GSM TCP/IP

Recommended Process

GSM/GPRS Module Series

Rev. GSM_TCP/IP_Recommended_Process_V2.0

Date: 2018-11-09

Status: Released

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

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

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

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2011-03-14	Colin HU	Initial
1.1	2012-07-05	Jean HU	Added flow chart of TCP/IP recommended process
1.2	2012-04-08	Chris PENG	Added applicable modules
2.0	2018-11-09	Oven TAO	Numerous changes were made to this document, and it should be read in its entirety.

Contents

Table Index

1 Introduction

This document introduces the simple process on how to use the embedded TCP/IP services and gives some recommendations for handling the abnormalities.

This document is applicable to all Quectel GSM modules.

2 Initialization

AT+IPR=115200&W	//Set fixed band rate as 115200bps.
OK	
AT+CPIN?	//Make sure the (U)SIM PIN is unlocked.
+CPIN: READY	//This indicates the (U)SIM PIN is unlocked and ready.
OK	
AT+CREG?	//Make sure the network is registered on CS service successfully.
+CREG: 0,1	//This indicates the network is registered on CS service successfully. Continue to execute command **AT+CREG?** if the registration fails. The network registration status can also be queried by **AT+CREG=1** when the module boots; and then **+CREG: 1** or **+CREG: 5** will be reported when the network registration status changes.
OK	
AT+CGATT?	//Query whether GPRS is attached to network successfully.
+CGATT: 1	//This indicates the GPRS is attached to network successfully. Continue to execute command **AT+CGATT?** if the attachment fails. The GPRS attachment can also be queried by **AT+CGREG=1** when the module boots; and then **+CGREG: 1** or **+CGREG: 5** will be reported when the GPRS attachment status changes.
OK	
AT+QIFGCNT=0	//Set the Context 0 as the foreground context. Since then the relevant operation is carried out for Context 0.
OK	
AT+QICSGP=1,"CMNET"	//Choose GPRS mode and set APN as "CMNET".
OK	

Other optional configurations for initialization are listed as below:

1. **Set address format of the server**. By default, the module considers the server address as an IP address. If the server address is a domain name, then execute **AT+QIDNSIP=1** to change the server address format to the domain name format.

2. **Set the mode to handle the received data**. Currently the module supports two modes to handle the received TCP/IP data. One is to output all the received data through UART directly (default), and the other is to output a notification **+QIRD:** instead of outputting the data at once after receiving it, then

AT+QIRD needs to be executed to retrieve the data. If the second mode is used, then the following command needs to be executed.

AT+QINDI=1	//After receiving the data, output the notification **+QIRDI: <id>,<sc>,<sid>**. Then retrieve data by command **AT+QIRD=<id>,<sc>,<sid>,<len>**
OK	

3. **Set the display format of received data.** When UART is selected to output the received data directly, the following commands can be used to set the display format of the received data and can be selected according to actual needs.

AT+QIHEAD=1	//Add head information "**IPD<len>**:" before the received data
OK	

AT+QISHOWRA=1	//Add the address and port number of the data source at the location of the received data head, and the detailed format of the added information is: **RECV FROM:<IP ADDRESS>:<PORT>**
OK	

AT+QISHOWPT=1	//Add the protocol type TCP or UDP of transmission layer before the received data. This application is rare.
OK	

4. **Set transparent mode**. The default mode is non-transparent mode. The methods for establishing TCP connection, sending and receiving data in this mode are introduced in ***Chapter 3***~ ***Chapter 7***. If transparent mode is required to be used, then the following command needs to be executed. For the specific application of the transparent mode, please refer to ***Chapter 8***.

AT+QIMODE=1	//Set transparent mode
OK	
AT+QITCFG=3,2,512,1	//Please pay attention to the middle two parameters “2” and “512”. The parameter "2" means that the module will wait for 200ms after it receives the data inputted from UART (if the length of all data inputted is less than 512 bytes), then send all the data that has been inputted. Parameter "512" means that when the length of the data received by the module from the serial port exceeds 512 bytes, the data will be sent as a 512-byte data immediately until the data length in the buffer is less than 512 bytes.
OK	

3 Establish TCP/UDP Connection

3.1. Establish TCP Connection

AT+QIOPEN=“TCP”,“220.180.239.212”,“8063”	//Connect the module to a remote TCP server whose address is 220.180.239.212:8063.
OK	//The syntax of the commands are right and TCP connection can be established in the current status of the module.

Analysis and treatment of error responses:

1. **ERROR**. There may be two reasons for the response:

   1) The command format is wrong. If the formats of all the commands are right, verify whether multiple TCP/IP sessions at the same time is disabled by command **AT+QIMUX?**. If it is disabled (**+QIMUX: 1** is returned for the read command), enable it by command **AT+QIMUX=0**.

   2) The current TCP/IP connection status is not "IP INITIAL", "IP STATUS" or "IP CLOSE" (query by command **AT+QISTAT**). If the current status is "TCP CONNECTING", execute command **AT+QICLOSE** to close the current failed TCP connection. Otherwise, execute command **AT+QIDEACT** to deactivate the current failed GPRS context.

2. **ALREADY CONNECT**. This means a TCP or UDP connection has been established. If new connection is required, execute command **AT+QICLOSE** to close the current connection.

CONNECT OK	//Successfully connected to the remote TCP server.

3. **CONNECT FAIL**. This means the TCP connection is failed to be established. The correct processing method is to send command **AT+QISTAT** to query the current TCP connection status at first.

   - If the current status is "TCP CONNECTING", it is recommended to close the failed connection by command **AT+QICLOSE**. In this way, GPRS is still active and there is no need to restart the GPRS context, so the program speed can be improved.
   - If the current status is not "TCP CONNECTING", it is recommended to execute command **AT+QIDEACT** to deactivate GPRS context because these status are usually caused by the failure of GPRS context activation.

- For more details about the treatment of the response for **AT+QIDEACT**, please refer to ***Chapter 7***. Theoretically the longest waiting time for this command is about 40s to 45s. Customers can set timeout value which is less than 40s in their own application according to their needs. The treatment method after the timeout is the same as that after receiving **CONNECT FAIL**.

3.2. Establish UDP Connection

Similarly, the module supports UDP mode too. Before using UDP mode, establish UDP connection first, and the way is the same as that of establishing TCP connection.

AT+QIOPEN="UDP","220.180.239.212","8062"	//Connect a remote UDP server whose address is 220.180.239.212:8062.
OK	
CONNECT OK	//Successfully connect to the remote UDP server. In fact, UDP connection does not need to be established in UDP mode. And the connection is for configuring the target IP address and port which the data will be sent to.

4 Send Data

4.1. Send Data in Unfixed Length

AT+QISEND	//Ready to send data.
>TEST<Ctrl+Z>	//Send data "TEST", and "<Ctrl+Z>" is used to end inputting data and begin to send all input data. Please note that the maximum length of the data to input at one time is 1460 bytes. And special characters "0x08" (means back space), "0x1A" (means <Ctrl+Z>) and "0x1B" (means ESC) cannot be sent in this mode. When module receives character "0x08", the last input character will be deleted. When module receives character "0x1A", it will stop receiving the following characters and begin to send all previously received data. And when module receives character "0x1B", it will exit from the sending operation.
SEND OK	//The data has been sent to TCP protocol layer successfully.
AT+QISACK	//Check whether the data has been sent successfully.
+QISACK: 4,4,0	//The first parameter "4" is the length of the data that has been sent. The second parameter "4" is the length of the data that has been acknowledged. The last parameter "0" is the length of the data that has been sent out but not acknowledged yet. If the last parameter is 0, it indicates that all the data have been sent successfully. This command is not meaningful in UDP mode because UDP is connectionless in the mode, and whether the data has been successfully sent cannot be confirmed. Therefore, in UDP mode, the parameter in the middle of the three parameters is always 0 after executing this command.
OK	

NOTES

1. The send buffer in the underlying socket is 7300. Therefore, there is no need to wait for acknowledgement of all data (which means the last parameter in the response of **AT+QISACK** is 0) before sending the next package. Given that the maximum length of one TCP package is 1448, it is recommended to set a threshold value as 3000. If the unacknowledged data length (the last parameter in the response of **AT+QISACK**) is less than 3000, continue to send the next package by command **AT+QISEND**. Otherwise, stop sending data, and then query the data length by

AT+QISACK every 5 seconds until the last parameter in the response of **AT+QISACK** is less than 3000. If the query time reaches a certain number (for example 20 times, which is equivalent to 100 seconds timeout) and the last parameter is always greater than 3000, it can be considered an abnormality occurred in the TCP connection. In this case close the TCP connection, re-establish it, and then continue to send all previously unacknowledged data and the data packages that need to be sent.

2. If non-transparent mode is selected for sending hexadecimal characters, especially 0x08, 0x1A and 0x1B, it is recommended to send data with fixed length. For detailed information, please refer to ***Chapter 4.2***.

4.2. Send Data in Fixed Length

```
AT+QISEND=3          //Specify the data length to be sent as 3.
>0x080x1A0x1B  //Send special characters “0x08”, “0x1A” and “0x1B” in turn. When inputting the data actually, be sure to input these special data in hexadecimal format 0x08, 0x1A and 0x1B directly rather than inputting a string of “0x080x1A0x1B” as the example shows. The example just gives a demonstration for users to see clearly.

SEND OK
AT+QISACK
+QISACK: 17,17,0

OK
```

5 Receive Data

5.1. Output the Data through UART Directly

The received data from the server will be outputted through UART without any head information in the default setting. In order to distinguish the received data from the responses of AT commands or any URC, it is recommended to add some head information to the received data. For more details, please refer to ***Chapter 2***. It is better to set the display format of the received data before the TCP connection has been established. Below is an example of the received data.

RECV FROM:220.180.239.212:8063<CR><LF>	//The module receives data from the remote server 220.180.239.212:8063. The notification will not be displayed if **AT+QISHOWRA=1** is not executed in initialization phase.
IPD36TCP:1234567890abcdefghijklmnopqrstuvwxyz	//Receive TCP data with the length of 36 bytes. The detailed data is the 36 characters after the colon symbol "**:**". The header "**IPD36TCP**:" will not be displayed if **AT+QIHEAD=1** is not executed in initialization phase. And the header "**TCP**" will not be displayed if **AT+QISHOWPT=1** is not executed in initialization phase.

NOTE

In this mode, the data will be outputted immediately once is received, so it is unavoidable that the data appears in the middle of the AT commands. In the current design, some timeout processing is carried out to avoid the received data from separating a complete AT command or its response, but it is still unavoidable separating the AT command and its response by the received data.

5.2. Retrieve the Received Data by Command

In data retrieving mode, after module receives the TCP data or UDP data, it outputs an URC to inform the client instead of outputting the data through UART immediately, and then the client can subsequently extract the received TCP data by the command.

Data retrieving mode is not supported by default and needs to be enabled by **AT+QINDI=1** before TCP connection is established. An example when the module receives the data in this mode is shown as below.

Suppose that the module receives the TCP data "1234567890abcdefghijklmnopqrstuvwxyz" from the remote server.

+QIRDI: 0,1,0	//The module receives the data based on Context 0, and acts as the client.

And then retrieve the data by the following command.

AT+QIRD=0,1,0,1500	//Retrieve the data from the module's socket buffer. The maximum length to retrieve is 1500 bytes. If the data length in the buffer is less than 1500 bytes, retrieve all the data from the buffer.
+QIRD: 220.180.239.212:8063,TCP,36<CR><LF> **1234567890abcdefghijklmnopqrstuvwxyz** **OK**	

6 TCP Connection Maintenance and Detection

Most of the modules are connected to the Internet through GPRS gateway and GPRS gateway has the similar functions as the router in LAN. Every time when the module tries to connect with one server on the internet, a port should be assigned to the module by GPRS gateway.

Since the port resource of GPRS gateway is limited, so it will have some restrictions on these ports for the terminals within the GPRS network. If there is no data transmission on the TCP connection for a period of time, it will release the port to other connection to achieve a rational allocation of port resource. So the TCP connection between the module and the server on the Internet may be disconnected by GPRS network without any notification.

There is no clear value at present about how long the resource will be released without data transmission. The test result in Shanghai is that the port resource will not be released within 10 minutes.

In order to avoid the TCP connection from disconnection by GPRS gateway without any notification, it is recommended to periodically send a small data packet to the remote server through which the TCP connection can be maintained and detected.

Example

For instance, according to the current results of our test, we set the interval for heartbeat packet sending as 10 minutes. (However, in some areas, operators may limit the expiration time of TCP connection as about 1 minute).

```
AT+QISACK                                   //Check the data sent.
+QISACK: 1448,1448,0                        //Suppose 1448 bytes data has been sent to the server successfully.
OK
AT+QISEND                                   //Send normal data.
>1234567890abcdefghijklmnopqrstuvwxyz<0x1A>   //Send data. “<0x1A>” indicates that to send the
                                              inputted data is required.

SEND OK                                     //The data has been sent to TCP protocol layer successfully
```

Assuming no data needs to be sent for a long time later, it is recommended to send a heartbeat packet to maintain the TCP connection. Before sending the heartbeat packet to maintain the TCP connection, confirm if the current TCP connection is normal by querying whether the current package has been sent

successfully. If the package is still not acknowledged two minutes later (query every 5 seconds for 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 it. After the connection has been established successfully, resent the unacknowledged data.

AT+QISACK	//Check the data sent.
+QISACK: 1484,1448,36	//The total length of the data which has been sent is 1484 bytes, but only 1448 bytes have been acknowledged, and the rest 36 bytes have not been acknowledged (cannot confirm whether the rest data has been sent successfully) yet. Check again every 5 seconds.
OK	
……	//Wait for 5 seconds.
AT+QISACK	//Check the data sent.
+QISACK: 1484,1484,0	//All 1484-byte data has been confirmed to be sent successfully. This means the TCP connection is normal by far.
OK	
……	//Wait for 8 minutes, no data is sent or received.
AT+QISEND	//Send a heartbeat data to server.
>Heart01<0x1A>	//Send heartbeat data “Heart 01”. It can be simpler. Users can define the specific format to avoid server misunderstanding.
SEND OK	
……	//Wait for 5 seconds, and then check if the heartbeat data has been sent successfully.
AT+QISACK	//Check the data sent.
+QISACK: 1491,1491,0	//The packet has been sent to server successfully, and the TCP connection is normal. If the packet has not been acknowledged (the third parameter in the response of **AT+QISACK** is not 0), please wait for 5 seconds and check again. Repeat the previous operations for several times until the packet has been acknowledged or the query times reaches a limitation (for example, 24 times that is 2 minutes timeout). If the query times reaches the limitation, the TCP connection may be abnormal, and it is recommended to re-establish the TCP connection.
OK	

7 Close TCP Connection

AT+QICLOSE	//Close the current TCP connection after all data has been sent.
CLOSE OK	
AT+QIDEACT	//If the GPRS context will not be used in a long period (for example more than one hour), it is recommended to close this context by command **AT+QIDEACT**. Normally the response time for this command is about 2s~5s. But when the network is very poor or in some abnormal conditions, the longest waiting time may reach about 40 seconds. It is recommended to set timeout value as less than 40 seconds in actual application. If **DEACT OK** is not received after timeout, user can restart the module by pins "EMERG_OFF" or "PWRKEY".
DEACT OK	

8 Transparent Session Mode

If **AT+QIMODE=1** is executed during initialization, the module enters into data mode after establishing TCP/UDP connection. In data mode, all the data inputted from UART will be considered as the data to be sent to the remote server, and all the data outputted from the UART is the data received from the remote server, except for "CLOSED", "+PDP DEACT" and other special texts. The following is an example of TCP transparent session mode.

8.1. Example

AT+QIOPEN=“TCP”,“220.180.239.212”,“8063”	//Connect the module to a remote TCP server whose IP address is 220.180.239.212:8063.
OK	//The syntax of the commands are right and TCP connection can be established in the current status of the module. For the details of other error responses, please refer to ***Chapter 3***.
CONNECT	//The module is successfully connected to the remote TCP server and UART has entered into the data mode. For the details of other error responses, please refer to ***Chapter 3***.
1234567890abcdefghijklmnopqrstuvwxyz	//The data can be sent now. For example, when “1234567890abcdefghijklmnopqrstuvwxyz" is inputted, the data will not be echoed by UART. Since the data length is less than 512 bytes (see configuration in ***Chapter 2***), the module will wait for 200ms to send the data (see configuration in ***Chapter 2***).
abcdefghijklmnopqrstuvwxyz1234567890	//These data are received from the server.
+++	//Input **+++** to quit the data mode.
OK	//The response of **+++**. **OK** means the UART has switched to the command mode successfully. Then if users want to close the TCP connection, please refer to the example in ***Chapter 7***.

8.2. Handling of Abnormalities

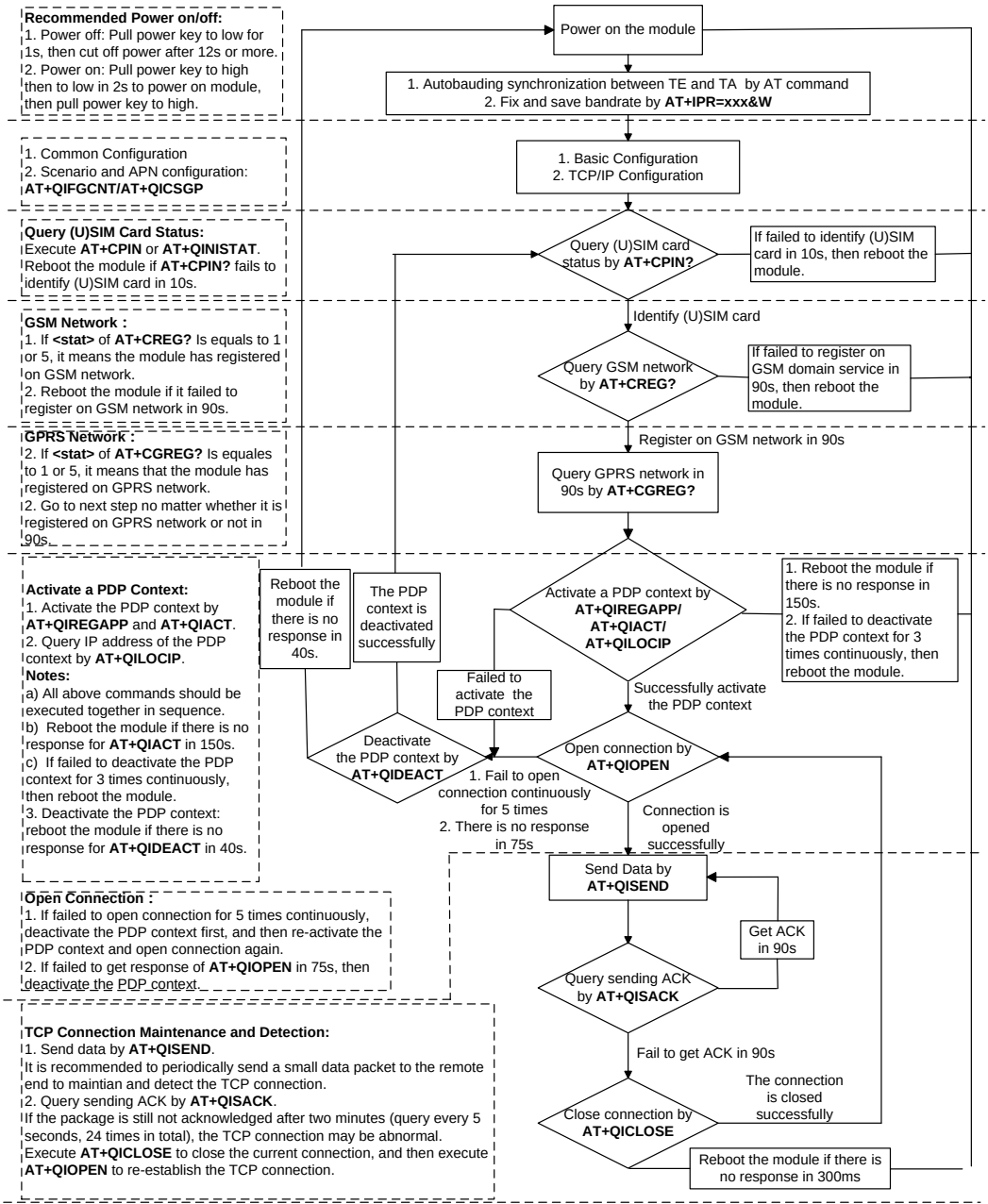
If the module receives **CLOSED** or **+PDP DEACT** in the data mode, it means TCP connection has been disconnected or some abnormalities have occurred. But it is also possible that these data are from the remote server.

In this case, it is recommended to input **+++** to confirm whether the UART is still in the data mode.

- When **+++** is inputted and **OK** is returned, it means the UART has switched to the command mode successfully and the previously received **CLOSED** or **+PDP DEACT** are TCP data from server.
- When **+++** is inputted, just **+++** is echoed (when echo mode is open) and **OK** is not returned, this means the module has already entered into command mode. That is, the previously received **CLOSED** or **+PDP DEACT** means the TCP connection has been disconnected or GPRS context has been deactivated. In this case, the TCP connection needs to be re-established by the steps below. If it is certain that the **CLOSED** and **+PDP DEACT** are not data sent by the remote server, then the TCP connection can be re-established directly and there is no need to use **+++** to judge the current state.

AT+QIDEACT	//Deactivate the current GPRS context. The response time for this command is normally about 2s~5s. But when the network is very poor or in some abnormal conditions, the longest waiting time will reach about 40 seconds. It is recommended to set timeout value as less than 40 seconds in actual application. If **DEACT OK** is not received after timeout, user can restart the module by pins “EMERG_OFF” and “PWRKEY”.
DEACT OK	

9 TCP/IP Recommended Process



Notes:

1. Ensure that TE and TA are correctly synchronized after rebooting the module. Send 15 AT commands continuously following the constraint that sleep 500ms after each AT command input, then fix and save bandrate configuration by AT+IPR=xxx&W after synchronized.
2. 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 failed to get response in 60s.
3. Reboot the module if the module has not got response of AT+QIACT in 150s or response of AT+QICLOSE in 300 ms and AT+QIDEACT in 40s .
4. It is NOT recommended to frequently reboot the module. When the module has been continuously rebooted for 3 times due to failed AT command execution, it can be rebooted immediately for the first time after that. If it still fails, reboot the module after 10 minutes for the second time, and reboot after 30 minutes for the third time, one hour for the fourth time, etc.

Figure 1: Recommended TCP/IP Process Flow Chart

10 Appendix A References

Table 1: Related Documents

SN	Document Name	Remark
[1]	Quectel_Mxx_AT_Commands_Manual	The introduction of the AT commands for GSM modules
[2]	Quectel_GSM_TCP(IP)_Application_Note	The introduction of how to use the internal TCP/IP stack for GSM modules

Table 2: Terms and Abbreviations

Abbreviation	Description
APN	Access Point Network
FGCNT	Foreground Context. The internal TCP/IP stack supports to activate two GPRS PDP contexts at the same time and Foreground context is the context controlled by the UART at present
GPRS	General Packet Radio Service
MUXIP	The function to visit several servers or listen to multiple clients based on the same GPRS/CSD context
TCP/IP	Transmission Control Protocol/Internet Protocol
UDP	User Datagram Protocol
UART	Universal Asynchronous Receiver/Transmitter