BG96 MQTT Application Note

LPWA Module Series

Version: 1.2

Date: 2021-06-01

Status: Released

General Notes

Disclaimer

Duty of Confidentiality

About the Document

Revision History

Version	Date	Author	Description
1.0	2018-01-04	Louis GU/ Serena SHEN	Initial
1.1	2019-05-20	Lane HAO/ Lucifer YAN	1. Added AT+QMTCFG="prefix". 2. Added AT+QMTCFG="recv/mode". 3. Updated the description of AT+QMTPUB. 4. Added AT+QMTPUBEX.
1.2	2021-06-01	Adonis CHEN	1. Chapter 3.1: Added the definitions of AT Command Syntax. 2. Chapter 3.2: Added the declaration of AT command examples. 3. Chapter 3.3.8: Updated the maximum length of messages to be published into 4096 bytes. 4. Chapter 3.3.10: Added **AT+QMTRECV**. 5. Chapter 4.2: Added URC **+QMTRECV: <client_idx>,<recvID>**. 6. Deleted the summary of error codes.

Contents

About the Document 3
Contents 4
Table Index 5

1 Introduction 6

2 MQTT Data Interaction 7

3 MQTT Related AT Commands 8
3.1. AT Command Syntax 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 for MQTT Client 15
3.3.4. AT+QMTCONN Connect a Client to MQTT Server 16
3.3.5. AT+QMTDISC Disconnect a Client from MQTT Server 17
3.3.6. AT+QMTSUB Subscribe to Topics 18
3.3.7. AT+QMTUNS Unsubscribe from Topics 19
3.3.8. AT+QMTPUB Publish Messages 20
3.3.9. AT+QMTPUBEX Publish Messages with Command Mode 22
3.3.10. AT+QMTRECV Read Messages from Buffers 23

4 MQTT Related URCs 25
4.1. +QMTSTAT: URC to Indicate State Change in MQTT Link Layer 25
4.2. +QMTRECV: URC to Notify Host to Read MQTT Packet Data 26

5 Examples 27
5.1. Example of MQTT Operation without SSL 27
5.2. Example of MQTT Operation with SSL 28

6 Appendix A Reference 31

Table Index

Table 1: Type of AT Commands and Responses 8
Table 2: Error Codes of the URC 25
Table 3: Related Documents 31
Table 4: Terms and Abbreviations 31

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 BG96 module through AT commands.

2 MQTT Data Interaction

This chapter describes the data interaction mechanism of MQTT function.

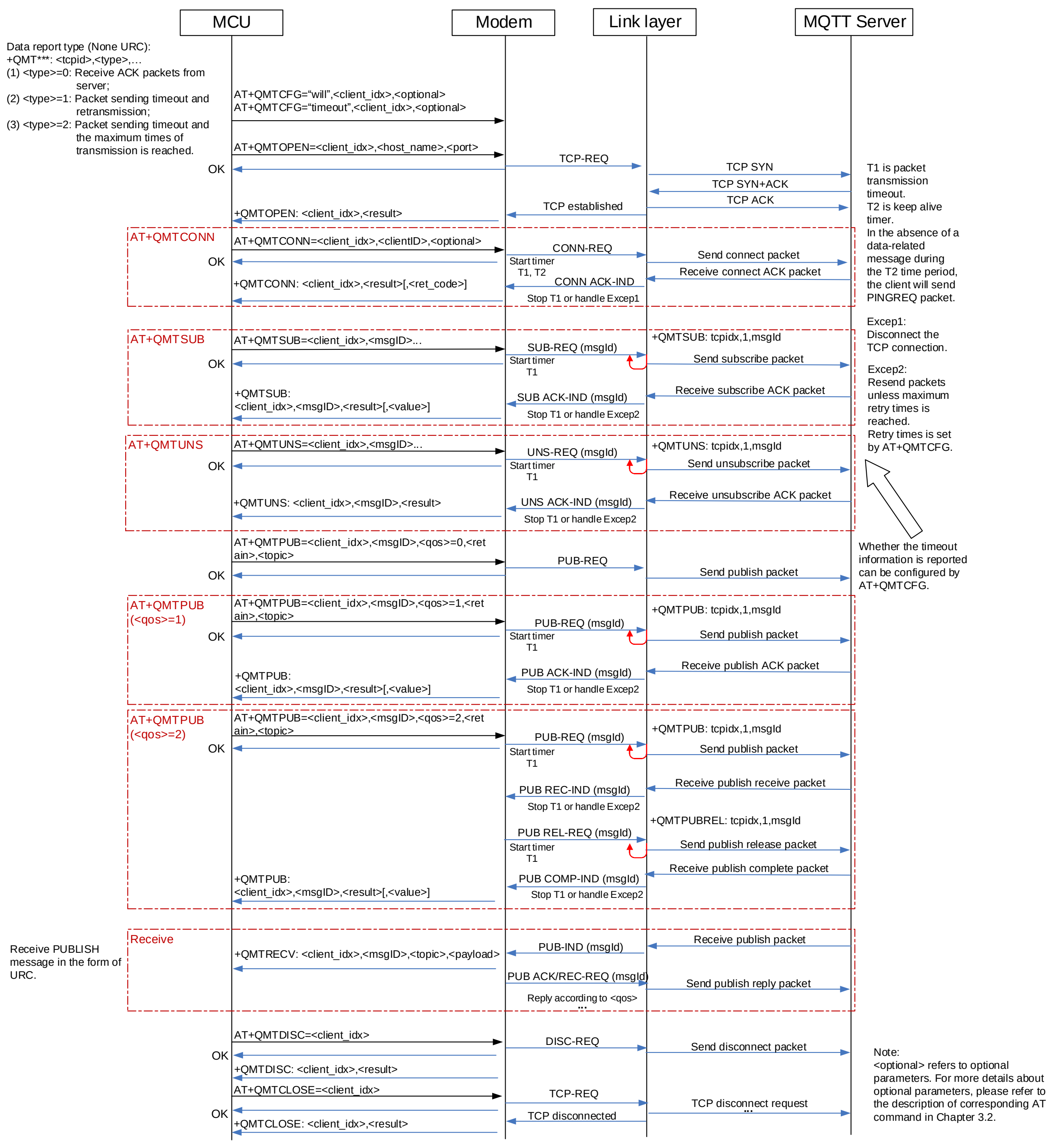
MCU
Modem
Link layer
MQTT Server
Data report type (None URC):
+QMT***: <tcpid>,<type>,…
(1) <type>=0: Receive ACK packets from server;
(2) <type>=1: Packet sending timeout and retransmission;
(3) <type>=2: Packet sending timeout and the maximum times of transmission is reached.
AT+QMTCFG="will",<client_idx>,<optional>
AT+QMTCFG="timeout",<client_idx>,<optional>
AT+QMTOPEN=<client_idx>,<host_name>,<port>
OK
TCP-REQ
TCP SYN
TCP SYN+ACK
TCP ACK
TCP established
+QMTOPEN: <client_idx>,<result>
T1 is packet transmission timeout.
T2 is keep alive timer.
In the absence of a data-related message during the T2 time period, the client will send PINGREQ packet.
AT+QMTCONN
AT+QMTCONN=<client_idx>,<clientID>,<optional>
OK
CONN-REQ
Start timer T1, T2
Send connect packet
Receive connect ACK packet
CONN ACK-IND
+QMTCONN: <client_idx>,<result>[,<ret_code>]
Stop T1 or handle Excep1
Excep1: Disconnect the TCP connection.
AT+QMTSUB
AT+QMTSUB=<client_idx>,<msgID>...
OK
SUB-REQ (msgId)
Start timer T1
+QMTSUB: tcpidx,1,msgId
Send subscribe packet
Excep2: Resend packets unless maximum retry times is reached.
Retry times is set by AT+QMTCFG.
+QMTSUB: <client_idx>,<msgID>,<result>[,<value>]
SUB ACK-IND (msgId)
Receive subscribe ACK packet
Stop T1 or handle Excep2
AT+QMTUNS
AT+QMTUNS=<client_idx>,<msgID>...
OK
UNS-REQ (msgId)
Start timer T1
+QMTUNS: tcpidx,1,msgId
Send unsubscribe packet
+QMTUNS: <client_idx>,<msgID>,<result>
UNS ACK-IND (msgId)
Receive unsubscribe ACK packet
Stop T1 or handle Excep2
Whether the timeout information is reported can be configured by AT+QMTCFG.
AT+QMTPUB=<client_idx>,<msgID>,<qos>=0,<retain>,<topic>
OK
PUB-REQ
Send publish packet
AT+QMTPUB (<qos>=1)
AT+QMTPUB=<client_idx>,<msgID>,<qos>=1,<retain>,<topic>
OK
PUB-REQ (msgId)
Start timer T1
+QMTPUB: tcpidx,1,msgId
Send publish packet
+QMTPUB: <client_idx>,<msgID>,<result>[,<value>]
PUB ACK-IND (msgId)
Receive publish ACK packet
Stop T1 or handle Excep2
AT+QMTPUB (<qos>=2)
AT+QMTPUB=<client_idx>,<msgID>,<qos>=2,<retain>,<topic>
OK
PUB-REQ (msgId)
Start timer T1
+QMTPUB: tcpidx,1,msgId
Send publish packet
PUB REC-IND (msgId)
Receive publish receive packet
Stop T1 or handle Excep2
+QMTPUBREL: tcpidx,1,msgId
PUB REL-REQ (msgId)
Start timer T1
Send publish release packet
Receive publish complete packet
+QMTPUB: <client_idx>,<msgID>,<result>[,<value>]
PUB COMP-IND (msgId)
Stop T1 or handle Excep2
Receive
Receive PUBLISH message in the form of URC.
+QMTRECV: <client_idx>,<msgID>,<topic>,<payload>
PUB-IND (msgId)
Receive publish packet
PUB ACK/REC-REQ (msgId)
Send publish reply packet
Reply according to <qos>
…
AT+QMTDISC=<client_idx>
OK
DISC-REQ
Send disconnect packet
+QMTDISC: <client_idx>,<result>
AT+QMTCLOSE=<client_idx>
OK
TCP-REQ
TCP disconnect request
…
TCP disconnected
+QMTCLOSE: <client_idx>,<result>
Note:
<optional> refers to optional parameters. For more details about optional parameters, please refer to the description of corresponding AT command in Chapter 3.2.

Figure 1: MQTT Data Interaction Mechanism

3 MQTT Related AT Commands

This chapter presents the AT commands for operating MQTT function.

3.1. AT Command Syntax

3.1.1. Definitions

● **<CR>**	Carriage return character.
● **<LF>**	Line feed character.
● **<...>**	Parameter name. Angle brackets do not appear on 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 omitted, the new value equals to the previous value or the default settings, unless otherwise specified.
● **<u>Underline</u>**	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>**. Throughout this document, only the commands and responses are presented, while carriage return and line feed characters are deliberately omitted.

Table 1: Type of AT Commands and Responses

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

Execution Command	**AT+<cmd>**	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 familiarize with AT commands and learn how to use them. 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 **AT+QMTCFG=?**	Response **+QMTCFG: "version",(**range of supported **<client_idx>**s**),(**list of supported **<vsn>**s**)** **+QMTCFG: "pdpcid",(**range of supported **<client_idx>**s**),(**range of supported **<cid>**s**)** **+QMTCFG: "ssl",(**range of supported **<client_idx>**s**),(**list of supported **<SSL_enable>**s**),(**range of supported **<ctxindex>**s**)** **+QMTCFG: "keepalive",(**range of supported **<client_idx>**s**),(**range of supported **<keep_alive_time>**s**)** **+QMTCFG: "session",(**range of supported **<client_idx>**s**),(**list of supported **<clean_session>**s**)** **+QMTCFG: "prefix",(**range of supported **<client_idx>**s**),"ipv6_prefix",(**list of supported **<prefix_length>**s**)** **+QMTCFG: "timeout",(**range of supported **<client_idx>**s**),(**range of supported **<pkt_timeout>**s**),(**range of supported **<retry_times>**s**),(**list of supported **<timeout_notice>**s**)** **+QMTCFG: "will",(**range of supported **<client_idx>**s**),(**list of supported**<will_fg>**s**),(**range of supported**<will_qos>**s**),(**list of supported **<will_retain>**s**),<will_topic>,<will_msg>** **+QMTCFG: "recv/mode",(**range of supported

	<client_idx>s**),(**list of supported **<msg_recv_mode>**s**),(**list of supported **<msg_len_enable>**s**)** **+QMTCFG: "aliauth",(**range of supported **<client_idx>**s**),<product_key>,<device_name>,<device_secret>** **OK**
Write Command **AT+QMTCFG="version",<client_idx>[,<vsn>]**	Response If the optional parameter is omitted, query the current MQTT protocol version: **+QMTCFG: "version",<vsn>** **OK** If the optional parameter is specified, set the MQTT protocol version: **OK** If there is any error: **ERROR**
Write Command **AT+QMTCFG="pdpcid",<client_idx>[,<cid>]**	Response If the optional parameter is omitted, query the PDP used by the MQTT client: **+QMTCFG: "pdpcid",<cid>** **OK** If the optional parameter is specified, set the PDP to be used by the MQTT client: **OK** If there is any error: **ERROR**
Write Command **AT+QMTCFG="ssl",<client_idx>[,<SSL_enable>[,<ctxindex>]]**	Response If the optional parameters are omitted, query the MQTT SSL mode and/or SSL context index: **+QMTCFG: "ssl",<SSL_enable>[,<ctxindex>]** **OK** If any of the optional parameters is specified, set the MQTT SSL mode and/or SSL context index: **OK**

	If there is any error: **ERROR**
Write Command **AT+QMTCFG="keepalive",<client_idx>[,<keep_alive_time>]**	Response If the optional parameter is omitted, query the keep-alive time: **+QMTCFG: "keepalive",<keep_alive_time>** **OK** If the optional parameter is specified, set the keep-alive time: **OK** If there is any error: **ERROR**
Write Command **AT+QMTCFG="session",<client_idx>[,<clean_session>]**	Response If the optional parameter is omitted, query the session type: **+QMTCFG: "session",<clean_session>** **OK** If the optional parameter is specified, set the session type: **OK** If there is any error: **ERROR**
Write Command **AT+QMTCFG="prefix",<client_idx>[,<ipv6_prefix>,<prefix_length>]**	Response If the optional parameters are omitted, query the prefix of IPv4-converted IPv6 address and the length of the prefix string: **+QMTCFG: "prefix",<ipv6_prefix>,<prefix_length>** **OK** If the optional parameters are specified, set the prefix of IPv4-converted IPv6 address and the length of the prefix string: **OK** If there is any error: **ERROR**
Write Command **AT+QMTCFG="timeout",<client_idx>[,<pkt_timeout>[,<retry_times>][,<timeout_notice>]]**	Response If the optional parameters are omitted, query the timeout of message delivery: **+QMTCFG: "timeout",<pkt_timeout>,<retry_times>,<timeout_notice>**

	OK If any of the optional parameters is specified, set the timeout of message delivery: **OK** If there is any error: **ERROR**
Write Command **AT+QMTCFG="will",<client_idx>[,<will_fg>[,<will_qos>,<will_retain>,<will_topic>,<will_msg>]]**	Response If the optional parameters are omitted, query the Will information: **+QMTCFG: "will",<will_fg>[,<will_qos>,<will_retain>,<will_topic>,<will_msg>]** **OK** If any of the optional parameters is specified, set the Will information: **OK** If there is any error: **ERROR**
Write Command **AT+QMTCFG="recv/mode",<client_idx>[,<msg_recv_mode>[,<msg_len_enable>]]**	Response If the optional parameters are omitted, query the MQTT message receiving mode: **+QMTCFG: "recv/mode",<msg_recv_mode>[,<msg_len_enable>]** **OK** If any of the optional parameters are specified, configure the MQTT message receiving mode: **OK** If there is any error: **ERROR**
Write Command **AT+QMTCFG="aliauth",<client_idx>[,<product_key>,<device_name>,<device_secret>]**	Response If the optional parameters are omitted, query the device information: **[+QMTCFG: "aliauth",<product_key>,<device_name>,<device_secret>]** **OK**

	If the optional parameters are specified, set the device information: **OK** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

<client_idx>	Integer type. MQTT socket identifier. Range: 0–5.
<vsn>	Integer type. MQTT protocol version. 3 MQTT v3.1 4 MQTT v3.1.1
<cid>	Integer type. The PDP to be used by the MQTT client. Range: 1–16. Default value: 1.
<will_fg>	Integer type. Whether to 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 in PUBLISH messages. 0 When a client sends a PUBLISH message to a server, the server does not retain the message after it has been delivered to the current subscribers 1 When a client sends a PUBLISH message to a server, the server retains the message after it has been delivered to the current subscribers
<will_topic>	String type. Will topic name Length range: 0–255 bytes.
<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. Length range: 0–255 bytes.
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 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 packet or not. 0 Not report timeout message when transmitting packet

	1 Report timeout message when transmitting packet
<clean_session>	Integer type. Configure the session type. 0 The server must store the subscriptions of the client after it disconnects 1 The server must discard any previously maintained information about the client after it disconnects and treat the connection as “clean”.
<ipv6_prefix>	String type. The prefix of IPv4-converted IPv6 address.
<prefix_length>	Integer type. The length of the prefix string. The value can be 32, 40, 48, 56, 64 or 96. Unit: byte.
<keep_alive_time>	Integer type. Keep-alive time. Range: 0–3600. Default value: 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. If the keep-alive time is 0, this mean the server is not required to disconnect the client on the grounds of inactivity.
<SSL_enable>	Integer type. MQTT SSL mode. 0 Use normal TCP connection for MQTT 1 Use SSL TCP secure connection for MQTT
<ctxindex>	Integer type. SSL context index. Range: 0–5. Valid only when **<SSL_enable>**=1.
<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. Whether the URC contains the length of MQTT message received from server. 0 The URC does not contain the length 1 The URC contains the length
<product_key>	String type. Product key obtained from AliCloud.
<device_name>	String type. Device name obtained from AliCloud.
<device_secret>	String type. Device secret key obtained from AliCloud.

NOTES

1. When **<will_fg>**=1, **<will_qos>**, **<will_retain>**, **<will_topic>** and **<will_msg>** must be specified. When **<will_fg>**=0, the four parameters should be omitted.
2. **<clean_session>**=0 is valid only when the server supports the operation.
3. If the MQTT connection is configured to SSL mode, **<ctxindex>** must exist. In addition, **AT+QSSLCFG** is needed to configure the SSL version, cipher suite, secure level, CA certificate, client certificate, client key and ignorance of RTC time, which is used in MQTT SSL handshake procedure.
4. Care must be taken to ensure message delivery does not time out while it is still being sent.
5. **AT+QMTCFG="aliauth"** is only used for AliCloud. If it is configured, **<username>** and **<password>** in **AT+QMTCONN** can be omitted.

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

This command opens a network connection for MQTT client.

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

Parameter

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

3.3.3. AT+QMTCLOSE Close a Network for MQTT Client

This command closes a network for MQTT client.

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

Parameter

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

3.3.4. AT+QMTCONN Connect a Client to MQTT Server

This command requests a connection to MQTT server for the client. When a TCP/IP socket connection is established from a client to a server, an MQTT 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 **<client_idx>**s**)**,**<clientID>**,**<username>**,**<password>** **OK**
Read Command **AT+QMTCONN?**	Response **[+QMTCONN: <client_idx>,<state>]** **OK**
Write Command **AT+QMTCONN=<client_idx>,<clientID>[,<username>[,<password>]]**	Response **OK** **+QMTCONN: <client_idx>,<result>[,<ret_code>]**

	If there is any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** (default 5 s), determined by the network
Characteristics	-

Parameter

<client_idx>	Integer type. MQTT client 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 Packet sent successfully and ACK received from server 1 Packet retransmission 2 Failed to send packet
<state>	Integer type. MQTT connection state. 1 MQTT is initializing 2 MQTT is connecting 3 MQTT is connected 4 MQTT is disconnecting
<ret_code>	Integer type. Returned code of the connection status. 0 Connection Accepted 1 Connection Refused: Unacceptable Protocol Version 2 Connection Refused: Identifier Rejected 3 Connection Refused: Server Unavailable 4 Connection Refused: Bad User Name or Password 5 Connection Refused: Not Authorized
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.

NOTE

If a new client with the same Client ID is connected to the server, the "older" client must be disconnected by the server before completing the CONNECT flow of the new client.

3.3.5. AT+QMTDISC Disconnect a Client from MQTT Server

This command requests a disconnection from MQTT server for the 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 **AT+QMTDISC=?**	Response **+QMTDISC: (**range of supported **<client_idx>**s**)** **OK**
Write Command **AT+QMTDISC=<client_idx>**	Response **OK** **+QMTDISC: <client_idx>,<result>** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	-

Parameter

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

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 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 **<client_idx>**s**),(**range of supported **<msgID>**s**),<topic>,(**range of supported **<qos>**s**)** **OK**
Write Command **AT+QMTSUB=<client_idx>,<msgID>,<topic1>,<qos1>[,<topic2>,<qos2>…]**	Response **OK** **+QMTSUB: <client_idx>,<msgID>,<result>[,<value>]** If there is any error: **ERROR**

Maximum Response Time	**<pkt_timeout>** × **<retry_times>** (default 15 s), determined by the network
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

<client_idx>	Integer type. MQTT client identifier. Range: 0–5.
<msgID>	Integer type. Message identifier of packet. Range: 1–65535.
<topic>	String type. 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 the packet
<value>	Integer type. If **<result>** = 0, it is a vector of granted QoS levels. If **<result>** = 1, it means the times of packet retransmission. If **<result>** = 2, it will not be presented.
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.

NOTE

The **<msgID>** is only present in messages where the QoS bits in the fixed header indicate QoS level 1 or 2. It must be unique among 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.

3.3.7. 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 named topics.

AT+QMTUNS Unsubscribe from Topics	
Test Command **AT+QMTUNS=?**	Response **+QMTUNS: (**range of supported **<client_idx>**s**),(**range of supported **<msgID>**s),**<topic>**

	OK
Write Command **AT+QMTUNS=<client_idx>,<msgID>,<topic1>[,<topic2>…]**	Response **OK** **+QMTUNS: <client_idx>,<msgID>,<result>** If there is any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** × **<retry_times>** (default 15 s), determined by the network
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

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

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 topic name. If a client subscribes to one or more topics, any messages published to those topics are sent by the server to the client as PUBLISH messages.

AT+QMTPUB Publish Messages	
Test Command **AT+QMTPUB=?**	Response **+QMTPUB: (**range of supported **<client_idx>**s**),(**range of supported **<msgID>**s**),(**range of supported **<qos>**s**),(**list of supported **<retain>**s**),<topic>,[**range of supported **<msglen>**s**]** **OK**

Write Command **AT+QMTPUB=<client_idx>,<msgID>,<qos>,<retain>,<topic>**	Response **>** Input the data to be sent after **>** is responded. Tap **Ctrl+Z** to send the data, and tap **Esc** to cancel the operation. **OK** **+QMTPUB: <client_idx>,<msgID>,<result>[,<value>]** If there is any error: **ERROR**
Write Command **AT+QMTPUB=<client_idx>,<msgID>,<qos>,<retain>,<topic>,<msglen>**	Response **>** Input the data to be sent after **>** is responded. The number of bytes of input data must equal to **<msglen>**. **OK** **+QMTPUB: <client_idx>,<msgID>,<result>[,<value>]** If there is any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** × **<retry_times>** (default 15 s), determined by the network
Characteristics	-

Parameter

<client_idx>	Integer type. MQTT client identifier. Range: 0–5.
<msgID>	Integer type. Message identifier of packet. Range: 0–65535. The value is 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 or not the server retains the message after it has been delivered to the current subscribers. 0 The server does not retain the message after it has been delivered to the current subscribers 1 The server retains the message after it has been delivered to the current subscribers
<topic>	String type. The topic that needs to be published.
<msglen>	Integer type. Length of message to be published. Range: 1–4096. Unit: byte.
<result>	Integer type. 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>	Integer type. If **<result>** = 1, it is the times of packet retransmission. If **<result>** = 0 or 2, it will not be presented.
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.

NOTES

1. The client should only continue to publish a new packet after this command is executed successfully and responded with **OK**. The maximum quantity of packet to be transmitted should not be greater than that of inflight windows (5).
2. After this command is executed, the client is ready to send data as payload. The maximum length of the input data is 4096 bytes at a time and the data will be sent by tapping **Ctrl+Z**.
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 is returned to notify the host to read the received data that is reported from MQTT server: **+QMTRECV: <client_idx>,<msgID>,<topic>,<payload>**. For more details about the URC, refer to ***4.2***.

3.3.9. AT+QMTPUBEX Publish Messages with Command Mode

This command publishes messages. Unlike **AT+QMTPUB**, customers can enter the specified payload directly by executing this command.

AT+QMTPUBEX Publish Messages	
Test Command **AT+QMTPUBEX=?**	Response **+QMTPUBEX: (**range of supported **<client_idx>**s**),(**range of supported **<msgID>**s**),(**range of supported **<qos>**s**),(**list of supported **<retain>**s**),<topic>,<msg>** **OK**
Write Command **AT+QMTPUBEX=<client_idx>,<msgID>,<qos>,<retain>,<topic>,<msg>**	Response **OK** **+QMTPUB: <client_idx>,<msgID>,<result>[,<value>]** If there is any error: **ERROR**

Maximum Response Time	**<pkt_timeout>** × **<retry_times>** (default 15 s), determined by the network
Characteristics	-

Parameter

<client_idx>	Integer type. MQTT client identifier. Range: 0–5.
<msgID>	Integer type. Message identifier of packet. Range: 0–65535. The value is 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 or not the server retains the message after it has been delivered to the current subscribers. 0 The server does not retain the message after it has been delivered to the current subscribers 1 The server retains the message after it has been delivered to the current subscribers
<topic>	String type. Topic that needs to be published.
<msg>	String type. Message to be published. Maximum message length: 560 bytes.
<result>	Integer type. 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>	Integer type. If **<result>** = 1, it is the times of packet retransmission. If **<result>** = 0 or 2, it will not be presented.
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: s.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.

3.3.10. AT+QMTRECV Read Messages from Buffers

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

AT+QMTRECV Read Messages from Buffers	
Test Command	Response

AT+QMTRECV=?	**OK**
Read Command **AT+QMTRECV?**	Response **+QMTRECV: <client_idx>,<store_status0>,<store_status1>,<store_status2>,<store_status3>,<store_status4>** **OK** If there is no MQTT connection: **OK**
Write Command **AT+QMTRECV=<client_idx>[,<recvID>]**	Response List of (**+QMTRECV: <client_idx>,<msgID>,<topic>,[<payload_len>,]<payload>**)s **OK** If there is no message received: **OK** If there is no MQTT connection: **ERROR**
Maximum Response Time	300 ms
Characteristics	-

Parameter

<client_idx>	Integer type. MQTT client identifier. Range: 0–5.
<store_statusX>	Integer type. Whether there is a message stored in the buffer. The maximum quantity of message that can be stored in the buffer is 5. Therefore, URC reports maximally 5 messages simultaneously. 0 No message in the buffer 1 The buffer stores one or more messages
<recvID>	Integer type. Serial number of every single message received. Range: 0–4. All buffer data will be read if this parameter is not specified.
<msgID>	Integer type. Message identifier of packet. Range: 0–65535. The value is 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 MQTT Related URCs

This chapter describes MQTT related URCs.

4.1. +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 MQTT link layer.

+QMTSTAT: URC to Indicate State Change in MQTT Link Layer	
+QMTSTAT: <client_idx>,<err_code>	Reported when the state of MQTT link layer is changed and the client closes the MQTT connection.

Parameter

<client_idx>	Integer type. MQTT socket identifier. Range: 0–5.
<err_code>	Integer type. Error code. Refer to the table below for details.

Table 2: Error Codes of the URC

<err_code>	**Description**	**How to do**
1	Connection is closed or reset by peer.	Execute **AT+QMTOPEN** to 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 input user name and password are correct. 2. Make sure that 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 input user name and password are correct. 2. Make sure that the client ID is not used. 3. Reopen MQTT connection and try to send CONNECT packet to server again.
5	The client sends DISCONNECT	This is a normal process.

	packet to sever and the server is initiative to close MQTT connection.	
6	The client closes MQTT connection due to packet sending failure all the time.	1. Make sure that the data is correct. 2. Try to reopen MQTT connection since there may be network congestion or an error.
7	The link is not valid or the server is unavailable.	Make sure that the link is valid or the server is available currently.
8-255	Reserved for future use.	-

4.2. +QMTRECV: URC to Notify Host to Read MQTT Packet Data

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

+QMTRECV: URC to Notify Host to Read MQTT Packet Data	
+QMTRECV: <client_idx>,<msgID>,<topic>,<payload>	Notify the host to read the received data that is reported from MQTT server.
+QMTRECV: <client_idx>,<recv_ID>	Notify that when the message received from MQTT server has been stored in buffer.

Parameter

<client_idx>	Integer type. MQTT socket identifier. Range: 0–5.
<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.
<recvID>	Integer type. The serial number of every single message received. Range: 0–4.

5 Examples

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

5.1. Example of MQTT Operation without SSL

```
AT+QMTCFG="aliauth",0,"oyjtmPl5a5j","MQTT_TEST","wN9Y6pZSIly7Exa5qVzcmigEGO4kAazZ"
                                            //Configure Alibaba device information for AliCloud.
OK
AT+QMTOPEN=?
+QMTOPEN: (0-5),<host_name>,(0-65535)

OK
AT+QMTOPEN=0,"iot-as-mqtt.cn-shanghai.aliyuncs.com",1883   //Open a network for MQTT client.
OK

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

OK
AT+QMTCONN=?
+QMTCONN: (0-5),<clientID>,<username>,<password>

OK
//Connect a client to MQTT server.
//If AliCloud is connected, AT+QMTCFG="aliauth" can be used to configure the device information in
advance, and <username>/<password> can be omitted.
AT+QMTCONN=0,"clientExample"
OK

+QMTCONN: 0,0,0                             //The client connected to MQTT server successfully.
AT+QMTSUB=?
+QMTSUB: (0-5),(1-65535),<topic>,(0-2)

OK
AT+QMTSUB=0,1,"topic/example",2             //Subscribe to topics.
```

```
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 that is published to the server by other devices, the module will report the
following information.
+QMTRECV: 0,0,"topic/example","This is the payload related to topic"
AT+QMTUNS=0,2,"topic/example"          //Unsubscribe from topics.
OK

+QMTUNS: 0,2,0
AT+QMTPUB=?
+QMTPUB : (0-5),(0-65535),(0-2),(0,1),<topic>,[(1-4096)]

OK
AT+QMTPUB=0,0,0,0,"topic/pub"          //Publish messages.
>This is test data, hello MQTT.        //After receiving >, input data This is test data, hello MQTT.
                                       And then send it. The maximum length of the data is 4096
                                       bytes and the data that beyond 4096 bytes will be omitted.
                                       After inputting data, tap Ctrl+Z to send.
OK

+QMTPUB: 0,0,0

//If a client subscribes to a topic named "topic/pub" and other devices publish the same topic to the server,
the module reports the following information.
+QMTRECV: 0,0,"topic/pub","This is test data, hello MQTT."
AT+QMTDISC=0                           //Disconnect a client from MQTT server.
OK

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

5.2. Example of MQTT Operation with SSL

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

//If SSL authentication mode is “server authentication”, store CA certificate to UFS.
```

```
AT+QFUPL="cacert.pem",1758,100
CONNECT
<Input the cacert.pem data, the size is 1758 bytes>
+QFUPL: 1758,384a

OK

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

OK

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

OK

AT+QSSLCFG="cacert",2,"cacert.pem"            //Configure CA certificate.
OK
AT+QSSLCFG="clientcert",2,"client.pem"        //Configure CC certificate.
OK
AT+QSSLCFG="clientkey",2,"user_key.pem"    //Configure CK certificate.
OK

//Configure SSL parameters.
AT+QSSLCFG="seclevel",2,2                     //SSL authentication mode: server authentication
OK
AT+QSSLCFG="sslversion",2,4                   //SSL authentication version
OK
AT+QSSLCFG="ciphersuite",2,0XFFFF             //Cipher suite
OK
AT+QSSLCFG="ignorelocaltime",1                    //Ignore the time of authentication.
OK
AT+QMTOPEN=0,"a1zgnxur10j8ux.iot.us-east-1.amazonaws.com",8883      //Start a MQTT SSL
                                                                     connection
OK

+QMTOPEN: 0,0
```

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

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

+QMTSUB: 0,1,0,1
AT+QMTPUB=0,1,1,0,"$aws/things/MQTT-1/shadow/update/accepted"    //Publish messages.
>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"
AT+QMTDISC=0                                  //Disconnect a client from MQTT server.
OK

+QMTDISC: 0,0
```

6 Appendix A Reference

Table 3: Related Documents

SN	Document Name	Remarks
[1]	MQTT V3.1 Protocol Specification	MQTT protocol specification version 3.1
[2]	MQTT V3.1.1 Protocol Specification	MQTT protocol specification version 3.1.1
[3]	Quectel_BG96_SSL_Application_Note	SSL Application Note for BG96 module

Table 4: 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
PDP	Packet Data Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
RTC	Real-Time Clock
UFS	User File System