RGx00U&RM500U Series
TCP/IP Application Note

5G Module Series

Version: 1.0

Date: 2023-07-20

Status: Released

At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local offices. 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 us at: support@quectel.com.

Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an "as available" basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

Use and Disclosure Restrictions

License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties ("third-party materials"). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel's or third-party's servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

Disclaimer

a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

About the Document

Revision History

Version	Date	Author	Description
-	2023-03-24	Wayen XU	Creation of the document
1.0	2023-07-20	Wayen XU	First official release

Contents

Table Index

1 Introduction

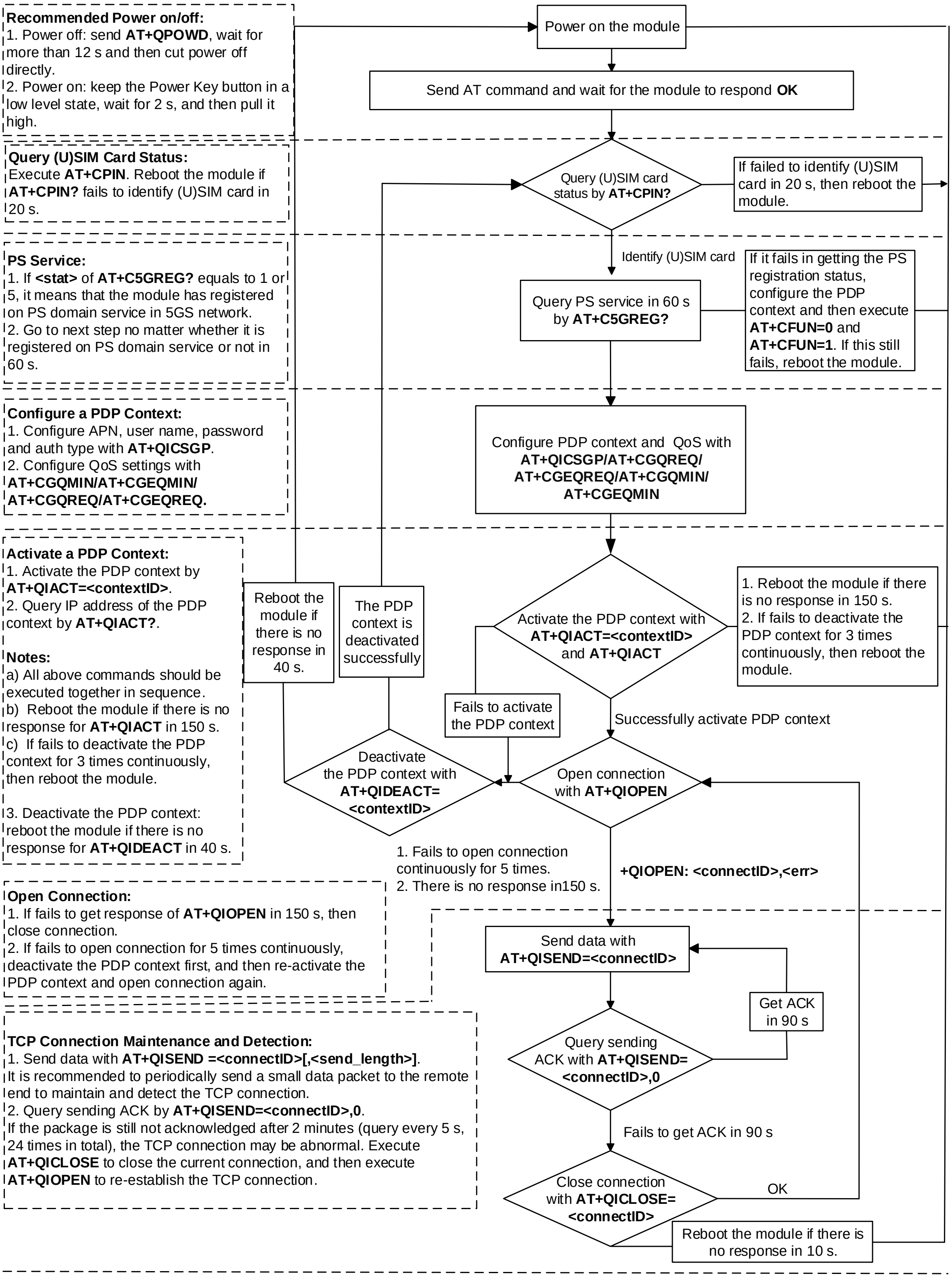
Quectel RG200U series, RG500U series and RM500U series modules feature an embedded TCP/IP stack, which enables the host to access the Internet directly through AT commands. This greatly reduces the dependence on external PPP and TCP/IP protocol stacks and thus minimizes the cost.

RG200U series, RG500U series and RM500U series modules provide the following socket services: TCP client, UDP client, TCP server and UDP server.

This document introduces how to use the TCP/IP function of Quectel RG200U series, RG500U series and RM500U series modules through AT commands.

1.1. The Process of Using TCP/IP AT Commands

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


Recommended Power on/off:
1. Power off: send AT+QPOWD, wait for more than 12 s and then cut power off directly.
2. Power on: keep the Power Key button in a low level state, wait for 2 s, and then pull it high.
Power on the module
Send AT command and wait for the module to respond OK
Query (U)SIM Card Status:
Execute AT+CPIN. Reboot the module if AT+CPIN? fails to identify (U)SIM card in 20 s.
Query (U)SIM card status by AT+CPIN?
If failed to identify (U)SIM card in 20 s, then reboot the module.
Identify (U)SIM card
PS Service:
1. If <stat> of AT+C5GREG? equals to 1 or 5, it means that the module has registered on PS domain service in 5GS network.
2. Go to next step no matter whether it is registered on PS domain service or not in 60 s.
Query PS service in 60 s by AT+C5GREG?
If it fails in getting the PS registration status, configure the PDP context and then execute AT+CFUN=0 and AT+CFUN=1. If this still fails, reboot the module.
Configure a PDP Context:
1. Configure APN, user name, password and auth type with AT+QICSGP.
2. Configure QoS settings with AT+CGQMIN/AT+CGEQMIN/ AT+CGQREQ/AT+CGEQREQ.
Configure PDP context and QoS with AT+QICSGP/AT+CGQREQ/ AT+CGEQREQ/AT+CGQMIN/ AT+CGEQMIN
Activate a PDP Context:
1. Activate the PDP context by AT+QIACT=<contextID>.
2. Query IP address of the PDP context by AT+QIACT?.
Notes:
a) All above commands should be executed together in sequence.
b) Reboot the module if there is no response for AT+QIACT in 150 s.
c) If fails to deactivate the PDP context for 3 times continuously, then reboot the module.
3. Deactivate the PDP context: reboot the module if there is no response for AT+QIDEACT in 40 s.
Reboot the module if there is no response in 40 s.
The PDP context is deactivated successfully
Activate the PDP context with AT+QIACT=<contextID> and AT+QIACT
1. Reboot the module if there is no response in 150 s.
2. If fails to deactivate the PDP context for 3 times continuously, then reboot the module.
Fails to activate the PDP context
Successfully activate PDP context
Deactivate the PDP context with AT+QIDEACT= <contextID>
Open connection with AT+QIOPEN
1. Fails to open connection continuously for 5 times.
2. There is no response in150 s.
+QIOPEN: <connectID>,<err>
Open Connection:
1. If fails to get response of AT+QIOPEN in 150 s, then close connection.
2. If fails to open connection for 5 times continuously, deactivate the PDP context first, and then re-activate the PDP context and open connection again.
Send data with AT+QISEND=<connectID>
Get ACK in 90 s
TCP Connection Maintenance and Detection:
1. Send data with AT+QISEND =<connectID>[,<send_length>].
It is recommended to periodically send a small data packet to the remote end to maintain and detect the TCP connection.
2. Query sending ACK by AT+QISEND=<connectID>,0.
If the package is still not acknowledged after 2 minutes (query every 5 s, 24 times in total), the TCP connection may be abnormal. Execute AT+QICLOSE to close the current connection, and then execute AT+QIOPEN to re-establish the TCP connection.
Query sending ACK with AT+QISEND= <connectID>,0
Fails to get ACK in 90 s
Close connection with AT+QICLOSE= <connectID>
OK
Reboot the module if there is no response in 10 s.

Notes:

1. Please note that users 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 60 s.
2. Reboot the module if the module has not got response of **AT+QIACT** in 150 s or response of **AT+QICLOSE** in 10s or response of **AT+QIDEACT** in 40 s.
3. It is NOT recommended to frequently reboot the module. When the module failed to reboot for 3 times continuously for AT command execution, it can be rebooted immediately for the first time. If it still fails, reboot the module after 10 minutes for the second time, and reboot after 30 minutes for the third time if it still fails for the second time. By analogy, gradually increase the reboot interval.

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

1.2. Description of Data Access Modes

RG200U series, RG500U series and RM500U series modules support the following three data access modes:

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

When opening a socket service by **AT+QIOPEN**, the data access mode can be specified by **<access_mode>**. After a socket service is successfully opened, the data access mode can be switched by **AT+QISWTMD**.

1. In buffer access mode: data can be sent by **AT+QISEND**. If the module has received the data from the Internet, it will buffer the data and report the URC **+QIURC: "recv",<connectID>**. Data can be read by **AT+QIRD**.

2. In direct push mode: data can be sent by **AT+QISEND**. If the module has received the data from the Internet, the data will be outputted to COM port directly in the format of **+QIURC: " recv",<connectID>,<currect_recvlength><CR><LF><data>** or **+QIURC: "recv",<connectID>,< current_recvlength>,<remoteIP>,<remote_port><CR><LF><data>**.

3. In transparent transmission mode: the corresponding port (such as UART, USB modem port, etc.) enters the exclusive mode. The data received from COM port will be sent to the Internet directly. The data received from the Internet will be outputted via COM port directly.

   - **Exit from transparent transmission mode**

     To exit from the transparent transmission mode, **+++** or DTR (**AT&D1** should be set first) can be used. To prevent the **+++** from being misinterpreted as data, the following operations should be followed:

     1) Do not input any character within 1 s or longer before inputting **+++**.
     2) Input only **+++** within 1 s with no other character during this period.
     3) Do not input any character within 1 s after **+++** has been inputted.
     4) Use **+++** or DTR (**AT&D1** should be set first) to make the module exit from transparent transmission mode, and wait until **OK** is returned.

   - **Return to transparent transmission mode**

     1) By executing **AT+QISWTMD**: specify **<access_mode>**=2 and it will return **CONNECT** when the connection enters transparent transmission mode successfully.
     2) By executing **ATO**. After exiting from transparent transmission mode, execute **ATO** will switch back to transparent transmission mode again. It will return **CONNECT** when the

connection enters transparent transmission mode successfully. If no connection has entered transparent transmission mode before, it will return **NO CARRIER** after executing **ATO**.

NOTE

1. In buffer access mode, if the buffer is not empty, the module will not report any new URC until all the received data has been read from the buffer by **AT+QIRD**.
2. In transparent transmission 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 transmission mode. In this case, **AT+QICLOSE** should be executed to close the socket service.

2 Description of TCP/IP AT Commands

This chapter introduces AT command related to TCP/IP.

2.1. AT Command Introduction

2.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on the command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on the command line. When an optional parameter is not given in a command, the new value equals to its previous value or the default settings, unless otherwise specified.
- **Underline** Default setting of a parameter.

2.1.2. AT Command Syntax

All command lines must start with **AT** or **at** and end with **<CR>**. Information responses and result codes always start and end with a carriage return character and a line feed character: **<CR><LF><response><CR><LF>**. In tables presenting commands and responses throughout this document, only the commands and responses are presented, and **<CR>** and **<LF>** are deliberately omitted.

Table 1: Types of AT Commands

Command Type	Syntax	Description
Test Command	**AT+<cmd>=?**	Test the existence of corresponding command and return information about the type, value, or range of its parameter.
Read Command	**AT+<cmd>?**	Check the current parameter value of corresponding command.
Write Command	**AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]**	Set user-definable parameter value.

Execution Command	**AT+<cmd>**	Return a specific information parameter or perform a specific action.

2.2. Declaration of AT Command Examples

The AT command examples in this document are provided to help you learn about the use of the AT commands introduced herein. The examples, however, should not be taken as Quectel's recommendations or suggestions about how to design a program flow or what status to set the module into. Sometimes multiple examples may be provided for one AT command. However, this does not mean that there is a correlation among these examples, or that they should be executed in a given sequence.

2.3. Description of AT Commands

2.3.1. AT+QICFG Configure Optional Parameters

This command configures optional parameters.

AT+QICFG Configure Optional Parameters	
Test Command **AT+QICFG=?**	Response: **+QICFG: "transpktsize",(**range of supported **<transpktsize>**s**)** **+QICFG: "transwaittm",(**range of supported **<transwaittm>** s**)** **+QICFG: "dataformat",(**list of supported **<send_data_format>**s**),(**list of supported **<recv_data_format>**s**)** **+QICFG: "viewmode",(**list of supported **<view_mode>**s**)** **+QICFG: "tcp/retranscfg",(**range of supported **<max_backoffs>** s**),(**range of supported **<max_rto>**s**)** **+QICFG: "dns/cache",(**list of supported **<DNS_cache>**s**)** **+QICFG: "qisend/timeout",(**list of supported **<time>**s**)** **+QICFG: "passiveclosed",(**list of supported **<closed>**s**)** **+QICFG: "tcp/accept",(**list of supported **<state>**s**)** **+QICFG: "tcp/keepalive",(**list of supported **<enable>**s**),(**range of supported **<idle_time>**s**),(**range of supported **<interval_time>**s**),(**range of supported **<probe_cnt>**s**)** **+QICFG: "recvind",(**list of supported **<enable>**s**)** **+QICFG: "close/mode",(**range of supported **<close_mode>**s**)** **+QICFG: "sendinfo",(**list of supported **<send_view_mode>**s**)** **+QICFG: "formatcfg",(**range of supported **<format>**s**)** **OK**

Write Command Set the maximum length of data to be sent **AT+QICFG="transpktsize"[,<transpktsize>]**	Response: If the optional parameter is omitted, query the current setting: **+QICFG: "transpktsize",<transpktsize>** **OK** If the optional parameter is specified, set the maximum length of data to be sent: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure the waiting time before sending data automatically in transparent transmission mode **AT+QICFG="transwaittm"[,<transwaittm>]**	Response: If the optional parameter is omitted, query the current setting: **+QICFG: "transwaittm",<transwaittm>** **OK** If the optional parameter is specified, configure the waiting time before sending data automatically in transparent transmission mode: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure the format of the data to be sent or received (only for non-transparent transmission mode) **AT+QICFG="dataformat"[,<send_data_format>,<recv_data_format>]**	Response: If the optional parameters are omitted, query the current setting: **+QICFG: "dataformat",<send_data_format>,<recv_data_format>** **OK** If the optional parameters are specified, configure the format of the data to be sent or received: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**

Write Command Configure the output format of received data (only for non-transparent transmission mode) **AT+QICFG="viewmode"[,<view_mode>]**	Response: If the optional parameter is omitted, query the current setting: **+QICFG: "viewmode",<view_mode>** **OK** If the optional parameter is specified, configure the output format of the received data: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure the maximum interval time and maximum times for TCP retransmission **AT+QICFG="tcp/retranscfg"[,<max_backoffs>,<max_rto>]**	Response: If the optional parameters are omitted, query the current setting: **+QICFG: "tcp/retranscfg",<max_backoffs>,<max_rto>** **OK** If the optional parameters are specified, configure the maximum interval time and maximum times for TCP retransmission: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Enable or disable the DNS cache **AT+QICFG="dns/cache"[,<DNS_cache>]**	Response: If the optional parameter is omitted, query the current setting: **+QICFG: "dns/cache",<DNS_cache>** **OK** If the optional parameter is specified, enable or disable the DNS cache: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure the timeout value for	Response: If the optional parameter is omitted, query the current setting:

input data **AT+QICFG="qisend/timeout"[,<time>]**	**+QICFG: "qisend/timeout",<time>** **OK** If the parameter is specified, configure the input data timeout: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Enable or disable the passive close of TCP connection when the server is closed **AT+QICFG="passiveclosed"[,<closed>]**	Response If the optional parameter is omitted, query the current setting: **+QICFG: "passiveclosed",<closed>** **OK** If the parameter is specified, enable or disable the passive close of TCP connection when the server is closed: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Enable or disable the automatic reception of the TCP connection from the client **AT+QICFG="tcp/accept"[,<state>]**	Response: If the optional parameter is omitted, query the current setting: **+QICFG: "tcp/accept",<state>** **OK** If the optional parameter is specified, enable or disable the automatic reception of the TCP connection from the client: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure whether to send TCP keep-alive information. **AT+QICFG="tcp/keepalive"[,<enable>[,<idle_time>,<interval**	Response If the optional parameters are omitted, query the current setting: **+QICFG: "tcp/keepalive",<enable>,<idle_time>,<interval_time>,<probe_cnt>**

_time>,<probe_cnt>]]	**OK** If the optional parameters are specified, configure whether to send TCP keep-alive information: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure whether to display the data length in the URC reported by the module after receiving the data in buffer mode **AT+QICFG="recvind"[,<enable>]**	Response If the parameter is omitted, query the current setting: **+QICFG: "recvind",<enable>** **OK** If the parameter is specified, configure whether to display the data length in the URC reported by the module after receiving the data: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Enable or disable to return response information of **AT+QICLOSE** in URC format **AT+QICFG="close/mode"[,<close_mode>]**	Response If the parameter is omitted, query the current setting: **+QICFG: "close/mode",<close_mode>** **OK** If the parameter is specified, enable or disable to return response information of **AT+QICLOSE** in URC format: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Enable or disable to return response information of **AT+QISEND** in URC format (only when **<service_type>**="TCP"、"UDP"	Response If the parameter is omitted, query the current setting: **+QICFG: "sendinfo",<send_view_mode>** **OK**

or "TCP INCOMING") **AT+QICFG="sendinfo"[,<send_view_mode>]**	If the parameter is specified, enable or disable to return response information of **AT+QISEND** in URC format: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Configure response information format after executing **AT+QISEND** **AT+QICFG="formatcfg"[,<format>]**	Response If the parameter is omitted, query the current setting: **+QICFG: "formatcfg",<format>** **OK** If the parameter is specified, configure response information format after executing **AT+QISEND**: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	-
Characteristic	The commands take effect immediately. The configurations are not saved.

Parameter

<transpktsize>	Integer type. The maximum length of the data packet to be sent in transparent transmission mode. Range: 1–1460; Default value: 1024; Unit: byte.
<transwaittm>	Integer type. The waiting time before sending the data automatically if the length of data to be sent is less than the value of **<transpktsize>** in transparent transmission mode. Range: 0–20; Default value: 2; Unit: 100 ms.
<send_data_format>	Integer type. The format of the data to be sent. The suffix “0x” is not needed when the mode is set as hex mode, and the module will automatically form two bytes to one ASCII code. 0 Text mode 1 Hex mode
<recv_data_format>	Integer type. The format of the data to be received. The suffix “0x” is not needed when the mode is set as hex mode, and the module will automatically form two bytes to one ASCII code. 0 Text mode

	1 Hex mode
<view_mode>	Integer type. The output format of the received data in non-transparent transmission mode. 0 data header\r\ndata. 1 data header,data.
<max_backoffs>	Integer type. The maximum times of TCP retransmission. Range: 3–20; Default value: 8.
<max_rto>	Integer type. The maximum interval time of TCP retransmission. Range: 5–1000; Default value: 600; Unit: 100 ms.
<DNS_cache>	Integer type. Enable or disable the DNS cache. 0 Enable 1 Disable
<time>	Integer type. The timeout value of **AT+QISEND**. After **>** is responded, if no data is inputted within the timeout period, **AT+QISEND** will be exited. Range: 0–3600; Default value: 0; Unit: ms.
<closed>	Integer type. Enable or disable the passive close of TCP connection when the server is closed. 0 Disable 1 Enable
<state>	Integer type. Enable or disable the automatic reception of the TCP connection from the client. 0 Disable 1 Enable
<enable>	Integer type. In **AT+QICFG="tcp/keepalive"**, it means whether to send TCP keepalive information. In **AT+QICFG="recvind"**, it configures whether to display the data length in the URC reported by the module after receiving the data in buffer mode. 0 Not send/Not display 1 Send/Display
<idle_time>	Integer type. The triggered keepalive cycle time. Range: 1–1800; Default value: 60; Unit: s.
<interval_time>	Integer type. The interval of sending packets in a cycle time. Range: 25–100; Default value: 25; Unit: s.
<probe_cnt>	Integer type. The times of sending packets in a cycle time. Range: 3–10; Default value: 3.
<close_mode>	Integer type. Enable or disable to report response information of **AT+QICLOSE** in URC format. 0 Disable 1 Enable
<send_view_mode>	Integer type. Enable or disable to report response information of **AT+QISEND** in URC format. 0 Disable 1 Enable
<format>	Integer type. The response information format after executing **AT+QISEND**.

	S3 is **ATS3** and S4 is **ATS4**. 0 "S3S4>" 1 "S3S4>S3S4"
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

NOTE

The setting of **AT+QICFG="tcp/retranscfg"[,<max_backoffs>,<max_rto>]** will take effect to all TCP Socket connections and will not be saved to NV.

2.3.2. AT+QICSGP Configure Parameter of TCP/IP Context

This command configures **<APN>**, **<username>**, **<password>** and other parameters of TCP/IP context. The QoS settings can be configured by **AT+CGQMIN**, **AT+CGEQMIN**, **AT+CGQREQ** and **AT+CGQREQ**. For more details about the AT commands, see ***document [1]***.

AT+QICSGP Configure Parameters of TCP/IP Context	
Test Command **AT+QICSGP=?**	Response: **+QICSGP: (**range of supported **<conntextID>**s**),(**range of supported **<context_type>**s**),<APN>,<username>,<password>,(**range of supported **<authentication>**s**)** **OK**
Write Command Configure the specified context **AT+QICSGP=<contextID>[,<context_type>,<APN>[,<username>,<password>,<authentication>]]**	Response: If the optional parameters are omitted, query the configuration of specified context: **+QICSGP: <context_type>,<APN>,<username>,<password>,<authentication>** **OK** If the optional parameters are specified, configure the specified context: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	-
Characteristics	The command takes effect immediately.

	The configurations are not saved.

Parameter

<contextID>	Integer type. The PDP context ID. Range: 1–5.
<context_type>	Integer type. The protocol type. 1 IPv4 2 IPv6 3 IPv4v6
<APN>	String type. Access point name.
<username>	String type. Username. Maximum length: 127; Unit: byte.
<password>	String type. Password. Maximum length: 127; Unit: byte.
<authentication>	Integer type. APN authentication methods. 0 None 1 PAP 2 CHAP 3 PAP or CHAP
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

Example

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

OK
AT+QICSGP=1,1,"UNINET","","",0            //Configure context 1. APN is configured as "UNINET"
                                           (China Unicom).
OK
```

2.3.3. AT+QIACT Activate PDP Context

Before activating PDP context by **AT+QIACT**, the context should be configured by **AT+QICSGP**. After activation, the IP address can be queried by **AT+QIACT?**.

The range of **<contextID>** is 1–5. Module supports up to 5 PDP contexts activated simultaneously. Depending on the network, it may take at most 150 s to return **OK** or **ERROR** after **AT+QIACT** is executed. Before the response is returned, other AT commands cannot be executed.

AT+QIACT Activate PDP Context	
Test Command **AT+QIACT=?**	Response: **+QIACT: (**range of supported **<contextID>**s**)**

	OK
Read command **AT+QIACT?**	Response: Return the list of all activated PDP contexts and corresponding IP addresses: If **<context_type>**=1 **+QIACT: 1,<context_state>,1,<IPv4_address>** **[...** **+QIACT: 5,<context_state>,1,<IPv4_address>]** **OK** Or **<context_type>**=2 **+QIACT: 1,<context_state>,2,<IPv6_address>** **[...** **+QIACT: 5,<context_state>,2,<IPv6_address>]** **OK** Or **<context_type>**=3 **+QIACT: 1,<context_state>,3,<IPv4_address>,<IPv6_address>** **[...** **+QIACT: 5,<context_state>,3,<IPv4_address>,<IPv6_address>]** **OK**
Write Command Activate specified PDP context **AT+QIACT=<contextID>**	Response: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	150 seconds, determined by the network.
Characteristics	-

Parameter

<contextID>	Integer type. The PDP context ID. Range: 1–5.
<context_state>	Integer type. The PDP context state. 0 Deactivated 1 Activated
<context_type>	Integer type. The protocol type.

	1 IPv4 2 IPv6 3 IPv4v6
<IPv4_address>	String type. Local IPv4 address after the context is activated.
<IPv6_address>	String type. Local IPv6 address after the context is activated.
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

2.3.4. AT+QIDEACT Deactivate PDP Context

This command deactivates specified context and closes all TCP/IP connections set up in this context. Depending on the network, it may take at most 40 s to return **OK** or **ERROR** after executing **AT+QIDEACT**. Before the response is returned, other AT commands cannot be executed.

AT+QIDEACT Deactivate PDP Context	
Test Command **AT+QIDEACT=?**	Response: **+QIDEACT: (**range of supported **<contextID>**s**)** **OK**
Write Command **AT+QIDEACT=<contextID>**	Response: **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	40 seconds, determined by network.
Characteristics	-

Parameter

<contextID>	Integer type. The PDP context ID. Range: 1–5.
<err>	Integer type. Error codes. See ***Chapter 4***.

2.3.5. AT+QIACTEX Activate PDP Context Asynchronously

This command activates specified PDP context asynchronously. Before activating specified PDP context by **AT+QIACTEX**, the context should be configured by **AT+QICSGP**. After activation, the IP address can be queried by **AT+QIACTEX?**.

The range of **<contextID>** is 1–5. Module supports up to 5 PDP contexts activated simultaneously. The number of PDP context activated simultaneously actually depends on (U)SIM card. The module returns **OK** or **ERROR** after executing **AT+QIACTEX**. After that, the result is reported by URC **+QIACTEX**.

AT+QIACTEX Activate PDP Context Asynchronously	
Test Command **AT+QIACTEX=?**	Response: **+QIACTEX: (**range of supported **<contextID>**s)**, (**list of supported **<view_mode>**s**)** **OK**
Read Command **AT+QIACTEX?**	Response: Return the list of all activated PDP contexts and corresponding IP addresses: If **<context_type>**=1 **+QIACTEX: 1,<context_state>,1,<IPv4_address>** **[...** **+QIACTEX: 5,<context_state>,1,<IPv4_address>]** Or **<context_type>**=2 **+QIACTEX: 1,<context_state>,2,<IPv6_address>** **[...** **+QIACTEX: 5,<context_state>,2,<IPv6_address>]** Or **<context_type>**=3 **+QIACTEX: 1,<context_state>,3,<IPv4_address>,<IPv6_address>** **[...** **+QIACTEX: 5,<context_state>,3,<IPv4_address>,<IPv6_address>]** **OK**
Write Command **AT+QIACTEX=<contextID>[,<view_mode>]**	Response: If **<view_mode>**=0： **OK** **+QIACTEX: <contextID>,<err>** If **<view_mode>**=1： **OK** **+QIACTEX: <contextID>,<err>[,1,<IPv4_address>]** Or **+QIACTEX: <contextID>,<err>[,2,<IPv6_address>]** Or

	+QIACTEX: <contextID>,<err>[,3,<IPv4_address>,<IPv6_address>] If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	150 seconds, determined by network.
Characteristics	-

Parameter

<contextID>	Integer type. The PDP context ID. Range: 1–5.
<view_mode>	Integer type. Whether to display IP address and type. 0 Not display 1 Display
<context_state>	Integer type. The PDP context state. 0 Deactivated 1 Activated
<context_type>	Integer type. Protocol type. 1 IPv4 2 IPv6 3 IPv4v6
<IPv4_address>	String type. Local IPv4 address of the activated context.
<IPv6_address>	String type. Local IPv6 address of the activated context.
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

2.3.6. AT+QIDEACTEX Deactivate PDP Context Asynchronously

This command deactivates PDP context asynchronously. After executing **AT+QIDEACTEX**, the module returns **OK** or **ERROR**. After that, the result of PDP context is reported by URC **+QIDEACTEX**.

AT+QIDEACTEX Deactivate PDP Context Asynchronously	
Test Command **AT+QIDEACTEX=?**	Response: **+QIDEACTEX: (**range of supported **<contextID>**s**)** **OK**
Write Command **AT+QIDEACTEX=<contextID>**	Response: **OK**

	+QIDEACTEX: <contextID>,<err> If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	40 seconds, determined by network.
Characteristics	-

Parameter

<contextID>	Integer type. The PDP context ID. Range: 1–5.
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

2.3.7. AT+QIOPEN Open Socket Service

This command opens socket service. The service type can be specified by **<service_type>**. The data access mode (buffer access mode, direct push mode and transparent transmission mode) can be specified by **<access_mode>**. The URC **+QIOPEN** indicates whether the socket service has been opened successfully.

1. If **<service_type>**="TCP LISTENER", the module works as TCP server. After accepting a new TCP connection, the module automatically specifies a **<connectID>** and reports a URC **+QIURC: "incoming",<connectID>,<serverID>,<remoteIP>,<remote_port>**. The range of **<connectID>** is 0–11. The type of this new TCP connection is "TCP INCOMING" and the value of **<access_mode>** is the same with that of **<access_mode>** of "TCP LISTENER".

2. If **<service_type>**="UDP SERVICE", UDP data can be sent to or received from the remote IP through **<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 data can be read by user by **AT+QIRD=<connectID>**.

It is suggested to wait for 150 seconds for the URC **+QIOPEN: <connectID>,<err>** to be outputted after the command is executed. If the URC cannot be received in 150 seconds, **AT+QICLOSE** should be used to close the socket.

AT+QIOPEN Open Socket Service	
Test Command **AT+QIOPEN=?**	Response **+QIOPEN: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),(**list of supported **<service_type>**s**),<address_string>,(**range of supported **<remote_port>**s**),(**range of supported **<local_port>**s**),(**range of supported **<access_mode>**s**),(**range of supported **<TCP_client_maxnum>**s**),(**range of supported **<connect_timeout>**s) **OK**
Write Command **AT+QIOPEN=<contextID>,<connectID>,<service_type>,<address_string>,<remote_port>[,<local_port>[,<access_mode>[,<TCP_client_maxnum>[,<connect_timeout>]]]]**	Response: If the service is in transparent transmission mode (**<access_mode>**=2) and is opened successfully: **CONNECT** If there is any error: **ERROR** Error description can be queried by **AT+QIGETERROR**. If the service is in buffer access mode (**<access_mode>**=0) or direct push mode (**<access_mode>**=1): **OK** **+QIOPEN: <connectID>,<err>** **<err>** is 0 when the service is opened successfully. In other cases, **<err>** is not 0.
Maximum Response Time	150 seconds, depending on the network.
Characteristics	-

Parameter

<contextID>	Integer type. The PDP context ID. Range: 1–5.	
<connectID>	Integer type. Socket ID. Range: 0–11.	
<service_type>	String type. Socket service type.	
	"TCP"	Start a TCP connection as a client
	"UDP"	Start a UDP connection as a client
	"TCP LISTENER"	Start a TCP server to listen to TCP connection
	"UDP SERVICE"	Start a UDP service
<address_string>	String type. The IP address or domain name of the remote server. It can be	

	set as **<IP_address>** or **<domain_name>**. **<IP_address>** String type. If **<service_type>**="TCP" or "UDP", it indicates the IP address of the remote server, such as "220.180.239.212". If **<service_type>**="TCP LISTENER" or "UDP SERVICE", input "127.0.0.1". **<domain_name>** String type. The domain name address of the remote server. Maximum length: 256; Unit: byte.
<remote_port>	Integer type. The port of the remote server. Range: 0–65535. If **<service_type>**="TCP" or "UDP", the port must be specified. If **<service_type>**="TCP LISTENER" or "UDP SERVICE", set it as 0.
<local_port>	Integer type. Local port. Range: 0–65535. If **<service_type>**="TCP" or "UDP", the local port will be assigned automatically if it is set as 0. Otherwise, the local port will be specified. If **<service_type>**="TCP LISTENER" or "UDP SERVICE", the port number must be specified.
<access_mode>	Integer type. The data access mode of the socket service. 0 Buffer access mode 1 Direct push mode 2 Transparent transmission mode
<TCP_client_maxnum>	Integer type. **<TCP_client_maxnum>** is valid only when **<service_type>**="TCP LISTENER" and it can control the maximum number of client connection. Range: 0–11; Default value: 11.
<connect_timeout>	Integer type. Reconnection timeout value. Range: 10–150; Default value: 0; Unit: s. The reconnection is not enabled by default.
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

NOTE

1. It is suggested to wait for 150 seconds for the URC **+QIOPEN: <connectID>,<err>** to be outputted.
2. You need to close socket service by executing **AT+QICLOSE=<connectID>** if the connection failed.
3. If the module works as the server, you need to configure the call mode as the router mode or the bridge mode by executing **AT+QCFG="nat",1** or **AT+QCFG="nat",2**. For more details about the command, see ***document [1]***.

2.3.8. AT+QICLOSE Close Socket Service

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

AT+QICLOSE Close Socket Service	
Test Command **AT+QICLOSE=?**	Response: **+QICLOSE: (**range of supported **<connectID>**s**),(**range of supported **<timeout>**s**)** **OK**
Write Command **AT+QICLOSE=<connectID>[,<timeout>]**	Response: If **<close_mode>**=0 (**AT+QICFG="close/mode"**): Close successfully: **OK** Fail to close: **ERROR** If **<close_mode>**=1 (**AT+QICFG="close/mode"**): Close successfully: **OK** **+QIURC: "closed",<connectID>** Fail to close: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	Default value: 10 s, determined by **<timeout>**.
Characteristics	-

Parameter

<contextID>	Integer type. The Socket ID. Range: 0–11.
<timeout>	Integer type. Timeout value for the response to be outputted. If the FIN ACK of other clients is not received within **<timeout>**, the module will be forced to close the socket. Range: 0–65535; Default value: 10; Unit: second.
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

2.3.9. AT+QISTATE Query Socket Service Status

This command queries the socket service status. If **<query_type>**=0, it will return the status of all existing socket services in the specified context. If **<query_type>**=1, it will return the status of a specified socket service.

AT+QISTATE Query Socket Service Status	
Test Command **AT+QISTATE=?**	Response: **+QISTATE: (**list of supported **<query_type>**s**),(**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**)** **OK**
Read Command/Execution Command **AT+QISTATE?** Or **AT+QISTATE**	Response: Return the status of all existing socket services: **+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<serverID>,<access_mode>,<AT_port>** **[…]** **OK**
Write Command If **<query_type>**=0, query the status of all socket services in a specified context **AT+QISTATE=<query_type>,<contextID>**	Response: Return the status of all existing socket service in a specified context **+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<serverID>,<access_mode>,<AT_port>** **[…]** **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command If **<query_type>**=1, query the status of specified socket service **AT+QISTATE=<query_type>,<connectID>**	Response: Return the status of specified socket service: **+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<serverID>,<access_mode>,<AT_port>** **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	300 ms

Characteristics	-

Parameter

<query_type> Integer type. The query type.
0 Query connection status of all existing socket services in a specified context
1 Query connection status of a specified socket service

<contextID> Integer type. The PDP context ID. Range: 1–5.

<connectID> Integer type. Socket ID. Range: 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> String type. 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> Integer type. Remote port number. Range: 1–65535.
If **<service_type>**="TCP" or "UDP", it is the port of remote server.
If **<service_type>**="TCP LISTENER" or "UDP SERVICE", **<remote_port>** is 0 and the port is invalid.
If **<service_type>**="TCP INCOMING", it is the port of remote client.

<local_port> Integer type. Local port number. Range: 0–65535.
If **<local_port>** is 0, 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 connection has been established
3 "Listening": server is listening
4 "Closing": connection is closed

<serverID> Integer type. It is valid only when **<service_type>**="TCP INCOMING". **<serverID>** indicates the server that accepts this TCP connection, and the value is the same as that of **<connectID>** of this server's "TCP LISTENER".

<access_mode> Integer type. Data access mode.
0 Buffer access mode
1 Direct push mode
2 Transparent transmission mode

<AT_port> String type. COM port of socket service.
"usbmodem" USB modem port
"usbat" USB AT port
"uart1" UART port 1

<err>	Integer type. Error codes. For more details, see ***chapter 4***.

2.3.10. AT+QISEND Send Data

This command sends Socket data in string type through the specified connection. If the data access mode of a specified socket service is buffer access mode (**<access_mode>**=0) or direct push mode (**<access_mode>**=1), the data can be sent by **AT+QISEND**. When the data is sent successfully, **SEND OK** will be returned. Otherwise, **SEND FAIL** or **ERROR** is returned.

- **SEND FAIL** indicates the sending buffer is full and resending of the data can be tried.
- **ERROR** indicates an error arises in the process of sending data. You should resend the data after a certain period of time. The maximum length is 1460 bytes.
- **SEND OK** does not mean the data has been sent to the server successfully. You can query whether the data has reached the server by executing **AT+QISEND=<connectID>,0**.

AT+QISEND Send Data	
Test Command **AT+QISEND=?**	Response: **+QISEND: (**range of supported **<connectID>**s**),(**range of supported **<send_length>**s**),<remoteIP>,(**range of supported **<remote_port>**s**)** **OK**
Write Command Send variable-length data when **<service_type>**="TCP", "UDP" or "TCP INCOMING" **AT+QISEND=<connectID>**	Response: **>** After the response **>**, input the data to be sent. Tap **Ctrl + Z** to send and tap **Esc** to cancel the operation If **<send_view_mode>**=0 (**AT+QICFG="sendinfo"**): If the connection has been established and the data is sent to the module successfully: **SEND OK** If the connection has been established but the sending buffer is full: **SEND FAIL** If there is any error: **ERROR** Or **+CME ERROR: <err>** If **<send_view_mode>**=1 (**AT+QICFG="sendinfo"**):

	If the connection has been established and the data is sent to the module successfully: **+QISEND: <connectID>,<status>,<free_size>** **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command Send fixed-length data when **<service_type>**="TCP", "UDP" or "TCP INCOMING" **AT+QISEND=<connectID>,<send_length>**	Response: **>** After the response **>,** input the data with the length equal to **<send_length>**. If **<send_view_mode>**=0 (**AT+QICFG="sendinfo"**): If the connection has been established and the data is sent to the module successfully: **SEND OK** If the connection has been established but the sending buffer is full: **SEND FAIL** If there is any error: **ERROR** Or **+CME ERROR: <err>** If **<send_view_mode>**=1 (**AT+QICFG="sendinfo"**): If the connection has been established and the data is sent to the module successfully: **+QISEND: <connectID>,<status>,<free_size>** **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command If **<service_type>**="UDP SERVICE" **AT+QISEND=<connectID>,<send_length>,<remoteIP>,<remote_port>**	Response: This command sends fixed-length data to a specified remote IP address and remote port. The **<service_type>** must be "UDP SERVICE".

	> After the response **>,** input the data with a length equal to **<send_length>** If the connection has been established and the data is sent successfully: **SEND OK** If the connection has been established but the sending buffer is full: **SEND FAIL** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command When **<send_length>**=0, query the sent data **AT+QISEND=<connectID>,0**	Response: If the specified connection exists: **+QISEND: <total_send_length>,<ackedbytes>,<unackedbytes>** **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	-
Characteristics	-

Parameter

<connectID>	Integer type. Socket ID. Range: 0–11.
<send_length>	Integer type. The length of data to be sent. Range: 0–1460; Unit: byte.
<remoteIP>	String type. The remote IP address (must be in dotted decimal format). It is valid only when **<service_type>**="UDP SERVICE".
<remote_port>	Integer type. Remote port. Range: 1–65535. It is valid only when **<service_type>**="UDP SERVICE".
<total_send_length>	Integer type. The total length of sent data. Unit: byte.
<ackedbytes>	Integer type. The total length of received data. Unit: byte.
<unackedbytes>	Integer type. The total length of data that are not received. Unit: byte.
<status>	Integer type.

	0 The data is sent to socket buffer successfully 1 The received buffer is full and fail to send
<free_size>	Integer type. The remaining space of the current buffer area. Range: 0–10240; Unit: byte.
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

2.3.11. AT+QIRD Read Received TCP/IP Data

In buffer access mode, module buffers data after receiving it and reports **+QIURC: "recv",<connectID>**, and then the data can be read by **AT+QIRD**.

Please note that if the buffer is not empty, when the module receives data again, it will not report a new URC until all the received data has been read.

AT+QIRD Read Received TCP/IP Data	
Test Command **AT+QIRD=?**	Response: **+QIRD: (**range of supported **<connectID>**s**),(**range of supported **<read_length>**s**)** **OK**
Write Command When **<service_type>**="TCP"、"UDP" or "TCP INCOMING" **AT+QIRD=<connectID>[,<read_length>]**	Response: If the specified connection has received the data: **+QIRD: <read_actual_length><CR><LF><data>** **OK** If there is no data: **+QIRD: 0** **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command When **<service_type>**="UDP SERVICE" **AT+QIRD=<connectID>**	Response: If data exists: **+QIRD: <read_actual_length>,<remoteIP>,<remote_port><CR><LF><data>** **OK** If there is no data:

	+QIRD: 0 **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Write Command When **<read_length>**=0, query the data that has been read **AT+QIRD=<connectID>,0**	Response: If the specified connection exists: **+QIRD: <total_receive_length>,<have_read_length>,<unread_length>** **OK** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	-
Characteristics	-

Parameter

<connectID>	Integer type. Socket ID. Range: 0–11.
<read_length>	Integer type. The maximum length of data to be read. Range: 0–1500; Unit: byte.
<read_actual_length>	Integer type. The length of data that has been actually read. Unit: byte.
<remoteIP>	String type. The remote IP address. It is valid only when **<service_type>**="UDP SERVICE".
<remote_port>	Integer type. Remote port. Range: 0–65535. It is valid only when **<service_type>**="UDP SERVICE".
<data>	The data that has been read.
<total_receive_length>	Integer type. The total length of the received data. Unit: byte.
<have_read_length>	Integer type. The length of data that has been read. Unit: byte.
<unread_length>	Integer type. The length of data that has not been read. Unit: byte.
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

2.3.12. AT+QISWTMD Switch Data Access Mode

This command switches the data access mode, including buffer access mode, direct push mode and transparent transmission mode. When a socket service is established, the data access mode can be specified by **<access_mode>** in **AT+QIOPEN**. After a socket service has been opened, the data access mode can be switched by **AT+QISWTMD**.

AT+QISWTMD Switch Data Access Mode	
Test Command **AT+QISWTMD=?**	Response **+QISWTMD: (**range of supported **<connectID>**s**),(**range of supported **<access_mode>**s**)** **OK**
Write Command **AT+QISWTMD=<connectID>,<access_mode>**	Response If **<access_mode>**=0 or 1 and data access mode is switched successfully: **OK** If **<access_mode>**=2 and data access mode is switched successfully, the module will enter transparent transmission mode: **CONNECT** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	-
Characteristic	This command takes effect immediately. The configurations are not saved.

Parameter

<connectID>	Integer type. Socket ID. Range: 0–11.
<access_mode>	Integer type. Data access mode. 0 Buffer access mode 1 Direct push mode 2 Transparent transmisson mode
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

2.3.13. AT+QPING Ping Remote Server

This command tests the Internet protocol reachability of a host. Before conducting Ping operations, the host should activate the context corresponding to **<contextID>** by **AT+QIACT**. It will return the result within **<timeout>** and the default value of **<timeout>** is 4 seconds.

AT+QPING Ping Remote Server	
Test Command **AT+QPING=?**	Response: **+QPING: (**range of supported **<contextID>**s**),<host>,(**range of supported **<timeout>**s**),(**range of supported **<pingnum>**s**),(**range of supported **<ping_size>**s**)** **OK**
Write Command **AT+QPING=<contextID>,<host>[,<timeout>[,<pingnum>[,<ping_size>]]]**	Response If a remote server is Pinged successfully: **OK** **+QPING: <result>[,<IP_address>,<bytes>,<time>,<ttl>]** **[…]** **+QPING: <finresult>[,<sent>,<rcvd>,<lost>,<min>,<max>,<avg>]** If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	-
Characteristic	This command takes effect immediately. The configurations are not saved.

Parameter

<contextID>	Integer type. The PDP context ID. Range: 1–5.
<host>	String type. The host address. The format is a domain name or a dotted decimal IP address.
<timeout>	Integer type. Sets the maximum time to wait for the response of each Ping request. Range: 1–255; Default value: 4; Unit: s.
<pingnum>	Integer type. Sets the maximum number of times for sending Ping request. Range: 1–10; Default value: 4.
<ping_size>	Integer type. Sets the maximum byte length of each Ping request Range: 32–1500;

	Default value: 32; Unit: byte.
<result>	Integer type. The result of each Ping request. 0 Received the Ping response from the server. In this case, detailed information is contained in **<IP_address>,<bytes>,<time>,<ttl>**. Other values Error codes. Please refer to ***Chapter 4***.
<IP_address>	String type. The IP address of the remote server in dotted decimal format.
<bytes>	Integer type. The actual length of each sent Ping request. Unit: byte.
<time>	Integer type. The time wait for the response of the Ping request. Unit: ms.
<ttl>	Integer type. Time-to-live (TTL) value of the response packet of the Ping request.
<finresult>	Integer type. 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, detailed information is contained in **<sent>,<rcvd>,<lost>,<min>,<max>,<avg>** Other values Error codes. Please refer to ***Chapter 4.***
<sent>	Integer type. Number of times the Ping requests that have been sent.
<rcvd>	Integer type. Number of the Ping requests that have received the response.
<lost>	Integer type. Number of the Ping requests that are timed out.
<min>	Integer type. The minimum response time. Unit: ms.
<max>	Integer type. The maximum response time. Unit: ms.
<avg>	Integer type. The average response time. Unit: ms.
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

2.3.14. AT+QIDNSCFG Configure Address of DNS Server

This command configures address of DNS server. Before the DNS address is set, the host must activate the context corresponding to **<contextID>** by **AT+QIACT** first.

AT+QIDNSCFG Configure Address of DNS Server	
Test command **AT+QIDNSCFG=?**	Response **+QIDNSCFG:** (range of supported **<contextID>**s)**,<pridnsaddr>,<secdnsaddr>** **OK**
Write Command **AT+QIDNSCFG=<contextID>[,<pridnsaddr>[,<secdnsaddr>]]**	Response If the optional parameters are omitted, query the current DNS server address of a specified PDP context: **+QIDNSCFG: <contextID>,<pridnsaddr>,<secdnsaddr>** **OK** If the optional parameters are specified, configure the primary and secondary DNS server addresses of a specified PDP context: **OK**

	If there is any error: **ERROR** Or **+CME ERROR: <err>**
Maximum Response Time	-
Characteristics	This command takes effect immediately. The configurations are not saved.

Parameter

<contextID>	Integer type. The PDP context ID. Range: 1–5.
<pridnsaddr>	String type. The primary DNS server address.
<secdnsaddr>	String type. The secondary DNS server address.
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

NOTE

1. Only IPv4 DNS address can be set in IPv4.
2. Only IPv6 DNS address can be set in IPv6.

2.3.15. AT+QISDE Control Whether to Echo the Data to be Sent by AT+QISEND

This command controls whether to echo the data to be sent by **AT+QISEND**, that is, whether to echo the input data to be sent.

AT+QISDE Control Whether to Echo the Data to be Sent by AT+QISEND	
Test Command **AT+QISDE=?**	Response **+QISDE: (**list of supported **<echo>**s**)** **OK**
Read Command **AT+QISDE?**	Response **+QISDE: <echo>** **OK**
Write Command **AT+QISDE=<echo>**	Response **OK** If there is any error: **ERROR** Or

	+CME ERROR: <err>
Maximum Response Time	-
Characteristic	This command takes effect immediately. The configurations are not saved.

Parameter

<echo>	Integer type. Whether to echo the data to be sent by **AT+QISEND**. 0 Not echo the data 1 Echo the data
<err>	Integer type. Error codes. For more details, see ***chapter 4***.

2.3.16. AT+QIGETERROR Query Error Code of the Last AT Command

If **ERROR** is returned after TCP/IP related commands are executed, the details of error can be queried by **AT+QIGETERROR**. Please note that **AT+QIGETERROR** only returns the error code of the last TCP/IP related AT command.

AT+QIGETERROR Query Error Code of the Last AT Command	
Test Command **AT+QIGETERROR=?**	Response **OK**
Execution Command **AT+QIGETERROR**	Response **+QIGETERROR: <err>,<errcode_description>** **OK**
Maximum Response Time	-
Characteristic	-

Parameter

<err>	Integer type. Error codes. For more details, see ***chapter 4***.
<errcode_description>	String type. Indicates the details of error. Please refer to ***Chapter 4*** for details.

2.4. URC Description

+QIURC: is used at the beginning of URCs of TCP/IP related AT commands to be reported to the host. The URC contains the reports about incoming data, closed connection, incoming connection and so on. Actually, **<CR><LF>** occurs at both the beginning and end of URCs, but **<CR><LF>** is not presented intentionally in this document.

2.4.1. +QIURC: "closed" URC Indicating Closed Connection

When TCP socket service is closed by remote client or due to network error, the URC will be reported, and the status of socket service will be "closing" (**<socket_state>**=4). **AT+QICLOSE=<connectID>** can be used to restore the **<socket_state>** to "initial" state.

+QIURC: "closed" URC Indicating Closed Connection	
+QIURC: "closed",<connectID>	Socket service connection is closed.

Parameter

<connectID>	Integer type. Socket ID. Range: 0–11.

2.4.2. +QIURC: "recv" URC Indicating Incoming Data

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

- In buffer access mode: the URC format is **+QIURC: "recv",<connectID>**. The host can read data by **AT+QIRD** after URC is reported. Please note that if the buffer is not empty, when the module receives data again, it will not report a new URC until all the received data has been read from the buffer by **AT+QIRD**.
- In direct push mode: the received data will be outputted via COM port directly.

+QIURC: "recv" URC Indicating Incoming Data	
+QIURC: "recv",<connectID>	It will be reported when the module receives the incoming data in buffer access mode. The host can receive data by **AT+QIRD**.
+QIURC: "recv",<connectID>,<current_recvlength><CR><LF><data>	It will be reported when the module receives the incoming data and the **<service_type>**="TCP", "UDP", or "TCP INCOMING" in direct push mode.
+QIURC: "recv",<connectID>,<curre	It will be reported when the module receives the incoming data

nt_recvlength>,<remoteIP>,<remote_port><CR><LF><data>	and **<service_type>**="UDP SERVICE" in direct push mode.

Parameter

<connectID>	Integer type. Socket ID. Range: 0–11.
<currentrecvlength>	Integer type. The length of actually received data.
<remoteIP>	String type. Remote IP address (It must be dotted decimal format).
<remote_port>	Integer type. Remote port. Range: 1–65535.
<data>	The received data.

2.4.3. +QIURC: "incoming full" URC Indicating Incoming Connection Reaches Limit

If the incoming connection reaches limit, or no socket system resources can be allocated, the module will report the URC **+QIURC: "incoming full",<serverID>** when there is new incoming connection request.

+QIURC: "incoming full" URC Indicating Incoming Connection Reaches Limit	
+QIURC: "incoming full"	The URC indicates the incoming connection reaches limit.

2.4.4. +QIURC: "incoming" URC Indicating Incoming Client Connection

If the **<service_type>**="TCP LISTENER", when a remote client connects to this server, the host will automatically assign an available **<connectID>** for the new connection. The range of **<connectID>** is 0–11. In such case, the module will report the URC. The **<service_type>** of the new connection will be "TCP INCOMING", and the **<access_mode>** will be buffer access mode.

+QIURC: "incoming" URC Indicating Incoming Client 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 reported by this URC.

Parameter

<connectID>	Integer type. Socket ID that is automatically specified by the module for the incoming client connection. Range: 0–11.
<serverID>	Integer type. Server of accepting the incoming **<connectID>**. **<service_type>** is "TCP LISTENER" and listening socket ID is **<serverID>** of the server.
<remoteIP>	String type. Remote IP address of the incoming **<connectID>**. Range: 1–65535.
<remote_port>	Integer type. Remote port of the incoming **<connectID>**. Range: 1–65535.

2.4.5. +QIURC: "pdpdeact" URC Indicating PDP Deactivation

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

+QIURC: "pdpdeact" URC Indicating PDP Deactivation	
+QIURC: "pdpdeact",<contextID>	The context corresponding to **<contextID>** is deactivated.

Parameter

<contextID>	Integer type. The PDP context ID. Range: 1–5.

3 Example

3.1. Configure and Activate a Context

3.1.1. Configure a Context

```
AT+QICSGP=1,1,"UNINET","","",0          //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
                                          response time is 150 s.
OK                                      //Activated the context successfully.
AT+QIACT?                               //Query the context state.
+QIACT: 1,1,1,"10.7.157.1"

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 40 s.
```

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 by AT+QIACT.
OK
```

```
+QIOPEN: 0,0                                  //TCP client is connected successfully. It is
                                               recommended to wait for the URC for 150 s.
                                               If it cannot be received in 150 s, the host
                                               could use AT+QICLOSE to close the socket.
AT+QISTATE=1,0                                 //Check whether the connection status of
                                                <connectID> is 0.
+QISTATE: 0,"TCP","220.180.239.201",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 by 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.
+QISEND: 9,9,0

OK
```

3.2.3. Receive Data from Remote Server in Buffer Access Mode

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

OK
AT+QICFG="recvind",1
OK
+QIURC: "recv",0,5           //5-byte data received when <connectID>=0
AT+QIRD=0,1500               //Read data and the length of data is 1500 bytes.
+QIRD: 5                     //The length of actually 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: 10,10,0

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 10 s.
OK
```

3.3. TCP Client Works in Transparent Transmission Mode

3.3.1. Set up a TCP Client Connection and Enter Transparent Transmission 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.
CONNECT                                          //TCP client is connected successfully. It is
                                                 recommended to wait for the URC CONNECT
                                                 for 150 s. If it cannot be received in 150 s, the
                                                 host could use AT+QICLOSE to close the
                                                 socket.
```

3.3.2. Send Data in Transparent Transmission Mode

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

3.3.3. Receive Data from Remote Server in Transparent Transmission 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
                                              transmission mode, the host could use
                                              AT+QICLOSE to close the TCP connection.
                                              Depending on the network, the maximum
                                              response time is 10 s.
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
                                                 executing AT+QIOPEN, the host should
                                                 activate the context with AT+QIACT.
OK

+QIOPEN: 0,0                                   //TCP client is connected successfully. It is
                                               suggested to wait for the URC for 150 s. If it
                                               cannot be received in 150 s, the host could use
                                               AT+QICLOSE to close Socket.
AT+QISTATE=1,0                                   //Query whether the connection status of
                                                 <connectID> is 0.
+QISTATE: 0,"TCP","220.180.239.201",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. 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

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 10 s.
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.
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","127.0.0.1",0,2020,3,1,1,0,"usbmodem"

OK
```

3.5.2. Accept TCP Incoming Connection from Client

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

3.5.3. Receive Data from Client

```
+QIURC: "recv",11                             //Received 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 total length of received data, including
                                                 read and unread data.
+QIRD: 4,4,0

OK
```

3.5.4. Close a TCP Server Connection

```
AT+QICLOSE=11                                 //Close the incoming connection. Depending on
                                               the network, the maximum response time is 10 s.
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. The <connectID> is 2
                                                   and <contextID> is 1. Before using
                                                   AT+QIOPEN, the host should activate the
                                                   context with AT+QIACT first.
OK

+QIOPEN: 2,0                                     //UDP service is started successfully.
AT+QISTATE=0,1                                    //Query whether the connection status of
                                                   <connectID> is 1.
```

```
+QISTATE: 2,"UDP SERVICE","127.0.0.1",0,3030,2,1,2,0,"usbmodem"

OK
```

3.6.2. Send UDP Data to Remote Server

```
AT+QISEND=2,10,"10.7.89.10",6969          //Send 10-byte data to remote server whose IP
                                           is 10.7.89.10 and the remote port is 6969.
>1234567890
SEND OK
```

3.6.3. Receive Data from Remote Side

```
+QIURC: “recv”,2                          //Receive data from remote side.
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 bytes. 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 the 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 target IP address, the host should activate the
                                                context with AT+QIACT.
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: 0,4,4,0,192,479,288
```

3.8. Get Last Error Code

```
AT+QIOPEN=1,"TCP","220.180.239.212",8009,0,1  //Open a socket service, and <connectID> is not
                                                specified.
ERROR
AT+QIGETERROR
+QIGETERROR: 552, invalid parameters

OK
```

4 Summary of Error Codes

If **ERROR** is returned after executing TCP/IP related AT commands, the details of error type can be queried by **AT+QIGETERROR**. Please note that **AT+QIGETERROR** just returns error code of the last TCP/IP AT command.

Table 2: Summary of Error Codes

<err>	<errcode_description>	Meaning
0	Operation success	Operation success
550	Unknown error	Unknown error
551	Operation blocked	Operation blocked
552	Invalid parameters	Invalid parameters
553	Memory not enough	Memory not enough
554	Socket creation failed	Socket creation failed
555	Operation not supported	Operation not supported
556	Socket bind failed	Socket bind failed
557	Socket listen failed	Socket listen failed
558	Socket write failed	Socket write failed
559	Socket read failed	Socket read failed
560	Socket accept failed	Socket accept failed
561	PDP context opening failed	PDP context open failed
562	PDP context closure failed	PDP context close failed
563	Socket identity has been used	Socket identity has been used
564	DNS busy	DNS busy
565	DNS parse failed	DNS parse failed

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

5 Appendix Reference

Table 3: Related Document

Document Name
[1] Quectel_RGx00U&RM500U_Series_AT_Commands_Manual

Table 4: Terms and Abbreviations

Abbreviation	Description
3GPP	3rd Generation Partnership Project
ACK	Acknowledgement
APN	Access Point Name
ASCII	American Standard Code for Information Interchange
CHAP	Challenge Handshake Authentication Protocol
CS	Circuit Switching
DNS	Domain Name System
FIN	Finish
ID	Identifier
IP	Internet Protocol
NTP	Network Time Protocol
NV	Non-Volatile
PAP	Password Authentication Protocol l
PDP	Packet Data Protocol
PPP	Point-to-Point Protocol
PS	Packet Switching
QoS	Quality of Service

TCP/IP	Transmission Control Protocol/Internet Protocol
TTL	Time to Live
UART	Universal Asynchronous Receiver& Transmitter
UDP	User Datagram Protocol
URC	Unsolicited Result Code
USB	Universal Serial Bus
(U)SIM	(Universal) Subscriber Identity Module
UTC	Coordinated Universal Time