



PREMIERE GENERALE

NUMERIQUE ET SCIENCES INFORMATIQUES

MINI-PROJET : STATION METEO



1 – STATIONS METEO

Les petites stations météo sont devenues des objets électroniques incontournables. Elles permettent de mesurer un certain nombre de grandeurs telles que, les températures intérieures et extérieures, le taux d'humidité de l'air, la pression atmosphérique, le taux de CO2 mais également la vitesse et la direction du vent ou encore la quantité de précipitations.



2 – MINI-PROJET

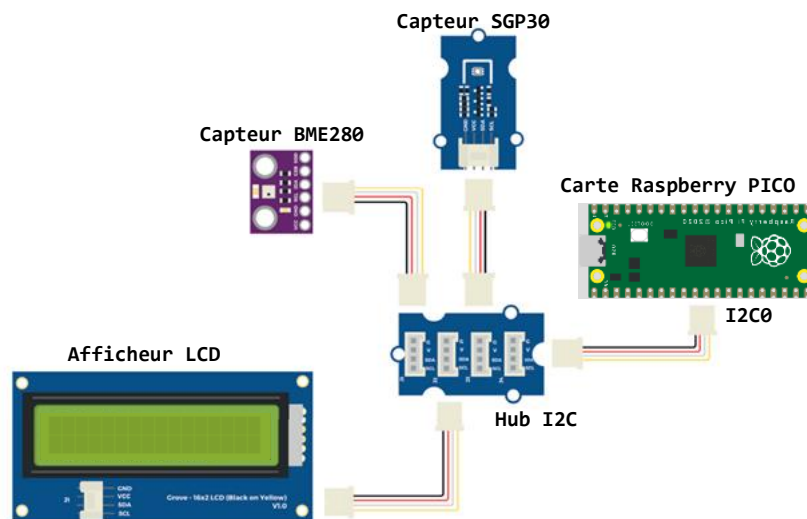
2.1 – Présentation du mini-projet

Mettre en œuvre un système qui permette la mesure et l'affichage des paramètres météorologiques suivants : **température**, **taux d'humidité** de l'air, **pression atmosphérique** et le **taux de CO2** dans l'air.

2.2 – Matériel disponible

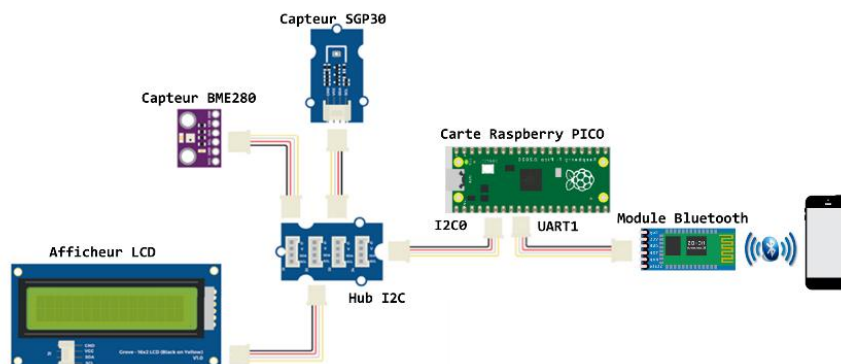
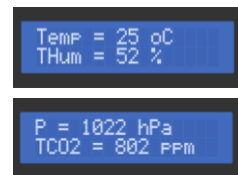
Carte raspberry PICO	
Capteur de température, de pression atmosphérique et d'humidité de l'air BME 280	
Capteur de gaz grove SGP30	
Afficheur LCD I2C Grove	

2.3 – Câblage des composants



2.4 – Cahier des charges

- Mesurer la valeur de la température, du taux d'humidité, de la pression atmosphérique et du taux de CO2.
- Afficher alternativement (en boucle) pendant 5s les valeurs de la température et du taux d'humidité puis les valeurs de la pression atmosphérique et le taux de CO2.
- Pour aller plus loin : Afficher les différentes mesures sur une application mobile. La transmission des données se fera à l'aide du module Bluetooth HC06.



3 – COMPOSANTS

3.1 – Capteur de Gaz SGP30

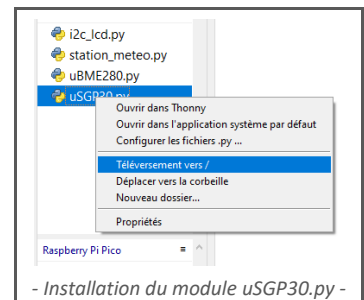
Le capteur de gaz SGP30 grove doit être connecté sur le port I2C de la carte Raspberry PICO. Il peut détecter les gaz suivants :

- Composés organiques volatiles COV : 0 – 60 000 ppb
- Dioxyde de carbone CO2 : 400 – 60 000 ppm



Le module, **uSGP30** permet de récupérer facilement les mesures du capteurs SGP30.

Pour pouvoir utiliser ce module il faut le téléverser dans la carte Raspberry PICO. Pour cela rechercher, dans la zone « Fichiers » le fichier **uSGP30.py** puis avec le bouton droit de la souris cliquer sur « Téléversement vers/ ».



Le capteur de gaz communique avec une liaison I2C. Il est donc nécessaire de créer un objet **I2C** en spécifiant les broches utilisées pour l'interface I2C et éventuellement la fréquence de la liaison. Il faut ensuite créer un objet **SGP30** qui nous permettra de récupérer les mesures.

```
from machine import Pin, I2C
import uSGP30

i2c = I2C(0, scl = Pin(9), sda = Pin(8), freq = 400000)
sgp30 = uSGP30.SGP30(i2c)
```

Les principales méthodes du module **uSGP30** sont :

Méthodes	Rôle	Exemple
SGP30()	Création d'un objet de la classe SGP30 communiquant via une liaison I2C et qui permettra d'accéder aux différentes mesures.	sgp30 = uSGP30.SGP30(i2c)
measure_iaq()	Lecture de la mesure du taux de composés volatiles COV et du taux du dioxyde de carbone CO2	t_CO2, t_COV = measure_iaq()

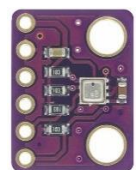
Brochage

Broche	Nom	Câble	Rôle
1	SCL	Jaune	Ligne Horloge liaison I2C
2	SDA	Blanc	Ligne Données liaison I2C
3	VCC	Rouge	Alimentation
4	GND	Noir	Masse (0V°)

3.2 – Capteur météo BME280

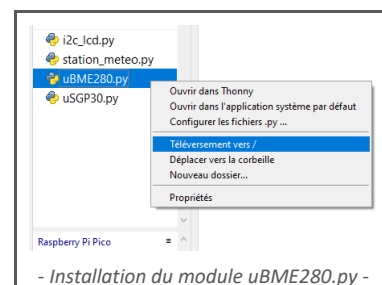
Le capteur météo BME280 peut être connecté sur le bus I2C ou le bus SPI de la carte Raspberry PICO. Il peut mesurer les paramètres suivants :

- La température: -40°C à +85°C à ± 1°C.
- L'humidité relative de l'air : 0 à 100%.
- La pression atmosphérique : 300 à 1100 hPa à ± 1hPa.



Le module, **uBME280** permet de récupérer facilement les mesures du capteur BME280.

Pour pouvoir utiliser ce module il faut le téléverser dans la carte Raspberry PICO. Pour cela rechercher, dans la zone « Fichiers » le fichier **uBME280.py** puis avec le bouton droit de la souris cliquer sur « Téléversement vers/ ».



Le capteur BME280 communique avec une liaison I2C. Il est donc nécessaire de créer un objet **I2C** en spécifiant les broches utilisées pour l'interface I2C et éventuellement la fréquence de la liaison. Il faut ensuite créer un objet **BME280** qui nous permettra de récupérer les mesures.

```
from machine import Pin, I2C
import uBME280

i2c = I2C(0, scl = Pin(9), sda = Pin(8), freq = 400000)
bme280 = uBME280.BME280()
```

Les principales méthodes du module **uBME280** sont :

Méthode	Rôle	Exemple
BME280()	Objet de la classe BME280 qui permettra d'accéder aux mesures.	bme280 = uBME280.BME280()
temperature()	Lecture de la mesure de la température	t = bme280.temperature()
pressure()	Lecture de la mesure de la pression atmosphérique	p = bme280.pressure()
humidity()	Lecture de la mesure du taux d'humidité relative de l'air	h = bme280.humidity()

Brochage



Broche	Nom	Câble	Rôle
1	SCL	Jaune	Ligne Horloge liaison I2C
2	SDA	Blanc	Ligne Données liaison I2C
3	VCC	Rouge	Alimentation
4	GND	Noir	Masse (0V°)

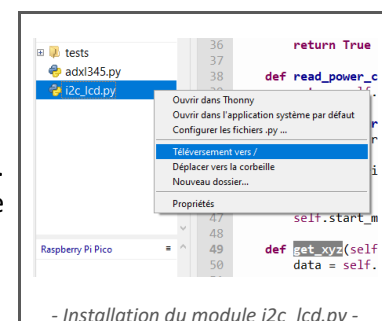
3.3 – Afficheur LCD

L'afficheur LCD doit être connecté sur le bus I2C de la carte Raspberry PICO. Il permet un affichage de 2 lignes de 16 caractères blancs sur un rétroéclairage bleu.



Le module, **i2c_lcd** permet d'utiliser cet afficheur.

Pour pouvoir utiliser ce module il faut le téléverser dans la carte Raspberry PICO. Pour cela rechercher, dans la zone « Fichiers » le fichier **i2c_lcd.py** puis avec le bouton droit de la souris cliquer sur « Téléversement vers/ ».



L'afficheur communique avec une liaison I2C. Il est donc nécessaire de créer un objet **I2C** en spécifiant les broches utilisées pour l'interface I2C et éventuellement la fréquence de la liaison. Il faut ensuite créer un objet **lcd** qui nous permettra de gérer l'afficheur.

```
from machine import Pin, I2C
from i2c_lcd import lcd

i2c = I2C(0, scl = Pin(9), sda = Pin(8), freq = 400000)
display = lcd(i2c)
```

Les principales méthodes de ce module sont :

Methode	Rôle	Exemple
lcd()	Création d'un objet de la classe lcd permettant de piloter l'afficheur	display = lcd()
write_char(c)	Affichage du caractère c	display.write_char('T')
write(chaine)	Affichage la chaine de caractères chaine	display.write('Temp')
cursor(s)	Curseur visible si s = 1 (par défaut) et invisible si s = 0	display.cursor(0)
setCursor(x, y)	Placement du curseur sur la colonne x et la ligne y	display.setCursor(0, 1)
display(s)	Affichage visible si s = 1 (par défaut) et invisible si s = 0	display.display(0)
clear()	Permet d'effacer l'affichage	display.clear()
home()	Placement du curseur sur la colonne 0 et la ligne 0	display.home()

Brochage



Broche	Nom	Câble	Rôle
1	SCL	Jaune	Ligne Horloge liaison I2C
2	SDA	Blanc	Ligne Données liaison I2C
3	VCC	Rouge	Alimentation
4	GND	Noir	Masse (0V°)

3.4 – Module bluetooth HC06

Le module Bluetooth HC06 doit être connecté sur le bus UART de la carte Raspberry PICO. Pour le projet, c'est l'interface UART1 qui sera utilisée.



Les interfaces UART sont gérées par la classe UART de la librairie machine. Il faut créer un objet de type UART, sur lequel seront appliqués les différentes méthodes, en spécifiant le numéro de l'interface, la vitesse de communication (paramètre **baudrate**) et le numéro des broches Rx et Tx.

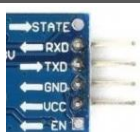
```
from machine import Pin, UART

bluetooth = UART(1, baudrate = 9600, tx = Pin(4), rx = Pin(5))
```

Les principales méthodes de cette classe sont :

Methode	Rôle	Exemple
read()	Lecture de la donnée reçue sur la ligne Rx.	donnee = bluetooth.read()
write(donnee)	Envoi de la chaine de caractères stockée dans donnees sur la ligne Tx.	bluetooth.write('Temp')

Brochage



Broche	Nom	Rôle
1	RXD	Ligne Réception à connecter sur broche Tx Raspberry
2	TXD	Ligne Emission à connecter sur broche Rx Raspberry
3	GND	Masse (0V)
4	VCC	Alimentation