

How to build a Production ROM including a JFFS2 Image Application Note V1.0

Publication Release Date: Sep. 2008

Support Chips:

NUC710A

NUC745A

Support Platforms:

uClinux



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.

This documentation may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from the Nuvoton Technology Corp.

Nuvoton Technology Corp. All rights reserved.

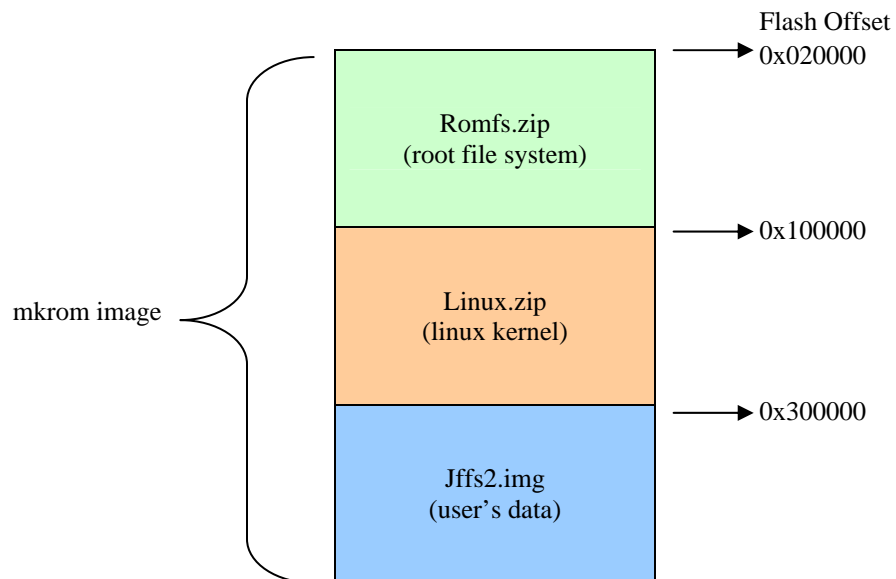
Table of Contents

1. Introduction	4
2. Solution.....	5
2.1. Produce a JFFS2 image on Linux PC	5
2.2. Get the bootloader.bin from bootloader SDK.....	5
2.3. Produce production ROM image	5
2.4. Configure Kernel	10
2.5. JFFS2 partition address.....	10
2.6. Mount the partition on the target board	11
3. Revision History	12

1. Introduction

For production purpose, user can use “mkrom” which is a utility program to generate a ROM image, and then write it to NOR Flash. Usually, the ROM image will include the rootfs and kernel. But sometimes user may also want to add another image most is a data image within that ROM image. After system up, the uClinux system can mount that image and use it by JFFS2 file system.

Suppose that the size of romfs.img is 512KB, linux.bin is 2MB, and the structure of production ROM includes jffs2 image. The ROM content should like as following figure.



User can follow the steps described in following sections to generate a JFFS2 image combined with production ROM produced by “mkrom” utility.

2. Solution

2.1. Produce a JFFS2 image on Linux PC

The first step is to produce a JFFS2 image used by uClinux system.

1. Install “mtd-tools” on the host PC.

```
> sudo apt-get install mtd-tools
```

2. Than produce JFFS2 image

```
> mkfs.jffs2 -d [DIR] -o jffs2.img [DIR] = directory path
```

3. A jffs2 image named jffs2.img will be built.

2.2. Get the bootloader.bin from bootloader SDK

1. Extract the bootloader.zip in the “Bootloader” directory of BSP CD.
2. The bootloader.bin file is at the \bootloader\WBLv1_1\image\bootloader\little directory.

2.3. Produce production ROM image

1. Put “bootloader.bin”, “linux.zip”, “romfs.zip”, “mkrom.exe”, “mkrom.ini”, “jffs2.img” together(or all in the same directory).
2. The mkrom.ini should like this.

```
# Configuration file for MKROM
#
#Introduction:
# The MKROM is a tool to generate a ROM image. The ROM image includes boot loader,
#Image 0 (boot information block), and user images.
#
#SYNOPSIS
# The MKROM generates a ROM image according to a configuration file. The default
#configuration file name is rom.ini. It should be placed in the same directory
as MKROM.exe.
```

```

#The configuration file also could be given by argument,i.e., MKROM
user_config.ini.

#The format of the configuration must be followed and it is case sensitive. It
includes four section types. These types
#are "[Target Image]", "[Boot Image]", "[Boot Info]", and "[Image N]". Where N
could be 1~7,i.e.,
#[Image 1], [Image 2], etc.
#
#NOTE: The sequence of each section is fixed. That is [Target Image] section first,
#[Boot Image] second, [Boot Info] third, then [Image N]. The sequence of each
"[Image N]"
#must according to its base address in the final ROM image. The [Image N] section
with lower
#base address must be placed in front of the [Image N] section with higher base
address.
#
# 1. [Target Image] section
#   Image="The file name of the target ROM image"
#
# 2. [Boot Image] section
#   Image="The file name of the boot loader image"
#
# 3. [Boot Info] section
#   mac="MAC address"
#   ip="IP address"
#   dhcp="DHCP client setting. Setting 1 to enable DHCP client, 0 to disable DHPC
client"
#   cache="Cache setting. Setting 1 to enable cache, 0 to disable cache"
#   buffer_base="The base address of the buffer used by boot loader"
#   buffer_size="The size of the buffer used by boot loader"
#   baudrate="baud rate of UART:
115200/57600/38400/28800/19200/14400/9600/4800/2400/1200"
#   serial_number="Serial number for the product"
#
# 4. [Image N] section (The value of N could be 1~7)
#   Image="The file name of the image"
#   Name="The alias name of the image"
#   Base="The base address of this image in the NUC710A system"
#   Load="The load address of this image in the NUC710A system"
#   Active="The IMAGE_ACTIVE attribute. Setting 1 to enable this attribute, 0
to disable it"

```

```
# Copy to RAM="The IMAGE_COPY2ROM attribute. Setting 1 to enable this
attribute, 0 to disable it"
# Execution="The IMAGE_EXEC attribute. Setting 1 to enable this attribute, 0
to disable it"
# Compressed="The IMAGE_COMPRESSED attribute. Setting 1 to enable this
attribute, 0 to disable it"
# File="The IMAGE_FILE attribute. Setting 1 to enable this attribute, 0 to
disable it"
#
#

[Target Image]
Image=rom.bin

[Boot Image]
Image=bootloader.bin

[Boot Info]
mac=00:00:00:00:00:01
ip=0.0.0.0
dhcp=1
cache=1
buffer_base=0x300000
buffer_size=0x100000
baudrate=115200
serial_number=12345678
usb=1

[Image 7]
Image=linux.zip
Name=linux
Base=0x7F020000
Load=0x8000
Active=1
Copy to RAM=1
Execution=1
Compressed=1
File=0
No Footer=0
```

```
[Image 6]
Image=romfs.zip
Name=romfs
Base=0x7F100000
Load=0xd00000
Active=1
Copy to RAM=1
Execution=0
Compressed=1
File=0
No Footer=0
```

```
[Image 5]
Image=jffs2.img
Name=jffs2
Base=0x7F300000
Load=0x7F300000
Active=0
Copy to RAM=0
Execution=0
Compressed=0
File=1
No Footer=1
```

3. Execute "mkrom.exe" in DOS environment to generate rom.bin.

```
NUC710A MKROM Tool v1.0 Rebuilt on Jul 25 2008 at 14:02:39

Found target image section
option=Image value=rom.bin
Found image section
option=Image value=bootloader.bin
Found boot information section
option=mac value=00:00:00:00:00:01
option=ip value=0.0.0.0
option=dhcp value=1
option=cache value=1
option=buffer_base value=0x300000
option=buffer_size value=0x100000
```



```
option=baudrate value=115200
option=serial_number value=12345678
option=usb value=1
Boot Loader Configuration:

    MAC Address      : 00:00:00:00:00:01
    IP Address       : 0.0.0.0
    DHCP Client      : Enabled
    CACHE            : Enabled
    BL buffer base   : 0x00300000
    BL buffer size   : 0x00100000
    Baud Rate        : 115200
    USB               : Disabled
    Serial Number    : 12345678

Found image 7 section
option=Image value=linux.zip
option=Name value=linux
option=Base value=0x7F020000
option=Load value=0x8000
option=Active value=1
option=Copy to RAM value=1
option=Execution value=1
option=Compressed value=1
option=File value=0
option=No Footer value=0
Found image 6 section
option=Image value=romfs.zip
option=Name value=romfs
option=Base value=0x7F100000
option=Load value=0xd00000
option=Active value=1
option=Copy to RAM value=1
option=Execution value=0
option=Compressed value=1
option=File value=0
option=No Footer value=0
Found image 5 section
option=Image value=jffs2.img
```

```
option=Name value=jffs2
option=Base value=0x7F300000
option=Load value=0x7F300000
option=Active value=0
option=Copy to RAM value=0
option=Execution value=0
option=Compressed value=0
option=File value=1
option=No Footer value=1

Image rom.bin creation success!!
```

4. Burn "rom.bin" into your target board.

2.4. Configure Kernel

1. Enable JFFS2 function in kernel.

```
Main Menu --->
  File systems --->
    [*] Journalling Flash File System v2 (JFFS2) support
    [*] JFFS2 ZLIB compression support (recommended)
```

2. Make the kernel.

2.5. JFFS2 partition address

User produces the "rom.bin" contains the JFFS2 image at offset address (0x300000) of flash at previous step. And now, user should check the JFFS2 configuration in the driver.

The uClinux system will create two JFFS2 (MTD) partitions at default.

```
Creating 2 MTD partitions on "POS-TAX flash device":
0x00000000-0x00300000 : "images 3M"
0x00300000-0x00400000 : "user 1M"
```

The address 0x00000000-0x00300000 reserves for images (linux.zip and romfs.zip) and address 0x00300000 - 0x00400000 is used for application.

The jffs2.img will be put at address 0x7f300000 (offset 0x300000) of flash, it should be the same with uClinux NOR flash user's partition.

User can modify this user's partition address or add another partition by modifying [\NUC710A-uClinux\uClinux-dist\linux-2.4.x\drivers\mtd\maps\NUC7xx_map.c](#)

```
static struct mtd_partition NUC745_partitions[] = {
{
    name: "images 3M",
    size: 0x300000,
    offset: 0
},
{
    /*if jffs2 can run on this partition the size can not less than 6 sectors*/
    name: "user 1M",
    size: 0x100000,
    offset: 0x00300000 //offset
}
}
```

If have any modification on this driver, please remember to rebuild the kernel.

2.6. Mount the partition on the target board

Finally, user can mount the user's partition

```
mount -t jffs2 /dev/mtdblock1 /flash
```

3. Revision History

Version	Date	Description
V1.0	Sept. 2008	• Created

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur. Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

*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.*
