

Auto Writer Tool User Guide

Document Information

| | |
|-----------------|---|
| Abstract | Introduce how to use Auto 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 | AUTO WRITER INTRODUCTION | 4 |
| 2.1 | Tools\PC_Tools\AutoWriter V3.01.002_N9H26K6 | 5 |
| 2.2 | Turbo Writer extra support NAND ID | 6 |
| 2.3 | Turbo Writer extra support SPI Flash ID | 6 |
| 2.4 | Turbo Writer Configuration for Boot Code Header | 6 |
| 2.5 | Auto Writer Configuration | 6 |
| 2.6 | Auto Writer Firmware | 6 |
| 2.7 | Auto Writer Tool Version | 7 |
| 3 | AUTOWRITER.INI | 8 |
| 3.1.1 | Loader File Name | 9 |
| 3.1.2 | Logo File Name | 10 |
| 3.1.3 | Execute File Name | 10 |
| 3.1.4 | Execute Image File Backup | 11 |
| 3.1.5 | System Reserved MegaB | 11 |
| 3.1.6 | Options | 12 |
| 3.1.7 | DRAM Download only | 13 |
| 3.1.8 | Executing File Name | 13 |
| 3.1.9 | ExecuteImage | 13 |
| 3.1.10 | DATA | 14 |
| 3.1.11 | ROMFS | 15 |
| 3.1.12 | UBI | 16 |
| 3.1.13 | Post Process | 17 |
| 4 | AUTO WRITER TOOL | 19 |
| 5 | POST PROCESSING | 25 |
| 6 | MULTI-PORT FUNCTION | 27 |
| 7 | 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 Auto Writer Introduction

The purpose of Auto Writer is to simplify the steps for image burning for mass production or development. For example, user needs to follow the step below to burn execute image manually by Turbo Writer.

- NVT Loader
 1. Image number “2”
 2. Browse the file “NVT_SDU0_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**”

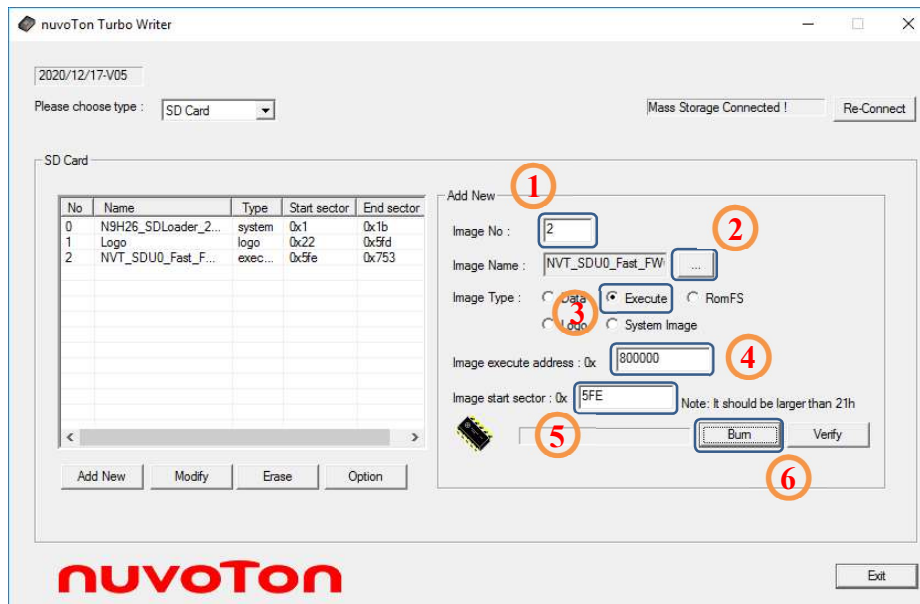


Figure 2-1 Turbo Writer Flow to burn execute image - NvtLoader

Using Auto Writer, user only needs to set the image properties in *AutoWriter.ini* one time. Auto Writer will connect to N329x / N9H2x when it connects to host and burn images according to the *AutoWriter.ini*. automatically.

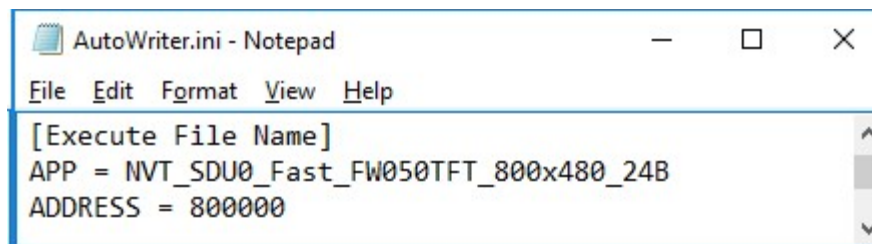


Figure 2-2 *AutoWriter.ini* setting for Execute image

If storage of the solution supports file system, user needs to create the disk image for the mass production or copy files to the disk which exported by NVTLoader through explorer.

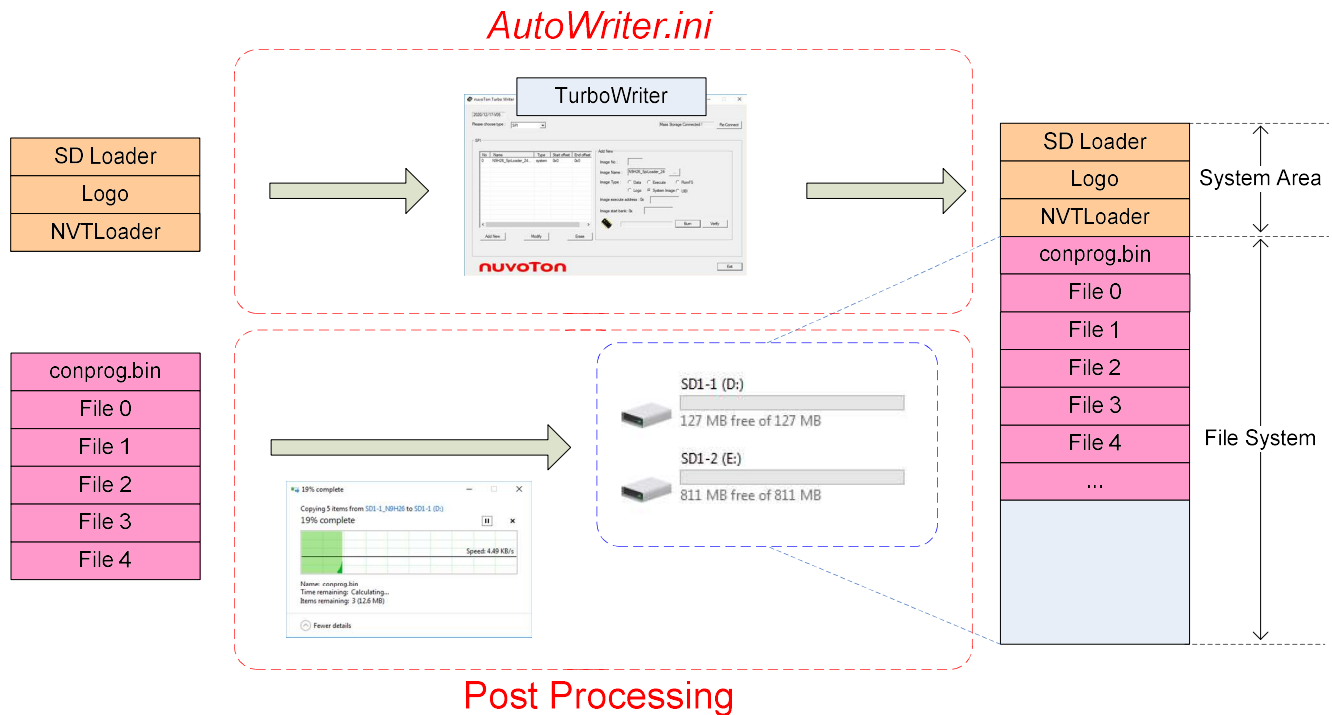


Figure 2-3 Flow to burn images and files

To simplify the steps, Auto Writer also supports Post Processing function for the file copy. AutoWriter.ini and Post Processing function will be described in Chapter 3 AutoWriter.ini and Chapter 5 Post Processing. The following lines describe the files within the Auto Writer folder.

2.1 Tools\PC_Tools\AutoWriter V3.01.002_N9H26K6

| | |
|------------------------|---|
| AutoWriter.exe | Auto Writer PC Tool execution file |
| NAND ID.ini | Turbo Writer extra support NAND ID |
| SPIFLASH ID.ini | Turbo Writer extra support SPI Flash ID |
| TurboWriter.ini | Turbo Writer Configuration for Boot Code Header |
| AutoWriter.ini | Auto Writer Configuration |
| xxxx_musb.bin | Auto Writer Firmware |

The files - *NAND ID.ini*, *SPIFLASH ID.ini*, and *TurboWriter.ini* must be the same from Turbo Writer folder user used for development.

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*. Further information about *TurboWriter* can be found at *BSP Tools/PC_Tools/TurboWriter Tool User Guide.pdf*.

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*. Further information about *TurboWriter* can be found at *BSP Tools/PC_Tools/TurboWriter Tool User Guide.pdf*.

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. Further information about *TurboWriter* can be found at *BSP Tools/PC_Tools/TurboWriter Tool User Guide.pdf*.

2.5 Auto Writer Configuration

AutoiWriter.ini file provides user a flexible way to change program setting. The file describes the file properties user wants to write to Flash in the Auto Writer folder. The detail will be described in following sections.

2.6 Auto Writer Firmware

Auto Writer is connected to Turbo writer firmware which loaded from PC to N329 / N9H2x series by USB boot. When Auto Writer connects to USB boot, Turbo Writer will send the corresponding binary file (Auto Writer firmware) to N329 / N9H2x series and connects to Auto 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> |

- Auto 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 ; .????????
```

- Auto Writer firmware marker : 0x2054564E
- Auto Writer firmware executing address : 0x001A00000
- Auto Writer firmware size : 0x02D4A8
- Auto Writer firmware version : 0x21042905 → 2021/10/28 V5
- Auto Writer firmware code starts from offset 0x20.

2.7 Auto Writer Tool Version

Auto Writer PC tool version can be seen from Auto Writer Tool. Please Click Auto Writer icon and select “About AutoWriter”. The Version will show the Auto Writer Tool Version.

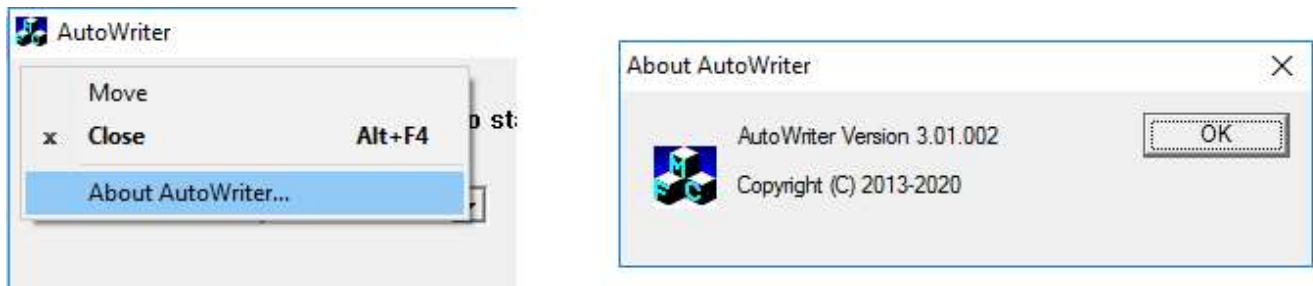


Figure 2-4 Auto Writer Tool version

3 AutoWriter.ini

The INI file means *AutoWriter.ini* file that provides user a flexible way to change program setting. The file describes the file properties user wants to write to NAND Flash / SPI Flash / SD Card in the Auto Writer folder.

- The files described in *AutoWriter.ini* must be in the Auto Writer folder, otherwise the processing of Auto Writer will stop.
- Block Unit in *AutoWriter.ini*
 - It is the sector unit (512 bytes) for SD.
 - It is the block unit (64 Kbytes) for SPI Flash.
- Block Number is hexadecimal
- If user comments or deletes the specified section, the specified function will not work.
- If user does not set "Start Block", AutoWriter will work the suitable start address of storage SPI/SD/NAND/SPIRAW automatically.

The *AutoWriter.ini* file provides some sections as below.

```
[Loader File Name]
APP = N9H26_SDLoader_240MHz_Fast.bin
ADDRESS = 900000

[Logo File Name]
APP = NuvotonLogo_800x480.bin
ADDRESS = 500000

[Execute File Name]
APP = NVT_SDU0_Fast_FW050TFT_800x480_24B.bin
ADDRESS = 800000

//[Execute Image File Backup]
//Backup count for Executing Image
//NO = 3

//[Executing File Name]
//APP = LinuxWriter.bin
//ADDRESS = 0

[System Reserved MegaB]
//Unit : Mega Byte
NO = 2
```



```
//[DRAM Download only]
//DATA = RomFS.bin
//ADDRESS = 0

[Options]
EraseAll=y
Verify=n

[Post Process]
DiskImage= y
// The following is MB unit, decimal unit.
SizeofNAND1-1 = 30
//SizeofNAND1-2 = 90
```

3.1.1 Loader File Name

The section [Loader File Name] is for Loader file and it is the System type of TurboWriter. The format for this setting is:

[Loader File Name]

APP = File name

ADDRESS = address

For example,

```
[Loader File Name]
APP = N9H26_SDLoader_240MHz_Fast.bin
ADDRESS = 900000
```

- The description of APP is the file name *N9H26_SDLoader_240MHz_Fast.bin* to be burned into Flash.
- The description of ADDRESS is the loader execute address. The following table is the default Loaders execute address for each chip.

| 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. |

About the SPIRAW mode, user could set only the binary file (for example, the binary file created by NRomMaker) to the APP of [Loader File Name], and comment the all the other section. Further information about NRomMaker can be found at *BSP Tools/MassProduction_Tools/NRomMaker / NRomMaker Tool User Guide.pdf*.

3.1.2 Logo File Name

The section [Logo File Name] is for Logo file and it is the Logo type of TurboWriter. The format for this setting is:

[Logo File Name]

APP = File name

ADDRESS = address

Start Block = Block Number

For example,

```
[Logo File Name]
APP = NuvotonLogo_800x480.bin
ADDRESS = 500000
```

- The description of APP is the file name *NuvotonLogo_800x480.bin* to be burned into Flash.
- The description of ADDRESS is the logo execute address

If user would like to set the start block of storage SPI/SD/NAND, set the section "Start Block = XXXX". For example, Logo execute address is 0x500000 and start block is 34.

```
[Logo File Name]
APP = NuvotonLogo_800x480.bin
ADDRESS = 500000
Start Block = 22
```

3.1.3 Execute File Name

The section [Execute File Name] is for Execute file and it is the Execute type of TurboWriter. The format for this setting is:

[Execute File Name]

APP = File name

ADDRESS = address

Start Block = Block number

For example,

```
[Execute File Name]
APP = NVT_SDU0_Fast_FW050TFT_800x480_24B.bin
ADDRESS = 800000
```

- The description of APP is the file name *NVT_SDU0_Fast_FW050TFT_800x480_24B.bin* to be burned into Flash.
- The description of ADDRESS is the execute address.

If user would like to set the start block of storage SPI/SD/NAND, set the section “Start Block = XXXX”. For example, Burn execute image – *NVT_SDU0_Fast_FW050TFT_800x480_24B.bin* from block (sector) 1534 and execute image is 0x00800000

```
[Execute File Name]
APP = NVT_SDU0_Fast_FW050TFT_800x480_24B.bin
ADDRESS = 800000
Start Block = 5FE
```

3.1.4 Execute Image File Backup

The section [Execute Image File Backup] is Backup count for Executing Image. The format for this setting is:

[Execute Image File Backup]

NO = Backup count for Executing Image

For example,

```
[Execute Image File Backup]
NO = 3
```

The section [Execute Image File backup] is the backup of execution file for 3 times, It will go on burning the execute file name for 3 times. If user comments or deletes the section, the function will not work

3.1.5 System Reserved MegaB

The section [System Reserved MegaB] is the system area size (Unit is Mega Byte). The format for this setting is:

[System Reserved MegaB]

NO = Backup count for Executing Image

For example,

```
[Execute Image File Backup]
NO = 4
```

The section [System Reserved MegaB] sets the system area of storage to be 4MB.

3.1.6 Options

The section [Options] can set NAND / SPI to erase operation and verify function. The format for this setting is:

[Options]

EraseAll = y / n / p

Verify = y / n

- Erase All function

If user comments or deletes the section, NAND / SPI will not be erased all. EraseAll = y will erase all NAND / SPI, EraseAll = n will not erase all, just erase corresponding blocks. For example, erase all NAND / SPI Flash

```
[Options]
EraseAll = y
Verify = y
```

- Partial Erase function

It could support partial erase from block n to block m Only for SPI Flash in the version 3.00.009 later as follows. For example, erase SPI Flash from block 0 to block 0x2F.

```
[Options]
EraseAll=p
StartBlock = 0
EndBlock= 2F
Verify = y
```

- Verify function

If the option Verify set to be 'Y' or 'y', then all downloaded files with verification function, it will take more time to finish. From V4.00.001, we suggest run Verify as a mandatory.

3.1.7 DRAM Download only

The section [DRAM Download only] is for the code to download to DRAM. The format for this setting is:

[DRAM Download only]

DATA = File name

ADDRESS = address

For example,

```
[DRAM Download only]
```

```
DATA = RomFS.bin
```

```
ADDRESS = 0
```

- The description of DATA is the file name *RomFS.bin* to download to DRAM.
- The description of ADDRESS is the DRAM address.

3.1.8 Executing File Name

The section [Executing File Name] is for execute coding on DRAM after all image is burned. The format for this setting is:

[Executing File Name]

APP = File name

ADDRESS = address

For example,

```
[Executing File Name]
```

```
APP = ResetByWDT.bin
```

```
ADDRESS = 800000
```

- The description of APP is the file name *ResetByWDT.bin* to be downloaded to DRAM.
- The description of ADDRESS is the execute address.

3.1.9 Executelmage

The section [Executelmage] is used to burn more than one execute files and it is the Execute type of TurboWriter. The format for this setting is:

[Executelmage]

No = n

APP1 = File name 1

APP2 = File name 2

...

APP n = File name n

Start Block1 = block number 1

Start Block2 = block number 2

...

Start Block n = block number n

The value “ n ” of “No” is the image number. If NO=2, there must exist APP1, APP2 for 2 images, otherwise the processing will be incorrect.

For example,

```
[ExecuteImage]
No=2
APP1 = image1.bin
APP2 = image2.bin
```

- The description of APP are the file names *image1.bin* & *image2.bin* to be burned into Flash.

If user would like to set the start block of storage SPI/SD/NAND, set the section “Start Block = XXXX”. For example, Burn execute image – *image1.bin* from block 16, and *image2.bin* from block 19.

```
[ExecuteImage]
No=2
APP1 = image1.bin
APP2 = image2.bin
Start Block1 = 10
Start Block2 = 13
```

3.1.10 DATA

The section [DATA] is used to burn data image files and it is the Data type of TurboWriter. The format for this setting is:

[DATA]

No = n

APP1 = File name 1

APP2 = File name 2

...

APP n = File name n

Start Block1 = block number 1

Start Block2 = block number 2

...

Start Block n = block number n

Execution Address1 = address1

Execution Address2 = address2

...

Execution Address n = address n

The value “ n ” of “No” is the image number. If NO = 2, there must exist APP1, APP2 for 2 images, otherwise the processing will be incorrect.

For example,

```
[DATA]
No=2
APP1 = image1.bin
APP2 = image2.bin
Execution Address1= 00200000
Execution Address2= 002A0000
```

- The description of APP are the file names *image1.bin* & *image2.bin* to be burned into Flash.
- The description of Execution Address are the execute address for *image1.bin* & *image2.bin*.

If user would like to set the start block of storage SPI/SD/NAND, set the section “Start Block = XXXX”. For example, Burn Data image – *image1.bin* from block 16, and *image2.bin* from block 19.

```
[DATA]
No=2
APP1 = image1.bin
APP2 = image2.bin
Start Block1 = 10
Start Block2 = 13
Execution Address1= 00200000
Execution Address2= 002A0000
```

3.1.11 ROMFS

The section [ROMFS] is used to burn ROMFS files and it is the RomFS type of TurboWriter. The format for this setting is:

[ROMFS]

No = n

APP1 = File name 1

APP2 = File name 2

...

APP n = File name n

Start Block1 = block number 1

Start Block2 = block number 2

...

Start Block n = block number n

Execution Address1 = address1

Execution Address2 = address2

...

Execution Address n = address n

The value “ n ” of “No” is the image number. If NO=2, there must exist APP1, APP2 for 2 images, otherwise the processing will be incorrect.

For example,

```
[ROMFS]
No=2
APP1 = image1.bin
APP2 = image2.bin
```

- The description of APP are the file names *image1.bin* & *image2.bin* to be burned into Flash.

If user would like to set the start block of storage SPI/SD/NAND, set the section “Start Block = XXXX”. For example, Burn RomFS image – *image1.bin* from block 16, and *image2.bin* from block 19.

```
[ROMFS]
No=2
APP1 = image1.bin
APP2 = image2.bin
Start Block1 = 10
Start Block2 = 13
```

3.1.12 UBI

The section [UBI] is used to burn UBI files and it is the UBI type of TurboWriter. The format for this setting is:

[UBI]

No = n

APP1 = File name 1

APP2 = File name 2

...

APPn = File name n

Start Block1 = block number 1

Start Block2 = block number 2

...

Start Blockn = block number n

The value “n” of “No” is the image number. If NO=2, there must exist APP1, APP2 for 2 images, otherwise the processing will be incorrect.

For example,

```
[UBI]
No=2
APP1 = image1.bin
APP2 = image2.bin
```

- The description of APP are the file names *image1.bin* & *image2.bin* to be burned into Flash.

If user would like to set the start block of storage SPI/SD/NAND, set the section “Start Block = XXXX”. For example, Burn UBI image – *image1.bin* from block 16, and *image2.bin* from block 31.

```
[UBI]
No=2
APP1 = image1.bin
APP2 = image2.bin
Start Block1 = 10
Start Block2 = 1F
```

3.1.13 Post Process

The section [Post Process] is to set size of storage. The format for this setting is:

[Post Process]

DiskImage = y

SizeofNAND1-1 = 30

SizeofNAND1-2 = 90

The section [Post Process] sets the size of storage for SD / NAND. The storage must set two volume labels NAND1-1 and NAND1-2 by setting "DiskImage=y". SizeofNAND1-1 is for the size of NAND1-1, SizeofNAND1-2 is for the size of NAND1-2. SizeofNAND1-1 must be set. The set of SizeofNAND1-2 is optional, then the size of SizeofNAND1-2 is "the size of storage minus the size of SizeofNAND1-1". For example, set NAND1-1 to 30 MB.

```
[Post Process]
DiskImage= y
// The following is MB unit, decimal unit.
SizeofNAND1-1 = 30
```

4 Auto Writer Tool

After setting *AutoWriter.ini*, user can start to use Auto Writer Tool. Take SD for example, the following is the *AutoWriter.ini* for test.

```
[Loader File Name]
APP = SDLoader_240MHz_Fast.bin
ADDRESS = 900000

[Logo File Name]
APP = NuvotonLogo_800x480.bin
ADDRESS = 500000

[Execute File Name]
APP = NVT_SDU0_Fast_FW050TFT_800x480_24B.bin
ADDRESS = 800000

[System Reserved MegaB]
NO = 4

[Options]
Verify=n

[Post Process]
DiskImage= y
SizeofNAND1-1 = 128
```

The files in Auto Writer folder is as follows.

| Name | Date modified | Type | Size |
|--|--------------------|-----------------------|--------|
| AutoWriter User Guide.pdf | 2018/8/10 下午 02... | Adobe Acrobat D... | 442 KB |
| AutoWriter.exe | 2021/1/25 上午 10... | Application | 88 KB |
| AutoWriter.ini | 2021/6/18 上午 08... | Configuration sett... | 1 KB |
| N9H26_musb.bin | 2021/4/23 上午 08... | BIN File | 182 KB |
| N9H26_musb_FullSpeed.bin | 2021/4/23 上午 08... | BIN File | 182 KB |
| NAND ID.ini | 2017/12/27 上午 1... | Configuration sett... | 1 KB |
| NuvotonLogo_800x480.bin | 2021/6/8 下午 03:49 | BIN File | 750 KB |
| NVT_SDU0_Fast_FW050TFT_800x480_24B.... | 2021/5/24 上午 10... | BIN File | 171 KB |
| SDLoader_240MHz_Fast.bin | 2021/4/29 下午 06... | BIN File | 14 KB |
| SPIFLASH ID.ini | 2014/3/19 下午 01... | Configuration sett... | 1 KB |
| Target.ini | 2021/6/18 上午 09... | Configuration sett... | 1 KB |
| TurboWriter.ini | 2021/5/7 下午 03:17 | Configuration sett... | 2 KB |

Figure 4-1 The files in Auto Writer folder

Please following the step to connect to Auto Writer.

1. Set the N329x / N9H26 to Recovery mode.
2. Coeect to host through USB
3. Execute Auto Wrtier
4. Select Current Target “SD”
 - User can press “**Save**” to save the setting.

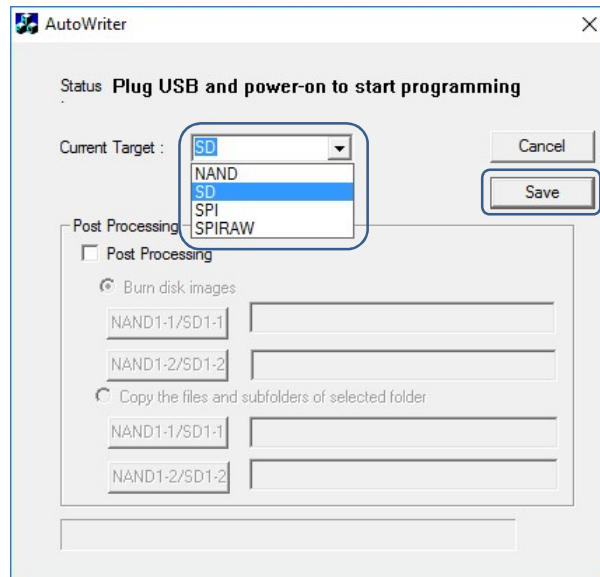


Figure 4-2 Select Target

5. Power on or reset N329x / N9H26
 - Auto Writer will start to detect device

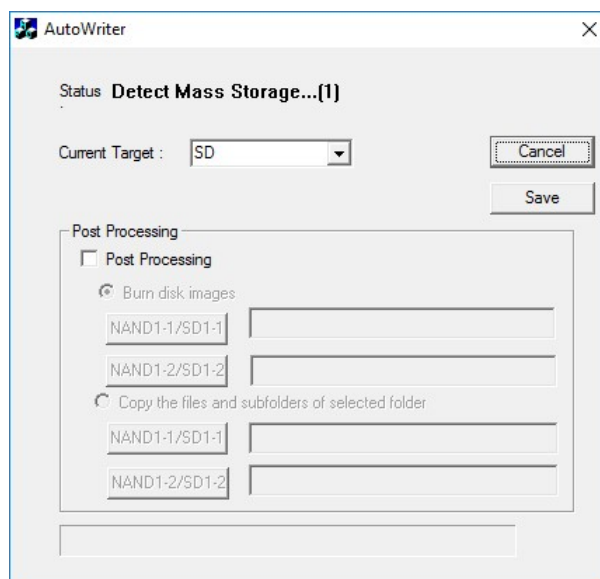


Figure 4-3 Detect device

6. Wait Auto Wrtier connected

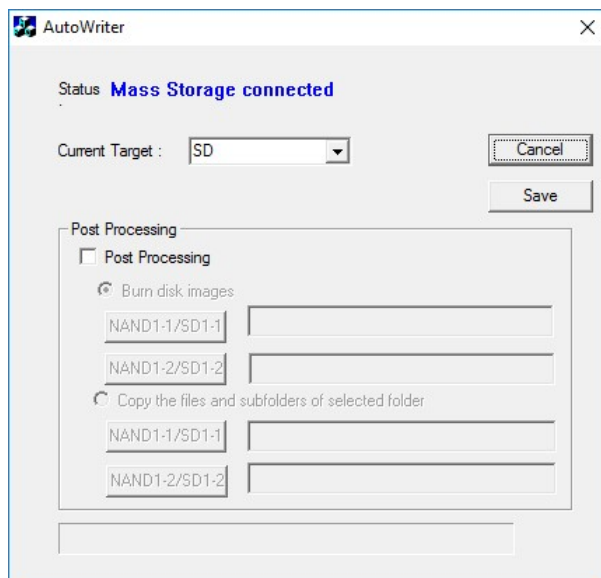


Figure 4-4 Auto Writer connects to device

7. Auto Wrtier starts to burn image according to *AutoWriter.ini*.

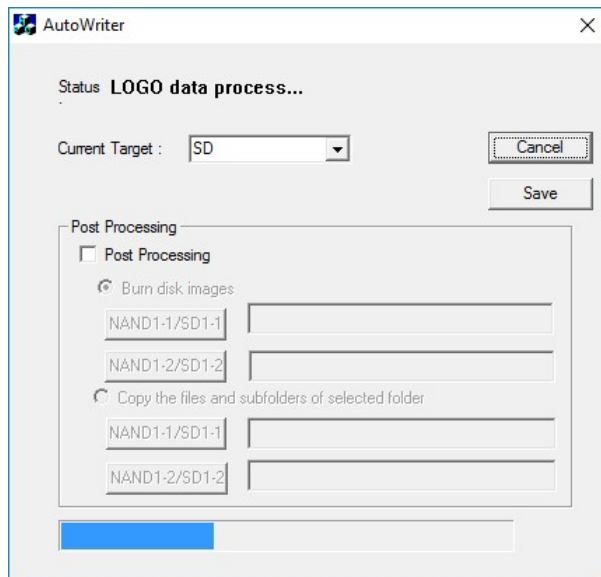


Figure 4-5 Burning images

8. Burn Success

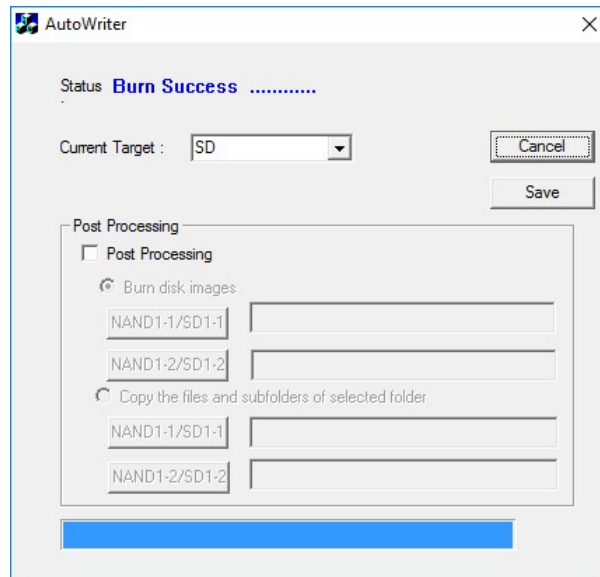


Figure 4-6 8. Burn Success

If the files described in *AutoWriter.ini* not in the Auto Writer folder, the processing of Auto Writer stops.

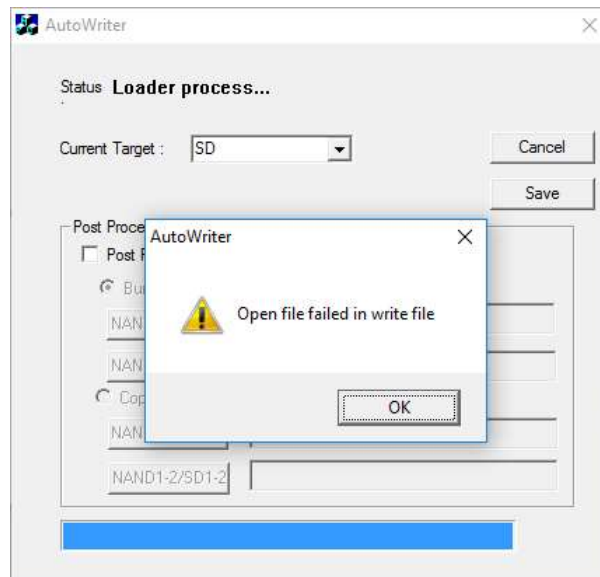


Figure 4-7 The processing of Auto Writer stops

\\ After Auto Writer process, the image list is the same as Turbo Writer process.

| N./ | Name | Type | Start se... | End sec |
|-----|-------------------------|---------|-------------|---------|
| 0 | SDLoader_240MHz_Fast... | system | 0x1 | 0x1c |
| 1 | NuvotonLogo_800x480.bin | logo | 0x22 | 0x5fd |
| 2 | NVT_SDU0_Fast_FW05... | execute | 0x5fe | 0x753 |

Figure 4-8 Image view in Turbo Writer

The memory map is as follows.

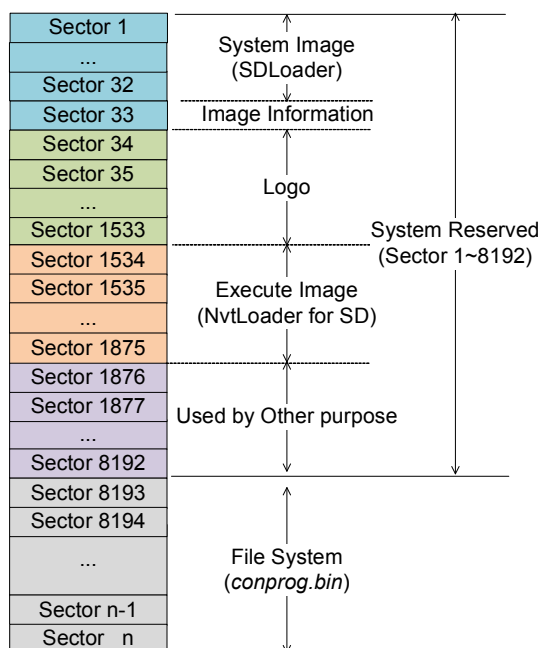


Figure 4-9 SD memory map

User needs to set the N329x / N9H2x to normal mode, and then runs the normal mode to enter USB mass storage (NVTLoader), to copy files into the disks SD1-1 and SD1-2 of the storage. The function of Posting Processing will work the above processing and it will be introduced in next Chapter.

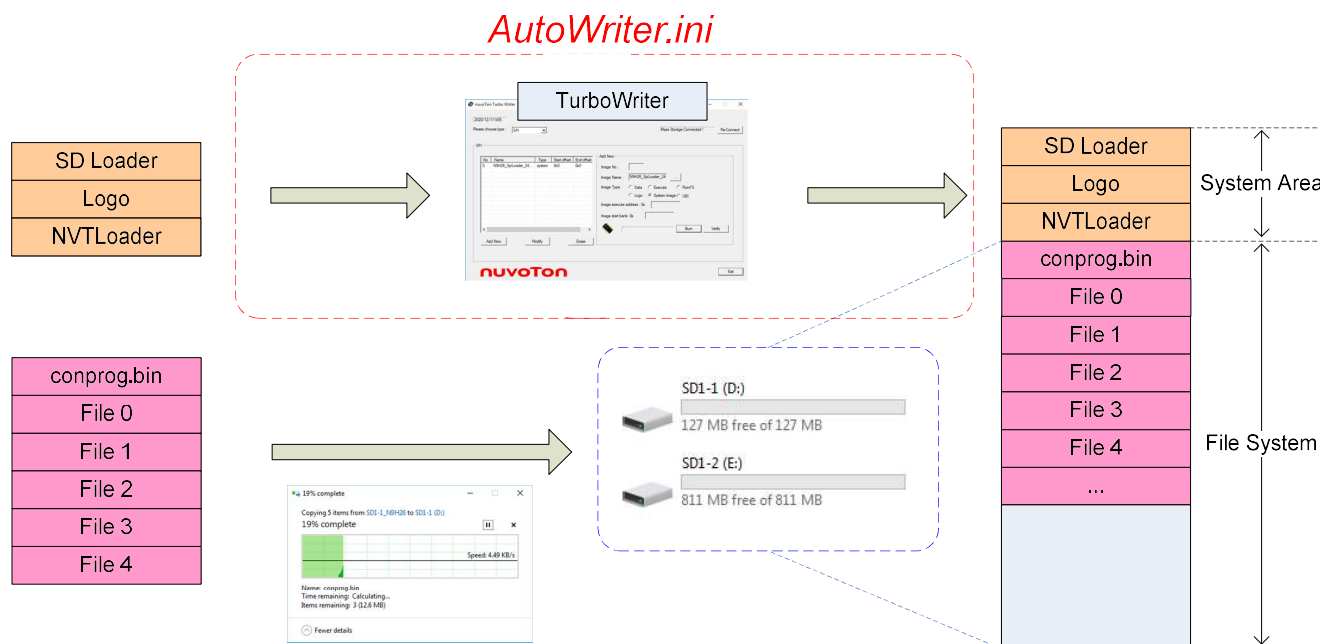


Figure 4-10 Using Auto Writer instead of Turbo Writer

5 Post Processing

The function of Posting Processing is to burn disk images or copy files to target disk.

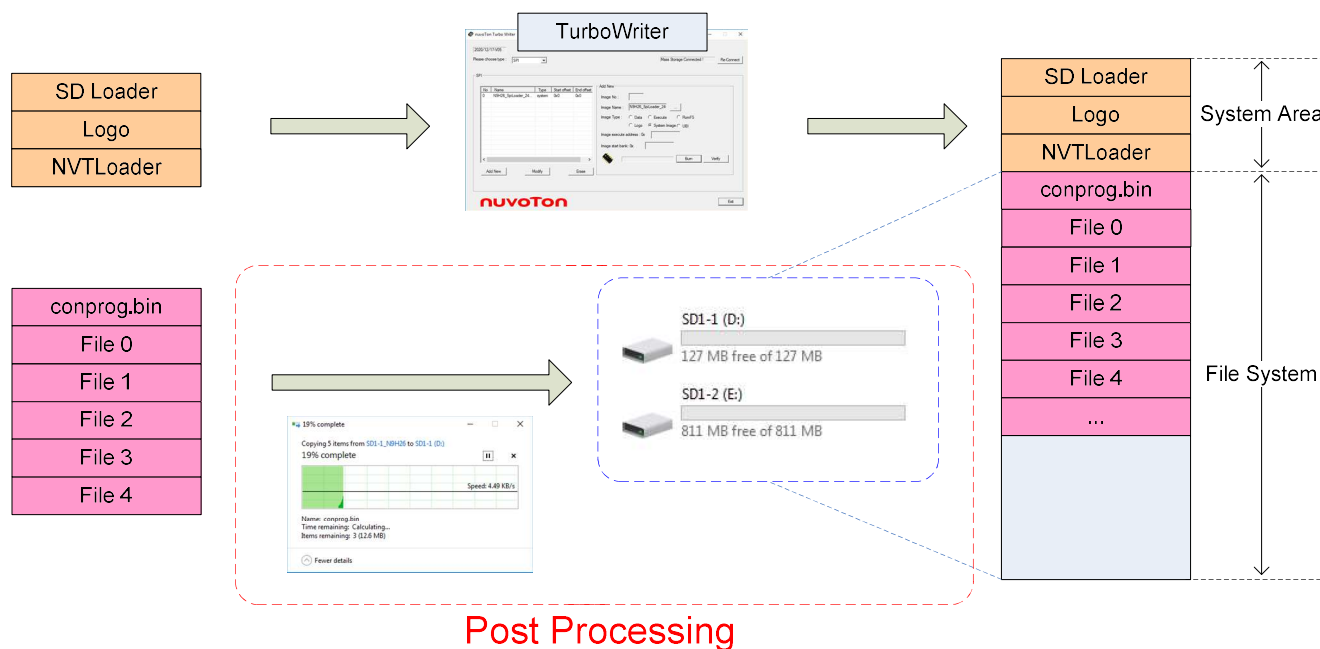


Figure 5-1 Using Auto Writer instead of copying file manually

Auto Writer supports two modes for Post Processing function. There are two radio buttons “Burn disk images” and “Copy the files and subfolders of selected folder”.

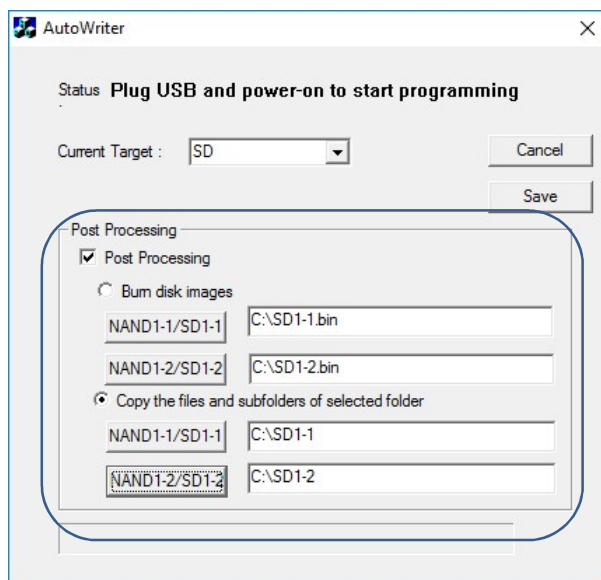


Figure 5-2 Post Processing Setting

- “Burn disk images”
User should use the disk image file created by NRomMaker.
- “Copy the files and subfolders of selected folder”
User should select the specified folders for the disks SD1-1 and SD1-2 of the storage, and copy all contents of the specified folder into the disks SD1-1 and SD1-2 of the storage.

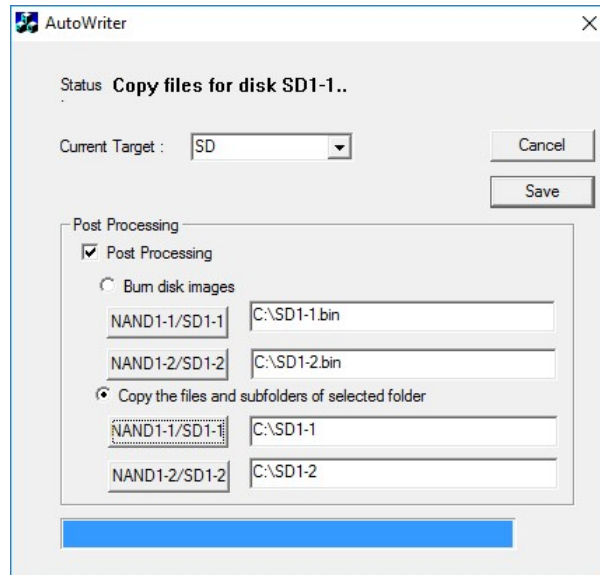


Figure 5-3 Post Processing – Copy files to SD1-1 & SD1-2

6 Multi-port function

From version 4.00.001, Auto Writer supports multi-ports function, user could select the processing number from the combo box “Max No.”. Currently four-ports is the maximum.

- If user selects four ports, plug in four devices to do the function. Otherwise Auto Writer fails to run. All devices must run the same mode (all run the recovery mode, or all run the normal mode). It fails that some runs the recovery mode and some runs the normal mode.
- User could use one USB hub to connect the four N3292x / N9H2x devices to run. Suggest run Windows 10 of PC excluding NB.
- User must plug one device in one by one, it is very important for the tool. In the beginning user plugs one board in and see the UI of tool, While the status displays the disk name or the error message, user could plug the second board in as follows. If user does not follow the above rule, the result will fail.

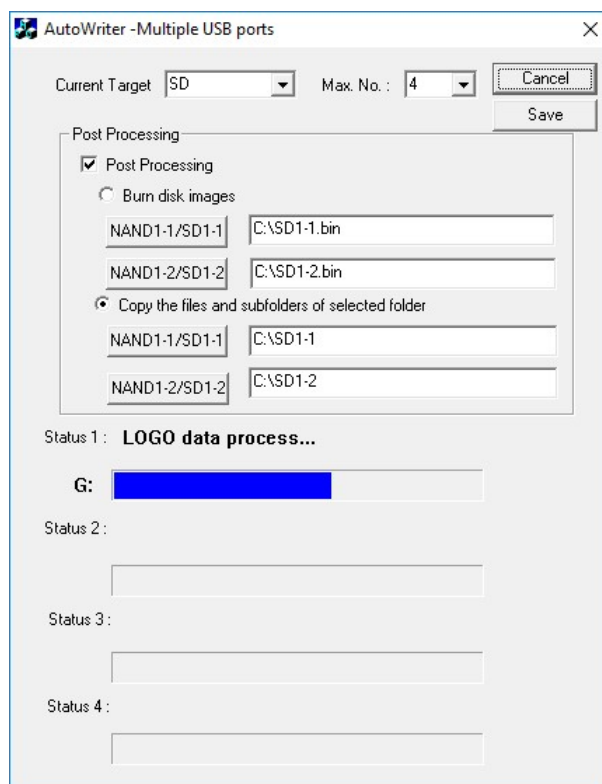


Figure 6-1 Multi-port function

7 Supporting Resources

The N9H2x / N329x 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 |
|------------|----------|--|
| 2021.06.17 | 4.01,003 | Revised |
| 2019.07.04 | 4.01,002 | Support plug in one by one continuously. |
| 2019.07.04 | 4.01,001 | Support plug in one by one, excluding plug in at the same time. |
| 2019.06.19 | 4.00,001 | Support Multiports for N3292X, N9H20 and N9H26 |
| 2018.04.27 | 3.01,001 | Support N9H20 and N9H26 |
| 2017.05.10 | 3.00.011 | Fix the issues of Windows 10 64 bits. |
| 2016.08.17 | 3.00.010 | Support IC of N32926 version D |
| 2015.09.16 | 3.00.009 | Add the partial erase of NAND |
| 2015.08.26 | 3.00.006 | Add verification function |
| 2015.01.16 | 3.00.005 | support to save the setting of UI |
| 2014.12.26 | 3.00.004 | support type "UBI" |
| 2014.11.04 | 3.00.003 | support type "DATA" support type "ROMFS" support to set the default target |
| 2014.06.27 | 3.00.002 | Support the partial erase for SPI flash |
| 2014.03.28 | 3.00.001 | Support the storage SD/SPI/SPI RAW |
| 2014.02.20 | 2.00.001 | Add the function of post process. |
| 2014.02.18 | 1.00.001 | Created |

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.*