ST87MXX Firmware Update Application Note

Purpose and scope

This document describes the procedure to update the firmware of the ST87MXX NB-IoT module using the bootloader.

Document status
Official

1. General information

1.1 Acronyms and terms

Table 1. Acronyms

Term	Definition
ATE	Automatic Test Equipment
GUI	Graphical User Interface
PC	Personal Computer
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter

1.2 Reference documents

The documents listed in Table 2 provide further information.

Table 2. Document references

Reference	Document
[1]	ST87MXX_SPI_Flash_Application_Note

1.3 Revision history

Table 3. Document revision history

Date	Version	Changes
2024-06-30	V1.0	- Official version
2025-09-11	V2.0	- Update for release

1.4 Table of contents

1.5 List of tables

1.6 List of figures

2. Overview

2.1 Introduction

The ST87MXX module is pre-programmed by STMicroelectronics during the manufacturing process with a default firmware version. However, customers may need to update this firmware to a newer version to incorporate functional improvements, additional features, or security enhancements.
Firmware updates can be performed using the ST87MXX graphical user interface (GUI), which is primarily intended for development and testing purposes. For production environments requiring automation and scalability, STMicroelectronics provides a command-line boot tool designed to efficiently manage firmware updates. This document describes a firmware update procedure tailored for industrial automation, enabling reliable and efficient mass programming.

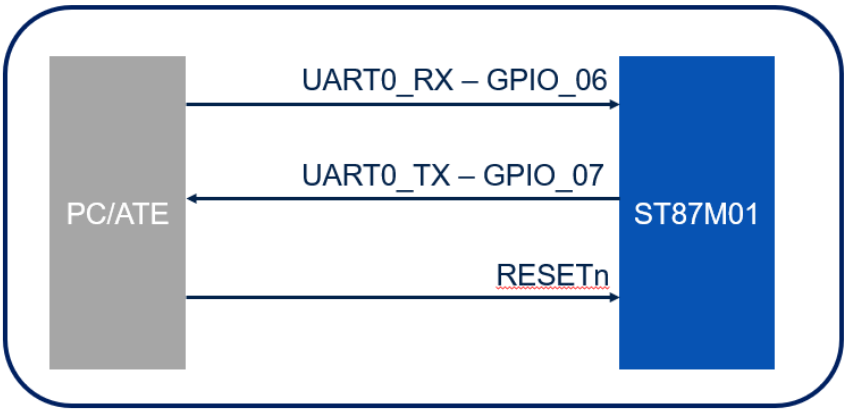
3. Hardware Configuration

The ST87MXX firmware update can be performed via two interfaces: UART and SPI.

3.1 UART Setup

The firmware update through the UART interface is supported via the UART0 TX and RX pins, as shown in the figure below.

Figure 1. UART Connection for Firmware Update

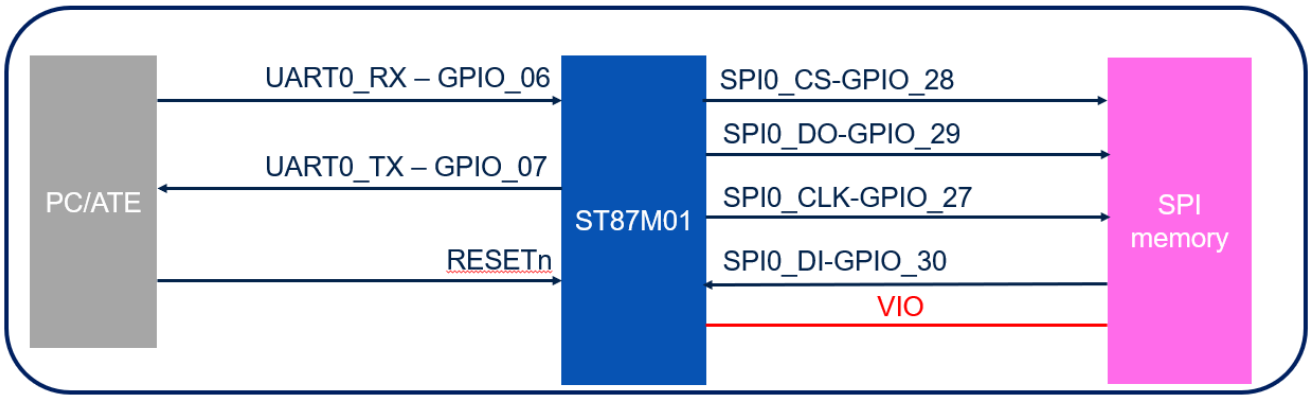


Note: The UART0 hardware flow control lines (RTS/CTS) are not required for the update.

3.2 SPI Setup

The firmware update via the SPI interface uses the SPI0 DI, DO, CS, and CLK pins, as illustrated in the figure below.

Figure 2. SPI Connection to External Flash Memory



In this configuration, the PC/ATE is connected directly to the ST87MXX module via UART for control and monitoring, while the external SPI flash memory interfaces directly with the module.

This setup reduces firmware loading time by enabling the ST87MXX module to autonomously retrieve the firmware image from the external SPI memory once the process is initiated by the PC/ATE.

3.2.1 List of supported SPI memories

The following SPI flash memory devices have been validated and are supported for use with the ST87MXX module firmware update process via the SPI interface:

- **Micron**
  - MT25QL128ABA
- **Winbond**
  - W25Q16FW
  - W25Q16JV-IQ/JQ
  - W25Q16JV-IM/JM
- **Atmel**
  - AT45D161D

4. Firmware Update Procedure

The firmware of the ST87MXX module can be updated via two methods: UART mode and SPI mode. The following sections provide detailed step-by-step instructions for each method.

4.1 UART Mode

In UART mode, the PC/ATE initiates the firmware update process and transfers the firmware image to the ST87MXX module. STMicroelectronics provides a tool called the **Boot Tool** to help with this process. This tool can be controlled via command-line instructions, making it suitable for automated factory environments. The UART firmware update procedure using the Boot Tool is as follows:

1. **Set the ST87MXX in Boot mode:**
   - Execute the command:
     - st8700_boot_tool.exe **at** <port>
     - **Parameters**:
       1. <port>: The serial port identifier used by the PC/ATE.
     - **Example**: st8700_boot_tool.exe at COM16
     - **Note**: This step may require resetting the ST87MXX module by toggling the RESETn line while the command is running.

   - Alternatively, this PC/ATE can:
     - Assert and release the RESETn line.
     - Continuously send synchronization byte 0xC5 via UART at **115200** baud rate until acknowledgment byte 0x9A is received.
     - **Note**: The synchronization byte should be sent immediately after releasing RESETn, within a 100-millisecond window, as the Bootloader waits for it before launching the default firmware.

2. **(Optional) Verify Boot Mode:**
   - Send the command over UART at **115200** baud rate to read the Bootloader version:
     - ATI01\r
   - Wait for the module response, for example:
     - ST8700 boot vX.X\r\n0\r
     - **Note**: vX.X indicates the Bootloader version number. The response ends with 0\r, indicating successful communication.

3. **Start the Firmware Upload:**
   - Execute the command:
     - st8700_boot_tool.exe **upload -b** <baudrate> <port> <image>
     - **Parameters**:
       1. <baudrate>: UART baud rate used during the upload. The maximum standard supported baud rate is **460800**. Additionally, a non-standard baud rate of 895556 is also supported.
       2. <port>: The serial port identifier used by the PC/ATE.
       3. <image>: Firmware image file to upload (with .img extension), specific to the ST87MXX module.
     - **Example**: st8700_boot_tool.exe upload -b 460800 COM16 ST87MXX1_A_x_x_x.img
     - **Note:** The Boot Tool automatically configures and opens the specified serial port.

4. **Wait for the Firmware upload to complete:**
   - On success, the PC/ATE receives exit code 0 followed by a carriage return (0x0D).

- On failure, a non-zero error code (e.g., 4004\r) followed by a carriage return (0x0D) is returned to the PC/ATE.

4.2 SPI Mode

In SPI mode, the ST87MXX module loads the firmware image itself from an external SPI flash memory. The PC/ATE only needs to start and monitor the update process.

The SPI firmware update procedure is as follows:

1. **Set the ST87MXX in Boot mode:**
   - Execute the command line:
     - st8700_boot_tool.exe **at** <port>
     - **Parameters**:
       1. <port>: The serial port identifier used by the PC/ATE.
     - **Example**: st8700_boot_tool.exe at COM16
     - **Note**: This step may require resetting the ST87MXX module by toggling the RESETn line while the command is running.
   - Alternatively, this PC/ATE can:
     - Assert and release the RESETn line.
     - Continuously send synchronization byte 0xC5 via UART at **115200** baud rate until acknowledgment byte 0x9A is received.
     - **Note**: The synchronization byte should be sent immediately after releasing RESETn, within a 100-millisecond window, as the Bootloader waits for it before launching the default firmware.
2. **(Optional) Verify Boot Mode:**
   - Send the command over UART at **115200** baud rate to read the Bootloader version:
     - ATI01\r
   - Wait for the module response, for example:
     - ST8700 boot vX.X\r\n0\r
     - **Note**: vX.X indicates the Bootloader version number. The response ends with 0\r, indicating successful communication.
3. **Start the Firmware Upload:**
   - Send the command at **115200** baud rate to initiate the firmware upload:
     - AT#SPIUP=<address>\r.
     - **Parameters**:
       1. <address>: Starting address in SPI memory for the firmware image, expressed as a hexadecimal string without the 0x prefix.
     - **Example**: AT#SPIUP=10000\r
     - **Note**: The command should end with a carriage return character (0x0D).
4. **Wait for the Firmware upload to complete**
   - On success, the PC/ATE receives exit code 0 followed by a carriage return (0x0D).
   - On failure, a non-zero error code (e.g., 4004\r) followed by a carriage return (0x0D) is returned to the PC/ATE.

4.2.1 External SPI Flash Memory Programming

This section explains how to program the external SPI flash memory using the ST87MXX module, a specific firmware image called SPI flash loader, and Python scripts provided by STMicroelectronics. The

spi_flash_loader_key04.img file is this firmware image. It allows the ST87MXX module to connect to the external SPI flash memory and program it. Loading this image into the module allows the SPI flash memory to be erased and programmed under the control of the PC/ATE.

To program the external SPI flash memory, follow these steps:

1. **Load the spi_flash_loader_key04.img file into the ST87MXX module**
   - Use the UART programming procedure described in section 4.1.

2. **Launch the execution of the loaded spi_flash_loader_key04.img**
   - Send the following command over UART at **115200** baud rate:
     - AT#BOOTRAM\r
     - **Note**: The command should end with a carriage return character (0x0D).

3. **If necessary, erase the SPI flash memory using the Python script:**
   - Execute the command line:
     - **python** spi_flash_erase.**py** <address> <size>
     - **Parameters:**
       1. <address>: Starting address in SPI memory for the firmware image, expressed as a hexadecimal string without the 0x prefix.
       2. <size>: Size of the area to erase, , expressed as a hexadecimal string without the 0x prefix.
     - **Example:** python spi_flash_erase.py 10000 20000

4. **Program the SPI flash memory using the Python script:**
   - Execute the command line:
     - **python** spi_flash_program.**py** <address> <image>
     - **Parameters:**
       1. <address>: Starting address in SPI memory for the firmware image, expressed as a hexadecimal string without the 0x prefix.
       2. < image>: Firmware image file to upload (with .img extension), specific to the ST87MXX module.
     - **Example:** python spi_flash_program.py 10000 ST87MXX1_A_x_x_x.img

Note: The ST87MXX module should not be reset during these steps. If a reset occurs, the procedure must be restarted from step 1.

5. Image Bundle Generation

If partial firmware updates of the module are required, multiple firmware images can be combined into a single file called a **bundle image**. This approach simplifies the programming process by enabling the upload of a single consolidated file to the ST87MXX module.

5.1 Procedure to Create the Bundle Image

To create and program a bundle image, follow these steps:

1. **Prepare the required files:**
   - Collect the individual firmware image files provided by STMicroelectronics (e.g., ST87MXX_A_x_x_x.img, …).
   - Ensure the STMicroelectronics Boot Tool executable (st8700_boot_tool.exe) is available.
   - Copy all these files into a directory with no space in its path (e.g., C:\temp).

2. **Generate the bundle image:**
   - Execute the command line:
     - st8700_boot_tool.exe **bundle** <image_1> ... <image_N> **-o** <bundle_image>
     - **Parameters**:
       1. <image_1> ... <image_N>: List of firmware image files to be combined.
       2. <bundle_image>: Name of the resulting bundle .img file.
     - **Example**: st8700_boot_tool.exe bundle ./ST87MXX_A_x_x_x.img ./ST87MXX_M_x_x_x.img -o ./ST87MXX_A_x_x_x_M_x_x_x_.img

3. **Program the bundle image into the module:**
   - Use the standard firmware update procedure described in sections 4.1 (UART mode) or 4.2 (SPI mode) to program the generated bundle image into the ST87MXX module.