WCDMA UGxx TCPIP

AT Commands Manual

UMTS/HSPA Module Series

Rev. WCDMA_UGxx_TCPIP_AT_Commands_Manual_V1.4

Date: 2016-05-05

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

Quectel Wireless Solutions Co., Ltd.

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, or to report documentation errors, please visit:

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

Or email to: Support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2014-08-01	Chris PENG	Initial
1.1	2015-03-02	Yolanda YAO	1. Changed the document name from "UG95" to "UGxx". 2. Added AT commands: AT+QISDE, AT+QISENDEX, AT+QICFG
1.2	2015-04-01	Yolanda YAO	Updated applicable modules.
1.3	2015-07-09	Chris PENG	Added flow chart of using TCPIP AT Commands
1.4	2016-05-05	Sophie ZHU	Added description about server mode for AT+QISWTMD

Contents

Table Index

1 Introduction

UGxx 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. UGxx provides the following socket services: TCP client, UDP client, TCP server and UDP service.
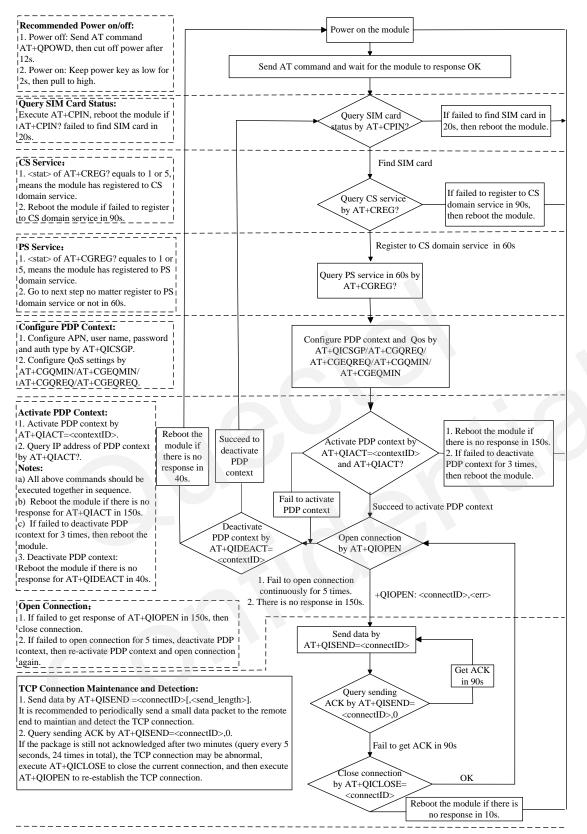
This document is applicable to UGxx modules.

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:

Recommended Power on/off:
1. Power off: Send AT command AT+QPOWD, then cut off power after 12s.
2. Power on: Keep power key as low for 2s, then pull to high.

Query SIM Card Status:
Execute AT+CPIN, reboot the module if AT+CPIN? failed to find SIM card in 20s.

CS Service:
1. <stat> of AT+CREG? equals to 1 or 5, means the module has registered to CS domain service.
2. Reboot the module if failed to register to CS domain service in 90s.

PS Service:
1. <stat> of AT+CGREG? equales to 1 or 5, means the module has registered to PS domain service.
2. Go to next step no matter register to PS domain service or not in 60s.

Configure PDP Context:
1. Configure APN, user name, password and auth type by AT+QICSGP.
2. Configure QoS settings by AT+CGQMIN/AT+CGEQMIN/ AT+CGQREQ/AT+CGEQREQ.

Activate PDP Context:
1. Activate PDP context by AT+QIACT=<contextID>.
2. Query IP address of PDP context by AT+QIACT?.
Notes:
a) All above commands should be executed together in sequence.
b) Reboot the module if there is no response for AT+QIACT in 150s.
c) If failed to deactivate PDP context for 3 times, then reboot the module.
3. Deactivate PDP context: Reboot the module if there is no response for AT+QIDEACT in 40s.

Open Connection:
1. If failed to get response of AT+QIOPEN in 150s, then close connection.
2. If failed to open connection for 5 times, deactivate PDP context, then re-activate PDP context and open connection again.

TCP Connection Maintenance and Detection:
1. Send data by AT+QISEND =<connectID>[,<send_length>].
It is recommended to periodically send a small data packet to the remote end to maintian and detect the TCP connection.
2. Query sending ACK by AT+QISEND=<connectID>,0.
If the package is still not acknowledged after two minutes (query every 5 seconds, 24 times in total), the TCP connection may be abnormal, execute AT+QICLOSE to close the current connection, and then execute AT+QIOPEN to re-establish the TCP connection.

Power on the module

Send AT command and wait for the module to response OK

Query SIM card status by AT+CPIN? → If failed to find SIM card in 20s, then reboot the module.

Find SIM card

Query CS service by AT+CREG? → If failed to register to CS domain service in 90s, then reboot the module.

Register to CS domain service in 60s

Query PS service in 60s by AT+CGREG?

Configure PDP context and Qos by AT+QICSGP/AT+CGQREQ/ AT+CGEQREQ/AT+CGQMIN/ AT+CGEQMIN

Activate PDP context by AT+QIACT=<contextID> and AT+QIACT? → 1. Reboot the module if there is no response in 150s. 2. If failed to deactivate PDP context for 3 times, then reboot the module.

Reboot the module if there is no response in 40s.

Succeed to deactivate PDP context

Fail to activate PDP context

Succeed to activate PDP context

Deactivate PDP context by AT+QIDEACT=<contextID>

Open connection by AT+QIOPEN

1. Fail to open connection continuously for 5 times.
2. There is no response in 150s.

+QIOPEN: <connectID>,<err>

Send data by AT+QISEND=<connectID>

Get ACK in 90s

Query sending ACK by AT+QISEND=<connectID>,0

Fail to get ACK in 90s

Close connection by AT+QICLOSE=<connectID> → OK

Reboot the module if there is no response in 10s.

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 UGxx includes 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, you can 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 a 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 output 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. 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. Use "+++" or DTR (AT&D1 should be set) to exit from the transparent access mode. 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 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 context 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:** **(1-20),1-IPV4,<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 1-20
<context_type>	Integer type, protocol type
	<u>1</u> IPV4
<apn>	String type, access point name
<username>	String type, user name
<password>	String type, password
<authentication>	Integer type, the authentication methods
	<u>0</u> NONE
	1 PAP

2	CHAP
3	PAP_OR_CHAP

Example

AT+QICSGP=1	//Query the configurations 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 1-20, 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: (1-20)** **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:** **20,<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.

Parameter

<contextID>	Integer type, context ID, range is 1-20
<context_state>	Integer type, context state
	0 Deactivated
	1 Activated
<context_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 **+QIDEACT: (1-20)** **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 1-20

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 or not 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 URC as +QIURC: "incoming",<connectID>,<serverID>,<remoteIP>,<remote_port>. The range of <connectID> is 0-11. The <service_type> for this new incoming connection is "TCP INCOMING" and the <access_mode> is buffer access mode.

2. If <service_type> is "UDP SERVICE", UDP data can be sent to and 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 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. The maximum timeout of TCP connect is 75 seconds.

AT+QIOPEN Startup Socket Service	
Test Command **AT+QIOPEN=?**	Response **+QIOPEN: (1-20),(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 set up successfully, else <err> is not 0.
Maximum Response Time	150 seconds, determined by network.

Parameter

<contextID>	Integer type, context ID, range is 1-20
<connectID>	Integer type, socket service index, range is 0-11
<service_type>	String type, socket service type
	"TCP" Start up a TCP connection as a client
	"UDP" Start up a UDP connection as a client
	"TCP LISTENER" Start up a TCP server to listen TCP connection
	"UDP SERVICE" Start up a UDP service
<IP_address>	String type
	If <service_type> is TCP or UDP, it indicates the IP address of remote server, such as "124.74.41.170"
	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"
<local_port>	The local port
	If <service_type> is "TCP LISTENER" or "UDP SERVICE", this parameter must be specified
	If <service_type> is "TCP" or "UDP", <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
	<u>0</u> Buffer access mode
	1 Direct push mode
	2 Transparent access mode
<err>	Integer type, error code. Please refer to Chapter 3.

2.5. AT+QICLOSE   Close Socket Service

Close the specified socket service by AT+QICLOSE. Depending on the Network, it may take some time 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 **+QICLOSE: (0-11) ,(1-60)** **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 1-60, default value is 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**
If <query_type> is 0, query by specified <contextID> **AT+QISTATE=<query_type=0>,<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=1>,<connectID>**	Response **+QISTATE:** **<connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<serverID>,<access_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 1-20
<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 "SSL handshake" SSL handshake
	3 "Connected" Client/incoming connection has been established
	4 "Listening" Server is listening
	5 "Closing" Connection is closing
<serverID>	Only valid when <service_type> is "TCP INCOMING". <service_type> represents which server accepts this TCP incoming connection, the value is the same as <connectID> for 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 MODME PORT
	"usbatport1"	USB AT PORT1
	"usbatport2"	USB AT PORT2
	"uart1"	UART PORT1

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 module successfully, return "SEND OK", else return "SEND FAIL". "SEND FAIL" indicates the process of sending data is blocked. 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 successfully, 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	Response **>**

length **AT+QISEND=<connectID>,<send_length>** Response ">", type data until the data length is equal to <send_length>	**<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**
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 data to specified remote IP address and remote port. The <service_type> must be "UDP SERVICE". **>** **<input data with specified length>** If connection established and sending succeeds, 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 exists, return: **+QISEND:** **<toatal_send_length>,<acklength>,<unacklength>** **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, only valid when <service_type> is "UDP SERVICE"
<remote_port>	Integer type, the remote port, only valid when <service_type> is "UDP SERVICE"
<toatal_send_length>	Integer type, the total length of sent data
<acklength>	Integer type, the total length of acknowledged data
<unacklength>	Integer type, the total length of unacknowledged data

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

In buffer access mode, after receiving data, the module will buffer it and report URC as +QIURC: "recv",<connectID> to notify 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 <read_length> should be ignored; and for the TCP socket, the <read_length> must be specified.

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" or "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 exits, return: **+QIRD:**

	<read_actual_length>,<remoteIP>,<remote_port><CR><LF><data> **OK** If no data, return: **+QIRD: 0** **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 exists, response: **+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 real 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),<hexstring>** **OK**
Write Command **AT+QISENDEX=<connectID>,<hexstring>**	Response **SEND OK** If no data, return: **SEND FAIL** If connection does not exist, return: **ERROR**

Parameter

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

2.10. 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: <m>** **OK**
Write Command **AT+QISDE=<m>**	Response **OK** **ERROR**

Parameter

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

2.11. 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 started, host can change the data access mode by AT+QISWTMD. If the module work in server mode, AT+QISWTMD can switch the data access mode of server, and the <connectID> should be given as client incoming connection ID.

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**

Parameter

<connectID>		Integer type, the client connection ID, 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

2.12. 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: (1-20),<host>,(1-255),(1-10)** **OK**
Write Command **AT+QPING=<contextID>,<host>[,<timeout>[,<pingnum>]]**	Response If ping successfully, return: **OK** **[<CR><LF>+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 1-20
<host>	The host address in string style, 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 ",<IP_address>,<bytes>,<time>,<ttl>"
	1　　　　Timeout for the ping request. In the case, no other information followed
<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>	The value of Time To Live 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>"

	For other value, please 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 were timeout
<min>	The minimum response time. Unit: ms
<max>	The maximum response time. Unit: ms
<avg>	The average response time. Unit: ms

2.13. AT+QNTP   Synchronize Local Time With NTP Server

NTP is intended to synchronize the Universal Time Coordinated (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: (1-20),<server>,(1-65535),(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>][,<auto_set_time>]**	Response If synchronize local time with NTP server successfully: **OK** **+QNTP: <err>,<time>** Else response: **ERROR**
Maximum Response Time	125s, determined by network.

Parameter

<contextID>	Integer type, indicates context ID, range is 1-20
<server>	String type, indicates the address of NTP server
<port>	Integer type, indicates the port of NTP server
<auto_set_time>	Integer type, indicates whether auto set synchronized time to local time

	0	Not set
	<u>1</u>	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	

2.14. 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: (1-20),<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 1-20
<pridnsaddr>	String type, indicates primary DNS server address
<secdnsaddr>	String type, indicates secondary DNS server address

2.15. 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: (1-20),<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 1-20
\<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\>

2.16. AT+QICFG　Configure Optional Parameters

AT+QICFG　Configure Optional Parameters	
Test Command **AT+QICFG=?**	Response **+QICFG: "tcpipcmdset",(0,1)** **+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)** **OK**
Set type of TCP commands **AT+QICFG="tcpipcmdset"[,\<tcpipcmdset\>]**	Response **[+QICFG: "tcpipcmdset ",\<tcpipcmdset\>]** **OK** **ERROR**
Set packet size for transparent mode **AT+QICFG="transpktsize"[,\<tran**	Response **[+QICFG: "transpktsize",\<transpktsize\>]**

spktsize>]	
	OK **ERROR**
Set waiting time for transparent mode **AT+QICFG="transwaittm"[,<tran swaittm >]**	Response **[+QICFG: "transwaittm",<transwaittm>]** **OK** **ERROR**
Set the data format for sending and receiving **AT+QICFG="dataformat"[,<send _data_format>,<recv_data_form at>]**	Response **[+QICFG: "dataformat",<send_data_format>,<recv_data_format>]** **OK** **ERROR**
Set the display format for receiving data **AT+QICFG="viewmode"[,<mode >]**	Response **[+QICFG: "viewmode",<mode>]** **OK** **ERROR**
Set the maximum interval time and number for TCP retransmission **AT+QICFG="tcp/retranscfg"[,<m ax_backoffs>,<max_rto>]**	Response **[+QICFG: "tcp/retranscfg",<max_backoffs>,<max_rto>]** **OK** **ERROR**

Parameter

<tcpipcmdset>	Integer type, set type of TCP commands 0 GSM TCPIP AT COMMAND SET 1 WCDMA TCPIP AT COMMAND SET
<transpktsize>	Integer type, indicates the maximum 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, indicates the format for sending data, only used to AT+QISEND command. 0 TEXT Mode 1 HEX String Mode
<recv_data_format>	Integer type, indicates the format for receiving data, only valid in non-transparent mode. 0 TEXT Mode

	1	HEX String Mode
<mode>	Integer type, indicates the display format for receiving data, only valid in non-transparent mode.	
	<u>0</u>	Separated the header and the content by CRLF
	1	Separated the header and the content by comma
<max_backoffs>	Integer type, indicates the maximum number of TCP retransmission Range: 3-20, default is 12	
<max_rto>	Integer type, indicates the maximum interval time of TCP retransmission Range: 5-1000, default is 600, unit: 100ms	

NOTES

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

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 connection closed, incoming data and incoming connection and so on. Actually, there are "<CR><LF>" both before and after URC, but "<CR><LF>" are omitted intentionally in this document.

2.18.1. URC of Connection Closed

When TCP socket service is closed by remote peer or network error, this report will be output. 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

2.18.2. 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 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 output 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>,<current recvlength><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>,<current recvlength>,<remoteIP>,<remote_port><CR><LF><data>	The URC of incoming data 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 \<connectID\> is 0-11. The module will report a URC as below. The \<service_type\> of new connection is "TCP INCOMING", and the \<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 which \<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 network. UGxx will report a URC to the host. Host should execute AT+QIDEACT to deactivate the context and reset all connections.

URC of PDP Deactivated

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

Parameter

<contextID>   Integer type, context ID, range is 1-20

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 parameter
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	activate PDP context failed
562	close PDP context failed
563	socket Identity has been used
564	DNS busy
565	DNS parse failed

566	socket connect failed
567	socket has been closed
568	operation busy
569	operation timeout
570	PDP context break down
571	cancel send
572	operation not allowed
573	APN not configured

4 Examples

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

AT+QIOPEN=1,1,"TCP","220.180.239.201",8713,0,0 //Context is 1 and <connectID> is 1. Before using
                                          AT+QIOPEN, host should activate the context

```
                                              with AT+QIACT first.
OK


+QIOPEN: 1,0                          //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=1,1                        //Query if connection state of <connectID> is 1.
+QISTATE: 1,"TCP","220.180.239.201",8713,0,3,1,0,0,"usbmodem"

OK
```

4.2.2. Send Data in Buffer Access Mode

```
AT+QISEND=1                           //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=1,4                         //Send fixed length data and the data length is 4.
>test
SEND OK
AT+QISEND=1,0                         //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",1                      //The <connectID> 1 received data.
AT+QIRD=1,1500                        //Read data, the length is 1500.
+QIRD: 5                              //The actual received data length is 5.
test1

OK
AT+QIRD=1,1500
+QIRD: 0                              //No data in buffer.

OK
AT+QIRD=1,0                           //Query length of total received data, including read and
```

unread.

+QIRD: 5,5,0

OK

4.2.4. Close Connection

AT+QICLOSE=1	//Close a connection whose <connectID> is 1. Depending on the Network, the maximum response time is 10s.
OK	

4.3. TCP Client Works in Transparent Access Mode

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

AT+QIOPEN=1,1,"TCP","220.180.239.201",8713,0,2	//Context is 1 and <connectID> is 1. 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>,<err>". 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

test1	//All data received from internet will be output to COM port directly.

4.3.4. Close TCP Client

AT+QICLOSE=1	//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 and Enter into Direct Push Mode

AT+QIOPEN=1,1,"TCP","220.180.239.201",8713,0,1	//Context is 1 and \<connectID\> is 1. Before using AT+QIOPEN, host should activate the context with AT+QIACT first.
OK	
+QIOPEN: 1,0	//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=1,1	//Query if connection state of \<connectID\> is 1.
+QISTATE: 1,"TCP","220.180.239.201",8713,0,3,1,0,1,"usbmodem"	
OK	

4.4.2. Send Data in Direct Push Mode

AT+QISEND=1	//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=1,5	//Send fixed length data and the data length is 5.
>test2	
SEND OK	
AT+QISEND=1,0	//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",1,4   //Receive data from remote server.
test

4.4.4. Close TCP Client

AT+QICLOSE=1   //Close the connection of which <connectID>
                                                     is 1. Depending on the Network, the
                                                     maximum response time is 10s.
OK

4.5. TCP Server Works in Buffer Access Mode

4.5.1. Start up 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 up TCP server successfully.

AT+QISTATE=0,1   //Query if connection state of <contextID> 1.
+QISTATE: 1,"TCP LISTENER","10.7.157.1",0,2020,4,1,0,0,"usbmodem"

OK

4.5.2. Accept TCP Incoming Connection

+QIURC: "incoming",11,1,"172.31.242.222",54091 //Accept a TCP connection, <service_type> 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 length of total received data, 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 up UDP Service

AT+QIOPEN=1,2,"UDP SERVICE","127.0.0.1",0,3030,0	//Start up 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 up UDP service successfully.
AT+QISTATE=0,1	//Query connection state of <contextID> 1.
+QISTATE: 2,"UDP SERVICE","10.7.157.1",0,3030,3,1,0,0,"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 whose 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 output. 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.

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: 0,4,4,0,96,147,111

4.8. Synchronize Local Time

AT+QNTP=1,"202.120.2.101",123	//Synchronize local time with NTP server "202.120.2.101". 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",8713,0,1	//Start socket service, forget the <connectID>.
ERROR	
AT+QIGETERROR	
+QIGETERROR: 552, invalid parameters	
OK	