

BG96

CoAP Application Note

LPWA Module Series

Version: 1.0

Date: 2021-03-04

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.

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 office. For more information, please visit:

<http://www.quectel.com/support/sales.htm>.

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

<http://www.quectel.com/support/technical.htm>

Or email to support@quectel.com.

General Notes

Quectel offers the information as a service to its customers. The information provided is based upon customers' requirements. Quectel makes every effort to ensure the quality of the information it makes available. Quectel does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information. All information supplied herein is subject to change without prior notice.

Disclaimer

While Quectel has made efforts to ensure that the functions and features under development are free from errors, it is possible that these functions and features could contain errors, inaccuracies and omissions. Unless otherwise provided by valid agreement, Quectel makes no warranties of any kind, implied or express, with respect to the use of features and functions under development. To the maximum extent permitted by law, Quectel excludes all liability for any loss or damage suffered in connection with the use of the functions and features under development, regardless of whether such loss or damage may have been foreseeable.

Duty of Confidentiality

The Receiving Party shall keep confidential all documentation and information provided by Quectel, except when the specific permission has been granted by Quectel. The Receiving Party shall not access or use Quectel's documentation and information for any purpose except as expressly provided herein. Furthermore, the Receiving Party shall not disclose any of the Quectel's documentation and information to any third party without the prior written consent by Quectel. For any noncompliance to the above requirements, unauthorized use, or other illegal or malicious use of the documentation and information, Quectel will reserve the right to take legal action.

Copyright

The information contained here is proprietary technical information of Quectel. Transmitting, reproducing, disseminating and editing this document as well as using the content without permission are forbidden. 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. 2021. All rights reserved.

About the Document

Revision History

| Version | Date | Author | Description |
|---------|------------|---------------------------------|--------------------------|
| - | 2020-12-28 | Sherlock ZHAO/ Terrence YANG | Creation of the document |
| 1.0 | 2021-03-04 | Sherlock ZHAO/ Terrence YANG | 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 Syntax..... | 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..... | 12 |
| 3.3.5. AT+QCOAPOPTION Configure CoAP Message Options | 13 |
| 3.3.6. AT+QCOAPSEND Send CoAP Message | 15 |
| 3.4. Description of CoAP URCS | 18 |
| 3.4.1. +QCOAPRECV Indicating Incoming CoAP Message..... | 18 |
| 3.4.2. +QCOAPACK Indicating Delivery Result of CoAP Message..... | 19 |
| 3.4.3. +QCOAPIND Notifying Retransmission Result..... | 19 |
| 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 |
| 6 Appendix A References..... | 25 |

Table Index

| | |
|--|----|
| Table 1: Types of AT Commands | 8 |
| Table 2: Option Definitions | 14 |
| Table 3: <code_value> and <code> Definitions | 16 |
| Table 4: Description of <result> Codes..... | 21 |
| Table 5: Related Documents..... | 25 |
| Table 6: Terms and Abbreviations | 25 |

1 Introduction

This document introduces how to use the CoAP feature on Quectel BG96 module 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.

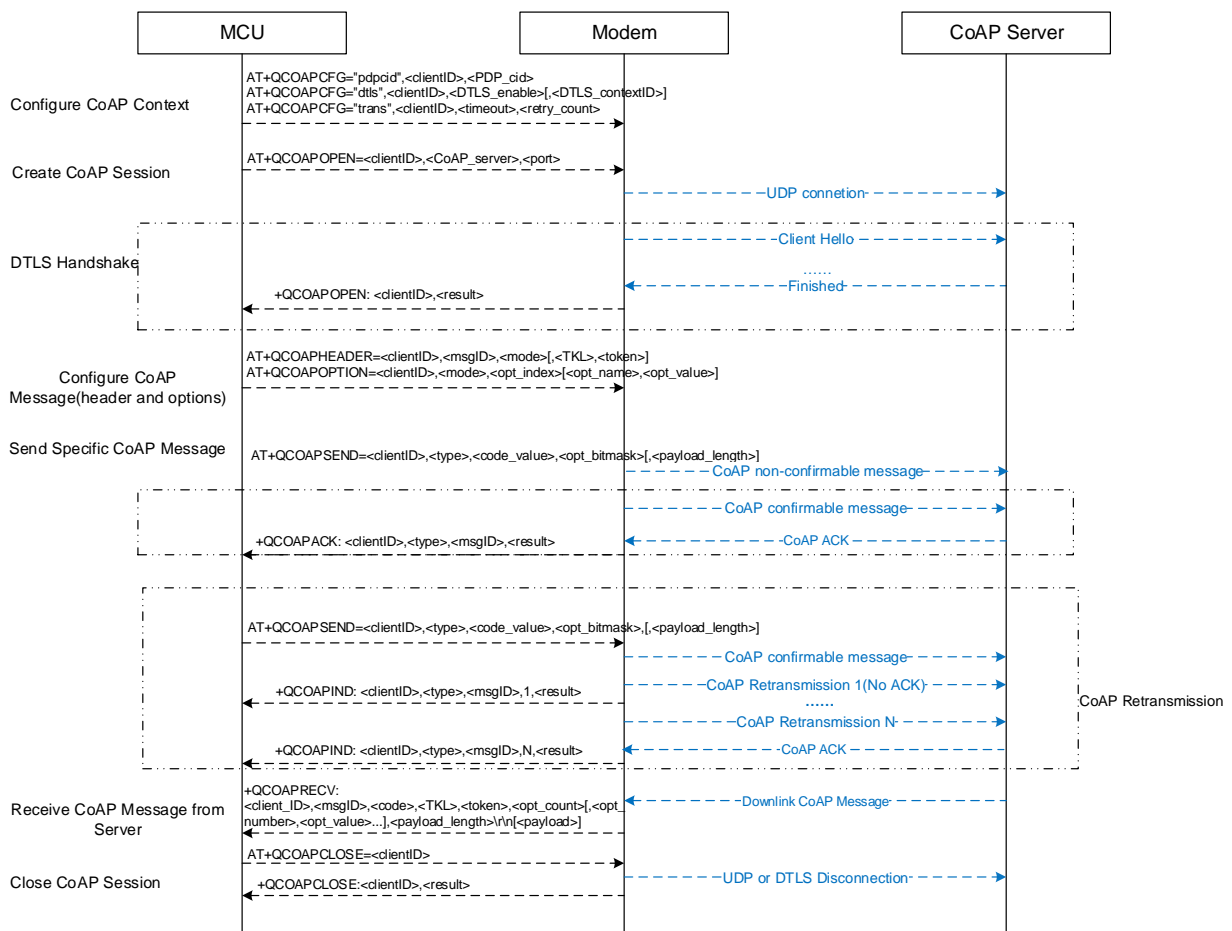


Figure 1: CoAP Data Interaction Diagram

3 Description of CoAP AT Commands

3.1. AT Command 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 to its previous value or the default settings, unless otherwise specified.
- **Underline** Default setting of a parameter.

3.1.2. AT Command Syntax

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

Table 1: Types of AT Commands

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

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 | |
|--|--|
| <p>Test Command</p> <p>AT+QCOAPCFG=?</p> | <p>Response</p> <p>+QCOAPCFG: "pdpcid",(range of supported <clientID>s),(range of supported <PDP_cid>s)</p> <p>+QCOAPCFG: "dtls",(range of supported <clientID>s),(list of supported <DTLS_enable>s),(range of supported <DTLS_contextID>s)</p> <p>+QCOAPCFG: "trans",(range of supported <clientID>s),(range of supported <timeout>s),(range of supported <retry_count>s)</p> <p>OK</p> |
| <p>Write Command</p> <p>Query/Set the PDP context for a specified CoAP client.</p> <p>AT+QCOAPCFG="pdpcid",<clientID>[,<PDP_cid>]</p> | <p>Response</p> <p>If the optional parameter is omitted, query the current setting:</p> <p>+QCOAPCFG: "pdpcid",<PDP_cid></p> <p>OK</p> <p>If the optional parameter is specified, set the PDP context of the specified CoAP client:</p> <p>OK</p> <p>If there is any error:</p> <p>ERROR</p> |
| <p>Write Command</p> <p>Query/Set the DTLS mode for a</p> | <p>Response</p> <p>If the optional parameters are omitted, query the current</p> |

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

Parameter

| | |
|-------------------------------|--|
| <clientID> | Integer type. CoAP client identifier. Range: 0–5. |
| <PDP_cid> | Integer type. The PDP context ID used by CoAP client. Range: 1–16. 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: 0–5. |
| <timeout> | Integer type. The acknowledgement timeout of CoAP confirmable message delivery. Range: 2–60. Default value: 2. Unit: second. |
| <retry_count> | Integer type. The maximum retransmission counts of CoAP confirmable message delivery. Range: 4–8. Default value: 5. |

NOTE

If DTLS mode is enabled for a CoAP session, the PSK file named `<DTLS_contextID>_server.psk` should be uploaded to UFS with **AT+QFUPL** (see **document [2]** 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: (range of supported <code><clientID>s</code>), <code><CoAP_server></code> ,(range of supported <code><port>s</code>) OK |
| Read Command AT+QCOAPOPEN? | Response [+QCOAPOPEN: <code><clientID></code> , <code><CoAP_server></code> , <code><port></code> , <code><status></code>] OK |
| Write Command Configure and connect to a specified CoAP server: AT+QCOAPOPEN=<clientID>,<CoAP_server>,<port> | Response OK +QCOAPOPEN: <code><clientID></code> , <code><result></code> 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: (range 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: (range of supported <clientID>s), (range of supported <msgID>s),(list of supported <m ode>s),(range of supported <TKL>s),<token> OK |
| Write Command | Response |

| | |
|--|--|
| AT+QCOAPHEADER=<clientID>,<msgID>,<mode>[,<TKL>,<token>] | 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 messages of type Acknowledgement/Reset to messages of type Confirmable/Non-confirmable. 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. The 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: (range of supported <clientID>s),(list of supported <mode>s),(range 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. The 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: (range of supported <clientID>s),(range of supported <type>s),<code_value>,<opt_bitmask>,(range 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. The 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. Currently options can be preset in CoAP client by 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 of option index 0 to the CoAP packet |
| | 2 | 00000010 Add the option of option index 1 to the CoAP packet |
| | 3 | 00000011 Add options of option index 0 and 1 to the CoAP packet |
| | 4 | 00000100 Add the option of option index 3 to the CoAP packet |
| | 5 | 00000101 Add options of option index 0 and 3 to the CoAP packet |
| | ... | |
| | 253 | 11111101 Add options of option index 0 and 2–7 to the CoAP packet |
| | 254 | 11111110 Add options of option index 1–7 to the CoAP packet |
| | 255 | 11111111 Add options of option index 0–7 to the CoAP packet |
| <payload_length> | Integer type. The length of the data to be sent. Maximum length: 1024 bytes. If this parameter is omitted, data of any length but 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 messages of type Acknowledgement/Reset to messages of type Confirmable/Non-confirmable. 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. |

NOTES

- <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.
- 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 Indicating Incoming CoAP Message

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

+QCOAPRECV Indicates Incoming CoAP Message

+QCOAPRECV: <clientID>,<msgID>,<code>,<TKL>,<token>,<opt_count>[,<opt_number>,<opt_value>...],<payload_length><CR><LF><payload>

This URC is reported when there is a new CoAP message received from CoAP server.

Parameter

| | |
|--------------|--|
| <clientID> | Integer type. The CoAP client identifier. |
| <msgID> | 16-bit unsigned integer in network byte order. Message ID. |
| <code> | Response code of incoming message. The format is "c.dd". See Table 3 for details. |
| <TKL> | Integer type. The 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 Indicating Delivery Result of CoAP Message

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

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

+QCOAPACK Indicating Delivery Result of CoAP Message

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

Parameter

| | |
|------------|--|
| <clientID> | Integer type. CoAP client identifier. Range: 0–5. |
| <type> | Integer type. The CoAP message type. 0 Confirmable 1 Non-confirmable 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 Notifying Retransmission Result

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

+QCOAPIND Notifying Retransmission Result

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

Parameter

| | |
|------------|---|
| <clientID> | Integer type. CoAP client identifier. Range: 0–5. |
| <type> | Integer type. The 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 by AT+QCOAPCFG="trans" . |
| <result> | Integer type. Result of retransmission. See Chapter 4 for details. |

4 Summary of Result Codes

The following table lists some of the general result codes.

Table 4: Description of <result> Codes

| Code of <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="pdpcid",0,1 //Set the PDP context ID as 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,"220.180.239.212",8028 //Create a CoAP session and connect to the CoAP
                                       //server.
OK
+QCOAPOPEN: 0,0 //Created the CoAP session successfully.
AT+QCOAPOPEN? //Query the current status of the CoAP session.
+QCOAPOPEN: 0,"220.180.239.212",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 20 bytes CoAP non-confirmable message to the server.
>Hello, CoAP Message!
OK
+COAPACK: 0,1,1234,0
```

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

5.2. CoAP Client Operation with DTLS

```
AT+QCOAPCFG="pdpctxid",0,1 //Set the PDP context ID as 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+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+QSSLCFG="dtlsversion",0,1 //Configure DTLS1.2 for CoAP session  
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,"leshan.eclipseprojects.io",5684 //Create a CoAP session and connect to the  
CoAP server.  
OK  
  
+QCOAPOPEN: 0,0 //Created the CoAP session 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 20 bytes CoAP non-confirmable message to the server.
>Hello, CoAP Message!
OK
+COAPACK: 0,1,1234,0
//Receive downlink error message from server side.
+QCOAPRECV: 0,1234,4.04,6,D204D52E814B,0,9
Not Found
AT+QCOAPCLOSE=0 //Close the CoAP session.
OK
+QCOAPCLOSE: 0,0 //Closed the CoAP session successfully.
```

6 Appendix A References

Table 5: Related Documents

| SN | Document Name | Description |
|-----|--|---|
| [1] | RFC 7252 The Constrained Application Protocol (CoAP) | This document is a product of the Internet Engineering Task Force (IETF). |
| [2] | Quectel_BG96_FILE_AT_Commands_Manual | FILE AT commands manual for BG96 module |

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 |