BG95&BG77 MQTT Application Note

LPWA Module Series

Rev. BG95&BG77_MQTT_Application_Note_V1.0

Date: 2019-08-12

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, China 200233
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

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2019-08-12	Lane HAO	Initial

Contents

Table Index

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 and BG77 modules through AT commands.

2 MQTT Data Interaction

This chapter gives the data interaction mechanism of MQTT function.

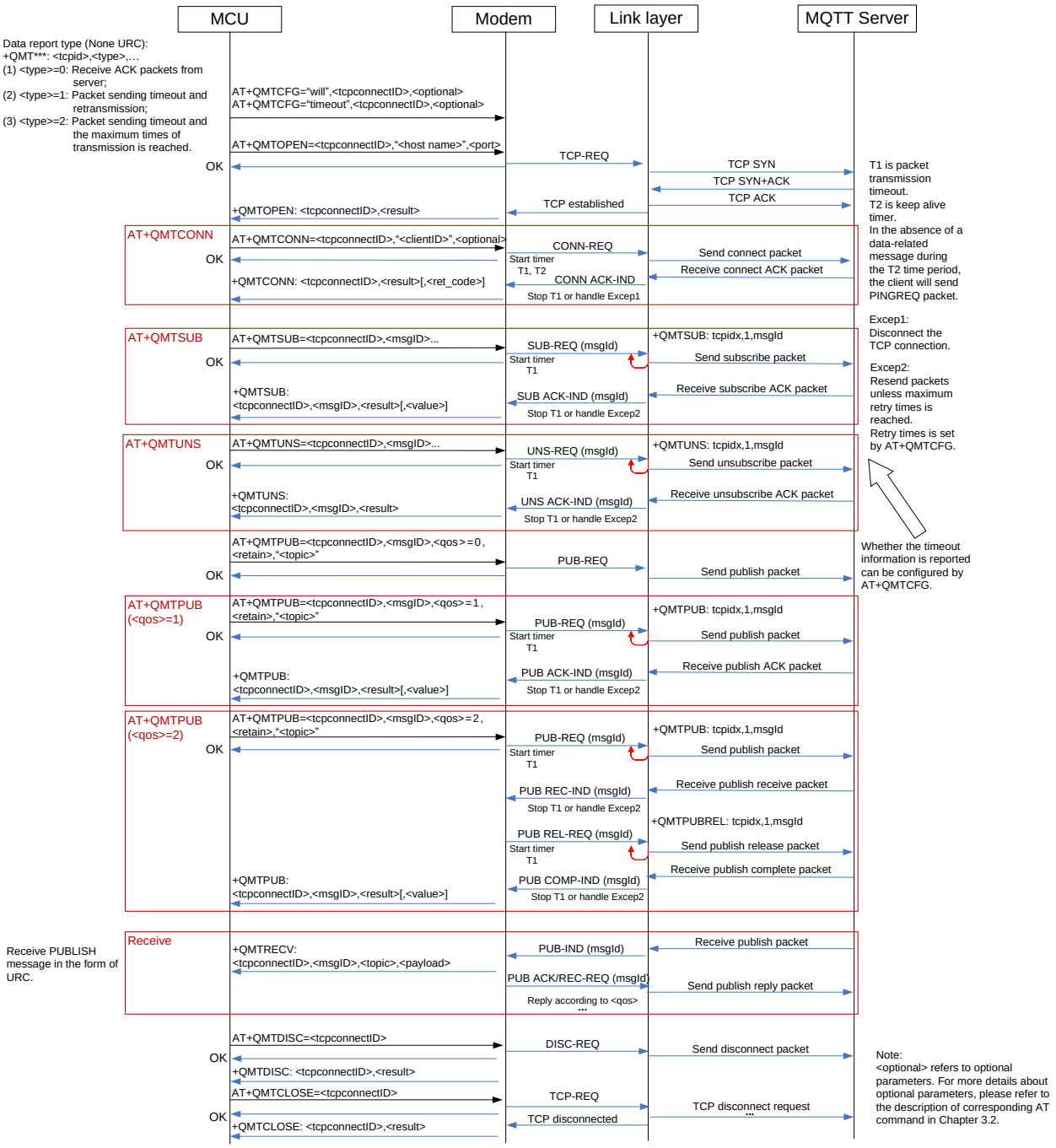

Figure 1: MQTT Data Interaction Diagram

3 MQTT Related AT Commands

This chapter presents the AT commands for operating MQTT function.

3.1. AT Command Syntax

Table 1: Types of AT Commands and Responses

Test Command	**AT+<x>=?**	This command returns the list of parameters and value ranges set by the corresponding Write Command or internal processes.
Read Command	**AT+<x>?**	This command returns the currently set value of the parameter or parameters.
Write Command	**AT+<x>=<…>**	This command sets the user-definable parameter values.
Execution Command	**AT+<x>**	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

The command is used to configure optional parameters of MQTT.

AT+QMTCFG Configure Optional Parameters of MQTT	
Test Command **AT+QMTCFG=?**	Response **+QMTCFG: "version",(0-5),(3,4)** **+QMTCFG: "pdpcid",(0-5),(1-16)** **+QMTCFG: "ssl",(0-5),(0,1),(0-5)** **+QMTCFG: "keepalive",(0-5),(0-3600)** **+QMTCFG: "session",(0-5),(0,1)** **+QMTCFG: "prefix",(0-5),<ipv6_prefix>,(32,40,48,56,64,96)** **+QMTCFG: "timeout",(0-5),(1-60),(0-10),(0,1)** **+QMTCFG: "will",(0-5),(0,1),(0-2),(0,1),<will_topic>,<will_m**

	essage> **+QMTCFG: "recv/mode",(0-5),(0,1),(0,1)** **+QMTCFG: "aliauth",(0-5),<product_key>,<device_name>, <device_secret>** **OK**
Write Command Configure/query the MQTT protocol version **AT+QMTCFG="version",<tcpconnectID>[,<vsn>]**	Response If **<vsn>** is present, configure the MQTT protocol version: **OK** If **<vsn>** is omitted, query the MQTT protocol version: **+QMTCFG: "version",<vsn>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure/query the PDP to be used by the MQTT client **AT+QMTCFG="pdpcid",<tcpconnectID>[,<cid>]**	Response If **<cid>** is present, configure the PDP to be used by the MQTT client: **OK** If **<cid>** is omitted, query the PDP used by the MQTT client: **+QMTCFG: "pdpcid",<cid>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure/query the MQTT SSL mode and SSL context index **AT+QMTCFG="ssl",<tcpconnectID>[,<sslenable>[,<ctxindex>]]**	Response If **<sslenable>** and **<ctxindex>** are present, configure the MQTT SSL mode and SSL context index: **OK** If **<sslenable>** and **<ctxindex>** are omitted, query the MQTT SSL mode and SSL context index: **+QMTCFG: "ssl",<sslenable>[,<ctxindex>]** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure/query the keep-alive time	Response If **<keep-alive time>** is present, configure the keep-alive time:

AT+QMTCFG="keepalive",<tcpconnectID>[,<keep-alive time>]	**OK** If **<keep-alive time>** is omitted, query the keep-alive time: **+QMTCFG: "keepalive",<keep-alive time>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure/query the session type **AT+QMTCFG="session",<tcpconnectID>[,<clean_session>]**	Response If **<clean_session>** is present, configure the session type: **OK** If **<clean_session>** is omitted, query the session type: **+QMTCFG: "session",<clean_session>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure/query the prefix of IPv4-converted IPv6 address **AT+QMTCFG="prefix",<tcpconnectID>[,<ipv6_prefix>,<prefix_length>]**	Response If **<ipv6_prefix>** and **<prefix_length>** are present, configure the prefix of IPv4-converted IPv6 address: **OK** If **<ipv6_prefix>** and **<prefix_length>** are omitted, query the prefix of IPv4-converted IPv6 address: **+QMTCFG: "prefix",<ipv6_prefix>,<prefix_length>** **OK**
Write Command Configure/query the timeout of message delivery **AT+QMTCFG="timeout",<tcpconnectID>[,<pkt_timeout>[,<retry_times>][,<timeout_notice>]]**	Response If all parameters are present, configure the timeout of message delivery: **OK** If **<pkt_timeout>**, **<retry_times>** and **<timeout_notice>** are omitted, query the timeout of message delivery: **+QMTCFG: "timeout",<pkt_timeout>,<retry_times>,<timeout_notice>** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**

Write Command Configure/query the Will information **AT+QMTCFG="will",<tcpconnectID>[,<will_fg>[,<will_qos>,<will_retain>,<will_topic>,<will_msg>]]**	Response If all parameters are present, configure the Will information: **OK** If **<will_fg>**, **<will_qos>**, **<will_retain>**, **<will_topic>** and **<will_msg>** are omitted, query the Will information: **+QMTCFG: "will",<will_fg>[,<will_qos>,<will_retain>,<will_topic>,<will_msg>]** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure/query the MQTT message receiving mode when the data is received from server **AT+QMTCFG="recv/mode",<tcpconnectID>[,<msg_recv_mode>[,<msg_len_enable>]]**	Response If **<msg_recv_mode>** and **<msg_len_enable>** are present, configure the MQTT message receiving mode: **OK** If **<msg_recv_mode>** and **<msg_len_enable>** are omitted, query the MQTT message receiving mode. **+QMTCFG: "recv/mode",<msg_recv_mode>[,<msg_len_enable>]** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Configure Alibaba device information for AliCloud **AT+QMTCFG="aliauth",<tcpconnectID>[,<product_key>,<device_name>,<device_secret>]**	Response If all parameters are present, configure the device information: **OK** If **<product_key>,<device_name>,<device_secret>** are omitted, query the device information: **[+QMTCFG: "aliauth",<product_key>,<device_name>,<device_secret>]** **OK** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300ms

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<vsn>	MQTT protocol version. 3 MQTT v3.1 4 MQTT v3.1.1
<cid>	The PDP to be used by the MQTT client. The range is 1-16. The default value is 1.
<will_fg>	Configure the Will flag. 0 Ignore the Will flag configuration 1 Require the Will flag configuration
<will_qos>	Quality of service for message delivery. 0 At most once 1 At least once 2 Exactly once
<will_retain>	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>	Will topic string.
<will_msg>	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.
<pkt_timeout>	Timeout of the packet delivery. The range is 1-60. The default value is 5. Unit: second.
<retry_times>	Retry times when packet delivery times out. The range is 0-10. The default value is 3.
<timeout_notice>	0 Not report timeout message when transmitting packet 1 Report timeout message when transmitting packet
<clean_session>	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”.
<ipv6_prefix>	The prefix of IPv4-converted IPv6 address.
<prefix_length>	The length of the prefix string. The value can be 32, 40, 48, 56, 64 or 96. Unit: Byte.
<keep-alive time>	Keep-alive time. The range is 0-3600. The default value is 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. 0 No limitations on the maximum time interval between messages received from a client
<sslenable>	MQTT SSL mode. 0 Use normal TCP connection for MQTT

	1 Use SSL TCP secure connection for MQTT
<ctxindex>	SSL context index. The range is 0-5.
<msg_recv_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_msg>** must be present. 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 present, 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"** command is only used for AliCloud. If it is configured, the parameters **<username>** and **<password>** in command **AT+QMTCONN** can be omitted.

3.2.2. AT+QMTOPEN Open a Network for MQTT Client

The command is used to open a network for MQTT client.

AT+QMTOPEN Open a Network for MQTT Client	
Test Command **AT+QMTOPEN=?**	Response **+QMTOPEN: (**list of supported **<tcpconnectID>**s**),<host_name>,(**list of supported **<port>**s**)** **OK**
Read Command **AT+QMTOPEN?**	Response **[+QMTOPEN: <tcpconnectID>,<host_name>,<port>]** **OK**
Write Command **AT+QMTOPEN=<tcpconnectID>,<host_name>,<port>**	Response **OK** **+QMTOPEN: <tcpconnectID>,<result>**

	If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	75s, determined by network

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<host_name>	The address of the server. It could be an IP address or a domain name. The maximum length is 100 bytes.
<port>	The port number of the server. The range is 0-65535.
<result>	Result of the command execution. -1 Failed to open network 0 Network opened successfully 1 Wrong parameter 2 MQTT socket identifier is occupied 3 Failed to activate PDP 4 Failed to parse domain name 5 Network disconnection error

3.2.3. AT+QMTCLOSE Close a Network for MQTT Client

The command is used to close a network for MQTT client.

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

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.

<result>	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

The 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.

AT+QMTCONN Connect a Client to MQTT Server	
Test Command **AT+QMTCONN=?**	Response **+QMTCONN: (**list of supported **<tcpconnectID>**s**),<clientID>[**,**<username>[**,**<password>]]** **OK**
Read Command **AT+QMTCONN?**	Response **[+QMTCONN: <tcpconnectID>,<state>]** **OK**
Write Command **AT+QMTCONN=<tcpconnectID>,<clientID>[,<username>[,<password>]]**	Response **OK** **+QMTCONN: <tcpconnectID>,<result>[,<ret_code>]** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	**<pkt_timeout>** (default 5s), determined by network

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<clientID>	The client identifier string.
<username>	User name of the client. It can be used for authentication.
<password>	Password corresponding to the user name of the client. It can be used for authentication.
<result>	Result of the command execution. 0 Packet sent successfully and ACK received from server 1 Packet retransmission 2 Failed to send packet
<state>	MQTT connection state. 1 MQTT is initializing

	2 MQTT is connecting 3 MQTT is connected 4 MQTT is disconnecting
<ret_code>	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>	Timeout of the packet delivery. The range is 1-60. The default value is 5. Unit: second.
<err>	The error code of operation. For more details, please refer to ***Chapter 4***.

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 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.

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

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.

<result>	Result of the command execution -1 Failed to close connection 0 Connection closed successfully
<err>	The error code of operation. For more details, please refer to ***Chapter 4***.

3.2.6. AT+QMTSUB Subscribe to Topics

The 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.

AT+QMTSUB Subscribe to Topics	
Test Command **AT+QMTSUB=?**	Response **+QMTSUB: (**list of supported **<tcpconnectID>**s**),(**list of supported **<msgID>**s**),<topic>,(**list of supported **<qos>**s**)** **OK**
Write Command **AT+QMTSUB=<tcpconnectID>,<msgID>,<topic1>,<qos1>[,<topic2>,<qos2>…]**	Response **OK** **+QMTSUB: <tcpconnectID>,<msgID>,<result>[,<value>]** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	**<pkt_timeout>** * **<retry_times>** (default 15s), determined by network

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<msgID>	Message identifier of packet. The range is 1-65535.
<topic>	The topic that the client wants to subscribe to or unsubscribe from.
<qos>	The QoS level at which the client wants to publish the messages. 0 At most once 1 At least once 2 Exactly once
<result>	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>	Timeout of the packet delivery. The range is 1-60. The default value is 5. Unit: second.
<retry_times>	Retry times when packet delivery times out. The range is 0-10. The default value is 3.
<err>	The error code of operation. For more details, please refer to ***Chapter 4***.

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

The 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.

AT+QMTUNS Unsubscribe from Topics	
Test Command **AT+QMTUNS=?**	Response **+QMTUNS: (**list of supported **<tcpconnectID>**s**),(**list of supported **<msgID>**s**),<topic>** **OK**
Write Command **AT+QMTUNS=<tcpconnectID>,<msgID>,<topic1>[,<topic2>…]**	Response **OK** **+QMTUNS: <tcpconnectID>,<msgID>,<result>** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	**<pkt_timeout>** * **<retry_times>** (default 15s), determined by network

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<msgID>	Message identifier of packet. The range is 1-65535.
<topic>	The topic that the client wants to subscribe to or unsubscribe from.
<result>	Result of the command execution. 0 Sent packet successfully and received ACK from server 1 Packet retransmission 2 Failed to send packet

<pkt_timeout>	Timeout of the packet delivery. The range is 1-60. The default value is 5. Unit: second.
<retry_times>	Retry times when packet delivery times out. The range is 0-10. The default value is 3.
<err>	The error code of operation. For more details, please refer to ***Chapter 4***.

3.2.8. AT+QMTPUB Publish Messages

The command is used to publish 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.

AT+QMTPUB Publish Messages	
Test Command **AT+QMTPUB=?**	Response **+QMTPUB: (**list of supported **<tcpconnectID>**s**),(**list of supported **<msgID>**s**),(**list of supported **<qos>**s**),(**list of supported **<retain>**s**),<topic>,[<msglen>]** **OK**
Write Command Publish variable-length messages **AT+QMTPUB=<tcpconnectID>,<msgID>,<qos>,<retain>,<topic>** After **>** is responded, input the data to be sent. Tap **CTRL+Z** to send, and tap **ESC** to cancel the operation.	Response **OK** **+QMTPUB: <tcpconnectID>,<msgID>,<result>[,<value>]** If there is an error related to ME functionality: **+CME ERROR: <err>**
Write Command Publish fixed-length messages **AT+QMTPUB=<tcpconnectID>,<msgID>,<qos>,<retain>,<topic>,<msglen>** After **>** is responded, input the data to be sent. The number of bytes of input data must equal to **<msglen>**.	Response **OK** **+QMTPUB: <tcpconnectID>,<msgID>,<result>[,<value>]** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	**<pkt_timeout>** * **<retry_times>** (default 15s), determined by network

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<msgID>	Message identifier of packet. The range is 0-65535. It will be 0 only when **<qos>**=0.

<qos>	The QoS level at which the client wants to publish the messages. 0 At most once 1 At least once 2 Exactly once
<retain>	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
<topic>	Topic that needs to be published.
<msglen>	Length of the message to be published. The range is 1-1548. Unit: byte.
<result>	Result of the command execution. 0 Packet sent successfully and ACK received from server (message that published when **<qos>**=0 does not require ACK) 1 Packet retransmission 2 Failed to send packet
<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>	Timeout of the packet delivery. The range is 1-60. The default value is 5. Unit: second.
<retry_times>	Retry times when packet delivery times out. The range is 0-10. The default value is 3.
<err>	The error code of operation. For more details, please refer to ***Chapter 4***.

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 1548 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: <tcpconnectID>,<msgID>,<topic>,<payload>
   For more details about the URC description, please refer to ***Chapter 5.2***.

3.2.9. AT+QMTPUBEX Publish Messages

The command is used to publish messages. It provides the same functions as **AT+QMTPUB**, except that the format is different.

AT+QMTPUBEX Publish Messages	
Test Command **AT+QMTPUBEX=?**	Response **+QMTPUBEX: (**list of supported **<tcpconnectID>**s**),(**list of supported **<msgID>**s**),(**list of supported **<qos>**s**),(**list of supported **<retain>**s**),<topic>,<msg>** **OK**
Write Command **AT+QMTPUBEX=<tcpconnectID>,<msgID>,<qos>,<retain>,<topic>,"<msg>**	Response **OK** **+QMTPUBEX: <tcpconnectID>,<msgID>,<result>[,<value>]** If there is an error related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	**<pkt_timeout>** * **<retry_times>** (default 15s), determined by network

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<msgID>	Message identifier of packet. The range is 0-65535. It will be 0 only when **<qos>**=0.
<qos>	The QoS level at which the client wants to publish the messages. 0 At most once 1 At least once 2 Exactly once
<retain>	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
<topic>	Topic that needs to be published.
<msg>	Message to be published.
<result>	Result of the command execution. 0 Packet sent successfully and ACK received from server (message that published when **<qos>**=0 does not require ACK) 1 Packet retransmission 2 Failed to send packet
<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>	Timeout of the packet delivery. The range is 1-60. The default value is 5. Unit:

	second.
<retry_times>	Integer type. Retry times when packet delivery times out. The range is 0-10. The default value is 3.
<err>	The error code of operation. For more details, please refer to ***Chapter 4***.

3.2.10. AT+QMTRECV Read Messages from Buffers

The command is used to read messages from storage buffer where the messages are stored when they are reported by the server.

AT+QMTRECV Read Messages from Buffers	
Test Command **AT+QMTRECV=?**	Response **OK**
Read Command **AT+QMTRECV?**	Response **+QMTRECV: <client_idx>,<store_status_0>,<store_status_1>,<store_status_2>,<store_status_3>,<store_status_4>** **OK** If there is no MQTT connection, response: **OK**
Write Command **AT+QMTRECV=<client_idx>[,<recv_id>]**	Response List of (**+QMTRECV: <client_idx>,<msg_id>,<topic>,[<payload_len>,]<payload>**)s **OK** If there is no message received, response: **OK** If there is no MQTT connection, response: **ERROR**
Maximum Response Time	

Parameter

<client_idx>	Integer type. MQTT client identifier. The range is 0-5.
<store_status>	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.
<recv_id>	Integer type. Indicate the serial number of every single message received. The range is 0-4.

<msg_id>	Integer type. Message identifier of packet. The range is 0-65535. It will be 0 only when **<qos>**=0.
<topic>	String type. Topic that needs to be published.
<payload_len>	Integer type. The length of payload.
<payload>	String type. The payload that relates to the topic name.

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 general error codes.

Table 1: Summary of Error Codes

<err>	Meaning
0	UE failure
1	No connection to UE
2	Adaptor link reserved
3	Operation not allowed
4	Operation not supported
5	PH-SIM PIN required
6	PH-FSIM PIN required
7	PH-FSIM PUK required
10	(U)SIM not inserted
11	(U)SIM PIN required
12	(U)SIM PUK required
13	(U)SIM failure
14	(U)SIM busy
15	(U)SIM wrong
16	Incorrect password
17	(U)SIM PIN2 required
18	(U)SIM PUK2 required

20	Memory full
21	Invalid index
22	Not found
23	Memory failure
24	Text string too long
25	Invalid characters in text string
26	Dial string too long
27	Invalid characters in dial string
30	No network service
31	Network timeout
32	Network not allowed - emergency calls only
40	Network personalization PIN required
41	Network personalization PUK required
42	Network subset personalization PIN required
43	Network subset personalization PUK required
44	Service provider personalization PIN required
45	Service provider personalization PUK required
46	Corporate personalization PIN required
47	Corporate personalization PUK required

5 MQTT Related URCs

This chapter gives MQTT related URCs and their descriptions.

Table 2: MQTT Related URCs

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

5.1. "+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	
+QMTSTAT: <tcpconnectID>,<err_code>	When the state of MQTT link layer is changed, the client will close the MQTT connection and report the URC.
Reference	

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<err_code>	An error code. Please refer to the table below for details.

Table 3: Error Codes of "+QMTSTAT" URC

<err_code>	Description	How to do
1	The connection is closed or reset by a peer.	Execute **AT+QMTOPEN** 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.	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 the 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 the server again.
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.	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.	

5.2. "+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	
+QMTRECV: <tcpconnectID>,<msgID>,"<topic>","<payload>"	Notify the host to read the received data that is reported from MQTT server.
Reference	

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<msgID>	The message identifier of packet.
<topic>	The topic that received from MQTT server.
<payload>	The payload that relates to the topic name.

6 Examples

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

6.1. Example of MQTT Operation without SSL

```
AT+QMTCFG="aliauth",0,"oyjtmPl5a5j","MQTT_TEST","wN9Y6pZSIly7Exa5qVzcmigEGO4kAazZ"
                                    //Configure Alibaba device information for AliCloud.
OK

AT+QMTOPEN=?
+QMTOPEN: <tcpconnectID>,<host_name>,<port>

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: <tcpconnectID>,<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: <tcpconnectID>,<msgID>,<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 : <tcpconnectID>,<msgID>,<qos>,<retain>,<topic>

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 1548 bytes and the data that 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.”

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

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

6.2. 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” (<seclevel>=2 in AT+QSSLCFG), upload 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, 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, 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 client.pem data, 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="seclevel",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

7 Appendix A References

Table 4: 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 5: 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