

# WCDMA UCxx MQTT

## Application Note

**UMTS/HSPA(+) Module Series**

Rev. WCDMA\_UCxx\_MQTT\_Application\_Note\_V1.0

Date: 2018-11-13

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.**

7<sup>th</sup> Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: [info@quectel.com](mailto: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](mailto: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**

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

***Copyright © Quectel Wireless Solutions Co., Ltd. 2018. All rights reserved.***

# About the Document

## History

Revision	Date	Author	Description
1.0	2018-11-13	Bryce LI	Initial

---

## Contents

About the Document.....	2
Contents .....	3
Table Index.....	4
<b>1 Introduction .....</b>	<b>5</b>
<b>2 MQTT Data Interaction.....</b>	<b>6</b>
<b>3 MQTT Related AT Commands .....</b>	<b>7</b>
3.1. AT Command Syntax.....	7
3.2. Description of MQTT Related AT Commands.....	7
3.2.1. AT+QMTCFG Configure Optional Parameters of MQTT .....	7
3.2.2. AT+QMTOPEN Open a Network for MQTT Client.....	12
3.2.3. AT+QMTCLOSE Close a Network for MQTT Client .....	13
3.2.4. AT+QMTCONN Connect a Client to MQTT Server.....	14
3.2.5. AT+QMTDISC Disconnect a Client from MQTT Server .....	15
3.2.6. AT+QMTSUB Subscribe to Topics .....	16
3.2.7. AT+QMTUNS Unsubscribe from Topics.....	17
3.2.8. AT+QMTPUB Publish Messages .....	18
3.2.9. AT+QMTRECV Read Messages from Buffers .....	20
<b>4 MQTT Related URCs .....</b>	<b>22</b>
4.1. "+QMTSTAT" URC to Indicate State Change in MQTT Link Layer.....	22
4.2. "+QMTRECV" URC to Notify the Host to Read MQTT Packet Data .....	23
<b>5 Examples .....</b>	<b>24</b>
5.1. Example of MQTT Operation without SSL.....	24
5.2. Example of MQTT Operation with SSL.....	26
<b>6 Appendix A References.....</b>	<b>29</b>

## Table Index

TABLE 1: TYPES OF AT COMMANDS AND RESPONSES .....	7
TABLE 2: MQTT RELATED URCS .....	22
TABLE 3: ERROR CODES OF THE URC .....	22
TABLE 4: RELATED DOCUMENTS .....	29
TABLE 5: TERMS AND ABBREVIATIONS .....	29

# 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 UCxx modules through AT commands.

In this document, UCxx refers to UC15 and UC20\* modules.

## NOTE

“\*” means MQTT for UC20 is under development.

## 2 MQTT Data Interaction

This chapter gives the data interaction mechanism of MQTT function.

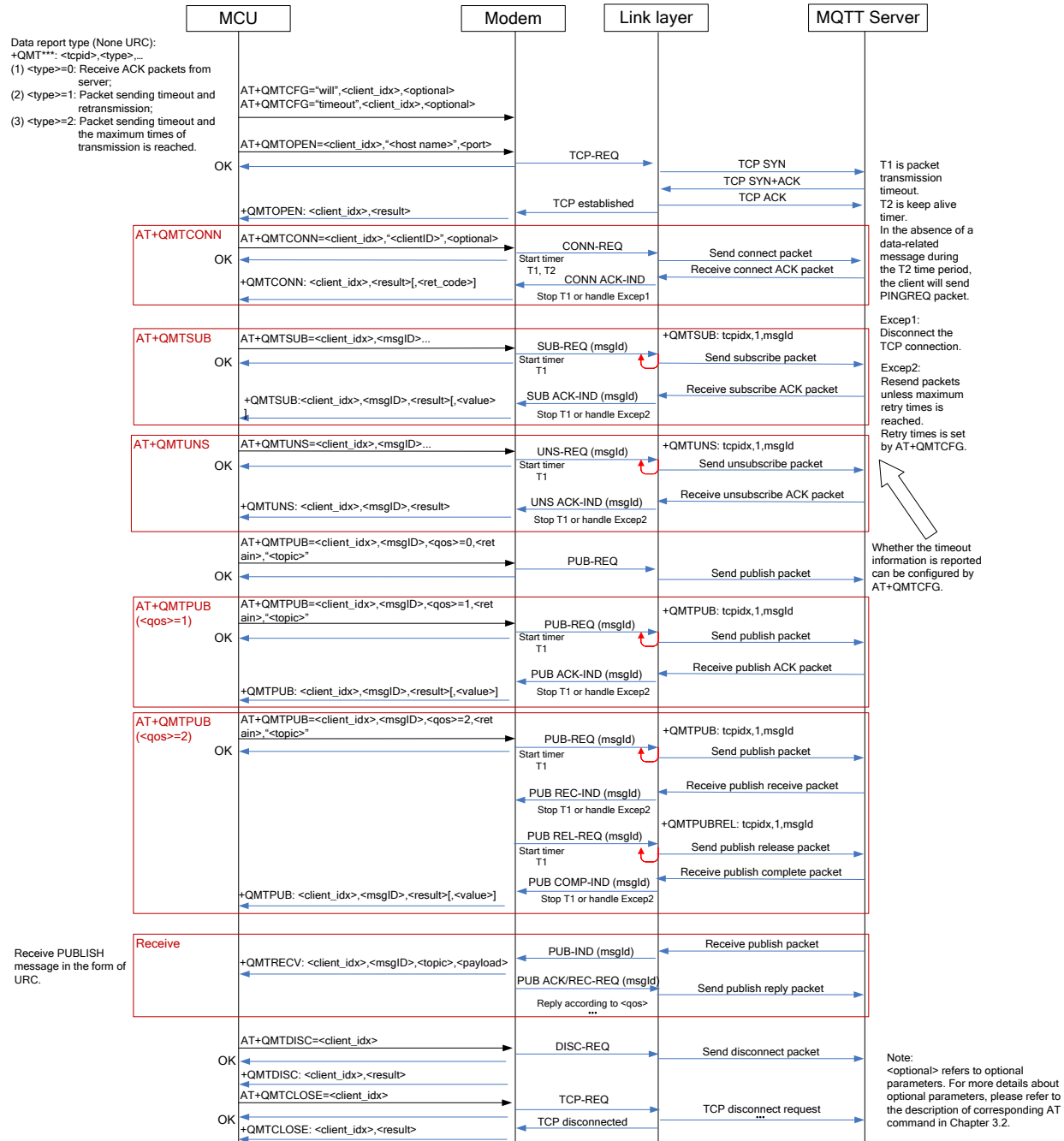


Figure 1: MQTT Data Interaction Diagram

## 3 MQTT Related AT Commands

This chapter presents the AT commands for operating MQTT function.

### 3.1. AT Command Syntax

Table 1: Types of AT Commands and Responses

Test Command	AT+<x>=?	This command returns the list of parameters and value ranges set by the corresponding Write Command or internal processes.
Read Command	AT+<x>?	This command returns the currently set value of the parameter or parameters.
Write Command	AT+<x>=<...>	This command sets the user-definable parameter values.
Execution Command	AT+<x>	This command reads non-variable parameters affected by internal processes in the 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: "version",(0-5),(3,4) +QMTCFG: "pdpcid",(0-5),(1-16) +QMTCFG: "ssl",(0-5),(0,1),(0-5) +QMTCFG: "keepalive",(0-5),(0-3600) +QMTCFG: "session",(0-5),(0,1) +QMTCFG: "timeout",(0-5),(1-60),(1-10),(0,1) +QMTCFG: "will",(0-5),(0,1),(0-2),(0,1),"willtopic","willmess age" +QMTCFG: "recv/mode",(0-5),(0,1)
-----------------------------	--



	<p>+QMTCFG: "aliauth",(0-5),"product key","device name", "device secret" +QMTCFG: "qmtping",(0-5),(5-60)</p> <p>OK</p>
<p>Write Command Configure the MQTT protocol version <b>AT+QMTCFG="version",&lt;client_idx&gt; [,&lt;vsn&gt;]</b></p>	<p>Response If &lt;vsn&gt; is present, configure the MQTT protocol version: OK</p> <p>If &lt;vsn&gt; is omitted, query the MQTT protocol version: +QMTCFG: "version",&lt;vsn&gt;</p> <p>OK</p> <p>If there is any error: ERROR</p>
<p>Write Command Configure the PDP to be used by the MQTT client <b>AT+QMTCFG="pdpcid",&lt;client_idx&gt;[,&lt;cid&gt;]</b></p>	<p>Response If &lt;cid&gt; is present, configure the PDP to be used by the MQTT client: OK</p> <p>If &lt;cid&gt; is omitted, query the PDP used by the MQTT client: +QMTCFG: "pdpcid",&lt;cid&gt;</p> <p>OK</p> <p>If there is any error: ERROR</p>
<p>Write Command Configure Will information <b>AT+QMTCFG="will",&lt;client_idx&gt;[,&lt;will_fg&gt;[,&lt;will_qos&gt;,&lt;will_retain&gt;,"&lt;will_topic&gt;",&lt;will_msg&gt;"]]</b></p>	<p>Response If &lt;will_fg&gt;, &lt;will_qos&gt;, &lt;will_retain&gt;, "&lt;will_topic&gt;" and "&lt;will_msg&gt;" are present, configure the Will information: OK</p> <p>If &lt;will_fg&gt;, &lt;will_qos&gt;, &lt;will_retain&gt;, "&lt;will_topic&gt;" and "&lt;will_msg&gt;" are omitted, query the Will information: +QMTCFG: "will",&lt;will_fg&gt;[,&lt;will_qos&gt;,&lt;will_retain&gt;,"&lt;will_topic&gt;",&lt;will_msg&gt;"]</p> <p>OK</p> <p>If there is any error: ERROR</p>
<p>Write Command Configure timeout of message delivery</p>	<p>Response If &lt;pkt_timeout&gt;, &lt;retry_times&gt; and &lt;timeout_notice&gt; are</p>

<p><b>AT+QMTCFG="timeout",&lt;client_idx&gt;[,&lt;pkt_timeout&gt;,&lt;retry_times&gt;][,&lt;timeout_notice&gt;]]</b></p>	<p>present, configure the timeout of message delivery: <b>OK</b></p> <p>If &lt;pkt_timeout&gt;, &lt;retry_times&gt; and &lt;timeout_notice&gt; are omitted, query the timeout of message delivery: <b>+QMTCFG: "timeout",&lt;pkt_timeout&gt;,&lt;retry_times&gt;,&lt;timeout_notice&gt;</b></p> <p><b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command Configure the session type <b>AT+QMTCFG="session",&lt;client_idx&gt;[,&lt;clean_session&gt;]</b></p>	<p>Response If &lt;clean_session&gt; is present, configure the session type: <b>OK</b></p> <p>If &lt;clean_session&gt; is omitted, query the session type: <b>+QMTCFG: "session",&lt;clean_session&gt;</b></p> <p><b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command Configure the keep-alive time <b>AT+QMTCFG="keepalive",&lt;client_idx&gt;[,&lt;keep-alive time&gt;]</b></p>	<p>Response If &lt;keep-alive time&gt; is present, configure the keep-alive time: <b>OK</b></p> <p>If &lt;keep-alive time&gt; is omitted, query the keep-alive time: <b>+QMTCFG: "keepalive",&lt;keep-alive time&gt;</b></p> <p><b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command Configure the MQTT SSL mode and SSL context index <b>AT+QMTCFG="ssl",&lt;client_idx&gt;[,&lt;sslenable&gt;[,&lt;sslctx_idx&gt;]]</b></p>	<p>Response If &lt;sslenable&gt; and &lt;sslctx_idx&gt; are present, configure the MQTT SSL mode and SSL context index: <b>OK</b></p> <p>If &lt;sslenable&gt; and &lt;sslctx_idx&gt; are omitted, query the MQTT SSL mode and SSL context index: <b>+QMTCFG: "ssl",&lt;sslenable&gt;[,&lt;sslctx_idx&gt;]</b></p> <p><b>OK</b></p>

	<p>If there is any error: <b>ERROR</b></p>
<p>Write Command</p> <p>Configure receiving mode when data is received from server</p> <p><b>AT+QMTCFG="recv/mode",&lt;client_idx&gt;[,&lt;msg_recv_mode&gt;]</b></p>	<p>Response</p> <p>If <b>&lt;msg_recv_mode&gt;</b> is present, configure the receiving mode when data is received from server: <b>OK</b></p> <p>If <b>&lt;msg_recv_mode&gt;</b> is omitted, query the receiving mode when data is received from server: <b>+QMTCFG: "recv/mode",&lt;msg_recv_mode&gt;</b></p> <p><b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command</p> <p>Configure the <b>&lt;qmtping_interval&gt;</b></p> <p><b>AT+QMTCFG="qmtping",&lt;client_idx&gt;[,&lt;qmtping_interval&gt;]</b></p>	<p>Response</p> <p>If <b>&lt;qmtping_interval&gt;</b> is present, configure the <b>&lt;qmtping_interval&gt;</b>: <b>OK</b></p> <p>If <b>&lt;qmtping_interval&gt;</b> is omitted, query the <b>&lt;qmtping_interval&gt;</b>: <b>+QMTCFG: "qmtping",&lt;qmtping_interval&gt;</b></p> <p><b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>
<p>Write Command</p> <p>Configure Alibaba device information for Alibaba Cloud</p> <p><b>AT+QMTCFG="aliauth",&lt;client_idx&gt;[, "&lt;product_key&gt;",&lt;device_name&gt;",&lt;device_secret&gt;"]</b></p>	<p>Response</p> <p>If <b>&lt;product_key&gt;</b>, <b>&lt;device_name&gt;</b> and <b>&lt;device_secret&gt;</b> are present, configure the device information: <b>OK</b></p> <p>If <b>&lt;product_key&gt;</b>, <b>&lt;device_name&gt;</b> and <b>&lt;device_secret&gt;</b> are omitted, query the device information: <b>[+QMTCFG: "aliauth",&lt;product_key&gt;,&lt;device_name&gt;,&lt;device_secret&gt;"]</b></p> <p><b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>

Maximum Response Time	300ms
-----------------------	-------

## Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. The range is 0-5.
<b>&lt;vsn&gt;</b>	Integer type. MQTT protocol version 3 MQTT protocol v3.1 4 MQTT protocol v3.1.1
<b>&lt;cid&gt;</b>	Integer type. The PDP to be used by the MQTT client. The range is 1-16. The default value is 1.
<b>&lt;will_fg&gt;</b>	Integer type. Configure the Will flag 0 Ignore the Will flag configuration 1 Require the Will flag configuration
<b>&lt;will_qos&gt;</b>	Integer type. Quality of service for message delivery 0 At most once 1 At least once 2 Exactly once
<b>&lt;will_retain&gt;</b>	Integer type. The Will retain flag is only used on PUBLISH messages. 0 When a client sends a PUBLISH message to a server, the server will not hold on to the message after it has been delivered to the current subscribers 1 When a client sends a PUBLISH message to a server, the server should hold on to the message after it has been delivered to the current subscribers
<b>&lt;will_topic&gt;</b>	String type. Will topic string
<b>&lt;will_msg&gt;</b>	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.
<b>&lt;pkt_timeout&gt;</b>	Integer type. Timeout of the packet delivery. The range is 1-60. The default value is 5. Unit: second.
<b>&lt;retry_times&gt;</b>	Integer type. Retry times when packet delivery times out. The range is 1-10. The default value is 3.
<b>&lt;timeout_notice&gt;</b>	Integer type. 0 Not report timeout message when transmitting packet 1 Report timeout message when transmitting packet
<b>&lt;clean_session&gt;</b>	Integer type. Configure the session type 0 The server must store the subscriptions of the client after it disconnects. 1 The server must discard any previously maintained information about the client and treat the connection as "clean".
<b>&lt;keep-alive time&gt;</b>	Integer type. 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

<b>&lt;sslenable&gt;</b>	Integer type. Configure the MQTT SSL mode 0 Use normal TCP connection for MQTT 1 Use SSL TCP secure connection for MQTT
<b>&lt;sslctx_idx&gt;</b>	Integer type. SSL context index. The range is 0-5.
<b>&lt;msg_rcv_mode&gt;</b>	Integer type. Configure 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.
<b>&lt;qmtping_interval&gt;</b>	Integer type, the longest time for <b>+QMTSTAT</b> report after <b>+QMTPING</b> . The default value is 5. Unit: second.
<b>&lt;product_key&gt;</b>	String type. Product key issued by Alibaba Cloud
<b>&lt;device_name&gt;</b>	String type. Device name issued by Alibaba Cloud
<b>&lt;device_secret&gt;</b>	String type. Device secret key issued by Alibaba Cloud

## NOTES

1. When MQTT connection has been established, please note that **ERROR** will be responded if optional parameters are present. And only query can be implemented after the establishment of MQTT connection.
2. If **<will\_flag>**=1, **<will\_qos>**, **<will\_retain>**, **<will\_topic>** and **<will\_msg>** must be present. Otherwise they will be omitted.
3. **<clean\_session>**=0 is only effective when the server supports the operation.
4. If MQTT connection is configured to SSL mode, **<sslctx\_idx>** must be present. Also, customers need to use **AT+QSSLCFG** command to configure the SSL version, cipher suite, secure level, CA certificate, client certificate, client key and ignorance of RTC time, which will be used in MQTT SSL handshake procedure.
5. Care must be taken to ensure message delivery does not time out while it is still being sent.
6. **AT+QMTCFG="aliauth"** command is only used for Alibaba Cloud. If it is configured, the parameters **<username>** and **<password>** in **AT+QMTCONN** command can be omitted.

### 3.2.2. AT+QMTOPEN Open a Network for MQTT Client

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

#### AT+QMTOPEN Open a Network for MQTT Client

Test Command <b>AT+QMTOPEN=?</b>	Response <b>+QMTOPEN:</b> (list of supported <b>&lt;client_idx&gt;</b> s)," <b>&lt;host_name&gt;</b> ",(list of supported <b>&lt;port&gt;</b> s)  <b>OK</b>
Read Command <b>AT+QMTOPEN?</b>	Response If the network has been opened: <b>+QMTOPEN:</b> <b>&lt;client_idx&gt;</b> ," <b>&lt;host_name&gt;</b> ", <b>&lt;port&gt;</b>

	<p><b>OK</b></p> <p>Otherwise:</p> <p><b>OK</b></p>
<p>Write Command</p> <p><b>AT+QMTOPEN=&lt;client_idx&gt;,"&lt;host_name&gt;",&lt;port&gt;</b></p>	<p>Response</p> <p><b>OK</b></p> <p><b>+QMTOPEN: &lt;client_idx&gt;,&lt;result&gt;</b></p> <p>If there is any error:</p> <p><b>ERROR</b></p>
Maximum Response Time	120s, determined by network

## Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. The range is 0-5.
<b>&lt;host_name&gt;</b>	String type. The address of the server. It could be an IP address or a domain name. The maximum size is 100 bytes.
<b>&lt;port&gt;</b>	Integer type. The port of the server. The range is 1-65535.
<b>&lt;result&gt;</b>	Integer type. Result of the command execution
	<p>-1 Failed to open network</p> <p>0 Network opened successfully</p> <p>1 Wrong parameter</p> <p>2 MQTT identifier is occupied</p> <p>3 Failed to activate PDP</p> <p>4 Failed to parse domain name</p> <p>5 Network connection error</p>

### 3.2.3. AT+QMTCLOSE Close a Network for MQTT Client

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

<b>AT+QMTCLOSE Close a Network for MQTT Client</b>	
<p>Test Command</p> <p><b>AT+QMTCLOSE=?</b></p>	<p>Response</p> <p><b>+QMTCLOSE: (list of supported &lt;client_idx&gt;s)</b></p> <p><b>OK</b></p>
<p>Write Command</p> <p><b>AT+QMTCLOSE=&lt;client_idx&gt;</b></p>	<p>Response</p> <p><b>OK</b></p> <p><b>+QMTCLOSE: &lt;client_idx&gt;,&lt;result&gt;</b></p>

	If there is any error: <b>ERROR</b>
Maximum Response Time	30s

## Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. The range is 0-5.
<b>&lt;result&gt;</b>	Integer type. 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 <b>AT+QMTCONN=?</b>	Response <b>+QMTCONN:</b> (list of supported <client_idx>s),“<client_idx>”[“<username>”][“<password>”]  <b>OK</b>
Read Command <b>AT+QMTCONN?</b>	Response If there is <state> of connection: <b>+QMTCONN:</b> <client_idx>,<state>  <b>OK</b>  If there is no <state> of connection: <b>OK</b>
Write Command <b>AT+QMTCONN=&lt;client_idx&gt;,”&lt;client_idx&gt;”[“&lt;username&gt;”][“&lt;password&gt;”]]</b>	Response <b>OK</b>  <b>+QMTCONN:</b> <client_idx>,<result>[,<ret_code>]  If there is any error: <b>ERROR</b>
Maximum Response Time	<pkt_timeout> (default 5s), determined by network

## Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. The range is 0-5.
<b>&lt;clientID&gt;</b>	String type. The client identifier string.
<b>&lt;username&gt;</b>	String type. User name of the client. It can be used for authentication.
<b>&lt;password&gt;</b>	String type. Password corresponding to the user name of the client. It can be used for authentication.
<b>&lt;result&gt;</b>	Integer type. Result of the command execution 0 Packet sent successfully and ACK received from server 1 Packet retransmission 2 Failed to send packet
<b>&lt;state&gt;</b>	Integer type. MQTT connection state 1 MQTT is initializing 2 MQTT is being connected 3 MQTT is connected 4 MQTT is being disconnected
<b>&lt;ret_code&gt;</b>	Integer type. Connection status return code 0 Connection Accepted 1 Connection Refused: Unacceptable Protocol Version 2 Connection Refused: Identifier Rejected 3 Connection Refused: Server Unavailable 4 Connection Refused: Bad User Name or Password 5 Connection Refused: Not Authorized
<b>&lt;pkt_timeout&gt;</b>	Integer type. Timeout of the packet delivery. The range is 1-60. The default value is 5. Unit: second.

### NOTE

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

### 3.2.5. AT+QMTDISC Disconnect a Client from MQTT Server

The command is used when a client requests a disconnection from MQTT server. A DISCONNECT message is sent from the client to the server to indicate that it is about to close its TCP/IP connection.

#### AT+QMTDISC Disconnect a Client from MQTT Server

Test Command <b>AT+QMTDISC=?</b>	Response <b>+QMTDISC:</b> (list of supported <b>&lt;client_idx&gt;</b> s)  <b>OK</b>
Write Command <b>AT+QMTDISC=&lt;client_idx&gt;</b>	Response <b>OK</b>



	<b>+QMTDISC: &lt;client_idx&gt;,&lt;result&gt;</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	30s

## Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. The range is 0-5.
<b>&lt;result&gt;</b>	Integer type. 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 <b>AT+QMTSUB=?</b>	Response <b>+QMTSUB:</b> (list of supported <client_idx>s),(list of supported <msgID>s),“<topic>”,(list of supported <qos>s)  <b>OK</b>
Write Command <b>AT+QMTSUB=&lt;client_idx&gt;,&lt;msgID&gt;,&lt;topic1&gt;,&lt;qos1&gt;[,&lt;topic2&gt;,&lt;qos2&gt;...]</b>	Response <b>OK</b>  <b>+QMTSUB: &lt;client_idx&gt;,&lt;msgID&gt;,&lt;result&gt;[,&lt;value&gt;]</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	<pkt_timeout> * <retry_times> (default 15s), determined by network

## Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. The range is 0-5.
<b>&lt;msgID&gt;</b>	Integer type. Message identifier of packet. The range is 1-65535.
<b>&lt;topic&gt;</b>	String type. Topic that the client wants to subscribe to or unsubscribe from.
<b>&lt;qos&gt;</b>	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
<b>&lt;result&gt;</b>	Integer type. Result of the command execution	
	0	Sent packet successfully and received ACK from server
	1	Packet retransmission
	2	Failed to send packet
<b>&lt;value&gt;</b>	Integer type.	
	If <b>&lt;result&gt;</b> is 0, it is a vector of granted QoS levels.	
	If <b>&lt;result&gt;</b> is 1, it means the times of packet retransmission.	
	If <b>&lt;result&gt;</b> is 2, it will not be presented.	
<b>&lt;pkt_timeout&gt;</b>	Integer type. Timeout of the packet delivery. The range is 1-60. The default value is 5. Unit: second.	
<b>&lt;retry_times&gt;</b>	Integer type. Retry times when packet delivery times out. The range is 0-10. The default value is 3.	

#### NOTES

1. The **<msgID>** is only present in messages where the QoS bits in the fixed header indicate QoS levels 1 or 2. It must be unique amongst the set of “inflight” messages in a particular direction of communication. It typically increases by exactly one from one message to the next, but is not required to do so.
2. Please note that the maximum number of topics subscribed at a time is only 5.

### 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 <b>AT+QMTUNS=?</b>	Response <b>+QMTUNS:</b> (list of supported <b>&lt;client_idx&gt;</b> s),(list of supported <b>&lt;msgID&gt;</b> s),“ <b>&lt;topic&gt;</b> ”  <b>OK</b>
Write Command <b>AT+QMTUNS=&lt;client_idx&gt;,&lt;msgID&gt;,&lt;topic1&gt;[,&lt;topic2&gt;]...</b>	Response <b>OK</b>  <b>+QMTUNS:</b> <b>&lt;client_idx&gt;</b> , <b>&lt;msgID&gt;</b> , <b>&lt;result&gt;</b> [, <b>&lt;value&gt;</b> ]  If there is any error: <b>ERROR</b>

Maximum Response Time	<b>&lt;pkt_timeout&gt;</b> * <b>&lt;retry_times&gt;</b> (default 15s), determined by network
-----------------------	--

## Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. The range is 0-5.
<b>&lt;msgID&gt;</b>	Integer type. Message identifier of packet. The range is 1-65535.
<b>&lt;topic&gt;</b>	String type. Topic that the client wants to subscribe to or unsubscribe from.
<b>&lt;result&gt;</b>	Integer type. Result of the command execution 0 Sent packet successfully and received ACK from server 1 Packet retransmission 2 Failed to send packet
<b>&lt;value&gt;</b>	Integer type. If <b>&lt;result&gt;</b> is 0, it is a vector of granted QoS levels. If <b>&lt;result&gt;</b> is 1, it means the times of packet retransmission. If <b>&lt;result&gt;</b> is 2, it will not be presented.
<b>&lt;pkt_timeout&gt;</b>	Integer type. Timeout of the packet delivery. The range is 1-60. The default value is 5. Unit: second.
<b>&lt;retry_times&gt;</b>	Integer type. Retry times when packet delivery times out. The range is 0-10. The default value is 3.

### NOTE

At most 5 topics can be unsubscribed at a time in one **AT+QMTUNS** command.

## 3.2.8. AT+QMTPUB Publish Messages

The command is used to publish messages with fixed length or non-fixed length 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 <b>AT+QMTPUB=?</b>	Response <b>+QMTPUB:</b> (list of supported <b>&lt;client_idx&gt;</b> s),(list of supported <b>&lt;msgID&gt;</b> s),(list of supported <b>&lt;qos&gt;</b> s),(list of supported <b>&lt;retain&gt;</b> s)," <b>&lt;topic&gt;</b> "  <b>OK</b>
Write Command Edit mode <sup>1)</sup> , publish non-fixed-length messages	Response <b>OK</b>

<b>AT+QMTPUB=&lt;client_idx&gt;,&lt;msgID&gt;,&lt;qos&gt;,&lt;retain&gt;,"&lt;topic&gt;"</b> After ">" is responded, input the data to be sent, tap CTRL+Z to send, tap ESC to cancel the operation <sup>2)</sup>	<b>+QMTPUB: &lt;client_idx&gt;,&lt;msgID&gt;,&lt;result&gt;[,&lt;value&gt;]</b>  If there is any error: <b>ERROR</b>
Write Command Edit mode <sup>1)</sup> , publish fixed-length messages <b>AT+QMTPUB=&lt;client_idx&gt;,&lt;msgID&gt;,&lt;qos&gt;,&lt;retain&gt;,"&lt;topic&gt;",&lt;msg_length&gt;</b> After ">" is responded, input the data to be sent. When the actual size of data is larger than <msg_length>, only the first part of data within <msg_length> byte(s) will be sent out. <sup>3)</sup>	Response <b>OK</b>  <b>+QMTPUB: &lt;client_idx&gt;,&lt;msgID&gt;,&lt;result&gt;[,&lt;value&gt;]</b>  If there is any error: <b>ERROR</b>
Write Command Command mode <sup>1)</sup> , publish messages <b>AT+QMTPUB=&lt;client_idx&gt;,&lt;msgID&gt;,&lt;qos&gt;,&lt;retain&gt;,"&lt;topic&gt;",&lt;payload&gt;" <sup>4)</sup></b>	Response <b>OK</b>  <b>+QMTPUB: &lt;client_idx&gt;,&lt;msgID&gt;,&lt;result&gt;[,&lt;value&gt;]</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	<b>&lt;pkt_timeout&gt; * &lt;retry_times&gt;</b> (default 15s), determined by network

## Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. The range is 0-5.
<b>&lt;msgID&gt;</b>	Integer type. Message identifier of packet. The range is 0-65535. It will be 0 only when <qos>=0.
<b>&lt;qos&gt;</b>	Integer type. The QoS level at which the client wants to publish the messages. <u>0</u> At most once 1 At least once 2 Exactly once
<b>&lt;retain&gt;</b>	Integer type. Whether to enable the server to retain the message after it has been delivered to the current subscribers. <u>0</u> Disable the server to retain the message after it has been delivered to the current subscribers 1 Enable the server to retain the message after it has been delivered to the current subscribers
<b>&lt;topic&gt;</b>	String type. Topic that needs to be published
<b>&lt;msg_length&gt;</b>	Integer type. Length of message to be published.
<b>&lt;payload&gt;</b>	String type. Represents the message to be published

<b>&lt;result&gt;</b>	Integer type. Result of the command execution 0 Packet sent successfully and ACK received from server (message that published when <b>&lt;qos&gt;</b> =0 does not require ACK) 1 Packet retransmission 2 Failed to send packet
<b>&lt;value&gt;</b>	Integer type. If <b>&lt;result&gt;</b> is 1, it means the times of packet retransmission. If <b>&lt;result&gt;</b> is 0 or 2, it will not be presented.
<b>&lt;pkt_timeout&gt;</b>	Integer type, Timeout of the packet delivery. The range is 1-60. The default value is 5. Unit: second.
<b>&lt;retry_times&gt;</b>	Integer type, Retry times when packet delivery times out. The range is 0-10. The default value is 3.

## NOTES

- <sup>1)</sup> **AT+QMTPUB=<client\_idx>,<msgid>,<qos>,<retain>,"<topic>",<msg\_length>** refers to edit mode for fixed-length messages and if the **<msg\_length>** is omitted, it refers to edit mode for non-fixed-length messages; if the parameter **<msg\_length>** is changed to **<payload>**, it refers to command mode for the message to be published.
- <sup>2)</sup> The maximum length of the message to be published is 1548 bytes in this mode.
- <sup>3)</sup> The length range of the message to be published (**<msg\_length>**) is 1-1500 bytes in this mode.
- <sup>4)</sup> The maximum length of the command in this mode is 560 bytes.
- If this command is executed successfully and **OK** is responded, the client can continue to publish new packet. The maximum quantity of transmitting packet should not be greater than that of inflight windows (5); otherwise, **ERROR** will be returned.
- 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: <client\_idx>,<msgid>,"<topic>",<payload\_length>,"<payload>"**  
For more details about the URC description, please refer to **Chapter 4.2**.

### 3.2.9. AT+QMTRECV Read Messages from Buffers

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

#### AT+QMTRECV Read Messages from Buffers

Test Command <b>AT+QMTRECV=?</b>	Response <b>OK</b>
Read Command <b>AT+QMTRECV?</b>	Response <b>+QMTRECV: &lt;client_idx&gt;,&lt;store_status_0&gt;,&lt;store_status_1&gt;,&lt;store_status_2&gt;,&lt;store_status_3&gt;,&lt;store_status_4&gt;</b>

	<p><b>OK</b></p> <p>If there is no MQTT connection, response:</p> <p><b>OK</b></p>
<p>Write Command</p> <p><b>AT+QMTRECV=&lt;client_idx&gt;[,&lt;recv_id&gt;]</b></p>	<p>Response</p> <p>(List of <b>+QMTRECV: &lt;client_idx&gt;,&lt;msg_id&gt;,"&lt;topic&gt;",&lt;payload_len&gt;,"&lt;payload&gt;"</b>)</p> <p><b>OK</b></p> <p>If there is no message received, response:</p> <p><b>OK</b></p> <p>If there is no MQTT connection, response:</p> <p><b>ERROR</b></p>

## Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. The range is 0-5.
<b>&lt;store_status&gt;</b>	Integer type. Indicate whether there is a message stored in the buffer. 0 means no, and 1 means yes. The maximum quantity of message that can be stored in the buffer is 5. Therefore, URC reports maximally 5 messages simultaneously.
<b>&lt;recv_id&gt;</b>	Integer type. Indicate the serial number of every single message received. The range is 0-4.
<b>&lt;msg_id&gt;</b>	Integer type. Message identifier of packet. The range is 0-65535. It will be 0 only when <b>&lt;qos&gt;=0</b> .
<b>&lt;topic&gt;</b>	String type. Topic that needs to be published.
<b>&lt;payload_len&gt;</b>	Integer type. The length of payload.
<b>&lt;payload&gt;</b>	String type. The payload that relates to the topic name.

## 4 MQTT Related URCs

This chapter gives MQTT related URCs and their descriptions.

Table 2: MQTT Related URCs

SN	URC Format	Description
[1]	<b>+QMTSTAT: &lt;client_idx&gt;,&lt;err_code&gt;</b>	When the state of MQTT link layer is changed, the client will close the MQTT connection and report this URC.
[2]	<b>+QMTRECV: &lt;client_idx&gt;,&lt;msgID&gt;,"&lt;topic&gt;",&lt;payload_len&gt;,"&lt;payload&gt;"</b>	Reported when the client has received the packet data from MQTT server.
[3]	<b>+QMTRECV: &lt;client_idx&gt;,&lt;recv_id&gt;</b>	Reported when the message that had been received from MQTT server has been stored in buffer.

### 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

<b>+QMTSTAT: &lt;client_idx&gt;,&lt;err_code&gt;</b>	When the state of MQTT link layer is changed, the client will close the MQTT connection and report this URC.
--	--

#### Parameter

<b>&lt;client_idx&gt;</b>	Integer type. MQTT client identifier. The range is 0-5.
<b>&lt;err_code&gt;</b>	Integer type. An error code. Please refer to the table below for details.

Table 3: Error Codes of the URC

<err_code>	Description	How to do
1	Connection is closed or reset by	Execute <b>AT+QMTOPEN</b> command and reopen

	peer.	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.	<ol style="list-style-type: none"> <li>1. Check whether the inputted user name and password are correct.</li> <li>2. Make sure the client ID is not used.</li> <li>3. Reopen MQTT connection and try to send CONNECT packet to server again.</li> </ol>
4	Receiving CONNECK packet timed out or failed.	<ol style="list-style-type: none"> <li>1. Check whether the inputted user name and password are correct.</li> <li>2. Make sure the client ID is not used.</li> <li>3. Reopen MQTT connection and try to send CONNECT packet to server again.</li> </ol>
5-255	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: <client_idx>,<msgid>,<topic>,<payload_len>,<payload>	Notify the host to read the received data that is reported from MQTT server.
+QMTRECV: <client_idx>,<recv_id>	Reported when the message that received from MQTT server has been stored in buffer.

### Parameter

<client_idx>	Integer type. MQTT client identifier. The range is 0-5.
<msgid>	Integer type. The message identifier of packet
<topic>	String type. The topic that received from MQTT server
<payload_len>	Integer type. The length of payload.
<payload>	String type. The payload that relates to the topic name
<recv_id>	Integer type. Indicate the serial number of every single message received. The range is 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

```
//Configure receiving mode.
AT+QMTCFG="recv/mode",0,0
OK

//Configure Alibaba device information for Alibaba Cloud.
AT+QMTCFG="aliauth",0,"oyjtmPI5a5j","MQTT_TEST","wN9Y6pZSIly7Exa5qVzcmigEGO4kAazZ"
OK
AT+QMTOPEN=?
+QMTOPEN: (0-5),"hostname",(1-65536)

OK

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

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

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

OK
AT+QMTCONN=?
+QMTCONN: (0-5),"clientid","username","password"

OK

//Connect a client to MQTT server.
//If Alibaba Cloud is connected, customers can use AT+QMTCFG="aliauth" command to configure the
device information in advance, and do not need to provide username/password here anymore.
AT+QMTCONN=0,"clientExample"
```

OK

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

AT+QMTSUB=?

+QMTSUB: (0-5), <msgid>,list of ["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",36,"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>,(0-2),(0,1),"topic","length"

OK

//Publish messages. After receiving '>', input data "This is test data, hello MQTT." and then send it. The maximum length of the data is 1500 bytes and the data that beyond 1500 bytes will be omitted.

AT+QMTPUB=0,0,0,0,"topic/pub",30

> This is test data, 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",30,"This is test data, hello MQTT."**

//Disconnect a client from MQTT server.

**AT+QMTDISC=0**

**OK**

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

## 5.2. Example of MQTT Operation with SSL

//Configure receiving mode.

**AT+QMTCFG="recv/mode",0,0**

**OK**

//Configure MQTT session into SSL mode.

**AT+QMTCFG="ssl",0,1,2**

**OK**

//If SSL authentication mode is "server authentication", store CA certificate to RAM.

**AT+QFUPL="RAM: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 RAM.

**AT+QFUPL="RAM: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 RAM.

**AT+QFUPL="RAM:user\_key.pem",1679,100**

**CONNECT**

<Input the client.pem data, the size is 1679 bytes>

**+QFUPL: 1679,335f**

**OK**

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

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

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

//Configure SSL parameters.
AT+QSSLCFG="seclevel",2,2           //SSL authentication mode: server authentication
OK
AT+QSSLCFG="sslversion",2,4         //SSL authentication version
OK
AT+QSSLCFG="ciphersuite",2,0xFFFF //Cipher suite
OK
AT+QSSLCFG="ignorelocaltime",2,1    //Ignore the time of authentication.
OK

//Start MQTT SSL connection
AT+QMTOPEN=0,"a1zgnxur10j8ux.iot.us-east-1.amazonaws.com",8883
OK

+QMTOPEN: 0,0

//Connect to MQTT server
AT+QMTCONN=0,"M26_0206"
OK

+QMTCONN: 0,0,0

//Subscribe to topics.
AT+QMTSUB=0,1,"$aws/things/M26_0206/shadow/update/accepted",1
OK

+QMTSUB: 0,1,0,1

//Publish messages.
AT+QMTPUB=0,1,1,0,"$aws/things/M26_0206/shadow/update/accepted",32
>This is publish data from client
```

OK

**+QMTPUB: 0,1,0**

//If a client subscribes to a topic named "\$aws/things/M26\_0206/shadow/update/accepted" and other devices publish the same topic to the server, the module will report the following information.

**+QMTRECV: 0,1,"\$aws/things/M26\_0206/shadow/update/accepted",32,"This is publish data from client"**

//Disconnect a client from MQTT server.

**AT+QMTDISC=0**

OK

**+QMTDISC: 0,0**

## 6 Appendix A References

**Table 4: Related Documents**

SN	Document Name	Remarks
[1]	MQTT V3.1 Protocol Specification	MQTT protocol specification version 3.1
[2]	MQTT V3.1.1 Protocol Specification	MQTT protocol specification version 3.1.1

**Table 5: Terms and Abbreviations**

Abbreviation	Description
ACK	Acknowledgement
MQTT	Message Queuing Telemetry Transport
QoS	Quality of Service
RAM	Random Access Memory
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
URC	Unsolicited Result Code