

TP1 : HTTP REST et MQTT avec ESP32, DHT11 et Node-RED

Objectif du TP

Ce TP a pour objectif de mettre en pratique deux architectures de communication bidirectionnelles en utilisant les protocoles HTTP REST et MQTT. Pour ce faire, nous allons utiliser un module Wifi ESP32, un capteur de température DHT11 et une LED intégrée à la carte ESP32.

Objectifs :

1. Créer deux démonstrations pour valider la communication bidirectionnelle.
 - **Uplink**: Transmettre les données de température et d'humidité du capteur DHT11 vers une interface web et les afficher sur un graphique.
 - **Downlink**: Contrôler une LED sur le module ESP32 depuis une interface web.
2. Utiliser Node-RED pour la partie Cloud, installé en local sur votre ordinateur, ou coder cette partie en Python ou Javascript si vous connaissez déjà bien ces technologies.
3. Installer et configurer le broker Mosquitto pour MQTT.

Matériel Requis

- Module Wifi ESP32
- Capteur de température DHT11
- LED intégrée sur ESP32
- Ordinateur avec Node-RED et Mosquitto installés

Instructions

Partie 1: Configuration de l'environnement

- Installez Node-RED et Mosquitto sur votre ordinateur si ce n'est pas déjà fait.
- Si vous avez accès à un serveur type OVH, AWS ou Azure avec une adresse IP publique, configurez ce dernier pour accueillir Node-RED et Mosquitto.

Partie 2: Configuration de l'ESP32 et du capteur DHT11

- Connectez le capteur DHT11 à votre ESP32.
- Téléchargez et installez les bibliothèques nécessaires pour faire fonctionner l'ESP32 et le DHT11.

Partie 3: Communication HTTP REST

- Programmez l'ESP32 pour qu'il lise les données du capteur DHT11.
- Envoyez ces données à Node-RED en utilisant le protocole REST.
- Utilisez Node-RED pour créer un graphique affichant ces données en temps réel.
- Implémentez une fonctionnalité pour contrôler la LED de l'ESP32 via une interface web dans Node-RED.

Partie 4: Communication MQTT

- Configurez l'ESP32 pour qu'il se connecte au broker Mosquitto.
- Modifiez le code de l'ESP32 pour qu'il envoie les données du DHT11 via MQTT.
- Utilisez Node-RED pour souscrire au topic MQTT et afficher les données sur un graphique.
- Implémentez une fonctionnalité pour contrôler la LED de l'ESP32 via MQTT.

Partie 5: Validation

- Faites vérifier par l'enseignant les deux solutions de communication.
- Présentez une démonstration de chaque solution.

Livrables

- Code source pour l'ESP32 pour les protocoles HTTP REST et MQTT.
- Flows de Node-RED.
- Rapport final expliquant comment configurer l'ensemble du système et contenant les résultats et l'analyse des deux méthodes de communication.

Évaluation

L'évaluation se fera sur la base des livrables fournis, de la qualité de la démonstration et de la compréhension des concepts sous-jacents des protocoles REST et MQTT.

Bon travail !