

# **BG95&BG77 TCP/IP**

## **Application Note**

**LPWA Module Series**

Rev. BG95&BG77\_TCP/IP\_Application\_Note\_V1.0

Date: 2019-08-12

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.**

Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai, China 200233

Tel: +86 21 5108 6236

Email: [info@quectel.com](mailto:info@quectel.com)

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

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

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

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

Or email to: [support@quectel.com](mailto: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. 2019. All rights reserved.***

# About the Document

## History

Revision	Date	Author	Description
1.0	2019-08-12	Terrence YANG	Initial

---

## Contents

<b>About the Document</b> .....	<b>2</b>
<b>Contents</b> .....	<b>3</b>
<b>Table Index</b> .....	<b>5</b>
<b>1 Introduction</b> .....	<b>6</b>
1.1. The Process of Using TCP/IP AT Commands .....	6
1.2. Description of Data Access Modes.....	8
<b>2 Description of TCP/IP AT Commands</b> .....	<b>10</b>
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.....	14
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 Data .....	20
2.1.10. AT+QISWTMD Switch Data Access Mode .....	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 .....	25
2.1.16. AT+QISDE Control Whether to Echo the Data for AT+QISEND.....	28
2.1.17. AT+QIGETERROR Query the Last Error Code.....	28
2.2. Description of URCs .....	29
2.2.1. URC Indicating Connection Closed .....	29
2.2.2. URC Indicating Incoming Data.....	29
2.2.3. URC Indicating Incoming Connection Full.....	30
2.2.4. URC Indicating Incoming Connection.....	30
2.2.5. URC Indicating PDP Deactivation.....	31
<b>3 Examples</b> .....	<b>32</b>
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 Buffer Access Mode.....	33
3.2.2. Send Data in Buffer Access Mode .....	33
3.2.3. Retrieve 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 Transparent Access Mode .....	34
3.3.2.	Send Data in Transparent Access Mode .....	35
3.3.3.	Retrieve 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 Direct Push Mode .....	35
3.4.2.	Send Data in Direct Push Mode.....	36
3.4.3.	Retrieve 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 .....	37
3.5.1.	Start a TCP Server .....	37
3.5.2.	Accept TCP Incoming Connection .....	37
3.5.3.	Retrieve Data from Incoming Connection.....	37
3.5.4.	Close a TCP Server .....	38
3.6.	UDP Service.....	38
3.6.1.	Start a UDP Service .....	38
3.6.2.	Send UDP Data to Remote .....	38
3.6.3.	Retrieve Data from Remote .....	38
3.6.4.	Close a UDP Service .....	39
3.7.	PING.....	39
3.8.	Synchronize Local Time.....	40
3.9.	Getting the Last Error Code .....	40
<b>4</b>	<b>Summary of Error Codes .....</b>	<b>41</b>
<b>5</b>	<b>Appendix A Reference.....</b>	<b>43</b>

## Table Index

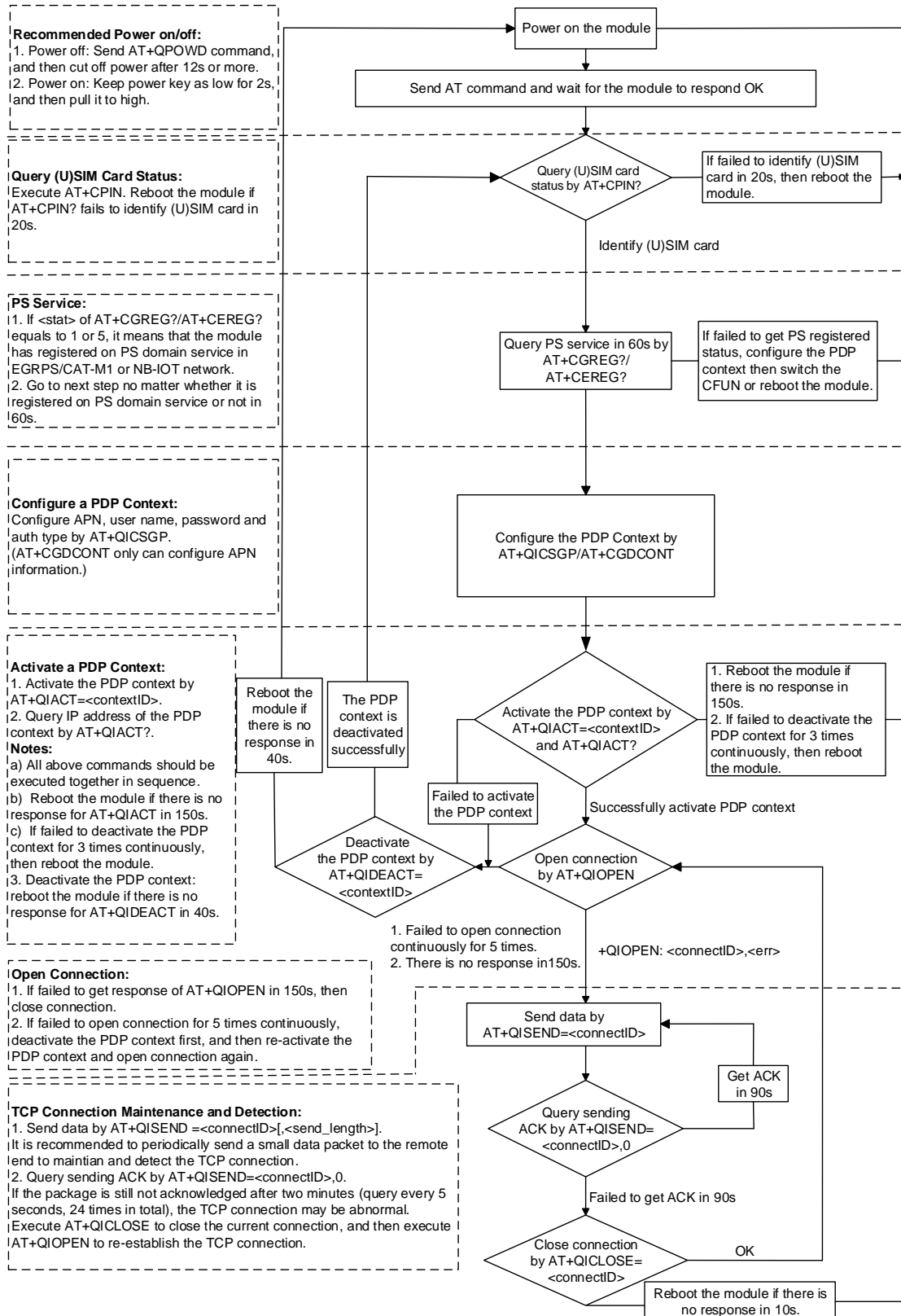
TABLE 1: SUMMARY OF ERROR CODES .....	41
TABLE 2: TERMS AND ABBREVIATIONS .....	43

# 1 Introduction

Quectel BG95 and BG77 modules feature embedded TCP/IP stack, which enables the host to access the Internet directly via AT commands. This greatly reduces the dependence on external PPP and TCP/IP protocol stacks and thus minimizes the cost. BG95 and BG77 modules provide 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

BG95 and BG77 modules support 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 the data from the Internet, it will buffer the data and report a URC "+QIURC: "recv",<connectID>". Customers can read the data via AT+QIRD command.
2. In direct push mode, data can be sent via AT+QISEND command, and if the module has received the 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 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 switch the data access mode back to transparent access mode.

- **Exit from transparent access mode**

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.

- **Return back to transparent access mode**

- 1) By AT+QISWTMD. Specify the <access\_mode> as 2 when executing this command. When transparent access mode has been entered successfully, "CONNECT" will be returned.
- 2) By ATO. After a connection exits from transparent access mode, executing ATO will switch the data access mode back to transparent access mode again. When transparent access mode has been entered successfully, "CONNECT" will be returned. If no connection has entered

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 the 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 the 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 <b>AT+QICSGP=?</b>	Response <b>+QICSGP: (1-16),(1-3),&lt;APN&gt;,&lt;username&gt;,&lt;password&gt;,(0-3)</b>  <b>OK</b>
Write Command Query the configuration of a specified context <b>AT+QICSGP=&lt;contextID&gt;</b>	Response <b>+QICSGP: &lt;context_type&gt;,&lt;APN&gt;,&lt;username&gt;,&lt;password&gt;,&lt;authentication&gt;</b>  <b>OK</b>
Write Command Configure the context <b>AT+QICSGP=&lt;contextID&gt;[,&lt;context_type&gt;,&lt;APN&gt;[,&lt;username&gt;,&lt;password&gt;[,&lt;authentication&gt;]]]</b>	Response <b>OK</b> Or <b>ERROR</b>

#### Parameter

<b>&lt;contextID&gt;</b>	Integer type. The context ID. The range is 1-16.
<b>&lt;context_type&gt;</b>	Integer type. The protocol type. <ol style="list-style-type: none"> <li>1 IPv4</li> <li>2 IPv6 (Not supported by this command currently)</li> <li>3 IPv4v6 (Not supported by this command currently)</li> </ol>
<b>&lt;APN&gt;</b>	String type. The access point name.
<b>&lt;username&gt;</b>	String type. The username.
<b>&lt;password&gt;</b>	String type. The password.

<b>&lt;authentication&gt;</b>	Integer type. The authentication methods.
<u>0</u>	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,"CMNBIOT","","",1 //Configure context 1. APN is "CMNBIOT" for China Mobile
NB-IoT network.

OK

```

### 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?.

Although the range of <contextID> is 1-16, the module supports maximum three PDP contexts activated simultaneously under LTE Cat M/EGPRS and maximum two under LTE Cat NB2. 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 <b>AT+QIACT=?</b>	Response <b>+QIACT: (1-16)</b>  <b>OK</b>
Read command <b>AT+QIACT?</b>	Response Return the list of the currently activated contexts and their IP addresses: <b>+QIACT: 1,&lt;context_state&gt;,&lt;context_type&gt;[,&lt;IP_addresses&gt;]</b> [..... <b>+QIACT: 16,&lt;context_state&gt;,&lt;context_type&gt;[,&lt;IP_addresses&gt;]]</b>  <b>OK</b>

Write Command Activate a specified PDP context <b>AT+QIACT=&lt;contextID&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	150 seconds, determined by the network.

### Parameter

<b>&lt;contextID&gt;</b>	Integer type. The context ID. The range is 1-16.
<b>&lt;context_state&gt;</b>	Integer type. The context state. 0 Deactivated 1 Activated
<b>&lt;context_type&gt;</b>	Integer type. The protocol type. <u>1</u> IPv4 2 IPv6
<b>&lt;IP_address&gt;</b>	The local IP address after the context is activated.

### 2.1.3. AT+QIDEACT Deactivate a PDP Context

The command is used to deactivate a 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 <b>AT+QIDEACT=?</b>	Response <b>+QIDEACT: (1-16)</b>  <b>OK</b>
Write Command <b>AT+QIDEACT=&lt;contextID&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	40 seconds, determined by the network.

### Parameter

<b>&lt;contextID&gt;</b>	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>. The data access mode (buffer access mode, direct push mode and transparent access mode) can be specified by <access\_mode>. The "+QIOPEN" URC indicates whether the socket service has been opened successfully.

1. 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 "+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 that of "TCP LISTENER".
2. If <service\_type> is "UDP SERVICE", UDP data can be sent to or received from the remote IP via <local\_port>.
  - Send data: execute AT+QISEND=<connectID>,<send\_length>,<remoteIP>,<remote\_port>.
  - Receive data in direct push mode: the module reports the URC "+QIURC: "recv",<connectID>,<currentrecvlength>,<remoteIP>,<remote\_port><CR><LF><data>".
  - Receive data in buffer access mode: the module reports the URC "+QIURC: "recv",<connectID>", and then customers can retrieve the data via AT+QIRD=<connectID>.
3. It is suggested to wait for 150 seconds for "+QIOPEN: <connectID>,<err>" to be outputted after executing the Write Command. If the URC cannot be received in 150 seconds, AT+QICLOSE should be used to close the socket.

#### AT+QIOPEN Open a Socket Service

<p>Test Command AT+QIOPEN=?</p>	<p>Response <b>+QIOPEN: (1-16),(0-11),"TCP/UDP/TCP LISTENER/UDP SERVICE", "&lt;IP_address&gt;/&lt;domain_name&gt;",&lt;remote_port&gt;,&lt;local_port&gt;,(0-2)</b></p> <p><b>OK</b></p>
<p>Write Command AT+QIOPEN=&lt;contextID&gt;,&lt;connectID&gt;,&lt;service_type&gt;,&lt;IP_address&gt;/&lt;domain_name&gt;,&lt;remote_port&gt;[,&lt;local_port&gt;[,&lt;access_mode&gt;]]</p>	<p>Response If in transparent access mode (&lt;access_mode&gt;=2) and the service is opened successfully: <b>CONNECT</b></p> <p>If there is any error: <b>ERROR</b> Error description can be got via AT+QIGETERROR.</p> <p>If in buffer access mode (&lt;access_mode&gt;=0) or direct push mode (&lt;access_mode&gt;=1): <b>OK</b></p>

	<p><b>+QIOPEN: &lt;connectID&gt;,&lt;err&gt;</b>          &lt;err&gt; is 0 when the service is opened successfully. In other cases, &lt;err&gt; is not 0.</p>
Maximum Response Time	150 seconds, determined by the network.

## Parameter

<b>&lt;contextID&gt;</b>	Integer type. The context ID. The range is 1-16.
<b>&lt;connectID&gt;</b>	Integer type. The socket service index. The range is 0-11.
<b>&lt;service_type&gt;</b>	String type. The socket service type. "TCP" Start a TCP connection as a client "UDP" Start a UDP connection as a client "TCP LISTENER" Start a TCP server to listen to TCP connection "UDP SERVICE" Start a UDP service
<b>&lt;IP_address&gt;</b>	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".
<b>&lt;domain_name&gt;</b>	String type. The domain name address of the remote server.
<b>&lt;remote_port&gt;</b>	The port of the remote server. The range is 0-65535. It must be specified when <service_type> is "TCP" or "UDP". It is to be assigned as 0 when <service_type> is not "TCP" or "UDP".
<b>&lt;local_port&gt;</b>	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": the local port will be assigned automatically if <local_port> is 0. Otherwise the local port is assigned as specified.
<b>&lt;access_mode&gt;</b>	Integer type. The data access mode of the socket service. 0 Buffer access mode 1 Direct push mode 2 Transparent access mode
<b>&lt;err&gt;</b>	Integer type. The error code of the operation. Please refer to <b>Chapter 4</b> .

### 2.1.5. AT+QICLOSE Close a Socket Service

The command is used to close a 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 <b>AT+QICLOSE=?</b>	Response <b>+QICLOSE: (0-11),(1-65535)</b>  <b>OK</b>
Write Command <b>AT+QICLOSE=&lt;connectID&gt;[,&lt;timeout&gt;]</b>	Response If the socket service is closed successfully: <b>OK</b>  If it is failed to close the socket service: <b>ERROR</b>

### Parameter

<b>&lt;connectID&gt;</b>	Integer type. The socket service index. The range is 0-11.
<b>&lt;timeout&gt;</b>	Integer type. The timeout value for the response to be outputted. If the FIN ACK of the other peer is not received within <timeout>, the module will be forced 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 a specified socket service.

## AT+QISTATE Query Socket Service Status

Test Command <b>AT+QISTATE=?</b>	Response <b>OK</b>
Read/Execution Command <b>AT+QISTATE? or AT+QISTATE</b>	Response Return the status of all existing connections: List of ( <b>+QISTATE: &lt;connectID&gt;,&lt;service_type&gt;,&lt;IP_address&gt;,&lt;remote_port&gt;,&lt;local_port&gt;,&lt;socket_state&gt;,&lt;contextID&gt;,&lt;serverID&gt;,&lt;access_mode&gt;,&lt;AT_port&gt;</b> )  <b>OK</b>
Write Command If <query_type> is 0, query the connection status of a specified context <b>AT+QISTATE=&lt;query_type&gt;,&lt;context ID&gt;</b>	Response Return the status of all existing connections in a specified context: List of ( <b>+QISTATE: &lt;connectID&gt;,&lt;service_type&gt;,&lt;IP_address&gt;,&lt;remote_port&gt;,&lt;local_port&gt;,&lt;socket_state&gt;,&lt;context</b>



	tID>,<serverID>,<access_mode>,<AT_port>)
	OK
Write Command If <query_type> is 1, query the connection status of a specified socket service AT+QISTATE=<query_type>,<connect ID>	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 of all socket services in a specified context 1 Query connection status of a specified socket service
<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. "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 "TCP INCOMING" Start a TCP connection accepted by a TCP server "UDP SERVICE" Start a UDP service
<IP_address>	IP address. If <service_type>="TCP" or "UDP", it is the IP address of remote server. If <service_type>="TCP LISTENER" or "UDP SERVICE", it is the local IP address. If <service_type>="TCP INCOMING", it is the IP address of remote client.
<remote_port>	Remote port number. If <service_type>="TCP" or "UDP", it is the port of remote server. If <service_type>="TCP LISTENER" or "UDP SERVICE", the port is invalid. If <service_type>="TCP INCOMING", it is the port of remote client.
<local_port>	Local port number. If <local_port> is 0, then the local port is assigned automatically.
<socket_state>	Integer type. The socket service status. 0 "Initial": connection has not been established 1 "Opening": client is connecting or server is trying to listen 2 "Connected": client/incoming connection has been established 3 "Listening": server is listening 4 "Closing": connection is closing
<serverID>	It is valid only 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
<b>&lt;AT_port&gt;</b>	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 the data access mode of a specified socket service is buffer access mode (<access\_mode>=0) or direct push mode (<access\_mode>=1), then the data can be sent via AT+QISEND. When 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, and customers should delay for some time to re-send the data. 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

Test Command <b>AT+QISEND=?</b>	Response <b>+QISEND: (0-11),(0-1460)</b>  <b>OK</b>
Write Command Send variable-length data when <service_type> is "TCP", "UDP" or "TCP INCOMING" <b>AT+QISEND=&lt;connectID&gt;</b>	Response > After the response ">", input the data to be sent. Tap CTRL+Z to send, and tap Esc to cancel the operation  If the connection has been established and the data is sent successfully: <b>SEND OK</b>  If the connection has been established but the sending buffer is full: <b>SEND FAIL</b>  If the connection has not been established, abnormally closed, or any parameter is incorrect, response: <b>ERROR</b>
Write Command Send fixed-length data when <service_type> is "TCP", "UDP" or "TCP	Response > After the response ">", input the data until the data length

<p>INCOMING" <b>AT+QISEND=&lt;connectID&gt;,&lt;send_length&gt;</b></p>	<p>equals &lt;send_length&gt;</p> <p>If the connection has been established and the data is sent successfully: <b>SEND OK</b></p> <p>If the connection has been established but the sending buffer is full: <b>SEND FAIL</b></p> <p>If the connection has not been established, abnormally closed, or any parameter is incorrect: <b>ERROR</b></p>
<p>Write Command If &lt;service_type&gt; is "UDP SERVICE" <b>AT+QISEND=&lt;connectID&gt;,&lt;send_length&gt;,&lt;remoteIP&gt;,&lt;remote_port&gt;</b></p>	<p>Response This command is used to send fixed length data to specified remote IP address and remote port. The &lt;service_type&gt; must be "UDP SERVICE". &gt; After response "&gt;", type the data until the data length equals &lt;send_length&gt;</p> <p>If the connection has been established and the data is sent successfully: <b>SEND OK</b></p> <p>If the connection has been established but the sending buffer is full: <b>SEND FAIL</b></p> <p>If the connection has not been established, abnormally closed, or any parameter is incorrect: <b>ERROR</b></p>
<p>Write Command When &lt;send_length&gt; is 0, query the sent data <b>AT+QISEND=&lt;connectID&gt;,0</b></p>	<p>Response If the specified &lt;connectID&gt; connection has existed: <b>+QISEND: &lt;total_send_length&gt;,&lt;ackedbytes&gt;,&lt;unackedbytes&gt;</b></p> <p><b>OK</b></p> <p>If there is any error: <b>ERROR</b></p>

## Parameter

<b>&lt;connectID&gt;</b>	Integer type. Socket service index. The range is 0-11.
<b>&lt;send_length&gt;</b>	Integer type. The length of data to be sent, which cannot exceed 1460 bytes.
<b>&lt;remoteIP&gt;</b>	String type. The remote IP address (must be dot format). It is valid only when <service_type> is "UDP SERVICE".
<b>&lt;remote_port&gt;</b>	Integer type. The remote port is only valid when <service_type> is "UDP SERVICE".
<b>&lt;total_send_length&gt;</b>	Integer type. The total length of sent data. Unit: byte.
<b>&lt;ackedbytes&gt;</b>	Integer type. The total length of acknowledged data. Unit: byte.
<b>&lt;unackedbytes&gt;</b>	Integer type. The total length of unacknowledged data. Unit: byte.

### 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 the 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	Response
<b>AT+QIRD=?</b>	<b>+QIRD: (0-11),(0-1500)</b>  <b>OK</b>
Write Command When <service_type> is "TCP", "UDP" or "TCP INCOMING" <b>AT+QIRD=&lt;connectID&gt;[,&lt;read_length&gt;]</b>	Response If the specified connection has received the data: <b>+QIRD: &lt;read_actual_length&gt;&lt;CR&gt;&lt;LF&gt;&lt;data&gt;</b>  <b>OK</b>  If there is no data: <b>+QIRD: 0</b>  <b>OK</b>  If the connection does not exist: <b>ERROR</b>
Write Command When <service_type> is "UDP SERVICE" <b>AT+QIRD=&lt;connectID&gt;</b>	Response If data exists: <b>+QIRD: &lt;read_actual_length&gt;,&lt;remoteIP&gt;,&lt;remote_port&gt;&lt;CR&gt;&lt;LF&gt;&lt;data&gt;</b>

	<p>OK</p> <p>If there is no data: <b>+QIRD: 0</b></p> <p>OK</p> <p>If the connection does not exist: <b>ERROR</b></p>
<p>Write Command</p> <p>When &lt;read_length&gt; is 0, query the retrieved data length</p> <p><b>AT+QIRD=&lt;connectID&gt;,0</b></p>	<p>Response</p> <p>If the specified connection exists: <b>+QIRD: &lt;total_receive_length&gt;,&lt;have_read_length&gt;,&lt;unread_length&gt;</b></p> <p>OK</p> <p>If there is any error: <b>ERROR</b></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 length of data that has been actually retrieved. Unit: byte.
<remoteIP>	String type. The remote IP address. It is valid only when <service_type> is "UDP SERVICE".
<remote_port>	Integer type. The remote port. It is valid only 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 data that has been retrieved. Unit: byte.
<unread_length>	The length of data that has not been retrieved. Unit: byte.

### 2.1.9. AT+QISENDEX Send Hex String Data

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

#### AT+QISENDEX Send Hex String Data

<p>Test Command</p> <p><b>AT+QISENDEX=?</b></p>	<p>Response</p> <p><b>+QISENDEX: (0-11),&lt;hex_string&gt;</b></p> <p>OK</p>
---	--

Write Command <b>AT+QISENDEX=&lt;connectID&gt;,&lt;hex_string&gt;</b>	Response If the hex string is sent successfully: <b>SEND OK</b>  If the sending buffer is full: <b>SEND FAIL</b>  If the connection does not exist: <b>ERROR</b>
--	--

### Parameter

<b>&lt;connectID&gt;</b>	Integer type. The socket service index. The range is 0-11.
<b>&lt;hex_string&gt;</b>	String type. Hex string data. The max length is 512 bytes.

### 2.1.10. AT+QISWTMD Switch Data Access Mode

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

#### AT+QISWTMD Switch Data Access Mode

Test Command <b>AT+QISWTMD=?</b>	Response <b>+QISWTMD: (0-11),(0-2)</b>  <b>OK</b>
Write Command <b>AT+QISWTMD=&lt;connectID&gt;,&lt;access_mode&gt;</b>	Response If data access mode is switched successfully and <access_mode> is 0 or 1: <b>OK</b>  If data access mode is switched successfully and <access_mode> is 2, the module will enter data mode, response: <b>CONNECT</b>  If there is any error: <b>ERROR</b>

## Parameter

<b>&lt;connectID&gt;</b>	Integer type. The socket service index. The range is 0-11.
<b>&lt;access_mode&gt;</b>	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 within <timeout> and the default value of <timeout> is 4 seconds.

#### AT+QPING Ping a Remote Server

Test Command <b>AT+QPING=?</b>	Response <b>+QPING: (1-16),&lt;host&gt;,(1-255),(1-10)</b>  <b>OK</b>
Write Command <b>AT+QPING=&lt;contextID&gt;,&lt;host&gt;[,&lt;timeout&gt;[,&lt;pingnum&gt;]]</b>	Response If a remote server is pinged successfully: <b>OK</b>  <b>[+QPING: &lt;result&gt;[,&lt;IP_address&gt;,&lt;bytes&gt;,&lt;time&gt;,&lt;tTL&gt;]&lt;CR&gt;&lt;LF&gt;...]</b> <b>+QPING: &lt;finresult&gt;[,&lt;sent&gt;,&lt;rcvd&gt;,&lt;lost&gt;,&lt;min&gt;,&lt;max&gt;,&lt;avg&gt;]</b>  If there is any error: <b>ERROR</b>

## Parameter

<b>&lt;contextID&gt;</b>	Integer type. The context ID. The range is 1-16.
<b>&lt;host&gt;</b>	The host address in string type. The format is a domain name or a dotted decimal IP address.
<b>&lt;timeout&gt;</b>	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.
<b>&lt;pingnum&gt;</b>	Integer type. Set the maximum number of times for sending ping request. The range is 1-10, and the default value is 4.
<b>&lt;result&gt;</b>	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 Error codes. Please refer to <b>Chapter 4</b> for details.
<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 Error codes. Please refer to <b>Chapter 4</b> for details.
<sent>	Total number of the ping requests that have been sent.
<rcvd>	Total number of the ping requests that received the response.
<lost>	Total number of the ping requests that are time out.
<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+QNTIP Synchronize Local Time with NTP Server

The command is used to synchronize the local time with the Universal Time Coordinated (UTC) through the NTP server. Before time synchronization, 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+QNTIP Synchronize Local Time with NTP Server	
Test command <b>AT+QNTIP=?</b>	Response <b>+QNTIP: (1-16),&lt;server&gt;,(list of supported &lt;port&gt;s),(0,1)</b>  <b>OK</b>
Read command <b>AT+QNTIP?</b>	Response If in the process of local time synchronization: <b>+QNTIP: &lt;server&gt;,&lt;port&gt;</b>  <b>OK</b>
Write command <b>AT+QNTIP=&lt;contextID&gt;,&lt;server&gt; [,&lt;port&gt;][,&lt;autosettime&gt;]</b>	Response If the local time is synchronized with NTP server successfully: <b>OK</b>  <b>+QNTIP: &lt;err&gt;,&lt;time&gt;</b>  If there is any error: <b>ERROR</b>
Maximum Response Time	125s, determined by the network.



## Parameter

<b>&lt;contextID&gt;</b>	Integer type. The context ID. The range is 1-16.
<b>&lt;server&gt;</b>	String type. The address of NTP server.
<b>&lt;port&gt;</b>	Integer type. The port of NTP server. The range is 1-65535.
<b>&lt;autosettime&gt;</b>	Integer type. Whether to automatically set synchronized time as local time. 0 Not set 1 Set
<b>&lt;err&gt;</b>	Integer type. Error code of operation. Please refer to <b>Chapter 4</b> for more details.
<b>&lt;time&gt;</b>	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 <b>AT+QIDNSCFG=?</b>	Response <b>+QIDNSCFG: (1-16),&lt;pridnsaddr&gt;,&lt;secdnsaddr&gt;</b>  <b>OK</b>
Write Command <b>AT+QIDNSCFG=&lt;contextID&gt;[,&lt;pridnsaddr&gt;[,&lt;secdnsaddr&gt;]]</b>	Response If <pridnsaddr> and <secdnsaddr> are omitted, query the current DNS server addresses of a specified PDP context: <b>+QIDNSCFG: &lt;contextID&gt;,&lt;pridnsaddr&gt;,&lt;secdnsaddr&gt;</b>  <b>OK</b>  If <pridnsaddr> and <secdnsaddr> are present, configure the primary and secondary DNS server addresses of a specified PDP context: <b>OK</b> Or <b>ERROR</b>
Maximum Response Time	

## Parameter

<b>&lt;contextID&gt;</b>	Integer type. The PDP context ID. The range is 1-16.
<b>&lt;pridnsaddr&gt;</b>	String type. The primary DNS server address.
<b>&lt;secdnsaddr&gt;</b>	String type. The secondary DNS server address.

**NOTE**

Configuration of DNS server address is not supported currently.

### 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 <b>AT+QIDNSGIP=?</b>	Response <b>+QIDNSGIP: (1-16),&lt;hostname&gt;</b>  <b>OK</b>
Write Command <b>AT+QIDNSGIP=&lt;contextID&gt;,&lt;hostname&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>  The result will be returned as URC. <b>+QIURC: "dnsgip",&lt;err&gt;,&lt;IP_count&gt;,&lt;DNS_ttl&gt;</b> [..... <b>+QIURC: "dnsgip",&lt;hostIPaddr&gt;</b>
Maximum Response Time	60s, determined by the network.

#### Parameter

<contextID>	Integer type. The PDP context ID. The range is 1-16.
<hostname>	String type. The domain name.
<err>	Integer type. Error code of operation. Please refer to <b>Chapter 4</b> for more details.
<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 <b>AT+QICFG=?</b></p>	<p>Response  <b>+QICFG: "transpktsize",(1-1460)</b>  <b>+QICFG: "transwaittm",(0-20)</b>  <b>+QICFG: "dataformat",(0,1),(0,1)</b>  <b>+QICFG: "viewmode",(0,1)</b>  <b>+QICFG: "recvind",(0,1)</b>  <b>+QICFG: "tcp/retranscfg",(3-20),(5-1000)</b>   <b>OK</b></p>
<p>Write Command Configure the packet size for transparent access mode <b>AT+QICFG="transpktsize"[,&lt;transpktsize&gt;]</b></p>	<p>Response  <b>[+QICFG: "transpktsize",&lt;transpktsize&gt;]</b>   <b>OK</b>  Or  <b>ERROR</b></p>
<p>Write Command Configure the waiting time for transparent access mode <b>AT+QICFG="transwaittm"[,&lt;transwaittm &gt;]</b></p>	<p>Response  <b>[+QICFG: "transwaittm",&lt;transwaittm&gt;]</b>   <b>OK</b>  Or  <b>ERROR</b></p>
<p>Write Command Configure the format of the data to be sent or received (only for buffer access mode and direct push mode ) <b>AT+QICFG="dataformat"[,&lt;send_data_format&gt;,&lt;recv_data_format&gt;]</b></p>	<p>Response  <b>[+QICFG: "dataformat",&lt;send_data_format&gt;,&lt;recv_data_format&gt;]</b>   <b>OK</b>  Or  <b>ERROR</b></p>
<p>Write Command Configure the output format of received data (only for non-transparent mode) <b>AT+QICFG="viewmode"[,&lt;view_mode&gt;]</b></p>	<p>Response  <b>[+QICFG: "viewmode",&lt;view_mode&gt;]</b>   <b>OK</b>  Or  <b>ERROR</b></p>
<p>Write Command Configure the URC format of TCP/IP buffer access mode. If this command is set, when the module receives data from a remote server,</p>	<p>Response  If <b>&lt;enable&gt;</b> is omitted, return the current configuration:  <b>[+QICFG: "recvind",&lt;enable&gt;]</b>   <b>OK</b></p>

the data length will be added to the URC. <b>AT+QICFG="recvind",[&lt;enable&gt;]</b>	If <b>&lt;enable&gt;</b> is present, configure the URC format: <b>OK</b> Or <b>ERROR</b>
Write Command Configure the maximum interval time and number of TCP retransmission <b>AT+QICFG="tcp/retranscfg",[&lt;max_backoffs&gt;,&lt;max_rto&gt;]</b>	Response <b>[+QICFG: "tcp/retranscfg",&lt;max_backoffs&gt;,&lt;max_rto&gt;]</b>  <b>OK</b> Or <b>ERROR</b>

## Parameter

<b>&lt;transpktsize&gt;</b>	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.
<b>&lt;transwaittm&gt;</b>	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.
<b>&lt;send_data_format&gt;</b>	Integer type. The format of the data to be sent. 0 Text mode 1 Hex mode
<b>&lt;recv_data_format&gt;</b>	Integer type. The format of the data to be received. 0 Text mode 1 Hex mode
<b>&lt;view_mode&gt;</b>	Integer type. Output format of received data. 0 Output format of received data: data header\r\n\data. 1 Output format of received data: data header,data.
<b>&lt;recvind&gt;</b>	Integer type. URC format. 0 URC format without data length. URC format is "+QIURC: "recv",<connectID>". 1 URC format with data length. URC format is "+QIURC: "recv",<connectID>,<data_len>".
<b>&lt;max_backoffs&gt;</b>	Integer type. The maximum number of TCP retransmission. The range is 3-20, and the default value is 8.
<b>&lt;enable&gt;</b>	Integer type. Configure the URC format of TCP/IP buffer access mode. 0 The data length will not be added to the URC 1 The data length will be added to the URC
<b>&lt;max_rto&gt;</b>	Integer type. The maximum interval time of TCP retransmission. The range is 5-1000, and the default value is 600. Unit: 100ms.

**NOTE**

The setting of AT+QICFG="tcp/retranscfg",<max\_backoffs>,<max\_rto>] will take effect to all TCP sockets and cannot be saved to NV.

### 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 <b>AT+QISDE=?</b>	Response <b>+QISDE: (0,1)</b>  <b>OK</b>
Read command <b>AT+QISDE?</b>	Response <b>+QISDE: &lt;echo&gt;</b>  <b>OK</b>
Write Command <b>AT+QISDE=&lt;echo&gt;</b>	Response <b>OK</b> Or <b>ERROR</b>

#### Parameter

<b>&lt;echo&gt;</b>	Numeric type. Whether to echo the data for AT+QISEND
0	Not echo the data
<u>1</u>	Echo the data

### 2.1.17. AT+QIGETERROR Query the Last Error Code

If <err> 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 <b>AT+QIGETERROR=?</b>	Response <b>OK</b>
Execution Command <b>AT+QIGETERROR</b>	Response <b>+QIGETERROR: &lt;err&gt;,&lt;errcode_description&gt;</b>

OK

## Parameter

<b>&lt;err&gt;</b>	Integer type. Error code of operation. Please refer to <b>Chapter 4</b> for details.
<b>&lt;errcode_description&gt;</b>	A string parameter indicates the details of error information. Please refer to <b>Chapter 4</b> for details.

## 2.2. Description of URCs

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 not presented intentionally.

### 2.2.1. URC Indicating Connection Closed

When TCP socket service is closed by remote peer or due to network error, the URC will be outputted, and the status of socket service will be "closing" (<socket\_state>=4). AT+QICLOSE=<connectID> can be used to change the <socket\_state> to "initial"

#### URC Indicating Connection Closed

<b>+QIURC: "closed",&lt;connectID&gt;</b>	Socket service connection is closed.
---	--------------------------------------

## Parameter

<b>&lt;connectID&gt;</b>	Integer type. The socket service index. The range is 0-11.
--------------------------	--

### 2.2.2. URC Indicating Incoming Data

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

In buffer access mode, after receiving data, the module will report "+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 Indicating Incoming Data

<b>+QIURC: "recv",&lt;connectID&gt;</b>	Indicating incoming data in buffer access mode. The host can retrieve data via AT+QIRD.
<b>+QIURC: "recv",&lt;connectID&gt;,&lt;currentrecvlength&gt;&lt;CR&gt;&lt;LF&gt;&lt;data&gt;</b>	Indicating incoming data in direct push mode when the <service_type> is "TCP", "UDP" or "TCP INCOMING".
<b>+QIURC: "recv",&lt;connectID&gt;,&lt;currentrecvlength&gt;,&lt;remoteIP&gt;,&lt;remote_port&gt;&lt;CR&gt;&lt;LF&gt;&lt;data&gt;</b>	Indicating data incoming in direct push mode when <service_type> is "UDP SERVICE".

#### Parameter

<b>&lt;connectID&gt;</b>	Integer type. The socket service index. The range is 0-11.
<b>&lt;currentrecvlength&gt;</b>	Integer type. The length of actually received data.
<b>&lt;remoteIP&gt;</b>	Remote IP address.
<b>&lt;remote_port&gt;</b>	Remote port number.
<b>&lt;data&gt;</b>	The received data.

### 2.2.3. URC Indicating 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 "+QIURC: "incoming full"" for the new incoming connection request.

#### URC Indicating Incoming Connection Full

<b>+QIURC: "incoming full"</b>	Incoming connection is full.
--------------------------------	------------------------------

### 2.2.4. URC Indicating 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, and the range of <connectID> is 0-11. In such a case, the module will report the URC "+QIURC: "incoming"". The <service\_type> of the new connection will be "TCP INCOMING", and the <access\_mode> will be buffer access mode.

#### URC Indicating Incoming Connection

<b>+QIURC: "incoming",&lt;connectID&gt;,&lt;serverID&gt;,&lt;remoteIP&gt;,&lt;remote_port&gt;</b>	When the new incoming connection is accepted by <serverID>, the allocated <connectID>, <remoteIP> and <remote_port> will be informed by this URC.
---	---

---

## Parameter

---

<b>&lt;connectID&gt;</b>	Integer type. Assign this socket service for the incoming connection, which is automatically specified by the module. The range is 0-11.
<b>&lt;serverID&gt;</b>	The incoming <connectID> accepted by the server whose <service_type> is "TCP LISTENER" and listening socket ID is <serverID>.
<b>&lt;remoteIP&gt;</b>	Remote IP address of the incoming <connectID>.
<b>&lt;remote_port&gt;</b>	Remote port of the incoming <connectID>.

---

### 2.2.5. URC Indicating PDP Deactivation

PDP context may be deactivated by the network. The module will report this URC to the host about PDP deactivation. In such a case, the host must execute AT+QIDEACT to deactivate the context and reset all connections.

#### URC Indicating PDP Deactivation

<b>+QIURC: "pdpdeact",&lt;contextID&gt;</b>	<contextID> context is deactivated.
---	-------------------------------------

---

## Parameter

---

<b>&lt;contextID&gt;</b>	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,"CMNBIOT","","",1 //Configure context 1. APN is "CMNBIOT" for China Mobile
OK //NB-IoT network.
```

### 3.1.2. Activate a Context

```
AT+QIACT=1 //Activate context 1. Depending on the network, the
OK //maximum response time is 150s.
//Activated the context successfully.

AT+QIACT? //Query the context state, protocol type and IP address of
+QIACT: 1,1,1,"10.7.157.1" context 1.
OK
```

### 3.1.3. Deactivate a Context

```
AT+QIDEACT=1 //Deactivate context 1.
OK //Deactivated the context 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 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 //TCP client connected successfully. It is suggested to wait for 150
seconds for the URC "+QIOPEN: <connectID>,<err>". If the URC cannot
be received in 150 seconds, the host could use AT+QICLOSE to close
the socket.

AT+QISTATE=1,0 //Query connection status of socket service 1.
+QISTATE: 0,"TCP","220.180.239.212",8009,65514,2,1,0,0,"usbmodem"

OK

```

### 3.2.2. Send Data in Buffer Access Mode

```

AT+QISEND=0 //Send variable-length data.
>test1<ctrl+Z>
SEND OK // "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.

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, acknowledged data and
unacknowledged data.
+QISEND: 9,9,0 //A total of 9 bytes data has been sent, and the 9-byte data has
been acknowledged by the server.

OK

AT+QISENDEX=0,"3132333435" //Send Hex string data.
SEND OK

AT+QISEND=0,0 //Query the length of sent data, acknowledged data and

```

unacknowledged data.

+QISEND: 14,14,0

OK

### 3.2.3. Retrieve Data from Remote Server in Buffer Access Mode

+QIURC: "recv",0 //The <connectID> 0 has received data.

**AT+QIRD=0,1500** //Retrieve the data, and the maximum length of data to be retrieved is 1500 bytes.

+QIRD: 5 //The length of actually retrieved 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 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

//TCP client connected successfully. It is suggested to wait for 150 seconds for the URC "CONNECT". If the URC 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. Retrieve 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 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

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

AT+QISTATE=1,0

//Query the connection state of socket service 0.

```
+QISTATE: 0,"TCP","220.180.239.212",8009,65344,2,1,0,1,"usbmodem"
```

```
OK
```

### 3.4.2. Send Data in Direct Push Mode

```
AT+QISEND=0 //Send variable-length data.
>test1<ctrl+Z>
SEND OK // "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.

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, acknowledged data and
unacknowledged data.
+QISEND: 10,10,0 //A total of 10 bytes data has been sent, and all the 10-byte data has
been acknowledged.

OK
```

### 3.4.3. Retrieve Data from Remote Server in Direct Push Mode

```
+QIURC: "recv",0,4 //Retrieve 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, the host should activate
                                                    //the context with AT+QIACT first.

OK

+QIOPEN: 1,0 //TCP server is opened successfully.

AT+QISTATE=0,1 //Query the connection status of context 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 //A new TCP connection is accepted. The
                                                    //<service_type> is "TCP incoming", and
                                                    //<connectID> is 11.
```

### 3.5.3. Retrieve Data from Incoming Connection

```
+QIURC: "recv",11 //Received data from remote incoming connection.

AT+QIRD=11,1500 //Retrieve the data received from incoming connection.
+QIRD: 4 //Actually retrieved data length is 4 bytes.
test

OK

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

OK

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

OK

### 3.5.4. Close a TCP Server

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

OK

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

OK

## 3.6. 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 //UDP service is opened successfully.
```

```
AT+QISTATE=0,1 //Query the connection status of context 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 the  
remote port is 6969.
```

```
>1234567890
```

```
SEND OK
```

### 3.6.3. Retrieve Data from Remote

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

```
AT+QIRD=2 //Retrieve 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 //The retrieved data length is 4 bytes. The remote IP address is
10.7.76.34 and the remote port is 7687.
AAAA
OK

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

AT+QISEND=2,10,"10.7.76.34",7687 //Send data to the remote whose IP is 10.7.76.34 and the
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. Getting the Last Error Code

```
AT+QIOPEN=1,"TCP","220.180.239.212",8009,0,1
```

//Start a socket service but and <connectID> is not present.

```
ERROR
```

```
AT+QIGETERROR
```

```
+QIGETERROR: 552, invalid parameters
```

```
OK
```

# 4 Summary of Error Codes

If <err> 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
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
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	Universal Time Coordinated