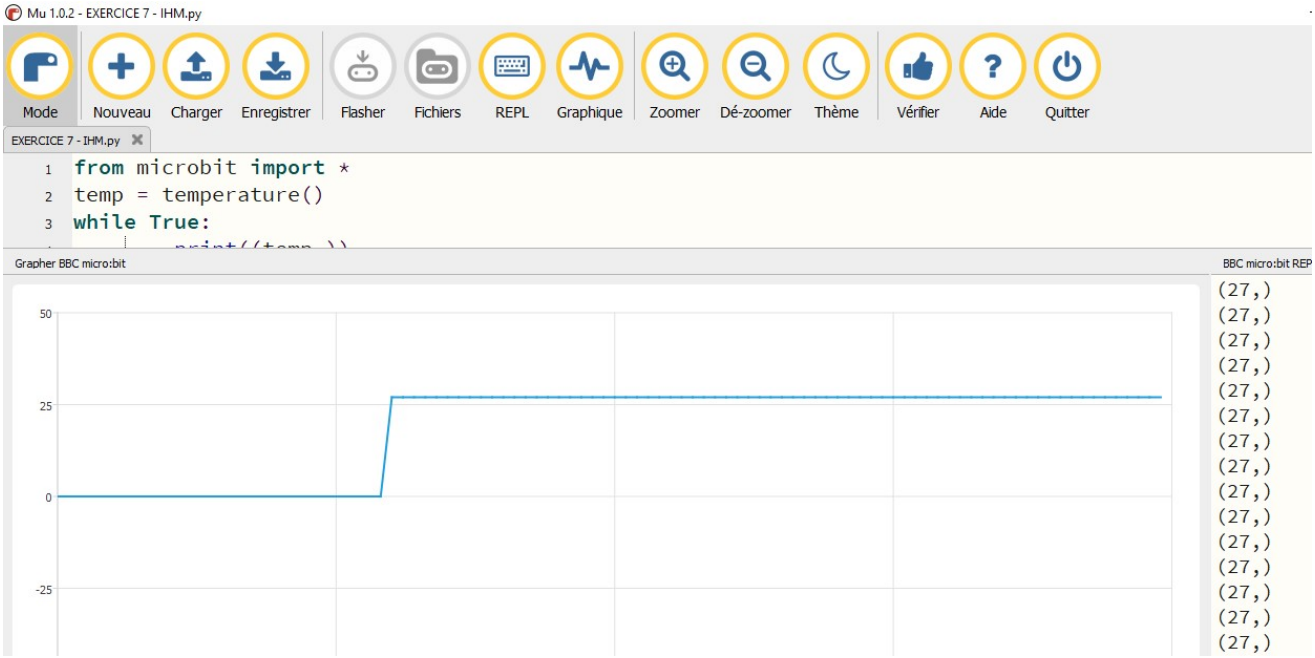
Puis Faire un RESET

Cliquer sur REPL et GRAPHIQUE



CORRECTION

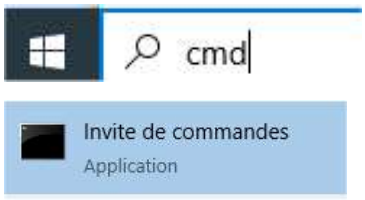
Activer le graphique et le REPL : faire un commentaire
Les valeurs de température s'affiche dans le REPL et le graphique affiche les valeurs de températures toutes les 1/2 secondes



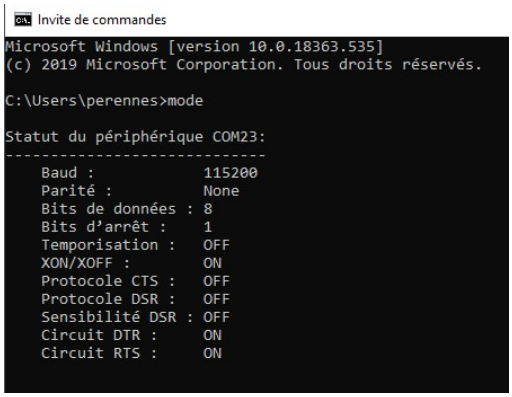
Exo n°8 : Comment lire les valeurs de température du port série avec putty :

Comment savoir quel port USB (com) est utilisé ?

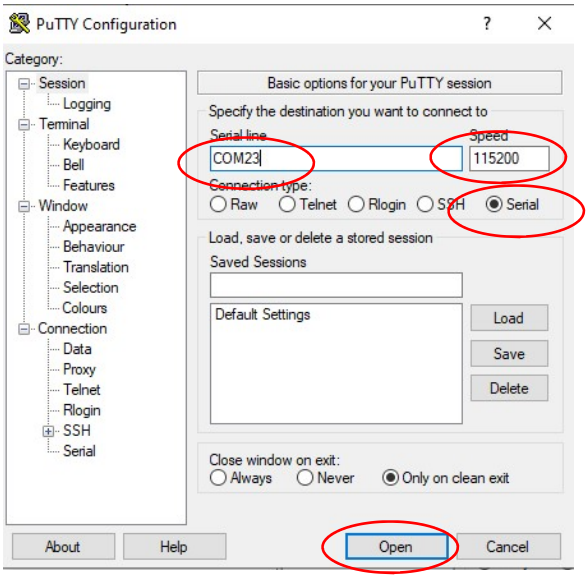
Ouvrir un fenêtre de commande (cmd)

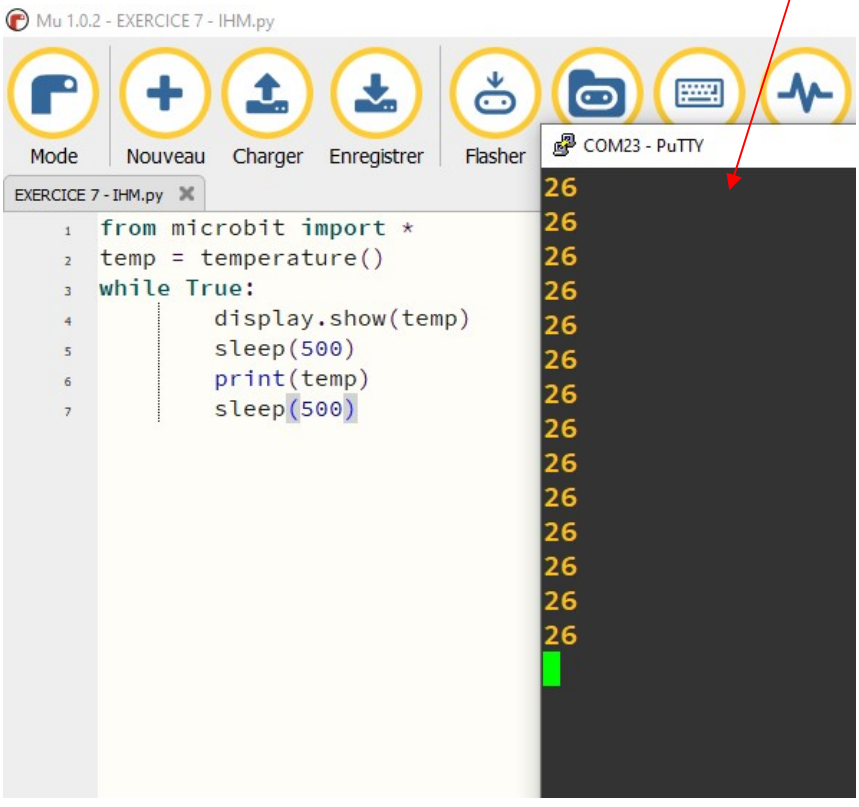


Puis taper mode



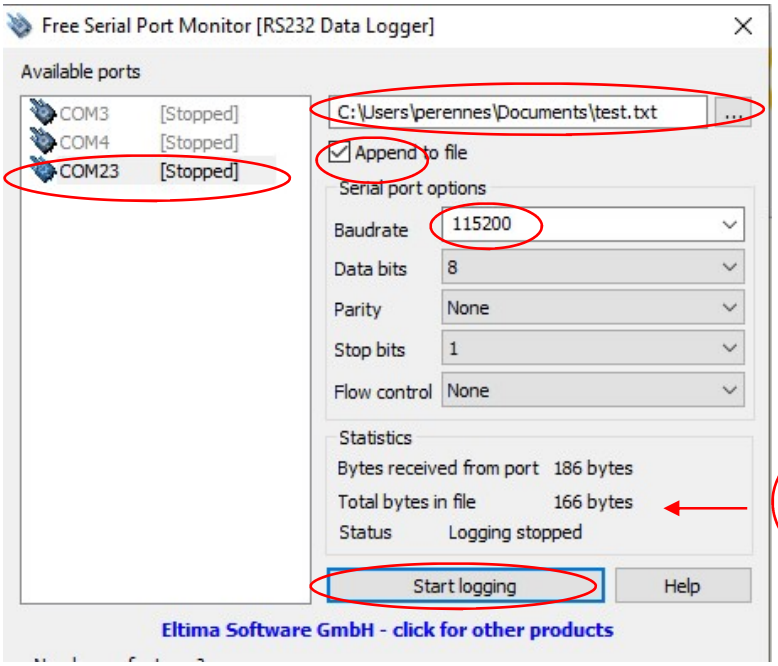
Entrer ces paramètres dans PUTTY

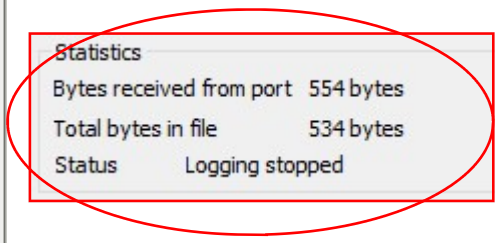


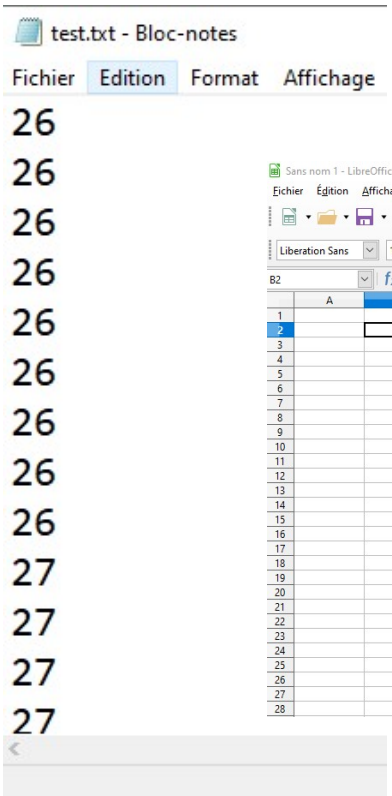


https://support.microbit.org/support/solutions/articles/19000022103-outputing-serial-data-from-the-micro-bit-to-a-computer

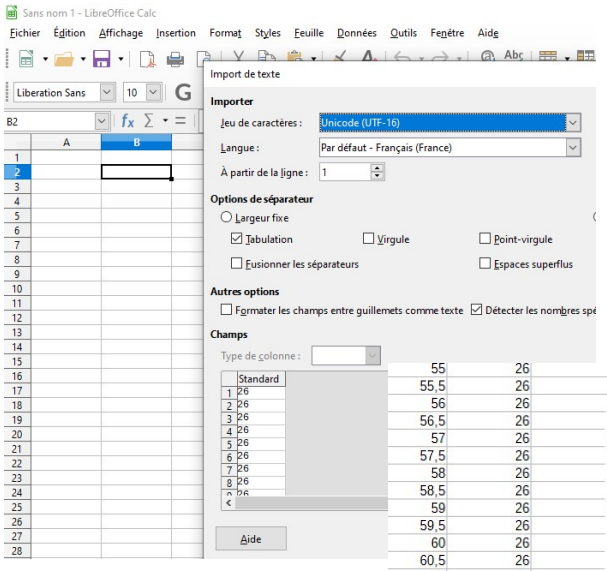
Exo n°9 : Comment extraire les valeurs de température du port série avec free serial monitor :



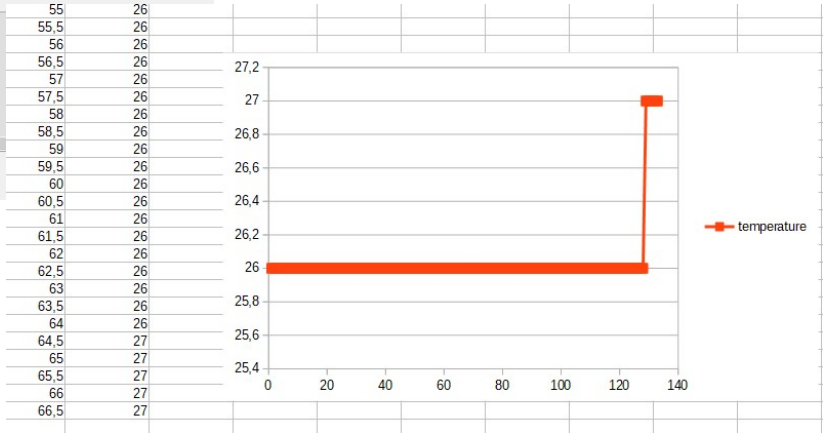




Copier les valeurs du fichier texte
Dans un tableur



Réaliser le graphique de température



Ce document a été inspiré entre autres par la lecture de l'article :

https://frederic-junier.org/SNT/Theme3_InformatiqueEmbarquee/ressources/Activite2-Microbit-2019V1.pdf

ETAPE 1 : Qu'est-ce qu'un système embarqué ? :

_____

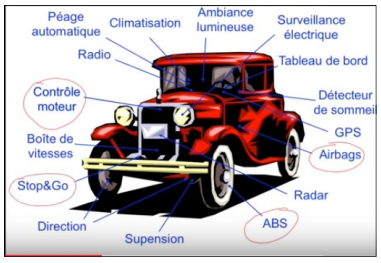
_____

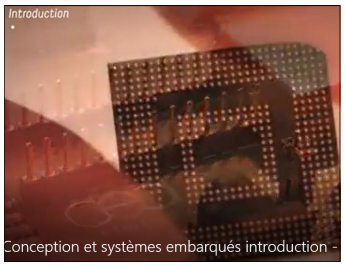
_____

_____

Lire les vidéos :

https://www.youtube.com/watch?v=H9iKiqeivg0 Et https://www.youtube.com/watch?v=0KQWaWVRtvY





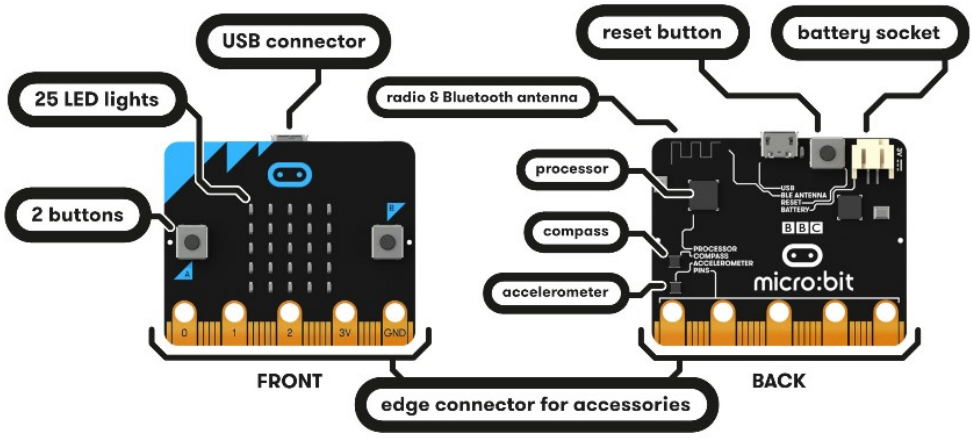
ETAPE 2 : Présentation de la carte BBC :

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/ Et **https://microbit-micropython.readthedocs.io/en/latest/**



https://microbit.org/fr/guide/features/

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

(Si on veut intégrer la carte dans un système embarqué, il est possible de la connecter à une alimentation externe **par piles ou batterie LIPO.)**

Ce document a été inspiré entre autres par la lecture de l'article :

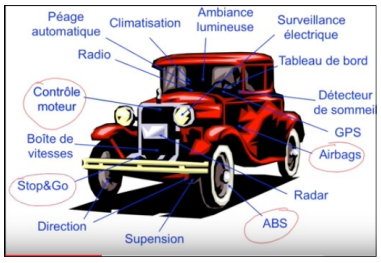
https://frederic-junier.org/SNT/Theme3_InformatiqueEmbarquee/ressources/Activite2-Microbit-2019V1.pdf

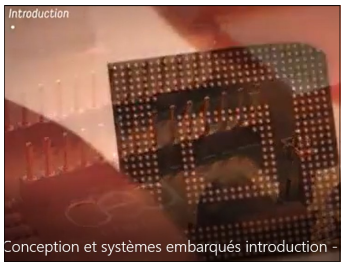
ETAPE 1 : Qu'est-ce qu'un système embarqué ? :

Un **système informatique embarqué** collecte des informations du monde réel à l'aide de **capteurs**, les traite dans un **micorprocesseur** 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**.

Lire les vidéos :

https://www.youtube.com/watch?v=H9iKiqeivg0 Et https://www.youtube.com/watch?v=0KQWaWVRtvY





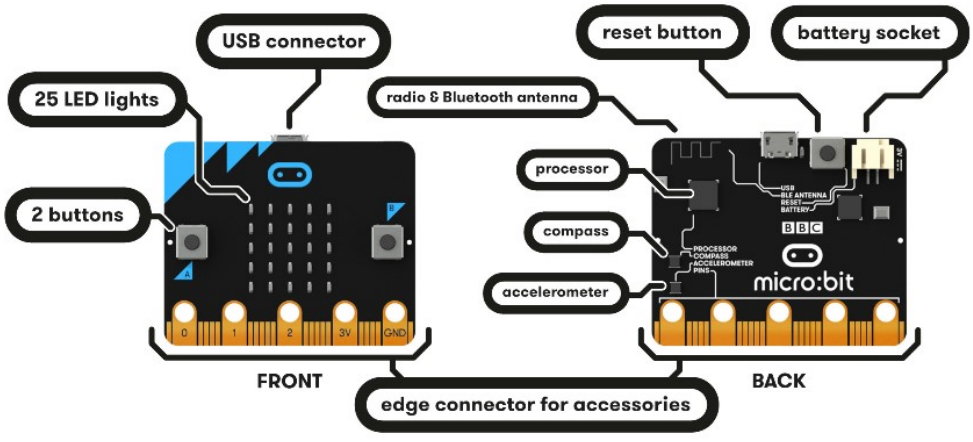
ETAPE 2 : Présentation de la carte BBC :

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/ Et **https://microbit-micropython.readthedocs.io/en/latest/**



https://microbit.org/fr/guide/features/

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

(Si on veut intégrer la carte dans un système embarqué, il est possible de la connecter à une alimentation externe **par piles ou batterie LIPO.)**

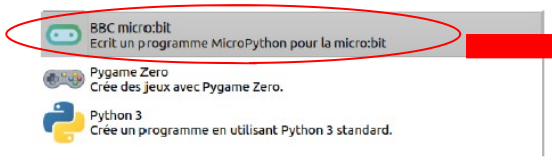
ETAPE 3 - Mes premiers programmes réalisant des IHM :

Exo n°1 : Comment lire la température avec TKINTER ? :

https://python.doctor/page-tkinter-interface-graphique-python-tutoriel

Partie A : réalisation du système embarqué

⚠ **Dans cette partie, on utilise Mu en mode `micro:bit`.**



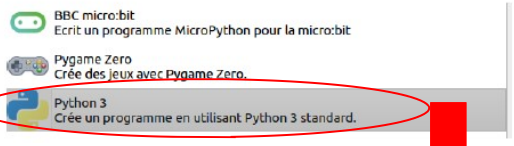
EXERCICE 7 - IHM.py

```
from microbit import *
temp = temperature()
while True:
        display.show(temp)
        sleep(500)
        print(temp)
        sleep(500)
```

transférer sur la carte microbit

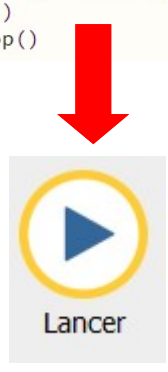
Partie B : réalisation de l'IHM

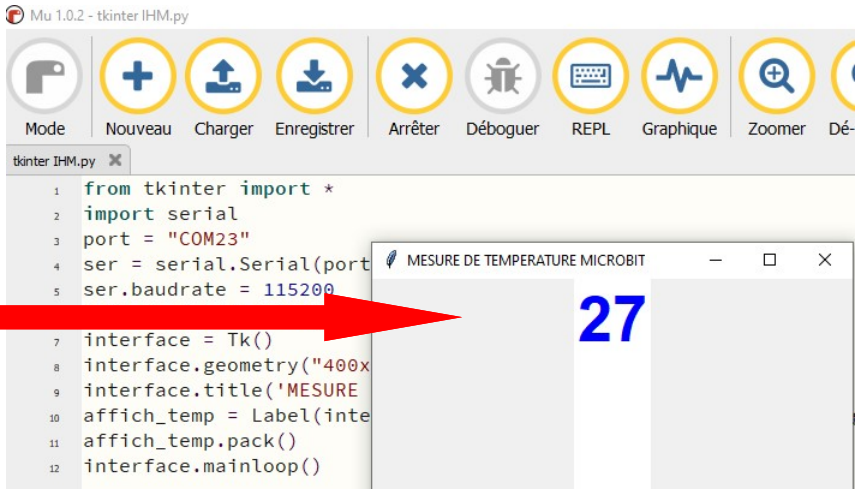
⚠ **Dans cette partie, on utilise Mu en mode Python.**



tkinter IHM.py

```
from tkinter import *
import serial
port = "COM23"
ser = serial.Serial(port)
ser.baudrate = 115200
data = ser.readline()
interface = Tk()
interface.geometry("400x300+300+200")
interface.title('MESURE DE TEMPERATURE MICROBIT')
affich_temp = Label(interface, text= data, width=0, bg='white',fg='blue',font='arial 40 bold')
affich_temp.pack()
interface.mainloop()
```





On obtient la fenêtre IHM avec la température de la carte

https://frederic-junier.org/SNT/Theme3_InformatiqueEmbarquee/ressources/Activite2-Microbit-2019V1.pdf

PERENNES Philippe
Collège de SAINT JAMES

THEME :
Informatique embarquée et objets connectés
Réaliser une IHM simple d'un système embarqué

Activité branchée

RESSOURCE 1

MEMENTO TKINTER :

https://python.doctor/page-tkinter-interface-graphique-python-tutoriel

Importation : from tkinter import *

WIDGETS CONTENEURS

Fenêtre pricipale
fen=Tk() et à la fin ne pas oublier fen.mainloop()
fen.title(' titre de la fenêtre ') pour mettre un titre !

Sous-fenêtre Frame
cadre1=Frame(fen, bd=4,bg='white',relief=RAISED)
cadre1.pack()
Reliefs possibles : RAISED, FLAT, SUNKEN,RIDGE,SOLID,.....

Pour les dessins Canvas
dessin=Canvas(fen, width=800, height=600)
dessin.pack()

POSITIONNEMENT

pack et grid (pas les deux à la fois dans un même conteneur!)
pack() : l'ordinateur fait comme il veut
grid() : positionnement par ligne et colonne
exemple : grid(row=2, column=3)
Étalement sur plusieurs colonnes ou lignes **columnspan** et **rowspan**

PRINCIPAUX WIDGETS

Label pour afficher des textes non modifiables par l'utilisateur
lab1=Label(fen, text= 'NOMBRE', width=10, bg='white',fg='red',font='arial 20 bold')
lab1.grid(row=0,column=0)
Modifier un label :
lab1.config(text=...., bg=.....)

Entry pour entrer des données
entr1=Entry(fen, width=15)
entr1.pack()
Récupérer le contenu d'un Entry :
texte=entr1.get() (c'est une chaîne !, penser aux conversions str, int, eval,...)

Button pour déclencher une action
bout=Button(fen, text= 'Calculer ', command=action)
bout.pack()
(action est une fonction définie au début du programme par def....)

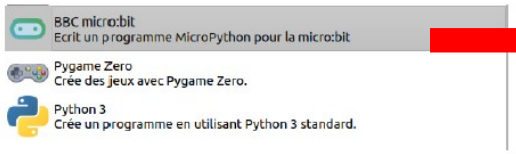
Remarque : créer une variable, ou plusieurs, pour régler la taille des polices dans tous les widgets, exemple police1='arial,20 bold', puis dans les widgets, font=police1

Exo n°2 : Comment envoyer un chiffre à la MICROBIT avec TKINTER ? :

transférer sur la carte microbit

Partie A : réalisation du système embarqué

Dans cette partie, on utilise Mu en mode `micro:bit`.

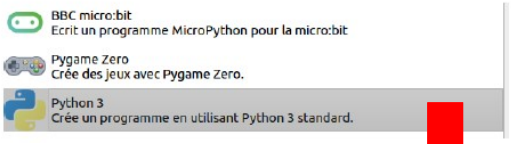


reception message.py

```
from microbit import *
uart.init(baudrate=115200)
while True:
    if uart.any():
        msg_bytes = uart.read()
        msg = str(msg_bytes, "utf-8")
        x = int(msg[0])
        display.scroll(x)
sleep(1000)
display.clear()
```

Partie B : réalisation de l'IHM

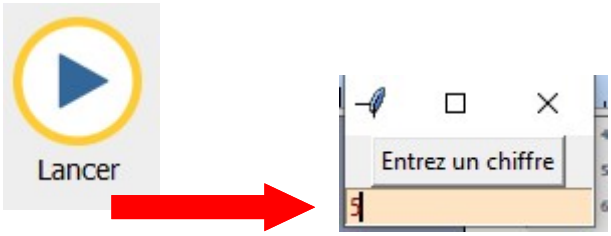
Dans cette partie, on utilise Mu en mode `Python`.



envoi message.py

```
from tkinter import *
import serial
port = "COM23"
ser = serial.Serial(port)
ser.baudrate = 115200
def envoie_message():
    msg = texte_message.get()
    message_bytes = bytes(msg, "utf-8")
    ser.write(message_bytes)
ma_fenetre = Tk()
ma_fenetre.title("IHM")
button_message = Button(ma_fenetre, text ="Entrez un chiffre", command = envoie_message)
button_message.pack()
texte_message = StringVar()
champ_message = Entry(ma_fenetre, textvariable= texte_message, bg ="bisque", fg="maroon", width="20")
champ_message.focus_set()
champ_message.pack()
ma_fenetre.mainloop()
```

Mode python 3—Lancer



https://frederic-junier.org/SNT/Theme3_InformatiqueEmbarquee/ressources/Activite2-Microbit-2019V1.pdf

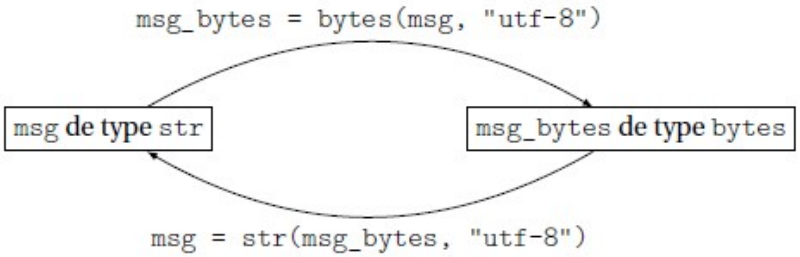
La réalisation de l'interface graphique est commentée dans le code, elle s'appuie sur le module tkinter qu'on charge avec from tkinter import *.

La communication série avec la carte micro:bit s'appuie sur le module serial qu'on charge avec from serial import *.

Méthode *Communication série côté* `ordinateur`

Le module `serial` permet de communiquer avec une carte connectée sur le port série `USB`.

- `serie = Serial(port)` permet d'ouvrir une connexion série sur le port de l'ordinateur où la carte est connectée. Sous Windows ce port peut être découvert avec la commande `mode` depuis la console lancée avec `cmd` dans la barre de recherche. Si l'accès n'est pas autorisé, il suffit de tester les différentes possibilités : `"COM3"`, `"COM4"` ou `"COM5"`.
- On commence par fixer la vitesse de transmission `serie.baudrate` à la même valeur que celle choisie sur la carte.
- Comme sur la carte les messages transmis par communication série sont nécessairement en `bytes`, ce qui nécessite une conversion entre chaîne de caractères de type `str` et `bytes` avant transmission ou après réception.



- On lit un message en `bytes` sur l'entrée série, avec `serie.read()` ou `serie.readline()` si on veut s'arrêter au prochain saut de ligne.
- On écrit un message en `bytes` sur l'entrée série, avec `serie.write(msg_bytes)`

https://frederic-junier.org/SNT/Theme3_InformatiqueEmbarquee/ressources/Activite2-Microbit-2019V1.pdf

Ce document a été inspiré entre autres par la lecture de l'article :

https://frederic-junier.org/SNT/Theme3_InformatiqueEmbarquee/ressources/Activite2-Microbit-2019V1.pdf

ETAPE 1 : Qu'est-ce qu'un objet connecté ? :

_____

_____

_____

_____

Lire les vidéos :

https://www.youtube.com/watch?v=8sgmHTHSqxw Et **https://www.youtube.com/watch?time_continue=24&v=p3FWAgGwr6c&feature=emb_logo**

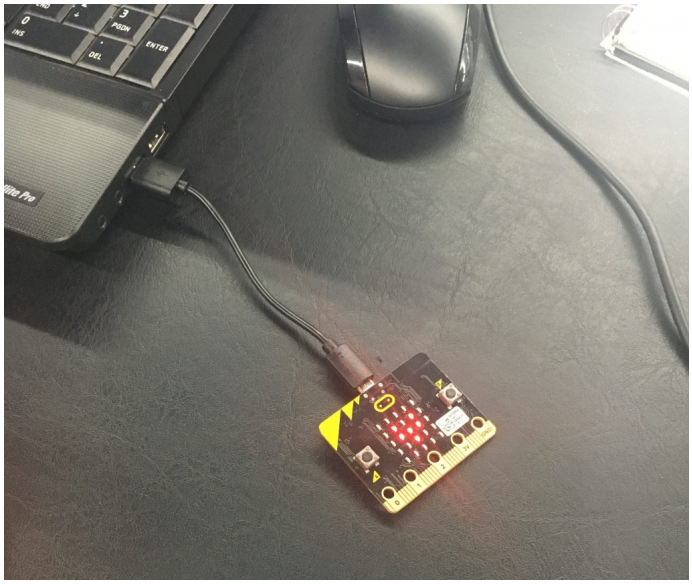
ETAPE 2 : Présentation du matériel nécessaire :

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

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

https://microbit.org/guide/hardware/usb/

Comme on ne dispose pas du module d'extension Wifi pour micro:bit, on utilise un programme sur l'ordinateur connecté pour servir de passerelle entre la carte et l'Internet.



Ce document a été inspiré entre autres par la lecture de l'article :

https://frederic-junier.org/SNT/Theme3_InformatiqueEmbarquee/ressources/Activite2-Microbit-2019V1.pdf

ETAPE 1 : Qu'est-ce qu'un objet connecté ? :

Un objet connecté est un système informatique embarqué **disposant d'une connexion à un réseau local ou à l'Internet.** Actuellement, il existe plus d'objets que d'humains connectés à Internet et leur nombre va augmenter fortement dans les prochaines années. **On parle d'IOT pour Internet Of Things.**

Lire les vidéos :

https://www.youtube.com/watch?v=8sgmHTHSqxw Et **https://www.youtube.com/watch?time_continue=24&v=p3FWAgGwr6c&feature=emb_logo**

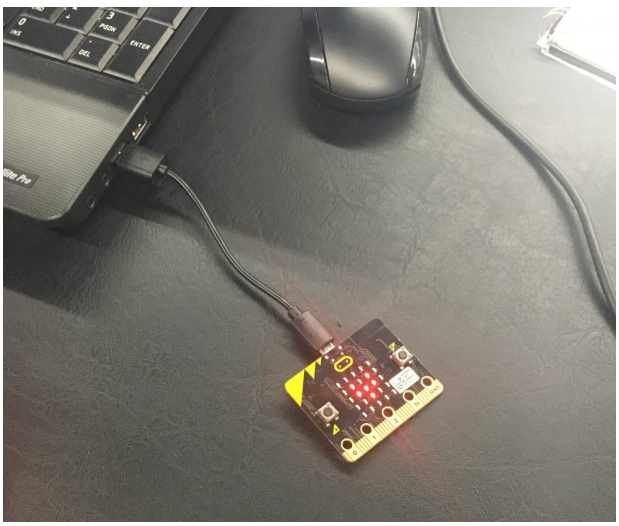
ETAPE 2 : Présentation du matériel nécessaire :

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

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

https://microbit.org/guide/hardware/usb/

Comme on ne dispose pas du module d'extension Wifi pour micro:bit, on utilise un programme sur l'ordinateur connecté pour servir de passerelle entre la carte et l'Internet.

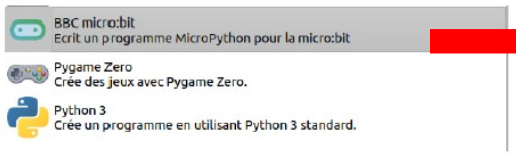


ETAPE 3 - Mes premiers programmes réalisant des IHM :

Exo n°1 : Comment envoyer l'API sur Internet avec le logiciel MU ? :

Partie A : réalisation du système embarqué

⚠ **Dans cette partie, on utilise Mu en mode `micro:bit`.**



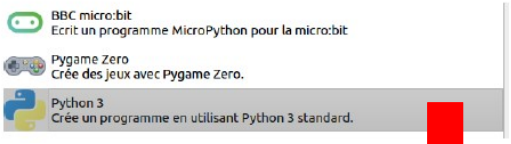
transférer sur la carte microbit

test API.py | EXERCICE 7 - IHM.py

```
1 from microbit import *
2 temp = temperature()
3 while True:
4         display.show(temp)
5         sleep(500)
6         print(temp)
7         sleep(500)
```

Partie B : réalisation de l'IHM

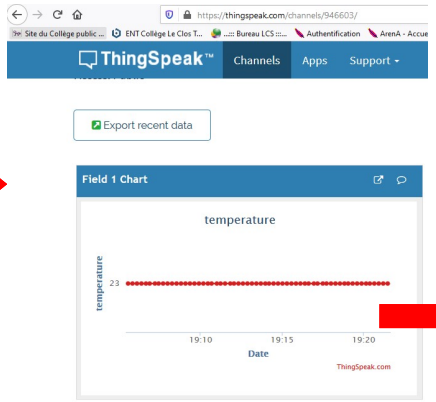
⚠ **Dans cette partie, on utilise Mu en mode `Python`.**



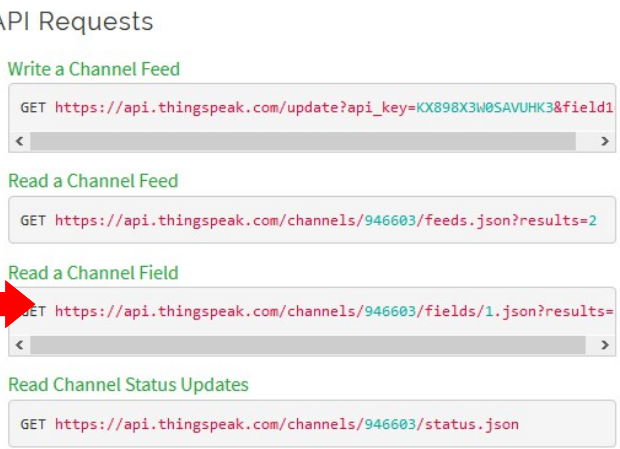
test API.py

```
1  from serial import *
2  from requests import *
3  import serial
4  port = "COM23"
5  ser = serial.Serial(port)
6  ser.baudrate = 115200
7  apiKey = "KX898X3W0SAVUHK3"
8  while True:
9      msg_bytes = ser.readline().strip()
10     msg = str(msg_bytes, "utf-8")
11     r = get("https://api.thingspeak.com/update.json", params = {"field1" : msg , "key" : apiKey})
```

Mode python 3—Lancer



APERCU SUR TINGSPEAK



https://thingspeak.com/channels/946603

ADRESSES DES API

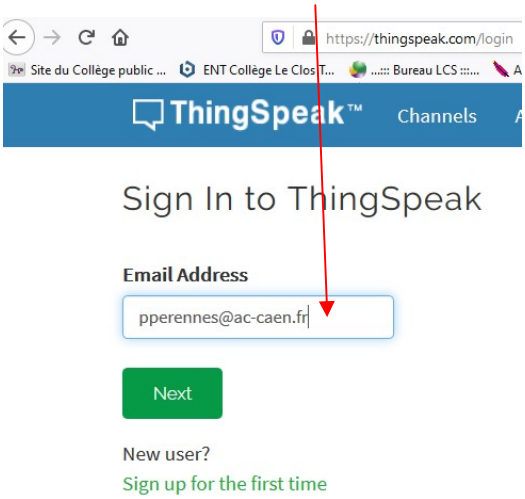
https://frederic-junier.org/SNT/Theme3_InformatiqueEmbarquee/ressources/Activite2-Microbit-2019V1.pdf

Comment paramétrer THINGSPEAK ? :

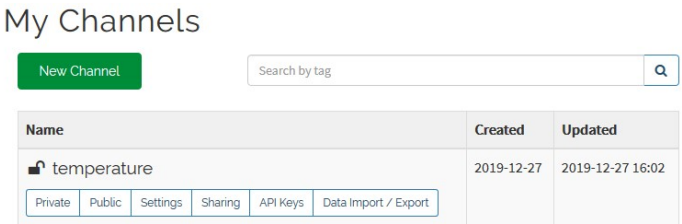
https://sites.google.com/sepne.ca/jardins-intelligents-du-sommet/envoyer-des-donn%C3%A9es-%C3%A0-thingspeak-avec-arduino

https://www.youtube.com/watch?v=Mx9B7aSSMjc

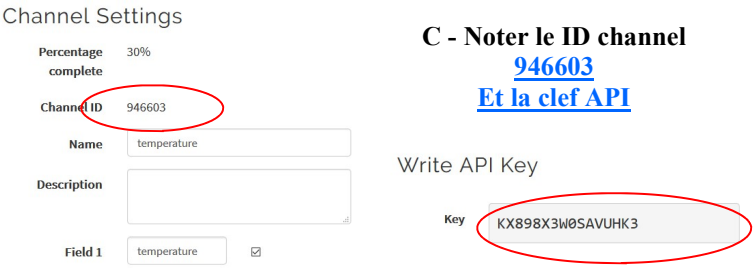
A - Réaliser un compte



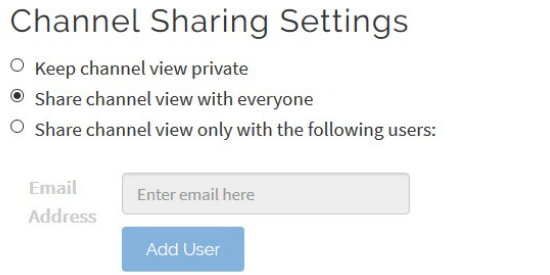
B - Créer un channel

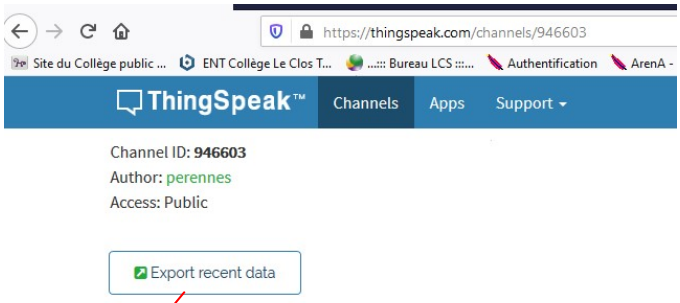


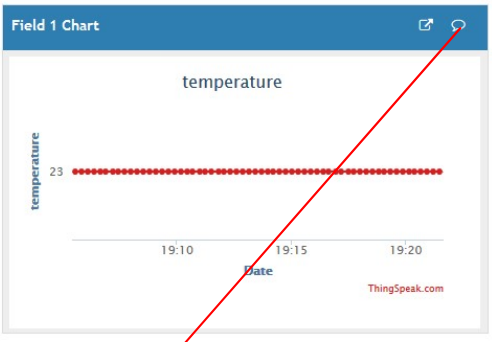
C - Noter le ID channel
946603
Et la clef API

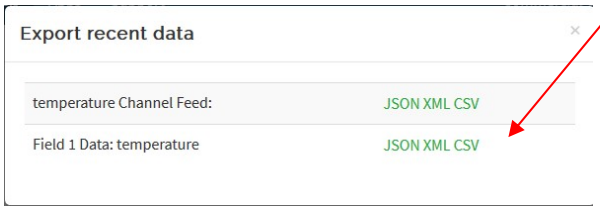


C - Partager les résultats
Adresse du type
https://thingspeak.com/channels/ID
https://thingspeak.com/channels/946603





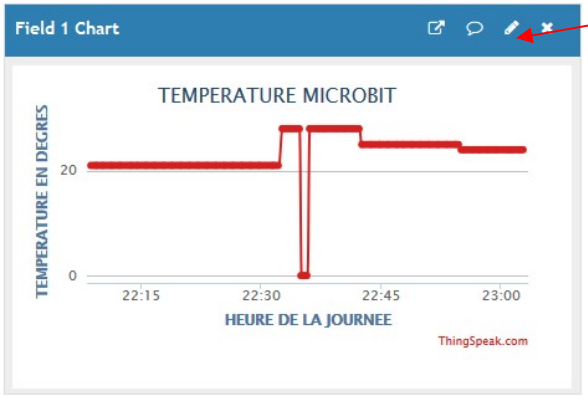


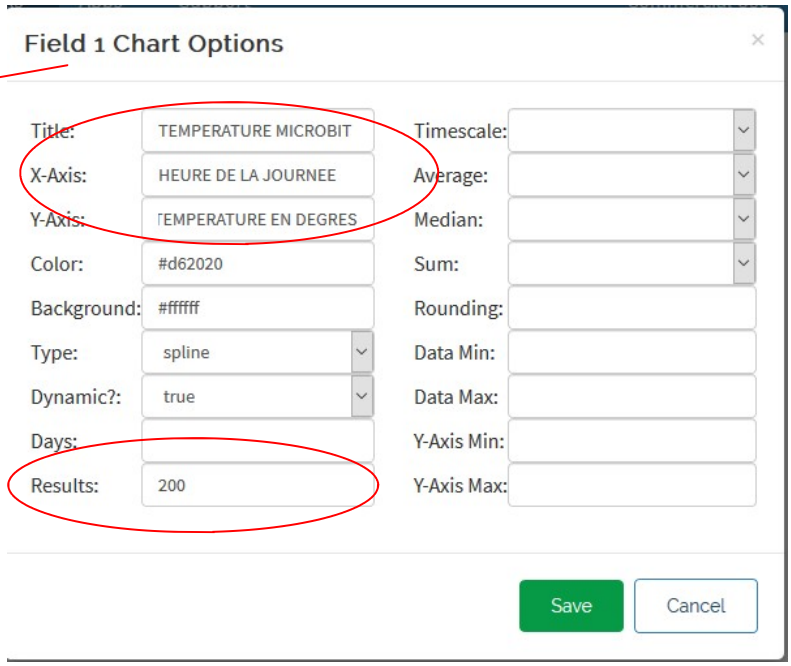


D– Exporter les résultats
Dans les formats
JSON - XML - CSV

E– Récupérer l'iframe pour l'inclure dans un site Internet

Comment paramétrer THINGSPEAK ? :

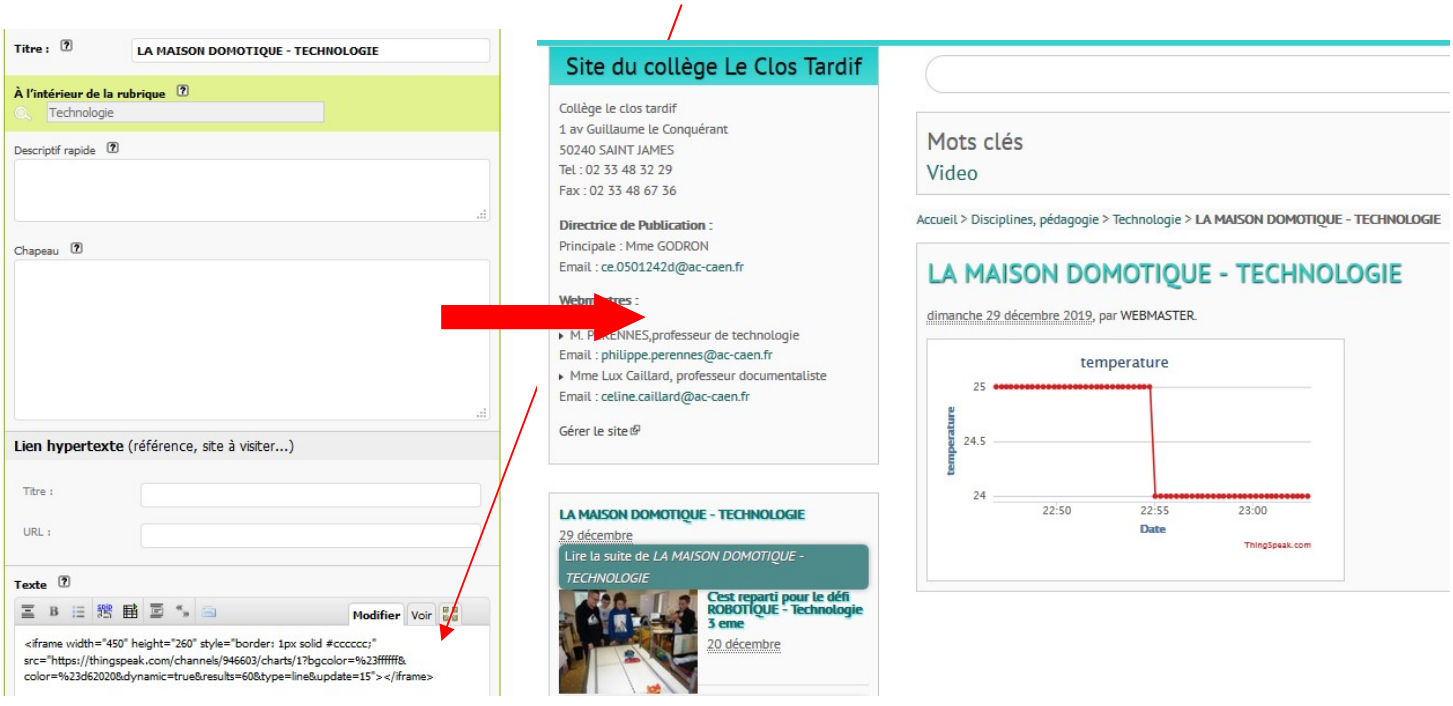




Inclure l'iframe dans SPIP :

Récupérer l'iframe pour l'inclure dans un site Internet

<iframe width="450" height="260" style="border: 1px solid #cccccc;" src="https://thingspeak.com/channels/946603/charts/1?bgcolor=%23ffffff&color=%23d62020&dynamic=true&results=200&title=TEMPERATURE+MICROBIT&type=spline&xaxis=HEURE+DE+LA+JOURNEE&yaxis=TEMPERATURE+EN+DEGRES"></iframe>

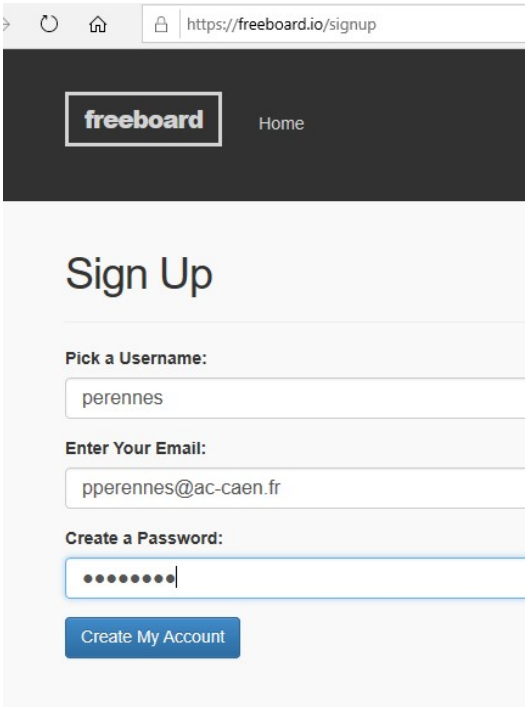


Comment récupérer les API sur FREEBOARD ? :

https://liveobjects.orange-business.com/cms/app/uploads/How-to-create-a-beautiful-Freeboard-dashboard-for-Live-Objects-1.pdf

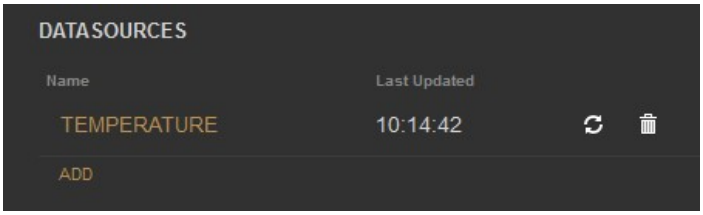
https://academy.visiplus.com/blog/analytics-2/data-visualisation-5-outils-de-dashboard-open-source-incontournables-2017-06-12

A - Réaliser un compte

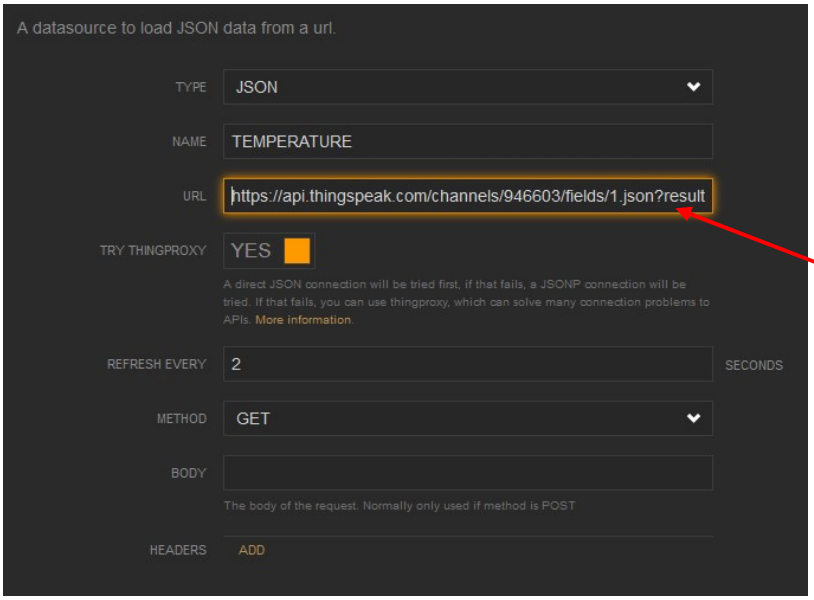


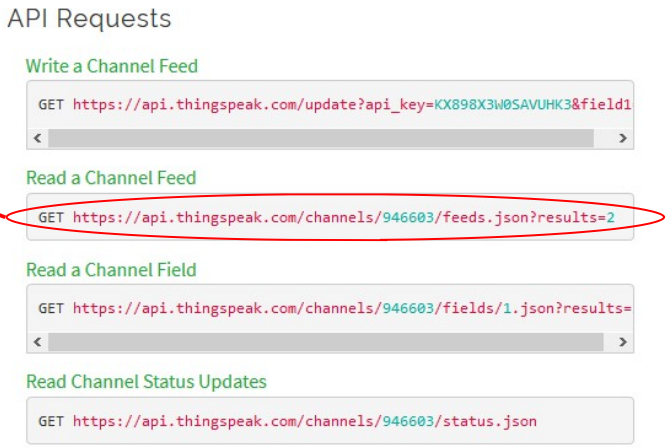
B - Créer un FREEBOARD

C - Ajouter les DATA SOURCES

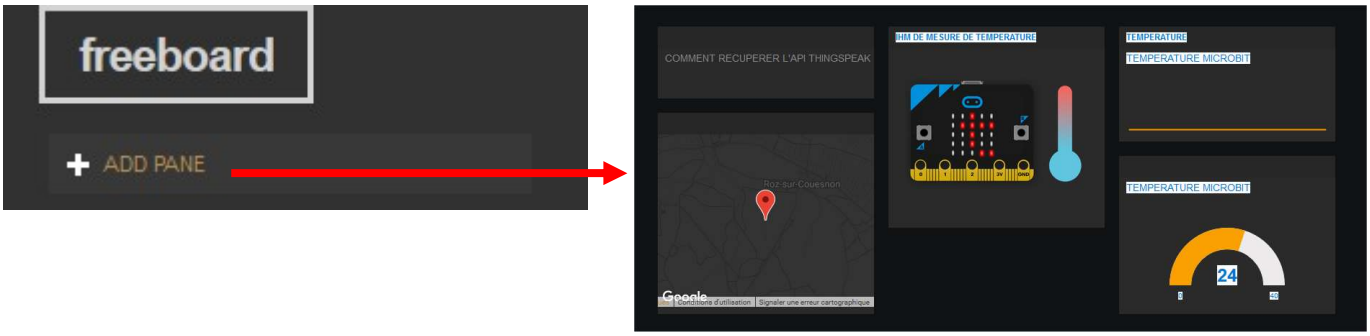


D - Bien paramétrer les DATA SOURCES

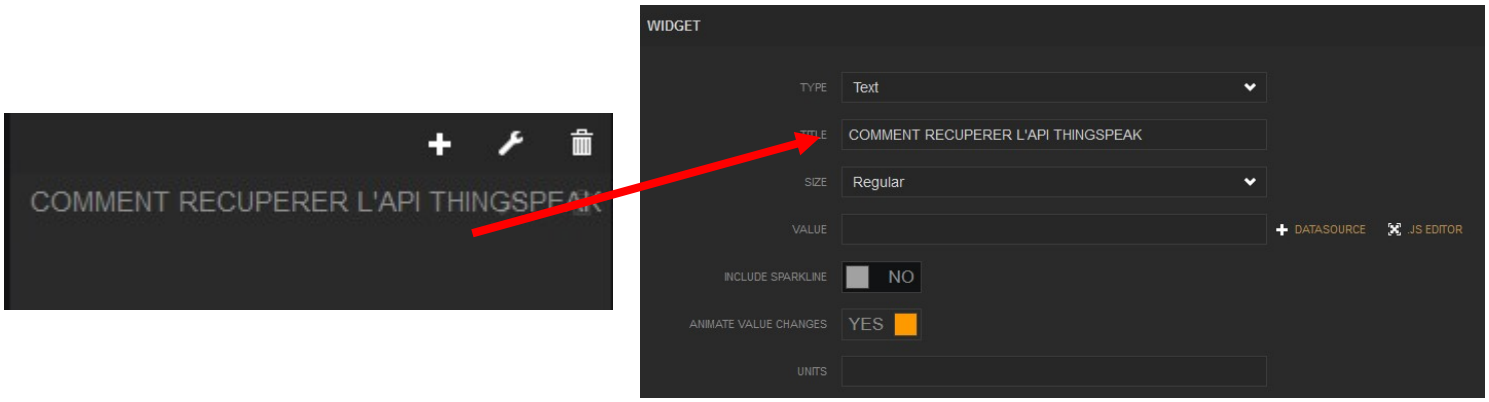


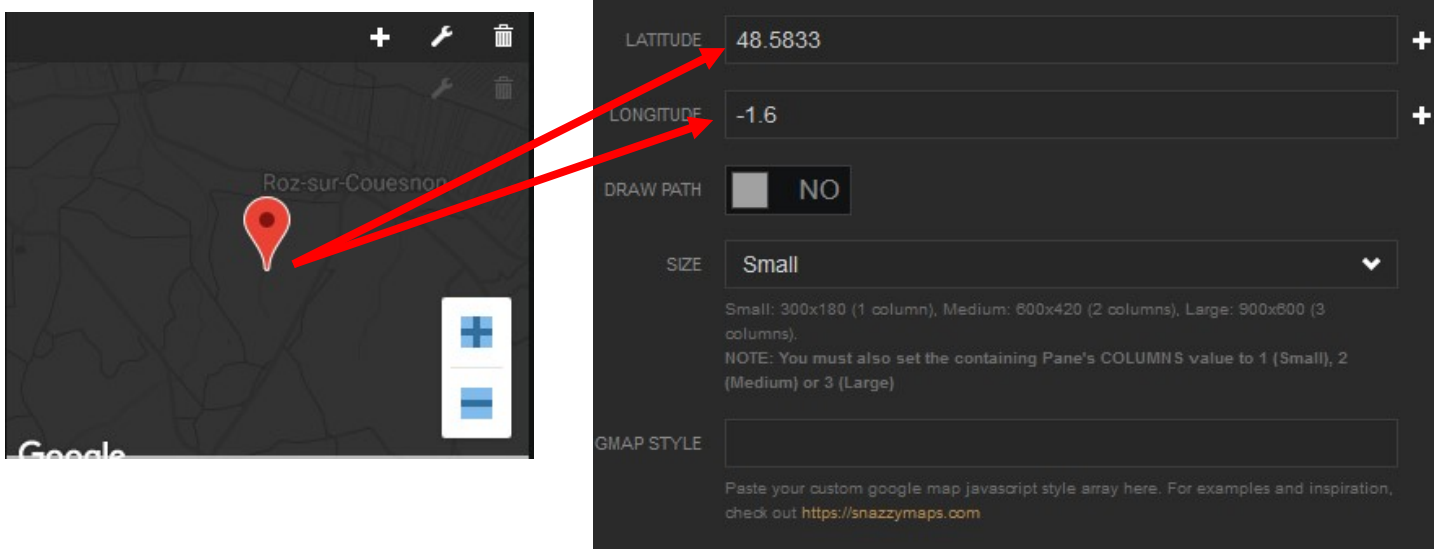


E - Ajouter les panels



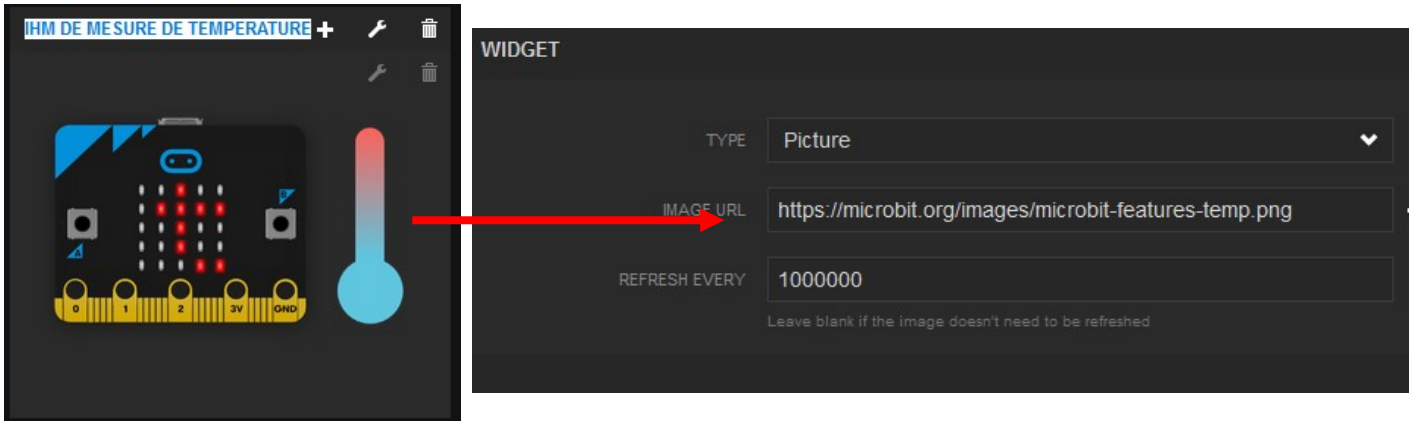
https://freeboard.io/board/XIvdeZ

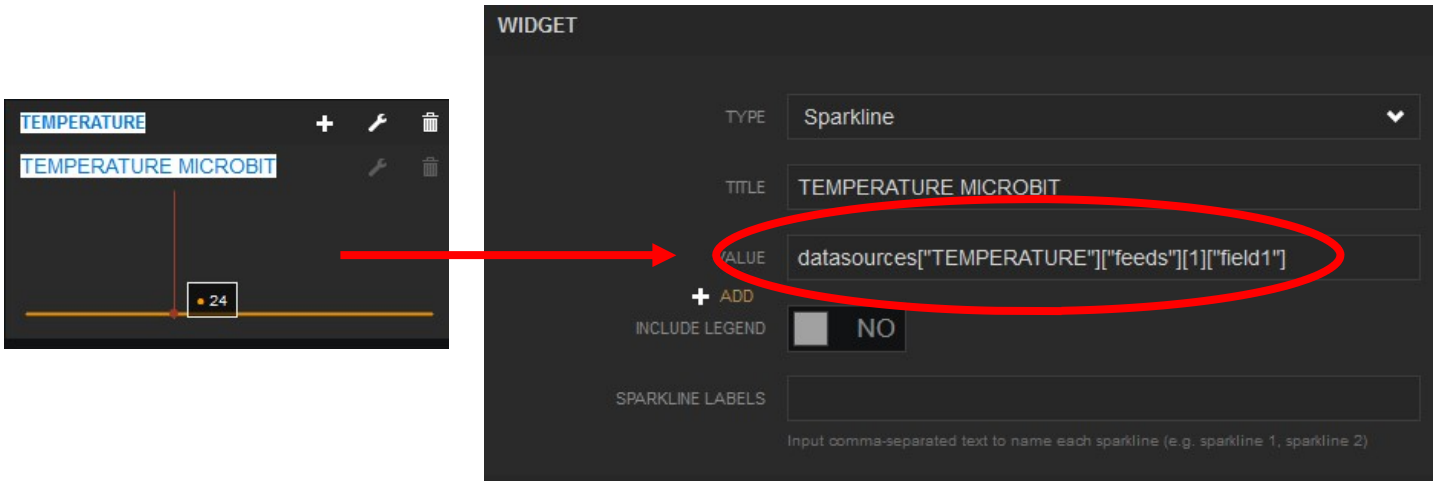


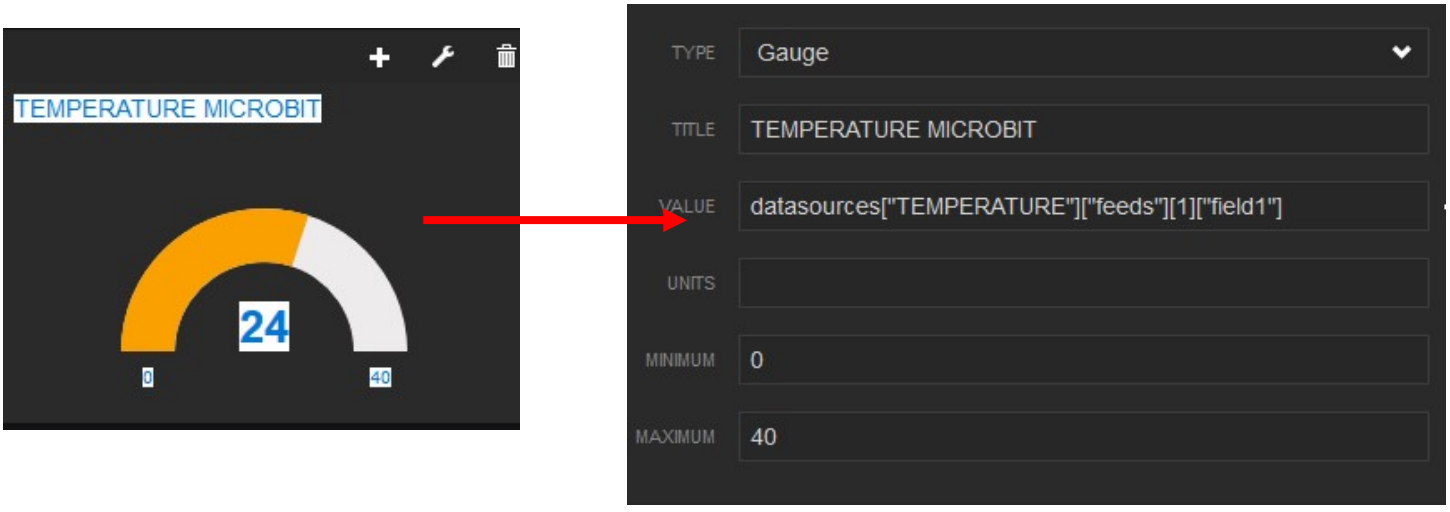


https://freeboard.io/board/XIvdeZ

E - Ajouter les panels







https://freeboard.io/board/XIvdeZ