BC26
AT Commands Manual

NB-IoT Module Series

Rev. BC26_AT_Commands_Manual_V1.0

Date: 2018-01-05

Status: Preliminary

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://quectel.com/support/technical.htm
Or email to: support@quectel.com

About the Document

History

Revision	Date	Author	Description
1.0	2018-01-05	Randy LI/ Lebron LIU	Initial

Contents

Table Index

1 Introduction

1.1. Scope of the Document

This document gives details of the AT Command Set supported by Quectel NB-IoT BC26 module.

1.2. AT Command Syntax

The "**AT**" or "**at**" prefix must be set at the beginning of each command line. To terminate a command line enter **<CR>**. Commands are usually followed by a response that includes "**<CR><LF><response><CR><LF>**". Throughout this document, only the responses are presented, "**<CR><LF>**" are omitted intentionally.

The AT Commands Set implemented by BC26 is a combination of *3GPP TS 27.005*, *3GPP TS 27.007* and *ITU-T recommendation V.250* as well as the AT Commands developed by Quectel.

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

● **Basic syntax**

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

● **S parameter syntax**

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

● **Extended syntax**

These commands can be operated in several modes, as following table:

Table 1: Types of AT Commands and Responses

Test Command	**AT+<x>=?**	This command returns the list of parameters and value ranges set by the corresponding Write Command or internal processes.
Read Command	**AT+<x>?**	This command returns the currently set value of the parameter or parameters.
Write Command	**AT+<x>=<…>**	This command sets the user-definable parameter values.
Execution Command	**AT+<x>**	This command reads non-variable parameters affected by internal processes in the module.

NOTE

Each time a single AT command is supported. When execution of the previous AT command is finished, the next AT command will be executed.

1.3. Supported Character Sets

The BC26 AT command interface defaults to the IRA character set. The BC26 supports the following character sets:

- GSM
- UCS2
- HEX
- IRA
- PCCP437
- PCDN
- 8859-1

The character set can be configured and interrogated using the **AT+CSCS** command (*3GPP TS 27.007*). The character set is defined in *3GPP specification 27.005*.

2 3GPP Commands (27.007)

2.1. ATI Display Product Identification Information

ATI Display Product Identification Information	
Execution Command **ATI**	Response **Quectel_Ltd** **\<Object Id>** **Revision:\<revision>** **OK** If there is any error, response: **+CME ERROR:\<err>**

Parameter

\<Object Id>	Identifier of device type
\<revision>	Revision of software release

Example

```
ATI
Quectel_Ltd
Quectel_BC26
Revision: BC26NAR01A01

OK
```

2.2. ATE Set Command Echo Mode

ATE Set Command Echo Mode	
Execution Command **ATE[<value>]**	Response **OK** If there is any error, response: **+CME ERROR:<err>**

Parameter

<value>	0	Echo mode OFF
	1	Echo mode ON

Example

```
ATE0
OK
ATI
Quectel
Quectel_BC26
Revision: BC26NAR01A01

OK
ATE1
OK
ATI
ATI
Quectel
Quectel_BC26
Revision: BC26NAR01A01

OK
```

2.3. AT+CGMI Request Manufacturer Identification

AT+CGMI Request Manufacturer Identification	
Execution Command **AT+CGMI**	Response **Quectel_Ltd** **<Object Id>** **Revision: MTK_2625** **OK** If there is any error, response: **+CME ERROR:<err>**
Test Command **AT+CGMI=?**	Response **OK**

Parameter

<Object Id> Identifier of device type

Example

AT+CGMI
Quectel_Ltd
Quectel_BC26
Revision: MTK_2625

OK

2.4. AT+CGMM Request Manufacturer Model

AT+CGMM Request Manufacturer Model	
Execution Command **AT+CGMM**	Response **<Object Id>** **OK** If there is any error, response: **+CME ERROR:<err>**

Test Command	Response
AT+CGMM=?	**OK**

Parameter

<Object Id>	Identifier of device type

Example

> **AT+CGMM**
> **Quectel_BC26**
>
> **OK**

2.5. AT+CGMR Request Manufacturer Revision

AT+CGMR Request Manufacturer Revision	
Execution Command **AT+CGMR**	Response **Revision :<revision>** **OK** If there is any error, response: **+CME ERROR:<err>**
Test Command **AT+CGMR=?**	Response **OK**

Parameter

<revision>	Revision of software release

Example

> **AT+CGMR**
> **Revision: BC26NAR01A01**
>
> **OK**

2.6. AT+CESQ   Extended Signal Quality

The execution command returns received signal quality parameters. If the current serving cell is not a GERAN cell, <rssi> and <ber> are set to value 99. If the current serving cell is not a UTRA FDD or UTRA TDD cell, <rscp> is set to 255. If the current serving cell is not a UTRA FDD cell, <ecno> is set to 255. If the current serving cell is not an E-UTRA cell, <rsrq> and <rsrp> are set to 255.

AT+CESQ Extended Signal Quality	
Execution Command **AT+CESQ**	Response **+CESQ: <rxlev>,<ber>,<rscp>,<ecno>,<rsrq>,<rsrp>** **OK** If there is any error, response: **+CME ERROR:<err>**
Test Command **AT+CESQ=?**	Response **+CESQ:** (list of supported **<rxlev>**s), (list of supported **<ber>**s), (list of supported **<rscp>**s), (list of supported **<ecno>**s), (list of supported **<rsrq>**s), (list of supported **<rsrp>**s) **OK**

Parameter

<rxlev>	Integer type. Rx signal strength level
	0 -110 dBm or less
	1 -110 dBm ≤ **<rssi>** < -109 dBm
	2 -109 dBm ≤ **<rssi>** < -108 dBm
	…..: ….
	61 -50dBm ≤ **<rssi>** < -49 dBm
	62 -49dBm ≤ **<rssi>** < -48 dBm
	63 -48dBm ≤ **<rssi>**
	99 not known or not detectable
<ber>	Integer type; channel bit error rate (in percent)
	0…7 as RXQUAL values RXQUAL_0…RXQUAL_7 as defined in 45.008.
	99 not known or not detectable
<rscp>	Integer type, received signal code power (see *3GPP 25.133* and *3GPP 25.123*)
	0 -120 dBm or less
	1 -120 dBm ≤ **<rscp>** < -119 dBm
	2 -119 dBm ≤ **<rscp>** < -118 dBm
	…: …

94	-27 dBm ≤ **\<rscp\>** < -26 dBm
95	-26 dBm ≤ **\<rscp\>** < -25 dBm
96	-25 dBm ≤ **\<rscp\>**
255	not known or not detectable

\<ecno\>　Integer type, Ec/No (see *3GPP 25.133*)

0	-24 dBm or less
1	-24dBm ≤ **\<ecno\>** < -23.5 dBm
2	-23.5dBm ≤ **\<ecno\>** < -23 dBm
…: …	
47	-1dBm ≤ **\<ecno\>** < -0.5 dBm
48	-0.5dBm ≤ **\<ecno\>** < 0 dBm
49	0 dBm ≤ **\<ecno\>**
255	not known or not detectable

\<rsrq\>　Integer type, reference signal received quality (see *3GPP 36.133*)

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

\<rsrp\>　Integer type, reference signal received power (see 3GPP 36.133)

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

Example

```
AT+CESQ
+CESQ: 15,99,255,255,8,30

OK
```

2.7. AT+CGSN   Request Product Serial Number Identification

AT+CGSN Request Product Serial Number Identification	
Execution/Write Command **AT+CGSN[=<snt>]**	Response When <snt>=0 (or omitted) and command is executed successfully: **<sn>** When <snt>=1 and command is executed successfully: **+CGSN:<imei>** When <snt>=2 and command is executed successfully: **+CGSN:<imeisv>** When <snt>=3 and command is executed successfully: **+CGSN:<svn>** **OK** If there is any error, response: **+CME ERROR:<err>**
Test Command **AT+CGSN=?**	Response When TE supports <snt> and command is executed successfully: **+CGSN:**(list of supported **<snt>**s) **OK**

Parameter

<snt>	Integer type indicating the serial number type that has been requested
	0 Returns <sn>
	1 Returns the IMEI number
	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, and shall not contain the sequence 0 <CR> or OK<CR>.
<imei>	String type in decimal format indicating the IMEI number
<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

2.8. AT+CEREG EPS Network Registration Status

The write command controls the presentation of an unsolicited result code +CEREG: <stat> when <n>=1 and there is a change in the UE's EPS network registration status, or result code +CEREG: <stat>[,<tac>,<ci>[,<AcT>]] when <n>=2 and there is a change of the network cell.

NOTE: If the EPS UE 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> that shows whether the network has currently indicated the registration of the UE. Location information elements <tac>, <ci> and <AcT> are returned only when <n>=2 and UE is registered in the network.

AT+CEREG EPS Network Registration Status	
Write Command **AT+CEREG=[<n>]**	Response **OK** If there is any error, response: **+CME ERROR:<err>**
Read Command **AT+CEREG?**	Response When <n>=0, 1, 2 or 3 and command is executed successfully: **+CEREG:<n>,<stat>[,[<tac>],[<ci>],[<AcT>[,<cause_type>,<reject_ cause>]]]** When <n>=4 or 5 and command is executed successfully: **+CEREG:<n>,<stat>[,[<lac>],[<ci>],[<AcT>],[<rac>][,[<cause_type>],[<reject_cause>][,[<Active-Time>],[<Periodic-TAU>]]]]** If there is any error, response: **+CME ERROR:<err>**
Test Command **AT+CEREG=?**	Response **+CEREG:**(list of supported **<n>**s) **OK**

Parameter

<n>	Integer type
	<u>0</u> 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 requests PSM, enable network registration and location information unsolicited result code:
"+CEREG:<stat>[,[<tac>],[<ci>],[<AcT>][,,[,[<Active-Time>],[<Periodic-TAU>]]]]"

    5   For a UE that requests PSM, enable network registration, location information and EMM cause value information unsolicited result code:
+CEREG:<stat>[,[<lac>],[<ci>],[<AcT>],[<rac>][,[<cause_type>],[<reject_cause>][,[<Ac tive-Time>],[<Periodic-RAU>],[<GPRS-READY-timer>]]]]

<stat>   Integer type, indicates the EPS registration status

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

    1   Registered, home network

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

    3   Registration denied

    4   Unknown

    5   Registered, roaming

    6   Registered for "SMS only", home network (applicable only when <Act> indicates NB-IoT)

    7   Registered for "SMS only", roaming (applicable only when <Act> indicates NB-IoT

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

<ci>   String type; four bytes E-UTRAN cell ID in hexadecimal format

<AcT>   Integer type; access technology of the registered network

    9   NB-IoT

<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 value

<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, please refer to the *GPRS Timer 2 IE in 3GPP TS 24.008 Table 10.5.163/3Gpp TS 24.008*, *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, please refer to the *GPRS Timers 3 IE in 3GPP TS 24.008 Table 10.5.163a/3GPP TS 24.008*, *3GPP TS 23.682* and *3GPP TS 23.401.*

Example

AT+CEREG=1	//Enable network registration URC.
OK	
AT+CEREG?	
+CEREG:1,1	
OK	
AT+CEREG=?	
+CEREG:(0-5)	
OK	

2.9. AT+CSCON Signalling Connection Status

The write 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.

For NB-IoT, only <n>=1 is supported.

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.

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.

Test command returns supported values as a compound value.

AT+CSCON Signalling Connection Status	
Write command **AT+CSCON=[<n>]**	Response **OK** If there is any error, response: **+CME ERROR:<err>**
Read Command **AT+CSCON?**	Response **+CSCON:<n>,<mode>[,<state>]**

	OK
	If there is any error, response:
	+CME ERROR:<err>
Test Command	Response
AT+CSCON=?	**+CSCON:**(list of supported **<n>**s)
	OK

Parameter

<n>	Integer type
	0 Disable unsolicited result code
	1 Enable unsolicited result code: "+CSCON:<mode>"
<mode>	Integer type; indicates the signalling connection status
	0 Idle
	1 Connected

Example

```
AT+CSCON=0
OK
AT+CSCON?
+CSCON:0,1

OK
AT+CSCON=?
+CSCON:(0-1)

OK
AT+CSCON=1
OK
AT+CSCON?
+CSCON:1,1

OK
```

2.10. AT+COPS   Operator Selection

AT+COPS Operator Selection	
Write Command	Response
AT+COPS=[<mode>[,<format>[,<oper	**OK**

>[,<AcT>]]]]	
	If there is any error, response:
	+CME ERROR:<err>
Read Command **AT+COPS?**	Response **+COPS:<mode>[,<format>,<oper>][,<AcT>]** **OK** If there is any error, response: **+CME ERROR:<err>**
Test Command **AT+COPS=?**	Response **+COPS:** list of supported **(<stat>,** long alphanumeric **<oper>,** short alphanumeric **<oper>,** numeric **<oper>[,<AcT>])**s **[,,**(list of supported **<mode>**s**),**(list of supported **<format>**s**)]** **OK** If there is any error, response: **+CME ERROR:<err>**

Parameter

<mode>	Integer type	
	<u>0</u>	Automatic mode; <oper> field is ignored
	1	Manual operator selection; <oper> field shall be present
	2	Manual deregister from network
	3	Set only <format> (for read command AT+COPS?), not shown in Read command response
	4	Manual/automatic selected; if manual selection fails, automatic mode (<mode>=0) is entered
<format>	Integer type	
	<u>0</u>	Long format alphanumeric <oper>; can be up to 16 characters long
	1	Short format alphanumeric <oper>
	2	Numeric <oper>; GSM Location Area Identification number
<oper>	String type; operator in format as per <mode>.	
<stat>	Integer type	
	0	Unknown
	1	Operator Available
	2	Operator Current
	3	Operator Forbidden
<AcT>	Integer type;	
	9	NB-IoT

> **NOTE**
>
> <AcT> field is fixed at 9 for NB-IoT product .

Example

```
AT+COPS=0
OK
AT+COPS?
+COPS:0,2,"46000",9

OK
```

2.11. AT+CGATT   GPRS/Packet Domain Attach/Detach

AT+CGATT GPRS/Packet Domain Attach/Detach	
Write Command **AT+CGATT=<state>**	Response **OK** If there is any error, response: **+CME ERROR:<err>**
Read Command **AT+CGATT?**	Response **+CGATT:<state>** **OK**
Test Command **AT+CGATT=?**	Response **+CGATT:**(list of supported **<state>**s) **OK**

Parameter

<state>	Integer type; indicates the state of GPRS/Packet Domain attachment
	    0 Detached
	    1 Attached
	Other values are reserved and will result in an ERROR response to the execution command.

Example

```
AT+CGATT?
+CGATT:0

OK
AT+CGATT=1
OK
AT+CGATT=?
+CGATT:(0,1)

OK
```

2.12. AT+CGDCONT   Define the PDP Context

The write command specifies PDP context parameter values for a PDP context identified by the (local) context identification parameter, <cid>. The number of PDP contexts that may be in a defined state at the same time is given by the range returned by the test command.

For EPS the PDN connection and its associated EPS default bearer is identified herewith. For EPS the <PDP_addr> shall be omitted.

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

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

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

The feature "initial PDP context" may be supported and is a manufacturer specific option. For this option, the context with <cid>=0 (context number 0) is defined upon startup and does not need to be created with the +CGDCONT command. The initial PDP context has particular manufacturer specific default settings disassociated with any other default settings of +CGDCONT. When in E-UTRAN, the initial PDP context is automatically activated by the MT following a successful registration to the network depending on the setting of AT+CIPCA command. If all active contexts are deactivated, the initial PDP context can be (re)established. This is manufacturer specific and depends on the current RAT as well as how the active contexts are deactivated.

AT+CGDCONT Define the PDP Context	
Write Command	Response
+CGDCONT=[<cid>[,<PDP_type>[,<A	OK

PN>[,<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>][,<Non-IP_MTU_discovery>]]]]]]]]]]]]]	If there is any error, response: **+CME ERROR:<err>**
Read Command **AT+CGDCONT?**	Response **[+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>[,<Non-IP_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>[,<Local_Addr_Ind>[,<Non-IP_MTU_discovery>]]]]]]]]]** **[...]]** **OK**
Test Command **AT+CGDCONT=?**	Response **+CGDCONT**:(rangeofsupported **<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),(list of supported **<Local_Addr_Ind>**s),(list of supported **<Non-IPMTUdiscovery>**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),(list of supported **<Local_Addr_Ind>**s),

	,(list of supported <Non-IP_MTU_discovery>s)[...]]
	OK

Parameter

<cid>   Integer type; a numeric parameter that specifies a particular PDP context definition. The parameter is local to the UE-TE 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.

<PDP_type>String type; a string parameter which specifies the type of packet data protocol.

       IP   Internet Protocol (IETF STD 5)

       IPv6   Internet Protocol, version 6 (IETF RFC 2460)

       IPv4v6   Virtual <PDP_type> introduced to handle dual IP stack UE capability

       Non-IP   Transfer of Non-IP data to external packet data Network

<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.

<PDP_addr>String type; a string parameter that identifies the UE in the address space applicable to the PDP. If the value is null or omitted, then a value may be provided by the TE during the PDP startup procedure or, failing that, a dynamic address will be requested. The read form of the command will continue to return the null string even if an address has been allocated during the PDP startup procedure. The allocated address may be read using the +CGPADDR command.

<d_comp> Integer type; controls PDP data compression

       0   Off

       1   On (manufacturer preferred compression)

       2   V.42bis

<h_comp> Integer type; controls PDP header compression

       0   Off (default if value is omitted)

       1   On (manufacturer preferred compression)

       2   RFC 1144 (applicable for SNDCP only)

       3   RFC 2507

       4   RFC 3095[ROHC] (applicable for PDCP only)

< Non-IP_MTU_discovery> Integer type; influences how the MT/TA requests to get the Non-IP MTU size, see *3GPP TS 24.008 sub-clause 10.5.6.3*.

       0   Preference of Non-IP MTU size discovery not influenced by +CGDCONT

       1   Preference of Non-IP MTU size discovery through NAS signaling

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

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

       1   Preference of IPv4 MTU size discovery through NAS signalling

Example

```
AT+CGDCONT=?
+CGDCONT: (1-15), "IP",,,(0-2),(0-4),(0),,,,,,(0-1),,(0-1)
+CGDCONT: (1-15), "IPV6",,,(0-2),(0-4),(0),,,,,,(0-1),,(0-1)
+CGDCONT: (1-15), "IPV4V6",,,(0-2),(0-4),(0),,,,,,(0-1),,(0-1)
+CGDCONT: (1-15), "Non-IP",,,(0-2),(0-4),(0),,,,,,(0-1),,(0-1)

OK
AT+CGDCONT=1,"IP","CMNET"
OK
AT+CGDCONT?
+CGDCONT: 1,"IP","CMNET" ,"",0,0,0,,,,,,0,,0

OK
```

2.13. AT+CGACT   PDP Context Activation/Deactivation

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

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

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

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

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

- The PDN connection is preserved in the protocol stack
- The PDN connection is disconnected at the middleware, so the <cid> for the PDN connection is marked as deactivated
- OK response is returned rather than ERROR

For EPS, the activation request for an EPS bearer resource will be answered by the network by either an

EPS dedicated bearer activation or EPS bearer modification request. The request must be accepted by the UE before the PDP context can be set in to established state.

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

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

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

AT+CGACT PDP Context Activation/Deactivation	
Write Command **AT+CGACT=[<state>[,<cid>[,<cid>[,...]]]]**	Response **OK** If there is any error, response: **+CME ERROR:<err>**
Read Command **AT+CGACT?**	Response **+CGACT:<cid>,<state>** **[<CR><LF>+CGACT: <cid>,<state>** **[…]]** **OK**
Test Command **AT+CGACT=?**	Response **+CGACT:**(list of supported **<state>**s) **OK**

Parameter

<state>	Integer type; indicates the state of PDP context activation. 0 Deactivated 1 Activated Other values are reserved and will result in an ERROR response to the execution command.
<cid>	Integer type; a numeric parameter which specifies a particular PDP context definition (see +CGDCONT command).

NOTES

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

Example

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

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

OK
```

2.14. AT+CGPADDR   Show PDP Addresses

AT+CGPADDR Show PDP Addresses	
Write Command **AT+CGPADDR[=<cid>[,<cid>[,...]]]**	Response **+CGPADDR: <cid>,<PDP_addr>** **[<CR><LF>+CGPADDR: <cid>,<PDP_addr>** **[...]]** **OK**
Test Command **AT+CGPADDR=?**	Response **+CGPADDR:**(list of defined **<cid>**s) **OK**

Parameter

<cid>   Integer type; a numeric parameter which specifies a particular PDP context definition (see AT+CGDCONT command). If no <cid> is specified, the addresses for all defined contexts are returned.

<PDP_addr>   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 AT+CGDCONT command when the context was defined.
For a dynamic address it will be the one assigned during the last PDP context activation that used the context definition referred to by <cid>. <PDP_address> is omitted if none is available.

Example

> **AT+CGPADDR=0**
> +CGPADDR:0,101.43.5.1
>
> OK
> **AT+CGPADDR=?**
> +CGPADDR:(0)
>
> OK

2.15. AT+CIMI   Request International Mobile Subscriber Identity

AT+CIMI Request International Mobile Subscriber Identity	
Execution Command **AT+CIMI**	Response **+CIMI: <IMSI>** **OK** If there is any error, response: **+CME ERROR:<err>**
Test Command **AT+CIMI=?**	Response **OK**

Parameter

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

Example

> **AT+CIMI**
> 460001357924680
>
> OK

2.16. AT+CFUN   Set Phone Functionality

AT+CFUN Set Phone Functionality	
Write Command **AT+CFUN=[<fun>[,<rst>]]**	Response **OK** If there is any error, response: **+CME ERROR:<err>**
Read Command **AT+CFUN?**	Response **+CFUN:<fun>** **OK** If there is any error, response: **+CME ERROR:<err>**
Test Command **AT+CFUN=?**	Response **+CFUN:**(list of supported **<fun>**s)**,** (list of supported **<rst>**s) **OK** If there is any error, response: **+CME ERROR:<err>**

Parameter

<fun>	Integer type
	0 Minimum functionality
	<u>1</u> Full functionality
	4 Disable phone both transmit and receive RF circuits
	7 Disable phone SIM only. Transmit and receive circuits still active
<rst>	Integer type; MT resetting
	<u>0</u> Do not reset the MT before setting it to <fun> power level.
	1 Only set to <fun> power level after MT has been reset, and for all subsequent resets.
	2 Do not reset MT before setting it to <fun> power level and save <fun> value in NVRAM for all subsequent resets.

Example

```
AT+CFUN=?
+CFUN:(0,1,4,7),(0-2)

OK
AT+CFUN=1
```

```
OK
AT+CFUN?
+CFUN:1

OK
```

2.17. AT+CMEE   Report Mobile Termination Error

AT+CMEE Report Mobile Termination Error	
Write Command **AT+CMEE=\<n>**	Response TA disables or enables the use of result code +CME ERROR: \<err> as an indication of an error relating to the functionality of the ME. **OK**
Read Command **AT+CMEE?**	Response **+CMEE:\<n>** **OK**
Test Command **AT+CMEE=?**	Response **+CMEE:**(list of supported **\<n>**s) **OK**

Parameter

\<n>	Integer type	
	0	Disable result code
	1	Enable result code and use numeric values
	2	Enable result code and use verbose values

2.18. AT+CPSMS   Power Saving Mode Setting

The write 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 TAU value in E-UTRAN and the requested Active Time value. See the unsolicited result codes provided by commands +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 TAU value in E-UTRAN and the requested Active Time value as compound values.

AT+CPSMS Power Saving Mode Setting	
Write Command **AT+CPSMS=[<mode>[,<Requested_Periodic-RAU>[,<Requested_GPRS-READY-timer>[,<Requested_Periodic-TAU>[,<Requested_Active-Time>]]]]]**	Response **OK** If there is any error, response: **+CME ERROR:<err>**
Read Command **AT+CPSMS?**	Response **+CPSMS:<mode>,[<Requested_Periodic-RAU>],[<Requested_GPRS-READY-timer>],[<Requested_Periodic-TAU>],[<Requested_Active-Time>]** **OK** If there is any error, response: **+CME ERROR:<err>**
Test Command **AT+CPSMS=?**	Response **+CPSMS:**(list of supported **<mode>**s),(list of supported **<Requested_Periodic-RAU>**s)**,**(list of supported **<Requested_GPRS-READY-timer>**s),(list of supported **<Requested_Periodic-TAU>**s),(list of supported **<Requested_Active-Time>**s) **OK**

Parameter

<mode>	Integer type, indicates 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; N/A for NB-IoT
<Requested_GPRS-READY-timer>	String type; N/A for NB-IoT
<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.* 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.

Example

```
AT+CPSMS=1,,,"01000011","01000011"
OK
AT+CPSMS?
+CPSMS:1,,,"01000011","01000011"

OK
AT+CPSMS=?
+CPSMS:(0-2),,,( "00000000"-"11111111"), ("00000000"-"11111111")

OK
```

2.19. AT+CEDRXS   eDRX Setting

The write 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 write 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.

AT+CEDRXS eDRX Setting	
Write Command **AT+CEDRXS=[<mode>,[,<AcT-type>[, <Requested_eDRX_value>]]]**	Response **OK** If there is any error, response: **+CME ERROR:<err>**
Read Command **AT+CEDRXS?**	Response **+CEDRXS:<AcT-type>,<Requested_eDRX_value>** **[<CR><LF>+CEDRXS:<AcT-type>,<Requested_eDRX_val** **ue>** **[...]]]** **OK**
Test Command **AT+CEDRXS=?**	Response **+CEDRXS:**(list of supported **<mode>s),**(list of supported **<AcT-type>s),**(list of supported **<Requested_eDRX_value>s**) **OK**

Parameter

<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. AT+CEDRXS? is used to specify the relationship between the type of access technology and the requested eDRX value.
	0 Access technology is not using eDRX. This parameter value is only used in the unsolicited result code.
	5 E-UTRAN (NB-S1 mode)
<Requested_eDRX_value>	String type; half a byte in a 4 bit format. The eDRX value refers to bit 4 to 1 of octet 3 of the extended DRX parameters information element (see *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 refers 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.*

Example

```
AT+CEDRXS=1,4,"0101"
OK

AT+CEDRXS?
+CEDRXS:4,"0101"

OK
AT+CEDRXS=?
+CEDRXS:(0-3),(4),("0000"-"1111")

OK
```

2.20. AT+CEDRXRDP   eDRX Read Dynamic Parameters

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.

AT+CEDRXRDP eDRX Read Dynamic Parameters	
Execution Command **AT+CEDRXRDP**	Response **+CEDRXRDP:\<AcT-type>[,\<Requested_eDRX_value>[,\<NW-provided_eDRX_value>[,\<Paging_time_window>]]]**

	OK
Test Command **AT+CEDRXRDP=?**	Response **OK**
Read Command **AT+CEDRXRDP?**	Response **OK** If there is any error, response: **+CME ERROR:<err>**

Parameter

<AcT-type>	Integer type, indicates the type of access technology. AT+CEDRXS? is used to specify the relationship between the type of access technology and the requested eDRX value.
	0 Access technology is not using eDRX. This parameter value is only used in the unsolicited result code.
	5 E-UTRAN (NB-S1 mode)
<Requested_eDRX_value>	String type; half a byte in a 4 bit format. The eDRX value refers to bit 4 to 1 of octet 3 of the extended DRX parameters information element (see *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 refers 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*.

Example

```
AT+CEDRXRDP
+CEDRXRDP:4,2,2,2

OK
AT+CEDRXRDP=?
OK
```

2.21. AT+CEER   Extended Error Report

AT+CEER Extended Error Report	
Execution Command **AT+CEER**	Response **+CEER <report>** **OK**
Test Command **AT+CEER=?**	Response **OK**

Parameter

<report>   Reason for last call release as string.

Example

AT+CEER
+CEER:EMM_CAUSE_EPS_AND_NON_EPS_SERVICES_NOT_ALLOWED

OK
AT+CEER=?
OK

3 PDN Commands

3.1. AT+QGACT   Activate/Deactivate a PDN Context

Activate or deactivate a specified PDN context.

There are three kinds of responses for both the activation requirement and the deactivation requirement. If the PDN context is active/inactive, "+QGACT=<cid>,<type>,<result>[,<activated_pdp_type>] OK" is returned immediately for activation/deactivation requirement. If not, "+QGACT=<cid> OK" is returned first and URC"+QGACT=<cid>,<type>,<result>[,< activated_pdp_type>]" is reported for the activation/deactivation result latter. If any error occurs, such as invalid parameter(s), "ERROR" is returned immediately. In any case, activated_pdp_type only exists for the activation requirement.

Besides, URC "+QGACT=<cid>,<type>" is reported when passive deactivation occurs.

For the activation requirement, the format of the QGACT is "+QGACT=<op>,<pdp_type>,<apn>,<user_name>,<pwd>[,<bearer_type>[,<sim_id>]]", while for the deactivation requirement, the format is "+QGACT=<op>,<cid>". For the purpose of normalization, the format of QGACT is:
"+QGACT=<op>,<pdp_type/cid>[,<apn>,<user_name>,<pwd>[,<bearer_type>[,<sim_id>]]]".

AT+QGACT Activate/Deactivate A PDN Context	
Write Command **AT+QGACT=<op>,<pdp_type/cid>[,<apn>,<user_name>,<pwd>[,<bearer_type>[,<sim_id>]]]**	Response **+QGACT=<cid>,<type>,<result>[,<activated_pdp_type>]** **OK** **+QGACT=<cid>** **OK** If there is any error, response: **ERROR**
	+QGACT=<cid>,<type>,<result>[,<activated_pdp_type>]

Parameter

\<op\>	Integer type;
	0 Deactivation requirement
	1 Activation requirement
\<pdp_type/cid\>	Integer type;
	If \<op\> is 0, it is pdp_type. Otherwise, it is cid.
pdp_type:	It is the pdp_type wanted to activate.
	1 IPv4
	2 IPv6
	3 IPv4v6
	4 Non-IP
cid:	It is a numeric parameter specifying a particular PDP context. Here it should be equal to the \<cid\> returned by the activation response.
\<apn\>	It is the access point name which is mandatory for the activation requirement and should be omitted for the deactivation requirement.
\<user_name\>	It is the user name for access to the IP network which is mandatory for the activation requirement and should be omitted for the deactivation requirement.
\<pwd\>	It is the password for access to the IP network which is mandatory for the activation requirement and should be omitted for the deactivation requirement.
\<bearer_type\>	It is the type of bearer wanted to activate which is optional for the activation requirement and should be omitted for the deactivation requirement.
	1 NBIOT (Only NBIOT is supported currently)
\<sim_id\>	It is the id of the SIM Card wanted to use which is optional for the activation requirement and should be omitted for the deactivation requirement.
	1 SIM Card 1 (Only SIM Card 1 is supported currently).
\<cid\>	It is a numeric parameter specifying a particular PDP context.
\<type\>	0 Result/URC for deactivation requirement
	1 Result/URC for activation requirement
	2 URC for passive deactivation
\<result\>	0 failure
	1 success
\<activated_pdp_type\>	It is the pdp_type actually activated.
	1 IPv4
	2 IPv6
	3 IPv4v6
	4 Non-IP

NOTES

This command name is temporary and may be changed in the future release.

Example

AT+QGACT=1,1,"apn_example","username_example","password_example"
// Activate a PDN context
+QGACT=1
OK   //Return OK immediately for no error

+QGACT=1,1,1,1   //Notify activation result via URC

AT+QGACT=0,1   //Deactivate a PDN context
+QGACT=1
OK   //Return OK immediately for no error

+QGACT=1,0,1   //Notify deactivation result via URC

4 Network Commands – PING

4.1. AT+QPING   Test IP Network Connectivity to a Remote Host

This command sends an ICMP packet to the specified host address.

AT+QPING 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.

AT+QPING Test IP Network Connectivity to a Remote Host	
Write Command **AT+QPING=<remote addr> [-l p_size>] [-n count]**	Response **OK** If there is any error, response: **ERROR**

Parameter

<remote addr>	Address of system sending the message IPv4: A dot notation IP address
<p_size>	Size in bytes of echo packet payload. Range of 8-1460 Default 64
<count>	Number of packet to send. Default 3

Example

AT+QPING=192.168.1.1
OK

5 Network Commands – Socket

The AT command names listed below are temporary, and may be changed in the future release.

5.1. AT+QSOC   Create a TCP/UDP Socket

This command creates a TCP or UDP socket.

AT+QSOC Create a TCP/UDP Socket	
Write Command **AT+QSOC=<domain>,<type>,<protocol>[,<cid>]**	Response **+QSOC=<socket_id>** **OK** If there is any error, response: **ERROR**
Test Command **AT+QSOC=?**	Response **+QSOC=**(list of supported **<domain>**s),(list of supported **<type>**s),(list of supported **<protocol>**s)[,(list of supported **<cid>**s)] **OK**

Parameter

<domain>	Integer type;	
	1	IPv4
	2	IPv6
<type>	Integer type;	
	1	TCP
	2	UDP
	3	RAW
<protocol>	Integer type;	
	1	IP
	2	ICMP
<cid>	Integer type; PDP context ID, AT+QGACT response.	

5.2. AT+QSOB   Bind Local Address and Local Port

This command is used to bind local address and local port. Used chiefly set local port.

AT+QSOB Bind Local Address and Local Port	
Write Command **AT+QSOB=<socket_id>,<local_port>[,<local_address>]**	Response **OK** If there is any error, response: **ERROR**
Test Command **AT+QSOB=?**	Response **+QSOB=**(list of supported **<socket_id>**s),(list of supported **<local_port>**s)**[,<local_address>]** **OK**

Parameter

<socket_id>	Integer, socket id, AT+QSOC's response. Range: 0-4
<local_port>	Integer, local port. Range: 0-65535
<local_address>	String, local address.

5.3. AT+QSOCON   Connect Socket to Remote Address and Port

This command is used to connect socket to remote address and port.

If the socket is TCP, it will start to send TCP SYN packet; if the socket is UDP, there is no packet being sent.

AT+QSOCON Connect Socket to Remote Address and Port	
Write Command **AT+QSOCON=<socket_id>,<remote_port>,<remote_address>**	Response **OK** If there is any error, response: **ERROR**
Test Command **AT+QSOCON=?**	Response **+QSOCON=**(list of supported **<socket_id>**s),(list of supported **<remote_port>**s)**,<remote_address>** **OK**

Parameter

<socket_id>	Integer, socket id, AT+QSOC's response. Range: 0-4
<remote_port>	Integer, remote port. Range: 0-65535
<remote_address>	String, remote address.

5.4. AT+QSOSEND   Send Data to Remote via Socket

This command is used to send data to network.

The response "OK" just mean the ATCMD format is right and data have been put to socket, waiting for send.

AT+QSOSEND Send Data to Remote via Socket	
Write Command **AT+QSOSEND=<socket_id>,<data_len>,<data>[,<flag>]**	Response **OK** If there is any error, response: **ERROR**
Test Command **AT+QSOSEND=?**	Response **+QSOSEND=**(list of supported **<socket_id>**s)**,<data_length>, <data>**[,(list of supported **<flag>**s)] **OK**

Parameter

<socket_id>	Integer, socket id, AT+QSOC's response. Range: 0-4
<data_len>	Integer, length of data.
<data>	Raw data, data context. Hex string
<flag>	Integer, send flag.
1	ack no delay.
2	no nagle.

5.5. AT+QSODIS　Disconnect Socket

This command is used to disconnect socket.

If the socket is TCP, it will start to send TCP FIN packet; if the socket is UDP, there is no packet being sent.

AT+QSODIS　Disconnect Socket	
Write Command **AT+QSODIS=\<socket_id\>**	Response **OK** If there is any error, response: **ERROR**
Test Command **AT+QSODIS=?**	Response **+QSODIS=**(list of supported **\<socket_id\>**s) **OK**

Parameter

\<socket_id\>　　Integer, socket id, AT+QSOC's response. Range: 0-4

5.6. AT+QSOCL　Close Socket

This command is used to close socket.

AT+QSOCL　Close Socket	
Write Command **AT+QSOCL=\<socket_id\>**	Response **OK** If there is any error, response: **ERROR**
Test Command **AT+QSOCL=?**	Response **+QSOCL=**(list of supported **\<socket_id\>**s) **OK**

Parameter

<socket_id>	Integer, socket id, AT+QSOC's response. Range: 0-4

5.7. +QSONMI   Socket Message Arrived Indicator

Indicated there is received some data from network.

+QSONMI Socket Message Arrived Indicator	
URC **+QSONMI:**	Possible Response **+QSOCL=<socket_id>,<data_len>,<data>**

Parameter

<socket_id>	Integer, socket id, AT+QSOC's response.
<data_len>	Integer, length of data.
<data>	Raw data.

5.8. +QSOERR   Socket Error Indicator

Indicated there is some error.

+QSOERR Socket Error Indicator	
URC **+QSOERR:**	Response **+QSOERR=<socket_id>,<error_code>**

Parameter

<socket_id>	Integer, socket id, AT+QSOC's response.
<error_code>	Integer, error code. 1 - Reset by peer point. 2 - Network disconnect.

5.9. Examples

5.9.1. Example of Creating a TCP Socket (Do Not Support Now)

```
AT+QPING=192.168.1.1
OK

AT+QGACT=1,1,"apn","user_name","pwd"          //Activate APN
+QGACT=<cid>,1,1,1
OK

AT+QSOC=1,1,1                                  //Create socket
+QSOC=0
OK

AT+QSOCON=0,1026, "220.180.239.212"           //Connection socket
OK

AT+QSOSEND=0,10,1234562112                     //Send data
OK
+QSONMI=0,3,3BX                                //Receive data

AT+QSODIS=0                                    //Disconnect socket
OK

AT+QSOCL=0                                     //Close socket
OK
```

5.9.2. Example of Creating a UDP Socket Example

```
AT+QGACT=1,1,"apn","user_name","pwd"          //Activate APN
+QGACT=<cid>,1,1,1
OK

AT+QSOC=1,2,1                                  //Create socket
+QSOC=0
OK

AT+QSOCON=0,1026, "220.180.239.212"           //Connection socket
OK

AT+QSOSEND=0,10,1234562112                     //Send data
```

OK
+QSONMI=0,3,3BX   //Receive data

AT+QSODIS=0   //Disconnect socket
OK

AT+QSOCL=0   //Close socket
OK

6 General Commands

The AT command names listed below are temporary, and may be changed in the future release.

6.1. AT+QBAND   Query Current Operating Band

This command is used to query current modem NB-IoT operating band.

AT+QBAND Query Current Operating Band	
Read Command **AT+QBAND?**	Response **+QBAND: <Current Band>** **OK** If there is any error, response: **ERROR**
Test Command **AT+QBAND=?**	Response **+QBAND=**(list of supported **<Current Bands>**s) **OK**

Parameter

<Current Band>   Integer, valid values : 1, 2, 3, 5, 8, 12, 13, 17, 18, 19, 20, 26, 28, 66

6.2. AT+QSM   Control Sleep Mode

This command is used to control sleep mode .

AT+QSM Control Sleep Mode	
Write Command **AT+QSM=<mode>**	Response **OK**

| | If there is any error, response: |
| | **ERROR** |

Parameter

| **<mode>** | UNLOCK | Enable AP Enter Deep Sleep Mode . |

> **NOTE**
>
> UE should send this AT command to enable AP enter deep sleep mode. The setting will be saved in NVRAM.

6.3. AT+QRESET Reset UE

This command is used to reset the module immediately.

AT+QRESET Reset UE	
Write Command **AT+QRESET=<mode>**	Response **OK** If there is any error, response: **ERROR**
Test Command **AT+QRESET=?**	Response +QRESET: (list of supported <mode >s) **OK** If there is any error, response: **ERROR**

Parameter

| **<mode>** | 1 | Reset the UE |

6.4. AT+QSPCHSC Set Scrambling Algorithm

This command is used to select new or old scrambling code. This is because code has been updated by 3GPPP, and UE needs to select correct code for network.

AT+QSPCHSC Set Scrambling Algorithm	
Execution Command **AT+QSPCHSC=<mode>**	Response OK Otherwise response: **ERROR**
Test Command **AT+QSPCHSC=?**	Response **+QSPCHSC:** (list of supported **<mode>**s) **OK** Otherwise response: **ERROR**
Read Command **AT+QSPCHSC?**	Response **+QSPCHSC: <mode>** **OK** Otherwise response: **ERROR**

Parameter

<mode>	Integer 0: New algorithm (default) 1: Old algorithm

6.5. AT+QFRCLLCK Frequency & Cell Lock

This command is used to lock UE to specific frequency and optionally Cell ID.

AT+QFRCLLCK Frequency & Cell Lock	
Execution Command **AT+QFRCLLCK=<lock>[,<earfcn>,<ea**	Response OK

rfcn_offset>[,<pci>]]	
	Otherwise response: **ERROR**
Test Command **AT+QFRCLLCK=?**	Response **OK**
Read command **AT+QFRCLLCK?**	Response **OK**

Parameter

<lock>	Integer value indicating whether to activate lock, or remove lock:
	0 Remove lock
	1 Activate lock
<earfcn>	Integer value indicating requested EARFCN on which to lock. Range 0- 262143. Value of 0 indicates to remove any lock for EARFCN and Cell
<earfcn_offset>	Integer value indicating requested EARFCN offset:
	0 Offset of -2
	1 Offset of -1
	2 Offset of -0.5
	3 Offset of 0
	4 Offset of 1
<pci>	Integer value: Physical cell ID. Range: 0-503

6.6. AT+QNBIOTRAI   NB-IoT Release Assistance Indication

This command is used to set the NB-IoT release assistance indication as follows:

- No information available
- TE will send only 1 UL packet and no DL packet is expected
- TE will send only 1 UL packet and only 1 DL packet is expected

AT+QNBIOTRAI NB-IoT Release Assistance Indication	
Test Command **AT+QNBIOTRAI=?**	Response **+QNBIOTRAI: (list of supported <rai>s)** **OK**
Execution command **AT+QNBIOTRAI=<rai>**	Response **OK**

	Otherwise response: **ERROR**
Read Command **AT+QNBIOTRAI?**	Response **+QNBIOTRAI**: <rai> **OK**

Parameter

\<rai>	Integer type. Specifies release assistance information:
	   0 No information available (or none of the other options apply) (default)
	   1 TE will send only 1 UL packet and no DL packets expected
	   2 TE will send only 1 UL packet and only 1 DL packet expected

6.7. AT+QATWAKEUP   Enable Deep Sleep Wakeup Indication

This command is used to enable an unsolicited result code on a channel that indicates when the modem is fully woken up after a deep sleep.

AT+QATWAKEUP Enable Deep Sleep Wakeup Indication	
Test Command **AT+QATWAKEUP=?**	Response **+QATWAKEUP: (list of supported <enable>s)** **OK**
Execution Command **AT+QATWAKEUP=\<enable>**	Response **OK** Otherwise response: **ERROR**
Read Command **AT+QATWAKEUP?**	Response **+QATWAKEUP: \<enable>** **OK**

Parameter

\<enable>	Integer; <u>0</u> Disable indication on this channel when modem wakes up from deep sleep 1 Enable indication on this channel when modem wakes up from deep sleep

Example

AT+QATWAKEUP=1   //Enable wakeup indication

OK
(Modem goes into deep sleep mode)
(Modem is woken up from deep sleep mode)

+QATWAKEUP   //Modem fully woken up and ready to receive AT commands/data

6.8. AT+QENGINFO Query Current Network Status

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

AT+QENGINFO Query Current Network Status	
Write Command **AT+QENGINFO=\<mode>**	When \<mode>=0: Response **+QENGINFOSC:\<sc_earfcn>,\<sc_earfcn_offset>,\<sc_pci>,\<sc_cellid>,[\<sc_rsrp>],[\<sc_rsrq>],[\<sc_rssi>],[\<sc_snr>],\<sc_band>,\<sc_tac>,[\<sc_ecl>],[\<sc_tx_pwr>]** **[*MENGINFONC:\<nc_earfcn>,\<nc_earfcn_offset>,\<nc_pci>,\<nc_rsrp>,[…]]** **OK** When \<mode>=1: Response **+QENGINFODT:** **\<RLC_UL_BLER>,\<RLC_DL_BLER>,\<MAC_UL_BLER>,\<MAC_DL_BLER>,\<MAC_UL_total_bytes>,\<MAC_DL_total_bytes>,\<MAC_UL_total_HARQ_TX>,\<MAC_DL_total_HARQ_TX>,\<MAC_UL_HARQ_re_TX>,\<MAC_DL_HARQ_re_TX>,\<RLC_UL_tput>,\<RLC_DL_tput>,\<MAC_UL_tput>,\<MAC_DL_tput>** **OK** If there is any error, response: **ERROR**

Test Command	Response
AT+QENGINFO=?	+QENGINFO=(list of supported **<mode>**s)
	OK

Parameter

<mode>	Integer value indicating requested engineering information.
	0 display Radio information for serving and neighbor cells
	1 display data transfer information only if modem in RRC-CONNECTED state
<sc_earfcn>	Integer value indicating the EARFCN for serving cell. Range 0-262143
<sc_earfcn_offset>	Integer value indicating the EARFCN offset for serving cell:
	0 Offset of -2
	1 Offset of -1
	2 Offset of -0.5
	3 Offset of 0
	4 Offset of 1
<sc_pci>	Integer value indicating the serving cell physical cell ID. Range 0 – 503.
<sc_cellid>	String type; four byte (28 bit) cell ID in hexadecimal format for serving cell.
<sc_rsrp>	Signed integer indicating serving cell RSRP value in units of dBm (can be negative value). Available only in RRC-IDLE state.
<sc_rsrq>	Signed integer indicating serving cell RSRQ value in units of dB (can be negative value). Available only in RRC-IDLE state.
<sc_rssi>	Signed integer indicating serving cell RSSI value in units of dBm (can be negative value). Available only in RRC-IDLE state.
<sc_snr>	Signed integer value. Last SNR value for serving cell in units of dB. Available only in RRC-IDLE state.
<sc_band>	Integer value; current serving cell band
<sc_tac>	String type; two byte tracking area code (TAC) in hexadecimal format (e.g. "00C3" equals 195 in decimal).
<sc_ecl>	Integer value. Last Enhanced Coverage Level (ECL) value for serving cell. Range 0-2.
<sc_tx_pwr>	Signed integer value indicating current UE transmit power. Units of cBm Centibels relative to one milliwatt (can be negative value).
<nc_earfcn >	Integer value indicating the EARFCN for neighbor cell. Range 0-262143
<nc_earfcn_offset >	Integer value indicating the EARFCN offset for neighbor cell:
	0 Offset of -2
	1 Offset of -1
	2 Offset of -0.5
	3 Offset of 0
	4 Offset of 1
<nc_pci >	Integer value indicating the neighbor cell physical cell ID. Range 0-503.
<nc_rsrp>	Signed integer indicating neighbor cell RSRP value in units of dBm (can be negative value).

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

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

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

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

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

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

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

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

<MAC_UL_HARQ_re_TX> Integer value. Number of HARQ retransmissions for transport blocks on UL-SCH. Calculated from the beginning of successfully established /resumed / re-established RRC connection, or since previous

	AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: retransmissions
<MAC_DL_HARQ_re_TX>	Integer value. Number of HARQ retransmissions for transport blocks on DL-SCH, excluding BCCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRCCONNECTED state. Unit: retransmissions.
<RLC_UL_tput>	Integer value. RLC uplink throughput. Calculated over all established RLC AM radio bearers. Calculated from the beginning of successfully established / resumed RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: kbits / s
<RLC_DL_tput>	Integer value. RLC downlink throughput. Calculated over all established RLC AM radio bearers Calculated from the beginning of successfully established /resumed RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: kbits / s
<MAC_UL_tput>	Integer value. UL throughput in MAC for UL-SCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later.Available only in RRC-CONNECTED state. Unit: kbits / s
<MAC_DL_tput>	Integer value. DL throughput in MAC for DL-SCH, excluding BCCH. Calculated from the beginning of successfully established / resumed / re-established RRC connection, or since previous AT*MENGINFO query with <mode>=1, whichever is later. Available only in RRC-CONNECTED state. Unit: kbits / s

7 OneNET Commands

7.1. AT+MIPLCREATE   Create OneNET Instance

The command is used to create an instance of communication to NB-IoT OneNET platform.

AT+MIPLCREATE Create OneNET Instance	
Write Command **AT+MIPLCREATE=<totalsize>,<config>,<index>,<currentsize>,<flag>**	Response If the flag and index equal to 0, and the format is right, respond: Ref: id **OK** Else if the flag and index not equal to 0, and the format is right, respond: **OK** Until the configure bin input is successful, respond: Ref: id **OK** Otherwise response: **ERROR**

Parameter

<totalsize>	Integer. The length of complete configure bin. The range is 1-500.
<config>	The current configure bin, in hex string format.
<index>	Integer. The index number of the data. If the configure bin is too long, and more than the AT command length, should split the configure bin. If split into N part, The order number of <index> is N-1 to 0 in descending order, and the AT command is called in the order from the largest to the small sequence number, If the <index> is 0, means this is the last message of the configure bin.
<currentsize>	Integer. The length of current configure bin. The current size must equal to the length of current configure bin.

\<flag>	Integer. The message indication. The range is 0-2. If \<flag>=1, means the first message of the configure bin. If \<flag>=2, means the middle message ofthe configure bin. If \<flag>=0, means the last message of the configure bin.
\<Ref>	If configure the ONENET complete, will return a \<Ref: id>.

7.2. AT+MIPLDELETE Delete OneNET Instance

The command is used to delete an OneNET communication instance.

AT+MIPLDELETE Delete OneNET Instance	
Write Command **AT+MIPLDELETE=\<Ref>**	Response **OK** If there is an error related to ME functionality: **ERROR**

Parameter

\<Ref>	Reference ID of OneNET communication instance.

7.3. AT+MIPLADDOBJ Add LWM2M Object

The command is used to add LWM2M object.

AT+MIPLADDOBJ Add LWM2M Object	
Execution Command **AT+MIPLADDOBJ=\<Ref>,\<objId>,\<insCount>,\<insBitmap>,\<attrCount>,\<actCount>**	Response **OK** If there is an error related to ME functionality: **ERROR**

Parameter

\<Ref>	Reference ID of OneNET communication instance.
\<objId>	Integer. Object identifier. If the object ID is not existed, the module will return ERROR.
\<insCount>	Integer. Instance count.
\<insBitmap>	Instance bitmap. For example, if \<insCount>=3, and the \<insBitmap>=101, means the

instance id 0/2 will registered, and the instance id 1 will deregistered.

<attrCount>	Integer. Attribute count
<actCount>	Integer. Action count.

7.4. AT+MIPLDELOBJ   Delete LWM2M Object

The command is used to delete LWM2M object.

AT+MIPLDELOBJ Delete LWM2M Object	
Write Command **AT+MIPLDELOBJ=<Ref>,<objId>**	Response **OK** If there is an error related to ME functionality: **ERROR**

Parameter

<Ref>	Reference ID of OneNET communication instance.
<objId>	Integer. Object identifier. If the object ID is not existed, the module will return ERROR.

7.5. AT+MIPLOPEN   Send OneNET Login Request

The command is used to send login request to OneNET.

AT+MIPLOPEN Send OneNET Login Request	
Write Command **AT+MIPLOPEN =<Ref>**	Response If format is right, respond: **OK** If there is an error related to ME functionality: **ERROR**

Parameter

<Ref>	Reference ID of OneNET communication instance.

7.6. AT+MIPLCLOSE Send OneNET Logout Request

The command is used to send OneNET Logout Request.

AT+MIPLCLOSE Send OneNET Logout Request	
Write Command **AT+MIPLCLOSE=\<Ref>**	Response **OK** If there is an error related to ME functionality: **ERROR**

Parameter

\<Ref>	Reference ID of OneNET communication instance.

7.7. AT+MIPLREADRSP Respond the Read Request from Application Server/IoT Platform

The command is used to respond the read request from Application Server or IoT platform.

AT+MIPLREADRSP Respond the Read request from Application Server/IoT platform	
Write Command **AT+MIPLREADRSP=\<Ref>,\<msgID>,\<objID>,\<insID>,\<resID>,\<valuetype>,\<len>,\<value>,\<index>,\<flag>**	Response **OK** If there is an error related to ME functionality: **ERROR**

Parameter

\<Ref>	Reference ID of OneNET communication instance.
\<msgID>	Integer. The message identifier, which from the URC "+MIPLREAD:"
\<objID>	Integer. Object identifier.
\<insID>	Integer. The instance identifier, which from the URC "+MIPLREAD:"
\<resID>	Integer. The resource identifier, which from the URC "+MIPLREAD:".
\<valuetype>	Integer. The value type.
	1 String
	2 Opaque
	3 Integer

4.Float

5.Bool

\<len\>	Integer. The value length.
	When \<valuetype\> is String,\<len\> is the string length of \<value\>;
	When \<valuetype\> is Opaque,\<len\> is the hex string length of \<value\>;
	When \<valuetype\> is Integer/Float/Bool,\<len\> is 1;
\<value\>	The value.
	When \<valuetype\> is String, \<value\> is in string format; The string should be marked with double quotation marks.
	When \<valuetype\> is Opaque, \<value\> is in hex string format;
	When \<valuetype\> is Integer/Float/Bool, \<value\> is Integer/Float/Bool type text.
\<index\>	The index number of the data. If the data is too long, and more than the AT command length, should split the configure bin. If split into N part, The order number of \<index\> is N-1 to 0 in descending order, and the AT command is called in the order from the largest to the small sequence number, If the \<index\> is 0, means this is the last message of the data.
\<flag\>	The message indication. The range is 0-2. If \<flag\>=1, means the first message of the configure bin. If \<flag\>=2, means the middle message of the data. If \<flag\>=0, means the last message of the data.

7.8. AT+MIPLWRITERSP   Respond the Write Request from Application Server/IoT Platform

The command is used to respond the write request from the Application Server or IoT platform.

AT+MIPLWRITERSP Respond the Write Request from Application Server/IoT Platform	
Write Command **AT+MIPLWRITERSP=\<Ref\>,\<msgID\>, \<result\>,\<index\>**	Response **OK** If there is an error related to ME functionality: **ERROR**

Parameter

\<Ref\>	Reference ID of OneNET communication instance.
\<msgID\>	Integer. The message identifier, which from the URC "+MIPLREAD:"
\<result\>	The process result.
	0 FAIL
	1 SUCCESS

<index>	The index number of the data. If the data is too long, and more than the AT command length, should split the configure bin. If split into N part, The order number of <index> is N-1 to 0 in descending order, and the AT command is called in the order from the largest to the small sequence number, If the <index> is 0, means this is the last message of the data.

7.9. AT+MIPLEXECUTERSP   Respond the Execute Request from Application Server/IoT Platform

The command is used to respond the execute request from Application Server. or IoT platform.

AT+MIPLEXECUTERSP Respond the Execute Request from Application Server/IoT Platform	
Write Command **AT+MIPLEXECUTERSP=<Ref>,<msgID>,<result>**	Response **OK** If there is an error related to ME functionality: **ERROR**

Parameter

<Ref>	Reference ID of OneNET communication instance.
<msgID>	Integer. The message identifier, which from the URC "+MIPLREAD:"
<result>	The process result.
	0 FAIL
	1 SUCCESS

7.10. AT+MIPLOBSERVERSP   Respond the Observe Request from Application Server/IoT Platform

The command is used to respond the observe request from Application Server or IoT platform.

AT+MIPLOBSERVERSP Respond the Observe Request from Application Server/IoT Platform	
Write Command **AT+MIPLOBSERVERSP=<Ref>,<msgID>,<result>**	Response **OK** If there is an error related to ME functionality:

	ERROR

Parameter

<Ref>	Reference ID of OneNET communication instance.
<msgID>	Integer. The message identifier, which from the URC "+MIPLREAD:"
<result>	The process result.
	0 FAIL
	1 SUCCESS

7.11. AT+MIPLNOTIFY   Notify the Data to Application Server or IoT Platform

The command is used to notify the data to Application Server or IoT platform.

AT+MIPLNOTIFY Notify the Data to Application Server or IoT Platform	
Write Command **AT+MIPLNOTIFY=<Ref>,<msgID>,<ob jID>,<insID>,<resID>,<valuetype>,<le n>,<value>,<index>,<flag>**	Response **OK** If there is an error related to ME functionality: **ERROR**

Parameter

<Ref>	Reference ID of OneNET communication instance.
<msgID>	Integer. The message identifier, which from the URC "+MIPLREAD:"
<objID>	Integer. Object identifier.
<insID>	Integer. The instance identifier, which from the URC "+MIPLREAD:"
<resID>	Integer. The resource identifier, which from the URC "+MIPLREAD:".
<valuetype>	Integer. The value type.
	1 String
	2 Opaque
	3 Integer
	4 Float
	5 Bool
<len>	Integer. The value length.
	When <valuetype> is String,<len> is the string length of <value>;
	When <valuetype> is Opaque,<len> is the hex string length of <value>;

When <valuetype> is Integer/Float/Bool,<len> is 1;

<value>   The value.

When <valuetype> is String, <value> is in string format; The string should be marked with double quotation marks.

When <valuetype> is Opaque, <value> is in hex string format;

When <valuetype> is Integer/Float/Bool, <value> is Integer/Float/Bool type text.

<index>   The index number of the data. If the data is too long, and more than the AT command length, should split the configure bin. If split into N part, The order number of <index> is N-1 to 0 in descending order, and the AT command is called in the order from the largest to the small sequence number, If the <index> is 0, means this is the last message of the data.

<flag>   The message indication. The range is 0-2. If <flag>=1, means the first message of the configure bin. If <flag>=2, means the middle message of the data. If <flag>=0, means the last message of the data.

7.12. AT+MIPLUPDATE   Send Update Request

The command is used to send Update Request to update lifetime and objects.

AT+MIPLUPDATE Send Update Request	
Write Command **AT+MIPLUPDATE=<ref>,<lifetime>,<** **withObjectFlag>**	Response **OK** If there is an error related to ME functionality: **ERROR**

Parameter

<Ref>   Reference ID of OneNET communication instance.
<lifetime>   Update lifetime value, unit is millisecond. The value of 0 means not to update lifetime.
<withObjectFlag>   Update with objects list flag.
      0   Update not with objects list.
      1   Update with objects list.

7.13. AT+MIPLVER   Query the OneNET Version

The command is used to query the current OneNET protocol version.

AT+MIPLVER Query the OneNET Version	
Read Command **AT+MIPLVER?**	Response **+MIPLVER: <version>** **OK**

Parameter

<version>   The OneNET protocol version.

8 Appendix A Reference

Table 2: Terms and Abbreviations

Abbreviation	Description
APN	Access Point Name
CDP	Connected Device Platform
CS	Circuit Switched
DCE	Data Communication Equipment
EARFCN	E-UTRA Absolute Radio Frequency Channel Number
eDRX	Extended Discontinuous Reception
EGPRS	Enhanced General Packet Radio Service
GERAN	GSM/EDGE Radio Access Network
GMT	Greenwich Mean Time
GPRS	General Packet Radio Service
HPLMN	Home Public Land Mobile Network
HSDPA	High Speed Downlink Packet Access
HSUPA	High-Speed Uplink Packet Access
ICMP	Internet Control Messages Protocol
IMEI	International Mobile Equipment Identity
IMEISV	International Mobile Equipment Identity and Software Version
MS	Mobile Station
NB-IoT	Narrow Band Internet of Thing

PDP	Packet Data Protocol
RRC	Radio Resource Control
RTC	Real Time Clock
SVN	Software Version Number
TA	Terminal Adapter
TCP	Transmission Control Protocol
TE	Terminal Equipment
TTL	Time To Live
UDP	User Datagram Protocol
UE	User Equipment
UICC	Universal Integrated Circuit Card
URC	Unsolicited Result Code
UUID	Universally Unique Identifier