

ENSEIRB-MATMECA



**CONTROLE D'UN OBJET CONNECTE
AVEC
MICROPYTHON**

Patrice KADIONIK
kadionik.enseirb-matmeca.fr

TABLE DES MATIERES

1.	<i>But des travaux pratiques</i>	3
2.	<i>Présentation de l'environnement</i>	4
2.1.	Carte cible RPi-Pico W	4
2.2.	Environnement de développement	6
3.	<i>EX 1 : application Hello World</i>	8
4.	<i>EX 2 : application Hello World par led</i>	8
5.	<i>EX 3 : capteurs de température et de pression</i>	8
6.	<i>EX 4 : Wifi</i>	8
7.	<i>EX 5 : Broker MQTT</i>	8
8.	<i>EX 6 : dashboard Adafruit</i>	9
9.	<i>EX 7 : client MQTT</i>	10
10.	<i>Références</i>	11
11.	<i>Annexe 1 : configuration réseau hôtes et cibles</i>	12
12.	<i>Annexe 2 : configuration des cibles RPi-Pico W</i>	13

1. BUT DES TRAVAUX PRATIQUES

Ces Travaux Pratiques ont pour but de présenter la mise en œuvre d'un *middleware* bas niveau, MicroPython dans la conception d'objets connectés.

L'objet connecté est conçu autour de la carte *Raspberry Pi Pico W* (RPI-Pico W) couplée à la carte d'accueil *Maker Pi Pico*.

Il convient de prendre en main la carte RPI-Pico W pour récupérer dans un premier temps les informations de température et de pression de la carte mère.

Puis, ces informations seront ensuite envoyées vers le *broker* MQTT io.adafruit.com de la société Adafruit.

Enfin, un *dashboard* sera créé pour représenter l'évolution au cours du temps de la température et de la pression de son objet connecté (*time serie*).

Mots clés : objet connecté, *Raspberry Pi Pico W*, *Maker Pi Pico*, environnement MicroPython, langage MicroPython, MQTT, *broker* Adafruit

2. PRESENTATION DE L'ENVIRONNEMENT

2.1. Carte cible RPi-Pico W

La carte RPi-Pico W est une carte permettant de concevoir des objets connectés bon marché et qui est utilisée pour le DIY (*Do It Yourself*) afin de développer de petits systèmes embarqués ou des objets connectés.

La carte RPi-Pico W possède ainsi les éléments suivants :

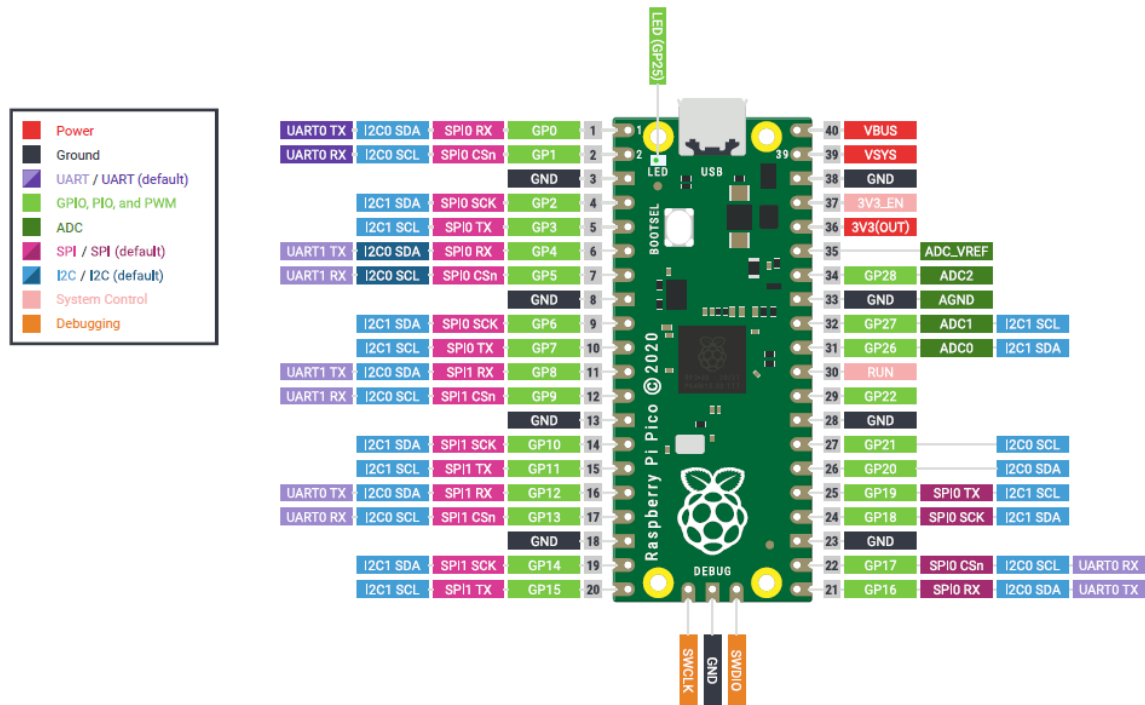
- Processeur ARM Cortex-M0+ *Dual Core* à 133 MHz (SRAM de 264 Ko et Flash de 2 Mo).
- Connectivité Wifi IEEE 802.11b/g/n avec le composant Infineon CYW43439.
- Alimentation 5 V via le connecteur micro USB.
- 26 ports d'entrées/sorties (E/S) comprenant :
 - 3 entrées analogique/numérique (12 bits).
 - 2 ports UART.
 - 2 bus I2C.
 - 16 ports PWM.
- 1 interface SWD de *debug*.
- Horloge RTC intégrée.
- 1 led programmable.
- 1 capteur de température intégré.

L'image suivante présente la carte cible RPi-Pico W :



Carte RPi-Pico W

L'image suivante présente le brochage de la carte cible RPi-Pico W :



Brochage de la carte RPi-Pico W

La carte RPi-Pico W est complétée d'une carte d'accueil, la carte *Maker Pi Pico* pour rajouter différents capteurs et E/S.

La carte *Maker Pi Pico* possède ainsi les éléments suivants :

- Leds sur les E/S GPIO.
- 1 led de couleur connectée sur l'E/S GP28.
- 3 boutons connectés sur les E/S GP20, GP21 et GP22.
- 1 *buzzer* piezo connecté sur l'E/S GP18.
- 1 sortie audio connectée sur les E/S GP18 (L) et GP19 (R).
- 1 *socket* ESP-01.
- 1 *socket* Micro SD sur les E/S GP10 à GP15.

L'image suivante présente la carte *Maker Pi Pico* :



Carte *Maker Pi Pico*

L'ensemble est aussi complété d'un capteur connecté sur le connecteur Grove numéro 0 : capteur BME680.

Le capteur BME680 muni d'une interface I2C propose en fait un ensemble de capteurs :

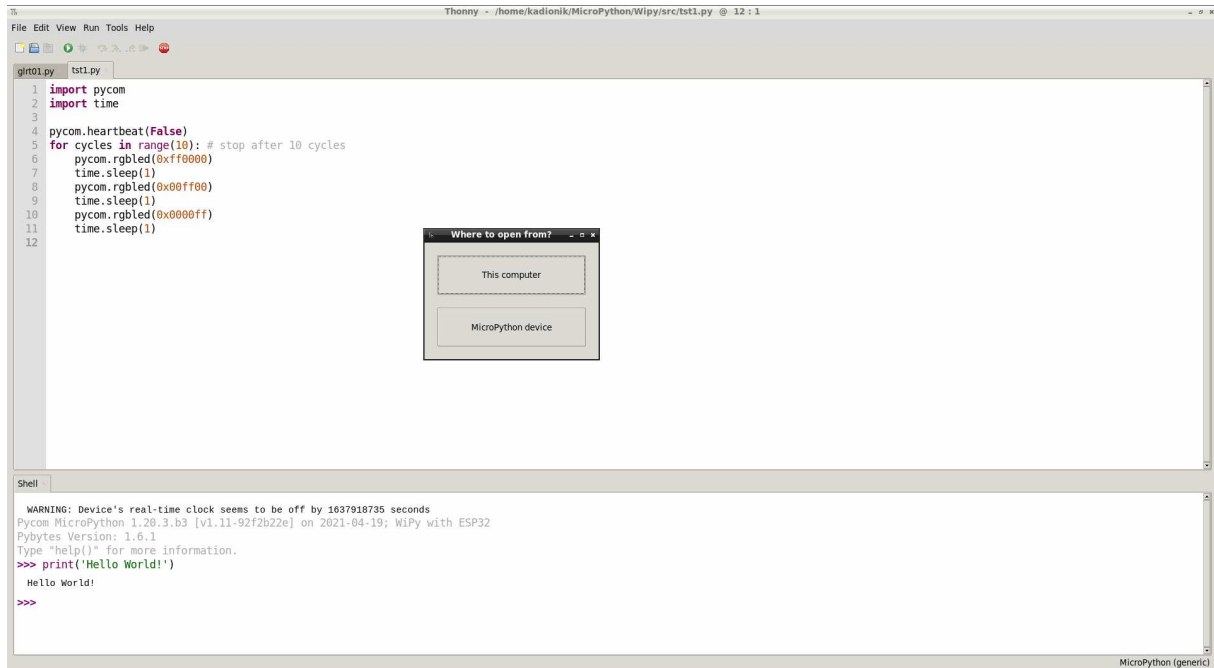
- Température : -40 °C à + 85°C avec une résolution de 0,01 °C.
- Humidité : de 0 à 100 % d'Humidité Relative avec une résolution de 0,008 % HR.
- Pression : de 300 à 1100 hPa avec une résolution de 0,18 Pa.

2.2. Environnement de développement

De nombreuses bibliothèques MicroPython permettent de développer simplement sur la carte RPi-Pico W.

Si l'on clique sur l'icône IDLE Thonny (icône sur le bureau), on a alors accès à un IDE (*Integrated Design Entry*) de développement Python et MicroPython.

Thonny a été configuré pour s'interfacer à la carte RPi-Pico W très simplement et avoir accès à l'interpréteur REPL de MicroPython via la liaison série.



IDE Thonny

L'IDE Thonny est simple et intuitif. La partie supérieure permet d'éditer du code MicroPython sur l'ordinateur. La partie inférieure donne accès à l'interpréteur REPL. L'icône du « triangle vert » permet d'exécuter un programme MicroPython sur la carte cible RPi-Pico W.

Dès lors, nous allons développer des programmes MicroPython avec l'IDE Thonny.

On pourra aussi s'aider des exemples donnés en cours...

3. EX 1 : APPLICATION HELLO WORLD

Nous allons écrire en langage MicroPython la célèbre application « *Hello World!* ».

- Se créer un répertoire de travail à son nom et s'y placer :
host% cd
host% mkdir mon_nom
host% cd mon_nom
- Ecrire en langage Python le programme `tp1.py` d'affichage à l'écran de « *Hello World!* ». Tester.

4. EX 2 : APPLICATION HELLO WORLD PAR LED

- Ecrire en langage MicroPython le programme `tp2.py` qui permet de faire clignoter la led connectée sur l'E/S GP10 toutes les secondes. Tester.

5. EX 3 : CAPTEURS DE TEMPERATURE ET DE PRESSION

- Ecrire le programme `tp3.py` d'affichage à l'écran de la pression et de la température de la carte RPi-Pico W. On affichera les valeurs avec un « chiffre après la virgule » (à 10^{-1} près). Tester.

6. EX 4 : WIFI

- Ecrire le programme `tp4.py` qui permet de se connecter au réseau Wifi « SE » avec le mot de passe donné en TP. Quelle est la configuration Internet de la carte RPi-Pico W. Tester.

7. EX 5 : BROKER MQTT

- Ecrire le programme `tp5.py` qui permet d'envoyer au broker MQTT `io.adafruit.com` la température et la pression de la carte RPi-Pico W. Tester.

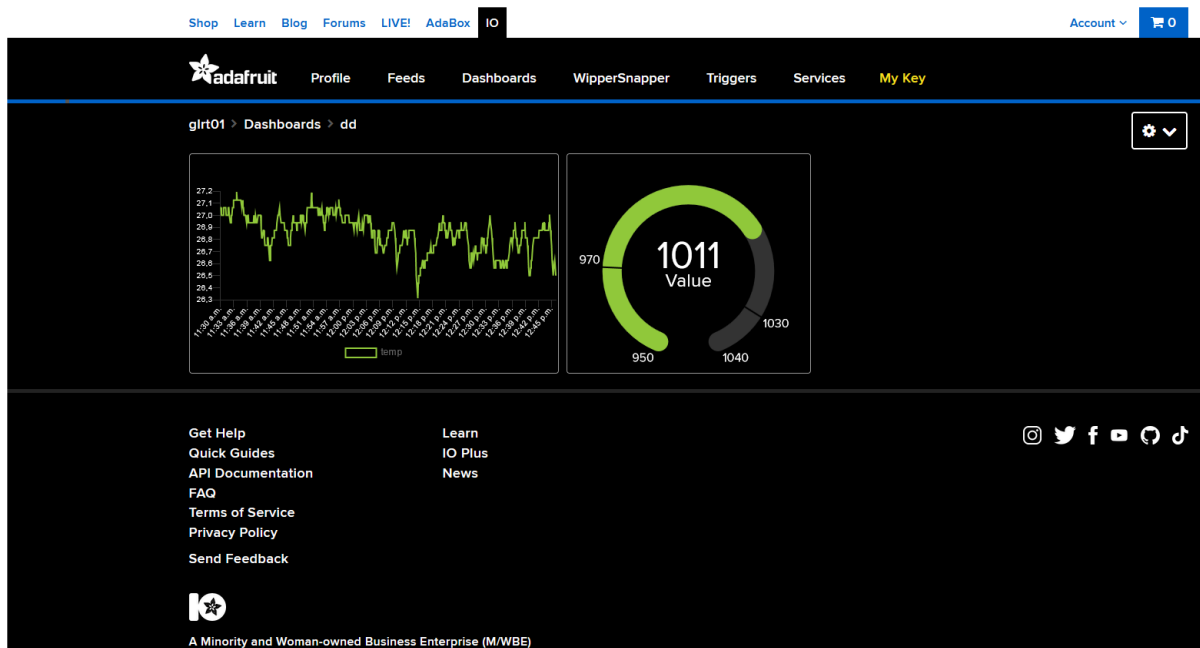
On pourra utiliser les informations données en annexe 4 pour la structuration du nom des 2 *topics* MQTT, pour les noms et mots de passe MQTT pour chaque cible RPi-Pico W.

Pour récupérer le mot de passe MQTT, on se connectera sur son compte avec les identifiants donnés en annexe 4 sur le site `io.adafruit.com` et on pourra le récupérer dans l'onglet « My Key » (champ `IO_KEY`) .

Dans l'onglet « *Feeds* » du site io.adafruit.com, on pourra vérifier la collecte en continu de la température et de la pression.

8. EX 6 : DASHBOARD ADAFRUIT

- A partir de l'onglet « *Dashboards* » du site io.adafruit.com, créer un nouveau dashboard pour représenter graphiquement via des *widgets* la température sous forme d'un graphique et la pression atmosphérique sous forme d'une gauge. Tester.



Dashboard io.adafruit.com

9. EX 7 : CLIENT MQTT

- Le code Python d'un client MQTT avec la bibliothèque Paho et le suivant :

```
import paho.mqtt.client as mqtt

MQTT_SERVER = "???"      # MQTT server address
MQTT_TOPIC = "???"      # Topic name

def on_connect(client, userdata, flags, rc):
    print("Connection : " + str(rc))
    # Subscribe to the topic
    client.subscribe(MQTT_TOPIC)

# A publish message is received from the server
def on_message(client, userdata, msg):
    print("Sujet : " + msg.topic + " Message : " + str(msg.payload))

client = mqtt.Client()

client.username_pw_set(username="???", password="??")

client.on_connect = on_connect
client.on_message = on_message
client.connect(MQTT_SERVER, 1883, 60)

client.loop_forever()
```

- Modifier le code source pour récupérer par MQTT les informations de pression et de température de sa carte RPi-Pico W.
- Lancer le script Python.
- Observer et analyser les traces MQTT capturées.

10. REFERENCES

- Carte *Raspberry Pi Pico W* : <https://www.lextronic.fr/carte-raspberry-pi-pico-w-avec-connecteurs-soudes-64870.html>
- Carte *Maker Pi Pico* : <https://www.lextronic.fr/maker-pi-pico-base-pour-raspberry-pi-pico-63666.html>
- Documentation *Maker Pi Pico. Simplifying Raspberry Pi Pico for Beginner* : <https://docs.google.com/document/d/1JoHsZk5IipQPCLXWbZYPDKjGlnkyAC0J1taUrKVSRg8>
- Documentation MicroPython : <http://docs.micropython.org/en/latest/index.html>
- Documentation MicroPython sur la carte *Raspberry Pi Pico* : <https://docs.micropython.org/en/latest/rp2/quickref.html>
- Apprendre Python : <http://apprendre-python.com/>

11. ANNEXE 1 : CONFIGURATION RESEAU HOTES ET CIBLES

POSTE PC01		
	NOM	ADRESSE IP
HOTE	rpi001	10.7.2.118
CIBLE	pico01	DNS Wifi 192.168.5.x

POSTE PC02		
	NOM	ADRESSE IP
HOTE	rpi002	10.7.2.119
CIBLE	pico02	DNS Wifi 192.168.5.x

POSTE PC03		
	NOM	ADRESSE IP
HOTE	rpi003	10.7.2.120
CIBLE	pico03	DNS Wifi 192.168.5.x

POSTE PC04		
	NOM	ADRESSE IP
HOTE	rpi004	10.7.2.121
CIBLE	pico04	DNS Wifi 192.168.5.x

POSTE PC05		
	NOM	ADRESSE IP
HOTE	rpi005	10.7.2.122
CIBLE	pico05	DNS Wifi 192.168.5.x

POSTE PC06		
	NOM	ADRESSE IP
HOTE	rpi006	10.7.2.123
CIBLE	pico06	DNS Wifi 192.168.5.x

12. ANNEXE 2 : CONFIGURATION DES CIBLES RPI-PICO W

POSTE PC01	
Nom	pico01
Nom compte io.adafruit.com	se01
Passwd compte io.adafruit.com	voir en TP
Username MQTT	se01
Passwd MQTT	voir onglet « My Key » sur io.adafruit.com
Topic MQTT température	se01/feeds/temp
Topic MQTT pression	se01/feeds/pression

POSTE PC02	
Nom	pico02
Nom compte io.adafruit.com	se02
Passwd compte io.adafruit.com	voir en TP
Username MQTT	se02
Passwd MQTT	voir onglet « My Key » sur io.adafruit.com
Topic MQTT température	se02/feeds/temp
Topic MQTT pression	se02/feeds/pression

POSTE PC03	
Nom	pico03
Nom compte io.adafruit.com	se03
Passwd compte io.adafruit.com	voir en TP
Username MQTT	se03
Passwd MQTT	voir onglet « My Key » sur io.adafruit.com
Topic MQTT température	se03/feeds/temp
Topic MQTT pression	se03/feeds/pression

POSTE PC04	
Nom	pico04
Nom compte io.adafruit.com	se04
Passwd compte io.adafruit.com	voir en TP
Username MQTT	se04
Passwd MQTT	voir onglet « My Key » sur io.adafruit.com
Topic MQTT température	se04/feeds/temp
Topic MQTT pression	se04/feeds/pression

POSTE PC05	
Nom	pico05
Nom compte io.adafruit.com	se05
Passwd compte io.adafruit.com	voir en TP
Username MQTT	se05
Passwd MQTT	voir onglet « My Key » sur io.adafruit.com
Topic MQTT température	se05/feeds/temp
Topic MQTT pression	se05/feeds/pression

POSTE PC06	
Nom	pico06
Nom compte io.adafruit.com	se06
Passwd compte io.adafruit.com	voir en TP
Username MQTT	se06
Passwd MQTT	voir onglet « My Key » sur io.adafruit.com
Topic MQTT température	se06/feeds/temp
Topic MQTT pression	se06/feeds/pression