

BC660K-GL&BC950K-GL

LwM2M Application Note

NB-IoT Module Series

Version: 1.2

Date: 2024-12-05

Status: Released



At Quectel, our aim is to provide timely and comprehensive services to our customers. If you require any assistance, please contact our headquarters:

Quectel Wireless Solutions Co., Ltd.

Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local offices. For more information, please visit:

<http://www.quectel.com/support/sales.htm>.

For technical support, or to report documentation errors, please visit:

<http://www.quectel.com/support/technical.htm>.

Or email us at: support@quectel.com.

Legal Notices

We offer information as a service to you. The provided information is based on your requirements and we make every effort to ensure its quality. You agree that you are responsible for using independent analysis and evaluation in designing intended products, and we provide reference designs for illustrative purposes only. Before using any hardware, software or service guided by this document, please read this notice carefully. Even though we employ commercially reasonable efforts to provide the best possible experience, you hereby acknowledge and agree that this document and related services hereunder are provided to you on an “as available” basis. We may revise or restate this document from time to time at our sole discretion without any prior notice to you.

Use and Disclosure Restrictions

License Agreements

Documents and information provided by us shall be kept confidential, unless specific permission is granted. They shall not be accessed or used for any purpose except as expressly provided herein.

Copyright

Our and third-party products hereunder may contain copyrighted material. Such copyrighted material shall not be copied, reproduced, distributed, merged, published, translated, or modified without prior written consent. We and the third party have exclusive rights over copyrighted material. No license shall be granted or conveyed under any patents, copyrights, trademarks, or service mark rights. To avoid ambiguities, purchasing in any form cannot be deemed as granting a license other than the normal non-exclusive, royalty-free license to use the material. We reserve the right to take legal action for noncompliance with abovementioned requirements, unauthorized use, or other illegal or malicious use of the material.

Trademarks

Except as otherwise set forth herein, nothing in this document shall be construed as conferring any rights to use any trademark, trade name or name, abbreviation, or counterfeit product thereof owned by Quectel or any third party in advertising, publicity, or other aspects.

Third-Party Rights

This document may refer to hardware, software and/or documentation owned by one or more third parties ("third-party materials"). Use of such third-party materials shall be governed by all restrictions and obligations applicable thereto.

We make no warranty or representation, either express or implied, regarding the third-party materials, including but not limited to any implied or statutory, warranties of merchantability or fitness for a particular purpose, quiet enjoyment, system integration, information accuracy, and non-infringement of any third-party intellectual property rights with regard to the licensed technology or use thereof. Nothing herein constitutes a representation or warranty by us to either develop, enhance, modify, distribute, market, sell, offer for sale, or otherwise maintain production of any our products or any other hardware, software, device, tool, information, or product. We moreover disclaim any and all warranties arising from the course of dealing or usage of trade.

Privacy Policy

To implement module functionality, certain device data are uploaded to Quectel's or third-party's servers, including carriers, chipset suppliers or customer-designated servers. Quectel, strictly abiding by the relevant laws and regulations, shall retain, use, disclose or otherwise process relevant data for the purpose of performing the service only or as permitted by applicable laws. Before data interaction with third parties, please be informed of their privacy and data security policy.

Disclaimer

- a) We acknowledge no liability for any injury or damage arising from the reliance upon the information.
- b) We shall bear no liability resulting from any inaccuracies or omissions, or from the use of the information contained herein.
- c) While we have made every effort to ensure that the functions and features under development are free from errors, it is possible that they could contain errors, inaccuracies, and omissions. Unless otherwise provided by valid agreement, we make no warranties of any kind, either implied or express, and exclude all liability for any loss or damage suffered in connection with the use of features and functions under development, to the maximum extent permitted by law, regardless of whether such loss or damage may have been foreseeable.
- d) We are not responsible for the accessibility, safety, accuracy, availability, legality, or completeness of information, advertising, commercial offers, products, services, and materials on third-party websites and third-party resources.

Copyright © Quectel Wireless Solutions Co., Ltd. 2024. All rights reserved.

About the Document

Revision History

Version	Date	Author	Description
-	2022-04-11	Randy LI	Creation of the document
1.0	2022-06-23	Randy LI	First official release
1.1	2023-04-25	Yance YANG/ Randy LI	Added the applicable module BC950K-GL.
1.2	2024-12-05	Carter ZHOU	1. Updated the declaration of AT command examples (Chapter 4.2). 2. Updated IPv4 addresses in examples (Chapter 4.3.1 and 5.2).

Contents

About the Document.....	3
Contents.....	4
Table Index.....	6
Figure Index.....	7
1 Introduction	8
2 General Overview of LwM2M	9
3 LwM2M Implementation	11
3.1. LwM2M Interfaces.....	11
3.1.1. Bootstrap Interface.....	11
3.1.2. Client Registration Interface.....	11
3.1.3. Device Management and Service Enablement Interface	12
3.1.4. Information Reporting Interface	14
3.2. LwM2M Object	14
3.2.1. Built-in Object.....	14
3.2.2. Custom Object	15
4 Description of LwM2M AT Commands	16
4.1. AT Command Introduction	16
4.1.1. Definitions.....	16
4.1.2. AT Command Syntax	16
4.2. Declaration of AT Command Examples	17
4.3. AT Commands Description	17
4.3.1. AT+QLWCONFIG Configure Registration Parameters	17
4.3.2. AT+QLWREG Send Registration Request.....	19
4.3.3. AT+QLWUPDATE Send Update Request.....	19
4.3.4. AT+QLWDEREG Send Deregister Request	21
4.3.5. AT+QLWADDOBJ Add LwM2M Object.....	21
4.3.6. AT+QLWDELOBJ Delete LwM2M Object	23
4.3.7. AT+QLWRDRSP Respond to Read Request.....	24
4.3.8. AT+QLWRRSP Respond to Write Request	25
4.3.9. AT+QLWEXERSP Respond to Execute Request	26
4.3.10. AT+QLWOBSRSP Respond to Observe Request	27
4.3.11. AT+QLWNOTIFY Report Data to Server.....	28
4.3.12. AT+QLWSTATUS Query Current LwM2M Status	29
4.3.13. AT+QLWRECOVER Manually Trigger LwM2M Context Recovery Process	30
4.4. LwM2M Related URCs	31
4.4.1. +QLWURC: "ping" Notify TE of Update Operation Result	32
4.4.2. +QLWURC: "write" Notify TE to Respond to Write Request.....	33
4.4.3. +QLWURC: "read" Notify TE to Respond to Read Request.....	33
4.4.4. +QLWURC: "execute" Notify TE to Respond to Execute Request	34

4.4.5.	+QLWURC: "observe" Notify TE to Respond to Observe Request	34
4.4.6.	+QLWURC: "bootstrapping" Notify TE that Bootstrap Phase Starts	35
4.4.7.	+QLWURC: "bs_finished" Notify TE that Bootstrap Phase Has Completed.....	35
4.4.8.	+QLWURC: "registering" Notify TE that Register Phase Starts.....	35
4.4.9.	+QLWURC: "report" Notify TE that CON Data Has Been Sent	36
4.4.10.	+QLWURC: "report_ack" Notify TE that CON Data Has Been Acked	36
4.4.11.	+QLWURC: "lifetime_changed" Notify TE of Change in Registration Lifetime.....	36
4.4.12.	+QLWURC: "binding_changed" Notify TE of Change in Binding Node	37
4.4.13.	+QLWURC: "recovered" Notify TE of LwM2M Context Recovery Result	37
5	Examples	39
5.1.	Manual Registration to LwM2M Server	39
5.2.	Custom Object Related Operations.....	40
5.3.	Registration to Leshan LwM2M Server	42
5.4.	LwM2M Updates (Leshan Server).....	45
5.4.1.	LwM2M Server Update Operation	45
5.4.2.	LwM2M Client Update Operation.....	46
5.5.	LwM2M Custom Object Related Operations (Leshan Server).....	50
5.5.1.	Observe Operation	50
5.5.2.	Notify Operation	51
5.5.3.	Read Operation.....	51
5.5.4.	Execute Operation	51
6	Error Codes	52
7	Appendix References	54

Table Index

Table 1: Types of AT Commands	16
Table 2: LwM2M Related URCs	31
Table 3: Updateable Parameters	46
Table 4: Summary of LwM2M <err>	52
Table 5: Summary of <status_code>	52
Table 6: References	54
Table 7: Terms and Abbreviations	54

Figure Index

Figure 1: Overall Architecture of LwM2M Enabler	9
Figure 2: Overall Architecture of 3GPP CIoT	10
Figure 3: Client Registration Interface Operation Flow	12
Figure 4: Example Data Flow of Device Management	13
Figure 5: Example Data Flow of Service Enablement	13
Figure 6: Example Data Flow of Information Reporting	14
Figure 7: Click "Add new client security configuration" in SECURITY Tab	42
Figure 8: Create "New security configuration" on Leshan Server	43
Figure 9: Check Created "New security configuration" on Leshan Server	43
Figure 10: Online Client on Leshan Server	44
Figure 11: "Read" Operation on Leshan Server	45
Figure 12: "Write" Operation on Leshan Server	45
Figure 13: "Execute" Operation on Leshan Server	46
Figure 14: Update Flow Example	47
Figure 15: Update Registration	48
Figure 16: Update Binding Mode	49
Figure 17: Update Object and Object Instance List	50

1 Introduction

OMA Lightweight M2M (LwM2M) is a device management protocol designed for sensor networks and for the demands of Machine-to-Machine (M2M) environments. The protocol, designed for remote management of M2M devices and related service enablement, features a modern architectural design based on REST, defines an extensible resource and data model and builds on an efficient secure data transfer standard called the Constrained Application Protocol (CoAP).

Quectel NB-IoT BC660K-GL and BC950K-GL modules support LwM2M V1.0. This document introduces the LwM2M protocol and architecture, and how to use this feature with BC660K-GL and BC950K-GL modules. It also describes the LwM2M-related AT commands set supported on the module.

2 General Overview of LwM2M

The OMA LwM2M enabler defines an application layer communication protocol between an LwM2M Server and an LwM2M Client as well as between an LwM2M Bootstrap Server and an LwM2M Client. An LwM2M Device includes an LwM2M Client component. The enabler includes device management and service enablement for LwM2M Devices. The target LwM2M Devices for this enabler are mainly resource constrained devices. Therefore, this enabler makes use of lightweight and compact protocol mechanisms, as well as an efficient resource data model.

Four interfaces are designed among the three entities, as shown in the architecture below:

- Bootstrap
- Client Registration
- Device Management and Service Enablement
- Information Reporting

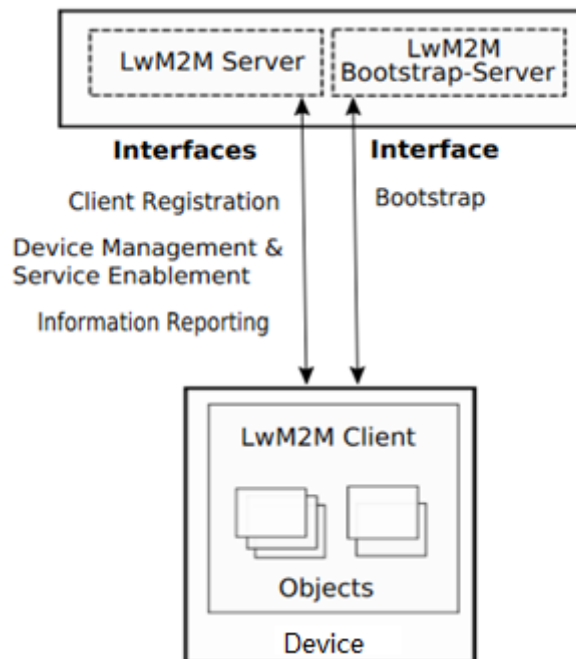


Figure 1: Overall Architecture of LwM2M Enabler

The figure below illustrates the overall architecture of 3GPP CIoT, which is provided to better understand the LwM2M protocol.

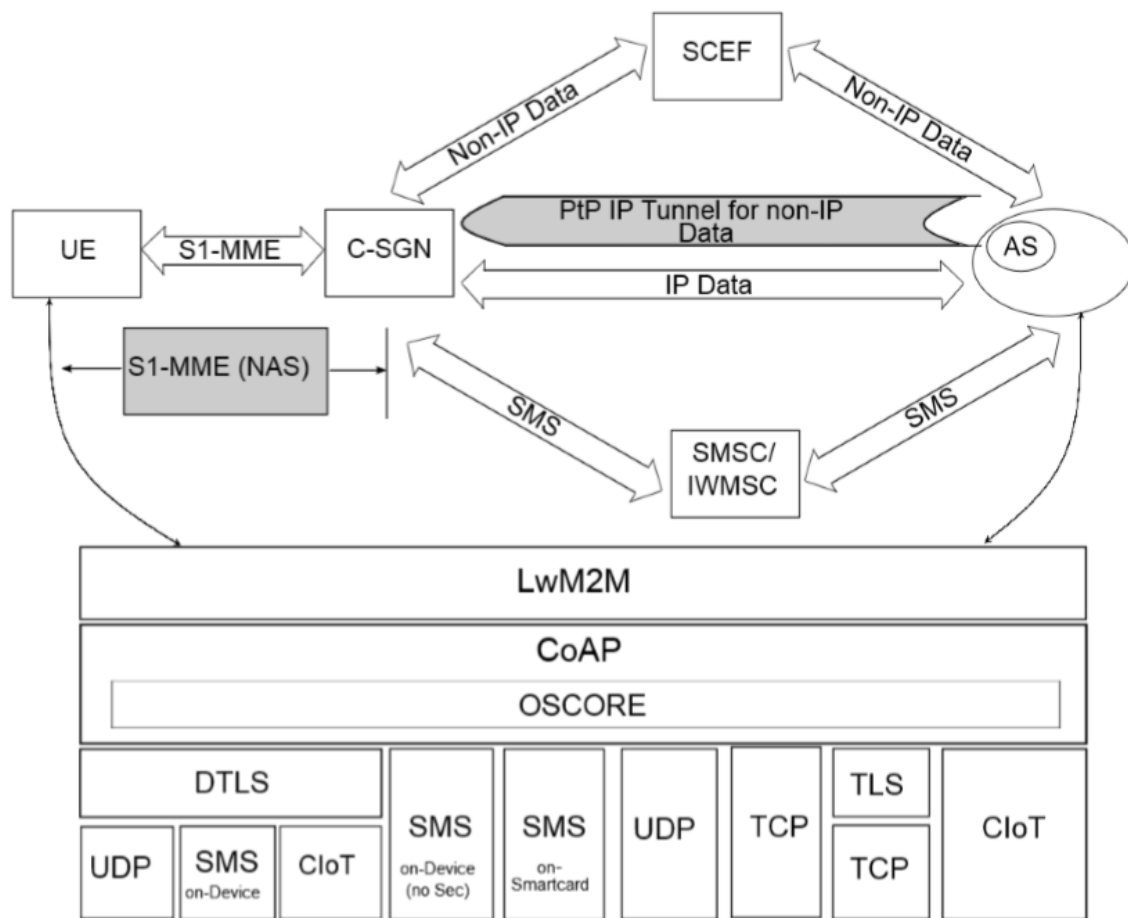


Figure 2: Overall Architecture of 3GPP CIoT

BC660K-GL and BC950K-GL support LwM2M Client only. MCU can communicate with LwM2M Server through LwM2M AT commands described in **Chapter 4.3**.

NOTE

OMA specifies a set of LwM2M protocol specifications. You can refer to the following links for more details:

- <https://www.omaspecworks.org/what-is-oma-specworks/iot/lightweight-m2m-lwm2m/>
- http://www.openmobilealliance.org/release/LightweightM2M/V1_1-20180710-A/OMA-TS-LightweightM2M_Core-V1_1-20180710-A.pdf

3 LwM2M Implementation

3.1. LwM2M Interfaces

3.1.1. Bootstrap Interface

The bootstrap interface is used to provision essential information into the LwM2M Client to enable the LwM2M Client to "register" to one or more LwM2M Servers.

- **Bootstrap Modes Supported on BC660K-GL and BC950K-GL**

There are four bootstrap modes supported by the LwM2M protocol: Factory Bootstrap, Bootstrap from Smartcard, Client Initiated Bootstrap and Server Initiated Bootstrap. Generally, the LwM2M Client must support at least one bootstrap mode specified in the bootstrap server. Currently, BC660K-GL and BC950K-GL support three bootstrap modes: Factory Bootstrap, Client Initiated Bootstrap and Server Initiated Bootstrap.

- **Server and Access Control Configurations**

The Bootstrap Server or LwM2M Server can be configured with **AT+QLWCONFIG**. **AT+QLWREG** can be used to register to a dedicated LwM2M Server. During the Client Initiated Bootstrap phase, if disconnection or failure occurs, back-off mechanism will be triggered internally.

3.1.2. Client Registration Interface

The client registration interface is used by an LwM2M Client to register to one or more LwM2M Servers, and to maintain registration to or de-registration from the LwM2M Server.

- **Register**

The LwM2M Client performs a "Register" operation to provide essential information required for registration to the LwM2M Server, such as endpoint name, PSK, registration lifetime, supported objects/instance etc.

- **Update**

After registration, the LwM2M Client performs an "Update" operation in either a manual mode or an automatic mode periodically before registration lifetime expires. If the LwM2M Server does not receive an update from the LwM2M Client within lifetime, the LwM2M Server will remove the last registration

and enter off state.

- **Deregister**

The LwM2M Client performs a "Deregister" operation to deregister from the LwM2M Server.

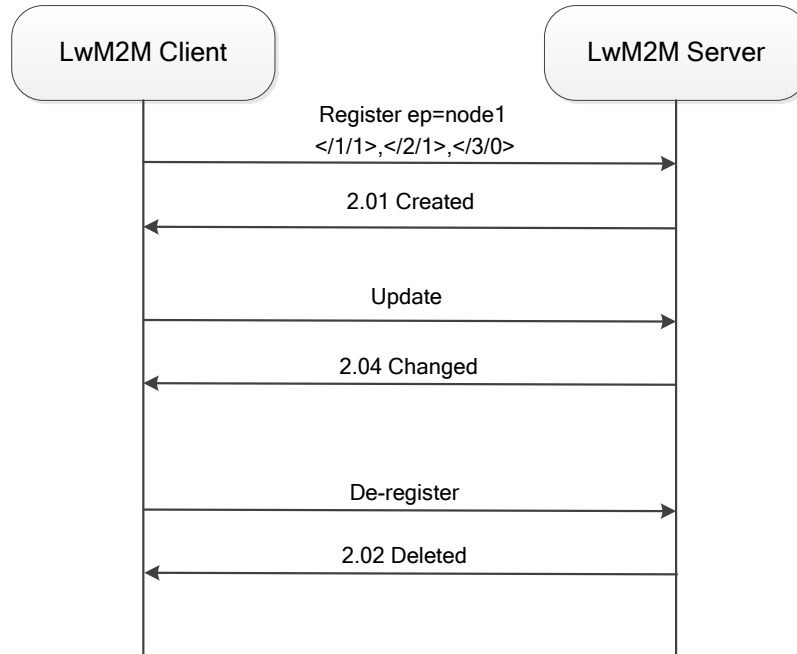


Figure 3: Client Registration Interface Operation Flow

3.1.3. Device Management and Service Enablement Interface

Device management and service enablement interface is a very important interface in LwM2M protocol.

- Allow LwM2M Server to access object instances and resources available on the LwM2M Client.
- Allow LwM2M Server to perform "Create", "Read", "Write", "Delete", "Execute", "Write Attributes", or "Discover" operations on the object instances and resources.
- Allow the resource related operations to be defined in object definition using the Object Template.

The data flow of device management and service enablement is shown as below:

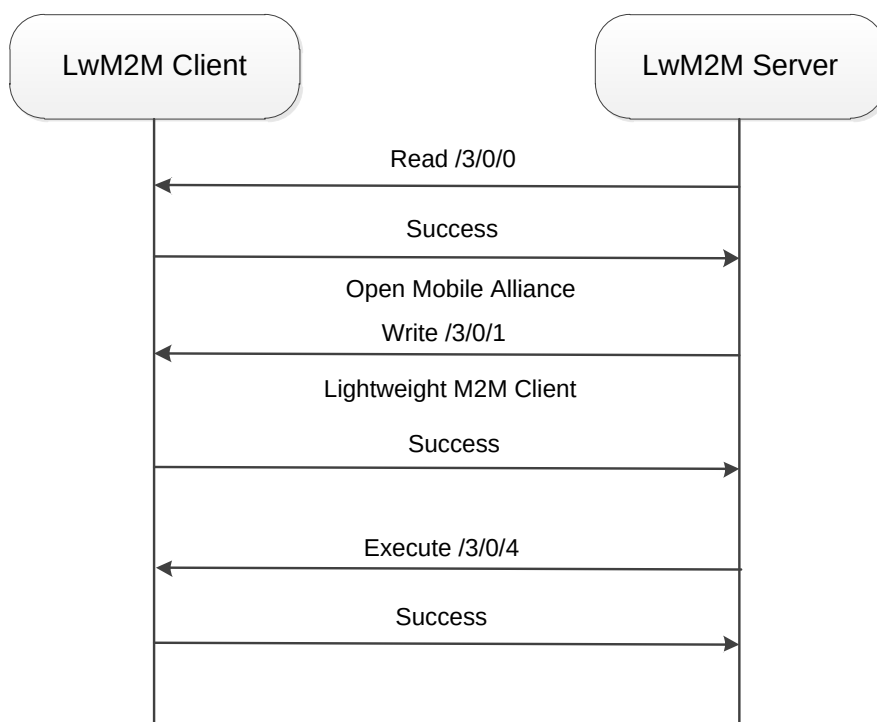


Figure 4: Example Data Flow of Device Management

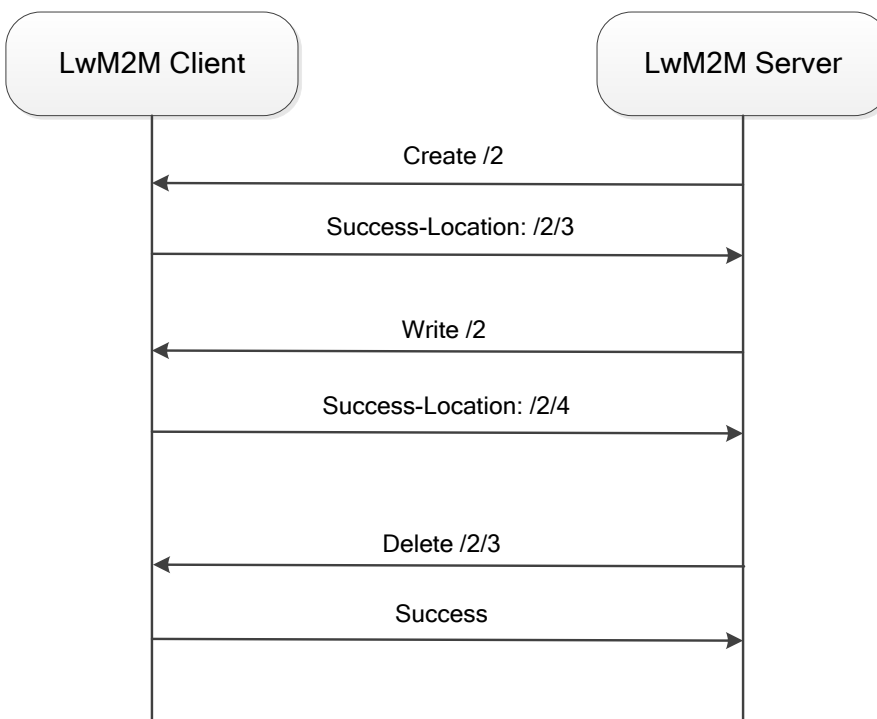


Figure 5: Example Data Flow of Service Enablement

3.1.4. Information Reporting Interface

The information reporting interface is used by an LwM2M Server to observe any change in a resource on the LwM2M Client, and to receive notifications when new values are available.

- The observation relationship is initiated by sending an "Observe" operation to LwM2M Client for an Object, an Object Instance or a Resource.
- An observation ends when a "Cancel Observation" operation is performed by the LwM2M Server.

BC660K-GL and BC950K-GL LwM2M Clients support observation and notification of objects, object instances and resources.

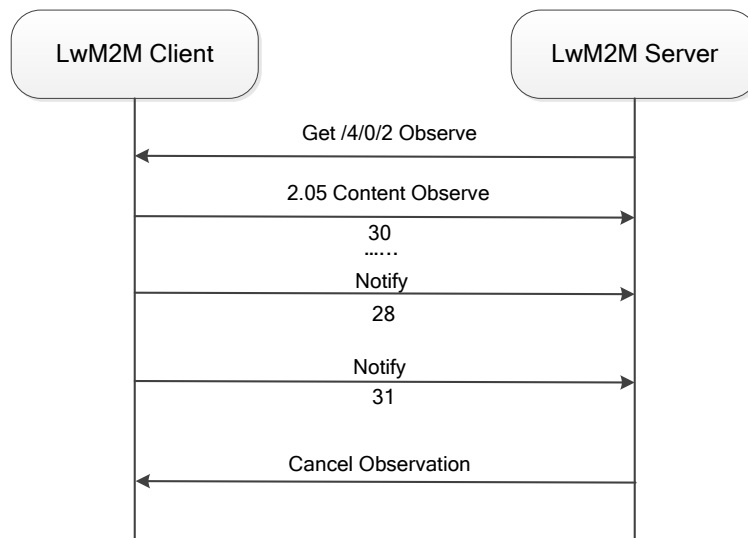


Figure 6: Example Data Flow of Information Reporting

3.2. LwM2M Object

3.2.1. Built-in Object

There are some built-in objects supported by BC660K-GL and BC950K-GL LwM2M Clients by default. The number in parentheses indicates the object ID.

- Security object (0)
- Server object (1)
- Access control object (2)
- Device object (3)
- Connectivity monitoring object (4)
- Firmware update object (5)
- Connectivity statistics object (7)
- Cellular connectivity object (10)

- APN connection profile object (11)
- Portfolio object (16)

3.2.2. Custom Object

BC660K-GL and BC950K-GL also support users to add custom objects through **AT+QLWADDOBJ**. Currently, each instance can be added with up to 7 resources.

```
//LwM2M Client has registered to LwM2M Server.
//Add a custom object.
AT+QLWADDOBJ=3303,0,7,5601,5602,5603,5604,5605,5700,5701
OK

+QLWADDOBJ: 0 //0 means that the custom object is added successfully.

+QLWURC: "ping",0,102,0 //Object list updated successfully.
AT+QLWADDOBJ? //Query custom object and object instance list.
+QLWADDOBJ: 3303,0

OK
```

BC660K-GL and BC950K-GL custom objects only support "Observe", "Read", "Write", and "Execute" operations. See **Chapter 5.5** for operation examples.

NOTE

In order to prevent BC660K-GL and BC950K-GL from entering deep sleep during data interaction, it is recommended that the MCU should actively disable sleep mode of the module with **AT+QSCLK=0** before data interaction and then enable sleep mode with **AT+QSCLK=1** after data interaction is completed. See **document [1]** for details of the command.

4 Description of LwM2M AT Commands

4.1. AT Command Introduction

4.1.1. Definitions

- **<CR>** Carriage return character.
- **<LF>** Line feed character.
- **<...>** Parameter name. Angle brackets do not appear on the command line.
- **[...]** Optional parameter of a command or an optional part of TA information response. Square brackets do not appear on the command line. When an optional parameter is not given in a command, the new value equals to its previous value or the default settings, unless otherwise specified.
- **Underline** Default setting of a parameter.

4.1.2. AT Command Syntax

All command lines must start with **AT** or **at** and end with **<CR>**. Information responses and result codes always start and end with a carriage return character and a line feed character: **<CR><LF><response><CR><LF>**. In tables presenting commands and responses throughout this document, only the commands and responses are presented, and **<CR>** and **<LF>** are deliberately omitted.

Table 1: Types of AT Commands

Command Type	Syntax	Description
Test Command	AT+<cmd>=?	Test the existence of the corresponding command and return information about the type, value, or range of its parameter.
Read Command	AT+<cmd>?	Check the current parameter value of the corresponding command.
Write Command	AT+<cmd>=<p1>[,<p2>[,<p3>[...]]]	Set user-definable parameter value.
Execution Command	AT+<cmd>	Return a specific information parameter or perform a specific action.

4.2. Declaration of AT Command Examples

The AT command examples in this document are provided to help you learn about the use of AT commands introduced herein. The examples, however, should not be taken as Quectel's recommendation or suggestions about how you should design a program flow or what status you should set the module into. Sometimes multiple examples may be provided for one AT command. However, this does not mean that there exists a correlation among these examples, or that they should be executed in a given sequence. The URLs, domain names, IP addresses, usernames/accounts, and passwords (if any) in the AT command examples are provided for illustrative and explanatory purposes only, and they should be modified to reflect your actual usage and specific needs.

4.3. AT Commands Description

4.3.1. AT+QLWCONFIG Configure Registration Parameters

This command configures parameters required for registration of LwM2M client to LwM2M server.

AT+QLWCONFIG Configure Registration Parameters	
Write Command AT+QLWCONFIG=<server_type>,<serverIP>,<port>,<endpoint_name>,<lifetime>,<security_mode>[,<PSK_ID>,<PSK>]	Response OK If there is any error: +CMEE ERROR: <err>
Read Command AT+QLWCONFIG?	Response When <server_type>=0 is configured, or when <server_type>=1 is configured but no Bootstrap or DM server is connected: +QLWCONFIG: <server_type>,<serverIP>,<port>,<endpoint_name>,<lifetime>,<security_mode>[,<PSK_ID>,<PSK>] OK When <server_type>=1 is configured and both Bootstrap and DM servers are connected: +QLWCONFIG: <server_type>,<serverIP>,<port>,<endpoint_name>,<lifetime>,<security_mode>[,<PSK_ID>,<PSK>] +QLWCONFIG: <server_type>,<serverIP>,<port>,<endpoint_name>,<lifetime>,<security_mode>[,<PSK_ID>,<PSK>]

	OK
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. Remains valid after deep-sleep wakeup. The configurations are saved to NVRAM automatically.

Parameter

<server_type>	Integer type. Server type. 0 DM server 1 Bootstrap server 2 Repository server 3 Diagnostics server
<serverIP>	String type. IP address or domain name of the LwM2M server. The maximum size is 150 bytes.
<port>	Integer type. Port number of LwM2M server. Range: 0–65535.
<endpoint_name>	String type. Endpoint name of an LwM2M device. The maximum size is 150 bytes.
<lifetime>	Integer type. Lifetime of registration. Range: 50–31536000 (365 days). Unit: second. If lifetime ≤ 100 s, then the Actual Update Interval = 30 + (lifetime - 50) × 4/5 If lifetime ≤ 300 s, then the Actual Update Interval = 70 + (lifetime - 100) × 9/10 If lifetime > 300 s, then the Actual Update Interval = 250 + (lifetime - 300) × 19/20 When the Actual Update Interval timer expires, the module automatically reports data, resets registration lifetime and updates the Actual Update Interval.
<security_mode>	Integer type. Security mode. 0 Secure mode of PRE_SHARED_KEY 3 No security
<PSK_ID>	String type. PSK identifier. The maximum size is 150 bytes.
<PSK>	String type. PSK content. It must be a hexadecimal string with an even number of characters. The maximum size is 256 bytes.
<err>	Error code. For details of error codes, see Chapter 6 .

Example

```

AT+QLWCONFIG=0,"192.0.2.2",8007,"866971030000710",86400,0,"866971030000710","77c7
55bd8cbb400abf2f0c420eed2cb0"
OK
AT+QLWCONFIG?
+QLWCONFIG: 0,"192.0.2.2",8007,"866971030000710",86400,0,"866971030000710","77c755bd8cb
b400abf2f0c420eed2cb0"
OK

```

4.3.2. AT+QLWREG Send Registration Request

This command sends a registration request to LwM2M server.

AT+QLWREG Send Registration Request	
Execution Command AT+QLWREG	Response OK +QLWREG: <status_code> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	-

Parameter

<status_code>	Integer type. For details of status codes, see Chapter 6 .
<err>	Error code. For details of error codes, see Chapter 6 .

Example

```

AT+QLWREG
OK

+QLWREG: 0           //Registered to LwM2M server successfully.

```

4.3.3. AT+QLWUPDATE Send Update Request

This command sends an update request to the LwM2M server.

AT+QLWUPDATE Send Update Request	
Write Command AT+QLWUPDATE=<mode>,<lifetime/ binding_mode>[,<server_type>]	Response +QLWUPDATE: <messageID> OK +QLWUPDATE: <status_code>,<messageID> If there is any error: +CMEE ERROR: <err>

Maximum Response Time	300 ms
Characteristics	-

Parameter

<mode>	Integer type. 0 Updates lifetime 1 Updates binding mode
<lifetime>	Integer type. Lifetime of registration. Range: 50–31536000 (365 days). Unit: second. If lifetime ≤ 100 s, then the Actual Update Interval = 30 + (lifetime - 50) × 4/5 If lifetime ≤ 300 s, then the Actual Update Interval = 70 + (lifetime - 100) × 9/10 If lifetime > 300 s, then the Actual Update Interval = 250 + (lifetime - 300) × 19/20 When the Actual Update Interval timer times out, the module automatically reports data, resets registration lifetime and updates the Actual Update Interval.
<binding_mode>	Integer type. The transport binding mode configured for the LwM2M client. 0 UDP mode 1 UDP & Queue mode
<server_type>	Integer type. Server type. 0 DM server 2 Repository server 3 Diagnostics server
<messageID>	Integer type. Message ID.
<status_code>	Integer type. For details of status codes, see Chapter 6 .
<err>	Error code. For details of error codes, see Chapter 6 .

Example

```

AT+QLWUPDATE=0,1000 //Update lifetime.
+QLWUPDATE: 59797

OK

+QLWUPDATE: 0,59797 //Updated successfully.
AT+QLWUPDATE=1,1 //Update binding mode as UDP & Queue mode.
+QLWUPDATE: 61459

OK

+QLWUPDATE: 0, 61459 //Updated successfully.

```

4.3.4. AT+QLWDEREG Send Deregister Request

This command sends a deregister request to the LwM2M server.

AT+QLWDEREG Send Deregister Request	
Execution Command AT+QLWDEREG	Response OK +QLWDEREG: <status_code> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	-

Parameter

<status_code>	Integer type. For details of status codes, see Chapter 6 .
<err>	Error code. For details of error codes, see Chapter 6 .

Example

```

AT+QLWDEREG           //Send a deregister request.
OK

+QLWDEREG: 0          //De-registered successfully.

```

4.3.5. AT+QLWADDOBJ Add LwM2M Object

This command adds a new LwM2M object.

AT+QLWADDOBJ Add LwM2M Object	
Write Command AT+QLWADDOBJ=<objectID>,<instanceID>,<res_num>,<resourceID>	Response OK +QLWADDOBJ: <err> If there is any error: +CMEE ERROR: <err>
Read Command AT+QLWADDOBJ?	Response +QLWADDOBJ: <objectID>,<instanceID>

	OK If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. Remain valid after deep-sleep wakeup. The configurations are saved to NVRAM automatically.

Parameter

<objectID>	Integer type. Object ID.
<instanceID>	Integer type. Instance ID.
<res_num>	Integer type. Number of resources.
<resourceID>	Integer type. Resource ID.
<err>	Error code. For details of error codes, see Chapter 6 .

NOTE

Currently, each instance can define up to 7 resources.

Example

//A custom object can be added after successful registration to LwM2M server or before the registration is initiated.

AT+QLWADDOBJ=3303,0,3,5601,5602,5603 //Add an object 3303 and an instance ID 0. There are 3 resources 5601, 5602 and 5603 configured.

OK

+QLWADDOBJ: 0

AT+QLWADDOBJ=3303,1,3,5601,5602,5603

OK

+QLWADDOBJ: 0

AT+QLWADDOBJ=3303,2,3,5601,5602,5603

OK

+QLWADDOBJ: 0

AT+QLWADDOBJ=3303,3,3,5601,5602,5603

OK

+QLWADDOBJ: 0

```
AT+QLWADDOBJ?
+QLWADDOBJ: 3303,0
+QLWADDOBJ: 3303,1
+QLWADDOBJ: 3303,2
+QLWADDOBJ: 3303,3

OK
```

4.3.6. AT+QLWDELOBJ Delete LwM2M Object

This command deletes a specified LwM2M object.

AT+QLWDELOBJ Delete LwM2M Object	
Write Command AT+QLWDELOBJ=<objectID>	Response OK +QLWDELOBJ: <err> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	The command takes effect immediately. Remain valid after deep-sleep wakeup. The configurations will be saved to NVRAM automatically.

Parameter

<objectID>	Integer type. Object ID.
<err>	Error code. For details of error codes, see Chapter 6 .

Example

```
AT+QLWDELOBJ=17 //Delete the object with object ID 17.
OK

+QLWDELOBJ: 0
```

4.3.7. AT+QLWRDRSP Respond to Read Request

This command allows the TE to respond to a read request from server.

AT+QLWRDRSP Respond to Read Request	
Write Command AT+QLWRDRSP=<messageID>,<result>,<objectID>,<instanceID>,<resourceID>,<value_type>,<len>,<value>,<index>	Response OK +QLWRDRSP: <err> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	-

Parameter

<messageID>	Integer type. Message ID.		
<result>	Integer type. Result code indicating result of read operation.		
	Result code	CoAP Response	Code Description
	1	2.05	Content (indicates the correct result)
	11	4.00	Bad Request
	12	4.01	Unauthorized
	13	4.04	Not Found
	14	4.05	Method Not Allowed
	15	4.06	Not Acceptable
<objectID>	Integer type. Object ID.		
<instanceID>	Integer type. Instance ID.		
<resourceID>	Integer type. Resource ID.		
<value_type>	Integer type. The value types.		
	1	String	
	2	Opaque	
	3	Integer	
	4	Float	
	5	Boolean	
<len>	Integer type. The length of <value> .		
	When <value_type> =1, the maximum length of <len> is 1024 bytes.		
	When <value_type> =2, the maximum length of <len> is 512 bytes.		
	When <value_type> =3, <len> indicates the number of characters.		
	When <value_type> =4, <len> indicates the number of characters.		
	When <value_type> =5, <len> should be 1.		
<value>	String type. When <value_type> =2, it is in hexadecimal string format.		

<index>	When <value_type> =5, it is only 0 or 1. Integer type. The index number of the data. If the data is a combination of several messages, it should be split into several parts. If it is split into N parts, the order number of <index> is from 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 that this is the last message of the data.
<err>	Error code. For details of error codes, see Chapter 6 .

Example

```
+QLWURC: "read",62953,9,0,0 //Receive the read request from server.
AT+QLWRDRSP=62953,1,9,0,0,1,5,"abcde",0
OK
+QLWRDRSP: 0
```

4.3.8. AT+QLWWRRSP Respond to Write Request

This command allows the TE to respond to a write request from server.

AT+QLWWRRSP Respond to the Write Request	
Write Command AT+QLWWRRSP=<messageID>,<result>	Response OK +QLWWRRSP: <err> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	-

Parameter

<messageID>	Integer. Message ID.		
<result>	Integer. Result code indicating result of write operation.		
	Result code	CoAP Response	Code Description
	2	2.04	Changed, indicates the correct result
	11	4.00	Bad Request
	12	4.01	Unauthorized
	13	4.04	Not Found
	14	4.05	Method Not Allowed

<err> Error code. For details of error codes, see **Chapter 6**.

Example

```
+QLWURC: "write",36560,9,0,0,2,7,"5155454354454C",0 //Receive a write request from server.
AT+QLWWRRSP=36560,2
OK
+QLWWRRSP: 0
```

4.3.9. AT+QLWEXERSP Respond to Execute Request

This command allows the TE to respond to the execute request from server.

AT+QLWEXERSP Respond to Execute Request

Write Command AT+QLWEXERSP=<messageID>,<result>	Response OK +QLWEXERSP: <err> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	-

Parameter

<messageID>	Integer type. Message ID.		
<result>	Integer type. Result code indicating result of execute operation.		
	Result code	CoAP Response	Code Description
	2	2.04	Changed, indicates the correct result
	11	4.00	Bad Request
	12	4.01	Unauthorized
	13	4.04	Not Found
	14	4.05	Method Not Allowed
<err>	Error code. For details of error codes, see Chapter 6 .		

Example

```
+QLWURC: "execute",39040,15,0,5 //Receive an execute request from server.
AT+QLWEXERSP=39040,2
OK
```

+QLWEXERSP: 0

4.3.10. AT+QLWOBSRSP Respond to Observe Request

This command allows the TE to respond to an observe request from server.

AT+QLWOBSRSP Respond to Observe Request	
Write Command AT+QLWOBSRSP=<messageID>,<result>,<objectID>,<instanceID>,<resourceID>,<value_type>,<len>,<value>,<index>	Response OK +QLWOBSRSP: <err> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	-

Parameter

<messageID>	Integer type. Message ID.		
<result>	Integer type. Result code indicating result of observe operation.		
	Result code	CoAP Response	Code Description
	1	2.05	Content, indicates the correct result
	11	4.00	Bad Request
	12	4.01	Unauthorized
	13	4.04	Not Found
	14	4.05	Method Not Allowed
	15	4.06	Not Acceptable
<objectID>	Integer type. Object ID.		
<instanceID>	Integer type. Instance ID.		
<resourceID>	Integer type. Resource ID.		
<value_type>	Integer type. The value types.		
	1	String	
	2	Opaque	
	3	Integer	
	4	Float	
	5	Boolean	
<len>	Integer type. The value length.		
	When <value_type> =1, the maximum length of <len> is 1000 bytes.		
	When <value_type> =2, the maximum length of <len> is 500 bytes.		
	When <value_type> =3, <len> indicates the number of characters.		

	When <value_type> =4, <len> indicates the number of characters.
	When <value_type> =5, <len> should be 1.
<value>	String type. When <value_type> =2, it is in hexadecimal string format.
	When <value_type> =5, it is only 0 or 1.
<index>	Integer type. The index number of the data. If the data is a combination of several messages, it 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 that this is the last message of the data.
<err>	Error code. For details of error codes, see Chapter 6 .

Example

```
+QLWURC: "observe",624,0,9,0,0           //Receive an observe request from server.
AT+QLWOBSRSP=624,1,9,0,0,1,5,"abcde",0   //Respond to the observe request.
OK
+QLWOBSRSP: 0
```

4.3.11. AT+QLWNOTIFY Report Data to Server

This command reports the data to server.

AT+QLWNOTIFY Report Data to Server	
Write Command AT+QLWNOTIFY=<objectID>,<instanceID>,<resourceID>,<value_type>,<len>,<value>,<index>[,<ack>[,<rai_flag>]]	Response OK +QLWNOTIFY: <err> [+QLWURC: "report",<messageID>] [+QLWURC: "report_ack",<status_code>,<messageID>] If there is any error: +CME ERROR: <err>
Maximum Response Time	300 ms
Characteristics	-

Parameter

<objectID>	Integer type. Object ID.
<instanceID>	Integer type. Instance ID.

<resourceID>	Integer type. Resource ID. -1 indicates all resources.
<value_type>	Integer type. The value types. 1 String 2 Opaque 3 Integer 4 Float 5 Boolean
<len>	Integer type. The length of data to be sent. When <value_type> =1, the maximum length of <len> is 1000 bytes. When <value_type> =2, the maximum length of <len> is 500 bytes. When <value_type> =3, <len> indicates the number of characters. When <value_type> =4, <len> indicates the number of characters. When <value_type> =5, <len> should be 1.
<value>	String type. When <value_type> =2, it is in hexadecimal string format. When <value_type> =5, it is only 0 or 1.
<index>	Integer type. The index number of the data. If the data is combined with several messages, it 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 that this is the last message of the data.
<ack>	Integer type. Response type marker. 0 NON type data 1 CON type data
<rai_flag>	Integer type. Release assistance information (currently not supported). 0 No information available 1 TE will send only 1 UL packet without expecting DL packets 2 TE will send only 1 UL packet and expect only 1 DL packet
<status_code>	Integer type. For details of status codes, see Chapter 6 .
<err>	Error code. For details of error codes, see Chapter 6 .

4.3.12. AT+QLWSTATUS Query Current LwM2M Status

This command queries the current LwM2M status.

AT+QLWSTATUS Query Current LwM2M Status	
Read Command AT+QLWSTATUS?	Response +QLWSTATUS: <status> OK If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms

Characteristics

-

Parameter

<status>	Integer type. Status of LwM2M.
0	Not Registered
1	Registering
2	Registered
3	Deregistering
4	Deregistered
5	Invalid
6	Recovering
7	Need Recovery
<err>	Error code. For details of error codes, see Chapter 6 .

4.3.13. AT+QLWRECOVER Manually Trigger LwM2M Context Recovery Process

This command manually triggers the LwM2M context recovery process. The command is required to be executed only when the module wakes up from deep sleep mode.

AT+QLWRECOVER Manually Trigger LwM2M Context Recovery Process

Execution Command AT+QLWRECOVER	Response OK +QLWURC: "recovered",<status_code>,<serverID>,<server_type> If there is any error: +CMEE ERROR: <err>
Maximum Response Time	300 ms
Characteristics	-

Parameter

<status_code>	Integer type. For details of status codes, see Chapter 6 .
<serverID>	Integer type. Server ID.
100	Bootstrap Server
101	Diagnostics Server
102	DM Server
1000	Repository Server

<server_type>	Integer type. Server type. 0 DM Server 1 Bootstrap Server 2 Repository Server 3 Diagnostics Server
<err>	Error code. For details of error codes, see Chapter 6 .

Example

//The module has already connected to the server, the lifetime has not expired, and the module is woken up from deep sleep mode.

AT+QLWSTATUS?

+QLWSTATUS: 7 //The LwM2M context needs to be recovered.

OK

AT+QLWRECOVER

OK

+QLWURC: "recovered",0,102,0 //Recovered successfully.

4.4. LwM2M Related URCs

This chapter introduces LwM2M related URCs.

Table 2: LwM2M Related URCs

Index	Notification Display	Description
[1]	+QLWURC: "ping",<status_code>,<serverID>,<server_type>	Notify TE of the result of periodic update operations.
[2]	+QLWURC: "write",<messageID>,<objectID>,<instanceID>,<resourceID>,<value_type>,<len>,<value>,<index>	Notify TE to respond to the write request from server.
[3]	+QLWURC: "read",<messageID>,<objectID>,<instanceID>,<resourceID>	Notify TE to respond to the read request from server.
[4]	+QLWURC: "execute",<messageID>,<objectID>,<instanceID>,<resourceID>	Notify TE to respond to the execute request from server.
[5]	+QLWURC: "observe",<messageID>,<flag>,<objectID>,<instanceID>,<resourceID>	Notify TE that there is an observe request from server.
[6]	+QLWURC: "bootstrapping"	Notify TE that the bootstrap phase starts.

[7]	+QLWURC: "bs_finished"	Notify TE that the bootstrap phase has completed.
[8]	+QLWURC: "registering",<serverID>,<server_type>	Notify TE that the register phase starts.
[9]	+QLWURC: "report",<messageID>	Notify TE that CON data has been sent.
[10]	+QLWURC: "report_ack",<status_code>,<messageID>	Notify TE that CON data has been acknowledged.
[11]	+QLWURC: "lifetime_changed",<lifetime>,<serverID>,<server_type>	Notify TE that the registration lifetime has been changed.
[12]	+QLWURC: "recovered",<status_code>,<ServerID>,<Server_Type>	Notify TE of the result of recovering LwM2M context from deep sleep mode.

4.4.1. +QLWURC: "ping" Notify TE of Update Operation Result

This URC notifies TE of a result of an update operation.

+QLWURC: "ping" Notify TE of Update Operation Result

+QLWURC: "ping",<status_code>,<serverID>,<server_type>	Notify TE of the result of an update operation.
---	---

Parameter

<status_code>	Integer type. For details of status codes, see Chapter 6 .
<serverID>	Integer type. Server ID.
	100 Bootstrap Server
	101 Diagnostics Server
	102 DM Server
	1000 Repository Server
<server_type>	Integer type. Server type.
	0 DM Server
	1 Bootstrap Server
	2 Repository Server
	3 Diagnostics Server

4.4.2. +QLWURC: "write" Notify TE to Respond to Write Request

This URC notifies the TE to respond to a write request from server. The TE should respond to the request with **AT+QLWRRSP**.

+QLWURC: "write" Notify TE to Respond to Write Request

+QLWURC: "write",<messageID>,<objectID>,<instanceID>,<resourceID>,<value_type>,<len>,<value>,<index> Notify TE to respond to the write request from server.

Parameter

<messageID>	Integer type. Message ID.
<objectID>	Integer type. Object ID.
<instanceID>	Integer type. Instance ID.
<resourceID>	Integer type. Resource ID.
<value_type>	Integer type. The value types. 1 String 2 Opaque 3 Integer 4 Float 5 Boolean
<len>	Integer type. The value length. When <value_type> =1, the maximum length of <len> is 1024 bytes. When <value_type> =2, the maximum length of <len> is 512 bytes. When <value_type> =3, <len> indicates the number of characters. When <value_type> =4, <len> indicates the number of characters. When <value_type> =5, <len> should be 1.
<value>	String type. The value in hexadecimal string format received from the server.
<index>	Integer type. The index number of the write request. If the write request is a combination of several messages, it 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 URC is sorted from the largest one to the smallest. If <index> is 0, it means that this is the last message of the write request. Currently, only <index> =0 is supported.

4.4.3. +QLWURC: "read" Notify TE to Respond to Read Request

This URC notifies the TE to respond to a read request from server. The TE should respond to the request with **AT+QLWRDRSP**.

+QLWURC: "read" Notify TE to Respond to Read Request

+QLWURC: "read",<messageID>,<objectID>,<instanceID>,<resourceID>,<value_type>,<len>,<value>,<index> Notify TE to respond to a read request from server.

```
jectID>,<instanceID>,<resourceID>
```

Parameter

<messageID>	Integer type. Message ID.
<objectID>	Integer type. Object ID.
<instanceID>	Integer type. Instance ID. -1 indicates all resources under a specific object.
<resourceID>	Integer type. Resource ID. -1 indicates all resources under a specific instance.

4.4.4. +QLWURC: "execute" Notify TE to Respond to Execute Request

This URC notifies the TE to respond to the execute request from server. The TE should respond to the request with **AT+QLWEXERSP**.

+QLWURC: "execute" Notify TE to Respond to Execute Request

```
+QLWURC: "execute",<messageID>,<objectID>,<instanceID>,<resourceID>
```

Notify TE to respond to the execute request from server.

Parameter

<messageID>	Integer type. Message ID.
<objectID>	Integer type. Object ID.
<instanceID>	Integer type. Instance ID.
<resourceID>	Integer type. Resource ID.

4.4.5. +QLWURC: "observe" Notify TE to Respond to Observe Request

This URC notifies the TE to respond to an observe request from server. The TE should respond to the request with **AT+QLWOBSRSP**.

+QLWURC: "observe" Notify TE to Respond to Observe Request

```
+QLWURC: "observe",<messageID>,<flag>,<objectID>,<instanceID>,<resourceID>
```

Notify TE to respond to an observe request from server.

Parameter

<messageID>	Integer type. Message ID.
<flag>	Indicates whether or not to observe.

	0	Observe
	1	Cancel observe
<objectID>	Integer type. Object ID.	
<instanceID>	Integer type. Instance ID. -1 indicates all resources under all instances.	
<resourceID>	Integer type. Resource ID. -1 indicates all resources under a specific instance.	

4.4.6. +QLWURC: "bootstrapping" Notify TE that Bootstrap Phase Starts

This URC notifies the TE that the bootstrap phase starts.

+QLWURC: "bootstrapping" Notify TE that the Bootstrap Phase starts

+QLWURC: "bootstrapping"	Notify TE that the bootstrap phase starts.
---------------------------------	--

4.4.7. +QLWURC: "bs_finished" Notify TE that Bootstrap Phase Has Completed

This URC notifies the TE that the bootstrap phase has completed.

+QLWURC: "bs_finished" Notify TE that Bootstrap Phase Has Completed

+QLWURC: "bs_finished"	Notify TE that the bootstrap phase has completed.
-------------------------------	---

4.4.8. +QLWURC: "registering" Notify TE that Register Phase Starts

This URC notifies the TE that the register phase starts.

+QLWURC: "registering" Notify TE that Register Phase Starts

+QLWURC: "registering",<serverID>,<server_type>	Notify TE that the register phase starts.
--	---

Parameter

<serverID>	Integer type. Server ID.
100	Bootstrap Server
101	Diagnostics Server
102	DM Server
1000	Repository Server
<server_type>	Integer type. Server type.
0	DM Server
1	Bootstrap Server
2	Repository Server
3	Diagnostics Server

4.4.9. +QLWURC: "report" Notify TE that CON Data Has Been Sent

This URC notifies the TE that CON data has been sent.

+QLWURC: "report" Notify TE that CON Data Has Been Sent

+QLWURC: "report",<messageID> Notify TE that CON data has been sent.

Parameter

<messageID> Integer type. Message ID.

4.4.10. +QLWURC: "report_ack" Notify TE that CON Data Has Been Acked

This URC notifies the TE that CON data has been acknowledged.

+QLWURC: "report_ack" Notify TE that CON Data Has Been Acked

+QLWURC: "report_ack",<status_code>,<messageID> Notify TE that CON data has been acknowledged.

Parameter

<status_code> Integer type. For details of status codes, see **Chapter 6**.

<messageID> Integer type. Message ID.

4.4.11. +QLWURC: "lifetime_changed" Notify TE of Change in Registration Lifetime

This URC notifies the TE of a change in registration lifetime.

+QLWURC: "lifetime_changed" Notify TE of Change in Registration Lifetime

+QLWURC: "lifetime_changed",<lifetime>,<serverID>,<server_type> Notify TE of a change in registration lifetime.

Parameter

<lifetime> Integer type. Lifetime of registration. Range: 50–31536000 (365 days). Unit: second.
 If lifetime ≤ 100 s, then the Actual Update Interval = 30 + (lifetime - 50) × 4/5
 If lifetime ≤ 300 s, then the Actual Update Interval = 70 + (lifetime - 100) × 9/10
 If lifetime > 300 s, then the Actual Update Interval = 250 + (lifetime - 300) × 19/20
 When the Actual Update Interval timer expires, the module automatically reports

	data, updates and resets registration lifetime.
<serverID>	Integer type. Server ID.
100	Bootstrap Server
101	Diagnostics Server
102	DM Server
1000	Repository Server
<server_type>	Integer type. Server type.
0	DM Server
1	Bootstrap Server
2	Repository Server
3	Diagnostics Server

4.4.12. +QLWURC: "binding_changed" Notify TE of Change in Binding Node

This URC notifies the TE of a change in the binding mode.

+QLWURC: "binding_changed" Notify TE of Binding Mode Changed

+QLWURC: "binding_changed",<binding_mode>,<serverID>,<server_type> Notify TE of a change in the binding mode.

Parameter

<binding_mode>	String type. Only support "U" or "UQ".
<serverID>	Integer type. Server ID.
100	Bootstrap Server
101	Diagnostics Server
102	DM Server
1000	Repository Server
<server_type>	Integer type. Server type.
0	DM Server
1	Bootstrap Server
2	Repository Server
3	Diagnostics Server

4.4.13. +QLWURC: "recovered" Notify TE of LwM2M Context Recovery Result

This URC notifies the TE of the result of recovering LwM2M context from deep sleep mode.

+QLWURC: "recovered" Notify TE of LwM2M Context Recovery Result

+QLWURC: "recovered",<status_code>,<ServerID>,<Server_Type> Notify TE of the result of recovering LwM2M context from deep sleep mode.

Parameter

<status_code>	Integer type. For details of status codes, see Chapter 6 .
<serverID>	Integer type. Server ID.
100	Bootstrap Server
101	Diagnostics Server
102	DM Server
1000	Repository Server
<server_type>	Integer type. Server type.
0	DM Server
1	Bootstrap Server
2	Repository Server
3	Diagnostics Server

5 Examples

5.1. Manual Registration to LwM2M Server

```

AT+CREG?
+CREG: 0,1           //EPS network registered. (+CREG: 0,5 indicating registered roaming
                      service is also possible.)

OK
AT+CGPADDR=0
+CGPADDR: 0, "100.88.40.249"

OK
AT+QSCLK=0           // Disable sleep mode.
OK
//Before configuring LwM2M information, you should check LwM2M status with AT+QLWSTATUS?,
LwM2M information can be configured only when LwM2M status is "Not Registered".
AT+QLWCONFIG=1,"bootp.iot.t-mobile.com",5584,"urn:imei:867997030052996",900,0,"urn:imei:86
7997030052996","3083693CF1E6766A2408BFD88278249A2DB498B40A6BF6049300FD47DE325CF
C"
OK
AT+QLWREG           //Send a registration request.
OK

+QLWURC:"bootstrapping"

+QLWURC: "lifetime_changed",300,100,1

+QLWURC: "binding_changed","U",100,1

+QLWURC: "bs_finished"

+QLWURC: "registering",102,0

+QLWREG: 0           //Registered successfully.
AT+QLWUPDATE=0,1000
+QLWUPDATE: 59797

```

```

OK

+QLWUPDATE: 0,59797      //Updated successfully.
AT+QLWDEREG
OK

+QLWDEREG: 0             //De-registered successfully.

```

5.2. Custom Object Related Operations

```

AT+CEREG?
+CEREG: 0,1              //EPS network registered. (+CEREG: 0,5 indicating registered roaming
                          //service is also possible.)

OK
AT+CGPADDR=0
+CGPADDR: 0, "100.88.40.249"

OK
AT+QSCLK=0               // Disable sleep mode.
OK

//Before configuring LwM2M information, you should check LwM2M status via AT+QLWSTATUS?, LwM2M
//information can be configured only when LwM2M status is "Not Registered".
AT+QLWCONFIG=0,192.0.2.2,8085,"Test_dev",30,3 //Configure server relevant parameters.
OK
AT+QLWREG                //Send a registration request.
OK

+QLWREG: 0               //Registered successfully.
AT+QLWADDOBJ=19,0,5,0,1,2,3,4 //Add a custom object.
OK

+QLWADDOBJ: 0

+QLWURC: "ping",0,102,0  //Updated the object to server successfully.
AT+QLWADDOBJ?            //Query the current custom object.

+QLWADDOBJ: 19,0

OK

```

//The application server has sent a read request to UE with the intention of reading the resource (9/0/0).

+QLWURC: "read",32191,9,0,0

AT+QLWRDRSP=32191,1,9,0,0,1,7,"quectel",0

OK

+QLWRDRSP: 0

//The server has sent a write request to UE with the intention of writing the resource (9/0/3).

+QLWURC: "write",32193,9,0,3,2,1,"31",0

AT+QLWWRRSP=32193,2

OK

+QLWWRRSP: 0

//The application server has sent an execute request to UE.

+QLWURC: "execute",32198,9,0,4

AT+QLWEXERSP=32198,2

OK

+QLWEXERSP: 0

//The application server has observed the resource (9/0/0).

+QLWURC: "observe",32201,0,19,0,0

AT+QLWOBSRSP=32201,1,19,0,0,1,7,"quectel",0

OK

+QLWOBSRSP: 0

AT+QLWNOTIFY=19,0,0,1,10,"0123456789",0,1

OK

+QLWNOTIFY: 0

+QLWURC: "report",34180

+QLWURC: "report_ack",0,34180

//Report the data to server successfully.

5.3. Registration to Leshan LwM2M Server

LwM2M Client features can be tested on Leshan server. Please follow the steps below to register to Leshan server:

- In non-security mode:

```
AT+QLWCONFIG=0,"leshan.eclipseprojects.io",5683,"urn:imei:866971030000717",900,3
OK
```

- In PSK security mode:

```
AT+QLWCONFIG=0,"leshan.eclipseprojects.io",5684,"urn:imei:867997030056338",900,0,"urn:imei:867997030056338","30313233343536373839"
OK
```

If PSK security mode is used, client security configuration should be added into the web interface of the server according to steps illustrated below.

The screenshot shows the Leshan web interface with the 'SECURITY' tab selected. The interface displays the Leshan public key and a button to 'Add new client security configuration'. Below this is a table listing existing client security configurations.

Client Endpoint	Security Mode	Security Information
866971030000710	Pre-Shared Key	Identity : 866971030000710 Key : 77c755bd8cbb400abf2f0c420eed2cb0
866971030000711	Pre-Shared Key	Identity : 866971030000711 Key : 77c755bd8cbb400abf2f0c420eed2cb1
866971030578730	Pre-Shared Key	Identity : bc26654321 Key : 1234567891

Figure 7: Click "Add new client security configuration" in SECURITY Tab

New security configuration

Client endpoint: urn:imei:867997030056338

Security mode: Pre-Shared Key

Identity: urn:imei:867997030056338

Key: 30313233343536373839

Hexadecimal format

Close Create

Figure 8: Create "New security configuration" on Leshan Server

The Leshan public key (SubjectPublicKeyInfo der encoded)

Elliptic Curve parameters: secp256r1 [NIST P-256, X9.62 prime256v1] (1.2.840.10045.3.1.7)

Public x coord: fcc28728c123b155be410fc1c0651da374fc6ebe7f96606e90d927d188894a73

Public y coord: d2ffaa73957d76984633fc1cc54d0b763ca0559a9dff9706e9f4557dacc3f52a

Hex: 3059301306072A8648CE3D020106082A8648CE3D03010703420004FCC28728C123B155BE410FC1C0651DA374FC6EBE7F96606E90D927D188894A73D2FFAA73957D76984633FC1CC54D0B763CA0559A9DFF9706E9F4557DACC3F52A

Base64: MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAE/MKHKMEjsVW+QQ/BwGUdo3T8br5/lmBukNkn0YiJ5SnPS/6pz1X12mEYz/BzFTQt2PKBVmp3/lwbp9FV9rMP1Kg==

Add new client security configuration

Client Endpoint	Security Mode	Security Information
866971030000710	Pre-Shared Key	Identity : 866971030000710 Key : 77c755bd8cbb400abf2f0c420eed2cb0
866971030000711	Pre-Shared Key	Identity : 866971030000711 Key : 77c755bd8cbb400abf2f0c420eed2cb1
866971030578730	Pre-Shared Key	Identity : bc26654321 Key : 1234567891
urn:imei:867997030056338	Pre-Shared Key	Identity : urn:imei:867997030056338 Key : 30313233343536373839

Figure 9: Check Created "New security configuration" on Leshan Server

```

AT+CEREG?
+CEREG: 0,1           //EPS network registered (+CEREG: 0,5 indicates that registered roaming
                        service is also possible).

OK
AT+CGPADDR=0
+CGPADDR: 0, "100.88.40.249"

OK
AT+QSCLK=0           // Disable sleep mode.
OK
AT+QLWCONFIG=0,"leshan.eclipseprojects.io",5684,"urn:imei:867997030056338",900,0,"urn:imei:
867997030056338","30313233343536373839"
OK
AT+QLWREG
OK

+QLWREG: 0           //Registered successfully.
AT+QLWSTATUS?        //Query the current LwM2M status.
+QLWSTATUS: 2        //LwM2M status: registered.

OK
    
```

After successful connection to Leshan server, the server will present various operation permissions for each resource on the webpage. For instance, the "Lifetime" value can be read through clicking **"Read"** button on the webpage, and any resource value can be updated through clicking **"Write"** button and then inputting a new value on the webpage.

 LESHAN CLIENTS SECURITY				
Connected clients: 1				
Client Endpoint	Registration ID	Registration Date	Last Update	
urn:imei:867997030056338	oyi5lowe7Z	Jun 17, 2019 8:11:38 PM	Jun 17, 2019 8:11:38 PM	 

Figure 10: Online Client on Leshan Server

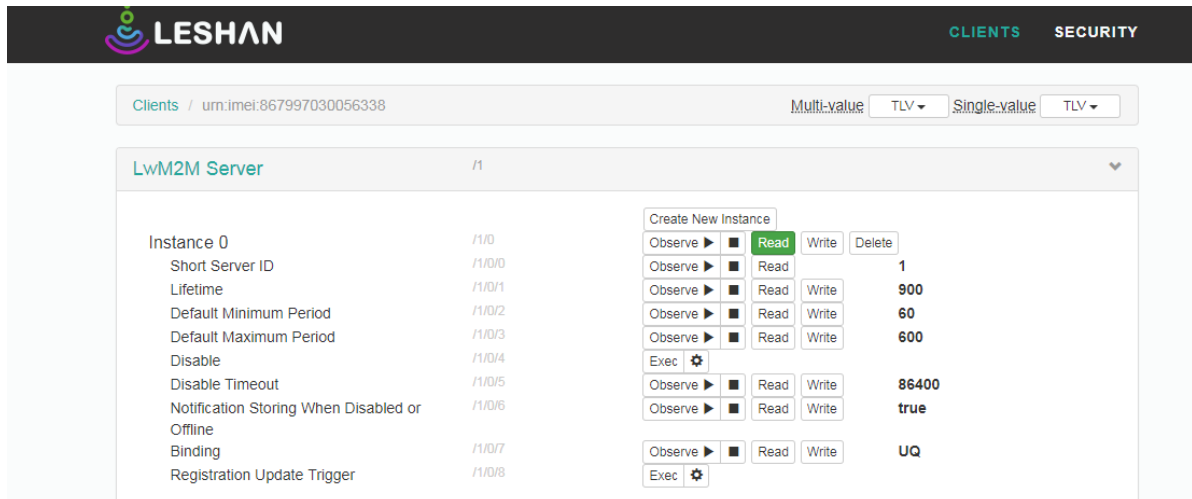


Figure 11: "Read" Operation on Leshan Server

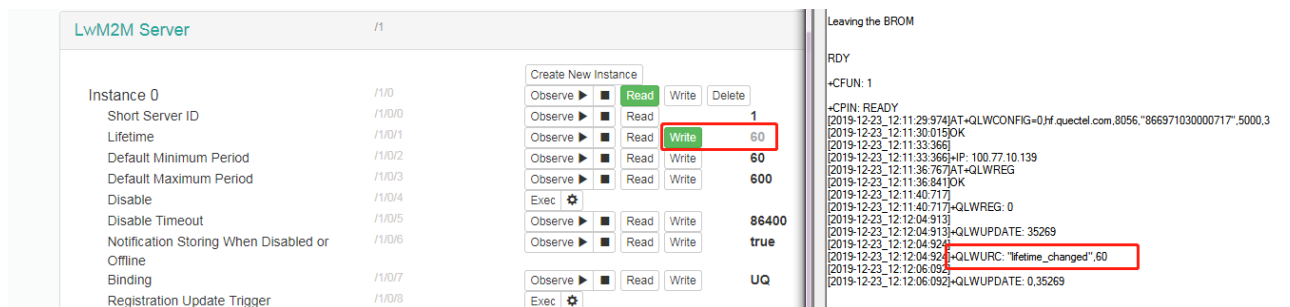


Figure 12: "Write" Operation on Leshan Server

5.4. LwM2M Updates (Leshan Server)

Periodically or based on certain events of LwM2M Client or initiated by the LwM2M Server, the LwM2M Client updates its registration information with an LwM2M Server by sending an "Update" request to the LwM2M Server.

5.4.1. LwM2M Server Update Operation

The "Update" operation can be initiated by the LwM2M Server via an "Execute" operation on the "Registration Update Trigger" Resource of the LwM2M Server Object. The LwM2M Client can perform an "Update" operation to refresh the lifetime of its registration to the LwM2M Server.

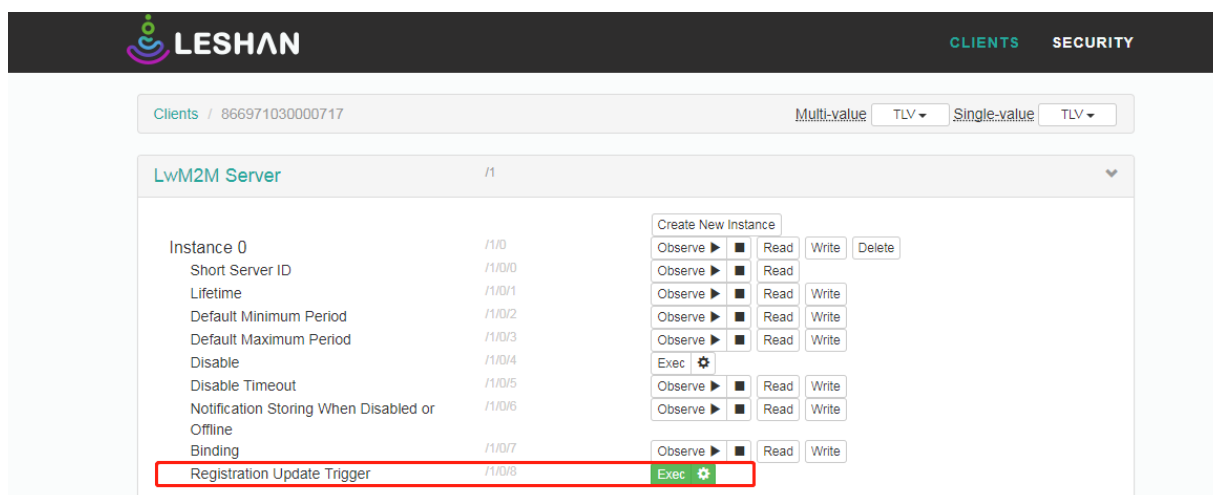


Figure 13: "Execute" Operation on Leshan Server

+QLWURC: "ping",0 // "Update" operation initiated by the LwM2M Server successfully.

5.4.2. LwM2M Client Update Operation

When any of the parameters listed in the table below changes, the LwM2M Client must send an "Update" operation to the LwM2M Server. The "Update" operation must only contain parameters which have changed values compared to the last registration to the LwM2M Server.

Table 3: Updateable Parameters

Parameter	Required
Lifetime	No
Binding Mode	No
SMS Number	No
Objects and Object Instances	No

Three common LwM2M Client update operations are provided in the subsequent chapters.

NOTE

SMS number updating is not supported currently.

5.4.2.1. Extend Lifetime of Registration

In this case the LwM2M Client sends an "Update" operation with no parameters or with lifetime.

The figure below shows an update flow example where the LwM2M Client sends an "Update" operation that only refreshes the registration, i.e. the message does not contain any parameter. With the second "Update" the Client changes the lifetime field to 6000 (unit: second) and hence the parameter **<lifetime>** is included in the message.

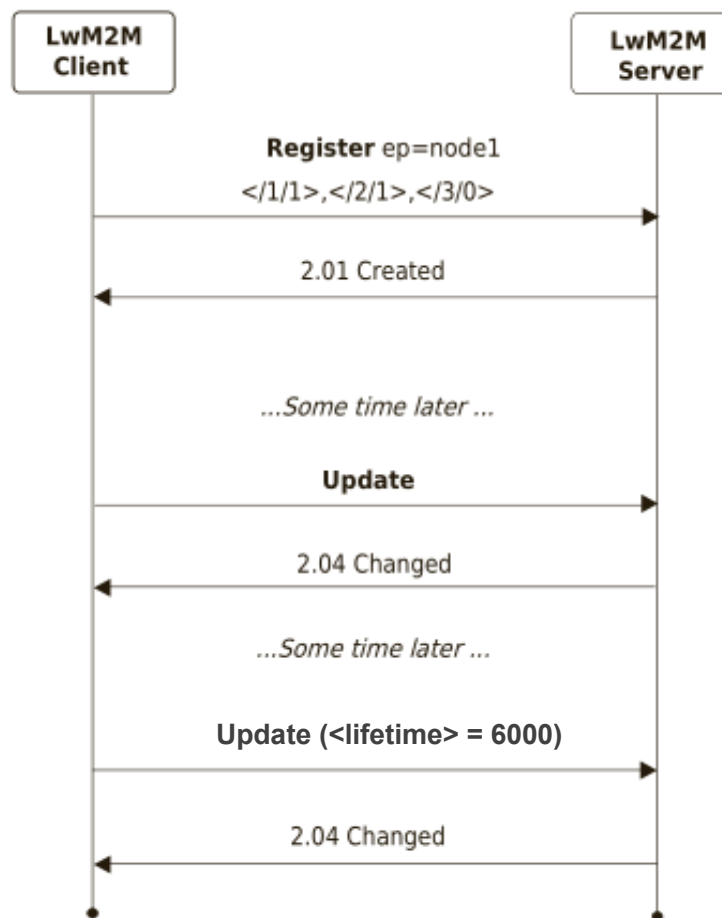


Figure 14: Update Flow Example

//The LwM2M Client has registered to Leshan Server successfully, and the default update period is 30 seconds. The module will update periodically (every 30 seconds), and can change the lifetime by **AT+QLWUPDATE**.

AT+CEREG?

+CEREG: 0,1

//EPS network registered (**+CEREG: 0,5** indicating registered roaming service is also possible).

OK

AT+CGPADDR?

+CGPADDR: 1,100.97.37.69

OK

AT+QLWCONFIG=0,"leshan.eclipse.org",5683,"urn:imei:866971030000717",30,3

OK

AT+QLWREG

OK

+QLWREG: 0 //Registered successfully.

AT+QLWUPDATE=0,200

+QLWUPDATE: 35270

OK

+QLWUPDATE: 0,35270 //Updated successfully.

No.	Time	Source	Destination	Protocol	Length	Info
454	2019/10/11 11:32:44	36.113.226.200	172.172.169.176	CoAP	385	CON, MID:35268, POST, TKN:c4 89 0a 00, /rd/lwm2m=1.0&reg-urn:imei:866971030000717&uqi=30
455	2019/10/11 11:32:44	172.172.169.176	36.113.226.200	CoAP	64	ACK, MID:35268, 2.01 Created, TKN:c4 89 0a 00, /rd
2540	2019/10/11 11:33:00	36.113.226.200	172.172.169.176	CoAP	64	CON, MID:35269, POST, TKN:c5 89 1b 00, /rd/ff7f105p53
2543	2019/10/11 11:33:00	172.172.169.176	36.113.226.200	CoAP	50	ACK, MID:35269, 2.04 Changed, TKN:c5 89 1b 00, /rd/ff7f105p53
2804	2019/10/11 11:33:02	36.113.226.200	172.172.169.176	CoAP	71	CON, MID:35270, POST, TKN:c6 89 1d 00, /rd/ff7f105p53?it=200
2805	2019/10/11 11:33:02	172.172.169.176	36.113.226.200	CoAP	50	ACK, MID:35270, 2.04 Changed, TKN:c6 89 1d 00, /rd/ff7f105p53

Update Registration Without Parameters

Update Registration With Lifetime

Frame 2540: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface 0 Ethernet II, Src: HuaweiE_45:32:150 (88:66:39:45:32:150), Dst: Dell_Ca:4b:2c (48:4d:7e:ca:4b:2c) Internet Protocol Version 4, Src: 36.113.226.200, Dst: 172.172.169.176 User Datagram Protocol, Src Port: 46656, Dst Port: 8856 Constrained Application Protocol, Confirmable, POST, MID:35269 01.. = Version: 1 ..00 = Type: Confirmable (0) 0100 = Token Length: 4 Code: POST (2) Message ID: 35269 Token: c5891040 Opt Name: #1: Uri-Path: rd Opt Name: #2: Uri-Path: ff7f105p53 [Response In: 2543] [Uri-Path: /rd/ff7f105p53]		0000 48 4d 7e ca 4b 2c 88 66 39 45 32 50 00 00 00 45 00 H=K, f 9E2P-E 0010 00 32 00 05 00 00 f3 11 6a 1f 24 71 e2 cb ac ac 2.....j5a 0020 a0 10 6c 40 1f 70 00 1e 5a 00 00 02 09 c5 c5 00 ..@...Z 0030 10 00 62 72 54 9a 06 60 37 50 63 36 53 70 53 33 rd ff 7f 10 5 p 5 3
---	--	--

Figure 15: Update Registration

5.4.2.2. Update Binding Mode of Registration

//The Lwm2m Client has registered to Leshan Server successfully.

AT+QLWUPDATE=1,0 //Update binding mode as UDP mode.

+QLWUPDATE: 38296

OK

+QLWUPDATE: 0,38296 //Updated successfully.

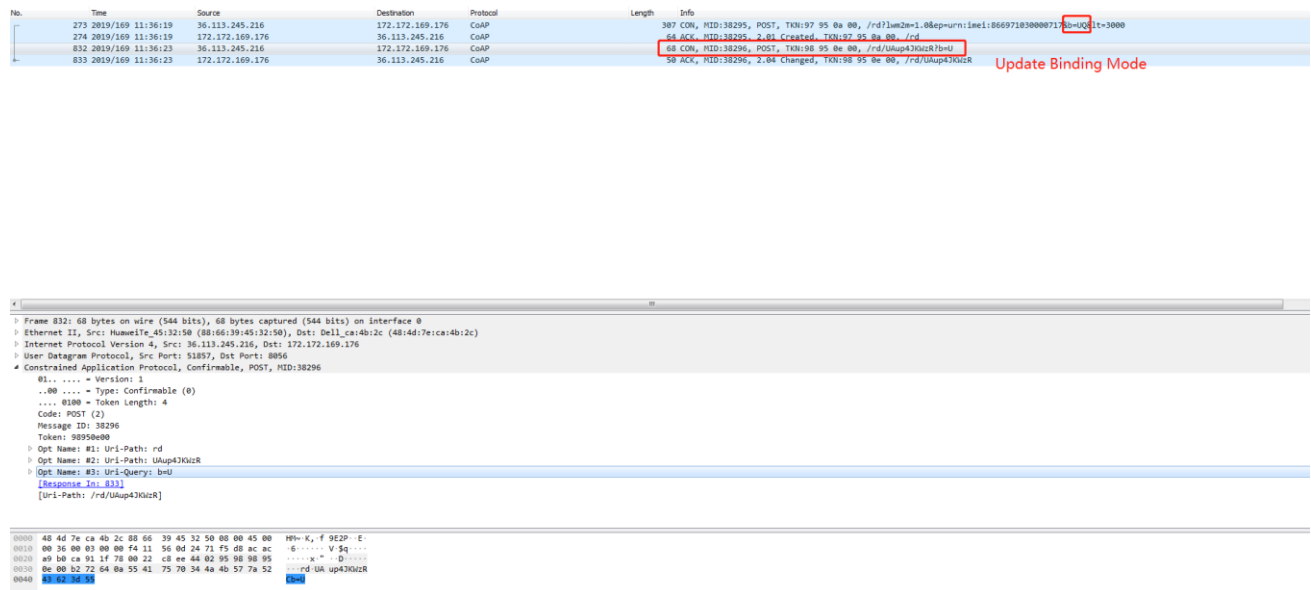


Figure 16: Update Binding Mode

5.4.2.3. Add/Remove Objects and Object Instances

In this case the LwM2M Client sends an "Update" operation with a body listing the complete list of objects and object instances.

As illustrated below, the LwM2M Client starts with an initial registration with lifetime of 3000 seconds. Later, **AT+QLWADDOBJ** is executed to add objects 3303/0 and 3303/1. With the "Update" operation, both of the LwM2M Client and the LwM2M Server include the new list of Objects and Object Instances.

```

AT+CEREG?
+CEREG: 0,1 //EPS network registered (+CEREG: 0,5 indicates that registered roaming
              service is also possible).

OK
AT+CGPADDR?
+CGPADDR: 1,100.97.37.69

OK
AT+QLWCONFIG=0,"leshan.eclipse.org",5683,"urn:imei:866971030000717",3000,3
OK
AT+QLWREG
OK
+QLWREG: 0 //Registered successfully.
AT+QLWADDOBJ=3303,0,3,5601,5602,5603
OK
  
```

+QLWADDOBJ: 0

+QLWURC: "ping",0 //Object list updated successfully.

No.	Time	Source	Destination	Protocol	Length	Info
1860	2019/169 14:18:54	36.113.238.130	172.172.169.176	CoAP	296	CON, MID:32242, POST, TKN:f2 7d 0b 00, /rd?lwm2m=1.0&ep=urn:imei:866971030000717&b=u&lt=3000
1861	2019/169 14:18:54	172.172.169.176	36.113.238.130	CoAP	64	ACK, MID:32242, 2.01 Created, TKN:f2 7d 0b 00, /rd
2539	2019/169 14:18:59	36.113.238.130	172.172.169.176	CoAP	264	CON, MID:32243, POST, TKN:f3 7d 10 00, /rd/ksyNlgCF4k
2540	2019/169 14:18:59	172.172.169.176	36.113.238.130	CoAP	50	ACK, MID:32243, 2.04 Changed, TKN:f3 7d 10 00, /rd/ksyNlgCF4k

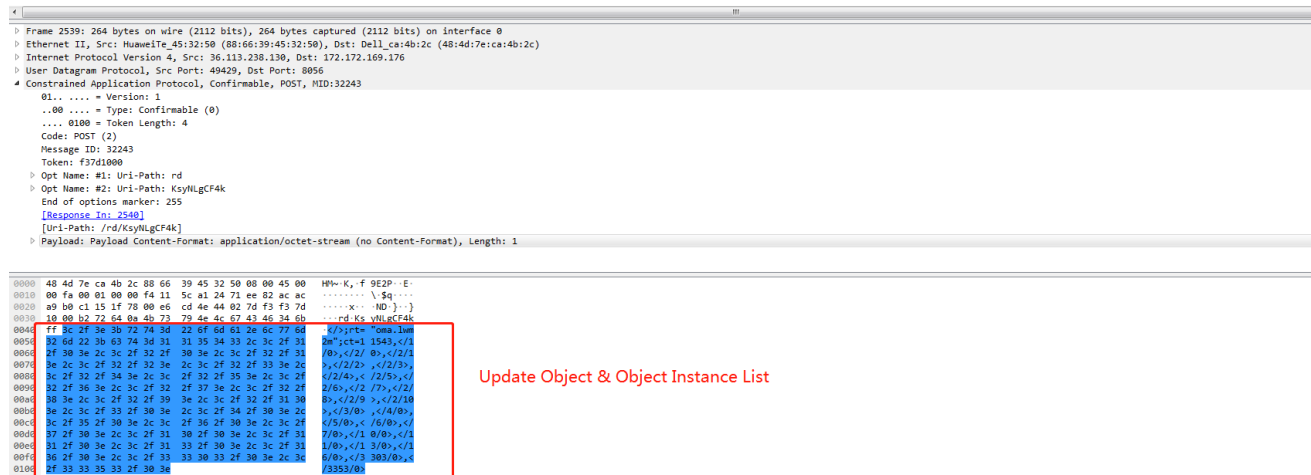


Figure 17: Update Object and Object Instance List

5.5. LwM2M Custom Object Related Operations (Leshan Server)

Currently, BC660K-GL and BC950K-GL support instance or resource level operations only for custom objects.

5.5.1. Observe Operation

//The LwM2M Client has registered to Leshan Server successfully, and custom object 3303 is observed by the server.

+QLWURC: "observe",16987,0,3303,0,5601 //The server initiates an observation request for resource level (3303/0/5601) change.

AT+QLWBSRSP=16987,1,3303,0,5601,4,4,88.88,0

OK

+QLWBSRSP: 0 //Observed successfully.

+QLWURC: "observe",16988,0,3303,0,-1 //The server initiates an observation request for instance

level (3303/0) change.

AT+QLWOBSRSP=16988,1,3303,0,5601,4,4,88.88,0

OK

+QLWOBSRSP: 0 //Observed successfully.

5.5.2. Notify Operation

//The LwM2M Client has registered to Leshan Server successfully and object 3303 has been observed successfully by the server.

AT+QLWNOTIFY=3303,0,5601,4,4,88.88,0,1 //Send CON data.

OK

+QLWNOTIFY: 0

+QLWURC: "report",17111

+QLWURC: "report_ack",0,17111 //Successfully responded to the notify request.

5.5.3. Read Operation

//The LwM2M Client has registered to Leshan Server successfully and a read request related to 3303/0/5601 has been initiated by the LwM2M Server.

+QLWURC: "read",17123,3303,0,5601

AT+QLWRDRSP=17123,1,3303,0,5601,4,4,66.66,0

OK

+QLWRDRSP: 0 //Successfully responded to the read request.

5.5.4. Execute Operation

//The LwM2M Client has registered to Leshan Server successfully and an execute request related to 3303/0/5605 has been initiated by the LwM2M Server.

+QLWURC: "execute",27452,3303,0,5605

AT+QLWEXERSP=27452,2

OK

+QLWEXERSP: 0 //Successfully responded to the execute request.

6 Error Codes

Table 4: Summary of LwM2M <err>

<err>	Description
0	Operation successful
1	Other error
2	Parameter number error
3	Parameter value error
4	Not register error
7	Disable error
13	Data length odd error
15	Not ready to receive error
32	Keep connecting error
33	Already registered error
34	Create LwM2M error

Table 5: Summary of <status_code>

<status_code>	Description
0	Success
1	Timeout
2	Packet not sent
3	Recovery failed
4	Update failed
9	Reset

10	Bad request
13	Forbidden
14	Not found
22	Precondition failed

7 Appendix References

Table 6: References

Reference
[1] Quectel_BC660K-GL&BC950K-GL_AT_Commands_Manual

Table 7:Terms and Abbreviations

Abbreviation	Description
3GPP	3 rd Generation Partnership Project
AS	Application Server
CIoT	Cellular Internet of Things
CoAP	Constrained Application Protocol
CON	Confirmable
DTLS	Datagram Transport Layer Security
IMEI	International Mobile Equipment Identity
IoT	Internet of Things
IP	Internet Protocol
IWMSC	Interworking Mobile Switching Center
LwM2M	Lightweight Machine to Machine
MME	Mobility Management Entity
NAS	Non-Access Stratum.
NB-IoT	Narrowband Internet of Things

NON	Non-Confirmable
OMA	Open Mobile Alliance
PSK	Pre-shared Key
REST	Representational State Transfer
SCEF	Service Creation Environment Function
SGN	Serving Gateway Node
SMS	Short Message Service
SMSC	Short Message Service Center
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UE	User Equipment
URC	Unsolicited Result Code
