ST87MXX LwM2M Application Note

Purpose and scope

This document provides details about the LwM2M application protocol embedded in the ST87MXX NB-IoT module.

Document status
Official

1 General information

1.1 Acronyms and terms

Table 1. Definitions of terms

CoAP	Constrained Application Protocol
DFOTA	Differential Fw update Over the Air
IP	Internet Protocol
LwM2M	Lightweight Machine to Machine
NVM	Non-Volatile Memory
OMA	Open Mobile Alliance
UDP	User Datagram Protocol
URI	Uniform Resource Identifier

1.2 Reference documents

The documents listed in Table 2 provide further information.

Table 2. Document references

Reference	Document
[1]	ST87MXX UM AT commands description
[2]	https://www.openmobilealliance.org/release/LightweightM2M/V1_2-20201110-A/OMA-TS-LightweightM2M_Core-V1_2-20201110-A.pdf
[3]	ST87MXX FOTA Application Note

1.3 Revision history

Table 3. Document revision history

Date	Version	Changes
	V1.0	First release

1.4 Table of contents

1.5 List of tables

2 Introduction

2.1 About LwM2M and ST87MXX

The Lightweight Machine to Machine (LwM2M) protocol, designed by the Open Mobile Alliance (OMA), supports various functionalities such as device configuration and management, data reporting, firmware updates, diagnostics, connection management and bootstrap.

The purpose of this application note is to provide information about the LwM2M functionalities embedded in the ST87MXX NB-IoT module.

The LwM2M application protocol can be used to start the FW update over the air (DFOTA) process as described in [3]. In addition to that, the current implementation of LwM2M client is accessible to Hosts through dedicated AT commands. This should be considered as preliminary feature not fully implemented or validated yet. For instance, the LwM2M bootstrap has not been implemented and the observation mechanism has been implemented only partially.

The ST87MXX, being a constrained device, implements only a restricted subset of the features defined in the LwM2M core specifications.

Figure 1 illustrates the integration of the LwM2M protocol stack into the ST87MXX.

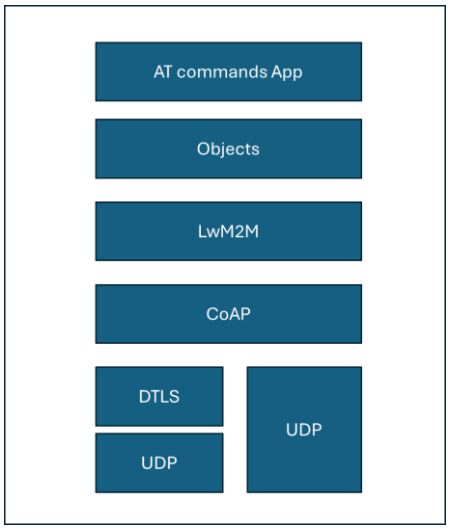

Figure 1. LwM2M Protocol stack

The ST87MXX supports the registration to a single LwM2M server at a time.

The supported transport layer is UDP, optionally secured with a DTLS layer.

As with most ST87MXX features, the LwM2M client must be controlled by a host application. The host application is responsible for configuring and starting the LwM2M stack and triggering registration to the server.

Once the device completes registration with the LwM2M server, the server can access the client's resources. Depending on the rights assigned to each resource, the server can read, write, or execute the resource. Examples of these operations are provided in the following chapters.

In addition to the ST87MXX built-in LwM2M objects, the host can define additional custom objects.

For a complete description of the AT commands and parameters, refer to the AT command user manual [1].

3 Built-in Object

Although many object IDs and resource IDs are defined in [2], the ST87MXX module implements only a subset of them. The table below lists the built-in LwM2M objects available in the ST87MXX module:

Object Name	Object ID
Server Object	1
Device Object	3
Firmware Update Object	5

Other built-in objects may be present, but they might not be fully implemented or could be reserved for other services. For example, the firmware update object is used for the DFOTA service embedded in the ST87MXX module.
A known limitation exists in the implementation of the observation mechanism on the client side: when the ST87MXX wakes after sleep, the list of resources on which the server had started the observation is not restored.
The following paragraphs describe the resource IDs available in the ST87MXX module.

3.1 LwM2M Device Object

Resource ID	Name	Initiated by	Server rights	Host rights	Instances	Type	Size	Description	Comment
0	Manufacturer	Host	R	RW	Single	String	char[20]	Human readable manufacturer name	HOST can change the value with AT#LWCHGRSCVAL command at any time
1	Model Number	Host	R	RW	Single	String	char[20]	A model identifier (manufacturer specified string)	HOST can change the value with AT#LWCHGRSCVAL command at any time
2	Serial Number	Host	R	RW	Single	String	char[20]	Serial Number	HOST can change the value with AT#LWCHGRSCVAL command at any time
3	Firmware Version	ST87MXX	R	R	Single	String	char[64]	Current firmware version of the ST87MXX module	It is filled automatically. Refer to ATI command for version string format.
4	Reboot	-	E	-	Single	None	-	Reboot the LwM2M Device to restore from unexpected firmware failure	It is equivalent to the AT#RESET command sent by HOST. Note: After Execution request, the Server may not receive the response due to the reset itself.
5	Factory Reset	-	E	-	Single	None	-	Perform factory reset of the LwM2M Device	It is equivalent to the AT#RFACTORY command sent by

									HOST. Note: After Execution request, the Server may not receive the response due to factory reset itself. After Factory reset, HOST should restore the module configuration.
9	Battery Level	Host	R	RW	Single	Integer	uint8_t	Current battery level as a percentage (0-100%)	HOST can change the value with AT#LWCHGRSCVAL command at any time
10	Memory Free	Host	R	RW	Single	Integer	int64_t	Estimated current available amount of storage space (in kilobytes)	HOST can change the value with AT#LWCHGRSCVAL command at any time
11	Error Code	Host	R	RW	Multiple	Integer	uint8_t[N]	0=No error, 1=Low battery power, 2=External power supply off, ...	HOST can change the value with AT#LWCHGRSCVAL command at any time
12	Reset Error Code	-	E	-	Single	None	-	Delete all error code Resource Instances and create only one zero-value error code	
13	Current Time	Host	RW	RW	Single	Time	int64_t	Current UNIX time of the LwM2M Client	HOST can change the value with AT#LWCHGRSCVAL command at any time
14	UTC Offset	Host	RW	RW	Single	String	char[7]	Indicates the UTC offset currently in effect for this LwM2M Device	HOST can change the value with AT#LWCHGRSCVAL command at any time
15	Timezone	Host	RW	RW	Single	String	char[30]	Indicates in which time zone the LwM2M Device is located (IANA Timezone format)	HOST can change the value with AT#LWCHGRSCVAL command at any time
16	Supported Binding and Modes	ST87MXX	R	R	Single	String	char[10]	Indicates which bindings and modes are supported in the LwM2M Client	It is filled automatically (set to UDP)
17	Device Type	Host	R	RW	Single	String	char[10]	Type of the device (manufacturer specified string)	HOST can change the value with AT#LWCHGRSCVAL command at any time
18	Hardware Version	Host	R	RW	Single	String	char[20]	Current hardware version of the device	HOST can change the value with AT#LWCHGRSCVAL command at any time
19	Software Version	Host	R	RW	Single	String	char[20]	Current software version of the device	HOST can change the value with AT#LWCHGRSCVAL command at any time
20	Battery Status	Host	R	RW	Single	Integer	uint8_t	0: Normal, 1: Charging	HOST can change the value with AT#LWCHGRSCVAL

								2: Charge Complete 3: Damaged 4: Low Battery 5: Not Installed 6: Unknown	command at any time

Table 4. Device Object table

3.2 ST87MXX LwM2M Server Object

The ID of the LwM2M Server Object is: 1

Resource ID	Name	Owner	Server rights	Host rights	Instances	Type	Size	Description	Comment
0	Short Server ID	Host	R	RW	Single	Integer	uint16_t	Used as link to associate server Object Instance.	User SHALL initialize it using AT#LWPARAMS command before LwM2MCFG (value different from 0)
1	Lifetime	Host	RW	RW	Single	Integer	uint32_t	Specify the lifetime of the registration in seconds (see Client Registration Interface).	Initial value SHALL be written by HOST with AT#LWPARAMS command before registration. Host can update it with AT#LWCHGRSCVAL command
6	Notification Storing When Disabled or Offline	ST87MXX	RW	-	Single	Boolean	bool_t	If true, stores “Notify” operations while server is disabled/offline ; if false, discards them.	Only Disable is supported
7	Binding	ST87MXX	RW	-	Single	String	char[10]	Defines the transport binding configured for the LwM2M Client.	Only UDP is currently supported
8	Registration Update Trigger	-	E	-	Single	None	-	If executed, triggers an “Update” operation with this LwM2M Server.	Execution cab be triggered by Server. Host can trigger registration update with AT#LWREG=2 command
13	Registration Priority Order	Host	R	RW	Single	Unsigned Integer	int32_t	Registration order for multiple servers.	To instantiate the resource, the HOST shall write the value using AT#LWCHGRSCVAL command before registration
14	Initial Registration Delay Timer	Host	RW	RW	Single	Unsigned Integer	int32_t	Delay before registration is attempted for this server.	To instantiate the resource, the HOST shall write the value using AT#LWCHGRSCVAL command before registration

15	Registration Failure Block	Host	R	RW	Single	Boolean	int32_t	If true, blocks registration to other servers on failure.	To instantiate the resource, the HOST shall write the value using AT#LWCHGRSCVAL command before registration
16	Bootstrap on Registration Failure	Host	R	RW	Single	Boolean	int32_t	If true, triggers re-bootstrap on registration failure.	To instantiate the resource, the HOST shall write the value using AT#LWCHGRSCVAL command before registration
17	Communication Retry Count	Host	RW	RW	Single	Unsigned Integer	int32_t	Number of communication attempts before failure.	To instantiate the resource, the HOST shall write the value using AT#LWCHGRSCVAL command before registration
18	Communication Retry Timer	Host	RW	RW	Single	Unsigned Integer	int32_t	Delay between communication attempts (exponential back-off).	To instantiate the resource, the HOST shall write the value using AT#LWCHGRSCVAL command before registration
19	Communication Sequence Delay Timer	Host	RW	RW	Single	Unsigned Integer	int32_t	Delay between communication sequences.	To instantiate the resource, the HOST shall write the value using AT#LWCHGRSCVAL command before registration
20	Communication Sequence Retry Count	Host	RW	RW	Single	Unsigned Integer	int32_t	Number of communication sequences before registration is considered failed.	To instantiate the resource, the HOST shall write the value using AT#LWCHGRSCVAL command before registration

Table 5. Server Object table

3.3 Firmware Update Object

The firmware update object is used for the DFOTA service embedded in the ST87MXX module.

ID	Name	Owner	Server rights	Host Rights	Instances	Type	Size	Description	Comment
0	Package	Device	W	-	Single	Opaque		Firmware package	Not supported
1	Package URI	Host	RW	R	Single	String	char[256]	URI from where the device can download the firmware package by an alternative mechanism.	
2	Update	Device	E	-	Single	None	uint8_t	Updates firmware by using the firmware package stored in Package, or, by using the firmware downloaded from the Package URI. This Resource is only executable when the value of the State	Note: Server may not receive the response due to FW installation process start.

								Resource is Downloaded.	
3	State	Device	R	R	Single	Integer	uint8_t	Indicates current state with respect to this firmware update. 0: Idle 1: Downloading 2: Downloaded 3: Updating	
5	Update Result	Device	R	R	Single	Integer		Contains the result of downloading or updating the firmware. 0: Initial value 1: Firmware updated successfull 2: Not enough flash memory 3: Out of RAM 4: Connection lost 5: Integrity check failure 6: Unsupported package type 7: Invalid URI 8: Firmware update failed 9: Unsupported protocol	
8	Firmware Update Protocol Support	Device	R	R	Multiple	Integer	uint8_t[6]	Indicates what protocols the LwM2M Client implements to retrieve firmware images.	constant value: CoAP
9	Firmware Update Delivery Method	Device	R	R	Single	Integer	uint8_t	Indicates support for transferring firmware images: 0: Pull only 1: Push only 2: Both.	Only Pull mode is supported.
10	Cancel	Device	E	-	Single				Equivalent to AT#FOTADL=0
11	Severity	Host	RW	RW	Single		uint8_t		Default value: 1
13	Maximum Defer Period	Host / Server	RW	RW	Single		uint32_t		

Table 6 Firmware Update Object table

4 Custom objects

4.1 About custom objects

In addition to the built-in LwM2M objects, the host can add custom objects to the LwM2M stack.

Unlike the built-in objects, the host must manage the resources of such objects, i.e. the resources belonging to custom objects are not stored in the ST87MXX module. The ST87MXX provides all the AT commands and URCs required to enable the host to handle requests from the LwM2M server.

A known limitation exists in the implementation of the observation mechanism on the client side: when the ST87MXX wakes after sleep, the list of resources on which the server had started the observation is not restored.

The maximum number of custom object instances is limited to 4. The maximum number of resources per object is limited to 32. Multiple instance resource is not supported.

Refer to the use case chapters below for details on AT command usage.

5 LwM2M usage tips

5.1 About Lifetime and Sleep periods

Lifetime is essential and must be set using AT#LWPARAMS command. It defines the timeout for registration update message. Refer to [2] for a full description. This parameter is also written into Resource ID 1 of the built-in Server object.

If the Server does not receive the Registration Update message within the time defined by the lifetime, it de-registers the client.

The Host can change the value of this resource using the AT#LWCHGRSCVAL command at any time.

The ST87MXX uses the Lifetime to define the maximum time it can sleep to ensure that the client can send the Registration Update message to the Server. When the ST87MXX is in sleep mode, based on network configuration, LwM2M server requests may be delayed or even not been delivered to the ST87MXX.

As a consequence, this parameter has a significant impact on power consumption. The Host must set the appropriate value according to the requirements of the device.

The Host can enable the LwM2M Queue mode through the binding_mode parameter. For example:

AT#LWPARAMS=ST87M01_CLIENT,1,1,COAP://23.97.187.154:5683,**UQ**,60,0,1,1

Refer to [2] for a full description of the Queue mode.

5.2 About registration request

The LwM2M client must keep sending the Registration Update to the server. In case of manual registration mode, the host must consider this process can fail (i.e. in case of bad radio conditions) and must re-send the AT#LWREG command to re-establish the registration to the server. Thus, it is recommended to enable the LwM2M URCs.

5.3 About Reboot and Factory Reset execution

Note that when the server sends an execution request for the Reboot or Factory Reset resources, the module immediately executes the request and the acknowledgment to the server may not being sent. For this reason, based on the server's specific implementation, the execution request may time out on the server side.

Users should be aware that executing the Factory Reset restores the default connection settings in the ST87MXX, which may not be suitable for re-establishing a connection to the network. The Host must handle this specific case by restoring all application-specific settings.

6 LwM2M Scenarios

6.1 Command descriptions

The LwM2M feature includes the following AT commands:

- Service management
  - AT#LWSTART
  - AT#LWSTOP
  - AT#LWCFG
  - AT#LWPARAMS
- Server management
  - AT#LWREG
  - AT#LWLISTSERV
- Built-in Object management
  - AT#LWCHGRSCVAL
  - AT#LWREADRSCVAL
- Custom Object Management
  - AT#LWADDOBJ
  - AT#LWRMOBJ
  - AT#LWNTYRDOBJ
  - AT#LWNTYWROBJ
  - AT#LWNTYEXEOBJ
  - AT#LWNTYTO
- Object Management
  - AT#LWDUMPOBJ
  - AT#LWLISTOBJ

See the AT command user manual [1] for more details on the parameters.

6.2 Manual registration to LwM2M Server

The example below shows the sequence of AT commands sent to the ST87Mxx module to register to a LwM2M server. In the following description, we consider that the ST87Mxx module is already connected to the NB-IoT network.

6.2.1 Parameters

Here is an example of AT commands sequence to manually register to LwM2M server.

The Host has to manage the creation of the UDP socket and start the LwM2M stack.
If not present already, the LwM2M parameters must be set into the device. Note that params set using AT#LWPARAMS command are saved into NVM.

- <client_name>: it is the LwM2M client name indicated to the Server
- <server_id>: it is the LwM2M server Id
- <server_type>: only Normal server type is supported.
- <LwM2M_Server_URL> should in the format <protocol>://<ip_address>:<port_number>
  - For example: coap://23.97.187.154:5683
- <binding_mode>: only UDP mode is supported
- <lifetime>: The lifetime parameter in a LwM2M connection indicates the expected lifetime of the registration for the LwM2M client. This value MUST be the same as the value held in the Resource named “Lifetime” of the corresponding instance of the Server Object (ID #1): /1/1/1. If the lifetime of a registration expires without receiving an

update from the LwM2M Client, the LwM2M Server will consider it a de-registration and remove the registration of that LwM2M Client and existing observations

- <storing>: only 0 (deactivated) is supported
- <auto_reg>: 0 for Manual registration
- <urc_enable>: 1 to enable LwM2M URCs

The host must configure the transport layer beforehand, the link could be secured or not. Then it can send the command to trigger the registration to server and wait for registration complete URC.

6.2.2 MSC

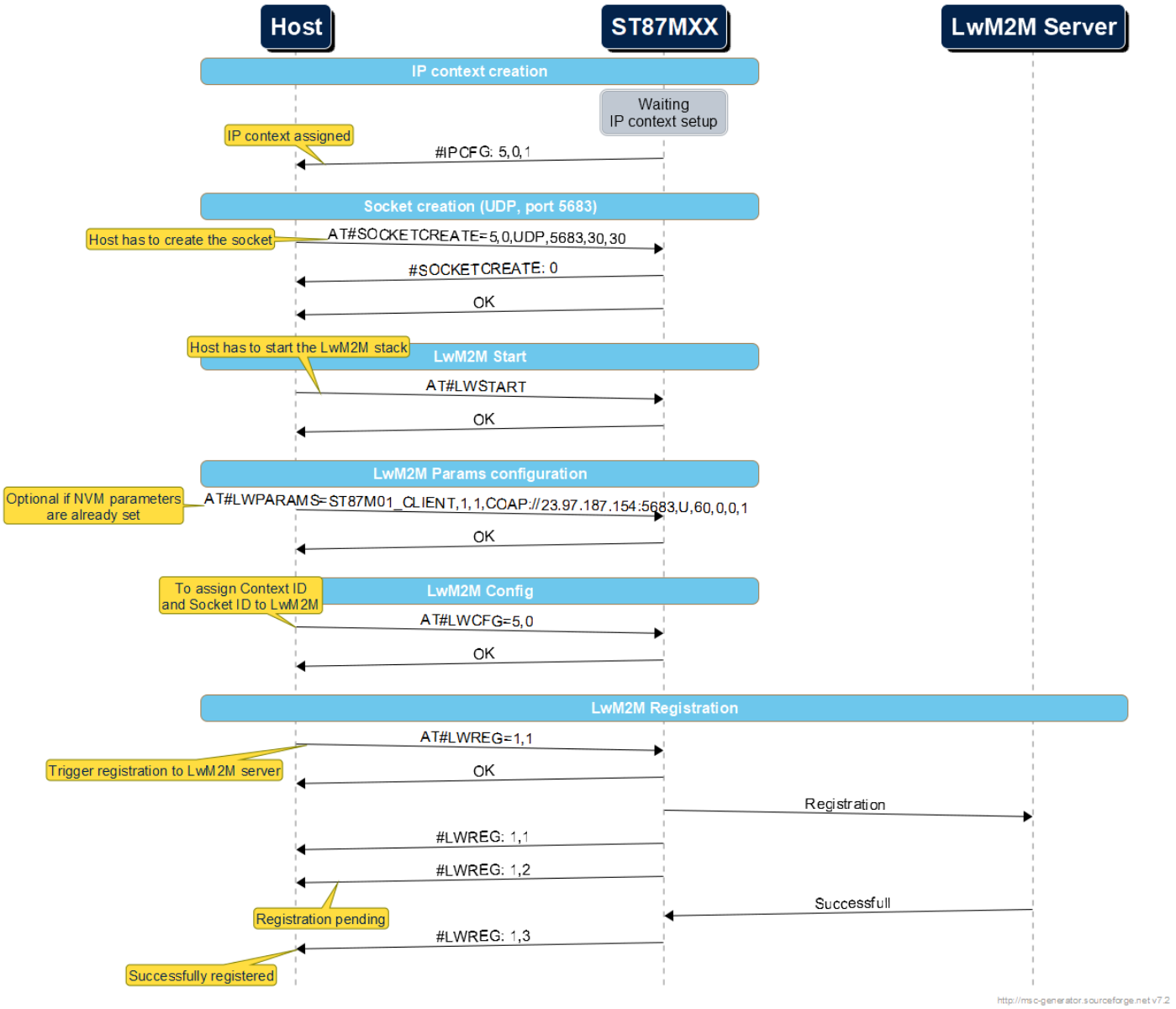

Figure 2. Manual registration to LwM2M Server

6.2.3 Terminal

```
AT#SOCKETCREATE=5,0,UDP,5683,30,30
#SOCKETCREATE: 0
OK

AT#LWSTART
```

```
OK

AT#LWPARAMS=ST87M01_CLIENT,1,1,COAP://23.97.187.154:5683,U,60,0,0,1
OK

AT#LWCFG=5,0
OK

AT#LWREG=1,1
OK

#LWREG: 1,1

#LWREG: 1,2

#LWREG: 1,3
```

6.3 Auto Registration to LwM2M Server

ST87Mxx is able to auto register the LWM2M server once the network attachment is complete. It will register to the server on its own without any actions from the host. The AT command AT#LWPARAMS can be used to enable the auto-registration mechanism.

The example below shows the sequence of AT commands to send to the ST87Mxx module to configure auto-registration to the LwM2M server. If auto-registration to the LwM2M server is enabled, the ST87Mxx also manages socket creation.

6.3.1 Parameters

Here is an example of AT commands sequence to enable auto registration to LwM2M server.

The Host has to enable the LwM2M auto registration using AT#LWPARAMS command

- <auto_reg>: 1 to enabled auto registration to LwM2M server

The Host has to save the LwM2M params into NVM.

6.3.2 MSC

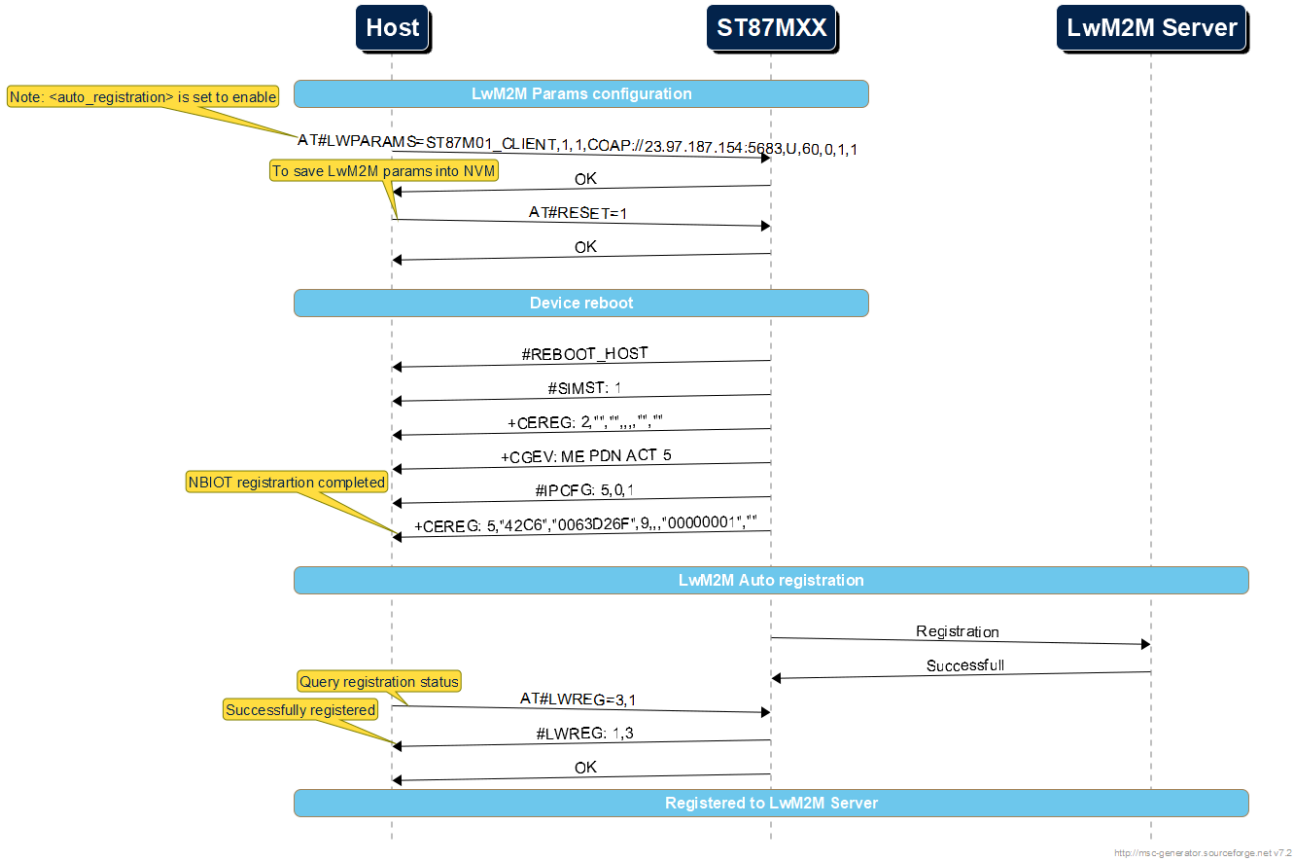

Figure 3. Auto registration to LwM2M server

6.3.3 Terminal

```
AT#LWPARAMS=ST87M01_CLIENT,1,1,COAP://23.97.187.154:5683,U,60,0,1,1
OK

AT#RESET=1
OK

#REBOOT_HOST
#SIMST: 1
+CEREG: 2,"","",,,,"",""
+CGEV: ME PDN ACT 5
#IPCFG: 5,0,1
+CEREG: 5,"42C6","0063D26F",9,,,"00000001",""

AT#LWREG=3,1
#LWREG: 1,3
OK
```

6.4 Server access Built-in Objects

Once the client is registered to the LwM2M server, the server can access client resources based on the specific rights defined for each resource.

No intervention is required by the host.
The host can change the value of the ST87MXX built-in resources at any time, based on resource rights.

In this use case, it is also possible to observe that the server can start an observation on client resources. Note that observation has not been fully implemented.

6.4.1 Parameters

In the example below, the host changes the value of an LwM2M built-in resource using the AT#LWCHGRSCVAL command.

6.4.2 MSC

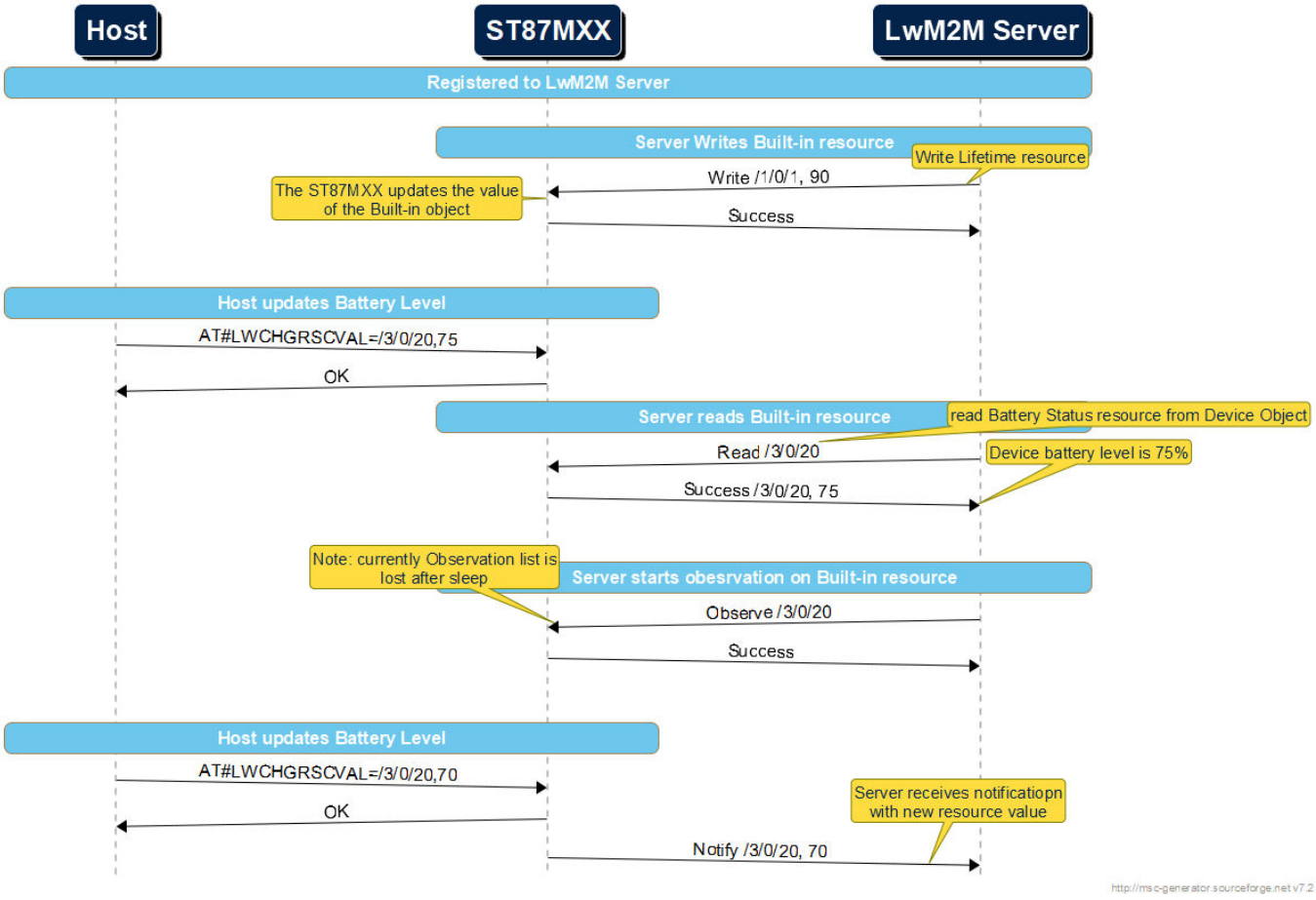

Figure 4. Server access built-in resources

6.4.3 Terminal

```
AT#LWCHGRSCVAL=/3/0/20,75
OK

AT#LWCHGRSCVAL=/3/0/20,70
OK
```

6.5 Server reads / write Custom objects

The host can define the LwM2M custom objects. Based on the resource rights, the server sends read, write, or execute requests.

6.5.1 Parameters

Custom objects must be created each time the ST87MXX is powered on by using the AT#LWADDOBJ command. The ST87MXX sends URCs to the host whenever it receives requests related to custom objects. The host must send the following command related to the server request within the predefined timeout.
See the AT command user manual **Error! Reference source not found.** for more details on the parameters.

6.5.2 MSC

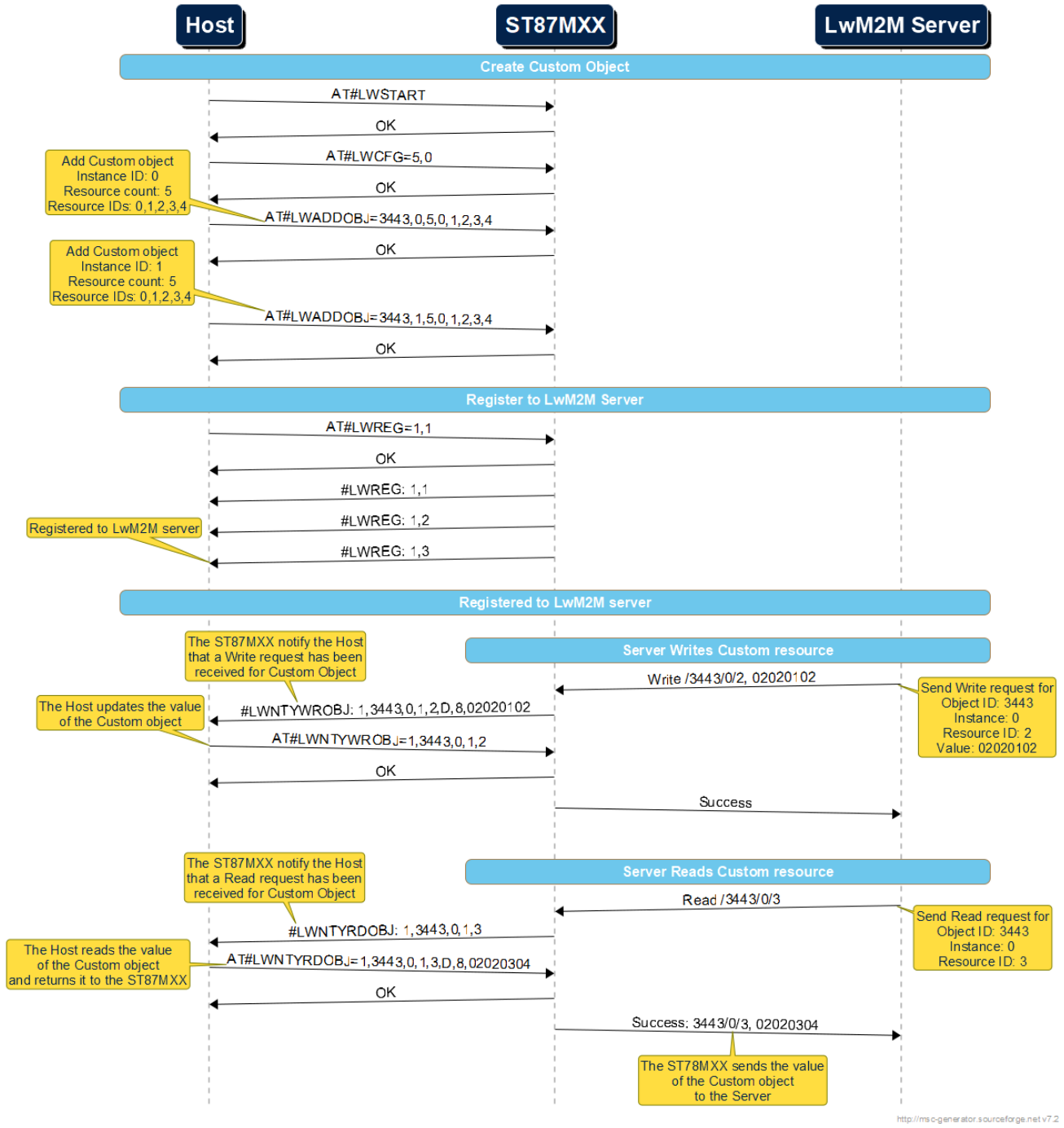

Figure 5. Server access custom resources

6.5.3 Terminal

```
AT#LWSTART
OK

AT#LWCFG=5,0
OK

AT#LWADDOBJ=3443,0,5,0,1,2,3,4
OK

AT#LWADDOBJ=3443,1,5,0,1,2,3,4
OK
```

```
AT#LWREG=1,1
OK
#LWREG: 1,1
#LWREG: 1,2
#LWREG: 1,3

#LWNTYWROBJ: 1,3443,0,1,2,D,8,02020102
AT#LWNTYWROBJ=1,3443,0,1,2
OK

#LWNTYRDOBJ: 1,3443,0,1,3
AT#LWNTYRDOBJ=1,3443,0,1,3,D,8,02020304
OK
```

6.6 Server triggers a Reset of the Client device

The LwM2M server can trigger a reboot of the device by sending an execution request to the Reboot resource of the Device object.

6.6.1 Parameters

No specific AT commands are required for this use case.

6.6.2 MSC

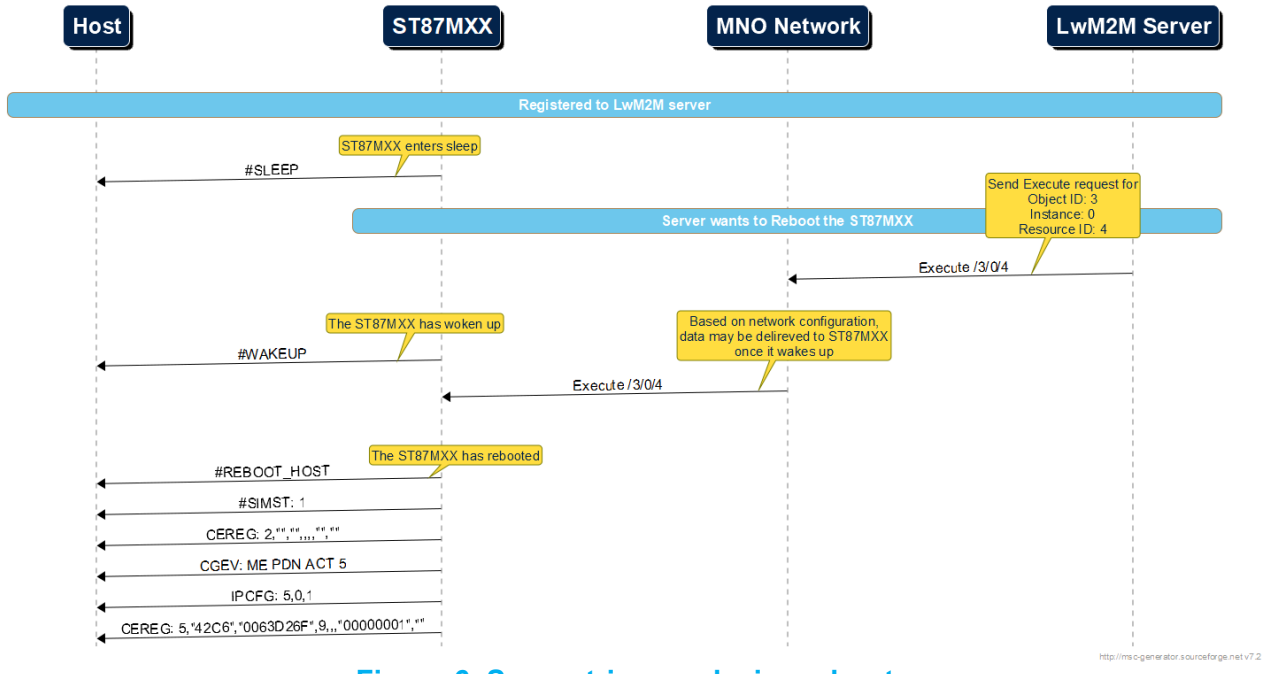

Figure 6. Server triggers device reboot

6.6.3 Terminal

```
AT#LWREG=1,1
OK
#LWREG: 1,1
#LWREG: 1,2
#LWREG: 1,3
```

```
#SLEEP

#WAKEUP
#REBOOT_HOST
#SIMST: 1
+CEREG: 2,"","",,,,"",""
+CEREG: 0,"","",,,,"",""
+CEREG: 2,"","",,,,"",""
+CGEV: ME PDN ACT 5
#IPCFG: 5,0,1
+CEREG: 5,"42C6","0063D26F",9,,,"00000001",""
```

6.7 LwM2M over DTLS connection

The ST87MXX also supports LwM2M over DTLS/UDP secure transport layer.

6.7.1 Parameters

Here the prerequisite:

- The host must provision the ST87MXX with all the cryptographic elements to set up a secure connection.
- The host must send the AT#SOCKETCREATE command specifying a <security_profile_id>.
- The host must explicitly create the DTLS connection using the AT#DTLSCONNECT command.

Note: LwM2M over DTLS scenario support only manual registration mode.

6.7.2 MSC

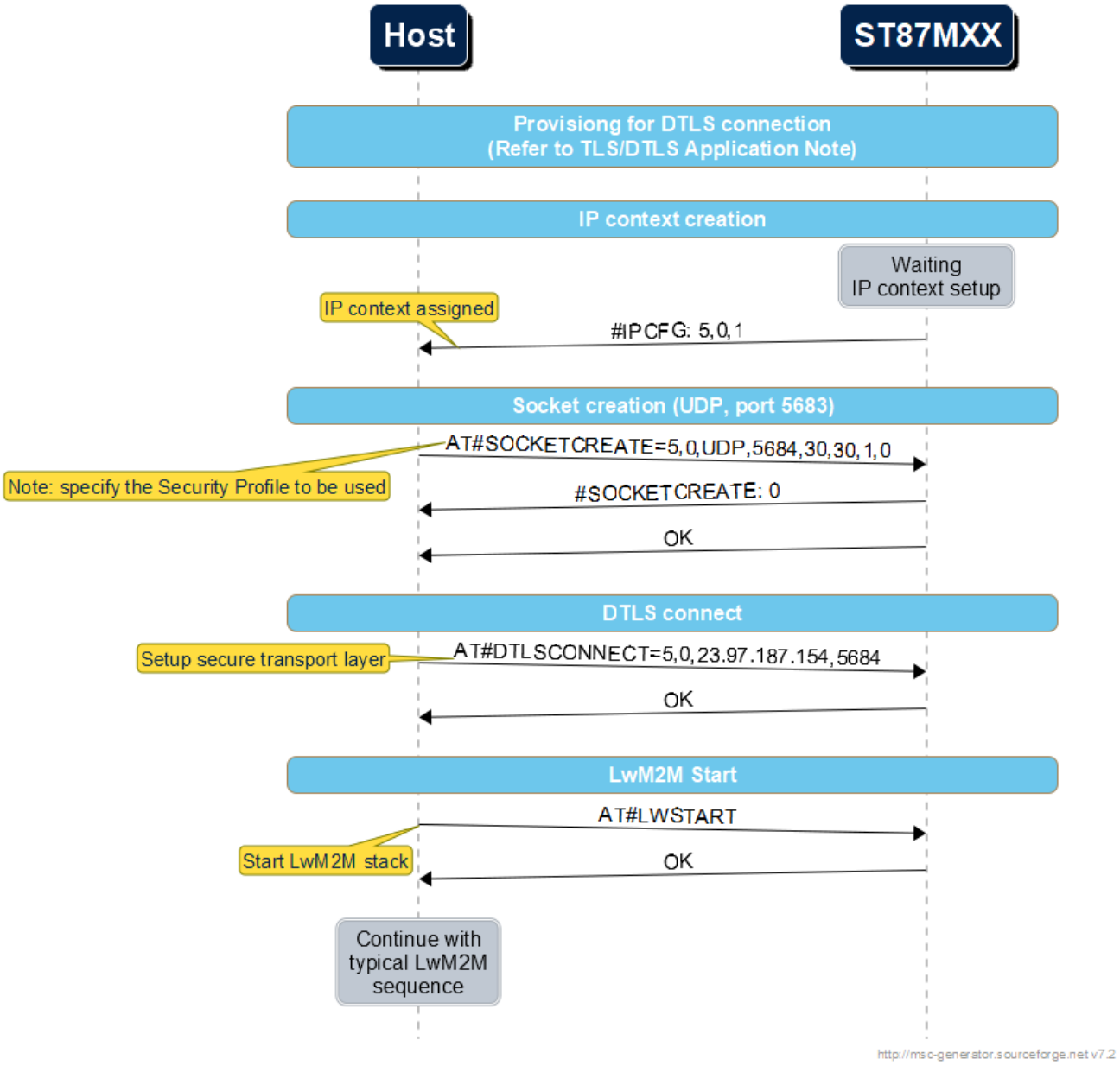

Figure 7. LwM2M over DTLS connection

6.7.3 Terminal

```
#IPCFG:5,0,1

AT#SOCKETCREATE=5,0,UDP,5684,30,30,1,0
#SOCKETCREATE: 0
OK

AT#DTLSCONNECT=5,0,192.168.21.112,5684
OK

AT#LWSTART
OK
```

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.
Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of
Purchasers’ products.
No license, express or implied, to any intellectual property right is granted by ST herein.
Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.
ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.
Information in this document supersedes and replaces information previously supplied in any prior versions of this document.