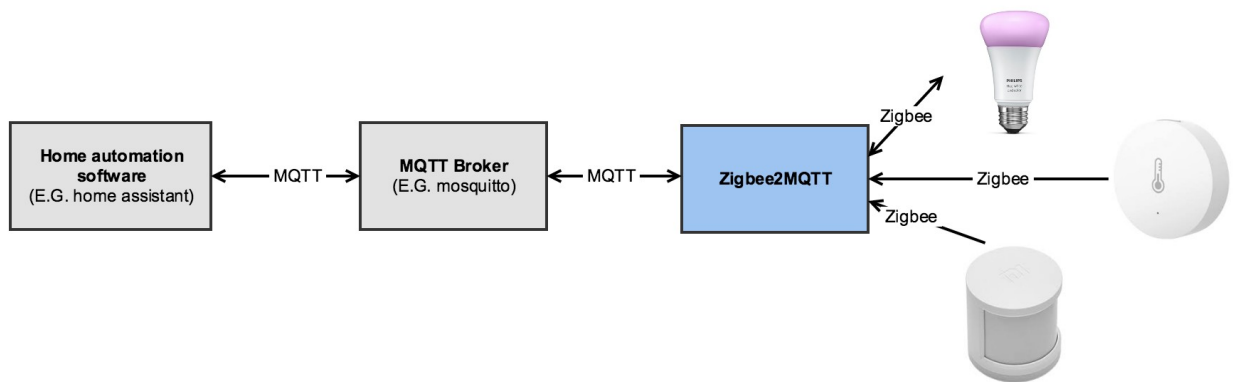


Passerelle ZigBee2MQTT	
Sommaire :	
<i>I - Présentation</i>	1
<i>II - Installation sur Debian 10 « Buster »</i>	1
II.1. Préparation et connexion de l'adaptateur ZigBee.....	1
II.2. Installation de Node.js version 18.x.....	2
II.3. Installation de Mosquitto (Broker MQTT).....	2
II.4. Installation de ZigBee2MQTT.....	3
II.5. Configuration de ZigBee2MQTT.....	3
II.6. Utilisation de l'interface Web.....	4
II.7. Utilisation des commandes MQTT.....	5
II.8. Démarrage automatique Zgbee2Mqtt.....	6

I - Présentation

<https://github.com/Koenkk/zigbee2mqtt>

Comme le montre la figure ci-dessous, la passerelle **ZigBee2MQTT** permet de piloter des périphériques **Zigbee** via le protocole **MQTT** :



II - Installation sur Debian 10 « Buster »

II.1. Préparation et connexion de l'adaptateur ZigBee

Il faut disposer d'un adaptateur **Zigbee** de type **SONOFF - ZBDONGLE-P** ou un dongle **USB TI CC2531** qui ne supporte que le **Zigbee 2.0**.

Si l'on utilise un dongle **USB TI CC2531**, il faut le flasher avec le firmware **CC2531ZNP-Prod.hex** (voir l'adresse https://github.com/Koenkk/Z-Stack-firmware/raw/master/coordinator/Z-Stack_Home_1.2/bin/default/CC2531_DEFAULT_20211115.zip).

Si l'on utilise un dongle **USB SONOFF - ZBDONGLE-P**, il faut le flasher avec **cc2538-bsl.py** :
`sudo python3 cc2538-bsl.py -ewv -p /dev/ttyUSB0 --bootloader-sonoff-usb ./CC1352P2_CC2652P_launchpad_coordinator_20220219.hex`

Il faut ensuite connecter l'adaptateur **ZigBee** sur le port **USB** du **Raspberry Pi**. On le localise avec les commandes suivantes :

Cas de l'adaptateur **USB TI CC2531** :

```
ls -l /dev/ttyACM0
crw-rw---- 1 root dialout 166, 0 May 16 19:15 /dev/ttyACM0

ls -l /dev/serial/by-id
total 0
lrwxrwxrwx 1 root root 13 août 10 08:36 usb-
Texas_Instruments_TI_CC2531_USB_CDC___0X00124B00016F0F3C-if00 ->
../ttyACM0
```

Cas de l'adaptateur **SONOFF - ZBDONGLE-P** :

```
ls -l /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 0 août 13 12:39 /dev/ttyUSB0

ls -l /dev/serial/by-id
total 0
lrwxrwxrwx 1 root root 13 août 13 12:39 usb-
ITead_Sonoff_Zigbee_3.0_USB_Dongle_Plus_8cf796495486ec11ae8ae13719c2d2
1c-if00-port0 -> ../ttyUSB0
```

II.2. Installation de Node.js version 18.x

On exécute les commandes suivantes :

```
sudo curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt-get install -y nodejs git make g++ gcc
```

ou en administrateur :

```
curl -sL https://deb.nodesource.com/setup_18.x | bash -
apt-get install -y nodejs git make g++ gcc
```

Si tout c'est bien passé, **Node.js** doit être dans une version **18.x** :

```
node --version
v18.7.0
npm --version
8.15.0
```

II.3. Installation de Mosquitto (Broker MQTT)

On exécute la commande suivante :

```
sudo apt-get install -y mosquitto
```

Pour créer un utilisateur nommé **iot**, il faut exécuter la commande suivante :

```
sudo mosquitto_passwd -c /etc/mosquitto/passwd iot
```

Dans le fichier **/etc/mosquitto/mosquitto.conf**, il faut rajouter les lignes :

```
allow_anonymous false
password_file /etc/mosquitto/passwd
```

Puis on relance le service Mosquitto :

```
sudo service mosquitto restart
```

Remarque : Pour supprimer un utilisateur nommé **user**, il faut exécuter la commande

suivante :

```
mosquitto_passwd -D /etc/mosquitto/passwd user
```

II.4. Installation de ZigBee2MQTT

Remarque : Zigbee2MQTT dans sa version **1.27.0** nécessite **Node.js** version **14**, **16** ou **18**.

On exécute les commandes suivantes :

```
sudo git clone https://github.com/Koenkk/zigbee2mqtt.git /opt/zigbee2mqtt
sudo chown -R pi:pi /opt/zigbee2mqtt
cd /opt/zigbee2mqtt
npm ci
```

Cette dernière commande génère des warnings que l'on ignore.

II.5. Configuration de ZigBee2MQTT

Il faut modifier le fichier **configuration.yaml** en éditant ce dernier avec la commande suivante :

```
nano /opt/zigbee2mqtt/data/configuration.yaml
```

On laisse le serveur MQTT en local :

```
mqtt:
  server: 'mqtt://localhost'
  user: iot
  password: iot
```

Normalement on visualise l'adaptateur :

```
serial:
  adapter: ezsp // Uniquement pour dongle SONOFF ZBDONGLE-E
  port: /dev/ttyACM0 (ou /dev/ttyUSB0)
```

On active l'interface Web de gestion (port 8080) en ajoutant :

```
frontend: true
```

Ou si l'on veut activer l'interface Web de gestion sur le port spécifique 8081 pour tous les hôtes on ajoute :

```
frontend:
  port: 8081
  host: 0.0.0.0
```

Puis on enregistre le fichier et on démarre **Zigbee2MQTT** avec l'utilisateur **pi** :

```
sudo chmod a+rw /dev/ttyACM0
cd /opt/zigbee2mqtt
npm start
```

```
> zigbee2mqtt@1.27.0 start
> node index.js
```

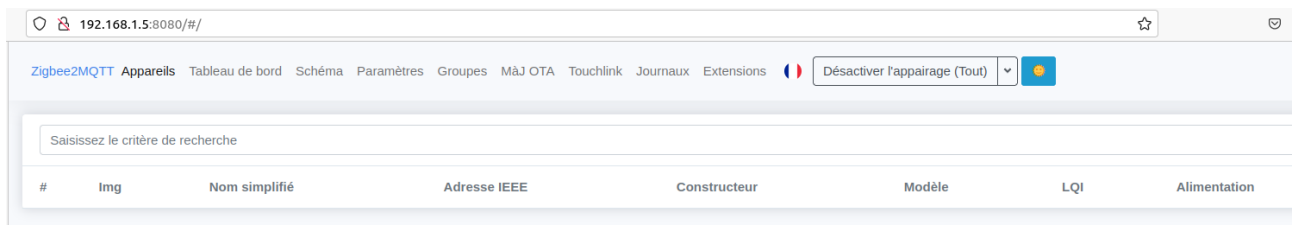
```
Building Zigbee2MQTT... (initial build), finished
Zigbee2MQTT:info 2022-08-10 08:44:18: Logging to console and directory:
'/opt/zigbee2mqtt/data/log/2022-08-10.08-44-18' filename: log.txt
Zigbee2MQTT:info 2022-08-10 08:44:18: Starting Zigbee2MQTT version 1.27.0
(commit #unknown)
Zigbee2MQTT:info 2022-08-10 08:44:18: Starting zigbee-herdsman (0.14.46)
Zigbee2MQTT:info 2022-08-10 08:44:21: zigbee-herdsman started (resumed)
```

```

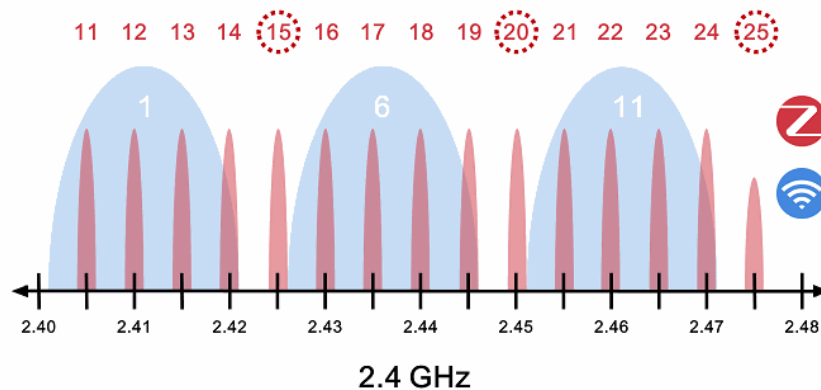
Zigbee2MQTT:info 2022-08-10 08:44:21: Coordinator firmware version:
'{"meta":
{"maintrel":3,"majorrel":2,"minorrel":6,"product":0,"revision":20211115,"transportrev":2},"type":"zStack12"}'
Zigbee2MQTT:info 2022-08-10 08:44:21: Currently 0 devices are joined:
Zigbee2MQTT:warn 2022-08-10 08:44:21: `permit_join` set to `true` in
configuration.yaml.
Zigbee2MQTT:warn 2022-08-10 08:44:21: Allowing new devices to join.
Zigbee2MQTT:warn 2022-08-10 08:44:21: Set `permit_join` to `false` once
you joined all devices.
Zigbee2MQTT:info 2022-08-10 08:44:21: Zigbee: allowing new devices to join.
Zigbee2MQTT:info 2022-08-10 08:44:21: Connecting to MQTT server at
mqtt://localhost
Zigbee2MQTT:info 2022-08-10 08:44:21: Connected to MQTT server
Zigbee2MQTT:info 2022-08-10 08:44:21: MQTT publish: topic
'zigbee2mqtt/bridge/state', payload 'online'
Zigbee2MQTT:info 2022-08-10 08:44:21: Started frontend on port
0.0.0.0:8080
Zigbee2MQTT:info 2022-08-10 08:44:22: MQTT publish: topic
'zigbee2mqtt/bridge/config', payload '{"commit":"unknown","coordinator":
{"meta":
{"maintrel":3,"majorrel":2,"minorrel":6,"product":0,"revision":20211115,"transportrev":2},"type":"zStack12"},"log_level":"info","network":
{"channel":11,"extendedPanID":"0x00124b00016f0f3c","panID":6754},"permit
join":true,"version":"1.27.0"}'
    
```

II.6. Utilisation de l'interface Web

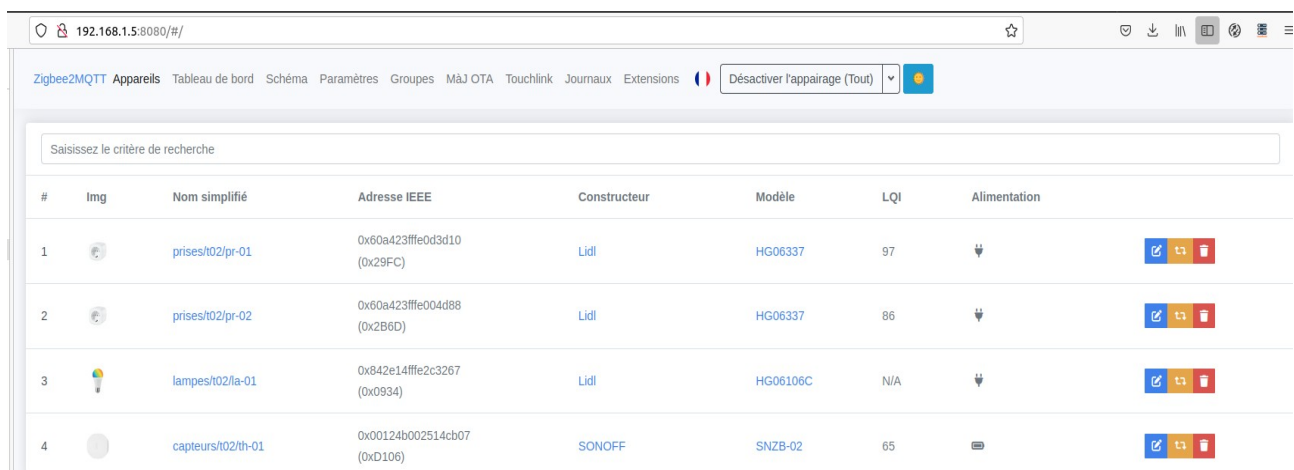
On peut maintenant se connecter à l'interface Web de **ZigBee2MQTT** (http://pi_addr:8080) :







Dans le menu **Paramètres**, on peut modifier la configuration et on choisit le canal **15=0x0F** (ou 20=0x14) pour ne pas interférer avec le Wifi ainsi que le **PAN ID 6754 (0x1A62)** :



On peut maintenant **appairer** à notre réseau **ZigBee** autant de modules que l'on veut :



#	Img	Nom simplifié	Adresse IEEE	Constructeur	Modèle	LQI	Alimentation
1		prises/t02/pr-01	0x60a423ffe0d3d10 (0x29FC)	Lidl	HG06337	97	🔌
2		prises/t02/pr-02	0x60a423ffe004d88 (0x2B6D)	Lidl	HG06337	86	🔌
3		lampes/t02/la-01	0x842e14ffe2c3267 (0x0934)	Lidl	HG06106C	N/A	🔌
4		capteurs/t02/th-01	0x00124b002514cb07 (0xD106)	SONOFF	SNZB-02	65	🔌

Pour **appairer** un module il suffit de regarder la documentation. Par exemple pour une prise ou un capteur il suffit d'appuyer sur le bouton pendant quelques secondes, pour une lampe il suffit de la brancher puis la débrancher plusieurs fois rapidement.

Lorsqu'on commande les appareils, des logs MQTT apparaissent sur le terminal :

```
Zigbee2MQTT:info 2022-08-04 15:03:57: MQTT publish: topic 'zigbee2mqtt/prises/t02/pr-01', payload '{"linkquality":70,"state":"OFF"}'
Zigbee2MQTT:info 2022-08-04 15:03:58: MQTT publish: topic 'zigbee2mqtt/prises/t02/pr-01', payload '{"linkquality":78,"state":"ON"}'
Zigbee2MQTT:info 2022-08-04 15:04:06: MQTT publish: topic 'zigbee2mqtt/lampes/t02/la-01', payload '{"brightness":156,"color":{"x":0.3111,"y":0.3212},"color_mode":"color_temp","color_temp":150,"linkquality":81,"state":"ON"}'
Zigbee2MQTT:info 2022-08-04 15:04:09: MQTT publish: topic 'zigbee2mqtt/lampes/t02/la-01', payload '{"brightness":156,"color":{"x":0.3111,"y":0.3212},"color_mode":"color_temp","color_temp":150,"linkquality":78,"state":"OFF"}'
```

II.7. Utilisation des commandes MQTT

https://www.zigbee2mqtt.io/guide/usage/mqtt_topics_and_messages.html#zigbee2mqtt-friendly-name

On peut s'abonner à ces messages MQTT, par exemple au topic '**zigbee2mqtt/prises/t02/pr-01**' :

```
mosquitto_sub -h 192.168.1.5 -t "zigbee2mqtt/prises/t02/pr-01"
{"linkquality":70,"state":"OFF"}
{"linkquality":78,"state":"ON"}
```

On peut aussi envoyer des commandes MQTT pour commander la prise par exemple :

```
mosquitto_pub -h 192.168.1.5 -t "zigbee2mqtt/prises/t02/pr-01/set" -m '{"state":"ON"}'
```

ou

```
mosquitto_pub -h 192.168.1.5 -t "zigbee2mqtt/prises/t02/pr-01/set/state" -m ON
```

Pour lire l'état de la prise par exemple :

```
mosquitto_pub -h 192.168.1.5 -t "zigbee2mqtt/prises/t02/pr-01/get" -m '{"state":""}'
```

Pour récupérer l'ensemble des devices par exemple :

```
mosquitto_sub -h 192.168.1.5 -t "zigbee2mqtt/bridge/devices"
```

II.8. Démarrage automatique Zigbee2Mqtt

Il faut d'abord créer un nouveau fichier de configuration qui va nous servir à démarrer **ZigBee2MQTT** au boot du Raspberry Pi comme un service.

```
sudo nano /etc/systemd/system/zigbee2mqtt.service
```

Puis on colle les lignes de configuration ci-dessous dans le fichier :

```
[Unit]  
Description=zigbee2mqtt  
After=network.target  
  
[Service]  
ExecStart=/usr/bin/npm start  
WorkingDirectory=/opt/zigbee2mqtt  
StandardOutput=inherit  
StandardError=inherit  
Restart=always  
User=pi  
  
[Install]  
WantedBy=multi-user.target
```

On enregistre le fichier puis on quitte via **CTRL + X** puis confirmation avec **Y** et enfin Entrée.

On vérifie que notre configuration fonctionne correctement avec une exécution manuelle depuis la commande :

```
sudo systemctl start zigbee2mqtt
```

On affiche le statut avec la commande :

```
systemctl status zigbee2mqtt.service
```

Si le service est dans l'état **active (running)** tout va bien. On interrompt le service avec la combinaison **CTRL + C** pour retrouver le prompt. Enfin, maintenant que nous nous sommes assurés que tout est fonctionnel, on peut configurer le démarrage automatique avec la commande :

```
sudo systemctl enable zigbee2mqtt.service
```

La commande informe de la création du lien symbolique. Pour finir et confirmer le bon fonctionnement de notre configuration du démarrage automatique, il convient de lancer un redémarrage du Raspberry Pi à l'aide de la commande :

```
sudo reboot
```

A présent, le système est automatisé. À des fins de maintenance vous pouvez agir sur le service avec les commandes suivantes :

```
# Stopping Zigbee2MQTT  
sudo systemctl stop zigbee2mqtt
```

```
# Starting Zigbee2MQTT  
sudo systemctl start zigbee2mqtt
```

```
# View the log of Zigbee2MQTT  
sudo journalctl -u zigbee2mqtt.service -f
```