BC95 OneNET

Application Note

NB-IoT Module Series

Rev. BC95_OneNET_Application_Note_V1.0

Date: 2018-01-18

Status: Released

www.quectel.com

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

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2018-01-18	Gary TANG/ Arnold ZHAO	Initial

Contents

Table Index

1 Introduction

LWM2M (Lightweight Machine to Machine) is a secure, efficient and deployable client-server protocol for managing resource constrained devices on a variety of networks. LWM2M uses a modern architectural design based on REST, defines an extensible resource and data model and reuses and builds on an efficient secure data transfer standard called the Constrained Application Protocol (CoAP). LWM2M is a profile for device services based on CoAP (RFC 7252). LWM2M defines a simple object model and a number of interfaces and operations for device management.

This document mainly introduces how to use the LWM2M function of Quectel BC95 module through AT commands to connect to the OneNET platform.

NOTE

Only BC95 modules with firmware version (queried by **ATI**) ended with “*_ONT*” support communication with OneNET platform.

2 OneNET Data Interaction Mechanism

This chapter gives the data interaction mechanism of OneNET platform.

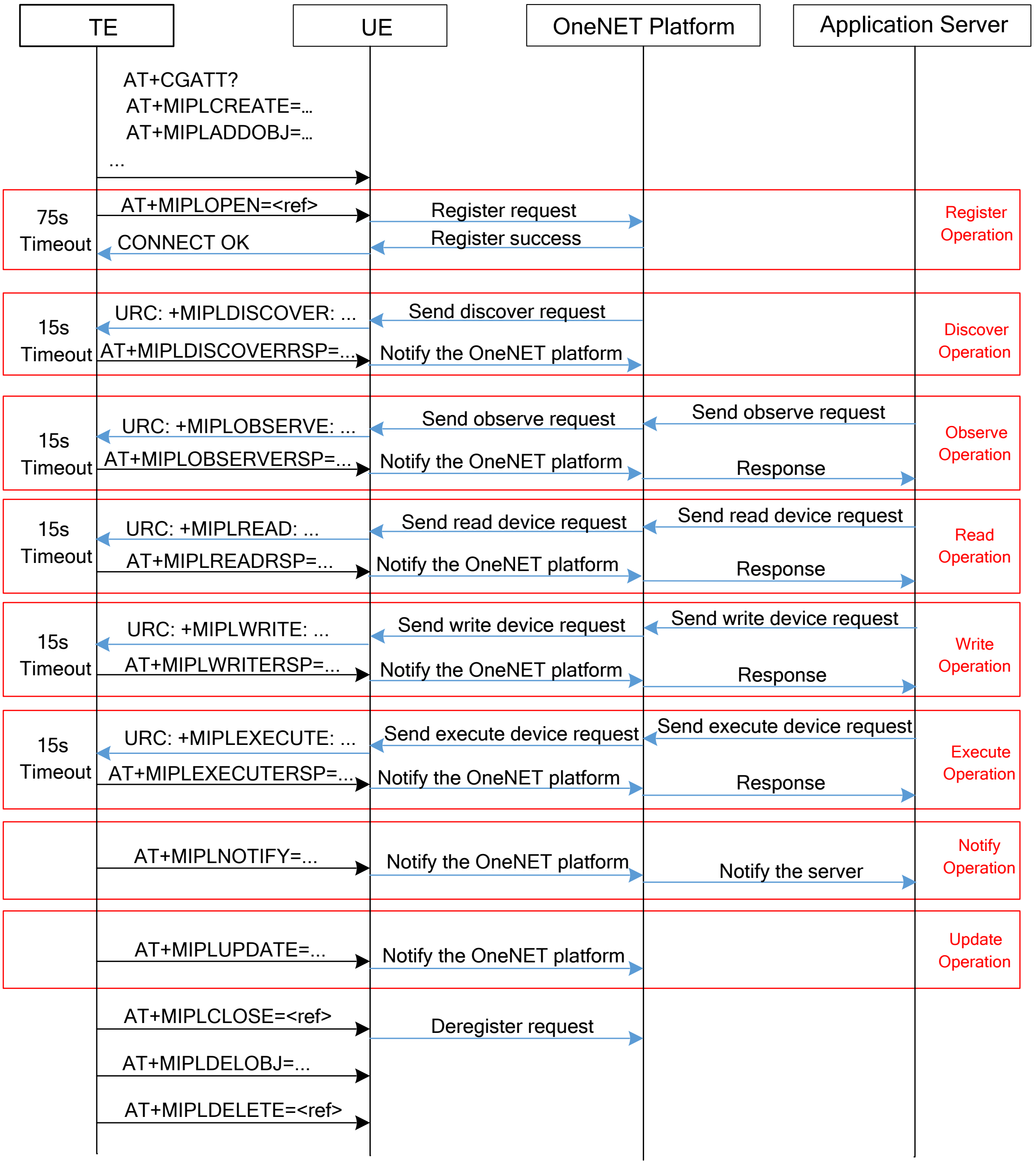

Figure 1: OneNET Data Interaction Diagram

3 OneNET Related AT Commands

This chapter presents the AT commands for operating OneNET function.

3.1. AT Command Syntax

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 UE.

3.2. Description of OneNET Related AT Commands

3.2.1. AT+MIPLCREATE Create a OneNET Communication Suite Instance

The command is used to create an instance of OneNET communication suite.

AT+MIPLCREATE Create a OneNET Communication Suite Instance	
Test Command **AT+MIPLCREATE=?**	Response **+MIPLCREATE: <totalsize>,<config>,<index>,<currentsize>,<flag>** **OK**
Write Command **AT+MIPLCREATE=<totalsize>,<config>,<index>,<currentsize>,<flag>**	Response If **<flag>** and **<index>** equal to 0, and the format is right: **ref: <ref>** **OK**

	If **<flag>** and **<index>** are not equal to 0, and the format is right: **OK** Until the configure bin input is successful: **ref: <ref>** **OK** If there is any error: **ERROR**

Parameter

<totalsize>	The length of complete configure bin. The range is 1-500.
<config>	The current configure bin, in hex string format.
<index>	The index number of the data. If the configure bin is too long, and exceeds the AT command length, then the configure bin should be split into several parts. If it is split into N parts, then 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 smallest number. If **<index>** is 0, it means this is the last message of the configure bin.
<currentsize>	The length of current configure bin.
<flag>	The message indication. The range is 0-2. If **<flag>**=1, it means the first message of the configure bin. If **<flag>**=2, it means the middle message of the configure bin. If **<flag>**=0, it means the last message of the configure bin.
<ref>	Instance ID of OneNET communication suite. If configuration of OneNET is completed, **ref: <ref>** will be returned.

3.2.2. AT+MIPLDELETE Delete a OneNET Communication Suite Instance

The command is used to delete an OneNET communication suite instance.

AT+MIPLDELETE Delete a OneNET Communication Suite Instance	
Test Command **AT+MIPLDELETE=?**	Response **+MIPLDELETE: <ref>** **OK**
Write Command **AT+MIPLDELETE=<ref>**	Response **OK** If there is any error: **ERROR**

Parameter

<ref>	Instance ID of OneNET communication suite.

3.2.3. AT+MIPLVER Query the Current OneNET Communication Suite Version

The command is used to query the current OneNET communication suite version.

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

Parameter

<version>	The current OneNET communication suite version .

3.2.4. AT+MIPLADDOBJ Add a LWM2M Object

The command is used to add a LWM2M object.

AT+MIPLADDOBJ Add a LWM2M Object	
Test Command **AT+MIPLADDOBJ=?**	Response **+MIPLADDOBJ: <ref>,<objId>,<insCount>,<insBitmap>,<attrCount>,<actCount>** **OK**
Write Command **AT+MIPLADDOBJ=<ref>,<objId>,<insCount>,<insBitmap>,<attrCount>,<actCount>**	Response **OK** If there is any error: **ERROR**

Parameter

<ref>	Instance ID of OneNET communication suite.
<objId>	Object identifier. If the object ID is not existed, the module will return **ERROR**.
<insCount>	Instance count.
<insBitmap>	Instance bitmap. For example, if **<insCount>**=3, and the **<insBitmap>**=101, it

	means the instance ID 0/2 will be registered, and the instance ID 1 will be deregistered.
<attrCount>	Attribute count.
<actCount>	Action count.

3.2.5. AT+MIPLDELOBJ Delete a LWM2M Object

The command is used to delete a LWM2M object.

AT+MIPLDELOBJ Delete a LWM2M Object	
Test Command **AT+MIPLDELOBJ=?**	Response **+MIPLDELOBJ: <ref>,<objId>** **OK**
Write Command **AT+MIPLDELOBJ=<ref>,<objId>**	Response **OK** If there is any error: **ERROR**

Parameter

<ref>	Instance ID of OneNET communication suite.
<objId>	Object identifier. If the object ID is not existed, the module will return **ERROR**.

3.2.6. AT+MIPLOPEN Send a Register Request

The command is used to send a register request to OneNET platform.

AT+MIPLOPEN Send a Register Request	
Test Command **AT+MIPLOPEN=?**	Response **+MIPLOPEN: <ref>** **OK**
Write Command **AT+MIPLOPEN=<ref>**	Response If the format is right: **OK** If there is any error: **ERROR** And then if the registration is successful, respond:

	CONNECT OK Otherwise respond: **CONNECT FAIL**
Maximum Response Time	75s

Parameter

<ref>	Instance ID of OneNET communication suite.

3.2.7. AT+MIPLCLOSE Send a Deregister Request

The command is used to send a deregister request to OneNET platform.

AT+MIPLCLOSE Send a Deregister Request	
Test Command **AT+MIPLCLOSE=?**	Response **+MIPLCLOSE: <ref>** **OK**
Write Command **AT+MIPLCLOSE=<ref>**	Response **OK** If there is any error: **ERROR**

Parameter

<ref>	Instance ID of OneNET communication suite.

3.2.8. AT+MIPLDISCOVERRSP Respond the Discover Request

The command is used to respond the discover request from OneNET platform.

AT+MIPLDISCOVERRSP Respond the Discover Request	
Test Command **AT+MIPLDISCOVERRSP=?**	Response **+MIPLDISCOVERRSP: <ref>,<msgid>,<length>,<valuestring>** **OK**
Write Command	Response

AT+MIPLDISCOVERRSP=<ref>,<msgid>,<length>,<valuestring>	**OK** If there is any error: **ERROR**
URC Format: **+MIPLDISCOVERRSP: TIMEOUT(<msgId>)**	If TE cannot respond correctly in 15 seconds, the UE returns the unsolicited result code.

Parameter

<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier, which comes from the URC "+ MIPLDISCOVER:".
<length>	The length of **<valuestring>**.
<valuestring>	A string which includes the attributes of the object and should be marked with double quotation marks. Each attribute should be split with comma, such as "1101,1102,1103".

3.2.9. AT+MIPLOBSERVERSP Respond the Observe Request

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

AT+MIPLOBSERVERSP Respond the Observe Request	
Test Command **AT+MIPLOBSERVERSP=?**	Response **+MIPLOBSERVERSP: <ref>,<msgId>,<result>** **OK**
Write Command **AT+MIPLOBSERVERSP=<ref>,<msgId>,<result>**	Response **OK** If there is any error: **ERROR**
URC Format: **+MIPLOBSERVERSP: TIMEOUT(<msgId>)**	If TE cannot respond correctly in 15 seconds, the UE returns the unsolicited result code.

Parameter

<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier, which comes from the URC "+ MIPLOBSERVE:".
<result>	The process result. 0 FAIL 1 SUCCESS

3.2.10. AT+MIPLREADRSP Respond the Read Request

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

AT+MIPLREADRSP Respond the Read Request	
Test Command **AT+MIPLREADRSP=?**	Response **+MIPLREADRSP: <ref>,<msgId>,<objId>,<insId>,<resId>,<valueType>,<len>,<value>,<index>,<flag>** **OK**
Write Command **AT+MIPLREADRSP=<ref>,<msgId>,<objId>,<insId>,<resId>,<valueType>,<len>,<value>,<index>,<flag>**	Response **OK** If there is any error: **ERROR**
URC Format: **+MIPLREADRSP: TIMEOUT(<msgId>)**	If TE cannot respond correctly in 15 seconds, the UE returns the unsolicited result code.

Parameter

<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier, which comes from the URC "+MIPLREAD:".
<objId>	Object identifier.
<insId>	The instance identifier, which comes from the URC "+MIPLREAD:".
<resId>	The resource identifier, which comes from the URC "+MIPLREAD:".
<valueType>	The value type. 1 String 2 Opaque 3 Integer 4 Float 5 Bool
<len>	The value length. When **<valueType>** is String, it is the string length of **<value>**. When **<valueType>** is Opaque, it is the hex string length of **<value>**. When **<valueType>** is Integer/Float/Bool, it is 1.
<value>	The value. When **<valueType>** is String, it is in string format, and the string should be marked with double quotation marks. When **<valueType>** is Opaque, it is in hex string format. When **<valueType>** is Integer/Float/Bool, it is an Integer/Float/Bool type text.
<index>	The index number of the data. If the data is too long, and exceeds the AT command length, then the configure pin should be split into several parts. If it is split into N

	parts, 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 smallest number. If **<index>** is 0, it means this is the last message of the data.
<flag>	The message indication. The range is 0-2. If **<flag>**=1, it means the first message of the configure bin. If **<flag>**=2, it means the middle message of the data. If **<flag>**=0, it means the last message of the data.

3.2.11. AT+MIPLWRITERSP Respond the Write Request

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

AT+MIPLWRITERSP Respond the Write Request	
Test Command **AT+MIPLWRITERSP=?**	Response **+MIPLWRITERSP: <ref>,<msgId>,<result>,<index>** **OK**
Write Command **AT+MIPLWRITERSP=<ref>,<msgId>,<result>,<index>**	Response **OK** If there is any error: **ERROR**

Parameter

<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier, which comes from the URC "+MIPLWRITE:".
<result>	The process result. 0 FAIL 1 SUCCESS
<index>	The index number of the data.

3.2.12. AT+MIPLEXECUTERSP Respond the Execute Request

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

AT+MIPLEXECUTERSP Respond the Execute Request	
Test Command **AT+MIPLEXECUTERSP=?**	Response **+MIPLEXECUTERSP: <ref>,<msgId>,<result>** **OK**
Write Command **AT+MIPLEXECUTERSP=<ref>,<msgId**	Response **OK**

>,<result>	If there is any error: **ERROR**

Parameter

<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier, which comes from the URC “+MIPLEXECUTE:”.
<result>	The process result. 0 FAIL 1 SUCCESS

3.2.13. AT+MIPLNOTIFY Notify the Data to OneNET Platform or Application Server

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

AT+MIPLNOTIFY Notify the Data to OneNET Platform or Application Server	
Test Command **AT+MIPLNOTIFY=?**	Response **+MIPLNOTIFY: <ref>,<msgId>,<objId>,<insId>,<resId><valueType>,<len>,<value>,<index>,<flag>** **OK**
Write Command **AT+MIPLNOTIFY=<ref>,<msgId>,<objId>,<insId>,<resId>,<valueType>,<len>,<value>,<index>,<flag>**	Response **OK** If there is any error: **ERROR**

Parameter

<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier, which comes from the URC “+MIPLOBSERVE:”.
<objId>	Object identifier.
<insId>	The instance identifier, which comes from the URC “+MIPLOBSERVE :”.
<resId>	The resource identifier, which comes from the URC “+MIPLOBSERVE :”.
<valueType>	The value type. 1 String 2 Opaque 3 Integer 4 Float 5 Bool
<len>	The value length. When **<valueType>** is String, it is the string length of **<value>**.

When **<valueType>** is Opaque, it is the hex string length of **<value>**.
When **<valueType>** is Integer/Float/Bool, it is 1.

<value> The value.
When **<valueType>** is String, it is in string format, and the string should be marked with double quotation marks.
When **<valueType>** is Opaque, it is in hex string format.
When **<valueType>** is Integer/Float/Bool, it is an Integer/Float/Bool type text.

<index> The index number of the data. If the data is too long, and exceeds the AT command length, the configure bin should be split into several parts. If it is split into N parts, 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 smallest number. If **<index>** is 0, it means this is the last message of the data.

<flag> The message indication. The range is 0-2. 3. If **<flag>**=1, it means the first message of the configure bin. If **<flag>**=2, it means the middle message of the data. If **<flag>**=0, it means the last message of the data.

3.2.14. AT+MIPLUPDATE Send Update Request

The command is used to send an update request to update lifetime and objects.

AT+MIPLUPDATE Send Update Request	
Test Command **AT+MIPLUPDATE=?**	Response **+MIPLUPDATE: <ref>,<lifetime>,<withObjectFlag>[,<autoUpdateFlag>]** **OK**
Write Command **AT+MIPLUPDATE=<ref>,<lifetime>,<withObjectFlag>[,<autoUpdateFlag>]**	Response **OK** If there is any error: **ERROR**

Parameter

<ref> Instance ID of OneNET communication suite.

<lifetime> Updated lifetime value. The range is 60-4294967294. Unit: second.

<withObjectFlag> Whether to update with objects list.
0 Update without objects list.
1 Update with objects list.

<autoUpdateFlag> Whether to update automatically.
0 Update manually by TE.
1 Update automatically by module, and the update period is lifetime*0.8.

3.2.15. AT+MIPLOBLIST Query the Current Observe List

The command is used to query the current observe list.

AT+MIPLOBLIST Query the Current Observe List	
Read Command **AT+MIPLOBLIST?**	Response **+MIPLOBLIST: <index>,<ref>,<msgId>,<objId>,<insId>,<resId>** **<index>,<ref>,<msgId>,<objId>,<insId>,<resId>** **…** **<index>,<ref>,<msgId>,<objId>,<insId>,<resId>** **OK**

Parameter

<index>	The index of the observe list information. The range is 1-10. The module supports 10 observation requests.
<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier.
<objId>	Object identifier.
<insId>	The instance identifier. If observe all resources under **<insId>**, the value of **<resId>** is -1.
<resId>	The resource identifier.

4 OneNET Related URCs

This chapter gives OneNET related URCs and their descriptions.

Table 2: OneNET Related URCs

Index	URC	Description
[1]	**+MIPLDISCOVER: <ref>,<msgId>,<objId>**	When the OneNET platform sends a discover request, the module will report the URC when it receives the request.
[2]	**+MIPLOBSERVE: <ref>,<msgId>,<flag>,<objId>[,<insId>[,<resId>]]**	When the OneNET platform or Application Server sends an observe request, the module will report the URC when it receives the request.
[3]	**+MIPLREAD: <ref>,<msgId>,<objId>,<insId>,<resId>**	When the OneNET platform or Application Server sends a read request, the module will report the URC when it receives the request.
[4]	**+MIPLWRITE: <ref>,<msgId>,<objId>,<insId>,<resId>,<valueType>,<len>,<value>,<index>**	When the OneNET platform or Application Server sends a write request, the module will report the URC when it receives the request.
[5]	**+MIPLEXECUTE: <ref>,<msgId>,<objId>,<insId>,<resId>**	When the OneNET platform or Application Server sends an execute request, the module will report the URC when it receives the request.

4.1. “+MIPLDISCOVER” URC to Notify the TE to Respond the Discover Request

The URC is mainly used to notify the TE to respond the discover request from OneNET platform.

“+MIPLDISCOVER” URC to Notify the TE to Respond the Discover Request	
URC Format: **+MIPLDISCOVER: <ref>,<msgId>,<objId>**	Notify the TE to respond the discover request from OneNET platform.

Parameter

<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier of packet.
<objId>	The object identifier that received from OneNET platform.
<insId>	The instance identifier that received from OneNET platform.

4.2. “+MIPLOBSERVE” URC to Notify the TE an Observe Request

The URC is mainly used to notify the TE that there is an observe request from OneNET platform or Application Server.

“+MIPLOBSERVE” URC to Notify the TE an Observe Request	
URC Format: **+MIPLOBSERVE: <ref>,<msgId>,<flag><objId>[,<insId>[,<resId>]]**	Notify the TE that there is an observe request from OneNET platform or Application Server.

Parameter

<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier of packet.
<flag>	Indicates whether or not to observe. 0 Observe 1 Cancel observe
<objId>	The object identifier that received from OneNET platform or Application Server.
<insId>	The instance identifier that received from OneNET platform or Application Server.
<resId>	The resource identifier that received from OneNET platform or Application Server.

4.3. “+MIPLREAD” URC to Notify the TE to Respond the Read Request

The URC is mainly used to notify the TE to respond the read request from OneNET platform or Application server.

“+MIPLREAD” URC to Notify the TE to Respond the Read Request	
URC Format: **+MIPLREAD: <ref>,<msgId>,<objId>,<insId>,<resId>**	Notify the TE to respond the read request from OneNET platform or Application Server.

Parameter

<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier of packet.
<objId>	The object identifier that received from OneNET platform or Application Server.
<insId>	The instance identifier that received from OneNET platform or Application Server.
<resId>	The resource identifier that received from OneNET platform or Application Server.

4.4. “+MIPLWRITE” URC to Notify the TE to Respond the Write Request

The URC is mainly used to notify the TE to respond the write request from OneNET platform or Application Server.

“+MIPLWRITE” URC to Notify the TE to Respond the Write Request	
URC Format: **+MIPLWRITE: <ref>,<msgId>,<objId>,<insId>,<resId>,<valueType>,<len>,<value>,<index>**	Notify the TE to respond the write request from OneNET platform or Application Server.

Parameter

<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier of packet.
<objId>	The object identifier that received from OneNET platform or Application Server.
<insId>	The instance identifier that received from OneNET platform or Application Server.
<resId>	The resource identifier received from OneNET platform or Application Server.
<valueType>	The value type (only shows in opaque currently). 1 String 2 Opaque 3 Integer 4 Float 5 Bool
<len>	The value length.
<value>	The value that received from OneNET platform or Application Server, in hex string format.

<index>	The index number of the data. If the data is too long, then OneNET platform may split the data. If the data is split into N parts, the order number of **<index>** is N-1 to 0 in descending order. If the **<index>** is 0, it means this is the last message of the data.

4.5. "+MIPLEXECUTE" URC to Notify the TE to Respond the Execute Request

The URC is mainly used to notify the TE to respond the execute request from OneNET platform or Application Server.

"+MIPLEXECUTE" URC to Notify the TE to Respond the Execute Request	
URC Format: **+MIPLEXECUTE: <ref>,<msgId>,<objId>,<insId>,<resId>**	Notify the TE to respond the execute request from OneNET platform or Application Server.

Parameter

<ref>	Instance ID of OneNET communication suite.
<msgId>	The message identifier of packet.
<objId>	The object identifier that received from OneNET platform or Application Server.
<insId>	The instance identifier that received from OneNET platform or Application Server.
<resId>	The resource identifier that received from OneNET platform or Application Server.

5 Examples

This chapter gives the examples to explain how to use OneNET related AT commands.

5.1. Example of OneNET Operation with Register and Discover Operations

```
//Configure a communication suite instance with a single command.
AT+MIPLCREATE=?
+MIPLCREATE: <lifetime>,<totalsize>,<configbin>,<index>,<currentsize>,<flag>

OK
AT+MIPLCREATE=51,130033f10003f2002105001100000000000000d3138332e3233302e34302e343
000044e554c4cf3000cea00c800044e554c4c,0,51,0
ref: 1                      //Configured the communication suite instance successfully.

OK

//Add a LWM2M object.
AT+MIPLADDOBJ=?
+MIPLADDOBJ: <ref>,<objId>,<insCount>,<insBitmap>,<attrCount>,<actCount>

OK
AT+MIPLADDOBJ=1,3311,1, 1,4,2
OK                       //Added the object successfully. And the instance ID 0 will be registered.
AT+MIPLADDOBJ=1,3304,6,101101,4,2
OK                       //Added the object successfully. And the instance ID 0/2/3/5 will be registered, and 1/4
                           will not be registered.
AT+MIPLOPEN=?
+MIPLOPEN: <ref>

OK

//Send a register request to the OneNET platform.
AT+MIPLOPEN=1
OK
```

```
CONNECT OK         //If registered to OneNET platform successfully.

CONNECT FAIL       //If failed to register to OneNET platform.

+MIPLDISCOVER: 1,29622,3311
AT+MIPLDISCOVERRSP=1,29622,19,"5850,5851,5706,5805"
OK

//If TE cannot respond correctly in 15 seconds, the UE returns the following unsolicited result code:
+MIPLDISCOVERRSP: TIMEOUT(29622)

//Send a deregister request to OneNET platform.
AT+MIPLCLOSE=?
+MIPLCLOSE: <ref>

OK
AT+MIPLCLOSE=1
OK

//Delete a communication suite instance.
AT+MIPLDELETE=?
+MIPLDELETE: <ref>

OK
AT+MIPLDELETE=1
OK                       //Deleted the communication suite instance successfully.
```

5.2. Example of OneNET Operation with Read Request

```
//Send a register request to the OneNET platform
AT+MIPLOPEN=1
OK

CONNECT OK                        //Registered to the OneNET platform successfully.

+MIPLREAD: 1,28587,3303,0,1       //The Application Server has sent a read request to the module, and
                                   wants to read the object ID (3303), instance ID (0) and resource ID (1)
                                  values.

//Respond the read request.
AT+MIPLREADRSP=?
+MIPLREADRSP:<ref>,<msgId>,<objId>,<insId>,<resId>,<valueType>,<len>,<value>,<index>,<flag
>
```

```
OK
AT+MIPLREADRSP=1,28587,3303,0,1,2,6,303132333435,0,0
OK                                 //Sent data 303132333435 to server successfully.

//If TE cannot respond correctly in 15 seconds, the UE returns the following unsolicited result code:
+MIPLREADRSP: TIMEOUT(28587)

//Send a deregister request to the OneNET platform.
AT+MIPLCLOSE=?
+MIPLCLOSE: <ref>

OK
AT+MIPLCLOSE=1
OK

//Delete a LWM2M object.
AT+MIPLDELOBJ=?
+MIPLDELOBJ: <ref>,<objId>

OK
AT+MIPLDELOBJ=1,3303
OK
AT+MIPLDELOBJ=1,3304
OK
AT+MIPLDELOBJ=1,3305
ERROR                        //The object does not exist in the communication suite instance.

//Delete a communication suite instance.
AT+MIPLDELETE=1
OK                           //Deleted the communication suite instance successfully.
```

5.3. Example of OneNET Operation with Write Request

```
//Send a register request to the OneNET platform.
AT+MIPLOPEN=1
OK

CONNECT OK                                   //Registered to the OneNET platform successfully.

+MIPLWRITE: 1,28606,3304,2,1,2,2,3039,0     //The Application Server has sent a write request to the
                                              module, and wants to read the object ID (3304), instance
                                              ID (2), resource ID (1), and the received value in hex
                                              string (3039).
```

```
//Respond the write request.
AT+MIPLWRITERSP=?
+MIPLWRITERSP: <ref>,<msgId>,<result>, <index>

OK
AT+MIPLWRITERSP=1,28606,1,0
OK                    //Send response to the server with result (1), which means “SUCCESS”.

//If TE cannot respond correctly in 15 seconds, the UE returns the following unsolicited result code:
+MIPLWRITERSP: TIMEOUT(28606)

//Send a deregister request to the OneNET platform.
AT+MIPLCLOSE=?
+MIPLCLOSE: <ref>

OK
AT+MIPLCLOSE=1
OK

//Delete a LWM2M object.
AT+MIPLDELOBJ=?
+MIPLDELOBJ: <ref>,<objId>

OK
AT+MIPLDELOBJ=1,3303
OK
AT+MIPLDELOBJ=1,3304
OK

//Delete a communication suite instance.
AT+MIPLDELETE=1
OK                    //Deleted the communication suite instance successfully.
```

5.4. Example of OneNET Operation with Execute Request

```
//Send a register request to the OneNET platform.
AT+MIPLOPEN=1
OK

CONNECT OK                              //Registered to the OneNET platform successfully.

+MIPLEXECUTE: 1,28607,3304,4,5700      //The Application Server has sent an execute request to the
                                         module.
```

```
//Respond the execute request.
AT+MIPLEXECUTERSP=?
+MIPLEXECUTERSP: <ref>,<msgId>,<result>

OK
AT+MIPLEXECUTERSP=1,28607,2
OK                      //Send response to the server with result (2), which means “SUCCESS”.

//If TE cannot respond correctly in 15 seconds, the UE returns the following unsolicited result code:
+MIPLEXECUTERSP: TIMEOUT(28607)

//Send a deregister request to the OneNET platform.
AT+MIPLCLOSE=?
+MIPLCLOSE: <ref>

OK
AT+MIPLCLOSE=1
OK

//Delete a LWM2M object.
AT+MIPLDELOBJ=?
+MIPLDELOBJ: <ref>,<objId>

OK
AT+MIPLDELOBJ=1,3303
OK
AT+MIPLDELOBJ=1,3304
OK

//Delete a communication suite instance.
AT+MIPLDELETE=1
OK              //Deleted the communication suite instance successfully.
```

5.5. Example of OneNET Operation with Notify Rrequest

```
//Send a register request to the OneNET platform.
AT+MIPLOPEN=1
OK

CONNECT OK                          //Registered to the OneNET platform successfully.

+MIPLOBSERVE: 1,29620,0,3311,0      //The Application Server has sent an observe request to the
                                     module.
```

```
AT+MIPLOBSERVERSP=1,29620,1
OK

//If TE cannot respond correctly in 15 seconds, the UE returns the following unsolicited result code:
+MIPLOBSERVERSP: TIMEOUT(29620)

//Query the current observe list.
AT+MIPLOBLIST?
+MIPLOBLIST: 1,1,29620,3311,0,-1

OK

//Respond the notify request.
AT+MIPLNOTIFY=?
+MIPLNOTIFY: <ref>,<msgId>,<objId>,<insId>,<resId>,<valueType>,<len>,<value>,<index>,<flag>

OK
AT+MIPLNOTIFY=1,29620,3311,0,1,2,6,303132333435,0,0
OK                          //Notified data 303132333435 to server successfully.

//Send a deregister request to the OneNET platform.
AT+MIPLCLOSE=?
+MIPLCLOSE: <ref>

OK
AT+MIPLCLOSE=1
OK

//Delete an object for communication suite instance.
AT+MIPLDELOBJ=?
+MIPLDELOBJ: <ref>,<objId>

OK
AT+MIPLDELOBJ=1,3311
OK

//Delete a communication suite instance.
AT+MIPLDELETE=1
OK               //Deleted the communication suite instance successfully.
```

6 Appendix A References

Table 3: Related Documents

SN	Document Name	Remark
[1]	Quectel_BC95_AT_Commands_Manual	BC95 AT Commands Manual
[2]	IPSO-Smart-Objects-Starter-Pack	Internet Protocol for Smart Objects (IPSO) Alliance
[3]	OMA-TS-LightweightM2M-V1_0	Open Mobile Alliance

Table 4: Terms and Abbreviations

Abbreviation	Description
LWM2M	Lightweight Machine to Machine
CoAP	Constrained Application Protocol
URC	Unsolicited Result Code
NB-IoT	Narrow Band Internet of Things
REST	Representational state transfer
TE	Terminal Equipment (Typically the MCU)
UE	User Equipment (Typically the Module)