CAPÍTULO

MQTT

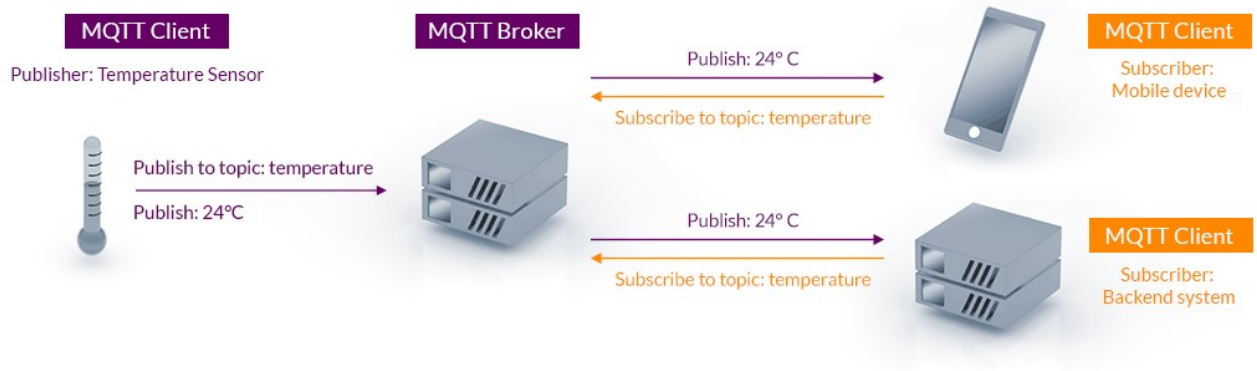
JPSantos
2021

10. MQTT

MQTT significa Message Queuing Telemetry Transport. Foi proposto pela IBM e normalizado pela OASIS e pela International Standard Organization (ISO) sob a designação IEC20922.

ISO/IEC 20922:2016 is a Client Server publish/subscribe messaging transport protocol

Os programas que implementam esta especificação, permitem o transporte de mensagens entre os seus dispositivos, não directamente, mas através de um dispositivo intermédio (Broker).
O Broker recebe mensagens com a informação, os tópicos "Topic", que os dispositivos querem publicar (Publish) e guarda essa informação.
Esses, ou outros dispositivos, enviam para o Broker mensagens a pedir "Subscribe" o valor de determinados tópicos, e nesse caso o Broker responde/envia mensagens para esses dispositivos, com os valores dos tópicos subscritos.



https://mqtt.org/

O transporte destas mensagens entre os dispositivos e o Broker, pode ser conseguida através do protocolo TCP/IP, num tipo de interação do tipo Cliente-Servidor.

O Broker é o Servidor, e os outros dispositivos são os Clientes que publicam ou/e subscrevem tópicos.

O transporte dessas mensagens, desses tópicos, pode ter três tipos de garantia de entrega "*Quality Of Service*":

- *"At most once", where messages are delivered according to the best efforts of the operating environment. Message loss can occur. This level could be used, for example, with ambient sensor data where it does not matter if an individual reading is lost as the next one will be published soon after.*

- *"At least once", where messages are assured to arrive but duplicates can occur.*

- *"Exactly once", where message are assured to arrive exactly once. This level could be used, for example, with billing systems where duplicate or lost messages could lead to incorrect charges being applied.*

Bibliografia:

https://mqtt.org

https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=mqtt

OASIS MQTT 3.1.1 specification document
http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

What is an MQTT Broker Clearly Explained
https://www.youtube.com/watch?v=WmKAWOVnwjE
Apresenta os conceitos e serviços disponibilizado pelo Broker MQTT (sem código exemplo)

An Introduction to MQTT for Beginners
https://www.youtube.com/watch?v=2aHV2Fn0I60&list=RDCMUCt12dTP8xvHM02Gc8HdGOcQ&index=2

Understanding The MQTT Protocol Packet Structure
https://www.youtube.com/watch?v=gLfmoptFgos

How to Use the Paho Python MQTT Client- (Beginners Guide)
https://www.youtube.com/watch?v=QAaXNt0oqSI&list=RDCMUCt12dTP8xvHM02Gc8HdGOcQ&index=3

10.1. Estrutura das mensagens MQTT

As mensagens MQTT têm um cabeçalho (Header) e os dados (Payload).

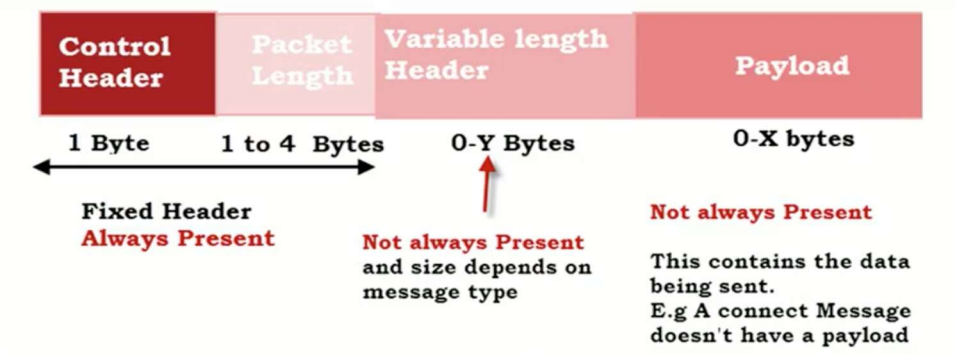


Fig - Estrutura das mensagens MQTT
(Steve Cope, https://www.youtube.com/watch?v=gLfmoptFgos)

Control Header
Este byte está organizado em dois conjuntos de 4 bits.

Os 4 bits mais significativos (da esquerda) identificam o tipo de mensagem, por exemplo:
"Connect 0001"
"ConnectAck 0010"
"Disconnect 1110"

Os 4 bits menos significativos do "control Header""(da direita), podem por exemplo identificar a Qualidade do Serviço pedido (QoS), numa mensagem do tipo "Publish", enviada para o Broker.

Packet Length
Este campo indica o comprimento, o número de bytes, do payload da mensagem (+ a parte variável do Header).
Podem ser usados até 4 bytes para representar o comprimento da mensagem.
Só são usados 7 bits de cada byte para representar o comprimento da mensagem.
Ou seja:

- com um byte podemos representar 2 ^7 = 128 números, em decimal de 0 a 127
- com dois bytes podemos representar 2^14 = 16384 números, em decimal de 0 a 16383
- com três bytes podemos representar 2^21 = 2 097 152 números, em decimal de 0 a 2 097 151
- com quatro bytes podemos representar 2^28 números

Modo confirmado entre o Cliente (Dispositivo) e o Servidor (Broker)

Connect
Connect Acknowledge

Publish
Publish Ack

Subscribe
Subscribe Ack

10.2. Instalar o Broker Mosquitto (MQTT)

1- descarregar o broker para windows (www.mosquitto.org)
e instalar na pasta "c:\program Files (x86)\mosquitto"
ou na pasta "c:\Program Files\mosquitto"

2- através de uma janela de DOS "cmd",
ative o Broker mosquito para ficar à escuta na porta TCP nº 1883
Edite o ficheiro mosquitto.conf disponível no directório onde instalou o mosquitto e insira duas linhas no ínicio:

```
listener 1883 0.0.0.0
allow_anonymous true
```

Depois, na linha de commandos, na janela de DOS, faça:

```
c:\program Files (x86)/mosquito > mosquitto -c mosquitto.conf -v
```

3- através de outra janela de DOS "cmd",
subscreva um tópico:

```
c:\program Files (x86)/mosquito > mosquitto_sub -h localhost  -p 1883 -t "temperatura"
```

4- através de outra janela DOS "cmd",
publique um tópico:

```
c:\program Files (x86)/mosquito > mosquitto_pub -h localhost  -p 1883 -t "temperatura" -m "45"
```

10.3. Instalar o Cliente Paho (cliente MQTT)

Pretende-se que o Cliente MQTT "Paho" consiga publicar e subscrever tópicos no servidor MQTT do exemplo anterior.
Faça o download do MQTT mosquito (from Eclipse), e instale o cliente MQTT "Paho" para publicar ou subscrever um tópico.

Instalar o Java Runtime Environment
https://www.oracle.com/java/technologies/javase-jre8-downloads.html
ou a partir do elearning:
https://elearning.ua.pt/mod/resource/view.php?id=830976

Instalar o Paho (Cliente MQTT)
https://www.eclipse.org/paho/index.php?page=components/tool/index.php
ou a partir do elearning:
https://elearning.ua.pt/mod/resource/view.php?id=830973

10.4. O ESP8266 como cliente MQTT

Pretende-se que o ESP consiga publicar e subscrever tópicos no broker MQTT,
sendo programado com o ambiente “Arduino IDE” e usando a biblioteca “PubSubClient”.

O exemplo seguinte apresenta um Cliente MQTT que “publica” o tópico “temperatura” no Broker.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

WiFiClient TCP_Client;                  // Objecto do tipo TCP
PubSubClient client(TCP_Client);        // Objecto do tipo cliente MQTT

void setup() {
  Serial.begin(115200);

  // WiFi network
  Serial.println(); Serial.print("Connecting to ");Serial.println("NOS-88C0");
  WiFi.mode(WIFI_STA);
  WiFi.begin("SSID", “password”);
  while (WiFi.status() != WL_CONNECTED) {
                    delay(500);  Serial.print(“.");
                    }

  // MQTT
  client.setServer("192.168.1.2", 1883);    // O Servidor MQTT, Broker, reside no pc “192.168.1.2”
  // client.setCallback(callback);
  client.connect("esp8266");                // O ESP regista-se no Broker, com o nome “esp8266”
}


void loop() {
  if (!client.connected()) {
          client.connect("esp8266");        // O ESP Regista-se no Broker com o nome esp8266
          }

  client.loop();
  Serial.println("Publish temperatura: 111 ");
  client.publish("temperatura", "111");     // Publica no Broker o tópico “temperatura” com o valor “111”
  delay(2000);
}
```

O exemplo seguinte apresenta um Cliente MQTT que "subscreve" o tópico "temperatura"

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

WiFiClient TCP_Client;
PubSubClient client(TCP_Client);


void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
        }

 Serial.println();
  // Switch on the LED if an 1 was received as first character
  if ((char)payload[0] == '1') {
        digitalWrite(2, LOW);   // Turn the LED on (Note that LOW is the voltage level
        }else{
        digitalWrite(2, HIGH);  // Turn the LED off by making the voltage HIGH
        }
}



void setup() {
  pinMode(2, OUTPUT);     // Initialize the BUILTIN_LED pin as an output
  Serial.begin(115200);

  // WiFi network
  Serial.println(); Serial.print("Connecting to ");Serial.println("NOS-88C0");
  WiFi.mode(WIFI_STA);
  WiFi.begin("NOS-88C0", "armagedao202");
  while (WiFi.status() != WL_CONNECTED) {delay(500); Serial.print(".");}
  Serial.println("");Serial.println("WiFi connected"); Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  client.setServer("192.168.1.2", 1883);
  client.setCallback(callback);
  client.connect("esp8266");
  Serial.println("Subscribe temperatura: ");
  client.subscribe("temperatura");
}



void loop() {
  if (!client.connected()) { client.connect("esp8266"); }
  client.loop();
  delay(1000);
}
```

Trab nº 7 – MQTT, módulo IOT como cliente MQTT, publica e subscreve tópicos

Iremos utilizar um módulo IOT que deve atuar como cliente MQTT, e um Broker Mosquitto que atuará como servidor MQTT.

Pretende-se que o módulo publique:

- SupEVin e SupEVout : o estado das electroválvulas de entrada e saída do reservatório (On/Off),

- SupMotor: o estado do motor/bomba de enchimento do reservatório (On/Off).

- SupVazio, SupMin,SupMax, SupTrans: o estado dos sensores de nível de água (Vazio, Min, Max, Transbordar). (On/Off quando deteta água passa ao estado On)

- SupNivel: o valor do nível de água (analógico, de 0 a 100%).

Pretende-se que o módulo subscreva os tópicos :

CtrMotor, CtrEVin, CtrEVout

Bibliografia:

MQTT - Como Comunicar 2 Dispositivos via Internet (Parte 1) - Vídeo #7
https://www.youtube.com/watch?v=nWXKUSiEPhY

MQTT - Como Comunicar 2 Dispositivos via Internet (parte 2) - Vídeo #8
https://www.youtube.com/watch?v=uXDRK6XNPNQ

MQTT - Como Comunicar 2 Dispositivos via Internet (parte 3) - Vídeo #9
https://www.youtube.com/watch?v=UkIt1wvq-Tk

#48 Connect ESP8266 with the world (and IFTT) through MQTT and Adafruit.io (Tutorial)
https://www.youtube.com/watch?v=9G-nMGcELG8

How to Get Started with MQTT
https://www.youtube.com/watch?v=tQmXWNd1pNk
Controlo da luz a partir de qualquer ponto do mundo.
Usa um cliente MQTT desenvolvido em Node-RED

MQTT Broker/Bridge on the ESP8266
https://www.youtube.com/watch?v=0K9q4IuB_oA
ESP publish and subscribe, Broker Mosquito (or uMQTT in ESP8266), Node-Red, sonoff como uBroker
uBroker : permite QoS=0, user name and password, wifi: STA ou AP,.
scripting language: OTA download, MQTT pub/sub, var, timers, NTP time, GPIO, console
(não apresenta código exemplo)

Test ESP8266 Grafana InfluxDB Node-RED MQTT Mosquitto IoT : PDAControl
https://www.youtube.com/watch?v=uJNqKF2gY7s

MQTT Protocol tutorial - LIVE DEMO using Mosquitto and CloudMQTT

https://www.youtube.com/watch?v=Oh3ZYAQBTko
MQTT sobre TCP , análise das mensagens, usa o Hercules cliente TCP/IP para …

Test ESP8266 Grafana InfluxDB Node-RED MQTT Mosquitto IoT : PDAControl
https://www.youtube.com/watch?v=uJNqKF2gY7s

****** MQTT ******
MQTT on ESP32 | Controlling Appliances and Monitoring ... - YouTube
https://www.youtube.com/watch?v=LvzCeBce2mU

How to Install Raspbian on Raspberry Pi 3 (5 min - ok)
https://www.youtube.com/watch?v=jsi50bCo_W4

Complete Installation Node-RED in Raspberry pi 3 Raspbian OS: PDAControl
https://www.youtube.com/watch?v=pLtZa5QeAMI

Node Red MQTT on the Raspberry Pi
https://www.youtube.com/watch?v=WxUTYzxIDns

Cheap MQTT Broker on Raspberry Zero W / DietPi / MQTT Message Logger SQlite / PHPliteAdmin
https://www.youtube.com/watch?v=bpg6RSHS4Zc

255 Node-Red, InfluxDB, and Grafana Tutorial on a Raspberry Pi
https://www.youtube.com/watch?v=JdV4x925au0