

BC66_TCPIP

AT Commands Manual

NB-IoT Module Series

Rev. BC66_TCPIP_AT_Commands_Manual_V1.0

Date: 2018-05-02



Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.

7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local office. For more information, please visit:

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

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

<http://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

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

Copyright © Quectel Wireless Solutions Co., Ltd. 2018. All rights reserved.

About the Document

History

Revision	Date	Author	Description
1.0	2018-04-24	Oven TAO	Initial

Contents

About the Document	2
Contents	5
1 Introduction	6
1.1. The Process of Using TCPIP AT Commands	6
1.2. Description of Data Access Mode	6
2 Description of AT Command	7
2.1. AT+QIOPEN Start Socket Service	7
2.2. AT+QICLOSE Close Socket Service	8
2.3. AT+QISTATE Query Socket Service Status	8
2.4. AT+QISEND Send Data	10
2.5. AT+QIRD Retrieve the Received TCP/IP Data	10
2.6. AT+QISENDEX Send HEX String	11
2.7. AT+QISWTMD Switch Data Access Mode	11
2.8. AT+QPING Ping a Remote Server	12
2.9. AT+QNTP Synchronize Local Time with NTP Server	13
2.10. AT+QIDNSGIP Get IP Address by Domain Name	14
2.11. AT+QICFG Configure Optional Parameters	15
2.12. AT+QIGETERROR Query the Last Error Code	16
2.13. Description of URC	16
2.13.1. URC of Connection Closed	16
2.13.2. URC of Incoming Data	17
2.13.3. URC of Incoming Data Buff Full	17
3 Description of Error Code	18
4 Examples	20
4.1. TCP Client Works in Buffer Access Mode	20
4.1.1. Set up a TCP Client Connection and Enter into Buffer Access Mode	20
4.1.2. Send Data in Buffer Access Mode	20
4.1.3. Receive Data from Remote Server in Buffer Access Mode	21
4.1.4. Close Connection	21
4.2. TCP Client Works in Direct Push Mode	21
4.2.1. Set up a TCP Client Connection and Enter into Direct Push Mode	21
4.2.2. Send Data in Direct Push Mode	21
4.2.3. Receive Data from Remote Server in Direct Push Mode	22
4.2.4. Close TCP Client	22
4.3. PING	22
4.4. Synchronize Local Time	22
4.5. Example of Getting Last Error Code	23

1 Introduction

The module embeds a TCP/IP stack. Host is able to access the Internet directly over AT commands. It can reduce the dependence on the PPP and TCP/IP protocol stack and thus minimize the cost. The module provides the following socket services: TCP client, UDP client.

1.1. The Process of Using TCPIP AT Commands

Through TCPIP AT commands, host can start/close socket service and send/receive data via socket service.

1.2. Description of Data Access Mode

The TCPIP AT commands of the module include two kinds of data access modes:

1. Buffer access mode
2. Direct push access mode

When opening a socket service via AT+QIOPEN, specify the data access mode by the parameter <access_mode>. After a socket service is opened, AT+QISWTMD could be used to change the data access mode.

1. In buffer access mode, send data by AT+QISEND. When the data has been received, the module will buffer the data and report an URC as +QIURC: "recv",<connectID>. The host can read data by AT+QIRD. Note: If the buffer is not empty, the module will not report a new URC until all the received data has been read by AT+QIRD from buffer.
2. In direct push mode, send data by AT+QISEND. The received data will be outputted to COM port directly by +QIURC: "recv",<connectID>,<currentrecvlength><CR><LF><data>.

2 Description of AT Command

2.1. AT+QIOPEN Start Socket Service

Start a socket service by AT+QIOPEN. The service type can be specified by the parameter <service_type>. The data access mode (buffer access mode, direct push access mode) can be specified by parameter <accessmode>. The URC "+QIOPEN" indicates whether the socket service is started successfully.

It is suggested to wait 40 seconds for the URC response as "+QIOPEN: <connectID>,<err>". If connected fail, host must execute AT+QICLOSE=<connectID> to change the <socket_state> to "initial".

AT+QIOPEN Start Socket Service	
Test Command AT+QIOPEN=?	Response +QIOPEN: (1-3),(0-4),"TCP/UDP", "<IP_address>/<domain_name>",<remote_port>,<local_port>,<0-1>[,<0-1>] OK
Read Command AT+QIOPEN?	Response OK
Write Command AT+QIOPEN=<contextID>,<connectio nID>,<service_type>,<IP_address>/< domain_name>,<remote_port>[,<local_port>[,<access_mode>][,<protocol_type>]]	Response OK +QIOPEN: <connectID>,<err> <err> is 0 when service is started successfully, else <err> is not 0.

Parameter

<contextID>	Integer type, context ID, range is 1-3
<connectID>	Integer type, socket service index, 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, it indicates the IP address of remote server, such as "220.18.23.22"
<domain_name>	String type, the domain name address of the remote server
<remote_port>	The port of the remote server, only valid when <service_type> is "TCP" or "UDP", range is 1-65535
<local_port>	The local port, range is 1-65535. When <service_type> is "TCP" or "UDP", if <local_port> is 0, then the local port will be assigned automatically, else the local port is assigned as specified
<access_Mode>	Integer type, the data access mode of the socket services 0 Buffer access mode 1 Direct push mode
<protocol_type>	Integer type, the protocol type 0 IPv4 1 IPv6
<err>	Integer type, error code. Please refer Chapter 3 for details

2.2. AT+QICLOSE Close Socket Service

Close the specified socket service by AT+QICLOSE.

AT+QICLOSE Close Socket Service

Test Command AT+QICLOSE=?	Response OK
Write Command AT+QICLOSE=<connectID>	Response OK If closes successfully, response CLOSE OK If failed to close, response CLOSE FAIL

Parameter

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

2.3. AT+QISTATE Query Socket Service Status

AT+QISTATE can be used to query the socket service status. If the <query_type> is 0, it will return the status of all existing socket services in the context of specified <contextID>. If the <query_type> is 1, it will

return the status of specified <connectID> socket service.

AT+QISATE Query Socket Service Status

Test Command	Response
AT+QISTATE=?	OK
If <query_type> is 0, query by specified <contextID> AT+QISTATE=<query_type>,<context_ID>	Response Return the status of all existing connections List of (+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode>) OK
If <query_type> is 1, query by specified <connectID> AT+QISTATE=<query_type>,<connect ID>	Response +QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<access_mode> OK

Parameter

<query_type>	Integer type, the query type 0 Query connection status by <conetxtID> 1 Query connection by <connectID>
<contextID>	Integer type, context ID, range is 1-3
<connectID>	Integer type, socket service index, range is 0-4
<service_type>	String type, the service type "TCP" TCP connection as a client "UDP" UDP connection as a client
<IP_address>	IP address, the IP address of remote client
<remote_port>	Remote port number, the port of remote server
<local_port>	Local port number, if <local_port> is 0, local is assigned automatically
<socket_state>	Integer type, socket service state 0 "initial" connection has not been established 1 "Opening" client is connectiong 2 "Connected" client connection has been established 3 "Closing" connection is closing
<access_mode>	Data access mode 0 Buffer access mode 1 Direct push mode

2.4. AT+QISEND Send Data

If data is sent successfully, return "SEND OK", else return "SEND FAIL".

AT+QLWCLOSE Send a Deregister Request

Test Command AT+QISEND=?	Response +QISEND: (0-4),(1-700),(DATA) OK
Write Command AT+QISEND=<connectID>,<send_length>,<data>	Response OK If data is sent successfully, response SEND OK Else SEND FAIL

Parameter

<connectID>	Integer type, socket service index, range is 0-4.
<send_length>	Integer type, the length of data to be sent, which cannot exceed 700
<data>	The data to be sent.

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

This command receive data from buffer.

AT+QIRD Retrieve the Received TCP/IP Data

Test Command AT+QIRD=?	Response +QIRD: (0-4),(1-700) OK
Write Command AT+QIRD=<connectID><read_length>	Response +QIRD: <read_actual_length><CR><LF><data> OK If connection does not exist, return: ERROR

Parameter

<connectID>	Integer type, the socket service index, the range is 0-4
<read_length>	The maximum length of data to be retrieved, the range is 1-700
<read_actual_length>	The actual length of received data
<data>	The retrieved data

2.6. AT+QISENDEX Send HEX String

This command is used to send hex string data.

AT+QISENDEX Send HEX String

Test Command AT+QISENDEX=?	Response +QISENDEX: (0-4),(1-700),<hex_string> OK
Write Command AT+QISENDEX=<connectID>,<send_length>,<hex_string>	Response OK If data is sent successfully, response SEND OK Else SEND FAIL

Parameter

<connectID>	Integer type, socket service index, range is 0-4.
<send_length>	Integer type, the length of data to be sent, which cannot exceed 700
<hex_string>	The data to be sent.

2.7. AT+QISWTMD Switch Data Access Mode

AT+QISWTMD can switch the data access mode: buffer access mode, direct push mode. When starting a socket service, host can specify the data access mode by the parameter <access_mode> of AT+QIOPEN. After a socket has been started, host can change the data access mode by AT+QISWTMD.

AT+QISWTMD Swicth Data Access Mode

Test Command	Response
--------------	----------

AT+QISWTMD=?	+QISWTMD: (0-4),(0-1)
	OK
Read Command AT+QISWTMD?	Response OK
Write Command AT+QISWTMD=<connectID>,<access_mode>	Response OK

Parameter

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

2.8. AT+QPING Ping a Remote Server

AT+QPING is used to test the Internet Protocol reachability of a host. It will return the result during <timeout> and the default value of <timeout> is 4 second.

AT+QPING Ping a Remote Server

Test Command AT+QPING=?	Response +QPING: (1-3),<host>,(1-255),(1-10) OK
Write Command AT+QPING=<contextID>,<host>[,<timeout>[,<pingnum>]]	Response If ping successfully, return: OK [+QPING: <result>[,<IP_address>,<bytes>,<time>,<ttl>]<CR><LF> ...] +QPING: <finresult>[,<sent>,<rcvd>,<lost>,<min>,<max>,<avg>] Else response: ERROR

Parameter

<connectID>	Integer type, context ID, range is 1-3
-------------	----------------------------------------

<host>	The host address in string type, format is a domain name or a dotted decimal IP address
<timeout>	Integer type, set the maximum time to wait for the response of each ping request. Unit: second, range: 1-255, default: 4
<pingnum>	Integer type, set the maximum time of ping request. Range: 1-10. Default: 4
<result>	The result of each ping request 0 Received the ping response from the server. In this case, it is followed by “,<IP_address>,<bytes>,<time>,<ttl>” Others Refer Chapter 3
<IP_address>	The IP address of the remote server formatted as a dotted decimal IP
<bytes>	The length of sending each ping request
<time>	The time wait for response of the ping request. Unit: ms
<ttl>	Time To Live value of the response packet for the ping request
<finresult>	The final result of the command 0 It is finished normally. It is successful to activate the context and find the host. In this case, it is followed by “,<sent>,<rcvd>,<lost>,<min>,<avg>” Others refer to Chapter 3
<sent>	Total number of sending the ping requests
<rcvd>	Total number of the ping requests that received the response
<lost>	Total number of the ping requests that are timeout
<min>	The minimum response time. Unit: ms
<max>	The maximum response time. Unit: ms
<avg>	The average response time. Unit: ms

2.9. AT+QNTTP Synchronize Local Time with NTP Server

NTP is intended to synchronize the Coordinated Universal Time (UTC) with the time server.

AT+QNTTP Synchronize Local Time with NTP Server	
Test Command AT+QNTTP=?	Response +QNTTP: (1-3),<server>,(list of supported<port>s,(0,1)) OK
Read Command AT+QNTTP?	Response If in the process of synchronizing local time: +QNTTP: <server>,<port> OK
Write Command AT+QNTTP=<contextID>,<server>[,<po	Response If synchronize local time with NTP server successfully:

rt>][,<autosettime>]

OK

+QNTP: <err>,<time>

Else response:

ERROR

Parameter

<contextID>	Integer type, context ID, range is 1-3
<server>	String type, indicates the address of NTP server
<port>	Integer type, indicates the port of NTP server
<autosettime>	Integer type, indicates whether auto set synchronized time to local time 0 Not set 1 Set
<err>	Integer type, indicates synchronized result, please refer to Chapter 3
<time>	String type, indicates time which is synchronized from NTP server The format is "YY/MM/DD,hh:mm:ss±zz". The range of zz is -48-56

2.10. AT+QIDNSGIP Get IP Address by Domain Name

Querying the DNS.

AT+QINDSGIP Get IP Address by Domain Name

Test Command AT+QIDNSGIP=?	Response +QIDNSGIP: (1-3),<hostname> OK
Write Command AT+QINDSGIP=<contextID>,<hostna me>	Response OK ERROR The result will be returned as URC. +QIURC: "dnsgip",<err>,<IP_count>,<DNS_ttl> [...] +QIURC: "dnsgip",<hostIPAddr>]

Parameter

<contextID>	Integer type, context ID, range is 1-3
<hostname>	String type, indicates domain name

<err>	Integer type, indicates synchronized result, please refer to Chapter 3
<IP_count>	Integer type, the number of the IP address corresponding to the <hostname>
<DNS_ttl>	Integer type, the time of the DNS to live
<hostIPAddr>	String type, the IP address of <hostname>

2.11. AT+QICFG Configure Optional Parameters

AT+QICFG Configure Optional Parameters	
Test Command AT+QICFG=?	Response +QICFG: "dataformat",(0,1),(0,1) +QICFG: "viewmode",(0,1) OK
Set the data format for sending and receiving (only for non-transparent mode) AT+QICFG="dataformat"[,<send_data_format>,<recv_data_format>]	Response [+QICFG: "dataformat",<send_data_format>,<recv_data_format>] OK ERROR
Set the data format for sending and receiving (only for non-transparent mode) AT+QICFG="viewmode"[,<view_mode>]	Response [+QICFG: "viewmode",<mode>] OK ERROR

Parameter

<send_data_format>	Integer type 0 Text mode 1 Hex mode
<recv_data_format>	Integer type 0 Text mode 1 Hex mode
<view_mode>	Integer type 0 Receive data output format: data header\r\n\data 1 Receive data output format: data header,data

2.12. AT+QIGETERROR Query the Last Error Code

If TCPIP AT commands response ERROR, the details of error can be queried by AT+QIGETERROR. Please note that AT+QIGETERROR just returns error code of the last TCPIP AT command.

AT+QIGETERROR Query the Last Error Code

Test Command AT+QIGETERROR=?	Response OK
Execution Command AT+QIGETERROR	Response +QIGETERROR: <err>,<errcode_desprition> OK

Parameter

<err>	Integer type, error code. Please refer to Chapter 3
<errcode_description>	A string parameter indicates the details of error information. Please refer to Chapter 3

2.13. Description of URC

The URC of TCPIP AT commands will be reported to the host by the type of "+QIURC:". It contains the report about incoming data closed. Actually, there is "<CR><LF>" both before and after URC, but "<CR><LF>" is omitted intentionally.

2.13.1. URC of Connection Closed

When TCP socket service is closed by remote peer or network error, this report will be outputted. The <socket_state> of <connectID> will be "closing". Host must execute AT+QICLOSE=<connectID> to change the <socket_state> to "initial". If the <access_mode> of <connectID> is Buffer access mode, host can also execute AT+QIRD=<connectID>,<read_length> to read the buffer data.

URC of Connection Closed

+QIURC: "closed",<connectID>	<connectID> connection is closed
-------------------------------------------	-----------------------------------------------

Parameter

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

2.13.2. URC of Incoming Data

In buffer access mode or direct push mode, after receiving data, the module will report URC of host.

In buffer access mode, after receiving data, the module will report URC as +QIURC:"recv",<connectID> to notify the host. Then host can retrieve data by AT+QIRD. Notes: if the buffer is not empty, and the module receives data again, it will not report a new URC until all the received data has been retrieved by AT+QIRD from buffer. In direct push mode, the received data will be putputted to COM port directly.

URC of Incoming Data

+QIURC: "recv",<connectID>	The URC of incoming data in buffer access mode. The host can receive data by AT+QIRD
+QIURC: "recv",<connectID>,<currentrecvlength><CR><LF><data>	The UTC of incoming data in direct push mode when the <service_type> is "TCP", "UDP".

Parameter

<connectID>	Integer type, socket service index, range is 0-4
<Currentrecvlength>	Integer type, the length of actual received data
<data>	The received data

2.13.3. URC of Incoming Data Buff Full

In buffer access mode, if there is no resources can be allocated, then the module will report the URC as +QIURC:"recv",<connectID>,"buff full".

URC of Incoming Data Buff Full

+QIURC: "recv",<connectID>,"buff full"	Incoming data is full.
----------------------------------------	------------------------

Parameter

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

3 Description of Error Code

If TCPIP AT commands response ERROR, the details of error can be queried by AT+QIGETERROR.
Note: AT+QIGETERROR just returns error code of the last TCP/IP AT command.

Table 1: The Description of Error Code

<err>	Description of Error Code
0	operation successfully
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 parsed failed
566	socket connect failed

567	socket has been closed
568	operation busy
569	operation timeout
570	PDP context break 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 into Buffer Access Mode

```
AT+QIOPEN=1,0,"TCP","220.180.239.212",8062,1234,0 //Context 1 and <connectID> is 0.
OK

+QIOPEN: 0,0 //Connect successfully. It is suggested to wait 40 seconds for the URC
               response as "+QIOPEN:<connectID>,<err>". If the URC response has
               not been received in 40 seconds, host could use AT+QICLOSE to close
               the socket.

AT+QISTATE=1,0 //Query if connection state of <connectID> is 0.
+QISTATE: 0,1,"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 length is 10.
OK

SEND OK

AT+QISENDEX=0,5,3031323334 //Send hex string
OK

SEND OK

AT+QISEND=0,10,1234567890 //Send data, and the length is 10.
OK
AT+QISEND=0,10,1234567890 //The SEND OK of the previous command has not returned,
                           so when send new data will return ERROR.

ERROR

SEND OK
```

4.1.3. Receive Data from Remote Server in Buffer Access Mode

```
+QIURC: "recv",0           //The <connectID> 0 received data.
AT+QIRD=0,512              //Read data, the length is 512.
+QIRD: 10
1234567890

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

OK

+QIURC: "recv",0,"buff full" //The <connectID> 0 indicates that the buffer is full, the host need use the
                             AT+QIRD to read the buffer data.
```

4.1.4. Close Connection

```
AT+QICLOSE=0              //Close a connection whose <connectID> 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 into Direct Push Mode

```
AT+QIOPEN=1,0,"TCP","220.180.239.212",8062,0,1 //Context 1 and <connectID> is 0.
OK

+QIOPEN: 0,0           //Connect successfully. It is suggested to wait 40 seconds for the URC
                        //response as "+QIOPEN:<connecyID>,<err>". If the URC response has
                        //not been received in 40 seconds, host could use AT+QICLOSE to close
                        //the socket.

AT+QISTATE=1,0          //Query if connection state of <connectID> is 0.
+QISTATE: 0,1,"220.180.239.212",8062,0,2,1,1

OK
```

4.2.2. Send Data in Direct Push Mode

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

SEND OK

```
AT+QISENDEX=0,5,3031323334 //Send hex string
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
```

4.2.4. Close TCP Client

```
AT+QICLOSE=0 //Close a connection whose <connectID> is 0.
OK

CLOSE OK
```

4.3. PING

```
AT+QPING=1,"hf.quectel.com" //Ping hf.quectel.com in context 1.
OK

+QPING: 0,"220.180.239.212",32,192,255

+QPING: 0,"220.180.239.212",32,240,255

+QPING: 0,"220.180.239.212",32,241,255

+QPING: 0,"220.180.239.212",32,479,255

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

4.4. Synchronize Local Time

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

```
+QNTTP: 0,"18/04,11:08:20:35+8"
```

4.5. Example of Getting Last Error Code

```
AT+QIOPEN=1,"UDP","220.180.239.212",8063,0,1 //Start socket service, forget the <connectID>  
ERROR  
AT+QIGETERROR  
+QIGETERROR: 552,invalid parameters  
ERROR
```