
SNT - Informatique embarquée et objets connectés-

Partie II-IHM : Interface Homme Machine

Une IHM est un dispositif avec lequel l'humain peut envoyer ou récupérer de l'information avec l'objet connecté.

Notre IHM consistera en une fenêtre permettant l'acquisition de données provenant de la carte micro:bit et permettant également de transmettre des informations à la carte.

1.1 La carte transmet sa température (programme "PRG1")

Le programme suivant appelé **PRG1** affiche la température sur la matrice mais surtout **envoie la valeur de la variable "temp" et donc sa température à l'ordinateur via l'instruction print(temp)**.

Rappel : Utiliser le mode BBC micro:bit pour programmer et flasher la carte.

Entrée []:



```
# PRG1 à flasher dans la micro:bit (cliquer sur "Flasher")
# Programme à flasher avec Le logiciel Mu en mode "BBC micro:bit"

from microbit import *      # Importe la bibliothèque microbit

while True:                # Boucle infinie
    temp=temperature()      # Initialise la variable temp à la valeur de la température
    display.scroll(temp)    # Affiche la température sur le microbit
    print(temp)             # Transmet la valeur de la variable à l'ordinateur
```

Cliquer sur le bouton Flasher du logiciel Mu pour téléverser et exécuter le programme dans la micro:bit

1.2 L'ordinateur reçoit et affiche la température

Pour créer l'IHM, nous allons utiliser la bibliothèque Python **tkinter** qui va nous permettre de **construire une fenêtre pour interagir avec la micro:bit**.

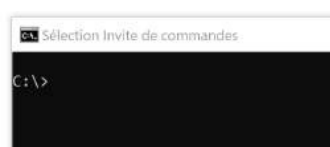
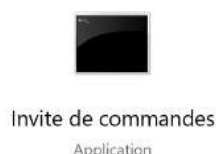
Nous utiliserons également la bibliothèque **serial** pour **gérer les entrées/sorties à travers les ports utilisés par le système**.

1.2.1 Identifier le port série (port COM) dédié à la micro:bit

Pour récupérer la valeur de la température, on va utiliser le port série (port USB de l'ordinateur). Il faut trouver le numéro de port sur lequel la micro:bit est branchée pour pouvoir communiquer.

Sur Windows:

Ouvrir l'invite de commande en tapant **cmd** dans la barre de recherche windows.



Taper **Mode** pour avoir la liste des périphériques branchés à l'ordinateur



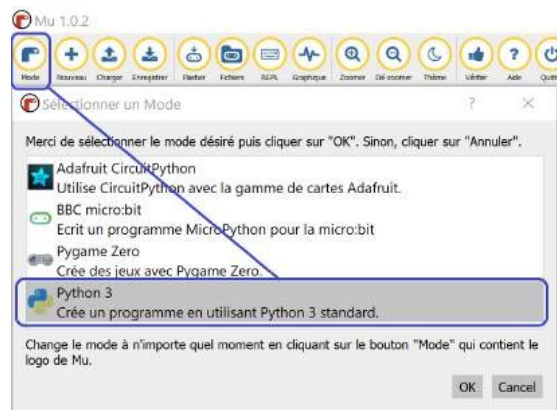
(COM1 et COM 2 sont parfois la souris et le clavier)

Seulement au lycée vous n'avez pas accès à l'invite de commande pour la détection du port COM3 ou COM4 etc... donc **pour déterminer le bon numéro de port** nous lancerons le **PRG2** et quand celui-ci ne **signalera plus d'erreur** de communication le numéro de port sera trouvé.

Ci-dessous exemple de message qui montre que la carte n'est pas branchée sur le port COM6

```
raise SerialException("could not open port {!r}: {!r}".format(self.portstr, ctypes.WinError()))
serial.serialutil.SerialException: could not open port 'COM6': FileNotFoundError(2, 'Le fichier spécifié est introuvable.', None, 2)
```

1.2.1 Mettre le logiciel Mu en mode "Python 3"



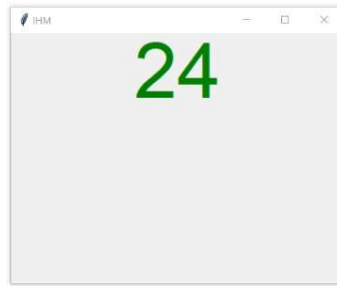
1.2.2 Exécuter le programme créant l'IHM ci-dessous (programme "PRG2")

```
# PRG2
# Programme à exécuter avec Le Logiciel Mu en mode "Python 3" (cliquer sur "Lancer")

import serial # Importe la bibliothèque serial
from tkinter import * # Importe la bibliothèque tkinter
port = "votre port" # "COM6" par exemple pour Windows
ser = serial.Serial(port) # Déclaration de la variable ser qui ouvre le port série
ser.baudrate = 115200 # Vitesse de transmission

data = ser.readline() # On lit sur le port série ce qui est envoyé par la micro:b
ma_fenetre = Tk() # On crée une fenêtre qui sert d'interface avec la carte
ma_fenetre.geometry("400x300") # Fixe la taille de la fenêtre
ma_fenetre.title("IHM") # Donne un titre à la fenêtre
affich_temp = Label(ma_fenetre, text=data, font=("arial", 70), fg="green") # Crée le texte
affich_temp.pack() # Positionne ce texte dans notre fenêtre
ma_fenetre.mainloop() # Pour que la fenêtre reste ouverte
```

Une fenêtre s'ouvre affichant la température lue par la carte.

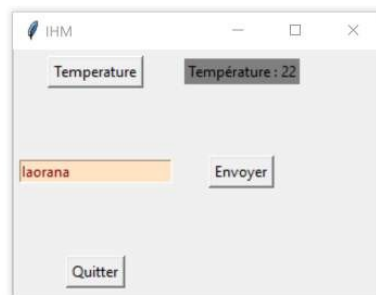


Exercice 1 : Modifier le programme précédent pour changer les paramètres de la fenêtre IHM (modifier les dimensions, le titre, la police de caractères, la taille de la police, la couleur de la police...). Le tester.

Exercice 2: Dans le **mode BBC Micro:bit** ; créer un nouveau programme **PRG1_lum**. Inspirez-vous du programme **PRG1** (côté carte) pour que la fenêtre affiche cette fois **le niveau de luminosité** (de 0 à 255) lu par la carte. Il n'est pas nécessaire de modifier le **PRG2** côté ordinateur.

***Rappel :** La commande `display.read_Light_Level()` retourne un entier compris entre 0 et 255 représentant le niveau de lumière.*

2. Réalisation d'une IHM plus complexe



Nous allons ajouter un **bouton pour récupérer la température** fournie par la carte et un **champ de saisie de texte** pour qu'il soit affiché sur la carte. Pour cela, nous allons utiliser les programmes *PRG1* et *PRG2* suivants.

PRG1 à flasher dans la carte micro:bit

```
# PRG1 à flasher dans la micro:bit (cliquer sur "Flasher")
# Programme à flasher avec Le logiciel Mu en mode "BBC micro:bit"

from microbit import *

uart.init(baudrate=115200)                # Vitesse de transfert

display.scroll('Pret')                    # La carte affiche "pret"

while True:
    if uart.any():                        # Si la carte reçoit une transmission du PC
        message = uart.read()            # On stocke la valeur dans la variable "message"
        if message == b'temperature':    # Si la valeur est "temperature",
            temp = temperature()          # La carte envoie la valeur de la température au PC
            print(temp)                   # La carte affiche la température sur sa matrice
            display.scroll(temp)           # Sinon si le message reçu est "pacman"
        elif message == b'pacman':        # La carte affiche l'image de Pacman
            display.show(Image.PACMAN)    # Sinon enfin
        else:
            textedofil = str(message, 'UTF-8') # Transforme le message en chaîne de car
            display.scroll(textedofil, delay=80) # La carte affiche le message reçu sous
        sleep(1000)
```

PRG2 à lancer dans l'ordinateur (Mode Python3)

```
# PRG2
# Programme à exécuter avec Le Logiciel Mu en mode "Python 3" (cliquer sur "Lancer")

import serial, time

from tkinter import *

port = "votre port"          # "COM6" par exemple pour Windows ou "/dev/tty.usbmodem14102" s
baud = 115200
s = serial.Serial(port)
s.baudrate = baud


def demande_temperature():      # Fonction pour demander La température à La carte mic
    s.write(b'temperature')
    data = s.readline()
    data = int(data[0:4])
    print(data)
    texte_temperature.set("Température : "+str(data))


def envoie_message():          # Fonction pour envoyer un message à faire défiler sur
    message_bytes = bytes(texte_message.get(), 'utf-8')
    s.write(message_bytes)


def quitter():                 # Fonction pour fermer La fenêtre d'interface
    s.close()
    ma_fenetre.destroy()


# Fenêtre principale
ma_fenetre = Tk()
ma_fenetre.title("IHM")
ma_fenetre.geometry("300x250")


# Création d'un bouton pour recuperer La temperature
button_temperature = Button(ma_fenetre, text="Temperature", command=demande_temperature)
button_temperature.grid(row=0, column=0, padx=5, pady=5)


# Creation d'un Label pour afficher La temperature
texte_temperature = StringVar()
texte_temperature.set("Température : ")
label_temperature = Label(ma_fenetre, textvariable=texte_temperature , bg="grey")
label_temperature.grid(row=0, column=1, padx=5, pady=5)


# Création d'un bouton pour envoyer un message
button_message = Button(ma_fenetre, text="Envoyer", command=envoie_message)
button_message.grid(row=2, column=1, padx=5, pady=50)


# Création d'un champ de saisie d'un message
texte_message = StringVar()
texte_message.set('Iaorana')
champ_message = Entry(ma_fenetre, textvariable=texte_message, bg="bisque", fg="maroon", width
"25")
champ_message.focus_set()
champ_message.grid(row=2, column=0, padx=5, pady=5)


# Création d'un bouton Quitter
button_quitter = Button(ma_fenetre, text="Quitter", command=quitter)
button_quitter.grid(row=3, column=0, padx=5, pady=5)
ma_fenetre.mainloop()
```

Vérifier le bon fonctionnement des programmes précédents en testant l'interface.

Question: Que se passe-t-il lorsqu'on envoie le texte "pacman" via le champ de saisie de l'interface?

Exercice 3:

Ajouter un bouton "Luminosité" à l'IHM qui récupère l'intensité de la luminosité lue par le capteur de luminosité de la carte. On pourra prendre exemple sur le bouton "Température".

Il faut changer le PROG1 pour mesurer la luminosité de la carte et changer le PROG2 pour placer le bouton « Luminosité » et le champ pour faire apparaître la valeur.

Conseil Pour placer le bouton « Luminosité et le champ utiliser la fonction « row = 1 » dans la fenêtre de tkinter cela veut dire « ligne 1 ».

La **Température** est en « ligne 0 » soit « row = 0 »

Le bouton **Envoyer** en « ligne 2 » soit « row = 2 »

Le bouton **Quitter** en « ligne 3 » soit « row = 3 »

Résultat attendu:

