SIM8200 Series_TCPIP _Application Note

5G Module

SIMCom Wireless Solutions Limited
Building B, SIM Technology Building, No.633, Jinzhong Road
Changning District, Shanghai P.R. China
Tel: 86-21-31575100
support@simcom.com
www.simcom.com

Document Title:	SIM8200 Series_TCPIP_Application Note
Version:	1.00
Date:	2020.8.17
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

SIMCom Wireless Solutions Limited
Building B, SIM Technology Building, No.633 Jinzhong 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

Version History

Version	Date	Owner	What is new
V1.00	2020.8.17	Yulong.Li	First Release

Contents

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] SIM8200 Series_AT Command Manual

2 TCPIP Introduction

TCPIP is used to setup connections between clients and servers, which are used for TCP/UDP clients communicating with servers.

2.1 Characteristic

- **Support connecting TCP/UDP servers;**
  - **TCP connections**
    Module works as TCP clients. It communicates with TCP servers by TCP connections.
  - **UDP connections**
    Module works as UDP clients. It communicates with UDP servers.
- **Support accepting TCP clients;**
  - **TCP servers**
    Module works as TCP servers. It listens TCP clients accept request and communicates with TCP clients.
- **Support multiple data transmission mode;**
  - **Direct Push Mode**
    Host data will be sent to internal protocol stack and forwarded to air interface. Data received from air interface will be transmitted to internal protocol stack and forwarded to COM ports.
  - **Buffer Access Mode**
    Host data will be sent to internal protocol stack and forwarded to air interface. Data received from air interface will be saved into local buffers. Host could retrieve buffer data by AT commands.
  - **Transparent Access Mode**
    Host data will be directly sent to air interface. Data received from air interface will be directly sent to COM ports.

2.2 TCPIP Commands Process

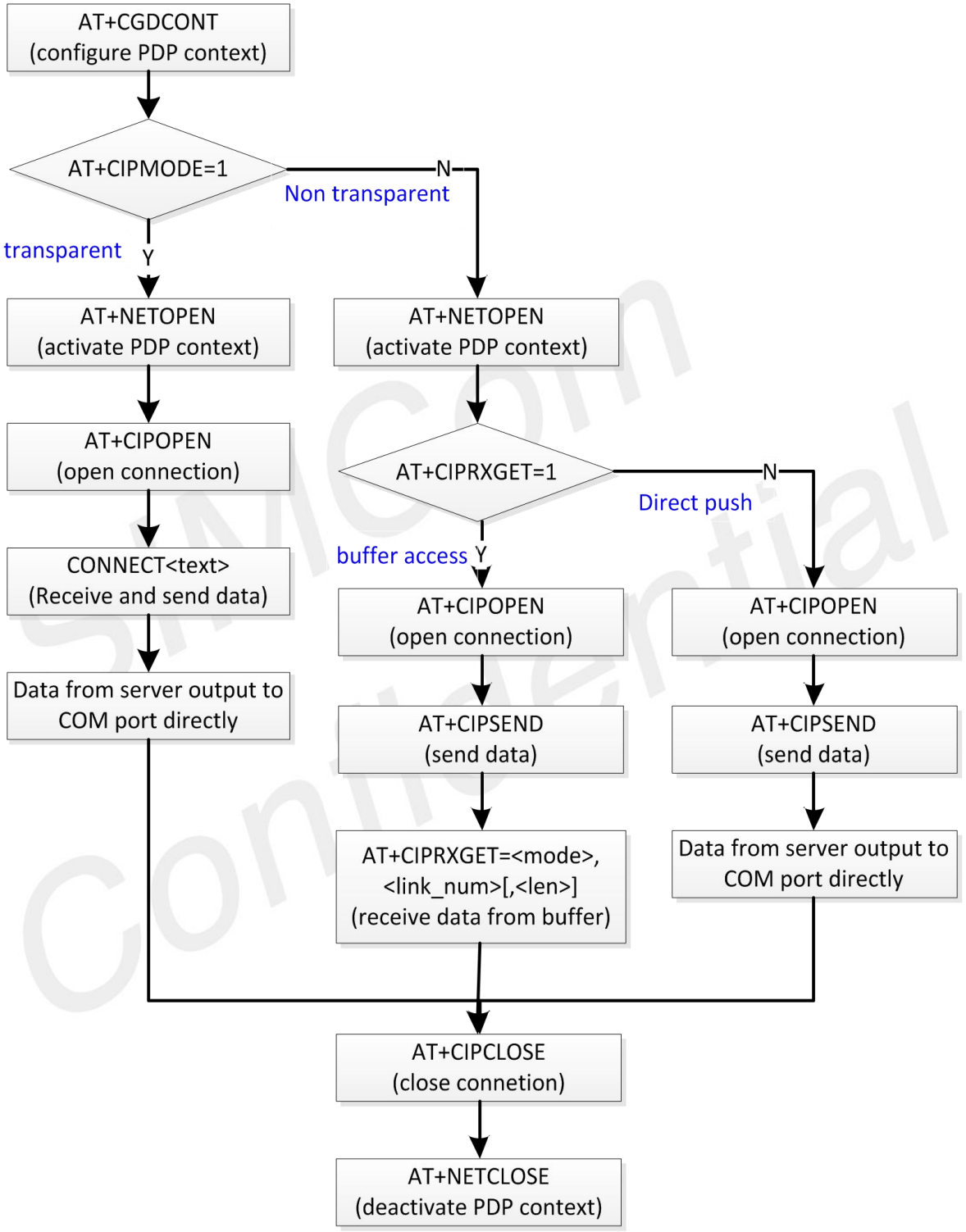

Figure 1: Flow Chart of Using TCP/IP Commands

3 AT Commands for TCPIP

Command	Description
AT+NETOPEN	Start TCPIP Service
AT+NETCLOSE	Stop TCPIP Service
AT+CIPOPEN	Setup TCP/UDP Client Socket Connections
AT+CIPCLOSE	Destroy TCP/UDP Client Socket Connections
AT+CIPSEND	Send TCP/UDP Data
AT+CIPRXGET	Retrieve TCP/UDP Buffer Data
AT+IPADDR	Get IP Address of PDP Context
AT+CIPHEAD	Set Length Domain of Reporting Head of Data
AT+CIPSRIP	Set IP Domain of Reporting Head of Data
AT+CIPMODE	Enable/Disable Transparent Access Mode
AT+CIPSENDMODE	Set Reporting Mode of CIPSEND result
AT+CIPTIMEOUT	Set Timeout Value of NETOPEN, CIPOPEN and CIPSEND
AT+CIPCCFG	Configure Socket Context
AT+SERVERSTART	Setup TCP Server Socket Connections
AT+SERVERSTOP	Destroy TCP Server Socket Connections
AT+CIPACK	Get Statistics Information of Data Communication
AT+CDNSGIP	Analysis IP Address from Domain Name
AT+CDNSGHNAME	Analysis Domain Name from IP Address
AT+CIPDNSSET	Configure DNS Context

For detail information, please refer to “SIM8200 Series_AT Command Manual”.

4 Bearer Configuration

Make sure that module has been attached to network successfully and APN information is configured.

```
AT+CPIN?
+CPIN: READY                                    // Check Status of SIM Card

OK
AT+CSQ
+CSQ: 27,99                                     // Check RF Signal

OK
AT+CGREG?
+CGREG: 0,1                                     // Check Status of PS Service

OK
AT+COPS?
+COPS: 0,0,"CHN-CT",9                           // Check Information of Operator

OK
AT+CPSI?
+CPSI:
LTE,Online,460-01,0x230A,175499523,318,EUT      // Check Information of Network
RAN-BAND3,1650,5,5,-10,-92,-62,12
+CPSI: NR5G,139,636654,-11,-92,12.6

OK
AT+CGDCONT=1, "IP", "CMNET"                     // Set PDP context Parameters
OK
AT+CGDCONT?
+CGDCONT:
1,"IPV4","CMNET","0.0.0.0.0.0.0.0.0.0.0.0.0.0.  // Check Information of PDP Context
0",0,0,0,0

OK
```

5 TCPIP Example

5.1 TCP Client

5.1.1 TCP Client Works in Direct Push Mode

5.1.1.1 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.
                                           // Before using AT+CIPOPEN, host should
                                           // activate PDP ontext with AT+NETOPEN first.
OK

+CIPOPEN: 1,0
```

5.1.1.2 Send data to Server

```
AT+CIPSEND=1,5                             // send data with fixed length
>HELLO
OK

+CIPSEND: 1,5,5

AT+CIPSEND=1,                              // send data with changeable length, <CTRL+Z>
                                           // to end
>HELLOWORLD<CTRL+Z>
OK

+CIPSEND: 1,10,10
```

5.1.1.3 Receive Data From Server

```
RECV FROM:117.131.85.139:5253          // data from server directly output to COM
+IPD16
data from server
```

5.1.1.4 Close TCP Connection

```
AT+CIPCLOSE=1
OK

+CIPCLOSE: 1,0
```

5.1.2 TCP Client Works in Buffer Access Mode

5.1.2.1 Set up TCP Client connection

```
AT+NETOPEN
OK

+NETOPEN: 0
AT+CIPRXGET=1                          // buffer access mode, get data by AT+CIPRXGET
OK
AT+CIPOPEN=1,"TCP","117.131.85.139",5253   // set up a TCP connection, <link_num> is 1
OK

+CIPOPEN: 1,0
```

5.1.2.2 Send Data to Server

```
AT+CIPSEND=1,5                         // send data with fixed length
>hello
OK

+CIPSEND: 1,5,5
```

5.1.2.3 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 <link_num>=1
+CIPRXGET: 4,1,16

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 been established
+IP ERROR: No data

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 is 0.
+CIPRXGET: 4,1,0

OK
```

5.1.2.4 Close TCP Connection

```
AT+CIPCLOSE=1
OK
```

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

5.1.3 TCP Client Works in Transparent Access Mode

5.1.3.1 Set up TCP Client Connection

```
AT+CIPMODE=1                              // Enter into transparent mode by at+cipmode=1
OK
AT+NETOPEN
OK

+NETOPEN: 0
AT+CIPOPEN=0,"TCP","117.131.85.139",5253  // only <link_num>=0 is allowed to operate with
                                          // transparent mode.
CONNECT 115200
```

5.1.3.2 Send Data to Server

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

5.1.3.3 Receive Data From Server

```
DATA FROM SERVERDATA FROM SERVER          // all the received data from server will be output to
                                          // com port directly
                                          // sequence of +++ to quit transparent mode
OK
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                                    // ATO to enter transparent mode again
CONNECT 115200
HELLO CLIENT
```

5.1.3.4 Close TCP Connection

```
AT+CIPCLOSE=0
OK

CLOSED

+CIPCLOSE: 0,0
```

5.2 UDP Client

5.2.1 UDP Client Works in Direct Push Mode

5.2.1.1 Set up UDP Client Connection

```
AT+NETOPEN
OK

+NETOPEN: 0

                                       // when set a UDP connection, the remote IP
AT+CIPOPEN=1,"UDP",,,5000              // address and port is not necessary, but the local
                                       // port must be specified.
+CIPOPEN: 1,0

OK
```

5.2.1.2 Send data to Server

```
                                       // for UDP connection, when sending data, user must
AT+CIPSEND=1,,"117.131.85.139",5254    // specify the remote IP address and port. Sending data
                                       // with changeable length
>HELLOSERVER
```

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

5.2.1.3 Receive Data From Server

```
RECV FROM:117.131.85.139:5254                    // data from server output to COM port directly
+IPD14
HELLO CLIENT
```

5.2.1.4 Close UDP Connection

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

OK
```

5.2.2 UDP Client Works in Buffer Access Mode

5.2.2.1 Set up UDP Client connection

```
AT+NETOPEN
OK

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

OK
```

5.2.2.2 Send Data to Server

```
                                        // for UDP connection, when sending data, user must
AT+CIPSEND=1,,"117.131.85.139",5254     // specify the remote IP address and port. Sending data
                                        // with changeable length, <CTRL+Z> to end
>HELLOSERVER
OK     <CTRL+Z>

+CIPSEND: 1,11,11
AT+CIPSEND=1,5,"117.131.85.139",5254    // send data with fixed length
>HELLO
OK

+CIPSEND: 1,5,5
```

5.2.2.3 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 <link_num>=1
+CIPRXGET: 4,1,16

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 been established
+IP ERROR: No data

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 is 0.
+CIPRXGET: 4,1,0

OK
```

5.2.2.4 Close UDP Connection

```
AT+CIPCLOSE=1
OK

+CIPCLOSE: 1,0
```

5.2.3 UDP Client Works in Transparent Access Mode

5.2.3.1 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
```

5.2.3.2 Send Data to Server

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

5.2.3.3 Receive Data From Server

```
HELLO CLIENT                                    // data from server output to COM port directly
HELLO CLIENT                                    // data from server output to COM port directly
OK                                              // sequence of +++ to quit transparent mode
AT+CIPOPEN?
+CIPOPEN: 0,"UDP","117.131.85.139",5254,-1
+CIPOPEN: 1
+CIPOPEN: 2
+CIPOPEN: 3
+CIPOPEN: 4
+CIPOPEN: 5
+CIPOPEN: 6
+CIPOPEN: 7
+CIPOPEN: 8
+CIPOPEN: 9

OK
```

5.2.3.4 Close UDP Connection

```
AT+CIPCLOSE=0
CLOSED

+CIPCLOSE: 0,0

OK
```

5.3 TCP Server

5.3.1 Transparent Mode

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

+NETOPEN: 0
```

```
AT+SERVERSTART=8080, 0                  // only <server_index>=0 is allowed to operate with
                                        // transparent mode.
OK

+CLIENT: 0,0,192.168.108.5:57202        // only <link_num> 0 can be used for transparent mode
                                        // operation.
CONNECT 115200
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
```

5.3.2 Non-Transparent Mode

Module supports 4 sockets to listen.

```
AT+NETOPEN
OK

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

+CLIENT: 0,1,192.168.108.5:57202        // If a socket is accepted, this URC will be reported

AT+CIPOPEN?                             // User can use AT+CIPOPEN? to check the
                                        // accepted socket
+CIPOPEN: 0,"TCP","192.168.108.5",57202,1
                                        // last parameter of 1 indicates this is an accepted
                                        // socket, this server index is 1
```

```
+CIPOPEN: 1
+CIPOPEN: 2
+CIPOPEN: 3
+CIPOPEN: 4
+CIPOPEN: 5
+CIPOPEN: 6
+CIPOPEN: 7
+CIPOPEN: 8
+CIPOPEN: 9

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

+CIPSEND: 0,5,5
AT+CIPCLOSE=0
OK

+CIPCLOSE: 0,0
AT+SERVERSTOP=0
+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
```

5.4 Extended Information

5.4.1 UDP Data Receiving Rules

One UDP connection could handle no more 1500 Bytes with once receiving. Please make sure that UDP server sends acceptable data packets, whose length is greater than 0 and less than 1500 Bytes.

5.5 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
+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
```