FA93 SPI Loader Reference Guide

V1.1

Publication Release Date: Apr. 2013

Support Chips:

W55FA Series

Support Platforms:

Non-OS

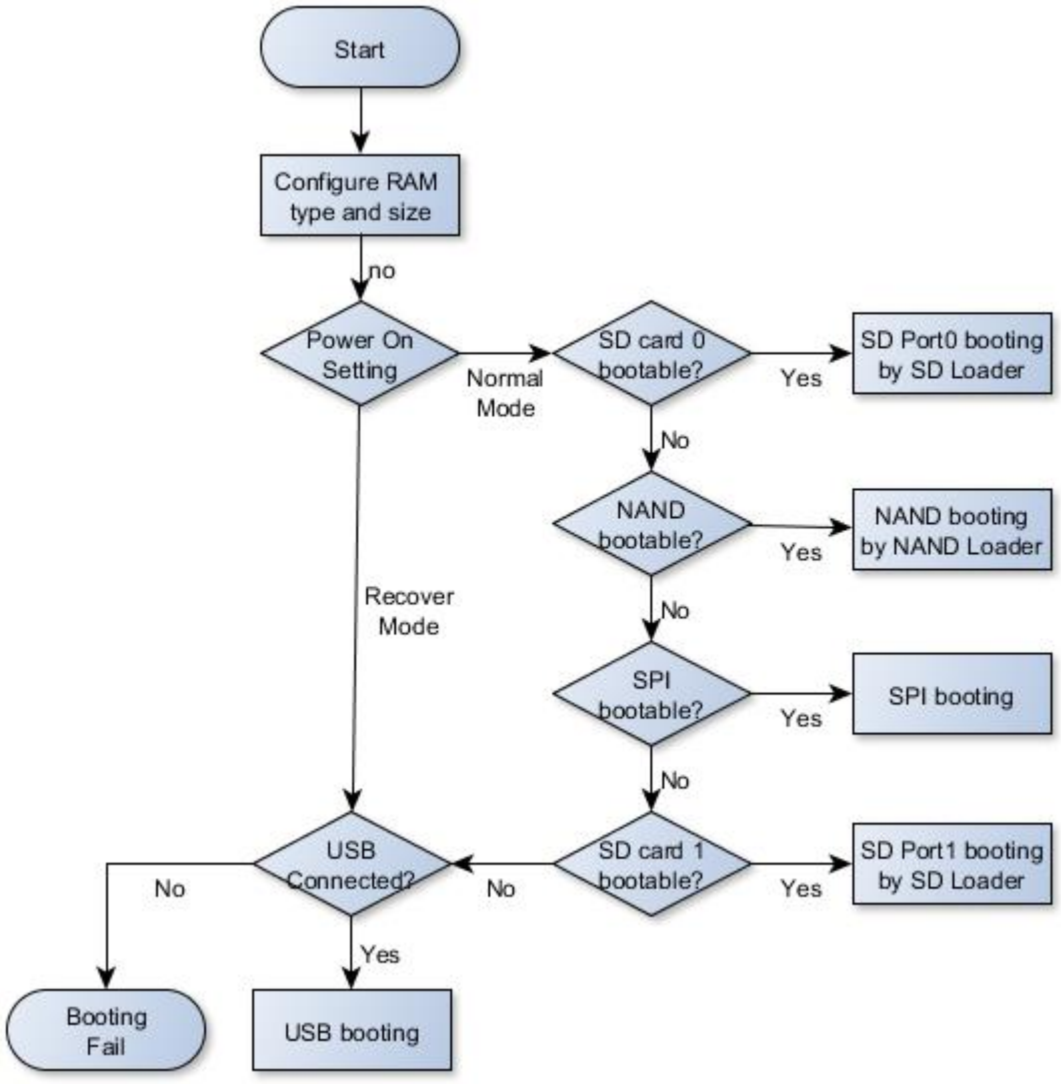
Table of Contents

1. General Description

FA93/VA93 Non-OS library consists of a set of libraries. These libraries are built to access those on-chip functions such as VPOST, APU, SIC, USBH, USBD, GPIO, I2C, SPI and UART, as well as File System (NVTFAT), USB MassStorage devices (UMAS) and NAND Flash devices (GNAND). This document describes the basic function of SPI Loader. With this introduction, user can quickly understand the SPI Loader on FA93/VA93 micro processor.

2. SPI Loader Overview

FA93/VA93 built-in 16K bytes IBR (Internal Booting ROM) where stored the boot loader to initial chip basically when power on, and then try to find out the next stage boot loader from different type of storage. It could be SD card, NAND, SPI Flash, or USB storage. The search sequence by IBR as below



The boot loader in IBR will hand over the chip controlling to SPI Loader if SD card 0 and NAND flash are not for booting. SPI Loader is a firmware stored at SPI Flash address 0x00000000.

2.1. SPI Loader Introduction

The SPI Loader has two version – One is SpiLoader & the other is SpiLoader_gzip which has decompression function.

SpiLoader

- Check, load, and display Logo image if it existed at SPI flash
- Check and load next firmware if it existed at SPI Flash
- Hand over chip controlling to next firmware.

SpiLoader_gzip

- Check, load, and display Logo image if it existed at SPI flash
- Check and load next firmware if it existed at SPI Flash
- Hand over chip controlling to next firmware.
    - It supports gzip decompression function for execute type image
        - If exxcute image has 64bytes u-Boot header, it will check the Compression type and decompression execute image to the execute address.
        - Execute type image address limitation
            - Because the compressed iamge is loaded to 0x200000 (2MB), user needs to make sure that the source data address is not conflict with destination address.

Please use SpiLoader project to build the SpiLoader or SpiLoader_gzip project to build the SpiLoader_gzip. The structure of SPI flash for SPI Loader is the same as NAND Loader and the basic unit of SPI Loader is one block (64KB). Here is an example for SpiLoader/SpiLoader_gzip.

	SPI Loader_gzip	**Logo Image**	**Execute Image**
Image No.	0	1	2
Image Name	*File name for SPI Loader on host*	*File name for Logo image on host*	*File name for Execute Image on host*
Image Type	System Image	Logo	Execute
Image execute address	***0x900000***	**0x500000**	**Any valid address**
Image start block	*Default value (0)*	*Behind Spi Loader*	*Behind Logo Image*

2.2. SpiLoader Framework

Difference between SpiLoader and SpiLoader_gzip

Because IBR SPI Booting Read operation takes more time than other booting, we hope the code size of SPI loader is as small as possible. We create two project files to build the SpiLoader with/without decompression function. (It takes about 11KB to deal with decompression) SpiLoader_gzip is used when code size is critical.

Image format for SpiLoader_gzip

The compressed file must created by gzip and it needs to have u-Boot image header as follows.

```
typedef struct image_header {
        uint32_t        ih_magic;       /* Image Header Magic Number        */
        uint32_t        ih_hcrc;        /* Image Header CRC Checksum        */
        uint32_t        ih_time;        /* Image Creation Timestamp         */
        uint32_t        ih_size;        /* Image Data Size                  */
        uint32_t        ih_load;        /* Data  Load  Address              */
        uint32_t        ih_ep;          /* Entry Point Address              */
        uint32_t        ih_dcrc;        /* Image Data CRC Checksum          */
        uint8_t         ih_os;          /* Operating System                 */
        uint8_t         ih_arch;        /* CPU architecture                 */
        uint8_t         ih_type;        /* Image Type                       */
        uint8_t         ih_comp;        /* Compression Type                 */
        uint8_t         ih_name[IH_NMLEN];      /* Image Name               */
} image_header_t;
```

[Note] SpiLoader only uses the fields ih_magic and ih_comp.

Spend time between SpiLoader and SpiLoader_gzip

Although the data SpiLoader_gzip needs to read is less than SpiLoader, it needs to take time to do decompression operation. Here is an example for SpiLoader/SpiLoader_gzip

Table 1 SpiLoader & SpiLoader_gzip size example

	Size
Normal spiLoader	17.8KB (18240Bytes)
spiLoader with gzip	29.0KB (29784Bytes)

Table 2 Spend time of SpiLoader & SpiLoader_gzip example (192MHz)

	Total Time	Un-Compressed time	Load image time	Image Size
Un-compressed image	2.156 s	N/A	1.828 s	2.62 MB (2748280Bytes)
gzip-compressed image	2.203 s	0.703 s	1.141 s	1.64 MB (1725536Bytes)

Table 3 Spend time of SpiLoader & SpiLoader_gzip example (240MHz)

	Total Time	Un-Compressed time	Load image time	Image Size
Un-compressed image	1.797 s	N/A	1.469 s	2.62 MB (2748280Bytes)
gzip-compressed image	1.828 s	0.536 s	0.922 s	1.64 MB (1725536Bytes)

[Note] The total time is from IBR starts to Linux Kernel Start.

Example – SpiLoader

- Environment
  - SPI Flash

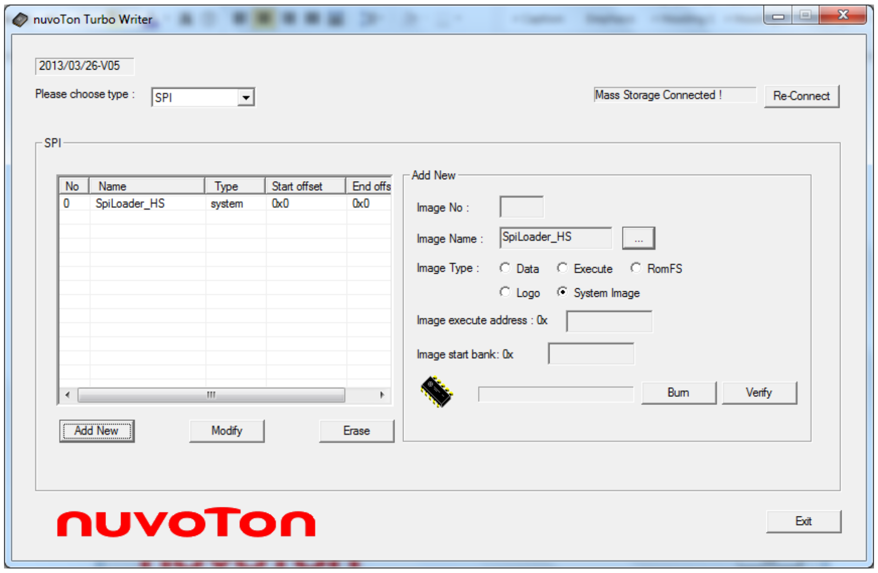
    SpiLoader_HS.bin –

    ◆ Choose the type “SPI”

    ◆ Set Image type “System Image”

    ◆ Browse the file “ SpiLoader_HS.bin”

    ◆ Press the button “Burn”



2013/03/26-V05

Please choose type : SPI

Mass Storage Connected !

No	Name	Type	Start offset	End offs
0	SpiLoader_HS	system	0x0	0x0

Image No :

Image Name : SpiLoader_HS

Image Type : Data Execute RomFS Logo [✓ System Image]

Image execute address : 0x

Image start bank: 0x

Figure 1 SpiLoader_gzip_HS.bin

NuvotonLogo_480x272.bin –

◆ Set Image type “Logo”

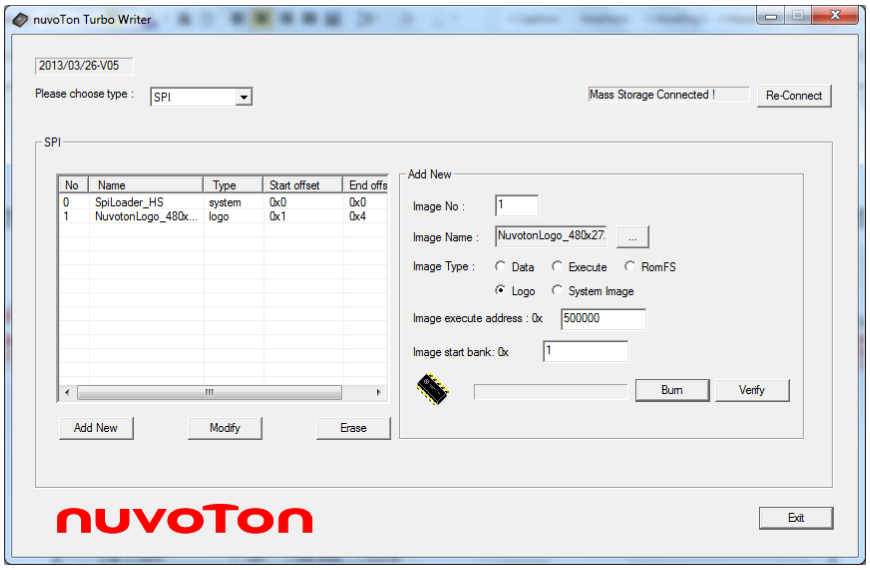
◆ Image number “1”

◆ Browse the file “NuvotonLogo_480x272.bin”

◆ Set the execute address: **0x500000**

◆ Set the start block number: **0x1**

◆ Press the button “Burn”.



nuvoTon Turbo Writer

2013/03/26-V05

Please choose type : SPI

Mass Storage Connected !

SPI

No	Name	Type	Start offset	End offs
0	SpiLoader_HS	system	0x0	0x0
1	NuvotonLogo_480x...	logo	0x1	0x4

Add New

Image No : 1

Image Name : NuvotonLogo_480x27

Image Type : ( ) Data ( ) Execute ( ) RomFS [✓ Logo] ( ) System Image

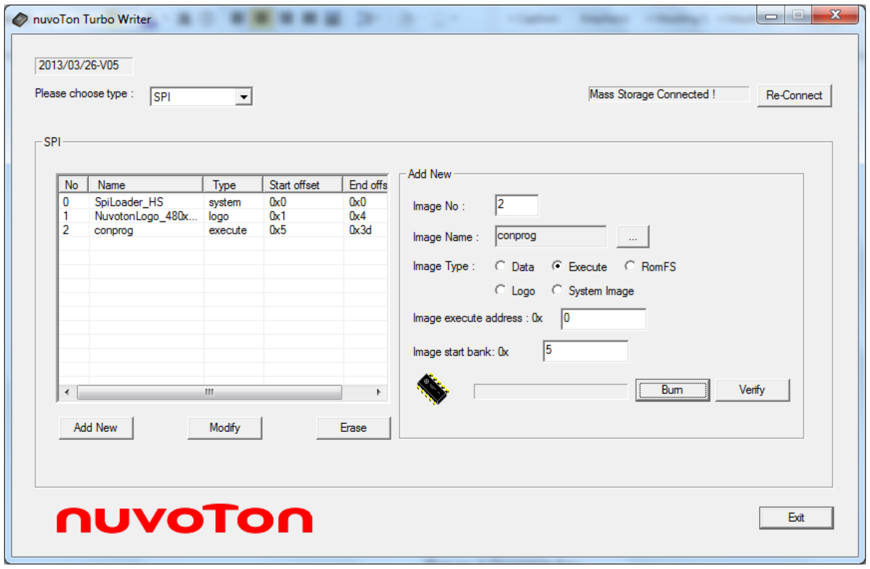
Image execute address : 0x 500000

Image start bank: 0x 1

Figure 2 NuvotonLogo_480x272.bin

Conprog.bin –

- ◆ Set Image type “Execute”
- ◆ Image number “1”
- ◆ Browse the file “Conprog.img”
- ◆ Set the execute address: **0x000000**
- ◆ Set the start block number: **0x5**
- ◆ Press the button “Burn”.



nuvoTon Turbo Writer

2013/03/26-V05

Please choose type : SPI

Mass Storage Connected !

SPI

No	Name	Type	Start offset	End offs
0	SpiLoader_HS	system	0x0	0x0
1	NuvotonLogo_480x...	logo	0x1	0x4
2	conprog	execute	0x5	0x3d

Add New

Image No : 2

Image Name : conprog

Image Type : ( ) Data [✓ Execute] ( ) RomFS ( ) Logo ( ) System Image

Image execute address : 0x 0

Image start bank: 0x 5

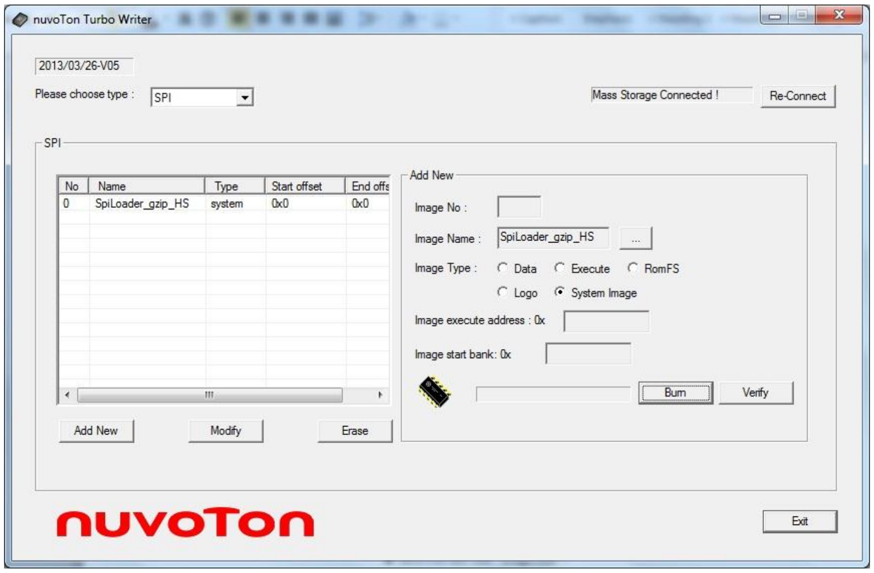
Figure 3 Conprog.bin

[Message Log – SpiLoader]

```
Init RTC....OK
DDR size: 32MB
SD Port0 Booting Fail - No/Bad Card Insert
NAND Booting (2K-page 4 Address Cycle) Fail - Not for Booting
SPI Booting Success
Clock Skew
  DQSODS 0x1010
  CKDQSDS 0xAAAA00
Code Executes at 0x00900000
SPI Loader start
Load Image Load file length 0x400, execute address 0x809048FC
Load file length 0x3FC00, execute address 0x500000
Load file length 0x388F00, execute address 0x0
Jump to kernel    Linux version 2.6.35.4 (root@CentOS.Server) (gcc version 4.2.1) #23
```

Example – SpiLoader_gzip

- Environment
  - SPI Flash
    SpiLoader_gzip_HS.bin –
    ◆ Choose the type "SPI"
    ◆ Set Image type "System Image"
    ◆ Browse the file " SpiLoader_gzip_HS.bin"
    ◆ Press the button "Burn"



Please choose type : SPI

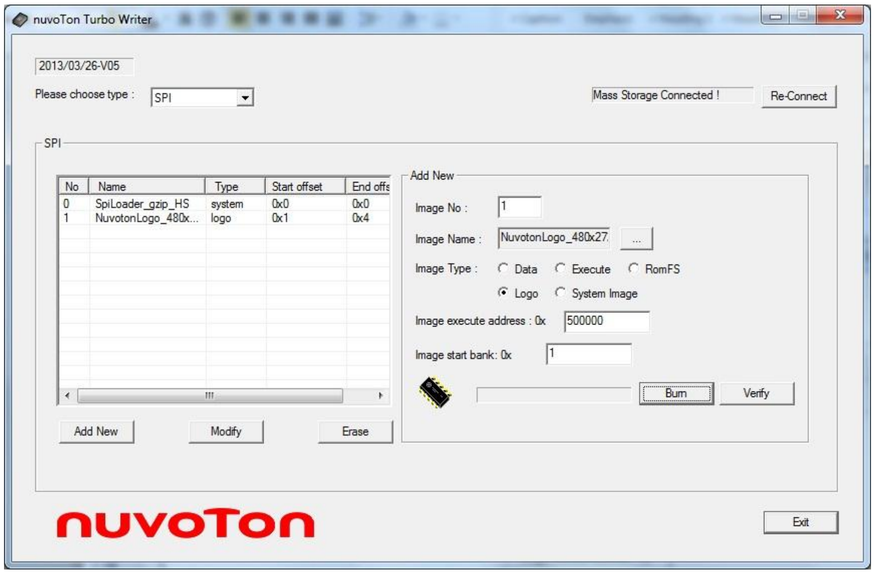
No	Name	Type	Start offset	End offs
0	SpiLoader_gzip_HS	system	0x0	0x0

Image No :
Image Name : SpiLoader_gzip_HS
Image Type : Data / Execute / RomFS / Logo / [✓ System Image]
Image execute address : 0x
Image start bank: 0x

Figure 4 SpiLoader_gzip_HS.bin

NuvotonLogo_480x272.bin –

◆ Set Image type “Logo”
◆ Image number “1”
◆ Browse the file “NuvotonLogo_480x272.bin”
◆ Set the execute address: **0x500000**
◆ Set the start block number: **0x1**
◆ Press the button “Burn”.



No	Name	Type	Start offset	End offs
0	SpiLoader_gzip_HS	system	0x0	0x0
1	NuvotonLogo_480x...	logo	0x1	0x4

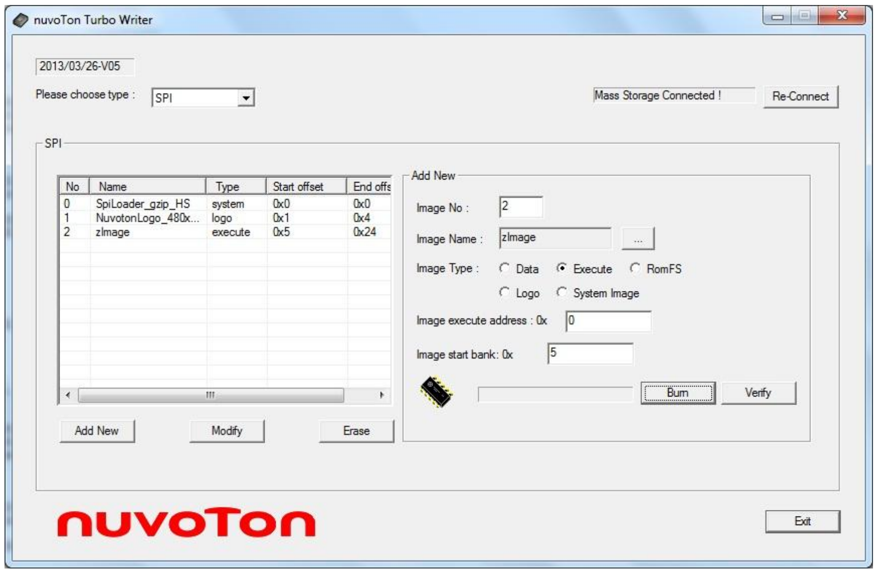
Add New
Image No : 1
Image Name : NuvotonLogo_480x27
Image Type : Data / Execute / RomFS / [✓ Logo] / System Image
Image execute address : 0x 500000
Image start bank: 0x 1

Figure 5 NuvotonLogo_480x272.bin

[Compressed execute image]

zImage.img –

◆ Set Image type “Execute”
◆ Image number “1”
◆ Browse the file “zImage.img”
◆ Set the execute address: **0x000000**
◆ Set the start block number: **0x5**
◆ Press the button “Burn”.



2013/03/26-V05

Please choose type : SPI

Mass Storage Connected !

SPI

No	Name	Type	Start offset	End offs
0	SpiLoader_gzip_HS	system	0x0	0x0
1	NuvotonLogo_480x...	logo	0x1	0x4
2	zImage	execute	0x5	0x24

Add New

Image No : 2

Image Name : zImage

Image Type : ( ) Data [✓ Execute] ( ) RomFS ( ) Logo ( ) System Image

Image execute address : 0x 0

Image start bank: 0x 5

Figure 6 zImage.img

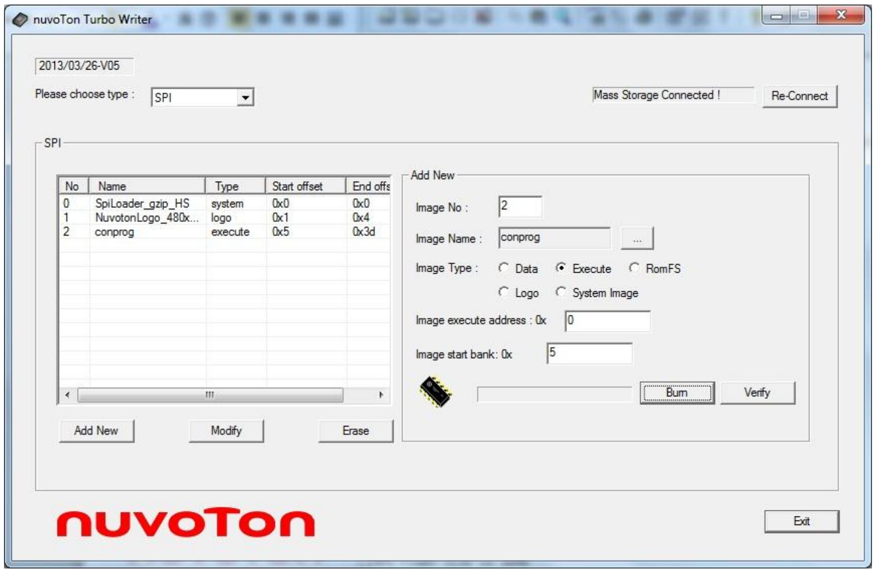
[Message Log – Compressed execute image]

```
Init RTC....OK
DDR size: 32MB
SD Port0 Booting Fail - No/Bad Card Insert
NAND Booting (2K-page 4 Address Cycle) Fail - Not for Booting
SPI Booting Success
Clock Skew
  DQSODS 0x1010
  CKDQSDS 0xAAAA00
Code Executes at 0x00900000
SPI Loader start
Load Image Load file length 0x400, execute address 0x80907618
Load file length 0x3FC00, execute address 0x500000
Load file length 0x40, execute address 0x200000
## Booting image at 0x00200000 ...
Get Magic Number
Load file length 0x1F6294, execute address 0x200000
## Booting image at 0x00200000 ...
Get Magic Number
      Gzip Uncompressing to 0x0 ... OK
Jump to kernelLinux version 2.6.35.4 (root@CentOS.Server) (gcc version 4.2.1) #23
```

[Un-compressed execute image]

Conprog.bin –

- ◆ Set Image type “Execute”
- ◆ Image number “1”
- ◆ Browse the file “Conprog. bin”
- ◆ Set the execute address: **0x000000**
- ◆ Set the start block number: **0x5**
- ◆ Press the button “Burn”.



2013/03/26-V05

Please choose type : SPI

Mass Storage Connected !

No	Name	Type	Start offset	End offs
0	SpiLoader_gzip_HS	system	0x0	0x0
1	NuvotonLogo_480x...	logo	0x1	0x4
2	conprog	execute	0x5	0x3d

Image No : 2

Image Name : conprog

Image Type : Data [✓ Execute] RomFS Logo System Image

Image execute address : 0x 0

Image start bank: 0x 5

Figure 7 Conprog.bin

[Message Log – Un-compressed execute image]

```
Init RTC....OK
DDR size: 32MB
SD Port0 Booting Fail - No/Bad Card Insert
NAND Booting (2K-page 4 Address Cycle) Fail - Not for Booting
SPI Booting Success
Clock Skew
  DQSODS 0x1010
  CKDQSDS 0xAAAA00
Code Executes at 0x00900000
SPI Loader start
Load Image Load file length 0x400, execute address 0x80907618
Load file length 0x3FC00, execute address 0x500000
Load file length 0x40, execute address 0x200000
## Booting image at 0x00200000 ...
Bad Magic Number
Load file length 0x388F00, execute address 0x0
Jump to kernelLinux version 2.6.35.4 (root@CentOS.Server) (gcc version 4.2.1) #23
```

3. Revision History

Version	Date	Description
V1.0	Mar. 3, 2011	● Created
V1.1	Apr. 2, 2013	● Modify for SpiLoader update ● Add description for SpiLoader_gzip