EC20 TCPIP AT Commands Manual

LTE Module Series

Rev. EC20_TCPIP_AT_Commands_Manual_V1.0

Date: 2015-05-08

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

Quectel Wireless Solutions Co., Ltd.
Office 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
Or Email: Support@quectel.com

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 IS 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. 2015. All rights reserved.

About the Document

History

Revision	Date	Author	Description
1.0	2015-05-08	Chris PENG	Initial

Quectel Confidential

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 7

2 Description of AT Command **8**
2.1. AT+QICSGP Configure Context 8
2.2. AT+QIACT Activate PDP Context 9
2.3. AT+QIDEACT Deactivate PDP Context 10
2.4. AT+QIOPEN Start Socket Service 10
2.5. AT+QICLOSE Close Socket Service 12
2.6. AT+QISTATE Query Socket Service Status 13
2.7. AT+QISEND Send Data 15
2.8. AT+QIRD Retrieve the Received TCP/IP Data 17
2.9. AT+QISENDEX Send HEX String 18
2.10. AT+QISWTMD Switch Data Access Mode 19
2.11. AT+QPING Ping a Remote Server 20
2.12. AT+QNTP Synchronize Local Time with NTP Server 21
2.13. AT+QIDNSCFG Configure Address of DNS Server 22
2.14. AT+QIDNSGIP Get IP Address by Domain Name 23
2.15. AT+QICFG Configure Optional Parameters 23
2.16. AT+QISDE Control Whether or Not to Echo the Data for QISEND 25
2.17. AT+QIGETERROR Query the Last Error Code 26
2.18. Description of URC 26
2.18.1. URC of Connection Closed 26
2.18.2. URC of Incoming Data 27
2.18.3. URC of Incoming Connection Full 27
2.18.4. URC of Incoming Connection 27
2.18.5. URC of PDP Deactivation 28

3 Description of Error Code **29**

4 Example **31**
4.1. Configure and Activate Context 31
4.1.1. Configure Context 31
4.1.2. Activate Context 31
4.1.3. Deactivate Context 31
4.2. TCP Client Works in Buffer Access Mode 31
4.2.1. Set up a TCP Client Connection and Enter into Buffer Access Mode 31
4.2.2. Send Data in Buffer Access Mode 32
4.2.3. Receive Data from Remote Server in Buffer Access Mode 32
4.2.4. Close Connection 33

Quectel Confidential

4.3. TCP Client Works in Transparent Access Mode 33
4.3.1. Set up a TCP Client Connection and Enter into Transparent Access Mode 33
4.3.2. Send Data in Transparent Access Mode 33
4.3.3. Receive Data from Remote Server in Transparent Access Mode 33
4.3.4. Close TCP Client 34
4.4. TCP Client Works in Direct Push Mode 34
4.4.1. Set up TCP Client Connection and Enter into Direct Push Mode 34
4.4.2. Send Data in Direct Push Mode 34
4.4.3. Receive Data from Remote Server in Direct Push Mode 35
4.4.4. Close TCP Client 35
4.5. TCP Server Works in Buffer Access Mode 35
4.5.1. Start a TCP Server 35
4.5.2. Accept TCP Incoming Connection 35
4.5.3. Receive Data from Incoming Connection 36
4.5.4. Close TCP Server 36
4.6. Example of UDP Service 36
4.6.1. Start UDP Service 36
4.6.2. Send UDP Data to Remote 37
4.6.3. Receive Data from Remote 37
4.6.4. Close UDP Service 37
4.7. PING 37
4.8. Synchronize Local Time 38
4.9. Example of Getting Last Error Code 38

Quectel Confidential

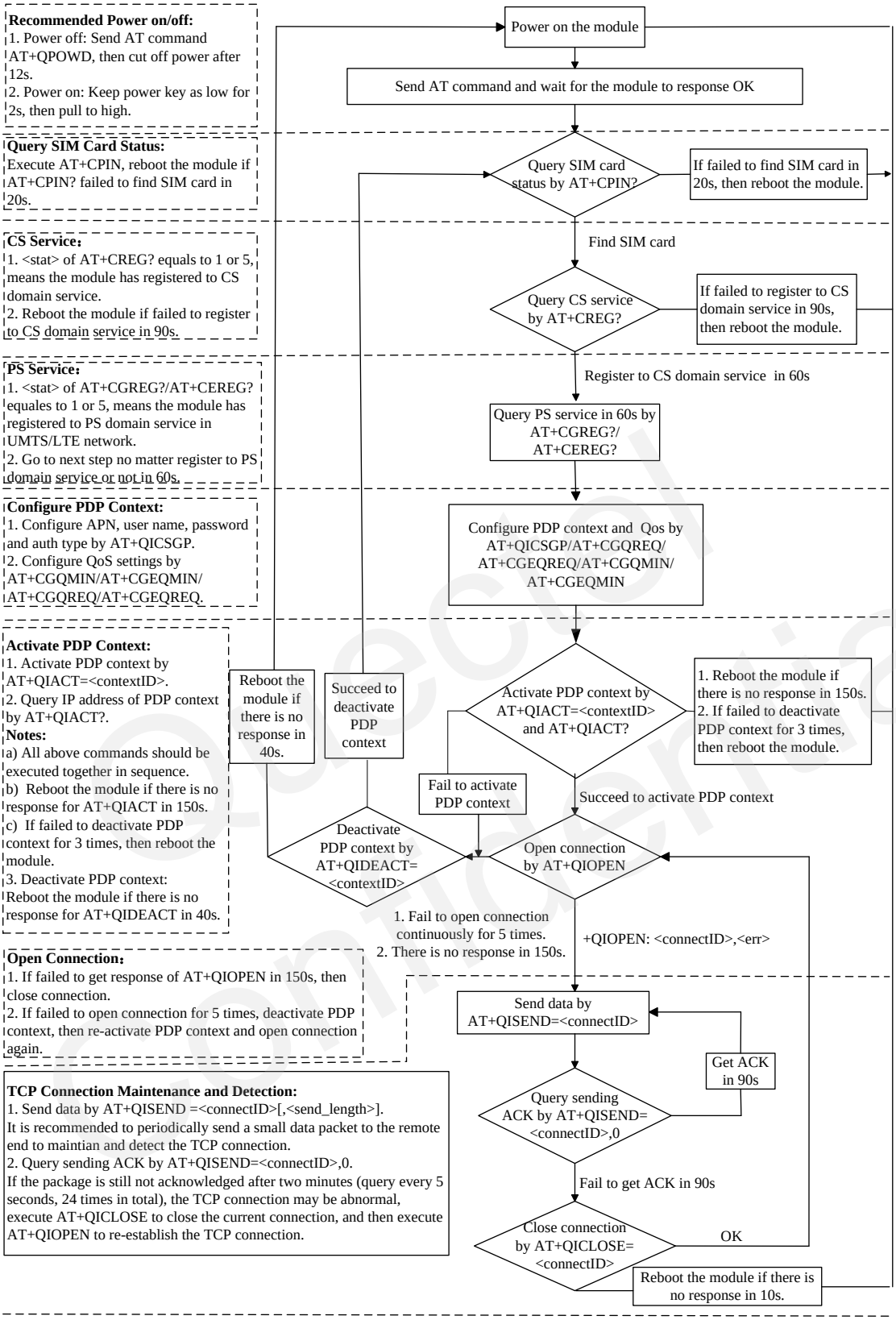
1 Introduction

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 and 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. Figure 1 illustrates how to use TCPIP AT commands:

Quectel Confidential



Notes:

1. Please note that you need to wait for the final response (for example OK, CME error, CMS error) of the last AT command you entered before you enter the next AT command. You can reboot the module if the module failed to get response in 60s.
2. Reboot the module if the module has not got response of AT+QIACT in 150s or response of AT+QICLOSE in 10s and AT+QIDEACT in 40s .
3. When you constantly failed to restart for 3 times, you can restart immediately for the first time after that, if it still fails, you should restart after 10 minutes for the second time, and restart after 30 minutes for the third time, 1 hour for the fourth time, etc..

Figure 1: Flow Chart of Using TCPIP AT Commands

1.2. Description of Data Access Mode

The TCPIP AT commands of the 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 <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>. 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>,<currectrecvlength><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 output 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.

Quectel Confidential

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,<APN>,<username>,<password>,(0-3)** **OK**
Query the configuration of context **AT+QICSGP=<contextID>**	Response **+QICSGP: <context_type>,<APN>,<username>,<password>,<authentication>** **OK**
Write Command, configure the context **AT+QICSGP=<contextID>,<context_type>,[<APN>[,<username>,<password>)[,<authentication>]]]**	Response **OK** **ERROR**

Parameter

<contextID>	Integer type, context ID, range is 1-16
<context_type>	Integer type, protocol type 1 IPV4
<APN>	String type, access point name
<username>	String type, username
<password>	String type, password
<authentication>	Integer type, the authentication methods 0 NONE 1 PAP 2 CHAP 3 PAP_OR_CHAP

Quectel Confidential

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,<context_state>,<context_type>[,<IP_address>]** **[.....** **+QIACT:** **16,<context_state>,<context_type>[,<IP_address>]]** **OK**
Write Command **AT+QIACT=<contextID>**	Response Activate the context of specified <contextID>: **OK** **ERROR**
Maximum Response Time	150 seconds, determined by network.

Quectel Confidential

Parameter

<contextID>	Integer type, context ID, range is 1-16
<context_state>	Integer type, context state 0 Deactivated 1 Activated
<contex_type>	Integer type, protocol type 1 IPV4
<IP_address>	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 <service_type>. The data access mode (buffer access mode, direct push access mode and transparent access mode) can be specified by parameter <access_mode>. The URC "+QIOPEN" indicates whether the socket service is started successfully.

1. If <service_type> 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>,<remote_port>". The range of <connectID> is 0-11. The type of this new incoming connection is "TCP INCOMING" and the <access_mode> of "TCP INCOMING" is the same with the <access_mode> of "TCP LISTENER".

2. If <service_type> is "UDP SERVICE", UDP data can be sent to or received from the remote IP via <local_port>.
   - Send data: AT+QISEND=<connectID>,<send_length>,<remoteIP>,<remote_port>.
   - Receive data in direct push access mode: report a URC as "+QIURC: "recv",<connectID>,<currentrecvlength>,<remoteIP>,<remote_port><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>,<err>". 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","<IP_address>/<domain_name>",<remote_port>,<local_port>,(0-2)** **OK**
Read Command **AT+QIOPEN?**	Response **OK**
Write Command **AT+QIOPEN=<contextID>,<connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>[,<local_port>[,<access_mode>]]**	Response If the <access_mode> 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 <access_mode> is buffer access mode or direct push mode, response: **OK** **+QIOPEN: <connectID>,<err>** <err> is 0 when service is started successfully, else <err> 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
<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 If <service_type> is TCP or UDP, it indicates the IP address of remote server, such as "220.180.239.201" If <service_type> is TCP LISTENER or UDP SERVICE, please enter "127.0.0.1"
<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 0-65535
<local_port>	The local port, range is 0-65535 If <service_type> is "TCP LISTENER" or "UDP SERVICE", this parameter must be specified If <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 2 Transparent access mode
<err>	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 1-65535. 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 <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+QISTATE Query Socket Service Status	
Test Command **AT+QISTATE=?**	Response **OK**
Read Command **AT+QISTATE?**	Response Return the status of all existing connections **List of (+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<serverID>,<access_mode>,<AT_port>)** **OK**
If <query_type> is 0, query by specified <contextID> **AT+QISTATE=<query_type>,<contextID>**	Response Return the status of all existing connections in specified <contextID> **List of (+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<serverID>,<access_mode>,<AT_port>)** **OK**
If <query_type> is 1, query by specified <connectID> **AT+QISTATE=<query_type>,<connectID>**	Response **+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<serverID>,<ac**

	cess_mode>,<AT_port> **OK**

Parameter

<query_type>	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
<service_type>	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
<IP_address>	IP address If <service_type>=“TCP” or “UDP”, <IP_address> is the IP address of remote server If <service_type>=“TCP LISTENER” or “UDP SERVICE”, <IP_address> is local IP address If <service_type>=“TCP INCOMING”, <IP_address> is the IP address of remote client
<remote_port>	Remote port number If <service_type>=“TCP” or “UDP”, <remote_port> is the port of remote server If <service_type>=“TCP LISTENER” or “UDP SERVICE”, <remote_port> is invalid If <service_type>=“TCP INCOMING”, <remote_port> is the port of remote client
<local_port>	Local port number If <local_port> is 0, local port is assigned automatically
<socket_state>	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 <service_type> is “TCP INCOMING”. <serverID> represents which server accepts this TCP incoming connection, the value is the same as <connectID> of this server’s “TCP LISTENER”
<access_mode>	Data access mode 0 Buffer access mode 1 Direct push mode 2 Transparent access mode
<AT_port>	String type, COM port of socket service “usbmodem” USB MODEM PORT

Quectel Confidential

“usbat”	USB AT PORT
“uart1”	UART PORT1
“cmux1”	MUX PORT 1
“cmux2”	MUX PORT 2
“cmux3”	MUX PORT 3
“cmux4”	MUX PORT 4

2.7. AT+QISEND Send Data

If <access_mode> 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	
Test Command **AT+QISEND=?**	Response **+QISEND: (0-11),(0-1460)** **OK**
If<service_type> 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	Response **>** **<input data>** **CTRL+Z** If connection has been established and sending is successful, response: **SEND OK** If connection has been established but sending buffer is full, response: **SEND FAIL** If connection has not been established, abnormally closed, or parameter is incorrect, response: **ERROR**
If<service_type> is “TCP”, “UDP” or “TCP INCOMING”, send data with fixed length	Response **>** **<input data with specified length>**

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

Parameter

<connectID>	Integer type, socket service index, range is 0-11
<send_length>	Integer type, the length of data to be sent, which cannot exceed 1460
<remoteIP>	String type, the remote IP address(must dot format), only valid when

	<service_type> is “UDP SERVICE”
<remote_port>	Integer type, the remote port, only valid when <service_type> is “UDP SERVICE”
<total_send_length>	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
<err>	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.

AT+QIRD Retrieve the Received TCP/IP Data	
Test Command **AT+QIRD=?**	Response **+QIRD: (0-11),(0-1500)** **OK**
When <service_type> is “TCP”/“UDP”/“TCP INCOMING” **AT+QIRD=<connectID>[,<read_length>]**	Response If the specified connection has received data, response: **+QIRD: <read_actual_length><CR><LF><data>** **OK** If no data, return: **+QIRD: 0** **OK** If connection does not exist, return: **ERROR**
When <service_type> is “UDP SERVICE” **AT+QIRD=<connectID>**	Response If data exists, return: **+QIRD: <read_actual_length>,<remoteIP>,<remote_port><CR><LF><data>** **OK** If there is no data, return: **+QIRD: 0**

Quectel Confidential

	OK If connection does not exist, return: **ERROR**
When <read_length> is 0, query retrieved data **AT+QIRD=<connectID>,0**	Response If the specified connection has existed, return: **+QIRD: <total_receive_length>,<have_read_length>,<unread_length>** **OK** Else response: **ERROR**

Parameter

<connectID>	Integer type, the socket service index, the range is 0-11
<read_length>	The maximum length of data to be retrieved, the range is 0-1500
<read_actual_length>	The actual length of received data
<remoteIP>	String type, the remote IP address, only valid when <service_type> is "UDP SERVICE"
<remote_port>	Integer type, the remote port, only valid when <service_type> is "UDP SERVICE"
<data>	The retrieved data
<total_receive_length>	The total length of received data
<have_read_length>	The read length of received data
<unread_length>	The unread length of received data

2.9. AT+QISENDEX Send HEX String

This command cannot be applied for "UDP SERVICE" and "TCP LISTENER" Socket.

AT+QISENDEX Send HEX String	
Test Command **AT+QISENDEX=?**	Response **+QISENDEX: (0-11),<hex_string>** **OK**
Write Command **AT+QISENDEX=<connectID>,<hex_string>**	Response **SEND OK**

	If no data, return: **SEND FAIL** If connection does not exist, return: **ERROR**

Parameter

<connectID>	Integer type, the socket service index, the range is 0-11
<hex_string>	String type, hex string, the max length is 512

2.10. 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 <access_mode> 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 <access_mode> is 0 or 1, response: **OK** If switch successfully and <access_mode> is 2, module will enter into data mode, response: **CONNECT** Else response: **ERROR**

Quectel Confidential

Parameter

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

2.11. 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 4 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** **[+QPING: <result>[,<IP_address>,<bytes>,<time>,<ttl>]<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: 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>"

Quectel Confidential

	Others Refer to 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 the 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>,<max>,<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

2.12. AT+QNTP 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+QNTP Synchronize Local Time with NTP Server	
Test command **AT+QNTP=?**	Response **+QNTP: (1-16),<server>,(list of supported<port>s),(0,1)** **OK**
Read command **AT+QNTP?**	Response If in the process of synchronizing local time: **+QNTP: <server>,<port>** **OK**
Write command **AT+QNTP=<contextID>,<server>[,<port>][,<autosettime>]**	Response If synchronize local time with NTP server successfully: **OK** **+QNTP: <err>,<time>** Else response: **ERROR**
Maximum Response Time	125s, determined by network.

Quectel Confidential

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
<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 “YYYY/MM/DD,hh:mm:ss±zz”. The range of zz is -48~56

2.13. 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

Quectel Confidential

2.14. 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** The result will be returned as URC. **+QIURC: "dnsgip",<err>,<IP_count>,<DNS_ttl>** [..... **+QIURC: "dnsgip",<hostIPaddr>]**
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
<err>	Integer type, 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>

2.15. AT+QICFG Configure Optional Parameters

AT+QICFG Configure Optional Parameters	
Test Command **AT+QICFG=?**	Response **+QICFG: "transpktsize",(1-1460)** **+QICFG: "transwaittm",(0-20)** **+QICFG: "dataformat",(0,1),(0,1)** **+QICFG: "viewmode",(0,1)** **+QICFG: "tcp/retranscfg",(3-20),(5-1000)**

Quectel Confidential

	OK
Set packet size for transparent mode **AT+QICFG="transpktsize"[,<transpktsize>]**	Response **[+QICFG: "transpktsize",<transpktsize>]** **OK** **ERROR**
Set waiting time for transparent mode **AT+QICFG="transwaittm"[,<transwaittm >]**	Response **[+QICFG: "transwaittm",<transwaittm>]** **OK** **ERROR**
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 display format for receiving data (only for non-transparent mode) **AT+QICFG="viewmode"[,<view_mode>]**	Response **[+QICFG: "viewmode",<mode>]** **OK** **ERROR**
Set the maximum interval time and number for TCP retransmission **AT+QICFG="tcp/retranscfg"[,<max_backoffs>,<max_rto>]**	Response **[+QICFG: "tcp/retranscfg",<max_backoffs>,<max_rto>]** **OK** **ERROR**

Parameter

<transpktsize>	Integer type, indicates the max length of sending data packet, range is 1-1460, default: 1024
<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: 100ms, range: 0-20, default: 2
<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

Quectel Confidential

	0 Receive data output format: data header\r\ndata
	1 Receive data output format: data header,data
<max_backoffs>	Integer type, indicates the maximum number of TCP retransmission Range: 3-20, default is 8
<max_rto>	Integer type, indicates the maximum interval time of TCP retransmission Range: 5-1000, default is 600, unit: 100ms

NOTE

The value of AT+QICFG="tcp/retranscfg"[,<max_backoffs>,<max_rto>] cannot be saved to NV, it will take effect to all the TCP sockets.

2.16. AT+QISDE Control Whether or Not to Echo the Data for QISEND

AT+QISDE Control Whether or Not to Echo the Data for QISEND	
Test command **AT+QISDE=?**	Response **+QISDE: (0,1)** **OK**
Read command **AT+QISDE?**	Response **+QISDE: <echo>** **OK**
Execution Command **AT+QISDE=<echo>**	Response **OK** **ERROR**

Parameter

<echo>	A numeric indicates whether or not to echo the data for AT+QISEND
	0 Do not echo the data
	1 Echo the data

Quectel Confidential

2.17. 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: <err>,<errcode_description>** **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

2.18. 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. Actually, there is "<CR><LF>" both before and after URC, but "<CR><LF>" is omitted intentionally.

2.18.1. 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".

URC of Connection Closed	
+QIURC: "closed",<connectID>	<connectID> connection is closed

Parameter

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

Quectel Confidential

2.18.2. URC of Incoming Data

In buffer access mode or direct push mode, after receiving data, the module will report URC to 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 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 <service_type> is “TCP”, “UDP” or “TCP INCOMING”.
+QIURC: “recv”,<connectID>,<currentrecvlength>,<remoteIP>,<remote_port><CR><LF><data>	The URC of data incoming in direct push mode when <service_type> 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
<remote_port>	Remote port
<data>	The received data

2.18.3. 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.

2.18.4. URC of Incoming Connection

If the <service_type> 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 URC as below. The <service_type> of new connection is “TCP INCOMING”, and the

Quectel Confidential

<access_mode> is buffer access mode.

URC of Incoming Connection	
+QIURC: "incoming",<connectID>,<serverID>, <remoteIP>,<remote_port>	When new incoming connection is accepted by <serverID>, the allocated <connectID>, <remoteIP> and <remote_port> 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 whose <service_type> is "TCP LISTENER" and listening socket ID is <serverID>
<remoteIP>	Remote IP address of the incoming <connectID>
<remote_port>	Remote port of the incoming <connectID>

2.18.5. URC of PDP Deactivation

PDP context may be deactivated by the network. The module will report 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

Quectel Confidential

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

<err>	Description of Error Code
0	operate 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 parse failed

Quectel Confidential

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

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
                                          response time is 150s.
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","220.180.239.201",8705,0,0   //Context is 1 and <connectID> is 0. Before
                                                   using AT+QIOPEN, host should activate the
                                                   context with AT+QIACT first.
OK
```

Quectel Confidential

```
+QIOPEN: 0,0                    //Connect successfully. It is suggested to wait 150 seconds for the
                                  URC response as "+QIOPEN: <connectID>,<err>". 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","220.180.239.201",8705,65514,2,1,0,0,"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
AT+QISENDEX=0,"3132333435"      //Send HEX string.
OK
AT+QISEND=0,0                   //Query the length of sent data.
+QISEND: 14,14,0

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.
```

Quectel Confidential

```
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","220.180.239.201",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 “CONNECT”. 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.
```

Quectel Confidential

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","220.180.239.201",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>,<err>". 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","220.180.239.201",8705,65344,2,1,0,1,"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
```

Quectel Confidential

```
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”,0,2020,3,1,1,0,“usbmodem”

OK
```

4.5.2. Accept TCP Incoming Connection

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

Quectel Confidential

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”,0,3030,2,1,2,0,“usbmodem”
```

Quectel Confidential

```
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
```

Quectel Confidential

```
+QPING: 0,"61.135.169.125",32,192,255

+QPING: 0,"61.135.169.125",32,240,255

+QPING: 0,"61.135.169.125",32,241,255

+QPING: 0,"61.135.169.125",32,479,255

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

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","220.180.239.201",8033,0,1     //Start socket service, forget the <connectID>.
ERROR
AT+QIGETERROR
+QIGETERROR: 552, invalid parameters

OK
```