Mode opératoire communication entre clavier et cartes micro:bit

**Ouvrir** le logiciel Thonny :

Régler l’affichage comme ceci.

**Copier** ce programme (A) sur une nouvelle page :

from microbit import \*

import radio

# Initialise la communication radio

radio.on()

# Dictionnaire de correspondance entre les touches et les images de flèches KEY\_ARROW\_MAPPING = {

"gauche": Image.ARROW\_W,

"droite": Image.ARROW\_E

}

while True: # Attend la réception d'un message radio de la carte connectée

message = radio.receive() # Vérifie si un message a été reçu

if message is not None: # Vérifie si le message correspond à une touche dans le dictionnaire

if message in KEY\_ARROW\_MAPPING: # Affiche l'image de la flèche correspondante sur l'écran

display.show(KEY\_ARROW\_MAPPING[message])

**Copier** ce programme (B) sur une nouvelle page :

from microbit import \*

import radio

# Initialise la communication radio

radio.on()

# Dictionnaire de correspondance entre les touches et les images de flèches

KEY\_ARROW\_MAPPING = {

 b"gauche": Image.ARROW\_W,

 b"droite": Image.ARROW\_E

}

while True:

 # Attend la réception d'un caractère sur le port série

 incoming = uart.read()

 # Vérifie si un caractère a été reçu

 if incoming is not None:

 # Vérifie si le caractère correspond à une touche dans le dictionnaire

 if incoming in KEY\_ARROW\_MAPPING:

 # Affiche l'image de la flèche correspondante sur l'écran

 display.show(KEY\_ARROW\_MAPPING[incoming])

 radio.send(incoming)

**Ouvrir** le menu Outils puis **cliquer** sur gérer les paquets, **chercher** sur Pypi la bibliothèque microbit puis **l’installer.**

**Recommencer** pour radio, serial, keyboard et time

**Brancher** une carte microbit sur un port usb de votre pc.

Lorsque le PC a reconnu la carte, **cliquer** sur Exécuter puis sur configurer l’interpréteur



**Choisir** Micropython (BBC micro :bit) puis OK

Si besoin, **cliquer** sur le bouton STOP, l’arborescence à gauche doit ressembler à cela :





**Cliquer** alors sur Fichier, Enregistrer sous..

Le masque suivant apparaît :



**Choisir** micro :bit puis **enregistrer** le fichier sous le nom main.py sur la carte.

Vous venez d’enregistrer le programme B sur la carte. **Repérer** la couleur de cette carte et **noter** la couleur ainsi que le programme sur une feuille.

**Recommencer** la même opération pour enregistrer l’autre programme (A) sur l’autre carte. **Noter** également la couleur de la carte associée à ce programme.

**Prendre** une nouvelle page et **copier** ce programme (C) :

import serial

import keyboard

import time

# Initialise la communication série avec la carte micro:bit

ser = serial.Serial('COM5', 115200, timeout=1) # Remplacez 'COM3' par le port sur lequel est connectée votre micro:bit

print("Appuyez sur une touche du clavier pour envoyer une lettre à la carte micro:bit")

while True:

 # Attend une saisie clavier

 touche = keyboard.read\_event()

 # Vérifie si une touche a été pressée

 if touche.event\_type == keyboard.KEY\_DOWN:

 # Envoie la touche pressée via le port série

 ser.write(touche.name.encode())

 # Attends un court instant pour éviter d'envoyer plusieurs fois la même lettre

 time.sleep(0.1)

**Configurer** l’interpréteur pour choisir cette fois Python 3 local



**Brancher** la carte avec le programme B sur votre PC, **brancher** la carte avec le programme A sur le PC voisin.

**Rechercher** le port de connexion de la carte du programme B et **modifier** le programme C en conséquence : ser = serial.Serial('COM5', 115200, timeout=1)

**Exécuter** le programme C et **appuye**r sur les flèches de direction gauche et droite. **Vérifier** que les flèches s’affichent de manière cohérente sur les deux cartes.

**Que s’est-il passé ?**

* Le programme C envoie l’appui d’une des touches du clavier sur le port série de la carte micro :bit
	+ ser.write(touche.name.encode())
* La carte avec le programme B écoute son port série :
	+ incoming = uart.read()
* la carte avec le programme B range ce qu’elle reçoit dans incoming
* Si incoming n’est pas vide, la carte avec le programme B affiche la flèche correspondant au dictionnaire défini plus haut :
	+ display.show(KEY\_ARROW\_MAPPING[incoming])
* la carte avec le programme B envoie en radio le même message :
	+ radio.send(incoming)
* la carte avec le programme A qui est juste alimentée, sans lien avec le PC, écoute sa radio
	+ message = radio.receive()
* la carte avec le programme A affiche de la même manière ce qu’elle reçoit :
	+ display.show(KEY\_ARROW\_MAPPING[message])

**Remarque :**

L’usage de plusieurs cartes dans la salle risque de perturber les communications puisque toutes les cartes envoient et écoutent la même fréquence par défaut.

Pour éviter les difficultés, il faut attribuer une fréquence par équipe qui devra être ajoutée aux programmes A et B avant radio.on().

Salle I18 : radio.config(channel=1) à radio.config(channel=5)

Salle I14 : radio.config(channel=6) à radio.config(channel=9)