

Turbo Writer Tool User Guide

Document Information

Abstract	Introduce how to use Turbo Writer for N329x & N9H2x series microprocessor (MPU).
Apply to	N329 series & N9H26 series

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

Table of Contents

1	INTRODUCTION	3
1.1	ISP Introduction	3
2	TURBO WRITER INTRODUCTION	4
2.1	Tools\PC_Tools\TurboWriter V2.30.003_N9H26K6	4
2.2	Turbo Writer extra support NAND ID	4
2.3	Turbo Writer extra support SPI Flash ID	4
2.4	Turbo Writer Configuration for Boot Code Header	5
2.5	Turbo Writer Firmware	6
2.6	Turbo Writer Tool Version	7
3	PRELIMINARY DEFINITION	8
3.1	Image Type	8
4	EXAMPLE	9
4.1	Connect to Turbo Writer	10
4.2	SDRAM	10
4.3	NAND Flash	12
4.4	SD	16
4.5	SPI	20
4.6	SPI (Raw Data)	24
5	SUPPORTING RESOURCES	28

1 Introduction

1.1 ISP Introduction

N329 / N9H2X series have two boot flows – one is Normal mode; the other is Recovery mode. If user wants to do ISP (In System Programming) through USB, the system should be set to Recovery mode. The boot flow descriptions are as follows

- N3290x
 - Normal mode boot flow
 - ◆ SD card 0 boot → NAND boot → SPI boot → SD card 1 boot → USB boot
 - Recovery mode boot flow
 - ◆ USB boot
- N3291x
 - Normal mode boot flow
 - ◆ SD card 0 boot → SPI boot (GPA[15:12]/NAND) → NAND CS 0 boot → NAND CS1 boot → SPI boot (GPD[15:12]) → SD card 1 boot → SD card 2 boot → USB boot
 - Recovery mode boot flow
 - ◆ USB boot
- N3292x
 - Normal mode boot flow
 - ◆ SD card 0 boot → NAND CS 0 boot → NAND CS1 boot → SPI boot → SD card 1 boot → SD card 2 boot → USB boot
 - Recovery mode boot flow
 - ◆ USB boot
- N9H20
 - Normal mode boot flow
 - ◆ SD card 0 boot → NAND boot → SPI boot → SD card 1 boot → USB boot
 - Recovery mode boot flow
 - ◆ USB boot
- N9H26
 - Normal mode boot flow
 - ◆ SD card 0 boot → NAND CS 0 boot → NAND CS1 boot → SPI boot → SD card 1 boot → SD card 2 boot → USB boot
 - Recovery mode boot flow
 - ◆ USB boot

2 Turbo Writer Introduction

Turbo Writer can program the images into specific flash or download to SDRAM and run. This document will introduce user how to program the application code into N329x / N9H2x demo board NAND flash, SD Card, or SPI Flash by Turbo Writer. The following lines describe the files within the Turbo Writer folder.

2.1 Tools\PC_Tools\TurboWriter V2.30.003_N9H26K6

<i>ChangeLog.txt</i>	All Turbo Writer tool and Turbo Writer firmware change history is listed in this file.
<i>Turbowriter.exe</i>	Turbo Writer PC Tool execution file
<i>NAND ID.ini</i>	Turbo Writer extra support NAND ID
<i>SPIFLASH ID.ini</i>	Turbo Writer extra support SPI Flash ID
<i>TurboWriter.ini</i>	Turbo Writer Configuration for Boot Code Header
<i>xxxx_musb.bin</i>	Turbo Writer Firmware

2.2 Turbo Writer extra support NAND ID

User can add new NAND ID that Turbo Writer doesn't support without modifying Turbo Writer firmware by modifying *NAND ID.ini*. For example, if user wants to add H27UAG8T2A to Turbo Writer NAND list without modifying Turbo Writer firmware, user can add the following information into *NAND ID.ini*.

[1stID]	[2ndID]	[3rdID]	[4thID]	[5thID]	[Page per Block]	[Block per Flash]
AD	D5	94	25	44	128	1024

2.3 Turbo Writer extra support SPI Flash ID

If SPI Flash ID isn't in the support list, the read function can't work in SPI (Raw Data) mode. User can add new SPI Flash ID that without modifying Turbo Writer firmware by modifying *SPIFLASH ID.ini*. For example, if user wants to add new SPI Flash to Turbo Writer SPI Flash list without modifying Turbo Writer firmware, user can add the following information into *SPIFLASH ID.ini*.

[ID]	[SIZE]
C84017	8192

2.4 Turbo Writer Configuration for Boot Code Header

IBR supports some user-defined items in Boot code header about boot code executing address, DRAM clock skew setting, Register writing function, and delay function. After confirming the flash content is for booting, IBR will do corresponding operation according to the boot code header before reading boot code to destination address. Using the boot code header, user can change clock setting, dram setting, engine clock, and so on. The file TurboWriter.ini is for NAND, SD, SPI flash and SPI flash raw data.

- [ADDRESS]
 - Boot code execution address. It can be modified if necessary. Generally, it is the execution address of Loader (SD/NAND/SPI).

Chip	RAM Size	Execution address
N9H20K5	32 MB DRAM	0x900000.
N9H20K3	8 MB DRAM	0x700000.
N9H20K1	2 MB SDRAM	0x180000.
N32905	32 MB DRAM	0x900000.
N32903	8 MB DRAM	0x700000.
N32901	2 MB SDRAM	0x180000.
N3291x	32 MB DRAM	0x900000.

- [CLOCK_SKEW]
 - DQSODS /CKDQSDS values
- [xxxx USER_DEFINE]
 - User-defined setting for specified chip

USER_DEFINE	Chip
[N3290 USER_DEFINE]	N3290
[N3291 USER_DEFINE]	N3291
[N3292 USER_DEFINE]	N3292
[N9H20 USER_DEFINE]	N9H20
[N9H26 USER_DEFINE]	N9H26

- User-defined setting - Register Writing & Delay function
 - ◆ Register Writing function
 - Format : Address = Value
For example, Set REG_CKDQSDS to 0x00CCDD00
B0003034 = 00CCDD00
 - ◆ Delay function
 - Format : 55AA55AA = Delay time (micro second)
For example, Set Delay 1 second
55AA55AA = 000F4240

2.5 Turbo Writer Firmware

Turbo Writer is connected to Turbo Writer firmware which loaded from PC to N329 / N9H2x series by USB boot. When Turbo Writer connects to USB boot, Turbo Writer will send the corresponding binary file (Turbo Writer firmware) to N329 / N9H2x series and connects to Turbo Writer firmware according to USB boot PID & USB Host (*FA9x_musb.bin* / *N9H2x_musb.bin* for high speed and *FA9x_musb_FullSpeed.bin* / *N9H2x_musb_FullSpeed.bin* for full speed).

- Firmware file name

Chip	Firmware File Name
N3290	<i>FA93_musb.bin</i> / <i>FA93_musb_FullSpeed.bin</i>
N3291	<i>FA95_musb.bin</i> / <i>FA95_musb_FullSpeed.bin</i>
N3292	<i>FA92_musb.bin</i> / <i>FA92_musb_FullSpeed.bin</i>
N9H20	<i>N9H20_musb.bin</i> / <i>N9H20_musb_FullSpeed.bin</i>
N9H26	<i>N9H26_musb.bin</i> / <i>N9H26_musb_FullSpeed.bin</i>

- Turbo Writer firmware binary file includes

```
00000000h: 4E 56 54 20 00 00 A0 01 A8 D4 02 00 05 29 04 21 ; NVT ..?(元...) .!  
00000010h: 18 F0 9F E5 18 F0 9F E5 18 F0 9F E5 18 F0 9F E5 ; .????????  
00000020h: 18 F0 9F E5 18 F0 9F E5 18 F0 9F E5 18 F0 9F E5 ; .????????
```

- Turbo Writer firmware marker : 0x2054564E
- Turbo Writer firmware executing address : 0x001A00000
- Turbo Writer firmware size : 0x02D4A8
- Turbo Writer firmware version : 0x21042905 → 2021/10/28 V5

- ◆ It can be seen from Turbo Writer Tool as follows.

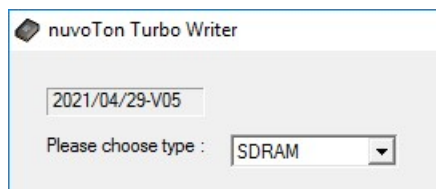


Figure 2-1 Turbo Writer Firmware version

- Turbo Writer firmware code starts from offset 0x20.

2.6 Turbo Writer Tool Version

Turbo Writer PC tool version can be seen from Turbo Writer Tool as follows.

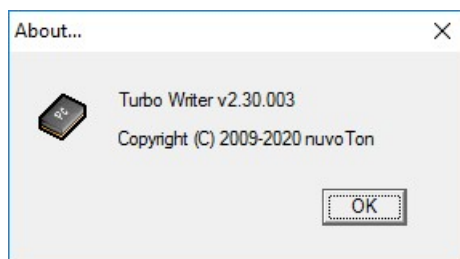


Figure 2-2 Turbo Writer Tool version

3 Preliminary Definition

3.1 Image Type

Turbo Writer defines five image types. The detailed descriptions are as follows.

- **System image**
 - Required image – ID is 0x03
 - It may be NAND / SD / SPI Loader
 - Loaded and Executed by Internal Boot Code (IBR)
 - Execution address is defined in TurboWriter.ini

Other images are loaded by NAND / SD / SPI Loader

- **Execute image**
 - Optional image – ID is 0x01
 - It may be NvtLoader for SD / NAND or application code
 - Loader will load image to execution address and run it.
- **Logo image**
 - Optional image – ID is 0x04
 - Logo binary file for display
- **Data image**
 - Optional image – ID is 0x00
 - Data image for user
- **RomFS image**
 - Optional image – ID is 0x02
 - Linux RomFS

4 Example

Take N9H26 demo flow to describe how to use Turbo Writer to burn images to execute Application Code. The following lines are described the image type for the demo flow.

- **System image**
 - NAND / SD Loader
 - ◆ *N9H26_NANDLoader_240MHz_Fast.bin*
 - ◆ *N9H26_SDLoader_240MHz_Fast.bin*
 - Execution address is 0x900000
- **Execute image**
 - NvtLoader for NAND / SD
 - ◆ *NVT_NAND_Fast_FW050TFT_800x480_24B.bin* for NAND
 - ◆ *NVT_SDU0_Fast_FW050TFT_800x480_24B.bin* for SD
 - Execution address is 0x800000
- **Logo image**
 - Logo binary file for display
 - ◆ *Logo.bin*
 - Display buffer address is 0x500000
- **Data image**
 - Optional image – ID is 0x00
 - Data image for user

N9H26 Booting flow from IBR to Application Code is as follows.

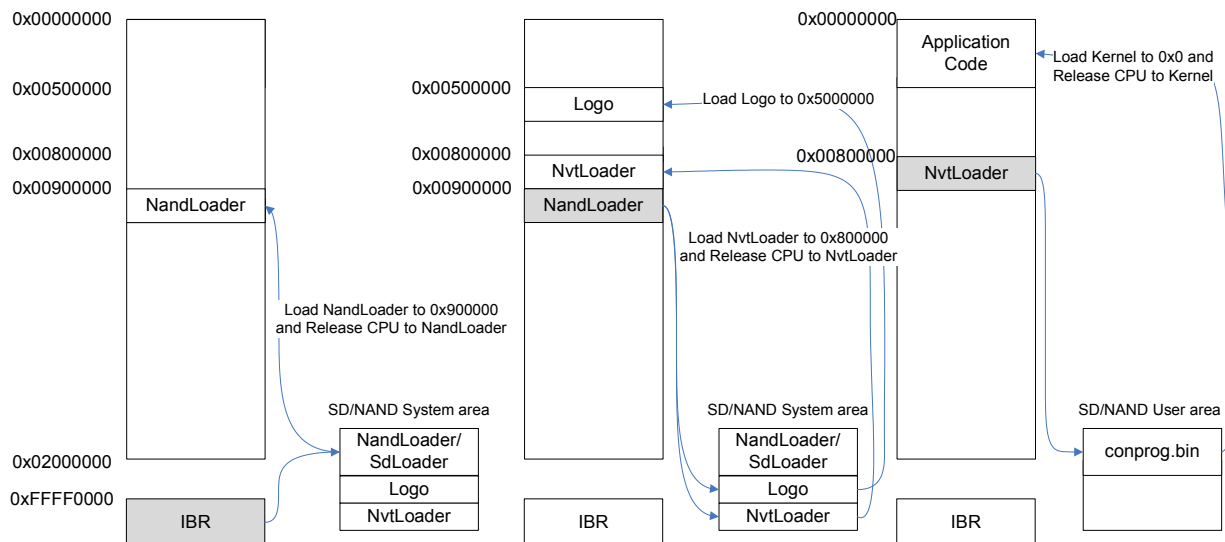


Figure 4-1 Demo booting flow for NAND / SD

4.1 Connect to Turbo Writer

Please following the step to connect to Turbo Writer.

1. Set the N329x / N9H26 to Recovery mode.
2. Coeeent to host through USB
3. Power on or reset N329x / N9H26
4. Execute Turbo Wrtier
5. Wait Turbo Wrtier connected
 - Turbo Wrtier will show “Mass Storage Connected!” and the firmware version

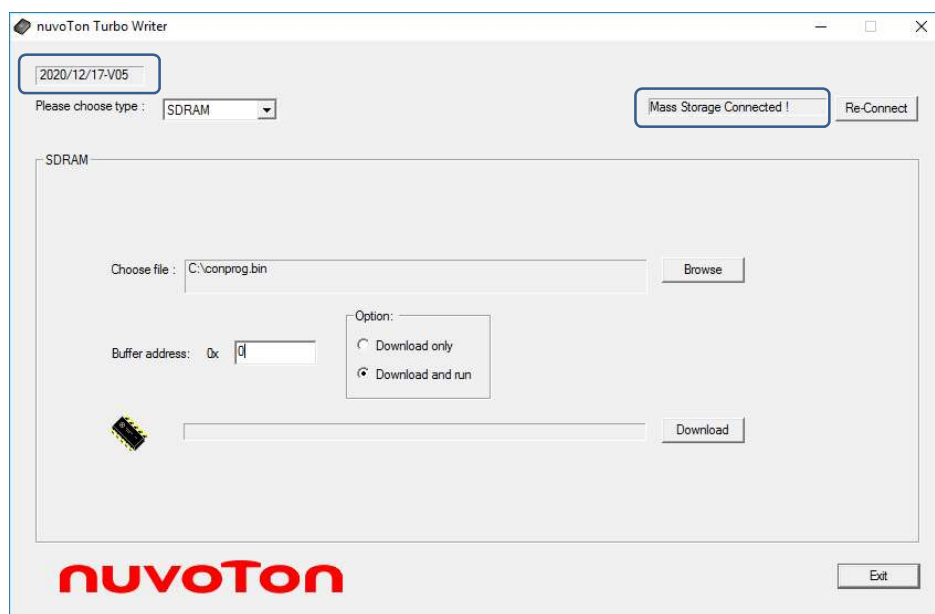


Figure 4-2 Connect to Turbo Writer

4.2 SDRAM

Turbo Writer SDRAM mode can download the application code and then execute it. For example, download Application Code file “*ConProg.bin*” to DRAM and execute it.

1. Choose the type “**SDRAM**”
2. Browse the file “*conprog.bin*”
3. Set the buffer address: 0x0
 - Application Code runs at 0x0.
4. Select the Option: Download and run
5. Press the button “**Download**”

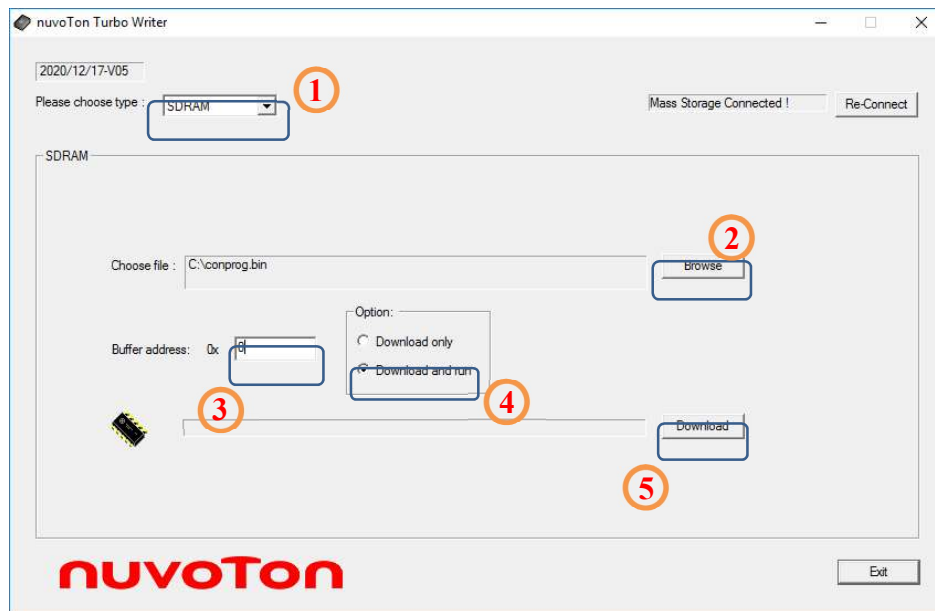


Figure 4-3 Download *conprog.bin* to SDRAM and Run

4.3 NAND Flash

The following lines describe how to use NAND Flash mode to burn images for Application Code.

- NAND Loader
 1. Choose the type “**NAND**”
 2. Press the button “**Add New**”
 3. Browse the file “*N9H26_NANDLoader_240MHz_Fast.bin*”
 4. Set Image type “**System Image**”
 5. Press the button “**Burn**”

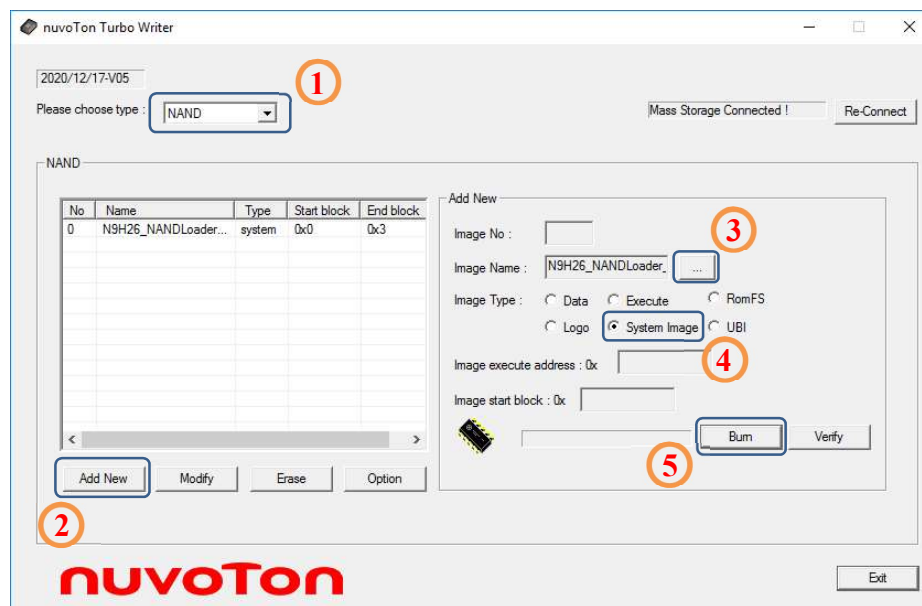


Figure 4-4 System image – NAND Loader

- Logo
 1. Image number “1”
 2. Browse the file “*Logo.bin*”
 3. Set Image type “**Logo**”
 4. Set the image execute address: 0x500000
 5. Set the start block number: 0x4
 - Because the burned NAND Loader occupies block 0~3, so we could select block 4 to burn the logo file.
 6. Press the button “**Burn**”

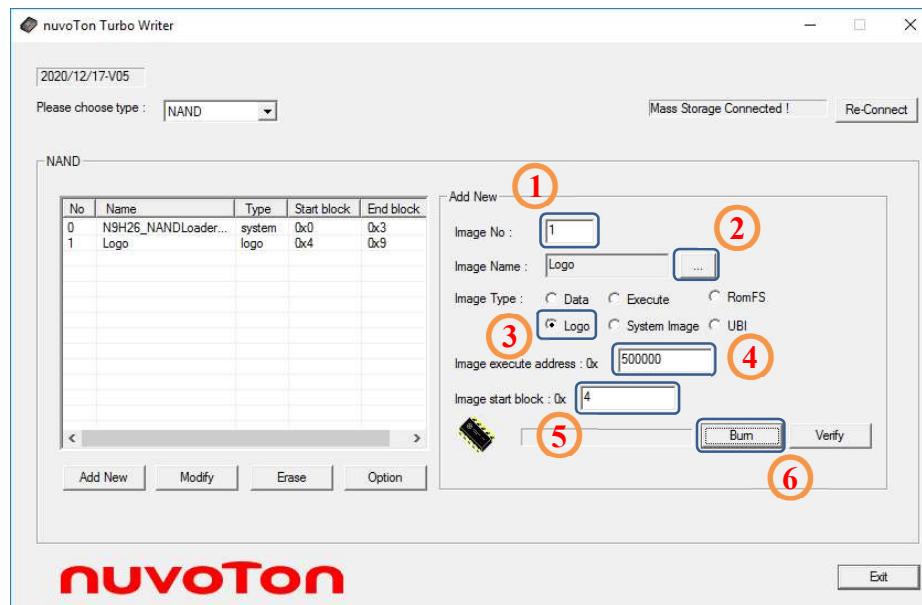


Figure 4-5 Logo image

- NVT Loader

1. Image number "2"
2. Browse the file "NVT_NAND_Fast_FW050TFT_800x480_24B.bin"
3. Set Image type "**Execute**"
4. Set the executed address: 0x800000
5. Set the start block number: 0xA.
6. Press the button "**Burn**"

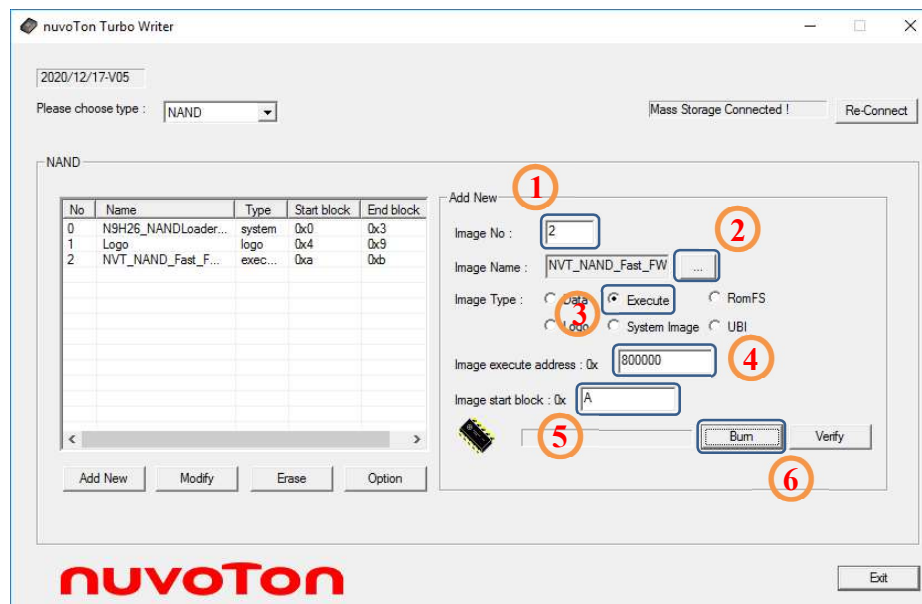


Figure 4-6 Execute image - NvtLoader

- Option function

Option function includes the information of total size, user can set “**System Reserved Area Size**”, and presses the button “**Apply**” to take effect.

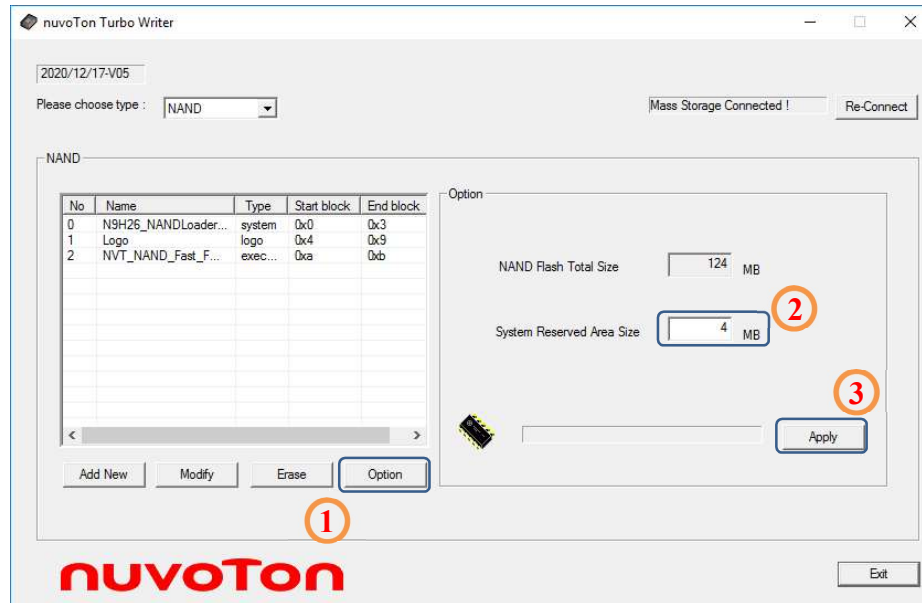


Figure 4-7 Reserved System Area Size

- Flash memory map after above step

- NAND is 2 KB page and 128 page per block

- ◆ Reserved 4 MB for System area (16 block)

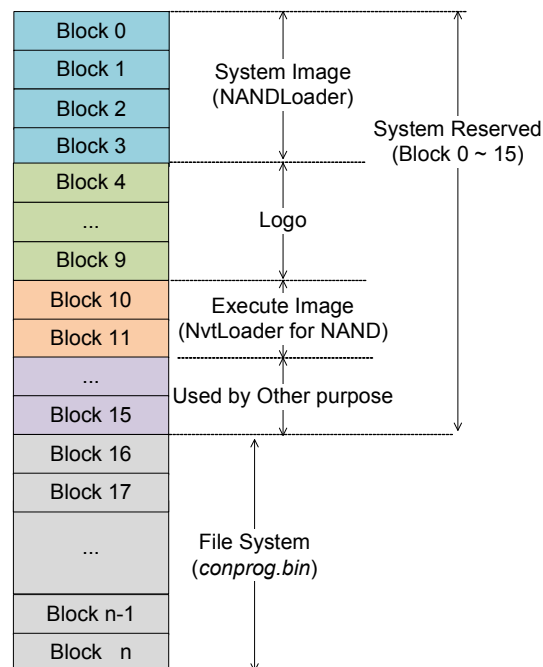


Figure 4-8 NAND Flash memory map

- Erase function

User can erase NAND Flash by Image No, all Flash, or System Area.

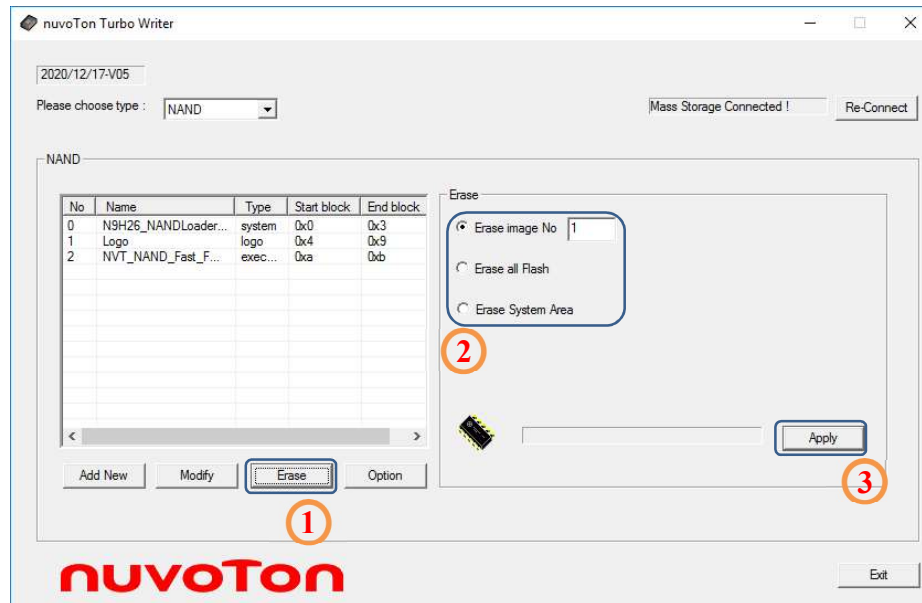


Figure 4-9 NAND Erase function

- Modify function

User can modify Image Type, and click down the button of “**Apply**” to take effect.

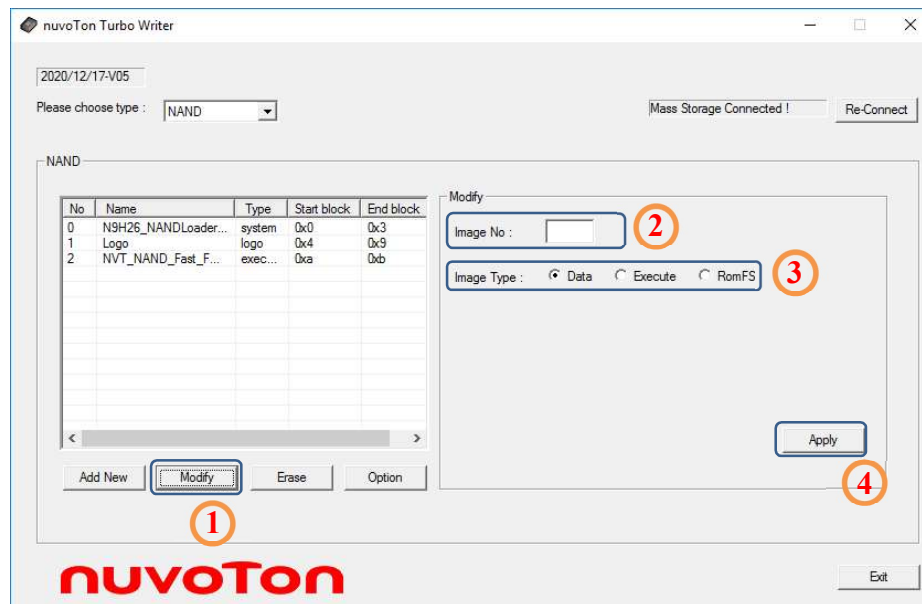


Figure 4-10 NAND Modify function

4.4 SD

The following lines describe how to use SD mode to burn images for Application Code.

- SD Loader
 1. Choose the type “SD”
 2. Press the button “Add New”
 3. Browse the file “N9H26_SDLoader_240MHz_Fast.bin”
 4. Set Image type “System Image”
 5. Press the button “Burn”

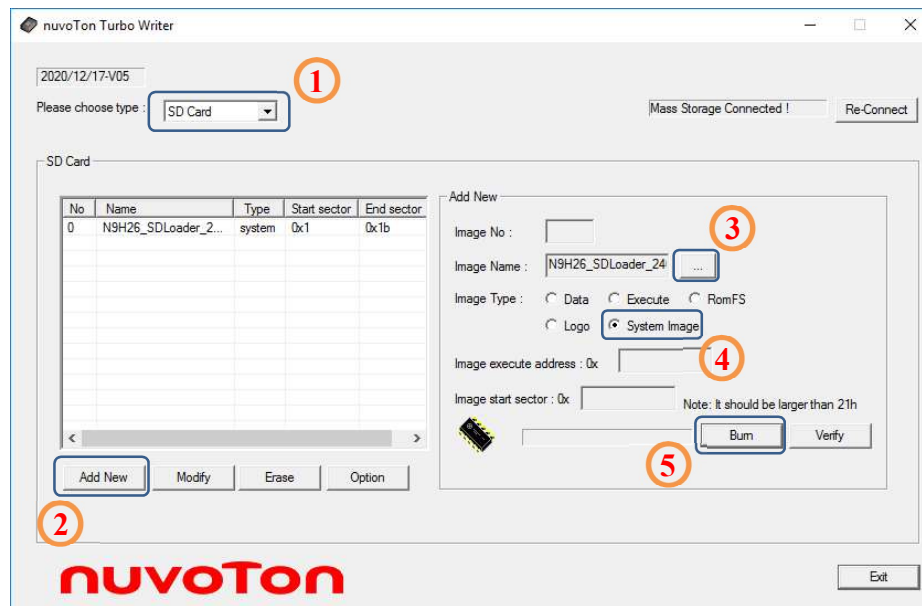


Figure 4-11 System image – SD Loader

- Logo
 1. Image number “1”
 2. Browse the file “Logo.bin”
 3. Set Image type “Logo”
 4. Set the image execute address: 0x500000
 5. Set the start block number: 0x22
 6. Press the button “Burn”

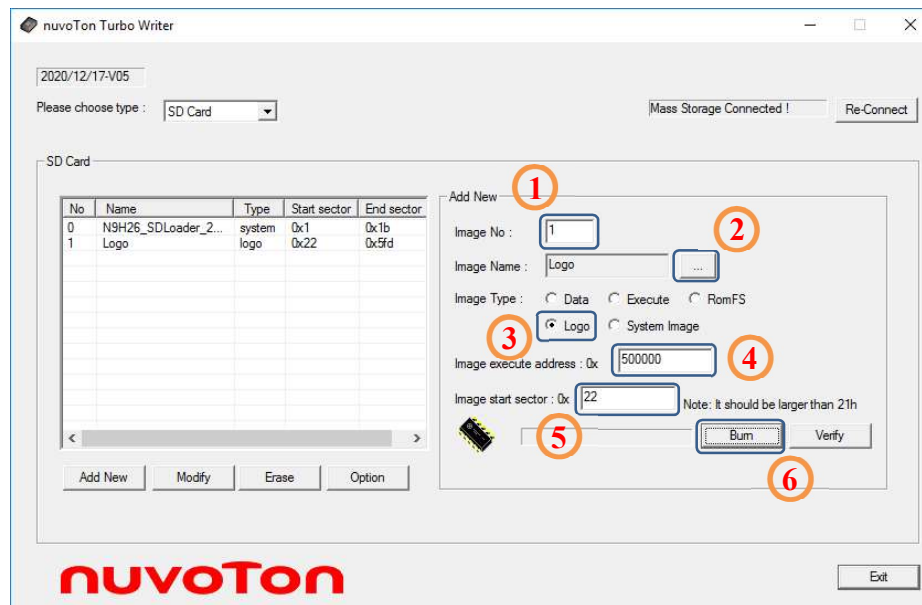


Figure 4-12 Logo image

- NVT Loader

1. Image number "2"
2. Browse the file "NVT_SDUI0_Fast_FW050TFT_800x480_24B.bin"
3. Set Image type "**Execute**"
4. Set the executed address: 0x800000
5. Set the start block number: 0x5FE.
6. Press the button "**Burn**"

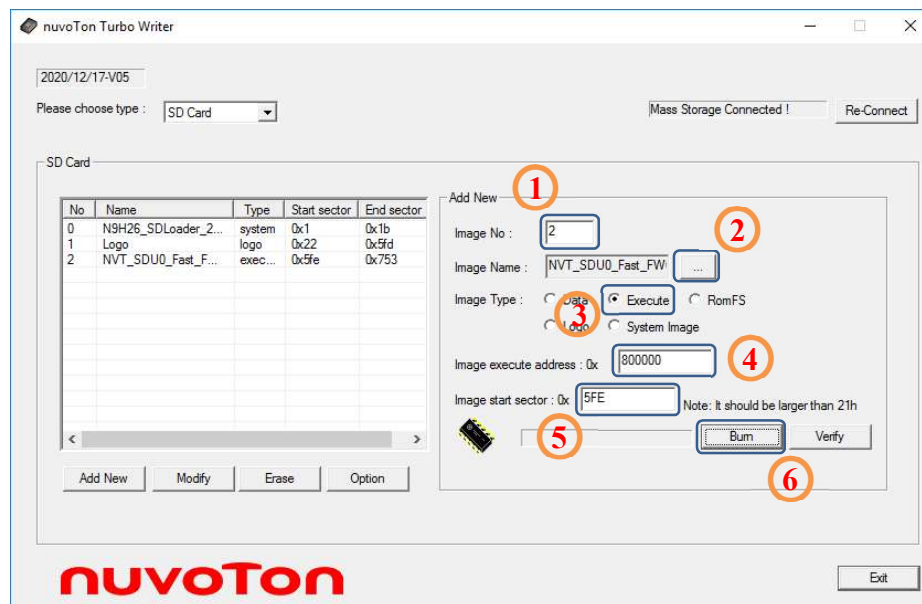


Figure 4-13 Execute image – NVT Loader

- Option function

Option function includes the information of total size, SD format, SD Port Selection function. User can set “**System Reserved Area Size**”, and presses the button “**Apply**” to take effect.

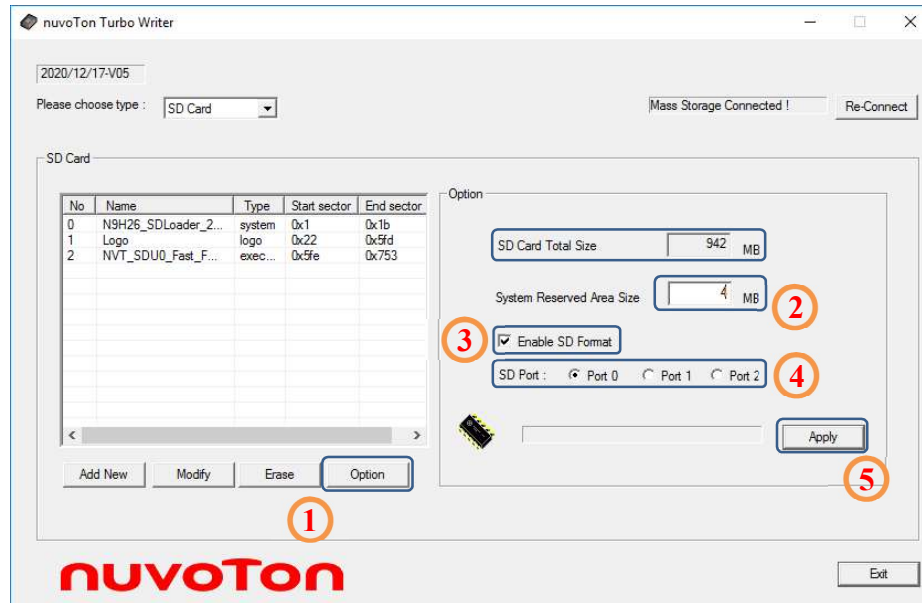


Figure 4-14 Reserved System Area Size

- Flash memory map after above step

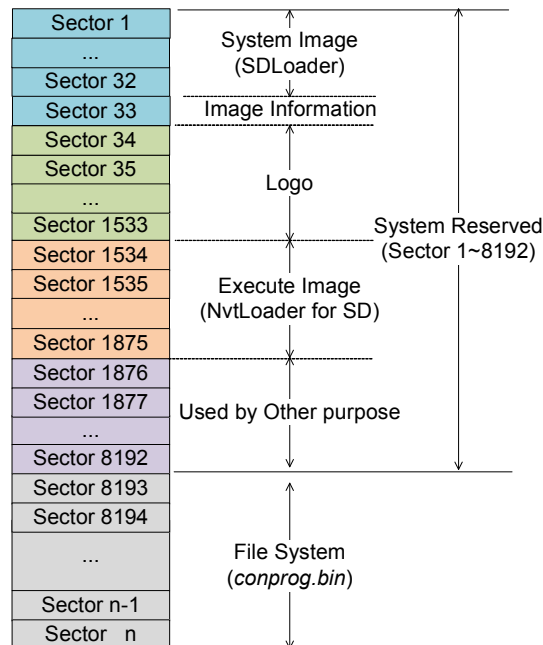


Figure 4-15 SD memory map

- Erase function

User can erase SD Card by Image No, or all Flash.

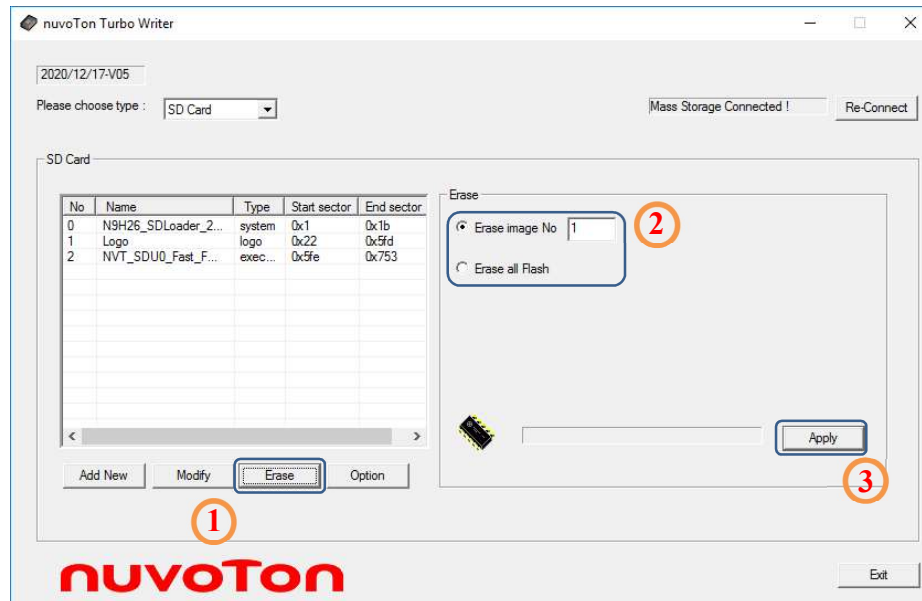


Figure 4-16 SD Erase function

- Modify function

User can modify Image Type, and click down the button of “**Apply**” to take effect.

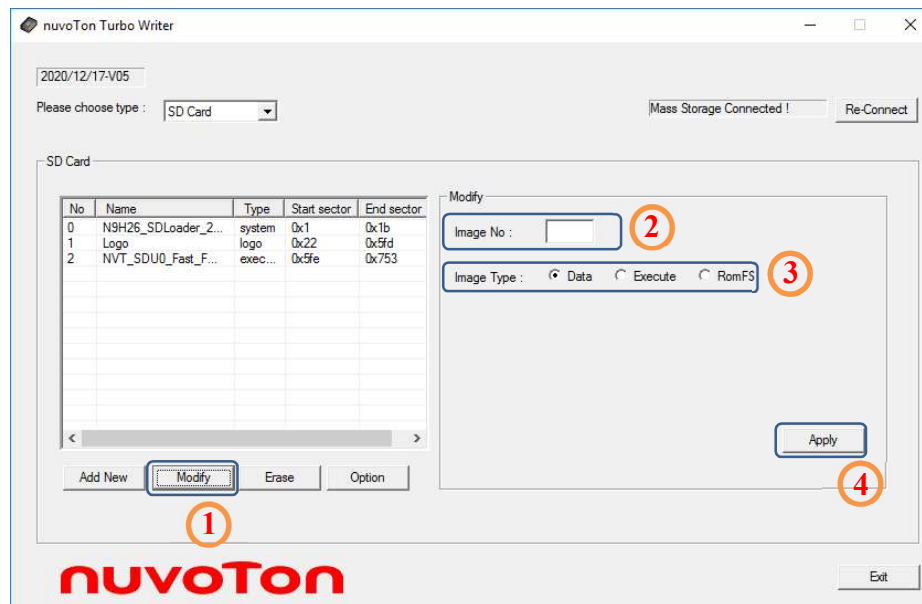


Figure 4-17 SD Modify function

4.5 SPI

The following lines describe how to use SPI mode to burn images for Application Code.

- SPI Loader
 1. Choose the type “**SPI**”
 2. Press the button “**Add New**”
 3. Browse the file “*N9H26_SpiLoader_240MHz_FW050TFT_800x480_24B.bin*”
 4. Set Image type “**System Image**”
 5. Press the button “**Burn**”

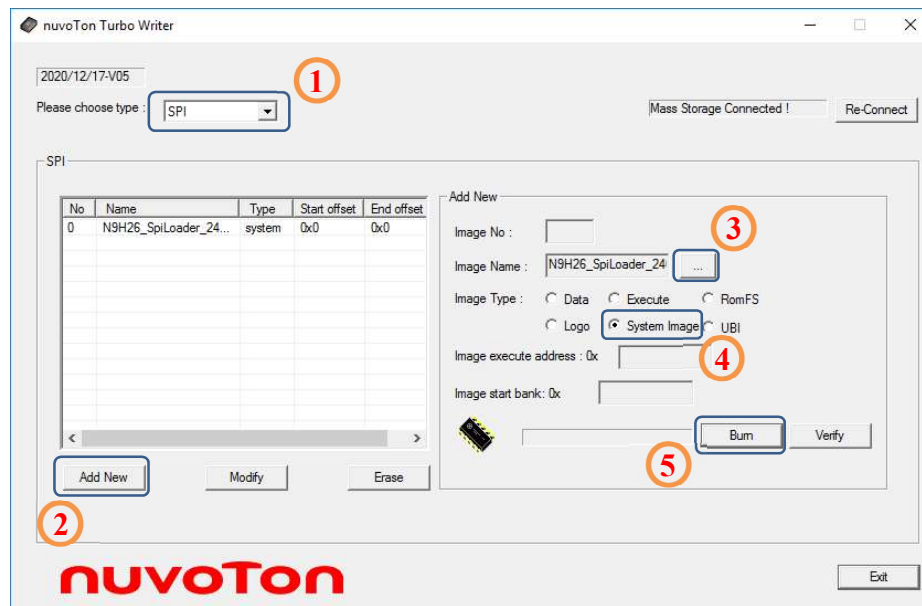


Figure 4-18 System image – SPI Loader

- Logo
 1. Image number “1”
 2. Browse the file “*Logo.bin*”
 3. Set Image type “**Logo**”
 4. Set the image execute address: 0x500000
 5. Set the start block number: 0x1
 6. Press the button “**Burn**”

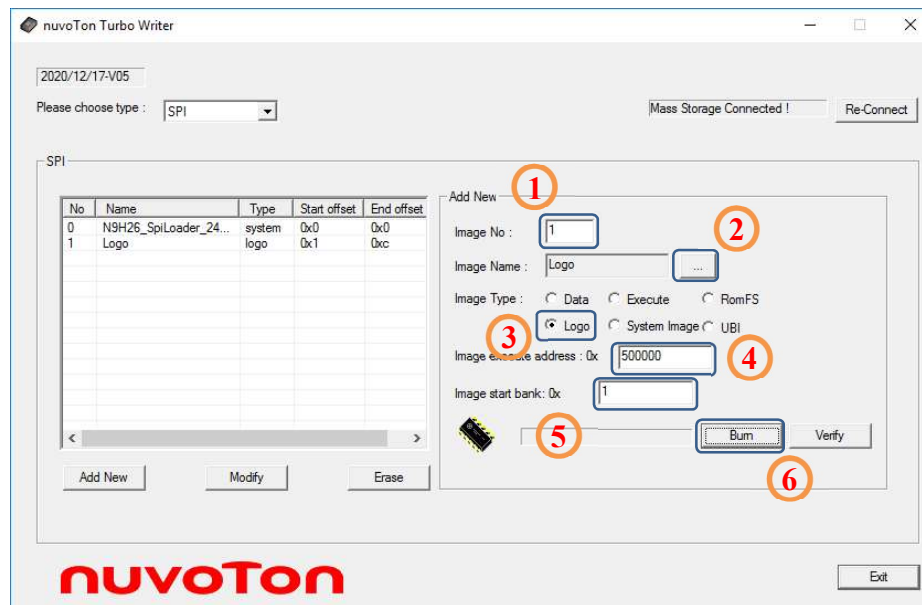


Figure 4-19 Logo image

- Application Code

1. Image number "2"
2. Set Image type "Execute"
3. Browse the file "conprog.bin"
4. Set the executed address: 0x0
5. Set the start block number: 0xD.
6. Press the button "Burn"

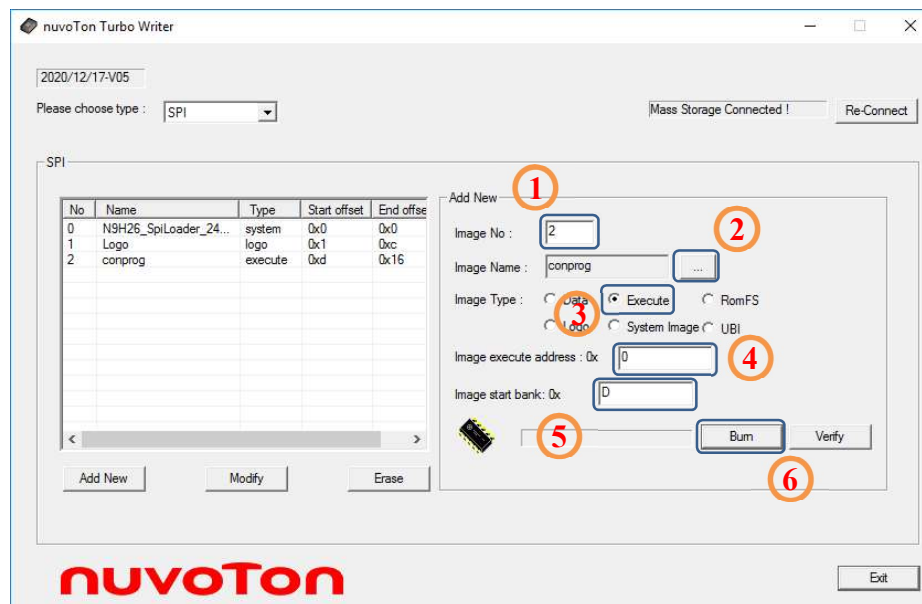


Figure 4-20 Execute image – Application Code

- Flash memory map after above step

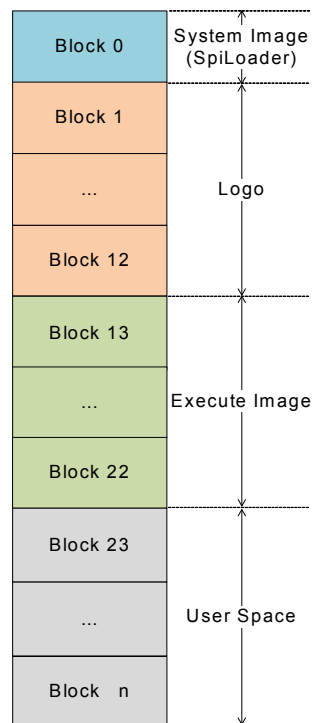


Figure 4-21 SPI memory map

- Erase function
It includes SPI Flash Size information and Erase function. User can erase SPI Flash by Image No or all Flash.

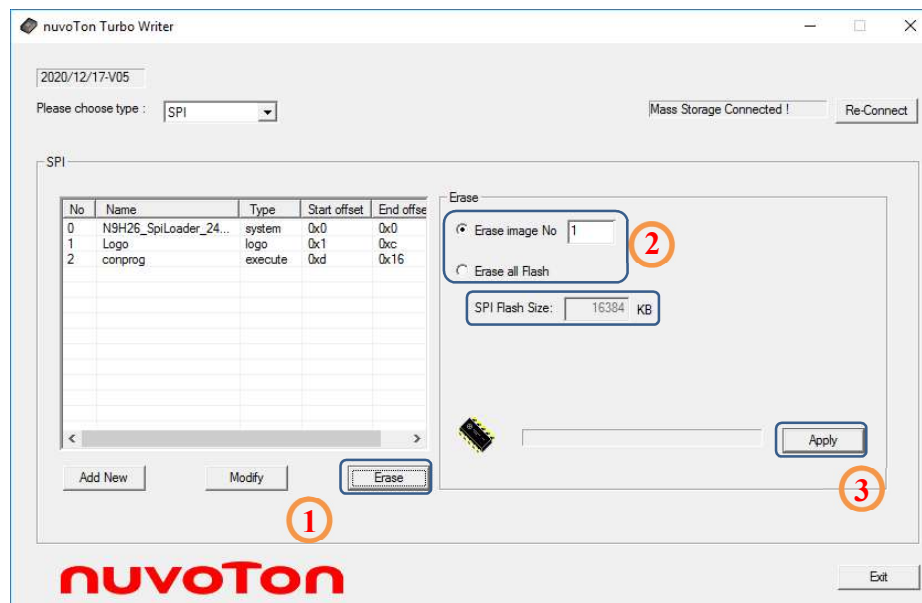


Figure 4-22 SPI Erase function

- Modify function

User can modify Image Type, and click down the button of “**Apply**” to take effect.

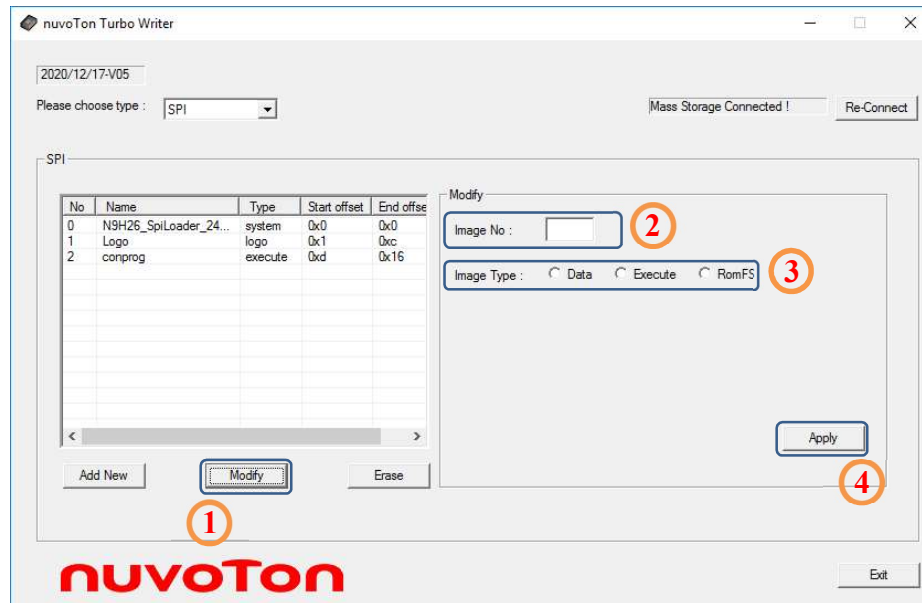


Figure 4-23 SPI Modify function

4.6 SPI (Raw Data)

SPI (Raw Data) displays the SPI flash Size information and supports Image burn, data read back from SPI Flash, Make Rom function. The **SPI (Raw Data)** interface is as follows.

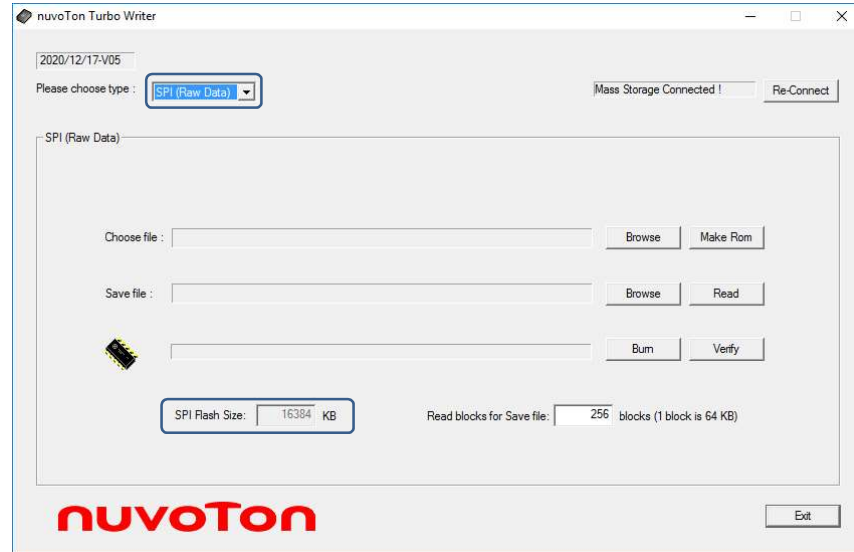


Figure 4-24 SPI(Raw Data) Mode

The followings are the sample to burn “conprog.bin”, read back (save) to “ReadBack.bin”, and “Make Rom” function.

- Burn function
 - User can burn the specified file into SPI Flash.
 1. Browse the file “Image.bin” (Choose file)
 2. Press the button “Burn”

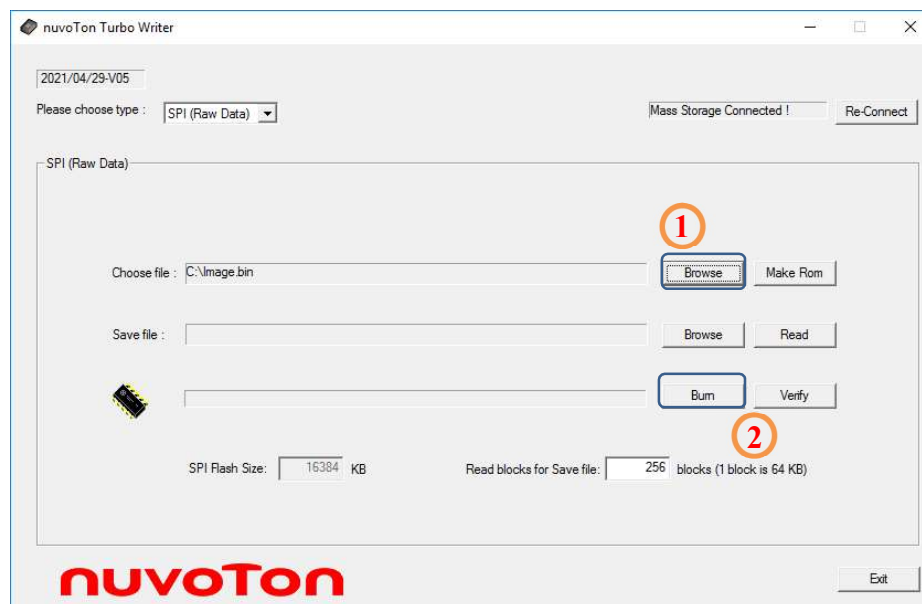


Figure 4-25 SPI (Raw Data) Burn function

- Verify function
 - User can verify the binary file choosed by “Burn” from SPI Flash

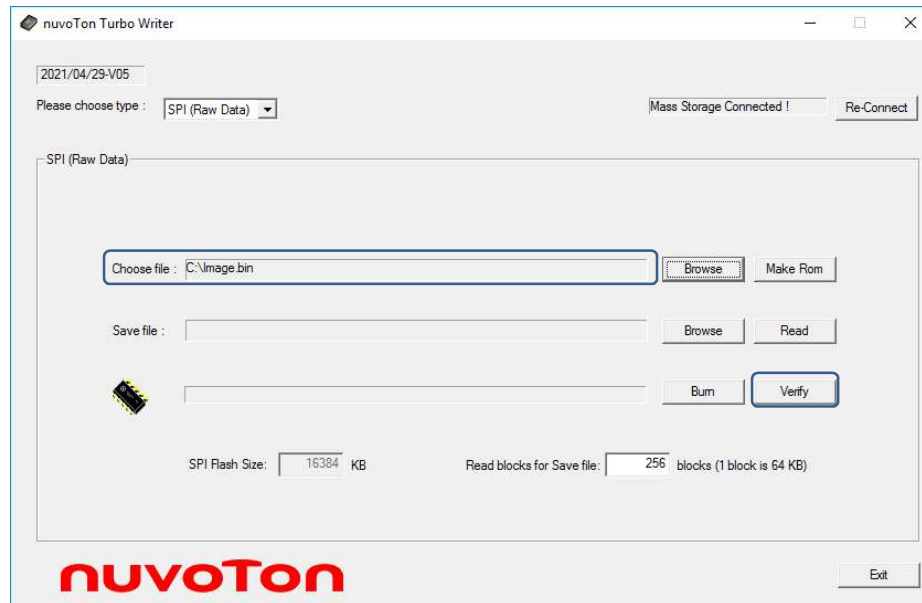


Figure 4-26 SPI (Raw Data) Verify function

- Read function
 1. Browse the file “ReadBack.bin” (Save file)
 2. Set the “Read blocks for Save file” (Minimum Read back unit: 64KB)
 - ◆ If SPI Flash Size is 0, please refer Section 2.3.
 3. Press the button “Read”

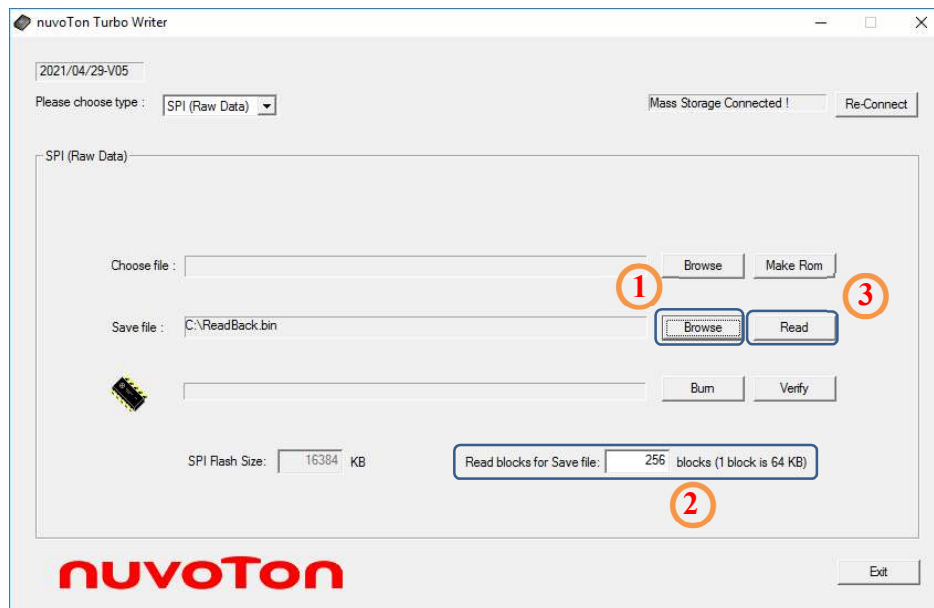


Figure 4-27 SPI (Raw Data) Read function

- Make Rom function
 - Turbo Writer supports SPI “**Make ROM**” utility for user to build a packed image for mass production. It provides “**Add**”, “**Modify**”, “**Remove**”, and “**Make**” function.

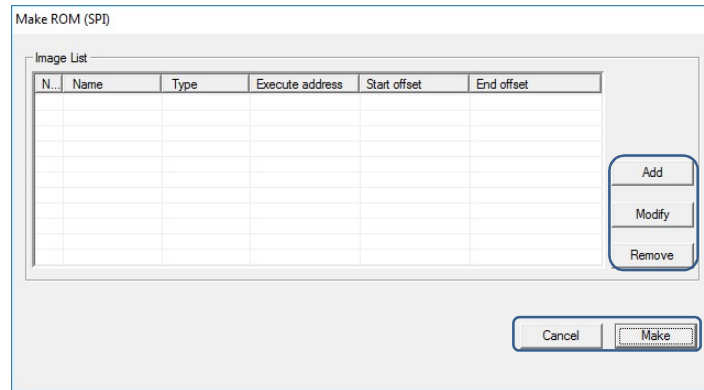


Figure 4-28 SPI (Raw Data) Make Rom function

- ◆ User can use “**Add**” function to add new image. The user interface is the same as SPI modes.

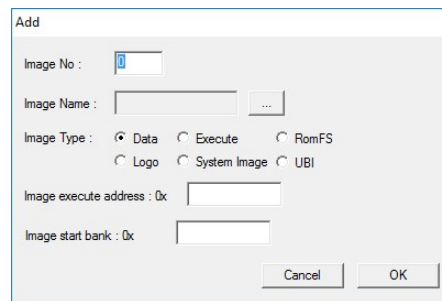


Figure 4-29 SPI (Raw Data) Make Rom – “**Add**” function

For example, *N9H26_SpiLoader_240MHz_FW050TFT_800x480_24B.bin*, *Logo.bin* and Application Code – *conprog.bin*.

- SPI Loader
 1. Press the button “**Add**”
 2. Image number “0”
 3. Set Image type “**System Image**”
 4. Browse the file
“*N9H26_SpiLoader_240MHz_FW050TFT_800x480_24B.bin*”
 5. Press the button “**OK**”
- Logo
 1. Press the button “**Add**”
 2. Image number “1”
 3. Set Image type “**Logo**”

4. Browse the file “*Logo.bin*”
5. Set the image execute address: 0x500000
6. Set the start block number: 0x1
7. Press the button “**OK**”
- Application Code
 1. Press the button “**Add**”
 2. Image number “2”
 3. Set Image type “**Execute**”
 4. Browse the file “*conprog.bin*”
 5. Set the executed address: 0x0
 6. Set the start block number: 0xD.
 7. Press the button “**OK**”

Press “**Make**” and output a packed image. This image can used for mass production.

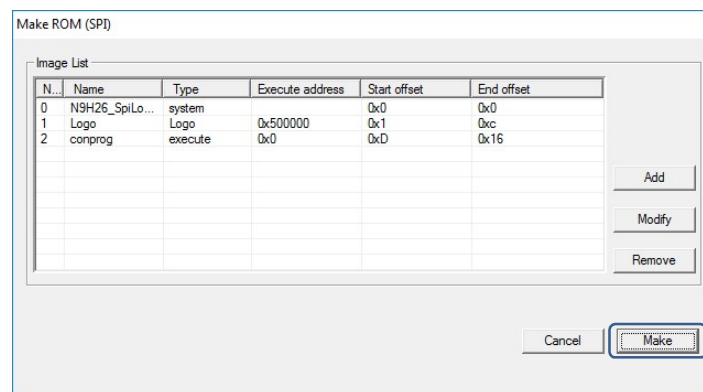


Figure 4-30 **SPI (Raw Data) Make Rom Result**

5 Supporting Resources

The N9H26 system related issues can be posted in Nuvoton's forum:

- ARM7/9 forum at: <http://forum.nuvoton.com/viewforum.php?f=12>.

Revision History

Date	Revision	Description
2021.06.10	2.01.009	Revise document
2018.08.09	2.01.008	<p>Add the chip N9H2X series including N9H20K1/N9H20K3/N9H20K5 and N9H26K5</p> <p>[USER_DEFINE] is obsolete.</p> <p>Rename FA93 into N3290X, FA95 into N3291X, FA92 into N3292X.</p> <p>Delete the statements of FA91.</p>
2013.03.26	2.01.007	<p>Add description for FA92</p> <p>Add FA93 Firmware number description about DRAM size (N32901U1DN / N32903U1DN / N32905U1DN)</p>
2012.05.29	2.01.006	<p>Add description for SPI (Raw Data)</p> <ul style="list-style-type: none"> ● Make ROM ● Read back Data from SPI flash
2012.02.23	2.01.005	<p>Add description for FA95</p> <p>Modify description for new version UI</p> <p>Add description for new function</p> <ul style="list-style-type: none"> ● SD format ● SD Port ● SPI Flash Size
2010.11.04	2.01.004	Change the file name of firmware for FA93. Its name is <i>FA93_musb.bin</i>
2010.10.25	2.01.003	16 bytes file name for FA91, and 32 bytes for FA93.
2010.10.19	2.01.002	Reduce the checking time for the file TurboWriter.ini,
2010.09.20	2.01.001	Support 32 bytes file name for FA93 later
2010.09.20	2.00.001	<p>Modify for W55FA series,</p> <p>Support header of boot loader image.</p>
2010.05	1.20	Add NVT Loader for NAND
2009.05	1.01	Add SPI and SPI (raw data) mode
2009.04	1.00	Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

*Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*