

Documentation de test

Extraction des valeurs

Table des matières

1. Contexte Python – IoT.....	2
2. Fonctionnalité.....	2
3. Code.....	2
4. Tests.....	2

1. Contexte Python – IoT

Nous devons réaliser des solutions avec l'Internet des Objets (IoT) pour simplifier la gestion des entrepôts de stockage ou faire des économies d'énergie de notre client.

Dans la partie python, nous devons lire en fonction du fichier de configuration crée par Java des données à l'aide du capteur AM107.

2. Fonctionnalité

Nous devons extraire les valeurs du capteur AM107, les données du capteurs sont au format JSON.

3. Code

Nous importons paho.mqtt.client pour pouvoir avoir les données du capteur

```
print("Connexion aux locaux de Blue Gym...")

client = mqtt.Client()
client.connect(config["servers"], config["port"], 600)

client.subscribe("application/1/device/+/event/up")

client.on_message = get_data

client.loop_forever()
```

Nous nous connectons en fonction du serveur, du port qui sont dans le fichier de configuration, mais par défaut, nous utilisons le serveur de l'IUT chirpstack.iut-blagnac.fr et le port 1883 puis nous nous abonnons dans le application 1 où il y a les capteurs AM107, le « + » prend un des capteurs disponibles et puis les données par « /event/up ». Une fois abonné, on utilise la fonction get_data en fonction de .on_message. On tourne le mqtt afin que les données du capteur soient prises.

4. Tests

Il n'y a pas vraiment de test, la fonction extrait les données par jsonMsg = json.loads(msg.payload) de la librairie json, puis pour extraire les valeurs fonctionne par double table, pour récupérer le co2 il faudra utiliser jsonMsg[« object »][« co2 »], il faut aller dans la table object et puis appeler le type de données.

```
def get_data(mqtt, obj, msg):
    jsonMsg = json.loads(msg.payload)
```