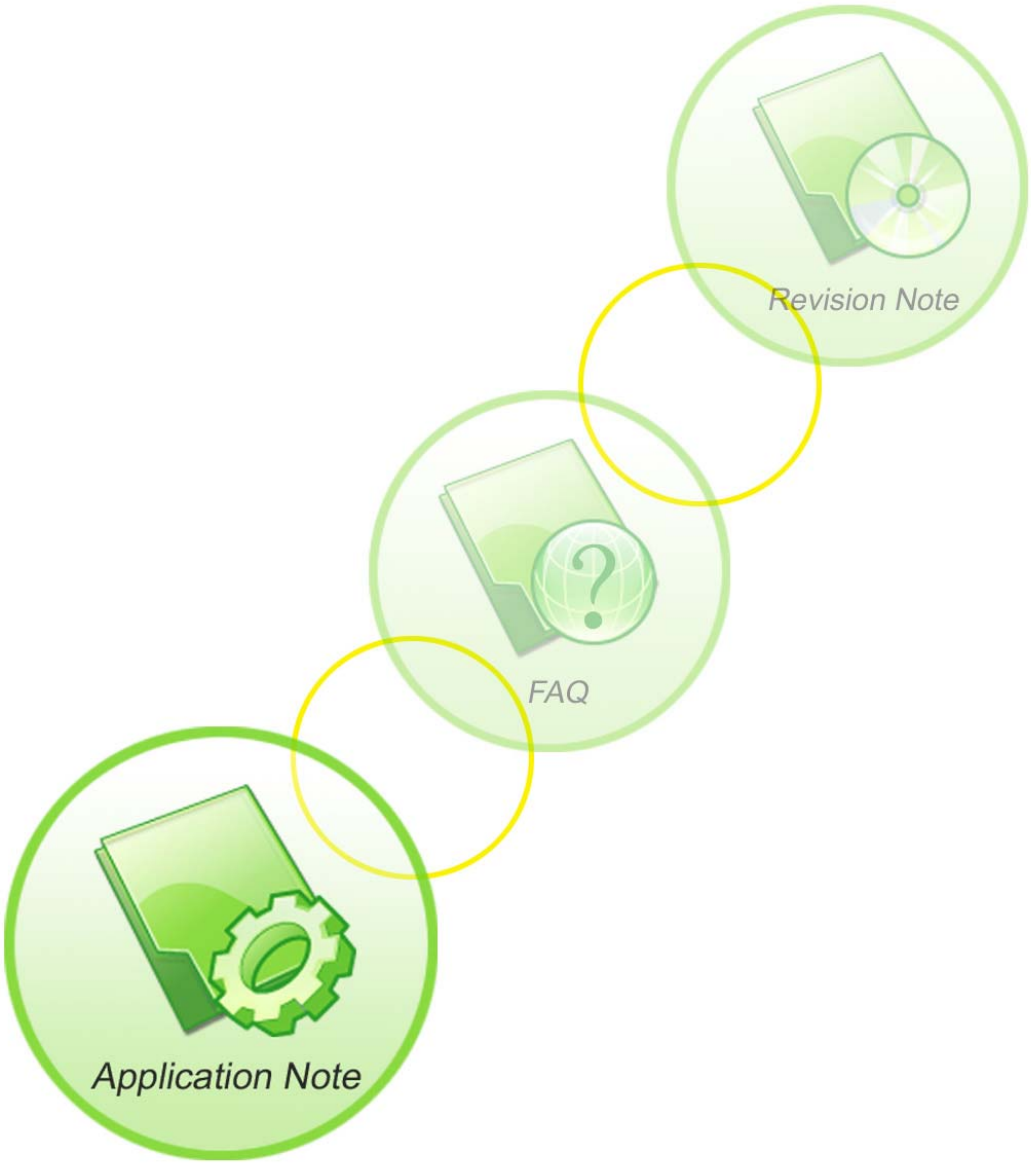
TCPIP Application Note for WCDMA Solution_V3.6



Scope

SIM5218, SIM5215, SIM5216, SIM5320, SIM6216, SIM6320

Reference

SIMCOM_SIM5320_ATC_EN_V1.31.doc

Version History

Version	Chapter	Comments
V1.0	New version	
V2.0	Support multiple TCP/IP connection and multiple TCP server, but only one TCP connection can be called to send data using AT+CIPSEND at the same time.	From 2012-03-29, all product firmwares built after that support this V2.0 application note. The AT+BT can be used to check the build time of firmware. For SIM5320E, it is supported from 1575B08SIM5320E.
V3.0	Support optimized TCP/IP operation mode. Multiple TCP sockets can send/receive data without interference.	From 2013-04-05, the product firmwares built after that support this V3.0 application note. For SIM5320E, it is supported from 1575B11SIM5320E.
V3.6	Support UDP transparent mode. Support one socket works on transparent mode using one port(for example USB-MODEM), other sockets work on AT command mode using another port(for example USB-AT).	From 2013-11-15, the product firmwares built after that support this V3.6 application note. For SIM5320E, it is supported form 1575B12SIM5320E.

Content

1. External PPP Setting

Port: USB->modem / UART, Hardware flow control

AT command:
AT+CGDCONT=1,”IP”,”apn”
ATD*99#

Note, Sequence of +++ could be issued to exit data mode.

2. SIMCom Internal TCPIP Protocol

2.1 Network Environment

TCPIP application is based on GPRS network; so, ensure GPRS network is available before TCPIP setup. Following is the recommended steps.

AT+CSQ
+CSQ: 23,0

OK
AT+CREG?
+CREG: 0,1

OK
AT+CPSI?
+CPSI: GSM,Online,460-00 0x1816,63905,81 EGSM 900,-68,0,31-31

OK
AT+CGREG?
+CGREG: 0,1

OK

2.2 PDP Context Enable/Disable

APN setting：
AT+CGSOCKCONT=1,"IP","CMNET"
OK
AT+CSOCKSETPN=1
OK

Note, usually CSOCKAUTH and CSOCKSETPN parameter are kept default if not care about.

Enable PDP context:

AT+CIPMODE=0 *// command mode, if not configured, it's 0 as default. If want data mode, please configure before Net open.*
OK
AT+NETOPEN
OK

+NETOPEN: 0

AT+IPADDR
+IPADDR: 10.113.43.157

OK

Disable PDP context:

AT+NETCLOSE
OK

+NETCLOSE: 0

2.3 Command Mode (Non-transparent mode)

Command mode is sometimes called non-transparent mode, which is default configured by module. Multi sockets are available under this mode.

2.3.1 TCP Client

AT+CIPOPEN=0,"TCP","116.236.221.75",8011//only IP address is supported
OK

+CIPOPEN: 0,0

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

+CIPSEND: 0,5,5

AT+CIPSEND=0, //the second parameter is empty means using <Ctrl+Z> to check the end
>HELLO<Ctrl+Z>
OK

+CIPSEND: 0,5,5

AT+CIPCLOSE=0 // *close by local*
OK

+CIPCLOSE: 0,0

Note, if connection closed by remote server, following URC will return:
+IPCLOSE: 0, 1
Here, the meaning of second parameter in this URC is following,
0 - closed by local, active
1 - closed by remote, passive
3 - Reset

2.3.2 UDP Connetion

One socket could communicate with multiple UDP channels.

AT+CIPOPEN=0,"UDP",,,9000//here 9000 is local port
+CIPOPEN: 0,0
OK

AT+CIPSEND=0,5,"16.236.221.75",9015

```
>hello
OK

+CIPSEND: 0,5,5

AT+CIPSEND=0,5,"16.236.221.75",8058
>12345
OK

+CIPSEND: 0,5,5

//the second parameter is empty means using <Ctrl+Z> to check the end
AT+CIPSEND=0,,"16.236.221.75",8058
>12345<Ctrl+Z>
OK

+CIPSEND: 0,5,5


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

2.3.3 Extended Information

Command AT+CIPHEAD is used to show IP head (data length) information, and command AT+CIPSRIP is used to show remote IP address and port once data received.

```
AT+CIPHEAD=1
AT+CIPSRIP=0
AT+CIPOPEN=0,"TCP","116.236.221.75",8011
OK

+CIPOPEN: 0,0

AT+CIPSEND=0,5
>11111
OK

+CIPSEND: 0,5,5

// here, remote data is coming
```

+IPD13
hello from pc
AT+CIPSRIP=1
OK
// here, remote data is coming
RECV FROM:116.236.221.75:8011
+IPD15
hello from pc 2
AT+CIPCLOSE=0
OK

+CIPCLOSE: 0,0

2.3.4 TCP SERVER

Module supports 4 sockects to listen.

AT+CGSOCKCONT=1,"IP","CMNET"
OK
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

//If a socket is accepted, the following URC will be reported:
+CLIENT: 0,1,192.168.108.5:57202
//User can use AT+CIPOPEN? to check the accepted socket
AT+CIPOPEN?
+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+SERVERSTOP=0 *// if unspicified, 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

Note, we can check connection status with command AT+CIPOPEN. If some socket needs to close, please issue command AT+CIPCLOSE=<linked_num>.

2.3.5 Connection Status Checking

AT+CIPOPEN?
+CIPOPEN: 0
+CIPOPEN: 1
+CIPOPEN: 2

+CIPOPEN: 3
+CIPOPEN: 4
+CIPOPEN: 5
+CIPOPEN: 6
+CIPOPEN: 7
+CIPOPEN: 8
+CIPOPEN: 9

OK
AT+CIPOPEN=0,"TCP","116.236.221.75",8011
OK

+CIPOPEN: 0,0

+IPD15
hello from pc 3
AT+CIPOPEN?
+CIPOPEN: 0, "TCP","116.236.221.75",8011,-1 *// last parameter of -1 indicates this connection is active, this socket acts as client*
+CIPOPEN: 1
+CIPOPEN: 2
+CIPOPEN: 3
+CIPOPEN: 4
+CIPOPEN: 5
+CIPOPEN: 6
+CIPOPEN: 7
+CIPOPEN: 8
+CIPOPEN: 9

OK

2.3.6 Receive data manually

AT+CIPRXGET=1//this only needs to be set once
OK
// here, remote data is coming
RECV FROM:116.236.221.75:8011
//now use AT command to retrieve the cached received data.
AT+CIPRXGET=2,1,1024
+CIPRXGET: 2,1,15,0
hello from pc 2

OK

2.4 Data mode (Transparent mode)

Currently, only one socket is available under transparent mode, either TCP client or TCP server. Command AT+CIPCCFG could be configured several parameters for data transmission under transparent mode. Before using data mode, the AT+CIPMODE=1 must be called first.

Note: **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.**

2.4.1 TCP Client

AT+NETOPEN
OK

+NETOPEN: 0
AT+CIPOPEN=**0**,"TCP","116.236.221.75",8011*//only <link_num>=0 is allowed to operate with transparent mode.*
CONNECT 115200
// sequence of +++ to quit data mode
OK
ATO *// command ATO to quit command mode*
CONNECT 115200
// sequence of +++ to quit data mode
OK
AT+CIPCLOSE=0
OK

CLOSED
+CIPCLOSE: 0,0

AT+NETCLOSE
OK

+NETCLOSE: 0

2.4.2 TCP Server

ATS0=7 *// ATS0 should be configured for TCP server application*

OK
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
// sequence of +++ to quit data mode
OK
AT+CIPCLOSE=0 *// close client connection*
OK

CLOSED
+CIPCLOSE: 0,0

AT+SERVERSTOP=0 *// close server socket*
+SERVERSTOP: 0,0
OK

2.4.3 UDP Socket

AT+NETOPEN
OK

+NETOPEN: 0
AT+CIPOPEN=**0**,"UDP","116.236.221.75",8011,8080*//only <link_num>=0 is allowed to operate with transparent mode.*
CONNECT 115200
// sequence of +++ to quit data mode
OK
ATO *// command ATO to quit command mode*
CONNECT 115200
// sequence of +++ to quit data mode
OK
AT+CIPCLOSE=0

CLOSED
+CIPCLOSE: 0,0
OK

AT+NETCLOSE
OK

+NETCLOSE: 0

Note, the factors which influence data rate are following:
AT&E1 the data rate should be the serial connection rate;
AT&E0 the data rate is the wireless connection speed (based on QOS, refer to command AT+CGSOCKQREQ/AT+CGSOCKEQREQ/AT+CGSOCKQMIN/AT+CGSOCKEQMIN).

2.5 Switch between data mode and command mode

Hardware flow control is recommended.
Currently, USB->modem port, USB->AT port and UART port all support hardware flow control.

Software switching: escape sequence +++. Please take care, this is a complete command, do not separate each character, also take care that the time delay before and after this sequence should be more than 1000 milliseconds, the interval of each character should not more than 900 milliseconds.
Hardware switching: DTR pin could be used to trigger data mode and command mode changed. Command AT&D1 should be configured before application.

2.6 TCP retransmission information

Each sending TCP packet needs to get a TCP ACK packet from peer socket. If the TCP ACK packet is not got in time, the module shall resend the same packet. The waiting for TCP ACK packet interval is <ESTIMATED_ROUND_TRIP_TIME>*$2^{(n-1)}$ seconds, while n is the retry times. Also for a packet sending, the total trying send time is 2 minutes. For example:

1. Send the TCP packet, here as a sample, the module measures <ESTIMATED_ROUND_TRIP_TIME> as 3 seconds. In runtime, each retransmission would use the latest measured <ESTIMATED_ROUND_TRIP_TIME> value in the following steps.
2. Wait 3 seconds, and if TCP ACK packet is not got, resend the packet
3. Wait another 6 seconds, and if TCP ACK packet is not got, resend the packet
4. Wait another 12 seconds, and if TCP ACK packet is not got, resend the packet

5. Wait another 24 seconds, and if TCP ACK packet is not got, resend the packet
6. Wait another 48 seconds, and if TCP ACK packet is not got, resend the packet
7. Wait another 27 seconds, and if TCP ACK packet is not got, regards socket sending failure and close the socket. (Here only 27 seconds waiting is because that the total trying time is 2 minutes).
8. If the TCP ACK packet is got within the previous steps, the packet is regarded as sending successfully.

User can modify the total allowed retrying send times by set the first parameter of AT+CIPCCFG. For example, if AT+CIPCCFG=3, then the packet sending should be as following:

1. Send the TCP packet, here as a sample, the module measures <ESTIMATED_ROUND_TRIP_TIME> as 3 seconds. In runtime, each retransmission would use the latest measured <ESTIMATED_ROUND_TRIP_TIME> value in the following steps.
2. Wait 3 seconds, and if TCP ACK packet is not got, resend the packet
3. Wait another 6 seconds, and if TCP ACK packet is not got, resend the packet
4. Wait another 12 seconds, and if TCP ACK packet is not got, resend the packet
5. Wait another 24 seconds, and if TCP ACK packet is not got, regards socket sending failure and close the socket
6. If the TCP ACK packet is got within the previous steps, the packet is regarded as sending successfully.

User also can modify the minimum waiting interval by setting the 7th parameter of AT+CIPCCFG. For example, if AT+CIPCCFG=,,,,,,10000, then the packet sending interval should be should be as following:

1. Send the TCP packet, here as a sample, the module measures <ESTIMATED_ROUND_TRIP_TIME> as 3 seconds. In runtime, each retransmission would use the latest measured <ESTIMATED_ROUND_TRIP_TIME> value in the following steps.
2. Wait MAX(10, $3*2^{(n-1)}$) = 10 seconds, and if TCP ACK packet is not got, resend the packet
3. Wait another MAX(10, $3*2^{(n-1)}$) = 10 seconds, and if TCP ACK packet is not got, resend the packet
4. Wait another MAX(10, $3*2^{(n-1)}$) = 12 seconds, and if TCP ACK packet is not got, regards socket sending failure and close the socket
5. Wait another MAX(10, $3*2^{(n-1)}$) = 24 seconds, and if TCP ACK packet is not got, resend the packet
6. Wait another MAX(10, $3*2^{(n-1)}$) = 48 seconds, and if TCP ACK packet is not got, resend the packet
7. Wait another 16 seconds, and if TCP ACK packet is not got, regards socket sending failure and close the socket. (Here only 16 seconds waiting is because that the total trying time is 2 minutes).
8. If the TCP ACK packet is got within the previous steps, the packet is regarded as sending successfully.

The two parameters can be used together and they may affect AT+CIPOPEN/AT+CIPSEND/AT+CIPCLOSE.

2.7 Set TCP maximum timeout value

User can set the maximum timeout value for AT+NETOPEN, AT+CIPOPEN and AT+CIPSEND using AT+CIPTIMEOUT command:
AT+CIPTIMEOUT=<netopen_timeout>,<connect_timeout>,<send_timeout>,
for example:
AT+CIPTIMEOUT=40000, 30000, 25000

2.8 Set DNS maximum timeout value

User can set the maximum timeout value for DNS query using AT+CIPDNSSET command:
AT+CIPDNSSET=<max_net_retries>,<net_timeout>,<max_query_retries>.

The timeout value for performing DNS query is <net_open_time> + 3000ms + 1000ms*<dns_query_retry_counter>. Here <net_open_time> is the time for opening PS network. <dns_query_retry_counter> is the retry counter for sending DNS query using UDP packet. By default, the maximum DNS query time is long, so the AT+CIPDNSSET=0,30000,5 is recommended to be used, for this setting, the maximum timeout value is 63 seconds.

2.9 Force to send FIN packet to peer when closing TCP socket

By default, when the module calls AT+CIPCLOSE in PS network dormancy state, It will close the socket immediately without notifying peer socket. User can set AT+CNVW=1341,0,”01”, this will force the AT+CIPCLOSE to send FIN packet to peer socket even in PS network dormancy state. This setting only needs to be performed once, and it will take effect from next power cycle.

3.0 Use TCP and voice call together

Currently GSM/CDMA/EVDO modes cannot use TCP and voice call together, When using voice call, the TCP transfer shall be suspended. So when using voice call in the three modes, don’t transfer data using TCP.

Contact us

SIMCom Wireless Solutions Co., Ltd.
Add: Building A, SIM Technology Building, No.633, Jinzhong Road, Changning District 200335
Tel: +86 21 3252 3300
Fax: +86 21 3252 3020
URL: http:/www.sim.com/wm/