

Découverte du protocole MQTT

Objectif

Durant cette séance, vous devez mettre en place un courtier MQTT et deux clients capables de publier et de s'abonner pour émettre et recevoir des données afin de piloter un objet communicant.

Premiers pas avec Mosquitto

1. Installez une version stable de Mosquitto pour votre environnement (<https://mosquitto.org/>)
2. Dans un terminal exécutez le client de souscription (mosquitto_sub) pour attendre des messages. Quels sont les arguments que vous devez indiquer ? A quoi correspondent ces options ?
3. Dans un autre terminal exécutez le client de publication (mosquitto_pub) pour publier un premier message en indiquant les arguments adéquats.
4. Par défaut, les messages que vous émettez sont-ils persistant ? Pourquoi ? Que proposez-vous ?
5. Testez maintenant l'échange de messages entre deux machines puis plusieurs machines. Quel paramètre devez-vous modifier ?
6. Manipulez les topics avec les wildcards + et # pour une arborescence :
 - #
 - temp/#
 - +temp
 - +/temp/#
7. Vous allez maintenant manipuler la qualité de service (QoS). Quelle commande devez-vous ajouter lors de l'émission d'un message ? Détaillez le comportement des messages pour les 3 niveaux de QoS.

Développez vos clients MQTT

La librairie la plus populaire pour développer des clients MQTT est sans doute PAHO (JAVA, C, Python) : <http://www.eclipse.org/paho/downloads.php>

8. Téléchargez la librairie dans votre langage de prédilection ou celui qui semble le plus pertinent pour la réalisation de votre projet
9. Ecrivez un client *publisher* permettant de :
 - a. se connecter au broker Mosquitto
 - b. publier un message simple dans un topic
10. Ajoutez la gestion de la qualité de service
11. Ecrivez maintenant un client, *subscriber*, permettant de récupérer un message
12. Testez vos clients (plusieurs langages) avec plusieurs machines et vérifiez en reprenant les questions de la première partie que les échanges avec le broker respectent bien le protocole MQTT.

Prendre en main Arduino

13. Installez l'IDE Arduino pour votre environnement : <https://www.arduino.cc/en/Main/Software#>
14. Effectuez le branchement présenté à la page 116 du livret d'accompagnement
15. Copiez le programme suivant dans l'IDE Arduino :

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  Serial.begin(9600);
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
}

void loop() {
  // variable contenant le caractère à lire
  char carlu = 0;
  // variable contenant le nombre de caractère disponibles dans le buffer
  int cardispo = 0;
  cardispo = Serial.available();
  while(cardispo > 0) // tant qu'il y a des caractères à lire
  {
    carlu = Serial.read(); // on lit le caractère
    // Print a message to the LCD.
    lcd.print(carlu);
    cardispo = Serial.available(); // on relit le nombre de caractères dispo
  }
}
```

16. Dans la console série de l'IDE, tapez un caractère et vérifiez qu'il s'affiche sur l'écran LCD
17. Modifiez votre client *subscriber* pour qu'il émette un message sur l'écran LCD à la réception d'un message sur un topic
18. Testez l'émission avec un client *publisher* depuis une autre machine