QSTM32
MQTT(S) Application Note

Confidentiality Level: (Tick the Box ■)		
Top Secret ☐	Confidential ☐	Public ☐

Document Control Records

Revision History			
Date	Revision	Revision Description	Author
2024-2-7	0	Initial	Zoran Romic
2025-08-29	V2.0	The second generation initial version	Wells Li Linkin Wang

Contents

1 Purpose

MQTT (Message Queuing Telemetry Transport) is a broker-based publish/subscribe messaging protocol designed to be open, simple, lightweight and easy to implement. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited.
This document introduces how to use the MQTT function on quectel module via AT commands.

2 Scope

This document applies to products with MCU mounted with quectel module.

3 Terms and Definitions

Quectel: Quectel Wireless Solutions Co., Ltd.
MQTT: Message Queuing Telemetry Transport
UE: User Equipment
SSL: Secure Socket Layer
CA: Certificate Authority

4 MQTT AT Commands and corresponding API

AT commands	**API functions**	**Functionality**
AT+QMTCFG="WILL"	ql_mqtt_setopt()	**Configure the will flag**
AT+QMTCFG="TIMEOUT"	ql_mqtt_setopt()	**Configure timeout message report**
AT+QMTCFG="SESSION"	ql_mqtt_setopt()	**Configure of storing info about the client**
AT+QMTCFG="KEEPALIVE"	ql_mqtt_setopt()	**Time after when server will disconnect a client**
AT+QMTCFG="recv/mode"	ql_mqtt_setopt()	**Enable URC**
AT+QMTCFG="SSL"	ql_mqtt_setopt()	**Configure the SSL**
AT+QSSLCFG="CACERT"	ql_mqtt_set_ssl()	**Set server certification**
AT+QSSLCFG="CLIENTCERT"	ql_mqtt_set_ssl()	**Set client certification**
AT+QSSLCFG="CLIENTKEY"	ql_mqtt_set_ssl()	**Set client key**
AT+QSSLCFG="SSLVERSION"	ql_mqtt_set_ssl()	**Configure SSL version**
AT+QSSLCFG="CIPHERSUITE"	ql_mqtt_set_ssl()	**Configure SSL chipper suites**
AT+QSSLCFG=" SECLEVEL"	ql_mqtt_set_ssl()	**Configure SSL authentication mode**
AT+QSSLCFG="IGNORELOCALTIME"	ql_mqtt_set_ssl()	**Ignore the time of authentication**
AT+QMTOPEN	ql_mqtt_connect()	**Open network connection for MQTT client**
AT+QMTCONN	ql_mqtt_connect()	**Request a connection to MQTT server**
AT+QMTSUB	ql_mqtt_sub()	**Subscribe one or more specific topics**
AT+QMTUNS	ql_mqtt_unsub()	**Un-subscribe from a specific topic**
AT+QMTPUBEX/ AT+QMTPUB	ql_mqtt_pub()	**Publish messages to a server for specific topic**
AT+QMTDISC	ql_mqtt_disconnect()	**Disconnect from MQTT server**
AT+QMTCLOSE	ql_mqtt_disconnect()	**Close a network for MQTT client**

5 MQTT Application Work Flow

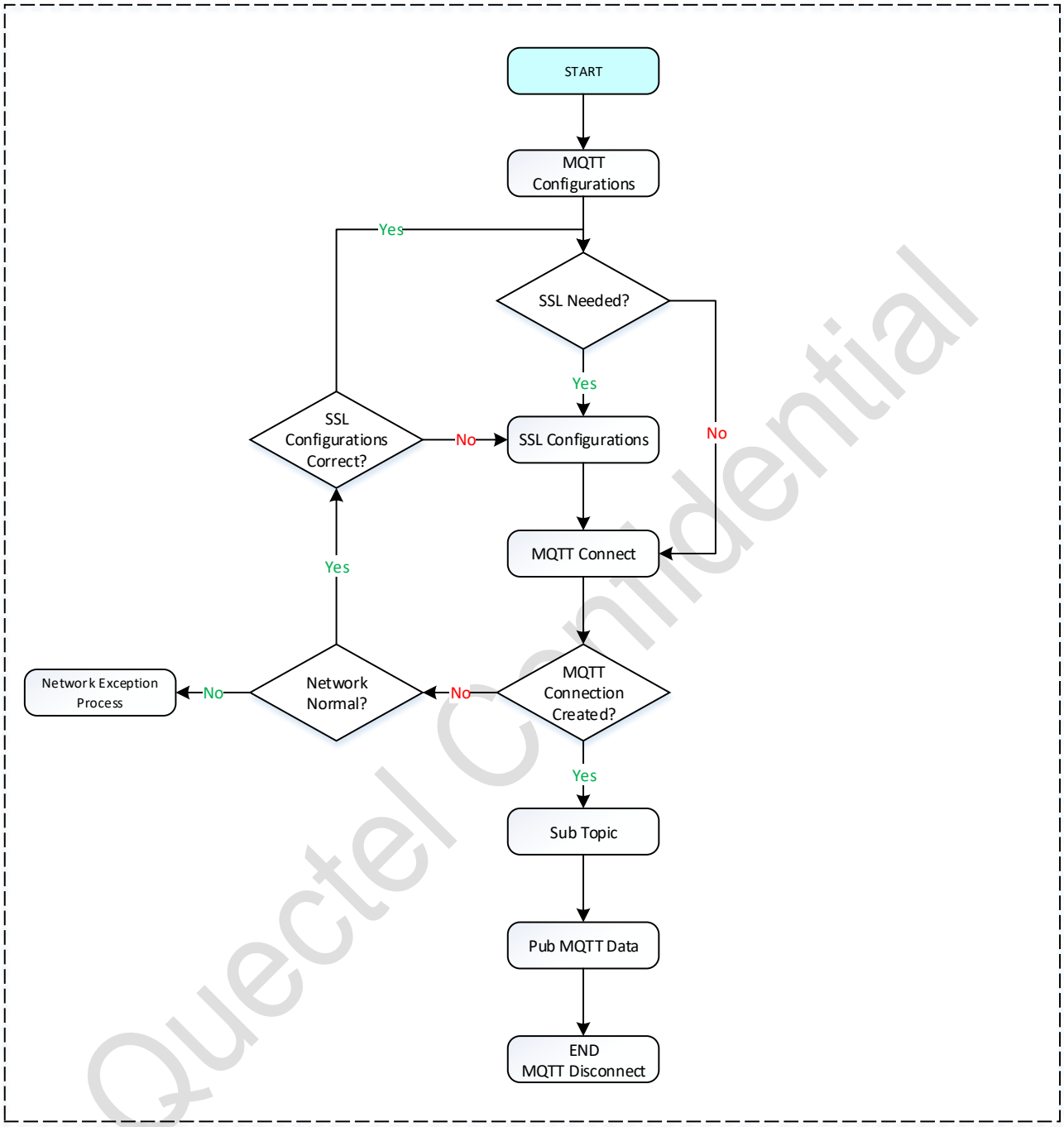

Figure 1: MQTT Application Workflow (1)

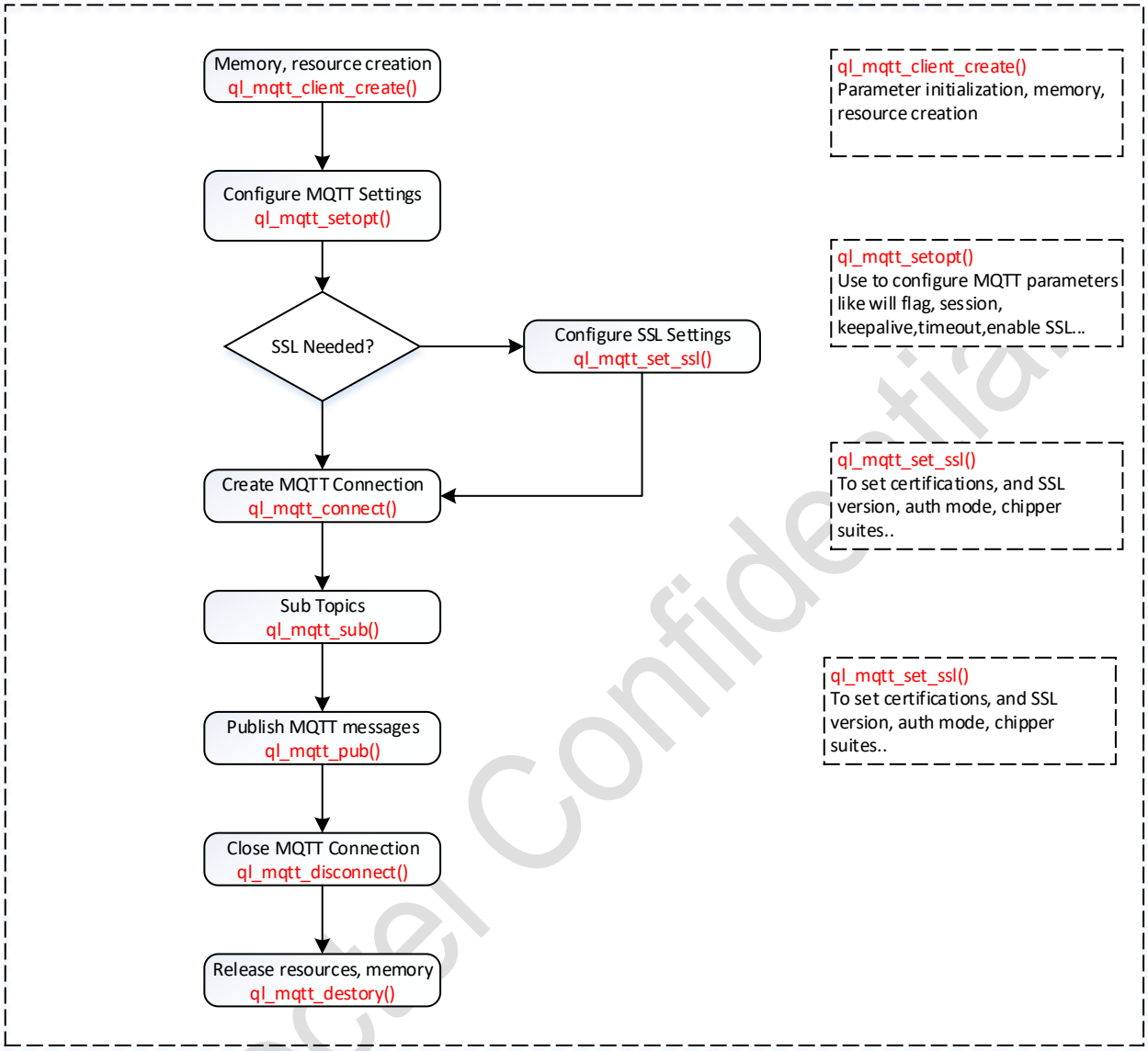

Figure 2: MQTT Application Workflow (2)

Here is description for MQTT workflow:

a) Call *ql_mqtt_setopt()* to configure MQTT configuration parameters, including will, timeout, session, and keepalive.

b) If SSL is needed, when accessing to platform such as amazon.aws.iot, it is necessary to call *ql_mqtt_set_ssl()* to configure a correct SSL context ID for SSL encryption, a suitable SSL version which matched with remote server, and a suitable cipher suite which matched with remote server. Additionally, it is needed to configure no authentication or perform server authentication (one-way authentication) or server and client authentication if requested by remote server (mutual authentication).

c) Call *ql_mqtt_connect()* to open network connection for MQTT client.
d) Call *ql_mqtt_connect()* to request a connection to MQTT server for a client.
e) Call *ql_mqtt_sub()* to subscribe one or more specific topics.
f) Call *ql_mqtt_pub()* to publish messages to a server for the specific topic.
g) Call *ql_mqtt_unsub()* to unsubscribe from specific topic.
h) Call *ql_mqtt_disconnect()* to disconnect from MQTT server.

6 MQTT with SSL Configuration Diagram

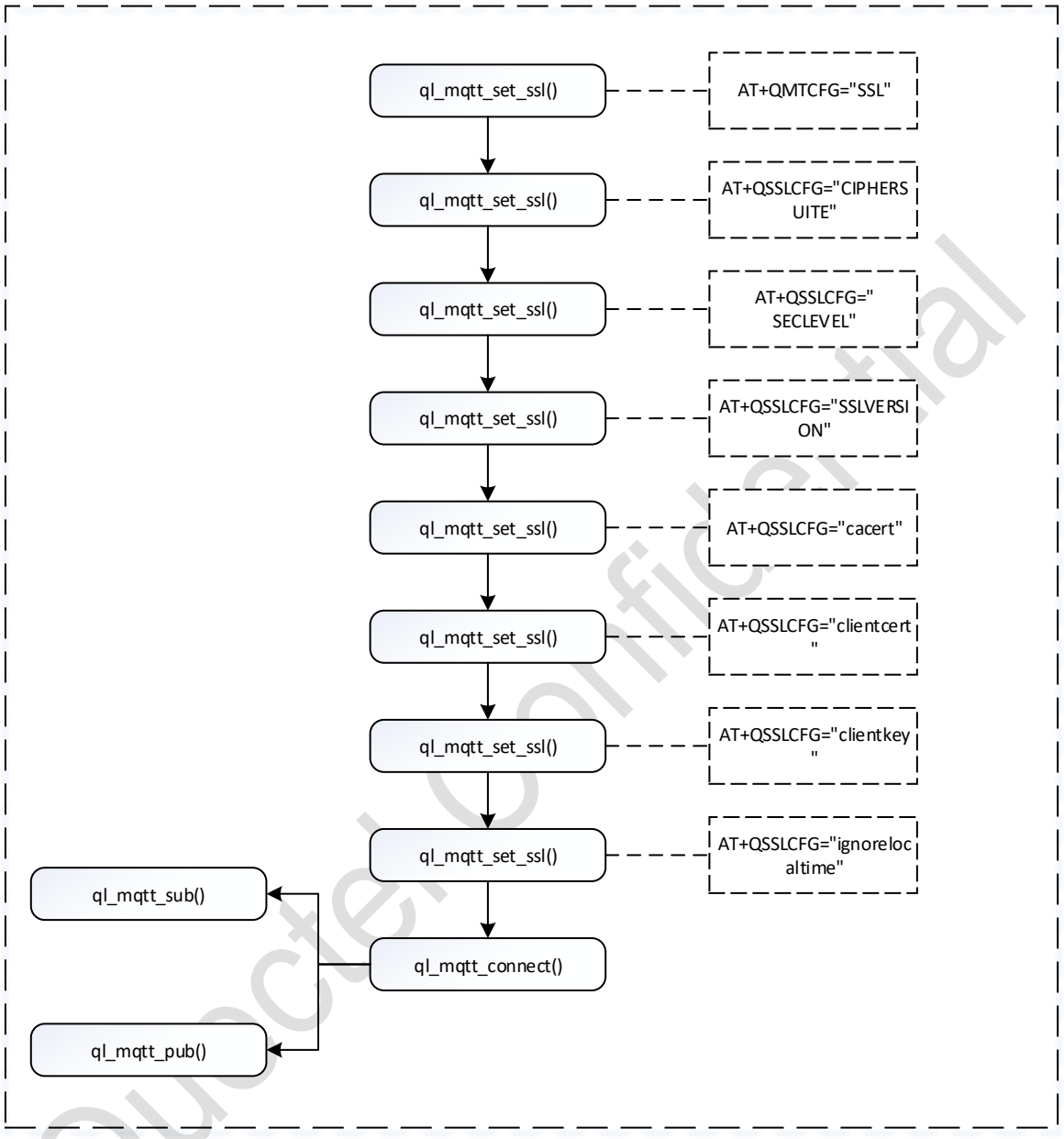

Figure 3: MQTT with SSL Configuration Diagram

8 Appendix A References

[1] Quectel_QSTM32_Network_Application_Note_V2.0
[2] Quectel_QSTM32_SDK_API_Design_V2.0