



NE310H2 and NL865H2 AT Command Reference Guide

1VV0301611 Rev. 7 – 2020-08-18

TELIT
TECHNICAL
DOCUMENTATION

SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE

NOTICE

While reasonable efforts have been made to assure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from use of the information obtained herein. The information in this document has been carefully checked and is believed to be reliable. However, no responsibility is assumed for inaccuracies or omissions. Telit reserves the right to make changes to any products described herein and reserves the right to revise this document and to make changes from time to time in content hereof with no obligation to notify any person of revisions or changes. Telit does not assume any liability arising out of the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

It is possible that this publication may contain references to, or information about Telit products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that Telit intends to announce such Telit products, programming, or services in your country.

COPYRIGHTS

This instruction manual and the Telit products described in this instruction manual may be, include or describe copyrighted Telit material, such as computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any copyrighted material of Telit and its licensors contained herein or in the Telit products described in this instruction manual may not be copied, reproduced, distributed, merged or modified in any manner without the express written permission of Telit. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit, as arises by operation of law in the sale of a product.

COMPUTER SOFTWARE COPYRIGHTS

The Telit and 3rd Party supplied Software (SW) products described in this instruction manual may include copyrighted Telit and other 3rd Party supplied computer programs stored in semiconductor memories or other media. Laws in the Italy and other countries preserve for Telit and other 3rd Party supplied SW certain exclusive rights for copyrighted computer programs, including the exclusive right to copy or reproduce in any form the copyrighted computer program. Accordingly, any copyrighted Telit or other 3rd Party supplied SW computer programs contained in the Telit products described in this instruction manual may not be copied (reverse engineered) or reproduced in any manner without the express written permission of Telit or the 3rd Party SW supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or otherwise, any license under the copyrights, patents or patent applications of Telit or other 3rd Party supplied SW, except for the normal non-exclusive, royalty free license to use that arises by operation of law in the sale of a product.

USAGE AND DISCLOSURE RESTRICTIONS

I. License Agreements

The software described in this document is the property of Telit and its licensors. It is furnished by express license agreement only and may be used only in accordance with the terms of such an agreement.

II. Copyrighted Materials

Software and documentation are copyrighted materials. Making unauthorized copies is prohibited by law. No part of the software or documentation may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, without prior written permission of Telit

III. High Risk Materials

Components, units, or third-party products used in the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: the operation of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems (High Risk Activities"). Telit and its supplier(s) specifically disclaim any expressed or implied warranty of fitness for such High Risk Activities.

IV. Trademarks

TELIT and the Stylized T Logo are registered in Trademark Office. All other product or service names are the property of their respective owners.

V. Third Party Rights

The software may include Third Party Right software. In this case you agree to comply with all terms and conditions imposed on you in respect of such separate software. In addition to Third Party Terms, the disclaimer of warranty and limitation of liability provisions in this License shall apply to the Third Party Right software.

TELIT HEREBY DISCLAIMS ANY AND ALL WARRANTIES EXPRESS OR IMPLIED FROM ANY THIRD PARTIES REGARDING ANY SEPARATE FILES, ANY THIRD PARTY MATERIALS INCLUDED IN THE SOFTWARE, ANY THIRD PARTY MATERIALS FROM WHICH THE SOFTWARE IS DERIVED (COLLECTIVELY "OTHER CODE"), AND THE USE OF ANY OR ALL THE OTHER CODE IN CONNECTION WITH THE SOFTWARE, INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF SATISFACTORY QUALITY OR FITNESS FOR A PARTICULAR PURPOSE.

NO THIRD PARTY LICENSORS OF OTHER CODE SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND WHETHER MADE UNDER CONTRACT, TORT OR OTHER LEGAL THEORY, ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE OTHER CODE OR THE EXERCISE OF ANY RIGHTS GRANTED UNDER EITHER OR BOTH THIS LICENSE AND THE LEGAL TERMS APPLICABLE TO ANY SEPARATE FILES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

APPLICABILITY TABLE

PRODUCTS

■ ■ NE310H2-W1

■ ■ NL865H2-W1

Contents

NOTICE	2
COPYRIGHTS	2
COMPUTER SOFTWARE COPYRIGHTS.....	2
USAGE AND DISCLOSURE RESTRICTIONS	3
I. License Agreements	3
II. Copyrighted Materials	3
III. High Risk Materials	3
IV. Trademarks.....	3
V. Third Party Rights	3
APPLICABILITY TABLE	4
CONTENTS	5
1. INTRODUCTION	13
1.1. Scope	13
1.2. Audience.....	13
1.3. Contact Information, Support	13
1.4. Text Conventions	14
2. AT COMMANDS.....	15
2.1. Definitions	15
2.2. AT Command Syntax.....	15
3. IDENTIFICATION COMMANDS	18
3.1. AT+CGMI Request manufacturer identification	18
3.2. AT+CGMM Request model identification	18
3.3. AT+CGMR Request revision identification	19
3.4. AT+CGSN Request IMEI	19
3.5. AT+CIMI Request international mobile subscriber identity	20
3.6. *MATREADY	21
3.7. AT+GMI Request manufacturer identification.....	21
3.8. AT+GMM Request model identification	22
3.9. AT+GMR Request revision identification.....	22
3.10. AT+GOI/+CGIO Request global object identification.....	23

3.11.	AT+GSN Request TA serial number identification (IMEI)	23
3.12.	AT+ICF Set TE-TA control character framing.....	24
3.13.	AT+CLCK Facility Lock	25
3.14.	AT+CPWD Change password.....	26
3.15.	AT+CREG Network registration	27
3.16.	AT+CRSM Restricted SIM Access	28
3.17.	AT+CSCS Select TE Character Set	29
3.18.	AT+CNUM Subscriber Number	30
3.19.	AT+CPOL Preferred operator list	30
3.20.	AT+CSIM Generic SIM Access	31
3.21.	AT+CTZR Time Zone Reporting	32
4.	GENERAL COMMANDS	33
4.1.	ATE Command Echo	33
4.2.	AT&D Set circuit Data Terminal Ready(DTR) function mod.....	33
4.3.	ATV DCE Response Format	34
4.4.	AT+CFUN Set Phone Functionality	34
4.5.	AT+CMEE Report Mobile Termination Error	35
4.6.	+CME ERROR ME Error Result Code	36
4.7.	AT+CEER Extended Error Report.....	38
4.8.	ATI Display product identification information.....	39
4.9.	ATQ Set result code presentation mode.....	40
4.10.	ATN1 initial setting to handshake at highest speed	40
4.11.	ATS0 Set number of rings before automatically answering the call	41
4.12.	ATS1 Ring Counter	41
4.13.	ATS2 Set escape sequence character	42
4.14.	ATS3 Set command line termination character	43
4.15.	ATS4 Set response formatting character.....	44
4.16.	ATS5 Set command line editing character	45
4.17.	ATS7 Set number of seconds to wait for connection completion	45
4.18.	ATS10 Set disconnect delay after indicating the absence of data carrier	46
4.19.	ATS12 Set Escape Code Guard Time.....	46
4.20.	ATS25 Set DTR change time	48
4.21.	ATS95 enable extended result codes.....	49
4.22.	ATX Set CONNECT result code format and call monitoring	49
4.23.	AT+FCLASS Set fax class	50
4.24.	AT+GCAP Request complete TA capabilities list	50

4.25.	AT&F Set all current parameters to manufacturer defaults	51
4.26.	AT&K Flow control setting	51
4.27.	AT&V Display current configuration.....	52
4.28.	AT&W Store current parameter to user defined profile	52
4.29.	AT&C Set circuit Data Carrier Detect (DCD) function mode	52
4.30.	AT&D Set circuit Data Terminal Ready (DTR) function mode.....	53
5.	SERIAL INTERFACE CONTROL COMMANDS	54
5.1.	AT+IPR Set the Baud rate of UART	54
5.2.	AT+IFC Local data flow control	55
5.3.	AT#TCOMWRT Firmware Upgrade through UART	56
5.4.	AT+CMUX Serial Multiplexer control.....	58
5.5.	AT#TCLRBDRT Clear historical baud rate	59
6.	SIM COMMANDS	60
6.1.	AT+CPIN Enter PIN	60
6.2.	AT*MICCID Read USIM ICCID file.....	61
7.	NETWORK SERVICE COMMANDS.....	62
7.1.	AT+CEREG EPS Network registration	62
7.2.	AT+COPS PLMN Reselection.....	64
7.3.	AT+CESQ Get signal quality	65
7.4.	AT+CSQ Get signal quality	66
7.5.	AT+CTZU Automatic Time Zone Update.....	67
7.6.	AT+CCLK Clock Management	68
7.7.	AT*MFRCLLCK Lock UE to specific frequency and optionally Cell ID.....	69
7.8.	AT*MBAND Query Current Modem NB-IOT Operating Band	70
7.9.	AT*MBSC Lock BAND	71
7.10.	AT*MENGINFO Query current network status and cell information.....	72
7.11.	AT*MNBIOTRAI Actively release RRC connection.....	75
7.12.	AT*RRCRELEASE Actively release RRC connection	76
7.13.	AT+CCIOTOPT CloT Optimization Configuration.....	77
7.14.	AT+CCHO Open UICC Logical Channel	78
7.15.	AT+CCHC Close UICC Logical Channel.....	78
7.16.	AT+CGLA Generic UICC Logical Channel Access.....	79
7.17.	AT+CPINR Remaining PIN Retries	80
7.18.	AT+CGATT GPRS/Packet Domain attach or detach.....	81
7.19.	AT+CGACT PDP context activate or deactivate.....	82

7.20.	AT+CGPADDR Show PDP address.....	83
7.21.	AT+IPCONFIG Query the device's IP address.....	84
7.22.	AT+CGEREP Packet Domain Event Reporting.....	84
7.23.	AT+CGREG Network registration status	86
7.24.	AT+CGPIAF Printing IP Address Format	87
7.25.	AT+CGDEL Delete Non-Active PDP Contexts	89
7.26.	AT+CGAUTH Define PDP Context Authentication Parameters.....	90
7.27.	AT+CGDATA Enter Data State	91
7.28.	AT*MBANDSL Set Modem NB-IOT Search Prefer Band List.....	91
7.29.	AT*MNBIOTDT NB-IoT Data Type.....	92
7.30.	AT*MLPINFO Report packet domain packet counters.....	93
8.	POWER SAVING COMMANDS.....	94
8.1.	AT+CEDRXS eDRX Settings	94
8.2.	AT+CEDRXRDP eDRX to read dynamic parameters.....	97
8.3.	AT*MEDRXCFCG eDRX configuration.....	98
8.4.	AT+CPSMS set power saving mode (PSM)	100
8.5.	AT*MNBIOTEVENT enable/disable active report of PSM	104
8.6.	AT#TSLK enable/disable UE enter sleep	105
9.	PS-DOMAIN COMMANDS	106
9.1.	AT*MCGDEFCONT set the PSD connection settings for PDN connection.....	106
9.2.	AT+CGDCONT Define PDP Context.....	107
9.3.	AT+CGCONTRDP PDP Context Read	108
9.4.	+IP IP address Unsolicited report command	109
10.	HARDWARE COMMANDS	110
10.1.	AT+TRST RESET the Module.....	110
10.2.	AT+TURNOFF Turn Off the Module.....	110
10.3.	AT#TADC Read pin ADC value	110
10.4.	AT#TCONTLED LED status control	111
10.5.	AT+EGPIO Operate GPIO	112
10.6.	AT#SYSHALT System Halt.....	114
10.7.	AT#SHDN Software Shut Down.....	115
10.8.	AT+CBC Query Power Supply Voltage	115
10.9.	AT#I2CWR - Write to I2C.....	116
10.10.	AT#I2CRD - Read to I2C	117
10.11.	AT#SPIOOPEN Initialize modem serial port with SPI protocol.....	118

10.12.	AT#SPICLOSE De-initialize modem serial port for SPI protocol.....	119
10.13.	AT#SPIRW Write a buffer to the SPI and prints the read data.....	120
10.14.	AT#GPIO General Purpose Input/Output Pin Control.....	121
10.15.	AT#E2RI Event Ring Indicator	123
11.	SNTP NETWORK COMMANDS.....	124
11.1.	AT+ESNTPSTART Start to query network time	124
11.2.	AT+ESNTPSTOP Stop to query network time.....	124
11.3.	+ESNTP received network time	124
12.	SMS COMMANDS.....	126
12.1.	AT+CSMS Select Message Service.....	126
12.2.	AT+CMGF Message Format	127
12.3.	AT+CSCA Service Centre Address.....	128
12.4.	AT+CSMP Set Text Mode Parameters.....	129
12.5.	AT+CSAS Save Settings.....	130
12.6.	AT+CRES Restore Settings	130
12.7.	AT+CNMI New Message Indications to TE	132
12.8.	AT+CMGL List Messages	133
12.9.	AT+CMGR Read Message	135
12.10.	AT+CMGS Send Message.....	138
12.11.	AT+CMSS Send Message from Storage.....	139
12.12.	AT+CMGW Write Message to Memory	140
12.13.	AT+CMGD Delete Message.....	141
12.14.	AT+CMGC Send Command.....	142
12.15.	AT+CPMS Preferred Message Storage	143
12.16.	AT+CSDH Show SMS text mode parameters	144
12.17.	AT+CNMA New Message Acknowledgment	145
12.18.	+CMS ERROR Message Service Failure Result Code.....	146
13.	TCP/IP COMMANDS	147
13.1.	AT+ESOC CreateTCP/UDP socket.....	147
13.2.	AT+ESOB Bind local address and local port	147
13.3.	AT+ESOCN Connect socket to remote address and port	148
13.4.	AT+ESOSND Send data to remote via socket	149
13.5.	AT+ESOCL Close socket.....	149
13.6.	+ESONMI Socket message arrived indicator	150
13.7.	+ESOERR Socket error indicator	150

13.8.	AT+ESODATAMODE Change to transparent mode.....	151
13.9.	AT+PING Test IP network connectivity to a remote host.....	152
13.10.	Example: Create TCP socket.....	153
13.11.	Example: Create UDP socket.....	153
14.	MQTT COMMANDS	154
14.1.	AT+EMQSCFG Encryption Configuration of MQTT.....	154
14.2.	AT+EMQSCFGRAW Encryption Configuration of MQTT through transparent transmission.....	157
14.3.	AT+EMQNEW New MQTT.....	160
14.4.	AT+EMQCFG Configure Parameters of MQTT	161
14.5.	AT+EMQCON Send MQTT Connection packet.....	162
14.6.	AT+EMQDISCON Disconnect MQTT.....	163
14.7.	+EMQDISCON Unsolicited report of MQTT disconnection.....	163
14.8.	AT+EMQSUB Send MQTT subscribe packet.....	163
14.9.	AT+EMQUNSUB Send MQTT unsubscribe packet.....	164
14.10.	AT+EMQPUB Send MQTT publish packet.....	164
14.11.	+EMQPUB –Unsolicited report of MQTT publish packet	165
14.12.	Example: create MQTT connection	165
15.	COAP COMMANDS	166
15.1.	AT+ECOAPNEW Create a CoAP client instance	166
15.2.	AT+ECOAPSEND Send CoAP data.....	166
15.3.	AT+ECOAPDEL Destroy the CoAP client instance	168
15.4.	+ECOAPNMI Content from CoAP server	169
15.5.	Example: COAP client send data	170
16.	FTP COMMANDS.....	172
16.1.	AT+FTPOPEN Start FTP file service.....	172
16.2.	AT+FTPCLOSE Close FTP file service	173
16.3.	AT+FTPSIZE Get the Size of FTP File.....	173
16.4.	AT+FTPGET FTP File download.....	174
16.5.	AT+FTPPUTCON Establish upload link	175
16.6.	AT+FTPPUTDATA FTP File upload.....	176
16.7.	AT+FTPCFG configuration FTP file content download format.....	177
16.8.	AT+FTPGETOTA download Difference packet from FTP server.....	177
16.9.	AT+OTAUP Excute FTP FOTA Update.....	178
16.10.	URC of FTP FOTA and LwM2M FOTA	178
16.11.	FTP Process Example	179

16.12.	FTP FOTA Process Example	180
17.	HTTP/HTTPS COMMANDS.....	181
17.1.	AT+EHTTPCREATE Create a HTTP/HTTPS client instance	181
17.2.	AT+EHTTPCON Establish the HTTP/HTTPS connection.....	182
17.3.	AT+EHTTPDISCON Close HTTP/HTTPS connection	182
17.4.	AT+EHTTPEDESTROY Destroy the HTTP/HTTPS client instance	183
17.5.	AT+EHTTPSEND Send HTTP/HTTPS package	183
17.6.	+EHTTPNMIH Header of the response from host	184
17.7.	+EHTTPNMIC Content of the response from host.....	185
17.8.	+EHTTPERR HTTP/HTTPS client connection error indicator.....	185
17.9.	Example: Create HTTP connection	186
17.10.	Example: Create HTTPS connection.....	187
18.	HTTP/HTTPS SIMPLIFIED COMMANDS.....	189
18.1.	AT+HTTPCFG Configure Parameters for HTTP(S) connection.....	189
18.2.	AT+HTTPURL Set URL of HTTP(S) Server	196
18.3.	AT+HTTPGET Send GET Request to HTTP(S) Server.....	197
18.4.	AT+HTTPPOST Send POST Request to HTTP(S) Server	198
18.5.	Result code of HTTP/HTTPS	200
19.	NIDD COMMANDS.....	201
19.1.	AT+NIDD Non-IP Data Delivery AT command	201
20.	LWM2M COMMANDS	202
20.1.	AT+ELMREG Configuration LwM2M instance and establish connection.....	202
20.2.	AT+ELMDEREG Delete LwM2M instance and disconnect.....	204
20.3.	AT+ELMBINDING Configuration binding mode.....	205
20.4.	AT+ELMADDOBJ Add LwM2M object	206
20.5.	AT+ELMDELOBJ Delete LwM2M object	206
20.6.	AT+ ELMUPDATE Update lifetime	207
20.7.	+ELMREAD URC indicate server sending a read request.....	208
20.8.	AT+ELMREADRSP Response to server's read request.....	208
20.9.	+ELMWRITE URC indicate server sending a write request.....	209
20.10.	AT+ELMWRITERSP Response to server's write request.....	210
20.11.	+ELMEXECUTE URC indicate server sending a excute request.....	210
20.12.	AT+ELMEXECUTERSP Response to server's execute request.....	211
20.13.	+ELMOBSERVEREQ URC indicate server sending a observe request	211
20.14.	AT+ELMOBSERVERSP Response to server's observe request	212

20.15.	AT+ELMSEND Send data to server	213
20.16.	AT+ELMSTATUS Query the LwM2M instance status	214
20.17.	+ELMEVT URC of information notification	215
20.18.	Create LwM2M connection example	216
21.	TLS NETWORK COMMANDS	218
21.1.	AT+TLSCFG Configuration TLS Context Parameters	218
21.2.	AT+TLSOPEN Establish TLS connection.....	220
21.3.	AT+TLSCLOSE Close TLS connection	221
21.4.	AT+TLSSSEND Send data in ASCII code	221
21.5.	AT+TLSSSENDRAW Send original data	222
21.6.	AT+TLSRECV URC of Receive data.....	223
21.7.	AT+TLSSTAT Get TLS connection state.....	223
21.8.	+TLSSTAT URC of TLS connection state	224
21.9.	TLS connection example.....	224
22.	NETWORK FRIENDLY MODE	225
22.1.	AT+NFM NETWORK Friendly Mode Enable/disable	225
22.2.	AT+NFM NETWORK Friendly Mode Enable/disable	225
23.	LIST OF ACRONYMS	226
24.	DOCUMENT HISTORY	227

1. INTRODUCTION

1.1. Scope

This document is aimed at providing a detailed specification and a comprehensive listing as a reference for the whole set of AT commands.

1.2. Audience

Readers of this document should be familiar with Telit modules and their ease of controlling by means of AT Commands.

1.3. Contact Information, Support

For general contact, technical support services, technical questions and report documentation errors contact Telit Technical Support at:

- TS-EMEA@telit.com
- TS-AMERICAS@telit.com
- TS-APAC@telit.com
- TS-SRD@telit.com

Alternatively, use:

<http://www.telit.com/support>

For detailed information about where you can buy the Telit modules or for recommendations on accessories and components visit:

<http://www.telit.com>

Our aim is to make this guide as helpful as possible. Keep us informed of your comments and suggestions for improvements.

Telit appreciates feedback from the users of our information.

1.4. Text Conventions



SET section – This section provides all information related to SET functionality of involved AT command. If it has got strictly and relevant SET information, these are located at section end.



READ section – This section provides all information related to READ functionality of involved AT command. If it has got strictly and relevant READ information, these are located at section end.



TEST section – This section provides all information related to TEST functionality of involved AT command. If it has got strictly and relevant TEST information, these are located at section end.



Additional info – This section provides any kind of additional and useful information related to the AT command section as well as command exceptions or special behavior cases.



REFERENCE section – This section provides useful references (standards or normative) related to involved AT command.



EXAMPLE section – This section provides useful examples related to involved AT command.



NOTE section – This section provides all information related to involved AT commands. Each note can provide a different level of information: danger, caution/warning and tip/information.



Danger – This information **MUST** be followed or catastrophic equipment failure or bodily injury may occur.



Caution or Warning – Alerts the user to important points about integrating the module, if these points are not followed, the module and end user equipment may fail or malfunction.



Tip or Information – Provides advice and suggestions that may be useful when integrating the module.

All dates are in ISO 8601 format, i.e. YYYY-MM-DD.

2. AT COMMANDS

The Telit wireless module family can be controlled via the serial interface using the standard AT commands¹. The Telit wireless module family is compliant with:

1. Hayes standard AT command set, to maintain the compatibility with existing SW programs.
2. 3GPP TS 27.007 specific AT command and GPRS specific commands.
3. 3GPP TS 27.005 specific AT commands for SMS (Short Message Service) and CBS (Cell Broadcast Service)

Moreover, Telit wireless module family supports also Telit proprietary AT commands for special purposes.

The following is a description of how to use the AT commands with the Telit wireless module family.

2.1. Definitions

The following syntactical definitions apply:

- <CR>** Carriage return character is the command line and result code terminator character.
- <LF>** Linefeed character is the character recognized as line feed character.
- <...>** Name enclosed in angle brackets is a syntactical element. They do not appear in the command line.
- [...]** Optional subparameter of a command or an optional part of TA information response is enclosed in square brackets. Brackets themselves do not appear in the command line.

2.2. AT Command Syntax

The AT command format supported by Telit module is following the 3GPP related standards.

There are two types of extended command:

- Parameter type commands. This type of commands may be:
 - Set command, AT+CMD=[<value>]<CR>, to store a value or values for later use
 - Read command, AT+CMD?<CR>, to determine the current value or values stored
 - Test command, AT+CMD=?<CR>, to give information about the type of its subparameters
- Action type commands. Execution command, This type of command may be “executed” or “tested”

¹ The AT is an ATTENTION command and is used as a prefix to other parameters in a string. The AT command combined with other parameters can be set up in the communications package or typed in manually as a command line instruction combined with other parameters can be set up in the communications package or typed in manually as a command line instruction.

If all the subparameters of a parameter type command +CMD are optional, issuing AT+CMD=<CR> causes the OK result code to be returned and the previous values of the omitted subparameters to be retained.

**NOTE:**

The command line buffer accepts a maximum of 400 characters. If this number is exceeded none of the commands will be executed and TA returns **ERROR**.

2.2.1. Command Lines

A command line is made up of three elements: the **prefix**, the **body** and the **termination character**. The **command line prefix** consists of the characters “AT” or “at”, or, to repeat the execution of the previous command line, the characters “A/” or “a/”. The **termination character** may be selected by a user option (parameter S3), the default being <CR>.

The basic structures of the command line are:

- AT command without parameter, **AT** is the command line prefix, **CMD1** is the body of a **basic command** and <CR> is the command line terminator character, format: AT[+ |&]<command>
Example: AT+CGMR
- Read command for checking current sub-parameter values, Format: AT[+ |&]<command>?
Example: AT+CMEE?
- Test command for checking possible sub-parameter values, Format: AT[+ |&]<command>=?
Example: AT+CGDCONT=?
- AT command with parameter, Format: AT[+ |&]<command>=<par1>,<par2>,<par3>...

The response of this command is:

- <CR><LF><response><CR><LF>
- <CR><LF><OK/ERROR>[ERROR information]<CR><LF>



NOTE: The command line buffer accepts a maximum of 80 characters. If this number is exceeded none of the commands will be executed and TA returns **ERROR**.

2.2.2. Information Responses and Result Codes

The TA response, in case of verbose response format enabled, for the previous examples command line could be as shown below:

Information response to +CMD1? <CR><LF>+CMD1:2,1,10<CR><LF>

Information response to +CMD1=? <CR><LF>+CMD1: (0-2),(0,1),(0-15)<CR><LF>

Final result code <CR><LF>OK<CR><LF>

Moreover there are other two types of result codes:

- result codes that inform about progress of TA operation (e.g. connection establishment CONNECT)
- result codes that indicate occurrence of an event not directly associated with issuance of a command from TE

2.2.3. Command Response Time-out

Every command issued to the Telit modules returns a result response if response codes are enabled (default). The time needed to process the given command and return the response varies, depending on the command type. Commands that do not interact with the SIM or the network, and involve only internal set up settings or readings, have an immediate response, depending on SIM configuration(e.g., number of contacts stored in the phonebook, number of stored SMS), or on the network the command may interact with.

In the table below are listed only the commands whose interaction with the SIM or the network could lead to long response timings. When not otherwise specified, timing is referred to set command.

Command	Estimated maximum time to get response (Seconds)
+COPS	180 (For test command)
+CLCK	180
+CPWD	180
+CPIN	5
+CRSM	5

2.2.4. Command Issuing Timing

The chain Command -> Response shall always be respected and a new command must not be issued before the module has terminated all the sending of its response result code (whatever it may be).

This applies especially to applications that “sense” the OK text and therefore may send the next command before the complete code <CR><LF>OK<CR><LF> is sent by the module. It is advisable anyway to wait for at least 20ms between the end of the reception of the response and the issue of the next AT command. If the response codes are disabled and therefore the module does not report any response to the command, then at least the 20ms pause time shall be respected.

During command mode, due to hardware limitations, under severe CPU load the serial port can lose some characters if placed in auto-baud at high speeds. Therefore, if you encounter this problem fix the baud rate with +IPR command.

3. IDENTIFICATION COMMANDS

3.1. AT+CGMI Request manufacturer identification

This command is used to Request manufacturer identification



AT+CGMI

Response

<CR><LF><manufacturer><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<manufacturer>: module manufacturer



AT+CGMI=?

Response

<CR><LF>OK<CR><LF>



AT+CGMI

TELIT

OK

3.2. AT+CGMM Request model identification

This command causes the DCE to transmit one or more lines of information text, determined by the manufacturer, which is intended to permit the user of the DCE to identify the specific model of device.



AT+CGMM

Response

<CR><LF><model><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<model>: module ID



AT+CGMM=?

Response

<CR><LF>OK<CR><LF>



AT+CGMM

NE310H2-W1

OK

3.3. AT+CGMR Request revision identification

Execution command returns the device model identification code without command echo



AT+CGMR
Response

<CR><LF><revision><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<revision>: firmware version



AT+CGMR=?
Response

<CR><LF>OK<CR><LF>



AT+CGMR
M0J.00000-B001

OK

3.4. AT+CGSN Request IMEI

This command is used to request IMEI/SN/SVN



AT+CGSN=<sn>

<CR><LF><sn><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<sn> it is SN in default

0	SN
1	IMEI
2	IMEISV
3	SVN



AT+CGSN=?

<CR><LF>list <sn><CR><LF>

<CR><LF>OK<CR><LF>



```

AT+CGSN
+CGSN: 1234567

OK
AT+CGSN=0
1234567

OK
AT+CGSN=1
+CGSN: 123456789012396

OK
AT+CGSN=2
+CGSN: 1234567890123901

OK
AT+CGSN=3
+CGSN: 01

OK

```

3.5. AT+CIMI Request international mobile subscriber identity

This command is used to request IMSI.



```

AT+CIMI
Response
<CR><LF><IMSI><CR><LF>
<CR><LF>OK<CR><LF>

or
<CR><LF>+CME ERROR: <err><CR><LF>

```

Parameter
 <IMSI>: International Mobile Subscriber Identity (string without double quotes)



```

AT+CIMI=?
Response
<CR><LF>OK<CR><LF>

```



```

AT+CIMI
010011234567890

OK

```

3.6. *MATREADY

This unsolicited result code is generated on any new AT channel that is enabled. It is not configurable and will always be generated when a channel is enabled. This URC is not generated on channels re-enabled after wakeup from deep sleep.



unsolicited result code
 *MATREADY: <ready_status><CR><LF>
Parameter
 <ready_status>
 1: AT channel ready (only valid value)



When module power on and AT channel enabled
 *MATREADY: 1
 +CFUN: 1
 +CPIN: READY

3.7. AT+GMI Request manufacturer identification



3GPP TS 27.007



Execute command
 AT+GMI
 Response
 • TA returns manufacturer identification text.
 • <manufacturer>
 OK
 Parameters
 • <manufacturer>



Test command
 AT+GMI=?
 Response
 • OK

3.8. AT+GMM Request model identification



3GPP TS 27.007



Execute command
AT+GMM
Response
• TA returns product model identification text
• <model> OK
Parameters
• <model>



Test command
AT+GMM=?
Response
OK

3.9. AT+GMR Request revision identification



3GPP TS 27.007



Execute command
AT+CGMR
Response
• TA returns product software version identification text
• <revision> OK
Parameters
• <revision>



Test command
AT+GMR=?
Response
OK

3.10. AT+GOI/+CGIO Request global object identification



ITU-T Recommendation V.25 ter



Execute command

AT+GOI/+CGIO

Response

- TA reports one or more lines of information text that permit the user to identify the device, based on the ISO system for registering unique object identifiers.

Parameter

- <Object Id> identifier of device type

Scope

Channel specific (response output only on channel which entered the command)



Test command

AT+GOI/+CGIO=?

Response

- OK

3.11. AT+GSN Request TA serial number identification (IMEI)



ITU-T Recommendation V.25 ter



Execute command

AT+GSN

Response

- TA reports the IMEI (International Mobile Equipment Identifier) number in information text that permits the user to identify the individual ME device.

- <sn>

OK

Parameter

- <sn> IMEI of the telephone (International Mobile station Equipment Identity)

Scope

Channel specific (response output only on channel which entered the command)



Test command

AT+GSN=?

Response

OK

3.12. AT+ICF Set TE-TA control character framing



ITU-T Recommendation V.25 ter



Set command

AT+ICF=[<format>,<parity>]

Response

- This parameter setting determines the serial interface character framing format and parity received by TA from TE.
- Note: +IPR=0 forces +ICF=0
- OK

Parameter

Note: The parity field is ignored if the format field specifies no parity.

- <format>
 - 1 8 data 0 parity 2 stop
 - 2 8 data 1 parity 1 stop
 - 3 8 data 0 parity 1 stop
 - 4 7 data 0 parity 2 stop
 - 5 7 data 1 parity 1 stop
 - 6 7 data 0 parity 1 stop
- <parity>
 - 0 odd
 - 1 even
 - 2 mark (1)
 - 3 space (0)

Scope

Channel Specific



Test command

AT+ICF=?

Response

- +ICF:(list of supported <format>s), (list of supported <parity>s) OK Parameter
- see set command



Read command

AT+ICF?

Response

- +ICF: <format>,<parity> OK
 - o Note: This framing is applied for command state
- Parameter
- see set command

3.13. 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>.



3GPP TS 27.007



Execute command

AT+CLCK = <fac>,<mode>[,<passwd>[,<class>]]

Response

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>.

- If <mode><>2 and command is successful
OK
- If <mode>=2 and command is successful
+CLCK: <status>[,<class1>[<CR><LF>
+CLCK: <status>, class2...]] OK
- If error is related to ME functionality:
+CME ERROR: <err>

Parameters

<fac> "SC" SIM (lock SIM card) (SIM asks password in ME power-up and when this lock command issued)

Proprietary additions:

<mode> 0 unlock

1 lock

2 query status

<passwd> password

<class> Field not required for NB-IOT, so will be ignored

<status>

0 off

1 on



Test command

AT+CLCK=?

Response

- +CLCK: (list of supported <fac>s) OK

Parameters

- see execute command

3.14. AT+CPWD Change password



3GPP TS 27.007



Execute command

AT+CPWD = <fac>,[<oldpwd>],<newpwd>

Response

TA sets a new password for the facility lock function.

- OK
- If error is related to ME functionality:

+CME ERROR: <err>

Parameters

- <fac>
- "SC" SIM (lock SIM card) (SIM asks password in ME power-up and when this lock command issued)
- <oldpwd> old password
- <newpwd> new password



Test command

AT+CPWD=?

Response

TA returns a list of pairs that present the available facilities and the maximum length of their password.

- +CPWD: list of supported (<fac>, <pwdlength>)s OK
- If error is related to ME functionality:

+CME ERROR: <err>

Parameters

- <fac> see execute command
- <pwdlength> integer maximum length of password

3.15. AT+CREG Network registration



3GPP TS 27.007



Set command

AT+CREG=[<n>]

Response

- UE controls the presentation of an unsolicited result code +CREG: <stat> when <n>=1 and there is a change in the ME network registration status.

- OK

Parameters

<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>[,<AcT>]]

<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

6 registered for "SMS only", home network (applicable only when <AcT> indicates NB-IOT

7 registered for "SMS only", roaming (applicable only when <AcT> indicates NB-IOT

<lac> string type; two byte location area code or tracking area code in hexadecimal format (e.g."00C3" equals 195 in decimal)

<ci> string type; four byte UTRAN/GERAN/E-UTRAN cell ID in hexadecimal format

<AcT> access technology of the registered network

9 NB-IoT

Unsolicited result code

- When <n>=1 and there is a change in the ME network registration status: +CREG: <stat>

- When <n>=2 and there is a change in the ME cell status:

+CREG: <stat>[,<lac>,<ci>[,<AcT>]]

Parameters

- see set command



Note

- For NB-IoT product, only <AcT> value of 9 is valid



Test command

AT+CREG=?

Response

- +CREG: list of supported <n>s OK

Parameters

- see set command



Read command

AT+CREG?

Response

UE 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>, <ci> and <AcT> are returned only when <n>=2 and ME is registered in the network.

- When <n> = 0 or 1:
+CREG: <n>,<stat> OK
 - When <n> = 2:
+CREG: <n>,<stat>[,<lac>,<ci>[,<AcT>]] OK
 - If error is related to ME functionality:
+CME ERROR: <err>
- Parameters
- see set command

3.16. AT+CRSM Restricted SIM Access



3GPP TS 27.007



Set command

AT+CRSM=<command>[,<fileid>[,<P1>,<P2>,<P3>[,<data>[,<pathid>]]]]

Response

- +CRSM: <sw1>,<sw2>[,<response>]
- +CME ERROR: <err>

Parameters

- <command> integer type: 3GPP 102.221 SIM command sent from the ME to the SIM
 - 176 READ BINARY
 - 178 READ RECORD
 - 192 GET RESPONSE
 - 214 UPDATE BINARY
 - 220 UPDATE RECORD
 - 242 STATUS
- <fileid> integer type; this is the identifier of an elementary datafile on SIM. Mandatory for every command except STATUS.
- <P1>,<P2>,<P3> integer type; parameters passed on by the ME to the SIM. These parameters are mandatory for every command, except GET RESPONSE and STATUS. The values are described in 3GPP 102.221
- <data> information which shall be written to the SIM (hexadecimal character format; refer +CSCS)
- <sw1>,<sw2> integer type; information from the SIM about the execution of the actual command. These parameters are delivered to the TE in both cases, on successful or failed execution of the command
- <response> response of a successful completion of the command previously issued (hexadecimal character format; refer +CSCS). STATUS and GET RESPONSE return data, which gives information about the current elementary data field. This information includes the type of file and its size (refer 3GPP 102.221). After READ BINARY or READ RECORD command the requested data will be returned.
<response> is not returned after a successful UPDATE BINARY or UPDATE RECORD command
- <pathid> String type; contains the path of an elementary file on the SIM/UICC in hexadecimal format as defined in ETSI TS 102.211 (e.g. "7F205F70" in SIM and UICC case). The <pathid> only used in the mode "select path from MF" as defined in ETSI TS 102.211.
<pathid> is only valid for USIMs



Test command
AT+CRSM=?
Response
• OK

3.17. AT+CSCS Select TE Character Set



3GPP TS 27.007



Set command
AT+CSCS=[<chset>]
Response
• Sets which character set <chset> is used by the TE. The TA can then convert character strings correctly between the TE and ME character sets.
Parameter
• <chset>
“GSM” GSM default alphabet.
“HEX” Hexadecimal numbers in character strings
“IRA” International reference alphabet (ITU-T T.50)
“PCCP” PC character set Code Page
“PCDN” PC Danish/Norwegian character set
“UCS2” UCS2 alphabet
“8859-1” ISO 8859 Latin (1) character set



Test command
AT+CSCS=?
Response
• +CSCS: (list of supported <chest>s)



Read command
AT+CSCS?
Response
• +CSCS: <chset> OK
Parameters
• see Set command

3.18. AT+CNUM Subscriber Number



3GPP TS 27.007



Test command
AT+CNUM=?
Response
<CR><LF>OK<CR><LF>

3.19. AT+CPOL Preferred operator list



3GPP TS 27.007



Set command
AT+CPOL=[<index>][,<format>[,<oper>[,<GSM_AcT>,<GSM_Compact_AcT>,<UTRAN_AcT>,<E-UTRAN_AcT>]]]
Response
• +CME ERROR: <err>
Parameters
<index> integer type: order number of operator in USIM preferred operator list
<format> 0 long format alphanumeric <oper>
1 short format alphanumeric <oper>
2 numeric <oper>
<opern> string type: <format> indicates whether alphanumeric or numeric format used (see +COPS command)

<GSM_AcTn> GSM Access technology;
0 access technology not selected
1 access technology selected

<GSM_Comp_AcTn> GSM compact Access technology;
0 access technology not selected
1 access technology selected

<UTRAN_AcTn> UTRA Access technology;
0 access technology not selected
1 access technology selected

<E-UTRAN_AcTn> E-UTRAN Access technology;
0 access technology not selected
1 access technology selected



Note

- Not all USIMs support the preferred operator list.



Test command
AT+CPOL=?
Response
• +CPOL: (list of supported <index>s),(list of supported <format>s)
Parameter
• See set command



Read command

AT+CPOL?

Response

- +CPOL: <index1>,<format>,<oper1>[,<GSM_Act1>,<GSMcomp_Act1>,<UTRAN_Act1>,<E-UTRAN_Act1>]
- [<CR><LF>+CPOL: <index2>,<format>,<oper2>[,<GSM_Act2>,<GSMcomp_Act2>,<UTRAN_Act2>,<E-UTRAN_Act2>]
- [...]]
- +CME ERROR: <err>
- Parameter
- See set command

3.20. AT+CSIM Generic SIM Access



3GPP TS 27.007



Set command

AT+CSIM=<length>,<command>

Response

- +CSIM: <length>,<response>
- +CME ERROR: <err>

Parameters

- <length> integer type: length of characters sent to the TE in <command> or <response> (i.e. twice the number of octets in the raw data)
- <command> string type: hex format: 3GPP 102.221 SIM command sent from the ME to the SIM
- <response> string type: hex format: 3GPP 102.221 response from SIM to <command>



Test command

AT+CSIM=?

Response

- OK

3.21. AT+CTZR Time Zone Reporting



3GPP TS 27.007



Set command

AT+CTZR=<onoff>

Response

<CR><LF>OK<CR><LF>

- If error is related to ME functionality:

<CR><LF>+CME ERROR: <err><CR><LF>

This set command enables and disables the time zone change event reporting. If the reporting is enabled the MT returns the unsolicited result code +CTZV: <tz> whenever the time zone is changed.

Parameter

- <onoff>

0 Disable time zone change event reporting (default value)

1 Enable time zone change event reporting by unsolicited result code +CTZV: <tz>

2 Enable extended time zone and local time reporting by unsolicited result code

+CTZE: <tz>,<dst>,[<time>].

<tz> String type. The sum of the local time zone (difference between the local time and GMT is expressed in quarters of an hour) plus daylight saving time. The format is “±zz”, expressed as a fixed width, two-digit integer with the range -48 ... +56. To maintain a fixed width, numbers in the range -9 ... +9 are expressed with a leading zero, e.g. “-09”, “+00” and “+09”.

<dst> Integer type. Indicates whether <tz> includes daylight savings adjustment

0 <tz> includes no adjustment for daylight saving time

1 <tz> includes +1 hour (equals 4 quarters in <tz>) adjustment for daylight saving time

2 <tz> includes +2 hours (equals 8 quarters in <tz>) adjustment for daylight saving time

<time> String type. The local time. The format is “YYYY/MM/DD, hh:mm:ss”, expressed as integers representing year (YYYY), month (MM), date (DD), hour (hh), minute (mm) and second (ss). This parameter can be provided by the network when delivering time zone information and will be presented in the unsolicited result code of extended time zone reporting if provided by the network. The format of “MM”, “hh” are variable width, numbers in the range 1~9 are expressed with 1~9.



Test command

AT+CTZR=?

Response

<CR><LF>+CTZR: (list of supported <onoff>s) <CR><LF>

<CR><LF>OK<CR><LF>

- If error is related to ME functionality:

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

- See set command



Read command

AT+CTZR?

Response

<CR><LF>+CTZR: <onoff><CR><LF>

<CR><LF>OK<CR><LF>

- If error is related to ME functionality:

<CR><LF>+CME ERROR: <err><CR><LF>

Parameters → See set command

4. GENERAL COMMANDS

4.1. ATE Command Echo

The setting of this parameter enables/disables the command echo.



AT E<val>
Response

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<val>:

0: DCE disables command echo

1: DCE enables command echo , hence command sent to the device are echoed back to the DTE. . (factory default)



ATE0

OK

//Input AT, but the device disable command echo

OK

4.2. AT&D Set circuit Data Terminal Ready(DTR) function mod

Set circuit Data Terminal Ready(DTR) function mode.this parameter detremines how the TA responds when circuit 108/2(DTR) is changed from the ON to the OFF condition during data mode.



AT&D[<value>]

Response

<CR><LF>OK<CR><LF>

Parameter

<value>:

0: TA ignores status on DTR.

1:ON->OFF on DTR:Change to command mode with call remaining connected

2:OFF->ON on DTR:Disconnect call,change to command mode,During state DTR=OFF is auto-answer off.

4.3. ATV DCE Response Format

The setting of this parameter determines the contents of the header and trailer transmitted with result codes and information responses. It also determines whether result codes are transmitted in a numeric form or an alphabetic (or "verbose") form. The text portion of information responses is not affected by this setting.



ATV<val>

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<val>:

0: DCE transmits limited headers and trailers and numeric text.

1: DCE transmits full headers and trailers and verbose response text.

4.4. AT+CFUN Set Phone Functionality

Set command selects the level of functionality <fun> in the MT.



AT+CFUN=[<fun>[,<rst>]]

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<fun>:

0: minimum functionality

1: full functionality (factory default)

4: disable phone both transmit and receive RF circuits

7: Disable SIM card; transmit and receive circuits are valid

Other value reserved

NOTE: issuing AT+CFUN=4[,0] actually causes the module to perform either a network deregistration and a SIM deactivation.

<rst>:

0: no need to reset the module to make <fun> effect.

1: need to reset the module to make <fun> effect.

2: reserved

NOTE: This shall be always default 0 when <rst> is not given

unsolicited result code When module power on and the module will report:
+CFUN: <fun><CR><LF>



AT+CFUN=?

Response

<CR><LF>+CFUN: (list of supported <fun>s), (list of supported <rst>s)<CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>



AT+CFUN?
Response

<CR><LF>+CFUN: <fun><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>



AT+CFUN?
+CFUN: 1

OK

AT+CFUN=?

+CFUN: (0,1,4,7),(0-2)

OK

4.5. AT+CMEE Report Mobile Termination Error

Set command disables or enables the use of result code +CME ERROR: <err> as an indication of an error relating to the functionality of the MT. When enabled, MT related errors cause +CME ERROR: <err>final result code instead of the regular ERROR final result code. ERROR is returned normally when error is related to syntax, invalid parameters, or TA functionality.

Read command returns the current value of subparameter <n>.

Test command returns values supported as a compound value.



AT+CMEE=[<n>]

Response

<CR><LF>OK<CR><LF>

Parameter

<n>:

0: disable +CME ERROR: <err> result code and use ERROR instead

1: enable +CME ERROR: <err> result code and use numeric <err> values

2: enable +CME ERROR: <err> result code and use verbose <err>values



AT+CMEE=?

Response

<CR><LF>+CMEE: (list of supported <n>s)<CR><LF>

<CR><LF>OK<CR><LF>



AT+CMEE?

Response

<CR><LF>+CMEE: <n><CR><LF>

<CR><LF>OK<CR><LF>



AT+CMEE=0

//+CME ERROR shall not be used

OK

AT+CGSN

ERROR

AT+CMEE=1

//use numeric <err>

```

OK
AT+CGSN

+CME ERROR: 23
AT+CMEE=2 //use verbose <err>
OK
AT+CGSN

+CME ERROR: memory failure

```

4.6. +CME ERROR ME Error Result Code

This is NOT a command, it is the error response to +Cxxx 3gpp TS 27.007 commands.



+CME ERROR ME Error Result Code
 <CR><LF>+CME ERROR: <err><CR><LF>
 Parameter
 <err> - error code can be either numeric or verbose (see +CMEE).

The possible values of <err> are reported in the table:

Numeric Format	Verbose Format
General error:	
0	Phone failure
1	No connection to phone
2	phone-adaptor link reserved
3	Operation not allowed
4	Operation not allowed
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 require
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 timer-out
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
48	Hidden key required (NOTE: This key is required when accessing hidden phonebook)
49	EAP method not supported
50	Incorrect parameters
100	unknown
GPRS-related errors	
Errors related to a failure to perform an Attach	
103	Illegal MS (#3)
106	Illegal ME (#6)
107	GPRS services not allowed (#7)
111	PLMN not allowed (#11)
112	Location area not allowed (#12)
113	Roaming not allowed in this location area (#13)
Errors related to a failure to Activate a Context	
132	service option not supported (#32)
133	requested service option not subscribed (#33)
134	service option temporarily out of order (#34)
149	PDP authentication failure
Other GPRS errors	
150	invalid mobile class

148	unspecified GPRS error
	VBS / VGCS and eMLPP -related errors
151	VBS/VGCS not supported by the network
152	No service subscription on SIM
153	No subscription for group ID
154	Group Id not activated on SIM
155	No matching notification
156	VBS/VGCS call already present
157	Congestion
158	Network failure
159	Uplink busy
160	No access rights for SIM file
161	No subscription for priority
162	operation not applicable or not possible

4.7. AT+CEER Extended Error Report

Reports extended error related to the last unsuccessful call



SIM Presence	Setting saved	Can be aborted	MAX timeout
Not required	No	No	-



AT+CEER

Execution command returns one or more lines of information text <report> offering the TA user an extended error report, in the format:

+CEER: <report>

This report regards some error condition that may occur:

- the failure in the last unsuccessful call setup (originating or answering)
- the last call release

If no error condition has occurred since power up, then "Normal, unspecified" condition is reported



AT+CEER=?

Test command returns OK result code.

4.8. ATI Display product identification information

ATI Display product identification information



ITU-T Recommendation V.25 ter

SIM Presence	Setting saved	Can be aborted	MAX timeout
Not required	No	No	-



ATI

Execution command ATI Display product identification information

Response

TA issues product information text

Example:

Telit

NE310H2-W1

OK

4.9. ATQ Set result code presentation mode



ITU-T Recommendation V.25 ter

SIM Presence	Setting saved	Can be aborted	MAX timeout
Not required			-



Set command

ATQ[<n>]

Response

This parameter setting determines whether the TA transmits any result code to the TE. Information text transmitted in response is not affected by this setting.

If <n>=0:

OK

If <n>=1:

(none)

Parameter

<n>

0 TA transmits result code

1 Result codes are suppressed and not transmitted

4.10. ATN1 initial setting to handshake at highest speed

Some PC modem driver initial setting to handshake at highest speed larger than S37



Set command

ATN1

Response

Some standard PC modem drivers will send this AT command to initialize the setting, but it is meaningless in the 3GPP standard. Return OK and no effect for the setting.

OK

4.11. ATSO Set number of rings before automatically answering the call



Set command

ATSO=<n>

Response

This parameter setting determines the number of rings before auto-answer. This command has no effect for the setting.

OK

Parameter

<n> 0 automatic answering is disabled

1-255 enable automatic answering on the ring number specified



Read command

ATSO?

Response

<n>

OK

4.12. ATS1 Ring Counter



Set command

ATS1=<n>

Response

This command will not alter the RING counter, but simply display. This command has no effect for the setting.

OK

Parameter

<n>

0-255 The number of "RING" strings sent to the TE as a result of receiving an incoming call.

4.13. ATS2 Set escape sequence character

ATS2 Set escape sequence character



ITU-T Recommendation V.25 ter

SIM Presence	Setting saved	Can be aborted	MAX timeout
Not required			-



Set command

ATS2=<n>

Response

- This parameter setting determines the character recognized by the TA to indicate the escape sequence.

- OK

Parameter

<n> 0-43-255 escape sequence character

Note: default 43 = '+'



Read command

ATS2?

Response

- <n> OK

4.14. ATS3 Set command line termination character

ATS3 Set command line termination character



ITU-T Recommendation V.25 ter

SIM Presence	Setting saved	Can be aborted	MAX timeout
Not required			-



Set command

ATS3=<n>

Response

- This parameter setting determines the character recognized by the TA to terminate an incoming command line. The TA also returns this character in output.

- OK

Parameter

- <n> 0-13-127 command line termination character

Note: default 13 = CR



Read command

ATS3?

Response

- <n> OK

4.15. ATS4 Set response formatting character

ATS4 Set response formatting character



ITU-T Recommendation V.25 ter

SIM Presence	Setting saved	Can be aborted	MAX timeout
Not required			-



Set command

ATS4=[<n>]

Response

This parameter setting determines the character generated by the TA for result code and information text.

OK

Parameter

- <n> 0-10-127 response formatting character

Note: default 10 = LF



Read command

ATS4?

Response

- <n> OK

4.16. AT5 Set command line editing character

AT5 Set command line editing character



ITU-T Recommendation V.25 ter

SIM Presence	Setting saved	Can be aborted	MAX timeout
Not required			-



Set command

AT5=[<n>]

Response

- This parameter setting determines the character recognized by TA as a request to delete from the command line the immediately preceding character.

- OK

Parameter

- <n> 0-8-127 command line editing character

Note: default 8 = Backspace



Read command

AT5?

Response

- <n> OK

4.17. AT57 Set number of seconds to wait for connection completion



Set command

AT57=<n>

Response

This parameter setting determines the amount of time to wait for the connection completion in case of answering or originating a call. This command has no effect for the setting.

OK

Parameter

<n> 1-255 number of seconds to wait for connection completion. default value is 255.

4.18. AT S10 Set disconnect delay after indicating the absence of data carrier

AT S10 Set disconnect delay after indicating the absence of data carrier



Set command

AT S10=<n>

Response

This parameter specifies the amount of time, in tenths of a second, that the DCE will remain connected to the line (off-hook) after the DCE has indicated the absence of received line signal. This command has no effect for the setting.

OK

Parameter

<n> 1-254 Number of tenths of a second of delay

4.19. AT S12 Set Escape Code Guard Time

This command sets the escape code guard time in fiftieths of a second. The escape guard time is used to measure when to detect the +++ escape sequence has been entered by the PC in order to drop out of data mode back to AT command mode.

The guard time determines the time that forms a guard period before and after three escape sequence characters. In order to distinguish an escape sequence from just three escape sequence characters in the data stream there is timing associated to the three escape sequence characters of an escape sequence.

The time between the last byte of the data stream and the first escape sequence character must be at least the guard time and the time between each escape sequence character of the escape sequence must be less than the guard time and no other byte is received after the third escape sequence character for the time of the guard time. If an escape sequence is detected, the OK result code will be sent to the DTE. Otherwise, the DCE will stay in data mode.



ITU-T 3GPP TS 27.007

Note

- Set Escape Code Guard Time command

For example: "<Guard time>+++<Guard time>"



Execution command

AT S12=<n>

Response

- OK

If error is related to wrong AT syntax:

- +CME ERROR: <err>

Parameters

- <n> Numeric value of the escape guard time value in 1/50 seconds:
 - o 000-255 Number of 20 ms. Default is 050 (1 sec).



Test Command AT+S12=?

Response

- ERROR
-



Read command

AT+S12?

Response

- <n>
 - OK
 - NB: <n> is in 3 decimal digits format (e.g. Default value is given as 050). If error is related to wrong AT syntax:
 - +CME ERROR: <err>
-

AT+S12=?

ERROR



AT+S12=100

OK

AT+S12?

100

OK

4.20. AT25 Set DTR change time

This command sets the S-register 25 Detect DTR change time that contain the threshold for noticing a change in DTR. This time permits to the modem to ignore DTR before taking action specified by &Dn (See AT&D Circuit 108 behavior).

The value unit is in 1/100 seconds. Default value is set to 5 (50ms delay after a DTR drop before the modem acts on it).



ITU-T 3GPP TS 27.007

Note

Set DTR change Time command



Execution command

AT25=<n>

Response

- OK
- If error is related to wrong AT syntax:

+CME ERROR: <err>

Parameters • <n> Numeric value of DTR delay in 10 milliseconds:

000-255 Number of 10 ms. Default is 005.



Test Command AT25=?

Response

- ERROR



Read command

AT25?

Response

- <n>
- OK
- <n> is in 3 decimal digits format (e.g. Default value is given as 000)
- If error is related to wrong AT syntax:

+CME ERROR: <err>

4.21. AT95 enable extended result codes

AT95 Some PC modem driver initial setting to enable extended result codes



Execution command

AT95=<n>

Response

Some standard PC modem drivers will send this AT command to initialize the setting, but it is meaningless in the 3gpp standard. So we just return OK and no effect for the setting.

<CR><LF>OK<CR><LF>

Parameters

- <n>
0-255 meaningless for the GSM, and GPRS/Packet Domain setting.



Read command

AT95?

Response

<CR><LF>OK<CR><LF>

4.22. ATX Set CONNECT result code format and call monitoring



Set command

ATX[<value>]

Response

This parameter setting determines whether or not the TA detected the presence of dial tone and busy signal and whether or not TA transmits particular result codes

<CR><LF>OK <CR><LF>

Parameter

<value>

- 0 CONNECT result code only returned, dial tone and busy detection are both disabled
- 1 CONNECT<text> result code only returned, dial tone and busy detection are both disabled
- 2 CONNECT<text> result code returned, dial tone detection is enabled, busy detection is disabled
- 3 CONNECT<text> result code returned, dial tone detection is disabled, busy detection is enabled
- 4 CONNECT<text> result code returned, dial tone and busy detection are both enabled

4.23. AT+FCLASS Set fax class



Set command
AT+FCLASS=<n>
Response
This command has no effect for the setting, and is supported for compatibility reasons.
<CR><LF>OK <CR><LF>
Parameter
<n>
0 data
1 fax class 1



AT+FCLASS=?
Response
+FCLASS: list of supported <n>s
OK



AT+FCLASS=?
Response
+FCLASS: <n>
OK

4.24. AT+GCAP Request complete TA capabilities list



Set command
AT+GCAP

Response
TA reports a list of additional capabilities.
+GCAP: <name>s
OK

Parameter
<name> example, +CGSM

4.25. AT&F Set all current parameters to manufacturer defaults



ITU-T Recommendation V.25 ter



Set command
 AT&F[value]
 Response
 TA sets all current parameters to the manufacturer defined profile.
 <CR><LF>OK<CR><LF>
 Parameter
 <value> 0 set all TA parameters to manufacturer defaults

4.26. AT&K Flow control setting



ITU-T Recommendation V.25 ter

Note: This command does not store anything in the profile data because it sets the AT+IFC settings when used:

AT&K0 is equivalent of entering AT+IFC=0,0
 AT&K3 is equivalent of entering AT+IFC=2,2
 AT&K4 is equivalent of entering AT+IFC=1,1



Set command
 AT&K[<value>]
 Response
 <CR><LF>OK <CR><LF>
 Parameter
 <value>
 0 No flow control
 3 RTS /CTS flow control (hardware)
 4 XON/XOFF flow control (software)
 Scope
 For S/W flow control, this sets on a per channel basis when using the 27.010 MUX. I.e. flow control characters are sent/received within the 27.010 MUX frame as part of the data.
 For hardware flow control the setting will apply to all channels routed through one connection level (e.g. USB, UART).

4.27. AT&V Display current configuration



Set command
 AT&V[<n>]
 Response
 TA returns the current parameter setting.
 <CR><LF><current configurations text><CR><LF>
 <CR><LF>OK<CR><LF>
 Parameter
 <n> 0 profile number

4.28. AT&W Store current parameter to user defined profile



Set command
 AT&W[<n>]
 Response
 TA stores the current parameter setting in the user-defined profile.
 Note: The user-defined profile is stored in non-volatile memory.
 <CR><LF>OK<CR><LF>
 Parameter
 <n> 0 profile number to store to
 Scope
 Channel Specific
 Only one user profile is stored in NVRAM. This command will store the current Generic parameters values and the Channel Specific values for the channel on which the command is received

4.29. AT&C Set circuit Data Carrier Detect (DCD) function mode



Set command
 AT&C[<value>]
 Response
 This parameter determines how the state of circuit 109(DCD) relates to the detection of received line signal from the distant end.
 This command has no effect for the setting.
 <CR><LF>OK<CR><LF>
 Parameter
 <value>
 0 DCD line is always ON
 1 DCD line is ON only in the presence of data carrier
 Scope
 Channel Specific

4.30. AT&D Set circuit Data Terminal Ready (DTR) function mode



Set command
AT&D[<value>]

Response

This parameter determines how the TA responds when circuit 108/2(DTR) is changed from the ON to the OFF condition during data mode.

This command has no effect for the setting.

<CR><LF>OK<CR><LF>

Parameter

<value>

0 TA ignores status on DTR

1 ON->OFF on DTR: Change to command mode with call remaining connected

2 ON->OFF on DTR: Disconnect call, change to command mode. During state DTR = OFF is auto-answer off.

Scope

Channel Specific

5. SERIAL INTERFACE CONTROL COMMANDS

5.1. AT+IPR Set the Baud rate of UART

This command is used to Set the Baud rate of UART. This command will take effect after restart the module and it will be saved after restart.



AT+IPR=<rate>

Response

<CR><LF>OK<CR><LF>

Parameter

<rate>Baud rate, default value is 0, that is enable baud rate auto detectable.



NOTE:

1. supported auto detectable <rate>s are:4800,9600,19200,38400,57600,115200, supported fixed-only<rate>s are: 0,4800,9600,19200,38400,57600,115200,230400,460800,921600.
2. If set the fixed rate, the baud rate auto detection is invalid. If you need baud rate auto detection, you need to restart the module.
3. If <rate>=0, that is enable baud rate auto detect, and the default baud rate is 115200.



AT+IPR=?

Response

<CR><LF>+IPR:(list of supported autodetectable rate values)[,(list of fixed-only rate values)]<CR><LF>

<CR><LF>OK<CR><LF>



AT+IPR?

Response

<CR><LF>+IPR: <rate><CR><LF>

<CR><LF>OK<CR><LF>



AT+IPR?

+IPR: 115200

OK

at+ipr=?

+IPR:

(110,300,1200,2400,4800,9600,19200,38400,57600,115200),(0,9600,19200,38400,57600,115200,230400,460800,921600)

OK

5.2. AT+IFC Local data flow control

This set command is used to control the operation of local flow control between the DTE and DCE during the data state. It accepts two numeric sub-parameters:

<DCE_by_DTE>, which specifies the method to be used by the DTE to control the flow of received data from the DCE;
<DTE_by_DCE>, which specifies the method to be used by the DCE to control the flow of transmitted data from the DTE.



AT+IFC=[<DCE_by_DTE> [,<DTE_by_DCE>]]
Response

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<DCE_by_DTE> specifies the method will be used by TE at receive of data from TA

- 0 No data flow control
- 1 XON/XOFF, don't pass characters on to data stack
- 2 Line 133: Ready for Receiving

<DTE_by_DCE> specifies the method will be used by TA at receive of data from TE

- 0 No data flow control
- 1 XON/XOFF
- 2 line 106: Clear to send(CTS)

NOTE: DC1 is IA5 1/1; DC3 is IA5 1/3.



AT+IFC=?
Response

<CR><LF>+IFC:(list of supported <DCE_by_DTE> values),(list of supported <DTE_by_DCE> values)<CR><LF>

<CR><LF>OK<CR><LF>



AT+IFC?
Response

<CR><LF>+IFC: <DCE_by_DTE>,<DTE_by_DCE><CR><LF>

<CR><LF>OK<CR><LF>



AT+IFC=?
+IFC: (0-2),(0-2)

OK
AT+IFC?
+IFC: 0,0

OK

5.3. AT#TCOMWRT Firmware Upgrade through UART

This command permits to run a Firmware upgrade UART using a Delta binary file. It is usable on MAIN UART port and consists in two phases: File transfer to memory and then upgrade. The command requires the file length as parameter. When run it will present a prompt ">" and a "CONNECT" message. After the CONNECT message it is possible to transfer the binary file. If the file length equal the <file_length>, the module will return OK after the transfer. Ensure to use a correct delta file from the version the module is using tot he new release (Contact Telit Technical support for details)



```
AT#TCOMWRT=<file_len><CR>
><file_data>
```

Response

```
<CR><LF>CONNECT<CR><LF>
```

If update success, it will report:

```
<CR><LF>FOTA SUCCESS<CR><LF>
```

If update failed, it will report:

```
<CR><LF>FOTA FAILED<CR><LF>
```

Parameter

<file_len> integer type, the file data length in byte.

<file_data> The update file /data

FW upgrade after system reset.

Note: 1. The returned ">" and "CONNECT" from module stands for module is ready to receive file, if you send file before "CONNECT", the data will lost and it will cause update failure.



```
AT#TCOMWRT=?
```

Response

```
<CR><LF>OK<CR><LF>
```



```
AT#TCOMWRT=34728
```

```
>
```

```
CONNECT
```

```
OK
```

```
AT+TRST
```

```
OK
```

```
+CPIN: NOT READY
```

```
F1: 0000 0000
```

```
V0: 0000 0000 [0001]
```

```
00: 0006 000C
```

```
01: 0000 0000
```

U0: 0000 0001 [0000]

T0: 0000 00B4

Leaving the BROM

*MATREADY: 1

+CFUN: 1
FOTA SUCCESS
+CPIN: READY

5.4. AT+CMUX Serial Multiplexer control

This command is used to enable CMUX function; there is no need to disable CMUX. When you close CMUX in MCU, the module will exit CMUX automatically.

Note: You need to send `at+eurconf=1` after you enable CMUX, and you need to send `at+eurconf=0` before you disable CMUX function.



AT+CMUX=<mode>[,<subset>[,<port_speed>[,<N1>[,<T1>[,<N2>[,<T2>[,<T3>[,<k>]]]]]]]]

Response

<CR><LF>OK<CR><LF>

or

<CR><LF> ERROR<CR><LF>

Parameter

<mode> (multiplexer Transparency Mechanism)

- 1 Multiplexer not active
- 0 Multiplexer enable
- 1 enhance mode(not support yet)

<subset>Initial control channel setup, not support yet

- 0 UIH frames used only (default value)
- 1 UI frames used only

<port_speed> Transmission rate (not relevant for USB interface,not support yet)

- 0 Auto-baud
- 1 9600 bps
- 2 19200 bps
- 3 38400 bps
- 4 57600 bps
- 5 115200 bps (Default value)
- 6 230400 bps
- 7 460800 bps

<N1>: Maximum frame size: 1-4096 (default value 31 for basic option), not yet supported

<T1>: Acknowledgement time in units of 10ms:1-255 (default value 10 (100ms), not yet supported

<N2>: Maximum number of re-transmissions: 0-100 (default value is 3), not yet supported

<T2>: Response timer for the MUX channel in units of 10ms: 2-255 (default value is 30 (300ms)), not yet supported

<T3> : Wake up response timer in seconds: 1-255 (default value is 10), not yet supported

<k> : Window size for Advanced operation with Error Recovery options: 1-7 (default value is 2), not yet supported



AT+CMUX=?

Response

<CR><LF>+CMUX: (list of supported <mode>s), (list of supported <subset>s), (list of supported<port_speed>s), (list of supported<N1>s), (list of supported<T1>s), (list of supported<N2>s), (list if supported<T2>s), (list of supported <T3>s), <list of supported <k>s)<CR><LF>

<CR><LF>OK<CR><LF>



AT+CMUX?

Response

If <mode> = 0, return:

<CR><LF> +CMUX:<mode>, [<subset>], <port_speed>, <N1>,<T>,<N2>,<T2>,<T3>[,<k>]<CR><LF>


<CR><LF>OK<CR><LF>

Otherwise , return:

<CR><LF>+CMUX:<mode><CR><LF>


<CR><LF>OK<CR><LF>

+CME ERROR: <err>


 AT+CMUX=0
OK

5.5. [AT#TCLRBDRT Clear historical baud rate](#)

This command is used to Clear historical baud rate.

 Set command
AT#TCLRBDRT

Response
<CR><LF>OK<CR><LF>

 AT+IPR=9600
OK

AT+IPR?
+IPR: 9600
OK

AT#TCLRBDRT
OK

AT+IPR?
+IPR: 0
OK

6. SIM COMMANDS

6.1. AT+CPIN Enter PIN

Set command sends to the MT 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 active application in the UICC (GSM or USIM) or SIM card.

Read command returns an alphanumeric string indicating whether some password is required or not.



AT+CPIN=<pin>[,<newpin>]

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<pin>,<newpin>: string type values

<code> values reserved by the present document:

READY MT is not pending for any password

SIM PIN MT is waiting UICC/SIM PIN to be given

SIM PUK MT is waiting UICC/SIM PUK to be given

PH-SIM PIN MT is waiting phone-to-SIM/UICC card password to be given

SIM PIN2 MT is waiting active application in the UICC (GSM or USIM) or SIM card PIN2 to be given (this <code> is recommended to be returned only when the last executed command resulted in PIN2 authentication failure (i.e. +CME ERROR: 17); if PIN2 is not entered right after the failure, it is recommended that MT does not block its operation)

SIM PUK2 MT is waiting active application in the UICC (GSM or USIM) or SIM card PUK2 to be given (this <code> is recommended to be returned only when the last executed command resulted in PUK2 authentication failure (i.e. +CME ERROR: 18); if PUK2 and new PIN2 are not entered right after the failure, it is recommended that MT does not block its operation)

PH-NET PIN MT is waiting network personalization password to be given

Note:

If the status of <code> is left below you need input commands

SIM PIN AT+CPIN=<pin>

SIM PUK AT+CPIN=<puk>,<newpin>

SIM PUK2 AT+CPIN=<puk2>,<newpin2>

SIM PIN 2 AT+CPIN=<pin2>



AT+CPIN=?

Response

<CR><LF>OK<CR><LF>



AT+CPIN?

Response

<CR><LF>+CPIN: <code><CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>



```
AT+CMEE=2
OK
AT+CPIN?
+CPIN: READY

OK
```

6.2. AT*MICCID Read USIM ICCID file

Read USIM ICCID file



```
AT*MICCID
Response
<CR><LF>*MICCID: <iccid><CR><LF>
<CR><LF>OK<CR><LF>
Parameter
<iccid>:SIM/USIM ICCID
```



```
AT*MICCID
*MICCID: 86860460020123456381

OK
```

7. NETWORK SERVICE COMMANDS

7.1. AT+CEREG EPS Network registration

Set command controls the presentation of an unsolicited result code +CEREG: <stat> when <n>=1 and there is a change in the MT network registration status, or code +CEREG: <stat>[,<lac>],[<ci>],[<AcT>]] 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 MT. Location information elements <lac> and <ci> are returned only when <n>=2 and MT is registered in the network.

Test command returns the range of supported <n>.



AT+CEREG=<n>

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<n>:

- 0 disable network registration unsolicited result code
- 1 enable network registration unsolicited result code: +CEREG: <stat>
- 2 enable network registration unsolicited result code: +CEREG: <stat>[,<tac>],[<ci>],[<AcT>],[rac]]
- 3 enable network registration unsolicited result code: +CEREG: <stat>[,<tac>],[<ci>],[<AcT>],[rac],[<cause_type>,<reject_cause>]]
- 4 enable network registration unsolicited result code: +CEREG: <stat>[,<lac>],[<ci>],[<AcT>],[rac][,],[<Active-Time>],[<Periodic-RAU>],[<GPRS-READY-timer>]]]]
- 5 enable network registration unsolicited result code: +CEREG:<stat>[,<tac>],[<ci>],[<AcT>],[rac][,],[<cause_type>],[<reject_cause>],[<Active-Time>],[<Periodic-RAU>],[<GPRS-READY-time>]]]]]]

<stat>:

- 0 not registered, MT is not currently searching a new operator to register to
- 1 registered, home network
- 2 not registered, but MT is currently searching a new operator to register to
- 3 registration denied
- 4 unknown
- 5 registered, roaming

<tac>: string type; two bytes tracking area code in hexadecimal format

<ci>: string type; four byte cell ID in hexadecimal format

<AcT>: integer type; access technology of the serving cell

9 NB-IOT

<rac>: NB-IOT, "00"

<cause_type>: integer type; indicates the type of <reject_cause>

0 Indicates that <reject_cause> contains an EMM cause value, see 3GPP TS 24.301 Annex A.

1 Indicates that <reject_cause> contains a manufacturer-specific cause.

<reject_cause>: integer type; contains the cause of the failed registration. The value is of type as defined by <cause_type>.

<Active-Time>: string type; one byte in an 8-bit format. Indicates the Active Time value (T3324) allocated to the UE in E-UTRAN. The Active Time value is coded as one byte (octet 3) of the GPRS Timer 2 information element coded as bit format (e.g. "00100100" equals 4 minutes). For the coding and the value range, see the GPRS Timer 2 IE in 3GPP TS 24.008 Table 10.5.163/3GPP TS 24.008. See also 3GPP TS 23.682 and 3GPP TS 23.401.

<Periodic-TAU>: string type; one byte in an 8-bit format. Indicates the extended periodic TAU value (T3412) allocated to the UE in E-UTRAN. The extended periodic TAU value is coded as one byte (octet 3) of the GPRS Timer 3 information element coded as bit format (e.g. "01000111" equals 70 hours). For the coding and the value range, see the GPRS Timer 3 IE in 3GPP TS 24.008, Table 10.5.163a/3GPP TS 24.008. See also 3GPP TS 23.682 and 3GPP TS 23.401.

**AT+CEREG?**

Response

when <n>=0, 1, 2 or 3 and command successful:

+CEREG: <n>,<stat>[,<tac>],[<ci>],[<AcT>,<rac>[,<cause_type>,<reject_cause>]]]

• when <n>=4 or 5 and command successful:

+CEREG:<n>,<stat>[,<tac>],[<ci>],[<AcT>],[<rac>][,<cause_type>],[<reject_cause>][,<Active-Time>],[<Periodic-TAU>]]]

• If error is related to wrong AT syntax or operation not allowed:

+CME ERROR: <err>**AT+CEREG=?**

Response

<CR><LF>**+CEREG: (list of supported <n>s)<CR><LF>**<CR><LF>**OK<CR><LF>****AT+CEREG?****+CEREG: 0,1**

OK

AT+CEREG=?**+CEREG: (0-5)**

OK

7.2. AT+COPS PLMN Reselection

Set command forces an attempt to select and register the LTE network operator. <mode> is used to select whether the selection is done automatically by the MT or is forced by this command to operator <oper>.

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.



AT+COPS=[<mode>[,<format>,<oper>[,<AcT>]]]

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<stat>:

- 0 unknown
- 1 available
- 2 current
- 3 forbidden

<oper> operator in format as per <mode>

<mode> 0 automatic mode; <oper> field is ignored

- 1 manual operator selection; <oper> field shall be present
- 2 manual deregister from network
- 3 set only <format> (for read command +COPS?) – not shown in Read command response
- 4 manual/automatic selected; if manual selection fails, automatic mode (<mode>=0) is entered

<format>

0: long format alphanumeric <oper>; can be up to 16 characters long

1: short format alphanumeric <oper>

2: numeric <oper>; GSM Location Area Identification number



AT+COPS?

Response

<CR><LF>+COPS: <mode>[,<format>,<oper>[,<AcT>]]<CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>



AT+COPS=?

Response

<CR><LF>+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)]<CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>



AT+COPS?

+COPS: 0,0,"46011",9

OK

7.3. AT+CESQ Get signal quality

Execution command returns received signal quality parameters. If the current serving cell is not an E-UTRA cell, <rsrq> and <rsrp> are set to 255. Test command returns values supported as compound values.



AT+CESQ
Response

```
<CR><LF>+CESQ:<rxlev>,<ber>,<rscp>,<ecno>,<rsrq>,<rsrp><CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>+CME ERROR: <err><CR><LF>
```

Parameter

<rxlev>:

- 0: -110 dBm or less
- 1: -110 dBm ≤ rssi < -109 dBm
- 2: -109 dBm ≤ rssi < -108 dBm
-:
- 61: -50dBm ≤ rssi < -49 dBm
- 62: -49dBm ≤ rssi < -48 dBm
- 63: -48dBm ≤ rssi
- 99: not known or not detectable

<ber>

- 0...7: as RXQUAL values RXQUAL_0...RXQUAL_7 asdefined in 45.008.
- 99: not known or not detectable

<rscp>

- 0: -120 dBm or less
- 1: -120 dBm ≤ rscp < -119 dBm
- 2: -119 dBm ≤ rscp < -118 dBm
-:
- 94: -27 dBm ≤ rscp < -26 dBm
- 95: -26 dBm ≤ rscp < -25 dBm
- 96: -25 dBm ≤ rscp
- 255: not known or not detectable

<ecno>

- 0: -24 dBm or less
- 1: -24dBm ≤ Ec/lo < -23.5 dBm
- 2: -23.5dBm ≤ Ec/lo < -23 dBm
-:
- 47: -1dBm ≤ Ec/lo < -0.5 dBm
- 48: -0.5dBm ≤ Ec/lo < 0 dBm
- 49: 0 dBm ≤ Ec/lo
- 255: not known or not detectable

<rsrq>:

- 0: -19.5 dB or less
- 1: -19.5dB ≤ rsrq < -19 dB
- 2: -19dB ≤ rsrq < -18.5 dB
-:
- 32: -4 dB ≤ rsrq < -3.5 dB
- 33: -3.5 dB ≤ rsrq < -3 dB
- 34: -3 dB ≤ rsrq
- 255: not known or not detectable

<rsrp>

- 0: -140 dBm or less
- 1: -140dBm ≤ rsrp < -139 dBm
- 2: -139dBm ≤ rsrp < -138 dBm
-:
- 95: -46dBm ≤ rsrp < -45 dBm
- 96: -45dBm ≤ rsrp < -44 dBm
- 97: -44dBm ≤ rsrp
- 255: not known or not detectable



AT+CESQ=?
Response

<CR><LF>+CESQ: (list of supports <rxlev>s), (list of supported <ber>s), (list of supported <rsrp>s),
(list of supported <ecno>s), (list of supported <rsrq>s), (list of supported <rsrp>s)<CR><LF>
<CR><LF>OK<CR><LF>



AT+CESQ=?
+CESQ: (0-63,99),(99),(255),(255),(0-34,255),(0-97,255)
OK

7.4. AT+CSQ Get signal quality

Execution command returns received signal quality parameters and ber.



AT+CSQ
Response

<CR><LF>+CSQ: <rsssi>,<ber><CR><LF>
<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<rsssi>:

0	-113dBm or less
1	-111dBm
2...30	-109... -53dBm
31	-51 dBm or greater
99	not known or not detectable

<ber> (in percent):

0...7	This parameter is only valid for GSM.
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	- more than 8.0%
99	-- not known or not detectable



AT+CSQ=?
Response

<CR><LF>+CSQ: (list of supported <rsssi>s),(list of supported <ber>s)<CR><LF>
<CR><LF>OK<CR><LF>



AT+CSQ
+CSQ: 17,99
OK

7.5. AT+CTZU Automatic Time Zone Update

Set command enables and disables automatic time zone update via NITZ. If setting fails in an MT error, +CME ERROR: <err> is returned. It will take effect once setting, and will invalid after restart the module. Read command returns the current settings in the MT.



AT+CTZU=<on/off>
Response

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<onoff>:

0 – Disable automatic time zone update via NITZ (default)

1 – Enable automatic time zone update via NITZ..



AT+CTZU=?
Response

<CR><LF>+CTZU: (list of supported <onoff>s)<CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>



AT+CTZU?
Response

<CR><LF>+CTZU: <onoff><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>



AT+CTZU=?
+CTZU: (0-1)

OK

AT+CTZU?
+CTZU: 0

OK

7.6. AT+CCLK Clock Management

Clock Management



AT+CCLK=<time>

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<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; range -47...+48). E.g. 6th of May 1994, 22:10:00 GMT+2 hours equals to "94/05/06,22:10:00+08"



AT+CCLK?

Response

<CR><LF>+CCLK: <time><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>



AT+CCLK=?

Response

<CR><LF>OK<CR><LF>



AT+CCLK="14/09/11,16:54:00+08"

OK

AT+CCLK?

+CCLK: "14/09/11,16:54:15+08"

OK

7.7. AT*MFRCLLCK Lock UE to specific frequency and optionally Cell ID

This command is used to lock UE to specific frequency and optionally Cell ID. It will take effect once setting, and will be invalid after restart the module.



```
AT*MFRCLLCK=<lock>,<earfcn>[,<earfcn_offset>[,<pci>]]
```

Response

```
<CR><LF>OK<CR><LF>
```

Parameter

<lock>: Integer value indicating whether to activate lock, or remove lock:

0: Remove lock

1: Activate lock

<earfcn>: Integer value indicating requested EARFCN on which to lock. Range 0- 262143. Value of 0 indicates to remove any lock for EARFCN and Cell

<earfcn_offset> Integer value indicating requested EARFCN offset:

0: Offset of -2

1: Offset of -1

2: Offset of -0.5

3: Offset of 0

4: Offset of 1

<pci>: Integer value: Physical cell ID. Range: 0-503



```
AT*MFRCLLCK=?
```

Response

```
<CR><LF>*MFRCLLCK: (0,1)[,(0-262143)],[,(0-4)[,(0-503)]]<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```



```
AT*MFRCLLCK?
```

Response

When not setting earfcn/cellid:

```
<CR><LF>*MFRCLLCK: 0<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

When setting earfcn/cellid

```
<CR><LF>*MFRCLLCK:<lock>[,<earfcn>,<earfcn_offset>[,<pci>]]<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```



```
AT*MFRCLLCK=1,10,3,301 // Lock to EARFCN 10, offset 0, PCI 301
```


```
OK
```


```
AT*MFRCLLCK=0 //Remove lock
```


```
OK
```

7.8. AT*MBAND Query Current Modem NB-IOT Operating Band

Query Current Modem NB-IOT Operating Band

 **AT*MBAND=?**
 Response
 <CR><LF>*MBAND: (list of valid <Current Band>s)<CR><LF>
 <CR><LF>OK<CR><LF>
Parameter
 <Current Band> Integer value indicating current selected NB-IOT band Valid values:
 1,2,3,5,8,12,13,17,18,19,20,26,28,66

 **AT*MBAND?**
 Response
 <CR><LF>*MBAND:<Current Band><CR><LF>
 <CR><LF>OK<CR><LF>
 Or
 <CR><LF>+CME ERROR: <err><CR><LF>

 **AT*MBAND=?**
 *MBAND: (1,2,3,5,8,12,13,17,18,19,20,26,28,66)
 OK
 AT*MBAND?
 *MBAND: 5
 OK

7.9. AT*MBSC Lock BAND

This command is used to lock band, and it will take effect after restart the module, the setting parameters will be saved even when power-off.



AT*MBSC=<number>,<support_bands_list>

Response

<CR><LF>*MBSC: 0<CR><LF>

<CR><LF>OK<CR><LF>

Or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

< number>: integer type, the band number need to lock.

<support_band_lists>: integer type, the bands list need to lock, and is divided by “,”



AT*MBSC?

Response

<CR><LF>*MBSC:<Support Band><CR><LF>

<CR><LF>OK<CR><LF>

Or

<CR><LF>+CME ERROR: <err><CR><LF>



AT*MBSC=3,3,5,8 //lock three bands, that is band3,5,8

*MBSC: 0

OK

7.10. AT*MENGINFO Query current network status and cell information

This command is used to query current network status, and modem status information for serving cell.



AT*MENGINFO=<mode>

Response

If <mode> = 0 display serving cell and up to 4 neighbor cell information:

<CR><LF>*MENGINFOSC:

<sc_earfcn>,<sc_earfcn_offset>,<sc_pci>,<sc_cellid>,<sc_rsrp>,<sc_rsrq>,<sc_rssi>,<sc_snr>,<sc_band>,<sc_tac>,<sc_ecl>,<sc_tx_pwr>]

[<CR><LF>*MENGINFONC: <nc_earfcn>,<nc_earfcn_offset>,<nc_pci>,<nc_rsrp>

[...]]

<CR><LF>OK<CR><LF>

If <mode> = 1 display data transfer information only if modem in RRC-CONNECTED state:

*MENGINFODT:

<RLC_UL_BLER>,<RLC_DL_BLER>,<MAC_UL_BLER>,<MAC_DL_BLER>,<MAC_UL_total_bytes>,<MAC_DL_total_bytes>,<MAC_UL_total_HARQ_TX>,<MAC_DL_total_HARQ_TX>,<MAC_UL_HARQ_re_TX>,<MAC_DL_HARQ_re_TX>,<RLC_UL_tput>,<RLC_DL_tput>,<MAC_UL_tput>,<MAC_DL_tput>

<CR><LF>OK<CR><LF>

If error is related to wrong AT syntax or incorrect <mode> or UE in incorrect state

+CME ERROR: <err>

Parameter

<mode> Integer value indicating requested engineering information.

0: Radio information for serving and neighbor cells

1: display data transfer information only if modem in RRC-CONNECTED state

Serving Cell/Neighbor Cell information:

<sc_earfcn> Integer value indicating the EARFCN for serving cell. Range 0- 262143

<sc_earfcn_offset> Integer value indicating the EARFCN offset for serving cell:

0: Offset of -2

1: Offset of -1

2: Offset of -0.5

3: Offset of 0

4: Offset of 1

<sc_pci> Integer value indicating the serving cell physical cell ID. Range 0 – 503.

<sc_cellid> String type; four byte (28 bit) cell ID in hexadecimal format for serving cell.

<sc_rsrp> Signed integer indicating serving cell RSRP value in units of dBm (can be negative value). Available only in RRC-IDLE state.

<sc_rsrq> Signed integer indicating serving cell RSRQ value in units of dB (can be negative value). Available only in RRC-IDLE state.

<sc_rssi> Signed integer indicating serving cell RSSI value in units of dBm (can be negative value). Available only in RRC-IDLE state.

<sc_snr> Signed integer value. Last SNR value for serving cell in units of dB. Available only in RRC-IDLE state.

<sc_band> Integer value; current serving cell band

<sc_tac> String type; two byte tracking area code (TAC) in hexadecimal format (e.g. "00C3" equals 195 in decimal).

<sc_ecl> Integer value. Last Enhanced Coverage Level (ECL) value for serving cell. Range 0-2.

<sc_tx_pwr> Signed integer value indicating current UE transmit power. Units of cBm Centibels relative to one milliwatt (can be negative value).

<nc_earfcn> Integer value indicating the EARFCN for neighbor cell. Range 0-262143

<nc_earfcn_offset> Integer value indicating the EARFCN offset for neighbor cell:

0: Offset of -2

1: Offset of -1

2: Offset of -0.5

3: Offset of 0

4: Offset of 1

<nc_pci> Integer value indicating the neighbor cell physical cell ID. Range 0-503.

<nc_rsrp> Signed integer indicating neighbor cell RSRP value in units of dBm (can be negative value).

Data Transfer Information:

<RLC_UL_BLER> Integer value. Represented in % value (range 0 to 100). UL block error rate (as per IRQ) in RLC. Calculated over all established RLC AM radio bearers. Calculated from the beginning of successfully established/resumed RRC connection or since previous AT*MENGINFO query with <mode>=1, whichever is later. Only valid in RRC-CONNECTED state.

<RLC_DL_BLER> Integer value. Represented in % value (range 0 to 100). DL block error rate (as per ARQ) in RLC. Calculated over all established RLC AM radio bearers. Calculated from the beginning of successfully established / resumed RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state.

<MAC_UL_BLER> Integer value. Represented in % value (range 0 to 100). UL block error rate (as per HARQ) in MAC for UL-SCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state.

<MAC_DL_BLER> Integer value. Represented in % value (range 0 to 100). DL block error rate (as per HARQ) in MAC for DL-SCH, excluding BCCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state.

<MAC_UL_total_bytes> Integer value. Total number of transport block bytes (re)transmitted on UL-SCH. Calculated for UL-SCH over all HARQ transmissions and retransmissions. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: bytes

<MAC_DL_total_bytes> Integer value. Total number of transport block bytes (re)transmitted on DL-SCH, excluding BCCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: bytes

<MAC_UL_total_HARQ_TX> Integer value. Total number of HARQ (re)transmissions for transport blocks on UL-SCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: (re)transmissions

<MAC_DL_total_HARQ_TX> Integer value. Total number of HARQ (re)transmissions for transport blocks on DL-SCH, excluding BCCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: (re)transmissions

<MAC_UL_HARQ_re_TX> Integer value. Number of HARQ retransmissions for transport blocks on UL-SCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: retransmissions

<MAC_DL_HARQ_re_TX> Integer value. Number of HARQ retransmissions for transport blocks on DL-SCH, excluding BCCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: retransmissions.

<RLC_UL_tput> Integer value. RLC uplink throughput. Calculated over all established RLC AM radio bearers. Calculated from the beginning of successfully established / resumed RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: kbits / s

<RLC_DL_tput> Integer value. RLC downlink throughput. Calculated over all established RLC AM radio bearers. Calculated from the beginning of successfully established / resumed RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: kbits / s

<MAC_UL_tput> Integer value. UL throughput in MAC for UL-SCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: kbits / s

<MAC_DL_tput> Integer value. DL throughput in MAC for DL-SCH, excluding BCCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: kbits / s

Note: when the module return +CME ERROR, the error code are shown as bellows:

- 1: No Service state
- 2: In cell search state
- 3: ERRC is deactivated
- 4: In cell re-selection state
- 5: In reestablishment state
- 6: In PSM state
- 7: No data transfer in idle state



AT*MENGINFO=?

Response

```
<CR><LF>*MENGINFO: (list of supported <mode>)
<CR><LF>OK<CR><LF>
```



AT*MENGINFO?

Response

```
<CR><LF>OK<CR><LF>
```



(power up in in CFUN=0 mode)

```
AT*MENGINFO=0
```

```
ERROR //Cannot display information in CFUN=0 mode
```

```
AT+CFUN=1
```

```
OK
```

(Not yet registered to network)

```
AT*MENGINFO=0
```

```
OK
```

(Registered to network)

```
AT*MENGINFO=0
```

```
*MENGINFOC: 3701,3,69,"27447553",-1073,-1175,-1145,290,18,11,0,0,-35
```

```
*MENGINFONC: 3701,0,60,-1073
```

```
*MENGINFONC: 3369,1,37,-1073
```

```
*MENGINFONC: 3210,2,23,-1073
```

```
*MENGINFONC: 3001,1,15,-1073
```

```
OK
```

7.11. AT*MNBIOTRAI Actively release RRC connection

This command is used for active RRC_Release in the following scenarios:

1. Need to upload 1 UL packet and no DL packet reply
2. Need to upload 1 UL packet and 1 DL packet reply



AT*MNBIOTRAI=<rai>
Response
<CR><LF>OK<CR><LF>

Parameter

<rai>:Integer value

0:No valid information (default)

1:upload 1 UL packet and no DL packet reply

2:upload 1 UL packet and 1 DL packet reply



AT*MNBIOTRAI?
Response
<CR><LF>*MNBIOTRAI:rai<CR><LF>
<CR><LF>OK<CR><LF>



AT*MNBIOTRAI=?
Response
<CR><LF>*MNBIOTRAI: (0-2)<CR><LF>
<CR><LF>OK<CR><LF>



AT*MNBIOTRAI?
*MNBIOTRAI: 0

OK
AT*MNBIOTRAI=?
*MNBIOTRAI: (0-2)

OK

AT*MNBIOTRAI=1
OK

7.12. AT*RRCRELEASE Actively release RRC connection

This command is used for active RRC_Release without any data bearer, and can directly release the RRC connection.



```
AT*RRCRELEASE  
Response  
<CR><LF>OK<CR><LF>
```



```
AT*RRCRELEASE=?  
Response  
<CR><LF>OK<CR><LF>
```



```
AT*RRCRELEASE?  
Response  
<CR><LF>OK<CR><LF>
```



```
AT*RRCRELEASE  
OK
```

```
AT*RRCRELEASE=?  
OK
```

```
AT*RRCRELEASE?  
OK
```

7.13. AT+CCIOTOPT CloT Optimization Configuration

The set command controls which CloT EPS optimizations the UE indicates as supported and preferred in the ATTACH REQUEST and TRACKING AREA UPDATE REQUEST messages. The command also allows reporting of the CloT EPS optimizations that are supported by the network. A UE supporting CloT functionality may support control plane CloT EPS optimization or user plane CloT EPS optimization or both (see 3GPP TS 24.301, sub-clause 9.9.3.34). Based on the application characteristics the UE may prefer to be registered for control plane CloT EPS optimization or for user plane CloT EPS optimization (see 3GPP TS 24.301, sub-clause 9.9.3.0B).

Further the network may support control plane CloT EPS optimization or user plane CloT EPS optimization or both (see 3GPP TS 24.301, sub-clause 9.9.3.12A).

The set command is used also to control the unsolicited result code +CCIOTOPTI. An unsolicited result code +CCIOTOPTI: <supported_Network_opt> is used to indicate the supported CloT EPS optimizations by the network.

The read command returns the current settings for supported and preferred CloT EPS optimizations and the current status of unsolicited result code +CCIOTOPTI.



Execution command

AT+CCIOTOPT=[<n>,[<supported_UE_opt>,<preferred_UE_opt>]]

Response

- OK
- If error is related to wrong AT syntax:
+CME ERROR: <err>

Unsolicited result code

- +CCIOTOPTI: <supported_Network_opt> is used to indicate the supported CloT EPS optimizations by the network.

Parameters

- <n>: integer type, enables or disables reporting of unsolicited result code +CCIOTOPTI.
 - 0 Disable reporting.
 - 1 Enable reporting.
 - 3 Disable reporting and reset the parameters for CloT EPS optimization to the default values.
- <supported_UE_opt>: integer type; indicates the UE's support for CloT EPS optimizations.
 - 1 Support for control plane CloT EPS optimization.
 - 3 Support for both control plane CloT EPS optimization and user plane CloT EPS optimization.
- <preferred_UE_opt> integer type; indicates the UE's preference for CloT EPS optimizations.
 - 0 No preference
 - 1 Preference for control plane CloT EPS optimization
 - 2 Preference for user plane CloT EPS optimization
- <supported_Network_opt>: integer type; indicates the Network support for CloT EPS optimizations.
 - 0 No support
 - 1 Support for control plane CloT EPS optimization.
 - 2 Support for user plane CloT EPS optimization.
 - 3 Support for both control plane CloT EPS optimization and user plane CloT EPS optimization.



Test Command AT+CCIOTOPT=?

Response

- +CCIOTOPT: (list of supported <n>s),(list of supported <supported_UE_opt>s),(list of supported <preferred_UE_opt>s)
- OK



Read command AT+CCIOTOPT?

Response

- +CCIOTOPT :<n>,<supported_UE_opt>,<preferred_UE_opt>

7.14. AT+CCHO Open UICC Logical Channel

Execution of this command causes the MT to return <sessionid> to allow the TE to identify a channel that is being allocated by the currently selected UICC, which is attached to ME. The currently selected UICC will open a new logical channel; select the application identified by the <dfname> received with this command and return a session Id as the response. The ME restricts the communication between the TE and the UICC to this logical channel.

This <sessionid> is used when sending commands with Generic UICC Logical Channel access +CGLA commands.



Execution command

AT+CCHO=<dfname>

Response

- <sessionid>
- OK
- If error is related to wrong AT syntax:
+CME ERROR: <err>

Parameters

- <dfname>: String type in hexadecimal character format. All selectable applications in the UICC are referenced by a DF name coded on 1 to 16 bytes
- <sessionid>: integer type; a session Id to be used to target a specific application on the smart card (e.g. (U)SIM, WIM, ISIM) using logical channels mechanism
- See 3GPP TS 31.101 for more information about defined values.

7.15. AT+CCHC Close UICC Logical Channel

This command asks the ME to close a communication session with the active UICC. The ME closes the previously opened logical channel. The TE will no longer be able to send commands on this logical channel. The UICC will close the logical channel when receiving this command.



Execution command

AT+CCHC=<sessionid>

Response

- CCHC
- OK
- If error is related to wrong AT syntax:
+CME ERROR: <err>

Parameters

- <sessionid>: integer type; the session used to target a specific application on the smart card (e.g. (U)SIM, WIM, ISIM) using logical channels mechanism
- See 3GPP TS 31.101 for more information about defined values.

7.16. AT+CGLA Generic UICC Logical Channel Access

Set command transmits to the MT the <command> is sent as is to the selected UICC. The UICC <response> is sent back by the MT to the TA as is.

This command allows a direct control of the currently selected UICC by a distant application on the TE. Although +CGLA allows the TE to take control over the UICC-MT interface, there are some functions of the UICC-MT interface that logically do not need to be accessed from outside the TA/MT and, for security reasons the GSM network authentication should not be handled outside the TA/MT. So, a Run GSM Algorithm command or an Authenticate command in GSM context shall not be allowed whether the +CGLA is locked or unlocked. However, the TE may send Authenticate commands in other security contexts (e.g. EAP security context).



Execution command

AT+CGLA=<sessionid>,<length>,<command>

Response

- +CGLA: <length>,<response>
- OK
- If error is related to wrong AT syntax:
+CME ERROR: <err>

Parameters

- <sessionid>: integer type; this is the identifier of the session used to send the APDU commands to the UICC. It is mandatory to send commands to the UICC when targeting applications on the smart card using a logical channel other than the default channel (channel "0").
 - <length> : integer type; length of the characters that are sent to TE in <command> or <response> (two times the actual length of the command or response)
 - <command> : command passed on by the MT to the UICC in the format as described in 3GPP TS 31.101 (hexadecimal character format)
 - <response> : response to the command passed on by the UICC to the MT in the format as described in 3GPP TS 31.101 (hexadecimal character format)

7.17. AT+CPINR Remaining PIN Retries

Execution command cause the UE to return the number of remaining PIN retries for the UE passwords with intermediate result code +CPINR: <code>,<retries>[,<default_retries>] for standard PINs and +CPINRE: <ext_code>,<retries>[,<default_retries>] for manufacturer specific PINs. One line with one intermediate result code is returned for every <code> or <ext_code> selected by <sel_code>.

When execution command is issued without the optional parameter <sel_code>, intermediate result codes are returned for all <code>s and <ext_code>s.

In the intermediate result codes, the parameter <default_retries> is an optional (manufacturer specific) parameter, per <code> and <ext_code>.

Note that the modem does not support the +CPINRE indication.



Execution Command

AT+CPINR[=<sel_code>]

Response

- [+CPINR: <code>,<retries>[,<default_retries>]
- [<CR>,<LF>:CPINR: <code>,<retries>[,<default_retries>]
- [...]]
- OK
- If error is related to wrong AT syntax:
+CME ERROR: <err>

Parameters

- <sel_code>: String type. Same values as for the <code> parameter. These values are strings and shall be indicated within double quotes. Wildcard match by "*", meaning match any (sub-)string, or '?' meaning an character can be used.
- <retries>: Integer type. Number of remaining retries per PIN.
- <default_retries>: Integer type. Number of default/initial retries per PIN.
- <code>: Type of PIN. All values listed under the description of the AT+CPIN Command, <code> parameter except "READY".



Test Command AT+CPINR=?

Response

- OK



Read Command AT+CPINR?

Response

- +CME ERROR: <err>



+CPINR examples

AT+CPINR=? Test command

OK

AT+CPINR? Read command

ERROR

AT+CPINR="SIM*" Execute command. Read all SIM PIN retries starting

+CPINR: "SIM PIN",3,3 with the string "SIM".

+CPINR: "SIM PUK",10,10

+CPINR: "SIM PIN2",3,3

+CPINR: "SIM PUK2",10,10

+CPINR: "SIM UPIN",3,3

+CPINR: "SIM UPUK",10,10

OK

AT+CPINR Execute command. Read all SIM PIN retries.


```
+CPINR: "SIM PIN",3,3    with the string "SIM".
+CPINR: "SIM PUK",10,10
+CPINR: "SIM PIN2",3,3
+CPINR: "SIM PUK2",10,10
+CPINR: "SIM UPIN",3,3
+CPINR: "SIM UPUK",10,10
OK
```

```
AT+CPINR="SIM PIN*" Execute command. Read all SIM PIN retries
starting
+CPINR: "SIM PIN",3,3    with the string "SIM PIN".
OK
```

7.18. AT+CGATT GPRS/Packet Domain attach or detach



3GPP TS 27.007



Set command

```
AT+CGATT=[<state>]
```

Response

- OK
- ERROR

Parameter

- <state> indicates the state of GPRS/Packet Domain attachment

0 – detached

1 – attached

Other values are reserved and will result in an ERROR response to the execution command.



Test command

```
AT+CGATT=?
```

Response

- +CGATT: (list of supported <state>s)

Parameter

- See set command



Read command

```
AT+CGATT?
```

Response

- +CGATT: <state>

Parameter

- See set command

7.19. AT+CGACT PDP context activate or deactivate

The execution command is used to activate or deactivate the specified PDP context (s). After the command has completed, the UE remains in V.250 command state. If any PDP context is already in the requested state, the state for that context remains unchanged.

If the UE is not PS attached when the activation form of the command is executed, the UE first performs a PS attach and then attempts to activate the specified contexts.

Note that in the 27.007 specification there is the following statement:

- For EPS, if an attempt is made to disconnect the last PDN connection, then the UE responds with ERROR or, if extended error responses are enabled, a +CME ERROR.

This applies when use of CID0 is enabled for the PDN connection activated during attach (AT*MLTEGCF=2). In fact, entering AT+CGACT=<0 or 1>,0 will generate ERROR response

If CID0 mode is not enabled (AT*MLTEGCF=0/1) then it is possible to enter the AT+CGACT command to deactivate the last PDN connection from the point of view of the middleware. When the user uses AT+CGACT to disconnect the last PDN connection the following occurs:

- The PDN connection is preserved in the protocol stack
- The PDN connection is disconnected at the middleware, so the <cid> for the PDN connection is marked as deactivated

- OK response is returned rather than ERROR

For EPS, the activation request for an EPS bearer resource will be answered by the network by either an EPS dedicated bearer activation or EPS bearer modification request. The request must be accepted by the UE before the PDP context can be set in to established state.

- If no <cid>s are specified the activation form of the command activates all defined contexts.
- If no <cid>s are specified the deactivation form of the command deactivates all active contexts.

The read command returns the current activation states for all the defined PDP contexts.

The test command is used for requesting information on the supported PDP context activation states.



Set command

AT+CGACT=[<state>[,<cid>[,<cid>[,...]]]]

Response

- OK
- NO CARRIER
- ERROR

Parameter

- <state> indicates the state of PDP context activation
 - 0 – deactivated
 - 1 – activated

Other values are reserved and will result in an ERROR response to the execution command.

- <cid> a numeric parameter which specifies a particular PDP context definition (see +CGDCONT command)



Reference 3GPP TS 27.007

- If context is deactivated successfully, NO CARRIER is returned
- If CID0 for PDN activated during attach is enabled, then AT+CGACT=<0 or 1>,0 will cause ERROR r



Test command

AT+CGACT=?

Response

- +CGACT: (list of supported <state>s)

Parameter

See set command



Read command

AT+CGACT?

Response

- +CGACT: <cid>,<state>

- [`<CR><LF>`]+CGACT: `<cid>`,`<state>`
- [...]]
- Parameter
- See set command

7.20. AT+CGPADDR Show PDP address



Set command

AT+CGPADDR= [`<cid>` [,`<cid>` [,...]]]

Response

+CGPADDR: `<cid>`,`<PDP_addr>`

[`<CR><LF>`]+CGPADDR: `<cid>`,`<PDP_addr>` [...]]

Parameter

- `<cid>` a numeric parameter which specifies a particular PDP context definition (see +CGDCONT command). If no `<cid>` is specified, the addresses for all defined contexts are returned.
- `<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 +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_address>` is omitted if none is available.



Test command

AT+CGPADDR=?

Response

- +CGPADDR: (list of defined `<cid>`s)

Parameter

- See set command



Reference

3GPP TS 27.007

7.21. AT+IPCONFIG Query the device's IP address

This command can be used to query the IP address for the device.



Active command
 AT+IPCONFIG
 Response
 • +IPCONFIG: <IP addr>
 OK
 • ERROR



```
AT+IPCONFIG
+IPCONFIG: fe80:0:0:0:3c:ffb8:f4c9:1207

+IPCONFIG: 2001:14bb:170:4c91:3c:ffb8:f4c9:1207

+IPCONFIG: 178.55.211.180

+IPCONFIG: 127.0.0.1

OK
```

7.22. AT+CGEREP Packet Domain Event Reporting

Set command enables or disables sending of unsolicited result codes, +CGEV: XXX from UE to TE in the case of certain events occurring in the Packet Domain UE 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. Read command returns the current mode and buffer settings. Test command returns the modes and buffer settings supported by the UE as compound values.



Execution Command
 AT+CGEREP=[<mode>]
 Response
 • OK
 • If error is related to wrong AT syntax:
 +CME ERROR: <err>
 Parameters
 <mode>
 0 buffer unsolicited result codes in the UE; if UE result code buffer is full, the oldest ones can be discarded. No codes are forwarded to the TE.
 1 discard unsolicited result codes when UE-TE link is reserved (e.g. in on-line data mode); otherwise forward them directly to the TE
 <bfr>
 0 UE buffer of unsolicited result codes defined within this command is cleared when <mode> 1 is entered

Unsolicited Result Codes supported:

For network attachment, the following unsolicited result codes and the corresponding events are defined:

+CGEV: NW DETACH

The network has forced a PS detach. This implies that all active contexts have been deactivated. These are not reported separately.

+CGEV: ME DETACH

The mobile termination has forced a PS detach. This implies that all active contexts have been deactivated. These are not reported separately.

For PDP context activation, the following unsolicited result codes and the corresponding events are defined:

+CGEV: NW PDN ACT <cid>

The network has activated a context. The context represents a Primary PDP context in GSM/UMTS. The <cid> for this context is provided to the TE. The format of the parameter <cid> is found in command +CGDCONT.

NOTE 1: This event is not applicable for EPS.

+CGEV: ME PDN ACT <cid>[,<reason>[,<cid_other>]]

The mobile termination has activated a context. The context represents a PDN connection in NB-IOT. The <cid> for this context is provided to the TE. This event is sent either in result of explicit context activation request (+CGACT), or in result of implicit context activation request associated to attach request (+CGATT=1). The format of the parameter <cid> and <cid other> are found in command +CGDCONT.

For PDP context deactivation, the following unsolicited result codes and the corresponding events are defined:

+CGEV: NW PDN DEACT <cid>

The network has deactivated a context. The context represents a PDN connection in NB-IOT. The associated <cid> for this context is provided to the TE. The format of the parameter <cid> is found in command +CGDCONT.

NOTE 2: Occurrence of this event replaces usage of the event

+CGEV: NW DEACT <PDP_type>, <PDP_addr>, [<cid>]

+CGEV: ME PDN DEACT <cid>

The mobile termination has deactivated a context. The context represents a PDN connection in NB-IOT. The <cid> for this context is provided to the TE. The format of the parameter <cid> is found in command +CGDCONT.

NOTE 3: Occurrence of this event replaces usage of the event

+CGEV: ME DEACT <PDP_type>, <PDP_addr>, [<cid>]

For other PDP context handling, the following unsolicited result codes and the corresponding events are defined:

+CGEV: REJECT <PDP_type>, <PDP_addr>

A network request for context activation occurred when the UE was unable to report it to the TE with a +CRING unsolicited result code and was automatically rejected. The format of the parameters <PDP_type> and <PDP_addr> are found in command +CGDCONT.

NOTE 6: This event is not applicable for EPS.

+CGEV: NW REACT <PDP_type>, <PDP_addr>, [<cid>]

The network has requested a context reactivation. The <cid> that was used to reactivate the context is provided if known to the UE. The format of the parameters <PDP_type>, <PDP_addr> and <cid> are found in command +CGDCONT.

NOTE 7: This event is not applicable for EPS.

Parameter:

<PDP_type> Packet Data Protocol type (see +CGDCONT command)

<PDP_addr> Packet Data Protocol address (see +CGDCONT command)

<cid> Context Id (see +CGDCONT command)

Note: <cid> only given if known to the UE.

<class> GPRS mobile class (see +CGCLASS command)

<event_type> Integer type parameter indicates whether this is an informational Event of whether the TE as acknowledged it.

0 Informational event

1 Information request: Acknowledgement required. The Acknowledgement can be accept or reject, see AT+CGANS.

<change_reason> Integer type parameter indicates what kind of change occurred.

1 TFT only changed

2 QoS only changed

3 Both TFT and QoS changed

<reason> Integer type parameter indicates the reason why the context activation request for PDP type IPV4V6 was not granted. This parameter is only included if the requested PDP type associated with <cid> is IPV4V6, and the PDP type assign by the network for <cid> is either IPV4 or IPV6

- 0 IPV4 only allowed
- 1 IPV6 only allowed
- 2 single address bearers only allowed
- 3 single address bearers only allowed and MT initiated context activation for a second address type bearer was not successful

<cid_other> Indicated the context identifier allocated by MT for an MT initiated context of a second address type. MT shall only include this parameter if <reason> parameter indicates single address bearers only allowed, and MT support MT initiated context activation of a second address type without additional commands from the TE, and MT has activated the PDN connection or PDP context associated with <cid_other>.



Test Command AT+CGEREP=?
Response

- +CGEREP: (list of supported <mode>s), (list of supported <bfr>s)
- OK



Read Command AT+CGEREP?
Response

- +CGEREP: <mode>,<bfr>
- OK

7.23. AT+CGREG Network registration status

This command is used to display the packet switched network registration status.



Set command
AT+CGREG=[<n>]

Response
Parameter

<n> 0 disable network registration unsolicited result code
1 enable network registration unsolicited result code +CGREG: <stat>
2 enable network registration and location information unsolicited result code +CGREG: <stat>,<lac>,<ci>[,<Act>,<rac>]
<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 for a new operator to register to
3 registration denied
4 unknown
5 registered, roaming
6 registered for "SMS only", home network (applicable only when <Act> indicates E-UTRAN
7 registered for "SMS only", roaming (applicable only when <Act> indicates E-UTRAN
<lac> string type; two byte location area code in hexadecimal format (e.g. "00C3" equals 195 in decimal)
<ci> string type; four byte UTRAN/GERAN/E-UTRAN cell ID in hexadecimal format
<Act> access technology of the registered network
9 NB-IoT
<rac> string type; one byte routing area code in hexadecimal format

Note

- The set command controls the presentation of an unsolicited result code +CGREG: <stat> when <n>=1 and there is a change in the UE's GPRS network registration status, or code

+CGREG: <stat>[,<lac>,<ci>[,<AcT>,<rac>]] when <n>=2 and there is a change of the network cell.

- For NB-IoT product, only <AcT> value of 9 is valid.



Test command

AT+CGREG=?

Response

- +CGREG: (list of supported <n>s)
- Parameter
- See set command



Read command

AT+CGREG?

Response

- +CGREG: <n>,<stat>[,<lac>,<ci>[,<AcT>]]
- +CME ERROR: <err>
- Parameter
- See set command



Reference

3GPP TS 27.007

7.24. AT+CGPIAF Printing IP Address Format

Set command decides what format to print IPV6 address parameters of other AT commands. See RFC 4291 for details of the IPv6 address format.

The +CGPIAF parameters <IPv6_AddressFormat>, <IPv6_SubnetNotation>, <IPv6_LeadingZeros> and <IPv6_CompressedZeros> affect the following commands and parameters:

- 1) in +CGTFT and +CGTFTRDP, the <source address and subnet mask>;
- 2) in +CGDCONT, the <PDP_addr>;
- 3) in +CGPADDR, the <PDP_addr_1> and <PDP_addr_2>;
- 4) in +CGCONTRDP, the <source address and subnet mask>, <DNS_prim_addr>,<DNS_sec_addr>,<P_CSCF_prim_addr> and <P_CSCF_sec_addr>;

Read command returns the current command parameter settings.

Test command returns values supported as compound values.



Execution Command

AT+CGPIAF=[IPv6_AddressFormat>[,<IPv6_SubnetNotation>[,<IPv6_LeadingZeros>[,<IPv6_CompressZero s>]]]]

Response

OK

If error is related to wrong AT syntax:

+CME ERROR: <err>

Parameters

<IPv6_AddressFormat>: Integer type, decides the IPV6 address format. Relevant for all AT command parameters that can hold an IPV6 address.

0: Use IPv4-like dot-notation. IP address, and Subnetwork mask if applicable, are dot-separated.

Example:

For <source address and subnet mask>:

"32.1.13.184.0.0.205.48.0.0.0.0.0.0.0.255.255.255.255.255.255.255.255.240.0.0.0.0.0.0.0"

For other IP address parameters: "32.1.13.184.0.0.205.48.0.0.0.0.0.0"

1: Use IPV6-like colon notation. IP address, and subnetwork mask if applicable and when given

explicitly, are separated by a space.

Example:

For <source address and subnet mask>:

"2001:0DB8:0000:CD30:0000:0000:0000:0000 FFFF:FFFF:FFFF:FFF0:0000:0000:0000:0000"

For other IP address parameters: "2001:0DB8:0000:CD80:0000:0000:0000:0000"

<IPv6_SubnetNotation>: Integer type, decides the subnet-notation for <source Address and subnet mask>. Setting does not apply if <IPv6_AddressFormat>=0.

0: Both IP Address and subnet mask are stated Explicitly, separated by a space.

Example:

"2001:0DB8:0000:CD30:0000:0000:0000:0000 FFFF:FFFF:FFFF:FFF0:0000:0000:0000:0000"

1 The printout format is applying / (forward slash) subnet-prefix Classless Inter-Domain Routing (CIDR) notation:

Example: "2001:0DB8:0000:CD30:0000:0000:0000:0000/60"

<IPv6_LeadingZeros>: Integer type, decides whether leading zeros are Omitted or not. Setting does not apply if <IPv6_AddressFormat>=0.

0: Leading zeros are omitted.

Example: "2001:DB8:0:CD30:0:0:0:0"

1: Leading zeros are included.

Example: "2001:0DB8:0000:CD30:0000:0000:0000:0000"

<IPv6_CompressZeros>: Integer type, decides whether 1-n instances of 16 bit zero-values are replaced by only "...". This Applies only once. Setting does not apply if <IPv6_AddressFormat>=0.

0: No zero compression.

Example: "2001:DB8:0:CD30:0:0:0:0"

1 Use zero compression.

Example: "2001:DB8:0:CD30::"



Read Command AT+CGPIAF?

Response

- +CGPIAF:

<IPv6_AddressFormat>,<IPv6_SubnetNotation>,<IPv6_LeadingZeros>,<IPv6_CompressZeros>

- OK

- +CME ERROR: <err>



Test Command AT+CGPIAF=?

Response

- +CGPIAF: (list of supported <IPv6_AddressFormat>s), (list of supported <IPv6_SubnetNotation>s), (list of supported <IPv6_LeadingZeros>s), (list of supported <IPv6_CompressZeros>s)

- OK



+CGPIAF examples

AT+CGPIAF=? Test command

+CGPIAF: (0,1),(0,1),(0,1),(0,1)

OK

AT+CGPIAF? Read command

+CGPIAF: 0,0,0,0 All Settings at default format

OK

AT+CGPIAF=1,1,0,1 Set command. Use IPV6-like notation, with the "/"

OK format for subnet mask, omit leading zeros and use zero compression.

7.25. AT+CGDEL Delete Non-Active PDP Contexts

The execution command +CGDEL=<cid> removes the indicated PDP context and removes all associated data related to the indicated PDP contexts that are not activated. The AT command will not delete or remove information for activated PDP contexts. The removed PDP context is listed by the +CGDEL: <cid> intermediate result code.

- If <cid> points to a primary PDP context, the PDP context will be deleted together with all linked secondary PDP contexts if none of the PDP contexts are activated.
- If <cid> points to a secondary PDP context, the PDP context will be deleted if it is not activated.

A special form of the command can be given as +CGDEL (with the =<cid> omitted). In this form, all primary PDP contexts that are not activated or have any activated secondary PDP contexts will be removed and all secondary PDP contexts that are not activated will be removed. The associated data of all the deleted PDP contexts will be removed, and the removed PDP context are listed by the +CGDEL: <cid>[,<cid>[,...]] intermediate result code. Activated PDP contexts will not cause this form of the command to return ERROR or +CME ERROR.

Note, +CGDEL will remove associated PDP context data that can be set by the AT commands +CGDCONT, +CGDSCONT, +CGTFT, +CGEQREQ, +CGEQMIN and +CGEQOS.

For an attempt to delete PDP context(s) which would violate these rules, a +CME ERROR response is returned.



Execution Command

AT+CGDEL=<cid>]

Response

- +CGDEL: <cid>[,<cid>[,...]]
- OK
- If error is related to wrong AT syntax:
+CME ERROR: <err>

Parameters

<cid>: a numeric parameter which specifies a particular PDP context Definition.



Read Command AT+CGDEL?

Response

- +CME ERROR: <err>



Test Command AT+CGDEL=?

Response

- OK

7.26. AT+CGAUTH Define PDP Context Authentication Parameters

Set command allows the user to specify authentication parameters for a PDP context identified by the (local) context identification parameter <cid> used during the PDP context activation and the PDP context modification procedures. Since the <cid> is the same parameter that is used in the +CGDCONT and +CGDSCONT commands, +CGAUTH is effectively as an extension to these commands.

Note, this AT command transfers information in the clear that can be regarded as sensitive in security terms. Care must be exercised in providing this command where the AT commands are used in insecure
The read command returns the current settings for each defined context.
The test command returns values supported as a compound value.



Execution Command

AT+CGAUTH=<cid>[,<auth_prot>[,<userid>[,<password>]]]

Response

- When <auth_prot>/<username>/<password> set:
 - OK
 - When no <auth_prot>/<username>/<password> set displays current auth_prot username and password for <cid>:
 - +CGAUTH:<cid>,<auth_prot>,<username>,<password>
 - OK
- OK
- If error is related to wrong AT syntax:
 - +CME ERROR: <err>

Parameters

<cid>: A numeric parameter which specifies a particular PDP context definition (see the +CGDCONT and +CGDSCONT commands).

<auth_prot>: Numeric parameter. Authentication protocol used for this PDP context.

0: None. Used to indicate that no authentication protocol is used for this PDP context. Username and password are removed if previously specified.

1: PAP

<userid>: String type. User name for access to the IP network.

<password>: String type. Password for access to the IP network.



Read Command AT+CGAUTH?

Response

- [+CGAUTH: <cid>,<auth_prot>,<userid>,<password>]
- [<CR><LF>+CGAUTH: <cid>,<auth_prot>,<userid>,<password>]
- [...]
- OK



Test Command AT+CGAUTH=?

Response

- +CGAUTH: (range of supported <cid>s),(list of supported <auth_prot>s),(range of supported <userid>s),(range of supported <password>s)
- OK

7.27. AT+CGDATA Enter Data State

This command is used activate a PDP context / EPS bearer rather than using the ATD*99# method. It will be used by the Packet Transport mechanism for activating a PDP context / EPS bearer.

Note that it is possible for AT+CGDATA to re-use an already active context as long as the context was activated with no data connection on the same channel (i.e. activated with AT+CGACT).



Set command

AT+CGDATA=[<L2P>[,<cid>[,<cid>[,...]]]]

Response

- OK
- ERROR

Parameter

- <L2P> a string parameter that indicates the layer 2 protocol to be used between the TE and MT:
M-PT – Packet Transport Mechanism protocol for a PDP such as IP
Other values are not supported and will result in an ERROR response to the execution command.
- <cid> a numeric parameter which specifies a particular PDP context definition (see +CGDCONT command)



Test command

AT+CGDATA=?

Response

- +CGDATA: (list of supported <L2P>s)

Parameter

- See set command

7.28. AT*MBANDSL Set Modem NB-IOT Search Prefer Band List



Set command

AT*MBANDSL=<Enable>[,<band number>,<band1>,<band2>,<band3>,<band4>]

Response

- OK
- +CME ERROR: <err>

Parameter

< Enable > Integer value indicating search prefer band list enable or disable

0 :disable

1:enable

< band number > Integer value indicating search prefer band number

Valid values: 1,2,3,4

< band1> Integer value indicating current search prefer NB-IOT band

Valid values: 1,2,3,5,8,11,12,13,17,18,19,20,21,25,26,28,31,66,70

7.29. AT*MNBLOTDT NB-IoT Data Type

This command is used to set the NB-IoT data type per APN (Normal or Exceptional data).



Execution command

AT*MNBLOTDT=<type>[,<cid>[,<cid>[,...]]]

Response

- OK

Parameters

- <type>: integer type
0 – Normal data (default)
1 – Exceptional data
- <cid>: integer type. Specifies a particular PDP context definition.
If no <cid>s are specified the command sets <type> for all active PDP contexts.

Note

- The UE will not remember this setting over sleep cycles (i.e. the UE will fall back to default setting after sleep)



Test Command AT*MNBLOTDT=?

Response

- *MNBLOTDT: (list of supported <types>s)
- OK



Read command

AT*MNBLOTDT?

Response

- Displays <type> for all active PDP contexts:
[*MNBLOTDT:<cid>,type]
[<CR><LF>*MNBLOTDT:<cid>,<type>]
[...]
OK

7.30. AT*MLPINFO Report packet domain packet counters

This command is used to report packet domain packet counters.



Execution command

AT*MLPINFO

Response

```
<CR><LF>*MLPINFO: <sleep_duration>,<rx_time>,<tx_time><CR><LF>
<CR><LF>OK<CR><LF>
```

- If error is related to wrong AT syntax
<CR><LF>+CME ERROR: <err><CR><LF>

Set Command

AT*MLPINFO=[<mode>]

Response

```
<CR><LF>*MLPINFO: <sleep_duration>,<rx_time>,<tx_time><CR><LF>
<CR><LF>OK<CR><LF>
```

- If error is related to wrong AT syntax
<CR><LF>+CME ERROR: <err><CR><LF>

Parameters

<mode> Integer value indicating the query mode.

0 Return the time in unit of 0.1 second

1 Return the time in unit of millisecond

<sleep_duration> Integer value indicating the total sleep duration from the latest boot up, in unit of 0.1 second or millisecond.

<rx_time> Integer value indicating the total RX time from the latest boot up, in unit of 0.1 second or millisecond.

<tx_time> Integer value indicating the total TX time from the latest boot up, in unit of 0.1 second or millisecond.

Note: Execution command AT*MLPINFO returns <sleep_duration>,<rx_time>,<tx_time> in 0.1 second.

8. POWER SAVING COMMANDS

8.1. AT+CEDRXS eDRX Settings

This setting instruction is used to set eDRX parameter, enable/disable eDRX, and request eDRX parameter set in a particular system. This command will take effect once setting, and the parameters will be saved when power-off

When <mode>=2, the setting instruction can also make the module report actively when an eDRX-related parameter provided by network changes.

+CEDRXP:<AcT-type>[,<Requested_eDRX_value>[,<NW-provided_eDRX_value>[,<Paging_time_window>]]]

When +CEDRXS=3 is sending, eDRX shall not be used and all the previous parameters set shall be removed to restore them to default values.

The query instruction is used for the current set value of all the defined <act-type>.



AT+CEDRXS=<mode>[,<AcT-type>[,<Requested_eDRX_value>]]

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<mode> integer data to enable/disable eDRX. This parameter can be applied to all the specified types of access technology,

0 Disable eDRX

1 Enable eDRX

2 Enable eDRX and enable active submission in the following formats:

+CEDRXP:<AcT-type>[,<Requested_eDRX_value>[,<NW-provided_eDRX_value>[,<Paging_time_window>]]]

3 Disable the use of eDRX and discard all parameters for eDRX or, if available, reset to the manufacturer specific default values.

<AcT-type>: integer data, access technology type

0 Access technology without using eDRX. The parameter is only used to actively report result codes.

5 E-UTRAN (NB-S1 mode)

<Requested_eDRX_value>: character type parameter occupying the bit1-bit4 of eDRX. The parameter can be applied to A/Gb mode, lu mode, or S1 mode. In different modes, the parameter is defined as follows:

In A/Gb mode (EC-GSM-IoT/GSM):

Bit 4 ~bit1 GERAN eDRX cycle duration. The number of 51-MF in every GERAN eDRX cycle

0000	About 1,88 seconds (Note 1, Note 2)	8
0001	About 3,76 seconds (Note 1, Note 2)	16
0010	About 7,53 seconds (Note 1, Note 2)	32
0011	12,24 seconds (Note 2)	52
0100	24,48 seconds (Note 2)	104
0101	48,96 seconds (Note 2)	208
0110	97,92 seconds (Note 2)	416
0111	195,84 seconds (Note 2)	832
1000	391,68 seconds (Note 2)	1664
1001	783,36 seconds (Note 2)	3328

1010 1566,72 seconds (Note 2) 6656

1011 3133,44 seconds (Note 2) 13312

All values other than those in the list will be automatically equated with the parameter 0000;

Note1: All the cycle parameter values above have been rounded to two decimals;

Note2: All the cycle parameter values can be obtained by the formula: $((3,06 / 13) * (\text{Number of 51-MF}))$

In NB-S1/WB-S1 mode (E-UTRAN):

Bit 4 ~bit1 E-UTRAN eDRX cycle length duration eDRX cycle parameter 'TeDRX'

0000 5,12 seconds (Note 4) Note 3

0001 10,24 seconds (Note 4) 20

0010 20,48 seconds 21

0011 40,96 seconds 22

0100 61,44 seconds (Note 5) 6

0101 81,92 seconds 23

0110 102,4 seconds (Note 5) 10

0111 122,88 seconds (Note 5) 12

1000 143,36 seconds (Note 5) 14

1001 163,84 seconds 24

1010 327,68 seconds 25

1011 655,36 seconds 26

1100 1310,72 seconds 27

1101 2621,44 seconds 28

1110 5242,88 seconds (Note 6) 29

1111 10485,76 seconds (Note 6) 210

All values other than those in the list will be automatically become parameter 0000 by default;

Note3: In E-UTRAN system, when eDRX cycle lasts 5.12S, parameter TeDRX is invalid.

Note4: The value is applied to WB-S1 mode. In NB-S1 mode, it is equivalent to the eDRX parameter not containing the cycle duration parameter.

Note5: The value is applied to WB-S1 mode. In NB-S1 mode, these parameters are automatically equated with 0010.

Note6: The value is applied to the NB-S1 mode.

<NW-provided_eDRX_value>: the eDRX value assigned by the network. The definition of the parameter is equivalent to <Requested_eDRX_value>

<Paging_time_window>: character type parameter, which occupies the bit5-bit8 of eDRX parameter. PTW value (paging time window). In different modes, the parameter is defined as follows:

In NB-S1 mode:

Bit 8~bit5 Paging Time Window length

0000 2,56 seconds

0001 5,12 seconds

0010 7,68 seconds

0011 10,24 seconds

0100 12,8 seconds

0101 15,36 seconds

0110 17,92 seconds

0111 20,48 seconds

1000 23,04 seconds

1001 25,6 seconds

1010	28,16 seconds
1011	30,72 seconds
1100	33,28 seconds
1101	35,84 seconds
1110	38,4 seconds
1111	40,96 seconds



AT+CEDRXS=?

Response

<CR><LF>+CEDRXS: (list of supported <mode>s),(list of supported <AcT-type>s),(list of supported <Requested_eDRX_value>s) <CR><LF>
<CR><LF>OK<CR><LF>



AT+CEDRXS?

Response

[<CR><LF>+CEDRXS: <AcT-type>,<Requested_eDRX_value>
[<CR><LF>+CEDRXS: <AcT-type>,<Requested_eDRX_value>
[...]]]
<CR><LF>OK<CR><LF>



AT+CEDRXS=1,5,"1101"

OK

AT+CEDRXS?

+CEDRXS: 5,"1101"

OK

AT+CEDRXS=?

+CEDRXS: (0-3),(5),("0000"-1111")

OK

8.2. AT+CEDRXRDP eDRX to read dynamic parameters

When eDRX is used for currently registered cell, the instruction is used to return <Requested_eDRX_value>, <NW-provided_eDRX_value> and <Paging_time_window>. If current registered cell does not use eDRX, the setting instruction returns AcT-type=0.



AT+CEDRXRDP Response

```
<CR><LF>+CEDRXRDP: <AcT-type>[,<Requested_eDRX_value>[,<NW-
provided_eDRX_value>[,<Paging_time_window>]]]<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

Parameter

<AcT-type>: integer data, access technology type.

- 0 Access technology without using eDRX. The parameter is only used to actively report result codes.
- 5 E-UTRAN (NB-S1 mode)

<Requested_eDRX_value>: character type parameter occupying the bit1-bit4 of eDRX. The parameter can be applied to A/Gb mode, Iu mode, or S1 mode. Refer to the parameter definition of <Requested_eDRX_value> of +CEDRXS? instruction.

<NW-provided_eDRX_value>: the eDRX value assigned by the network. The definition of the parameter is equivalent to <Requested_eDRX_value>

<Paging_time_window>: character type parameter, which occupies the bit5-bit8 of eDRX parameter. PTS value (paging time window). Refer to the parameter definition of <Paging_time_window> of +CEDRXS? instruction.



AT+CEDRXRDP=? Response

```
<CR><LF>OK<CR><LF>
```



```
AT+CEDRXRDP
CEDRXRDP: 5,"1101","1101","0100"
```

```
OK
```

```
AT+CEDRXRDP=?
```

```
OK
```

8.3. AT*MEDRXCFCG eDRX configuration

The set command controls the setting of the UEs eDRX parameters. The command controls whether the UE wants to apply eDRX or not, as well as the requested eDRX value and requested paging time window value for each specified type of access technology.

The set command also controls the presentation of an unsolicited result code +CEDRXP: <AcT-type>[,<Requested_eDRX_value>[,<NW-provided_eDRX_value>[,<Paging_time_window>]]] when <n>=2 and there is a change in the eDRX parameters provided by the network.

A special form of the command can be given as *MEDRXCFCG=3. In this form, eDRX will be disabled and data for all parameters in the command *MEDRXCFCG will be removed or, if available, set to the manufacturer specific default values.



Execution command

```
AT*MEDRXCFCG=[<mode>[,<AcT-
type>[,<Requested_eDRX_value>[,<Requested_Paging_time_window_value>]]]]
```

Response

- OK
- If error is related to wrong AT syntax:

```
+CME ERROR: <err>
```

Unsolicited result code

- +CEDRXP:<AcT-type>[,<Requested_eDRX_value>[,<NW-provided_eDRX_value>[,<Paging_time_window>]]]

<mode>: integer type, indicates to disable or enable the use of eDRX in the UE. This parameter is applicable to all specified types of access technology, i.e. the most recent setting of

<mode> will take effect for all specified values of <AcT>.

- 0 Disable the use of eDRX
- 1 Enable the use of eDRX
- 2 Enable the use of eDRX and enable the unsolicited result code +CEDRXP: <AcT-type>[,<Requested_eDRX_value>[,<NW-provided_eDRX_value>[,<Paging_time_window>]]]
- 3 Disable the use of eDRX and discard all parameters for eDRX or, if available, reset to the manufacturer specific default values.

<AcT-type>: integer type, indicates the type of access technology. This AT-command is used to specify the relationship between the type of access technology and the requested eDRX value.

- 0 Access technology is not using eDRX. This parameter value is only used in the unsolicited result code.

5 E-UTRAN (NB-S1 mode)

<Requested_eDRX_value>: string type; half a byte in a 4-bit format. The eDRX value refers to bit 4 to 1 of octet 3 of the Extended DRX parameters information element (see sub-clause 10.5.5.32 of 3GPP TS 24.008). For the coding and the value range, see Extended DRX parameters information element in 3GPP TS 24.008 Table 10.5.5.32/3GPP TS 24.008. The default value, if available, is manufacturer specific.

<Requested_Paging_time_window_value>: string type; half a byte in a 4-bit format. The paging time window refers to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see sub-clause 10.5.5.32 of 3GPP TS 24.008). For the coding and the value range, see the Extended RX parameters information element in 3GPP TS 24.008 Table 10.5.5.32/3GPP TS 24.008.

<NW-provided_eDRX_value>: string type; half a byte in a 4-bit format. The eDRX value refers to bit 4 to 1 of octet 3 of the Extended DRX parameters information element (see sub-clause 10.5.5.32 of 3GPP TS 24.008). For the coding and the value range, see Extended DRX parameters information element in 3GPP TS 24.008 Table 10.5.5.32/3GPP TS 24.008.

<Paging_time_window>: string type; half a byte in a 4-bit format. The paging time window refers to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see sub-clause 10.5.5.32 of 3GPP TS 24.008). For the coding and the value range, see the Extended DRX parameters information element in 3GPP TS 24.008 Table 10.5.5.32/3GPP TS 24.008.



Test Command AT*MEDRXCFG=?

Response

- *MEDRXCFG: (list of supported <mode>s),(list of supported <AcT-type>s),(list of supported <Requested_eDRX_value>s),(list of supported <Requested_Paging_time_window_value>s)
- OK



Read command

AT*MEDRXCFG?

Response

[*MEDRXCFG: <AcT-type>,<Requested_eDRX_value>[,<Requested_Paging_time_window_value>]

[<CR><LF>*MEDRXCFG: <AcT-type>,<Requested_eDRX_value>[,<Requested_Paging_time_window_value>]

- [...]]]
- OK

8.4. AT+CPSMS set power saving mode (PSM)

This command is used to set the PSM (power saving mode) mode of the module: enable and disable this mode. This command can be used to set RAU (routing area update) cycle and the time of GPRS READY Timer under GSM network and set TAU (location update) cycle and Active Time value under LTE Cat.M/NB-IOT network. This command will take effect after restart, and the parameters will be saved when power-off

AT+CPSMS=2 instruction is mainly used to disable PSM mode and remove all the parameters previously set to restore them to default values.

The query instruction is mainly used to return current parameter settings;

The help instruction is mainly used to return the range of supported parameters.



AT+CPSMS=[<mode>[,<Requested_Periodic-RAU>[,<Requested_GPRS-READY-timer>[,<Requested_Periodic-TAU>[,<Requested_Active-Time>]]]]]

Response

OK

+CME ERROR: <err>

Defined values

mode

0 Disable PSM

1 Enable PSM

① <Requested_Periodic-RAU>: string type; 8-bit single byte. Request setting RAU cycle (T3412) under GERAN/UTRAN network. The parameter is defined as follows:

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
Unit				Timer value			
0 0 0 – 10 minutes				Bit5 - 1 represents the value of a binary encoded timer.			
0 0 1 – 1 hour							
0 1 0 –10 hours							
0 1 1 –2 seconds							
1 0 0 -30 seconds							
1 0 1 –1 minute							
1 1 0 -320 hours							
1 1 1– T3412 timer invalid							

For example: "01000111": 010 means the unit is 10 hours and 00111 represents decimal number 7. Therefore, the parameter is equal to 70 hours, 25200S;

"01110101": 011 means the unit is 25 and 10101 represents decimal number 21. Therefore, the parameter is equal to 42S;

"10111101": 101 means the unit is 1 minute and 11101 represents decimal number 29. Therefore, the parameter is equal to 29 minutes, 1740S;

Supported parameters (in seconds):

0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58,60,62,90,120,150,180,210,240,270,300,330,360,390,420,450,480,510,540,570,600,630,660,690,720,750,780,810,840,870,900,930,960,1020,1080,1140,1200,1260,1320,1380,1440,1500,1560,1620,1680,1740,1800,1860,2400,3000,3600,4200,4800,5400,6000,6600,7200,7800,8400,9000,9600,10200,10800,11400,12000,12600,13200,13800,14400,15000,15600,16200,16800,17400,18000,18600,21600,25200,28800,32400,36000,39600,43200,46800,50400,54000,57600,61200,64800,68400,72000,75600,79200,82800,86400,90000,93600,97200,100800,104400,108000,111600,115200,118800,122400,126000,129600,133200,136800,140400,144000,147600,151200,154800,158400,162000,165600,169200,172800,176400,180000,183600,187200,190800,194400,198000,201600,205200,208800,212400,216000,219600,223200,226800,230400,234000,237600,241200,244800,248400,252000,255600,259200,262800,266400,270000,273600,277200,280800,284400,288000,291600,295200,298800,302400,306000,309600,313200,316800,320400,324000,327600,331200,334800,338400,342000,345600,349200,352800,356400,360000

② <Requested_GPRS-READY-timer>: string type; 8-bit single byte. Request setting GPRS READY cycle (T3324) under GERAN/UTRAN network. The parameter is defined as follows:

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
Unit				Timer value			
0 0 0 - 2 seconds				Bit5 - 1 represents the value of a binary encoded timer.			
0 0 1 - 1 minute							
0 1 0 - 6 minutes							
1 1 1 - T3324 timer invalid							

For example: "01000011" is equal to 18 minutes.

"01000011": 010 means the unit is 6 minutes and 00011 represents decimal number 3. Therefore, the parameter is equal to 18 minutes 1080S;

"00110101": 001 means the unit is 1 minute and 10101 represents decimal number 21. Therefore, the parameter is equal to 21 minutes 1260S;

"00011101": 000 means the unit is 2 minutes and 11101 represents decimal number 29. Therefore, the parameter is equal to 58 seconds;

Supported parameters (in seconds):

0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58,60,62,120,180,240,300,360,420,480,540,600,660,720,780,840,900,960,1020,1080,1140,1200,1260,1320,1380,1440,1500,1560,1620,1680,1740,1800,1860,2160,2520,2880,3240,3600,3960,4320,4680,5040,5400,5760,6120,6480,6840,7200,7560,7920,8280,8640,9000,9360,9720,10080,10440,10800,11160

Remark: The duration set for <Requested_Periodic-RAU> shall be longer than that of <Requested_GPRS-READY-timer>.

③ <Requested_Periodic-TAU>: string type; 8-bit single byte. Request setting RAU cycle (T3412) under LTE network. The parameter is defined as follows:

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
Unit			Timer value				
0 0 0 – 10 minutes			Bit5 - 1 represents the value of a binary encoded timer.				
0 0 1 -1 hours							
0 1 0 -10 hours							
0 1 1 -2 seconds							
1 0 0 -30 seconds							
1 0 1 –1 minute							
1 1 0 -320 hours							
1 1 1– T3412 timer invalid.							

For example:

"01000111": 010 means the unit is 10 hours and 00011 represents decimal number 7. Therefore, the parameter is equal to 70 hours 25200S;

"01110101": 011 means the unit is 2S and 10101 represents decimal number 21. Therefore, the parameter is equal to 42S;

"10111101": 101 means the unit is 1 minute and 11101 represents decimal number 29. Therefore, the parameter is equal to 29 minutes, 1740S;

Supported parameters (in seconds):

0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58,60,62,90,120,150,180,210,240,270,300,330,360,390,420,450,480,510,540,570,600,630,660,690,720,750,780,810,840,870,900,930,960,1020,1080,1140,1200,1260,1320,1380,1440,1500,1560,1620,1680,1740,1800,1860,2400,3000,3600,4200,4800,5400,6000,6600,7200,7800,8400,9000,9600,10200,10800,11400,12000,12600,13200,13800,14400,15000,15600,16200,16800,17400,18000,18600,21600,25200,28800,32400,36000,39600,43200,46800,50400,54000,57600,61200,64800,68400,72000,75600,79200,82800,86400,90000,93600,97200,100800,104400,108000,111600,144000,180000,216000,252000,288000,324000,360000,396000,432000,468000,504000,540000,576000,612000,648000,684000,720000,756000,792000,828000,864000,900000,936000,972000,1008000,1044000,1080000,1116000,1152000,2304000,3456000,4608000,5760000,6912000,8064000,9216000,10368000,11520000,12672000,13824000,14976000,16128000,17280000,18432000,19584000,20736000,21888000,23040000,24192000,25344000,26496000,27648000,28800000,29952000,31104000,32256000,33408000,34560000,35712000

④ <Requested_Active-Time>: string format, 8 bytes. Request the Active Time value T3324 of the module. The default value is 20 seconds, that is, the module goes into dormant state 20s after reporting TAU to network by default. The parameter is defined as follows:

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
-------	-------	-------	-------	-------	-------	-------	-------

Unit	Timer value
0 0 0 - 2 seconds	Bit5 - 1 represents the value of a binary encoded timer.
0 0 1 –1 minute	
0 1 0 – 6 minutes	
1-1 - T3324 timer is invalid.	
<p>For example: "00100100" is equal to 4 minutes.</p> <p>"00100111": 001 means the unit is 1 minute and 00111 represents decimal number 7. Therefore, the parameter is equal to 7 minutes, 420S;</p> <p>"01010101": 010 means the unit is 6 minutes and 10101 represents decimal number 21. Therefore, the parameter is equal to 126 minutes, 7560S;</p> <p>"00011101": 000 means the unit is 2 seconds and 11101 represents decimal number 29. Therefore, the parameter is equal to 58S;</p>	
<p>Supported parameters (in seconds):</p> <p>0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58,60,62,120,180,240,300,360,420,480,540,600,660,720,780,840,900,960,1020,1080,1140,1200,1260,1320,1380,1440,1500,1560,1620,1680,1740,1800,1860,2160,2520,2880,3240,3600,3960,4320,4680,5040,5400,5760,6120,6480,6840,7200,7560,7920,8280,8640,9000,9360,9720,10080,10440,10800,11160</p>	

Remark: The duration set for <Requested_Periodic-TAU> shall be longer than that of <Requested_Active-Time>.



AT+CPSMS=?

Response

+CPSMS: (list of supported <mode>s),(list of supported <Requested_Periodic-RAU>s),(list of supported <Requested_GPRS-READY-timer>s),(list of supported <Requested_Periodic-TAU>s),(list of supported <Requested_Active-Time>s)

OK



AT+CPSMS?

Response

+CPSMS: <mode>,[<Requested_Periodic-RAU>],[<Requested_GPRS-READY-timer>],[<Requested_Periodic-TAU>],[<Requested_Active-Time>]

OK



AT+CPSMS=1,, "00011000","00001010" //It goes into dormant state 20s after reporting TAU to network every 4 hours in LTE network.

OK

AT+CPSMS=1, "00011000","00001010",, //It goes into dormant state 20s after reporting TAU to network every 4 hours in

GERAN/UTRAN network.

OK

8.5. AT*MNBIOTEVENT enable/disable active report of PSM

This command is used to disable/enable the function of active report PSM status. It will take effect immediately after set, and not saved when the module is powered off.



AT*MNBIOTEVENT=<enable>,<event>

Response

<CR><LF>OK<CR><LF>

Parameter:

<enable>:

0 disable the function of active report(default value)

1 enable the function of active report

When <enable>=1 open the active report function: <CR><LF>*MNBIOTEVENT: <state><CR><LF>

<event>:

1 PSM state

<state>: char type

"ENTER PSM"----the module enters into PSM

"EXIT PSM"----the module quits PSM

when <event>=1, the module will active report PSM status as: *MNBIOTEVENT: "ENTER PSM" or *MNBIOTEVENT: "EXIT PSM".



AT*MNBIOTEVENT=?

Response

<CR><LF>OK<CR><LF>



AT*MNBIOTEVENT?

Response

<CR><LF>OK<CR><LF>



AT*MNBIOTEVENT=1,1

OK

*MNBIOTEVENT: "ENTER PSM" //when the module enters PSM, report as this

8.6. AT#TSLK enable/disable UE enter sleep

This set command will take effect once setting, and won't save the set value after restart the module.



AT#TSLK=<mode>

Response

<CR><LF>OK<CR><LF>

Parameter:

<mode>:

0 enable UE enter light sleep and deep/deeper sleep(default value)

1 disable UE enter light sleep and deep/deeper sleep



AT#TSLK=?

Response

<CR><LF>#TSLK:(0,1)<CR><LF>

<CR><LF>OK<CR><LF>



AT#TSLK?

Response

<CR><LF>#TSLK:<mode><CR><LF>

<CR><LF>OK<CR><LF>

9. PS-DOMAIN COMMANDS

9.1. AT*MCGDEFCONT set the PSD connection settings for PDN connection

This command is used to set the PSD connection settings for PDN connection on power up. In NB-IOT, when you attach to the NB-IOT network on power-on then you must also perform a PDN connection setup. In order to allow this to happen we must store PDN connection settings in NVRAM to be used by the modem during the attach procedure.

Note, that if this command is not entered, there will already be default settings stored in NVRAM at compile time.

Note that this command is similar in syntax to AT+CGDCONT, but without the <cid> parameter, and with additional parameters <username> and <password>.

The set parameters of this command will take effect after restart the module



```
AT*MCGDEFCONT=<PDP_type>[,<APN>[,<username>[,<password>]]]
```

Response

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>+CME ERROR: <err><CR><LF>
```

Parameter

<PDP_type> (Packet Data Protocol type) a string parameter which specifies the type of packet data protocol :

IP Internet Protocol (IETF STD 5)

IPV6 Internet Protocol, version 6 (IETF RFC 2460)

IPV4V6 Virtual <PDP_type> introduced to handle dual IP stack UE capability (see 3GPP TS 24.301).

Non-IP Transfer of Non-IP data to external packet data Network (see 3GPP TS 24.301).

<APN> (Access Point Name) a string parameter that is a logical name that is used to select the GGSN or the external packet data network. If the value is null or omitted, then the subscription value will be requested.

<username> String value. Username for the connection to the service provider

<password> String value. Password for the connection to the service provider



```
AT*MCGDEFCONT=?
```

Response

```
<CR><LF>*MCGDEFCONT: (list of supported <PDP_type>)<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```



```
AT*MCGDEFCONT?
```

Response

```
<CR><LF>*MCGDEFCONT:<PDP_type>,<APN>,<username>,<password><CR><LF>
```

```
<CR><LF>OK<CR><LF>
```



```
AT*MCGDEFCONT=?
```

```
*MCGDEFCONT: ("IP","IPV6","IPV4V6","Non-IP")
```

```
OK
```

```
AT*MCGDEFCONT?
```

```
*MCGDEFCONT: "IP","internet","username","password"
```

```
OK
```

```
AT*MCGDEFCONT="IP","ctnb"
```

```
OK
```

9.2. 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 read command returns the current settings for each defined context.

The test command returns values supported as a compound value. If the MT supports several PDP types, <PDP_type>, the parameter value ranges for each <PDP_type> are returned on a separate line.



AT+CGDCONT=[<cid>[,<PDP_type>[,<APN>[,<PDP_addr>[,<d_comp>[,<h_comp>[,<pd1>[,...[,pdN]]]]]]]]]

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>+CME ERROR: <err><CR><LF>

Parameter

<cid>: (PDP Context Identifier) a numeric parameter which specifies a particular PDP context definition.

<PDP_type>: (Packet Data Protocol type) a string parameter which specifies the type of packet data protocol

IP:Internet Protocol (IETF STD 5)

IPV6:Internet Protocol, version 6 (IETF RFC 2460)

PPP:Point to Point Protocol (IETF STD 51)

IPV4V6:Virtual <PDP_type> introduced to handle dual IP stack UE capability. (See 3GPPTS24.301[83])

<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.

If the value is null or omitted, then the subscription value will be requested.

<PDP_address>:

a string parameter that identifies the MT in the address space applicable to the PDP.

If the value is null or omitted, then a value may be provided by the TE during the PDP startup procedure or, failing that, a dynamic address will be requested.

The allocated address may be read using the +CGPADDR command.

<d_comp>: a numeric parameter that controls PDP data compression

0 - off (default if value is omitted)

1 - on (manufacturer preferred compression)

2 - V.42bis

<h_comp>: a numeric parameter that controls PDP header compression (refer 3GPP TS 04.65)

0 - off (default if value is omitted)

1 - on (manufacturer preferred compression)

2 - RFC1144

3 - RFC2507

4 - RFC3095 (applicable for PDCP only)

<pd1> ... <pdN>: zero to N string parameters whose meanings are specific to the <PDP_type>



AT+CGDCONT?

Response

<CR><LF>+CGDCONT: (range of supported <cid>s),<PDP_type>,,,(list of supported<d_comp>s),(list of supported <h_comp>s)[,(list of supported <pd1>s)[,...[(list of supported <pdN>s)]]]<CR><LF>

[+CGDCONT: (range of supported <cid>s), <PDP_type>,,,(list of supported <d_comp>s),(list of supported <h_comp>s)[,(list of supported <pd1>s)[,...[(list of supported <pdN>s)]]]<CR><LF>

[...]

<CR><LF>OK<CR><LF>



```
AT+CGDCONT?
+CGDCONT: 1,"IP","uninet","0.0.0.0",0,0
```

```
OK
```

9.3. AT+CGCONTRDP PDP Context Read

The execution command returns the relevant information for a primary PDP Context established by the network with the primary context identifier <cid>. If the context cannot be found an ERROR response is returned.

If the UE has dual stack capabilities, two lines of information are returned per <cid>. First one line with the IPv4 parameters followed by one line with the IPv6 parameters.

If the parameter <cid> is omitted, the relevant information for all established PDP contexts are returned.

Note: The dynamic part of the PDP context will only exist if established by the network. And this command can't get the high 64-bit of IPV6 address



```
AT+CGCONTRDP=[<cid>]
```

Response

```
<CR><LF>+CGCONTRDP: <cid>, <bearer_id>, <apn>[, <local address and subnetmask>[,
<gw_addr>[, <DNS_prim_addr>[, <DNS_sec_addr>]]]]<CR><LF>
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>+CME ERROR: <err><CR><LF>
```

Parameter

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

<bearer_id>: a numeric parameter which identifies the bearer, EPS Bearer in EPS and NSAPI in UMTS/GPRS.

<APN>: a string parameter which is a logical name that was used to select the GGSN or the external packet data network.

<local address and subnet mask>: a string parameter which shows the IP Address and subnet mask of the UE. 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.

<gw_addr>: a string parameter which shows the Gateway Address of the UE. The string is given as dot-separated numeric (0-255) parameters.

<DNS_prim_addr>: a string parameter which shows the IP Address of the primary DNS Server.

<DNS_sec_addr>: a string parameter which shows the IP address of the secondary DNS Server.

<IPv4_MTU>: integer type; shows the IPv4 MTU size in octets.

<Non-IP_MTU>: integer type; shows the Non-IP MTU size in octets.

<Serving_PLMN_rate_control_value>: integer type; indicates the maximum number of uplink messages the UE is allowed to send in a 6-minute interval. This refers to octet 3 to 4 of the Serving PLMN rate control IE as specified in 3GPP TS 24.301 sub-clause 9.9.4.28.



```
AT+CGCONTRDP=?
```

Response

```
<CR><LF>+CGCONTRDP: (list of <cid>s associated with active contexts)<CR><LF>
<CR><LF>OK<CR><LF>
```

```
</> AT+CGCONTRDP=1
+CGCONTRDP: 1,5,"ctnb","10.51.205.249.255.255.255.0"
OK
```

9.4. +IP IP address Unsolicited report command

IP address Unsolicited report command.

The module supports establish ps-domain connection automatically. And will report +IP:xxx.xxx.xxx.xxx when PS connection established.

```
<CR><LF>+IP: <IP_Address><CR><LF>
Parameter
<IP_Address>: The IP address assigned by network
```

```
</> *MATREADY: 1

+CFUN: 1

+CPIN: READY

+IP: 192.168.4.6
```

10. HARDWARE COMMANDS

10.1. AT+TRST RESET the Module

This command is used to reset the module. Module will deactivate PDP, release all resources, and then reset.



AT+TRST
Response
OK

10.2. AT+TURNOFF Turn Off the Module

This command is used to turn off the modem. Module will deactivate PDP, release all resources, and then shut down.



AT#TURNOFF
Response
<CR><LF>OK<CR><LF>
<CR><LF>+CPIN: NOT READY<CR><LF>

10.3. AT#TADC Read pin ADC value

This command is used to read the pin voltage of ADC value from modem.



AT#TADC?
Response
<CR><LF>#TADC: (0,1400)<CR><LF>
<CR><LF>OK<CR><LF>



AT#TADC=?
Response
<CR><LF>#TADC: <value><CR><LF>
<CR><LF>OK<CR><LF>

Parameter
<value> :integer type, the range is 0~1400mV, value of the pin ADC, which is expressed in mV.



```
AT#TADC? #TADC:(0,1400)
```

```
OK
```

```
AT#TADC=? #TADC:722
```

```
OK
```

10.4. AT#TCONTLED LED status control

This command is used to enable/disable LED status indication function. This command will take effect once setting, and will be still take effect after restart.

There are four modes of LED status:

1. Power off, no power supply or light/deep sleep mode. In these scenarios, the LED is off
2. Offline mode: A: no network (such as no antenna connection); B: AT+CFUN=0; C: AT+CEREG? Return is not "0,1". In these scenarios, the LED flashes of Duty cycle 50%
3. Online mode: AT+CEREG? Return "0,1". the LED flashes of Duty cycle 10%
4. Data sending: get IP address, and can send data. the LED flashes in 10Hz, of Duty cycle 50%

Note:

1. If there is no SIM card inserted, LED will be in offline status.
2. If the module enters light sleep state, the LED will be all in off status.



```
AT#TCONTLED=<mode>
Response
<CR><LF>OK<CR><LF>
Parameter
<mode>
0: disable LED function (default value)
1: enable LED function
```



```
AT#TCONTLED=?
Response
<CR><LF>#TCONTLED: (0,1)<CR><LF>
<CR><LF>OK<CR><LF>
```



```
AT#TCONTLED?
Response
<CR><LF>#TCONTLED:<mode><CR><LF>
<CR><LF>OK<CR><LF>
```



```
AT#TCONTLED=1
OK
AT#TCONTLED=?
#TCONTLED: (0,1)

OK

AT#TCONTLED?
#TCONTLED:1

OK
```

10.5. AT+EGPIO Operate GPIO

Query GPIO configuration and Change GPIO configuration. This AT command is case sensitive.



Set command
 AT+EGPIO=GPIO_GET:<gpio_sid>~<gpio_eid>
 Response
 • ATCI_CMD_MODE_EXECUTION
 OK
 +EGPIO<gpio_id>, md=<md_val>, <s_dir>, di=<di_val>, do=<do_val>, <s_current_state>,
 <s_pull_type>
 • ERROR



Set command
 AT+EGPIO=GPIO_SET_MODE:<gpio_id>,<mod_val>
 Response
 • ATCI_CMD_MODE_EXECUTION
 OK
 SET GPIO<gpio_id> to mode<md_val> done!
 • ERROR



Set command
 AT+EGPIO=GPIO_SET_DIR:<gpio_id>, <di_val>
 Response
 • ATCI_CMD_MODE_EXECUTION
 OK
 SET GPIO<gpio_id> to <s_dir> done!
 • ERROR



AT+EGPIO=GPIO_SET_PULL:<gpio_id>,<pull_val>
 • ATCI_CMD_MODE_EXECUTION
 OK
 SET GPIO<gpio_id> to <s_pull_type> done!
 • ERROR



Set command
 AT+EGPIO=GPIO_SET_OD:<gpio_id>,<od_val>
 Response
 • ATCI_CMD_MODE_EXECUTION
 OK
 SET GPIO<gpio_id> output <od_val> done!
 • ERROR



Set command
 AT+EGPIO=GPIO_SET_DRV:<gpio_id>,<drv_val>
 Response
 • ATCI_CMD_MODE_EXECUTION
 OK
 SET GPIO<gpio_id> current driving <s_current_state> done!
 • ERROR
 Parameter

- <gpio_id>: integer gpio id number
- <gpio_sid>: integer start gpio id number
- <gpio_eid>: integer end gpio id number
- <md_val>: integer GPIO mode value
- <s_dir>: string "input" or "output"
- <di_val>: integer input direction value; 0 means input, 1 means output
- <od_val>: integer output direction value; 0 means low, 1 means high
- <s_current_stat>: string driving current; "4ma", "8ma", "12ma", "16ma"
- <pull_value>: integer pull value; 0 means pull-up, 1 means pull-down.
- <s_pull_type>: string "NO_PULL", "PU_75K", "PD_75K", "PU_47K", "PD_47K", "PU_23.5K", "PD_23.5K", "PUPD_Err"
- <drv_val>: integer the current driving. 0 - 4ma; 1 - 8ma, 2 - 12ma, 3 - 16ma



Test command

AT+EGPIO=?

Response

- +EGPIO=(GPIO_GET: get all GPIO information,GPIO_SET: set several configuration to one pin,GPIO_SET_MODE: set mode to one pin,GPIO_SET_DIR: set direction to one pin,GPIO_SET_PULL: set pull to one pin,GPIO_SET_OD: set output data to one pin)

OK

- ERROR



+EGPIO examples

```
AT+EGPIO=GPIO_GET:5~10 // get GPIO_5~GPIO_10 status; 0x0d, 0x0a at the end of string
ATCI_CMD_MODE_EXECUTION OK
OK
```

```
+EGPIO05, md=1, input, di=1, do=0, 4ma, NO_PULL
```

```
+EGPIO06, md=0, input, di=1, do=0, 4ma, PU_47K
```

...

```
+EGPIO10, md=0, output, di=0, do=1, 4ma, PU_47K
```

```
AT+EGPIO=GPIO_SET_MODE:22,3 // set GPIO_22 to mode_3; 0x0d, 0x0a at the end of string
ATCI_CMD_MODE_EXECUTION OK
```

OK

SET GPIO22 to mode3 done!

```
AT+EGPIO=GPIO_SET_DIR:3,1 // set GPIO_3 to output direction; 0x0d, 0x0a at the end of string
ATCI_CMD_MODE_EXECUTION OK
```

OK

SET GPIO3 to output done!

```
AT+EGPIO=GPIO_SET_PULL:1,1 // set GPIO_1 to pull-down state; 0x0d, 0x0a at the end of string
ATCI_CMD_MODE_EXECUTION OK
```

OK

SET GPIO1 to PD_75K done!

```
AT+EGPIO=GPIO_SET_OD:11,1 // set GPIO_11 to data-high; 0x0d, 0x0a at the end of string
ATCI_CMD_MODE_EXECUTION OK
```

OK

SET GPIO11 to 1 done!

```
AT+EGPIO=GPIO_SET_DRV:33,3 // set GPIO_33 current to 16mA; 0x0d, 0x0a at the end of
```

```
string
ATCI_CMD_MODE_EXECUTION OK
OK

SET GPIO33 current driving 16ma done!
```

10.6. AT#SYSHALT System Halt

Execution command causes device into halt state, it can be awaked by PowerKey, Wakeup_pin, Reset pin.



AT#SYSHALT Execution command causes device into halt state, it can be awaked by PowerKey, Wakeup_pin, Reset pin.

Response
<CR><LF>OK<CR><LF>



AT#SYSHALT=? Test command reports supported range of values for all parameters.

Response
<CR><LF>#SYSHALT: (0-1)<CR><LF><CR><LF>
<CR><LF>OK<CR><LF>

Parameter
0-1: no meaning at the moment

10.7. AT#SHDN Software Shut Down

Execution command causes device detach from the network and shut down.



AT#SHDN
Before definitive shut down an OK response is returned.



AT#SHDN? Read command has the same behaviour as Execution command.
<CR><LF>OK<CR><LF>



AT#SHDN=? Test command returns the OK result code.
<CR><LF>OK<CR><LF>



Note:

- After the issuing of this command any previous activity is terminated and the device will not respond to any further command.
- To turn it on again Hardware pin ON/OFF must be tied low.

10.8. AT+CBC Query Power Supply Voltage

The command is used to query the voltage value of power supply.



AT+CBC?
Response
<CR><LF>+CBC: <value><CR><LF>
<CR><LF>OK<CR><LF>
Parameter
<Value>:integer, VBAT means Power Supply Voltage, the unit is mV



Example
AT+CBC?
+CBC: 3300

OK

10.9. AT#I2CWR - Write to I2C



AT#I2CWR=<sdaPin>,<sclPin>,<deviceId>,<registerId>,<len>

This command is used to Send Data to an I2C peripheral connected to module GPIOs

Parameters:

<sdaPin >: GPIO number for SDA . Valid range is “any input/output pin”, it can be 1 or 3 (see Test Command.)

<sclPin>: GPIO number to be used for SCL. Valid range is “any output pin” , it can be 2 or 4 (see Test Command). If <sdaPin >=1,<sclPin> need to be 2; else if <sdaPin >=3,<sclPin> need to be 4

<deviceId>: address of the I2C device, with the LSB, used for read/write command. It doesn't matter if the LSB is set to 0 or to 1. 10 bit addressing supported.

Value has to be written in hexadecimal form (without 0x).

<registerId>: Register to write data to , range 0..255. Value has to be written in hexadecimal form (without 0x).

<len>: number of data to send. Valid range is 1-254.

Response:

The module responds to the command with the prompt '>' and awaits for the data to send.

To complete the operation send Ctrl-Z char (0x1A hex); to exit without writing the message send ESC char (0x1B hex).

Data shall be written in Hexadecimal Form.

If data are successfully sent, then the response is OK.

If data sending fails for some reason, an error code is reported.

Example if CheckAck is set and no Ack signal was received on the I2C bus

E.g.

```
AT#I2CWR=1,2,20,10,14
```

```
> 00112233445566778899AABBCCDD<ctrl-z>
```

```
OK
```

Set GPIO1 as SDA, GPIO2 as SCL;

Device I2C address is 0x20;

0x10 is the address of the first register where to write I2C data;

14 data bytes will be written since register 0x10

NOTE: At the end of the execution GPIO will be restored to the original setting (check AT#GPIO Command)

NOTE: device address, register address where to read from\ write to, and data bytes have to be written in hexadecimal form without 0x.



AT#I2CWR=?

Test command reports the supported list of currently available <service>s.

Response

```
#I2CWR:I2C_MASTER_1(SDA:3 SCL:4)
```

```
#I2CWR:I2C_MASTER_2(SDA:1 SCL:2)
```

```
#I2CWR:REGISTERID (0-255)
```

```
#I2CWR:LEN (1-254)
```

```
OK
```

10.10. AT#I2CRD - Read to I2C



AT#I2CRD=<sdaPin>,<sclPin>,<deviceId>,<registerId>,<len>
 This command is used to Receive Data from an I2C peripheral connected to module GPIOs
 <sdaPin>: GPIO number for SDA . Valid range is “any input/output pin” (see Test Command).
 You can refer to <sdaPin> of #I2CWR command for detail.
 <sclPin>: GPIO number to be used for SCL. Valid range is “any output pin” (see Command Test).
 You can refer to <sdaPin> of #I2CWR command for detail.
 <deviceId>: address of the I2C device, with the LSB, used for read\write command. It doesn't
 matter if the LSB is set to 0 or to 1. 10 bit addressing supported.
 Value has to be written in hexadecimal form (without 0x before).
 <registerId>: Register to read data from, range 0..255.
 Value has to be written in hexadecimal form (without 0x before).
 <len>: number of data to receive. Valid range is 1-254.

Data Read from I2C will be dumped in Hex:
 E.g.
 AT#I2CRD=2,3,20,10,12
 #I2CRD: 00112233445566778899AABBCC
 OK

NOTE: If data requested are more than data available in the device, dummy data (normally 0x00
 or 0xff) will be dumped.

NOTE: At the end of the execution GPIO will be restored to the original setting (check AT#GPIO
 Command)

NOTE: device address, register address where to read from\ write to, and date bytes have to be
 written in hexadecimal form without 0x.



AT#I2CRD=? Test command reports the supported list of currently available <service>.
 Response
 #I2CRD:I2C_MASTER_2(SDA:1 SCL:2)
 #I2CRD:I2C_MASTER_1(SDA:3 SCL:4)
 #I2CRD:REGISTERID(0-255)
 #I2CRD:LEN(1-254)
 OK

10.11. AT#SPIOOPEN Initialize modem serial port with SPI protocol



AT#SPIOOPEN=<ID>,<speed>,<mode>
 This command initializes the provided modem serial port for SPI protocol.
 Response
 <CR><LF>OK<CR><LF>
 or
 <CR><LF>ERROR<CR><LF>
 Parameters:
 <ID> - available device id; Just supported value is 0
 <speed> - supported speed value:
 1 for 1 Mhz
 2 for 3 Mhz
 3 for 6 Mhz
 4 for 12 Mhz
 <mode> - CPOL CPH setting:
 0 Clock signal is active high and data is sampled in rising edge.
 1 Clock signal is active high and data is sampled in falling edge.
 2 Clock signal is active low and data is sampled in rising edge.
 3 Clock signal is active low and data is sampled in falling edge
 Example
 AT#SPIOOPEN=0,1,0
 OK



AT#SPIOOPEN=? Test command reports available values for parameters <ID>, <speed> and <mode>.
 Response
 (0),(1-4),(0-3)
 OK



Note: Only for NE310H2-W1

10.12. AT#SPICLOSE De-initialize modem serial port for SPI protocol



AT#SPICLOSE=<ID>

This command de-initializes the provided modem serial port for the SPI protocol .

Parameters:

<ID> - available device id; Just supported value is 0

Note: returns OK if de-initialization complete, ERROR otherwise
1 Clock signal is active high and data is sampled in falling edge.

2 Clock signal is active low and data is sampled in rising edge.

3 Clock signal is active low and data is sampled in falling edge

Example

```
AT#SPIOOPEN=0,1,0
```

```
OK
```



AT#SPICLOSE=? Test command reports available values for parameter <ID>.

Response

```
AT#SPICLOSE=?
```

```
<ID>
```

```
[0]
```

```
OK
```



Note: Only for NE310H2-W1

10.13. AT#SPIRW Write a buffer to the SPI and prints the read data



AT#SPIRW=[<length>] [ID] [rx_length] This command writes a buffer to the SPI and prints the read data.

Parameters:

<length> - buffer length : MIN 1 byte, MAX 128 bytes

The module responds to the command with the prompt <greater_than><space> and waits for the data to send.

When <length> bytes have been sent, operation is automatically completed.

If data are successfully sent, the module answer with the bytes read on the SPI RX channel.

The received data can be read on the AT console, the amount of printed data is the same received that is the length of the sent data.

<ID> - available device id; Just supported value is 0 when It is omit

<rx_length>- buffer length of receive data

When it is omit. The rx_length eque to length of send data. The max length is 128,min length is 1.

Note: the modem serial port on which the SPI data must be sent has to be initialized previously with a AT#SPIOOPEN command, otherwise it will return ERROR.

The send data and received data will display with ASCII code.



AT#SPIRW=? Test command reports available value for parameter <length>.

Response

#SPIRW: (1-128)

OK



Note: Only for NE310H2-W1

10.14. AT#GPIO General Purpose Input/Output Pin Control



AT#GPIO=[<pin>,<mode>[,<dir>[,<save]]]

Execution command sets the value of the general purpose output pin GPIO<pin> according to <dir> and <mode> parameter.

Not all configurations for the three parameters are valid.

Parameters:

<pin> - GPIO pin number; supported range is from 1~6

<mode> - its meaning depends on <dir> setting:

<dir> - GPIO pin direction

<mode> <dir>	0	1	2	3	4
0(pin direction is INPUT)	/	/	Reports the read value of the input pin	INPUT, enable Pull-Up	INPU enabl Down
1(pin direction is OUTPUT)	Output low level	Output high level	Reports the logic value of output pin	/	/
2(pin direction is ALTERNATE FUNCTION 1)	/	/	/	/	/
3(pin is set to PULL DOWN, obsolete, just for compatibility with older versions. Now use <mode> to set pull down)	/	/	/	/	/
4(pin direction is ALTERNATE FUNCTION 2)	/	/	/	/	/
5(pin direction is ALTERNATE FUNCTION 3)	/	/	/	/	/
6(pin direction is ALTERNATE FUNCTION 4)	/	/	/	/	/

"/" in the table above stands for no meaning.

Note1: 2nd ALTERNATE FUNCTION of GPIO1~4 is SIMIN function. ALTERNATE FUNCTION of GPIO5~6 is SIMIN function.

For all the ALTERNATE FUNCTIONS of GPIO, you can refer to the table below.

Note2: when <mode>=2 (and <dir> is omitted) the command reports the direction and value of pin GPIO<pin> in the format: #GPIO: <dir>,<stat>

where:

<dir> - current direction setting for the GPIO<pin>

<stat> logic value read from pin GPIO<pin> in the case the pin <dir> is set to input; logic value present in output of the pin GPIO<pin> in the case the pin <dir> is currently set to output; no meaning value for the pin GPIO<pin> in the case the pin <dir> is set to alternate function or Tristate pull down

ALTERNATE FUNCTION list:

NE310 I/O (Default)	ALTERNATE FUNCTION 1	ALTERNATE FUNCTION 2	ALTERNATE FUNCTION 3	ALTERNATE FUNCTION 4
GPIO_1	I2C2_SDA	SIM_IN*	DTR*	RING*
GPIO_2	I2C2_SCL	SIM_IN	RING*	DTR*
GPIO_3	I2C1_SDA	SIM_IN	DTR	RING
GPIO_4	I2C1_SCL	SIM_IN	RING	DTR
GPIO_5	SIM_IN*	DTR	RING	/
GPIO_6	SIM_IN	RING	DTR	/

Note: the pin with "*" stands for you can choose any alternate available GPIO for this function, but when you choose this GPIO, other GPIO(s) can't be used as this function. For example, in column of ALTERNATE FUNCTION 1, the RING function can be set by GPIO_1/GPIO_3, if you use GPIO_1 for RING function, you can't use GPIO_3 for RING.

<save> - GPIO pin save configuration

0 - pin configuration is not saved

1 - pin configuration is saved

Note: Tristate pull down/ pull up settings are available only on some products and GPIO. In case they are not available, automatically the setting is reverted to INPUT. Check the product HW user guide to verify if pull down/ pull up settings are available and if the pull down is the default at system start-up AT#GPIO?



AT#GPIO? Read command reports the read direction and value of all GPIO pins, in the format:
Response

```
#GPIO: <dir>,<stat>[<CR><LF>#GPIO: <dir>,<stat>[...]]
```

OK

Or

OK

where

<dir> - as seen before

<stat> - as seen before



AT#GPIO=? Test command reports the supported range of values of the command parameters <pin>, <mode> and <dir>.

Response

```
pin number(1-6),mode(0-4),dir(0-4),save(0-1)
```

OK



Example

```
AT#GPIO=3,0,1
```

OK

```
AT#GPIO=3,2
```

```
#GPIO: 1,0
```

OK

```
AT#GPIO=4,1,1
```

OK

```
AT#GPIO=5,0,0
```

OK

```
AT#GPIO=6,2
```

```
#GPIO: 0,1
```

OK

```
AT#GPIO=3,0,1,1
```

OK

10.15. AT#E2RI Event Ring Indicator

Set command enables/disables the Ring Indicator pin response to one or more events. If an event has been enabled, a negative going pulse is generated when event happens. The duration of this pulse is determined by the value of <duration>..



AT#E2RI=<event_mask>,<duration><CR>

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<event_mask> : (Only support parameter 40 yet)

0 – disables all events

hexadecimal number representing the list of events:

1 – Power Saving Mode

2 – Socket Listen

4 – OTA firmware upgrade (NOT SUPPORT)

8 – MT SMS has been received

10 – +CREG will change status

20 – +CGREG will change status

40 – #QSS become 2 (SIM INSERTED and PIN UNLOCKED)

80 – MO SMS has been delivered

100 – Jamming Detection & Reporting (JDR) (NOT SUPPORT)

The hexadecimal number is actually a bit mask, where each bit, when set/not set, indicates that the corresponding event has been enabled/disabled.

<duration> :

50..1150 - the duration in ms of the pulse generated

Note: The values set by the command are stored in the profile extended section and they don't depend on the specific AT instance.



AT#E2RI?

Read command reports a line for each event and the duration in ms of the pulse generated, in the format:

Response

<CR><LF>#E2RI: <event_mask><CR><LF>

<CR><LF>Event(Hex):Duration(Dec)

<CR><LF>1: <duration>

<CR><LF>2: <duration>

<CR><LF>4: <duration>

<CR><LF>8: <duration>

<CR><LF>10: <duration>

<CR><LF>20: <duration>

<CR><LF>40: <duration>

<CR><LF>80: <duration>

<CR><LF>100: <duration><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>



AT#E2RI=?

Test command returns supported values of parameters <event_mask> and <duration>

11. SNTP NETWORK COMMANDS

11.1. AT+ESNTPSTART Start to query network time

Start SNTP, it will start a task and query current network time on server

Note: If you do not stop SNTP (AT + ESNTPSTOP), the SNTP task will run all the time and report the latest network time obtained every hour. After obtaining the current network time, the module time will be synchronized. At this time, you can use the "AT+CCLK?" Command to query the module time and see if it works.



AT+ESNTPSTART=<url>

Response

<CR><LF>OK<CR><LF>

or

<CR><LF> ERROR<CR><LF>

Parameter

<url>The domain name need resolution(need Double quotation marks)



Example

AT+ESNTPSTART="ntp1.aliyun.com" ----- start SNTP querying from server ntp1.aliyun.com
OK

+ESNTP:19,10,15,3,26,27,14511 ----- get current time successfully

11.2. AT+ESNTPSTOP Stop to query network time

Start SNTP, it will start a task and query current network time on server



AT+ESNTPSTOP

Response

<CR><LF>OK<CR><LF>

or

<CR><LF> ERROR<CR><LF>

11.3. +ESNTP received network time

Received network time.

<CR><LF>+ESNTP:<year>,<month>,<day>,<hour>,<minute>,<second>,<millisecond><CR><LF>



AT+ESNTPSTATRT="115.28.122.198" ----- start SNTP querying from server 115.28.122.198
OK

+ESNTP:17,11,01,10,15,04,100 ----- get current time successfully

AT+ESNTPSTATRT="115.28.122.198" ----- start SNTP querying from server 115.28.122.198
OK

----- wait for alog time but doesn't receive the network time indication
AT+ESNTPSTOP ----- interrupt SNTP before it return current time
OK

12. SMS COMMANDS

12.1. AT+CSMS Select Message Service



AT+CSMS=<service>

Response

<CR><LF>+CSMS: <mt>,<mo>,<bm><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>+CMS ERROR: <err><CR><LF>

Parameter

<service>:

0 3GPP 23.040 and 23.041.

1 3GPP 23.040 and 23.041, with a requirement that a message routed directly to TE should be acknowledged via +CNMA

128 SMS PDU mode - TPDU only used for sending/receiving SMSs.

<mt> for mobile terminated messages,

0 type not supported

1 type supported

<mo> for mobile originated messages

0 type not supported

1 type supported

<bm> for broadcast type messages

type not supported

Note

< bm> message type is not supported



AT+CSMS=?

Response

<CR><LF>+CSMS: (list of supported <service>s)<CR><LF>

<CR><LF>OK<CR><LF>



AT+CSMS?

Response

<CR><LF>+CSMS: <service>,<mt>,<mo>,<bm><CR><LF>

<CR><LF>OK<CR><LF>



```
//CSMS:1
```

```
AT+CSMS=1
```

```
+CSMS: 1,1,0
```

```
OK
```

```
AT+CMGS="18891006239"
```

```

> this
+CMGS: 248,"12/11/05,14:45:39+32"

OK

//CSMS:0
AT+CSMS=0
+CSMS: 1,1,0

OK
AT+CMGS="18891006239"

> this
+CMGS: 249

OK

```

12.2. AT+CMGF Message Format

Set command selects input and output format of messages to use. This AT command can be enabled with FEA_ATC_SMS.



```

AT+CMGF=[<mode>]
Response
<CR><LF>OK<CR><LF>
Parameter
<mode>:
0      PDU mode
1      Text mode

```



```

AT+CMGF=?      Test command returns supported modes as a compound value.
Response
<CR><LF>+CMGF: (list of supported <mode>s) <CR><LF>
<CR><LF>OK<CR><LF>

```



```

AT+CMGF?      Read command reports the current value of the parameter <mode>.
Response
<CR><LF>+CMGF: <mode><CR><LF>
<CR><LF>OK<CR><LF>

```



```

AT+CMGF=?
+CMGF: (0-1)

OK
AT+CMGF?
+CMGF: 0

OK

```

12.3. AT+CSCA Service Centre Address

Set command sets the Service Center Address to be used for module originated SMS transmissions. In text mode, setting is used by send and write commands. In PDU mode, setting is used by the same commands, but only when the length of the SMSC address coded into <pdu> parameter equals zero. Note: The command writes the parameters in NON-VOLATILE memory.

Read command reports the current value of the SCA.

Test command returns the OK result code.



AT+CSCA=<sca>[,<tosca>]

Response

<CR><LF>OK<CR><LF>

Parameter

<sca>:

3GPP 24.011 RP SC 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 (specified by +CSCS); type of address given by <tosca>

<tosca>:

Service center address format 3GPP 24.011 RP SC address Type-of-Address octet in integer format (default refer <toda>)



AT+CSCA=?

Response

<CR><LF>OK<CR><LF>



AT+CSCA?

Response

<CR><LF>+CSCA: <sca>,<tosca><CR><LF>

<CR><LF>OK<CR><LF>



AT+CSCA?

+CSCA: "+390123456789",145

OK

12.4. AT+CSMP Set Text Mode Parameters

Set 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 (AT+CMGF=1).

It is possible to set the validity period starting from when the SM is received by the SMSC (<vp> is in range 0...255) or define the absolute time of the validity period termination (<vp> is a string). If TA supports the EVPF, see 3GPP TS 23.040, it shall be given as a hexadecimal coded string (refer e.g. <pdu>) with double quotes.



AT+CSMP=[<fo>,<vp>,<pid>,<dc>]]]]

Response

<CR><LF>OK<CR><LF>

Parameter

<fo>: first octet of 3GPP 23.040 SMS-DELIVER, SMS-SUBMIT in integer format. The following fields can be modified:

Bit7: RP: Reply Path,

0—no set;

1—set

Bit6: UDHI, User Data Header Information,

0—no User Data Header Information

1—include User Data Header Information (used by long SMS)

Bit5: SRR, Status Report Request

1— need Status Report

0— no need Status Report

Bit4-bit3: VPF: Validity Period Format

00—Not present;

01—reserved

10—integer (standard)

11—Semi-Octet Represented

Bit2: RD : Reject Duplicates

0—receipt duplicates

1—Reject Duplicates

Bit1-bit0: MTI: Message Type Indicator

00—Deliver

01—Submit

02-SMS COMMAND and SMS STATUS REPORT.

Default value is 17 (SMS-SUBMIT and Validity Period in relative format). When concatenated

SMS is supported by MT2625, attempts to change the following fields from the default will

produce an ERROR:

TP-User-Data-Header-Indicator (bit 6) – the UDHI field is used for concatenated SMSs and is set by the Background Layer where appropriate.

<vp>: depending on SMS-SUBMIT <fo> setting: 3GPP 23.040 [3] TP-Validity-Period either in integer format (default 167)

<pid>: the default value is 0. 3GPP23.040 TP-Protocol-Identifier in integer format.

<dc>: depending on the command or result code: 3GPP 23.038 [2] SMS Data Coding Scheme (default 0)



AT+CSMP? Read command reports the current setting in the format

Response

<CR><LF>+CSMP: <fo>,<vp>,<pid>,<dc><CR><LF>

<CR><LF>OK<CR><LF>



AT+CSMP? Test command returns the OK result code.

Response

<CR><LF>OK<CR><LF>



```
AT+CSMP=17,169,0,0
OK
AT+CSMP=25,169,0,0
ERROR
```

12.5. AT+CSAS Save Settings



AT+CSAS[=<profile>]

Response

TA saves current message service settings for +CMGF, and +CSDH, to a non-volatile memory.

```
<CR><LF>OK<CR><LF>
```

If error is related to ME functionality:

```
<CR><LF>+CMS ERROR:<err><CR><LF>
```

Parameters

<profile>

0 manufacturer specific profile number where settings are to be stored



AT+CSAS=?

Response

```
<CR><LF>+CSAS: list of supported <profile>s<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

12.6. AT+CRES Restore Settings



AT+CRES[=<profile>]

TA restores SMS settings for +CMGF, +CNMI, and +CSDH from non-volatile memory to active memory.

Response

```
<CR><LF>OK<CR><LF>
```

If error is related to ME functionality:

```
<CR><LF>+CMS ERROR:<err><CR><LF>
```

Parameters

<profile>

0 manufacturer specific profile number to store the settings



AT+CRES=?

Response

<CR><LF>+CRES: list of supported <profile>s<CR><LF>

<CR><LF>OK<CR><LF>

12.7. AT+CNMI New Message Indications to TE

Set command selects the procedure, how receiving of new messages from the network is indicated to the DTE.

If ME does not support requested item (although TA does), final result code +CMS ERROR: <err> is returned. See chapter Message Service Failure Result Code for a list of <err> values.

Test command gives the settings supported by the TA as compound values.



AT+CNMI=[<mode>[,<mt>[,<bm>[,<ds>[,<bfr>]]]]]

Response

<CR><LF>OK<CR><LF>

<CR><LF>+CMS ERROR: <err><CR><LF>

Parameter

<mode>- controls the processing of unsolicited result codes specified within this command

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.

3 Forward unsolicited result codes directly to the TE. TA-TE link specific in band technique used to embed result codes and data when TA is in on-line data mode.

<mt> - (the rules for storing received SMSs depend on its data coding scheme (refer 3GPP 23.038 [2])):

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: <mem>,<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><pdu> (PDU mode enabled); or

+CMT: <oa>,<alpha>,<scts>[,<toa>,<fo>,<pid>,<dc>,<sca>,<tosca>,<length>]<data> (text mode enabled; about parameters in italics, refer command Show Text Mode Parameters

+CSDH). Class 2 messages result in indication as defined in <mt>=1.

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> - (the rules for storing received CBMs depend on its data coding scheme (refer 3GPP 23.038 [2]), the setting of Select CBM Types (+CSCB) and this value):

0 No CBM indications are routed to the TE (default)

<ds>-for SMS-STATUS-REPORTs:

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 SMS status reports are stored and indication of memory location routed to TE using unsolicited result +CDSI: "SR",<index>

<bfr>:

0 TA buffer of unsolicited result codes defined within this command is flushed to the TE when <mode> 1...3 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...3 is entered.



AT+CNMI=?

Response

<CR><LF>+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) <CR><LF>

```
<CR><LF>OK<CR><LF>
```



AT+CNMI?

Response

```
<CR><LF>+CNMI: <mode>,<mt>,<bm>,<ds>,<bfr><CR><LF>
<CR><LF>OK<CR><LF>
```



AT+CNMI=2,1

OK

12.8. AT+CMGL List Messages



AT+CMGL[=<stat>]

Parameters

If text mode:

<stat> "REC UNREAD" Received unread messages (default)

"REC READ" Received read messages

"STO UNSENT" Stored unsent messages

"STO SENT" Stored sent messages

"ALL" All messages

If PDU mode:

<stat> 0 Received unread messages (default)

1 Received read messages

2 Stored unsent messages

3 Stored sent messages

4 All messages

Response

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

If text mode (+CMGF=1) and command successful:

for SMS-SUBMITs and/or SMS-DELIVERs:

```
+CMGL: <index>,<stat>,<oa/da>,[<alpha>],[<scts>],[<tooa/toda>,<length>]<CR><LF><data>[<CR><LF>
```

```
+CMGL: <index>,<stat>,<da/oa>,[<alpha>],[<scts>],[<tooa/toda>,<length>]<CR><LF><data>[...]]
```

OK

If SMS-STATUS-REPORT and text mode:

```
+CMGL: <index>,<stat>,<fo>,<mr>,[<ra>],[<tora>],[<scts>,<dt>,<st>
```

```
[<CR><LF> +CMGL: <index>,<stat>,<fo>,<mr>,[<ra>],[<tora>],[<scts>,<dt>,<st>
```

[...]

OK

If PDU mode (+CMGF=0) and command successful:

+CMGL: <index>,<stat>,<alpha>,<length><CR><LF><pdu>

[<CR><LF>+CMGL: <index>,<stat>,<alpha>,<length><CR><LF><pdu>

[...]

OK

If error is related to ME functionality:

+CMS ERROR: <err>

Parameters

<alpha> string type alphanumeric representation of <da> or <oa> corresponding to the entry found in MT phonebook; implementation of this feature is manufacturer specific

<da> 3GPP 23.040 TP-Destination-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted to characters; type of address given by < toda>

<data>

In the case of SMS: 3GPP 23.040 TP-User-Data in text mode responses;

format:

o if < dcs> indicates that 3GPP 23.038 default alphabet is used and <fo> indicates that 3GPP 23.040 TP-User-Data-Header-Indication is not set: ME/TA converts GSM alphabet into current TE character set according to rules of Annex A

o if < dcs> indicates that 8-bit or UCS2 data coding scheme is used, or <fo> indicates that 3GPP 23.040 TP-User-Data-Header-Indication is set: ME/TA converts each 8-bit octet into two IRA character long hexadecimal number (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65))

In the case of CBS: 3GPP 23.041 CBM Content of Message in text mode responses;

format:

o if < dcs> indicates that 3GPP 23.038 default alphabet is used:

ME/TA converts GSM alphabet into current TE character set according to rules of Annex A

o 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 number

<length> integer type value indicating in the text mode (+CMGF=1) the length of the message body <data> (or < cdata>) in characters; or in PDU mode (+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)

<index> integer type; value in the range of location numbers supported by the associated memory

<oa> 3GPP 23.040 TP-Originating-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted to characters; type of address given by < tooa>

<pdu> In the case of SMS: 3GPP 24.011 SC address followed by 3GPP 23.040 TPDU in hexadecimal format: ME/TA converts each octet of TP data unit into two IRA character long hexadecimal number (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)). In the case of CBS: 3GPP 23.041 TPDU in hexadecimal format.

<scts> 3GPP 23.040 TP-Service-Center-Time-Stamp in time-string format (refer < dt>)

<toda> 3GPP 24.011 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)

<tooa> 3GPP 24.011 TP-Originating-Address Type-of-Address octet in integer format (default refer < toda>)

<fo> First byte of SMS-DELIVER, SMS-SUBMIT or SMS-STATUS-REPORT in integer format.

<mr> Message reference. Integer format.

<ra> Recipient address. String type.

<tor> Type of address of <ra>.

145: International number

129: National number

<scts> Service centre time stamp. String format: "yy/MM/dd,hh:mm:ss+/-zz"
(Year/Month/Dat,Hour:Seconds+/TimeZone)

<dt> Discharge time. String format: "yy/MM/dd,hh:mm:ss+/-zz"
(Year/Month/Dat,Hour:Seconds+/TimeZone)

<st> Status of an SMS-STATUS-REPORT. Integer format.

Note

- Read of SMS-SRs when <mem1> is set to "SR" can read SIM memory
- A concatenated SMS is stored on the SIM as number of smaller SMSs. (This is not possible when MMI is present).
- When the MT2625 software does not have concatenated SMS handling in the modem enabled, the maximum text SMS length is restricted depending on the data coding scheme (160 for 7-bit, 140 for 8-bit, 80 for 16-bit).



AT+CMGL=?

Response

<CR><LF>+CMGL: list of supported <stat>s<CR><LF>

<CR><LF>OK<CR><LF>

12.9. AT+CMGR Read Message



AT+CMGR=<index>

Parameters

<index> integer type; value in the range of location numbers supported by the associated memory

Response

TA returns SMS message with location value <index> from message storage <mem1> to the TE. If status of the message is 'received unread', status in the storage changes to 'received read'.

If text mode (+CMGF=1) and command successful:

o for SMS-DELIVER:

+CMGR: <stat>,<oa>,[<alpha>],<scts>[,<tooa>,<fo>,<pid>,<dcs>,<sca>,<tosca>,<length>]<CR><LF>

<data>

OK

o for SMS-SUBMIT:

+CMGR: <stat>,<da>,[<alpha>][,<toda>,<fo>,<pid>,<dcs>,<vp>],<sca>,<tosca>,<length>]<CR><LF>

<data>

OK

□ If SMS-STATUS-REPORT and text mode:

+CMGR: <stat>,<fo>,<mr>,<ra>,<tora>,<scts>,<dt>,<st>

OK

If PDU mode (+CMGF=0) and command successful:

+CMGR: <stat>,<alpha>,<length><CR><LF>

<pdu>

OK

□ If error is related to ME functionality:

+CMS ERROR: <err>

Parameters

<alpha> string type alphanumeric representation of <da> or <oa> corresponding to the entry found in MT phonebook; implementation of this feature is manufacturer specific.

<da> 3GPP 23.040 TP-Destination-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 (specified by +CSCS); type of address given by <toda>

<data> In the case of SMS: 3GPP 23.040 TP-User-Data in text mode responses;

format:

o if <dcsc> indicates that 3GPP 23.038 default alphabet is used and <fo> indicates that 3GPP 23.040 TP-User-Data-Header-Indication is not set:

ME/TA converts GSM alphabet into current TE character set according to rules of Annex A

o if <dcsc> indicates that 8-bit or UCS2 data coding scheme is used, or <fo> indicates that 3GPP 23.040 TP-User-Data-Header-Indication is set: ME/TA converts each 8-bit octet into two IRA character long hexadecimal number (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)).

In the case of CBS: 3GPP 23.041 CBM Content of Message in text mode responses;

format:

o if <dcsc> indicates that 3GPP 23.038 default alphabet is used:

ME/TA converts GSM alphabet into current TE character set according to rules of Annex A

o if <dcsc> indicates that 8-bit or UCS2 data coding scheme is used: ME/TA converts each 8-bit octet into two IRA character long hexadecimal number

<dcsc> depending on the command or result code: 3GPP 23.038 SMS Data Coding Scheme (default 0), or Cell Broadcast Data Coding Scheme in integer format

<fo> depending on the command or result code: first octet of 3GPP 23.040 SMS-DELIVER, SMS-SUBMIT (default 17), SMS-STATUS-REPORT, or SMS-COMMAND (default 2) in integer format

<length> integer type value indicating in the text mode (+CMGF=1) the length of the message body <data> (or <cdata>) in characters; or in PDU mode (+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)

<mid> 3GPP 23.041 CBM Message Identifier in integer format

<oa> 3GPP 23.040 TP-Originating-Address Address-Value field in string format; BCD numbers (or GSM default alphabet characters) are converted characters of the currently selected TE character set (specified by +CSCS); type of address given by <toa>

<pdu> In the case of SMS: 3GPP 24.011 SC address followed by 3GPP 23.040 TPDU in hexadecimal format: ME/TA converts each octet of TP data unit into two IRA character long hexadecimal number (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)). In the case of CBS: 3GPP 23.041 TPDU in hexadecimal format.

<sca> 3GPP 24.011 RP SC 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 (specified by +CSCS);; type of address given by <tosca>

<scts> 3GPP 23.040 TP-Service-Centre-Time-Stamp in time-string format (refer <dt>)

<stat>

- 0 "REC UNREAD" Received unread messages
- 1 "REC READ" Received read messages
- 2 "STO UNSENT" Stored unsent messages
- 3 "STO SENT" Stored sent messages
- 4 "ALL" All messages

<toda> 3GPP 24.011 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)

<tooa> 3GPP 24.011 TP-Originating-Address Type-of-Address octet in integer format (default refer <toda>)

<tosca> 3GPP 24.011 RP SC address Type-of-Address octet in integer format (default refer <toda>)

<vp> depending on SMS-SUBMIT <fo> setting: 3GPP 23.040 TP-Validity-Period either in integer format (default 167) or in time-string format (refer <dt>)

<mr> Message reference. Integer format.

<ra> Recipient address. String type.

<tora> Type of address of <ra>.

- o 145: International number
- o 129: National number

<dt> Discharge time. String format: "yy/MM/dd,hh:mm:ss+/-zz"
(Year/Month/Dat,Hour:Seconds+/TimeZone)

<st> Status of an SMS-STATUS-REPORT. Integer format.

Note

Read of SMS-SRs when <mem1> is set to "SR" can read SIM memory

A concatenated SMS is stored on the SIM as number of smaller SMSs. (This is not possible when MMI is present).

When the MT2625 software does not have concatenated SMS handling in the modem enabled, the maximum text SMS length is restricted depending on the data coding scheme (160 for 7-bit, 140 for 8-bit, 80 for 16-bit).



AT+CMGR=?

Response

<CR><LF>OK<CR><LF>

12.10. AT+CMGS Send Message

Execution command sends message from a TE to the network (SMS-SUBMIT).

Note: Sending can be cancelled by giving <ESC> character (IRA 27).

<ctrl-Z> (IRA 26) must be used to indicate the ending of PDU.



if text mode (+CMGF=1):

AT+CMGS=<da>[,<tda>]<CR>

text is entered<ctrl-Z/ESC>

if text mode (+CMGF=1) and sending successful, Response,:

<CR><LF>+CMGS: <mr><CR><LF>

<CR><LF>OK<CR><LF>

if PDU mode (+CMGF=0):

AT+CMGS=<length><CR>

PDU is given<ctrl-Z/ESC>

if PDU mode (+CMGF=0) and sending successful, Response:

<CR><LF>+CMGS: <mr><CR><LF>

<CR><LF>OK<CR><LF>

if sending fails:

<CR><LF>+CMS ERROR: <err><CR><LF>

Parameter

<da>: 3GPP 23.040 TP-Destination-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 (specified by +CSCS); type of address given by <tda>

<tda>; 3GPP 24.011 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)

<length>: integer type value indicating in the text mode (+CMGF=1) the length of the message body <data> (or <cdata>) in characters; or in PDU mode (+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)

TA transmits SMS message from a TE to the network (SMS-SUBMIT). Message reference value <mr> is returned to the TE on successful message delivery. Value can be used to identify message upon unsolicited delivery status report result code.

<mr>: 3GPP 23.040 TP-Message-Reference in integer format



AT+CMGS=?

Response

OK



AT+CMGS="10086"

```
> 1234
+CMGS: 242
OK
```

12.11. AT+CMSS Send Message from Storage



AT+CMSS=<index>[,<da>[,<toda>]]

Response

TA sends message with location value <index> from message storage <mem2> to the network (SMS-SUBMIT). If new recipient address <da> is given, it shall be used instead of the one stored with the message. Reference value <mr> is returned to the TE on successful message delivery. Values can be used to identify message upon unsolicited delivery status report result code.

If text mode(+CMGF=1) and sending successful:

```
+CMSS: <mr>
OK
```

If PDU mode(+CMGF=0) and sending successful:

```
+CMSS: <mr>
OK
```

If error is related to ME functionality:

```
+CMS ERROR: <err>
```

Parameters

<index> integer type; value in the range of location numbers supported by the associated memory

<da> 3GPP 23.040 TP-Destination-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 (specified by +CSCS); type of address given by <toda>

<toda> 3GPP 24.011 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)

<mr> 3GPP 23.040 TP-Message-Reference in integer format



AT+CMSS=?

Response

```
OK
```

12.12. AT+CMGW Write Message to Memory



If text mode (+CMGF=1):
 AT+CMGW[=<oa/da>[,<toa/toda>[,<stat>[,<scts>]]]]
 <CR> text is entered
 <ctrl-Z/ESC>
 If PDU mode (+CMGF=0):
 AT+CMGW=<length>[,<stat>]<CR>
 PDU is given
 <ctrl-Z/ESC>
 Note: <ESC> quits without sending.
 Response
 TA transmits SMS message (either SMS-DELIVER or SMS-SUBMIT) from TE to memory storage <mem2>. Memory location <index> of the stored message is returned. By default message status will be set to 'stored unsent', but parameter <stat> allows also other status values to be given.

If writing is successful:
 +CMGW: <index>
 OK
 If error is related to ME functionality:
 +CMS ERROR: <err>

Parameters

<oa> 3GPP 23.040 TP-Originating-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 (specified by +CSCS); type of address given by <toa>
 <da> 3GPP 23.040 TP-Destination-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 (specified by +CSCS); type of address given by <toda>
 <toa> 3GPP 24.011 TP-Originating-Address Type-of-Address octet in integer format (default refer <toda>)
 <toda> 3GPP 24.011 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)
 <length> integer type value indicating in the text mode (+CMGF=1) the length of the message body <data> (or <cdata>) in characters; or in PDU mode (+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)
 <pdu> In the case of SMS: 3GPP 24.011 SC address followed by 3GPP 23.040 TPDU in hexadecimal format: ME/TA converts each octet of TP data unit into two IRA character long hexadecimal number (e.g. octet with integer value 42 is presented to TE as two characters 2A (IRA 50 and 65)). In the case of CBS: 3GPP 23.041 TPDU in hexadecimal format.
 <index> Index of message in selected storage <mem2>
 <stat> Status of message when stored in memory
 0 "REC UNREAD" Received unread message
 1 "REC READ" Received read message
 2 "STO UNSENT" Stored unsent message (default)
 3 "STO SENT" Stored sent message
 <scts> MediaTek proprietary. 3GPP 23.040 TP-Service-Centre-Time-Stamp in time-string format String format: "yy/MM/dd,hh:mm:ss+/-zz" (Year/Month/Day,Hour:Seconds+/TimeZone). Field only required when writing SMS-DELIVER.



AT+CMGW=?
 Response
 OK

12.13. AT+CMGD Delete Message



AT+CMGD=<index>[,<DelFlag>]

Response

TA deletes one or several messages from preferred message storage <mem1> location <index> or message groups indicated by <DelFlag>.

OK

If error is related to ME functionality:

+CMS ERROR <err>

Parameters

<index> integer type; value in the range of location numbers supported by the associated memory. This value is only used if <DelFlag> = 0

<DelFlag> message delete mode

- 0 Delete message at location <index> (Default value)
- 1 Delete all read messages
- 2 Delete all READ and SENT messages
- 3 Delete all READ, SENT and UNSENT messages
- 4 Delete all messages



AT+CMGD=?

Response

OK

12.14. AT+CMGC Send Command

Execution command sends a command message from a TE to the network (SMS-COMMAND).



if text mode (+CMGF=1):
 AT+CMGC=<fo>,<ct>[<pid>[,<mn>[,<da>[,<toda>]]]]
 <CR>
 text is entered
 <ctrl-Z/ESC>
 ESC quits without sending
 if PDU mode (+CMGF=0):
 AT+CMGC=<length><CR>
 PDU is given<ctrl-Z/ESC>

Parameter

<fo> first octet of 3GPP 23.040 SMS-COMMAND (default 2) in integer format
 <ct> 3GPP 23.040 TP-Command-Type in integer format (default 0)
 <pid> 3GPP 23.040 TP-Protocol-Identifier in integer format (default 0)
 <mn> 3GPP 23.040 TP-Message-Number in integer format
 <da> 3GPP 23.040 TP-Destination-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 (specified by +CSCS);; type of address given by <toda>
 <toda> 3GPP 24.011 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)
 <length> integer type value indicating in PDU mode (+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)

Response

TA transmits SMS Command message from a TE to the network (SMS-COMMAND). Message reference value <mr> is returned to the TE on successful message delivery. Value can be used to identify message upon unsolicited delivery status report result code.

If text mode (+CMGF=1) and sending successful:

+CMGC: <mr>

OK

If PDU mode (+CMGF=0) and sending successful:

+CMGC: <mr>

OK

If error is related to ME functionality:

+CMS ERROR: <err>



AT+CMGC=?
 Response
 <CR><LF>OK<CR><LF>

12.15. AT+CPMS Preferred Message Storage



AT+CPMS =<mem1>[,<mem2>[,<mem3>]]

Response

TA selects memory storages <mem1>, <mem2> and <mem3> to be used for reading, writing, etc.

+CPMS: <used1>,<total1>,<used2>,<total2>,<used3>,<total3>

OK

If error is related to ME functionality:

+CMS ERROR:<err>

Parameter

<mem1> Messages to be read and deleted from this memory storage

"SM" SIM message storage

"SR" Status Report message storage (EF-SMR if available on SIM). SR in SIM are only associated with SMSs stored on SIM. If EF-SMR not available and MMI is present then status reports are stored in NVRAM. In addition MMI can store status reports in NVRAM as well as ones stored on SIM (EF-SMR file), if available.

<mem2> Messages will be written and sent to this memory storage

"SM" SIM message storage

<mem3> Received messages will be placed in this memory storage if routing to PC is not set (" +CNMI")

"SM" SIM message storage

<usedx> Number of messages currently in <memx>

<totalx> Number of messages storable in <memx>



AT+CPMS=?

Response

+CPMS: (list of supported <mem1>s),(list of supported <mem2>s) ,(list of supported <mem3>s)

OK



AT+CPMS?

Response

+CPMS: <mem1>,<used1>,<total1>,<mem2>,<used2>,<total2>,<mem3>,<used3>,<total3>

OK

If error is related to ME functionality:

+CMS ERROR

12.16. AT+CSDH Show SMS text mode parameters



AT+CSDH=<show>

Response

TA determines whether detailed header information is shown in text mode result codes.

OK

Parameter

<show>

0 do not show header values defined in commands +CSCA and +CSMP (<sca>, <tosca>, <fo>, <vp>, <pid> and <dcsc>) nor <length>, <tda> or <tooa> in +CMT, +CMGL, +CMGR result codes in text mode

1 show the values in result codes



AT+CSDH=?

Response

+CSDH: list of supported <show>s

OK



AT+CSDH?

Response

+CSDH:<show>

OK

12.17. AT+CNMA New Message Acknowledgment



AT+CNMA=<n>[,<length>][<CR>PDU is entered<CTRL-Z/ESC]]

Response

After SMS is routed to the TA (based on message class and +CNMI settings as defined in 27.005), TA sends acknowledgement command to the network. If command is executed but no acknowledgement is expected, or some other error ME related error occurs:

<CR><LF>+CMS ERROR:<err><CR><LF>

If PDU mode(+CMGF=0)

Parameter

<n>:

0 Operates similarly as defined for text mode (if ME doesn't get acknowledgement within required time, ME should respond as specified in 3GPP 24.001, and ME shall automatically disable routing to TE setting both <mt> and <ds> values of CNMI to zero)

1 Send positive acknowledgement to the network with optional PDU message

2 Send negative acknowledgement to the network with optional PDU message

<length> length of the optional PDU message. Integer type



AT+CNMA=?

Response

If PDU mode (+CMGF=0):

<CR><LF>+CMMA: (list of supported <n>s)<CR><LF>

<CR><LF>OK<CR><LF>

12.18. +CMS ERROR Message Service Failure Result Code

This is NOT a command; it is the error response to +Cxxx 3gpp TS 27.005 commands

Final result code +CMS ERROR: <err> indicates an error related to mobile equipment or network. The operation is similar 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.

Command	Response
	<CR><LF>+CMS ERROR: <err><CR><LF>

Parameter

<err> values used by common messaging commands.

The <err> values are reported in the table:

Numeric Format	Meaning
General error:	
0...127	3GPP TS 24.011 Annex E-2 values
128...255	3GPP TS 24.040 sub clause 9.2.3.22 values
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 time-out
340	no +CNMA acknowledgement
500	unknown error

13. TCP/IP COMMANDS

13.1. AT+ESOC CreateTCP/UDP socket

This command is used to CreateTCP/UDP socket



AT+ESOC=<domain>,<type>,<protocol>

Response
 <CR><LF>+ESOC=<socket_id><CR><LF>
 <CR><LF>OK<CR><LF>
 or
 <CR><LF>ERROR<CR><LF>

Parameter

<domain>integer type

1 - IPv4

2 - IPv6

<type>integer type

1 - TCP

2 - UDP

<protocol>integer type

1 - IP

2 - ICMP

3 - UDP_LITE

<socket_id>:The supported range is 0-4



AT+ESOC=1,1,1

+ESOC=0

OK

13.2. AT+ESOB Bind local address and local port

This command used to bind local address and local port. Used chiefly set local port.



Set command

AT+ESOB=<socket_id>,<local_port>[,<local_address>]

Response

- OK
- ERROR

Parameter

- <socket_id> integer socket id, AT+ESOC's response.
- <local_port> integer local port.
- <local_address> string local address. [option]

13.3. AT+ESOCON Connect socket to remote address and port

This command is used to Connect socket to remote address and port



AT+ESOCON=<socket_id>,<remote_port>,<remote_address>

Response

<CR><LF>OK<CR><LF>

Or

<CR><LF>ERROR<CR><LF>

Parameter

<socket_id>integer type, socket id, This value is the response <socket_id> of AT+ESOC.

<remote_port>integer type, Remoteport

<remote_address>string type, remoteaddress



AT+ESOCON?

Response

<CR><LF>+ESOCON=<socket_id>,<type>,<protocol>,<remote_port>,<remote_address><CR><LF>

[<CR><LF>+ESOCON=<socket_id>,<type>,<protocol>,<remote_port>,<remote_address><CR><LF>]

...

<CR><LF>OK<CR><LF>



```
AT+ESOC=1,1,1
+ESOC=2
```

OK

AT+ESOCON=2,8885,"192.168.1.2"

OK

AT+ESOCON?

+ESOCON=1,1,1,8885,"192.168.1.2"

+ESOCON=2,1,1,8885,"192.168.1.2"

OK

13.4. AT+ESOSEND Send data to remote via socket

This command is used to Send data to remote via socket



AT+ESOSEND=<socket_id>,<data_len>,<data>[,<flag>]

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<socket_id>integer type, socket id, This value is the response <socket_id> of AT+ESOC.

<data_len>integer type, length of data, we recommend the length not exceed 512bytes.

<data > data context, ASCII code in HEX format.

<flag>integer type, send flag

1-When the sent data reaches 4K bytes and the buffer is full, the AT port is unavailable. After the signal is restored, the data is automatically sent to network. Then the AT port is available again.

Non 1- default value(0),When the sent data reaches 4K bytes and the buffer is full, the AT port is still available, but the response of AT commands are all ERROR. After the signal is restored, the data is automatically sent to network. Then the AT commands' response are normally.



```
AT+ESOSEND=0,5,3131313131
OK
```

13.5. AT+ESOCL Close socket

This command is used to Close socket



AT+ESOCL=<socket_id>

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<socket_id>: integer type, socket id, This value is the response <socket_id> of AT+ESOC.



```
AT+ESOCL=0
OK
```

13.6. +ESONMI Socket message arrived indicator

This command is used to indicate there is received some data from network

```
<CR><LF>+ESONMI=<socket_id>,<data_len>,<data><CR><LF>
```

Parameter

<socket_id> integer type, socket id, This value is the response <socket_id> of AT+ESOC.

<data_len> integer, length of data.

<data >HEX



```
+ESONMI=0,3,303132
```

13.7. +ESOERR Socket error indicator

This command is used to indicate there is Socket error



```
<CR><LF>+ESOERR=<socket_id>,<error_code><CR><LF>
```

Parameter

<socket_id> integer type, socket id, This value is the response <socket_id> of AT+ESOC.

<error_code> integer, error code.

-1 : stands for general error

1 : No routing information is found, which usually occurs when the network is dropped

2: TCP connection is disconnected, usually occurs in no network situation

3: TCP connection is disconnected by server, usually device receive reset package from server.

4: TCP connection error

5: illegal value

6: memory error

7: socket block

8: Address has been used

9: Connecting ...

10: Connection is established

11: network interface error



```
+ESOERR=0,1
```

13.8. AT+ESODATAMODE Change to transparent mode

This command is used to change a connected socket AT channel into transparent mode.
The response "CONNECT" means create data mode succeed, and can send raw data by indicated socket later.
The response "NO CARRIER" means close data mode succeed, and should send data by hex format by indicated connected socket later.



Set command
AT+ESODATAMODE=<socket_id>,<data_mode>

Response
<CR><LF>CONNECT<CR><LF>
<CR><LF>NO CARRIER<CR><LF>
<CR><LF>ERROR<CR><LF>

When the data arrived, it will report:
<CR><LF><data><CR><LF>

Parameter
<socket_id> integer socket id, AT+ESOC's response.
<data_mode> integer
1 – create data mode.
0 – close data mode.

Note: you can use +++ to exit transparent transmit mode

13.9. AT+PING Test IP network connectivity to a remote host

This command is used to Test IP network connectivity to a remote host

Note :Prerequisite for ping is the need to establish a PDN connection



AT+PING=<remote addr>[-l/L <p_size>] [-n/N <count>][-w/W <time>][-6][-i <value>][-d<value>]

When ping success, it will return:

<CR><LF>OK<CR><LF>

<CR><LF>+ping: begin, xx.xx.xx.xx, data size= x<CR><LF>

<CR><LF>+ping: finish, Packets: Sent = x, Received =x, Lost =x (x% loss)<CR><LF>

<CR><LF>+ping: RTT statistics: Minimum = xx, Maximum =xx, Average = xx<CR><LF>

When ping failure (<type>=1) , it will return:

<CR><LF>ERROR<CR><LF>

When ping command is send repeated, it will return:

<CR><LF>BUSY<CR><LF>

Parameter

<remote addr> remote address (IPV4address)

-l <value>: payload package size in byte, the valid range is: 8-1460 (64 is in default) .

-n <value>: the number of ping package, default 3.

-w <value>: the setting timeout in unit of ms (10s in default,that is 10000)

-6: the address is IPV6

-i<value>: the interval between ping packages in unit of ms

-d<value>: enable/disable debug information

1: enable debug information

0: disable debug information



AT+PING=192.168.1.2

OK

+ping: begin, 192.168.1.2, data size= 64

+ping: finish, Packets: Sent = 3, Received =3, Lost = 0 (0% loss)

+ping: RTT statistics: Minimum = 650, Maximum =1180, Average = 876

at+ping=192.168.1.2 -d 1

OK

+ping: begin, 192.168.1.2, data size= 64

+ping: 192.168.1.2, received=64 bytes, rtt=500 ms, ttl=115

+ping: 192.168.1.2, received=64 bytes, rtt=850 ms, ttl=115

+ping: 192.168.1.2, received=64 bytes, rtt=310 ms, ttl=115

+ping: finish, Packets: Sent = 3, Received =3, Lost = 0 (0% loss)

+ping: RTT statistics: Minimum = 310, Maximum =850, Average = 553

13.10. Example: Create TCP socket

The socket_id of module NE310 is assigned by module itself, and can't be set by customer in manual. The example below is for the TCP connection establish and data sending:

```
*MATREADY: 1

+CFUN: 1

+CPIN: READY

+IP: 192.168.4.6
AT+ESOC=1,1,1      // (create socket)

+ESOC=0           //the response value is 0, that is the created socket id is 0, this value can't set by customer.
OK

AT+ESOCON=0,8885,"192.168.1.2" // Connect socket to remote address and port, the first parameter 0 is the
response<socket_id>value of AT+ESOC.

OK

AT+ESOSEND=0,8,3131313131313131 //Send data of ASCII code in HEX format
OK

+ESONMI=0,8,3131313131313131 // receive data send by server
AT+ESOCL=0           //close socket
OK
```

13.11. Example: Create UDP socket

```
*MATREADY: 1

+CFUN: 1

+CPIN: READY

+IP: 192.168.4.6
AT+ESOC=1,2,1 // create socket
+ESOC=0
OK
AT+ESOCON=0,7000,"192.168.1.2" // connection socket
OK
AT+ESOSEND=0,8,3131313131313131 // send data
OK
AT+ESOCL=0 // close socket
OK
```

14. MQTT COMMANDS

14.1. AT+EMQSCFG Encryption Configuration of MQTT

The command is used to do Encryption configure of MQTT. This command will take effect once setting, and won't be saved after restart.

Description

After the encrypted file is successfully set, it is valid only for the mqtt instance to be created. Once the mqtt instance is created successfully, the encrypted file will be cleared. If you need to create the mqtt instance again, you need to reset the encrypted file



AT+EMQSCFG=<ssl_state>[,<cert_key_type>,<flag>,<cert_key_content>]

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>



AT+EMQSCFG?

Response

<CR><LF>+EMQSCFG:<mqtt_id>,<ssl_state>[,<cert_key_type>,<size>...]<CR><LF>

...

<CR><LF>OK<CR><LF>

When mqtt instance is not created, return:

<CR><LF>+EMQSCFG:NULL<CR><LF>

<CR><LF>OK<CR><LF>

Or

<CR><LF>ERROR<CR><LF>

Parameter

<ssl_state>Unsigned integer;

0:Disable SSL, no need to set and <cert_key_type>,<flag>,<cert_key_content>.This command has two functions, one is to set mqtt to not encrypt the connection; the second is to clear the settings of the encrypted file;

1:enable SSL, and <cert_key_type>,<flag>,<cert_key_content> can't be omitted.

<cert_key_type> integer,the required encrypt file type

0:ca crt,ca Certificate;

1:client crt,Client Certificate;

2:client key,Client key;

<flag> integer

0:this is the last package

1: this is not the last package.

<cert_key_content>: The content of encrypt file in hexadecimal string, certificate or secret key length is limited by the AT processing mechanism. The length of a single AT command cannot exceed 1,400+ bytes. Therefore, the length of encrypted content that is sent in a single AT is: 1~ 1400, if the total length of the encrypted content exceeds 1400 hexadecimal strings, it need to be split into several AT commands.

<mqtt_id>:MQTT instance identifier. The range is 0-4.

<size> integer, the length of corresponding encrypt file


```
2304f42425945464d3338584c586845386653646457723732733861423776324574654d41774741315564457745422f7751434d4
1417744675944565230500a4151482f42415144416765414d4130474353714753496233445145424377554141344942415141366
9466368436751304279677163583478537744562b4454480a4246346951683677386878445753306f507635596a39717456396a7
85567744d517765457637735030346c6e53667747526b6d797a4a4f4c5a2f3734356671300a7545352b42506a7677445059466b43
676833506633534352557639637555705963304c7859664f796b4e6838364c4e68632f51626e6f447175655157624743410a55656
d6834543662386f5931724d786e5259564b31336e582b704a5367326475352b4f6f644c4c6d724f4a4d35304d5930545275625562
7633636950373947740a764246444f7a533861777446424b587230597172716e54632f396c4b414765462f457a4f624362565a7a68
42525076366a4e4a7453464c6e49647176325264630a3341316e6b49366b4a3643754d317858636c6d472f464259316b496b6b79
693830644b635250367849334b652f43367a38454356442f5a4f424b39760a2d2d2d2d2d454e442043455254494649434154452d2
d2d2d2d
```

OK

```
/***** set client crtend *****/
```

```
/***** set Client keystart *****/
```

```
AT+EMQSCFG=1,2,1,2d2d2d2d2d424547494e205253412050524956415445204b45592d2d2d2d0a4d4949456f77494241414b4
34151454131302b6b66696f624c4e456d52574958575475364543683662304b6d2f43623166747962466a676e636932516537766
50a3530613876734a6a6c4b47384c7537696a6c2b7a5a71386836336d6a474e485550556b62506a51716274785067783458534b6
e6e6b745974795a6a79716939340a71593555466a716a6643524b6341616849437144594f7279354e72797663536145644242423
0346e386b4e6d7059486f73372f483379333564756d73763438490a6638326933366d44334d567247764b37685075676e7177655
342774457322f7375505279794c494e566e6c6939347466386154376e4944575a58426b326953630a724e315a34365a6876775755
34486f576d4a4d4d753433534a5962704d776f76514365793034714c34587334726a62446249753675732b586c5a495a494458470
a7039545562534773314a757344624236676837774e655a6c6133313134587056753562555651494441514142416f49424151436
26f544b714379396e587170470a4870663335516368704e435332562b7742553050374d2b67364f6743544e4d527a514e6173724f
395175384e77545976796e59794b52535931574130503944580a494e78446a444e4837426237325030554c6b7a774e666477474b
59794150377561712f53423148615733704f516b4558356f4c53714567635a7463342b7458590a7532494f4844545533665a62303
8492f516235394a744f4843306b31584845626f415a69557348324671373775613163715179434438337932532f784134437a0a2b
5a4d494b62793050447a6662624469725475726467432f754b58386a41657653433359786f4e48377473646b316f56496e2f30566
5503148316572434a30350a31463175586f52484b327245436e7437576f
```

OK

```
AT+EMQSCFG=1,2,1,624d49567332355379687558785043307839396544524a427734376551553959764f446a51744c5a78445361
735a0a5476636474684a42416f4742414f777a6470374b704d52566130674471304f392f46312f713855627679797558506c363949
4a7877644e574731627a504c6c610a3744516b514c41663332387445627641356f74396539784a564b65664f412f6753376b51393
5575571434a724a785351492f57366f63393658545071644a78560a335a546b7a67304f45674533727036354644662f306c4a55685
930726275676b4b4b53354261476a6e506a69796c75716f79595a765a4a6c416f4742414f6c620a365563697179674b3762757847
3534704a6f7462426f587058456c6553432f48625579714c394c2f6561692b644c715a75425a333538464e34495556484f32350a5a6
f79474e5257644869765451594b3664434346375778576b6931665779705a596972665576537a364c6432384e4c69656e687a4e7
84e4d5a334b684d50554f0a32486c61437368384c6f556e376268736679334f476d47505074525759796d417342796367694d7841
6f4741525173694548616878545a4733733062776669470a4458716c5a6a2b66475674366c2f36462f4a3975446532736e6a554a7
5684447334e4d414362753358314d6f544e3674525863507671753830436174683436440a382f7675733032352f42326756645677
66594c706937463044786e6f726b34334d432f5676324575694361547035666f5068704d6933356a62346d54714d33780a4233324
3464f617a6d746b495639564154594857745455436759414341456e46314752485761436b5257614a4f713062384e486a6742624
d676b52344e47755a0a68725157535236503965496d583338576f6a79596963527448377335787163386555496e5230756b384a4
4565143675378314d37354a6f6f54584d49722b2b4a0a6b7053
```

OK

```
AT+EMQSCFG=1,2,0,78694946735956613757446e596f70524d5642466e434437564f795663465175576f496e396c6c51526973796
b5a656f514127833306133554130682f0a566c412f59514b4267425741596a455433544d594a656f49552f4b2b544e6a6c2b33465
2757074303963303776344e4b6e2b4a775a6e455473714768767351470a6b4963304b472f6d5151745369617a49324f306165524c
6c626a453831574e3041453859676749514d4f6c514c513871546358726437474a69594833544535370a5a6c33326f4a476f447a57
52586844366b647965734d7766546f3148496a7655557a7544574f3975436430467559474c545a71700a2d2d2d2d2d454e4420525
3412050524956415445204b45592d2d2d2d2d
```

OK

```
/***** set Client keyend *****/
```

```
/****** Query encryption configuration******/
```

```
AT+EMQSCFG? /* Query the encryption configuration status of the created mqtt instance */
+EMQSCFG:0,1,0,1731,1,1219,2,1674
```

```
OK
```

```
/* the parameters of response */
```

```
/* mqtt_id:0,ssl_state:1,crt_key_type:0,size:1731,crt_key_type:1,size:1219,crt_key_type:2,size:1674 */
```

14.2. AT+EMQSCFGRAW Encryption Configuration of MQTT through transparent transmission

The command is used to do Encryption configure of MQTT through transparent transmission. This command will take effect once setting, and won't be saved after restart.

Description

After the encrypted file is successfully set, it is valid only for the mqtt instance to be created. Once the mqtt instance is created successfully, the encrypted file will be cleared. If you need to create the mqtts instance again, you need to reset the encrypted file.



```
AT+EMQSCFGRAW=<ssl_state>[,<crt_key_type>,<size>]
```

Response

```
<CR><LF>CONNECT<CR><LF>
```

```
<input_content>
```

```
receive xxx data<CR><LF>
```

```
receive xxx data<CR><LF>
```

```
.....
```

```
<CR><LF>NO CARRIER<CR><LF>
```

```
or
```

```
<CR><LF>ERROR<CR><LF>
```



```
AT+EMQSCFGRAW?
```

Response

```
<CR><LF>+EMQSCFG:<mqtt_id>,<ssl_state>[,<crt_key_type>,<size>...]<CR><LF>
```

```
...
```

```
<CR><LF>OK<CR><LF>
```

When mqtt instance is not created, return:

```
<CR><LF>+EMQSCFG:NULL<CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

Or

```
<CR><LF>ERROR<CR><LF>
```

Parameter

<ssl_state>Unsigned integer;

0:Disable SSL, no need to set and <cert_key_type>,<flag>,<cert_key_content>.This command has two functions, one is to set mqtt to not encrypt the connection; the second is to clear the settings of the encrypted file;

1:enable SSL, and <cert_key_type>,<flag>,<cert_key_content> can't be omitted.

<cert_key_type> integer,the required encrypt file type

0:ca crt,ca Certificate;

1:client crt,Client Certificate;

2:client key,Client key;

<mqtt_id>:MQTT instanceidentifier. The range is 0-4.

<size> integer, the length of corresponding encrypt file,range is 1-2048 hexadecimal string. If length is exceed 2048, it's recommended to use +EMQSCFG to set.

<input_content>: The content of encrypt filein hexadecimal string.The part whose length exceeds <size> is automatically discarded.



```

/***** set ca crt start *****/
AT+EMQSCFGRAW=1,0,1731
CONNECT /* wait for receive CA crt */
receive 350 data
receive 350 data
receive 350 data
receive 350 data
receive 331 data

NO CARRIER /* Receive finished, exit transparent transmission */
/***** set ca crt end *****/

/***** set client crt start *****/
AT+EMQSCFGRAW=1,1,1219
CONNECT /* wait for receive client crt */
receive 350 data
receive 350 data
receive 350 data
receive 169 data

NO CARRIER /* Receive finished, exit transparent transmission */
/***** set client crt end *****/

/***** set client key start *****/
AT+EMQSCFGRAW=1,2,1674
CONNECT /* wait for receive client key */
receive 350 data
receive 350 data
receive 350 data
receive 350 data
receive 274 data

NO CARRIER /* Receive finished, exit transparent transmission */
/***** set client key end *****/

/*****Clear the settings of the encrypted file*****/
AT+EMQSCFGRAW=0
OK
/***** Query encryption configuration *****/
AT+EMQSCFGRAW? /* Query the encryption configuration status of the created
mqtt instance */
+EMQSCFG:0,1,0,1731,1,1219,2,1674

```

```
+EMQSCFG:1,1,0,1731,1,1219,2,1674
```

```
OK
```

```
/* the parameters of response */
```

```
/*
```

```
mqtt_id:0,ssl_state:1,crt_key_type:0,size:1731,crt_key_type:1,size:1219,crt_key_type:2,size:1674
```

```
*/
```

```
/* mqtt_id:1,ssl_state:1,crt_key_type:0,size:
```

```
1731,crt_key_type:1,size:1219,crt_key_type:2,size:1674 */
```

14.3. AT+EMQNEW New MQTT

This command is used to new MQTT over TCP connection with server.



AT+EMQNEW=<server>,<port>,<command_timeout_ms>,<bufsize>[,<cid>]

Response

<CR><LF>+EMQNEW: <mqtt_id><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<server>string type, MQTT server IP address

<port>string type , MQTT server port

<command_timeout_ms> integer type, AT command timeout in unit of ms.the valid range can be 0-4294967295

<bufsize> integer type, the size of send and receive buffer. Recommend not exceed 10k. This stand for the size of MQTT pdu need to be saved send or receive, if set too large, it will waste the memory of module.

<cid>integer type, PDP context ID. [optional]

<mqtt_id>: integer type, MQTT local instance identifier. The returned value of +EMQNEW, and only supports creating one local instance.



AT+EMQNEW="192.168.1.2","9000",12000,100 //new MQTT over TCP connection with server

+EMQNEW: 0

OK

14.4. AT+EMQCFG Configure Parameters of MQTT

The command is used to configure optional parameters of MQTT. This command will take effect once setting, and won't be saved after restart.



```
AT+EMQCFG=<mqtt_id>,<key>,<value>
Response
<CR><LF>OK<CR><LF>
or
<CR><LF>ERROR<CR><LF>
```



```
AT+EMQCFG=<mqtt_id>,<key>
Response
<CR><LF>+EMQCFG: <key>,<value><CR><LF>
<CR><LF>OK<CR><LF>
or
<CR><LF>ERROR<CR><LF>
```



```
AT+EMQCFG=<mqtt_id>
Response
<CR><LF>+EMQCFG: mqtt_id,<mqtt_id><CR><LF>
<CR><LF>+EMQCFG:SSL,<value><CR><LF>
<CR><LF>+EMQCFG:CACERT,<value><CR><LF>
<CR><LF>+EMQCFG:CLIENTCERT,<value><CR><LF>
<CR><LF>+EMQCFG:CLIENTKEY,<value><CR><LF>
<CR><LF>OK<CR><LF>
or
<CR><LF>ERROR<CR><LF>
```

Parameter

<mqtt_id>:MQTT socket identifier. The range is 1-5.

<key>: The key of configuration. It can be any of the following types:

key	
SSL	Enable/disable SSL
CACERT	CA CA Certificate
CLIENTCERT	Client Certificate
CLIENTKEY	Client key

<value>:

key	value
SSL	0:disable SSL 1:enable SSL
CACERT	CA Certificate file name, Length: 0~128
CLIENTCERT	Client Certificate file name , Length: 0~128
CLIENTKEY	Client key file name, Length: 0~128



```
AT+EMQCFG=1,SSL,0
OK
```

```
AT+EMQCFG=1,CACERT,ca.crt
OK
```

```
AT+EMQCFG=1,CLIENTCERT,cli.crt
OK
```

```
AT+EMQCFG=1,Clientkey,cli.key
OK
```

```
AT+EMQCFG=1,SSL
+EMQCFG: SSL:0
```

```
OK
```

```
AT+EMQCFG=1
+EMQCFG: mqtt_id,1
+EMQCFG: SSL,0
+EMQCFG: CACERT,ca.crt
+EMQCFG: CLIENTCERT,cli.crt
+EMQCFG: CLIENTKEY,cli.key
```

```
OK
```

14.5. AT+EMQCON Send MQTT Connection packet

This command is used to Send MQTT Connection packet



Set command

```
AT+EMQCON=<mqtt_id>,<version>,<client_id>,<keepalive_interval>,<cleansession>,<will_flag>[,<will_options>][,<username>,<password>]
```

Response

```
<CR><LF> OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

Parameter

<mqtt_id>integer type, MQTT id, AT+EMQNEW's response

<version>integer type, MQTT version, can be 3 or 4

<client_id>string type , client ID, should be unique

<keepalive_interval>integer type, keep alive interval, range is 0~65535, unit is second.

<cleansession> integer type, clean session, can be 0 or 1 (0:Server save the subscriber information of client after client disconnect; 1: :Server don't save the subscriber information of client after client disconnect)

<will_flag>integer type, will flag, can be 0 or 1(1: there is will option)

<will_options>string type , optional, if <will_flag>=1, this <will_option> is required. The format of will option is: topic=xxx,QoS=xxx,retained=xxx,message_id=xxx,message=xxx

<username> string type , username (optional)

<password>string type , password (optional)



Example

```
AT+EMQCON=0,3,"myclientid",1000,1,0 //send MQTT connection packet
```

```
OK
```

```
AT+EMQCON=1,3,"myclientid2",1000,1,1,"topic=WILL,QoS=1,retained=0,message_len=4,message=3838" //send MQTT connection packet with will options, the will message is "88"
```

```
OK
```

This command only support set format, and can't support test and read format

14.6. AT+EMQDISCON Disconnect MQTT

This command is used to send MQTT Disconnect packet, disconnect TCP connection with MQTT server and delete MQTT

This command only support set format, and can't support test and read format



```
AT+EMQDISCON=<mqtt_id>
Response
<CR><LF> OK<CR><LF>
or
<CR><LF>ERROR<CR><LF>
Parameter
<mqtt_id> integer type, MQTT id, AT+EMQNEW's response
```



```
AT+EMQDISCON =0 //send MQTT Disconnect packet
OK
```

14.7. +EMQDISCON Unsolicited report of MQTT disconnection

This command is used to unsolicited report the MQTT disconnection.

```
<CR><LF> +EMQDISCON:<mqtt_id><CR><LF>
Parameter
<mqtt_id>integer type, MQTT id, AT+EMQNEW's response
```



```
+EMQDISCON:0 //Unsolicited report of MQTT disconnection
```

14.8. AT+EMQSUB Send MQTT subscribe packet

This command is used to Send MQTT subscribe packet. This command only support set format, and can't support test and read format



```
AT+EMQSUB=< mqtt_id>,<topic>,<QoS>
Response
<CR><LF> OK<CR><LF>
or
<CR><LF>ERROR<CR><LF>
Parameter
<mqtt_id> integer type, MQTT id, AT+EMQNEW's response
<topic > string type , topic of subscribe message. The length should not exceed the buffer length
set by command "AT+EMQNEW", as MQTT PDU include topic, the whole MQTT pdu can't exceed
the length of flash.
<Qos > integer type, message QoS, 0, 1 or 2
```



```
AT+EMQSUB=0,"topic",1//Send MQTT subscribe packet
OK
```

14.9. AT+EMQUNSUB Send MQTT unsubscribe packet

This command is used to Send MQTT unsubscribe packet. This command only support set format, and can't support test and read format



AT+EMQUNSUB=< mqtt_id>,<topic>

Response

<CR><LF> OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<mqtt_id> integer type, MQTT id, AT+EMQNEW's response

<topic > string type , topic of subscribe message. The length should not exceed the buffer length set by command "AT+EMQNEW", as MQTT PDU include topic, the whole MQTT pdu can't exceed the length of flash.



AT+EMQUNSUB=0,"topic"//Send MQTT unsubscribe packet

OK

14.10. AT+EMQPUB Send MQTT publish packet

This command is used to Send MQTT publish packet . This command only support set format, and can't support test and read format



AT+EMQPUB=<mqtt_id>,<topic>,<QoS>,<retained>,<dup>,<message_len>,<message>

Response

<CR><LF> OK<CR><LF>

or

<CR><LF> ERROR<CR><LF>

Parameter

<mqtt_id> integer type, MQTT id, AT+EMQNEW's response

<topic > string type , topic of subscribe message. The length should not exceed the buffer length set by command "AT+EMQNEW", as MQTT PDU include topic, the whole MQTT pdu can't exceed the length of flash.

<QoS > integer type, message QoS, 0, 1 or 2

<retained > integer type, retained flag,0 or 1(0: not retain information; 1: retain information)

<dup> integer type, duplicate flag, 0 or 1 (0: send the first time; 1 duplicate send)

<message_len > integer type, Length of publish message

<message> string type , content of of publish message in ASCII code, not support other format, or the send data maybe null.



AT+EMQPUB=0,"mytopic",1,0,0,4,"31323334"

message is "1234"

OK

//Send MQTT publish packet, the publish

14.11. +EMQPUB –Unsolicited report of MQTT publish packet

This command is used to Unsolicited report of MQTT publish packet



```
<CR><LF> +EMQPUB: <mqtt_id>,<topic>,<QoS>,<retained>,<dup>,<message_len>,<message><CR><LF>
```

Parameter
 <mqtt_id> integer type, MQTT id, AT+EMQNEW's response



```
+EMQPUB:0,"topic",1,0,0,4,"31323334"//report the received MQTT publish packet
```

14.12. Example: create MQTT connection

```
*MATREADY: 1

+CFUN: 1

+CPIN: READY

+IP: 192.168.4.6

AT+EMQNEW="192.168.1.2","9000",12000,100 // new MQTT over TCP connection with server
+EMQNEW: 0
OK
AT+EMQCON=0,3,"myclientid",1000,1,0 // Send MQTT Connection packet
OK
AT+EMQSUB=0,"mytopic",1 //Send MQTT subscribe packet
OK
AT+EMQPUB=0,"mytopic",1,0,0,4,"31323334" //Send MQTT publish packet
OK

+EMQPUB:0,"mytopic",1,0,0,4,"31323334" // receive MQTT publish packet
AT+EMQUNSUB=0,"mytopic" //Send MQTT unsubscribe packet
OK
AT+EMQDISCON=0 //send MQTT Disconnect packet
OK
```

15. COAP COMMANDS

15.1. AT+ECOAPNEW Create a CoAP client instance

Create a CoAP client instance



```
AT+ECOAPNEW=<ip_addr>,<port>,<cid>
```

Response

```
<CR><LF>+ECOAPNEW:<coap_id>
```

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF> ERROR<CR><LF>
```

Parameter

<ip_addr>: CoAP server IP address

<port>: CoAP server port (spec default 5683)

<cid>: network number

<coap_id>: CoAP server instance id created by the command



```
AT+ECOAPNEW=139.196.187.107,5683,1
```

```
+ECOAPNEW: 2
```

```
OK
```

15.2. AT+ECOAPSEND Send CoAP data

Send data to CoAP server with the created CoAP client instance.



```
AT+ECOAPSEND=<coap_id>,<data_len>,<data>
```

```
<CR><LF>+ECOAPNMI:<coap_id>,<srv_data_len>,<srv_data>
```

```
<CR><LF><CR>OK<LF>
```

```
<CR><LF> ERROR<CR><LF>
```

Parameter

<coap_id>: CoAP client instance id created by the AT+COAPNEW command

<data_len>: Send data length (by byte)

<data>: the hex data streaming

<srv_data_len> : integer type, the response data length from server (by byte)

<srv_data>: Hexinteger type, the response data from server (Hex data streaming)



```

</> AT+ECOAPSEND=1,11,40013b06b474657374c102 coap get method
OK
+ECOAPNMI:
1,47,60451784c0211e9102ff547970653a20302028434f4e290a436f64653a20312028474554290a4d49443a2036303230

AT+ECOAPSEND=1,24,400325b6b876616c696461746510ff313233313233343536 coap put method
OK
+ECOAPNMI: 1,9,6044178544268069b9

AT+ECOAPSEND=1,10,40040214b47465737410 coap delete method
OK
+ECOAPNMI: 1,4,60421786

AT+ECOAPSEND=1,20,40020216b47465737410ff313233313233343536 coap post method
OK
+ECOAPNMI: 1,34,60411787896c6f636174696f6e31096c6f636174696f6e32096c6f636174696f6e33

```

Note: For detail of the hex data streaming<data>, that is coap protocol packet, you can refer to: <http://www.rfc-base.org/rfc-7252.html>. We do simple introduce as follows:

1. Coap Get method: get the resource from server

Take command AT+ECOAPSEND=1,11,40013b06b474657374c102 content for example. the stream of <data> is :

40013b06b474657374c102

40013b06 : 4 bytes of CoAP header. The stream of it is :

0x40 is 01000000 in bit, 1 byte

bit1&bit2 is version; bit3&bit4 is type (Confirmable (0), Non-confirmable (1), Acknowledgement (2), or Reset (3)). bit5-bit8 token length

0x01 code in coap protocol (get(0x01), post(0x02), put(0x03), delete(0x04)) , 1 byte. 0x01 is stand for get method.

0x3b06 message id, 2 bytes. The value is not fixed, it is incremented according to the sending instruction.

b474657374c102: Coap options, include URI resource and others

2. Coap put method: update the resource of server

Take command AT+ECOAPSEND=1,24,400325b6b876616c696461746510ff313233313233343536 content for example:

the stream of <data> is :

400325b6b876616c696461746510ff313233313233343536

400325b6: 4 bytes of CoAP header

b876616c696461746510ff: Coap options, include URI resource and others

f: options marker

313233313233343536: effective payload

3. Coap delete method: delete the resource on server

Take command AT+ECOAPSEND=1,10,40040214b47465737410 content for example:

the stream of <data> is :

40040214b47465737410

40040214: 4 bytes of CoAP header

b47465737410: Coap options, include URI resource and others

4. Coap post method: create or update the resource on server

Take command AT+ECOAPSEND=1,20,40020216b47465737410ff313233313233343536 content for example:

the stream of <data> is :

40020216b47465737410ff313233313233343536

40020216: 4 bytes of CoAP header

b47465737410: Coap options, include URI resource and others

ff: options marker

313233313233343536: effective payload

15.3. AT+ECOAPDEL Destroy the CoAP client instance

Destroy the CoAP client instance



AT+ECOAPDEL=<coap_id>

Response

<CR><LF><CR>OK<LF>

or

<CR><LF> ERROR<CR><LF>

Parameter

<coap_id>: CoAP client instance id created by the AT+COAPNEW command



AT+ECOAPDEL=1

OK

15.4. +ECOAPNMI Content from CoAP server

Indicate content from CoAP server.



```
<CR><LF>+ECOAPNMI:<coap_id>,<data_len>,<data><CR><LF>
```

Parameter

<coap_id>: CoAP client instance id created by the AT+COAPNEW command
 <data_len>: data length (by byte)
 <data>: the hex data streaming



```
AT+ECOAPSEND=1,11,40013b06b474657374c102 coap get method
```

OK

```
+ECOAPNMI:
```

```
1,47,60451784c0211e9102ff547970653a20302028434f4e290a436f64653a20312028474554290a4d49443a2036303230
```

```
AT+ECOAPSEND=1,24,400325b6b876616c696461746510ff313233313233343536 coap put method
```

OK

```
+ECOAPNMI: 1,9,6044178544268069b9
```

```
AT+ECOAPSEND=1,10,40040214b47465737410 coap delete method
```

OK

```
+ECOAPNMI: 1,4,60421786
```

```
AT+ECOAPSEND=1,20,40020216b47465737410ff313233313233343536 coap post method
```

OK

```
+ECOAPNMI: 1,34,60411787896c6f636174696f6e31096c6f636174696f6e32096c6f636174696f6e33
```

Note: About the hex data streaming<data>, there is four methods of the Possible Response of get, put, post, delete. You can refer to address for detail: <http://www.rfc-base.org/rfc-7252.html> . We do simple introduce as follows:

1. data stream analyze for GET method:

```
60451784c0211e9102ff547970653a20302028434f4e290a436f64653a20312028474554290a4d49443a2036303230
```

60451784: 4 bytes of CoAP header. The stream of it is :

```
0x60-01100000 in binary,1 byte
```

bit1&bit2 is version; bit3&bit4 is type (Confirmable (0), Non-confirmable (1), Acknowledgement (2), or Reset (3)). bit5-bit8 token length

```
0x45 --coap code, it is 2.05 Content, request response is ok . 1 byte.
```

```
0x1784 -- message id2 bytes. The value is not fixed, it is incremented according to the sending instruction.
```

```
c0211e9102: Coap options, include URI resource and others
```

f: options marker

```
547970653a20302028434f4e290a436f64653a20312028474554290a4d49443a2036303230: useful payload
```

2. data stream analyze for PUT method

```
6044178544268069b9
```

```
604417854 4 bytes of CoAP header
```

0x44 (01000100, c: 010, dd:00100) is response code, the format is c.dd. 1 byte. Other parameter you can refer

to the get method.

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	1	0	0	0	1	0	0
This 3bits stand for C code, 2 in Decimal 2: Operation success 4: client error 5: server error			This 5bits stand for dd code, 04 in Decimal 2.01-Created, 2.02-Deleted, 2.03 -Valid, 2.04-changed, 2.05-Content 4.00-Band request, 4.01 -Unauthorized, 4.02- Bad Option,4.03 - Forbidden,4.04 -Not Found,4.05- Method Not Allowed, 4.06 -Not Acceptable,4.12 -Precondition Failed,4.13 -Request Entity Too Large, 4.15 -Unsupported Content-Format 5.00 -Internal Server Error,5.01 -Not Implemented, 5.02- Bad Gateway, 5.03 -Service Unavailable, 5.04- Gateway Timeout, 5.05- Proxying Not Supported				
0x44 the response is success, that is 2.04 Changed,							

44268069b9--Coap options

3. data stream analyze for DELETE method

60421786

604217864 bytes of CoAP header

0x42 is response succes, coap code is 2.02 Deleted, 1 byte. Other parameter you can refer to the get method.

4. data stream analyze for POST method

60411787896c6f636174696f6e31096c6f636174696f6e32096c6f636174696f6e33

604117874 bytes of CoAP header

0x41 is response success, coap code is 2.03 Valid, 1 byte. Other parameter you can refer to the get method.

896c6f636174696f6e31096c6f636174696f6e32096c6f636174696f6e33--Coap options

15.5. Example: COAP client send data

```
*MATREADY: 1

+CFUN: 1

+CPIN: READY

+IP: 192.168.4.6

at+ping=192.168.1.2 // ping server to test if connect to network
+ping: finish, Packets: Sent = 3, Received =3, Lost = 0 (0% loss)

AT+ECOAPNEW=139.196.187.107,5683,1 // Create CoAP client
+ECOAPNEW: 1 --- return CoAP client ID
OK

AT+ECOAPSEND=1,11,40013b06b474657374c102 // client send data get method
+ECOAPNMI:
1,47,60451784c0211e9102ff547970653a20302028434f4e290a436f64653a20312028474554290a4d49443a2036303230 //
the server response
OK

AT+ECOAPSEND=1,24,400325b6b876616c696461746510ff313233313233343536 // client send data put method
+ECOAPNMI: 1,9,6044178544268069b9 // Content from CoAP server
OK
```

AT+ECOAPSEND=1,10,40040214b47465737410 // client send data delete method

+ECOAPNMI: 1,4,60421786 --- Content from CoAP server

OK

AT+ECOAPSEND=1,20,40020216b47465737410ff313233313233343536 // client send data post method

+ECOAPNMI: 1,34,60411787896c6f636174696f6e31096c6f636174696f6e32096c6f636174696f6e33 // the server response

OK

AT+ECOAPDEL=1 // Destroy the CoAP client instance

OK

16. FTP COMMANDS

16.1. AT+FTPOPEN Start FTP file service

Start FTP file service. After the module report “+IP:<IP_address>”, you can start FTP connection.

Note: When FTP related commands fail, it will automatically disconnect from the server and clean up local resources.



AT+FTPOPEN=<serverip>,<serverport>,<username>,<password>,<type>,<ftp/ftps>[,<timeout>]

Response

```
<CR><LF>OK<CR><LF>
<CR><LF>+FTPOPEN: OPEN SUCCESS<CR><LF>
or
<CR><LF>OK<CR><LF>
<CR><LF>+FTPOPEN: OPEN FAIL<CR><LF>
or
<CR><LF>ERROR<CR><LF>
```

Parameter

<serverip>: char string, destination ip address of FTP server
 <serverport>: destination port of FTP server
 <username>: char string, FTP username, the maximum length is 100 bytes
 <password>: char string, FTP password, the maximum length is 100 bytes
 <type>: transmit type, 0 ASCII mode; 1 binary mode;
 <ftp/ftps>: int type, 0 means ftp, 1 means ftps
 <timeout>: login FTP server timeout, unit: ms, default value is 60000ms, that is 1 min. The range can be 5000~300000



AT+FTPOPEN=?

Response
 <CR><LF>OK<CR><LF>



AT+FTPOPEN?

Response
 <CR><LF>+FTPOPEN: <connection_status><CR><LF>
 <CR><LF>OK<CR><LF>



AT+FTPOPEN=192.168.1.1,21,TEST,TEST,1,0

OK // FTP connection is established

+FTPOPEN: OPEN SUCCESS

16.2. AT+FTPCLOSE Close FTP file service

Close FTP file service



AT+FTPCLOSE

Response
<CR><LF>OK<CR><LF>



AT+FTPCLOSE
OK

16.3. AT+FTPSIZE Get the Size of FTP File

Get the Size of FTP File



AT+FTPSIZE=<remote_path>,<remote_filename>

Response
<CR><LF>+FTPSIZE: <length><CR><LF>
<CR><LF>OK<CR><LF>
or
<CR><LF>ERROR<CR><LF>

Parameter

<remote_path>: char string, the file path in the server

<remote_filename>: char string, file name

<length>: the file size

Note: If the file is incorrectly operated or downloaded file not exist (such as "+FTPGET: FAIL" or "+FTPSIZE: File not found"), the module will actively disconnect from the server.



AT+FTPSIZE=/ttest/ftp1.c
+FTPSIZE: 48

OK

16.4. AT+FTPGET FTP File download

This command is used to download the FTP file, you can download the whole file or part of the file. The maximum length of file is 6K byte. If length is exceed 6K, the download will be failed.



AT+FTPGET=<path>,<filename>,<offset>,<length>

Response

```
<CR><LF>OK<CR><LF>
+FTPGET:<LENGTH>,<file_content>
[<CR><LF>+FTPGET:<LENGTH>,<file_content>][...]
<CR><LF>+FTPGET: END <CR><LF>
Or
<CR><LF>+FTPGET: FAIL<CR><LF>
Or
<CR><LF>ERROR<CR><LF>
```

Parameter

<path>: char string, the file path users need to download

<filename>: char string, the file name

<offset>: int type, the offset of the file

<length>: int type, Maximum length of single downloaded data, the range can be 1~ 1024 bytes. If the length of file (can get by command +FTPSIZE) is more than this parameter <length>, it will be divided to multi packages to show the file content.

When <viewmode> of +FTPCFG is set to 0, the max. Length can reach 1024, while <viewmode> of +FTPCFG is set to 1, the max. Length can reach 512,

<file_content>: content of FTP file

<LENGTH>: the length of each package of <file_content>.

Note:

If the file is incorrectly operated or downloaded file not exist (such as "+FTPGET: FAIL" or "FTPSIZE: File not found"), the module will actively disconnect from the server.



```
AT+FTPSIZE=/,test.txt
+FTPSIZE: 20
```

OK

```
AT+FTPGET=/ttest/,ftptest.c,0,10
```

OK

```
+FTPGET:10,11223344556677889900
```

```
+FTPGET:10,39663234316564220d0a
```

```
+FTPGET: END
```

16.5. AT+FTPPUTCON Establish upload link

This command is used to create an upload link.



```
AT+FTPPUTCON=<path>,<remote_filename>
```

Response

```
<CR><LF>OK<CR><LF>
```

```
<CR><LF>+FTPPUTCON: SUCCESS<CR><LF>
```

or

```
<CR><LF>OK<CR><LF>
```

```
<CR><LF>+FTPPUTCON: FAIL<CR><LF>
```

Parameter

<path>: char string, the file path that users want to upload.

<remote_filename>: char string, file name in the server that need to be uploaded



```
AT+FTPPUTCON=/ttest/,ftp6.c
```

```
OK
```

```
+FTPPUTCON: SUCCESS
```

16.6. AT+FTPPUTDATA FTP File upload

This command is used to send data to FTP server, and data length should not larger than 500 bytes. If data length is larger than 500bytes, it will prompt: data length must less 500.

Note: you need to use AT+FTPPUTCON to establish upload link before upload file.



AT+FTPPUTDATA=<is_eof>

Response

<CR><LF>CONNECT<CR><LF>

filecontext

<CR><LF>NO CARRIER<CR><LF>

<CR><LF>+FTPPUTDATA: END SUCCESS<CR><LF>

or

<CR><LF>CONNECT<CR><LF>

<CR><LF>data length must less 500<CR><LF>

<CR><LF>NO CARRIER<CR><LF>

Parameter

<is_eof>: interger type

0: not the last data package;

1: the last data package

Note: Other ftp-related operations are not allowed before the data transfer finished

(AT+FTPPUTDATA=1).



AT+FTPPUTCON=/,1234567.txt

OK

+FTPPUTCON: SUCCESS

AT+FTPPUTDATA=1

CONNECT

"Ljab2rDAoGBAN2uAM2gchoFiu4v1HFL8D7lweEpi6ZnMJnEu/dEgGQJFjwdpLnPbsj4c75odQ4Gz8
gsw9lao9VVzbusoRE/JGI4aTdOOpATXyG7eG1Qu+5Yc1YGXcCrlIA2xM9xx+d7f+smPzN+WIEg5GJDY
ZDjAZHG5BNvi/FfM1C9dOtv2dAoGAF0t5KmwbjWHBhcVqO4IcBVvN3Bllc1ue2YRXEDlxY5b0r8N4XceM
gKmW18OHApZxf8uPDauWZLXOgl4uepvwhZC3EuWrSyyICNhLY21Ah7hblEBPF3L3ZsOwC+UErL+dXW
LdB56Jgy3gZaBeW7bvDrEnocJbqCm7lukhXHOBK8CgYEAwqdHB0hqyNSzIOGY7v9abzB6pUdA3BZi
QvEs3LjHVd4HPJ2x0N8CgrBIWoe0q8+0hSMmeE96WW/7jD3fPWwCR5zIXknxBwertyuiuytsdfghgfa

NO CARRIER

+FTPPUTDATA: END SUCCESS

16.7. AT+FTPCFG configuration FTP file content download format

Extended command, this command is used to configure FTP file content download format in hex string or in raw data format.



AT+FTPCFG=<viewmode>,<value>

Response

<CR><LF>OK<CR><LF>

Or

<CR><LF>ERROR<CR><LF>

Parameter

<viewmode> string type, only support fixed value viewmode at currently.

<value> int type

0: raw data format(default value)

1:HEX format



AT+FTPCFG=viewmode,1

OK

16.8. AT+FTPGETOTA download Difference packet from FTP server

This command is used to execute FTP FOTA Update, that is download Difference packet from FTP server, and then write the package to specific position of flash(the storage used for fota update)



AT+FTPGETOTA=<path>,<filename>

Response

<CR><LF>OK<CR><LF>

Or

<CR><LF>ERROR<CR><LF>

Parameter

<path>: string, specific path for differential package; no need quotation marks; can't exceed 200 characters;

<filename>: string, file name of differential package; no need quotation marks; can't exceed 200 characters;

Note:

1. This command is asynchronous and returns OK or ERROR immediately after being sent. The download result(FOTA DOWNLOADED or FOTA DOWNLOADFAIL) will be unsolicited reported. During the execution of this command, other FTP commands cannot be successfully executed. .
2. It is recommended not to perform the FTP process during the FTP FOTA process.



AT+FTPGETOTA=/tlttest,update.zip

OK

+FOTADL: 100%

FOTA DOWNLOADED

FOTA MD5 OK

16.9. AT+OTAUP Excute FTP FOTA Update

This command is used to execute FTP FOTA Update, that is restarting the module and updating the firmware.



AT+OTAUP

Response

<CR><LF>OK<CR><LF>

Or

<CR><LF>ERROR<CR><LF>



AT+OTAUP

OK

FOTA UPDATING

16.10. URC of FTP FOTA and LwM2M FOTA

URC	Description
+FOTADL: <precent>	Download progress, reported as a percentage
FOTA DOWNLOADED	Differential package download completed
FOTA DOWNLOADFAIL	Differential package download failed
FOTA MD5 OK	Differential package md5 check succeeded
FOTA MD5 ERROR	Differential package md5 check failed
FOTA UPDATING	Ready to update (restart the module)
FOTA SUCCESS	update succeeded
FOTA FAILED	update failed

16.11. FTP Process Example

```
*MATREADY: 1
+CFUN: 1
+CPIN: READY
+IP: 192.168.4.6

AT+FTPOPEN=192.168.1.2,test,test,0
OK

+FTPOPEN: OPEN SUCCESS
AT+FTPSIZE=/log/123.txt
+FTPSIZE: 212

OK
AT+FTPGET=/ttest/,ftp1.c,0,48

OK
+FTPGET:22,at+ping=192.168.1.2

FTPGET: END

AT+FTPPUTCON=/,you.txt
OK

+FTPPUTCON: SUCCESS

AT+FTPPUTDATA=0
CONNECT

T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"

NO CARRIER

AT+FTPPUTDATA=1
CONNECT

T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"
T+IMQTTAUTH="a1FGLQAA1sE","2bXZ0bBdJqPKQInCVjna","CN3EgKIGvvnucqh01W4Kz2hckwCUYvMh"

NO CARRIER

+FTPPUTDATA: END SUCCESS

AT+FTPCLOSE
OK
```

16.12. FTP FOTA Process Example

```

//Querying the version number before upgrading

AT+FTPOPEN=ftp.telit.com,21,fest345,gre888h,1,0           // connect to FTP server
OK

+FTPOPEN: OPEN SUCCESS                                   // Success to connect to FTP server
AT+FTPGETOTA=/tlttest,updateB.zip                         // start to download differential package
OK

+FOTADL: 100%                                           // Download progress report

FOTA DOWNLOADED                                         // Differential package download completed

FOTA MD5 OK                                             // Differential package md5 check successes

AT+FTPCLOSE                                           // Close FTP connection
OK

AT+OTAUP                                               // Excute FTP FOTA update
OK

FOTA UPDATING                                           // start update ,ready to restart the module

F1: 0000 0000                                           // restart the module

V0: 0000 0000 [0001]

00: 0006 000C

01: 0000 0000

U0: 0000 0001 [0000]

T0: 0000 00B4

Leaving the BROM

*MATREADY: 1

+CFUN: 1
FOTA SUCCESS                                           // update succeeded
+CPIN: READY

+IP: 124.70.100.3

// Query version number after updating

```

17. HTTP/HTTPS COMMANDS

17.1. AT+EHTTPCREATE Create a HTTP/HTTPS client instance

This command is used to Create a HTTP/HTTPS client instance.

If the <host> parameter starts with "https://", the device will create an HTTPS instance.



```
AT+EHTTPCREATE=<flag>,<total_len>,<len>,<host>[,<auth_user>,<auth_password>,<server_cert_len>,<server_cert>,<client_cert_len>,<client_cert>,<client_pk_len>,<client_pk>]
```

Response

```
<CR><LF>+EHTTPCREATE:<httpclient_id>
```

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

Parameter

<flag> integer type

1: there is multiple packets

0: this packet is the last one

<total_len> integer type, the total length of command, if there is multiple packets, it stands for the total length of all the packets.

<len>integer type, the total length of current package

The real parameter is defined from <host> to <clnt_pk>, if the total length of parameter string is too long, it must be splited to multiple packages. And every string should be wrapped up by quotation marks. For example: "http://192.144.130.27:80/,,,0,,0,,0," stands for one package, you need to add quotation marks, that is ("http://192.144.130.27:80/,,,0,,0,,0,")

< host>string type , http server host

<auth_user>string type , authorization name, optional

<auth_password>>string type , authorization password, optional

<server_cert_len>integer type, https server certification length, optional

<server_cert>string type , https server certification, optional

<client_cert_len>integer type, https client certification length, optional

<client_cert>string type , https client certification, optional

<client_pk_len>:integer type, https private key length, optional

<client_pk>,https private key, optional

<httpclient_id>integer type, http client ID



```
AT+EHTTPCREATE=0,38,38,"http:// 192.168.1.2:80/,,,0,,0,,0,"
```

```
+EHTTPCREAT:0
```

```
OK
```

17.2. AT+EHTTPCON Establish the HTTP/HTTPS connection

This command is used to Establish the HTTP/HTTPS connection. This command only support set format, and can't support test and read format



AT+EHTTPCON=<httpclient_id>

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<httpclient_id>integer type. The indicator of http client instance created by the AT+EHTTPCREATE command



AT+EHTTPCON=0

OK

17.3. AT+EHTTPDISCON Close HTTP/HTTPS connection

This command is used to create http instance to disconnect the connection with host. After disconnected and before destroy the http instance, you can use AT+EHTTPCON to connect it again. This command only support set format, and can't support test and read format



AT+EHTTPCON=<httpclient_id>

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<httpclient_id>integer type. The indicator of http client instance created by the AT+EHTTPCREATE command



AT+EHTTPDISCON=0

OK

17.4. AT+EHTTPDESTROY Destroy the HTTP/HTTPS client instance

This command is used to Destroy the HTTP/HTTPS client instance. This command only support set format, and can't support test and read format



AT+EHTTPDESTROY=<httpclient_id>

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<httpclient_id> integer type. The indicator of http client instance created by the AT+EHTTPCREATE command



AT+EHTTPDESTROY=0
OK

17.5. AT+EHTTPSEND Send HTTP/HTTPS package

This command is used to Send HTTP/HTTPS package to server. This command only support set format, and can't support test and read format



AT+EHTTPSEND=<flag>,<total_len>,<len>,<httpclient_id>,<method>,<path_len>,<path>,<customer_header_len>,<customer_header>,<content_type_len>,<content_string_len>,<content_string>

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<flag> integer type

flag: 1 means there are more packages, 0 means this package is the last one

<total_len> integer type, the total length of command, if there is multiple packets, it stands for the total length of all the packets.

<len> integer type, the total length of current package

The real parameter is defined from <host> to <client_pk>, if the total length of parameter string is too long, it must be splitted to multi packages. And every string should be wrapped up by quotation marks. For example: 0,0,9,"/Test.txt",0,,0,, is one package

<httpclient_id> integer type, The indicator of http client instance created by the AT+EHTTPCREATE command

<method> integer type, http method

0: HTTPCLIENT_GET

1: HTTPCLIENT_POST

2: HTTPCLIENT_PUT

3: HTTPCLIENT_DELETE

<path_len>: integer type, length of path

<path>: string type, path. The resource path on server, ex. "/html/login/index.html" means the url full path is "<host>/html/login/index.html"

<customer_header_len> integer type, Length of customer header

<customer_header> string type, The string converted from customer header string, the string length must equal original header string length * 2. The original customer header string doesn't include host, path, content type, content length.

<content_type_len> integer type. The length of Content_type

<Content_type> string type. A string indicate the content type of the content, if the method is not POST and PUT, it must be empty

<content_string_len>integer type. Must be the string length of content_string, equals hex data size * 2
 <Content_string>:string type . The string converted from content hex data, the string length must equal hex data size * 2



```
AT+EHTTSEND=0,27,27,"0,0,9,/Test.txt",0,,0,0,"
OK
+EHTTPMIH:0,0,800,Content-Type: text/plain
Content-Length: 9
Accept-Ranges: bytes
Server: HFS 2.3 beta
Last-Modified: Sat, 13 Jan 2018 02:50:00 GMT
Content-Disposition: filename="Test.txt";
\0

+EHTTPMIC:0,0,9,18,313233313233313233
+EHTTPERR:0,-2
```

17.6. +EHTTPMIH Header of the response from host

The response from host have 2 parts. This is the header part and content part will follow this URC.
 This is the Header of the response from host, and it will be followed by content part reported by command +EHTTPMIC

<CR><LF>+EHTTPMIH:<httpclient_id>,<flag>,<header_max_length>,<header><CR><LF>

Parameter

<httpclient_id>integer type. The indicator of http client instance created by the AT+EHTTPCREATE command

<flag>integer type. flag: The flag to indicate if there are more data of the HTTP header

0: one package

1: multiple packages

<header_max_length>integer type. The maximum length (buffer size) of the header string

<header>string type. header data of response



```
+EHTTPMIH:0,0,800,Content-Type: text/plain
Content-Length: 9
Accept-Ranges: bytes
Server: HFS 2.3 beta
Last-Modified: Sat, 13 Jan 2018 02:50:00 GMT
Content-Disposition: filename="Test.txt";
\0
```


17.7. +EHTTPNMIC Content of the response from host

The response from host have 2 parts. This is the content part and follow by the header part URC. And there are multi content URC follow one header URC.



```
<CR><LF>+EHTTPNMIC:<httpclient_id>,<flag>,<total_length><content_package_len>,<content_package_string><CR><LF>
```

Parameter

<httpclient_id>integer type.The indicator of http client instance created by the AT+EHTTPCREATE command

<flag>integer type. flag: The flag to indicate if there are more data of the HTTP header

0: one package

1: multiple packages

<total_length> The total length of the content. It is get from header "Content-Length : xxx", so if the response is not 200 OK, maybe the value is -1

Note:

1. if there is errors (404, 405, ...) there won't contain content length in response

2. translate content by trunked mode (HTTP1.1 protocol)

<content_package_len>: content data length of current URC

<content_package_string>: Content data string which is converted from content hex data.The length must be original content hex data size * 2



```
+EHTTPNMIC:0,0,9,18,313233313233313233
```

17.8. +EHTTPERR HTTP/HTTPS client connection error indicator

When the URC send, there is some error happen on the http client. Normally is TCP connection is disconnected.



```
<CR><LF>+EHTTPERR:<httpclient_id>,<error_code><CR><LF>
```

Parameter

<httpclient_id>: The indicator of http client instance created by the AT+EHTTPCREATE command

<error_code>: If it is -1, means disconnected by network problem. If it is -2, means connection is disconnected by server.

If the URC send out, the HTTP client will be disconnected automatically. If user want to send HTTP message to server, he must use AT+EHTTPCON command to reconnect.



```
+EHTTPERR:0,-2
```

17.9. Example: Create HTTP connection

```

*MATREADY: 1
+CFUN: 1
+CPIN: READY

+IP: 192.168.2.1
AT+EHTTPCREATE=0,41,41,"http://192.168.2.1:61613/","0,0,0," // Create a HTTP client
+EHTTPCREAT:0 //0: client id

OK
AT+EHTTPCON=0 //Connect to server
OK
AT+EHTTSEND=0,27,27,"0,0,9,/Test.txt",0,0,0," // Get the file with path is /Test.txt
OK

+EHTTPNMIH:0,404,157,Server: nginx/1.12.2 //The header part of response
Date: Thu, 20 Dec 2018 09:14:27 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

+EHTTPNMIC:0,0,129,258,7b2274696d657374616d70223a22323031382d31322d32305430393a31343a32372e3338302b303
03030222c22737461747573223a3430342c226572726f72223a224e6f7420466f756e64222c226d657373616765223a224e6f206
d65737361676520617661696c61626c65222c2270617468223a222f546573742e747874227d // The Content part of response,
the content of file Test.txt is ABCDEFG

AT+EHTTSEND=0,113,113,"0,1,9,/postjson",0,,16,"application/json",66,7b2274657374223a3132333435362c226d65737361
6765223a313333323432347d," //Send HTTP-post request
OK

+EHTTPNMIH:0,200,157,Server: nginx/1.12.2
Date: Thu, 20 Dec 2018 09:14:41 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

+EHTTPNMIC:0,0,33,66,7b2274657374223a3132333435362c226d657373616765223a313333323432347d

AT+EHTTSEND=0,77,77,"0,2,4,/put",0,,33,"application/x-www-form-urlencoded",18,746573743d74657374,"
OK //Send HTTP-put request

+EHTTPNMIH:0,200,143,Server: nginx/1.12.2
Date: Thu, 20 Dec 2018 09:14:57 GMT
Content-Type: text/plain;charset=UTF-8
Content-Length: 14

```

Connection: keep-alive

`AT+EHTTTPDISCON=0` //Disconnect HTTP

OK

`AT+EHTTTPDESTROY=0` //Destroy the http client instance

OK

17.10. Example: Create HTTPS connection

*MATREADY: 1

+CFUN: 1

+CPIN: READY

+IP: 100.124.192.22

`AT+EHTTTPCREATE=0,42,42,"https://219.144.130.26:61614/",,,0,,0,,0,"` //Create a HTTPS client

+EHTTTPCREAT:0

OK

`AT+EHTTTPCON=0` // Connect to server

OK

`AT+EHTTTPSEND=0,27,27,"0,0,9,/Test.txt",0,,0,,0,"` //Send HTTPS-get request

OK

+EHTTTPNMIH:0,404,157,Server: nginx/1.12.2 //The header part of response

Date: Thu, 20 Dec 2018 09:22:43 GMT

Content-Type: application/json;charset=UTF-8

Transfer-Encoding: chunked

Connection: keep-alive

+EHTTTPNMIH:0,0,129,258,7b2274696d657374616d70223a22323031382d31322d32305430393a32323a34332e3431392b30303030222c22737461747573223a3430342c226572726f72223a224e6f7420466f756e64222c226d657373616765223a224e6f206d65737361676520617661696c61626c65222c2270617468223a222f546573742e747874227d // The Content part of response,

`AT+EHTTTPSEND=0,113,113,"0,1,9,/postjson",0,,16,"application/json",66,7b2274657374223a3132333435362c226d657373616765223a313333323432347d,"` //Send HTTPS-post Request

OK

+EHTTTPNMIH:0,200,157,Server: nginx/1.12.2

Date: Thu, 20 Dec 2018 09:23:04 GMT

Content-Type: application/json;charset=UTF-8

Transfer-Encoding: chunked

Connection: keep-alive

```
+EHTTPNMIC:0,0,33,66,7b2274657374223a3132333435362c226d657373616765223a313333323432347d
AT+EHTTSEND=0,112,112,"0,2,8,"/putjson",0,,16,"application/json",66,7b2274657374223a3132333435362c226d657373616765223a313333323432347d," //Send HTTPS-put request
```

OK

```
+EHTTPNMIH:0,200,157,Server: nginx/1.12.2
Date: Thu, 20 Dec 2018 09:23:24 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
```

```
+EHTTPNMIC:0,0,33,66,7b2274657374223a3132333435362c226d657373616765223a313333323432347d
AT+EHTTSEND=0,33,33,"0,3,14,"/del?test=test",0,,0,,," //Send HTTPS-delete request
```

OK

```
+EHTTPNMIH:0,200,143,Server: nginx/1.12.2
Date: Thu, 20 Dec 2018 09:23:36 GMT
Content-Type: text/plain;charset=UTF-8
Content-Length: 14
Connection: keep-alive
```

```
+EHTTPNMIC:0,0,14,28,746573742069732064656c657465
```

```
AT+EHTTPDISCON=0 // Disconnect HTTP
```

OK

```
AT+EHTTPDESTROY=0 // Destroy the http client instance
```

OK

18. HTTP/HTTPS SIMPLIFIED COMMANDS

Note:

1. It is prohibited to mix the HTTP/HTTPS simplified commands in this chapter with the HTTP/HTTPS related commands in Chapter 17. Only one set of commands can be used at the same time.
2. This chapter is a new simplified HTTP/HTTPS command. It is recommended that customers use this chapter command to implement HTTP related functions. The commands in Chapter 17 are mainly used to be compatible with older versions of HTTP functions.
3. If your module does not support this simplified instruction, please try to upgrade to the latest firmware version.

18.1. AT+HTTPCFG Configure Parameters for HTTP(S) connection

The command is used to configure the parameters for HTTP(S) server, including configuring customizing HTTP(S) request header, outputting HTTP(S) response header, timeout and SSL settings. If the Write Command only executes one parameter, it will query the current settings.

It will take effect immediately after setting, and will not be saved after power off. It must be configured before AT+HTTPGET and AT+HTTPPOST.



```
+HTTPCFG=response_header[,<switch>]
```

Response

If <switch> is not omitted, set whether to report the response message header:

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

If <switch> is omitted, query the current settings:

```
<CR><LF>+HTTPCFG: response_header,<switch><CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

```
+HTTPCFG=custom_header[,<item>]
```

Response

If <item> is not omitted, set customized HTTP(s) request header:

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

If <item> is omitted, and not set or clean the former setting, response:

```
<CR><LF>OK<CR><LF>
```

If <item> is omitted, query the current settings:

```
<CR><LF>+HTTPCFG: custom_header,<item><CR><LF>
```

...

```
<CR><LF>OK<CR><LF>
```

```
+HTTPCFG=timeout[,<timeout_value>]
```

Response

If <timeout_value> is not omitted, set timeout of response message:

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

If <timeout_value> is omitted, query the current settings:

```
<CR><LF>+HTTPCFG: timeout,<timeout_value><CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

```
+HTTPCFG=report_mode[,<report_mode_value>]
```

Response

If <report_mode_value> is not omitted, set report mode of response message:

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

If <report_mode_value> is omitted, query the current settings:

```
<CR><LF>+HTTPCFG:report_mode,<report_mode_value><CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

```
+HTTPCFG=tls_version[,<tls_version_value>]
```

Response

If <tls_version_value> is not omitted, set the version of TLS:

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

If <tls_version_value> is omitted, query the current settings:

```
<CR><LF>+HTTPCFG:tls_version,<tls_version_value><CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

```
+HTTPCFG=tls_security_level[,<tls_security_level_value>]
```

Response

If <tls_security_level_value> is not omitted, set the TLS authentication method:

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

If <tls_security_level_value> is omitted, query the current settings:

```
<CR><LF>+HTTPCFG:tls_security_level,<tls_security_level_value><CR><LF>
```

```
<CR><LF>OK<CR><LF>
```

```
+HTTPCFG=tls_ca_cert[,<operation>]
```

Response

If <operation> is not omitted, set or clean TLS CA certification.

When the value of <operation> is an integer, it represents the length of the CA cert. In this case, enter the transparent transmission mode to configure the CA cert. Return:

```
<CR><LF>CONNECT<CR><LF>
```

```
<content> //input certification content, won't report or echo through AT port  
[receive 350 data<CR><LF>
```

```
...]
```

```
receive xxx data<CR><LF>
```

```
<CR><LF>NO CARRIER<CR><LF>
```

If <operation>= "clean", clean TLS CA certification:

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

If <tls_ca_cert_len> is omitted, query the current length of TLS CA:
 <CR><LF>+HTTPCFG: tls_ca_cert,<tls_ca_cert_len><CR><LF>
 <CR><LF>OK<CR><LF>

+HTTPCFG=tls_client_cert[,<operation>]

Response

If <operation> is not omitted, set or clean TLS client cert

When the value of <operation> is an integer, it represents the length of the TLS client cert. In this case, enter the transparent transmission mode to configure the TLS client cert. Return:

<CR><LF>CONNECT<CR><LF>

< content> //input certification content, won't report or echo through AT port

[receive 350 data<CR><LF>

...]

receive xxx data<CR><LF>

<CR><LF>NO CARRIER<CR><LF>

If <operation>="clean",clean TLS client certification:

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

If <tls_client_cert_len> is omitted, query the current length of TLS client cert:

<CR><LF>+HTTPCFG: tls_client_cert,<tls_client_cert_len><CR><LF>

<CR><LF>OK<CR><LF>

+HTTPCFG=tls_client_key[,<operation>]

Response

If <operation> is not omitted, set or clean TLS client key

When the value of <operation> is an integer, it represents the length of the TLS client key. In this case, enter the transparent transmission mode to configure the TLS client key. Return:

<CR><LF>CONNECT<CR><LF>

< content> //input certification content, won't report or echo through AT port

[receive 350 data<CR><LF>

...]

receive xxx data<CR><LF>

<CR><LF>NO CARRIER<CR><LF>

If <operation>="clean",clean TLS client key:

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

If <tls_client_key_len> is omitted, query the current length of TLS client key:

<CR><LF>+HTTPCFG: tls_client_key,<tls_client_key_len><CR><LF>

<CR><LF>OK<CR><LF>

Parameters

<switch>: Numeric type. Disable or enable to customize HTTP(S) response message

0 Disable

1 Enable(default value)

<item>: String type. A single custom header used for request messages, or a special value "clean" to clear all custom headers. This parameter needs to follow the HTTP protocol format. The format

is: "Accept: text/xml".

<timeout_value>:timeout of HTTP/HTTPS received data, the range is [5,300], unit: second, and is 30s in default. This parameter does not represent the response timeout of AT commands.

<report_mode_value>: Report mode, the range is [0,1]:

- 0 report in hex format(default value)
- 1 report in string format

Note: this mode can only control the content part of the HTTP response message, the header part is still reported in string format.

<tls_version_value>: TLS version, only available for HTTPS:

Value	Meaning
1	TLS1.0
2	TLS1.1
3	TLS1.2 (default value)

<tls_security_level_value>: TLS authentication method ,only valid for HTTPS:

Value	Meaning
0	No authentication (default value)
1	Optional certification
2	Must be certified

Parameters

<switch>: Numeric type. Disable or enable to customize HTTP(S) response message

- 0 Disable
- 1 Enable(default value)

<item>: String type. A single custom header used for request messages, or a special value "clean" to clear all custom headers. This parameter needs to follow the HTTP protocol format. The format is: "Accept: text/xml".

<timeout_value>:timeout of HTTP/HTTPS received data, the range is [5,300], unit: second, and is 30s in default. This parameter does not represent the response timeout of AT commands.

<report_mode_value>: Report mode, the range is [0,1]:

- 0 report in hex format(default value)
- 1 report in string format

Note: this mode can only control the content part of the HTTP response message, the header part is still reported in string format.

<tls_version_value>: TLS version, only available for HTTPS:

- | Value | Meaning |
|-------|------------------------|
| 1 | TLS1.0 |
| 2 | TLS1.1 |
| 3 | TLS1.2 (default value) |

<tls_security_level_value>: TLS authentication method ,only valid for HTTPS:

- | Value | Meaning |
|-------|-----------------------------------|
| 0 | No authentication (default value) |
| 1 | Optional certification |
| 2 | Must be certified |

<operation>: This parameter used to configure or clean TLS CA cert/TLS client cert/ TLS client key.

1. When the first parameter is `tls_ca_cert`, configure or clean TLS CA cert. `<operation>` can be integer or "clean", respectively used to configure or clean TLS CA cert. For example, when send "AT+HTTPCFG=tls_ca_cert,2094" will configure a CA cert of 2094 bytes, while "AT+HTTPCFG=tls_ca_cert,clean" clean TLS CA cert.
2. When the first parameter is `tls_client_cert`, configure or clean TLS client cert. `<operation>` can be integer or "clean", respectively used to configure or clean TLS client cert. For example, when send "AT+HTTPCFG=tls_client_cert,1980", will configure a CA cert of 1980 bytes; while "AT+HTTPCFG=tls_client_cert,clean" clean TLS client cert
3. When the first parameter is `tls_client_key`, configure or clean TLS client key, `<operation>` can be integer or "clean", respectively used to configure or clean TLS client key. For example, when send "AT+HTTPCFG=tls_client_key,3243", will configure a CA cert of 3243 bytes; while "AT+HTTPCFG=tls_client_key,clean" clean TLS client key.

`<tls_ca_cert_len>`: TLS CA certificate size, only valid for HTTPS, string type, maximum length is 3600, the configured CA certificate length should be the same as the actual CA certificate length transparently transmitted into the module. If the actual length of transparent transmission is greater than the configured length, it will cause the module to crash and restart.

`<tls_client_cert_len>`: TLS client certificate size, only valid for HTTPS, string type, maximum length is 3600, the configured client certificate length should be the same as the actual client certificate length transparently transmitted into the module. If the actual length of transparent transmission is greater than the configured length, it will cause the module to crash and restart.

`<tls_client_key_len>`: TLS client private key size, only valid for HTTPS, string type, maximum length is 3600, the configured client key length should be the same as the actual client key length transparently transmitted into the module. If the actual length of transparent transmission is greater than the configured length, it will cause the module to crash and restart.

Note:

1. Use AT+HTTPCFG=custom_header,<item> can only configure 1 header at a time, and support up to 10 configurations.
2. When using AT+HTTPCFG=custom_header,<item> to configure an existing header attribute, it will overwrite the previously configured value.
3. When item is a special value "clean", it means to clear all configured custom headers.



```

////////////////////////////////////
AT+HTTPCFG=response_header,0 // Disable to customize HTTP(S) response message header
OK
AT+HTTPCFG=response_header // query the current header settings
+HTTPCFG: response_header,0
OK
////////////////////////////////////

*****
AT+HTTPCFG=custom_header,"Accept: text/xml" //Set a single custom header used for request
messages. The type of data want to receive is text/xml
OK
AT+HTTPCFG=custom_header,"Connection: keep-alive"
OK

```

```

AT+HTTPCFG=custom_header // query the current header settings
+HTTPCFG: custom_header,"Accept: text/xml"
+HTTPCFG: custom_header,"Connection: keep-alive"
OK
AT+HTTPCFG=custom_header,"clean" // clear all configured custom headers.
OK
AT+HTTPCFG=custom_header
OK
*****
////////////////////////////////////
AT+HTTPCFG=timeout,60 // Set the response timeout to 60 seconds
OK
AT+HTTPCFG=timeout // query the current response timeout
+HTTPCFG: timeout,60
OK
////////////////////////////////////
*****
AT+HTTPCFG=report_mode,1 // set string report mode of response message content
OK
AT+HTTPCFG=report_mode // query the current report mode
+HTTPCFG: report_mode,1
OK
*****
*****
AT+HTTPCFG=tls_version,1 //Set TLS version as 1.0
OK
AT+HTTPCFG=tls_version // query the current TLS version
+HTTPCFG: tls_version,1

OK
*****
////////////////////////////////////
AT+HTTPCFG=tls_security_level,1// Set TLS authentication method
OK
AT+HTTPCFG=tls_security_level // query the current TLS authentication method
+HTTPCFG: tls_security_level,1

OK
////////////////////////////////////
////////////////////////////////////
AT+HTTPCFG=tls_ca_cert,1326 //Configure CA
CONNECT

receive 350 data
receive 350 data
receive 350 data
receive 276 data

NO CARRIER

AT+HTTPCFG=tls_ca_cert // query the current CA certification length
+HTTPCFG: tls_ca_cert,1326

OK
////////////////////////////////////
*****
AT+HTTPCFG=tls_client_cert,1330 // Configure client certification

```

CONNECT

receive 350 data
receive 350 data
receive 350 data
receive 280 data

NO CARRIER

AT+HTTPCFG=tls_client_cert // query the current client certification length
+HTTPCFG: tls_client_cert,1330

OK

////////////////////////////////////

AT+HTTPCFG=tls_client_key,1700 // Configure client key

CONNECT

receive 350 data
receive 350 data
receive 350 data
receive 350 data
receive 300 data

NO CARRIER

AT+HTTPCFG=tls_client_key // query the current client key length
+HTTPCFG: tls_client_key,1700

OK

////////////////////////////////////

18.2. AT+HTTPURL Set URL of HTTP(S) Server



+HTTPURL=<url_str>

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

+HTTPURL?

Response

<CR><LF>+HTTPURL: <url_str><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameters

<url_str>: String type. HTTP URL, maximum length is 511 bytes. This parameter must follow the HTTP protocol format. <url_str> must start with http:// or https://, whether encryption is based on the URL. If the HTTP(s) server port number is not specified in <url_str>, the module uses the default HTTP port number, that is 80 of http://, 443 of HTTPS://



AT+HTTPURL="http://www.test.com/file/1/" // Set URL of HTTP Server

OK

AT+HTTPURL? //Query the URL setting

+HTTPURL:http://www.test.com/file/1/

OK

18.3. AT+HTTPGET Send GET Request to HTTP(S) Server

Send GET Request to HTTP(S) Server. You need to set URL of server first, or it will report +HTTPGET: 4 after OK.



+HTTPGET

Response

When success, response:

<CR><LF>OK<CR><LF>

[<HTTP RESPONSES>]

<CR><LF>+HTTPGET: <result_code>[,<response_len>]<CR><LF>

When failed, response:

<CR><LF>ERROR<CR><LF>

Parameter

<HTTP RESPONSES> :HTTP response information, you can refer HTTP protocol specification for detail information.The response can be null, and not report this parameter at this time.

<result_code>: Result code of HTTP response message. 0 means success, please refer to 18.5 for detail

<response_len>: When <result_code> is 0, this parameter indicates the length of the response message actually received. When <result_code> is not 0, this parameter is not displayed.



*MATREADY: 1

+CFUN: 1

+CPIN: READY

+IP: 100.90.214.81

AT+HTTPCFG=response_header,1

OK

AT+HTTPCFG=report_mode,1

OK

AT+HTTPURL="http://219.144.130.26:61613/"

OK

AT+HTTPGET

OK

HTTP/1.1 200

Server: nginx/1.12.2

Date: Mon, 20 Apr 2020 02:05:43 GMT

Content-Type: text/plain;charset=UTF-8

Content-Length: 30

Connection: keep-alive

Abcdefghilmnopqrstuvwxyz1234567890

+HTTPGET:0,30

18.4. AT+HTTPPOST Send POST Request to HTTP(S) Server

Send POST Request to HTTP(S) Server, input POST data after "CONNECT" reported, once the input data length reach <content_length> value, it will end automatically. You need to set URL of server first, or it will report +HTTPGET: 4 after OK.



+HTTPPOST=<content_length>[,<input_timeout>]

Response
 <CR><LF>CONNECT<CR><LF>
 <content>
 [receive 350 data<CR><LF>
 ...]
 receive xxx data<CR><LF>

When parameters right, return:
 <CR><LF>OK<CR><LF>
 [<HTTP RESPONSES>]
 <CR><LF>+HTTPPOST: <result_code>[,<response_len>]<CR><LF>

When parameters error, or the length of <content> is less than <content_length> value and wait timeout, it will return:
 <CR><LF>ERROR<CR><LF>

Parameter
 <content_length>: the POST data length in byte, range is [0,1024]. When it's 0, the module won't enter transparent mode. At this time, even if the second parameter <input_timeout> is within the value range, it does not take effect because the transparent mode is not entered at this moment.
 <input_timeout>: timeout for data inputting in second, the value range [5,60], if you do not enter this parameter, the default value is 30 seconds.
 <content>: input POST data, when the parameter length reaches the length of <content_length>, it will automatically end. Not displayed in the returned results.
 <HTTP RESPONSES>: HTTP response information, you can refer HTTP protocol specification for detail information. The response can be null, and not report this parameter at this time.
 <result_code>: Result code of HTTP response message. 0 means success, please refer to 18.5 for detail
 <response_len>: When <result_code> is 0, this parameter indicates the length of the response message actually received. When <result_code> is not 0, this parameter is not displayed.

+HTTPPOST=?
 Response
 <CR><LF>+HTTPPOST: (0,1024)[,(0,60)]<CR><LF>
 <CR><LF>OK<CR><LF>



```
*MATREADY: 1
+CFUN: 1
+CPIN: READY

+IP: 100.82.27.165
AT+HTTPCFG=response_header,1
OK

AT+HTTPCFG=report_mode,1
OK

AT+HTTPCFG=custom_header,"Content-Type: application/json"
```

OK

AT+HTTPURL="http://219.144.130.26:61613/postjson" /* HTTP */
OK

AT+HTTPPOST=33,60

CONNECT
receive 33 data

NO CARRIER
OK

HTTP/1.1 200

Server: nginx/1.12.2
Date: Mon, 20 Apr 2020 02:40:09 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

{"test":123456,"message":1332424}

+HTTPPOST:0,0
AT+HTTPURL="https://219.144.130.26:61614/postjson" /* HTTPS */
OK

AT+HTTPPOST=33,60

CONNECT
receive 33 data

NO CARRIER
OK

HTTP/1.1 200

Server: nginx/1.12.2
Date: Mon, 20 Apr 2020 02:40:09 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

{"test":123456,"message":1332424}

+HTTPPOST:0,0

18.5. Result code of HTTP/HTTPS

The value and meaning of the HTTP <result_code> are shown in the following list:

<result_code>	Meaning
0	Operation successful
1	Failed to open URL
2	Failed to add request header
3	Failed to adding request body when execute command AT+HTTPPOST
4	URL not set
5	Failed to execute request method

19. NIDD COMMANDS

19.1. AT+NIDD Non-IP Data Delivery AT command



AT+NIDD=<cause_num>,<apn_name>[,<data>]

Response

<CR><LF> OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<cause_num>integer type

1create NIDD

2send data

< apn_name >string type , apn_name

<data> string type, send data



AT+CFUN=0 // 1 enter flying mode

OK

AT*MCGDEFCONT="Non-IP","cmiot" // 2 config NIDD apn name

OK

AT+CFUN=1 // 3 attach

OK

+CPIN: READY

AT+NIDD=1,cmiot // 4 create NIDD count

OK

AT+EGACT=1,4,"cmiot","web","password" // 5 active NIDD APN

+EGACT:1

OK

AT+NIDD=2,cmiot,1234567890abcdefghijklmnopqrstuvwxy // 6 send data

20. LWM2M COMMANDS

20.1. AT+ELMREG Configuration LwM2M instance and establish connection

This command is used to configure an LwM2M instance and establish connection, Only one instance can be created.



```
AT+ELMREG=<bs>,<s_host>,<s_port>,<epname>,<lifetime>[,<pskid>,<psk>]
```

Response

```
<CR><LF> +ELMREG=<lwm2m_id><CR><LF>
```

```
<CR><LF> OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

Parameter

<bs>: Integer; set bootstrap, 0: for false; 1: for true;

<s_host>: string; server host address; maximum length 40; This parameter can support domain name too, but you need to set available domain server firstly by command AT+EDNSSET

<s_port>: string; server port;

<epname>: string; endpoint name; maximum length 32;

<lifetime>: Integer; lifetime value, unit s, value range: 90 to 2592000 (30 days); when the value is not within the range, the lifetime value is equal to 100; when bs is 1, this value does not take effect, the lifetime is determined by the server. (When the lifetime is less than or equal to 40, it is multiplied by 0.8. When the lifetime is less than 90, it is multiplied by 0.9). It can be re-set by the ELMUPDATE command.

<pskid>: string; pre-shared key indentivity; provided by encryption method; maximum length 32;

<psk>: string(HEX format); pre-shared key; required to be encrypted; maximum length 32;



//host is the IP address, bootstrap is false

```
AT+ELMREG=0,"192.168.2.1","5683","T868613030014676",300
```

```
+ELMREG=0
```

```
OK
```

```
+ELMEVT:connect success
```

```
+ELMEVT:registering
```

```
+ELMEVT:register success
```

//host is the domain name, bootstrap is false

```
AT+ELMREG=0,"enr-api.devicewise.com","5683","T868613030014676",300
```

```
+ELMREG=0
```

```
OK
```

```
+ELMEVT:connect success
```

```
+ELMEVT:registering
```

```
+ELMEVT:register success
```

///host is the IP address, bootstrap is true

```
AT+ELMREG=1,"54.209.23.144","5683","BS868613030014677",300
```

```
+ELMREG=0
```

OK

+ELMEVT:connect success

+ELMEVT:bootstrapping

+ELMEVT:lifetime changed,value:21600

+ELMEVT:bootstrap success

+ELMEVT:connect success

+ELMEVT:registering

+ELMEVT:register success

// host is the domain name, bootstrap is true

AT+ELMREG=1,"bs-engr.telit.io","5683","BS868613030014677",300

+ELMREG=0

OK

+ELMEVT:connect success

+ELMEVT:bootstrapping

+ELMEVT:lifetime changed,value:21600

+ELMEVT:bootstrap success

+ELMEVT:connect success

+ELMEVT:registering

+ELMEVT:register success

//PSK

AT+ELMREG=0,"54.209.23.144","5684","PSK868613030014676",300,"868613030014676","3132333536"

+ELMREG=0

OK

+ELMEVT:connect success

+ELMEVT:registering

+ELMEVT:register success

//PSK, host is the IP address, bootstrap is true

AT+ELMREG=1,"54.209.23.144","5684","BSPSK868613030014678",300,"868613030014678","363534333231"

+ELMREG=0

OK

+ELMEVT:connect success

+ELMEVT:bootstrapping

+ELMEVT:lifetime changed,value:21600

```
+ELMEVT:bootstrap success
```

```
+ELMEVT:connect success
```

```
+ELMEVT:registering
```

```
+ELMEVT:register success
```

20.2. AT+ELMDEREG Delete LwM2M instance and disconnect

This command used to delete a LwM2M instance and disconnect. Local resources are released regardless of the report is deregister success or failed.



```
AT+ELMDEREG=<lwm2m_id>
```

Response

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

Parameter

<lwm2m_id> integer, LwM2M id, AT+ELMREG's response.



```
AT+ELMDEREG=0
```

```
OK
```

```
+ELMEVT:deregister success
```

```
+ELMEVT:clean complete
```

20.3. AT+ELMBINDING Configuration binding mode

This command is used to configure the binding mode(NON/CON Mode) of client for specified LwM2M instance. When not send this command, it means the data will be transferred in U mode.



```
AT+ELMBINDING=<lwm2m_id>,<binding>
```

Response

```
<CR><LF>OK<CR><LF>
```

or

```
<CR><LF>ERROR<CR><LF>
```

Parameter

<lwm2m_id>: lwm2m id, AT+ELMREGresponse.

<binding>:

- 0: U mode(UDP mode)
- 1: UQ mode(UDP queue mode)



```
AT+ELMBINDING=0,1
```

```
OK
```

```
+ELMEVT:set success
```

20.4. AT+ELMADDOBJ Add LwM2M object

This command is used to add LwM2M object.
You cannot add a basic object (0, 1, 3), you can only add up to two objects.



AT+ELMADDOBJ=<lwm2m_id>,<object_id>,<instance_id>

Response

<CR><LF> OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<lwm2m_id>integer type, LwM2M id,AT+ELMREG`'s response.

<object_id>integer type, object id.

<instance_id>integer type, instance id.



AT+ELMADDOBJ=0,3200,0

OK

+ELMEVT:update success

20.5. AT+ELMDELOBJ Delete LwM2M object

This command is used to delete LwM2M object.
You cannot delete a basic object (0, 1, 3).



AT+ ELMDELOBJ=< lwm2m_id >,< object_id>

Response

<CR><LF> OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<lwm2m_id>integer type, LwM2M id,AT+ELMREG`'s response.

<object_id>integer type, object id.



AT+ELMDELOBJ=0,3200

OK

+ELMEVT:update success

20.6. AT+ ELMUPDATE Update lifetime

This command is used to update lifetime. When the NAT ages and the server wants to send downlink data, the device can use this command to re-establish NAT.



AT+ELMUPDATE=<lwm2m_id>,[<lifetime>]

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<lwm2m_id>: LwM2Minstanceid, AT+ELMREG's response.

<lifetime>: the range is the same as <lifetime> of AT+ELMREG, if this value is omitted or out of range, this value is the last <lifetime> value.



AT+ELMUPDATE=0,200

OK

+ELMEVT:set success

AT+ELMUPDATE=0

OK

+ELMEVT:set success

20.7. +ELMREAD URC indicate server sending a read request

Unsolicited report command, indicate the server sends a read request



URC format description:

+ELMREAD:<lwm2m_id>,<object_id>,<instance_id>,<count>[,<resource_id>,<resource_id>,...]

Parameter

<lwm2m_id>integer type, LwM2M id,AT+ELMREG`'s response.

<object_id>integer type, object id.

<instance_id>integer type, instance id.

<count> integer, if this value is 0, it means multiple resources of the instance can be read.

<resource_id>integer type, resource id.



+ELMREAD:0,3200,0,1,5501

20.8. AT+ELMREADRSP Response to server's read request

This command is used to response to read request



AT+ELMREADRSP=<lwm2m_id>,<object_id>,<instance_id>,<resource_cnt>,<resource_id>,<value_type>,<len>,<value>
[,<resource_id>,<value_type>,<len>,<value>,...]

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<lwm2m_id>integer type, LwM2M id,AT+ELMREG`'s response.

<object_id>integer type, object id.

<instance_id>integer type, instance id.

< resource_cnt > integer,number of resources

<resource_id>integer type, resource id.

<value_type> string type, value type.

I:integer type;

F:Float type;

B:Boolean type;

D:opaque type;

S:String type;

<len>integer,Value length

<value> value type,Value context



+ELMREAD:0,3200,0,1,5501

20.9. +ELMWRITE URC indicate server sending a write request

Unsolicited report command, indicate the server sends a write request

URC format description:

```
+ELMWRITE:<lwm2m_id>,<object_id>,<instance_id>,<resource_cnt>,<resource_id>,<value_type>,<len>,<value>[,<resource_id>,<value_type>,<len>,<value>,...]
```

Parameter

<lwm2m_id>: integer; LwM2M instance id, +ELMREG returned result value;

<object_id>: integer; object id;

<instance_id>: integer; object instance id

<resource_count>: integer; the number of resources;

<resource_id>: integer; resource id;

<value_type>: string; value data type;

D: opaque type;

S: String type;

<len>: integer; value length;

<value>: value content;

Note:It's recommended that the length of <value> is not exceed 512 bytes.



```
+ELMWRITE: 0, 5, 2, 1,1,S, 5, "ABCDE"
```

```
AT+ ELMWRITE=0,0
```

```
OK
```

20.10. AT+ELMWRITERSP Response to server's write request

This command is used to response to write request



AT+ELMWRITERSP=<lwm2m_id>,<result>

Response

<CR><LF> OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<lwm2m_id>: integer; LwM2M instance id, +ELMREG returned result value;

<result> integer, write result,

1- failed (Code:4.05 Method Not Allowed) ,

0 - success (Code:2.04 Changed) .



AT+ELMWRITERSP=0,0

OK

20.11. +ELMEXECUTE URC indicate server sending a excute request

Unsolicited report command, indicate the server sends a excute request

URC format description:

+ELMEXECUTE:<lwm2m_id>,<object_id>,<instance_id>,<resource_id>[,<len>,<value>]

Parameter

<lwm2m_id>: integer; LwM2M instance id, +ELMREG returned result value;

<object_id>: integer; object id;

<instance_id>: integer; object instance id

<resource_id>: integer; resource id;

<len>: integer; value length;

<value>: content;



+ELMEXECUTE:0,3303,0,5605

20.12. AT+ELMEXECUTERSP Response to server's execute request

This command is used to response to execute request



Set command
AT+ELMEXECUTERSP=<lwm2m_id>,<result>

Response
<CR><LF>OK<CR><LF>
or
<CR><LF>ERROR<CR><LF>

Parameter
<lwm2m_id> integer, LwM2M id, AT+ELMREG's response.
<result> integer, result of execution command,
1- failed (Code:4.05 Method Not Allowed) ,
0 - success (Code:2.04 Changed) .



AT+ELMEXECUTERSP=0,0
OK

20.13. +ELMOBSERVEREQ URC indicate server sending a observe request

Unsolicited report command, indicate the server sends a observe request

URC format:
ELMOBSERVEREQ:<lwm2m_id>,<object_id>,<instance_id>,<count>[,<resource_id>,...]

Parameter
<lwm2m_id> integer, LwM2M id, AT+ELMREG's response.
<object_id> integer, object id.
<instance_id> integer, instance id.
<resource_id> integer, resource id, -1: all of resource about the instance.
<count>: Integer; if this value is 0, it means multiple resources of the subscription instance;



+ELMOBSERVEREQ:0,3203,0,1,5603

20.14. AT+ELMOBSERVERSP Response to server's observe request

This command is used to response to observe request



Set command

AT+ELMOBSERVERSP=<lwm2m_id>,<result>,<object_id>,<instance_id>,<resource_cnt>,<resource_id>,<value_type>,<len>,<value>[,<resource_id>,<value_type>,<len>,<value>...]

Response

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<lwm2m_id> integer, LwM2M id, AT+ELMREG's response.

<result> integer, result of observe request,

1- reject

0 - accept

<object_id> integer, object id.

<instance_id> integer, instance id.

<resource_count>: integer; number of resources; you can observe 25 resource at most.

<resource_id>: integer; resource id;

<value_type>: value data type;

I: integer;

F: Float;

B: Boolean;

S: String;

<len>: integer; value length;

<value>: value content;



```
AT+ELMOBSERVERSP=0,0,3200,0,1,5501,I,1,0
OK
```

```
AT+ELMOBSERVERSP=0,0,3200,0,1,5500,B,1,1
OK
```

```
AT+ELMOBSERVERSP=0,0,3200,0,1,5750,S,2,ab
OK
```

```
AT+ELMOBSERVERSP=0,0,3203,0,1,5603,F,3,1.0
OK
```

20.15. AT+ELMSEND Send data to server

After the observe succeeds, the device can use the command to send data to the server for the observed resource.



AT+ELMSEND=<lwm2m_id>,<confirm>,<object_id>,<instance_id>,<resource_id>,<value_type>,<len>,<value>

Response

<CR><LF>OK<CR><LF>

Or

<CR><LF>ERROR<CR><LF>

Parameter

<lwm2m_id>: integer; LwM2M instance id, +ELMREG returned value;

<confirm>: integer; 1:CON; 0:NON;

<object_id>: integer; object id;

<instance_id>: integer; object instance id;

<resource_id>: integer; resource id;

<value_type>: value type;

I: integer;

F:Float;

B: Boolean;

D:opaque;

S:String

<len>: integer; <value> length;

<value>: data content;



//int type

AT+ELMSEND=0,1,3200,0,5501,I,2,33

OK

+ELMEVT:send success

+ELMEVT:send confirm success //When <confirm> is CON, this URC is reported

//bool type

AT+ELMSEND=0,1,3200,0,5500,B,2,1

OK

+ELMEVT:send success

+ELMEVT:send confirm success // When <confirm> is CON, this URC is reported

//string type

AT+ELMSEND=0,1,3200,0,5750,S,2,"ab"

OK

+ELMEVT:send success

+ELMEVT:send confirm success // When <confirm> is CON, this URC is reported

//float type

AT+ELMSEND=0,1,3203,0,5603,F,3,1.1

OK

+ELMEVT:send success

+ELMEVT:send confirm success // When <confirm> is CON, this URC is reported

//opaque type

AT+ELMSEND=0,1,3200,0,5750,D,4,"4142"

OK

+ELMEVT:send success

+ELMEVT:send confirm success // When <confirm> is CON, this URC is reported

20.16. AT+ELMSTATUS Query the LwM2M instance status

This command is used to query the LwM2M instance status



AT+ELMSTATUS=<lwm2m_id>

Response

<CR><LF>+ELMSTATUS:<state><CR><LF>

<CR><LF>OK<CR><LF>

Or

<CR><LF>ERROR<CR><LF>

Parameter

<lwm2m_id>: integer; LwM2M instance id, +ELMREG returned value;

<state>:

- 0: not connected
- 1: bootstrap request
- 2: bootstrap processing
- 3: register request
- 4: register processing
- 5: connection succeeded

If report ERROR, it indicates that <lwm2m_id> does not exist.

20.17. +ELMEVT URC of information notification

URC	Description
+ELMEVT:connect success	Establish connection success
+ELMEVT:connect failed	Establish connection failed
+ELMEVT:bootstrapping	bootstrapping
+ELMEVT:bootstrap success	bootstrap success
+ELMEVT:bootstrap failed	bootstrap failed
+ELMEVT:registering	registering
+ELMEVT:register success	register success
+ELMEVT:register failed	register failed
+ELMEVT:deregister success	deregister success
+ELMEVT:deregister failed	deregister failed
+ELMEVT:lifetime changed,value:xx	In the bootstrap process, the server will issue a lifetime, report the value xx, and modify the RTC according to the value.
+ELMEVT:set success	Update lifetime/set binding mode successfully
+ELMEVT:set failed	Update lifetime/set binding mode failed
+ELMEVT:update success	Heartbeat packet sent / added objec / delete object successfully
+ELMEVT:update failed	Heartbeat packet sent/added objec/delete object failed
+ELMEVT:send success	The device sends data to the server successfully, and does not guarantee that the server receives successfully.
+ELMEVT:send failed	Device failed to send data to server
+ELMEVT:send confirm success	When CON is sent, the device successfully sends data to the server and receives a successful response from the server.
+ELMEVT:send confirm failed	When CON is sent, the device successfully sends data to the server and receives a failed response from the server.
+ELMEVT:clean complete	The local LwM2M resource has been removed

Note:

When report one of the responses of below, there will be followed process of removing local LwM2M resource, so it will report +ELMEVT:clean complete later.

1. When report +ELMEVT:deregister success or +ELMEVT:deregister failed
2. When report +ELMEVT:register failed
3. When report +ELMEVT:bootstrap failed
4. When report +ELMEVT:connect failed
5. When report +ELMEVT:update failed

20.18. Create LwM2M connection example

The following is an example of communication with leshan LwM2M server:

```
// Configuration LwM2M instance and establish connection
```

```
AT+ELMREG=0,"54.209.23.144","5683","T868613030014676",300
```

```
+ELMREG=0
```

```
OK
```

```
+ELMEVT:connect success
```

```
+ELMEVT:registering
```

```
+ELMEVT:register success
```

```
//Set binding mode to UQ
```

```
AT+ELMBINDING=0,1
```

```
OK
```

```
+ELMEVT:set success
```

```
//add object
```

```
AT+ELMADDOBJ=0,3200,0
```

```
OK
```

```
+ELMEVT:update success
```

```
AT+ELMADDOBJ=0,3303,0
```

```
OK
```

```
+ELMEVT:update success
```

```
//read operation
```

```
+ELMREAD:0,3200,0,1,5501 //read request URC
```

```
AT+ELMREADRSP=0,3200,0,1,5501,1,2,33 //read response operation
```

```
OK
```

```
//write operation
```

```
+ELMWRITE:0,19,0,1,4,S,11,"6,1,6,2,6,3" //write request URC
```

```
AT+ELMWRITERSP=0,0 // write response operation
```

```
OK
```


//execute operation

+ELMEXECUTE:0,3303,0,5605 //execute request URC

AT+ELMEXECUTERSP=0,0 //execute response operation

OK

//observe operation

+ELMOBSERVEREQ:0,3203,0,1,5603 //observe request

AT+ELMOBSERVERSP=0,0,3203,0,1,5603,F,3,1.2 //observe response operation

OK

//after observe successfully, send data to server.

AT+ELMSEND=0,1,3203,0,5603,F,3,1.1 //float type, CON

OK

+ELMEVT:send success

+ELMEVT:send confirm success // When <confirm> is CON, this URC is reported

//Delete object

AT+ELMDELOBJ=0,3303

OK

+ELMEVT:update success

//Delete Lwm2M instance

AT+ELMDEREG=0

OK

+LELMEVT:deregister success

+ELMEVT:clean complete

21. TLS NETWORK COMMANDS

21.1. AT+TLSCFG Configuration TLS Context Parameters

Configure TLS Context parameters. The parameters take effect once setting, and won't be saved after reboot.



```
AT+TLSCFG="host_name",<host_name>
Response
<CR><LF> OK<CR><LF>
or
<CR><LF>ERROR<CR><LF>
```



```
AT+TLSCFG="auth_mode",<auth_mode>
Response
<CR><LF> OK<CR><LF>
or
<CR><LF>ERROR<CR><LF>
```



```
AT+TLSCFG="handshake_timeout",<handshake_timeout>
Response
<CR><LF> OK<CR><LF>
or
<CR><LF>ERROR<CR><LF>
```



```
AT+TLSCFG="view_mode",<view_mode>
Response
<CR><LF> OK<CR><LF>
or
<CR><LF>ERROR<CR><LF>
```



```
AT+TLSCFG="ca_cert",<send_len>
Response
<CR><LF>CONNECT<CR><LF>
<CR><LF>receive <recv_len> data <CR><LF>

NO CARRIER
or
<CR><LF>ERROR<CR><LF>
```



```
AT+TLSCFG="cli_cert",<send_len>
Response
<CR><LF>CONNECT<CR><LF>
<CR><LF>receive <recv_len> data <CR><LF>

NO CARRIER
or
<CR><LF>ERROR<CR><LF>
```



```
AT+TLSCFG="cli_key",<send_len>
Response
<CR><LF>CONNECT<CR><LF>
<CR><LF>receive <recv_len> data <CR><LF>

NO CARRIER
or
<CR><LF>ERROR<CR><LF>
```

Parameters

<host_name>: String, configure the host name, the maximum length of the host name is 64 characters.

<auth_mode>: interger, the range is 0-2, Authentication mode,
 0, none
 1, optional
 2, required(default value)

<handshake_timeout>: interger, the encryption handshake's timeout for TLS connection, the unit is ms, and range is 5000-300000.Default vaule is 60000

<view_mode>: interger, range is 0-1, Set the format of received data
 0 original data(default value)
 1 Hexadecimal string

„ca_cert“ : Set CA certificate, adopt transparent transmission configuration; <send_len> is the length of CA certificate; <recv_len> is the length of the data received in a single pass.

„cli_cert“ : Set client certificate, adopt transparent transmission configuration; <send_len> is the length of client certificate; <recv_len> is the length of the data received in a single pass.

„cli_key“ : Client private key, adopt transparent transmission configuration; <send_len> is the length of Client private key; <recv_len> is the length of the data received in a single pass.



```
AT+TLSCFG="host_name",localhost // Set hostname
OK

AT+TLSCFG="auth_mode",2 // Set Authentication mode
OK

AT+TLSCFG="handshake_timeout",60000 // Set the encryption handshake's timeout for TLS connection
OK

AT+TLSCFG="view_mode",1 // Set the format of received data
OK

AT+TLSCFG="CA_CERT",1326 // Set CA certificate

CONNECT
receive 350 data
receive 350 data
receive 350 data
receive 276 data

NO CARRIER

OK
AT+TLSCFG="CLI_CERT",1330 // Set client certificate

CONNECT
receive 350 data
```

```

receive 350 data
receive 350 data
receive 280 data

NO CARRIER

OK
AT+TLSCFG="CLI_KEY",1700           // Set Client private key

CONNECT
receive 350 data
receive 350 data
receive 350 data
receive 350 data
receive 300 data

NO CARRIER

OK

```

21.2. AT+TLSOPEN Establish TLS connection

This command is used to establish a TLS connection with the remote server, and the status of the connection is indicated by the actively reported +TLSSTAT.



Set command

AT+ TLSOPEN=<host>,<port>

Response

<CR><LF>OK<CR><LF>

<CR><LF>+TLSSTAT:<state><CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<host>: string, server IP or domain name.

<port>: Integer, server port number.

<state>: Integer, the value range is 0-1, 0 means disconnected, 1 means connected.



```

AT+TLSOPEN=39.104.78.27,8557
+TLSSTAT:1

```

OK

21.3. AT+TLSCLOSE Close TLS connection

This command is used to close the TLS connection, and the connection status is indicated by the actively reported +TLSSTAT.



Set command

AT+TLSCLOSE

Response

<CR><LF>OK<CR><LF>

<CR><LF>+TLSSTAT:<state><CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<state>: Integer, the value range is 0-1, 0 means disconnected, 1 means connected.



AT+TLSCLOSE

+TLSSTAT:0

OK

21.4. AT+TLSEND Send data in ASCII code

This command is used to send data in ASCII code. The data can only be sent when the connection is established.



Set command

AT+TLSEND=<data>

Response

<CR><LF>+TLSEND:<length><CR><LF>

<CR><LF>OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameter

<data>: The data content in ASCII code, the data length is up to 512 bytes (the corresponding hexadecimal string length is up to 1024 characters), and the data is encoded in hexadecimal. For example: "48656C6C6F21" stands for "Hello!".

<length>: The length of the original data that has been sent.



AT+TLSEND=48656C6C6F21

+TLSEND:6

OK

21.5. AT+TLSENDDRAW Send original data

This command is used to send the original data. After successful execution, the module will receive the original data from the serial device until the received data length exceeds the set length or the reception timeout expires.



Set command

AT+TLSENDDRAW=<length>[,<timeout>]

Response

<CR><LF>>CONNECT<CR><LF>

<CR><LF>receive <recv_len> data <CR><LF>

NO CARRIER

<CR><LF>+TLSEND:<send_len><CR><LF>

OK

or

<CR><LF>ERROR<CR><LF>

Parameter

<length>: Integer, ranging from 1-512 to the length of the data to be transparently transmitted.

<timeout>: Integer, transparent transmission timeout, optional parameters, the default value is 30 seconds, the value ranges from 5 to 60, in seconds.

<recv_len>: integer, data length received in transparent mode

<send_len>: Integer, transparent transmission mode to send data length



AT+TLSENDDRAW=30,45

CONNECT

receive 30 data

NO CARRIER

+TLSEND:6

OK

21.6. AT+TLSRECV URC of Receive data

This command is an active report type and is used to report data received on a TLS connection. After receiving the data, it will be reported automatically. The maximum length of the data reported in a single time is 512 bytes (the corresponding hexadecimal string length is 1024).

URC format
`<CR><LF>+TLSRECV:<length>,<data><CR><LF>`

Parameter
`<length>`: The length of the reported data.
`<data>`: The reported data, the default format is the original data. If you need to convert to hexadecimal format display, please use `AT+TLSCFG="view_mode",1` command to modify the display mode to hexadecimal format.

```
</> +TLSRECV:11,hello world
AT+TLSCFG="view_mode",1
+TLSRECV:11,68656c6c6f20776f726c64
```

21.7. AT+TLSSTAT Get TLS connection state

This command is used to actively obtain the status of the TLS connection.



Set command

`AT+TLSSTAT`

Response

`<CR><LF>+TLSSTAT:<state><CR><LF>`

`<CR><LF>OK<CR><LF>`

or

`<CR><LF>ERROR<CR><LF>`

Parameter

`<state>`: Integer, the value range is 0-1, 0 means disconnected, 1 means connected.



`AT+TLSSTAT`

`+TLSSTAT:1`

OK

21.8. +TLSSTAT URC of TLS connection state

This command is used to unsolicited report the status of the TLS connection.

```

URC command
<CR><LF>+TLSSTAT:<state><CR><LF>
Parameter
<state>: Integer, the value range is 0-1, 0 means disconnected, 1 means connected.

```

```
</> +TLSSTAT:1
```

21.9. TLS connection example

```

AT+TLSOPEN=192.168.1.1,8557           //Established TLS connection
+TLSSTAT:1

OK

AT+TLSEND=746573745f636f6d6d616e645f73656e64 //Client send data in HEX format
+TLSEND:17

OK

+TLSRCV: 17,test_command_send         // Client receive original data

AT+TLSENDRAW=20,60                    // Client send original data

CONNECT                               // Enter transparent transmission state
receive 20 data                        // data length received in transparent mode
NO CARRIER                            // Exit transparent transmission state

+TLSEND:20                             // transparent transmission mode to send data length

OK

+TLSRCV: 20, test_command_send        // Client receive original data

AT+TLSSTAT                             // Get TLS connection state
+TLSSTAT:1                             //1 means connected

OK

AT+TLSCLOSE                             // Close TLS connection
+TLSSTAT:0

OK

```


22. NETWORK FRIENDLY MODE

22.1. AT+NFM NETwork Friendly Mode Enable/disable

The Communications Module shall allow the IoT Device Application to switch the Network Friendly Mode on and off using an AT-command. In addition, the module shall allow the application to switch the start timer on and off using the same command. Start timer can be used for e.g. smart metering applications to spread network wake-up attempts. This command will take effect after restart the module and it will be saved after restart.



AT+NFM=[<NFM Active>[,<Start Timer Active>]]

Response

<CR><LF> OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameters

<NFM Active>

- 0 - Deactivated
- 1 - Active

<Start Timer Active>

- 0 – Disable Start Timer
- 1 – Enable Start Timer

Start Timer:

- Start Timer applies only if active (<Start Timer Active > is 1).
- If Start Timer is enabled, the Start Timer will be started at every power cycle and the registration procedures will be allowed only at Start Timer expiry.

22.2. AT+NFM NETwork Friendly Mode Enable/disable

The Communications Module shall allow the IoT Device Application to configure the Back-off Base Intervals using +NFM C command.

This command will take effect after restart the module and it will be saved after restart.



AT+NFM C=[<NFMPar1>[,<NFMPar2>[,<NFMPar3>[,<NFMPar4>[,<NFMPar5>[,<NFMPar6>[,<NFMPar7>[,<STPar>]]]]]]]]]

Response

<CR><LF> OK<CR><LF>

or

<CR><LF>ERROR<CR><LF>

Parameters

- <NFMPar1> - NFM iteration counter 1 time interval in seconds 1-15360 – (default is 60);
- <NFMPar2> - NFM iteration counter 2 time interval in seconds 1-15360 – (default is 120);
- <NFMPar3> - NFM iteration counter 3 time interval in seconds 1-15360 – (default is 240);
- <NFMPar4> - NFM iteration counter 4 time interval in seconds 1-15360 – (default is 480);
- <NFMPar5> - NFM iteration counter 5 time interval in seconds 1-15360 – (default is 960);
- <NFMPar6> - NFM iteration counter 6 time interval in seconds 1-15360 – (default is 1920);
- <NFMPar7> - NFM iteration counter 7 time interval in seconds 1-15360 – (default is 3840);
- <STPar> - ST time interval in seconds 1-15360 – (default is 60);

Start Timer:

- If STPar is the number contained in the parameter <STPar> then the value of the ST timer is calculated with the following formula Start Timer = 1 + (IMSI % STPar). The IMSI get by command +CIMI is in format of HEX, for example, 0x460040841064151 change to Decimal format is 315256407413113169, so the start timer= 1+315256407413113169 %60 =30s.

23. LIST OF ACRONYMS

Acronym	Meaning
ARFCN	Absolute Radio Frequency Channel Number
AT	Attention command
BA	BCCH Allocation
BCCH	Broadcast Control Channel
CA	Cell Allocation
CBM	Cell Broadcast Message
CBS	Cell Broadcast Service
CCM	Current Call Meter
CLIR	Calling Line Identification Restriction
CTS	Clear To Send
CUG	Closed User Group
DCD	Data Carrier Detect
DCE	Data Communication Equipment
DCS	Digital Cellular System
DGPS	Differential GPS, the use of GPS measurements, which are differentially corrected
DNS	Domain Name System
DSR	Data Set Ready
DTE	Data Terminal Equipment
DTMF	Dual Tone Multi Frequency
DTR	Data Terminal Ready
GGA	GPS Fix data
GLL	Geographic Position – Latitude/Longitude
GLONASS	Global positioning system maintained by the Russian Space Forces
GMT	Greenwich Mean Time
GNSS	Any single or combined satellite navigation system (GPS, GLONASS and combined GPS/GLONASS)
GPRS	Global Packet Radio Service
GPS	Global Positioning System
GSA	GPS DOP and Active satellites
GSM	Global System Mobile
GSV	GPS satellites in view
HDLC	High Level Data Link Control
HDOP	Horizontal Dilution of Precision
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IRA	International Reference Alphabet
IWF	Interworking Function
ME	Mobile Equipment
MO	Mobile Originated
MT	<i>either</i> Mobile Terminated <i>or</i> Mobile Terminal
NMEA	National Marine Electronics Association
NVM	Non-Volatile Memory
PCS	Personal Communication Service
PDP	Packet Data Protocol
PDU	Packet Data Unit
PIN	Personal Identification Number
PPP	Point to Point Protocol
PUK	Pin Unblocking Code
RLP	Radio Link Protocol
RMC	Recommended minimum Specific data
RTS	Request To Send
SAP	SIM Access Profile
SCA	Service Center Address
SMS	Short Message Service
SMSC	Short Message Service Center
SMTP	Simple Mail Transport Protocol
TA	Terminal Adapter
TCP	Transmission Control Protocol
TE	Terminal Equipment
UDP	User Datagram Protocol
USSD	Unstructured Supplementary Service Data
UTC	Coordinated Universal Time
VDOP	Vertical dilution of precision
VTG	Course over ground and ground speed
WAAS	Wide Area Augmentation System

24. DOCUMENT HISTORY

Revision	Date	Changes
0	2019-07-01	First issue
1	2019-08-14	Updated overall document Added NL865H2
2	2019-09-18	Removed EDNS command
3	2019-11-08	Added Watermark Added commands: #I2CWR, #I2CRD, #SPIOOPEN, #SPICLOSE, #SPIRW, #SYSHALT, #SHDN, +CEER, +ESODATAMODE, +EMQSCFGRAW, +EMQSCFG, +CBC, #SYSHALT, +NFM, +ELMUPDATE, +ELMCONFIRM, SMS relative commands Updated: #GPIO, #TADC Deleted: +CPLS
4	2020-03-18	Updated: +ELMEXECUTE, +ELMADDOBJ, +CEDRXS Added: ATS95, +TCOMWRT
5	2020-04-21	Added *MLPINFO, +FTPPUTDATA, +FTPFCFG, +FTPGETOTA, +OTAUP, AT&C, AT&D, ATN1, ATS0, ATS1, ATS7, #TSLK, #E2RI, +IISTEST Updated LwM2M commands, TLS commands, SMS Commands, *MFRCLLCK, +EMQNEW, +ESOC, #GPIO, +CMUX, +CNUM, +GCAP, ATS10, ATS25, ATX, +FCLASS, #TCONTLED, ATV, +CTZR, +ESOSEND
6	2020-07-27	Added: #TCLRBDRT, +ESODATAMODE simplified HTTP/HTTPS commands, FOTA md5 verification result code report Updated: #E2RI, #TCONTLED, ATV, ATS7, #TURNOFF, +GCAP, #I2CWR, #GPIO, +ESNTPSTART, +ELMBINDING, +ELMUPDATE, #TSLK, +CGCONTRDP Removed: +IISTEST
7	2020-08-18	Updated: FTP commands (chapter 21)



SUPPORT INQUIRIES

Link to www.telit.com and contact our technical support team for any questions related to technical issues.

www.telit.com



Telit Communications S.p.A.
Via Stazione di Prosecco, 5/B
I-34010 Sgonico (Trieste), Italy

Telit Wireless Solutions Inc.
3131 RDU Center Drive, Suite 135
Morrisville, NC 27560, USA

Telit Wireless Solutions Ltd.
10 Habarzel St.
Tel Aviv 69710, Israel

Telit IoT Platforms LLC
5300 Broken Sound Blvd, Suite 150
Boca Raton, FL 33487, USA

Telit Wireless Solutions Co., Ltd.
8th Fl., Shinyoung Securities Bld.
6, Gukjegeumyung-ro8-gil, Yeongdeungpo-gu
Seoul, 150-884, Korea

Telit Wireless Solutions
Tecnologia e Servicos Ltda
Avenida Paulista, 1776, Room 10.C
01310-921 São Paulo, Brazil

Telit reserves all rights to this document and the information contained herein. Products, names, logos and designs described herein may in whole or in part be subject to intellectual property rights. The information contained herein is provided "as is". No warranty of any kind, either express or implied, is made in relation to the accuracy, reliability, fitness for a particular purpose or content of this document. This document may be revised by Telit at any time. For most recent documents, please visit www.telit.com

Copyright © 2016, Telit

Mod. 0805 2017-01 Rev.6