Introduction a MQTT

1) MQTT, qu'est-ce que c'est ?
2) Définition d'un topic
3) Publier / souscrire à un topic
4) Qualité de service
5) La dernière valeur d'un topic peut être conservée.
6) Un petit coup d'oeil sur la sécurité des données
7) Et les Arduinos dans tout cela ?
8) Une application pour l'exemple: Owntracks

Thierry Vorms février 2017
vormsty@gmail.com

HB9FXO

1) MQTT qu’est ce que c’est ?

MQTT (Message Queuing Telemetry Transport) est un protocole de transport d’informations open source développé par IBM

- Ce protocole décrit un broker, c’est-à-dire le fonctionnement d’une poste… On envoie des paquets et qq les reçoit
- Des capteurs envoient des messages au broker.
- Des abonnés reçoivent des notifications du broker pour chaque nouvelle donnée publiée.
- Ce protocole est conçu pour être léger, donc implémenté sur de petites machines.
- L’implémentation la plus connue sur Linux s’appelle Mosquito.

2) Définition d’un topic

Un topic correspond à l’identification de la donnée.

Exemple: maison/1er étage/température

Il existe des wilcards

Exemple: maison/+/température

→ capteurs de température de tous les étages

Autre exemple : maison/1er étage/#

→ tous les capteurs du 1er étage (humidité, température, etc.)

3)Publier / souscrire a un topic

- Publier ou souscrire à un topic est extrêmement simple.
- Il faut se connecter au serveur et publier ou souscrire.
- Plusieurs clients peuvent s’abonner au même topic.
- Un client peut s’abonner à plusieurs topics.
- Pour chaque langage courant (PHP, Python, Java, Java script, C++), il existe une librairie adaptée.
- On peut transmettre un message jusqu’a
- à 260 Mb pour l’implémentation Mosquitto, mais que du texte...(UTF-8)

4) Qualité de service

Il existe 3 possibilités pour la qualité de service:

- QoS = 0 Le message est distribué une fois tout au plus, ou n'est pas distribué du tout. Sa distribution via le réseau n'est pas accompagnée d'un accusé de réception.Le message n'est pas stocké. Le message peut être perdu si le client est déconnecté ou si le serveur échoue.
- Qos = 1 Le message est toujours distribué au moins une fois. Si l'expéditeur ne reçoit pas d'accusé de réception, le message est renvoyé avec l'indicateur DUP défini jusqu'à la réception de l'accusé de réception. En conséquence, le destinataire peut recevoir le même message à plusieurs reprises, et le traiter plusieurs fois.
- QoS = 2 Le message doit être stocké localement au niveau de l'expéditeur et du destinataire jusqu'à ce qu'il soit traité.
- Il faut au moins deux paires de transmissions entre l'expéditeur et le destinataire avant la suppression du message côté expéditeur. Le message peut être traité côté destinataire après la première transmission.

  Dans la première paire de transmissions, l'expéditeur transmet le message et reçoit un accusé de réception du destinataire indiquant qu'il a stocké le message. Si l'expéditeur ne reçoit pas d'accusé de réception, le message est renvoyé avec l'indicateur DUP défini jusqu'à la réception de l'accusé de réception.

  Dans le cadre de la seconde paire de transmissions, l'expéditeur dit au destinataire qu'il peut terminer le traitement du message, "PUBREL". Si l'expéditeur ne reçoit pas d'accusé de réception du message "PUBREL", le message "PUBREL" est renvoyé jusqu'à la réception de l'accusé de réception. L'expéditeur supprime le message qu'il a enregistré lors de la réception de l'accusé de réception au message "PUBREL"

5) La dernière valeur d’un topic peut être conservée.

Supposons qu’un client publie une donnée et qu’un autre client souscrit a cette même donnée mais plus tard.

- Si le flag retained est à true, le client va recevoir la dernière valeur vue par le broker.
- Si le flag retained est à false, le client ne va rien recevoir.

6) Un petit coup d’œil sur la sécurité des données

De base, l’implémentation du protocole propose une liaison par socket et par web socket.

- Concernant la sécurité, il est nécessaire de pouvoir effectuer une validation mutuelle. C’est a dire que contrairement à la situation normale client serveur, ici, le serveur doit être sur que pas n’importe qui publie n’importe quel donnée. Mosquitto propose la validation par certificat et le cryptage des données (TLS).
- Il est aussi possible de restreindre l’accès aux données par une acl (access list user password).

7) Et les Arduinos dans tout cela ?

- Comme évoqué, il existe la librairie C++ qu'il faut simplement installer.
- Concernant la sécurité, je pense qu'avec un Arduino Uno il sera difficile d'aller plus loin.
- Avec un microcontrôleur un peut plus puissant, il est possible de stocker une emprunte de clef. (Je n'ai pas encore expérimenté, mais j'ai vu qu'il est possible de transmettre des fichiers directement dans la mémoire flash du micro contrôleur.

  Concernant l'ESP 8266, il y a un stack wifi secure qui offre la possibilité de faire du TLS).

8) Une application: Owntrack

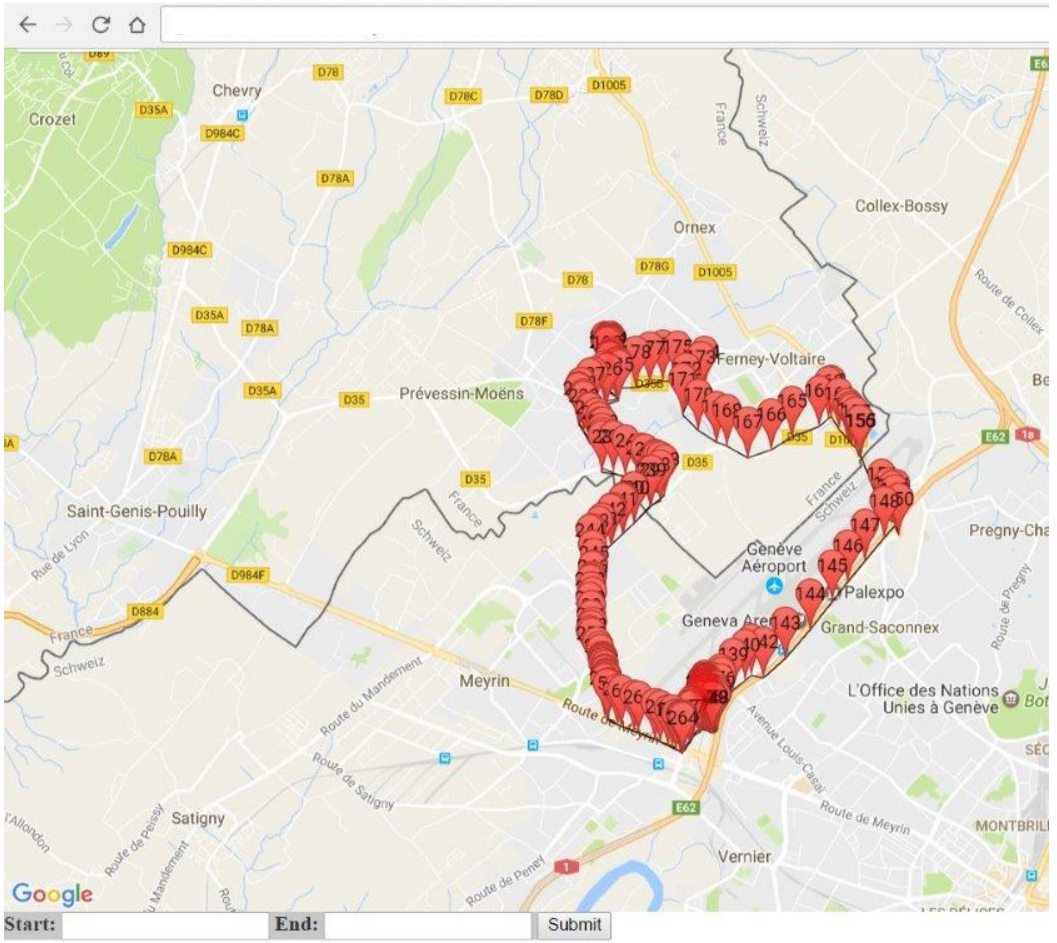
Owntrack est une application a installer sur un smartphone Android ou IOS.

- Cette application publie la position GPS a un broker préalablement paramétré.
- Un petit module de persistance, un peu de python, php, jquery, sql, javascript et beaucoup de patience… on arrive à cela (voir page suivante).

Merci pour votre attention, pour toute information complémentaire, mon adresse mail est :

vormsty@gmail.com

My own tracks



Pour toute information complémentaire, je me ferais une joie de vous répondre.

- N’hésitez pas à communiquer avec moi : vormsty@gmail.com