
Présentation de la SAE

1.02

DORO Antoine

HOUTMANN Cylian-Nataï

CHÉLONÉ Émilie

CALPÉTARD Mahé

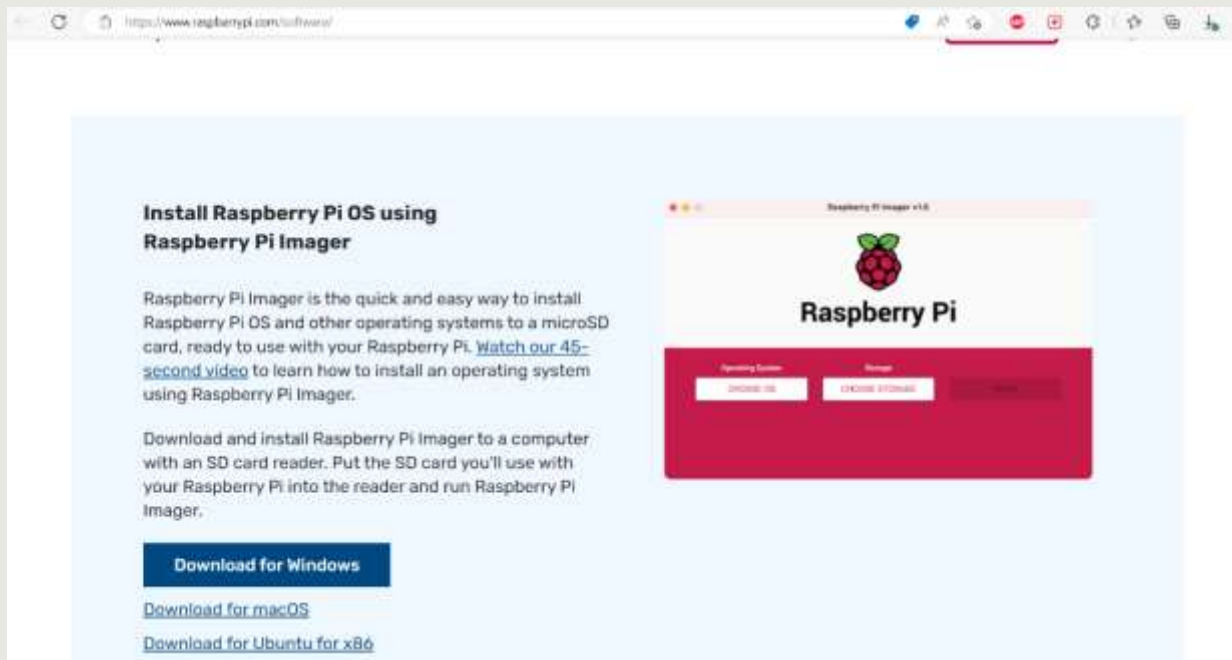
Sommaire

- Configuration de la Raspberry PI
- Configuration du broker sur Ubidots
- Configuration du broker sur la Raspberry PI
- Vérification du bon fonctionnement du système

Configuration de la Raspberry PI

- Installation de Raspberry Pi OS
- Configuration de premier démarrage
- Installation de SSH

Installation de l'image Raspberry Pi OS sur une carte SD

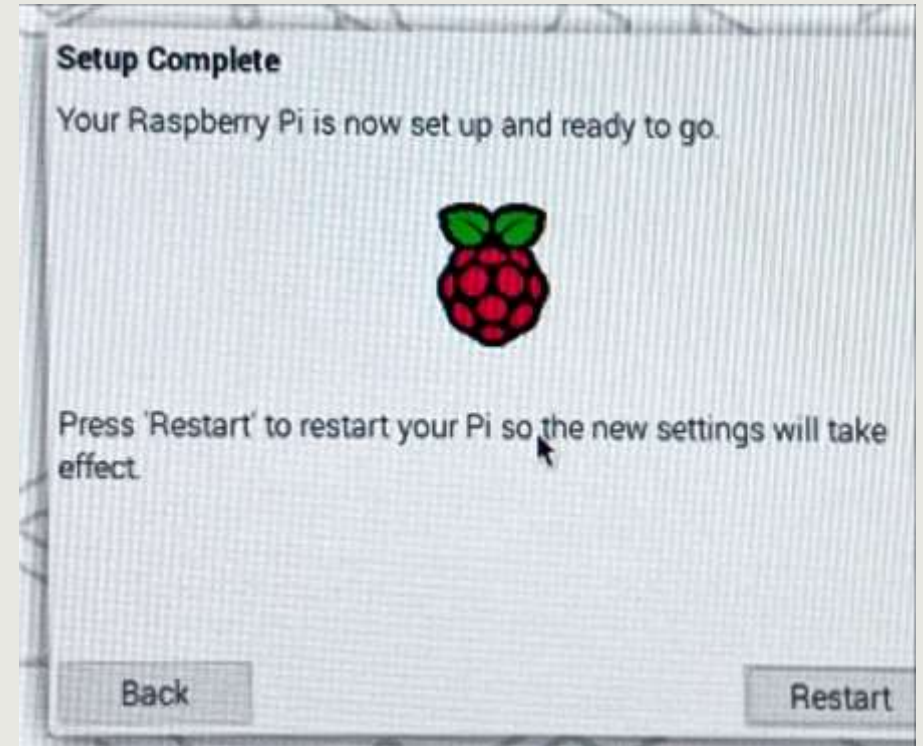


- Installer le client sur sa machine
- Insérer la carte SD sur sa machine
- Sur le support, sélectionner l'image à copier sur la carte SD
- Sélectionner la carte SD
- Lancer la copie

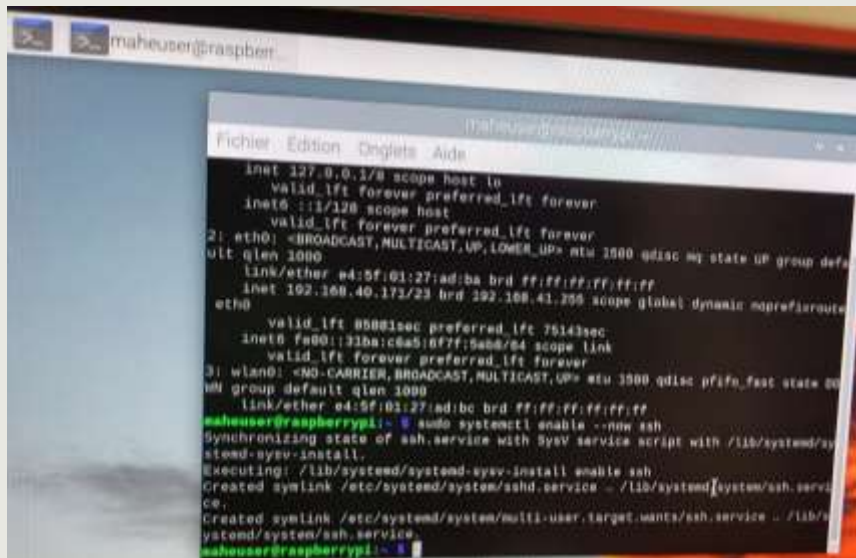
Installation de l'OS sur la Raspberry PI

Mise à jour de la Raspberry PI après l'installation :

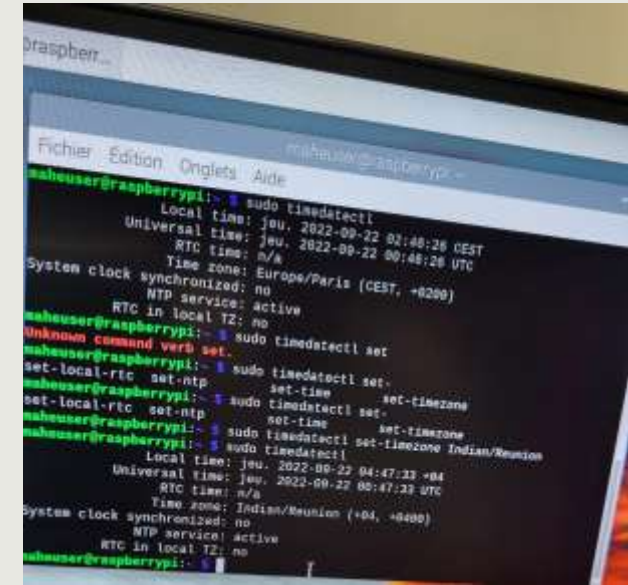
```
sudo apt-get update && sudo apt full-upgrade && sudo apt autoremove
```



Activation de SSH



```
maheuser@raspberrypi:~$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/sy
stemd-sysv-install.
Executing: /lib/systemd/sd-sysv-install enable ssh
Created symlink /etc/systemd/system/ssh.service → /lib/sy
stemd/system/ssh.service.
maheuser@raspberrypi:~$
```



```
maheuser@raspberrypi:~$ sudo timedatectl
Local time: jeu. 2022-09-22 02:48:26 CEST
Universal time: jeu. 2022-09-22 00:48:26 UTC
RTC time: n/a
Time zone: Europe/Paris (CEST, +0200)
System clock synchronized: no
NTP service: active
RTC in local TZ: no
maheuser@raspberrypi:~$ sudo timedatectl set-
Unknown command verb set.
maheuser@raspberrypi:~$ sudo timedatectl set-
set-local-rtc set-ntp set-time set-timezone
maheuser@raspberrypi:~$ sudo timedatectl set-
set-local-rtc set-ntp set-time set-timezone
maheuser@raspberrypi:~$ sudo timedatectl set-timezone Indian/Reunion
Local time: jeu. 2022-09-22 04:47:33 +04
Universal time: jeu. 2022-09-22 00:47:33 UTC
RTC time: n/a
Time zone: Indian/Reunion (+04, +0400)
System clock synchronized: no
NTP service: active
RTC in local TZ: no
maheuser@raspberrypi:~$
```

Commandes utilisées :

sudo apt install ssh

➤ sudo systemctl enable --now ssh

➤ Sudo timedatectl set-timezone Indian/Reunion

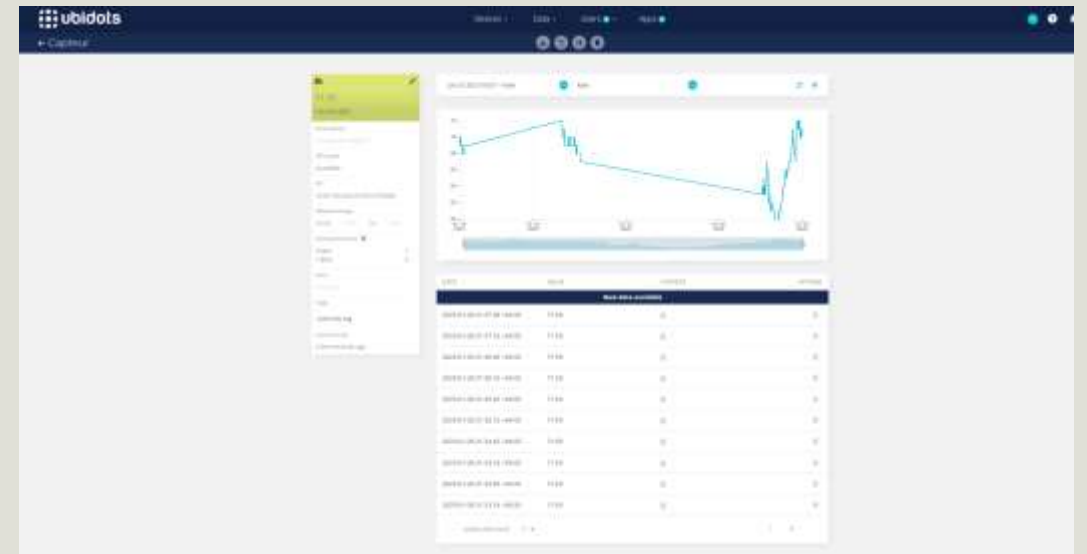
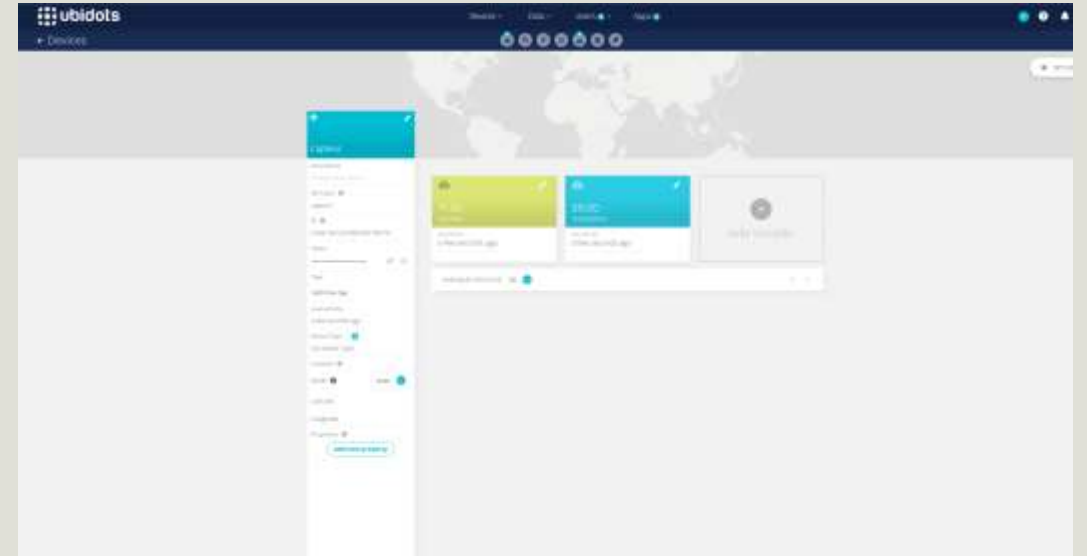
Configuration du broker en ligne

- Création d'un compte sur Ubidots
- Création d'un appareil sur Ubidots avec comme mesures "temperature" et "humidite"

Création du compte sur Ubidots et connexion au dashboard



Création d'un appareil sur Ubidots



Configuration du broker sur la Raspberry PI

- Configuration du script python

```
# Importation des librairies
import board
import digitalio
import time

import adafruit_dht
import paho.mqtt.client as mqtt
import ssl

# Sélection de la led
led = digitalio.DigitalInOut(board.D27)
led.direction = digitalio.Direction.OUTPUT

# Sélection du DHT
dht_device = adafruit_dht.DHT11(board.D17)

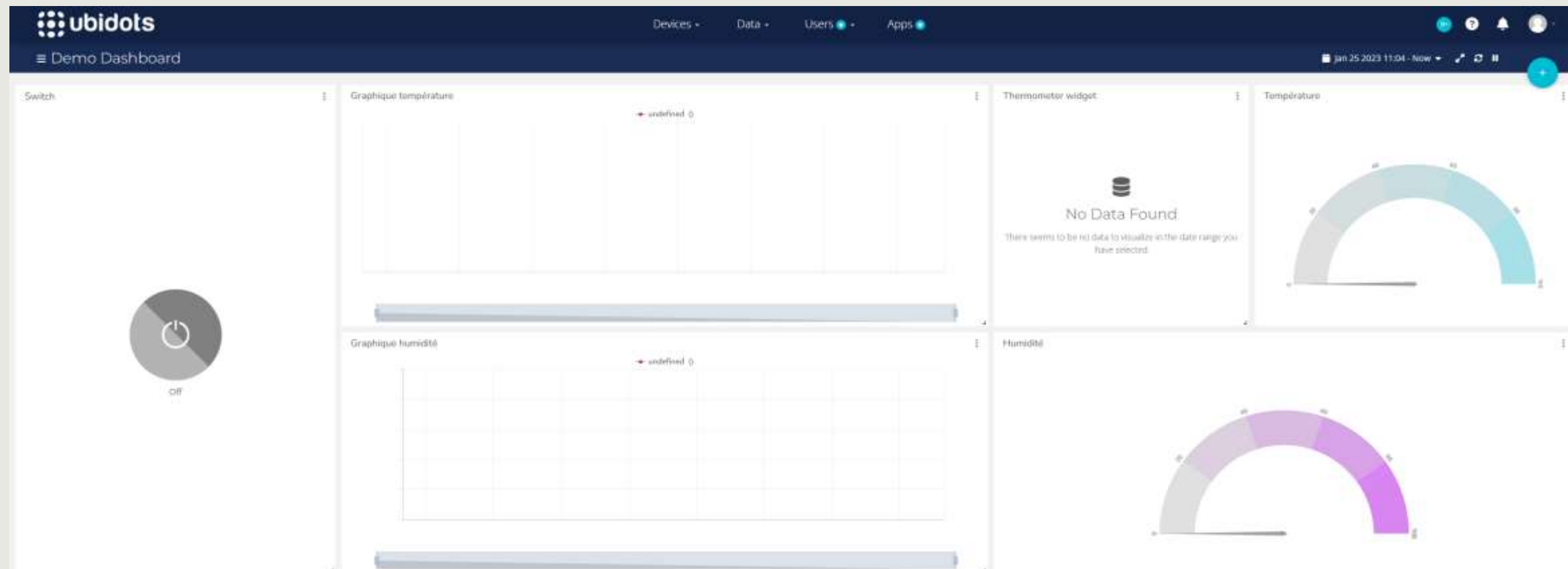
# CLIENT MQTT
topic = "/v1.6/devices/capteur"
TLS_CERT_PATH="/home/user/Documents/tls.pem"

client = mqtt.Client()
client.username_pw_set("BBFF-SQbdy4VWB1xfAqrjU8DYkXntB55NQu","")
client.tls_set(ca_certs=TLS_CERT_PATH, certfile=None, keyfile=None, cert_reqs=ssl.CERT_REQUIRED, tls_version=ssl.PROTOCOL_TLSv1_2, ciphers=None)
client.tls_insecure_set(False)
client.connect("industrial.api.ubidots.com", 8883)

# Boucle de données
while (1) :
    msg = '{"Température" : %s, "Humidité" : %s}' % (dht_device.temperature, dht_device.humidity)
    client.publish(topic,msg)
    led.value = False
    time.sleep(30)
    led.value = True
```

Vérification du bon fonctionnement du système

- Réception des données du capteur



Conclusion

[Site de mesure](#)

Cette SAE nous a permis d'apprendre le protocole MQTT par le biais d'une Raspberry pi prélevant des mesures grâce à l'utilisation d'un DHT11.
