

Mise en pratique : démarche projet

Contenus	Capacités attendues
Systèmes informatiques embarqués	Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs, l'IHM et les actions des actionneurs dans des systèmes courants.
Interface homme-machine (IHM)	Réaliser une IHM simple d'un objet connecté.
Commande d'un actionneur, acquisition des données d'un capteur	Écrire des programmes simples d'acquisition de données ou de commande d'un actionneur.



Mise en pratique : démarche projet

Réaliser un compteur de personnes pour un spectacle

Dans cette période de COVID la jauge de la salle est de 120 personnes afin de permettre la distanciation entre les sièges



Mise en œuvre matérielle

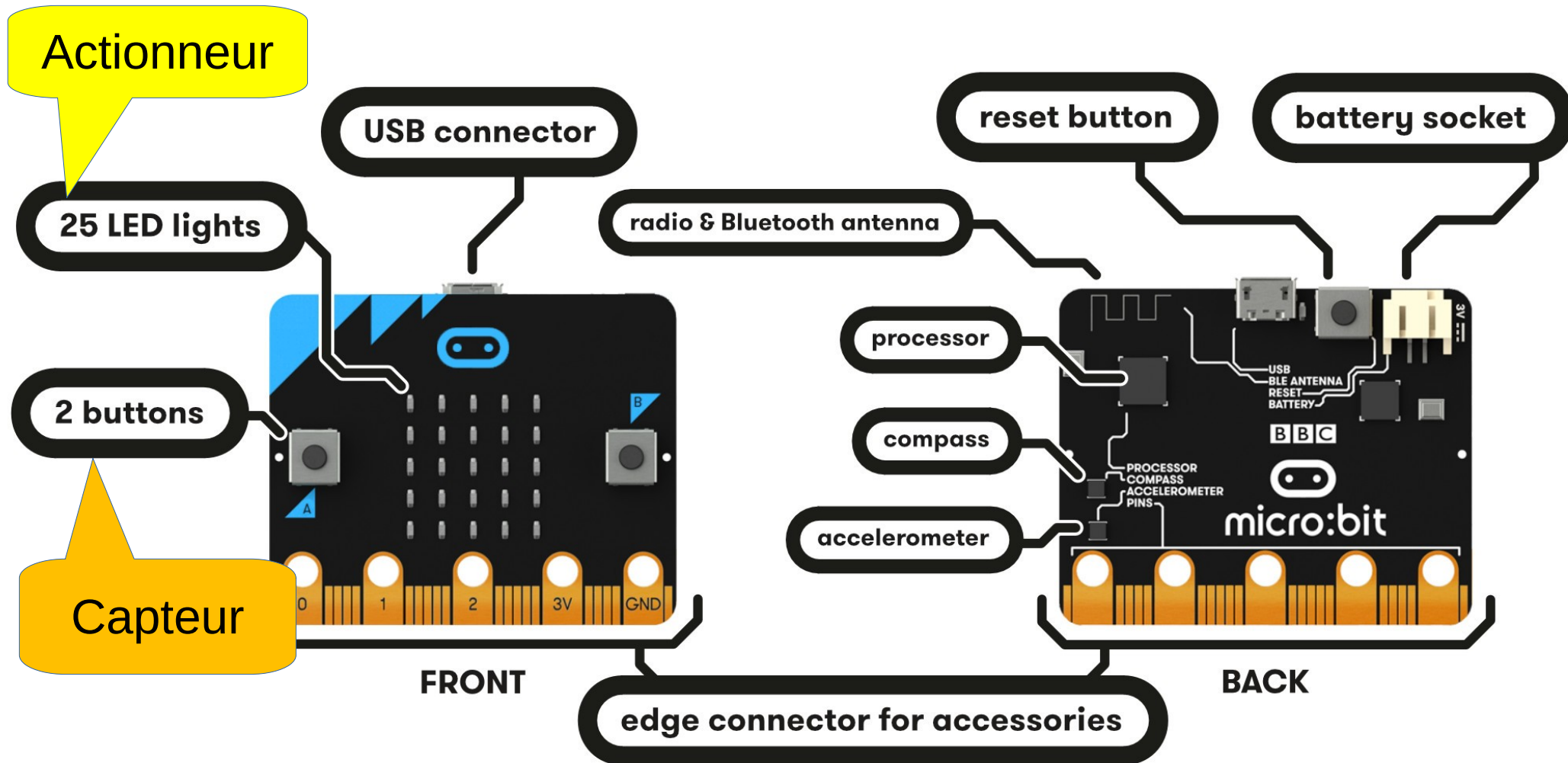
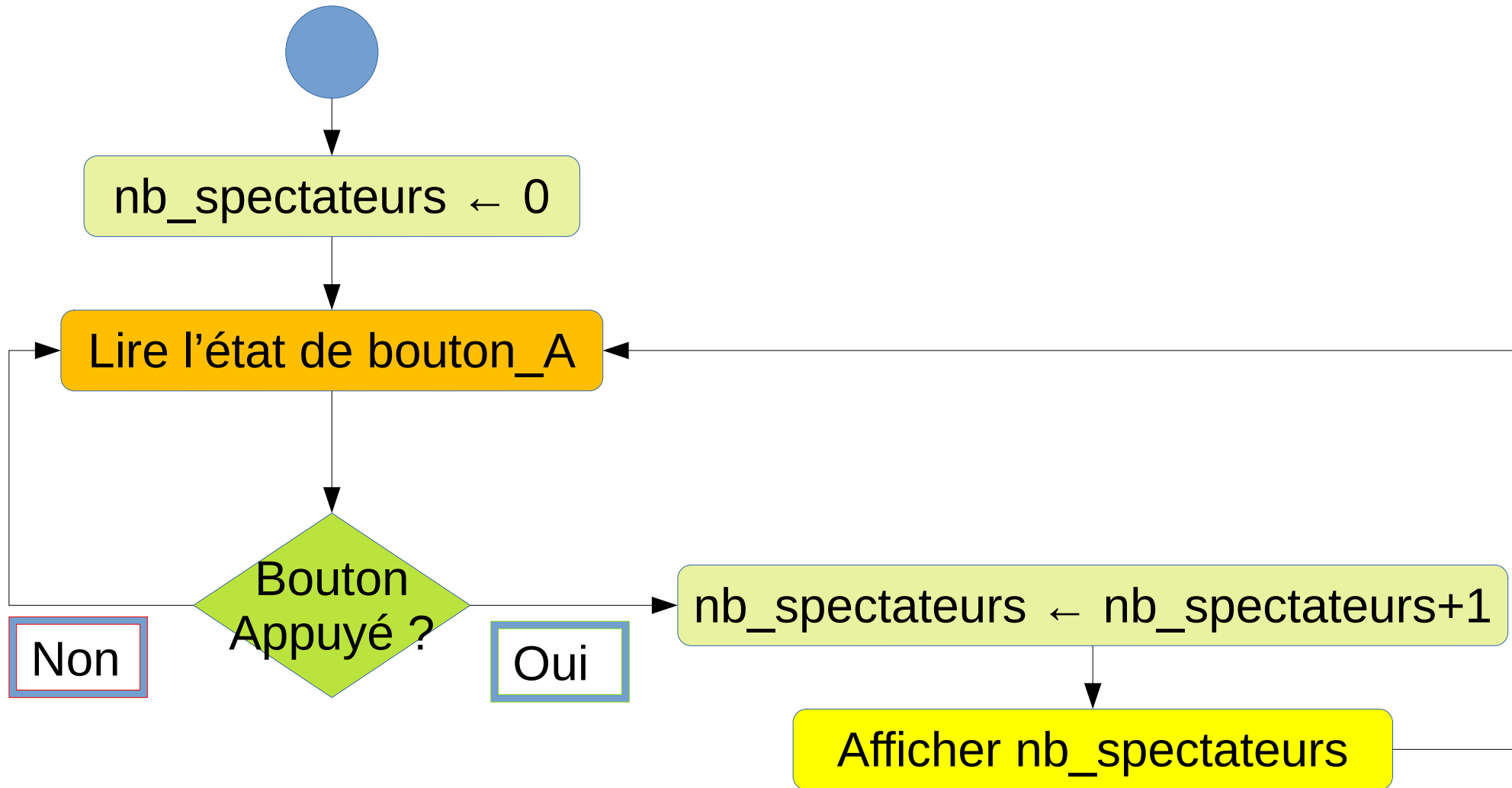


Diagramme fonctionnel



Algorithme

$\text{nb_spectateurs} \leftarrow 0$

Répéter indéfiniment :

Si bouton_A a été pressé :

$\text{nb_spectateurs} \leftarrow \text{nb_spectateurs} + 1$

afficher nb_spectateurs

Fin Si

Fin répéter

Programme Python EXERCICE 1



x compteur1.py

```
1 from microbit import *
2
3 nb_spectateurs = 0
4
5 # Répéter indéfiniment
6 while True:
7     if button_a.was_pressed():
8         nb_spectateurs = nb_spectateurs+1
9         display.show(nb_spectateurs)
```

Programme Python EXERCICE 1

```
from microbit import *
```

```
nb_spectateurs = 0
```

```
# Répéter indéfiniment
```

```
while True:
```

```
    if button_a.was_pressed():
```

```
        nb_spectateurs = nb_spectateurs+1
```

```
        display.show(nb_spectateurs)
```

Simulation EXERCICE 1

Simulateur en ligne : <https://create.withcode.uk>

The screenshot shows the online micro:bit simulator interface. On the left, a code editor displays a Python script for counting button presses. The code is as follows:

```
1 from microbit import *
2
3 nb_spectateurs = 0
4
5 # Répéter indéfiniment
6 while True:
7     if button_a.was_pressed():
8         nb_spectateurs = nb_spectateurs+1
9         display.show(str(nb_spectateurs))
```

On the right, the simulator window titled "mycode.py" shows a virtual micro:bit board. Above the board is a toolbar with buttons for "Micro:bit tools", "Help and tips", "Accelerometer", "Compass", "I/O Pins", "Thermometer", and "Radio". Below these are buttons for "Download HEX" and "Press A + B". The micro:bit board itself has a 5x5 LED matrix, two buttons labeled "A" and "B", and a pin header at the bottom with pins labeled 0, 1, 2, 3V, and GND. At the bottom of the simulator window, a disclaimer reads: "This micro:bit simulator is an unofficial alpha test version. Create.withcode.uk is not affiliated".

Téléversement sur la carte



Bravo! Aucun problème détecté.



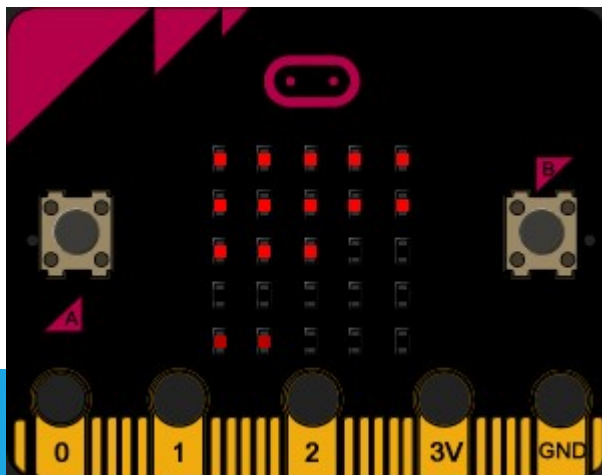
Démarche pédagogique

- Montrer le système que l'on souhaite au final
- Identifier les actionneurs et capteurs en jeu
- Analyser le fonctionnement (diagramme)
- Concevoir un algorithme
- Programmer cet algorithme en Python
- Déployer le programme sur la carte
- Tester le fonctionnement
- Imaginer des évolutions possibles

Évolutions possibles

Version monocarte

- V1 : Le bouton B réinitialise le compteur
- V2 : Le compteur s'arrête à 120 affichant une image
- V3 Pour les plus aguerris : Le compteur s'affiche sur la matrice LED sous forme de points : les 4 premières lignes permettent de compter jusqu'à 20 et la dernière ligne compte le nombre de pages de 20 complétées. On a ainsi un affichage condensé jusqu'à 120 sur un seul écran.



Valeur affichée : $13 + 2 \times 20 = 53$

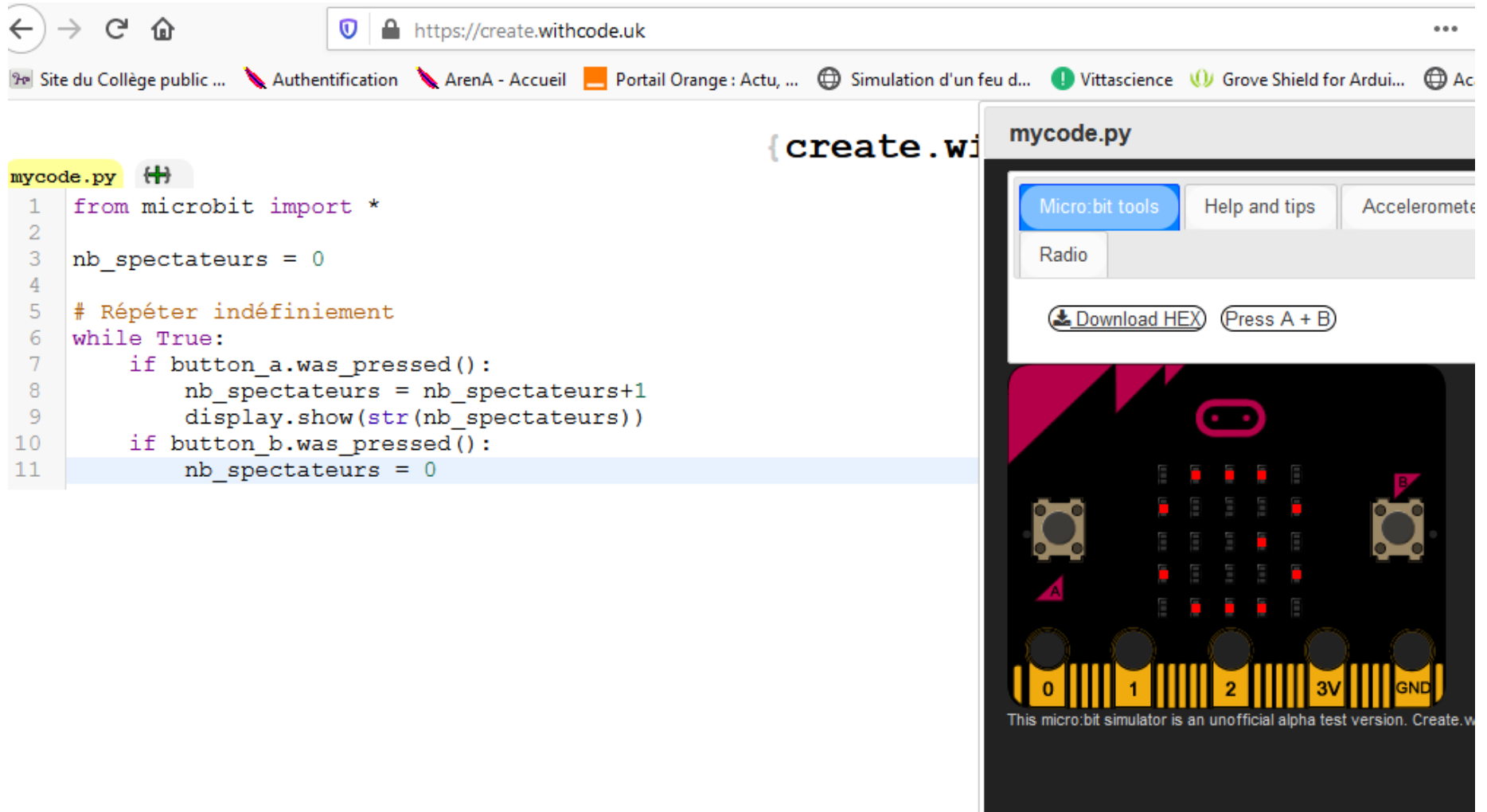
Programme Python EXERCICE 2

- Le bouton B remet le compteur à 0

```
compteur - EXERCICE 2.py X
1  from microbit import *
2
3  nb_spectateurs = 0
4
5  # Répéter indéfiniment
6  while True:
7      if button_a.was_pressed():
8          nb_spectateurs = nb_spectateurs+1
9          display.show(nb_spectateurs)
10     if button_b.was_pressed():
11         nb_spectateurs = 0
12
```

Evolutions possibles – EXERCICE 2

- Le bouton B remet le compteur à 0



The screenshot displays the 'create.withcode.uk' web interface. On the left, a code editor shows the following Python code for 'mycode.py':

```
1 from microbit import *
2
3 nb_spectateurs = 0
4
5 # Répéter indéfiniment
6 while True:
7     if button_a.was_pressed():
8         nb_spectateurs = nb_spectateurs+1
9         display.show(str(nb_spectateurs))
10    if button_b.was_pressed():
11        nb_spectateurs = 0
```

On the right, the simulator interface for 'mycode.py' is shown. It includes a 'Micro:bit tools' tab, a 'Radio' section, and a 'Download HEX' button. Below these, there is a 'Press A + B' button. The main area displays a virtual Micro:bit board with a grid of LEDs and two buttons labeled 'A' and 'B'. At the bottom, there is a row of pins labeled 0, 1, 2, 3V, and GND. A disclaimer at the bottom states: 'This micro:bit simulator is an unofficial alpha test version. Create.w'.

Programme Python EXERCICE 3

- La jauge de la salle est de 120 – il faut afficher une image si la salle est pleine – IMAGE SAD par ex

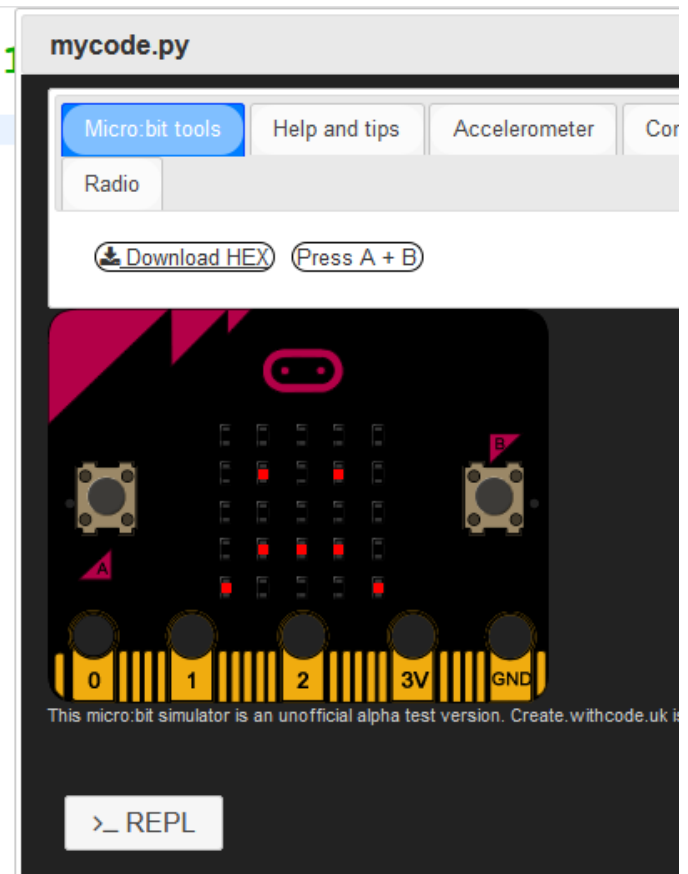
```
compteur - EXERCICE 3.py X
1  from microbit import *
2
3  nb_spectateurs = 0
4
5  # Répéter indéfiniment
6  while nb_spectateurs < 120:
7      if button_a.was_pressed():
8          nb_spectateurs = nb_spectateurs + 1
9          display.show(nb_spectateurs)
10     if button_b.was_pressed():
11         nb_spectateurs = 0
12 display.show(Image.SAD)
```

Programme Python EXERCICE 3

- La jauge de la salle est de 120 – il faut afficher une image si la salle est pleine – IMAGE SAD par ex

```
mycode.py {+}
1 from microbit import *
2
3 nb_spectateurs = 0
4
5 # Répéter indéfiniment
6 while nb_spectateurs < 20:
7     if button_a.was_pressed():
8         nb_spectateurs = nb_spectateurs + 1
9         display.show(str(nb_spectateurs))
10    if button_b.was_pressed():
11        nb_spectateurs = 0
12    display.show(Image.SAD)
```

{7437A2F4}



Evolutions possibles – EXERCICE 4

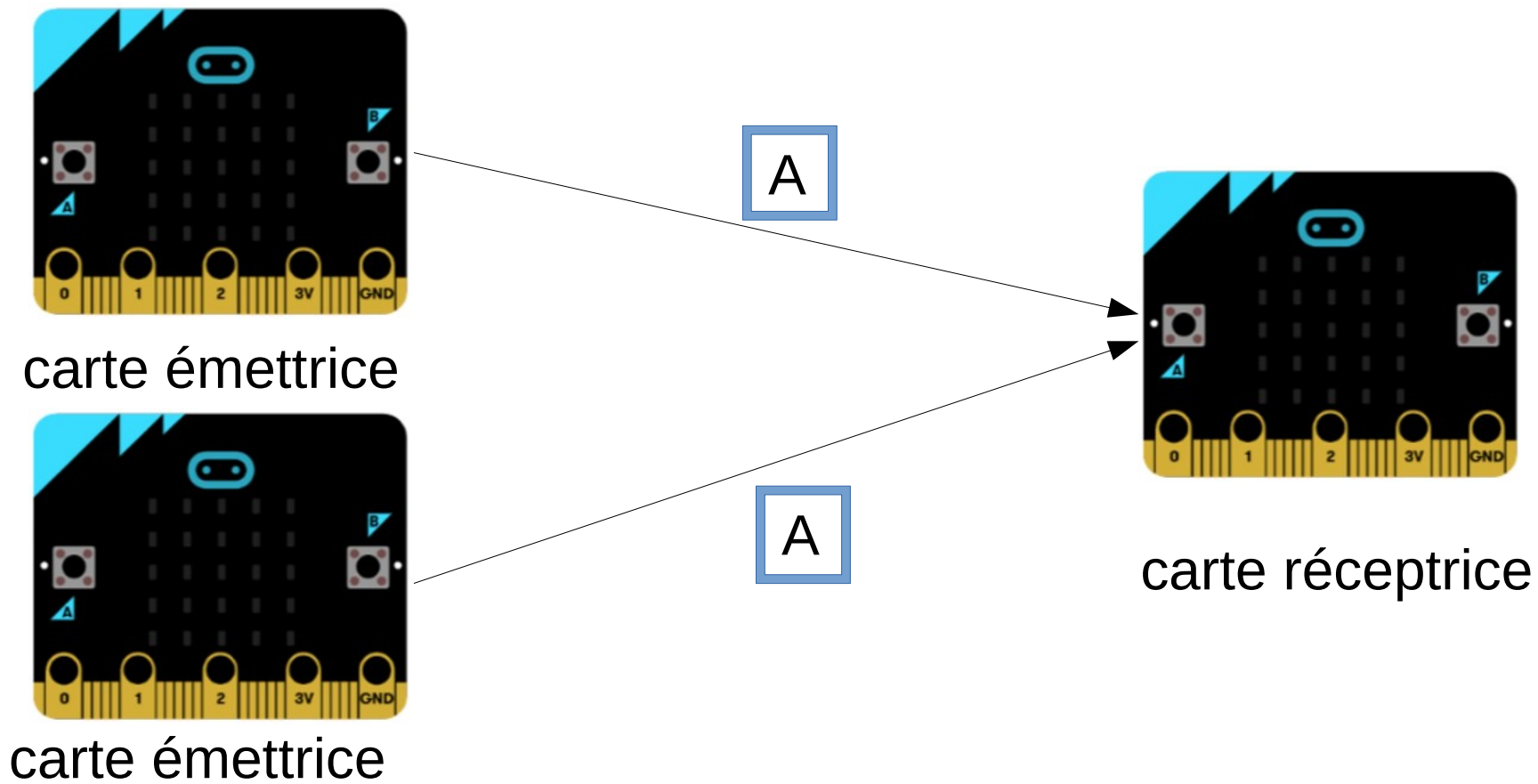
- Le bouton B remet le compteur à 0
- Cas de plusieurs entrées à la salle de spectacle
 - Plusieurs cartes pour compter les personnes
 - Une carte centralise le compte total
 - Nécessité de connecter nos objets via radio

Evolutions possibles EXE 4

- Définir un protocole de communication
- Concevoir le programme sur la carte émettrice
- Concevoir le programme sur la carte réceptrice
- Tester le système

Protocole de communication EXE 4

- Règles communes de communication entre le ou les cartes émettrices et la carte réceptrice



Algorithmes – EXERCICE 4

Carte Récepteur

Configurer la radio

$\text{nb_spectateurs} \leftarrow 0$

Répéter indéfiniment :

Si on reçoit le message 'A' :

$\text{nb_spectateurs} \leftarrow \text{nb_spectateurs} + 1$

Si bouton_A pressé :

Afficher nb_spectateurs

Cartes Emetteur

Configurer la radio

Répéter indéfiniment :

Si bouton_A pressé :

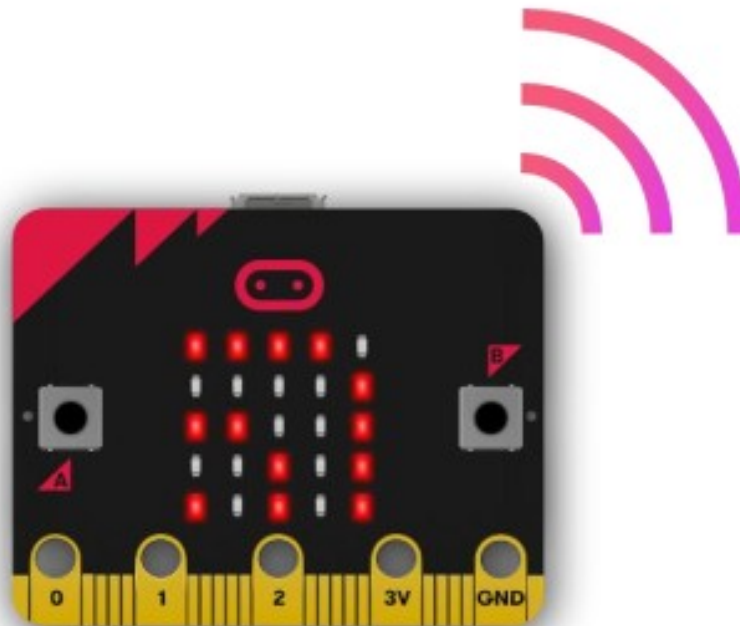
Envoyer le message 'A'

RAPPELS FONCTION RADIO EX4

https://lecluseo.scenari-community.org/CircuitPython/co/g_radio.html

La Radio

- **Qu'est-ce que c'est ?** La Radio te permet de communiquer sans fil entre plusieurs micro:bits. Utilise la radio pour envoyer des messages aux autres micro:bits



RAPPELS FONCTION RADIO EX4

https://lecluseo.scenari-community.org/CircuitPython/co/g_radio.html

Radio

- Pour accéder à ce module, vous devez:

```
import radio
```

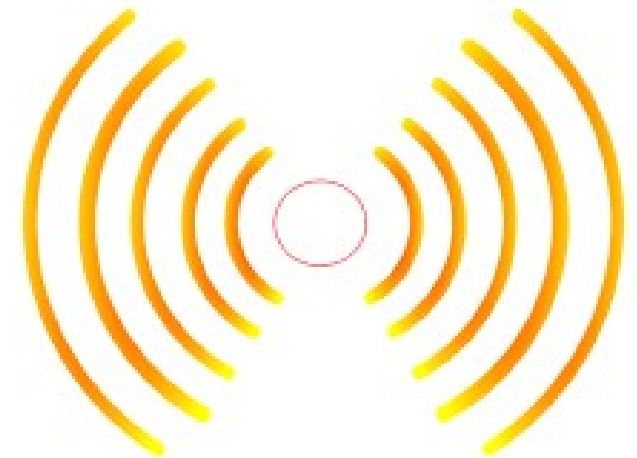
- Fonction:

radio.on() : Allume la **radio**

radio.off() : Etteint la radio

radio.send(message) : envoyer un message

radio.receive() : recevoir un message



Programme Émetteur EX4

```
1 from microbit import *
2 import radio
3 # Configuration radio
4 radio.config(group=1)
5 radio.on()
6
7 while True:
8     if button_a.was_pressed():
9         radio.send("A")
```

Programme Récepteur EX4

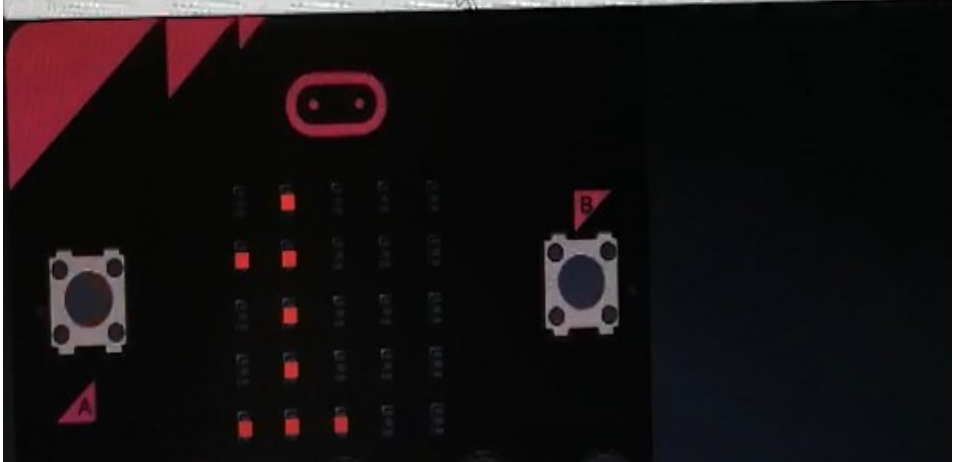
```
1 from microbit import *
2 import radio
3 # Configuration radio
4 radio.config(group=1)
5 radio.on()
6
7 nb_spectateurs = 0
8 while True:
9     if radio.receive() == "A":
10         nb_spectateurs = nb_spectateurs+1
11     if button_a.was_pressed():
12         display.show(nb_spectateurs)
```


SIMULER LE PROGRAMME EX4

<https://create.withcode.uk/>

```
mycode.py   
1 from microbit import *  
2 import radio  
3 # Configuration radio  
4 radio.config(group=1)  
5 radio.on()  
6  
7 nb_spectateurs = 0  
8 while True:  
9     if radio.receive() == "A":  
10         nb_spectateurs = nb_spectateurs+1  
11     if button_a.was_pressed():  
12         display.show(str(nb_spectateurs))
```

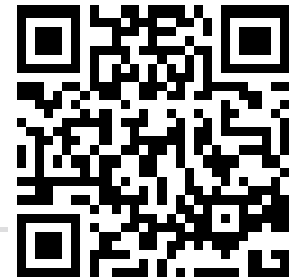
Channel (0-100): Address (32 bit hex):
 Group (0-255):

Tuned in to radio module
Data:
Message:


SIMULER LE PROGRAMME EX4

<https://makecode.microbit.org/#editor>

https://makecode.microbit.org/_W93Mj9KzLiwR



Évolutions possibles

Version Multicartes :

- V1 : En cas de dépassement de 120 spectateurs, une image triste est affichée sur la carte récepteur, mais aussi sur les cartes émetteur
- V2 : Affichage condensé jusqu'à 120 sur l'écran du récepteur, en temps réel sans appuyer sur A

Envoyer SAD - la salle est pleine EX 5

En cas de dépassement de 120 spectateurs, une image triste est affichée sur la carte réceptrice, mais aussi sur les cartes émettrices.

compteur - EXERCICE 5_emit.py

```
1 from microbit import *
2 import radio
3 # Configuration radio
4 radio.config(group=1)
5 radio.on()
6
7 while True:
8     if button_a.was_pressed():
9         .....
10         radio.send("A")
11
12     if radio.receive() == "Z":
13         .....
14         display.show(Image.SAD)
```

cartes émettrices

Envoyer SAD - la salle est pleine EX 5

En cas de dépassement de 120 spectateurs, une image triste est affichée sur la carte récepteur, mais aussi sur les cartes émetteur

```
compteur - EXERCICE 5_emit.py X compteur - EXERCICE 5_recv.py X
1 from microbit import *
2 import radio
3 # Configuration radio
4 radio.config(group=1)
5 radio.on()
6
7 nb_spectateurs = 0
8 while True:
9     if radio.receive() == "A":
10         nb_spectateurs = nb_spectateurs+1
11         if nb_spectateurs > 120:
12             radio.send("Z")
13             display.show(Image.SAD)
14         if button_a.was_pressed():
15             display.show(nb_spectateurs)
16
```

carte réceptrice

QUELQUES LIENS :

Pour aller plus loin :

- Sites de Olivier LECLUSE :
- https://lecluseo.scenari-community.org/CircuitPython/co/module_Micropython.html
- <https://www.lecluse.fr/snt/>
- <http://gcworks.fr/tutoriel/microbit/CarteMicrobit.html>