

# Activité connectée : Interface Homme Machine

## - Contrôle de l'ouverture d'une serre -

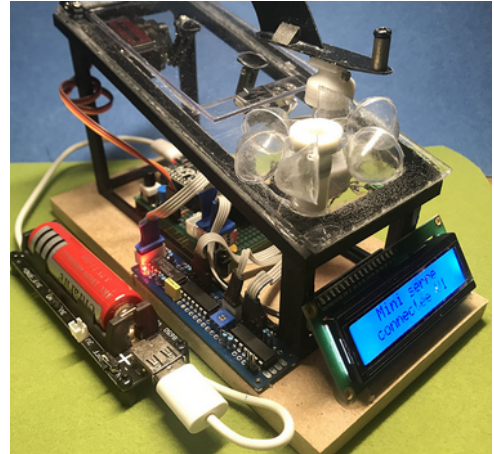


### 1ere Etape : La chaîne de traitement de l'information

#### 1-1 – Les éléments d'une chaîne de traitement de l'information

A partir de la **vue détaillée en annexe**, repérer les différents éléments qui interviennent dans la chaîne de traitement d'une information :

- Source d'alimentation électrique,
- Capteur(s),
- Traitement, transmission des informations
- Actionneur(s)



#### 1-2 Capteurs : mesurer une grandeur ou déterminer l'état d'un système

En utilisant la fiche technique de la maquette et en inspectant celle-ci, déterminez si les capteurs présents sur celle-ci sont :

- des capteurs analogiques, numériques ou booléens,
- des capteurs de contact ou de proximité,

Capteur	Type de mesure	Contact ou proximité
BME280		
Détection OUVERT / FERMÉ		
Girouette et anémomètre		
Luminosité		

Déterminer la fréquence d'échantillonnage du BME280 pour une utilisation typique.

Déterminer les intervalles de mesures possibles pour le BME280.

--	--	--

Le BME280 est-il un capteur adapté à un usage à l'abri dans une serre non chauffée ?

### 1-3 Traitement / transmission de l'information / Interface Homme Machine

Le microcontrôleur chargé de traiter et transmettre l'information pour l'IHM est un ESP32 programmé en microPython. Vous avez ci-dessous un extrait des lignes de code de la bibliothèque python pour la gestion de la serre :

```
from machine import Pin, I2C, PWM, ADC, time_pulse_us
from bme280 import BME280
from lcd_i2c import My_LCD
... / ...

pin_servo = 5 # Broche du servo_moteur
pin_close = 12 # Broche du fin de course fermeture
fdc_close = Pin ( pin_close, Pin.IN ) # GPIO in
pin_open = 4 # Broche du fin de course Ouverture
fdc_open = Pin ( pin_open, Pin.IN ) # GPIO in
pin_LDR = 33 # Broche de connexion vers la LDR
... / ...
ldr = ADC( Pin(pin_LDR) ) # Capteur de lumière
... / ...
# Utilisation du BME280 en mode i2C
bus_i2c = I2C(1, scl=Pin(22), sda=Pin(21), freq=400000)
sensor = BME280 ( i2c = bus_i2c )
... / ...
temp = "{:02.01f}°C".format(sensor.temperature)
press = " / {:.1f}".format(sensor.pressure)
lcd.afficher(0,1,temp+press)
```

Extrait du fichier :  
serre\_biblio.py

Extrait du fichier :  
MiniSerre.py

```
... / ...
if fdc_open.value() == 1 :
    print("Le toit est ouvert.")
if fdc_close.value() == 1 :
    print("Le toit est fermé.")
... / ...
```

Sur quelles broches sont connectés les capteurs et actionneurs ?

Quelle instruction permet de déterminer si le toit est ouvert ou fermé ?

Quelles instructions permettent de mesurer la température et la pression ambiante ?

Le BME280, la LDR et le capteur de fin\_de\_course en fermeture sont accessibles via les instructions : `BME.temperature` / `LDR.read()` / `fdc_close.value()`

A votre avis, pourquoi trois instructions différentes pour obtenir les informations ?



## 2ème étape : Modifier l'interface de contrôle

### 2-1 Se connecter puis afficher l'IHM

Vous pouvez exécuter le fichier de script microPython **MiniSerre.py** et vous connecter à l'ESP32.

Pour cela, connectez-vous au point d'accès réseau qui correspond à votre groupe (exemple pour le groupe 2) :

```
ssid = 'WifiGrp2'
password = 'groupe02'
```

Rentrez alors l'adresse IP de l'ESP32 dans un navigateur (elle est indiquée sur la maquette).

Vous pouvez tester le fonctionnement de l'IHM pour commander l'ouverture et la fermeture du toit de la serre et les informations affichées sur les grandeurs mesurées.

**On souhaite simplifier l'interface de contrôle de l'ouverture de la mini serre avec un nouveau cahier des charges :**

**Objectif n°1 :**

- n'afficher le bouton [ OUVRIR ] que si le toit est fermé,
- n'affiche le bouton [ FERMER ] que si le toit est ouvert.

Pour cela vous pouvez modifier la fonction `web_page(...)` ainsi que les fichiers contenus dans le répertoire `www` : `script.js` et `style.css`

*Il est possible que Thonny n'ouvre pas les fichiers autres que .py. Dans ce cas, il vous faut télécharger les fichiers sur votre disque dur pour les modifier, puis les retransférer vers le dossier `www` de l'ESP32.*

**Pour vous aidez, vous disposez de deux fiches méthodes ...**

**Proposez-nous votre solution avant d'effectuer des modifications ...**

*Note importante : si vous modifiez une bibliothèque en python qui a déjà été chargée en mémoire, il est nécessaire de forcer le rechargement de la bibliothèque en mémoire en réinitialisant le microcontrôleur ( ctrl + D dans la fenêtre de [ console ] )*



**Objectif n°2 :**

Proposer deux modes de fonctionnement :

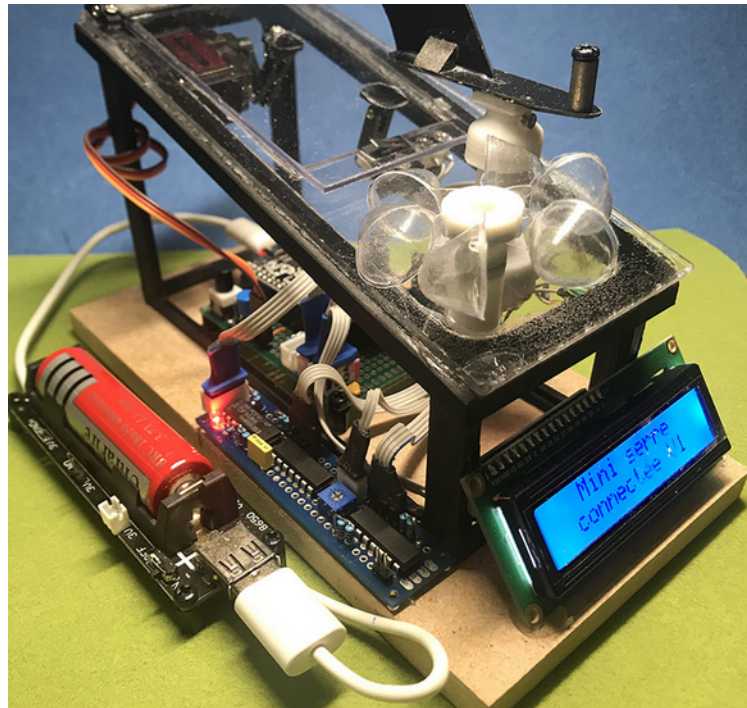
- ouverture / fermeture manuel (Cf Objectif n°1),
- **ouverture / fermeture automatique** avec une température de contrainte déterminée dans le code Python.  
( ouverture automatique au dessus de 30°C par exemple et fermeture automatique en dessous de 25° ).

**Proposez-nous votre solution  
avant d'effectuer des modifications ...**

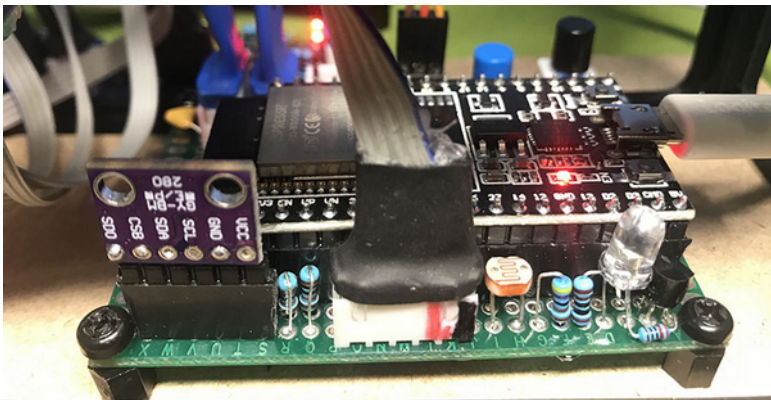
## Annexe 1 :

### Vue détaillée des fonctionnalités de la mini-serre connectée

Vue n°1 : Accumulateur LiPo / Afficheur LCD / Anémomètre / Girouette / Trappe d'aération



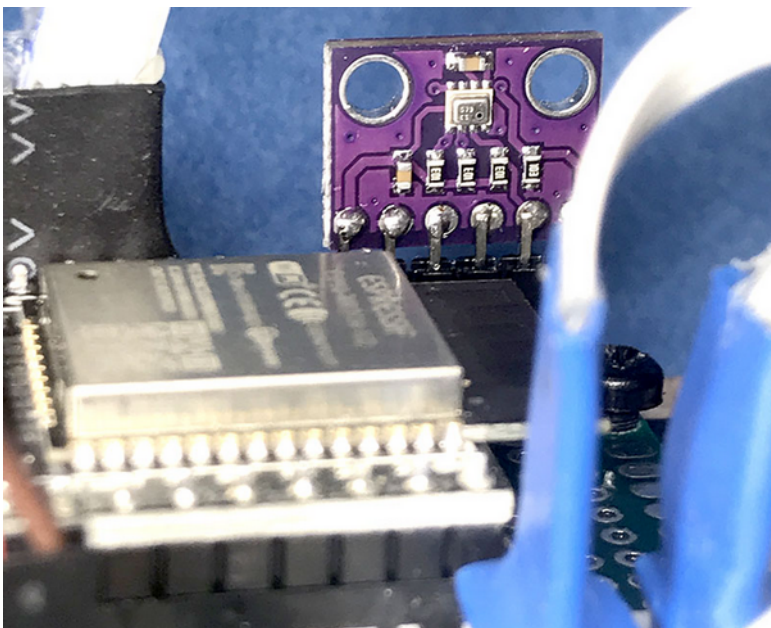
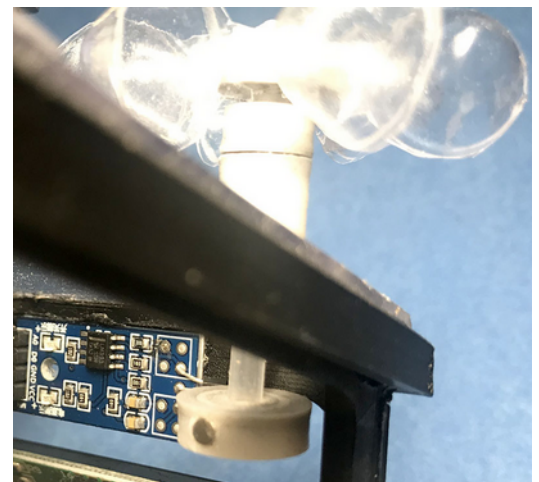
Vue n°2 : BME280 / LDR / LED Blanche / ESP32



Vue n°3 : Anémomètre

Aimant fixé dans un cylindre  
solidaire de l'axe de rotation  
des coupes.

Capteur à effet Hall et  
adaptation du signal  
électrique.



Vue n°4 :  
microcontrôleur ESP32  
BME280



## Annexe 2 : Utilisation de l'IDE Thonny pour communiquer via la connexion USB avec le microcontrôleur ESP32

### 1ère étape : Ressources

Le connecteur USB permet d'établir une connexion avec le microcontrôleur. Il est nécessaire pour cela d'**installer un pilote** :

<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>  
choisir le pilote *Silab CP2102* ...

Nous allons utiliser l'**IDE Python : thonny.org**.

Cet éditeur propose deux interpréteurs Python :

- Python 3.7.2 (*application pour l'ordinateur*)
- **MicroPython** (*Python pour microcontrôleurs*)

→ **Choisir l'interpréteur micropython (ESP32) :**

### Choisir MicroPython pour ESP32

Si tout c'est bien déroulé, vous devriez obtenir l'information suivante :

Vous pouvez afficher les fichiers ressources enregistrés dans le microcontrôleur.

Ce sont ces fichiers que nous allons utiliser pour l'activité. Pour cela activer la fenêtre d'affichage des fichiers :

De même, nous vous proposons d'afficher les numéros de lignes dans l'éditeur pour se repérer plus facilement dans les lignes de code.

### 2ème étape : Se connecter à la mini serre

Ouvrir le programme **MiniSerre.py** et l'**exécuter en appuyant sur F5 ou cliquant sur l'icône (rond – vert)**.

L'ESP32 génère un point d'accès wifi avec un SSID qui correspond à votre numéro de groupe

(voir le numéro indiqué sur la mini serre)

Vous pouvez vous connecter sur le point d'accès avec le mot de passe indiqué.

