

# Instalación y configuración de MQTT-SN

Para comprender mejor qué es MQTT-SN, su arquitectura y sus aplicaciones se recomienda leer detenidamente la publicación [1]

## Instalación del broker

El primer paso para desplegar un escenario IoT con MQTT-SN es instalar su broker llamado *RSMB* (Real Small Message Broker). Su documentación y binarios se encuentran en [2]

IMPORTANTE: El broker *RSMB* no soporta QoS 2

En Linux, basta con hacer:

```
git clone https://github.com/eclipse/mosquitto.rsmb.git
```

Una vez se ha clonado el repositorio anterior, se ejecutan los siguientes comandos para asegurarnos de que tenemos las herramientas necesarias para compilar el proyecto:

```
sudo apt install make
```

```
sudo apt install build-essential
```

Se obtiene así una carpeta llamada *mosquitto.rsmb*. Dentro de ésta, debemos acceder al path *./rsmb/src*

```
ubuntu@ubuntuBroker:~/MQTT-SN/mosquitto.rsmb/rsmb/src$
```

Ahí ejecutamos el siguiente comando:

```
make
```

El output debería ser similar al mostrado en la siguiente imagen:

```
ubuntu@ubuntuBroker:~/MQTT-SN/mosquitto.rsmb/rsmb/src$ make
cc -Wall -Os *.c -o broker
cc -Wall -ggdb *.c -o broker_dbg
cc -DMQTT5 -Wall -Os *.c -o broker_mqtts
MQTT5Protocol.c: In function 'MQTT5Protocol_handleAdvertises':
MQTT5Protocol.c:273:9: warning: 'ftime' is deprecated: Use gettimeofday or clock
_gettime instead [-Wdeprecated-declarations]
  273 |         ftime(&ts);
      |         ^~~~~
In file included from MQTT5Protocol.c:39:
/usr/include/x86_64-linux-gnu/sys/timeb.h:29:12: note: declared here
   29 | extern int ftime (struct timeb *__timebuf)
      |         ^~~~~
perl tools/be/be.pl
```

De todos modos, si queremos comprobar si la operación ha sido exitosa, basta con ejecutar el comando *echo \$?*

Si el output de dicha acción es un cero (como en la siguiente figura) significa que la ejecución ha sido exitosa.

```
ubuntu@ubuntuBroker:~/MQTT-SN/mosquitto.rsmb/rsmb/src$ echo $?  
0
```

Listamos los ficheros dentro del path *mosquitto.rsmb/rsmb/src* y deberíamos de apreciar la presencia de dos nuevos ficheros : *broker* y *broker\_mqtts*

```
ubuntu@ubuntuBroker:~/MQTT-SN/mosquitto.rsmb/rsmb/src$ ls  
Bridge.c      Log.h          MQTTSClient    Socket.c  
Bridge.h      Makefile       MQTTSPacket.c  Socket.h  
broker        Messages.1.3.0.2 MQTTSPacket.h  StackTrace.c  
Broker.c      Messages.c     MQTTSPacketSerialize.c StackTrace.h  
broker_dbg    Messages.h     MQTTSPacketSerialize.h SubsEngine.c  
Broker.h      Messages.txt   MQTTSProtocol.c SubsEngine.h  
broker_mqtts  MQTTPacket.c  MQTTSProtocol.h tools  
Clients.c     MQTTPacket.h  MQTTSProtocolOut.c Topics.c  
Clients.h     MQTTPacketOut.c MQTTSProtocolOut.h Topics.h  
Filter.c      MQTTPacketOut.h Persistence.c   Tree.c  
Filter.h      MQTTProtocol.c Persistence.h   Tree.h  
Heap.c        MQTTProtocolClient.c Protocol.c      Users.c  
Heap.h        MQTTProtocolClient.h Protocol.h      Users.h  
LinkedList.c  MQTTProtocol.h rsmb.ini  
LinkedList.h  MQTTProtocolOut.c SocketBuffer.c  
Log.c         MQTTProtocolOut.h SocketBuffer.h
```

El primer fichero es un broker de MQTT sobre TCP (básicamente *mosquitto*) mientras que el segundo es la implementación del broker RSMB para MQTT-SN

Para iniciar el broker RSMB es importante crear un archivo de configuración que llamaremos *rsmb\_conf.txt* y cuyo contenido será:

```
trace_output protocol  
listener 1884 INADDR_ANY mqtts
```

Es importante que el puerto no sea el mismo que se usa en el broker *mosquitto* de MQTT si se está instalando el broker RSMB en el mismo equipo ya que podría haber conflicto.

Ejecutamos el siguiente comando para iniciar el broker RSMB con la configuración anterior:

```
./broker_mqtts rsmb_conf.txt
```

El output debería ser similar al de la siguiente figura:

```
ubuntu@ubuntuBroker:~/MQTT-SN/mosquitto.rsmb/rsmb/src$ ./broker_mqtts rsmb_conf.txt
20230320 090810.328 CWNAN9999I Really Small Message Broker
20230320 090810.329 CWNAN9998I Part of Project Mosquitto in Eclipse
(http://projects.eclipse.org/projects/technology.mosquitto)
20230320 090810.329 CWNAN0049I Configuration file name is rsmb_conf.txt
20230320 090810.329 CWNAN0053I Version 1.3.0.2, Mar 19 2023 17:35:48
20230320 090810.329 CWNAN0054I Features included: bridge MQTTS
20230320 090810.329 CWNAN9993I Authors: Ian Craggs (icraggs@uk.ibm.com), Nicholas O'Leary
20230320 090810.329 CWNAN0300I MQTT-S protocol starting, listening on port 1884
```

## Instalación del cliente

La librería del cliente MQTT-SN viene ya incluida en este repositorio en la carpeta *mqttsnclient*. Aún así, se recomienda visitar [3] para comprender cómo se instala y cuál es su funcionamiento.

El fichero *publisher.py* contiene el código necesario para entender la publicación de mensajes mientras que el fichero *subscriber.py* contiene el código para la subscripción. Ambos programas son una buena toma de contacto para estudiar el protocolo y construir escenarios más complejos a partir de ellos.

## Enlaces

- [1] <http://www.steves-internet-guide.com/mqtt-sn/>
- [2] <http://www.steves-internet-guide.com/mqtt-sn-rsmb-install/>
- [3] <http://www.steves-internet-guide.com/python-mqttsn-client/>