

# BC66&BC66-NA CoAP

# Application Note

**NB-IoT Module Series**

Version: 1.0

Date: 2022-04-21

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](mailto: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](mailto: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. 2022. All rights reserved.**

# About the Document

## Revision History

Version	Date	Author	Description
-	2021-03-19	Milo WANG/ Arno DONG	Creation of the document
1.0	2022-04-21	Milo WANG/ Arno DONG	First official release

## Contents

About the Document .....	3
Contents .....	4
Table Index .....	5
<b>1 Introduction .....</b>	<b>6</b>
<b>2 Description of CoAP AT Commands .....</b>	<b>7</b>
2.1. AT Command Syntax .....	7
2.1.1. Definitions.....	7
2.1.2. AT Command Syntax.....	7
2.2. Declaration of AT Command Examples.....	8
2.3. Description of CoAP AT Commands .....	8
2.3.1. AT+QCOAPCREATE Create CoAP Context.....	8
2.3.2. AT+QCOAPDEL Delete CoAP Context.....	9
2.3.3. AT+QCOAPADDRES Add Resources for CoAP Client.....	10
2.3.4. AT+QCOAPHEAD Configure Message ID and Token for CoAP Messages.....	11
2.3.5. AT+QCOAPOPTION Configure CoAP Options.....	12
2.3.6. AT+QCOAPCFG Configure CoAP Related Parameters .....	14
2.3.7. AT+QCOAPSEND Send CoAP Messages.....	16
2.3.8. AT+QCOAPDATASTATUS Query CON Messages Status.....	18
<b>3 Description of CoAP URCs.....</b>	<b>20</b>
3.1. +QCOAPURC: "rsp" Response URC .....	20
3.2. +QCOAPURC: "req" Request URC.....	21
3.3. +QCOAPURC: recovered Recovery URC .....	22
<b>4 Examples .....</b>	<b>24</b>
4.1. Sending CoAP Requests to CoAP Server.....	24
4.2. Registering to China Telecom IoT Platform.....	25
4.3. Registering to China Telecom IoT Platform by DTLS .....	26
4.4. CoAP Configuration Commands Usage.....	28
<b>5 Appendix A References .....</b>	<b>30</b>

## Table Index

Table 1: Types of AT Commands .....	7
Table 2: CoAP Related URCs.....	20
Table 3: Terms and Abbreviations.....	30

# 1 Introduction

This document gives details of CoAP-related AT command set supported by Quectel NB-IoT BC66 and BC66-NA modules.

# 2 Description of CoAP AT Commands

## 2.1. AT Command Syntax

### 2.1.1. 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.

### 2.1.2. 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	<b>AT+&lt;cmd&gt;=?</b>	Test the existence of the corresponding command and return information about the type, value, or range of its parameter.
Read Command	<b>AT+&lt;cmd&gt;?</b>	Check the current parameter value of the corresponding command.
Write Command	<b>AT+&lt;cmd&gt;=&lt;p1&gt;[,&lt;p2&gt;[,&lt;p3&gt;[...]]]</b>	Set user-definable parameter value.
Execution Command	<b>AT+&lt;cmd&gt;</b>	Return a specific information parameter or perform a specific action.



## 2.2. 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.

## 2.3. Description of CoAP AT Commands

### 2.3.1. AT+QCOAPCREATE Create CoAP Context

This command creates the CoAP context.

<b>AT+QCOAPCREATE Create CoAP Context</b>	
Test Command <b>AT+QCOAPCREATE=?</b>	Response <b>+QCOAPCREATE:</b> (range of supported <local_port>s),(list of supported <mode>s),<psk_id>,<psk> <b>OK</b>  If there is any error: <b>ERROR</b>
Write Command <b>AT+QCOAPCREATE=&lt;local_port&gt;,&lt;mode&gt;[,&lt;psk_id&gt;,&lt;psk&gt;]</b>	Response <b>OK</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	The configurations will not be saved to NVRAM. This command remains valid after the module wakes up from deep sleep.

#### Parameter

<local_port>	Integer type. CoAP context local port. Range: 1–65535.
<mode>	Integer type. Encryption mode. 0 Non-encryption. <psk_id> and <psk> are not specified in this case. 1 Encryption: PRE_SHARED_KEY.
<psk_id>	String type. Pre-shared key ID. The maximum length is 150 bytes.
<psk>	String type. Pre-shared key. In hexadecimal string format. The maximum size is

256 bytes.

**Example**

```

AT+QCOAPCREATE=?
+QCOAPCREATE: (1-65535),(0,1),<psk_id>,<psk>

OK

AT+QCOAPCREATE=56830,0 //Create the non-encryption CoAP context.
OK

AT+QCOAPCREATE=56830,1,"867997030111859","010203040506" //Create the encryption CoAP
context.
OK
    
```

**2.3.2. AT+QCOAPDEL Delete CoAP Context**

This command deletes the CoAP context.

<b>AT+QCOAPDEL Delete CoAP Context</b>	
Execution Command <b>AT+QCOAPDEL</b>	Response <b>OK</b>  <b>+QCOAPDEL: &lt;err&gt;</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	-

**Parameter**

**<err>** Integer type. Prompt that CoAP context is deleted successfully. Currently the value is always 0.

**Example**

```

AT+QCOAPCREATE=56830,0 //Create CoAP context.
OK
    
```

```
AT+QCOAPDEL //Delete CoAP context.
OK
+QCOAPDEL: 0
```

### 2.3.3. AT+QCOAPADDRES Add Resources for CoAP Client

This command adds resources for the CoAP client.

AT+QCOAPADDRES Add Resources for CoAP Client	
Test Command <b>AT+QCOAPADDRES=?</b>	Response <b>+QCOAPADDRES:</b> (range of supported <length>s),<resource>  <b>OK</b>  If there is any error: <b>ERROR</b>
Write Command <b>AT+QCOAPADDRES=&lt;length&gt;,&lt;resource&gt;</b>	Response <b>OK</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	The configurations will not be saved to NVRAM. This command remains valid after the module wakes up from deep sleep.

#### Parameter

<length>	Integer type. Length of CoAP client resources. The maximum length is 50 bytes.
<resource>	String type. Resource value.

#### Example

```
AT+QCOAPADDRES=4,"/t/d" //Add the CoAP resource with the value "/t/d".
OK
```

### 2.3.4. AT+QCOAPHEAD Configure Message ID and Token for CoAP Messages

This command configures the message ID and token for CoAP messages.

<b>AT+QCOAPHEAD Configure Message ID and Token for CoAP messages</b>	
Test Command <b>AT+QCOAPHEAD=?</b>	Response <b>+QCOAPHEAD:</b> (range of supported <b>&lt;mode&gt;s</b> ),(range of supported <b>&lt;messageID&gt;s</b> ),(range of supported <b>&lt;token_length&gt;s</b> ), <b>&lt;token&gt;</b>  <b>OK</b>  If there is any error: <b>ERROR</b>
Write Command <b>AT+QCOAPHEAD=&lt;mode&gt;[,&lt;messageID&gt;][,&lt;token_length&gt;,&lt;token&gt;]</b>	Response <b>OK</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	-

#### Parameter

<b>&lt;mode&gt;</b>	Integer type. CoAP header and token parameters. 1 Generate the message ID and token value randomly. 2 Generate the message ID randomly, and configure the token value. 3 Only configure the message ID, and the token value is omitted. 4 Configure the message ID, and generate the token value randomly. 5 Configure the message ID and token value.
<b>&lt;messageID&gt;</b>	Integer type. The message ID. It is required only when <b>&lt;mode&gt;</b> is 3, 4, 5. Range: 0–65535.
<b>&lt;token_length&gt;</b>	Integer type. Token value length. It is required only when <b>&lt;mode&gt;</b> is 2, 5. Range: 1–8.
<b>&lt;token&gt;</b>	String type. Token value in hexadecimal format. It is required only when <b>&lt;mode&gt;</b> is 2, 5.

**NOTE**

If this command is not executed, the message ID is generated randomly by default and the token value is not needed.

**Example**

```

AT+QCOAPHEAD=1 //Generate the message ID and token value randomly.
OK

AT+QCOAPHEAD=2,8,"0102030405060708" //Generate the message ID randomly, and set the
token value to "0102030405060708".
OK

AT+QCOAPHEAD=3,13940 //Set the message ID to 13940, and the token value is
omitted.
OK

AT+QCOAPHEAD=4,13940 //Set the message ID to 13940, and generate the token
value randomly.
OK

AT+QCOAPHEAD=5,13940,4,"02040608" //Set the message ID and token value to 13940 and
"02040608" respectively.
OK
    
```

**2.3.5. AT+QCOAPOPTION Configure CoAP Options**

This command configures CoAP options.

AT+QCOAPOPTION Configure CoAP Options	
Test Command <b>AT+QCOAPOPTION=?</b>	Response <b>+QCOAPOPTION:</b> (range of supported <opt_cnt>s),(list of supported <opt_number>s),<opt_value>,...  <b>OK</b>  If there is any error: <b>ERROR</b>
Write Command <b>AT+QCOAPOPTION=&lt;opt_cnt&gt;,&lt;opt_number&gt;,&lt;opt_value&gt;[,...]</b>	Response <b>OK</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	-

**Parameter**

<b>&lt;opt_cnt&gt;</b>	Integer type. The count of CoAP options. Range: 1–12.
<b>&lt;opt_number&gt;</b>	Integer type. The option number. Refer to the RFC 7252, RFC 7641 and RFC 7959.
	1 If-Match
	3 Uri-Host
	4 ETag
	5 If-None-Match
	6 Observe
	7 Uri-Port
	8 Location-Path
	11 Uri-Path
	12 Content-Format
	14 Max-Age
	15 Uri-Query
	17 Accept
	20 Location-Query
	23 Block2
	27 Block1
	28 SIZE
	35 Proxy-Uri
	39 Proxy-Scheme
	60 Size1
<b>&lt;opt_value&gt;</b>	String type. The option value. The string length is 1–180 bytes. If <b>&lt;opt_number&gt;</b> is 12 or 17, <b>&lt;opt_value&gt;</b> should be:
	"0" text-plain
	"40" application/link-format
	"41" application/xml
	"42" application/octet-stream
	"47" application/exi
	"50" application/json

**Example**

```

AT+QCOAPOPTION=1,11,"rd" //Configure the CoAP option 11 (Uri-Path) with the value "rd".
OK

AT+QCOAPOPTION=2,11,"rd",15,"ep=86370303" //Configure the CoAP option 11 (Uri-Path) with the
OK //Configure the CoAP option 11 (Uri-Path) with the value "rd", and option 15 (Uri-Query) with the
value "ep=86370303".

AT+QCOAPOPTION=2,11,".well-know",11,"core" //Configure the CoAP option 11 (Uri-Path) with the
OK //Configure the CoAP option 11 (Uri-Path) with the value ".well-know", and option 11 (Uri-Path) with
the value "core".
    
```

### 2.3.6. AT+QCOAPCFG Configure CoAP Related Parameters

This command configures the parameters related to CoAP.

<b>AT+QCOAPCFG CoAP Configuration</b>	
Test Command <b>AT+QCOAPCFG=?</b>	Response <b>+QCOAPCFG: "Showra",</b> (list of supported <Showra>s) <b>+QCOAPCFG: "Showrspopt",</b> (list of supported <Showrspopt>s)  <b>OK</b>  If there is any error: <b>ERROR</b>
Read Command <b>AT+QCOAPCFG?</b>	Response <b>+QCOAPCFG: "Showra",</b> <Showra> <b>+QCOAPCFG: "Showrspopt",</b> <Showrspopt>  <b>OK</b>  If there is any error: <b>ERROR</b>
Write Command <b>AT+QCOAPCFG="Showra",</b> <Showra>]	Response If the optional parameter is omitted, query the current setting: <b>+QCOAPCFG: "Showra",</b> <Showra>  <b>OK</b>  If the optional parameter is specified, configure whether to display the IP address and port of the sender: <b>OK</b>  If there is any error: <b>ERROR</b>
Write Command <b>AT+QCOAPCFG="Showrspopt",</b> <Showrspopt>]	Response If the optional parameter is omitted, query the current setting: <b>+QCOAPCFG: "Showrspopt",</b> <Showrspopt>  <b>OK</b>  If the optional parameter is specified, configure whether to display the CoAP option of the sender: <b>OK</b>

	If there is any error: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	The parameters <b>&lt;Showra&gt;</b> and <b>&lt;Showrspopt&gt;</b> are not saved in NVM, so the configuration will be reset to the default value (0) after shutdown.

**Parameter**

- <Showra>** Integer type. Set whether or not to display the address and port of the sender.
- 0 Not show the address or port.
  - 1 Show the address and port through "rsp" or "req" URC in the format as follows:  
**+QCOAPURC: "rsp",<ip\_addr>,<port>,<type>,<rspcode>,<messageID>[,<length>,<data>]**  
**+QCOAPURC: "req",<ip\_addr>,<port>,<type>,<method>,<messageID>,<mode>,<token\_length>,<token>[,<length>,<data>]**
- <Showrspopt>** Integer type. Set whether or not to display the CoAP option of the sender.
- 0 Not show the CoAP option.
  - 1 Show CoAP options through the URC in the format as follows:  
**+QCOAPURC: "rsp",<type>,<rspcode>,<messageID>,<opt\_cnt>,<opt\_name>,<opt\_value>,...[,<length>,<data>]**

**NOTE**

1. It is recommended to configure the parameters via **AT+QCOAPCFG** before executing **AT+QCOAPSEND**.
2. For more information about URCs, please refer to **Chapter 3**.

**Example**

```

AT+QCOAPCFG?
+QCOAPCFG: "Showra",0
+QCOAPCFG: "Showrspopt",0

OK

AT+QCOAPCFG="Showra",1

OK

AT+QCOAPCFG="Showrspopt",1

OK
    
```



### 2.3.7. AT+QCOAPSEND Send CoAP Messages

This command sends CoAP messages to the IoT platform. **AT+QCOAPDATASTATUS** command shall be used to query the status of CON data sent before sending the next CON or NON data.

AT+QCOAPSEND Send CoAP Messages	
Test Command <b>AT+QCOAPSEND=?</b>	Response <b>+QCOAPSEND:</b> (range of supported <b>&lt;type&gt;s</b> ), <b>&lt;method/rspcode&gt;</b> , <b>&lt;ip_addr&gt;</b> ,(range of supported <b>&lt;port&gt;s</b> ),(range of supported <b>&lt;length&gt;s</b> ), <b>&lt;data&gt;</b>  <b>OK</b>  If there is any error: <b>ERROR</b>
Write Command Send data in fixed length <b>AT+QCOAPSEND=&lt;type&gt;</b> , <b>&lt;method/rspcode&gt;</b> , <b>&lt;ip_addr&gt;</b> , <b>&lt;port&gt;</b> , <b>&lt;length&gt;</b> [, <b>&lt;data&gt;</b> ]	Response <b>OK</b>  <b>+QCOAPSEND: &lt;err&gt;</b>  If there is any error: <b>ERROR</b>
Write Command Send data in fixed length <b>AT+QCOAPSEND=&lt;type&gt;</b> , <b>&lt;method/rspcode&gt;</b> , <b>&lt;ip_addr&gt;</b> , <b>&lt;port&gt;</b> , <b>&lt;length&gt;</b>  After > is responded, the module enters data mode. After that, type the data to be sent until the data length equals to <b>&lt;length&gt;</b> .	Response <b>OK</b>  <b>+QCOAPSEND: &lt;err&gt;</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	-

#### Parameter

<b>&lt;type&gt;</b>	Integer type. The CoAP message type. Range: 0–3. Refer to the RFC 7252. 0 Confirmable (CON) 1 Non-confirmable (NON) 2 Acknowledgement (ACK) 3 Reset (RST)
<b>&lt;method&gt;</b>	Integer type. The CoAP method code. Refer to the RFC 7252.

	1	GET	
	2	POST	
	3	PUT	
	4	DELETE	
<b>&lt;rspcode&gt;</b>	Integer type. The CoAP response code. Refer to the RFC 7252.		
	Value	Response Code	Description
	0	0.00	Empty Message
	201	2.01	Created
	202	2.02	Deleted
	203	2.03	Valid
	204	2.04	Changed
	205	2.05	Content
	400	4.00	Bad Request
	401	4.01	Unauthorized
	402	4.02	Bad Option
	403	4.03	Forbidden
	404	4.04	Not Found
	405	4.05	Method Not Allowed
	406	4.06	Not Acceptable
	412	4.12	Precondition Failed
	413	4.13	Request Entity Too Large
	415	4.15	Unsupported Content-Format
	500	5.00	Internal Server Error
	501	5.01	Not Implemented
	502	5.02	Bad Gateway
	503	5.03	Service Unavailable
	504	5.04	Gateway Timeout
	505	5.05	Proxying Not Supported
<b>&lt;ip_addr&gt;</b>	String type. The IP address of the CoAP server.		
<b>&lt;port&gt;</b>	Integer type. The CoAP server port. Range: 1–65535.		
<b>&lt;length&gt;</b>	Integer type. The length of the input data. It should be set to 0 when sending information without payload. The maximum value is 512 in normal mode and 1024 in data mode. Unit: Byte.		
<b>&lt;data&gt;</b>	String type. The data to be sent. In normal mode, it is an ASCII-encoded hexadecimal string. In data mode, the module does not perform encoding and decoding, and directly sends the received data.		
<b>&lt;err&gt;</b>	Integer type. Result of data sending.		
	0	Success	
	2	DTLS handshake failure	
	3	Failure	

**Example**

```

AT+QCOAPSEND=1,1,"139.196.41.136",5683,0 //Send GET request with NON type to the server.
OK

+QCOAPSEND: 0
AT+QCOAPSEND=0,2,"139.196.41.136,5683",2,"0102" //Send POST request with CON type to the
server.
OK

+QCOAPSEND: 0
AT+QCOAPSEND=1,1,"139.196.41.136",5683,4 //Send GET request with NON type to the server.

>0102 //After > is responded, the module enters data
mode. After that, type the data to be sent until the
data length equals to <length>.

OK

+QCOAPSEND: 0
    
```

**NOTE**

1. In data mode, the maximum data length is 1024 bytes. Data exceeding 1024 bytes will be ignored.
2. If you need to send data with CON type, you must use **AT+QCOAPDATASTATUS** command to get the status of CON data sent before sending the next CON or NON data. See **Chapter 2.3.8** for details about **AT+QCOAPDATASTATUS**.

**2.3.8. AT+QCOAPDATASTATUS Query CON Messages Status**

This command queries the status of the CON data sent to CoAP server.

<b>AT+QCOAPDATASTATUS Query CON Messages Status</b>	
Read Command <b>AT+QCOAPDATASTATUS?</b>	Response <b>+QCOAPDATASTATUS: &lt;status&gt;</b>  <b>OK</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	300 ms
Characteristics	-

## Parameter

<b>&lt;status&gt;</b>	Integer Type. Status of the CON data that has been sent.
0	Have not sent
1	Sent, waiting for response from CoAP server
2	Data sending failed
3	Timeout
4	Success
5	Reset message received

## Example

```

AT+QCOAPSEND=0,1,"139.196.41.136",5683,0 //Send GET request with CON type to CoAP server.
OK

+QCOAPSEND: 0
AT+QCOAPDATASTATUS?
+QCOAPDATASTATUS:1 //Data has been sent. Waiting for response from CoAP server.

OK

+QCOAPURC: "rsp",2,2.05,61440,24,"4E692048616F2066726F6D20436F41502E4E455420524643"
//Received data with the message ID 61440. The data length is 24 bytes and the code is 205.
AT+QCOAPDATASTATUS?
+QLWDATASTATUS:4 //Success.

OK
    
```

# 3 Description of CoAP URCs

This chapter gives CoAP related notifications and their descriptions.

**Table 2: CoAP Related URCs**

URC	Description
+QCOAPURC: "rsp"[,<ip_addr>,<port>],<type>,<rspcode>,<messageID>[,<opt_cnt>,<opt_number>,<opt_value>,...][,<length>,<data>]	When the CON message is sent, the CoAP server will respond with ACK or RST, if the module receives this response, it will report this URC.
+QCOAPURC: "req"[,<ip_addr>,<port>],<type>,<method>,<messageID>,<mode>[,<token_length>,<token>][,<opt_number>,<opt_value>,...][,<length>,<data>]	When the CoAP server sends a request, if the module receives this request, it will report this URC.
+QCOAPURC: recovered,<state>	After the module wakes up from deep sleep, it will report this URC to notify whether the created CoAP context is recovered.

## 3.1. +QCOAPURC: "rsp" Response URC

The module reports the response from the CoAP server to the TE.

### +QCOAPURC: "rsp" Response URC

+QCOAPURC: "rsp"[,<ip_addr>,<port>],<type>,<rspcode>,<messageID>[,<opt_cnt>,<opt_number>,<opt_value>,...][,<length>,<data>]	Notify TE of the response from the CoAP server.
---	---

#### Parameter

<ip_addr>	String type. The IP address of the CoAP server. It will be displayed when setting <b>AT+QCOAPCFG="Showra",1</b> .
<port>	Integer type. The port of the CoAP server. It will be displayed when setting <b>AT+QCOAPCFG="Showra",1</b> .

<b>&lt;type&gt;</b>	Integer type. The CoAP message type. Range: 0–3. Refer to the RFC 7252. 0 Confirmable (CON) 1 Non-confirmable (NON) 2 Acknowledgement (ACK) 3 Reset (RST)
<b>&lt;rspcode&gt;</b>	String type. The CoAP response code. Refer to the RFC 7252.
<b>&lt;messageID&gt;</b>	Integer type. CoAP message ID.
<b>&lt;opt_cnt&gt;</b>	Integer type. The count of options. It will be displayed when setting <b>AT+QCOAPCFG="Showrspopt",1</b>
<b>&lt;opt_number&gt;</b>	Integer type. The option number. It will be displayed when setting <b>AT+QCOAPCFG="Showrspopt",1</b> .
<b>&lt;opt_value&gt;</b>	String type. The option value. It will be displayed when setting <b>AT+QCOAPCFG="Showrspopt",1</b> .
<b>&lt;length&gt;</b>	Integer type. The data length. The maximum length of a URC is 1300 bytes, and the data over 1300 bytes will be discarded.
<b>&lt;data&gt;</b>	String type. If the <b>&lt;opt_number&gt;</b> is 12, and the <b>&lt;opt_value&gt;</b> is "0" (text-plain), "41" (application/xml) or "50" (application/json), the data format is in text string, else in hexadecimal format.

**NOTE**

Options currently supported for parsing are as follows: Observe, Uri-Port, Location-Path, Uri-Path, Content-Format, Uri-Query.

### 3.2. +QCOAPURC: "req" Request URC

The module reports the request from the CoAP server to the TE.

**+QCOAPURC: "req" Request URC**

**+QCOAPURC: "req" [<ip\_addr>, <port>], <type>, <method>, <messageID>, <mode> [<token\_length>, <token>] [<opt\_number>, <opt\_value>, ...] [<length>, <data>]** Notify the request from the CoAP server.

**Parameter**

**<ip\_addr>** String type. The IP address of the CoAP server. It will be displayed when setting

	<b>AT+QCOAPCFG="Showra",1.</b>
<b>&lt;port&gt;</b>	Integer type. The port of the CoAP server. It will be displayed when setting <b>AT+QCOAPCFG="Showra",1.</b>
<b>&lt;type&gt;</b>	Integer type. The CoAP message type. Range: 0-3. Refer to the RFC 7252. 0 Confirmable (CON) 1 Non-confirmable (NON)
<b>&lt;method&gt;</b>	Integer type. The CoAP method code. Refer to the RFC 7252. 1 GET 2 POST 3 PUT 4 DELETE
<b>&lt;messageID&gt;</b>	Integer type. The CoAP message ID.
<b>&lt;mode&gt;</b>	Integer type. Indicate the existence of token, option and data in hexadecimal format. Bit 0 The token exists. Bit 1-6 The count of option. Bit 7 The data exists.
<b>&lt;token_length&gt;</b>	Integer type. Token value length.
<b>&lt;token&gt;</b>	String type. Token value, in hexadecimal format.
<b>&lt;opt_number&gt;</b>	Integer type. The option number.
<b>&lt;opt_value&gt;</b>	String type. The option value.
<b>&lt;length&gt;</b>	Integer type. The data length. The maximum length of a URC is 1300 bytes, and the data over 1300 bytes will be discarded.00 bytes will be discarded.
<b>&lt;data&gt;</b>	String type. If the <b>&lt;opt_number&gt;</b> is 12, and the <b>&lt;opt_value&gt;</b> is "0" (text-plain), "41" (application/xml) or "50" (application/json), the data is in text format, else in hexadecimal format.

**NOTE**

Options currently supported for parsing are as follows: Observe, Uri-Port, Location-Path, Uri-Path, Content-Format, Uri-Query.

### 3.3. +QCOAPURC: recovered Recovery URC

**+QCOAPURC: recovered Recovery URC**

<b>+QCOAPURC: recovered,&lt;state&gt;</b>	After the module wakes up from deep sleep, it will report this URC to notify whether the created CoAP context is recovered.
---	---

**Parameter**

---

<b>&lt;state&gt;</b>	Integer type. CoAP context recovery results.
0	Success
1	Failure. It has to re-create CoAP context using <b>AT+QCOAPCREATE</b> .

---



# 4 Examples

## 4.1. Sending CoAP Requests to CoAP Server

```

AT+CGATT? //Query the PS service attaching status.
+CGATT: 1 //PS service is attached.

OK
AT+QCOAPCREATE=56830,0 //Create the CoAP context.
OK
AT+QCOAPOPTION=1,11,"validate"//Configure the CoAP option 11 (Uri-Path) with the value "validate".
OK
AT+QCOAPSEND=0,1,"139.196.187.107",5683,0 //Send GET request with CON type to the server.
OK

+QCOAPSEND: 0

AT+QCOAPDATASTATUS? //Query the data sent status.
+QCOAPDATASTATUS: 1 //Data is sent. Waiting for response from the server.

OK

//Data is received from the server. The code is 2.05, the message ID is 12194, the data length is 9 bytes,
and the data is in text string.
+QCOAPURC: "rsp",2,2.05,12194,9,123123456

AT+QCOAPDATASTATUS? //Query the data sent status.
+QCOAPDATASTATUS: 4 //Success.

OK
AT+QCOAPHEAD=3,5566 //Set the CoAP message ID to 5566, and the token value is omitted.
OK
AT+QCOAPOPTION=1,11,"validate"//Configure the CoAP option 11(Uri-Path) with the value "validate".
OK
AT+QCOAPSEND=0,1,"139.196.187.107",5683,0 //Send GET request with CON type to the server.
OK
    
```

```

+QCOAPSEND: 0
//Data is received from the server. The code is 2.05, the message ID is 5566 (same as the set ID), the data
length is 9 bytes, and the data is in text string.
+QCOAPURC: "rsp",2,2.05,5566,9,123123456

AT+QCOAPHEAD=5,6677,4,"30323436" //Configure the header message ID and token value
parameters of CoAP messages.
OK
AT+QCOAPOPTION=1,11,"test" //Configure the CoAP option 11 (Uri-Path) with the
value "test".
OK
AT+QCOAPSEND=0,1,"139.196.187.107",5683,0 //Send GET request with CON type to the server.
OK

+QCOAPSEND: 0
//Data is received from the server. The code is 2.05, the message ID is 6677, the data length is 53 bytes,
and the data is in text string.
+QCOAPURC: "rsp",2,2.05,6677,53,Type: 0 (CON)
Code: 1 (GET)
MID: 6677
Token: 30323436
AT+QCOAPDEL // Delete CoAP context.
OK

+QCOAPDEL: 0

```

## 4.2. Registering to China Telecom IoT Platform

```

AT+QCOAPCREATE=56830,0 //Create CoAP context.
OK
AT+QCOAPADDRES=6,"/4/0/2" //Add the CoAP resource, the value is "/4/0/2".
OK
AT+QCOAPADDRES=6,"/4/0/3" //Add the CoAP resource, the value is "/4/0/3".
OK
AT+QCOAPADDRES=6,"/4/0/8" //Add the CoAP resource, the value is "/4/0/8".
OK
AT+QCOAPADDRES=6,"/3/0/9" //Add the CoAP resource, the value is "/3/0/9".
OK
AT+QCOAPADDRES=6,"/3/0/7" //Add the CoAP resource, the value is "/3/0/7".
OK
AT+QCOAPADDRES=7,"/19/0/0" //Add the CoAP resource, the value is "/19/0/0".
OK

```

```

AT+QCOAPADDRES=7,"/19/1/0" //Add the CoAP resource, the value is "/19/1/0".
OK
AT+QCOAPHEAD=1 //Generate message ID and token values randomly.
OK

//Configure the CoAP option 11 (Uri-Path) with the value "rd", option 12 with the value "42", option 15 (Uri-
Query) with the value "lwm2m=1.0", option 15 (Uri-Query) with the value
"ep=867725030002525;460041850403693", option 15 (Uri-Query) with the value "b=U", and option 15
(Uri-Query) with the value "lt=86400".
AT+QCOAPOPTION=6,11,"rd",12,"42",15,"lwm2m=1.0",15,"ep=867725030002525;46004185040369
3",15,"b=U",15,"lt=86400"
OK
//Send registration messages to China Telecom IoT platform.
AT+QCOAPSEND=0,2,"180.101.147.115",5683,62,"3c2f3e3b72743d226f6d612e6c776d326d222c3c2
f312f303e2c3c2f332f303e2c3c2f342f303e2c3c2f352f303e2c3c2f31392f303e2c3c2f31392f313e"
OK

+QCOAPSEND: 0
//Response is received from the server. The type is ACK, the code is 2.01, and the message ID is 55018.
+QCOAPURC: "rsp",2,2.01,55018
//Request is received from the server. The type is CON, the code is GET, the message ID is 62084, the
<mode> is 09, the token value is "1A98EB114F501C07", and the option count is 4.
+QCOAPURC: "req",0,1,62084,09,8,"1A98EB114F501C07",6,"0",11,"19",11,"0",11,"0"

//When receiving the request data, the client should respond it.
//Configure the CoAP token as that of in the request message.
AT+QCOAPHEAD=5,62084,8,"1A98EB114F501C07"
OK
AT+QCOAPSEND=2,205,"180.101.147.115",5683,0
OK

+QCOAPSEND: 0
AT+QCOAPDEL // Delete the CoAP context.
OK

+QCOAPDEL: 0

```

### 4.3. Registering to China Telecom IoT Platform by DTLS

```

AT+QCOAPCREATE=56830,1,"867997030314412","010203040506070809" //Create the CoAP context.
OK
AT+QCOAPADDRES=6,"/4/0/8" //Add the CoAP resource, the value is "/4/0/8".

```

```

OK
AT+QCOAPADDRES=7,"/19/1/0" //Add the CoAP resource, the value is "/19/1/0".
OK
AT+QCOAPADDRES=7,"/19/0/0" //Add the CoAP resource, the value is "/19/0/0".
OK
AT+QCOAPHEAD=1 //Generate message ID and token values randomly.
OK
//Configure the CoAP option 11 (Uri-Path) with the value "rd", option 12 with the value "40", option 15 (Uri-
Query) with the value "lwm2m=1.0", option 15 (Uri-Query) with the value "ep=867997030314412", option
15 (Uri-Query) with the value "b=U"; and option 15 (Uri-Query) with the value "lt=86400".
AT+QCOAPOPTION=6,11,"rd",12,"40",15,"lwm2m=1.0",15,"ep=867997030314412",15,"b=U",15,"lt=
86400"
OK
//Send registration messages to China Telecom IoT Platform.
AT+QCOAPSEND=0,2,"180.101.147.115",5684,62,"3c2f3e3b72743d226f6d612e6c776d326d222c3c2
f312f303e2c3c2f332f303e2c3c2f342f303e2c3c2f352f303e2c3c2f31392f303e2c3c2f31392f313e"
OK

+QCOAPSEND: 0
//Response is received from the server. The type is ACK, the code is 2.01, and the message ID is 43586.
+QCOAPURC: "rsp",2,2.01,43586

//Request is received from the server. The type is CON, the method code is GET, the message ID is 9620,
the <mode> is 09, the token value is "2160BCAE7A128C04", and the option count is 4.
+QCOAPURC: "req",0,1,9620,09,8,"2160BCAE7A128C04",6,"0",11,"19",11,"0",11,"0"

//When receiving the request, the client should respond it.
//Configure the CoAP token as that of the request message
AT+QCOAPHEAD=5,9620,8,"2160BCAE7A128C04"
OK
AT+QCOAPSEND=2,205,"180.101.147.115",5684,0
OK

+QCOAPSEND: 0
//Request is received from the server. The type is CON, the method code is GET, the message ID is 9621,
the <mode> is 0B, the token value is "ABE58C92", and the option count is 5.
+QCOAPURC: "req",0,1,9621,0B,4,"ABE58C92",7,"7593",11,"4",11,"0",11,"8",12,"0"
AT+QENG=0 //Query Cell ID.
+QENG: 0,3738,2,105,"5C4BC33",-69,-11,-58,4,8,"4C10",0,-30,3

OK
AT+QCOAPHEAD=5,9621,4,"ABE58C92"
OK
AT+QCOAPSEND=2,205,"180.101.147.115",5684,8,"3936373739333135"

```

```

OK

+QCOAPSEND: 0
AT+QCOAPHEAD=2,8,"2160BCAE7A128C04" //Set the same token value to "2160BCAE7A128C04",
                                         and generate message ID randomly.

OK
AT+QCOAPSEND=0,205,"180.101.147.115",5684,3,"010203"//Send data "010203" to the server.
OK

+QCOAPSEND: 0

+QCOAPURC: "rsp",2,0.00,43587
//Request is received from the server. The type is CON, the method code is POST, the message ID is 9624,
the <mode> is 8B, the token value is "4733", the option count is 5, and the data is "FFFE01134C9A0000".

+QCOAPURC: "req",0,2,9624,8B,2,"4733",7,"7593",11,"19",11,"1",11,"0",12,"42",8,"FFFE01134C9
A0000"
AT+QCOAPHEAD=5,9624,2,"4733"
OK
AT+QCOAPSEND=2,204,"180.101.147.115",5684,0 //Answer the server.
OK

+QCOAPSEND: 0
AT+QCOAPDEL //Delete the CoAP context.
OK

+QCOAPDEL: 0
    
```

#### 4.4. CoAP Configuration Commands Usage

```

AT+QCOAPCREATE=56830,0 //Create the CoAP context.
OK
AT+QCOAPCFG="Showsport",1 //Show the CoAP option of the response URC.
OK
AT+QCOAPOPTION=5,11,"rd",12,"40",15,"ep=863405040004373",15,"b=U",15,"lt=2700"
OK
AT+QCOAPSEND=0,2,"220.180.239.212",8063,22,"3c2f3e3b72743d226f6d612e6c776d326d222c3c2
f30"
OK

+QCOAPSEND: 0
AT+QCOAPDATASTATUS?
    
```

**+QCOAPDATASTATUS: 4**

OK

**+QCOAPURC: "rsp",2,2.01,13357,2,8,"rd",8,"863405040004373"**

//Response is received via URC with the options of 8(location-path) "rd" and 8(location path) "863405040004373".

//Set the <Showra> to 1. When data from the peer is received, IP address and port will be displayed.

**AT+QCOAPCFG="Showra",1**

OK

**AT+QCOAPOPTION=5,11,"rd",12,"40",15,"ep=863405040004373",15,"b=U",15,"It=2700"**

OK

**AT+QCOAPSEND=0,2,"220.180.239.212",8063,22,"3c2f3e3b72743d226f6d612e6c776d326d222c3c2f30"**

OK

**+QCOAPSEND: 0**

**+QCOAPURC: "rsp",220.180.239.212,8063,2,2.01,13358,2,8,"rd",8,"863405040004373"**

//Response is received via URC with the options of 8 (location-path) "rd" and 8 (location-path) "863405040004373", and also the remote server IP address and port.

**AT+QCOAPCFG="Showrspopt",0**

OK

**AT+QCOAPCFG="Showra",0**

OK

**AT+QCOAPOPTION=2,11,"hello",15,"name=aaaa"**

OK

**AT+QCOAPSEND=0,1,"220.180.239.212",8098,0**

OK

**+QCOAPSEND: 0**

//The content-format is "0", and the data format is in text string.

**+QCOAPURC: "rsp",2,2.05,54750,41,hi name=aaaa,this is quectel iot platform**

**AT+QCOAPDEL** //Delete the CoAP context.

OK

**+QCOAPDEL: 0**

# 5 Appendix A References

**Table 3: Terms and Abbreviations**

Abbreviation	Description
ACK	Acknowledgement
CoAP	Constrained Application Protocol
CR	Carriage Return
ID	Identification
IP	Internet Protocol
IoT	Internet of Things
LF	Line Feed
ME	Mobile Equipment
OPT	Option
PDU	Protocol Data Unit
PIN	Personal Identification Number
PS	Packet Switch
PUK	PIN Unlock Key
RST	Reset
SMS	Short Message Service
SMSC	Short Message Service Center
TE	Terminal Equipment
URC	Unsolicited Result Code
USIM	Universal Subscriber Identity Module