

# **BG95&BG77&BG600L Series**

## **MQTT Application Note**

**LPWA Module Series**

Rev. BG95&BG77&BG600L\_Series\_MQTT\_Application\_Note\_V1.1

Date: 2020-05-28

Status: Released



**Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:**

**Quectel Wireless Solutions Co., Ltd.**

Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China.

Tel: +86 21 5108 6236

Email: [info@quectel.com](mailto:info@quectel.com)

**Or our local office. For more information, please visit:**

<http://www.quectel.com/support/sales.htm>

**For technical support, or to report documentation errors, please visit:**

<http://www.quectel.com/support/technical.htm>

Or email to: [support@quectel.com](mailto:support@quectel.com)

**GENERAL NOTES**

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

**COPYRIGHT**

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT WITHOUT PERMISSION ARE FORBIDDEN. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

***Copyright © Quectel Wireless Solutions Co., Ltd. 2020. All rights reserved.***

# About the Document

## Revision History

Version	Date	Author	Description
1.0	2019-08-08	Lane HAO	Initial
1.1	2020-05-28	Jaryoung LI	<ol style="list-style-type: none"><li>1. Added an applicable module BG600L-M3.</li><li>2. Updated the maximum length of messages to be published into 4096 bytes (&lt;msglen&gt; of AT+QMTPUB).</li><li>3. Deleted AT+QMTCFG="prefix".</li><li>4. Updated the error response of the AT commands in Chapter 3.2.</li><li>5. Deleted the summary of error codes.</li></ol>

## Contents

About the Document .....	2
Contents .....	3
Table Index.....	4
<b>1 Introduction .....</b>	<b>5</b>
1.1. Applicable Modules.....	5
<b>2 MQTT Data Interaction.....</b>	<b>6</b>
<b>3 MQTT Related AT Commands .....</b>	<b>7</b>
3.1. AT Command Syntax .....	7
3.1.1. Definitions.....	7
3.1.2. AT Command Syntax .....	7
3.2. Description of MQTT Related AT Commands.....	8
3.2.1. AT+QMTCFG Configure Optional Parameters of MQTT .....	8
3.2.2. AT+QMTOPEN Open a Network for MQTT Client.....	13
3.2.3. AT+QMTCLOSE Close a Network for MQTT Client .....	13
3.2.4. AT+QMTCONN Connect a Client to MQTT Server.....	14
3.2.5. AT+QMTDISC Disconnect a Client from MQTT Server.....	16
3.2.6. AT+QMTSUB Subscribe to Topics .....	16
3.2.7. AT+QMTUNS Unsubscribe from Topics.....	17
3.2.8. AT+QMTPUB Publish Messages .....	18
3.2.9. AT+QMTPUBEX Publish Messages .....	20
3.2.10. AT+QMTRECV Read Messages from Buffers .....	22
<b>4 MQTT Related URCs .....</b>	<b>24</b>
4.1. +QMTSTAT: URC to Indicate State Change in MQTT Link Layer .....	24
4.2. +QMTRECV: URC to Inform the Host to Read MQTT Packet Data.....	25
<b>5 Examples .....</b>	<b>26</b>
5.1. Example of MQTT Operation without SSL .....	26
5.2. Example of MQTT Operation with SSL.....	28
<b>6 Appendix A References.....</b>	<b>30</b>

## Table Index

Table 1: MQTT Related URCs .....	24
Table 2: Error Codes of +QMTSTAT: URC.....	24
Table 3: Related Documents .....	30
Table 4: Terms and Abbreviations .....	30

# 1 Introduction

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 mainly introduces how to use the MQTT function of Quectel BG95 series, BG77 and BG600L-M3 modules through AT commands.

## 1.1. Applicable Modules

**Table 1: Applicable Modules**

Module Series	Model	Description
<b>BG95</b>	BG95-M1	Cat M1 only
	BG95-M2	Cat M1/Cat NB2
	BG95-M3	Cat M1/Cat NB2/EGPRS
	BG95-N1	Cat NB2 Only
	BG95-M4	Cat M1/Cat NB2, 450 MHz Supported
	BG95-M5	Cat M1/Cat NB2/EGPRS, Power Class 3
	BG95-M6	Cat M1/Cat NB2, Power Class 3
	BG95-MF	Cat M1/Cat NB2, Wi-Fi Positioning
<b>BG77</b>	BG77	Cat M1/Cat NB2
<b>BG600L</b>	BG600L-M3	Cat M1/Cat NB2/EGPRS

# 2 MQTT Data Interaction

This chapter gives the data interaction mechanism of MQTT function.

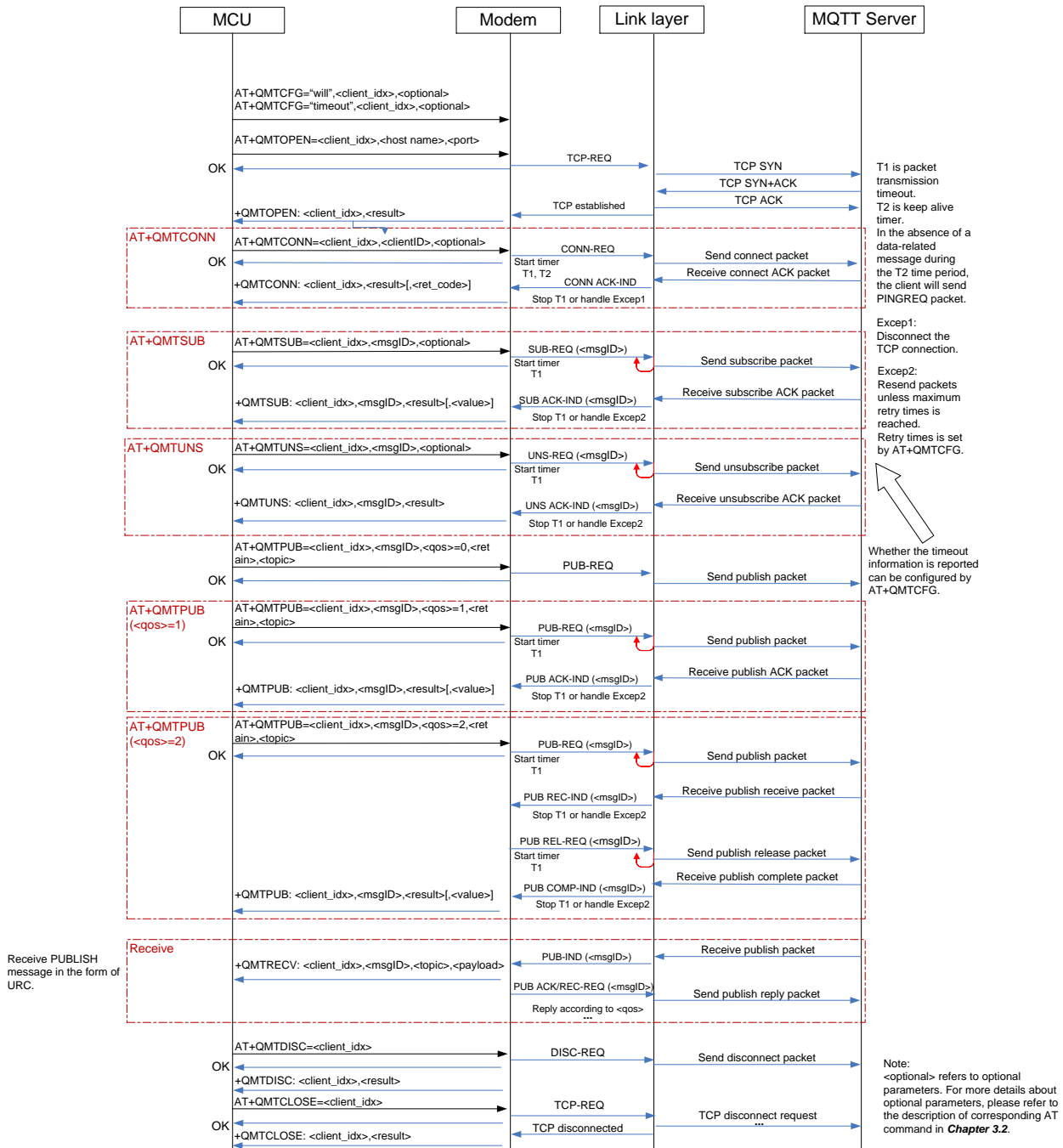


Figure 1: MQTT Data Interaction Diagram

# 3 MQTT Related AT Commands

This chapter presents the AT commands for operating MQTT function.

## 3.1. AT Command Syntax

### 3.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on command line. When an optional parameter is omitted, the new value equals its previous value or its default setting, unless otherwise specified.
- **Underline** Default setting of a parameter.

### 3.1.2. AT Command Syntax

The **AT** or **at** prefix must be added at the beginning of each command line. Entering **<CR>** will terminate a command line. Commands are usually followed by a response that includes **<CR><LF><response><CR><LF>**. Throughout this document, only the response **<response>** will be presented, **<CR><LF>** are omitted intentionally.

**Table 2: Type of AT Commands and Responses**

<b>Test Command</b>	<b>AT+&lt;cmd&gt;=?</b>	This command returns the list of parameters and value ranges set by the corresponding Write Command or internal processes.
<b>Read Command</b>	<b>AT+&lt;cmd&gt;?</b>	This command returns the currently set value of the parameter or parameters.
<b>Write Command</b>	<b>AT+&lt;cmd&gt;=&lt;p1&gt; [,&lt;p2&gt;[,&lt;p3&gt;[...]]]</b>	This command sets the user-definable parameter values.
<b>Execution Command</b>	<b>AT+&lt;cmd&gt;</b>	This command reads non-variable parameters affected by internal processes in the module.



## 3.2. Description of MQTT Related AT Commands

### 3.2.1. AT+QMTCFG Configure Optional Parameters of MQTT

This command configures optional parameters of MQTT.

AT+QMTCFG Configure Optional Parameters of MQTT	
<p>Test Command</p> <p>AT+QMTCFG=?</p>	<p>Response</p> <p>+QMTCFG: "version",(range of supported &lt;client_idx&gt;s),(list of supported &lt;vsns&gt;s)</p> <p>+QMTCFG: "pdpcid",(range of supported &lt;client_idx&gt;s),(range of supported &lt;cid&gt;s)</p> <p>+QMTCFG: "ssl",(range of supported &lt;client_idx&gt;s),( list of supported &lt;SSL_enable&gt;s),(range of supported &lt;ctx_index&gt;s)</p> <p>+QMTCFG: "keepalive",(range of supported &lt;client_idx&gt;s,(range of supported &lt;keep_alive_time&gt;s)</p> <p>+QMTCFG: "session",(range of supported &lt;client_idx&gt;s),(list of supported &lt;clean_session&gt;s)</p> <p>+QMTCFG: "timeout",(range of supported &lt;client_idx&gt;s),(range of supported &lt;pkt_timeout&gt;s),(range of supported &lt;retry_times&gt;s),(list of supported &lt;timeout_notice&gt;s)</p> <p>+QMTCFG: "will",(range of supported &lt;client_idx&gt;s),(list of supported &lt;will_fg&gt;s),(range of supported &lt;will_qos&gt;s),(list of supported &lt;will_retain&gt;s),&lt;will_topic&gt;,&lt;will_message&gt;</p> <p>+QMTCFG: "recv/mode",(range of supported &lt;client_idx&gt;s),(list of supported &lt;msg_recv_mode&gt;s),(list of supported &lt;msg_len_enable&gt;s)</p> <p>+QMTCFG: "aliauth",(range of supported &lt;client_idx&gt;s),&lt;product_key&gt;,&lt;device_name&gt;,&lt;device_secret&gt;</p> <p>OK</p>
<p>Write Command</p> <p>Query/set the MQTT protocol version</p> <p>AT+QMTCFG="version",&lt;client_idx&gt;[,&lt;vsns&gt;]</p>	<p>Response</p> <p>If the optional parameter is omitted, query the MQTT protocol version:</p> <p>+QMTCFG: "version",&lt;vsns&gt;</p> <p>OK</p> <p>If the optional parameter is specified, set the MQTT protocol version:</p> <p>OK</p>

	<p>If there is any error: <b>ERROR</b></p>
<p>Write Command Query/set the PDP to be used by the MQTT client <b>AT+QMTCFG="pdpcid",&lt;client_id x&gt;[,&lt;cid&gt;]</b></p>	<p>Response If the optional parameter is omitted, query the PDP used by the MQTT client: <b>+QMTCFG: "pdpcid",&lt;cid&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameter is specified, set the PDP to be used by the MQTT client: <b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command Query/set the MQTT SSL mode and SSL context index <b>AT+QMTCFG="ssl",&lt;client_idx&gt;[,&lt;SSL_enable&gt;[,&lt;ctx_index&gt;]]</b></p>	<p>Response If the optional parameters are omitted, query the MQTT SSL mode and SSL context index: <b>+QMTCFG: "ssl",&lt;SSL_enable&gt;[,&lt;ctx_index&gt;]</b></p> <p><b>OK</b></p> <p>If any of the optional parameters is specified, set the MQTT SSL mode and SSL context index: <b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command Query/set the keep-alive time <b>AT+QMTCFG="keepalive",&lt;client_idx&gt;[,&lt;keep_alive_time&gt;]</b></p>	<p>Response If the optional parameter is omitted, query the keep-alive time: <b>+QMTCFG: "keepalive",&lt;keep_alive_time&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameter is specified, set the keep-alive time: <b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command Query/set the session type <b>AT+QMTCFG="session",&lt;client_id x&gt;[,&lt;clean_session&gt;]</b></p>	<p>Response If the optional parameter is omitted, query the session type: <b>+QMTCFG: "session",&lt;clean_session&gt;</b></p> <p><b>OK</b></p>

	<p>If the optional parameter is specified, set the session type: <b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command Query/set the timeout of message delivery <b>AT+QMTCFG="timeout",&lt;client_idx&gt;[,&lt;pkt_timeout&gt;[,&lt;retry_times&gt;][,&lt;timeout_notice&gt;]]</b></p>	<p>Response If the optional parameters are omitted, query the timeout of message delivery: <b>+QMTCFG: "timeout",&lt;pkt_timeout&gt;,&lt;retry_times&gt;,&lt;timeout_notice&gt;</b></p> <p><b>OK</b></p> <p>If any of the optional parameters is specified, set the timeout of message delivery: <b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command Query/set the Will information <b>AT+QMTCFG="will",&lt;client_idx&gt;[,&lt;will_fg&gt;[,&lt;will_qos&gt;,&lt;will_retain&gt;,&lt;will_topic&gt;,&lt;will_message&gt;]]</b></p>	<p>Response If the optional parameters are omitted, query the Will information: <b>+QMTCFG: "will",&lt;will_fg&gt;[,&lt;will_qos&gt;,&lt;will_retain&gt;,&lt;will_topic&gt;,&lt;will_message&gt;]</b></p> <p><b>OK</b></p> <p>If any of the optional parameters is specified, set the Will information: <b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command Query/set the MQTT message receiving mode when the data is received from server <b>AT+QMTCFG="recv/mode",&lt;client_idx&gt;[,&lt;msg_rcv_mode&gt;[,&lt;msg_len_enable&gt;]]</b></p>	<p>Response If the optional parameters are omitted, query the MQTT message receiving mode. <b>+QMTCFG: "recv/mode",&lt;msg_rcv_mode&gt;[,&lt;msg_len_enable&gt;]</b></p> <p><b>OK</b></p> <p>If any of the optional parameters is specified, set the MQTT message receiving mode: <b>OK</b></p>

	If there is any error: <b>ERROR</b>
Write Command Query/set Alibaba device information for AliCloud <b>AT+QMTCFG="aliauth",&lt;client_idx&gt;[,&lt;product_key&gt;,&lt;device_name&gt;,&lt;device_secret&gt;]</b>	Response If the optional parameters are omitted, query the device information: <b>[+QMTCFG: "aliauth",&lt;product_key&gt;,&lt;device_name&gt;,&lt;device_secret&gt;]</b>  <b>OK</b>  If the optional parameters are specified, set the device information: <b>OK</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations will not be saved.

## Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. Range: 0–5.
<b>&lt;vsn&gt;</b>	Integer type. MQTT protocol version. 3 MQTT v3.1 4 MQTT v3.1.1
<b>&lt;cid&gt;</b>	Integer type. The PDP to be used by the MQTT client. Range: 1–16. Default value : 1.
<b>&lt;will_fg&gt;</b>	Integer type. Configure the Will flag. <u>0</u> Ignore the Will flag configuration 1 Require the Will flag configuration
<b>&lt;will_qos&gt;</b>	Integer type. Quality of service for message delivery. <u>0</u> At most once 1 At least once 2 Exactly once
<b>&lt;will_retain&gt;</b>	Integer type. The Will retain flag is only used on PUBLISH messages. <u>0</u> When a client sends a PUBLISH message to a server, the server will not hold on to the message after it has been delivered to the current subscribers 1 When a client sends a PUBLISH message to a server, the server should hold on to the message after it has been delivered to the current subscribers
<b>&lt;will_topic&gt;</b>	Will topic string. The maximum length is 255 bytes.
<b>&lt;will_message&gt;</b>	String type. The Will message defines the content of the message that is published to the Will topic if the client is unexpectedly disconnected. It can be a

	zero-length message. The maximum length is 255 bytes.
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.
<timeout_notice>	Integer type. 0 Not report timeout message when transmitting packet 1 Report timeout message when transmitting packet
<clean_session>	Integer type. Configure the session type. 0 The server must store the subscriptions of the client after it is disconnected. 1 The server must discard any previously maintained information about the client and treat the connection as “clean”.
<keep_alive_time>	Integer type. Keep-alive time. Range: 0–3600. Default value: 120. Unit: second. It defines the maximum time interval between messages received from a client. If the server does not receive a message from the client within 1.5 times of the keep-alive time period, it disconnects the client as if the client has sent a DISCONNECT message. If the keep-alive value is 0, this means the server is not required to disconnect the client on the grounds of inactivity.
<SSL_enable>	Integer type. MQTT SSL mode. 0 Use normal TCP connection for MQTT 1 Use SSL TCP secure connection for MQTT
<ctx_index>	Integer type. SSL context index. Range: 0–5.
<msg_rcv_mode>	Integer type. The MQTT message receiving mode. 0 MQTT message received from server will be contained in URC 1 MQTT message received from server will not be contained in URC
<msg_len_enable>	Integer type. 0 Length of MQTT message received from server will not be contained in URC 1 Length of MQTT message received from server will be contained in URC
<product_key>	Product key issued by AliCloud.
<device_name>	Device name issued by AliCloud.
<device_secret>	Device secret key issued by AliCloud.

## NOTES

1. If <will\_fg>=1, then <will\_qos>, <will\_retain>, <will\_topic> and <will\_message> must be specified. Otherwise they will be omitted.
2. <clean\_session>=0 is only effective when the server supports the operation.
3. If MQTT connection is configured to SSL mode, <ctx\_index> must be specified, and **AT+QSSLCFG** must be used to configure the SSL version, cipher suite, secure level, CA certificate, client certificate, client key and ignorance of RTC time, which will be used in MQTT SSL handshake procedure. For more details of **AT+QSSLCFG**, please refer to *Quectel\_BG95&BG77\_SSL\_AT\_Commands\_Manual*.
4. Care must be taken to ensure message delivery does not time out while it is still being sent.
5. **AT+QMTCFG="aliauth"** is only used for AliCloud. If it is configured, <username> and <password> in **AT+QMTCONN** can be omitted.

### 3.2.2. AT+QMTOPEN Open a Network for MQTT Client

This command opens a network for MQTT client.

<b>AT+QMTOPEN Open a Network for MQTT Client</b>	
Test Command <b>AT+QMTOPEN=?</b>	Response <b>+QMTOPEN:</b> (range of supported <client_idx>s),<host_name>,(range of supported <port>s)  <b>OK</b>
Read Command <b>AT+QMTOPEN?</b>	Response <b>[+QMTOPEN:</b> <client_idx>,<host_name>,<port>]  <b>OK</b>
Write Command <b>AT+QMTOPEN=&lt;client_idx&gt;,&lt;host_name&gt;,&lt;port&gt;</b>	Response <b>OK</b>  <b>+QMTOPEN:</b> <client_idx>,<result>  If there is any error: <b>ERROR</b>
Maximum Response Time	Determined by network
Characteristics	/

#### Parameter

<client_idx>	Integer type. MQTT client identifier. Range: 0–5.
<host_name>	String type. The address of the server. It could be an IP address or a domain name. The maximum size is 100 bytes.
<port>	Integer type. The port number of the server. Range: 0–65535.
<result>	Integer type. Result of the command execution. -1 Failed to open network 0 Network opened successfully 1 Wrong parameter 2 MQTT client identifier is occupied 3 Failed to activate PDP 4 Failed to parse domain name 5 Network connection error

### 3.2.3. AT+QMTCLOSE Close a Network for MQTT Client

This command closes a network for MQTT client.

<b>AT+QMTCLOSE Close a Network for MQTT Client</b>	
Test Command <b>AT+QMTCLOSE=?</b>	Response <b>+QMTCLOSE:</b> (range of supported <client_idx>s)  <b>OK</b>
Write Command <b>AT+QMTCLOSE=&lt;client_idx&gt;</b>	Response <b>OK</b>  <b>+QMTCLOSE:</b> <client_idx>,<result>  If there is any error: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	/

### Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. Range: 0–5.
<b>&lt;result&gt;</b>	Integer type. Result of the command execution.
-1	Failed to close network
0	Network closed successfully

### 3.2.4. AT+QMTCONN Connect a Client to MQTT Server

This command is used when a client requests a connection to MQTT server. When a TCP/IP socket connection is established from a client to a server, a protocol level session must be created using a CONNECT flow.

<b>AT+QMTCONN Connect a Client to MQTT Server</b>	
Test Command <b>AT+QMTCONN=?</b>	Response <b>+QMTCONN:</b> (range of supported <client_idx>s),<clientID>,<username>,<password>  <b>OK</b>
Read Command <b>AT+QMTCONN?</b>	Response <b>[+QMTCONN:</b> <client_idx>,<state>]  <b>OK</b>
Write Command <b>AT+QMTCONN=&lt;client_idx&gt;,&lt;clientID&gt;[,&lt;username&gt;[,&lt;password&gt;]]</b>	Response <b>OK</b>

	+QMTCONN: <client_idx>,<result>[,<ret_code>]  If there is any error: <b>ERROR</b>
Maximum Response Time	<pkt_timeout> (default 5 s), determined by network
Characteristics	/

## Parameter

<client_idx>	Integer type. MQTT client identifier. Range: 0–5.
<clientID>	The client identifier string.
<username>	String type. User name of the client. It can be used for authentication.
<password>	String type. Password corresponding to the user name of the client. It can be used for authentication.
<result>	Integer type. Result of the command execution. 0 Packet sent successfully and ACK received from server 1 Packet retransmission 2 Failed to send packet
<state>	Integer type. MQTT connection state. 1 MQTT is initializing 2 MQTT is connecting 3 MQTT is connected 4 MQTT is disconnecting
<ret_code>	Integer type. Connection status return code. 0 Connection Accepted 1 Connection Refused: Unacceptable Protocol Version 2 Connection Refused: Identifier Rejected 3 Connection Refused: Server Unavailable 4 Connection Refused: Bad User Name or Password 5 Connection Refused: Not Authorized
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.

### NOTES

1. If a client with the same Client ID is already connected to the server, the "older" client must be disconnected by the server before completing the CONNECT flow of the new client.
2. **AT+QMTCFG="aliauth"** is only used for AliCloud. If it is configured, then **<username>** and **<password>** in **AT+QMTCONN** can be omitted.



### 3.2.5. AT+QMTDISC Disconnect a Client from MQTT Server

This command is used when a client requests a disconnection from MQTT server. A DISCONNECT message is sent from the client to the server to indicate that it is about to close its TCP/IP connection.

<b>AT+QMTDISC Disconnect a Client from MQTT Server</b>	
Test Command <b>AT+QMTDISC=?</b>	Response <b>+QMTDISC:</b> (range of supported <client_idx>s)  <b>OK</b>
Write Command <b>AT+QMTDISC=&lt;client_idx&gt;</b>	Response <b>OK</b>  <b>+QMTDISC:</b> <client_idx>,<result>  If there is any error: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	/

#### Parameter

<client_idx>	Integer type. MQTT client identifier. Range: 0–5.
<result>	Integer type. Result of the command execution
-1	Failed to close connection
0	Connection closed successfully

### 3.2.6. AT+QMTSUB Subscribe to Topics

This command is used to subscribe to one or more topics. A SUBSCRIBE message is sent by a client to register an interest in one or more topic names with the server. Messages published to these topics are delivered from the server to the client as PUBLISH messages.

<b>AT+QMTSUB Subscribe to Topics</b>	
Test Command <b>AT+QMTSUB=?</b>	Response <b>+QMTSUB:</b> (range of supported <client_idx>s),(range of supported <msgID>s),<topic>,(range of supported <qos>s)  <b>OK</b>
Write Command <b>AT+QMTSUB=&lt;client_idx&gt;,&lt;msgID&gt;,&lt;topic1&gt;,&lt;qos1&gt;[,&lt;topic2&gt;,&lt;qos2&gt;</b>	Response <b>OK</b>

>...]	+QMTSUB: <client_idx>,<msgID>,<result>[,<value>]  If there is any error: <b>ERROR</b>
Maximum Response Time	<pkt_timeout> x <retry_times> (default 15 s), determined by network
Characteristics	The command takes effect immediately. The configurations will not be saved.

## Parameter

<client_idx>	Integer type. MQTT client identifier. Range: 0–5.
<msgID>	Integer type. Message identifier of packet. Range: 1–65535.
<topic>	String type. The topic that the client wants to subscribe to or unsubscribe from.
<qos>	Integer type. The QoS level at which the client wants to publish the messages. 0 At most once 1 At least once 2 Exactly once
<result>	Integer type. Result of the command execution. 0 Sent packet successfully and received ACK from server 1 Packet retransmission 2 Failed to send packet
<value>	If <result> is 0, it is a vector of granted QoS levels. If <result> is 1, it means the times of packet retransmission. If <result> is 2, it will not be presented.
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.

### NOTE

The <msgID> is only present in messages where the QoS bits in the fixed header indicate QoS levels 1 or 2. It must be unique amongst the set of "inflight" messages in a particular direction of communication. It typically increases by exactly one from one message to the next, but is not required to do so.

### 3.2.7. AT+QMTUNS Unsubscribe from Topics

This command is used to unsubscribe from one or more topics. An UNSUBSCRIBE message is sent by the client to the server to unsubscribe from named topics.

<b>AT+QMTUNS Unsubscribe from Topics</b>	
Test Command <b>AT+QMTUNS=?</b>	Response <b>+QMTUNS:</b> (range of supported <b>&lt;client_idx&gt;s</b> ),(range of supported <b>&lt;msgID&gt;s</b> ), <b>&lt;topic&gt;</b>  <b>OK</b>
Write Command <b>AT+QMTUNS=&lt;client_idx&gt;,&lt;msgID&gt;,&lt;topic1&gt;[,&lt;topic2&gt;...]</b>	Response <b>OK</b>  <b>+QMTUNS:</b> <b>&lt;client_idx&gt;,&lt;msgID&gt;,&lt;result&gt;</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	<b>&lt;pkt_timeout&gt;</b> × <b>&lt;retry_times&gt;</b> (default 15 s), determined by network
Characteristics	The command takes effect immediately. The configurations will not be saved.

### Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. Range: 0–5.
<b>&lt;msgID&gt;</b>	Integer type. Message identifier of packet. Range: 1–65535.
<b>&lt;topic&gt;</b>	String type. The topic that the client wants to subscribe to or unsubscribe from.
<b>&lt;result&gt;</b>	Integer type. Result of the command execution. 0 Sent packet successfully and received ACK from server 1 Packet retransmission 2 Failed to send packet
<b>&lt;pkt_timeout&gt;</b>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.
<b>&lt;retry_times&gt;</b>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.

### 3.2.8. AT+QMTPUB Publish Messages

This command publishes messages by a client to a server for distribution to interested subscribers. Each PUBLISH message is associated with a topic name. If a client subscribes to one or more topics, any message published to those topics are sent by the server to the client as a PUBLISH message.

<b>AT+QMTPUB Publish Messages</b>	
Test Command <b>AT+QMTPUB=?</b>	Response <b>+QMTPUB:</b> (range of supported <b>&lt;client_idx&gt;s</b> ),(range of supported <b>&lt;msgID&gt;s</b> ),(range of supported <b>&lt;qos&gt;s</b> ),(list of

	supported <b>&lt;retain&gt;s),&lt;topic&gt;,(range of supported &lt;msglen&gt;s)</b>
	<b>OK</b>
Write Command Publish variable-length messages <b>AT+QMTPUB=&lt;client_idx&gt;,&lt;msgID&gt;,&lt;qos&gt;,&lt;retain&gt;,&lt;topic&gt;</b>	Response > After > is responded, input the data to be sent. Tap <b>Ctrl+Z</b> to send the data, and tap <b>Esc</b> to cancel the operation. <b>OK</b> <b>+QMTPUB: &lt;client_idx&gt;,&lt;msgID&gt;,&lt;result&gt;[,&lt;value&gt;]</b>
	If there is any error: <b>ERROR</b>
Write Command Publish fixed-length messages <b>AT+QMTPUB=&lt;client_idx&gt;,&lt;msgID&gt;,&lt;qos&gt;,&lt;retain&gt;,&lt;topic&gt;,&lt;msglen&gt;</b>	Response > After > is responded, input the data to be sent. The number of bytes of input data must equal <b>&lt;msglen&gt;</b> . <b>OK</b> <b>+QMTPUB: &lt;client_idx&gt;,&lt;msgID&gt;,&lt;result&gt;[,&lt;value&gt;]</b>
	If there is any error: <b>ERROR</b>
Maximum Response Time	<b>&lt;pkt_timeout&gt; x &lt;retry_times&gt;</b> (default 15 s), determined by network
Characteristics	/

## Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. Range: 0–5.
<b>&lt;msgID&gt;</b>	Integer type. Message identifier of packet. Range: 0–65535. It will be 0 only when <b>&lt;qos&gt;=0</b> .
<b>&lt;qos&gt;</b>	Integer type. The QoS level at which the client wants to publish the messages. 0 At most once 1 At least once 2 Exactly once
<b>&lt;retain&gt;</b>	Integer type. Whether or not the server will retain the message after it has been delivered to the current subscribers. 0 The server will not retain the message after it has been delivered to the current subscribers 1 The server will retain the message after it has been delivered to the current subscribers

<b>&lt;topic&gt;</b>	String type. Topic that needs to be published.
<b>&lt;msglen&gt;</b>	Integer type. Length of the message to be published. Range: 1–4096. Unit: byte.
<b>&lt;result&gt;</b>	Integer type. Result of the command execution. 0 Packet sent successfully and ACK received from server (message that published when <b>&lt;qos&gt;</b> =0 does not require ACK) 1 Packet retransmission 2 Failed to send packet
<b>&lt;value&gt;</b>	If <b>&lt;result&gt;</b> is 1, it means the times of packet retransmission. If <b>&lt;result&gt;</b> is 0 or 2, it will not be presented.
<b>&lt;pkt_timeout&gt;</b>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.
<b>&lt;retry_times&gt;</b>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.

**NOTES**

1. If this command is executed successfully and gets **OK** back, the client can continue to publish new packets. The maximum quantity of packet to be transmitted should not be greater than that of inflight windows (5).
2. After executing this command, the client will be ready to send data, which will be sent as payload. The maximum length of the input data is 4096 bytes at a time and please tap **Ctrl+Z** to send the data.
3. PUBLISH messages can be sent either from a publisher to the server, or from the server to a subscriber. When a server publishes messages to a subscriber, the following URC will be returned to notify the host to read the received data that is reported by MQTT server:  
**+QMTRECV: <client\_idx>,<msgID>,<topic>,<payload>**  
 For more details about the URC description, please refer to **Chapter 4.2**.

**3.2.9. AT+QMTPUBEX Publish Messages**

This command publishes messages. It provides the same functions as **AT+QMTPUB**, except that the format is different.

<b>AT+QMTPUBEX Publish Messages</b>	
Test Command <b>AT+QMTPUBEX=?</b>	Response <b>+QMTPUBEX:</b> (range of supported <b>&lt;client_idx&gt;</b> s),( range of supported <b>&lt;msgID&gt;</b> s),(range of supported <b>&lt;qos&gt;</b> s),(list of supported <b>&lt;retain&gt;</b> s), <b>&lt;topic&gt;</b> , <b>&lt;msg&gt;</b>  <b>OK</b>
Write Command <b>AT+QMTPUBEX=&lt;client_idx&gt;,&lt;msgID&gt;,&lt;qos&gt;,&lt;retain&gt;,&lt;topic&gt;,&lt;msg&gt;</b>	Response <b>OK</b>  <b>+QMTPUB:</b> <b>&lt;client_idx&gt;,&lt;msgID&gt;,&lt;result&gt;</b> [ <b>&lt;value&gt;</b> ]

	If there is any error: <b>ERROR</b>
Maximum Response Time	<pkt_timeout> × <retry_times> (default 15 s), determined by network
Characteristics	/

## Parameter

<client_idx>	Integer type. MQTT client identifier. Range: 0–5.
<msgID>	Integer type. Message identifier of packet. Range: 0–65535. It will be 0 only when <qos>=0.
<qos>	Integer type. The QoS level at which the client wants to publish the messages. <ul style="list-style-type: none"> <li>0 At most once</li> <li>1 At least once</li> <li>2 Exactly once</li> </ul>
<retain>	Integer type. Whether or not the server will retain the message after it has been delivered to the current subscribers. <ul style="list-style-type: none"> <li>0 The server will not retain the message after it has been delivered to the current subscribers</li> <li>1 The server will retain the message after it has been delivered to the current subscribers</li> </ul>
<topic>	String type. Topic that needs to be published.
<msg>	String type. Message to be published. Maximum length is 560 bytes.
<result>	Integer type. Result of the command execution. <ul style="list-style-type: none"> <li>0 Packet sent successfully and ACK received from server (message that published when &lt;qos&gt;=0 does not require ACK)</li> <li>1 Packet retransmission</li> <li>2 Failed to send packet</li> </ul>
<value>	If <result> is 1, it means the times of packet retransmission. If <result> is 0 or 2, it will not be presented.
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.

### NOTE

**AT+QMTPUBEX** is an extension command of **AT+QMTPUB**, and its response follows **AT+QMTPUB**.

### 3.2.10. AT+QMTRECV Read Messages from Buffers

This command reads messages from storage buffer where the messages are stored when they are reported by the server.

<b>AT+QMTRECV Read Messages from Buffers</b>	
Test Command <b>AT+QMTRECV=?</b>	Response <b>OK</b>
Read Command <b>AT+QMTRECV?</b>	Response <b>+QMTRECV: &lt;client_idx&gt;,&lt;store_status_0&gt;,&lt;store_status_1&gt;,&lt;store_status_2&gt;,&lt;store_status_3&gt;,&lt;store_status_4&gt;</b>  <b>OK</b>  If there is no MQTT connection: <b>OK</b>
Write Command <b>AT+QMTRECV=&lt;client_idx&gt;[,&lt;recv_id&gt;]</b>	Response List of ( <b>+QMTRECV: &lt;client_idx&gt;,&lt;msgID&gt;,&lt;topic&gt;,[&lt;payload_len&gt;],[&lt;payload&gt;]</b> )s  <b>OK</b>  If there is no message received: <b>OK</b>  If there is no MQTT connection: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	/

#### Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. Range: 0–5.
<b>&lt;store_status&gt;</b>	Integer type. Indicate whether there is a message stored in the buffer. 0 means no, and 1 means yes. The maximum quantity of message that can be stored in the buffer is 5. Therefore, URC reports maximally 5 messages simultaneously.
<b>&lt;recv_id&gt;</b>	Integer type. Indicate the serial number of every single message received. Range: 0–4. All buffer data will be read if this parameter is not specified.
<b>&lt;msgID&gt;</b>	Integer type. Message identifier of packet. Range: 0–65535. It will be 0 only when <b>&lt;qos&gt;=0</b> .
<b>&lt;topic&gt;</b>	String type. Topic that needs to be published.

---

<b>&lt;payload_len&gt;</b>	Integer type. The length of payload.
<b>&lt;payload&gt;</b>	String type. The payload that relates to the topic name.

---



# 4 MQTT Related URCs

This chapter gives MQTT related URCs and their descriptions.

**Table 1: MQTT Related URCs**

SN	URC Format	Description
[1]	<b>+QMTSTAT: &lt;client_idx&gt;,&lt;err_code&gt;</b>	When the state of MQTT link layer is changed, the client will close the MQTT connection and report the URC.
[2]	<b>+QMTRECV: &lt;client_idx&gt;,&lt;msgid&gt;,&lt;topic&gt;,&lt;payload&gt;</b>	Reported when the client has received the packet data from MQTT server.
[3]	<b>+QMTRECV: &lt;client_idx&gt;,&lt;recv_id&gt;</b>	Reported when the message received from MQTT server has been stored in buffer.

## 4.1. +QMTSTAT: URC to Indicate State Change in MQTT Link Layer

The URC begins with **+QMTSTAT:**. It will be reported when there is a change in the state of MQTT link layer.

### +QMTSTAT: URC to Indicate State Change in MQTT Link Layer

<b>+QMTSTAT: &lt;client_idx&gt;,&lt;err_code&gt;</b>	When the state of MQTT link layer is changed, the client will close the MQTT connection and report the URC.
--	---

#### Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. Range: 0-5.
<b>&lt;err_code&gt;</b>	Integer type. An error code. Please refer to the table below for details.

**Table 2: Error Codes of +QMTSTAT: URC**

<err_code>	Description	How to do
1	The connection is closed or reset by a peer.	Execute <b>AT+QMTOPEN</b> command to reopen the MQTT connection.

2	Sending PINGREQ packet timed out or failed.	Deactivate PDP first, and then activate PDP and reopen MQTT connection.
3	Sending CONNECT packet timed out or failed.	<ol style="list-style-type: none"> <li>1. Check whether the inputted user name and password are correct.</li> <li>2. Make sure the client ID is not used.</li> <li>3. Reopen MQTT connection and try to send CONNECT packet to the server again.</li> </ol>
4	Receiving CONNACK packet timed out or failed.	<ol style="list-style-type: none"> <li>1. Check whether the inputted user name and password are correct.</li> <li>2. Make sure the client ID is not used.</li> <li>3. Reopen MQTT connection and try to send CONNECT packet to the server again.</li> </ol>
5	The client sends DISCONNECT packet to sever and the server is initiative to close MQTT connection.	This is a normal process.
6	The client takes the initiative to close the MQTT connection due to packet sending failure all the time.	<ol style="list-style-type: none"> <li>1. Make sure the data is correct.</li> <li>2. Try to reopen MQTT connection since there may be network congestion or an error.</li> </ol>
7	The link is not alive or the server is unavailable.	Make sure the link is alive or the server is available currently.
8–255	Reserved for future use.	

## 4.2. +QMTRECV: URC to Inform the Host to Read MQTT Packet Data

The URC begins with **+QMTRECV:**. It is mainly used to inform the host to read the received MQTT packet data that is reported from MQTT server.

### +QMTRECV: URC to Inform the Host to Read MQTT Packet Data

<b>+QMTRECV: &lt;client_idx&gt;,&lt;msgid&gt;,&lt;topic&gt;,&lt;payload&gt;</b>	Notify the host to read the received data that is reported from MQTT server.
<b>+QMTRECV: &lt;client_idx&gt;,&lt;recv_id&gt;</b>	Notify that the message received from MQTT server has been stored in buffer.

### Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. Range: 0–5.
<b>&lt;msgid&gt;</b>	Integer type. The message identifier of packet.
<b>&lt;topic&gt;</b>	String type. The topic that received from MQTT server.
<b>&lt;payload&gt;</b>	String type. The payload that relates to the topic name.
<b>&lt;recv_id&gt;</b>	Integer type. The serial number of every single message received. Range: 0–4.

# 5 Examples

This chapter gives some examples to explain how to use MQTT related AT commands.

## 5.1. Example of MQTT Operation without SSL

```
AT+QMTCFG="aliauth",0,"oyjtmPI5a5j","MQTT_TEST","wN9Y6pZSIly7Exa5qVzcmigEGO4kAazZ"
//Configure Alibaba device information for AliCloud.
OK
AT+QMTOPEN=?
+QMTOPEN: (0-5),<host_name>,(0-65535)
OK
AT+QMTOPEN=0,"iot-as-mqtt.cn-shanghai.aliyuncs.com",1883 //Open a network for MQTT client.
OK
+QMTOPEN: 0,0 //Opened the MQTT client network successfully.
AT+QMTOPEN?
+QMTOPEN: 0,"iot-as-mqtt.cn-shanghai.aliyuncs.com",1883
OK
AT+QMTCONN=?
+QMTCONN: (0-5),<clientID>,<username>,<password>
OK
//Connect a client to MQTT server.
//If AliCloud is connected, AT+QMTCFG="aliauth" can be used to configure the device information in
advance, and there is no need to provide username/password here anymore.
AT+QMTCONN=0,"clientExample"
OK
+QMTCONN: 0,0,0 //Connected the client to MQTT server successfully.
AT+QMTSUB=?
+QMTSUB: (0-5),(1-65535),<topic>,(0-2)
OK
```

//Subscribe to topics.

**AT+QMTSUB=0,1,"topic/example",2**

OK

+QMTSUB: 0,1,0,2

**AT+QMTSUB=0,1,"topic/pub",0**

OK

+QMTSUB: 0,1,0,0

//If a client subscribes to a topic and other devices publish the same topic to the server, the module will report the following information.

**+QMTRECV: 0,0,"topic/example","This is the payload related to topic"**

//Unsubscribe from topics.

**AT+QMTUNS=0,1,"topic/example"**

OK

+QMTUNS: 0,2,0

**AT+QMTPUB=?**

**+QMTPUB : (0-5),(0-65535),(0-2),(0,1),<topic>,(1-4096)**

OK

//Publish messages.

**AT+QMTPUB=0,0,0,0,"topic/pub"**

**>This is test data, hello MQTT.** //After receiving **>**, input data **This is test data, hello MQTT.** and then send it. The maximum length of the data is 4096 bytes and the data that beyond 4096 bytes will be omitted. After inputting data, tap **Ctrl+Z** to send.

OK

+QMTPUB: 0,0,0

//If a client subscribes to a topic named "topic/pub" and other devices publish the same topic to the server, the module will report the following information.

**+QMTRECV: 0,0,"topic/pub","This is test data, hello MQTT."**

**AT+QMTDISC=0** //Disconnect a client from MQTT server.

OK

**+QMTDISC: 0,0** //Connection closed successfully.

## 5.2. Example of MQTT Operation with SSL

```
//Configure MQTT session into SSL mode.
AT+QMTCFG="ssl",0,1,2
OK

//If SSL authentication mode is intended to be set as "manage server and client authentication if
requested by the remote server" (<seclvl>=2 in AT+QSSLCFG), upload server root CA certificate,
client certificate and client private key to UFS.
AT+QFUPL="cacert.pem",1758,100 //Upload CA certificate to UFS.
CONNECT
<Input the cacert.pem data, and the size is 1758 bytes>
+QFUPL: 1758,384a

OK

AT+QFUPL="client.pem",1220,100 //Upload client certificate to UFS.
CONNECT
<Input the client.pem data, and the size is 1220 bytes>
+QFUPL: 1220,2d53

OK

AT+QFUPL="user_key.pem",1679,100 //Upload client private key to UFS.
CONNECT
<Input the user_key.pem data, and the size is 1679 bytes>
+QFUPL: 1679,335f

OK

//Configure the path of CA certificate for SSL context 2.
AT+QSSLCFG="cacert",2,"cacert.pem"
OK

//Configure the path of client certificate for SSL context 2.
AT+QSSLCFG="clientcert",2,"client.pem"
OK

//Configure the path of client private key for SSL context 2.
AT+QSSLCFG="clientkey",2,"user_key.pem"
OK

//Configure the authentication mode for SSL context 2.
AT+QSSLCFG="seclvl",2,2 //SSL authentication mode: server and client authentication
```

```
if requested by the remote server
OK
AT+QSSLCFG="sslversion",2,4 //SSL authentication version
OK
AT+QSSLCFG="ciphersuite",2,0XFFFF //Cipher suite
OK
AT+QSSLCFG="ignorelocaltime",2,1 //Ignore the time of authentication.
OK

//Start MQTT SSL connection
AT+QMTOPEN=0,"a1zgnxur10j8ux.iot.us-east-1.amazonaws.com",8883
OK

+QMTOPEN: 0,0

//Connect to MQTT server
AT+QMTCONN=0,"MQTT-1"
OK

+QMTCONN: 0,0,0

//Subscribe to topics.
AT+QMTSUB=0,1,"$aws/things/ MQTT-1/shadow/update/accepted",1
OK

+QMTSUB: 0,1,0,1

//Publish messages.
AT+QMTPUB=0,1,1,0,"$aws/things/MQTT-1/shadow/update/accepted"
>This is publish data from client
OK

+QMTPUB: 0,1,0

//If a client subscribes to a topic named "$aws/things/MQTT-1/shadow/update/accepted" and other
devices publish the same topic to the server, the module will report the following information.
+QMTRECV: 0,1,"$aws/things/MQTT-1/shadow/update/accepted", "This is publish data from client"

//Disconnect a client from MQTT server.
AT+QMTDISC=0
OK

+QMTDISC: 0,0
```

# 6 Appendix A References

**Table 3: Related Documents**

SN	Document Name	Remarks
[1]	MQTT V3.1 Protocol Specification	MQTT protocol specification version 3.1
[2]	MQTT V3.1.1 Protocol Specification	MQTT protocol specification version 3.1.1

**Table 4: Terms and Abbreviations**

Abbreviation	Description
ACK	Acknowledgement
CA	Certificate Authority
LPWA	Low Power Wide Area
MQTT	Message Queuing Telemetry Transport
QoS	Quality of Service
RAM	Random Access Memory
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
URC	Unsolicited Result Code