骐俊物联
CHEERZING
证券代码：833504

ML5515 ML5535
NB-IoT Module

AT Commands Manual

Rev：1.5
2019/1/3

Title		ML5515 ML5535 NB-IoT Module AT Commands Manual			
Revision		1.5			
Date		2019/01/03			
Doc ID		ML5515 ML5535_AT_CM_R1.1			
Status					
拟制	Wangyihuang	审核	zhuzhiqiang	批准	Wudeqing

Revision History

Revision	Date	Subjects (major changes)
0.1	2018-05-14	Initial
0.2	2018-06-22	CTCC IoT Related Commands
1.0	2018-08-06	add CMCC IoT Related Commands
1.1	2018-08-09	add AT+CZIOTP
1.2	2018-10-23	add AT+ MLWM2MENABLE
1.2	2018-10-23	add AT+MCRITICALDATA
1.2	2018-10-23	add AT+NITZ
1.2	2018-10-23	add AT+CPIN
1.2	2018-10-23	add AT+CPINR
1.2	2018-10-23	Modify AT+CCLK
1.3	2018-12-27	add AT+CZADC
1.4	2018-12-27	Modify AT+NCONFIG
1.5	2019-01-03	Add HI2115B500 AT Commands

Copyright and permission notice

Disclaimers

This document is based on the information made available, its content is subject to change without prior notice. In preparing the document, the company has made efforts to make its content is accurate and reliable, without omission, inaccuracy in this document or layout errors assumes no liability for loss and damage

Contents

1 Introduction

This document gives details of the AT Command Set supported by Cheerzing ML5515 and ML5535 series NB-IoT module.

At boot the following string will be output:

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

After this string has been received the AT Command processor is ready to accept AT commands.

If the Applications core was rebooted or restarted for any reason that wasn't a normal power-on sequence, a message will be output before the <CR><LF>Cheerzing<CR><LF>OK<CR><LF> string. See Chapter 0 for more details.

In case external MCU intervene the process of update, unsolicited information informs the external MCU update state. Please reference AT document [1] for more detail information

<CR><LF>FIRMWARE DOWNLOADING<CR><LF>	Indicates UE is downloading update package.
<CR><LF>FIRMWARE DOWNLOAD FAILED<CR><LF>	Indicates download failed.
<CR><LF>FIRMWARE DOWNLOADED<CR><LF>	Indicates download finished.
<CR><LF>FIRMWARE UPDATING<CR><LF>	Indicates UE is updating.
<CR><LF>FIRMWARE UPDATE SUCCESS<CR><LF>	Indicates update success, but not report update state to firmware package server yet.
<CR><LF>FIRMWARE UPDATE FAILED<CR><LF>	Indicates UE update fail.
<CR><LF>FIRMWARE UPDATE OVER<CR><LF>	Indicates reported update state to firmware package server.

During FOTA procedure, device shouldn't operate modem until " FIRMWARE UPDATE OVER" is reported. E.g. AT+NRB, power off are not permitted, otherwise, indescribable error will be made.[1]

1.1 Definitions

- <CR>

carriage return character

- <LF>

line feed character

- <..>parameter name. Angle brackets do not appear on command line
- [..] Option parameter. Square brackets do not appear on the command line

1.2 AT Command Syntax

AT+<cmd>	Execute Command	Execute Command
AT+<cmd>=p1[,p2[,p3[.....]]]	Set Command	Set Command
AT+<cmd>?	Read Command	Check current sub-parameter values
AT+<cmd> =?	Test Command	Check possible sub-parameter values

Multiple commands can be placed on a single line using a semi-colon (';') between commands. Only the first command should have AT prefix. Commands can be in upper or lower case.

When entering AT commands spaces are ignored except in the following cases:

- within quoted strings, where they are preserved
- within an unquoted string, numeric parameter
- within an IP address
- within the AT command name upto and including a '=', '?' or '=?'

They can be used to make the input more human readable. On input, at least a carriage-return is required. A new-line character is ignored so it is permissible to use carriage-return line-feed pairs on the input.

If no command is entered after the AT token, OK will be returned. If an invalid command is entered, ERROR will be returned.

Optional parameters, unless explicitly stated, need to be provided up to the last parameter being entered.

1.3 3GPP Alignment

3GPP commands are aligned to 3GPP TS 27.007 v14.3.0 (2017-03). For clarification on 3GPP commands, please refer to this document.

1.4 AT Command Responses

When the AT Command processor has finished processing a line it will output either OK, ERROR or an error code indicating that it is ready to accept a new command. Solicited informational responses are sent before the final OK or ERROR. Unsolicited information responses will never occur between a solicited informational response and the final OK or ERROR.

For AT Commands aligned to the 3GPP 27.007 specification error codes are enabled with the CMEE command, else the ERROR message will be returned. For AT Commands aligned to the 3GPP 27.005 specification, a CMS error will be returned where specified.

Responses will be of the format:

```
<CR><LF>+CMD1:<parameters><CR><LF>
<CR><LF>OK<CR><LF>
```

or

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

2 Commands (3GPP aligned to 27.007)

2.1 Request Manufacturer Identification (+CGMI)

Command	Response	Example
+CGMI	<manufacturer> +CME ERROR:<err>	AT+CGMI Cheerzing OK
+CGMI=?		AT+CGMI=? OK

Description

This command returns the manufacturer information. By default this will return “Cheerzing” on the standard platform. Refer to Chapter 6: Error Values for possible <err> values.

Defined values

`<manufacturer>`: the total number of characters, including line terminators, in the information text shall not exceed 2048 characters.

Text shall not contain the sequence `0<CR>` or

`OK<CR>` **Implementation**

2.2 Request Manufacturer Model (+CGMM)

Command	Response	Example
+CGMM	<model> +CME ERROR:<err>	AT+CGMM ML5515 OK
+CGMM=?		AT+CGMM=? OK

Description

This command returns the manufacturer model information. By default this will return “ML5515” on the standard platform. Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

`<model>`: the total number of characters, including line terminators, in the information text shall not exceed 2048 characters.

Text shall not contain the sequence `0<CR>` or

`OK<CR>` **Implementation**

2.3 Request Manufacturer Revision (+CGMR)

This command returns the manufacturer revision. The text is human readable and is not intended for microcontroller parsing.

By default this will return the firmware revision – release and build.

Command	Response	Example
+CGMR	<revision> +CME ERROR:<err>	AT+CGMR SSB,UNKNOWN-SHA1000135005 SECURITY_A,V150R100C10B180 PROTOCOL_A,V150R100C10B180 APPLICATION_A,V150R100C10B180 SECURITY_B,V150R100C10B180 OK
+CGMR=?		AT+CGMR=? OK

Description

Execution command returns one or more lines of information text <revision>. Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<revision>: the total number of characters, including line terminators, in the information text shall not exceed 2048 characters.

Text shall not contain the sequence 0<CR> or

OK<CR> **Implementation**

- <revision> will change format over time. It should be treated as an opaque identifier.

2.4 Request Product Serial Number (+CGSN)

Command	Response	Example
+CGSN[=<snt>]	when <snt>=0 (or omitted) and command successful: <sn> when <snt>=1 and command successful: +CGSN:<imei> when <snt>=2 and command successful: +CGSN:<imeisv> when <snt>=3 and command successful: +CGSN:<svn> *+CME ERROR:<err>* when TE supports <snt> and command successful: +CGSN:(list of supported <snt>s)	AT+CGSN=0 123E4567-E89B-12D3-A456-426655440000 OK AT+CGSN=1 +CGSN:49015420323751 OK
+CGSN=?		AT+CGSN=? +CGSN:(0,1,2,3) OK

Description

Execution command returns the IMEI (International Mobile station Equipment Identity number) and related information

For a TA which does not support <snt>, only OK is returned. Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<snt> integer type indicating the serial number type that has been requested.

0 returns <sn>

1 returns the IMEI (International Mobile station Equipment Identity)

2 returns the IMEISV (International Mobile station Equipment Identity and Software Version number)

3 returns the SVN (Software Version Number)

<sn> The 128-bit UUID of the UE. The total number of characters, including line terminators, in the information text shall not exceed 2048 characters. Text shall not contain the sequence 0<CR> or OK<CR>

<imei> string type in decimal format indicating the IMEI

<imeisv> string type in decimal format indicating the IMEISV

<svn> string type in decimal format indicating the current SVN which is a part of IMEISV;

Implementation

2.5 EPS network registration status (+CEREG)

Command	Response	Example
+CEREG=<n>	*+CME ERROR:<err>*	AT+CEREG=1 OK
+CEREG?	when <n>=0, 1, 2 or 3 and command successful: +CEREG:<n>,<stat>[,[<tac>],[<ci>],[<AcT>[,<cause_type>,<reject_cause>]]]	AT+CEREG? +CEREG:1,1 OK
+CEREG=?	when <n>=4 or 5 and command successful: +CEREG:<n>,<stat>[,[<lac>],[<ci>],[<AcT>],[<rac>][,[<cause type>],[<reject_cause>][,[<Active-Time>],[<Periodic-TAU>]]]] +CEREG:(list of supported <n>s)	AT+CEREG=? +CEREG:(0,1,2,3,4,5) OK

Description

The set command controls the presentation of an unsolicited result code +CEREG:<stat> when <n>=1 and there is a change in the MT's EPS network registration status in E-UTRAN, or unsolicited result code +CEREG:<stat>[,[<tac>],[<ci>],[<AcT>]] when <n>=2 and there is a change of the network cell in E-UTRAN. The parameters <AcT>, <tac> and <ci> are provided only if available. The value <n>=3 further extends the unsolicited result code with [,<cause_type>,<reject_cause>], when available, when the value of <stat> changes.

If the UE wants to apply PSM for reducing its power consumption the set command controls the presentation of an unsolicited result code +CEREG:<stat>[,[<tac>],[<ci>],[<AcT>][,[<cause_type>],[<reject_cause>][,[<Active-Time>],[<Periodic-TAU>]]]]. When <n>=4 the unsolicited result code will provide the UE with additional information for the Active Time value and the extended periodic TAU value if there is a change of the network cell in E-UTRAN. The value <n>=5 further enhances the unsolicited result code with <cause_type> and <reject_cause> when the value of <stat> changes. The parameters <AcT>, <tac>, <ci>, <cause_type>, <reject_cause>, <Active-Time> and <Periodic-TAU> are provided only if available.

Refer to Chapter 6: *Error Values* for possible <err> values.

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

The read command returns the status of result code presentation and an integer <stat> which shows whether the network has currently indicated the registration of the MT. Location information elements <tac>, <ci> and <AcT>, if available, are returned only when <n>=2 and MT is registered in the network. The parameters [,<cause_type>,<reject_cause>], if available, are returned when <n>=3.

Test command returns supported parameter values.

Defined values

<n>: integer type

0 disable network registration unsolicited result code

1 enable network registration unsolicited result code +CEREG:<stat>

2 enable network registration and location information unsolicited result code +CEREG:<stat>[,[<tac>],[<ci>],[<AcT>]]

3 enable network registration, location information and EMM cause value information unsolicited result code
+CEREG:<stat>[,[<tac>],[<ci>],[<AcT>][,<cause_type>,<reject_cause>]]

4 For a UE that wants to apply PSM, enable network registration and location information Unsolicited result code
+CEREG:<stat>[,[<tac>],[<ci>],[<AcT>][,,[,[<Active-Time>],[<Periodic-TAU>]]]]

5 For a UE that wants to apply PSM, enable network registration, location information and EMM Cause value information unsolicited result code
+CEREG:<stat>[,[<tac>],[<ci>],[<AcT>][,[<cause_type>],[<reject_cause>][,[<Active-Time>],[<Periodic-TAU>]]]]

<stat>: integer type; indicates the EPS registration status

0 not registered, MT is not currently searching for an operator to register to

1 registered, home network

2 not registered, but MT is currently trying to attach or searching for an operator to register to

3 registration denied

4 unknown (e.g. out of E-UTRAN coverage)

5 registered, roaming

6 registered for "SMS only", home network (not applicable)

7 registered for "SMS only", roaming (not applicable)

8 attached for emergency bearer services only

9 registered for "CSFB not preferred", home network (not applicable)

10 registered for "CSFB not preferred", roaming (not applicable)

- NOTE 2: 3GPP TS 24.008 and 3GPP TS 24.301 specify the condition when the MS is considered as attached for emergency bearer services.

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

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

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

0 GSM (not applicable)

1 GSM Compact (not applicable)

2 UTRAN (not applicable)

3 GSM w/EGPRS (not applicable)

4	UTRAN w/HSDPA (not applicable)
5	UTRAN w/HSUPA (not applicable)
6	UTRAN w/HSDPA and HSUPA (not applicable)
7	E-UTRAN
8	EC-GSM-IoT (A/Gb mode) (see NOTE 5) (not applicable)
9	E-UTRAN (NB-S1 mode) (see NOTE 6)

- NOTE 3: 3GPP TS 44.060 specifies the System Information messages which give the information about whether the serving cell supports EGPRS.
- NOTE 4: 3GPP TS 25.331 specifies the System Information blocks which give the information about whether the serving cell supports HSDPA or HSUPA.
- NOTE 5: 3GPP TS 44.018 [156] specifies the EC-SCH INFORMATION message which, if present, indicates that the serving cell supports EC-GSM-IoT.
- NOTE 6: 3GPP TS 36.331 [86] specifies the System Information blocks which give the information about whether the serving cell supports NB-IoT, which corresponds to E-UTRAN (NB-S1 mode).

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

0	Indicates that <reject_cause> contains an EMM cause value
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

Implementation

- <stats> return values supported are 0-5
- Unsolicited result code when <stat> changes.

2.6 Signaling connection status (+CSCON)

This command gives details of the terminal's perceived radio connection status (i.e. to the base-station). It returns an indication of the current state. Note, however, that this state is only updated when radio events, such as send and receive, take place. This means that the current state may be out of date. The terminal may think it is "Connected" yet cannot currently use a base station due to a change in the link quality.

The set command controls the presentation of an unsolicited result code +CSCON. If <n>=1, +CSCON: <mode> is sent from the MT when the connection mode of the MT is changed

Command	Response	Example
+CSCON=<n>	*+CME ERROR:<err>*	AT+CSCON=0 OK
+CSCON?	+CSCON:<n>,<mode>[,<state>] *+CME ERROR:<err>*	AT+CSCON? +CSCON:0,1 OK
+CSCON=?	+CSCON:(list of supported <n>s)	AT+CSCON=? +CSCON:(0,1) OK
	+CSCON:<mode>,[state]	+CSCON:1

Description

The set command controls the presentation of an unsolicited result code +CSCON.

If <n>=1, +CSCON:<mode> is sent from the MT when the connection mode of the MT is changed. If <n>=2 and there is a state within the current mode, +CSCON:<mode>[,<state>] is sent from the MT. If <n>=3,

+CSCON:<mode>[,<state>[,<access>]] is sent from the MT. If setting fails, an MT error, +CME ERROR:<err> is returned.
Refer to Chapter 6: *Error Values* for possible <err> values.

When the MT is in UTRAN or E-UTRAN, the mode of the MT refers to idle when no PS signaling connection and to connected mode when a PS signaling connection between UE and network is setup. When the UE is in GERAN, the mode refers to idle when the MT is in either the IDLE state or the STANDBY state and to connected mode when the MT is in READY state.
The <state> value indicates the state of the MT when the MT is in GERAN, UTRAN connected mode or E-UTRAN.

The read command returns the status of result code presentation and an integer <mode> which shows whether the MT is currently in idle mode or connected mode. State information <state> is returned only when <n>=2. Radio access type information <access> is returned only when <n>=3.

Test command returns supported values as a compound value.

Defined values

<n>: integer type

0	disable unsolicited result code
1	enable unsolicited result code +CSCON:<mode>
2	enable unsolicited result code +CSCON:<mode>[,<state>]
3	enable unsolicited result code +CSCON:<mode>[,<state>[,<access>]]

<mode>: integer type; indicates the signaling connection status

0	idle
1	connected
2-255	<reserved for future use>

<state>: integer type; indicates the CS or PS state while in GERAN and the RRC state information if the MT is in connected Mode while in UTRAN and E-UTRAN.

0	UTRAN URA_PCH state
1	UTRAN Cell_PCH state
2	UTRAN Cell_FACH state
3	UTRAN Cell_DCH state
4	GERAN CS connected state
5	GERAN PS connected state
6	GERAN CS and PS connected state
7	E-UTRAN connected state

<access>: integer type; indicates the current radio access type.

0	Indicates usage of radio access of type GERAN
1	Indicates usage of radio access of type UTRAN TDD
2	Indicates usage of radio access of type UTRAN FDD
3	Indicates usage of radio access of type E-UTRAN TDD
4	Indicates usage of radio access of type E-UTRAN FDD

Implementation

- Only <n>=0 and <n>=1 are supported. <n>=0 is the default value.

2.7 List Available Commands (+CLAC)

This command lists the available AT commands.

Command	Response	Example
+CLAC	<AT Command> [<CR><LF><AT Command> [...]] *+CME ERROR:<err>*	AT+CLAC AT+GMEE AT+CGMI ...

		AT+CGSN AT+CLAC OK
+CLAC=?	*+CME ERROR:<err>*	AT+CLAC=? OK

Description

Execution command causes the MT to return one or more lines of AT Commands. Refer to Chapter 6: *Error Values* for possible <err> values

NOTE: This command only returns the AT commands that are available for the user.

Defined values

<AT Command>: Defines the AT command including the prefix AT. Text shall not contain the sequence 0<CR> or OK<CR>

Implementation

2.8 Get signal strength indicator (+CSQ)

The terminal will provide a current signal strength indicator of 0 to 255 where larger is generally better. This information is based on a single measurement so can be expected to change greatly over short periods of time and may never use all possible (or even the majority) of the entire possible range or codes.

Command	Response	Example
+CSQ	+CSQ:<rssi>,<ber> *+CME ERROR:<err>*	AT+CSQ +CSQ:4,99 OK
+CSQ=?	+CSQ:(list of supported <rssi>s),(list of supported <ber>s)	AT+CSQ=? +CSQ:(0-31,99),(99) OK

Description

Execution command returns received signal strength indication <rssi> and channel bit error rate <ber> from the MT. Refer to Chapter 6: *Error Values* for possible <err> values

Test command returns values supported as compound values.

Defined values

<rssi>: integer type

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

<ber>: integer type; channel bit error rate (in percent)

0...7	as RXQUAL values (refer to 3GPP specification)
99	not known or not detectable

Implementation

- <ber> is currently not implemented, and will always be 99.

2.9 Show PDP Addresses (+CGPADDR)

This command returns the IP address of the device.

Command	Response	Example
+CGPADDR[=<cid>[,<cid>[,...]]]	[+CGPADDR:<cid>[,<PDP addr 1>[,<PDP_addr_2>]]] [<CR><LF>+CGPADDR:<cid>,[<PDP_addr_1>[,<PDP_addr_2>]] [...]]	AT+CGPADDR +CGPADDR:1,101.43.5.1 +CGPADDR:2,2001:db8:85a3::8a2e:370 +CGPADDR:3
	IPv4: The string is given as	AT+CGPADDR=1

	dot-separated numeric (0-255) parameter of the form: a1.a2.a3.a4 IPv6: The string is given as colon-separated hexadecimal parameter.	+CGPADDR:1,101.43.5.1 OK
+CGPADDR=?	+CGPADDR:(list of defined <cid>s)	AT+CGPADDR=? +CGPADDR:(0,1) OK

Description

The execution command returns a list of PDP addresses for the specified context identifiers.
If no <cid> is specified, the addresses for all defined contexts are returned.
Refer to Chapter 6: *Error Values* for possible <err> values

The test command returns a list of defined <cid>s. These are <cid>s that have been activated and may or may not have an IP address associated with them.

Defined values

- <cid>: integer type; specifies a particular PDP context definition (see the +CGDCONT and +CGDSCONT commands).
- <PDP_addr_1> and <PDP_addr_2>: each is a string type 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 and +CGDSCONT commands 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>. Both <PDP_addr_1> and <PDP_addr_2>are omitted if none is available. Both <PDP_addr_1> and <PDP_addr_2> are included when both IPv4 and IPv6 addresses are assigned, with <PDP_addr_1> containing the IPv4 address and <PDP_addr_2> containing the IPv6 address.
- The string is given as dot-separated numeric (0-255) parameter of the form:
  a1.a2.a3.a4 for IPv4 and
  a1.a2.a3.a4.a5.a6.a7.a8.a9.a10.a11.a12.a13.a14.a15.a16 for IPv6.
- When +CGPIAF is supported, its settings can influence the format of the IPv6 address in parameter <PDP_addr_1> or <PDP_addr_2> returned with the execute form of +CGPADDR.

NOTE: In dual-stack terminals (<PDP_type> IPV4V6), the IPv6 address will be provided in <PDP_addr_2>. For terminals with a single IPv6 stack (<PDP_type> IPv6) or due to backwards compatibility, the IPv6 address can be provided in parameter <PDP_addr_1>.

Implementation

- <cid> values between 0 & 10 are supported.

2.10 PLMN selection (+COPS)

Command	Response	Example
+COPS=<mode>[,<format>[,<oper>[,<AcT>]]]	*+CME ERROR:<err>*	AT+COPS=1,2,"320160" OK
+COPS?	+COPS:<mode>[,<format>,<oper>][,<AcT>] *+CME ERROR:<err>*	AT+COPS? +COPS:1,2,"320160" OK
+COPS=?	+COPS: [list of supported (<stat>, long alphanumeric <oper>, alphanumeric <oper>, numeric <oper>[,<Act>])s][,,(list supported <mode>s),	AT+COPS=? +COPS:(2,,,"24405"),(0,,,"24491"),,(0-2),(2) OK

(list of supported <format>s)]

Description

Set command forces an attempt to select and register the GSM/UMTS/EPS network operator using the SIM/USIM card installed in the currently selected card slot. <mode> is used to select whether the selection is done automatically by the MT or is forced by this command to operator <oper> (it shall be given in format <format>) to a certain access technology, indicated in <AcT>. If the selected operator is not available, no other operator shall be selected (except <mode>=4). If the selected access technology is not available, then the same operator shall be selected in other access technology. The selected operator name format shall apply to further read commands (+COPS?) also. <mode>=2 forces an attempt to deregister from the network. The selected mode affects to all further network registration (e.g. after <mode>=2, MT shall be unregistered until <mode>=0 or 1 is selected). This command should be abortable when registration/deregistration attempt is made. Refer to Chapter 6: *Error Values* for possible <err> values

Read command returns the current mode, the currently selected operator and the current Access Technology. If no operator is selected, <format>, <oper> and <AcT> are omitted.

Test command returns a set of five parameters, each representing an operator present in the network. A set consists of an integer indicating the availability of the operator <stat>, long and short alphanumeric format of the name of the operator, numeric format representation of the operator and access technology. Any of the formats may be unavailable and should then be an empty field. The list of operators shall be in order: home network, networks referenced in SIM or active application in the UICC (GSM or USIM) in the following order: HPLMN selector, User controlled PLMN selector, Operator controlled PLMN selector and PLMN selector (in the SIM or GSM application), and other networks.

It is recommended (although optional) that after the operator list TA returns lists of supported <mode>s and <format>s. These lists shall be delimited from the operator list by two commas.

The access technology selected parameters, <AcT>, should only be used in terminals capable to register to more than one access technology. Selection of <AcT> does not limit the capability to cell reselections, even though an attempt is made to select an access technology, the UE may still re-select a cell in another access technology.

Defined values

<mode>: integer type

0 automatic (<oper> field is ignored)

1 manual (<oper> field shall be present, and <AcT> optionally)

2 deregister from network

3 set only <format> (for read command +COPS?), do not attempt registration/deregistration (<oper> and <AcT> fields are ignored); this value is not applicable in read command response

4 manual/automatic (<oper> field shall be present); if manual selection fails, automatic mode (<mode>=0) is entered

<format>: integer type

0 long format alphanumeric <oper>

1 short format alphanumeric <oper>

2 numeric <oper>

<oper>: string type; <format> indicates if the format is alphanumeric or numeric; long alphanumeric format can be upto 16 characters long and short format up to 8 characters; numeric format is the GSM Location Area Identification number which consists of a three BCD digit ITU-T country code coded, plus a two or three BCD digit network code, which is administration specific.

<stat>: integer type

0 unknown

1 available

2 current

3 forbidden

<AcT>: integer type; access technology selected

0 GSM

1 GSM Compact

2 UTRAN

3 GSM w/EGPRS (see NOTE 1)

4 UTRAN w/HSDPA (see NOTE 2)

5 UTRAN w/HSUPA (see NOTE 2)

6 UTRAN w/HSDPA and HSUPA (see NOTE 2)

7 E-UTRAN
8 EC-GSM-IoT (A/Gb mode) (see NOTE 3)
9 E-UTRAN (NB-S1 mode) (see NOTE 4)

- NOTE 1: 3GPP TS 44.060 specifies the System Information messages which give the information about whether the serving cell supports EGPRS.
- NOTE 2: 3GPP TS 25.331 specifies the System Information blocks which give the information about whether the serving cell supports HSDPA or HSUPA.
- NOTE 3: 3GPP TS 44.018 [156] specifies the EC-SCH INFORMATION message which, if present, indicates that the serving cell supports EC-GSM-IoT.
- NOTE 4: 3GPP TS 36.331 [86] specifies the System Information blocks which give the information about whether the serving cell supports NB-IoT, which corresponds to E-UTRAN (NB-S1 mode).

Implementation

- <Act>, if provided, must be set to 9
- Only <format>=2 is supported
- Only <mode>=0, <mode>=1 & <mode>=2 are supported
- When <mode>=1 is used, the plmn setting will not persist after the UE is rebooted
- <mode>=1 is only for development use. <mode>=0 should be used in production, which the mode used when AUTOCONNECT is enabled.
- The test command currently returns the configured values rather than performing a plmn search.
- Not return <ACT> for AT+COPS?
- <oper> field couldn't be present when <mode>=0;
- +COPS=? will start background plmn search when MT is idle state and wait searching result, other state will return ERROR.
  Return the searched plmn list in the network after searching.
  And could be aborted by any characters when searching. See Chapter 8 for more details about abort.

2.11 PS attach or detach (+CGATT)

Command	Response	Example
+CGATT=<state>	*+CME ERROR:<err>*	AT+CGATT=1 OK
+CGATT?	+CGATT:<state> *+CME ERROR:<err>*	AT+CGATT? +CGATT:0 OK
+CGATT=?	+CGATT:(list of supported <state>s) *+CME ERROR:<err>*	AT+CGATT=? +CGATT:(0,1) OK

Description

The execution command is used to attach the MT to, or detach the MT from, the Packet Domain service. After the command has completed, the MT remains in V.250 command state. If the MT is already in the requested state, the command is ignored and the OK response is returned. If the requested state cannot be achieved, an ERROR or +CME ERROR response is returned.
Refer to Chapter 6: *Error Values* for possible <err> values.

- NOTE 1: If the initial PDP context is supported, the context with <cid>=0 is automatically defined at startup.

Any active PDP contexts will be automatically deactivated when the attachment state changes to detached.
The read command returns the current Packet Domain service state.
The test command is used for requesting information on the supported Packet Domain service states.

- NOTE 2: This command has the characteristics of both the V.250 action and parameter commands. Hence it has the read form in addition to the execution/set and test forms.

Defined Values

<state> integer type; indicates the state of PDP context activation. The default value is manufacturer specific.
0 detached
1 attached
<err> error value

Implementation

- When <state>=1 is selected, +COPS=0 is automatically selected.
- If a CGATT is in progress, further execution of the CGATT= command prior to the attach or detach completing will return error.

2.12 PDP context activate or deactivate (+CGACT)

Command	Response	Example
+CGACT=[<state>[,<cid>[,<cid>[,...]]]]	*+CME ERROR:<err>*	AT+CGACT=0,1 OK
+CGACT?	[+CGACT:<cid>,<state>] [<CR><LF>+CGACT:<cid>,<state> [...]] *+CME ERROR:<err>*	AT+CGACT? +CGACT:1,0 OK
+CGACT=?	+CGACT:(list of supported <state>s) *+CME ERROR:<err>*	AT+CGACT=? +CGACT:(0,1) OK

Description

The execution command is used to activate or deactivate the specified PDP context (s). After the command has completed, the MT 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 requested state for any specified context cannot be achieved, an ERROR or +CME ERROR response is returned. Extended error responses are enabled by the +CMEE command. If the MT is not PS attached when the activation form of the command is executed, the MT first performs a PS attach and then attempts to activate the specified contexts. If the attach fails then the MT responds with ERROR or, if extended error responses are enabled, with the appropriate failure-to-attach error message. Refer subclause 9.2 for possible <err> values.

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

NOTE: If the initial PDP context is supported, the context with <cid>=0 is automatically defined at startup, see subclause 10.1.0.

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 MT 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 non-emergency 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.

NOTE. This command has the characteristics of both the V.250 action and parameter commands. Hence it has the read form in addition to the execution/set and test forms.

Defined Values

<state>: integer type; indicates the state of PDP context activation. The default value is manufacturer specific.

0 deactivated

1 activated

<cid>: integer type; specifies a particular PDP context definition (see the +CGDCONT command).

Implementation

- Could only activate or deactivate one cid one time.
- Could not be executed when plmn searching, attaching or detaching.
- Could disconnect the last PDN connection when MT and Core Network all support without PDN connection.

2.13 Request international mobile subscriber identity (+CIMI)

Command	Response	Example

+CIMI	<IMSI> *+CME ERROR:<err>*	AT+CIMI 460001357924680 OK
+CIMI=?		AT+CIMI=? OK

Description

Execution command causes the TA to return <IMSI>, which is intended to permit the TE to identify the individual SIM card or active application in the UICC (GSM or USIM) which is attached to MT. Refer to Chapter 6: *Error Values* for possible <err> values

Defined values

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

Implementation

- IMSI may not be displayed for a few seconds after power-on

2.14 Define PDP Context (+CGDCONT)

Command	Response	Example
+CGDCONT=<cid>[,<PDP_type>[,<APN>[,<PDP_addr>[,<d_comp>[,<h_comp>[,<IPv4AddrAlloc>[,<request_type>[,<P-CSCF_discovery>[,<IM_CN_Signalling_Flag_Ind>[,<NSLPI>[,<securePCO>[,<IPv4_MTU_discovery>][,<Local_Addr_Ind>]]]]]]]]]]]		AT+CGDCONT=1,”IP”,”HUAWEI.COM” OK
+CGDCONT?	[+CGDCONT:<cid>,<PDP_type>,<APN>,<PDP_addr>,<d_comp>,<h_comp>[,<IPv4AddrAlloc>[,<request_type>[,<P-CSCF_discovery>[,<IM_CN_Signalling_Flag_Ind>[,<NSLPI>[,<securePCO>[,<IPv4_MTU_discovery>]]]]]]]] [<CR><LF>+CGDCONT: <cid>,<PDP_type>,<APN>,<PDP_addr>,<d_comp>,<h_comp>[,<IPv4AddrAlloc>[,<request_type>[,<P-CSCF_discovery>[,<IM_CN_Signalling_Flag_Ind>[,<NSLPI>[,<securePCO>[,<IPv4_MTU_discovery>]]]]]]] [...]]	AT+CGDCONT? +CGDCONT:1,”IP”,”HUAWEI.COM”,,0,0,,,,,0 OK
+CGDCONT=?	+CGDCONT:(range of supported <cid>s),<PDP type>,,,(list of supported <d_comp>s),(list of supported <h_comp>s),(list of supported <IPv4AddrAlloc>s),(list of supported <request_type>s),(list of supported <P-CSCF_discovery>s),(list of supported <IM_CN_Signalling_Flag_Ind>s) ,(list of supported <NSLPI>s),(list of supported <securePCO>s),(list of supported <IPv4_MTU_discovery>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 <IPv4AddrAlloc>s),(list of supported <request_type>s),(list of supported <P-CSCF_discovery>s),(list of supported <IM_CN_Signalling_Flag_Ind>s) ,(list of supported <NSLPI>s),(list of supported <securePCO>s,(list of supported <IPv4_MTU_discovery>s)) [...]]	AT+CGDCONT=? +CGDCONT:(0-10),(\"IP\",\"NONIP\",\"IPV6\",\"IPV4V6\"),,,(0),(0),,,,,(0,1) OK

Description

The set command specifies PDP context parameter values for a PDP context identified by the (local) context identification parameter, <cid> and also allows the TE to specify whether security protected transmission of

ESM information is requested, because the PCO can include information that requires ciphering. There can be other reasons for the UE to use security protected transmission of ESM information, e.g. if the UE needs to transfer an APN. The number of PDP contexts that may be in a defined state at the same time is given by the range returned by the test command. Refer to Chapter 6: *Error Values* for possible <err> values.

For EPS the PDN connection and its associated EPS default bearer is identified herewith.

A special form of the set command, +CGDCONT=<cid> causes the values for context number <cid> to become undefined.

If the initial PDP context is supported, the context with <cid>=0 is automatically defined at startup, see 3GPP TS 27.007 V13.5.0, subclause 10.1.0. As all other contexts, the parameters for <cid>=0 can be modified with +CGDCONT. If the initial PDP context is supported, +CGDCONT=0 resets context number 0 to its particular default settings.

The read command returns the current settings for each defined context.

The test command returns values supported as compound values. If the MT supports several PDP types, <PDP_type>, the parameter value ranges for each <PDP_type> are returned on a separate line.

Defined values

<cid>: integer type; specifies a particular PDP context definition. The parameter is local to the TE-MT interface and is used in other PDP context-related commands. The range of permitted values (minimum value = 1 or if the initial PDP context is supported, minimum value = 0) is returned by the test form of the command.

- NOTE 1: The <cid>s for network-initiated PDP contexts will have values outside the ranges indicated for the <cid> in the test form of the commands +CGDCONT and +CGDSCONT.

<PDP_type>: string type; specifies the type of packet data protocol. The default value is manufacturer specific.

X.25	ITU-T/CCITT X.25 layer 3 (Obsolete)
IP	Internet Protocol (IETF STD 5 [103])
IPV6	Internet Protocol, version 6
IPV4V6	Virtual <PDP_type> introduced to handle dual IP stack UE capability.
OSPIH	Internet Hosted Octect Stream Protocol (Obsolete)
PPP	Point to Point Protocol (IETF STD 51 [104])

- NOTE 2: Only IP, IPV6 and IPV4V6 values are supported for EPS services.

NONIP None Ip

<APN>: string type; 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. The APN is a string of up to 82 characters.

<PDP_addr>: string type; identifies the MT in the address space applicable to the PDP. This parameter can be left blank.

- When +CGPIAF is supported, its settings can influence the format of this parameter returned with the read form of +CGDCONT.NOTE 3: The value of this parameter is ignored with the set command. The parameter is included in the set command for backwards compatibility reasons only.

<d_comp>: integer type; controls PDP data compression

0	off
1	on (manufacturer preferred compression)
2	V.42bis
3	V.44

<h_comp>: integer type; controls PDP header compression

0	off
1	on (manufacturer preferred compression)
2	RFC 1144 [105] (applicable for SNDCP only)
3	RFC 2507 [107]
4	RFC 3095 [108] (applicable for PDCP only)

<IPv4AddrAlloc>: integer type; controls how the MT/TA requests to get the IPv4 address information

0 IPv4 address allocation through NAS signaling

1 IPv4 address allocated through DHCP

<request_type>:integer type; indicates the type of PDP context activation request for the PDP context. , see 3GPP TS 24.301 (subclause 6.5.1.2) and 3GPP TS 24.008 subclause 10.5.6.17). If the initial PDP context is supported it is not allowed to assign <cid>=0 for emergency bearer services. According to 3GPP TS 24.008 (subclause 4.2.4.2.2 and subclause 4.2.5.1.4) and 3GPP TS 24.301 (subclause 5.2.2.3.3 and subclause 5.2.3.2.2), a separate PDP context must be established for emergency bearer services.

- NOTE 4: If the PDP context for emergency bearer services is the only activated context, only emergency calls are allowed, see 3GPP TS 23.401 subclause 4.3.12.9.

0 PDP context is for new PDP context establishment or for handover from a non-3GPP access network (how the MT decides whether the PDP context is for new PDP context establishment or for handover is implementation specific)

1 PDP context is for emergency bearer services

2 PDP context is for new PDP context establishment

3 PDP context is for handover from a non-3GPP access network

<P-CSCF_discovery>: integer type; influences how the MT/TA requests to get the P-CSCF address, see 3GPP TS 24.229 [89] annex B and annex L.

0 Preference of P-CSCF address discovery not influenced by +CGDCONT

1 Preference of P-CSCF address discovery through NAS signalling

2 Preference of P-CSCF address discovery through DHCP

<IM_CN_Signalling_Flag_Ind>: integer type; indicates to the network whether the PDP context is for IM CN subsystem-related signalling only or not.

0 UE indicates that the PDP context is not for IM CN subsystem-related signalling only

1 UE indicates that the PDP context is for IM CN subsystem-related signalling only

<NSLPI>: integer type; indicates the NAS signalling priority requested for this PDP context:

0 indicates that this PDP context is to be activated with the value for the low priority indicator configured in the MT.

1 indicates that this PDP context is to be activated with the value for the low priority indicator set to "MS is not configured for NAS signalling low priority".

- NOTE 5: The MT utilises the provide NSLPI information as specified in 3GPP TS 24.301 [83] and 3GPP TS 24.008 .

<securePCO>: integer type. Specifies if security protected transmission of PCO is requested or not (applicable for EPS only)

0 Security protected transmission of PCO is not requested

1 Security protected transmission of PCO is requested

<IPv4_MTU_discovery>: integer type; influences how the MT/TA requests to get the IPv4 MTU size, see 3GPP TS 24.008 subclause 10.5.6.3.

0 Preference of IPv4 MTU size discovery not influenced by +CGDCONT

1 Preference of IPv4 MTU size discovery through NAS signalling

Implementation

- Only <PDP_type>=“IP” ,”NONIP”,” IPV6”, “IPV4V6”are supported.
- supports +CGDCONT=<cid>,<PDP_type>,<APN>,,,,,,<NSLPI>only.
- <cid> values of 0-10 are supported
- Only <hcomp> and <dcomp> values of 0 are supported.
- No default <PDP_type > value specific.
- <cid> value of 7 can’t be set when bip is enabled.
- Maximum <APN> string size is 63.

2.15 Change device functionality (+CFUN)

Command	Response	Example
+CFUN=<fun>[,<rst>]	*+CME ERROR:<err>*	AT+CFUN=1 OK
+CFUN?	+CFUN:<fun>	AT+CFUN? +CFUN:1 OK
+CFUN=?	+CFUN:(list of supported <fun>s),(list of supported <rst>s)	AT+CFUN=? +CFUN:(0,1),(0,1) OK

Description

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

Refer to Chapter 6: *Error Values* for possible <err> values.

NOTE 1: It is manufacturer specific if this command affects network registration. Command Operator Selection +COPS is used to force registration/deregistration. Read command returns the current setting of <fun>.

Test command returns values supported by the MT as compound values.

Defined values

<fun>: integer type

0 minimum functionality

1 full functionality. Enable (turn on) the transmit and receive RF circuits for all supported radio access technologies. For MTs supporting +CSRA, this equals the RATs indicated by the response of +CSRA=?. Current +CSRA setting is ignored. It is not required that the MT transmit and receive RF circuits are in a disabled state for this setting to have effect.

2 disable (turn off) MT transmit RF circuits only

3 disable (turn off) MT receive RF circuits only

4 disable (turn off) both MT transmit and receive RF circuits

5...127 reserved for manufacturers as intermediate states between full and minimum functionality

128 Full functionality with radio access support according to the setting of +CSRA. Enables (turns on) the transmit and receive RF circuits if not already enabled. This <fun> setting is applicable for MTs supporting +CSRA.

129 Prepare for shutdown. This setting has its prime use when some of the MT's resources (e.g. file system) are located on a tightly integrated TE (host). The MT will execute pending actions resulting in "permanent" changes, e.g. execute pending file system operations. The MT will also make an orderly network detach. After this action and +CFUN has returned OK, the MT can be shut down with <fun>=0, or by other means. After setting <fun>=129, only <fun>=0 is valid. All other values will make +CFUN return ERROR.

<rst>: integer type

0 do not reset the MT before setting it to <fun> power level

NOTE 2: This shall be always default when <rst> is not given.

1 reset the MT before setting it to <fun> power level

Implementation

- Only <fun> = 0 & 1 are supported.
- <rst> is not supported and will be ignored.
- DeepSleep will be entered when the system is quiescent, but only if it has been enabled by the network.
- Automatic register will not start after power on for AT+CFUN=1.

2.16 Report mobile termination error (+CMEE)

Command	Response	Example
+CMEE=<n>		AT+CMEE=1 OK
+CMEE?	+CMEE:<n>	AT+CMEE?

		+CMEE:1 OK
+CMEE=?	+CMEE:(list of supported <n>s)	AT+CMEE=? +CMEE:(0,1) OK

Description

Set command disables or enables the use of final 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 setting of <n>.

Test command returns values supported as a compound value.

Refer to Chapter 6: *Error Values* for possible <err> values

Defined values

<n>: integer type

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

1 enable +CME ERROR:<err> result code and use numeric <err> values (Refer to Chapter *6: Error Values* for possible <err> values

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

Implementation

- Only n=0 & n=1 are supported.

2.17 Return current date & time (+CCLK)

Command	Response	Example
+CCLK?	*+CME ERROR:<err>*	AT+NITZ=0 OK AT+CCLK=”17/01/09,14:53:12+01” OK
+CCLK?	*+CCLK:[<yy/MM/dd,hh:mm:ss>[<±zz>]]* *+CME ERROR:<err>*	AT+CCLK=? OK
+CCLK=?		

Description

The clock will be set automatically once the UE has connected to the network.

Read command returns the current setting of the clock.

Refer to Chapter 6: *Error Values* for possible <err> values

Defined values

<time>: string type value; format is "yy/MM/dd,hh:mm:ss±zz", where characters indicate year (two last digits), month, day, hour, minutes, seconds and time zone (indicates the difference, expressed in quarters of an hour, between the local time and GMT; range -96...+96). E.g. 6th of May 1994, 22:10:00 GMT+2 hours equals to "94/05/06,22:10:00+08"

NOTE: If MT does not support time zone information then the three last characters of <time> are not returned by +CCLK?

Implementation

- No value will be returned before core network sent EMM INFORMATION.
- Need to send AT+NITZ=0 first before set time.

2.18 Power saving mode setting (+CPSMS)

Command	Response	Example
+CPSMS=[<mode>[,<Requested_Periodic-RAU>[,<Requested_GPRS-READY-timer>[,<Requested_Periodic-TAU>[,<Requested_Active-	+CME ERROR:<err>	AT+CPSMS=1,,,01000011,01000011 OK

Time>]]]]]		
+CPSMS?	+CPSMS:<mode>,[<Requested_Periodic-RAU>],[<Requested_GPRS-READY-timer>],[<Requested_Periodic-TAU>],[<Requested_Active-Time>]	AT+CPSMS? +CPSMS:1,,,01000011,01000011 OK
+CPSMS=?	+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)	AT+CPSMS=? +CPSMS:(0,1,2),,,(00000000-11111111), (00000000-11111111) OK

Description

The set command controls the setting of the UEs power saving mode (PSM) parameters. The command controls whether the UE wants to apply PSM or not, as well as the requested extended periodic RAU value and the requested GPRS READY timer value in GERAN/UTRAN, the requested extended periodic TAU value in E-UTRAN and the requested Active Time value. See the unsolicited result codes provided by command +CEREG for the Active Time value and the extended periodic TAU value that are allocated to the UE by the network in E-UTRAN.

A special form of the command can be given as +CPSMS=2. In this form, the use of PSM will be disabled and data for all parameters in the command +CPSMS will be removed or, if available, set to the manufacturer specific default values.

The read command returns the current parameter values.

The test command returns the supported <mode>s and the value ranges for the requested extended periodic RAU value and the requested GPRS READY timer value in GERAN/UTRAN, the requested extended periodic TAU value in E-UTRAN and the requested Active Time value as compound values.

Refer to Chapter 6: *Error Values* for possible <err> values

Defined values

<mode>: integer type. Indication to disable or enable the use of PSM in the UE.

0 Disable the use of PSM

1 Enable the use of PSM

2 Disable the use of PSM and discard all parameters for PSM or, if available, reset

to the manufacturer specific default values.

<Requested_Periodic-RAU>: string type; one byte in an 8 bit format. Requested extended periodic RAU value (T3312) to be allocated to the UE in GERAN/UTRAN. The requested extended periodic RAU 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 [8] Table 10.5.163a/3GPP TS 24.008. See also 3GPP TS 23.682 [149] and 3GPP TS 23.060 [47]. The default value, if available, is manufacturer specific.

<Requested_GPRS-READY-timer>: string type; one byte in an 8 bit format. Requested GPRS READY timer value (T3314) to be allocated to the UE in GERAN/UTRAN. The requested GPRS READY timer value is coded as one byte (octet 2) of the GPRS Timer information element coded as bit format (e.g. "01000011" equals 3 decihours or 18 minutes). For the coding and the value range, see the GPRS Timer IE in 3GPP TS 24.008 [8] Table 10.5.172/3GPP TS 24.008. See also 3GPP TS 23.060 [47]. The default value, if available, is manufacturer specific.

<Requested_Periodic-TAU>: string type; one byte in an 8 bit format. Requested extended periodic TAU value (T3412) to be allocated to the UE in E-UTRAN. The requested 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 [8] Table 10.5.163a/3GPP TS 24.008. See also 3GPP TS 23.682 [149] and 3GPP TS 23.401 [82]. The default value, if available, is manufacturer specific.

<Requested_Active-Time>: string type; one byte in an 8 bit format. Requested Active Time value (T3324) to be allocated to the UE. The requested 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 [8] Table 10.5.163/3GPP TS 24.008. See also 3GPP TS 23.682 [149], 3GPP TS 23.060 [47] and 3GPP TS 23.401 [82]. The default value, if available, is manufacturer specific.

Implementation

- RAU is not supported by NB-IOT. No value will be output, and any input will be ignored.

- <Requested_Periodic-TAU> and <Requested_Active-Time> could be quoted.

2.19 eDRX setting (+CEDRXS)

Command	Response	Example
+CEDRXS=[<mode>,[,<AcT-type>[,<Requested_eDRX_value>]]]	*+CME ERROR:<err>*	AT+CEDRXS=1,5,0101 OK
+CEDRXS?	[+CEDRXS:<AcT-type>,<Requested_eDRX_value> [<CR><LF>+CEDRXS:<AcT-type>,<Requested_eDRX_value> [...]]]	AT+CEDRXS? +CEDRXS:5,"0101" OK
+CEDRXS=?	+CEDRXS:(list of supported <mode>s),(list of supported <AcT-type>s),(list of supported <Requested_eDRX_value>s)	AT+*CEDRXS*=? *+CEDRXS:(0,1,2,3),(5),("0000"-"1111")* OK

Description

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 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 +CEDRXS=3. In this form, eDRX will be disabled and data for all parameters in the command +CEDRXS will be removed or, if available, set to the manufacturer specific default values.

The read command returns the current settings for each defined value of <AcT-type>.

The test command returns the supported <mode>s and the value ranges for the access technology and the requested eDRX value as compound values.

Refer to Chapter 6: *Error Values* for possible <err> values

Defined values

<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.

1 EC-GSM-IoT (A/Gb mode)

2 GSM (A/Gb mode)

3 UTRAN (Iu mode)

4 E-UTRAN (WB-S1 mode)

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 subclause 10.5.5.32 of

3GPP TS 24.008 [8]). For the coding and the value range, see Extended DRX parameters information element in 3GPP TS 24.008 [8] Table 10.5.5.32/3GPP TS 24.008. The default value, if available, is manufacturer specific.

<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 subclause 10.5.5.32 of 3GPP TS 24.008 [8]). For the coding and the value range, see Extended DRX parameters information element in 3GPP TS 24.008 [8] 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 referes to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see subclause 10.5.5.32 of 3GPP TS 24.008 [8]). For the coding and the value range, see the Extended DRX parameters information element in 3GPP TS 24.008 [8] Table 10.5.5.32/3GPP TS 24.008.

Implementation

- <AcT-type> value 5 is supported.

2.20 Extended Error Reporting (+CEER)

Command	Response	Example
AT+CEER	+CEER:<report>	AT+CEER +CEER:ILLEGAL_ME OK
AT+CEER=?		AT+CEER=? OK

Description

Execution command causes the TA to return one or more lines of information text <report>, determined by the MT manufacturer, which should offer the user of the TA an extended report of the reason for

- the failure in the last unsuccessful call setup (originating or answering) or in-call modification;
- the last call release;
- the last unsuccessful PDP context activation;
- the last PDP context deactivation.

Typically, the text will consist of a single line containing the cause information given by GSM/UMTS network in textual format.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<report>: the total number of characters, including line terminators, in the information text shall not exceed 2041 characters.

Text shall not contain the sequence 0<CR> or OK<CR>.

Implementation

2.21 eDRX read dynamic parameters (+CEDRXRDP)

Command	Response	Example
AT+CEDRXRDP	+CEDRXRDP: <AcT-type>[,<Requested_eDRX_value>[,<NW-provided_eDRX_value>[,<Paging_time_window>]]]	AT+CEDRXRDP +CEDRXRDP:5,”0010”,”1110”,”0101” OK
AT+CEDRXRDP=?		AT+CEDRXRDP=? OK

Description

The execution command returns <AcT-type> and <Requested_eDRX_value>, <NW-provided_eDRX_value> and <Paging_time_window> if eDRX is used for the cell that the MS is currently registered to.

If the cell that the MS is currently registered to is not using eDRX, AcT-type=0 is returned.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined 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

1 EC-GSM-IoT (A/Gb mode)

2 GSM (A/Gb mode)

3 UTRAN (Iu mode)

4 E-UTRAN (WB-S1 mode)

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 subclause 10.5.5.32 of 3GPP TS 24.008 [8]). For the coding and the value range, see Extended DRX parameters information element in 3GPP TS 24.008 [8] 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 subclause 10.5.5.32 of 3GPP TS 24.008 [8]). For the coding and the value range, see Extended DRX parameters information element in 3GPP TS 24.008 [8] 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 referes to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see subclause 10.5.5.32 of 3GPP TS 24.008 [8]). For the coding and the value range, see the Extended DRX parameters information element in 3GPP TS 24.008 [8] Table 10.5.5.32/3GPP TS 24.008.

Implementation

- <AcT-type> value 0 and 5 is supported.

2.22 Time Zone Reporting (+CTZR)

Command	Response	Example
+CTZR[=<reporting>]	*+CME ERROR:<err>*	AT+CTZR=1 OK
+CTZR?	*+CTZR:<reporting>* *+CME ERROR:<err>*	AT+CTZR? +CTZR:1 OK
+CTZR=?	*+CZTR:(list of supported <reporting>s)* *+CME ERROR:<err>*	AT+CTZR=? +CTZR:(0,1,2,3) OK

Description

This set command controls the time zone change event reporting. If reporting is enabled the MT returns the unsolicited result code +CTZV: <tz>, +CTZE: <tz>,<dst>,[<time>], or +CTZEU: <tz>,<dst>,[<utime>] whenever the time zone is changed. The MT also provides the time zone upon network registration if provided by the network. If setting fails in an MT error, +CME ERROR: <err> is returned. Refer to Chapter 6: *Error Values* for possible <err> values.

Read command returns the current reporting settings in the MT.

Test command returns supported <reporting>-values as a compound value.

NOTE: The Time Zone reporting is not affected by the Automatic Time Zone setting command, +CTZU if implemented.

Command	Response	Example
+CTZV:	<tz>	+CTZV:4
+CTZE:	<tz>,<dst>,[<time>]	+CTZE:4,0
+CTZEU:	<tz>,<dst>,[<utime>]	+CTZEU:4,0

Defined values

<reporting>: integer type value indicating:

0 disable time zone change event reporting.

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>]`.

3 Enable extended time zone and universal time reporting by unsolicited result code `+CTZEU: <tz>,<dst>,[<utime>]`.

`<tz>`: string type value representing the sum of the local time zone (difference between the local time and GMT 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 value indicating 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 value representing 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). The local time can be derived by the MT from information provided by the network at the time of delivering time zone information and will be present in the unsolicited result code for extended time zone and local time reporting if the universal time is provided by the network.

`<utime>`: string type value representing the universal 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). The universal time can be provided by the network at the time of delivering time zone information and will be present in the unsolicited result code for extended time zone and universal time reporting if provided by the network.

Implementation

2.23 Initial PDP context activation (+CIPCA)

Command	Response	Example
+CIPCA=[<n>[,<AttachWithoutPDN>]]	+CME ERROR: <err>	AT+CIPCA=3,1 OK
+CIPCA?	+CIPCA:<n>[,<AttachWithoutPDN>]	AT+CIPCA? +CIPCA:3,1 OK
+ CIPCA =?	+CIPCA:(list of supported <n>s),(list of supported <AttachWithoutPDN>s)	AT+CIPCA=? +CIPCA:(3),(0,1) OK

Description

The set command controls whether an initial PDP context (see subclause 10.1.0) shall be established automatically following an attach procedure when the UE is attached to GERAN or UTRAN RATs and whether the UE is attached to E-UTRAN with or without a PDN connection.

For `<n>≠0`, deactivating the last (active) PDP context can lead to a (re)establishment of the initial PDP context. Changing setting of `<n>` from 0 to 1 will cause an immediate attempt to (re)establish the initial PDP context if no PDP context is active. Changing `<n>` from 0 to 2 will if not roaming cause an immediate attempt to (re)establish the initial PDP context if no other PDP context is active. The value of `<n>=3` applies to E-UTRAN RATs and does not change the setting of PDP context activation in GERAN or UTRAN RATs. Changing `<n>` will never cause a PDP context deactivation.

For `<AttachWithoutPDN>=1`, the EPS Attach is performed without a PDN connection.

NOTE: For this command, the term roaming corresponds to being registered to a VPLMN which is not equivalent to HPLMN or EHPLMN.

The read command returns the current setting of the command.

The test command returns values supported as a compound value.

Refer to Chapter 6: *Error Values* for possible <err> values

Defined values

<n>: integer type. Activation of PDP context upon attach.

0 Do not activate

1 Always activate

2 Activate when not roaming

3 No change in current setting

<AttachWithoutPDN>: integer type. EPS Attach with or without PDN connection.

0 EPS Attach with PDN connection

1 EPS Attach without PDN connection

Implementation

- Only <n>=3 is supported.
- If <AttachWithoutPDN> is omitted, will use the default value 0.

2.24 APN rate control (+CGAPNRC)

Command	Response	Example
+CGAPNRC[=<cid>] **APN rate control (+CGAPNRC)**	[+CGAPNRC: <cid>[,<Additional_exception_reports>[,<Uplink_time_unit>[,<Maximum_uplink_rate>]]] [<CR><LF>+CGAPNRC: <cid>[,<Additional_exception_reports>[,<Uplink_time_unit>[,<Maximum_uplink_rate>]]] [...]]] *+CME ERROR:<err>*	AT+CGAPNRC +CGAPNRC=1,0,2,2 +CGAPNRC=2,1,2,3 OK AT+CGAPNRC=?
+CGAPNRC=?	+CGAPNRC: (list of <cid>s associated with active contexts) *+CME ERROR:<err>*	+CGAPNRC:1,2 OK

Description

This execution command returns the APN rate control parameters (see 3GPP TS 24.008 [8]) associated to the provided context identifier <cid>.

If the parameter <cid> is omitted, the APN rate control parameters for all active PDP contexts are returned.

The test command returns a list of <cid>s associated with secondary and non secondary active PDP contexts.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<cid>: integer type; specifies a particular PDP context definition (see the +CGDCONT and +CGDSCONT commands).

<Additional_exception_reports>: integer type; indicates whether or not additional exception reports are allowed to be sent when the maximum uplink rate is reached. This refers to bit 4 of octet 1 of the APN rate control parameters IE as specified in 3GPP TS 24.008 [8] subclause 10.5.6.3.2.

0 Additional_exception_reports at maximum rate reached are not allowed to be sent.

1 Additional_exception_reports at maximum rate reached are allowed to be sent.

<Uplink_time_unit>: integer typ; specifies the time unit to be used for the maximum uplink rate. This refers to bits 1 to 3 of octet 1 of the APN rate control parameters IE as specified in 3GPP TS 24.008 [8] subclause 10.5.6.3.2.

0 unrestricted

1 minute

2 hour

3 day

4 week

<Maximum_uplink_rate>: integer type; specifies the maximum number of messages the UE is restricted to send per uplink time unit. This refers to octet 2 to 4 of the APN rate control parameters IE as specified in 3GPP TS 24.008 [8] subclause 10.5.6.3.2.

Implementation

2.25 Sending of originating data via the control plane (+CSODCP)

Send a non-IP message

Command	Response	Example
+CSODCP=<cid>,<cpdata_length>,<cpdata>[,<RAI>[,<type_of_user_data>[,<sequence>]]]	*+CME ERROR:<err>*	AT+CSODCP=1,3,112233 OK
+CSODCP=?	**+CSODCP:(rane of supported <cid>s),(maximum number of bytes of** the <cpdata_length>),(list of supported <RAI>s),(list of supported <type_of_user_data>s),(list of supported <sequence>s)	AT+CSODCP=? +CSODCP:(0-10),(1358),(0,1,2),(0,1),(1-255) OK

Description

The set command is used by the TE to transmit data over control plane to network via MT. Context identifier <cid> is used to link the data to particular context.

This command optionally indicates that the application on the MT expects that the exchange of data:

- will be completed with this uplink data transfer; or
- will be completed with the next received downlink data.

This command also optionally indicates whether or not the data to be transmitted is an exception data.

This command causes transmission of an ESM DATA TRANSPORT message, as defined in 3GPP TS 24.301 [83].

Test command returns the maximum number of bytes of the user data container supported by the MT, supported <RAI>s and supported <type_of_user_data>s as a compound value.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<cid>: integer type. A numeric parameter which specifies a particular PDP context or EPS bearer context definition. The <cid> parameter is local to the TE-MT interface and identifies the PDP or EPS bearer contexts which have been setup via AT command (see the +CGDCONT and +CGDSCONT commands).

<cpdata_length>: integer type. Indicates the number of bytes of the <cpdata> information element. When there is no data to transmit, the value shall be set to zero.

<cpdata>: string of octets. Contains the user data container contents (refer 3GPP TS 24.301 [83] subclause 9.9.4.24). When there is no data to transmit, the <cpdata> shall be an empty string ("").

This parameter shall not be subject to conventional character conversion as per +CSCS. The coding format of the user data container and the maximum length of <cpdata> are implementation specific.

<RAI>: integer type. Indicates the value of the release assistance indication, refer 3GPP TS 24.301 [83] subclause 9.9.4.25.

0 No information available.

1 The MT expects that exchange of data will be completed with the transmission of the ESM DATA TRANSPORT message.

2 The MT expects that exchange of data will be completed with the receipt of an ESM DATA TRANSPORT message.

<type_of_user_data>: integer type. Indicates whether the user data that is transmitted is regular or exceptional.

0 Regular data.

1 Exception data.

<sequence>: Sequence of data, range 1-255.If omit, will not report data sent status.

NOTE: <sequence> is not defined by 3GPP but only used for HiSi.

Implementation

- Maximum data length will be 1358 bytes when non-ip is used, otherwise will be 0.
- Only one message will be buffered at any one time.

2.26 Reporting of terminating data via the control plane (+CRTDCP)

Receive a message from the CDP server.

Command	Response	Example
+CRTDCP=<reporting>	*+CME ERROR:<err>*	AT+CRTDCP=1 OK
+CRTDCP?	+CRTDCP:<reporting>	AT+CRTDCP? +CRTDCP:1 OK
+CRTDCP=?	+CRTDCP:(list of supported <reporting>s),(range of supported <cid>s),(maximum number of octets of user data indicated by <cpdata_length>)	AT+CRTDCP=? +CRTDCP:(0-1),(0-10),(1358) OK

Description

The set command is used to enable and disable reporting of data from the network to the MT that is transmitted via the control plane in downlink direction. If reporting is enabled, the MT returns the unsolicited result code +CRTDCP:<cid>,<cpdata_length>,<cpdata> when data is received from the network.

Read command returns the current settings.

Test command returns supported values as compound values.

Refer to Chapter 6: *Error Values* for possible <err> values.

Command	Response	Example
+CRTDCP:	<cid>,<cpdata_length>,<cpdata>	+CRTDCP:0,2,”ab”

Defined values

<reporting>: integer type, controlling reporting of mobile terminated control plane data events

0 Disable reporting of MT control plane data.

1 Enable reporting of MT control plane data by the unsolicited result code +CRTDCP.

<cid>: integer type. A numeric parameter which specifies a particular PDP context or EPS bearer context definition. The <cid> parameter is local to the TE-MT interface and identifies the PDP or EPS bearer contexts which have been setup via AT command (see the +CGDCONT and +CGDSCONT commands).

<cpdata_length>: integer type. Indicates the number of bytes of the <cpdata> information element. When there is no data to transmit, the value shall be set to zero.

<cpdata>: string of octets. Contains the user data container contents (refer 3GPP TS 24.301 [83] subclause 9.9.4.24). When there is no data to transmit, the <cpdata> shall be an empty string (""). This parameter shall not be subject to conventional character conversion as per +CSCS. The coding format of the user data container and the maximum length of <cpdata> are implementation specific.

Implementation

- Maximum data length will be 1358 bytes when non-ip is used, otherwise will be 0.
- Only one message will be buffered at any one time.
- Terminating data via the control plane before enable reporting will be discarded.

2.27 PDP context read dynamic parameters +CGCONTRDP

Command	Response	Example
+CGCONTRDP[=<cid>]	[+CGCONTRDP: <cid>,<bearer id>,<apn>[,<local_addr and subnet_mask>[,<gw_addr>[,<DNS_prim_addr>[,<DNS_sec_addr>[,<P-CSCF_prim_addr>[,<P-CSCF_sec_addr>[,<IM_CN_Signalling_Flag>[,<LIPA_indication>[,<IPv4_MTU>[,<WLAN_Offload>[,<Local_Addr_Ind>[,<Non-IP_MTU>[,<Serving_PLMN_rate_control_value>[,<Reliable_Data_Service]]]]]]]]]]]]]] [<CR><LF>+CGCONTRDP: <cid>,<bearer_id>,<apn>[,<local_addr and subnet_mask>[,<gw_addr>[,<DNS_prim_addr>[,<DNS_sec_addr>[,<P-CSCF_prim_addr>[,<P-CSCF_sec_addr>[,<IM_CN_Signalling_Flag>[,<LIPA_indication>[,<IPv4_MTU>[,<WLAN_Offload>[,<Local_Addr_Ind>[,<Non-IP_MTU>[,<Serving_PLMN_rate_control_value>[,<Reliable_Data_Service]]]]]]]]]]]]]	AT+CGCONTRDP +CGCONTRDP:0,,,,,10.121.34.223 +CGCONTRDP:1,,,,,10.121.34.224 OK AT+CGCONTRDP=1 +CGCONTRDP:1,,,,,10.121.34.223 OK
+CGCONTRDP=?	[...]] +CGCONTRDP: (list of <cid>s associated with active contexts)	AT+CGCONTRDP=? +CGCONTRDP:(0,1) OK

Description

The execution command returns the relevant information <bearer_id>, <apn>, <local_addr and subnet_mask>, <gw_addr>, <DNS_prim_addr>, <DNS_sec_addr>, <P-CSCF_prim_addr>, <P-CSCF_sec_addr>, <IM_CN_Signalling_Flag>, <LIPA_indication>, <IPv4_MTU>, <WLAN_Offload>, <Non-IP_MTU>, <Serving_PLMN_rate_control_value> and <Reliable_Data_Service> for an active non secondary PDP context with the context identifier <cid>. If the MT indicates more than two IP addresses of P-CSCF servers or more than two IP addresses of DNS servers, multiple lines of information per <cid> will be returned.

If the MT has dual stack capabilities, at least one pair of lines with information is returned per <cid>. First one line with the IPv4 parameters followed by one line with the IPv6 parameters. If this MT with dual stack capabilities indicates more than two IP addresses of P-CSCF servers or more than two IP addresses of DNS servers, multiple of such pairs of lines are returned.

NOTE: If the MT doesn't have all the IP addresses to be included in a line, e.g. in case the UE received four IP addresses of DNS servers and two IP addresses of P-CSCF servers, the parameter value representing an IP address that can not be populated is set to an empty string or an absent string.

If the parameter <cid> is omitted, the relevant information for all active non secondary PDP contexts is returned.

The test command returns a list of <cid>s associated with active non secondary

contexts. **Defined values**

<cid>: integer type; specifies a particular non secondary PDP context definition. The parameter is local to the TE-MT interface and is used in other PDP context-related commands (see the +CGDCONT and +CGDSCONT commands).

<bearer_id>: integer type; identifies the bearer, i.e. the EPS bearer in EPS and the NSAPI in UMTS/GPRS.

<apn>: string type; a logical name that was used to select the GGSN or the external packet data network.

<local_addr and subnet_mask>: string type; shows the IP address and subnet mask of the MT. 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.

When +CGPIAF is supported, its settings can influence the format of this parameter returned with the execute form of +CGCONTRDP.

<gw_addr>: string type; shows the Gateway Address of the MT. The string is given as dot-separated numeric (0-255) parameters.

When +CGPIAF is supported, its settings can influence the format of this parameter returned with the execute form of +CGCONTRDP.

<DNS_prim_addr>: string type; shows the IP address of the primary DNS server.

When +CGPIAF is supported, its settings can influence the format of this parameter returned with the execute form of +CGCONTRDP.

<DNS_sec_addr>: string type; shows the IP address of the secondary DNS server.

When +CGPIAF is supported, its settings can influence the format of this parameter returned with the execute form of +CGCONTRDP.

<P_CSCF_prim_addr>: string type; shows the IP address of the primary P-CSCF server.

When +CGPIAF is supported, its settings can influence the format of this parameter returned with the execute form of +CGCONTRDP.

<P_CSCF_sec_addr>: string type; shows the IP address of the secondary P-CSCF server.

When +CGPIAF is supported, its settings can influence the format of this parameter returned with the execute form of +CGCONTRDP.

<IM_CN_Signalling_Flag>: integer type; shows whether the PDP context is for IM CN subsystem-related signalling only or not.

0 PDP context is not for IM CN subsystem-related signalling only

1 PDP context is for IM CN subsystem-related signalling only

<LIPA_indication>: integer type; indicates that the PDP context provides connectivity using a LIPA PDN connection. This parameter cannot be set by the TE.

0 indication not received that the PDP context provides connectivity using a LIPA PDN connection

1 indication received that the PDP context provides connectivity using a LIPA PDN connection

<IPv4_MTU>: integer type; shows the IPv4 MTU size in octets.

<WLAN_Offload>: integer type; indicates whether traffic can be offloaded using the specified PDN connection via a WLAN or not. This refers to bits 1 and 2 of the WLAN offload acceptability IE as specified in 3GPP TS 24.008 [8] subclause 10.5.6.20.

0 offloading the traffic of the PDN connection via a WLAN when in S1 mode or when in Iu mode is not acceptable.

1 offloading the traffic of the PDN connection via a WLAN when in S1 mode is acceptable, but not acceptable in Iu mode.

2 offloading the traffic of the PDN connection via a WLAN when in Iu mode is acceptable, but not acceptable in S1 mode.

3 offloading the traffic of the PDN connection via a WLAN when in S1 mode or when in Iu mode is acceptable.

<Local_Addr_Ind>: integer type; indicates whether or not the MS and the network support local IP address in TFTs (see 3GPP TS 24.301 [83] and 3GPP TS 24.008 [8] subclause 10.5.6.3).

0 indicates that the MS or the network or both do not support local IP address in TFTs

1 indicates that the MS and the network support local IP address in TFTs

<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 [8] subclause 9.9.4.28.

<Reliable_Data_Service>: integer type; indicates whether the UE is using Reliable Data Service for a PDN connection or not, see 3GPP TS 24.301 [83] and 3GPP TS 24.008 [8] subclause 10.5.6.3.

0 Reliable Data Service is not being used for the PDN connection

1 Reliable Data Service is being used for the PDN connection

Implementation

- Only support <cid>, <DNS_prim_addr>,<DNS_sec_addr> now.
- <DNS_prim_addr>,<DNS_sec_addr> could display only when <PDP_type> is “IP” or “IPV6”.
- Could configure PDP context dynamic parameter items to read by +NCPCDPR.

2.28 Define PDP context authentication parameters (+CGAUTH)

Command	Response	Example
+CGAUTH=<cid>[,<auth prot>[,<userid>[,<password>]]]	*+CME ERROR:<err>*	AT+CGAUTH=1,2,”1234”,”1234” OK
+CGAUTH?	**[+CGAUTH: <cid>,<auth_prot>,<use**rid>,<password>] **[<CR><LF>+CGAUTH: <cid>,<auth_pr**ot>,<userid>,<password> [...]]	AT+CGAUTH? +CGAUTH:1, 1, “1234”, “1234” +CGAUTH:2, 1, “1234”, “1234” OK
+CGAUTH=?	+CGAUTH: (range of supported <cid>s),(list of supported <auth_prot>s),(range of Supported <userid>s),(range of Supported <password>s)	AT+CGAUTH=? +CGAUTH:(0-10),(0,1,2),(0-60),(0-60)

Description

Set command allows the TE 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. Refer subclause 9.2 for possible <err> values.

The read command returns the current settings for each defined context.
The test command returns values supported as compound values.

Defined values

<cid>: integer type. Specifies a particular PDP context definition (see the +CGDCONT and +CGDSCONT commands).

<auth_prot>: integer type. 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

2 CHAP

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

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

Implementation

- <userid> is needed when <auth_prot> is 1 or 2.
- Maximum <userid> string and <password> string length is 60.

2.29 No more PS data (+CNMPSD)

Command	Response	Example
+CNMPSD	*+CME ERROR:<err>*	AT+CNMPSD OK
+CNMPSD=?	*OK*	AT+CNMPSD=? OK

Description

This command indicates that no application on the MT is expected to exchange data.

When in UTRAN, if further conditions defined in 3GPP TS 25.331 [74] are met, this can cause transmission of a SIGNALLING CONNECTION RELEASE INDICATION message with the cause "UE Requested PS Data session end". When in E-UTRAN, if further conditions defined in 3GPP TS 36.331 [86] are met, this can cause transmission of a UEAssistanceInformation message with powerPrefIndication set to "lowPowerConsumption" to the network.

NOTE: See subclause 10.1.38 for a command applicable to an MT using E-UTRAN, where the MT has a preference for a configuration that is not optimized for power saving.

This command may be used in both normal and modem compatibility modes.

Defined values

Implementation

2.30 Enter PIN (+CPIN)

Command	Response	Example
+CPIN=<pin>[,<newpin>]	*+CME ERROR:<err>*	AT+CPIN=”1234 ” OK
+CPIN?	+CPIN:<code> +CME ERROR: <err>	AT+CPIN? +CPIN: READY OK
+CPIN?	OK	

Description

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. Refer to Chapter 6 for possible <err> values.

NOTE 1: SIM PIN, SIM PUK, PH-SIM PIN, PH-FSIM PIN, PH-FSIM PUK, SIM PIN2 and SIM PUK2 refer to the PIN of the selected application on the UICC. For example, in an UTRAN context, the selected application on the currently selected UICC should be a USIM and the SIM PIN then represents the PIN of the selected USIM. See 3GPP TS 31.101 [65] for further details on application selection on the UICC.

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

NOTE 2: Commands which interact with MT that are accepted when MT is pending SIM PIN, SIM PUK, or PH-SIM are: +CGMI, +CGMM, +CGMR, +CGSN, D112; (emergency call), +CPAS, +CFUN, +CPIN, +CPINR, +CDIS (read and test command only), and +CIND (read and test command only). It is implementation specific whether additional commands can be accepted when MT is pending SIM PIN, SIM PUK, or PH-SIM.

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

Defined values

<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 SIM PIN to be given
SIM PUK	MT is waiting SIM PUK to be given
PH-SIM PIN	MT is waiting phone-to-SIM card password to be given
PH-FSIM PIN	MT is waiting phone-to-very first SIM card password to be given
PH-FSIM PUK	MT is waiting phone-to-very first SIM card unblocking password to be given
SIM PIN2	MT is waiting SIM 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 SIM 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)
SIM PUK2	MT is waiting SIM 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
PH-NET PUK	MT is waiting network personalization unblocking password to be given
PH-NETSUB PIN	MT is waiting network subset personalization password to be given
PH-NETSUB PUK	MT is waiting network subset personalization unblocking password to be given
PH-SP PIN	MT is waiting service provider personalization password to be given
PH-SP PUK	MT is waiting service provider personalization unblocking password to be given
PH-CORP PIN	MT is waiting corporate personalization password to be given
PH-CORP PUK	MT is waiting corporate personalization unblocking password to be given

Implementation

Set and read could execute only when power on.

Only support <code> value of “READY”, “SIM PIN”, “SIM PUK”, “SIM PUK BLOCKED”.

2.31 Remaining PIN retries (+CPINR)

Command	Response	Example
+CPINR	*+CME ERROR:<err>*	AT+CPINR OK
+ CPINR=?	OK	AT+CPINR=? OK

Description

Execution command cause the MT to return the number of remaining PIN retries for the MT 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>`.

Refer to Chapter 6 for possible `<err>` values.

Defined values

<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'.
<ext_code>: Extended, manufacturer specific codes.
<sel_code>: String type. Same values as for the <code> and <ext_code> parameters. These values are strings and shall be indicated within double quotes. It is optional to support wildcard match by '*', meaning match any (sub-)string.
Example: AT+CPINR="SIM*" will return the lines:

+CPINR: SIM PIN,<retries>,<default_retries>
+CPINR: SIM PUK,<retries>,<default_retries>
+CPINR: SIM PIN2,<retries>,<default_retries>
+CPINR: SIM PUK2,<retries>,<default_retries>

Example: AT+CPINR="*SIM*" will additionally return the lines:

+CPINR: PH-SIM PIN,<retries>,<default_retries>
+CPINR: PH-FSIM PIN,<retries>,<default_retries>
+CPINR: PH-FSIM PUK,<retries>,<default_retries>

Implementation

Set and read could execute only when power on.

2.32 Open logical channel (+CCHO)

Command	Response	Example
+CCHO=<dfname>	<sessionid> *+CME ERROR:<err>*	AT+CCHO=2 0 OK
+CCHO=?	OK	AT+CCHO=? OK

Description

Execution of the 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 shall restrict the communication between the TE and the UICC to this logical channel.

This <sessionid> is to be used when sending commands with Restricted UICC Logical Channel access +CRLA or Generic UICC Logical Channel access +CGLA commands.

NOTE: The logical channel number is contained in the CLASS byte of an APDU command, thus implicitly contained in all APDU commands sent to a UICC. In this case it will be up to the MT to manage the logical channel part of the APDU CLASS byte and to ensure that the chosen logical channel is relevant to the <sessionid> indicated in the AT command. See 3GPP TS 31.101 [65] for further information on logical channels in APDU commands protocol.

Refer to Chapter 6 for possible <err> values.

Defined values

<dfname>: 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 in order to target a specific application on the smart card (e.g. (U)SIM, WIM, ISIM) using logical channels mechanism

See 3GPP TS 31.101 [65] for more information about defined values.

Implementation

2.33 Close logical channel (+CCHC)

Command	Response	Example
+CCHC=<sessionid>	*+CME ERROR:<err>*	AT+CCHC=1 OK
+CCHC=?	OK	AT+CCHC=? OK

Description

This command asks the ME to close a communication session with the active UICC. The ME shall close 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. Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<sessionid>: integer type; a session Id to be used in order to target a specific application on the smart card (e.g. (U)SIM, WIM, ISIM) using logical channels mechanism.

Implementation

2.34 Generic UICC logical channel access (+CGLA)

Command	Response	Example
+CGLA=<sessionid>,<length>,<command>	+CGLA: <length>,<response> *+CME ERROR:<err>*	AT+CGLA OK
+CGLA=?	OK	AT+CGLA=? OK

Description

Set command transmits to the MT the <command> it then shall send as it is to the selected UICC. In the same manner the UICC <response> shall be sent back by the MT to the TA as it is. Refer Refer to Chapter 6: *Error Values* for possible <err> values.

This command allows a direct control of the currently selected UICC by a distant application on the TE. The TE shall then take care of processing UICC information within the frame specified by GSM/UMTS.

Although Generic UICC Logical Channel Access command +CGLA allows 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. Moreover, for security reason the GSM network authentication should not be handled outside the TA/MT. Therefore it shall not be allowed to execute a Run GSM Algorithm command or an Authenticate command in GSM context from the TE using +CGLA at all time whether the +CGLA is locked or unlocked. This shall not forbid the TE to send Authenticate commands in other security contexts (e.g. EAP security context).

For example, the TA/MT shall forbid the transfer of the Authenticate command to a USIM application when parameters P2 = 0 (GSM security context). See 3GPP TS 31.102 [59] for USIM authenticate command definition.

NOTE: Compared to Restricted UICC Access command +CRLA, the definition of +CGLA allows TE to take more control over the UICC-MT interface. The locking and unlocking of the interface may be done by a special <command> value or automatically by TA/MT (by interpreting <command> parameter). In case that TE application does not use the unlock command (or does not send a <command> causing automatic unlock) in a certain timeout value, MT may release the locking.

Defined values

<sessionid>: integer type; this is the identifier of the session to be used in order to send the APDU commands to the UICC. It is manadatory in order 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 [65] (hexadecimal character format; refer +CSCS)

<response> : response to the command passed on by the UICC to the MT in the format as described in 3GPP TS 31.101 [65] (hexadecimal character format; refer +CSCS).

Implementation

2.35 C Restricted SIM access (+CRSM)

Command	Response	Example
+CRSM=<command>[,<fileid>[,<P1>,<P2>,<P3>[,<data>[,< pathid>]]]]	+CRSM: <sw1>,<sw2>[,<response>] +CME ERROR: <err>	AT+CRSM=176,1 OK
+CRSM=?	OK	AT+CRSM=? OK

Description

By using this command instead of Generic SIM Access +CSIM TE application has easier but more limited access
to the SIM database. Set command transmits to the MT the SIM <command> and its required parameters. If a SIM installed in the currently selected card slot, the MT handles internally all SIM-MT interface locking and file selection routines. As response to the command, MT sends the actual SIM information parameters and response
data. MT error result code +CME ERROR may be returned when the command cannot be passed to the SIM, but failure in the execution of the command in the SIM is reported in <sw1> and <sw2> parameters. Refer to Chapter 6: *Error Values* for possible <err> values.
Coordination of command requests to SIM and the ones issued by GSM/UMTS application inside the MT is implementation dependent. However the TE should be aware of the precedence of the GSM/UMTS application commands to the TE commands.

Defined values

<command>: (command passed on by the MT to the SIM; refer 3GPP TS 51.011 [28]):
176 READ BINARY
178 READ RECORD
192 GET RESPONSE
214 UPDATE BINARY
220 UPDATE RECORD
242 STATUS
203 RETRIEVE DATA
219 SET DATA

all other values are reserved, command 203 and 219 don’ t support now.

NOTE 1: The MT internally executes all commands necessary for selecting the desired file, before performing the actual command.

<fileid>: integer type; this is the identifier of a elementary datafile on SIM. Mandatory for every command except STATUS

NOTE 2: The range of valid file identifiers depends on the actual SIM and is defined in 3GPP TS 51.011 [28]. Optional files may not be present at all.

<P1>, <P2>, <P3>: integer type; parameters passed on by the MT to the SIM. These parameters are mandatory for every command, except GET RESPONSE and STATUS. The values are described in 3GPP TS 51.011 [28]

<data>: information which shall be written to the SIM (hexadecimal character format; refer +CSCS)

<pathid>: string type; contains the path of an elementary file on the SIM/UICC in hexadecimal format as defined in ETSI TS 102 221 [60] (e.g. "7F205F70" in SIM and UICC case). The <pathid> shall only be used in the mode "select by path from MF" as defined in ETSI TS 102 221 [60].

NOTE 3: Since valid elementary file identifiers may not be unique over all valid dedicated file identifiers the <pathid> indicates the targeted UICC/SIM directory path in case of ambiguous file identifiers. For earlier versions of this specification or if <pathid> is omitted, it could be implementation specific which one will be selected.

<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 datafield. This information includes the type of file and its size (refer 3GPP TS 51.011 [28]). After READ BINARY, READ RECORD or RETRIEVE DATA command the requested data will be returned. <response> is not returned after a successful UPDATE BINARY, UPDATE RECORD or SET DATA command.

Implementation

<command> don’ t support 203 and 219.

2.36 Generic SIM access (+CSIM)

Command	Response	Example
+CSIM=<length>,<command>	+CSIM: <length>,<response> +CME ERROR: <err>	AT+CSIM=2,AABB OK
+CSIM=?	OK	**+CSIM=?** OK

Description

Set command transmits to the MT the <command> it then shall send as it is to the SIM. In the same manner, the SIM <response> shall be sent back by the MT to the TA as it is. Refer to Chapter 6: *Error Values*.
for possible <err> values.
This command allows a direct control of the SIM that is installed in the currently selected card slot, by an distant
application on the TE. The TE shall then take care of processing SIM information within the frame specified by GSM/UMTS.
NOTE: Compared to Restricted SIM Access command +CRSM, the definition of +CSIM allows TE to take more control over the SIM-MT interface. The locking and unlocking of the interface may be done by a special <command> value or automatically by TA/MT (by interpreting <command> parameter). In case that TE application does not use the unlock command (or does not send a <command> causing automatic unlock) in a certain timeout value, MT may release the locking.

Defined values

<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 SIM in the format as described in 3GPP TS 51.011 [28] (hexadecimal character format; refer +CSCS)
<response>: response to the command passed on by the SIM to the MT in the format as described in 3GPP TS 51.011 [28] (hexadecimal character format; refer +CSCS)

Implementation

2.37 Enter data state (+CGDATA)

Command	Response	Example
+CGDATA[=<L2P>[,<cid>[,<cid>[,...]]]]	CONNECT ERROR +CME ERROR: <err>	AT+CGDATA="PPP",1 OK
+CGDATA=?	+CGDATA: (list of supported <L2P>s)	AT+CGDATA=? +CGDATA:("PPP") OK

Description

The execution command causes the MT to perform whatever actions are necessary to establish communication
between the TE and the network using one or more Packet Domain PDP types. This may include performing a PS attach and one or more PDP context activations. If the <L2P> parameter value is unacceptable to the MT, the MT shall return an ERROR or +CME ERROR response. Refer subclause 9.2 for possible <err> values. Otherwise, the MT issues the intermediate result code CONNECT and enters V.250 online data state.

NOTE: If the initial PDP context is supported, the context with <cid>=0 is automatically defined at startup, see subclause 10.1.0.

Commands following +CGDATA command in the AT command line shall not be processed by the MT.
The detailed behaviour after the online data state has been entered is dependent on the PDP type. It is described
briefly in 3GPP TS 27.060 [34] and in more detail in 3GPP TS 29.061 [39] and the specifications for the relevant PDPs. PS attachment and PDP context activation procedures may take place prior to or during the PDP startup if they have not already been performed using the +CGATT and +CGACT commands.
If context activation takes place during the PDP startup, one or more <cid>s may be specified in order to provide the information needed for the context activation request(s).
During each PDP startup procedure the MT may have access to some or all of the following information -

The MT may have a priori knowledge, for example, it may implement only one PDP type.

The command may have provided an <L2P> parameter value.

The TE may provide a PDP type and/or PDP address to the MT during in the PDP startup procedure.

If any of this information is in conflict, the command will fail.
Any PDP type and/or PDP address present in the above information shall be compared with the PDP type and/or
PDP address in any context definitions specified in the command in the order in which their <cid>s appear. For a context definition to match:

The PDP type must match exactly.

The PDP addresses are considered to match if they are identical or if either or both addresses are unspecified. For example, a PPP NCP request specifying PDP type = IP and no PDP address would cause the MT to search through the specified context definitions for one with PDP type = IP and any PDP address.

The context shall be activated using the matched value for PDP type and a static PDP address if available, together with the other information found in the PDP context definition. If a static PDP address is not available then a dynamic address is requested.
If no <cid> is given or if there is no matching context definition, the MT shall attempt to activate the context with whatever information is available to the MT. The other context parameters shall be set to their default values.
If the activation is successful, data transfer may proceed.
After data transfer is complete, and the layer 2 protocol termination procedure has completed successfully, the
V.250 command state is re-entered and the MT returns the final result code OK.

In the event of an erroneous termination or a failure to start up, the V.250 command state is re-entered and the
MT returns the final result code NO CARRIER or, if enabled, +CME ERROR. Refer to Chapter 6: *Error Values*
for possible <err> values. Attach, activate and other errors may be reported.
The test command is used for requesting information on the supported layer 2 protocols.
This command may be used in both normal and modem compatibility modes.
.

Defined values

<L2P>: string type; indicates the layer 2 protocol to be used between the TE and MT
NULL none, for PDP type OSP:IHOSS (Obsolete)
PPP Point-to-point protocol for a PDP such as IP
PAD character stream for X.25 character (triple X PAD) mode (Obsolete)
X25 X.25 L2 (LAPB) for X.25 packet mode (Obsolete)
M-xxxx manufacturer-specific protocol (xxxx is an alphanumeric string)
If the value is omitted, the layer 2 protocol is unspecified. Other values are reserved and will result in an ERROR response.
<cid>: integer type; specifies a particular PDP context definition (see the +CGDCONT and +CGDSCONT commands).

Implementation

- Only <L2P> = “PPP” is supported.
- The baud rate needs to be greater than or equal to 57600.
- Could specifies one <cid> one time.

2.38 CIoT optimization configuration (+CCIOTOPT)

Command	Response	Example
+CCIOTOPT=[<n>,[<supported_UE_opt>[,<preferred_UE_opt>]]]	+CME ERROR: <err>	AT+CSIM=2,AABB OK
+CCIOTOPT?	+CCIOTOPT:<n>,<supported_UE_opt> <preferred_UE_opt>	AT+CCIOTOPT? +CCIOTOPT:1,3,2 OK
+CCIOTOPT=?	+CCIOTOPT:(list of supported <n>s),(list of supported <supported_UE_opt>s),(list of supported <preferred_UE_opt>s)	AT+CCIOTOPT=? +CCIOTOPT:(0,1,3),(1,3),(1,2) OK

Description

The set command controls which CIoT 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 CIoT EPS optimizations that are supported by the network. A UE supporting CIoT functionality may support control plane CIoT EPS optimization or user plane CIoT EPS optimization or both (see 3GPP TS 24.301 [83], subclause 9.9.3.34). Based on the application characteristics the UE may prefer to be registered for control plane CIoT EPS optimization or for user plane CIoT EPS optimization (see 3GPP TS 24.301 [83], subclause 9.9.3.0B). Further the network may support control plane CIoT EPS optimization or user plane CIoT EPS optimization or both (see 3GPP TS 24.301 [83], subclause 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 CIoT EPS optimizations by the network.

Refer to Chapter 6: *Error Values* for possible <err> values.

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

The test command returns values supported as compound values.

.

Defined values

<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 CIoT EPS optimization to the default values.

<supported_UE_opt>: integer type; indicates the UE's support for CIoT EPS optimizations.

0 No support.
1 Support for control plane CIoT EPS optimization.
2 Support for user plane CIoT EPS optimization.
3 Support for both control plane CIoT EPS optimization and user plane CIoT EPS optimization.

<preferred_UE_opt>: integer type; indicates the UE's preference for CIoT EPS optimizations.

0 No preference.
1 Preference for control plane CIoT EPS optimization.
2 Preference for user plane CIoT EPS optimization.

<supported_Network_opt>: integer type; indicates the Network support for CIoT EPS optimizations.

0 No support.
1 Support for control plane CIoT EPS optimization.
2 Support for user plane CIoT EPS optimization.
3 Support for both control plane CIoT EPS optimization and user plane CIoT EPS optimization.

Implementation

- <supported_UE_opt> only support 1&3.
- <preferred_UE_opt> only support 1&2.
- Could set <preferred_UE_opt> to 2 only when <supported_UE_opt> is 3.

2.39 PDP context modify (+CGCMOD)

Command	Response	Example
+CGCMOD[=<cid>[,<cid>[,...]]]	+CME ERROR: <err>	AT+CGCMOD=1 OK
+CGCMOD=?	+CGCMOD:(list of <cid>s associated with active contexts)	AT+CGCMOD=? +CGCMOD:(1) OK
+CCIOTOPT=?	+CCIOTOPT:(list of supported <n>s),(list of supported <supported_UE_opt>s),(list of supported <preferred_UE_opt>s)	AT+CCIOTOPT=? +CCIOTOPT:(0,1,3),(1,3),(1,2) OK

Description

The execution command is used to modify the specified PDP context (s) with repect to QoS profiles and TFTs. After the command has completed, the MT returns to V.250 online data state. If the requested modification for any specified context cannot be achieved, an ERROR or +CME ERROR response is returned. Extended error responses are enabled by the +CMEE command. Refer to Chapter 6: *Error Values* for possible <err> values.For EPS, the modification request for an EPS bearer resource will be answered by the network by an EPS bearer modification request. The request must be accepted by the MT before the PDP context is effectively changed.If no <cid>s are specified the activation form of the command modifies all active contexts.
The test command returns a list of <cid>s associated with active contexts..
.

Defined values

<cid>: integer type; specifies a particular PDP context definition (see the +CGDCONT and +CGDSCONT commands).

Implementation

- Could only modify one cid one time.
- Will block other commands when processing.

2.40 Define EPS quality of service (+CGEQOS)

Command	Response	Example
+CGEQOS=[<cid>[,<QCI>[,<DL_GBR>,<UL_GBR>[,<DL_MBR>,<UL_MBR]]]]	+CME ERROR: <err>	AT+CGEQOS=0,0 OK
+CGEQOS?	[+CGEQOS:<cid>,<QCI>,[<DL_GBR>,<UL_GBR>],[<DL_MBR>,<UL_MBR>]] [<CR><LF>+CGEQOS:<cid>,<QCI>,[<DL_GBR>,<UL_GBR>],[<DL_MBR>,<UL_MBR>] [...]]	AT+CGCMOD? +CGCMOD:0,0 OK
+CGEQOS=?	+CGEQOS:(range of supported <cid>s),(list of supported <QCI>s),(list of supported <DL_GBR>s),(list of supported <UL_GBR>s),(list of supported <DL_MBR>s),(list of supported <UL_MBR>s)	AT+CGEQOS=? +CCIOTOPT:(0),(0,5,6,7,8,9,79) OK

Description

The set command allows the TE to specify the EPS Quality of Service parameters <cid>, <QCI>, [<DL_GBR> and <UL_GBR>] and [<DL_MBR> and <UL_MBR>] for a PDP context or Traffic Flows (see 3GPP TS 24.301 [83] and 3GPP TS 23.203 [85]). When in UMTS/GPRS the MT applies a mapping function to UTMS/GPRS Quality of Service. Refer to Chapter 6: *Error Values* for possible <err> values.
A special form of the set command, +CGEQOS= <cid> causes the values for context number <cid> to become undefined.
The read command returns the current settings for each defined QoS.
The test command returns the ranges of the supported parameters as compound values.
.

Defined values

<cid>: integer type; specifies a particular EPS Traffic Flows definition in EPS and a PDP Context definition in UMTS/GPRS (see the +CGDCONT and +CGDSCONT commands).
<QCI>: integer type; specifies a class of EPS QoS (see 3GPP TS 23.203 [85] and 3GPP TS 24.301 [83]).
0 QCI is selected by network

[1 – 4] value range for guaranteed bit rate Traffic Flows

75 value for guaranteed bit rate Traffic Flows

[5 – 9] value range for non-guarenteed bit rate Traffic Flows

79 value for non-guaranteed bit rate Traffic Flows

[128 – 254] value range for Operator-specific QCIs

The QCI values 65, 66, 69 and 70 are not allowed to be requested by the UE. If the TE requests a QCI parameter
65, 66, 69 or 70, the MT responds with result code +CME ERROR: 181 (unsupported QCI value).

<DL_GBR>: integer type; indicates DL GBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI (see 3GPP TS 24.301 [83]).

<UL_GBR>: integer type; indicates UL GBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI (see 3GPP TS 24.301 [83]).

<DL_MBR>: integer type; indicates DL MBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI (see 3GPP TS 24.301 [83]).

<UL_MBR>: integer type; indicates UL MBR in case of GBR QCI. The value is in kbit/s. This parameter is omitted for a non-GBR QCI (see 3GPP TS 24.301 [83]).

Implementation

- Could set <QCI> only and only <QCI> = 0 & 5 & 6 & 7 & 8 & 9 & 79 are supported.
- Could execute only when power on and pdp context defined.

2.41 Traffic flow template (+CGTFT)

Command	Response	Example
+CGTFT=[<cid>,[<packet filter identifier>,<evaluation precedence index>[,<remote address and subnet mask>[,<protocol number (ipv4) / next header (ipv6)>[,<local port range>[,<remote port range>[,<ipsec security parameter index (spi)>[,<type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>[,<flow label (ipv6)>[,<direction>[,<local address and subnet mask>]]]]]]]]]]]	+CME ERROR: <err>]	AT+CGTFT=0,2,6,"32.1.11.160.0.0.0.0.0.0.0.0.0.0.0.0.255.255.255.255.0.0.0.0.0.0.0.0.0.0.0.0",17,"60001.60001","60350.60450",0x123456,168.252,0x12345,1
+CGTFT?	[+CGTFT: <cid>,<packet filter identifier>,<evaluation precedence index>,<remote address and subnet mask>,<protocol number (ipv4) / next header (ipv6)>,<local port range>,<remote port range>,<ipsec security parameter index (spi)>,<type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>,<flow label (ipv6)>,<direction>,<local address and subnet mask>] [<CR><LF>+CGTFT: <cid>,<packet filter identifier>,<evaluation precedence index>,<remote address and subnet mask>,<protocol number (ipv4) / next header (ipv6)>,<local port range>,<remote port range>,<ipsec security parameter index (spi)>,<type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>,<flow label (ipv6)>,<direction>,<local address and subnet mask>	AT+CGTFT? +CGTFT:0,2,6,"2001:BA0::.FFFF:FFFF::",17,"60001.60001","60350.60450",0x123456,"168.252",0x12345,1 OK
+CGTFT=?	+CGTFT: <PDP_type>,(list of supported <packet filter identifier>s),(list of supported <evaluation precedence index>s),(list of supported <remote address and subnet mask>s),(list of supported <protocol number (ipv4) / next header (ipv6)>s),(list of supported <local port range>s),(list of supported <remote port range>s),(list of supported <ipsec security parameter index (spi)>s),(list of supported <type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>s),(list	+CGTFT:"IP",(1,16),(0,255),"(0.0.0.1,255.255.255.255).(0.0.0.1,255.255.255.255)",(0,255),"(0,65535).(0,65535)","(0,65535).(0,65535)",(0x0,0xFFFFFFFF),"(0,255).(0,255)",,(0,3)) +CGTFT:"NONIP",(1,16),(0,255),,(0,255),"(0,65535).(0,65535)","(0,65535).(0,65535)",(0x0,0xFFFFFFFF),"(0,255)(0,255)",,(0,3))

	of supported <flow label (ipv6)>s),(list of supported <direction>s),(list of supported <local address and subnet mask>s) [<CR><LF>+CGTFT: <PDP_type>,(list of supported <packet filter identifier>s),(list of supported <evaluation precedence index>s),(list of supported <remote address and subnet mask>s),(list of supported <protocol number (ipv4) / next header (ipv6)>s),(list of supported <local port range>s),(list of supported <remote port range>s),(list of supported <ipsec security parameter index (spi)>s),(list of supported <type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>s),(list of supported <flow label (ipv6)>s),(list of supported <direction>s),(list of supported <local address and subnet mask>s) [...]]	+CGTFT:"IPV6",(1,16),(0,255),"(::1,FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF).(::1,FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)",(0,255),"(0,65535).(0,65535)","(0,65535).(0,65535)",(0x0,0xFFFFFFFF),"(0,255).(0,255)",(0x0,0xFFFFF),(0,3)) +CGTFT:"IPV4V6",(1,16),(0,255),"(0.0.0.1,255.255.255.255)\(::1,FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF).(0.0.0.1,255.255.255.255)\(::1,FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)",(0,255),"(0,).(0,65535)",(0x0,0xFFFFFFFF),"(0,255).(0,255)",(0x0,0xFFFFF),(0,3)) OK

Description

This command allows the TE to specify a Packet Filter - PF for a Traffic Flow Template - TFT that is used in the GGSN in UMTS/GPRS and Packet GW in EPS for routing of packets onto different QoS flows towards the TE. The

concept is further described in the 3GPP TS 23.060 [47]. A TFT consists of from one and up to 16 Packet Filters, each identified by a unique <packet filter identifier>. A Packet Filter also has an <evaluation precedence index> that is unique within all TFTs associated with all PDP contexts that are associated with the same PDP address.

The set command specifies a Packet Filter that is to be added to the TFT stored in the MT and used for the context identified by the (local) context identification parameter, <cid>. The specified TFT will be stored in the GGSN in UMTS/GPRS and Packet GW in EPS only at activation or MS-initiated modification of the related context.

Since this is the same parameter that is used in the +CGDCONT and +CGDSCONT commands, the +CGTFT command is effectively an extension to these commands. The Packet Filters consist of a number of parameters,each of which may be set to a separate value.

A special form of the set command, +CGTFT=<cid> causes all of the Packet Filters in the TFT for context number <cid> to become undefined. At any time there may exist only one PDP context with no associated TFT amongst all PDP contexts associated to one PDP address. At an attempt to delete a TFT, which would violate this rule, an ERROR or +CME ERROR response is returned. Extended error responses are enabled by the +CMEE command. Refer to Chapter 6 for possible <err> values.

The read command returns the current settings for all Packet Filters for each defined context.

The test command returns values supported as compound values. If the MT supports several PDP types, the parameter value ranges for each PDP type are returned on a separate line. TFTs shall be used for PDP-type IP and PPP only. For PDP-type PPP a TFT is applicable only when IP traffic is carried over PPP. If PPP carries headercompressed IP packets, then a TFT cannot be used.

.

Defined values

<cid>: integer type. Specifies a particular PDP context definition (see the +CGDCONT and +CGDSCONT commands).

<PDP_type>: string type. Specifies the type of packet data protocol (see the +CGDCONT command).

<packet filter identifier>: integer type. Value range is from 1 to 16.

<evaluation precedence index>: integer type. The value range is from 0 to 255.

<remote address and subnet mask>: string type. The string is given as dot-separated numeric (0-255) parameters on the form:
"a1.a2.a3.a4.m1.m2.m3.m4" for IPv4 or
"a1.a2.a3.a4.a5.a6.a7.a8.a9.a10.a11.a12.a13.a14.a15.a16.m1.m2.m3.m4.m5.m6.m7.m8.m9.m10.m11.m1 2.m13.m14.m15.m16", for IPv6.
When +CGPIAF is supported, its settings can influence the format of this parameter returned with the read form of +CGTFT.

<protocol number (ipv4) / next header (ipv6)>: integer type. Value range is from 0 to 255.

<local port range>: string type. The string is given as dot-separated numeric (0-65535) parameters on the form "f.t".

<remote port range>: string type. The string is given as dot-separated numeric (0-65535) parameters on the form "f.t".

<ipsec security parameter index (spi)>: numeric value in hexadecimal format. The value range is from 00000000 to FFFFFFFF.

<type of service (tos) (ipv4) and mask / traffic class (ipv6) and mask>:

string type. The string is given as dot-separated numeric (0-255) parameters on the form "t.m".

<flow label (ipv6)>: numeric value in hexadecimal format. The value range is from 00000 to FFFFF. Valid for IPv6 only.

<direction>: integer type. Specifies the transmission direction in which the packet filter shall be applied.

0 Pre-Release 7 TFT filter (see 3GPP TS 24.008 [8], table 10.5.162)
1 Uplink
2 Downlink
3 Birectional (Up & Downlink)

<local address and subnet mask>: string type. The string is given as dot-separated numeric (0-255) parameters on the form:
"a1.a2.a3.a4.m1.m2.m3.m4" for IPv4 or
"a1.a2.a3.a4.a5.a6.a7.a8.a9.a10.a11.a12.a13.a14.a15.a16.m1.m2.m3.m4.m5.m6.m7.m8.m9.m10.m11.m1 2.m13.m14.m15.m16", for IPv6.

When +CGPIAF is supported, its settings can influence the format of this parameter returned with the read form of +CGTFT.

Some of the above listed attributes may coexist in a Packet Filter while others mutually exclude each other, the possible combinations are shown in 3GPP TS 23.060 [47].

Implementation

- Could execute only when power on and pdp context defined.
- <local address and subnet mask> is not supported yet.

3 Commands(3GPP aligned to 27.005)

3.1 Select Message Service (+CSMS)

Command	Response	Example
+CSMS=<service>	+CSMS:<mt>,<mo>,<bm> +CMS ERROR:<err>	AT+CSMS=1 +CSMS:1,1,1 OK
+CSMS?	+CSMS:<service>,<mt>,<mo>,<bm>	AT+CSMS? +CSMS:1,1,0,1 OK
+CSMS=?	+CSMS: (list of supported <service>s)	AT+CSMS=? +CSMS:(0,1) OK

Description
Set command selects messaging service <service>. It returns the types of messages supported by the ME: <mt> for mobile terminated messages, <mo> for mobile originated messages and <bm> for broadcast type messages. If chosen service is not supported by the ME (but is supported by the TA), final result code +CMS ERROR: <err> shall be returned. Also read command returns supported message types along the current service setting. Test command returns a list of all services supported by the TA.

Refer to Chapter 6: *Error Values* for possible <err> values

Defined values

<service>: integer type

0	3GPP TS 23.040 [3] and 3GPP TS 23.041 [4]
1	3GPP TS 23.040 [3] and 3GPP TS 23.041 [4] the requirement of <service> setting 1 is mentioned under corresponding command descriptions)
2...127	reserved
128...	manufacturer specific

<mt>, <mo>, <bm>: integer type

0	type not supported
1	type supported

Implementation

- MT will report to user if received SMS messages:
  +CMT: [<alpha>],<length><CR><LF><pdu> (PDU mode enabled); (According to +CNMI in 3GPP TS 27005)

<alpha> Just to comply with 3GPP TS 27005, no need to care about it and always to be 0.

3.2 New Message Acknowledgement to ME/TA (+CNMA)

Command	Response	Example
+CNMA=[<n>[,<length>[<CR> *PDU is given*<ctrl-Z/ESC>]]]	*+CMS ERROR:<err>*	AT+CNMA=1 OK
+CNMA=?	+CNMA:(list of supported <n>s)	AT+CNMA=? +CNMA:(1,2) OK

Description
Execution command confirms reception of a new message (SMS-DELIVER or SMS-STATUS-REPORT) which is routed directly to the TE (refer command +CNMI table 3.4.1-3 and table 3.4.1-5). This acknowledgement command shall be used when +CSMS parameter <service> equals 1. In PDU mode, it is possible to send either positive (RPACK) or negative (RP-ERROR) acknowledgement to the network. Parameter <n> defines which one will be sent. Optionally (when <length> is greater than zero) an acknowledgement TPDU (SMS-DELIVER-REPORT for RPACK or RP-ERROR) may be sent to the network. The entering of PDU is done similarly as specified in command Send Message +CMGS, except that the format of

<ackpdu> is used instead of <pdu> (i.e. SMSC address field is not present). PDU shall not be bounded by double quotes. TA shall not send another +CMT or +CDS result code to TE before previous one is acknowledged.

If ME does not get acknowledgement within required time (network timeout), ME should respond as specified in 3GPP TS 24.011 [6] to the network. ME/TA shall automatically disable routing to TE by setting both <mt> and <ds> values of +CNMI to zero.

If command is executed, but no acknowledgement is expected, or some other ME related error occurs, final result code +CMS ERROR: <err> is returned

> NOTE: In case that a directly routed message must be buffered in ME/TA (possible when +CNMI parameter <mode> equals 0 or 2) or AT interpreter remains too long in a state where result codes cannot be sent to TE (e.g. user is entering a message using +CMGS), acknowledgement (RP-ACK) must be sent to the network without waiting +CNMA command from TE. Later, when buffered result codes are flushed to TE, TE must send +CNMA[=0] acknowledgement for each result code. In this way, ME/TA can determine if message should be placed in non-volatile memory and routing to TE disabled (+CNMA[=0] not received). Refer command +CNMI for more details how to use <mode> parameter reliably.

Test command returns a list of supported <n> values. If the only value supported is 0, the device does not support sending of TPDU.

Refer to Chapter 6: *Error Values* for possible <err> values

Defined Values

<n>: integer type

0	command operates similarly as defined for the text mode (UE don't support text mode for SMS currently)
1	send RP-ACK (or buffered result code received correctly)
2	send RP-ERROR (if PDU is not given, ME/TA shall send SMS-DELIVER-REPORT with 3GPP TS 23.040 [3] TP-FCS value set to 'FF' (unspecified error cause))

<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)

Implementation

- Only <n>=1 and <n>=2 are supported.
- +CNMI is not supported now.
- Length range 0 – 232.
- MT will report to user if received SMS messages:

+CMT: [<alpha>],<length><CR><LF><pdu> (PDU mode enabled);

(According to +CNMI in 3GPP

TS

27005)

<alpha>

Just to comply with 3GPP TS 27005, no need to care about it and always to be 0.

3.3 Service Centre Address (+CSCA)

Command	Response	Example
+CSCA=<sca>[,<tosca>]	+CMS ERROR:<err>	AT+CSCA=358501234567,145 OK
+CSCA?	+CSCA:<sca>[,<tosca>]	AT+CSCA? +CSCA:”358501234567”,145 OK
+CSCA=?		AT+CSCA=? OK

Description Set command updates the SMSC address, through which mobile originated SMs are transmitted. 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.

Refer to Chapter 6: *Error Values* for possible <err> values

Defined values

<sca> 3GPP TS 24.011 [6] RP SC address Address-Value field in string format; BCD numbers (or GSM 7 bit default alphabet characters) are converted to characters of the currently selected TE character set (refer command +CSCS in 3GPP TS 27.007 [9]); type of address given by <tosca>

<tosca> 3GPP TS 24.011 [6] RP SC address Type-of-Address octet in integer format (when first character of <da> is + (IRA 43) default is 145, otherwise default is 129)

Implementation

- Could get or set service centre address only when power on.
- <sca>: length range 1 – 20.

3.4 Send SMS Message (+CMGS)

Command	Response	Example

+CMGS=<length><CR>
***PDU is given*<ctrl-Z/ESC>**
	if sucessful
	+CMGS:<mr>[,<ackpdu>]]
	if fails
	+CMS ERROR:<err>
		AT+CMGS=8
		1A1B1C1D1F1G1C1D
		+CMGS:1
		OK

+CMGS=?		AT+CMGS=? OK

Description

Execution command sends message from a TE to the network (SMS-SUBMIT). Message reference value <mr> is returned to the TE on successful message delivery. Optionally (when +CSMS <service> value is 1 and network supports) <ack pdu> is returned. Values can be used to identify message upon unsolicited delivery status report result code. If sending fails in a network or an ME error, final result code +CMS ERROR: <err> is returned. This command should be abortable.

- <length> must indicate the number of octets coded in the TP layer data unit to be given (i.e. SMSC address octets are excluded).
- the TA shall send a four character sequence <CR><LF><greater_than><space> (IRA 13, 10, 62, 32) after command line is terminated with <CR>; after that PDU can be given from TE to ME/TA.
- the DCD signal shall be in ON state while PDU is given.
- the echoing of given characters back from the TA is controlled by V.25ter echo command E.
- the PDU shall be hexadecimal format (similarly as specified for <pdu>) and given in one line; ME/TA converts this coding into the actual octets of PDU.

- when the length octet of the SMSC address (given in the PDU) equals zero, the SMSC address set with command Service Centre Address +CSCA is used; in this case the SMSC Type-of-Address octet shall not be present in the PDU, i.e. TPDU starts right after SMSC length octet.
- sending can be cancelled by giving <ESC> character (IRA 27).
- <ctrl-Z> (IRA 26) must be used to indicate the ending of PDU.

Refer to Chapter 6: *Error Values* for possible <err>

values **Defined values**

<mr> 3GPP TS 23.040 [3] TP-Message-Reference 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)

<ackpdu> 3GPP TS 23.040 [3] RP-User-Data element of RP-ACK PDU; format is same as for <pdu> in case of SMS, but without 3GPP TS 24.011 [6] SC address field and parameter shall be bounded by double quote characters like a normal string type parameter

Implementation

- <length> range 7-220.
- Don't return ackpdu now.
- For address field in PDU(3GPP TS 24.011), should put the country code at the start of it (eg, China:86).

3.5 Send SMS Command (+CMGC)

Command	Response	Example

+CMGC=<length><CR>
***PDU is given*<ctrl-Z/ESC>**
	if successful
	+CMGC:<mr>[,<ackpdu>]
	if fails
	+CMS ERROR:<err>
		AT+CMGC=8
		1A1B1C1D1F1G1C1D
		+CMGC:1
		OK

+CMGC=?		AT+CMGC=? OK

Description

Execution command sends a command message from a TE to the network (SMS-COMMAND). The entering of text (3GPP TS 23.040 [3] TP-Command-Data) is done similarly as specified in command Send Message +CMGS, but the format is fixed to be a sequence of two IRA character long hexadecimal numbers which ME/TA converts into 8-bit octets (refer +CMGS). Message reference value <mr> is returned to the TE on successful message delivery. Optionally (when +CSMS <service> value is 1 and network supports) <ackpdu> is returned. Values can be used to identify message upon unsolicited delivery status report result code. If sending fails in a network or an ME error, final result code +CMS ERROR: <err> is returned.. This command should be abortable.

Refer to Chapter 6: *Error Values* for possible <err>

values **Defined values**

<mr> 3GPP TS 23.040 [3] TP-Message-Reference 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)

<ackpdu> 3GPP TS 23.040 [3] RP-User-Data element of RP-ACK PDU; format is same as for <pdu> in case of SMS, but without 3GPP TS 24.011 [6] SC address field and parameter shall be bounded by double quote characters like a normal string type parameter

Implementation

- <length> range 8-220.
- Don’t return ackpdu now.
- For address field in PDU (3GPP TS 24.011), should put the country code at the start of it (eg, China:86).

3.6 Send SMS Command (+CMGC)

Command	**Response**	**Example**
+CMMS=[<n>]	+CMS ERROR:<err>	AT+CMMS=0 OK
+CMMS?	+CMMS:<n>	AT+CMMS? +CMMS:0 OK
+CMMS=?	+CMMS:(list of supported <n>s)	AT+CMMS=? +CMMS:(0,1,2) OK

Description

Set command controls the continuity of SMS relay protocol link. When feature is enabled (and supported by network) multiple messages can be sent much faster as link is kept open.

Defined values

<n>: integer type

0 disable (default value)

1 keep enabled until the time between the response of the latest message send command (+CMGS, +CMSS, etc.) and the next send command exceeds 1-5 seconds (the exact value is up to ME implementation), then ME shall close the link and TA switches <n> automatically back to 0

2 enable (if the time between the response of the latest message send command and the next send command exceeds 1-5 seconds (the exact value is up to ME implementation), ME shall close the link but TA shall not switch automatically back to <n>=0)

Implementation

4 Commands(General)

4.1 Reboot (+NRB)

Command	Response	Example
+NRB	REBOOTING<CR><LF>	AT+NRB REBOOTING

Description

This command reboots the terminal. There is a short delay after issuing this command before the terminal reboots. No further AT commands will be processed.
Refer to Chapter 6: *Error Values* for possible <err> values.

Note that there is no final OK to signal that the command line has finished processing as AT command processing terminates with this command. No confirmation messages are expected until the reboot.

Defined values

Implementation

- +nrb has the highest priority, can be used at any time except when executing a blocking and abortable
  AT command or AT command being interrupted.

4.2 Query UE Statistics (+NUESTATS)

Command	Response	Example
+NUESTATS=RADIO	Signal power:<signal power in centibels> Total power:<total power in centibels> TX power:<current Tx power level in centibels > TX time:<total Tx time since last reboot in millisecond> RX time:<total Rx time since last reboot in millisecond> Cell ID:<last cell ID> ECL:<last ECL value> SNR:< last snr value> EARFCN:< last earfcn value> PCI:< last pci value> RSRQ:<rsrq in centibels> OPERATOR MODE:<operator mode> CURRENT BAND:<current band> *+CME ERROR:<err>*	AT+NUESTATS=RADIO NUESTATS:RADIO,Signal power,50 NUESTATS:RADIO,Total power,500 NUESTATS:RADIO,TX power,30 NUESTATS:RADIO,TX time,1234567 NUESTATS:RADIO,RX time,12345 NUESTATS:RADIO,Cell ID,70 NUESTATS:RADIO,ECL,1 NUESTATS:RADIO,SNR:20 NUESTATS:RADIO,EARFCN:30 NUESTATS:RADIO,PCI:11 NUESTATS:RADIO,RSRQ:0 NUESTATS:RADIO,OPERATOR MODE:1 NUESTATS:RADIO,CURRENT BAND:8 OK
+NUESTATS=CELL	NUESTATS:CELL,<earfcn>,<physical cell id>,<primary cell>,<rsrp>,<rsrq>,<rssi>,<snr> [... NUESTATS:CELL,<earfcn>,<physical cell id>,<primary cell>,<rsrp>,<rsrq>,<rssi>,<snr> *+CME ERROR:<err>* i	AT+NUESTATS=CELL NUESTATS:CELL,3569,69,1,23,-1073,-1145,5 NUESTATS:CELL,3569,69,0,23,-1073,-1145,5 NUESTATS:CELL,3569,69,0,23,-1073,-1145,5 OK
+NUESTATS=THP	NUESTATS:THP,<throughput type>,<throughput> [... NUESTATS:THP,<throughput type>,<throughput>] *+CME ERROR:<err>*	AT+NUESTATS=THP NUESTATS:THP,RLC UL,100 NUESTATS:THP,RLC DL,98 NUESTATS:THP,MAC UL,103 NUESTATS:THP,MAC DL,100 OK
+NUESTATS=BLER	NUESTATS:BLER,<block error rate type>,< block error rate > [... NUESTATS:BLER,<block error rate type>,< block error rate >] *+CME ERROR:<err>*	AT+NUESTATS=BLER NUESTATS:BLER,RLC UL BLER,10 NUESTATS:BLER,RLC DL BLER,5 NUESTATS:BLER,MAC UL BLER,8 NUESTATS:BLER,MAC DL BLER,3 NUESTATS:BLER,Total TX bytes,1080 NUESTATS:BLER,Total RX bytes,900 NUESTATS:BLER,Total TX blocks,80 NUESTATS:BLER,Total RX blocks,80 NUESTATS:BLER,Total RTX blocks,100 NUESTATS:BLER,Total ACK/NACK RX,100 OK
AT+NUESTATS=APPSMEM	APPSMEM:Current Allocated,<allocated> APPSMEM:Total Free,<free> APPSMEM:Max Free,<max free> APPSMEM:Num Allocs,<num allocs> APPSMEM:Num Frees,<num frees> APPSMEM:Current Allocated,<allocated>	AT+NUESTATS=APPSMEM APPSMEM:Current Allocated,8240 APPSMEM:Total Free,198 APPSMEM:Max Free,8496 APPSMEM:Num Allocs, 300 APPSMEM:Num Frees, 240 OK

+NUESTATS=<type>	*NUESTATS:<type>,<name/value>,<value>[,<value[,<value>[...]]] [... NUESTATS:<type>,<name/value>,<value>[,<value[,<value>[...]]]] +CME ERROR:<err>*	
+NUESTATS=?	NUESTATS:<supported types> *+CME ERROR:<err>*	AT+NUESTATS=? NUESTATS:(RADIO,CELL,BLER,THP,APPSMEM, ALL) OK

Description

This command fetches the most recent operational statistics. Refer to Chapter 6: *Error Values* for possible <err> values. It can take an optional parameter that allows different sets of statistics to be displayed. The <type> RADIO provides the default set of values. <type>=ALL will print all data.

Defined values

<*type*> Type of data to be displayed as an unquoted string.

Supported values of <type> are:

RADIO	radio specific information
CELL	per-cell information for the top 8 cells
BLER	block error rate information
THP	throughput
APPSMEM	application Core dynamic memory usage
ALL	all information. The value of <type> output is the correct one for each data type.

<*type*> = RADIO

<signal power in centibels>
<total power in centibels>
<current TX power level in centibels >
<total TX time since last reboot in millisecond>
<total RX time since last reboot in millisecond>
<last SIB1 cell ID>
<last ECL value>
< last snr value>
< last earfcn value>
< last pci value>
<rsrq in centibels>

<operator mode> operator mode for SIB1.

0	Unknown mode.
1	Inband different pci mode.
2	Inband same pci mode.
3	Guardband mode.
4	Standalone mode.

<current band> The band of Service cell.

<*type*> = CELL

per-cell information for the top 5 cells.

Returned entries are of the form:

<earfcn>,<physical cell id>,<primary cell>,<rsrp>,<rsrq>[,<rssi>,<snr>]

Neighbourhood cells won’t output <rssi> and <snr>.

<physical cell id>
<primary cell>

<rsrp>
<rsrq>
<rssi>
<snr>

physical id of the cell
1 indicates the current serving cell
0 indicates the neighbourhood cells searchedl
reference signal received power
reference signal received quality

received signal strength indicator
signal to noise ratio

<*type*> = BLER
block error rate
<rlc_ul_bler> RLC layer block error rate (uplink). Integer %
<rlc_dl_bler> RLC layer block error rate (downlink). Integer %
<mac_ul_bler> physical layer block error rate (uplink). Integer %
<mac_dl_bler> physical layer block error rate (downlink). Integer %
<total bytes transmitted>
<total bytes received>
<transport blocks sent>
<transport blocks received>
<transport blocks retransmitted>
<total ack/nack messages received>

<*type*> = THP
throughput
<rlc_ul> RLC layer throughput (uplink). Integer bps
<rlc_dl> RLC layer throughput (downlink). Integer bps
<mac_ul> physical layer throughput (uplink). Integer bps
<mac_dl> physical layer throughput (downlink). Integer bps

<*type*> = APPSMEM
Application Core dynamic memory usage
<allocated>
<free>
<max free >
<num allocs >
<num frees>

Implementation

- The variant of NUESTATS without an argument prints out the RADIO arguments without the command and variant prefixes. This will be deprecated in a future release.
- +NUESTATS and +NUESTATS=RADIO, CELL, BLER, THP will be reset to invalid value when power off or out of service.
- +NUESTATS and +NUESTATS=RADIO have default invalid values, hey are
  Signal power:-32768
  Total power:-32768
  TX power:-32768
  TX time:0
  RX time:0
  Cell ID:4294967295
  ECL:255
  SNR:-32768
  EARFCN:4294967295
  PCI:65535
  RSRQ:-32768
  OPERATOR MODE:0
  CURRENT BAND:255

4.3 Specify search frequencies (+NEARFCN)

Command	Response	Example
+NEARFCN=<search mode>[,<param1>[,<param2>[,...]]]	*+CME ERROR:<err>*	AT+NEARFCN=0,10,10A OK

+NEARFCN=?		+NEARFCN=? OK

Description

The set command provides a mechanism to lock to a specific E-ULTRA Absolute Radio Frequency Channel Number (EARFCN) and, if desired, Physical Cell ID. All actions will be locked to this carrier until either the lock is removed or the UE is rebooted. It is not persistent over reboots.

If the specified EARFCN is not present, the UE will enter out of service mode. If the specified PCI is not present, the UE will enter out of service mode.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

`<search_mode>`

`<earfcn>`

`<pci>`

Specifies the type of search and defines the supplied parameters.

0 `<earfcn>,<pci>`

lock to a specific earfcn

A number in the range 0-65535 representing the earfcn to search. An `<earfcn>` value of 0 will remove the earfcn restriction and any associated Physical Cell ID lock.

string type; E-UTRAN physical cell ID in hexadecimal format. Valid range 0 - 1F7.

4.4 Create Socket (+NSOCR)

Create a socket and associate with specified protocol.

Command	Response	Example
+NSOCR=<type>,<protocol>[,<listen port>[,<receive control>[,<af_type>[,<ip address>]]]]	<socket> *+CME ERROR:<err>*	AT+NSOCR=DGRAM,17,56,1,AF INET 1 OK

Description

This command creates a socket on the UE. If the port is set, receiving is enabled and unsolicited +NSONMI messages will appear for any message that is received on that port. Refer to Chapter 6: *Error Values* for possible <err> values.

If a socket has already been created for a protocol, port combination, +NSOCR will fail if requested a second time.

Defined values

`<type>` Socket Type. DGRAM for UDP, STREAM for TCP

`<protocol>` Standard internet protocol definition. For example, UDP is 17.

`<listen port>` A number in the range 0-65535. This is the local port that will be included in

`<socket>` sent messages and on which messages will be received. If it is 0 or omitted, LWIP will assign a random `<listen port>` for this socket.

This is a reference to the created socket. It is an integer greater than or equal to 0

`<receive control>` Set to 1 if incoming messages should be received, 0 if incoming messages should be ignored. Defaults to 1 (messages will be received)

`<af_type>` String type, “AF_INET” for IPv4 and “AF_INET6” for IPv6. Default value “AF_INET”.

`<ip address>` The ip address NW assigned to MT.

Implementation

- A maximum of 7 sockets are supported, but other services may reduce this number.
- Only UDP(protocol 17), TCP(protocol 6) supported.
- `<listen port>` value of 20000 for TCP can’t be set when bip is enabled.

4.5 SendTo Command (UDP only) (+NSOST)

Send a UDP datagram containing length bytes of data to remote_port on remote_addr.

Command	Response	Example
+NSOST=<socket>,<remote addr>,<remote_port>,<length>,<data>[,<sequence>]	<socket>,<length> *+CME ERROR:<err>*	AT+NSOST=1,192.158.5.1,1024,2,AB30,1 1,2 OK

Description

This command sends a UDP datagram to the specified host:port. It will return with the socket and the number of bytes of data to be sent. If the amount of data is larger than the largest datagram that can be sent, the +NSOST return value will indicate how much of the data was successfully sent.

If <sequence>is not omitted, when datagram is sent by rf or discard, will report the result.

Command	Response	Example
	+NSOSTR:<socket>,<sequence>,<status>	+NSOSTR:1,2,1

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<socket> Socket number returned by +NSOCR
<remote addr> IP address
<remote port> A number in the range 0-65535. This is the remote port that messages will be received on
<length> Decimal length of data to be sent.
<data> Data to be transmitted in hexstring format, or quoted string format.
<sequence> Sequence of data, range 1-255.If omit, will not report data sent status.
<status> The status of datagram.
0 Error
1 Sent

Implementation

- Maximum data size is 1358 bytes.
- <data> Only hexstring format is supported
- IP addresses can be specified in decimal, octal or hexadecimal notation.
- Data can’t be transferred if MT don’t have address (could use +cgpaddr to get).

4.6 SendTo Command with Flags (UDP only) (+NSOSTF)

Send a UDP datagram containing length bytes of data to remote_port on remote_addr and allows meta-data flags to be set.

Command	Response	Example
+NSOSTF=<socket>,<remote addr>,<remote_port>,<flag>,<length>,<data>[,<sequence>]	<socket>,<length> *+CME ERROR:<err>*	AT+NSOSTF=1,192.158.5.1,1024,0x100,2,AB30,1 1,2 OK

Description

This command sends a UDP datagram to the specified host:port. It will return with the socket and the number of bytes of data to be sent. If the amount of data is larger than the largest datagram that can be sent, the +NSOSTF return value will indicate how much of the data was successfully sent.

If <sequence>is not omitted, when datagram is sent by rf or discard, will report the result.

Command	Response	Example
	+NSOSTR: <socket>,<sequence>,<status>	+NSOSTR:1,2,1

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<socket> Socket number returned by +NSOCR
<remote addr> IP address

<remote port> A number in the range 0-65535. This is the remote port that messages will be received on

<flag> Specifies the type of message transmission. Values of this argument are in hex format and are formed by logically OR'ing zero or more of the following flags:
0x100 Exception Message: Send message with high priority
0x200 Release Indicator: indicate release after next message
0x400 Release Indicator: indicate release after next message has been replied to
If no flags are set, a value of 0 should be provided.

<length> Decimal length of data to be sent.

<data> Data to be transmitted in hexstring format.

<sequence> Sequence of data, range 1-255.If omit, will not report data sent status.

<status> The status of datagram.
0 Error
1 Sent

Implementation

- Maximum data size is 1358 bytes.
- <data> Only hexstring format is supported
- IP addresses can be specified in decimal, octal or hexadecimal notation.
- Data can’t be transferred if MT don’t have address (could use +cgpaddr to get).

4.7 Query Socket Messages Sent Pending Data List (+NQSOS)

Command	Response	Example
+NQSOS=<socket>[,<socket>[,<socket>[...]]]	[+NQSOS:<socket>,<sequence><CR><LF>[+NQSOS:<socket>,<sequence>][…]]	AT+NQSOS=1,2 +NQSOS:1,2 +NQSOS:2,3 OK
+NQSOS?	[+NQSOS:<socket>,<sequence><CR><LF>[+NQSOS:<socket>,<sequence>][…]] *+CME ERROR:<err>*	AT+NQSOS? +NQSOS:1,2 +NQSOS:2,3 OK

Description

This command queries the list of the pending upstream message by MT.
Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<socket>
<sequence>
Socket number.
The sequence of pending
up steam message. Range 1-255.

4.8 Receive Command (+NSORF)

Command	Response	Example
+NSORF=<socket>,<req_length>	<socket>,<ip addr>,<port>,<length>,<data>,<remaining_length> *+CME ERROR:<err>*	AT+NSORF=1,10 1,192.168.5.1,1024,2,ABAB,0 OK

Description

Reads up to <req length> characters of data from <socket>.
Returned length is the actual number of characters returned.

Receive data on a socket. When data arrives a +NSONMI response will be generated that indicates the socket the message was received on and the amount of data. The +NSORF command takes a length, which is the maximum amount of data that will be returned.

If the requested length is larger than the actual size of the returned data, only the length of returned data is provided, and the remaining length is returned as 0. If the requested length is less than the amount of data returned, only the requested amount of data will be returned, plus an indication of the number of bytes

remaining.. Once a message has been fully read, a new +NSONMI notification will be sent if there is another message to process.
Refer to Chapter 6: *Error Values* for possible <err> values.

If messages arrive faster than they are read, and the internal message buffer is full, the most recent message will be discarded.

Defined values

<socket>
<req_length>
<remote addr>
<remote port>

<length>
<remaining_length>
<data>

Socket number returned by +NSOCR
Maximum amount of data to be returned as a decimal byte length.
Address of system sending the message
A number in the range 0-65535. This is the remote port that messages was sent from
Amount of data returned as a decimal byte length
Amount of data left to read for this message as a decimal byte length
Data received in hexstring format

Implementation

- Maximum data size is 1358 bytes.
- Remaining length is always 0. The remaining data is readable.

4.9 Connect Command (TCP only) (+NSOCO)

Connect to a TCP server to `remote_port` on `remote_addr`.

Command	Response	Example
+NSOCO=<socket>,<remote addr>, <remote_port>	*+CME ERROR: <err>*	AT+NSOCO=1,192.158.5.1,1024 OK

Description

This command Connect to a TCP sever to the specified host:port.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values <socket> <remote_addr> <remote_port>

Socket number returned by +NSOCR
Address of system sending the message
A number in the range 0-65535. This is the remote port that connect to.

Implementation

- IP addresses can be specified in decimal, octal or hexadecimal notation.

4.10 Send Command (TCP only) (+NSOSD)

Send a TCP datagram containing `length` bytes of `data` to TCP server.

Command	Response	Example
+NSOSD=<socket>,<length>,<data> [,<flag>[,sequence]]	socket, length *+CME ERROR: <err>*	AT+NSOSD=1,2,AB30 1,2 OK AT+NSOSD=1,2,AB30,0x100

		1,2 OK AT+NSOSD=1,2,AB30,0x100,255 1,2 OK

Description

This command sends a TCP datagram to the TCP server. It will return with the socket that it was sent on, and the number of bytes of data sent. If the amount of data is larger than the largest datagram that can be sent, the +NSOSD return value will indicate how much of the data was successfully sent.

If <sequence>is not omitted, when datagram is acked by server or discard by UE, will report the result.

Command	Response	Example
	+NSOSTR: <socket>,<sequence>,<status>	+NSOSTR:1,2,1

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<socket> Socket number returned by +NSOCR.

<length> Decimal length of data to be sent.

<data> Data to be transmitted in hexstring format.

<flag> Specifies the type of message transmission. Values of this argument are in hex format and are formed by logically OR'ing zero or more of the following flags:

0x100 Exception Message: Send message with high priority

0x200 Release Indicator: indicate release after next message

0x400 Release Indicator: indicate release after next message has been replied to

If no flags are set, a value of 0 should be provided.

<sequence> Sequence of data, range 1-255.If omit, will not report data sent status.

<status> The status of datagram.

0 Error

1 Sent

Implementation

- Maximum data size is 1358 bytes.
- <data> Only hexstring format is supported
- IP addresses can be specified in decimal, octal or hexadecimal notation.
- <flag> couldn’t work now.

4.11 Close Socket (+NSOCL)

Command	Response	Example
+NSOCL=<socket>	*+CME ERROR:<err>*	AT+NSOCL=1 OK

Description

Close the specified socket. If there are pending messages to be read, they will be dropped. No further unsolicited +NSONMI notifications will be generated. If the socket has already been closed, or was never created, an error will be returned. Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values <socket>

Socket number returned by +NSOCR

Implementation

4.12 Socket message arrived indicator (+NSONMI) (Response Only)

Command	Response	Example
+NSONMI:	<socket>,<length>	+NSONMI:1,10

Description

Unsolicited message to notify that data has been received on a socket and is ready to be read. Returns socket number and number of bytes of data available to read for the first message that is queued. If another message is received on the same socket, it will only be notified when the preceding message has been completely read.
Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values <socket> <length>

Socket on which data is received. Decimal number returned by +NSOCR Number of bytes of data
in the first message.

Implementation

- This message can occur at any point if it is indicating a new message with no messages buffered. If there are buffered messages it will occur in the AT+NSORF command before the data is returned.

4.13 Socket close indicator (+NSOCLI) (Response Only)

Command	Response	Example
+NSOCLI:	<socket>	+NSOCLI:1

Description
Unsolicited message to notify that socket has been closed by LWIP internally. Returns socket number.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values
<socket>

Socket on which is closed. Decimal number returned by +NSOCR.

Implementation

- Only TCP could report +NSOCLI.

4.14 Test IP network connectivity to a remote host (+NPING)

Command	Response	Example
+NPING=<remote_address>[, <p_size>[,<timeout>]]	*+CME ERROR:<err>*	AT+NPING=192.168.1.1 OK
+NPING	+NPING:<remote_address>,<ttl>,<rtt>	+NPING:192.168.1.1,20,50
+NPINGERR	+NPINGERR:<err>	+NPINGERR:1

Description
This command sends an ICMP packet to the specified host address.
Refer to Chapter 6: *Error Values* for possible <err> values.

AT+NPING initiates the sending of a PING packet to the specified address. This will either cause a packet to be returned if the remote system is connected and responding to PING packets, or no response will be received. . A maximum of 1 ping attempts will be tried. If none of the packets receive a response within the timeout period, an error will be raised.

If a response is received, the unsolicited +NPING message will be returned. If no response is received the +NPINGERR unsolicited response will be returned with an error value.

Defined values

<remote addr>	address of system sending the message
<p_size>	size in bytes of echo packet payload. Range of 12-1500 Default 12
<timeout>	maximum time in ms to wait for an echo reply response.

Range 10-600000.
Default 10000

<ttl> ttl received in the response packet

<rtt> elapsed time in msec from packet sent to response received.

<err> An integer value to provide some information on why the ping request failed.

1	No response from remote host within timeout period
2	Failed to send ping request

Implementation

- IP addresses can be specified in decimal, octal or hexadecimal notation.
- Data can’t be transferred if MT don’t have address (could use +cgpaddr to get).

4.15 Set Debug Logging Level (+NLOGLEVEL)

Command	Response	Example
+NLOGLEVEL=<core>,<level>	+CME ERROR:<err>	AT+NLOGLEVEL=PROTOCOL,ERROR OK
+NLOGLEVEL?	+NLOGLEVEL:<core>,<level>	AT+NLOGLEVEL? +NLOGLEVEL:PROTOCOL,ERROR +NLOGLEVEL:SECURITY,NONE +NLOGLEVEL:APPLICATION,WARNING OK
+NLOGLEVEL=?	+NLOGLEVEL:(<core>,...),(<level>,<level>,...)	AT+NLOGLEVEL=? +NLOGLEVEL:(PROTOCOL,APPLICATION,SECURITY),(VERBOSE,NORMAL,WARNING,ERROR,NONE) OK

Description

This command sets the logging level. It can take one of the following values: This value is persistent across reboots.

Defined values

<core> Core required
PROTOCOL
SECURITY
APPLICATION

<level> Logging level required
VERBOSE
NORMAL
WARNING
ERROR
NONE

Implementation

- Default logging level is NORMAL

4.16 Configure UE Behaviour (+NCONFIG)

Command	Response	Example
+NCONFIG=<function>,<value>	+CME ERROR:<err>	AT+NCONFIG=AUTOCONNECT,TRUE OK
+NCONFIG?	+NCONFIG:<function>,<value> [+NCONFIG:<function>,<value> [..]]	AT+NCONFIG? +NCONFIG:AUTOCONNECT,FALSE +NCONFIG:CR_0354_0338_SCRAMBLING,TRUE +NCONFIG:CR_0859_SI_AVOID,TRUE +NCONFIG:COMBINE_ATTACH,FALSE +NCONFIG:CELL_RESELECTION,TRUE +NCONFIG:ENABLE_BIP,FALSE +NCONFIG:MULTITONE,TRUE +NCONFIG:NAS_SIM_POWER_SAVING_ENABLE,TRUE +NCONFIG:BARRING_RELEASE_DELAY,64 +NCONFIG:RELEASE_VERSION,13 +NCONFIG:RPM,FALSE +NCONFIG:SYNC_TIME_PERIOD,0 +NCONFIG:IPV6_GET_PREFIX_TIME,15 +NCONFIG:NB_CATEGORY,2 +NCONFIG:RAI,FALSE +NCONFIG:HEAD_COMPRESS,FALSE +NCONFIG:RLF_UPDATE,TRUE +NCONFIG:CONNECTION_REESTABLISHMENT,FA

		LSE +NCONFIG:TWO_HARQ,FALSE +NCONFIG:PCO_IE_TYPE,EPCO +NCONFIG:T3324_T3412_EXT_CHANGE_REPORT,FA LSE +NCONFIG:NON_IP_NO_SMS_ENABLE,FALSE +NCONFIG:SUPPORT_SMS,TRUE OK
AT+NCONFIG=<function>	+NCONFIG: <function>,<value>	AT+NCONFIG=AUTOCONNECT +NCONFIG:AUTOCONNECT,FALSE OK

+NCONFIG=?	+NCONFIG:(<function>,(<value1>,<value2>[,<value3>[,...]])
	[+NCONFIG:(<function>,(<value1>,<value2>[,<value3>[,...]])
	[,...]]
		AT+NCONFIG=?
		+NCONFIG:(AUTOCONNECT,(FALSE,TRUE))
		+NCONFIG:(CR_0354_0338_SCRAMBLING,(FALSE,TRUE))
		+NCONFIG:(CR_0859_SI_AVOID,(FALSE,TRUE))
		+NCONFIG:(COMBINE_ATTACH,(FALSE,TRUE))
		+NCONFIG:(CELL_RESELECTION,(FALSE,TRUE))
		+NCONFIG:(ENABLE_BIP,(FALSE,TRUE))
		+NCONFIG:(MULTITONE,(FALSE,TRUE))
		+NCONFIG:(NAS_SIM_POWER_SAVING_ENABLE,(FA
		LSE,TRUE))
		+NCONFIG:(BARRING_RELEASE_DELAY,(0-1800))
		+NCONFIG:(RELEASE_VERSION,(13,14))
		+NCONFIG:(RPM,(FALSE,TRUE))
		+NCONFIG:(SYNC_TIME_PERIOD,(0-65535))
		+NCONFIG:(IPV6_GET_PREFIX_TIME,(0-65535))
		+NCONFIG:(NB_CATEGORY,(1,2))
		+NCONFIG:(RAI,(FALSE,TRUE))
		+NCONFIG:(HEAD_COMPRESS,(FALSE,TRUE))
		+NCONFIG:(RLF_UPDATE,(FALSE,TRUE))
		+NCONFIG:(CONNECTION_REESTABLISHMENT,(FAL
		SE,TRUE))
		+NCONFIG:(TWO_HARQ,(FALSE,TRUE))
		+NCONFIG:(PCO_IE_TYPE,(PCO,EPCO))
		+NCONFIG:(T3324_T3412_EXT_CHANGE_REPORT,(
		FALSE,TRUE))
		+NCONFIG:(NON_IP_NO_SMS_ENABLE,(FALSE,TRU
		E))
		+NCONFIG:(SUPPORT_SMS,(FALSE,TRUE))
		OK

Description

This command allows the configuration of certain aspects of UE behavior. It takes a function and a value that controls operation of that function.

Defined values

<function> UE function to configure

AUTOCONNECT

Control if the platform will automatically attempt to connect to the network after power-on or reboot. When enabled, will set+ CFUN=1 and read the plmn from the SIM. It will use the APN provided by the network.

COMBINE_ATTACH

Enable/disable combine attach.

CELL_RESELECTION

Enable support for RRC cell reselection

ENABLE_BIP

Enable/disable bip.

MULTITONE

Enable/disable multitone. Need rf support multitone too if enable multitone.

NAS_SIM_POWER_SAVING_ENABLE

Enable/disable sim card power saving mode.

BARRING_RELEASE_DELAY

Time (in seconds) to delay release from barring. Value from 0-1800.

Should only be set when bar release timer not running.

RELEASE_VERSION

Release version, only 13 and 14 are supported. Could set to 13 only when MT don’t use power class 6.

RPM

Enable/disable rpm,if RPM files present on SIM the SIM settings will have precedence

SYNC_TIME_PERIOD

The sync time period from eNB. Unit is hour, range 0-65535. 0 means close sync time.

IPV6_GET_PREFIX_TIME

Set the maximum time of getting IPV6 prefix. Unit is second, range 0-65535. 0 means to get IPV6 prefix immediately, 65535 close to get IPV6 prefix.

NB_CATEGORY

Configure the NB category. Only support 1 and 2 now.

RAI

Enable/Disable release assistance indication

HEAD_COMPRESS

Enable/Disable head compress.

RLF_UPDATE

“FALSE” represents the occurrence of radio link failure, NAS will TAU immediately. And “TRUE” representative when occurred radio link failure, ACK is expected before TAU.

CONNECTION_REESTABLISHMENT

Enable/Disable connection reestablishment.

TWO_HARQ

Enable/Disable two harq

PCO_IE_TYPE

Pco ie type. Support “PCO” and“EPCO”.

T3324_T3412_EXT_CHANGE_REPORT

Enable/Disable display unsolicited AT result for +CEREG=5 when NAS timer for <Active-Time> and/or <Periodic-TAU> is changed.

NON_IP_NO_SMS_ENABLE

“TRUE” represents when <PDP_type> is NONIP, will not support SMS.

SUPPORT_SMS

Enable/Disable SMS.

Implementation

- CR_ functions are temporary and will be retired when no longer required.
- MULTITONE,NAS_SIM_POWER_SAVING_ENABLE,RELEASE_VERSION,SYNC_TIME_PERIOD, NB_CATEGORY, RAI,HEAD_COMPRESS, PCO_IE_TYPE could set only when MT is power off.

4.17 Set ID (+NTPERMID)

Migrate system identities into permanent storage. This is a temporary command that will be replaced with a production tool

Command	Response	Example
+NTPERMID=<snt>,<dest>	+CME ERROR:<err>	AT+NTPERMID=1,1 OK
+NTPERMID=?	+NTPERMID:(list of supported <snt>s),(list of supported <dest>s)	AT+NTPERMID=? +NTPERMID:(0,1),(1,2) OK

Description

Move serial number or IMEI into non-modifiable storage. This is either flash that can’t be written or one-time-programmable storage that can never be erased.

Defined values

<snt> integer type indicating the serial number type that has been requested.

0 sets <sn>

1 sets the IMEI (International Mobile station Equipment Identity)

<dest>

1 permanent flash

2 one-time programmable storage

Implementation

- This command must be executed when the radio is inactive (AT+CFUN=0 will force this state).

4.18 Set ID (+NTSETID)

Set system identities. This is a temporary command that will be replaced with a production tool

Command	Response	Example
+NTSETID=<snt>,<data>	+CME ERROR:<err>	AT+NTSETID=1,123456789012345 OK

Description

Set uuid or IMEI value.

Defined values

<snt> integer type indicating the serial number type that has been requested.

0	sets <sn>
1	sets the IMEI (International Mobile station Equipment Identity)
2	returns the SVN (Software Version Number)

<data>

0	<sn>
1	<imei>
2	<svn>

<sn>

The 128-bit UUID of the UE as a hexstring.

<imei>

15-character string type in decimal format

<svn>

2 digit SVN

Implementation

- The Serial Number and IMEI values assigned is persistent across reboots.
- SVN is not persistent and will default to 0x00.
- Need to send AT+NCDP to configure CDP Server again after set new IMEI. Otherwise, UE will use the old IMEI to connect with CDP Server.

4.19 Card Identification (+NCCID)

Command	Response	Example
+NCCID	+NCCID:<ICCID> +CME ERROR:<err>	AT+NCCID +NCCID:4412345678901234567 OK
+NCCID?	+NCCID:<ICCID> +CME ERROR:<err>	AT+NCCID? +NCCID:4412345678901234567 OK
+NCCID=?		AT+NCCID=? OK

Description

Both the execute and read commands read the ICCID off the SIM card. If no SIM card is present, or the SIM card is unreadable, no data is returned.

Refer to Chapter 6: *Error Values* for possible <err> values

Defined Values

<ICCID> SIM Card Identification Number

4.20 Set Supported Bands (+NBAND)

Command	Response	Example
+NBAND=n[,n[,n[...]]]	+CME ERROR:<err>	AT+NBAND=5,8 OK
+NBAND?	+NBAND:n[,n[,n[...]]]	AT+NBAND? +NBAND:5,8,28,3 OK
+NBAND=?	+NBAND:(n[,n[,n[...]]])	AT+NBAND=?

		+NBAND: (1,2,3,5,8,12,13,14,17,18,19,20,25,26,28,66) OK

Description
This command restricts the set of bands to be used. Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<n>

Band as a decimal number.

Implementation

- AT+NBAND=? returns the list of bands supported by the hardware
- +NBAND set could be exclude only when MT is power off.

4.21 Set the mapping for band and power class (+NPOWERCLASS)

Command	Response	Example
+NPOWERCLASS =<band>,<power class>	+CME ERROR:<err>	AT+NPOWERCLASS=5,3 OK
+NPOWERCLASS?	[+NPOWERCLASS:<band>,<power class><CR><LF>[+NPOWERCLASS:<band>,<power class>][…]] +CME ERROR:<err>	AT+NPOWERCLASS? +NPOWERCLASS:5,3 OK
+NPOWERCLASS=?	+NPOWERCLASS:(list of supported <band>s),(list of supported <power class>s)	AT+NPOWERCLASS=? +NPOWERCLASS:(5,8,20),(3,5,6) OK

Description
Set the mapping for band and power class.
The read command list all mapping of bands and power class.
Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<band>

<power class>

Configured band by +NBAND set.

Implementation

- <power class> only support 3, 5, 6 now.
- <power class> could set to 6 only when <function> "RELEASE_VERSION" in +NCONFIG is 14.
- Could set only when MT is power off.

4.22 Power saving mode status report(+NPSMR)

Command	Response	Example
+NPSMR=<n>	+CME ERROR:<err>	AT+NPSMR=0 OK
+NPSMR?	+NPSMR:<n>[,<mode>] +CME ERROR:<err>	AT+NPSMR? +NPSMR:1,1 OK
+NPSMR=?	+NPSMR:(list of supported <n>s)	AT+NPSMR=? +NPSMR:(0,1) OK
	+NPSMR:<mode>	+NPSMR:1

Description

The set command controls the presentation of an unsolicited result code +NPSMR.
If <n>=1, +NPSMR: <mode> is sent from MT when the power mode of MT is changed.
The read command returns +NPSMR:<n> when <n> is 0, and return +NPSMR:<n>,<mode> when <n> is 1.
Test command returns supported values as a compound value.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<n>: integer type
  0 disable unsolicited result code
  1 enable unsolicited result code +NPSMR: <mode>
<mode>: integer type; indicates the power mode of MT.
  0 normal mode
  1 power saving mode

Implementation

4.23 Paging time window value and eDRX setting (+NPTWEDRXS)

Command	Response	Example
+NPTWEDRXS=[<mode>,[,<AcT-type>[,<Requested_Paging_time_window>[,<Requested_eDRX_value>]]]]	+CME ERROR: <err>	AT+NPTWEDRXS=1,5,”1110”,”0101”
+NPTWEDRXS?	[+NPTWEDRXS: <AcT-type>,<Requested_Paging_time_window>,<Requested_eDRX_value> [<CR><LF>+NPTWEDRXS: <AcT-type>,<Requested_Paging_time_window>,<Requested_eDRX_value> [...]]]	AT+NPTWEDRXS? +NPTWEDRXS:5,”1110”,”0101” OK
+NPTWEDRXS=?	+NPTWEDRXS: (list of supported <mode>s),(list of supported <AcT-type>s) ,(list of supported <Requested_Paging_time_window>s),(list of supported <Requested_eDRX_value>s)	AT+NPTWEDRXS=? +NPTWEDRXS:(0,1,2,3),(5) , (“0000”-“1111”) , (“0000”-“1111”) OK

Command	Response	Example
Requested_Paging_time_window>[,<Requested_eDRX_value>]]]]		
+NPTWEDRXS?	[+NPTWEDRXS: <AcT-type>,<Requested_Paging_time_window>,<Requested_eDRX_value> [<CR><LF>+NPTWEDRXS: <AcT-type>,<Requested_Paging_time_window>,<Requested_eDRX_value> [...]]]	AT+NPTWEDRXS? +NPTWEDRXS:5,”1110”,”0101” OK
+NPTWEDRXS=?	+NPTWEDRXS: (list of supported <mode>s),(list of supported <AcT-type>s) ,(list of supported <Requested_Paging_time_window>s),(list of supported <Requested_eDRX_value>s)	AT+NPTWEDRXS=? +NPTWEDRXS:(0,1,2,3),(5) , (“0000”-“1111”) , (“0000”-“1111”) OK

Description

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

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

A special form of the command can be given as +NPTWEDRXS=3. In this form, paging time window and eDRX will be disabled and data for all parameters in the command +NPTWEDRXS will be removed or, if available, set to the manufacturer specific default values.
The read command returns the current settings for each defined value of <AcT-type>.

The test command returns the supported <mode>s and the value ranges for the access technology and the requested paging time window and requested eDRX value as compound values.

Refer to Chapter 6: *Error Values* for possible <err>
values **Defined values**

<mode>: integer type, indicates to disable or enable the use of requested paging time window and 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 requested paging time window and eDRX

1 Enable the use of requested paging time window and eDRX

2 Enable the use of requested paging time window and eDRX and enable the unsolicited result code +NPTWEDRXP:<AcT-type>[,< Requested_Paging_time_window>][,<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.

1 EC-GSM-IoT (A/Gb mode)

2 GSM (A/Gb mode)

3 UTRAN (Iu mode)

4 E-UTRAN (WB-S1 mode)

5 E-UTRAN (NB-S1 mode)

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

<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 subclause 10.5.5.32 of 3GPP TS 24.008 [8]). For the coding and the value range, see Extended DRX parameters information element in 3GPP TS 24.008 [8] Table 10.5.5.32/3GPP TS 24.008. The default value, if available, is manufacturer specific.

<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 subclause 10.5.5.32 of 3GPP TS 24.008 [8]). For the coding and the value range, see Extended DRX parameters information element in 3GPP TS 24.008 [8] 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 referes to bit 8 to 5 of octet 3 of the Extended DRX parameters information element (see subclause 10.5.5.32 of 3GPP TS 24.008 [8]). For the coding and the value range, see the Extended DRX parameters information element in 3GPP TS 24.008 [8] Table 10.5.5.32/3GPP TS 24.008.

Implementation

- <AcT-type> value 5 is supported.

4.24 Firmware Update (+NFWUPD)

Command	Response	Example
+NFWUPD=<cmd>[,<sn>,<len>,<data>,<crc>]	*<result>* *+CME ERROR:<err>*	AT+NFWUPD=0 OK
+NFWUPD=?	+NFWUPD:(list of supported <cmd>s)	AT+NFWUPD=? +NFWUPD:(0-5) OK

Description

This command supports Firmware Updating. It allows erase package zone, package download, checking package last validation result, package name and version querying and firmware upgrading.

Before upgrading firmware, it needs to erase package zone and download package first. After downloading finished, it sends firmware upgrading command. Then the system would reboot and validate the package, and if the package is legal, it would start to upgrade the firmware to new version, otherwise it would reboot and not to upgrade, then the user can check the package validation error information with corresponding AT command.

Defined values

<cmd> Package process command

0 Erase package zone in internal flash

1 <sn>,<len>,<data>,<crc>
Download a package segment. Download a FOTA package.
Package segments are continuous segments of the FOTA package.
Segments should be length of 32/64/128/256/512, and must be provided in order.

2 Get the package validation result of last updating

3 Get package name

4 Get package version

5 Upgrade firmware

<sn> Sequence number. It start with 0, and increments by one for each package segment.

<len> Data length in bytes of data, should be 32/64/128/256/512.

<data> Data to be transmitted in hexstring format,

<crc> CRC of the package segment binary data. CRC is sent as a Hex String.
The <crc> is an xor8 of each byte in the package segment.,

Implementation

4.25 Configure AT Uart Baud Rate (+NATSPEED)

Command	Response	Example
+NATSPEED=<baud rate>,<timeout>,<store>,<sync mode>[,<stop bits>[,<parity>[,<xonxoff>]]]	*+CME ERROR: <err>*	AT+NATSPEED=9600,30,1,3,1,0,0 OK
+NATSPEED?	+NATSPEED:<baud rate>,<sync mode>,<stop bits>,<parity>,<xonoff>	AT+NATSPEED? +NATSPEED:9600,3,1,0,0 OK
+NATSPEED=?	+NATSPEED:(<baud rate>),(<timeout>),(<store>),(<sync mode>),(<stop bits>),(parity),(xonxoff)	+NATSPEED:(4800,9600,57600,115200,230400,460800),(0-30),(0,1),(0-3),(1,2),(0-2),(0,1) OK

Description

Configure AT uart baud rate.

Refer to Chapter 6: *Error Values* for possible <err> values

Defined values

<baud rate> Integer type indicating the at uart baud rate that has been requested.

<timeout> Integer type indicating the time to wait for communication before switching back to the original speed, unit second. Default value is 3 secondes. Max value is 30.

<store> Integer type indicating parameter <baud rate> and <sync mode> store to nv or not. Default value is 0.

0 Do not store to nv, need configure again after reboot.

1 Will store to nv, while there is interaction before timeout.

<sync mode> The LP UART synchronies to each start bit that it detects and uses this to configure its optimum sampling point for each subsequent bit in a data word. The Sync Mode field allows this sampling point to be modified if required.

0 Default value for most operations

1 Sample later

2 Sample earlier

3 Sample even earlier

<stop bits> The AT UART stop bits, default value is 1.

1 1 stop bit

2 2 stop bits

<parity> The AT UART parity, default value is 0.

0 No parity enabled

1 Odd parity

2 Even parity

<xonxoff> AT UART Software (XON/XOFF) Flow Control

0 Software flow control disabled

1 Software flow control enabled

Implementation

- `<baud rate>` is higher than the fastest speed supported by the Low Power UART will disable Deep Sleep Low Power Operation.
- `<sync mode>` may be removed in a future release.
- <sync mode> don't support 3 when <baud rate> is 4800 and 57600.

4.26 PIN operator (+NPIN)

Command	Response	Example
+NPIN=<command>,<parameter1>[,< parameter2>]	+CME ERROR:<err> +NPIN:<npin result>	AT+NPIN=0,29102394 OK +NPIN:OK

Description

Set command is used to verify, change, enable, disable, or unblock pin. Will receive an unsolicited result when execute pin command is completed.

NOTE 1: SIM PIN, SIM PUK refer to the PIN of the selected application on the UICC. For example, in an UTRAN context, the selected application on the currently selected UICC should be a USIM and the SIM PIN then represents the PIN of the selected USIM. See 3GPP TS 31.101 [65] for further details on application selection on the UICC.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

`<command>`

0 <pin>
Verify pin.

1 <old pin>,<new pin>
Change pin.

2 <pin>
Enable pin

3 <pin>
Disable pin.

4 <puk>,<pin>
Unblock pin.

`<pin>,<old pin>, <new pin>` Decimal string format; `<old pin>` shall be the same as password specified for the facility from the MT user interface or with command `AT+NPIN=1,<old pin>, <new pin>` and `<new pin>` is the new password; minimum length is 4, maximum length is 8.

`<puk>` Decimal string format, string size is 8.

`<npin result>` "OK"

"ERROR PIN disabled"

"ERROR PIN blocked"

"ERROR wrong PIN <pin retries remaining>"

"ERROR wrong format"

"ERROR"

`<pin retries remaining>` Decimal type, remaining PIN retries. If no retries left

then PIN is blocked,need to input

+npin=4,<puk>,<pin> to unblock pin.

Implementation

- Need send AT+CFUN=1 to power on MT first.

4.27 Debug and Exception Log (+NXLOG)

Command	Response	Example
AT+NXLOG=<Command>[,parameter]	[<report>] +CME ERROR:<err>	AT+NXLOG=DUMP OK
AT+NXLOG?	+NXLOG:<log mode> +CME ERROR:<err>	AT+NXLOG? AT+NXLOG:1 OK
AT+NXLOG=?	NXLOG:(<list of supported commands>) +CME ERROR:<err>	AT+NXLOG=? NXLOG:(DUMP,DELETE,ENABLE,DISABLE) OK

Description

NXLOG provides an interface to debug and exception log. The debug log is defined for debug. The exception log stores all debug error message of verbosity ERROR.
It allows the debug log to be enabled, disabled or cleared.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<command>

DUMP	<dump mode> dump out the debug and exception log.
DELETE	delete all debug and exception log data
ENABLE	enable logging of debug data
DISABLE	disable logging of debug data

<report>	hexstring encoded debug and exception log when <dump mode> is 0. binary encoded debug and exception log when <dump mode> is 1.
<dump mode>	dump mode, default value is 0. 0: dump over at port 1: dump over debug serial port
<log mode>	log mode, default value is 0. 0: disable logging of debug data 1: enable logging of debug data

Implementation

- <command>= DELETE / ENABLE/DISABLE and +NXLOG? could work only when using external flash.

4.28 Clear Stored Earfcn (+NCSEARFCN)

Command	Response	Example
+NCSEARFCN	*+CME ERROR:<err>*	AT+NCSEARFCN OK

Description

This command clear Stored Earfcn.
Refer to Chapter 6: *Error Values* for possible <err> values.

Implementation

- Could exclude only when MT is power off.

4.29 IP address info Report(+NIPINFO)

Command	Response	Example
+NIPINFO=<n>	+CME ERROR:<err>	AT+NIPINFO=1 OK
+NIPINFO?	+NIPINFO:<n>	AT+NIPINFO? AT+NIPINFO:1 OK
+NIPINFO=?	+NIPINFO:(list of supported <n>s)	AT+NIPINFO=? +NIPINFO:(0,1) OK

Description

This command is used to report IP address information

The set command controls the presentation of an unsolicited result code .

If the IP address is obtained successfully, <failure_cause> will be omitted.

The format: +NIPINFO:<cid>,<IP_type>,<IP_addr>.

If it fails to obtain an IP address, <IP_addr> will be omitted.

The format: +NIPINFO:<cid>,<IP_type>,,<failure_cause>.

The read command returns the status of the current setting <n>.

Test command returns supported parameter values.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<n>: integer type;

0 disable unsolicited result code (default value)

1 enable unsolicited result code +NIPINFO:<cid>,<IP_type>,[<IP_addr>][,<failure_cause>]

<cid>: integer type; specifies a particular PDP context definition (see the +CGDCONT and +CGDSCONT commands).

<IP_type>: string type; specifies the type of IP address.

IP	Internet Protocol (IETF STD 5 [103])
IPV6	Internet Protocol, version 6
IPV4V6	Virtual <PDP_type> introduced to handle dual IP stack UE capability.

<IP_addr>: string type; IP address. The IPV4 address is expressed in decimal numbers and dots, such as 100.1.0.26. IPV6 addresses are expressed in hexadecimal numbers and colon, such as 108F: 0: 0: 0: 8: 800: 200C: 417A. This parameter is not displayed when the address acquisition failed

<failure_cause>: integer type; Cause values for failed to get IP address

1	PDN type IPv4 only allowed
2	PDN type IPv6 only allowed
3	Single address bearers only allowed
4	IPV6 RA timeout
5	Unspecified

Implementation

4.30 Configure PDP context dynamic parameter items to read(+NCPCDPR)

Command	Response	Example
+NCPCDPR=<parameter>,<state>	+CME ERROR:<err>	*AT+NCPCDPR=1,1* *OK*
+NCPCDPR	+NCPCDPR:<parameter>,<state> [+NCPCDPR:<parameter>,<state> [..]]	AT+NCPCDPR +NCPCDPR:0,1 +NCPCDPR:1,1 OK
+NCPCDPR=?	+NCPCDPR:(list of	

	<parameter>s),(0,1)	AT+NCPCDPR=? +NCPCDPR:(0,1) OK

Description This command allows the configuration of PDP context dynamic parameters request item.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<parameter> Unsigned char formate. Dynamic parameter to read

0 Get DNS Server IPv4 Address request.

1Get DNS Server IPv6 Address request.

<state> PDP context dynamic parameter items read state.

0 Disable to get

1 Enable to get

Implementation

- Could set only when power off.

4.31 Query pending originating data list via the control plane (+NQPODCP)

Command	Response	Example
+NQPODCP	[+NQPODCP:[<sequence>[<sequence>…]]] +CME ERROR:<err>	AT+NQPODCP +QPODCP:1,2,3 OK
+NQPODCP=?	OK	AT+NQPODCP=? OK

Description

This command queries the pending originating data list via the control plane by MT.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<sequence> The sequence of pending sending originating message. Range 1-255.

Implementation

4.32 Time update mode set (+NITZ)

Command	Response	Example
+NITZ=<mode>	+CME ERROR:<err>	*AT+ NITZ =1* *OK*
+**NITZ?**	+NITZ:<mode> +CME ERROR: <err>	AT+NITZ? +NITZ=1 OK
+NITZ=?	+NITZ:(list of supported <mode>s) +CME ERROR: <err>	AT+NITZ=? +NITZ:(0,1) OK

Description

Set time update mode.

If setting fails in an MT error, +CME ERROR: <err> is returned. Refer to Chapter 6 for possible <err> values.

Read command returns the current time update mode in the MT.

Test command returns supported time update modes.

Defined values

<mode>: integer type value indicating

0 Update time via local.
1 Update time via NITZ.

Implementation

4.33 Sending Non-ip Data (+NSNPD)

Command	Response	Example
+NSNPD=<cid>,<non_ip_data_length>,<non_ip_data>[,<RAI>[,<type_of_user_data>[,<sequence>]]]	+CME ERROR:<err>	AT+NSNPD=0,3,112233 *OK*
+NSNPD=?	+NSNPD:(rane of supported <cid>s),(maximum number of bytes of the < non_ip_data_length>),(list of supported <RAI>s),(list of supported <type_of_user_data>s),(list of supported <sequence>s) +NSNPD:(rane of supported	AT+NSNPD=? +NSNPD:(0-10),(1358),(0,1,2),(0,1),(1-255) OK

Description

The set command is used by the TE to transmit non-ip data over control plane or user plane to network via MT. Context identifier <cid> is used to link the data to particular context.

This command optionally indicates that the application on the MT expects that the exchange of data:

- will be completed with this uplink data transfer; or
- will be completed with the next received downlink data.

This command also optionally indicates whether or not the data to be transmitted is an exception data..

This command causes transmission of an ESM DATA TRANSPORT message, as defined in 3GPP TS 24.301 [83].

Test command returns the maximum number of bytes of the user data container supported by the MT, supported <RAI>s and supported <type_of_user_data>s as a compound value.

If <sequence>is not omitted, when datagram is acked by server or discard by UE, will report the result.

Command	Response	Example
	+NSNPD:<sequence>,<status>	+NSNPD:2,1

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<cid>: integer type. A numeric parameter which specifies a particular PDP context or EPS bearer context definition. The <cid> parameter is local to the TE-MT interface and identifies the PDP or EPS bearer contexts which have been setup via AT command (see the +CGDCONT and +CGDSCONT commands).

< non_ip_data_length>: integer type. Indicates the number of bytes of the <non_ip_data> information element. When there is no data to transmit, the value shall be set to zero.

<non_ip_data>: string of octets. Contains the user data container contents (refer 3GPP TS 24.301 [83]

subclause 9.9.4.24). When there is no data to transmit, the <non_ip_data> shall be an empty string (""). This parameter shall not be subject to conventional character conversion as per +CSCS. The coding format of the user data container and the maximum length of <non_ip_data> are implementation specific.

<RAI>: integer type. Indicates the value of the release assistance indication, refer 3GPP TS 24.301 [83] subclause 9.9.4.25.

0 No information available.
1 The MT expects that exchange of data will be completed with the transmission of the ESM DATA
TRANSPORT message.
2 The MT expects that exchange of data will be completed with the receipt of an ESM DATA TRANSPORT message.

<type_of_user_data>: integer type. Indicates whether the user data that is transmitted is regular or exceptional.

0 Regular data.
1 Exception data.

<sequence>: Sequence of data, range 1-255.If omit, will not report data sent status.

<status> The status of datagram.

0 Error
1 Sent

Implementation

- Maximum data length will be 1358 bytes when non-ip is used, otherwise will be 0.
- Only one message will be buffered at any one time.

4.34 Set Reporting of Non-ip Data Mode (+NRNPDM)

Command	Response	Example
+NRNPDM=<reporting>	+CME ERROR:<err>	AT+NRNPDM=1 OK
+NRNPDM?	+NRNPDM:<reporting>	AT+NRNPDM? +NRNPDM:1 OK
+NRNPDM=?	+NRNPDM:(list of supported <reporting>s),(range of supported <cid>s),(maximum number of octets of user data indicated by <non_ip_data_length>)	AT+NRNPDM=? +NRNPDM:(0-1),(0-10),(1358) OK

Description

The set command is used to enable and disable reporting of non-ip data from the network to the MT that is transmitted via the control plane or user plane in downlink direction. If reporting is enabled, the MT returns the unsolicited result code +NRNPDM:<cid>,<cpdata_length>,<cpdata> when data is received from the network.

Read command returns the current settings.

Test command returns supported values as compound values.

Refer to Chapter 6: *Error Values* for possible <err> values.

Command	Response	Example
	+NRNPDM:<cid>,<non_ip_data_length>,<non_ip_data>	+NRNPDM:0,2,"ab"

Defined values

<reporting>: integer type, controlling reporting of mobile terminated non-ip data events

0 Disable reporting of MT non-ip data.

1 Enable reporting of MT non-ip data by the unsolicited result code +NRNPDM.

<cid>: integer type. A numeric parameter which specifies a particular PDP context or EPS bearer context
definition. The <cid> parameter is local to the TE-MT interface and identifies the PDP or EPS bearer
contexts which have been setup via AT command (see the +CGDCONT and +CGDSCONT commands).

<non_ip_data_length>: integer type. Indicates the number of bytes of the <non_ip_data> information element. When there is no data to transmit, the value shall be set to zero.

<non_ip_data>: string of octets. Contains the user data container contents (refer 3GPP TS 24.301 [83] subclause 9.9.4.24). When there is no data to transmit, the <non_ip_data> shall be an empty string (""). This parameter shall not be subject to conventional character conversion as per +CSCS. The coding format of the user data container and the maximum length of <non_ip_data> are implementation specific.

Implementation

- Maximum data length will be 1358 bytes when non-ip is used, otherwise will be 0
- Only one message will be buffered at any one time
- Terminating data before enable reporting will be discarded

4.35 Query Pending Non-ip Data List (+NQPNPD)

Command	Response	Example
+NQPNPD	[+NQPNPD:[<sequence>[<sequence>…]]] +CME ERROR:<err>	AT+NQPNPD +NQPNPD:1,2,3 OK

+NQPNPD=?	OK	AT+NQPNPD=? OK

Description

This command queries the pending non-ip data list via the control plane or user plane by MT.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<sequence> The sequence of pending sending non-ip data message. Range 1-255.

Implementation

4.36 Get socket status(+NSOSTATUS)

Command	Response	Example
+NSOSTATUS=<socket id>	+NSOSTATUS:<socket id >[<status>,[backoff value]] *+CME ERROR:<err>*	AT+NSOSTATUS=1 +NSOSTATUS:1,3 OK
+NSOSTATUS	[+NSOSTATUS:<socket id >[<status>,[backoff value]][+NSOSTATUS:<socket id >[<status>,[backoff value]][...] +NSOSTATUS:<list of supported sockets>	AT+NSOSTATUS +NSOSTATUS:0,0 +NSOSTATUS:1,0 +NSOSTATUS:2,1 +NSOSTATUS:3,1 +NSOSTATUS:4,1 +NSOSTATUS:5,1 +NSOSTATUS:6,1 OK
+NSOSTATUS=?	OK	AT+NSOSTATUS=? +NSOSTATUS:(0-6) OK

Description

Get socket status.

The execution command returns a list of all supported socket status.

The test command returns a list of supported sockets.

Refer to Chapter 6: *Error Values* for possible <err> values

Defined values

<socket id> Socket number.

<status> integer type.

0 vailable

1 not exist. This error is also returned if the socket is not bound to the correct ip address NW assigned.

2 flow control

3 back off

<value> back off timer/T3396 timer remainder time(in seconds).Only show when status is back off.

Implementation

- Socket 0 is used for COAP, and will be created while booting up if COAP is supported

4.37 Get socket status(+NSOSTATUS)

Command	Response	Example
+NCIDSTATUS=<cid>	+NCIDSTATUS:<cid>[<status>,[backoff value]] *+CME ERROR:<err>*	AT+NCIDSTATUS=1 +NCIDSTATUS:1,3 OK
+NCIDSTATUS	[+NCIDSTATUS:<cid>[<status>,[backoff value]][+NSOSTATUS:<cid>[<status>,[backoff value]][...]	AT+NCIDSTATUS? +NCIDSTATUS:1,2 +NCIDSTATUS:2,2 OK

+NCIDSTATUS =?	+NCIDSTATUS:<list of supported cids> OK	AT+NCIDSTATUS=? +NCIDSTATUS:(0-10) OK

Description

Get pdp context status.
The execution command returns a list of all defined pdp context status.
The test command returns a list of supported cids.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<cid> integer type; specifies a particular PDP context definition (see the +CGDCONT command).

<status> integer type.

0 available
1 not exist. This status will also returned if PDP context is inactive.
2 flow control
3 back off

<value> back off timer/T3396 timer remainder time(in seconds).Only show when status is back off.

Implementation

4.38 PDP context activate or deactivate result report (+NGACTR)

Command	Response	Example
+NGACTR=<n>	*+CME ERROR:<err>*	AT+NGACTR=0 OK
+NGACTR?	+NGACTR:<n> *+CME ERROR:<err>*	AT+NGACTR? +NGACTR:0 OK
+NGACTR=?	+NGACTR:(list of supported <n>s) *+CME ERROR:<err>*	AT+NGACTR=? +NGACTR:(0,1) OK

Description

The set command controls the presentation of an unsolicited result code +NGACTR.
If <n>=1, +NGACTR:<n> is sent from the MT when the pdp context activate or deactivate complete. If setting
fails, an MT error, +CME ERROR:<err> is returned.
Refer to Chapter 6: *Error Values* for possible <err> values.
The read command returns the status of result code presentation

Test command returns supported values as a compound value.

Defined values

<n>: integer type

0 disable unsolicited result code
1 enable unsolicited result code +NGACTR:<cid>,<state>,<result>

<cid>: integer type; specifies a particular PDP context definition (see the +CGDCONT command).

<state>: integer type; indicates the state of PDP context activation. The default value is manufacturer specific.

0 deactivated
1 activated

<result>: integer type; indicates the result of PDP context activate or deactivate

0	success
1	fail
13	activate fail because service error

Implementation

4.39 Set UICC Power Mode (+NUICC)

Command	Response	Example
+NUICC=<mode>	*+CME ERROR:<err>*	AT+NUICC=1 OK
+NUICC=?	+NUICC:(list of supported <mode>s) +CME ERROR: <err>	AT+NUICC=? +NUICC:(0,1) OK

Description

Power on/off UICC. Used only when the configuration item NAS_SIM_POWER_SAVING_ENABLE of +NCONFIG is true, and is needed when want to use +CSIM and +CRSM.It increase the power consumption when power on. If setting fails in an MT error, +CME ERROR: <err> is returned. Refer to Chapter 6 for possible <err> values. Test command returns the supported modes.

Defined values

<mode>: integer type.

0	power off.
1	power on.

Implementation

4.40 Choose IoT Plaform (+CZIOTP)

Command	Response	Example
+CZIOTP?	+CZIOPT: <type> OK	AT+CZIOTP? +CZIOTP:0 OK
+CZIOTP=<type>	OK	AT+CZIOTP=1 OK

Description

This command set IoT platform to connect to. This configuration effect the behavior of the IoT commands in Chapter 6 and 7.

Defined values

<type>

0	China Telecom IoT platform
1	China Mobile IoT platform
2	China Unicom IoT platform

4.41 Read ADC Value (+CZADC)

Command	Response	Example
+CZADC?	+CZADC: <value> OK	AT+CZADC? +CZADC:621 OK
+CZADC	+CZADC: <value> OK	AT+CZADC +CZADC:621

```
OK
```

Description

This command Read the ADC value.

Defined values

`<value>`

The ADC value

5 Commands (Production and Configuration)

5.1 Execute a Radio Test Command (+NRDTEST)

Command	Response	Example
+NRDTEST=<len>,<data>	*<response length>,<response>* *+CME ERROR:<err>*	AT+NRDTEST=8,0000400213020000 OK
+NRDTEST=?		AT+NRDTEST=? OK

Description

This command will pass a binary encoded radio test command to the protocol core to be executed. It should only be used as instructed by HiSi.

Defined values

<length>	Decimal length of data to be sent in bytes.
<data>	Data to send in hexstring format.
<response length>	Length of response in bytes of binary data.
<response>	Data returned in hexstring format.

Implementation

- Couldn’t use combine AT command with this command.

5.2 RF Config Control Commands (+NRDCTRL)

Command	Response	Example
+NRDCTRL=<cmd>	+CME ERROR:<err>	AT+NRDCTRL=RESET OK
+NRDCTRL=?	+NRDCTRL:(list of supported commands)	AT+NRDCTRL=? +NRDCTRL:(ACTIVATE,DEACTIVATE,RESET) OK

Description

Radio configuration control operations

Refer to Chapter 6: *Error Values* for possible <err> values.

. Defined values

<command>

ACTIVATE	LL1 activate
DEACTIVATE	LL1 deactivate
RESET	Reset radio configuration to idle

Implementation

- AT+NRDCTRL=ACTIVATE will take about 1s to complete.
- Should send AT+NRDCTRL= ACTIVATE first to activate RF.

5.3 RF Config Set Command (+NRDSET)

Command	Response	Example
+NRDSET=<cmd>[,<arg1[,{arg2>,...]]	+CME ERROR:<err>	AT+NRDSET=RIO,1,500 OK
+NRDSET?	[+NRDSET:<cmd>,<val1>[,<val2> ...]]	AT+NRDSET? +NRDSET:PIO,0xe000080000 +NRDSET:GAIN,0,0,0,503

		OK
+NRDSET=?	+NRDSET:{list of supported commands)	AT+NRDSET=? +NRDSET:(RIO,PIO,TONE,GAIN) OK

Description

Allow various states to be set to support Radio calibration tests. The read variant will return the current values. Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<command>

RIO

<rio>,<mv>

PIO

Set specified RIO pin to specified voltage.

TONE

Set Radio PIOs for calibration tests

<frequency>

GAIN

<rio>

<mv>

<piomask>

<frequency>

<tx_drive_level>

<tx_coarse_gain>

<rf_gain>

Transmit a tone at a specific frequency for RF Transmit Calibration tests

<tx_drive_level>,<tx_coarse_gain>,<rf_gain>,<mv>

Set gain levels for RF TX configuration

select RIO to be used This is the integer number of the RIO.

voltage in mV. Integer value

bit mask specifying which PIOs are set. This is an ‘0x’ prefixed hexstring that specifies upto 64 PIOs. Only the PIOs supported by the platform, starting at bit 0, will be used. Leading ‘0’ characters can be omitted.

integer type indicating the frequency in Hz of the carrier to transmit (uint32 value)

Tx DAC drive level. Integer between 0 and 1023.

Coarse Analog Tx gain. Integer 0-3

RF gain setting. Integer 0-63.

Implementation

- AT+NRDSET? does not return values for RIO and TONE
- Should send AT+NRDCTRL= ACTIVATE first to activate RF.

5.4 RF Config Execute Command (+NRDEXEC)

Command	Response	Example

+NRDEXEC=<cmd>,<parameters>	[+NRDEXEC:<cmd>,<val1>[,<val2>...]] +CME ERROR:<err>	AT+NRDEXEC=RXRSSI,1500,3,0 +NRDEXEC:RXRSSI,-32768,-32768 OK
+NRDEXEC=?	+NRDEXEC:(list of supported commands)	AT+NRDEXEC=? +NRDEXEC:(RXRSSI,TXTEST,TXVERIFY,TXCALIST,RXCALIST,RXVERIFY) OK

Description

Allow users to verify their design calibration and compensation tables.

Refer to Chapter 6: *Error Values* for possible <err>

values **Defined values**

<cmd> Command to execute

RXRSSI Execute an RF Receive Calibration test

Input:

`<frequency>,<gain_index>,<port>`

Output:

`<rssi>,<snr>`

TXTEST Transmit a continuous test signal, either a tone (if num_sc is set to 0) or an

NB-IOT signal using the specified subcarrier(s). All internal gains are set to

zero, and front-end configuration is left in the idle state, ready to be set using

AT+NRDSET commands.

Input:

`<frequency>,<band>,<num_sc>[,<sc_index>]`

Output:

OK or ERROR

TXVERIFY Transmit a test signal, either a tone (if num_sc is set to 0) or an NB-IOT signal

using the specified subcarrier(s). The gains and front-end settings used are

read from the completed .cal file tables, based on the specified tx_power

and band, temperature, voltage.

Input:

`<frequency>,<band>,<temparature>,<voltage>,<tx_power>,`
`<tx_duration>,<num_sc>[,<sc_index>]`

Output:

OK or ERROR

RXVERIFY verify Rx design calibration and compensation tables

Input:

`<frequency>,<band>,<temperature>,<voltage>,<gain_index>`

Output:

`<rssi>.<snr>.<cbm>`

TXCALIST A sequence of single tones transmit with a fixed interval. The module can use the same set of Gain value to transmit single tones at different frequencies with a fixed interval currently 10ms. And also supports transmitting a set of power with the same frequency, but the power based on the loaded default tlv in the firmware that means we can sweep from -40dBm to 23dBm using the loaded tlv. When using TXCALIST for calibration, it is recommended to set <num_sc> to 1 and <sc_index> to 6.

Input:

<start_up_frequency>,<step_up_frequency>,<stop_up_freqecuency>,<start_up_power>,<step_up_power>,<stop_

up_power>,<num_sc>,<sc_index>

Output:

OK or ERROR

RXCALIST

To control the chip to receive the signal with a fixed interval and calculate
the value of RSSIs. The chip will send a TX trigger signal first, Tx trigger signal duration is 8ms, Then the chip waits for the meter to be ready, waiting for 4ms, then starts to receive different frequencies or level signals at a fixed interval. <rx_sweep_duration> to set the size of the fixed interval, can be set to a value greater than 5, recommended to set to 5 or 10, the unit of <rx_sweep_duration> is milliseconds

Input:

<start_dl_frequency>,<step_dl_frequency>,<stop_dl_frequency>,<start_agc_index>,<step_agc_index>,<stop_agc_inde>,<up_trigger_power>,<up_trigger_freq>,<rx_sweep_duration>,<sweep_band>

Output:

<data length>,<the number of RSSI data measured and RSSI data>

<frequency> Integer type indicating the frequency to test at (uint32 Hz value)

<gain_index> integer value 0-10, or 255 to let the chip's AGC select a gain setting.

<port> Integer value 0 or 1, corresponding to the RX port.

<band> EUTRA band number to use. Integer value 0-255

<temperature> signed 16-bit integer, either the temperature to use in the lookup table, or -999 to use the on-chip temperature sensor

<voltage> signed 16-bit integer, either the battery voltage to use in the lookup table, or -999 to use the on-chip voltage measurement. ,

<tx_power> TX power level in dBm. For TXTEST, the special value -99 tells the firmware not to read any power tables or FE timer tables from the calibration file, instead setting gains to zero ready to be modified by NRDSET commands
.

<num_sc> Number of subcarriers: 1, 3, 6, 12; or 0 to request a tone.

<sc_index> Position of subcarrier (first subcarrier position if transmitting multiple
carriers):

0
-
1
1 for num_sc = 1
0, 3, 6, 9 for num_sc = 3
0,
6 for num_sc = 6
0 for num_sc = 12
Ignored for num_sc = 0.

<tx_duration> Duration of transmission in reference clock cycles

<rssi> Gain of the receiver in centibels at the input of the DSP. Integer value.

<snr> Signal to noise ratio in dB. Fixed point (Q7.8) value.

<cbm> Received signal level in cBm.

<data length> Integer type

<the number of RSSI data measured and RSSI data> Hexstring format.
The first two bytes represent the number of RSSI data measured, the
remaining are RSSI data.
Need to pay attention to endian.
For example, if the data returned by RCCALIST is
"\r\n24,
0b00220228022d022f023a023602340277028902a802c5
02\r\n\r\nOK\r\
n", then the first two bytes are 0x0B00, convert the byte order, so the
actual value is 0x000b, which means there are 11 RSSI values returned.
Then every two bytes represent one RSSI value. E.g, the third and fourth
bytes are 0x2202, then actual value is 0x0222, indicate that the first RSSI
value returned is 546.

Implementation

- Should send AT+NRDCTRL= ACTIVATE first to activate RF

5.5 Read System Information (+NCHIPINFO)

Command	Response	Example
+NCHIPINFO=<cmd>	*<result>* *+CME ERROR:<err>*	AT+NCHIPINFO=ALL +NCHIPINFO:VBAT,1800 +NCHIPINFO:TEMP,25 OK
+NCHIPINFO=?		AT+NCHIPINFO=? +NTCHIPINFO:(ALL,VBAT,TEMP) OK

Description

Return system information, including temperature and battery voltage.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<cmd> Command to execute
ALL: return all data
TEMP: current temperature in degrees C.
VBAT: battery voltage in mV
<result> Integer value corresponding to specified command.

Implementation

5.6 Calibrate Temperature Sensor(+NCALTEMPSENSOR)

Command	Response	Example
+NCALTEMPSENSOR=<current temp>	*+CME ERROR:<err>*	AT+NCALTEMPSENSOR=32 OK
+NCALTEMPSENSOR=?		OK

Description

Calibrate temperature sensor by <current temp>.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<current temp> current chip temperature to Calibrate in degrees C.

Implementation

5.7 MIPI register read/write (+NRDMIPI)

Command	Response	Example
+NRDMIPI=<cmd>,<mipi slave address>,<mipi slave register data address>[,<mipi slave register data>]	[+NRDMIPI:READ,<mipi slave register data>]] *+CME ERROR:<err>*	[+NRDMIPI:READ,<mipi slave register data>]] *+CME ERROR:<err>* [+NRDMIPI:READ,<mipi slave register data>]]
+NRDMIPI=?	OK	AT+NQPODCP=? OK

Description

MIPI register read and write.

Refer to Chapter 6: *Error Values* for possible <err> values.

Defined values

<cmd> Command to execute. String type.

READ Read mipi register data.

Input:

<mipi slave address>,<mipi slave register data address>

WRITE Write mipi register data.

Input:

<mipi slave address>,<mipi slave register data address>,<mipi slave register data>

<mipi slave address>	integer type, range 0-255
<mipi slave register data address>	integer type, range 0-255
<mipi slave register data>	integer type, range 0-255

Implementation

6 Commands (CTCC IoT)

6.1 AT+NSECSWT Command

Command Syntax

Command	Command Output	Example
+NSECSWT=<type>[,<renegotiation time>]	OK +CME ERROR:<err>	no encryption: AT+NSECSWT=0 OK using standard DTLS: AT+NSECSWT=1,100 OK using optimized DTLS 时: AT+NSECSWT=2 OK
+NSECSWT?	+NSECSWT:<type>[,<renegotiation time>] OK +CME ERROR:<err>	AT+NSECSWT? +NSECSWT:0 OK

Command Description

This command is used to set the data encryption mode and renegotiation time for standard DTLS session timeout

Parameter Description

Parameter	Description
<type>	Indicates the encryption mode. 0: no encryption 1: encryption using standard DTLS 2: encryption using optimized DTLS The default value is 0. Data is not encrypted.
<renegotiation time>	Indicates the renegotiation time. The unit is minute. The value range is 1 to 525600. Only valid for standard DTLS.

Implementation Description

When <type> is set to 2, the renegotiation interval is fixed to one month.

The <renegotiation time> parameter is valid only when <type> is set to 1. The <renegotiation time> parameter must be set to a value less than the network's NAT change time.

After resetting DTLS type and renegotiation time, it will take effect after restart.

6.2 AT+NSETPSK Command

Command Syntax

Command	Command Output	Example
+NSETPSK=<pskid>,<psk>	OK +CME ERROR:<err>	AT+NSETPSK=201703230000024,0123456789ABCDEF0123456789ABCDEF OK
+NSETPSK?	+NSETPSK:<pskid>,<psk> OK +CME ERROR:<err>	AT+NSETPSK? +NSETPSK:201703230000024,*** OK

Command Description

This command is used to set PSKID and PSK.

Parameter Description

Parameter	Description
<pskid>	Indicates the PSK index. This parameter must be set to the IMEI (a 15-digit number) of the equipment. In addition, this parameter must also be set to the same value on the IoT platform.
<psk>	Indicates the PSK. This parameter must be set to a 32-digit hexadecimal number. In addition, this parameter must also be set to the same value on the IoT platform.

Implementation Description

When <pskid> is set to 0, the IMEI is automatically used as the PSKID.

After resetting PSK ID and PSK, it will take effect after restart.

6.3 AT+NCDP Command

Command Syntax

Command	Command Output	Example
+NCDP=<ip_addr>[,<port>]	OK +CME ERROR:<err>	AT+NCDP=192.168.5.1,5683 OK
+NCDP?	+NCDP:<ip_addr>,<port> OK +CME ERROR:<err>	AT+NCDP? +NCDP:192.168.5.1,5683 OK

Command Description

Set and query the IP address and port of the NB-IoT platform.

Parameter Description

Parameter	Description
<ip_addr>	IPv4 address in dot-separated numeric (0-255) parameter of the form: a1.a2.a3.a4
<port>	0~65535

Implementation Description

Only IPv4 is supported.

IMEI must be set prior to execute this command.

If no port is specified the previously set port will be used.

If no port is specified, and no port was previously set, the default port (5683) will be used.

If port 0 is provided, the default port (5683) will be used.

The changes will take effect after reboot.

6.4 AT+NNMI Command

Command Syntax

Command	Command Output	Example
+NNMI=<status>	OK +CME ERROR:<err>	AT+NNMI=1 OK
+NNMI?	+NNMI:<status> OK	AT+NNMI? +NNMI:1 OK

Command Description

This command sets or gets whether new message indications are sent. New message indications can be sent when a downstream message is received by the module from the IoT platform.

When neither new message indications nor messages are enabled (NNMI=0), no indications are sent.

When new message indications and messages are enabled (NNMI=1), all currently buffered messages will be returned.

Response	Example
+NNMI:<length>,<data>	+NNMI:5,48656C6C6F

If indications alone are turned on (NNMI=2), each newly received message triggers an indication that a new datagram is waiting using the unsolicited informational response. The buffered messages can be collected using AT+NMGR.

Response	Example
+NNMI	+NNMI

Parameter Description

Parameter	Description
<status>	0: No indications 1: Indications and Message 2: Indications only

Implementation Description

The default setting is 1.

The <status> will restore to the default value (1 by default) after reboot.

6.5 AT+NSMI Command

Command Syntax

Command	Command Output	Example
+NSMI=<indications>	OK +CME ERROR:<err>	AT+NSMI=1 OK
+NSMI?	+NSMI:<indications> OK	AT+NSMI? +NSMI:1 OK

Command Description

This command sets or gets whether indications are sent when an upstream message is sent to the IoT platform.

If sent message indications are turned off (NSMI=0), no indications are sent.

If sent message indications are turned on (NSMI=1), the following unsolicited informational response will be issued when a new message is sent to the IoT platform.

Response	Example
+NSMI:<status>[,<seq_num>]	+NSMI:SENT

The default setting is 0: no indications are sent.

Parameter Description

Parameter	Description
<indications>	0: No indications 1: Indications will be sent
<status>	SENT SENT_TO_AIR_INTERFACE DISCARDED
<seq_num>	1~255, Indicates the non-zero seq_num carried in the AT command to send data to IoT platform.

Implementation Description

Only when there is non-zero <seq_num> in the AT command to send data, the notification message sent by the module to the terminal contains <seq_num>, and the value of <seq_num> is the same as the value of <seq_num> carried by the AT command.

When there is non-zero <seq_num> in the AT command to send CoAP data and the CoAP data is sent to NB-IoT platform successfully, the value of <status> in notification message sent by the module to the terminal is SENT_TO_AIR_INTERFACE. When there is no <seq_num> or the value of the <seq_num> is zero in the AT command to send CoAP data and the CoAP data is sent to NB-IoT platform successfully, the value of <status> in notification message sent by the module to the terminal is SENT.

For CON data, SENT and SENT_TO_AIR_INTERFACE indicate the results of sending data to the NB-IoT platform.

For NON data, SENT indicates that data is sent to the C core. SENT_TO_AIR_INTERFACE indicates that data is sent to the air interface of the base station.

The <indications> will restore to the default value (0 by default) after reboot.

6.6 AT+NMGS Command

Command Syntax

Command	Command Output	Example
+NMGS=<length>,<data>[,<seq_num>]	OK +CME ERROR:<err>	AT+NMGS=3,AA11BB OK

Command Description

A terminal sends the data to the module by running this command. The module then sends the data to the IoT platform in compliance with the LWM2M protocol.

Parameter Description

Parameter	Description
<length>	Decimal length of message
<data>	Data to be transmitted in hexstring format
<seq_num>	0~255

Implementation Description
There is a maximum data length of 512 bytes, like AT+MLWULDATA.

If the NB module is not registered to NB-IoT platform, or the resource 19\0\0 hasn’t been observed, this command commands module to initiates registration only, and module discards the sent data.

If non-zero <seq_num> is used to send CoAP data and there is CON or NON CoAP data with the same

<seq_num> which hasn’t been sent completely, the data to be sent will be discarded and an error will be returned.

6.7 AT+NMGR Command

Command Syntax

Command	Command Output	Example
+NMGR	<length>,<data> OK +CME ERROR:<err>	AT+NMGR 5,48656C6C6F OK

Command Description
Receive a message from the IoT platform. This command (+NMGR) returns the oldest buffered message and deletes from the buffer. If there are no messages then no command response will be given. If new message indications are turned on (+NNMI=1) then received messages will not be available via this command.

Parameter Description

Parameter	Description
<length>	Decimal length of message
<data>	Data received in hexstring format

Implementation Description
Maximum received data length is 512 bytes.

6.8 AT+NQMGR Command

Command Syntax

Command	Command Output	Example
+NQMGR	BUFFERED=<buffered>, RECEIVED=<received>, DROPPED=<dropped> OK +CME ERROR:<err>	AT+NQMGR BUFFERED=0,RECEIVED=34,DROPPED=2 OK

Command Description
This command queries the status of the received downstream messages received from the IoT platform.

Parameter Description

Parameter	Description
<buffered>	The number of messages waiting to be read in the downstream buffer
<received>	The total number of messages received by the module since module boot
<dropped>	The number of messages dropped by the module since module boot

Implementation Description

6.9 AT+NQMGS Command

Command Syntax

Command	Command Output	Example
+NQMGS	PENDING=<pending>,SENT=<sent>,ERROR=<error> OK +CME ERROR:<err>	AT+NQMGS PENDING=1,SENT=34,ERROR=0 OK

Command Description

This command queries the status of the upstream messages sent to the IoT platform

Parameter Description

Parameter	Description
<pending>	The number of messages waiting to be sent
<sent>	The total number of uplink messages sent to the IoT platform since module boot
<error>	The number of messages that could not be sent by the module due to an error since module boot

Implementation Description

For CON CoAP data, the statistics are modified only after timeout, RST, sent_fail, or sent_success.

For NON CoAP data, if a non-zero seq_num is carried during data transmission, the statistics are modified only when the data is sent to the air interface. If the sent data does not carry seq_num or the seq_num is zero, the statistics are modified when the data is sent to the C core.

6.10 AT+NMSTATUS Command

Command Syntax

Command	Command Output	Example
AT+NMSTATUS?	+NMSTATUS:<registration_status> OK +CME ERROR:<err>	AT+NMSTATUS? +NMSTATUS:REGISTERED OK
AT+NMSTATUS=?	<list of supported<registration_status> s>	AT+NMSTATUS=? UNINITIALISED MISSING_CONFIG INIITIALISING INIITIALISED INIT_FAILED REGISTERING REGISTERED DEREGISTERED MO_DATA_ENABLED NO_UE_IP REJECTED_BY_SERVER TIMEOUT_AND_RETRYING REG_FAILED DEREG_FAILED

		OK

Command Description

Report the current registration status when connected to the IoT platform.

Parameter Description

Parameter	Description
<registration_status>	UNINITIALISED MISSING_CONFIG INIITIALISING INIITIALISED INIT_FAILED REGISTERING REGISTERED DEREGISTERED MO_DATA_ENABLED NO_UE_IP REJECTED_BY_SERVER TIMEOUT_AND_RETRYING REG_FAILED DEREG_FAILED

Implementation Description

When LWM2M is in the state of MO_DATA_ENABLED, the terminal can send data.

6.11 AT+MLWSREGIND Command

Command Syntax

Command	Response	Example
+MLWSREGIND =<type>	OK +CME ERROR: <err>	AT+MLWSREGIND=0 OK

Command Description

Controlling NB module initiate registration and cancellation to NB-IoT platform by using this command.

Parameter Description

Parameter	Description
<type>	0: initiating a registration message 1: initiating a deregistration message

Implementation Description

6.12 AT+MLWULDATA Command

Command Syntax

Command	Response	Example
+MLWULDATA =<length>,<data>[,<seq_num>]	OK +CME ERROR: <err>	AT+MLWULDATA =3,AA34BB

		OK

Command Description

A terminal sends the data to the module by running this command. The module then sends the data to the IoT platform in compliance with the LWM2M protocol.

Parameter Description

Parameter	Description
<length>	Decimal length of message.
<data>	Data to be transmitted in hexstring format.
<seq_num>	0~255

Implementation Description

There is a maximum data length of 512 bytes, like AT+NMGS.

If the NB module is not registered to NB-IoT platform, or the resource 19\0\0 hasn’t been observed, this command commands module to initiates registration only, and module discards the sent data.

If non-zero <seq_num> is used to send CoAP data and there is CON or NON CoAP data with the same <seq_num> which hasn’t been sent completely, the data to be sent will be discarded and an error will be returned.

6.13 AT+MLWULDATAEX Command

Command Syntax

Command	Response	Example
+MLWULDATAEX=<length>,<data>,<mode>[,<seq_num>]	OK +CME ERROR: <err>	AT+MLWULDATAEX =3,AA34BB,0x0001 OK

Command Description

A terminal sends the data to the module by running this command. The module then sends the CON or NON data with the identifier of RAI (Release Assistant Indication) to the IoT platform in compliance with the LWM2M protocol.

Parameter	Description
<length>	Decimal length of message.
<data>	Data to be transmitted in hexstring format.

<mode>	0x0000: Send NON message;
	0x0001: Send Non message and carry RELEASE RAI;
	0x0100: Send CON message;
	0x0101: Send CON message and carry RELEASE_AFTER_REPLY RAI.

<seq_num>	0~255

Parameter Description

Implementation Description

There is a maximum data length of 512 bytes.

If the NB module is not registered to NB-IoT platform, or the resource 19\0\0 hasn’t been observed, this command commands module to initiates registration only, and module discards the sent data.

If non-zero <seq_num> is used to send CoAP data and there is CON or NON CoAP data with the same <seq_num> which hasn’t been sent completely, the data to be sent will be discarded and an error will be returned.

6.14 AT+MLWULDATASTATUS Command

Command Syntax

Command	Response	Example
+MLWULDATASTATUS?	+MLWULDATASTATUS: <indications>[,<seq_num>]	AT+MLWULDATASTAT US? +MLWULDATASTATUS: 4 OK

Command Description

This command queries the status of the sending CON data to NB-IoT platform.

Parameter Description

Parameter	Description
0	have not sent
1	sent, waiting response of IoT platform.
2	sent failed
3	timeout
4	success
5	got reset message

Implementation Description

This command only queries the status of the CON data that has been sent.

When non-zero seq_num is used to send CON CoAP data, the response of this command is

+MLWULDATASTATUS: <indications>,<seq_num>; When there is no seq_num in the AT command to send CON CoAP data or the seq_num is zero, the response of this command is +MLWULDATASTATUS: <indications>.

6.15 AT+MLWFOTAIND Command

Command Syntax

Command	Response	Example
+MLWFOTAIND=<type>	OK +CME ERROR:<err>	AT+MLWFOTAIND=1 OK

Command Description

This command is used to set the FOTA upgrade mode to either of the following modes.

Automatic mode: FOTA-based download and upgrade are automatically completed without command instruction from the terminal.

Controlled mode: the terminal uses this command to control whether to download or upgrade firmware.

Parameter Description

Parameter	Description
<type>	0: The automatic FOTA upgrade mode is used. 1:The controlled FOTA upgrade mode is used. 2: Terminal notify module to start downloading the version file. 3: Terminal notify module to cancel version file download. 4: Terminal notify module to start update. 5: Terminal notify module to cancel update.

Implementation Description

In controlled mode, when AT+MLWEVTIND=6 (received “put package uri” message) or AT+MLWEVTIND=7 (received “update” message) has been received, the terminal uses this command to control whether to download or upgrade firmware.

6.16 AT+MLWEVTIND Command

Command Syntax

Command	Response	Example
	+MLWEVTIND =<type>	AT+MLWEVTIND=4

Command Description

If some events of the module LWM2M process need to be perceived by device, run this command to inform terminals. Device will process the events according to its own service process.

Parameter Description

Parameter	Description
<type>	0: registration completed 1: deregistration completed 2: registration update completed 3: 19 object subscription completed 4: bootstrap completed 5: 5/0/3 resource subscription observed 6: inform device of receiving “put package uri” message 7: inform device of receiving “update” message 9: 19 object subscription canceled

Implementation Description

In the following scenes, a) NB module registration completed, deregistration completed or registration update completed; b) 19 object has been observed or canceled; c) bootstrap completed; d) 5\0\3 resource has been observed, NB module uses this command to inform terminals.

In controlled mode, NB module notifies the terminal by AT+MLWEVTIND=<type> message when it receives the update package URL or the request of upgrade. Then terminal determines whether to download or upgrade firmware.

6.17 AT+MREGSWT Command

Command Syntax

Command	Command Output	Example

+MREGSWT=<type>	OK +CME ERROR: <err>	AT+MREGSWT =1 OK
+MREGSWT?	+ MREGSWT:<type>	AT+MREGSWT? +MREGSWT:1 OK

Command Description

This command is used to set automatic registration after the module rebooting. If the switch value is 1, the module would register automatically after rebooting and linking into network. And if the switch value is 0, the module would notify the message of “REGISTERNOTIFY” to terminal device after rebooting, then the terminal device should trigger registration by AT command.

Parameter Description

Parameter	Description
<type>	0: Non-automatic registration mode 1: Automatic mode (default value)

Implementation Description

If the automatic registration switch is not set, the default mode is automatic mode.

6.18 AT+MRESETDTLS Command

Command Syntax

Command	Command Output	Example
+MRESETDTLS	OK	AT+MRESETDTLS OK

Command Description

If DTLS has completed the handshake or re-negotiation, the DTLS state can be set to INIT state through this AT command, and the handshake process will run when the data is sent next time.

Implementation Description

If the current state of DTLS is shaking hands or re-negotiation, the command will returns success directly. The next data sent does not trigger a heavy handshake.

6.19 AT+MDTLSSTAT Command

Command Syntax

Command	Command Output	Example
+MDTLSSTAT?	OK +MDTLSSTAT: <type>	AT+MDTLSSTAT? OK +MDTLSSTAT:0

Parameter Description

Parameter	Description
<type>	0: DTLS negotiation or handshake completion

	1: Not start negotiation or handshake 2: in the negotiation or handshake state 3: negotiation or handshake fail

Implementation Description

This command is used to query the current link state of DTLS.

In every 8 seconds of sending uplink data, if DTLS is not negotiated, the DTLS state will not be able to query. It will take 8 seconds then can query the state.

If DTLS negotiate can be completed within 8 seconds, DTLS status can be queried after completion of negotiation.

If it is register by the module power on or manually register by the AT command, the registration process is triggered by LwM2M and does not block the AT command.

After DTLS negotiation or handshake completion, the module initiatively sends the +MDTLSSTAT:0/3 message to the terminal.

6.20 AT+MBOOTSTRAPHOLDOFF Command

Command Syntax

Command	Command Output	Example
+ MBOOTSTRAPHOLDOFF = <ClientHoldOffTime >	OK +CME ERROR: <err>	AT+ MBOOTSTRAPHOLDOFF=600 OK
+ MBOOTSTRAPHOLDOFF?	+ MBOOTSTRAPHOLDOFF:<ClientHoldOffTime>] OK +CME ERROR: <err>	AT+ MBOOTSTRAPHOLDOFF? + MBOOTSTRAPHOLDOFF: 600 OK

Command Description

Set and query ClientHoldOffTime, default 600 seconds. Recommend 600 seconds =<ClientHoldOffTime <3000 seconds;

if configured this time too large (>3000s), the terminal Bootstrap Sequence time will be delayed, then affecting the consumption of terminal access.

If the configuration is too small (<600s), the time of Factory Bootstrap and Server Initiated Bootstrap is too short, which will affect the success rate of Factory Bootstrap and Server Initiated bootstrap.so if the module needs to execute the Server Initiated bootstrap process (without configuring BootStrap Server and LWM2M server addresses), or the LWM2M server address has been configured, the ClientHoldOffTimer value is not recommended to be less than 600s.

Implementation Description

When the BootStrap Server address is configured, the ClientHoldOffTime can be set 0, and the module will go directly into the Client Initiated Bootstrap and shorten the Client Initiated Bootstrap time.

If the module does not configure the BootStrap Server address, AT+MBOOTSTRAPHOLDOFF=0 will returns ERROR.

If BS Server does not exist, the ClientHoldOffTimer can not be set to 0. If the ClientHoldOffTimer timer time is 0, the BS Server IP can not be erased.

6.21 AT+MBOOTSTRAPSERVERIP Command

Command Syntax

Command	Command Output	Example
+MBOOTSTRAPSERVERIP =<type>,<ip_addr>[,<port>]	+CME ERROR: <err> OK	AT+ MBOOTSTRAPSERVERIP =BS，192.168.5.1 OK AT+ MBOOTSTRAPSERVERIP =LWM2M，192.168.5.2 OK AT+ MBOOTSTRAPSERVERIP =DEL，192.168.5.2 OK AT+ MBOOTSTRAPSERVERIP =BS,www.baidu.com OK AT+ MBOOTSTRAPSERVERIP =LWM2M,www.baidu.com OK AT+ MBOOTSTRAPSERVERIP =DEL,www.baidu.com OK

Parameter Description

Parameter	Description
<type>	BOOTSTRAP Server IP type: LWM2M: IoT Server IP BS: Bootstrap Server IP DEL: Delete the IP information
<addr>	Domain name or IP address. IP address format: point to decimal format IPv4 address:a1.a2.a3.a4. The range of value is 0~255.
<port>	0~65535

Implementation Description

Support set and delete the IP address BOOTSTRAP and IOT platform.

The kv using in the command of AT+MBOOTSTRAPSERVERIP and AT+NCDP is the same.

AT+MBOOTSTRAPSERVERIP=DEL,IP,PORT will delete when the IP and PORT can matching at the same time.

AT+MBOOTSTRAPSERVERIP=DEL,IP: delete when IP and PORT when PORT can matching.

6.22 AT+MSETBSPSK Command

Command Syntax

Command	Command Output	Example
+MSETBSPSK=<pskid>,<psk>	OK +CME ERROR:<err>	AT+ MSETBSPSK =201703230000024,0123456789AB CDEF0123456789ABCDEF OK
+MSETBSPSK?	+ MSETBSPSK:<pskid>,<psk> OK +CME ERROR:<err>	AT+ MSETBSPSK? + MSETBSPSK:201703230000024,** * OK

Command Description

This command is used to configure PSK ID and PSK using to set DTLS connection with BS Server.

Parameter Description

Parameter	Description
<pskid>	Key index: for indexing PSK, fixed length of 15 bits decimal digits number, the value needs to be the same with the IMEI of device, and it needs to be consistent with the IoT platform.
<psk>	Preset key: fixed length of 16 bytes hexadecimal number, which need to be consistent with the IoT platform.

Implementation Description

when<pskid>=0， will set IMEI as the<pskid>.

Before set pskid, need set AT+CFUN=0 first.

6.23 AT+MBSSECSWT Command

Command Syntax

Command	Command Output	Example
+ MBSSECSWT =<type>[,<NAT type >]	OK +CME ERROR:<err>	DTLS Closed: AT+NSECSWT=0 OK Using standard DTLS: AT+NSECSWT=1,1 OK
+ MBSSECSWT?	+ MBSSECSWT:<type>[,<	AT+ MBSSECSWT?

NAT type >] OK +CME ERROR:<err>	+ MBSSECSWT:0 OK AT+ MBSSECSWT? + MBSSECSWT:1,1 OK

Command Description

This command is used to configure DTLS connection switches with BS Server and standard DTLS negotiation time. Start the timer when the DTLS negotiated at first time, restart the timer when received the DTLS message from BS Server. When the uplink message is sent and the timer has expired, it is necessary to trigger a re- negotiation.

Parameter Description

Parameter	Description

<type>	Encryption mode type:
	0: None encryption;
	1: Standard DTLS encrypt data;

<NAT type>	Standard DTLS NATnetwork: 0: The DTLS link is rebuilt when the data send every time; 1: Only after the power on and module IP address changes can trigger the DTLS link;

6.24 AT+MDNS Command

Command Syntax

Command	Response	Example
+MDNS=<mode>[,hostname]	OK +CME ERROR: <err>	AT+ MDNS =0,www.baidu.com OK
Display the result of domain name resolution	+MDNS:result	+MDNS:10.10.10.1 +MDNS:2000::1:2345:6789:abcd +MDNS:FAIL

Command Description

This command is used to trigger the DNS domain name resolution by the AT command.

Parameter Description

Parameter	Description
<mode>	0: DNS domain name resolution, hostname can not be NULL 1: If hostname is not NULL, then clear the hostname parsing data in memory and KV; if hostname is empty, all parsing data will be cleared.
[hostname]	Domain need to be parsed

6.25 AT+ MLWM2MENABLE Command

Command Syntax

Command	Response	Example
+MLWM2MENABLE=<state>	OK +CME ERRORs: <err>	AT+ MLWM2MENABLE =1 OK
+MLWM2MENABLE?	+ MLWM2MENABLE:ressult	+ MLWM2MENABLE:1 OK

Command Description

This command is used to control LWM2M model enable or disable.

Parameter Description

Parameter	Description
< state >	0: LWM2M model is disenable 1: LWM2M model is enable

Implementation Description

The command can be used after reboot the module.

6.26 AT+MCRITICALDATA Command

Command Syntax

Command	Response	Example
+MCRITICALDATA =<state>	OK ERROR	AT+ MCRITICALDATA =1 OK

Command Description

In the process of FOTA upgrade, if there is an emergency data send request, we can send this AT first, if the response to OK, emergency data can be sent, if the response to ERROR, emergency data can not be sent.

Parameter Description

Parameter	Description
< state >	1: The value can only be set to 1, to query whether the emergency data can be sent

Implementation Description

This command is used to determine whether emergency data can be sent in the FOTA process, and if so, stop the current FOTA process and give priority to send data. The platform upgrade results rely on timeout processing.

7 Commands (CMCC IoT)

7.1 Create OneNET Instance (+MIPLCREATE)

Command	Response	Example
+MIPLCREATE	<ref> OK +CME ERROR:<err>	AT+MIPLCREATE 0 OK

Description

This command create an instance of communication to OneNET platform.

Defined values

Implementation

Now only support <ref>=0

7.2 Delete OneNET Instance(+MIPLDELETE)

Command	Response	Example
+MIPLDELETE=<ref>	OK +CME ERROR:<err>	AT+MIPLDELETE=0 OK
+MIPLDELETE=?	+MIPLDELETE:(list of supportedcommands)	AT+MIPLDELETE=? +MIPLDELETE:(0) OK

Description

Delete a specified OneNET communication instance.

Defined values

<ref> reference ID of OneNET communication instance.

Implementation

7.3 Send OneNET Login Request (+MIPLOPEN)

Command	Response	Example
+MIPLOPEN=<ref>,<life time>[,<timeout>]	+CME ERROR:<err>	AT+MIPLOPEN=0,3600,30 OK
+MIPLOPEN?	+MIPLOPEN:(list of supported commands)	AT+MIPLOPEN=? +MIPLOPEN:(0),(1-0xFFFFFFFF),(1-65535) OK

Description

Sending login request to OneNET with set command, and querying login state with read command.

Defined values

<ref> reference ID of OneNET communication instance.

<lifetime> lifetime unit(s).

<timeout> login timeout

<conState> 1 login,0 not login

Implementation

Set command is an asynchronous command, and returning OK means sending request successfully.

If successful login in time, the UE returns:

+MIPLEVENT:0,1

+MIPLEVENT:0,2

+MIPLEVENT:0,4

+MIPLEVENT:0,6

7.4 Send OneNET Logout Request(+MIPLCLOSE)

Command	Response	Example
+MIPLCLOSE=<ref>	+CME ERROR:<err>	AT+MIPLCLOSE=0 OK

Description

Sending Logout request to OneNET according to specified reference ID.

Defined values

<ref> reference ID of OneNET communication instance

Implementation

7.5 Add LWM2M Object (+MIPLADDOBJ)

Command	Response	Example
+MIPLADDOBJ=<ref>,<objectid>,<instancecount>,<instancebitmap>,<attributecount>,<actioncount>	+CME ERROR:<err>	AT+MIPLADDOBJ=0,3200,1,"1",10,10 OK
+MIPLADDOBJ=?	+MIPLADDOBJ:(list of supported commands)	AT+MIPLADDOBJ=? +MIPLADDOBJ:(0),(0-32767),(1-128),"",(0-128),(0-128) OK

Description

Add a lwm2m object to a specified OneNET instance. Concepts and definitions of Object, instance and resource, please refer to Lightweight Machine to Machine Technical Specification, ext-label Objects Produced by IPSO Alliance and oma-label Objects Produced by OMA.

http://www.openmobilealliance.org/wp/OMNA/LwM2M/LwM2MRegistry.html

Defined values

<ref> reference ID of OneNET communication instance

<objectid> object identity.(refer to documentation above)

<instancecount> how manny instance num. (1-128)

<instancebitmap> instance map,like "1011", means instance 0、1、3 be define

<attributecount> attributecount num

<actioncount> actioncount num

Implementation

7.6 Delete LWM2M Object(+MIPLDELOBJ)

Command	Response	Example
+MIPLDELOBJ=<ref>,<objectid>	+CME ERROR:<err>	AT+MIPLDELOBJ=0,3200 OK

Description

Delete an object from a specified OneNET instance. If uploading a deleted object, an error occurs.

Defined values

<ref> reference ID of OneNET communication instance

<objectid> object identity

Implementation

7.7 Notify Value Changes (+MIPLNOTIFY)

Command	Response	Example
+MIPLNOTIFY=<ref>,<msgid>,<objectid>,<instanceid>,<resourceid>,<valuetype>,<len><value>,<index>,<flag>[,<ackid>]	+CME ERROR:<err>	AT+MIPLNOTIFY=0,27,3200,0,5500,5,1,"1",0,0,100 OK
+MIPLNOTIFY=?	+MIPLNOTIFY:(list of supported commands)	AT+MIPLNOTIFY=? +MIPLNOTIFY:(0),(0-65535),(0-32767),(0-32767),(0-32767),(1-5),(1-500),"",(0-10), (0-10)[,(1-65535)]

		OK

Description

Notify OneNET that specified values changed.

Defined values

<ref> reference ID of OneNET communication instance

<msgid>message identity from+OBSERVE.

<objectid> object identity

<instanceid> instance identity

<resourceid> resource identity

<valuetype> data type

1	2	3	4	5
string	opaque	integer	float	bool

Opaque type are shown as character array on OneNET

If there are some special control characters in string, like 0x0D, 0x0A, 0x1B etc, must be converted to hexadecimal string and use hex_str type to avoid AT-parsing errors.

<len>The length of the data, length should less than 500 Bytes.

<value> values of specified value type, value length should less than 500 Bytes.

<index>instruction serial number. If a Notify operation requires N message combinations to be a complete instruction, index is descending from n-1 to 0. When index number is 0, the Notify instruction is finished

<flag>: message id

1. First message

2: intermediate information

0: last message

flag to indicate whether it' s the end of data, if set to 0, all the notified values of specified object & instance will upload to servers, else indicates further data expected.

<ackid>

If <ackid> omitted, no ACK will response, set to 0 causes errors.

Implementation

The data type sent is as follows:

String =1, // string type.

Opaque =2, // opaque type

Integer =3, // integer type, <len> is the number of bytes used for the integer(2 or 4 or 8 bytes).

Float =4, // floating type, <len> is the number of bytes used for the float (4 or 8 bytes)

Bool =5,// Bool type, 0 for false, 1 for true.

Examples are:

String: AT+MIPLNOTIFY=0,58417,3200,0,5750,1,4,"abcd",0,0,102

Opaque: AT+MIPLNOTIFY=0,12448,3200,0,5505,2,2,0132,0,0,102

Int: AT+MIPLNOTIFY=0,12448,3200,0,5501,3,2,655,0,0,102

Float: AT+MIPLNOTIFY=0,13728,3202,0,5600,4,4,100.111111,0,0,103

Bool: AT+MIPLNOTIFY=0,12448,3200,0,5500,5,1,0,0,0,104

7.8 Upload Read Messages (+MIPLREADRSP)

Command	Response	Example
+MIPLREADRSP=<ref>,<msgid>,<result>[,<objid>,<insid>,<resid>,<type>,<len>,<value>,<index>,<flag>]	+CME ERROR:<err>	AT+MIPLREADRSP=0,100,1,3200,0,5500,5,1,"1",0,0 OK
+MIPLREADRSP=?	+MIPLREADRSP:(list of supported commands)	+MIPLREADRSP:(0),(0-65535),(0-255)[,(0-32767),(0-32767),(0-32767),(1-5), (1-500),"",(0-10),(0-10)] OK

Description

The read command set specified resource values, and when flag set to 1, upload these updating values to OneNET. This should be operated when +MIPLREAD URC is received as reply of remote read command.

Defined values

<ref> reference ID of OneNET communication instance

<msgid> message identity from +MIPLREAD

<result> The result of the < result > : read operations; Can the return code is as follows:

1	2.05 Content	Read operations done correctly
11	4.00 Bad Request	
12	4.01 Unauthorized	
13	4.04 Not Found	
14	4.05 Method Not Allowed	
15	4.06 Not Acceptable	

<objectid> object identity

<instanceid> instance identity

<resourceid> resource identity

<valuetype> data type of value, refer to +MIPLNOTIFY command

<len>The length of the data, length should less than 500 Bytes.

<value> value

<index>instruction serial number. If a Notify operation requires N message combinations to be a complete instruction, index is descending from n-1 to 0. When index number is 0, the Notify instruction is finished

<flag>: message id

1. First message

2: intermediate information

0: last message

flag to indicate whether it’s the end of data, if set to 0, all the notified values of specified object & instance will upload to servers, else indicates further data expected.

Implementation

The data type sent is as follows:

String =1, // string type.

Opaque =2, // opaque type

Integer =3, // integer type, <len> is the number of bytes used for the integer(2 or 4 or 8 bytes).

Float =4, // floating type, <len> is the number of bytes used for the float (4 or 8 bytes)

Bool =5,// Bool type, 0 for false, 1 for true.

If the result is not 1, the < objectid >, < instanceid >, < resourceid >, < valuetype >, < len >, < value >, < the index >, < flag > parameters are omitted. End result is not to represent 1 reads the message which, if a resource of read request message communications suite has informed the basis, the success will only read news notice to basic communications suite, and by the underlying communication report suite to the end of the network and the reading process; If there are no read the notice before the wrong result to any data, the report read error to the end of the platform and the read operation. All resources, while reading error MCU only need to return an error result to communications suite, based communications suite will be reported to read error back to the platform.

7.9 Upload Write Result (+MIPLWRITERSP)

Command	Response	Example
+MIPLWRITERSP=<ref>,<msgid >,<result>	+CME ERROR:<err>	AT+MIPLWRITERSP=0,321,2 OK
+MIPLWRITERSP=?	+MIPLWRITERSP:(list of supported commands)	+MIPLWRITERSP:(0),(0-65535),(0-255) OK

Description

This command used as reply of remote write command after +MIPLWRITE URC received, to feedback the results of updating specified resource value.

Defined values

<ref> reference ID of OneNET communication instance
<msgid> message identity in +MIPLWRIT.
<result> The result of the < result > : read operations; Can the return code is as follows:
2 2.04 Changed write operations done correctly
11 4.00 Bad Request
12 4.01 Unauthorized
13 4.04 Not Found
14 4.05 Method Not Allowed
15 4.06 Not Acceptable
16 2.31 Continue
17 4.08 Request Entity Incomplete
18 4.13 Request entity too large
19 4.15 Unsupported content format

Implementation

7.10 Upload Execute Result(+MIPLEXECUTERSP)

Command	Response	Example
+MIPLEXECUTERSP=<ref>,<msgid>,<result>	+CME ERROR:<err>	AT+ MIPLEXECUTERSP=0,321,2 OK
+ MIPLEXECUTERSP=?	+ MIPLEXECUTERSP:(list of supported commands)	+MIPLEXECUTERSP:(0),(0-65535),(0-255) OK

Description

This command used as reply of remote execute command after +MIPLEXECUTE URC received to feedback the results of user-defined operation.

Defined values

<ref> reference ID of OneNET communication instance
<msgid> message identity in +MIPLEXECUTE message
<result> The result of the < result > : read operations; Can the return code is as follows:
2 2.04 Changed execute operations done correctly
11 4.00 Bad Request
12 4.01 Unauthorized
13 4.04 Not Found
14 4.05 Method Not Allowed

Implementation

7.11 Upload Observe Result (+MIPLOBSERVERSP)

Command	Response	Example
+MIPLOBSERVERSP=<ref>,<msgid>,<result>	+CME ERROR:<err>	AT+MIPLOBSERVERSP=0,100,1 OK
+MIPLOBSERVERSP=?	+MIPLOBSERVERSP:(list of supported commands)	+MIPLOBSERVERSP:(0)(0-65535),(0-255) OK

Description

This command used as reply of remote execute command after +MIPLOBSERVE URC received to feedback the results of user-defined operation.

Defined values

<ref> reference ID of OneNET communication instance
<msgid> message identity in +MIPLOBSERVE message
<result> The result of the < result > : read operations; Can the return code is as follows:
1 2.05 Context observe operations done correctly

11 4.00 Bad Request
12 4.01 Unauthorized
13 4.04 Not Found
14 4.05 Method Not Allowed

Implementation

7.12 Upload Discover Messages(+MIPLDISCOVERRSP)

Command	Response	Example
+MIPLDISCOVERRSP=<ref>,<msg id>,<result>,<length>,<valuestring>	+CME ERROR:<err>	AT+MIPLDISCOVERRSP=0,100,1,14,"5500;5501;5750" OK
+ MIPLDISCOVERRSP=?	+MIPLDISCOVERRSP:(list of supported commands)	+MIPLDISCOVERRSP:(0),(0-65535),(0-255),(1-500),"" OK

Description

This command used as reply of remote execute command after +MIPLDISCOVER URC received to feedback the results of user-defined operation.

Defined values

<ref> reference ID of OneNET communication instance
<msgid> message identity in +MIPLDISCOVER message
<result> The result of the < result > : read operations; Can the return code is as follows:
1 2.05 Context discover operations done correctly
11 4.00 Bad Request
12 4.01 Unauthorized
13 4.04 Not Found
14 4.05 Method Not Allowed
15 4.06 Not Acceptable
<length>valuestring data length
<valuestring>Object attribute requirements, with a “;” between multiple attributes, separated ″1101;1102;1103″.

Implementation

7.13 Upload PARAMETER Result (+MIPLPARAMETERRSP)

Command	Response	Example
+MIPLPARAMETERRSP=<ref>,<msgid>,<result>	+CME ERROR:<err>	AT+MIPLPARAMETERRSP =0,100,2 OK

Description

This command used as reply of remote execute command after +MIPLPARAMETER URC received to feedback the results of user-defined operation.

Defined values

<ref> reference ID of OneNET communication instance
<msgid> message identity in +MIPLPARAMETER message
<result> The result of the < result > : read operations; Can the return code is as follows:
2 2.04 Changed parameter operations done correctly
11 4.00 Bad Request
12 4.01 Unauthorized
13 4.04 Not Found
14 4.05 Method Not Allowed

Implementation

7.14 Unsolicited Read Request (+MIPLREAD) (Response Only)

Command	Response	Example
+MIPLREAD:<ref>,<msgid>,<objid>,<insid>,<resid>		+MIPLREAD:0,289,3200,0,-1

Description

Unsolicited result code of remote reading command. It represents that server request device to upload their specified local resource values.

Defined values

<ref> reference ID of OneNET communication instance

<msgid> identity of this message

<objectid> object identity

<instanceid> instance identity, -1 means read all the instances belong to the object, else only read specified instance.

<resourceid> resource identity, -1 means read all the resources belong to the instance, else only read specified resource.

Implementation

There are 3 kinds of read operations:

1. Read specified object/instance/resource
2. Read all resources of specified object/instance
3. Read all resources of all instances of specified object

Reading result should be replied to server using +MIPLREADRSP command,

The MCU shall be within seconds(3s) of the specified operation of the platform

7.15 Unsolicited Write Message (+MIPLWRITE) (Response Only)

Command	Response	Example
+MIPLWRITE:<ref>,<msgid>,<objid>,<insid>,<resid>,<valuetype>,<len>, <value>,<flag>,<index>		+MIPLWRITE:0,2158,3200,0,5750,2,3,616263,0,0

Description

Unsolicited result code of remote writing command. It represents that server request device to modify their specified local resource values.

Defined values

<ref> reference ID of OneNET communication instance

<msgid> identity of message

<objectid> object identity

<instanceid> instance identity

<resourceid> resource identity

<valuetype> data type of value

<len>The length of the data, length should less than 500 Bytes.

<value> value

<flag>: message id

1. First message

2: intermediate information

0: last message

flag to indicate whether it' s the end of data, if set to 0, all the notified values of specified object & instance will upload to servers, else indicates further data expected.

<index>instruction serial number.

If a Notify operation requires N message combinations to be a complete instruction, index is descending from n-1 to 0. When index number is 0, the Notify instruction is finished

Implementation
All the data type of value is opaque, value is hexstring,The user depending on the type of data
Executing result should be replied to server using +MIPLWRITERSP command.
resources to parsing

7.16 Unsolicited Execute Message (+MIPLEXECUTE) (Response Only)

Command	Response	Example
+MIPLEXECUTE:<ref>,<msgid>,<objid>,<insid>,<resid>[,<len>,<cmd>]		+MIPLEXECUTE:0,291,3200,0,5500,4,ping

Description
Unsolicited result code of remote executing command. It represents that server request device to execute some pre-defined operations on specified resource.
Defined values
<ref> reference ID of OneNET communication instance
<msgid> identity of message
<objectid> object identity
<instanceid> instance identity
<resourceid> resource identity
<len> length of cmd
<cmd> command that remote server request the resource to execute
Implementation
Executing result should be replied to server using +MIPLEXECUTERSP command.

7.17 Unsolicited Observe Message (+MIPLOBSERVE) (Response Only)

Command	Response	Example
+MIPLOBSERVE:<ref>,<msgid>,<flag>,<objectid>,<instanceid>,<resourceid>		+MIPLOBSERVE:0,2657,1,3200,0,-1

Description
Unsolicited result code of observe message from server. It represents that server request device to upload the values of specified resource, or all resources of specified instance when they change.
Defined values
<ref> reference ID of OneNET communication instance
<msgid> identity of message
<flag> 1 add observe,0 cancel observe
<objectid> object identity
<instanceid> instance identity, -1 means observing all the instances and resources belong to the object, else only observing specified instance.
<resourceid>resourceidentity,-1 means observing all the resources belong to the instances.
Implementation
Executing result should be replied to server using +MIPLOBSERVERSP command.

7.18 Unsolicited Observe Message (+MIPLDISCOVER) (Response Only)

Command	Response	Example
+MIPLDISCOVER:<ref>,<msgid>,<objectid>		+MIPLDISCOVER:0,2657,3200

Description

Unsolicited result code of discover message from server.it notifies the MCU obtain the property of the specified object

Defined values

<ref> reference ID of OneNET communication instance

<msgid> identity of message

<objectid> object identity

Implementation

Executing result should be replied to server using +MIPLDISCOVERRSP command.

7.19 Unsolicited Parameter (+MIPLPARAMETER) (Response Only)

Command	Response	Example
+MIPLPARAMETER:<ref>,<msgid>,<objectid>,<instanceid>,<resourceid>,<len>,<parameter>		+MIPLPARAMETER:0,44256,3200,0,5500,39,pmin=1;pmax=100;lt=10.0;gt=100.0;st=5.1

Description

Unsolicited result code of observe parameter message from server, like interval, threshold value of specified resource etc.

Defined values

<ref> reference ID of OneNET communication instance

<msgid> identity of message

<objectid> object identity

<instanceid> resource identity, -1 means the parameter are valid in all the instances and resources belong to the object

<resourceid> resource identity, -1 means the parameter are valid in all the instances belong to the resource

<len> length of parameter string

<parameter> string type, like:
pmin=xxx;pmax=xxx;gt=xxx;lt=xxx;stp=xxx

Implementation

Executing result should be replied to server using +MIPLPARAMETERRSP command.

7.20 Read Version (+MIPLVER)

Command	Response	Example
+MIPLVER?	<version>	AT+MIPLVER? +MIPLVER:1.0.0 OK

Description

Get version of OneNET protocol.

Defined values

None

Implementation

7.21 Update registration information(+MIPLUPDATE)

Command	Response	Example
+MIPLUPDATE=<ref>,<lifetime>,<withObjectFlag>	+CMEERROR:<err>	AT+ MIPLUPDATE=0,100,1 OK
+ MIPLUPDATE=?	+MIPLUPDATE:(list of supported commands)	+MIPLUPDATE:(0),(0-0xFFFFFFFF),(0,1) OK

Description

This command used as Update registration information.

Defined values

<ref> reference ID of OneNET communication instance

<lifetime> device lifetime (s),if set 0,use +MIPLOPEN set.

<withObjectFlag> Whether you need to update the registered Object Object at the same time

Implementation

7.22 Unsolicited Event (+MIPLEVENT)

Command	Response	Example
+MIPLEVENT:<ref>,<eventid>[,<ackid>]		+MIPLEVENT:0,26,103

Description

Unsolicited result code of report a state event.

Defined values

<ref> reference ID of OneNET communication instance

<eventid>state value:

<ackid>When the return is CIS_EVENT_NOTIFY_SUCCESS, carrying issued corresponding ackid Notify.

```
#define CIS_EVENT_BASE  0
#define CIS_EVENT_BOOTSTRAP_START              CIS_EVENT_BASE + 1
#define CIS_EVENT_BOOTSTRAP_SUCCESS            CIS_EVENT_BASE + 2
#define CIS_EVENT_BOOTSTRAP_FAILED                    CIS_EVENT_BASE + 3
#define CIS_EVENT_CONNECT_SUCCESS              CIS_EVENT_BASE + 4
#define CIS_EVENT_CONNECT_FAILED               CIS_EVENT_BASE + 5
#define CIS_EVENT_REG_SUCCESS                         CIS_EVENT_BASE + 6
#define CIS_EVENT_REG_FAILED                                 CIS_EVENT_BASE + 7
#define CIS_EVENT_REG_TIMEOUT                         CIS_EVENT_BASE + 8
#define CIS_EVENT_LIFETIME_TIMEOUT                    CIS_EVENT_BASE + 9
#define CIS_EVENT_STATUS_HALT                         CIS_EVENT_BASE + 10
#define CIS_EVENT_UPDATE_SUCCESS               CIS_EVENT_BASE + 11
#define CIS_EVENT_UPDATE_FAILED                       CIS_EVENT_BASE + 12
#define CIS_EVENT_UPDATE_TIMEOUT               CIS_EVENT_BASE + 13
#define CIS_EVENT_RESPONSE_FAILED              CIS_EVENT_BASE + 20
#define CIS_EVENT_NOTIFY_FAILED                       CIS_EVENT_BASE + 21
#define CIS_EVENT_NOTIFY_SUCCESS               CIS_EVENT_BASE + 26

#define    CIS_EVENT_FIRMWARE_DOWNLOADING          CIS_EVENT_BASE + 40
#define    CIS_EVENT_FIRMWARE_DOWNLOAD_FAILED      CIS_EVENT_BASE + 41
#define    CIS_EVENT_FIRMWARE_DOWNLOADED           CIS_EVENT_BASE + 42
#define    CIS_EVENT_FIRMWARE_UPDATING             CIS_EVENT_BASE + 43
#define    CIS_EVENT_FIRMWARE_UPDATE_SUCCESS       CIS_EVENT_BASE + 44
#define    CIS_EVENT_FIRMWARE_UPDATE_FAILED        CIS_EVENT_BASE + 45
```

```
#define     CIS_EVENT_FIRMWARE_UPDATE_OVER          CIS_EVENT_BASE + 46
#define     CIS_EVENT_FIRMWARE_DOWNLOAD_DISCONNECT   CIS_EVENT_BASE + 47
#define     CIS_EVENT_FIRMWARE_ERASE_SUCCESS        CIS_EVENT_BASE + 48
#define     CIS_EVENT_FIRMWARE_ERASE_FAIL           CIS_EVENT_BASE + 49
```

Implementation

<msgid>: CIS_EVENT_RESPONSE_FAILED and CIS_EVENT_RESPONSE_FAILED events will carry the msgid of the corresponding command.

8 Error Values

The error codes listed below are enabled when CMEE is set to mode 1.

8.1 Overview

Error codes are aligned to the 3GPP spec. Refer to 3GPP TS 27.007 V13.5.0, sub-clause 9.2 for all possible <err> values. The error codes listed are those returned for the Hi2115 implementation.

Error codes 0-255 are reserved and defined in 3GPP TS 27.007 and may be used by Hisi in future releases.

8.2 General Errors (27.007)

Error Code	Error Text	Error Code	Error Text	Error Code	Error Text
3	Operation not allowed	4	Operation not supported	5	Need to enter PIN
23	Memory failure	30	No Network Service	50	Incorrect parameters
51	Command implemented but currently disabled	52	Command aborted by user	100	Unknown
159	Uplink Busy/Flow Control				

8.3 General Errors (27.005)

Unused error codes will be removed.

Error Code	Error Text	Error Code	Error Text	Error Code	Error Text
300	ME failure	312	PH-(U)SIM PIN required	321	invalid memory index
301	SMS service of ME reserved	313	(U)SIM failure	322	memory full
302	operation not allowed	314	(U)SIM busy	330	SMSC address unknown
303	operation not supported	315	(U)SIM wrong	331	no network service
304	invalid PDU mode parameter	316	(U)SIM PUK required	332	network timeout
305	invalid text mode parameter	317	(U)SIM PIN2 required	340	no +CNMA acknowledgement expected
310	(U)SIM not inserted	318	(U)SIM PUK2 required	311	(U)SIM PIN required
320	memory failure	500	unknown error		

8.4 specific error codes

Hisi specific error codes are in the range 512 onwards.

Error code 512 was previously error code 256. Error code 513 was previously error code 257.

Error Code	Error Text	Error Code	Error Text	Error Code	Error Text
512	Required parameter not configured	513	TUP not registered	514	AT Internal Error

515	CID is active	516	Incorrect State for Command	517	Cid is invalid
518	CID is not active	520	Deactive last active cid	521	Cid is not defined
522	Uart parity error	523	Uart frame error	524	Mt not power on
525	sent data sequence repeat error	526	at command abort error	527	command interrupted
528	configuration conflicts	529	FOTA is updating	530	Not the at allocated socket
531	SIM PIN is blocked	532	SIM PUK is blocked	533	File not found
535	conditions of use not satisfied	536	at uart buffer error	537	Back off timer is running

Cmcc IoT Specific error codes are in the range 600 onwards.

Error Code	Error Text	Error Code	Error Text	Error Code	Error Text
601	Parameter error	602	state error	651	init Error
652	not support error	653	in progress	654	progress error
655	no register network	656	data package too large	100	unknown error

8.5 Possible Error Causes

Error Code	Error Text	Possible Causes
512	Required parameter not configured	For AT+NCDP, if the IMEI is not set, the command will fail. AT+NTSETID=1 can be used to set IMEI.
513	TUP not registered	IMEI needs to be set. AT+NTSETID=1 can be used to set IMEI. The CDP IP address has not been set. AT+NCDP=<ip addr> to set AT+NMSTATUS to query TUP status
515	CID is active	Disable CID prior to changing CID configuration.
516	Incorrect State for Command	Some commands have to be executed in a specific order. Refer to the documentation for the specific command.
517	Cid is invalid	Beyond the accepted range of cid values for +CSODCP set.
518	CID is not active	Deactivate the pdp context which is not active.
520	deactivate last active cid	For +CGACT, the protocol say: if an attempt is made to disconnect the last PDN connection, then the MT responds with error. Return this error code when deactivate last active cid,
521	Cid is not defined	Activate or deactivate cid which is not defined.
522	Uart parity error	The mismatch between setting and using for uart parity.
523	Uart frame error	Uart frame error.
524	MT not power on	Some command could only exclude only when power on like +CIMI, otherwise will return ERROR:524

525	at command abort in processing	Send uart characters when MT is aborting at command.
526	at command abort error	Abort blocking command error.
527	command interrupted	When start back ground search by +COPS=?, will start a protect timer. It will abort when timer timeout if back ground search not finished, then return this error code.
528	configuration conflicts	When send +NPOWERCLASS=<band>,6 to set powerclass 6 and the “RELEASE_VERSION” in +NCONFIG is 13 will return this error code because release 13 don’t support powerclass 6.
529	FOTA is updating	Send UDP/TCP data when FOTA is updating.
530	not the at allocated socket	Send AT+NSOCL to close a socket which is not created.
531	SIM PIN is blocked	Continuously enter the wrong PIN 3 times.
532	SIM PUK is blocked	Continuously enter the wrong PUK 10 times.

9 Reset Reasons

If the Applications core rebooted for any reason apart from either being power cycled or being externally reset, it will return a message before the <CR><LF>Cheerzing<CR><LF>OK<CR><LF> message that indicates the reason for the reboot.

The list of possible reboot reasons are:

Reboot Message Displayed

REBOOT_CAUSE_SECURITY_RESET_UNKNOWN
REBOOT_CAUSE_SECURITY_SYSRESETREQ
REBOOT_CAUSE_SECURITY_WATCHDOG
REBOOT_CAUSE_SECURITY_SELF
REBOOT_CAUSE_SECURITY_ALTBOOT
REBOOT_CAUSE_SECURITY_REG_0
REBOOT_CAUSE_SECURITY_REG_3
REBOOT_CAUSE_SECURITY_STANDARD_CHIP_WATCHDOG
REBOOT_CAUSE_SECURITY_UPDATER_CHIP_WATCHDOG
REBOOT_CAUSE_SEUCRITY_SCAN_ENTER_EXIT
REBOOT_CAUSE_SECURITY_PMU_POWER_ON_RESET
REBOOT_CAUSE_SECURITY_RESET_PIN
REBOOT_CAUSE_SECURITY_REGIONS_UPDATED
REBOOT_CAUSE_SECURITY_FOTA_UPGRADE
REBOOT_CAUSE_PROTOCOL_SYSRESETREQ
REBOOT_CAUSE_PROTOCOL_WATCHDOG
REBOOT_CAUSE_PROTOCOL_MONITOR_REBOOT_REQ
REBOOT_CAUSE_PROTOCOL_RPC_TIMEOUT
REBOOT_CAUSE_APPLICATION_SYSRESETREQ
REBOOT_CAUSE_APPLICATION_WATCHDOG
REBOOT_CAUSE_APPLICATION_AT
REBOOT_CAUSE_APPLICATION_RPC_TIMEOUT
REBOOT_CAUSE_PROTOCOL_IMSI_UPDATE
REBOOT_CAUSE_UNKNOWN

10 Abort process

Some action commands that require time to execute may be aborted while in progress; these are explicitly noted in the description of the command. Aborting of commands is accomplished by the transmission from the DTE to the DCE of any character. A single character shall be sufficient to abort the command in progress;

however, characters transmitted during the first 125 milliseconds after transmission of the termination character shall be ignored (to allow for the DTE to append additional control characters such as line feed after the command line termination character). To insure that the aborting character is recognized by the DCE, it should be sent at the same rate as the preceding command line; the DCE may ignore characters sent at other rates. When such an aborting event is recognized by the DCE, it shall terminate the command in progress and return an appropriate result code to the DTE, as specified for the particular command.

For Implementation.

1. If command is blocking and abort is allowed.

Current command state	In 125 ms	Over 125ms	Over 125ms, current progressing command is abortable but protocol core current state is unabortable. (almost impossible)	aborting
React for sending any characters	ignore	Start aborting process, return nothing	+CME ERROR:526	ignore

2. If command is blocking but abort is not allowed.

	In 125 ms	Over 125ms
Send any characters	ignore	+CME ERROR:51

3. Abort return value.

Abort result	abort success	abort fail	abort when command complete
Return value	+CME ERROR: 52	+CME ERROR:526	Return origin command result

NOTE:

1. Only +COPS=? is abortable now.
2. If sent more than 1 character for abort over 125ms, only the first character is the abort character.
   Other character may treat as an at command string. So advice just send 1 character for abort character.

11 Examples

11.1 UDP Sockets

11.1.1 Sending a message

A simple example sending a UDP datagram. Once the socket is closed, no replies will be received.

```
AT+NSOCR=DGRAM,17,56,1
      1
      OK
      AT+NSOST=1,192.158.5.1,8080,20,”http://www.Hisi.com”
      1,19
      OK
      AT+NSOCL=1
      OK
```

11.1.2 Receiving messages from multiple remote systems

An example receiving messages from multiple hosts and consuming different amounts of the received data.

```
AT+NSOCR=DGRAM,17,1024,1
      1
      OK
      +NSONMI:1,11
      AT+NSORF=1,5
      1,192.168.5.1,1024,5,68656C6C6F,6
      OK
      AT+NSORF=1,999
      1,192.168.5.1,1024,6,20776F726C64,0
      OK
      +NSONMI:1,8
      AT+NSORF=1,8
      1,10.11.12.13,32701,8,616172647661726B,0
      OK
```

11.1.3 Server

Example of a server-style implementation, where multiple remote systems can request to communicate with the UE.

Messages:

```
      63616E204920636F6E6E656374   “can I connect”
      73757265                     “sure”
```

Example

```
AT+NSOCR=DGRAM,17,56,1
      1
      OK
      +NSONMI:1,13
      AT+NSORF=1,13
      1,192.168.5.1,1234,13,63616E204920636F6E6E656374,0
      OK
      AT+NSOCR=DGRAM,17,45678,1
       2
      OK
      AT+NSOST=2,192.158.5.1,1234,4,73757265
      2,4
      OK
```

```
+NSONMI:1,13
AT+NSORF=1,13
1,10.11.12.13,2345,13,63616E204920636F6E6E656374,0
OK
AT+NSOCR=DGRAM,17,45679,1
3
OK
AT+NSOST=3,210.11.12.13,2345,4,73757265
3,4
OK
```

12 Contact

Xiamen Cheerzing IOT Technology Co., Ltd.
TEL：+86-592-5950030
FAX：+86-592-5950028
Home：www.cheerzing.com
Add：8/F,No.7 Building, Guanyinshan International Business Center, Siming District, Xiamen, China, 361000