

**ENSEIRB-MATMECA**



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

**Patrice KADIONIK**  
<http://kadionik.vvv.enseirb-matmeca.fr/>

---

## TABLE DES MATIERES

1.	<i>But des travaux pratiques.....</i>	3
2.	<i>Présentation de l'environnement.....</i>	4
2.1.	<i>Carte cible Wipy .....</i>	4
2.2.	<i>Environnement de développement.....</i>	5
3.	<i>EX 1 : application Hello World .....</i>	7
4.	<i>EX 2 : application Hello World par led.....</i>	7
5.	<i>EX 3 : capteurs de température et de pression.....</i>	7
6.	<i>EX 4 : Wifi.....</i>	7
7.	<i>EX 5 : Broker MQTT.....</i>	7
8.	<i>EX 6 : dashboard Adafruit.....</i>	8
9.	<i>Références.....</i>	9
10.	<i>Annexe 1 : résumé du langage Python.....</i>	10
11.	<i>Annexe 2 : mémento du langage Python.....</i>	11
12.	<i>Annexe 3 : configuration réseau hôtes et cibles .....</i>	12
13.	<i>Annexe 4 : configuration des cibles Wipy.....</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 *Wipy* couplée à la carte *Pysense* de la société Pycom.

Il convient de prendre en main la carte *Wipy* pour récupérer dans un premier temps les informations de température et de pression de la carte *Pysense*.

Puis, ces informations seront ensuite envoyées vers le *broker* MQTT [io.adafruit.com](https://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, *Wipy*, *Pysense*, environnement MicroPython, langage MicroPython, MQTT, *broker* Adafruit

## 2. PRESENTATION DE L'ENVIRONNEMENT

### 2.1. Carte cible Wipy

La carte *Wipy* 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 *Wipy* possède ainsi les éléments suivants :

- Un SoC (*System on Chip*) Expressif ESP32 Xtensa avec un processeur double cœur.
- 520 Ko de RAM.
- 8 Mo de Flash.
- Wifi 802.11 b/g/n.
- Bluetooth et Bluetooth Low Energy.
- GPIO, 18 ADC, 2 DAC, 3 UART, 2 I2C, 3 SPI, 1 CAN, PWM.
- 1 slot carte SD.

L'image suivante présente la carte cible *Wipy* :



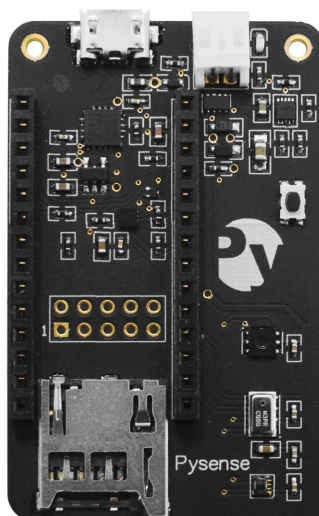
**Carte Wipy**

La carte *Wipy* est complétée d'une carte d'accueil, la *Pysense* pour rajouter différents capteur.

La carte *Pysense* possède ainsi les éléments suivants :

- 1 accéléromètre : LIS2HH12.
- 1 capteur de lumière : LTR329ALS01.
- 1 capteur de pression : MPL3115A2.
- 1 capteur de température / humidité : SI7006A20.

L'image suivante présente la carte *Pysense* :



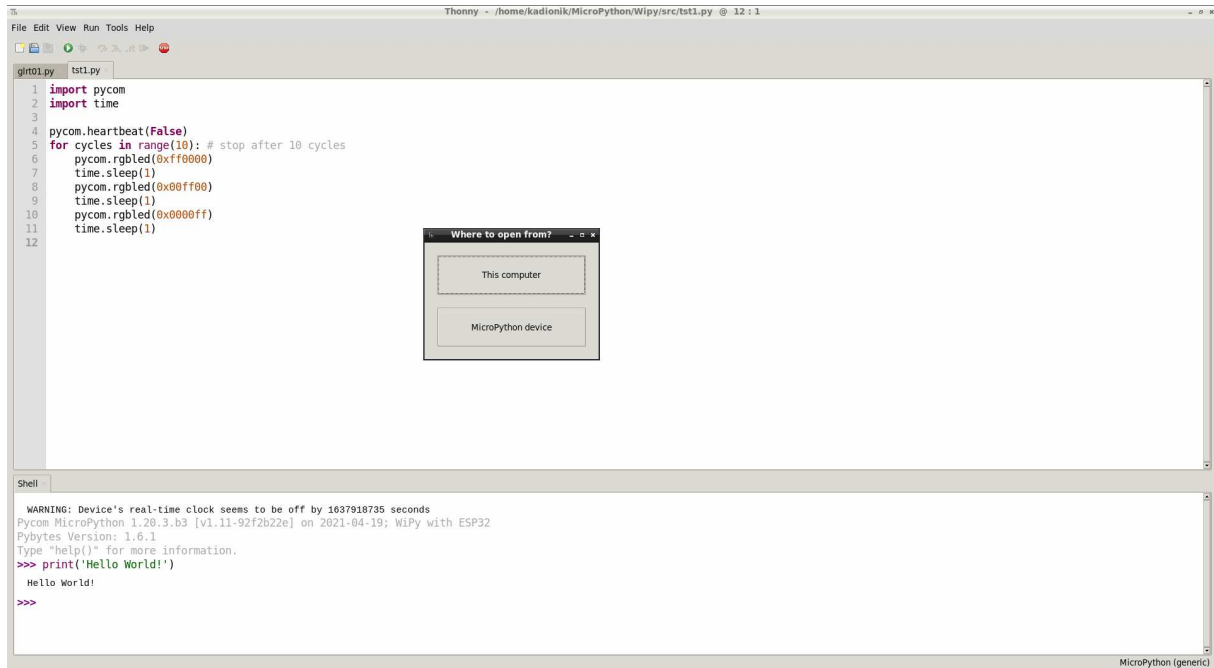
Carte *Pysense*

## 2.2. Environnement de développement

De nombreuses bibliothèques MicroPython permettent de développer simplement sur la carte *Wipy*. Il existe par exemple des bibliothèques pour piloter très simplement la carte *Pysense*.

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 *Wipy* 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 Wipy.

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 changer indéfiniment la couleur de la led en rouge puis verte puis bleue de la carte *Wipy* 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 *Wipy*. 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. On aura au départ avant la connexion Wifi la led de la carte *Wipy* de couleur rouge puis après connexion Wifi, la led de couleur verte. 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 *Wipy*. Tester.

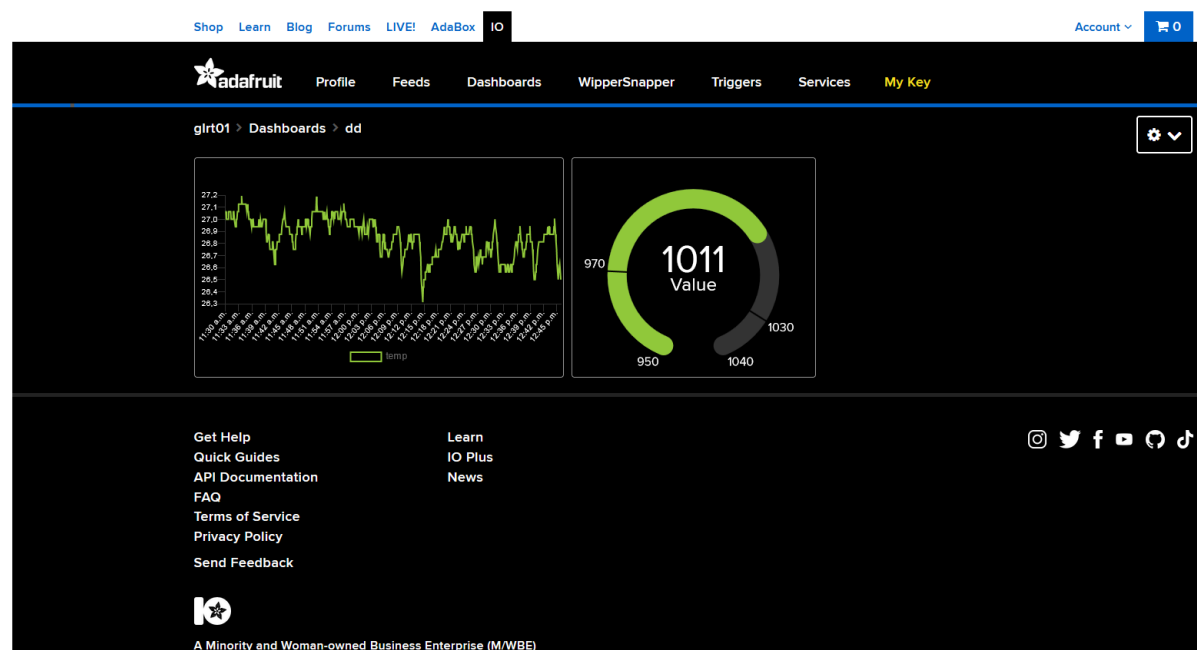
On pourra utiliser les informations données en annexe 4 pour la structuration du nom des 2 *topics* MQTT, pour les noms et mots de passe MQTT pour chaque cible *Wipy*.

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](https://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](https://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](https://io.adafruit.com), créer un nouveau dashboard pour représenter graphiquement via des *widjets* la température sous forme d'un graphique et la pression atmosphérique sous forme d'une gauge. Tester.



*Dashboard io.adafruit.com*



## 9. REFERENCES

- Carte *Wipy* : <https://pycom.io/product/wipy-3-0/>
- Carte *Pysense* : <https://pycom.io/product/pysense-2-0-x/>
- Documentation Pycom sur les cartes *Wipy* et *Pysense* :  
<https://docs.pycom.io/gettingstarted/>
- Documentation MicroPython : <http://docs.micropython.org/en/latest/index.html>
- Documentation MicroPython sur la carte *Wipy* :  
<http://docs.micropython.org/en/latest/wipy/quickref.html>
- Apprendre Python : <http://apprendre-python.com/>

## **10. ANNEXE 1 : RESUME DU LANGAGE PYTHON**

## **11. ANNEXE 2 : MEMENTO DU LANGAGE PYTHON**

## 12. ANNEXE 3 : CONFIGURATION RESEAU HOTES ET CIBLES

<b>POSTE PC01</b>		
	<b>NOM</b>	<b>ADRESSE IP</b>
<b>HOTE</b>	rpi001	10.7.2.118
<b>CIBLE</b>	wipy01	DNS Wifi 192.168.5.x

<b>POSTE PC02</b>		
	<b>NOM</b>	<b>ADRESSE IP</b>
<b>HOTE</b>	rpi002	10.7.2.119
<b>CIBLE</b>	wipy02	DNS Wifi 192.168.5.x

<b>POSTE PC03</b>		
	<b>NOM</b>	<b>ADRESSE IP</b>
<b>HOTE</b>	rpi003	10.7.2.120
<b>CIBLE</b>	wipy03	DNS Wifi 192.168.5.x

<b>POSTE PC04</b>		
	<b>NOM</b>	<b>ADRESSE IP</b>
<b>HOTE</b>	rpi004	10.7.2.121
<b>CIBLE</b>	wipy04	DNS Wifi 192.168.5.x

<b>POSTE PC05</b>		
	<b>NOM</b>	<b>ADRESSE IP</b>
<b>HOTE</b>	rpi005	10.7.2.122
<b>CIBLE</b>	wipy05	DNS Wifi 192.168.5.x

<b>POSTE PC06</b>		
	<b>NOM</b>	<b>ADRESSE IP</b>
<b>HOTE</b>	rpi006	10.7.2.123
<b>CIBLE</b>	wipy06	DNS Wifi 192.168.5.x

**13. ANNEXE 4 : CONFIGURATION DES CIBLES WIPY**

<b>POSTE PC01</b>	
<b>Nom</b>	wipy01
<b>Nom compte io.adafruit.com</b>	se01
<b>Passwd compte io.adafruit.com</b>	voir en TP
<b>Username MQTT</b>	se01
<b>Passwd MQTT</b>	voir onglet « My Key » sur io.adafruit.com
<b>Topic MQTT température</b>	se01/feeds/temp
<b>Topic MQTT pression</b>	se01/feeds/pression

<b>POSTE PC02</b>	
<b>Nom</b>	wipy02
<b>Nom compte io.adafruit.com</b>	se02
<b>Passwd compte io.adafruit.com</b>	voir en TP
<b>Username MQTT</b>	se02
<b>Passwd MQTT</b>	voir onglet « My Key » sur io.adafruit.com
<b>Topic MQTT température</b>	se02/feeds/temp
<b>Topic MQTT pression</b>	se02/feeds/pression

<b>POSTE PC03</b>	
<b>Nom</b>	wipy03
<b>Nom compte io.adafruit.com</b>	se03
<b>Passwd compte io.adafruit.com</b>	voir en TP
<b>Username MQTT</b>	se03
<b>Passwd MQTT</b>	voir onglet « My Key » sur io.adafruit.com
<b>Topic MQTT température</b>	se03/feeds/temp
<b>Topic MQTT pression</b>	se03/feeds/pression

<b>POSTE PC04</b>	
<b>Nom</b>	wipy04
<b>Nom compte io.adafruit.com</b>	se04
<b>Passwd compte io.adafruit.com</b>	voir en TP
<b>Username MQTT</b>	se04
<b>Passwd MQTT</b>	voir onglet « My Key » sur io.adafruit.com
<b>Topic MQTT température</b>	se04/feeds/temp
<b>Topic MQTT pression</b>	se04/feeds/pression

---

<b>POSTE PC05</b>	
<b>Nom</b>	wipy05
<b>Nom compte io.adafruit.com</b>	se05
<b>Passwd compte io.adafruit.com</b>	voir en TP
<b>Username MQTT</b>	se05
<b>Passwd MQTT</b>	voir onglet « My Key » sur io.adafruit.com
<b>Topic MQTT température</b>	se05/feeds/temp
<b>Topic MQTT pression</b>	se05/feeds/pression

<b>POSTE PC06</b>	
<b>Nom</b>	wipy06
<b>Nom compte io.adafruit.com</b>	se06
<b>Passwd compte io.adafruit.com</b>	voir en TP
<b>Username MQTT</b>	se06
<b>Passwd MQTT</b>	voir onglet « My Key » sur io.adafruit.com
<b>Topic MQTT température</b>	se06/feeds/temp
<b>Topic MQTT pression</b>	se06/feeds/pression