BC68-GV&BC95-GV

MQTT Application Note

NB-IoT Module Series

Version: 1.0

Date: 2022-05-26

Status: Released

At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local offices. For more information, please visit:
http://www.quectel.com/support/sales.htm.

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm.
Or email us at: support@quectel.com.

Legal Notices

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

Use and Disclosure Restrictions

License Agreements

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

Copyright

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

Trademarks

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

Third-Party Rights

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

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

Privacy Policy

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

Disclaimer

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

About the Document

Revision History

Version	Date	Author	Description
-	2021-12-03	Benjamin LU	Creation of the document
1.0	2022-05-26	Benjamin LU	First official release

Contents

Table Index

1 Introduction

Quectel BC68-GV and BC95-GV modules support MQTT. MQTT is a broker-based publish/subscribe messaging protocol designed to be open, simple, lightweight and easy to implement. The biggest advantage of MQTT is that it provides real-time and reliable messaging services for remote device connections using minimum codes and limited bandwidth.

This document introduces how to use the MQTT function of Quectel BC68-GV and BC95-GV modules through AT commands.

2 MQTT Data Interaction

This chapter introduces the data interaction mechanism of MQTT function.

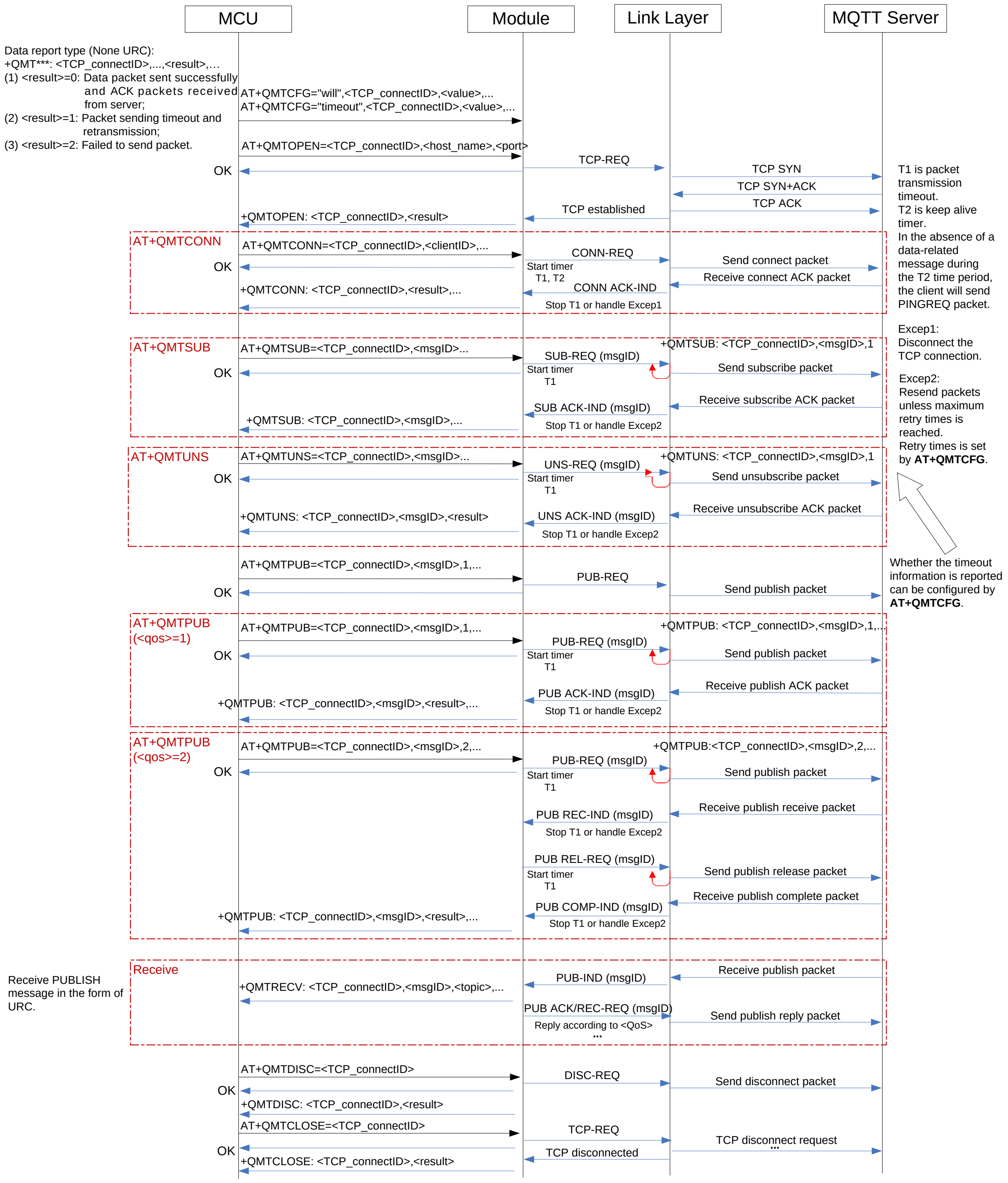
MCU
Module
Link Layer
MQTT Server
Data report type (None URC):
+QMT***: <TCP_connectID>,...,<result>,...
(1) <result>=0: Data packet sent successfully and ACK packets received from server;
(2) <result>=1: Packet sending timeout and retransmission;
(3) <result>=2: Failed to send packet.
AT+QMTCFG="will",<TCP_connectID>,<value>,...
AT+QMTCFG="timeout",<TCP_connectID>,<value>,...
AT+QMTOPEN=<TCP_connectID>,<host_name>,<port>
OK
TCP-REQ
TCP SYN
TCP SYN+ACK
TCP ACK
TCP established
+QMTOPEN: <TCP_connectID>,<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=<TCP_connectID>,<clientID>,...
OK
CONN-REQ
Start timer T1, T2
Send connect packet
Receive connect ACK packet
+QMTCONN: <TCP_connectID>,<result>,...
CONN ACK-IND
Stop T1 or handle Excep1
Excep1: Disconnect the TCP connection.
AT+QMTSUB
AT+QMTSUB=<TCP_connectID>,<msgID>...
OK
SUB-REQ (msgID)
Start timer T1
+QMTSUB: <TCP_connectID>,<msgID>,1
Send subscribe packet
Receive subscribe ACK packet
SUB ACK-IND (msgID)
Stop T1 or handle Excep2
+QMTSUB: <TCP_connectID>,<msgID>,...
Excep2: Resend packets unless maximum retry times is reached.
Retry times is set by AT+QMTCFG.
AT+QMTUNS
AT+QMTUNS=<TCP_connectID>,<msgID>...
OK
UNS-REQ (msgID)
Start timer T1
+QMTUNS: <TCP_connectID>,<msgID>,1
Send unsubscribe packet
Receive unsubscribe ACK packet
+QMTUNS: <TCP_connectID>,<msgID>,<result>
UNS ACK-IND (msgID)
Stop T1 or handle Excep2
AT+QMTPUB=<TCP_connectID>,<msgID>,1,...
OK
PUB-REQ
Send publish packet
Whether the timeout information is reported can be configured by AT+QMTCFG.
AT+QMTPUB (<qos>=1)
AT+QMTPUB=<TCP_connectID>,<msgID>,1,...
OK
PUB-REQ (msgID)
Start timer T1
+QMTPUB: <TCP_connectID>,<msgID>,1,...
Send publish packet
Receive publish ACK packet
PUB ACK-IND (msgID)
Stop T1 or handle Excep2
+QMTPUB: <TCP_connectID>,<msgID>,<result>,...
AT+QMTPUB (<qos>=2)
AT+QMTPUB=<TCP_connectID>,<msgID>,2,...
OK
PUB-REQ (msgID)
Start timer T1
+QMTPUB:<TCP_connectID>,<msgID>,2,...
Send publish packet
Receive publish receive packet
PUB REC-IND (msgID)
Stop T1 or handle Excep2
PUB REL-REQ (msgID)
Start timer T1
Send publish release packet
Receive publish complete packet
PUB COMP-IND (msgID)
Stop T1 or handle Excep2
+QMTPUB: <TCP_connectID>,<msgID>,<result>,...
Receive
Receive PUBLISH message in the form of URC.
+QMTRECV: <TCP_connectID>,<msgID>,<topic>,...
PUB-IND (msgID)
Receive publish packet
PUB ACK/REC-REQ (msgID)
Reply according to <QoS>
...
Send publish reply packet
AT+QMTDISC=<TCP_connectID>
OK
DISC-REQ
Send disconnect packet
+QMTDISC: <TCP_connectID>,<result>
AT+QMTCLOSE=<TCP_connectID>
OK
TCP-REQ
TCP disconnect request
...
TCP disconnected
+QMTCLOSE: <TCP_connectID>,<result>

Figure 1: MQTT Data Interaction Diagram

3 MQTT Related AT Commands

This chapter introduces the MQTT related AT commands.

3.1. AT Command Introduction

3.1.1. Definition

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

3.1.2. AT Command Syntax

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

Table 1: Types of AT Commands

Command Type	Syntax	Description
Test Command	**AT+<cmd>=?**	Test the existence of the corresponding command and return information about the type, value, or range of its parameter.
Read Command	**AT+<cmd>?**	Check the current parameter value of the corresponding 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 learn about the use of the AT commands introduced herein. The examples, however, should not be taken as Quectel's recommendations or suggestions about how to design a program flow or what status to set the module into. Sometimes multiple examples may be provided for one AT command. However, this does not mean that there is a correlation among these examples, or that they should be executed in a given sequence.

3.3. Description of MQTT 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",(**list of supported **<TCP_connectID>**s**),(**list of supported **<vsn>**s**)** **+QMTCFG: "keepalive",(**list of supported **<TCP_connectID>**s**),(**range of supported **<keep-alive_time>**s**)** **+QMTCFG: "session",(**list of supported **<TCP_connectID>**s**),(**list of supported **<clean_session>**s**)** **+QMTCFG: "timeout",(**list of supported **<TCP_connectID>**s**),(**range of supported **<pkt_timeout>**s**),(**range of supported **<retry_times>**s**),(**list of supported **<timeout_notice>**s**)** **+QMTCFG: "will",(**list of supported **<TCP_connectID>**s**),(**list of supported **<will_fg>**s**),(**range of supported **<will_QoS>**s**),(**list of supported **<will_retain>**s**),<will_topic>,<will_msg>** **+QMTCFG: "aliauth",(**list of supported **<TCP_connectID>**s**),<product_key>,<device_name>,<device_secret>** **OK**
Write Command Configure/query the MQTT protocol version **AT+QMTCFG="version",<TCP_connectI**	Response If the optional parameter is omitted, query the MQTT protocol version:

D>[,<vsn>]	**+QMTCFG: "version",<vsn>** **OK** If the optional parameter is specified, configure the MQTT protocol version: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure/query the keep-alive time **AT+QMTCFG="keepalive",<TCP_connectID>[,<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, configure the keep-alive time: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure/query the session type **AT+QMTCFG="session",<TCP_connectID>[,<clean_session>]**	Response If the optional parameter is omitted, query the session type: **+QMTCFG: "session",<clean_session>** **OK** If the optional parameter is specified, configure the session type: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure/query timeout of message delivery	Response If the optional parameters are omitted, query the timeout

AT+QMTCFG="timeout",<TCP_connectID>[,<pkt_timeout>,<retry_times>,<timeout_notice>]	of message delivery: **+QMTCFG: "timeout",<pkt_timeout>,<retry_times>,<timeout_notice>** **OK** If the optional parameters are specified, configure timeout of message delivery: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure/query Will information **AT+QMTCFG="will",<TCP_connectID>[,<will_fg>[,<will_QoS>,<will_retain>,<will_topic>,<will_msg>]]**	Response If the optional parameters are omitted, query the current configuration: **+QMTCFG: "will",<will_fg>[,<will_QoS>,<will_retain>,<will_topic>,<will_msg>]** **OK** If the optional parameters are specified, configure the Will information: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure/query Alibaba device information for Alibaba Cloud **AT+QMTCFG="aliauth",<TCP_connectID>[,<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, configure Alibaba device information for Alibaba Could: **OK** If there is any error: **ERROR**

	Or **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<TCP_connectID> Integer type. MQTT socket identifier. Currently only one MQTT instance is supported, the value is always 0.

<vsn> Integer type. MQTT protocol version.
3 MQTT protocol v3.1
4 MQTT protocol v3.1.1

<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 for PUBLISH messages.
0 When a client sends a PUBLISH message to a server, the server will 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 should retain the message after it has been delivered to the current subscribers.

<will_topic> String type. Name of Will topic. The maximum length is 255 bytes.

<will_msg> String type. The Will message defines the message content that is published to the Will topic when the client is unexpectedly disconnected. The maximum length is 255 bytes.

<pkt_timeout> Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 10. 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 packets.
0 Do 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 retained information about the client and treat the connection as "clean".

<keep-alive_time> Integer type. Keep-alive time. Range: 0–3600. Default value: 120. Unit: s. It defines the maximum time interval for receiving messages from a client. If the server does not receive a message from the client within the time that is 1.5 times

	as much as the keep-alive time, it disconnects the client as if the client has sent a DISCONNECT message. If this parameter is set to 0, the client is not disconnected.
<product_key>	String type. Product key issued by Alibaba Cloud.
<device_name>	String type. Device name issued by Alibaba Cloud.
<device_secret>	String type. Device secret key issued by Alibaba Cloud.
<err>	Integer type. Error codes. See ***Chapter 5*** for details.

NOTE

1. If **<will_fg>**=1, then **<will_QoS>**, **<will_retain>**, **<will_topic>** and **<will_msg>** must be configured. Otherwise, they are omitted.
2. **<clean_session>**=0 is valid only when the server supports the operation of "clean" session.
3. It is crucial to ensure message delivery does not time out while the message is still being sent.
4. **AT+QMTCFG="aliauth"** is only used for Alibaba Cloud. If **<product_key>**, **<device_name>** and **<device_secret>** are configured, **<username>** and **<password>** in **AT+QMTCONN** can be omitted.

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

This command opens the network connection for MQTT client.

AT+QMTOPEN Open the Network Connection for MQTT Client	
Test Command **AT+QMTOPEN=?**	Response **+QMTOPEN: (**list 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 any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately.

	The configurations are not saved.

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Currently only one MQTT instance is supported, the value is always 0.
<host_name>	String type. The address of the server. It could be an IP address or a domain name. The maximum length is 100 bytes.
<port>	Integer type. The port of the server. Range: 0–65535.
<result>	Integer type. Results of the command execution. -1 Failed opening network connection 0 Successfully opening network connection 1 Wrong parameter 2 MQTT identifier occupied 3 PDP activation failed. 4 Domain name resolution failed 5 Network connection lost
<err>	Integer type. Error codes. See ***Chapter 5*** for details.

3.3.3. AT+QMTCLOSE Close the Network Connection for MQTT Client

This command closes the network connection for MQTT client.

AT+QMTCLOSE Close the Network Connection for MQTT Client	
Test Command **AT+QMTCLOSE=?**	Response **+QMTCLOSE: (**list of supported **<TCP_connectID>**s**)** **OK**
Write Command **AT+QMTCLOSE=<TCP_connectID>**	Response **OK** **+QMTCLOSE: <TCP_connectID>,<result>** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Currently only one MQTT instance is supported, the value is always 0.
<result>	Integer type. Results of the command execution. -1 Failed closing network connection 0 Successfully closing network connection
<err>	Integer type. Error codes. See ***Chapter 5*** for details.

3.3.4. AT+QMTCONN Connect a Client to MQTT Server

This command connects a client to MQTT server. When a TCP/IP socket connection is established between a client and a server, a protocol layer 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 **<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 any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	**<pkt_timeout>** (default 10 s), determined by network
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Currently only one MQTT instance is supported, the value is always 0.
<clientID>	String type. Client identifier. The maximum length is 128 bytes.

<username>	String type. Username of the client. It can be used for authentication. The maximum length is 64 bytes.
<password>	String type. Password to the username of the client. It can be used for authentication. The maximum length is 128 bytes.
<result>	Integer type. Results of the command execution. 0 Data packet sent successfully and ACK received from server 1 Data packet retransmitted 2 Failed sending data packet
<state>	Integer type. MQTT connection states. 1 Initializing 2 Connecting 3 Connected 4 Disconnecting
<retcode>	Integer type. Return codes of connection status. 0 Connection accepted 1 Connection refused: unacceptable protocol version 2 Connection refused: identifier rejected 3 Connection refused: server unavailable 4 Connection refused: username or password wrong 5 Connection refused: unauthorized
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 10. Unit: s.
<err>	Integer type. Error codes. See ***Chapter 5*** for details.

NOTE

If the new client has the same client ID as the old one that has been connected to the server, then the old one will be disconnected by the server before the CONNECT flow of the new client is finished.

3.3.5. AT+QMTDISC Disconnect a Client from MQTT Server

This command disconnects a client 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 **<TCP_connectID>**s**)** **OK**
Write Command **AT+QMTDISC=<TCP_connectID>**	Response **OK** **+QMTDISC: <TCP_connectID>,<result>**

	If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Currently only one MQTT instance is supported, the value is always 0.
<result>	Integer type. Results of the command execution. -1 Failed closing connection 0 Successfully closing connection
<err>	Integer type. Error codes. See ***Chapter 5*** for details.

3.3.6. AT+QMTSUB Subscribe to Topics

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

AT+QMTSUB Subscribe to Topics	
Test Command **AT+QMTSUB=?**	Response **+QMTSUB: (**list of supported **<TCP_connectID>**s**),(**range of supported **<msgID>**s**),<topic>,(**range of supported **<QoS>**s**)[,<topic>,(**range of supported **<QoS>**s**)...]** **OK**
Write Command **AT+QMTSUB=<TCP_connectID>,<msgID>,<topic>,<QoS>[,<topic>,<QoS>...]**	Response **OK** **+QMTSUB: <TCP_connectID>,<msgID>,<result>[,<value>…]** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	**<pkt_timeout>** × **<retry_times>** (default 15 s), determined by network

Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<TCP_connectID> Integer type. MQTT socket identifier. Currently only one MQTT instance is supported, the value is always 0.

<msgID> Integer type. Message identifier. Range: 1–65535.

<topic> String type. Topic that the client subscribes to or unsubscribes from. The maximum length: 255 bytes.

<QoS> Integer type. The QoS level at which the client publishes the messages.
0 At most once
1 At least once
2 Exactly once

<result> Integer type. Results of the command execution.
0 Data packet sent successfully and ACK received from server
1 Data packet retransmitted
2 Failed sending data packet

<value> Integer type.
If **<result>** is 0, **<value>** is a granted QoS level. At this time, **<value>**=128 means that the server refuses to subscribe.
If **<result>** is 1, it means the times of packet retransmission.
If **<result>** is 2, it is not presented.

<pkt_timeout> Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 10. Unit: s.

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

<err> Integer type. Error codes. See ***Chapter 5*** for details.

NOTE

The **<msgID>** is only valid in messages with the value of **<QoS>** being 1 or 2. It must be unique amongst the set of "inflight" messages in the particular direction of communication. Generally, the value of **<msgID>** increases one by one according to the message quantity.

3.3.7. AT+QMTUNS Unsubscribe from Topics

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

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

	supported **<msgID>**s),**<topic>[,<topic>...]** **OK**
Write Command **AT+QMTUNS=<TCP_connectID>,<msgID>,<topic>[,<topic>…]**	Response **OK** **+QMTUNS: <TCP_connectID>,<msgID>,<result>** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	**<pkt_timeout>** × **<retry_times>** (default 15 s), determined by network
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Currently only one MQTT instance is supported, the value is always 0.
<msgID>	Integer type. Message identifier. Range: 1–65535.
<topic>	String type. Topic that the client subscribes to or unsubscribes from. The maximum length: 255 bytes.
<result>	Integer type. Results of the command execution. 0 Data packet sent successfully and ACK received from server 1 Data packet retransmitted 2 Failed sending data packet
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 10. Unit: s.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.
<err>	Integer type. Error codes. See ***Chapter 5*** for details.

3.3.8. AT+QMTPUB Publish Messages

This command publishes messages from a client to a server that are then distributed to those subscribers who have already subscribed to topics. Each PUBLISH message is associated with a topic name. If a client subscribes to one or more topics, any message published to those topics is sent by the server to the client as a PUBLISH message.

AT+QMTPUB Publish Messages	
Test Command **AT+QMTPUB=?**	Response **+QMTPUB: (**list of supported **<TCP_connectID>**s)**,(**range

	of supported **<msgID>**s),(range of supported **<QoS>**s),(list of supported **<retain>**s),**<topic>**,(range of supported **<len>**s) **OK**
Write Command Publish messages with unfixed length **AT+QMTPUB=<TCP_connectID>,<msgID>,<QoS>,<retain>,<topic>**	Response **>** After **>** is responded, input the data to be sent. Press "**CTRL**" + "**Z**" to send, and press "**ESC**" to cancel the operation. **OK** **+QMTPUB: <TCP_connectID>,<msgID>,<result>[,<value>]** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Publish messages with fixed length **AT+QMTPUB=<TCP_connectID>,<msgID>,<QoS>,<retain>,<topic>,<len>**	Response **>** After **>** is responded, input the data to be sent. When the data length reaches the length specified by **<len>**, it will be sent automatically. **OK** **+QMTPUB: <TCP_connectID>,<msgID>,<result>[,<value>]** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	**<pkt_timeout>** × **<retry_times>** (default 15 s), determined by network
Characteristics	The command takes effect immediately. The configurations are not saved.

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Currently only one MQTT instance is supported, the value is always 0.
<msgID>	Integer type. Message identifier. Range: 0–65535. It is 0 only when **<QoS>**=0.
<QoS>	Integer type. The QoS level at which the client publishes 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 Do not retain
1 Retain

<topic> String type. Topic that needs to be published. The maximum length is 255 bytes.

<len> Integer type. Length of message to be published. Range: 0–1024. Unit: byte.

<result> Integer type. Results of the command execution.
0 Data packet sent successfully and ACK received from server (published messages do not require ACK when **<QoS>**=0)
1 Data packet retransmitted
2 Failed sending data packet

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

<pkt_timeout> Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 10. Unit: s.

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

<err> Integer type. Error codes. See ***Chapter 5*** for details.

NOTE

1. If this command is executed successfully and returns **OK**, the client can continue to publish new data packets. The maximum quantity of to-be-transmitted data packets should not be greater than the maximum quantity of inflight window size (5).
2. After executing this command, the client is ready to send data, namely, MQTT payload. When the input data reaches the maximum length 1024 bytes, the data will be sent automatically. If the input bytes is shorter than the maximum bytes, you need to press "**CTRL**" + "**Z**" to send the data.
3. PUBLISH messages can be sent either by a publisher to the server, or by the server to a subscriber. When a server publishes messages to a subscriber, the following URC is returned to notify MCU of the data sent by MQTT server:
   +QMTRECV: <TCP_connectID>,<msgID>,<topic>,<payload_len>,<payload>
   For more details about the URC information, please refer to ***Chapter 4.2***.

4 MQTT Related URCs

This chapter describes MQTT related URCs.

Table 2: MQTT Related URCs

No.	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_len>,<payload>**	Report the URC when the client has received the data from MQTT server.

4.1. +QMTSTAT URC to Indicate the State Change of MQTT Link Layer

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

+QMTSTAT URC to Indicate State Change of 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. Currently only one MQTT instance is supported, the value is always 0.
<err_code>	Integer type. Error codes. See ***Table 3*** for details.

Table 3: Error Codes of the URC +QMTSTAT

<err_code>	Description	How to handle
1	Connection is closed or reset by a peer.	Execute **AT+QMTOPEN** and reopen MQTT connection.

2	Sending PINGREQ data packet timed out or failed.	Deactivate PDP then active PDP and reopen MQTT connection.
3	Sending CONNECT packet timed out or failed.	1. Check whether the input username and password are correct. 2. Make sure the client ID is not used. 3. Reopen MQTT connection and try to send CONNECT data packet to server again.
4	Receiving CONNACK packet timed out or failed.	1. Check whether the input username and password are correct. 2. Make sure the client ID is not used. 3. Reopen MQTT connection and try to send CONNECT data packet to server again.
5	The client sends DISCONNECT packet to sever and the server closes MQTT connection.	This is a normal process.
6	The client closes MQTT connection due to packet sending failure all the time.	1. Make sure the data are 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 URC to Notify the MCU to Read MQTT Packet Data

This URC begins with **+QMTRECV:** and it is used to notify the MCU to read the received MQTT packet data received from MQTT server.

+QMTRECV URC to Notify the MCU to Read MQTT Packet Data	
+QMTRECV: <TCP_connectID>,<msgID>,<topic>,<payload_len>,<payload>	Notify the MCU to read the received data received from MQTT server.

Parameter

<TCP_connectID>	Integer type. MQTT socket identifier. Currently only one MQTT instance is supported, the value is always 0.
<msgID>	Integer type. The message identifier. Range: 0–65535.
<topic>	String type. The topic received from MQTT server.
<payload_len>	Integer type. The length of payload.
<payload>	String type. The payload that relates to the topic name.

5 Summary of Error Codes

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

Table 4: Common Error Codes (27.007)

Code of <err>	Meaning
3	Operation not allowed
4	Operation not supported
23	Memory failure
30	No network service
50	Incorrect parameters
51	Command implemented but currently disabled
52	Command aborted by user
159	Uplink busy/flow control

Table 5: Common Error Codes (27.005)

Code of <err>	Meaning
300	ME failure
301	SMS service of ME reserved
302	Operation not allowed
303	Operation not supported
304	Invalid PDU mode parameter
305	Invalid text mode parameter

310	USIM not inserted
311	USIM PIN required
312	PH-USIM PIN required
313	USIM failure
314	USIM busy
315	USIM error
316	USIM PUK required
317	USIM PIN2 required
318	USIM PUK2 required
320	Memory failure
321	Invalid memory index
322	Memory full
330	SMSC address unknown
331	No network service
332	Network timeout
340	No +CNMA acknowledgement expected
500	Unknown error

NOTE

AT+CMEE=<n> can disable (**<n>**=0) or enable (**<n>**=1) the use of final result code **"+CME ERROR:<err>"**. When **<n>** is set to the default value 0, the response is returned as **ERROR**; when **<n>** is set to 1, the response is returned as **+CME ERROR: <err>**. For more details, please refer to ***document [1]***.

6 Examples

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

```
//Configure Alibaba device information for Alibaba cloud.
AT+QMTCFG="aliauth",0,"oyjtmPl5a5j","MQTT_TEST","wN9Y6pZSIIy7Exa5qVzcmigEGO4kAazZ"
OK

AT+QMTOPEN=?
+QMTOPEN: (0),<host_name>,(0-65535)

OK

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

+QMTOPEN: 0,0                       //Opened the network connection of MQTT client successfully.

AT+QMTOPEN?
+QMTOPEN: 0,"iot-as-mqtt.cn-shanghai.aliyuncs.com",1883

OK

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

OK

//If Alibaba Cloud is connected, you can use AT+QMTCFG="aliauth" to configure the device information
in advance, so that you don't have to provide username/password every time.
AT+QMTCONN=0,"clientExample"          //Connect a client to MQTT server.
OK

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

AT+QMTSUB=?
+QMTSUB: (0),(1-65535),<topic>,(0-2)[,<topic>,(0-2)...]
```

```
OK

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

+QMTSUB: 0,1,0,2

AT+QMTSUB=0,1,"topic/pub",0               //Subscribe to the topic "topic/pub".
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",36,This is the payload related to topic

AT+QMTUNS=0,2,"topic/example"             // Unsubscribe from the topic "topic/example".
OK

+QMTUNS: 0,2,0

AT+QMTPUB=?
+QMTPUB: (0),(0-65535),(0-2),(0,1),<topic>,(0-1024)

OK

AT+QMTPUB=0,0,0,0,"topic/pub"             //Publish the message.
>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
                                          1024 bytes and the data that beyond 1024 bytes are omitted.
                                          After the input is complete, press “CTRL“ + “Z" to send.
OK

+QMTPUB: 0,0,0

//If a client subscribes to a topic named "topic/pub", the module reports the following information.
+QMTRECV: 0,0,"topic/pub",30,This is test data, hello MQTT.

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

+QMTDISC: 0,0
```

7 Appendix References

Table 6: Related Document

Document Name
[1] Quectel_BC68-GV&BC95-GV_AT_Commands_Manual

Table 7: Terms and Abbreviations

Abbreviation	Description
ACK	Acknowledgement
MQTT	Message Queuing Telemetry Transport
QoS	Quality of Service
TCP	Transmission Control Protocol
IP	Internet Protocol
URC	Unsolicited Result Code
URL	Uniform Resource Locator