|  |  |  |  |
| --- | --- | --- | --- |
| logo2014 | STS SN | Module IOT | http://iotcenter.dk/wp-content/uploads/2016/09/iot_logo.png |
| Activité 2 : Découverte du broker Mosquitto | |

**Matériel nécessaire :**

* RPI 3 connectée à internet (distribution : “Raspbian Buster with desktop and recommended software” 2020-02-13)
* PC équipé de putty, wireshark et MQTT-FX.

# Installation de Mosquitto sur RaspBerry

Avant d’installer Mosquitto, mettre à jour votre système :

***sudo apt update && sudo apt upgrade -y***

Installer le broker Mosquitto :

***sudo apt-get install mosquitto***

Installer le client mosquitto pour effectuer des tests :

***sudo apt-get install mosquitto-clients***

Le client mosquitto permet de souscrire à des topics (***mosquitto\_sub***), de publier des messages (***mosquitto\_pub***) et de configurer les utilisateurs (***mosquitto\_passw***).

# Tester le broker mosquitto

Ouvrir deux terminales.

Dans le premier, souscrire au topic « sensor/temperature » :

***mosquitto\_sub -h localhost -t "g133/temperature"***

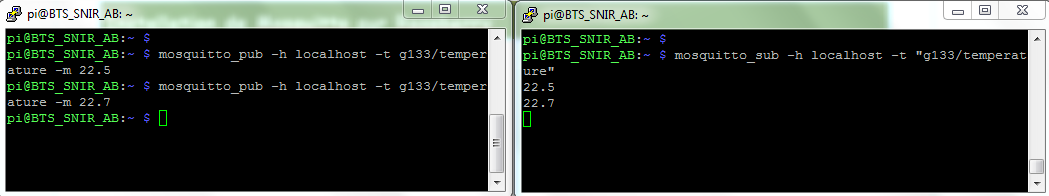
Se connecte au serveur (-h) localhost

Sélectionner le topic (-t) g133/temperature

Dans le deuxième terminale, publier une valeur (-m) 22.5 sur le même topic « sensor/temperature » :

***mosquitto\_pub -h localhost -t g133/temperature -m 22.5***

Le résultat est le suivant :



Terminale correspondant au « Publisher » Terminale correspondant au « Subscriber »

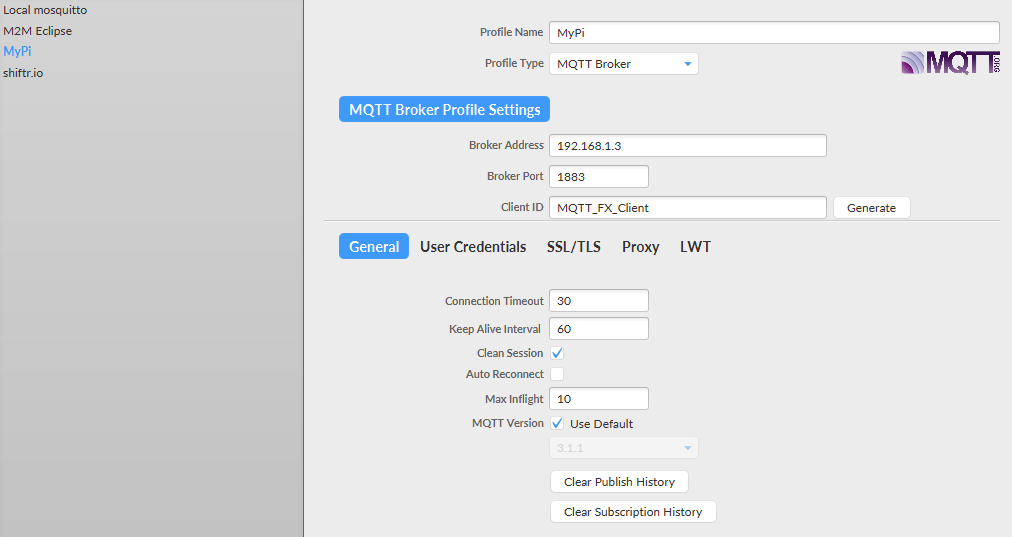
# Tester la qualité de service (QoS) avec MQTT

Par défaut un message est envoyé avec un **QoS niveau 0** « At most once » (il sera délivré au plus une fois).

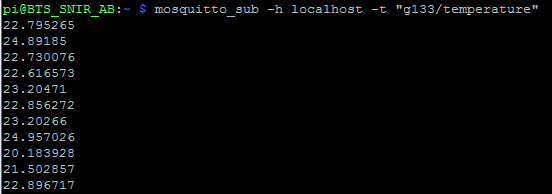
Nous allons analyser avec Wireshark les échanges entre le broker Mosquitto installé sur la Rasp et un publisher installé sur un PC (logiciel MQTT-FX). Mais pour lancer une capture, il faut d’abord déterminer un filtre à appliquer (protocole et port utilisé)

## Configurer les échanges MQTT entre PC et RaspBerry

Lancer sur votre PC le logiciel MQTT-FX et créer un nouveau profil de connexion afin de cibler le broker mosquitto de votre RaspBerry.



Publier sur le topic « g133/temperature » après avoir lancé un client mosquitto sur votre RaspBerry qui souscrit au même topic.

Vous devez obtenir le résultat suivant :

## Lancer la capture avec WireShark

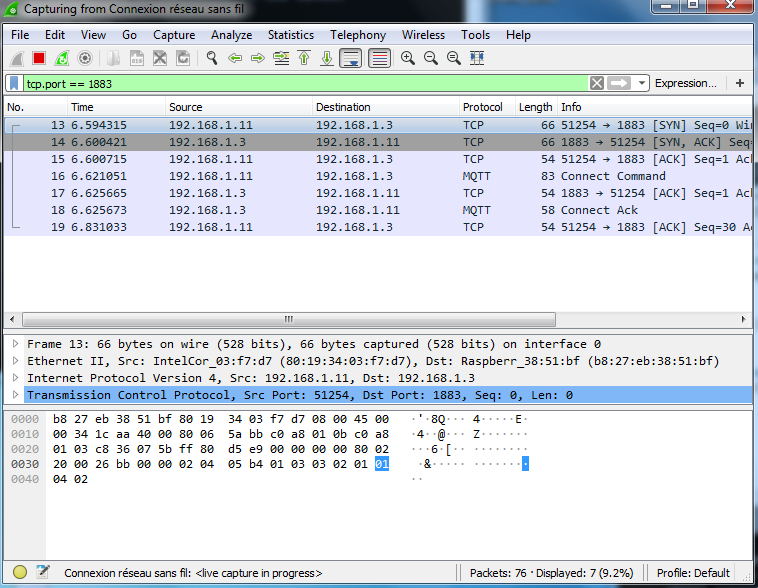
Afin de filtrer les données sous WireShark, il faut déterminer le protocole et le port utiliser par le broker mosquitto.

Pour cela, vous devez vous mettre en root (***sudo su***) et utiliser la commande ***netstat –taupen***

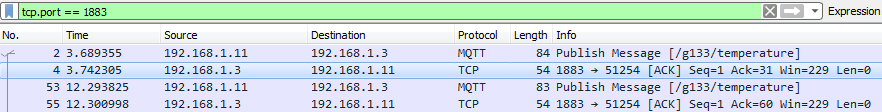
Déconnecter MQTT-FX et lancer wireshark sur le pc (pubisher) et ajouter un filtre : ***tcp.port==1883***

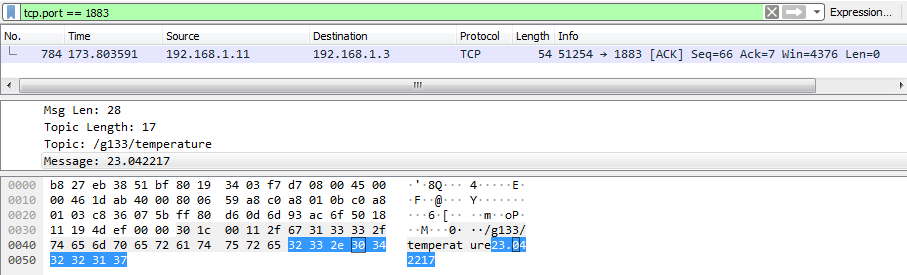
Relancer le MQTT-FX et capturer les échanges lors de la connection au broker.

Vous devez obtenir le résultat suivant :



Refaire une nouvelle capture lorsque le publisher publie une température, vous devez obtenir le résultat suivant :

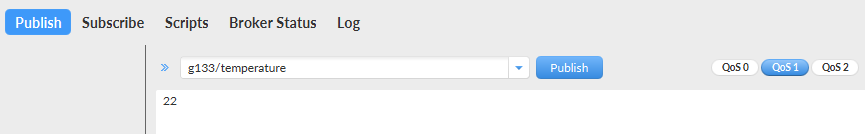




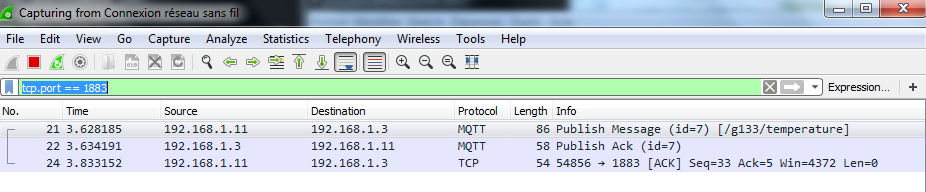
Par défaut, le QoS du logiciel MQTT-FX est de niveau 0. On constate qu’avec se niveau de qualité de service, les informations sont simplement transmises et validées uniquement par TCP.

## Modifier le QoS

Pour modifier le QoS du publisher afin d’avoir un QoS de niveau 1

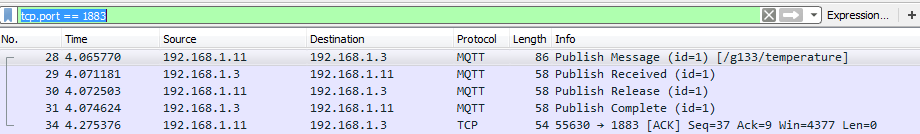


Relancer une capture wireshark, vous devez obtenir le résultat suivant :

On constate que :

* Le broker génère cette fois un accusé de réception MQTT. En effet, avec un QoS de niveau 1, les messages sont acquittés par le receveur, si cet acquittement n’arrive pas le message est retransmis.

Modifier le QoS au niveau 2 et refaire la manipulation. Vous devez obtenir le résultat suivant :



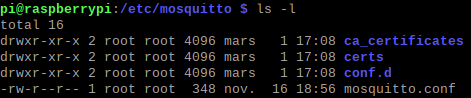
On constate que :

* Le broker génère cette fois un accusé de réception et confirme que le message à bien été ensuite effacé. En effet, avec un QoS de niveau 2, l’acquittement est lui même validé afin de garantir que le message n’arrive qu’une seule fois.

# Sécuriser le protocole MQTT (au niveau des accès)

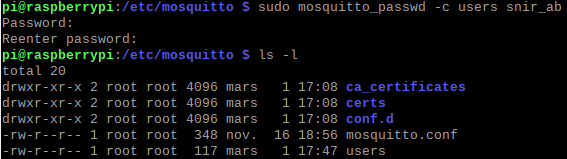
Par défaut, n’importe quel client peut se connecter au broker Mosquitto. Ce n’est pas très gênant s’il n’y a que des informations basiques qui sont publiées (mesure de température, d’humidité, …), mais si vous voulez commander des appareils, il faut sécuriser un minimum le broker.

Placez-vous dans le dossier de configuration de Mosquitto : ***cd /etc/mosquitto/***



Mosquitto crée un fichier qui contient le nom de l’utilisateur et son mot de passe (crypté). Pour ajouter un nouvel utilisateur exécutez la commande ***mosquitto\_passwd -c nomfichier nomutilisateur***. Remplacez nomfichier par le nom de fichier qui contiendra les identifiants et nomutilisateur, le nom de l’utilisateur à ajouter : ***sudo mosquitto\_passwd -c users snir\_ab***

Mosquitto demande la saisie du mot de passe (2 fois).

Un nouveau fichier contant les utilisateurs est créé :

Ouvrez le fichier de configuration de Mosquitto avec nano (sudo obligatoire sur Linux) :***sudo nano mosquitto.conf***

Ajoutez à la fin la commande allow\_anonymous false et indiquez le chemin vers le fichier des mots de passe :

***allow\_anonymous false***

***password\_file /etc/mosquitto/users***

Maintenant il ne reste plus qu’à redémarrer Mosquitto pour prend en compte les modifications : ***sudo service mosquitto restart***

Utilisez la commande Linux cat pour vérifier que le nouvel utilisateur à bien été ajouté au fichier de configuration :



Maintenant, il faudra s’authentifier pour pouvoir publier ou souscrire à un Topic. En ligne de commande, on dispose de l’option -u pour spécifier l’utilisateur et -P pour le mot de passe. Ce qui donne pour publier un message : ***mosquitto\_pub -h localhost -t test -m demo -u nomutilisateur -P motdepasse***

Et pour souscrire : ***mosquitto\_sub -h localhost -t "test" -v -u nomutilisateur -P motdepasse***

Au niveau du logiciel MQTT-FX, il faut modifier les paramètres de connexions :

