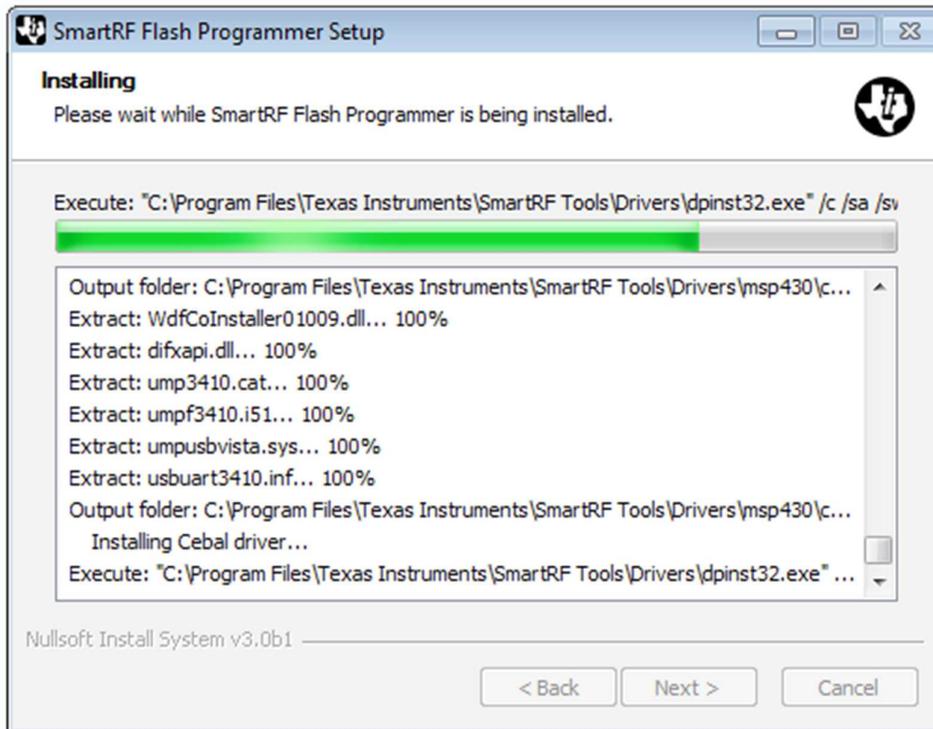


# Installation du firmware sur le sniffer CC2531

## Installer SmartRF Flash Programmer

logiciel SmartRF Flash



## Connecter le CC Debugger au Sniffer CC2531

Branchez le Sniffer au CC Debugger à l'aide d'un câble de liaison. Le Sniffer est équipé d'un connecteur au pas de 1mm. La carte GBAN sert juste à changer de format de câble. Il n'y a pas de détrompeur sur le CC2531. Utilisez la photo ci-dessous pour orienter correctement le câble (le détrompeur vers l'intérieur du Sniffer).

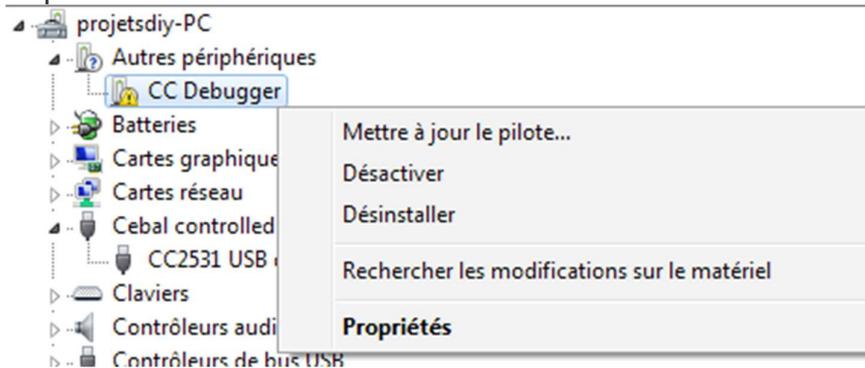


Branchez le CC Debugger et le Sniffer CC2531 sur l'ordinateur. La LED du Debugger va être rouge au branchement., Appuyez sur le bouton RESET pour la passer en vert. On est prêt pour flasher le firmware.

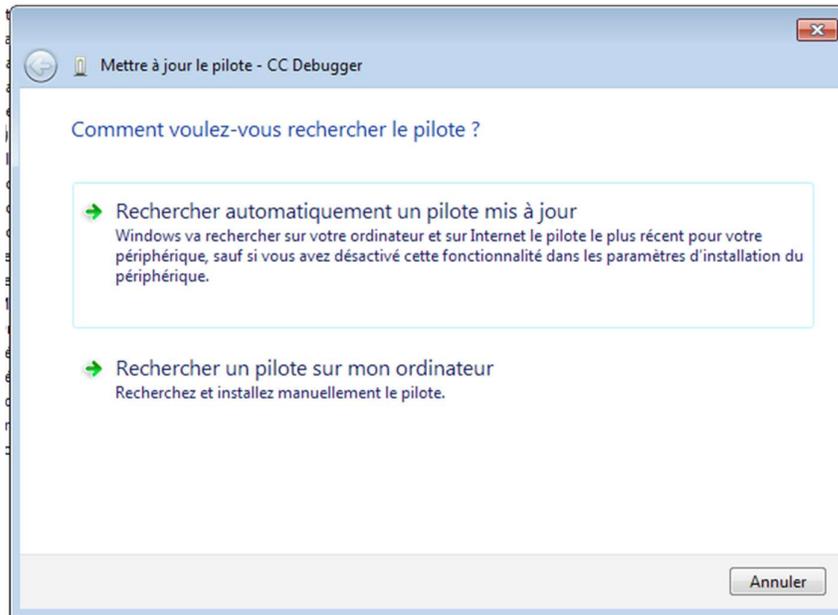


## Installer les drivers

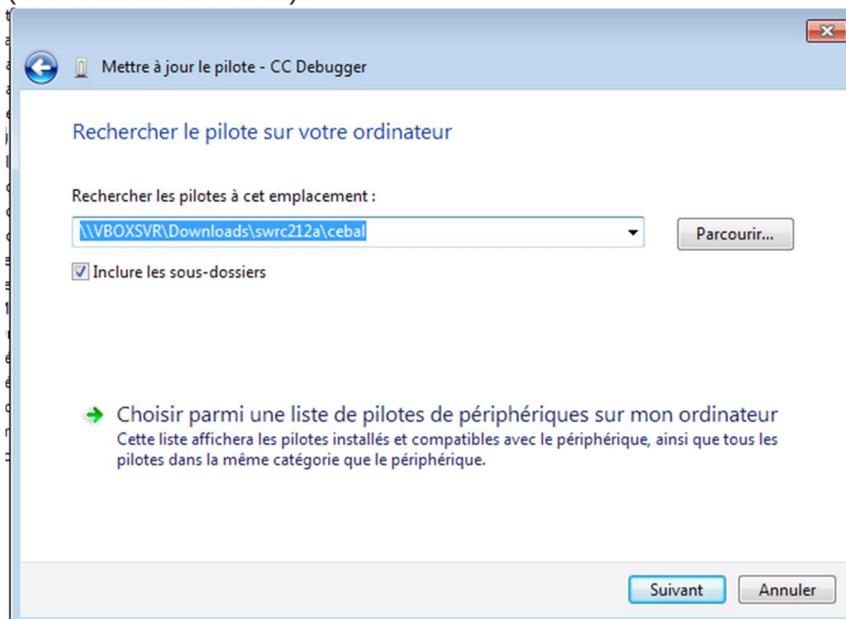
Télécharger le **driver Windows swrc212** du boitier CC Debugger ([ici](#)).  
un clic droit sur le CC Debugger puis de choisir l'option Mettre à jour le pilote...



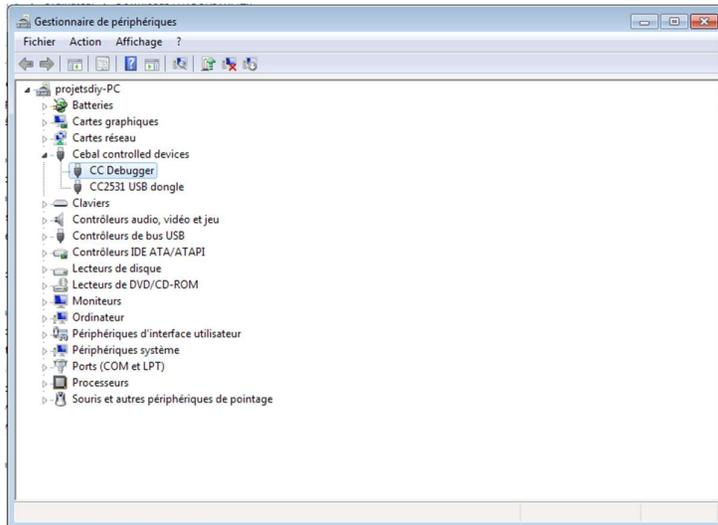
Dans la fenêtre qui s'ouvre, choisissez **Rechercher un pilote sur mon ordinateur**



Puis, **indiquez le chemin** vers le dossier swrc212 qui contient le drivers (32-bits et 64-bits).



Windows scan le répertoire à la recherche du drivers adapté à votre version. Validez le drivers trouvé pour terminer l'installation. A la fin, les deux appareils doivent être correctement installés.



## Télécharger le firmware

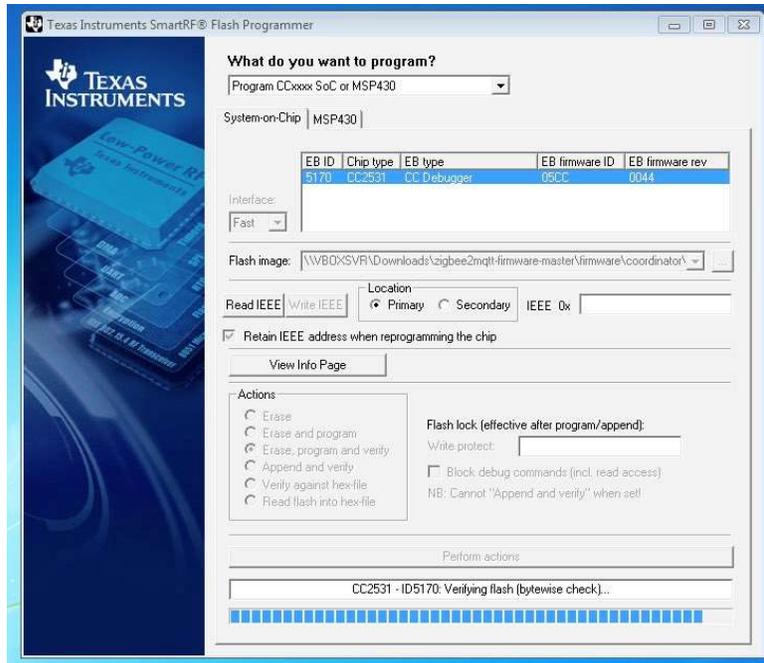
Le firmware Zigbee2MQTT est disponible sur GitHub sur [cette page](#).

Télécharger la dernière version (fichier hex)

Attention, le précédent lien <http://github.com/Koenkk/Zigbee2MQTT-firmware/tree/master/cebal/cc2531> (merci Nicolas)

## Flasher le firmware

Lancez le programme SmartRF Flash Programmer. Indiquez le chemin vers l'image du firmware (fichier hex précédemment téléchargé) puis cliquez sur Perform Action. L'installation dure moins d'une minute. Ne débranchez rien pendant l'opération.



## Installer la passerelle Zigbee2MQTT sur le Raspberry Pi

Branchez le sniffer USB sur le Raspberry Pi et ouvrez un Terminal (ou connectez vous en SSH). Exécutez cette commande pour trouver le port sur lequel se trouve le Sniffer.

```
ls -l /dev/tty*
```

En général, il est branché sur le port **ttyACM0**.

Vérifiez maintenant que Nodejs et npm sont installés sur Raspbian. Si vous utilisez Jeedom ou Node-RED, vous n'avez rien à installer.

```
node --version # v8.X.X mini
```

```
npm --version # v5.X.X mini
```

Si ce n'est pas le cas, exécutez ces deux commandes ou suivez [ce tutoriel](#) pour plus de détail.

```
sudo curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
```

```
sudo apt-get install -y nodejs git make g++ gcc
```

Maintenant, il ne reste plus qu'à installer Zigbee2MQTT

```
# Clone zigbee2mqtt
```

```
sudo git clone
https://github.com/Koenkk/zigbee2mqtt.git
/opt/zigbee2mqtt
sudo chown -R freebox /opt/zigbee2mqtt

( freebox ou le compte utiliser pour installation
jeedom )
```

```
# Installation des dépendances - Install
dependencies
cd /opt/zigbee2mqtt
npm install
```

La commande **npm install** reste en attente à la fin de l'installation.  
Vous pourrez la quitter manuellement à la lecture du message  
node-pre-gyp info ok

## Configurer la passerelle

Le fichier de configuration se trouve dans le dossier data.

```
nano /opt/zigbee2mqtt/data/configuration.yaml
Modifiez les paramètres MQTT :
```

- topic de base
- adresse IP du broker MQTT (server). Ne changez rien si le sniffer est branché sur le même Raspberry Pi
- user et password si le broker est protégé par un mot de passe (fortement conseillé, suivez [ce tutoriel](#) pour le configurer)

```
# MQTT settings
mqtt:
  # MQTT base topic for zigbee2mqtt MQTT messages
  base_topic: zigbee2mqtt
  # MQTT server URL
  server: 'mqtt://localhost'
  # MQTT server authentication, uncomment if
required:
  # user: my_user
  # password: my_password
```

Enregistrez avec CTRL+X puis O. Tout est prêt !

## Ajouter un accessoire Xiaomi Aqara ou Mijia

Lancez le script de la gateway

```
cd /opt/zigbee2mqtt  
npm start
```

Attendez d'avoir le message **Zigbee: allowing new devices to join**. Pour le test, je vous propose d'intégrer le détecteur de présence Xiaomi Mi Motion Sensor. La procédure est identique quelque soit l'accessoire.

Prenez un trombone et maintenez le micro-interrupteur durant 3 à 5 secondes jusqu'à ce que la LED blue clignote. Vous devriez obtenir directement l'identifiant unique de votre accessoires. Le script s'occupe du jumelage. Contrairement à la [gateway officielle](#), il faut empêcher l'accessoire de s'endormir en appuyant régulièrement (1 fois par seconde) sur le bouton d'association jusqu'à ce que le processus d'association débute.

Dès que le premier message arrive (détection de mouvement, mesure de température...), c'est terminé, votre accessoire Zigbee est maintenant associé.

```
2018-6-4 12:37:41 INFO Zigbee: allowing new devices  
to join.  
2018-6-4 12:37:48 WARN Message without device!  
spinlock: false []  
2018-6-4 12:38:04 INFO New device with address  
0x00158d000155d2b7 connected!  
2018-6-4 12:38:04 INFO MQTT publish, topic:  
'zigbee2mqtt/bridge/log', payload:  
'{"type":"device_connected","message":"0x00158d00015  
5d2b7"}'  
2018-6-4 12:38:08 INFO Connecting with device...  
2018-6-4 12:38:08 INFO MQTT publish, topic:  
'zigbee2mqtt/bridge/log', payload:  
'{"type":"pairing","message":"connecting with  
device"}'  
2018-6-4 12:38:13 INFO Connecting with device...
```

```
2018-6-4 12:38:40 INFO MQTT publish, topic:
'zigbee2mqtt/bridge/log', payload:
'{"type":"pairing","message":"device incoming"}'
2018-6-4 12:38:45 INFO MQTT publish, topic:
'zigbee2mqtt/0x00158d000155d2b7', payload:
'{"click":"double"}'
```

## Renommer les accessoires

Par défaut le script utilise l'identifiant unique de l'accessoire domotique comme nom. Ce n'est pas forcément le plus facile pour s'y retrouver. On peut facilement modifier le nom en allant le changer dans le fichier de configuration. Arrêtez le script et ouvrez de nouveau le fichier de configuration

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

En bas du fichier de configuration, vous allez trouver une nouvelle section nommée **devices**. Indiquez le nom de chaque accessoire avec la clé `friendly_name`. Evitez de mettre des espaces dans le nom des accessoires pour éviter les problèmes de Topic avec MQTT.

```
devices:
  '0x00158d000155d2b7':
    friendly_name: 'Aqara_Switch'
    retain: false
  '0x00158d000159a218':
    friendly_name: 'PIR'
    retain: false
  '0x00158d000149a352':
    friendly_name: 'Porte'
    retain: false
```

## Démarrer automatiquement le pont Zigbee2mqtt au démarrage de Raspbian avec systemctl

Jens Willmer, le développeur à l'origine du projet [vient de publier](#) le script nécessaire pour démarrer le pont au démarrage de Raspbian. En cas de coupure de courant ou de redémarrage du Raspberry Pi suite à une mise à jour, c'est plus pratique. Créez un nouveau fichier de configuration systemctl

```
sudo nano /etc/systemd/system/zigbee2mqtt.service
puis collez le code suivant
```

```
[Unit]
Description=zigbee2mqtt
After=network.target

[Service]
ExecStart=/usr/bin/npm start
WorkingDirectory=/opt/zigbee2mqtt
StandardOutput=inherit
StandardError=inherit
Restart=always
User=freebox

[Install]
WantedBy=multi-user.target
```

Pour démarrer le pont manuellement, exécutez la commande suivante

```
sudo systemctl start zigbee2mqtt
```

Ensuite, vérifiez que tout fonctionne comme prévu avec la commande `status`

```
sudo systemctl status zigbee2mqtt
```

Le service est correctement démarré

```
● zigbee2mqtt.service - zigbee2mqtt
   Loaded: loaded
          (/etc/systemd/system/zigbee2mqtt.service; disabled;
          vendor preset: enabled)
   Active: active (running) since Tue 2018-09-11
11:45:24 UTC; 7s ago
   Main PID: 18086 (npm)
   CGroup: /system.slice/zigbee2mqtt.service
           └─18086 npm
              └─18095 /usr/bin/node
                 /usr/lib/node_modules/npm/node_modules/update-
                 notifier/check.js
                 {"pkg":{"name":"npm","version":"5.6.0"}}
                    └─18109 sh -c node index.js
                       └─18110 node index.js
sept. 11 11:45:24 raspberrypi systemd[1]: Started
zigbee2mqtt.
sept. 11 11:45:29 raspberrypi npm[18086]: >
zigbee2mqtt@0.1.0 start /opt/zigbee2mqtt
sept. 11 11:45:29 raspberrypi npm[18086]: > node
index.js
```

On peut activer le démarrage automatique au prochain démarrage de Raspbian

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

### **D'autres commandes utiles**

- **sudo systemctl stop zigbee2mqtt**, pour arrêter le service. Mise à jour, renommer les devices...
- **sudo journalctl -u zigbee2mqtt.service -f**, permet d'afficher le journal d'exécution

### **Mise à jour de Zigbee2mqtt**

La mise à jour n'est pas ce qu'il y a de plus simple mais il suffit de suivre à la lettre les commandes indiquées à Jens

```
# On arrête le service et on se place dans le
répertoire d'installation du projet
sudo systemctl stop zigbee2mqtt
cd /opt/zigbee2mqtt
```

```
# Backup de la configuration
cp -R data data-backup
```

```
# Update
git checkout HEAD -- npm-shrinkwrap.json
git pull
rm -rf node_modules
npm install
```

```
# On restaure la configuration
cp -R data-backup/* data
rm -rf data-backup
```

```
# On re-démarre le service zigbee2mqtt
sudo systemctl start zigbee2mqtt
```