BG95&BG77&BG600L Series

TCP/IP Application Note

LPWA Module Series

Version: 1.2

Date: 2022-06-15

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
1.0	2019-08-12	Terrence YANG	Initial
1.1	2021-03-29	Terrence YANG/ Sherlock ZHAO	1. Added an applicable module BG600L-M3. 2. Updated the valid value range of <timeout> in AT+QICLOSE from 1–65535 to 0–65535 (Chapter 2.3.5). 3. Added <RAI_info> for AT+QISENDEX to configure RAI (Chapter 2.3.9). 4. Changed the default value of <max_backoffs> of AT+QICFG="tcp/retranscfg" from 8 to 12 (Chapter 2.3.15). 5. Added AT+QICFG="sack" and AT+QICFG="tcp/keepalive" commands (Chapter 2.3.15). 6. Added some examples about TCP/UDP operations under IPv6 (Chapter 3.3, 3.7 and 3.9) and an example about UDP client with RAI feature (Chapter 3.10).
1.2	2022-06-15	Water WANG/ Sherlock ZHAO/ Terrence YANG	1. Added notes about AT+QICSGP (Chapter 2.3.1). 2. Added the maximum length of <username>, <password> in AT+QICSGP (Chapter 2.3.1). 3. Added the maximum length of <domain_name> in AT+QIOPEN (Chapter 2.3.4). 4. Added the valid value range of <remote_port> and <local_port> (Chapter 2.3). 5. Added AT+QICFG="tcp/inactivetm", AT+QICFG="ipv6prior" and AT+QICFG="asyncmode" (Chapter 2.3.15).

Contents

Table Index

1 Introduction

Quectel BG95 series, BG77 and BG600L-M3 modules feature an 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 series, BG77 and BG600L-M3 modules provide the following socket services:

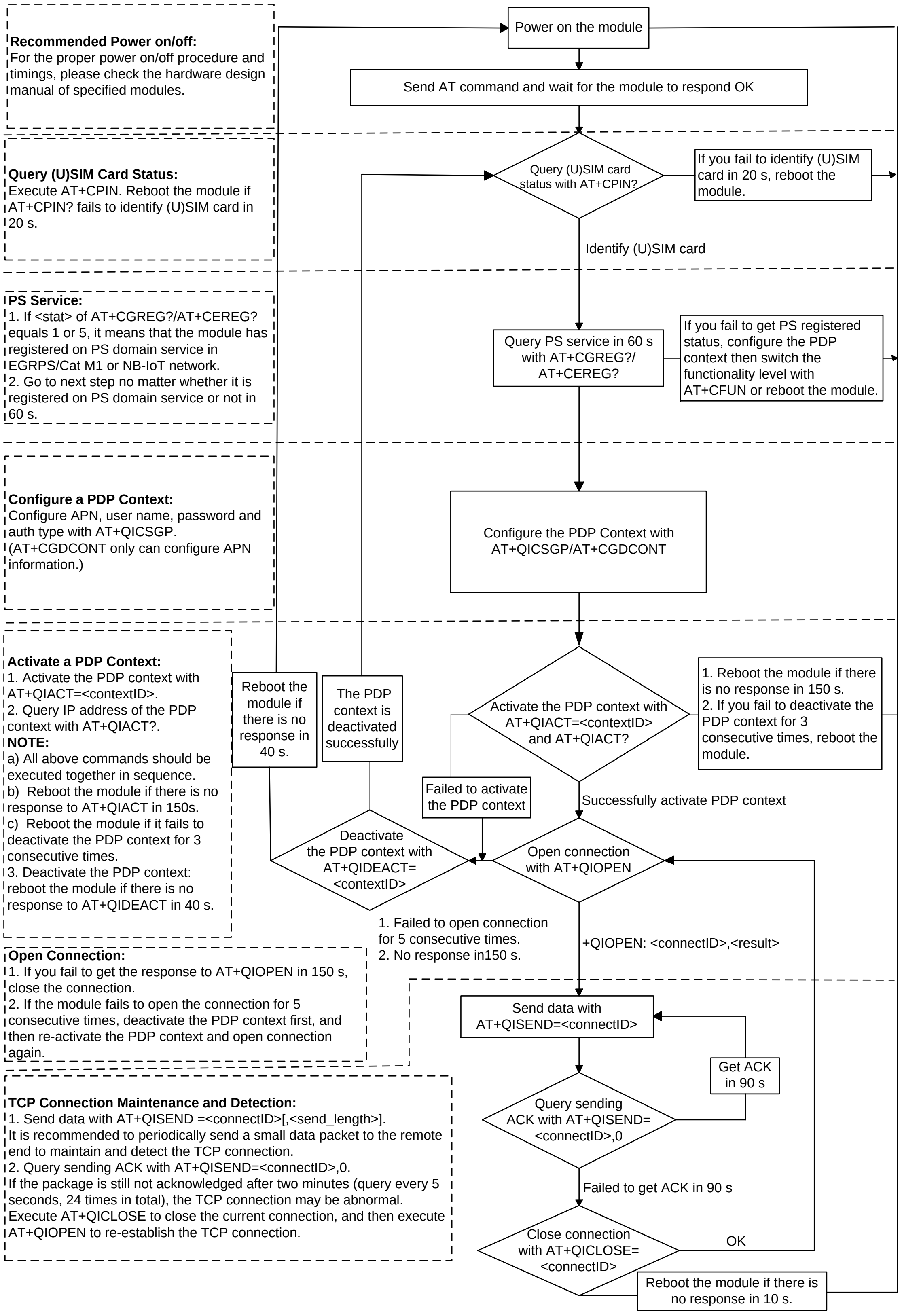
- TCP client
- UDP client
- TCP server
- UDP server

NOTE

See the firmware release notes of corresponding module models to check whether the function has been supported.

1.1. Usage of TCP/IP AT Commands

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


Power on the module
Send AT command and wait for the module to respond OK
Query (U)SIM card status with AT+CPIN?
If you fail to identify (U)SIM card in 20 s, reboot the module.
Identify (U)SIM card
Query PS service in 60 s with AT+CGREG?/ AT+CEREG?
If you fail to get PS registered status, configure the PDP context then switch the functionality level with AT+CFUN or reboot the module.
Configure the PDP Context with AT+QICSGP/AT+CGDCONT
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 you fail to deactivate the PDP context for 3 consecutive times, reboot the module.
Reboot the module if there is no response in 40 s.
The PDP context is deactivated successfully
Failed to activate the PDP context
Successfully activate PDP context
Deactivate the PDP context with AT+QIDEACT= <contextID>
Open connection with AT+QIOPEN
1. Failed to open connection for 5 consecutive times.
2. No response in150 s.
+QIOPEN: <connectID>,<result>
Send data with AT+QISEND=<connectID>
Get ACK in 90 s
Query sending ACK with AT+QISEND= <connectID>,0
Failed to get ACK in 90 s
Close connection with AT+QICLOSE= <connectID>
OK
Reboot the module if there is no response in 10 s.

Recommended Power on/off:
For the proper power on/off procedure and timings, please check the hardware design manual of specified modules.

Query (U)SIM Card Status:
Execute AT+CPIN. Reboot the module if AT+CPIN? fails to identify (U)SIM card in 20 s.

PS Service:
1. If <stat> of AT+CGREG?/AT+CEREG? equals 1 or 5, it means that the module has registered on PS domain service in EGRPS/Cat M1 or NB-IoT network.
2. Go to next step no matter whether it is registered on PS domain service or not in 60 s.

Configure a PDP Context:
Configure APN, user name, password and auth type with AT+QICSGP.
(AT+CGDCONT only can configure APN information.)

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

Open Connection:
1. If you fail to get the response to AT+QIOPEN in 150 s, close the connection.
2. If the module fails to open the connection for 5 consecutive times, deactivate the PDP context first, and then re-activate the PDP context and open connection again.

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 with AT+QISEND=<connectID>,0.
If the package is still not acknowledged after two minutes (query every 5 seconds, 24 times in total), the TCP connection may be abnormal. Execute AT+QICLOSE to close the current connection, and then execute AT+QIOPEN to re-establish the TCP connection.

NOTE:
1. Please note that you need to wait for the final response (for example OK, CME error, CMS error) of the last AT command you entered before you enter the next AT command. You can reboot the module if the module fails to get response in 60 s.
2. Reboot the module if you do not get the response to AT+QIACT in 150 s or response to AT+QICLOSE in 10 s and AT+QIDEACT in 40 s.
3. DO NOT reboot the module too frequently. If the module has been rebooted for 3 consecutive times due to a failed AT command, it can immediately be rebooted for the fourth time. If the command fails again, reboot the module for the fifth time in 10 minutes. If the command fails, reboot for the sixth time in 30 minutes, and if the command still fails, reboot for the seventh time in 1 hour.

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

1.2. Description of Data Access Modes

The modules support three data access modes:

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

When opening a socket service with **AT+QIOPEN**, you can specify the data access mode with **<access_mode>**. After the socket service is opened, the data access mode can be changed with **AT+QISWTMD**.

1. In buffer access mode, the module buffers data upon receiving them and reports a URC in the format of **+QIURC: "recv",<connectID>[,<data_len>]** to notify the host of the incoming data. In this case, the host can retrieve the buffered data with **AT+QIRD**.

2. In direct push mode, the module outputs the received data directly through a URC in the format of **+QIURC: "recv",<connectID>,<currentrecvlength><CR><LF><data>** or **+QIURC: "recv",<connectID>,<currentrecvlength>,<remoteIP>,<remote_port><CR><LF><data>**.
   See ***Chapter 2.4.2*** for more details.

3. In transparent transmission mode, the corresponding COM port (such as UART port, USB modem port, etc.) is exclusively used for sending/receiving data directly to/from the Internet. It cannot be used for other purposes such as running AT commands, etc.

   - **Exit transparent transmission mode**

   To make the module exit transparent transmission mode either:

   1) Execute **+++**. Follow the requirements below to prevent the **+++** from being misinterpreted as data:

      a) Do not input any character for at least 1 second before and after inputting **+++**.
      b) Input **+++** within 1 second, and wait until **OK** is returned. After **OK** is returned, the module switches to buffer access mode.

   OR

   2) Change MAIN_DTR from LOW to HIGH to make the module enter command mode. In this case, set **AT&D1** (see ***document [1]***) before the module enters transparent transmission mode.

   - **Return to transparent transmission mode**

   To return to transparent transmission mode either:

   1) Execute **AT+QISWTMD**. Specify **<access_mode>** as 2 when executing this command. Once transparent transmission mode is entered successfully, **CONNECT** is returned.

OR

2) Execute **ATO**. After a connection exits from transparent transmission mode, executing **ATO** switches the data access mode back to transparent transmission mode. Once transparent transmission mode is entered successfully, **CONNECT** is returned. If no connection has entered transparent transmission mode before, **ATO** returns **NO CARRIER**.

NOTE

1. In buffer access mode, if the buffer is not empty, and the module receives data again, it does not report a new URC until all the received data have been retrieved with **AT+QIRD** from the buffer.
2. In transparent transmission mode, AT commands cannot be executed. If the socket connection is closed because of a network error or other errors, the module reports **NO CARRIER** and exits the transparent transmission mode. In this case, execute **AT+QICLOSE** to close the socket service.

2 TCP/IP AT Commands

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 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 the corresponding command and return information about the type, value, or range of its parameter.
Read Command	**AT+<cmd>?**	Check the current parameter value of the 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 TCP/IP AT Commands

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

This command configures the **<APN>**, **<username>**, **<password>** and other parameters of a TCP/IP context.

AT+QICSGP Configure Parameters of a TCP/IP Context	
Test Command **AT+QICSGP=?**	Response **+QICSGP: (**range of supported **<contextID>**s**),(**range of supported **<context_type>**s**),<APN>,<username>,<password>,(**range of supported **<authentication>**s**)** **OK**
Write Command Query/set a specified context **AT+QICSGP=<contextID>[,<context_type>,<APN>[,<username>,<password>[,<authentication>]]]**	Response If the optional parameters are omitted, query the current setting: **+QICSGP: <context_type>,<APN>,<username>,<password>,<authentication>** **OK** If any of the optional parameters is specified, set a specified context: **OK** Or **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately (actually determined by network). The configuration is saved automatically.

Parameter

<contextID>	Integer type. Context ID. Range: 1–16.
<context_type>	Integer type. Protocol type. 1 IPv4 2 IPv6 3 IPv4v6
<APN>	String type. Access Point Name.
<username>	String type. Username. Maximum length: 128 bytes.
<password>	String type. Password. Maximum length: 128 bytes.
<authentication>	Integer type. Authentication methods. 0 None 1 PAP 2 CHAP 3 PAP or CHAP

NOTE

1. The configurations of **<context_type>** and **<authentication>** are saved automatically. This means that if **<context_type>** or **<authentication>** has been configured before, the next time when **AT+QICSGP** is executed with any of the two parameters omitted, the parameter setting equals its previous value rather than the default one.
2. On BG95 series, BG77 and BG600L-M3 modules, **AT+QICSGP** cannot configure a same APN profile on different context IDs.

Example

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

OK
AT+QICSGP=1,1,"CMNBIOT","","",1          //Configure context 1. China Mobile NB-IoT network APN:
                                          CMNBIOT.
OK
```

2.3.2. AT+QIACT Activate a PDP Context

Before activating a PDP context with **AT+QIACT**, the context should be configured with **AT+QICSGP**. After activation, the IP address can be queried with **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 max. 150 seconds to return **OK** or **ERROR** after executing **AT+QIACT**. Other AT commands can be executed only after the response is returned.

AT+QIACT Activate a PDP Context	
Test Command **AT+QIACT=?**	Response **+QIACT: (**range of supported **<contextID>**s**)** **OK**
Read command **AT+QIACT?**	Response Return the list of the currently activated contexts and their IP addresses: **+QIACT: 1,<context_state>,<context_type>[,<IP_address>]** **[.....** **+QIACT: 16,<context_state>,<context_type>[,<IP_address>]]** **OK**
Write Command Activate a specified PDP context **AT+QIACT=<contextID>**	Response **OK** Or **ERROR**
Maximum Response Time	150 s, determined by the network.
Characteristics	/

Parameter

<contextID>	Integer type. Context ID. Range: 1–16.
<context_state>	Integer type. Context state. 0 Deactivated 1 Activated
<context_type>	Integer type. Protocol type. 1 IPv4 2 IPv6
<IP_address>	String type. Local IP address after the context is activated.

2.3.3. AT+QIDEACT Deactivate a PDP Context

This command deactivates a specific context and closes all TCP/IP connections set up in this context. Depending on the network, it may take max. 40 seconds to return **OK** or **ERROR** after executing **AT+QIDEACT**. Other AT commands can be executed only after the response is returned.

AT+QIDEACT Deactivate a PDP Context	
Test Command **AT+QIDEACT=?**	Response **+QIDEACT: (**range of supported **<contextID>**s**)** **OK**
Write Command **AT+QIDEACT=<contextID>**	Response **OK** Or **ERROR**
Maximum Response Time	40 seconds, determined by the network.
Characteristics	/

Parameter

<contextID>	Integer type. Context ID. Range: 1–16.

2.3.4. AT+QIOPEN Open a Socket Service

This command opens a 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 response **+QIOPEN: <connectID>,<result>** indicates whether the socket service has been opened successfully.

1. If **<service_type>** is "TCP LISTENER", the module works as a TCP server. After accepting a new TCP connection, the module automatically specifies a **<connectID>** and reports the 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 as 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>[,<data_len>]**, and then the received data can be retrieved with **AT+QIRD=<connectID>**.

3. It is suggested to wait for 150 seconds for **+QIOPEN: <connectID>,<result>** to be outputted after

executing the Write Command. If the response cannot be received in 150 seconds, use **AT+QICLOSE** to close the socket.

AT+QIOPEN Open a Socket Service	
Test Command **AT+QIOPEN=?**	Response **+QIOPEN: (**range of supported **<contextID>**s**),(**range of supported **<connectID>**s**),"TCP/UDP/TCP LISTENER/UDP SERVICE","<IP_address>/<domain_name>",(**range of supported **<remote_port>**s**),(**range of supported **<local_port>**s**),(**range of supported **<access_mode>**s**)** **OK**
Write Command **AT+QIOPEN=<contextID>,<connectID>,<service_type>,<IP_address>/<domain_name>,<remote_port>[,<local_port>[,<access_mode>]]**	Response If in transparent transmission mode (**<access_mode>**=2) and the service is opened successfully: **CONNECT** If there is any error: **ERROR** Error description can be got with **AT+QIGETERROR**. If in buffer access mode (**<access_mode>**=0) or direct push mode (**<access_mode>**=1): **OK** **+QIOPEN: <connectID>,<result>** **<result>** is 0 when the service is opened successfully. In other cases, **<result>** is not 0.
Maximum Response Time	150 seconds, determined by the network.
Characteristics	/

Parameter

<contextID>	Integer type. Context ID. Range: 1–16.	
<connectID>	Integer type. Socket service index. 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 for TCP incoming connections
	"UDP SERVICE"	Start a UDP service
<IP_address>	String type.	

If **<service_type>** is "TCP" or "UDP", it indicates the IP address of remote server, such as "220.180.239.212".
If **<service_type>** is "TCP LISTENER" or "UDP SERVICE", enter "127.0.0.1".

<domain_name> String type. Domain name address of the remote server.
Maximum length: 256 bytes.

<remote_port> Integer type. Port number of the remote server. Range: 0–65535.
If **<service_type>** is "TCP" or "UDP", this parameter must be specified. Range: 1–65535.
If **<service_type>** is "TCP LISTENER" or "UDP SERVICE", specify this parameter as 0.

<local_port> Integer type. Local port number. Range: 0–65535.
If **<service_type>** is "TCP LISTENER" or "UDP SERVICE", this parameter must be specified. Range: 1–65535.
If **<service_type>** is "TCP" or "UDP", this parameter can be omitted and the default value is 0. The local port is assigned automatically if **<local_port>** is 0. Otherwise the local port is assigned as specified.

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

<result> Integer type. Result code. See ***Chapter 4*** for details.

NOTE

1. Only one socket service can be set to transparent transmission mode at a time.
2. If **<service_type>** is "TCP LISTENER", **<access_mode>** cannot be transparent transmission mode. When a remote client connects to this server, **AT+QSWTMD** can be used to switch the incoming connection to transparent transmission mode.

2.3.5. AT+QICLOSE Close a Socket Service

This command closes a specified socket service. Depending on the network, it takes max. 10 seconds (default value, can be modified by **<timeout>**) to return **OK** or **ERROR** after executing **AT+QICLOSE**. Other AT commands can be executed only after the response is returned.

AT+QICLOSE Close a Socket Service	
Test Command **AT+QICLOSE=?**	Response **+QICLOSE: (**range of supported **<connectID>**s**),(**range of supported **<timeout>**s**)** **OK**
Write Command	Response

AT+QICLOSE=<connectID>[,<timeout>]	If the socket service is closed successfully: **OK** If the command failed to close the socket service: **ERROR**
Maximum Response Time	Determined by **<timeout>**
Characteristics	/

Parameter

<connectID>	Integer type. Socket service index. Range: 0–11.
<timeout>	Integer type. 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. Range: 0–65535. Default value: 10. Unit: second.

2.3.6. AT+QISTATE Query Socket Service Status

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

AT+QISTATE Query Socket Service Status	
Test Command **AT+QISTATE=?**	Response **OK**
Read/Execution Command **AT+QISTATE?** or **AT+QISTATE**	Response Return the status of all existing connections: **+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<serverID>,<access_mode>,<AT_port>** **[…]** **OK**
Write Command If **<query_type>** is 0, query the connection status of a specified context **AT+QISTATE=<query_type>,<contextID>**	Response Return the status of all existing connections in a specified context: **+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<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>,<connectID>**	Response **+QISTATE: <connectID>,<service_type>,<IP_address>,<remote_port>,<local_port>,<socket_state>,<contextID>,<serverID>,<access_mode>,<AT_port>** **OK**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<query_type> Integer type. 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. Context ID. Range: 1–16.

<connectID> Integer type. Socket service index. 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 for TCP incoming connections
"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>** is "TCP" or "UDP", it is the IP address of a remote server.
If **<service_type>** is "TCP LISTENER" or "UDP SERVICE", it is the local IP address.
If **<service_type>** is "TCP INCOMING", it is the IP address of a remote client.

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

<local_port> Integer type. Local port number. Range: 0–65535.
If **<local_port>** is 0, then the local port is assigned automatically.

<socket_state> Integer type. 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> Integer type. Only valid when **<service_type>** is "TCP INCOMING". **<serverID>** represents the ID of the server that accepts this TCP incoming connection, and the value is the same as **<connectID>** of "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 "uart1" UART1 port

2.3.7. AT+QISEND Send Data

In buffer access mode (**<access_mode>**=0) or direct push mode (**<access_mode>**=1), the data can be sent with **AT+QISEND**. If the data have been sent to the module successfully, **SEND OK** is returned, otherwise **SEND FAIL** or **ERROR** is returned.

- **SEND FAIL** indicates that the sending buffer is full. In this case, data can be resent.
- **ERROR** indicates an error in the data sending process. In this case, wait for some time before you resend the data.
- **SEND OK** means that the data have been sent to the peer, but it does not mean that they have reached the peer successfully. You can query whether the data have reached the peer with **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**)** **OK**
Write Command Send variable-length data when **<service_type>** is "TCP", "UDP" or "TCP INCOMING" **AT+QISEND=<connectID>**	Response **>** After the response **>**, input the data to be sent. Tap **Ctrl+Z** to send the data, and tap **Esc** to cancel the sending operation If the connection is established and the data is sent successfully: **SEND OK** If the connection is established but the sending buffer is full: **SEND FAIL** If the connection is not established, abnormally closed, or any parameter is incorrect: **ERROR**
Write Command	Response

Send fixed-length data when **<service_type>** is "TCP", "UDP" or "TCP INCOMING" **AT+QISEND=<connectID>,<send_length>**	**>** After the response **>**, input the data until the data length equals **<send_length>** If the connection is established and the data is sent successfully: **SEND OK** If the connection is established but the sending buffer is full: **SEND FAIL** If the connection is not established, abnormally closed, or any parameter is incorrect: **ERROR**
Write Command When **<service_type>** is "UDP SERVICE" **AT+QISEND=<connectID>,<send_length>,<remoteIP>,<remote_port>**	Response Send fixed length data to specified remote IP address and remote port. The **<service_type>** must be "UDP SERVICE". **>** After response **>**, type the data until the data length equals **<send_length>** If the connection is established and the data is sent successfully: **SEND OK** If the connection is established but the sending buffer is full: **SEND FAIL** If the connection is not established, abnormally closed, or any parameter is incorrect: **ERROR**
Write Command When **<send_length>** is 0, query the sent data **AT+QISEND=<connectID>,0**	Response If the specified **<connectID>** exists: **+QISEND: <total_send_length>,<ackedbytes>,<unackedbytes>** **OK** If the specified **<connectID>** does not exist, or there is any other error: **ERROR**
Maximum Response Time	120 s, determined by network.

Characteristics	/

Parameter

<connectID>	Integer type. Socket service index. Range: 0–11.
<send_length>	Integer type. Length of the data to be sent. Range: 0–1460. Unit: byte.
<remoteIP>	String type. Remote IP address (must be dot format). Valid only when **<service_type>** is "UDP SERVICE".
<remote_port>	Integer type. Remote port. Valid only when **<service_type>** is "UDP SERVICE". Range: 0–65535.
<total_send_length>	Integer type. Total length of sent data. Unit: byte.
<ackedbytes>	Integer type. Total length of acknowledged data. Unit: byte.
<unackedbytes>	Integer type. Total length of unacknowledged data. Unit: byte.

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

In buffer access mode, after receiving data, the module buffers them and reports **+QIURC: "recv",<connectID>[,<data_len>]**, then the data can be retrieved with **AT+QIRD**.

Please note that if the buffer is not empty, and the module receives data again, it does not report a new URC until all received data have been retrieved with **AT+QIRD** from the buffer.

AT+QIRD Retrieve the 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>** is "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 are no data: **+QIRD: 0** **OK** If the connection does not exist: **ERROR**

Write Command When **<service_type>** is "UDP SERVICE" **AT+QIRD=<connectID>**	Response If data exist: **+QIRD: <read_actual_length>,<remoteIP>,<remote_port><CR><LF><data>** **OK** If there are no data: **+QIRD: 0** **OK** If the connection does not exist: **ERROR**
Write Command When **<read_length>** is 0, query the retrieved data length **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**
Maximum Response Time	120 s, determined by network.
Characteristics	/

Parameter

<connectID>	Integer type. Socket service index. Range: 0–11.
<read_length>	Integer type. Maximum length of the data to be retrieved. Range: 0–1500. Default value: 1500. Unit: byte.
<read_actual_length>	Integer type. Length of actually retrieved data. Unit: byte.
<remoteIP>	String type. Remote IP address. Valid only when **<service_type>** is "UDP SERVICE".
<remote_port>	Integer type. Remote port number. Valid only when **<service_type>** is "UDP SERVICE". Range: 0–65535.
<data>	String type. Retrieved data.
<total_receive_length>	Integer type. Total length of received data. Unit: byte.
<have_read_length>	Integer type. Length of the data that have been retrieved. Unit: byte.
<unread_length>	Integer type. Length of the data that have not been retrieved. Unit: byte.

2.3.9. AT+QISENDEX Send Hex String Data

This command sends hex string data and cannot be applied for "UDP SERVICE" and "TCP LISTENER" sockets.

AT+QISENDEX Send Hex String Data	
Test Command **AT+QISENDEX=?**	Response **+QISENDEX: (**range of supported **<connectID>**s**),<hex_string>,(**range of supported **<RAI_info>**s**)** **OK**
Write Command **AT+QISENDEX=<connectID>,<hex_string>[,<RAI_info>]**	Response If the hex string is sent successfully: **SEND OK** If the sending buffer is full: **SEND FAIL** If the connection does not exist: **ERROR**
Maximum Response Time	120 s, determined by network.
Characteristics	/

Parameter

<connectID>	Integer type. Socket service index. Range: 0–11.
<hex_string>	String type. Hex string data. Maximum length: 512 bytes.
<RAI_info>	Integer type. RAI (Release Assistance Indication) information. 0 No RAI information available. 1 No further uplink or downlink data transmission. 2 Only a single downlink data transmission and no further uplink data transmission.

NOTE

1. RAI (Release Assistance Indication) is introduced in *3GPP Release 14* and it is primarily a MAC layer feature.
2. The RAI feature is only applicable to NB-IoT RAT. This feature allows tracking the predictable nature of data traffic and requests for connection release when there are no additional data to be sent. It is essentially a ‘fast dormancy’ feature to save UE battery power.

2.3.10. AT+QISWTMD Switch Data Access Mode

This command switches the data access mode among buffer access mode, direct push mode and transparent transmission mode. When starting a socket service, the data access mode can be specified via the **<access_mode>** of **AT+QIOPEN**. After opening a socket, the data access mode can be changed with **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>** is 0 or 1, and data access mode is switched successfully: **OK** If **<access_mode>** is 2 and data access mode is switched successfully, the module enters data mode (i.e. transparent transmission mode): **CONNECT** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<connectID>	Integer type. Socket service index. Range: 0–11.
<access_mode>	Integer type. Data access modes of the connection. 0 Buffer access mode 1 Direct push mode 2 Transparent transmission mode

2.3.11. AT+QPING Ping a Remote Host

This command tests the reachability of a host on an Internet protocol network. Before using the ping utility, the host should activate the context of the corresponding **<contextID>** with **AT+QIACT**. The command returns the result within **<timeout>**.

AT+QPING Ping a Remote Host	
Test Command **AT+QPING=?**	Response **+QPING: (**range of supported **<contextID>**s)**,<host>,(**range of supported **<timeout>**s)**,(**range of supported **<pingnum>**s) **OK**
Write Command **AT+QPING=<contextID>,<host>[,<timeout>[,<pingnum>]]**	Response If a remote host is pinged successfully: **OK** **+QPING: <ping_result>[,<IP_address>,<bytes>,<time>,<TTL>]** **[…]** **+QPING: <finresult>[,<sent>,<rcvd>,<lost>,<min>,<max>,<avg>]** If there is any error: **ERROR**
Maximum Response Time	300 s
Characteristics	/

Parameter

<contextID>	Integer type. Context ID. Range: 1–16.	
<host>	String type. Host address. It is a domain name or a dotted decimal IP address.	
<timeout>	Integer type. Maximum time to wait for the response of each ping request. Range: 1–255. Default value: 4. Unit: second.	
<pingnum>	Integer type. Maximum number of ping requests. Range: 1–10. Default value: 4.	
<ping_result>	Integer type. Result of each ping request.	
	0	Received the ping response from the host. In this case, it is followed by **,<IP_address>,<bytes>,<time>,<TTL>**.
	Others	See ***Chapter 4*** for specific result codes.
<IP_address>	String type. IP address of the remote host in dotted-decimal notation.	
<bytes>	Integer type. Length of each sent ping request. Unit: byte.	
<time>	Integer type. Time wait for the response of the ping request. Unit: ms.	
<TTL>	Integer type. Time to live value of the response packet for the ping request.	
<finresult>	Integer type. The final result of the command.	
	0	Pinged successfully. 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	See ***Chapter 4*** for specific result codes.
<sent>	Integer type. Total number of the ping requests that have been sent.	
<rcvd>	Integer type. Total number of the ping requests that have received responses.	
<lost>	Integer type. Total number of the ping requests that are time out.	
<min>	Integer type. Minimum response time. Unit: ms.	
<max>	Integer type. Maximum response time. Unit: ms.	
<avg>	Integer type. Average response time. Unit: ms.	

2.3.12. AT+QNTP Synchronize Local Time with NTP Server

This command synchronizes 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>** with **AT+QIACT**.

AT+QNTP Synchronize Local Time with NTP Server	
Test Command **AT+QNTP=?**	Response **+QNTP: (**range of supported **<contextID>**s**),<server>,(**range of supported **<port>**s**),(**list of supported **<auto_set_time>**s**)** **OK**
Read Command **AT+QNTP?**	Response If in the process of local time synchronization: **+QNTP: <server>,<port>** **OK**
Write Command **AT+QNTP=<contextID>,<server>[,<port>[,<auto_set_time>]]**	Response If the local time is synchronized with NTP server successfully: **OK** **+QNTP: <result>,<time>** If there is any error: **ERROR**
Maximum Response Time	125 s, determined by the network.
Characteristics	The command takes effect immediately (actually determined by network). The configuration is not saved.

Parameter

<contextID>	Integer type. Context ID. Range: 1–16.
<server>	String type. NTP server address.
<port>	Integer type. NTP server port number. Range: 1–65535. Default value: 123.
<auto_set_time>	Integer type. Whether to automatically set synchronized time as local time. 0 Not set 1 Set
<result>	Integer type. Result code. See ***Chapter 4*** for details.
<time>	String type. Time synchronized with NTP server. The format is "YYYY/MM/DD,hh:mm:ss±zz". Range of "zz": -48 to 56.

2.3.13. AT+QIDNSCFG Configure Address of DNS Server

Before setting the DNS server address, the host must activate the context of corresponding **<contextID>** with **AT+QIACT**.

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 addresses of a specified PDP context: **+QIDNSCFG: <contextID>,<pridnsaddr>,<secdnsaddr>** **OK** If any of the parameters is specified, configure the primary and secondary DNS server addresses of a specified PDP context: **OK** Or **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configurations will not be saved.

Parameter

<contextID>	Integer type. PDP context ID. Range: 1–16.

<pridnsaddr>	String type. Primary DNS server address.
<secdnsaddr>	String type. Secondary DNS server address.

2.3.14. AT+QIDNSGIP Get IP Address by Domain Name

This commands gets an IP address by domain name. Before querying the DNS, the host should activate the context of corresponding **<contextID>** with **AT+QIACT**.

AT+QIDNSGIP Get IP Address by Domain Name	
Test Command **AT+QIDNSGIP=?**	Response **+QIDNSGIP: (**range of supported **<contextID>**s**),<host_name>** **OK**
Write Command **AT+QIDNSGIP=<contextID>,<host_name>**	Response **OK** **+QIURC: "dnsgip",<result>,<IP_count>,<DNS_ttl>** **[.....** **+QIURC: "dnsgip",<host_IP_addr>]** If there is any error: **ERROR**
Maximum Response Time	60 s, determined by the network.
Characteristics	/

Parameter

<contextID>	Integer type. Context ID. Range: 1–16.
<host_name>	String type. Domain name.
<result>	Integer type. Result code of the operation. See ***Chapter 4*** for details.
<IP_count>	Integer type. Number of the IP addresses corresponding to the **<host_name>**.
<DNS_ttl>	Integer type. Time to live of the DNS.
<host_IP_addr>	String type. IP address of **<host_name>**.

2.3.15. 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 **<transpkt_size>**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: "recvind",(**list of supported **<recv_ind>**s) **+QICFG: "sack",(**list of supported **<SACK>**s) **+QICFG: "tcp/inactivetm",(**range of supported **<inactivetm>**s) **+QICFG: "tcp/keepalive",(**list of supported **<keepalive>**s)**,(**range of supported **<idle_time>**s)**,(**range of supported **<interval_time>**s),(range of supported **<probe_cnt>**>s) **+QICFG: "tcp/retranscfg",(**range of supported **<max_backoffs>**s)**,(**range of supported **<max_rto>**s) **+QICFG: "ipv6prior",(**list of supported **<IPv6_prior>**s) **+QICFG: "asyncmode",(**list of supported **<async_mode>**s) **OK**
Write Command Query/set the packet size for transparent transmission mode **AT+QICFG="transpktsize"[,<transpkt_size>]**	Response If the optional parameter is omitted, query the current setting: **+QICFG: "transpktsize",<transpkt_size>** **OK** If the optional parameter is specified, set the packet size for transparent transmission mode: **OK** Or **ERROR**
Write Command Query/set the waiting time for 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, set the waiting time for transparent transmission mode: **OK** Or **ERROR**
Write Command Query/set the format of the data to be sent and received (only for buffer access mode and direct push mode) **AT+QICFG="dataformat"[,<send**	Response If the optional parameter is omitted, query the current setting: **+QICFG: "dataformat",<send_data_format>,<recv_data_format>** **OK**

_data_format>,<recv_data_format>]	If the optional parameter is specified, set the format for the data to be sent and received: **OK** Or **ERROR**
Write Command Query/set the output format of received data (only for non-transparent 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, set the output format of received data: **OK** Or **ERROR**
Write Command Query/set the URC format in buffer access mode. **AT+QICFG="recvind",[<recv_ind>]**	Response If the optional parameter is omitted, query the current setting: **+QICFG: "recvind",<recv_ind>** **OK** If the optional parameter is specified, set the URC format in buffer access mode: **OK** Or **ERROR**
Write Command Query/set whether to enable SACK option in TCP header. **AT+QICFG="sack",[<SACK>]**	Response If the optional parameter is omitted, query the current setting: **+QICFG: "sack",<SACK>** **OK** If the optional parameter is specified, configure whether to enable SACK option in TCP header: **OK** Or **ERROR**
Write Command Query/set the TCP idle time. **AT+QICFG="tcp/inactivetm"[,<inactivetm>]**	Response If the optional parameter is omitted, query the current setting: **+QICFG: "tcp/inactivetm",<inactivetm>** **OK**

	If the optional parameter is specified, set TCP idle time: **OK** If there is any error: **ERROR**
Write Command Query/set whether to enable TCP keepalive feature and the relevant parameter values. **AT+QICFG="tcp/keepalive"[,<keepalive>[,<idle_time>,<interval_time>,<probe_cnt>]]**	Response If the optional parameters are omitted, query the current setting: **+QICFG: "tcp/keepalive",<keepalive>,<idle_time>,<interval_time>,<probe_cnt>** **OK** If any of the optional parameter is specified, set whether to enable TCP keepalive feature and the relevant parameter values: **OK** Or **ERROR**
Write Command Query/set the maximum interval time and number of 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, set the maximum interval and the number of TCP retransmission: **OK** Or **ERROR**
Write Command Query/set the prior PDP type when the APN profile is IPv4v6 **AT+QICFG="ipv6prior"[,<IPv6_prior>]**	Response If the optional parameter is omitted, query the current setting: **+QICFG: "ipv6prior",<IPv6_prior>** **OK** If the optional parameter is specified, set the prior PDP type when the APN profile is IPv4v6 **OK** If there is any error: **ERROR**
Write Command Query/set the execution mode of **AT+QIACT** **AT+QICFG="asyncmode"[,<async_mode>]**	Response If the optional parameter is omitted, query the current setting: **+QICFG: "asyncmode",<async_mode>** **OK**

	If the optional parameter is specified, set the execution mode of **AT+QIACT**: **OK** If there is any error: **ERROR**
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<transpkt_size> Integer type. Maximum length of the data packet to be sent. Range: 1–1460. Default value: 1024. Unit: byte.

<transwaittm> Integer type. In transparent transmission mode, if the length of data received from the port is less than the specified value of **<transpkt_size>**, after exceeding the time specified by **<transwaittm>**, the data will be sent directly. Range: 0–20. Default value: 2. Unit: 100 ms.

<send_data_format> Integer type. Format of the data to be sent.
0 Text mode
1 Hex mode

<recv_data_format> Integer type. Format of the data to be received.
0 Text mode
1 Hex mode

<view_mode> Integer type. Output format of received data.
0 Output format of received data: data header\r\ndata.
1 Output format of received data: data header,data.

<recv_ind> 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>**.

<SACK> Integer type. Whether to enable SACK option in TCP header.
0 Disable
1 Enable

<inactivetm> Integer type. TCP idle time.
0 Disable TCP idle timeout
1–65535 Idle timeout. Unit: seconds

<keepalive> Integer type. Whether to enable TCP keepalive feature. When it is 0, **<idle_time>**, **<interval_time>** and **<probe_cnt>** should be ignored.
0 Disable
1 Enable

<idle_time>	Integer type. Default keepalive time. Range: 1–120. Unit: minute.
<interval_time>	Integer type. Interval between keepalive probes. Range: 25–100. Unit: second.
<probe_cnt>	Integer type. Maximum number of keepalive probes. Range: 3–10.
<max_backoffs>	Integer type. Maximum number of TCP retransmissions. Range: 3–20. Default value: 12.
<max_rto>	Integer type. Maximum interval between TCP retransmissions. Range: 5–1000. Default value: 600. Unit: 100 ms.
<IPv6_prior>	Integer type. The prior PDP type to be set when the APN profile is IPv4v6. 0 IPv4 1 IPv6
<async_mode>	Integer type. Execution mode of **AT+QIACT**. In different execution modes, the maximum response time of **AT+QIACT** varies. In synchronous mode, the maximum response time is 150 s and determined by the network, while in synchronous mode, the response is returned immediately after command execution. 0 Synchronous mode 1 Asynchronous mode

NOTE

1. Settings of **AT+QICFG="tcp/keepalive",<keepalive>,<idle_time>,<interval_time>,<probe_cnt>** and **AT+QICFG="tcp/retranscfg",<max_backoffs>,<max_rto>** take effect to all TCP sockets.
2. Settings of **AT+QICFG="tcp/inactivetm"** take effect to socket type of "TCP", "TCP INCOMING" and "SSLClient". For more information on "SSLClient", see ***document [2]***.

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

This command controls whether to echo the data for **AT+QISEND**.

AT+QISDE Control Whether to Echo the Data for AT+QISEND	
Test Command **AT+QISDE=?**	Response **+QISDE: (**list of supported **<echo>**s**)** **OK**
Read Command **AT+QISDE?**	Response **+QISDE: <echo>** **OK**
Write Command **AT+QISDE=<echo>**	Response **OK** Or **ERROR**

Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. The configuration is not saved.

Parameter

<echo>	Integer type. Whether to echo the data for **AT+QISEND**. 0 Do not echo the data 1 Echo the data

2.3.17. AT+QIGETERROR Query the Last Result Code

If **<result>** is returned after executing TCP/IP commands, the detailed information about a result code can be queried with **AT+QIGETERROR**. Please note that **AT+QIGETERROR** just returns the result code of the last TCP/IP AT command.

AT+QIGETERROR Query the Last Result Code	
Test command **AT+QIGETERROR=?**	Response **OK**
Execution Command **AT+QIGETERROR**	Response **+QIGETERROR: <result>,<result_code_description>** **OK**
Maximum Response Time	300 ms
Characteristics	/

Parameter

<result>	Integer type. Result code. See ***Chapter 4*** for details.
<result_code_description>	String type. Result code description. See ***Chapter 4*** for details.

2.4. Description of URCs

The URC of TCP/IP AT commands will be reported to the host in the format that begins with **+QIURC:**. It contains the reports about incoming data, closed connection and incoming connection and so on. Actually, there is **<CR><LF>** both before and after URC, but **<CR><LF>** is not presented intentionally.

2.4.1. +QIURC: "closed" Indicate Connection Closed

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

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

Parameter

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

2.4.2. +QIURC: "recv" Indicate Incoming Data

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

In buffer access mode, after receiving data, the module reports **+QIURC: "recv",<connectID>[,<data_len>]** to notify the host. The host can then retrieve data with **AT+QIRD**. Please note that if the buffer is not empty, and the module receives data again, it does not report a new URC until all the received data have been retrieved with **AT+QIRD** from the buffer.

In direct push mode, the received data is outputted to COM port directly.

+QIURC: "recv" Indicate Incoming Data	
+QIURC: "recv",<connectID>[,<data_len>]	Indicates incoming data in buffer access mode. The host can retrieve data with **AT+QIRD**.
+QIURC: "recv",<connectID>,<currentrecvlength><CR><LF><data>	Indicates incoming data in direct push mode when the **<service_type>** is "TCP", "UDP" or "TCP INCOMING".
+QIURC: "recv",<connectID>,<currentrecvlength>,<remoteIP>,<remote_port><CR><LF><data>	Indicates incoming data in direct push mode when **<service_type>** is "UDP SERVICE".

Parameter

<connectID>	Integer type. Socket service index. Range: 0–11.
<data_len>	Integer type. Data length. Unit: byte.
<currentrecvlength>	Integer type. Length of actually received data.
<remoteIP>	String type. Remote IP address.
<remote_port>	Integer type. Remote port number.

<data>	String type. Received data.

2.4.3. +QIURC: "incoming full" Indicate Incoming Connection Reaches the Limit

If the number of incoming connections reaches the limit, or no socket system resources can be allocated, the module reports **+QIURC: "incoming full"** for the new incoming connection request.

+QIURC: "incoming full" Indicate Incoming Connection Reaches the Limit	
+QIURC: "incoming full"	Indicates that the number of incoming connections has reached the limit.

2.4.4. +QIURC: "incoming" Indicate Incoming Connection

If **<service_type>** is "TCP LISTENER", when a remote client connects to this server, the host automatically assigns a free **<connectID>** to the new connection, and the range of **<connectID>** is 0–11. In this case, the module reports **+QIURC: "incoming"**. The **<service_type>** of the new connection will be "TCP INCOMING", and the **<access_mode>** will be buffer access mode.

+QIURC: "incoming" Indicate Incoming Connection	
+QIURC: "incoming",<connectID>,<serverID>,<remoteIP>,<remote_port>	When the new incoming connection is accepted by **<serverID>**, the allocated **<connectID>**, **<remoteIP>** and **<remote_port>** are informed by this URC.

Parameter

<connectID>	Integer type. Index of the socket service assigned to the incoming connection, which is automatically specified by the module. Range: 0–11.
<serverID>	Integer type. ID of the listening socket whose **<service_type>** is "TCP LISTENER" and which accepts the incoming **<connectID>**.
<remoteIP>	String type. Remote IP address of the incoming **<connectID>**.
<remote_port>	Integer type. Remote port number of the incoming **<connectID>**.

2.4.5. +QIURC: "pdpdeact" Indicate PDP Deactivation

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

+QIURC: "pdpdeact" Indicate PDP Deactivation	
+QIURC: "pdpdeact",<contextID>	**<contextID>** context is deactivated.

Parameter

<contextID>	Integer type. Context ID. Range: 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. China Mobile NB-IoT network APN:
                                          CMNBIOT.
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, protocol type and IP address of context 1.
+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 under IPv4

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

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

```
OK

+QIOPEN: 0,0                    //TCP client connected successfully. It is suggested to wait for 150
                                seconds for the URC +QIOPEN: <connectID>,<result>. 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 have reached the server
                                     successfully. The host can query whether the data has reached
                                     the server with 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 of data have been sent, and the 9 bytes of
                                     data have 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 10 s by default.
OK
```

3.3. TCP Client Works in Buffer Access Mode under IPv6

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

```
//Context is 1 and <connectID> is 0. Before using AT+QIOPEN, the host should activate the context with
AT+QIACT.
AT+QIOPEN=1,0,"TCP","2001:468:2100:1:0:0:0:0",5001,0,0
OK

+QIOPEN: 0,0            //TCP client is connected successfully. It is suggested to wait for 150 seconds for
                         the URC +QIOPEN: <connectID>,<result>. If the URC cannot be received in
                         150 seconds, the host can use AT+QICLOSE to close the socket.
AT+QISTATE=1,0          //Query the connection status of socket service 1.
+QISTATE: 0,"TCP","2001:468:2100:1:0:0:0:0",5001,11921,2,1,0,0,"usbmodem"

OK
```

3.3.2. 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 have reached the server
```

```
                                  successfully. The host can query whether the data have reached
                                  the server with 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 of data have been sent, and the 9 bytes of data
                                  have 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
```

3.3.3. Receive Data from Remote Server in Buffer Access Mode

```
+QIURC: "recv",0          //<connectID> 0 has received data.
AT+QIRD=0,1500            //Retrieve the data. 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
```

3.3.4. Close a Connection

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

3.4. TCP Client Works in Transparent Transmission Mode

3.4.1. Set up a TCP Client Connection and Enter Transparent Transmission Mode

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

3.4.2. Send Data in Transparent Transmission Mode

```
<All data received from COM port will be sent directly to the Internet >
```

3.4.3. Retrieve Data from Remote Server in Transparent Transmission Mode

```
Test 1                     //All data received from the Internet are outputted via the COM port directly.
```

3.4.4. 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 link. Depending on the
                                network, the maximum response time is 10 seconds by default.
OK
```

3.5. TCP Client Works in Direct Push Mode

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

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

```
+QIOPEN: 0,0                  //TCP client connected successfully. It is suggested to wait for 150
                              seconds for the URC +QIOPEN: <connectID>,<result>. 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
```

3.5.2. 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 have reached the server successfully. The
                     host can query whether the data has reached the server with 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 of data have been sent, and all the 10 bytes of data have been
                     acknowledged.

OK
```

3.5.3. Retrieve Data from Remote Server in Direct Push Mode

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

3.5.4. Close a TCP Client

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

3.6. TCP Server Works in Buffer Access Mode under IPv4 Network

3.6.1. Start a TCP Server

```
//Context is 1 and <connectID> is 1. Before using AT+QIOPEN, the host should activate the context with
AT+QIACT.
AT+QIOPEN=1,1,"TCP LISTENER","127.0.0.1",0,2020,0
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
```

3.6.2. 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.
```

3.6.3. 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               //Length of actually retrieved data 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
```

3.6.4. Close a TCP Server

```
AT+QICLOSE=11                   //Close the incoming connection. Depending on the network,
                                the maximum response time is 10 seconds by default.
OK
AT+QICLOSE=1                    //Close the listening TCP server.
OK
```

3.7. TCP Server Works in Buffer Access Mode under IPv6 Network

3.7.1. Start a TCP Server

```
//Context is 1 and <connectID> is 1. Before using AT+QIOPEN, the host should activate the context with
AT+QIACT.
AT+QIOPEN=1,1,"TCP LISTENER","0:0:0:0:0:0:0:1",0,12345,0
OK

+QIOPEN: 1,0                    //TCP server is opened successfully.
AT+QISTATE=0,1                  //Query the connection status of context 1.
+QISTATE: 1,"TCP LISTENER","2001:468:2100:1:2001:468:2100:1",0,12345,3,1,1,0,"usbmodem"

OK
```

3.7.2. Accept TCP Incoming Connection

```
+QIURC: "incoming",11,1,"2001:468:2100:1:0:0:0:0",54091   //A new TCP connection is accepted.
                                                           The <service_type> is "TCP incoming",
                                                           and <connectID> is 11.
```

3.7.3. 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                        //Length of actually retrieved data 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
```

3.7.4. Switch Incoming Connection to Transparent Transmission Mode

```
AT+QSWTMD=11,2                        //Switch the incoming connection whose socket service
                                      index is 11 to transparent transmission mode.
CONNECT
1234567890                            //Send data directly.
+++                                   //Exit transparent transmission mode.
OK
```

3.7.5. Close a TCP Server

```
AT+QICLOSE=11                         //Close the incoming connection. Depending on the
                                      network, the maximum response time is 10 seconds by
                                      default.
OK
AT+QICLOSE=1                          //Close the listening TCP server.
OK
```

3.8. UDP Service under IPv4 Network

3.8.1. Start a UDP Service

```
//Start a UDP service whose <connectID> is 2 and <context> is 1. Before using AT+QIOPEN, the host
should activate the context with AT+QIACT.
AT+QIOPEN=1,2,"UDP SERVICE","127.0.0.1",0,3030,0
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
```

3.8.2. Send UDP Data to Remote Client

```
AT+QISEND=2,10,"10.7.89.10",6969   //Send 10 bytes of data to a remote client whose IP is
                                   10.7.89.10 and the remote port is 6969.
>1234567890
SEND OK
```

3.8.3. Retrieve Data from Remote Client

```
+QIURC: "recv",2                   //Data received from the remote client.
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 IP address of the
                                   remote client is 10.7.76.34 and the 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 client whose IP is 10.7.76.34 and the
                                   port is 7687.
>1234567890
SEND OK
```

3.8.4. Close a UDP Service

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

3.9. UDP Service under IPv6 Network

3.9.1. Start a UDP Service

```
//Start a UDP service whose <connectID> is 2 and <context> is 1. Before using AT+QIOPEN, the host
should activate the context with AT+QIACT.
AT+QIOPEN=1,2,"UDP SERVICE","0:0:0:0:0:0:0:1",0,12345,0
OK
```

```
+QIOPEN: 2,0                                 //UDP service is opened successfully.
AT+QISTATE=0,1                               //Query the connection status of context 1.
+QISTATE: 2,"UDP SERVICE","2001:468:2100:1:2001:468:2100:1",0,12345,2,1,2,0,"usbmodem"

OK
```

3.9.2. Send UDP Data to Remote Client

```
AT+QISEND=2,10,"2001:468:2100:1:0:0:0:0",5001   //Send 10 bytes of data to a remote client
                                                whose IP is 2001:468:2100:1:0:0:0:0 and the
                                                port is 5001.
>1234567890
SEND OK
```

3.9.3. Retrieve Data from Remote Client

```
+QIURC: "recv",2                             //Received data from the remote client.
AT+QIRD=2                                    //Retrieve UDP data. One whole UDP packet will be
                                             outputted. There is no need to specify the read length.
+QIRD: 4,"2001:468:2100:1:0:0:0:0",5001      //The retrieved data length is 4 bytes. The IP address of
                                             the remote client is 2001:468:2100:1:0:0:0:0 and the port
                                             is 5001.
AAAA

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

OK
```

3.9.4. Close a UDP Service

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

3.10. UDP Client with RAI Feature

3.10.1. Start a UDP Client

```
//Start a UDP client whose <connectID> is 0 and <context> is 1. Before using AT+QIOPEN, the host
should activate the context with AT+QIACT.
AT+QIOPEN=1,0,"UDP","220.180.239.212",8009
OK
```

3.10.2. Send UDP Data with RAI Information

```
AT+QCSCON=1                          //Enable RRC state reported via URC.
OK
AT+QISENDEX=0,"313233",0             //Send “123” without RAI information.
SEND OK

//18:48:14:649
+QCSCON: 1,1                         //RRC connection set up successfully.

//18:48:35:790
+QCSCON: 1,0                         //After 20 s, RRC connection is released.
AT+QISENDEX=0,"313233",1             //Send “123” and release RRC connection right away.
SEND OK

//18:51:59:905
+QCSCON: 1,1                         //RRC connection set up successfully.

//18:52:01:135
+QCSCON: 1,0                          //Within 1 s, RRC connection is released.
AT+QISENDEX=0,"313233",2             //Send “123” and release RRC connection after receiving a
                                     packet of downlink data.
SEND OK

//19:00:28:037
+QCSCON: 1,1                         //RRC connection set up successfully.

//19:00:35:115
+QIURC: "recv",0                     //Receive a packet of downlink data from peer.

//19:00:35:264
+QCSCON: 1,0                         //After receiving downlink data, RRC state became idle.
```

3.10.3. Close a UDP Client

```
AT+QICLOSE=0                                    //Close the client.
OK
```

3.11. PING

3.11.1. PING under IPv4

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

3.11.2. PING under IPv6

```
AT+QPING=1,"2001:468:2100:1:0:0:0:0"            //Ping 2001:468:2100:1:0:0:0:0 in context 1. Before
                                                pinging the destination IP address, the host should
                                                activate the context with AT+QIACT.
OK

+QPING: 0,"2001:468:2100:1:0:0:0:0",32,58,255

+QPING: 0,"2001:468:2100:1:0:0:0:0",32,49,255

+QPING: 0,"2001:468:2100:1:0:0:0:0",32,39,255

+QPING: 0,"2001:468:2100:1:0:0:0:0",32,40,255

+QPING: 0,4,4,0,39,58,46
```

3.12. Synchronize Local Time

```
AT+QNTP=1,"202.120.2.101",123

OK

+QNTP: 0,"2013/08/19,11:10:10+32"
AT+CCLK?
+CCLK: "13/08/19,11:10:57+32"

OK
```

//Synchronize local time with NTP server "202.120.2.101:123". Before synchronizing the time, the host should activate the context with **AT+QIACT**.

3.13. Getting the Last Error Code

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

ERROR
AT+QIGETERROR
+QIGETERROR: 552, invalid parameters

OK
```

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

4 Summary of Result Codes

If **<result>** is returned after executing TCP/IP AT commands, the detailed information about result codes can be queried with **AT+QIGETERROR**. Please note that **AT+QIGETERROR** just returns the result code of the last TCP/IP AT command.

Table 2: Summary of Result Codes

<result>	<result_code_description>
0	operate successfully
550	unknown error
551	operation blocked
552	invalid parameters
553	Memory allocation failed
554	create socket failed
555	operation not supported
556	socket bind failed
557	socket listen failed
558	socket write failed
559	socket read failed
560	socket accept failed
561	Activate pdp context failed
562	Deactivate pdp context failed
563	socket identity has been used
564	dns busy

565	dns failed
566	socket connect failed
567	connection reset
568	operation busy
569	operation timeout
570	pdp context deactivated
571	user cancel send
572	operation not allowed
573	port busy
574	pdp has been actived

5 Appendix References

Table 3: Related Documents

Document Name
[1] Quectel_BG95&BG77&BG600L_Series_AT_Commands_Manual
[2] Quectel_BG95&BG77&BG600L_Series_SSL_Application_Note

Table 4: Terms and Abbreviations

Abbreviation	Description
3GPP	3rd Generation Partnership Project
ACK	Acknowledge
APN	Access Point Name
CHAP	Challenge Handshake Authentication Protocol
DNS	Domain Name System
FIN	Finish
ID	Identify
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
LPWA	Low Power Wide Area
LTE	Long-Term Evolution
NB-IoT	Narrowband Internet of Things
NTP	Network Time Protocol

NVM	Non-Volatile Memory
PAP	Password Authentication Protocol l
PDP	Packet Data Protocol
PPP	Point-to-Point Protocol
SACK	Selective Acknowledgment
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver & Transmitter
UDP	User Datagram Protocol
URC	Unsolicited Result Code
USB	Universal Serial Bus
UTC	Coordinated Universal Time