

EC200x&EG912Y&EC600S Series

TCP/IP Application Note

LTE Standard Module Series

Rev. EC200x&EG912Y&EC600S_Series_TCP/IP_Application_Note_V1.0

Date: 2020-07-24

Status: Released



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

Quectel Wireless Solutions Co., Ltd.

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

Tel: +86 21 5108 6236 Email: info@quectel.com

Or our local office. For more information, please visit: <http://www.quectel.com/support/sales.htm>.

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

<http://www.quectel.com/support/technical.htm> or email to support@quectel.com.

GENERAL NOTES

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

DISCLAIMER

WHILE QUECTEL HAS MADE EFFORTS TO ENSURE THAT THE FUNCTIONS AND FEATURES UNDER DEVELOPMENT ARE FREE FROM ERRORS, IT IS POSSIBLE THAT THESE FUNCTIONS AND FEATURES COULD CONTAIN ERRORS, INACCURACIES AND OMISSIONS. UNLESS OTHERWISE PROVIDED BY VALID AGREEMENT, QUECTEL MAKES NO WARRANTIES OF ANY KIND, IMPLIED OR EXPRESS, WITH RESPECT TO THE USE OF FEATURES AND FUNCTIONS UNDER DEVELOPMENT. TO THE MAXIMUM EXTENT PERMITTED BY LAW, QUECTEL EXCLUDES ALL LIABILITY FOR ANY LOSS OR DAMAGE SUFFERED IN CONNECTION WITH THE USE OF THE FUNCTIONS AND FEATURES UNDER DEVELOPMENT, REGARDLESS OF WHETHER SUCH LOSS OR DAMAGE MAY HAVE BEEN FORESEEABLE.

COPYRIGHT

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

Copyright © Quectel Wireless Solutions Co., Ltd. 2020. All rights reserved.

About the Document

Revision History

Version	Date	Author	Description
1.0	2020-07-24	Larson LI	Initial

Contents

About the Document.....	2
Contents.....	3
Table Index.....	5
1 Introduction	6
1.1. The Process of Using TCP/IP AT Commands	6
1.2. Description of Data Access Modes.....	8
2 TCP/IP AT Commands	10
2.1. AT Command Syntax	10
2.1.1. Definitions.....	10
2.1.2. AT Command Syntax	10
2.2. AT Commands.....	11
2.2.1. AT+QICSGP Configure Parameters of a TCP/IP Context	11
2.2.2. AT+QIACT Activate a PDP Context	12
2.2.3. AT+QIDEACT Deactivate a PDP Context.....	13
2.2.4. AT+QIOPEN Open a Socket Service	14
2.2.5. AT+QICLOSE Close a Socket Service.....	16
2.2.6. AT+QISTATE Query Socket Service Status	16
2.2.7. AT+QISEND Send Data	18
2.2.8. AT+QIRD Read the Received TCP/IP Data	21
2.2.9. AT+QISENDEX Send Hex String Data.....	22
2.2.10. AT+QISWTMD Switch Data Access Mode.....	23
2.2.11. AT+QPING Ping a Remote Server	24
2.2.12. AT+QNTP Synchronize Local Time with NTP Server	25
2.2.13. AT+QIDNSCFG Configure Address of DNS Server	26
2.2.14. AT+QIDNSGIP Get IP Address by Domain Name.....	27
2.2.15. AT+QICFG Configure Optional Parameters	28
2.2.16. AT+QISDE Control Whether to Echo the Data for AT+QISEND.....	32
2.2.17. AT+QIGETERROR Query the Last Error Code.....	33
2.3. Description of URCs	33
2.3.1. +QIURC: "closed" URC Indicating Connection Closed.....	34
2.3.2. +QIURC: "recv" URC Indicating Incoming Data.....	34
2.3.3. +QIURC: "incoming full" URC Indicating Incoming Connection Full.....	35
2.3.4. +QIURC: "incoming" URC Indicating Incoming Connection	35
2.3.5. +QIURC: "pdpdeact" URC Indicating PDP Deactivation	35
3 Examples	37
3.1. Configure and Activate a Context	37
3.1.1. Configure a Context	37
3.1.2. Activate a Context	37
3.1.3. Deactivate a Context.....	37
3.2. TCP Client Works in Buffer Access Mode	37

3.2.1.	Set up a TCP Client Connection and Enter into Buffer Access Mode	37
3.2.2.	Send Data in Buffer Access Mode	38
3.2.3.	Receive Data from Remote Server in Buffer Access Mode.....	38
3.2.4.	Close a Connection.....	39
3.3.	TCP Client Works in Transparent Access Mode.....	39
3.3.1.	Set up a TCP Client Connection and Enter into Transparent Access Mode	39
3.3.2.	Send Data in Transparent Access Mode	39
3.3.3.	Receive Data from Remote Server in Transparent Access Mode	39
3.3.4.	Close a TCP Client.....	40
3.4.	TCP Client Works in Direct Push Mode.....	40
3.4.1.	Set up a TCP Client Connection and Enter into Direct Push Mode	40
3.4.2.	Send Data in Direct Push Mode.....	40
3.4.3.	Receive Data from Remote Server in Direct Push Mode	41
3.4.4.	Close a TCP Client.....	41
3.5.	TCP Server Works in Buffer Access Mode	41
3.5.1.	Start a TCP Server	41
3.5.2.	Accept TCP Incoming Connection	41
3.5.3.	Receive Data from Incoming Connection	42
3.5.4.	Close a TCP Server	42
3.6.	Example of UDP Service	42
3.6.1.	Start a UDP Service	42
3.6.2.	Send UDP Data to Remote	43
3.6.3.	Receive Data from Remote.....	43
3.6.4.	Close a UDP Service	43
3.7.	PING.....	43
3.8.	Synchronize Local Time.....	44
3.9.	Get Last Error Code.....	44
4	Summary of Error Codes	45
5	Appendix A Reference.....	47

Table Index

Table 1: Types of AT Commands and Responses	10
Table 2: Summary of Error Codes.....	45
Table 3: Terms and Abbreviations	47

1 Introduction

Quectel EC200x and EG912Y series modules and EC600S-CN modules feature embedded TCP/IP stack, which enables the host to access the Internet directly via AT commands. This greatly reduces the dependence on external PPP and TCP/IP protocol stacks and thus minimizes the costs.

EC200x and EG912Y series modules and EC600S-CN modules provide the following socket services: TCP client, UDP client, TCP server and UDP server.

This document is applicable to the following modules:

- EG912Y Series
- EC200T Series
- EC200S-CN
- EC600S-CN

1.1. The Process of Using TCP/IP AT Commands

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

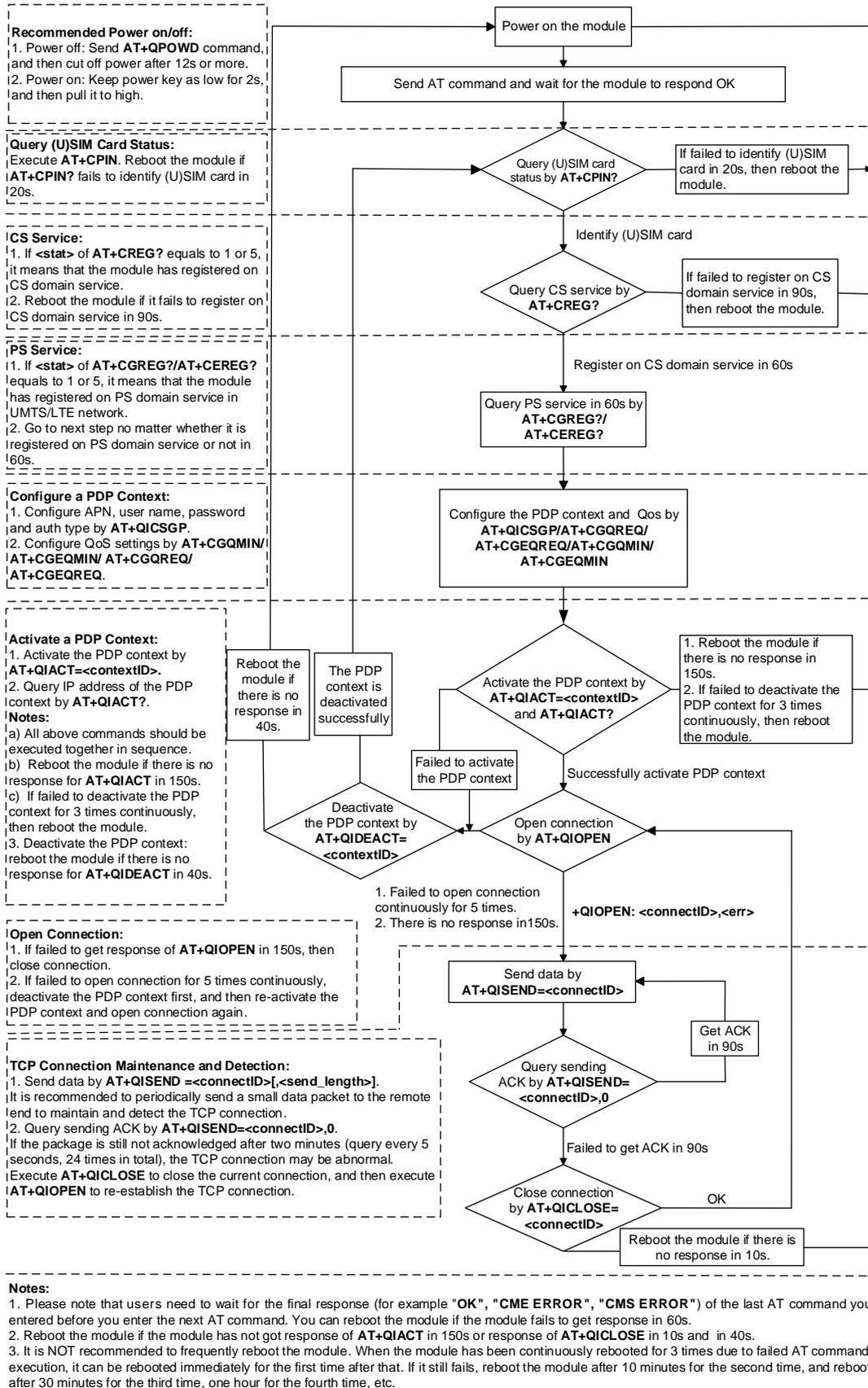


Figure 1: Flow Chart of Using TCP/IP AT Commands

1.2. Description of Data Access Modes

EC200x and EG912Y series modules and EC600S-CN modules support the following three kinds of data access modes:

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

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

1. In buffer access mode, data can be sent via **AT+QISEND**, and 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 via **AT+QIRD**.
2. In direct push mode, data can be sent via **AT+QISEND**, and if the module has received the data from the Internet, the data will be outputted to COM port directly in the following format:
+QIURC: "recv",<connectID>,<currentrecvlength><CR><LF><data>.
3. In transparent access mode, the corresponding port (such as UART, USB modem port, etc.) enters the exclusive mode. The data received from COM port will be sent to the Internet directly, and the data received from Internet will be outputted via COM port directly. **+++** can be used to exit from transparent access mode. When **OK** is returned, the module will be switched to buffer access mode. **AT+QISWTMD** can be used to switch the data access mode back to transparent access mode.

- **Exit from transparent access mode**

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

- 1) Do not input any character within 1 s or longer before inputting **+++**.
- 2) Input only **+++** within 1 s. Input no other characters during the time.
- 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 access mode, and wait until **OK** is returned.

- **Return to transparent access mode**

- 1) By **AT+QISWTMD**. Specify the **<access_mode>** as 2 when executing this command. When transparent access mode has been entered successfully, **CONNECT** will be returned.
- 2) By **ATO**. After a connection exits from transparent access mode, executing **ATO** will switch the data access mode back to transparent access mode again. When transparent access mode has

been entered successfully, **CONNECT** will be returned. If no connection has entered transparent access mode before, **ATO** will return **NO CARRIER**.

NOTES

1. In buffer access mode, if the buffer is not empty, the module will not report any new URC until all the received data has been read via **AT+QIRD** from the buffer.
2. In transparent access mode, AT commands cannot be executed. If the socket connection is closed because of network error or other errors, the module will report **NO CARRIER** and exit from the transparent access mode. In this case, **AT+QICLOSE** should be executed to close the socket service.

2 TCP/IP AT Commands

This chapter describes AT commands related to TCP/IP.

2.1. AT Command Syntax

2.1.1. Definitions

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

2.1.2. AT Command Syntax

The **AT** or **at** prefix must be added at the beginning of each command line. Entering **<CR>** will terminate a command line. Commands are usually followed by a response that includes **<CR><LF><response><CR><LF>**. Throughout this document, only the response **<response>** will be presented, **<CR><LF>** are omitted intentionally.

Table 1: Types of AT Commands and Responses

Test Command	AT+<cmd>=?	This command returns the list of parameters and value ranges set by the corresponding Write Command or in internal processes.
Read Command	AT+<cmd>?	This command returns the currently set value of the parameter or parameters.
Write Command	AT+<cmd>=<p1> [,<p2>[,<p3>[...]]]	This command sets the user-definable parameter values.
Execution Command	AT+<cmd>	This command reads non-variable parameters affected by internal processes in the module.

2.2. AT Commands

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

This command configures the **<APN>**, **<username>**, **<password>** and other parameters of a TCP/IP context. The QoS settings can be configured by **AT+CGQMIN**, **AT+CGEQMIN**, **AT+CGQREQ** and **AT+CGEQREQ**. For more details about the AT commands, refer to *Quectel_EC200T-CN_AT_Commands_Manual*.

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),(list of supported <CDMA_pwd> s) OK
Write Command Query the configuration of a specific context AT+QICSGP=<contextID>	Response +QICSGP: <context_type> , <APN> , <username> , <password> , <authentication> OK
Write Command Configure the context AT+QICSGP=<contextID>[,<context_type>,<APN>[,<username>,<password>][,<authentication>[,<CDMA_pwd>]]]	Response OK If there is any error: ERROR
Maximum Response Time	/
Characteristic	The command takes effect immediately. The configurations will not be saved.

Parameter

<contextID>	Integer type. The context ID. Range: 1–15.
<context_type>	Integer type. The protocol type. 1 IPv4 2 IPv6 3 IPv4v6
<APN>	String type. The access point name.
<username>	String type. The username. The maximum length: 127 bytes.

<password>	String type. The password. The maximum length: 127 bytes.
<authentication>	Integer type. The authentication methods. <u>0</u> None 1 PAP 2 CHAP
<CDMA_pwd>	Integer type. To configure whether to enable saving of <username> and <password> over CDMA network. <u>0</u> Disable 1 Enable

Example

```

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

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

2.2.2. AT+QIACT Activate a PDP Context

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

Although the range of **<contextID>** is 1–15, the module supports a maximum of three PDP contexts activated simultaneously. Depending on the network, it may take at most 150 seconds to return **OK** or **ERROR** after **AT+QIACT** is executed. Before the response is returned, other AT commands cannot be executed.

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_addresses> s> +QIACT: 15,<context_state>,<context_type>,<IP_addresses> s>

	OK
Write Command Activate a specified PDP context AT+QIACT=<contextID>	Response OK If there is any error: ERROR
Maximum Response Time	150 seconds, depending on the network.
Characteristic	/

Parameter

<contextID>	Integer type. The context ID. Range: 1–15.
<context_state>	Integer type. The context state. 0 Deactivated 1 Activated
<context_type>	Integer type. The protocol type. <u>1</u> IPv4 2 IPv6 3 IPv4v6
<IP_address>	String type. The local IP address after the context is activated.

2.2.3. 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 at most 40 seconds to return **OK** or **ERROR** after **AT+QIDEACT** is executed. Before the response is returned, other AT commands cannot be executed.

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 If there is any error: ERROR
Maximum Response Time	40 seconds, depending on the network.

Characteristic

/

Parameter

<contextID> Integer type. The context ID. Range: 1–15.

2.2.4. AT+QIOPEN Open a Socket Service

The 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 access mode) can be specified by **<access_mode>**. The URC **+QIOPEN** indicates whether the socket service has been opened successfully.

- If **<service_type>** is "TCP LISTENER", the module works as TCP server. After accepting a new TCP connection, the module will automatically specify a **<connectID>** and report a URC as **+QIURC: "incoming",<connectID>,<serverID>,<remoteIP>,<remote_port>**. The range of **<connectID>** is 0–11. The type of this new incoming connection is "TCP INCOMING" and the **<access_mode>** of "TCP INCOMING" is the same with that of "TCP LISTENER".
- If **<service_type>** is "UDP SERVICE", UDP data can be sent to or received from the remote IP via **<local_port>**.
 - Send data: execute **AT+QISEND=<connectID>,<send_length>,<remoteIP>,<remote_port>**.
 - Receive data in direct push mode: the module reports the URC as **+QIURC: "recv",<connectID>,<currentrecvlength>,<remoteIP>,<remote_port><CR><LF><data>**.
 - Receive data in buffer access mode: the module reports the URC as **+QIURC: "recv",<connectID>**, and then data can be retrieved via **AT+QIRD=<connectID>**.
- It is suggested to wait for 150 seconds for **+QIOPEN: <connectID>,<err>** to be outputted. If the URC cannot be received in 150 seconds after the Write Command is executed, **AT+QICLOSE** should be used 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>**",**<remote_ports>**,**<local_port>**,(range of supported **<access_mode>**s)

OK

Write Command

Response

AT+QIOPEN=<contextID>,<connectID>,<service_type>,"<IP_address>/<domain_name>",<remote_port>[,<local_port>[,<access_mode>]]	<p>If the service is in transparent access mode (<access_mode>=2) and is opened successfully: CONNECT</p> <p>If there is any error: ERROR Error description can be got via AT+QIGETERROR.</p> <p>If the service is in buffer access mode (<access_mode>=0) or direct push mode (<access_mode>=1): OK</p> <p>+QIOPEN: <connectID>,<err> <err> is 0 when the service is opened successfully. In other cases, <err> is not 0.</p>
Maximum Response Time	150 seconds, depending on the network.
Characteristic	/

Parameter

<contextID>	Integer type. Context ID. Range: 1–15.
<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
<IP_address>	String type. If <service_type> is "TCP" or "UDP", it indicates the IP address of remote server, such as 220.180.239.212. If <service_type> is "TCP LISTENER" or "UDP SERVICE", please enter 127.0.0.1.
<domain_name>	String type. The domain name address of the remote server.
<remote_port>	The port of the remote server. Range: 0–65535. It must be specified when <service_type> is "TCP" or "UDP".
<local_port>	The local port. Range: 0–65535. If <service_type> is "TCP LISTENER" or "UDP SERVICE", this parameter must be specified. If <service_type> is "TCP" or "UDP", the local port will be assigned automatically if <local_port> is 0. Otherwise the local port is assigned as specified.
<access_mode>	Integer type. The data access mode of the socket service. 0 Buffer access mode 1 Direct push mode

2 Transparent access mode

<err>

Integer type. Error codes of the operation. Please refer to **Chapter 4**.

2.2.5. AT+QICLOSE Close a Socket Service

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

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 AT+QICLOSE=<connectID>[,<timeout>]	Response If the socket service is closed successfully: OK If it is failed to close the socket service: ERROR
Maximum Response Time	10 s by default, depending on the setting of <timeout>.
Characteristic	/

Parameter

<connectID>	Integer type. The Socket ID. Range: 0–11.
<timeout>	Integer type. The timeout value for the response to be outputted. If the FIN ACK of the other peer is not received within <timeout>, the module will be forced to close the socket. Range: 0–65535. Default value: 10. Unit: second.

2.2.6. AT+QISTATE Query Socket Service Status

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

AT+QISTATE Query Socket Service Status

Test Command 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	/
Characteristic	/

Parameter

<query_type>	Integer type. The query type. 0 Query connection status of all socket services in a specified context 1 Query connection status of a specified socket service
<contextID>	Integer type. The context ID. Range: 1–15.
<connectID>	Integer type. The 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 "UDP INCOMING" Start a UDP connection accepted by a UDP server
<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" or "UDP INCOMING", it is the IP address of remote client.
<remote_port>	Integer type. Remote port number.
	If <service_type> ="TCP" or "UDP", it is the port of remote server.
	If <service_type> ="TCP LISTENER" or "UDP SERVICE", the port is invalid.
	If <service_type> ="TCP INCOMING" or "UDP INCOMING", it is the port of remote client.
<local_port>	Integer type. Local port number.
	If <local_port> =0, then the local port is assigned automatically.
<socket_state>	Integer type. The socket service status.
	0 "Initial": connection has not been established
	1 "Opening": client is connecting or server is trying to listen
	2 "Connected": client/incoming connection has been established
	3 "Listening": server is listening
	4 "Closing": connection is closing
<serverID>	Integer type. It is valid only when <service_type> is "TCP INCOMING". <serverID> represents which server accepts this TCP incoming connection, and the value is the same as <connectID> of this server's "TCP LISTENER".
<access_mode>	Integer type. Data access mode.
	0 Buffer access mode
	1 Direct push mode
	2 Transparent access mode
<AT_port>	String type. COM port of socket service.
	"usbmodem" USB modem port
	"usbat" USB AT port
	"uart1" UART port1
	"cmux1" MUX port 1
	"cmux2" MUX port 2
	"cmux3" MUX port 3
	"cmux4" MUX port 4

2.2.7. AT+QISEND Send Data

If the data access mode of a specified socket service is buffer access mode (**<access_mode>**=0) or direct push mode (**<access_mode>**=1), then the data can be sent via this command. When the data is sent to the module successfully, **SEND OK** will be returned. Otherwise it will return **SEND FAIL** or **ERROR**. **SEND FAIL** indicates the sending buffer is full and resending of the data can be tried. **ERROR** indicates an error arises in the process of sending data. Resending of the data should be delayed for some time. The maximum length of data to be sent 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 **AT+QISEND=<connectID>,0**.

AT+QISEND Send Data	
<p>Test Command AT+QISEND=?</p>	<p>Response +QISEND: (range of supported <connectID>s),(range of supported <send_length>s)</p> <p>OK</p>
<p>Write Command Send variable-length data when <service_type> is "TCP", "UDP" or "TCP INCOMING" AT+QISEND=<connectID></p>	<p>Response ></p> <p>After the response >, input the data to be sent. Tap Ctrl + Z to send, and tap Esc to cancel the operation</p> <p>If the connection has been established and the data is sent successfully: SEND OK</p> <p>If the connection has been established but the sending buffer is full: SEND FAIL</p> <p>If the connection has not been established, abnormally closed, or any parameter is incorrect: ERROR</p>
<p>Write Command Send fixed-length data when <service_type> is "TCP", "UDP" or "TCP INCOMING" AT+QISEND=<connectID>,<send_length></p>	<p>Response ></p> <p>After the response >, input the data until the data length equals to <send_length>.</p> <p>If the connection has been established and the data is sent successfully: SEND OK</p> <p>If the connection has been established but the sending buffer is full: SEND FAIL</p> <p>If the connection has not been established, abnormally closed, or any parameter is incorrect: ERROR</p>
<p>Write Command If <service_type> is "UDP SERVICE" AT+QISEND=<connectID>,<send_length>,<remoteIP>,<remote_port></p>	<p>Response</p> <p>This command is used to send fixed length data to a specified remote IP address and remote port. The <service_type> must be "UDP SERVICE".</p>

	<p>></p> <p>After the response >, input the data until the data length equals to <send_length></p> <p>If the connection has been established and the data is sent successfully: SEND OK</p> <p>If the connection has been established but the sending buffer is full: SEND FAIL</p> <p>If the connection has not been established, abnormally closed, or any parameter is incorrect: ERROR</p>
<p>Write Command</p> <p>When <send_length> is 0, query the sent data</p> <p>AT+QISEND=<connectID>,0</p>	<p>Response</p> <p>If the specified connection exists: +QISEND: <total_send_length>,<ackedbytes>,<unacked bytes></p> <p>OK</p> <p>If there is any error: ERROR</p>
Maximum Response Time	/
Characteristic	/

Parameter

<connectID>	Integer type. Socket ID. Range: 0–11.
<send_length>	Integer type. The length of data to be sent, which cannot exceed 1460 bytes.
<remoteIP>	String type. The remote IP address (must be dot format). It is valid only when <service_type> is "UDP SERVICE".
<remote_port>	Integer type. Remote port. It is only valid when <service_type> is "UDP SERVICE".
<total_send_length>	Integer type. The total length of sent data. Unit: byte.
<ackedbytes>	Integer type. The total length of acknowledged data. Unit: byte.
<unackedbytes>	Integer type. The total length of unacknowledged data. Unit: byte.
<err>	Integer type. Error codes of the operation. Please refer to Chapter 4 .

2.2.8. AT+QIRD Read the Received TCP/IP Data

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

Please note that if the buffer is not empty, and the module receives data again, it will not report a new URC until all the received data has been read via **AT+QIRD** from buffer.

AT+QIRD Read 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", "TCP INCOMING" or "UDP INCOMING" AT+QIRD=<connectID>[,<read_length>]	Response If the specified connection has received the data, response: +QIRD: <read_actual_length> <data> OK If there is 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 exists: +QIRD: <read_actual_length>,<remoteIP>,<remote_port> <data> OK If there is no data: +QIRD: 0 OK If the connection does not exist: ERROR
Write Command	Response

When <read_length> is 0, query the retrieved data length AT+QIRD=<connectID>,0	If the specified connection exists: +QIRD: <total_receive_length>,<have_read_length>,<unread_length> OK If there is any error: ERROR
Maximum Response Time	/
Characteristic	/

Parameter

<connectID>	Integer type. The Socket ID. Range: 0–11.
<read_length>	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> is "UDP SERVICE".
<remote_port>	Integer type. Remote port. It is valid only when <service_type> is "UDP SERVICE".
<data>	The data that has been read.
<total_receive_length>	Integer type. The total length of the read 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.

2.2.9. 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> OK
Write Command AT+QISENDEX=<connectID>,<hex_string>	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	/
Characteristic	/

Parameter

<connectID>	Integer type. The Socket ID. Range: 0–11.
<hex_string>	String type. Hex string data. The max length is 512 bytes.

2.2.10. AT+QISWTMD Switch Data Access Mode

This command switches the data access modes between buffer access mode, direct push mode and transparent access mode. When a socket service is started, the data access mode can be specified via the **<access_mode>** of **AT+QIOPEN**. After a socket has been opened, the data access mode can be changed via **AT+QISWTMD**.

AT+QISWTMD Switch Data Access Mode

Test Command 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 data access mode is switched successfully and <access_mode> is 0 or 1: OK If data access mode is switched successfully and <access_mode> is 2, the module will enter into data mode: CONNECT If there is any error: ERROR
Maximum Response Time	/
Characteristic	The command takes effect immediately. The configuration is not saved.

Parameter

<connectID>	Integer type. The Socket ID. Range: 0–11.
<access_mode>	Integer type. The data access modes of the connection. 0 Buffer access mode 1 Direct push mode 2 Transparent access mode

2.2.11. AT+QPING Ping a Remote Server

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

AT+QPING Ping a Remote Server

Test Command 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 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
Maximum Response Time	/
Characteristic	/

Parameter

<contextID>	Integer type. The context ID. Range: 1–15.
<host>	String type. The host address in string type. The format is a domain name or a dotted decimal IP address.
<timeout>	Integer type. Set the maximum time to wait for the response of each ping request. Range: 1–255. Default value: 4. Unit: second.

<pingnum>	Integer type. Set the maximum number of times for sending ping request. Range: 1–10. Default value: 4.
<result>	The result of each ping request. 0 Received the ping response from the server. In this case, it is followed by <IP_address> , <bytes> , <time> , <tll> . Others Please refer to Chapter 4 .
<IP_address>	String type. The IP address of the remote server formatted as a dotted decimal IP.
<bytes>	Integer type. The length of each sent ping request. Unit: byte.
<time>	Integer type. The time wait for the response of the ping request. Unit: ms.
<tll>	Integer type. Time to live value of the response packet for 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, it is followed by <sent> , <rcvd> , <lost> , <min> , <max> , <avg> . Others Error codes. Please refer to Chapter 4 for details.
<sent>	Integer type. Total number of the ping requests that have been sent.
<rcvd>	Integer type. Total number of the ping requests that are returned with the response.
<lost>	Integer type. Total number of the ping requests that time 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.

2.2.12. AT+QNTIP Synchronize Local Time with NTP Server

This command synchronizes the local time with Universal Time Coordinated (UTC) through the NTP server. Before time synchronization, the host should activate the context corresponding to **<contextID>** via **AT+QIACT** first. Depending on the network, it will take at most 125 seconds to return the result.

AT+QNTIP Synchronize Local Time with NTP Server	
Test command AT+QNTIP=?	Response +QNTIP: (range of supported <contextID> s), <server> ,(range of supported <port> s),(list of supported <autosettime> s) OK
Read command AT+QNTIP?	Response If in the process of local time synchronization: +QNTIP: <server> , <port> OK
Write command AT+QNTIP=<contextID> , <server> [,<port>[,<autosettime>]]	Response If the local time is synchronized with NTP server successfully: OK +QNTIP: <err> , <time>

	If there is any error: ERROR
Maximum Response Time	125 seconds, depending on the network.
Characteristic	The command takes effect immediately. The configuration is not be saved.

Parameter

<contextID>	Integer type. The context ID. Range: 1–15.
<server>	String type. The address of NTP server.
<port>	Integer type. The port of NTP server. Range: 1–65535.
<autosettime>	Integer type. Whether to automatically set synchronized time as local time. 0 Not set <u>1</u> Set
<err>	Integer type. Error codes of the operation. Please refer to Chapter 4 for details.
<time>	String type. The time synchronized from NTP server. The format is "YYYY/MM/DD, hh:mm:ss±zz". Range of zz: -48 to +56.

2.2.13. AT+QIDNSCFG Configure Address of DNS Server

Before setting the DNS address, the host must activate the context corresponding to **<contextID>** via **AT+QIACT** first.

AT+QIDNSCFG Configure Address of DNS Server	
Test command AT+QIDNSCFG=?	Response +QIDNSCFG: (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 optional parameters is specified, configure the primary and secondary DNS server addresses of a specified PDP context: OK

	If there is any error: ERROR
Maximum Response Time	/
Characteristic	The command takes effect immediately. The configuration is not saved.

Parameter

<contextID>	Integer type. The PDP context ID. Range: 1–15.
<pridnsaddr>	String type. The primary DNS server address.
<secdnsaddr>	String type. The secondary DNS server address.

2.2.14. AT+QIDNSGIP Get IP Address by Domain Name

Before querying the DNS, the host should activate the context corresponding to <contextID> via **AT+QIACT** first. Depending on the network, it will take at most 60 seconds to return the result.

AT+QIDNSGIP Get IP Address by Domain Name	
Test Command AT+QIDNSGIP=?	Response +QIDNSGIP: (range of supported <contextID>s),<hostname> OK
Write Command AT+QIDNSGIP=<contextID>,<hostname>	Response OK If there is any error: ERROR The result will be returned as URC. +QIURC: "dnsgip",<err>,<IP_count>,<DNS_ttl> [..... +QIURC: "dnsgip",<hostIPaddr>
Maximum Response Time	60 seconds, depending on the network.
Characteristic	/

Parameter

<contextID>	Integer type. The PDP context ID. Range: 1–15.
<hostname>	String type. The domain name.
<err>	Integer type. Error codes of the operation. Please refer to Chapter 4 .
<IP_count>	Integer type. The number of the IP addresses corresponding to the <hostname>.

<DNS_ttl>	Integer type. The time-to-live of the DNS.
<hostIPAddr>	String type. The IP address of <hostname>.

2.2.15. AT+QICFG Configure Optional Parameters

This command configures optional parameters.

AT+QICFG Configure Optional Parameters	
<p>Test Command</p> <p>AT+QICFG=?</p>	<p>Response</p> <p>+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: "passiveclosed", (list of supported <closed>s) +QICFG: "udp/readmode", (list of supported <mode>s) +QICFG: "udp/sendmode", (list of supported <mode>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 <show_length>s) +QICFG: "tcp/accept", (list of supported <state>s)</p> <p>OK</p>
<p>Write Command</p> <p>Configure the packet size for transparent access mode</p> <p>AT+QICFG="transpktsize"[,<transpktsize>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query the current setting: +QICFG: "transpktsize",<transpktsize></p> <p>OK</p> <p>If the optional parameter is specified, configure the length of the data to be sent: OK</p> <p>If there is any error: ERROR</p>
<p>Write Command</p> <p>Configure the waiting time for transparent access mode</p> <p>AT+QICFG="transwaittm"[,<transwaittm>]</p>	<p>Response</p> <p>If the optional parameter is omitted, query the current setting: +QICFG: "transwaittm",<transwaittm></p> <p>OK</p> <p>If the optional parameter is specified, configure the waiting time: OK</p>

	<p>If there is any error: ERROR</p>
<p>Write Command Configure the format of the data to be sent or received (only for non-transparent mode) AT+QICFG="dataformat",<send_data_format>,<recv_data_format>]</p>	<p>Response If the optional parameters are omitted, query the current setting: +QICFG: "dataformat",<send_data_format>,<recv_data_format> at> OK If the optional parameters are entered, configure the format of the data to be sent or received: Ok If there is any error: ERROR</p>
<p>Write Command Configure the output format of received data (only for non-transparent mode) AT+QICFG="viewmode",<view_mode>]</p>	<p>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</p>
<p>Write Command Enable or disable the passive close of TCP connection when the server is closed AT+QICFG="passiveclosed",<closed>]</p>	<p>Response If the optional parameter is omitted, query the current setting: +QICFG: "passiveclosed",<closed> OK If the optional parameter is specified, enable or disable the passive close of TCP connection: OK If there is any error: ERROR</p>
<p>Write Command Configure the mode when reading UDP data AT+QICFG="udp/readmode",<mode>]</p>	<p>Response If the optional parameter is omitted, query the current setting: +QICFG: "udp/readmode",<mode> OK</p>

	<p>If the optional parameter is specified, configure the UDP data read mode: OK</p> <p>If there is any error: ERROR</p>
<p>Write Command Configure the mode when sending UDP data AT+QICFG="udp/sendmode",<mode>]</p>	<p>Response If the optional parameter is omitted, query the current setting: +QICFG: "udp/sendmode",<mode></p> <p>OK</p> <p>If the optional parameter is specified, configure the UDP data sending mode: OK</p> <p>If there is any error: ERROR</p>
<p>Write Command Configure whether to send TCP keep-alive information. AT+QICFG="tcp/keepalive",<enable>,<idle_time>,<interval_time>,<probe_cnt>]</p>	<p>Response If the optional parameters are omitted, query the current configurations +QICFG: "tcp/keepalive",<enable>,<idle_time>,<interval_time>,<probe_cnt></p> <p>OK</p> <p>If the optional parameters are entered, configure whether to send TCP keep-alive information: OK</p> <p>If there is any error: ERROR</p>
<p>Write Command Enable or disable data length presence in URC reported after data reception in TCP/IP buffer mode AT+QICFG="recvind",<show_length>]</p>	<p>Response If the optional parameter is omitted, query the current setting: [+QICFG: "recvind",<show_length>]</p> <p>OK</p> <p>If the optional parameter is specified, enable or disable data length presence in URC reported after data reception in TCP/IP buffer mode: OK</p>

	If there is any error: ERROR
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
Maximum Response Time	/
Characteristic	The command takes effect immediately. The configuration is not saved.

Parameter

<transpktsize>	Integer type. The max length of the data packet to be sent. Range: 1–1460. Default value: 1024. Unit: byte.
<transwaittm>	Integer type. In transparent access mode, if the length of data received from the port is less than the specified value of <transpktsize> , after exceeding the time of <transwaittm> , data will be sent directly. 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 as the module will automatically form two bytes to one ASCII code. <u>0</u> 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 as the module will automatically form two bytes to one ASCII code. <u>0</u> Text mode 1 Hex mode
<view_mode>	Integer type. The output mode of the received data. <u>0</u> data header\r\n\data. 1 data header,data.
<mode>	Integer type. <u>0</u> Disable block mode. 1 Enable stream mode.
<state>	Integer type.

	0 Disable auto accepting incoming TCP connection from the client. 1 Enable auto accepting incoming TCP connection from the client.
<enable>	Integer type. Enable or disable sending of TCP keep-alive information. 0 Disable 1 Enable
<show_length>	Integer type. Enable or disable data length presence in URC reported after data reception in TCP/IP buffer mode. 0 Disable 1 Enable
<idle_time>	Integer type. Indicates the cycle time of keepalive triggered. Range: 1–1800. Unit: second.
<interval_time>	Integer type. Indicates the interval of send a packet in a cycle time. Range: 25–100. Unit: second.
<probe_cnt>	Integer type. The count of packet transmission in a cycle time. Range: 3–10.
<closed>	Integer type. Enable or disable the passive close of TCP connection when the server is closed. 0 Disable 1 Enable

2.2.16. 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 If there is any error: ERROR
Maximum Response Time	/
Characteristic	The command takes effect immediately. The configuration is not saved.

Parameter

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

2.2.17. AT+QIGETERROR Query the Last Error Code

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

AT+QIGETERROR Query the Last Error Code

Test command AT+QIGETERROR=?	Response OK
Execution Command AT+QIGETERROR	Response +QIGETERROR: <err>,<errcode_description> OK
Maximum Response Time	/
Characteristic	/

Parameter

<err>	Integer type. Error codes of the operation. Please refer to Chapter 4 .
<errcode_description>	A string parameter indicates the details of error information. Please refer to Chapter 4 for details.

2.3. Description of URCs

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

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

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

+QIURC: "closed" URC Indicating Connection Closed

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

Parameter

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

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

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

In buffer access mode, after receiving data, the module will report URC as **+QIURC: "recv",<connectID>** to notify the host. Then host can retrieve data via **AT+QIRD**. Please note that if the buffer is not empty, and the module receives data again, it will not report a new URC until all the received data has been retrieved via **AT+QIRD** from buffer.

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

+QIURC: "recv" URC Indicating Incoming Data

+QIURC: "recv",<connectID>	The URC indicates the incoming data in buffer access mode. The host can receive data via AT+QIRD .
+QIURC: "recv",<connectID>,<currentrecvlength><CR><LF><data>	The URC indicates the incoming data in direct push mode when the <service_type> is "TCP", "UDP", "UDP INCOMING" or "TCP INCOMING".
+QIURC: "recv",<connectID>,<currentrecvlength>,<remoteIP>,<remote_port><CR><LF><data>	The URC indicates data incoming in direct push mode when <service_type> is "UDP SERVICE".

Parameter

<connectID>	Integer type. The Socket ID. Range: 0–11.
<currentrecvlength>	Integer type. The length of actually received data.
<remoteIP>	String type. Remote IP address.
<remote_port>	Integer type. Remote port.

<data> The received data.

2.3.3. +QIURC: "incoming full" URC Indicating Incoming Connection Full

If the incoming connection reaches the limit, or no socket system resources can be allocated, then the module will report the URC as **+QIURC: "incoming full"** for the new incoming connection request.

+QIURC: "incoming full" URC Indicating Incoming Connection Full

+QIURC: "incoming full" The URC indicates the incoming connection is full.

2.3.4. +QIURC: "incoming" URC Indicating Incoming Connection

If the <service_type> is "TCP LISTENER", when a remote client connects to this server, the host will automatically assign a free <connectID> for the new connection. The range of <connectID> is 0–11. In such a 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 Connection

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

Parameter

<connectID> Integer type. Assign this socket service for the incoming connection, which is automatically specified by the module. Range: 0–11.

<serverID> Integer type. The incoming <connectID> accepted by the server whose <service_type> is "TCP LISTENER" and listening socket ID is <serverID>.

<remoteIP> String type. Remote IP address of the incoming <connectID>.

<remote_port> Integer type. Remote port of the incoming <connectID>.

2.3.5. +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 a case, the host must execute **AT+QIDEACT** to deactivate the context and reset all connections.

+QIURC: "pdpdeact" URC Indicating PDP Deactivation

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

Parameter

<contextID> Integer type. The context ID. Range: 1–15.

3 Examples

3.1. Configure and Activate a Context

3.1.1. Configure a Context

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

3.1.2. Activate a Context

```
AT+QIACT=1 //Activate context 1. Depending on the network, the maximum  
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 into Buffer Access Mode

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

```
+QIOPEN: 0,0 //TCP client is connected successfully. It is
                suggested to wait for 150 seconds for the
                URC response as +QIOPEN:
                <connectID>,<err>. If the URC response
                cannot be received in 150 seconds, the host
                could use AT+QICLOSE to close the socket.
                //Query the connection status of socket
                service 1.

AT+QISTATE=1,0
+QISTATE: 0,"TCP","220.180.239.212",8009,65514,2,1,0,0,"usbmodem"

OK
```

3.2.2. Send Data in Buffer Access Mode

```
AT+QISEND=0 //Send variable-length data.
> test1<ctrl+Z>
SEND OK //SEND OK does not mean the data has been sent to the server
        successfully. The host can query whether the data has reached the
        server via AT+QISEND=0,0.

AT+QISEND=0,4 //Send fixed-length data and the data length is 4 bytes.
> test
SEND OK

AT+QISEND=0,0 //Query the length of sent data.
+QISEND: 9,9,0

OK

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

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

OK
```

3.2.3. Receive Data from Remote Server in Buffer Access Mode

```
+QIURC: "recv",0 //The received data when <connectID>=0.
AT+QIRD=0,1500 //Read data, the maximum length of data to be retrieved is 1500 bytes.
+QIRD: 14 //The length of actually received data is 14 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: 14,14,0

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 seconds.

OK
```

3.3. TCP Client Works in Transparent Access Mode

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

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

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

3.3.2. Send Data in Transparent Access Mode

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

3.3.3. Receive Data from Remote Server in Transparent Access Mode

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

3.3.4. Close a TCP Client

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

3.4. TCP Client Works in Direct Push Mode

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

```
AT+QIOPEN=1,0,"TCP","220.180.239.212",8009,0,1 //Context is 1 and <connectID> is 0. Before using AT+QIOPEN, the host should activate the context via AT+QIACT first.  
  
OK  
  
+QIOPEN: 0,0 //TCP client is connected successfully. It is suggested to wait for 150 seconds for the URC response as +QIOPEN: <connectID>,<err>. If the URC response cannot be received in 150 seconds, the host could use AT+QICLOSE to close the socket.  
  
AT+QISTATE=1,0 //Query if the connection state of <connectID> is 0.  
  
+QISTATE: 0,"TCP","220.180.239.212",8009,65344,2,1,0,1,"usbmodem"  
  
OK
```

3.4.2. Send Data in Direct Push Mode

```
AT+QISEND=0 //Send variable-length data.  
> test1<ctrl+Z>  
SEND OK //SEND OK does not mean the data has been sent to the server successfully. 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 lengths 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  
test
```

```
//Receive data from remote server.
```

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 seconds.
```

```
OK
```

3.5. TCP Server Works in Buffer Access Mode

3.5.1. Start a TCP Server

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

```
OK
```

```
+QIOPEN: 1,0
```

```
//TCP server is opened successfully.
```

```
AT+QISTATE=0,1
```

```
//Query whether the connection state of  
<contextID> is 1.
```

```
+QISTATE: 1,"TCP LISTENER","10.7.157.1",0,2020,3,1,1,0,"usbmodem"
```

```
OK
```

3.5.2. Accept TCP Incoming Connection

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

3.5.3. Receive Data from Incoming Connection

```
+QIURC: "recv",11 //Receive data from remote incoming connection.
AT+QIRD=11,1500 //Read data received from incoming connection.
+QIRD: 4 //Actual data length is 4 bytes.
test

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

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

OK
```

3.5.4. Close a TCP Server

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

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 opened successfully.
AT+QISTATE=0,1 //Query if the connection status of
<contextID> is 1.

+QISTATE: 2,"UDP SERVICE","10.7.157.1",0,3030,2,1,2,0,"usbmodem"
```

OK

3.6.2. Send UDP Data to Remote

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

3.6.3. Receive Data from Remote

```
+QIURC: "recv",2 //Receive data from remote.
AT+QIRD=2 //Read UDP data. One whole UDP packet will be outputted.
           There is no need to specify the read length.
+QIRD: 4,"10.7.76.34",7687 //Data length is 4. The remote IP address is 10.7.76.34 and
                           remote port is 7687.
AAAA
OK
AT+QIRD=2 //Read data.
+QIRD: 0 //No data in buffer.
OK
AT+QISEND=2,10,"10.7.76.34",7687 //Send data to 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
                                   destination IP address, the host should activate the context by
                                   AT+QIACT first.
```

OK

+QPING: 0,"61.135.169.125",32,192,255

+QPING: 0,"61.135.169.125",32,240,255

+QPING: 0,"61.135.169.125",32,241,255

+QPING: 0,"61.135.169.125",32,479,255

+QPING: 0,4,4,0,192,479,287

3.8. Synchronize Local Time

AT+QNTP=1,"202.112.10.36",123

//Synchronize local time with NTP server "202.112.10.36:123". Before synchronizing the time, the host should activate the context with **AT+QIACT** first.

OK

+QNTP: 0,"2019/07/21,06:10:59+00"

AT+CCLK?

+CCLK: "19/07/21,06:11:05+00"

OK

3.9. Get Last Error Code

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

//Start 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 TCP/IP AT commands are executed, the details of **ERROR** can be queried via **AT+QIGETERROR**. Please note that **AT+QIGETERROR** just returns error code of the last TCP/IP AT command.

Table 2: Summary of Error Codes

<err>	Meaning
0	Operation success
550	Unknown error
551	Operation blocked
552	Invalid parameters
553	Memory not enough
554	Socket creation 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	PDP Context opening failed
562	PDP context closure failed
563	Socket identity has been used
564	DNS busy

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

5 Appendix A Reference

Table 3: 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 I
PDP	Packet Data Protocol
PPP	Point-to-Point Protocol
PS	Packet Switching
QoS	Quality of Service
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol

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
