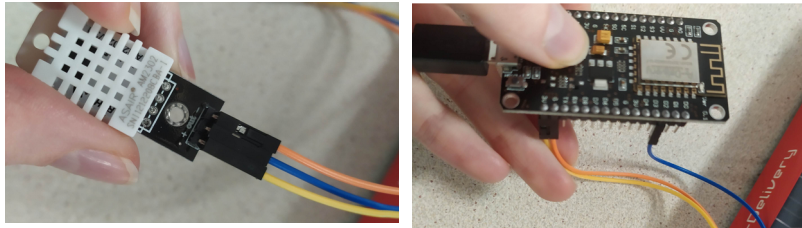


Solutions numériques de l'industrie 4.0

Travaux pratiques

1. Récupération des données des capteurs



Dans un premier temps nous avons branché le capteur de température et d'humidité à la carte ESP8266.

Nous avons choisi la pin D2 (GPIO4) pour récupérer les données du capteur.

Une fois le branchement réalisé, nous avons écrit un code pour récupérer et afficher à la console les données captées en temps réel. Nous avons pu faire les tests sur le logiciel ArduinoIDE et voir les données suivantes :

```
{ "Temperature":22.70,"Humidity":36.30}
```

2. Envoi des données sur le broker Raspberry

Une fois cela fait, nous avons modifié notre code pour qu'il envoie sur le broker ces données. Notre client MQTT a donc envoyé nos données sur le topic ESIEA/grp1 comme montré ci-dessous :

```
#define wifi_ssid "Invite-ESIEA"  
#define wifi_password "bQV86deaaZQZPu9a"  
  
#define mqtt_server "10.8.128.250"  
  
#define topic "ESIEA/grp1"
```

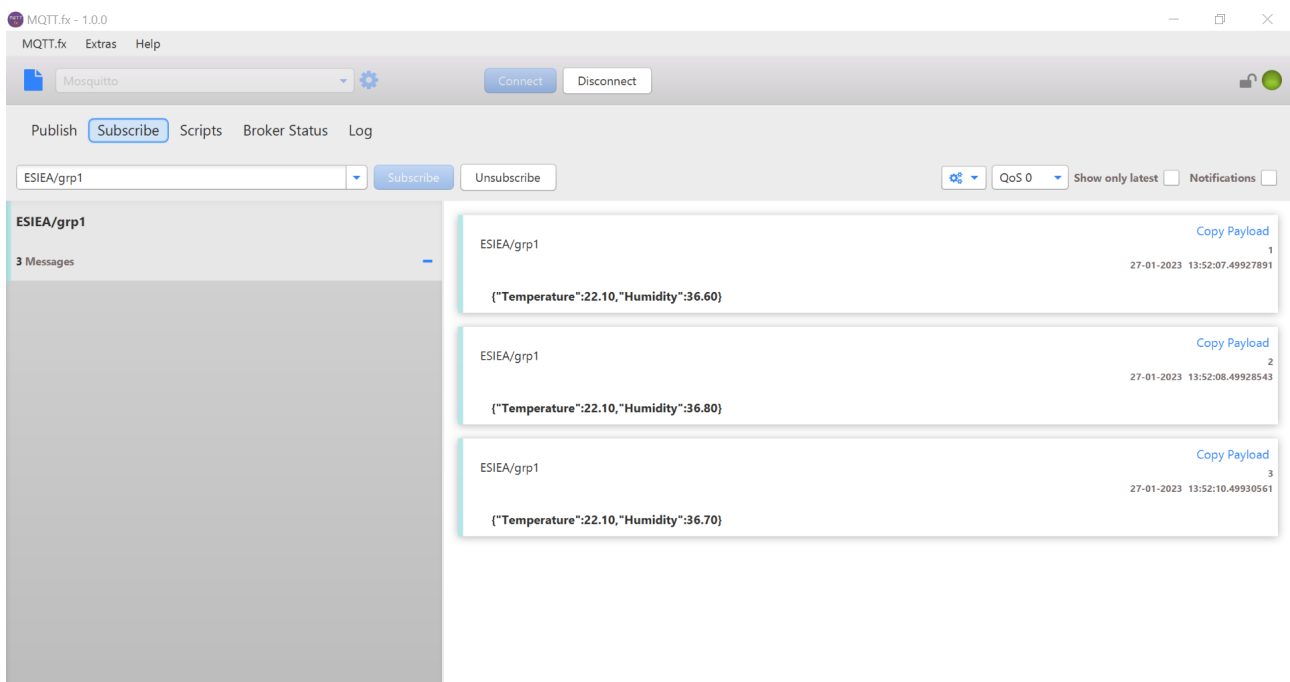
Le broker recevait bien nos données :

```
{ "Temperature":22.70,"Humidity":36.30}  
OK      36.4      22.7  
{ "Temperature":22.70,"Humidity":36.40}  
OK      36.3      22.7  
{ "Temperature":22.70,"Humidity":36.30}  
OK      36.5      22.8  
{ "Temperature":22.80,"Humidity":36.50}  
OK      36.6      22.8
```

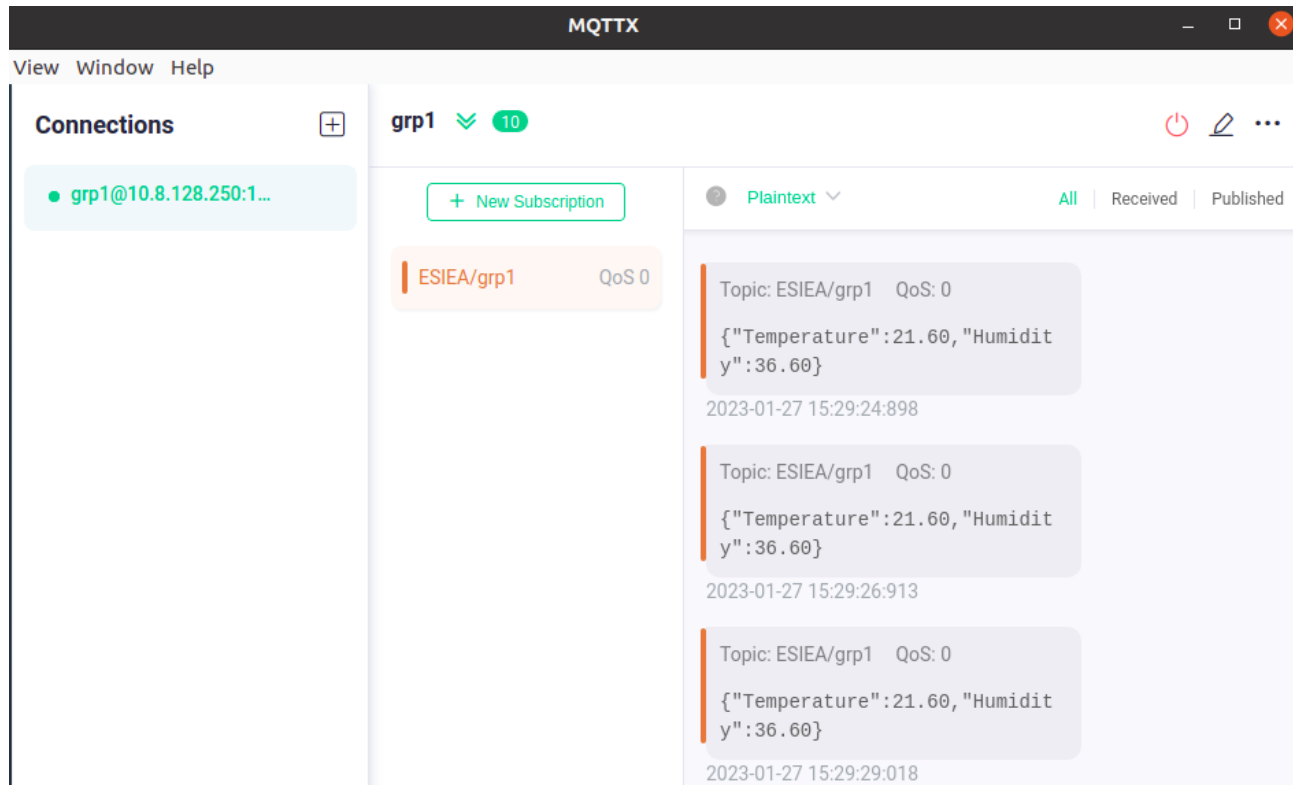
3. Visualisation des données via un client

Pour vérifier que nos données ont été envoyées, nous avons créé un client sur le logiciel MQTT.FX qui a souscrit à notre topic sur le broker.

La capture ci-dessous montre que nos données ont bien été reçues.

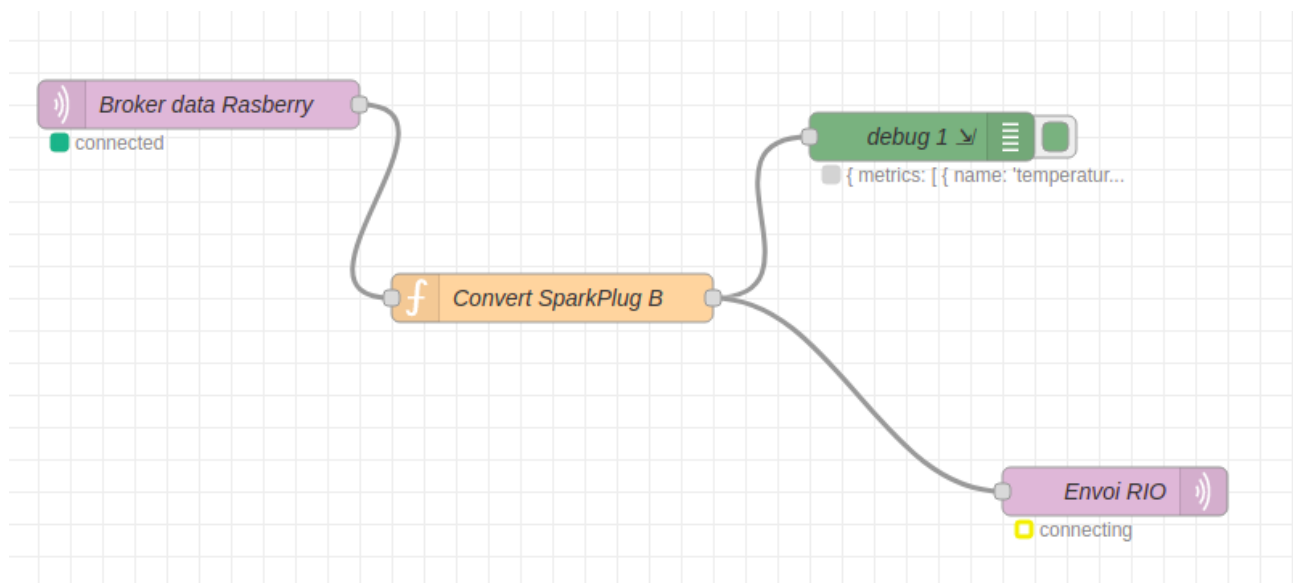


Nous avons également testé sur le logiciel MQTTeX qui fonctionne lui sur Linux.



4. Standardisation des données

Pour que nos données soient comprises par tous les clients, nous les avons standardisées selon le standard Sparkplug B sur le logiciel Node Red. Ci-dessous notre schéma de standardisation.



Les données sortent de Node Red comme suit :

```
ESIEA/grp1 : msg.payload : Object
  ▼ object
  ▼ metrics: array[1]
    ▼ 0: object
      name: "temperature&humidity"
      ▼ value: object
        Temperature: 23.9
        Humidity: 33.1
```

5. Envoi des données standardisées au RIO

Enfin, pour vérifier si notre standard est correct, nous envoyons grâce à NodeRed nos nouvelles données vers le broker RIO, à l'emplacement spBv1_0/ESIEA/DDATA/raspberry/grp1.