

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

22/02/2022

Documentation TP2

Robin, Axel, Luc, et Adrien

Several thin, dark blue wavy lines originate from the bottom of the vertical bar and curve upwards and to the right, resembling stylized grass or reeds.

Sommaire

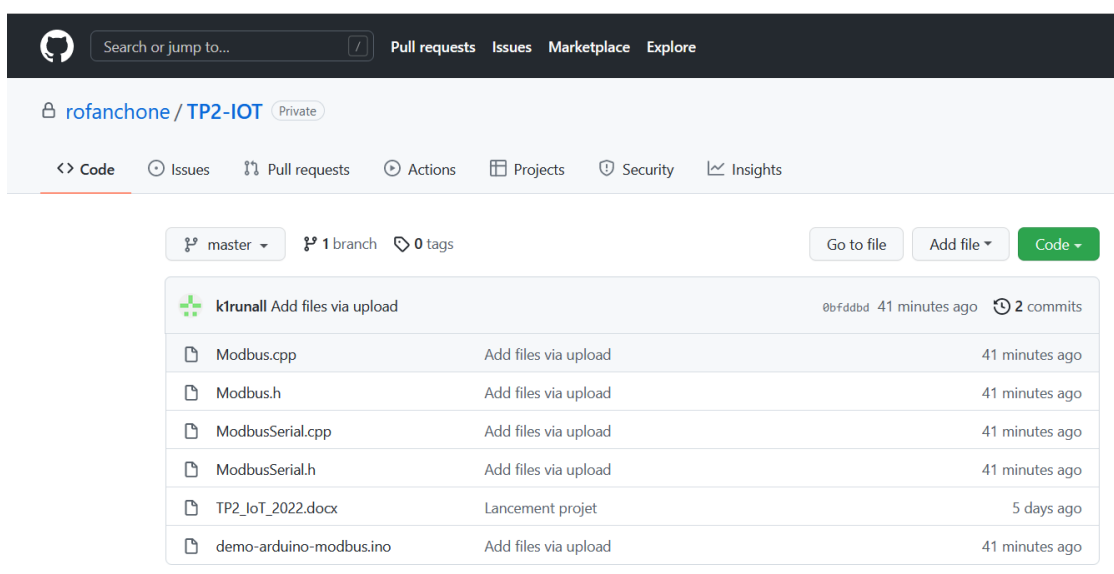
I- Recherche	2
I.I- Contrôle de sources	2
II- Prototype	3
III- Mode de transport	3
III.I- MQTT	3
III.II- Protocole série	3
IV- Scripts	3
IV.I- Arduino	3
IV.II- Script Raspberry PI	3
V- Site Web	3
V.I- Visualisation des données	3
V.II- Sécuritaire	3
V.III- REST API	3
VI- Base de Donnée	3

I- Recherche

I.I- Contrôle de sources

Comme nous étions en groupe de 4 pour la réalisation de ce projet, il a fallu trouver un moyen de communiquer les codes de projet de chacun, et les documentations utiles. Pour cela nous avons utilisé GITHUB, car c'est un outil qui remplit toutes les conditions pour notre projet, de plus notre travail sera partagé sur internet donc si le moindre problème arrive nous pourrions toujours récupérer notre code et nos documentations.

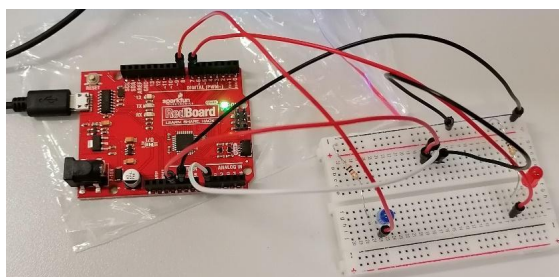
<https://github.com/rofachone/TP2-IOT>



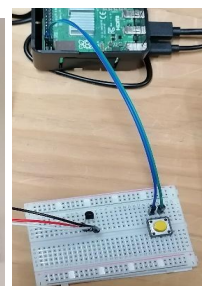
II- Prototype

Pour notre prototype, nous avons pris exemple sur d'anciens TP de cours, puis nous les avons modifiés pour qu'ils correspondent à nos besoins.

Prototype Pour la carte Arduino



Prototype Raspberry Pi



III- Mode de transport

III.I- MQTT

MQTT (Message Queuing Telemetry Transport) est un protocole de transport que nous avons utilisé pour la communication entre le serveur de base de données et le raspberry pi.



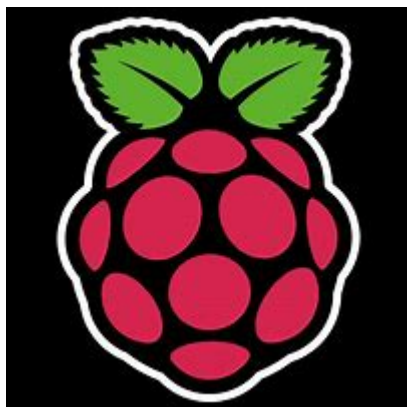
III.II- Protocole série

Nous avons utilisé un protocole série sur le raspberry et le arduino via le module python pyserial pour la transition des températures du arduino vers le raspberry.

IV- Scripts

IV.II- Script Raspberry PI

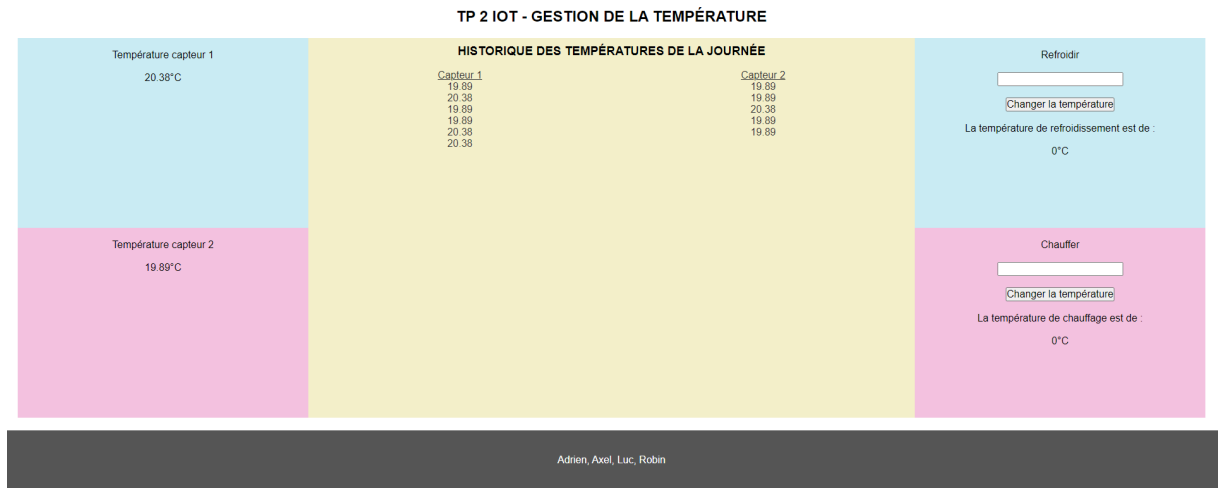
Le script python SerialAndMQTT sur le Raspberry PI nous permet d'envoyer la température des capteurs vers le serveur via MQTT. Un client mqtt nodejs récupère ensuite les données et les insère dans la base de données.



V- Site Web

V.I- Visualisation des données

Notre site web permet de visualiser les températures de la base de données et la température actuelle des capteurs. Le site permet aussi de modifier la température à partir de laquelle les leds s'allument.



v.II- Sécurité

On a utilisé HTTPS avec un certificat auto généré pour sécuriser un minimum. Toutes les requêtes à destination du port http 80 seront redirigées sur le port 443 https.

v.III- REST API

Les apis des deux capteurs se situent dans /api/capteur1 et /api/capteur2 et contiennent les températures qui se trouvent dans la base de données ainsi que le ID dans la base.

VI- Base de Donnée

Pour la base de données, nous avons décidé de l'installer directement sur l'ordinateur. Puis est venu le choix de quelle base utilisée, nous avons hésité entre Mariadb et Mongodb. Finalement, nous avons choisi Mongodb.



Nous avons ensuite utilisé la documentation Mongo pour l'installation:
<https://docs.mongodb.com/manual/tutorial/install-mongodb-on-debian/>

La base de données permet d'enregistrer les données envoyées par les deux capteurs dans leur table respective. Avec l'utilisation du client mqtt nodejs, on va récupérer les données pour les envoyer à la base de données et pour pouvoir les récupérer pour notre api et afficher les données sur le site web en temps réel.

