BC65&BC92

TCP/IP Application Note

NB-IoT Module Series

Rev. BC65&BC92_TCP/IP_Application_Note_V1.0

Date: 2020-07-08

Status: Released

Our aim is to provide customers with timely and comprehensive services. 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
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

COPYRIGHT

About the Document

Revision History

Version	Date	Author	Description
1.0	2020-07-08	Quinten SONG	Initial

Contents

Table Index

1 Introduction

Quectel BC65 and BC92 modules feature an embedded TCP/IP stack, which enables the host to access the Internet directly over AT commands. This greatly reduces the dependence on external PPP and TCP/IP protocol stacks and thus minimizes the cost.

BC65 and BC92 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 the socket and send/receive data via socket service.

1.2. Description of Data Access Modes

BC65 and BC92 modules support the following two data access modes:

- Buffer Access Mode
- Direct Push Mode

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

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

- In Direct Push Mode, the data can be sent with **AT+QISEND** or **AT+QISENDEX**. The received data will be outputted directly in a URC in the following format:
  +QIURC: "recv",<connectID>[,<current_recv_length>[<CR><LF>]]<data>

NOTES

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 the **AT+QICFG** command to configure whether to display the **<current_recv_length>** parameter.

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 omitted, the new value equals its previous value or its default setting, unless otherwise specified.
- **Underline** Default setting of a parameter.

2.1.2. AT Command Syntax

The **AT** or **at** prefix must be added at the beginning of each command line. Entering **<CR>** will terminate a command line. Commands are usually followed by a response that includes **<CR><LF><response><CR><LF>**. Throughout this document, only the response **<response>** will be presented, **<CR><LF>** are omitted intentionally.

Table 1: Type 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.

2.2. 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 can be specified by **<access_mode>**. The URC **+QIOPEN: <connectID>,<err>** will be 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/UDP SERVICE","<IP_address>/<domain_name>",<remote_port>[,<local_port>[,(**range of supported **<access_mode>**s**)[,(**range 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 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>** will not be saved to NVRAM. The configuration of **<access_mode>** will be 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"	Start a TCP connection as a client
	"UDP"	Start a UDP connection as a client
	"TCP LISTENER"	Start a TCP server to listen to TCP connection
	"UDP SERVICE"	Start a UDP service
<IP_address>	String type. IP address of remote server, such as "220.18.23.22".	

<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" or "UDP SERVICE", 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
<protocol_type>	Integer type. Internet protocol type. 0 IPv4 1 IPv6
<err>	Integer type. The error code. Please refer ***Chapter 3*** for details.

NOTES

1. "UDP SERVICE" of **<service_type>** is not supported now.
2. It is recommended to wait (140 + **<open_time>**) seconds for the URC **+QIOPEN: <connectID>,<err>** if the connection has been established with **<domain_name>**, and to wait **<open_time>** seconds for the URC if the connection has been established with **<IP_address>**.
3. When the module wakes up from deep sleep, the TCP connection needs to re-open the socket with **AT+QIOPEN**; the UDP connection supports to wake up the module from deep sleep mode. It can return to the previous state before entering deep sleep without re-opening the socket and can directly send and receive data. At this time, it will report error if the same socket is opened.
4. 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 the port value between 1000 and 1500. In addition, the specified local port cannot be reused.
5. If **<local_port>** is set to the specified local port number, after the socket is closed with **AT+QICLOSE**, it is recommended to wait for 120 seconds before reusing **AT+QIOPEN**.

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 connection as a client “UDP” UDP connection as a client “TCP LISTENER” TCP server to listen to TCP connection “TCP INCOMING” TCP connection accepted by a TCP server “UDP SERVICE” UDP service
<IP_address>	String type. IP address of the remote client. If **<service_type>** is "TCP" or "UDP", it is the IP address of the remote server. If **<service_type>** is "TCP LISTENER" or "UDP SERVICE", 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 of the remote server. If **<service_type>** is “TCP” or “UDP”, it is the port of the remote server If **<service_type>** is “TCP LISTENER” or “UDP SERVICE”, the port is invalid, the value is always 0. If **<service_type>** is “TCP INCOMING”, it is the port of the remote client
<local_port>	Integer type. The assigned local port number. Range: 0–65535. 0 The local port is assigned automatically. 0 is the default of **<local_port>**
<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 5 “closed”: the client connection is closed
<access_mode>	Integer type. Data access mode. 0 Buffer Access Mode 1 Direct Push Mode

NOTE

"UDP SERVICE" of **<service_type>** is not supported now.

2.2.4. AT+QISEND Send 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>"** **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 any error: **ERROR**
Write Command Send data in variable length **AT+QISEND=<connectID>** After the response **>**, the module will enter data mode. After that, input the data to be sent. Tap **Ctrl** + **Z** to send the data, and tap **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 data in fixed length **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 connection has been established but the sending buffer is full: **OK** **SEND FAIL** If 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 or hex string format [1]. The data to be sent. In non-data mode, the data is sent in fixed length and the content needs to be enclosed in double quotes, such as "data". In data mode, the content does not need to be enclosed in double quotes.

NOTES

1. [1)] Sending of hex string data is not supported in this command currently.
2. **SEND OK** only indicates that the data has arrived the protocol stack.
3. **<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" or "UDP SERVICE", **ERROR** will be returned.
4. In data mode, after **>** is responded, if the sent data is empty, **SEND FAIL** will be returned immediately after tapping **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>"** **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 any error: **ERROR**
Write Command Send data in variable length **AT+QISENDEX=<connectID>** After the response **>**, the module will enter data mode. After that, input the hex string data to be sent. Tap **Ctrl+Z** to send the data, and tap **Esc** to cancel the operation.	Response **>** **OK** If connection has been established and the data is sent successfully: **SEND OK**

	If 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 data with fixed length **AT+QISENDEX=<connectID>,<send_length>** After response **>**, the module will enter 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 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, the data is sent in fixed length and the content needs to be enclosed in double quotes, such as "3031323334". In data mode, the content does not need to be enclosed in double quotes.

NOTES

1. **SEND OK** only indicates that the data arrives 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" or "UDP SERVICE",

ERROR will be returned.

3. In data mode, after **>** is responded, if the sent data is empty, **SEND FAIL** will be returned immediately after tapping **CTRL+Z**.

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

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

In Buffer Access Mode, after receiving data, the module will buffer 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 **AT+QIRD=?**	Response **+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 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 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>** and **<remaining_length>** will be prompted in the response of **AT+QIRD=<connectID>,<read_length>**. Please refer to ***Chapter 2.2.11***.
3. The remaining length is not the total received bytes in the buffer. It only indicates the current remaining data stored in one node.

2.2.7. AT+QISWTMD Switch Data Access Modes

This command switches 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 **<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 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 will be 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

2.2.8. AT+QPING Ping a Remote Server

This command tests 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 a remote server is pinged successfully: **OK** **+QPING: <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>	The host address in string type. The format 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 time of ping request. Range: 1–10. Default: 4.
<ping_size>	Integer type. The ping size. Range: 32–200. Unit: byte. Default: 32.
<result>	Integer type. The result of each ping request. 0 Received the ping response of remote server. **[<IP_address>,<bytes>,<time>,<ttl>]** will be displayed 1 Ping timeout. 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: 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 successfully 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: 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 through NTP Server

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

AT+QNTP Synchronize Local Time with 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: <NTP_err>,<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 set synchronized time to local time. 0 Not set 1 Set

<NTP_err>	Integer type. The NTP related error code. 0 Time synchronization succeeds. 1 Time synchronization fails. Others Refer to ***Chapter 3*** for specified NTP error codes.
<time>	String type. The time synchronized from 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 will be updated by the synchronized time automatically.

2.2.10. AT+QIDNSGIP Get IP Address by Domain Name

This command coverts 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 any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<contextID>	Integer type. Context ID. Range: 1–3.
<hostname>	String type. Domain name.
<IP_count>	Integer type. The number of the IP addresses corresponding to **<hostname>**.
<DNS_ttl>	Integer type. The time to live of the DNS.

<hostIPaddr>	String type. The IP address of **<hostname>**.
<err>	Integer type. The error code. Please refer to ***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 Set the format of data to be sent or received **AT+QICFG="dataformat"[,<send_data_format>,<recv_data_format>]**	Response **+QICFG: "dataformat",<send_data_format>,<recv_data_format>** **OK** If there is any error: **ERROR**
Write Command Set the output format of the received data **AT+QICFG="viewmode"[,<view_mode>]**	Response **+QICFG: "viewmode",<view_mode>** **OK** If there is any error: **ERROR**
Write Command Set whether to show the optional data length related parameters in Buffer Access Mode **AT+QICFG="showlength"[,<show_length_mode>]**	Response **+QICFG: "showlength",<show_length_mode>** **OK** If there is any error: **ERROR**
Write Command Set the TCP connection timeout time **AT+QICFG="open_time"[,<open_time>]**	Response **+QICFG: "open_time",<open_time>** **OK**

	If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	Take effect immediately; 1. The configurations of **<send_data_format>**, **<recv_data_format>**, **<view_mode>** as well as **<show_length_mode>** will be saved to NVRAM automatically. Remain valid after the module is wakened up from deep sleep mode. 2. **<open_time>** will not be saved to NVRAM automatically. Remain 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.

NOTES

1. Currently **<send_data_format>** can only be configured to 0; To send a data of hex string format, use the command **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 **<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 any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

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

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: <PrimaryDNS>,<SecondaryDNS>** **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 will not be saved to NVRAM.

Parameter

<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.

NOTE

The DNS server cannot be configured until the module is successfully registered to the network.

2.3. Description of URC

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.

2.3.1. 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>** of **AT+QISTATE?** (indicating the status of the socket service) will change to 3 (“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 buffered data.

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. 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 of 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 actual received data.
<data>	The received data.

2.3.3. URC Indicating that 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.

URC of 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. URC Indicating that Incoming Connection Reaches the Limit

If the incoming connection reaches the limit, or no socket system resource can be allocated, the module will report the following URC when a new incoming connection is requested.

URC Indicating Incoming Connection Full	
+QIURC: "incoming full"	Indicating the incoming connection is full.

2.3.5. URC Indicating Incoming Connection

If **<service_type>** is "TCP LISTENER", when a remote client connects to this server, the module will automatically assign a free **<connectID>** for the new connection. At this time, the module will report the following URC. When the new incoming connection is accepted by **<serverID>**, the allocated **<connectID>**, **<remoteIP>** and **<remote_port>** will be informed via this URC. The **<service_type>** of the new connection will be "TCP INCOMING", and the **<access_mode>** will be Buffer Access Mode.

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 incoming **<connectID>** accepted by the server whose **<service_type>** is "TCP LISTENER" and the listening socket ID is **<serverID>**.
<remoteIP>	String type. Remote IP address of the incoming **<connectID>**.
<remote_port>	Integer type. Remote port of the incoming **<connectID>**.

3 Summary of <err> Codes

If an error code is returned after executing TCP/IP AT commands, the details of the errors can be queried via **AT+QIGETERROR**. Please note that **AT+QIGETERROR** only returns the error 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 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

<result> Code	Description of Error Code
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 Result

<NTP_err> Code	Description of Error Code
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 into Buffer Access Mode

```
//Please ensure that the module is registered to network and 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 responded
```

```
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 first.
AT+QIOPEN=1,0,"TCP LISTENER","192.168.2.6",1,2020,0   //The context ID is 1 and 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, <service_type> is "TCP incoming", and <connectID> is 2.
```

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 with NTP server ntp5.aliyun.com.
OK

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

4.6. Getting Last Error Code

```
//Open a socket service without specifying a socket service.
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

No.	Document Name	Remark
[1]	Quectel_BC65_AT_Commands_Manual	The AT command manual of BC65 module
[2]	Quectel_BC92_AT_Commands_Manual	The AT command manual of BC92 module

Table 6: Terms and Abbreviations

Abbreviation	Description
ACK	Acknowledgement
DNS	Domain Name System
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	Universal Time Coordinated