

BeagleBone et Linux Debian

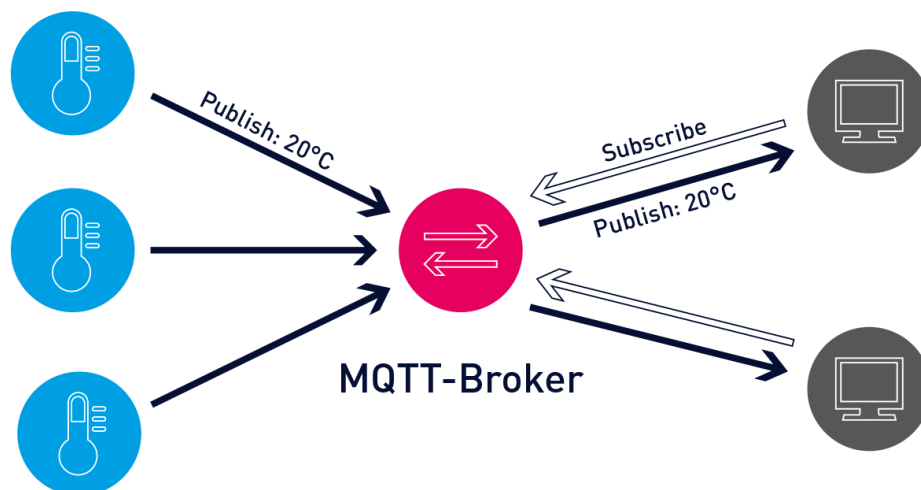
Travail supplémentaire

VOUS POUVEZ RÉALISER LE TRAVAIL DANS L'ORDRE QUE VOUS SOUHAITEZ.

1 MQTT

1. Test de communication avec le Broker MQTT

MQTT est un protocole simple et léger pour l'envoi des données, axé sur la communication machine-machine. Le protocole fonctionne avec un broker (serveur) qui permet l'agrégation des données de multiples clients qui publient des données selon des sujets spécifiques (« publish »). Les clients peuvent également « souscrire » (« subscribe ») à un sujet spécifique et recevoir les nouvelles données publiées sur le broker. Le fonctionnement est schématisé à la figure ci-dessous.



Un broker MQTT est déjà en place à l'ES sur l'adresse IP 10.228.134.240. Pour commencer avec MQTT, téléchargez un client GUI qui permet d'interagir de manière graphique avec un broker :

<http://mqtt-explorer.com/>

Connectez-vous au broker de l'ES et essayez de publier des données sur un sujet de votre création. Observez le résultat sur le broker.

- Grâce à la librairie Python « paho mqtt », complétez votre programme afin que les données du capteur de température soient envoyées de manière régulière sur le broker de l'ES. Observez le bon fonctionnement grâce à MQTT explorer.
- Une manière de visualiser les données d'un broker MQTT est de les inscrire dans une base de donnée automatiquement et de les afficher grâce à un framework appelé « Grafana ». Une base de donnée a déjà été mise en place sur la même machine que le broker. Demandez aux enseignants un accès à l'utilitaire Grafana et essayez d'afficher les données que vous envoyez au broker.

2 SERVEUR WEB

Il s'agit ici de mettre en place une page web dynamique qui affiche et met à jour régulièrement les valeurs lues par le capteur. Ce sera un framework Python qui permettra de mettre en place le serveur web. Il s'agit du framework "Flask", qui permet de mettre en place des solutions de backend pour des applications web de manière relativement aisée grâce au langage Python.

1. Installer le package Python Flask :

```
sudo pip install flask
```

Lorsque vous installez Flask, faites attention à la problématique Python2/Python3 et également pour quel utilisateur vous l'installez.

2. Création de la page html :

Il s'agit de créer une page html où les éléments variables seront placés entre {{doubles accolades}}. Ce type de contenu dynamique est basé sur un moteur de rendu appelé « Jinja2 ». Celui-ci permet de générer des pages html de manière simple et rapide. Il existe bien entendu de multiples manières de générer des contenus dynamiques, notamment avec Javascript, php et plein d'autres langages.

Le fichier html que vous devez créer est appelé « index.html » et doit par défaut se trouver dans un sous répertoire "/templates" par rapport au script python.

Exemple

```
<!--Template affichant l'heure du serveur-->
<!DOCTYPE html>
<html>
  <head>
    <title>Ici se trouve le titre de la page</title>
  </head>
  <body>
    <h1>Titre de paragraphe</h1>
    <p>
      Il est {{heure}}.
    </p>
  </body>
</html>
```

3. Mise en place du serveur dynamique en Python :

Votre script utilisera Flask, qui utilisera votre fichier html et remplacera les variables par leurs valeurs.

Exemple

```
#script Python pour lancement serveur flask et affichage heure
from flask import Flask, render_template #librairie Flask
import time #pour lecture de l'heure système

app = Flask(__name__)

# fonction qui répond lors de l'accès au répertoire web racine
@app.route('/')
def station_mesure():
    str_heure = time.strftime("%H:%M")
    return render_template('index.html', heure=str_heure)

# démarre le serveur si le script est lancé directement en tant que prg principal (pas appelé par un autre)
if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80, debug=True)
```



Le lancement d'un serveur web Flask nécessite les droits d'administrateurs. Vous devrez donc lancer votre script python avec sudo pour que cela fonctionne.

4. Utilisation d'une mise en forme CSS grâce à Bootstrap :

Les pages html peuvent être mise en forme de manière complexe grâce à un langage appelé CSS. Vous pouvez écrire vous-même du code CSS afin de spécifier des couleurs, des tailles d'éléments etc. Il est également possible d'utiliser des frameworks déjà disponibles qui permettent de mettre en place très rapidement des pages complexes. Un de ces framework s'appelle bootstrap (<https://getbootstrap.com/>).

Pour vous entraîner à utiliser Bootstrap, essayez d'intégrer le texte de votre page html dans une boîte centrée dans le navigateur.

Exemple :