BC66&BC66-NA TCP/IP Application Note

NB-IoT Module Series

Rev. BC66&BC66-NA_TCP/IP_Application_Note_V2.0

Date: 2020-03-17

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.

COPYRIGHT

About the Document

Revision History

Version	Date	Author	Description
1.0	2018-08-28	Oven TAO	Initial
2.0	2020-03-17	Koala LI	1. Added applicable module BC66-NA. 2. Added the characteristics of AT commands.

Contents

Table Index

1 Introduction

Quectel BC66 and BC66-NA modules feature embedded TCP/IP stack, which enables the host to access the Internet directly over AT commands. This greatly reduces the dependence on PPP and external TCP/IP protocol stacks and thus minimizes the cost.

Quectel BC66&BC66-NA modules provide the following socket services: TCP client and UDP client.

1.1. The Process of Using TCP/IP AT Commands

Through TCP/IP AT commands, the host can open/close socket services and send/receive data via socket service.

1.2. Description of Data Access Modes

BC66 and BC66-NA modules support the following two kinds of data access modes:

- Buffer access mode
- Direct push mode

When opening a socket service with **AT+QIOPEN**, the data access mode can be specified by the parameter **<access_mode>**. After a socket service is opened, the data access mode can be changed with **AT+QISWTMD**.

1. In buffer access mode, the data can be sent with **AT+QISEND** or **AT+QISENDEX** command. When the data is received, the module will buffer the data and report a URC in format of **+QIURC: "recv",<connectID>[,<current_recv_length>]**. The host can read data with **AT+QIRD**.

NOTE

In buffer access mode, if the buffer is not empty, the module will not report a new URC until all the received data has been read with **AT+QIRD** from the buffer.

2. In direct push mode, the data can be sent with **AT+QISEND** or **AT+QISENDEX** command. The received data will be outputted directly via the following URC:
   +QIURC: "recv",<connectID>,<current_recv_length><CR><LF><data>.

2 Description of TCP/IP AT Commands

2.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on command line. When an optional parameter is not given, the new value equals to its previous value or its default setting, unless otherwise specified.
- **Underline** Default setting of a parameter.

2.2. Types of AT Commands

Table 1: Types of AT Commands and Responses

Test Command	**AT+<cmd>=?**	This command returns the list of parameters and value ranges set by the corresponding Write Command or internal processes.
Read Command	**AT+<cmd>?**	This command returns the currently set value of the parameter or parameters.
Write Command	**AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]**	This command sets the user-definable parameter values.
Execution Command	**AT+<cmd>**	This command reads non-variable parameters affected by internal processes in the module.

2.3. Description of AT Commands

2.3.1. AT+QIOPEN Open a Socket Service

This command is used to open a socket service. The service type can be specified by **<service_type>**, and the data access mode can be specified by **<access_mode>**. The URC **+QIOPEN: <connectID>,<err>** will be reported to indicate whether the socket service has been opened successfully.

AT+QIOPEN Open a Socket Service	
Test Command **AT+QIOPEN=?**	Response **+QIOPEN: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"TCP/UDP","<IP_address>/<domain_name>",<remote_port>,<local_port>,(**list of supported **<access_mode>**s**),(**list of supported **<protocol_type>**s**)** **OK**
Write Command **AT+QIOPEN=<contextID>,<connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>[,<local_port>[,<access_mode>][,<protocol_type>]]**	Response **OK** **+QIOPEN: <connectID>,<err>** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. Remain valid after deep-sleep wakeup, but the configurations of **<IP_address>/<domain_name>**, **<remote_port>**, **<local_port>**, **<access_mode>** and **<protocol_type>** will not be saved to NVRAM.

Parameter

<contextID>	Integer type. Context ID. The range is 1-3.
<connectID>	Integer type. Socket service index. The range is 0-4.
<service_type>	String type. Socket service type. "TCP" Start a TCP connection as a client "UDP" Start a UDP connection as a client
<IP_address>	String type. IP address of the remote server, such as 220.18.23.22. The maximum size is 150 bytes.
<domain_name>	String type. Domain name address of the remote server. The maximum size is 150 bytes.

<remote_port>	Integer type. Port number of the remote server. The range is 1-65535.
<local_port>	Integer type. Local port number. 0 The local port will be assigned automatically 1-65535 The local port will be assigned as specified
<access_mode>	Integer type. Data access mode of socket services. 0 Buffer access mode 1 Direct push mode
<protocol_type>	Integer type. Internet protocol type. 0 IPv4 1 IPv6
<err>	Integer type. The error code. Please refer to ***Chapter 3*** for details.

NOTES

1. Currently, only **<contextID>**=1 is supported.
2. It is recommended to wait for 60 seconds for URC response **+QIOPEN: <connectID>,<err>**.
3. If the connection failed, **AT+QICLOSE=<connectID>** must be executed to close the socket.
4. This command should be executed after the IP address URC (e.g. **+IP: 10.18.237.42**, indicating successfully registered to the network) is reported.
5. The module cannot enter deep sleep mode when a TCP session is created. After the connection is closed, the module will be able to enter deep sleep mode as expected.
6. When a UDP session is created, the module can automatically backup the latest UDP configurations, and the MCU can send/receive data directly after being woken up from sleep.

2.3.2. AT+QICLOSE Close a Socket Service

This command is used to close a specified socket service.

AT+QICLOSE Close a Socket Service	
Test Command **AT+QICLOSE=?**	Response **+QICLOSE: (**range of supported **<connectID>**s**)** **OK**
Write Command **AT+QICLOSE=<connectID>**	Response If closed successfully: **OK** **CLOSE OK** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms

Characteristics	/

Parameter

<connectID>	Integer type. Socket service index. The range is 0-4.

2.3.3. AT+QISTATE Query Socket Service Status

This command is used to query the socket service status.

AT+QISTATE Query Socket Service Status	
Test Command **AT+QISTATE=?**	Response **OK**
Read Command **AT+QISTATE?**	Response Return the status of all existing connections: **[**List of **(+QISTATE: <connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>)]** **OK** If there is an error related to ME functionality: **ERROR**
Write Command If **<query_type>** is 0, check the connection status of the specified context **AT+QISTATE=<query_type>,<contextID>**	Response Return the status of all existing connections: **[**List of **(+QISTATE: <connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>)]** **OK** If there is an error related to ME functionality: **ERROR**
Write Command If **<query_type>** is 1, check the connection status of a specified socket service **AT+QISTATE=<query_type>,<connectID>**	Response Return the connection status of a specified socket service: **[+QISTATE: <connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>]** **OK**

	If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<query_type>	Integer type. Query type. 0 Query connection status by **<contextID>** 1 Query connection status by **<connectID>**
<contextID>	Integer type. Context ID. The range is 1-3.
<connectID>	Integer type. Socket service index. The range is 0-4.
<service_type>	String type. Service type. "TCP" TCP connection as a client "UDP" UDP connection as a client
<IP_address>	String type. IP address of remote client.
<domain_name>	String type. Domain name address of the remote server.
<remote_port>	Integer type. Port number of the remote server.
<local_port>	Integer type. Local port number assigned.
<socket_state>	Integer type. Socket service state. 0 "Initial": client connection has not been established 1 "Connecting": client is connecting 2 "Connected": client connection has been established 3 "Closing": client connection is closing 4 "Remote Closing": client connection being closed by the remote server
<access_mode>	Integer type. Data access mode. 0 Buffer access mode 1 Direct push mode

NOTES

1. Currently, only **<contextID>**=1 is supported.
2. If no list of **+QISTATE:** is displayed in the response, then there is no connection currently.

2.3.4. AT+QISEND Send Hex/Text String Data

This command is used to send socket data in hex/text string format via a specified connection.

AT+QISEND Send Hex/Text String Data	
Test Command **AT+QISEND=?**	Response **+QISEND: (**range of supported **<connectID>**s**),(**range of supported **<send_length>**s**),<data>** **OK**
Write Command **AT+QISEND=<connectID>,<send_length>,<data>**	Response If data is sent successfully: **OK** **SEND OK** Otherwise: **OK** **SEND FAIL** If there is an error related to ME functionality: **ERROR**
Write Command Send data in variable-length **AT+QISEND=<connectID>**	Response **>** After **>** is responded, the module enters data mode. After that, type the data to be sent. Tap **Ctrl+Z** to send the data, or tap **Esc** to cancel the operation. If data is sent successfully: **OK** **SEND OK** Otherwise: **OK** **SEND FAIL** If there is an error related to ME functionality: **ERROR**
Write Command Send data in fixed-length **AT+QISEND=<connectID>,<send_length>**	Response **>** After **>** is responded, the module enters data mode. After that, type the data to be sent until the data length equals to **<send_length>**.

	If data is sent successfully: **OK** **SEND OK** Otherwise: **OK** **SEND FAIL** If there is an error related to ME functionality: **ERROR**
Write Command Check the total length of the data (sent by both **AT+QISEND** and **AT+QISENDEX**) that has been acknowledged and that has been sent but not acknowledged **AT+QISEND=<connectID>,0**	Response **+QISEND: <sent>,<acked>,<nAcked>** **OK** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<connectID>	Integer type. Socket service index. The range is 0-4.
<send_length>	Integer type. The length of data to be sent in bytes. The max length is 1024 bytes in **Text mode** and 512 bytes in **Hex mode**. In data mode, the max length is 1460 bytes in **Text mode** and 730 bytes in **Hex mode**.
<data>	String type. The hex/text string data to be sent.
<sent>	Integer type. A numeric indicates the total length of the data that has been sent through the session. Unit: byte.
<acked>	Integer type. A numeric indicates the total length of the data that has been acknowledged by the remote server, only applicable for TCP session.
<nAcked>	Integer type. A numeric indicates the total length of the data has been sent but not acknowledged by the remote server, only applicable for TCP session.

NOTES

1. **SEND OK** only indicates that the data has arrived the protocol stack.
2. Please note that **<send_length>** must equal to the length of **<data>**.

3. Please enclose **<data>** in double quotation marks if special characters such as JSON are included. Currently, **<data>** does not support special command characters such as semicolons.
4. The MCU should wait for **SEND OK/SEND FAIL** indication message before issuing the next data sending operation.

2.3.5. AT+QIRD Retrieve the Received TCP/IP Data

This command is used to read the received socket data from a specified connection.

In buffer access mode, after receiving data, the module will buffer it and then report URC **+QIURC: "recv",<connectID>[,<current_recv_length>]** to the external MCU to indicate incoming data.

AT+QIRD Retrieve the Received TCP/IP Data	
Test Command **AT+QIRD=?**	Response **+QIRD: (**range of supported **<connectID>**s**),(**range of supported **<read_length>**s**)** **OK**
Write Command **AT+QIRD=<connectID>,<read_length>**	Response **+QIRD: <actual_read_length>[,<remaining_length>]** **<data>** **OK** If there is no data: **+QIRD: 0** **OK** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<connectID>	Integer type. Socket service index. The range is 0-4.
<read_length>	Integer type. The maximum length of data to be retrieved. The range is 1-512. Unit: byte.
<actual_read_length>	Integer type. The actual length of received data. Unit: byte.
<remaining_length>	Integer type. The remaining length of last received data. Unit: byte.

<data>	The retrieved data.

NOTES

1. If the module receives data again when the receive buffer is not empty, then it will not report a new URC until all the received data has been retrieved from the buffer.
2. **<current_recv_length>** in the incoming data indicating URC and **<remaining_length>** in the response of **AT+QIRD** will be prompted only when **AT+QICFG="showlength",1** is set.
3. The remaining length is not the total received bytes in buffer. It only indicates the current remaining data stored in one node.

2.3.6. AT+QISENDEX Send Hex String Data

This command can be used to send socket data in hex string format via a specified connection.

AT+QISENDEX Send Hex String Data	
Test Command **AT+QISENDEX=?**	Response **+QISENDEX: (**range of supported **<connectID>**s**),(**range of supported **<send_length>**s**),<hex_string>** **OK**
Write Command **AT+QISENDEX=<connectID>,<send_length>,<hex_string>**	Response If the hex string data is sent successfully: **OK** **SEND OK** Otherwise: **OK** **SEND FAIL** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<connectID>	Integer type. Socket service index. The range is 0-4.
<send_length>	Integer type. The length of data to be sent, and the max length is 512 bytes.

<hex_string>	String type. The hex string data to be sent.

NOTES

1. **SEND OK** only indicates that the data arrives the protocol stack.
2. The MCU should wait for **SEND OK/SEND FAIL** indication message before issuing the next data sending operation.

2.3.7. AT+QISWTMD Switch Data Access Modes

This command is used to switch the data access modes: buffer access mode or direct push mode. When starting a new socket service, the host can specify the data access mode by the parameter **<access_mode>** via **AT+QIOPEN**.

AT+QISWTMD Switch Data Access Modes	
Test Command **AT+QISWTMD=?**	Response **+QISWTMD: (**range of supported **<connectID>**s**),(**list of supported **<access_mode>**s**)** **OK**
Read Command **AT+QISWTMD?**	Response **OK**
Write Command **AT+QISWTMD=<connectID>,<access_mode>**	Response **OK** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. Remain valid after deep-sleep wakeup. The configuration will not be saved to NVRAM.

Parameter

<connectID>	Integer type. Socket service index. The range is 0-4.
<access_mode>	Integer type. The data access modes of the socket service. 0 Buffer access mode 1 Direct push mode

2.3.8. AT+QPING Ping a Remote Server

This command is used to test the Internet Protocol reachability of a host device.

AT+QPING Ping a Remote Server	
Test Command **AT+QPING=?**	Response **+QPING: (**range of supported **<contextID>**s**),"<host>",(**range of supported **<time_out>**s**),(**range of supported **<ping_num>**s**),(**range of supported **<ping_size>**s**)** **OK**
Write Command **AT+QPING=<contextID>,<host>[,<time_out>[,<ping_num>[,<ping_size>]]]**	Response If ping a remote server successfully: **OK** **+QPING: <result>[,<IP_address>,<bytes>,<time>,<ttl>]** **[…]** **+QPING: <finresult>[,<sent>,<rcvd>,<lost>,<min>,<max>,<avg>]** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<contextID>	Integer type. Context ID. The range is 1-3.
<host>	The host address in string type. The format is a domain name or a dotted decimal IP address. The maximum size is 150 bytes.
<time_out>	Integer type. The maximum time to wait for the response of each ping request. Unit: second. Range: 1-255. Default: 4.
<ping_num>	Integer type. The maximum time of ping request. Range: 1-10. Default: 4.
<ping_size>	Integer type. The ping size. Range: 32-200. Default: 32.
<result>	Integer type. The result of each ping request. 0 Received the ping response from the server. Others Refer to ***Chapter 3*** for specified error codes.
<IP_address>	String type. The IP address of the remote server formatted as a dotted decimal IP.

<bytes>	Integer type. The length of each sending ping request. Unit: byte.
<time>	Integer type. The time consuming of the ping request. Unit: ms.
<ttl>	Integer type. The time to live value of the ping request.
<finresult>	Integer type. The final result of the ping operation. 0 Ping successful Others Refer to ***Chapter 3*** for specified error codes.
<sent>	Integer type. The total number of bytes sent by the ping requests.
<rcvd>	Integer type. The total number of bytes received in the ping response.
<lost>	Integer type. The total number of bytes lost in the ping requests.
<min>	Integer type. The minimum response time. Unit: ms.
<max>	Integer type. The maximum response time. Unit: ms.
<avg>	Integer type. The average response time. Unit: ms.

NOTES

1. Currently, only **<contextID>**=1 is supported.
2. Currently IPv6 ping is not supported.

2.3.9. AT+QNTP Synchronize Local Time through NTP Server

This command is used to synchronize the local time with the Universal Time Coordinated (UTC) via the NTP server.

AT+QNTP Synchronize Local Time through NTP Server	
Test Command **AT+QNTP=?**	Response **+QNTP: (**range of supported **<contextID>**s**),"<server>",<port>,(**list of supported **<auto_set_time>**s**)** **OK**
Write Command **AT+QNTP=<contextID>,<server>[,<port>][,<auto_set_time>]**	Response If successfully synchronized: **OK** **+QNTP: <err>,<time>** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<contextID>	Integer type. Context ID. The range is 1-3.
<server>	String type. Address of the NTP server. The format is a domain name or a dotted decimal IP address.
<port>	Integer type. Port number of the NTP server (123 by default).
<auto_set_time>	Integer type. Whether to automatically set synchronized time to local time. 0 Not set 1 Set
<time>	String type. The time synchronized from NTP server. The format is "YY/MM/DD,hh:mm:ss±zz". The range of "zz" is -47 to 48.
<err>	Integer type. The error code. Please refer to ***Chapter 3*** for details.

NOTES

1. Currently, only **<contextID>**=1 is supported.
2. When **<auto_set_time>** is set to 1, RTC will be updated by the synchronized time automatically. And then **AT+QCCLK?** can be used to check the updated time.
3. The module will automatically update the RTC time after successfully registered to the network.

2.3.10. AT+QIDNSGIP Get IP Address by Domain Name

This command is used to covert a specified domain name to IP address format.

AT+QIDNSGIP Get IP Address by Domain Name	
Test Command **AT+QIDNSGIP=?**	Response **+QIDNSGIP: (**range of supported **<contextID>**s**),"<hostname>"** **OK**
Write Command **AT+QIDNSGIP=<contextID>,<hostname>**	Response **OK** **+QIURC: "dnsgip",<err>,<IP_count>,<DNS_ttl>** **[+QIURC: "dnsgip",<hostIPaddr>]** **[…]** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms

Characteristics	/

Parameter

<contextID>	Integer type. Context ID. The range is 1-3.
<hostname>	String type. Domain name.
<IP_count>	Integer type. The number of the IP addresses corresponding to the **<hostname>**.
<DNS_ttl>	Integer type. The time to live value of the DNS.
<hostIPaddr>	String type. The IP address of **<hostname>**.
<err>	Integer type. The error code. Please refer to ***Chapter 3*** for details.

NOTE

Currently, only **<contextID>**=1 is supported

2.3.11. AT+QIDNSCFG Configure DNS Server Address

This command is used to configure the primary and secondary DNS server addresses.

AT+QIDNSCFG Configure DNS Server Address	
Test Command **AT+QIDNSCFG=?**	Response **+QIDNSCFG: (**range of supported **<contextID>**s**),<pridnsaddr>,<secdnsaddr>** **OK**
Write Command Check the primary and secondary DNS server addresses after successful configuration **AT+QIDNSCFG=<contextID>**	Response **[+QIDNSCFG: <contextID>,<pridnsaddr_ipv4>,<secdnsaddr_ipv4>,<pridnsaddr_ipv6>,<secdnsaddr_ipv6>** **OK** If there is an error related to ME functionality: **ERROR**
Write Command Configure the primary and secondary DNS server addresses **AT+QIDNSCFG=<contextID>,<pridnsaddr>[,<secdnsaddr>]**	Response **OK** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms

Characteristics	The command takes effect immediately. Remain valid after deep-sleep wakeup. The configurations will not be saved to NVRAM.

Parameter

<contextID>	Integer type. Context ID. The range is 1-3.
<pridnsaddr>	String type. Primary DNS server address in IP format. The maximum size is 64 bytes.
<secdnsaddr>	String type. Secondary DNS server address in IP format. The maximum size is 64 bytes.
<pridnsaddr_ipv4>	String type. IPv4 primary DNS server address in IP format.
<secdnsaddr_ipv4>	String type. IPv4 secondary DNS server address in IP format.
<pridnsaddr_ipv6>	String type. IPv6 primary DNS server address in IP format.
<secdnsaddr_ipv6>	String type. IPv6 secondary DNS server address in IP format.

NOTES

1. Currently, only **<contextID>**=1 is supported.
2. If the network supports IPv4, then only IPv4 DNS addresses can be set.
3. If the network supports IPv6, then only IPv6 DNS addresses can be set.
4. The DNS server address should be configured after IP address URC (e.g. **+IP: 10.18.237.42**, indicating successfully registered to network) is reported.

2.3.12. AT+QICFG Configure Optional Parameters

This command is used to configure optional parameters for TCP/IP functionalities.

AT+QICFG Configure Optional Parameters	
Test Command **AT+QICFG=?**	Response **+QICFG: "dataformat",(**list of supported **<send_data_format>**s**),(**list of supported **<recv_data_format>**s**)** **+QICFG: "viewmode",(**list of supported **<view_mode>**s**)** **+QICFG: "showlength",(**list of supported **<show_length_mode>**s**)** **+QICFG: "echomode",(**list of supported **<echo_mode>**s**)** **OK**
Write Command Set the data format for sending and receiving	Response If **<send_data_format>** and **<recv_data_format>** are omitted, query the current configuration:

AT+QICFG="dataformat"[,<send_data_format>,<recv_data_format>]	**+QICFG: "dataformat",<send_data_format>,<recv_data_format>** **OK** If **<send_data_format>** and **<recv_data_format>** are specified, configure the data format for sending and receiving: **OK** If there is an error related to ME functionality: **ERROR**
Write Command Set the output format of received data **AT+QICFG="viewmode"[,<view_mode>]**	Response If **<view_mode>** is omitted, query the current configuration: **+QICFG: "viewmode",<view_mode>** **OK** If **<view_mode>** is specified, configure output format of the received data: **OK** If there is an error related to ME functionality: **ERROR**
Write Command Set whether to show the optional data-length parameters[1)] in buff access mode **AT+QICFG="showlength"[,<show_length_mode>]**	Response If **<show_length_mode>** is omitted, query the current configuration: **+QICFG: "showlength",<show_length_mode>** **OK** If **<show_length_mode>** is specified, configure whether to show the optional data-length parameters in buffer access mode: **OK** If there is an error related to ME functionality: **ERROR**
Write Command Set whether to echo the input data to UART in data mode **AT+QICFG="echomode"[,<echo_mode>]**	Response If **<echo_mode>** is omitted, query the current configuration: **+QICFG: "echomode",<echo_mode>** **OK**

	If **<echo_mode>** is specified, configure whether to echo the input data to UART in data mode: **OK** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms
Characteristics	These commands take effect immediately. 1. The configurations of **<send_data_format>**, **<recv_data_format>**, **<view_mode>** and **<show_length_mode>** will be saved to NVRAM automatically. Remain valid after deep-sleep wakeup. 2. The configuration of **<echo_mode>** will not be saved to NVRAM. Invalid after deep-sleep wakeup.

Paramete

<send_data_format>	Integer type. Sending data format. 0 Text mode 1 Hex mode
<recv_data_format>	Integer type. Receiving data format. 0 Text mode 1 Hex mode
<view_mode>	Integer type. Received data output format. 0 Received data output format: data header\r\ndata 1 Received data output format: data header,data
<show_length_mode>	Integer type. Whether to show the optional data length parameters in buffer access mode. 0 Do not show them in buffer access mode 1 Show them in buffer access mode
<echo_mode>	Integer type. Whether to echo the input data to UART in data mode 0 Do not echo the input data to UART 1 Echo the input data to UART

NOTES

1. [1)] The optional data-length parameters are:
   - **<current_recv_length>** in URC **+QIURC: "recv",<connectID>[,<current_recv_length>]**
   - **<remaining_length>** in the response of **AT+QIRD**
2. **<echo_mode>** is only valid for data mode transferring.

2.3.13. AT+QIGETERROR Query the Last Error Code

This command is used to query the **<err>** code and specific description of the **<err>** code returned by the last TCP/IP command.

AT+QIGETERROR Query the Last Error Code	
Test Command **AT+QIGETERROR=?**	Response **OK**
Execution Command **AT+QIGETERROR**	Response **+QIGETERROR: <err>,<errcode_description>** **OK** If there is an error related to ME functionality: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<err>	Integer type. The error code. Please refer to ***Chapter 3*** for details.
<errcode_description>	A string parameter indicates the details of error information. Please refer to ***Chapter 3*** for details of **<err>** codes and corresponding description.

2.4. Description of URCs

The URC of TCP/IP AT commands will be reported in the following format: **<CR><LF>+QIURC: <type>[…]<CR><LF>**. For convenience, **<CR><LF>** at the beginning and end of each URC is omitted intentionally.

NOTES

1. When the module is in PSM, URCs will not be reported.
2. When the module is in DRX/eDRX mode, there will be a delay in URC reporting and the time delay depends on the paging cycle.
3. When the module is in connected mode, URCs will be reported promptly.
4. The maximum length of each URC is 1400 bytes.
5. When receiving data in UDP direct push mode, the module will discard URCs exceeding 1400 bytes.

2.4.1. URC Indicating Connection Closed

When TCP socket service is closed by a remote peer or due to network error, the URC **+QIURC: "closed",<connectID>** will be outputted, and the **<socket_state>** (indicating the status of the socket service) will be changed to "closing". The host must execute **AT+QICLOSE=<connectID>** to change the **<socket_state>** to "initial".

In buffer access mode, the host can also execute **AT+QIRD=<connextID>,<read_length>** to read the buffer data.

URC Indicating Connection Closed	
+QIURC: "closed",<connectID>	Indicating socket service connection closed.

Parameter

<connectID>	Integer type. The socket service index. The range is 0-4.

2.4.2. URC Indicating Incoming Data

In buffer access mode or direct push mode, the module will report URC to the host when data is received from the server.

In buffer access mode, the URC format is:
+QIURC: "recv",<connectID>[,<current_recv_length>]

In direct push mode, the URC format is:
+QIURC: "recv",<connectID>,<current_recv_length><CR><LF><data>

URC Indicating Incoming Data	
+QIURC: "recv",<connectID>,[<current_recv_length>]	Indicating incoming data in buffer access mode.
+QIURC: "recv",<connectID>,<current_recv_length><CR><LF><data>	Indicating incoming data in direct push mode.

Parameter

<connectID>	Integer type. The socket service index. The range is 0-4.
<current_recv_length>	Integer type. The length of actual received data.
<data>	The received data.

NOTE

After configuring **AT+QICFG="showlength",1** (refer to ***Chapter 2.3.12***), the buffer access mode will contain the optional parameter only if the downstream data reported has reached URC.

2.4.3. URC Indicating Incoming Data Buffer Full

In buffer access mode, if no resources can be allocated for incoming data, the module will report the following URC.

URC Indicating Incoming Data Buffer Full	
+QIURC: "recv",<connectID>,"buff full"	Indicating the incoming data buffer is full.

Parameter

<connectID>	Integer type. The socket service index. The range is 0-4.

NOTE

The maximum node number allocated for buffer access mode is 20.

3 Summary of <err> Codes

If **<err>** is returned after executing TCP/IP AT commands, the details of errors can be queried via **AT+QIGETERROR**. Please note that **AT+QIGETERROR** just returns the **<err>** code of the last TCP/IP AT command.

Table 2: Summary of Error Codes

<err> Code	Description of Error Code
0	Operation successful
550	Unknown error
551	Operation blocked
552	Invalid parameters
553	Memory not enough
554	Create socket failed
555	Operation not supported
556	Socket bind failed
557	Socket listen failed
558	Socket write failed
559	Socket read failed
560	Socket accept failed
561	Open PDP context failed
562	Close PDP context failed
563	Socket identity has been used
564	DNS busy

565	DNS parse failed
566	Socket connection failed
567	Socket has been closed
568	Operation busy
569	Operation timeout
570	PDP context broken down
571	Cancel send
572	Operation not allowed
573	APN not configured
574	Port busy

4 Examples

4.1. TCP Client Works in Buffer Access Mode

4.1.1. Set up a TCP Client Connection and Enter Buffer Access Mode

```
//Open a socket service, and the context ID is 1 and socket service index is 0.
AT+QIOPEN=1,0,"TCP","220.180.239.212",8062,1234,0
OK

+QIOPEN: 0,0                    //Connected successfully. It is recommended to wait for 60 s for the URC
                                to be reported.
AT+QISTATE=1,0                  //Query the connection status of socket service 0.
+QISTATE: 0,"TCP","220.180.239.212",8062,1234,2,1,0

OK
```

4.1.2. Send Data in Buffer Access Mode

```
AT+QISEND=0,10,1234567890           //Send data, and the data length is 10 bytes.
OK

SEND OK
AT+QISEND=0                         //Send variable-length data.
>
1234567890<ctrl+Z>
OK

SEND OK
AT+QISEND=0,10                      //Send fixed-length data and the data length is 10 bytes.
>
1234567890
OK

SEND OK
AT+QISENDEX=0,5,3031323334          //Send hex string data.
```

```
OK

SEND OK
```

4.1.3. Receive Data from Remote Server in Buffer Access Mode

```
+QIURC: "recv",0          //Socket service 0 received data.
AT+QIRD=0,512             //Read data, and the data length is 512 bytes.
+QIRD: 10
1234567890

OK
AT+QIRD=0,512             //Read data, and the data length is 512 bytes.
+QIRD: 0                  //No data in buffer.

OK
AT+QICFG="showlength",1   //Enable to show optional parameters <current_recv_length>
                          and <remaining_length> in buff access mode.
OK

+QIURC: "recv",0,12       //Socket service 0 has received data, and the data length is 12 bytes.
AT+QIRD=0,10              //Read data, and the data length is 10 bytes.
+QIRD: 10,2               //10 bytes have been read, and 2 bytes remained.
1234567890

OK

+QIURC: "recv",0,"buff full"   //Socket service 0 indicates that the buffer is full, and the host
                               has to use AT+QIRD to read the buffer data.
AT+QICFG="viewmode",1          //Received data output format: data header,data
OK
AT+QISEND=0,12,"012345678901"
OK

SEND OK

+QIURC: "recv",0,12
AT+QIRD=0,10
+QIRD: 10,2,0123456789

OK
```

4.1.4. Close a Connection

```
AT+QICLOSE=0        //Close a connection whose socket service index is 0.
OK

CLOSE OK
```

4.2. TCP Client Works in Direct Push Mode

4.2.1. Set up a TCP Client Connection and Enter Direct Push Mode

```
//Open a socket service, and the context ID is 1 and socket service index is 0.
AT+QIOPEN=1,0,"TCP","220.180.239.212",8062,0,1
OK

+QIOPEN: 0,0                    //Connected successfully. It is recommended to wait for 60 s for the URC
                                to be reported.
AT+QISTATE=1,0                  // Query the connection status of socket service 0.
+QISTATE: 0,"TCP","220.180.239.212",8062,0,2,1,1

OK
```

4.2.2. Send Data in Direct Push Mode

```
AT+QISEND=0,5,12345                 //Send data, and the data length is 5 bytes.
OK

SEND OK
AT+QISEND=0                         //Send variable-length data.
>
12345<Ctrl+Z>
OK

SEND OK
AT+QISEND=0,5                       //Send fixed-length data and the data length is 5 bytes.
>
12345
OK

SEND OK
AT+QISENDEX=0,5,3132333435          //Send hex string data.
```

```
OK

SEND OK
```

4.2.3. Receive Data from Remote Server in Direct Push Mode

```
+QIURC: "recv",0,5                    //Receive data from remote server.
12345
AT+QICFG="viewmode",1                 //Received data output format: data header,data
OK
AT+QISEND=0,12,"012345678901"
OK

SEND OK

+QIURC: "recv",0,12,012345678901
```

4.2.4. Close a Connection

```
AT+QICLOSE=0        //Close a connection whose socket service index is 0.
OK

CLOSE OK
```

4.3. Ping a Remote Server

```
AT+QPING=1,"iot.quectel.com"

OK

+QPING: 0,"47.100.63.174",32,560,88

+QPING: 0,"47.100.63.174",32,220,88

+QPING: 0,"47.100.63.174",32,230,88

+QPING: 0,"47.100.63.174",32,280,88

+QPING: 0,4,4,0,220,560,322
```

4.4. Synchronize Local Time

```
AT+QNTP=1,"ntp5.aliyun.com"          //Synchronize local time with NTP server ntp5.aliyun.com.
OK

+QNTP: 0,"19/06/11,11:08:20:35+32"
```

4.5. Configure DNS Server

```
AT+QIDNSCFG=1,"218.2.2.2","8.8.8.8"
OK
AT+QIDNSCFG=1
+QIDNSCFG: 1,"218.2.2.2","8.8.8.8"

OK
```

4.6. Getting Last Error Code

```
//Open a socket service without specifying socket service index.
AT+QIOPEN=1,"UDP","220.180.239.212",8063,0,1
ERROR
AT+QIGETERROR
+QIGETERROR: 552,invalid parameters

OK
```

5 Appendix A Reference

Table 3: Terms and Abbreviations

Abbreviation	Description
eDRX	extended Discontinuous Reception
DNS	Domain Name System
DRX	Discontinuous Reception
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ME	Mobile Equipment
NTP	Network Time Protocol
NVRAM	Non-Volatile Radon Access Memory
PPP	Point to Point Protocol
PSM	Power Saving Mode
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URC	Unsolicited Result Code
UTC	Coordinated Universal Time