Flash and load UBIFS image on iMX35 NAND

Quick Steps

Four quick steps to build and flash a UBIFS image on iMX35 NAND (K9LBG08U0D-PCB0), for information on how to you another memory, please see next secction.

1. Enable MTD_UBI and UBIFS_FS on kernel
2. Create UBI image from rootfs (used for NFS) - ON PC
    1. mkfs.ubifs -v -r rootfs -m 2048 -e 258048 -c 966 -o ubifs.img
    2. ubinize -o ubi.img -m 2048 -p 256KiB -s 2048 ubinize.cfg
3. Format NAND using UBI image - ON TARGET
    1. ubiformat -f ubi.img /dev/mtd8
4. Load UBI file system
    1. load -r -b 0x100000 zImage*
    2. fis create -f 0x300000 kernel
       ##fis load kernel
       ##exec -c "noinitrd console=ttymxc0 115200 ubi.mtd=8 root=ubi0:rootfs rw rootfstype=ubifs ip=none"

How To

First of all, install mtd-utils on both target and host:

Target:

./ltib -c

Package list

[*] mtd-utils

Host

sudo aptget install mtd-utils

1 Enable MTD_UBI and UBIFS_FS on kernel

MTD_UBI

```
-> Device Drivers

   -> Memory Technology Device (MTD) support (MTD =y)

       -> UBI - Unsorted block images

                 <*> Enable UBI

                 (4096) UBI wear-leveling threshold (NEW)
                 (1) Percentage of reserved eraseblocks for bad eraseblocks
handling (NEW)
               < > MTD devices emulation driver (gluebi) (NEW)
               ** UBI debugging options **
               [ ] UBI debugging (NEW)
```

UBIFS_FS

```
-> File systems

   ->Miscellaneous filesystems

        <*> UBIFS file system support
        [ ] Extended attributes support (NEW)
        [ ] Advanced compression options (NEW)
        [ ] Enable debugging (NEW)
```

2 Create UBI image

On TARGET Collect some information needed in order to create the UBI image according to your NAND device.

```
root@freescale ~$ cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00080000 00020000 "Bootloader"
mtd1: 00400000 00020000 "nor.Kernel"
mtd2: 01e00000 00020000 "nor.userfs"
mtd3: 01c00000 00020000 "nor.rootfs"
```

mtd4: 00003000 00020000 "FIS directory"
mtd5: 02001000 00020000 "Redboot config"
mtd6: 00300000 00040000 "nand.bootloader"
mtd7: 00500000 00040000 "nand.kernel"
mtd8: 10000000 00040000 "nand.rootfs"
mtd9: 00800000 00040000 "nand.configure"
mtd10: 6f000000 00040000 "nand.userfs"

I will use mtd8, because I want the NAND rootfs MTD partition.

(from http://www.linux-mtd.infradead.org/faq/ubifs.html#L_mkfubifs)

root@freescale ~$ ubiattach /dev/ubi_ctrl -m 8
UBI: attaching mtd8 to ubi0
UBI: physical eraseblock size: 262144 bytes (256 KiB)
UBI: logical eraseblock size: 258048 bytes
UBI: smallest flash I/O unit: 2048
UBI: VID header offset: 2048 (aligned 2048)
UBI: data offset: 4096
UBI: empty MTD device detected
UBI: create volume table (copy #1)
UBI: create volume table (copy #2)
UBI: attached mtd8 to ubi0
UBI: MTD device name: "nand.rootfs"
UBI: MTD device size: 256 MiB
UBI: number of good PEBs: 979
UBI: number of bad PEBs: 45
UBI: max. allowed volumes: 128
UBI: wear-leveling threshold: 4096
UBI: number of internal volumes: 1
UBI: number of user volumes: 0
UBI: available PEBs: 966
UBI: total number of reserved PEBs: 13
UBI: number of PEBs reserved for bad PEB handling: 9
UBI: max/mean erase counter: 0/0
UBI: image sequence number: 0
UBI: background thread "ubi_bgt0d" started, PID 2098
UBI device number

You will need:

- -p = physical eraseblock size = 256KiB
- -e = logical eraseblock size = 258048
- -m = smallest flash I/O unit = 2048

- -s = VID header offset = 2048 (some flash will also have a sub-page size reported when you run ubiattach, which is what you should use with –s)
- -c = available PEB = 966

(Values for iMX35 PDK NAND - K9LBG08U0D-PCB0)

ON HOST - Now, create the images (two steps)

```
$ mkfs.ubifs -v -r rootfs -m 2048 -e 258048 -c 966 -o ubifs.img
mkfs.ubifs
    root:        rootfs/
    min_io_size:  2048
    leb_size:     258048
    max_leb_cnt:  966
    output:       ubifs.img
    jrn_size:     8388608
    reserved:     0
    compr:        lzo
    keyhash:      r5
    fanout:       8
    orph_lebs:    1
    super lebs:   1
    master lebs:  2
    log_lebs:     4
    lpt_lebs:     2
    orph_lebs:    1
    main_lebs:    132
    gc lebs:      1
    index lebs:   2
    leb_cnt:      142
    UUID:         CC2057F9-B20F-46D1-A399-1FCA95DCAFF7
Success!
$ ubinize -o ubi.img -m 2048 -p 256KiB -s 2048 ubinize.cfg
$ ls -lh u*
-rw-r--r- 1 daiane daiane 35M 2010-11-26 15:21 ubifs.img
-rw-r--r- 1 daiane daiane 36M 2010-11-26 15:22 ubi.img
-rw-r--r- 1 daiane daiane 113 2010-11-26 15:22 ubinize.cfg
$ sudo cp ubi.img rootfs/home/
```

You need to create ubinize.cfg file!

4 Format NAND using UBI image - ON TARGET

Turn on target (or reset it) and format MTD partition

```
$ cd /home

$ ubiformat -f ubi.img /dev/mtd8
```

5 Load UBI file system

Reset and change redboot script:

```
fis load kernel

exec -c "noinitrd console=ttymxc0 115200 ubi.mtd=8 root=ubi0:rootfs rw
rootfstype=ubifs ip=none"
```