



SIM7672X & SIM7652X Series_TCPIP_Application Note

LTE Module

SIMCom Wireless Solutions Limited

SIMCom Headquarters Building, Building 3, No. 289 Linhong
Road, Changning District, Shanghai P.R. China

Tel: 86-21-31575100

support@simcom.com

www.simcom.com

Document Title:	SIM7672X & SIM7652X Series_TCPIP_Application Note
Version:	1.00
Date:	2023.05.22
Status:	Released

GENERAL NOTES

SIMCOM OFFERS THIS INFORMATION AS A SERVICE TO ITS CUSTOMERS, TO SUPPORT APPLICATION AND ENGINEERING EFFORTS THAT USE THE PRODUCTS DESIGNED BY SIMCOM. THE INFORMATION PROVIDED IS BASED UPON REQUIREMENTS SPECIFICALLY PROVIDED TO SIMCOM BY THE CUSTOMERS. SIMCOM HAS NOT UNDERTAKEN ANY INDEPENDENT SEARCH FOR ADDITIONAL RELEVANT INFORMATION, INCLUDING ANY INFORMATION THAT MAY BE IN THE CUSTOMER'S POSSESSION. FURTHERMORE, SYSTEM VALIDATION OF THIS PRODUCT DESIGNED BY SIMCOM WITHIN A LARGER ELECTRONIC SYSTEM REMAINS THE RESPONSIBILITY OF THE CUSTOMER OR THE CUSTOMER'S SYSTEM INTEGRATOR. ALL SPECIFICATIONS SUPPLIED HEREIN ARE SUBJECT TO CHANGE.

COPYRIGHT

THIS DOCUMENT CONTAINS PROPRIETARY TECHNICAL INFORMATION WHICH IS THE PROPERTY OF SIMCOM WIRELESS SOLUTIONS LIMITED. COPYING, TO OTHERS AND USING THIS DOCUMENT, ARE FORBIDDEN WITHOUT EXPRESS AUTHORITY BY SIMCOM. OFFENDERS ARE LIABLE TO THE PAYMENT OF INDEMNIFICATIONS. ALL RIGHTS RESERVED BY SIMCOM IN THE PROPRIETARY TECHNICAL INFORMATION, INCLUDING BUT NOT LIMITED TO REGISTRATION GRANTING OF A PATENT, A UTILITY MODEL OR DESIGN. ALL SPECIFICATION SUPPLIED HEREIN ARE SUBJECT TO CHANGE WITHOUT NOTICE AT ANY TIME.

SIMCom Wireless Solutions Limited

SIMCom Headquarters Building, Building 3, No. 289 Linhong Road, Changning District, Shanghai P.R. China

Tel: +86 21 31575100

Email: simcom@simcom.com

For more information, please visit:

https://www.simcom.com/technical_files.html

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

https://www.simcom.com/online_questions.html or email to: support@simcom.com

Copyright © 2023 SIMCom Wireless Solutions Limited All Rights Reserved.

About Document

Version History

Revision	Date	Owner	Description
V1.00	2023.05.22		New version

SIMCom
Confidential

Scope

Based on module AT command manual, this document will introduce TCPIP application process. Developers could understand and develop application quickly and efficiently based on this document. This document applies to SIM7672X Series, SIM7652X Series.

SIMCom
Confidential

Contents

About Document	2
Version History	2
Scope	3
Contents	4
1 Introduction	6
1.1 Purpose of the document	6
1.2 Related documents	6
1.3 Conventions and abbreviations	6
1.4 The process of Using TCPIP AT Commands	7
1.5 Error Handling	10
1.5.1 Executing TCP AT Commands Fails	10
1.5.2 PDP Activation Fails	10
1.5.3 Error Response of TCPIP Server	10
1.5.4 Description of Data Access Mode	10
2 AT Commands for TCPIP	12
2.1 Overview of AT Commands for TCP/IP	12
2.2 Detailed Description of AT Commands for TCP/IP	13
2.2.1 AT+NETOPEN Start Socket Service	13
2.2.2 AT+NETCLOSE Stop Socket Service	14
2.2.3 AT+CIPOPEN Establish Connection in Multi-Socket Mode	15
2.2.4 AT+CIPSEND Send data through TCP or UDP Connection	18
2.2.5 AT+CIPRXGET Set the Mode to Retrieve Data	21
2.2.6 AT+CIPCLOSE Close TCP or UDP Socket	24
2.2.7 AT+IPADDR Inquire Socket PDP address	26
2.2.8 AT+CIPHEAD Add an IP Header When Receiving Data	27
2.2.9 AT+CIPSRIP Show Remote IP Address and Port	28
2.2.10 AT+CIPMODE Set TCP/IP Application Mode	30
2.2.11 AT+CIPTIMEOUT Set TCP/IP Timeout Value	31
2.2.12 AT+CIPCCFG Configure Parameters of Socket	32
2.2.13 AT+SERVERSTART Startup TCP Sever	33
2.2.14 AT+SERVERSTOP Stop TCP Sever	35
2.2.15 AT+CIPACK Query TCP Connection Data Transmitting Status	36
2.2.16 AT+CDNSGIP Query the IP Address of Given Domain Name	37
2.2.17 AT+CSOCKETPN Set active PDP context's profile	38
2.2.18 AT+CTCPKA Conigure TCP heartbeat	39
2.2.19 AT+CDNSCFG Configure Domain Name Server	40
2.2.20 AT+CIPSENDMODE Set Sending Mode	42

3	TCPIP Examples	44
3.1	Configure and Activate context	44
3.1.1	Network Environment	44
3.1.2	Configure Context	44
3.1.3	Activate context	45
3.1.4	Deactivate Context	45
3.2	TCP Client	46
3.2.1	TCP Client Works in Direct Push Mode	46
3.2.2	TCP Client Works in Buffer Access Mode	46
3.2.3	TCP Client Works in Transparent Access Mode	48
3.3	UDP Client	49
3.3.1	UDP Client Works in Direct Push Mode	49
3.3.2	UDP Client Works in Buffer Access Mode	50
3.3.3	UDP Client Works in Transparent Access Mode	51
3.4	TCP Server	52
3.4.1	Transparent Mode	52
3.4.2	Non-Transparent Mode	53
3.4.3	Query Connection Status	54
4	Appendix	56
4.1	Summary of Error Codes	56
4.2	Unsolicited Result Codes	57

SIMCom
Confidential

1 Introduction

1.1 Purpose of the document

Based on module AT command manual, this document will introduce TCPIP application process. Developers could understand and develop application quickly and efficiently based on this document.

1.2 Related documents

[1] SIM7672X & SIM7652X Series_AT Command Manual

1.3 Conventions and abbreviations

In this document, the GSM engines are referred to as following term:

ME (Mobile Equipment);

MS (Mobile Station);

TA (Terminal Adapter);

DCE (Data Communication Equipment);

In application, controlling device controls the GSM engine by sending AT Command via its serial interface.

The controlling device at the other end of the serial line is referred to as following term:

TE (Terminal Equipment);

DTE (Data Terminal Equipment) or plainly "the application" which is running on an embedded system;

Other Conventions:

PDP(Packet Data Protocol);

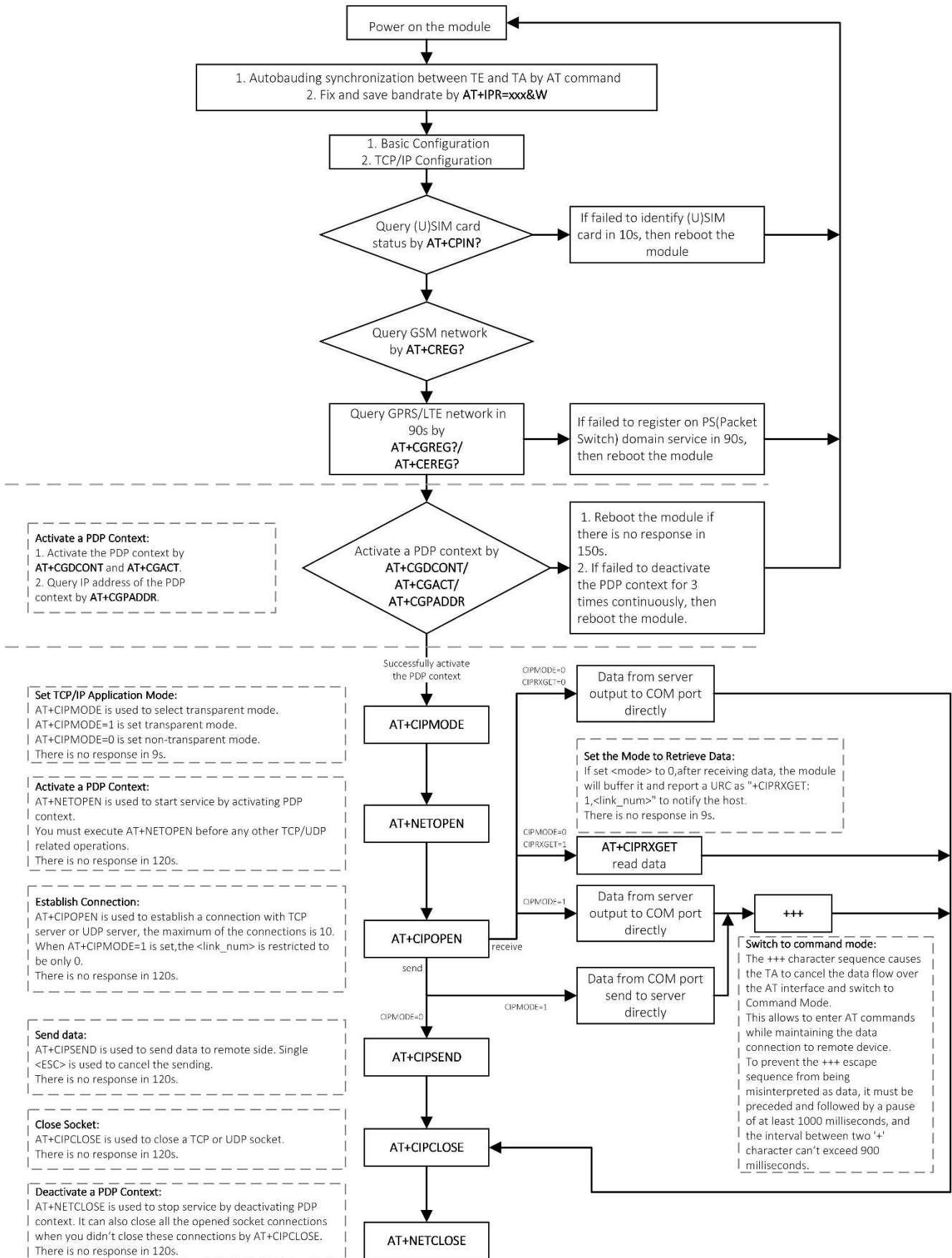
TCP(Terminal Control Protocol);

UDP(User Datagram Protocol);

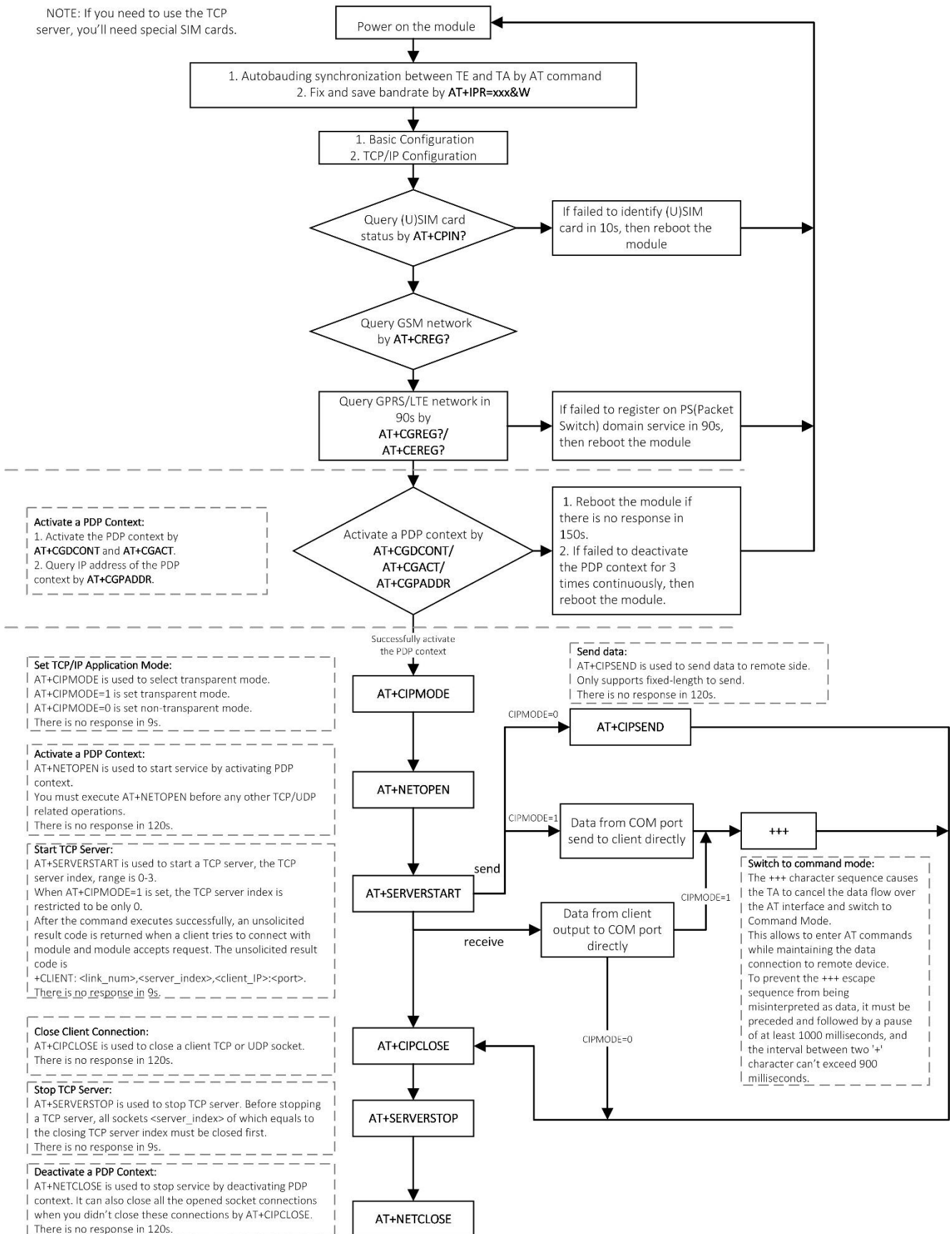
1.4 The process of Using TCPIP AT Commands

Figure illustrates how to use TCP/IP AT commands:

SIMCom
Confidential



NOTE: If you need to use the TCP server, you'll need special SIM cards.



1.5 Error Handling

1.5.1 Executing TCP AT Commands Fails

When executing TCPIP AT commands, if ERROR response is received from the module, please check whether the U(SIM) card is inserted and whether it is **+CPIN: READY** returned when executing **AT+CPIN?**.

1.5.2 PDP Activation Fails

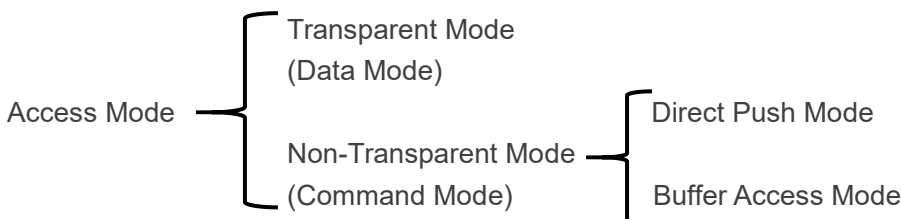
If it is failed to activate a PDP context with **AT+NETOPEN** command, please make sure the PDP is not activated. You can use **AT+NETOPEN?** to query it.

If all above configurations are correct, but activating the PDP context by **AT+NETOPEN** command still fails, please reboot the module to resolve this issue. After rebooting the module, please check the configurations mentioned above for at least.

1.5.3 Error Response of TCPIP Server

If you encounter other errors, please refer to chapter 4.1 to correct them.

1.5.4 Description of Data Access Mode



The default mode is direct push mode.

1. Direct Push Mode

In direct push mode, user can send data by **AT+CIPSEND**. The received data will be outputted to COM port

directly by URC as "+RECV FROM:<IP ADDRESS>:<PORT><CR><LF>+IPD(data length)<CR><LF><data>".

2. Buffer Access Mode

AT+CIPRXGET=1 is used to enter into buffer access mode. In buffer access mode, user sends data by **AT+CIPSEND**. After receiving data, the module will buffer it and report a URC as "+CIPRXGET: 1,<link_num>" to notify the host. Then host can retrieve data by **AT+CIPRXGET**.

3. Transparent Access Mode

AT+CIPMODE=1 is used to enter into transparent access mode. In transparent mode, the data received from COM port will be sent to internet directly, and the received data from Internet will be output to COM port directly as well. "+++" is used to exit from transparent access mode. When "+++" returns OK, the module will be switched to command mode. In transparent access mode, host cannot execute any AT command. Note: Currently, only one socket is available under transparent mode, either TCP client or TCP server. In transparent mode, the first server (<server_index> = 0) and the first client socket (<link_num> = 0) are used for transparent mode operation. Other servers (<server_index> = 1-3) and other client sockets (<link_num> = 1-9) are still used in command mode.

4. Switch Between Data Mode and Command Mode

(1) Data mode -> Command mode

Software switching: escape sequence +++. Please take care, this is a complete command, do not separate each character. And the time delay before and after this sequence should be more than 1000 milliseconds, the interval of each character should not be more than 900 milliseconds.

Hardware switching: DTR pin could be used to trigger data mode and command mode. Command **AT&D1** should be configured before application.

(2) Command Mode -> Data Mode

ATO is used to enter into transparent access mode from command mode. If it enters into transparent access mode successfully, CONNECT<text> will be returned.

2 AT Commands for TCPIP

2.1 Overview of AT Commands for TCP/IP

Command	Description
AT+NETOPEN	Start Socket Service
AT+NETCLOSE	Stop Socket Service
AT+CIOPEN	Establish Connection in Multi-Socket Mode
AT+CIPSEND	Send data through TCP or UDP Connection
AT+CIPRXGET	Set the Mode to Retrieve Data
AT+CIPCLOSE	Close TCP or UDP Socket
AT+IPADDR	Inquire Socket PDP address
AT+CIPHEAD	Add an IP Header When Receiving Data
AT+CIPSRIP	Show Remote IP Address and Port
AT+CIPMODE	Set TCP/IP Application Mode
AT+CIPTIMEOUT	Set TCP/IP Timeout Value
AT+CIPCCFG	Configure Parameters of Socket
AT+SERVERSTART	Startup TCP Server
AT+SERVERSTOP	Stop TCP Server
AT+CIPACK	Query TCP Connection Data Transmitting Status
AT+CDNSGIP	Query the IP Address of Given Domain Name
AT+CSOCKSETPN	Set active PDP context's profile
AT+CTCPKA	Configure TCP heartbeat
AT+CDNSCFG	Configure Domain Name Server
AT+CIPSENDMODE	Set Sending Mode

For more detailed information, please refer to SIM7672X & SIM7652X Series_AT Command Manual.

2.2 Detailed Description of AT Commands for TCP/IP

2.2.1 AT+NETOPEN Start Socket Service

AT+NETOPEN is used to start service by activating PDP context. You must execute **AT+NETOPEN** before any other TCP/UDP related operations.

AT+NETOPEN Start Socket Service	
Read Command AT+NETOPEN?	Response +NETOPEN: <net_state> OK
Execute Command AT+NETOPEN	Response 1)If the PDP context has not been activated or the network closed abnormally, response: OK +NETOPEN: <err> 2)When the PDP context has been activated successfully, if you execute AT+NETOPEN again, response: +IP ERROR: Network is already opened ERROR 3)other: ERROR
Parameter Saving Mode	NO_SAVE
Max Response Time	Range: 3000ms-120000ms default: 120000ms (it can be set by AT+CIPTIMEOUT)
Reference	3GPP TS 27.005

Defined Values

<net_state>	Integer type, indicates the state of PDP context activation. 0 network close (deactivated) 1 network open(activated)
<err>	Integer type, the result of operation. 0 is success, other value is failure, please refer to Chapter 4.1 for

details

Examples

AT+NETOPEN?

+NETOPEN: 1

OK

AT+NETOPEN

OK

+NETOPEN: 0

2.2.2 AT+NETCLOSE Stop Socket Service

AT+NETCLOSE is used to stop service by deactivating PDP context. It can also close all the opened socket connections when you didn't close these connections by **AT+CIPCLOSE**.

AT+NETCLOSE Stop Socket Service

Test Command

AT+NETCLOSE=?

Response

OK

Response

1)If the PDP context has been activated, response:

OK

+NETCLOSE: <err>

2)If the PDP context has been activated and one connection is in non-transparent mode when service type is TCP, response:

OK

Execute Command

AT+NETCLOSE

+CIPCLOSE: <link_num>,<err>

+NETCLOSE: <err>

3)If the PDP context has been activated and one connection is in transparent mode when service type is TCP, response:

OK

CLOSED

+CIPCLOSE: <link_num>,<err>

	<p>+NETCLOSE: <err> 4)If the PDP context has been activated and one connection is in non-transparent mode when service type is UDP, response: +CIPCLOSE: <link_num>,<err></p> <p>OK</p> <p>+NETCLOSE: <err> 5)If the PDP context has not been activated, response: +NETCLOSE: <err></p> <p>ERROR 6)Others: ERROR</p>
Parameter Saving Mode	NO_SAVE
Max Response Time	Range: 3000ms-120000ms default: 120000ms (it can be set by AT+CIPTIMEOUT)
Reference	

Defined Values

<err>	Integer type, the result of operation. 0 is success, other value is failure, please refer to Chapter 4.1 for details
-------	---

Examples

AT+NETCLOSE

OK

+NETCLOSE: 0

2.2.3 AT+CIPOPEN Establish Connection in Multi-Socket Mode

AT+CIPOPEN is used to establish a connection with TCP server and UDP server, the maximum of the connections is 4.

AT+CIPOPEN Establish Connection in Multi-Socket Mode

Test Command	Response
--------------	----------

<p>AT+CIOPEN=?</p>	<p>+CIOPEN: (0-3),("TCP","UDP","UDP SERVER")</p> <p>OK</p>
<p>Read Command AT+CIOPEN?</p>	<p>Response +CIOPEN: <link_num>[,<type>,<serverIP>,<serverPort>,<index>] +CIOPEN: <link_num>[,<type>,<serverIP>,<serverPort>,<index>] [...]</p> <p>OK If a connection identified by <link_num> has not been established successfully, only +CIOPEN: <link_num> will be returned.</p>
<p>Write Command TCP connection AT+CIOPEN=<link_num>,"TCP",<serverIP>,<serverPort>[,<localPort>]</p>	<p>Response 1)if PDP context has been activated successfully, response: OK +CIOPEN: <link_num>,<err> 2)when the <link_num> is greater than 3, response: +IP ERROR: Invalid parameter</p> <p>ERROR 3)If PDP context has not been activated, or the connection has been established, or parameter is incorrect, or when AT+CIPMODE=1 is set, the <link_num> is greater than 0, or other errors, response: +CIOPEN: <link_num>,<err></p> <p>ERROR 4)Transparent mode for TCP connection: When you want to use transparent mode to transmit data, you should set AT+CIPMODE=1 before AT+NETOPEN. And if AT+CIPMODE=1 is set, the <link_num> is restricted to be only 0. if success CONNECT [<text>] if failure CONNECT FAIL</p> <p>5)Others: ERROR</p>
<p>Write Command UDP Connection AT+CIOPEN=<link_num>,"UDP",,,<localPort></p>	<p>1)If PDP context has been activated successfully, response: +CIOPEN: <link_num>,0</p> <p>OK 2)When the <link_num> is greater than 3, response: +IP ERROR: Invalid parameter</p>

	<p>ERROR If PDP context has not been activated, or the connection has been established, or parameter is incorrect, or other errors, response: +CIPOPEN: <link_num>,<err></p> <p>ERROR 3)Others:</p> <p>ERROR</p>
<p>Write Command UDP server Connection AT+CIPOPEN=<link_num>,"UDP SERVER",,<localPort></p>	<p>1)If PDP context has been activated successfully, response: +CIPOPEN: <link_num>,0</p> <p>OK 2)When the <link_num> is greater than 3, response: +IP ERROR: Invalid parameter</p> <p>ERROR If PDP context has not been activated, or the connection has been established, or parameter is incorrect, or other errors, response: +CIPOPEN: <link_num>,<err></p> <p>ERROR 3)Others:</p> <p>ERROR</p>
<p>Parameter Saving Mode</p>	<p>NO_SAVE</p>
<p>Max Response Time</p>	<p>Range: 3000ms-120000ms default: 120000ms (it can be set by AT+CIPTIMEOUT)</p>
<p>Reference</p>	

Defined Values

<p><link_num></p>	<p>Integer type, identifies a connection. Range is 0-3. If AT+CIPMODE=1 is set, the <link_num> is restricted to be only 0.</p>
<p><type></p>	<p>String type, identifies the type of transmission protocol.</p> <p>TCP Transmission Control Protocol UDP User Datagram Protocol UDP SERVER User Datagram Protocol service</p>
<p><serverIP></p>	<p>String type, identifies the IP address of server. The IP address format consists of 4 octets, separated by decimal point, like "AAA.BBB.CCC.DDD". Also the domain name is supported here.</p>
<p><serverPort></p>	<p>Integer type, identifies the port of TCP server, range is 0-65535. NOTE: When open port as TCP, the port must be the opened TCP port;</p>

	When open port as UDP, the port may be any port.
<localPort>	Integer type, identifies the port of local socket, range is 0-65535.
<index>	Integer type, indicates whether the module is used as a client or server. When used as server, the range is 0-3, <index> is the server index to which the client is linked. -1 TCP client 0-3 TCP server index
<text>	String type, indicates CONNECT result code.
<err>	Integer type, the result of operation. 0 is success, other value is failure, please refer to Chapter 4.1 for details

Examples

AT+CIPOPEN=?

+CIPOPEN: (0-3),("TCP","UDP")

OK

AT+CIPOPEN?

+CIPOPEN: 0,"TCP","183.230.174.137",6041,-1

+CIPOPEN: 1

+CIPOPEN: 2

+CIPOPEN: 3

OK

AT+CIPOPEN=0,"TCP","183.230.174.137",6031

OK

//TCP connection

+CIPOPEN: 0,0

AT+CIPOPEN=5,"UDP",,,6031

+CIPOPEN: 5,0

// UDP Connection

OK

2.2.4 AT+CIPSEND Send data through TCP or UDP Connection

AT+CIPSEND is used to send data to remote side. If service type is TCP, the data is firstly sent to the module's internal TCP/IP stack, and then sent to server by protocol stack. The <length> field may be empty. While it is empty, each <Ctrl+Z> means end of the input data. Each <ESC> is used to cancel the sending. And <Ctrl+Z> is 0x1A, <ESC> is 0x1B.

AT+CIPSEND Send data through TCP or UDP Connection	
<p>Test Command</p> <p>AT+CIPSEND=?</p>	<p>Response</p> <p>+CIPSEND: (0-3),(1-1500)</p> <p>OK</p>
<p>Write Command</p> <p>If service type is "TCP", send data with changeable length</p> <p>AT+CIPSEND=<link_num></p> <p>Response ">", then type data to send, tap CTRL+Z to send data, tap ESC to cancel the operation</p>	<p>Response</p> <p>1) If the connection identified by <link_num> has been established successfully, response:</p> <p>></p> <p><input data></p> <p>CTRL+Z</p> <p>OK</p> <p>+CIPSEND: <link_num>,<reqSendLength>,<cnfSendLength></p> <p>2) If <reqSendLength> is equal <cnfSendLength>, it means that the data has been sent to TCP/IP protocol stack successfully.</p> <p>3)If the connection has not been established, abnormally closed, or parameter is incorrect, response:</p> <p>+CIPERROR: <err></p> <p>ERROR</p> <p>4)Others:</p> <p>ERROR</p>
<p>Write Command</p> <p>If service type is "TCP", send data with fixed length</p> <p>AT+CIPSEND=<link_num>,<length></p>	<p>Response</p> <p>1)If the connection identified by <link_num> has been established successfully, response:</p> <p>></p> <p><input data with specified length></p> <p>OK</p> <p>+CIPSEND: <link_num>,<reqSendLength>,<cnfSendLength></p> <p>2) If <reqSendLength> is equal <cnfSendLength>, it means that the data has been sent to TCP/IP protocol stack successfully.</p> <p>3)If the connection has not been established, abnormally closed, or parameter is incorrect, response:</p> <p>+CIPERROR: <err></p> <p>ERROR</p> <p>4)Others:</p> <p>ERROR</p>
<p>Write Command</p> <p>If service type is "UDP", send data with changeable length</p> <p>AT+CIPSEND=<link_num>,<server></p>	<p>Response</p> <p>1)If the connection identified by <link_num> has been established successfully, response:</p> <p>></p> <p><input data></p>

<p>verIP>,<serverPort> Response ">", then type data to send, tap CTRL+Z to send data, tap ESC to cancel the operation</p>	<p>CTRL+Z OK</p> <p>+CIPSEND: <link_num>,<reqSendLength>,<cnfSendLength></p> <p>2)If the connection has not been established, abnormally closed, or parameter is incorrect, response: +CIPERROR: <err></p> <p>ERROR 3)Others: ERROR</p>
<p>Write Command If service type is "UDP", send data with fixed length AT+CIPSEND=<link_num>,<length>,<serverIP>,<serverPort> Response ">", type data until the data length is equal to <length></p>	<p>Response 1)If the connection identified by <link_num> has been established successfully, response: > <input data with specified length> OK</p> <p>+CIPSEND: <link_num>,<reqSendLength>,<cnfSendLength></p> <p>2)If the connection has not been established, abnormally closed, or parameter is incorrect, response: +CIPERROR: <err></p> <p>ERROR 3)Others: ERROR</p>
<p>Parameter Saving Mode</p>	<p>NO_SAVE</p>
<p>Max Response Time</p>	<p>Range: 3000ms-120000ms default: 120000ms (it can be set by AT+CIPTIMEOUT)</p>
<p>Reference</p>	

Defined Values

<p><link_num></p>	<p>Integer type, identifies a connection. Range is 0-3.</p>
<p><length></p>	<p>Integer type, indicates the length of sending data, range is 1-1500.</p>
<p><serverIP></p>	<p>String type, identifies the IP address of server. The IP address format consists of 4 octets, separated by decimal point, like "AAA.BBB.CCC.DDD". Also the domain name is supported here.</p>
<p><serverPort></p>	<p>Integer type, identifies the port of TCP server, range is 0-65535. NOTE: When open port as TCP, the port must be the opened TCP port; When open port as UDP, the port may be any port.</p>

	But, for Qualcomm, connecting the port 0 is regarded as an invalid operation.
<reqSendLength>	Integer type, the length of the data requested to be sent
<cnfSendLength>	Integer type, the length of the data confirmed to have been sent -1 the connection is disconnected. 0 own send buffer or other side's congestion window are full. Note: If the <cnfSendLength> is not equal to the <reqSendLength>, the socket then cannot be used further.
<err>	Integer type, the result of operation. 0 is success, other value is failure, please refer to Chapter 4.1 for details

Examples

AT+CIPSEND=?

+CIPSEND: (0-3),(1-1500)

OK

AT+CIPSEND=1,5

>12345

OK

// If service type is "TCP", send data with fixed length

+CIPSEND: 1,5,5

AT+CIPSEND=8,5,"183.230.174.137",6031

>12345

OK

// If service type is "UDP", send data with fixed length

+CIPSEND: 8,5,5

NOTE

If you use UDP to send more than 1400 bytes of data when the server does not receive data, this may be the reason for the carrier, in this case please send no more than 1400 bytes of data.

If you use TCP to send data, the instruction can be followed by a comma just like "**AT+CIPSEND=0,**" or "**AT+CIPSEND=0,10,**" without an error, but it doesn't make any sense

2.2.5 AT+CIPRXGET Set the Mode to Retrieve Data

If set <mode> to 1, after receiving data, the module will buffer it and report a URC as "+CIPRXGET: 1, <link_num>" to notify the host. Then host can retrieve data by **AT+CIPRXGET**.

If set <mode> to 0, the received data will be outputted to COM port directly by URC as "RCV FROM :<IP ADDRESS>:<PORT><CR><LF>+IPD(data length)<CR><LF><data>".

The default value of <mode> is 0.

AT+CIPRXGET Set the Mode to Retrieve Data

Test Command AT+CIPRXGET=?	Response +CIPRXGET: (0-4),(0-3),(1-1500) OK
Read Command AT+CIPRXGET?	Response +CIPRXGET: <mode> OK
Write Command AT+CIPRXGET=<mode> In this case,<mode> can only be 0 or 1	Response 1) If the parameter is correct, response: OK 2)If the parameter is incorrect or other error, response: +IP ERROR: <err_info> ERROR 3)Others: ERROR
Write Command AT+CIPRXGET=2,<link_num>[,<len>] Retrieve data in ACSII form	Response 1) If <len> field is empty, the default value to read is 1500. If the buffer is not empty, response: +CIPRXGET: <mode>,<link_num>,<read_len>,<rest_len><data>ACSII form OK 2)If the buffer is empty, response: +IP ERROR: No data ERROR 3)If the parameter is incorrect or other error, response: +IP ERROR: <err_info> ERROR 4)Others: ERROR
Write Command AT+CIPRXGET=3,<link_num>[,<len>] Retrieve data in hex form	Response 1) If <length> field is empty, the default value to read is 750. If the buffer is not empty, response: +CIPRXGET: <mode>,<link_num>,<read_len>,<rest_len>

	<p><data> hex form</p> <p>OK 2)If the buffer is empty, response: +IP ERROR: No data</p> <p>ERROR 3)If the parameter is incorrect or other error, response: +IP ERROR: <err_info></p> <p>ERROR 4)Others: ERROR</p>
Write Command AT+CIPRXGET=4,<link_num>	<p>Response 1)If the parameter is correct, response: +CIPRXGET: 4,<link_num>,<rest_len></p> <p>OK 2)If the parameter is incorrect or other error, response: +IP ERROR: <err_info></p> <p>ERROR 3)Others ERROR</p>
Parameter Saving Mode	NO_SAVE
Max Response Time	9s
Reference	

Defined Values

<mode>	<p>Integer type, sets the mode to retrieve data</p> <p><u>0</u> set the way to get the network data automatically</p> <p>1 set the way to get the network data manually</p> <p>2 read data, the max read length is 1500</p> <p>3 read data in HEX form, the max read length is 750</p> <p>4 get the rest data length</p>
<link_num>	Integer type, identifies a connection. Range is 0-3.
<len>	<p>Integer type, the data length to be read.</p> <p>Not required, the default value is 1500 when <mode>=2, and 750 when <mode>=3.</p>
<read_len>	Integer type, the length of data that has been read.

<rest_len>	Integer type, the length of data which has not been read in the buffer.
<err_info>	String type, displays the cause of occurring error, please refer to Chapter 4.1 for more details.

Examples

AT+CIPRXGET=?

+CIPRXGET: (0-4),(0-3),(1-1500)

OK

AT+CIPRXGET?

+CIPRXGET: 1

OK

AT+CIPRXGET=1

OK

AT+CIPRXGET=2,0

+CIPRXGET: 2,0,6,0

123456

OK

AT+CIPRXGET=3,0

+CIPRXGET: 3,0,6,0

313233343536

OK

AT+CIPRXGET=4,0

+CIPRXGET: 4,0,18

OK

NOTE

1. When data is received and reported, the maximum length of <data length> is 1500 each time.
2. When **AT+CIPRXGET=1**, if multiple TCP/UDP connections are connected and the server is sending data all the time, it is necessary to read the data sent by the server in time.

2.2.6 AT+CIPCLOSE Close TCP or UDP Socket

AT+CIPCLOSE is used to close a TCP or UDP Socket

AT+CIPCLOSE Close TCP or UDP Socket

<p>Test Command AT+CIPCLOSE=?</p>	<p>Response +CIPCLOSE: (0-3)</p> <p>OK</p>
<p>Read Command AT+CIPCLOSE?</p>	<p>Response +CIPCLOSE: <link0_state>,<link1_state>,<link2_state>,<link3_state></p> <p>OK</p>
<p>Write Command AT+CIPCLOSE=<link_num></p>	<p>Response</p> <p>1)If service type is TCP and the connection identified by <link_num> has been established, response OK</p> <p>+CIPCLOSE: <link_num>,<err></p> <p>2)If service type is TCP and the access mode is transparent mode, response: OK</p> <p>CLOSED</p> <p>+CIPCLOSE: <link_num>,<err></p> <p>3)If service type is UDP and the connection identified by <link_num> has been established and closed successfully, response: +CIPCLOSE: <link_num>,0</p> <p>OK</p> <p>4)If service type is UDP and access mode is transparent mode, response: CLOSED</p> <p>+CIPCLOSE: <link_num>,<err></p> <p>OK</p> <p>5)If the connection has not been established, abnormally closed, or parameter is incorrect, response: +CIPCLOSE: <link_num>,<err></p> <p>ERROR</p> <p>6)Others: ERROR</p>

Parameter Saving Mode	NO_SAVE
Max Response Time	Range: 3000ms-120000ms default: 120000ms (it can be set by AT+CIPTIMEOUT)
Reference	

Defined Values

<link_num>	Integer type, identifies a connection. Range is 0-3.
<linkX_state>	Integer type, indicates state of connection identified by <link_num>. Range is 0-1. 0 disconnected 1 connected
<err>	Integer type, the result of operation. 0 is success, other value is failure, please refer to Chapter 4.1 for details

Examples

```

AT+CIPCLOSE=?
+CIPCLOSE: (0-3)

OK
AT+CIPCLOSE?
+CIPCLOSE: 0,0,0,0

OK
AT+CIPCLOSE=0
OK

+CIPCLOSE: 0,0

```

2.2.7 AT+IPADDR Inquire Socket PDP address

AT+IPADDR is used to get active PDP address.

AT+IPADDR Inquire Socket PDP Address

Test Command	Response
AT+IPADDR=?	OK

Execute Command AT+IPADDR	Response 1)If PDP context has been activated successfully, response +IPADDR: <ip_address> OK 2) +IP ERROR: Network not opened ERROR
Parameter Saving Mode	NO_SAVE
Max Response Time	default: 9000ms
Reference	-

Defined Values

<ip_address>	String type, identifies the IP address of current active socket PDP.
--------------	--

Examples

```
AT+IPADDR
+IPADDR: 10.84.17.161

OK
```

2.2.8 AT+CIPHEAD Add an IP Header When Receiving Data

AT+CIPHEAD is used to add an IP header when receiving data.

AT+CIPHEAD Add an IP Header When Receiving Data

Test Command AT+CIPHEAD=?	Response +CIPHEAD: (0-1) OK
Read Command AT+CIPHEAD?	Response +CIPHEAD: <mode> OK
Write Command AT+CIPHEAD=<mode>	Response 1)If the parameter is correct, response:

	OK 2) ERROR
Execute Command AT+CIPHEAD	Response Set default value:(<mode>=1) OK
Parameter Saving Mode	NO_SAVE
Max Response Time	default: 9000ms
Reference	-

Defined Values

<mode>	Integer type, indicates whether adding an IP header or not when receiving data 0 not add IP header <u>1</u> add IP header, the format is "+IPD(data length)"
--------	--

Examples

```
AT+CIPHEAD=?  
+CIPHEAD: (0-1)
```

```
OK  
AT+CIPHEAD?  
+CIPHEAD: 1
```

```
OK  
AT+CIPHEAD=1  
OK  
AT+CIPHEAD  
OK
```

2.2.9 AT+CIPSRIP Show Remote IP Address and Port

AT+CIPSRIP is used to set whether to display IP address and port of server when receiving data.

AT+CIPSRIP Show Remote IP Address and Port

Test Command AT+CIPSRIP=?	Response +CIPSRIP: (0-1)
-------------------------------------	------------------------------------

Read Command AT+CIPSRIP?	OK Response +CIPSRIP: <mode>
Write Command AT+CIPSRIP=<mode>	OK Response 1)If the parameter is correct, response: OK 2) ERROR
Execute Command AT+CIPSRIP	Response Set default value:(<mode>=1) OK
Parameter Saving Mode	NO_SAVE
Max Response Time	default: 9000ms
Reference	-

Defined Values

<mode>	Integer type, indicates whether to show IP address and port of server or not when receiving data. 0 not show <u>1</u> show, the format is as follows: "RECV FROM:<IP ADDRESS>:<PORT>"
---------------------	--

Examples

```
AT+CIPSRIP=?
+CIPSRIP: (0-1)
```

```
OK
```

```
AT+CIPSRIP?
+CIPSRIP: 1
```

```
OK
```

```
AT+CIPSRIP=0
```

```
OK
```

```
AT+CIPSRIP
```

```
OK
```

2.2.10 AT+CIPMODE Set TCP/IP Application Mode

AT+CIPMODE is used to select transparent mode(data mode) or non-transparent mode(command mode).The default mode is non-transparent mode.

AT+CIPMODE Set TCP/IP Application Mode	
Test Command AT+CIPMODE=?	Response +CIPMODE: (0-1) OK
Read Command AT+CIPMODE?	Response +CIPMODE: <mode> OK
Write Command AT+CIPMODE=<mode>	Response 1)If the parameter is correct, response: OK 2) ERROR
Execute Command AT+CIPMODE	Response Set default value:(<mode>=0) OK
Parameter Saving Mode	NO_SAVE
Max Response Time	default: 9000ms
Reference	-

Defined Values

<mode>	Integer type, sets TCP/IP application mode <u>0</u> Non transparent mode 1 Transparent mode
---------------------	---

Examples

```
AT+CIPMODE=?
+CIPMODE: (0-1)

OK
AT+CIPMODE?
+CIPMODE: 0
```

```
OK
AT+CIPMODE=1
OK
AT+CIPMODE
OK
```

NOTE

To use transparent mode for data transmission, **AT+CIPMODE=1** should be set before **AT+NETOPEN**.

2.2.11 AT+CIPTIMEOUT Set TCP/IP Timeout Value

AT+CIPTIMEOUT is used to set timeout value for **AT+NETOPEN/AT+CIPOPEN/AT+CIPSEND**.

AT+CIPTIMEOUT Set TCP/IP Timeout Value

Read Command AT+CIPTIMEOUT?	Response +CIPTIMEOUT: <netopen_timeout>,<cipopen_timeout>,<cipsend_timeout>
Write Command AT+CIPTIMEOUT=[<netopen_timeout>],[<cipopen_timeout>],[<cipsend_timeout>]]	OK Response 1)If the parameter is correct, response: OK 2) ERROR
Parameter Saving Mode	NO_SAVE
Max Response Time	default: 9000ms
Reference	-

Defined Values

<netopen_timeout>	Integer type, timeout value for AT+NETOPEN . Default is120000ms. Range is 3000ms-120000ms.
<cipopen_timeout>	Integer type, timeout value for AT+CIPOPEN . Default is120000ms. Range is 3000ms-120000ms.
<cipsend_timeout>	Integer type, timeout value for AT+CIPSEND . Default is120000ms. Range is 3000ms-120000ms.

	packet. Range is 0-10. The default value is 10.
<DelayTm>	Integer type, number of milliseconds to delay to output data of Receiving. Range is 0-1000. The default value is 0.
<Ack>	Integer type, it can only be set to 0. It's used to be compatible with old TCP/IP command set.
<errMode>	Integer type, sets mode of reporting <err_info>, default value is 1. 0 error result code with numeric values 1 error result code with string values
<HeaderType>	Integer type, select which data header is used when receiving data, it only takes effect in multi-client mode. Default value is 0. 0 add data header, the format is "+IPD<data length>" 1 add data header, the format is "+RECEIVE,<link num>,<data length>"
<AsyncMode>	Integer type, range is 0-1. Default value is 0. It's used to be compatible with old TCP/IP command set.
<TimeoutVal>	Integer type, set the minimum retransmission timeout value for TCP connection. Range is 500ms-120000ms. Default is 500ms.

Examples

AT+CIPCCFG=?

+CIPCCFG: (0-10),(0-1000),(0),(0-1),(0-1),(0-1),(500-120000)

OK

AT+CIPCCFG?

+CIPCCFG: 10,0,0,1,0,0,500

OK

AT+CIPCCFG=2

OK

AT+CIPCCFG

OK

2.2.13 AT+SERVERSTART Startup TCP Sever

AT+SERVERSTART is used to startup a TCP server, and the server can receive the request of TCP client. After the command executes successfully, an unsolicited result code is returned when a client tries to connect with module and module accepts request. The unsolicited result code is +CLIENT:
<link_num>,<server_index>,<client_IP>:<port>.

AT+SERVERSTART Startup TCP Sever

Test Command AT+SERVERSTART=?	Response +SERVERSTART: (10000-10015),(0-1) OK
Read Command AT+SERVERSTART?	Response 1)If the PDP context has not been activated successfully, response: +CIPERROR: <err> ERROR 2)If there exists opened server, response: [+SERVERSTART: <server_index>,<port> ...] OK 3)Others: ERROR
Write Command AT+SERVERSTART=<port>,<server_index>[,<backlog>]	Response 1)If there is no error, response: OK 2)If the PDP context has not been activated, or the server identified by <server_index> has been opened, or the parameter is not correct, or other errors, response: +CIPERROR: <err> ERROR 3)Others: ERROR
Parameter Saving Mode	NO_SAVE
Max Response Time	default: 9000ms
Reference	-

Defined Values

<port>	Integer type, identifies the listening port of module when used as a TCP server. Range is 10000-10015.
<server_index>	Integer type, the TCP server index, range is 0-3.
<backlog>	Integer type, the maximum connections can be queued in listening queue. Range is 1-3. Default is 3.

Examples

AT+SERVERSTART=?

+SERVERSTART: (10000-10015),(0-1)

OK

AT+SERVERSTART?

OK

AT+SERVERSTART=10000,0

OK

2.2.14 AT+SERVERSTOP Stop TCP Sever

AT+SERVERSTOP is used to stop TCP server. Before stopping a TCP server, all sockets <server_index> of which equals to the closing TCP server index must be closed first.

AT+SERVERSTOP Stop TCP Sever

Write Command

AT+SERVERSTOP=<server_index>
ex>

Response

1)If there exists open connection with the server identified by <server_index>, or the server identified by <server_index> has not been opened, or the parameter is incorrect, response:

+SERVERSTOP: <server_index>,<err>

ERROR

2)If the server socket is closed immediately, response:

+SERVERSTOP: <server_index>,0

OK

(In general, the result is shown as below.)

3)If the server socket starts to close, response:

OK

+SERVERSTOP: <server_index>,<err>

4)Others:

ERROR

Parameter Saving Mode

NO_SAVE

Max Response Time

default: 9000ms

Reference

-

Defined Values

<server_index> Integer type, the TCP server index, range is 0-1.

<err> Integer type, the result of operation.

0 is success, other value is failure, please refer to Chapter 4.1 for details

Examples

AT+SERVERSTOP=0

OK

+SERVERSTOP: 0,0

2.2.15 AT+CIPACK Query TCP Connection Data Transmitting Status

AT+CIPACK is used to query TCP connection data transmitting status.

AT+CIPACK Query Connection Data Transmitting State

Test Command AT+CIPACK=?	Response +CIPACK: (range of supported <link_num>s) OK
Write Command AT+CIPACK=<link_num>	Response 1)If the PDP context has not been activated, or the connection identified by <link_num> has not been established, abnormally closed, or the parameter is incorrect, or other errors, response: +IP ERROR: <err_info> ERROR 2)If the connection has been established, and the service type is "TCP", response: +CIPACK: <sent_data_size>,<ack_data_size>,<recv_data_size> OK
Parameter Saving Mode	NO_SAVE
Max Response Time	default: 9000ms
Reference	-

Defined Values

<link_num>	Integer type, identifies a connection. Range is 0-3.
<sent_data_size>	Integer type, the total length of sent data
<ack_data_size>	Reserve

<recv_data_size>	Integer type, the total length of received data
<err>	Integer type, the result of operation. 0 is success, other value is failure, please refer to Chapter 4.1 for details
<err_info>	String type, displays the cause of occurring error, please refer to Chapter 4.1 for details.

Examples

AT+CIPACK=?

+CIPACK: (0-3)

OK

AT+CIPACK=0

+CIPACK: 10,10,5

OK

2.2.16 AT+CDNSGIP Query the IP Address of Given Domain Name

AT+CDNSGIP is used to query the IP address of given domain name.

AT+CDNSGIP Query the IP Address of Given Domain Name

Test Command	Response
AT+CDNSGIP=?	OK
Write Command	Response
AT+CDNSGIP=<domain name>	1)If the given domain name has related IP, response: +CDNSGIP: 1,<domain name>,<IP address>
	OK
	2)If the given name has no related IP, response: +CDNSGIP: 0,<dns error code>
	ERROR
	3)Others: ERROR
Parameter Saving Mode	NO_SAVE
Max Response Time	default: 6s
Reference	-

Defined Values

<domain name>	String type (string should be included in quotation marks), indicates the domain name. The maximum length of domain name is 254. Valid characters allowed in the domain name area include a-z, A-Z, 0-9, "-" (hyphen)and ".". A domain name is made up of one label name or more label names separated by "." (eg: AT+CDNSGIP="aa.bb.cc"). For label names separated by ".", length of each label must be no more than 63 characters. The beginning character of the domain name and of labels should be an alphanumeric character.
<IP address>	String type, indicates the IP address corresponding to the domain name.
<dns error code>	Integer type, indicates the error code. 10 DNS GENERAL ERROR

Examples

```

AT+CDNSGIP=?
OK
AT+CDNSGIP="www.baidu.com"
+CDNSGIP: 1,"www.baidu.com","61.135.169.121"
OK

```

2.2.17 AT+CSOCKSETPN Set active PDP context's profile

This command sets default active PDP context's profile number and type. When we activate PDP by using **AT+NETOPEN** command, we need use the default profile number and type, and the context of this profile is set by **AT+CGDCONT** command.

AT+CSOCKSETPN Set active PDP context's profile

Test Command AT+CSOCKSETPN=?	Response +CSOCKSETPN: 1,(1,6) OK
Read Command AT+CSOCKSETPN?	Response +CSOCKSETPN: <profile_num>,<ip_family> OK
Write Command AT+CSOCKSETPN=<profile_nu	Response 1)If the parameter is correct, response:

m>[,<ip_family>]	OK 2)If the parameter is wrong, or NETOPEN is already active, response: ERROR
Parameter Saving Mode	NO_SAVE
Maximum Response Time	default: 9000ms
Reference	-

Defined Values

<profile_num>	Packet Data Protocol context's profile number. Now only 1 is supported for this parameter value.
<ip_family>	Packet Data Protocol type 1 IPv4 6 IPv6

NOTE

This platform <profile_num> can only be set to 1.

Examples

```
AT+CSOCKSETPN=?
+CSOCKSETPN: 1,(1,6)

OK
AT+CSOCKSETPN?
+CSOCKSETPN: 1,1

OK
AT+CSOCKSETPN=1,6
OK
```

2.2.18 AT+CTCPKA Conigure TCP heartbeat

This command is used to set TCP heartbeat parameters. Set this up after we activate PDP by using **AT+NETOPEN** command.

AT+CTCPKA Conigure TCP heartbeat

Test Command AT+CTCPKA=?	Response OK
Read Command AT+CTCPKA?	Response +CTCPKA: <keepalive>,<keepidle>,<keepcount>,<keepinterval>
Write Command AT+CTCPKA=<keepalive>,<keepidle>,<keepcount>,<keepinterval>	Response 1)If successfully: OK 2)If failed: ERROR
Parameter Saving Mode	NO_SAVE
Maximum Response Time	default: 9000ms
Reference	-

Defined Values

<keepalive >	Set TCP keepalive option. 0 Disable TCP keep alive mechanism 1 Enable TCP keep alive mechanism
<keepidle>	The unit is minute. If there is no data interaction within this period, the probe is performed. (1-120)
<keepcount>	Number of probe retries. If all times out, the connection is considered Invalid.(1-10)
<keepinterval>	The unit is minute. Interval for sending probe packets during probe.

Examples

```
AT+CTCPKA=1,2,5,1
```

```
OK
```

```
AT+CTCPKA?
```

```
+CTCPKA: 1,2,5,1
```

```
OK
```

2.2.19 AT+CDNSCFG Configure Domain Name Server

This command is used to configure Domain Name Server.

AT+CDNSCFG Configure Domain Name Server

Test Command AT+CDNSCFG=?	Response +CDNSCFG: ("Primary DNS"),("Secondary DNS"),type OK
Read Command AT+CDNSCFG?	Response Primary IPv4 DNS: <pri_dns>,Secondary IPv4 DNS: <pri_dns> Primary IPv6 DNS: <pri_dns>,Secondary IPv6 DNS: <pri_dns> OK
Write Command AT+CDNSCFG=<pri_dns>[,<sec_dns>][,<type>]	Response 1)If successfully: OK 2)If failed: ERROR
Parameter Saving Mode	NO_SAVE
Maximum Response Time	default: 9000ms
Reference	-

Defined Values

<pri_dns>	A string parameter which indicates the IP address of the primary domain name server.
<sec_dns>	A string parameter which indicates the IP address of the secondary domain name server.
<type>	0 Set the server for the ipv4 network 1 Set the server for the ipv6 network

Examples

AT+CDNSCFG?

Primary IPv4 DNS: 183.230.126.224,Secondary IPv4 DNS: 183.230.126.225

OK

AT+CDNSCFG=183.230.126.224,183.230.126.225,0

OK

NOTE

This platform does not support IPv6 by default. If you want to set the IPv6 DNS server, first use **AT+CFUN=0** to shut down, then use **AT+CGDCONT** to set the APN of IPV4V6, and then use **AT+CFUN=1** to start up.

2.2.20 AT+CIPSENDMODE Set Sending Mode

AT+CIPSENDMODE is used to select sending mode when service type is "TCP".

If set <mode> to 1, when sending data by **AT+CIPSEND**, the URC "**+CIPSEND:**

<link_num>,<reqSendLength>,<cnfSendLength>" will not be returned until module receives the server's ACK message to the sent data last time.

If set <mode> to 0, the URC "**+CIPSEND: <link_num>,<reqSendLength>,<cnfSendLength>**" will be returned If the data has been sent to module's internal TCP/IP protocol stack. In this case, the module doesn't need to wait for the server's ACK message.

The default mode is sending without waiting peer TCP ACK mode.

AT+CIPSENDMODE Set Sending Mode	
Test Command AT+CIPSENDMODE=?	Response +CIPSENDMODE: (0-1) OK
Read Command AT+CIPSENDMODE?	Response +CIPSENDMODE: <mode> OK
Write Command AT+CIPSENDMODE=<mode>	Response 1)If the parameter is correct, response: OK 2) ERROR
Parameter Saving Mode	NO_SAVE
Max Response Time	default: 9000ms
Reference	-

Defined Values

<mode>	Integer type, sets sending mode <u>0</u> sending without waiting peer TCP ACK mode 1 sending wait peer TCP ACK mode
--------	---

Examples

```
AT+CIPSENDMODE=?
+CIPSENDMODE: (0-1)
```

```
OK
AT+CIPSENDMODE=1
OK
AT+CIPSENDMODE?
+CIPSENDMODE: 1
OK
```

SIMCom
Confidential

3 TCPIP Examples

3.1 Configure and Activate context

3.1.1 Network Environment

TCP/IP application is based on GPRS network. Please make sure that GPRS network is available before TCP/IP setup.

```
AT+CSQ
+CSQ: 23,0

OK
AT+CREG?
+CREG: 0,1

OK
```

3.1.2 Configure Context

If based on IPv4

```
AT+CGDCONT=1,"IP","CMNET"
OK
```

If based on IPv6

```
AT+CGDCONT=1,"IPV6","CMNET"
OK
```

//The CGDCONT IP_TYPE is set to IPV6 instead of IP

3.1.3 Activate context

If based on IPv4

```
AT+NETOPEN
```

```
OK
```

```
+NETOPEN: 0
```

```
AT+IPADDR
```

```
+IPADDR: 10.148.0.17
```

```
OK
```

If based on ipv6

```
AT+CSOCKETPN=1,6
```

```
//See ATC 2.2.16 to set the IP_TYPE to IPV6
```

```
OK
```

```
AT+NETOPEN
```

```
OK
```

```
+NETOPEN: 0
```

```
AT+IPADDR
```

```
+IPADDR: 2409:8960:1e64:94d8:1:0:3b3b:7118
```

```
//The queried IP address is an IPv6 address
```

```
OK
```

Other commands are used in the same way based on IPv4 or IPv6.

3.1.4 Deactivate Context

```
AT+NETCLOSE
```

```
OK
```

```
+NETCLOSE: 0
```

```
AT+IPADDR
```

```
+IP ERROR: Network not opened
```

```
ERROR
```

3.2 TCP Client

3.2.1 TCP Client Works in Direct Push Mode

```
//Set up TCP Client Connection
```

```
AT+NETOPEN
```

```
OK
```

```
+NETOPEN: 0
```

```
AT+CIPOPEN=1,"TCP","117.131.85.139",5253
```

```
// Set up a TCP connection, <link_num> is 1.
```

```
OK
```

```
Before using AT+CIPOPEN, host should activate  
PDP Context with AT+NETOPEN first.
```

```
+CIPOPEN: 1,0
```

```
//Send Data To Server
```

```
AT+CIPSEND=1,5
```

```
// Send data with fixed length
```

```
>HELLO
```

```
OK
```

```
+CIPSEND: 1,5,5
```

```
//Receive Data From Server
```

```
RCV FROM:117.131.85.139:5253
```

```
// Data from server directly output to COM
```

```
+IPD16
```

```
data from server
```

```
//Close TCP Connection
```

```
AT+CIPCLOSE=1
```

```
OK
```

```
+CIPCLOSE: 1,0
```

3.2.2 TCP Client Works in Buffer Access Mode

```
//Set up TCP Client Connection
```

```
AT+NETOPEN
```

OK

+NETOPEN: 0

AT+CIPRXGET=1

// Buffer access mode, get data by

OK

AT+CIPRXGET

AT+CIOPEN=1,"TCP","117.131.85.139",5253

OK

+CIOPEN: 1,0

//Send Data to Server

AT+CIPSEND=1,5

// Send data with fixed length

>hello

OK

+CIPSEND: 1,5,5

//Receive Data from Server

+CIPRXGET: 1,1

// URC to notify host of data from server

AT+CIPRXGET=4,1

// Query the length of data in the buffer of socket with

+CIPRXGET: 4,1,16

// <link_num>=1

OK

AT+CIPRXGET=2,1,5

// Get data in ASCII form

+CIPRXGET: 2,1,5,11

// Read 5 bytes data and left 11 bytes

Data1

OK

AT+CIPRXGET=3,1,5

// Get data in hex form

+CIPRXGET: 3,1,5,6

66726F6D20

OK

AT+CIPRXGET=4,1

// Read the length of unread data in buffer

+CIPRXGET: 4,1,6

OK

AT+CIPRXGET=2,2

// The connection identified by link_num=2 has not been established

+IP ERROR: No data

ERROR

AT+CIPRXGET=2,1

+CIPRXGET: 2,1,6,0

server

OK

AT+CIPRXGET=4,1

+CIPRXGET: 4,1,0

// All the data in buffer has been read, the rest_len is 0.

OK

//Close TCP Connection

AT+CIPCLOSE=1

OK

+CIPCLOSE: 1,0

3.2.3 TCP Client Works in Transparent Access Mode

//Set up TCP Client Connection

AT+CIPMODE=1

OK

AT+NETOPEN

OK

// Enter into transparent mode by
AT+CIPMODE=1

+NETOPEN: 0

AT+CIPOEPN=0,"TCP","117.131.85.139",5253

CONNECT 115200

// Only <link_num>=0 is allowed to operate with transparent mode.

//Send Data to Server

All data got from com port will be sent to internet directly

//Receive Data From Server

DATA FROM SERVERDATA FROM SERVER

OK

//All the received data from server will be output to com port directly

//sequence of +++ to quit transparent mode

AT+CIOPEN?

+CIOPEN: 0,"TCP","117.131.85.139",5253,-1

+CIOPEN: 1

+CIOPEN: 2

+CIOPEN: 3

```
OK
ATO //ATO to enter transparent mode again
CONNECT 115200
HELLO CLIENT
OK
```

```
//Close TCP Connection
```

```
AT+CIPCLOSE=0
```

```
OK
```

```
CLOSED
```

```
+CIPCLOSE: 0,0
```

3.3 UDP Client

3.3.1 UDP Client Works in Direct Push Mode

```
//Set up UDP Client Connection
```

```
AT+NETOPEN
```

```
OK
```

```
+NETOPEN: 0
```

```
AT+CIPOPEN=1,"UDP",,,5000
```

```
+CIPOPEN: 1,0
```

// When set a UDP connection, the remote IP address and port is not necessary, but the local port must be specified.

```
OK
```

```
//Send data to Server
```

```
AT+CIPSEND=1,"117.131.85.139",5254
```

```
>HELLOSERVER
```

```
OK <CTRL+Z>
```

// For UDP connection, when sending data, user must specify the remote IP address and port
//send data with changeable length, <CTRL+Z> to end

```
+CIPSEND: 1,11,11
```

```
AT+CIPSEND=1,5,"117.131.85.139",5254
```

//Send data with fixed length

```
>HELLO
```

OK

+CIPSEND: 1,5,5

//Receive Data From Server

RECV FROM:117.131.85.139:5254

//Data from server output to COM port directly

+IPD14

HELLO CLIENT

//Close UDP Connection

AT+CIPCLOSE=1

+CIPCLOSE: 1,0

OK

3.3.2 UDP Client Works in Buffer Access Mode

//Set up UDP Client Connection

AT+NETOPEN

OK

+NETOPEN: 0

AT+CIPRXGET=1

// Buffer access mode, get data by

OK

AT+CIPRXGET

AT+CIPOPEN=1,"UDP",,,5000

// When set a UDP connection, the remote IP

+CIPOPEN: 1,0

address and port is not necessary, but the local port must be specified.

OK

//Send Data to Server

AT+CIPSEND=1,,"117.131.85.139",5254

// For UDP connection, when sending data, user

>HELLOSERVER

must specify the remote IP address and port

OK <CTRL+Z>

// Send data with changeable length, <CTRL+Z>

to end

+CIPSEND: 1,11,11

AT+CIPSEND=1,5,"117.131.85.139",5254

// Send data with fixed length

>HELLO

OK

+CIPSEND: 1,5,5

```
//Receive Data From Server
+CIPRXGET: 1,1 // URC to notify host of data from server
AT+CIPRXGET=4,1 // Query the length of data in the buffer of socket
+CIPRXGET: 4,1,16 with <link_num>=1

OK
AT+CIPRXGET=2,1,5 // Get data in ASCII form
+CIPRXGET: 2,1,5,11
data

OK
AT+CIPRXGET=3,1,5 // Get data in hex form
+CIPRXGET: 3,1,5,6
66726F6D20

OK
AT+CIPRXGET=4,1 // Read the length of unread data in buffer
+CIPRXGET: 4,1,6

OK
AT+CIPRXGET=2,2 // The connection identified by link_num=2 has not
+IP ERROR: No data been established

ERROR
AT+CIPRXGET=2,1
+CIPRXGET: 2,1,6,0
server

OK
AT+CIPRXGET=4,1 // All the data in buffer has been read, the rest_len
+CIPRXGET: 4,1,0 is 0.

OK
```

```
//Close UDP Connection
AT+CIPCLOSE=1
+CIPCLOSE: 1,0

OK
```

3.3.3 UDP Client Works in Transparent Access Mode

//Set up UDP Client Connection

AT+CIPMODE=1

OK

AT+NETOPEN

OK

+NETOPEN: 0

AT+CIPOPEN=0,"UDP","117.131.85.139",5254,5000 //Only <link_num>=0 is allowed to operate with transparent mode.

CONNECT 115200

//Send Data to Server

All data got from com port will be sent to internet directly

//Receive Data From Server

HELLO CLIENT

//Data from server output to COM port directly

HELLO CLIENT

OK

// Sequence of +++ to quit transparent mode

AT+CIPOPEN?

+CIPOPEN: 0,"UDP","117.131.85.139",5254,-1

+CIPOPEN: 1

+CIPOPEN: 2

+CIPOPEN: 3

OK

//Close UDP Connection

AT+CIPCLOSE=0

+CIPCLOSE: 0,0

OK

3.4 TCP Server

3.4.1 Transparent Mode

AT+CIPMODE=1

OK

AT+NETOPEN

OK

```
+NETOPEN: 0
AT+SERVERSTART=10000,0 //Only <server_index>=0 is allowed to operate
OK //with transparent mode.
+CLIENT: 0,0,192.168.108.5:57202 //Only <link_num> 0 can be used for transparent
CONNECT 115200 //mode operation.

OK // Sequence of +++ to quit data mode
AT+CIPCLOSE=0 // Close client connection
OK

CLOSED

+CIPCLOSE: 0,0
AT+SERVERSTOP=0 // Close server socket
+SERVERSTOP: 0,0

OK
```

3.4.2 Non-Transparent Mode

```
AT+NETOPEN
OK

+NETOPEN: 0
AT+SERVERSTART=10000,0 //Only <server_index>=0 is allowed to operate
OK //with transparent mode.
AT+SERVERSTART=10001,1
OK
AT+SERVERSTART=10002,2
OK
AT+SERVERSTART=10003,3
OK
+CLIENT: 0,0,192.168.108.5:57202 //If a socket is accepted, the following URC will be
//reported:
AT+CIPOPEN? //User can use AT+CIPOPEN? to check the
+CIPOPEN: 0,"TCP","192.168.108.5",57202,1 //accepted socket
+CIPOPEN: 1 //Last parameter of 1 indicates this is an accepted
+CIPOPEN: 2 //socket, this server index is 1
+CIPOPEN: 3

OK
AT+CIPSEND=0,5 // Only supports fixed-length to send
>HELLO
OK
```

```
+CIPSEND: 0,5,5
AT+SERVERSTOP=0 // If unspecified, it will close 0 channel
+SERVERSTOP: 0,0

OK
AT+SERVERSTOP=1
+SERVERSTOP: 1,0

OK
AT+SERVERSTOP=2
+SERVERSTOP: 2,0

OK
AT+SERVERSTOP=3
+SERVERSTOP: 3,0

OK
AT+NETCLOSE
OK

+NETCLOSE: 0
```

3.4.3 Query Connection Status

```
AT+CIPOPEN=1,"TCP","117.131.85.139",5253
OK

+CIPOPEN: 1,0
AT+CIPOPEN? // Query the current state of all sockets
+CIPOPEN: 0
+CIPOPEN: 1,"TCP","117.131.85.139",5253,-1
+CIPOPEN: 2
+CIPOPEN: 3

OK
AT+CIPCLOSE?
+CIPCLOSE: 0,1,0,0

OK
AT+CIPCLOSE=1
OK

+CIPCLOSE: 1,0
AT+CIPCLOSE?
```

+CIPCLOSE: 0,0,0,0

OK

NOTE

The appeal sample server is for demonstration purposes only, not for commercial purpose.

SIMCom
Confidential

4 Appendix

4.1 Summary of Error Codes

In case of any issues while using the commands: **AT+CIPRXGET**, error information may be reported as +IP ERROR: <err_info>.

The fourth parameter <errMode> of **AT+CIPCCFG** (TODO) is used to determine how <err_info> is displayed.

If <errMode> is set to 0, the <err_info> is displayed with numeric value.

If <errMode> is set to 1, the <err_info> is displayed with string value.

The default is displayed with string value.

The following list is the description of the <err_info>.

Numeric Value	String Value
0	Connection time out
1	Bind port failed
2	Port overflow
3	Create socket failed
4	Network is already opened
5	Network is already closed
6	No clients connected
7	No active client
8	Network not opened
9	Client index overflow
10	Connection is already created
11	Connection is not created
12	Invalid parameter
13	Operation not supported
14	DNS query failed
15	TCP busy
16	Net close failed for socket opened
17	Sending time out
18	Sending failure for network error
19	Open failure for network error
20	Server is already listening

21	Operation failed
22	No data

In case of any issues while using the commands: **AT+NETOPEN**, **AT+NETCLOSE**, **AT+CIPOPEN**, **AT+CIPSEND**, **AT+CIPCLOSE**, **AT+SERVERSTART**, **AT+SERVERSTOP**, the wrong number will be reported.

The following list is the description of the <err>.

<err>	Description of <err>
0	operation succeeded
1	Network failure
2	Network not opened
3	Wrong parameter
4	Operation not supported
5	Failed to create socket
6	Failed to bind socket
7	TCP server is already listening
8	Busy
9	Sockets opened
10	Timeout
11	DNS parse failed for AT+CIPOPEN
12	Unknown error

4.2 Unsolicited Result Codes

Information	Description
+CIPEVENT: NETWORK CLOSED UNEXPECTEDLY	Network is closed for network error(Out of service, etc.). When this event happens, user's application needs to check and close all opened sockets, and then uses AT+NETCLOSE to release the network library if AT+NETOPEN? Shows the network library is still opened.
+IPCLOSE: <client_index>,<close_reason>	Socket is closed passively. <client_index> is the link number. <close_reason>: 0 - Closed by local, active 1 - Closed by remote, passive 2 - Closed for sending timeout or DTR off

**+CLIENT: <
link_num>,<server_index>,<client_IP>:<port>**

TCP server accepted a new socket client, the index is <link_num>, and the TCP server index is <server_index>. The peer IP address is <client_IP>, the peer port is <port>.

SIMCom
Confidential