GSM MQTT Application Note

GSM/GPRS/GNSS Module Series

Rev. GSM_MQTT_Application_Note_V1.3

Date: 2020-04-26

Status: Released

www.quectel.com

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

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

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

About the Document

Revision History

Version	Date	Author	Description
1.0	2017-06-12	Louis GU/ Sherlock ZHAO	Initial
1.1	2017-10-23	Louis GU	Added command AT+QMTCFG=”ALIAUTH” for Ali cloud device authentication (Chapter 3.2.1)
1.2	2018-08-31	Jaryoung LI/ Sandy YE	1. Updated the applicable modules of MQTT (Chapter 1) 2. Added Write Command AT+QMTCFG=“VERSION” for configuring MQTT version (Chapter 3.2.1) 3. Added Write Command for sending data with fixed length (Chapter 3.2.8) 4. Updated the maximum response time for AT+QMTCONN/QMTSUB/QMTUNS/QMTPUB (Chapter 3.2.4, 3.2.6, 3.2.7 and 3.2.8)
1.3	2020-04-26	Jaryoung LI	1. Updated applicable modules (Chapter 1) 2. Updated MQTT data interaction diagram (Chapter 2) 3. Updated the overall description of AT commands (Chapter 3.2) 4. Added description of <err> in the parameter table (Chapter 3.2)

Contents

Table Index

1 Introduction

MQTT 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 GSM modules through AT commands.

1.1. Applicable Modules

This document is applicable to following Quectel modules.

Table 1: Applicable Modules

Module Series	Module
Mxx	M66
	M95
	M65
	M08-R
MCxx	MC60
	MC90

2 MQTT Data Interaction

This chapter gives the data interaction mechanism of MQTT function.

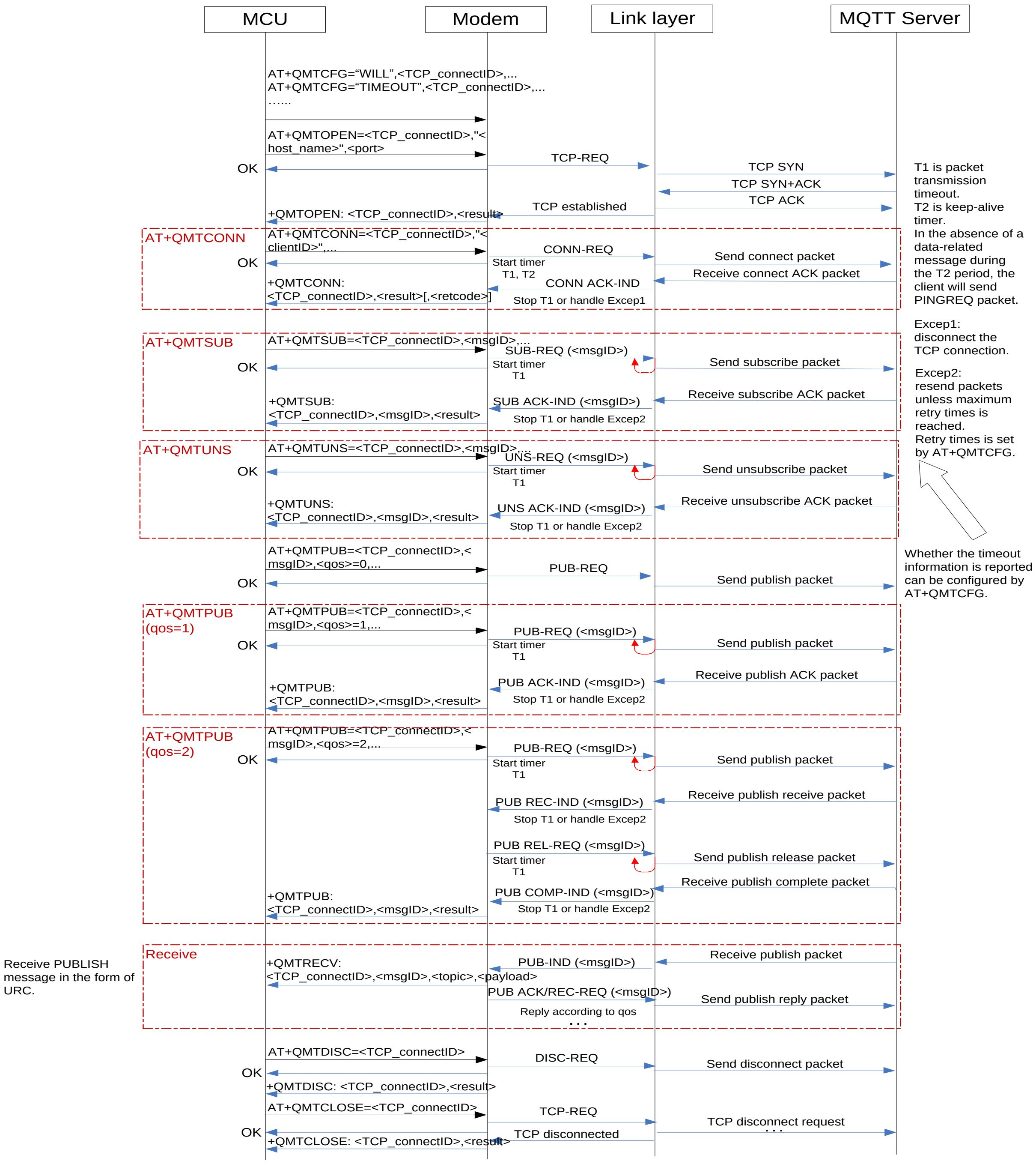
MCU
Modem
Link layer
MQTT Server
AT+QMTCFG="WILL",<TCP_connectID>,...
AT+QMTCFG="TIMEOUT",<TCP_connectID>,...
……
AT+QMTOPEN=<TCP_connectID>,"<host_name>",<port>
TCP-REQ
OK
TCP SYN
TCP SYN+ACK
TCP ACK
TCP established
+QMTOPEN: <TCP_connectID>,<result>
AT+QMTCONN
AT+QMTCONN=<TCP_connectID>,"<clientID>",...
CONN-REQ
OK
Start timer T1, T2
Send connect packet
Receive connect ACK packet
CONN ACK-IND
+QMTCONN: <TCP_connectID>,<result>[,<retcode>]
Stop T1 or handle Excep1
T1 is packet transmission timeout.
T2 is keep-alive timer.
In the absence of a data-related message during the T2 period, the client will send PINGREQ packet.
Excep1: disconnect the TCP connection.
Excep2: resend packets unless maximum retry times is reached.
Retry times is set by AT+QMTCFG.
AT+QMTSUB
AT+QMTSUB=<TCP_connectID>,<msgID>,...
SUB-REQ (<msgID>)
OK
Start timer T1
Send subscribe packet
+QMTSUB: <TCP_connectID>,<msgID>,<result>
SUB ACK-IND (<msgID>)
Receive subscribe ACK packet
Stop T1 or handle Excep2
AT+QMTUNS
AT+QMTUNS=<TCP_connectID>,<msgID>,...
UNS-REQ (<msgID>)
OK
Start timer T1
Send unsubscribe packet
+QMTUNS: <TCP_connectID>,<msgID>,<result>
UNS ACK-IND (<msgID>)
Receive unsubscribe ACK packet
Stop T1 or handle Excep2
AT+QMTPUB=<TCP_connectID>,<msgID>,<qos>=0,...
PUB-REQ
OK
Send publish packet
Whether the timeout information is reported can be configured by AT+QMTCFG.
AT+QMTPUB (qos=1)
AT+QMTPUB=<TCP_connectID>,<msgID>,<qos>=1,...
PUB-REQ (<msgID>)
OK
Start timer T1
Send publish packet
+QMTPUB: <TCP_connectID>,<msgID>,<result>
PUB ACK-IND (<msgID>)
Receive publish ACK packet
Stop T1 or handle Excep2
AT+QMTPUB (qos=2)
AT+QMTPUB=<TCP_connectID>,<msgID>,<qos>=2,...
PUB-REQ (<msgID>)
OK
Start timer T1
Send publish packet
PUB REC-IND (<msgID>)
Receive publish receive packet
Stop T1 or handle Excep2
PUB REL-REQ (<msgID>)
Start timer T1
Send publish release packet
Receive publish complete packet
+QMTPUB: <TCP_connectID>,<msgID>,<result>
PUB COMP-IND (<msgID>)
Stop T1 or handle Excep2
Receive
Receive PUBLISH message in the form of URC.
+QMTRECV: <TCP_connectID>,<msgID>,<topic>,<payload>
PUB-IND (<msgID>)
Receive publish packet
PUB ACK/REC-REQ (<msgID>)
Send publish reply packet
Reply according to qos
AT+QMTDISC=<TCP_connectID>
DISC-REQ
Send disconnect packet
OK
+QMTDISC: <TCP_connectID>,<result>
AT+QMTCLOSE=<TCP_connectID>
TCP-REQ
TCP disconnect request
OK
TCP disconnected
+QMTCLOSE: <TCP_connectID>,<result>

Figure 1: MQTT Data Interaction Diagram

3 MQTT 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

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

3.2. Description of AT Commands

3.2.1. AT+QMTCFG Configure Optional Parameters of MQTT

The command configures optional parameters of MQTT.

AT+QMTCFG Configure Optional Parameters of MQTT	
Test Command **AT+QMTCFG=?**	Response **+QMTCFG: "<config_type>",(**range of supported **<TCP_connectID>**s**)[,<value>…]** **OK**
Write Command Configure Will information **AT+QMTCFG="WILL",<TCP_connectID>[,<will_fg>[,<will_qos>,<will_retain>,"<will_topic>","<will_msg>"]]**	Response If **<will_fg>**, **<will_qos>**, **<will_retain>**, **"<will_topic>"** and **"<will_msg>"** are omitted, query the Will information:If **<will_fg>**, **<will_qos>**, **<will_retain>**, **<will_topic>** and **<will_msg>** are omitted, query the "WILL" information: **+QMTCFG: <will_fg>[,<will_qos>,<will_retain>,"<will_topic>","<will_msg>"]** **OK** If **<will_fg>**, **<will_qos>**, **<will_retain>**, **"<will_topic>"** and **"<will_msg>"** are specified, configure Will information: **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure timeout of message delivery **AT+QMTCFG="TIMEOUT",<TCP_connectID>[,<pkttimeout>[,<retry_times>[,<timeout_notice>]]]**	ResponseIf **<pkttimeout>**, **<retry_times>** and **<timeout_notice>** are omitted, query the "TIMEOUT" information: If **<pkttimeout>**, **<retry_times>** and **<timeout_notice>** are omitted, query the timeout of message delivery: **+QMTCFG: <pkttimeout>,<retry_times>,<timeout_notice>** **OK** If **<pkttimeout>**, **<retry_times>** and **<timeout_notice>** are specified, configure the timeout of message delivery: **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**

Write Command Configure session type **AT+QMTCFG="SESSION",<TCP_connectID>[,<clean_session>]**	Response If **<clean_session>** is omitted, query the session type:If **<clean_session>** is omitted, query the "SESSION" information: **+QMTCFG: <clean_session>** **OK** If **<clean_session>** is specified, configure the session type: **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure keep-alive time **AT+QMTCFG="KEEPALIVE",<TCP_connectID>[,<keep_alive_time>]**	Response If **<keep_alive_time>** is omitted, query the keep-alive time: **+QMTCFG: <keep_alive_time>** **OK** If **<keep_alive_time>** is specified, configure keep-alive time: **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure SSL secure connection. **AT+QMTCFG="SSL",<TCP_connectID>[,<SSL_enable>[,<ctxindex>]]**	Response If **<SSL_enable>** is omitted, query the SSL information: **+QMTCFG: <SSL_enable>[,<ctxindex>]** **OK** If **<SSL_enable>** is specified, configure SSL secure connection: **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure Ali device information for Ali Cloud **AT+QMTCFG="ALIAUTH",<TCP_connectID>[,"<product_key>","<device_name>","<device_secret>"]**	Response If **"<product_key>"**, **"<device_name>"** and **"<device_secret>"** are omitted, query the device information: **[+QMTCFG: "<product_key>","<device_name>","<device_secret>"]** **OK**

	If **"<product_key>"**, **"<device_name>"** and **"<device_secret>"** are specified, configure Ali device information for Ali Cloud: **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure MQTT version **AT+QMTCFG="VERSION",<TCP_connectID>[,<version_num>]**	Response If **<version_num>** is omitted, query MQTT version:If **<version_num>** is omitted, query the "VERSION" information: **+QMTCFG: <version_num>** **OK** If **<version_num>** is specified, configure MQTT version: **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

<config_type>	String type. The type of configuration. It can be any of the following types: WILL TIMEOUT SESSION KEEPALIVE SSL ALIAUTH VERSION
<TCP_connectID>	Integer type. MQTT socket identifier. Range: 0–5.
<value>	Configuration value of MQTT optional parameters.
<will_fg>	Integer type. Configure the Will flag. 0 Ignore the Will flag configuration 1 Require the Will flag configuration
<will_qos>	Integer type. Quality of service for message delivery. 0 At most once 1 At least once 2 Exactly once

<will_retain> Integer type. The Will retain flag is only used on PUBLISH messages.
0 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.

<will_topic> String type. Will topic string.

<will_msg> 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.

<pkttimeout> Integer type. Timeout of the packet delivery. Range: 1–60. Default: 5. Unit: s.

<retry_times> Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.

<timeout_notice> Integer type. Whether to report timeout message when transmitting package.
0 Not report
1 Report

<clean_session> Integer type. Configure the session type.
0 The server must store the subscriptions of the client after it disconnects.
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: 120. Unit: s. 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.
0 The client will not be disconnected.

<SSL_enable> Integer type. Configure the MQTT SSL mode.
0 Use normal TCP connection for MQTT
1 Use SSL TCP secure connection for MQTT

<ctxindex> Integer type. SSL context index. Range: 0–5.
The parameter must be omitted when **<SSL_enable>**=0.

<product_key> String type. Product key obtained from Ali Cloud.

<device_name> String type. Device name obtained from Ali Cloud.

<device_secret> String type. Device secret obtained from Ali Cloud.

<version_num> Integer type. MQTT version number.
0 MQTT Version 3.1
1 MQTT Version 3.1.1

<err> Integer type. Error codes. For details, please refer to ***Chapter 7***.

NOTES

1. If **<will_fg>**=1, **<will_qos>**, **<will_retain>**, **"<will_topic>"** and **"<will_msg>"** 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, **<ctxindex>** must be specified. Also, customers need to use **AT+QSSLCFG** 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.
4. Note that the configured timeout period should not be too short, so that no timeout will occur during message transmission.
5. **AT+QMTCFG="ALIAUTH"** is only used for Ali Cloud. If it is configured, **"<username>"** and **"<password>"** in **AT+QMTCONN** can be omitted.

3.2.2. AT+QMTOPEN Open a Network for MQTT Client

The command opens a network for MQTT client.

AT+QMTOPEN Open a Network for MQTT Client	
Test Command **AT+QMTOPEN=?**	Response **+QMTOPEN: (**range of supported **<TCP_connectID>**s**),"<host_name>",<port> (**range of supported **<TCP_connectID>**s**),<host_name>,(**range of supported **<port>**s**)** **OK**
Read Command **AT+QMTOPEN?**	Response **[+QMTOPEN: <TCP_connectID>,"<host_name>",<port>]** **OK**
Write Command **AT+QMTOPEN=<TCP_connectID>,"<host_name>",<port>**	Response **OK** **+QMTOPEN: <TCP_connectID>,<result>** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	75 s, determined by network.
Characteristics	/

Parameter

<TCP_connectID>	Integer type. MQTT socket 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 of the server. Range: 1–65535.
<result>	Integer type. Result of the command execution. -1 Failed to open network 0 Opened network successfully 1 Wrong parameter 2 MQTT identifier is occupied 3 Failed to activate PDP 4 Failed to parse domain name 5 Network disconnection error
<err>	Integer type. Error codes. For details, please refer to ***Chapter 7***.

3.2.3. AT+QMTCLOSE Close a Network for MQTT Client

The command closes a network for MQTT client.

AT+QMTCLOSE Close a Network for MQTT Client	
Test Command **AT+QMTCLOSE=?**	Response **+QMTCLOSE: (**range of supported **<TCP_connectID>**s**)** **OK**
Write Command **AT+QMTCLOSE=<TCP_connectID>**	Response **OK** **+QMTCLOSE: <TCP_connectID>,<result>** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Range: 0–5.
<result>	Integer type. Result of the command execution. -1 Failed to close network 0 Network closed successfully

<err>	Integer type. Error codes. For details, please refer to ***Chapter 7***.

3.2.4. AT+QMTCONN Connect a Client to MQTT Server

The command requests a connection to MQTT server from the client. 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.

AT+QMTCONN Connect a Client to MQTT Server	
Test Command **AT+QMTCONN=?**	Response **+QMTCONN: (**range of supported **<TCP_connectID>**s)**,"<clientID>"[,"<username>"[,"<password>"]]** **OK**
Read Command **AT+QMTCONN?**	Response **[+QMTCONN: <TCP_connectID>,<state>]** **OK**
Write Command **AT+QMTCONN=<TCP_connectID>,"<clientID>"[,"<username>"[,"<password>"]]**	Response **OK** **+QMTCONN: <TCP_connectID>,<result>[,<retcode>]** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	**<pkttimeout>** × (**<retry_times>** + 1) The default response time is 20 s, and the actual response time is determined by network.
Characteristics	/

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Range: 0–5.
<clientID>	String type. 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 Sent packet successfully and received ACK 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
<retcode>	Integer type. Connect 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
<pkttimeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default: 5. Unit: s.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.
<err>	Integer type. Error codes. For details, please refer to ***Chapter 7***.

NOTE

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.

3.2.5. AT+QMTDISC Disconnect a Client from MQTT Server

The command 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.

AT+QMTDISC Disconnect a Client from MQTT Server	
Test Command **AT+QMTDISC=?**	Response **+QMTDISC: (**range of supported **<TCP_connectID>**s**)** **OK**
Write Command **AT+QMTDISC=<TCP_connectID>**	Response **OK** **+QMTDISC: <TCP_connectID>,<result>** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Range: 0–5.
<result>	Integer type. Result of the command execution. -1 Failed to close connection 0 Connection closed successfully
<err>	Integer type. Error codes. For details, please refer to ***Chapter 7***.

3.2.6. AT+QMTSUB Subscribe to Topics

The command subscribes 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.

AT+QMTSUB Subscribe to Topics	
Test Command **AT+QMTSUB=?**	Response **+QMTSUB: (**range of supported **<TCP_connectID>**s)**,<msgID>,"<topic>",<qos>[,"<topic>",<qos>...]** **OK**
Write Command **AT+QMTSUB=<TCP_connectID>,<msgID>,"<topic1>",<qos1>[,"<topic2>",<qos2>…]**	Response **OK** **+QMTSUB: <TCP_connectID>,<msgID>,<result>[,<value>]** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	**<pkttimeout>** × (**<retry_times>** + 1) The default response time is 20 s, and the actual response time is determined by network.
Characteristics	/

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Range: 0–5.
<msgID>	Integer type. Message identifier of packet. Range: 0–65535.
<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>	Integer type. If **<result>**=0, it is a vector of granted QoS levels when the subscription is accepted by the server, or 128 which indicates the subscription is rejected by the server. If **<result>**=1, it means the times of packet retransmission. If **<result>**=2, it will not be presented.
<pkttimeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default: 5. Unit: s.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default: 3.
<err>	Integer type. Error codes. For details, please refer to ***Chapter 7***.

NOTE

The **<msgID>** is only presented in messages where the QoS bits in the fixed header indicate QoS level 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, which is not compulsory in actual practice.

3.2.7. AT+QMTUNS Unsubscribe from Topics

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

AT+QMTUNS Unsubscribe from Topics	
Test Command **AT+QMTUNS=?**	Response **+QMTUNS: (**range of supported **<TCP_connectID>**s**),<msgID>,"<topic>"[,"<topic>"...]** **OK**
Write Command **AT+QMTUNS=<TCP_connectID>,<msgID>,"<topic1>"[,"<topic2>"…]**	Response **OK** **+QMTUNS: <TCP_connectID>,<msgID>,<result>** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	**<pkttimeout>** × (**<retry_times>** + 1) The default response time is 20 s, and the actual response time is determined by network.

Characteristics	/

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Range: 0–5.
<msgID>	Integer type. Message identifier of packet. Range: 0–65535.
<topic>	String type. Topic that the client wants to subscribe to or unsubscribe from.
<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
<pkttimeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default: 5. Unit: s.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default: 3.
<err>	Integer type. Error codes. For details, please refer to ***Chapter 7***.

3.2.8. AT+QMTPUB Publish Messages

The command publishes messages from 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 will be sent by the server to the client as a PUBLISH message.

AT+QMTPUB Publish Message	
Test Command **AT+QMTPUB=?**	Response **+QMTPUB: (**range of supported **<TCP_connectID>**s**),<msgID>,<qos>,<retain>,"<topic>","<msg>"** **OK**
Write Command Send data with changeable length **AT+QMTPUB=<TCP_connectID>,<msgID>,<qos>,<retain>,"<topic>"**	Response **>** After the response **>**, input the data to be sent. Tap CTRL+Z to send, and tap ESC to cancel the operation. **OK** **+QMTPUB: <TCP_connectID>,<msgID>,<result>[,<value>]** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Send data with fixed length **AT+QMTPUB=<TCP_connectID>,<**	Response **>** After the response **>**, input the data until the data length equals

msgID>,<qos>,<retain>,"<topic>",<size>	**<size>.** **OK** **+QMTPUB: <TCP_connectID>,<msgID>,<result>[,<value>]** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	**<pkttimeout>** × (**<retry_times>** + 1) The default response time is 20 s, and the actual response time is determined by network.
Characteristics	/

Parameter

<TCP_connectID> Integer type. MQTT socket 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.
0 At most once
1 At least once
2 Exactly once

<retain> Integer type. Whether the server will retain the message after it has been delivered to the current subscribers or not.
0 Not retain
1 Retain

<topic> String type. Topic that needs to be published.

<msg> String type. Message that needs to be published.

<result> Integer type. Result of the command execution.
0 Sent packet successfully and received ACK from server (message published when **<qos>**=0 does not require ACK)
1 Packet retransmission
2 Failed to send packet

<value> Integer type. If **<result>** is 1, it means the times of packet retransmission.
If **<result>** is 0 or 2, it will not be presented.

<size> Integer type. Data length to be specified. Range: 1–1548.

<pkttimeout> Integer type. Timeout of the packet delivery. Range: 1–60. Default: 5. Unit: s.

<retry_times> Integer type. Retry times when packet delivery times out. Range: 0–10. Default: 3.

<err> Integer type. Error codes. For details, please refer to ***Chapter 7***.

NOTES

1. If this command is executed successfully and gets **OK** back, the client can continue to publish new packets. The maximum quantity of transmitting packet should not be greater than the inflight window size: 5. Otherwise, **+CME ERROR: 8512** will be returned. For more details, please refer to ***Chapter 7***.
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 1548 bytes at a time and 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 the server publishes messages to a subscriber, **+QMTRECV: <TCP_connectID>,<msgID>,<topic>,<payload>** will be returned to notify the host to read thereceived data that is reported from MQTT server. For more details, please refer to ***Chapter 7.***

4 MQTT Related URCs

This chapter gives MQTT related URCs and their descriptions.

Table 3: MQTT Related URCs

SN	URC Format	Description
[1]	**+QMTSTAT: <TCP_connectID>,<err_code>**	When the state of MQTT link layer is changed, the client will close the MQTT connection and report the URC.
[2]	**+QMTRECV: <TCP_connectID>,<msgID>,<topic>,<payload>**	The client has received the packet data from MQTT server.

4.1. +QMTSTAT 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 Indicate State Change in MQTT Link Layer	
+QMTSTAT: <TCP_connectID>,<err_code>	When the state of MQTT link layer is changed, the client will close the MQTT connection and report the URC.

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier.
<err_code>	Integer type. Error codes. Please refer to the table below for details.

Table 4: Error Codes of the URC

<err_code>	Description	How to do
1	Connection is closed or reset by peer.	Execute **AT+QMTOPEN** command and reopen 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.	1. Check whether the inputted user name and password are correct. 2. Make sure the client ID is not used. 3. Reopen MQTT connection and try to send CONNECT packet to server again.
4	Receiving CONNACK packet timed out or failed.	1. Check whether the inputted user name and password are correct. 2. Make sure the client ID is not used. 3. Reopen MQTT connection and try to send CONNECT packet to server again.
5	Client sends DISCONNECT packet to sever but the server closes MQTT connection.	This is a normal process.
6	Client closes MQTT connection due to packet sending failure all the time.	1. Make sure the data is correct. 2. Try to reopen MQTT connection since there may be network congestion or an error.
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 Notify the Host to Read MQTT Packet Data

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

+QMTRECV Notify the Host to Read MQTT Packet Data	
+QMTRECV: <TCP_connectID>,<msgID>,<topic>,<payload>	Notify the host to read the received data that is reported from MQTT server.

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier.
<msgID>	Integer type. The message identifier of packet.
<topic>	String type. The topic that received from MQTT server.
<payload>	String type. The payload that relates to the topic name.

5 Examples

This chapter gives the examples to explain how to use MQTT commands.

5.1. Example of MQTT Operation without SSL

```
//Configure Ali device information for Ali Cloud.
AT+QMTCFG="ALIAUTH",0,"oyjtmPl5a5j","MQTT_TEST","wN9Y6pZSIly7Exa5qVzcmigEGO4kAazZ"
OK
AT+QMTOPEN=?
+QMTOPEN: (0-5), "<host_name>",<port>

OK

//Open a network for MQTT client.
AT+QMTOPEN=0,"iot-as-mqtt.cn-shanghai.aliyuncs.com",1883
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 Ali Cloud is connected, customers can use AT+QMTCFG="ALIAUTH" command to configure the device information in advance. Then "<username>" and "<password>" can be omitted here.
AT+QMTCONN=0,"clientExample"
OK

+QMTCONN: 0,0,0                         //Connected the client to MQTT server successfully.
```

```
AT+QMTSUB=?
+QMTSUB: (0-5),<msgID>,"<topic>",<qos>[,"<topic>",<qos>...]

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=0,2,"topic/example"
OK

+QMTUNS: 0,2,0
AT+QMTPUB=?
+QMTPUB: (0-5),<msgID>,<qos>,<retain>,"<topic>","<msg>"

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
                                      send it. The maximum length of the data is 1548 bytes and the data
                                      beyond 1548 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.

//Disconnect a client from MQTT server.
```

```
AT+QMTDISC=0
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 "server and client authentication", store server root CA certificate to RAM.
AT+QSECWRITE="RAM:cacert.pem",1758,100
CONNECT
<Input the cacert.pem data, the size is 1758 bytes>
+QSECWRITE: 1758,384a

OK

//If SSL authentication mode is "server and client authentication", store CC certificate to RAM.
AT+QSECWRITE="RAM:client.pem",1220,100
CONNECT
<Input the client.pem data, the size is 1220 bytes>
+QSECWRITE: 1220,2d53

OK

//If SSL authentication mode is "server and client authentication", store CK certificate to RAM.
AT+QSECWRITE="RAM:user_key.pem",1679,100
CONNECT
<Input the user_key.pem data, the size is 1679 bytes>
+QSECWRITE: 1679,335f

OK

//Configure server root CA certificate.
AT+QSSLCFG="cacert",2,"RAM:cacert.pem"
OK

//Configure CC certificate.
AT+QSSLCFG="clientcert",2,"RAM:client.pem"
OK
```

```
//Configure CK certificate.
AT+QSSLCFG="clientkey",2,"RAM:user_key.pem"
OK

//Configure SSL parameters.
AT+QSSLCFG="seclevel",2,2              //SSL authentication mode: server and client authentication
OK
AT+QSSLCFG="sslversion",2,4            //SSL authentication version
OK
AT+QSSLCFG="ciphersuite",2,"0xFFFF"   //Cipher suite
OK
AT+QSSLCFG="ignorertctime",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,"M35_0206"
OK

+QMTCONN: 0,0,0

//Subscribe to topics.
AT+QMTSUB=0,1,"$aws/things/M35_0206/shadow/update/accepted",1
OK

+QMTSUB: 0,1,0,1

//Publish messages.
AT+QMTPUB=0,1,1,0,"$aws/things/M35_0206/shadow/update/accepted"
>This is publish data from client
OK

+QMTPUB: 0,1,0

//If a client subscribes to a topic named "$aws/things/M35_0206/shadow/update/accepted" and other
devices publish the same topic to the server, the module will report the following information.
+QMTRECV: 0,1,$aws/things/M35_0206/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 5: Related Document

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

Table 6: Terms and Abbreviations

Abbreviation	Description
ACK	Acknowledgement
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

7 Appendix B Error Codes

<err> indicates an error related to mobile equipment or network. Details about **<err>** are described in the following table.

Table 7: Summary of Error Codes

Code of <err>	Meaning
8503	MQTT link 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 remaining transmission buffer space.
8509	MQTT buffer overflow
8510	MQTT out of memory
8511	MQTT memory error
8600	MQTT unknown error
8512	MQTT maximum quantity of transmitting packets exceeds the inflight window size.