**Rendu TP HTTP et MQTT**

Prérequis : installation de node.js (version 10)

Le montage des pins a été respecté selon les consignes, à savoir :

* la led sur la pin D19
* le sensor sur la pin D23
* le photoresistor sur A0 (36) (non utilisé)

**Présentation de HTTP et MQTT :**

**HTTP :**

La structure du code est la suivante : l’ESP32 va envoyer un fichier JSON en suivant les requêtes effectuées par une API en Node.js. Nous pouvons observer la température sur le client (la page web se rafraichit toute les secondes). Nous pouvons aussi allumer ou éteindre la led.

Les procédures à effectuer sont les suivantes :

* Se rendre dans le dossier « HTTP ».
* Modifier le ssid et password pour le fichier « http.ino »
* Charger « http.ino » sur l’ESP32
* Aller dans le dossier serveur
* Lancer la commande « npm install »
* Pensez a vérifier que le serveur est bien connecté en wifi avec le moniteur série.
* Pensez à vérifier l’ip du arduino pour voir si elle correspond à celle indiqué dans le fichier serveur.js (situé dans /http/Serveur/server.js) et à être connecter sur la même connexion pour le PC et l’ESP32
* Lancer la commande « node server.js »
* Dans le client, ouvrir le fichier « index.html » dans le navigateur.

**MQTT :**

Connexion au broker broker.shiftr.io

Le serveur permet de faire clignoter toutes les secondes. On peut aussi récupérer la température en temps « réel » (c’est-à-dire la valeur qui se trouve sur le broker). Le résultat est afficher dans la console.

Les procédures à effectuer pour lancer :

* Se rendre dans le dossier « MQTT »
* Modifier le ssid et password pour le fichier « mqtt.ino »
* Charger « mqtt.ino » sur l’ESP32
* Pensez à vérifier que le serveur est bien connecté en wifi avec le moniteur série.
* Allez dans le dossier serveur
* Lancer la commande « npm install »
* Lancer la commande « node server.js »
* Dans le client, ouvrir le fichier « index.html » dans le navigateur.