

# **ADH8060/8066 GSM/GPRS Module**

Enhanced AT Command  
Reference Manual

<V1.6>

[ADH Technology Co.,LTD](http://www.adh-tech.com.tw)

Subject to changes in technology, design and availability

[www.adh-tech.com.tw](http://www.adh-tech.com.tw)

## Revision History

No.	Date	Notes
V1.6	2009/10/15	Add GPRS/Socket status commands
V1.5	2009/9/1	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>4</u>
1.1. DEFINE PDP CONTEXT +AIPDCONT .....	4
1.2. QUALITY OF SERVICE PROFILE (REQUESTED) +AIQREQ .....	5
1.3. QUALITY OF SERVICE PROFILE (MINIMUM) +AIQMIN .....	5
1.4. GPRS ATTACH OR DETACH +AIPA .....	6
1.5. OPEN SOCKET AND CONNECT TO INDICATED PORT AND IP +AIPO .....	6
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 +AIPSS .....	10
<u>2. TCP/UDP EVENT INTERFACE</u> .....	<u>12</u>
2.1. RECEIVE TCP DATA +AIPRTCP .....	12
2.2. RECEIVE UDP DATA +AIPRUDP .....	12
<u>3. EXAMPLES: USING TCP/UDP AT COMMANDS AND EVENTS</u> .....	<u>14</u>
3.1. ATTACH TO GPRS NETWORK .....	14
3.2. USING TCP FUNCTION: .....	14
3.3. USING UDP FUNCTION: .....	15
<u>4. CELL INFORMATION AT COMMANDS</u> .....	<u>17</u>
4.1. CHECK SERVING CELL INFORMATION +CSCI .....	17
4.2. CHECK NEIGHBORING CELL INFORMATION +CNCI .....	17
4.3. CHECK DEDICATED MODE INFORMATION +CDMI .....	17
<u>5. SPECIAL FUNCTION AT COMMANDS</u> .....	<u>19</u>
5.1. CHECK FIRMWARE VERSION +SFUN=VER .....	19
5.2. POWER OFF MODULE +SFUN=OFF .....	19
5.3. RESET MODULE +SFUN=RST .....	19

5.4.	CSQ DETECTION +SFUN=CSQ .....	19
5.5.	CREG DETECTION +SFUN=CRG.....	19
5.6.	SIM CART DETECTION +SFUN=SIM.....	20
5.7.	AT COMMAND READY DETECTION +SFUN=RDY.....	20
5.8.	SIM CARD ICCID +SFUN=SID .....	20
5.9.	NORMAL PLMN DIGITS ORDER +SFUN=OPN .....	20
5.10.	ORIGIANL PLMN DIGITS ORDER +SFUN=OPO.....	21
<b>6.</b>	<b>AUTO BAUD RATE .....</b>	<b>22</b>
6.1.	SET AUTO BAUD RATE MODE +IPR=0.....	22
6.2.	HOW TO DETECT BAUD RATE .....	22
<b>7.</b>	<b>ERROR CODE .....</b>	<b>23</b>
7.1.	TCP/IP AT COMMAND RETURN ERROR CODE .....	23
7.2.	QUERY LAST ERROR CODE +AIPESTAT .....	23
7.3.	RETURN ERROR CODE TABLE .....	24
7.4.	TCP/IP AT COMMAND RETURN EVENT.....	25

## 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 attach or detach +AIPA

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

#### Supported parameter values

<mode>: 0,1 (0: detach, 1: attach)

#### Response values

AT+AIPA=1: +AIPA: 1, IP address, seconds for attach

AT+AIPA=0: +AIPA: 0, seconds for detach

AT+AIPA?: +AIPA: 1, IP address, seconds for attach, or +AIPA: 0 not attached

#### Restriction

You must close all socket before detach GPRS.

#### 1.5. 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>	+AIPO: (same as input arguments) 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 timeout duration waiting for server's response when using TCP protocol. (default: 30 seconds)

<dnr>: 0: disable( default value),1: DNR function enable. If parameters #3 is domain name, you need set to 1 and enable DNR function to query actually IP address for that domain name. If parameters #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 send "+++" to end this data mode and back to command.

## Data transfer mode

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

**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.

**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)

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.

### Case 2:

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)

### Case 3:

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>	+AIPW: <Socket ID>, <protocol>,<recv buf len>,<send buf len>,<write bytes> 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.

### Response values



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

+AIPW: <socket id>, <protocol>,<recv buf len>,<send buf len>,<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)

### 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.

### 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: (same as input arguments) 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)

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=1

+AIPSST: Socket ID, 1, local IP address, local port, Remote IP, Remote port, Protocol (0:TCP, 1:UDP)

#### Case 3:

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

AT+AIPSST=1

+AIPSST: Socket ID, 2, local IP address, local port, Remote IP, Remote port, Protocol (0:TCP, 1:UDP)

#### 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) )

.....

+AIPSST: 8, (0..2), ( local IP address, local port, Remote IP, Remote port, Protocol (0:TCP, 1:UDP) )

## 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/GRPS 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 // Response status includes local IP address
"221.120.5.138" that system assigned and
connection time that is now 0 second.
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: // If a local port is not assigned, system will
1,5678,"203.160.10.251",12000,0 assign free local port. Ex: This case the local
port is 5678.

```

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

## 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,"414243444546"

// 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

+AIPC: 1

server and free this socket for next connection.

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 )

```
+AIPO: 1,5679,"203.160.10.251",10000,1
```

If a local port is not assigned, system will assign a free local port. Ex: This case is local port 5679.

```
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
```

## 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,"4142434444546"
```

// 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=1
```

Close 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

Confidential

## 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 Card 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**

&lt;id no&gt;: 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

<br>: 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: <br>

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

### 7.2. Query last error code +AIPESTAT

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

EVENT of AT for TCPIP		DESCRIPTION
<b>+AIPCI: socket id</b>		TCPIP SOCK Remote Closed
<b>+AIPEVNTI: socket id, 3</b>		TCPIP Buffer Free
<b>+AIPEVNTI: socket id, 8</b>		TCPIP SOCK Send Wnd Opened
<b>+AIPEVNTI: socket id, 10</b>		TCPIP SOCK Host Unreach

Confidential