Laboratoire 4 – MQTT

Questionnement :

- Quelle serait la façon d'échanger de l'information à travers le réseau internet pour un système à ressources limitées, tel l'ESP32 et Arduino?
- Quels sont les outils de diagnostic du protocole MQTT?
- Comment utiliser le protocole MQTT sur un ESP32?
- Comment réaliser une application complète d'échange d'information entre deux microcontrôleurs à travers le réseau internet?

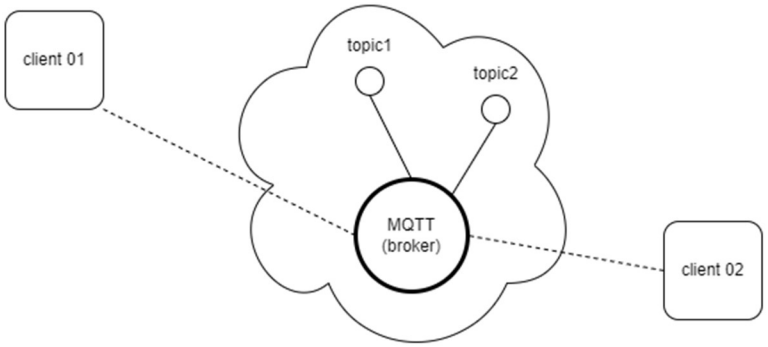
Matériel requis :

- Carte protoTPhys
- Blindage Ethernet
- Câble Ethernet

Théorie associée :

Le protocole MQTT est un protocole de communication au même titre que le http mais mieux adapté pour l'internet des objets. En effet, le protocole ne requiert plus de spécifier une méthode d'accès comme GET ou POST ainsi que son protocole restreignant (entête et corps). Le canal de communication est plus flexible : bidirectionnel, facilitant ainsi l'envoi et réception. À l'instar d'un compte X, l'application publie et s'abonne sur des canaux (*topics*), gérés par un serveur central, le courtier MQTT (*broker*).

Le schéma de principe de ce labo repose sur le suivant :



NOTE : NE PAS UTILISER LES NOMS (entités et topics) MONTRÉS SUR CETTE IMAGE (*topic1*, *topic2*)

Pour plus de détails, consulter les notes de cours et les références fournies.

Manipulation 1 – Le protocole MQTT

Le but de cette première partie est de vous familiariser avec le protocole MQTT en utilisant l'utilitaire client MQTTX.

Lors de cette partie, vous échangerez des messages, en équipe de 2, chacun à l'aide du poste de travail de la classe. Vous devrez configurer l'application MQTTX pour vous connecter au service MQTT indiqué plus bas. Vous devrez vous abonner (*subscribe*) à un « *topic* », le vôtre. Vous devrez aussi publier sur un « *topic* », celui de votre collègue. Les noms utilisés des « *topics* » doivent être bien planifiés. En effet, si vous envoyez un message à un « *topic* », il faut bien qu'il y ait un abonné (*subscriber*) qui le reçoit.

L'utilitaire client MQTTX de la firme EMQ vous sera très pratique. Ce logiciel est déjà installé sur les postes de travail du CEGEP. Vous devez configurer la connexion au fournisseur *shiftr.io*.

Pour cette partie, nous utiliserons le service infonuagique MQTT de *shiftr.io* : https://www.shiftr.io/

Une instance MQTT avec un courtier (*broker*) a été créée sur la plateforme *shiftr.io*, dont les paramètres sont les suivants :

- Name : vous devez nommer cette configuration pour l'application MQTTX.
- ClientID : un identifiant unique vis-à-vis le *broker*, sera suggéré par l'enseignant.
- Host : **technophyscal.cloud.shiftr.io**
- Protocole : **mqtt://**
- Le port est **1883**.
- Username : **technophyscal**
- Password : sera fourni par l'enseignant.
- Aucun SSL/TLS n'est utilisé.
- La version de protocole MQTT est **3.1.1**

Une fois la connexion confirmée par MQTTX, vous devriez voir votre entité sur la carte réseau projetée en avant de la classe, par l'enseignant.

Vous devez ensuite souscrire à un « *topic* » (*subscription*), entendez-vous entre collègue d'équipe pour des noms de *topic* judicieux.

Confirmer l'envoi et la réception de messages entre les deux stations de travail, à travers le service MQTT.

Démontrer à l'enseignant le bon fonctionnement de l'envoi et réception entre 2 postes de travail.

Manipulation 2 – le client MQTT ESP32

Le but de cette partie est de transférer ce que vous avez expérimenté dans la manipulation 1 à une application basée sur le ESP32. Il s'agit de transférer des messages entre un ESP32 et une station de travail via le client MQTTX.

En guise de familiarisation avec la programmation, la librairie *PubSubClient* est préconisée, on vous suggère le code exemple **CommObj_ESP32_clientMQTT-shiftr_io-Ethernet_Vetudiant.ino**

Ce code doit être finalisé et adapté afin de parvenir à connecter votre ESP32 au nuage du courtier MQTT de *shiftr.io*. Outre l'emploi des bons paramètres, dans les bonnes variables (similaire à ce que vous avez pratiqué en configurant l'application MQTTX), vous devrez utiliser le réseau Ethernet (**aucun Wifi**).

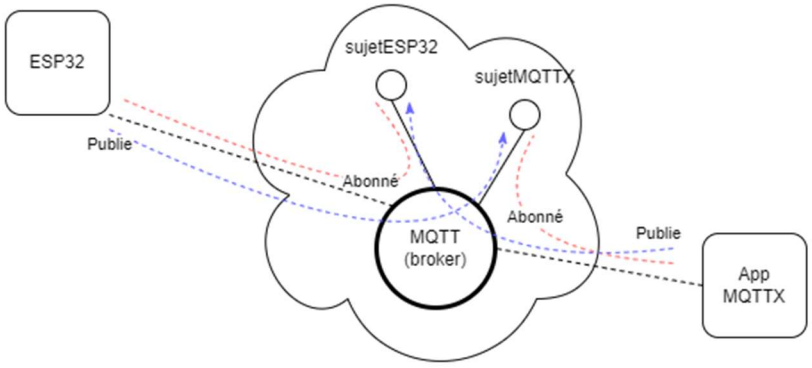
L'enseignant vous fournira le jeton (*token*) ainsi qu'un identifiant vous autorisant à opérer sur la plateforme infonuagique *technophyscal.cloud.shiftr.io*.

Votre travail dans cette partie du laboratoire:

- Lire attentivement le code logiciel fourni en exemple, lire les commentaires.
- Installer la librairie *PubSubClient* indiquée dans le code exemple. Consulter la documentation de la librairie pour une compréhension des méthodes.
- Établir les noms des entités et des sujets (*topics*) qui seront utilisés, de façon distincte, du point de vue du courtier MQTT (*broker*). Référez-vous au schéma de principe ci-bas.
- Compléter le code
  - À TOUS les endroits commençant par « [code …] »
  - Paramétriser la connectivité MQTT à *shiftr.io*
  - Valider à l'aide d'une compilation pour le ESP32 DOIT
- Vérifier la connectivité au nuage MQTT de *shiftr.io* en utilisant l'utilitaire MQTTX.
- Confirmer la connectivité avec la carte que projette l'enseignant à l'écran de la classe.

Il faut voir dans MQTTX le message publié par le ESP32 et voir au moniteur série du ESP32 le message publié à partir de MQTTX, pour le ESP32.

Schéma de principe :



Présentez à l'enseignant le résultat de votre application PC <-> ESP32.

Manipulation 3 – commande à distance à l'aide de MQTT

Le but de cette partie est de réaliser, sur une base de paire de dispositifs ESP32, un système télécommandé. Les deux dispositifs doivent être des clients MQTT qui s'envoient mutuellement une commande devant allumer et éteindre une DEL. Cette DEL pouvant éventuellement représenter un actionneur, un moteur, une alarme, ou tout autre système quelconque, commandé à distance.

Pour cette réalisation, vous **devez** travailler en équipe de 2.

Le client émetteur attendra au moniteur série un message d'action (à définir) qui enverra un message (format JSON) contenant une commande au client commandé. Par exemple : « allume la DEL 1 ». Le client commandé devra recevoir et décoder l'information afin de prendre action selon le contenu du message. Le client retourne finalement une confirmation (un message, format JSON) au client émetteur et exprimant le résultat de sa demande d'action.

Vous allez devoir définir le contenu du message, **obligatoirement en format JSON**. Le client émetteur prépare le contenu, le sérialise et l'envoie sur un sujet (*MQTT topic*) auquel le client commandé est abonné. Le client commandé reçoit le message, le désérialise, en interprète le contenu dans le but de prendre action. On veut simplement contrôler une des 2 DELs de la plaquette protoTPhys.

Dans le but d'obtenir une certaine robustesse de l'ensemble, le client commandé confirme l'action au client demandeur à travers un retour de message (format JSON). Le client émetteur affiche à l'écran OLED et sur le moniteur série la confirmation.

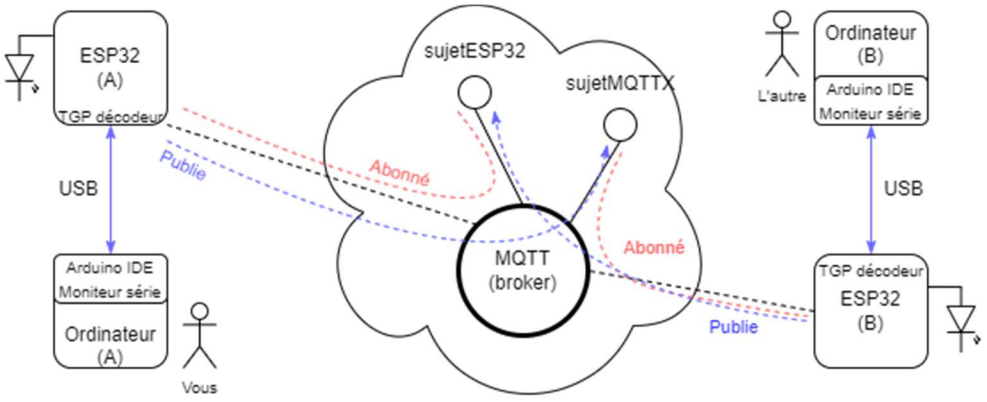
Pour vous aider dans cette réalisation, vous avez le code de base pour connecter, publier et s'abonner au service de messagerie MQTT, testé lors de la manipulation 2.

Vous devez utiliser la librairie **TGP_Decodeur** afin que le client demandeur traite les commandes lancées depuis l'interface série (connecté à l'ordinateur).

Débuter l'analyse par du pseudocode, définissant les étapes de traitement de chacun des rôles.

Prenez aussi un moment pour établir les champs d'information et leur contenu du message JSON.

Schéma de principe :



Présentez à l'enseignant le résultat de votre application.

Remise

Consignes pour le résumé de laboratoire.

- Le résumé de laboratoire représente avant tout un recueil de vos notes, observations et résultats de vos travaux. Il a pour but de forger en vous une bonne pratique professionnelle, en tant que futur technologue, de consigner et de décrire vos travaux sur des mandats qui vous seront confiés.
- 1 remise de travail par équipe.
- Je veux avoir, sur peu de pages, sections bien identifiées :

  Une **page couverture** vous identifiant, titre du travail, présenté à, la date de remise, lieu de remise (CÉGEP A-L).

  Introduction :

  - Date des manipulations accompagné d'une brève description des travaux réalisés pour chaque période.
  - Rappel des objectifs du laboratoire (pourquoi faire ce labo?).

  Développement :

  - Veuillez développer/expliquer (**sommairement**) le déroulement général de votre code de la manipulation **3**. Chaque manipulation, séparément, bien identifiées, justifiez ce que font les parties pertinentes ajoutées ou modifiées par rapport au code de base fourni. À titre d'exemple, prenez comment le site RandomNerdsTutorial présente sa section « *How the Code Works* ».

  Conclusion :

  - Selon l'expérience de la **manipulation 3**, décrivez une application potentielle, réelle, utile de ce système.
  - Discussion sur les résultats obtenus, notes et observations pertinentes.
  - Ce que vous avez appris ou découvert (ou peut-être pas).
  - Une critique constructive de cet exercice de laboratoire. Votre appréciation de ce laboratoire. Soyez honnête.

- Fournir le code logiciel complet de la **manipulations 3**, bien identifiée. N'oubliez pas d'appliquer correctement et entièrement le modèle de code (entête, section, commentaires).

- Consigne pour la remise
  - Rapport format **Microsoft Word**
  - Remettre le programme Arduino de la manipulation 3 seulement
  - **SVP: Aucun fichier archive (zip)**