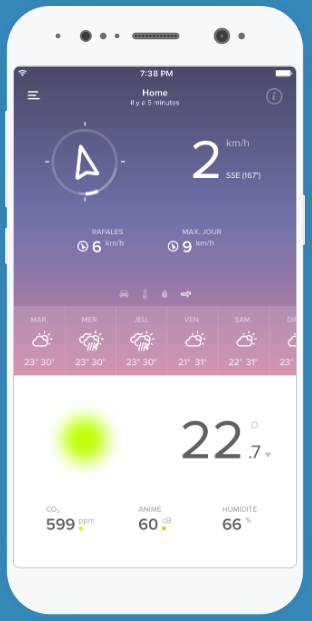
Découverte de l’Internet des objets (IOT)

# Objets connectés

Exemple d’objet connecté.



<https://www.netatmo.com/fr-fr/weather>

Visualise la vidéo suivante

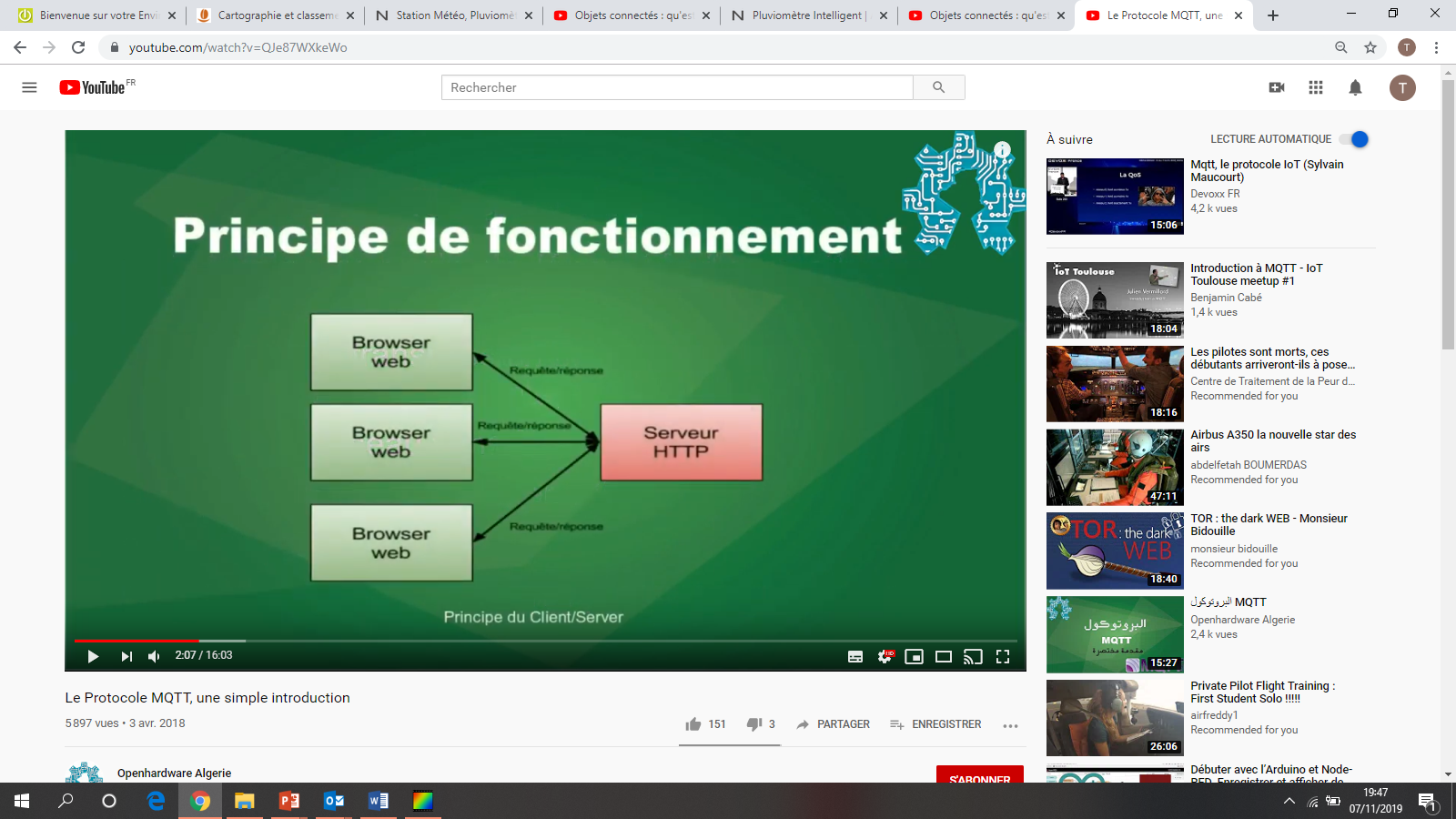


<https://www.youtube.com/watch?time_continue=16&v=Ii49ynk6K5I>

Qu’est-ce qu’un objet connecté ?

De quoi a besoin un objet pour se connecter ?

Tu as vu que pour échanger des données sur le web on pouvait utiliser le protocole http qui utilise le principe d’échange client-serveur.

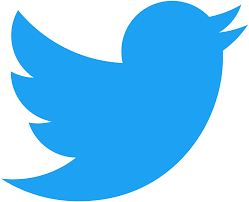


Pour que les objets puissent échanger des données, on n’utilise pas ce principe qui n’est pas très pratique pour les échanges, mais une autre.

# Principe de fonctionnement des échanges

Imagine que tu souhaites t’abonner à un compte Twitter d’une personne que tu souhaites suivre. Mais tu ne souhaites consulter sans arrêter son compte pour vérifier s’il a fait une nouvelle publication. Tu souhaiterais alors recevoir un simple mail qui t’indiquerait que la personne a publié un nouveau twitte que tu pourras alors lire tranquillement.





1. souscription

2. publication



3. lecture

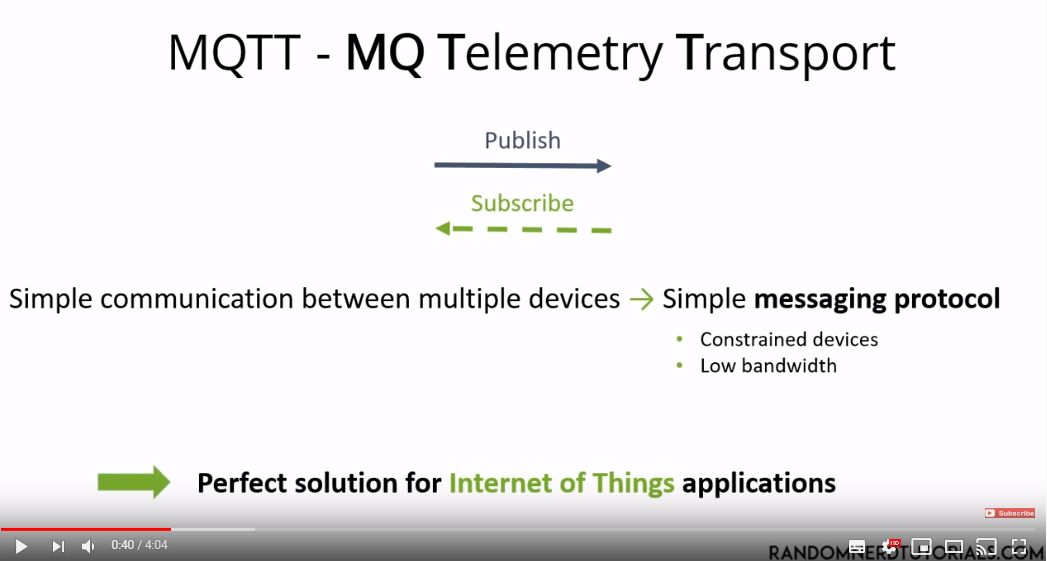
Internaute lecteur de tweet

Auteur de Tweet

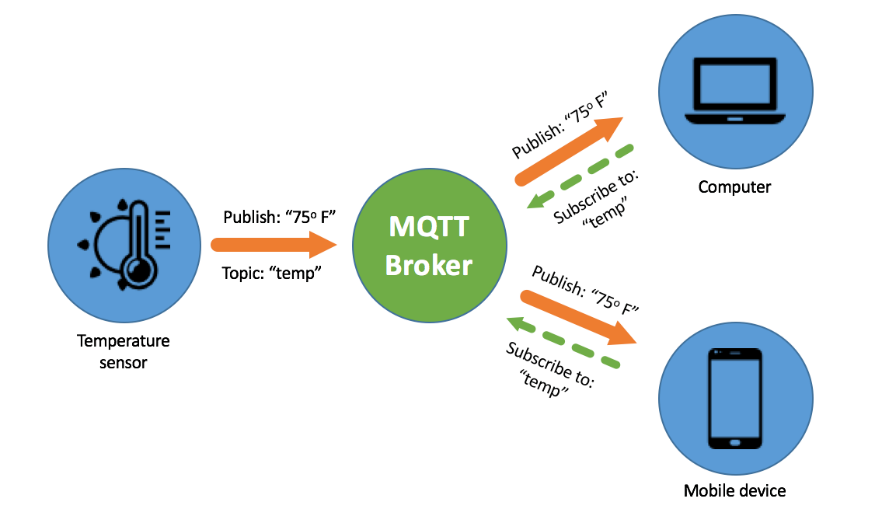
Serveur de Twitter

# Détails d’un protocole d’échange pour l’IOT

Visualise la vidéo suivante :



<https://www.youtube.com/watch?time_continue=231&v=EIxdz-2rhLs>

[](https://devblog.axway.com/wp-content/uploads/MQTT_1.png)Quel est alors un exemple de protocole utilisé pour l’IOT ?

***Principe de la publication et de la souscription***

1. Souscription (Subscribe).

Un client s’inscrit à une ressource sur un broker (comme un serveur) qui va héberger et centraliser les données des objets connectés

Cette resource est localisée dans une arborescence appelée Topic

Topic = arborescence, organisation des données.

1. Publication (Publish)

Un client peut publier (Publish) une donnée, ou un message (appelé Payload), comme une donnée capteur.

Ce protocole s’appelle le protocole MQTT pour MQ Telemetry Transport

Il est possible de gérer la qualité de service (de transport des données)

niveau 0 = livré au mieux une fois

niveau 1 = livré au moins une foix

niveau 2 = livré exactement une fois

Le protocole MQTT utilise aussi le protocole TCP pour acheminer les données entre clients et broker.

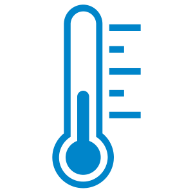
# Essai de publication et souscription

Tu vas utiliser le broker Hivemq disponible gratuitement sur Internet.

Chaque binôme va simuler un capteur. Il faudra alors publier une donnée régulièrement sur le réseau à l’aide d’un publish

Chaque binôme va souscrire à un capteur à l’aide d’un subscribe.





1

2

Capteur

(client)

3

Consultation des données capteurs (client)

Broker (serveur)

1 → un client souscrit à un topic (une donnée d’un objet connecté)

2 → un autre client (objet connecté) publie une donnée (comme une valeur d’un capteur) sur un topic (organisation de gestion des données (payload)).

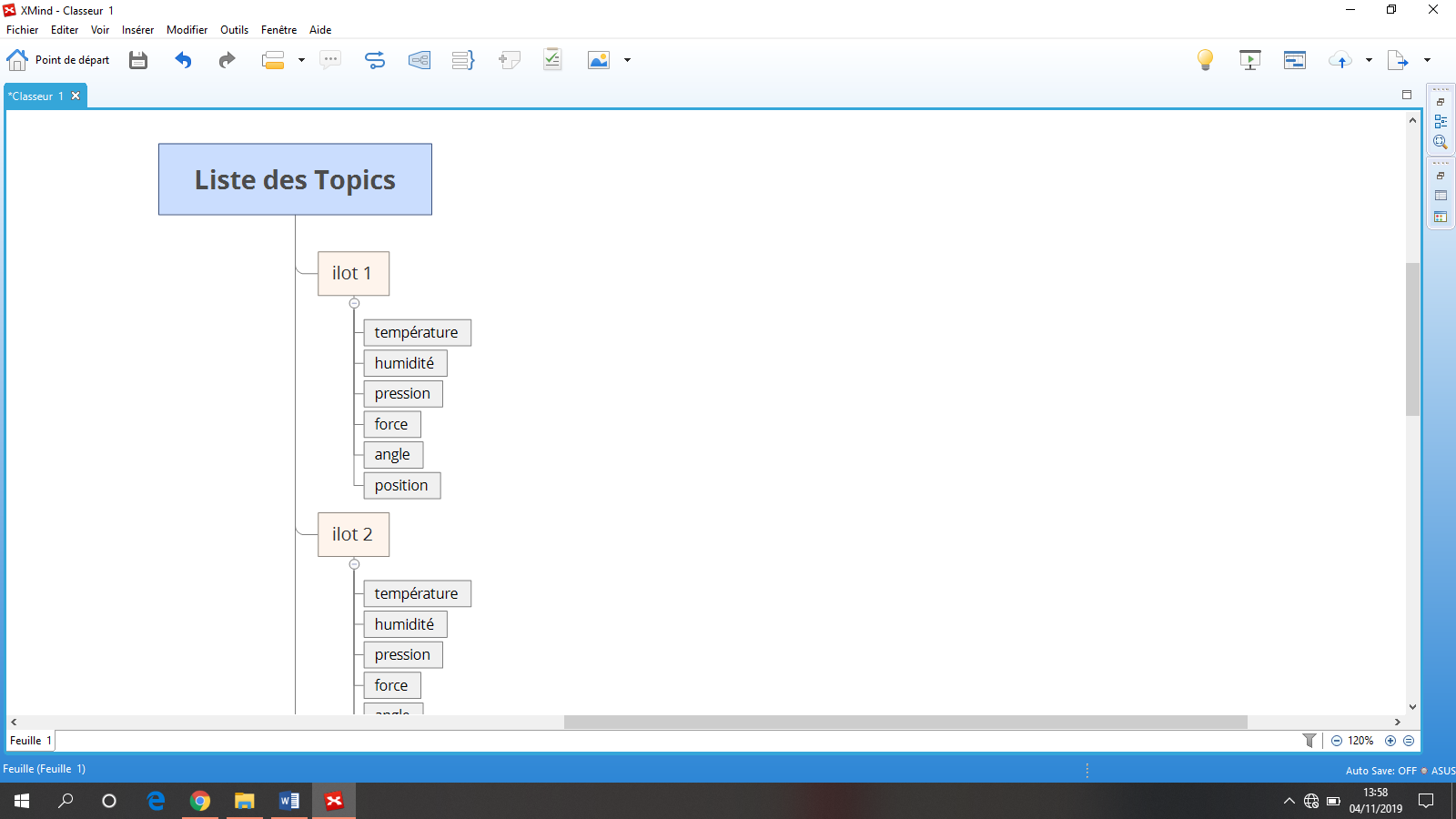
3 → le broker publie alors la donnée au client, qui s’est inscrit au topic.

Organisation dans la classe : 18 binômes

|  |  |  |
| --- | --- | --- |
| **Binôme** | **Capteur simulé (publish)** | **Souscription à un capteur (subscribe)** |
| N°1 | Température °C | Force (N) |
| N°2 | Humidité (%) | Angle (degrés) |
| N°3 | Pression (hPa) | Position (m) |
| N°4 | Force (N) | Température °C |
| N°5 | Angle (degrés) | Humidité (%) |
| N°6 | Position (m) | Pression (hPa) |
| N°7 | Température °C | Force (N) |
| N°8 | Humidité (%) | Angle (degrés) |
| N°9 | Pression (hPa) | Position (m) |
| N°10 | Force (N) | Température °C |
| N°11 | Angle (degrés) | Humidité (%) |
| N°12 | Position (m) | Pression (hPa) |
| N°13 | Température °C | Force (N) |
| N°14 | Humidité (%) | Angle (degrés) |
| N°15 | Pression (hPa) | Position (m) |
| N°16 | Force (N) | Température °C |
| N°17 | Angle (degrés) | Humidité (%) |
| N°18 | Position (m) | Pression (hPa) |

Pour publier une donnée (payload), il faut définir un topic (arborescence d’accès à la donnée)

Exemple de liste de topics.



1. **Souscription à un topic**

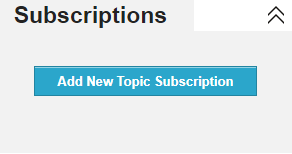
Tu vas utiliser un client MQTT pour souscrire à une donnée capteur (topic) imposée dans le tableau précédent.

Q1. Quel ce topic (subscribe) auquel tu dois t’inscrire ?

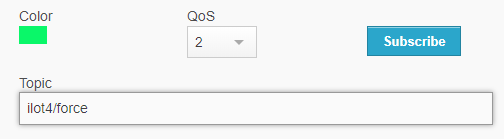
Connecte-toi sur le site d’un client MQTT comme celui-ci :

<http://www.hivemq.com/demos/websocket-client/>

Clique sur le bouton Add New Topic Subscription le nom de ce topic et clique

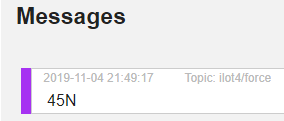


Puis tape le nom du topic sur lequel tu souhaites t’inscrire et clique sur subscribe



Tu es maintenant abonné aux données du capteur. Patiente que le capteur t’envoie des données.

Tu visualises alors les messages qui arrivent ici :

 ici le capteur indique une force de 45 Newton par exemple.

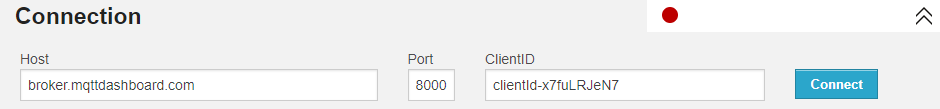
1. **Publication d’une donnée capteur**

Tu vas utiliser un client MQTT pour publier une donnée capteur.

Pour cela connecte toi sur une nouvelle fenêtre de navigation sur le même site. Ne surtout pas fermer l’autre fenêtre du navigateur où tu visualises ! :

<http://www.hivemq.com/demos/websocket-client/>

Tu te connectes sur un broker en cliquant sur connect



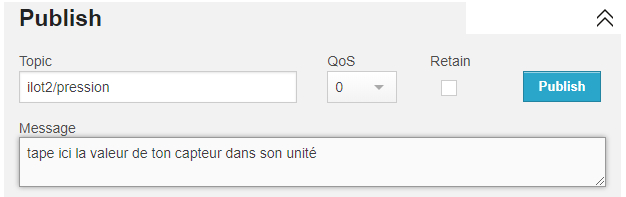
Q1. Quel est le site web de ce broker ? Que remarques-tu ?

Puis saisies la valeur du topic sous forme d’arborescence (pas d’accent, pas de majuscule, par d’espace, de caractère spécial)

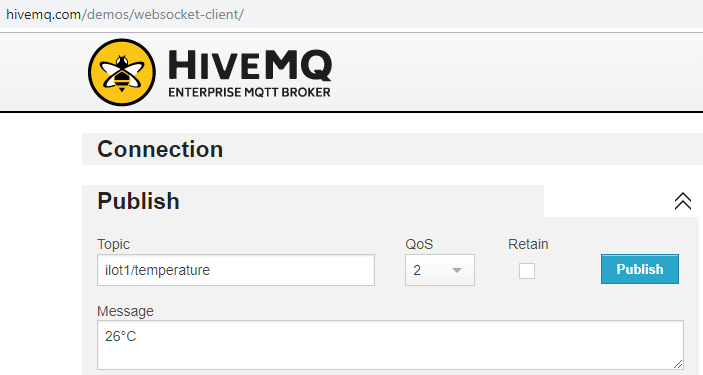
Q2. Quel est ton topic ?

Pour cela regarde le tableau précédent.

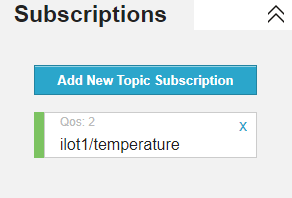
Saisies ton topic, puis saisie la valeur de ton capteur à inventer, et clique sur publish

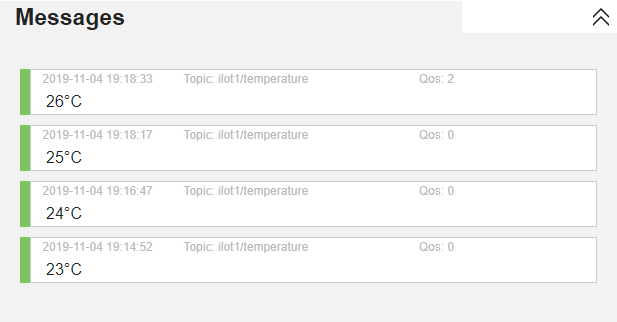


Exemple de simulation d’un objet connecté qui publie une donnée température sur un broker donné.

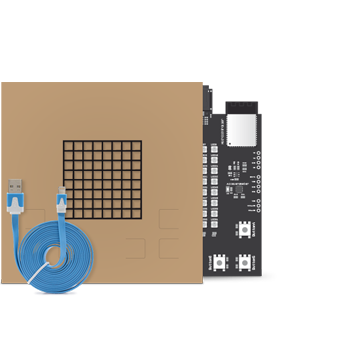


Un autre client se connecte au même broker, et au même topic. Il visualise alors la donnée (payload) de l’objet connecté précédent ?



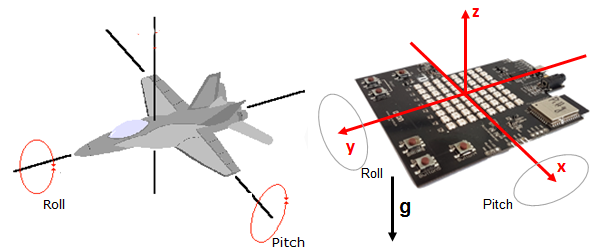


# Publication à l’aide de l’Oxocard

Tu vas utiliser une carte mini-ordinateur autonome qui permet de simuler un objet connecté réel.

Dans un premier temps, tu vas publier une donnée capteur régulièrement, toutes les secondes, par exemple.

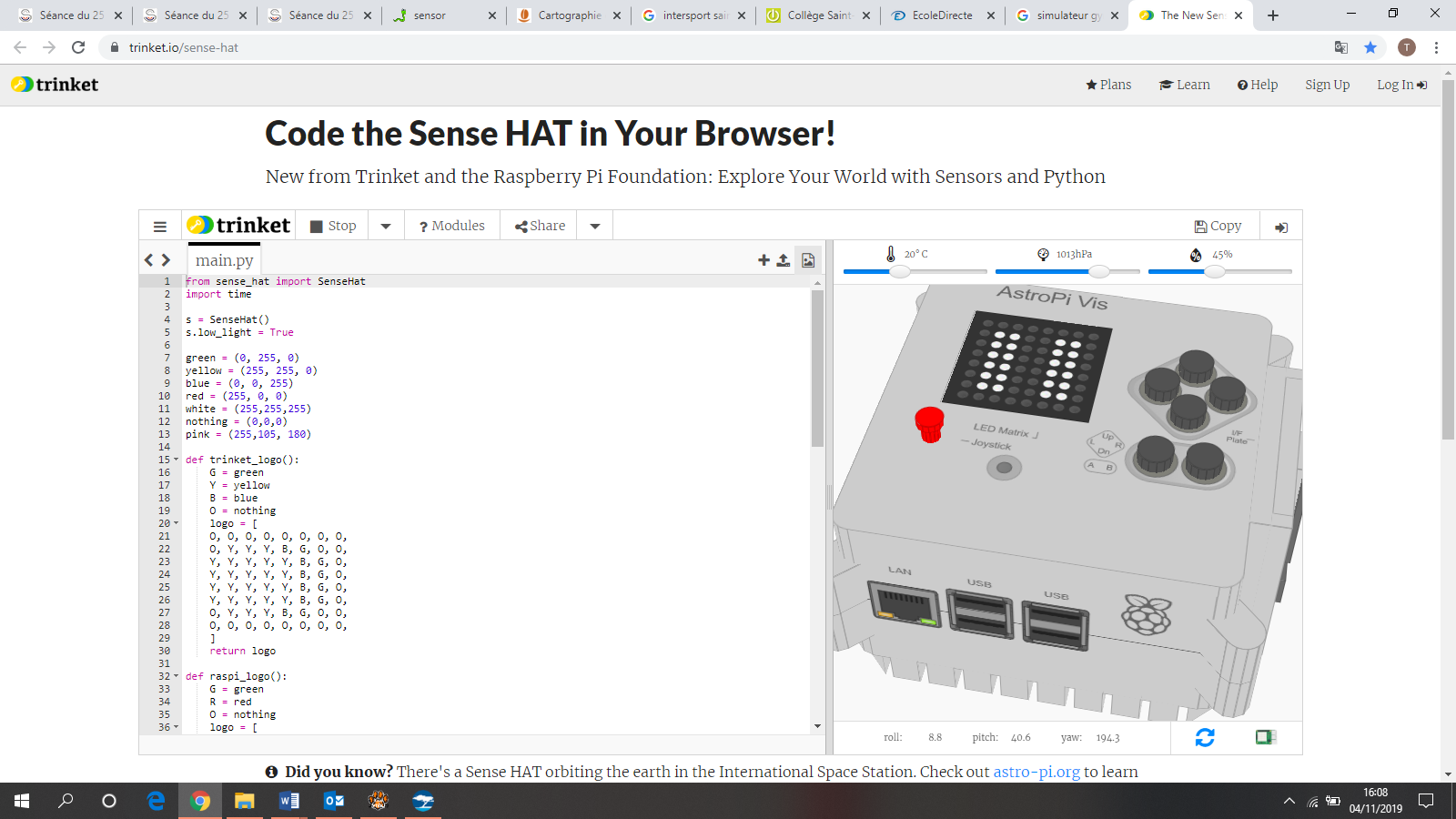
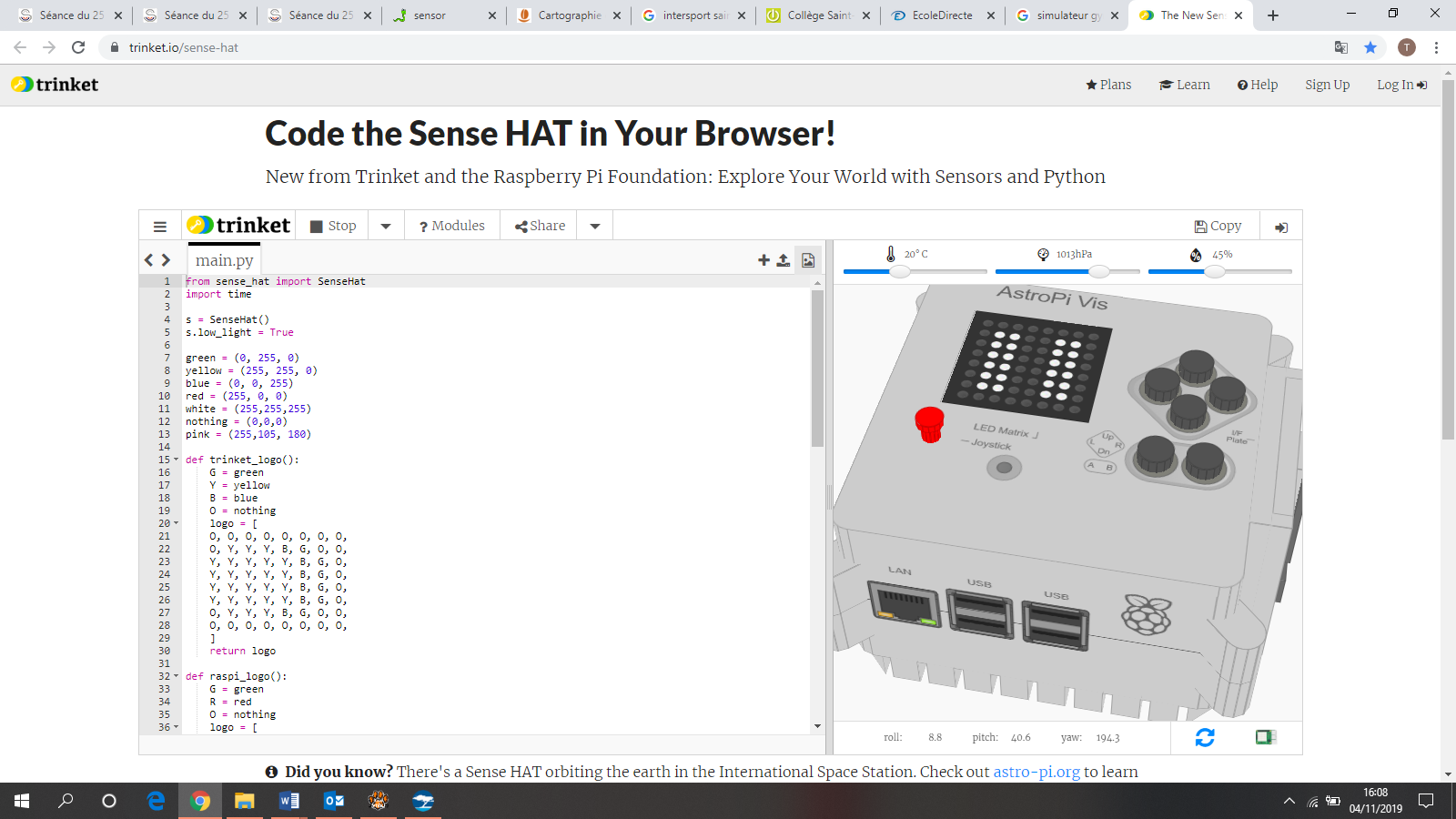
Ici, on souhaite donner la position de la carte oxocard, pour connaitre sa position, par rapport au plan de l’horizontal.

[](https://www.tigerjython4kids.ch/robotik/oxocard/bilder/pitch-roll.png)

Pour cela on va afficher l’angle de tangage (pitch).

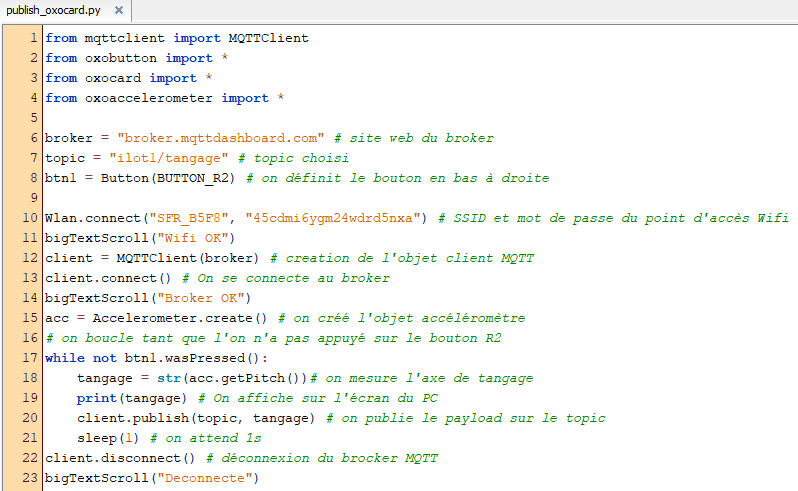
Fais un essai sur le simulateur suivant. Maintient clic gauche sur la boite et fait la pivoter. Regarde les informations en dessous.

<https://trinket.io/sense-hat>



Lance l’IDE 

Ouvre le programme publish.py disponible sur le serveur.



Connecte-toi sur un client MQTT HiveMQ

Fait une souscription au topic que tu as défini sur le même broker.

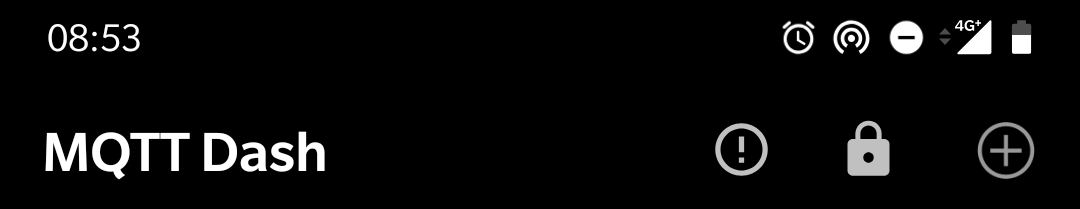
Puis visualise les données de tangage qui doivent évoluer lorsque tu bouges ta carte Oxocard suivant l’axe de tangage.

# Affichage sur smartphone (IHM Dashboard)

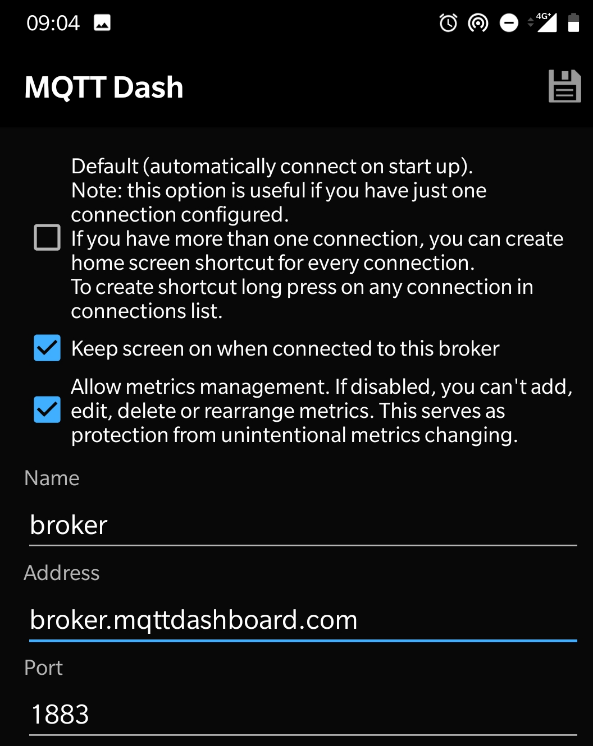
Télécharger sur votre smartphone l’application MQTT Dashboard

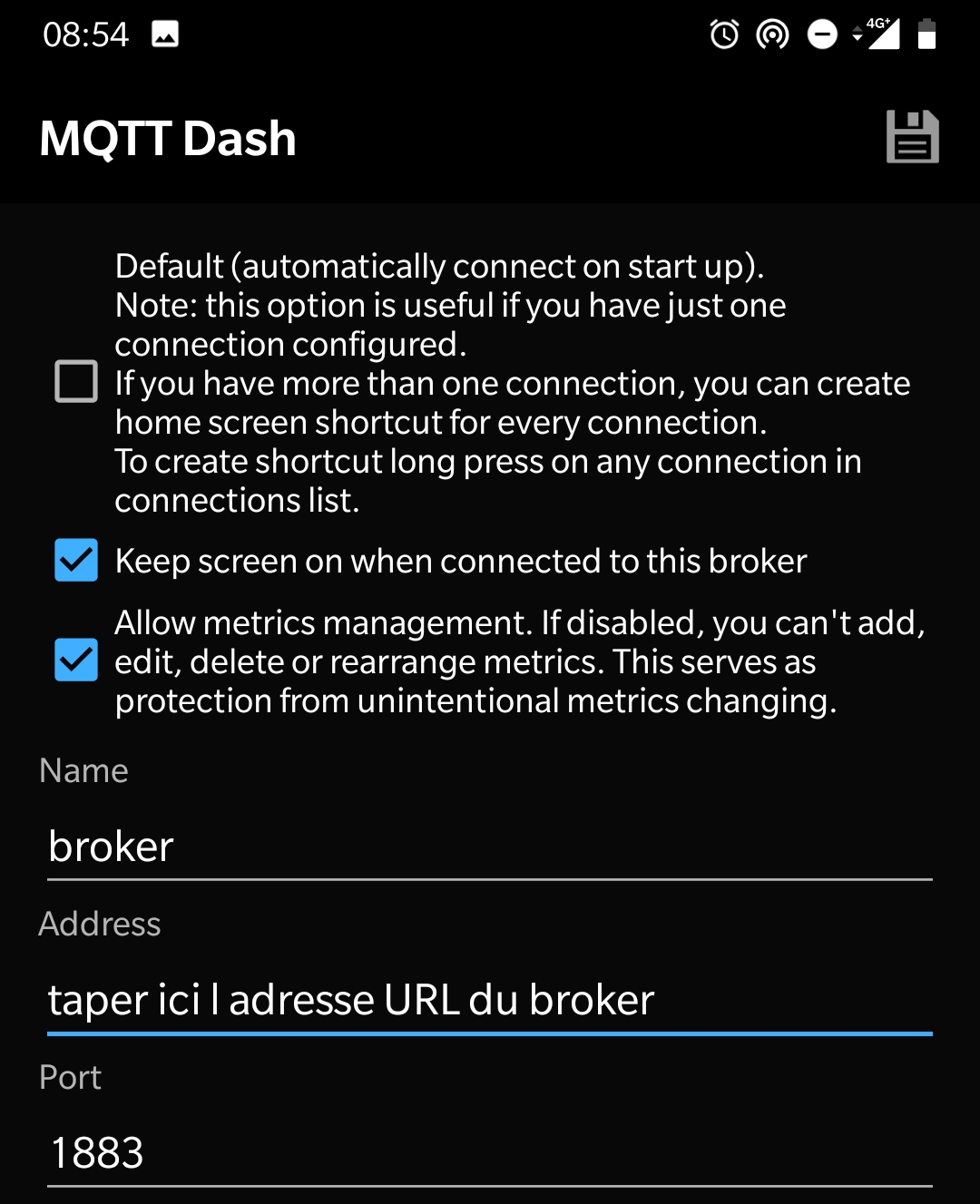
<https://play.google.com/store/apps/details?id=net.routix.mqttdash&hl=fr>

clique sur

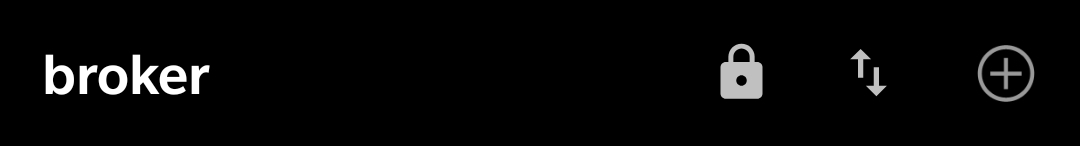


Tape le nom du broker

Saisie l’URL du broker



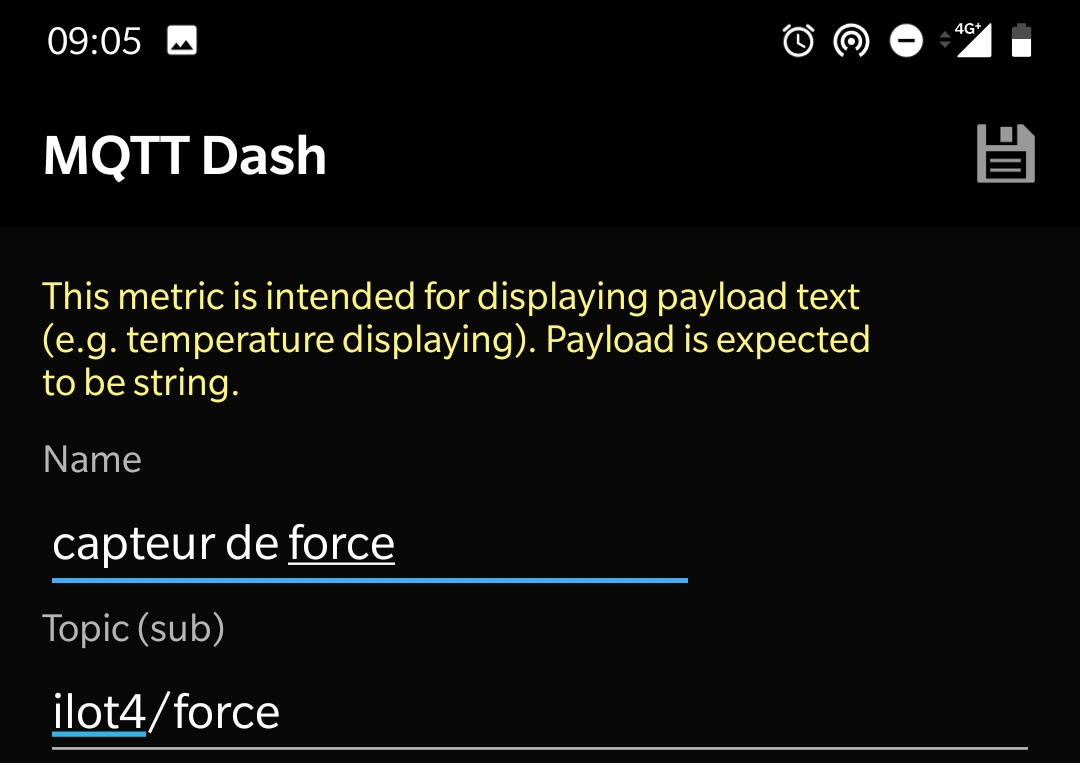
Puis tu as le broker et clique sur.



Clique sur

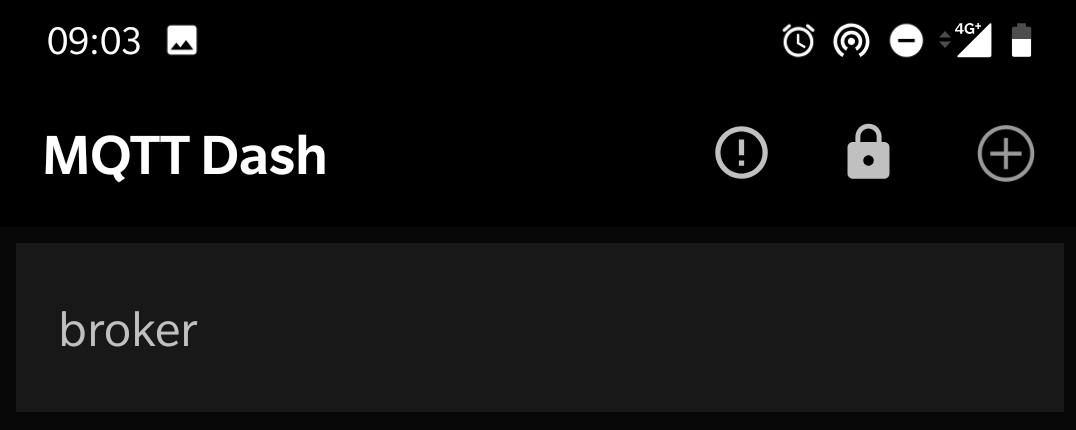


Saisie le nom du capteur à visualiser



Saisir le topic

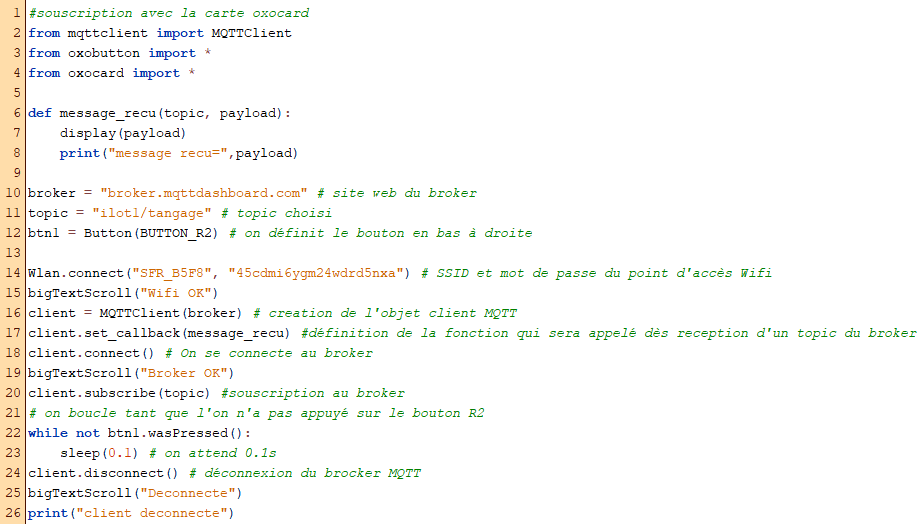
Puis clique sur broker

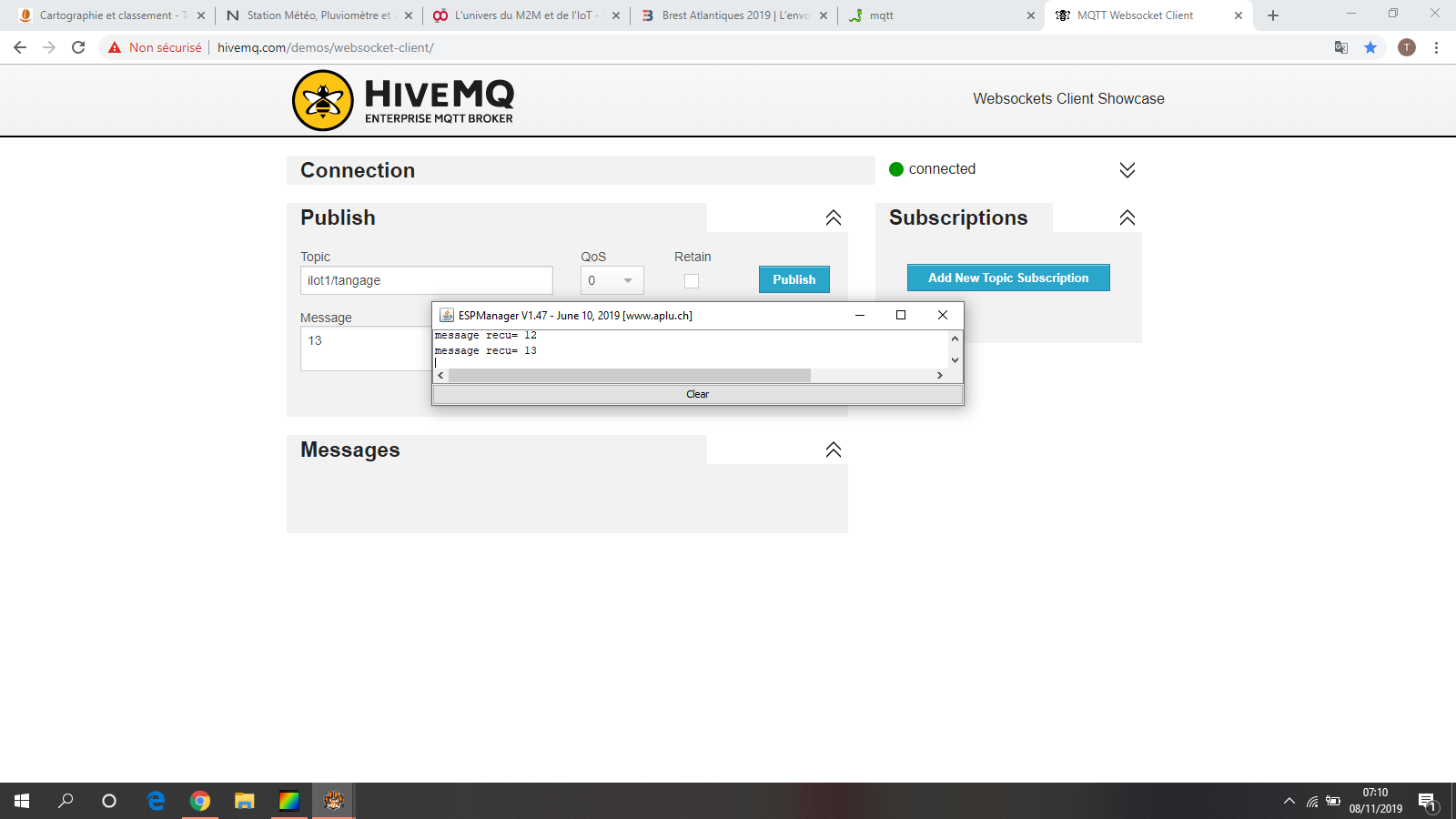


Tu vois la donnée arriver (payload)



# Souscription avec l’Oxocard





# Piloter une LED connectée avec l’Oxocard en utilisant le protocole MQTT

1. Recherche le schéma du réseau à créer à partir de 2 cartes Oxocard. Une carte joue le rôle de bouton connecté et a deuxième de lampe connecté. On utilisera une box (point d’accès + routeur).
2. Décrire le protocole de communication utilisé. Définir les payloads, topic utilisés.
3. Décrire les algorithmes sur chacune des cartes.
4. Implémenter les programmes en Python associés
5. Faire ses essais et valider ses solutions.