MTD NAND Solution Guide

v1.02

Publication Release Date: Aug, 2014

Contents

1. Introduction

A Memory Technology Device (MTD) is a type of device file in Linux for interacting with flash memory. The MTD subsystem was created to provide an abstraction layer between the hardware-specific device drivers and higher-level applications.

Some resources for you reference:

(1) YAFFS2
http://en.wikipedia.org/wiki/YAFFS

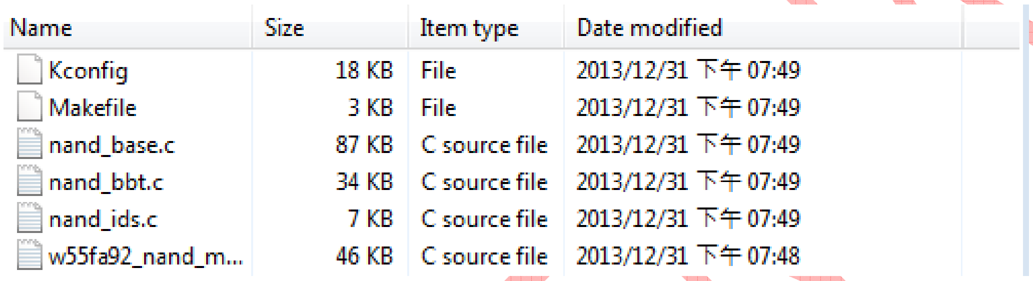
(2) UBIFS
http://en.wikipedia.org/wiki/UBIFS
http://www.linux-mtd.infradead.org/doc/ubifs.html#L_overview

(3) FLASH FILE SYSTEMS FOR MTD DEVICES: JFFS2, YAFFS2, UBIFS AND LOGFS.
http://www.lindusembedded.com/blog/2010/05/20/flash-file-systems-for-mtd-devices-jffs2-yaffs2-ubifs-and-logfs/

2. How to update source files

2.1.To update NAND controller driver

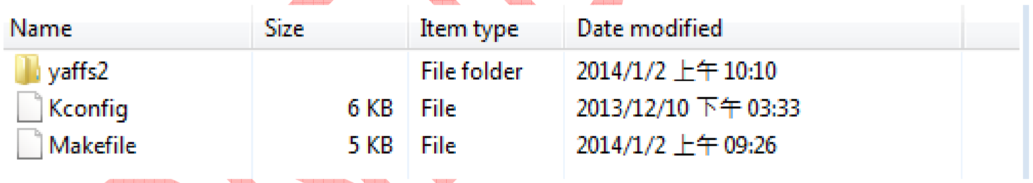
Update these files to the path as <BSP>\linux_2.6.35.4_fa92\drivers\mtd\nand\



Name	Size	Item type	Date modified
Kconfig	18 KB	File	2013/12/31 下午 07:49
Makefile	3 KB	File	2013/12/31 下午 07:49
nand_base.c	87 KB	C source file	2013/12/31 下午 07:49
nand_bbt.c	34 KB	C source file	2013/12/31 下午 07:49
nand_ids.c	7 KB	C source file	2013/12/31 下午 07:49
w55fa92_nand_m...	46 KB	C source file	2013/12/31 下午 07:48

2.2.To update YAFFS2

Copy these files to the path as <BSP>\linux_2.6.35.4_fa92\fs\



Name	Size	Item type	Date modified
yaffs2		File folder	2014/1/2 上午 10:10
Kconfig	6 KB	File	2013/12/10 下午 03:33
Makefile	5 KB	File	2014/1/2 上午 09:26

3. How to apply in kernel configuration

3.1.Built-in MTD NAND Controller driver

In Linux kernel menu configuration, you must to select options as follows in Linux kernel menu configuration.

```
     Device Drivers  --->
          <*> Memory Technology Device (MTD) support  --->
               [*]   MTD partitioning support
               <*>  Direct char device access to MTD devices
               -*-   Common interface to block layer for MTD 'translation layers
               <*>  Caching block device access to MTD devices
               <*>  NAND Device Support  --->
                    <*>   Support for NAND on Nuvoton W55FA92 demo board
```

Notice:

(1) The driver conflicts with NAND driver based on SCSI layer.

OOB layout on NAND flash:

Bad block marker	User available	ECC parity code
Always 4-Bytes	Reserved	The size depends on BCH algorithm and OOB size.

3.2.How to enable UBIFS

If you prefer UBIFS, you can select options as follows in Linux kernel menu configuration.

```
     Device Drivers  -→
        <*> Memory Technology Device (MTD) support  --->
          UBI - Unsorted block images  --->
            <*> Enable UBI
            (4096) UBI wear-leveling threshold
            (5) Percentage of reserved eraseblocks for bad eraseblocks hand
    File systems  --->
          [*] Miscellaneous filesystems  --->
            <*>     UBIFS file system support
            [*]     Extended attributes support
            [*]     Advanced compression options
            [*]       LZO compression support
            [*]       ZLIB compression support
```

3.3.How to enable YAFFS2

If you prefer YAFFS2, you can select as below options in Linux kernel menu configuration.

```
     File systems  --->
          [*] Miscellaneous filesystems  --->
            <*>   yaffs2 file system support
            -*-      512 byte / page devices
            -*-      2048 byte (or larger) / page devices
            [*]        Autoselect yaffs2 format
            [*]      Enable yaffs2 xattr support
```

3.4.How to enable VPOST Init By Driver

If you prefer to enable LCM driver, please also need to set “vpost init by driver” as below options in Linux kernel menu configuration.

```
     Device Drivers  --->
          Graphics support  --->
            --- Support for frame buffer devices
            [*]   W55FA92 LCD framebuffer support
                 Select VPOST init setting (vpost init by driver)  --->
```

4. How to layout MTD partition table

You can modify the MTD partition table in drivers/mtd/nand/w55fa92_nand_mtd.c.

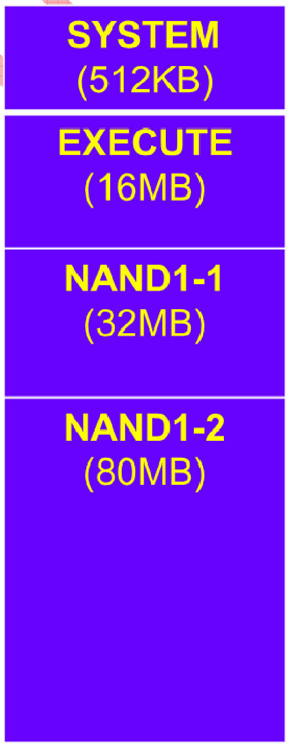
For example, we layout four partitions in the table as :

(1) The **SYSTEM** partition size always is fixed to 4-block by default. The driver will calculate automatically after identifying what flash on board. It is used to save NAND loader and booting information.

(2) The **EXECUTE** partition size is decided by user. It is used to save Linux kernel and logo part.

(3) The **NAND1-1/NAND1-2** is optional. The partition size is decided by user. It is used to save user-data based on file system.

```
/* Define your partitions */
static const struct mtd_partition partitions[]
= {
 {
    .name = "SYSTEM",
    .offset = 0,
    .ecclayout = &w55fa92_nand_SYSTEM_oob
 },
 {
    .name = "EXECUTE",
    .offset = MTDPART_OFS_APPEND,
    .ecclayout = &w55fa92_nand_EXECUTE_oob
    .size = 16 * 1024 * 1024,
 },
 {
    .name = "NAND1-1",
    .offset = MTDPART_OFS_APPEND,
    .size = 32 * 1024 * 1024
 },
 {
    .name = "NAND1-2",
    .offset = MTDPART_OFS_APPEND,
    .size = MTDPART_SIZ_FULL
 }
};
```



Notice: Please keep SYSTEM and EXECUTE partition in the table. They will be used for System-area update.///

5. How to make image on PC

5.1.To make YAFFS2 image

We ported YAFFS2 utility from YAFFS2 source pack for making image. The usage is as follows:

```
#mkyaffs2image: image building tool for YAFFS2 built Jan  2 2014
#Missing <dir> and/or <image.yaffs2> argument
#Usage: mkyaffs2image [Options] <dir> <image_file>
#          <dir>         the directory tree to be converted
#          <image.yaffs2> the output file to hold the image
#Options:
#  -c | --convert       produce a big-endian image from a little-endian machine
#  -p | --page-size      Page size of target NAND device [2048]
#  -o | --oob-size      OOB size of target NAND device [64]
#  -b | --block-size     Block size of target NAND device [0x20000]
#  -i | --inabnd-tags     Use Inband Tags
#  -n | --no-pad       Do not pad to end of block
```

If the page size of NAND flash is 2K and oobsize is 64, you can execute the command:

```
# Make an YAFFS2 image with inband-tags (blocksize=128KB, pagesize=2048, oobsize=64)
./mkyaffs2image -i ./foo ./foo_yaffs2.img
```

For 4K+224 NAND example:

```
# Make an YAFFS2 image with inband-tags (blocksize=1MB, pagesize=4096, oobsize=224)
./mkyaffs2image -b 0x100000 -p 4096 -o 224 -i ./foo ./foo_yaffs2.4k.img
```

5.2.To make UBIFS image

If the page size of NAND flash is 2K, you can execute the command:

Step 1: Make a UBIFS file system image from an existing directory tree

```
Usage: mkfs.ubifs [OPTIONS] target
-r, -d, --root=DIR       build file system from directory DIR
-e, --leb-size=SIZE      logical erase block size
-c, --max-leb-cnt=COUNT  maximum logical erase block count
-o, --output=FILE        output to FILE

# Make an UBIFS image 2K
mkfs.ubifs -r /tmp/foo -m 2048 -e 126976 -c 256 -o ./foo_ubifs
```

Step 2: To generate a UBIFS images.

```
Usage: ubinize [-o filename] [-p <bytes>] [-m <bytes>] [-s <bytes>] [-O <num>] [-e
<num>]
            [-x <num>] [-Q <num>] [-v] [-h] [-V] [--output=<filename>]
    [--peb-size=<bytes>]
            [--min-io-size=<bytes>] [--sub-page-size=<bytes>]
    [--vid-hdr-offset=<num>]
            [--erase-counter=<num>] [--ubi-ver=<num>] [--image-seq=<num>]
    [--verbose] [--help]
            [--version] ini-file

echo "[ubifs]" > ./ubifs.cfg
echo "mode=ubi" >> ./ubifs.cfg
echo "vol_id=0" >> ./ubifs.cfg
echo "vol_type=dynamic" >> ./ubifs.cfg
echo "vol_alignment=1" >> ./ubifs.cfg
echo "vol_name=foo" >> ./ubifs.cfg
echo "vol_flags=autoresize" >> ./ubifs.cfg
echo "image=foo_ubifs" >> ./ubifs.cfg

# Notice vol_size value < mtd partition size
# echo "vol_size=60MiB" >> ./ubifs.cfg

# Generate an UBIFS image 2K
ubinize -o ./foo_ubifs.img -m 2048 -p 128KiB -s 2048 ./ubifs.cfg
```

6. Some utilities

6.1.fs_writer

We provide some script files to introduce about “How to erase/burn/mount on N329 series platform”.

For YAFFS2 example,

```
#!/bin/sh
mkdir /mnt/yaffs2
./mtdutils/flash_erase /dev/mtd3 0 0
if ./mtdutils/nandwrite -a -m /dev/mtd3 foo_yaffs2.img > /dev/null; then
    if mount -t yaffs2 -o"inband-tags" /dev/mtdblock3 /mnt/yaffs2; then
        echo "Success"
        cat /mnt/yaffs2/hi.txt
        exit 0
    fi
fi
```

※ You can burn YAFFS2 image with inband-tags to NAND chip by Turbowriter.

For UBIFS example,

```
#!/bin/sh

PAGESIZE=2048
UBIFSIMGPATH=./ubifs/foo_ubifs.img

echo "mtdutils/flash_erase"
./mtdutils/flash_erase /dev/mtd3 0 0
echo "mtdutils/ubiformat"
if ./mtdutils/ubiformat /dev/mtd3 -O $PAGESIZE -s $PAGESIZE -f $UBIFSIMGPATH >
/dev/null 2>&1; then
    echo "mtdutils/ubiattach"
    if ./mtdutils/ubiattach /dev/ubi_ctrl -m 3 > /dev/null 2>&1; then
        echo "mtdutils/ubimkvol"
        mkdir -p /mnt/ubifs
        echo "mount"
        if mount -t ubifs ubi0:foo /mnt/ubifs > /dev/null 2>&1; then
            echo "Success!!"
            cat /mnt/ubifs/helloworld.txt
            exit 0
        fi
    fi
fi
```

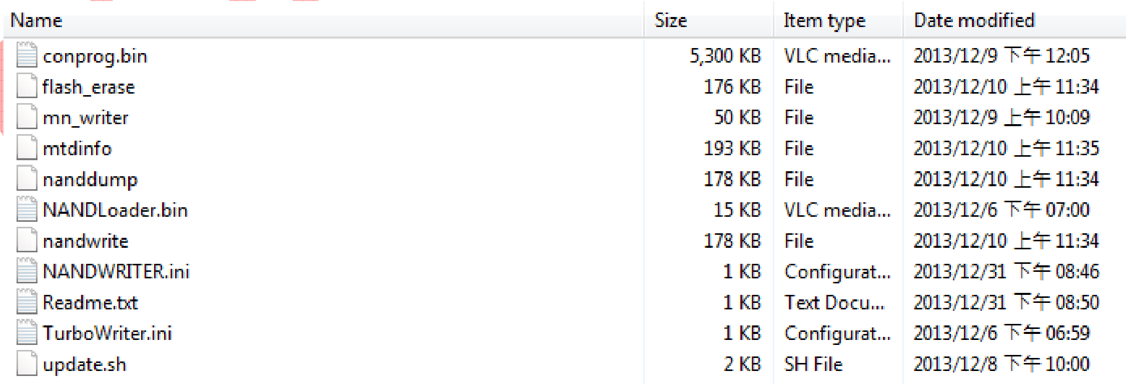
```
./mtdutils/ubidetach -m 3
```

For mounting empty partition as UBIFS

```
#!/bin/sh

PAGESIZE=2048
PartitionSize=60MiB

./mtdutils/flash_erase /dev/mtd3 0 0
echo "mtdutils/ubiformat"
if ./mtdutils/ubiformat /dev/mtd3 -O $PAGESIZE -s $PAGESIZE > /dev/null 2>&1; then
    echo "mtdutils/ubiattach"
    if ./mtdutils/ubiattach /dev/ubi_ctrl -m 3 > /dev/null 2>&1; then
        echo "mtdutils/ubimkvol"
        if ./mtdutils/ubimkvol /dev/ubi0 -N test_volume -s $PartitionSize >
/dev/null 2>&1; then
            if [ ! -d "/mnt/ubifs" ]; then
                mkdir /mnt/ubifs
            fi
            if mount -t ubifs ubi0:test_volume /mnt/ubifs > /dev/null 2>&1; then
                echo "Success!!"
                exit 0
            fi
        fi
    fi
fi
./mtdutils/ubidetach -m 3
```

6.2. mn_writer

This tool can program system area parts which include SYSTEM and EXECUTE.



Name	Size	Item type	Date modified
conprog.bin	5,300 KB	VLC media...	2013/12/9 下午 12:05
flash_erase	176 KB	File	2013/12/10 上午 11:34
mn_writer	50 KB	File	2013/12/9 上午 10:09
mtdinfo	193 KB	File	2013/12/10 上午 11:35
nanddump	178 KB	File	2013/12/10 上午 11:34
NANDLoader.bin	15 KB	VLC media...	2013/12/6 下午 07:00
nandwrite	178 KB	File	2013/12/10 上午 11:34
NANDWRITER.ini	1 KB	Configurat...	2013/12/31 下午 08:46
Readme.txt	1 KB	Text Docu...	2013/12/31 下午 08:50
TurboWriter.ini	1 KB	Configurat...	2013/12/6 下午 06:59
update.sh	2 KB	SH File	2013/12/8 下午 10:00

For example:

Please define the binary file name in NANDWRITER.ini file. Include [NandLoader File Name], [Execute File Name], [Execute Image Number] and [Execute Image Address] items.

[NandLoader File Name]: It indicates NAND loader firmware of N329x series.
[Execute File Name]: It indicates executable firmware of N329x series.
[Execute Image Number]: How many backup number of executable firmware you want?
[Execute Image Address]: The Executable firmware will put in system memory. You can define the number. It is usually zero.

```
[NandLoader File Name]
// All file name length MUST <= 511 bytes
NANDLoader.bin

[Execute File Name]
conprog.bin

[Execute Image Number]
// Backup count for Executing Image
1

[Execute Image Address]
// Execute address with Hex format for Executing Image
0
```

Finally, please run 'update.sh' to start.

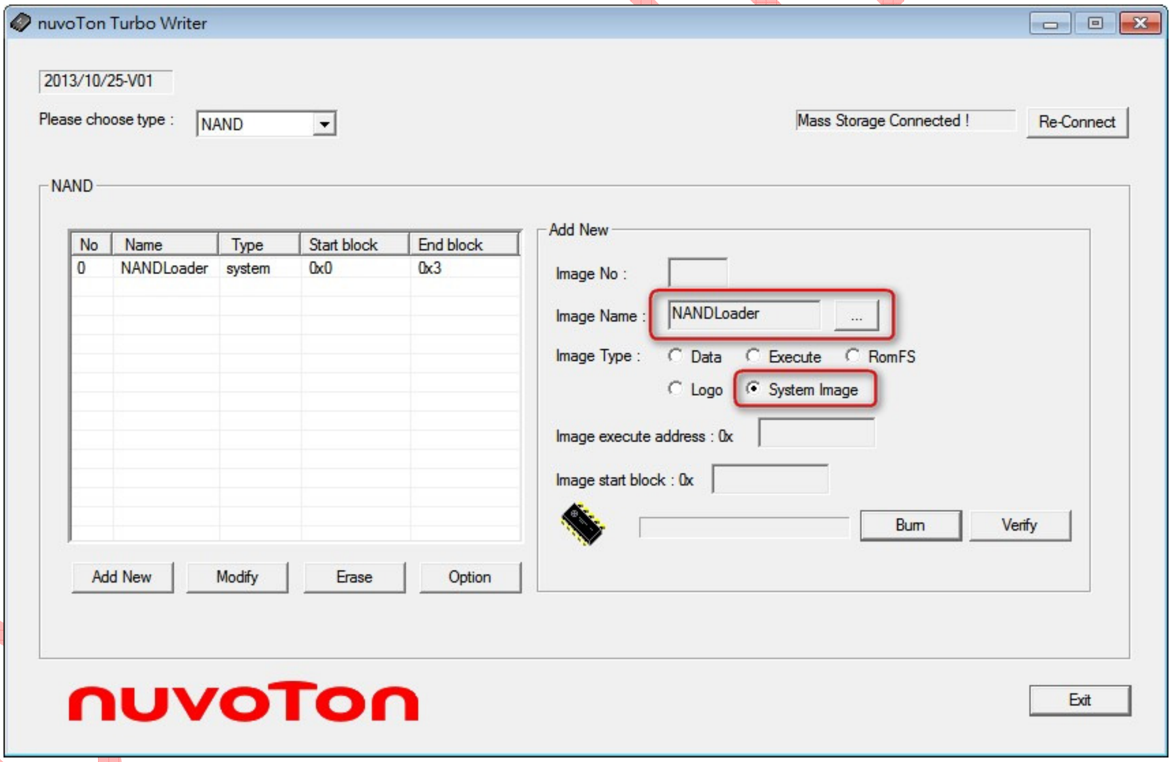
7. Example for MTD Solution

This chapter descripts the ways to program MTD into NAND flash chip step by step.

7.1. Turbowriter

You can program Nand Loader and Linux kernel into all new NAND flash by Turbowriter.

Step 1: Program Nand Loader into NAND flash as “System Image”.



nuvoTon Turbo Writer

2013/10/25-V01

Please choose type : NAND

Mass Storage Connected !

NAND

No	Name	Type	Start block	End block
0	NANDLoader	system	0x0	0x3

Add New

Image No :

Image Name : NANDLoader

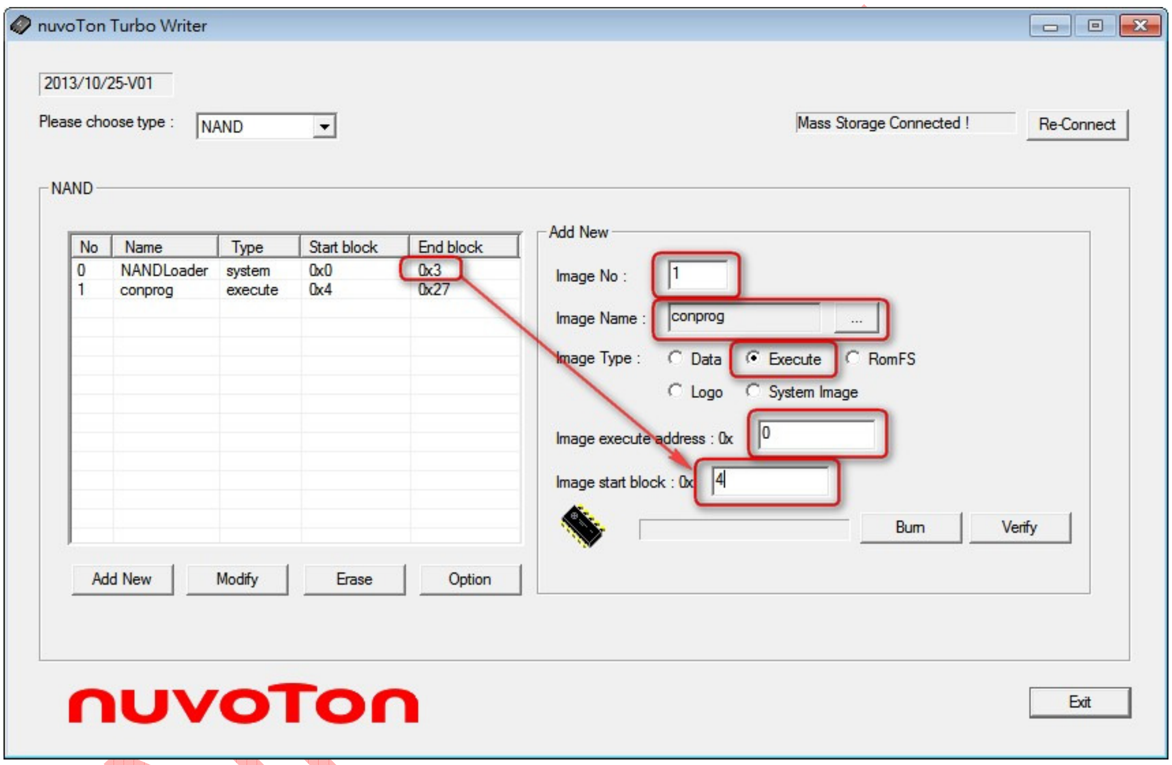
Image Type : ( ) Data ( ) Execute ( ) RomFS ( ) Logo [✓ System Image]

Image execute address : 0x

Image start block : 0x

Step 2: Program Linux Kernel into NAND flash as “Execute”. Please note that the “Image Execute Address” must be **0x0**.

After step 1 and step 2, you can reboot to run Linux kernel that support MTD device.


nuvoTon Turbo Writer

2013/10/25-V01

Please choose type : NAND

Mass Storage Connected !

NAND

No	Name	Type	Start block	End block
0	NANDLoader	system	0x0	0x3
1	conprog	execute	0x4	0x27

Add New

Image No : 1

Image Name : conprog

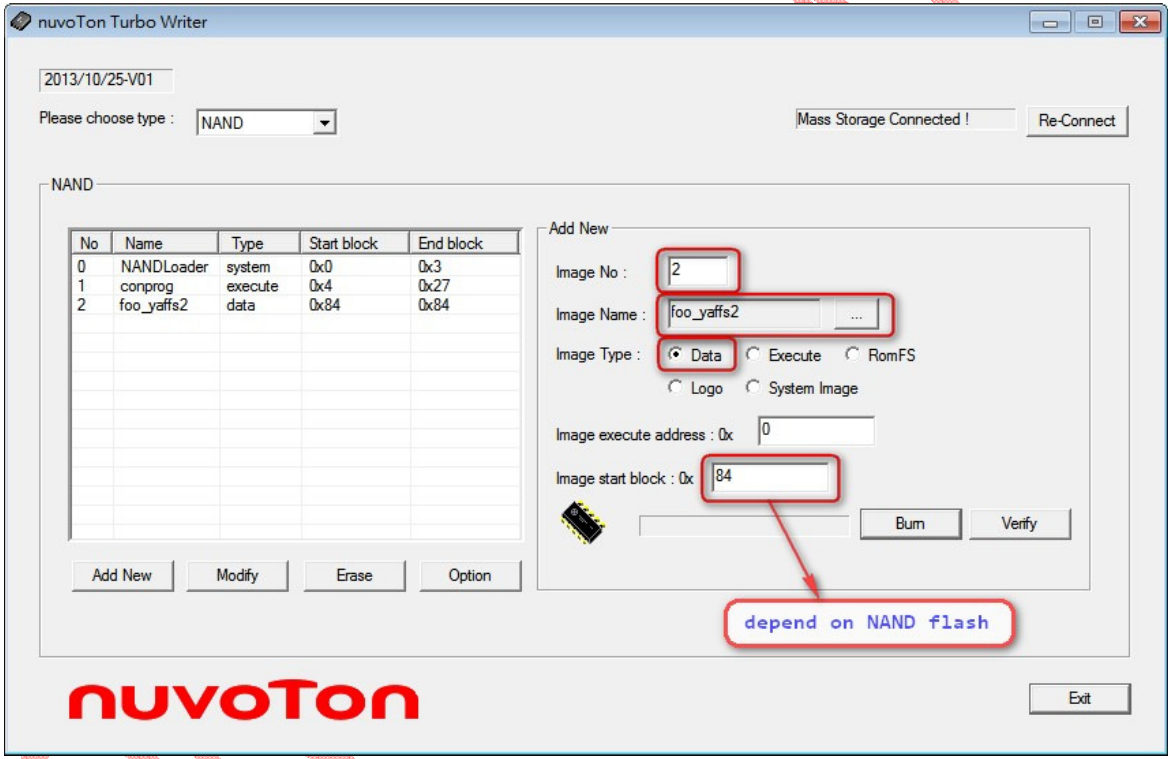
Image Type : ( ) Data [✓ Execute] ( ) RomFS ( ) Logo ( ) System Image

Image execute address : 0x 0

Image start block : 0x 4

v1.02

Step 3: (Optional) Program YAFFS2 file system image into NAND flash as “Data”. Please note that the “Image Start Block” must be **the start address of MTD partition NAND1-1 or NAND1-2**. For example, if we use the NAND flash with block size *128KB*, the block number of MTD partition SYSTEM is always 4, the size of MTD partition EXECUTE is always 16MB and occupy 16MB / block size = 128 blocks, so the “Image Start Block” is 4 + 128 = 132 = *0x84*.



nuvoTon Turbo Writer

2013/10/25-V01

Please choose type : NAND

Mass Storage Connected !

NAND

No	Name	Type	Start block	End block
0	NANDLoader	system	0x0	0x3
1	conprog	execute	0x4	0x27
2	foo_yaffs2	data	0x84	0x84

Add New

Image No : 2

Image Name : foo_yaffs2

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

Image execute address : 0x 0

Image start block : 0x 84

depend on NAND flash

Step 4: (Optional) if you program file system image into NAND flash by Turbowriter, you need to mount it to Linux file system before use it. Try following command to mount YAFFS2 file system on *MTD partition NAND1-1*:

```
mount -t yaffs2 -o"inband-tags" /dev/mtdblock2 /mnt/nand1-1
```

If you want to mount YAFFS2 file system on *MTD partition NAND1-2*, try

```
mount -t yaffs2 -o"inband-tags" /dev/mtdblock3 /mnt/nand1-2
```

7.2. Autowriter

You also can program Nand Loader, Linux kernel, and file system image into all new NAND flash by Autowriter.

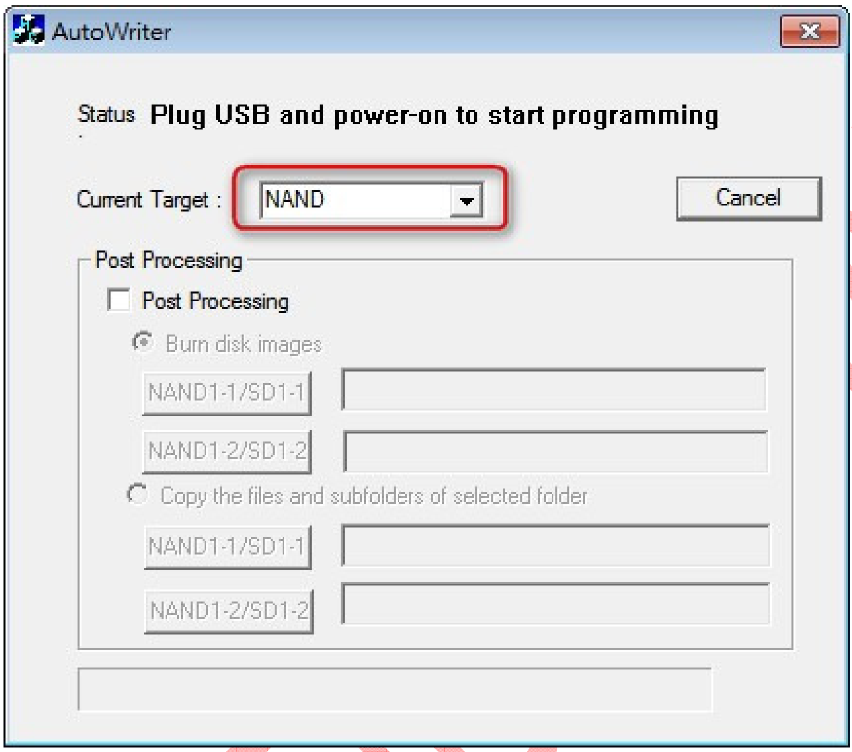
Step 1: Setup the AutoWriter.ini. Please refer to the following example.

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

[Execute File Name]
APP = mtdnand_yaffs2_conprog.bin
ADDRESS = 0

[Options]
EraseAll=y

[ExecuteImage]
No=2
APP1 = mtdnand_rtl8188_nt99141_dev_nand1-1_yaffs2.img
APP2 = mtdnand_rtl8188_nt99141_dev_nand1-2_yaffs2.img
Start Block1 = 84
Start Block2 = 184
```

The [Execute File Name] is the Linux kernel that support MTD device. It will be program into NAND flash and run after reboot.

The [ExecuteImage] is to burn file system images into the specific block of NAND flash. The key words "Start Block1" and "Start Block2" must be **the start address of MTD partition NAND1-1 or NAND1-2**. It is the hexadecimal number.

Step 2: Run Autowriter tool to auto program NAND flash according to the setting of AutoWriter.ini.



7.3. fs_writer

If you never program any file system image into NAND flash by Turbowriter or Autowriter, you can do it by fs_writer utility under Linux kernel.

Step 1: Prepare file system image file and fs_writer utility that include shell script and executable file.

Step 2: Run shell script of fs_writer. Refer to chapter 6.1 for detail description.

8. Revision history

Version	Date	Chang Log
v1.00	Jan. 2, 2014	• Create
v1.01	Jan. 15, 2014	• Modified MTD layout of example. • Append YAFFS2 description in section 6.1.
V1.02	Aug. 29, 2014	• Add example to program MTD to NAND flash. • Modify kernel configure about MTD and VPOST initial.

Important Note