A76XX Series_TCP/IP_Application Note

LTE Module
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/download/list-863-en.html

For technical support, or to report documentation errors, please visit:
https://www.simcom.com/ask/or email to: support@simcom.com

Copyright © 2021SIMCom Wireless Solutions Limited All Rights Reserved.
About Document

Version History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Chapter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.00</td>
<td>2020.6.19</td>
<td></td>
<td>New version</td>
</tr>
<tr>
<td>V1.01</td>
<td>2020.8.25</td>
<td>3.2.3 TCP Client</td>
<td>Modify the title</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Works in Buffer Access</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mode</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2021.02.03</td>
<td>All</td>
<td>Add support on A7678 Series</td>
</tr>
<tr>
<td>V1.02</td>
<td>2021.11.08</td>
<td>Scope</td>
<td>Scope description is updated</td>
</tr>
</tbody>
</table>
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 A1803S Series, A1603 Series, A1601 Series and A1802 Series.
4.2 Unsolicited Result Codes........................................................................................................................................ 25
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


1.3 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) or facsimile DCE (FAX modem, FAX board);

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:
**SIM Card Status:**
Execute AT+CPIN?/response is +CPIN READY: means SIM Card Status is normal, Reboot the module or check SIM card status if AT+CPIN? fails to identify SIM card or in 20s.

**Signal Quality:**
Execute AT+CSQ to query signal quality. If the result is equals to 99 please check SIM card status or reboot the module.

**Function Quality:**
This command is used to query the function level <fun> in AT+CFUN. If the function level is 0.1 means that you are not at this level, please set it to 1.0.

**CS Service:**
If the value of AT+CREG equals to 3, it means that the module has registered on CS domain service. Reboot the module or check SIM card status if AT+CREG fails to identify the SIM card on CS domain.

**PS Service:**
If the value of AT+CGREG/AT+CREG equals to 1, it means that the module has registered on PS domain service.

**UE System Information:**
If (System Mode) = "NO SERVICE", it means network status has some problem.

**Set PDP Context:**
The set command specifies PDP context parameter values for a PDP context identified by the (local) context identification parameter <sid>. The number of PDP contexts that may be in a defined state at any time is given by the range returned by the test command.

**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. When you want to use TCP, you must set the mode AT+CIPMODE before AT+NETOPEN.

**Activate a PDP Context:**
AT+NPDP is used to start or activate a PDP context. You must execute AT+NPDP before any TCP/UDP related operations.

**Establish a connection with TCP server and UDP server:**
The maximum of the connections is 10. When AT+CIPMOD=1 is set, the <link_num> is restricted to be only 0, and it will report a URC as CONNECT 1000.

**Send Data:**
AT+CIPSEND is used to send data to remote side. The <ESC> is used to cancel the sending.

**Switch to Command Mode:**
Injection of escape causes the AT to cancel the data flow over the AT interface and switch to Command Mode. This allows the entry of AT commands while maintaining the data connection to the remote device.

**Close a PDP Context:**
AT+NPDP is used to close a TCP or UDP Socket.

**Data received:**
You can receive all or part of the data you receive in different ways:

**Establish Connection:**
(establish a connection with TCP server and UDP server, the maximum of the connections is 10)

- Received data: AT+CIPRXGET is used to receive data from the remote side. Single <ESC> is used to cancel the sending.
- Close Socket: AT+NPDP is used to close a TCP or UDP Socket.
NOTE: If you need to use the TCP server, you'll need special SIM cards.

**SIM Card Status:**
- Execute AT+CPIN? if response is +CPIN:READY, means SIM Card Status is normal. Reboot the module if check SIM card status if AT+CPIN? Falls to identify SIM card in 30s.

**Signal quality:**
- Execute AT+CSQ to query signal quality. If RSSI is equal to 99 please check SIM card status or reboot the module.

**Function quality:**
- This command is used to query the function level <fun> in ME. The “1” level is the highest level of power. If you are not at this level, please set it to this level.

**CS Service:**
- If the <stat> of AT+CREG? is equal to 1 means that the module has registered on CS domain service. Reboot the module if it fails to registered on CS domain.

**PS Service:**
- If the <stat> of AT+CREG? is equal to 1 means that the module has registered on PS domain service.

**UE system information:**
- If the <System Mode> is "NO SERVICE", it means network status has some problem.

**Specifies PDP context:**
- The set command specifies PDP context parameter values for a PDP context identified by the (local) context identification parameter <cd>. The number of PDP contexts that may be in a defined state at the same time is given by the range returned by the <range> command.

**Set TCP/IP Application Mode:**
- AT+CPMMODE is used to select transparent (data mode) or non-transparent mode (command mode). The transparent mode mode when you want to use transparent mode to transmit data, you should set AT+CPMMODE=1 before AT+NETOPEN.

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

**Open Connection:**
- Establish a connection with TCP server and UDP server. The maximum of the connections is 10. When AT+CPMMODE=1 is set, the <link_num> is restricted to be only 0.

**Switch to command mode:**
- The ‘+++’ character sequence causes the AT to cancel the data flow over the AT interface and switch to Command Mode. This allows you to enter AT commands while maintaining the data connection to the remote device.

**Close Socket:**
- AT+CPFCLOSE is used to close a TCP or UDP socket.

**Deactivate a PDP Context:**
- AT+NETCLOSE is used to stop service by deactivating a PDP context. It can also close all the opened socket connections when you didn’t close these connections by AT+CPFCLOSE.
1.5 Error Handling

1.5.1 Executing FTP(S) 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 to correct them.

1.5.4 Description of Data Access Mode

```
Access Mode
     [ Transparent Mode
       (Data Mode)
     ]
     [ Non-Transparent Mode
       (Command Mode)
     ]
     [ Push Mode
     ]
     [ Buffer 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 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 TCPIP Services AT

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT+NETOPEN</td>
<td>Start Socket Service</td>
</tr>
<tr>
<td>AT+NETCLOSE</td>
<td>Stop Socket Service</td>
</tr>
<tr>
<td>AT+CIPOPEN</td>
<td>Establish Connection in Multi-Socket Mode</td>
</tr>
<tr>
<td>AT+CIPSEND</td>
<td>Send data through TCP or UDP Connection</td>
</tr>
<tr>
<td>AT+CIPRXGET</td>
<td>Set the Mode to Retrieve Data</td>
</tr>
<tr>
<td>AT+CIPCLOSE</td>
<td>Close TCP or UDP Socket</td>
</tr>
<tr>
<td>AT+IPADDR</td>
<td>Inquire Socket PDP address</td>
</tr>
<tr>
<td>AT+CIPHEAD</td>
<td>Add an IP Header When Receiving Data</td>
</tr>
<tr>
<td>AT+CIPSRSpec</td>
<td>ShowRemoteIP Address and Port</td>
</tr>
<tr>
<td>AT+CIPMODE</td>
<td>Set TCP/IP Application Mode</td>
</tr>
<tr>
<td>AT+CIPSENDMODE</td>
<td>Set Sending Mode</td>
</tr>
<tr>
<td>AT+CIPTIMEOUT</td>
<td>Set TCP/IP Timeout Value</td>
</tr>
<tr>
<td>AT+CIPCCFG</td>
<td>Configure Parameters of Socket</td>
</tr>
<tr>
<td>AT+SERVERSTART</td>
<td>Startup TCP Server</td>
</tr>
<tr>
<td>AT+SERVERSTOP</td>
<td>Stop TCP Server</td>
</tr>
<tr>
<td>AT+CIPACK</td>
<td>Query TCP Connection Data Transmitting Status</td>
</tr>
<tr>
<td>AT+CDNSGIP</td>
<td>Query the IP Address of Given Domain Name</td>
</tr>
</tbody>
</table>
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
AT+CGREG?
+CGREG: 0,1
OK
```

3.1.2 Configure Context

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

3.1.3 Activate context
3.1.4 Deactivate Context

AT+NETCLOSE
OK

+NETCLOSE: 0
AT+IPADDR
+IPADDR: 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
OK

+CIPOPEN: 1,0

// Set up a TCP connection, <link_num> is 1.
Before using AT+CIPOPEN, host should activate PDPContext with AT+NETOPEN first.

// Send Data To Server
AT+CIPSEND=1,5

// send data with fixed length
3.2.2 TCP Client Works in Buffer Access Mode

//Set up TCP Client Connection
AT+NETOPEN
OK

+NETOPEN: 0
AT+CIPRXGET=1
OK
AT+CIPOPEN=1,"TCP","117.131.85.139",5253
OK

+CIPOPEN: 1,0

//Send Data to Server
AT+CIPSEND=1,5
>hello
OK

+CIPSEND: 1,5,5

//Receive Data from Server
+CIPRXGET: 1,1
AT+CIPRXGET=4,1
+CIPRXGET: 4,1,16
OK
AT+CIPRXGET=2,1,5
+CIPRXGET: 2,1,5,11data
OK
AT+CIPRXGET=3,1,5
+CIPRXGET: 3,1,5,6
66726F6D20
OK
AT+CIPRXGET=4,1
+CIPRXGET: 4,1,6
OK
AT+CIPRXGET=2,2
+IP ERROR:  No data
ERROR
AT+CIPRXGET=2,1
+CIPRXGET: 2,1,6,0
+IP ERROR:  No data
OK
AT+CIPRXGET=4,1
+CIPRXGET: 4,1,0
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  // Enter into transparent mode by at+cipmode=1
AT+NETOPEN
OK

+NETOPEN: 0
AT+CIPOPEN=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+CIPOPEN?
+CIPOPEN: 0,"TCP","117.131.85.139",5253,-1
+CIPOPEN: 1
+CIPOPEN: 2
+CIPOPEN: 3
+CIPOPEN: 4
+CIPOPEN: 5
+CIPOPEN: 6
+CIPOPEN: 7
+CIPOPEN: 8
+CIPOPEN: 9
OK

ATO
CONNECT 115200
HELLO CLIENT
OK

//ATO to enter transparent mode again

//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
OK

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

// Send data to Server
AT+CIPSEND=1,,"117.131.85.139",5254
>HELLO SERVER
OK <CTRL+Z>

+CIPSEND: 1,11,11
AT+CIPSEND=1,5,"117.131.85.139",5254
>HELLO
OK

+CIPSEND: 1,5,5

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

// send data with fixed length

// Receive Data From Server
RECV FROM:117.131.85.139:5254
+IPD14
HELLO CLIENT

// data from server output to COM port directly

// 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
  OK
  AT+CIPOPEN=1,"UDP",,5000
  +CIPOPEN: 1,0
  OK
  ```
  // buffer access mode, get data by AT+CIPRXGET
  // when set a UDP connection, the remote IP address and port is not necessary, but the local port must be specified.

- **Send Data to Server**
  ```
  AT+CIPSEND=1,"117.131.85.139",5254
  >HELLOSERVER
  OK <CTRL+Z>
  +CIPSEND: 1,11,11
  AT+CIPSEND=1,5,"117.131.85.139",5254
  >HELLO
  OK
  +CIPSEND: 1,5,5
  ```
  // for UDP connection, when sending data, user must specify the remote IP address and port
  // send data with changeable length, <CTRL+Z> to end
  // send data with fixed length

- **Receive Data From Server**
  ```
  +CIPRXGET: 1,1
  AT+CIPRXGET=4,1
  +CIPRXGET: 4,1,16
  OK
  AT+CIPRXGET=2,1,5
  +CIPRXGET: 2,1,5,11
  data
  OK
  AT+CIPRXGET=3,1,5
  +CIPRXGET: 3,1,5,6
  66726F6D20
  ```
  // URC to notify host of data from server
  // query the length of data in the buffer of socket with <link_num>=1
  // get data in ASCII form
  // get data in hex form
OK
AT+CIPRXGET=4,1
+CIPRXGET: 4,1,6
// read the length of unread data in buffer

OK
AT+CIPRXGET=2,2
+CIPRXGET: 2,1,6,0
// the connection identified by link_num=2 has not been established

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 UDP Connection
AT+CIPCLOSE=1
OK
+CIPCLOSE: 1,0

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
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
### 3.4 TCP Server

#### 3.4.1 Transparent Mode

```plaintext
AT+CIPMODE=1
OK
AT+NETOPEN
OK
+NETOPEN: 0
AT+SERVERSTART=8080, 0
OK
+CLIENT: 0,0,192.168.108.5:57202
CONNECT 115200

OK
AT+CIPCLOSE=0
OK
CLOSED
```

**Note:**
- `+NETOPEN: 0` is allowed to operate with transparent mode.
- `+CLIENT: 0,0,192.168.108.5:57202` can be used for transparent mode operation.
- Sequence of `+++` to quit data mode
- Close client connection
3.4.2 Non-Transparent Mode

AT+NETOPEN
OK

+NETOPEN: 0
AT+SERVERSTART=8080, 0
OK
AT+SERVERSTART=9090, 1
OK
AT+SERVERSTART=7070, 2
OK
AT+SERVERSTART=6060, 3
OK
+CLIENT: 0,0,192.168.108.5:57202

AT+CIPOPEN?
+CIPOPEN: 0,"TCP","192.168.108.5",57202,1
+CIPOPEN: 1
+CIPOPEN: 2
+CIPOPEN: 3
+CIPOPEN: 4
+CIPOPEN: 5
+CIPOPEN: 6
+CIPOPEN: 7
+CIPOPEN: 8
+CIPOPEN: 9

OK
AT+CIPSEND=0,5
>HELLO
OK

+CIPSEND: 0,5,5
AT+SERVERSTOP=0
+SERVERSTOP: 0,0
OK
AT+SERVERSTOP=1
+SERVERSTOP: 1,0

// close server socket

// only <server_index>=0 is allowed to operate with transparent mode.

//If a socket is accepted, the following URC will be reported:

//User can use AT+CIPOPEN? to check the accepted socket

//last parameter of 1 indicates this is an accepted socket, this server index is 1

// only supports fixed-length to send

// if unspecified, it will close 0 channel
OK
AT+SERVERSTOP=2
+SERVERSTOP: 2,0

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

OK
AT+NETCLOS
OK
+NETCLOSE: 0

3.4.3 Query Connection Status

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

+CIPOPEN: 1,0
AT+CIPOPEN?
+CIPOPEN: 0
+CIPOPEN: 1,"TCP","117.131.85.139",5253,-1
+CIPOPEN: 2
+CIPOPEN: 3
+CIPOPEN: 4
+CIPOPEN: 5
+CIPOPEN: 6
+CIPOPEN: 7
+CIPOPEN: 8
+CIPOPEN: 9

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

OK
AT+CIPCLOSE=1
OK

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

OK
4 Appendix

4.1 Summary of Error Codes

When you use these commands: `AT+CIPACKAT+CIPRXGET`, if something goes wrong, they 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>`.

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

When you use these commands: AT+NETOPEN, AT+NETCLOSE, AT+CIPOPEN, AT+CIPSEND, AT+CIPCLOSE, AT+SERVERSTART, AT+SERVERSTOP. If something goes wrong, they will report the wrong number.

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

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

### 4.2 Unsolicited Result Codes

<table>
<thead>
<tr>
<th>Information</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+CIPEVENT: NETWORK CLOSED</td>
<td>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.</td>
</tr>
<tr>
<td>CLOSED UNEXPECTEDLY</td>
<td></td>
</tr>
</tbody>
</table>
+IPCLOSE:

Socket is closed passively.

<table>
<thead>
<tr>
<th>&lt;client_index&gt;,&lt;close_reason&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;client_index&gt; is the link number.</td>
</tr>
<tr>
<td>&lt;close_reason&gt;:</td>
</tr>
<tr>
<td>0 - Closed by local, active</td>
</tr>
<tr>
<td>1 - Closed by remote, passive</td>
</tr>
<tr>
<td>2 - Closed for sending timeout or DTR off</td>
</tr>
</tbody>
</table>

+CLIENT:

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

<table>
<thead>
<tr>
<th>&lt;link_num&gt;,&lt;server_index&gt;,&lt;client_IP&gt;:&lt;port&gt;</th>
</tr>
</thead>
</table>

<client_index> is the link number.