

# IOTB-TP1-Rest/MQTT-raport

ivan.imbert

Juin 2024

## Contents

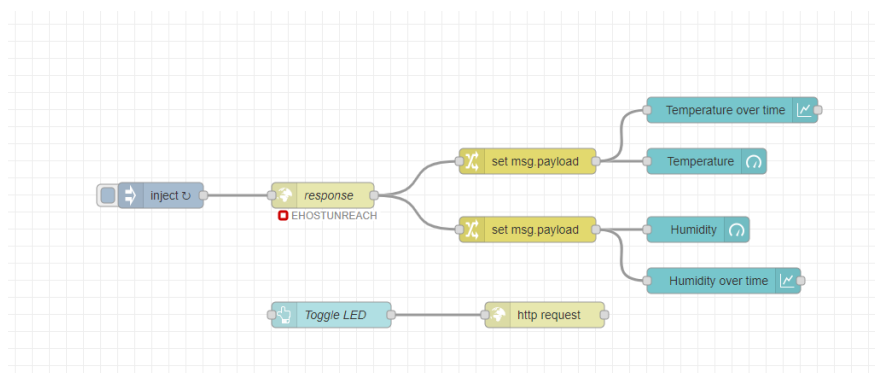
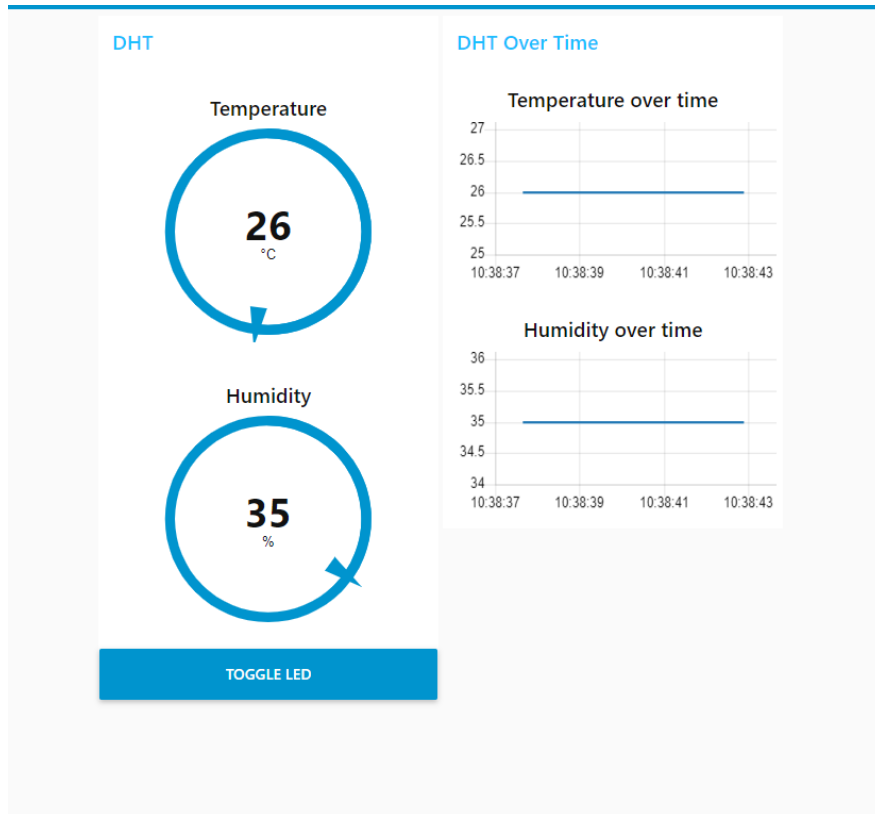
<b>1</b>	<b>Rest</b>	<b>2</b>
1.1	Rendu . . . . .	2
1.2	Reproduire . . . . .	2
1.2.1	Prérequis . . . . .	2
1.2.2	Setup . . . . .	3
<b>2</b>	<b>MQTT</b>	<b>3</b>
2.1	Rendu . . . . .	3
2.2	Reproduire . . . . .	4
2.2.1	Prérequis . . . . .	4
2.2.2	Setup . . . . .	4
<b>3</b>	<b>Analyse</b>	<b>5</b>

# 1 Rest

## 1.1 Rendu

Pour la partie Rest, l'esp32 sert de serveur et node red de client.

Pour le côté serveur, l'esp32 récolte la valeur du capteur régulièrement. Au moment des requêtes, le serveur renvoie la valeur actuelle des variables.



## 1.2 Reproduire

### 1.2.1 Prérequis

Pour la partie Node-Red, il suffit d'ajouter à la palette 'node-red-dashboard'

Pour la partie Esp32, il faut d'abord installer de quoi upload sur la carte esp32 ensuite, il faut installer la librairie DHT11 de "Dhruba Saha".

Les bibliothèques Wifi et aRest sont normalement déjà présentes sur l'IDE.

Concernant le setup hardware, la led utilisée est la led présente sur le port 2 (led builtin) et le capteur est utilisé sur le port D13.

### 1.2.2 Setup

Dans le script 'rest.ino' il faut changer les champs 'ssid' et 'password' (respectivement nom et mot de passe du wifi utilisé).

Pour la suite il faut upload le script sur la carte et si besoin reset la carte avec le bouton présent.

Sur le Serial Monitor, une ip devrait être affichée, il faut garder cette ip qui est l'adresse du serveur.

Maintenant dans node-red il faut modifier l'ip dans les nodes. L'ip est précisée 2 fois dans le flow. 1 fois dans la node 'response' et une autre fois dans la node 'http request'.

Une fois fini il suffit de deploy le flow et d'ouvrir le dashboard.

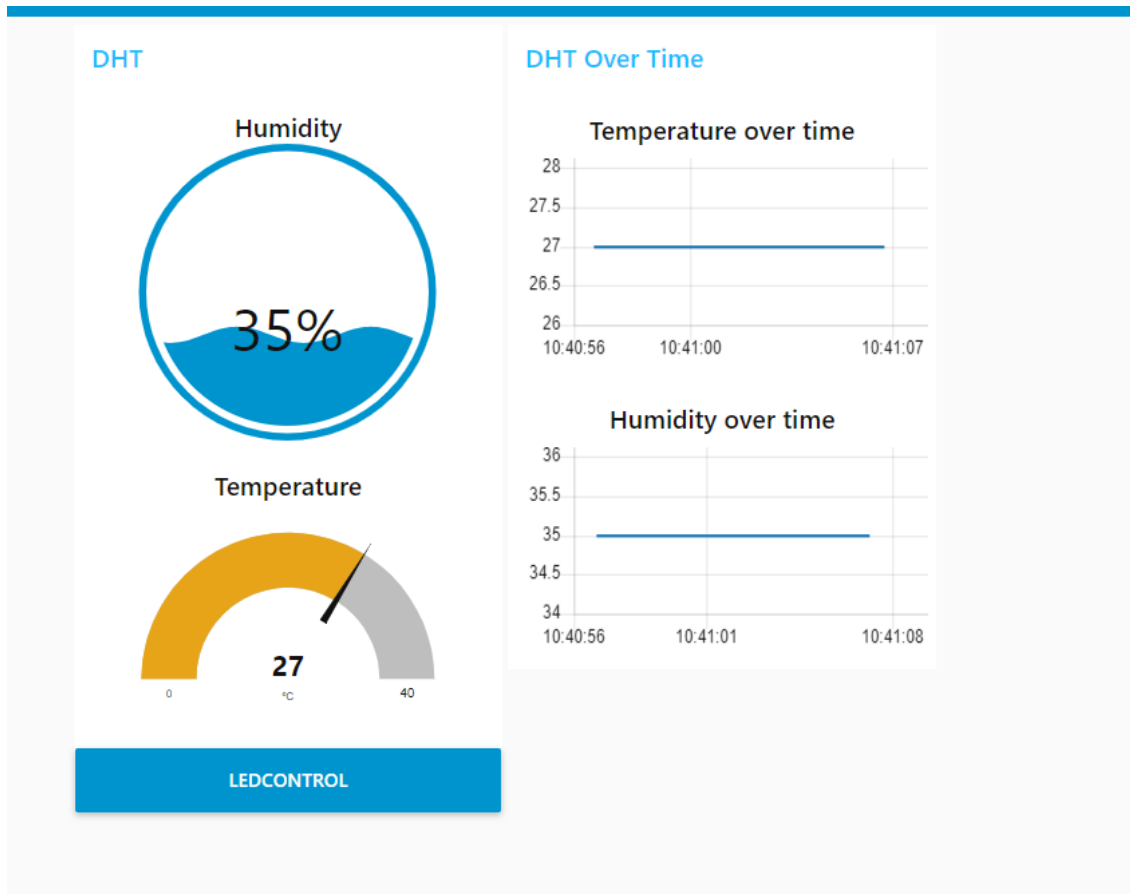
## 2 MQTT

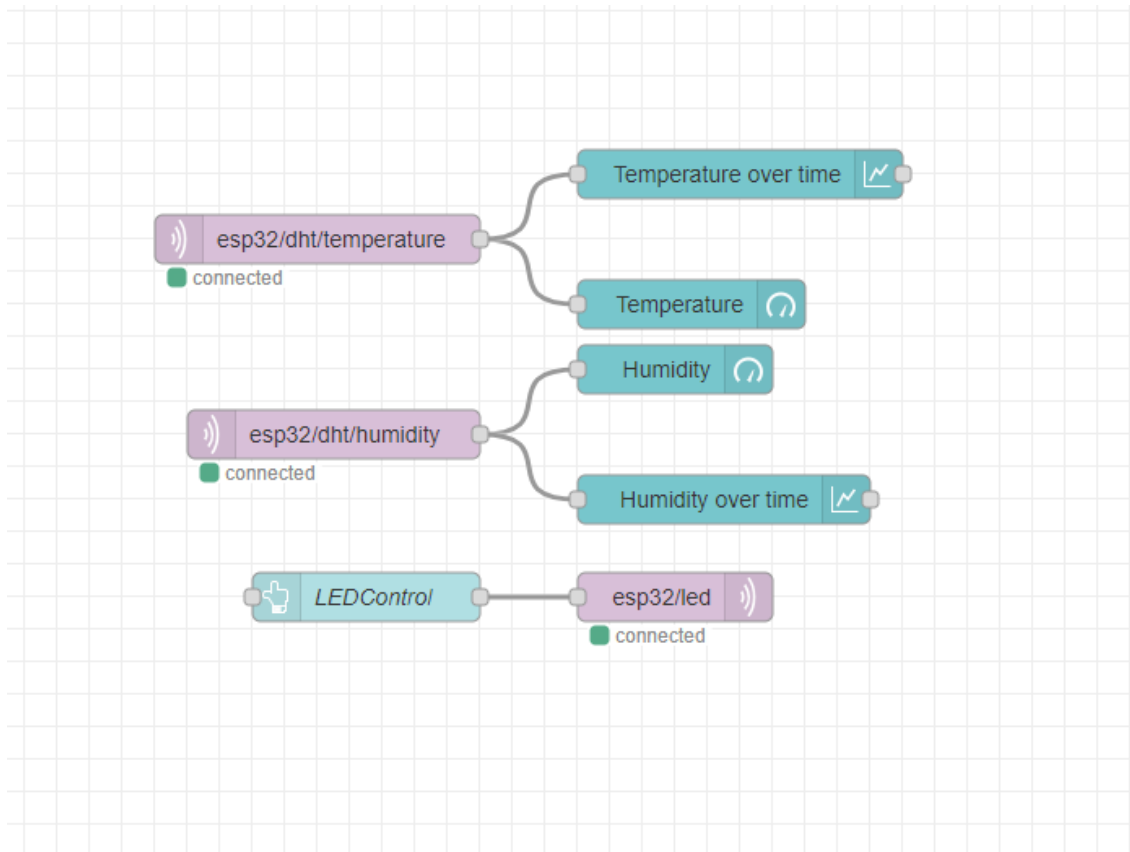
### 2.1 Rendu

Pour la partie MQTT, l'ordi sert de serveur tandis que l'esp32 et node-red servent de publisher/subscriber.

Du côté de l'esp32 la valeur du capteur est récoltée régulièrement. Au moment de la récolte des valeurs, il envoie un message sur le topic approprié afin d'informer les subscribers d'un potentiel changement de valeur. Pour la led, l'esp32 écoute sur un topic qui l'informe d'une demande de changement d'état.

Côté node-red, on écoute sur les topics concernant l'humidité et la température et on affiche les valeurs en direct. Concernant la led, le bouton envoie simplement un message sur le topic 'esp32/led'





## 2.2 Reproduire

### 2.2.1 Prérequis

Pour la partie Node-Red, il suffit d'ajouter à la palette 'node-red-dashboard'

Pour la partie Esp32, il faut d'abord installer de quoi upload sur la carte esp32 ensuite, il faut installer la librairie DHT11 de "Dhruba Saha", et la librairie ArduinoMqttClient d'Arduino.

Pour la partie serveur il faut lancer le serveur et changer la config afin d'autoriser les connections extérieures et noter l'ip de l'ordi.

La librairie Wifi est normalement déjà présente sur l'IDE.

Concernant le setup hardware, la led utilisée est la led présente sur le port 2 (led builtin) et le capteur est utilisé sur le port D13.

### 2.2.2 Setup

Dans le script 'mqtt.ino' il faut changer les champs 'ssid' et 'password' (respectivement nom et mot de passe du wifi utilisé). De plus il faut changer les champs broker et port qui sont l'ip du serveur MQTT et le port sur lequel il écoute.

Pour la suite il faut upload le script sur la carte et si besoin reset la carte avec le bouton présent.

Sur le Serial Monitor, l'esp32 devrait afficher une connection au Wifi ainsi qu'une écoute sur le topic 'esp32/led'

Maintenant dans node-red il faut modifier l'ip dans les nodes. L'ip est précisée 3 fois dans le flow. 1 fois dans la node 'esp32/dht/temperature', une fois dans la node 'esp32/dht/humidity' et une autre fois dans la node 'esp32/led', l'ip est noté dans un server, il faudra potentiellement créer un nouveau "serveur" contenant la bonne ip et le sélectionner dans la liste des choix disponibles.

Une fois fini il suffit de deploy le flow et d'ouvrir le dashboard.

### 3 Analyse

Concernant ces deux méthodes de communication, ce qui serait le mieux serait d'utiliser la partie MQTT pour la température et l'humidité afin d'avoir des données en direct et pouvoir plus simplement distribuer les données dans un cas où il y aurait plusieurs dashboards connectés.

Pour la partie led je pense qu'il serait plus pratique d'utiliser un serveur REST pour ne pas avoir à passer par le broker pour si peu.

Il est complètement inutile d'envoyer à tout le monde ce changement, la seule utilité serait dans le cas où plusieurs capteurs seraient entrain d'écouter à la même commande mais si un seul capteur est utilisé et plusieurs dashboard un serveur rest serait plus adaptés

À la limite un autre topic MQTT sur lequel le capteur renverrait l'état actuel de la led en cas de changement pourrait être utile afin d'avoir un feedback de l'état de la led.