

# BC660K-GL MQTT

# Application Note

**NB-IoT Module Series**

Version: 1.1

Date: 2021-12-20

Status: Released



At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our 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 offices. 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 us at: [support@quectel.com](mailto:support@quectel.com).

## Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

## Use and Disclosure Restrictions

### License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

### Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

## Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

## Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties ("third-party materials"). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

## Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel's or third-party's servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

## Disclaimer

- a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
- b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
- c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
- d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

**Copyright © Quectel Wireless Solutions Co., Ltd. 2021. All rights reserved.**

# About the Document

## Revision History

Version	Date	Author	Description
-	2021-03-16	Arno DONG	Creation of the document
1.0	2021-07-21	Arno DONG	First official release
1.1	2021-12-20	Arno DONG	<ol style="list-style-type: none"> <li>1. Added the SSL function of MQTT with corresponding example, updated the note about the SSL function (Chapter 3.3.1 and Chapter 6.2).</li> <li>2. Updated the length ranges of &lt;will_topic&gt; and &lt;will_msg&gt; of AT+QMTCFG (Chapter 3.3.1).</li> <li>3. Updated the description of &lt;msg_len&gt; of AT+QMTPUB (Chapter 3.3.8).</li> </ol>

## Contents

About the Document .....	3
Contents .....	4
Table Index .....	5
<b>1 Introduction .....</b>	<b>6</b>
<b>2 MQTT Data Interaction.....</b>	<b>7</b>
<b>3 MQTT AT Commands .....</b>	<b>8</b>
3.1. AT Command Introduction .....	8
3.1.1. Definitions.....	8
3.1.2. AT Command Syntax .....	8
3.2. Declaration of AT Command Examples .....	9
3.3. Description of MQTT Related AT Commands.....	9
3.3.1. AT+QMTCFG Configure Optional Parameters of MQTT .....	9
3.3.2. AT+QMTOPEN Open a Network Connection for MQTT Client .....	15
3.3.3. AT+QMTCLOSE Close a Network Connection for MQTT Client.....	16
3.3.4. AT+QMTCONN Connect a Client to MQTT Server.....	17
3.3.5. AT+QMTDISC Disconnect a Client from MQTT Server .....	18
3.3.6. AT+QMTSUB Subscribe to Topics .....	19
3.3.7. AT+QMTUNS Unsubscribe from Topics .....	21
3.3.8. AT+QMTPUB Publish Messages .....	22
<b>4 Summary of Error Codes .....</b>	<b>25</b>
<b>5 MQTT Related URCs .....</b>	<b>26</b>
5.1. +QMTSTAT URC to Indicate State Change in MQTT Link Layer .....	26
5.2. +QMTRECV URC to Notify the Host to Read MQTT Packet Data.....	27
<b>6 Examples .....</b>	<b>28</b>
6.1. Example of MQTT Operation Without SSL.....	28
6.2. Example of MQTT Operation with SSL.....	30
<b>7 Appendix References .....</b>	<b>32</b>

## Table Index

Table 1: Types of AT Commands .....	8
Table 2: Summary of Error Codes .....	25
Table 3: MQTT Related URCs .....	26
Table 4: Error Codes of +QMTSTAT URC .....	26
Table 5: Related Documents .....	32
Table 6: Terms and Abbreviations .....	32

# 1 Introduction

MQTT (Message Queuing Telemetry Transport) is an OASIS standard messaging protocol for the Internet of Things (IoT). It is designed as an extremely lightweight publish/subscribe messaging transport that is ideal for connecting remote devices with a small code footprint and minimal network bandwidth.

This document introduces how to implement MQTT on the Quectel BC660K-GL module through AT commands.

# 2 MQTT Data Interaction

This chapter illustrates the mechanism of data interaction based on MQTT.

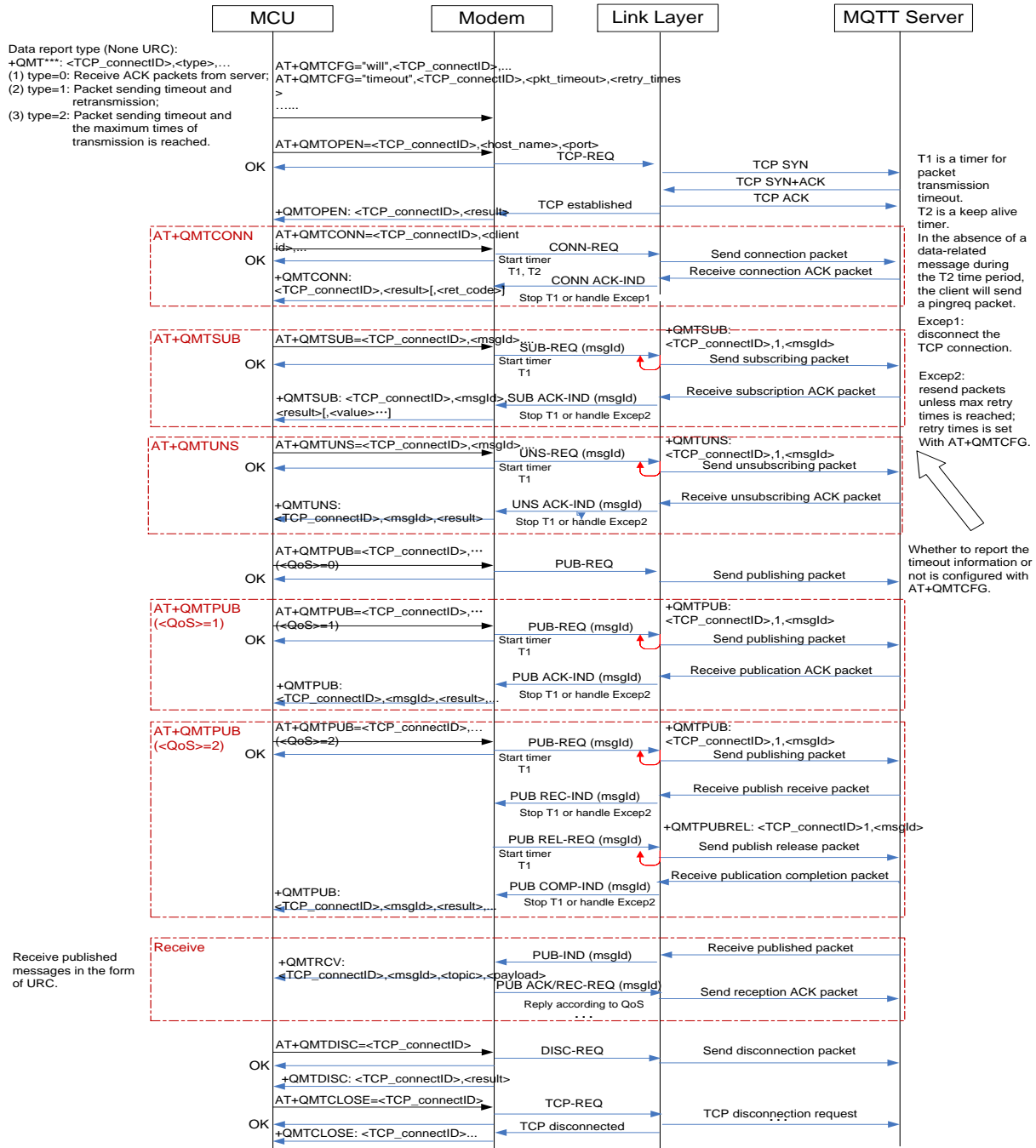


Figure 1: MQTT Data Interaction Diagram



# 3 MQTT AT Commands

This chapter presents the AT commands for operating MQTT function.

## 3.1. AT Command Introduction

### 3.1.1. Definitions

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

### 3.1.2. AT Command Syntax

All command lines must start with **AT** or **at** and end with **<CR>**. Information responses and result codes always start and end with a carriage return character and a line feed character: **<CR><LF><response><CR><LF>**. In tables presenting commands and responses throughout this document, only the commands and responses are presented, and **<CR>** and **<LF>** are deliberately omitted.

Table 1: Types of AT Commands

Command Type	Syntax	Description
Test Command	<b>AT+&lt;cmd&gt;=?</b>	Test the existence of corresponding Write Command and return information about the type, value, or range of its parameter.
Read Command	<b>AT+&lt;cmd&gt;?</b>	Check the current parameter value of a corresponding Write Command.
Write Command	<b>AT+&lt;cmd&gt;=&lt;p1&gt;[,&lt;p2&gt;[,&lt;p3&gt; [...]]]</b>	Set user-definable parameter value.

Execution Command	<b>AT+&lt;cmd&gt;</b>	Return a specific information parameter or perform a specific action.
-------------------	-----------------------	---

## 3.2. Declaration of AT Command Examples

The AT command examples in this document are provided to help you learn about the use of the AT commands introduced herein. The examples, however, should not be taken as Quectel's recommendation or suggestions about how you should design a program flow or what status you should set the module into. Sometimes multiple examples may be provided for one AT command. However, this does not mean that there exists a correlation among these examples and that they should be executed in a given sequence.

## 3.3. Description of MQTT Related AT Commands

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

This command configures optional parameters of MQTT.

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

Test Command <b>AT+QMTCFG=?</b>	Response <b>+QMTCFG: "will",</b> (range of supported <TCP_connectID>s),(list of supported <will_fg>s),(range of supported <will_QoS>s),(list of supported <will_retain>s),<will_topic>,<will_msg> <b>+QMTCFG: "timeout",</b> (range of supported <TCP_connectID>s),(range of supported <pkt_timeout>s),(range of supported <retry_times>s),(list of supported <timeout_notice>s) <b>+QMTCFG: "session",</b> (range of supported <TCP_connectID>s),(list of supported <clean_session>s) <b>+QMTCFG: "keepalive",</b> (range of supported <TCP_connectID>s),(range of supported <keep_alive_time>s) <b>+QMTCFG: "aliauth",</b> (range of supported <TCP_connectID>s),<product_key>,<device_name>,<device_secret> <b>+QMTCFG: "version",</b> (range of supported <TCP_connectID>s),(list of supported <version_num>s) <b>+QMTCFG: "showrecvlen",</b> (range of supported <TCP_connectID>s),(list of supported <show_flag>s) <b>+QMTCFG: "echo_mode",</b> (range of supported <TCP_connectID>s),(list of supported <echo_mode>s)
------------------------------------	--

	<p><b>+QMTCFG: "dataformat",</b>(range of supported &lt;TCP_connectID&gt;s),(list of supported &lt;send_format&gt;s),(list of supported &lt;recv_format&gt;s)</p> <p><b>+QMTCFG: "ssl",</b>(range of supported &lt;TCP_connectID&gt;s),(list of supported &lt;SSL_enable&gt;s),(range of supported &lt;SSL_contextID&gt;s),(range of supported &lt;SSL_connectID&gt;s)</p> <p><b>OK</b></p>
<p>Write Command</p> <p><b>AT+QMTCFG="will",&lt;TCP_connectID&gt;[,&lt;will_fg&gt;[,&lt;will_QoS&gt;,&lt;will_retain&gt;,&lt;will_topic&gt;,&lt;will_msg&gt;]]</b></p>	<p>Response</p> <p>If the optional parameters are omitted, query the current setting:</p> <p><b>+QMTCFG: &lt;will_fg&gt;[,&lt;will_QoS&gt;,&lt;will_retain&gt;,&lt;will_topic&gt;,&lt;will_msg&gt;]</b></p> <p><b>OK</b></p> <p>If any of the optional parameters are specified, configure Will information:</p> <p><b>OK</b></p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>Write Command</p> <p><b>AT+QMTCFG="timeout",&lt;TCP_connectID&gt;[,&lt;pkt_timeout&gt;[,&lt;retry_times&gt;][,&lt;timeout_notice&gt;]]</b></p>	<p>Response</p> <p>If the optional parameters are omitted, query the current setting:</p> <p><b>+QMTCFG: &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 are specified, configure Timeout information:</p> <p><b>OK</b></p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>Write Command</p> <p><b>AT+QMTCFG="session",&lt;TCP_connectID&gt;[,&lt;clean_session&gt;]</b></p>	<p>Response</p> <p>If the optional parameter is omitted, query the current setting:</p>

	<p><b>+QMTCFG: &lt;clean_session&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameter is specified, configure the session type:</p> <p><b>OK</b></p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>Write Command</p> <p><b>AT+QMTCFG="keepalive",&lt;TCP_connectID&gt;[,&lt;keep_alive_time&gt;]</b></p>	<p>Response</p> <p>If the optional parameter is omitted, query the current setting:</p> <p><b>+QMTCFG: &lt;keep_alive_time&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameter is specified, configure the keep-alive time:</p> <p><b>OK</b></p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>Write Command</p> <p><b>AT+QMTCFG="aliauth",&lt;TCP_connectID&gt;[,&lt;product_key&gt;,&lt;device_name&gt;,&lt;device_secret&gt;]</b></p>	<p>Response</p> <p>If the optional parameters are omitted, query the current setting:</p> <p><b>[+QMTCFG: &lt;product_key&gt;,&lt;device_name&gt;,&lt;device_secret&gt;]</b></p> <p><b>OK</b></p> <p>If the optional parameters are specified, configure the device information for Alibaba Cloud:</p> <p><b>OK</b></p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>

<p>Write Command</p> <p><b>AT+QMTCFG="version",&lt;TCP_connectID&gt;[,&lt;version_num&gt;]</b></p>	<p>Response</p> <p>If the optional parameter is omitted, query the current setting:</p> <p><b>+QMTCFG: &lt;version_num&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameter is specified, configure the version of MQTT:</p> <p><b>OK</b></p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>Write Command</p> <p><b>AT+QMTCFG="showrecvlen",&lt;TCP_connectID&gt;[,&lt;show_flag&gt;]</b></p>	<p>Response</p> <p>If the optional parameter is omitted, query the current setting:</p> <p><b>+QMTCFG: &lt;show_flag&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameter is specified, configure whether to show the length of received data:</p> <p><b>OK</b></p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>Write Command</p> <p><b>AT+QMTCFG="echomode",&lt;TCP_connectID&gt;[,&lt;echo_mode&gt;]</b></p>	<p>Response</p> <p>If the optional parameter is omitted, query the current setting:</p> <p><b>+QMTCFG: &lt;echo_mode&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameter is specified, configure whether to echo the input data to the UART in data mode:</p> <p><b>OK</b></p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>Write Command</p> <p><b>AT+QMTCFG="dataformat",&lt;TCP_conn</b></p>	<p>Response</p> <p>If the optional parameters are omitted, query the current</p>

ectID>[,<send_format>,<recv_format>]	<p>setting:</p> <p><b>+QMTCFG: &lt;send_format&gt;,&lt;recv_format&gt;</b></p> <p><b>OK</b></p> <p>If the optional parameters are specified, set the format for sending and receiving data:</p> <p><b>OK</b></p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
<p>Write Command</p> <p><b>AT+QMTCFG="ssl",&lt;TCP_connectID&gt;[,&lt;SSL_enable&gt;[,&lt;SSL_contextID&gt;,&lt;SSL_connectID&gt;]]</b></p>	<p>Response</p> <p>If the optional parameters are omitted, query the current configuration:</p> <p><b>+QMTCFG: &lt;SSL_enable&gt;[,&lt;SSL_contextID&gt;,&lt;SSL_connectID&gt;]</b></p> <p><b>OK</b></p> <p>If any of the optional parameters is specified, configure whether to use SSL secure connection:</p> <p><b>OK</b></p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
Maximum Response Time	300 ms
Characteristics	<p>The command takes effect immediately.</p> <p>The configuration will not be saved.</p>

## Parameter

<b>&lt;TCP_connectID&gt;</b>	Integer type. MQTT socket identifier. Range: 0–4.
<b>&lt;will_fg&gt;</b>	Integer type. Configuration of Will flag.
	0 Ignore Will flag configuration
	1 Require Will flag configuration
<b>&lt;will_QoS&gt;</b>	Integer type. QoS for message delivery.
	0 At most once - the message is sent only once and the client and server take no additional steps to acknowledge delivery
	1 At least once - the message is re-tried by the sender multiple times until

	acknowledgement is received
	2 Exactly once - the sender and receiver engage in a two-level handshake to ensure only one copy of the message is received
<b>&lt;will_retain&gt;</b>	Integer type. Specifies if the Will message is to be retained when it is published. 0 When a client sends a Will message to a server, the server does not retain the message after delivering it to the current subscribers 1 When a client sends a Will message to a server, the server retains the message after delivering it to the current subscribers
<b>&lt;will_topic&gt;</b>	String type. The topic of the Will message. The length range is 1–256 bytes.
<b>&lt;will_msg&gt;</b>	String type. The payload of the Will message published to the Will topic if the client is unexpectedly disconnected. The length range is 1–256 bytes.
<b>&lt;pkt_timeout&gt;</b>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 30. Unit: second.
<b>&lt;retry_times&gt;</b>	Integer type. The number of retransmissions after the timeout of a packet delivery. Range: 0–10. Default value: 0.
<b>&lt;timeout_notice&gt;</b>	Integer type. Whether to report timeout message when a packet delivery times out. 0 Not report 1 Report
<b>&lt;clean_session&gt;</b>	Integer type. Configure the type of a session. 0 Non-clean session - the server saves client subscriptions after the client is disconnected. 1 Clean session - the server deletes all the previously retained information about the client when the client is disconnected.
<b>&lt;keep_alive_time&gt;</b>	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 any 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. 0 The client is not disconnected
<b>&lt;product_key&gt;</b>	String type. Product key obtained from Alibaba Cloud. Maximum length: 64 bytes.
<b>&lt;device_name&gt;</b>	String type. Device name obtained from Alibaba Cloud. Maximum length: 64 bytes.
<b>&lt;device_secret&gt;</b>	String type. Device secret obtained from Alibaba Cloud. Maximum length: 64 bytes.
<b>&lt;version_num&gt;</b>	Integer type. MQTT version number 0 MQTT version 3.1.0 1 MQTT version 3.1.1
<b>&lt;show_flag&gt;</b>	Integer type. Indicates whether to show the length of received data ( <b>&lt;payload_len&gt;</b> ). 0 Not show the length of received data 1 Show the length of received data
<b>&lt;echo_mode&gt;</b>	Integer type. Whether to echo input data to the UART in data mode. 0 Not echo input data to the UART 1 Echo input data to the UART
<b>&lt;send_format&gt;</b>	Integer type. The data format for sending. 0 Text mode

	1 Hex mode
<b>&lt;recv_format&gt;</b>	Integer type. The data format for receiving.
	0 Text mode
	1 Hex mode
<b>&lt;SSL_enable&gt;</b>	Integer type. Indicates whether to use SSL/TLS secure connection for MQTT.
	0 Do not use SSL/TLS TCP secure connection
	1 Use SSL/TLS TCP secure connection
<b>&lt;SSL_contextID&gt;</b>	Integer type. SSL context ID. Range: 0–10. Currently only 0 is supported. Default value: 0.
<b>&lt;SSL_connectID&gt;</b>	Integer type. SSL connect ID. Range: 0–4. Currently only 0 is supported. Default value: 0.
<b>&lt;err&gt;</b>	Integer type. Error codes. See <b>Chapter 4</b> for details.

#### NOTE

1. **<will\_QoS>**, **<will\_retain>**, **<will\_topic>** and **<will\_msg>** must be specified when **<will\_fg>=1** and should be omitted when **<will\_fg>=0**.
2. **<clean\_session>=1** is valid only when the server supports session cleaning.
3. Care must be taken to ensure message delivery does not time out while it is still being sent.
4. **AT+QMTCFG="aliauth"** is only applicable to Alibaba Cloud. If the relevant parameters have been configured by this command, **<user\_name>** and **<pass\_word>** in **AT+QMTCONN** can be omitted.
5. The configuration of **<echo\_mode>** is only valid in data transmission mode.
6. The settings of "will", "session", "keepalive", "aliauth", "version" and "ssl" have to be configured before executing **AT+QMTOPEN**.

### 3.3.2. AT+QMTOPEN Open a Network Connection for MQTT Client

This command opens a network connection for an MQTT client.

#### AT+QMTOPEN Open a Network Connection for MQTT Client

Test Command <b>AT+QMTOPEN=?</b>	Response <b>+QMTOPEN:</b> (range of supported <b>&lt;TCP_connectID&gt;s</b> ), <b>&lt;host_name&gt;</b> ,(range of supported <b>&lt;port&gt;s</b> )  <b>OK</b>
Read Command <b>AT+QMTOPEN?</b>	Response <b>[+QMTOPEN:</b> <b>&lt;TCP_connectID&gt;</b> , <b>&lt;host_name&gt;</b> , <b>&lt;port&gt;</b> ]  <b>OK</b>
Write Command <b>AT+QMTOPEN=&lt;TCP_connectID&gt;,&lt;host_name&gt;,&lt;port&gt;</b>	Response <b>OK</b>  <b>+QMTOPEN:</b> <b>&lt;TCP_connectID&gt;</b> , <b>&lt;result&gt;</b>



	If there is any error: <b>ERROR</b> Or <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	176 s, determined by network
Characteristics	/

## Parameter

<b>&lt;TCP_connectID&gt;</b>	Integer type. MQTT socket identifier. Range: 0–4.
<b>&lt;host_name&gt;</b>	String type. The address of the server. It can be an IP address or a domain name. Maximum length: 100 bytes.
<b>&lt;port&gt;</b>	Integer type. The port of the server. Range: 1–65535.
<b>&lt;result&gt;</b>	Integer type. Result of the command execution. -1 Failed to open network connection 0 Network connection opened successfully 1 Invalid parameter 2 The MQTT socket is occupied 3 Failed to activate the PDP 4 Failed to parse the domain name
<b>&lt;err&gt;</b>	Integer type. Error codes. See <b>Chapter 4</b> for details.

### 3.3.3. AT+QMTCLOSE Close a Network Connection for MQTT Client

This command closes the network connection for an MQTT client.

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

Characteristics

/

## Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Range: 0–4.
<result>	Integer type. Result of the command execution. -1 Failed to close network connection 0 Network connection closed successfully
<err>	Integer type. Error codes. See <b>Chapter 4</b> for details.

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

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

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

Test Command <b>AT+QMTCONN=?</b>	Response <b>+QMTCONN:</b> (range of supported <TCP_connectID>s),<clientID>,<username>,<password>  <b>OK</b>
Read Command <b>AT+QMTCONN?</b>	Response <b>[+QMTCONN: &lt;TCP_connectID&gt;,&lt;state&gt;]</b>  <b>OK</b>
Write Command <b>AT+QMTCONN=&lt;TCP_connectID&gt;,&lt;clientID&gt;[,&lt;username&gt;[,&lt;password&gt;]]</b>	Response <b>OK</b>  <b>+QMTCONN: &lt;TCP_connectID&gt;,&lt;result&gt;[,&lt;ret_code&gt;]</b>  If there is any error: <b>ERROR</b> Or <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	<b>&lt;pkt_timeout&gt; × (&lt;retry_times&gt; + 1); 30 s by default; determined by network</b>
Characteristics	/

## Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Range: 0–4.
-----------------	---

<clientID>	String type. The client identifier. Maximum length: 128 bytes.
<user_name>	String type. User name of the client; used for authentication. Maximum length: 256 bytes.
<pass_word>	String type with double quotes. Password corresponding to the user name of the client; used for authentication. Maximum length: 256 bytes.
<result>	Integer type. Result of the command execution. 0 Sent packet successfully and received an ACK from the server 1 Retransmit packet 2 Failed to send packet
<state>	Integer type. MQTT connection state. 1 MQTT connection is initializing 2 connecting 3 connected 4 disconnecting
<ret_code>	Integer type. Connection status code. 0 Connection Accepted 1 Connection Refused: Unacceptable Protocol Version 2 Connection Refused: Identifier Rejected 3 Connection Refused: Server Unavailable 4 Connection Refused: Incorrect User Name or Password 5 Connection Refused: Unauthorized
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 30. Unit: second.
<retry_times>	Integer type. The number of retransmissions after the timeout of packet delivery. Range: 0–10. Default value: 0.
<err>	Integer type. Error codes. See <b>Chapter 4</b> for details.

#### NOTE

A client already connected to the server must be disconnected from the server before the CONNECT flow of a new client with the same client ID can be completed.

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

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

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

Test Command <b>AT+QMTDISC=?</b>	Response <b>+QMTDISC:</b> (range of supported <TCP_connectID>s)  <b>OK</b>
Write Command	Response

AT+QMTDISC=<TCP_connectID>	<p>OK</p> <p>+QMTDISC: &lt;TCP_connectID&gt;,&lt;result&gt;</p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>
Maximum Response Time	300 ms
Characteristics	/

## Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Range: 0–4.
<result>	Integer type. Result of the command execution.
	-1 Failed to disconnect the client
	0 Disconnected the client successfully
<err>	Integer type. Error codes. See <b>Chapter 4</b> for details.

### 3.3.6. AT+QMTSUB Subscribe to Topics

This command subscribes to one or more topics. A SUBSCRIBE message is sent by a client to register its interests in one or more topics with the server. Messages published to these topics are delivered from the server to the client as PUBLISH messages.

AT+QMTSUB Subscribe to Topics	
Test Command AT+QMTSUB=?	<p>Response</p> <p>+QMTSUB: (range of supported &lt;TCP_connectID&gt;s),(range of supported &lt;msgid&gt;s),&lt;topic1&gt;,(range of supported &lt;QoS1&gt;s),&lt;topic2&gt;,(range of supported &lt;QoS2&gt;s)...</p> <p>OK</p>
Write Command AT+QMTSUB=<TCP_connectID>,<msgid>,<topic1>,<QoS1>[,<topic2>,<QoS2>...]	<p>Response</p> <p>OK</p> <p>+QMTSUB: &lt;TCP_connectID&gt;,&lt;msgid&gt;,&lt;result&gt;[,&lt;value&gt;...]</p> <p>If there is any error:</p> <p><b>ERROR</b></p> <p>Or</p> <p><b>+CME ERROR: &lt;err&gt;</b></p>

Maximum Response Time	$\langle \text{pkt\_timeout} \rangle \times (\langle \text{retry\_times} \rangle + 1)$ ; 30 s by default; determined by network
Characteristics	/

## Parameter

<b>&lt;TCP_connectID&gt;</b>	Integer type. MQTT socket identifier. Range: 0–4.
<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. Maximum length: 256 bytes.
<b>&lt;QoS&gt;</b>	Integer type. The QoS level at which the client wants to publish messages. 0 At most once - the message is sent only once and the client and the server take additional steps to acknowledge delivery 1 At least once - the message is re-tried by the sender multiple times until acknowledgement is received 2 Exactly once - the sender and receiver engage in a two-level handshake to ensure only one copy of the message is received
<b>&lt;result&gt;</b>	Integer type. Result of the command execution 0 Sent packet successfully and received an ACK from server 1 Retransmit packet 2 Failed to send packet
<b>&lt;value&gt;</b>	Integer type. When <b>&lt;result&gt;</b> =0, it is a vector of granted QoS levels. <b>&lt;value&gt;</b> =128 indicate that the subscription is rejected by the server. When <b>&lt;result&gt;</b> =1, it is the number of times of packet retransmission. When <b>&lt;result&gt;</b> =2, it will not be presented.
<b>&lt;pkt_timeout&gt;</b>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 30. Unit: second.
<b>&lt;retry_times&gt;</b>	Integer type. The number of retransmissions after the timeout of a packet delivery. Range: 0–10. Default value: 0.
<b>&lt;err&gt;</b>	Integer type. Error codes. See <b>Chapter 4</b> for details.

### NOTE

The **<msgid>** is only present in messages where the QoS bits in the fixed header indicate that the QoS level is 1 or 2. **<msgid>** of each message in the inflight window of a particular communication must be unique. Ideally, it increases by exactly one from a message to the next.

### 3.3.7. AT+QMTUNS Unsubscribe from Topics

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

AT+QMTUNS Unsubscribe from Topics	
Test Command <b>AT+QMTUNS=?</b>	Response <b>+QMTUNS:</b> (range of supported <TCP_connectID>s),(range of supported <msgID>s),<topic1>,<topic2>...  <b>OK</b>
Write Command <b>AT+QMTUNS=&lt;TCP_connectID&gt;,&lt;msgID&gt;,&lt;topic1&gt;[,&lt;topic2&gt;...]</b>	Response <b>OK</b>  <b>+QMTUNS:</b> <TCP_connectID>,<msgID>,<result>  If there is any error: <b>ERROR</b> Or <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	<b>&lt;pkt_timeout&gt;</b> × ( <b>&lt;retry_times&gt;</b> + 1); 30 s by default; determined by network
Characteristics	/

#### Parameter

<b>&lt;TCP_connectID&gt;</b>	Integer type. MQTT socket identifier. Range: 0–4.
<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 unsubscribe from. Maximum length: 256 bytes.
<b>&lt;result&gt;</b>	Integer type. Result of the command execution. 0 Packet sent successfully and ACK received from the server 1 Packet retransmission 2 Failed to send the packet
<b>&lt;pkt_timeout&gt;</b>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 30. Unit: second.
<b>&lt;retry_times&gt;</b>	Integer type. The number of retransmissions after the timeout of a packet delivery. Range: 0–10. Default value: 0.
<b>&lt;err&gt;</b>	Integer type. Error codes. See <b>Chapter 4</b> for details.

### 3.3.8. 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 specified topic. If a client subscribes to one or more topics, any message published to those topics will be sent by the server to the client as a PUBLISH message.

#### AT+QMTPUB Publish Messages

Test Command  
**AT+QMTPUB=?**

Response

**+QMTPUB:** (range of supported <TCP\_connectID>s), (range of supported <msgid>s), (list of supported <QoS>s), (list of supported <retain>s), <topic>, (range of supported <msg\_len>s), <msg>

**OK**

Write Command  
Publish variable-length messages in data mode  
**AT+QMTPUB=<TCP\_connectID>,<msgid>,<QoS>,<retain>,<topic>**

Response

**>**

After **>** is responded, input the data to be sent. Tap “CTRL” + “Z” to send the data, and tap **Esc** to cancel the operation.

**OK**

**+QMTPUB:** <TCP\_connectID>,<msgid>,<result>[,<value>]

If there is any error:

**ERROR**

Or

**+CME ERROR: <err>**

Write Command  
Publish fixed-length messages in data mode  
**AT+QMTPUB=<TCP\_connectID>,<msgid>,<QoS>,<retain>,<topic>,<msg\_len>**

Response

**>**

After **>** is responded, input the data to be sent. Length of the data must be equal to <msg\_len>.

**OK**

**+QMTPUB:** <TCP\_connectID>,<msgid>,<result>[,<value>]

If there is any error:

**ERROR**

Or

**+CME ERROR: <err>**

Write Command  
Publish fixed-length messages in non-data mode

Response

**OK**

AT+QMTPUB=<TCP_connectID>,<msgID>,<QoS>,<retain>,<topic>,<msg_len>,<msg>	<b>+QMTPUB: &lt;TCP_connectID&gt;,&lt;msgID&gt;,&lt;result&gt;[,&lt;value&gt;]</b>  If there is any error: <b>ERROR</b> Or <b>+CME ERROR: &lt;err&gt;</b>
Maximum Response Time	<b>&lt;pkt_timeout&gt;</b> × ( <b>&lt;retry_times&gt;</b> + 1); 30 s by default; determined by network
Characteristics	/

## Parameter

<b>&lt;TCP_connectID&gt;</b>	Integer type. MQTT socket identifier. Range: 0–4.
<b>&lt;msgID&gt;</b>	Integer type. Message identifier of packet. Range: 0–65535. It should be set to 0 when <b>&lt;QoS&gt;</b> =0.
<b>&lt;QoS&gt;</b>	Integer type. The QoS level at which the client wants to publish the messages. <u>0</u> At most once - the message is sent only once and the client and the server take no additional steps to acknowledge delivery 1 At least once - the message is re-tried by the sender multiple times until acknowledgement is received 2 Exactly once - the sender and receiver engage in a two-level handshake to ensure only one copy of the message is received
<b>&lt;retain&gt;</b>	Integer type. Whether or not the message will be retained by the server after it has been delivered to the current subscribers. <u>0</u> The message will not be retained by the server after it has been delivered to current subscribers 1 The message will be retained by the server after it has been delivered to current subscribers
<b>&lt;topic&gt;</b>	String type. Topic to which the message is to be published. Maximum length: 256 bytes.
<b>&lt;msg&gt;</b>	String type. Message to be published.
<b>&lt;msg_len&gt;</b>	Integer type. Data length to be specified. The range of the data length in text mode is 0–1460. The range of the data length in hex mode is 0–730. Unit: byte.
<b>&lt;result&gt;</b>	Integer type. Result of the command execution. 0 Sent packet successfully and received an ACK from the server (messages published when <b>&lt;QoS&gt;</b> =0 do not require any ACK) 1 Retransmit packet 2 Failed to send the packet
<b>&lt;value&gt;</b>	Integer type. When <b>&lt;result&gt;</b> =1, it is the number of times of packet retransmission. When <b>&lt;result&gt;</b> =0 or <b>&lt;result&gt;</b> =2, it is not presented.
<b>&lt;pkt_timeout&gt;</b>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 30. Unit:



	second.
<retry_times>	Integer type. The number of retransmissions after the timeout of a packet delivery. Range: 0–10. Default value: 0.
<err>	Integer type. Error codes. See <b>Chapter 4</b> for details.

#### NOTE

1. If this command is executed successfully and responded with **OK**, the client is able to continue to publish new packets after the final result of the command is returned.
2. After this command is executed, the client is ready to send the data as payload. The maximum length of data that can be input at a time is 1460 bytes and the data is sent by tapping “**CTRL**” + “**Z**”. The part of data that exceeds 1460 bytes is omitted.
3. PUBLISH messages can be sent either from a publisher to a server, or from a server to a subscriber. When a server publishes messages to a subscriber, the following URC is returned to notify the client host to read the data received from the server:  
**+QMTRECV: <TCP\_connectID>,<msgID>,<topic>[,<payload\_len>],<payload>**  
For more details about this URC, see **Chapter 5.2**.
4. **<result>=2** indicates a packet transmission timeout or a case of stack block.
5. After entering data mode, if no data is sent in 60 s, the module will exit data mode.

# 4 Summary of Error Codes

Final result code **+CME ERROR: <err>** indicates an error related to mobile equipment or network. The following table lists some of the common error codes.

**Table 2: Summary of Error Codes**

<err>	Meaning
8503	MQTT link layer error
8504	MQTT illegal packet
8505	MQTT illegal character
8506	MQTT illegal UTF8
8507	MQTT invalid parameter
8508	The length of transmitted data exceeds the size of the remaining transmission buffer
8509	MQTT buffer overflow
8510	MQTT out of memory
8511	MQTT memory error
8600	MQTT unknown error

# 5 MQTT Related URCs

This chapter presents MQTT related URCs and their descriptions.

Table 3: MQTT Related URCs

SN	Notification Display	Description
[1]	<b>+QMTSTAT: &lt;TCP_connectID&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;TCP_connectID&gt;,&lt;msgID&gt;,&lt;topic&gt;[,&lt;payload_len&gt;],&lt;payload&gt;</b>	The URC is reported when the client receives a packet from the MQTT server.

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

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

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

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

#### Parameter

<b>&lt;TCP_connectID&gt;</b>	Integer type. MQTT socket identifier. Range: 0–4.
<b>&lt;err_code&gt;</b>	Error codes. See <b>Table 4</b> for details.

Table 4: Error Codes of +QMTSTAT URC

<err_code>	Description	How to do
1	Connection is closed or reset by its remote peer.	Execute <b>AT+QMTOPEN</b> to reopen the MQTT connection.
2	Sending PINGREQ packet timed out or failed.	Deactivate and activate the PDP, and then reopen the 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 occupied.</li> <li>3. Reopen the MQTT connection and try to send the 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 occupied.</li> <li>3. Reopen the MQTT connection and try to send the CONNECT packet to the server again.</li> </ol>
5	The client sent a DISCONNECT packet to the sever but the server took the initiative in closing the MQTT connection.	This is a normal process.
6	The client took the initiative in closing the MQTT connection due to constant packet sending failures.	<ol style="list-style-type: none"> <li>1. Make sure the data is correct.</li> <li>2. Given that there may be network congestion or error, try to reopen the MQTT connection.</li> </ol>
7	The link is not alive or the server is inaccessible.	Make sure the link is alive and the server is accessible.
8-255	Reserved for future use.	/

## 5.2. +QMTRECV URC to Notify the Host to Read MQTT Packet Data

The URC begins with **+QMTRECV:**. It notifies the client host to read the packet data it has received from the MQTT server. You can configure with **AT+QMTCFG="SHOWRECVLEN"** whether to display **<payload\_len>** or not.

### +QMTRECV URC to Notify the Host to Read MQTT Packet Data

**+QMTRECV: <TCP\_connectID>,<msgid>** Notify the client host to read the packet data it has received  
**D>,<topic>[,<payload\_len>],<payload>** from the MQTT server.

#### Parameter

<b>&lt;TCP_connectID&gt;</b>	Integer type. MQTT socket identifier. Range: 0–4.
<b>&lt;msgid&gt;</b>	Integer type. Message identifier of packet.
<b>&lt;topic&gt;</b>	String type. The topic received from the MQTT server.
<b>&lt;payload_len&gt;</b>	Integer type. Length of the payload.
<b>&lt;payload&gt;</b>	String type. The payload relevant to the topic.

# 6 Examples

This chapter provides the examples illustrating the use of MQTT commands.

## 6.1. Example of MQTT Operation Without SSL

```

AT+QSCCLK=0                                //Disable sleep modes.

OK
AT+QMTCFG="aliauth",0,"oyjtmPI5a5j","MQTT_TEST","wN9Y6pZSIly7Exa5qVzcmigEGO4kAaz
Z"                                           //Configure device information for Alibaba Cloud.
OK
AT+QMTOPEN=0,"oyjtmPI5a5j.iot-as-mqtt.cn-shanghai.aliyuncs.com",1883
                                           //Open a network connection for MQTT client.
OK

+QMTOPEN: 0,0                             //MQTT client connection is opened successfully.
AT+QMTOPEN?
+QMTOPEN: 0,"oyjtmPI5a5j.iot-as-mqtt.cn-shanghai.aliyuncs.com",1883

OK
AT+QMTCONN=?
+QMTCONN: (0-4),<clientID>,<username>,<password>

OK
//If you have configured the device information in advance with AT+QMTCFG="aliauth", you can omit to
configure <username> and <password> here.
AT+QMTCONN=0,"clientExample"              //Connect a client to the MQTT server.
OK

+QMTCONN: 0,0,0                            //The client has connected to the MQTT server successfully.
AT+QMTSUB=?
+QMTSUB: (0-4),(1-65535),<topic>,(0-2),<topic>,(0-2),...

OK
AT+QMTSUB=0,1," /oyjtmPI5a5j/S2fXbFBKh4NSwwyljSC4/user/get",1
                                           //Subscribe to a topic.

```

OK

+QMTSUB: 0,1,0,1

AT+QMTSUB=0,1," /oyjtmPI5a5j/S2fXbFBKh4NSwwyljSC4/user/update",1

OK

+QMTSUB: 0,1,0,1

//If the client subscribes to topics published to the server by other clients, the module will report the following information to the client host:

+QMTRECV: 0,1023," /oyjtmPI5a5j/S2fXbFBKh4NSwwyljSC4/user/get","This is the payload related to topic"

AT+QMTUNS=0,2," /oyjtmPI5a5j/S2fXbFBKh4NSwwyljSC4/user/get"

//Unsubscribe from a topic.

OK

+QMTUNS: 0,2,0

AT+QMTPUB=?

+QMTPUB : (0-4),(0-65535),(0-2),(0,1),<topic>,(0-1460),<msg>

OK

AT+QMTPUB=0,0,0,0," /oyjtmPI5a5j/S2fXbFBKh4NSwwyljSC4/user/update"

//Publish a variable-length message.

>

This is test data, hello MQTT.

//After receiving >, input data "This is test data, hello MQTT." and send it. The maximum length of the data is 1460 bytes and the data that exceeds 1460 bytes will be omitted. After inputting data, tap "CTRL" + "Z" to send it.

OK

+QMTPUB: 0,0,0

//If the client has subscribed to the topic "/oyjtmPI5a5j/S2fXbFBKh4NSwwyljSC4/user/update", when other clients publish messages to the same topic, the module will report the following information to the client host:

+QMTRECV: 0,0,"/oyjtmPI5a5j/S2fXbFBKh4NSwwyljSC4/user/update","This is test data, hello MQTT."

AT+QMTDISC=0

//Disconnect the client from the MQTT server.

OK

+QMTDISC: 0,0

//Connection closed successfully.

AT+QSCLK=1

//Enable sleep modes.

OK

## 6.2. Example of MQTT Operation with SSL

```

AT+QSCCLK=0                                //Disable sleep modes.
OK

//Configure certificates and keys
AT+QSSLCFG=0,0,"secllevel",2                //Manage server and client authentication.
OK
AT+QSSLCFG=0,0,"cacert"                    //Configure CA certificate.
>                                           //Input the content of trusted CA certificate in PEM format, and
                                           //the content of trusted CA certificate will be displayed.
                                           Tap "CTRL" + "Z" to send.
+QSSLCFG: 0,0,"cacert",1282

OK
AT+QSSLCFG=0,0,"clientcert"                //Configure client certificate.
>                                           //Input the content of the client certificate in PEM format, and the
                                           //content of the client certificate will be displayed.
                                           Tap "CTRL" + "Z" to send.
+QSSLCFG: 0,0,"clientcert",1216

OK
AT+QSSLCFG=0,0,"clientkey"                 //Configure client private key.
>                                           //Input the content of the client private key in PEM format, and the
                                           //content of the client private key will be displayed.
                                           Tap "CTRL" + "Z" to send.
+QSSLCFG: 0,0,"clientkey",1679

OK
AT+QSCCLK=1                                //Enable light sleep and deep sleep, and wakeup by PSM_EINT
                                           //(falling edge).

OK
AT+QMTCFG="ssl",3,1,0,0                    //Enable SSL and configure SSL context/connect index.
OK
AT+QMTCFG="version",3,1                    //Configure the MQTT version. Azure IoT Hub supports MQTT
                                           //v3.1.1 only.

OK
//Open a network for Azure MQTT client with TLS 1.2.
AT+QMTOPEN=3,"quectel-iot-hub.azure-devices.net",8883
OK

+QMTOPEN: 3,0                               //Open the MQTT client network successfully.
AT+QMTCONN=3,"quectel-device-x509self","quectel-iot-hub.azure-devices.net/quectel-device-

```

```

x509self"
OK

+QMTCONN: 3,0,0 //Connect the client to MQTT server successfully.
AT+QMTSUB=3,1,"devices/quectel-device-x509self/messages/devicebound/#",1
OK

+QMTSUB: 3,1,0,1

+QMTRECV: 3,2,"devices/quectel-device-x509self/messages/devicebound/%24.mid=419cfb05-70
53-4c7a-ba6a-68eb2c5077d6&%24.to=%2Fdevices%2Fquectel-device-x509self%2Fmessages%2F
deviceBound&iotHub-ack=full","hi quectel" //Received cloud-to-device messages.

//Publish fixed-length messages in data mode
AT+QMTPUB=3,0,0,0,"devices/quectel-device-x509self/messages/events/",17, "{"a":"1","b":"2"}"
OK

+QMTPUB: 3,0,0
//Publish variable-length messages in data mode
AT+QMTPUB=3,0,0,0,"devices/quectel-device-x509self/messages/events/"
>
hello azure iot hub //Input the data to be published and then tap "CTRL" + "Z" to send.
OK

+QMTPUB: 3,0,0
AT+QMTDISC=3 //Disconnect the client from MQTT server.
OK

+QMTDISC: 3,0 //Connection closed successfully.

```

#### NOTE

For details of **AT+QSSLCFG** and **AT+QSCLK**, see *document [1]* and *document [2]*.



# 7 Appendix References

**Table 5: Related Documents**

Document Name
[1] Quectel_BC660K-GL_SSL_Application_Note
[2] Quectel_BC660K-GL_AT_Commands_Manual

**Table 6: Terms and Abbreviations**

Abbreviation	Description
ACK	Acknowledgement
CA	Certificate Authority
ID	Identifier
IP	Internet Protocol
NB-IoT	Narrowband Internet of Things
MCU	Microprogrammed Control Unit
MQTT	Message Queuing Telemetry Transport
PDP	Packet Data Protocol
PEM	Privacy Enhanced Mail
QoS	Quality of Service
TA	Terminal Adapter
SSL	Secure Socket Layer
TCP	Transmission Control Protocol

TLS	Transport Layer Security
URC	Unsolicited Result Code
UTF	Unicode Transformation Format