BC26 MQTT Application Note

LPWA Module Series

Rev. BC26_MQTT_Application_Note_V1.1

Date: 2019-04-22

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.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

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

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

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2018-07-09	Louis GU	Initial
1.1	2019-04-22	Milo WANG	1. Optimized AT+QMTCFG command to enable configuration of MQTT version, sent/received data format, and the data echo mode for data transfer. 2. Added the maximum length of <topic> for AT+QMTSUB and AT+QMTUNS. 3. Added the maximum length of <topic> and <msg> for AT+QMTPUB. 4. Updated the response of all Write Commands into “ERROR” when an error occurs, and deleted the description of <err> codes. 5. Updated the example in Chapter 5.

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 BC26 module through AT commands.

2 MQTT Data Interaction

This chapter gives the data interaction mechanism of MQTT function.

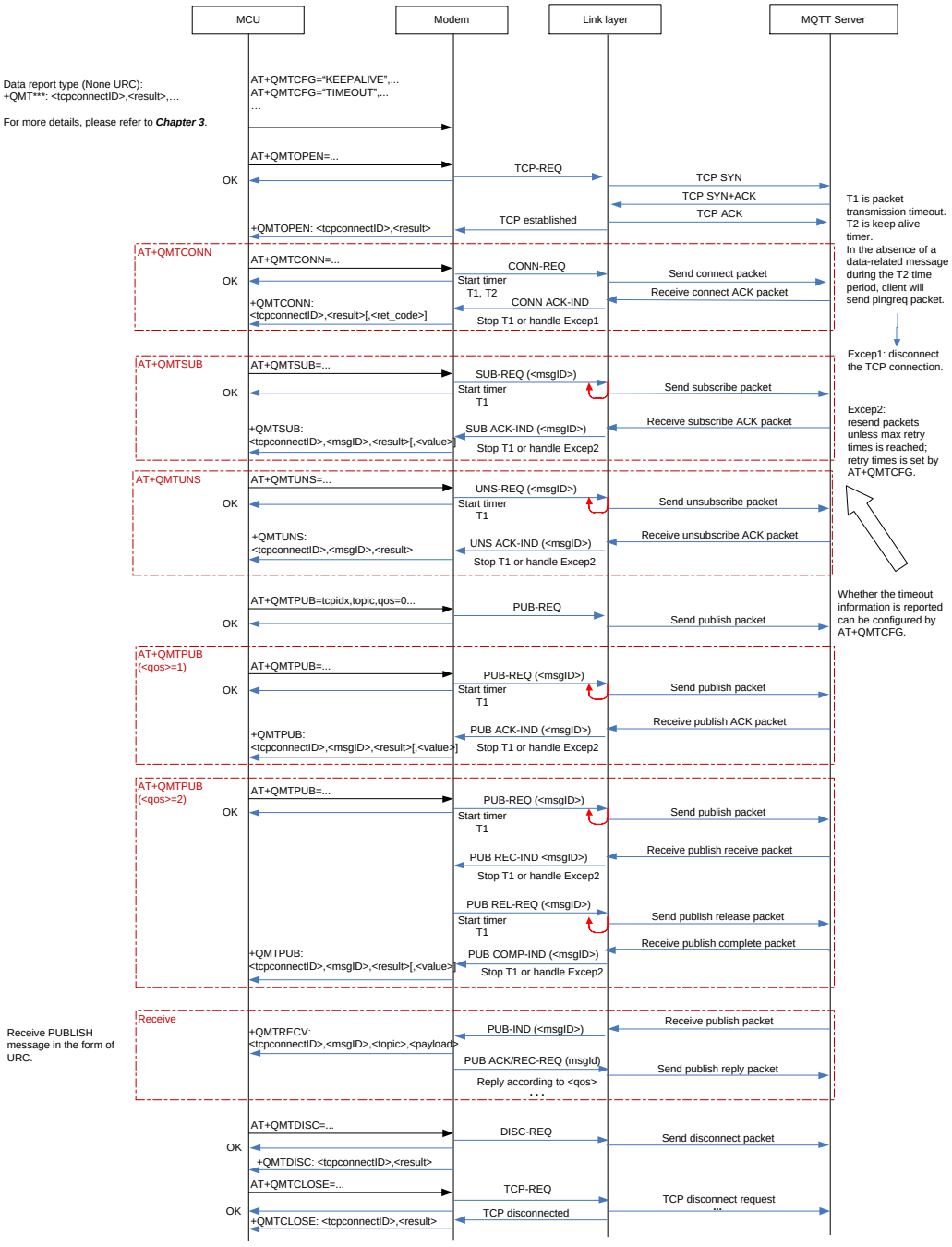

Figure 1: MQTT Data Interaction Mechanism

3 MQTT Related AT Commands

This chapter presents the AT commands for operating MQTT function.

3.1. AT Command Syntax

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 UE.

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: "echomode",(0-5),(0,1)** **+QMTCFG: "dataformat",(0-5),(0,1),(0,1)** **+QMTCFG: "keepalive",(0-5),(0-3600)** **+QMTCFG: "session",(0-5),(0,1)** **+QMTCFG: "timeout",(0-5),(1-60),(1-10),(0,1)** **+QMTCFG: "will",(0-5),(0,1),(0-2),(0,1),"will_topic","will_msg"** **+QMTCFG: "version",(0-5),(3,4)**

	+QMTCFG: "aliauth",(0-5),"productkey","devicename","devicesecret" **OK**
Write Command Configure Will information **AT+QMTCFG="WILL",<tcpconnectID>[,<will_fg>[,<will_qos>,<will_retain>,"<will_topic>","<will_msg>"]]**	Response **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 any error: **ERROR**
Write Command Configure timeout of message delivery **AT+QMTCFG="TIMEOUT",<tcpconnectID>[,<pkt_timeout>[,<retry_times>][,<timeout_notice>]]**	Response **OK** If **<pkt_timeout>**, **<retry_times>**, **<timeout_notice>** are omitted, query the timeout value of message delivery: **+QMTCFG: "timeout",<pkt_timeout>,<retry_times>,<timeout_notice>** **OK** If there is any error: **ERROR**
Write Command Configure the session type **AT+QMTCFG="SESSION",<tcpconnectID>[,<clean_session>]**	Response **OK** If **<clean_session>** is omitted, query the session type: **+QMTCFG: "session",<clean_session>** **OK** If there is any error: **ERROR**
Write Command Configure the keep-alive time **AT+QMTCFG="KEEPALIVE",<tcpconnectID>[,<keep-alive time>]**	Response **OK** If **<keep-alive time>** is omitted, query the keep-alive time: **+QMTCFG: "keepalive",<keep-alive time>**

	OK If there is any error: **ERROR**
Write Command Configure the device information for AliCloud **AT+QMTCFG="ALIAUTH",<tcpconnectID>[,"<product_key>","<device_name>","<device_secret>"]**	Response **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 any error: **ERROR**
Write Command Configure the MQTT protocol version to be used **AT+QMTCFG="VERSION",<tcpconnectID>[,<version>]**	Response **OK** If **<version>** is omitted, query the MQTT protocol version: **[+QMTCFG: "version",<version>]** **OK** If there is any error: **ERROR**
Write Command Configure the format of sent/received data **AT+QMTCFG="dataformat",<tcpconnectID>,[<send_format>,[<recv_format>]]**	Response **OK** If **<send_format>** and **<recv_format>** are both omitted, query the format of sent/received data: **[+QMTCFG: "dataformat",<send_format>,<recv_format>]** **OK** If there is any error: **ERROR**
Write Command Configure whether to echo the input data to UART in data mode **AT+QMTCFG="echomode",<tcpconnectID>,[<echo_mode>]**	Response **OK** If **<echo_mode>** is omitted, query the data echo mode: **[+QMTCFG: "echomode",<echo_mode>]**

	OK If there is any error: **ERROR**
Maximum Response Time	300ms

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<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 10. 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 disconnects. 1 The server must discard any previously maintained information about the client and treat the connection as “clean”.
<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 The client is not disconnected
<product_key>	Product key issued by AliCloud
<device_name>	Device name issued by AliCloud
<device_secret>	Device secret key issued by AliCloud

<version>	Version of MQTT protocol 3 MQTT v3.1 4 MQTT v3.1.1
<send_format>	The format of sent data 0 Text mode 1 Hex mode
<recv_format>	The format of received data 0 Text mode 1 Hex mode
<echo_mode>	Integer type. Whether to echo the input data to UART in data mode 0 Do not echo the input data to UART 1 Echo the input data to UART

NOTES

1. If **<will_flag>**=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. Care must be taken to ensure message delivery does not time out while it is still being sent.
4. **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.
5. **<echo_mode>** is only valid in data mode transferring.

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 any error: **ERROR**
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 size is 100 bytes.
<port>	The port of the server. The range is 1-65535.
<result>	Result of the command execution -1 Failed to open network 0 Opened network successfully 1 Wrong parameter 2 MQTT identifier is occupied 3 Failed to activate PDP 4 Failed to parse domain name 5 Network disconnection error

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 any error: **ERROR**
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 any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** (default 10s), 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 Sent packet successfully and received ACK from server 1 Packet retransmission 2 Failed to send packet
<state>	MQTT connection state

	1 MQTT is initial 2 MQTT is connecting 3 MQTT is connected 4 MQTT is disconnecting
<ret_code>	Connect return code 0 Connection Accepted 1 Connection Refused: Unacceptable Protocol Version 2 Connection Refused: Identifier Rejected 3 Connection Refused: Server Unavailable 4 Connection Refused: Bad User Name or Password 5 Connection Refused: Not Authorized
<pkt_timeout>	Timeout of the packet delivery. The range is 1-60. The default value is 10. Unit: second.

NOTE

If a client with the same Client ID is already connected to the server, the "older" client will be disconnected by the server automatically after 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 any error: **ERROR**
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

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 any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** * **<retry_times>** (default 40s), determined by network

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<msgID>	Message identifier of packet. The range is 1-65535.
<topic>	Topic that the client wants to subscribe to or unsubscribe from. The maximum length is 255 bytes.
<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. At the same time, the value is 128, indicating that the subscription was rejected by the server. 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 10. Unit: second.
<retry_times>	Retry times when packet delivery times out. The range is 0-10. The default value is 3.

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 "in flight" 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 any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** * **<retry_times>** (default 40s), determined by network

Parameter

<tcpconnectID>	MQTT socket identifier. The range is 0-5.
<msgID>	Message identifier of packet. The range is 1-65535.
<topic>	Topic that the client wants to subscribe to or unsubscribe from. Maximum length is 255 bytes.
<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 10. Unit: second.
<retry_times>	Retry times when packet delivery times out. The range is 0-10. The default value is 3.

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>"**,**"<msg>"** **OK**
Write Command **AT+QMTPUB=<tcpconnectID>,<msgID>,<qos>,<retain>,"<topic>"** After "**>**" is responded, the module enters data mode and then please input the data to be sent. Tap "**Ctrl+Z**" to send data, and tap "**Esc**" to cancel the operation.	Response **OK** **+QMTPUB: <tcpconnectID>,<msgID>,<result>[,<value>]** If there is any error: **ERROR**
Write Command **AT+QMTPUB=<tcpconnectID>,<msgID>,<qos>,<retain>,"<topic>","<msg>"**	Response **OK** **+QMTPUB: <tcpconnectID>,<msgID>,<result>[,<value>]** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	**<pkt_timeout>** * **<retry_times>** (default 40s), 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. The maximum length is 255 bytes.
<msg>	Message that needs to be published. Maximum length is 700 bytes. If in data mode, the maximum length is 1024 bytes.
<result>	Result of the command execution 0 Sent packet successfully and received ACK from server (message published when **<qos>**=0 does not require ACK) 1 Packet retransmission 2 Failed to send packet
<value>	If **<result>** is 1, it means the times of packet retransmission. If **<result>** is 0 or 2, it will not be presented.

NOTE

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 from MQTT server:

+QMTRECV: <tcpconnectID>,<msgID>,<topic>,<payload>

For more details about the URC description, please refer to ***Chapter 4.2***.

4 MQTT Related URCs

This chapter gives MQTT related URCs and their descriptions.

Table 1: MQTT Related URCs

Index	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.

4.1. "+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 2: Error Codes of the URC

Code of <err>	Description	How to do
1	Connection is closed or reset by peer.	Execute **AT+QMTOPEN** command and reopen MQTT connection.

2	Sending PINGREQ packet timed out or failed.	Deactivate PDP first, and then active PDP and reopen MQTT connection.
3	Sending CONNECT packet timed out or failed	1. Check whether the inputted user name and password are correct or not. 2. Make sure the client ID is not used. 3. Reopen MQTT connection and try to send CONNECT packet to server again.
4	Receiving CONNACK packet timed out or failed	1. Check whether the inputted user name and password are correct. 2. Make sure the client ID is not used. 3. Reopen MQTT connection and try to send CONNECT packet to server again.
5	The client sends DISCONNECT packet to sever but the server is initiative to close MQTT connection.	This is a normal process.
6	The client is initiative to close MQTT connection due to packet sending failure all the time.	1. Make sure the data is correct. 2. Try to reopen MQTT connection since there may be network congestion or an error.
7	The link is not alive or the server is unavailable.	Make sure the link is alive or the server is available currently.
8-255	Reserved for future use	

4.2. “+QMTRECV” URC to Notify the Host to Read MQTT Packet Data

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

“+QMTRECV” URC to Notify 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.

5 Example

This chapter gives an example to explain how to use MQTT related commands.

```
AT+QMTOPEN=?
+QMTOPEN: (0-5),"<host_name>",<port>

OK

//Open a network for MQTT client.
AT+QMTOPEN=0,"220.180.239.212",8401
OK

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

AT+QMTOPEN?
+QMTOPEN: 0,"220.180.239.212",8401

OK

AT+QMTCONN=?
+QMTCONN: (0-5),“<clientID>”[,“<username>”[,“<password>”]]

OK

AT+QMTCONN=0,“clientExample”
OK

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

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

OK

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

+QMTSUB: 0,1,0,2
```

```
AT+QMTSUB=0,1,“topic/pub”,0
OK

+QMTSUB: 0,1,0,0

//If a client subscribes to a topic and other devices publish the same topic to the server, the module will report the following information.
+QMTRECV: 0,0,“topic/example”,“This is the payload related to topic”

//Unsubscribe from topics.
AT+QMTUNS=0,2,“topic/example”
OK

+QMTUNS: 0,2,0

AT+QMTPUB=?
+QMTPUB: (0-5),<msgID>,<qos>,<retain>,“<topic>”,“<msg>”

OK

//Publish messages.
AT+QMTPUB=0,0,0,0,“topic/pub”,“hello MQTT.”
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”,“hello MQTT.”

//Publish messages with “>” data mode.
AT+QMTPUB=0,0,0,0,“topic/pub”
>
This is test data, hello MQTT.          //Tap “CTRL+Z” to send
OK

+QMTPUB: 0,0,0

//If a client subscribes to a topic named “topic/pub” and other devices publish the same topic to the server, the module will report the following information.
+QMTRECV: 0,0,"topic/pub","This is test data, hello MQTT."

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

```
OK

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

6 Appendix A References

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

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