

GSM/GPRS Module

Enhanced AT Command Reference Manual <V1.8 Beta>

[ADH Technology Co.,LTD](#)

Subject to changes in technology, design and availability

URL: <http://www.adh-tech.com.tw>

Email sales@adh-tech.com.tw

Revision History

No.	Date	Notes
V1.8	2010/2/8	Enhance APIA, AIPO, AIPW and AIPSST with more parameters. Add new data transfer mode (non-transparent mode).
V1.6	2009/12/18	Add new data transfer mode (data mode)
V1.5	2009/10/15	Add GPRS/Socket status commands Auto Baud Rate function
V1.4	2009/6/17	Modify TCP/IP AIPO command and add AIPDNR, AIPR commands. Add special function AT commands.
V1.3	2008/9/25	Add some notice for TCP/IP
V1.2	2008/9/3	Combine TCP/IP and Cell information to one document, and add error code
V1.1	2008/8/15	Modify and remove some TCP/IP, Cell information command set.
V1.0	2008/7/14	TCP/IP, Cell information command set formal definition.
V0.1	2008/6/23	TCP/IP, Cell information command set draft definition.

Table of Content

<u>REVISION HISTORY</u>	<u>2</u>
<u>TABLE OF CONTENT</u>	<u>3</u>
<u>1. TCP/UDP AT COMMANDS INTERFACE</u>	<u>5</u>
1.1. DEFINE PDP CONTEXT +AIPDCONT	5
1.2. QUALITY OF SERVICE PROFILE (REQUESTED) +AIPQREQ.....	5
1.3. QUALITY OF SERVICE PROFILE (MINIMUM) +AIPQMIN	6
1.4. GPRS ATTACH OR DETACH +AIPA.....	6
1.5. OPEN SOCKET AND CONNECT TO INDICATED PORT AND IP +AIPO	7
1.6. CLOSE SOCKET +AIPC.....	8
1.7. SEND DATA +AIPW	8
1.8. QUERY DNR OR IP +AIPDNR	9
1.9. READ DATA +AIPR.....	10
1.10. QUERY SOCKET STATUS +AIPSST.....	11
<u>2. TCP/UDP EVENT INTERFACE.....</u>	<u>13</u>
2.1. RECEIVE TCP DATA +AIPRTCP	13
2.2. RECEIVE UDP DATA +AIPRUDP	13
<u>3. EXAMPLES: USING TCP/UDP AT COMMANDS AND EVENTS.....</u>	<u>15</u>
3.1. ATTACH TO GPRS NETWORK	175
3.2. USING TCP FUNCTION:	175
3.3. USING UDP FUNCTION:.....	16
<u>4. CELL INFORMATION AT COMMANDS.....</u>	<u>18</u>
4.1. CHECK SERVING CELL INFORMATION +CSCI	18
4.2. CHECK NEIGHBORING CELL INFORMATION +CNCI	18
4.3. CHECK DEDICATED MODE INFORMATION +CDMI.....	18
<u>5. SPECIAL FUNCTION AT COMMANDS</u>	<u>20</u>
5.1. CHECK FIRMWARE VERSION +SFUN=VER	20
5.2. POWER OFF MODULE +SFUN=OFF.....	20
5.3. RESET MODULE +SFUN=RST	20

5.4.	CSQ DETECTION +SFUN=CSQ.....	20
5.5.	CREG DETECTION +SFUN=CRG	20
5.6.	SIM CARD DETECTION +SFUN=SIM	21
5.7.	AT COMMAND READY DETECTION +SFUN=RDY	21
5.8.	SIM CARD ICCID +SFUN=SID	21
5.9.	NORMAL PLMN DIGITS ORDER +SFUN=OPN	21
5.10.	ORIGIANL PLMN DIGITS ORDER +SFUN=OPO	22
6.	AUTO BAUD RATE.....	23
6.1.	SET AUTO BAUD RATE MODE +IPR=0	23
6.2.	HOW TO DETECT BAUD RATE.....	23
7.	ERROR CODE.....	24
7.1.	TCP/IP AT COMMAND RETURN ERROR CODE.....	24
7.2.	QUERY LAST ERROR CODE +AIPESTAT	24
7.3.	RETURN ERROR CODE TABLE.....	25
7.4.	TCP/IP AT COMMAND RETURN EVENT	26

1. TCP/UDP AT Commands Interface

1.1. Define PDP context +AIPDCONT

Command	Possible response(s)
+AIPDCONT=<APN>,<USERNAME>,<PASSWORD>	+ AIPDCONT ="APN","Username","Password" ERROR
+ AIPDCONT =?	+ AIPDCONT: (list of supported parameters)

Supported parameter values

<APN>: APN Setting (Access point name)

<USERNAME>: ACSII String (provided by service provider)

<PASSWORD>: ACSII String (provided by service provider)

Response values

<APN>: Access point name

<Username>: user name

<Password>: password

1.2. Quality of Service Profile (Requested) +AIPQREQ

Command	Possible response(s)
+AIPQREQ=<precedence>,<delay>,<reliability>,<peak>,<mean>	+ AIPQREQ =3,4,3,1,31 ERROR
+ AIPQREQ =?	+ AIPQREQ: (list of supported parameters)

Supported parameter values

<precedence>: 0,1..3

<delay>: 0,1..4

<reliability>: 0,1..5

<peak>: 0,1..9

<mean>: 0,1..18,31

Response values

<precedence>: 0,1..3

<delay>: 0,1..4
 <reliability>: 0,1..5
 <peak>: 0,1..9
 <mean>: 0,1..18,31

1.3. Quality of Service Profile (Minimum) +AIPQMIN

Command	Possible response(s)
+AIPQMIN=<precedence>,<delay>,<reliability>,<peak>,<mean>	+ AIPQMIN =3,4,3,1,31 ERROR
+ AIPQMIN =?	+ AIPQMIN: (list of supported parameters)

Supported parameter values

<precedence>: 0,1..3
 <delay>: 0,1..4
 <reliability>: 0,1..5
 <peak>: 0,1..9
 <mean>: 0,1..18,31

Response values

<precedence>: 0,1..3
 <delay>: 0,1..4
 <reliability>: 0,1..5
 <peak>: 0,1..9
 <mean>: 0,1..18,31

1.4. GPRS active or inactive +AIPA

Command	Possible response(s)
+AIPA=<mode>	+AIPA=1: 1, "xxx.xxx.xxx.xxx",<sec>,<rx Bytes>,<tx Bytes> +AIPA=0: 0, <sec>,<rx Bytes>, <tx Bytes> ERROR
+AIPA?	+AIPA?: 1, "xxx.xxx.xxx.xxx",<sec>,<rx Bytes>, <tx Bytes> +AIPA?: 0
+AIPA=?	+AIPA: (list of supported parameters)

Supported parameter values

<mode>: 0,1 (0: inactive, 1: active)

Response values

AT+AIPA=1: +AIPA: 1, IP address, seconds for active, bytes for rx, bytes for tx

AT+AIPA=0: +AIPA: 0, seconds for inactive, bytes for rx, bytes for tx

AT+AIPA?: +AIPA: 1, IP address, seconds for active, bytes for rx, bytes for tx

or +AIPA: 0 not active

Restriction

You must close all sockets before inactive GPRS.

1.5. Open socket and connect to indicated port and IP +AIPO

Command	Possible response(s)
+AIPO=<socket id>,<lport>,<IP/DN>,<rport>,<p protocol>,<timeout>,<dnr>,<data transfer mode>,<retransmission timeout>,<retransmission back off>	+AIPO: Socket ID, local IP, local port, Remote IP, Remote port, Protocol, timeout, dnr, data transfer mode, retransmission timeout, retransmission back off, receive buffer window, send buffer window, receive data buffering size, send buffering size(size for no ack , size for not send out) ERROR
+AIPO?	+AIPO: (list of available socket id)
+AIPO=?	+AIPO: list of supported parameters)

Restriction

When opening a socket (socket id) and connect to the port of IP, if you didn't indicate a local port, a local port is automatically assigned.

Supported parameter values

<socket id>: 1..8

<lport>: local port (the port of this module)(range: 0~65535)

<IP/DN>: "xxx.xxx.xxx.xxx" remote IP or domain name (the IP or domain name of server),

<rport>: remote port (the port of server)(range: 0~65535)

<protocol>:0,1 communication protocol (0: TCP, 1: UDP)

<timeout>:0..75 seconds, the time duration wait for server's response when using TCP protocol. (default: 30 seconds)

<dnr>: 0: disable(default value),1: enable DNR function. If parameter #3 is domain name, you need set to 1 and enable DNR function to query actually IP address for that domain name. If parameter #3 is IP address, the DNR function

will be disable even the parameter value is 1.

<data transfer mode>:0,1,2. default value is 0. When set to 1, there will be a notice "+AIPRIPI: <socket id>, <protocol>" when the data is coming. You can use "AT+AIPR" to retrieve the data. When set to 2, there will be a notice "CONNECT" and the system is entering data mode. The data will direct send to the server on this mode. You can input "+++" to end this data mode and back to command.

<retransmission timeout>: 5120 ms ~65535 ms, default value is 64000ms

<retransmission back off>: 1~16 times, default value is 8.

Data transfer mode

0 Command mode: Use +AIPW to send data and wait +AIPRTCP/+AIPRUDP notice to get data immediately. Data is not keeping in buffer.

1 Request mode: Use AIPW to send data and wait +AIPRTCP/+AIPRUDP notice. Data is keeping in buffer and use +AIPR to retrieve data from buffer.

2 Data mode: System direct enter data mode and send/get data to/from server immediately. Use "+++" to leave data mode.

Response values

Case 1:

If socket ID has not been allocated for the remote connection.

AT+AIPW=1,,"Remote IP", Remote Port, Protocol

+AIPW: Socket ID, local IP address, local port, Remote IP, Remote port, protocol (0:TCP, 1:UDP), timeout, dnr, data transfer mode, retransmission timeout, retransmission back off, receive buffer window, send buffer window, receive data buffering size, send buffering size(size for no ack , size for not send out)

Note: If you don't input local port, system will assign free local port for this socket ID.

Note: If user does not input timeout period, system will assign default timeout period (30 sec.) for this socket ID connection.

Note: If user does not input data transfer mode, system will assign default transfer mode 0 for this socket ID connection.

Note: If user does not input retransmission timeout, system will assign default value 64000ms for this socket ID connection.

Note: If user does not input retransmission back off time, system will assign default value 8 for this socket ID connection.

Case 2:

AT+AIPW=1,,"Remote IP", Remote Port, Protocol, , ,2

+AIPO: Socket ID, local IP address, local port, Remote IP, Remote port, protocol (0:TCP, 1:UDP), timeout, dnr, data transfer mode, retransmission timeout, retransmission back off, receive buffer window, send buffer window, receive data buffering size, send buffering size(size for no ack , size for not send out)

CONNECT

.....DATA INPUT/OUTPUT

(input “+++” to end this data mode)

OK

Case 3:

If socket ID has been allocated for the remote connection.

AT+AIPO=1

+AIPO: Socket ID, local IP address, local port, Remote IP, Remote port, Protocol (0:TCP, 1:UDP), timeout, dnr, data transfer mode, retransmission timeout, retransmission back off, receive buffer window, send buffer window, receive data buffering size, send buffering size(size for no ack , size for not send out)

Case 4:

Query for which socket ID is free.

AT+AIPO?

+AIPO: 2,5,6,8 (System has free Socket ID 2,5,6,8 that can be allocated to remote connection)

1.6. Close socket +AIPC

Command	Possible response(s)
+AIPC=<Socket ID>	+AIPC: <Socket ID> ERROR
+AIPC?	+AIPC: (list of allocated socket ID)
+AIPC=?	+AIPC: (list of supported parameters)

Supported parameter values

<socket id>: Socket ID 1..8

Response values

Query which socket ID is allocated.

AT+AIPC?

+AIPC: 1,3,4,7 (System has allocated Socket ID 1,3,4,7 that currently used to some remote connections)

1.7. Send data +AIPW

Command	Possible response(s)
+AIPW=<Socket ID>,<data stream> or <data count>	+AIPW: <Socket ID>, <protocol>,<recv buffer length>,<send buffer length>,<write bytes> CONNECT ... OK ERROR
+AIPW?	+AIPW: <Socket ID>
+AIPW=?	+AIPW: (list of supported parameters)

Supported parameter values

<socket id>: Socket ID

<data stream>: the data to be sent (ASCII code). The maximum size of data stream in ASCII format is 1536 bytes.

<data count>: ready to send data size (only input data). This will direct enter to data mode and will auto leave when data size is meet the count. The maximum size is 65535 bytes.

Response values**Case 1:**

AT+AIPW=<socket id>,<data string>

+AIPW: <socket id>, <protocol>,<receive buffer length>,<send buffer length>,<write bytes>

<socket id>: Socket ID

<protocol>: 0: TCP, 1: UDP

<recv buf len>: current free receiving buffer length, value of UDP protocol is zero.

<send buf len>: current free sending buffer length, value of UDP protocol is zero.

<write bytes>: the length of data stream (bytes)

Case 2:

AT+AIPW=<socket id>,<data count>

CONNECT

.....DATA INPUT

(auto end this data mode when data size is meet)

OK

Notice

1. The data stream is in ASCII code. It means you must convert your data to ASCII code first and the length of data stream will be twice length of your data.

For example, if the string you want to send is "Hello", convert to its ASCII code 0x48 0x65 0x6C 0x6C 0x6F first. Then use AT+AIPW=1, "48656C6C6F".

(Socket ID 1) to send it out. The server will get string data "Hello".

2. Suggest you waiting until getting the response message +AIPRTCP before you send next data stream to avoid some unpredictable error condition.
3. Suggest you using ATE0 command to disable command echo before sending data stream.
4. To prevent this situation that there is not enough buffers to keep the sending or receiving data. When you are sending a big data steam (every data stream is more than 512X2 bytes) continuously. Suggest you had better wait or idle for one minute after every 5 minutes. Or you can send one data stream every 3 seconds continuously and don't need to wait.
5. The data count mode is useful for transfer binary data to remote side, if you do not want to use the AIPO data transfer mode 2.

1.8. Query DNR or IP +AIPDNR

Command	Possible response(s)
+AIPDNR=<Query type>, <DN/IP>	+AIPDNR: <IP address/Domain name> ERROR
+AIPDNR=?	+AIPDNR: (list of supported parameters)

Supported parameter values

<Query type>: 1 or 2.

Query type 1 (type "A") for IPv4 address. Query type 2 (type "PTR") for domain name from IPv4 address

<DN/IP>: If query type is 1, must fill DN parameter (domain name from IPv4 address).If query type is 2, must fill IP parameter (IPv4 address)

Response values

AT+AIPDNR=1,<DN>

+AIPDNR: "IP address"

AT+AIPDNR=2,<IP>

+AIPDNR: "Domain name"

1.9. Read data +AIPR

Command	Possible response(s)
+AIPR=<Socket ID>,<received buffer len>	+AIPR: <socket id>, <protocol>, <recv buf len>, <send buf len>, <recv bytes>, <ASCII data stream>

	ERROR
+AIPR?	+AIPR: <Socket ID>
+AIPR=?	+AIPR: (list of supported parameters)

Supported parameter values

<socket id>: Socket ID

<received buffer len>: the data stream length to be received (ASCII code). The maximum size of data stream in ASCII format is 1536 bytes.

Response values

AT+AIPR=<socket id>,< received buffer len >

+AIPR: <socket id>, <protocol>,<recv buf len>,<send buf len>,<recv bytes>,<ASCII data stream>

<socket id>: Socket ID

<protocol>: 0: TCP, 1: UDP

<recv buf len>: current free receiving buffer length, value of UDP protocol is zero.

<send buf len>: current free sending buffer length, value of UDP protocol is zero.

<recv bytes>: the length of data stream retrieved from receive buffer(bytes)

<ASCII data stream>: the data stream retrieved from receive buffer (ASCII code).

1.10. Query socket Status +AIPSST

Command	Possible response(s)
+AIPSST=<socket id, (A, a)>	+AIPSST: Socket ID, Socket Status, local IP address, local port, Remote IP, Remote port, Protocol (0:TCP, 1:UDP) ,dnr, data transfer mode, receive buffer window, send buffer window, receive data buffering size, send buffering size (size for no ack , size for not send out) ERROR
+AIPSST?	+AIPSST: (list of available socket id)
+AIPSST=?	+AIPSST: list of supported parameters)

Restriction

None.

Supported parameter values

<socket id>: 1..8, A, a

Response values

+AIPSST: Socket ID, Socket Status, local IP address, local port, Remote IP, Remote port, Protocol (0:TCP, 1:UDP) , dnr, data transfer mode, receive buffer window, send buffer window, receive data buffering size, send buffering size (size for no ack , size for not send out)

Note: Socket Status has 3 status:

- 0: close and not allocated socket resource.
- 1: Connect and allocated socket resource
- 2: Close by remote connection. Need to close by AT+AIPC to disconnect the connection and release the socket resource

Case 1:

If socket ID has not been allocated for the remote connection.

AT+AIPSST= Socket ID

+AIPSST: Socket ID, 0

Case 2:

If socket ID has been allocated and connected to the remote host.

AT+AIPSST= Socket ID

+AIPSST: Socket ID, 1, local IP address, local port, Remote IP, Remote port, Protocol (0:TCP, 1:UDP) , dnr, data transfer mode, receive buffer window, send buffer window, receive data buffering size, send buffering size (size for no ack , size for not send out)

Case 3:

If socket ID does not close, but has been closed by remote connection.

AT+AIPSST= Socket ID

+AIPSST: Socket ID, 2, local IP address, local port, Remote IP, Remote port, Protocol (0:TCP, 1:UDP) , dnr, data transfer mode, receive buffer window, send buffer window, receive data buffering size, send buffering size (size for no ack , size for not send out)

Case 4:

List all socket status.

AT+AIPSST=A or AT+AIPSST=a, AT+AIPSST?

+AIPSST: 1, (0..2), (local IP address, local port, Remote IP, Remote port, Protocol (0:TCP, 1:UDP) , dnr, data transfer mode, receive buffer window, send buffer window, receive data buffering size, send buffering size (size for no ack , size for not send out))

+AIPSST: 2,

.....

+AIPSST: 8, (0..2), (local IP address, local port, Remote IP, Remote port, Protocol (0:TCP, 1:UDP), dnr, data transfer mode, receive buffer window, send buffer window, receive data buffering size, send buffering size (size for no ack , size for not send out))

2. TCP/UDP Event Interface

2.1. Receive TCP data +AIPRTCP

Event
+AIPRTCP: <socket id>, <protocol>, <recv buf len>,<send buf len>,<recv bytes>,<ASCII data stream>

Note: *The module will output this message automatically when it gets the TCP data.

Parameter values

(The TCP data received)

<socket id>: Socket ID

<protocol>: 0 //TCP

<recv buf len>: current free receiving buffer length.

<send buf len>: current free sending buffer length.

<recv bytes>: data stream length(byte)

<ASCII data stream>: The received data (ASCII code), length is <recv bytes> x 2

Note: The maximum size of each ASCII data string is 1536 bytes. It means that it can actually receive 768 bytes data during each time. Therefore, if a 1000 bytes data packet is received, system will break it into two data parts, one with 768 bytes, and another with 232 bytes.

2.2. Receive UDP data +AIPRUDP

Event
+AIPRUDP: <socket id>, <protocol>, <recv buf len>,<send buf len>,<recv bytes>,<ASCII data stream>

Note: * The module will output this message automatically when it gets the UDP data.

Parameter values

(The UDP data received)

<socket id>: Socket ID

<protocol>: 1 //UDP

<recv buf len>: This value of UDP protocol is always zero.

<send buf len>: This value of UDP protocol is always zero

<recv bytes>: the length of data stream (byte)

<ASCII data stream>: The received data (ASCII code) length is <recv bytes> x 2

Note: The maximum size of each ASCII data string is 1536 bytes. It means that it can actually receive 768 bytes data during each time. Therefore if a 1000 bytes data packet is received, system will break it to two data parts, one with 768 bytes, and another with 232 bytes

Confidential

3. Examples: Using TCP/UDP AT Commands and Events

3.1. Attach to GPRS Network

Start up GSM/GPRS module and send the AT commands to make sure system is ready to attach to GPRS network.

Sample procedure:

AT	
OK	// Make sure system is ready
AT+CSQ	
+CSQ: 25,99	
OK	// Make sure module has camped on GSM network
AT+CPIN?	
+CPIN: READY	
OK	// Make sure SIM card is ready
AT+AIPDCONT="CMNET" or "Internet"	// Input APN name "CMNET" to use GPRS network
+AIPDCONT: "CMNET","", ""	
OK	
AT+AIPA=1	// Attach to GPRS Network
+AIPA: 1,221.120.5.138,0,0,0	// Response status includes local IP address "221.120.5.138" that system assigned and connection time that is now 0 second, rx 0 Byte, and tx 0 Byte.
OK	

3.2. Using TCP function:

1. Connection:

AT+AIPO=1,,"203.160.10.251",12000,0	// Establish a connection to remote IP "203.168.10.251", make sure that IP address is a physical address but not virtual. (Note: 192.xxx.xxx.xxx is virtual IP)
+AIPO: 1,"116.59.62.91",5678,"203.160.10.251",12000,0,30,0,0,64000,8,7300,65535,0,0(0,0)	// If a local port is not assigned, system will assign free local port. Ex: This case the local port is 5678, local IP is "116.59.62.91", connection timeout is 30 seconds, DNR is 0, transfer mode is 0,

retransmission timeout is 64000ms, retransmission back off time is 8, receive. Window is 7300, sending window is 65535, receive data buffering size is 0, send data buffering size is 0, (sending buffering no ack size is 0, not send buffering data size is 0).

OK

2. Send Data:

AT+AIPW=1,"414243444546"

// Send ASCII String "414243444546". Actually the data is 0x41 0x42 0x43 0x44 0x45 0x46. Total data length is 6 bytes.

+AIPW: 1,0,7300,5840,6

OK

or

AT+AIPW=1,6

// Send for 6 bytes of total data length

CONNECT

123456

Input "123456" data into this connection from PC or MCU through UART interface

OK

3. Received Data:

System will send event through AT interface to notify that system has received data from the remote server as bellow:

+AIPRTCP:1,0,7300,5840,6,"41424344546"

// Free remaining receiving buffer is 7300. Sending buffer is 5840.

4. Remote Server Closes Connection:

+AIPC: 1

// When this event is received, it means that remote server has been closed. Therefore you have to close module socket ID and re-connect to remote server or free this socket for next connection.

AT+AIPC=1

// Close socket ID 1 and free this socket for next connection.

+AIPC: 1

OK

5. Close Connection:

AT+AIPC=1

Close socket ID 1 and disconnect remote server and free this socket for next connection.

+AIPC: 1

OK

3.3. Using UDP function:

1. Connection:

AT+AIPO=1,,"203.160.10.251",10000,1

// Establish a connection to remote "IP 203.160.10.251", make sure that IP address is a physical address but not virtual. (Note: 192.xxx.xxx.xxx is virtual IP)

+AIPPO:

1, "116.59.62.91",5679,"203.160.10.251",
10000,1,,,0,1,,,0,0,0,0(0,0)

If a local port is not assigned, system will assign a free local port. Ex: This case is local port 5679, local IP is "116.59.62.91", no connection timeout, no DNR, transfer mode is 0, none of retransmission timeout, none of retransmission back off, none of receive. Window is 0, none of sending window is 0, none of receive data buffering size is 0, none of send data buffering size is 0, (no ne of sending buffering no ack size is 0, none of not send buffering data size is 0.

OK

2. Send Data:

AT+AIPW=1,"4142434444546"

Send ASCII String "4142434444546". Actually the data is 0x4142434444546. Total data length is 6 bytes.

+AIPW:1,1,0,0,6

OK

or

AT+AIPW=1,6

// Send for 6 bytes of total data length

CONNECT**123456**Input "123456" data into this connection from PC or
MCU through UART interface**OK**

3. Receive Data:

System will send event through AT interface to notify that system has received data from the remote server as bellow:

```
+AIPRUDP:1,1,0,0,6,"414243444546" // Value of receiving buffer and sending buffer at  
UDP protocol is 0. It means that it can receive UDP  
data until system has no free buffer.
```

4. Close Connection:

AT+AIPC=1Close socket ID 1, disconnect remote server and
free this socket for next connection**+AIPC: 1****OK**

4. Cell information AT Commands

4.1. Check Serving Cell Information +CSCI

Command	Possible response(s)
+CSCI	+CSCI: Info of serving cell
+CSCI?	ERROR
+CSCI=?	+CSCI: (list of supported serving cell info)

Supported parameter values

None

Response values

+CSCI: BCCH, BSIC, LAC, RAC, Rxlev, Cell ID, MCC, MNC

4.2. Check Neighboring Cell Information +CNCI

Command	Possible response(s)
+CNCI=<index>	+CNCI: <no> +CNCI: Info of neighbor cell <index> ERROR
+CNCI?	+CNCI: (list of supported neighbor cell info) ERROR
+CNCI=?	+CNCI: (each parameter values)

Restriction

If neighboring index does not exist, return error.

Supported parameter values

<no>: total number of neighboring cells. (0 .. the maximum neighboring cells)

<index>: the index of neighboring cell. (0.. <no-1>) or A or a

Index=A or a show info of all neighboring cells

Response values

+CNCI: Index of Cell, BCCH, BSIC, LAC, Rxlev, Cell ID, MCC, MNC

4.3. Check Dedicated Mode Information +CDMI

Command	Possible response(s)
+CDMI	+CDMI: Info of dedicated mode
+CDMI?	ERROR
+CDMI=?	+CDMI: (list of supported dedicated mode

	info)
--	-------

Supported parameter values

None

Response values

+CDMI: RxLevel, TxLevel, RxQualityFull, FreqHopInd, Cplnd

5. Special Function AT Commands

5.1. Check Firmware Version *+SFUN=VER*

Command	Possible response(s)
+SFUN=VER	FW Ver: <version no>

Response values

<version no>: 16 digitals firmware version no

5.2. Power Off Module *+SFUN=OFF*

Command	Possible response(s)
+ SFUN =OFF	None

* Power off (shut down) the module. It will need some time to process this procedure.

Response values

None

5.3. Reset Module *+SFUN=RST*

Command	Possible response(s)
+ SFUN =RST	None

* Reset (restart) the module. It will need some time to process the restart procedure.

Response values

None

5.4. CSQ Detection *+SFUN=CSQ*

Command	Possible response(s)
+ SFUN =CSQ	CSQ detect mode

* GPIO3 will set to high only when the value of CSQ greater than 15.

Response values

GPIO3: CSQ detect mode

5.5. CREG Detection *+SFUN=CRG*

Command	Possible response(s)
+ SFUN =CRG	CREG detect mode.

* GPIO3 will set to high only when the value of CREG is set to 1 (camp on network).

Response values

GPIO3: CREG detect mode.

5.6. SIM Cart Detection +SFUN=SIM

Command	Possible response(s)
+ SFUN =SIM	SIM detect mode

* GPIO10 will set to high when the SIM card is exist and set to low when SIM card is unexist.

* It is must to restart the module after use this command.

Response values

GPIO10: HW SIM detect mode

5.7. AT Command Ready Detection +SFUN=RDY

Command	Possible response(s)
+ SFUN =RDY	AT command ready mode

* GPIO10 will set to high when the system is ready to input AT command when module start up.

* It is must to restart the module after use this command.

Response values

GPIO10: Normal ready mode

5.8. SIM card ICCID +SFUN=SID

Command	Possible response(s)
+ SFUN =SID	SIM ICCID: <id no>

Response values

<id no>: 10 digitals of SIM card ICCID no

5.9. Normal PLMN Digits Order +SFUN=OPN

Command	Possible response(s)
+ SFUN =OPN	Normal PLMN digits order (no inverse)

* PLMN includes MCC and MNC.

* Both AT+COPS=? and AT+COPS=? will change the MCC and MNC digits order

* Use AT+COPS=3,2 and AT+COPS=? to test

Response values

COPS: MCC MNC digits normal mode

5.10. *Original PLMN Digits Order +SFUN=OPO*

Command	Possible response(s)
+ SFUN =OPO	Original PLMN digits order (inverse)

* PLMN includes MCC and MNC.

* Use AT+COPS=3,2 and AT+COPS? to test

Response values

COPS: MCC MNC digits original mode

6. Auto Baud Rate

6.1. Set Auto Baud Rate Mode +IPR=0

Command	Possible response(s)
+IPR=0	OK +Auto BR: (Baud rate)
+IPR= 	OK
+IPR?	+IPR: (current Baud rate) +IPR: 0 (auto Baud rate)
+IPR=?	+IPR: (list of all available Baud rate)

Restriction

Only the module supported Baud rates are allowed to be set.

Supported parameter values

: 0,1200,2400,4800,9600,14400,19200,28800,38400,57600,115200

br=0, module will be changed to auto Baud rate mode.

Auto Baud rate mode only support 6 rates: 4800,9600,19200,38400,57600,115200

Response values

+IPR:

+Auto BR:

6.2. How to Detect Baud Rate

The timing of system to detect Baud rate

If IPR is set to 0, system will enter the auto detect mode. Only when boot up or reboot the module (or after the moment of use AT command AT+IPR=0), system will active the auto detect Baud rate function and change to detect mode.

How to detect Baud rate

When system is already change to detect mode. Send "AT" and "<CR>" to module. Module will detect and auto change to correct Baud rate. And response +Auto BR: (Baud rate).

Note

You can not send "AT" and "<CR>" when system is boot up or reboot until system is ready (wait about 5 seconds). If you send wrong command (not AT<CR>) or wait less than 5 seconds, or other reason system no response +Auto BR. You must reboot the module and try again.

After all, you can resend "AT"+"<CR>" again to check the system is change to correct Baud rate or not. The module should response "OK" this time.

The time interval between send each code ("A", "T" and "<CR>") is about 200ns. You had better send each code individually when you use another programming system to do this.

7. Error Code

7.1. TCP/IP AT command return error code

ERROR	ID	DESCRIPTION
PARAMETERS ERROR	300	Input AT commands format of TCPIP is incorrect
GPRSNETESTERROR	301	Does not connect to GPRS network
OPENSOCKETERROR	302	Socket opening error (socket was not opened, or the socket is used for other connection)
BINDSOCKETERROR	303	Port or IP address error
CONNECTSOCKETERROR	304	Fail connection at this socket

7.2. Query last error code **+AIPESTAT**

Command	Possible response(s)
+AIPESTAT +AIPESTAT?	+AIPESTAT: <ERROR ID>
+AIPESTAT=?	(+AIPESTAT: <ERROR ID>)

Supported parameter values

None

Response values

<ERROR ID>: the error id. Check the description of this id in the error code table as bellow.

7.3. Return error code table

ERROR	ID	DESCRIPTION
BSD_EWOULDBLOCK	201	the socket is marked non-blocking and the requested action will block
BSD_EINPROGRESS	202	operation is running
BSD_ENOSTOCK	203	socket identifier is not a socket
BSD_EMSGSIZE	204	msg sent by the IP Stack User is bigger than MTU
BSD_EPROTONOSUPPORT	205	the protocol specified is not supported within this domain
BSD_EOPNOTSUPP	206	option is not supported (can happen when using <code>bsd_ioctl</code>)
BSD_EAFNOSUPPORT	207	wrong address family
BSD_EADDRNOTAVAIL	208	Bind to Invalid local IP address
BSD_EADDRINUSE	209	Local Port of IP address is already in use
BSD_ENETDOWN	210	link is down
BSD_ECONNABORTED	211	the connection is aborted
BSD_ECONNRESET	212	the connection has been reset by peer
BSD_ENOBUFS	213	no buffer free in PPP
BSD_EISCONN	214	the TCP connection is not closed
BSD_ENOTCONN	215	the socket is not connected (<code>bsd_connect</code> has not been called)
BSD_ETIMEDOUT	216	the connection has been aborted because timeout occurred
BSD_ECONNREFUSED	217	the remote system refused the connection
BSD_EHOSTUNREACH	218	host unreachable : no application is running on the net side
BSD_EINVAL	219	the parameters given in the last call of <code>bsd</code> function are invalid
BSD_EINTR	220	operation was interrupted before any data were available or sent
BSD_ENOMEM	221	no more memory available
BSD_ENOTRECORDED	222	the application is not registered : call <code>_bsd_stack_up</code> to do it
BSD_EALREADY	223	stack is already up
BSD_ESOCKTNOSUPPORT	224	the socket type is not supported
BSD_ENOTBOUND	225	the socket is not bound (function <code>bsd_bind</code> has not been called)
BSD_ESNDWNCLOSED	226	the remote TCP advertise a null windows, so that we enter in persist state : no data could be sent on this TCP connection by local TCP
BSD_ESNDWNDFULL	227	the sending windows advertised by remote TCP is full : we must wait for acknowledgment from remote TCP to go on transfer in progress

7.4. TCP/IP TCP server AT command Event

EVENT of AT for TCPIP		DESCRIPTION
+AIPCI: socket id		TCPIPSockRemoteClosed
+AIPEVNTI: socket id, 3		TCPIPBufferFree
+AIPEVNTI: socket id, 8		TCPIPSockSendWndOpened
+AIPEVNTI: socket id, 10		TCPIPSockHostUnreach

2010 1201 TCPIP NEW Features

8. Update 1: TCP/UDP AT Commands Interface

8.1. Open socket and connect to indicated port and IP +AIPO

Command	Possible response(s)
+AIPO=<socket id>,<lport>,<IP/DN>,<rport>,<protocol>,<timeout>,<dnr>,<data transfer mode>,<retransmission timeout>,<retransmission back off>	+AIPO: Socket ID, local IP, local port, Remote IP, Remote port, Protocol, timeout, dnr, data transfer mode, retransmission timeout, retransmission back off, receive buffer window, send buffer window, receive data buffering size, send buffering size(size for no ack , size for not send out) ERROR
+AIPO?	+AIPO: (list of available socket id)
+AIPO=?	+AIPO: list of supported parameters)

Restriction

When opening a socket (socket id) and connect to the port of IP, if you didn't indicate a local port, a local port is automatically assigned.

Supported parameter values

<socket id>: 1..8

<lport>: local port (the port of this module)(range: 0~65535)

<IP/DN>: "xxx.xxx.xxx.xxx" remote IP or domain name (the IP or domain name of server),

<rport>: remote port (the port of server)(range: 0~65535)

<protocol>:0,1 communication protocol (0: TCP, 1: UDP)

<timeout>:0..75 seconds, the time duration wait for server's response when using TCP protocol. (default: 30 seconds)

<dnr>: 0: disable(default value),1: enable DNR function. If parameter #3 is domain name, you need set to 1 and enable DNR function to query actually IP address for that domain name. If parameter #3 is IP address, the DNR function will be disable even the parameter value is 1.

<data transfer mode>:0,1,2. default value is 0. When set to 1, there will be a notice "+AIPRIPI: <socket id>, <receive buffering data size>, <receive total data

size that has been received from Application>” when the data is coming. You can use “AT+AIPR” to retrieve the data. When set to 2, there will be a notice “CONNECT” and the system is entering data mode. The data will direct send to the server on this mode. You can input “+++” to end this data mode and back to command.

<retransmission timeout>: 5120 ms ~65535 ms, default value is 64000ms

<retransmission back off>: 1~16 times, default value is 8.

Data transfer mode

0 Command mode: Use +AIPW to send data and wait +AIPRTCP/+AIPRUDP notice to get data immediately. Data is not keeping in buffer.

1 Request mode: Use AIPW to send data and wait +AIPRTCP/+AIPRUDP notice. Data is keeping in buffer and use +AIPR to retrieve data from buffer.

2 Data mode: System direct enter data mode and send/get data to/from server immediately. Use “+++” to leave data mode.

Response values

Case 1:

If socket ID has not been allocated for the remote connection.

AT+AIPW=1,,"Remote IP", Remote Port, Protocol

+AIPW: Socket ID, local IP address, local port, Remote IP, Remote port, protocol (0:TCP, 1:UDP), timeout, dnr, data transfer mode, retransmission timeout, retransmission back off, receive buffer window, send buffer window, receive data buffering size, send buffering size(size for no ack , size for not send out)

Note: If you don't input local port, system will assign free local port for this socket ID.

Note: If user does not input timeout period, system will assign default timeout period (30 sec.) for this socket ID connection.

Note: If user does not input data transfer mode, system will assign default transfer mode 0 for this socket ID connection.

Note: If user does not input retransmission timeout, system will assign default value 64000ms for this socket ID connection.

Note: If user does not input retransmission back off time, system will assign default value 8 for this socket ID connection.

Case 2:

AT+AIPW=1,,"Remote IP", Remote Port, Protocol, , ,2

+AIPW: Socket ID, local IP address, local port, Remote IP, Remote port, protocol (0:TCP, 1:UDP), timeout, dnr, data transfer mode, retransmission timeout,

retransmission back off, receive buffer window, send buffer window, receive data buffering size, send buffering size(size for no ack , size for not send out)

CONNECT

.....DATA INPUT/OUTPUT

(input “+++” to end this data mode)

OK

Case 3:

If socket ID has been allocated for the remote connection.

AT+AIPO=1

+AIPO: Socket ID, local IP address, local port, Remote IP, Remote port, Protocol (0:TCP, 1:UDP), timeout, dnr, data transfer mode, retransmission timeout, retransmission back off, receive buffer window, send buffer window, receive data buffering size, send buffering size(size for no ack , size for not send out)

Case 4:

Query for which socket ID is free.

AT+AIPO?

+AIPO: 2,5,6,8 (System has free Socket ID 2,5,6,8 that can be allocated to remote connection)

8.2. Send data +AIPW

Command	Possible response(s)
+AIPW=<Socket ID>,<data stream> or <data count>	+AIPW: <Socket ID>, <protocol>,<recv buffer length>,<send buffer length>,<write bytes> CONNECT ... OK ERROR +AIPW: <Socket ID>, <data transfer mode>,<send buffer data window>,<send buffering data size (data size for no ack , data size for not yet send out)>,<send total data size that has been ack from remote host>
+AIPW?	+AIPW: <Socket ID>

+AIPW=?

+AIPW: (list of supported parameters)

Supported parameter values

<socket id>: Socket ID

<data stream>: the data to be sent (ASCII code). The maximum size of data stream in ASCII format is 1536 bytes.

<data count>: ready to send data size (only input data). This will direct enter to data mode and will auto leave when data size is meet the count. The maximum size is 65535 bytes.

Response values**Case 1:**

AT+AIPW=<socket id>,<data string>

+AIPW: <socket id>, <protocol>, <receive buffer length>, <send buffer length>, <write bytes>

<socket id>: Socket ID

<protocol>: 0: TCP, 1: UDP

<recv buf len>: current free receiving buffer length, value of UDP protocol is zero.

<send buf len>: current free sending buffer length, value of UDP protocol is zero.

<write bytes>: the length of data stream (bytes)

Case 2:

AT+AIPW=<socket id>,<data count>

CONNECT

.....DATA INPUT

(auto end this data mode when data size is meet)

OK

Case 3:

Query the connection send data status

AT+AIPW=<socket id>

+AIPW: socket id, data transfer mode, send buffer data window, send buffering data size (data size for no ack , data size for not yet send out), send total data size that has been ack from remote host.

Note: send buffer data window means that currently system can accept total data size and sent them to remote host.. It depends on connection negotiation of remote host.

Note: sent buffering data size == data size for no ack + data size for not yet sent out

Note: sent total ack data size + sent buffering data size == send total data size

Notice

1. The data stream is in ASCII code. It means you must convert your data to ASCII code first and the length of data stream will be twice length of your data. For example, if the string you want to send is "Hello", convert to its ASCII code 0x48 0x65 0x6C 0x6C 0x6F first. Then use AT+AIPW=1, "48656C6C6F". (Socket ID 1) to send it out. The server will get string data "Hello".
2. Suggest you waiting until getting the response message +AIPRTCP before you send next data stream to avoid some unpredictable error condition.
3. Suggest you using ATE0 command to disable command echo before sending data stream.
4. To prevent this situation that there is not enough buffers to keep the sending or receiving data. When you are sending a big data steam (every data stream is more than 512X2 bytes) continuously. Suggest you had better wait or idle for one minute after every 5 minutes. Or you can send one data stream every 3 seconds continuously and don't need to wait.
5. The data count mode is useful for transfer binary data to remote side, if you do not want to use the AIPO data transfer mode 2.

8.3. Read data +AIPR

Command	Possible response(s)
+AIPR=<Socket ID>,<received buffer len>	+AIPR: <socket id>, <protocol>, <recv buf len>, <send buf len>, <recv bytes>, <ASCII data stream> ERROR
+AIPR?	+AIPR: <Socket ID>
+AIPR=?	+AIPR: (list of supported parameters)

Supported parameter values

<socket id>: Socket ID

<received buffer len>: the data stream length to be received (ASCII code). The maximum size of data stream in ASCII format is 1536 bytes.

Response values**Case 1:**

AT+AIPR=<socket id>,< received buffer len >

+AIPR: <socket id>, <protocol>,<recv buf len>,<send buf len>,<recv bytes>,<ASCII data stream>

<socket id>: Socket ID

<protocol>: 0: TCP, 1: UDP

<recv buf len>: current free receiving buffer length, value of UDP protocol is zero.

<send buf len>: current free sending buffer length, value of UDP protocol is zero.

<recv bytes>: the length of data stream retrieved from receive buffer(bytes)

<ASCII data stream>: the data stream retrieved from receive buffer (ASCII code).

Case 2:

Query the connection receive data status

AT+AIPR=<socket id>

+AIPR: socket id, data transfer mode, receive buffer data window, receive buffering data size, receive total data size that has been received from Application.

Note: receive buffer data window means that currently system can receive total data size from remote host. The size is fixed at 7300 bytes, but It could be modified during connection negotiation.

8.4. Receive TCP/UDP data-in message +AIPRIPI

Event
+AIPRIPI: <socket id>,<receive buffering data size>, <receive total data size>

Note: *The module will output this message automatically when it gets the TCP/UDP data.

Parameter values

(The TCP/UDP data-in message)

<socket id>: Socket ID

<receive buffering data size >: received TCP or UDP data and store them in buffer.

<receive total data size>: receive total data size that has been received from Application.

9. Update 2: New TCP/UDP AT Commands Interface

9.1. Set TCPIP option +AIPSET

Command	Possible response(s)
+AIPSET=<Client socket ID>, <Option Type>, <Parameters>	+AIPSET: client socket id, option type, current parameters ERROR
+AIPSET?	+ AIPSET: (list of all client socket option parameters status)
+AIPSET=?	+AIPSET: <Socket ID>,<Option type>,<Parameters>

Supported parameter values

<Client socket ID>: Server Socket ID 1..8

<Option>:

Option Type <1>: transfer data mode

Option Type <2>: retransmission timeout

Option Type <3>: retransmission back off

Option Type <4>: enable ack option of sending data (ex: AT+AIPW)

<Parameters>:

Option Type <1>: Data transfer mode 0,1,2

Option Type <2>: 5120 ms~65535 ms, default value is 64000ms

Option Type <3>: 1~16 times, default value is 8

Option Type <4>: 0,1; 0 for ack disable, 1 for ack enable

Response values

<Client socket ID>: Server Socket ID 1..8

<Option>:

Option Type <1>: transfer data mode

Option Type <2>: retransmission timeout

Option Type <3>: retransmission back off

Option Type <4>: enable ack option of sending data (ex: AT+AIPW)

<Parameters>:

Option Type <1>: Data transfer mode 0,1,2

Option Type <2>: 5120 ms~65535 ms, default value is 64000ms

Option Type <3>: 1~16 times, default value is 8

Option Type <4>: 0,1; 0 for ack disable, 1 for ack enable

Case 1:

AT+AIPSET=1,1,2

+AIPSET: 1,1,2

Note: Set client socket id 1, and option 1 (data transfer mode) to data transfer mode 2.

Case 2:

AT+AIPSET=1,1

+AIPSET: 1,1,2

Note: Get client socket id 1, and parameters of option 1 (data transfer mode), and now data transfer mode is 2.

Case 3:

AT+AIPSET=1,2,5120

+AIPSET: 1,2,5120

Note: Set client socket id 1, and option 2 (retransmission timeout) to 5120ms.

Case 4:

AT+AIPSET=1,2

+AIPSET: 1,2,5120

Note: Get client socket id 1, and parameters of option 2 (retransmission timeout), and now retransmission timeout is 5120.

Case 5:

AT+AIPSET=1,3,5

+AIPSET: 1,3,5

Note: Set client socket id 1, and option 3 (retransmission back off) to 5 times.

Case 6:

AT+AIPSET=1,3

+AIPSET: 1,3,5

Note: Get client socket id 1, and parameters of option 3 (retransmission back off), and now retransmission back off is 5 times.

Case 7:

AT+AIPSET=1,4,1

+AIPSET: 1,4,1

Note: Set client socket id 1, and option 4, and ack option to enable.

Case 8:

AT+AIPSET=1,4

+AIPSET: 1,4,1

Note: Get client socket id 1, and parameters of option 4, and now ack option is enable.

10. Update 3: Error Code

10.1. TCP/IP AT command return error code

ERROR	ID	DESCRIPTION
PARAMETERS ERROR	300	Input AT commands format of TCPIP is incorrect
GPRSNETESTERROR	301	Does not connect to GPRS network
OPENSOCKETERROR	302	Socket opening error (socket was not opened, or the socket is used for other connection)
BINDSOCKETERROR	303	Port or IP address error
CONNECTSOCKETERROR	304	Fail connection at this socket
READDATAEERROR	305	Read data error
WRITEDATAEERROR	306	Write data error

10.2. update and new event for ACK of AIPW

EVENT of AT for TCPIP	DESCRIPTION
+AIPRIPI: <socket id>, <receive buffering data size>, <receive total data size>	Event that notify the TCP or UDP data-in, and need application use "AT+AIPR" command to receive data
+AIPWACK:<socket id>,<send buffering data	Event that report ack data, and sending data status if any sending data has been ack.(PS: AT+AIPSET to

size (data size for no ack , data size for not yet send out)>, <send total data size that has been ack from remote host>		enable the function.)
--	--	------------------------

11. TCP Server AT Commands Interface and events

11.1. *Open server socket and accept remote client connection*

+ATCPSEVERO

Command	Possible response(s)
+ATCPSEVERO=<server listen socket id>,<server listen port>,<max client connections>	+ATCPSEVERO: server listen socket ID, server socket status, server local IP, server listen port, max client connections, current client

	connections, waiting client connections ERROR
+ATCPSEVERO?	+ ATCPSEVERO: (list of available server socket id)
+ATCPSEVERO=?	+ ATCPSEVERO: list of supported parameters

Restriction

When opening a server socket (socket id) and set server listen port, if you didn't set max client connections, server would set automatically assign default max client connections to 8.

Supported parameter values

< server listen socket id>: 1..4

< server listen port >: server listen port (the local port of this module)(range: 0~65535)

< max client connections >: 1~8, default value is 8

Data transfer mode of client connection

System default **data mode 2** for client connection..

Response values

< server listen socket id>: 1..4

< server socket status >

< server local IP >

< server listen port > server listen port (the local port of this module)(range: 0~65535)

< max client connections >: 1~8, default value is 8

< current client connections >: 0~max client connections

< waiting client connections >: 0~3, system could accept 3 waiting client connections at the same time.

Case 1:

If server socket ID has not assigned max client connection.

AT+ATCPSEVERO=1, 10000

+ATCPSEVERO: 1,1,"110.26.63.152",12000,8,0,0

Note: If you don't input max connections system will assign max connections to 8. System can accept 8 client connections at the same time.

Case 2:

If server socket ID has assigned max client connection.

AT+ATCPSERVEO=2, 12000, 1

+ATCPSERVERO: 1,1,"110.26.63.152",12000,1,0,0

Note: The TCP server input max client connection. System only can accept 1 client connections at the same time.

Case 3:

Only server socket ID has assigned.

AT+ATCPSERVEO=1

+ATCPSERVERO: 1,1,"110.26.63.152",12000,8,0,0

Note: The TCP server socket status response: server socket id is 1, server socket status is ready (1), server IP address is "110.26.63.152", server listen port is 12000, there are 8 max client connections, there is only one current connection, and there is only one waiting connection.

Case 4:

Only server socket ID has assigned and accept one client connection.

AT+ATCPSERVEO=2

+ATCPSERVERO: 2,1,"110.26.63.152",12000,1,1,0

+ATCPCLNTSID: 1

Note: +ATCPCLNTSID means that the server socket id 2 has accepted 1 client connections, and the client connection socket id is 1. Then application can use AIPO, AIPW, AIPR, AIPC, AIPSST AT command sets to control the client connection.

Note: reference +ATCPCLNTIN event for client connection.

Note: reference +ATCPSERVERAI event for waiting client connection.

Case 5:

Only server socket ID has been closed , but still have one client connection

AT+ATCPSERVEO=2

+ATCPSERVERO: 2,2," 110.26.63.152""",12000,1,1,0

+ATCPCLNTSID: 1

Note: The server socket id has not been real closed. And it has one more remote connection. Therefore, second parameter of "+ATCPSERVERO" is 2, and means that it has enter close mode, and not open mode 1.

Case 6:

Query free server socket id.

AT+ATCPSERVERO?

+ATCPSERVERO: 3,4

Note: System can open 4 TCP Server. Now free server socket id is 3, 4.

Case 7:

Query ATCPSEVERO parameters

AT+ATCPSEVERO=?

+ATCPSEVERO: <SID>,(1..65535),(1..8)

Note: Server listen port is from 1 to 65535, and max client connections are from 1 to 8.

11.2. Close socket +ATCPSEVERC

Command	Possible response(s)
+ATCPSEVERC=<Socket ID>	+ATCPSEVERC: <Server Socket ID> ERROR
+ATCPSEVERC?	+ ATCPSEVERC: (list of allocated server socket ID)
+ATCPSEVERC =?	+ ATCPSEVERC: (list of supported parameters)

Supported parameter values

<server socket id>: Server Socket ID 1..4

Response values

Query which socket ID is assigned.

AT+ATCPSEVERC?

+ATCPSEVERC: 1,2 (System has assigned Socket ID 1,2 that currently used to TCP server.)

Case 1:

If server socket ID has not accepted any client connections now.

AT+ATCPSEVERC=1

+ATCPSEVERC: 1

Note: If the server socket id has not accepted any remote connections now, the TCP server would be closed normally.

Case 2:

If server socket ID has accepted one more client connections.

AT+ATCPSEVERC=2

ERROR: 311

+ATCPSEVERCI: 1

Note: If the server socket id has accepted one more remote connections now.

TCP server socket would be not closed immediately, and response ERROR 311.

Later, the TCP server socket would enter close mode, and response “+ATCPSERVERCI: 1” event to notify that the socket has been closed halfly. System need to close client socket id 1 to disconnect all remote connection about the server socket id. And then run the Case 1 to real close server socket id.

Note: About query the server socket id connection status, and could reference Case 5 of “+ATCPSEVERO”

11.3. Query server socket +ATCPSERVERS

Command	Possible response(s)
+ATCPSERVERS=<Socket ID>	+ATCPSERVERS: server listen socket ID, server socket status, server local IP, server listen port, max client connections, current client connections, waiting client connections +ATCPCLNTO: server listen socket ID,client socket ID,client IP address, client port ERROR
+ATCPSERVERS?	+ ATCPSERVERS: (list of allocated server socket ID, and client socket ID connection)
+ATCPSERVERS =?	+ ATCPSERVERS (list of supported parameters)

Supported parameter values

<server socket id>: Server Socket ID 1..4

Response values

Server socket Part: +ATCPSERVERS

< server listen socket id>: 1..4

< server socket status >: 1 for open status, 2 for close status, 0 for idle status (not open)

< server local IP >: server local IP address

< server listen port > server listen port (the local port of this module)(range: 0~65535)

< max client connections >: 1~8, default value is 8
 < current client connections >: 0~max client connections
 < waiting client connections >: 0~3, system could accept 3 waiting client connections at the same time

Client socket Part: +ATCPCLNTO

< server listen socket id>: 1..4
 < client socket id>: 1..8
 < client local IP >: remote client IP address
 < client port >: remote client port (range: 0~65535)

Case 1:

If server socket ID has not accepted any client connection.

```
AT+ATCPSERVES=1
+ATCPSERVERS: 1,1,"110.26.63.152",12000,1,1,0
```

Case 2:

If server socket ID has accept one client connection.

```
AT+ATCPSERVES=1
+ATCPSERVERS: 1,1,"110.26.63.152",12000,1,1,0
+ATCPCLNTO: 1,1,219.84.14.102,2904
```

Note: + ATCPCLNTO means that the server socket id 1 has accepted 1 client connections, and the client connection socket id is 1. Then application can use AIPO, AIPW, AIPR, AIPC, AIPSST AT command sets to control the client connection.

Case 3:

List all server socket ID status

```
AT+ATCPSERVES=a
+ATCPSERVERS: 1,1,"110.26.63.152",12000,1,1,0
+ATCPSERVERS: 2,0
+ATCPSERVERS: 3,0
+ATCPSERVERS: 4,0
```

Case 3:

List all server socket ID status and client socket connecion.

```
AT+ATCPSERVES? (AT+ATCPSERVES=A)
+ATCPSERVERS: 1,1,"110.26.63.152",12000,1,1,0
+ATCPCLNTO: 1,1,219.84.14.102,2904
+ATCPSERVERO: 2,0
```

+ATCPSEVERO: 3,0

+ATCPSEVERO: 4,0

11.4. TCP/IP TCP server AT command Even

EVENT of AT for TCPIP		DESCRIPTION
+ATCPCLNTIN: <server socket id>,<client socket id, client IP address>, <client port>		Accept client connecton event
+ATCPSEVERAI : <server socket id>, <number waiting client connections>		Event that keep client connection request and waiting new release client connection service of the server socket
+ATCPSEVERCI : <server socket id>		Event that server socket has been close.

12. Update Examples: Using TCP/UDP AT Commands and Events

1. Update AT+AIPR/AIPW,

- (1) 查詢接收資料狀況, 可以使用 AT+AIPR=Socket ID
- (2) 查詢傳送資料狀況, 可以使用 AT+AIPW=Socket ID

2. +AIPRIPI Event 新增兩個參數, 以便顯示目前資料接收狀況

3. 新增 AT+AIPSET 分別設定四種連線的參數

4. 新增+AIPWACK Event 來顯示目前未傳送出去或尚未有 ACK 的資料數, 並顯示所

有已經有 ACK 的資料數

13. Examples: Using TCP server AT Commands and Events

1. 建立 TCP server:

Server socket id 有 4 個, 1,2,3,4, 可同時建立 4 個 TCP server 的服務, 現在用 Socket id 建立 port 12000 的 TCP server, 限制最多可有 2 個 Client 連線:

EX:

(1) 連線 GPRS, 115.80.115.135 為配發的 IP ADDRESS

AT+AIPA=1

+AIPA: 1,115.80.115.135,0,0,0

(2) 建立 TCP SERVER 服務, Listen Port 為 12000, 連線數為 2

AT+ATCPSEVERO=1,12000,2

+ATCPSEVERO: 1,1,"115.80.115.135",12000,2,0,0

2. 接受遠端服務

(1)當遠端 Host 連線進入已設定的 TCP server socket 時, GPRS Module 會發出的 Event, 此時可以使用所分配的 Client Socket ID 服務此遠端連線, 如下:

+ATCPCLNTIN: 1,1,219.84.14.102,3424

server socket id 為 1, client socket id 分配也是 1, client IP address 為 219.84.14.102, client port 為 3424

(2)此時可用 A+AIPSST 查詢 Client socket id 配置狀態

AT+AIPSST=1

+AIPSST:1,1,"115.80.115.135",12000,"219.84.14.102",3424,0,0,2,7300,17520,0,0(0,0)

(3)應用端程式可用 AIPO/AIPR/AIPW/AIPC, 操作 client socket id 1 來服務遠端連線

3.查詢 TCP server 狀態

(1) 可用"AT+ATCPSEVERO" 簡單確認 TCP server 狀態, 如下:

AT+ATCPSEVERO=1

+ATCPSEVERO: 1,1,"115.80.115.135",12000,2,1,0

+ATCPCLNTSID: 1

確認為開啟狀態, +ATCPSEVERO: Server Socket 1, Server IP address 為 115.80.115.135, 1 為開啟狀態, 最大連接數為 2, 目前連接數為 1, 等待連接數為 0. 並列出遠端連接 Socket ID, +ATCPCLNTSID: 目前僅有 Client Socket ID 1

(2)可用"AT+ATCPSEVERO" 詳細查詢 TCP server 狀態, 如下:

AT+ATCPSEVERO=1

+ATCPSEVERO: 1,1,"115.80.115.135",12000,2,1,0

+ATCPCLNTID: 1,1,219.84.14.102,3424

以上列出 Server Socket 1 相關資訊+ATCPSEVERO: Server Socket 1, Server IP address 為 115.80.115.135, 1 為開啟狀態, 最大連接數為 2, 目前連接數為 1, 等待連接數為 0. 並依序列出遠端連接 Client Socket id 1 的狀態+ATCPCLNTID: Server socket 為 1, Client Socket 為 1, 1 為開啟狀態, client IP address 為 219.84.14.102, client port 為 3424

4. 關閉 TCP server 服務:

(1) 請先關閉 Client 連線:

AT+AIPC=1

+AIPC: 1

(2) 最後再關閉 TCP server 服務

AT+ATCPSEVERO=1

+ATCPSEVERO: 1