ST87MXX CoAP Application Note

Introduction

This document gives details of the AT Commands usage for embedded CoAP stack supported by the ST87MXX NB-IoT module.

Document status
Official

1 General information

1.1 Acronyms and terms

Table 1. Definitions of terms

Term	Definition
COAP	Constrained Application Protocol
UDP	User Datagram Protocol
REST	Representational State Transfer

1.2 Reference documents

The documents listed in Table 2 provide further information.

Table 2. Document references

Reference	Document
[1]	RFC7232 The Constrained Application Protocol (CoAP)
[2]	RFC7959 Block-Wise Transfers in the Constrained Application Protocol (CoAP)
[3]	ST87MXX UM AT commands description

1.3 Revision history

Table 3. Document revision history

Date	Version	Changes
2024-06-05	V1.0	- First release
2025-09-22	V2.0	- Update for release

1.4 Table of contents

1.5 List of tables

1.6 List of figures

2 Introduction

2.1 Overview

The Constrained Application Protocol (CoAP) [1] is a specialised web transfer protocol designed for nodes and networks with limited resources on the Internet of Things (IoT). It enables straightforward mapping to HTTP for seamless web integration while maintaining simplicity suitable for implementation on devices with limited battery power and memory.

CoAP is lightweight and specifically optimized for Machine-to-Machine (M2M) communication. It adheres to RESTful principles like HTTP, supporting standard methods such as GET, POST, PUT, and DELETE for resource manipulation. Additionally, CoAP supports multicast communication, enhancing efficiency in IoT deployments.

A defining characteristic of CoAP is its minimal protocol overhead, making it ideal for constrained environments. Utilizing UDP as its transport protocol reduces communication overhead compared to TCP, while built-in features ensure reliable message delivery, basic congestion control, and low-latency operation.

In summary, CoAP offers an efficient, scalable, and lightweight communication solution tailored for IoT applications, enabling devices with limited resources to communicate effectively over constrained networks.

The **Error! Reference source not found.** ilustrates the integration of the CoAP protocol stack into the ST87MXX.

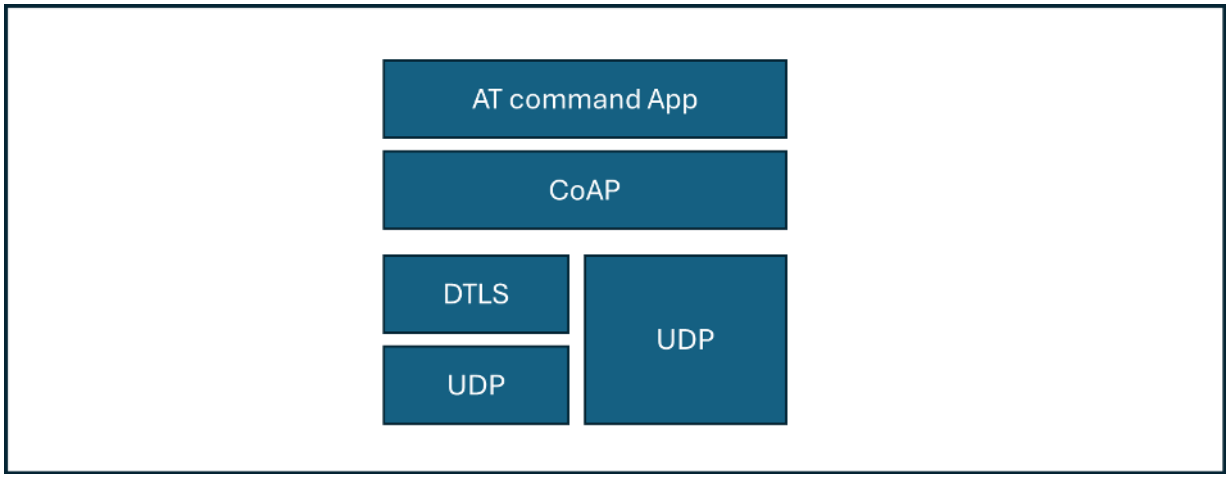


Figure 1. CoAP protocol stack

For a complete description of the AT commands and parameters, refer to the AT command user manual **Error! Reference source not found.**.

3 CoAP scenarios

3.1 CoAP configuration

To send CoAP requests to a server, the ST87MXX module should be properly configured. This chapter explains the steps the Host should perform:

- Socket creation
- CoAP stack initialization
- CoAP connection setup

3.1.1 Parameters

To create the socket, the Host should send the following command:

AT#SOCKETCREATE=<context_id>,<ip_version>,<socket_type>,[<local_port>],<send_timeout>,<receive_timeout>[,<frame_received_urc>,<security_profile_id>]

Example of IPv4 UDP Socket creation:

- AT#SOCKETCREATE=5,0,UDP,0,1,1

Where:

- **5**: is the context ID used for IP connection - 5 is the one used by default.
- **0:** for IPv4, 1: for IPv6 – it should be the same chosen during IP configuration.
- **UDP:** is the type of socket used to create it.
- **0:** is the automatic local port number configuration, set the value greater than 0 to specify port number.
- **1:** is the send timeout. Note: it is ignored by CoAP.
- **1:** is the receive timeout. Note: it is ignored by CoAP.

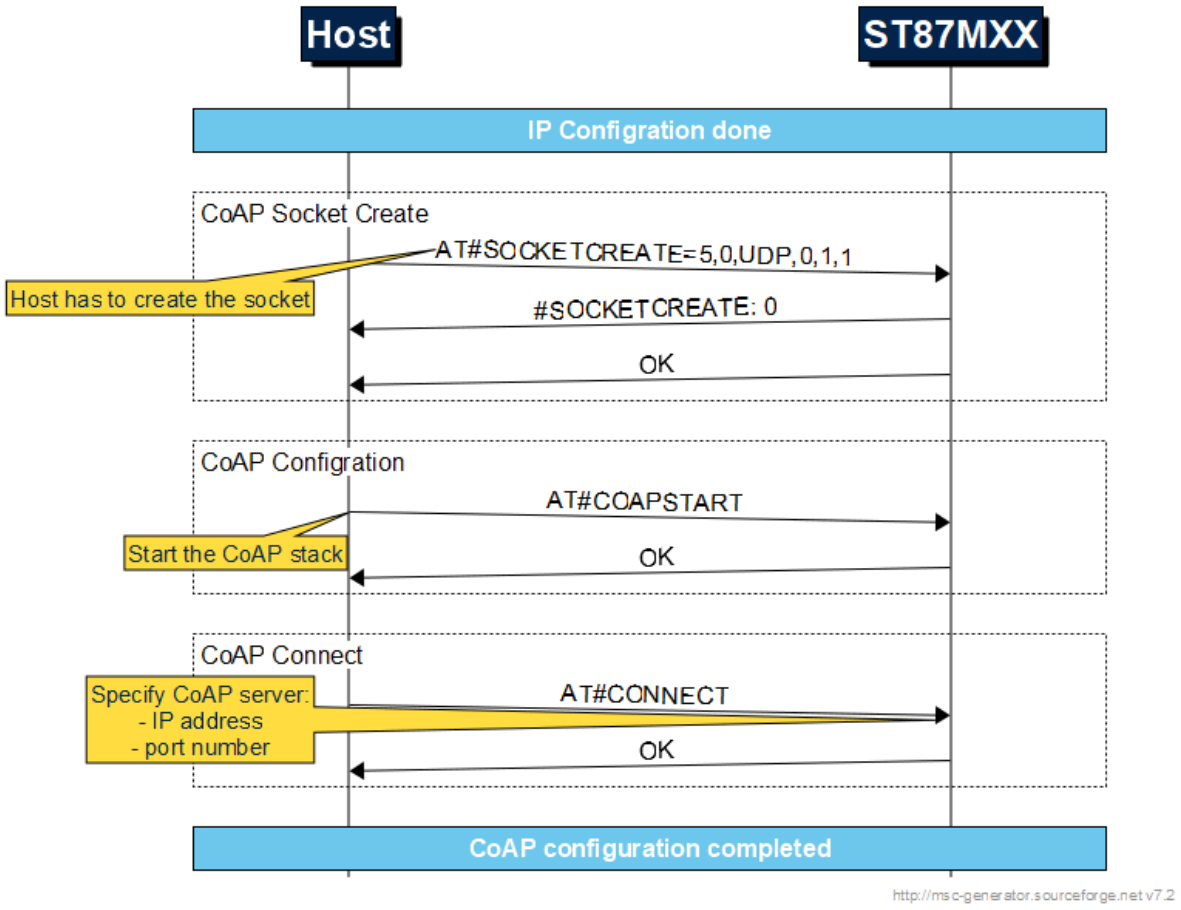
To start the CoAP protocol, the Host should send the command to initialize the CoAP stack:

- **AT#COAPSTART**

The Host should establish the connection with the endpoint by sending the following command:

- **AT#COAPCONNECT**

3.1.2 MSC



3.1.3 Terminal

Below is the sequence obtained in the UART terminal when executing the scenario (in **Bold** the commands sent to the module):

```
AT#SOCKETCREATE=5,0,UDP,0,1,1
#SOCKETCREATE: 0
OK

AT#COAPSTART;
OK

AT#CONNECT=5,0,,,
OK
```

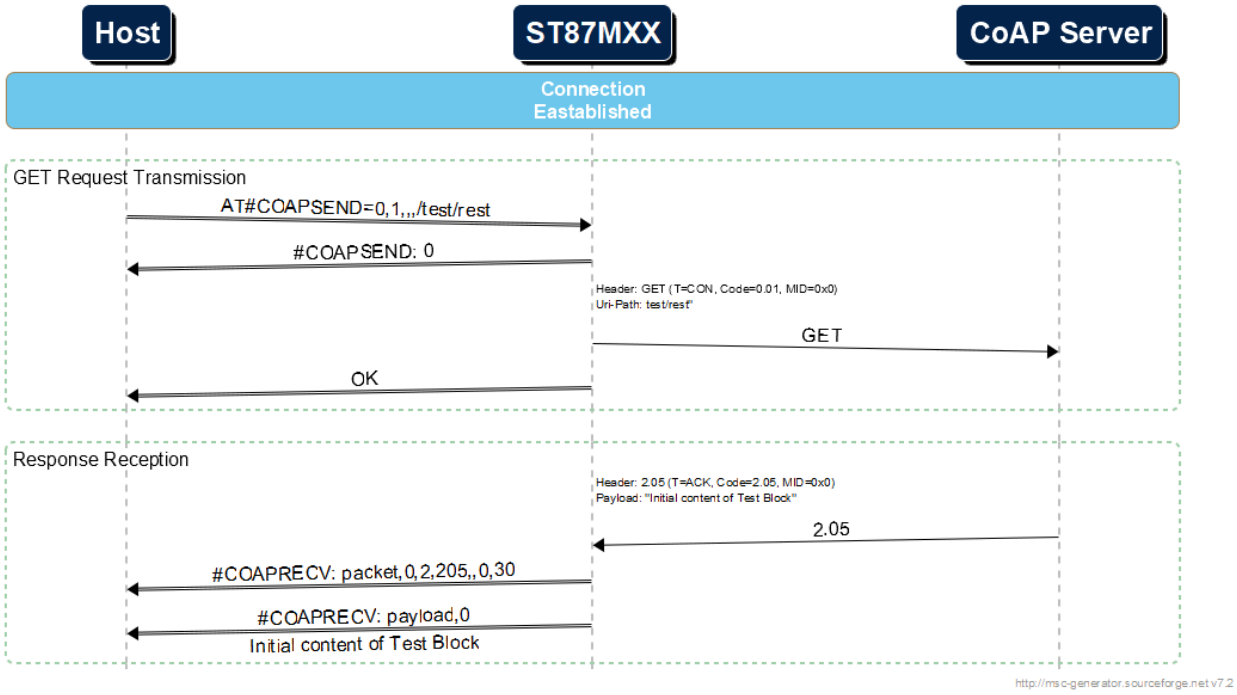
3.2 GET Request

This chapter provides an example of a CoAP GET request.

3.2.1 Parameters

To send a CoAP GET request, the Host should use the AT#COAPSEND command, specifying the <method> parameter with the value 1.

3.2.2 MSC



3.2.3 Terminal

Below is the sequence obtained in the UART terminal when executing the scenario (in **Bold** the commands sent to the module):

```
AT#COAPSEND=0,1,,,/test/rest
#COAPSEND: 0
OK

#COAPRECV: packet,0,2,205,,0,30
#COAPRECV: payload,0
```

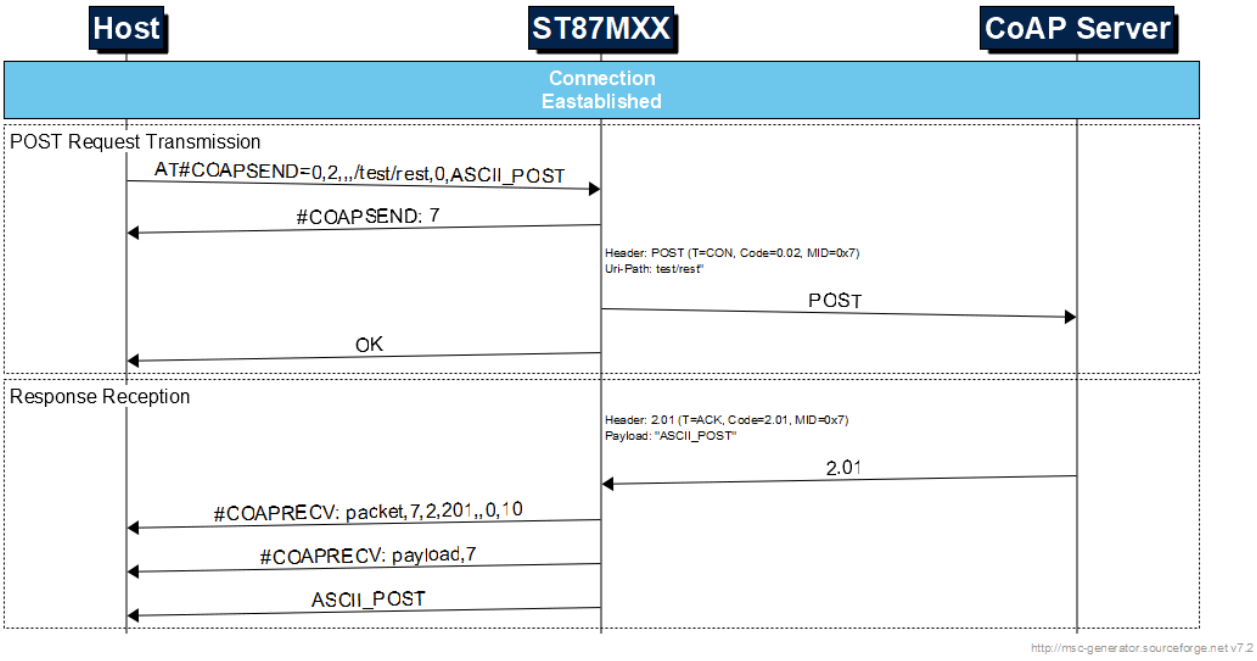
3.3 POST Request

This chapter provides an example of a CoAP POST request.

3.3.1 Parameters

To send a CoAP POST request, the Host must send the AT#COAPSEND command with the <method> parameter set to 2.

3.3.2 MSC



3.3.3 Terminal

Below is the sequence obtained in the UART terminal when executing the scenario (in **Bold** the commands sent to the module):

```
AT#COAPSEND=0,2,,,/test/rest,0,ASCII_POST
#COAPSEND: 7
OK

#COAPRECV: packet,7,2,201,,0,10
#COAPRECV: payload,7
ASCII_POST
```

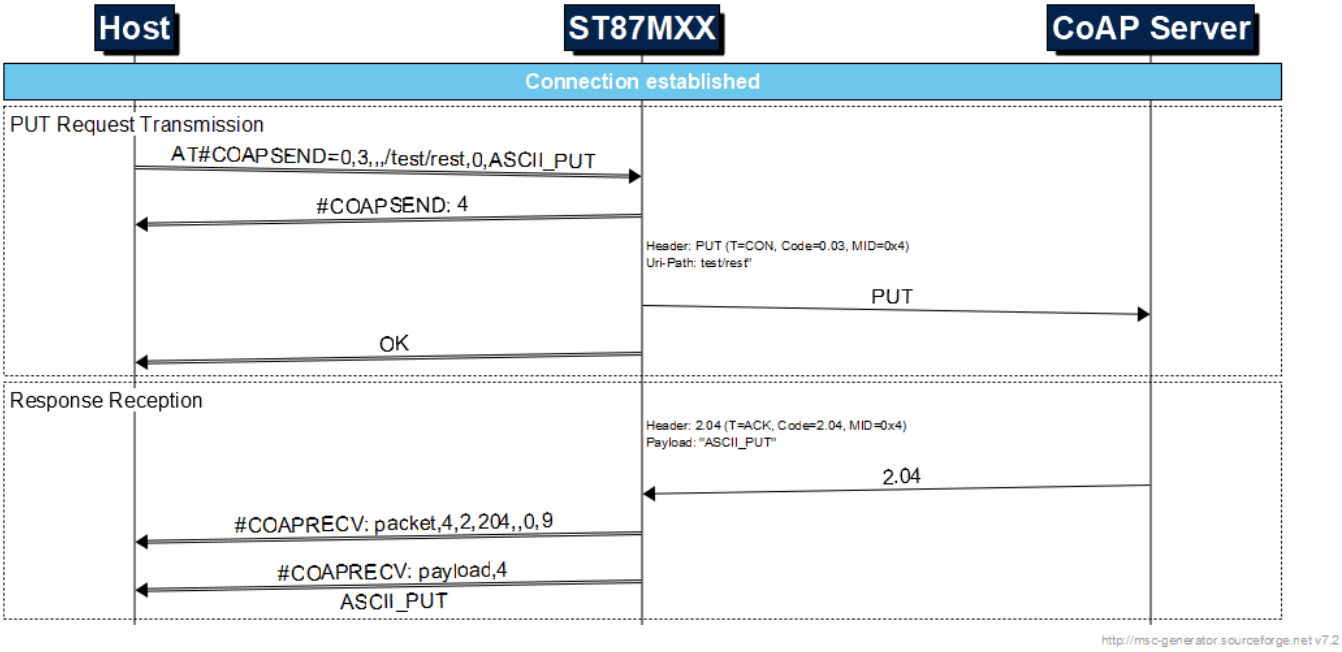
3.4 PUT Request

This chapter provides an example of a CoAP PUT request.

3.4.1 Parameters

To send a CoAP POST request, the Host must send the AT#COAPSEND command with the <method> parameter set to 3.

3.4.2 MSC



3.4.3 Terminal

Below is the sequence obtained in the UART terminal when executing the scenario (in **Bold** the commands sent to the module):

```
AT#COAPSEND=0,3,,,/test/rest,0,ASCII_PUT
#COAPSEND: 4
OK

#COAPRECV: packet,4,2,204,,0,9
#COAPRECV: payload,4
ASCII_PUT
```

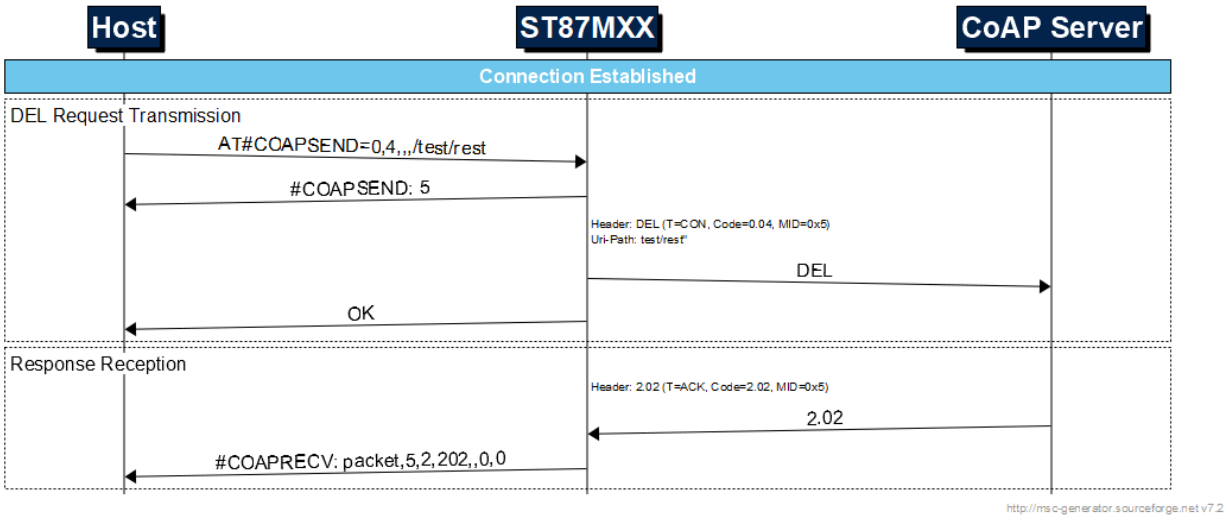
3.5 DEL Request

This chapter provides an example of a CoAP DEL request.

3.5.1 Parameters

To send a CoAP POST request, the Host must send the AT#COAPSEND command with the <method> parameter set to 4.

3.5.2 MSC



3.5.3 Terminal

Below is the sequence obtained in the UART terminal when executing the scenario (in **Bold** the commands sent to the module):

```
AT#COAPSEND=0,4,,,/test/rest
COAPSEND: 5
OK

#COAPRECV: packet,5,2,202,,0,0
```

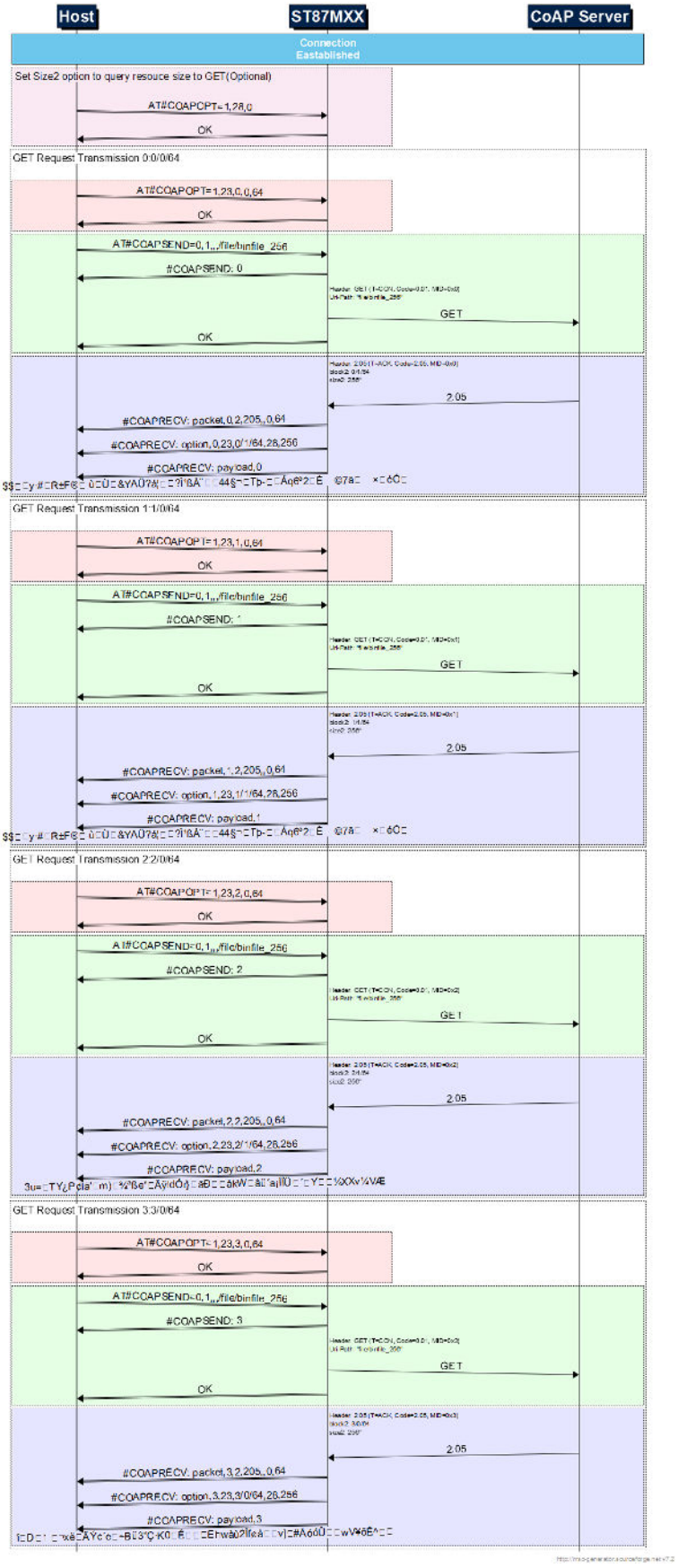
3.6 CoAP GET request with BLOCK2 option

This example shows a GET t using the Block2 option, which allows retrieving a resource block by block. For a detailed description of the Block2 option, please refer to [2].

3.6.1 Parameters

To use the Block2 option, the Host should set it by sending the AT#COAPOPT command with the <opt> parameter set to 23.

3.6.2 MSC


Host
ST87MXX
CoAP Server
Connection Eastablished
Set Size2 option to query resource size to GET(Optional)
AT#COAPOPT=1,28,0
OK
GET Request Transmission 0:0/0/64
AT#COAPOPT=1,23,0,0,64
OK
AT#COAPSEND=0,1,,,/file/binfile_256
#COAPSEND: 0
GET
OK
2.05
#COAPRECV: packet,0,2,205,,0,64
#COAPRECV: option,0,23,0,1,64,28,256
#COAPRECV: payload,0
GET Request Transmission 1:1/0/64
AT#COAPOPT=1,23,1,0,64
OK
AT#COAPSEND=0,1,,,/file/binfile_256
#COAPSEND: 1
GET
OK
2.05
#COAPRECV: packet,1,2,205,,0,64
#COAPRECV: option,1,23,1,1,64,28,256
#COAPRECV: payload,1
GET Request Transmission 2:2/0/64
AT#COAPOPT=1,23,2,0,64
OK
AT#COAPSEND=0,1,,,/file/binfile_256
#COAPSEND: 2
GET
OK
2.05
#COAPRECV: packet,2,2,205,,0,64
#COAPRECV: option,2,23,2,1,64,28,256
#COAPRECV: payload,2
GET Request Transmission 3:3/0/64
AT#COAPOPT=1,23,3,0,64
OK
AT#COAPSEND=0,1,,,/file/binfile_256
#COAPSEND: 3
GET
OK
2.05
#COAPRECV: packet,3,2,205,,0,64
#COAPRECV: option,3,23,3,0,64,28,256
#COAPRECV: payload,3

3.6.3 Terminal

Below is the sequence obtained in the UART terminal when executing the scenario (in **Bold** the commands sent to the module):

```
AT#COAPOPT=1,28,0
OK

AT#COAPOPT=1,23,0,0,64
OK

AT#COAPSEND=0,1,,,/file/binfile_256
#COAPSEND: 0
OK

#COAPRECV: packet,0,2,205,,0,64
#COAPRECV: option,0,23,0/1/64,28,256
#COAPRECV: payload,0
$$□y·#R±F®□ ùÙ□&YAÜ?á¦^□?Ì¹ßÅ¨□44§¬□Tþ-□Áq6°2È     ©7ã□     ×□éÓ


AT#COAPOPT=1,23,1,0,64
OK

AT#COAPSEND=0,1,,,/file/binfile_256
#COAPSEND: 1
OK

#COAPRECV: packet,1,2,205,,0,64
#COAPRECV: option,1,23,1/1/64,28,256
#COAPRECV: payload,1
$$□y·#R±F®□ ùÙ□&YAÜ?á¦^□?Ì¹ßÅ¨□44§¬□Tþ-□Áq6°2È     ©7ã□     ×□éÓ


AT#COAPOPT=1,23,2,0,64
OK

AT#COAPSEND=0,1,,,/file/binfile_256
#COAPSEND: 2
OK

#COAPRECV: packet,2,2,205,,0,64
#COAPRECV: option,2,23,2/1/64,28,256
#COAPRECV: payload,2
3u=□TY¿P¢Ìã'□m)□¾ªße"□Âÿ1dÓr\}□ãÐ□åkW□âü´a¡ÍÎÜ□´□Y¼XXv¼VÆ


AT#COAPOPT=1,23,3,0,64
OK

AT#COAPSEND=0,1,,,/file/binfile_256
#COAPSEND: 3
OK

#COAPRECV: packet,3,2,205,,0,64
#COAPRECV: option,3,23,3/0/64,28,256
#COAPRECV: payload,3
î□D1 □¬xë□ÃÝc´e□~Bü3"Ç·K0□É□□□Ehwàù2Îføåv]#ÅóóÛwV¥õË^□□
```