

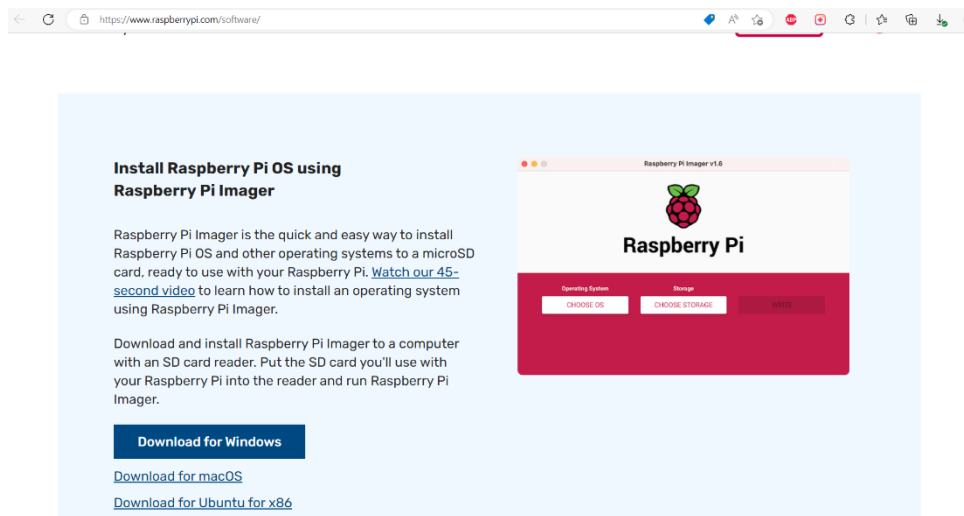
SAÉ-102 S'initier aux réseaux informatiques

1) Configuration d'une carte Raspberry Pi connectée

Dans un premier temps nous allons configurer la carte Raspberry Pi afin de pouvoir la programmer sans avoir à brancher systématiquement un écran et un clavier. L'objectif est de pouvoir se connecter en SSH, indépendamment du réseau sur lequel la carte est connectée.

Étape 1 : Installation du système d'exploitation de la carte Raspberry Pi

Téléchargement de l'installateur sur le site de Raspberry

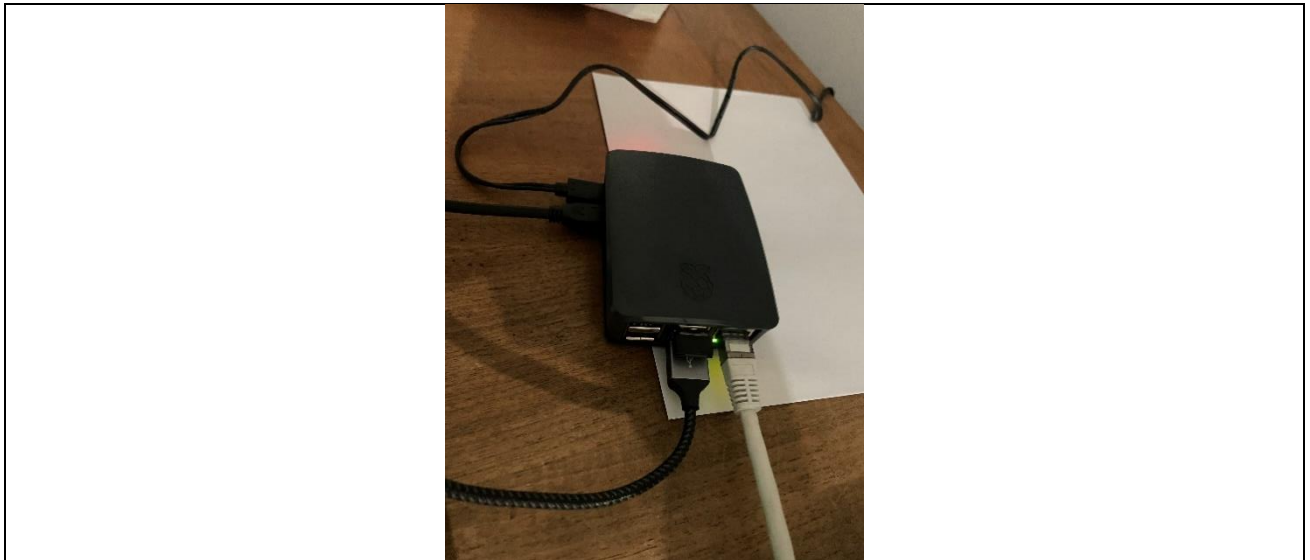


Téléversement du fichier iso sur la carte micro SD

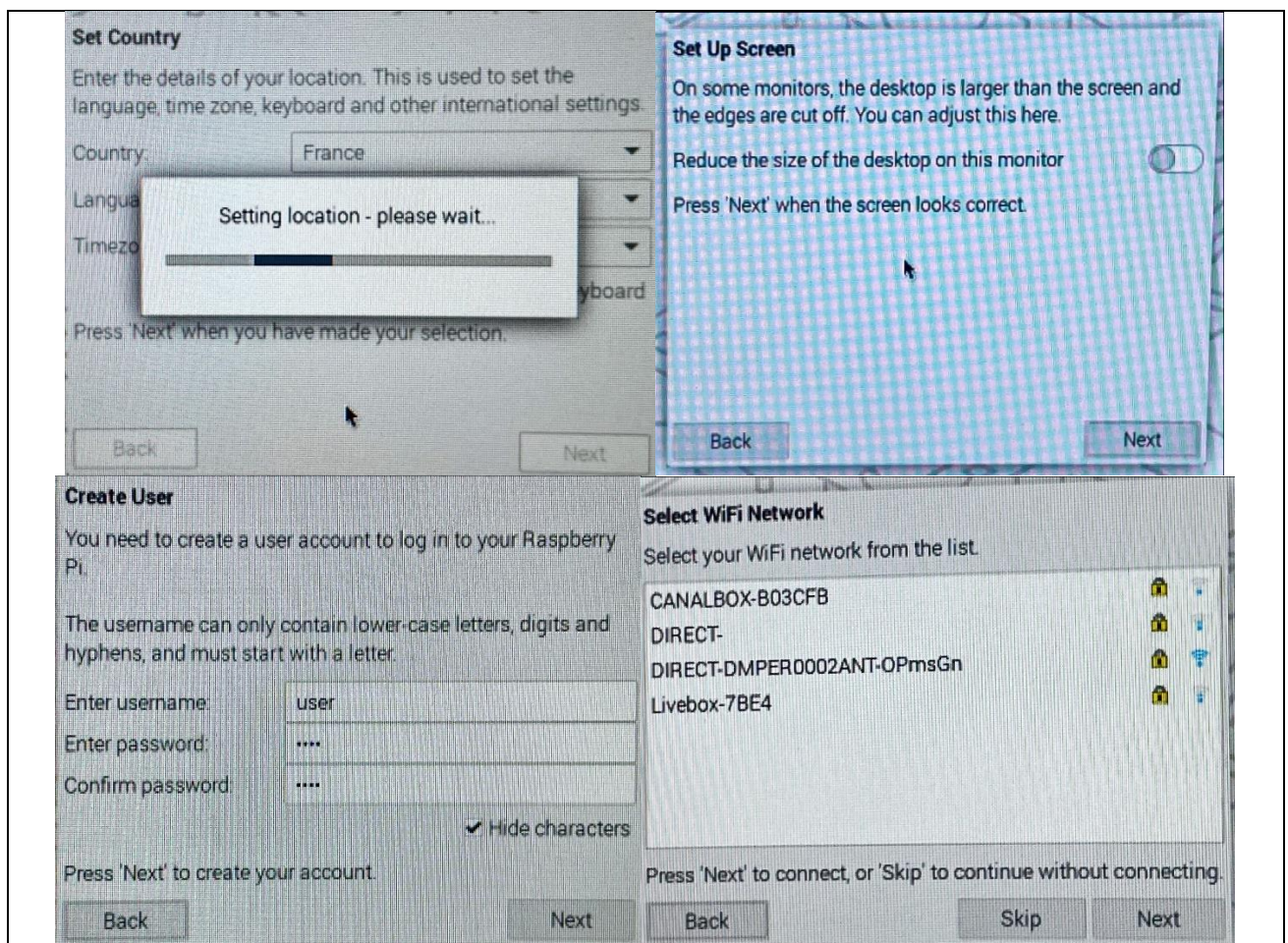


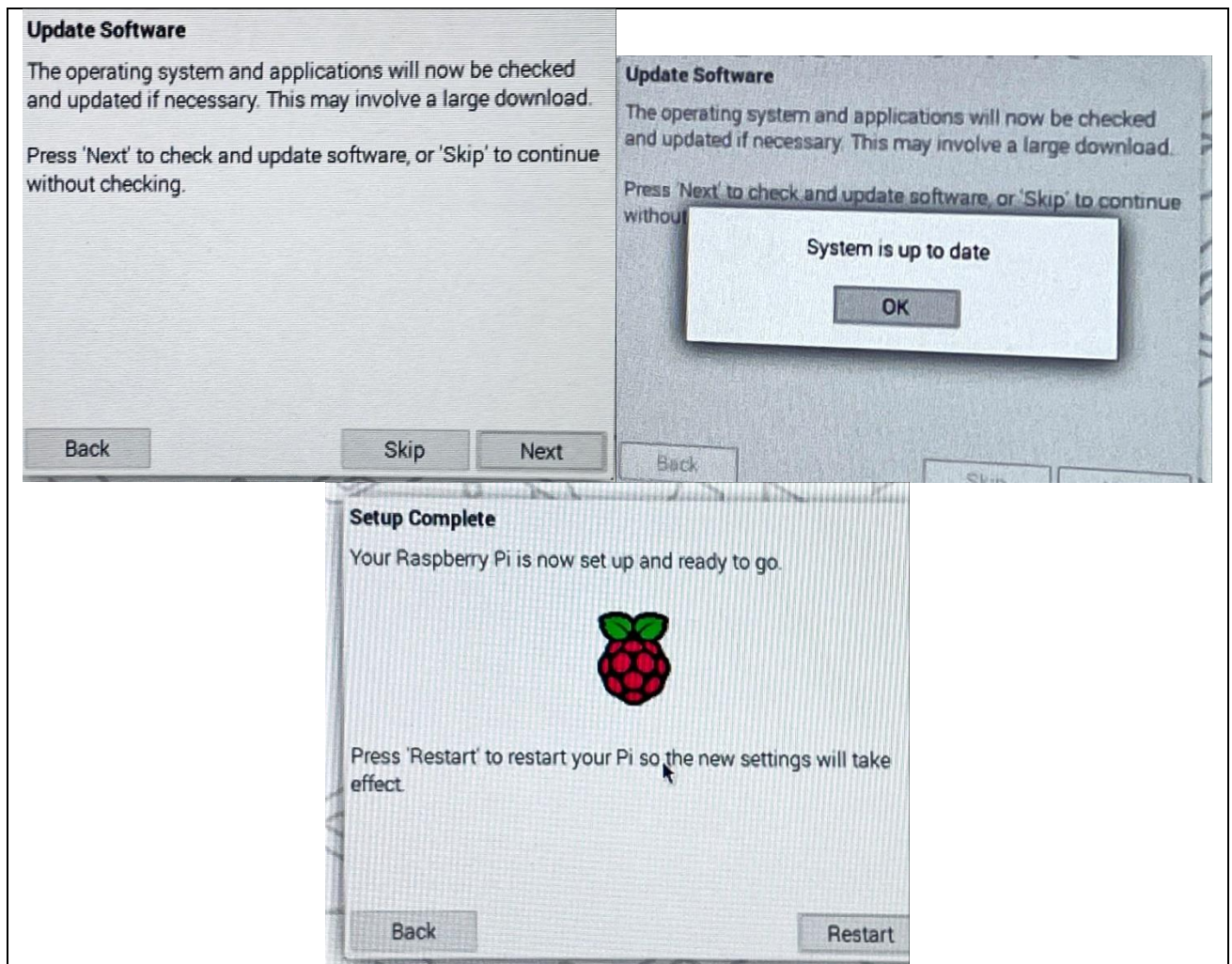
Étape 2 : Démarrage de la carte

Raccordement des périphériques

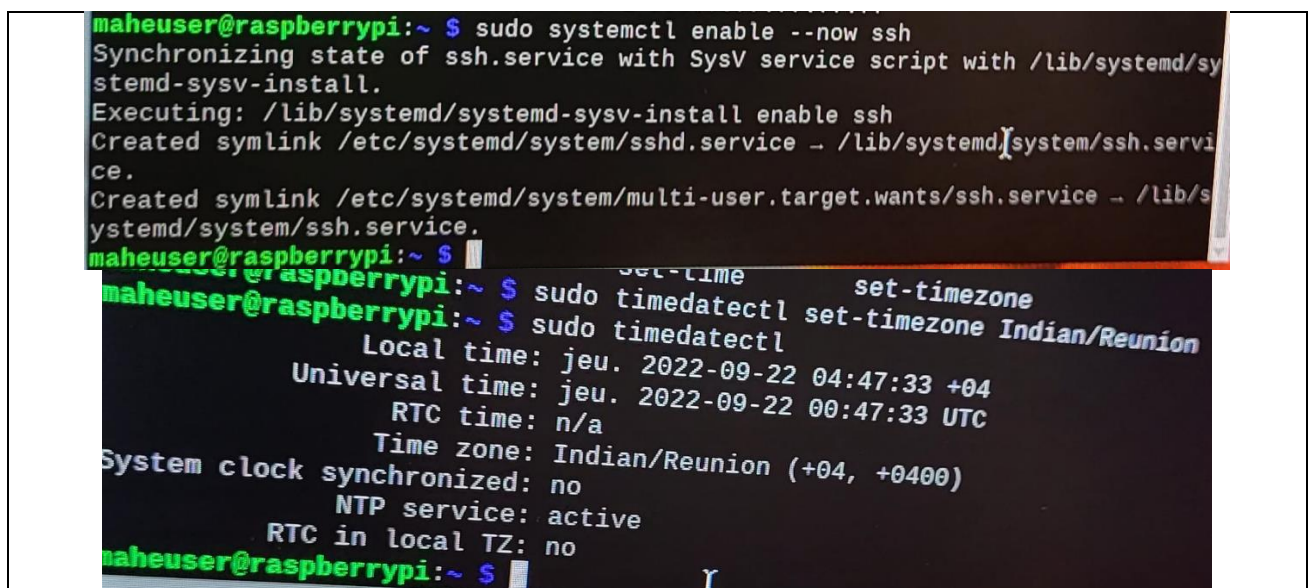


Alimentation et premier démarrage de la carte





Procédure de configuration du SSH et de l'heure



Pour le SSH, nous devons d'abord l'installer avec "**sudo apt install ssh**" puis activer le service "**sudo systemctl enable --now ssh**". Pour que la date soit à la bonne heure, nous tapons la commande "**sudo timedatectl set-timezone Indian/Reunion**".

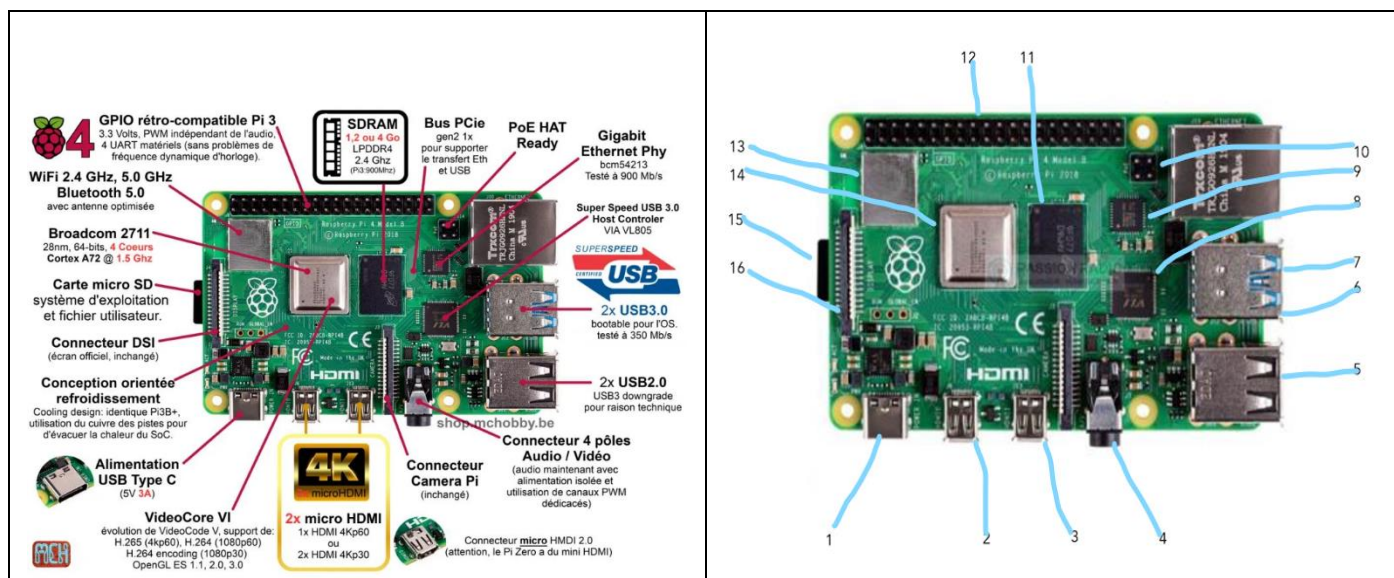
Ensuite nous récupérons l'adresse MAC de notre Raspberry (**e4 :5f :01 :27 :ad :ba**) avec la commande "**ip a**". Cela nous permettra de récupérer l'adresse IP afin de pouvoir se connecter en SSH sans écran lors des prochaines manipulations.

Nous utilisons Wireshark avec le filtre **eth.addr==<adresse MAC>**, pour filtrer l'adresse MAC de notre Raspberry afin de récupérer son adresse IP.

2) Données techniques, caractéristique et fonctionnalités de la carte Raspberry Pi

			Commande utilisée / source de l'information
Raspberry Pi	Numéro de série		
	Modèle	Raspberry Pi 4	neofetch
Processeur	Modèle	Broadcom BCM2711	
	Architecture	ARM v8 64-bit	
	Nombre de CPU	1 CPU de 4 cœurs	neofetch
	Nombre de Tread / CPU	0	
	Fréquence Max (Hz)		
	Fréquence Min (Hz)	1.5GHz	
Mémoire	Mémoire vive	8 Go	
	Mémoire de stockage	32 Go	lsblk
Carte graphique	Modèle		
	Taille mémoire		
Carte Réseau	@Mac Ethernet	e4 :5f :01 :27 :ad :ba	ip a
	Débit	1Gbit/s	
	@Mac wifi	E4 :5f :01 :27 :ad :bc	(comme le wifi ai pas utilisé il faut utiliser l'option -d pour afficher les périphériques inactifs) ip -d a
	Bluetooth	Oui la version 5.5	« Bluetoothctl » puis taper « version »
Système d'exploitation	Nom / Version	Raspbian	neofetch

	Environnement graphique	lxde	apt list --installed grep lxde
Ports et interfaces	Nombre Entrée / sortie	40	
	Nombre de sortie analogique	26	
	HDMI	2 micros hdmi	
Alimentation	Tension V	5.1V	
	Intensité A	3A	
Contraintes	Température de fonctionnement	0 – 50 degrees C ambient	
	Tension limite sur les entrées / sorties	5V	

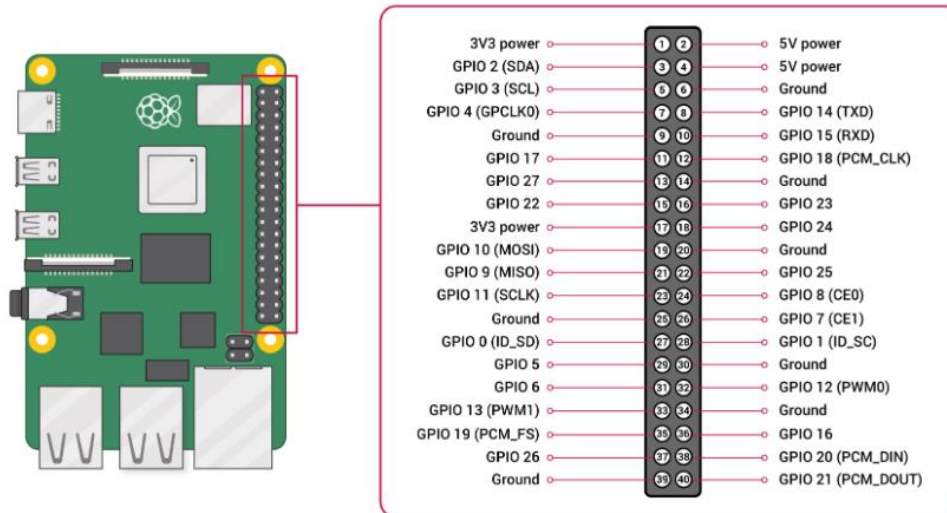


1:	Alimentation USB Type-C
2:	Micro HDMI
3:	Connecteur caméra Pi
4:	Connecteur 4 pôles audio/vidéo
5:	USB 2.0
6:	USB 3.0
7:	USB 3.0
8:	Super Speed USB 3.0 Host Controller
9:	Gigabit Ethernet Phy
10:	PoE HAT Ready

11:	SDRAM 1,2 ou 4Go
12:	GPIO retro-compatible Pi 3
13:	Wifi 2.4 GHz, 5.0 GHz, Bluetooth 5.0
14:	Broadcom 2711
15:	Carte micro-SD
16:	Connecteur DSI

3) Les entrées et sorties de la carte Raspberry Pi et montage des différents composants

Nous allons prendre connaissance des caractéristiques de l'interface GPIO (General Purpose Input Output) de la carte Raspberry Pi.



Nous voyons que les tensions disponibles sur cette interface GPIO sont 3,3V et 5V. La tension maximale à ne pas dépasser est 5V. Dans notre cas nous nous limiterons à une tension maximale de 3,3V.

Nous pouvons faire un montage permettant d'obtenir un schéma de câblage pour

- Interfacer un capteur de température
- Piloter une LED

Notre montage est composé de :

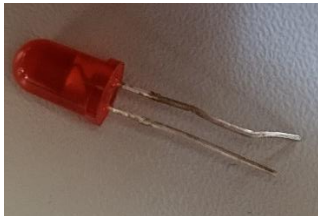
- Une carte Raspberry Pi 4 :



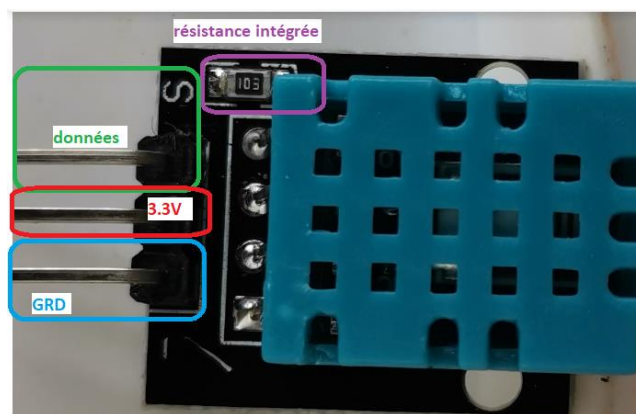
- Des LEDs
 - o Une LED bleue, pour signaler la mise sous tension du capteur :



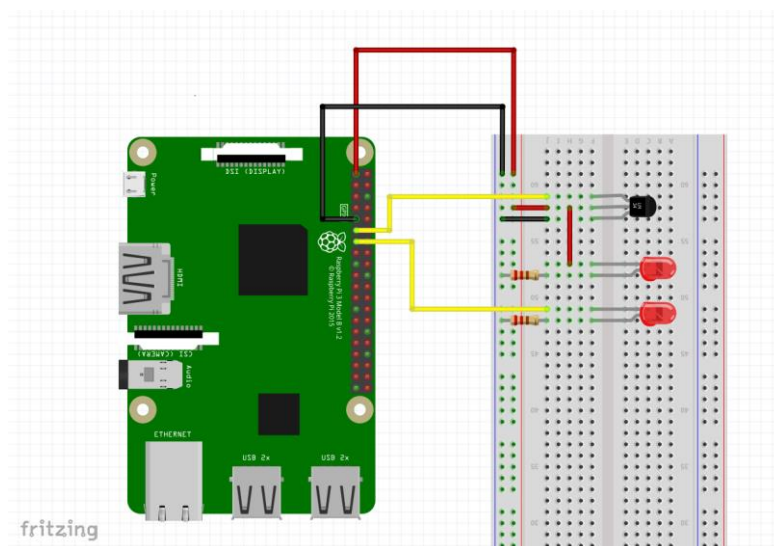
- Une LED rouge, pour signaler l'envoi d'une requête MQTT :



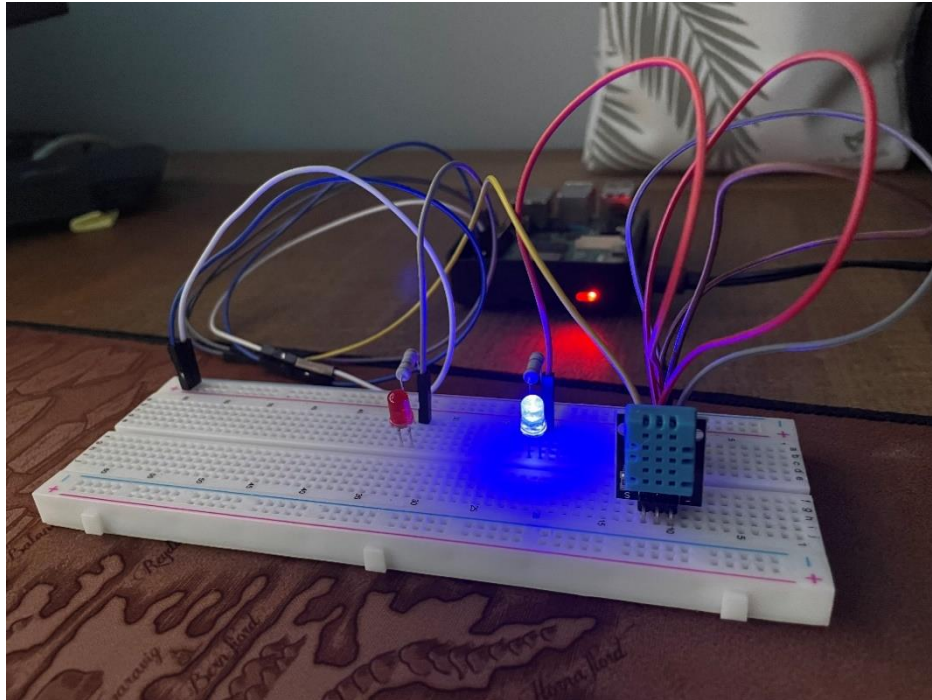
- Un capteur de température DHT11 :



Nous obtenons donc ce résultat :



Nous pouvons maintenant mettre en place notre câblage avec la vrai Raspberry Pi.



4) Réalisation d'un script python, afin de transmettre les données récupérées par le capteur au broker

Pour réaliser notre script, nous importerons les librairies : adafruit (pour la récupération des données du capteur), paho.mqtt.client (pour l'envoi des données au broker) et ssl (pour des échanges sécurisée avec SSL).

```
# 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
```

Informations d'envois au broker
URL : industrial.api.ubidots.com

Port : 8883 à cause du SSL/TLS

Nom d'utilisateur (sans mot de passe) : <BBFF-SQbdy4VWB1xfAqrjU8DYkXntB55NQu>x

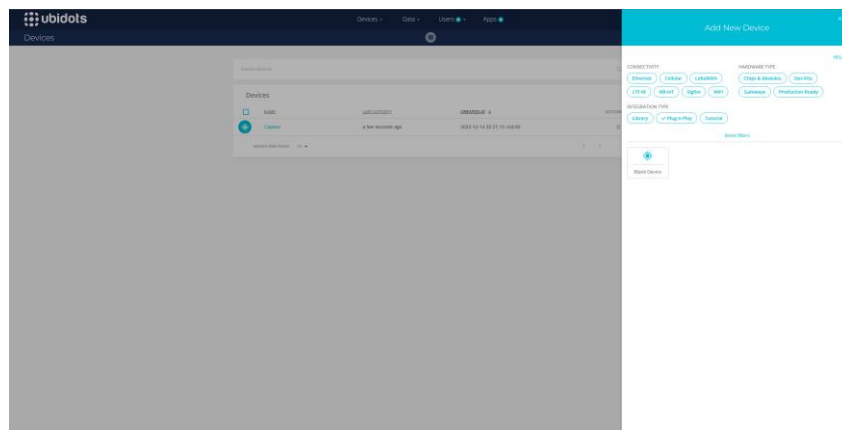
Topics : /V1.6/devices/capteur/

Les données seront envoyées au format json, afin d'être interpréter au niveau du broker.

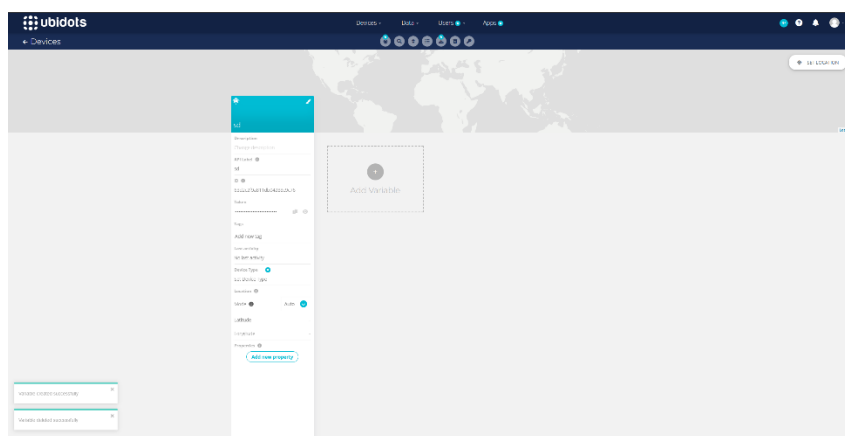
5) Paramétrage du broker

Pour notre broker, nous avons choisis Ubidots. Il nous permettra de récupérer les données, de les afficher sous format graphique et de les analysées statistiquement.

Etape 1 : création de notre capteur

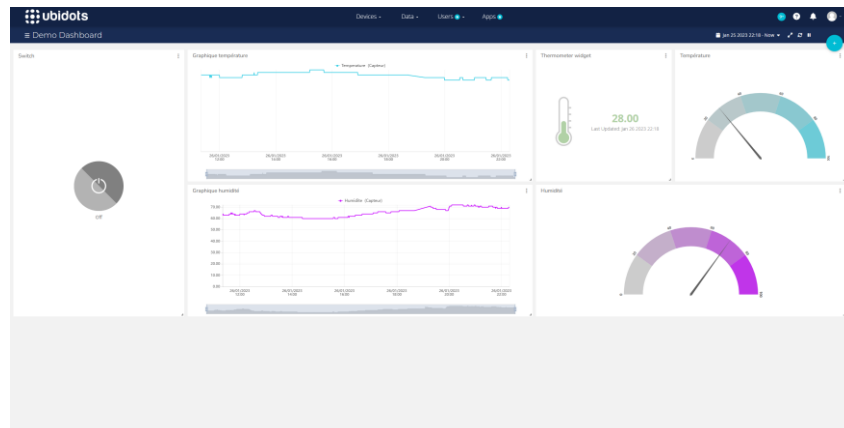


Etape 2 : création des variables



On nomme les variables en fonction du nom des données envoyées au format json.

Etape 3 : Affichage graphique des données



6) Exportation des données du broker pour un affichage public

Nous pouvons avec Ubidots, exporter facilement les modules d'affichage sur une page web, grâce à des iframe. Le site en-dessous est créer grâce à l'implémentation de ces iframe ([Site Web](#)).

