ST87MXX MQTT Application Note

Introduction

This document details the usage of AT Commands for the embedded MQTT stack supported by the ST87MXX NB-IoT module.

Document status
Official

1 General information

1.1 Acronyms and terms

Table 1. Definitions of terms

Term	Definition
DNS	Domain Name System
IP	Internet Protocol
JTAG	Joint Test Action Group
MBR	Maximum Bit Rate
ME	Mobile Equipment
MQTT	Message Queuing Telemetry Transport
NCP	Network Control Protocol
PDP	Packet Data Protocol
PPP	Point-to-Point Protocol
QCI	QoS Class Identifier
QoS	Quality of Service
TA	Terminal Adaptor, e.g. a GSM data card (equal to DCE; Data Circuit terminating Equipment)
TCP	Transport Control Protocol
TE	Terminal Equipment, e.g. a computer (equal to DTE; Data Terminal Equipment)
TFT	Traffic Flow Template
UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol

1.2 Reference documents

The documents listed in Table 2 provide further information.

Table 2. Document references

Reference	Document
[1]	ST87MXX UM AT commands description
[2]	ST87MXX TCP/UDP/IP Application Note

1.3 Revision history

Table 3. Document revision history

Date	Version	Changes
2023-12-08	V1.0	Official release
2024-03-26	V1.1	Update of socket interface Update of Errors codes and URC list
2024-10-10	V1.2	Update of Errors codes and URC list
2025-01-08	V1.3	Update Socket Creation and remove Error and URC tables
2025-07-01	V1.4	Remove AT cmds tables and add ST87MXX UM AT commands description reference
2025-08-28	V2.0	Update for release

1.4 Table of contents

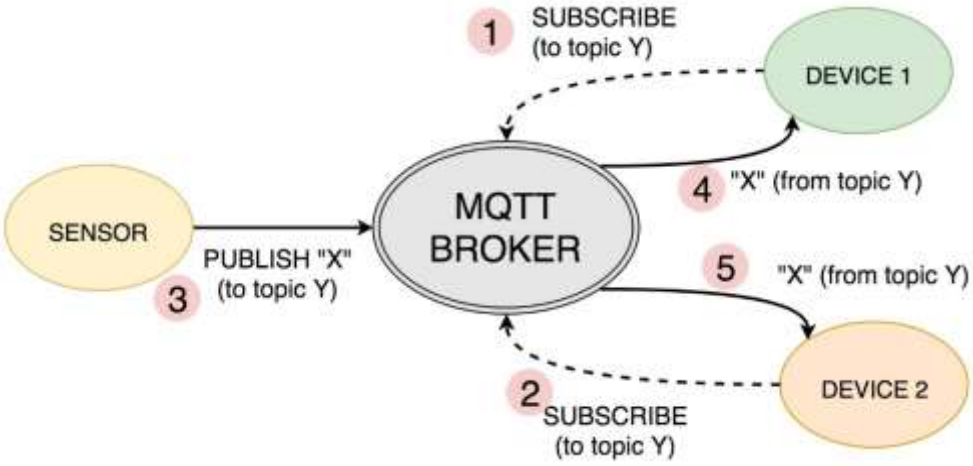
1.5 List of tables

2 Introduction

2.1 MQTT overview

The MQTT protocol is a machine-to-machine (M2M) protocol widely used in IoT. It is a message-based protocol. It is extremely lightweight, and for this reason, it has seen wide adoption in the IoT ecosystem. Almost all IoT cloud platforms support MQTT to send and receive data from smart objects.
The protocol is a set of rules that defines how IoT devices can publish and subscribe to data over the Internet.

The protocol is event-driven and connects devices using the publish/subscribe (Pub/Sub) pattern. The sender (Publisher) and the receiver (Subscriber) communicate via Topics and are decoupled from each other. The connection between them is handled by the MQTT broker. The MQTT broker filters all incoming messages and distributes them correctly to the Subscribers.



3 MQTT Configuration

3.1 Preamble

To set up an MQTT connection, the ME must first be attached (PDP context connection established) to an NB-IoT network.
This is notified by the message: +CGEV: ME PDN ACT 0 to the Host.

3.2 MQTT configuration

The MQTT configuration is linked to the MQTT client configuration before connecting this client to the server.
An AT command is used to configure the MQTT client.
This command configures the MQTT protocol stack before use.
If this command is not called, the parameters saved in NVM are used.

AT#MQTTCFG=[<client_name>],[<connexion_timeout>],[<protocol_timeout>],[<publish_retry>],[<keep_alive_pub_msg>]

By default, the following parameters are used if they are not provided by AT cmd:

Parameters	Value
Client Name	mqtt-st8700-client
MQTT_CONNEXION_TIMEOUT	20 seconds
MQTT_PROTOCOL_TIMEOUT	20 seconds
MQTT_PUBLISH_RETRY	10 times
MQTT_KEEP_ALIVE_PUBLISH	10 seconds

An example of configuration:

```
AT#MQTTCFG=mqtt-st8700-client,20,20,10,10
OK
```

4 MQTT Sockets

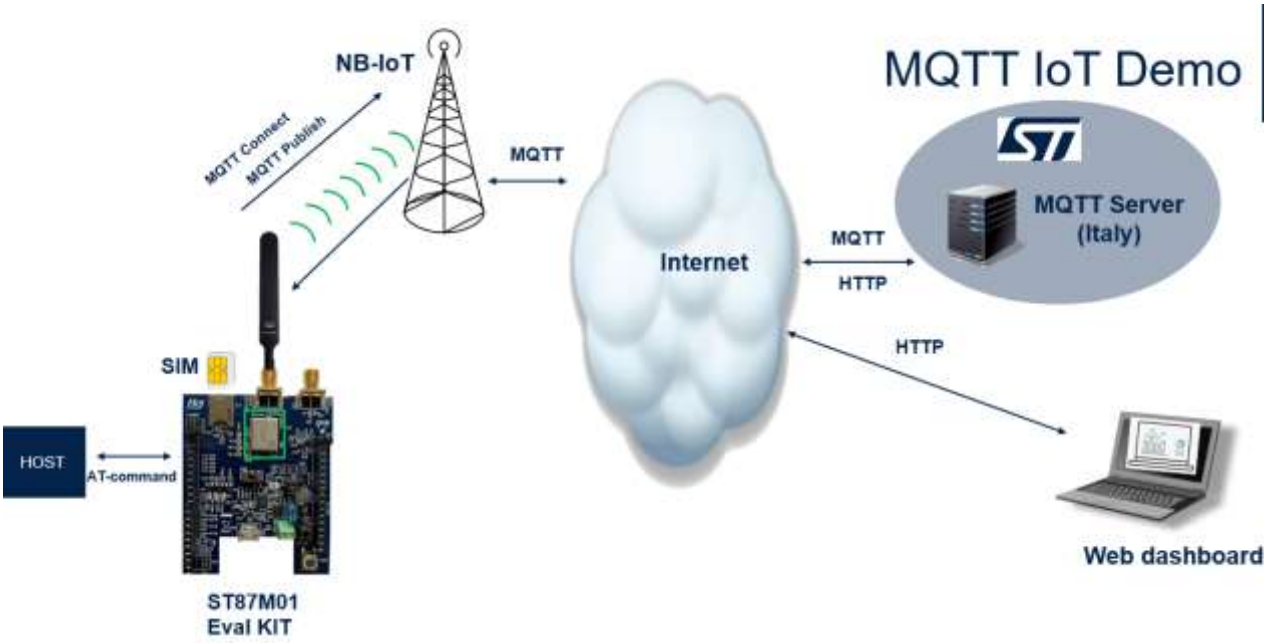
4.1 Preamble

For MQTT connection and for MQTT packets exchange, the ME shall first set the IP layer.
See ST87MXX_TCP_UDP_IP_Application_Note document to setup the IP layer [2].

Now, the ME can create a MQTT connection.

It is assumed that the customer testing this MQTT functionality has a MQTT server (Broker) running correctly inside the Internet network or inside their own network.

Application Note following example is based on this infrastructure:



4.2 Create MQTT socket

To exchange, MQTT packets, we need first to create a socket.
It will be used to exchange data.
Most of time the MQTT protocol is based on TCP/IP protocol.

The ME can create socket with the following AT command:

AT#SOCKETCREATE=<context_id>,<ip_version>,<socket_type>,<local_port>,<send_timeout>,<receive_timeout>[,<frame_received_urc>,<security_profile_id>]

Example of IPV4 TCP Socket Creation without port and URC RECV disabled:

AT#SOCKETCREATE=5,0,TCP,1883,10,10

5: is the context ID used for IP connection: 5 it is the one used by default
0: for IPV4 , **1** :for IPV6 it should be the same chosen during IP configuration
TCP is the type of socket used to create it.
1883: TCP local port chosen by customer (Ex: Most used for MQTT service: **1883**)
10 is the timeout set in seconds to send to the server the request.
10 is the timeout set in seconds to send and receive the response from the server.

#SOCKETCREATE: 0 -----> 0 is the socket id delivered by IP stack

OK

Be Careful: Please ensure that the Timeout values used are large enough for communication delay between device and broker server.

4.3 Connect MQTT socket

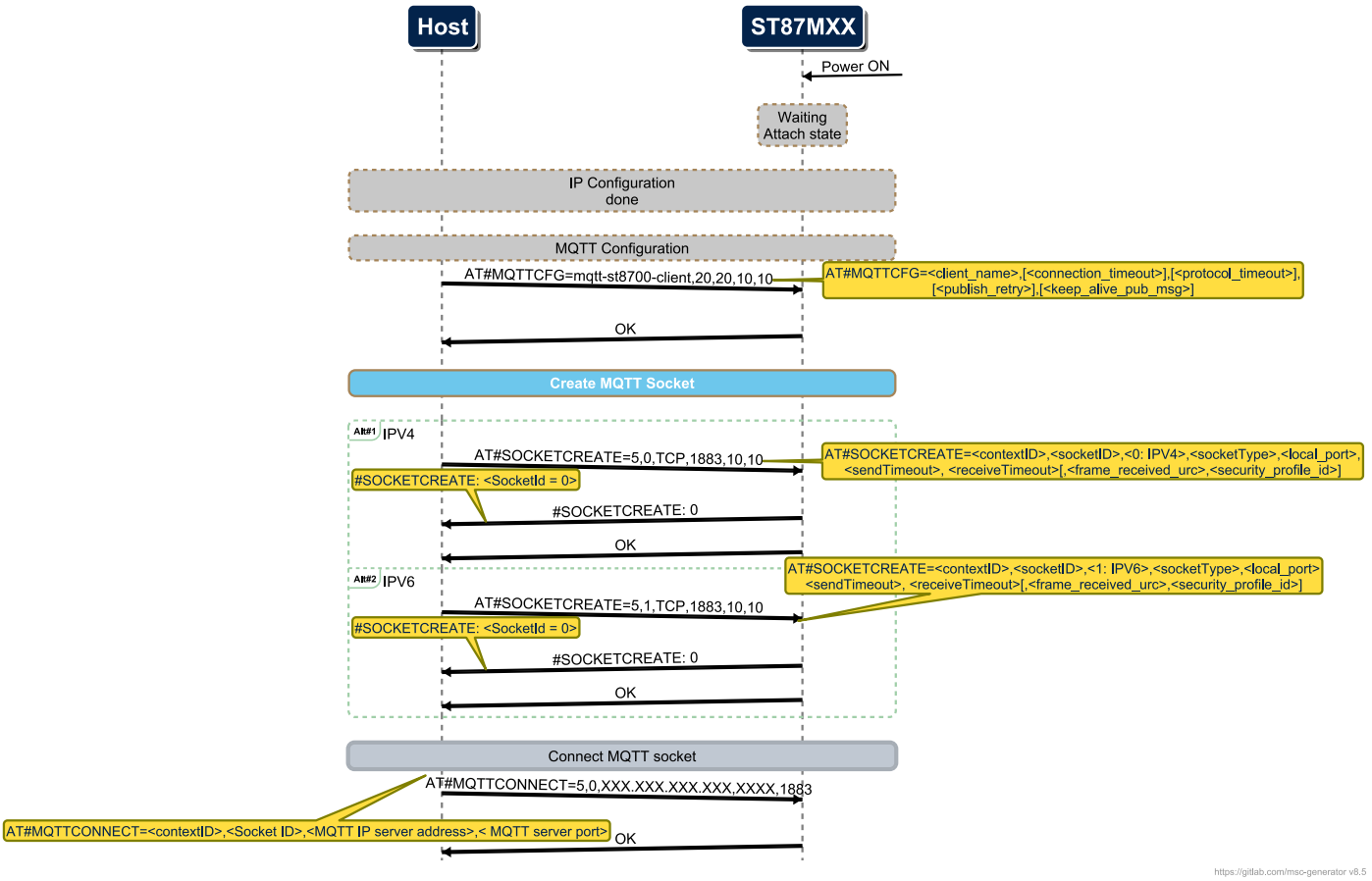
For example, after the creation of the socket, the ME can connect the MQTT layer to the remote server with the following command:

AT#MQTTCONNECT=5,0,<MQTT IP Remote Server address>,<TCP Remote Server port>

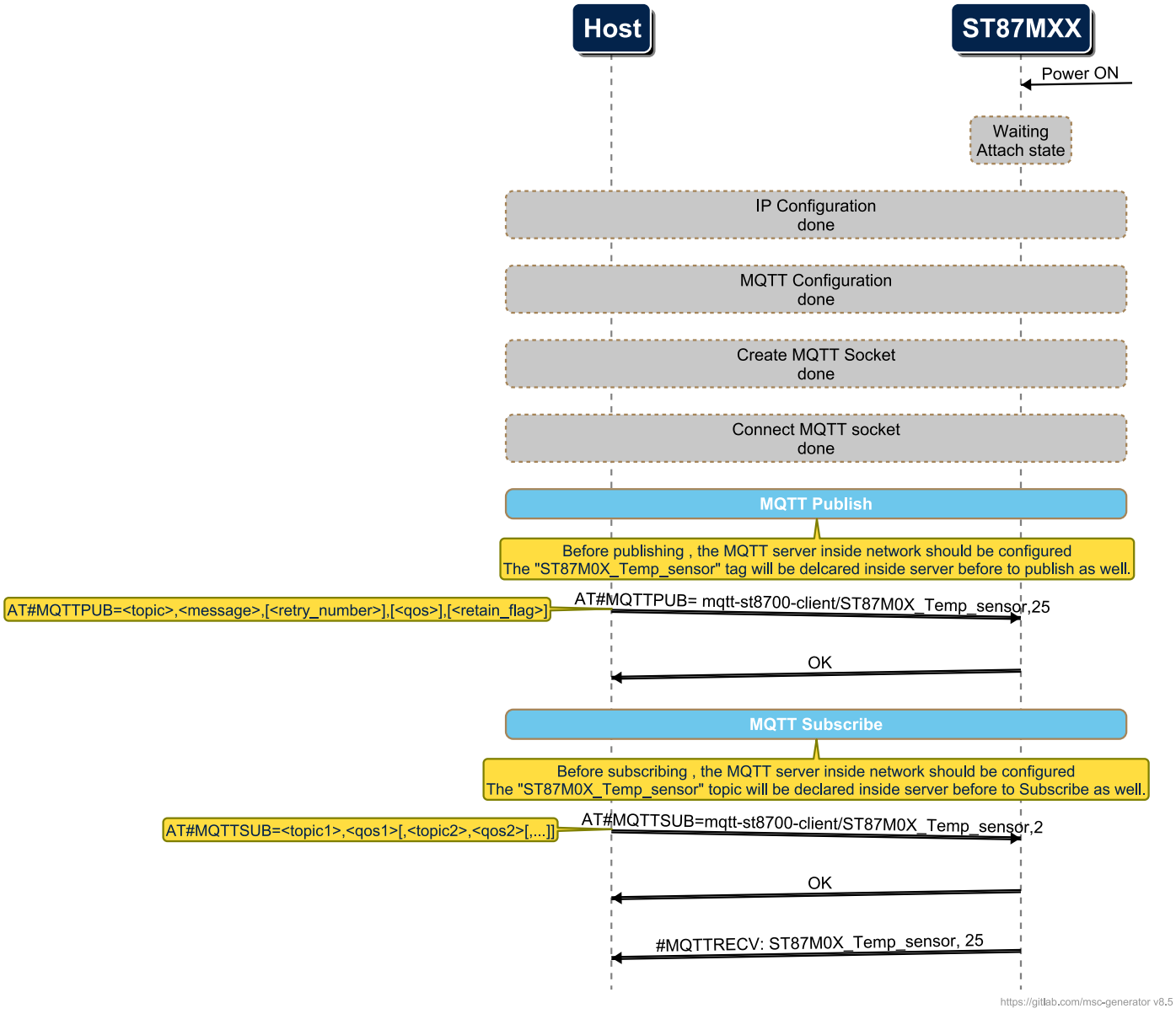
5: is the context ID used for IP connection.
0: is the socket ID used for IP connection and received when socket was created

The MQTT IP Remote Server address and the TCP Remote Server port parameters will be the parameters of customer MQTT broker server.

4.4 MSC for MQTT Configuration and Connection



4.1 MSC for MQTT publish and Subscribe



IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.
Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of
Purchasers’ products.
No license, express or implied, to any intellectual property right is granted by ST herein.
Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.
ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.
Information in this document supersedes and replaces information previously supplied in any prior versions of this document.