SIM7500_SIM7600 Series_ AT Command Manual

LTE Module

SIMCom Wireless Solutions Limited
Building B, SIM Technology Building, No.633, Jinzhong Road
Changning District, Shanghai P.R. China
Tel: 86-21-31575100
support@simcom.com
www.simcom.com

Document Title:	SIM7500_SIM7600 Series_AT Command Manual
Version:	2.00
Date:	2020.8.6
Status:	Released

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

SIMCom Wireless Solutions Limited
Building B, SIM Technology Building, No.633 Jinzhong Road, Changning District, Shanghai P.R. China
Tel: +86 21 31575100
Email: simcom@simcom.com

For more information, please visit:
https://www.simcom.com/download/list-863-en.html

For technical support, or to report documentation errors, please visit:
https://www.simcom.com/ask/ or email to: support@simcom.com

Version History

Version	Date	Chapter	What is new
V2.00	2020.8.6		New version

Contents

THIS DOCUMENT IS A REFERENCE GUIDE TO ALL THE AT COMMANDS.

1. Introduction

1.1 Scope of the document

This document presents the AT Command Set for SIMCom SIM7500 and SIM7600 series.

1.2 Related documents

You can visit the SIMCom Website using the following link:
http://www.simcom.com

1.3 Conventions and abbreviations

In this document, the GSM engines are referred to as following term:
ME (Mobile Equipment);
MS (Mobile Station);
TA (Terminal Adapter);
DCE (Data Communication Equipment) or facsimile DCE (FAX modem, FAX board);

In application, controlling device controls the GSM engine by sending AT Command via its serial interface. The controlling device at the other end of the serial line is referred to as following term:
TE (Terminal Equipment);
DTE (Data Terminal Equipment) or plainly "the application" which is running on an embedded system;

1.4 AT Command syntax

The "AT" or "at" or "aT" or "At" prefix must be set at the beginning of each Command line. To terminate a Command line enter **<CR>**.

Commands are usually followed by a response that includes. **"<CR><LF><response><CR><LF>"** Throughout this document, only the responses are presented,<**CR><LF>** are omitted intentionally.

The AT Command set implemented by SIM7500&SIM7600 Series is a combination of 3GPP TS 27.005, 3GPP TS 27.007 and ITU-T recommendation V.25ter and the AT commands developed by SIMCom.

NOTE

Only enter AT Command through serial port after SIM7500&SIM7600 Series is powered on and Unsolicited Result Code "RDY" is received from serial port. If auto-bauding is enabled, the Unsolicited Result Codes "RDY" and so on are not indicated when you start up the ME, and the "AT" prefix, or "at" prefix must be set at the beginning of each command line.

All these AT commands can be split into three categories syntactically: "**basic**", "**S parameter**", and "**extended**". These are as follows:

1.4.1 Basic syntax

These AT commands have the format of "**AT<*x*><*n*>**", or "**AT&<x><n>**", where "**<*x*>**"is the Command, and "**<*n*>**"is/are the argument(s) for that Command. An example of this is "**ATE<*n*>**", which tells the DCE whether received characters should be echoed back to the DTE according to the value of "**<*n*>**". "**<*n*>**" is optional and a default will be used if missing.

1.4.2 S Parameter syntax

These AT commands have the format of "**ATS<*n*>=<*m*>**", where "**<*n*>**" is the index of the **S** register to set, and *"***<*m*>***"* is the value to assign to it. "**<*m*>**" is optional; if it is missing, then a default value is assigned.

1.4.3 Extended Syntax

These commands can operate in several modes, as in the following table:

Table 1: Types of AT commands and responses	
Test Command	The mobile equipment returns the list of parameters and value ranges set with the corresponding Write Command or by internal processes.

AT+<x>=?	
Read Command	This command returns the currently set value of the parameter or parameters.
AT+<x>?	
Write Command	This command sets the user-definable parameter values.
AT+<x>=<…>	
Execution Command	The execution command reads non-variable parameters affected by internal processes in the GSM engine.
AT+<x>	

1.4.4 Combining AT commands on the same Command line

You can enter several AT commands on the same line. In this case, you do not need to type the "**AT**" or "**at**" prefix before every command. Instead, you only need type "**AT**" or "**at**" the beginning of the command line. Please note to use a semicolon as the command delimiter after an extended command; in basic syntax or S parameter syntax, the semicolon need not enter, for example:
ATE1Q0S0=1S3=13V1X4;+IFC=0,0;+IPR=115200.

The Command line buffer can accept a maximum of 559 characters (counted from the first command without "AT" or "at" prefix) or 39 AT commands. If the characters entered exceeded this number then none of the Command will executed and TA will return "**ERROR**".

1.4.5 Entering successive AT commands on separate lines

When you need to enter a series of AT commands on separate lines, please Note that you need to wait the final response (for example OK, CME error, CMS error) of last AT Command you entered before you enter the next AT Command.

1.5 Supported character sets

The SIM7500&SIM7600 Series AT Command interface defaults to the **IRA** character set. The SIM7500&SIM7600 Series supports the following character sets:
GSM format
UCS2
IRA
The character set can be set and interrogated using the "AT+CSCS" Command (3GPP TS 27.007). The character set is defined in GSM specification 3GPP TS 27.005.

The character set affects transmission and reception of SMS and SMS Cell Broadcast messages, the entry and display of phone book entries text field and SIM Application Toolkit alpha strings.

1.6 Flow control

Flow control is very important for correct communication between the GSM engine and DTE. For in the case such as a data or fax call, the sending device is transferring data faster than the receiving side is ready to accept. When the receiving buffer reaches its capacity, the receiving device should be capable to cause the sending device to pause until it catches up.

There are basically two approaches to achieve data flow control: software flow control and hardware flow control. SIM7500&SIM7600 Series support both two kinds of flow control.
In Multiplex mode, it is recommended to use the hardware flow control.

1.6.1 Software flow control (XON/XOFF flow control)

Software flow control sends different characters to stop (XOFF, decimal 19) and resume (XON, decimal 17) data flow. It is quite useful in some applications that only use three wires on the serial interface.

The default flow control approach of SIM7500&SIM7600 Series is hardware flow control (RTS/CTS flow control), to enable software flow control in the DTE interface and within GSM engine, type the following AT Command:
AT+IFC=1,1

Ensure that any communications software package (e.g. Hyper terminal) uses software flow control.

NOTE

Software Flow control should not be used for data calls where binary data will be transmitted or received (e.g. TCP/IP) as the DTE interface may interpret binary data as flow control characters.

1.6.2 Hardware flow control (RTS/CTS flow control)

Hardware flow control achieves the data flow control by controlling the RTS/CTS line. When the data transfer should be suspended, the CTS line is set inactive until the transfer from the receiving buffer has completed. When the receiving buffer is ok to receive more data, CTS goes active once again.

To achieve hardware flow control, ensure that the RTS/CTS lines are present on your application platform.

1.7 Definitions

1.7.1 Parameter Saving Mode

For the purposes of the present document, the following syntactical definitions apply:

- **NO_SAVE**: The parameter of the current AT command will be lost if module is rebooted or current AT command doesn't have parameter.
- **AUTO_SAVE**: The parameter of the current AT command will be kept in NVRAM automatically and take in effect immediately, and it won't be lost if module is rebooted.
- **AUTO_SAVE_REBOOT**: The parameter of the current AT command will be kept in NVRAM automatically and take in effect after reboot, and it won't be lost if module is rebooted.

1.7.2 Max Response Time

Max response time is estimated maximum time to get response, the unit is seconds.

2. AT Commands According to V.25TER

2.1 Overview of AT Commands According to V.25TER

Command	Description
A/	Re-issues the last command given
ATD	Mobile originated call to dial a number
ATE	Set command echo mode
ATH	Disconnect existing connection
ATI	Display product identification information
ATL	Set monitor speaker loudness
ATM	Set monitor speaker mode
+++	Switch from data mode or ppp online mode to command mode
ATO	Switch from command mode to data mode
ATQ	Set result code presentation mode
ATS0	Set number of rings before automatically answering the call
ATS3	Set command line termination character
ATS4	Set response formatting character
ATS5	Set command line editing character
ATS6	Pause before blind dialling
ATS7	Set number of seconds to wait for connection completion
ATS8	Set number of seconds to wait for comma dial modifier encountered in dial string of D command
ATS10	Set disconnect delay after indicating the absence of data carrier
ATV	TA response format
ATX	Set connect result code format and monitor call progress
ATZ	Reset default configuration
AT&C	Set DCD function mode
AT&D	Set DTR function mode
AT&E	Set CONNECT Result Code Format About Speed
AT+GCAP	Request complete TA capabilities list
AT+GMI	Request manufacturer identification
AT+GMM	Request TA model identification
AT+GMR	Request TA revision identification of software release

AT+GOI	Request global object identification
AT+GSN	Request TA serial number identification (IMEI)
AT+ICF	Set TE-TA control character framing
AT+IPR	Set TE-TA fixed local rate

2.2 Detailed Description of AT Commands According to V.25TER

2.2.1 A/ Re-issues the Last Command Given

A/ Re-issues the Last Command Given	
Execution Command **A/**	Response Re-issues the previous Command
Parameter Saving Mode	NO_SAVE
Maximum Response Time	120000ms
Reference	

Example

```
A/
+GCAP:+CGSM,+FCLASS,+DS
OK
```

2.2.2 ATD Mobile Originated Call to Dial A Number

This command can be used to set up outgoing data calls. It also serves to control supplementary services.

ATD Mobile Originated Call to Dial A Number	
Execution Command **ATD<n>[<mgsm]**	Response If error is related to ME functionality **+CME ERROR: <err>** If no dial tone and (parameter setting ATX2 or ATX4) **NO DIALTONE** If busy and (parameter setting ATX3 or ATX4) **BUSY**

	If a connection cannot be established **NO CARRIER** If the remote station does not answer **NO ANSWER** If connection successful and non-voice call. **CONNECT<text>** TA switches to data mode. Note: <text> output only if ATX<value> parameter setting with the <value> >0 When TA returns to command mode after call release **OK**
Parameter Saving Mode	NO_SAVE
Maximum Response Time	Timeout set with ATS7 (data call)
Reference	

Defined Values

<n>	String of dialing digits and optionally V.25ter modifiers dialing digits: 0-9,*, #,+,A,B,C Following V.25ter modifiers are ignored: ,(comma),T,P,!,W,@
Emergency call:	
<n>	Standardized emergency number 112 (no SIM needed)
<mgsm>	String of GSM modifiers: I Actives CLIR (Disables presentation of own number to called party) i Deactivates CLIR (Enable presentation of own number to called party) G Activates Closed User Group invocation for this call only g Deactivates Closed User Group invocation for this call only

Example

```
ATD10086;
OK
VOICE CALL:BEGIN
```

NOTE

- This command may be aborted generally by receiving an ATH Command or a character during execution. The aborting is not possible during some states of connection establishment such as

handshaking.

2.2.3 ATD><mem><n> Originate call from specified memory

This command is used to originate a call using specified memory and index number.

ATD><mem><n> Originate call from specified memory	
Execution Command **ATD<mem><n>[;]**	Response a)If originate a voice call successfaully: **OK** **VOICE CALL:BEGIN** b) If Originate a data call successfully: **CONNECT [<text>]** c) Originate a call unsuccessfully during command execution: **ERROR** d)Originate a call unsuccessfully for failed connection recovery: **NO CARRIER** e)Originate a call unsuccessfully for error related to the MT: **+CME ERROR: <err>**
Maximum Response Time	Timeout set with ATS7 (data call)
Reference V.25ter	

Defined Values

<mem>	Phonebook storage: (For detailed description of storages see AT+CPBS) "DC" ME dialed calls list "MC" ME missed (unanswered received) calls list "RC" ME received calls list "SM" SIM phonebook "ME" UE phonebook "FD" SIM fixed dialing phonebook "ON" MSISDN list "LD" Last number dialed phonebook "EN" Emergency numbers
<n>	Integer type memory location in the range of locations available in the

	selected memory, i.e. the index returned by AT+CPBR.
<;>	The termination character ";" is mandatory to set up voice calls. It must not be used for data and fax calls.
<text>	CONNECT result code string; the string formats please refer ATX/AT\V/AT&E command.
<err>	Service failure result code string; the string formats please refer +CME ERROR result code and AT+CMEE command.

Example

```
ATD>SM3                          //Specify the <mem>.
OK
VOICE CALL: BEGIN
```

2.2.4 ATD><n> Originate call from active memory(1)

This command is used to originate a call to specified number. Telecom does not support this command.

ATD><n> Originate call from active memory	
Execution Command **ATD><n>[;]**	Response a)If originate a voice call successfaully: **OK** **VOICE CALL:BEGIN** b) If Originate a data call successfully: **CONNECT [<text>]** c) Originate a call unsuccessfully during command execution: **ERROR** d)Originate a call unsuccessfully for failed connection recovery: **NO CARRIER** e)Originate a call unsuccessfully for error related to the MT: **+CME ERROR: <err>**
Maximum Response Time	Timeout set with ATS7 (data call)
Reference V.25ter	

Defined Values

<n>	Integer type memory location in the range of locations available in the

	selected memory, i.e. the index returned by AT+CPBR.
<;>	The termination character ";" is mandatory to set up voice calls. It must not be used for data and fax calls.
<text>	CONNECT result code string; the string formats please refer ATX/AT\V/AT&E command.
<err>	Service failure result code string; the string formats please refer +CME ERROR result code and AT+CMEE command.

Example

```
ATD>2;
OK
VOICE CALL: BEGIN
```

2.2.5 ATD><str> Originate call from active memory(2)

This command is used to originate a call to specified number. Telecom does not support this command.

ATD><n> Originate call from active memory	
Execution Command **ATD><str>[;]**	Response a)If originate a voice call successfaully: **OK** **VOICE CALL:BEGIN** b) If Originate a data call successfully: **CONNECT [<text>]** c) Originate a call unsuccessfully during command execution: **ERROR** d)Originate a call unsuccessfully for failed connection recovery: **NO CARRIER** e)Originate a call unsuccessfully for error related to the MT: **+CME ERROR: <err>**
Maximum Response Time	Timeout set with ATS7 (data call)
Reference V.25ter	

Defined Values

<str>	String type value, which should equal to an alphanumeric field in at

	least one phone book entry in the searched memories. <str> formatted as current TE character set specified by AT+CSCS.<str> must be double quoted.
<;>	The termination character ";" is mandatory to set up voice calls. It must not be used for data and fax calls.
<text>	CONNECT result code string; the string formats please refer ATX/AT\V/AT&E command.
<err>	Service failure result code string; the string formats please refer +CME ERROR result code and AT+CMEE command.

Example

```
ATD>"kobe";
OK
VOICE CALL: BEGIN
```

2.2.6 ATA Call answer

This command is used to make remote station to go off-hook, e.g. answer an incoming call. If there is no an incoming call and entering this command to TA, it will be return "**NO CARRIER**" to TA.

ATA Call answer	
Execution Command **ATA**	Response a)If originate a voice call successfaully: **OK** **VOICE CALL:BEGIN** b)For data call, and TA switches to data mode: **CONNECT** c)No connection or no incoming call: **NO CARRIER**
Reference V.25ter	

Example

```
ATA
VOICE CALL: BEGIN
OK
```

2.2.7 ATH Disconnect existing call

This command is used to disconnect existing call. Before using **ATH** command to hang up a voice call, it must set **AT+CVHU=0**. Otherwise, ATH command will be ignored and "*OK*" response is given only.
This command is also used to disconnect PS data call, and in this case it doesn't depend on the value of **AT+CVHU**.

ATH Disconnect existing call	
Execution Command **ATH**	Response a) If AT+CVHU=0: **VOICE CALL:END:<time>** **OK**
Reference V.25ter	

Defined Values

<time>	Voice call connection time: Format – HHMMSS (HH: hour, MM: minute, SS: second)

Example

```
AT+CVHU=0
OK
ATH
VOICE CALL:END:000017
OK
```

2.2.8 ATS0 Automatic answer incoming call

The S-parameter command controls the automatic answering feature of the Module. If set to 000, automatic answering is disabled, otherwise it causes the Module to answer when the incoming call indication (RING) has occurred the number of times indicated by the specified value; and the setting will not be stored upon power-off, i.e. the default value will be restored after restart.

ATS0 Automatic answer incoming call	
Read Command **ATS0?**	Response a)If succes: **<n>** **OK**

	b) If failed **ERROR**
Write command **ATS0=<n>**	Response a)If succes: **OK** b)If failed **ERROR**
Reference V.25ter	

Defined Values

<n>	000 Automatic answering mode is disable. (default value when power-on) 001–255 Enable automatic answering on the ring number specified.

NOTE

1.The S-parameter command is effective on voice call and data call.
2.If **<n>** is set too high, the remote party may hang up before the call can be answered automatically.

Example

```
ATS0?
000
OK
ATS0=003
OK
```

2.2.9 +++ Switch from data mode to command mode

This command is only available during a connecting PS data call. The **+++** character sequence causes the TA to cancel the data flow over the AT interface and switch to Command Mode. This allows to enter AT commands while maintaining the data connection to the remote device.

+++ Switch from data mode to command mode	
Execution Command **+++**	Response **OK**
Reference V.25ter	

NOTE

To prevent the +++ escape sequence from being misinterpreted as data, it must be preceded and followed by a pause of at least 1000 milliseconds, and the interval between two '+' character can't exceed 900 milliseconds.

2.2.10 ATO Switch from command mode to data mode

ATO is the corresponding command to the **+++** escape sequence. When there is a PS data call connected and the TA is in Command Mode, **ATO** causes the TA to resume the data and takes back to Data Mode.

ATO Switch from command mode to data mode	
Execution Command **ATO**	Response a) TA/DCE switches to Data Mode from Command Mode: **CONNECT [<baud rate>]** b) If connection is not successfully resumed: **NO CARRIER** **ERROR**
Reference V.25ter	

Example

```
ATO
CONNECT 115200
```

2.2.11 ATI Display product identification information

This command is used to request the product information, which consists of manufacturer identification, model identification, revision identification, International Mobile station Equipment Identity (IMEI) and overall capabilities of the product.

ATI Display product identification information	
Execution Command **ATI**	Response **Manufacturer: <manufacturer>**

	Model: <model> **Revision: <revision>** **IMEI: [<sn>]** **+GCAP: list of <name>s** **OK**
Reference V.25ter	

Defined Values

<manufacturer>	The identification of manufacturer.
<model>	The identification of model.
<revision>	The revision identification of firmware.
<sn>	Serial number identification, which consists of a single line containing IMEI (International Mobile station Equipment Identity) number.
<name>	List of additional capabilities: +CGSM GSM function is supported +FCLASS FAX function is supported +DS Data compression is supported +ES Synchronous data mode is supported. +CIS707-A CDMA data service command set +CIS-856 EVDO data service command set +MS Mobile Specific command set

Example

```
ATI
Manufacturer:                SIMCOM
INCORPORATED
Model: SIMCOM_SIM7600C
Revision: SIM7600C _V1.0
IMEI: 351602000330570
+GCAP: +CGSM,+FCLASS,+DS

OK
```

2.2.12 AT+IPR Set local baud rate temporarily

This command sets the baud rate of module's serial interface temporarily, after reboot the baud rate is set to value of IPREX.

AT+IPR Set local baud rate tmporarily	
Test Command **AT+IPR=?**	Response **+IPR: (list of supported<speed>s)** **OK**
Read Command **AT+IPR?**	Response **+IPR: <speed>** **OK**
Write Command **AT+IPR=<speed>**	Response **OK** or **ERROR**
Execution Command **AT+IPR=<speed>**	Set the value to boot value: **OK**

Defined Values

<speed>	Baud rate per second: 0, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800,921600, 3000000,3200000,3686400

2.2.13 AT+ICF Set control character framing

This command sets character framing which contains data bit, stop bit and parity bit.

AT+IPR Set local baud rate tmporarily	
Test Command **AT+ICF=?**	Response **+ICF: (list of supported<format>s), (list of supported<parity>s)** **OK**
Read Command **AT+ICF?**	Response **+ICF: <format>,<parity>** **OK**
Write Command **AT+ICF=<format>[,<parity>]**	Response **OK** or **ERROR**
Execution Command **AT+ICF**	Set default value: **OK**
Reference V.25ter	

Defined Values

<format>	1 – data bit 8, stop bit 2 2 – data bit 8, parity bit 1,stop bit 1 3 – data bit 8, stop bit 1 4 – data bit 7, stop bit 2 5 – data bit 7, parity bit 1,stop bit 1 6 – data bit 7, stop bit 1
<parity>	0 – Odd 1 – Even 2 – Space 3 – none

Example

```
AT+ICF?
+ICF: 3,3
OK
AT+ICF=?
+ICF: (1-6),(0-3)
OK
AT+ICF=3,3
OK
```

2.2.14 AT+IFC Set local data flow control

The command sets the flow control mode of the module.

AT+IFC Set local data flow control	
Test Command **AT+IFC=?**	Response **+IFC: (list of supported<DCE>s), (list of supported<DTE>s)** **OK** or **ERROR**
Read Command **AT+IFC?**	Response **+IFC: <DCE>,<DTE>** **OK** or **ERROR**
Write Command **AT+IFC=<DCE>[,<DTE>]**	Response **OK** or **ERROR**
Execution Command	Set default value:

AT+IFC	OK
Reference V.25ter	

Defined Values

<DCE>	0 – none (default) 2 – RTS hardware flow control
<DTE>	0 – none (default) 2 – CTS hardware flow control

Example

```
AT+IFC?
+ICF: 0,0
OK
AT+IFC=?
+ICF: (0,2),(0,2)
OK
AT+ICF=2,2
OK
```

2.2.15 AT&C Set DCD function mode

This command determines how the state of DCD PIN relates to the detection of received line signal from the distant end.

AT&C Set DCD function mode	
Execution Command **AT&C[<value>]**	Response **OK** or **ERROR**
Reference V.25ter	

Defined Values

<value>	0 DCD line shall always be on. 1 DCD line shall be on only when data carrier signal is present. 2 Setting winks(briefly transitions off,then back on)the DCD line when data calls end.

Example

```
AT&C1
OK
```

2.2.16 ATE Enable command echo

This command sets whether or not the TA echoes characters.

ATE Enable command echo	
Execution Command **ATE[<value>]**	Response **OK** or **ERROR**
Reference V.25ter	

Defined Values

<value>	0 – Echo mode off 1 – Echo mode on

Example

```
ATE1
OK
```

2.2.17 AT&V Display current configuration

This command returns some of the base configuration parameters settings.

AT&V Display current configuration	
Execution Command **AT&V**	Response **<text>** **OK** or **ERROR**
Reference V.25ter	

Defined Values

<text>	All relative configuration information.

Example

```
AT&V
&C: 0; &D: 2; &F: 0; E: 1; L: 0; M: 0; Q:
0; V: 1; X: 0; Z: 0; S0: 0;
S3: 13; S4: 10; S5: 8; S6: 2; S7: 50; S8:
2; S9: 6; S10: 14; S11: 95;
+FCLASS: 0; +ICF: 3,3; +IFC: 2,2;
+IPR: 115200; +DR: 0; +DS: 0,0,2048,6;
+WS46: 12; +CBST: 0,0,1;
……
OK
```

2.2.18 AT&D Set DTR function mode

This command determines how the TA responds when DTR PIN is changed from the ON to the OFF condition during data mode.

AT&D Set DTR function mode	
Execution Command **AT&V[<value>]**	Response **OK** or **ERROR**
Reference V.25ter	

Defined Values

<value>	0 TA ignores status on DTR. 1 ON->OFF on DTR: Change to Command mode with remaining the connected call 2 ON->OFF on DTR: Disconnect call, change to Command mode.During state DTR = OFF is auto-answer off.

Example

```
AT&D1
```

OK

2.2.19 AT&S Set DSR function mode

The command determines how the state of DSR pin works.

AT&D Set DSR function mode	
Execution Command **AT&S[<value>]**	Response **OK** or **ERROR**
Reference V.25ter	

Defined Values

<value>	0 DSR line shall always be on. 1 DSR line shall be on only when DTE and DCE are connected.

Example

```
AT&S0
OK
```

2.2.20 ATV Set result code format mode

This parameter setting determines the contents of the header and trailer transmitted with result codes and information responses.

ATV Set result code format mode	
Write Command **ATV[<value>]**	Response If <value> =0 **0** If <value> =1 **OK**
Reference V.25ter	

Defined Values

<value>	0 Information response: <text><CR><LF> Short result code format: <numeric code><CR> 1 Information response: <CR><LF><text><CR><LF> Long result code format: <CR><LF><verbose code><CR><LF>

Example

```
ATV1
OK
```

2.2.21 AT&F Set all current parameters to manufacturer defaults

This command is used to set all current parameters to the manufacturer defined profile.

AT&F Set all current parameters to manufacturer defaults	
Write Command **AT&F[<value>]**	Response **OK** or **ERROR**
Reference V.25ter	

Defined Values

<value>	0 — Set some temporary TA parameters to manufacturer defaults. The setting after power on or reset is same as value 0.

Example

```
AT&F
OK
```

2.2.22 ATQ Set Result Code Presentation Mode

Specify whether the TA transmits any result code to the TE or not. Text information transmitted in response is not affected by this setting.

ATQ Set Result Code Presentation Mode	
Write Command **ATQ<n>**	Response **If <n>=0:**

	OK **If <n>=1:** **No Responses**
ATQ	Set default value: 0 **OK** **No Responses**
Reference V.25ter	

Defined Values

<n>	0 – DCE transmits result code 1 – DCE not transmits result code

Example

```
ATQ0
OK
```

2.2.23 ATX Set CONNECT Result Code Format

This parameter setting determines whether the TA transmits unsolicited result codes or not. The unsolicited result codes are
<CONNECT><SPEED><COMMUNICATION PROTOCOL>[<TEXT>]

ATX Set CONNECT Result Code Format	
Write Command **ATX<n>**	Response **OK** or **ERROR**
Execution Command **ATX**	Set default value: 1 **OK** or **ERROR**
Reference V.25ter	

Defined Values

<value>	0 – CONNECT result code returned

	1,2,3,4 – May be transmits extern result codes according to AT&E and AT\V settings. Refer to AT&E.

Example

```
ATX1
OK
```

2.2.24 AT\V Set CONNECT Result Code Format About Protocol

This parameter setting determines whether report the communication protocol. If PS call, it also determines wether report APN, uplink rate, downlink rate.

AT\V Set CONNECT Result Code Format About Protocol	
Write Command **AT\V<value>**	Response **OK** or **ERROR**
Execution Command **AT\V**	Set default value: 0 **OK** or **ERROR**
Reference V.25ter	

Defined Values

<value>	0 – Don't report 1 – Report communication protocol. And report APN, uplink rate, downlink rate if PS call. Refer to AT&E. The maybe communication protocol report include "NONE","PPPoverUD","AV32K","AV64K","PACKET". And APN in string format while uplink rate and downlink rate in integer format with kb unit.

Example

```
AT\V0
OK
```

2.2.25 AT&E Set CONNECT Result Code Format About Speed

This parameter setting determines to report Serial connection rate or Wireless connection speed. It is valid only ATX above 0.

AT&E Set CONNECT Result Code Format About Speed	
Write Command **AT&E<value>**	Response **OK** or **ERROR**
Execution Command **AT&E**	Set default value: 1 **OK** or **ERROR**
Reference V.25ter	

Defined Values

<value>	0 – Wireless connection speed in integer format. 1 – Serial connection rate in integer format. Such as: “115200”

Example

```
AT&E0
OK
```

2.2.26 AT&W Save the user setting to ME

This command will save the user settings to ME which set by ATE, ATQ, ATV, ATX, AT&C AT&D, AT&S, AT\V, AT+IFC and ATS0.

AT&W Save the user setting to ME	
Write Command **AT&W<value>**	Response **OK** or **ERROR**
Execution Command **AT&W**	Set default value: 0 **OK** or **ERROR**

Reference V.25ter	

Defined Values

<value>	0 – Save

Example

```
AT&W0
OK
```

2.2.27 ATZ Restore the user setting from ME

This command will restore the user setting from ME which set by ATE, ATQ, ATV, ATX, AT&C AT&D, AT&S, AT\Q, AT\V, and ATS0.

ATZ Restore the user setting from ME	
Write Command **ATZ<value>**	Response **OK** or **ERROR**
Execution Command **ATZ**	Set default value: 0 **OK** or **ERROR**
Reference V.25ter	

Defined Values

<value>	0 – Restore

Example

```
ATZ0
OK
```

2.2.28 AT+CGMI Request manufacturer identification

This command is used to request the manufacturer identification text, which is intended to permit the user of the Module to identify the manufacturer.

AT+CGMI Request manufacturer identification	
Test Command **AT+CGMI=?**	Response **OK**
Execution Command **AT+CGMI**	Response **<manufacturer>** **OK** or **ERROR**
Reference V.25ter	

Defined Values

<manufacturer>	The identification of manufacturer.

Example

```
AT+CGMI
SIMCOM INCORPORATED
OK
```

2.2.29 AT+CGMM Request model identification

This command is used to requests model identification text, which is intended to permit the user of the Module to identify the specific model.

AT+CGMM Request model identification	
Test Command **AT+CGMM=?**	Response **OK**
Execution Command **AT+CGMM**	Response **<model>** **OK** or **ERROR**
Reference V.25ter	

Defined Values

<model>	The identification of model.

Example

```
AT+CGMM
SIMCOM_SIM7600C
OK
```

2.2.30 AT+CGMR Request revision identification

This command is used to request product firmware revision identification text, which is intended to permit the user of the Module to identify the version.

AT+CGMR Request revision identification	
Test Command **AT+CGMR=?**	Response **OK**
Execution Command **AT+CGMR**	Response **+CGMR: <revision>** **OK** or **ERROR**
Reference V.25ter	

Defined Values

<revision>	The revision identification of firmware.

Example

```
AT+CGMR
+CGMR: LE11B01SIM7600C
OK
```

2.2.31 AT+CGSN Request product serial number identification

This command requests product serial number identification text, which is intended to permit the user of the Module to identify the individual ME to which it is connected to.

AT+CGSN Request product serial number identification	
Test Command **AT+CGSN=?**	Response **OK**
Execution Command **AT+CGSN**	Response **<sn>** **OK** or **+CME ERROR: memory failure**
Reference V.25ter	

Defined Values

<sn>	Serial number identification, which consists of a single line containing the IMEI (International Mobile station Equipment Identity) number of the MT. If in CDMA/EVDO mode ,it will show ESN(Electronic Serial Number)

Example

```
AT+CGSN
351602000330570
OK
```

2.2.32 AT+CSCS Select TE character set

Write command informs TA which character set <chest> is used by the TE. TA is then able to convert character strings correctly between TE and MT character sets.
Read command shows current setting and test command displays conversion schemes implemented in the TA.

AT+CSCS Select TE character set	
Test Command **AT+CSCS=?**	Response **+CSCS: (list of supported <chset>s)** **OK**
Read Command **AT+CSCS?**	Response **+CSCS: <chset>** **OK**
Write Command **AT+CSCS=<chset>**	Response **OK** **ERROR**

Execution Command **AT+CSCS**	Set subparameters as default value: **OK**
Reference V.25ter	

Defined Values

<chset>	Character set, the definition as following: "IRA" International reference alphabet. "GSM" GSM default alphabet; this setting causes easily software flow control (XON /XOFF) problems. "UCS2" 16-bit universal multiple-octet coded character set; UCS2 character strings are converted to hexadecimal numbers from 0000 to FFFF.

Example

```
AT+CSCS="IRA"
OK
```

2.2.33 AT+CIMI Request international mobile subscriber identity

Execution command causes the TA to return <IMSI>, which is intended to permit the TE to identify the individual SIM card which is attached to MT.
NOTE: If USIM card contains two apps, like China Telecom 4G card, one RUIM/CSIM app, and another USIM app; so there are two IMSI in it; AT+CIMI will return the RUIM/CSIM IMSI; AT+CIMIM will return the USIM IMSI.

AT+CIMI Request international mobile subscriber identity	
Test Command **AT+CIMI=?**	Response **OK**
Execution Command **AT+CIMI**	Response **<IMSI>** **OK** or **+CME ERROR: memory failure**
Reference V.25ter	

Defined Values

<IMSI>	International Mobile Subscriber Identity (string, without double

	quotes).

Example

```
AT+CIMI
460010222028133
OK
```

2.2.34 AT+CIMIM Request another international mobile subscriber identity

Execution command causes the TA to return **<IMSI>**, which is intended to permit the TE to identify the individual SIM card which is attached to MT.
NOTE: If USIM card contains two apps, like China Telecom 4G card, one RUIM/CSIM app, and another USIM app; so there are two IMSI in it; AT+CIMIM will return the USIM IMSI; AT+CIMI will return the RUIM/CSIM IMSI.

AT+CIMIM Request another international mobile subscriber identity	
Test Command **AT+CIMIM=?**	Response **OK**
Execution Command **AT+CIMIM**	Response **<IMSI>** **OK** or **+CME ERROR: memory failure**
Reference V.25ter	

Defined Values

<IMSI>	International Mobile Subscriber Identity (string, without double quotes).

Example

```
AT+CIMIM
460010222028133
OK
```

2.2.35 AT+GCAP Request overall capabilities

Execution command causes the TA reports a list of additional capabilities.

AT+GCAP Request overall capabilities	
Test Command **AT+GCAP=?**	Response **OK**
Execution Command **AT+GCAP**	Response **+GCAP: (list of <name>s)** **OK**
Reference V.25ter	

Defined Values

<name>	List of additional capabilities. +CGSM GSM function is supported +FCLASS FAX function is supported +DS Data compression is supported +ES Synchronous data mode is supported. +CIS707-A CDMA data service command set +CIS-856 EVDO data service command set +MS Mobile Specific command set

Example

```
AT+GCAP
+GCAP:+CGSM,+FCLASS,+DS
OK
```

3. AT Commands for Status Control

3.1 Overview of AT Commands for Status Control

Command	Description
AT+CFUN	Set phone functionality
AT+CPIN	Enter PIN
AT+CICCID	Read ICCID from SIM card
AT+CSIM	Generic SIM access
AT+CRSM	Restricted SIM access
AT+SPIC	Times remain to input SIM PIN/PUK
AT+CSPN	Get service provider name from SIM
AT+CSQ	Query signal quality
AT+AUTOCSQ	Set CSQ report
AT+CSQDELTA	Set RSSI delta change threshold
AT+CATR	Configure URC destination interface
AT+CPOF	Power down the module
AT+CRESET	Reset the module
AT+CACM	Accumulated call meter
AT+CAMM	Accumulated call meter maximum
AT+CPUC	Price per unit and currency table
AT+CCLK	Real time clock management
AT+CMEE	Report mobile equipment error
AT+CPAS	Phone activity status
AT+SIMEI	Set IMEI for the module
AT+SMEID	RequestMobile Equipment Identifier
AT+CSVM	Voice Mail Subscriber number

3.2 Detailed Description of AT Commands for Status Control

3.2.1 AT+CFUN Set phone functionality

Description

This command is used to select the level of functionality <fun> in the ME. Level "full functionality" is where the highest level of power is drawn. "Minimum functionality" is where minimum power is drawn. Level of functionality between these may also be specified by manufacturers. When supported by manufacturers, ME resetting with <rst> parameter may be utilized.

NOTE: AT+CFUN=6 must be used after setting AT+CFUN=7. If module in offline mode, must execute AT+CFUN=6 or **restart** module to online mode.

AT+CFUN Set phone functionality	
Test Command **AT+CFUN=?**	Response **+CFUN: (list of supported <fun>s),(list of supported <rst>s)** **OK** or **ERROR** or **+CME ERROR: <err>**
Read Command **AT+CFUN?**	Response **+CFUN: <fun>** **OK** or **ERROR** or **+CME ERROR: <err>**
Write Command **AT+CFUN=<fun>[,<rst>]**	Response **OK** or **ERROR** or **+CME ERROR: <err>**

Defined values

<fun>	0 – minimum functionality 1 – full functionality, online mode 4 – disable phone both transmit and receive RF circuits 5 – Factory Test Mode

	6 – Reset 7 – Offline Mode
<rst>	0 – do not reset the ME before setting it to <fun> power level 1 – reset the ME before setting it to <fun> power level. This value only takes effect when <fun> equals 1.

Examples

```
AT+CFUN?
+CFUN: 1

OK
AT+CFUN=0
OK
```

3.2.2 AT+CPIN Enter PIN

Description

This command is used to send the ME a password which is necessary before it can be operated (SIM PIN, SIM PUK, PH-SIM PIN, etc.). If the PIN is to be entered twice, the TA shall automatically repeat the PIN. If no PIN request is pending, no action is taken towards MT and an error message, **+CME ERROR**, is returned to TE.

If the PIN required is SIM PUK or SIM PUK2, the second pin is required. This second pin, <newpin>, is used to replace the old pin in the SIM.

AT+CPIN Enter PIN	
Test Command **AT+CPIN=?**	Response **OK**
Read Command **AT+CPIN?**	Response **+CPIN: <code>** **OK** or **ERROR** or **+CME ERROR: <err>**

Write Command	Response
AT+CPIN=<pin>[,<newpin>]	**OK** or **ERROR** or **+CME ERROR: <err>**

Defined values

<pin>	String type values.
<newpin>	String type values.
<code>	Values reserved by the present document: READY – ME is not pending for any password SIM PIN – ME is waiting SIM PIN to be given SIM PUK – ME is waiting SIM PUK to be given PH-SIM PIN – ME is waiting phone-to-SIM card password to be given SIM PIN2 – ME is waiting SIM PIN2 to be given SIM PUK2 – ME is waiting SIM PUK2 to be given PH-NET PIN – ME is waiting network personalization password to be given

Examples

```
AT+CPIN?
+CPIN: SIM PUK2

OK
```

3.2.3 AT+CICCID Read ICCID from SIM card

Description

This command is used to Read the ICCID from SIM card

AT+CICCID Read ICCID from SIM card	
Test Command **AT+CICCID=?**	Response **OK**

Execution Command **AT+CICCID**	Response **+ICCID: <ICCID>** **OK** or **ERROR** or **+CME ERROR: <err>**

Defined values

<ICCID>	Integrate circuit card identity, a standard ICCID is a 20-digit serial number of the SIM card,it presents the publish state, network code, publish area, publish date, publish manufacture and press serial number of the SIM card.

Examples

```
AT+CICCID
+ICCID: 898600700907A6019125

OK
```

3.2.4 AT+CSIM Generic SIM access

Description

This command is used to control the SIM card directly.

Compared to restricted SIM access command AT+CRSM, AT+CSIM allows the ME to take more control over the SIM interface.

For SIM–ME interface please refer 3GPP TS 11.11.

NOTE： The SIM Application Toolkit functionality is not supported by AT+CSIM. Therefore the following SIM commands can not be used: TERMINAL PROFILE, ENVELOPE, FETCH and TEMINAL RESPONSE.

AT+CSIM Generic SIM access	
Test Command **AT+CSIM=?**	Response **OK**

Write Command **AT+CSIM=<length>,<command>**	Response **+CSIM: <length>,<response>** **OK** or **ERROR** or **+CME ERROR: <err>**

Defined values

<length>	Interger type; length of characters that are sent to TE in <command>or<response>
<command>	Command passed from MT to SIM card.
<response>	Response to the command passed from SIM card to MT.

Examples

```
AT+CSIM=?
OK
```

3.2.5 AT+CRSM Restricted SIM access

Description

By using AT+CRSM instead of Generic SIM Access AT+CSIM, TE application has easier but more limited access to the SIM database.

Write command transmits to the MT the SIM <command> and its required parameters. MT handles internally all SIM-MT interface locking and file selection routines. As response to the command, MT sends the actual SIM information parameters and response data. MT error result code **+CME ERROR** may be returned when the command cannot be passed to the SIM, but failure in the execution of the command in the SIM is reported in <sw1> and <sw2> parameters.

AT+CRSM Restricted SIM access	
Test Command **AT+CRSM=?**	Response **OK**

Write Command **AT+CRSM=<command>[,<fileID>[,<p1>,<p2>, <p3>[,<data>]]]**	Response **+CRSM: <sw1>,<sw2>[,<response>]** **OK** or **ERROR** or **+CME ERROR: <err>**

Defined values

<command>	Command passed on by the MT to the SIM: 176 – READ BINARY 178 – READ RECORD 192 – GET RESPONSE 214 – UPDATE BINARY 220 – UPDATE RECORD 242 – STATUS 203 – RETRIEVE DATA 219 – SET DATA
<fileID>	Identifier for an elementary data file on SIM, if used by <command>. The following list the fileID hex value, user needs to convet them to decimal. EFs under MF 0x2FE2 ICCID 0x2F05 Extended Language Preferences 0x2F00 EF DIR 0x2F06 Access Rule Reference EFs under USIM ADF 0x6F05 Language Indication 0x6F07 IMSI 0x6F08 Ciphering and Integrity keys 0x6F09 C and I keys for pkt switched domain 0x6F60 User controlled PLMN selector w/Acc Tech 0x6F30 User controlled PLMN selector 0x6F31 HPLMN search period 0x6F37 ACM maximum value 0x6F38 USIM Service table 0x6F39 Accumulated Call meter 0x6F3E Group Identifier Level 0x6F3F Group Identifier Level 2 0x6F46 Service Provider Name 0x6F41 Price Per Unit and Currency table

0x6F45	Cell Bcast Msg identifier selection
0x6F78	Access control class
0x6F7B	Forbidden PLMNs
0x6F7E	Location information
0x6FAD	Administrative data
0x6F48	Cell Bcast msg id for data download
0x6FB7	Emergency call codes
0x6F50	Cell bcast msg id range selection
0x6F73	Packet switched location information
0x6F3B	Fixed dialling numbers
0x6F3C	Short messages
0x6F40	MSISDN
0x6F42	SMS parameters
0x6F43	SMS Status
0x6F49	Service dialling numbers
0x6F4B	Extension 2
0x6F4C	Extension 3
0x6F47	SMS reports
0x6F80	Incoming call information
0x6F81	Outgoing call information
0x6F82	Incoming call timer
0x6F83	Outgoing call timer
0x6F4E	Extension 5
0x6F4F	Capability Config Parameters 2
0x6FB5	Enh Multi Level Precedence and Pri
0x6FB6	Automatic answer for eMLPP service
0x6FC2	Group identity
0x6FC3	Key for hidden phonebook entries
0x6F4D	Barred dialling numbers
0x6F55	Extension 4
0x6F58	Comparison Method information
0x6F56	Enabled services table
0x6F57	Access Point Name Control List
0x6F2C	De-personalization Control Keys
0x6F32	Co-operative network list
0x6F5B	Hyperframe number
0x6F5C	Maximum value of Hyperframe number
0x6F61	OPLMN selector with access tech
0x6F5D	OPLMN selector
0x6F62	HPLMN selector with access technology
0x6F06	Access Rule reference
0x6F65	RPLMN last used access tech
0x6FC4	Network Parameters
0x6F11	CPHS: Voice Mail Waiting Indicator
0x6F12,	CPHS: Service String Table
0x6F13	CPHS: Call Forwarding Flag

0x6F14	CPHS: Operator Name String
0x6F15	CPHS: Customer Service Profile
0x6F16	CPHS: CPHS Information
0x6F17	CPHS: Mailbox Number
0x6FC5	PLMN Network Name
0x6FC6	Operator PLMN List
0x6F9F	Dynamic Flags Status
0x6F92	Dynamic2 Flag Setting
0x6F98	Customer Service Profile Line2
0x6F9B	EF PARAMS - Welcome Message
0x4F30	Phone book reference file
0x4F22	Phone book synchronization center
0x4F23	Change counter
0x4F24	Previous Unique Identifier
0x4F20	GSM ciphering key Kc
0x4F52	GPRS ciphering key
0x4F63	CPBCCH information
0x4F64	Investigation scan
0x4F40	MExE Service table
0x4F41	Operator Root Public Key
0x4F42	Administrator Root Public Key
0x4F43	Third party Root public key
0x6FC7	Mail Box Dialing Number
0x6FC8	Extension 6
0x6FC9	Mailbox Identifier
0x6FCA	Message Waiting Indication Status
0x6FCD	Service Provider Display Information
0x6FD2	UIM_USIM_SPT_TABLE
0x6FD9	Equivalent HPLMN
0x6FCB	Call Forwarding Indicator Status
0x6FD6	GBA Bootstrapping parameters
0x6FDA	GBA NAF List
0x6FD7	MBMS Service Key
0x6FD8	MBMS User Key
0x6FCE	MMS Notification
0x6FD0	MMS Issuer connectivity parameters
0x6FD1	MMS User Preferences
0x6FD2	MMS User connectivity parameters
0x6FCF	Extension 8
0x5031	Object Directory File
0x5032	Token Information File
0x5033	Unused space Information File
	EFs under Telecom DF
0x6F3A	Abbreviated Dialing Numbers
0x6F3B	Fixed dialling numbers
0x6F3C	Short messages

	0x6F3D Capability Configuration Parameters 0x6F4F Extended CCP 0x6F40 MSISDN 0x6F42 SMS parameters 0x6F43 SMS Status 0x6F44 Last number dialled 0x6F49 Service Dialling numbers 0x6F4A Extension 1 0x6F4B Extension 2 0x6F4C Extension 3 0x6F4D Barred Dialing Numbers 0x6F4E Extension 4 0x6F47 SMS reports 0x6F58 Comparison Method Information 0x6F54 Setup Menu elements 0x6F06 Access Rule reference 0x4F20 Image 0x4F30 Phone book reference file 0x4F22 Phone book synchronization center 0x4F23 Change counter 0x4F24 Previous Unique Identifier
<p1><p2><p3>	Integer type; parameters to be passed on by the Module to the SIM.
<data>	Information which shall be written to the SIM (hexadecimal character format, refer AT+CSCS).
<sw1><sw2>	Status information from the SIM about the execution of the actual command. It is returned in both cases, on successful or failed execution of the command.
<response>	Response data in case of a successful completion of the previously issued command. “STATUS” and “GET RESPONSE” commands return data, which gives information about the currently selected elementary data field. This information includes the type of file and its size. After “READ BINARY” or “READ RECORD” commands the requested data will be returned. <response> is empty after “UPDATE BINARY” or “UPDATE RECORD” commands.

Examples

```
AT+CRSM=?
OK
```

3.2.6 AT+SPIC Times remain to input SIM PIN/PUK

Description

This command is used to inquire times remain to input SIM PIN/PUK.

AT+SPIC Times remain to input SIM PIN/PUK	
Test Command **AT+SPIC=?**	Response **OK**
Execution Command **AT+SPIC**	Response **+SPIC: <pin1>,<puk1>,<pin2>,<puk2>** **OK**

Defined values

<pin1>	Times remain to input PIN1 code.
<puk1>	Times remain to input PUK1 code.
<pin2>	Times remain to input PIN2 code.
<puk2>	Times remain to input PUK2 code.

Examples

```
AT+SPIC=?
OK
AT+SPIC
+SPIC: 3,10,0,10

OK
```

3.2.7 AT+CSPN Get service provider name from SIM

Description

This command is used to get service provider name from SIM card.

AT+CSPN Get service provider name from SIM	
Test Command **AT+CSPN=?**	Response **OK** or **ERROR**
Read Command **AT+CSPN?**	Response **+CSPN: <spn>,<display mode>** **OK** or **ERROR** or **+CME ERROR: <err>**

Defined values

<spn>	String type; service provider name on SIM
<display mode>	0 – doesn't display PLMN. Already registered on PLMN. 1 – display PLMN

Examples

```
AT+CSPN=?
OK
AT+CSPN?
+CSPN: "CMCC",0

OK
```

3.2.8 AT+CSQ Query signal quality

Description

This command is used to return received signal strength indication <rssi> and channel bit error rate <ber> from the ME. Test command returns values supported by the TA as compound values.

AT+CSQ Query signal quality

Test Command **AT+CSQ=?**	Response **+CSQ: (list of supported <rssi>s),(list of supported <ber>s)** **OK**
Execution Command **AT+CSQ**	Response **+CSQ: <rssi>,<ber>** **OK** or **ERROR**

Defined values

<rssi>	0 – -113 dBm or less 1 – -111 dBm 2...30 – -109... -53 dBm 31 – -51 dBm or greater 99 – not known or not detectable 100 – -116 dBm or less 101 – -115 dBm 102…191 – -114... -26dBm 191 – -25 dBm or greater 199 – not known or not detectable 100…199 – expand to TDSCDMA, indicate RSCP received
<ber>	(in percent) 0 – <0.01% 1 – 0.01% --- 0.1% 2 – 0.1% --- 0.5% 3 – 0.5% --- 1.0% 4 – 1.0% --- 2.0% 5 – 2.0% --- 4.0% 6 – 4.0% --- 8.0% 7 – >=8.0% 99 – not known or not detectable

Examples

```
AT+CSQ
+CSQ: 22,0

OK
```

3.2.9 AT+AUTOCSQ Set CSQ report

Description

This command is used to enable or disable automatic report CSQ information, when automatic report enabled, the module reports CSQ information every five seconds or only after <rssi>or<ber> is changed, the format of automatic report is "+CSQ: <rssi>,<ber>".

AT+AUTOCSQ Set CSQ report	
Test Command **AT+AUTOCSQ=?**	Response **+AUTOCSQ: (list of supported<auto>s),(list of supported<mode>s)** **OK**
Read Command **AT+AUTOCSQ?**	Response **+AUTOCSQ: <auto>,<mode>** **OK**
Write Command **AT+AUTOCSQ=<auto>[,<mode>]**	Response **OK** or **ERROR**

Defined values

<auto>	0 – disable automatic report 1 – enable automatic report
<mode>	0 – CSQ automatic report every five seconds 1 – CSQ automatic report only after <rssi>or<ber>is changed **NOTE:** If the parameter of <mode> is omitted when executing write command, <mode> will be set to default value.

Examples

```
AT+AUTOCSQ=?
+AUTOCSQ: (0-1),(0-1)

OK
AT+AUTOCSQ?
```

```
+AUTOCSQ: 1,1

OK
AT+AUTOCSQ=1,1
OK

+CSQ: 23,0 (when <rssi>or<ber>changing)
```

3.2.10 AT+CSQDELTA Set RSSI delta change threshold

Description

This command is used to set RSSI delta threshold for signal strength reporting.

AT+CSQDELTA Set RSSI delta change threshold	
Test Command **AT+CSQDELTA=?**	Response **+CSQDELTA: (list of supported <delta>s)** **OK**
Read Command **AT+CSQDELTA?**	Response **+CSQDELTA: <delta>** **OK** or **ERROR**
Write Command **AT+CSQDELTA=<delta>**	Response **OK** or **ERROR**
Execution Command **AT+CSQDELTA**	Response ***Set default value（<delta>=5）:*** **OK**

Defined values

<delta>	Range: from 0 to 5.

Examples

```
AT+CSQDELTA?
+CSQDELTA: 5

OK
```

3.2.11 AT+CATR Configure URC destination interface

Description

This command is used to configure the serial port which will be used to output URCs. We recommend configure a destination port for receiving URC in the system initialization phase, in particular, in the case that transmitting large amounts of data, e.g. use TCP/UDP and MT SMS related AT command.

AT+CATR Configure URC destination interface	
Test Command **AT+CATR=?**	Response **+CATR: (list of supported <port>s)** **OK**
Read Command **AT+CATR?**	Response **+CATR: <port>** **OK**
Write Command **AT+CATR=<port>**	Response **OK** or **ERROR**

Defined values

<port>	0 – all ports 1 – use UART port to output URCs 2 – use MODEM port to output URCs 3 – use ATCOM port to output URCs 4 – use cmux virtual port1 to output URCs 5 – use cmux virtual port2 to output URCs 6 – use cmux virtual port3 to output URCs 7 – use cmux virtual port4 to output URCs

Examples

```
AT+CATR=1
OK
AT+CATR?
+CATR: 1

OK
```

3.2.12 AT+CPOF Power down the module

Description

This command is used to power off the module. Once the AT+CPOF command is executed, The module will store user data and deactivate from network, and then shutdown.

AT+CPOF Power down the module	
Test Command **AT+CPOF=?**	Response **OK**
Execution Command **AT+CPOF**	Response **OK**

Examples

```
AT+CPOF
OK
```

3.2.13 AT+CRESET Reset the module

Description

This command is used to reset the module.

AT+CRESET Reset the module	
Test Command **AT+CRESET=?**	Response **OK**

Execution Command	Response
AT+CRESET	**OK**

Examples

```
AT+CRESET=?
OK
AT+CRESET
OK
```

3.2.14 AT+CACM Accumulated call meter

Description

This command is used to reset the Advice of Charge related accumulated call meter value in SIM file EF_{ACM}.

AT+CACM Accumulated call meter	
Test Command **AT+CACM=?**	Response **OK** or **ERROR**
Read Command **AT+CACM?**	Response **+CACM: <acm>** **OK** or **ERROR** or **+CME ERROR: <err>**
Write Command **AT+CACM=<passwd>**	Response **OK** or **ERROR** or **+CME ERROR: <err>**

Execution Command **AT+CACM**	Response **OK** or **ERROR** or **+CME ERROR: <err>**

Defined values

<passwd>	String type, SIM PIN2.
<acm>	String type, accumulated call meter value similarly coded as <ccm> under +CAOC.

Examples

```
AT+CACM?
+CACM: "000000"

OK
```

3.2.15 AT+CAMM Accumulated call meter maximum

Description

This command is used to set the Advice of Charge related accumulated call meter maximum value in SIM file EF_{ACMmax}.

AT+CAMM Accumulated call meter maximum	
Test Command **AT+CAMM=?**	Response **OK** or **ERROR**

Read Command **AT+CAMM?**	Response **+CAMM: <acmmax>** **OK** or **ERROR** or **+CME ERROR: <err>**
Write Command **AT+CAMM=<acmmax>[,<passwd>]**	Response **OK** or **ERROR** or **+CME ERROR: <err>**
Execution Command **AT+CAMM**	Response **OK** or **ERROR** or **+CME ERROR: <err>**

Defined values

<acmmax>	String type, accumulated call meter maximum value similarly coded as <ccm> under AT+CAOC, value zero disables ACMmax feature.
<passwd>	String type, SIM PIN2.

Examples

```
AT+CAMM?
+CAMM: "000000"

OK
```

3.2.16 AT+CPUC Price per unit and currency table

Description

This command is used to set the parameters of Advice of Charge related price per unit and currency table in SIM file EF_{PUCT}..

AT+CPUC Price per unit and currency table	
Test Command **AT+CPUC=?**	Response **OK** or **ERROR**
Read Command **AT+CPUC?**	Response **+CPUC: [<currency>,<ppu>]** **OK** or **ERROR** or **+CME ERROR: <err>**
Write Command **AT+CPUC=<currency>,<ppu>[,<passwd>]**	Response **OK** or **ERROR** or **+CME ERROR: <err>**

Defined values

<currency>	String type, three-character currency code (e.g. "GBP", "DEM"), character set as specified by command Select TE Character Set AT+CSCS.
<ppu>	String type, price per unit, dot is used as a decimal separator. (e.g. "2.66").
<passwd>	String type, SIM PIN2.

Examples

```
AT+CPUC?
+CPUC: “GBP” , “2.66”

OK
```

3.2.17 AT+CCLK Real time clock management

Description

This command is used to manage Real Time Clock of the module.

AT+CCLK Real time clock management	
Test Command **AT+CCLK=?**	Response **OK**
Read Command **AT+CCLK?**	Response **+CCLK: <time>** **OK**
Write Command **AT+CCLK=<time>**	Response **OK** or **ERROR**

Defined values

<time>	String type value; format is "yy/MM/dd,hh:mm:ss±zz", where characters indicate year (two last digits), month, day, hour, minutes, seconds and time zone (indicates the difference, expressed in quarters of an hour, between the local time and GMT; three last digits are mandatory, range -47…+48). E.g. 6th of May 2008, 14:28:10 GMT+8 equals to "08/05/06,14:28:10+32". **NOTE:** 1. Time zone is nonvolatile, and the factory value is invalid time zone. 2. Command +CCLK? will return time zone when time zone is valid, and if time zone is 00, command +CCLK? will return "+00", but not "-00".

Examples

```
AT+CCLK="08/11/28,12:30:33+32"
OK
AT+CCLK?
+CCLK: "08/11/28,12:30:35+32"
```

```
OK
AT+CCLK=“08/11/26,10:15:00”
OK
AT+CCLK?
+CCLK: “08/11/26,10:15:02+32”

OK
```

3.2.18 AT+CMEE Report mobile equipment error

Description

This command is used to disable or enable the use of result code “**+CME ERROR**: <err>” or “+CMS **ERROR**: <err>” as an indication of an error relating to the functionality of ME; when enabled, the format of <err> can be set to numeric or verbose string.

AT+CMEE Report mobile equipment error	
Test Command **AT+CMEE=?**	Response **+CMEE: (list of supported <n>s)** **OK**
Read Command **AT+CMEE?**	Response **+CMEE: <n>** **OK**
Write Command **AT+CMEE=<n>**	Response **OK** or **ERROR**
Execution Command **AT+CMEE**	Response ***Set default value:*** **OK**

Defined values

<n>	0 –Disable result code,i.e. only “**ERROR**” will be displayed. 1 –Enable error result code with numeric values. 2 –Enable error result code with string values.

Examples

```
AT+CMEE?
+CMEE: 2

OK
AT+CPIN="1234","1234"
+CME ERROR: incorrect password
AT+CMEE=0

OK
AT+CPIN="1234","1234"
ERROR
AT+CMEE=1
OK
AT+CPIN="1234","1234"
+CME ERROR: 16
```

3.2.19 AT+CPAS Phone activity status

Description

This command is used to return the activity status <pas> of the ME. It can be used to interrogate the ME before requesting action from the phone.

NOTE: This command is same as AT+CLCC, but AT+CLCC is more commonly used. So AT+CLCC is recommended to use.

AT+CPAS Phone activity status	
Test Command **AT+CPAS=?**	Response **+CPAS: (list of supported <pas>s)** **OK**
Execution Command **AT+CPAS**	Response **+CPAS: <pas>** **OK**

Defined values

<pas>	0 – ready (ME allows commands from TA/TE) 3 – ringing (ME is ready for commands from TA/TE, but the ringer is active) 4 – call in progress (ME is ready for commands from TA/TE, but a call is in progress)

Examples

```
RING (with incoming call)
AT+CPAS
+CPAS: 3

OK
AT+CPAS=?
+CPAS: (0,3,4)

OK
```

3.2.20 AT+SIMEI Set IMEI for the module

Description

This command is used to set the module's IMEI value.

AT+SIMEI Set IMEI for the module	
Test Command **AT+SIMEI=?**	Response **OK**
Read Command **AT+SIMEI?**	Response **+SIMEI: <imei>** **OK** or **ERROR**
Write Command **AT+SIMEI=<imei>**	Response **OK** or **ERROR**

Defined values

<imei>	The 15-digit IMEI value.

Examples

```
AT+SIMEI=357396012183170
OK
AT+SIMEI?
+SIMEI:357396012183170

OK
AT+SIMEI=?
OK
```

3.2.21 AT+SMEID RequestMobile Equipment Identifier

Description

Only task effect in 7600CE

AT+SMEID RequestMobile Equipment Identifier	
Read Command **AT+SMEID?**	Responses **+SMEID: <MEID>** **OK** or **ERROR**

Defined values

<MEID>	Mobile Equipment Identifier (string, without double quotes).

Examples

```
AT+SMEID?
+SMEID: A1000021A5906F
```

OK

3.2.22 AT+CSVM Voice Mail Subscriber number

Description

Execution command returns the voice mail number related to the subscriber.

AT+CSVM Voice Mail Subscriber number	
Test Command **AT+CSVM=?**	Response **+CSVM: (0-1), "(0-9,+)", (128-255)** **OK** or **ERROR**
Read Command **AT+CSVM?**	Response **+CSVM: <valid>, "<number>",<type>** **OK** or **ERROR**
Write Command **AT+CSVM=<valid>, "<number>",<type>**	Response **OK** or **ERROR**

Defined values

<valid>	Whether voice mail number is valid: 0 – Voice mail number is invalid. 1 – Voice mail number is valid.
<number>	String type phone number of format specified by <type>.
<type>	Type of address octet in integer format. see also AT+CPBR <type>

Examples

```
AT+CSVM?
+CSVM: 1 ,"13697252277",129

OK
```

3.2.23 Indication of Voice Mail

This module supports voice mail function; the subscriber number is configured by AT+CSVM command, the following table shows the URC related Voice Mail.

Indication of Voice Mail	
Box Empty **+VOICEMAIL: EMPTY**	Description This indication means the voice mail box is empty
New Message **+VOICEMAIL: NEW MSG**	Description This indication means there is a new voice mail message notification received. This is for CPHS.
Voice Mail Status Updated **+VOICEMAIL: WAITING, <count>**	Description This indication means that there are <count> number of voice mail messages that needs to be got.

Defined values

<count>	Count of voice mail message that waits to be got.

Examples

```
+VOICEMAIL: WAITING, <count>
+VOICEMAIL: WAITING, 5
```

3.3 Summary of CME ERROR codes

This result code is similar to the regular ERROR result code. The format of <err> can be ethier numeric or verbose string, by setting AT+CMEE command.

<err> of numeric format	<err> of verbose format

0	Phone failure
1	no connection to phone
2	phone adaptor link reserved
3	operation not allowed
4	operation not supported
5	PH-SIM PIN required
6	PH-FSIM PIN required
7	PH-FSIM PUK required
10	SIM not inserted
11	SIM PIN required
12	SIM PUK required
13	SIM failure
14	SIM busy
15	SIM wrong
16	incorrect password
17	SIM PIN2 required
18	SIM PUK2 required
20	memory full
21	invalid index
22	not found
23	memory failure
24	text string too long
25	invalid characters in text string
26	dial string too long
27	invalid characters in dial string
30	no network service
31	network timeout
32	network not allowed – emergency calls only
40	network personalization PIN required
41	network personalization PUK required
42	network subset personalization PIN required
43	network subset personalization PUK required
44	service provider personalization PIN required
45	service provider personalization PUK required
46	corporate personalization PIN required
47	corporate personalization PUK required
100	Unknown
103	Illegal message
106	Illegal ME
107	GPRS services not allowed
111	PLMN not allowed

112	Location area not allowed
113	Roaming not allowed in this location area
132	service option not supported
133	requested service option not subscribed
134	service option temporarily out of order
148	unspecified GPRS error
149	PDP authentication failure
150	invalid mobile class
257	network rejected request
258	retry operation
259	invalid deflected to number
260	deflected to own number
261	unknown subscriber
262	service not available
263	unknown class specified
264	unknown network message
273	minimum TFTS per PDP address violated
274	TFT precedence index not unique
275	invaild parameter combination

“CME ERROR” codes of FTP

201	Unknown error for FTP
202	FTP task is busy
203	Failed to resolve server address
204	FTP timeout
205	Failed to read file
206	Failed to write file
207	It’s not allowed in current state
208	Failed to login
209	Failed to logout
210	Failed to transfer data
211	FTP command rejected by server
212	Memory error
213	Invalid parameter
214	Network error

Example

```
AT+CPIN=”1234”,”1234”
```

+CME ERROR: incorrect password

3.4 Summary of CMS ERROR codes

Final result code +CMS ERROR: <err> indicates an error related to mobile equipment or network. The operation is simialer to ERROR result code. None of the following commands in the same command line is executed. Neither ERROR nor OK result code shall be returned. ERROR is returned normally when error is related to syntax or invalid parameters. The format of <err> can be either numeric or verbose. This is set with command AT+CMEE.

<err> of numeric format	<err> of verbose format
300	ME failure
301	SMS service of ME reserved
302	Operation not allowed
303	Operation not supported
304	Invalid PDU mode parameter
305	Invalid text mode parameter
310	SIM not inserted
311	SIM PIN required
312	PH-SIM PIN required
313	SIM failure
314	SIM busy
315	SIM wrong
316	SIM PUK required
317	SIM PIN2 required
318	SIM PUK2 required
320	Memory failure
321	Invalid memory index
322	Memory full
330	SMSC address unknown
331	No network service
332	Network timeout
340	NO +CNMA ACK EXPECTED
341	Buffer overflow
342	SMS size more than expected
500	Unknown error

Example

```
AT+CMGS=02112345678
+CMS ERROR: 304
```

4. AT Commands for Network

4.1 Overview of AT Commands for Network

Command	Description
AT+CREG	Network Registration
AT+COPS	Operator selection
AT+CLCK	Facility lock
AT+CPWD	Change password
AT+CCUG	Closed User Group
AT+CUSD	Unstructured supplementary service data
AT+CAOC	Advice of Charge
AT+CSSN	Supplementary service notifications
AT+CPOL	Preferred mode selection
AT+COPN	Read operator names
AT+CNMP	Preferred mode selection
AT+CNBP	Preferred band selection
AT+CNAOP	Acquisition order preference
AT+CPSI	Inquiring UE system information
AT+CNSMOD	Show network system mode
AT+CEREG	EPS network registration status
AT+CTZU	Automatic time and time zone update
AT+CTZR	Time and time zone reporting

4.2 Detailed Description of AT Commands for Network

4.2.1 AT+CREG Network registration

This command is used to control the presentation of an unsolicited result code +CREG: <stat> when <n>=1 and there is a change in the ME network registration status, or code +CREG: <stat>[,<lac>,<ci>] when <n>=2 and there is a change of the network cell.

Read command returns the status of result code presentation and an integer <stat> which shows whether the network has currently indicated the registration of the ME. Location information elements <lac> and <ci> are returned only when <n>=2 and ME is registered in the network.

AT+CREG Network registration	
Test Command **AT+CREG=?**	Response **+CREG:** (list of supported <**n**>s) **OK**
Read Command **AT+CREG?**	Response **+CREG: <n>,<stat>[,<lac>,<ci>]** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Write Command **AT+CREG=<n>**	Response **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Execution Command **AT+CREG**	Response (Set default value “<n>=0”): **OK**

Defined Values

<n>	0 disable network registration unsolicited result code 1 enable network registration unsolicited result code +CREG: <stat> 2 enable network registration and location information unsolicited result code +CREG: <stat>[,<lac>,<ci>]
<stat>	0 – not registered, ME is not currently searching a new operator to register to 1 registered, home network 2 not registered, but ME is currently searching a new operator to register to 3 registration denied 4 unknown 5 registered, roaming
<lac>	Two byte location area code in hexadecimal format(e.g.”00C3” equals 193 in decimal). NOTE: The <lac> not supported in CDMA/HDR mode
<ci>	Cell Identify in hexadecimal format.

	GSM : Maximum is two byte WCDMA : Maximum is four byte TDS-CDMA : Maximum is four byte NOTE: The <ci> not supported in CDMA/HDR mode

Example

```
AT+CREG?
+CREG: 0,1

OK
```

NOTE

- Location information elements <lac> and <ci> are returned only when <n>=2 and ME is registered in the network

4.2.2 AT+COPS Operator selection

Write command forces an attempt to select and register the GSM/UMTS network operator. <mode> is used to select whether the selection is done automatically by the ME or is forced by this command to operator <oper> (it shall be given in format <format>). If the selected operator is not available, no other operator shall be selected (except <mode>=4). The selected operator name format shall apply to further read commands (AT+COPS?) also. <mode>=2 forces an attempt to deregister from the network. The selected mode affects to all further network registration (e.g. after <mode>=2, ME shall be unregistered until <mode>=0 or 1 is selected).
Read command returns the current mode and the currently selected operator. If no operator is selected, <format> and <oper> are omitted.
Test command returns a list of quadruplets, each representing an operator present in the network. Quadruplet consists of an integer indicating the availability of the operator <stat>, long and short alphanumeric format of the name of the operator, and numeric format representation of the operator. Any of the formats may be unavailable and should then be an empty field. The list of operators shall be in order: home network, networks referenced in SIM, and other networks.
It is recommended (although optional) that after the operator list TA returns lists of supported <mode>s and <format>s. These lists shall be delimited from the operator list by two commas. When executing AT+COPS=? , any input from serial port will stop this command.

AT+COPS Operator selection	
Test Command	Response

AT+COPS=?	**[+COPS: [**list of supported **(<stat>,long alphanumeric <oper> ,short alphanumeric <oper>,numeric <oper>[,< AcT>])**s**]** **[,,(**list of supported **<mode>**s**),(**list of supported **<format>**s**)]]** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Read Command **AT+COPS?**	Response **+COPS: <mode>[,<format>,<oper>[,< AcT>]]** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Write Command **AT+COPS=<mode>[,<format >[,<oper>[,< AcT>]]]**	Response **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Execution Command **AT+COPS**	Response **OK**

Defined Values

<mode>	0 automatic 1 manual 2 force deregister 3 set only <format> 4 manual/automatic 5 manual,but do not modify the network selection mode(e.g GSM,WCDMA) after module resets. NOTE: if <mode> is set to 1, 4, 5 in write command, the <oper> is needed.
<format>	0 long format alphanumeric <oper> 1 short format alphanumeric <oper> 2 numeric <oper>
<oper>	string type, <format> indicates if the format is alphanumeric or numeric.
<stat>	0 unknown 1 available

	2 current 3 forbidden
<AcT>	Access technology selected 0 GSM 1 GSM Compact 2 UTRAN 7 EUTRAN 8 CDMA/HDR NOTE: the value 8 do not follow the 3gpp spec, we add this value to distinguish cdma/hdr.

Example

```
AT+COPS?
+ COPS: 0,0,"China Mobile Com",0

OK

AT+COPS=?
+ COPS: (2,"China Unicom","Unicom","46001",0),(3,"China Mobile Com","DGTMPT",
"46000",0),,(0,1,2,3,4,5),(0,1,2)

OK
```

NOTE

- When executing AT+COPS=? , any input from serial port will stop this command.

4.2.3 AT+CLCK Facility lock

This command is used to lock, unlock or interrogate a ME or a network facility <fac>. Password is normally needed to do such actions. When querying the status of a network service (<mode>=2) the response line for 'not active' case (<status>=0) should be returned only if service is not active for any <class>.

AT+CLCK Facility lock	
Test Command **AT+CLCK=?**	Response **+CLCK: (**list of supported **<fac>**s**)** **OK**

	or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Write Command **AT+CLCK=<fac>,<mode>[,<passwd>[,<class>]]**	Response (When <mode>=2 and command successful:) **[+CLCK:<status>[,<class1>[<CR><LF>** **+CLCK: <status>,<class2>** **[...]]** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Defined Values

<fac>	"PF" lock Phone to the very First inserted SIM card or USIM card "SC" lock SIM card or USIM card "AO" Barr All Outgoing Calls "OI" Barr Outgoing International Calls "OX" Barr Outgoing International Calls except to Home Country "AI" Barr All Incoming Calls "IR" Barr Incoming Calls when roaming outside the home country "AB" All Barring services (only for <mode>=0) "AG" All outGoing barring services (only for <mode>=0) "AC" All inComing barring services (only for <mode>=0) "FD" SIM fixed dialing memory feature "PN" Network Personalization "PU" network subset Personalization "PP" service Provider Personalization "PC" Corporate Personalization
<mode>	0 unlock 1 lock 2 query status
<status>	0 not active 1 active
<passwd>	Password. string type; shall be the same as password specified for the facility from the ME user interface or with command Change Password +CPWD
<classX>	It is a sum of integers each representing a class of information (default 7): 1 voice (telephony) 2 data (refers to all bearer services)

	4 fax (facsimile services) 8 short message service 16 data circuit sync 32 data circuit async 64 dedicated packet access 128 dedicated PAD access 255 The value 255 covers all classes
<nlength>	Integer type value indicating the maximum length of field <number>
<tlength>	Integer type value indicating the maximum length of field <text>.

Example

```
AT+CLCK="SC",2
+CLCK: 0

OK
```

NOTE

- When querying the status of a network service (<mode>=2) the response line for 'not active' case (<status>=0) should be returned only if service is not active for any <class>.

4.2.4 AT+CPWD Change password

Write command sets a new password for the facility lock function defined by command Facility Lock AT+CLCK.
Test command returns a list of pairs which present the available facilities and the maximum length of their password.

AT+CPWD Change password	
Test Command **AT+CPWD=?**	Response **+CPWD: (**list of supported **(<fac>,<pwdlength>)**s**)** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Write Command **AT+CPWD=<fac>,<oldpwd>,**	Response **OK**

<newpwd>	or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Defined Values

<fac>	Refer Facility Lock +CLCK for other values: "SC" SIM or USIM PIN1 "P2" SIM or USIM PIN2 "AB" All Barring services "AC" All inComing barring services (only for <mode>=0) "AG" All outGoing barring services (only for <mode>=0) "AI" Barr All Incoming Calls "AO" Barr All Outgoing Calls "IR" Barr Incoming Calls when roaming outside the home country "OI" Barr Outgoing International Calls "OX" Barr Outgoing International Calls except to Home Country
<oldpwd>	String type, it shall be the same as password specified for the facility from the ME user interface or with command Change Password AT+CPWD.
<newpwd>	String type, it is the new password; maximum length of password can be determined with <pwdlength>.
<pwdlength	Integer type, max length of password.

Example

```
AT+CPWD=?
+CPWD: ("AB",4),("AC",4),("AG",4),("AI",4),("AO",4),("IR",4),("OI",4),("OX",4),(
"SC",8),("P2",8)

OK
```

4.2.5 AT+CCUG Closed user group

This command allows control of the Closed User Group supplementary service. Set command enables the served subscriber to select a CUG index, to suppress the Outgoing Access (OA), and to suppress the preferential CUG.

AT+CCUG Closed user group	
Test Command **AT+CCUG=?**	Response **OK**

	or **ERROR**
Read Command **AT+CCUG?**	Response **+CCUG: <n>,<index>,<info>** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Write Command **AT+CCUG=<n>[,<index>[,<info>]]**	Response **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Execution Command **AT+CCUG**	Response (Set default value): **OK**

Defined Values

<n>	0 disable CUG temporary mode 1 enable CUG temporary mode
<index>	0...9 CUG index 10 no index (preferred CUG taken from subscriber data)
<info>	0 no information 1 suppress OA 2 suppress preferential CUG 3 suppress OA and preferential CUG

Example

```
AT+CCUG?
+CCUG: 0,0

OK
```

NOTE

- This command not supported in CDMA/HDR mode

4.2.6 AT+CUSD Unstructured supplementary service data

This command allows control of the Unstructured Supplementary Service Data (USSD). Both network and mobile initiated operations are supported. Parameter <n> is used to disable/enable the presentation of an unsolicited result code (USSD response from the network, or network initiated operation) +CUSD: <m>[,<str>,<dcs>] to the TE. In addition, value <n>=2 is used to cancel an ongoing USSD session.

AT+CUSD Unstructured supplementary service data	
Test Command **AT+CUSD=?**	Response **+CUSD: (**list of supported **<n>**s**)** **OK**
Read Command **AT+CUSD?**	Response **+CUSD: <n>** **OK**
Write Command **AT+CUSD=<n>[,<str>[,<dcs>]]**	Response **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Execution Command **AT+CUSD**	Response (Set default value): **OK**

Defined Values

<n>	0 disable the result code presentation in the TA 1 enable the result code presentation in the TA 2 cancel session (not applicable to read command response)
<str>	String type USSD string.
<dcs>	Cell Broadcast Data Coding Scheme in integer format (default 0).
<m>	0 no further user action required (network initiated USSD Notify, or no further information needed after mobile initiated operation) 1 further user action required (network initiated USSD Request, or further information needed after mobile initiated operation) 2 USSD terminated by network 4 operation not supported 5 network time out

Example

```
AT+CUSD?
+ CUSD: 1
```

OK

AT+CUSD=0
OK

NOTE

- This command not supported in CDMA/HDR mode

4.2.7 AT+CAOC Advice of Charge

This command refers to Advice of Charge supplementary service that enables subscriber to get information about the cost of calls. With <mode>=0, the execute command returns the current call meter value from the ME.
This command also includes the possibility to enable an unsolicited event reporting of the CCM information. The unsolicited result code +CCCM: <ccm> is sent when the CCM value changes, but not more that every 10 seconds. Deactivation of the unsolicited event reporting is made with the same command.

AT+CAOC Advice of Charge	
Test Command **AT+CAOC=?**	Response **+CAOC: (**list of supported **<mode>**s**)** **OK**
Read Command **AT+CAOC?**	Response **+CUSD: <mode>** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Write Command **AT+CAOC=<mode>**	Response **+ CAOC: <ccm>** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Execution Command **AT+CAOC**	Response (Set default value): **OK** or **ERROR**

Defined Values

<mode>	0 query CCM value 1 deactivate the unsolicited reporting of CCM value 2 activate the unsolicited reporting of CCM value
ccm>	String type, three bytes of the current call meter value in hexadecimal format (e.g. "00001E" indicates decimal value 30), value is in home units and bytes are similarly coded as ACMmax value in the SIM.

Example

```
AT+CAOC=0
+CAOC: "000000"

OK
```

NOTE

- This command not supported in CDMA/HDR mode

4.2.8 AT+CSSN Supplementary service notifications

This command refers to supplementary service related network initiated notifications. The set command enables/disables the presentation of notification result codes from TA to TE.

When <n>=1 and a supplementary service notification is received after a mobile originated call setup, intermediate result code +CSSI: <code1>[,<index>] is sent to TE before any other MO call setup result codes presented in the present document. When several different <code1>s are received from the network, each of them shall have its own +CSSI result code.

When <m>=1 and a supplementary service notification is received during a mobile terminated call setup or during a call, or when a forward check supplementary service notification is received, unsolicited result code +CSSU: <code2>[,<index>[,<number>,<type>[,<subaddr>,<satype>]]] is sent to TE. In case of MT call setup, result code is sent after every +CLIP result code (refer command "Calling line identification presentation +CLIP") and when several different <code2>s are received from the network, each of them shall have its own +CSSU result code.

AT+CSSN Supplementary service notifications	
Test Command **AT+CSSN?**	Response **+CSSN: (**list of supported **<n>**s**),(**list of supported **<m>**s**)** **OK**
Execution Command **AT+CSSN=<value>**	Response **+CSSN: <n>,<m>** **OK** or **ERROR**
Write Command **AT+CSSN=<n>[,<m>]**	Response **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Defined Values

<n>	Parameter sets/shows the +CSSI result code presentation status in the TA: 0 disable 1 enable
<m>	Parameter sets/shows the +CSSU result code presentation status in the TA: 0 disable 1 enable
<code1>	0 unconditional call forwarding is active 1 some of the conditional call forwarding are active 2 call has been forwarded 3 call is waiting 5 outgoing calls are barred
<index>	Refer "Closed user group +CCUG".
<code2>	0 this is a forwarded call (MT call setup) 2 call has been put on hold (during a voice call) 3 call has been retrieved (during a voice call) 5 call on hold has been released (this is not a SS notification) (during a voice call)
<number>	String type phone number of format specified by <type>.
<type>	Type of address octet in integer format; default 145 when dialing string includes international access code character "+", otherwise 129.
<subaddr>	String type sub address of format specified by <satype>.
<satype>	Type of sub address octet in integer format, default 128.

Example

```
AT+CSSN=1
OK

AT+CSSN?
+CSSN: 1,1

OK
```

NOTE

- This command not supported in CDMA/HDR mode

4.2.9 AT+CPOL Preferred operator list

This command is used to edit the SIM preferred list of networks.

AT+CPOL Preferred operator list	
Test Command **AT+CPOL**	Response **+CPOL: (**list of supported **<index>**s**), (**list of supported **<format>**s**)** **OK**
Read Command **AT+CPOL?**	Response **[+CPOL:<index1>,<format>,<oper1>[<GSM_AcT1>,<GSM_Compact_AcT1>,<UTRAN_AcT1>,<LTE_AcT1>][<CR><LF>** **+CPOL:** **<index2>,<format>,<oper2>[,<GSM_AcT1>,<GSM_Compact_AcT1>,<UTRAN_AcT1>,<LTE_AcT1>]** **[...]]]** **OK** or **ERROR**
Write Command **AT+CPOL=<index>[,<format>[,<oper>][,<GSM_AcT1>,<GSM_Compact_AcT1>,<UTRAN_AcT1>,<LTE_AcT1>]]** NOTE: If using USIM card, the	Response **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

last four parameters must set.

Defined Values

<index>	Integer type, the order number of operator in the SIM preferred operator list. If only input <index>, command will delete the value indicate by <index>.
<format>	0 long format alphanumeric <oper> 1 short format alphanumeric <oper> 2 numeric <oper>
<operX>	String type.
<GSM_AcTn>	GSM access technology: 0 access technology not selected 1 access technology selected
<GSM_Compact_AcTn>	GSM access technology: 0 access technology not selected 1 access technology selected
<UTRA_AcTn>	UTRA access technology: 0 access technology not selected 1 access technology selected
<LTE_AcTn>	LTE access technology: 0 access technology not selected 1 access technology selected

Example

```
AT+CPOL?
+CPOL: 1,2,"46001",0,0,1,0

OK

AT+CPOL=?
+CPOL: (1-8),(0-2)

OK
```

4.2.10 AT+COPN Read operator names

This command is used to return the list of operator names from the ME. Each operator code <numericX> that has an alphanumeric equivalent <alphaX> in the ME memory shall be returned.

AT+COPN Read operator names	
Test Command **AT+COPN=?**	Response **OK** or **ERROR**
Write Command **AT+COPN**	Response **[+COPN:<numeric1>,<alpha1>[<CR><LF>** **+COPN: <numeric2>,<alpha2>** **[...]]** **OK** or If error is related to ME functionality: **+CME ERROR: <err>**

Defined Values

<numericX>	String type, operator in numeric format (see AT+COPS).
<alphaX>	String type, operator in long alphanumeric format (see AT+COPS).

Example

```
AT+COPN
+COPN: "46000","China Mobile Com"
+COPN: "46001"," China Unicom"

.....

OK
```

4.2.11 AT+CNMP Preferred mode selection

This command is used to select or set the state of the mode preference.

AT+CNMP Preferred mode selection	
Test Command **AT+CNMP=?**	Response **+CNMP: (**list of supported **<mode>**s**)** **OK**
Read Command **AT+CNMP?**	Response **+CNMP: <mode>**

	OK
Write Command **AT+CNMP=<mode>**	Response **OK** or (If <mode> not supported by module, this command will return ERROR.) **ERROR**

Defined Values

<mode>	2 Automatic 13 GSM Only 14 WCDMA Only 38 LTE Only 59 TDS-CDMA Only 9 CDMA Only 10 EVDO Only 19 GSM+WCDMA Only 22 CDMA+EVDO Only 48 Any but LTE 60 GSM+TDSCDMA Only 63 GSM+WCDMA+TDSCDMA Only 67 CDMA+EVDO+GSM+WCDMA+TDSCDMA Only 39 GSM+WCDMA+LTE Only 51 GSM+LTE Only 54 WCDMA+LTE Only

Example

```
AT+CNMP=13
OK

AT+CNMP?
+CNMP: 13

OK
```

NOTE

- The set value in Write Command will take efficient immediately; The set value will retain after module reset
- The response will be returned immediately for Test Command and Read Command; The maximum response time for Write Command is 10 seconds

4.2.12 AT+CNBP Preferred band selection

This command is used to select or set the state of the band preference.

AT+CNBP Preferred band selection	
Test Command **AT+CNBP?**	Response **+CNBP: <mode>[,<lte_mode>][,<tds_mode>]** **OK**
Read Command **AT+CNBP=<mode>[,<lte_mode>][,<tds_mode>]**	Response **OK** Or **ERROR**

Defined Values

<mode>	64 bit number, the value is "1" << "<pos>", then or by bit. Some special mode value declared below: 0x40000000 BAND_PREF_NO_CHANGE
<pos>	Value: 0xFFFFFFFF7FFFFFFF Any (any value) 7 GSM_DCS_1800 8 GSM_EGSM_900 9 GSM_PGSM_900 16 GSM_450 17 GSM_480 18 GSM_750 19 GSM_850 20 GSM_RGSM_900 21 GSM_PCS_1900 22 WCDMA_IMT_2000 23 WCDMA_PCS_1900 24 WCDMA_III_1700 25 WCDMA_IV_1700 26 WCDMA_850 27 WCDMA_800 48 WCDMA_VII_2600 49 WCDMA_VIII_900 50 WCDMA_IX_1700
<lte_mode>	64/256 bit number, the value is "1" << "<lte_pos>", then or by bit. NOTE: FDD(band1 ~ band32, band66 , band252, and band255),

	TDD(band33 ~ band42)
<lte_pos>	Value: 0x48000000000000000000000000000000000000000000000000200000 7FF3FDF3FFF Any (any value) 0 EUTRAN_BAND1(UL:1920-1980; DL:2110-2170) 1 EUTRAN_BAND2(UL:1850-1910; DL:1930-1990) 2 EUTRAN_BAND3(UL:1710-1785; DL:1805-1880) 3 EUTRAN_BAND4(UL:1710-1755; DL:2110-2155) 4 EUTRAN_BAND5(UL: 824-849; DL: 869-894) 5 EUTRAN_BAND6(UL: 830-840; DL: 875-885) 6 EUTRAN_BAND7(UL:2500-2570; DL:2620-2690) 7 EUTRAN_BAND8(UL: 880-915; DL: 925-960) 8 EUTRAN_BAND9(UL:1749.9-1784.9; DL:1844.9-1879.9) 9 EUTRAN_BAND10(UL:1710-1770; DL:2110-2170) 10 EUTRAN_BAND11(UL:1427.9-1452.9; DL:1475.9-1500.9) 11 EUTRAN_BAND12(UL:698-716; DL:728-746) 12 EUTRAN_BAND13(UL: 777-787; DL: 746-756) 13 EUTRAN_BAND14(UL: 788-798; DL: 758-768) 16 EUTRAN_BAND17(UL: 704-716; DL: 734-746) 17 EUTRAN_BAND18(UL: 815-830; DL: 860-875) 18 EUTRAN_BAND19(UL: 830-845; DL: 875-890) 19 EUTRAN_BAND20(UL: 832-862; DL: 791-821) 20 EUTRAN_BAND21(UL: 1447.9-1462.9; DL: 1495.9-1510.9) 22 EUTRAN_BAND23(UL: 2000-2020; DL: 2180-2200) 23 EUTRAN_BAND24(UL: 1626.5-1660.5;

	DL: 1525 -1559) 24 EUTRAN_BAND25(UL: 1850-1915; DL: 1930 -1995) 25 EUTRAN_BAND26(UL: 814-849; DL: 859 -894) 26 EUTRAN_BAND27(UL: 807.5-824; DL: 852 -869) 27 EUTRAN_BAND28(703-748; DL: 758-803) 28 EUTRAN_BAND29(UL:1850-1910 or 1710-1755; DL:716-728) 29 EUTRAN_BAND30(UL: 2305-2315 ; DL: 2350 - 2360) 32 EUTRAN_BAND33(UL: 1900-1920; DL: 1900-1920) 33 EUTRAN_BAND34(UL: 2010-2025; DL: 2010-2025) 34 EUTRAN_BAND35(UL: 1850-1910; DL: 1850-1910) 35 EUTRAN_BAND36(UL: 1930-1990; DL: 1930-1990) 36 EUTRAN_BAND37(UL: 1910-1930; DL: 1910-1930) 37 EUTRAN_BAND38(UL: 2570-2620; DL: 2570-2620) 38 EUTRAN_BAND39(UL: 1880-1920; DL: 1880-1920) 39 EUTRAN_BAND40(UL: 2300-2400; DL: 2300-2400) 40 EUTRAN_BAND41(UL: 2496-2690; DL: 2496-2690) 41 EUTRAN_BAND42(UL: 3400-3600; DL: 3400-3600) 42 EUTRAN_BAND43(UL: 3600-3800; DL: 3600-3800) 65 EUTRAN_BAND66(UL: 1710-1780; DL: 2110-2200) 70 EUTRAN_BAND71(UL: 663-698; DL: 617-652) 251 EUTRAN_BAND252(DL: 5150-5250) 254 EUTRAN_BAND255(DL: 5725-5850)
<tds_mode>	64bit number, the value is “1” << “<tds_pos>”, then or by bit.
<tds_pos>	Value: 0x000000000000003F Any (any value) 0 TDS Band A (1900-1920 MHz, 2010-2020 MHz)

	1	TDS Band B (1850-1910 MHz, 1930-1990 MHz)
	2	TDS Band C (1910-1930 MHz)
	3	TDS Band D (2570-2620 MHz)
	4	TDS Band E (2300-2400 MHz)
	5	TDS Band F (1880-1920 MHz)
<term_mode>	0	term permanent
	1	term until a power cycle

Example

```
AT+CNBP=,0x0000000000000095
OK

AT+CNBP?
+CNBP:
0x0002000004FF0387,0x00000000000000000000000000000000000000000000000000000000000000000000000000095,0x000000000000003F

OK

AT+CNBP=,0x480000000000000000000000000000000000000000000000020000000000000095
OK

AT+CNUM: "","13697252277",129
+ CNBP:
0x0002000004FF0387,0x480000000000000000000000000000000000000000000000020000000000000095,0x000000000000003F

OK
```

4.2.13 AT+CNAOP Acquisitions order preference

This command is used to reset the state of acquisitions order preference.

AT+CNAOP Acquisitions order preference	
Read Command **AT+CNAOP?**	Response **+CNAOP:** **<mode>[,<sys_mode1>,[<sys_mode2>[,<sys_mode3>[,<sys_mode4>[,<sys_mode5>[,<sys_mode6>]]]]]]**

	OK
Write Command **AT+CNAOP=<mode>[,<sys_mode1>[,<sys_mode2>[,<sys_mode3>[,<sys_mode4>[,<sys_mode5>[,<sys_mode6>]]]]]]**	Response **OK** or **ERROR**

Defined Values

<mode>	7 Acquistion by priority order list <sys_mode*n*>s.
<sys_mode>	sys_mode values: 2 CDMA 3 GSM 4 HDR 5 WCDMA 9 LTE 11 TDSCDMA

Example

```
AT+CNAOP=7,9,5,3,11,2,4
OK

AT+CNAOP?
+ CNAOP: 7,9,5,3,11,2,4

OK
```

4.2.14 AT+CPSI Inquiring UE system information

This command is used to return the UE system information.

AT+CPSI Inquiring UE system information	
Test Command **AT+CPSI=?**	Response **+CPSI:** (scope of **<time>**) **OK**
Read Command **AT+CPSI?**	Response If camping on a cdma/evdo cell: **+CPSI: CDMA,<Operation Mode>[,<MCC>-<MNC>,<CDMA ch num>,<CDMA pilot PN>,<CDMA RX Chain 0 AGC>,<CDMA RX**

Chain 1 AGC>,<CDMA Chain 0 LNA>,<CDMA Chain 1 LNA>,<CDMA TX AGC>,<SID>,<NID>,<CDMA EC/IO>,<BID>]
+CPSI: EVDO,<Operation Mode>[,<MCC>-<MNC>,<EVDO ch num>,<EVDO RX Chain 0 AGC>,<EVDO RX Chain 1 AGC>,<EVDO TX AGC>,<EVDO Serving PN>,<EVDO Rel0 SCI>,<EVDO RelA SCI>,<EVDO EC/IO>]

OK

If camping on a gsm cell:

+CPSI:<System Mode>,<Operation Mode>,<MCC>-<MNC>,<LAC>,<Cell ID>,<Absolute RF Ch Num>,<RxLev>,
<Track LO Adjust>,<C1-C2>

OK

If camping on a wcdma cell:

+CPSI: <System Mode>,<Operation Mode>,<MCC>-<MNC>,<LAC>,<Cell ID>,<Frequency Band>,<PSC>,<Freq>,<SSC>,<EC/IO>,<RSCP>,<Qual>,<RxLev>,
<TXPWR>

OK

If camping on a tds-cdma cell:

+CPSI: <System Mode>,<Operation Mode>,<MCC>-<MNC>,<LAC>,<Cell ID>,<Frequency Band>,<Uarfcn>,<Cpid>

OK

If camping on a lte cell:

+CPSI: <System Mode>,<Operation Mode>[,<MCC>-<MNC>,<TAC>,<SCellID>,<PCellID>,<Frequency Band>,<earfcn>,<dlbw>,<ulbw>,<RSRQ>,<RSRP>,<RSSI>,<RSSNR>]

OK

If camping on a cdma/evdo cell:

+CPSI: CDMA,<Operation Mode>[,<MCC>-<MNC>,<CDMA ch num>,<CDMA pilot PN>,<CDMA RX Chain 0 AGC>,<CDMA RX Chain 1 AGC>,<CDMA Chain 0 LNA>,<CDMA Chain 1 LNA>,<CDMA TX AGC>,<SID>,<NID>,<CDMA EC/IO>,<BID>]
+CPSI: EVDO,<Operation Mode>[,<MCC>-<MNC>,<EVDO ch num>,<EVDO RX Chain 0 AGC>,<EVDO RX Chain 1 AGC>,<EVDO TX AGC>,<EVDO Serving PN>,<EVDO Rel0 SCI>,<EVDO RelA SCI>,<EVDO EC/IO>]

OK

	If camping on a cdma/ehrpd cell: **+CPSI: CDMA,<Operation Mode>[,<MCC>-<MNC>,<CDMA ch num>,<CDMA pilot PN>,<CDMA RX Chain 0 AGC>,<CDMA RX Chain 1 AGC>,<CDMA Chain 0 LNA>,<CDMA Chain 1 LNA>,<CDMA TX AGC>,<SID>,<NID>,<CDMA EC/IO>,<BID>]** **+CPSI: eHRPD,<Operation Mode>[,<MCC>-<MNC>,<EVDO ch num>,<EVDO RX Chain 0 AGC>,<EVDO RX Chain 1 AGC>,<EVDO TX AGC>,<EVDO Serving PN>,<EVDO Rel0 SCI>,<EVDO RelA SCI>,<EVDO EC/IO>]** **OK** If camping on 1xlte cell: **+CPSI: CDMA,<Operation Mode>[,<MCC>-<MNC>,<CDMA ch num>,<CDMA pilot PN>,<CDMA RX Chain 0 AGC>,<CDMA RX Chain 1 AGC>,<CDMA Chain 0 LNA>,<CDMA Chain 1 LNA>,<CDMA TX AGC>,<SID>,<NID>,<CDMA EC/IO>,<BID>]** **+CPSI: LTE,<Operation Mode>[,<MCC>-<MNC>,<TAC>,<SCellID>,<PCellID>,<Frequency Band>,<earfcn>,<dlbw>,<ulbw>,<RSRQ>,<RSRP>,<RSSI>,<RSSNR>]** **OK** If no service: **+CPSI: NO SERVICE, Online** **OK** or **ERROR**
Write Command **AT+CPSI=<time>**	Response **OK** or **ERROR**

Defined Values

<time>	The range is 0-255, unit is second, after set <time> will report the system information every the seconds.
<System mode>	System mode, values: “NO SERVICE”, “GSM”, “WCDMA”, “LTE”, “TDS”… If module in LIMITED SERVICE state and +CNLSA command is set to 1, the system mode will display as “GSM-LIMITED”, “WCDMA-LIMITED”…
<Operation mode>	UE operation mode, values: “Unknown”, “Online”, “Offline”, “Factory Test Mode”, “Reset”, “Low Power Mode”.
<MCC>	Mobile Country Code (first part of the PLMN code)

<MNC>	Mobile Network Code (second part of the PLMN code)
<LAC>	Location Area Code (hexadecimal digits)
<Cell ID>	Service-cell Identify.
<Absolute RF Ch Number>	AFRCN for service-cell.
<Track LO Adjust>	Track LO Adjust
<C1>	Coefficient for base station selection
<C2>	Coefficient for Cell re-selection
<Frequency Band>	Frequency Band of active set
<PSC>	Primary synchronization code of active set.
<Freq>	Downlink frequency of active set.
<SSC>	Secondary synchronization code of active set
<EC/IO>	Ec/Io value Received Signal Code Power
<RSCP>	Received Signal Code Power
<Qual>	Quality value for base station selection
<RxLev>	RX level value for base station selection
<TXPWR>	UE TX power in dBm. If no TX, the value is 500.
<Cpi>	Cell Parameter ID
<TAC>	Tracing Area Code
<PCellID>	Physical Cell ID
<earfcn>	E-UTRA absolute radio frequency channel number for searching LTE cells
<dlbw>	Transmission bandwidth configuration of the serving cell on the downlink
<ulbw>	Transmission bandwidth configuration of the serving cell on the uplink
<RSRP>	Current reference signal received power in -1/10 dBm. Available for LTE
<RSRQ>	Current reference signal receive quality as measured by L1.
<RSSNR>	Average reference signal signal-to-noise ratio of the serving cell
<BID>	Base ID

Example

```
AT+CPSI?
+CPSI: GSM,Online,460-00,0x182d,12401,27 EGSM 900,-64,2110,42-42

OK

AT+CPSI?
+CPSI: WCDMA,Online,460-01,0xA809,11122855,WCDMA IMT 2000,279,10663,0,1.5,62,33,
52,500

OK
```

```
AT+CPSI=?
+CPSI: (0-255)

OK
```

4.2.15 AT+CNSMOD Show network system mode

This command is used to return the current network system mode.

AT+CNSMOD Show network system mode	
Test Command **AT+CNSMOD=?**	Response **+CNSMOD: (**list of supported **<n>s)** **OK**
Read Command **AT+CNSMOD=?**	Response **+CNSMOD: <n>,<stat>** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Write Command **AT+CNSMOD=<n>**	Response **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Defined Values

<n>	0 disable auto report the network system mode information 1 auto report the network system mode information, command: +CNSMOD:<stat>
<stat>	0 no service 1 GSM 2 GPRS 3 EGPRS (EDGE) 4 WCDMA 5 HSDPA only(WCDMA) 6 HSUPA only(WCDMA) 7 HSPA (HSDPA and HSUPA, WCDMA)

	8 LTE 9 TDS-CDMA 10 TDS-HSDPA only 11 TDS- HSUPA only 12 TDS- HSPA (HSDPA and HSUPA) 13 CDMA 14 EVDO 15 HYBRID (CDMA and EVDO) 16 1XLTE(CDMA and LTE) 23 eHRPD 24 HYBRID(CDMA and eHRPD)
<type>	Type of address octet in integer format.see also AT+CPBR <type>

Example

```
AT+CNSMOD?
+CNSMOD: 0,2

OK
```

4.2.16 AT+CEREG EPS network registration status

The set command controls the presentation of an unsolicited result code +CEREG: <stat> when <n>=1 and there is a change in the MT's EPS network registration status in E-UTRAN, or unsolicited result code +CEREG: <stat>[,<tac>,<ci>[,<AcT>]] when <n>=2 and there is a change of the network cell in E-UTRAN; in this latest case <AcT>, <tac> and <ci> are sent only if available.

NOTE 1: If the EPS MT in GERAN/UTRAN/E-UTRAN also supports circuit mode services and/or GPRS services, the +CREG command and +CREG: result codes and/or the +CGREG command and +CGREG: result codes apply to the registration status and location information for those services.

The read command returns the status of result code presentation and an integer <stat> which shows whether the network has currently indicated the registration of the MT. Location information elements <tac>, <ci> and <AcT>, if available, are returned only when <n>=2 and MT is registered in the network.

AT+CEREG EPS network registration status	
Test Command **AT+CEREG=?**	Response **+CEREG: (**list of supported **<n>s)** **OK** or **ERROR**
Read Command **AT+CEREG?**	Response **+CEREG: <n>,<stat>[,<tac>,<ci>[,<AcT>]]**

	OK or **ERROR**
Write Command **AT+CEREG[=<n>]**	Response **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Execution Command **AT+CEREG**	Response (Set default value(<n>=0) **OK** or **ERROR**

Defined Values

<n>	0 disable network registration unsolicited result code 1 enable network registration unsolicited result code +CEREG: <stat> 2 enable network registration and location information unsolicited result code +CEREG: <stat>[,<tac>,<ci>[,<AcT>]]
<stat>	0 not registered, MT is not currently searching an operator to register to 1 registered, home network 2 not registered, but MT is currently trying to attach or searching an operator to register to 3 registration denied 4 unknown (e.g. out of E-UTRAN coverage) 5 registered, roaming 6 registered for "SMS only", home network (not applicable) 7 registered for "SMS only", roaming (not applicable) 8 attached for emergency bearer services only (See NOTE 2)
<tac>	string type; two byte tracking area code in hexadecimal format (e.g. "00C3" equals 195 in decimal)
<ci>	string type; four byte E-UTRAN cell identify in hexadecimal format
<AcT>	A numberic parameter that indicates the access technology of serving cell 0 GSM (not applicable) 1 GSM Compact (not applicable) 2 UTRAN (not applicable) 3 GSM w/EGPRS (see NOTE 3) (not applicable) 4 UTRAN w/HSDPA (see NOTE 4) (not applicable) 5 UTRAN w/HSUPA (see NOTE 4) (not applicable) 6 UTRAN w/HSDPA and HSUPA (see NOTE 4) (not applicable)

	7 E-UTRAN

Example

```
AT+CEREG?
+CEREG: 0,4

OK
```

NOTE

- If the EPS MT in GERAN/UTRAN/E-UTRAN also supports circuit mode services and/or GPRS services, the +CREG command and +CREG: result codes and/or the +CGREG command and +CGREG: result codes apply to the registration status and location information for those services.

4.2.17 AT+CTZU Automatic time and time zone update

This command is used to enable and disable automatic time and time zone update via NITZ.

AT+CTZU Automatic time and time zone update	
Test Command **AT+CTZU=?**	Response **+ CTZU: (**list of supported **<on/off>s)** **OK**
Execution Command **AT+CTZU?**	Response **+CTZU: <on/off >** **OK** or If error is related to ME functionality: **+CME ERROR: <err>**
Write Command **AT+CTZU=<on/off>**	Response **OK** or **ERROR**

Defined Values

<on/off>	Integer type value indicating:

	0 Disable automatic time zone update via NITZ (default). 1 Enable automatic time zone update via NITZ. NOTE: 1. The value of < on/off > is nonvolatile, and factory value is 0. 2. For automatic time and time zone update is enabled (+CTZU=1): If time zone is only received from network and it isn't equal to local time zone (AT+CCLK), time zone is updated automatically, and real time clock is updated based on local time and the difference between time zone from network and local time zone (Local time zone must be valid). If Universal Time and time zone are received from network, both time zone and real time clock is updated automatically, and real time clock is based on Universal Time and time zone from network.
<number>	String type phone number of format specified by <type>.
<type>	Type of address octet in integer format.see also AT+CPBR <type>

Example

```
AT+CTZU?
+ CTZU: 0

OK

AT+CTZU=1
OK
```

4.2.18 AT+CTZR Time and time zone reporting

This command is used to enable and disable the time zone change event reporting. If the reporting is enabled the MT returns the unsolicited result code +CTZV: <tz>[,<time>][,<dst>]whenever the time zone is changed.

AT+CTZR Time and time zone reporting	
Test Command **AT+CTZR=?**	Response **+CTZR: (**list of supported **<on/off>s)** **OK**
Read Command **AT+CTZR**	Response **+CTZR: <on/off>** **OK**
Write Command **AT+CTZR=<on/off>**	Response **OK**

	or **ERROR**
Execution Command **AT+CTZR**	Response (Set default value) **OK**

Defined Values

<on/off>	Integer type value indicating: 0 Disable time zone change event reporting (default). 1 Enable time zone change event reporting.
+CTZV: <tz>[,<time>][,<dst>]	Unsolicited result code when time zone received from network isn't equal to local time zone, and if the informations from network don't include date and time, time zone will be only reported, and if network daylight saving time is present, it is also reported. For example: +CTZV: 32 (Only report time zone) +CTZV: 32,1 (Report time zone and network daylight saving time) +CTZV: 32,08/12/09,17:00:00 (Report time and time zone) +CTZV: 32,08/12/09,17:00:00,1 (Report time, time zone and daylight saving time) For more detailed informations about time and time zone, please refer 3GPP TS 24.008. <tz> Local time zone received from network. <time> Universal time received from network, and the format is "yy/MM/dd,hh:mm:ss", where characters indicate year (two last digits), month, day, hour, minutes and seconds. <dst> Network daylight saving time, and if it is received from network, it indicates the value that has been used to adjust the local time zone. The values as following: 0 – No adjustment for Daylight Saving Time. 1 – +1 hour adjustment for Daylight Saving Time. 2 – +2 hours adjustment for Daylight Saving Time. NOTE: Herein, <time> is Universal Time or NITZ time, but not local time.

Example

```
AT+CTZR?
+CTZR: 0

OK

AT+CTZR=1
OK
```

NOTE

- The time zone reporting is not affected by the Automatic Time and Time Zone command AT+CTZU.

5. AT Commands for Call Control

5.1 Overview of AT Commands for Call Control

Command	Description
AT+CVHU	Voice hang up control
AT+CHUP	Hang up call
AT+CBST	Select bearer service type
AT+CRLP	Radio link protocol
AT+CR	Service reporting control
AT+CRC	Cellular result codes
AT+CLCC	List current calls
AT+CEER	Extended error report
AT+CCWA	Call waiting
AT+CHLD	Call related supplementary services
AT+CCFC	Call forwarding number and conditions
AT+CLIP	Calling line identification presentation
AT+CLIR	Calling line identification restriction
AT+COLP	Connected line identification presentation
AT+VTS	DTMF and tone generation
AT+VTD	Tone duration
AT+CSTA	Select type of address
AT+CMOD	Call mode

5.2 Detailed Description of AT Commands for Call Control

5.2.1 AT+CVHU Voice hang up control

Write command selects whether ATH or “drop DTR” shall cause a voice connection to be disconnected or not. By voice connection is also meant alternating mode calls that are currently in voice mode.

AT+CVHU Voice hang up control	
Test Command **AT+CVHU=?**	Response **+CVHU: (list of supported <mode>s)** **OK**
Read Command **AT+CVHU?**	Response **+CVHU: <mode>** **OK**
Write Command **AT+CVHU=<mode>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Execution Command **AT+CVHU**	Response **OK**
Maximum Response Time	120000ms

Defined Values

<mode>	0 – "Drop DTR" ignored but OK response given. ATH disconnects. 1 – "Drop DTR" and ATH ignored but OK response given.

Examples

```
AT+CVHU=0
OK

AT+CVHU?
+CVHU: 0
OK
```

5.2.2 AT+CHUP Hang up call

This command is used to cancel voice calls. If there is no call, it will do nothing but OK response is given. After running AT+CHUP, multiple "VOICE CALL END: " may be reported which relies on how many calls exist before calling this command.

AT+CHUP Hang up call	
Test Command **AT+CHUP=?**	Response **OK**

Execution Command **AT+CHUP**	Response **VOICE CALL: END: <time>** **[…** **VOICE CALL: END: <time>]** **OK** *No call:* **OK**
Maximum Response Time	120000ms

Defined Values

<time>	Voice call connection time. Format – HHMMSS (HH: hour, MM: minute, SS: second)

Examples

```
AT+CHUP
VOICE CALL:END: 000017
OK
```

5.2.3 AT+CBST Select bearer service type

Write command selects the bearer service <name> with data rate <speed>, and the connection element <ce> to be used when data calls are originated. Values may also be used during mobile terminated data call setup, especially in case of single numbering scheme calls.

AT+CBST Select bearer service type	
Test Command **AT+CBST=?**	Response **+CBST: (list of supported <speed>s), (list of supported <name>s), (list of supported <ce>s)** **OK**
Read Command **AT+CBST?**	Response **+CBST: <speed>,<name>,<ce>** **OK**
Write Command **AT+CBST=<speed>[,<name>[,<ce>]]**	Response a)If successfully: **OK** b)If failed: **ERROR**
Execution Command **AT+CBST**	Response **OK**

Maximum Response Time	120000ms

Defined Values

<speed>	0 – autobauding(automatic selection of the speed; this setting is possible in case of 3.1 kHz modem and non-transparent service) 7 – 9600 bps (V.32) 12 – 9600 bps (V.34) 14 – 14400 bps(V.34) 16 – 28800 bps(V.34) 17 – 33600 bps(V.34) 39 – 9600 bps(V.120) 43 – 14400 bps(V.120) 48 – 28800 bps(V.120) 51 – 56000 bps(V.120) 71 – 9600 bps(V.110) 75 – 14400 bps(V.110) 80 – 28800 bps(V.110 or X.31 flag stuffing) 81 – 38400 bps(V.110 or X.31 flag stuffing) 83 – 56000 bps(V.110 or X.31 flag stuffing) 84 – 64000 bps(X.31 flag stuffing) 116 – 64000 bps(bit transparent) 134 – 64000 bps(multimedia)
<name>	0 – Asynchronous modem 1 – Synchronous modem 4 – data circuit asynchronous (RDI)
<ce>	0 – transparent 1 – non-transparent
NOTE: If <speed> is set to 116 or 134, it is necessary that <name> is equal to 1 and <ce> is equal to 0.	

Examples

```
AT+CBST=0,0,1
OK

AT+CBST?
+CBST: 0,0,1
OK
```

5.2.4 AT+CRLP Radio link protocol

Radio Link Protocol(RLP) parameters used when non-transparent data calls are originated may be altered with write command.
Read command returns current settings for each supported RLP version <verX>. Only RLP parameters applicable to the corresponding <verX> are returned.
Test command returns values supported by the TA as a compound value. If ME/TA supports several RLP versions <verX>, the RLP parameter value ranges for each <verX> are returned in a separate line.

AT+CRLP Radio link protocol	
Test Command **AT+CRLP=?**	Response **+CRLP: (list of supported <iws>s), (list of supported <mws>s), (list of supported <T1>s), (list of supported <N2>s) [,<ver1> [,(list of supported <T4>s)]][<CR><LF>** **+CRLP: (list of supported <iws>s), (list of supported <mws>s), (list of supported <T1>s), (list of supported <N2>s) [,<ver2> [,(list of supported <T4>s)]]** **[...]]** **OK**
Read Command **AT+CRLP?**	Response **+CRLP: <iws>, <mws>, <T1>, <N2> [,<ver1> [, <T4>]][<CR><LF>** **+CRLP:<iws>,<mws>,<T1>,<N2>[,<ver2>[,<T4>]]** **[...]]** **OK**
Write Command **AT+CRLP=<iws> [,<mws>[,<T1>[,<N2> [,<ver>[,<T4>]]]]]**	Response a)If successfully: **OK** b)If failed: **ERROR**
Execution Command **AT+CRLP**	Response **OK**
Maximum Response Time	120000ms

Defined Values

<ver>,<verX>	RLP version number in integer format, and it can be 0, 1 or 2; when version indication is not present it shall equal 1.
<iws>	IWF to MS window size.
<mws>	MS to IWF window size.
<T1>	Acknowledgement timer.
<N2>	Retransmission attempts.
<T4>	Re-sequencing period in integer format.
NOTE: <T1> and <T4> are in units of 10 ms.	

Examples

```
AT+CRLP=0
+CRLP:61,61,48,6,0
+CRLP:61,61,48,6,1
+CRLP:240,240,52,6,2
OKs
```

5.2.5 AT+CR Service reporting control

Write command controls whether or not intermediate result code "+CR: <serv>" is returned from the TA to the TE. If enabled, the intermediate result code is transmitted at the point during connect negotiation at which the TA has determined which speed and quality of service will be used, before any error control or data compression reports are transmitted, and before the intermediate result code CONNECT is transmitted.

AT+CR Service reporting control	
Test Command **AT+CR=?**	Response **+CR: (list of supported <mode>s)** **OK**
Read Command **AT+CR?**	Response **+CR: <mode>** **OK**
Write Command **AT+CR=<mode>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Execution Command **AT+CR**	Response **OK**
Maximum Response Time	120000ms

Defined Values

<mode>	0 – disables reporting 1 – enables reporting
<serv>	ASYNC asynchronous transparent SYNC synchronous transparent REL ASYNC asynchronous non-transparent REL sync synchronous non-transparent GPRS [<L2P>] GPRS The optional <L2P> proposes a layer 2 protocol to use between the MT and the TE.s

Examples

```
AT+CR=1
OK

AT+CR?
+CR: 1
OK
```

5.2.6 AT+CRC Cellular result codes

Write command controls whether or not the extended format of incoming call indication or GPRS network request for PDP context activation is used. When enabled, an incoming call is indicated to the TE with unsolicited result code "+CRING: <type>" instead of the normal RING.
Test command returns values supported by the TA as a compound value.

AT+CRC Cellular result codes	
Test Command **AT+CRC=?**	Response **+CRC: (list of supported <mode>s)** **OK**
Read Command **AT+CRC?**	Response **+CRC: <mode>** **OK**
Write Command **AT+CRC=<mode>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Execution Command **AT+CRC**	Response **OK**
Maximum Response Time	120000ms

Defined Values

<mode>	0 – disable extended format 1 – enable extended format	
<type>	ASYNC	asynchronous transparent
	SYNC	synchronous transparent
	REL ASYNC	asynchronous non-transparent
	REL SYNC	synchronous non-transparent
	FAX	facsimile
	VOICE	normal voice

	VOICE/XXX voice followed by data(XXX is ASYNC, SYNC, REL ASYNC or REL SYNC) ALT VOICE/XXX alternating voice/data, voice first ALT XXX/VOICE alternating voice/data, data first ALT FAX/VOICE alternating voice/fax, fax first GPRS GPRS network request for PDP context activation

Examples

```
AT+CRC=1
OK

AT+CRC?
+CRC: 1
OK
```

5.2.7 AT+CLCC List current calls

This command issued to return list of current calls of ME. If command succeeds but no calls are available, no information response is sent to TE.

AT+CLCC List current calls	
Test Command **AT+CLCC=?**	Response **+CLCC: (list of supported <n>s)** **OK**
Read Command **AT+CLCC?**	Response **+CLCC: <n>** **OK**
Write Command **AT+CLCC=<n>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Execution Command **AT+CLCC**	Response **+CLCC:<id1>,<dir>,<stat>,<mode>,<mpty>[,<number>,<type>[,<alpha>]][<CR><LF>** **+CLCC:<id2>,<dir>,<stat>,<mode>,<mpty>[,<number>,<type>[,<alpha>]]** **[...]]** **OK**
Maximum Response Time	120000ms

Defined Values

<n>	0 – Don't report a list of current calls of ME automatically when the current call status changes. 1 – Report a list of current calls of ME automatically when the current call status changes.
<idX>	Integer type, call identification number, this number can be used in +CHLD command operations.
<dir>	0 – mobile originated (MO) call 1 – mobile terminated (MT) call
<stat>	State of the call: 0 – active 1 – held 2 – dialing (MO call) 3 – alerting (MO call) 4 – incoming (MT call) 5 – waiting (MT call) 6 – disconnect
<mode>	bearer/teleservice: 0 – voice 1 – data 2 – fax 9 – unknown
<mpty>	0 – call is not one of multiparty (conference) call parties 1 – call is one of multiparty (conference) call parties
<number>	String type phone number in format specified by <type>.
<type>	Type of address octet in integer format; 128 – Restricted number type includes unknown type and format 145 – International number type 161 – national number.The network support for this type is optional 177 – network specific number,ISDN format 129 – Otherwise
<alpha>	String type alphanumeric representation of <number> corresponding to the entry found in phonebook; used character set should be the one selected with command Select TE Character Set AT+CSCS.

Examples

```
ATD10011;
OK
```

```
AT+CLCC
+CLCC: 1,0,0,0,0,"10011",129,"sm"
OK

RING (with incoming call)

AT+CLCC
+CLCC: 1,1,4,0,0,"02152063113",128,"gongsi"
OK
```

5.2.8 AT+CEER Extended error report

Execution command causes the TA to return the information text <report>, which should offer the user of the TA an extended report of the reason for:

1 The failure in the last unsuccessful call setup(originating or answering) or in-call modification.
2 The last call release.
3 The last unsuccessful GPRS attach or unsuccessful PDP context activation.
The last GPRS detach or PDP context deactivation.

AT+CEER Extended error report	
Test Command **AT+CEER=?**	Response **OK**
Execution Command **AT+CEER**	Response **+CEER:<report>** **OK**
Maximum Response Time	120000ms

Defined Values

<report>	Wrong information which is possibly occurred.

Examples

```
AT+CEER
+CEER: Invalid/incomplete number
OK
```

5.2.9 AT+CCWA Call waiting

This command allows control of the Call Waiting supplementary service. Activation, deactivation and status query are supported. When querying the status of a network service (<mode>=2) the response line for 'not active' case (<status>=0) should be returned only if service is not active for any <class>. Parameter <n> is used to disable/enable the presentation of an unsolicited result code +CCWA: <number>,<type>,<class> to the TE when call waiting service is enabled. Command should be abortable when network is interrogated.

AT+CCWA Call waiting	
Test Command **AT+CCWA=?**	Response **+CCWA: (list of supported <n>s)** **OK**
Read Command **AT+CCWA?**	Response **+CCWA: <n>** **OK**
Write Command **AT+CCWA=<n>[,<mode>[,<class>]]**	Response a)If successfully: **When <mode>=2 and command successful:** **+CCWA:<status>,<class>[<CR><LF>** **+CCWA: <status>, <class>[...]]** **OK** b)If failed: **ERROR**
Execution Command **AT+CCWA**	Response **OK**
Maximum Response Time	120000ms

Defined Values

<n>	Sets/shows the result code presentation status in the TA 0 – disable 1 – enable
<mode>	When <mode> parameter is not given, network is not interrogated: 0 – disable 1 – enable 2 – query status
<class>	It is a sum of integers each representing a class of information (default 7) 1 – voice (telephony) 2 – data (refers to all bearer services) 4 – fax (facsimile services) 7 – voice,data and fax(1+2+4) 8 – short message service 16 – data circuit sync 32 – data circuit async 64 – dedicated packet access 128 – dedicated PAD access

	255 – The value 255 covers all classes
<status>	0 – not active 1 – active
<number>	String type phone number of calling address in format specified by <type>.
<type>	Type of address octet in integer format; 128 – Restricted number type includes unknown type and format 145 – International number type 129 – Otherwise

Examples

```
AT+CCWA=?
+CCWA:(0-1)
OK

AT+CCWA?
+CCWA: 0
OK
```

5.2.10 AT+CHLD Call related supplementary services

This command allows the control the following call related services:
1. A call can be temporarily disconnected from the ME but the connection is retained by the network.
2. Multiparty conversation (conference calls).
3. The served subscriber who has two calls (one held and the other either active or alerting) can connect the other parties and release the served subscriber's own connection.
Calls can be put on hold, recovered, released, added to conversation, and transferred. This is based on the GSM/UMTS supplementary services.

AT+CHLD Call related supplementary services	
Test Command **AT+CHLD=?**	Response **OK**
Write Command **AT+CHLD=<n>**	Response **OK** or **ERROR**
Execution Command **AT+CHLD** **Default to <n>=2.**	Response **OK** or **ERROR**

	or **+CME ERROR: <err>**
Parameter Saving Mode	NO_SAVE
Maximum Response Time	-
Reference	

Defined Values

<n>	0 – Terminate all held calls; or set User Determined User Busy for a waiting call 1 – Terminate all active calls and accept the other call (waiting call or held call) 1X – Terminate a specific call X 2 – Place all active calls on hold and accept the other call (waiting call or held call) as the active call 2X – Place all active calls except call X on hold 3 – Add the held call to the active calls 4 – Connect two calls and cut off the connection between users and them simultaneously

Example

```
AT+CHLD=?
+CHLD: (0,1,1x,2,2x,3,4)
OK
```

5.2.11 AT+CCFC Call forwarding number and conditions

This command allows control of the call forwarding supplementary service. Registration, erasure, activation, deactivation, and status query are supported.

AT+CCFC Call forwarding number and conditions	
Test Command **AT+CCFC=?**	Response **+CCFC: (list of supported <reason>s)** **OK**
Write Command **AT+CCFC=<reason>,<mode>[,<number>[,<type>[,<class>[,<subaddr>[,<satype>[,<time>]]]]]]**	Response When <mode>=2 and command successful: **+CCFC: <status>,<class1>[,<number>,<type>[,<subaddr>,<satype>[,<time>]]][<CR><LF>** **+CCFC: <status>,<class2>[,<number>,<type>[,<subaddr>,<satype>[,<time>]]][...]]** **OK**

	When <mode>!=2 and command successful: **OK** or **ERROR** or **+CME ERROR:<err>**
Parameter Saving Mode	NO_SAVE
Maximum Response Time	-
Reference	

Defined Values

<reason>	0 – unconditional 1 – mobile busy 2 – no reply 3 – not reachable 4 – all call forwarding 5 – all conditional call forwarding
<mode>	0 – disable 1 – enable 2 – query status 3 – registration 4 – erasure
<number>	String type phone number of forwarding address in format specified by <type>.
<type>	Type of address octet in integer format: 145 – dialing string <number> includes international access code character '+' 129 – otherwise
<subaddr>	String type sub address of format specified by <satype>. Subaddr length is 0-19.
<satype>	Type of sub address octet in integer format, default 128.
<classX>	It is a sum of integers each representing a class of information (default 7): 1 – voice (telephony) 2 – data (refers to all bearer services) 4 – fax (facsimile services) 16 – data circuit sync 32 – data circuit async 64 – dedicated packet access 128 – dedicated PAD access 255 – The value 255 covers all classes
<time>	1...30 – when "no reply" is enabled or queried, this gives the time in seconds to wait before call is forwarded, default value 20.

<status>	0 – not active 1 – active

Example

```
AT+CCFC=?
+CCFC: (0,1,2,3,4,5)
OK
AT+CCFC=0,2
+CCFC: 0,255
OK
```

5.2.12 AT+CLIP Calling line identification presentation

This command refers to the GSM/UMTS supplementary service CLIP (Calling Line Identification Presentation) that enables a called subscriber to get the calling line identity (CLI) of the calling party when receiving a mobile terminated call.
Write command enables or disables the presentation of the CLI at the TE. It has no effect on the execution of the supplementary service CLIP in the network.
When the presentation of the CLI at the TE is enabled (and calling subscriber allows), +CLIP: <number>,<type>,,[,[<alpha>][,<CLI validity>]] response is returned after every RING (or +CRING: <type>; refer sub clause "Cellular result codes +CRC") result code sent from TA to TE. It is manufacturer specific if this response is used when normal voice call is answered.

AT+CLIP Calling line identification presentation	
Test Command **AT+CLIP=?**	Response **+CLIP: (list of supported <n>s)** **OK**
Read Command **AT+CLIP?**	Response **+CLIP: <n>,<m>** **OK** or **ERROR** or **+CME ERROR: <err>**
Write Command **AT+CLIP=<n>**	Response **OK** or **ERROR**
Execution Command **AT+CLIP**	Response Set default value(<n>=0): **OK**

Parameter Saving Mode	NO_SAVE
Maximum Response Time	-
Reference	

Defined Values

<n>	Parameter sets/shows the result code presentation status in the TA: 0 – disable 1 – enable
<m>	0 – CLIP not provisioned 1 – CLIP provisioned 2 – unknown (e.g. no network, etc.)
<number>	String type phone number of calling address in format specified by <type>
<type>	Type of address octet in integer format; 128 – Restricted number type includes unknown type and format 145 – International number type 161 – national number.The network support for this type is optional 177 – network specific number,ISDN format 129 – Otherwise
<alpha>	String type alphanumeric representation of <number> corresponding to the entry found in phone book.
<CLI validity>	0 – CLI valid 1 – CLI has been withheld by the originator 2 – CLI is not available due to interworking problems or limitations of originating network

Example

```
AT+CLIP=1
OK
RING (with incoming call)
+CLIP: "02152063113",128,,,"gongsi",0
```

5.2.13 AT+CLIR Calling line identification restriction

This command refers to CLIR service that allows a calling subscriber to enable or disable the presentation of the CLI to the called party when originating a call.
Write command overrides the CLIR subscription (default is restricted or allowed) when temporary mode is provisioned as a default adjustment for all following outgoing calls. This adjustment can be revoked by using the opposite command.. If this command is used by a subscriber without provision of CLIR in permanent mode the network will act.

Read command gives the default adjustment for all outgoing calls (given in <n>), and also triggers an interrogation of the provision status of the CLIR service (given in <m>).
Test command returns values supported as a compound value.

AT+CLIR Calling line identification restriction	
Test Command **AT+CLIR=?**	Response **+CLIR: (list of supported <n>s)** **OK**
Read Command **AT+CLIR?**	Response **+CLIR <n>,<m>** **OK** or **ERROR** or **+CME ERROR: <err>**
Write Command **AT+CLIR=<n>**	Response **OK** or **ERROR** or **+CME ERROR: <err>**
Parameter Saving Mode	NO_SAVE
Maximum Response Time	-
Reference	

Defined Values

<n>	0 – presentation indicator is used according to the subscription of the CLIR service 1 – CLIR invocation 2 – CLIR suppression
<m>	0 – CLIR not provisioned 1 – CLIR provisioned in permanent mode 2 – unknown (e.g. no network, etc.) 3 – CLIR temporary mode presentation restricted 4 – CLIR temporary mode presentation allowed

Example

```
AT+CLIR=?
+CLIR:(0-2)
OK
```

5.2.14 AT+COLP Connected line identification presentation

This command refers to the GSM/UMTS supplementary service COLP(Connected Line Identification Presentation) that enables a calling subscriber to get the connected line identity (COL) of the called party after setting up a mobile originated call. The command enables or disables the presentation of the COL at the TE. It has no effect on the execution of the supplementary service COLR in the network.
When enabled (and called subscriber allows), +COLP:<number>, <type> [,<subaddr>, <satype> [,<alpha>]] intermediate result code is returned from TA to TE before any +CR responses. It is manufacturer specific if this response is used when normal voice call is established.
When the AT+COLP=1 is set, any data input immediately after the launching of “ATDXXX;” will stop the execution of the ATD command, which may cancel the establishing of the call.

AT+COLP Connected line identification presentation	
Test Command **AT+COLP=?**	Response **+COLP: (list of supported <n>s)** **OK**
Read Command **AT+COLP?**	Response **+COLP: <n>,<m>** **OK** **or** **ERROR** or **+CME ERROR: <err>**
Write Command **AT+COLP =<n>**	Response **OK** or **ERROR** or **+CME ERROR: <err>**
Execution Command **AT+COLP**	Response Set default value(<n>=0, <m>=0): **OK**
Parameter Saving Mode	NO_SAVE
Maximum Response Time	-
Reference	

Defined Values

<n>	Parameter sets/shows the result code presentation status in the TA: 0 – disable 1 – enable
<m>	0 – COLP not provisioned 1 – COLP provisioned 2 – unknown (e.g. no network, etc.)

Example

```
AT+COLP?
+COLP: 1,0
OK
ATD10086;
VOICE CALL: BEGIN

+COLP: "10086",129,,,

OK
```

5.2.15 AT+VTS DTMF and tone generation

This command allows the transmission of DTMF tones and arbitrary tones which cause the Mobile Switching Center (MSC) to transmit tones to a remote subscriber. The command can only be used in voice mode of operation (active voice call).

NOTE: The END event of voice call will terminate the transmission of tones, and as an operator option, the tone may be ceased after a pre-determined time whether or not tone duration has been reached.

AT+VTS DTMF and tone generation	
Test Command **AT+VTS=?**	Response **+VTS: (list of supported<dtmf>s)** **OK**
Write Command **AT+VTS=<dtmf>** **[,<duration>]** **AT+VTS=<dtmf-string>**	Response **OK** or **ERROR**
Parameter Saving Mode	NO_SAVE
Maximum Response Time	-
Reference	

Defined Values

<dtmf>	A single ASCII character in the set 0-9, *, #, A, B, C, D.
<duration>	Tone duration in 1/10 seconds, from 0 to 255. This is interpreted as a DTMF tone of different duration from that mandated by the AT+VTD command, otherwise, the duration which be set the AT+VTD command will be used for the tone (<duration> is omitted).
<dtmf-string>	A sequence of ASCII character in the set 0-9, *, #, A, B, C, D, and

	maximal length of the string is 29. The string must be enclosed in double quotes (“”), and separated by commas between the ASCII characters (e.g. “1,3,5,7,9,*”). Each of the tones with a duration which is set by the AT+VTD command.

NOTE

- The value of <mode> shall be set to zero after a successfully completed alternating mode call. It shall be set to zero also after a failed answering. The power-on, factory and user resets shall also set the value to zero. This reduces the possibility that alternating mode calls are originated or answered accidentally.

Example

```
AT+VTS=1
OK
AT+VTS=1,20
OK
AT+VTS=”1,3,5”
OK
AT+VTS=?
+VTS: (0-9,*,#,A,B,C,D)
OK
```

5.2.16 AT+VTD Tone duration

This refers to an integer <n> that defines the length of tones emitted as a result of the AT+VTS command. A value different than zero causes a tone of duration <n>/10 seconds.

AT+VTD Tone duration	
Test Command **AT+VTD=?**	Response **+VTD: (list of supported <n>s)** **OK**
Read Command **AT+VTD?**	Response **+VTD: <n>** **OK**
Write Command **AT+VTD=<n>**	Response **OK**
Parameter Saving Mode	NO_SAVE
Maximum Response Time	-

Reference	

Defined Values

<n>	Tone duration in integer format, from 0 to 255, and 0 is factory value. 0 Tone duration of every single tone is dependent on the network. 1…255 one duration of every single tone in 1/10 seconds.

Example

```
AT+VTD=?
+VTD: (0-255)
OK
AT+VTD?
+VTD: 0
OK
AT+VTD=5
OK
```

5.2.17 AT+CSTA Select type of address

Write command is used to select the type of number for further dialing commands (ATD) according to GSM/UMTS specifications.
Read command returns the current type of number.
Test command returns values supported by the Module as a compound value.

AT+CSTA Select type of address	
Test Command **AT+CSTA=?**	Response **+CSTA:(list of supported <type>s)** **OK**
Read Command **AT+CSTA?**	Response **+CSTA:<type>** **OK**
Write Command **AT+CSTA=<type>**	Response **OK** or **ERROR**
Execution Command **AT+CSTA**	Response **OK**
Parameter Saving Mode	NO_SAVE
Maximum Response Time	-

Reference	

Defined Values

<type>	Type of address octet in integer format: 145 – when dialling string includes international access code character "+" 161 – national number.The network support for this type is optional 177 – network specific number,ISDN format 129 – otherwise

NOTE

- Because the type of address is automatically detected on the dial string of dialing command, command AT+CSTA has really no effect.

Example

```
AT+CSTA?
+CSTA: 129
OK
AT+CSTA=145
OK
```

5.2.18 AT+CMOD Call mode

Write command is used to select the type of number for further dialing commands (ATD) according to GSM/UMTS specifications.
Read command returns the current type of number.
Test command returns values supported by the Module as a compound value.

AT+CMOD Call mode	
Test Command **AT+CMOD=?**	Response **+CMOD: (list of supported <mode>s)** **OK**
Read Command **AT+CMOD?**	Response **+CMOD: <mode>** **OK**
Write Command **AT+CMOD=<mode>**	Response **OK** or

	ERROR
Execution Command **AT+CMOD**	Response **Set default value:** **OK**
Parameter Saving Mode	NO_SAVE
Maximum Response Time	-
Reference	

Defined Values

<mode>	0 – single mode(only supported)

NOTE

- **NOTE:** The value of <mode> shall be set to zero after a successfully completed alternating mode call. It shall be set to zero also after a failed answering. The power-on, factory and user resets shall also set the value to zero. This reduces the possibility that alternating mode calls are originated or answered accidentally.

Example

```
AT+CMOD?
+CMOD: 0
OK
AT+CMOD=0
OK
```

6. AT Commands for Phonebook

6.1 Overview of AT Commands for Phonebook

Command	Description
AT+CPBS	Select Phonebook memory storage
AT+CPBR	Read Phonebook entries
AT+CPBF	Find Phonebook entries
AT+CPBW	Write Phonebook entry
AT+CNUM	Subscriber number

6.2 Detailed Description of AT Commands for Phonebook

6.2.1 AT+CPBS Select Phonebook memory storage

AT+CPBS Select Phonebook memory entries	
Test Command **AT+CPBS=?**	Response **+CPBS:** (list of supported <**storage**>s) **OK**
Read Command **AT+CPBS?**	Response **+CPBS: <storage>[,<used>,<total>]** **OK**
Write Command **AT+CPBS=<storage>**	Response **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Execution Command **AT+CPBS**	Response (Set default value “SM”) **OK**

Defined Values

<storage>	Values reserved by the present document: "DC" ME dialed calls list Capacity: max. 100 entries AT+CPBW command is not applicable to this storage. "MC" ME missed (unanswered received) calls list Capacity: max. 100 entries AT+CPBW command is not applicable to this storage. "RC" ME received calls list Capacity: max. 100 entries AT+CPBW command is not applicable to this storage. "SM" SIM phonebook Capacity: depending on SIM card "ME" Mobile Equipment phonebook Capacity: max. 500 entries "FD" SIM fixdialling-phonebook Capacity:depending on SIM card "ON" MSISDN list Capacity:depending on SIM card "LD" Last number dialed phonebook Capacity: depending on SIM card AT+CPBW command is not applicable to this storage "EN" Emergency numbers Capacity: depending on SIM card AT+CPBW command is not applicable to this storage.
<used>	Integer type value indicating the number of used locations in selected memory.
<total>	Integer type value indicating the total number of locations in selected memory.

Example

```
AT+CPBS=?
+CPBS:
("SM","DC","FD","LD","MC","ME","RC","EN"
,"ON")

OK
AT+CPBS=”SM”
OK

AT+CPBS?
+CPBS: "SM",1,200
```

OK

NOTE

- Select the active phonebook storage,i.e.the phonebook storage that all subsequent phonebook commands will be operating on

6.2.2 AT+CPBR Read Phonebook entries

AT+CPBR Read Phonebook entries	
Test Command **AT+CPBR=?**	Response **+CPBR: (<minIndex>-<maxIndex>), [<nlength>], [<tlength>]** **OK** or If error is related to ME functionality: **+CME ERROR: <err>**
Write Command **AT+CPBR=<index1>[,<index2>]**	Response **[+CPBR: <index1>,<number>,<type>,<text>[<CR><LF>** **+CPBR: <index2>,<number>,<type>,<text>[…]]]** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Defined Values

<index1>	Integer type value in the range of location numbers of phonebook memory.
<index2>	Integer type value in the range of location numbers of phonebook memory.
<index>	Integer type. the current position number of the Phonebook index.
<minIndex>	Integer type the minimum <index> number.
<maxIndex>	Integer type the maximum <index> number.
<number>	String type, phone number of format <type>, the maximum length is <nlength>.
<type>	Type of phone number octet in integer format, default 145 when dialing string includes international access code character "+",

	otherwise 129.
<text>	String type field of maximum length <tlength>; often this value is set as name.
<nlength>	Integer type value indicating the maximum length of field <number>
<tlength>	Integer type value indicating the maximum length of field <text>.

Example

```
AT+CPBS?
+CPBS: "SM",2,200

OK

AT+CPBS=1,10
+CPBR: 1,"1234567890",129,"James"
+CPBR: 2,"0987654321",129,"Kevin"

OK
```

NOTE

- If the storage is selected as “SM” then the command will return the record in SIM phonebook, the same to others.

6.2.3 AT+CPBF Find Phonebook entries

AT+CPBF Find Phonebook entries	
Test Command **AT+CPBF=?**	Response **+CPBF: [<nlength>],[<tlength>]** **OK**
Write Command **AT+CPBF=[<findtext>]**	Response **[+CPBF: <index1>,<number>,<type>,<text>[<CR><LF>** **+CPBF: <indexN>,<number>,<type>,<text>[…]]]** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Defined Values

<findtext>	String type, this value is used to find the record. Character set should be the one selected with command AT+CSCS.
<index>	Integer type. the current position number of the Phonebook index.
<number>	String type, phone number of format <type>, the maximum length is <nlength>.
<type>	Type of phone number octet in integer format, default 145 when dialing string includes international access code character "+", otherwise 129.
<text>	String type field of maximum length <tlength>; often this value is set as name.
<nlength>	Integer type value indicating the maximum length of field <number>
<tlength>	Integer type value indicating the maximum length of field <text>.

Example

```
AT+CPBF="James"
+CPBF: 1,"1234567890",129,"James"

OK
```

NOTE

- If <findtext> is null, it will lists all the entries.

6.2.4 AT+CPBW Write Phonebook entry

AT+CPBW Write Phonebook entry	
Test Command **AT+CPBW=?**	Response **+CPBW:(list of supported <index>s),[<nlength>],** **(list of supported <type>s),[<tlength>]** **OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Write Command	Response

AT+CPBW=[<index>][,<number>[,<type>[,<text>]]]	**OK** or **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Defined Values

<index>	Integer type values in the range of location numbers of phonebook memory.If <index> is not given, the first free entry will be used. If <index> is given as the only parameter, the phonebook entry specified by <index> is deleted.If record number <index> already exists, it will be overwritten.
<number>	String type, phone number of format <type>, the maximum length is <nlength>. It must be an non-empty string.
<type>	Type of address octet in integer format, The range of value is from 129 to 255. If <number> contains a leading "+" <type> = 145 (international) is used.Supported value are: 145 – when dialling string includes international access code character "+" 161 – national number.The network support for this type is optional 177 – network specific number,ISDN format 129 – otherwise NOTE: Other value refer TS 24.008 [8] subclause 10.5.4.7.
<text>	String type field of maximum length <tlength>; character set as specified by command Select TE Character Set AT+CSCS.
<nlength>	Integer type value indicating the maximum length of field <number>.
<tlength>	Integer type value indicating the maximum length of field <text>.

Example

```
AT+CPBW=3,"88888888",129,"John"
OK

AT+CPBW=,"6666666",129,"mary"
OK

AT+CPBW=1
OK
```

NOTE

- NOTE: If the parameters of <type> and <text> are omitted and the first character of <number> is

'+', it will specify <type> as 145(129 if the first character isn't '+') and <text> as NULL.

6.2.5 AT+CNUM Subscriber number

AT+CNUM Subscriber number	
Test Command **AT+CNUM=?**	Response **OK**
Execution Command **AT+CNUM**	Response **[+CNUM: <alpha>,<number>,<type>[<CR><LF>** **+CNUM: <alpha>, <number>,<type> [...]]]** **OK** or If error is related to ME functionality: **+CME ERROR: <err>**

Defined Values

<alpha>	Optional alphanumeric string associated with <number>, used character set should be the one selected with command Select TE Character Set AT+CSCS.
<number>	String type phone number of format specified by <type>.
<type>	Type of address octet in integer format.see also AT+CPBR <type>

Example

```
AT+CNUM
+CNUM: "","13697252277",129

OK
```

NOTE

- If subscriber has different MSISDN for different services, each MSISDN is returned in a separate line

7. AT Commands for SIM Application Toolkit

7.1 Overview of AT Commands for SIM Application Toolkit

Command	Description
AT+STIN	SAT Indication
AT+STGI	Get SAT information
AT+STGR	SAT respond
AT+STK	STK switch
AT+STKFMT	Set STK pdu format
AT+STENV	Original STK PDU Envelope Command
AT+STSM	Get STK Setup Menu List with PDU Mode

7.2 Detailed Description of AT Commands for SIM Application Toolkit

7.2.1 AT+STIN SAT Indication

AT+STIN SAT Indication	
Test Command **AT+STIN=?**	Response **OK**
Read Command **AT+STIN?**	**+STIN: <cmd_id>** **OK**

Unsolicited Result Codes

<cmd_id>	Proactive Command notification 21 Display text 22 Get inkey 23 Get input

	24 Select item
+STIN: 25	Notification that SIM Application has returned to main menu. If user doesn't do any action in 2 minutes, application will return to main menu automatically.

Defined Values

<cmd_id>	21 Display text 22 Get inkey 23 Get input 24 Select item 25 Set up menu 81 Session end (pdu mode only) 0 None command
<time>	Service time

Example

```
AT+STIN?
+STIN: 24

OK
```

NOTE

- Every time the SIM Application issues a Proactive Command, via the ME, the TA will receive an indication. This indicates the type of Proactive Command issued.

7.2.2 AT+STGI Get SAT information

AT+STGI Get SAT information	
Test Command **AT+STGI=?**	Response **OK**
Write Command **AT+STGI=<cmd_id>**	Response (***PDU format)*** **+STGI: <cmd_id>,<tag>,<pdu_len>,<pdu_value>** **OK**
AT+STGI=<cmd_id>	Response (**NOT *PDU format,* listed below*)*** If <cmd_id>=10: **OK** If <cmd_id>=21:

	+STGI: 21,<prio>,<clear_mode>,<text_len>,<text> **OK**
	If <cmd_id>=22: **+STGI: 22,< rsp_format>,< help>,<text_len>,<text>** **OK**
	If <cmd_id>=23: **+STGI: 23,<rsp_format>,<max_len>,<min_len>,<help>,<show>,<text_len>,<text>** **OK**
	If <cmd_id>=24: **+STGI: 24,<help>,<softkey>,<present>,<title_len>,<title>,<item_num>** **+STGI: 24,<item_id>,<item_len>,<item_data>** **[…]** **OK**
	If <cmd_id>=25: **+STGI: 25,<help>,<softkey>,<title_len>,<title>,<item_num>** **+STGI: 25,<item_id>,<item_len>,<item_data>** **[...]** **OK**

Defined Values

<cmd_id>	Proactive Command notification 21 Display text 22 Get inkey 23 Get input 24 Select item 25 Set up menu
<prio>	Priority of display text 0 Normal priority 1 High priority
<clear_mode>	0 Clear after a delay 1 Clear by user
<text_len>	Length of text
<rsp_format>	0 SMS default alphabet 1 YES or NO 2 numerical only 3 UCS2
<help>	0 Help unavailable 1 Help available
<max_len>	Maximum length of input
<min_len>	Minimum length of input
<show>	0 Hide input text

	1 Display input text
<softkey>	0 No softkey preferred 1 Softkey preferred
<present>	Menu presentation format available for select item 0 Presentation not specified 1 Data value presentation 2 Navigation presentation
<title_len>	Length of title
<item_num>	Number of items in the menu
<item_id>	Identifier of item
<item_len>	Length of item
<title>	Title in ucs2 format
<item_data>	Content of the item in ucs2 format
<text>	Text in ucs2 format.
<tag>	Not used now.
<pdu_len>	Integer type, pdu string length
<pdu_val>	String type, the pdu string.

Example

```
AT+STGI=25 (NOT PDU format)
+STGI:
25,0,0,10,"795E5DDE884C59295730",15
+STGI: 25,1,8,"8F7B677E95EE5019"
+STGI: 25,2,8,"77ED4FE17FA453D1"
+STGI: 25,3,8,"4F1860E05FEB8BAF"
+STGI: 25,4,8,"4E1A52A17CBE9009"
+STGI: 25,5,8,"8D448D3963A88350"
+STGI: 25,6,8,"81EA52A9670D52A1"
+STGI: 25,7,8,"8F7B677E5F6994C3"
+STGI: 25,8,8,"8BED97F367425FD7"
+STGI: 25,9,10,"97F34E506392884C699C"
+STGI: 25,10,8,"65B095FB59296C14"
+STGI: 25,11,8,"94C358F056FE7247"
+STGI: 25,12,8,"804A59294EA453CB"
+STGI: 25,13,8,"5F005FC34F1195F2"
+STGI: 25,14,8,"751F6D3B5E388BC6"
+STGI:
25,21,12,"00530049004D53614FE1606F"

OK

AT+STGI=24 (PDU format)
+STGI:
```

```
24,0,48,"D02E810301240082028185098070
ED70B963A883508F0A018053057F574E078C
618F0C02809177917777ED6D88606F"
OK
```

7.2.3 AT+STGR SAT respond

AT+STGR SAT respond	
Test Command **AT+STGR=?**	Response **OK**
Write Command **AT+STGR=<cmd_id>[,<data>]**	Response (***NOT PDU format)*** **OK**
AT+STGR=<pdu_len>,<pdu_value>	Response (***PDU format)*** **OK**

Defined Values

<cmd_id>	Proactive Command notification 21 Display text 22 Get inkey 23 Get input 24 Select item 25 Set up menu 81 Session end 83 Session end by user 84 Go backward
<data>	If <cmd_id>=22: Input a character If <cmd_id>=23: Input a string. If <rsp_format> is YES or NO, input of a character in case of ANSI character set requests one byte, e.g. “Y”. If <rsp_format> is numerical only, input the characters in decimal number, e.g. “123” If <rsp_faomat> is UCS2, requests a 4 byte string, e.g. “0031” <rsp_faomat> refer to the response by AT+STGI=23 If <cmd_id>=24: Input the identifier of the item selected by user If <cmd_id>=25: Input the identifier of the item selected by user If <cmd_id>=83:

	<data> ignore Note: It could return main menu during Proactive Command id is not 22 or 23 If <cmd_id>= 84: <data> ignore
<pdu_len>	Integer type, pdu string length
<pdu_value>	String type, the pdu string.

Example

```
AT+STGR=25,1     (NOT PDU format)
OK
+STIN: 24

AT+STGR=30,"810301240002028281830100900101"    (PDU format)
OK
```

NOTE

- After selected an item, different SIM/USIM cards will report different +STIN: command.

7.2.4 AT+STK STK switch

AT+STK STK switch	
Test Command **AT+STK=?**	Response **+STK:** (list of supported <**value**>s) OK
Read Command **AT+STK?**	Response **+STK: <value>** **OK**
Write Command **AT+STK=<value>**	Response **OK** or **ERROR**
Execution Command **AT+STK**	Response **OK**

Defined Values

<value>	0 Disable STK 1 Enable STK

Example

```
AT+STK=1
OK
```

NOTE

- Module should reboot to take effective

7.2.5 AT+STKFMT Set STK pdu format

AT+STKFMT Set STK pdu format	
Read Command **AT+STKFMT?**	Response **+STKFMT: <value>** **OK**
Write Command **AT+STKFMT=<value>**	Response **OK** or **ERROR**

Defined Values

<value>	0 Disable STK pdu format, decoded command mode. 1 Enable STK pdu format

Example

```
AT+STKFMT=1
OK
```

NOTE

- Module should reboot to take effective

7.2.6 AT+STENV Original STK PDU Envelope Command

AT+STENV Original STK PDU Envelope Command	
Test Command **AT+STENV=?**	Response **OK**
Write Command **AT+STENV=<len>,<pdu>**	Response **OK** or **ERROR**

Defined Values

<len>	Integer type, pdu string length
<pdu>	String type, pdu value

Example

```
AT+STENV=18,"D30782020181900101"
OK
```

NOTE

- Module should reboot to take effective

7.2.7 AT+STSM Get STK Setup Menu List with PDU Mod

AT+STSM Get STK Setup Menu List with PDU Mod	
Test Command **AT+STSM=?**	Response **OK**
Read Command **AT+STSM?**	Response **+STSM: <cmd_id>,<tag>,<pdu_len>, <pdu_value>** **OK** or **ERROR**

Defined Values

<cmd_id>	Integer type, please refer to AT+STIN
<tag>	Not used now.
<pdu_len>	Integer type, pdu string length
<pdu_value>	String type, the pdu string.

Example

```
AT+STSM?
+STSM:
25,0,120,"D07681030125008202818285078065B052BF529B8F0A018070ED70B963A883508F06028070AB94C38F0A03806D41884C77ED4FE18F0A048081EA52A9670D52A18F0A0580624B673A97F34E508F0606808D854FE18F0A07805A314E50753162118F0A0880767E53D8751F6D3B8F0A09806D596C5F98919053"

OK
```

NOTE

- Setup main menu info got first before envelope command sent.

8. AT Commands for GPRS

8.1 Overview of AT Commands for GPRS

Command	Description
AT+CGREG	GPRS network registration status
AT+CGATT	Packet domain attach or detach
AT+CGACT	PDP context activate or deactivate
AT+CGDCONT	Define PDP context
AT+CGDSCONT	Define Secondary PDP Context
AT+CGTFT	Traffic Flow Template
AT+CGQREQ	Quality of service profile (requested)
AT+CGEQREQ	3G quality of service profile (requested)
AT+CGQMIN	Quality of service profile (minimum acceptable)
AT+CGEQMIN	3G quality of service profile (minimum acceptable)
AT+CGDATA	Enter data state
AT+CGPADDR	Show PDP address
AT+CGCLASS	GPRS mobile station class
AT+CGEREP	GPRS event reporting
AT+CGAUTH	Set type of authentication for PDP-IP connections of GPRS

8.2 Detailed Description of AT Commands for GPRS

8.2.1 AT+CGREG GPRS network registration status

This command controls the presentation of an unsolicited result code “+CGREG: <stat>” when <n>=1 and there is a change in the MT's GPRS network registration status.

The read command returns the status of result code presentation and an integer <stat> which shows whether the network has currently indicated the registration of the MT.

AT+CGREG GPRS network registration status	
Test Command **AT+CGREG=?**	Response **+CGREG:** (list of supported **<n>**s) **OK**
Read Command **AT+CGREG?**	Response **+CGREG: <n>,<stat>[,<lac>,<ci>]** **OK**
Write Command **AT+CGREG=<n>**	Response **OK**
Execution Command Set default value: **AT+CGREG**	Response **OK**

Defined Values

<n>	0 – disable network registration unsolicited result code 1 – enable network registration unsolicited result code +CGREG: <stat> 2 – there is a change in the ME network registration status or a change of the network cell: +CGREG: <stat>[,<lac>,<ci>]
<stat>	0 – not registered, ME is not currently searching an operator to register to 1 – registered, home network 2 – not registered, but ME is currently trying to attach or searching an operator to register to 3 – registration denied 4 – unknown 5 – registered, roaming
<lac>	Two bytes location area code in hexadecimal format (e.g.”00C3” equals 193 in decimal).
<ci>	Cell ID in hexadecimal format. GSM : Maximum is two byte WCDMA : Maximum is four byte TDS-CDMA : Maximum is four byte

NOTE

The **<lac>** not supported in CDMA/HDR mode
The **<ci>** not supported in CDMA/HDR mode

Example

```
AT+CGREG=?
+CGREG: (0-1)

OK
AT+CGREG?
+CGREG: 0,0

OK
```

8.2.2 AT+CGATT Packet domain attach or detach

The write command is used to attach the MT to, or detach the MT from, the Packet Domain service.
The read command returns the current Packet Domain service state.

AT+CGATT Packet domain attach or detach	
Test Command **AT+CGATT=?**	Response **+CGATT: (list of supported <state>s)** **OK**
Read Command **AT+CGATT?**	Response **+CGATT: <state>** **OK**
Write Command **AT+CGATT=<state>**	Response **OK** or **ERROR** or **+CME ERROR: <err>**

Defined Values

<state>	Indicates the state of Packet Domain attachment: 0 – detached 1 – attached

Example

```
AT+CGATT?
+CGATT: 0
```

```
OK
AT+CGATT=1
OK
```

8.2.3 AT+CGACT GPRS network registration status

The write command is used to activate or deactivate the specified PDP context (s).

AT+CGACT GPRS network registration status	
Test Command **AT+CGACT=?**	Response **+ CGACT:** (list of supported **<state>**s) **OK**
Read Command **AT+CGACT?**	Response **+CGACT: [<cid>, <state> [<CR><LF>** **+CGACT: <cid>, <state>** **[...]]]** **OK**
Write Command **AT+CGACT=<state>[,<cid>]**	Response **OK** or **ERROR** or **+CME ERROR: <err>**

Defined Values

<state>	Indicates the state of PDP context activation: 0 – deactivated 1 – activated
<cid>	A numeric parameter which specifies a particular PDP context definition (see AT+CGDCONT command). 1…16

Example

```
AT+CGACT=?
+CGACT: (0,1)

OK
```

```
AT+CGACT?
+CGACT: 1,1

OK
AT+CGACT=0,1
OK
```

8.2.4 AT+CGDCONT Define PDP context

The set command specifies PDP context parameter values for a PDP context identified by the (local) context identification parameter **<cid>**. The number of PDP contexts that may be in a defined state at the same time is given by the range returned by the test command. A special form of the write command (**AT+CGDCONT=<cid>**) causes the values for context **<cid>** to become undefined.

AT+CGDCONT Define PDP context	
Test Command **AT+CGDCONT=?**	Response **+CGDCONT: (range of supported<cid>s),<PDP_type>,,,(list of supported <d_comp>s),(list of supported <h_comp>s) (list of <ipv4_ctrl>s),(list of <emergency_flag>s)** **OK** or **ERROR**
Read Command **AT+CGDCONT?**	Response **+CGDCONT: [<cid>, <PDP_type>,<APN>,<PDP_addr>,<d_comp>, <h_comp><ipv4_ctrl>,<emergency_flag>[<CR><LF>** **+CGDCONT: <cid>, <PDP_type>, <APN>,<PDP_addr>, <d_comp>, <h_comp>< ipv4_ctrl>,<emergency_flag>[...]]]** **OK** or **ERROR**
Write Command **AT+CGDCONT=<cid>[,<PDP_type>[,<APN>[,<PDP_addr>[,<d_comp>[,<h_comp>[,<ipv4_ctrl>[,<emergency_flag>]]]]]]]**	Response **OK** or **ERROR**
Execution Command Set default value: **AT+CGDCONT**	Response **OK** or

	ERROR

Defined Values

<cid>	(PDP Context Identifier) a numeric parameter which specifies a particular PDP context definition. The parameter is local to the TE-MT interface and is used in other PDP context-related commands. The range of permitted values (minimum value = 1) is returned by the test form of the command. 1…16,100…179
<PDP_type>	(Packet Data Protocol type) a string parameter which specifies the type of packet data protocol. IP Internet Protocol PPP Point to Point Protocol IPV6 Internet Protocol Version 6 IPV4V6 Dual PDN Stack
<APN>	(Access Point Name) a string parameter which is a logical name that is used to select the GGSN or the external packet data network.
<PDP_addr>	A string parameter that identifies the MT in the address space applicable to the PDP. Read command will continue to return the null string even if an address has been allocated during the PDP startup procedure. The allocated address may be read using command **AT+CGPADDR**.
<d_comp>	A numeric parameter that controls PDP data compression, this value may depend on platform: 0 – off (default if value is omitted) 1 – on 2 – V.42bis
<h_comp>	A numeric parameter that controls PDP header compression, this value may depend on platform: 0 – off (default if value is omitted) 1 – on 2 – RFC1144 3 – RFC2507 4 – RFC3095

Example

```
AT+CGDCONT =?
+CGDCONT:
(1-24,100-179),"IP",,,(0-2),(0-4),(0-1),(0-1)
+CGDCONT:
(1-24,100-179),"PPP",,,(0-2),(0-4),(0-1),(0-1)
+CGDCONT:
(1-24,100-179),"IPV6",,,(0-2),(0-4),(0-1),(0-1)
+CGDCONT:
```

(1-24,100-179),"IPV4V6",,,(0-2),(0-4),(0-1),(0-1) OK AT+ CGDCONT? +CGDCONT: 1,"IP","","0.0.0.0",0,0 OK	

8.2.5 AT+CGDSCONT Define Secondary PDP Context

The set command specifies PDP context parameter values for a Secondary PDP context identified by the (local) context identification parameter, **<cid>**. The number of PDP contexts that may be in a defined state at the same time is given by the range returned by the test command. A special form of the set command, AT+CGDSCONT=**<cid>** causes the values for context number **<cid>** to become undefined.

AT+CGDSCONT Define Secondary PDP Context	
Test Command AT+CGDSCONT=?	Response **+CGDSCONT: (range of supported <cid>s),(list of <p_cid>s for active primary contexts), <PDP_type>,,,(list of supported <d_comp>s),(list of supported <h_comp>s)** **OK** or **ERROR**
Read Command AT+CGDSCONT?	Response **+CGDSCONT: [<cid>,<p_cid>,<d_comp>,<h_comp>** **[<CR><LF>+CGDSCONT: <cid>,<p_cid>,<d_comp>,<h_comp>** **[...]]]** **OK** or **ERROR**
Write Command AT+CGDSCONT=<cid>[,<p_cid>[,<d_comp>[,<h_comp>]]]	Response **OK** or **ERROR**

Defined Values

<cid>	a numeric parameter which specifies a particular PDP context definition. The parameter is local to the TE-MT interface and is used in other PDP

	context-related commands. The range of permitted values (minimum value = 1) is returned by the test form of the command.
<p_cid>	a numeric parameter which specifies a particular PDP context definition which has been specified by use of the +CGDCONT command. The parameter is local to the TE-MT interface. The list of permitted values is returned by the test form of the command.
<PDP_type>	(Packet Data Protocol type) a string parameter which specifies the type of packet data protocol. IP Internet Protocol PPP Point to Point Protocol IPV6 Internet Protocol Version 6 IPV4V6 Dual PDN Stack
<d_comp>	a numeric parameter that controls PDP data compression (applicable for SNDCPonly) (refer 3GPP TS 44.065 [61]) 0 off 1 on (manufacturer preferred compression) 2 V.42bis Other values are reserved.
<h_comp>	a numeric parameter that controls PDP header compression (refer 3GPP TS 44.065 [61] and 3GPP TS 25.323 [62]) 0 off 1 on (manufacturer preferred compression) 2 RFC1144 (applicable for SNDCP only) 3 RFC2507 4 RFC3095 (applicable for PDCP only) Other values are reserved.

NOTE

The <cid>s for network-initiated PDP contexts will have values outside the ranges indicated for the <cid> in the test form of the commands +CGDCONT and +CGDSCONT.

Example

```
AT+CGDSCONT=?
+CGDSCONT: (1-24,100-179),(),"IP",(0-2),(0-4)
+CGDSCONT: (1-24,100-179),(),"PPP",(0-2),(0-4)
+CGDSCONT:
(1-24,100-179),(),"IPV6",(0-2),(0-4)
+CGDSCONT:
(1-24,100-179),(),"IPV4V6",(0-2),(0-4)
```

```
OK
AT+CGDSCONT?
+CGDSCONT: 2,1,0,0

OK
AT+CGDSCONT=2,1
OK
```

8.2.6 AT+CGTFT Traffic Flow Template

This command allows the TE to specify a Packet Filter - PF for a Traffic Flow Template - TFT that is used in the GGSN in UMTS/GPRS and Packet GW in EPS for routing of packets onto different QoS flows towards the TE. The concept is further described in the 3GPP TS 23.060 [47]. A TFT consists of from one and up to 16 Packet Filters, each identified by a unique <packet filter identifier>. A Packet Filter also has an <evaluation precedence index> that is unique within all TFTs associated with all PDP contexts that are associated with the same PDP address.

AT+CGTFT Traffic Flow Template	
Test Command AT+CGTFT=?	Response **+CGTFT: <PDP_type>,(list of supported <packet filter identifier>s),(list of supported <evaluation precedence index>s),(list of supported <source address and subnet mask>s),(list of supported <protocol number (ipv4) / next header (ipv6)>s),(list of supported <destination port range>s),(list of supported <source port range>s),(list of supported <ipsec security parameter index (spi)>s),(list of supported <type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>s),(list of supported <flow label (ipv6)>s),(list of supported <direction>s)** **[<CR><LF>+CGTFT: <PDP_type>,(list of supported <packet filter identifier>s),(list of supported <evaluation precedence index>s),(list of supported <source address and subnet mask>s),(list of supported <protocol number (ipv4) / next header (ipv6)>s),(list of supported <destination port range>s),(list of supported <source port range>s),(list of supported <ipsec security parameter index (spi)>s),(list of supported <type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>s),(list of supported <flow label (ipv6)>s),(list of supported <direction>s)** **[...]]** **OK**

	or **ERROR**
Read Command **AT+CGTFT?**	Response **+CGTFT: [<cid>,<packet filter identifier>,<evaluation precedence index>,<source address and subnet mask>,<protocol number (ipv4) / next header (ipv6)>,<destination port range>,<source port range>,<ipsec security parameter index (spi)>,<type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>,<flow label (ipv6)>,<direction>** **[<CR><LF>+CGTFT: <cid>,<packet filter identifier>,<evaluation precedence index>,<source address and subnet mask>,<protocol number (ipv4) / next header (ipv6)>,<destination port range>,<source port range>,<ipsec security parameter index (spi)>,<type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>,<flow label (ipv6)>,<direction>** **[...]]]** **OK** or **ERROR**
Write Command **AT+CGTFT=<cid>[,[<packet filter identifier>,<evaluation precedence index>[,<source address and subnet mask>[,<protocol number (ipv4) / next header (ipv6)>[,<destination port range>[,<source port range>[,<ipsec security parameter index (spi)>[,<type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>[,<flow label (ipv6)>[,<direction>]]]]]]]]]**	Response **OK** or **ERROR**
Execution Command **AT+CGTFT**	Response **OK** or **ERROR**

Defined Values

<cid>	a numeric parameter which specifies a particular PDP context definition (see the AT+CGDCONT and AT+CGDSCONT commands).
<PDP_type>	(Packet Data Protocol type) a string parameter which specifies the type of packet data protocol. IP Internet Protocol PPP Point to Point Protocol IPV6 Internet Protocol Version 6 IPV4V6 Dual PDN Stack
<packet filter identifier>	a numeric parameter, value range from 1 to 16.
<evaluation precedence index>	a numeric parameter. The value range is from 0 to 255.
<source address and subnet mask>	string type The string is given as dot-separated numeric (0-255) parameters on the form: "a1.a2.a3.a4.m1.m2.m3.m4" for IPv4 or "a1.a2.a3.a4.a5.a6.a7.a8.a9.a10.a11.a12.a13.a14.a15.a16.m1.m2.m3.m4.m5.m6.m7.m8.m9.m10.m11.m12.m13.m14.m15.m16", for IPv6.
<protocol number (ipv4) / next header (ipv6)>	a numeric parameter, value range from 0 to 255.
<destination port range>	string type. The string is given as dot-separated numeric (0-65535) parameters on the form "f.t".
<source port range>	string type. The string is given as dot-separated numeric (0-65535) parameters on the form "f.t".
<ipsec security parameter index (spi)>	numeric value in hexadecimal format. The value range is from 00000000 to FFFFFFFF.
<type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>	string type. The string is given as dot-separated numeric (0-255) parameters on the form "t.m".
<flow label (ipv6)>	numeric value in hexadecimal format. The value range is from 00000 to FFFFF. Valid for IPv6 only.
<direction>	a numeric parameter which specifies the transmission direction in which the packet filter shall be applied. 0 Pre-Release 7 TFT filter (see 3GPP TS 24.008 [8], table 10.5.162) 1 Uplink 2 Downlink 3 Birectional (Up & Downlink)

Example

```
AT+CGTFT=?
+CGTFT:
"IP",(1-16),(0-255),,(0-255),(0-65535.0-65535),(0-65535.0-65535),(0-FFFFFFFF),(0-255.0-255),(0-FFFFF)
+CGTFT:
"PPP",(1-16),(0-255),,(0-255),(0-65535.0-65535),(0-65535.0-65535),(0-FFFFFFFF),(0-255.0-255),(0-FFFFF)
```

```
+CGTFT:
"IPV6",(1-16),(0-255),,(0-255),(0-65535.0-65535),(0-65535.0-65535),(0-FFFFFFFF),(0-255.0-255),(0-FFFFF)
+CGTFT:
"IPV4V6",(1-16),(0-255),,(0-255),(0-65535.0-65535),(0-65535.0-65535),(0-FFFFFFFF),(0-255.0-255),(0-FFFFF)

OK
AT+CGTFT?
+CGTFT: 2,1,0,"74.125.71.99.255.255.255.255",0,0.0,0.0,0,0.0,0

OK
AT+CGTFT=2,1,0,"74.125.71.99.255.255.255.255"

OK
```

8.2.7 AT+CGQREQ Quality of service profile (requested)

This command allows the TE to specify a Quality of Service Profile that is used when the MT sends an Activate PDP Context Request message to the network. A special form of the set command (AT+CGQREQ=<cid>) causes the requested profile for context number <cid> to become undefined.

AT+CGQREQ Quality of service profile (requested)	
Test Command **AT+CGQREQ=?**	Response **+CGQREQ: <PDP_type>, (list of supported <precedence>s), (list of supported <delay>s), (list of supported <reliability>s) , (list of supported <peak>s), (list of supported <mean>s) [<CR><LF> +CGQREQ: <PDP_type>, (list of supported <precedence>s), (list of supported <delay>s), (list of supported <reliability>s) , (list of supported <peak>s), (list of supported <mean>s) […]]** **OK** or **ERROR**
Read Command **AT+CGQREQ?**	Response **+CGQREQ: [<cid>, <precedence >, <delay>, <reliability>, <peak>, <mean>[<CR><LF>** **+CGQREQ: <cid>, <precedence >, <delay>, <reliability.>, <peak>, <mean>[…]]]** **OK** or

	ERROR
Write Command **AT+CGQREQ=<cid>[,<precedence>[,<delay>[,<reliability>[,<peak> [,<mean>]]]]]**	Response **OK** or **ERROR**
Execution Command **AT+CGQREQ**	Response **OK** or **ERROR**

Defined Values

<cid>	A numeric parameter which specifies a particular PDP context definition (see AT+CGDCONT command). The range is from 1 to 42,100 to 179.
<PDP_type>	(Packet Data Protocol type) a string parameter which specifies the type of packet data protocol. IP Internet Protocol PPP Point to Point Protocol IPV6 Internet Protocol Version 6 IPV4V6 Dual PDN Stack
<precedence>	A numeric parameter which specifies the precedence class: 0 – network subscribed value 1 – high priority 2 – normal priority 3 – low priority
<delay>	A numeric parameter which specifies the delay class: 0 – network subscribed value 1 – delay class 1 2 – delay class 2 3 – delay class 3 4 – delay class 4
<reliability>	A numeric parameter which specifies the reliability class: 0 – network subscribed value 1 – Non real-time traffic,error-sensitive application that cannot cope with data loss 2 – Non real-time traffic,error-sensitive application that can cope with infrequent data loss 3 – Non real-time traffic,error-sensitive application that can cope with data loss, GMM/- SM,and SMS 4 – Real-time traffic,error-sensitive application that can cope with data loss 5 – Real-time traffic error non-sensitive application that can cope with data loss
<peak>	A numeric parameter which specifies the peak throughput class:

	0 – network subscribed value 1 – Up to 1000 (8 kbit/s) 2 – Up to 2000 (16 kbit/s) 3 – Up to 4000 (32 kbit/s) 4 – Up to 8000 (64 kbit/s) 5 – Up to 16000 (128 kbit/s) 6 – Up to 32000 (256 kbit/s) 7 – Up to 64000 (512 kbit/s) 8 – Up to 128000 (1024 kbit/s) 9 – Up to 256000 (2048 kbit/s)
<mean>	A numeric parameter which specifies the mean throughput class: 0 – network subscribed value 1 – 100 (~0.22 bit/s) 2 – 200 (~0.44 bit/s) 3 – 500 (~1.11 bit/s) 4 – 1000 (~2.2 bit/s) 5 – 2000 (~4.4 bit/s) 6 – 5000 (~11.1 bit/s) 7 – 10000 (~22 bit/s) 8 – 20000 (~44 bit/s) 9 – 50000 (~111 bit/s) 10 – 100000 (~0.22 kbit/s) 11 – 200000 (~0.44 kbit/s) 12 – 500000 (~1.11 kbit/s) 13 – 1000000 (~2.2 kbit/s) 14 – 2000000 (~4.4 kbit/s) 15 – 5000000 (~11.1 kbit/s) 16 – 10000000 (~22 kbit/s) 17 – 20000000 (~44 kbit/s) 18 – 50000000 (~111 kbit/s) 31 – optimization

Example

```
AT+CGQREQ=?
+CGQREQ: "IP",(0-3),(0-4),(0-5),(0-9),(0-18,31)
+CGQREQ: "PPP",(0-3),(0-4),(0-5),(0-9),(0-18,31)
+CGQREQ: "IPV6",(0-3),(0-4),(0-5),(0-9),(0-18,31)
+CGQREQ: "IPV4V6",(0-3),(0-4),(0-5),(0-9),(0-18,31)
OK
AT+CGREG?
+CGQREQ:

OK
```

8.2.8 AT+CGEQREQ 3G quality of service profile (requested)

The test command returns values supported as a compound value.

The read command returns the current settings for each defined context for which a QOS was explicitly specified.

The write command allows the TE to specify a Quality of Service Profile for the context identified by the context identification parameter **<cid>** which is used when the MT sends an Activate PDP Context Request message to the network.

A special form of the write command, **AT+CGEQREQ=<cid>** causes the requested profile for context number **<cid>** to become undefined.

AT+CGEQREQ 3G quality of service profile (requested)	
Test Command **AT+CGEQREQ=?**	Response **+CGEQREQ: <PDP_type>,(list of supported <Traffic class>s),(list of supported <Maximum bitrate UL>s),(list of supported <Maximum bitrate DL>s),(list of supported <Guaranteed bitrate UL>s,(list of supported <Guaranteed bitrate DL>s),(list of supported <Delivery order>s),(list of supported <Maximum SDU size>s),(list of supported <SDU error ratio>s),(list of supported <Residual bit error** **Ratio>s),(list of supported <Delivery of erroneous SDUs>s),(list of** **Supported <Transfer delay>s),(list of supported <Traffic handling priority>s)** **OK** or **ERROR**
Read Command **AT+CGEQREQ?**	Response **+CGEQREQ: [<cid>,<Traffic class>,<Maximum bitrate UL>,<Maximum bitrate DL>,<Guaranteed bitrate UL>,<Guaranteed bitrate DL>,<Delivery order>,<Maximum SDU size>,<SDU error ratio>,** **<Residual bit error ratio>,<Delivery of erroneous SDUs>,<Transfer** **Delay>,<Traffic handling priority>][<CR><LF>** **+CGEQREQ: <cid>,<Traffic class>,<Maximum bitrate UL>,<Maximum bitrate DL>,<Guaranteed bitrate UL>,<Guaranteed bitrate DL>,<Delivery order>,<Maximum SDU size>,<SDU error ratio>,** **<Residual bit error ratio>,<Delivery of erroneous SDUs>,<Transfer**

	Delay>,<Traffic handling priority>[…]] **OK** or **ERROR**
Write Command **AT+CGEQREQ=<cid>[,<Traffic class>[,<Maximum bitrate UL>[,<Maximum bitrate DL>[,<Guaranteed bitrateUL>[,<Guaranteed bitrate DL>[,<Delivery order>[,<Maximum SDU size>[,<SDU error ratio>[,<Residual bit error ratio>[,<Delivery of erroneous SDUs>[,<Transfer delay>[,<Traffic handling priority>]]]]]]]]]]]]**	Response **OK** or **ERROR** or **+CME ERROR: <err>**
Execution Command **AT+ CGEQREQ**	Response **OK** or **ERROR**

Defined Values

<cid>	Parameter specifies a particular PDP context definition.The parameter is also used in other PDP context-related commands. The range is from 1 to 24,100 to 179.
<Traffic class>	0 – conversational 1 – streaming 2 – interactive 3 – background 4 – subscribed value
<Maximum bitrate UL>	This parameter indicates the maximum number of kbits/s delivered to UMTS(up-link traffic)at a SAP. As an example a bitrate of 32kbit/s would be specified as 32(e.g. AT+CGEQREQ=…,32,…). The range is from 0 to 8460. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<Maximum bitrate DL>	This parameter indicates the maximum number of kbits/s delivered to UMTS(down-link traffic)at a SAP.As an example a bitrate of 32kbit/s would be specified as 32(e.g. AT+CGEQREQ=…,32,…). The range is from 0 to 8460. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.

<Guaranteed bitrate UL>	This parameter indicates the guaranteed number of kbit/s delivered to UMTS(up-link traffic)at a SAP(provided that there is data to deliver).As an example a bitrate of 32kbit/s would be specified as 32(e.g.AT+CGEQREQ=…,32,…). The range is from 0 to 8460. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<Guaranteed bitrate DL>	This parameter indicates the guaranteed number of kbit/s delivered to UMTS(down-link traffic)at a SAP(provided that there is data to deliver).As an example a bitrate of 32kbit/s would be specified as 32(e.g.AT+CGEQREQ=…,32,…). The range is from 0 to 8460. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<Delivery order>	This parameter indicates whether the UMTS bearer shall provide in-sequence SDU delivery or not. 0 – no 1 – yes 2 – subscribed value
<Maximum SDU size>	This parameter indicates the maximum allowed SDU size in octets. The range is from 0 to 1520. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<SDU error ratio>	This parameter indicates the target value for the fraction of SDUs lost or detected as erroneous.SDU error ratio is defined only for conforming traffic.As an example a target SDU error ratio of 5*10-3 would be specified as “5E3”(e.g.AT+CGEQREQ=..,”5E3”,…). “0E0” – subscribed value “1E2” “7E3” “1E3” “1E4” “1E5” “1E6” “1E1”
<Residual bit error ratio>	This parameter indicates the target value for the undetected bit error ratio in the delivered SDUs. If no error detection is requested,Residual bit error ratio indicates the bit error ratio in the delivered SDUs.As an example a target residual bit error ratio of 5*10-3 would be specified as “5E3”(e.g. AT+CGEQREQ=…,”5E3”,..).

	"0E0" – subscribed value "5E2" "1E2" "5E3" "4E3" "1E3" "1E4" "1E5" "1E6" "6E8"
<Delivery of erroneous SDUs>	This parameter indicates whether SDUs detected as erroneous shall be delivered or not. 0 – no 1 – yes 2 – no detect 3 – subscribed value
<Transfer delay>	This parameter indicates the targeted time between request to transfer an SDU at one SAP to its delivery at the other SAP,in milliseconds. The range is from 0 to 4000. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<Traffic handling priority>	This parameter specifies the relative importance for handling of all SDUs belonging to the UMTS Bearer compared to the SDUs of the other bearers. The range is from 0 to 3. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<PDP_type>	(Packet Data Protocol type) a string parameter which specifies the type of packet data protocol. IP Internet Protocol PPP Point to Point Protocol IPV6 Internet Protocol Version 6 IPV4V6 Dual PDN Stack

Example

```
AT+CGEQREQ=?
+CGEQREQ: "IP",(0-4),(0-384),(0-7168),(0-384),(0-7168),(0-2),(0-1520),("0E0","1E
```

```
1","1E2","7E3","1E3","1E4","1E5","1E6"),("0E0","5E2","1E2","5E3","4E3","1E3","1E
4","1E5","1E6","6E8"),(0-3),(0-4000),(0-3) ,(0,1),(0,1)
+CGEQREQ: "PPP",(0-4),(0-384),(0-7168),(0-384),(0-7168),(0-2),(0-1520),("0E0","1
E1","1E2","7E3","1E3","1E4","1E5","1E6"),("0E0","5E2","1E2","5E3","4E3","1E3","1
E4","1E5","1E6","6E8"),(0-3),(0-4000),(0-3) ,(0,1),(0,1)
+CGEQREQ: "IPV6",(0-4),(0-384),(0-7168),(0-384),(0-7168),(0-2),(0-1520),("0E0","
1E1","1E2","7E3","1E3","1E4","1E5","1E6"),("0E0","5E2","1E2","5E3","4E3","1E3","
1E4","1E5","1E6","6E8"),(0-3),(0-4000),(0-3) ,(0,1),(0,1)
+CGEQREQ:"IPV4V6",(0-4),(0-5760),(0-14000),(0-5760),(0-14000),(0-2),(0-1520),("0E0","1E1","1E
2","7E3","1E3","1E4","1E5","1E6"),("0E0","5E2","1E2","5E3","4E3","1E3","1E4","1E5","1E6","6
E8"),(0-3),(0-4000),(0-3),(0,1),(0,1)

OK
AT+CGEQREQ?
+CGEQREQ:

OK
```

8.2.9 AT+CGQMIN Quality of service profile (minimum acceptable)

This command allows the TE to specify a minimum acceptable profile which is checked by the MT against the negotiated profile returned in the Activate PDP Context Accept message. A special form of the set command. **AT+CGQMIN=<cid>** causes the minimum acceptable profile for context number **<cid>** to become undefined.

AT+CGQMIN Quality of service profile (minimum acceptable)	
Test Command AT+CGQMIN=?	Response **+CGQMIN: <PDP_type>, (list of supported <precedence>s), (list of supported <delay>s), (list of supported <reliability>s) , (list of supported <peak>s), (list of supported <mean>s) [<CR><LF> +CGQMIN: <PDP_type>, (list of supported <precedence>s), (list of supported <delay>s), (list of supported <reliability>s) , (list of supported <peak>s), (list of supported <mean>s)[…]]** **OK** or **ERROR**
Read Command AT+CGQMIN?	Response **+CGQMIN: [<cid>,<precedence>,<delay>,<reliability>,<peak>, <mean>[<CR><LF>** **+CGQMIN: <cid>,<precedence>,<delay>,<reliability.>,<peak>, <mean>**

	[…]]] **OK** or **ERROR**
Write Command **AT+CGQMIN=<cid>[,<precedence>[,<delay>[,<reliability>[,<peak> [,<mean>]]]]]**	Response **OK** or **ERROR**
Execution Command **AT+CGQMIN**	Response **OK** or **ERROR**

Defined Values

<cid>	A numeric parameter which specifies a particular PDP context definition (see AT+CGDCONT command). The range is from 1 to 24,100 to 179.
<PDP_type>	(Packet Data Protocol type) a string parameter which specifies the type of packet data protocol. IP Internet Protocol PPP Point to Point Protocol IPV6 Internet Protocol Version 6 IPV4V6 Dual PDN Stack
<precedence>	A numeric parameter which specifies the precedence class: 0 – network subscribed value 1 – high priority 2 – normal priority 3 – low priority
<delay>	A numeric parameter which specifies the delay class: 0 – network subscribed value 1 – delay class 1 2 – delay class 2 3 – delay class 3 4 – delay class 4
<reliability>	A numeric parameter which specifies the reliability class: 0 – network subscribed value 1 – Non real-time traffic,error-sensitive application that cannot cope with data loss 2 – Non real-time traffic,error-sensitive application that can cope with infrequent data loss 3 – Non real-time traffic,error-sensitive application that can cope with data loss, GMM/- SM,and SMS 4 – Real-time traffic,error-sensitive application that can cope with data

	loss 5 – Real-time traffic error non-sensitive application that can cope with data loss
<peak>	A numeric parameter which specifies the peak throughput class: 0 – network subscribed value 1 – Up to 1000 (8 kbit/s) 2 – Up to 2000 (16 kbit/s) 3 – Up to 4000 (32 kbit/s) 4 – Up to 8000 (64 kbit/s) 5 – Up to 16000 (128 kbit/s) 6 – Up to 32000 (256 kbit/s) 7 – Up to 64000 (512 kbit/s) 8 – Up to 128000 (1024 kbit/s) 9 – Up to 256000 (2048 kbit/s)
<mean>	A numeric parameter which specifies the mean throughput class: 0 – network subscribed value 1 – 100 (~0.22 bit/s) 2 – 200 (~0.44 bit/s) 3 – 500 (~1.11 bit/s) 4 – 1000 (~2.2 bit/s) 5 – 2000 (~4.4 bit/s) 6 – 5000 (~11.1 bit/s) 7 – 10000 (~22 bit/s) 8 – 20000 (~44 bit/s) 9 – 50000 (~111 bit/s) 10 – 100000 (~0.22 kbit/s) 11 – 200000 (~0.44 kbit/s) 12 – 500000 (~1.11 kbit/s) 13 – 1000000 (~2.2 kbit/s) 14 – 2000000 (~4.4 kbit/s) 15 – 5000000 (~11.1 kbit/s) 16 – 10000000 (~22 kbit/s) 17 – 20000000 (~44 kbit/s) 18 – 50000000 (~111 kbit/s) 31 – optimization

Example

```
AT+CGQMIN=?
+CGQMIN: "IP",(0-3),(0-4),(0-5),(0-9),(0-18,31)
+CGQMIN: "PPP",(0-3),(0-4),(0-5),(0-9),(0-18,31)
+CGQMIN: "IPV6",(0-3),(0-4),(0-5),(0-9),(0-18,31)
+CGQMIN:
"IPV4V6",(0-3),(0-4),(0-5),(0-9),(0-18,31)
```

```
OK
AT+CGQMIN?
+CGQMIN:

OK
```

8.2.10 AT+CGEQMIN 3G quality of service profile (minimum acceptable)

The test command returns values supported as a compound value.

The read command returns the current settings for each defined context for which a QOS was explicitly specified.

AT+CGEQMIN 3G quality of service profile (minimum acceptable)	
Test Command **AT+CGEQMIN=?**	Response **+CGEQMIN: <PDP_type>,(list of supported <Traffic class>s),(list of supported <Maximum bitrate UL>s),(list of supported <Maximum bitrate DL>s),(list of supported <Guaranteed bitrate UL>s,(list of supported <Guaranteed bitrate DL>s),(list of supported <Delivery order>s),(list of supported <Maximum SDU size>s),(list of supported <SDU error ratio>s),(list of supported <Residual bit error** **Ratio>s),(list of supported <Delivery of erroneous SDUs>s),(list of** **Supported <Transfer delay>s),(list of supported <Traffic handling priority>s)** **OK** or **ERROR**
Read Command **AT+CGEQMIN?**	Response **+CGEQMIN: [<cid>,<Traffic class>,<Maximum bitrate UL>,<Maximum bitrate DL>,<Guaranteed bitrate UL>,<Guaranteed bitrate DL>,<Delivery order>,<Maximum SDU size>,<SDU error ratio>,** **<Residual bit error ratio>,<Delivery of erroneous SDUs>,<Transfer** **Delay>,<Traffic handling priority>][<CR><LF>** **+CGEQMIN: <cid>,<Traffic class>,<Maximum bitrate UL>,<Maximum bitrate DL>,<Guaranteed bitrate UL>,<Guaranteed bitrate DL>,<Delivery order>,<Maximum SDU size>,<SDU error ratio>,** **<Residual bit error ratio>,<Delivery of erroneous SDUs>,<Transfer**

	Delay>,<Traffic handling priority>[…]] **OK** or **ERROR**
Write Command **AT+CGEQMIN=<cid>[,<Traffic class>[,<Maximum bitrate UL>[,<Maximum bitrate DL>[,<Guaranteed bitrateUL>[,<Guaranteed bitrate DL>[,<Delivery order>[,<Maximum SDU size>[,<SDU error ratio>[,<Residual bit error ratio>[,<Delivery of erroneous SDUs>[,<Transfer delay>[,<Traffic handling priority>]]]]]]]]]]]]**	Response **OK** or **ERROR** or **+CME ERROR: <err>**
Execution Command **AT+ CGEQMIN**	Response **OK** or **ERROR**

Defined Values

<cid>	Parameter specifies a particular PDP context definition.The parameter is also used in other PDP context-related commands. The range is from 1 to 24,100 to 179.
<Traffic class>	0 – conversational 1 – streaming 2 – interactive 3 – background 4 – subscribed value
<Maximum bitrate UL>	This parameter indicates the maximum number of kbits/s delivered to UMTS(up-link traffic)at a SAP.As an example a bitrate of 32kbit/s would be specified as 32(e.g. AT+CGEQMIN=…,32,…). The range is from 0 to 8460. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<Maximum bitrate DL>	This parameter indicates the maximum number of kbits/s delivered to UMTS(down-link traffic)at a SAP.As an example a bitrate of 32kbit/s would be specified as 32(e.g. AT+CGEQMIN=…,32,…). The range is from 0 to 8460. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.

<Guaranteed bitrate UL>	This parameter indicates the guaranteed number of kbit/s delivered to UMTS(up-link traffic)at a SAP(provided that there is data to deliver).As an example a bitrate of 32kbit/s would be specified as 32(e.g.AT+CGEQMIN=…,32,…). The range is from 0 to 8460. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<Guaranteed bitrate DL>	This parameter indicates the guaranteed number of kbit/s delivered to UMTS(down-link traffic)at a SAP(provided that there is data to deliver).As an example a bitrate of 32kbit/s would be specified as 32(e.g.AT+CGEQMIN=…,32,…). The range is from 0 to 8460. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<Delivery order>	This parameter indicates whether the UMTS bearer shall provide in-sequence SDU delivery or not. 0 – no 1 – yes 2 – subscribed value
<Maximum SDU size>	This parameter indicates the maximum allowed SDU size inoctets. The range is from 0 to 1520. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<SDU error ratio>	This parameter indicates the target value for the fraction of SDUs lost or detected as erroneous.SDU error ratio is defined only for conforming traffic.As an example a target SDU error ratio of 5*10-3 would be specified as “5E3”(e.g.AT+CGEQMIN=..,”5E3”,…). “0E0” – subscribed value “1E2” “7E3” “1E3” “1E4” “1E5” “1E6” “1E1”
<Residual bit error ratio>	This parameter indicates the target value for the undetected bit error ratio in the delivered SDUs. If no error detection is requested,Residual bit error ratio indicates the bit error ratio in the delivered SDUs.As an example a target residual bit error ratio of 5*10-3 would be specified as “5E3”(e.g. AT+CGEQMIN=…,”5E3”,..).

	“0E0” – subscribed value “5E2” “1E2” “5E3” “4E3” “1E3” “1E4” “1E5” “1E6” “6E8”
<Delivery of erroneous SDUs>	This parameter indicates whether SDUs detected as erroneous shall be delivered or not. 0 – no 1 – yes 2 – no detect 3 – subscribed value
<Transfer delay>	This parameter indicates the targeted time between request to transfer an SDU at one SAP to its delivery at the other SAP,in milliseconds. The range is from 0 to 4000. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<Traffic handling priority>	This parameter specifies the relative importance for handling of all SDUs belonging to the UMTS Bearer compared to the SDUs of the other bearers. The range is from 0 to 3. The default value is 0. If the parameter is set to '0' the subscribed value will be requested.
<PDP_type>	(Packet Data Protocol type) a string parameter which specifies the type of packet data protocol. IP Internet Protocol PPP Point to Point Protocol IPV6 Internet Protocol Version 6 IPV4V6 Dual PDN Stack

Example

```
AT+CGEQMIN=?
+CGEQMIN:
"IP",(0-4),(0-11520),(0-42200),(0-11520),(0-42200),(0-2),(0-1520),("0E0","1E1","1E2","7E3","1E3","
```

```
1E4","1E5","1E6"),("0E0","5E2","1E2","5E3","4E3","1E3","1E4","1E5","1E6","6E8"),(0-3),(0,100-
4000),(0-3),(0-1),(0-1)
+CGEQMIN:
"PPP",(0-4),(0-11520),(0-42200),(0-11520),(0-42200),(0-2),(0-1520),("0E0","1E1","1E2","7E3","1E3
","1E4","1E5","1E6"),("0E0","5E2","1E2","5E3","4E3","1E3","1E4","1E5","1E6","6E8"),(0-3),(0,1
00-4000),(0-3),(0-1),(0-1)
+CGEQMIN:
"IPV6",(0-4),(0-11520),(0-42200),(0-11520),(0-42200),(0-2),(0-1520),("0E0","1E1","1E2","7E3","1E3
","1E4","1E5","1E6"),("0E0","5E2","1E2","5E3","4E3","1E3","1E4","1E5","1E6","6E8"),(0-3),(0,1
00-4000),(0-3),(0-1),(0-1)
+CGEQMIN:
"IPV4V6",(0-4),(0-11520),(0-42200),(0-11520),(0-42200),(0-2),(0-1520),("0E0","1E1","1E2","7E3","1
E3","1E4","1E5","1E6"),("0E0","5E2","1E2","5E3","4E3","1E3","1E4","1E5","1E6","6E8"),(0-3),(0
,100-4000),(0-3),(0-1),(0-1)

OK
AT+CGEQMIN?
+CGEQMIN:

OK
```

8.2.11 AT+CGDATA Enter data state

The command causes the MT to perform whatever actions are necessary to establish communication between the TE and the network using one or more Packet Domain PDP types. This may include performing a PS attach and one or more PDP context activations.

AT+CGDATA Enter data state	
Test Command **AT+ CGDATA=?**	Response **+ CGDATA: (list of supported <L2P>s)** **OK** or **ERROR**
Write Command **AT+CGDATA=[<L2P>,[<cid>]]**	Response **NO CARRIER** or **OK** or **ERROR** or **+CME ERROR: <err>**

Defined Values

<L2P>	A string parameter that indicates the layer 2 protocol to be used between the TE and MT. PPP Point-to-point protocol for a PDP such as IP
<text>	CONNECT result code string; the string formats please refer ATX/AT\V/AT&E command.
<cid>	A numeric parameter which specifies a particular PDP context definition (see AT+CGDCONT command). 1…16

Example

```
AT+CGDATA=?
+CGDATA: ("PPP")

OK
AT+CGDATA="PPP",1
CONNECT 115200
```

8.2.12 AT+CGPADDR Show PDP address

The write command returns a list of PDP addresses for the specified context identifiers.

AT+CGPADDR Show PDP address	
Test Command **AT+CGPADDR=?**	Response [+CGPADDR: (list of defined **<cid>**s)] **OK** or **ERROR**
Write Command **AT+CGPADDR=<cid>[,<cid>[,…]]**	Response **OK** or **ERROR** or **+CME ERROR: <err>**
Execution Command **AT+CGPADDR**	Response **[+CGPADDR: <cid>,<PDP_addr>]**

	+CGPADDR: <cid>,<PDP_addr>[...]]] **OK** or **ERROR** or **+CME ERROR: <err>**

Defined Values

<cid>	A numeric parameter which specifies a particular PDP context definition (see AT+CGDCONT command). If no <cid> is specified, the addresses for all defined contexts are returned. 1…24,100…179
<PDP_addr>	A string that identifies the MT in the address space applicable to the PDP. The address may be static or dynamic. For a static address, it will be the one set by the AT+CGDCONT command when the context was defined. For a dynamic address it will be the one assigned during the last PDP context activation that used the context definition referred to by <cid>. <PDP_addr> is omitted if none is available.

Example

```
AT+CGPADDR=?
+CGPADDR: (1)

OK
AT+CGPADDR=1
+CGPADDR: 1,"0.0.0.0"

OK
```

8.2.13 AT+CGCLASS GPRS network registration status

This command is used to set the MT to operate according to the specified GPRS mobile class.

AT+CGCLASS GPRS network registration status	
Test Command **AT+CGCLASS=?**	Response **+CGCLASS:(list of supported <class>s)** **OK**

	or **ERROR**
Read Command **AT+CGCLASS?**	Response **+CGCLASS: <class>** **OK** or **ERROR**
Write Command **AT+CGCLASS=<class>**	Response **OK** or **ERROR** or **+CME ERROR: <err>**
Execution Command Set default value: **AT+CGCLASS**	Response **OK** or **ERROR**

Defined Values

<class>	A string parameter which indicates the GPRS mobile class (in descending order of functionality) A – class A (highest)

Example

```
AT+CGCLASS=?
+CGCLASS: ("A")

OK
AT+CGCLASS?
+CGCLASS: "A"

OK
```

8.2.14 AT+CGEREP GPRS event reporting

The write command enables or disables sending of unsolicited result codes, "+CGEV" from MT to TE in the case of certain events occurring in the Packet Domain MT or the network. <mode> controls the processing of unsolicited result codes specified within this command. <bfr> controls the effect on buffered codes when <mode> 1 or 2 is entered. If a setting is not supported by the MT, ERROR or +CME ERROR: is returned.

Read command returns the current <mode> and buffer settings.

Test command returns the modes and buffer settings supported by the MT as compound values.

AT+CGEREP GPRS event reporting	
Test Command **AT+CGEREP=?**	Response **+CGEREP: (list of supported <mode>s), (list of supported <bfr>s)** **OK** or **ERROR**
Read Command **AT+CGEREP?**	Response **+CGEREP: <mode>,<bfr>** **OK** or **ERROR**
Write Command **AT+CGEREP=<mode>[,<bfr>]**	Response **OK** or **ERROR** or **+CME ERROR: <err>**
Execution Command **AT+CGEREP**	Response **OK** or **ERROR**

Defined Values

<n>	0 – disable network registration unsolicited result code 1 – enable network registration unsolicited result code +CGREG: <stat> 2 – there is a change in the ME network registration status or a change of the network cell: +CGREG: <stat>[,<lac>,<ci>]
<stat>	0 – not registered, ME is not currently searching an operator to register to 1 – registered, home network 2 – not registered, but ME is currently trying to attach or searching an operator to register to 3 – registration denied 4 – unknown 5 – registered, roaming
<lac>	Two bytes location area code in hexadecimal format (e.g.”00C3” equals 193 in decimal).

<ci>	Cell ID in hexadecimal format. GSM : Maximum is two byte WCDMA : Maximum is four byte TDS-CDMA : Maximum is four byte

NOTE

The **<lac>** not supported in CDMA/HDR mode
The **<ci>** not supported in CDMA/HDR mode

Example

```
AT+CGREG=?
+CGREG: (0-2)

OK
AT+CGREG?
+CGREG: 0,1

OK
```

8.2.15 AT+CGAUTH Set type of authentication for PDP-IP connections of GPRS

This command is used to set type of authentication for PDP-IP connections of GPRS.

AT+CGAUTH Set type of authentication for PDP-IP connections of GPRS	
Test Command **AT+CGAUTH=?**	Response **+CGAUTH: ,,127,127(for CDMA1x-EvDo only)** **+CGAUTH: (range of supported<cid>s),(list of supported <auth_type>s),,** **OK** or **ERROR** or **+CME ERROR: <err>**
Read Command **AT+CGAUTH?**	Response **+CGAUTH: <cid>,<auth_type>[,<user>]<CR><LF>** **+CGAUTH: <cid>,<auth_type>[,<user>]<CR><LF>**

	... **OK** **OK** or **ERROR** or **+CME ERROR: <err>**
Write Command **AT+CGAUTH=<cid>[,<auth_type>[,<passwd>[,<user>]]]**	Response **OK** or **ERROR** or **+CME ERROR: <err>**
Execution Command **AT+CGAUTH**	Response **OK** or **ERROR** or **+CME ERROR: <err>**

Defined Values

<cid>	Parameter specifies a particular PDP context definition. This is also used in other PDP context-related commands. 1…42,100…179
<auth_type>	Indicate the type of authentication to be used for the specified context. If CHAP is selected another parameter <passwd> needs to be specified. If PAP is selected two additional parameters <passwd> and <user> need to specified. 0 – none 1 – PAP 2 – CHAP 3 – PAP or CHAP
<passwd>	Parameter specifies the password used for authentication.
<user>	Parameter specifies the user name used for authentication.

Example

```
AT+CGAUTH=?
+CGAUTH: ,,127,127(for CDMA1x-EvDo only)
+CGAUTH: (1-24,100-179),(0-3),127,127

OK
```

```
AT+CGAUTH=1,1,"123","SIMCOM"
OK
```

9. AT Commands for SMS

9.1 Overview of AT Commands for SMS Control

Command	Description
AT+CSMS	Select message service
AT+CPMS	Preferred message storage
AT+CMGF	Select bearer service type
AT+CSCA	SMS service centre address
AT+CSCB	Select cell broadcast message indication
AT+CSMP	Set text mode parameters
AT+CSDH	Show text mode parameters
AT+CNMA	New message acknowledgement to ME/TA
AT+CNMI	New message indications to TE
AT+CGSMS	Select service for MO SMS messages
AT+CMGL	List SMS messages from preferred store
AT+CMGR	Read message
AT+CMGS	Send message
AT+CMSS	Send message from storage
AT+CMGW	Write message to memory
AT+CMGD	Delete message
AT+CMGMT	Change message status
AT+CMVP	Set message valid period
AT+CMGRD	Read and delete message
AT+CMGSEX	Send message
AT+CMSSEX	Send multi messages from storage
AT+CMGP	Set cdma/evdo text mode parameters

9.2 Detailed Description of AT Commands for SMS Control

9.2.1 AT+CSMS Select message service

This command is used to select messaging service <service>.
Note: This command not support in CDMA/EVDO mode

AT+CSMS Select message service	
Test Command **AT+CSMS=?**	Response a)If start HTTP service successfully: **+CSMS: (list of supported <service>s)** **OK** b)If failed: **ERROR**
Read Command **AT+CSMS?**	Response **+CSMS: <service>,<mt>,<mo>,<bm>** **OK**
Write Command **AT+CSMS=<service>**	Response a) **+CSMS: <mt>,<mo>,<bm>** **OK** b)If failed: **+CMS ERROR: <err>**

Defined Values

<service>	0 – SMS at command is compatible with GSM phase 2. 1 – SMS at command is compatible with GSM phase 2+.
<mt>	Mobile terminated messages: 0 – type not supported. 1 – type supported.
<mo>	Mobile originated messages: 0 – type not supported. 1 – type supported1 – SMS at command is compatible with GSM phase 2+.
<bm>	Broadcast type messages: 0 – type not supported. 1 – type supported.

Example

```
AT+CSMS=0
+CSMS:1,1,1
OK
```

9.2.2 AT+CPMS Preferred message storage

This command is used to select memory storages <mem1>, <mem2> and <mem3> to be used for reading, writing, etc.

AT+CPMS Preferred message storage	
Test Command **AT+CPMS=?**	Response a)If start HTTP service successfully: **+CPMS: (list of supported <mem1>s), (list of supported <mem2>s), (list of supported <mem3>s)** **OK** b)If failed: **ERROR**
Read Command **AT+CPMS?**	Response **+CPMS:<mem1>,<used1>,<total1>,<mem2>,<used2>,<total2>,<mem3>,<used3>,<total3>** **OK**
Write Command **AT+CPMS=<mem1>[,<mem2>[,<mem3>]]**	Response a) **+CPMS: <used1>,<total1>,<used2>,<total2>,<used3>,<total3>** **OK** b)If failed: **+CMS ERROR: <err>**

Defined Values

<mem1>	String type, memory from which messages are read and deleted (commands List Messages AT+CMGL, Read Message AT+CMGR and Delete Message AT+CMGD). "ME" and "MT" FLASH message storage "SM" SIM message storage "SR" Status report storage (not used in CDMA/EVDO mode)

<mem2>	String type, memory to which writing and sending operations are made (commands Send Message from Storage AT+CMSS and Write Message to Memory AT+CMGW). "ME" and "MT" FLASH message storage "SM" SIM message storage
<mem3>	String type, memory to which received SMS is preferred to be stored (unless forwarded directly to TE; refer command New Message Indications AT+CNMI). "ME" FLASH message storage "SM" SIM message storage GSM phase 2+.
<usedX>	Integer type, number of messages currently in <memX>.
<totalX>	Integer type, total number of message locations in <memX>.

Example

```
AT+CSMS=0
+CSMS:1,1,1
OK
```

9.2.3 AT+CMGF Select SMS message format

This command is used to specify the input and output format of the short messages.

AT+CMGF Select SMS message format	
Test Command **AT+CMGF=?**	Response a)If start HTTP service successfully: **+CMGF: (list of supported <mode>s)** **OK** b)If failed: **ERROR**
Read Command **AT+CMGF?**	Response **+CMGF: <mode>** **OK**
Write Command **AT+CMGF=<mode>**	Response a) **OK** b)If failed: **ERROR**
Execution Command	Response

AT+CMGF	a) **OK** b)If failed: **ERROR**

Defined Values

<mode>	0 – PDU mode 1 – Text mode

Example

```
AT+CMGF=1
OK
```

9.2.4 AT+CSCA SMS service centre address

This command is used to update the SMSC address, through which mobile originated SMS are transmitted.
Note: This command not support in CDMA/EVDO mode

AT+CSCA SMS service centre address	
Test Command **AT+CSCA=?**	Response a)If start HTTP service successfully: **+CMGF: (list of supported <mode>s)** **OK** b)If failed: **ERROR**
Read Command **AT+CSCA?**	Response **+CMGF: <mode>** **OK**
Write Command **AT+CSCA=<sca>[,<tosca>]**	Response a) **OK** b)If failed: **ERROR**

Defined Values

<mode>	0 – PDU mode

	1 – Text mode

Example

```
AT+CMGF=1
OK
```

9.2.5 AT+CSCB Select cell broadcast message indication

The test command returns the supported <mode>s as a compound value.
The read command displays the accepted message types.
Depending on the <mode> parameter, the write command adds or deletes the message types accepted.
Note: This command not support in CDMA/EVDO mode

AT+CSCB Select cell broadcast message indication	
Test Command **AT+CSCB=?**	Response a)If start HTTP service successfully: **+CSCB: (list of supported <mode>s)** **OK** b)If failed: **ERROR**
Read Command **AT+CSCB?**	Response **+CSCB: <mode>,<mids>,<dcss>** **OK** b)If failed: **ERROR**
Write Command **AT+CSCB=<mode>[,<mids>[,<dcss>]]**	Response a) **OK** b)If failed: **ERROR** c)If failed: **+CMS ERROR: <err>**

Defined Values

<mode>	0 – message types specified in <mids> and <dcss> are accepted. 1 – message types specified in <mids> and <dcss> are not

	accepted.
<mids>	String type; all different possible combinations of CBM message identifiers.
<dcss>	String type; all different possible combinations of CBM data coding schemes(default is empty string)

Example

```
AT+CSCB=?
+CSCB: (0-1)
OK
```

9.2.6 AT+CSMP Set text mode parameters

This command is used to select values for additional parameters needed when SM is sent to the network or placed in storage when text format message mode is selected.
Note: This command not support in CDMA/EVDO mode

AT+CSMP Set text mode parameters	
Test Command **AT+CSMP=?**	Response **OK**
Read Command **AT+CSMP?**	Response **+CSMP: <fo>,<vp>,<pid>,<dcs>** **OK**
Write Command **AT+CSMP=[<fo>[,<vp>[,<pid>[,<dcs>]]]]**	Response a) **OK** b)If failed: **ERROR**

Defined Values

<fo>	Depending on the Command or result code: first octet of GSM 03.40 SMS-DELIVER, SMS-SUBMIT (default 17), SMS-STATUS-REPORT, or SMS-COMMAND (default 2) in integer format. SMS status report is supported under text mode if <fo> is set to 49.
<vp>	Depending on SMS-SUBMIT <fo> setting: GSM 03.40,TP-Validity-Period either in integer format (default 167), in time-string format, or if is supported, in enhanced format (hexadecimal coded string with quotes), (<vp> is in range 0... 255).

<pid>	GSM 03.40 TP-Protocol-Identifier in integer format (default 0).
<dcs>	GSM 03.38 SMS Data Coding Scheme (default 0), or Cell Broadcast Data Coding Scheme in integer format depending on the command or result code.

Example

```
AT+CSMP=17,23,64,244
OK
```

9.2.7 AT+CSDH Show text mode parameters

This command is used to select values for additional parameters needed when SM is sent to the network or placed in storage when text format message mode is selected.
Note: This command not support in CDMA/EVDO mode

AT+CSDH Show text mode parameters	
Test Command **AT+CSDH=?**	Response a) **+CSDH: (list of supported <show>s)** **OK** b)If failed: **ERROR**
Read Command **AT+CSDH?**	Response **+CSDH: <show>** **OK**
Write Command **AT+CSDH=<show>**	Response a) **OK** b)If failed: **ERROR**
Execution Command **AT+CSDH**	Response a) Set default value (<show>=0): **OK** b)If failed: **ERROR**

Defined Values

<fo>	Depending on the Command or result code: first octet of GSM 03.40 SMS-DELIVER, SMS-SUBMIT (default 17),

	SMS-STATUS-REPORT, or SMS-COMMAND (default 2) in integer format. SMS status report is supported under text mode if <fo> is set to 49.
<vp>	Depending on SMS-SUBMIT <fo> setting: GSM 03.40,TP-Validity-Period either in integer format (default 167), in time-string format, or if is supported, in enhanced format (hexadecimal coded string with quotes), (<vp> is in range 0... 255).
<pid>	GSM 03.40 TP-Protocol-Identifier in integer format (default 0).
<dcs>	GSM 03.38 SMS Data Coding Scheme (default 0), or Cell Broadcast Data Coding Scheme in integer format depending on the command or result code.

Example

```
AT+CSDH=1
OK
```

9.2.8 AT+CNMA New message acknowledgement to ME/TA

This command is used to confirm successful receipt of a new message (SMS-DELIVER or SMS-STATUSREPORT) routed directly to the TE. If ME does not receive acknowledgement within required time (network timeout), it will send RP-ERROR to the network.

AT+CNMA New message acknowledgement to ME/TA	

Test Command
AT+CNMA=?
	Response
	if text mode(AT+CMGF=1):
	OK
	if PDU mode (AT+CMGF=0):
	+CNMA: (list of supported <n>s)
	OK

Write Command **AT+CNMA=<n>**	Response a) **OK** b)If failed: **ERROR** c)If failed: **+CMS ERROR: <err>**
Write Command **AT+CNMA**	Response a) **OK** b)If failed:

	ERROR c)If failed: **+CMS ERROR: <err>**

Defined Values

<n>	Parameter required only for PDU mode. 0 – Command operates similarly as execution command in text mode. 1 – Send positive (RP-ACK) acknowledgement to the network. Accepted only in PDU mode. 2 – Send negative (RP-ERROR) acknowledgement to the network. Accepted only in PDU mode.

Example

AT+CNMI=1,2,0,0,0
OK

+CMT:" 1380022xxxx" ,"" ," 02/04/03,11 :06 :38+32" <CR><LF>

Testing
(receive new short message)
AT+CNMA(send ACK to the network)
OK
AT+CNMA

+CMS ERROR : 340

(the second time return error, it needs ACK only once)

NOTE

- NOTE: The execute / write command shall only be used when AT+CSMS parameter <service> equals 1 (= phase 2+) and appropriate URC has been issued by the module, i.e.:
- <+CMT> for <mt>=2 incoming message classes 0, 1, 3 and none;
- <+CMT> for <mt>=3 incoming message classes 0 and 3;
- <+CDS> for <ds>=1.
- This command not support in CDMA/EVDO mode

9.2.9 AT+CNMI New message indications to TE

This command is used to select the procedure how receiving of new messages from the network is indicated to the TE when TE is active, e.g. DTR signal is ON. If TE is inactive (e.g. DTR signal is OFF). If set <mt> = 3 or <ds> = 1, make sure <mode> = 1, If set <mt>=2,make sure <mode>=1 or 2, otherwise it will return error.

AT+CNMI New message indications to TE	
Test Command **AT+CNMI=?**	Response **+CNMI: (list of supported <mode>s),(list of supported <mt>s),(list of supported <bm>s),(list of supported <ds>s),(list of supported <bfr>s)** **OK**
Read Command **AT+CNMI?**	Response **+CNMI: <mode>,<mt>,<bm>,<ds>,<bfr>** **OK**
Write Command **AT+CNMI=<mode>[,<mt>[,<bm>[,<ds> [,<bfr>]]]]**	Response a) **OK** b)If failed: **ERROR** c)If failed: **+CMS ERROR: <err>**
Execution Command **AT+CNMI**	Set default value:b) **OK**

Defined Values

<mode>	0 – Buffer unsolicited result codes in the TA. If TA result code buffer is full, indications can be buffered in some other place or the oldest indications may be discarded and replaced with the new received indications. 1 – Discard indication and reject new received message unsolicited result codes when TA-TE link is reserved (e.g. in on-line data mode). Otherwise forward them directly to the TE. 2 – Buffer unsolicited result codes in the TA when TA-TE link is reserved (e.g. in on-line data mode) and flush them to the TE after reservation. Otherwise forward them directly to the TE.
<mt>	The rules for storing received SMS depend on its data coding scheme, preferred memory storage (AT+CPMS) setting and this

	value: 0 – No SMS-DELIVER indications are routed to the TE. 1 – If SMS-DELIVER is stored into ME/TA, indication of the memory location is routed to the TE using unsolicited result code: +CMTI: <mem3>,<index>. 2 – SMS-DELIVERs (except class 2 messages and messages in the message waiting indication group (store message)) are routed directly to the TE using unsolicited result code: +CMT:[<alpha>],<length><CR><LF><pdu> (PDU mode enabled); or +CMT:<oa>,[<alpha>],<scts>[,<tooa>,<fo>,<pid>,<dcs>,<sca>,<tosca>,<length>] <CR> <LF><data> (text mode enabled, about parameters in italics, refer command Show Text Mode Parameters AT+CSDH). 3 – Class 3 SMS-DELIVERs are routed directly to TE using unsolicited result codes defined in <mt>=2. Messages of other data coding schemes result in indication as defined in <mt>=1.
<bm>	(not used in CDMA/EVDO mode) The rules for storing received CBMs depend on its data coding scheme, the setting of Select CBM Types (AT+CSCB) and this value: 0 – No CBM indications are routed to the TE. 2 – New CBMs are routed directly to the TE using unsolicited result code: +CBM: <length><CR><LF><pdu> (PDU mode enabled); or +CBM: <sn>,<mid>,<dcs>,<page>,<pages><CR><LF><data> (text mode enabled)
<ds>	(not used in CDMA/EVDO mode) 0 – No SMS-STATUS-REPORTs are routed to the TE. 1 – SMS-STATUS-REPORTs are routed to the TE using unsolicited result code: +CDS: <length><CR><LF><pdu> (PDU mode enabled); or +CDS: <fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st> (text mode

	enabled) 2 – If SMS-STATUS-REPORT is stored into ME/TA, indication of the memory location is routed to the TE using unsolicited result code: +CDSI: <mem3>,<index>.
<bfr>	0 – TA buffer of unsolicited result codes defined within this command is flushed to the TE when <mode> 1 to 2 is entered (OK response shall be given before flushing the codes). 1 – TA buffer of unsolicited result codes defined within this command is cleared when <mode> 1 to 2 is entered.

Example

```
AT+CNMI=2,1 (unsolicited result codes after received messages.)
OK
```

9.2.10 AT+CGSMS Select service for MO SMS messages

The write command is used to specify the service or service preference that the MT will use to send MO SMS messages.
The test command is used for requesting information on which services and service preferences can be set by using the AT+CGSMS write command
The read command returns the currently selected service or service preference.
Note: This command not support in CDMA/EVDO mode

AT+CGSMS Select service for MO SMS messages	
Test Command **AT+CGSMS=?**	Response **+CGSMS: (list of supported <service>s)** **OK**
Read Command **AT+CGSMS?**	Response **+CGSMS: <service>** **OK**
Write Command **AT+CGSMS=<service>**	Response a) **OK** b)If failed: **ERROR** c)If failed: **+CMS ERROR: <err>**

Defined Values

<service>	A numeric parameter which indicates the service or service preference to be used 0 – GPRS(value is not really supported and is internally mapped to 2) 1 – circuit switched(value is not really supported and is internally mapped to 3) 2 – GPRS preferred (use circuit switched if GPRS not available) 3 – circuit switched preferred (use GPRS if circuit switched not available)

Example

```
AT+CGSMS?
+CGSMS: 3
OK
```

9.2.11 AT+CMGL List SMS messages from preferred store

This command is used to return messages with status value <stat> from message storage <mem1> to the TE.
If the status of the message is 'received unread', the status in the storage changes to 'received read'.

AT+CMGL List SMS messages from preferred store	
Test Command **AT+CMGL=?**	Response **+CMGL: (list of supported <stat>s)** **OK**
Write Command **AT+CMGL=<stat>**	Response a)If text mode (AT+CMGF=1), command successful and SMS-SUBMITs and/or SMS-DELIVERs: **+CMGL:<index>,<stat>,<oa>/<da>,[<alpha>],[<scts>][,<tooa>/<toda>,<fo>,<pid>,<dcs>,<sca>,<tosca>,<length>]<CR><LF><data>[<CR><LF>** **+CMGL:<index>,<stat>,<oa>/<da>,[<alpha>],[<scts>][,<tooa>/<toda>,<fo>,<pid>,<dcs>,<sca>,<tosca>,<length>]<CR><LF><data>[...]]**

	OK b)If text mode (AT+CMGF=1), command successful and SMS-STATUS-REPORTs: **+CMGL:<index>,<stat>,<fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st>[<CR><LF>** **+CMGL:<index>,<stat>,<fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st>[...]]** **OK** c)If text mode (AT+CMGF=1), command successful and SMS-COMMANDs: **+CMGL: <index>,<stat>,<fo>,<ct>[<CR><LF>** **+CMGL: <index>,<stat>,<fo>,<ct>[...]]** **OK** d)If text mode (AT+CMGF=1), command successful and CBM storage: **+CMGL:<index>,<stat>,<sn>,<mid>,<page>,<pages>** **<CR><LF><data>[<CR><LF>** **+CMGL:<index>,<stat>,<sn>,<mid>,<page>,<pages>** **<CR><LF><data>[...]]** **OK** e)If PDU mode (AT+CMGF=0) and Command successful: **+CMGL:<index>,<stat>,[<alpha>],<length><CR><LF><pdu>[<CR><LF>** **+CMGL:<index>,<stat>,[<alpha>],<length><CR><LF><pdu>** **[...]]** **OK** f)If failed: **+CMS ERROR: <err>**

Defined Values

<stat>	1. Text Mode: "REC UNREAD" received unread message (i.e. new message) "REC READ" received read message "STO UNSENT" stored unsent message "STO SENT" stored sent message "ALL" all messages 2. PDU Mode: 0 – received unread message (i.e. new message) 1 – received read message 2 – stored unsent message

	3 – stored sent message 4 – all messages
<index>	Integer type; value in the range of location numbers supported by the associated memory and start with zero.
<oa>	Originating-Address, Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <tooa>.
<da>	Destination-Address, Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <toda>.
<alpha>	String type alphanumeric representation of <da> or <oa> corresponding to the entry found in MT phonebook; implementation of this feature is manufacturer specific; used character set should be the one selected with command Select TE Character Set AT+CSCS.
<scts>	TP-Service-Centre-Time-Stamp in time-string format (refer <dt>).
<da>	Destination-Address, Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <toda>.
<tooa>	TP-Originating-Address, Type-of-Address octet in integer format. (default refer <toda>).
<toda>	TP-Destination-Address, Type-of-Address octet in integer format. (when first character of <da> is + (IRA 43) default is 145, otherwise default is 129). The range of value is from 128 to 255.
<length>	Integer type value indicating in the text mode (AT+CMGF=1) the length of the message body <data> in characters; or in PDU mode (AT+CMGF=0), the length of the actual TP data unit in octets. (i.e. the RP layer SMSC address octets are not counted in the length)
<data>	In the case of SMS: TP-User-Data in text mode responses; format: 1. If <dcs> indicates that GSM 7 bit default alphabet is used and <fo> indicates that TP-User-Data-Header-Indication is not set: a. If TE character set other than "HEX": ME/TA converts GSM alphabet into current TE character set. b. If TE character set is "HEX": ME/TA converts each 7-bit character of GSM 7 bit default alphabet into two IRA character long hexadecimal numbers. (e.g. character (GSM 7 bit default alphabet 23) is presented as 17 (IRA 49 and 55)) 2. If <dcs> indicates that 8-bit or UCS2 data coding scheme is used, or <fo> indicates that TP-User-Data-Header-Indication is set: ME/TA converts each 8-bit octet into two IRA character long

	hexadecimal numbers. (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)) 3. If <dcs> indicates that GSM 7 bit default alphabet is used: a. If TE character set other than "HEX": ME/TA converts GSM alphabet into current TE character set. b. If TE character set is "HEX": ME/TA converts each 7-bit character of the GSM 7 bit default alphabet into two IRA character long hexadecimal numbers. 4. If <dcs> indicates that 8-bit or UCS2 data coding scheme is used: ME/TA converts each 8-bit octet into two IRA character long hexadecimal numbers.
<fo>	Depending on the command or result code: first octet of GSM 03.40 SMS-DELIVER, SMS-SUBMIT (default 17), SMS-STATUS-REPORT, or SMS-COMMAND (default 2) in integer format. SMS status report is supported under text mode if <fo> is set to 49.
<mr>	Message Reference GSM 03.40 TP-Message-Reference in integer format.
<ra>	Recipient Address GSM 03.40 TP-Recipient-Address Address-Value field in string format;BCD numbers (or GSM default alphabet characters) are converted to characters of the currently selected TE character set(refer to command AT+CSCS);type of address given by <tora>
<tora>	Type of Recipient Address GSM 04.11 TP-Recipient-Address Type-of-Address octet in integer format (default refer <toda>)
<dt>	Discharge Time GSM 03.40 TP-Discharge-Time in time-string format: ″yy/MM/dd,hh:mm:ss+zz″ ,where characters indicate year (two last digits),month,day,hour,minutes,seconds and time zone.
<st>	Status GSM 03.40 TP-Status in integer format 0...255
<ct>	Command Type GSM 03.40 TP-Command-Type in integer format 0...255
<sn>	Serial Number GSM 03.41 CBM Serial Number in integer format
<mid>	Message Identifier GSM 03.41 CBM Message Identifier in integer format
<page>	Page Parameter

	GSM 03.41 CBM Page Parameter bits 4-7 in integer format
<pages>	Page Parameter GSM 03.41 CBM Page Parameter bits 0-3 in integer format
<pdu>	In the case of SMS: SC address followed by TPDU in hexadecimal format: ME/TA converts each octet of TP data unit into two IRA character long hexadecimal numbers. (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)).

Example

```
AT+CMGL="ALL"
+CMGL: 1,"STO UNSENT","+10011",,,145,4
Hello World
OK
```

9.2.12 AT+CMGR Read message

This command is used to return message with location value <index> from message storage <mem1> to the TE.

AT+CMGR Read message	
Test Command **AT+CMGR=?**	Response **OK**
Write Command **AT+CMGR=<index>**	a)If text mode (AT+CMGF=1), command successful and SMS-DELIVER: **+CMGR: <stat>,<oa>,[<alpha>],<scts>[,<tooa>,<fo>,<pid>,<dcs>, <sca>, <tosca>, <length>]<CR><LF><data>** **OK** b)If text mode (AT+CMGF=1), command successful and SMS-SUBMIT: **+CMGR:<stat>,<da>,[<alpha>][,<toda>,<fo>,<pid>,<dcs>,[<vp>], <sca>, <tosca>,<length>]<CR><LF><data>** **OK** c)If text mode (AT+CMGF=1), command successful and SMS-STATUS-REPORT: **+CMGR: <stat>,<fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st>** **OK** d)If text mode (AT+CMGF=1), command successful and SMS-COMMAND: **+CMGR:<stat>,<fo>,<ct>[,<pid>,[<mn>],[<da>],[<toda>],<leng**

	th>]<CR><LF><data> **OK** e)If text mode (AT+CMGF=1), command successful and CBM storage: **+CMGR:<stat>,<sn>,<mid>,<dcs>,<page>,<pages><CR><LF><data>** **OK** f)If PDU mode (AT+CMGF=0) and Command successful: **+CMGR:<stat>,[<alpha>],<length><CR><LF><pdu>** **OK** g)If failed: **+CMS ERROR: <err>**
Execution Command **AT+CSDH**	Response a) Set default value (<show>=0): **OK** b)If failed: **ERROR**

Defined Values

<stat>	1. Text Mode: "REC UNREAD" received unread message (i.e. new message) "REC READ" received read message "STO UNSENT" stored unsent message "STO SENT" stored sent message "ALL" all messages 2. PDU Mode: 0 – received unread message (i.e. new message) 1 – received read message 2 – stored unsent message 3 – stored sent message 4 – all messages
<index>	Integer type; value in the range of location numbers supported by the associated memory and start with zero.
<oa>	Originating-Address, Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <tooa>.
<pid>	Protocol Identifier

	GSM 03.40 TP-Protocol-Identifier in integer format 0...255
<alpha>	String type alphanumeric representation of <da> or <oa> corresponding to the entry found in MT phonebook; implementation of this feature is manufacturer specific; used character set should be the one selected with command Select TE Character Set AT+CSCS.
<dcs>	Depending on the command or result code: SMS Data Coding Scheme (default 0), or Cell Broadcast Data Coding Scheme in integer format..
<sca>	RP SC address Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <tosca>.
<tosca>	RP SC address Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <tosca>.
<scts>	TP-Service-Centre-Time-Stamp in time-string format (refer <dt>).
<da>	Destination-Address, Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <toda>.
<tooa>	TP-Originating-Address, Type-of-Address octet in integer format. (default refer <toda>).
<toda>	TP-Destination-Address, Type-of-Address octet in integer format. (when first character of <da> is + (IRA 43) default is 145, otherwise default is 129). The range of value is from 128 to 255.
<length>	Integer type value indicating in the text mode (AT+CMGF=1) the length of the message body <data> in characters; or in PDU mode (AT+CMGF=0), the length of the actual TP data unit in octets. (i.e. the RP layer SMSC address octets are not counted in the length)
<data>	In the case of SMS: TP-User-Data in text mode responses; format: 1. If <dcs> indicates that GSM 7 bit default alphabet is used and <fo> indicates that TP-User-Data-Header-Indication is not set: a. If TE character set other than "HEX": ME/TA converts GSM alphabet into current TE character set. b. If TE character set is "HEX": ME/TA converts each 7-bit character of GSM 7 bit default alphabet into two IRA character long hexadecimal numbers. (e.g. character (GSM 7 bit default alphabet 23) is presented as 17 (IRA 49 and 55)) 2. If <dcs> indicates that 8-bit or UCS2 data coding scheme is used, or <fo> indicates that TP-User-Data-Header-Indication is set: ME/TA converts each 8-bit octet into two IRA character long

	hexadecimal numbers. (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)) 3. If <dcs> indicates that GSM 7 bit default alphabet is used: a. If TE character set other than "HEX": ME/TA converts GSM alphabet into current TE character set. b. If TE character set is "HEX": ME/TA converts each 7-bit character of the GSM 7 bit default alphabet into two IRA character long hexadecimal numbers. 4. If <dcs> indicates that 8-bit or UCS2 data coding scheme is used: ME/TA converts each 8-bit octet into two IRA character long hexadecimal numbers.
<fo>	Depending on the command or result code: first octet of GSM 03.40 SMS-DELIVER, SMS-SUBMIT (default 17), SMS-STATUS-REPORT, or SMS-COMMAND (default 2) in integer format. SMS status report is supported under text mode if <fo> is set to 49.
<vp>	Depending on SMS-SUBMIT <fo> setting: TP-Validity-Period either in integer format (default 167) or in time-string format (refer <dt>).
<mr>	Message Reference GSM 03.40 TP-Message-Reference in integer format.
<ra>	Recipient Address GSM 03.40 TP-Recipient-Address Address-Value field in string format;BCD numbers (or GSM default alphabet characters) are converted to characters of the currently selected TE character set(refer to command AT+CSCS);type of address given by <tora>
<tora>	Type of Recipient Address GSM 04.11 TP-Recipient-Address Type-of-Address octet in integer format (default refer <toda>)
<dt>	Discharge Time GSM 03.40 TP-Discharge-Time in time-string format: ″yy/MM/dd,hh:mm:ss+zz″,where characters indicate year (two last digits),month,day,hour,minutes,seconds and time zone.
<st>	Status GSM 03.40 TP-Status in integer format 0...255
<ct>	Command Type GSM 03.40 TP-Command-Type in integer format 0...255
<sn>	Serial Number GSM 03.41 CBM Serial Number in integer format

<mn>	Message Number GSM 03.40 TP-Message-Number in integer format
<mid>	Message Identifier GSM 03.41 CBM Message Identifier in integer format
<page>	Page Parameter GSM 03.41 CBM Page Parameter bits 4-7 in integer format
<pages>	Page Parameter GSM 03.41 CBM Page Parameter bits 0-3 in integer format
<pdu>	In the case of SMS: SC address followed by TPDU in hexadecimal format: ME/TA converts each octet of TP data unit into two IRA character long hexadecimal numbers. (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)).

Example

```
AT+CMGR=1
+CMGR: "STO UNSENT","+10011",,145,17,0,0,167,"+8613800100500",145,11
Hello World
OK
```

9.2.13 AT+CMGS Send message

This command is used to send message from a TE to the network (SMS-SUBMIT).

AT+CMGS Send message	
Test Command **AT+CMGS=?**	Response **OK**
Write Command If text mode (AT+CMGF=1): **AT+CMGS=<da>[,<toda>]<CR>Text is entered.** **<CTRL-Z/ESC>** If PDU mode(AT+CMGF=0): **AT+CMGS=<length><CR>** **PDU is entered** **<CTRL-Z/ESC>**	Response a)If sending successfully: **+CMGS: <mr>[,<time_stamp>]** **OK** b)If cancel sending: **OK** c)If sending fails: **ERROR** d)If sending fails: **+CMS ERROR: <err>**

Defined Values

<da>	Destination-Address, Address-Value field in string format; BCD

	numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <toda>.
<toda>	TP-Destination-Address, Type-of-Address octet in integer format. (when first character of <da> is + (IRA 43) default is 145, otherwise default is 129). The range of value is from 128 to 255.
<length>	integer type value indicating in the text mode (AT+CMGF=1) the length of the message body <data> > (or <cdata>) in characters; or in PDU mode (AT+CMGF=0), the length of the actual TP data unit in octets. (i.e. the RP layer SMSC address octets are not counted in the length)
<mr>	Message Reference GSM 03.40 TP-Message-Reference in integer format.

Example

```
AT+CMGS="13012832788"<CR>(TEXT MODE)
> ABCD<ctrl-Z/ESC>
+CMGS: 46
OK
```

NOTE

- NOTE: In text mode, the maximum length of an SMS depends on the used coding scheme: It is 160 characters if the 7 bit GSM coding scheme is used.

9.2.14 AT+CMSS Send message from storage

This command is used to send message with location value <index> from preferred message storage <mem2> to the network (SMS-SUBMIT or SMS-COMMAND).

AT+CMSS Send message from storage	
Test Command **AT+CMSS=?**	Response **OK**
Write Command **AT+CMSS=** **<index> [,<da>[,<toda>]]**	Response a) **+CMSS: <mr>[,<time_stamp>]** **OK** b)If failed: **ERROR**

	c)If sending fails: **+CMS ERROR: <err>**
Execution Command **AT+CSDH**	Response a) Set default value (<show>=0): **OK** b)If failed: **ERROR**

Defined Values

<index>	Integer type; value in the range of location numbers supported by the associated memory and start with zero.
<da>	Destination-Address, Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <toda>.
<mr>	Message Reference GSM 03.40 TP-Message-Reference in integer format.
<toda>	TP-Destination-Address, Type-of-Address octet in integer format. (when first character of <da> is + (IRA 43) default is 145, otherwise default is 129). The range of value is from 128 to 255.

Example

```
AT+CMSS=3
+CMSS: 0
OK
AT+CMSS=3,"13012345678"
+CMSS: 55
OK
```

NOTE

- NOTE: In text mode, the maximum length of an SMS depends on the used coding scheme: It is 160 characters if the 7 bit GSM coding scheme is used.

9.2.15 AT+CMGW Write message to memory

This command is used to store message (either SMS-DELIVER or SMS-SUBMIT) to memory storage <mem2>.

AT+CMGW Write message to memory	
Test Command **AT+CMGW=?**	Response **OK**
Write Command If text mode (AT+CMGF=1): **AT+CMGW=<oa>/<da>[,<tooa>/<toda>[,<stat>]]<CR>Text is entered.** **<CTRL-Z/ESC>** If PDU mode(AT+CMGF=0): **AT+CMGW=<length>[,<stat>]<CR>PDU is entered.** **<CTRL-Z/ESC>**	Response a)If write successfully: **+CMGW: <index>** **OK** b)If cancel write: **OK** c)If write fails: **ERROR** d)If write fails: **+CMS ERROR: <err>**

Defined Values

<index>	Integer type; value in the range of location numbers supported by the associated memory and start with zero.
<oa>	Originating-Address, Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <tooa>.
<tooa>	TP-Originating-Address, Type-of-Address octet in integer format. (default refer <toda>).
<da>	Destination-Address, Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <toda>.
<toda>	TP-Destination-Address, Type-of-Address octet in integer format. (when first character of <da> is + (IRA 43) default is 145, otherwise default is 129). The range of value is from 128 to 255.
<length>	Integer type value indicating in the text mode (AT+CMGF=1) the length of the message body <data> > (or <cdata>) in characters; or in PDU mode (AT+CMGF=0), the length of the actual TP data unit in octets. (i.e. the RP layer SMSC address octets are not counted in the length).
<stat>	1. Text Mode: "STO UNSENT" stored unsent message "STO SENT" stored sent message 2. PDU Mode: 2 – stored unsent message 3 – stored sent message

Example

```
AT+CMGW="13012832788" <CR> (TEXT MODE)
ABCD<ctrl-Z/ESC>
+CMGW:1
OK
```

NOTE

- NOTE: In text mode, the maximum length of an SMS depends on the used coding scheme: It is 160 characters if the 7 bit GSM coding scheme is used.

9.2.16 AT+CMGD Delete message

This command is used to delete message from preferred message storage <mem1> location <index>. If <delflag> is present and not set to 0 then the ME shall ignore <index> and follow the rules for <delflag> shown below.

AT+CMGD Delete message	
Test Command **AT+CMGD=?**	Response **+CMGD: (list of supported <index>s)[,(list of supported <delflag>s)]** **OK**
Write Command **AT+CMGD=<index>[,<delflag>]**	Response a) **OK** b)If failed: **ERROR** c)If failed: **+CMS ERROR: <err>**

Defined Values

<index>	Integer type; value in the range of location numbers supported by the associated memory and start with zero.
<delflag>	0 – (or omitted) Delete the message specified in <index>. 1 – Delete all read messages from preferred message storage,

	leaving unread messages and stored mobile originated messages (whether sent or not) untouched. 2 – Delete all read messages from preferred message storage and sent mobile originated messages, leaving unread messages and unsent mobile originated messages untouched. 3 – Delete all read messages from preferred message storage, sent and unsent mobile originated messages leaving unread messages untouched. 4 – Delete all messages from preferred message storage including unread messages.

Example

```
AT+CMGD=1
OK
```

NOTE

- NOTE: If set <delflag>=1, 2, 3 or 4, <index> is omitted, such as AT+CMGD=,1.

9.2.17 AT+CMGMT Change message status

This command is used to change the message status. If the status is unread, it will be changed read. Other statuses don't change.
Note: This command not support in CDMA/EVDO mode

AT+CMGMT Change message status	
Test Command **AT+CMGMT=?**	Response **OK**
Write Command **AT+CMGMT=<index>**	Response a) **OK** b)If failed: **ERROR** c)If failed: **+CMS ERROR: <err>**

Defined Values

<index>	Integer type; value in the range of location numbers supported by the associated memory and start with zero.

Example

```
AT+CMGMT=1
OK
```

9.2.18 AT+CMVP Set message valid period

This command is used to set valid period for sending short message.
Note: This command not support in CDMA/EVDO mode

AT+CMVP Set message valid period	
Test Command **AT+CMVP=?**	Response **+CMVP: (list of supported <vp>s)** **OK**
Read Command **AT+CMVP?**	Response **+CMVP:<vp>** **OK**
Write Command **AT+CMVP=<vp>**	Response a) **OK** b)If failed: **ERROR** c)If failed: **+CMS ERROR: <err>**

Defined Values

<vp>	Validity period value: 0 to 143 (<vp>+1) x 5 minutes (up to 12 hours) 144 to 167 12 hours + (<vp>-143) x 30 minutes 168 to 196 (<vp>-166) x 1 day 197 to 255 (<vp>-192) x 1 week

Example

```
AT+CMVP=167
OK
```

9.2.19 AT+CMGRD Read and delete message

This command is used to read message, and delete the message at the same time. It integrate AT+CMGR and AT+CMGD, but it doesn't change the message status.
Note: This command not support in CDMA/EVDO mode

AT+CMGRD Read and delete message	
Test Command **AT+CMGRD=?**	Response **OK**
Write Command **AT+CMGRD=<index>**	Response a)If text mode(AT+CMGF=1),command successful and SMS-DE-LIVER: **+CMGRD:<stat>,<oa>,[<alpha>],<scts>[,<tooa>,<fo>,<pid>,<dcs>, <sca>,<tosca>,<length>]<CR><LF><data>** **OK** b)If text mode(AT+CMGF=1),command successful and SMS-SU-BMIT: **+CMGRD:<stat>,<da>,[<alpha>][,<toda>,<fo>,<pid>,<dcs>,[<vp>], <sca>,<tosca>,<length>]<CR><LF><data>** **OK** c)If text mode(AT+CMGF=1),command successful and SMS-STA-TUS- REPORT: **+CMGRD: <stat>,<fo>,<mr>,[<ra>],[<tora>],<scts>,<dt>,<st>** **OK** d)If text mode(AT+CMGF=1),command successful and SMS-CO-MMAND: **+CMGRD:<stat>,<fo>,<ct>[,<pid>,[<mn>],[<da>],[<toda>],<length><CR><LF><data>]** **OK** e)If text mode(AT+CMGF=1),command successful and CBM sto-rage: **+CMGRD:<stat>,<sn>,<mid>,<dcs>,<page>,<pages><CR><LF><data>** **OK** f)**If PDU mode(AT+CMGF=0) and command successful:** **+CMGRD: <stat>,[<alpha>],<length><CR><LF><pdu>** **OK** **b)**If failed: **ERROR** **c)**If failed: **+CMS ERROR: <err>**

Defined Values

Refer to command AT+CMGR.

Example

```
AT+CMGRD=6
+CMGRD:"REC READ","+8613917787249",,"06/07/10,12:09:38+32",145,4,0,0, "+86138002105
00",145,4
How do you do
OK
```

9.2.20 AT+CMGSEX Send message

This command is used to send message from a TE to the network (SMS-SUBMIT).
Note: This command not support in CDMA/EVDO mode

AT+CMGSEX Send message	
Test Command **AT+CMGSEX=?**	Response **OK**
Write Command If text mode (AT+CMGF=1): **AT+CMGSEX=<da>[,<toda>][,<mr>,<msg_seg>,<msg_total>]<CR>Text is entered.<CTRL-Z/ESC>**	Response a)If sending successfully: **+CMGSEX: <mr>** **OK** b)If cancel sending: **OK** c)If sending fails: **ERROR** d)If sending fails: **+CMS ERROR: <err>**

Defined Values

<da>	Destination-Address, Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set, type of address given by <toda>.
<toda>	TP-Destination-Address, Type-of-Address octet in integer format. (When first character of <da> is + (IRA 43) default is 145, otherwise default is 129). The range of value is from 128 to 255.
<mr>	Message Reference GSM 03.40 TP-Message-Reference in integer format. The

	maximum length is 255.
<msg_seg>	The segment number for long sms
<msg_total>	The total number of the segments for long sms. Its range is from 2 to 255.

Example

```
AT+CMGSEX="13012832788", 190, 1, 2<CR>(TEXT MODE)
> ABCD<ctrl-Z/ESC>
+CMGSEX: 190
OK
AT+CMGSEX="13012832788", 190, 2, 2<CR>(TEXT MODE)
> ABCD<ctrl-Z/ESC>
+CMGSEX: 191
OK
```

NOTE

- NOTE: In text mode, the maximum length of an SMS depends on the used coding scheme: For single SMS, it is 160 characters if the 7 bit GSM coding scheme is used; For multiple long sms, it is 153 characters if the 7 bit GSM coding scheme is used.

9.2.21 AT+CMSSEX Send multi messages from storage

This command is used to send messages with location value <index1>,<index2>,<index3>… from preferred message storage <mem2> to the network (SMS-SUBMIT or SMS-COMMAND).The max count of index is 13 one time.
Note: This command not support in CDMA/EVDO mode

AT+CMSSEX Send multi messages from storage	
Test Command **AT+CMSSEX=?**	Response **OK**
Write Command **AT+CMSSEX=** **<index> [,<index >[,…]]**	Response a) **+CMSSEX: <mr>[,<mr>[,…]]** **OK** b)If failed: **ERROR** c)If sending fails:

	[+CMSSEX: <mr>[,<mr>[,...]]] **+CMS ERROR: <err>**

Defined Values

<index>	Integer type; value in the range of location numbers supported by the associated memory and start with zero.
<mr>	Message Reference GSM 03.40 TP-Message-Reference in integer format.
NOTE: In text mode, the maximum length of an SMS depends on the used coding scheme: It is 160 characters if the 7 bit GSM coding scheme is used.	

Example

```
AT+CMSSEX=0,1
+CMSSEX: 239,240
OK
```

NOTE

- NOTE: In text mode, the maximum length of an SMS depends on the used coding scheme: It is 160 characters if the 7 bit GSM coding scheme is used.

9.2.22 AT+CMGP Set cdma/evdo text mode parameters

The command is used to select values for additional parameters needed when SM is sent to the network or placed in storage when text format message mode is selected.
NOTE: take effect in CDMA/EVDO mode

AT+CMGP Set cdma/evdo text mode parameters	
Test Command **AT+CMGP=?**	Response **OK**
Read Command **AT+CMGP?**	Response **+CMGP: <tid>,<vpf>,<vp>,<ddtf>,<ddt>** **OK**
Write Command **AT+CMGP=[Tid][,<vpf>,<vp>[,<ddtf>,<ddt>]]**	Response **OK**

Defined Values

<tid>	Teleservice ID,value maybe 4095,4096,4097,4098,4099,4100,4101,4102 Default 4098
<vpf>	Valid Period Format 0, Absolute 1, Relative
<vp>	Valid Period "YY/MM/DD,HH/MM/SS" if vpf=0, Integer not exceed 248 if vpf=1
<ddtf>	Deferred Delivery Time Format 0, Absolute 1, Relative
<ddt>	Deferred Delivery Time "YY/MM/DD,HH/MM/SS" if ddtf=0, Integer not exceed 248 if ddtf=1

Example

```
AT+CMGP=4098,0,"11/04/22,16:21:00",1,12

OK
```

10. AT Commands for SSL

10.1 Overview of AT Commands for SSL

Command	Description
AT+CCHSTART	Start SSL service
AT+CCHSTOP	Stop SSL service
AT+CCHOPEN	Connect to SSL server
AT+CCHCLOSE	Disconnect from SSL server
AT+CCHSEND	Send data to SSL server
AT+CCHRECV	Read the cached data that received from the SSL server
AT+CCHCFG	Configure the client context
AT+CCHSSLCFG	Set the SSL context
AT+CCHSET	Configure the report mode of sending and receiving data
AT+CCHMODE	Configure the mode of sending and receiving data
AT+CCHADDR	Get the IPV4 address
AT+CSSLCFG	Configure the SSL context
AT+CCERTDOWN	Download certificate into the module
AT+CCERTLIST	List certificates
AT+CCERTDELE	Delete certificates

10.2 Detailed Description of AT Commands for SSL

10.2.1 AT+CCHSTART Start SSL service

AT+CCHSTART Start SSL service	
Execution Command **AT+CCHSTART**	Response a)If start SSL service successfully: **OK** **+CCHSTART: 0**

	b)If start SSL service successfully: **+CCHSTART: 0** **OK** c)If failed: **ERROR** d)If failed: **OK** **+CCHSTART: <err>**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	-

Defined Values

<err>	Integer type, which indicates the result code.

Example

```
AT+CCHSTART
OK

+CCHSTART: 0
```

NOTE

- You must execute AT+CCHSTART before any other SSL related operations

10.2.2 AT+CCHSTOP Stop SSL service

AT+CCHSTOP Stop SSL service	
Execution Command **AT+CCHSTOP**	Response a)If stop SSL service successfully: **+CCHSTOP: 0** **OK** b)If stop SSL service successfully: **OK** **+CCHSTOP: 0** c)If failed:

	ERROR
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<err>	Integer type, which indicates the result code.

Example

```
AT+CCHSTOP
OK

+CCHSTOP: 0
```

10.2.3 AT+CCHOPEN Connect to SSL server

AT+CCHOPEN Connect to SSL server	
Test Command **AT+CCHOPEN=?**	Response **+CCHOPEN: (0-1),"ADDRESS",(1-65535)[,(1-2)[,(1-65535)]]** **OK**
Read Command **AT+CCHOPEN?**	Response If connect to a server, it will show the connected information. Otherwise, the connected information is empty. **+CCHOPEN: 0,"<host>",<port>,<client_type>[,<bind_port>]** **+CCHOPEN: 1,"<host>",<port>,<client_type>[,<bind_port>]** **OK**
Write Command **AT+CCHOPEN=<session_id>,"host",<port>[,<client_type>[,<bind_port>]]**	Response a)If connect successfully: **+CCHOPEN: <session_id>,0** **OK** b)If connect successfully: **OK** **+CCHOPEN: <session_id>,0** c)If connect successfully in transparent mode: **CONNECT [<text>]** d)If failed:

	OK **+CCHOPEN: <session_id>,<err>** **[+CCHCLOSE: <session_id>,<err>]** e)If failed: **ERROR** f)If failed in transparent mode: **CONNECT FAIL**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<session_id>	The session index to operate. It's from 0 to 1.
<host>	The server address, length range is 1 to 256.
<port>	The server port which to be connected, the range is from 1 to 65535.
<client_type>	The type of client: 1 – TCP client. 2 – SSL/TLS client. Default value is 2.
<bind_port>	The local port for channel, the range is from 1 to 65535.
<text>	CONNECT result code string; the string formats please refer ATX/AT\V/AT&E command.
<err>	Integer type, the result of operation.0 is success, other value is failure.

Example

```
AT+CCHOPEN=0,"www.baidu.com",443,2
OK

+CCHOPEN: 0,0
```

NOTE

- If you don't set the SSL context by AT+CCHSSLCFG before connecting a SSL/TLS server by AT+CCHOPEN, it will use the <session_id> (the 1'st parameter of AT+CCHOPEN) SSL context when connecting to the server.

10.2.4 AT+CCHCLOSE Disconnect from SSL server

AT+CCHCLOSE Disconnect from SSL server	
Write Command **AT+CCHCLOSE=<session_id>**	Response a)If successfully： **+CCHCLOSE: <session_id>,0** **OK** b)If successfully： **OK** **+CCHCLOSE: <session_id>,0** c)If successfully in transparent mode： **OK** **CLOSED** d)If failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<session_id>	The session index to operate. It's from 0 to 1.
<err>	Integer type, the result of operation. 0 is success, other value is failure

Example

```
AT+CCHCLOSE=0
OK

+CCHCLOSE: 0,0
```

10.2.5 AT+CCHSEND Send data to SSL server

AT+CCHSEND Send data to SSL server	
Test Command **AT+CCHSEND=?**	Response **+CCHSEND: (0-1),(1-2048)** **OK**
Read Command **AT+CCHSEND?**	Response **+CCHSEND: 0,<unsent_len_0>,1,<unsent_len_1>**

	OK
Write Command **AT+CCHSEND=<session_id>,<len>**	Response a)if parameter is right: > **<input data here>** When the total size of the inputted data reaches <len>, TA will report the following code. Otherwise, the serial port will be blocked. **OK** b)If parameter is wrong or other errors occur: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<session_id>	The session index to operate. It's from 0 to 1.
<len>	The length of data to send. Its range is from 1 to 2048 bytes.
<unsent_len_0>	The data of connection 0 cached in sending buffer which is waiting to be sent.
<unsent_len_1>	The data of connection 1 cached in sending buffer which is waiting to be sent.

Example

```
AT+CCHSEND=0,125
> GET / HTTP/1.1
Host: www.google.com.hk
User-Agent: MAUI htp User Agent
Proxy-Connection: keep-alive
Content-Length: 0

OK
```

10.2.6 AT+CCHRECV Read the cached data that received from the server

AT+CCHRECV Read the cached data that received from the server	
Read Command **AT+CCHRECV?**	Response **+CCHRECV: LEN,<cache_len_0>,<cache_len_1>** **OK**

Write Command **AT+CCHRECV=<session_id>[,<max_recv_len>]**	Response a)if parameter is right and there are cached data: **OK** **[+CCHRECV: DATA,<session_id>,<len>** **…** **+CCHRECV: DATA,<session_id>,<len>** **…]** **+CCHRECV: <session_id>,<err>** b) if parameter is not right or any other error occurs: **+CCHRECV: <session_id>,<err>** **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<session_id>	The session_id to operate. It's from 0 to 1.
<max_recv_len>	Maximum bytes of data to receive in the current AT+CCHRECV calling. It will read all the received data when the value is greater than the length of RX data cached for session <session_id>. 0 means the maximum bytes to receive is 2048 bytes. (But, when 2048 is greater than the length of RX data cached for session <session_id>, 0 means the length of RX data cached for session <session_id>). The default value is the length of RX data cached for session <session_id>. It will be not allowed when there is no data in the cache.
<cache_len_0>	The length of RX data cached for connection 0.
<cache_len_1>	The length of RX data cached for connection 1.
<len>	The length of data followed.
<err>	String type, displays the cause of occurring error, please refer to Chapter 10.3 for details.

Example

AT+CCHRECV=1
OK
+CCHRECV: DATA,1,249
HTTP/1.1 200 OK
Content-Type: text/html
Content-Language: zh-CN
Content-Length: 57

```
Date: Tue, 31 Mar 2009 01:56:05 GMT
Connection: Close
Proxy-Connection: Close

<html>
<header>test</header>
<body>
Test body
</body>

+CCHRECV: 1,0
```

10.2.7 AT+CCHADDR Get IPV4 address

AT+CCHADDR Get IPV4 address	
Execution Command **AT+CCHADDR**	Response: **+CCHADDR: <ip_address>** **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<ip_address>	A string parameter that identifies the IPv4 address after PDP activated.

Example

```
AT+CCHADDR
+CCHADDR: 10.71.155.118

OK
```

10.2.8 AT+CCHCFG Configure the client context

AT+CCHCFG Configure the client context	
Test Command	Response

AT+CCHCFG=?	**+CCHCFG: "sendtimeout",(0-1),(60-150)** **+CCHCFG: "sslctx",(0-1),(0-9)** **OK**
Read Command **AT+CCHCFG?**	Response **+CCHCFG: 0,<sendtimeout_val>,<sslctx_index>** **+CCHCFG: 1,<sendtimeout_val>,<sslctx_index>** **OK**
Write Command /*Configure the timeout value of the specified client when sending data*/ **AT+CCHCFG="sendtimeout",<session_id>,<sendtimeout_val>**	Response If successfully: **OK** If failed: **ERROR**
Write Command /*Configure the SSL context index, it's as same as AT+CSSLCFG*/ **AT+CCHCFG="sslctx",<session_id>,<sslctx_index>**	Response If successfully: **OK** If failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<session_id>	The session_id to operate. It's from 0 to 1.
<sendtimeout_val>	The timeout value used in sending data stage. The range is 60-150 seconds. The default value is 150.
<sslctx_index>	The SSL context ID which will be used in the SSL connection. Refer to the <ssl_ctx_index> of **AT+CSSLCFG**.

Example

```
AT+CCHCFG="sendtimeout",0,60
OK
```

NOTE

- This command must be called before AT+CCHOPEN and after AT+CCHSTART. The setting will be cleared after AT+CCHOPEN failed or AT+CCHCLOSE.

10.2.9 AT+CCHSSLCFG Set the SSL context

AT+CCHSSLCFG Set the SSL context	
Test Command **AT+CCHSSLCFG=?**	Response **+CCHSSLCFG: (0-1),(0-9)** **OK**
Read Command **AT+CCHSSLCFG?**	Response **+CCHSSLCFG: <session_id>,[ssl_ctx_index]** **+CCHSSLCFG: <session_id>,[ssl_ctx_index]** **OK**
Write Command **AT+CCHSSLCFG=<session_id>,<ssl_ctx_index>**	Response a) If successfully: **OK** b) If failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<session_id>	The session_id to operate. It's from 0 to 1.
<ssl_ctx_index>	The SSL context ID which will be used in the SSL connection. Refer to the <ssl_ctx_index> of **AT+CSSLCFG**.

Example

```
AT+CCHSSLCFG=?
+CCHSSLCFG: (0-1),(0-9)

OK
AT+CCHSSLCFG=1,1
OK
```

NOTE

- This command must be called before AT+CCHOPEN and after AT+CCHSTART. The setting will be cleared after AT+CCHOPEN failed or AT+CCHCLOSE.
- If you don't set the SSL context by this command before connecting to SSL/TLS server by AT+CCHOPEN, the CCHOPEN operation will use the SSL context as same as index <session_id> (the 1st parameter of AT+CCHOPEN) when connecting to the server.

10.2.10 AT+CCHMODE Configure the mode of sending and receiving data

AT+CCHMODE Configure the mode of sending and receiving mode	
Test Command **AT+CCHMODE=?**	Response **+CCHMODE: (0-1)** **OK**
Read Command **AT+CCHMODE?**	Response **+CCHMODE: <mode>** **OK**
Write Command **AT+CCHMODE=<mode>**	Response a) If successfully: **OK** b) If failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<mode>	The mode value: 0 – Normal 1 – Transparent mode The default value is 0.

Example

```
AT+CCHMODE=?
+CCHMODE: (0-1)

OK
AT+CCHMODE=1
OK
```

NOTE

- This command must be called before AT+CCHSTART.
- There is only one session in the transparent mode, it's the first session.

10.2.11 AT+CCHSET Configure the report mode of sending and receiving data

AT+CCHSET Configure the report mode of sending and receiving data	
Test Command **AT+CCHSET=?**	Response **+CCHSET: (0-1),(0,1)** **OK**
Read Command **AT+CCHSET?**	Response **+CCHSET: <report_send_result>,<recv_mode>** **OK**
Write Command **AT+CCHSET=<report_send_result>[,<recv_mode>]**	Response a) If successfully: **OK** b) If failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<report_send_result>	Whether to report result of CCHSEND, the default value is 0: 0 – No. 1–Yes. Module will report +CCHSEND: <session_id>,<err> to MCU when complete sending data.
<recv_mode>	The receiving mode: 0 -- Output the data to MCU whenever received data. 1 -- Module caches the received data and notifies MCU with +CCHEVENT: <session_id>, RECV EVENT. MCU can use AT+CCHRECV to receive the cached data (only in manual receiving mode).

Example

```
AT+CCHSET=?
+CCHSET: (0-1),(0,1)

OK
AT+CCHSET=1,1
OK
```

NOTE

- This command must be called before AT+CCHSTART.

10.2.12 AT+CSSLCFG Configure the SSL context

AT+CSSLCFG Configure the SSL context	
Test Command **AT+CSSLCFG=?**	Response **+CSSLCFG: "sslversion",(0-9),(0-4)** **+CSSLCFG: "authmode",(0-9),(0-3)** **+CSSLCFG: "ignorelocaltime",(0-9),(0,1)** **+CSSLCFG: "negotiatetime",(0-9),(10-300)** **+CSSLCFG: "cacert",(0-9),(5-128)** **+CSSLCFG: "clientcert",(0-9),(5-128)** **+CSSLCFG: "clientkey",(0-9),(5-128)** **+CSSLCFG: "enableSNI",(0-9),(0,1)** **+CSSLCFG: "keypwd",(0-9),(0-128)** **+CSSLCFG: "ciphersuites",(0-9),(0x002F,0xFFFF)** **OK**
Read Command **AT+CSSLCFG=?**	Response **+CSSLCFG: 0,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enalbeSNI_flag>,<keypwd>,<ciphersuites>** **+CSSLCFG: 1,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enalbeSNI_flag>,<keypwd>,<ciphersuites>** **+CSSLCFG: 2,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enalbeSNI_flag>,<keypwd>,<ciphersuites>** **+CSSLCFG: 3,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enalbeSNI_flag>,<keypwd>,<ciphersuites>** **+CSSLCFG: 4,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enalbeSNI_flag>,<keypwd>,<ciphersuites>** **+CSSLCFG:**

	5,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enalbeSNI_flag>,<keypwd>,<ciphersuites> **+CSSLCFG: 6,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enalbeSNI_flag>,<keypwd>,<ciphersuites>** **+CSSLCFG: 7,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enalbeSNI_flag>,<keypwd>,<ciphersuites>** **+CSSLCFG: 8,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enalbeSNI_flag>,<keypwd>,<ciphersuites>** **+CSSLCFG: 9,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enalbeSNI_flag>,<keypwd>,<ciphersuites>** **OK**
Write Command /*Query the configuration of the specified SSL context*/ **AT+CSSLCFG=<ssl_ctx_index>**	Response **+CSSLCFG: <ssl_ctxindex>,<sslversion>,<authmode>,<ignoreltime>,<negotiatetime>,<ca_file>,<clientcert_file>,<clientkey_file>,<enalbeSNI_flag>,<keypwd>,<ciphersuites>** **OK**
Write Command /*Configure the version of the specified SSL context*/ **AT+CSSLCFG="sslversion",<ssl_ctx_index>,<sslversion>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the authentication of the specified SSL context*/ **AT+CSSLCFG="authmode",<ssl_ctx_index>,<authmode>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the ignore local time flag of the specified SSL context*/ **AT+CSSLCFG="ignorelocalti**	Response a)If successfully: **OK** b)If failed: **ERROR**

me",<ssl_ctx_index>,<ignore ltime>	
Write Command /*Configure the negotiate timeout value of the specified SSL context*/ **AT+CSSLCFG="negotiatetim e",<ssl_ctx_index>,<negotiat etime>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the server root CA of the specified SSL context*/ **AT+CSSLCFG="cacert",<ssl_ ctx_index>,<ca_file>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the client certificate of the specified SSL context*/ **AT+CSSLCFG="clientcert",<s sl_ctx_index>,<clientcert_file >**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the client key of the specified SSL context*/ **AT+CSSLCFG="clientkey",<s sl_ctx_index>,<clientkey_file >**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the enableSNI flag of the specified SSL context*/ **AT+CSSLCFG="enableSNI",< ssl_ctx_index>,<enableSNI_F lag>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the password of the specified SSL context*/ **AT+CSSLCFG="keypwd",<ss l_ctx_index>,<keypwd>**	Response a)If successfully: **OK** b)If failed: **ERROR**
Write Command /*Configure the ciphersuite of the specified SSL context*/	Response a)If successfully: **OK**

AT+CSSLCFG="ciphersuites",<ssl_ctx_index>,<ciphersuites>	b)If failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<ssl_ctx_index>	The SSL context ID. The range is 0-9.
<sslversion>	The SSL version, the default value is 4. 0 – SSL3.0 1 – TLS1.0 2 – TLS1.1 3 – TLS1.2 4 – All The configured version should be support by server. So you should use the default value if you can't confirm the version which the server supported.
<authmode>	The authentication mode, the default value is 0. 0 – no authentication. 1 –server authentication. It needs the root CA of the server. 2 –server and client authentication. It needs the root CA of the server, the cert and key of the client. 3 –client authentication and no server authentication. It needs the cert and key of the client.
<ignoreltime>	The flag to indicate how to deal with expired certificate, the default value is 1. 0 – care about time check for certification. 1 – ignore time check for certification When set the value to 0, it need to set the right current date and time by AT+CCLK when need SSL certification.
<negotiatetime>	The timeout value which is used in SSL negotiating stage. The range is 10-300 seconds. The default value is 300.
<ca_file>	The root CA file name of SSL context. The file name must have type like ".pem" or ".der". The length of filename is from 5 to 128 bytes. If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename's UTF8 code). There are two ways to download certificate files to module:

	1. By AT+CCERTDOWN. 2. By FTPS or HTTPS commands. Please refer to chapter 12 and 13.
<clientcert_file>	The client cert file name of SSL context. The file name must have type like ".pem" or ".der". The length of filename is from 5 to 128 bytes. If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename's UTF8 code). There are two ways to download certificate files to module: 1. By AT+CCERTDOWN. 2. By FTPS or HTTPS commands. Please refer to chapter 12 and 13.
<clientkey_file>	The client key file name of SSL context. The file name must have type like ".pem" or ".der". The length of filename is from 5 to 128 bytes. If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename's UTF8 code). There are two ways to download certificate files to module: 1. By AT+CCERTDOWN. 2. By FTPS or HTTPS commands. Please refer to chapter 12 and 13.
<enableSNI_flag>	The flag to indicate that enable the SNI flag or not, the default value is 0. 0 – not enable SNI. 1 – enable SNI.
<keypwd>	The password of the client key file of SSL context. When the client needs to be authorized, client key file is needed. Because the client key file may be encrypted, we need the <**keypwd**> to decrypt it. The length of <**keypwd**> is from 0 to 128 bytes.
<ciphersuites>	Numeric type, SSL ciphersuites 0x002F TLS_RSA_WITH_AES_128_CBC_SHA 0xFFFF Support all

Example

```
AT+CSSLCFG="sslversion",1,1
OK
```

10.2.13 AT+CCERTDOWN Download certificate into the module

AT+CCERTDOWN Download certificate into the module	
Test Command **AT+CCERTDOWN=?**	Response **+CCERTDOWN: (5-128),(1-10240)** **OK**
Write Command **AT+CCERTDOWN=<filename>,<len>**	Response a)If it can be download: > **<input data here>** **OK** b)If failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<filename>	The name of the certificate/key file. The file name must have type like ".pem" or ".der". The length of filename is from 5 to 128 bytes. If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename's UTF8 code). For example: If you want to download a file with name "中华.pem", you should convert the "中华.pem" to UTF8 coding (中华.pem), then input the hexadecimal (262378344532443B262378353334453B2E70656D) of UTF8 coding.
len>	The length of the file data to send. The range is from 1 to 10240 bytes.

Example

```
AT+CCERTDOWN="client_key.der",611
> file content……
OK
```

10.2.14 AT+CCERTLIST List certificates

AT+CCERTLIST List certificates	
Execution Command **AT+CCERTLIST**	Response **[+CCERTLIST: <file_name>**

	[+CCERTLIST: <file_name>] **…** **<CR><LF>]** **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<file_name>	The certificate/key files which has been downloaded to the module. If the filename contains non-ASCII characters, it will show the non-ASCII characters as UTF8 code.

Example

```
AT+CCERTLIST
+CCERTLIST: “ca_cert.der”
+CCERTLIST: “client_key.pem””

OK
```

10.2.15 AT+CCERTDELE Delete certificates

AT+CCERTDELE Delete certificates	
Write Command **AT+CCERTDELE=<filename>**	Response a)If delete successfully: **OK** b)If failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<filename>	The name of the certificate/key file. The file name must have type like “.pem” or “.der”. The length of filename is from 5 to 128 bytes. If the filename contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark (The string in the quotation mark should be hexadecimal of the filename’s

	UTF8 code). For example: If you want to download a file with name “中华.pem”, you should convert the “ 中 华 .pem” to UTF8 coding (中华.pem), then input the hexadecimal (262378344532443B262378353334453B2E70656D) of UTF8 coding.

Example

```
AT+CCERTDELE=”server_ca.der”
OK
```

10.3 Command result <err> codes

Result Code	
0	Operation succeeded
1	Alerting state(reserved)
2	Unknown error
3	Busy
4	Peer closed
5	Operation timeout
6	Transfer failed
7	Memory error
8	Invalid parameter
9	Network error
10	Open session error
11	State error
12	Create socket error
13	Get DNS error
14	Connect socket error
15	Handshake error
16	Close socket error
17	Nonet
18	Send data timeout
19	Not set certificates

10.4 Unsolicited result codes

Information	Description
+CCHEVENT: <session_id>,RECV EVENT	In manual receiving mode, when new data of a connection arriving to the module, this unsolicited result code will be reported to MCU.
+CCH_RECV_CLOSED: <session_id>,<err>	When receive data occurred any error, this unsolicited result code will be reported to MCU.
+CCH_PEER_CLOSED: <session_id>	The connection is closed by the server.

11. AT Commands for TCPIP

11.1 Overview of AT Commands for TCPIP

Command	Description
AT+NETOPEN	Start TCPIP service
AT+NETCLOSE	Stop TCPIP service
AT+CIPOPEN	Setup TCP/UDP client socket connection
AT+CIPCLOSE	Destroy TCP/UDP client socket connection
AT+CIPSEND	Send TCP/UDP data
AT+CIPRXGET	Retrieve TCP/UDP buffered data
AT+IPADDR	Get IP address of PDP context
AT+CIPHEAD	Add an IP header when receiving data
AT+CIPSRIP	Show remote IP address and port
AT+CIPMODE	Select TCP/IP application mode
AT+CIPSENDMOE	Set sending mode
AT+CIPTIMEOUT	Set TCP/IP timeout value
AT+CIPCCFG	Configure parameters of socket
AT+SERVERSTART	Startup TCP server
AT+SERVERSTOP	Stop TCP server
AT+CIPACK	Query TCP connection data transmitting status
AT+CDNSGIP	Query the IP address of given domain name
AT+CDNSGHNAME	Query the domain name of given IP address
AT+CIPDNSSET	Set DNS query parameters
AT+CPING	Ping destination address
AT+CPINGSTOP	Stop an ongoing ping session

11.2 Detailed Description of AT Commands for TCPIP

11.2.1 AT+NETOPEN Start TCPIP service

AT+NETOPEN Start TCPIP service	
Read Command **AT+NETOPEN?**	Response **+NETOPEN: <net_state>** **OK**
Execution Command **AT+NETOPEN**	Response If the PDP context has not been activated or the network closed abnormally, response: **OK** **+NETOPEN: <err>** when the PDP context has been activated successfully, if you execute AT+NETOPEN again, response: **+IP ERROR: Network is already opened** **ERROR** other: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	-

Defined Values

<net_state>	Integer type, which indicates the state of PDP context activation. 0 network close (deactivated) 1 network open(activated)
<err>	Integer type, the result of operation. 0 is success, other value is failure.

Example

```
AT+NETOPEN
OK

+NETOPEN: 0
AT+NETOPEN?
+NETOPEN: 1

OK
```

NOTE

- You must execute AT+NETOPEN before any other TCP/UDP related operations

11.2.2 AT+NETCLOSE Stop TCPIP service

AT+NETCLOSE Stop TCPIP service	
Execution Command **AT+NETCLOSE**	Response If the PDP context has been activated, response: **OK** **+NETCLOSE: <err>** If the PDP context has not been activated, response: **+NETCLOSE: <err>** **ERROR** other: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<err>	Integer type, the result of operation.0 is success, other value is failure.

Example

```
AT+NETCLOSE
OK

+NETCLOSE: 0
```

NOTE

- "AT+NETCLOSE" can close all the opened socket connections when you didn't close these connections by "AT+CIPCLOSE".

11.2.3 AT+CIPOPEN Setup TCP/UDP client socket connection

AT+CIPOPEN Setup TCP/UDP client socket connection

Test Command **AT+CIPOPEN=?**	Response **+CIPOPEN: (0-9),("TCP","UDP")** **OK**
Read Command **AT+CIPOPEN?**	Response **+CIPOPEN: <link_num> [,<type>,<serverIP>,<serverPort>,<index>]** **+CIPOPEN: <link_num> [,<type>,<serverIP>,<serverPort>,<index>]** **[…]** **OK** If a connection identified by <link_num>has not been established successfully, +CIPOPEN: <link_num> will be returned.
Write Command TCP connection **AT+CIPOPEN=<link_num>,"TCP",<serverIP>,<serverPort>,[,<localPort>]**	Response if PDP context has been activated successfully, response: **OK** **+CIPOPEN: <link_num>,<err>** when the <link_num> is greater than 10, response： **+IP ERROR: Invalid parameter** **ERROR** If PDP context has not been activated, or the connection has been established, or parameter is incorrect, or other errors, response: **+CIPOPEN: <link_num>,<err>** **ERROR** Transparent mode for TCP connection: When you want to use transparent mode to transmit data, you should set AT+CIPMODE=1 before AT+NETOPEN. And if AT+CIPMODE=1 is set, the <link_num> is restricted to be only 0. if success **CONNECT [<text>]** if failure **CONNECT FAIL** other: **ERROR**
Write Command UDP connection **AT+CIPOPEN=<link_num>,"UDP",,,<localPort>**	if PDP context has been activated successfully, response: **+CIPOPEN: <link_num>,0** **OK** when the <link_num> is greater than 10, response：

	+IP ERROR: Invalid parameter **ERROR** If PDP context has not been activated, or the connection has been established, or parameter is incorrect, or other errors, response: **+CIPOPEN: <link_num>,<err>** **ERROR** Transparent mode for UDP connection: When you want to use transparent mode to transmit UDP data, you should set AT+CIPMODE=1 before AT+NETOPEN. And if AT+CIPMODE=1 is set, the <link_num> is restricted to be only 0. <serverIP> and <serverPort> should be set if AT+CIPMODE=1. if success **CONNECT [<text>]** if failure **CONNECT FAIL** Other: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	-

Defined Values

<link_num>	Integer type, identifies a connection. Range is 0-9. If AT+CIPMODE=1 is set, the <link_num> is restricted to be only 0.
<type>	String type, identifies the type of transmission protocol. TCP Transmission Control Protocol UDP User Datagram Protocol
<serverIP>	String type, identifies the IP address of server. The IP address format consists of 4 octets, separated by decimal point, like ”AAA.BBB.CCC.DDD”. Also the domain name is supported here. **NOTE:** If the domain name is inputted here, the timeout value for the AT+CIPOPEN shall be decided by **AT+CIPDNSSET.**
<serverPort>	Integer type, identifies the port of TCP server, range is 0-65535. **NOTE:** When open port as TCP, the port must be the opened TCP port; When open port as UDP, the port may be any port. But, for Qualcomm, connecting the port 0 is regarded as an invalid

	operation.
<localPort>	Integer type, identifies the port of local socket, range is 0-65535.
<index>	Integer type, which indicates whether the module is used as a client or server. When used as server, the range is 0-3. <index> is the server index to which the client is linked. (-1) -- TCP/UDP client (0-3) -- TCP server index
<text>	String type, which indicates CONNECT result code. Please refer to ATX/AT\V/AT&E command for the string formats.
<err>	Integer type, the result of operation.0 is success, other value is failure.

Example

```
AT+CIPOPEN=0,”TCP”,”116.228.221.51”,100
OK

+CIPOPEN: 0,0
AT+CIPOPEN=1,”UDP”,,,8080
+CIPOPEN: 1,0

OK
AT+CIPOPEN=?
+CIPOPEN: (0-9),(“TCP”,”UDP”)

OK
AT+CIPOPEN?
+CIPOPEN: 0,”TCP”,”116.228.221.51”,100,-1
+CIPOPEN: 1
+CIPOPEN: 2
+CIPOPEN: 3
+CIPOPEN: 4
+CIPOPEN: 5
+CIPOPEN: 6
+CIPOPEN: 7
+CIPOPEN: 8
+CIPOPEN: 9

OK
```

11.2.4 AT+CIPCLOSE Destroy TCP/UDP client socket connection

AT+CIPCLOSE Destroy TCP/UDP client socket connection	
Test Command **AT+CIPCLOSE=?**	Response **+CIPCLOSE: (0-9)** **OK**
Read Command **AT+CIPCLOSE?**	Response **+CIPCLOSE: <link0_state>,<link1_state>,<link2_state>,<link3_state>,<link4_state>,<link5_state>,<link6_state>,<link7_state>,<link8_state>,<link9_state>** **OK**
Write Command **AT+CIPCLOSE=<link_num>**	Response If service type is TCP and the connection identified by <link_num> has been established, response: **OK** **+CIPCLOSE: <link_num>,<err>** If service type is TCP and the access mode is transparent mode, response: **OK** **CLOSED** **+CIPCLOSE: <link_num>,<err>** If service type is UDP and the connection identified by <link_num> has been established, response: **+CIPCLOSE: <link_num>,0** **OK** If service type is UDP and access mode is transparent mode, response: **CLOSED** **+CIPCLOSE: <link_num>,<err>** **OK** If the connection has not been established, abnormally closed, or parameter is incorrect, response: **+CIPCLOSE: <link_num>,<err>** **ERROR**

	Other: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<link_num>	Integer type, which identifies a connection. Range is 0-9.
<link_state>	Integer type, which indicates the state of connection identified by <link_num>. Range is 0-1. 0 -- disconnected 1 -- connected
<err>	Integer type, the result of operation. 0 is success, other value is failure

Example

```
AT+CIPCLOSE?
+CIPCLOSE: 1,0,0,0,0,0,0,0,0,0

OK
AT+CIPCLOSE=?
+CIPCLOSE: (0-9)

OK
AT+CIPCLOSE=0
OK

+CIPCLOSE: 0,0
```

11.2.5 AT+CIPSEND Send TCP/UDP data

AT+CIPSEND Send TCP/UDP data	
Test Command **AT+CIPSEND=?**	Response **+CIPSEND: (0-9),(1-1500)** **OK**
Write Command If service type is “TCP”, send data with changeable length	Response If the connection identified by <link_num> has been established successfully, response:

AT+CIPSEND=<link_num>, Response “>”, then type data to send, tap CTRL+Z to send data, tap ESC to cancel the operation	>**<input data>** **CTRL+Z** **OK** **+CIPSEND: <link_num>,<reqSendLength>, <cnfSendLength>** If <reqSendLength> is equal <cnfSendLength>, it means that the data has been sent to TCP/IP protocol stack successfully. If the connection has not been established, abnormally closed, or parameter is incorrect, response: **+CIPERROR: <err>** **ERROR** Other: **ERROR**
Write Command If service type is “TCP”, send data with fixed length **AT+CIPSEND=<link_num>,<length>** Response “>”, type data until the data length is equal to <length>	Response: If the connection identified by <link_num> has been established successfully, response: > **<input data with specified length>** **OK** **+CIPSEND: <link_num>,<reqSendLength>, <cnfSendLength>** If <reqSendLength> is equal <cnfSendLength>, it means that the data has been sent to TCP/IP protocol stack successfully. If the connection has not been established, abnormally closed, or parameter is incorrect, response: **+CIPERROR: <err>** **ERROR** Other: **ERROR**
Write Command If service type is “UDP”, send data with changeable length **AT+CIPSEND=<link_num>,,<serverIP>,<serverPort>** Response “>”, then type data to send, tap CTRL+Z to send	Response: If the connection identified by <link_num> has been established successfully, response: > **<input data>** **CTRL+Z** **OK** **+CIPSEND: <link_num>,<reqSendLength>, <cnfSendLength>**

data, tap ESC to cancel the operation	If the connection has not been established, abnormally closed, or parameter is incorrect, response: **+CIPERROR: <err>** **ERROR** Other: **ERROR**
Write Command If service type is “UDP”, send data with fixed length **AT+CIPSEND=<link_num>,<length>,<serverIP>,<serverPort>** Response “>”, type data until the data length is equal to <length>	Response: If the connection identified by <link_num> has been established successfully, response: > **<input data with specified length>** **OK** **+CIPSEND: <link_num>,<reqSendLength>, <cnfSendLength>** If the connection has not been established, abnormally closed, or parameter is incorrect, response: **+CIPERROR: <err>** **ERROR** Other: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	-

Defined Values

<link_num>	Integer type, identifies a connection. Range is 0-9.
<length>	Integer type, indicates the length of sending data, range is 1-1500.
<serverIP>	String type, which identifies the IP address of server. The IP address format consists of 4 octets, separated by decimal point, like ”AAA.BBB.CCC.DDD”.
<serverPort>	Integer type, identifies the port of TCP server, range is 0-65535. **NOTE:** When open port as TCP, the port must be the opened TCP port; When open port as UDP, the port may be any port. But, for Qualcomm, connecting the port 0 is regarded as an invalid operation.
<reqSendlength>	Integer type, the length of the data requested to be sent
<cnfSendLength>	Integer type, the length of the data confirmed to have been sent. -1 the connection is disconnected. 0 own send buffer or other side’s congestion window are full.

	Note: If the <cnfSendLength> is not equal to the <reqSendLength>, the socket then cannot be used further.
<err>	Integer type, the result of operation.0 is success, other value is failure.

Example

```
AT+CIPSEND=0,1
>S
OK

+CIPSEND: 0,1,1
AT+CIPSEND=1,1,”116.236.221.75”,6775
>S
OK

+CIPSEND: 1,1,1
AT+CIPSEND=2,
>Hello<Ctrl+Z>
OK

+CIPSEND: 2,5,5
AT+CIPSEND=3,,”116.236.221.75”,6775
>Hello World<Ctrl+Z>
OK

+CIPSEND: 3,11,11
AT+CIPSEND=2,
>Hello<ESC>
ERROR
AT+CIPSEND?
+CIPSEND: (0-9),(1-1500)

OK
```

NOTE

- Each <Ctrl+Z> character present in the data should be coded as <ETX><Ctrl+Z>. Each <ESC> character present in the data should be coded as <ETX><ESC>. Each <ETX> character will be coded as <ETX><ETX>. Single <Ctrl+Z> means end of the input data. Single <ESC> is used to cancel the sending.
- <ETX> is 0x03, and <Ctrl+Z> is 0x1A and <ESC> is 0x1B.

11.2.6 AT+CIPRXGET Retrieve TCP/UDP buffered data

AT+CIPRXGET Retrieve TCP/UDP buffered data	
Test Command **AT+CIPRXGET=?**	Response **+CIPRXGET: (0-4),(0-9),(1-1500)** **OK**
Read Command **AT+CIPRXGET?**	Response **+CIPRXGET: <mode>** **OK**
Write Command **AT+CIPRXGET=<mode>** In this case, <mode> can only be 0 or 1	Response If the parameter is correct, response: **OK** Else, response: **ERROR**
Write Command **AT+CIPRXGET=2,<link_num>[,<len>]** Retrieve data in ACSII form	Response: If <length> field is empty, the default value to read is 1500. If the buffer is not empty, response: **+CIPRXGET: <mode>,<link_num>,<read_len>,<rest_len>** <data>**ACSII form** **OK** If the buffer is empty, response: **+IP ERROR: No data** **ERROR** If the parameter is incorrect or other error, response: **+IP ERROR: <err_info>** **ERROR** Other: **ERROR**
Write Command **AT+CIPRXGET=3,<link_num>[,<len>]** Retrieve data in hex form	Response: If <length> field is empty, the default value to read is 750. If the buffer is not empty, response: **+CIPRXGET: <mode>,<link_num>,<read_len>,<rest_len>** <data>**hex form** **OK** If the buffer is empty, response: **+IP ERROR: No data** **ERROR**

	If the parameter is incorrect or other error, response: **+IP ERROR: <err_info>** **ERROR** Other: **ERROR**
Write Command **AT+CIPRXGET=4,<link_num>**	Response: If the parameter is correct, response: **+CIPRXGET: 4,<link_num>,<rest_len>** **OK** If the parameter is incorrect or other error, response: **+IP ERROR: <err_info>** **ERROR** Other: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<mode>	Integer type, sets the mode to retrieve data. Default value is 0. 0 – set the way to get the network data automatically 1 – set the way to get the network data manually 2 – read data, the max read length is 1500 3 – read data in HEX form, the max read length is 750 4 – get the rest data length
<link_num>	Integer type, identifies a connection. Range is 0-9.
<len>	Integer type, the data length to be read. Not required, the default value is 1500 when <mode>=2, and 750 when <mode>=3.
<read_len>	Integer type, the length of data that has been read.
<rest_len>	Integer type, the length of data which has not been read in the buffer.
<err_info>	String type, displays the cause of occurring error, please refer to Chapter 3 for details.

Example

```
AT+CIPRXGET=?
+CIPRXGET: (0-4),(1-1500)

OK
```

```
AT+CIPRXGET?
+CIPRXGET: 1

OK
AT+CIPRXGET=1
OK
AT+CIPRXGET=2,0,100
+CIPRXGET: 2,0,100,1300
01234567890123456789012345678901234567
89012345678901234567890123456789012345
67890123456789012345678 9

OK
AT+CIPRXGET=3,0,100
+CIPRXGET: 3,0,100,1200
30313233343536373839303132333435363738
39303132333435363738393031323334353637
38393031323334353637383930313233343536
37383930313233343536373839303132333435
36373839303132333435363738393031323334
3536373839

OK
AT+CIPRXGET=4,0
+CIPRXGET: 4,0,1200

OK
```

NOTE

- If set <mode> to 1, after receiving data, the module will buffer it and report a URC as “+CIPRXGET: 1,<link_num>” to notify the host. Then host can retrieve data by AT+CIPRXGET.
- If set <mode> to 0, the received data will be outputted to COM port directly by URC as “RECV FROM:<IP ADDRESS>:<PORT><CR><LF>+IPD(data length)<CR><LF><data>”.
- If the buffer is not empty, and the module receives data again, then it will not report a new URC until all the received data has been retrieved by AT+CIPRXGET from buffer.
- The default value of <mode> is 0. When <mode> is set to 1 and the 2-4 mode will take effect.
- If initially set <mode> to 1, after doing some data transmitting , set <mode> to 0, then the buffered data of the previously established connection will be output to the serial port directly, and the maximum length of output data at a time is 1500.

11.2.7 AT+IPADDR Get IP address of PDP context

AT+IPADDR Get IP address of PDP context	
Execution Command **AT+IPADDR**	Response: If PDP context has been activated successfully, response **+IPADDR: <ip_address>** **OK** Else, response: **+IP ERROR: Network not opened** **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<ip_address>	String type, identifies the IP address of current active socket PDP.

Example

```
AT+IPADDR
+IPADDR: 10.71.155.118

OK
```

11.2.8 AT+CIPHEAD Add an IP header when receiving data

AT+CIPHEAD Add an IP header when receiving data	
Test Command **AT+CIPHEAD=?**	Response **+CIPHEAD: (0-1)** **OK**
Read Command **AT+CIPHEAD?**	Response **+CIPHEAD: <mode>** **OK**
Write Command **AT+CIPHEAD=<mode>**	Response If the parameter is correct, response: **OK**

	Else, response: **ERROR**
Execution Command **AT+CIPHEAD**	Response: Set default value:(<mode>=1) **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<mode>	Integer type, indicates whether adding an IP header or not when receiving data. Default value is 1. 0 – not add IP header 1 – add IP header, the format is "+IPD(data length)"

Example

```
AT+CIPHEAD=?
+CIPHEAD: (0-1)

OK
AT+CIPHEAD=0
OK
```

11.2.9 AT+CIPSRIP Show remote IP address and port

AT+CIPSRIP Show remote IP address and port	
Test Command **AT+CIPSRIP=?**	Response **+CIPSRIP: (0-1)** **OK**
Read Command **AT+CIPSRIP?**	Response **+CIPSRIP: <mode>** **OK**
Write Command **AT+CIPSRIP=<mode>**	Response If the parameter is correct, response: **OK** Else, response: **ERROR**

Execution Command **AT+CIPSRIP**	Response: Set default value:(<mode>=1) **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<mode>	Integer type, indicates whether to show IP address and port of server or not when receiving data. Default value is 1. 0 – not show 1 – show, the format is as follows: "RECV FROM:<IP ADDRESS>:<PORT>"

Example

```
AT+CIPSRIP=?
+CIPSRIP: (0-1)

OK
AT+CIPSRIP=1
OK
```

11.2.10 AT+CIPMODE Select TCP/IP application mode

AT+CIPMODE Select TCP/IP application mode	
Test Command **AT+CIPMODE=?**	Response **+CIPMODE: (0-1)** **OK**
Read Command **AT+CIPMODE?**	Response **+CIPMODE: <mode>** **OK**
Write Command **AT+CIPMODE=<mode>**	Response If the parameter is correct, response: **OK** Else, response: **ERROR**
Execution Command **AT+CIPMODE**	Response: Set default value:(<mode>=0)

	OK
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<mode>	Integer type, sets TCP/IP application mode. Default value is 0. 0 – Non transparent mode 1 – Transparent mode

Example

```
AT+CIPMODE=?
+CIPMODE: (0-1)

OK
AT+CIPMODE=1
OK
```

11.2.11 AT+CIPSENDMODE Set sending mode

AT+CIPSENDMODE Set sending mode	
Test Command **AT+CIPSENDMODE=?**	Response **+CIPSENDMODE: (0-1)** **OK**
Read Command **AT+CIPSENDMODE?**	Response **+CIPSENDMODE: <mode>** **OK**
Write Command **AT+CIPSENDMODE=<mode>**	Response If the parameter is correct, response: **OK** Else, response: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<mode>	Integer type, sets sending mode. Default value is 0. 0 – Sending without waiting peer TCP ACK mode 1 – Sending wait peer TCP ACK mode

Example

```
AT+CIPSENDMODE=?
+CIPSENDMODE: (0-1)

OK
AT+CIPSENDMODE=1
OK
```

11.2.12 AT+CIPTIMEOUT Set TCP/IP timeout value

AT+CIPTIMEOUT Set TCP/IP timeout value	
Read Command **AT+CIPTIMEOUT?**	Response **+CIPTIMEOUT: <netopen_timeout>,<cipopen_timeout>,<cipsend_timeout>** **OK**
Write Command **AT+CIPTIMEOUT=[<netopen_timeout>][,[<cipopen_timeout>][,[<cipsend_timeout>]]]**	Response If the parameter is correct, response: **OK** Else, response: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<netopen_timeout>	Integer type, timeout value for AT+NETOPEN. default is120000ms. Range is 3000ms-120000ms.
<cipopen_timeout>	Integer type, timeout value for AT+CIPOPEN. default is120000ms. Range is 3000ms-120000ms.
<cipsend_timeout>	Integer type, timeout value for AT+CIPSEND. default is120000ms. Range is 3000ms-120000ms.

Example

```
AT+CIPTIMEOUT?
+CIPTIMEOUT: 30000,20000,40000

OK
AT+CIPTIMEOUT=30000,20000,40000
OK
```

11.2.13 AT+CIPCCFG Configure parameters of socket

AT+CIPCCFG Configure parameters of socket	
Test Command **AT+CIPCCFG=?**	Response **+CIPCCFG: (0-10),(0-1000),(0),(0-1),(0-1),(0-1),(500-120000)** **OK**
Read Command **AT+CIPCCFG?**	Response **+CIPCCFG: <NmRetry>,<DelayTm>,<Ack>,<errMode>,<HeaderType>,<AsyncMode>,<TimeoutVal>** **OK**
Write Command **AT+CIPCCFG=[<NmRetry>][,[<DelayTm>][,[<Ack>][,[<errMode>][,]<HeaderType>][,[[<AsyncMode>][,[<TimeoutVal>]]]]]]]]**	Response If the parameter is correct, response: **OK** Else, response: **ERROR**
Execution Command **AT+CIPCCFG**	Response Set default value: **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<NmRetry>	Integer type, number of retransmission to be made for an IP packet. Range is 0-10. The default value is 10.
<DelayTm>	Integer type, number of milliseconds to delay to output data of Receiving. Range is 0-1000. The default value is 0.
<Ack>	Integer type, it can only be set to 0. It's used to be compatible with old TCP/IP command set.

<errMode>	Integer type, sets mode of reporting <err_info>, default value is 1. 0 error result code with numeric values 1 error result code with string values
<HeaderType>	Integer type, select which data header is used when receiving data, it only takes effect in multi-client mode. Default value is 0. 0 add data header, the format is "+IPD<data length>" 1 add data header, the format is "+RECEIVE,<link num>,<data length>"
<AsyncMode>	Integer type, range is 0-1. Default value is 0. It's used to be compatible with old TCP/IP command set.
<TimeoutVal>	Integer type, set the minimum retransmission timeout value for TCP connection. Range is 500ms-120000ms. Default is 500ms.

Example

```
AT+CIPCCFG=?
+CIPCCFG:
(0-10),(0-1000),(0),(0-1),(0-1),(0),(500-120000)

OK
AT+CIPCCFG=3,500,0,1,1,1,500
OK
```

11.2.14 AT+SERVERSTART Startup TCP server

AT+SERVERSTART Startup TCP server	
Test Command **AT+SERVERSTART=?**	Response **+SERVERSTART: (0-65535),(0-3)** **OK**
Read Command **AT+SERVERSTART?**	Response If the PDP context has not been activated successfully, response: **+CIPERROR: <err>** **ERROR** If there exists opened server, response: [**+SERVERSTART: <server_index>,< port>** **…**] **OK** Other:

	ERROR
Write Command **AT+SERVERSTART=<port>,<server_index>[,<backlog>]**	Response If there is no error, response: **OK** If the PDP context has not been activated, or the server identified by <server_index> has been opened, or the parameter is not correct, or other errors, response: **+CIPERROR: <err>** **ERROR** Other: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<port>	Integer type, identifies the listening port of module when used as a TCP server. Range is 0-65535.
<server_index>	Integer type, the TCP server index, range is 0-3.
<Ack>	Integer type, it can only be set to 0. It's used to be compatible with old TCP/IP command set.
<backlog>	Integer type, the maximum connections can be queued in listening queue. Range is 1-3. Default is 3.

Example

```
AT+SERVERSTART=?
+SERVERSTART: 0,1000

OK
AT+SERVERSTART=8080,1
OK
```

NOTE

- After the "AT+SERVERSTART" executes successfully, an unsolicited result code is returned when a client tries to connect with module and module accepts request. The unsolicited result code is+CLIENT: < link_num>,<server_index>,<client_IP>:<port>.

11.2.15 AT+SERVERSTOP Stop TCP server

AT+SERVERSTOP Stop TCP server	
Write Command **AT+SERVERSTOP=<server_index>**	Response If there exists open connection with the server identified by <server_index>, or the server identified by <server_index> has not been opened, or the parameter is incorrect, response: **+SERVERSTOP: <server_index>,<err>** **ERROR** If the server socket is closed immediately, response: **+SERVERSTOP: <server_index>,0** **OK** (In general, the result is shown as below.) If the server socket starts to close, response: **OK** **+SERVERSTOP: <server_index>,<err>** Other: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<server_index>	Integer type, the TCP server index, range is 0-3.
<err>	Integer type, the result of operation.0 is success, other value is failure.

Example

```
AT+SERVERSTOP=0
+SERVERSTOP: 0,0

OK
```

NOTE

- Before stopping a TCP server, all sockets <server_index> of which equals to the closing TCP server index must be closed first.

11.2.16 AT+CIPACK Query TCP connection data transmitting status

AT+CIPACK Query TCP connection data transmitting status	
Test Command **AT+CIPACK=?**	Response **+CIPACK: (0-9)** **OK**
Write Command **AT+CIPACK=<link_num>**	Response If the PDP context has not been activated, or the connection identified by <link_num> has not been established, abnormally closed, or the parameter is incorrect, or other errors, response: **+IP ERROR: <err_info>** **ERROR** If the connection has been established, and the service type is "TCP", response: **+CIPACK: <sent_data_size>,<ack_data_size>,<recv_data_size>** **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<link_num>	Integer type, identifies a connection. Range is 0-9.
<sent_data_size>	Integer type, the total length of sent data
<ack_data_size>	Integer type, the total length of acknowledged data.
<recv_data_size>	Integer type, the total length of received data
<err>	Integer type, the result of operation. 0 is success, other value is failure.
<err_info>	String type, displays the cause of occurring error, please refer to Chapter 3 for details.

Example

```
AT+CIPACK=?
+CIPACK: (0-9)

OK
AT+CIPACK=0
+CIPACK: 16,16,5

OK
```

11.3 DNS&PING

11.3.1 AT+CDNSGIP Query the IP address of given domain name

AT+CDNSGIP Query the IP address of given domain name	
Test Command **AT+CDNSGIP=?**	Response **OK**
Write Command **AT+CDNSGIP=<domain name>**	Response If the given domain name has related IP, response: **+CDNSGIP: 1,<domain name>,<IP address>** **OK** If the given name has no related IP, response: **+CDNSGIP: 0,<dns error code>** **ERROR** Other: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<domain name>	String type (string should be included in quotation marks), indicates the domain name. The maximum length of domain name is 254. Valid characters allowed in the domain name area-z, A-Z, 0-9, "-"(hyphen) and ".". A domain name is made up of one label name or more label names separated by "." (e.g. AT+CDNSGIP="aa.bb.cc"). For label names separated by ".", length of each label must be no more than 63 characters. The beginning character of the domain name and of labels should be an alphanumeric character.
<IP address>	String type, indicates the IP address corresponding to the domain name.
<dns error code>	Integer type, indicates the error code. 10 DNS GENERAL ERROR

Example

```
AT+CDNSGIP=”www.baidu.com”
+CDNSGIP:
1,”www.baidu.com”,”61.135.169.21”

OK
```

11.3.2 AT+CDNSGHNAME Query the domain name of given IP address

AT+CDNSGHNAME Query the domain name of given IP address	
Test Command **AT+CDNSGHNAME=?**	Response **OK**
Write Command **AT+CDNSGHNAME=<IP address>**	Response If the given IP address has related domain name, response: **+CDNSGHNAME: <index>,<domain name>,<IP address>** **OK** If the given IP address has no related domain name, response: **+CDNSGHNAME: 0,<dns error code>** **ERROR** Other: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<domain name>	String type (string should be included in quotation marks), indicates the domain name. The maximum length of domain name is 254. Valid characters allowed in the domain name area-z, A-Z, 0-9, “-“(hyphen) and “.”. A domain name is made up of one label name or more label names separated by “.” (e.g. AT+CDNSGIP=”aa.bb.cc”). For label names separated by “.”, length of each label must be no more than 63 characters. The beginning character of the domain name and of labels should be an alphanumeric character.
<IP address>	String type (string should be included in quotation marks), indicates the IP address corresponding to the domain name.
<dns error code>	Integer type, which indicates the error code. 10 DNS GENERAL ERROR
<index>	Integer type, which indicates DNS result index. This value is always 1 if performing successfully. Currently only the

	first record returned from the DNS server will be reported.

Example

```
AT+CDNSGHNAME=”58.32.231.148”
+CDNSGHNAME: 1,”mail.sim.com”,”58.32.231.148”

OK
```

11.3.3 AT+CIPDNSSET Set DNS query parameters

AT+CIPDNSSET Set DNS query parameters	
Read Command **AT+CIPDNSSET?**	Response **+CIPDNSSET: 3,30000,7** **OK**
Write Command **AT+CIPCCFG=[<max_net_retries>][,[<net_timeout>][,[<max_query_retries>]]]**	Response If the parameter is correct, response: **OK** Else, response: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<max_net_retries>	Integer type, maximum retry times for opening PS network to perform DNS query. Range is 0-3. Default is 3.
<netopen_timeout>	Integer type, timeout value for each opening PS network operation when performing DNS query. Range is 3000ms-120000ms. Default value is 30000ms.
<max_query_retries>	Integer type, maximum retry times for performing DNS query using UDP packet. Range is 0-7. Default value is 7.

Example

```
AT+CIPDNSSET?
+CIPDNSSET: 1,30000,3

OK
```

```
AT+CIPDNSSET=1,30000,1
OK
```

11.3.4 AT+CPING Ping destination address

AT+CPING Ping destination address	
Test Command **AT+CPING=?**	Response **+CPING: IP address, (list of supported <dest_addr_type>s),(1-100),(4-188),(1000-10000),(10000-100000), (16-255)** **OK**
Write Command **AT+CPING=<dest_addr>,<dest_addr_type>[,<num_pings>[,<data_packet_size>[,<interval_time>[,<wait_time>[,<TTL>]]]]]**	Response **OK** If ping's result_type = 1 **+CPING: <result_type>,<resolved_ip_addr>,<data_packet_size>,<rtt>,<TTL>** If ping's result_type = 2 **+CPING: <result_type>** If ping's result_type = 3 **+CPING: <result_type>,<num_pkts_sent>,<num_pkts_recvd>,<num_pkts_lost>,<min_rtt>,<max_rtt>,<avg_rtt>**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined values

<dest_addr>	The destination is to be pinged; it can be an IP address or a domain name.
<dest_addr_type>	Integer type. Address family type of the destination address 1 – IPv4. 2 – IPv6(reserved)
<num_pings>	Integer type. The num_pings specifies the number of times the ping request (1-100) is to be sent. The default value is 4.
<data_packet_size>	Integer type. Data byte size of the ping packet (4-188). The default value is 64 bytes.

<interval_time>	Integer type. Interval between each ping. Value is specified in milliseconds (1000ms-10000ms). The default value is 2000ms.
<wait_time>	Integer type. Wait time for ping response. An ping response received after the timeout shall not be processed. Value specified in milliseconds (10000ms-100000ms). The default value is 10000ms
<TTL>	Integer type. TTL(Time-To-Live) value for the IP packet over which the ping(ICMP ECHO Request message) is sent (16-255), the default value is 255.
<result_type>	1 – Ping success 2 – Ping time out 3 – Ping result
<num_pkts_sent>	Indicates the number of ping requests that were sent out.
<num_pkts_recvd>	Indicates the number of ping responses that were received.
<num_pkts_lost>	Indicates the number of ping requests for which no response was received
<min_rtt>	Indicates the minimum Round Trip Time(RTT).
<max_rtt>	Indicates the maximum RTT.
<avg_rtt>	Indicates the average RTT.
<resolved_ip_addr>	Indicates the resolved ip address.
< rtt>	Round Trip Time.

Examples

```
AT+CPING="www.baidu.com",1,4,64,1000,10
000,255
OK

+CPING: 1,119.75.217.56,64,410,255

+CPING: 1,119.75.217.56,64,347,255

+CPING: 1,119.75.217.56,64,346,255

+CPING: 1,119.75.217.56,64,444,255

+CPING: 3,4,4,0,346,444,386
```

11.3.5 AT+CPINGSTOP Stop an ongoing ping session

AT+CPINGSTOP	Stop an ongoing ping session

Test Command **AT+CPINGSTOP=?**	Response **OK**
Write Command **AT+CPINGSTOP**	**+CPING: <result_type>,<num_pkts_sent>,<num_pkts_recvd>,<num_pkts_lost>,<min_rtt>,<max_rtt>,<avg_rtt>** **OK** **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined values

<result_type>	1 – Ping success 2 – Ping time out 3 – Ping result
<num_pkts_sent>	Indicates the number of ping requests that were sent out.
<num_pkts_recvd>	Indicates the number of ping responses that were received.
<num_pkts_lost>	Indicates the number of ping requests for which no response was received.
<resolved_ip_addr>	Indicates the resolved ip address.
<min_rtt>	Indicates the minimum Round Trip Time (RTT).
<max_rtt>	Indicates the maximum RTT.
<avg_rtt>	Indicates the average RTT.

Examples

```
AT+CPINGSTOP
OK
```

11.4 Information Elements related to TCP/IP

Information	Description
+CIPEVENT: NETWORK CLOSED UNEXPECTEDLY	Network is closed for network error (Out of service, etc). When this event happens, user's application needs to check and close all opened sockets, and then uses AT+NETCLOSE to release the network library if "AT+NETOPEN?" shows the network

	library is still opened.
+IPCLOSE: <client_index>, <close_reason>	Socket is closed passively. <client_index> is the link number. <close_reason>: 0 - Closed by local, active 1 - Closed by remote, passive 2 - Closed for sending timeout
+CLIENT: < link_num>,<server_index>,<client_IP>:<port>	While TCP server accepted a new socket client, the index is <link_num>. The TCP server index is <server_index>. The peer IP address is <client_IP>. The peer port is <port>.

11.5 Description of <err_info>

The fourth parameter <errMode> of AT+CIPCCFG is used to determine how <err_info> is displayed.
If <errMode> is set to 0, the <err_info> is displayed with numeric value.
If <errMode>is set to 1, the <err_info> is displayed with string value.
The default is displayed with string value.

Numeric Value	String Value
21	Operation failed
0	Connection time out
1	Bind port failed
2	Port overflow
3	Create socket failed
4	Network is already opened
5	Network is already closed
6	No clients connected
7	No active client
8	Network not opened
9	Client index overflow
10	Connection is already created
11	Connection is not created
12	Invalid parameter
13	Operation not supported
14	DNS query failed
15	TCP busy
16	Netclose failed for socket opened

17	Sending time out
18	Sending failure for network error
19	Open failure for network error
20	Server is already listening
21	No data
22	Port overflow

11.6 Description of <err>

<err>	Description of <err>
0	Operation succeeded
1	Network failure
2	Network not opened
3	Wrong parameter
4	Operation not supported
5	Failed to create socket
6	Failed to bind socket
7	TCP server is already listening
8	Busy
9	Sockets opened
10	Timeout
11	DNS parse failed for AT+CIPOPEN
12	Unknown error

12. AT Commands for FTPS

12.1 Overview of AT Commands for FTPS

Command	Description
AT+CFTPSSTART	Start FTP(S) service
AT+CFTPSSTOP	Stop FTP(S) Service
AT+CFTPSLOGIN	Login to a FTP(S)server
AT+CFTPSLOGOUT	Logout FTP(S) server
AT+CFTPSMKD	Create a new directory on FTP(S) server
AT+CFTPSRMD	Delete a directory on FTP(S) server
AT+CFTPSDELE	Delete a file on FTP(S) server
AT+CFTPSCWD	Delete a file on FTP(S) server
AT+CFTPSPWD	Get the current directory on FTP(S) server
AT+CFTPSTYPE	set the transfer type on FTP(S) serve
AT+CFTPSLIST	List the items in the directory on FTP(S) server
AT+CFTPSGETFILE	Get a file from FTP(S) server to module
AT+CFTPSPUTFILE	Put a file from module to FTP(S) server
AT+CFTPSGET	Get a file from FTP(S) server to serial port
AT+CFTPSPUT	Put a file to FTP(S) server through serial port
AT+CFTPSSINGLEIP	Set FTP(S) data socket address type
AT+CFTPSCACHERD	Set FTP(S) data socket address type
AT+CFTPSABORT	Abort FTP(S) operations
AT+CFTPSSIZE	Get the File Size on FTP(S) server

12.2 Detailed Description of AT Commands for FTPS

12.2.1 AT+CFTPSSTART Start FTP(S) service

AT+CFTPSSTART Start FTP(S) service	
Execution Command	Response
AT+CFTPSSTART	**OK**

	+CFTPSSTART: <errcode> or **+CFTPSSTART: <errcode>** **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<errcode>	The result of start FTP(S) service,0 is success, others are failure. Please refer to chapter 12.3.1.

Example

```
AT+CFTPSSTART
OK

+CFTPSSTART: 0
```

12.2.2 AT+CFTPSSTOP Stop FTP(S) Service

AT+CFTPSSTOP Stop FTP(S) Service	
Execution Command **AT+CFTPSSTOP**	Response **OK** **+CFTPSSTOP: <errcode>** or **+CFTPSSTOP: <errcode>** **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<errcode>	The result of stop FTP(S) service,0 is success, others are failure. Please refer to chapter 12.3.1.

Example

```
AT+CFTPSSTOP
OK

+CFTPSSTOP: 0
```

12.2.3 AT+CFTPSLOGIN Login to a FTP(S) server

AT+CFTPSLOGIN Login to a FTP(S) server	
Test Command **AT+CFTPSLOGIN=?**	Response **+CFTPSLOGIN: "ADDRESS",(1-65535)[,"USERNAME","PASSWORD"[,(0-3)]]** **OK**
Write Command **AT+CFTPSLOGIN="<host>",<port>,"<username>","<password>"[<server_type>]**	Response **OK** **+CFTPSLOGIN: <errcode>** or **+CFTPSLOGIN: <errcode>** **OK** or **+CFTPSLOGIN: <errcode>** **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<host>	Host address, string type, maximum length is 256
<port>	The host listening port for FTP(S), the range is from 1 to 65535
<username>	FTP(S) user name, string type, maximum length is 256

<password>	The user password, string type, maximum length is 256
<server_type>	FTP(S)server type,numeric,from0-3,default is 3 0 – FTP server. 1 – Explicit FTPS server with AUTH SSL. 2 – Explicit FTPS server with AUTH TLS. 3 – Implicit FTPS server.
<errcode>	The result code of the FTP/FTPS login. 0 is success. Others are failure, please refer to chapter 12.3.1.

Example

```
AT+CFTPSLOGIN="112.74.93.163",21,"tmf","t
mf123",0
OK

+CFTPSLOGIN: 0
```

12.2.4 AT+CFTPSLOGOUT Logout FTP(S) server

AT+CFTPSLOGOUT Logout FTP(S) server	
Test Command **AT+CFTPSLOGOUT=?**	Response **OK**
Execution Command **AT+CFTPSLOGOUT**	Response **OK** **+CFTPSLOGOUT: <errcode>** or **+CFTPSLOGOUT: <errcode>** **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<errcode>	The result code of the FTP/FTPS logout. 0 is success. Others are failure, please refer to chapter 12.3.1.

Example

```
AT+CFTPSLOGOUT
OK

+CFTPSLOGOUT: 0
```

12.2.5 AT+CFTPSMKD Create a new directory on FTP(S) server

AT+CFTPSMKD Create a new directory on FTP(S) server	
Test Command **AT+CFTPSMKD=?**	Response **+CFTPSMKD: "DIR"** **OK**
Write Command **AT+CFTPSMKD="<dir>"**	Response **OK** **+CFTPSMKD: 0** or **OK** **+CFTPSMKD: <errcode>** or **ERROR** or **+CFTPSMKD: <errcode>** **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<dir>	The directory to be created, string type, maximum length is 256.
<errcode>	The result of create directory, 0 is success, others are failure, please refer to chapter 12.3.1.

Example

```
AT+CFTPSMKD="TEST"
OK
```

```
+CFTPSMKD: 0
```

12.2.6 AT+CFTPSRMD Delete a directory on FTP(S) server

AT+CFTPSRMD Delete a directory on FTP(S) server	
Test Command **AT+CFTPSRMD=?**	Response **+CFTPSRMD: "DIR"** **OK**
Write Command **AT+CFTPSRMD="<dir>"**	Response 1)if delete the directory successfully： **OK** **+CFTPSRMD: 0** 2)if delete the directory failed: **OK** **+CFTPSRMD: <errcode>** 3) if parameter format or any errors: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<dir>	The directory to be removed. If the directory contains non-ASCII characters, the <dir> parameter should contain a prefix of {non-ascii}.String type, maximum length is 256.
<errcode>	The result of remove directory, 0 is success, others are failure, please refer to chapter 12.3.1.

Example

```
AT+CFTPSRMD="test"
OK

+CFTPSRMD: 0
```

12.2.7 AT+CFTPSDELE Delete a file on FTP(S) server

AT+CFTPSDELE Delete a file on FTP(S)server	
Test Command **AT+CFTPSDELE=?**	Response **+CFTPSDELE: "FILENAME"** **OK**
Write Command **AT+CFTPSDELE="<filename>"**	Response 1)if delete file successfully: **OK** **+CFTPSDELE: 0** 2)if failed: **OK** **+CFTPSDELE: <errcode>** 3)if parameter format or any other errors: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<filename>	The name of the file to be deleted. If the file name contains non-ASCII characters, the <filename> parameter should contain a prefix of {non-ascii}.String type,the maximum length is 256.
<errcode>	The result of delete a file, 0 is success, others are failure,please refer to chapter 12.3.1.

Example

```
AT+CFTPSDELE="TEST.txt"
OK

+CFTPSDELE: 0
```

12.2.8 AT+CFTPSCWD Change the current directory on FTP(S) server

AT+CFTPSCWD Change the current directory on FTP(S) sever	
Test Command	Response

AT+CFTPSCWD=?	**+CFTPSCWD: "DIR"** **OK**
Write Command **AT+CFTPSCWD="<dir>"**	Response 1)if delete file successfully:： **OK** **+CFTPSCWD: 0** 2)if failed: **OK** **+CFTPSCWD: <errcode>** 2)if failed: **+CFTPSCWD: <errcode>** **ERROR** 3)if parameter format or any other errors: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<dir>	The directory to be changed. If the directory contains non-ASCII characters, the <dir> parameter should contain a prefix of {non-ascii}.String type,the maximum length is 256.
<errcode>	The result of change the current directory, 0 is success, others are failure, please refer to chapter 12.3.1.

Example

```
AT+CFTPSCWD="/lu.liu/TEST7600"
OK

+CFTPSCWD: 0
```

12.2.9 AT+CFTPSPWD Get the current directory on FTPS server

AT+CFTPSPWD Get the current directory on FTPS server	
Execution Command **AT+CFTPSPWD**	Response **OK**

	+CFTPSPWD: ”<dir>” or **+CFTPSPWD: ”<dir>”** **OK** or **+CFTPSPWD: <errcode>** **ERROR** or **OK** **+CFTPSPWD: <errcode>** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<dir>	The name of the file to be deleted. If the file name contains non-ASCII characters, the <filename> parameter should contain a prefix of {non-ascii}.String type,the maximum length is 256.
<errcode>	The result of change current directory, 0 is success, others are failure, please refer to chapter 12.3.1.

Example

```
AT+CFTPSPWD
OK

+CFTPSPWD: "/test12"
```

12.2.10 AT+CFTPSTYPE Set the transfer type on FTP(S) server

AT+CFTPSTYPE Set the transfer type on FTP(S) server	
Test Command **AT+CFTPSTYPE=?**	Response **+CFTPSTYPE: (A,I)** **OK**

Read Command **AT+CFTPSTYPE?**	Response **+CFTPSTYPE: <type>** **OK**
Write Command **AT+CFTPSTYPE=<type>**	Response a)if set type successfully: **OK** **+CFTPSTYPE: 0** b)if set type failed: **OK** **+CFTPSTYPE: <errcode>**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<type>	The type of transferring: A – ASCII. I– Binary
<errcode>	The result of set type, 0 is success, others are failure, please refer to chapter 12.3.1.

Example

```
AT+CFTPTYPE=A
OK

+CFTPSTYPE: 0
```

12.2.11 AT+CFTPSLIST List the items in the directory on FTP(S) server

AT+CFTPSLIST List the items in the directory on FTP(S) server	
Write Command **AT+CFTPSLIST[="<dir>"]**	Response a)if set type successfully: **OK** **+CFTPSLIST: DATA,<len>** **…** **+CFTPSLIST: 0**

	b)if set type failed: **OK** **+CFTPSLIST: <errcode>** c)if parameter format or any other errors: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<dir>	The directory to be listed. If the directory contains non-ASCII characters, the <dir> parameter should contain a prefix of {non-ascii}.String type, the maximum length is 256
<errcode>	The result code of the listing.0 is success, others are failure,please refer to chapter12.3.1.

Example

```
AT+CFTPSLIST="/"
OK

+CFTPSLIST: DATA,1480
-rw-r--r-- 1 ftp ftp              10 Mar 19
13:51 111.TXT
-rw-r--r-- 1 ftp ftp               7 Mar 18
10:39 1111.txt
-rw-r--r-- 1 ftp ftp           10240 Mar 23
10:20 112.txt
-rw-r--r-- 1 ftp ftp              10 Mar 16
15:26 11K4.txt
-rw-r--r-- 1 ftp ftp            1434 Mar 18
10:47 1434B.txt
-rw-r--r-- 1 ftp ftp          307200 Mar 18
10:40 300K.txt
-rw-r--r-- 1 ftp ftp               9 Mar 18
10:53 333.txt
-rw-r--r-- 1 ftp ftp              16 Mar 17
14:11 36.txt

+CFTPSLIST: 0
```

12.2.12 AT+CFTPSGETFILE Get a file from FTP(S) server to module

AT+CFTPSGETFILE Get a file from FTP(S) server to module	
Test Command **AT+CFTPSGETFILE=?**	Response **+CFTPSGETFILE: [{non-ascii}]"FILEPATH"[,(1-4)]** **OK**
Write Command **AT+CFTPSGETFILE="<filepath>"[,<dir>[,<offset>]]**	Response a) if download file successfully : **OK** **+CFTPSGETFILE: 0** b) if failed: **OK** **+CFTPSGETFILE: <errcode>** c) if parameter format or any other errors: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<filepath>	The remote file path. When the file path doesn't contain"/", this command transfers file from the current remote FTPS directory. If the file path contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii}.String type, maximum length is 256.
<dir>	The directory to save the downloaded file.Numeric type, range is 1-4， default is 1(/cache) 1 – F:/ (/cache) 2 – D:/(sd card) 3 – E:/ (/data/media/) 4 – /mssl_cert/(this is for CA file downloading)
<offset>	Integer type, the download start position used for resume-from-break-point.
<errcode>	The result code of download file from FTP(s) server.0 is success, others are failure,please refer to chapter 12.3.1.

Example

```
AT+CFTPSGETFILE="settings.dat",3
OK
```

+CFTPSGETFILE: 0

12.2.13 AT+CFTPSPUTFILE Put a file from module to FTP(S) server

AT+CFTPSPUTFILE Put a file from module to FTP(S) server	
Test Command **AT+CFTPSPUTFILE=?**	Response **+CFTPSPUTFILE: [{non-ascii}]"FILEPATH"[,(1-3),(0-2147483647)]** **OK**
Write Command **AT+CFTPSPUTFILE="<filepath>"[,<dir>[,<rest_size>]]**	Response a)if upload file successfully : **OK** **+CFTPSPUTFILE: 0** b)if failed: **OK** **+CFTPSPUTFILE: <errcode>**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<filepath>	The remote file path. When the file path doesn't contain "/", this command transfers file to the current remote FTPS directory. If the file path contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii}.String type, maximum length is 256.
<dir>	The directory that contains the uploaded file. Numeric type, range is 1-3, default is 1(/cache) 1 – F:/ (/cache) 2 – D:/(sd card) 3 – E:/ (/data/media/)
<rest_size>	The value for FTP "REST" command which is used for broken transfer when transferring failed last time. Numeric type, the range is from 0 to 2147483647.
<errcode>	The result code of upload file to FTP(S)server.0 is success, others are failure,please refer to chapter12.3.1.

Example

```
AT+CFTPSPUTFILE="/LK/LM/sim_ZXX.TXT"
OK

+CFTPSPUTFILE: 0
```

12.2.14 AT+CFTPSGET Get a file from FTP(S) server to serial port

AT+CFTPSGET Get a file from FTP(S) server to serial port	
Test Command **AT+CFTPSGET=?**	Response **+CFTPSGET: [{non-ascii}]"FILEPATH"[,<rest_size>[(0,1)]]** **OK**

Write Command
AT+CFTPSGET="<filepath>"[,<rest_size>[,<using_cache>]]
	Response
	a)if<using_cache> is 0(default),and get file successfully :
	
	OK
	
	+CFTPSGET: DATA,<len>
	…
	+CFTPSGET: DATA, <len>
	…
	…
	+CFTPSGET: 0
	b) if <using_cache> is 1 and get file successfully:
	OK
	
	+CFTPS: RECV EVENT
	
	AT+CFTPSCACHERD?
	//you can use this command to check the size of the received data
	+CFTPSCACHERD: 102400
	
	OK
	
	//output cached data now:
	AT+CFTPSCACHERD
	+CFTPSGET: DATA,<len>
	……
	OK
	……
	+CFTPSGET: 0

	c)if failed: **OK** **+CFTPSGET: <errcode>**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<filepath>	The remote file path. When the file path doesn't contain "/", this command transfer file from the current remote FTPS directory. If the file path contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii}.String type, maximum length is 256.
<rest_size>	The value for FTP "REST" command which is used for broken transfer when transferring failed last time. Numeric type, the range is from 0 to 2147483647
<using_cache>	Numeric, rang is 0-1 0–Do not use cache, module will output the items data to serial port when list successfully. 1 – Use cache, module will report "+CFTPS: RECV EVENT" when list successfully (Data will be output using AT+CFTPSCACHERD command)
<errcode>	The result code of download file from FTP(s) server.0 is success, others are failure, please refer to chapter 12.3.1.

Example

```
AT+CFTPSGET="/BBB.TXT"
OK

+CFTPSGET: DATA,110
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHH
+CFTPSGET: 0
```

12.2.15 AT+CFTPSPUT Put a file to FTP(S) server through serial port

AT+CFTPSPUT Put a file to FTP(S) server through serial port

Test Command **AT+CFTPSPUT=?**	Response **+CFTPSPUT: [{non-ascii}]"FILEPATH"[,<data_len>[,<rest_size>]]** **OK**
Write Command **AT+CFTPSPUT="<filepath>"[,<data_len>[,<rest_size>]]**	Response a)if upload file through serial port successfully: **OK** **+CFTPSPUT: 0** b)if failed before input data: **+CFTPSPUT: <errcode>** **ERROR** c)if failed after input data: **OK** **+CFTPSPUT: <errcode>** d)if parameter format i or any other errors: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<filepath>	The remote file path. When the file path doesn't contain "/", this command transfers file to the current remote FTPS directory. If the file path contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii}.String type, maximum length is 256.
<data_len>	Numeric type,The length of the data to send, the maximum length is 2048.if parameter<data_len> is omitted, Each <Ctrl+Z>character present in the data flow of serial port when downloading FTP data will be coded as <ETX><Ctrl+Z>. Each <ETX> character will be coded as <ETX><ETX>. Single <Ctrl+Z> means end of the FTP data.<ETX> is 0x03, and <Ctrl+Z> is 0x1A.
<rest_size>	The value for FTP "REST" command which is used for broken transfer when transferring failed last time. Numeric type, the range is from 0 to 2147483647.
<errcode>	The result code of upload data to FTP(s) server.0 is success, others are failure, please refer to chapter 12.3.1.

Example

```
AT+CFTPSPUT="/LK/LM/LO.TXT"
>123457860
OK

+CFTPSPUT: 0
```

12.2.16 AT+CFTPSSINGLEIP Set FTP(S) data socket address type

AT+CFTPSSINGLEIP Set FTP(S) data socket address type	
Test Command **AT+CFTPSSINGLEIP=?**	Response **+CFTPSSINGLEIP: (0,1)** **OK**
Read Command **AT+CFTPSSINGLEIP?**	Response **+CFTPSSINGLEIP: <singleip>** **OK**
Write Command **AT+CFTPSSINGLEIP=<singleip>**	Response If parameter format is right and set successfully: **OK** If parameter format is not right or any other error occurs: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<singleip>	The FTPS data socket IP address type: 0 – decided by PORT response from FTPS server 1 – the same as the control socket.

Example

```
AT+CFTPSSINGLEIP=1
OK
```

12.2.17 AT+CFTPSCACHERD Output cached data to MCU

AT+CFTPSCACHERD Output cached data to MCU	
Read Command **AT+CFTPSCACHERD?**	Response **+CFTPSCACHERD: <len>** **OK**
Execution Command **AT+CFTPSCACHERD**	Response If cache data is AT+CFTPSGET, and everything goes well: **+CFTPSGET: DATA,<out_len><CR><LF>** **…** **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<len>	Numeric type, The bytes of data cached in FTPS module.
<out_len>	The bytes of data to output. The maximum value is 1024 for each AT+CFTPSCACHERD calling.

Example

```
AT+CFTPSCACHERD?
+CFTPSCACHERD: 21078

OK
```

12.2.18 AT+CFTPSABORT Abort FTP(S) Operations

AT+CFTPSABORT Abort FTP(S) Operations	
Execution Command **AT+CFTPSABORT**	Response if abort FTP(S) operation successfully: **OK** **+CFTPSABORT: 0** sometimes abort successfully returns:

	+CFTPSABORT: 0
	
	OK
	
	if failed:
	OK
	
	+CFTPSABORT: <errcode>
	
	if any other error occurs:
	ERROR

Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<errcode>	The result of abort FTP(S) service,0 is success, others are failure. Please refer to chapter 12.3.1.

Example

```
AT+CFTPSABORT
OK

+CFTPSABORT: 0
```

12.2.19 AT+CFTPSSIZE Get the File Size on FTP(S) server

AT+CFTPSSIZE Get the File Size on FTP(S) server	
Test Command **AT+CFTPSSIZE=?**	Response **+CFTPSSIZE: "<filepath>"** **OK**
Write Command **AT+CFTPSSIZE="<filepath>"**	Response **OK** **+CFTPSSIZE: <filesize>** or **OK** **+CFTPSSIZE: <errcode>**

	or **ERROR** or **+CFTPSSIZE: <errcode>** **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<filepath>	The remote filepath on FTP(S) server.String type,max length is 256
<filesize>	Numeric type，size of the remote file on FTP(S) server.
<errcode>	The result code of get file size. Please refer to chapter 12.3.1.

Example

```
AT+CFTPSSIZE="TEST.txt"
OK

+CFTPSSIZE: 1024
```

12.3 Summary of result codes for FTPS

12.3.1 Summary of Command result <errcode>

Code of <errcode>	Meaning
0	Success
1	SSL alert
2	Unknown error
3	Busy
4	Connection closed by server
5	Timeout
6	Transfer failed
7	File not exists or any other memory error
8	Invalid parameter
9	Operation rejected by server

10	Network error
11	State error
12	Failed to parse server name
13	Create socket error
14	Connect socket failed
15	Close socket failed
16	SSL session closed
17	File error,file not exist or other error.
421	Server response connection time out, while received error code 421,you need do AT+CFTPSLOGOUT to logout server then AT+CFTPSLOGIN again for further operations.

12.3.2 Summary of Unsolicited Result Codes

Unsolicited codes	Description
+CFTPSNOTIFY:PEER CLOSED	When client disconnect passively, URC "+CFTPSNOTIFY: PEER CLOSED" will be reported, then user need to execute AT+CFTPSLOGOUT andlog in again.

13. AT Commands for HTTPS

13.1 Overview of AT Commands for HTTPS

Command	Description
AT+HTTPINIT	Sart HTTP(S) service
AT+HTTPTERM	Stop HTTP(S) service.
AT+HTTPPARA	Set HTTP(S) Parameter
AT+HTTPACTION	HTTP(S) Method Action
AT+HTTPHEAD	Read the HTTP(S) Header Information of Server Response
AT+HTTPREAD	Read the response Information of HTTP(S) Server
AT+HTTPDATA	Input HTTP(S) Data
AT+HTTPPOSTFILE	Send HTTP(S) Request to HTTP server by File
AT+HTTPREADFILE	Receive HTTP(S) Response Content to a file

13.2 Detailed Description of AT Commands for HTTPS

13.2.1 AT+HTTPINIT Start HTTP(S) service

AT+HTTPINIT Start HTTP(S) service	
Execution Command **AT+HTTPINIT**	Response a)If start HTTP service successfully: **OK** b)If failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	

Example

```
AT+HTTPINIT
```

OK

13.2.2 AT+HTTPTERM Stop HTTP(S) Service

AT+HTTPTERM Stop HTTP(S) service	
Execution Command **AT+HTTPTERM**	Response a)If stop HTTP service successfully: **OK** b)If failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	

Example

```
AT+HTTPTERM
OK
```

13.2.3 AT+HTTPPARA Set HTTP(S) Parameters value

AT+HTTPPARA Set HTTP(S) Parameters value	
Write Command **AT+HTTPPARA="URL","<url>"**	Response a)If parameter format is right: **OK** b) If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="CONNECTTO",<conn_timeout>**	Response a)If parameter format is right: **OK** b) If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="RECVTO",<recv_timeout>**	Response a)If parameter format is right: **OK** b) If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="CONTENT",**	Response a)If parameter format is right:

"<content_type>"	**OK** b) If parameter format is not rightor other errors occur: **ERROR**
Write Command **AT+HTTPPARA="ACCEPT","<accept-type>"**	Response a)If parameter format is right: **OK** b) If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="UA","<user_agent>"**	Response a)If parameter format is right: **OK** b) If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="SSLCFG","<sslcfg_id>"**	Response a)If parameter format is right: **OK** b) If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="USERDATA","<user_data>"**	Response a)If parameter format is right: **OK** b) If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="BREAK",<break>**	Response a)If parameter format is right: **OK** b) If parameter format is not right or other errors occur: **ERROR**
Write Command **AT+HTTPPARA="BREAKEND",<breakend>**	Response a)If parameter format is right: **OK** b) If parameter format is not right or other errors occur: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<url>	URL of network resource.String,start with “http://” or”https://” a)http://’server’/’path’:’tcpPort’. b)https://’server’/’path’:’tcpPort’ “server”: DNS domain name or IP address “path”: path to a file or directory of a server “tcpPort”: http default value is 80,https default value is 443.(can be

	omitted)
<conn_timeout>	Timeout for accessing server, Numeric type, range is 20-120s, default is 120s.
<recv_timeout>	Timeout for receiving data from server, Numeric type range is 2-120s, default is 10s.
<content_type>	This is for HTTP “Content-Type” tag, String type, max length is 256, default is “text/plain”.
<accept-type>	This is for HTTP “Accept-type” tag, String type, max length is 256,default is “*/*”.
<user_agent>	Parameter for HTTP header User-Agent information.String type,max l ength is 256.
<sslcfg_id>	This is setting SSL context id, Numeric type, range is 0-9. Default is 0.
<user_data>	The customized HTTP header information. String type,max length is 512.
<break>	Parameter for HTTP method “GET”, used for resuming broken transfer. The start of the broken transfer. Default is 0.
<breakend>	Parameter for HTTP method “GET”, used for resuming broken transfer. The end of the broken transfer. Default is 0. If both “break” and “breakend” are 0, the resume broken transfer function is disabled. If “breakend” is bigger than “break”, the transfer scope is from “break” to “breakend”. If “breakend” is smaller than “break”, the transfer scope is from “break” to the end of the file.

Example

```
AT+HTTPPARA="USERDATA","Authorization: Basic Y2FycGx1c2dvOmNhcnBsdXgz"
OK
```

13.2.4 AT+HTTPACTION HTTP(S) Method Action

AT+HTTPACTION HTTP(S) Method Action	
Test Command **AT+HTTPACTION=?**	Response **+HTTPACTION: (0-3)** **OK**
WriteCommand **AT+HTTPACTION=<method>**	Response a)If parameter format is right : **OK** **+HTTPACTION: <method>,<statuscode>,<datalen>**

	b)If parameter format is not right or other errors occur: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	

Defined Values

<method>	HTTP method specification: 0: GET 1: POST 2: HEAD 3: DELETE
<statuscode>	Please refer to chapter 13.3.1
<datalen>	The length of data received

Example

```
AT+HTTPACTION=1
OK

+HTTPACTION: 1,200,2800
```

13.2.5 AT+HTTPHEAD Read the HTTP(S) Header Information of Server Response

AT+HTTPHEAD Read the HTTP(S) Header Information of Server Response	
Execution Command **AT+HTTPHEAD**	Response a)If read the header information successfully： **+HTTPHEAD: DATA,<data_len>** **<data>** **OK** b)If read failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<data_len>	The length of HTTP header
<data>	The header information of HTTP response

Example

```
AT+CHTTPHEAD
+HTTPHEAD: 750
HTTP/1.1 200 OK
Date: Thu, 29 Mar 2018 09:21:12 GMT
Content-Type: text/html
Content-Length: 14615
Last-Modified: Thu, 15 Mar 2018 08:23:00 GMT
Connection: Keep-Alive
Vary: Accept-Encoding
Set-Cookie: BAIDUID=EF38663A5539EBEAE702321037D5491B:FG=1; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: BIDUPSID=EF38663A5539EBEAE702321037D5491B; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1522315272; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/; domain=.baidu.com
P3P: CP=" OTI DSP COR IVA OUR IND COM "
Server: BWS/1.1
X-UA-Compatible: IE=Edge,chrome=1
Pragma: no-cache
Cache-control: no-cache
Accept-Ranges: bytes
OK
```

13.2.6 AT+HTTPREAD Read the Response Information of HTTP(S) Server

AT+HTTPREAD Read the Response Information of HTTP(S) Server	
Read Command **AT+HTTPREAD?**	Response a)If check successfully: **+HTTPREAD: LEN,<len>** **OK** b)If failed(no more data other error): **ERROR**
Write Command **AT+HTTPREAD=<byte_size>**	Response a)If read the response info successfully： **OK** **+HTTPREAD: DATA,<data_len>** **<data>**

	[+HTTPREAD: DATA,<data_len> **<data>** **…]** **+HTTPREAD: 0** If <byte_size> is bigger than the data size received, module will only return actual data size. b) If read failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	

Defined Values

<byte_size>	The length of data to read
<data_len>	The actual length of read data
<data>	Response content from HTTP server
<len>	Total size of data saved in buffer

Example

```
AT+HTTPREAD=0,10
OK

+HTTPREAD: 10
<!doctyped
+HTTPREAD: 0
```

13.2.7 AT+HTTPDATA Input HTTP(S) Data

AT+HTTPDATA Input HTTP(S) Data	
Write Command **AT+HTTPDATA=<size>,<time>**	Response a)if parameter format is right: **DOWNLOAD** **<input data here>** When the total size of the inputted data reaches <size>, TA will report the following code. Otherwise, the serial port will be blocked. **OK** b)If parameter format is wrong or other errors occur:

	ERROR
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<size>	Size in bytes of the data to post. range is 1- 153600 (bytes)
<time>	Maximum time in seconds to input data, range is 10-65535.

Example

```
AT+HTTPDATA=14,10000
DOWNLOAD
1234567890qwer
OK
```

13.2.8 AT+HTTPPOSTFILE Send HTTP Request to HTTP(S) server by File

AT+HTTPPOSTFILE Send HTTP Request to HTTP(S) server by File	
Test Command **AT+HTTPPOSTFILE=?**	Response **+HTTPPOSTFILE: <filename>[,(1-3)[,(0-3)[,(0-1)]]]** **OK**
Write Command **AT+HTTPPOSTFILE=<filename>[,<path>][,<method>][,<send_header>]**	Response a)if parameter format is right and server connected successfully: a.1 server response and content is not null **OK** **+HTTPPOSTFILE: <method>,<httpstatuscode>,<content_len>** a.2 server response but has no content **OK** **+HTTPPOSTFILE: <method>,<httpstatuscode>,0** b)if parameter format is right but server connected unsuccessfully: **OK** **+HTTPPOSTFILE: <method>,<errcode>,0** c)if parameter format is not right or any other error occurs: **ERROR**

Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	

Defined Values

<filename>	String type, filename, the max length is 64.unit:byte.
<path>	The directory where the sent file saved. Numeric type, range is 1-3 1 –F:/ (/cache) 2 – D:/(sd card) 3 –E:/ (/data/media/)
<method>	HTTP method specification: 0–GET 1– POST 2– HEAD 3– DELETE
<httpstatuscode>	Please refer to chapter 13.3.1
<errcode>	Please refer to chapter13.3.2
<send_header>	Send file as HTTP header and Body or Only as Body. Numeric type, the range is 0-1, the default is 0. 0 –Send file as HTTP header and body 1 – Send file as Body

Example

```
AT+HTTPPOSTFILE="baidu.txt",3
OK

+HTTPPOSTFILE: 1,714,0
```

13.2.9 AT+HTTPREADFILE Receive HTTP(S) Response Content to a file

AT+HTTPREADFILE Receive HTTP(S) Response Content to a File	
Test Command **AT+HTTPREADFILE=?**	Response **+HTTPREADTFILE: <filename>[,(1-4)]** **OK**
Write Command **AT+HTTPREADFILE=<filename>[,<path>]**	Response a)if parameter format is right : **OK** **+HTTPREADFILE: <result>**

	b)if parameter format is right: **+HTTPREADFILE: <result>** **OK** c)if failed: **+HTTPREADFILE: <result>** **ERROR** d)if parameter format is not right or any other error occurs: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<filename>	String type, filename, the max length is 64.unit:byte.
<path>	1 – F:/ (/cache/) 2 – D:/(sd card) 3 – E:/ (/data/media/) 4 – /mssl_cert/(this is for CA file downloading)

Example

```
AT+HTTPREADFILE="baidu.txt",3
OK

+HTTPREADFILE: 0
```

13.3 Summary of result codes for HTTPS

13.3.1 Summary of HTTP(S) Response Code

Code of <httpstatuscode>	Meaning
100	Continue
101	Switching Protocols

200	OK
201	Created
201	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified
305	Use Proxy
307	Temporary Redirect
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Large
415	Unsupported Media Type
416	Requested range not satisfiable
417	Expectation Failed
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway timeout
505	HTTP Version not supported
600	Not HTTP PDU
601	Network Error
602	No memory

603	DNS Error
604	Stack Busy

13.3.2 Summary of HTTP(S) error Code

Code of <errcode>	Meaning
0	Success
701	Alert state
702	Unknown error
703	Busy
704	Connection closed error
705	Timeout
706	Receive/send socket data failed
707	File not exists or other memory error
708	Invalid parameter
709	Network error
710	start a new ssl session failed
711	Wrong state
712	Failed to create socket
713	Get DNS failed
714	Connect socket failed
715	Handshake failed
716	Close socket failed
717	No network error
718	Send data timeout
719	CA missed

14. AT Commands for HTP

14.1 Overview of AT Commands for HTP

Command	Description
AT+CHTPSERV	Set HTP server info
AT+CHTPUPDATE	Updating date time using HTP protocol

14.2 Detailed Description of AT Commands for HTP

14.2.1 AT+CHTPSERV Set HTP server info

AT+CHTPSERV Set HTP server info	
Test Command **AT+CHTPSERV=?**	Response **+CHTPSERV: "ADD","HOST",(1-65535), (0-1)["PROXY",(1-65535)]** **+CHTPSERV: "DEL",(0-15)** **OK**
Read Command **AT+CHTPSERV?**	Response **+CHTPSERV: <index>"<host>",<port>,<http_version>["<proxy>",<proxy_port>]** **…** **+CHTPSERV: <index>"<host>",<port>["<proxy>",<proxy_port>]** **OK** or **OK (if HTP server not setted)**
Write Command **AT+CHTPSERV="<cmd>","<host_or_idx>"[,<port>,<http_version>["<proxy>",<prox**	Response **OK** or **ERROR**

y_port>]]	
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<cmd>	This command to operate the HTP server list. “ADD”: add a HTP server item to the list “DEL”: delete a HTP server item from the list
<host_or_idx>	If the <cmd> is “ADD”, this field is the same as <host>, needs quotation marks; If the <cmd> is “DEL”, this field is the index of the HTP server item to be deleted from the list, does not need quotation marks.
<host>	The HTP server address.
<port>	The HTP server port.
<http_version>	The HTTP version of the HTP server: 0-HTTP 1.0 1-HTTP 1.1
<proxy>	The proxy address
<proxy_port>	The port of the proxy
<index>	The HTP server index.

Example

```
AT+CHTPSERV=”ADD”,”www.google.com”,80,1
OK
```

14.2.2 AT+CHTPUPDATE Updating date time using HTP protocol

AT+CHTPUPDATE Updating date time using HTP protocol	
Test Command **AT+CHTPUPDATE=?**	Response **OK**
Read Command **AT+CHTPUPDATE?**	Response **+CHTPUPDATE: <status>** **OK**
Execution Command **AT+CHTPUPDATE**	Response **OK**

	+CHTPUPDATE: <err> or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<status>	The status of HTP module: Updating: HTP module is synchronizing date time NULL: HTP module is idle now
<err>	The result of the HTP updating

Example

```
AT+CHTPUPDATE
OK

+CHTPUPDATE: 0
```

14.2.3 Unsolicited HTP Codes

Code of <err>

0	Operation succeeded
1	Unknown error
2	Wrong parameter
3	Wrong date and time calculated
4	Network error

15. AT Commands for NTP

15.1 Overview of AT Commands for NTP

Command	Description
AT+CNTP	Update system time

15.2 Detailed Description of AT Commands for NTP

15.2.1 AT+CNTP Update system time

AT+CNTP Update system time	
Test Command **AT+CNTP=?**	Response **+CNTP: 255,(-96~96)** **OK**
Read Command **AT+CNTP?**	Response **+CNTP: <host>,<timezone>** **OK**
Write Command **AT+CNTP=”<host>”[,<timezone>]**	Response **OK** or **ERROR**
Execution Command **AT+CNTP**	Response **+CNTP: <host>,<timezone>** **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-

Reference	-

Defined Values

<host>	NTP server address, length is 255.
<timezone>	Local time zone,the range is(-96 to 96), default value is 0.

Example

```
AT+CNTP="202.120.2.101",32
OK
AT+CNTP
OK

+CNTP: 0
```

15.2.2 Unsolicited NTP Codes

Code of <err>

0	Operation succeeded
1	Unknown error
2	Wrong parameter
3	Wrong date and time calculated
4	Network error
5	Time zone error
6	Time out error

16. AT Commands for MQTT(S)

16.1 Overview of AT Commands for MQTT(S)

Command	Description
AT+CMQTTSTART	Start MQTT service
AT+CMQTTSTOP	STOP MQTT service
AT+CMQTTACCQ	Acquire a client
AT+CMQTTREL	Release a client
AT+CMQTTSSLCFG	Set the SSL context
AT+CMQTTWILLTOPIC	Input the will topic
AT+CMQTTWILLMSG	Input the will message
AT+CMQTTCONNECT	Connect to MQTT server
AT+CMQTTDISC	Disconnect from server
AT+CMQTTTOPIC	Input the publish message topic
AT+CMQTTPAYLOAD	Input the publish message body
AT+CMQTTPUB	Publish a message to server
AT+CMQTTSUBTOPIC	Input a subscribe message topic
AT+CMQTTSUB	Subscribe a message to server
AT+CMQTTUNSUBTOPIC	Input a unsubscribe message topic
AT+CMQTTUNSUB	Unsubscribe a message to server
AT+CMQTTCFG	Configure the MQTT Context

16.2 Detailed Description of AT Commands for MQTT(S)

16.2.1 AT+CMQTTSTART Start MQTT service

AT+CMQTTSTART Start MQTT service	
Execution Command **AT+CMQTTSTART**	Response **OK** **+CMQTTSTART: <err>**

	or **+CMQTTSTART: <err>** **OK** or **ERROR** **+CMQTTSTART: <err>** or **+CMQTTSTART: <err>** **ERROR** or **ERROR**

Defined Values

<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTSTART
OK

+CMQTTSTART: 0
```

NOTE

- It must be executed before any other MQTT related operations

16.2.2 AT+CMQTTSTOP STOP MQTT service

AT+CMQTTSTOP STOP MQTT service	
Execution Command **AT+CMQTTSTOP**	Response **OK** **+CMQTTSTOP: <err>** or **+CMQTTSTOP: <err>**

	OK or **ERROR** **+CMQTTSTOP: <err>** or **+CMQTTSTOP: <err>** **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTSTOP
OK

+CMQTTSTOP: 0
```

16.2.3 AT+CMQTTACCQ Acquire a client

AT+CMQTTACCQ Acquire a client	
Test Command **AT+CMQTTACCQ=?**	Response **+CMQTTACCQ: (0-1),(1-128),(0-1),(3-4)** **OK**
Read Command **AT+CMQTTACCQ?**	Response **+CMQTTACCQ: <client_index>,<clientID>,<server_type>** **+CMQTTACCQ: <client_index>,<clientID>,<server_type>** **OK**
Write Command **AT+CMQTTACCQ=<client_index>,<clientID>[,<server_type>[,<mqtt_version>]]**	Response **OK** or **+CMQTTACCQ: <client_index>,<err>**

	ERROR or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<clientID>	The UTF-encoded string. It specifies a unique identifier for the client. The string length is from 1 to 128 bytes.
<server_type>	A numeric parameter that identifies the server type. The default value is 0. 0 - MQTT server with TCP 1 - MQTT server with SSL/TLS
<mqtt_version>	A numeric parameter that identifies the MQTT protocol version. The permitted value is 3 or 4. 3 - MQTT version 3.1 4 - MQTT version 3.1.1
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTACCQ=0,"client test0",0,4
OK
```

NOTE

- It must be called before all commands about MQTT connect and after AT+CMQTTSTART

16.2.4 AT+CMQTTREL Release a client

AT+CMQTTREL Release a client	
Test Command **AT+CMQTTREL=?**	Response **+CMQTTREL: (0-1)** **OK**

Read Command **AT+CMQTTREL?**	Response **OK**
Write Command **AT+CMQTTREL=<client_index>**	Response **OK** or **+CMQTTREL: <client_index>,<err>** **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTREL=0
OK
```

NOTE

- It must be called after AT+CMQTTDISC and before AT+CMQTTSTOP

16.2.5 AT+CMQTTSSLCFG Set the SSL context

AT+CMQTTSSLCFG Set the SSL context	
Test Command **AT+CMQTTSSLCFG=?**	Response **+CMQTTSSLCFG: (0,1),(0-9)** **OK**
Read Command **AT+CMQTTSSLCFG?**	Response **+CMQTTSSLCFG: <session_id>,[<ssl_ctx_index >]** **+CMQTTSSLCFG: <session_id>,[<ssl_ctx_index >]**

	OK
Write Command **AT+CMQTTSSLCFG=<session_id>,<ssl_ctx_index>**	Response **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<session_id>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<ssl_ctx_index>	The SSL context ID which will be used in the SSL connection. Refer to the <ssl_ctx_index> of AT+CSSLCFG

Example

```
AT+CMQTTSSLCFG=0,1
OK
```

NOTE

- If you don't set the SSL context by this command before connecting to server by AT+CMQTTCONNECT, the CMQTTCONNECT operation will use the SSL context as same as index <session_id> (the 1st parameter of AT+ CMQTTCONNECT) when connecting to the server

16.2.6 AT+CMQTTWILLTOPIC Input the will topic

AT+CMQTTWILLTOPIC Input the will topic	
Test Command **AT+CMQTTWILLTOPIC=?**	Response **+CMQTTWILLTOPIC: (0-1),(1-1024)** **OK**
Write Command **AT+CMQTTWILLTOPIC=<client_index>,<req_length>**	Response **>** **<input data here>** **OK** or

	+CMQTTWILLTOPIC: <client_index>,<err> **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<req_length>	The length of input topic. The will topic should be UTF-encoded string. The range is from 1 to 1024 bytes.
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTWILLTOPIC=0,15
>simcomwilltopic
OK
```

16.2.7 AT+CMQTTWILLMSG Input the will message

AT+CMQTTWILLMSG Input the will message	
Test Command **AT+CMQTTWILLMSG=?**	Response **+CMQTTWILLMSG: (0-1),(1-1024),(0-2)** **OK**
Write Command **AT+CMQTTWILLMSG=<client_index>,<req_length>,<qos>**	Response **>** **<input data here>** **OK** or **+CMQTTWILLMSG: <client_index>,<err>** **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-

Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<req_length>	The length of input data. The will message should be UTF-encoded string. The range is from 1 to 1024 bytes.
<qos>	The qos value of the will message. The range is from 0 to 2.
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTWILLMSG=0,17,0
>simcomwillmessage
OK
```

16.2.8 AT+CMQTTDISC Disconnect from server

AT+CMQTTDISC Disconnect from server	
Test Command **AT+CMQTTDISC=?**	Response **+CMQTTDISC: (0-1),(0,60-180)** **OK**
Read Command **AT+CMQTTDISC?**	Response **+CMQTTDISC: 0,<disc_state>** **+CMQTTDISC: 1,<disc_state>** **OK**
Write Command **AT+CMQTTDISC=<client_index>,<timeout>**	Response **OK** **+CMQTTDISC: <client_index>,<err>** or **+CMQTTDISC: <client_index>,<err>** **OK** or **+CMQTTDISC: <client_index>,<err>** **ERROR** or

	ERROR
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<timeout>	The timeout value for disconnection. The unit is second. The range is 60s to 180s. The default value is 0s (not set the timeout value)
<disc_state>	1 - disconnection 0 - connection
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTDISC=0,120
OK

+CMQTTDISC: 0,0
```

16.2.9 AT+CMQTTCONNECT Connect to MQTT server

AT+CMQTTCONNECT Connect to MQTT server	
Test Command **AT+CMQTTCONNECT=?**	Response **+CMQTTCONNECT: (0-1),(9-256),(1-64800),(0-1)** **OK**
Read Command **AT+CMQTTCONNECT?**	Response **+CMQTTCONNECT: 0[,<server_addr>,<keepalive_time>,<clean_session>[,<user_name>[,<pass_word>]]]** **+CMQTTCONNECT: 1[,<server_addr>,<keepalive_time>,<clean_session>[,<user_name>[,<pass_word>]]]** **OK**
Write Command **AT+CMQTTCONNECT=<client_index>,<server_addr>,<keepalive_time>,<clean_sess**	Response **OK** **+CMQTTCONNECT: <client_index>,<err>**

ion>[,<user_name>[,<pass_word>]]	or **+CMQTTCONNECT: <client_index>,<err>** **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<server_addr>	The string that described the server address and port. The range of the string length is 9 to 256 bytes. The string should be like this “tcp://116.247.119.165:5141”, must begin with “tcp://”. If the <server_addr> not include the port, the default port is 1883.
<keepalive_time>	The time interval between two messages received from a client. The client will send a keep-alive packet when there is no message sent to server after song long time. The range is from 1s to 64800s (18 hours)
<clean_session>	The clean session flag. The value range is from 0 to 1, and default value is 0. 0 - the server must store the subscriptions of the client after it disconnected. This includes continuing to store QoS 1 and QoS 2 messages for the subscribed topics so that they can be delivered when the client reconnects. The server must also maintain the state of in-flight messages being delivered at the point the connection is lost. This information must be kept until the client reconnects. 1 - the server must discard any previously maintained information about the client and treat the connection as "clean". The server must also discard any state when the client disconnects.
<user_name>	The user name identifies the name of the user which can be used for authentication when connecting to server. The string length is from 1 to 256 bytes.
<password>	The password corresponding to the user which can be used for authentication when connecting to server. The string length is from 1 to 256 bytes.
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTCONNECT=0,"tcp://hooleeping.com:8883",60,1
OK
```

```
+CMQTTCONNECT: 0,0
```

NOTE

- If you don't set the SSL context by AT+CMQTTSSLCFG before connecting a SSL/TLS MQTT server by AT+CMQTTCONNECT, it will use the <client_index> (the 1st parameter of AT+CMQTTCONNECT) SSL context when connecting to the server.

16.2.10 AT+CMQTTTOPIC Input the publish message topic

AT+CMQTTTOPIC Input the publish message topic	
Test Command **AT+CMQTTTOPIC=?**	Response **+CMQTTTOPIC: (0-1),(1-1024)** **OK**
Write Command **AT+CMQTTTOPIC=<client_index>,<req_length>**	Response **>** **<input data here>** **OK** or **+CMQTTTOPIC: <client_index>,<err>** **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<req_length>	The length of input topic data. The publish message topic should be UTF-encoded string. The range is from 1 to 1024 bytes.
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTTOPIC=0,11
>simcomtopic
OK
```

NOTE

- The topic will be clean after execute AT+CMQTTPUB

16.2.11 AT+CMQTTPAYLOAD Input the publish message body

AT+CMQTTPAYLOAD Input the publish message body	
Test Command **AT+CMQTTPAYLOAD=?**	Response **+CMQTTPAYLOAD: (0-1),(1-10240)** **OK**
Write Command **AT+CMQTTPAYLOAD=<client_index>,<req_length>**	Response **>** **<input data here>** **OK** **+CMQTTPAYLOAD: <client_index>,<err>** **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<req_length>	The length of input message data. The publish message should be UTF-encoded string. The range is from 1 to 10240 bytes
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTPAYLOAD=0,13
>simcompayload
OK
```

NOTE

- The payload will be clean after execute AT+CMQTTPUB

16.2.12 AT+CMQTTPUB Publish a message to server

AT+CMQTTPUB Publish a message to server	
Test Command **AT+CMQTTPUB=?**	Response **+CMQTTPUB: (0-1),(0-2),(60-180),(0-1),(0-1)** **OK**
Write Command **AT+CMQTTPUB=<client_index>,<qos>,<pub_timeout>[,<ratained> [,<dup>]]**	Response **OK** **+CMQTTPUB: <client_index>,<err>** or **+CMQTTPUB: <client_index>,<err>** **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<qos>	The publish message's qos. The range is from 0 to 2. 0 – at most once 1 – at least once 2 – exactly once
<pub_timeout>	The publishing timeout interval value. Since the client publish a message to server, it will report failed if the client receive no response from server after the timeout value seconds. The range is from 60s to 180s
<ratained>	The retain flag of the publish message. The value is 0 or 1. The default value is 0. When a client sends a PUBLISH to a server, if the retain flag is set to

	1, the server should hold on to the message after it has been delivered to the current subscribers
<dup>	The dup flag to the message. The value is 0 or 1. The default value is 0. The flag is set when the client or server attempts to re-deliver a message
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTPUB=0,0,120
OK

+CMQTTPUB: 0,0
```

16.2.13 AT+CMQTTSUBTOPIC Input a subscribe message topic

AT+CMQTTSUBTOPIC Input a subscribe message topic	
Test Command **AT+CMQTTSUBTOPIC=?**	Response **+CMQTTSUBTOPIC: (0-1),(1-1024),(0-2)** **OK**
Write Command **AT+CMQTTSUBTOPIC=<client_index>,<req_length>,<qos>**	Response **>** **<input data here>** **OK** or **+CMQTTSUBTOPIC: <client_index>,<err>** **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<req_length>	The length of input topic data. The publish message topic should be UTF-encoded string. The range is from 1 to 1024 bytes.

	NOTE: The max length of the total cached topics is 5120
<qos>	The publish message's qos. The range is from 0 to 2. 0 – at most once 1 – at least once 2 – exactly once
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTSUBTOPIC=0,11,0
>simcomtopic
OK
```

NOTE

- The topic will be clean after execute AT+CMQTTSUB.

16.2.14 AT+CMQTTSUB Subscribe a message to server

AT+CMQTTSUB Subscribe a message to server	
Test Command **AT+CMQTTSUB=?**	Response **+CMQTTSUB: (0-1),(1-1024),(0-2),(0-1)** **OK**
Write Command /*subscribe one or more topicswhichinputby AT+CMQTTSUBTOPIC*/ **AT+CMQTTSUB=<client_index>[,<dup>]**	Response **OK** **+CMQTTSUB: <client_index>,<err>** or **+CMQTTSUB: <client_index>,<err>** **ERROR** or **ERROR**
Write Command /* subcribe one topic*/ **AT+CMQTTSUB=<client_index>,<reqLength>,<qos>[,<dup>]**	Response **>** **<input data here>** **OK** **+CMQTTSUB: <client_index>,<err>** or

	+CMQTTSUB: <client_index>,<err> **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<req_length>	The length of input topic data. The message topic should be UTF-encoded string. The range is from 1 to 1024 bytes.
<qos>	The publish message's qos. The range is from 0 to 2. 0 – at most once 1 – at least once 2 – exactly once
<dup>	The dup flag to the message. The value is 0 or 1. The default value is 0. The flag is set when the client or server attempts to re-deliver a message.
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTSUB=0
OK

+CMQTTSUB: 0,0
```

16.2.15 AT+CMQTTUNSUBTOPIC Input a unsubscribe message topic

AT+CMQTTUNSUBTOPIC Input a unsubscribe message topic	
Test Command **AT+CMQTTUNSUBTOPIC=?**	Response **+CMQTTUNSUBTOPIC: (0-1),(1-1024)** **OK**
Write Command **AT+CMQTTUNSUBTOPIC=<client_index>,<req_length>**	Response **>** **<input data here>** **OK**

	or **+CMQTTUNSUBTOPIC: <client_index>,<err>** **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<req_length>	The length of input topic data. The publish message topic should be UTF-encoded string. The range is from 1 to 1024 bytes.
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTUNSUBTOPIC=0,11
>simcomtopic
OK
```

NOTE

- The max length of the total cached topics is 5120.
- The topic will be clean after execute AT+CMQTTUNSUB

16.2.16 AT+CMQTTUNSUB Unsubscribe a message to server

AT+CMQTTUNSUB Unsubscribe a message to server	
Test Command **AT+CMQTTUNSUB=?**	Response **+CMQTTUNSUB: (0-1),(1-1024),(0-1)** **OK**
Write Command /* unsubscribe one or more topics which input by AT+CMQTTUNSUBTOPIC*/	Response **OK** **+CMQTTUNSUB: <client_index>,<err>**

AT+CMQTTUNSUB=<client_index>,<dup>	or **+CMQTTUNSUB: <client_index>,<err>** **ERROR** or **ERROR**
Write Command /* unsubscribe one topic*/ **AT+CMQTTUNSUB=<client_index>,<reqLength>,<dup>**	Response **>** **<input data here>** **OK** **+CMQTTUNSUB: <client_index>,<err>** or **+CMQTTUNSUB: <client_index>,<err>** **ERROR** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	120000ms
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<req_length>	The length of input topic data. The message topic should be UTF-encoded string. The range is from 1 to 1024 bytes.
<dup>	The dup flag to the message. The value is 0 or 1. The default value is 0. The flag is set when the client or server attempts to re-deliver a message.
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTUNSUB =0,0
OK

+CMQTTUNSUB: 0,0
```

16.2.17 AT+CMQTTCFG Configure the MQTT Context

AT+CMQTTCFG Configure the MQTT Context	
Test Command **AT+CMQTTCFG=?**	Response **+CMQTTCFG: "checkUTF8",(0-1),(0-1)** **+CMQTTCFG: "optimeout",(0-1),(20-120)** **OK**
Read Command **AT+CMQTTCFG?**	Response **+CMQTTCFG: 0,<checkUTF8_flag>,<optimeout_val>** **+CMQTTCFG: 1,<checkUTF8_flag>,<optimeout_val>** **OK**
Write Command /*Configure the check UTF8 flag of the specified MQTT client context*/ **AT+CMQTTCFG="checkUTF8",<client_index>,<checkUTF8_flag>**	Response **OK** or **+CMQTTCFG: <client_index>,<err>** **OK** or **ERROR**
Write Command /*Configure the max timeout interval of the send or receive data operation*/ **AT+CMQTTCFG="optimeout",<client_index>,<optimeout_val>**	Response **OK** or **+CMQTTCFG: <client_index>,<err>** **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<checkUTF8_flag>	The flag to indicate whether to check the string is UTF8 coding or not, the default value is 1. 0 – Not check UTF8 coding. 1 – Check UTF8 coding.
<optimeout_val>	The max timeout interval of sending or receiving data operation. The range is from 20 seconds to 120 seconds, the default value is 120 seconds.
<err>	The result code, please refer to chapter 16.3.1

Example

```
AT+CMQTTCFG="checkUTF8",0,0
OK
AT+CMQTTCFG="optimeout",0,120
OK
```

NOTE

- It must be called before AT+CMQTTCONNECT and after AT+CMQTTACCQ. The setting will be cleared after AT+CMQTTREL

16.3 Summary of result codes for MQTT(S)

16.3.1 Summary of Command result <err> codes

Code of <err>	Meaning
0	operation succeeded
1	failed
2	bad UTF-8 string
3	sock connect fail
4	sock create fail
5	sock close fail
6	message receive fail
7	network open fail
8	network close fail
9	network not opened
10	client index error
11	no connection
12	invalid parameter
13	not supported operation
14	client is busy
15	require connection fail
16	sock sending fail
17	timeout
18	topic is empty
19	client is used

20	client not acquired
21	client not released
22	length out of range
23	network is opened
24	packet fail
25	DNS error
26	socket is closed by server
27	connection refused: unaccepted protocol version
28	connection refused: identifier rejected
29	connection refused: server unavailable
30	connection refused: bad user name or password
31	connection refused: not authorized
32	handshake fail
33	not set certificate
34	open SSL session failed

16.3.2 Summary of Unsolicited Result Codes

Unsolicited codes	Description
+CMQTTCONNLOST: <client_index>,<cause>	When client disconnect passively, URC "+CMQTTCONNLOST" will be reported, then user need to connect MQTT server again.
+CMQTTPING: <client_index>,<err>	When send ping (which keep-alive to the server) to server failed, the module will report this URC. If received this message, you should disconnect the connection and re-connect
+CMQTTNONET	When the network is become no network, the module will report this URC. If received this message, you should restart the MQTT service by AT+CMQTTSTART.
+CMQTTRXSTART: <client_index>,<topic_total_len>,<payload_total_len> **+CMQTTRXTOPIC: <client_index>,<sub_topic_len><sub_topic>** /*for long topic, split to multiple packets to report*/ **[<CR><LF>+CMQTTRXTOPIC: <client_index>,<sub_topic_len><sub_topic>]**	If a client subscribes to one or more topics, any message published to those topics are sent by the server to the client. The following URC is used for transmitting the message published from server to client. 1)**+CMQTTRXSTART: <client_index>,<topic_total_len>,<payload_total_len>** At the beginning of receiving published message, the module will report this to user, and indicate client index with <client_index>, the topic total length with <topic_total_len> and the payload total length with <payload_total_len>. 2)**+CMQTTRXTOPIC: <client_index>,<sub_topic_len>\r\n<sub_topic>**

+CMQTTRXPAYLOAD: <client_index>,<sub_payload_len> **<sub_payload>** /*for long payload, split to multiple packets to report*/ **[+CMQTTRXPAYLOAD: <client_index>,<sub_payload_len>** **<sub_payload>]** **+CMQTTRXEND: <client_index>**	After the command "+CMQTTRXSTART" received, the module will report the second message to user, and indicate client index with <client_index>, the topic packet length with <sub_topic_len> and the topic content with <sub_topic> after "\r\n". For long topic, it will be split to multiple packets to report and the command "+CMQTTRXTOPIC" will be send more than once with the rest of topic content. The sum of <sub_topic_len> is equal to <topic_total_len>. 3)**+CMQTTRXPAYLOAD: <client_index>,<sub_payload_len>\r\n<sub_payload>** After the command "+CMQTTRXTOPIC" received, the module will send third message to user, and indicate client index with <client_index>, the payload packet length with <sub_payload_len> and the payload content with <sub_payload> after "\r\n". For long payload, the same as "+CMQTTRXTOPIC". 4) **+CMQTTRXEND: <client_index>** At last, the module will send fourth message to user and indicate the topic and payload have been transmitted completely.

Defined Values

<client_index>	A numeric parameter that identifies a client. The range of permitted values is 0 to 1.
<cause>	The cause of disconnection. 1 – Socket is closed passively. 2 – Socket is reset. 3 – Network is closed.
<topic_total_len>	The length of message topic received from MQTT server. The range is from 1 to 1024 bytes.
<payload_total_len>	The length of message body received from MQTT server. The range is from 1 to 10240 bytes.
<sub_topic_len>	The sub topic packet length, The sum of <sub_topic_len> is equal to <topic_total_len>.
<sub_topic>	The sub topic content.
<sub_payload_len>	Max length is 1500. The sub message body packet length. The sum of <sub_payload_len> is equal to <payload_total_len>.
<sub_payload>	The sub message body content.
<err>	The result code, please refer to chapter 16.3.1

17. AT Commands for GPS

17.1 Overview of AT Commands for GPS

Command	Description
AT+CGPS	Start/Stop GPS session
AT+CGPSINFO	Get GPS fixed position information
AT+CGPSCOLD	Cold start GPS
AT+CGPSHOT	Hot start GPS
AT+CGPSURL	Set AGPS drfault server URL
AT+CGPSSSL	Set AGPS transport security
AT+CGPSAUTO	Start GPS automatic
AT+CGPSNMEA	Configure NMEA sentence type
AT+CGPSNMEARATE	Set NMEA output rate
AT+CGPSMD	Configure AGPS MO method
AT+CGPSFTM	Start GPS test mode
AT+CGPSDEL	Delete the GPS information
AT+CGPSXE	Enable/Disable GPS XTRA function
AT+CGPSXD	Download XTRA assistant file
AT+CGPSXDAUTO	Download XTRA assistant file automatically
AT+CGPSINFOCFG	Report GPS NMEA-0183 sentence
AT+CGPSPMD	Configure positioning mode
AT+CGPSMSB	Configure based mode switch to standalone
AT+CGPSHOR	Configure positioning desired accuracy
AT+CGPSNOTIFY	LCS respond position request
AT+CGNSSINFO	Get GNSS fixed position information
AT+CGNSSMODE	Configure GNSS support mode
AT+CGPSIPV6	Set AGPS IPV6 Addr&Port
AT+CGPSXTRADATA	Query the validity of the current gpsOne XTRA data

17.2 Detailed Description of AT Commands for GPS

17.2.1 AT+CGPS Start/Stop GPS session

AT+CGPS Start/Stop GPS Session	
Test Command **AT+CGPS=?**	Response **+CGPS: (list of supported <on/off>s),(list of supported <mode>s)** OK
Read Command **AT+CGPS?**	Response **+CGPS: <on/off>,<mode>** **OK**
Write Command **AT+CGPS=<on/off>[,<mode>]**	Response **OK** *If UE-assisted mode, when fixed will report indication:* **+CGPS:<lat>,<lon>,<alt>,<date>,<time>** *If <off>, it will report indication:* **+CGPS:0** or **ERROR**

Defined Values

<on/off>	Values reserved by the present document: 0 – stop GPS session 1 – start GPS session
<mode>	Ignore - standalone mode 1 – standalone mode 2 – UE-based mode 3 – UE-assisted mode
<lat>	Latitude of current position. Unit is in 10^8 degree
<lon>	Longitude of current position. Unit is in 10^8 degree
<alt>	MSL Altitude. Unit is meters.
<date>	UTC Date. Output format is ddmmyyyy
<time>	UTC Time. Output format is hhmmss.s
<unconfidence>	Unconfidence of the location, GPS fixed report 39, cell fixed report 100.
<uncertainty_meter>	Uncertainty meters.

Example

```
AT+CGPS?
+CGPS:1
```

```
OK
AT+CGPS=1
OK
```

NOTE

- Output of NMEA sentences is automatic; no control via AT commands is provided. If executing AT+CGPS=1, the GPS session will choose cold or hot start automatically.
- UE-based and UE-assisted mode depend on URL (AT+CGPSURL). When UE-based mode fails, it will switch standalone mode.
- UE-assisted mode is singly fixed. Standalone and UE-based mode is consecutively fixed.
- After the GPS closed, it should to wait about 2s~30s for start again. Reason： If the signal conditions are right (strong enough signals to allow ephemeris demodulation) or ephemeris demodulation is on going, sometimes MGP will stay on longer in order to demodulate more ephemeris. This will help the engine provide faster TTFF and possibly better yield later (up to 2 hours), because it has the benefit of more ephemeris available.
- For SIM7600E-H-M2/SIM7600SA-H-M2/SIM7600A-H-M2 module, GPS started should be decided by the physical switch of GPS flight mode in the module firstly. Close the switch, GPS will be started automatically, then you can open or close gps by AT command, otherwize, GPS could not be started in any way.

17.2.2 AT+CGPSINFO Get GPS fixed position information

AT+CGPSINFO Get GPS fixed position infomation	
Test Command **AT+CGPSINFO=?**	Response **+CGPSINFO: (scope of <time>)** **OK**
Read Command **AT+CGPSINFO?**	Response **+CGPSINFO: <time>** **OK**
Write Command **AT+CGPSINFO=<time>**	Response **OK** **+CGPSINFO:[<lat>],[<N/S>],[<log>],[<E/W>],[<date>],[<UTC time>],[<alt>],[<speed>],[<course>]** *If <off>, it will report indication:* **OK** *(if <time>=0)*
Execution Command **AT+CGPSINFO**	Response **+CGPSINFO:[<lat>],[<N/S>],[<log>],[<E/W>],[<date>],[<UTC time>],[<alt>],[<speed>],[<course>]**

	OK

Defined Values

<lat>	Latitude of current position. Output format is ddmm.mmmmmm
<N/S>	N/S Indicator, N=north or S=south
<log>	Longitude of current position. Output format is dddmm.mmmmmm
<E/W>	E/W Indicator, E=east or W=west
<date>	Date. Output format is ddmmyy
<UTC time>	UTC Time. Output format is hhmmss.s
<alt>	MSL Altitude. Unit is meters.
<speed>	Speed Over Ground. Unit is knots.
<course>	Course. Degrees.
<time>	The range is 0-255, unit is second, after set <time> will report the GPS information every the seconds.

Example

```
AT+CGPSINFO=?
+CGPSINFO: (0-255)

OK
AT+CGPSINFO?
+CGPSINFO: 0

OK
AT+CGPSINFO
+CGPSINFO:3113.343286,N,12121.234064,E,250311,072809.3,44.1,0.0,0

OK
```

17.2.3 AT+CGPSCOLD Cold Start GPS

AT+CGPSCOLD Cold Start GPS	
Test Command **AT+CGPSCOLD=?**	Response **OK**
Execution Command **AT+CGPSCOLD**	Response **OK**

Example

```
AT+CGPSCOLD=?
OK
AT+CGPSCOLD
OK
```

NOTE

- Before using this command,it must use AT+CGPS=0 to stop GPS session.

17.2.4 AT+CGPSHOT Hot Start GPS

AT+CGPSHOT Hot Start GPS	
Test Command **AT+CGPSHOT=?**	Response **OK**
Execution Command **AT+CGPSHOT**	Response **OK**

Example

```
AT+CGPSHOT=?
OK
AT+CGPSHOT
OK
```

NOTE

- Before using this command,it must use AT+CGPS=0 to stop GPS session.

17.2.5 AT+CGPSURL Set AGPS default server URL

AT+CGPSURL Set AGPS default server URL	
Test Command **AT+CGPSURL=?**	Response **OK**
Read Command **AT+CGPSURL?**	Response **+CGPSURL**: <**URL**> **OK**

Write Command	Response
AT+CGPSURL=<URL>	**OK** or **ERROR**

Defined Values

<URL>	AGPS default server URL. It needs double quotation marks. NOTE: Max length of URL is 128.

Example

```
AT+CGPSURL=”123.123.123.123:8888”
OK
AT+CGPSURL?
+CGPSURL: ”123.123.123.123:8888”
OK
```

NOTE

- It will take effect only after restarting.

17.2.6 AT+CGPSSSL Set AGPS transport security

AT+CGPSSSL Set AGPS transport security	
Test Command **AT+CGPSSSL=?**	Response **+CGPSSSL: (list of supported<SSL>s)** **OK**
Read Command **AT+CGPSSSL?**	Response **+CGPSSSL**: **<SSL>** **OK**
Write Command **AT+CGPSSSL=<SSL>**	Response **OK** or **ERROR**

Defined Values

<SSL>	0 – don’t use certificate

	1	– use certificate

Example

```
AT+CGPSSSL=0
OK
```

17.2.7 AT+CGPSAUTO Start GPS automatic

AT+CGPSAUTO Start GPS automatic	
Test Command **AT+CGPSAUTO=?**	Response **+CGPSAUTO: (list of supported<auto>s)** **OK**
Read Command **AT+CGPSAUTO?**	Response **+CGPSAUTO**: **<auto>** **OK**
Write Command **AT+CGPSAUTO=<auto>**	Response **OK** or **ERROR**

Defined Values

<auto>	0	– Non-automatic
	1	– automatic

Example

```
AT+CGPSAUTO=1
OK
```

NOTE

- If GPS start automatically, its operation mode is standalone mode..

17.2.8 AT+CGPSNMEA Configure NMEA sentence type

AT+CGPSNMEA Configure NMEA sentence type	
Test Command **AT+CGPSNMEA=?**	Response **+CGPSNMEA: (list of supported<auto>s)** **OK**
Read Command **AT+CGPSNMEA?**	Response **+CGPSNMEA: <nmea>** **OK**
Write Command **AT+CGPSNMEA=<nmea>**	Response **OK** or *If GPS engine is running:* **ERROR**

Defined Values

<auto>	Range – 0 to 262143 Each bit enables an NMEA sentence output as follows: Bit 0 – GPGGA (global positioning system fix data) Bit 1 – GPRMC (recommended minimum specific GPS/TRANSIT data) Bit 2 – GPGSV (GPS satellites in view) Bit 3 – GPGSA (GPS DOP and active satellites) Bit 4 – GPVTG (track made good and ground speed) Bit 5 – PQXFI (Global Positioning System Extended Fix Data.) Bit 6 – GLGSV (GLONASS satellites in view GLONASS fixes only) Bit 7 – GNGSA (1. GPS/2. Glonass/3. GALILE DOP and Active Satellites.) Bit 8 – GNGNS (fix data for GNSS receivers;output for GPS,GLONASS,GALILEO) Bit 9 – Reserved Bit 10 – GAGSV (GALILEO satellites in view) Bit 11 –Reserved Bit 12 –Reserved Bit 13 –Reserved Bit 14 –Reserved Bit 15 –Reserved, Bit 16 –BDGSA/PQGSA (BEIDOU/QZSS DOP and active satellites) Bit 17 –BDGSV/PQGSV (BEIDOUQZSS satellites in view) Set the desired NMEA sentence bit(s). If multiple NMEA sentence formats are desired, “OR” the desired bits together.

	NOTE: Reserved default 0, set invalid.

Example

```
AT+CGPSNMEA=200191
OK
```

NOTE

- If nmea bit 2 GPGSV doesn't configure, GPGSV sentence also doesn't output on AT/modem port even set AT+CGPSFTM=1.
- Module should reboot to take effect.

17.2.9 AT+CGPSNMEARATE Set NMEA output rate

AT+CGPSNMEARATE Set NMEA output rate	
Test Command **AT+CGPSNMEARATE=?**	Response **+CGPSNMEARATE: (list of supported<rate>)** **OK**
Read Command **AT+CGPSNMEARATE?**	Response **+CGPSNMEARATE: <rate>** **OK**
Write Command **AT+CGPSNMEA=<rate>**	Response **OK** or **ERROR**

Defined Values

<rate>	0	output rate 1HZ
	1	output rate 10HZ

Example

```
AT+CGPSNMEARATE=1
OK
```

17.2.10 AT+CGPSMD Configure AGPS MO method

AT+CGPSMD Configure AGPS MO method	
Test Command **AT+CGPSMD=?**	Response **+CGPSMD: (scope of<method>)** **OK**
Read Command **AT+CGPSMD?**	Response **+CGPSMD**: <method> **OK**
Write Command **AT+CGPSMD=<method>**	Response **OK** or *If GPS engine is running:* **ERROR**

Defined Values

<method>	0 – Control plane 1 – User plane

Example

```
AT+CGPSMD=1
OK
```

17.2.11 AT+CGPSFTM Start GPS test mode

AT+CGPSFTM Start GPS test mode	
Test Command **AT+CGPSFTM=?**	Response **OK**
Read Command **AT+CGPSFTM?**	Response **+CGPSFTM**: <on/off> **OK**
Write Command **AT+CGPSFTM=<on/off>**	Response **OK** or **ERROR**

Defined Values

<on/off>	0 – Close test mode 1 – Start test mode
<CNo>	Satellite CNo value. Floating value.
URC format	$GPGSV[,<SV>,<CNo>][...] $GLGSV[,<SV>,<CNo>][...] $BDGSV[,<SV>,<CNo>][...] $GAGSV[,<SV>,<CNo>][...] $PQGSV[,<SV>,<CNo>][...]

Example

```
AT+CGPSFTM=1
OK
$GLGSV,78,20.6,66,25.6,77,21.6,79,21.9,67,26.2,68,23.6

$GPGSV,10,36.3,12,33.5,14,26.5,15,27.0,18,30.6,20,29.4,21,14.9,
24,32.8,25,30.6,31,29.1,32,27.0

$BDGSV,201,28.7,204,29.0,206,27.3,207,25.9,209,25.0,210,18.5
```

NOTE

- If test mode starts, the URC will report on AT port, Modem port and UART port.
- If testing on actual signal, <SV> should be ignored, and GPS must be started by AT+CGPS, AT+CGPSCOLD or AT+CGPSHOT.
- If testing on GPS signal simulate equipment, <SV> must be choiced, and GPS will start automatically.
- URC sentence will report every 1 second.

17.2.12 AT+CGPSDEL Delete the GPS information

AT+CGPSDEL Delete the GPS infomation	
Test Command **AT+CGPSDEL=?**	Response **OK**
Execution Command **AT+CGPSDEL**	Response **OK**

Example

```
AT+CGPSDEL=?
OK
AT+CGPSDEL
OK
```

NOTE

- This command must be executed after GPS stopped

17.2.13 AT+CGPSXE Enable/Disable GPS XTRA function

AT+CGPSXE Enable/Disable GPS XTRA function	
Test Command **AT+CGPSXE=?**	Response **+CGPSXE: (list of supported<on/off>s)** **OK**
Read Command **AT+CGPSXE?**	Response **+CGPSXE**: **<on/off>** **OK**
Write Command **AT+CGPSXE=<on/off>**	Response **OK** or **ERROR**

Defined Values

<on/off>	0 – Disable GPS XTRA 1 – Enable GPS XTRA

Example

```
AT+CGPSXE=?
+CGPSXE: (0-1)

OK
AT+CGPSXE=0
OK
```

NOTE

- XTRA function must download the assistant file from network by HTTP, so the APN must be set by AT+CGDCONT command.

17.2.14 AT+CGPSXD Download XTRA assistant file

AT+CGPSXD Download XTRA assistant file	
Test Command **AT+CGPSXD=?**	Response **+CGPSXD: (list of supported<server>)** **OK**
Read Command **AT+CGPSXD?**	Response **+CGPSXD**: **<server>** **OK**
Write Command **AT+CGPSXD=<server>**	Response **OK** **+CGPSXD : <resp>** or **+CGPSXD : <resp>** **ERROR**

Defined Values

<server>	0 – XTRA primary server (precedence) 1 – XTRA secondary server 2 – XTRA tertiary server
<resp>	refer to Unsolicited XTRA download Codes

Example

```
AT+CGPSXD=?
+CGPSXD: (0-2)

OK
AT+CGPSXD=0
OK
+CGPSXD: 0
```

17.2.15 AT+CGPSXDAUTO Download XTRA assistant file automatically

AT+CGPSXDAUTO Download XTRA assistant file automatically	
Test Command **AT+CGPSXDAUTO=?**	Response **+CGPSXDAUTO: (list of supported<on/off>)** **OK**
Read Command **AT+CGPSXDAUTO?**	Response **+CGPSXDAUTO: <on/off>** **OK**
Write Command **AT+CGPSXDAUTO=<on/off>**	Response **OK** Or **ERROR**

Defined Values

<on/off>	0 – disable download automatically 1 – enable download automatically

Example

```
AT+CGPSXDAUTO=?
+CGPSXD: (0,1)

OK
AT+CGPSXDAUTO=0
OK
```

NOTE

- Some URCs will report when downloading, it's same as AT+CGPSXD command.

17.2.16 AT+CGPSINFOCFG Download Report GPS NMEA-0183 sentence

AT+CGPSINFOCFG Download Report GPS NMEA-0183 sentence	
Test Command **AT+CGPSINFOCFG=?**	Response **+CGPSINFOCFG: (scope of <time>),(scope of <config>)**

	OK
Read Command **AT+CGPSINFOCFG?**	Response **+CGPSINFOCFG**: <**time**>,<**config**> **OK**
Write Command **AT+CGPSINFOCFG=<time>[,<config>]**	Response **OK** **(NMEA-0183 Sentence)** **.....** **OK(if<time>=0>)**

Defined Values

<time>	The range is 0-255, unit is second, after set <time> will report the GPS NMEA sentence every the seconds. If <time>=0, module stop reporting the NMEA sentence.
<config>	Range – 0 to 262143 Each bit enables an NMEA sentence output as follows: Bit 0 – GPGGA (global positioning system fix data) Bit 1 – GPRMC (recommended minimum specific GPS/TRANSIT data) Bit 2 – GPGSV (GPS satellites in view) Bit 3 – GPGSA (GPS DOP and active satellites) Bit 4 – GPVTG (track made good and ground speed) Bit 5 – PQXFI (Global Positioning System Extended Fix Data.) Bit 6 – GLGSV (GLONASS satellites in view GLONASS fixes only) Bit 7 – GNGSA (1. GPS/2. Glonass/3. GALILE DOP and Active Satellites.) Bit 8 – GNGNS (fix data for GNSS receivers;output for GPS,GLONASS,GALILEO) Bit 9 – Reserved Bit 10 – GAGSV (GALILEO satellites in view) Bit 11 –Reserved Bit 12 –Reserved Bit 13 –Reserved Bit 14 –Reserved Bit 15 –Reserved, Bit 16 –BDGSA/PQGSA (BEIDOU/QZSS DOP and active satellites) Bit 17 –BDGSV/PQGSV (BEIDOUQZSS satellites in view) Set the desired NMEA sentence bit(s). If multiple NMEA sentence formats are desired, “OR” the desired bits together. Reserved default 0, set invalid.

Example

```
AT+CGPSINFOCFG=?
+CGPSINFO: (0-255),(0-262143)

OK
AT+CGPSINFOCFG=10,31
OK
$GPGSV,4,1,16,04,53,057,44,02,55,334,44,10,61,023,44,05,45,253
,43*7D
$GPGSV,4,2,16,25,10,300,40,17,25,147,40,12,22,271,38,13,28,053
,38*77
$GPGSV,4,3,16,26,09,187,35,23,06,036,34,24,,,,27,,,*7A
$GPGSV,4,4,16,09,,,,31,,,,30,,,,29,,,*7D
$GPGGA,051147.0,3113.320991,N,12121.248076,E,1,10,0.8,47.5,
M,0,M,,*45
$GPVTG,NaN,T,,M,0.0,N,0.0,K,A*42
$GPRMC,051147.0,A,3113.320991,N,12121.248076,E,0.0,0.0,2112
11,,,A*66
$GPGSA,A,3,02,04,05,10,12,13,17,23,25,26,,,1.4,0.8,1.2*3B
```

17.2.17 AT+CGPSPMD Configure positioning mode

AT+CGPSPMD Configure positioning mode	
Test Command **AT+CGPSPMD=?**	Response **+CGPSPMD: (scope of <mode>)** **OK**
Read Command **AT+CGPSPMD?**	Response **+CGPSPMD**: <mode> **OK**
Write Command **AT+CGPSPMD =<mode>**	Response **OK** or **ERROR**

Defined Values

<mode>	Default - 65407 Range - 1 to 65407 Each bit enables a supported positioning mode as follows: Bit 0 – Standalone Bit 1 – UP MS-based

	Bit 2 – UP MS-assisted Bit 3 – CP MS-based (2G) Bit 4 – CP MS-assisted (2G) Bit 5 – CP UE-based (3G) Bit 6 – CP UE-assisted (3G) Bit 7 – NOT USED Bit 8 – UP MS-based (4G) Bit 9 – UP MS-assisted(4G) Bit 10 – CP MS-based (4G) Bit 11 – CP MS-assisted (4G) Set the desired mode sentence bit(s). If multiple modes are desired, "OR" the desired bits together. Example, support standalone, UP MS-based and UP MS-assisted, set Binary value 0000 0111, is 7.

Example

```
AT+CGPSPMD=127
OK
```

NOTE

- Need to restart the module after setting the mode.

17.2.18 AT+CGPSMSB Configure based mode switch to standalone

AT+CGPSMSB Configure based mode switch to standalone	
Test Command **AT+CGPSMSB=?**	Response **+CGPSMSB: (scope of <mode>)** **OK**
Read Command **AT+CGPSMSB?**	Response **+CGPSMSB**: <mode> **OK**
Write Command **AT+CGPSMSB =<mode>**	Response **OK** or **ERROR**

Defined Values

<mode>	0 – Don't switch to standalone mode automatically 1 – Switch to standalone mode automatically

Example

```
AT+CGPSMSB=0
OK
```

NOTE

- This command must be executed after GPS stopped.

17.2.19 AT+CGPSHOR Configure positioning desired accuracy

AT+CGPSHOR Configure positioning desired accuracy	
Test Command **AT+CGPSHOR=?**	Response **+CGPSHOR: (scope of <acc>),(scope of <acc_f>)** **OK**
Read Command **AT+CGPSHOR?**	**+CGPSHOR: <acc>,<acc_f>** **OK**
Write Command **AT+CGPSHOR=<acc>,[,<acc_f>]**	**OK** or **ERROR**

Defined Values

<acc>	Range – 0 to 1800000 Default value is 50
<acc_f>	Reserved

Example

```
AT+CGPSHOR=50
OK
```

NOTE

- This command must be executed after GPS stopped.

17.2.20 AT+CGPSNOTIFY LCS respond positioning request

AT+CGPSNOTIFY LCS respond positioning request	
Test Command **AT+CGPSNOTIFY=?**	Response **+CGPSNOTIFY: (list of supported <resp>s)** **OK**
Write Command **AT+CGPSNOTIFY=<resp>**	Response **OK** or **ERROR**

Defined Values

<resp>	0 – LCS notify verify accept 1 – LCS notify verify deny 2 – LCS notify verify no response

Example

```
AT+CGPSNOTIFY=?
+CGPSNOTIFY: (0-2)

OK
AT+CGPSNOTIFY=0
OK
```

17.2.21 AT+CGNSSINFO LCS Get GNSS fixed position information

AT+CGNSSINFO Get GNSS fixed position information	
Test Command **AT+CGNSSINFO=?**	Response **+CGNSSINFO: (scope of <time>)** **OK**
Read Command **AT+CGNSSINFO?**	Response +**CGNSSINFO**: <time>

	OK
Write Command **AT+CGNSSINFO=<time>**	Response **OK** **+CGNSSINFO:** **[<mode>],[<GPS-SVs>],[<GLONASS-SVs>],[<BEIDOU-SVs>],** **[<lat>],[<N/S>],[<log>],[<E/W>],[<date>],[<UTC-time>],[<alt>],** **[<speed>],[<course>],[<PDOP>],[HDOP],[VDOP]** **OK (if <time>>=0)**
Execution Command **AT+CGNSSINFO**	Response **+CGNSSINFO:** **[<mode>],[<GPS-SVs>],[<GLONASS-SVs>],[<BEIDOU-SVs>],** **[<lat>],[<N/S>],[<log>],[<E/W>],[<date>],[<UTC-time>],[<alt>],** **[<speed>],[<course>],[<PDOP>],[<HDOP>],[<VDOP>]** **OK**

Defined Values

<mode>	Fix mode 2=2D fix 3=3D fix
<GPS-SVs>	GPS satellite valid numbers scope: 00-12
<GLONASS-SVs>	GLONASS satellite valid numbers scope: 00-12
<BEIDOU-SVs>	BEIDOU satellite valid numbers scope: 00-12
<lat>	Latitude of current position. Output format is ddmm.mmmmmm
<N/S>	N/S Indicator, N=north or S=south
<log>	Longitude of current position. Output format is dddmm.mmmmmm
<E/W>	E/W Indicator, E=east or W=west
<date>	Date. Output format is ddmmyy
<UTC-time>	UTC Time. Output format is hhmmss.s
<alt>	MSL Altitude. Unit is meters.
<speed>	Speed Over Ground. Unit is knots.
<course>	Course. Degrees.
<PDOP>	Position Dilution Of Precision.
<HDOP>	Horizontal Dilution Of Precision.
<VDOP>	Vertical Dilution Of Precision.

Example

```
AT+CGNSSINFO=?
+CGNSSINFO: (0-255)

OK
AT+CGNSSINFO?
+CGNSSINFO: 0
```

```
OK
AT+CGNSSINFO
+CGNSSINFO:
2,09,05,00,3113.330650,N,12121.262554,E,131117,091918.0,32.9,
0.0,255.0,1.1,0.8,0.7
OK
AT+CGNSSINFO    (if not fix,will report null)
+CGNSSINFO: ,,,,,,,,,,,,,,,
OK
```

17.2.22 AT+CGNSSMODE LCS Configure GNSS support mode

AT+CGNSSMODE Configure GNSS support mode	
Test Command **AT+CGNSSMODE=?**	Response **+CGNSSMODE: (scope of <gnss_mode>),(scope of <dpo_mode>)** **OK**
Read Command **AT+CGNSSMODE?**	Response **+CGNSSMODE: <gnss_mode>,<dpo_mode>** **OK**
Write Command **AT+CGNSSMODE=<gnss_mode>[,<dpo_mode>]**	Response **OK** or **ERROR**

Defined Values

<gnss_mode>	Range – 0 to 15 Bit0: GLONASS Bit1: BEIDOU Bit2: GALILEO Bit3: QZSS 1: enable 0:disable GPS always support
<dpo_mode>	1: enable DPO 0: disable DPO

Example

```
AT+CGNSSMODE=15,1
```

```
OK
```

NOTE

- Module should reboot to take effective.

17.2.23 AT+CGPSIPV6 Set AGPS IPV6 Addr&Port

AT+CGPSIPV6 Set AGPS IPV6 Addr&Port	
Test Command **AT+CGPSIPV6=?**	Response **OK**
Read Command **AT+CGPSIPV6?**	Response **+CGPSIPV6: <ipv6_addr>,<port>** **OK**
Write Command **AT+CGPSIPV6=<ipv6_addr>,<port>**	Response **OK** or **ERROR**

Defined Values

<ipv6_addr>	AGPS IPV6 addr. It needs double quotation marks.
<port>	AGPS IPV6 port.

Example

```
AT+CGPSIPV6=”2001:0268:1AFF:0000:0000:0000:B6F8:A5D2”,7
275
OK

AT+CGPSIPV6?
+CGPSIPV6:  ”2001:0268:1AFF:0000:0000:0000:B6F8:A5D2”,727
5

OK
```

NOTE

- It will take effect only after restarting.

17.2.24 AT+CGPSXTRADATA Query the validity of the current gpsOne XTRA Data

AT+CGPSXTRADATA Query the validity of the current gpsOne XTRA Data	
Test Command **AT+CGPSXTRADATA=?**	Response **OK**
Read Command **AT+CGPSXTRADATA?**	Response **+CGPSXTRADATA: <xtradatadurtime>,<injecteddatatime>** **OK**

Defined Values

<xtradatadurtime>	Valid time of injected gpsOneXTRA data,unit:minute 0 No gpsOneXTRA file or gpsOneXTRA file is overdue 1-10080 Valid time of gpsOneXTRA file
<injecteddatatime>	Starting time of the valid time of XTRA data, format: "YYYY/MM/DD,hh:mm:ss",e.g. "2019/09/26,15:31:20"

Example

```
AT+CGPSXTRADATA=?
OK

AT+CGPSXTRADATA?
+CGPSXTRADATA: 168,"2019/09/25,05:00:00"

OK
```

NOTE

- It needs to execute AT+CGPSXE to enable before execute the AT+CGPSXTRADATA read.

18. AT Commands for LBS

18.1 Overview of AT Commands for LBS

Command	Description
AT+CLBS	Base station Location
AT+CLBSCFG	Base station Location configure

18.2 Detailed Description of AT Commands for LBS

18.3 AT Commands for Open/Close Network

18.3.1 Overview of AT Commands for Open/Close Network

Command	Description
AT+CNETSTART	Open network
AT+CNETSTOP	Close network
AT+CNETIPADDR	Inquire PDP address

18.3.2 Detailed Description of AT Commands for Open/Close Network

18.3.2.1 AT+CNETSTART Open network

AT+CNETSTART Open network	
Read Command **AT+CNETSTART?**	Response **+CNETSTART: <net_stat>** **OK** or **ERROR**
Execution Command **AT+CNETSTART**	Response **OK** **+CNETSTART**: **<err>** or **+CNETSTART**: **<err>** **OK** or **+CNETSTART**: **<err>** **ERROR** or **ERROR**

Defined Values

<net_state>	a numeric parameter that indicates the state of PDP context activation: 0 network close (deactivated) 1 network is opening 2 network open(activated) 3 network is closing
<err>	The result of operation, 0 is success, other value is failure.

Example

```
AT+CNETSTART?
+CNETSTART: 0

OK
AT+CNETSTART
OK

+CNETSTART: 0
```

18.3.2.2 AT+CNETSTOP Close network

AT+CNETSTOP Close network	
Execution Command **AT+CNETSOP**	Response **OK** **+CNETSTOP: <err>** or **+CNETSTOP: <err>** **OK** or **+CNETSTOP: <err>** **ERROR** or **ERROR**

Defined Values

<err>	The result of operation, 0 is success, other value is failure.

Example

```
AT+CNETSTOP
+CNETSTOP: 0

OK
```

18.3.2.3 AT+CNETIPADDR Inquire PDP address

AT+CNETIPADDR Inquire PDP address	
Read Command **AT+CNETIPADDR?**	Response **+CNETIPADDR: <ip_address>** **OK** or **+CNETIPADDR: <err_info>**

	ERROR or **ERROR**

Defined Values

<ip_address>	A string parameter that identifies the IP address of current active socket PDP.
<err_info>	A string parameter that displays the cause of occurring error.

Example

```
AT+CNETIPADDR?
+CNETIPADDR: 10.71.155.118

OK
```

18.3.3 Unsolicited Open/Close network command <err> Codes

Code of <err>	Description
0	Operation succeeded
1	Unknown error
2	Open network failed
3	Close network failed
4	Network not opened
5	Operation not support
6	Busy
7	Network has been opened
8	Network is also in use

18.3.3.1 AT+CLBS Base station Location

AT+CLBS Base station Location	
Test Command **AT+CLBS=?**	Response **+CLBS: (list of supported <type>s),(range of supported <cid>s),(range of supported <longitude>s),(range of supported <latitude>s),(list of supported <lon_type>s)**

	OK
Write Command **AT+CLBS=<type>,<cid>,[[<longitude>,<latitude>],[<lon_type>]]**	Response 1)<type>=1,get longitude and latitude **+CLBS: <locationcode>[,<longitude>,<latitude>,<acc>]** **OK** 2)<type>=4,get longitude latitude and date time **+CLBS: <locationcode>[,<longitude>,<latitude>,<acc>,<date>,<time>]** **OK** If error is related to ME functionality: **+CME ERROR: <err>**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<type>	1 Use 3 cell's information 4 Get longitude latitude and date time
<cid>	Bearer profile identifier, refer to <pdpidx> of AT+CNACT
<locationcode>	0 Success If the operation failed, the location code is not 0, such as: 1 Location Failed 2 Time Out 3 NET Error 4 DNS Error 5 Service Overdue 6 Authenticate Failed 7 Other Error 80 Report LBS to server success 81 Report LBS to server parameter error 82 Report LBS to server failed
<longitude>	Current longitude in degrees. -180.000000-180.000000
<latitude>	Current latitude in degrees -90.000000-90.000000
<acc>	Positioning accuracy
<lon_type>	The type of longitude and latitude 0 WGS84 1 GCJ02
<times>	Access service times
<date>	Service date

<time>	Service time

Example

```
AT+CLBS=?
+CLBS:
(1,3,4,9),(0-3),(-180.000000-180.000000),(-90.0
00000-90.000000),(0,1)

OK
```

NOTE

- If customers feel that the positioning error is too large, <type>=9 can be used to report this information. The error can be improved by this information.

18.3.3.2 AT+CLBSCFG Base station Location configure

AT+CLBSCFG Base station Location configure	
Test Command **AT+CLBSCFG=?**	Response **+CLBSCFG: (list of supported <operate>s),(range of supported <para>s),<len_value>** **OK**
Write Command **AT+CLBSCFG=<operate>,<para>[,<value>]**	Response **+CLBSCFG: 0,<para>,<value>** **OK** or **OK** If error is related to ME functionality: **+CME ERROR: <err>**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<operate>	0	Read operator
	1	Set operator

<para>	1 Customer ID 2 Times have used positioning command 3 Server's address lbs-simcom.com:3001 lbs-simcom.com:3000 lbs-simcom.com:3002 (Default)
<value>	String type. The value of parameter If <operate> is 1 and <para> is 3, <value> can be set.
<len_value>	Max length of <value>

Example

```
AT+CLBSCFG?
+CLBSCFG: (0-1),3,"Param Value"

OK
```

NOTE

- Server's address of"lbs-simcom.com:3002" is free. The other two servers are charged.
- If you want to use the charged address, the IMEI, customer information and software version must be provided to SIMCom.

19. AT Commands for Hardware

19.1 Overview of AT Commands for Hardware Related

Command	Description
AT+CVALARM	Low and high voltage Alarm
AT+CVAUXS	Set state of the pin named VREG_AUX1
AT+CVAUXV	Set voltage value of the pin named VREG_AUX1
AT+CADC	Read ADC value
AT+CADC2	Read ADC2 value
AT+CMTE	Control the module whether power shutdown when the module's temperature upon the critical temperature
AT+CPMVT	Low and high voltage Power Off
AT+CDELTA	Set the module go to recovery mode
AT+CRIIC	Read values from register of IIC device
AT+CWIIC	Write values to register of IIC device
AT+CBC	Read the voltage value of the power supply
AT+CPMUTEMP	Read the temperature of the module
AT+CFDISK	SD Card/EMMC Flash

19.2 Detailed Description of AT Commands for Hardware Related

Command	Description
AT+CVALARM	Low and high voltage Alarm
AT+CVAUXS	Set state of the pin named VREG_AUX1
AT+CVAUXV	Set voltage value of the pin named VREG_AUX1
AT+CADC	Read ADC value
AT+CADC2	Read ADC2 value
AT+CMTE	Control the module whether power shutdown when the module's temperature upon the critical temperature
AT+CPMVT	Low and high voltage Power Off
AT+CDELTA	Set the module go to recovery mode
AT+CRIIC	Read values from register of IIC device

AT+CWIIC	Write values to register of IIC device
AT+CBC	Read the voltage value of the power supply
AT+CPMUTEMP	Read the temperature of the module
AT+CFDISK	SD Card/EMMC Flash

19.2.1 AT+CVALARM Low and high voltage Alarm

AT+CVALARM Low and high voltage Alarm	
Test Command **AT+CVALARM=?**	Response **+CVALARM: (list of supported <enable>s), (list of supported <low voltage>s), (list of supported high <high voltage>s)** **OK**
Read Command **AT+CVALARM?**	Response **+CVALARM: <enable>,<low voltage>,<high voltage>** **OK**
Write Command **AT+CVALARM=<enable>[,<low voltage>],[<high voltage>]**	Response **OK** or **ERROR**

Defined Values

<enable>	0: Close 1: Open. If voltage < <low voltage>, it will report “UNDER-VOLTAGE WARNNING” every 10s. If voltage > <high voltage>, it will report “OVER-VOLTAGE WARNNING” every 10s.
<low voltage>	Between 3300mV and 4000mV. Default value is 3300.
<high voltage>	Between 4000mV and 4300mV. Default value is 4300.

Example

```
AT+CVALARM?
+CVALARM: 1,3400,4300
OK

AT+CVALARM=?
+CVALARM: (0,1),(3300-4000),(4000-4300)
OK

AT+CVALARM=1,3400,4300
```

OK

19.2.2 AT+CVAUXS Set state of the pin named VREG_AUX1

AT+CVAUXS Set state of the pin named VREG_AUX1	
Test Command **AT+CVAUXS=?**	Response **+CVAUXS: (list of supported <state>s)** **OK**
Read Command **AT+CVAUXS?**	Response **+CVAUXS: <state>** **OK**
Write Command **AT+CVAUXS=<state>**	Response **OK** Or **ERROR**

Defined Values

<state>	0: the pin is closed. 1: the pin is opend(namely, open the pin).

Example

```
AT+CVAUXS?
+CVAUXS: 1
OK

AT+CVAUXS =1
OK
```

NOTE

- For SIM7600E-H-M2/SIM7600SA-H-M2/SIM7600A-H-M2, the default value is 0.

19.2.3 AT+CVAUXV Set voltage value of the pin named VREG_AUX1

AT+CVAUXV Set voltage value of the pin named VREG_AUX1	
Test Command **AT+CVAUXV=?**	Response **+CVAUXV: (list of supported <voltage>s)** **OK**
Read Command **AT+CVAUXV?**	Response **+CVAUXV: <voltage>** **OK**
Write Command **AT+CVAUXV=<voltage>**	Response **OK** or **ERROR**

Defined Values

<voltage>	Voltage value of the pin which is named VREG_AUX1. The unit is in mV. And the value must the multiple of 50mv.

Example

```
AT+CVAUXV =?
+CVAUXV: (1700-3050)
OK

AT+CVAUXV =2800
OK

AT+CVAUXV?
+CVAUXV: 2800
OK
```

19.2.4 AT+CADC Read ADC value

AT+CADC Read ADC value	
Test Command **AT+CADC=?**	Response **+CADC: (range of supported <adc>s)** **OK**
Write Command **AT+CADC=<adc>**	Response **+CADC: <value>**

	OK or **ERROR**

Defined Values

<adc>	ADC type: 0 – raw type. 2 – voltage type(mv)
<value>	Integer type value of the ADC.

Example

```
AT+CADC=?
+CADC: (0,2)
OK

AT+CADC =0
+CADC: 187
OK
```

19.2.5 AT+CADC2 Read ADC2 value

AT+CADC2 Read ADC2 value	
Test Command **AT+CADC2=?**	Response **+CADC2: (range of supported <adc>s)** **OK**
Write Command **AT+CADC2=<adc>**	Response **+CADC2: <value>** **OK** or **ERROR**

Defined Values

<adc>	ADC2 type: 0 – raw type. 2 – voltage type(mv)
<value>	Integer type value of the ADC2.

Example

```
AT+CADC2=?
+CADC2: (0,2)
OK

AT+CADC2=0
+CADC2: 187
OK
```

19.2.6 AT+CMTE Control the module whether power shutdown when the module's temperature upon the critical temperature

AT+CMTE Control the module whether power shutdown when the module's temperature upon the critical temperature	
Test Command **AT+CMTE=?**	Response **+CMTE: (list of supported<on/off>s)** **OK**
Read Command **AT+CMTE?**	Response **+CMTE: <on/off >** **OK**
Write Command **AT+CMTE=<on/off >**	Response **OK** or **ERROR**

Defined Values

<on/off>	0 – Disable temperature detection 1 – Enable temperature detection

Example

```
AT+CMTE?
+CMTE: 1
OK

AT+CMTE =1
OK
```

```
AT+CMTE=?
+CMTE: (0/1)
OK
```

NOTE

- When temperature is extreme high or low, product will power off.
- URCs indicating the alert level “+CMTE:-1” or “+CMTE:1” are intended to enable the user to take appropriate precaution, such as protect the module from exposure to extreme conditions, or save or back up data etc
- Level “+CMTE:-2”or “+CMTE:2” URCs are followed by immediate shutdown.

19.2.7 AT+CPMVT Low and high voltage Power Off

AT+CPMVT Low and high voltage Power Off	
Test Command **AT+CPMVT=?**	Response **+CPMVT: (list of supported <enable>s), (list of supported <low voltage>s), (list of supported <high voltage>s)** **OK**
Read Command **AT+CPMVT?**	Response **+CPMVT: <enable>,<low voltage>, <high voltage>** **OK**
Write Command **AT+CPMVT=<enable>[,<low voltage>],[<high voltage>]**	Response **OK** Or **ERROR**

Defined Values

<enable>	0: Close 1: Open. If voltage < <low voltage>, it will report “UNDER-VOLTAGE WARNNING POWER DOWN” and power off the module. If voltage > <high voltage>, it will report “OVER-VOLTAGE WARNNING POWER DOWN” and power off the module
<low voltage>	Between 3200mV and 4000mV. Default value is 3200.
<high voltage>	Between 4000mV and 4300mV. Default value is 4300.

Example

```
AT+CPMVT=1,3400,4300
OK

AT+CPMVT?
+CVALARM: 1,3400,4300
OK

AT+CPMVT=?
+CVALARM: (0,1),(3300-4000),(4000-4300)
OK
```

19.2.8 AT+CDELTA Set the module go to recovery mode

AT+CDELTA Set the module go to recovery mode	
Write Command **AT+CDELTA**	Response **OK** or **ERROR**

Example

```
AT+CDELTA
OK
```

NOTE

- the command will write flag to the module and reboot the module, then the module will reboot and read the flag and enter recovery mode to update the firmware.

19.2.9 AT+CRIIC Read values from register of IIC device

AT+CRIIC Read values from register of IIC device	
Test Command **AT+CRIIC=?**	Response **OK**
Write Command **AT+CRIIC=<addr>,<reg>,<len>**	Response **+CRIIC: <data>** **OK**

	or **ERROR**

Defined Values

<addr>	Device address. Input format must be hex, such as 0xFF.
<reg>	Register address. Input format must be hex, such as 0xFF.
<len>	Read length. Range:1-4; unit:byte.
<data>	Data read. Input format must be hex, such as 0xFF.

Example

```
AT+CRIIC=0x34, 0x02, 2
+CRIIC: 0x01,0x5d
OK
```

19.2.10 AT+CWIIC Write values to register of IIC device

AT+CWIIC Write values to register of IIC device	
Test Command **AT+CWIIC=?**	Response **OK**
Write Command **AT+CWIIC=<addr>,<reg>,<data>,<len>**	Response **OK** or **ERROR**

Defined Values

<addr>	Device address. Input format must be hex, such as 0xFF.
<reg>	Register address. Input format must be hex, such as 0xFF.
<len>	Read length. Range: 1-4; unit: byte.
<data>	Data written. Input format must be hex, such as 0xFF – 0xFFFFFFFF.

Example

```
AT+CWIIC=0x34, 0x03, 0x5d, 1
OK
```

19.2.11 AT+CBC Read the voltage value of the power supply

AT+CBC Read the voltage value of the power supply	
Read Command **AT+CBC**	Response **+CBC: <vol>** **OK** or **ERROR**

Defined Values

<vol>	The voltage value, such as 3.8.

Example

```
AT+CBC
+CBC: 3.591V
OK
```

19.2.12 AT+CPMUTEMP Read the temperature of the module

AT+CPMUTEMP Read the temperature of the module	
Read Command **AT+CPMUTEMP**	Response **+CPMUTEMP: <temp>** **OK** or **ERROR**

Defined Values

<temp>	The Temperature value, such as 29.

Example

```
AT+CPMUTEMP
+CPMUTEMP: 29
OK
```

19.2.13 AT+CFDISK SD Card/EMMC Flash

AT+CFDISK SD Card/EMMC Flash	
Test Command **AT+CFDISK=?**	Response **+CFDISK: (1-4)[…]** **OK** or **ERROR**
Read Command **AT+CFDISK?**	Response **+CFDISK: <num>,<size>** **OK** or **ERROR**
Write Command **AT+CFDISK=<num>[,<size>,…]**	Response **OK** or **ERROR**
Write Command (Formatting all partitions) **AT+CFDISK**	Response **OK** or **ERROR**

Defined Values

<num>	Partition size.The unit is KB

Example

```
AT+CFDISK=?
+CFDISK: (1-4)[...]
OK

AT+CFDISK=4,50000,50000,50000
OK

AT+CFDISK
OK

AT+CFDISK?
+CFDISK: 1,50040
+CFDISK: 2,50048
+CFDISK: 3,50048
+CFDISK: 4,3708288
```

```
OK
```

NOTE

- The last partition size does not need to be set. The size of the last partition is the size of the disk remaining.

20. AT Commands for File System

The file system is used to store files in a hierarchical (tree) structure, and there are some definitions and conventions to use the Module.

Local storage space is mapped to “C:”, “D:” for TF card, “E:” for multimedia, “F:” for cache.

NOTE

General rules for naming (both directories and files):

- ✧ The length of actual fully qualified names of directories and files can not exceed 254.
- ✧ Directory and file names can not include the following characters: \ : * ? “ < > | , ;
- ✧ Between directory name and file/directory name, use character “/” as list separator, so it can not appear in directory name or file name.
- ✧ The first character of names must be a letter or a numeral or underline, and the last character can not be period “.” and oblique “/”.
- ✧ 7600M1+1 can not support “D:”and “E:”, if all the following AT are executed, “ERROR” will be returned.

20.1 Overview of AT Commands for File System

Command	Description
AT+FSCD	Select directory as current directory
AT+FSMKDIR	Make new directory in current directory
AT+FSRMDIR	Delete directory in current directory
AT+FSLS	List directories/files in current directory
AT+FSDEL	Delete file in current directory
AT+FSRENAME	Rename file in current directory
AT+FSATTRI	Request file attributes
AT+FSMEM	Check the size of available memory
AT+FSLOCA	Select storage place
AT+FSCOPY	Copy an appointed file
AT+CFTRANRX	Transfer a file to EFS
AT+CFTRANTX	Transfer a file from EFS to host

20.2 Detailed Description of AT Commands for File System

20.2.1 AT+FSCD Select directory as current directory

This command is used to select a directory. The Module supports absolute path and relative path.
Read Command will return current directory without double quotation marks. Support "C:", "D:", "E:", "F:".

AT+FSCD Select directory as current directory	
Test Command **AT+FSCD=?**	Response **OK**
Read Command **AT+FSCD?**	Response **+FSCD: <curr_path>** **OK**
Write Command **AT+FSCD=<path>**	Response **+FSCD: <curr_path>** **OK** or **ERROR**

Defined Values

<path>	String without double quotes, directory for selection.
<curr_path>	String without double quotes, current directory.

NOTE

If **<path>** is "..", it will go back to previous level of directory.

Example

```
AT+FSCD=C:
+FSCD: C:/

OK
AT+FSCD=C:/
+FSCD: C:/
```

```
OK
AT+FSCD?
+FSCD: C:/

OK
AT+FSCD=..
+FSCD: C:/

OK
AT+FSCD=D:
+FSCD: D:/

OK
AT+FSCD?
+FSCD: D:/

OK
```

20.2.2 AT+FSMKDIR Make new directory in current directory

This command is used to create a new directory in current directory. Support "C:", "D:", "E:", "F:".

AT+FSMKDIR Make new directory in current directory	
Test Command **AT+FSMKDIR=?**	Response **OK**
Write Command **AT+FSMKDIR=<dir>**	Response **OK** or **ERROR**

Defined Values

<dir>	String without double quotes, directory name which does not already exist in current directory.

Example

```
AT+FSMKDIR=SIMTech
OK
AT+FSCD?
+FSCD: E:/
```

OK AT+FSLS	
+FSLS: SUBDIRECTORIES Audio SIMTech OK	

20.2.3 AT+FSRMDIR Delete directory in current directory

This command is used to delete existing directory in current directory. Support "C:", "D:", "E:", "F:".

AT+FSRMDIR Delete directory in current directory	
Test Command AT+FSRMDIR=?	Response OK
Write Command AT+FSRMDIR=<dir>	Response OK or ERROR

Defined Values

<dir>	String without double quotes.

Example

AT+FSRMDIR=SIMTech OK AT+FSCD? +FSCD: E:/ OK AT+FSLS +FSLS: SUBDIRECTORIES Audio OK	

20.2.4 AT+FSLS List directories/files in current directory

This command is used to list information of directories and/or files in current directory. Support "C:", "D:", "E:", "F:".

AT+FSLS List directories/files in current directory	
Test Command **AT+FSLS=?**	Response **+FSLS: (list of supported <type>)** **OK**
Read Command **AT+FSLS?**	Response **+FSLS: SUBDIRECTORIES<dir_num>,FILES:<file_num>** **OK**
Write Command **AT+ FSLS=<type>**	Response **[+FSLS: SUBDIRECTORIES:** **<list of subdirectories>** **<CR><LF>]** **[+FSLS: FILES:** **<list of files>** **<CR><LF>]** **OK**
Execution Command **AT+ FSLS**	Response **[+FSLS: SUBDIRECTORIES:** **<list of subdirectories>** **<CR><LF>]** **[+FSLS: FILES:** **<list of files>** **<CR><LF>]** **OK**

Defined Values

<dir_num>	Integer type, the number of subdirectories in current directory.
<file_num>	Integer type, the number of files in current directory.
<type>	0 – list both subdirectories and files 1 – list subdirectories only 2 – list files only

Example

```
AT+FSLS?
+FSLS: SUBDIRECTORIES:2,FILES:2
```

```
OK
AT+FSLS
+FSLS: SUBDIRECTORIES:
FirstDir
SecondDir

+FSLS: FILES:
image_0.jpg
image_1.jpg

OK
AT+FSLS=2
+FSLS: FILES:
image_0.jpg
image_1.jpg

OK
```

20.2.5 AT+FSDEL Delete file in current directory

This command is used to delete a file in current directory. Before do that, it needs to use AT+FSCD select the father directory as current directory. Support "C:", "D:", "E:", "F:".

AT+FSDEL Delete file in current directory	
Test Command **AT+FSDEL=?**	Response **OK**
Write Command **AT+FSDEL=<filename>**	Response **OK** or **ERROR**

Defined Values

<filename>	String with or without double quotes, file name which is relative and already existing. If **<filename>** is *.*, it means delete all files in current directory. If the file path contains non-ASCII characters, the filename parameter should contain a prefix of {non-ascii} and the quotation mark.

Example

```
AT+FSDEL=image_0.jpg
OK
```

20.2.6 AT+FSRENAME Rename file in current directory

This command is used to rename a file in current directory. Support "C:", "D:", "E:", "F:".

AT+FSRENAME Rename file in current directory	
Test Command **AT+FSRENAME=?**	Response **OK**
Write Command **AT+FSRENAME=<old_name>,<new_name>**	Response **OK** or **ERROR**

Defined Values

<old_name>	String with or without double quotes, file name which is existed in current directory. If the file path contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark.
<new_name>	New name of specified file, string with or without double quotes. If the file path contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark.

Example

```
AT+FSRENAME=image_0.jpg, image_1.jpg
OK
AT+FSRENAME="my test.jpg",
{non-ascii}"E6B58BE8AF95E99984E4BBB62E6A7067"
OK
```

20.2.7 AT+FSATTRI Request file attributes

This command is used to request the attributes of file which exists in current directory. Support "C:", "D:", "E:", "F:".

AT+FSATTRI Request file attributes

Test Command **AT+FSATTRI=?**	Response **OK**
Write Command **AT+FSATTRI=<dir>**	Response **+FSATTRI: <file_size>,<create_date>** **OK** or **ERROR**

Defined Values

<filename>	String with or without double quotes, file name which is in current directory. If the file path contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark.
<file_size>	The size of specified file, and the unit is in Byte.
<create_date>	Create date and time of specified file, the format is YYYY/MM/DD HH:MM:SS Week. Week – Mon, Tue, Wed, Thu, Fri, Sat, Sun

Example

```
AT+FSATTRI=image_0.jpg
+FSATTRI: 8604, 2008/04/28 10:24:46 Tue

OK
AT+FSATTRI={non-ascii}"E6B58BE8AF95E99984E4BBB62E6A7067"
+FSATTRI: 6296, 2012/01/06 00:00:00 Sun

OK
```

20.2.8 AT+FSMEM Check the size of available memory

This command is used to check the size of available memory. The response will list total size and used size of local storage space if present and mounted. Support "C:", "D:", "E:", "F:".

AT+FSMEM Check the size of available memory	
Test Command **AT+FSMEM=?**	Response **OK**
Write Command **AT+FSMEM**	Response **+FSMEM: <loctype>:(<total>, <used>)**

	OK

Defined Values

<loctype>	Support "C:", "D:", "E:", "F:".
<total>	The total size of local storage space.The unit of storage space size is in Byte.
<used>	The used size of local storage space.The unit of storage space size is in Byte.

Example

```
AT+FSMEM
+FSMEM: C:(11348480, 2201600)

OK
```

20.2.9 AT+FSLOCA Select storage place

This command is used to set the storage place for media files. Support "C:".

AT+FSLOCA Select storage place	
Test Command **AT+FSLOCA=?**	Response **+FSLOCA: (list of supported <loca>s)** **OK**
Read Command **AT+FSLOCA?**	**+FSLOCA: <loca>** **OK**
Write Command **AT+FSLOCA=<loca>**	Response **OK** or **ERROR**

Defined Values

<loca>	0 – store media files to local storage space (namely "C:/")

Example

```
AT+FSLOCA=0
OK
```

```
AT+FSLOCA?
+FSLOCA: 0

OK
```

20.2.10 AT+FSCOPY Copy an appointed file

This command is used to copy an appointed file on C:/ to an appointed directory on C:/, the new file name should give in parameter. Support "C:","D:", "E:", "F:", but copying from "C:" to "D:", "E:", "F:" or from "D:", "E:", "F:" to "C:" is not supported.

AT+FSCOPY Copy an appointed file	
Test Command **AT+FSCOPY=?**	Response **OK**
Write Command **AT+FSCOPY=<file1>,<file2>[,<sync_mode>]**	Response **Sync mode** **+FSCOPY: <percent><CR><LF>** **[+FSCOPY: <percent><CR><LF>]** **OK** **Async mode** **OK** **+FSCOPY: <percent><CR><LF>** **[+FSCOPY: <percent><CR><LF>]** **+FSCOPY: END<CR><LF>** Or **When error, shows one of the following errors and ERROR** **SD CARD NOT PLUGGED IN** **FILE IS EXISTING** **FILE NOT EXISTING** **DIRECTORY IS EXISTED** **DIRECTORY NOT EXISTED** **FORBID CREATE DIRECTORY UNDER \"C:/\"** **FORBID DELETE DIRECTORY** **INVALID PATH NAME** **INVALID FILE NAME** **SD CARD HAVE NO ENOUGH MEMORY** **EFS HAVE NO ENOUGH MEMORY** **FILE CREATE ERROR** **READ FILE ERROR** **WRITE FILE ERROR**

	ERROR

Defined Values

<file1>	The sources file name or the whole path name with sources file name. If the file path contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark.
<file2>	The destination file name or the whole path name with destination file name. If the file path contains non-ASCII characters, the file path parameter should contain a prefix of {non-ascii} and the quotation mark.
<percent>	The percent of copy done. The range is 0.0 to 100.0
<sync_mode>	The execution mode of the command: 0 – synchronous mode 1 – asynchronous mode

NOTE

1. The **<file1>** and **<file2>** should give the whole path and name, if only given file name, it will refer to current path (**AT+FSCD**) and check the file's validity.
2. If <file2> is a whole path and name, make sure the directory exists, make sure that the file name does not exist or the file name is not the same name as the sub folder name, otherwise return error.
3. **<percent>** report refer to the copy file size. The big file maybe report many times, and little file report less.
4. If **<sync_mode>** is 1, the command will return **OK** immediately, and report final result with **+FSCOPY: END**.

Example

```
AT+FSCD?
+FSCD: C:/

OK
AT+FSCOPY= C:/TESTFILE,COPYFILE    (Copy file TESTFILE on C:/ to C:/COPYFILE)
+FSCOPY: 1.0

+FSCOPY: 100.0

OK
AT+FSCOPY= "my test.jpg", {non-ascii}"E6B58BE8AF95E99984E4BBB62E6A7067"
+FSCOPY:1.0
```

```
+FSCOPY:100.0

OK
```

20.2.11 AT+CFTRANRX Transfer a file to EFS

This command is used to transfer a file to EFS.Support SDcard.

AT+CFTRANRX Transfer a file to EFS	
Test Command **AT+CFTRANRX=?**	Response **+CFTRANRX: [{non-ascii}]"FILEPATH"** **OK**
Write Command **AT+CFTRANRX="<filepath>",<len>**	Response **>** **OK** or **>** **ERROR** or **ERROR**

Defined Values

<filepath>	The path of the file on EFS.
<len>	The length of the file data to send. The range is from 0 to 2147483647.

NOTE

The **<filepath>** must be a full path with the directory path.

Example

```
AT+CFTRANRX="c:/MyDir/t1.txt",10
><input data here>

OK
AT+CFTRANRX="d:/MyDir/t1.txt",10
><input data here>
```

```
OK
```

20.2.12 AT+CFTRANTX Transfer a file from EFS to host

This command is used to transfer a file from EFS to host. Before using this command, the AT+CATR must be used to set the correct port used. Support SDcard.

AT+CFTRANTX Transfer a file from EFS to host	
Test Command **AT+CFTRANTX=?**	Response **+CFTRANTX: [{non-ascii}]"FILEPATH"** **OK**
Write Command **AT+CFTRANTX="<filepath>"[,<location>,<size>]**	Response **[+CFTRANTX: DATA,<len>** **…** **+CFTRANTX: DATA,<len>]** **+CFTRANTX: 0** **OK** or **ERROR**

Defined Values

<filepath>	The path of the file on EFS.
<len>	The length of the following file data to output.
<location>	The beginning of the file data to output.
<size>	The length of the file data to output.

NOTE

The **<filepath>** must be a full path with the directory path.

Example

```
AT+CFTRANTX="c:/MyDir/t1.txt"
OK
+CFTRANTX: DATA, 11
```

```
Testcontent

+CFTRANTX: 0

OK
AT+CFTRANTX=”d:/MyDir/t1.txt”,1,4
+CFTRANTX: DATA, 4
estc

+CFTRANTX: 0

OK
```

21. AT Commands for AUDIO

21.1 Overview of AT Commands for AUDIO

Command	Description
AT+CREC	Record wav audio file
AT+CRECAMR	Record amr audio file
AT+CCMXPLAYWAV	Play wav audio file
AT+CCMXSTOPWAV	Stop playing wav audio file
AT+CCMXPLAY	Play audio file
AT+CCMXSTOP	Stop playing audio file

21.2 Detailed Description of AT Commands for AUDIO

21.2.1 AT+CREC Record wav audio file

AT+CREC Record wav audio file	
Read Command **AT+CREC?**	Response **+ CREC: <status>** **OK**
Write Command **AT+CREC=<record_path>,<filename>**	Response **+CREC:1** **OK** or **ERROR**
Write Command **AT+CREC=<mode>**	Response **+CREC:0** **OK** **+RECSTATE: crec stop**
Parameter Saving Mode	-
Maximum Response Time	-

Reference	-

Defined Values

<status>	Indicate whether the recording is going on. 0 – free, not recording 1 – busy, recording
<record_path>	Source of recorded sound 1 – local path 2 – remote path 3 – local and remote sound mixing
<filename>	The location and name of wav file.
<mode>	Stop recording wav audio file 0 –stop

NOTE

- **<filename>**,The file should be put into the “E:/”. Maximum filename length is 240 bytes. (including "")

  <record_path>,Only during the call, **<record_path>** can be set to 2 or 3

Example

```
AT+CREC=1,”e:/rec.wav”
+CREC:1

OK
AT+CREC=0
+CREC:0

OK

+RECSTATE: crec stop
```

21.2.2 AT+CRECAMR Record amr audio file

AT+CRECAMR	Record amr audio file

Read Command **AT+CRECAMR?**	Response **+ CRECAMR: <status>** **OK**
Write Command **AT+CRECAMR=<record_path>,<filename>**	Response **+CRECAMR:<status>** **OK** or **ERROR**
Write Command **AT+CRECAMR=<mode>**	Response **+CRECAMR:<status>** **OK** **+RECSTATE: crecamr stop**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<status>	Indicate whether the recording is going on. 0 – free, not recording 1 – busy, recording
<record_path>	Source of recorded sound 1 – local path 2 – remote path
<filename>	The location and name of amr file.
<mode>	Stop recording wav audio file 0 –stop

NOTE

- **<filename>**,The file should be put into the “E:/”. Maximum filename length is 240 bytes. (including "")

  <record_path>,Only during the call, **<record_path>** can be set to 2

Example

```
AT+CRECAMR=1,”e:/rec.amr”
+CRECAMR:1

OK
AT+CRECAMR=0
+CRECAMR:0
```

OK

+RECSTATE: crecamr stop

21.2.3 AT+CCMXPLAYWAV Play wav audio file

AT+CCMXPLAYWAV Play wav audio file	
Read Command **AT+CCMXPLAYWAV?**	Response **+CCMXPLAYWAV:<play_path>,<repeat>** **OK**
Write Command **AT+CCMXPLAYWAV=<filename>,<play_path>[,<repeat>]**	Response **+WAVSTATE: wav play** **OK** **+WAVSTATE: wav play stop** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<play_path>	Play to local or to remote. 1 – remote 2 – local
<repeat>	How much times can be played. Default 0
<filename>	The location and name of wav file.

NOTE

- **<filename>**,The wav audio file should be located at “E:/”. Maximum filename length is 240 bytes. (including "")

  <play_path>,Only during the call, **<play_path>** can be set to 1 successfully.Only 8k 16bit wav audio can be played to remote successful at present.

  <repeat>,This parameter is reserved, not used at present, you can input this parameter or not. (0--255)

Example

```
AT+CCMXPLAYWAV=?
+CCMXPLAYWAV: (1-2),(0-255)

OK
AT+CCMXPLAYWAV=”E:/rec.wav”,2
+WAVSTATE: wav play

OK

+WAVSTATE: wav play stop
```

21.2.4 AT+CCMXSTOPWAV Stop playing wav audio file

AT+CCMXSTOPWAV Stop playing wav audio file	
Read Command AT+CCMXSTOPWAV=?	Response OK
Write Command AT+CCMXSTOPWAV	Response +CCMXSTOPWAV: OK +WAVSTATE: wav play stop
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

-	-

Example

```
AT+CCMXSTOPWAV
+CCMXSTOPWAV:

OK

+WAayVSTATE: wav pl stop
```

21.2.5 AT+CCMXPLAY Play audio file

AT+CCMXPLAY Play audio file	
Read Command **AT+CCMXPLAY?**	Response **+CCMXPLAY:<play_path>,<repeat>** **OK**
Write Command **AT+CCMXPLAY=<filename>[,<play_path>][,<repeat>]**	Response **+CCMXPLAY:** **OK** **+AUDIOSTATE: audio play** **+AUDIOSTATE: audio play stop** or **ERROR** or **+CCMXPLAY:** **OK** **+AUDIOSTATE: audio play** **+AUDIOSTATE: audio play error**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<play_path>	Play to local or to remote. Default 0 0 – local 1 – remote
<repeat>	How much times can be played. Default 0
<filename>	The location and name of wav file.

NOTE

<filename>,The wav audio file should be located at “E:/”. Maximum filename length is 240 bytes. (including ""). Support audio file format mp3, aac, amr, wav.

<play_path>,Only during the call, **<play_path>** can be set to 1 successfully.Only 8k 16bit wav audio and amr audio can be played to remote at present.

<repeat>,This parameter is reserved, not used at present, you can input this parameter or not. (0--255)

Example

```
AT+CCMXPLAY=?
+CCMXPLAY: (0-1),(0-255)

OK
AT+CCMXPLAY=”E:/rec.mp3”,0,0
+CCMXPLAY:

OK

+AUDIOSTATE: audio play

+AUDIOSTATE: audio play stop
```

21.2.6 AT+CCMXSTOP Stop playing audio file

AT+CCMXSTOP Stop playing audio file	
Read Command AT+CCMXSTOP=?	Response OK
Write Command AT+CCMXSTOP	Response +CCMXSTOP: OK +AUDIOSTATE: audio play stop
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

-	-

Example

```
AT+CCMXSTOP
+CCMXSTOP:

OK

+AUDIOSTATE: audio play stop
```

22. AT Commands for TTS

22.1 Overview of AT Commands for TTS

Command	Description
AT+CDTAM	TTS play path, local or remote
AT+CTTS	TTS operation, play or stop
AT+CTTSPARAM	TTS parameters, set or get

22.2 Detailed Description of AT Commands for TTS

22.2.1 AT+CDTAM TTS play path, local or remote

AT+CDTAM TTS play path, local or remote	
Test Command **AT+CDTAM=?**	Response **+CDTAM: (0-1)** **OK**
Read Command **AT+CDTAM?**	Response **+ CDTAM: <status>** **OK**
Write Command **AT+CDTAM=<mode>**	Response **+CDTAM:** **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<status>	Indicate play path, play TTS to local or play to remote. 0 – Local path 1 – Remote path
<mode>	Set TTS play path, local or remote. Default value is 0. 0 – Local path 1 – Remote path

Example

```
AT+CDTAM=1
+CDTAM:

OK
```

22.2.2 AT+CTTS TTS operation, play or stop

AT+CTTS TTS operation, play or stop	
Test Command **AT+CTTS=?**	Response **OK**
Read Command **AT+CTTS?**	Response **+CTTS: <status>** **OK**
Write Command **AT+CTTS=<mode>[,<text>]**	Response If **<mode>**is 0， then **<text>** is not required. When TTS is playing, return： **+CTTS:0** **OK** If **<mode>**is 0，then **<text>** is not required. When TTS is not playing, return： **OK** If **<mode>**is 1 or 2，then **<text>** is must be required. return: **OK** **+CTTS:0** or **ERROR**
Write Command	Response

AT+CTTS=<mode>[,<text>][,<filename>]	If **<mode>**is 3 or 4，then **<text>** and **<filename>** are must be required. return： **OK** **+CTTS:0** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<status>	Indicate playing thread status. Default value is 0 0 – NO_WORKING 1 – PLAY_WAV_WORKING 2 – AMR_WORKING 3 – MP3_WORKING 4 – AAC_WORKING 5 – WAV_WORKING 6 – TTS_WORKING 8 – CREC_WORKING
<mode>	Stop or play TTS. 0 – Stop TTS 1 – **<text>** is in UCS2 coding format, Start to synth and play 2 –**<text>** is in ASCII coding format for English,Chinese text is in GBK coding format. Start to synth and play 3 – **<text>** is in ASCII coding format for English,Chinese text is in GBK coding format. Start to synth and play, and save pcm data as wav file. 4 – **<text>** is in UCSII coding format . Start to synth and play, and save pcm data as wav file.
<filename>	Location and filename for wav file

NOTE

- **<text>**, which is synthetized to speed to be played, maximum data length is 512 bytes. (including "")

  <filename>,The file should be put into the “E:/filename.wav”. Maximum filename length is 240 bytes. (including "")

Example

```
AT+CTTS=1,"6B228FCE4F7F75288BED97F3540862107CFB7EDF"
```

OK **+CTTS:0** **AT+CTTS=3,"欢迎使用语音合成系统","E:/tts.wav"** **OK** **+CTTS:0** **AT+CTTS=0** **OK** **+CTTS:0**	

22.2.3 AT+CTTSPARAM TTS Parameters, set or get

AT+CTTSPARAM TTS Parameters, set or get	
Test Command **AT+CTTSPARAM=?**	Response **+CTTSPARAM: (0-2), (0-3),(0-3),(0-2),(0-2)** **OK**
Read Command **AT+CTTSPARAM?**	Response **+CTTS: <volume>,<sysvolume>,<digitmode>,<pitch>,<speed>** **OK**
Write Command **AT+CTTSPARAM=<volume>[,<sysvolume>[,<digitmode>[,<pitch>[,<speed>]]]]**	Response **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	-

Defined Values

<volume>	TTS Speech Volume, default：2. 0 – The mix volume 1 – The normal volume 2 – The max volume
<sysvolume>	The module system volume, default：3. 0 – The mix system volume

	1 – The small system volume 2 – The normal system volume 3 – The max system volume
<digitmode>	The digit read mode, default：0 0 – Auto read digit based on number rule first. 1 – Auto read digit bases on telegram rule first. 2 – Read digit based on telegram rule. 3 – Read digit based on number rule.
<pitch>	The voice tone, default：1 0 – The mix voice tone. 1 – The normal voice tone. 2 – The max voice tone.
<speed>	The voice speed, default：1 0 – The mix speed 1 – The normal speed 2 – The max speed

NOTE

- **<sysvolume>**, It takes no effect to set **<sysvolume>**,reserved at present

Example

```
AT+CTTSPARAM=1,3,0,1,1

OK
```

23. AT Commands for FOTA

23.1 Overview of AT Commands for FOTA

Command	Description
AT+CAPFOTA	Start/Close FOTA Service
AT+CSCFOTA	Configure parameters and download upgrade package

23.2 Detailed Description of AT Commands for FOTA

23.2.1 AT+CAPFOTA Start/Close FOTA Service

AT+CAPFOTA Start/Close FOTA Service	
Test Command **AT+CAPFOTA=?**	Response **+CAPFOTA:** (list of supported <**on/off**>s) **OK**
Read Command **AT+CAPFOTA?**	Response **+CAPFOTA:** 1 OK
Write Command **AT+CAPFOTA=<on/off>**	Response **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<on/off>	The service status on/off, the default value is 0. 0 Close FOTA program

	1 Active FOTA program The function will take effect immediately.

Example

```
AT+CAPFOTA?
+CAPFOTA: 1

OK
```

23.2.2 AT+CSCFOTA Configure parameters and download upgrade package

AT+CSCFOTA Configure parameters and download upgrade package	
Write Command **AT+CSCFOTA=<OEM>,<models>,<productID>,<productSecret>,<target version>**	Response If successfully: **OK** **+CSCFOTA: <err>** b)If failed: **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<OEM>	The name of project design company. This name must be the same as the OEM created on the cloud platform. Otherwise, it will cause upgrade failed.
<models>	The name of the device model. This name must be the same as the device model created on the cloud platform. Otherwise, it will cause upgrade failed.
<productID>	The product ID that must be the same as the product ID generated on the cloud platform.
<productSecret>	The product secret is used to confirm the identity and usage rights of the user. It must be the same as the product secret generated on the cloud platform.
<target version>	The version that needs to be upgraded to. This version is published by the cloud platform.

<ERR>	
1	unknown error
2	Check version is finished

3	Download is finished
4	Download partial finished
5	No matched version
301	No enough memory
302	Invalid parameter
303	Invalid operation
304	IO failed
305	IO timeout
306	Download file verification failed
307	got canceled
308	Interface nesting error
401	Invalid device information
402	Invalid platform information
403	Missing device information
404	Version number is not configured
405	Internal error (contact supplier)
501	Invalid URL
502	Unable to resolve domain name
503	cannot connect to the server
504	Invalid request, server returned error
505	Not in range
506	HTTP POST request error
507	Re-download start error
508	Operation is aborted
509	Operation not completed
510	Too many retargeting times
511	Unable to get data from SOCKET
512	Error sending data via SOCKET
513	Error receiving data via SOCKET
514	Invalid SOCKET connection

Example

```
AT+CSCFOTA="SIMCOM","7600M21","15409
07004","f9bbb0d76f894da090b6b6925361656
1","SIM7600M21_LE11_181025_V2.00"


OK
+CSCFOTA: 2
+CSCFOTA: 3
```

24. AT Commands for UIM hotswap

24.1 Overview of AT Commands for UIM hotswap

Command	Description
AT+UIMHOTSWAPON	Set UIM hotswap function on
AT+UIMHOTSWAPLEVEL	Set UIM card detection level

24.2 Detailed Description of AT Commands for UIM hotswap

24.2.1 AT+UIMHOTSWAPON Set UIM hotswap function on

AT+UIMHOTSWAPON Set UIM hotswap function on	
Read Command **AT+UIMHOTSWAPON?**	Response **+UIMHOTSWAPON: <onoff>** **OK**
Write Command **AT+UIMHOTSWAPON=<onoff>**	Response **OK** or **ERROR**

Defined Values

<onoff>	0 The UIM hotswap function is disabled 1 The UIM hotswap function is enabled

Example

```
AT+UIMHOTSWAPON?
+UIMHOTSWAPON: 0

OK
```

```
AT+UIMHOTSWAPON=1
OK
```

NOTE

- Module reset to take effect

24.2.2 AT+UIMHOTSWAPLEVEL Set UIM card detection level

AT+UIMHOTSWAPLEVEL Set UIM card detection level	
Read Command **AT+UIMHOTSWAPLEVEL?**	Response **+UIMHOTSWAPLEVEL: <level>** **OK**
Write Command **AT+UIMHOTSWAPLEVEL=<level>**	Response **OK** or **ERROR**

Defined Values

<level>	0 ACTIVE LOW 1 ACTIVE HIGH

Example

```
AT+UIMHOTSWAPLEVEL?
+UIMHOTSWAPLEVEL: 0

OK
AT+UIMHOTSWAPLEVEL=1
OK
```

NOTE

- Module reset to take effect
- Set UIM card detection level to active low. //Refer to the used SIM card holder, usually it's a "normal open kind" one.
- The default value 1

25. AT Commands for HSIC_LAN

25.1 Overview of AT Commands for HAIC_LAN

Command	Description
AT+CENABLELAN	Enable LAN function
AT+CLANMODE	Set LAN mode
AT+CLANCTRL	Set LAN configure
AT+CHSICSLEEP	Allow Hsic Device Go to AutoSleep

25.2 Detailed Description of AT Commands for HSIC_LAN

25.2.1 AT+CENABLELAN Enable LAN function

AT+CENABLELAN Enable LAN function	
Write Command **AT+CENABLELAN=<onoff>**	Response **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<onoff>	0 Close the LAN9730 1 Open the LAN9730

Example

```
AT+CENABLELAN=1
OK
```

NOTE

- LAN9730 is not opened in default, if want to open the LAN9730, you can run AT+CENABLELAN=1. After run this command, the module will restart automatically, then the LAN9730 will be opened.
- If want to close the LAN9730, you can run AT+CENABLELAN=0. After run this command, the module will restart automatically, then the LAN9730 will be closed.
- WIFI firmware doesn't care this AT command.

25.2.2 AT+CLANMODE Set LAN mode

AT+CLANMODE Set LAN mode	
Test Command **AT+CLANMODE=?**	Response **+CLANMODE:** (list of supported <**mode**>s) OK
Read Command **AT+CLANMODE?**	Response **+CLANMODE:** <**mode**> OK
Write Command **AT+CLANMODE=<mode>**	Response **OK** or **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<mode>	0 lan mode 1 wan mode 2 static ip mode

Example

```
AT+CLANMODE?
+CLANMODE: (0,1)

OK
AT+CLANMODE=1
```

OK

NOTE

- Module works in lan mode in default. If want to use another mode, need to run at+clanmode=<mode>, after run this command, module will restart automatically, then the module will work in target mode.

25.2.3 AT+CLANCTRL Set LAN configure

AT+CLANCTRL Set LAN configure	
Test Command **AT+CLANCTRL=?**	Response **+CLANCTRL:** (list of supported <**option**>s) OK
Write Command **AT+CLANCTRL=<option>,[[type/ip],[netmask]]**	Response **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<option>	0 uninstall driver 1 install driver 2 set mac address 3 set ip address 4 bring up eth0 5 bring down eth0
<type>	1 bcm898xx 2 at803x
<ip>	LAN ip address (Range: 192.168.*.*).
<netmask >	Range: 255.255.*.* if the parameter is not set, will use the default value:255.255.255.0

Example

AT+CLANCTRL=3,"192.168.1.1" **OK** **AT+CLCANCTRL=3,"192.168.1.1","255.255.255.0"** **OK**	**Set ip. The netmask is use default value 255.255.255.0** **Set ip and netmask. The netmask is 255.255.255.0**

NOTE

- Uninstall driver (option=0). Not support for HSIC LAN, Only Support SGMII LAN. Please don't run this command on HSIC LAN module
- Install driver (option=1). Not support for HSIC LAN, Only Support SGMII LAN.
- Set mac address (option=2). Support for HSIC LAN, But the module will auto set the Mac address. So there is no need to run the command.
- Set ip address (option=3). When module work in static ip mode. Use this command set ip and netmask.
- Bring up eth0 bring up eth0 (option=4). equal to "ifconfig eth0 up"
- Bring down eth0 (option=5). equal to "ifconfig eth0 down"

25.2.4 AT+CHSICSLEEP Allow HSIC Device Go to AutoSleep

AT+CHSICSLEEP Allow HSIC Device Go to AutoSleep	
Test Command **AT+CHSICSLEEP=?**	Response **+CHSICSLEEP: (**list of supported **<state>**s**)** OK
Read Command **AT+CHSICSLEEP?**	Response **+CHSICSLEEP: <**state**>** OK
Write Command **AT+CHSICSLEEP=<state>**	Response **OK** or **ERROR**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<state>	0 Don't allow the hsic device go to autosleep 1 Allow the hsic device go to autosleep

Example

```
AT+CHSICSLEEP =1
OK

AT+CHSICSLEEP?
+CHSICSLEEP: 1

OK
```

NOTE

- If the module needs to go to sleep, user needs to execute following steps:
  1. AT+CHSICSLEEP=1
  2. AT+CLANCTRL=5

26. AT Commands for Ecall

26.1 Overview of AT Commands for Ecall

Command	Description
AT+CECALLS	Make e-call
AT+CECALLE	Hang up e-call
AT+CECALLCFG	Configure e-call MSD information
AT+CECALLPOS	Set position information
AT+CECALLTIME	Set timestamp
AT+CMSDVERSION	Set MSD serialize version
AT+CECALLTOUT	Set T5,T6,T7 timeout value
AT+CMSDMESSAGEID	Set the initiatory message identifier of msd data Description
AT+CMSDOIDDATA	Set the optional additional data
AT+CMSD	Input hex Minimum set of data(MSD)
AT+CMSDCONTROL	Set the control data in Minimum set of data

26.2 Detailed Description of AT Commands for Ecall

26.2.1 AT+CECALLS Make e-call

The command is used to make an e-call.

AT+CECALLS Make an e-call	
Test Command **AT+CECALLS=?**	Response **+CECALLS: (scope of <cannedMSD>)** **OK**
Write Command **AT+CECALLS=<num>,<cannedMSD>**	Response **OK**

	ERROR

Defined Values

<num>	Dialing number.
<cannedMSD>	Use the canned GPS information or real GPS information. 0 — Send real MSD 1 — Send canned MSD

Example

```
AT+CECALLS=15865451120,1
OK
```

26.2.2 AT+CECALLE Hang up e-call

The command is used to hang up the e-call.

AT+CECALLE Hang up an e-call	
Test Command **AT+CECALLE=?**	Response **+CECALLS: (0-1)** **OK**
Read Command **AT+CECALLE?**	Response **+CECALLE: <n>** **OK**
Write Command **AT+CECALLE=<n>**	Response **OK** **VOICE CALL: END: <time>** No call: **OK**

Defined Values

<n>	0 – Stop an active eCall, change the state into "ECALL_APP_ECALL_INACTIVE" and clear callbackTimer. When set to 0, module cannot receive a MT ECALL from PSAP. 1 – End an active ecall, but keep state "ECALL_APP_IDLE_ALLOW_MT_ECALL", not clear callbackTimer. When set to 1, module can receive a MT ECALL from PSAP.

<time>	Voice call connection time. Format – HHMMSS (HH: hour, MM: minute, SS: second)

Example

```
AT+CECALLE=0
OK
```

26.2.3 AT+CECALLCFG Configure e-call MSD information

The command is used to configure the MSD information.

AT+CECALLCFG Configure e-call MSD information	
Test Command **AT+CECALLCFG=?**	Response **OK**
Write Command **AT+CECALLCFG=<vehicletype>,<storage>,<num>,<vin>,<vehicledirection>,<delta1_lon>,<delta1_lat>,<delta2_lon>,<delta2_lat>**	Response **OK** **ERROR**

Defined Values

<vehicletype>	1 — Passenger vehicle class M1 2 — Buses and coaches class M2 3 — Buses and coaches class M3 4 — Light commercial vehicles class N1 5 — Heavy duty vehicles class N2 6 — Heavy duty vehicles class N3 7 — Motorcycles class L1e 8 — Motorcycles class L2e 9 — Motorcycles class L3e 10 — Motorcycles class L4e 11 — Motorcycles class L5e 12 — Motorcycles class L6e 13 — Motorcycles class L7e
<storage>	Propulsion storage: It should choice multi-storage. decimal number **NOTE** Example: Choice “Electric energy storage” and “Diesel tank present”, the <storage> must be set by 18. （i.e. 2 or 16 equal 18） 0 — Unknown or other type of energy storage 1 — Hydrogen storage

	2 — Electric energy storage 4 — Liquid propane gas 8 — Compressed natural gas 16 — Diesel tank present 32 — Gas online tank present Range is 0~63.
<num>	Number of passenger. Range is 0~255.
<vin>	Vehicle id number. Length of <vin> must be 17. VIN number according to ISO 3779. including: 1.World Manufacturer Index (WMI) 2.Vehicle Type Descriptor (VDS) 3.Vehicle Identification Sequence (VIS) The character in VIN must be the member of this table: ("A".."H"|"J".."N"|"P"|"R".."Z"|"0".."9")
<vehicledirection>	The direction of travel in 2°-degrees steps from magnetic north (0– 358, clockwise). Only values from 0 to 179 are valid. If direction of travel is invalid or unknown, the value 0xFF shall be used. Unit is 2 degree. Range of <vehicledirection> is 0~179.
<delta1_lon>	Description of recent vehicle longitude location before the incident. 1 Unit = 100 miliarcseconds, which is approximately 3m. Coded value range (-512..511) representing -51200 to +51100 miliarcseconds, or from 51,2''S to 51,1''N from the reference position.
<delta1_lat>	Description of recent vehicle latitude location before the incident. 1 Unit = 100 miliarcseconds, which is approximately 3m. Coded value range (-512..511) representing -51200 to +51100 miliarcseconds, or from 51,2''S to 51,1''N from the reference position.
<delta2_lon>	Description of recent vehicle latitude location before the incident. 1 Unit = 100 miliarcseconds, which is approximately 3m.
<delta2_lat>	Description of recent vehicle latitude location before the incident. 1 Unit = 100 miliarcseconds, which is approximately 3m. Coded value range (-512..511) representing -51200 to +51100 miliarcseconds, or from 51,2''S to 51,1''N from the reference position.

Example

```
AT+CECALLCFG=5,18,8,"WMJVDSVDSYA123456",14,10,-10,20,-20
OK
```

26.2.4 AT+CECALLPOS Set position information

The command is used to set position information.

AT+CECALLPOS Set position information	
Test Command **AT+CECALLPOS=?**	Response **OK**
Write Command **AT+CECALLPOS=<lon>,<lat>**	Response **OK** **ERROR**

Defined Values

<lon>	Longitude of current position, format is ddd.dddddd. Unit is degree. Range is -180~180.
<lat>	Latitude of current position, format is dd.dddddd. Unit is degree. Range is -90~90.

Example

```
AT+CECALLPOS="121.354138","31.221938"
OK
```

26.2.5 AT+CECALLTIME Set timestamp

The command is used to set timestamp.

AT+CECALLTIME Set timestamp	
Test Command **AT+CECALLPOS=?**	Response **OK**
Write Command **AT+CECALLTIME=<flag>[,<year>,<month>,<day>,<hour>,<minute>,<second>]**	Response **OK** **ERROR**

Defined Values

<flag>	0 - use system time, not need to set <year>, <month>, <day>, <hour>, <minute>,<second> 1 - must set <year>,<month>,<day>,<hour>,<minute>,<second>
<year>	Year :integer Range is 1970~2100
<month>	Month : integer Range is 1~12

<day>	Day : integer Input range : Jan \ Mar \ May \ Jul \ Aug \ Oct \ Dec: 1~31 Feb: 1~28 (1~29 if leap year) Apr \ Jun \ Sep \ Nov: 1~30
<hour>	Hour : integer Range is 0~23
<minute>	Minute : integer Range is 0~59
<second>	Second : integer Rang is 0~59

Example

```
AT+CECALLTIME=1,2011,10,20,15,30,30
OK
```

26.2.6 AT+CECALLVERSION Set MSD serialize version

The command is used to set MSD pack format.

AT+CECALLVERSION Set MSD serialize version	
Test Command **AT+CECALLVERSION=?**	Response **+CMSDVERSION:（1-2）** **OK**
Read Command **AT+CECALLVERSION?**	Response **+CMSDVERSION: <ver>** **OK**
Write Command **AT+CMSDVERSION=<ver>**	Response **OK** **ERROR**

Defined Values

<ver>	1 - set MSD serialize version 1 (qualcomm default version,other European country) 2 - set MSD serialize version 2 (just for Russia ecall)

Example

```
AT+CMSDVERSION=1
OK
```

26.2.7 AT+CECALLTOUT Set T5,T6,T7 timeout value

The command is used to set T5,T6,T7 timeout value.

AT+CECALLTOUT Set T5,T6,T7 timeout value	
Read Command **AT+CECALLVERSION?**	Response **+CECALLTOUT: T5=<timeoutvalue>, T6=<timeoutvalue>, T7=<timeoutvalue>** **OK**
Write Command **AT+CECALLTOUT=<TX>,<timeoutvalue>**	Response **OK** **ERROR**

Defined Values

<TX>	T5 - The timer of IVS waiting for START, default timeout value is 2 seconds. The timeout value will not be saved to NV. You should set the timeout value before organizing the eCall. For further information about this timer, please refer to EN 16062. Range is 2000-255000 ms. Default value 2000 ms T6 - The timer of IVS waiting for HACK, default timeout value is 5 seconds. The timeout value will not be saved to NV. You should set the timeout value before organizing the eCall. For further information about this timer, please refer to EN 16062. Range is 5000-255000 ms. Default value 5000 ms. T7 - The timer for MSD transmission, default timeout value is 20 seconds. The timeout value will not be saved to NV. You should set the timeout value before organizing the eCall. For further information about this timer, please refer to EN 16062. Range is 20000-255000 ms. Default value 20000 ms

Example

```
AT+CECALLTOUT="T5",4000
```

```
OK
```

26.2.8 AT+CMSDMESSAGEID Set the initiatory message identifier of msd data Description

The command is used to set the initiatory message identifier of msd data.

AT+CMSDMESSAGEID Set the initiatory message identifier of msd data Description	
Test Command **AT+CMSDMESSAGEID=?**	Response **+CMSDMESSAGEID: (list of supported <messageid>)** **OK**
Read Command **AT+CMSDMESSAGEID?**	Response **+CMSDMESSAGEID: <messageid>** **OK**
Write Command **AT+CMSDMESSAGEID=<messageid>**	Response **OK** **ERROR**

Defined Values

<messageid>	starting with 1 for each new eCall session and to be incremented with every application layer MSD retransmission following a new 'Send MSD' request after the incident event .(1-255)

Example

```
AT+CMSDMESSAGEID=1
OK
```

26.2.9 AT+CMSDOIDDATA Set the optional additional data

The command is used to set the optional additional data.

AT+CMSDOIDDATA Set the optional additional data

Test Command **AT+CMSDOIDDATA=?**	Response **OK**
Write Command **AT+CMSDOIDDATA=<oid>,<odata>**	Response **OK** **ERROR**

Defined Values

<oid>	Object identifier which uniquely identifies the format and meaning of the data which follows. (oid is decimal string x.x.xxx), the length must be 7.
<odata>	Transparent optional additional data. (odata is hex string) which maximum size is 100 bytes.

Example

```
AT+CMSDOIDDATA="1.2.125","30304646"
OK
```

26.2.10 AT+CMSD Input hex Minimum set of data

The command is used to input hex Minimum set of data.

AT+CMSD Input hex Minimum set of data	
Test Command **AT+CMSD=?**	Response **OK**
Write Command **AT+CMSDOIDDATA=<MSD>,<activationType>,<eCallType>**	Response **OK** **ERROR**

Defined Values

<msd>	the hex msd data generated by user which maximum size is 140 bytes.
<activation>	0 - Manual activation 1 - Automatic activation
<eCallType>	0 - Emergency call 1 - Test call

Example

```
AT+CMSD="015C068150820442001426400042OD101404E80DA4C89A3B2F09905B6440E829F682
9EC020301027D04303046460",0,1
OK
```

26.2.11 AT+CMSDCONTROL Set the control data in Minimum set of data

The command is used to set the control data in Minimun set of data(MSD).

AT+CMSDCONTROL Set the control data in Minimum set of data	
Test Command **AT+CMSDCONTROL=?**	Response **OK**
Write Command **AT+CMSDCONTROL=<activationType>,<callType>,<positionCanBeTrusted>**	Response **OK** **ERROR**

Defined Values

<activationType>	Manual activation(by pushing the emergency button) or automatic activation(by hitting sensors). 0 — Manual activation 1 — Automatic activation
<callType>	e-call type: 0 — Test call 1 — Emergency call
<positionCanBeTrusted>	0 — low confidence in position 1 — Position can be trusted

Example

```
AT+CMSDCONTROL=0,0,1
OK
```

27. AT Commands for MIFI

27.1 Overview of AT Commands for MIFI W58

Command	Description
AT+CWMAP	Open/Close WIFI
AT+CWSSID	SSID setting
AT+CWBCAST	Broadcast setting
AT+CWAUTH	Authentication type, encrypt mode and password setting
AT+CWMOCH	80211 mode and channel setting
AT+CWISO	Client isolation setting
AT+CWDHCP	Get the current DHCP configuration
AT+CWNAT	NAT type setting
AT+CWCLICNT	Get client number connected to the WIFI
AT+CWRSTD	Restore to default setting
AT+CWMAPCFG	WIFI configuration setting
AT+CWLANSRV	LAN SERVER setting
AT+CWLANMSG	Send message
AT+CWMACADDR	Get MAC address
AT+CWNETCNCT	Query the connection to the network
AT+CWSTAIP	Get STA mode IP address
AT+CWSTASCAN	Scan WIFI network
AT+CWSTACFG	STA mode configuration setting
AT+CWUSRINFO	Auth info of wifi data call setting

27.2 Overview of AT Commands for MIFI W58L(RTL)

Command	Description
AT+CWMAP	Open/Close WIFI
AT+CWSSID	SSID setting
AT+CWBCAST	Broadcast setting
AT+CWAUTH	Authentication type, encrypt mode and password setting

AT+CWMOCH	80211 mode and channel setting
AT+CWDHCP	Get the current DHCP configuration
AT+CWCLICNT	Get client number connected to the WIFI
AT+CWRSTD	Restore to default setting
AT+CWLANSRV	LAN SERVER setting
AT+CWLANMSG	Send message
AT+CWMACADDR	Get MAC address
AT+CWNETCNCT	Query the connection to the network
AT+CWSTAIP	Get STA mode IP address
AT+CWSTASCAN	Scan WIFI network
AT+CWSTACFG	STA mode configuration setting
AT+CWSTAINIT	STA mode setting
AT+CWUSRINFO	Auth info of wifi data call setting

27.3 Detailed Description of AT Commands for MIFI

27.3.1 AT+CWMAP Open/Close WIFI

AT+CWMAP Open/Close WIFI	
Test Command **AT+CWMAP=?**	Response **+CWMAP: (0-1)** **OK**
Read Command **AT+CWMAP?**	Response **+CWMAP: <flag>** **OK**
Write Command **AT+CWMAP=<flag>**	Response **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<flag>	0	Close
	1	Open

Example

```
AT+CWMAP?
+CWMAP: 1

OK
AT+CWMAP=0
OK
```

27.3.2 AT+CWSSID SSID setting

AT+CWSSID SSID setting	
Read Command **AT+CWSSID?**	Response **+CWSSID: <ssid>** **OK**
Write Command **AT+CWSSID=<ssid>**	Response **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<ssid>	new ssid string 1. The max length of <ssid> is 32 bytes when the <ssid> include only ASCII characters. 2. The max length of <ssid> is 20 bytes when <ssid> include only Chinese (One Chinese characters is 2 bytes, so the max Chinese count is 10). 3. The max length of <ssid> is 22 bytes when <ssid> include ASCII and Chinese characters (One Chinese character is 2 bytes, one ASCII character is 1 byte). The default value is SIM7600MIFI.

Example

```
AT+CWSSID?
+CWSSID: “SIM7600MIFI”

OK
```

27.3.3 AT+CWBCAST Broadcast setting

AT+CWBCAST Broadcast setting	
Test Command **AT+CWBCAST=?**	Response **+CWBCAST: (0-1)** **OK**
Read Command **AT+CWBCAST?**	Response **+CWBCAST: <broadcast>** **OK**
Write Command **AT+CWBCAST=<broadcast>**	Response **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<broadcast>	0 disabled 1 enabled

Example

```
AT+CWBCAST?
+CWBCAST: 1

OK
AT+CWBCAST=0
OK
```

27.3.4 AT+CWAUTH Authentication setting

AT+CWAUTH Authentication type, encrypt mode and password setting	
Read Command **AT+CWAUTH?**	Response **+CWAUTH:<auth>,<encrypt>[,<password>]**

	OK
Write Command **AT+CWAUTH=<auth>,<encrypt> [,<password>]**	Response **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<auth>	0 open/share 1 open 2 share 3 wpa 4 wpa2 5 wpa/wpa2
<encrypt>	0 null 1 WEP 2 TKIP 3 AES 4 TKIP-AES
<password>	password string, the length is 5 or betwwen 8 to 64. The char in the password is only allow the ASCII ‘s decimal code betwwen 32 to 126.

NOTE

The parameter need to meet the following conditions:

1. If (auth = 0 or auth = 1) then (encrypt = 0 or encrypt = 1)
2. If (auth =2) then (encrypt = 1)
3. If (auth >=3) then (encrypt >=2)
4. If(encrypt = 0) then (password is null)
5. If(encrypt = 1) then
   {
   1) password can’t be set null
   2) password format: (5 ASCII character) or (10 hexadecimal number) or(13 ASCII character) or(26 hexadecimal number)
   }
6. if(encrypt >= 2) then
   {
   1) password can’t be set null
   2) password format: (8~63 ASCII character or 64 hexadecimal number)
   }

Example

```
AT+CWAUTH?
+CWAUTH: 0,1, "11111"

OK
AT+CWAUTH?
+CWAUTH: 5,4, "12345678"

OK
AT+CWAUTH=0,0                                   //Auth:open/share  encrypt:null
OK
AT+CWAUTH=0,1,”11111”                           //Auth:open/share  encrypt:WEP
OK
AT+CWAUTH=2,1,”12345”                           //Auth:share  encrypt:WEP
OK                                              (ASCII character  password:12345)
AT+CWAUTH=2,1,”3132333435”                      //Auth:share  encrypt :WEP
OK                                              (sixteen hexadecimal number:password  12345)
AT+CWAUTH=5,4,”abcd1234”                        //Auth:WPA/WPA2  encrypt:TIKP-AES
OK
```

27.3.5 AT+CWMOCH 80211 mode and channel setting

AT+CWMOCH 80211 mode and channel setting	
Read Command AT+CWMOCH?	Response +CWMOCH: <mode>,<channel> OK
Write Command AT+CWMOCH=<mode>,<channel>	Response OK
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<mode>	2	b	2.4G mode
	3	b/g	2.4G mode
	4	b/g/n	2.4G mode

<channel>	0	auto select
	1~11	2.4Gmode channel number

Example

```
AT+CWMOCH?
+ CWMOCH: 4,0

OK
AT+CWMOCH=3,1
OK
```

27.3.6 AT+CWISO Client isolation setting

AT+CWISO Client isolation setting	
Test Command **AT+CWISO=?**	Response **+CWISO: (0-1)** **OK**
Read Command **AT+CWISO?**	Response **+CWISO: <isolation>** **OK**
Write Command **AT+CWISO=<isolation>**	Response **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<isolation>	0	close
	1	open

Example

```
AT+CWISO?
+CWISO: 1

OK
AT+CWISO=0
```

OK

27.3.7 AT+CWDHCP Get the current DHCP configuration

AT+CWDHCP Get the current DHCP configuration	
Read Command **AT+CWDHCP?**	Response **+CWDHCP:<host_ip>,<range_start_ip>,<range_end_ip>,<leasetime>** **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<host_ip>	the AP IP
<range_start_ip>	the start IP of the IP range that assigned to the client
<range_end_ip>	the end IP of the IP range that assigned to the client
<leasetime>	the lease time

Example

```
AT+CWDHCP?
+CWDHCP: "192.168.1.250","192.168.1.128","192.168.1.249",240h

OK
```

27.3.8 AT+CWNAT NAT type setting

AT+CWNAT NAT type setting	
Test Command **AT+CWNAT=?**	Response **+CWNAT: (0-1)** **OK**
Read Command **AT+CWNAT?**	Response **+CWNAT: <type>** **OK**

Write Command **AT+CWNAT=<type>**	Response **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<type>	0 Symmetric 1 Cone

Example

```
AT+CWNAT?
+CWNAT: 1

OK
AT+CWNAT=0
OK
```

27.3.9 AT+CWCLICNT Get client number connected to the WIFI

AT+CWCLICNT Get the client number connected to the WIFI	
Read Command **AT+CWCLICNT?**	Response **+CWCLICNT: <cnt>** **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<cnt>	the connected client count, range is from 0 to 31.

Example

```
AT+CWCLICNT?
+CWCLICNT: 1

OK
```

27.3.10 AT+CWRSTD Restore to default setting

AT+CWRSTD Restore all MIFI setting to default	
Write Command **AT+CWRSTD**	Response **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Example

```
AT+CWRSTD
OK
```

27.3.11 AT+CWMAPCFG WIFI configuration setting

AT+CWMAPCFG WIFI mode, configuration AP ID setting	
Test Command **AT+CWMAPCFG=?**	Response **+CWMAPCFG: ("enablessid2","configselect"),(0-2)** **OK**
Read Command **AT+CWMAPCFG?**	Response **+CWMAPCFG: <enablessid2_value>,<configselect_value>** **OK**
Write Command **AT+CWMAPCFG=<option>,<value>**	Response **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<enablessid2_value>	0 AP mode 1 AP-AP mode 2 STA-AP mode
<configselect_value>	Current AP ID (0 or 1 or 2)

<option>	”enablessid2” set WIFI mode ”configselect” set the current AP ID
<value>	the value of the options.

NOTE

If (option=”enablessid2”)
0 AP mode
1 AP-AP mode
2 STA-AP mode
If (option=”configselect”)
Current AP ID (0 or 1 or 2) to be set.
When current AP ID is 0, the AT+CWSSID/AT+CWBCAST/AT+CWAUTH/AT+CWMOCH/AT+CWISO/AT+CWDHCP/AT+CWCLICNT/AT+CWMACADDR will modify the first AP’s settings;
When current AP ID is 1, the AT+CWSSID/AT+CWBCAST/AT+CWAUTH/AT+CWMOCH/AT+CWISO/ AT+CWDHCP/AT+CWCLICNT/AT+CWMACADDR will modify the second AP’s settings;
When current AP ID is 2, the AT+CWSSID/AT+CWBCAST/AT+CWAUTH/AT+CWMOCH/AT+CWISO/ AT+CWDHCP/AT+CWCLICNT/AT+CWMACADDR will modify the third AP’s settings, the AT+CWSTAIP/AT+CWSTASCAN/AT+CWSTACFG will modify the STA’s settings.

NOTE

1. It can’t set the configselect value to 1 when enablessid2 is 0.
2. The configselect value will be changed due to enablessid2.

enablessid2	configselect
0	0
1	0 or 1
2	2

Example

```
AT+CWMAPCFG=?
+CWMAPCFG: ("enablessid2","configselect"),(0-2)

OK
AT+CWMAPCFG?
+CWMAPCFG: 0,0

OK
AT+CWMAPCFG=”enablessid2”,1                    // Set enablessid2
OK
AT+CWMAPCFG=”configselect”,0                   // Set configselect
```

OK

27.3.12 AT+CWLANSRV LAN SERVER setting

AT+CWLANSRV LAN server setting	
Read Command **AT+CWLANSRV?**	Response **+CWLANSRV: <server_ip >,<server_port>,<recv_mode>** **OK**
Write Command **AT+CWLANSRV=<value>**	Response **OK**
Write Command **AT+CWLANSRV=0,<server_port>[,<recv_mode>]**	Response **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<server_ip>	Default 192.168.225.1	
<server_port>	Default 5555 The range of permitted values is 1024 to 65535.	
<recv_mode>	0	Report messages directly with URC(+CWLANMSG)
	1	Report cached bytes when new messages are received (+CWLANMSG: <cached_len>).And use AT+CWLANMGET to get cached bytes.
<value>	0	close the server
	1	open the server

Example

```
AT+CWLANSRV?
+CWLANSRV: 192.168.225.1,5555,0

OK
AT+CWLANSRV=1
OK

+CWLANMSG: 123456789
```

```
AT+CWLANSRV=0,44444,1
OK
AT+CWLANSRV?
+CWLANSRV: 192.168.225.1,44444,1

OK
AT+CWLANSRV=1
OK

+CWLANMSG: 10

+CWLANMSG: 20

+CWLANMSG: 30

+CWLANMSG: 40

+CWLANMSG: 50

AT+CWLANMGET=30
+CWLANMGET: 030,12345678901234567890123456789O

OK
AT+CWLANMGET=30
+CWLANMGET: 020,12345678901234567890

OK
```

27.3.13 AT+CWLANMSG Send message

Must open the lan server first (AT+CWLANSRV=1).

AT+CWLANMSG Send message	
Write Command **AT+CWLANMSG=<tx_msg>**	Response **OK**
Received urc message **+CWLANMSG: <rx_msg><tail>**	
Received urc message **+CWLANMSG: <cached_len>**	
Parameter Saving Mode	-

Maximum Response Time	-
Reference	

Defined Values

<tx_msg>	Hexadecimal string.The max length of message is 512.
<rx_msg>	ASCII string. (1)The message must end with 0x0A from the client. (2)The max length of <message> is 1024,and ignore others.
<tail>	0x0D0A0D0D0A Normal tail. 0x0D0D0A The message has 0x00.
<cached_len>	Cached bytes. The max length is 10*1024.

Example

```
AT+CWLANSRV=1
OK
AT+CWLANMSG=”31323434”
OK

+CWLANMSG: 1234\r\n\r\r\n
```

27.3.14 AT+CWLANMGET Manual get cached bytes

Must open the lan server first (AT+CWLANSRV=1).

AT+CWLANMGET Manual get cached bytes	
Read Command **AT+CWLANMGET?**	Response **+CWLANMGET: <cached_len>** **OK**
Write Command **AT+CWLANMGET=<len>**	Response **+CWLANMGET: <len>** **<msg>** **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<len>	The length customer want to get.Max length is 100.
<msg>	Received message.
<cached_len>	Cached bytes. The max length is 10*1024.

Example

```
AT+CWLANSRV=1
OK

+CWLANMSG: 110

AT+CWLANMGET=100
+CWLANMGET: 100
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

OK
AT+CWLANMGET?
+CWLANMGET: 10

OK
```

27.3.15 AT+CWMACADDR Get MAC address

AT+CWMACADDR Get MAC address	
Read Command **AT+CWMACADDR?**	Response **[<number>,<mac_addr>** **[… …]]** **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<number>	0 host mac addr 1 client mac addr … client mac addr
<mac_addr>	Device mac address

Example

```
AT+CWMACADDR?
0,00:0A:F5:88:88:8F
1,74:23:44:8f:64:fd

OK
```

27.3.16 AT+CWNETCNCT Query the connection to the network

AT+CWNETCNCT Query the connection to the network	
Read Command AT+CWNETCNCT?	Response +CWNETCNCT: <flag> OK
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<flag>	0	disconnect
	1	connect

Example

```
AT+CWNETCNCT?
+CWNETCNCT: 1

OK
```

27.3.17 AT+CWSTAIP Get STA mode IP address

AT+CWSTAIP Get STA mode IP address	
Read Command AT+CWSTAIP?	Response [+CWSTAIP: <ip address>] OK
Parameter Saving Mode	-

Maximum Response Time	-
Reference	

Defined Values

<ip address>	the station IP address

Example

```
AT+CWSTAIP?
+CWSTAIP: 192.168.11.27

OK
```

27.3.18 AT+CWSTASCAN Scan WIFI network

AT+CWSTASCAN Scan WIFI network	
Read Command **AT+CWSTASCAN?**	Response **+CWSTASCAN: <flag_show_signal>** **OK**
Write Command **AT+CWSTASCAN=<flag_show_signal >**	Response **OK**
Read Command **AT+CWSTASCAN**	Response **[+CWSTASCAN:** **<bssid>,<ssid>[,signal]** **[… …]]** **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<flag_show_signal>	0 – Don't show the signal level. It's the default value. 1 – Show the signal level.
<bssid>	The MAC address of external wireless network.
<ssid>	The SSID name of external wireless network.
<signal>	The signal level of external wireless network.

Example

```
AT+CWSTASCAN
+CWSTASCAN:
4c:e6:76:49:2a:48, simtest

OK
AT+CWSTASCAN=1
OK
AT+CWSTASCAN?
+CWSTASCAN: 1

OK
AT+CWSTASCAN
+CWSTASCAN:
f4:83:cd:d8:24:c8,TP-LINK_24C8,-52
80:89:17:10:e6:23,TP-LINK_SW2,-58
14:2d:27:24:98:61,Public,-58
bc:46:99:38:e2:ca,TP-LINK_E2CA,-64
0c:72:d9:49:25:8b,nubia-WD670-258B,-92
50:2b:73:c0:aa:d9,Tenda_C0AAD9,-68

OK
```

27.3.19 AT+CWSTACFG STA mode configuration setting

AT+CWSTACFG STA mode configuration setting	
Read Command AT+CWSTACFG?	Response +CWSTACFG: <ssid>[,<security>,<proto>,<psk>] OK
Write Command AT+CWSTACFG=<ssid>[,<security>,<proto>,<psk>]	Response OK
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<ssid>	The SSID name of external wireless network.
<security>	Reserved value.

<proto>	Reserved value.
<psk>	The password of external wireless network.

NOTE

1. The configselect value must set to 2;
2. The <security> and <proto> are reserved value which is in ort to compatible with previous versions. These 2 parameters can be entered NULL or any combination.

Example

```
AT+CWSTACFG= "simtest",2,1,"1234567890"
OK
AT+CWSTACFG?
+CWSTACFG: "simtest",,,"1234567890"

OK
AT+CWSTACFG= "simtest",,,"1234567890"
OK
AT+CWSTACFG?
+CWSTACFG: "simtest",,,"1234567890"

OK
AT+CWSTACFG= "simtest",,,""
OK
AT+CWSTACFG?
+CWSTACFG: "simtest"

OK
AT+CWSTACFG= "simtest"
OK
AT+CWSTACFG?
+CWSTACFG: "simtest"

OK
```

27.3.20 AT+CWSTAINIT STA mode setting

AT+CWSTAINIT STA mode setting	
Test Command	Response
AT+CWSTAINIT=?	**+CWSTAINIT: (0-1)**

	OK
Read Command **AT+CWSTAINIT?**	Response **+CWSTAINIT: <type>** **OK**
Write Command **AT+CWSTAINIT=<type>**	Response **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<type>	0 close station mode 1 open station mode

Example

```
AT+CWSTAINIT=?
+CWSTAINIT: (0-1)

OK
AT+CWSTAINIT=0
OK
AT+CWSTAINIT?
+CWSTAINIT: 0

OK
```

27.3.21 AT+CWUSRINFO Auth info of wifi data call setting

The username and password are only for CDMA/EVDO network mode.

AT+CWUSRINFO Auth information of wifi data call setting	
Test Command **AT+CWUSRINFO=?**	Response **+CWUSRINFO: (1-127),(1-127)** **OK**
Read Command **AT+CWUSRINFO?**	Response **+CWUSRINFO: <usrname>,<password>**

	OK
Write Command **AT+CWUSRINFO=<usrname>,<password>**	Response **OK**
Parameter Saving Mode	-
Maximum Response Time	-
Reference	

Defined Values

<usrname>	username string. The length is from 1 to 127.
<password>	password string. The length is from 1 to 127.

NOTE

1. It need to reset when set the username and password.
2. If not set the username and password, the default value is "ctnet@mycdma.cn" and "vnet.mobi".

Example

```
AT+CWUSRINFO=?
+CWUSRINFO: (1-127),(1-127)

OK
AT+CWUSRINFO?
+CWUSRINFO: "ctnet@mycdma.cn","vnet.mobi"

OK
AT+CWUSRINFO=”username”,"pwd"
OK
```

28. AT Commands for BT

28.1 Overview of AT Commands for BT

Command	Description
AT+BTPOWER	Open/Close BT
AT+BTHOST	Get/Set host name
AT+BTSCAN	Scan BT devices
AT+BTIOCAP	IOCAP Mode Setting
AT+BTPAIR	Pair with the paired BT devices
AT+BTUNPAIR	Unpair with the paired BT devices
AT+BTPAIRED	Get Paired BT devices
AT+BTSPPSRV	Active/Deactive spp server
AT+BTSPPPROF	Get remote device spp status
AT+BTSPPCONN	SPP connect/disconnect
AT+BTSPPSEND	SPP send data
AT+BTGATTREG	GATT Register
AT+BTGATTACT	GATT Active
AT+BTGATTCREDB	GATT Create DB
AT+BTGATTCRESRV	GATT Create Service
AT+BTGATTCRECHAR	Create Service characteristic
AT+BTGATTCRECHARDES	Create Service characteristic description
AT+BTGATTSRVADD	DB Add To GATT Server
AT+BTGATTREADCFM	Response to BTGATTREADIND
AT+BTGATTWRCFM	Response to BVTGATTWRIND
AT+BTGATTNOTIFY	Send Notification to client
AT+BTGATTSENDIND	Send Indication to client
+BTSPPRECV	SPP receive data
+BTGATTCONN	Client connect status
+BTGATTREADIND	Receive client read request
+BTGATTWRIND	Receive client write request

28.2 Detailed Description of AT Commands for BT

Command	Description
AT+BTPOWER	Open/Close BT
AT+BTHOST	Get/Set host name
AT+BTSCAN	Scan BT devices
AT+BTIOCAP	IOCAP Mode Setting
AT+BTPAIR	Pair with the paired BT devices
AT+BTUNPAIR	Unpair with the paired BT devices
AT+BTPAIRED	Get Paired BT devices
AT+BTSPPSRV	Active/Deactive spp server
AT+BTSPPPROF	Get remote device spp status
AT+BTSPPCONN	SPP connect/disconnect
AT+BTSPPSEND	SPP send data
AT+BTGATTREG	GATT Register
AT+BTGATTACT	GATT Active
AT+BTGATTCREDB	GATT Create DB
AT+BTGATTCRESRV	GATT Create Service
AT+BTGATTCRECHAR	Create Service characteristic
AT+BTGATTCRECHARDES	Create Service characteristic description
AT+BTGATTSRVADD	DB Add To GATT Server
AT+BTGATTREADCFM	Response to BTGATTREADIND
AT+BTGATTWRCFM	Response to BVTGATTWRIND
AT+BTGATTNOTIFY	Send Notification to client
AT+BTGATTSENDIND	Send Indication to client
+BTSPPRECV	SPP receive data
+BTGATTCONN	Client connect status
+BTGATTREADIND	Receive client read request
+BTGATTWRIND	Receive client write request

28.2.1 AT+BTPOWER Open/Close BT

AT+BTPOWER Open/Close BT	
Test Command **AT+BTPOWER=?**	Response **+BTPOWER: (0-1)** **OK**
Read Command	**+BTPOWER: <flag>**

AT+BTPOWER?	**OK**
Write Command **AT+BTPOWER=<flag>[,<debug_switch>]**	Response **OK** or **ERROR**

Defined Values

<flag >	0: Stop bt csr app 1: Start bt csr app
<debug_switch >	Only allowed set to 1, means to save bt log file after csr app is start.

Example

```
AT+BTPOWER?
+BTPOWER: 1
OK
AT+BTPOWER=0
OK
AT+BTPOWER=1,1
OK
```

NOTE

- When <**flag**> set to 0, **<debug_switch>** can not be set.

28.2.2 AT+BTHOST Get/Set host name

AT+BTHOST Get/Set host name	
Read Command **AT+BTHOST?**	Response **+BTHOST: <host_name>,<host mac addr>** **OK**
Write Command **AT+BTHOST=<"btname">**	Response **OK** or **ERROR**

Defined Values

<btname>	new Bluetooth name string. Support Chinese characters. Max length 64
<host mac addr>	Bluetooth mac address format(xx:xx:xx:xx:xx:xx), x(0-9,A-F) The default value is SIM7600_BT_xxxxxx(mac addr 3 lower bytes).

Example

```
AT+BTHOST?
+BTHOST: SIM7600_BT_AC8DD9, 00:02:5B:AC:8D:D9

OK
AT+BTHOST =”abc”
OK
```

28.2.3 AT+BTSCAN Scan BT devices

AT+BTSCAN Scan BT devices	
Test Command **AT+BTSCAN=?**	Response **+BTSCAN: (0-1),(0-1),(6-60)** **OK**
Write Command **AT+BTSCAN=<”doscan”>[,<mode>[,<timeout>]]**	Response **OK** **+BTSCAN: <scan status>, <index1>, <BT name>, <Mac Addr>, <RSSI level>** **+BTSCAN: <scan status>, <index2>, <BT name>, <Mac Addr>, <RSSI level>** [...]] **+BTSCAN: 1 //scan end flag** or **ERROR**

Defined Values

<doscan>	0:stop scan 1:scan
<mode>	0:don’t hide paired devices 1:hide paired devices
<timeout>	Timeout seconds. Default value is 10

<scan status>	0:scanning 1:scan ended
<index>	The index of remote bluetooth device, the value start with 1.
<BT name>	The bluetooth name of remote device.
<Mac Addr>	The bluetooth mac address of the remote device.
<RSSI level>	the rssi level of the device

Example

```
AT+BTSCAN=1,0,10
OK
+BTSCAN: 0, 1, MKRJ2B-GONGYONG, B8:86:87:43:4B:6A, 186
+BTSCAN: 0, 2, MK-JUMPING, 00:19:86:00:08:60, 184
+BTSCAN: 0, 3, OPPO A57, 4C:18:9A:89:88:7E, 174
+BTSCAN: 0, 4, ww炸, C4:0B:CB:3E:68:62, 173
+BTSCAN: 0, 5, ofo, F7:51:3B:1F:AF:B5, 165
+BTSCAN: 1
AT+BTSCAN=0,0,10
OK
```

28.2.4 AT+BTIOCAP IOCAP Mode Setting

AT+BTIOCAP IOCAP Mode Setting	
Test Command **AT+BTIOCAP=?**	Response **+BTIOCAP: (0-3)** **OK**
Write Command **AT+BTIOCAP=<mode>**	Response **+BTIOCAP: 1** **OK** or **ERROR**

Defined Values

<mode>	0:Display Only Device 1:Display and Yes and No Capable 2:Keyboard Only 3:No Display or Input Device

Example

```
AT+BTIOCAP=3
+BTIOCAP: 1

OK
```

28.2.5 AT+BTPAIR Pair with other BT device

AT+BTPAIR Pair with other BT device	
Test Command **AT+BTPAIR=?**	Response **+BTPAIR: (index)** **OK**
Pair Command **AT+BTPAIR=0,<scan index>**	Response **OK** **+BTPAIRING: <mode>, <device name>,<device mac>,[<passkey>]** or **ERROR**
Accept Command **AT+BTPAIR=<mode>,<accept>[,<pakey>]**	Response **OK** **+BTPAIR: <pair result>[,<device name>,<device mac>]** or **ERROR**

Defined Values

<mode>	1:Compare mode need user send accept command 2:Passkey mode need user send accept command and passkey 3:Rebond mode need user send accept command 4:Notify mode just notify user pairing status, user do nothing 5:Just work mode will not receive this mode, user do nothing 6:Pin code mode need user send accept command and pin code
<passkey>	Random generate 6 numberic code
<scan index>	BTSCAN response index
<device name>	The bluetooth name of connected device
<device mac>	The bluetooth mac address of the connected device
<pair result>	0:fail

	1:success
<accept>	0:reject 1:accept

Example

```
AT+BTSCAN=1,0,10
+BTSCAN: 0, 1, OPPO R7Plusm, 2C:5B:B8:1A:33:3C, 189
+BTSCAN: 0, 2, MK-JUMPING, 00:19:86:00:08:60, 183
+BTSCAN: 0, 3, MI Band 2, C8:EB:37:B3:56:57, 179
+BTSCAN: 0, 4, BU3-ZHANGWEI, 00:1A:7D:DA:71:11, 178
+BTSCAN: 0, 5, ww, C4:0B:CB:3E:68:62, 174
+BTSCAN:1

OK

AT+BTPAIR=0,5
OK
+BTPAIRING: 1, ww, C4:0B:CB:3E:68:62, 623850

AT+BTPAIR=1,1
OK
+BTPAIR: 1, ww, C4:0B:CB:3E:68:62
```

NOTE

- The time out of pairing is about 30 seconds
- Whether the pairing is initiative or passive, "AT+BTPAIR" Accept command must be execute after "+BTPAIRING: **<mode>**, **<device name>**, **<device mac>**, [**<passkey>**]" urc was reported.

28.2.6 AT+BTUNPAIR Unpair with other BT device

AT+BTUNPAIR Unpair with other BT device	
Test Command **AT+BTUNPAIR=?**	Response **+BTUNPAIR: (index)** **OK**
Write Command **AT+BTUNPAIR=<paired index>**	Response **OK** **+BTUNPAIR: <status>** or **ERROR**

Defined Values

<index>	Interger, the response of AT+BTPAIRED.
<status>	0:fail 1:success

Example

```
AT+BTUNPAIR=1
+BTUNPAIR: 1

OK
```

28.2.7 AT+BTPAIRED Get paired with BT device

AT+BTPAIRED Get paired with BT device	
Read Command **AT+BTPAIRED?**	Response **OK** **+BTPAERED: <paired devices num>,<index>,<BT name>,<MAC addr>**

Defined Values

<paired devices num>	The total number of bonded devices
<index>	The index of current bond device
<BT name>	refer to AT+BTSCAN
<MAC addr>	refer to AT+BTSCAN

Example

```
AT+BTPAIRED?
OK
+BTPAIRED: 2, 1, Honor V8, 60:83:34:82:CC:A3
+BTPAIRED: 2, 2, ww C4:0B:CB:3E:68:62
```

28.2.8 AT+BTSPPSRV Active/Deactive spp server

AT+BTSPPSRV Active/Deactive spp server	
Test Command **AT+BTSPPSRV=?**	Response **+BTSPPSRV: (0-1)** **OK**
Read Command **AT+BTSPPSRV?**	Response **+BTSPPSRV: <status>** **OK**
Write Command **AT+BTSPPSRV=<flag>**	Response **OK** **+BTSPPSRV: <status>** or **ERROR**

Defined Values

<flag>	0:deactive 1:active
<status>	0:deactived 1:actived

Example

```
AT+BTSPPSRV?
+BTSERVER: 0
OK
AT+BTSPPSRV=1
OK
+BTSPPSRV: 1
```

28.2.9 AT+BTSPPPROF Get remote device spp status

AT+BTSPPPROF Get remote device spp status	
Read Command **AT+BTSPPPROF=<index>**	Read Command **+BTSPPPROF: <status>** **OK** or **ERROR**

Defined Values

<index>	the index of response list of AT+BTPAIRED command
<status>	0:device SPP service is not active 1:device SPP service is active

Example

```
AT+BTPAIRED?
OK
+BTPAIRED: 2, 1, Honor V8, 60:83:34:82:CC:A3
+BTPAIRED: 2, 2, ww C4:0B:CB:3E:68:62
AT+BTSPPPROF=2
OK
+BTSPPPROF:1
```

28.2.10 AT+BTSPPCONN SPP connect/disconnect

AT+BTSPPCONN SPP connect/disconnect	
Test Command **AT+BTSPPCONN=?**	Response **+BTSPPCONN: (0-1)** **OK**
Read Command **AT+BTSPPCONN?**	Response **+BTSPPCONN: <status>** **OK**
Write Command **AT+BTSPPCONN=<action>[,<paired index>]**	Response **OK** **+BTSPPCONN: <status>[,<max frame size>][,<device mac>]** or **ERROR**

Defined Values

<action>	0:disconect 1:connect
<paired index>	The response of AT+BTPAIRED. The max value is 64.
<status>	0:disconnected 1:connected

Example

```
AT+BTSPPCONN?
+BTSPPCONN: 0
OK
AT+BTSPPCONN=1,1
OK
+BTSPPCONN: 1, 990, C4:07:2F:C5:D1:8A
```

NOTE

- The device may receive **+BTSPPCONN:<status>[,<max frame size>] [,<device mac>]** when other device connected successfully.

28.2.11 AT+BTSPPSEND SPP send data

AT+BTSPPSEND SPP send data	
Write Command **AT+BTSPPSEND=<data>**	Response **OK** **+BTSPPSEND: <result>** or **ERROR**

Defined Values

<data>	Format : ucs2 "ucs2": 16-bit universal multiple-octet coded character set; UCS2 character strings are converted to hexadecimal number from 0000 to FFFF. For examples : If we want to send a string "123abc" The data is : 00310032003300610062006300
<result>	0:send fail 1:send success

Example

```
AT+BTSPPSEND=00310032003300610062006300
OK
AT+BTSPPSEND: 1
```

28.2.12 AT+BTGATTREG GATT Register

AT+BTGATTREG GATT Register	
Write Command **AT+BTGATTREG=<status>**	Response **+BTGATTREG: <status>** **OK** or **ERROR**

Defined Values

<status>	1: register 0: unregister

Example

```
AT+BTGATTREG=1
+BTGATTREG: 1

OK
```

28.2.13 AT+BTGATTACT GATT Active

AT+BTGATTACT GATT Active	
Execution Command **AT+BTGATTACT**	Response **+BTGATTACT: <status>** **OK** or **ERROR**
Write Command **AT+BTGATTACT=<auto_broadcast>[,<perferedMTU>]**	Response **OK** or **ERROR**

Defined Values

<status>	1:active 0:not avtive
<auto_brodcast>	0 - disable auto activate GATT after a connection was closed 1 - enable auto activate GATT after a connection was closed
<perferedMTU>	A integer value from 24 to 512, means to the maximum size of any packet sent between a client and a server. If not set, default packet size is 23bytes. The details refer to Note.

Example

```
AT+BTGATTACT
+BTGATTACT: 1

OK
```

28.2.14 AT+BTGATTCREDB GATT Create DB

AT+BTGATTCREDB GATT Create DB	
Execution Command **AT+BTGATTCREDB**	Response **+BTGATTCREDB: <status>** **OK** or **ERROR**

Defined Values

<status>	1:sucess 0:fail

Example

```
AT+BTGATTCREDB
+BTGATTCREDB: 1

OK
```

28.2.15 AT+BTGATTCRESRV GATT Create Service

AT+BTGATTCRESRV GATT Create Service	
Write Command **AT+BTGATTCRESRV=<uuid>**	Response **+BTGATTCRESRV: <status>** **OK** or **ERROR**

Defined Values

<uuid>	Service id,4 Hex character or 32 Hex character
<status>	1:sucess 0:fail

Example

```
AT+BTGATTCRESRV=34A3
+BTGATTCRESRV: 1

OK
```

28.2.16 AT+BTGATTCRECHAR Create Service characteristic

AT+BTGATTCRECHAR Create Service characteristic	
Write Command **AT+BTGATTCRECHAR=<uuid>,<property>,<permission>**	Response **+BTGATTCRECHAR: <status>,<0Xuuid>,<handle>** **OK** or **ERROR**

Defined Values

<uuid>	UUID of this characteristic. A string with hex value. The length of it only can be set 4 or 32.
<property>	Properties of this characteristic.
<permission>	Permission of this characteristic.

<status>	1:sucess 0:fail
<handle>	Int, Characteristic handle

Example

```
AT+BTGATTCRECHAR=34567,2,16
+BTGATTCRECHAR: 1,0X4567,13

OK
```

28.2.17 AT+BTGATTCRECHARDES Create Service characteristic description

AT+BTGATTCRECHARDES	Create Service characteristic description
Excution Command AT+BTGATTCRECHARDES	Response **+BTGATTCRECHARDES: <status>** **OK** or **ERROR**

Defined Values

<status>	1:sucess 0:fail

Example

```
AT+BTGATTCRECHARDES
+BTGATTCRECHARDES: 1

OK
```

28.2.18 AT+BTGATTSRVADD DB Add To GATT Server

AT+BTGATTSRVADD	DB Add To GATT Server
Excution Command AT+BTGATTSRVADD	Response **+BTGATTSRVADD: <status>**

	OK or **ERROR**

Defined Values

<status>	1:sucess 0:fail

Example

```
AT+BTGATTSRVADD
+BTGATTSRVADD: 1

OK
```

28.2.19 AT+BTGATTREADCFM Response to BTGATTREADIND

AT+BTGATTREADCFM Response to BTGATTREADIND	
Write Command **AT+BTGATTREADCFM=<respCode>,<data>**	Response **+BTGATTREADCFM: 1** **OK** or **ERROR**

Defined Values

<respCode>	Response result for client request. The range is 0-255. 0: sucess Others： not support, invalid parameter
<data>	character, Response data to BTGATTREADIND,if data length less than maxlen(BTGATTREADIND return), data will be send immediately to client， if data length equal to maxlen， the module will receive BTGATTREADIND again till data length less than maxlen.

Example

```
+BTGATTREADIND: 13,22

AT+BTGATTREADCFM=0,123456
```

```
+BTGATTREADCFM:1

OK
```

28.2.20 AT+BTGATTWRCFM Response to BTGATTWRIND

AT+BTGATTWRCFM Response to BTGATTWRIND	
Write Command **AT+BTGATTWRCFM=<result >**	Response **+BTGATTWRCFM: <status>** **OK** or **ERROR**

Defined Values

<result>	0: sucess
<status>	1: sucess

Example

```
+BTGATTWRIND: 15,DB12C8

AT+BTGATTWRCFM=0
+BTGATTWRCFM: 1

OK
```

28.2.21 AT+BTGATTNOTIFY Send Notification to client

AT+BTGATTNOTIFY Send Notification to client	
Write Command **AT+BTGATTNOTIFY=<handle>,<data>**	Response **+BTGATTNOTIFY: <status>** **OK** or **ERROR**

Defined Values

<handle>	Int, Characteristic handle, (2.17 response returns, nd the characteristic's property is indication)
<data>	character, Data to be send, (max length is 20)
<status>	1:sucess 0:fail

Example

```
AT+BTGATTNOTIFY=17,34567
+BTGATTNOTIFY:1

OK
```

28.2.22 AT+BTGATTSENDIND Send Indication to client

AT+BTGATTSENDIND Send Indication to client	
Write Command **AT+BTGATTSENDIND=<handle>,<data>**	Response **+BTGATTSENDIND: <status>** **OK** or **ERROR**

Defined Values

<handle>	Int, Characteristic handle, (2.17 response returns, nd the characteristic's property is indication)
<data>	character, Data to be send, (max length is 20)
<status>	1:sucess 0:fail

Example

```
AT+BTGATTSENDIND=19,34567
+BTGATTSENDIND: 1

OK
```

28.2.23 +BTSPPRECV SPP receive data

+BTSPPRECV SPP receive data	
	Response **+BTSPPRECV: <data len>,<data>**

Defined Values

<data len>	Integer type，0 - 100
<data>	Format : ucs2 For examples : If we have received a string 00310032003300610062006 3 Means receive a string “123abc”

Example

```
+BTSPPRECV=12, 003100320033006100620063
+BTGATTSENDIND: 1
```

28.2.24 +BTGATTCONN Client connect status

+BTGATTCONN Client connect status	
	Response **+BTGATTCONN: <status>,<device mac>**

Defined Values

<status>	1:connected 0:disconnected

Example

```
+BTGATTCONN: 1, 68:68:79:6D:75:26
```

28.2.25 +BTGATTREADIND Receive client read request

+BTGATTREADIND Receive client read request	
	Response **+BTGATTREADIND: <handle>,<maxlen>**

Defined Values

<handle>	Int, Characteristic handle
<maxlen>	The maximum length that the value of the attribute must have.

Example

```
+BTGATTREADIND: 13,22
```

28.2.26 +BTGATTWRIND Receive client write request

+BTGATTWRIND Receive client write request	
	Response **+BTGATTWRIND: <handle>,<data>**

Defined Values

<handle>	Int, Characteristic handle
<data>	Data to be writed (Hex charcters)

Example

```
+BTGATTWRIND: 15,DB12C8
```