BG95&BG77&BG600L Series

MQTT-SN Application Note

LPWA Module Series

Version: 1.0

Date: 2022-11-17

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
-	2021-10-13	Terrence YANG	Creation of the document
1.0	2022-11-17	Terrence YANG	First official release

Contents

Table Index

1 Introduction

MQTT-SN is a publish/subscribe messaging protocol for wireless sensor networks (WSNs), with the aim of extending the MQTT protocol beyond the reach of TCP/IP infrastructure for sensor and actuator solutions.

This document mainly introduces how to use the MQTT-SN function of Quectel BG95 series, BG77 and BG600L-M3 modules through AT commands.

2 MQTT-SN Data Interaction

This chapter gives the data interaction mechanism of MQTT-SN function.

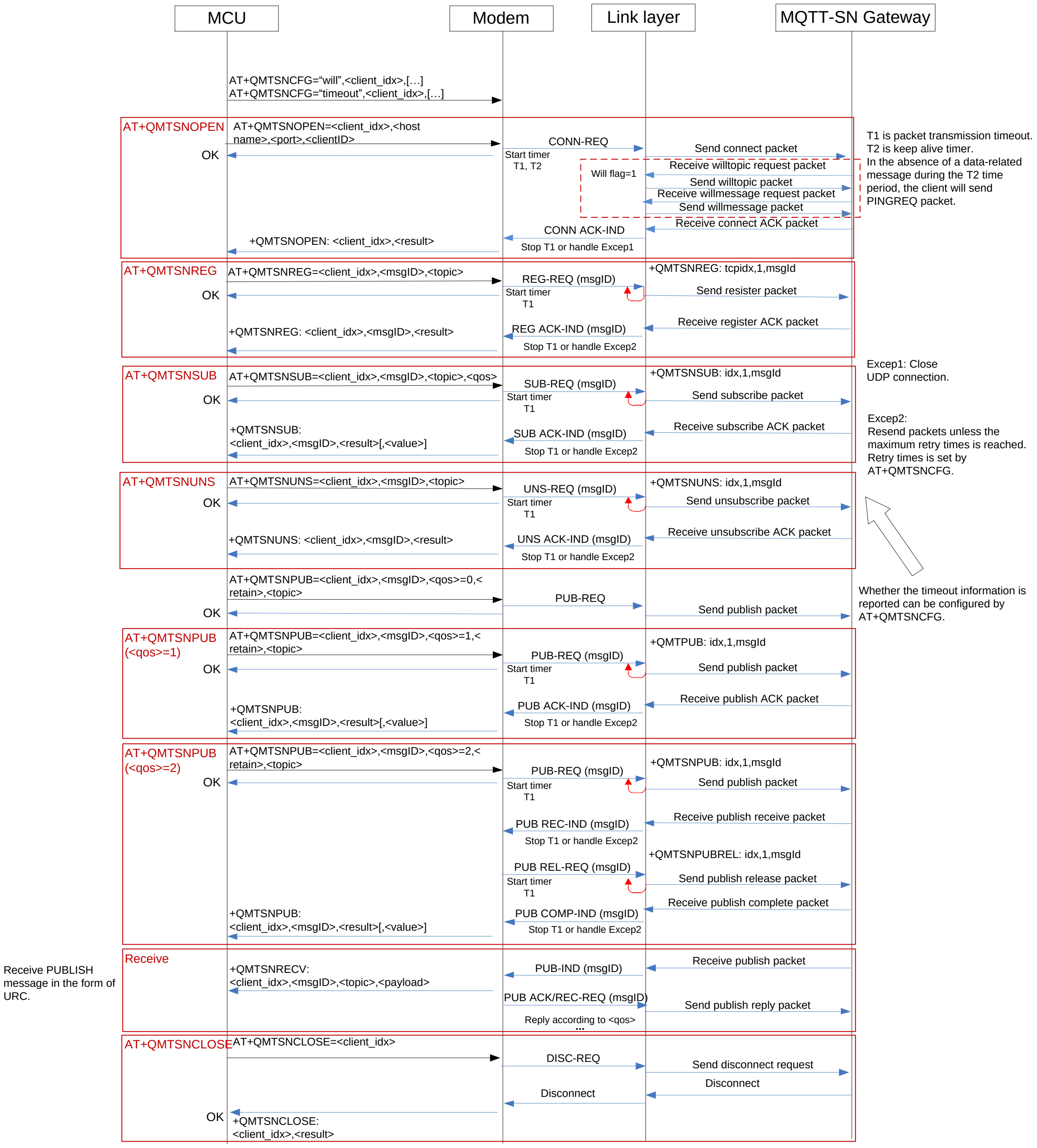
MCU
Modem
Link layer
MQTT-SN Gateway
AT+QMTSNCFG="will",<client_idx>,[…]
AT+QMTSNCFG="timeout",<client_idx>,[…]
AT+QMTSNOPEN
AT+QMTSNOPEN=<client_idx>,<host name>,<port>,<clientID>
OK
CONN-REQ
Start timer T1, T2
Send connect packet
Will flag=1
Receive willtopic request packet
Send willtopic packet
Receive willmessage request packet
Send willmessage packet
Receive connect ACK packet
CONN ACK-IND
Stop T1 or handle Excep1
+QMTSNOPEN: <client_idx>,<result>
T1 is packet transmission timeout.
T2 is keep alive timer.
In the absence of a data-related message during the T2 time period, the client will send PINGREQ packet.
AT+QMTSNREG
AT+QMTSNREG=<client_idx>,<msgID>,<topic>
OK
REG-REQ (msgID)
Start timer T1
+QMTSNREG: tcpidx,1,msgId
Send resister packet
Receive register ACK packet
REG ACK-IND (msgID)
Stop T1 or handle Excep2
+QMTSNREG: <client_idx>,<msgID>,<result>
Excep1: Close UDP connection.
AT+QMTSNSUB
AT+QMTSNSUB=<client_idx>,<msgID>,<topic>,<qos>
OK
SUB-REQ (msgID)
Start timer T1
+QMTSNSUB: idx,1,msgId
Send subscribe packet
Receive subscribe ACK packet
SUB ACK-IND (msgID)
Stop T1 or handle Excep2
+QMTSNSUB: <client_idx>,<msgID>,<result>[,<value>]
Excep2: Resend packets unless the maximum retry times is reached. Retry times is set by AT+QMTSNCFG.
AT+QMTSNUNS
AT+QMTSNUNS=<client_idx>,<msgID>,<topic>
OK
UNS-REQ (msgID)
Start timer T1
+QMTSNUNS: idx,1,msgId
Send unsubscribe packet
Receive unsubscribe ACK packet
UNS ACK-IND (msgID)
Stop T1 or handle Excep2
+QMTSNUNS: <client_idx>,<msgID>,<result>
AT+QMTSNPUB=<client_idx>,<msgID>,<qos>=0,<retain>,<topic>
OK
PUB-REQ
Send publish packet
Whether the timeout information is reported can be configured by AT+QMTSNCFG.
AT+QMTSNPUB (<qos>=1)
AT+QMTSNPUB=<client_idx>,<msgID>,<qos>=1,<retain>,<topic>
OK
PUB-REQ (msgID)
Start timer T1
+QMTPUB: idx,1,msgId
Send publish packet
Receive publish ACK packet
PUB ACK-IND (msgID)
Stop T1 or handle Excep2
+QMTSNPUB: <client_idx>,<msgID>,<result>[,<value>]
AT+QMTSNPUB (<qos>=2)
AT+QMTSNPUB=<client_idx>,<msgID>,<qos>=2,<retain>,<topic>
OK
PUB-REQ (msgID)
Start timer T1
+QMTSNPUB: idx,1,msgId
Send publish packet
Receive publish receive packet
PUB REC-IND (msgID)
Stop T1 or handle Excep2
+QMTSNPUBREL: idx,1,msgId
PUB REL-REQ (msgID)
Start timer T1
Send publish release packet
Receive publish complete packet
PUB COMP-IND (msgID)
Stop T1 or handle Excep2
+QMTSNPUB: <client_idx>,<msgID>,<result>[,<value>]
Receive PUBLISH message in the form of URC.
Receive
+QMTSNRECV: <client_idx>,<msgID>,<topic>,<payload>
PUB-IND (msgID)
Receive publish packet
PUB ACK/REC-REQ (msgID)
Reply according to <qos>
…
Send publish reply packet
AT+QMTSNCLOSE
AT+QMTSNCLOSE=<client_idx>
DISC-REQ
Send disconnect request
Disconnect
Disconnect
OK
+QMTSNCLOSE: <client_idx>,<result>

Figure 1: MQTT-SN Data Interaction Diagram

3 MQTT-SN Related AT Commands

This chapter presents the AT commands for operating MQTT-SN function.

3.1. AT Command Syntax

3.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on the command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on the command line. When an optional parameter is not given in a command, the new value equals 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-SN Related AT Commands

3.3.1. AT+QMTSNCFG Configure Optional Parameters of MQTT-SN

This command configures optional parameters of MQTT-SN.

AT+QMTSNCFG Configure Optional Parameters of MQTT-SN	
Test Command **AT+QMTSNCFG=?**	Response **+QMTSNCFG: "pdpcid",(**range of supported **<client_idx>**s**),(**range of supported **<cid>**s**)** **+QMTSNCFG: "keepalive",(**range of supported **<client_idx>**s**),(**range of supported **<keep_alive_time>**s**)** **+QMTSNCFG: "session",(**range of supported **<client_idx>**s**),(**list of supported **<clean_session>**s**)** **+QMTSNCFG: "timeout",(**range of supported **<client_idx>**s**),(**range of supported **<pkt_timeout>**s**),(**range of supported **<retry_times>**s**),(**list of supported **<timeout_notice>**s**)** **+QMTSNCFG: "dtls",(**range of supported **<client_idx>**s**),(**list of supported **<DTLS_enable>**s**),(**range of supported **<DTLS_contextID>**s**)** **+QMTSNCFG: "will",(**range of supported **<client_idx>**s**),(**list of supported**<will_fg>**s**),(**range of supported**<will_qos>**s**),(**list of supported **<will_retain>**s**),<will_topic>,<will_message>** **OK**
Write Command Query/set the PDP to be used by the MQTT-SN client **AT+QMTSNCFG="pdpcid",<client_idx>[,<cid>]**	Response If the optional parameter is omitted, query the PDP used by the MQTT-SN client: **+QMTSNCFG: "pdpcid",<cid>** **OK**

	If the optional parameter is specified, set the PDP to be used by the MQTT-SN client: **OK** If there is any error: **ERROR**
Write Command Query/set the keep-alive time **AT+QMTSNCFG="keepalive",<client_idx>[,<keep_alive_time>]**	Response If the optional parameter is omitted, query the keep-alive time: **+QMTSNCFG: "keepalive",<keep_alive_time>** **OK** If the optional parameter is specified, set the keep-alive time: **OK** If there is any error: **ERROR**
Write Command Query/set the session type **AT+QMTSNCFG="session",<client_idx>[,<clean_session>]**	Response If the optional parameter is omitted, query the session type: **+QMTSNCFG: "session",<clean_session>** **OK** If the optional parameter is specified, set the session type: **OK** If there is any error: **ERROR**
Write Command Query/set timeout of message delivery **AT+QMTSNCFG="timeout",<client_idx>[,<pkt_timeout>[,<retry_times>][,<timeout_notice>]]**	Response If the optional parameters are omitted, query the timeout of message delivery: **+QMTSNCFG: "timeout",<pkt_timeout>,<retry_times>,<timeout_notice>** **OK** If any of the optional parameters is specified, set the timeout of message delivery: **OK** If there is any error: **ERROR**
Write Command Query/set DTLS security mode of	Response If the optional parameters are omitted, query DTLS security

MQTT-SN client: **AT+QMTSNCFG="dtls",<client_idx>[,<DTLS_enable>,<DTLS_contextID>]**	mode of MQTT-SN client: **+QMTSNCFG: "dtls",<DTLS_enable>,<DTLS_contextID>** **OK** If the optional parameters are specified, set DTLS security mode for MQTT-SN client: **OK** If there is any error: **ERROR**
Write Command Query/set Will information **AT+QMTSNCFG="will",<client_idx>[,<will_fg>[,<will_qos>,<will_retain>,<will_topic>,<will_message>]]**	Response If the optional parameters are omitted, query the Will information: **+QMTSNCFG: "will",<will_fg>[,<will_qos>,<will_retain>,<will_topic>,<will_message>]** **OK** If any of the optional parameters is specified, set the Will information: **OK** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

<client_idx> Integer type. MQTT-SN socket identifier. Range: 0–5.

<cid> Integer type. The PDP to be used by the MQTT-SN client. Range: 1–16. Default value: 1.

<keep_alive_time> Integer type. Keep-alive time. Range: 0–3600. Default value: 120. Unit: second. It defines the maximum interval time for 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.

<clean_session> Integer type. Session type.
0 The server must store the subscriptions of the client after it is disconnected
1 The server must discard any previously maintained information about the client and treat the connection as "clean"

<pkt_timeout> Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.

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

<timeout_notice> Integer type. Whether to report timeout message when transmitting packet.
0 Not report
1 Report

<DTLS_enable> Integer type. MQTT-SN SSL mode.
0 Use normal UDP connection for MQTT-SN client
1 Use DTLS security connection for MQTT-SN client

<DTLS_contextID> Integer type. DTLS context identifier. Range: 0–5.

<will_fg> Integer type. Will flag configuration.
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. Will retain flag only used on 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 will retain the message after it has been delivered to the current subscribers

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

<will_message> String type. Define the content of the message that is published to the will topic if the client is unexpectedly disconnected. It can be zero-length message. The maximum length is 255 bytes.

NOTE

1. If **<will_fg>**=1, then **<will_qos>**, **<will_retain>**, **<will_topic>** and **<will_message>** must be specified. Otherwise, they will be omitted.
2. **<clean_session>**=0 is only effective when the server supports the operation.
3. It is crucial to ensure the message delivery does not time out while the message is still being sent.

3.3.2. AT+QMTSNOPEN Open a Network for MQTT-SN Client

This command opens a network and sends CONNECT message for MQTT-SN client.

AT+QMTSNOPEN Open a Network for MQTT-SN Client	
Test Command **AT+QMTSNOPEN=?**	Response **+QMTSNOPEN: (**range of supported **<client_idx>**s**),<host_name>,(**range of supported **<port>**s**),<clientID>** **OK**
Read Command **AT+QMTSNOPEN?**	Response **[+QMTSNOPEN: <client_idx>,<host_name>,<port>,<state>]** **OK**
Write Command **AT+QMTSNOPEN=<client_idx>,<host_name>,<port>,<clientID>**	Response **OK** **+QMTSNOPEN: <client_idx>,<result>** If there is any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** (5 s by default), determined by network
Characteristics	-

Parameter

<client_idx>	Integer type. MQTT-SN socket identifier. Range: 0–5.
<host_name>	String type. The address of the MQTT-SN gateway. It could be an IP address or a domain name. The maximum size is 255 bytes.
<port>	Integer type. The port of the MQTT-SN gateway. Range: 0–65535.
<clientID>	String type. The client identifier. The maximum size is 255 bytes.
<state>	Integer type. Current state of the specified MQTT-SN client. 0 Idle state 1 MQTT-SN client is connecting 2 MQTT-SN client is connected 3 MQTT-SN client is disconnecting
<result>	Integer type. Result of the command execution. -1 Failed to open network 0 Network opened successfully 1 Wrong parameter 2 MQTT-SN client identifier is occupied

	3 Failed to activate PDP 4 Failed to parse domain name 5 Network connection error
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.

3.3.3. AT+QMTSNCLOSE Close a Network for MQTT-SN Client

This command closes a network and sends DISCONNECT message for MQTT-SN client.

AT+QMTSNCLOSE Close a Network for MQTT-SN Client	
Test Command **AT+QMTSNCLOSE=?**	Response **+QMTSNCLOSE:** (range of supported **<client_idx>**s) **OK**
Write Command **AT+QMTSNCLOSE=<client_idx>**	Response **OK** **+QMTSNCLOSE: <client_idx>,<result>** If there is any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** (5 s by default), determined by network
Characteristics	-

Parameter

<client_idx>	Integer type. MQTT-SN socket identifier. Range: 0–5.
<result>	Integer type. Result of the command execution. -1 Failed to close network 0 Network closed successfully 2 Failed to send DISCONNECT packet
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.

3.3.4. AT+QMTSNSUB Subscribe to Topics

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

AT+QMTSNSUB Subscribe to Topics	
Test Command **AT+QMTSNSUB=?**	Response **+QMTSNSUB:** (range of supported **<client_idx>**s),(range of supported **<msgID>**s),**<topic>**,(range of supported **<qos>**s) **OK**
Write Command **AT+QMTSNSUB=<client_idx>,<msgID>,<topic>,<qos>**	Response **OK** **+QMTSNSUB: <client_idx>,<msgID>,<result>** If there is any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** × **<retry_times>** (15 s by default), determined by network
Characteristics	-

Parameter

<client_idx>	Integer type. MQTT-SN socket identifier. Range: 0–5.
<msgID>	Integer type. Message identifier of packet. Range: 1–65535.
<topic>	String type. The topic that the client wants to subscribe to. The maximum size is 255 bytes.
<qos>	Integer type. The QoS level at which the client wants to publish the messages. 0 At most once 1 At least once 2 Exactly once
<result>	Integer type. Result of the command execution. 0 Sent packet successfully and received ACK from server 1 Packet retransmission 2 Failed to send packet
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.

NOTE

The **<msgID>** is only present in messages where the QoS bits in the fixed header indicate QoS level 1 or 2. It must be unique amongst the set of "inflight" messages in a particular direction of communication. It typically increases by exactly one from one message to the next, but is not required to do so.

3.3.5. AT+QMTSNUNS Unsubscribe from Topics

This command unsubscribes from a named topic. An UNSUBSCRIBE message is sent by the client to the gateway to unsubscribe from a named topic.

AT+QMTSNUNS Unsubscribe from Topics	
Test Command **AT+QMTSNUNS=?**	Response **+QMTSNUNS: (**range of supported **<client_idx>**s**),(**range of supported **<msgID>**s**),<topic>** **OK**
Write Command **AT+QMTSNUNS=<client_idx>,<msgID>,<topic>**	Response **OK** **+QMTSNUNS: <client_idx>,<msgID>,<result>** If there is any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** × **<retry_times>** (15 s by default), determined by network
Characteristics	-

Parameter

<client_idx>	Integer type. MQTT-SN socket identifier. Range: 0–5.
<msgID>	Integer type. Message identifier of packet. Range: 1–65535.
<topic>	String type. The topic that the client wants to unsubscribe from. The maximum size is 255 bytes.
<result>	Integer type. Result of the command execution. 0 Sent packet successfully and received ACK from server 1 Packet retransmission 2 Failed to send packet
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.

3.3.6. AT+QMTSNREG Request Topic ID

This command requests a topic ID from the MQTT-SN gateway when sending a topic name.

AT+QMTSNREG Request Topic ID	
Test Command **AT+QMTSNREG=?**	Response **+QMTSNREG:** (range of supported **<client_idx>**s),(range of supported **<msgID>**s),**<topic>** **OK**
Read Command **AT+QMTSNREG?**	Response **[+QMTSNREG: <client_idx>,<topicID>,<topic>** **[...]]** **OK**
Write Command **AT+QMTSNREG=<client_idx>,<msgID>,<topic>**	Response **OK** **+QMTSNREG: <client_idx>,<msgID>,<result>** If there is any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** (5 s by default), determined by network
Characteristics	-

Parameter

<client_idx> Integer type. MQTT-SN socket identifier. Range: 0–5.

<msgID> Integer type. Message identifier of packet. Range: 0–65535.

<topic> String type. The topic that the client subscribed to. The maximum size is 255 bytes

<topicID> Integer type. The value of topic ID that shall be used in the PUBLISH message. Range: 0–65535.

<result> Integer type. Result of the command execution.
0 Sent packet successfully and received ACK from server
1 Packet retransmission
2 Failed to send packet

<pkt_timeout> Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.

3.3.7. AT+QMTSNPUB Publish Messages

This command publishes messages by a client to a server for distribution to interested subscribers. Each PUBLISH message is associated with a topic name. **AT+QMTSNREG** or **AT+QMTSNSUB** should be executed with associated topic name before the message can be published.

AT+QMTSNPUB Publish Messages	
Test Command **AT+QMTSNPUB=?**	Response **+QMTSNPUB:** (range of supported **<client_idx>**s),(range of supported **<msgID>**s),(range of supported **<qos>**s),(list of supported **<retain>**s),**<topic>**,(range of supported **<msglen>**s) **OK**
Write Command **AT+QMTSNPUB=<client_idx>,<msgID>,<qos>,<retain>,<topic>**	Response **>** After "**>**" is responded, input the data to be sent. Tap "**CTRL + Z**" to send, and tap "**ESC**" to cancel the operation. **OK** **+QMTSNPUB: <client_idx>,<msgID>,<result>[,<value>]** If there is any error: **ERROR**
Write Command **AT+QMTSNPUB=<client_idx>,<msgID>,<qos>,<retain>,<topic>,<msglen>**	Response **>** After "**>**" is responded, input the data to be sent. The number of bytes of inputted data must be equal to **<msglen>**. **OK** **+QMTSNPUB: <client_idx>,<msgID>,<result>[,<value>]** If there is any error: **ERROR**
Maximum Response Time	**<pkt_timeout>** × **<retry_times>** (15 s by default), determined by network
Characteristics	-

Parameter

<client_idx>	Integer type. MQTT-SN socket identifier. Range: 0–5.
<msgID>	Integer type. Message identifier of packet. Range: 0–65535. It will be 0 only when **<qos>**=0.

<qos>	Integer type. The QoS level at which the client wants to publish the messages. 0 At most once 1 At least once 2 Exactly once
<retain>	Integer type. Whether or not the server will retain the message after it has been delivered to the current subscribers. 0 Not retain 1 Retain
<topic>	String type. Topic that needs to be published. The maximum size is 255 bytes.
<msglen>	Integer type. Length of message to be published. Range: 1–1024. Unit: byte.
<result>	Integer type. Result of the command execution. 0 Sent packet successfully and received ACK from server (message that published when **<qos>**=0 does not require ACK) 1 Packet retransmission 2 Failed to send packet
<value>	If **<result>** is 1, it means the times of packet retransmission. If **<result>** is 0 or 2, it will not be presented.
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.
<retry_times>	Integer type. Retry times when packet delivery times out. Range: 0–10. Default value: 3.

NOTES

1. After executing this command, the client is ready to send data, which will be sent as payload. The maximum length of the input data is 1024 bytes at a time and tap **Ctrl + Z** to send the data.
2. 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-SN gateway: **+QMTSNRECV: <client_idx>,<msgID>,<topic>,<payload>**. For more details about the URC description, please refer to ***Chapter 4.2***.

3.3.8. AT+QMTSNWILLUPD Update Will Topic or Will Message

This command updates Will topic or Will message stored in gateway/server. If **<mode>** is 2, Will topic and Will message stored in gateway/server will be deleted.

AT+QMTSNWILLUPD Update Will Topic or Will Message	
Test Command **AT+QMTSNWILLUPD=?**	Response **+QMTSNWILLUPD: (**range of supported **<client_idx>**s)**,(**range of supported **<mode>**s)**,(**range of supported **<will_qos>**s)**,(**list of supported **<will_retain>**s)**,<will_topic>/<will_message>** **OK**
Write Command **AT+QMTSNWILLUPD=<client_idx>,<mode>,[<will_qos>,<will_retain>,<will_topic>/<will_message>]**	Response **OK** **+QMTSNWILLUPD: <client_idx>,<result>** If there is an error: **ERROR**
Maximum Response Time	**<pkt_timeout>** (5 s by default), determined by network
Characteristics	-

Parameter

<client_idx>	Integer type. MQTT-SN socket identifier. Range: 0–5.
<mode>	Integer type. Update or delete Will topic or/and Will message. 0 Update Will topic 1 Update Will message 2 Delete Will topic and Will message
<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 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>	String type. Will topic. The maximum length is 255 bytes.
<will_message>	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. The maximum length is 255 bytes.

<result> Integer type. Result of the command execution.
0 Sent packet successfully and received ACK from server
1 Packet retransmission
2 Failed to send packet
3 Request rejected from server

<pkt_timeout> Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.

3.3.9. AT+QMTSNSLEEP Configure the Current State of the Module

This command configures the current state of the module and notifies the gateway that the client enters asleep, awake, or active state.

AT+QMTSNSLEEP Configure the Current Status of the Module	
Test Command **AT+QMTSNSLEEP=?**	Response **+QMTSNSLEEP: (**range of supported **<client_idx>**s)**,(**range of supported **<sleep_mode>**s)**,(**range of supported **<duration>**s**)** **OK**
Write Command **AT+QMTSNSLEEP=<client_idx>,<sleep_mode>[,<duration>]**	Response **OK** **+QMTSNSLEEP: <client_idx>,<result>** If there is an error: **ERROR**
Maximum Response Time	**<pkt_timeout>** (5 s by default), determined by network
Characteristics	-

Parameter

<client_idx> Integer type. MQTT-SN socket identifier. Range: 0–5.

<sleep_mode> Integer type. The current state of the module.
0 Asleep state
1 Awake state
2 Active state

<duration> Integer type. Sleep duration. It is only valid when **<sleep_mode>** is 0. Range: 900–64800. Unit: second.

<result> Integer type. Result of the command execution.
0 Sent packet successfully and received ACK from server

	1 Packet retransmission 2 Failed to send packet
<pkt_timeout>	Integer type. Timeout of the packet delivery. Range: 1–60. Default value: 5. Unit: second.

4 MQTT-SN Related URCs

This chapter gives MQTT-SN related URCs and their descriptions.

Table 2: MQTT-SN Related URCs

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

4.1. +QMTSNSTAT URC to Indicate State Change in MQTT-SN Link Layer

The URC begins with **+QMTSNSTAT:**. It will be reported when there is a change in the state of MQTT-SN link layer.

+QMTSNSTAT URC to Indicate State Change in MQTT-SN Link Layer	
+QMTSNSTAT: <client_idx>,<err_code>	When the state of MQTT-SN link layer is changed, the client will close the MQTT-SN connection and report the URC.

Parameter

<client_idx>	Integer type. MQTT-SN socket identifier. Range: 0–5.
<err_code>	Error codes. Please refer to the table below for details.

Table 3: Error Codes of the URC

<err_code>	Description	How to do
1	The connection is closed or reset by a peer.	Execute **AT+QMTSNOPEN** and reopen MQTT-SN connection.
2	Sending PINGREQ packet timed out or failed.	Deactivate PDP first, and then active PDP and reopen MQTT-SN connection.
3	Sending CONNECT packet timed out or failed.	1. Check whether the inputted user name and password are correct. 2. Make sure the client ID is not used. 3. Reopen MQTT-SN 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-SN connection and try to send CONNECT packet to server again.
5	The client sends DISCONNECT packet to sever and the server is initiative to close MQTT-SN connection.	This is a normal process.
6	The client is initiative to close MQTT-SN connection due to packet sending failure all the time.	1. Make sure the data is correct. 2. Try to reopen MQTT-SN 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. +QMTSNRECV URC to Inform the Host to Read MQTT-SN Packet Data

The URC begins with **+QMTSNRECV:**. It is mainly used to inform the host to read the received MQTT-SN packet data that is reported from MQTT-SN gateway.

+QMTSNRECV URC to Inform the Host to Read MQTT-SN Packet Data	
+QMTSNRECV: <client_idx>,<msgID>,<topic>,<payload>	Inform the host to read the received data that is reported from MQTT-SN gateway.

Parameter

<client_idx>	Integer type. MQTT-SN socket identifier. Range: 0–5.
<msgID>	Integer type. The message identifier of packet. Range: 0–65535.
<topic>	String type. The topic that received from MQTT-SN gateway. The maximum size is 255 bytes.
<payload>	String type. The payload that relates to the topic name.

5 Example

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

5.1. Example of MQTT-SN Operation

```
//Open a network for MQTT-SN client.
AT+QMTSNOPEN=0,"220.180.239.212",8027,"clientid"
OK

+QMTSNOPEN: 0,0                         //Opened the MQTT-SN client network successfully.
AT+QMTSNOPEN?
+QMTSNOPEN: 0,"220.180.239.212",8027,2

OK
AT+QMTSNREG=0,1,"topic/example"
OK

+QMTSNREG: 0,1,0
AT+QMTSNREG=0,1,"topic/pub"
OK

+QMTSNREG: 0,1,0
AT+QMTSNSUB=?
+QMTSNSUB: (0-5),(1-65535),<topic>,(0-2)

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

+QMTSNSUB: 0,1,0
AT+QMTSNSUB=0,1,"topic/pub",0
OK

+QMTSNSUB: 0,1,0
```

```
//If a client subscribes to a topic and other devices publish message related to the same topic to the server,
the module reports the following information.
+QMTSNRECV: 0,0,"topic/example","This is the payload related to topic"

//Unsubscribe from topics.
AT+QMTSNUNS=0,2,"topic/example"
OK

+QMTSNUNS: 0,2,0
AT+QMTSNPUB=?
+QMTSNPUB: (0-5),(0-65535),(0-2),(0,1),<topic>,(1-1024)

OK
//Publish messages.
AT+QMTSNPUB=0,0,0,0,"topic/pub"
>This is test data, hello MQTT-SN. //After receiving >, input data This is test data, hello MQTT-SN. and
                                   then send it. The maximum length of the data is 1024 bytes and the
                                   data that beyond 1024 bytes will be omitted. After inputting data, tap
                                   Ctrl + Z to send.
OK

+QMTSNPUB: 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.
+QMTSNRECV: 0,0,"topic/pub","This is test data, hello MQTT-SN."

//Disconnect a client from MQTT-SN gateway.
AT+QMTSNCLOSE=0
OK

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

6 Appendix A References

Table 4: Terms and Abbreviations

Abbreviation	Description
ACK	Acknowledgement
DTLS	Datagram Transport Layer Security
IP	Internet Protocol
LPWA	Low Power Wide Area
MQTT	Message Queuing Telemetry Transport
MQTT-SN	MQTT For Sensor Networks
PDP	Packet Data Protocol
QoS	Quality of Service
RAM	Random Access Memory
SSL	Security Socket Layer
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URC	Unsolicited Result Code
WSN	Wireless Sensor Networks