



# TCPIP Application Note

**SIM900\_AN\_TCPIP\_V1.00**



<b>Document Title:</b>	TCPIP Application Note
<b>Version:</b>	1.00
<b>Date:</b>	2010-1-20
<b>Status:</b>	Created
<b>Document Control ID:</b>	SIM900_AN_TCPIP_ V1.00

### **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 Limited., copying of this document and giving it to others and the using or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights reserved in the event of grant of a patent or the registration of a utility model or design. All specification supplied herein are subject to change without notice at any time.

*Copyright © Shanghai SIMCom Wireless Solutions Ltd. 2010*

## Contents

Version history .....	5
1 Introduction .....	6
1.1 Scope of the document .....	6
1.2 Related documents .....	6
1.3 Conventions and abbreviations .....	6
2 Structure .....	7
3 Single Connection .....	7
3.1 Non-transparent mode .....	7
3.1.1 How to establish a TCP client connection .....	8
3.1.2 How to establish a UDP client connection .....	9
3.1.3 How to establish a TCP server connection .....	10
3.2 Transparent mode .....	12
3.2.1 What is transparent mode .....	12
3.2.2 How to configure transparent mode .....	12
3.2.3 How to establish connection under transparent mode .....	12
3.2.4 How to switch between data mode and command mode .....	13
3.2.4 How to handle incoming call and short message in data mode .....	14
3.3 Fix local port of TCP/UDP client .....	14
4 Multi Connection .....	14
4.1 As a client .....	14
4.2 As a TCP server .....	16
5 DNS function .....	19
6 Data sending related .....	20
6.1 Fixed length sending .....	20
6.2 Timed sending .....	20
6.3 Query available data length to be sent .....	20
6.4 Select data transmitting mode .....	21
6.5 Query data transmitting amount .....	21
7 Data receiving related .....	23
8 gprs states exchange related .....	23
9 Connection closing related .....	25
10 Connection activity checking related .....	25
11 Power consumption with existing connection .....	25
12 Error handling .....	26

## VERSION HISTORY

Now document: SIM900 TCPIP Application Note” Version 1.00

Chapter	Page	What is new
V1.00		Origin

# 1 INTRODUCTION

## 1.1 Scope of the document

This document describes how to use the internal TCPIP stack of SIM900 through AT commands. Examples are also given for reference. This document can be used for SIM900 serial modules, like SIM900, SIM900D, and SIM900A.

## 1.2 Related documents

[1] SIM900 AT Commands Set.

## 1.3 Conventions and abbreviations

TCP	Transmission Control Protocol
UDP	User Datagram Protocol
APN	Access Point Name
GPRS	General Packet Radio Service
PDP	Packet Data Protocol
DNS	Domain Name System

## 2 STRUCTURE

There are two modes of connection for SIM900 TCP/IP application: Single connection and Multi connection. When in single connection mode, SIM900 can work at both transparent mode and non-transparent mode; and under these two transmission modes, SIM900 can be configured as either TCP/UDP client or TCP server. When in multi connection mode, SIM900 only can work at non-transparent mode. In this mode, SIM900 can work as an absolute TCP/UDP client, which can establish 8 connections in total. In this mode, it also can be configured as one TCP server, which allows 7 TCP/UDP clients to connect in; and the TCP server also can act as a client, establishing 7 connections to one remote server. The structure of the TCP/IP application is given as below.

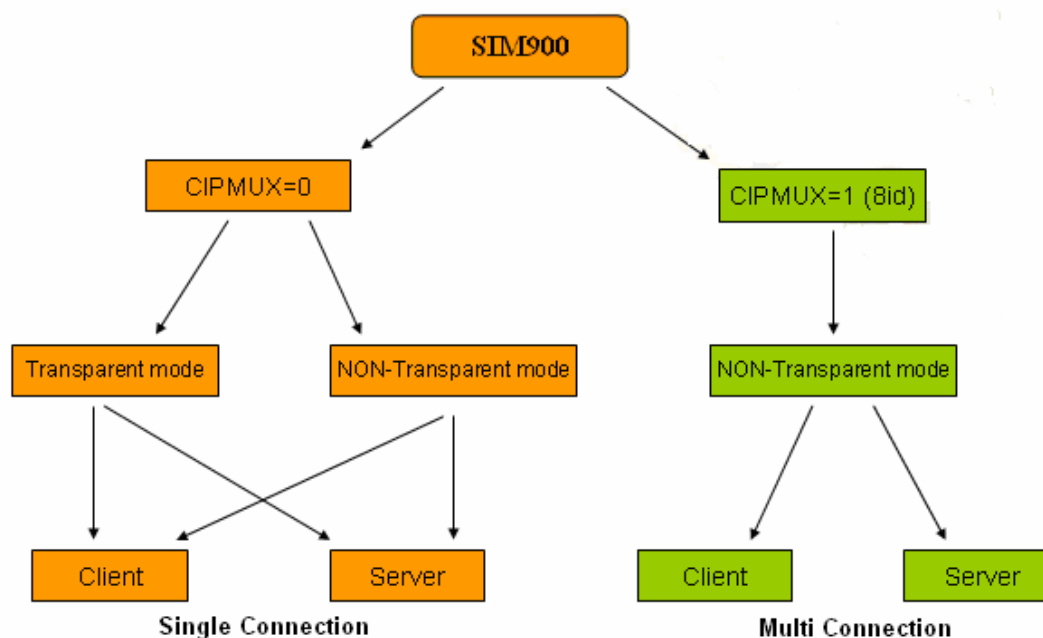


Figure1: SIM900 TCPIP Structure

## 3 SINGLE CONNECTION

Command AT+CIPMUX=<n> is used for selecting TCPIP connection mode, when n=0, it is single connection; n=1, it is multi connection. The default configuration is single connection. For single connection, SIM900 supports both transparent mode and non-transparent mode.

### 3.1 Non-transparent mode

Command AT+CIPMODE=<n> is used for selecting TCPIP application mode, when n=0, it is non-transparent mode(normal mode); n=1, it is transparent mode. The default configuration is non-transparent mode. There are three working modes for SIM900 under this mode: TCP client,

UDP client and TCP server.

### 3.1.1 How to establish a TCP client connection

Firstly, before any TCP/UDP related operation, the module should be connected to GSM or GPRS network. You can use the commands “AT+CREG?” and “AT+CGATT?” to query the GSM network registration status whether the module has been attached to GPRS service.

```
AT+CPIN?
```

```
+CPIN: READY
```

```
OK
```

```
AT+CSQ
```

```
CSQ: 20,0
```

```
OK
```

```
AT+CREG?
```

```
+CREG: 0,1
```

```
OK
```

```
AT+CGATT?
```

```
+CGATT: 1
```

```
OK
```

Secondly, you should use the command group AT+CSTT, AT+CIICR and AT+CIFSR to start the task and activate the wireless connection. Lastly, you can establish TCP connection between SIM900 and server by AT command (AT+CIPSTART=”TCP”,”IP Address of server”, “port number of server”). If the connection is established successfully, response “CONNECT OK” will come up from the module. Now you can send data to server with “AT+CIPSEND”. “AT+CIPSEND” will return with promoting mark “>”, you should write data after “>” then issue CTRL+Z (0x1a) to send. If sending is successful, it will respond “SEND OK”. And if there is data coming from server, the module will receive the data automatically from the serial port. You can close the TCP connection with “AT+CIPCLOSE” command. Below is an example of TCP connection to remote server.

```
AT+CGATT?
```

```
+CGATT: 1
```

```
OK
```

**SIM900\_TCPIP\_Application Note**

```
AT+CSTT="CMNET" // Start task and set APN. See Note[1].

OK

AT+CIICR // Bring up wireless connection (GPRS or
CSD)

OK

AT+CIFSR // Get local IP address

10.78.245.128

AT+CIPSTART="TCP", "116.228.221.51", "8500" // Start up the connection

OK

CONNECT OK // The TCP connection has been established
successfully.

AT+CIPSEND

> hello TCP server //Send data to remote server, CTRL+Z (0x1a) to
send. See Note[2]

SEND OK // The remote server receives the data.

hello sim900 // Received data from remote server

CLOSED // Remote server closed the connection
```

Note[1]: The default APN is "CMNET", with no username or password. Please ask your local GSM provider to get the APN.

Note[2]: You should write data only after the promoting mark ">", and then use CTRL+Z to send. You can use command "AT+CIPSPRT" to set whether echo promote ">" after issuing "AT+CIPSEND".

**3.1.2 How to establish a UDP client connection**

The process of establishing UDP connection is similar to TCP.

```
AT+CGATT?

+CGATT: 1

OK

AT+CSTT="CMNET"
```



OK

AT+CIICR

OK

AT+CIFSR

10.77.65.18

AT+CIPSTART="UDP","116.228.221.51","9600"

CONNECT OK

AT+CIPSEND

> *sim900 UDP test*

SEND OK

// The data has been sent out from the serial port, but it is unknown if the data reaches the UDP server. See Note[3].

*UDP test*

//Receive data from remote server

AT+CIPCLOSE

// Close the UDP connection

CLOSE OK

**Note[3]:** For TCP, "SEND OK" means the data has been sent out and received successfully by the remote server, due to the TCP connection-oriented protocol; for UDP, "SEND OK" just means the data has been sent out from the serial port of module, not meaning the data reaching the server, due to the UDP simpler message-based connectionless protocol.

### 3.1.3 How to establish a TCP server connection

In single connection mode, when configured as TCP server, SIM900 allows one client to connect in. You should use the command "AT+CIPSERVER=1,<port>" to start the server function and set listening port. If successfully, response "SERVER OK" will be returned, and now the server starts to listen the TCP port. And then, you should use command "AT+CIFSR" to get local IP address, which is the server IP address.

If the client connects to the server successfully, the IP address of remote client will be displayed at server side. And the server can receive TCP data from remote side. Also you can use command AT+CIPSEND to send data to remote client. You can use "AT+CIPSERVER=0" to close the listening status. Also, you can use AT+CIPCLOSE to close the TCP connection.

AT+CGATT?

```
+CGATT: 1

OK

AT+CIPSERVER=1,1234 //Start the TCP server, listening port:1234

OK

SERVER OK //TCP server is established successfully

AT+CIFSR //Get local IP address

10.78.33.227

AT+CIPSTATUS

OK

STATE: SERVER LISTENING // Query current status: listening

REMOTE IP: 10.78.103.220 // Remote client connected in

hello server // Receive data from remote client

AT+CIPSND

>hello client // Send data to remote client

SEND OK

AT+CIPSERVER=0 //Just close the listening status, the current
connection is still active.

OK

SERVER CLOSE

hello sim900 //Receive data from remote client

AT+CIPCLOSE //Close the TCP connection

CLOSE OK
```

## 3.2 Transparent mode

### 3.2.1 What is transparent mode

SIM900 supports transparent mode which provides a special data mode for data receiving and sending by TCP/IP application task. Once the connection is established under transparent mode, the module will be in data mode. All received data from serial port will be treated as data packet to be transferred later, similarly all data received from remote side will be sent to serial port directly. In transparent mode, all AT commands are not available. Methods are provided to switch back and forth between data mode and command mode. Once switched to command mode, all AT commands can be used again.

### 3.2.2 How to configure transparent mode

To enable transparent mode, the command AT+CIPMODE should be set to 1. In transparent mode, the command AT+CIPCCFG is used for configuring transfer mode, which has 4 parameters NmRetry, WaitTm, SendSz, Esc. NmRetry is the number of retries to be made for an IP packet. WaitTm is the number of 200ms intervals to wait for incoming data before sending the packet. SendSz is the size of data block to be received from serial port before sending. Esc represents whether the escape sequence is allowed, default is TRUE.

### 3.2.3 How to establish connection under transparent mode

The transparent mode is only supported under single connection. In transparent mode, SIM900 can work as three modes too: TCP client, UDP client and TCP server. Once the connection is established, CONNECT will be returned in the serial port and the module will be in data mode. At the same time, DCD pin will be active low. Below is an example of TCP client connection.

```
AT+CIPMODE=1 //Enable transparent mode
```

```
OK
```

```
AT+CGATT?
```

```
+CGATT: 1
```

```
OK
```

```
AT+CSTT="CMNET"
```

```
OK
```

```
AT+CIICR
```

```
OK
```

```
AT+CIFSR
```

10.76.157.180

AT+CIPSTART="TCP","116.228.221.51","8500"

OK

CONNECT //Connected successfully, enter into data mode

..... //Input data to serial port

OK //Quit data mode by pulling DTR or using  
"+++"

ATO //Return to data mode

CONNECT

*sim900 channel test* //Data received from remote server

CLOSED //Closed by remote server, quit data mode

### 3.2.4 How to switch between data mode and command mode

To switch from data mode to command mode, following methods are available:

- (1) The escape sequence can be used if the last parameter of AT+CIPCCFG is TRUE. The default escape sequence is +++, and to use this sequence, there should be 1000ms idle period before this sequence and 500ms idle period after this sequence. Besides, the interval between each + should not exceed 500ms, otherwise it will be treated as TCP/IP data.
- (2) DTR line of serial port can also be used. To use this method, AT&D1 should be set firstly. Pull DTR line to ground for at least 1 second and then pull up, the module will switch from data mode to command mode and OK will be returned indicating the module is in command mode.
- (3) For TCP client connection, if the remote server closes the connection, the module will switch back to command mode automatically.
- (4) For TCP server connection, if the remote client closes the connection, the module will switch back to command mode automatically.
- (5) If the module is deactivated from PDP context (+PDP DEACT) during data transferring, module will switch back to command mode automatically.

ATO command can be used to switch from command mode to data mode if the connection is active, and CONNECT will be returned again.

### 3.2.4 How to handle incoming call and short message in data mode

RI pin of serial port will give a 50ms low pulse when a call is coming during data mode. RI pin will give a 120ms low pulse when a short message is received.

To handle the incoming call or SMS, you need to enter command mode first (using DTR or +++), and then the module will give the normal unsolicited responses as following:

```
RING //incoming call
```

```
+CMTI: "SM", 10 //short message
```

And then it is able to answer the call or read the short message.

### 3.3 Fix local port of TCP/UDP client

Normally, if client SIM900 establishes TCP/UDP connection to remote server, the TCP/UDP connection will be dynamically allocated a local port when AT+CLPORT is set to 0. If AT+CLPORT is set to a nonzero value, the local port will be fixed at this value. The default configuration of this command is 0. Be aware that this command will be effective only in single connection and when module as a client.

## 4 MULTI CONNECTION

To enable multi connection, the command AT+CIPMUX must be set to 1. Be aware that in the multi connection, SIM900 only supports non-transparent mode. Due to this matter, the command AT+CIPMODE is invalid under multi connection.

In multi connection, SIM900 can work as a TCP/UDP client or a TCP server.

As a client, it can establish 8 connections to remote server in total, both TCP and UDP.

As a TCP server, it allows remote clients to connect in; at the same time, it can establish TCP/UDP connections to upper remote servers. Total 7 available connections are supported, with the server itself occupying one connection.

### 4.1 As a client

Being a client, SIM900 can establish both TCP and UDP connection to remote server. Total 8 connections are supported. The command "AT+CIPSTART=<n>,<mode>,<IP Address>,<port>" is needed. When the connection is established successfully, "<n>, CONNECT OK" will be returned. And then you can use command AT+CIPSNED=<n> to send data to the connection <n>. You should write data after the promoting mark ">" and use CTRL+Z (0x1a) to send. If sending

## SIM900\_TCPIP\_Application Note

successfully, “<n>, SEND OK” will be returned. You can use command AT+CIPCLOSE=<n> to close one specific connection with number <n>. Also you can send AT+CIPSHUT to close all connections.

```

AT+CIPMUX=1 //Enable multi connection

OK
AT+CGATT?

+CGATT: 1

OK

AT+CSTT="CMNET"

OK
AT+CIICR

OK
AT+CIFSR

10.76.166.72
AT+CIPSTART=0,"TCP","116.228.221.51","8500" //Establish a TCP connection, connection
number 0

OK

0, CONNECT OK
AT+CIPSTART=1,"UDP","116.228.221.51","9600" //Establish a UDP connection, connection
number 1

OK

1, CONNECT OK

AT+CIPSEND=0 //Send data to connection 0

> TCP test

0, SEND OK
AT+CIPSEND=1 //Send data to connection 1

>UDP test

1, SEND OK

```

```

+RECEIVE,0,17: //Received data from connection 0, data
                length 17

sim900 TCP test

+RECEIVE,1,17 //Received data from connection 1, data
                length 17

sim900 UDP test

1, CLOSED //Connection 1 is closed by remote server

AT+CIPSTATUS //Query the current connection status

OK

STATE: IP PROCESSING

C: 0,0,"TCP","116.228.221.51","8500","CONNECTED"
C: 1,0,"UDP","116.228.221.51","9600","CLOSED"
C: 2,,"","","","INITIAL"
C: 3,,"","","","INITIAL"
C: 4,,"","","","INITIAL"
C: 5,,"","","","INITIAL"
C: 6,,"","","","INITIAL"
C: 7,,"","","","INITIAL"

```

## 4.2 As a TCP server

Being a TCP server, SIM900 allows remote clients to connect in; in the meanwhile, it can establish TCP/UDP connections to remote servers too.

Before launching the server function, command group “AT+CSTT, AT+CIICR, AT+CIFSR” should be executed to activate the PDP context and get local IP address. Then you can send command “AT+CIPSERVER=1,<port>” to start the server function. If successfully, response “SERVER OK” will be returned, and now the server starts to listen the TCP port.

If the clients connect to the server successfully, the IP addresses of remote clients together with connection numbers <n> will be displayed at server side. Then the server can receive TCP data from remote clients. Also you can use command AT+CIPSEND=<n> to send data to remote client <n>. Simultaneously, you can connect the server SIM900 to remote servers by TCP/UDP using command “AT+CIPSTART=<n>,<mode>,<IP Address>,<port>”. Command “AT+CIPSERVER=0” can be used to close the listening status. You can use command AT+CIPCLOSE=<n> to close one specific connection with number <n> and use AT+CIPSHUT to close all connections.

AT+CIPMUX=1

OK

AT+CGATT?

+CGATT: 1

OK

AT+CSTT="CMNET"

OK

AT+CIICR

OK

AT+CIFSR

10.76.40.73

AT+CIPSERVER=1,8888

//Start server;listeninig port:8888

OK

SERVER OK

0, REMOTE IP: 10.76.40.73

//Remote client connect in, connection  
number 0 allocated

+RECEIVE,0,39:

//Received data from remote client, data  
length 39

*sim900 multi connection TCP server test*

AT+CIPSEND=0

//Send data to remote client

*> hello client*

0, SEND OK

AT+CIPSTART=1,"TCP","116.228.221.51","8500"

//Establish TCP connection to remote  
server

OK

1, CONNECT OK



**SIM900\_TCPIP\_Application Note**

AT+CIPSTART=2,"UDP","116.228.221.51","9600" //Establish UDP connection to remote server

OK

2, CONNECT OK

AT+CIPSEND=1 // Send TCP data to remote server

> data from connection 1

1, SEND OK

1, data to connection 1 //Received TCP data from remote server

AT+CIPCLOSE=2 //Close the UDP connection with remote server

2, CLOSE OK

AT+CIPSTATUS //Query current connection status

OK

STATE: IP PROCESSING

S: 0,0,"8888","LISTENING" //The TCP server is in the process of listening, occupying one connection

C: 0,0,"TCP","211.136.130.129","2020","CONNECTED"

C: 1,0,"TCP","116.228.221.51","8500","CONNECTED"

C: 2,0,"UDP","116.228.221.51","9600","CLOSED"

C: 3,,,"","","INITIAL"

C: 4,,,"","","INITIAL"

C: 5,,,"","","INITIAL"

C: 6,,,"","","INITIAL"

C: 7,,,"","","INITIAL"

AT+CIPSERVER=0 //Close the server

OK

SERVER CLOSE

AT+CIPSHUT //Deactivate the PDP context &close all connections

SHUT OK

## 5 DNS FUNCTION

SIM900 supports DNS auto parsing, so you can establish TCP/UDP connection to remote sever using domain name directly by the command “AT+CIPSTART=<mode>, <domain name>, <port>”. And then you can send data to remote server by AT+CIPSEND. In multi connection, the connection number <n> is needed.

You also can use command AT+CDNSGIP=<domain name > to query the IP address of given domain name.

AT+CDNSGIP="WWW.SIM.COM"

OK

+CDNSGIP: 1,"WWW.SIM.COM", "58.32.231.146" // Parsing succeeds, the IP address of ”  
[WWW.SIM.COM](http://WWW.SIM.COM)” is

“58.32.231.146”

AT+CDNSGIP="abctest"

OK

+CDNSGIP: 0,11 //Parsing fails, reason is 11

AT+CIPSTART="TCP", "WWW.SIM.COM",80 //Establish TCP connection to server by  
domain name

OK

CONNECT OK

AT+CIPSNED

> DNS test

SEND OK

## 6 DATA SENDING RELATED

SIM900 provides three ways to send data: changeable data length sending, fixed data length sending and timed sending. SIM900 also provides a method to let you know how much data is sent out from the module and received by remote server on an active TCP connection.

### 6.1 Fixed length sending

You can send the fixed length of data with “AT+CIPSEND=<LENGTH>”, then input the data after getting promoting mark “>”. The data will be sent automatically when the length of the input data equals to the value “LENGTH”. You don’t need the terminal symbol CTRL+Z (0x1a) in this case. For multi connection, the command is “AT+CIPSEND=<n>,<LENGTH>”.

### 6.2 Timed sending

There is another way to send data automatically. First, set the timer of sending data automatically by the command “AT+CIPATS=<mode>,<time>”, then issue “AT+CIPSEND” (“AT+CIPSEND=<n>” for multi connection) and lastly input the data after getting promoting mark “>”. The data will be sent automatically when the set timer expires.

### 6.3 Query available data length to be sent

You can use the command “AT+CIPSEND?” to query the current maximum data length available to be sent, which is determined by the network. In fact, the maximum data length is variable, depending on the actual network. The maximum data length can be up to 1460 bytes. In multi connection, the command “AT+CIPSEND?” will tell you current available maximum data length for all active connections.

AT+CIPSEND?

+CIPSEND: 1380

//Current maximum value 1380

OK

AT+CIPSEND=1381

//The fixed length exceeds maximum value

ERROR

AT+CIPSEND

>123....999

//The input data exceeds maximum value

ERROR

## 6.4 Select data transmitting mode

SIM900 supports quick sending mode. When command AT+CIPQSEND=0, it is normal sending mode. In normal mode, after you send data by AT+CIPSEND, if the server receives TCP data, it will give ACK message to module and the module will response SEND OK. When command AT+CIPQSEND=1, it is quick sending mode. When the data is input to the serial port of module by AT+CIPSEND, it will response DATA ACCEPT, while not response SEND OK. In such case, you can continuously use AT+CIPSEND to send data to server.

```
AT+CIPQSEND=1 //Enable quick sending mode
```

```
OK
```

```
AT+CIPSEND
```

```
> hello
```

```
DATA ACCEPT:5
```

## 6.5 Query data transmitting amount

The command AT+CIPACK is used to query previous connection data transmitting state. In single connection, the execution command AT+CIPACK will return +CIPACK:<txlen>,<acklen>,<nacklen>. The first parameter <txlen> is the data amount which has been sent; the second parameter <acklen> is the data amount confirmed successfully by the server; and the third parameter <nacklen> is the data amount without confirmed by the server. As long as the connection is still active, you can know how much TCP data you sent to server and how much is received successfully by the server in total. By this means, you can get the total data transmitting amount.

For multi connection, the correct command type is AT+CIPACK=<n>. <n> is the connection number.

```
AT+CIPSTATUS
```

```
OK
```

```
STATE: IP PROCESSING
```

```
C: 0,,"",,"", "INITIAL"
```

```
C: 1,0,"TCP", "116.228.221.51", "8500", "CONNECTED"
```

```
C: 2,0,"UDP", "116.228.221.51", "9600", "CONNECTED"
```

C: 3,,",",",",", "INITIAL "

C: 4,,",",",",", "INITIAL "

C: 5,,",",",",", "INITIAL "

C: 6,,",",",",", "INITIAL "

C: 7,,",",",",", "INITIAL "

AT+CIPSEND=1

> *TCP*

1, SEND OK

AT+CIPACK=1

+CIPACK: 3, 3, 0

// 3 bytes has been sent, and all of them are  
received by the remote server.

OK

AT+CIPSEND=1

> *TCP again*

1, SEND OK

AT+CIPACK=1

+CIPACK: 12, 12, 0

OK

AT+CIPSEND=2

> *UDP*

2, SEND OK

AT+CIPACK=2

+CIPACK: 3, 0, 3

OK

## 7 DATA RECEIVING RELATED

The module will receive data automatically if there is data coming from remote server. Several commands can help to get the information header.

“AT+CIPHEAD=1” helps to add IP header in the format “+IPD(data length):payload”.

“AT+CIPSRIP=1” helps to show the data source information in the format “RCV FROM: <IP ADDRESS>:<PORT >”.

“AT+CIPSHOWTP” helps to show the protocol (TCP/UDP) in the IP header. It takes effect only if “CIPHEAD” is enabled.

With these information, you can easily know the source of the data frame, the amount of the payload and the protocol. It also can help you distinguish the received data from AT command responses.

## 8 GPRS STATES EXCHANGE RELATED

For single connection, there are 9 GPRS states in total; for multi connection, there are 7 GPRS states. After some AT commands are executed, the corresponding state will be changed. You can get a general idea from the following diagrams.

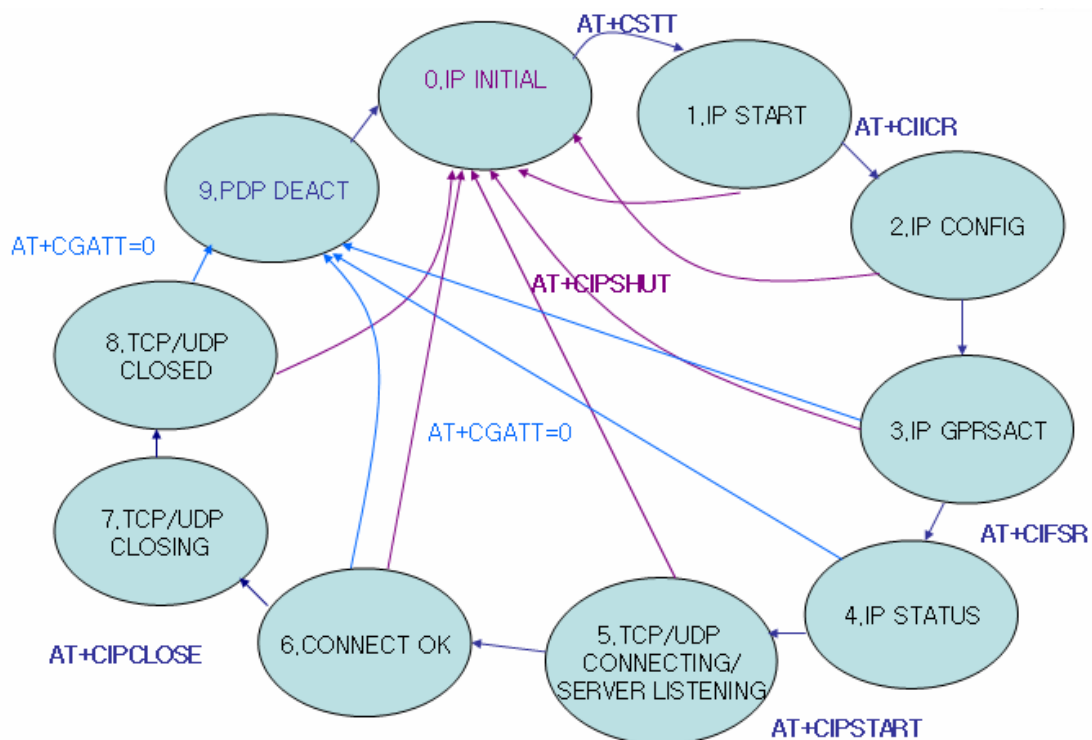


Figure2: GPRS States Diagram for single connection

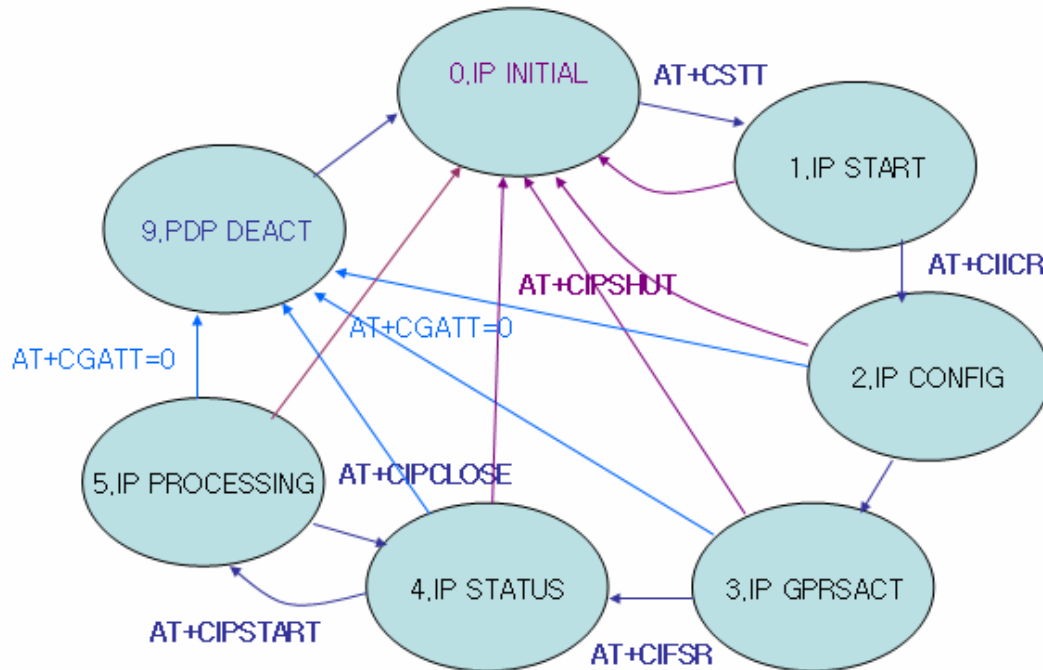


Figure3: GPRS States Diagram for multi connection

- IP INTIAL: GPRS initial status
- IP START: Start a TCP/UDP task
- IP CONFIG: Configure PDP context
- IP GPRSACT: Context active already
- IP STATUS: Get local IP address
- TCP/UDP CONNECTING: Connecting server now
- SERVER LISTENING: Listening server port now
- IP PROCESSING: Processing the existing connection now
- CONNECT OK: Connect server success

- TCP/UDP CLOSING: Closing connection now
- TCP/UDP CLOSED: Connection closed (local IP/PDP context still there)
- PDP DEACT: PDP context deactivated

## 9 CONNECTION CLOSING RELATED

You can use the command `AT+CIPCLOSE=<mode>` to close the TCP or UDP connection. If `<mode>` is 0, it is slow closing, if `<mode>` is 1, it is quick closing. In slow closing, the module will interactive with the server when close the TCP connection. Thus, the time of returning “CLOSE OK” will be a bit long. This method is suitable for steady network. In quick closing, the module will disconnect the connection compulsorily and return “CLOSE OK” immediately, without interaction with the server.

The default setting is slow closing, so the `<mode>` 0 can be omitted. And in multi connection, the connection number `<n>` should be added in front of `<mode>`.

Please be noted that command `AT+CIPCLOSE` only closes current TCP/UDP connection, but PDP context is still active. Also you can close connection by `AT+CIPSHUT`, with current PDP context being deactivated.

## 10 CONNECTION ACTIVITY CHECKING RELATED

You can use the command `AT+CIPSTATUS` to query current connection status.

In transparent mode, DCD pin can be used for this purpose. If TCP/UDP connection exists, the DCD pin will be active (low). At any time if the connection is dropped, DCD pin will go inactive (high).

## 11 POWER CONSUMPTION WITH EXISTING CONNECTION

The module can enter sleep mode to save power consumption with existing connection, in which the current consumption is reduced to only several mA. The sleep mode function is enabled by setting “`AT+CSCLK=1`”. With this setting, if the DTR maintains HIGH and there is no interruption on either H/W of S/W or event over-the-air, module will go into sleep. In the sleep mode, the serial port stops working, which means no AT command can be issued any more. During the sleep mode with existing connection, the incoming data from the remote end can wake up the module. Any event over-the-air, such as incoming call or SMS etc, will wake up the module too.



## **SIM900\_TCPIP\_Application Note**

---

At the module self end, two alternative ways can help to wake up the module:

(1) Pull DTR pin low.

The serial port will be active after DTR is pulled to low for 20ms.

(2) RTC alarm expires.

## **12 ERROR HANDLING**

If an error occurs in TCP/UDP connection, for example TCP sending data error or TCP connection dropping, it is suggested to close the connection by command AT+CIPCLOSE and then restart the connection by AT+CIPSTART. If the error still occurs, please use AT+CIPSHUT to shut off the PDP context and then restart the connection. If these two methods above can't help to solve it, we recommend you to reset the module.

**Contact us:**

**Shanghai SIMCom Wireless Solutions Ltd.**

Add: Building A, SIM Technology Building, No.633 Jinzhong Road, Changning District, Shanghai, P. R. China 200335

Tel: +86 21 3252 3300

Fax: +86 21 3252 3301

URL: [www.sim.com/wm](http://www.sim.com/wm)