

*Using AT commands to control  
TCP/IP stack on SM5100B-D  
modules*

## Revision history

Version	Date	Author	Update description
1.0	20080620	Steven Shao	Create
1.1	20080920	Steven Shao	Added AT+SDATASEND/SDATAREAD and support for 1460bytes per packet

# Contents

1: Overview Of The TCP/UDP AT Commands .....	4
2: Use of the AT commands and examples.....	5
2.1 Mobile Originated GPRS session .....	5
2.2 Disconnection and retransmission.....	7
2.3 Error handling .....	7

# 1: Overview of the TCP/UDP AT Commands

The TCP/IP features provided by the AT commands for the SM5100B-D are intended to provide a subset of the features normally available at the socket level when using a conventional TCP/IP stack, with some simplifications and customizations based on the specific features of the SM5100B-D.

The new commands allow the application writer to create and destroy UDP and TCP sockets, to control underlying GPRS PDP contexts, to transfer data to and from the module, and to interrogate IP status information about the active link. The internal TCP/IP stack allows TCP connections are limited to a maximum of 10.

The socket interface is provided by a series of AT commands outlined below.

AT+SDATACONF	Configure the parameters of TCP/UDP connection
AT+SDATASTART	Start the TCP/UDP connection
AT+SDATASEND	Send the TCP/UDP data(transparent, maximum length is 1460 bytes)
AT+SDATASEND	Send the TCP/UDP data(hex format)
AT+SSTRSEND	Send the TCP/UDP data(string format)
AT+SDATASTATUS	Query the socket status and the TCP sent/acknowledged/received bytes
AT+SDATAREAD	Read the received TCP/UDP data
AT+SDATATREAD	Read the TCP/UDP data(transparent)
AT+SDATARXMD	Set the received mode(hex or string format)
AT+TRT	Set the TCP maximum retry times

For more detailed information on the TCP/IP AT commands and their parameters, please refer to the AT commands manual for the SM5100B-D

This document applies to firmware versions IDH.SC6600D-BASE4\_R1.0.0011\_B03 or newer.

## 2: Use of the AT commands and examples

### 2.1 Mobile Originated GPRS session

#### System Power On:

```
+SIND: 1 (SIM INSERTED)
+SIND: 10,"SM",1,"FD",1,"LD",1,"MC",1,"RC",1,"ME",1 (SIM READY)
+SIND: 11 (REGISTERED TO NETWORK)
+SIND: 3 (CALL READY)
+SIND: 4 (SMS READY)
```

#### -> Query and waiting for GPRS attached to network

```
AT+CGATT?
+CGATT : 1 (GPRS Attached)
OK
```

#### -> Setup PDP Context (in this example China Mobile in China)

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

#### -> Configure the PDP context parameters of PCO if the user and password is needed for GPRS APN

```
AT+CGPCO=0,"user","password",1
OK
```

#### -> Activate PDP Context

```
AT+CGACT=1,1
OK
```

#### -> Configure the remote host and port, we can open a TCP connection

```
AT+SDATACONF=1,"TCP","www.sina.com.cn",80 (to a remote host name)
OK
```

or

```
AT+SDATACONF=1,"TCP","211.123.11.21",1234 (to a remote IP address)
OK
```

#### -> Start TCP/UDP connecting

```
AT+SDATASTART=1,1
OK
```

#### -> Query the socket connecting status

AT+SDATASTATUS=1  
+SOCKSTATUS: 1,0,0104,0,0,0 (0 means socket not connected, 0104 means socket is connecting)

AT+SDATASTATUS=1  
+SOCKSTATUS: 1,1,0102,0,0,0 (1 means socket connected)

-> Start sending data

AT+SDATASEND=1,10  
>1234567890 <Ctrl+Z>  
OK (data is sent to socket buffer and waiting to send)

-> If necessary you can query if the data is sent successfully and acknowledged by remote host (for TCP only)

AT+SDATASTATUS=1  
+SOCKSTATUS: 1,1,0102,10,10,0 (total sent 10 bytes and acked 10 bytes, data is successfully sent)

-> Clear the total sent and acknowledged bytes if necessary

AT+SDATASTATUS=0  
OK

-> Query the total sent and acknowledged bytes

AT+SDATASTATUS=1  
+SOCKSTATUS: 1,1,0102,0,0,0

-> When data is received from the remote host, following URC appears

+STCPD: 1 (connection 1 has received data from remote host)

-> Read the data

AT+SDATAREAD=1 (read the data from connection 1)  
+SDATA: 1, 10,1234567890 (received 10 bytes from remote host, hex format)

-> Close the TCP/UDP connection

AT+SDATASTART=1,0 (close the connection 1)  
OK

-> If the socket has been terminated by the remote host, following URC appears

+STCPC: 1 (connection 1 is terminated by the remote host)

## 2.2 Disconnection and retransmission

TCP provides reliable transmission through the use of re transmission of unacknowledged data. The module behaves in the following manner to determine when this should happen.

1: If the link at some point drops out i.e. server crashes, RF coverage is lost, etc, the unit will start retransmissions.

2: The first retransmission will occur 2 seconds after initial transmission, the second 4 seconds later, the third 8 seconds, the fourth 16 seconds, and the fifth 32 seconds later, from then on they will occur every 64 seconds.

3: The socket is closed after the appropriate number of retransmissions has been reached as determined by the AT+TST command, the maximum retransmission number is 12.

Obviously this does not apply to UDP as it is connectionless and unreliable.

## 2.3 Error handling

The most possible errors during TCP/UDP session are listed following:

26	+CME ERROR:28	+CME ERROR: GPRS operation failure
27	+CME ERROR:29	+CME ERROR: GPRS send data failure

+CME ERROR: 28

This means GPRS PDP context activate failure, the most possible cause is due to the wrong APN, user and password or GPRS network failure. If the APN, user and password is correct, firstly you need close the TCP/UDP connection (AT+SDATASTART=1, 0) if the connection is still active, then deactivate (AT+CGACT=0, 1) and activate PDP context (AT+CGACT=1, 1), restart the TCP/UDP connection if PDP Context is successfully activated. If it still doesn't work, you may need detach (AT+CGATT=0) and attach (AT+CGATT=1) the GPRS network. Otherwise you need use AT+CFUN to restart the module.

+CME ERROR: 29

This means TCP/UDP data send failure, the most possible cause is due to the socket is disconnected. You can use AT+SDATASTATUS check the TCP connection status, if the socket is disconnected, you need restart the TCP/UDP connection. Otherwise if the socket is still connected you need check whether the length of input data is right.