BC65&BC92&BC95-GR TCP/IP Application Note

NB-IoT Module Series

Version: 1.2

Date: 2023-10-20

Status: Released

Legal Notices

Use and Disclosure Restrictions

License Agreements

Copyright

About the Document

Revision History

Version	Date	Author	Description
1.0	2020-07-07	Quinten SONG	Initial
1.1	2021-03-04	Albert ZHANG /Quinten SONG	1. Added the TCP transparent transmission mode (<access_mode>=2). 2. Updated AT+QIDNSGIP to support the resolution of IPv6 domain names. 3. Updated AT+QIDNSCFG to support the configuration of IPv6 DNS Server. 4. Added <incoming_num> for AT+QIOPEN. 5. Added a protocol type for AT+QIOPEN to support automatic protocol adaptation.
1.2	2023-10-20	Miles MA	1. Added the applicable module BC95-GR. 2. Added a note of <incoming_num> (Chapter 2.2.1).

Contents

Table Index

1 Introduction

Quectel BC65, BC92 and BC95-GR modules feature an embedded TCP/IP stack, which enables the host to access the Internet directly via AT commands, thus greatly reducing the dependence on the external PPP and TCP/IP protocol stack and minimizing the cost.

BC65, BC92 and BC95-GR modules provide three types of socket services: TCP client, UDP client, and TCP listener.

1.1. Usage of TCP/IP AT Commands

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

1.2. Description of Data Access Modes

- BC65 and BC95-GR support two data access modes: buffer access mode and direct push mode;
- BC92 supports three data access modes: buffer access mode, direct push mode, and transparent transmission mode.

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

- In buffer access mode, the module buffers data upon receiving it and reports a URC in the format of **+QIURC: "recv",<connectID>[,<current_recv_length>]**. The host can read the buffered data with **AT+QIRD**.

- In direct push mode, the received data is outputted directly in this URC: **+QIURC: "recv",<connectID>,<current_recv_length><CR><LF><data>**.

- In transparent transmission mode, the corresponding UART port enters exclusive mode, where data received from the port is sent to the Internet directly, and data received from the Internet is outputted via the port directly.

- **Exit from transparent transmission mode**

To exit from transparent transmission mode, execute **+++** and wait until **OK** is returned. To prevent the **+++** from being misinterpreted as data, please do as follows:

1) Do not input any character within 500 ms before and after inputting **+++**.
2) Input **+++** within 500 ms.

- **Return to transparent transmission mode**

1) By **AT+QISWTMD**. Specify **<access_mode>** as 2 when executing this command. When transparent transmission mode is entered successfully, **CONNECT** will be returned.
2) By **ATO0**. After a connection exits from transparent transmission mode, executing **ATO0** will switch the data access mode back to transparent transmission mode. When transparent transmission mode is entered successfully, **CONNECT** will be returned.

NOTE

1. 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. You can use **AT+QICFG** to configure whether to display **<current_recv_length>**.

1.3. 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.

2 Description of TCP/IP 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 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.

2.1.2. 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: Type of AT Commands

Command Type	Syntax	Description
Test Command	**AT+<cmd>=?**	Test the existence of the corresponding command and return information about the type, value, or range of its parameter.
Read Command	**AT+<cmd>?**	Check the current parameter value of a corresponding command.
Write Command	**AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]**	Set user-definable parameter value.
Execution Command	**AT+<cmd>**	Return a specific information parameter or perform a specific action.

2.2. Related AT Commands

2.2.1. AT+QIOPEN Open a Socket Service

This command opens a socket service. The service type can be specified by **<service_type>**, and the data access mode by **<access_mode>**. The URC **+QIOPEN: <connectID>,<result>** is reported to indicate whether the socket service is 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/TCP LISTENER”,"<IP_address>/<domain_name>",<remote_port>[,<local_port>[,**(range of supported **<access_mode>**s)**[,**(range of supported **<protocol_type>**s)**[,(list of supported <incoming_num>s)]]]]** **OK**
Write Command **AT+QIOPEN=<contextID>,<connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>[,<local_port>[,<access_mode>[,<protocol_type>[,<incoming_num>]]]]**	Response **OK** **+QIOPEN: <connectID>,<result>** If **<access_mode>** is 2 and **<service_type>** is "TCP" and the connection is successful: **OK** **CONNECT** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately; Remain valid after the module is wakened up from deep sleep mode. The configurations of **<IP_address>/<domain_name>**, **<remote_port>**, **<local_port>** and **<protocol_type>** are not saved to NVRAM. If **<access_mode>** is 0 or 1, the configuration of **<access_mode>** is saved to NVRAM; If **<access_mode>** is 2, the configuration of **<access_mode>** is not saved to NVRAM.

Parameter

<contextID>	Integer type. Context ID. Range: 1–3.
<connectID>	Integer type. Socket ID. Range: 0–5.
<service_type>	String type. Socket service type. "TCP" TCP client "UDP" UDP client "TCP LISTENER" TCP server listening for incoming TCP connections
<IP_address>	String type. IP address of remote server, such as "220.18.23.22". The maximum size is 50 bytes.
<domain_name>	String type. Domain name address of the remote server. The maximum size is 50 bytes.
<remote_port>	Integer type. Port number of the remote server. Only valid when **<service_type>** is "TCP" or "UDP". Range: 1–65535.
<local_port>	Integer type. Local port number. Maximum: 65535. If **<service_type>** is "TCP LISTENER", this parameter must be specified. Range: 1–65535; If **<service_type>** is "TCP" or "UDP", this parameter can be omitted and the default is 0, which indicates that the local port will be assigned automatically.
<access_mode>	Integer type. Data access mode of socket services. 0 Buffer Access Mode 1 Direct Push Mode 2 Transparent Transmission Mode
<protocol_type>	Integer type. Internet protocol type. 0 IPv4 1 IPv6 2 IPv4v6
<incoming_num>	Integer type. The numbers of clients which the module can accept when it works in TCP connection. Range: 1–5. Default: 1.
<result>	Integer type. The result code. See ***Chapter 3*** for details. 0 Socket opened successfully Others See ***Chapter 3*** for details

NOTE

1. It is recommended to wait for (140 + **<open_time>**) seconds for the URC **+QIOPEN: <connectID>,<result>** if the connection has been established with **<domain_name>**, and to wait for **<open_time>** seconds for the URC if the connection has been established with **<IP_address>**.
2. When the module wakes up from deep sleep, to resume a TCP connection, you need to re-open the socket with **AT+QIOPEN**; while in the case of a UDP connection, the module can resume its previous connection state and send/receive data directly without re-opening the socket. In the latter case, the module will report error if you try to re-open the socket.

3. If **<service_type>** is "TCP" or "UDP", it is recommended to configure **<local_port>** to 0 to make the local port automatically allocated by the module. If it is required to set **<local_port>** to another value, avoid using port values between 1000 and 1500. Besides, an already specified local port cannot be reused.
4. If **<local_port>** is set to a specific local port number, after the socket is closed with **AT+QICLOSE**, it is recommended to wait for 120 seconds before reusing **AT+QIOPEN** to open a new socket on the same local port specified.
5. In a UDP connection, **<access_mode>** cannot be set to 2.
6. If **<service_type>** is "TCP LISTENER" and **<access_mode>** is 2, **<incoming_num>** shall be 1, after the client successfully connects to the module, **+QIURC:" incoming"** will be output, then the module enters TCP transmission mode.
7. If **<service_type>** is "TCP LISTENER", **<IP_address>** must be a local IP address, otherwise an error will be reported.
8. **<incoming_num>** is only applicable for the module BC92.

2.2.2. AT+QICLOSE Close a Socket Service

This command closes the 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 any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately.

Parameter

<connectID>	Integer type. Socket ID. Range: 0–5.

2.2.3. AT+QISTATE Query Socket Service Status

This command queries 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: **[+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>]** **[…]** **OK** If there is any error: **ERROR**
Write Command If **<query_type>** is 0, check the connection status of a specified context: **AT+QISTATE=<query_type>,<contextID>**	Response **[+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>]** **[…]** **OK** If there is any error: **ERROR**
Write Command If **<query_type>** is 1, check the connection status of a specified socket service: **AT+QISTATE=<query_type>,<connectID>**	Response **[+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>]** **OK** If there is any error: **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. Range: 1–3.
<connectID>	Integer type. Socket ID. Range: 0–5.
<service_type>	String type. Service type. "TCP" TCP client "UDP" UDP client "TCP LISTENER" TCP server listening for TCP incoming connections "TCP INCOMING" TCP connection accepted by a TCP server
<IP_address>	String type. IP address. If **<service_type>** is "TCP" or "UDP", it is the IP address of the remote server. If **<service_type>** is "TCP LISTENER", it is the local IP address. If **<service_type>** is "TCP INCOMING", it is the IP address of the remote client.
<remote_port>	Integer type. Port number. If **<service_type>** is "TCP" or "UDP", it is the port number of the remote server. If **<service_type>** is "TCP LISTENER", it is invalid and the value is always 0. If **<service_type>** is "TCP INCOMING", it is the port number of the remote client.
<local_port>	Integer type. The assigned local port number. Range: 0–65535. 0 The local port is assigned automatically.
<socket_state>	Integer type. Socket service state. 0 "idle": the client connection has not been established 1 "connecting": the client is connecting 2 "connected": the client connection has been established 3 "closing": the client connection is closing 4 "remote closing": the server connection is closing
<access_mode>	Integer type. Data access mode. 0 Buffer Access Mode 1 Direct Push Mode 2 Transparent Transmission Mode

2.2.4. AT+QISEND Send Text String Data

This command sends socket data in text string format via the specified **<connectID>**.

AT+QISEND Send Data	
Test Command **AT+QISEND=?**	Response **+QISEND: (**range of supported **<connectID>**s**),(**range of supported **<send_length>**s**),"<data>"**,(range of supported **<rai_mode>**s) **OK**
Write Command Send fixed-length data in non-data mode **AT+QISEND=<connectID>,<send_length>,<data>[,<rai_mode>]**	Response If the connection has been established and the data is sent successfully: **OK**

	SEND OK If the connection has been established but the sending buffer is full or the data fails to be sent: **OK** **SEND FAIL** If the connection has not been established, or has been abnormally closed, or the parameter is incorrect: **ERROR**
Write Command Send variable-length data in data mode **AT+QISEND=<connectID>** After the response **>**, the module enters data mode. After that, input the data to be sent. Tap **CTRL** + **Z** to send the data, or **Esc** to cancel the operation.	Response **>** If the connection has been established and the data is sent successfully: **OK** **SEND OK** If the connection has been established but the sending buffer is full or the data fails to be sent: **OK** **SEND FAIL** If the connection has not been established, or has been abnormally closed, or the parameter is incorrect: **ERROR**
Write Command Send fixed-length data in data mode **AT+QISEND=<connectID>,<send_length>** After the response **>**, the module enters data mode. After that, input the data to be sent until the data length equals to **<send_length>**	Response **>** If the connection has been established and the data is sent successfully: **OK** **SEND OK** If the connection has been established but the sending buffer is full: **OK** **SEND FAIL**

	If the connection has not been established, or has been abnormally closed, or the parameter is incorrect: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<connectID>	Integer type. Socket ID. Range: 0–5.
<send_length>	Integer type. Length of the data to be sent. The maximum value is 1440. Unit: byte.
<data>	Text format. The data to be sent. In non-data mode, data is sent in fixed lengths and the content needs to be enclosed in double quotes, such as "data". While in the data mode, data can be sent in variable lengths and the content does not need to be enclosed in double quotes.
<rai_mode>	Integer type. Release assistance indication. Range: 0–2. 0 Do not use release assistance indication. 1 Request the core network to release RRC connection immediately after receiving an uplink data packet. 2 Request the core network to release RRC connection immediately after sending a downlink data packet.

NOTE

1. **SEND OK** only indicates that the data has arrived at the protocol stack.
2. **<data>** can be sent successfully only when **<service_type>** of the socket checked out with **AT+QISTATE** is "TCP", "UDP" or "TCP INCOMING"; if it is "TCP LISTENER", **ERROR** is returned.
3. In data mode, after **>** is returned, if the sent data is empty, **SEND FAIL** is returned immediately after you tap **CTRL** + **Z**.

2.2.5. AT+QISENDEX Send Hex String Data

This command sends 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>"**,(range of supported **<rai_mode>**s) **OK**

Write Command Send fixed-length data in non-data mode **AT+QISENDEX=<connectID>,<send_length>,<hex_string>[,<rai_mode>]**	Response If the connection has been established and the data is sent successfully: **OK** **SEND OK** If the connection has been established but the sending buffer is full or the data fails to be sent: **OK** **SEND FAIL** If the connection has not been established, or has been abnormally closed, or the parameter is incorrect: **ERROR**
Write Command Send variable-length data in data mode **AT+QISENDEX=<connectID>** After the response **>**, the module enters data mode. After that, input the hex string data to be sent. Tap **Ctrl** + **Z** to send the data, or tap **Esc** to cancel the operation.	Response **>** **OK** If the connection has been established and the data is sent successfully: **SEND OK** If the connection has been established but the sending buffer is full or the data fails to be sent: **SEND FAIL** If the connection has not been established, or has been abnormally closed, or the parameter is incorrect: **ERROR**
Write Command Send fixed-length data in data mode **AT+QISENDEX=<connectID>,<send_length>** After response **>**, the module enters data mode. After that, input the hex string data to be sent until the data length equals to **<send_length>**.	Response **>** If the connection has been established and the data is sent successfully: **OK** **SEND OK** If the connection has been established but the sending buffer is full: **OK**

	SEND FAIL If the connection has not been established, or has been abnormally closed, or the parameter is incorrect: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<connectID>	Integer type. Socket ID. Range: 0–5.
<send_length>	Integer type. Length of the data to be sent. The maximum value is 1440. Unit: byte.
<hex_string>	Hex string type. The data to be sent. In non-data mode, data is sent in fixed lengths and the content needs to be enclosed in double quotes, such as "3031323334". In data mode, data can be sent in variable lengths and the content does not need to be enclosed in double quotes.
<rai_mode>	Integer type. Release assistance indication. Range: 0–2. 0 Do not use release assistance indication. 1 Request the core network to release RRC connection immediately after receiving an uplink data packet. 2 Request the core network to release RRC connection immediately after sending a downlink data packet.

NOTE

1. **SEND OK** only indicates that the data arrives at the protocol stack.
2. **<hex_string>** can be sent successfully only when **<service_type>** of socket checked out with **AT+QISTATE** is "TCP", "UDP" or "TCP INCOMING"; if it is "TCP LISTENER", **ERROR** will be returned.
3. In data mode, after **>** is returned, if the sent data is empty, **SEND FAIL** will be returned immediately after you tap **CTRL** + **Z**.

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

This command reads the socket data received from a specified connection.

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

AT+QIRD Retrieve the Received TCP/IP Data	
Test Command	Response

AT+QIRD=?	**+QIRD: (**range of supported **<connectID>**s)**,(**range of supported **<read_length>**s**)** **OK**
Write Command When the service type is "TCP", "UDP" or "TCP INCOMING" **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 any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<connectID>	Integer type. Socket ID. Range: 0–5.
<read_length>	Integer type. The specified length of data to be retrieved. Range: 1–1024; Unit: byte.
<actual_read_length>	Integer type. The actual length of retrieved data. Unit: byte.
<remaining_length>	Integer type. The remaining length of the last received data. Unit: byte.
<data>	String type. The retrieved data.

NOTE

1. If the module receives data when the receive buffer is not empty, it will not report a new URC until all the received data has been retrieved from the buffer.
2. When **AT+QICFG="showlength",1** is configured, **<current_recv_length>** is included in the URC **+QIURC: "recv",<connectID>,[<current_recv_length>]**, and **<remaining_length>** in the response **AT+QIRD=<connectID>,<read_length>**. See ***Chapter 2.2.11***.
3. The remaining length is not of the total bytes received and buffered, but only of the data currently stored in one node.

2.2.7. AT+QISWTMD Switch Data Access Modes

This command switches the connection among buffer access mode, direct push mode and transparent transmission mode. When starting a new socket service, you can specify the data access mode by **<access_mode>** of **AT+QIOPEN**. After a socket is opened, you can use **AT+QISWTMD** to switch the specified data access mode to another.

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 **<access_mode>** is 2 and the connection is established: **OK** **CONNECT** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. Remain valid after the module is wakened up from deep sleep mode. The configuration is saved to NVRAM.

Parameter

<connectID>	Integer type. Socket ID. Range: 0–5.
<access_mode>	Integer type. The data access modes of the socket service. 0 Buffer Access Mode 1 Direct Push Mode 2 Transparent Transmission Mode

NOTE

1. When **<access_mode>** is 2, data in NVRAM will not be saved.

2. When **<access_mode>** is 2, UDP connections cannot be switched to.
3. The **<access_mode>** of a listening socket cannot be changed to 2 during the monitoring process.

2.2.8. AT+QPING Ping a Remote Host

This command tests the reachability of a remote host on an Internet Protocol network.

AT+QPING Ping a Remote Host	
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 a remote host is pinged successfully: **OK** **+QPING: <ping_result>[,<IP_address>,<bytes>,<time>,<TTL>]** **[…]** **+QPING: <finresult>[,<sent>,<rcvd>,<lost>,<min>,<max>,<avg>]** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<contextID>	Integer type. Context ID. Range: 1–3.
<host>	String type. The host address in string type. It is a domain name or a dotted decimal IP address.
<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 number of ping requests. Range: 1–10. Default: 4.
<ping_size>	Integer type. The payload size of a ping request. Range: 32–200. Unit: byte. Default: 32.
<ping_result>	Integer type. The result of each ping request.

	0	Received ping response from the remote server. **<IP_address>**,**<bytes>**,**<time>**,**<TTL>** is displayed
	1	Ping timeout.
	Others	Refer to ***Chapter 3*** for specific result codes.
<IP_address>	String type. The IP address of the remote server in dotted decimal notation.	
<bytes>	Integer type. The length of each ping request. Unit: byte.	
<time>	Integer type. The round-trip time of the ping request. Unit: millisecond.	
<TTL>	Integer type. The time to live value of the ping request.	
<finresult>	Integer type. The final result of the ping operation.	
	2	Ping successful
	Others	Refer to ***Chapter 3*** for specific result codes.
<sent>	Integer type. The total number of bytes sent in ping requests.	
<rcvd>	Integer type. The total number of bytes received in ping responses.	
<lost>	Integer type. The total number of bytes lost in ping requests.	
<min>	Integer type. The minimum response time. Unit: millisecond.	
<max>	Integer type. The maximum response time. Unit: millisecond.	
<avg>	Integer type. The average response time. Unit: millisecond.	

2.2.9. AT+QNTP Synchronize Local Time via NTP Server

This command synchronizes the local time with the Coordinated Universal Time (UTC) via the NTP server.

AT+QNTP Synchronize Local Time via 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 it is successfully synchronized: **OK** **+QNTP: <NTPtimesync_result>,<time>** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<contextID>	Integer type. Context ID. Range: 1–3.
<server>	String type. The address of the NTP server.
<port>	Integer type. The port of the NTP server.
<auto_set_time>	Integer type. Whether to automatically synchronize local time. 0 Not synchronize automatically 1 Synchronize automatically
<NTPtimesync_result>	Integer type. The NTP time synchronization related result code. 0 Time synchronization succeeded. 1 Time synchronization failed. Others Refer to ***Chapter 3*** for specific NTP time synchronization result codes.
<time>	String type. The time synchronized via the NTP server. The format is "YYYY/MM/DD,hh:mm:ss±zz". The range of "zz" is -48 to 56.

NOTE

When **<auto_set_time>** is set to 1, RTC is updated with the synchronized time automatically.

2.2.10. AT+QIDNSGIP Get IP Address by Domain Name

This command converts a specified domain name to its IP address(es).

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",<result>,<IP_count>,<DNS_ttl>** **[+QIURC: "dnsgip",<hostIPaddr>]** **[…]** If there is any error: **ERROR**
Maximum Response Time	300 ms

Characteristics	/

Parameter

<contextID>	Integer type.	Context ID. Range: 1–3.
<hostname>	String type.	Domain name. The maximum size is 50 bytes.
<IP_count>	Integer type.	The number of the IP address(es) corresponding to **<hostname>**.
<DNS_TTL>	Integer type.	The time to live of the DNS.
<hostIPaddr>	String type.	An/The IP address of **<hostname>**.
<result>	Integer type.	The result code. 0 Get IP address successfully Others See ***Chapter 3*** for details

2.2.11. AT+QICFG Configure Optional Parameters

The command configures 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: "open_time",(**range of supported **<open_time>**s**)** **OK**
Write Command **AT+QICFG="dataformat"[,<send_data_format>,<recv_data_format>]**	Response If the optional parameters are omitted, query the current setting. **+QICFG: "dataformat",<send_data_format>,<recv_data_format>** **OK** If the optional parameters are specified, set the data format for sending and receiving. **OK** If there is any error: **ERROR**
Write Command	Response If the optional parameter is omitted, query the current setting.

AT+QICFG="viewmode"[,<view_mode>]	**+QICFG: "viewmode",<view_mode>** **OK** If the optional parameter is specified, set the output format of the received data. **OK** If there is any error: **ERROR**
Write Command **AT+QICFG="showlength"[,<show_length_mode>]**	Response If the optional parameter is omitted, query the current setting. **+QICFG: "showlength",<show_length_mode>** **OK** If the optional parameter is specified, set whether to show the optional data length related parameters in buffer access mode. **OK** If there is any error: **ERROR**
Write Command **AT+QICFG="open_time"[,<open_time>]**	Response If the optional parameter is omitted, query the current setting. **+QICFG: "open_time",<open_time>** **OK** If the optional parameter is specified, set the TCP connection timeout time. **OK** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	Take effect immediately; The configurations of such parameters as **<send_data_format>**, **<recv_data_format>**, **<view_mode>** and **<show_length_mode>** are saved to NVRAM automatically and remain valid after the module is wakened up from deep sleep. However, the configuration of **<open_time>**

	is not saved to NVRAM automatically. Thus, its value is invalid after the module is wakened up from deep sleep mode.

Parameter

<send_data_format>	Integer type. The format of data to be sent. 0 Text string 1 Hex string
<recv_data_format>	Integer type. The format of data to be received. 0 Text mode 1 Hex mode
<view_mode>	Integer type. The output format of received data. 0 data header\r\ndata 1 data header,data
<show_length_mode>	Integer type. Whether to show the optional data length related parameters in buffer access mode. 0 Do not show them 1 Show them
<open_time>	Integer type. Configure the TCP connection timeout time. Range: 1–36; Default value: 36. Unit: second.

NOTE

1. Currently **<send_data_format>** can only be configured to 0; To send a data in hex string format, use **AT+QISENDEX**.
2. Optional data length related parameters include:
   - **<current_recv_length>** in the URC **+QIURC: "recv",<connectID>,[<current_recv_length>]**
   - **<remaining_length>** in the response of **AT+QIRD**.

2.2.12. AT+QIGETERROR Query the Last Error Code

This command queries the last error code and its specific description.

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 any error: **ERROR**

Maximum Response Time	300 ms
Characteristics	/

Parameter

<err>	Integer type. The error code. See ***Chapter 3*** for all possible error codes.
<errcode_description>	String type. Description of the error code. See ***Chapter 3*** for the description of each error code.

2.2.13. AT+QIDNSCFG Configure DNS Server Address

This command configures the primary and secondary DNS server addresses.

AT+QIDNSCFG Configure DNS Server Address	
Test Command **AT+QIDNSCFG=?**	Response **+QIDNSCFG: <PrimaryDNS>[,<SecondaryDNS>]** **OK**
Read Command **AT+QIDNSCFG?**	Response **+QIDNSCFG: <pridnsaddr_IPv4>,<secdnsaddr_IPv4>,<pridnsaddr_IPv6>,<secdnsaddr_IPv6>** **OK** If there is any error: **ERROR**
Write Command **AT+QIDNSCFG=<PrimaryDNS>[,<SecondaryDNS>]**	Response **OK** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. Remain valid after the module is wakened up from deep sleep mode; The configurations are not saved to NVRAM.

Parameters

<PrimaryDNS>	String type. The IP address of primary DNS server, such as "220.18.23.22". The maximum size is 50.
<SecondaryDNS>	String type. The IP address of alternate DNS server, such as "220.18.23.22". The maximum size is 50.
<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.

NOTE

The DNS server cannot be configured before the module has successfully registered to the network.

2.3. Description of URC

The URCs of TCP/IP AT commands are reported in this format: **<CR><LF>+QIURC:<type>[…]<CR><LF>**. **<CR><LF>** at the beginning and end of each URC are omitted for brevity.

2.3.1. +QIURC: "closed" URC Indicating Connection Closed

When a TCP socket service is closed by a remote peer or due to network error, the URC **+QIURC: "closed",<connectID>** is outputted.

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

Parameter

<connectID>	Integer type. Socket ID. Range: 0–5.

2.3.2. +QIURC: "recv" URC Indicating Incoming Data

In buffer access mode or direct push mode, the module reports 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>

+QIURC: "recv" 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. Socket ID. Range: 0–5.
<current_recv_length>	Integer type. The length of actually received data.
<data>	String type. The received data.

2.3.3. +QIURC: "recv" URC Indicating Incoming Data Buffer is Full

In Buffer Access Mode, if no resource can be allocated for incoming data, the module will report the following URC.

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

Parameter

<connectID>	Integer type. The Socket ID. Range: 0–5.

2.3.4. +QIURC: "incoming full" Incoming Connection Number Limit Reached

When the total number of incoming connections reaches the upper limit, or no socket system resource can be allocated, the module will report the following URC when a new incoming connection is requested.

+QIURC: "incoming full" Incoming Connection Number Limit Reached	
+QIURC: "incoming full"	Indicating incoming connections has reached the maximum number.

2.3.5. +QIURC: "incoming" URC Indicating Incoming Connection

If **<service_type>** is "TCP LISTENER", when a remote client connects to your server, the module automatically assigns a free **<connectID>** for the new connection and reports the following URC. After the new incoming connection is accepted by **<serverID>**, the allocated **<connectID>**, and the **<remoteIP>**

and **<remote_port>** are informed via this URC. The **<service_type>** of the new connection is "TCP INCOMING", and the **<access_mode>** of it is buffer access mode.

+QIURC: "incoming" URC Indicating Incoming Connection	
+QIURC: "incoming",<connectID>,<serverID>,<remoteIP>,<remote_port>	Indicating incoming connection.

Parameter

<connectID>	Integer type. The socket assigned for the incoming connection, which is automatically specified by the module. Range: 0–5.
<serverID>	Integer type. The ID of the listening socket, whose **<service_type>** is "TCP LISTENER" and which accepts the incoming **<connectID>**.
<remoteIP>	String type. Remote IP address of the incoming **<connectID>**.
<remote_port>	Integer type. Remote port of the incoming **<connectID>**.

NOTE

If the **<access_mode>** of the monitoring link (the TCP LISTENER connection) is 2, after the URC of the connection is output, the established connection automatically enters the TCP transparent transmission mode.

3 Summary of Result Codes

This chapter lists and describes all the possible result codes, including the error codes, of the TCP/IP AT commands introduced in this document. If there is any error occurred after the execution of a TCP/IP AT command, the corresponding error code along with its explanation can be queried via **AT+QIGETERROR**. Note that **AT+QIGETERROR** only returns the error code and description corresponding to the last error.

Table 2: Summary of Result Codes

<result> Code	Description
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 connect 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

Table 3: Summary of Ping Result Codes

<ping_result> Code	Description
0	PING_ECHO
1	PING_TIMEOUT
2	PING_SUCCESS
3	PING_TCPIP_ERROR
4	PING_ADDRESS_NOT_FOUND
5	PING_PDP_ACT_FAIL

Table 4: Summary of NTP Time Synchronization Result Codes

<NTPtimesync_result> Code	Description
0	APP_NTP_OK
1	APP_NTP_ERROR
2	APP_NTP_NO_REPLY

3	APP_NTP_ERROR_BUSY
4	APP_NTP_DNS_ERROR
5	APP_NTP_BEARER_FAIL

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

```
//Ensure that the module is registered to network and has obtained an IP address in advance.
AT+CGATT?                              //Query network status.
+CGATT: 1                              //Attached to the network successfully.

OK
AT+CGPADDR?                            //Query whether an IP address is obtained.
+CGPADDR: 1,"100.127.243.105"          //IP address is obtained

OK
AT+QIOPEN=1,0,"TCP","220.18.39.22",8062,1234,0     //Context ID is 1 and the socket ID is 0.
OK

+QIOPEN: 0,0                           //Connected successfully. It is recommended to wait for 36 s
                                         for the URC to be reported.
AT+QISTATE=1,0                         //Query the connection status of Socket 0.
+QISTATE: 0,"TCP","220.18.39.22",8062,1234,2,1,0

OK
```

4.1.2. Send Data in Buffer Access Mode

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

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

SEND OK
AT+QISEND=0,10,”1234567890”        //Send data, and the data length is 10 bytes.
OK
AT+QISEND=0,10,”1234567890”        //Send data again before SEND OK is returned
```

```
ERROR                              //SEND OK of the previous command has not been returned, so
                                     ERROR is returned when new data is sent.

SEND OK
```

4.1.3. Receive Data from Remote Server in Buffer Access Mode

```
+QIURC: "recv",0                   //Socket 0 received data.
AT+QIRD=0,512                      //Read the data in the buffer, and the specified length is 512 bytes.
+QIRD: 10                          //The actual length of the read data is 10 bytes.
1234567890                         //The data is "1234567890".

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

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

OK

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

OK

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

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 the connection of Socket 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

```
AT+QIOPEN=1,0,"TCP","220.18.39.22",8062,0,1     //The context ID is 1 and the socket ID is 0.
OK

+QIOPEN: 0,0                    //Connected successfully. It is recommended to wait for 36 s for the URC
                                  to be reported.
AT+QISTATE=1,0                  //Query the connection status of Socket 0.
+QISTATE: 0,"TCP","220.18.39.22",8062,0,2,1,1

OK
```

4.2.2. Send Data in Direct Push Mode

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

SEND OK
AT+QISENDEX=0,5,"3031323334"             //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 the remote server.
12345
AT+QICFG="viewmode",1                    //Received data output format: data header,data
+QICFG: "viewmode",1

OK
AT+QISEND=0,12,"012345678901"
OK
```

```
SEND OK

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

4.2.4. Close TCP Client

```
AT+QICLOSE=0                              //Close the connection of Socket 0.
OK

CLOSE OK
```

4.3. TCP Server Works in Buffer Access Mode

4.3.1. Start a TCP Server

```
//Before using AT+QIOPEN, the host should activate the context with AT+CGACT.
AT+QIOPEN=1,0,"TCP LISTENER","192.168.2.6",1,2020,0   //The context ID is 1; the Socket ID is 0.
OK

+QIOPEN: 0,0                                          //TCP server is opened successfully.
AT+QISTATE=0,1                                        //Query connection status of context 1
+QISTATE: 0,"TCP LISTENER","192.168.2.6",1,2020,2,1,0

OK
```

4.3.2. Accept TCP Incoming Connection

```
+QIURC: "incoming",1,0,"192.168.2.2",36566     //Accept a TCP connection whose service type is
                                                 "TCP incoming" and socket ID is 1.
```

4.3.3. Receive Data from Incoming Connection

```
+QIURC: "recv",1                 //Received data from remote incoming connection.
AT+QIRD=1,512                    //Read data received from the incoming connection.
+QIRD: 4                         //Actual data length is 4 bytes.
test                             //The data is "test".

OK
AT+QIRD=1,512
+QIRD: 0                         //No data in the buffer.

OK
```

4.3.4. Close a TCP Server

```
AT+QICLOSE=1                                //Close incoming connection. Depending on the network,
                                             the maximum response time is 10 s.
OK

CLOSE OK
AT+QICLOSE=0                                 //Close TCP server listening.
OK

CLOSE OK
```

4.4. Ping a Remote Server

```
AT+QPING=1,"sh.quectel.com"                  //Ping sh.quectel.com of Context 1.
OK

+QPING: 0,220.18.29.21,32,192,255
+QPING: 0,220.18.23.21,32,240,255
+QPING: 0,220.18.23.21,32,241,255
+QPING: 0,220.18.23.21,32,479,255

+QPING: 2,4,4,0,192,479,287
```

4.5. Synchronize Local Time

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

+QNTP: 0,"2018/04/20,11:08:20+32"
```

4.6. Query the Last Error Code

```
//Open a socket service without specifying the socket ID.
AT+QIOPEN=1,"UDP","220.18.39.22",8063,0,1
ERROR

AT+QIGETERROR
```

```
+QIGETERROR: 552,invalid parameters

OK
```

5 Appendix A References

Table 5: Related Documents

Document Name
[1] Quectel_BC65&BC95-GR_AT_Commands_Manual
[2] Quectel_BC92_AT_Commands_Manual

Table 6: Terms and Abbreviations

Abbreviation	Description
ACK	Acknowledgement
APN	Access Point Name
DNS	Domain Name System
HEX	Hexadecimal
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
PDP	Packet Data Protocol
PPP	Point to Point Protocol
PSM	Power Saving Mode
TA	Terminal Adaptor
TCP	Transmission Control Protocol

UDP	User Datagram Protocol
URC	Unsolicited Result Code
UTC	Universal Time Coordinated