

BG770S-GL&BG950S-GL

CoAP Application Note

LPWA Products

Version: 1.0

Date: 2025-12-01

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:

<https://www.quectel.com/contact/>.

For technical support, or to report documentation errors, please visit:

<https://www.quectel.com/tech-support/>.

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.

Copyright © Quectel Wireless Solutions Co., Ltd. 2025. All rights reserved.

About the Document

Revision History

Version	Date	Author	Description
-	2025-11-12	Aleksandar Petos	Creation of the document
1.0	2025-12-01	Aleksandar Petos	First official release

Contents

About the Document	3
Contents	4
Table Index	5
1 Introduction	6
2 General Overview of CoAP	7
3 Description of CoAP AT Commands	8
3.1. AT Command Introduction	8
3.1.1. Definitions	8
3.1.2. AT Command Syntax	8
3.2. Declaration of AT Command Examples	9
3.3. Description of CoAP AT Commands	9
3.3.1. AT+QCOAPCFG Configure Optional Parameters of CoAP Client	9
3.3.2. AT+QCOAPOPEN Create a CoAP Session	11
3.3.3. AT+QCOAPCLOSE Disconnect from CoAP Server	12
3.3.4. AT+QCOAPHEADER Configure CoAP Message Header	13
3.3.5. AT+QCOAPOPTION Configure CoAP Message Options	14
3.3.6. AT+QCOAPSEND Send CoAP Message	16
3.4. Description of CoAP URCs	19
3.4.1. +QCOAPRECV Incoming CoAP Message Indication	19
3.4.2. +QCOAPACK Delivery Result of CoAP Message Indication	19
3.4.3. +QCOAPIND Retransmission Result Notification	20
4 Summary of Result Codes	21
5 Examples	22
5.1. CoAP Client Operation Without DTLS	22
5.2. CoAP Client Operation with DTLS	23
5.3. CoAP Client Sending Data in Hexadecimal Format to the Server	24
6 Appendix References	26

Table Index

Table 1: Types of AT Commands	8
Table 2: Option Definitions	15
Table 3: <code_value> and <code> Definitions	17
Table 4: Description of <result> Codes.....	21
Table 5: Related Document	26
Table 6: Terms and Abbreviations	26

1 Introduction

This document introduces how to use the CoAP feature on Quectel BG770S-GL and BG950S-GL modules through AT commands.

2 General Overview of CoAP

The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks. The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation.

CoAP provides a request/response interaction model between application endpoints, supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs and Internet media types. CoAP is designed to easily interface with HTTP for integration with the Web while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments.

This chapter gives the data interaction mechanism of CoAP feature.

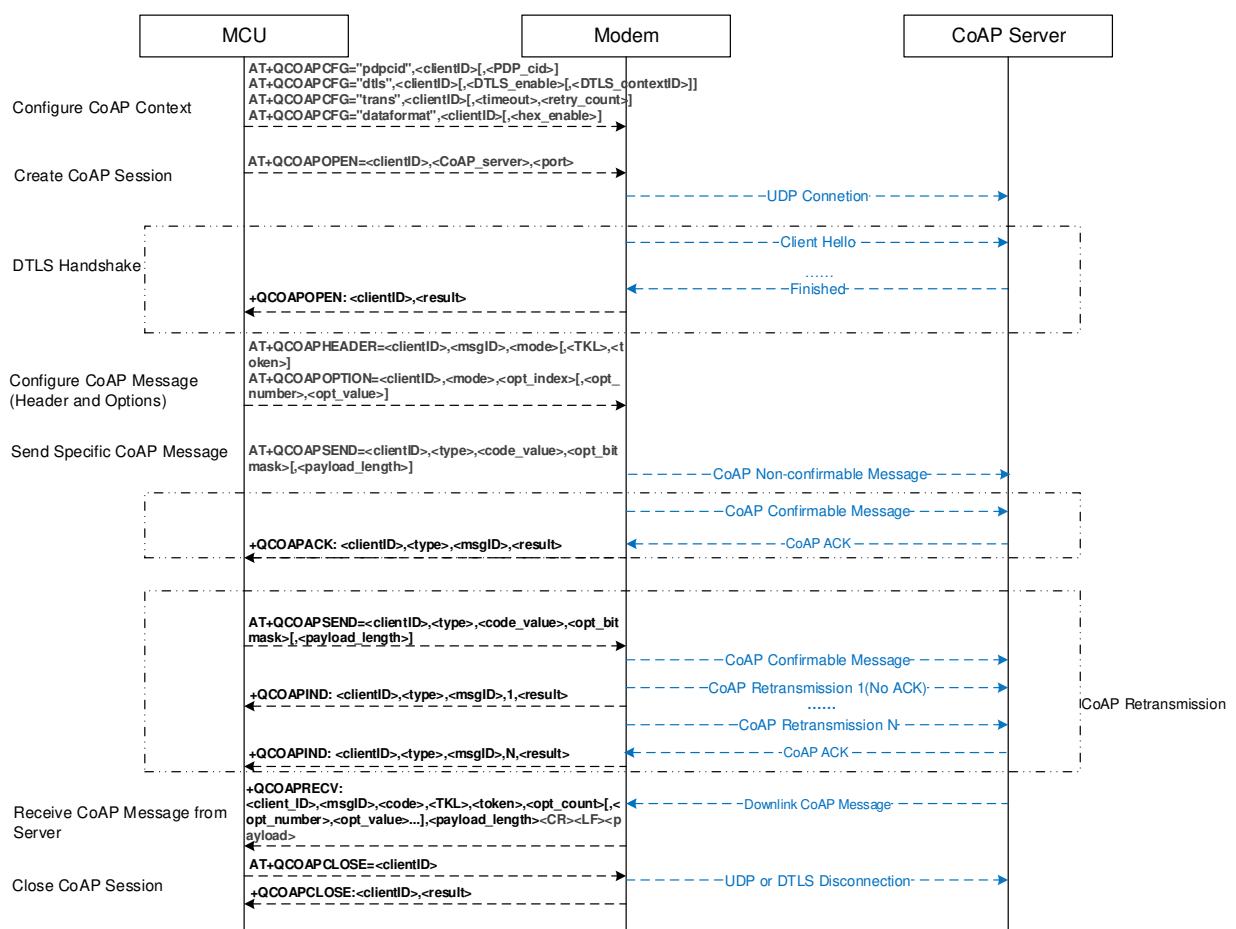


Figure 1: CoAP Data Interaction Diagram

3 Description of CoAP AT Commands

3.1. AT Command Introduction

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. The URLs, domain names, IP addresses, usernames/accounts, and passwords (if any) in the AT command examples are provided for illustrative and explanatory purposes only, and they should be modified to reflect your actual usage and specific needs.

3.3. Description of CoAP AT Commands

3.3.1. AT+QCOAPCFG Configure Optional Parameters of CoAP Client

This command configures optional parameters of a CoAP client.

AT+QCOAPCFG Configure Optional Parameters of CoAP Client	
Test Command AT+QCOAPCFG=?	Response +QCOAPCFG: "pdpcid", (list of supported <clientID>s),(list of supported <PDP_cid>s) +QCOAPCFG: "dtls", (list of supported <clientID>s),(list of supported <DTLS_enable>s),(list of supported <DTLS_contextID>s) +QCOAPCFG: "trans", (list of supported <clientID>s),(list of supported <timeout>s),(list of supported <retry_count>s) +QCOAPCFG: "dataformat", (list of supported <clientID>s),(list of supported <hex_enable>s) OK
Write Command Query/Set the PDP context for a specified CoAP client. AT+QCOAPCFG="pdpcid",<clientID>[,<PDP_cid>]	Response If the optional parameter is omitted, query the current setting. +QCOAPCFG: "pdpcid",<PDP_cid> OK If the optional parameter is specified, set the PDP context of the specified CoAP client. OK If there is any error:

	ERROR
Write Command Query/Set the DTLS mode for a specified CoAP client. AT+QCOAPCFG="dtls",<clientID>[,<DTLS_enable>[,<DTLS_contextID>]]	Response If the optional parameters are omitted, query the current setting. +QCOAPCFG: "dtls",<DTLS_enable>[,<DTLS_contextID>] OK If the optional parameters are specified, set the DTLS mode for the specified CoAP client. OK If there is any error: ERROR
Write Command Query/Set retransmission settings for a specified CoAP client. AT+QCOAPCFG="trans",<clientID>[,<timeout>,<retry_count>]	Response If the optional parameters are omitted, query the current setting. +QCOAPCFG: "trans",<timeout>,<retry_count> OK If the optional parameters are specified, set the retransmission settings for the specified CoAP client. OK If there is any error: ERROR
Write Command Query/Set format of sending data for a specified CoAP client. AT+QCOAPCFG="dataformat",<clientID>[,<hex_enable>]	Response If the optional parameter is omitted, query the current setting. +QCOAPCFG: "dataformat",<hex_enable> OK If the optional parameter is specified, set the format of sending data for the specified CoAP client. OK If there is any error: ERROR
Maximum Response Time	300 ms
Characteristics	The commands take effect immediately. The configurations are not saved.

Parameter

<clientID>	Integer type. CoAP client identifier. Range: 0–5.
<PDP_cid>	Integer type. PDP context ID used by CoAP client. Range: 1–2. Default value: 1.
<DTLS_enable>	Integer type. Whether to enable DTLS mode for CoAP client. 0 Use normal UDP connection for CoAP client 1 Use DTLS connection for CoAP client
<DTLS_contextID>	Integer type. DTLS context identifier. Range: 0–1.
<timeout>	Integer type. Acknowledgement timeout of CoAP confirmable message delivery. Range: 2–60. Default value: 2. Unit: second.
<retry_count>	Integer type. Maximum retransmission counts of CoAP confirmable message delivery. Range: 4–8. Default value: 5.
<hex_enable>	Integer type. Whether to enable sending data in hexadecimal format for CoAP client. 0 Enable sending data in normal text format 1 Enable sending data in hexadecimal format

NOTE

If DTLS mode is enabled for a CoAP session, the PSK file named <DTLS_contextID>_server.psk should be uploaded to UFS or EUFS with **AT+QFUP** (see **document [1]** for details), and the content of the file should be in the format of "<PSK identifier>&<PSK key>". CoAP client uses this PSK file for establishing DTLS session.

3.3.2. AT+QCOAPOPEN Create a CoAP Session

This command creates a CoAP session.

AT+QCOAPOPEN Create a CoAP Session	
Test Command AT+QCOAPOPEN=?	Response +QCOAPOPEN: (list of supported <clientID>s),<CoAP_server>,(list of supported <port>s) OK
Read Command AT+QCOAPOPEN?	Response [+QCOAPOPEN: <clientID>,<CoAP_server>,<port>,<status>] OK
Write Command Configure and connect to a specified CoAP server: AT+QCOAPOPEN=<clientID>,<CoAP	Response OK +QCOAPOPEN: <clientID>,<result>

_server>,<port>	If there is any error: ERROR
Maximum Response Time	75 s, determined by network
Characteristics	-

Parameter

<clientID>	Integer type. CoAP client identifier. Range: 0–5.
<CoAP_server>	String type. Address of CoAP server. It can be an IP address or a domain name. Maximum size: 255 bytes.
<port>	Integer type. Port of CoAP server. Range: 1–65535.
<status>	Integer type. Current status of the specified CoAP client. 0 Idle state or connection disconnected. 1 CoAP client is opening. 2 CoAP client is connecting to the CoAP server. 3 CoAP client is connected. 4 CoAP connection is disconnecting.
<result>	Integer type. Result of the command execution. See Chapter 4 for details.

3.3.3. AT+QCOAPCLOSE Disconnect from CoAP Server

This command disconnects a client from the CoAP server.

AT+QCOAPCLOSE Disconnect from CoAP Server	
Test Command AT+QCOAPCLOSE=?	Response +QCOAPCLOSE: (list of supported <clientID>s) OK
Write Command AT+QCOAPCLOSE=<clientID>	Response OK +QCOAPCLOSE: <clientID>,<result> If there is any error: ERROR
Maximum Response Time	75 s, determined by network
Characteristics	-

Parameter

<clientID>	Integer type. CoAP client identifier. Range: 0–5.
<result>	Integer type. Result of the command execution. See Chapter 4 for details.

3.3.4. AT+QCOAPHEADER Configure CoAP Message Header

This command configures the header of a CoAP message.

AT+QCOAPHEADER Configure CoAP Message Header	
Test Command AT+QCOAPHEADER=?	Response +QCOAPHEADER: (list of supported <clientID> s),(list of supported <msgID> s),(list of supported <mode> s),(list of supported <TKL> s), <token> OK
Write Command AT+QCOAPHEADER=<clientID>,<msgID>,<mode>[,<TKL>,<token>]	Response OK If there is any error: ERROR
Maximum Response Time	300 ms
Characteristics	-

Parameter

<clientID>	Integer type. CoAP client identifier. Range: 0–5.
<msgID>	16-bit unsigned integer in network byte order. Message ID. Used to detect message duplication and to match Acknowledgement/Reset messages to Confirmable/Non-confirmable messages. Range: 0–65535.
<mode>	Integer type. Whether to generate token value automatically. 0 Do not generate token value automatically. 1 Generate token value automatically.
<TKL>	4-bit unsigned integer. Length of the variable-length <token> field. Range: 0–8. Unit: bytes. Only valid when <mode> =0.
<token>	Hex string type. Token value of CoAP message. Only valid when <mode> =0.

3.3.5. AT+QCOAPOPTION Configure CoAP Message Options

This command configures the options of a CoAP message.

AT+QCOAPOPTION Configure CoAP Message Options	
Test Command AT+QCOAPOPTION=?	Response +QCOAPOPTION: (list of supported <clientID>s),(list of supported <mode>s),(list of supported <opt_index>s),<opt_number>,<opt_value> OK
Write Command AT+QCOAPOPTION=<clientID>,<mode>,<opt_index>[,<opt_number>,<opt_value>]	Response OK If there is any error: ERROR
Maximum Response Time	300 ms
Characteristics	-

Parameter

<clientID>	Integer type. CoAP client identifier. Range: 0–5.
<mode>	Integer type. Operation mode of this command. 0 Add a new option for CoAP message 1 Remove an existing option from CoAP message
<opt_index>	Integer type. Index of the option to be added/deleted. Range: 0–7.
<opt_number>	Integer type. Option number. Only valid when <mode>=0. The individual CoAP options are summarized and explained in <i>RFC 7252 section 5.10</i> .
Option Number Option Name	
1	If-Match
3	Uri-Host
4	ETag
5	If-None-Match
7	Uri-Port
8	Location-Path
11	Uri-Path
12	Content-Format
14	Max-Age
15	Uri-Query
17	Accept
20	Location-Query
35	Proxy-Uri

	39	Proxy-Scheme
	60	Size1
<opt_value>	Option value that corresponds to each option number. Only valid when <mode>=0. See Table 2 for details such as format and length of this parameter.	

Table 2: Option Definitions

Option Number	Option Name	Option Value Format	Option Value Length (Bytes)
1	If-Match	Opaque	0–8
3	Uri-Host	String	1–255
4	ETag	Opaque	1–8
5	If-None-Match	Empty	0
7	Uri-Port	Unsigned integer	0–2
8	Location-Path	String	0–255
11	Uri-Path	String	0–255
12	Content-Format	Unsigned integer	0–2
14	Max-Age	Unsigned integer	0–4
15	Uri-Query	String	0–255
17	Accept	Unsigned integer	0–2
20	Location-Query	String	0–255
35	Proxy-Uri	String	1–1034
39	Proxy-Scheme	String	1–255
60	Size1	Unsigned integer	0–4

3.3.6. AT+QCOAPSEND Send CoAP Message

This command sends a CoAP message. After you input the payload with a specified length, the command first serializes the input data to a CoAP packet and then sends it to the CoAP server.

AT+QCOAPSEND Send CoAP Message	
Test Command AT+QCOAPSEND=?	Response +QCOAPSEND: (list of supported <clientID>s),(list of supported <type>s),<code_value>,<opt_bitmask>,(list of supported <payload_length>s) OK
Write Command AT+QCOAPSEND=<clientID>,<type>,<code_value>,<opt_bitmask>[,<payload_length>]	Response > After > is returned, input the data to be sent. Tap Ctrl + Z to send the data, or tap Esc to cancel the operation. OK +QCOAPACK: <cliendID>,<type>,<msgID>,<result> If there is any error: ERROR
Maximum Response Time	300 ms
Characteristics	-

Parameter

<clientID>	Integer type. CoAP client identifier. Range: 0–5.		
<type>	Integer type. CoAP message type.		
	0	Confirmable (CON)	
	1	Non-confirmable (NON)	
	2	Acknowledgement	
	3	Reset	
<code_value>	Integer type. Request or response code value. See Table 3 for details.		
<opt_bitmask>	Integer type. Current options can be preset in CoAP client with AT+QCOAPOPTION . If any bit in the bitmask of this parameter is set to 1, the corresponding option will be added to a CoAP packet.		
	Value	Bitmask	Description
	0	00000000	No option is added to the CoAP packet
	1	00000001	Add the option with index 0 to the CoAP packet
	2	00000010	Add the option with index 1 to the CoAP packet
	3	00000011	Add options with indexes 0 and 1 to the CoAP packet

	4	00000100	Add the option with index 3 to the CoAP packet
	5	00000101	Add options with indexes 0 and 3 to the CoAP packet
	...		
	253	11111101	Add options with indexes 0 and 2–7 to the CoAP packet
	254	11111110	Add options with indexes 1–7 to the CoAP packet
	255	11111111	Add options with indexes 0–7 to the CoAP packet
<payload_length>	Integer type. Length of data to be sent. Maximum length: 1024 bytes. If this parameter is omitted, data of any length not exceeding 1024 bytes can be input.		
<msgID>	16-bit unsigned integer in network byte order. Message ID. Used to detect message duplication and to match Acknowledgement/Reset messages to Confirmable/Non-confirmable messages. Range: 0–65535.		
<result>	Integer type. Result of the command execution. See Chapter 4 for details.		

Table 3: <code_value> and <code> Definitions

<code_value>	<code>	Description
0	0.00	Empty message.
1	0.01	GET. The GET method retrieves a representation for the information that currently corresponds to the resource identified by the request URI.
2	0.02	POST. The POST method requests that the representation enclosed in the request be processed.
3	0.03	PUT. The PUT method requests that the resource identified by the request URI be updated or created with the enclosed representation.
4	0.04	DELETE. The DELETE method requests that the resource identified by the request URI be deleted.
65	2.01	Created. Like HTTP 201 "Created", but only used in response to POST and PUT requests. The payload returned with the response, if any, is a representation of the action result.
66	2.02	Deleted. This Response Code is like HTTP 204 "No Content" but only used in response to requests that cause the resource to cease being available, such as DELETE and, in certain circumstances, POST. The payload returned with the response, if any, is a representation of the action result.
67	2.03	Valid. This Response Code is related to HTTP 304 "Not Modified" but only used to indicate that the response identified by the entity-tag identified by the included ETag Option is valid.
68	2.04	Changed. This Response Code is like HTTP 204 "No Content" but only used in response to POST and PUT requests. The payload returned with the response, if any, is a representation of the action result.
69	2.05	Content. This Response Code is like HTTP 200 "OK" but only used in response to GET requests.

128	4.00	Bad request.
129	4.01	Unauthorized. The client is not authorized to perform the requested action.
130	4.02	Bad option. The request could not be understood by the server due to one or more unrecognized or malformed options.
131	4.03	Forbidden.
132	4.04	Not found.
133	4.05	Method not allowed.
134	4.06	Not acceptable.
140	4.12	Precondition failed.
141	4.13	Request entity too large.
143	4.15	Unsupported Content-Format.
160	5.00	Internal server error.
161	5.01	Not implemented.
162	5.02	Bad gateway.
163	5.03	Service unavailable.
164	5.04	Gateway timeout.
165	5.05	Proxying not supported. The server is unable or unwilling to act as a forward-proxy for the URI specified in the Proxy-Uri Option or using Proxy-Scheme.

NOTE

1. **<code>** is an 8-bit unsigned integer, split into a 3-bit class (most significant bits) and a 5-bit detail (least significant bits), documented as "c.dd" where "c" is a digit from 0 to 7 for the 3-bit subfield and "dd" are two digits from 00 to 31 for the 5-bit subfield. The class can indicate a request (0), a success response (2), a client error response (4), or a server error response (5). (All other class values are reserved.) As a special case, Code 0.00 indicates an Empty message.
2. When **<code>** is *c.dd*, **<code_value>** = $c \times 32 + dd$.

3.4. Description of CoAP URCs

CoAP URCs are reported to the host when a CoAP Client is in registration, observation, or application data transmission procedure.

3.4.1. +QCOAPRECV Incoming CoAP Message Indication

This URC is reported when CoAP client receives a downlink CoAP message from the remote CoAP server.

+QCOAPRECV Incoming CoAP Message Indication

+QCOAPRECV: <clientID>,<msgID>,<code>,<TKL>,<token>,<opt_count>[,<opt_number>,<opt_value>...],<payload_length><CR><LF><payload>	This URC is reported when a new CoAP message is received from CoAP server.
--	--

Parameter

<clientID>	Integer type. CoAP client identifier. Range: 0–5.
<msgID>	16-bit unsigned integer in network byte order. Message ID.
<code>	Response code of incoming message. Format: “c.dd”. See Table 3 for details.
<TKL>	Integer type. Length of a variable-length token field.
<token>	Token value of a CoAP message.
<opt_count>	Number of options included in the CoAP message.
<opt_number>	Option number.
<opt_value>	Option value that corresponds to the option number.
<payload_length>	Payload length of incoming CoAP message.
<payload>	Payload data.

3.4.2. +QCOAPACK Delivery Result of CoAP Message Indication

If a CoAP message is sent, the client needs an acknowledgement message from the server.

- If a confirmable message is delivered, this URC indicates if the message has been acknowledged by the server.
- If a non-confirmable, acknowledgement or reset message is delivered, this URC indicates if the message has been sent.

+QCOAPACK Delivery Result of CoAP Message Indication

+QCOAPACK: <clientID>,<type>,<msgID>,<result> This URC is reported to indicate if a CoAP message has been sent or acknowledged by the server.

Parameter

<clientID>	Integer type. CoAP client identifier. Range: 0–5.
<type>	Integer type. CoAP message type. 0 Confirmable (CON) 1 Non-confirmable (NON) 2 Acknowledgement 3 Reset
<msgID>	16-bit unsigned integer in network byte order. Message ID.
<result>	Integer type. Result of CoAP message delivery. See Chapter 4 for details.

3.4.3. +QCOAPIND Retransmission Result Notification

This URC is reported to notify the retransmission result when a client retransmits a confirmable message that is not acknowledged by the server.

+QCOAPIND Retransmission Result Notification

+QCOAPIND: <clientID>,<type>,<msgID>,<retry_times>,<result> This URC is reported to notify the retransmission status of a CoAP confirmable message.

Parameter

<clientID>	Integer type. CoAP client identifier. Range: 0–5.
<type>	Integer type. CoAP message type. 0 Confirmable
<msgID>	16-bit unsigned integer in network byte order. Message ID.
<retry_times>	Integer type. Retransmission count after sending a confirmable message. The maximum retransmission count is configured with AT+QCOAPCFG="trans" .
<result>	Integer type. Retransmission result. 1 The module has retransmitted the confirmable message. 2 The number of retransmissions has reached the maximum retransmission count.

4 Summary of Result Codes

The following table lists some of the general result codes.

Table 4: Description of <result> Codes

<result>	Meaning
0	Operation successful
-1	Invalid parameter
-2	Operation in processing
-3	Operation not allowed
-4	Network failure
-5	DNS error
-6	Data call activating
-7	Socket connection failure
-8	Out of memory error
-9	DTLS handshaking failure
-10	CoAP client identifier occupied
-11	Data sending failure

5 Examples

5.1. CoAP Client Operation Without DTLS

```

AT+QCOAPCFG="pdpctxid",0,1 //Set the PDP context ID to 1 for CoAP client 0.
OK
AT+QCOAPCFG="trans",0,4,5 //Configure retransmission settings for CoAP client 0. (The ACK
                           timeout is 4 seconds and the maximum retransmission count is 5.)
OK
AT+QCOAPOPEN=0,"192.0.2.2",8028 //Create a CoAP session and connect to the CoAP server.
OK

+QCOAPOPEN: 0,0 //CoAP session created successfully.
AT+QCOAPOPEN? //Query the current status of the CoAP session.
+QCOAPOPEN: 0,"192.0.2.2",8028,3

OK

//Set CoAP message header.
AT+QCOAPHEADER=0,1234,1 //Set CoAP message ID to 1234 and generate token automatically.
OK

//Add CoAP options.
AT+QCOAPOPTION=0,0,11,"19" //Add an option (option number: 11, corresponding to option name of
                           Uri-Path; option value: "19") to option index 0.
OK
AT+QCOAPOPTION=0,0,1,12,40 //Add an option (option number: 12, corresponding to option name of
                           Content-Format; option value: 40, corresponding to media type of
                           application/link-format) to option index 1.
OK
AT+QCOAPSEND=0,1,2,1,20 //Send a 20-byte CoAP non-confirmable message to the server.
>Hello, CoAP Message!
OK

+COAPACK: 0,1,1234,0

//Receive downlink error message from server.
+QCOAPRECV: 0,1234,4.04,6,D204D52E814B,0,9

```

Not Found

AT+QCOAPCLOSE=0 //Close the CoAP session.

OK

+QCOAPCLOSE: 0,0 //CoAP session closed successfully.

5.2. CoAP Client Operation with DTLS

AT+QCOAPCFG="pdpctxid",0,1 //Set the PDP context ID to 1 for CoAP client 0.

OK

AT+QCOAPCFG="trans",0,4,5 //Configure retransmission settings for CoAP client 0. (ACK timeout is 4 s. Maximum retransmission count is 5.)

OK

AT+QCOAPCFG="dtls",0,1,0 //Enable DTLS mode for CoAP client 0.

OK

AT+QSSLCFG="dtls",0,1 //Set DTLS mode (enable DTLS feature).

OK

AT+QFUPL="UFS:0_server.psk",34 //Upload the PSK file for DTLS session to UFS.

CONNECT

864508030012428&313233343446473839 //Input the content of the PSK file.

+QFUPL: 34,2802

OK

AT+QCOAPOPEN=0,"lshan.eclipseprojects.io",5684 //Create a CoAP session and connect to the CoAP server.

OK

+QCOAPOPEN: 0,0 //CoAP session created successfully.

//Set CoAP message header.

AT+QCOAPHEADER=0,1234,1 //Set CoAP message ID to 1234 and generate token automatically.

OK

//Add CoAP options.

AT+QCOAPOPTION=0,0,0,11,"19" //Add an option (option number: 11, corresponding to option name of Uri-Path; option value: "19") to option index 0.

OK

AT+QCOAPOPTION=0,0,1,12,40 //Add an option (option number: 12, corresponding to option name of Content-Format; option value: 40, corresponding to media type of application/link-format) to option index 1.

OK

AT+QCOAPSEND=0,1,2,1,20 //Send a 20-byte CoAP non-confirmable message to the server.

>Hello, CoAP Message!

```

OK

+COAPACK: 0,1,1234,0

//Receive downlink error message from server.
+QCOAPRECV: 0,1234,4.04,6,D204D52E814B,0,9
Not Found
AT+QCOAPCLOSE=0           //Close the CoAP session.
OK

+QCOAPCLOSE: 0,0          //CoAP session closed successfully.

```

NOTE

See [document \[2\]](#) for details of **AT+QSSLCFG**.

5.3. CoAP Client Sending Data in Hexadecimal Format to the Server

```

AT+QCOAPCFG="pdpcid",0,1 //Set the PDP context ID to 1 for CoAP client 0.
OK
AT+QCOAPCFG="trans",0,4,5 //Configure retransmission settings for CoAP client 0. (The ACK
                           //timeout is 4 seconds and the maximum retransmission count is
                           //5.)
OK
AT+QCOAPOPEN=0,"leshan.eclipseprojects.io",5683 //Create a CoAP session and connect to the
                                                //CoAP server.
OK
+QCOAPOPEN: 0,0 //CoAP session created successfully.
AT+QCOAPHEADER=0,1235,0,4,"1713b697" //Set CoAP message header.
OK
//Add CoAP options.
AT+QCOAPOPTION=0,0,0,11,"rd" //Add an option (option number: 11, corresponding to option name
                             //of Uri-Path; option value: "rd") to option index 0.
OK
AT+QCOAPOPTION=0,0,1,12,40 //Add an option (option number: 12, corresponding to option name
                             //of Content-Format; option value: 40, corresponding to media type
                             //of application/link-format) to option index 1.
OK
AT+QCOAPOPTION=0,0,2,15,"ep=urn:imei:868494060001388" //Add an option (option number: 15,
                                                         //corresponding to option name of

```

```

Uri-Query;      option      value:
"ep=urn:imei:868494060001388")
to option index 2.

OK
AT+QCOAPOPTION=0,0,3,15,"lwm2m=1.0" //Add an option (option number: 15, corresponding to
option name of Uri-Query; option value: "lwm2m=1.0")
to option index 3.

OK
AT+QCOAPOPTION=0,0,4,15,"b=UQS" //Add an option (option number: 15, corresponding to
option name of Uri-Query; option value: "b=UQS") to
option index 4.

OK
AT+QCOAPOPTION=0,0,5,15,"lt=60" //Add an option (option number: 15, corresponding to
option name of Uri-Query; option value: "lt=60") to
option index 5.

OK
AT+QCOAPCFG="dataformat",0,1 //Enable sending data in hexadecimal format.

OK
AT+QCOAPSEND=0,0,2,63,47 //Send CoAP message.
>3c2f3e3b72743d226f6d612e6c776d326d223b63743d313534332c3c2f312f303e2c3c2f332f303e2c3c
2f342f303e
OK

+QCOAPACK: 0,2,1235,0

+QCOAPRECV: 0,1235,2.01,4,1713B697,2,8,"rd",8,"1fszHLnn8g",0

```

6 Appendix References

Table 5: Related Documents

Document Name
[1] Quectel_BG770S-GL&BG950S-GL_FILE_Application_Note
[2] Quectel_BG770S-GL&BG950S-GL_SSL_Application_Note

Table 6: Terms and Abbreviations

Abbreviation	Description
ACK	Acknowledgement
CoAP	Constrained Application Protocol
DTLS	Datagram Transport Layer Security
HTTP(S)	Hypertext Transfer Protocol (Secure)
ID	Identifier
LPWA	Low-Power Wide-Area
M2M	Machine to Machine
MCU	Microcontroller Unit
PDP	Packet Data Protocol
PSK	Pre-Shared Key
UDP	User Datagram Protocol
UFS	User File System
URC	Unsolicited Result Code
URI	Uniform Resource Identifier
TA	Terminal Adapter