

UC20 TCPIP

AT Commands Manual

UMTS/HSPA Module Series

Rev. UC20_TCPIP_AT_Commands_Manual_V1.1

Date: 2013-12-05



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

Quectel Wireless Solutions Co., Ltd.

Room 501, Building 13, No.99, Tianzhou Road, Shanghai, China, 200233

Tel: +86 21 5108 6236

Mail: info@quectel.com

Or our local office, for more information, please visit:

<http://www.quectel.com/support/salesupport.aspx>

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

<http://www.quectel.com/support/techsupport.aspx>

GENERAL NOTES

QUECTEL OFFERS THIS 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 ARE SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

THIS INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL CO., LTD. TRANSMITTABLE, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THIS CONTENTS 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. 2013. All rights reserved.

About the Document

History

Revision	Date	Author	Description
1.0	2013-05-20	Jonathan WEN	Initial
1.1	2013-12-05	Gibson PAN	<ul style="list-style-type: none">Added NTP functionAdded DNS functionAdded maximum response time to AT commands: AT+QIDEACT/AT+QIOPEN/AT+QICLOSE/AT+QPING/AT+QNTP/AT+QIDNSGIP

Contents

About the Document	2
Contents	3
1 Introduction	5
1.1. The Process of Using TCPIP AT Commands	5
1.2. Description of Data Access Mode	5
2 Description of AT Command	7
2.1. AT+QICSGP Configure Context	7
2.2. AT+QIACT Activate PDP Context	8
2.3. AT+QIDEACT Deactivate PDP Context	9
2.4. AT+QIOPEN Start Socket Service	9
2.5. AT+QICLOSE Close Socket Service	11
2.6. AT+QISTATE Query Socket Service Status	12
2.7. AT+QISEND Send Data	13
2.8. AT+QIRD Retrieve the Received TCP/IP Data	16
2.9. AT+QISWTMD Switch Data Access Mode	17
2.10. AT+QPING Ping a Remote Server	18
2.11. AT+QNTP Synchronize Local Time with NTP Server	20
2.12. AT+QIDNSCFG Configure Address of DNS Server	21
2.13. AT+QIDNSGIP Get IP Address by Domain Name	21
2.14. AT+QICFG Configure Optional Parameters	22
2.15. AT+QIGETERROR Query the Last Error Code	23
2.16. Description of URC	23
2.16.1. URC of Connection Closed	23
2.16.2. URC of Incoming Data	24
2.16.3. URC of Incoming Connection Full	24
2.16.4. URC of Incoming Connection	25
2.16.5. URC of PDP Deactivation	25
3 Description of Error Code	26
4 Example	28
4.1. Configure and Activate Context	28
4.1.1. Configure Context	28
4.1.2. Activate Context	28
4.1.3. Deactivate Context	28
4.2. TCP Client Works in Buffer Access Mode	28
4.2.1. Set up a TCP Client Connection and Enter into Buffer Access Mode	28
4.2.2. Send Data in Buffer Access Mode	29
4.2.3. Receive Data from Remote Server in Buffer Access Mode	29
4.2.4. Close Connection	30
4.3. TCP Client Works in Transparent Access Mode	30

4.3.1.	Set up a TCP Client Connection and Enter into Transparent Access Mode	30
4.3.2.	Send Data in Transparent Access Mode	30
4.3.3.	Receive Data from Remote Server in Transparent Access Mode	30
4.3.4.	Close TCP Client.....	30
4.4.	TCP Client Works in Direct Push Mode	31
4.4.1.	Set up TCP Client Connection and Enter into Direct Push Mode	31
4.4.2.	Send Data in Direct Push Mode.....	31
4.4.3.	Receive Data from Remote Server in Direct Push Mode	32
4.4.4.	Close TCP Client.....	32
4.5.	TCP Server Works in Buffer Access Mode	32
4.5.1.	Start a TCP Server	32
4.5.2.	Accept TCP Incoming Connection	32
4.5.3.	Receive Data from Incoming Connection	32
4.5.4.	Close TCP Server	33
4.6.	Example of UDP Service	33
4.6.1.	Start UDP Service	33
4.6.2.	Send UDP Data to Remote	34
4.6.3.	Receive Data from Remote.....	34
4.6.4.	Close UDP Service	34
4.7.	PING.....	34
4.8.	Synchronize Local time.....	35
4.9.	Example of Getting Last Error Code.....	35

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, TCP server, UDP service.

1.1. The Process of Using TCPIP AT Commands

Through TCPIP AT commands, host can configure PDP context, activate/deactivate PDP context, start/close socket service and send/receive data via socket service. The general process is shown as follows:

- Step 1:** Configure the <apn>, <username>, <password> and other parameters of a context by AT+QICSGP. If QoS settings need to be updated, configure them by the commands AT+CGQMIN, AT+CGEQMIN, AT+CGQREQ and AT+CGEQREQ.
- Step 2:** Activate PDP context by AT+QIACT, then the assigned IP address can be queried by AT+QIACT?.
- Step 3:** Start a socket service by AT+QIOPEN. The service type can be specified by the parameter <servicetype>. The URC "+QIOPEN" indicates whether the socket service is started successfully.
- Step 4:** Send and receive data via the socket. The module has three kinds of data access mode. The AT command flow may be different according to the data access mode. Please refer to Chapter 1.2 for details. For example, in buffer access mode, host can send data by AT+QISEND. After data has been received, the module will report a URC as +QIURC: "recv",<connectid> to notify the host. Then host can retrieve data by AT+QIRD.
- Step 5:** Close the socket service by AT+QICLOSE. Step 3 to Step 5 can be repeated.
- Step 6:** Deactivate PDP context by AT+QIDEACT.

1.2. Description of Data Access Mode

The TCPIP AT commands of this module include three kinds of data access modes:

1. Buffer access mode
2. Direct push access mode
3. Transparent access mode

When opening a socket service via AT+QIOPEN, specify the data access mode by the parameter <accessmode>. 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>. Then 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>.
3. In transparent access mode, the corresponding port (such as UART, USB modem port, etc.) enters into the exclusive mode (Note that the USB AT port does not support transparent access mode). The data received from COM port will be sent to Internet directly, and the received data from Internet will be outputted to COM port directly as well. "+++" is used to exit from transparent access mode. When "+++" returns OK, the module will be switched to buffer access mode. AT+QISWTMD can be used to return back to transparent access mode. Note: In transparent access mode, host cannot execute AT command. If the socket connection is closed because of network error or else, the module will report NO CARRIER and exit from the transparent access mode. In this case, AT+QICLOSE should be executed to close socket service.
4. To exit from the transparent access mode, use "+++" or DTR (AT&D1 should be set). To prevent the "+++" from being misinterpreted as data, it should comply to the following sequence:
 - 1) Do not input any character within T1 time (1 second) before inputting "+++".
 - 2) Input "+++" during 1s, and no other characters can be inputted during this time.
 - 3) Do not input any character within T1 time (1 second) after "+++" has been inputted.
 - 4) Exit from transparent access mode, return OK.
5. There are two methods to return back to the transparent access mode:
 - 1) By AT+QISWTMD. Specify the <access mode> as 2. If it enters into transparent access mode successfully, CONNECT will be returned.
 - 2) By ATO. ATO will change the access mode of connection that exits from transparent access mode lately. If it enters into transparent access mode successfully, CONNECT will be returned. If there is no connection enters into transparent access mode before, ATO will return NO CARRIER.

2 Description of AT Command

2.1. AT+QICSGP Configure Context

Configure the <apn>, <username>, <password> and other contexts by AT+QICSGP. The QoS of the context can be configured by AT+CGQMIN, AT+CGEQMIN, AT+CGQREQ and AT+CGEQREQ.

AT+QICSGP Configure Context	
Test Command AT+QICSGP=?	Response +QICSGP: (1-16),1-IPV4,APN,USER NAME, PASSWORD,(0-3) OK
Query the configuration of context AT+QICSGP=<contextid>	Response +QICSGP: <contexttype>,<apn>,<username>,<password>,<authentication> OK
Write Command, configure the context AT+QICSGP=<contextid>,<contexttype>,[<apn>,<username>,<password>],<authentication>]]	Response OK ERROR

Parameter

<contextid>	Integer type, context ID, range is 1-16
<contexttype>	Integer type, protocol type 1 IPV4
<apn>	String type, access point name
<username>	String type, user name
<password>	String type, password
<authentication>	Integer type, the authentication methods 0 NONE 1 PAP 2 CHAP 3 PAH_OR_CHAP

Example

```

AT+QICSGP=1 //Query the configuration of context 1.
+QICSGP: 1,"", "", "", "", 0

OK
AT+QICSGP=1,1,"UNINET","", "", 1 //Configure context 1, APN is "UNINET" for China Unicom.

OK

```

2.2. AT+QIACT Activate PDP Context

Before activating context by AT+QIACT, host should configure the context by AT+QICSGP. After activation, the IP address can be queried by AT+QIACT?. The range of <contextid> is 1-16, but the maximum number of context which can be activated at the same time is 3. Depending on the Network, it may take at most 150 seconds to return OK or ERROR after executing AT+QIACT. Before the response is returned, other AT commands cannot be executed.

AT+QIACT Activate PDP Context	
Test Command AT+QIACT=?	Response +QIACT: (1-16) OK
Read command AT+QIACT?	Response Return the list of the current activated context and its IP address: +QIACT: 1,<contextstate>,<contexttype>[,<ipaddress>] [..... +QIACT: 16,<contextstate>,<contexttype>[,<ipaddress>]] OK
Write Command AT+QIACT=<contextid>	Response Activate the context of specified <contextid>: OK ERROR
Maximum Response Time	150 seconds, determined by network.

Parameter

<contextid>	Integer type, context ID, range is 1-16
<contextstate>	Integer type, context state
	0 Deactivated
	1 Activated
<contexttype>	Integer type, protocol type
	1 IPV4
<ipaddress>	The local IP address after context is activated

2.3. AT+QIDEACT Deactivate PDP Context

AT+QIDEACT will deactivate the specific context <contextid> and close all TCPIP connections set up in this context. Depending on the Network, it may take at most 40 seconds to return OK or ERROR after executing AT+QIDEACT. Before the response is returned, other AT commands cannot be executed.

AT+QIDEACT Deactivate PDP Context	
Test Command AT+QIDEACT=?	Response OK
Write Command AT+QIDEACT=<contextid>	Response OK ERROR
Maximum Response Time	40 seconds, determined by network.

Parameter

<contextid>	Integer type, context ID, range is 1-16
-------------	-----------------------------------------

2.4. AT+QIOPEN Start Socket Service

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

1. If <servicetype> is "TCP LISTENER", module works as TCP SERVER. After accepting a new TCP connection, module will auto specify a <connectid> and report a URC as "+QIURC:

"incoming",<connectid>,<serverid>,<remoteip>,<remoteport>". The range of <connectid> is 0-11. The type of this new incoming connection is "TCP INCOMING" and the <accessmode> of "TCP INCOMING" is the same with the <accessmode> of "TCP LISTENER".

2. If <servicetype> is "UDP SERVICE", UDP data can be sent to or received from the remote IP via <localport>.
 - Send data: AT+QISEND=<connectid>,<sendlength>,<remoteip>,<remoteport>.
 - Receive data in direct push access mode: report a URC as "+QIURC: "recv",<connectid>,<currentrecvlength>,<remoteip>,<remoteport><CR><LF><data>"
 - Receive data in buffer access mode: report a URC as "+QIURC: "recv",<connectid>, then retrieve data by AT+QIRD=<connectid>.
3. It is suggested to wait 150 seconds for the URC response as "+QIOPEN: <connectid>,<errorcode>". If the URC response has not been received in 150 seconds, host could use AT+QICLOSE to close the socket.

AT+QIOPEN Start Socket Service

Test Command AT+QIOPEN=?	Response +QIOPEN: (1-16),(0-11),"TCP/UDP/TCP LISTENER/UDP SERVICE","<ipaddress>/<domainname>",<remoteport>,<localport>,(0-2) OK
Read Command AT+QIOPEN?	Response OK
Write Command AT+QIOPEN=<contextid>,<connectid>,<servicetype>,<ipaddress>/<domainname>,<remoteport>[,<localport>[,<accessmode>]]	Response If the <accessmode> is transparent access mode and it is successful to start the service, response: CONNECT Else, response: ERROR Error description can be got via AT+QIGETERROR. If the <accessmode> is buffer access mode or direct push mode, response: OK +QIOPEN: <connectid>,<errorcode> <errorcode> is 0 when service is started successfully, else <errorcode> is not 0.
Maximum Response Time	150 seconds, determined by network.

Parameter

<contextid>	Integer type, context ID, range is 1-16
<connectid>	Integer type, socket service index, range is 0-11
<servicetype>	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
<ipaddress>	String type If <servicetype> is TCP or UDP, it indicates the IP address of remote server, such as "61.190.19.174" If <servicetype> is TCP LISTENER or UDP SERVICE, please enter "127.0.0.1"
<domainname>	String type, the domain name address of the remote server
<remoteport>	The port of the remote server, only valid when <servicetype> is "TCP" or "UDP"
<localport>	The local port If <servicetype> is "TCP LISTENER" or "UDP SERVICE", this parameter must be specified If <servicetype> is "TCP" or "UDP", if <localport> is 0, then the local port will be assigned automatically, else the local port is assigned as specified
<accessmode>	Integer type, the data access mode of the socket services 0 Buffer access mode 1 Direct push mode 2 Transparent access mode
<errorcode>	Integer type, error code. Please refer to Chapter 3 for details

2.5. AT+QICLOSE Close Socket Service

Close the specified socket service by AT+QICLOSE. Depending on the Network, It will take at most 10 seconds (default value, host could modify the time with <timeout>) to return OK or ERROR after executing AT+QICLOSE. Before the response is returned, other AT commands cannot be executed.

AT+QICLOSE Close Socket Service

Test Command AT+QICLOSE=?	Response OK
Write Command AT+QICLOSE=<connectid>[,<timeout>]	Response If closes successfully, response: OK If failed to close, response: ERROR

Parameter

<connectid>	Integer type, socket service index, range is 0-11
<timeout>	Integer type. If the FIN ACK of the other peers is still not received until <timeout> expires, the module will force to close the socket. Range is 0-65535. Default: 10. Unit: second

2.6. AT+QISTATE Query Socket Service Status

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

AT+QISTATE Query Socket Service Status	
Test Command AT+QISTATE=?	Response OK
If <querytype> is 0, query by specified <contextid> AT+QISTATE=<querytype>,<contextid>	Response Return the status of all existing connections in specified <contextid> List of (+QISTATE: <connectid>,<servicetype>,<ipaddress>,<remoteport>,<localport>,<socketstate>,<contextid>,<serverid>,<accessmode>,<atport>) OK
If <querytype> is 1, query by specified <connectid> AT+QISTATE=<querytype>,<connectid>	Response +QISTATE: <connectid>,<servicetype>,<ipaddress>,<remoteport>,<localport>,<socketstate>,<contextid>,<serverid>,<accessmode>,<atport> OK

Parameter

<querytype>	Integer type, the query type 0 Query connection status by <contextid> 1 Query connection status by <connectid>
<contextid>	Integer type, context ID, range is 1-16
<connectid>	Integer type, socket service index, range is 0-11

<servicetype>	String type, the service type
“TCP”	TCP connection as a client
“UDP”	UDP connection as a client
“TCP LISTENER”	TCP listener socket as a TCP server
“TCP INCOMING”	TCP connection accepted by a TCP server
“UDP SERVICE”	UDP service
<ipaddress>	IP address
If <servicetype>=“TCP” or “UDP”, <ipaddress> is the IP address of remote server	
If <servicetype>=“TCP LISTENER” or “UDP SERVICE”, <ipaddress> is local IP address	
If <servicetype>=“TCP INCOMING”, <ipaddress> is the IP address of remote client	
<remoteport>	Remote port number
If <servicetype>=“TCP” or “UDP”, <remoteport> is the port of remote server	
If <servicetype>=“TCP LISTENER” or “UDP SERVICE”, <remoteport> is invalid	
If <servicetype>=“TCP INCOMING”, <remoteport> is the port of remote client	
<localport>	Local port number
If <localport> is 0, local port is assigned automatically	
<socketstate>	Integer type, socket service state
0	“Initial” connection has not been established
1	“Opening” client is connecting or server is trying to listen
2	“Connected” client/incoming connection has been established
3	“Listening” server is listening
4	“Closing” connection is closing
<serverid>	Only valid when <servicetype> is “TCP INCOMING”. <servicetype> represents which server accepts this TCP incoming connection, the value is the same as <connectid> of this server’s “TCP LISTENER”
<accessmode>	Data access mode
0	Buffer access mode
1	Direct push mode
2	Transparent access mode
<atport>	String type, COM port of socket service
“usbmodem”	USB MODEM PORT
“usbat”	USB AT PORT
“uart”	UART PORT
“cmux1”	MUX PORT 1
“cmux2”	MUX PORT 2
“cmux3”	MUX PORT 3
“cmux4”	MUX PORT 4

2.7. AT+QISEND Send Data

If <accessmode> of the specified socket service is buffer access mode or direct push mode, send data by AT+QISEND. If data is sent to the module successfully, return “SEND OK”, else return “SEND FAIL” or

“ERROR”. “SEND FAIL” indicates the send buffer is full and host can try to resend the data. “ERROR” indicates it encounters error in the process of sending data. The host should delay some time for sending data. The maximum length of sending data is 1460. “SEND OK” does not mean the data has been sent to the server successfully. Through the “AT+QISEND=<connectid>, 0” command, host can query whether the data has reached the server.

AT+QISEND Send Data	
<p>Test Command AT+QISEND=?</p>	<p>Response +QISEND: (0-11),(0-1460)</p> <p>OK</p>
<p>If<servicetype> is “TCP”, “UDP” or “TCP INCOMING”, send data with changeable length AT+QISEND=<connectid> Response “>”, then type data to send, tap CTRL+Z to send, tap ESC to cancel the operation</p>	<p>Response > <input data> CTRL+Z</p> <p>If connection has been established and sending is successful, response: SEND OK</p> <p>If connection has been established but sending buffer is full, response: SEND FAIL</p> <p>If connection has not been established, abnormally closed, or parameter is incorrect, response: ERROR</p>
<p>If<servicetype> is “TCP”, “UDP” or “TCP INCOMING”, send data with fixed length AT+QISEND=<connectid>,<sendlength> h> Response “>”, type data until the data length is equal to <sendlength></p>	<p>Response > <input data with specified length></p> <p>If connection has been established and sending is successful, response: SEND OK</p> <p>If connection has been established but sending buffer is full, response: SEND FAIL</p> <p>If connection has not been established, abnormally closed, or parameter is incorrect, response: ERROR</p>

<p>If <servicetype> is "UDP SERVICE"</p> <p>AT+QISEND=<connectid>,<sendlength>,<remoteip>,<remoteport></p> <p>Response ">", type data until the data length is equal to <sendlength></p>	<p>Response</p> <p>This command is used to send fixed length data to specified remote IP address and remote port. The <servicetype> must be "UDP SERVICE".</p> <p>></p> <p><input data with specified length></p> <p>If connection has been established and sending is successful, response:</p> <p>SEND OK</p> <p>If connection has been established but sending buffer is full, response:</p> <p>SEND FAIL</p> <p>If connection has not been established, abnormally closed, or parameter is incorrect, response:</p> <p>ERROR</p>
<p>When <sendlength> is 0, query the sent data</p> <p>AT+QISEND=<connectid>,0</p>	<p>Response</p> <p>If the specified <connectid> connection has existed, return:</p> <p>+QISEND:</p> <p><totalsendlength>,<ackedbytes>,<unackedbytes></p> <p>OK</p> <p>Else, return</p> <p>ERROR</p>

Parameter

<connectid>	Integer type, socket service index, range is 0-11
<sendlength>	Integer type. The length of data to be sent, which cannot exceed 1460
<remoteip>	String type, the remote IP address, only valid when <servicetype> is "UDP SERVICE"
<remoteport>	Integer type, the remote port, only valid when <servicetype> is "UDP SERVICE"
<totalsendlength>	Integer type, the total length of sent data
<ackedbytes>	Integer type, the total length of acknowledged data
<unackedbytes>	Integer type, the total length of unacknowledged data
<errorcode>	Integer type, error code. Please refer to Chapter 3 for details

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

In buffer access mode, after receiving data, the module will buffer it and report a 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, then it will not report a new URC until all the received data has been retrieved by AT+QIRD from buffer.

For UDP socket, the <readlength> should be ignored; and for TCP socket, the <readlength> must be specified.

AT+QIRD Retrieve the Received TCP/IP Data

Test Command

AT+QIRD=?

Response

+QIRD: (0-11),(0-1500)

OK

When <servicetype> is "TCP" or "TCP INCOMING"

AT+QIRD=<connectid>,<readlength>

Response

If the specified connection has received data, response:

+QIRD: <readactuallength><CR><LF><data>

OK

If no data, return:

+QIRD: 0

OK

If connection does not exist, return:

ERROR

When <servicetype> is "UDP"

AT+QIRD=<connectid>

Response

If the specified connection has received data, response:

+QIRD: <readactuallength><CR><LF><data>

OK

If no data, return:

+QIRD: 0

OK

If connection does not exists, return:

ERROR

When <servicetype> is "UDP SERVICE"

AT+QIRD=<connectid>

Response

If data exists, return:

+QIRD:

	<p><readactualelength>,<remoteip>,<remoteport><CR><LF><data></p> <p>OK</p> <p>If there is no data, return:</p> <p>+QIRD: 0</p> <p>OK</p> <p>If connection does not exist, return:</p> <p>ERROR</p>
<p>When <readlength> is 0, query retrieved data</p> <p>AT+QIRD=<connectid>,0</p>	<p>Response</p> <p>If the specified connection has existed, return:</p> <p>+QIRD:</p> <p><totalreceivelength>,<havereadlength>,<unreadlength></p> <p>OK</p> <p>Else response:</p> <p>ERROR</p>

Parameter

<connectid>	Integer type, the socket service index, the range is 0-11
<readlength>	The maximum length of data to be retrieved, the range is 0-1500
<readactualelength>	The real length of received data
<remoteip>	String type, the remote IP address, only valid when <servicetype> is "UDP SERVICE"
<remoteport>	Integer type, the remote port, only valid when <servicetype> is "UDP SERVICE"
<data>	The retrieved data
<totalreceivelength>	The total length of received data
<havereadlength>	The read length of received data
<unreadlength>	The unread length of received data

2.9. AT+QISWTMD Switch Data Access Mode

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

AT+QISWTMD Switch Data Access Mode

Test Command AT+QISWTMD=?	Response +QISWTMD: (0-11),(0-2) OK
Read Command AT+QISWTMD?	Response OK
Write Command AT+QISWTMD=<connectid>,<access mode>	Response If switch successfully and <accessmode> is 0 or 1, response: OK If switch successfully and <accessmode> is 2, module will enter into data mode, response: CONNECT Else response: ERROR

Parameter

<connectid>	Integer type, the socket service index, the range is 0-11
<accessmode>	Integer type, the data access mode of the connection
0	Buffer access mode
1	Direct push mode
2	Transparent access mode

2.10. AT+QPING Ping a Remote Server

AT+QPING is used to test the Internet Protocol reachability of a host. Before using ping tools, host should activate the context corresponding to <contextid> via AT+QIACT first. It will return the result during <timeout> and the default value of <timeout> is 1 second.

AT+QPING Ping a Remote Server

Test Command AT+QPING=?	Response +QPING: (1-16),"HOST",(1-255),(1-10) OK
Write Command AT+QPING=<contextid>,<host>[,<timeout>[,<pingnum>]]	Response If ping successfully, return: OK

```
<CR><LF>[+QPING:
<result>[,<ipaddress>,<bytes>,<time>,<ttd>]<CR><LF>...]
<CR><LF>+QPING:
<finresult>[,<sent>,<rcvd>,<lost>,<min>,<max>,<avg>]
Else response:
ERROR
```

Parameter

<contextid>	Integer type, the context ID, the range is 1-16
<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: 1
<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 “,<ipaddress>,<bytes>,<time>,<ttd>”
	1 Timeout for the ping request. In the case, no other information followed
<ipaddress>	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 the response of the ping request. Unit: ms
<ttd>	Time To Live value of the response packet for the ping request
<finresult>	The final result of the command
	2 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>,<max>,<avg>”
	3 The TCP/IP stack is busy now. In the case, no other information followed
	4 Not find the host. In the case, no other information followed
	5 Failed to activate PDP context. In this case, no other information followed
<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

2.11. AT+QNTF Synchronize Local Time with NTP Server

NTP is intended to synchronize the Coordinated Universal Time (UTC) with the time server. Before using NTP, host should activate the context corresponding to <contextid> via AT+QIACT first. Depending on the Network, it will take at most 125 seconds to return the result.

AT+QNTF Synchronize Local Time with NTP Server

Test command AT+QNTF=?	Response +QNTF: (1-16),"SERVER",(list of supported<port>s),(0,1) OK
Read command AT+QNTF?	Response If in the process of synchronizing local time: +QNTF: <server>,<port> OK Else response: ERROR
Write command AT+QNTF=<contextid>,<server>[<port>][,<autosettime>]	Response If synchronize local time with NTP server successfully OK +QNTF: <errorcode>,<time> Else response: ERROR
Maximum Response Time	125s, determined by network.

Parameter

<contextid>	Integer type, indicates context ID, range is 1-16
<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
<errorcode>	Integer type, indicates synchronized result, please refer to Chapter 3
<time>	String type, indicates time which is synchronized from NTP server The format is "YYYY/MM/DD,hh:mm:ss±zz". The range of zz is -48~56

2.12. AT+QIDNSCFG Configure Address of DNS Server

Before setting the DNS address, host must activate the context corresponding to <contextid> via AT+QIACT first.

AT+QIDNSCFG Configure Address of DNS Server	
Test command AT+QIDNSCFG=?	Response +QIDNSCFG: (1-16),"pridnsaddr","secdnsaddr" OK
Configure DNS server address for specified PDP context AT+QIDNSCFG=<contextid>,<pridnsaddr>[,<secdnsaddr>]	Response OK ERROR
Query DNS server address of specified PDP context AT+QIDNSCFG=<contextid>	Response +QIDNSCFG: <contextid>,<pridnsaddr>,<secdnsaddr> OK

Parameter

<contextid>	Integer type, indicates PDP context ID, range is 1-16
<pridnsaddr>	String type, indicates primary DNS server address
<secdnsaddr>	String type, indicates secondary DNS server address

2.13. AT+QIDNSGIP Get IP Address by Domain Name

Before querying the DNS, host should activate the context corresponding to <contextid> via AT+QIACT first. Depending on the Network, it will take at most 60 seconds to return the result.

AT+QIDNSGIP Get IP Address by Domain Name	
Test Command AT+QIDNSGIP=?	Response +QIDNSGIP: (1-16),"hostname" OK
Write Command AT+QIDNSGIP=<contextid>,<hostname>	Response OK ERROR

	<p>The result will be returned as URC.</p> <p>+QIURC: "dnsgip",<errorcode>,<ipcount>, <dnsttl> [..... +QIURC: "dnsgip",<hostipaddr>]</p>
Maximum Response Time	60s, determined by network.

Parameter

<contextid>	Integer type, indicates PDP context ID, range is 1-16
<hostname>	String type, indicates domain name
<ipaddrn>	String type, indicates dot format IP address
<errorcode>	Integer type, please refer to Chapter 3
<ipcount>	Integer type, the number of the IP address corresponding to the <hostname>
<dnsttl>	Integer type, the time of the DNS to live
<hostipaddr>	String type, the IP address of <hostname>

2.14. AT+QICFG Configure Optional Parameters

AT+QICFG Configure Optional Parameters	
<p>Test Command</p> <p>AT+QICFG=?</p>	<p>Response</p> <p>+QICFG: "transpktsize",(1-1460) +QICFG: "transwaittm",(1-10)</p> <p>OK</p>
<p>Set packet size for transparent mode</p> <p>AT+QICFG="transpktsize"[,<transpktsize>]</p>	<p>Response</p> <p>+QICFG: "transpktsize",<transpktsize></p> <p>OK or OK ERROR</p>
<p>Set waiting time for transparent mode</p> <p>AT+QICFG="transwaittm"[,<transwaittm >]</p>	<p>Response</p> <p>+QICFG: "transwaittm",<transwaittm></p> <p>OK or OK ERROR</p>

Parameter

<transpktsize>	Integer type, indicates the max length of sending data packet, range is 1-1460, default: 1460
<transwaittm>	Integer type, in transparent access mode, if the length of data received from the port is less than the specified value of <transpktsize> , after exceeding the time of <transwaittm> , data will be sent directly. Unit: ms, range: 1-10, default: 10

2.15. AT+QIGETERROR Query the Last Error Code

If TCPIP AT commands respond 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: <errorcode>,<descriptionoferrcode> OK

Parameter

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

2.16. Description of URC

The URC of TCPIP AT commands will be reported to the host by the type of "+QIURC:". It contains the reports about incoming data, connection closed and incoming connection and so on.

2.16.1. URC of Connection Closed

When TCP socket service is closed by remote peer or network error, this report will be outputted. The **<socketstate>** of **<connectid>** will be "closing". Host must execute AT+QICLOSE=**<connectid>** to change the **<socketstate>** to "initial".

URC of Connection Closed

+QIURC: "closed",<connectid> <connectid> connection is closed

Parameter

<connectid> Integer type, socket service index, range is 0-11

2.16.2. URC of Incoming Data

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

In buffer access mode, after receiving data, the module will report a 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 outputted 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 URC of incoming data in direct push mode when the <servicetype> is "TCP", "UDP" or "TCP INCOMING".
+QIURC: "recv",<connectid>,<currentrecvlength>,<remoteip>,<remoteport><CR><LF><data>	The URC of data incoming in direct push mode when <servicetype> is "UDP SERVICE".

Parameter

<connectid>	Integer type, socket service index, range is 0-11
<currentrecvlength>	Integer type, the length of actual received data
<remoteip>	Remote IP address
<remoteport>	Remote port
<data>	The received data

2.16.3. URC of Incoming Connection Full

If the incoming connection reaches the limit, or no socket system resources can be allocated, then the module will report the URC as +QIURC: "incoming full" for the new incoming connection request.

URC of Incoming Connection Full

+QIURC: "incoming full" Incoming connection is full.

2.16.4. URC of Incoming Connection

If the <servicetype> is "TCP LISTENER", when remote client connects to this server, host will auto assign a free <connectid> for the new connection. The range of <connectid> is 0-11. The module will report a URC as below. The <servicetype> of new connection is "TCP INCOMING", and the <accessmode> is buffer access mode.

URC of Incoming Connection

+QIURC:
"incoming",<connectid>,<serverid>,<remoteip>,<remoteport>
When new incoming connection is accepted by <serverid>, the allocated <connectid>, <remoteip> and <remoteport> will be informed from this URC.

Parameter

<connectid>	Integer type, assign this <connectid> for the incoming connection, which is auto specified by the module. The range is 0-11
<serverid>	The incoming <connectid> accepted by the server which <servicetype> is "TCP LISTENER" and listening socket ID is <serverid>
<remoteip>	Remote IP address of the incoming <connectid>
<remoteport>	Remote port of the incoming <connectid>
<remoteport>	Remote port

2.16.5. URC of PDP Deactivation

PDP context may be deactivated by the network. The module will report a URC to the host. Host must execute AT+QIDEACT to deactivate the context and reset all connections.

URC of PDP Deactivation

+QIURC: "pdpdeact",<contextid> <contextid> context is deactivated.

Parameter

<contextid>	Integer type, context ID, range is 1-16
-------------	-----------------------------------------

3 Description of Error Code

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

Table 1: The Description of Error Code

<errorcode>	Description of Error Code
0	operate successfully
552	unknown error
553	operation blocked
554	open connection failed
555	Connection has been closed
556	active PDP failed
557	invalid IP address
558	invalid parameters
559	operation not supported
560	create socket failed
561	invalid socket ID
562	specified socket index has been used
563	PDP has already been activated
564	invalid APP type
565	DNS parse failed
566	invalid operation
567	socket listen failed

568	bind socket failed
569	memory not enough
570	socket index unused
571	operation timeout
572	no NTP response returns
573	PDP activating
574	DNS parse busy

Quectel
Confidential

4 Example

4.1. Configure and Activate Context

4.1.1. Configure Context

```
AT+QICSGP=1,1,"UNINET","", "",1 //Configure context 1, APN is "UNINET" for China Unicom.  
OK
```

4.1.2. Activate Context

```
AT+QIACT=1 //Activate context 1. Depending on the Network, the maximum  
OK //Activate successfully.  
AT+QIACT? //Query the state of context.  
+QIACT: 1,1,1,"10.7.157.1"  
OK
```

4.1.3. Deactivate Context

```
AT+QIDEACT=1 //Deactivate context 1.  
OK //Deactivate successfully. Depending on the Network, the  
maximum response time is 40s.
```

4.2. TCP Client Works in Buffer Access Mode

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

```
AT+QIOPEN=1,0,"TCP","61.190.19.174",8705,0,0 //Context is 1 and <connectid> is 0. Before using  
OK AT+QIOPEN, host should activate the context  
with AT+QIACT first.
```

```
+QIOPEN: 0,0                                     //Connect successfully. It is suggested to wait 150 seconds for the
                                                    URC response as "+QIOPEN: <connectid>,<errorcode>". If the
                                                    URC response has not been received in 150 seconds, host
                                                    could use AT+QICLOSE to close the socket.

AT+QISTATE=1,0                                    //Query if connection state of <connectid> is 0.
+QISTATE: 0,"TCP","61.190.19.174",8705,0,2,1,0,0,"usbmodem"

OK
```

4.2.2. Send Data in Buffer Access Mode

```
AT+QISEND=0                                       //Send changeable length data. "SEND OK" does not mean the
                                                    data has been sent to the server successfully. Host can query
                                                    whether the data has reached the server via AT+QISEND=0,0.

>test1<ctrl+Z>
SEND OK

AT+QISEND=0,4                                    //Send fixed length data and the data length is 4.
>test
SEND OK

AT+QISEND=0,0                                    //Query the length of sent data.
+QISEND: 9,9,0

OK
```

4.2.3. Receive Data from Remote Server in Buffer Access Mode

```
+QIURC: "recv",0                                //The <connectid> 0 received data.

AT+QIRD=0,1500                                   //Read data, the length is 1500.
+QIRD: 5                                          //The actual received data length is 5.
test1

OK

AT+QIRD=0,1500
+QIRD: 0                                          //No data in buffer.

OK

AT+QIRD=0,0                                       //Query length of total received data, including read and unread.
+QIRD: 5,5,0
```

OK

4.2.4. Close Connection

AT+QICLOSE=0

//Close a connection whose <connectid> is 0.
Depending on the Network, the maximum
response time is 10s.

OK

4.3. TCP Client Works in Transparent Access Mode

4.3.1. Set up a TCP Client Connection and Enter into Transparent Access Mode

AT+QIOPEN=1,0,"TCP","61.190.19.174",8705,0,2

//Context is 1 and <connectid> is 0. Before using
AT+QIOPEN, host should activate the context
with AT+QIACT first.

CONNECT

//Connect successfully. It is suggested to wait 150
seconds for the URC response as "+QIOPEN:
<connectid>,<errorcode>". If the URC response
has not been received in 150 seconds, host could
use AT+QICLOSE to close the socket.

4.3.2. Send Data in Transparent Access Mode

<All data got from COM port will be sent to internet directly>

4.3.3. Receive Data from Remote Server in Transparent Access Mode

Test 1

//All data received from internet will be outputted to
COM port directly.

4.3.4. Close TCP Client

AT+QICLOSE=0

//After using "+++" to exit from the transparent access
mode, host could use AT+QICLOSE to close the
TCP link. Depending on the Network, the maximum

response time is 10 seconds.

OK

4.4. TCP Client Works in Direct Push Mode

4.4.1. Set up TCP Client Connection and Enter into Direct Push Mode

AT+QIOPEN=1,0,"TCP","61.190.19.174",8705,0,1 //Context is 1 and <connectid> is 0. Before using AT+QIOPEN, host should activate the context with AT+QIACT first.

OK

+QIOPEN: 0,0

//Connect successfully. It is suggested to wait 150 seconds for the URC response as "+QIOPEN: <connectid>,<errorcode>". If the URC response has not been received in 150 seconds, host could use AT+QICLOSE to close the socket.

AT+QISTATE=1,0

//Query if connection state of <connectid> is 0.

+QISTATE: 0,"TCP","61.190.19.174",8705,0,2,1,0,1,"usbmodem"

OK

4.4.2. Send Data in Direct Push Mode

AT+QISEND=0

//Send changeable length data. "SEND OK" does not mean the data has been sent to the server successfully. Host can query whether the data has reached the server via AT+QISEND=0,0.

>test1<ctrl+Z>

SEND OK

AT+QISEND=0,5

//Send fixed length data and the data length is 5.

>test2

SEND OK

AT+QISEND=0,0

//Query the length of sent data. .

+QISEND: 10,10,0

OK

4.4.3. Receive Data from Remote Server in Direct Push Mode

```
+QIURC: "recv",0,4 //Receive data from remote server.  
test
```

4.4.4. Close TCP Client

```
AT+QICLOSE=0 //Close the connection of which <connectid>  
is 0. Depending on the Network, the  
maximum response time is 10s.  
OK
```

4.5. TCP Server Works in Buffer Access Mode

4.5.1. Start a TCP Server

```
AT+QIOPEN=1,1,"TCP LISTENER","127.0.0.1",0,2020,0 //Context is 1 and <connectid> is 1. Before  
using AT+QIOPEN, host should activate  
the context with AT+QIACT first.  
OK  
+QIOPEN: 1,0 //Start TCP server successfully.  
AT+QISTATE=0,1 //Query if connection state of <contextid> is 1.  
+QISTATE: 1,"TCP LISTENER","10.7.157.1",0,2020,3,1,0,0,"usbmodem"  
OK
```

4.5.2. Accept TCP Incoming Connection

```
+QIURC: "incoming",11,1,"172.31.242.222",54091 //Accept a TCP connection, <servicetype> is TCP  
incoming, <connectid> is 11.
```

4.5.3. Receive Data from Incoming Connection

```
+QIURC: "recv",11 //Receive data from remote incoming connection.  
AT+QIRD=11,1500 //Read data received from incoming connection
```

```
+QIRD: 4 //Actual data length is 4.
test

OK

AT+QIRD=11,1500
+QIRD: 0 //No data in buffer.

OK

AT+QIRD=11,0 //Query the data length of total received, including
read and unread.
+QIRD: 4,4,0

OK
```

4.5.4. Close TCP Server

```
AT+QICLOSE=11 //Close incoming connection. Depending on the
Network, the maximum response time is 10s.

OK

AT+QICLOSE=1 //Close TCP server listening.

OK
```

4.6. Example of UDP Service

4.6.1. Start UDP Service

```
AT+QIOPEN=1,2,"UDP SERVICE","127.0.0.1",0,3030,0 //Start a UDP Service, <connectid> is 2 and
<context> is 1. Before using AT+QIOPEN,
host should activate the context with
AT+QIACT first.

OK

+QIOPEN: 2,0 //Start UDP service successfully.

AT+QISTATE=0,1 //Query if connection status of <contextid> is 1
+QISTATE: 2,"UDP SERVICE","10.7.157.1",0,3030,2,1,0,0,"usbmodem"
```

OK

4.6.2. Send UDP Data to Remote

```
AT+QISEND=2,10,"10.7.89.10",6969 //Send 10 bytes data to remote which IP is
                                     10.7.89.10 and remote port is 6969.
>1234567890
SEND OK
```

4.6.3. Receive Data from Remote

```
+QIURC: "recv",2 //Receive data from remote.

AT+QIRD=2 //Read UDP data. One whole UDP packet will be
           outputted. There is no need to specify the
           read length.

+QIRD: 4,"10.7.76.34",7687 //Data length is 4, remote IP address is 10.7.76.34
                           and remote port is 7687 test.

AAAA

OK

AT+QIRD=2 //Read data.
+QIRD: 0 //No data in buffer.

OK

AT+QISEND=2,10,"10.7.76.34",7687 //Send data to remote (10.7.76.34:7687).
>1234567890
SEND OK
```

4.6.4. Close UDP Service

```
AT+QICLOSE=2 //Close the service.
OK
```

4.7. PING

```
AT+QPING=1,"www.baidu.com" //Ping www.baidu.com in context 1. Before pinging
```

the destination IP address, host should activate the context with AT+QIACT first.

OK

+QPING: 0,61.135.169.125,0,0,147

+QPING: 0,61.135.169.125,0,0,108

+QPING: 0,61.135.169.125,0,0,96

+QPING: 0,61.135.169.125,0,0,96

+QPING: 2,4,4,0,96,147,111

4.8. Synchronize Local time

AT+QNTP=1,"202.120.2.101",123

//Synchronize local time with NTP server "202.120.2.101:123". Before synchronizing the time, host should activate the context with AT+QIACT first.

OK

+QNTP: 0,"2013/08/19,11:10:10+32"

AT+CCLK?

+CCLK: "13/08/19,11:10:57+32"

OK

4.9. Example of Getting Last Error Code

AT+QIOPEN=1,"TCP","61.190.19.174",8033,0,1

//Start socket service, forget the <connectid>.

ERROR

AT+QIGETERROR

+QIGETERROR: 558, invalid parameters

OK