

BG96 TCP/IP

AT Commands Manual

LTE Module Series

Rev. BG96_TCP/IP_AT_Commands_Manual_V1.0

Date: 2017-11-23

Status: Released



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

Quectel Wireless Solutions Co., Ltd.

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

Tel: +86 21 5108 6236

Email: info@quectel.com

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

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

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

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

Or email to: support@quectel.com

GENERAL NOTES

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

COPYRIGHT

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

Copyright © Quectel Wireless Solutions Co., Ltd. 2017. All rights reserved.

About the Document

History

Revision	Date	Author	Description
1.0	2017-11-23	Walker HAN/ Lucifer YAN	Initial

Contents

About the Document.....	2
Contents.....	3
Table Index.....	5
1 Introduction	6
1.1. The Process of Using TCP/IP AT Commands	6
1.2. Description of Data Access Modes.....	8
2 Description of TCP/IP AT Commands	10
2.1. Description of AT Commands	10
2.1.1. AT+QICSGP Configure Parameters of a TCP/IP Context	10
2.1.2. AT+QIACT Activate a PDP Context	11
2.1.3. AT+QIDEACT Deactivate a PDP Context.....	12
2.1.4. AT+QIOPEN Open a Socket Service.....	13
2.1.5. AT+QICLOSE Close a Socket Service.....	15
2.1.6. AT+QISTATE Query Socket Service Status.....	15
2.1.7. AT+QISEND Send Data	17
2.1.8. AT+QIRD Retrieve the Received TCP/IP Data	19
2.1.9. AT+QISENDEX Send Hex String	21
2.1.10. AT+QISWTMD Switch Data Access Modes.....	21
2.1.11. AT+QPING Ping a Remote Server.....	22
2.1.12. AT+QNTP Synchronize Local Time with NTP Server.....	23
2.1.13. AT+QIDNSCFG Configure Address of DNS Server	24
2.1.14. AT+QIDNSGIP Get IP Address by Domain Name.....	25
2.1.15. AT+QICFG Configure Optional Parameters	26
2.1.16. AT+QISDE Control Whether to Echo the Data for AT+QISEND.....	27
2.1.17. AT+QIGETERROR Query the Last Error Code.....	28
2.2. Description of URC	28
2.2.1. URC of Connection Closed.....	29
2.2.2. URC of Incoming Data	29
2.2.3. URC of Incoming Connection Full.....	30
2.2.4. URC of Incoming Connection	30
2.2.5. URC of PDP Deactivation	30
3 Examples	32
3.1. Configure and Activate a Context	32
3.1.1. Configure a Context	32
3.1.2. Activate a Context.....	32
3.1.3. Deactivate a Context.....	32
3.2. TCP Client Works in Buffer Access Mode	33
3.2.1. Set up a TCP Client Connection and Enter into Buffer Access Mode.....	33
3.2.2. Send Data in Buffer Access Mode	33
3.2.3. Receive Data from Remote Server in Buffer Access Mode.....	34

3.2.4.	Close a Connection.....	34
3.3.	TCP Client Works in Transparent Access Mode.....	34
3.3.1.	Set up a TCP Client Connection and Enter into Transparent Access Mode	34
3.3.2.	Send Data in Transparent Access Mode	34
3.3.3.	Receive Data from Remote Server in Transparent Access Mode	35
3.3.4.	Close a TCP Client.....	35
3.4.	TCP Client Works in Direct Push Mode.....	35
3.4.1.	Set up a TCP Client Connection and Enter into Direct Push Mode	35
3.4.2.	Send Data in Direct Push Mode.....	35
3.4.3.	Receive Data from Remote Server in Direct Push Mode	36
3.4.4.	Close a TCP Client.....	36
3.5.	TCP Server Works in Buffer Access Mode	36
3.5.1.	Start a TCP Server	36
3.5.2.	Accept TCP Incoming Connection	37
3.5.3.	Receive Data from Incoming Connection	37
3.5.4.	Close a TCP Server	37
3.6.	Example of UDP Service	37
3.6.1.	Start a UDP Service	37
3.6.2.	Send UDP Data to Remote	38
3.6.3.	Receive Data from Remote.....	38
3.6.4.	Close a UDP Service	38
3.7.	PING.....	39
3.8.	Synchronize Local Time.....	39
3.9.	Example of Getting Last Error Code.....	39
4	Summary of Error Codes	40
5	Appendix A Reference.....	42

Table Index

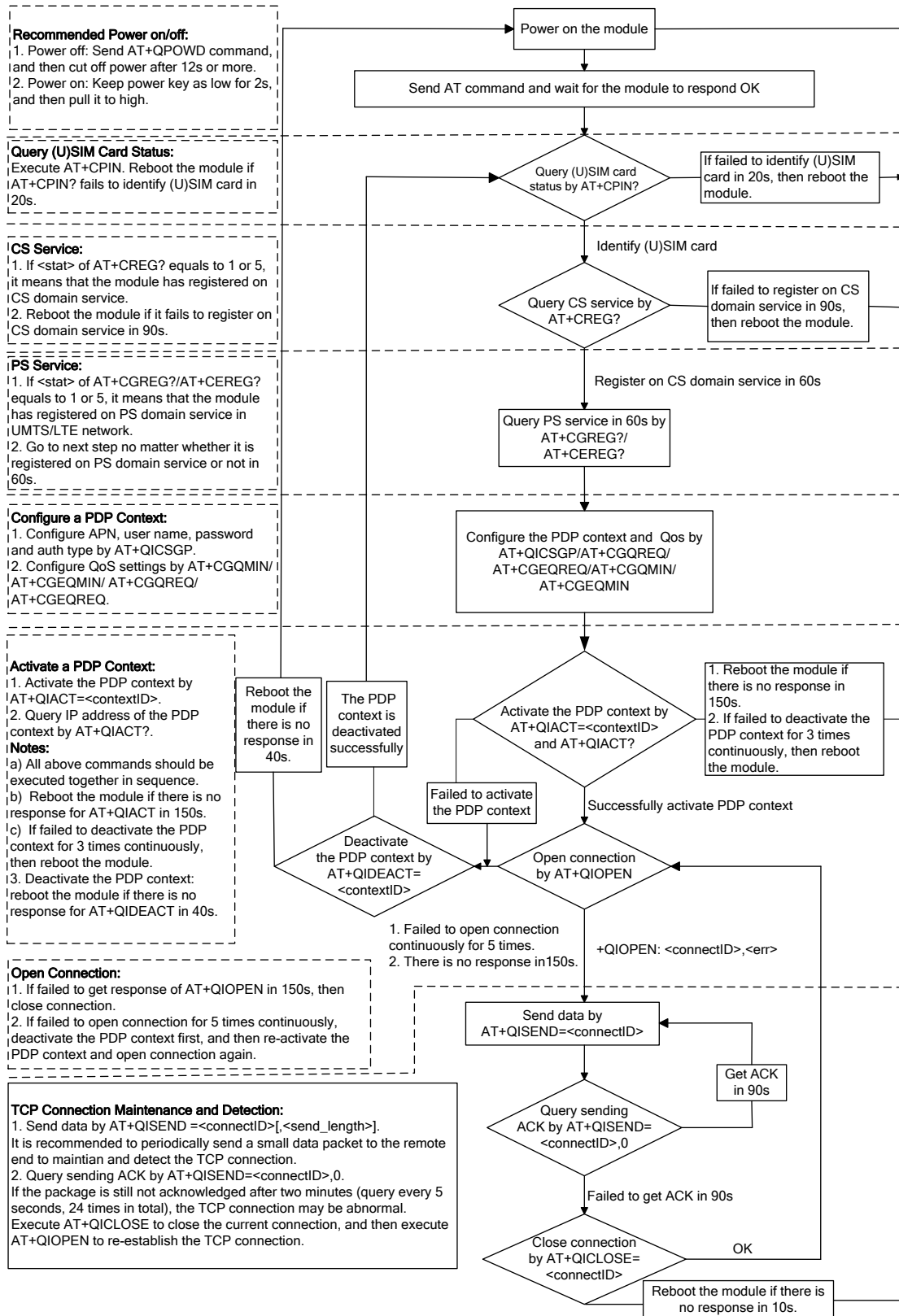
TABLE 1: SUMMARY OF ERROR CODES	40
TABLE 2: TERMS AND ABBREVIATIONS	42

1 Introduction

Quectel BG96 module features embedded TCP/IP stack, which enables the host to access the Internet directly via AT commands. This greatly reduces the dependence on the PPP and TCP/IP protocol stacks and thus minimizes the cost. BG96 module provides the following socket services: TCP client, UDP client, TCP server and UDP server.

1.1. The Process of Using TCP/IP AT Commands

Through TCP/IP AT commands, the host can configure a PDP context, activate/deactivate the PDP context, start/close socket service and send/receive data via socket service. The following figure illustrates how to use TCP/IP AT commands.



Notes:

- 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 fails to get response in 60s.
- 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.
- It is NOT recommended to frequently reboot the module. When the module has been continuously rebooted for 3 times due to failed AT command execution, it can be rebooted immediately for the first time after that. If it still fails, reboot the module after 10 minutes for the second time, and reboot after 30 minutes for the third time, one hour for the fourth time, etc.

Figure 1: Flow Chart of Using TCP/IP AT Commands

1.2. Description of Data Access Modes

BG96 module supports the following three kinds of data access modes:

- Buffer access mode
- Direct push mode
- Transparent access mode

When opening a socket service via AT+QIOPEN, the data access mode can be specified by the parameter <access_mode>. After a socket service is opened, customers can switch the access mode via AT+QISWTMD.

1. In buffer access mode, data can be sent via AT+QISEND command, and if the module has received data from the Internet, it will buffer the data and report a URC as "+QIURC: "recv",<connectID>". Customers can read data via AT+QIRD command.
2. In direct push mode, data can be sent via AT+QISEND command, and if the module has received data from the Internet, the data will be outputted to COM port directly in the following format:
"+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 the Internet directly, and the data received from Internet will be outputted via COM port directly. Customers can use "+++" to exit from transparent access mode. When "OK" is returned, the module will be switched to buffer access mode. AT+QISWTMD can be used to return back to transparent access mode.
4. To exit from the transparent access mode, "+++" or DTR (AT&D1 should be set first) can be used. To prevent the "+++" from being misinterpreted as data, the following sequence should be followed:
 - 1) Do not input any character within 1s or longer before inputting "+++".
 - 2) Input "+++" within 1s, and no other characters can be inputted during the time.
 - 3) Do not input any character within 1s after "+++" has been inputted.
 - 4) Use "+++" or DTR (AT&D1 should be set first) to make the module exit from transparent access mode, and wait until "OK" is returned.
5. There are two methods to return back to the transparent access mode:
 - 1) By AT+QISWTMD. Specify the <access_mode> as 2 when executing this command. If entering 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 entering into transparent access mode successfully, "CONNECT" will be returned. If there is no connection entering into transparent access mode before, ATO will return "NO CARRIER".

NOTES

1. In buffer access mode, if the buffer is not empty, the module will not report a new URC until all the received data has been read via AT+QIRD from buffer.
2. In transparent access mode, AT commands cannot be executed. If the socket connection is closed because of network error or other errors, 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.

2 Description of TCP/IP AT Commands

2.1. Description of AT Commands

2.1.1. AT+QICSGP Configure Parameters of a TCP/IP Context

The command can be used to configure the <APN>, <username>, <password> and other parameters of a TCP/IP context.

AT+QICSGP Configure Parameters of a TCP/IP Context

Test Command AT+QICSGP=?	Response +QICSGP: (1-16),(1,2),<APN>,<username>,<password>,(0-3) OK
Write Command Query the configuration of a 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. The context ID. The range is 1-16.
<context_type>	Integer type. The protocol type. 1 IPV4 2 IPV6*
<APN>	String type. The access point name.
<username>	String type. The username.
<password>	String type. The password.

<authentication>	Integer type. The authentication methods.
0	NONE
1	PAP
2	CHAP
3	PAP or CHAP

Example

```

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

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

```

NOTE

"*" means under development.

2.1.2. AT+QIACT Activate a PDP Context

Before activating a PDP context via AT+QIACT, the context should be configured by AT+QICSGP. After activation, the IP address can be queried via AT+QIACT?. The range of <contextID> is 1-16. According to 3GPP specifications, the module only supports three PDP contexts activated simultaneously, so customers must make sure the number of activated PDP contexts is less than 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 a PDP Context

Test Command AT+QIACT=?	Response +QIACT: (1-16) OK
Read command AT+QIACT?	Response Return the list of the current activated contexts and their IP addresses: +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 specified context: OK Or ERROR
Maximum Response Time	150 seconds, determined by network.

Parameter

<contextID>	Integer type. The context ID. The range is 1-16.
<context_state>	Integer type. The context state. 0 Deactivated 1 Activated
<context_type>	Integer type. The protocol type. <u>1</u> IPV4 2 IPV6*
<IP_address>	The local IP address after the context is activated.

NOTE

“*” means under development.

2.1.3. AT+QIDEACT Deactivate a PDP Context

The command is used to deactivate the specific context and close all TCP/IP 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 a PDP Context

Test Command AT+QIDEACT=?	Response +QIDEACT: (1-16) OK
Write Command AT+QIDEACT=<contextID>	Response OK Or ERROR

Maximum Response Time	40 seconds, determined by network.
-----------------------	------------------------------------

Parameter

<contextID>	Integer type. The context ID. The range is 1-16.
-------------	--

2.1.4. AT+QIOPEN Open a Socket Service

The command is used to open a socket service. The service type can be specified by <service_type> parameter. The data access mode (buffer access mode, direct push mode and transparent access mode) can be specified by <access_mode> parameter. The "+QIOPEN" URC indicates whether the socket service has been opened successfully.

- If <service_type> is "TCP LISTENER", the module works as TCP SERVER. After accepting a new TCP connection, the module will automatically 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".
- If <service_type> is "UDP SERVICE", UDP data can be sent to or received from the remote IP via <local_port>.
 - Send data: execute AT+QISEND=<connectID>,<send_length>,<remoteIP>,<remote_port>.
 - Receive data in direct push mode: the module reports a URC as "+QIURC: "recv",<connectID>,<currentrecvlength>,<remoteIP>,<remote_port><CR><LF><data>".
 - Receive data in buffer access mode: the module reports a URC as "+QIURC: "recv",<connectID>", and then customers can retrieve data via AT+QIRD=<connectID>.
- It is suggested to wait for 150 seconds for "+QIOPEN: <connectID>,<err>" to be outputted. If the URC cannot be received in 150 seconds, AT+QICLOSE should be used to close the socket.

AT+QIOPEN Open a 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
Write Command AT+QIOPEN=<contextID>,<connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>[,<local_po	Response If the <access_mode> is transparent access mode and it is successful to open the service, response: CONNECT

rt>[,<access_mode>]]	<p>If there is any error, response: ERROR Error description can be got via AT+QIGETERROR.</p> <p>If the <access_mode> is buffer access mode or direct push mode, response: OK</p> <p>+QIOPEN: <connectID>,<err> <err> is 0 when service is opened successfully. In other cases, <err> is not 0.</p>
Maximum Response Time	150 seconds, determined by network.

Parameter

<contextID>	Integer type. The context ID. The range is 1-16.								
<connectID>	Integer type. The socket service index. The range is 0-11.								
<service_type>	String type. The socket service type. <table border="0" style="width: 100%;"> <tr> <td style="padding-right: 20px;">“TCP”</td> <td>Start a TCP connection as a client</td> </tr> <tr> <td>“UDP”</td> <td>Start a UDP connection as a client</td> </tr> <tr> <td>“TCP LISTENER”</td> <td>Start a TCP server to listen to TCP connection</td> </tr> <tr> <td>“UDP SERVICE”</td> <td>Start a UDP service</td> </tr> </table>	“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
“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 “220.180.239.212”. 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”. The range is 0-65535.								
<local_port>	The local port. The 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. Otherwise the local port is assigned as specified.								
<access_mode>	Integer type. The data access mode of the socket service. <table border="0" style="width: 100%;"> <tr> <td style="padding-right: 20px;">0</td> <td>Buffer access mode</td> </tr> <tr> <td>1</td> <td>Direct push mode</td> </tr> <tr> <td>2</td> <td>Transparent access mode</td> </tr> </table>	0	Buffer access mode	1	Direct push mode	2	Transparent access mode		
0	Buffer access mode								
1	Direct push mode								
2	Transparent access mode								
<err>	Integer type. The error code of the operation. Please refer to Chapter 4 .								

2.1.5. AT+QICLOSE Close a Socket Service

The command is used to close the specified socket service. Depending on the network, it will take at most 10 seconds (default value, can be modified by <timeout>) to return “OK” or “ERROR” after executing AT+QICLOSE. Before the response is returned, other AT commands cannot be executed.

AT+QICLOSE Close a Socket Service

Test Command AT+QICLOSE=?	Response +QICLOSE: (0-11),(0-65535) OK
Write Command AT+QICLOSE=<connectID>[,<timeout>]	Response If successfully closed, response: OK If failed to close, response: ERROR

Parameter

<connectID>	Integer type. The socket service index. The range is 0-11.
<timeout>	Integer type. If the FIN ACK of the other peers is not received within <timeout>, the module will force to close the socket. The range is 1-65535, and the default value is 10. Unit: second.

2.1.6. AT+QISTATE Query Socket Service Status

The command is 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 specified context. If the <query_type> is 1, it will return the status of the specified socket service.

AT+QISTATE Query Socket Service Status

Test Command AT+QISTATE=?	Response OK
Read/Execution Command AT+QISTATE? or 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>,<ac cess_mode>,<AT_port>) OK

<p>Write Command</p> <p>If <query_type> is 0, query connection status of the specified context</p> <p>AT+QISTATE=<query_type>,<context ID></p>	<p>Response</p> <p>Return the status of all existing connections in specified context:</p> <p>List of (+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port> ,<local_port>,<socket_state>,<contextID>,<serverID>,<access_mode>,<AT_port>)</p> <p>OK</p>
<p>Write Command</p> <p>If <query_type> is 1, query connection status of the specified socket service</p> <p>AT+QISTATE=<query_type>,<connect ID></p>	<p>Response</p> <p>+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port> ,<local_port>,<socket_state>,<contextID>,<serverID>,<access_mode>,<AT_port></p> <p>OK</p>

Parameter

<p><query_type></p>	<p>Integer type. The query type.</p> <p>0 Query connection status of the specified context</p> <p>1 Query connection status of the specified socket service</p>
<p><contextID></p>	<p>Integer type. The context ID. The range is 1-16.</p>
<p><connectID></p>	<p>Integer type. The socket service index. The range is 0-11.</p>
<p><service_type></p>	<p>String type. The socket service type.</p> <p>“TCP” Start a TCP connection as a client</p> <p>“UDP” Start a UDP connection as a client</p> <p>“TCP LISTENER” Start a TCP server to listen to TCP connection</p> <p>“TCP INCOMING” Start a TCP connection accepted by a TCP server</p> <p>“UDP SERVICE” Start a UDP service</p>
<p><IP_address></p>	<p>IP address.</p> <p>If <service_type>=“TCP” or “UDP”, it is the IP address of remote server.</p> <p>If <service_type>=“TCP LISTENER” or “UDP SERVICE”, it is the local IP address.</p> <p>If <service_type>=“TCP INCOMING”, it is the IP address of remote client.</p>
<p><remote_port></p>	<p>Remote port number.</p> <p>If <service_type>=“TCP” or “UDP”, it is the port of remote server.</p> <p>If <service_type>=“TCP LISTENER” or “UDP SERVICE”, the port is invalid.</p> <p>If <service_type>=“TCP INCOMING”, it is the port of remote client.</p>
<p><local_port></p>	<p>Local port number.</p> <p>If <local_port> is 0, then the local port is assigned automatically.</p>
<p><socket_state></p>	<p>Integer type. The socket service status.</p> <p>0 “Initial”: connection has not been established</p> <p>1 “Opening”: client is connecting or server is trying to listen</p>

	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, and 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
	“usbat”	USB AT PORT
	“uart1”	UART PORT1

2.1.7. AT+QISEND Send Data

If <access_mode> of the specified socket service is buffer access mode or direct push mode, data can be sent via AT+QISEND. If the data is sent to the module successfully, “SEND OK” will be returned. Otherwise it will return “SEND FAIL” or “ERROR”. “SEND FAIL” indicates the sending buffer is full and customers can try to resend the data. “ERROR” indicates it encounters an error in the process of sending data. Customers should delay some time for the data to be sent. The maximum data length is 1460 bytes. “SEND OK” does not mean the data has been sent to the server successfully. Customers can query whether the data has reached the server by AT+QISEND=<connectID>, 0 command.

AT+QISEND Send Data

<p>Test Command</p> <p>AT+QISEND=?</p>	<p>Response</p> <p>+QISEND: (0-11),(0-1460)</p> <p>OK</p>
<p>Write Command</p> <p>If <service_type> is “TCP”, “UDP” or “TCP INCOMING”, send data with changeable length</p> <p>AT+QISEND=<connectID></p> <p>After response “>”, input the data to be sent. Tap CTRL+Z to send, and tap ESC to cancel the operation</p>	<p>Response</p> <p>></p> <p><input data></p> <p>CTRL+Z</p> <p>If connection has been established and sending is successful, response:</p> <p>SEND OK</p> <p>If connection has been established but sending buffer is full, response:</p> <p>SEND FAIL</p>

	<p>If connection has not been established, abnormally closed, or the parameter is incorrect, response: ERROR</p>
<p>Write Command If <service_type> is “TCP”, “UDP” or “TCP INCOMING”, send data with fixed length AT+QISEND=<connectID>,<send_length> After response “>”, input data until the data length is equal to <send_length></p>	<p>Response > <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 the parameter is incorrect, response: ERROR</p>
<p>Write Command If <service_type> is “UDP SERVICE” AT+QISEND=<connectID>,<send_length>,<remoteIP>,<remote_port> After response “>”, type data until the data length is equal to <send_length></p>	<p>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</p>
<p>Write Command When <send_length> is 0, query the sent data AT+QISEND=<connectID>,0</p>	<p>Response If the specified <connectID> connection has existed, response: +QISEND: <total_send_length>,<ackedbytes>,<unackedbytes> OK If there is any error, response:</p>

ERROR

Parameter

<connectID>	Integer type. Socket service index. The range is 0-11.
<send_length>	Integer type. The length of data to be sent, which cannot exceed 1460 bytes.
<remoteIP>	String type. The remote IP address (must be 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. Unit: byte.
<ackedbytes>	Integer type. The total length of acknowledged data. Unit: byte.
<unackedbytes>	Integer type. The total length of unacknowledged data. Unit: byte.
<err>	Integer type. The error code of the operation. Please refer to Chapter 4 .

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

In buffer access mode, after receiving data, the module will buffer it and report "+QIURC: "recv",<connectID>", then data can be retrieved by AT+QIRD.

Please note that 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 via AT+QIRD from buffer.

AT+QIRD Retrieve the Received TCP/IP Data

Test Command AT+QIRD=?	Response +QIRD: (0-11),(0-1500) OK
Write Command 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 there is no data, response: +QIRD: 0 OK If connection does not exist, response: ERROR
Write Command When <service_type> is	Response If data exists, response:

<p>“UDP SERVICE” AT+QIRD=<connectID></p>	<p>+QIRD: <read_actual_length>,<remoteIP>,<remote_port><CR><LF> ><data></p> <p>OK</p> <p>If there is no data, response: +QIRD: 0</p> <p>OK</p> <p>If connection does not exist, response: ERROR</p>
<p>Write Command When <read_length> is 0, query the retrieved data AT+QIRD=<connectID>,0</p>	<p>Response If the specified connection has existed, response: +QIRD: <total_receive_length>,<have_read_length>,<unread_length></p> <p>OK</p> <p>If there is any error, response: ERROR</p>

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. Unit: byte.
<read_actual_length>	The actual length of received data. Unit: byte.
<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. Unit: byte.
<have_read_length>	The length of read data. Unit: byte.
<unread_length>	The length of unread data. Unit: byte.

2.1.9. AT+QISENDEX Send Hex String

This command cannot be applied for “UDP SERVICE” and “TCP LISTENER” sockets.

AT+QISENDEX Send Hex String

Test Command AT+QISENDEX=?	Response +QISENDEX: (0-11),<hex_string> OK
Write Command AT+QISENDEX=<connectID>,<hex_string>	Response If sent successfully, response: SEND OK If there is no data, response: SEND FAIL If connection does not exist, response: 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 bytes.

2.1.10. AT+QISWTMD Switch Data Access Modes

The command can be used to switch the data access modes: buffer access mode, direct push mode and transparent access mode. When starting a socket service, data access modes can be specified via the <access_mode> parameter of AT+QIOPEN. After a socket has been opened, the data access modes can be changed via AT+QISWTMD.

AT+QISWTMD Switch Data Access Modes

Test Command AT+QISWTMD=?	Response +QISWTMD: (0-11),(0-2) OK
Write Command AT+QISWTMD=<connectID>,<access_mode>	Response If data access mode is switched successfully and <access_mode> is 0 or 1, response: OK

If data access mode is switched successfully and <access_mode> is 2, the module will enter into data mode, response:

CONNECT

If there is any error, response:

ERROR

Parameter

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

2.1.11. AT+QPING Ping a Remote Server

The command is used to test the Internet protocol reachability of a host. Before using ping tools, the 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 seconds.

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 a remote server successfully, response: OK [+QPING: <result>[,<IP_address>,<bytes>,<time>,<ttd>]<CR><LF>...] +QPING: <finresult>[,<sent>,<rcvd>,<lost>,<min>,<max>,<avg>] If there is any error, response: ERROR

Parameter

<contextID>	Integer type. The context ID. The range is 1-16.
<host>	The host address in string type. The 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. The range is 1-255, and the default value is 4. Unit: second.
<pingnum>	Integer type. Set the maximum number of time for sending ping request. The range is 1-10, and the default value is 4.
<result>	The result of each ping request. 0 Received the ping response from the server. In this case, it is followed by “,<IP_address>,<bytes>,<time>,<tTl>”. Others Please refer to Chapter 4 .
<IP_address>	The IP address of the remote server formatted as a dotted decimal IP.
<bytes>	The length of each sent ping request. Unit: byte.
<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 Please refer to Chapter 4 .
<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.

2.1.12. 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, the 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 synchronized successfully: OK +Qntp: <err>,<time> If there is any error, response: ERROR
Maximum Response Time	125s, determined by network.

Parameter

<contextID>	Integer type. The context ID. The range is 1-16.
<server>	String type. The address of NTP server.
<port>	Integer type. The port of NTP server. The range is 1-65535.
<autosettime>	Integer type. Whether to automatically set synchronized time to local time. 0 Not set <u>1</u> Set
<err>	Integer type. The error code of the operation. Please refer to Chapter 4 .
<time>	String type. The time synchronized from NTP server. The format is "YYYY/MM/DD,hh:mm:ss±zz". The range of "zz" is -48~56.

2.1.13. AT+QIDNSCFG Configure Address of DNS Server

Before setting the DNS address, the 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
Write Command Configure DNS server address for specified PDP context AT+QIDNSCFG=<contextID>,<pridnsaddr>[,<secdnsaddr>]	Response OK Or ERROR
Write Command Query DNS server address of specified PDP context AT+QIDNSCFG=<contextID>	Response +QIDNSCFG: <contextID>,<pridnsaddr>,<secdnsaddr> OK

Parameter

<contextID>	Integer type. The PDP context ID. The range is 1-16.
<pridnsaddr>	String type. The primary DNS server address.
<secdnsaddr>	String type. The secondary DNS server address.

2.1.14. AT+QIDNSGIP Get IP Address by Domain Name

Before querying the DNS, the 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 Or 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. The PDP context ID. The range is 1-16.
<hostname>	String type. The domain name.
<err>	Integer type. The error code of the operation. Please refer to Chapter 4 .
<IP_count>	Integer type. The number of the IP addresses corresponding to the <hostname>.
<DNS_ttl>	Integer type. The time to live of the DNS.
<hostIPAddr>	String type. The IP address of <hostname>.

2.1.15. AT+QICFG Configure Optional Parameters

The command is used to configure optional parameter.

AT+QICFG Configure Optional Parameters

<p>Test Command</p> <p>AT+QICFG=?</p>	<p>Response</p> <p>+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)</p> <p>OK</p>
<p>Write Command</p> <p>Set packet size for transparent mode</p> <p>AT+QICFG="transpktsize"[,<transpktsize>]</p>	<p>Response</p> <p>[+QICFG: "transpktsize",<transpktsize>]</p> <p>OK Or ERROR</p>
<p>Write Command</p> <p>Set waiting time for transparent mode</p> <p>AT+QICFG="transwaittm"[,<transwaittm >]</p>	<p>Response</p> <p>[+QICFG: "transwaittm",<transwaittm>]</p> <p>OK Or ERROR</p>
<p>Write Command</p> <p>Set the data format for sending and receiving (only for non-transparent mode)</p> <p>AT+QICFG="dataformat"[,<send_data_format>,<recv_data_format>]</p>	<p>Response</p> <p>[+QICFG: "dataformat",<send_data_format>,<recv_data_format>]</p> <p>OK Or ERROR</p>
<p>Write Command</p> <p>Set the display format of received data (only for non-transparent mode)</p> <p>AT+QICFG="viewmode"[,<view_mode>]</p>	<p>Response</p> <p>[+QICFG: "viewmode",<view_mode>]</p> <p>OK Or ERROR</p>
<p>Write Command</p> <p>Set the maximum interval time and number for TCP retransmission</p> <p>AT+QICFG="tcp/retranscfg"[,<m</p>	<p>Response</p> <p>[+QICFG: "tcp/retranscfg",<max_backoffs>,<max_rto>]</p> <p>OK</p>

ax_backoffs>,<max_rto>] Or
ERROR

Parameter

<transpktsize>	Integer type. The max length of the data packet to be sent. The range is 1-1460, and the default value is 1024. Unit: byte.
<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. The range is 0-20, and the default value is 2. Unit: 100ms.
<send_data_format>	Integer type. 0 Text mode 1 Hex mode
<recv_data_format>	Integer type. 0 Text mode 1 Hex mode
<view_mode>	Integer type. 0 Output format of received data: data header\r\n\data. 1 Output format of received data: data header,data.
<max_backoffs>	Integer type. The maximum number of TCP retransmission. The range is 3-20, and the default value is 8.
<max_rto>	Integer type. The maximum interval time of TCP retransmission. The range is 5-1000, and the default value is 600. Unit: 100ms.

NOTE

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

2.1.16. AT+QISDE Control Whether to Echo the Data for AT+QISEND

The command is used to control whether to echo the data for AT+QISEND.

AT+QISDE Control Whether to Echo the Data for AT+QISEND

Test command AT+QISDE=?	Response +QISDE: (0,1) OK
Read command AT+QISDE?	Response +QISDE: <echo>

	OK
Write Command AT+QISDE=<echo>	Response OK Or ERROR

Parameter

<echo>	Numeric type. Whether to echo the data for AT+QISEND
0	Not echo the data
1	Echo the data

2.1.17. AT+QIGETERROR Query the Last Error Code

If “ERROR” is returned after executing TCP/IP commands, the details of error can be queried via AT+QIGETERROR. Please note that AT+QIGETERROR just returns error code of the last TCP/IP 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. The error code of the operation. Please refer to Chapter 4 .
<errcode_description>	A string parameter indicates the details of error information. Please refer to Chapter 4 for details.

2.2. Description of URC

The URC of TCP/IP AT commands will be reported to the host in the format of beginning with “+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.

2.2.1. URC of Connection Closed

When TCP socket service is closed by remote peer or due to network error, the URC will be outputted. The status of the socket service will be "closing". AT+QICLOSE=<connectID> can be used to change the <socket_state> to "initial".

URC of Connection Closed

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

Parameter

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

2.2.2. URC of Incoming Data

In buffer access mode or direct push mode, after receiving data, the module will report URC to the 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 via AT+QIRD. Please note that 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 via 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 via AT+QIRD.
+QIURC: "recv",<connectID>,<currentrecvleng th><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>,<currentrecvleng th>,<remotelP>,<remote_port><CR><LF><data>	The URC of data incoming in direct push mode when <service_type> is "UDP SERVICE".

Parameter

<connectID>	Integer type. The socket service index. The range is 0-11.
<currentrecvleng th>	Integer type. The length of actual received data.
<remotelP>	Remote IP address.

<remote_port>	Remote port.
<data>	The received data.

2.2.3. 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.
-------------------------	------------------------------

2.2.4. URC of Incoming Connection

If the <service_type> is “TCP LISTENER”, when a remote client connects to this server, the host will automatically assign a free <connectID> for the new connection. The range of <connectID> is 0-11. The module will report a URC. The <service_type> of the new connection will be “TCP INCOMING”, and the <access_mode> will be buffer access mode.

URC of Incoming Connection

+QIURC: “incoming”,<connectID>,<serverID>, <remoteIP>,<remote_port>	When the new incoming connection is accepted by <serverID>, the allocated <connectID>, <remoteIP> and <remote_port> will be informed by this URC.
---	---

Parameter

<connectID>	Integer type. Assign this socket service for the incoming connection, which is automatically 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>.

2.2.5. URC of PDP Deactivation

PDP context may be deactivated by the network. The module will report URC to the host about PDP deactivation. The 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. The context ID. The range is 1-16.

3 Examples

3.1. Configure and Activate a Context

3.1.1. Configure a Context

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

3.1.2. Activate a Context

```
AT+QIACT=1 //Activate context 1. Depending on the network, the maximum  
OK //Activated successfully. response time is 150s.  
AT+QIACT? //Query the state of context.  
+QIACT: 1,1,1,"10.7.157.1"  
OK
```

3.1.3. Deactivate a Context

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

3.2. TCP Client Works in Buffer Access Mode

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

```

AT+QIOPEN=1,0,"TCP","220.180.239.212",8009,0,0 //Context is 1 and <connectID> is 0. Before
using AT+QIOPEN, the host should activate
the context with AT+QIACT first.

OK

+QIOPEN: 0,0 //Connected successfully. It is suggested to
wait for 150 seconds for the URC response
as "+QIOPEN: <connectID>,<err>". If the
URC response cannot be received in 150
seconds, the host could use AT+QICLOSE
to close the socket.

AT+QISTATE=1,0 //Query if the connection state of <connectID>
is 0.

+QISTATE: 0,"TCP","220.180.239.201",8705,65514,2,1,0,0,"usbmodem"

OK

```

3.2.2. 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. The 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 bytes.

>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 data.
SEND OK

AT+QISEND=0,0 //Query the length of sent data.
+QISEND: 14,14,0

OK

```

3.2.3. Receive Data from Remote Server in Buffer Access Mode

```
+QIURC: "recv",0 //The <connectID> 0 received data.
AT+QIRD=0,1500 //Read data, and the length is 1500 bytes.
+QIRD: 5 //The length of actual received data is 5 bytes.
test1

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

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

OK
```

3.2.4. Close a Connection

```
AT+QICLOSE=0 //Close a connection whose <connectID> is 0. Depending on the
network, the maximum response time is 10s.

OK
```

3.3. TCP Client Works in Transparent Access Mode

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

```
AT+QIOPEN=1,0,"TCP","220.180.239.212",8009,0,2 //Context is 1 and <connectID> is 0. Before using
AT+QIOPEN, the host should activate the
context with AT+QIACT first.

CONNECT //Connected successfully. It is suggested to wait
for 150 seconds for the URC response as
"CONNECT". If the URC response cannot be
received in 150 seconds, the host could use
AT+QICLOSE to close the socket.
```

3.3.2. Send Data in Transparent Access Mode

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

3.3.3. Receive Data from Remote Server in Transparent Access Mode

Test 1 //All data received from internet will be outputted via COM port directly.

3.3.4. Close a TCP Client

AT+QICLOSE=0 //After using “+++” to exit from the transparent access mode, the host could use AT+QICLOSE to close the TCP link. Depending on the network, the maximum response time is 10s.
OK

3.4. TCP Client Works in Direct Push Mode

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

AT+QIOPEN=1,0,"TCP","220.180.239.212",8009,0,1 //Context is 1 and <connectID> is 0. Before using AT+QIOPEN, the host should activate the context via AT+QIACT first.
OK
+QIOPEN: 0,0 //Connected successfully. It is suggested to wait for 150 seconds for the URC response as “+QIOPEN: <connectID>,<err>”. If the URC response cannot be received in 150 seconds, the 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

3.4.2. 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 bytes.
>test2
SEND OK
AT+QISEND=0,0 //Query the length of sent data. .
+QISEND: 10,10,0

OK
```

3.4.3. Receive Data from Remote Server in Direct Push Mode

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

3.4.4. Close a TCP Client

```
AT+QICLOSE=0 //Close the connection whose <connectID> is 0.
//Depending on the network, the maximum response time
//is 10s.

OK
```

3.5. TCP Server Works in Buffer Access Mode

3.5.1. 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 //TCP server is opened successfully.
AT+QISTATE=0,1 //Query whether the connection state of
//<contextID> is 1.
+QISTATE: 1,"TCP LISTENER","10.7.157.1",0,2020,3,1,1,0,"usbmodem"

OK
```

3.5.2. Accept TCP Incoming Connection

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

3.5.3. 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 bytes.  
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 data.  
+QIRD: 4,4,0  
  
OK
```

3.5.4. Close a 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
```

3.6. Example of UDP Service

3.6.1. Start a 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,  
the host should activate the context with  
AT+QIACT first.
```

OK

```
+QIOPEN: 2,0 //Start UDP service successfully.  
AT+QISTATE=0,1 //Query if the connection status of <contextID>  
is 1.  
+QISTATE: 2,"UDP SERVICE","10.7.157.1",0,3030,2,1,2,0,"usbmodem"
```

OK

3.6.2. 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
```

3.6.3. 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. The remote IP address is  
10.7.76.34 and remote port is 7687.  
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 whose IP is 10.7.76.34 and  
remote port is 7687.  
>1234567890  
SEND OK
```

3.6.4. Close a UDP Service

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

3.7. PING

AT+QPING=1,"www.baidu.com"

//Ping www.baidu.com in context 1. Before pinging the destination IP address, the host should activate the context by AT+QIACT first.

OK

+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

3.8. 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, the 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

3.9. Example of Getting Last Error Code

AT+QIOPEN=1,"TCP", "220.180.239.212",8009,0,1 //Start the socket service, and omit <connectID>.

ERROR

AT+QIGETERROR

+QIGETERROR: 552, invalid parameters

OK

4 Summary of Error Codes

If an “ERROR” is returned after executing TCP/IP AT commands, the details of error can be queried via AT+QIGETERROR. Please note that AT+QIGETERROR just returns error code of the last TCP/IP AT command.

Table 1: Summary of Error Codes

<err>	Meaning
0	Operation successful
550	Unknown error
551	Operation blocked
552	Invalid parameters
553	Memory not enough
554	Create socket failed
555	Operation not supported
556	Socket bind failed
557	Socket listen failed
558	Socket write failed
559	Socket read failed
560	Socket accept failed
561	Open PDP context failed
562	Close PDP context failed
563	Socket identity has been used
564	DNS busy

565	DNS parse failed
566	Socket connect failed
567	Socket has been closed
568	Operation busy
569	Operation timeout
570	PDP context broken down
571	Cancel send
572	Operation not allowed
573	APN not configured
574	Port busy

5 Appendix A Reference

Table 2: Terms and Abbreviations

Abbreviation	Description
APN	Access Point Name
CHAP	Challenge Handshake Authentication Protocol
DNS	Domain Name System
IP	Internet Protocol
NTP	Network Time Protocol
NV	Non-Volatile
PAP	Password Authentication Protocol I
PDP	Packet Data Protocol
PPP	Point-to-Point Protocol
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver & Transmitter
UDP	User Datagram Protocol
URC	Unsolicited Result Code
USB	Universal Serial Bus
UTC	Coordinated Universal Time