AutoWriter
User Guide
V3.01.001

Publication Release Date: Aug. 2018

Support Chips:

N329 Series; M9H2X Series

Support Platforms:

Windows

The information in this document is subject to change without notice.

The Nuvoton Technology Corp. shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

Table of Contents

1. Introduction..4

1.1. AutoWriter Introduction..4

2. Opertaions 6

2.1. The usage of the file AutoWriter.ini...6

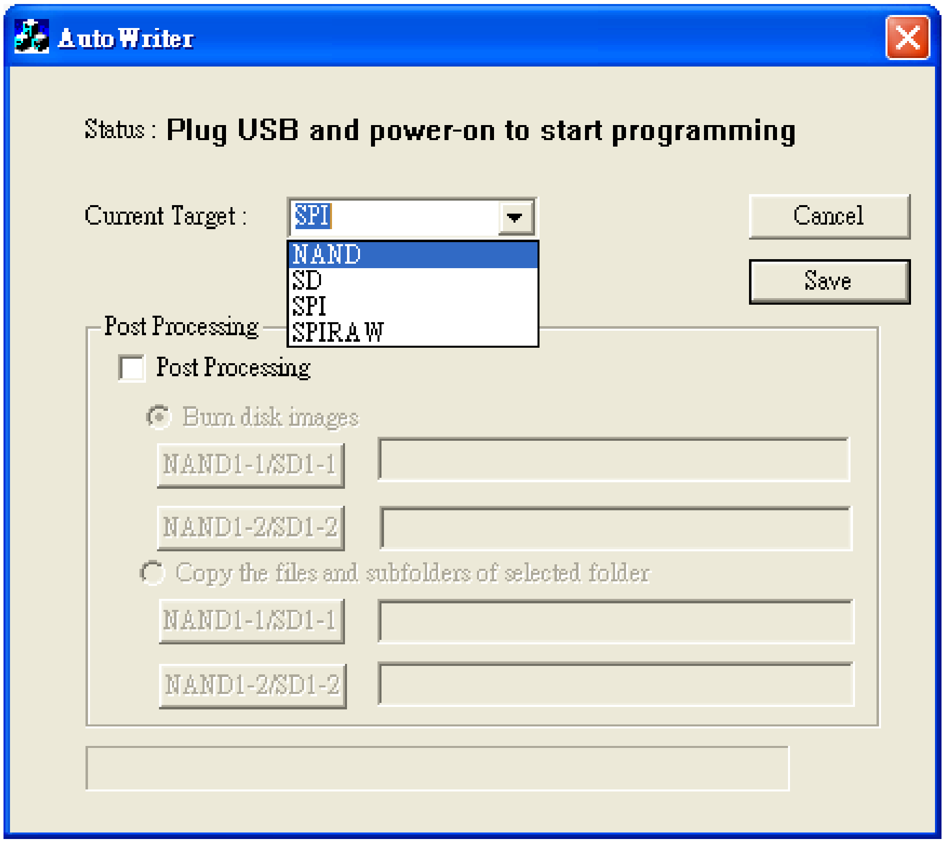
2.2. The operation of Post Processing 9

3. Revision History..12

1. Introduction

1.1. AutoWriter Introduction

AutoWriter is similar to TurboWriter, it is the simplified version of TurboWriter. AutoWriter supports the storage NAND/SD/SPI like TurboWriter for N329/N9H2X series. All binary files and *.ini files should be in the same folder of AutoWriter. There are 3 ini files, TurboWriter.ini, SPIFLASH ID.ini and AutoWriter.ini. User could obtain the file TurboWriter.ini from the tool TurboWriter if the file TurboWriter.ini does not exist. Make sure what chip do you use. If you use the chip N9H26K5, you must use the file TurboWriter.ini from the tool TurboWriter of N9H26K5. The file SPIFLASH ID.ini sets the size of SPI flash if AutoWriter does not support the SPI flash. AutoWriter will read the file SPIFLASH ID.ini to set the size of unsupported SPI flash. The file AutoWriter.ini is the script, we will introduce it later on.
After running AutoWriter, the UI of tool is shown as follows.



Status : Plug USB and power-on to start programming

Current Target : SPI

Post Processing

□ Post Processing

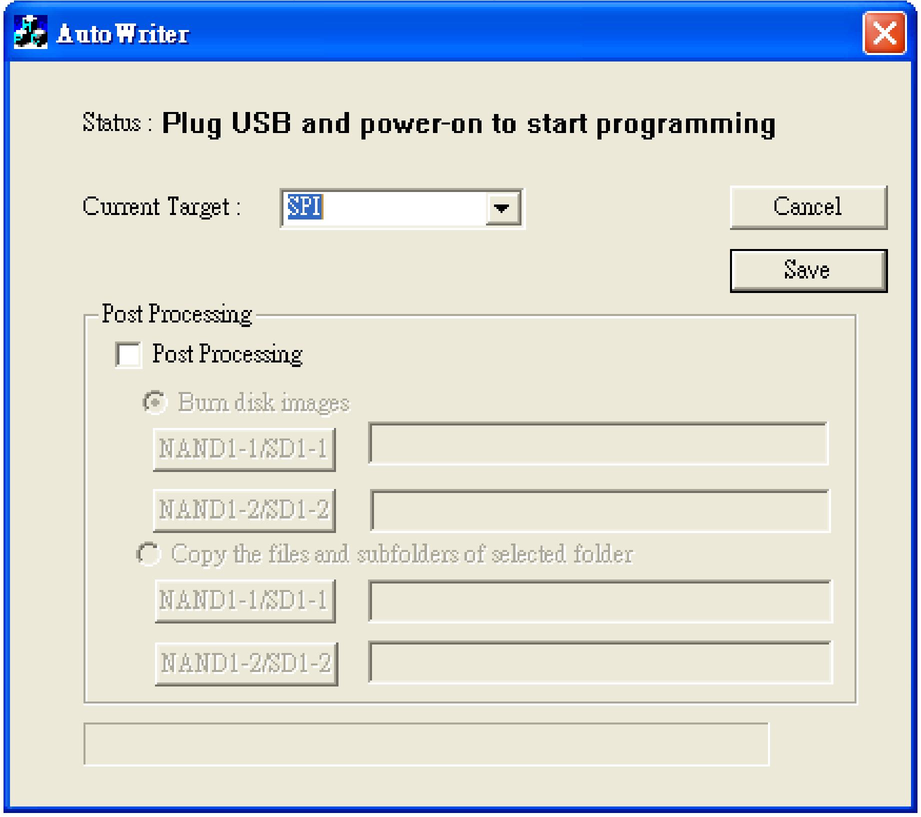
● Burn disk images

○ Copy the files and subfolders of selected folder

User must make sure plug USB cable in between EVB board and PC, and then restart the EVB board.
AutoWriter will burn some files by following the script of the file AutoWriter.ini
AutoWriter could support "predefined storage" to be the default target. There is one file Target.ini defined as follows

```
[Target]
SPI
```

User runs AutoWriter firstly, and Current Target is SPI, not NAND.



Status : Plug USB and power-on to start programming

Current Target : SPI

Post Processing

□ Post Processing

● Burn disk images

○ Copy the files and subfolders of selected folder

There are 4 kinds targets to set, NAND, SD, SPI and SPIRAW.
If user does not use the file target.ini, then NAND is the default target as before.
If user would like to save the setting of UI and uses the setting next time, press the button "Save" to exit.
Otherwise press the button "Cancel" to exit.

2. Opertaions

2.1. The usage of the file AutoWriter.ini

There is one file AutoWriter.ini within the same folder of AutoWriter, we will introduce some scripts as follows.

```
[Loader File Name]
APP = NandLoader.bin
ADDRESS= 900000
//Start Block = 0
```

The key [Loader File Name] is for Loader file, The description of APP is the file name NANDLoader.bin to be burned into the address 0x900000 of DRAM. It is the system type of TurboWriter. Under the folder of AutoWriter, the file NANDLoader.bin that user sets must exist, otherwise the processing of AutoWriter will stop. Currently the file name NANDLoader.bin is the pseudo file name, user must set the correct file name within the folder of AutoWriter. If you comment or delete the key, the function will not work. For N9H20K1/N329X1, the address should be 180000, the address is 700000 for N9H20K3/N329X3.

If user would like to set the start block of storage SPI/SD, set the key “Start Block = XXXX”, XXXX is the hexadecimal number. It is the sector unit (512 bytes) for SD, 64 Kbytes unit for SPI. If user does not the set key “Start Block=XXXX”, AutoWriter will work the suitable start address of storage SPI/SD automatically. About the storage SPIRAW, user could set only the file into the APP of [Loader File Name], and comment the other keys [Logo File Name] , [Execute File Name],….

```
[Logo File Name]
APP = Logo.dat
ADDRESS =    500000
```

The key [Logo File Name] is for Logo file, The description of APP is the file name Logo.dat to be burned into the address 0x500000 of DRAM. It is the logo type of TurboWriter. Under the folder of AutoWriter, the file logo.dat that user sets must exist, otherwise the processing of AutoWriter will stop. Currently the file name logo.dat is the pseudo file name, user must set the correct file name within the folder of AutoWriter. If you comment or delete the key, the function will not work.

If user would like to set the start block of storage SPI/SD, set the key “Start Block = XXXX”, XXXX is the hexadecimal number. It is the sector unit (512 bytes) for SD, 64 Kbytes unit for SPI. If user does not the set key “Start Block=XXXX”, AutoWriter will work the suitable start address of storage SPI/SD automatically.

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

The key [Execute File Name] is for execution file, The description of APP is the file name NvtLoader.bin to be burned into the address 0x800000 of DRAM. It is the exectute type of TurboWriter. Under the folder of AutoWriter, the file NvtLoader.bin that user sets must exist, otherwise the processing of AutoWriter will stop. Currently the file name NVTLoader.bin is the pseudo file name,

user must set the correct file name within the folder of AutoWriter. If you comment or delete the key, the function will not work.
If user would like to set the start block of storage SPI/SD, set the key “Start Block = XXXX”, XXXX is the hexadecimal number. It is the sector unit (512 bytes) for SD, 64 Kbytes unit for SPI. If user does not the set key “Start Block=XXXX”, AutoWriter will work the suitable start address of storage SPI/SD automatically.

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

The key [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 you comment or delete the key, the function will not work

```
[System Reserved MegaB]
//Unit : Mega Byte
NO =4
```

The key [System Reserved MegaB] sets the system area of storage to be 4MB. If you comment or delete the key, the function will not work.

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

The key [Options] sets NAND to be erased all or not. If you comment or delete the key, NAND will not be erased all. EraseAll=y will erase all NAND, EraseAll=n will not erase. It could support partial erase from block n to block m for SPI flash as follows. It could erase the block 0 to the block 0x1F for SPI flash and NAND. If the option Verify set to be ‘Y’ or ‘y’, then all downloaded files with verification function, it will take more time to finish.

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

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

The key [DRAM Download only] downloads into DRAM only, the file name RomFS.bin of DATA is at the address 0 of DRAM. Under the folder of AutoWriter, the file RomFS.bin that user sets must exist, otherwise the processing of AutoWriter will stop. Currently the file name RomFS.bin is the pseudo file name, user must set the correct file name within the folder of AutoWriter. If you comment or delete the key, the function will not work.

```
[Executing File Name]
APP = ResetByWDT.BIN
ADDRESS = 0
```

The key [Executing File Name] is to download and run. The description of APP is the file name ResetByWDT.BIN to be burned into the address 0 of DRAM, and.then run at address 0. Under the folder of AutoWriter, the file ResetByWDT.BIN must exist, otherwise the processing of AutoWriter will stop. User could do the file by youself. If you comment or delete the key, the function will not work.

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

```
APP2 = image2.bin
Start Block1 = 10
Start Block2 = 13
```

The key [ExecuteImage] is to burn some execute images into the storage, you could use TurboWriter to obtain the correct start block for the burned images. If NO=2, there must exist APP1, APP2 for 2 images, otherwise the processing will be incorrect. It is the exectute ype of TurboWriter. Under the folder of AutoWriter, the files image1.bin and image2.bin must exist, otherwise the processing of AutoWriter will stop. Currently the files image1.bin and image2.bin are the pseudo files, user must set the correct files within the folder of AutoWriter.If you comment or delete the key, the function will not work.

If user would like to set the start block of storage SPI/SD, set the key “Start Block1 = XXXX”, XXXX is the hexadecimal number. It is the sector unit (512 bytes) for SD, 64 Kbytes unit for SPI. If user does not the set key “Start Block1=XXXX”, AutoWriter will work the suitable start address of storage SPI/SD automatically.

```
[DATA]
No=2
APP1 = image1.bin
APP2 = image2.bin
Start Block1 = 10
Start Block2 = 1F
Execution Address1= 00200000
Execution Address2= 002A0000
```

The key [DATA] is to burn some data images into the storage. If NO=2, there must exist APP1, APP2 for 2 images, otherwise the processing will be incorrect. It is the data ype of TurboWriter. Under the folder of AutoWriter, the files image1.bin and image2.bin must exist, otherwise the processing of AutoWriter will stop. Currently the files image1.bin and image2.bin are the pseudo files, user must set the correct files within the folder of AutoWriter. If you comment or delete the key, the function will not work.

If user would like to set the start block of storage SPI/SD, set the key “Start Block1 = XXXX”, XXXX is the hexadecimal number. It is the sector unit (512 bytes) for SD. AutoWriter will work the suitable start address of storage SPI/SD/NAND automatically. Execution Address is the execution address of DRAM for specified data.

```
[ROMFS]
No=2
APP1 = image1.bin
APP2 = image2.bin
Start Block1 = 10
Start Block2 = 1F
Execution Address1= 00200000
Execution Address2= 002A0000
```

The key [ROMFS] is to burn some romfs images into the storage. If NO=2, there must exist APP1, APP2 for 2 images, otherwise the processing will be incorrect. It is the romfs type of TurboWriter. Under the folder of AutoWriter, the files image1.bin and image2.bin must exist, otherwise the processing of AutoWriter will stop. Currently the files image1.bin and image2.bin are the pseudo files, user must set the correct files within the folder of AutoWriter. If you comment or delete the key, the function will not work.

If user would like to set the start block of storage SPI/SD, set the key “Start Block1 = XXXX”, XXXX is the hexadecimal number. It is the sector unit (512 bytes) for SD. AutoWriter will work the suitable start address of storage SPI/SD/NAND automatically. Execution Address is the execution address of DRAM for specified data.

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

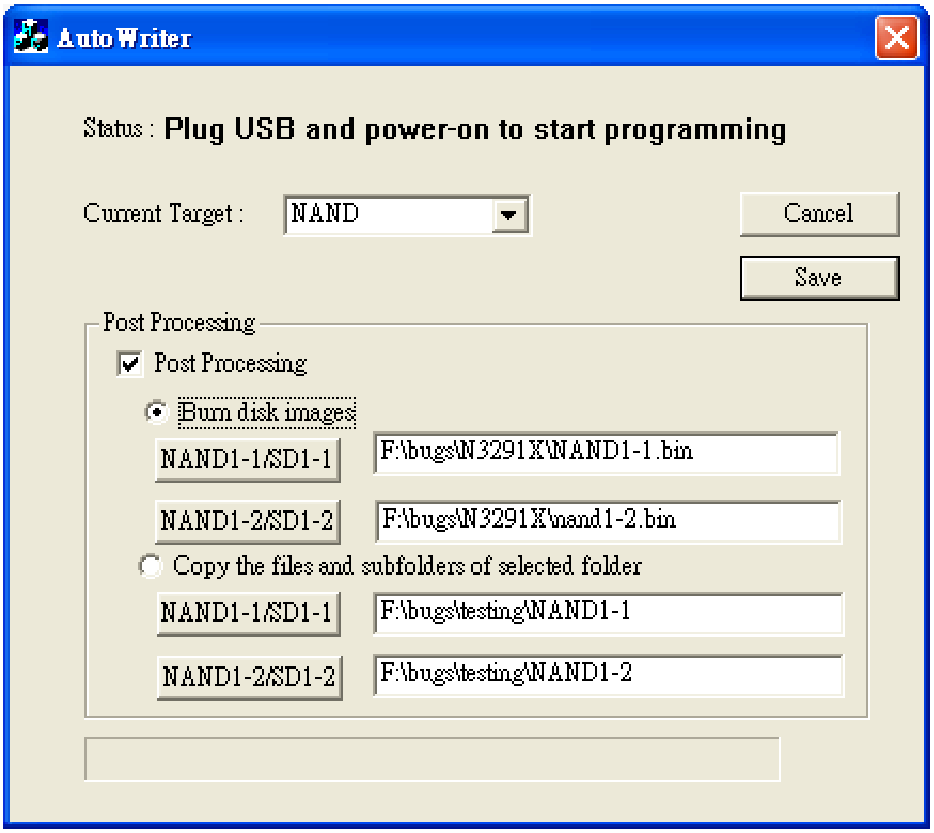
The key [UBI] is to burn some ubi images into the storage. If NO=2, there must exist APP1, APP2 for 2 images, otherwise the processing will be incorrect. It is the ubi type of TurboWriter. Under the folder of AutoWriter, the files image1.bin and image2.bin must exist, otherwise the processing of AutoWriter will stop. Currently the files image1.bin and image2.bin are the pseudo files, user must set the correct files within the folder of AutoWriter. If you comment or delete the key, the function will not work.
If user would like to set the start block of storage SPI/NAND, set the key “Start Block1 = XXXX”, XXXX is the hexadecimal number. AutoWriter will work the suitable start address of storage SPI/NAND automatically. Execution Address is the execution address of DRAM for specified data.

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

The key [Post Process] sets the size of storage. 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”.

2.2. The operation of Post Processing

User could set the check box “Post Processing” to work the jobs of normal mode as follows. Currently the post processing works for storage NAND.



Status : Plug USB and power-on to start programming

Current Target : NAND

Post Processing

☑ Post Processing

● Burn disk images

NAND1-1/SD1-1 F:\bugs\N3291X\NAND1-1.bin

NAND1-2/SD1-2 F:\bugs\N3291X\nand1-2.bin

○ Copy the files and subfolders of selected folder

NAND1-1/SD1-1 F:\bugs\testing\NAND1-1

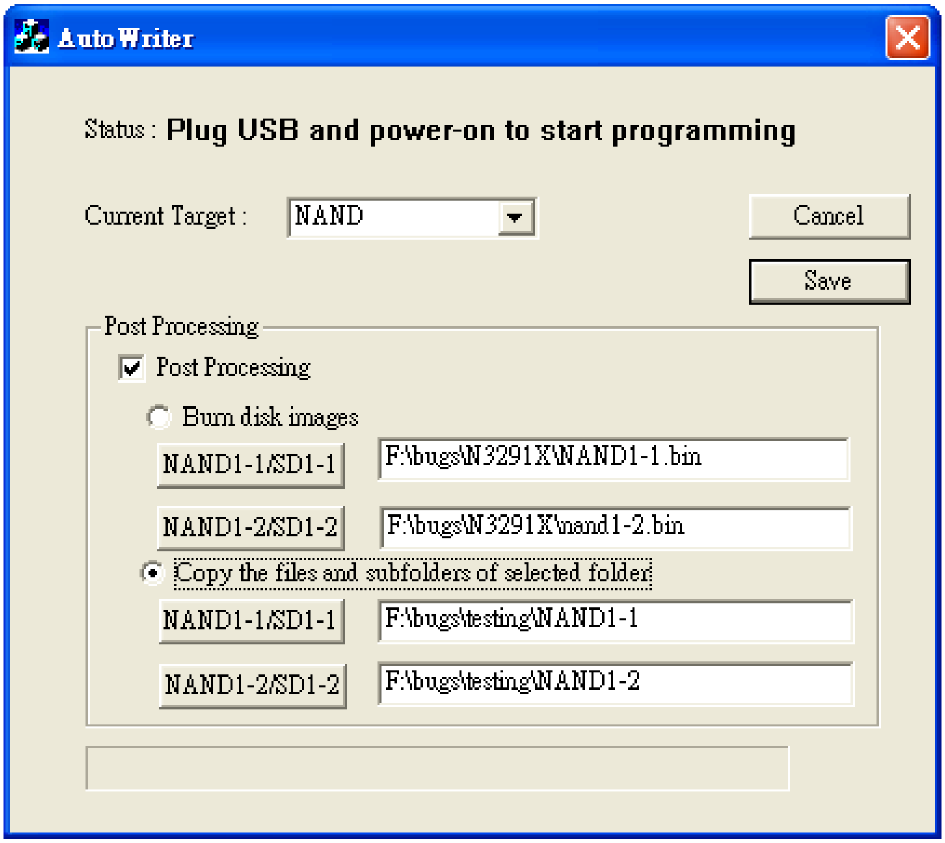
NAND1-2/SD1-2 F:\bugs\testing\NAND1-2

In general user burns the files “Loader.bin”, “logo.dat” and “NVTLoader.bin”, and then he needs to set the demo board to be normal mode, and then runs the normal mode to enter USB mass storage , to copy some files into the disks NAND1-1and NAND1-2 of the storage. The function of Posting Processing will work the above processing.

There are two radio buttons “Burn disk images” and “Copy the files and subfolders of selected folder”.

“Burn disk images” should use the built image file from NRomMaker. In the above figure, we burn two binary files NAND1-1.bin and NAND1-2.bin from NRomMaker. If user could not set the binary file by using the button NAND1-1 or NAND1-2, burn processing goes but no file exists within the disk NAND1-1 or NAND1-2 of the storge,

“Copy the files and subfolder of selected folder” should select the specified folders for the disks NAND1-1 and NAND1-2 of the storage, and copy all contents of the specified folder into the disks NAND1-1 and NAND1-2 of the storage. If user could not set the specified folder by using the button NAND1-1 or NAND1-2, burn processing goes but no file exists within the disk NAND1-1 or NAND1-2 of the storge,



Status : Plug USB and power-on to start programming

Current Target : NAND

Post Processing

☑ Post Processing

○ Burn disk images

F:\bugs\N3291X\NAND1-1.bin

F:\bugs\N3291X\nand1-2.bin

● Copy the files and subfolders of selected folder

F:\bugs\testing\NAND1-1

F:\bugs\testing\NAND1-2

3. Revision History

Version	Date	Description
V3.01,001	Aug. 10, 2018	● Support N9H2X series including N9H20K1/N9H20K3/N9H20K5 and N9H26K5.
V3.00.011	May 10, 2017	● Fix the issues of Windows 10 64 bits.
V3.00.010	Aug. 17, 2016	● Support IC of N32926 version D
V3.00.009	Sep. 16, 2015	● Add the partial erase of NAND
V3.00.006	Aug. 26, 2015	● Add verification function
V3.00.005	Jan. 16, 2015	● support to save the setting of UI
V3.00.004	Dec. 26, 2014	● support type “UBI”
V3.00.003	Nov. 4, 2014	● support type “DATA” ● support type “ROMFS” ● support to set the default target
V3.00.002	Jun.. 27, 2014	● Support the partial erase for SPI flash
V3.00.001	Mar. 28, 2014	● Support the storage SD/SPI/SPI RAW
V2.00.001	Feb. 20, 2014	● Add the function of post process.
V1.00.001	Feb. 18, 2014	● Created

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in equipment or systems intended for surgical implantation, atomic energy control instruments, aircraft or spacecraft instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for any other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications whereby failure could result or lead to personal injury, death or severe property or environmental damage.

Nuvoton customers using or selling these products for such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from their improper use or sales.