Build and Run UBIFS on TCI6638 EVM

1. Environment preparation

This document based on the following environment:

- MCSDK version: 03_00_02_14 (can get from http://software-dl.ti.com/sdoemb/sdoemb_public_sw/mcsdk/03_00_02_14/index_FDS.html)
- Linux source code version: K2_LINUX_03.08.04_13.09_01 (can get from http://arago-project.org/git/projects/?p=linux-keystone.git;a=summary)
- U-Boot source code version: K2_UBOOT_2013-01_13.09_01 (can get from http://arago-project.org/git/projects/?p=u-boot-keystone.git;a=summary)
- Boot monitor source code version: K2_BM_13.08 (can get from http://arago-project.org/git/projects/?p=boot-monitor.git;a=summary)

The typographical conventions of this document are:

Character shading	Denote content in a source code
Character with border and shading	Denote command or code inputted by users

1.1 U-Boot support for UBIFS

(1) Command definition for UBI

In the directory of U-Boot source code, related command definition for UBI can be found in the /include/configs/tci6638_evm.h as below.

```
……
#define CONFIG_CMD_NAND
#define CONFIG_CMD_UBI
#define CONFIG_CMD_UBIFS
#define CONFIG_CMD_NAND_ECCLAYOUT
……
```

(2) Configuration for NAND partition

The configuration for NAND partition can be found in /include/configs/tci6638_evm.h as below, we can also change it in the U-Boot environment variables when the autoboot is stopped.

```
……
"mtdparts=mtdparts=davinci_nand.0:"                    \
```

```
        "1024k(bootloader)ro,512k(params)ro,522752k(ubifs)\0"
……
```

Note: 522752k, i.e. 510.5MiB, this parameter will be used in calculate c parameter for mkfs.ubifs, please see Appendix I for details.

1.2 Linux kernel configuration

Run make menuconfig command in the terminal of root directory for Linux kernel source code, to enable UBIFS support (Pressing <Y> includes, <N> excludes, <M> modularizes features)

(1) Enable UBI Support:

a. Select ***Device Drivers*** from the main menu:

```
x     [*] Networking support  --->
x         Device Drivers  --->
x         Firmware Drivers  --->
x         File systems  --->
```

b. Select ***Memory Technology Device (MTD)*** Supportas shown here:

```
x         Bus devices  --->
x     < > Connector - unified userspace <-> kernelspace linker  --->
x     <*> Memory Technology Device (MTD) support  --->
x     < > Parallel port support  --->
```

c. Select ***Enable UBI - Unsorted block images*** as shown here:

```
x       <M>   OneNAND Device Support  --->
x             LPDDR flash memory drivers  --->
x       <*>   Enable UBI - Unsorted block images  --->
```

(2) Select NAND Device Support:

a. Select ***Device Drivers*** and ***Memory Technology Device (MTD) Support***

b. Select ***NAND Device Support*** as shown here:

```
x       [ ]   NAND ECC Smart Media byte order
x       <*>   NAND Device Support  --->
x       <M>   OneNAND Device Support  --->
x             LPDDR flash memory drivers  --->
```

(3) Select UBIFS file system support

a. Select ***File Systems*** from the main menu

```
x          Device Drivers  --->
x          File systems  --->
x          Kernel hacking  --->
```

b. Select ***Miscellaneous filesystems*** as shown here:

```
x          Pseudo filesystems  --->
x        [*] Miscellaneous filesystems  --->
x        [*] Network File Systems  --->
```

c. Select ***UBIFS file system support*** as shown here:

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

Exit and save configuration, then run `make keystone2_defconfig` command, `make uImage` command, and `make k2hk-evm.dtb` command

1.3 UBIFS tools

There are two utilities [1]:

- ***mkfs.ubifs*** which creates UBIFS images
- ***ubinize*** which creates UBI images

UBI stands for "Unsorted Block Images". It is a volume management system for raw flash devices which manages multiple logical volumes on a single physical flash device, for more information, please check reference [2].

UBIFS is a new flash file system developed by Nokia engineers with help of the University of Szeged. In a way, UBIFS may be considered as the next generation of the JFFS2 file-system, for more information, please check reference [3].

Detailed method to build these two tools, please go to Appendix II for reference.

2. Creating UBIFS and UBI images

If you need to create an image which should be flashed to the raw flash, you should first create related UBIFS images, then create UBI image. In other words, the process has 2 steps. For example, in this document, according to the content of ubinize.cfg file, boot and rootfs ubifs images will be created firstly, then ubi image that includes rootfs and boot (kernel, dtb, and boot monitor) ubifs image will be created [4].

2.1 Preparation

(1) The work directory for testing is ~/tool/ubifs/kii

(2) Using ti-rootfs.tar.gz from mcsdk_linux_3_00_02_14\images directory as rootfs source file, untar it to rootfs folder, shown as below. Users can add new folder or application files in the rootfs folder with their requirement.

```
damon@cifae:~/tool/ubifs/kii/rootfs$ ls
tisdk-rootfs.tar.gz
damon@cifae:~/tool/ubifs/kii/rootfs$ sudo tar zxf tisdk-rootfs.tar.gz
damon@cifae:~/tool/ubifs/kii/rootfs$ sudo rm tisdk-rootfs.tar.gz
damon@cifae:~/tool/ubifs/kii/rootfs$ ls
bin   dev  home  media  proc  srv  tmp  var
boot  etc  lib   mnt    sbin  sys  usr  www
damon@cifae:~/tool/ubifs/kii/rootfs$
```

(3) Copy *mkfs.ubifs*, *ubinize* created by above method to the work directory, and copy ubinize.cfg file from mcsdk_linux_3_00_02_14\images directory

(4) Copy uImage, k2hk-evm.dtb and skern.bin (create by run make command in the root directory of boot monitor source code) to the boot-image sub-folder in the work directory for next step boot ubifs image create. Then, the work directory should look like as below.

```
damon@cifae:~/tool/ubifs/kii$ ls
boot-image  mkfs.ubifs  rootfs  ubinize  ubinize.cfg
damon@cifae:~/tool/ubifs/kii$ cd boot-image/
damon@cifae:~/tool/ubifs/kii/boot-image$ ls
k2hk-evm.dtb  skern.bin  uImage
damon@cifae:~/tool/ubifs/kii/boot-image$
```

Note: The name of files in boot-image directory must be the same with related setting in the U-Boot parameter (k2hk-evm.dtb must be the same with **name_fdt** parameter, uImage must be the same with **name_kern** parameter, skern.bin must be the same with **name_mon** parameter), users can either choose to change file name here to be the same with setting in the U-Boot parameter, or change U-Boot parameter setting

(5) And if we don't need recoveryfs, we can delete below definition from ubinize.cfg file.

```
[rootfs-recovery]
mode=ubi
image=arago-console-image-keystone-evm.ubifs
vol_id=1
vol_type=dynamic
vol_size=15MiB
vol_name=rootfs-recovery
```

(6) And modify the ubinize.cfg file for using as below:

```
[boot]
```

```
mode=ubi
image=keystone-evm-boot.ubifs
vol_type=dynamic
vol_id=0
vol_name=boot
vol_size=10MiB
[rootfs]
mode=ubi
image=tisdk-rootfs.ubifs
vol_id=1
vol_type=dynamic
vol_name=rootfs
vol_flags=autoresize
```

2.2 Create keystone-evm-boot.ubifs file

In order to create keystone-evm-boot.ubifs (this name was assigned in the ubinize.cfg) file, run this command in the work directory:

```
sudo ./mkfs.ubifs -r boot-image/ -F -o keystone-evm-boot.ubifs -m 2048 -e 126976 -c 75
```

```
damon@cifae:~/tool/ubifs/kii$ sudo ./mkfs.ubifs -r boot-image/ -F -o keystone-ev
m-boot.ubifs -m 2048 -e 126976 -c 75
[sudo] password for damon:
damon@cifae:~/tool/ubifs/kii$ ls
boot-image  keystone-evm-boot.ubifs  mkfs.ubifs  rootfs  ubinize  ubinize.cfg
damon@cifae:~/tool/ubifs/kii$
```

2.3 Create tisdk-rootfs.ubifs file

```
sudo ./mkfs.ubifs -r rootfs/ -F -o tisdk-rootfs.ubifs -m 2048 -e 126976 -c 4039
```

Where:

-m 2KiB (or 2048)

The minimum I/O size of the underlying UBI and MTD devices. In our case, we are running the flash with no sub-page writes, so this is a 2KiB page.

-e 124KiB (or 126976)

Erase Block Size: UBI requires 2 minimum I/O units out of each Physical Erase Block (PEB) for overhead: 1 for maintaining erase count information, and 1 for maintaining the Volume ID information. The PEB size for the Micron flash is 128KiB, so this leads to each Logical Erase Block (LEB) having 124KiB available for data.

-c 4039

The maximum size of the file system (in LEBs). See Appendix I for calculation explanation

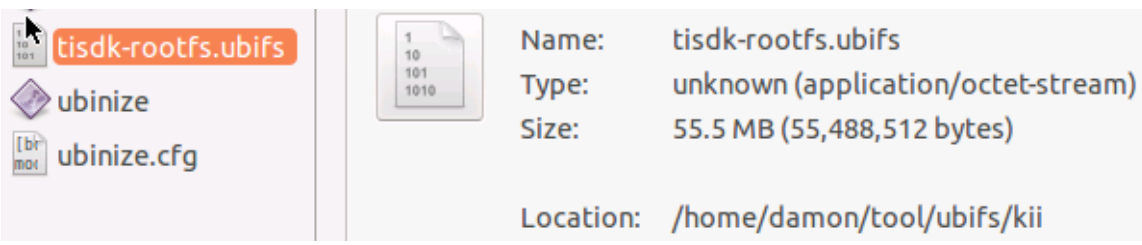
-r rootfs

Use the contents of the 'rootfs/' directory to generate the initial file system image.

-o tisdk-rootfs.ubifs

Output file

```
damon@cifae:~/tool/ubifs/kii$ sudo ./mkfs.ubifs -r rootfs/ -F -o tisdk-rootfs.ub
ifs -m 2048 -e 126976 -c 4039
[sudo] password for damon:
damon@cifae:~/tool/ubifs/kii$ ls
boot-image                mkfs.ubifs  tisdk-rootfs.ubifs  ubinize.cfg
keystone-evm-boot.ubifs   rootfs       ubinize
damon@cifae:~/tool/ubifs/kii$
```



2.4 Create tci6638-evm-ubifs.ubi file

```
sudo ./ubinize -o tci6638-evm-ubifs.ubi -m 2048 -p 128KiB -s 2048 -O 2048 ubinize.cfg
```

Where:

-o tci6638-evm-ubifs.ubi

Output file

-m 2KiB (or 2048)

Minimum flash I/O size of 2KiB page

-p 128KiB

Size of the physical erase block of the flash this UBI image is created for

-s 2048

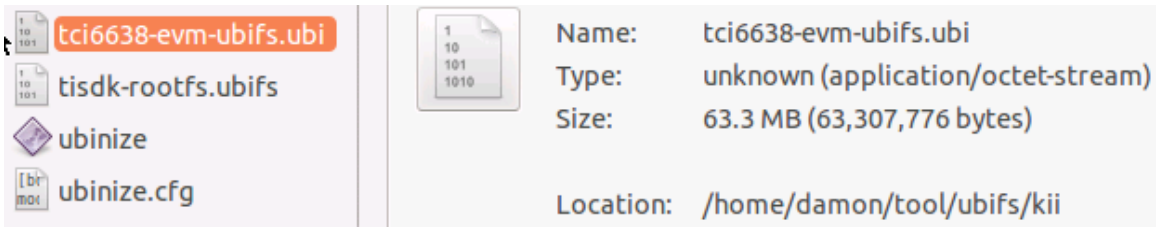
Use a subpage size of 2048

-O 2048

Offset if the VID header from start of the physical erase block

The output of the above command, ' **tci6638-evm-ubifs.ubi** ' is the required UBI image.

```
damon@cifae:~/tool/ubifs/kii$ sudo ./ubinize -o tci6638-evm-ubifs.ubi -m 2048 -p
 128KiB -s 2048 -O 2048 ubinize.cfg
ubinize: volume size was not specified in section "rootfs", assume minimum to fi
t image "tisdk-rootfs.ubifs"55488512 bytes (52.9 MiB)
damon@cifae:~/tool/ubifs/kii$ ls
boot-image                mkfs.ubifs  tci6638-evm-ubifs.ubi  ubinize
keystone-evm-boot.ubifs   rootfs       tisdk-rootfs.ubifs     ubinize.cfg
damon@cifae:~/tool/ubifs/kii$
```



Then we can copy the tci6638-evm-ubifs.ubi file to tftp root path, prepare for flashing to the NAND.

3. Flash UBI image to NAND

3.1 Preparation

Power on TCI6638 EVM, stop autoboot, resetting to default environment and set related boot parameter for UBI boot (here we can refer to document [5] for details).

```
TCI6638 EVM # env default -f -a
## Resetting to default environment
TCI6638 EVM # saveenv
Saving Environment to NAND...
Erasing Nand...
Erasing at 0x120000 -- 100% complete.
Writing to Nand... done
TCI6638 EVM # pri
addr_fdt=0x87000000
addr_fs=0x82000000
addr_kern=0x88000000
addr_mon=0x0c5f0000
addr_ubi=0x82000000
addr_uboot=0x87000000
args_all=setenv bootargs console=ttyS0,115200n8 rootwait=1
args_net=setenv bootargs ${bootargs} rootfstype=nfs root=/dev/nfs rw
nfsroot=${serverip}:${nfs_root},${nfs_options} ip=dhcp
args_ramfs=setenv bootargs ${bootargs} earlyprintk rdinit=/sbin/init rw root=/dev/ram0
initrd=0x802000000,9M
args_ubi=setenv bootargs ${bootargs} rootfstype=ubifs root=ubi0:rootfs rootflags=sync
rw ubi.mtd=2,2048
args_uinitrd=setenv bootargs ${bootargs} earlyprintk rdinit=/sbin/init rw
root=/dev/ram0
baudrate=115200
boot=ramfs
bootcmd=run init_${boot} get_fdt_${boot} get_mon_${boot} get_kern_${boot}
run_mon run_kern
bootdelay=3
bootfile=uImage
burn_ubi=nand erase.part ubifs; nand write ${addr_ubi} ubifs ${filesize}
burn_uboot=sf probe; sf erase 0 0x100000; sf write ${addr_uboot} 0 ${filesize}
fdt_high=0xffffffff
get_fdt_net=dhcp ${addr_fdt} ${tftp_root}/${name_fdt}
get_fdt_ramfs=dhcp ${addr_fdt} ${tftp_root}/${name_fdt}
get_fdt_ubi=ubifsload ${addr_fdt} ${name_fdt}
```

```
get_fdt_uinitrd=dhcp ${addr_fdt} ${tftp_root}/${name_fdt}
get_fs_ramfs=dhcp ${addr_fs} ${tftp_root}/${name_fs}
get_fs_uinitrd=dhcp ${addr_fs} ${tftp_root}/${name_uinitrd}
get_kern_net=dhcp ${addr_kern} ${tftp_root}/${name_kern}
get_kern_ramfs=dhcp ${addr_kern} ${tftp_root}/${name_kern}
get_kern_ubi=ubifsload ${addr_kern} ${name_kern}
get_kern_uinitrd=dhcp ${addr_kern} ${tftp_root}/${name_kern}
get_mon_net=dhcp ${addr_mon} ${tftp_root}/${name_mon}
get_mon_ramfs=dhcp ${addr_mon} ${tftp_root}/${name_mon}
get_mon_ubi=ubifsload ${addr_mon} ${name_mon}
get_mon_uinitrd=dhcp ${addr_mon} ${tftp_root}/${name_mon}
get_ubi_net=dhcp ${addr_ubi} ${tftp_root}/${name_ubi}
get_uboot_net=dhcp ${addr_uboot} ${tftp_root}/${name_uboot}
has_mdio=0
init_net=run set_fs_none args_all args_net
init_ramfs=run set_fs_none args_all args_ramfs get_fs_ramfs
init_ubi=run set_fs_none args_all args_ubi; ubi part ubifs; ubifsmount boot
init_uinitrd=run set_fs_uinitrd args_all args_uinitrd get_fs_uinitrd
initrd_high=0xffffffff
mem_lpae=1
mem_reserve=512M
mtdparts=mtdparts=davinci_nand.0:1024k(bootloader)ro,512k(params)ro,522752k(ubif
s)
name_fdt=uImage-k2hk-evm.dtb
name_fs=arago-console-image.cpio.gz
name_kern=uImage-keystone-evm.bin
name_mon=skern-keystone-evm.bin
name_ubi=keystone-evm-ubifs.ubi
name_uboot=u-boot-spi-keystone-evm.gph
name_uinitrd=uinitrd.bin
nfs_options=v3,tcp,rsize=4096,wsize=4096
nfs_root=/export
no_post=1
run_kern=bootm ${addr_kern} ${addr_uinitrd} ${addr_fdt}
run_mon=mon_install ${addr_mon}
set_fs_none=setenv addr_uinitrd -
set_fs_uinitrd=setenv addr_uinitrd ${addr_fs}
tftp_root=/

Environment size: 2793/262140 bytes

TCI6638 EVM # setenv boot 'ubi'
```

```
TCI6638 EVM # setenv serverip '157.87.43.15'
TCI6638 EVM # setenv tftp_root damon
TCI6638 EVM # setenv name_ubi tci6638-evm-ubifs.ubi
TCI6638 EVM # setenv size_ubi 0x3c60000
TCI6638 EVM # setenv erase_nand_ubi 'nand erase.part ubifs'
TCI6638 EVM # setenv write_nand_ubi 'nand write ${addr_fs} ubifs ${size_ubi}'
TCI6638 EVM # setenv get_tftp_ubi 'dhcp ${addr_fs} ${tftp_root}/${name_ubi}'
TCI6638 EVM # setenv name_fdt k2hk-evm.dtb
TCI6638 EVM # setenv name_mon skern.bin
TCI6638 EVM # setenv name_kern uImage
TCI6638 EVM # saveenv
Saving Environment to NAND...
Erasing Nand...
Erasing at 0x120000 -- 100% complete.
Writing to Nand... done
TCI6638 EVM #
```

3.2 Operation procedure

(1) Erase ubifs part in flash

```
run erase_nand_ubi
```

(2) Get tci6638-evm-ubifs.ubi file from tftp

```
run get_tftp_ubi
```

(3) Write tci6638-evm-ubifs.ubi file to flash

```
run write_nand_ubi
```

You can see the similar print information on screen for these three procedures:

```
TCI6638 EVM # run erase_nand_ubi

NAND erase.part: device 0 offset 0x180000, size 0x1fe80000
Skipping bad block at  0x1ff80000
Skipping bad block at  0x1ffa0000
Skipping bad block at  0x1ffc0000
Skipping bad block at  0x1ffe0000

OK
TCI6638 EVM # run get_tftp_ubi
BOOTP broadcast 1
DHCP client bound to address 157.87.43.197
Using TCI6638_EMAC device
TFTP from server 157.87.43.15; our IP address is 157.87.43.197
Filename 'damon/tci6638-evm-ubifs.ubi'.
Load address: 0x82000000
Loading: ##################################################################
         ##################################################################
         ##################################################################
```

……

```
         ##################################################################
         ##################################################################
         ###########################
         1.1 MiB/s
done
Bytes transferred = 63307776 (3c60000 hex)
TCI6638 EVM # run write_nand_ubi

NAND write: device 0 offset 0x180000, size 0x3c60000
 63307776 bytes written: OK
TCI6638 EVM #
```

(4) Boot from UBIFS

Boot

Then we can get below log information and finally boot successfully

Creating 1 MTD partitions on "nand0":
0x000000180000-0x000020000000 : "mtd=2"
UBI: attaching mtd1 to ubi0
UBI: physical eraseblock size: 131072 bytes (128 KiB)
UBI: logical eraseblock size: 126976 bytes
UBI: smallest flash I/O unit: 2048
UBI: VID header offset: 2048 (aligned 2048)
UBI: data offset: 4096
UBI: attached mtd1 to ubi0
UBI: MTD device name: "mtd=2"
UBI: MTD device size: 510 MiB

```
UBI: number of good PEBs:       4080
UBI: number of bad PEBs:        4
UBI: max. allowed volumes:      128
UBI: wear-leveling threshold:   4096
UBI: number of internal volumes: 1
UBI: number of user volumes:    2
UBI: available PEBs:            0
UBI: total number of reserved PEBs: 4080
UBI: number of PEBs reserved for bad PEB handling: 40
UBI: max/mean erase counter: 1/0
UBIFS: mounted UBI device 0, volume 0, name "boot"
UBIFS: mounted read-only
UBIFS: file system size:   8253440 bytes (8060 KiB, 7 MiB, 65 LEBs)
UBIFS: journal size:       1523712 bytes (1488 KiB, 1 MiB, 12 LEBs)
UBIFS: media format:       w4/r0 (latest is w4/r0)
UBIFS: default compressor: LZO
UBIFS: reserved for root:  0 bytes (0 KiB)
Loading file 'k2hk-evm.dtb' to addr 0x87000000 with size 44605 (0x0000ae3d)...
Done
Loading file 'skern.bin' to addr 0x0c5f0000 with size 45056 (0x0000b000)...
Done
Loading file 'uImage' to addr 0x88000000 with size 3926240 (0x003be8e0)...
Done
## installed monitor, freq [194560000], status 0
## Booting kernel from Legacy Image at 88000000 ...
   Image Name:   Linux-3.8.4
   Created:      2013-12-16  10:31:30 UTC
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    3926176 Bytes = 3.7 MiB
   Load Address: 80008000
   Entry Point:  80008000
   Verifying Checksum ... OK
## Flattened Device Tree blob at 87000000
   Booting using the fdt blob at 0x87000000
   Loading Kernel Image ... OK
OK
   Using Device Tree in place at 87000000, end 8700de3c

Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0
```

```
[    0.000000] Linux version 3.8.4 (damon@cifae) (gcc version 4.7.3 20130226 (prerelease)
(crosstool-NG linaro-1.13.1-4.7-2013.03-20130313 - Linaro GCC 2013.03) ) #1 SMP Mon Dec 16
15:31:17 EST 2013
[    0.000000] CPU: ARMv7 Processor [412fc0f4] revision 4 (ARMv7), cr=30c7387d
[    0.000000] CPU: PIPT / VIPT nonaliasing data cache, PIPT instruction cache
[    0.000000] Machine: KeyStone2, model: Texas Instruments Keystone 2 SoC
[    0.000000] switching to high address space at 0x800000000
[    0.000000] cma: CMA: reserved 16 MiB at 1f000000
[    0.000000] Memory policy: ECC disabled, Data cache writealloc
[    0.000000] PERCPU: Embedded 8 pages/cpu @c0bc6000 s11648 r8192 d12928 u32768
[    0.000000] Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 130048
[    0.000000] Kernel command line: console=ttyS0,115200n8 rootwait=1 rootfstype=ubifs
root=ubi0:rootfs rootflags=sync rw ubi.mtd=2,2048
[    0.000000] PID hash table entries: 2048 (order: 1, 8192 bytes)
[    0.000000] Dentry cache hash table entries: 65536 (order: 6, 262144 bytes)
[    0.000000] Inode-cache hash table entries: 32768 (order: 5, 131072 bytes)
[    0.000000] __ex_table already sorted, skipping sort
[    0.000000] Memory: 512MB = 512MB total
[    0.000000] Memory: 495228k/495228k available, 29060k reserved, 0K highmem
[    0.000000] Virtual kernel memory layout:
[    0.000000]     vector  : 0xffff0000 - 0xffff1000   (   4 kB)
[    0.000000]     fixmap  : 0xfff00000 - 0xfffe0000   ( 896 kB)
[    0.000000]     vmalloc : 0xe0800000 - 0xff000000   ( 488 MB)
[    0.000000]     lowmem  : 0xc0000000 - 0xe0000000   ( 512 MB)
[    0.000000]     pkmap   : 0xbfe00000 - 0xc0000000   (   2 MB)
[    0.000000]     modules : 0xbf000000 - 0xbfe00000   (  14 MB)
[    0.000000]       .text : 0xc0008000 - 0xc06f943c   (7110 kB)
[    0.000000]       .init : 0xc06fa000 - 0xc073dd80   ( 272 kB)
[    0.000000]       .data : 0xc073e000 - 0xc0781720   ( 270 kB)
[    0.000000]        .bss : 0xc0781720 - 0xc07ac45c   ( 172 kB)
[    0.000000] SLUB: Genslabs=11, HWalign=64, Order=0-3, MinObjects=0, CPUs=4, Nodes=1
[    0.000000] Hierarchical RCU implementation.
[    0.000000] NR_IRQS:16 nr_irqs:16 16
[    0.000000] ipc irq: irqchip registered, range 512-539
[    0.000000] main_pll_clk rate is 1167360000, postdiv = 2, mult = 18,prediv = 0
[    0.000000] pll_clk parent_rate(122880000 Hz), rate(327680000 Hz),postdiv = 6, mult = 15,
prediv = 0
[    0.000000] Architected local timer running at 194.56MHz (phys).
[    0.000000] Switching to timer-based delay loop
[    0.000000] sched_clock: 32 bits at 194MHz, resolution 5ns, wraps every 22075ms
[    0.000000] Console: colour dummy device 80x30
```

```
[    0.000069] Calibrating delay loop (skipped), value calculated using timer frequency.. 389.12
BogoMIPS (lpj=1945600)
[    0.000078] pid_max: default: 4096 minimum: 301
[    0.000205] Mount-cache hash table entries: 512
[    0.007588] CPU: Testing write buffer coherency: ok
[    0.007772] CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
[    0.007797] Setting up static identity map for 0x80510920 - 0x80510954
[    0.057896] CPU1: Booted secondary processor
[    0.057911] CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
[    0.107105] CPU2: Booted secondary processor
[    0.107120] CPU2: thread -1, cpu 2, socket 0, mpidr 80000002
[    0.156306] CPU3: Booted secondary processor
[    0.156321] CPU3: thread -1, cpu 3, socket 0, mpidr 80000003
[    0.156369] Brought up 4 CPUs
[    0.156388] SMP: Total of 4 processors activated (1556.48 BogoMIPS).
[    0.168876] NET: Registered protocol family 16
[    0.169846] DMA: preallocated 256 KiB pool for atomic coherent allocations
[    0.178249] hw-breakpoint: found 5 (+1 reserved) breakpoint and 4 watchpoint registers.
[    0.178255] hw-breakpoint: maximum watchpoint size is 8 bytes.
[    0.188147] bio: create slab <bio-0> at 0
[    0.188395] keystone-pcie: keystone_pcie_rc_init - start
[    0.188566] keystone2_pcie_serdes_setup
[    0.190634] keystone2_pcie_serdes_setup done, en_link_train = 1
[    0.190666] keystone-pcie: MEM 0x0000000050000000..0x000000005fffffff ->
0x0000000050000000
[    0.190675] keystone-pcie: IO 0x0000000024000000..0x0000000024003fff ->
0x0000000000000000
[    0.190707] keystone-pcie: pcie - number of legacy irqs = 4
[    0.190752] keystone-pcie: pcie - number of MSI host irqs = 8, msi_irqs = 32
[    0.298049] keystone-pcie: Doing PCI Setup...Done
[    0.298055] keystone-pcie: Starting PCI scan...
[    0.298175] PCI host bridge to bus 0000:00
[    0.298186] pci_bus 0000:00: root bus resource [mem 0x50000000-0x5fffffff]
[    0.298194] pci_bus 0000:00: root bus resource [io  0x0000-0x3fff]
[    0.298201] pci_bus 0000:00: No busn resource found for root bus, will use [bus 00-ff]
[    0.298245] PCI: bus0: Fast back to back transfers enabled
[    0.298260] keystone-pcie: Ending PCI scan...
[    0.298269] keystone-pcie: keystone_pcie_rc_init - end
[    0.298440] vgaarb: loaded
[    0.298758] SCSI subsystem initialized
[    0.299145] usbcore: registered new interface driver usbfs
[    0.299231] usbcore: registered new interface driver hub
```

[0.299330] usbcore: registered new device driver usb
[0.300576] pca953x 2-0020: failed reading register
[0.300593] pca953x: probe of 2-0020 failed with error -121
[0.300755] pps_core: LinuxPPS API ver. 1 registered
[0.300761] pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo Giometti <giometti@linux.it>
[0.300823] PTP clock support registered
[0.300944] keystone-hwqueue hwqueue.4: qmgr start queue 0, number of queues 8192
[0.301034] keystone-hwqueue hwqueue.4: added qmgr start queue 0, num of queues 8192, reg_peek e0840000, reg_status e0804000, reg_config e0806000, reg_region e0808000, reg_push e0880000, reg_pop e08c0000
[0.301042] keystone-hwqueue hwqueue.4: qmgr start queue 8192, number of queues 8192
[0.301122] keystone-hwqueue hwqueue.4: added qmgr start queue 8192, num of queues 8192, reg_peek e0900000, reg_status e080a400, reg_config e080c000, reg_region e080e000, reg_push e0940000, reg_pop e0980000
[0.301848] keystone-hwqueue hwqueue.4: qos: sched port @8096, drop sched @8000
[0.302710] keystone-hwqueue hwqueue.4: qos: sched port @6496, drop sched @6400
[0.303559] keystone-hwqueue hwqueue.4: added pool pool-net: 2048 descriptors of size 128
[0.303568] keystone-hwqueue hwqueue.4: added pool pool-rio: 128 descriptors of size 256
[0.303577] keystone-hwqueue hwqueue.4: added pool pool-udma: 1636 descriptors of size 256
[0.303585] keystone-hwqueue hwqueue.4: added pool pool-xge: 2048 descriptors of size 128
[0.303593] keystone-hwqueue hwqueue.4: added pool pool-crypto: 512 descriptors of size 128
[0.306146] keystone-hwqueue hwqueue.4: registered queues 0-16383
[0.306491] keystone-hwqueue hwqueue.4: qos version 0x2000105, magic valid
[0.306982] keystone-hwqueue hwqueue.4: qos version 0x2000105, magic valid
[0.312301] keystone-pktdma 2004000.pktdma: registered 24 logical channels, flows 32, tx chans: 9, rx chans: 24
[0.314884] keystone-pktdma 2a08000.pktdma: registered 24 logical channels, flows 32, tx chans: 32, rx chans: 32, loopback
[0.315400] keystone-pktdma 2fa1000.pktdma: registered 4 logical channels, flows 32, tx chans: 16, rx chans: 16
[0.315531] Switching to clocksource arch_sys_counter
[0.332024] NET: Registered protocol family 2
[0.332382] TCP established hash table entries: 4096 (order: 3, 32768 bytes)
[0.332445] TCP bind hash table entries: 4096 (order: 3, 32768 bytes)
[0.332503] TCP: Hash tables configured (established 4096 bind 4096)
[0.332533] TCP: reno registered
[0.332542] UDP hash table entries: 256 (order: 1, 8192 bytes)
[0.332558] UDP-Lite hash table entries: 256 (order: 1, 8192 bytes)
[0.332715] NET: Registered protocol family 1

```
[    0.332857] RPC: Registered named UNIX socket transport module.
[    0.332863] RPC: Registered udp transport module.
[    0.332867] RPC: Registered tcp transport module.
[    0.332872] RPC: Registered tcp NFSv4.1 backchannel transport module.
[    0.333254] hw perfevents: enabled with ARMv7 Cortex-A15 PMU driver, 7 counters available
[    0.418970] Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
[    0.419243] NTFS driver 2.1.30 [Flags: R/O].
[    0.419506] jffs2: version 2.2. (NAND) © 2001-2006 Red Hat, Inc.
[    0.419853] msgmni has been set to 999
[    0.420722] NET: Registered protocol family 38
[    0.420920] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 252)
[    0.420927] io scheduler noop registered
[    0.420933] io scheduler deadline registered
[    0.421071] io scheduler cfq registered (default)
[    0.422158] keystone-udma udma0.5: registered udma device udma0
[    0.464793] Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
[    0.466035] 2530c00.serial: ttyS0 at MMIO 0x2530c00 (irq = 309) is a 16550A
[    1.260623] console [ttyS0] enabled
[    1.264616] 2531000.serial: ttyS1 at MMIO 0x2531000 (irq = 312) is a 16550A
[    1.274554] loop: module loaded
[    1.277840] at24 0-0050: 131072 byte 24c1024 EEPROM, writable, 1 bytes/write
[    1.285710] Generic platform RAM MTD, (c) 2004 Simtec Electronics
[    1.299364] ONFI param page 0 valid
[    1.302860] ONFI flash detected
[    1.306015] NAND device: Manufacturer ID: 0x2c, Chip ID: 0xac (Micron MT29F4G08ABBDAHC),
512MiB, page size: 2048, OOB size: 64
[    1.317697] Bad block table found at page 262080, version 0x01
[    1.323951] Bad block table found at page 262016, version 0x01
[    1.330072] 3 ofpart partitions found on MTD device 30000000.nand
[    1.336186] Creating 3 MTD partitions on "30000000.nand":
[    1.341600] 0x000000000000-0x000000100000 : "u-boot"
[    1.347183] 0x000000100000-0x000000180000 : "params"
[    1.352703] 0x000000180000-0x000020000000 : "ubifs"
[    1.358321] davinci_nand 30000000.nand: controller rev. 2.5
[    1.364424] spi_davinci 21000400.spi: master is unqueued, this is deprecated
[    1.371751] m25p80 spi32766.0: found n25q128a11, expected n25q128
[    1.377870] m25p80 spi32766.0: n25q128a11 (16384 Kbytes)
[    1.383201] 2 ofpart partitions found on MTD device spi32766.0
[    1.389054] Creating 2 MTD partitions on "spi32766.0":
[    1.394205] 0x000000000000-0x000000080000 : "u-boot-spl"
[    1.400108] 0x000000080000-0x000001000000 : "test"
[    1.405743] spi_davinci 21000400.spi: Controller at 0xe0878400
```

```
[    1.412467] libphy: GPIO Bitbanged MDIO: probed
[    1.418734] tun: Universal TUN/TAP device driver, 1.6
[    1.423798] tun: (C) 1999-2004 Max Krasnyansky <maxk@qualcomm.com>
[    1.430401] keystone-netcp 2f00000.netcp: No streaming regs defined
[    1.436779] keystone-netcp 2090000.netcp: cpts rftclk freq not defined
[    1.444074] keystone-netcp 2090000.netcp: Created interface "eth0"
[    1.450284] keystone-netcp 2090000.netcp: dma_chan_name nettx0
[    1.456837] keystone-netcp 2090000.netcp: Created interface "eth1"
[    1.463039] keystone-netcp 2090000.netcp: dma_chan_name nettx1
[    1.470038] keystone-netcp 2f00000.netcp: Created interface "eth2"
[    1.476248] keystone-netcp 2f00000.netcp: dma_chan_name xgetx0
[    1.482453] keystone-netcp 2f00000.netcp: Created interface "eth3"
[    1.488662] keystone-netcp 2f00000.netcp: dma_chan_name xgetx1
[    1.495344] keystone-dwc3 2690000.dwc: usbss revision 47914300
[    1.501236] keystone-dwc3 2690000.dwc: mapped irq 425 to virq 608
[    1.709313] xhci-hcd xhci-hcd: xHCI Host Controller
[    1.714215] xhci-hcd xhci-hcd: new USB bus registered, assigned bus number 1
[    1.722017] xhci-hcd xhci-hcd: irq 608, io mem 0x02690000
[    1.727503] usb usb1: New USB device found, idVendor=1d6b, idProduct=0002
[    1.734311] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[    1.741559] usb usb1: Product: xHCI Host Controller
[    1.746454] usb usb1: Manufacturer: Linux 3.8.4 xhci-hcd
[    1.751779] usb usb1: SerialNumber: xhci-hcd
[    1.756399] hub 1-0:1.0: USB hub found
[    1.760163] hub 1-0:1.0: 1 port detected
[    1.764263] xhci-hcd xhci-hcd: xHCI Host Controller
[    1.769170] xhci-hcd xhci-hcd: new USB bus registered, assigned bus number 2
[    1.776311] usb usb2: New USB device found, idVendor=1d6b, idProduct=0003
[    1.783119] usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[    1.790368] usb usb2: Product: xHCI Host Controller
[    1.795258] usb usb2: Manufacturer: Linux 3.8.4 xhci-hcd
[    1.800586] usb usb2: SerialNumber: xhci-hcd
[    1.805178] hub 2-0:1.0: USB hub found
[    1.808954] hub 2-0:1.0: 1 port detected
[    1.813103] Initializing USB Mass Storage driver...
[    1.818104] usbcore: registered new interface driver usb-storage
[    1.824126] USB Mass Storage support registered.
[    1.828944] mousedev: PS/2 mouse device common for all mice
[    1.834712] i2c /dev entries driver
[    1.838606] gpio_poweroff_probe: Could not get GPIO 2
[    1.843500] poweroff-gpio: probe of gpio_poweroff.7 failed with error -16
[    1.850621] watchdog 22f0080.wdt: heartbeat 60 sec
```

[1.857076] keystone-crypto 20c0000.crypto: crypto accelerator enabled
[1.863927] usbcore: registered new interface driver usbhid
[1.869521] usbhid: USB HID core driver
[1.873631] remoteproc0: 2620040.dsp0 is available
[1.878527] remoteproc0: Note: remoteproc is still under development and considered experimental.
[1.887518] remoteproc0: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[1.897922] remoteproc0: no firmware found
[1.902207] rproc-user 2620040.dsp0: registered misc device dsp0
[1.908441] remoteproc1: 2620044.dsp1 is available
[1.913331] remoteproc1: Note: remoteproc is still under development and considered experimental.
[1.922323] remoteproc1: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[1.932735] remoteproc1: no firmware found
[1.937027] rproc-user 2620044.dsp1: registered misc device dsp1
[1.943238] remoteproc2: 2620048.dsp2 is available
[1.948137] remoteproc2: Note: remoteproc is still under development and considered experimental.
[1.957128] remoteproc2: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[1.967533] remoteproc2: no firmware found
[1.971823] rproc-user 2620048.dsp2: registered misc device dsp2
[1.978043] remoteproc3: 262004c.dsp3 is available
[1.982932] remoteproc3: Note: remoteproc is still under development and considered experimental.
[1.991925] remoteproc3: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[2.002327] remoteproc3: no firmware found
[2.006612] rproc-user 262004c.dsp3: registered misc device dsp3
[2.012826] remoteproc4: 2620050.dsp4 is available
[2.017724] remoteproc4: Note: remoteproc is still under development and considered experimental.
[2.026715] remoteproc4: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[2.037125] remoteproc4: no firmware found
[2.041408] rproc-user 2620050.dsp4: registered misc device dsp4
[2.047627] remoteproc5: 2620054.dsp5 is available
[2.052517] remoteproc5: Note: remoteproc is still under development and considered experimental.

[2.061509] remoteproc5: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[2.071912] remoteproc5: no firmware found
[2.076204] rproc-user 2620054.dsp5: registered misc device dsp5
[2.082412] remoteproc6: 2620058.dsp6 is available
[2.087310] remoteproc6: Note: remoteproc is still under development and considered experimental.
[2.096301] remoteproc6: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[2.106704] remoteproc6: no firmware found
[2.110989] rproc-user 2620058.dsp6: registered misc device dsp6
[2.117211] remoteproc7: 262005c.dsp7 is available
[2.122101] remoteproc7: Note: remoteproc is still under development and considered experimental.
[2.131093] remoteproc7: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[2.141505] remoteproc7: no firmware found
[2.145792] rproc-user 262005c.dsp7: registered misc device dsp7
[2.151832] rproc-user dspmem.3: kick gpio
[2.156038] rproc-user dspmem.3: registered misc device dspmem
[2.162167] oprofile: using arm/armv7-ca15
[2.166454] GACT probability on
[2.169604] Mirror/redirect action on
[2.173273] Simple TC action Loaded
[2.177149] netem: version 1.3
[2.180211] u32 classifier
[2.182920] Performance counters on
[2.186768] input device check on
[2.190435] Actions configured
[2.193847] Netfilter messages via NETLINK v0.30.
[2.198576] nf_conntrack version 0.5.0 (7993 buckets, 31972 max)
[2.204979] ctnetlink v0.93: registering with nfnetlink.
[2.210546] IPv4 over IPv4 tunneling driver
[2.215109] gre: GRE over IPv4 demultiplexor driver
[2.220006] ip_gre: GRE over IPv4 tunneling driver
[2.225293] ip_tables: (C) 2000-2006 Netfilter Core Team
[2.230708] ipt_CLUSTERIP: ClusterIP Version 0.8 loaded successfully
[2.237105] arp_tables: (C) 2002 David S. Miller
[2.241768] TCP: cubic registered
[2.245088] Initializing XFRM netlink socket
[2.249857] NET: Registered protocol family 10
[2.254981] NET: Registered protocol family 17

```
[    2.259458] NET: Registered protocol family 15
[    2.263959] Bridge firewalling registered
[    2.267987] Ebtables v2.0 registered
[    2.271632] 8021q: 802.1Q VLAN Support v1.8
[    2.276289] sctp: Hash tables configured (established 16384 bind 16384)
[    2.283019] NET: Registered protocol family 40
[    2.287600] VFP support v0.3: implementor 41 architecture 4 part 30 variant f rev 0
[    2.295293] Registering SWP/SWPB emulation handler
[    2.300949] UBI: attaching mtd2 to ubi0
[    3.951758] UBI: scanning is finished
[    3.962324] UBI warning: print_rsvd_warning: cannot reserve enough PEBs for bad PEB
handling, reserved 40, need 76
[    3.974519] UBI: attached mtd2 (name "ubifs", size 510 MiB) to ubi0
[    3.980814] UBI: PEB size: 131072 bytes (128 KiB), LEB size: 126976 bytes
[    3.987626] UBI: min./max. I/O unit sizes: 2048/2048, sub-page size 2048
[    3.994346] UBI: VID header offset: 2048 (aligned 2048), data offset: 4096
[    4.001243] UBI: good PEBs: 4080, bad PEBs: 4, corrupted PEBs: 0
[    4.007269] UBI: user volume: 2, internal volumes: 1, max. volumes count: 128
[    4.014424] UBI: max/mean erase counter: 1/0, WL threshold: 4096, image sequence number:
819262559
[    4.023417] UBI: available PEBs: 0, total reserved PEBs: 4080, PEBs reserved for bad PEB
handling: 40
[    4.032675] UBI: background thread "ubi_bgt0d" started, PID 1242
[    4.032985] input: gpio_keys.8 as /devices/soc.0/gpio_keys.8/input/input0
[    4.046121] UBIFS: parse sync
[    4.050551] UBIFS: background thread "ubifs_bgt0_1" started, PID 1247
[    4.077353] UBIFS: start fixing up free space
[    7.580960] UBIFS: free space fixup complete
[    7.598495] UBIFS: mounted UBI device 0, volume 1, name "rootfs"(null)
[    7.605043] UBIFS: LEB size: 126976 bytes (124 KiB), min./max. I/O unit sizes: 2048 bytes/2048
bytes
[    7.614211] UBIFS: FS size: 500539392 bytes (477 MiB, 3942 LEBs), journal size 9023488 bytes
(8 MiB, 72 LEBs)
[    7.624161] UBIFS: reserved for root: 0 bytes (0 KiB)
[    7.629231] UBIFS: media format: w4/r0 (latest is w4/r0), UUID 58EA5B74-9FC1-457C-8A02-
5D08ADEA3BF0, small LPT model
[    7.641281] VFS: Mounted root (ubifs filesystem) on device 0:11.
[    7.647444] Freeing init memory: 268K
INIT: version 2.88 booting
Starting udev
Starting Bootlog daemon: bootlogd.
```

```
Configuring network interfaces... [    9.070693] keystone-netcp 2090000.netcp: initializing cpsw
version 1.3 (1) SGMII identification value 0x4ed1
[    9.080844] keystone-netcp 2090000.netcp: Created a cpsw ale engine
[    9.087205] keystone-netcp 2090000.netcp: initialized cpsw ale revision 1.3
[    9.097048] pps pps0: new PPS source ptp0
[    9.101085] cpts rftclk rate(583680000 HZ),mult(1839607018),shift(30)
[    9.143597] keystone-netcp 2090000.netcp: Using Packet Accelerator Firmware version
0x01030008
[    9.152249] keystone-netcp 2090000.netcp: pa_clk_rate(163840000 HZ),mult(25000),shift(12)
[    9.164377] net eth0: netcp device eth0 opened
[    9.170522] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[    9.176381] 8021q: adding VLAN 0 to HW filter on device eth0
[    9.182058] net eth0: adding rx vlan id: 0
udhcpc (v1.20.2) started
[    9.195788] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
Sending discover...
Sending select for 157.87.43.197...
Lease of 157.87.43.197 obtained, lease time 28800
/etc/udhcpc.d/50default: Adding DNS 192.0.2.2
/etc/udhcpc.d/50default: Adding DNS 192.0.2.3
[   10.645899] keystone-netcp 2090000.netcp: initializing cpsw version 1.3 (1) SGMII
identification value 0x4ed1
[   10.661558] net eth1: netcp device eth1 opened
[   10.666638] IPv6: ADDRCONF(NETDEV_UP): eth1: link is not ready
[   10.672488] 8021q: adding VLAN 0 to HW filter on device eth1
[   10.678170] net eth1: adding rx vlan id: 0
udhcpc (v1.20.2) started
[   10.755636] IPv6: ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready
Sending discover...
Sending discover...
Sending discover...
No lease, failing
done.
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.all.rp_filter = 1
Tue Oct  1 06:41:00 UTC 2013
Configuring update-modules.
Configuring debianutils.
update-alternatives: Linking //usr/sbin/add-shell to add-shell.debianutils
update-alternatives: Linking //usr/sbin/installkernel to installkernel.debianutils
update-alternatives: Linking //usr/sbin/mkboot to mkboot.debianutils
update-alternatives: Linking //usr/sbin/remove-shell to remove-shell.debianutils
```

```
update-alternatives: Linking //usr/bin/savelog to savelog.debianutils
update-alternatives: Linking //usr/bin/sensible-browser to sensible-browser.debianutils
update-alternatives: Linking //usr/bin/sensible-editor to sensible-editor.debianutils
update-alternatives: Linking //usr/bin/sensible-pager to sensible-pager.debianutils
update-alternatives: Linking //usr/bin/which to which.debianutils
update-alternatives: Error: cannot register alternative run-parts to /bin/run-parts since it is
already registered to /usr/bin/run-parts
update-alternatives: Linking //bin/tempfile to tempfile.debianutils
Configuring ndisc6-ndisc6.
Configuring ndisc6-rltraceroute6.
Configuring ndisc6-rdisc6.
INIT: Entering runlevel: 5
Starting system message bus: dbus.
Starting Dropbear SSH server: Will output 1024 bit rsa secret key to
'/etc/dropbear/dropbear_rsa_host_key'
Generating key, this may take a while...
Public key portion is:
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAAAgwCz4mdL+FftZHSiz0iiwX0Dbonr6+g1cjghOdww5O+CUmq
aMRcR7RrREklh2H+ouN5IzWwEIaBQBznIbXaioeZeKYh1gOUw+k2xjRXIxqCIrPC2+U4h9Gay1Tb4v
mZC32qs/5vsk8v/H/dX/KrMIwLPI64mx+ORHtp23m8Vt9D11qcR root@keystone-evm
Fingerprint: md5 69:41:54:50:c3:f2:ab:c3:51:32:bf:cf:4e:89:be:a2
dropbear.
Starting mpmsrv daemon.
Starting telnet daemon.
#>>>>> LCD 12
IP Address:
157.87.43.197
Starting tiipclad daemon.
Starting syslog-ng:.
Starting thttpd.
Starting Lighttpd Web Server: lighttpd.
2013-10-01 06:41:01: (log.c.166) server started
* starting FTP Server: vsftpd... done.
*****************************************************************
*****************************************************************
NOTICE: This file system contains the followin GPLv3 packages:
    binutils-symlinks
    binutils
    gdb
    gdbserver
```

If you do not wish to distribute GPLv3 components please remove
the above packages prior to distribution. This can be done using
the opkg remove command. i.e.:

opkg remove <package>

Where <package> is the name printed in the list above

NOTE: If the package is a dependency of another package you
will be notified of the dependent packages. You should
use the --force-removal-of-dependent-packages option to
also remove the dependent packages as well

****

****

Stopping Bootlog daemon: bootlogd.

```
 _____          _____      _     _
| _ |___ ___ ___ ___  | _ |___ ___  |_|___ ___| |_
|   | _| .'| . | . |  |  __|  _| . | | | -_|  _|  _|
|__|__|_| |__,|_  |___|  |__|  |_| |___|_| |___|___|_|
       |___|          |___|
```

Arago Project http://arago-project.org keystone-evm ttyS0

Arago 2013.04 keystone-evm ttyS0

keystone-evm login:

```
 _____                  _____          _         _
|   _   |___ ___ ___ ___   |   _   |___ ___   |_|___ ___| |_
|       |  _| .'| . | . |  |    __|  _| . | | | -_|  _|  _|
|__|__|_| |__,|_  |___|  |__|  |_| |___|_| |___|___|_|
              |___|                   |___|

Arago Project http://arago-project.org keystone-evm ttyS0

Arago 2013.04 keystone-evm ttyS0

keystone-evm login: root
root@keystone-evm:~# cd /
root@keystone-evm:/# ls
bin     debug   etc     ipaddr  media   proc    srv     tmp     var
boot    dev     home    lib     mnt     sbin    sys     usr     www
root@keystone-evm:/#
```

Appendix I

Usable Size Calculation As documented here, UBI reserves a certain amount of space for management and bad PEB handling operations. Specifically:

- 2 PEBs are used to store the UBI volume table
- 1 PEB is reserved for wear-leveling purposes;
- 1 PEB is reserved for the atomic LEB change operation;
- a % of PEBs is reserved for handling bad EBs. The default for NAND is 1%
- UBI stores the erase counter (EC) and volume ID (VID) headers at the beginning of each PEB.
- 1 min I/O unit is required for each of these.

To calculate the full overhead, we need the following values:

Symbol	Meaning	value
SP	PEB Size	128KiB
SL	LEB Size	128KiB-2x2KiB=124KiB
P	Total number of PEBs on the MTD device	510.5MiB / 128KiB = 4084
B	Number of PEBs reserved for bad PEB handling	1% of P = 40.84(round to 41)
O	The overhead related to storing EC and VID headers in bytes, i.e. O=SP-SL	4KiB

UBI Overhead = (B + 4) * SP + O * (P - B - 4) = (41 + 4) * 128KiB + 4 KiB * (4084- 41- 4) = 21916 KiB = 171.21875 PEBs (round to 171)

This leaves us with 3913 PEBs or 500864 KiB available for user data.

Note that we used "-c 4039 " in the above mkfs.ubifs command line to specify the maximum filesystem size, not "-c 3913" The reason for this is that mkfs.ubifs operates in terms of LEB size (124 KiB), not PEB size (128Kib). 500864KiB / 124 KiB = 4039.22 (round to 4039)

Use this calculation method to calculate the size required for each ubifs image going into each of the ubifs volumes on NAND, for example, we can use 10MiB parameter, which is defined in ubinize.cfg, to calculate c parameter (which is 75) for creating keystone-evm-boot.ubifs.

Appendix II

The 'mtd-utility' requires zlib, lzo, and uuid (from e2fsprogs) libraries [6], we can create mkfs.ubifs and ubinize utilities as follows:

- Source code preparation:
  (1) zlib: Download zlib from http://zlib.net/, the zlib version 1.2.5 is tested.
  (2) lzo: Download from http://www.oberhumer.com/opensource/lzo/download/, the lzo version 2.0.6 is tested.
  (3) e2fsprogs: Download e2fsprogs from http://e2fsprogs.sourceforge.net/, the e2fsprogs version 1.42 is tested.
  (4) MTD-Utils: MTD utils are available from http://git.infradead.org/mtd-utils.git, the MTD-Utils version release 1.4.8 is tested.

We put these tar files in ~/mtd as base directory, and we install build results in local sub-directory in ~/mtd/install

```
damon@damon-laptop:~/mtd$ ls
e2fsprogs-1.42.tar.gz  lzo-2.06.tar.gz          zlib-1.2.5.tar.gz
install                mtd-utils-d37fcc0.tar.gz
damon@damon-laptop:~/mtd$
```

- Detailed procedures for compilling mtd-utils tools

  (1) Related compile command for zlib

```
host$ tar xvf zlib-1.2.5.tar.gz
host$ cd zlib-1.2.5/
host$ ./configure --prefix=~/mtd/install
host$ make
host$ make install
host$ cd ..
```

Result should be libz.a in ~/mtd/install/lib directory and zlib's headers in ~/mtd/install/include.

```
damon@damon-laptop:~/mtd/install/lib$ ls
libz.a  libz.so  libz.so.1  libz.so.1.2.5  pkgconfig
damon@damon-laptop:~/mtd/install/lib$ cd ../include/
damon@damon-laptop:~/mtd/install/include$ ls
zconf.h  zlib.h
damon@damon-laptop:~/mtd/install/include$
```

  (2) Related compile command for lzo

```
host$ tar xvf lzo-2.06.tar.gz
host$ cd lzo-2.06/
host$ ./configure --build=i686-pc-linux --prefix=~/mtd/install
host$ make
```

host$ make install

host$ cd ..

Note: In some case, we may encounter the configure error as below, and we can use absolute directory to solve it, use /home/<your account name>/mtd to replace ~/mtd

```
damon@damon-laptop:~/mtd/lzo-2.06$ ./configure --build=i686-pc-linux --prefix
=~/mtd/install/
configure: error: expected an absolute directory name for --prefix: ~/mtd/ins
tall
```

Result should be liblzo2.a in ~/mtd/install/lib directory and lzo's headers in ~/mtd/install/include/lzo.

```
damon@damon-laptop:~/mtd/install/lib$ ls
liblzo2.a  liblzo2.la  libz.a  libz.so  libz.so.1  libz.so.1.2.5  pkgconfig
damon@damon-laptop:~/mtd/install/lib$ cd ../include/lzo/
damon@damon-laptop:~/mtd/install/include/lzo$ ls
lzo1a.h  lzo1c.h  lzo1.h   lzo1y.h  lzo2a.h    lzoconf.h  lzoutil.h
lzo1b.h  lzo1f.h  lzo1x.h  lzo1z.h  lzo_asm.h  lzodefs.h
damon@damon-laptop:~/mtd/install/include/lzo$
```

(3) Related compile command for e2fsprogs

host$ tar xvf e2fsprogs-1.42.tar.gz

host$ cd e2fsprogs-1.42/

host$ ./configure --build=i686-pc-linux --prefix=~/mtd/install

host$ make

host$ make install

host$ cd lib/uuid/

host$ make install

host$ cd ../../../

Note: In some case, we may need use absolute directory, like /home/<username>/mtd to replace ~/mtd

Result should be libuuid.a in ~/mtd/install/lib directory and uuid's headers in ~/mtd/install/include/uuid.

```
damon@damon-laptop:~/mtd/install/lib$ ls
e2initrd_helper  liblzo2.la  libz.a   libz.so.1      pkgconfig
liblzo2.a        libuuid.a   libz.so  libz.so.1.2.5
damon@damon-laptop:~/mtd/install/lib$ cd ../include/uuid
damon@damon-laptop:~/mtd/install/include/uuid$ ls
uuid.h
damon@damon-laptop:~/mtd/install/include/uuid$
```

(4) Related compile command for mtd-utils

host$ tar xvf mtd-utils-d37fcc0.tar.gz

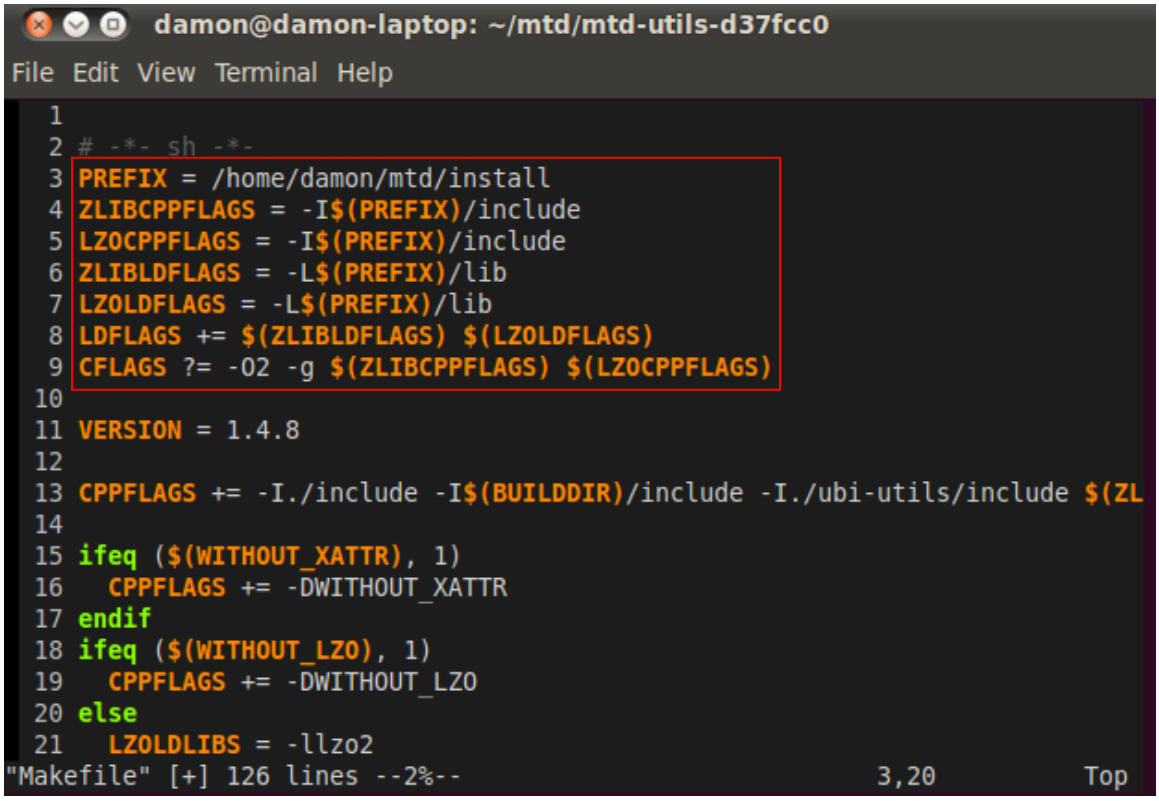
host$ cd mtd-utils-d37fcc0/

mtd-utils doesn't have a configure script, so we have to edit the Makefile again. Depending on the version of mtd-utils, make sure head of top level Makefile has:

```
host$ vi Makefile
```

```
PREFIX = /home/<username>/mtd/install
ZLIBCPPFLAGS = -I$(PREFIX)/include
LZOCPPFLAGS = -I$(PREFIX)/include
ZLIBLDFLAGS = -L$(PREFIX)/lib
LZOLDFLAGS = -L$(PREFIX)/lib
LDFLAGS += $(ZLIBLDFLAGS) $(LZOLDFLAGS)
CFLAGS ?= -O2 -g $(ZLIBCPPFLAGS) $(LZOCPPFLAGS)
```

Save and close vi editor



Edit the common.mk file and comment out the PREFIX=/usr line

```
host$ vi common.mk
      #PREFIX=/usr
```

Save and close vi editor

```
damon@damon-laptop: ~/mtd/mtd-utils-d37fcc0
File  Edit  View  Terminal  Help
CFLAGS += $(WFLAGS)
SECTION_CFLAGS := $(call cc-option,-ffunction-sections -fdata-sections -
CFLAGS += $(SECTION_CFLAGS)

ifneq ($(WITHOUT_LARGEFILE), 1)
  CPPFLAGS += -D_FILE_OFFSET_BITS=64
endif

DESTDIR?=
#PREFIX=/usr
EXEC_PREFIX=$(PREFIX)
SBINDIR=$(EXEC_PREFIX)/sbin
MANDIR=$(PREFIX)/share/man
INCLUDEDIR=$(PREFIX)/include

ifndef BUILDDIR
ifeq ($(origin CROSS),undefined)
  BUILDDIR := $(CURDIR)
else
# Remove the trailing slash to make the directory name
  BUILDDIR := $(CURDIR)/$(CROSS:-=)
"common.mk" [+] 91 lines --26%--                          24,1          20%
```

host$ WITHOUT_XATTR=1 make

host$ make install DESTDIR=~/mtd/install

host$ cd ..

Directory ~/mtd/install/sbin/ should now contain compiled MTD utils you can use on Linux host. The binaries for the tools used to create a UBIFS root file system will be found here:

~/mtd/install/home/<username>/mtd/install/sbin/mkfs.ubifs
~/mtd/install/home/<username>/mtd/install/sbin/ubinize

```
damon@damon-laptop:~/mtd/install/home/damon/mtd/install/sbin$ ls
docfdisk        flash_unlock  mtdinfo       rfdformat     ubinfo
doc_loadbios    ftl_check     nanddump      serve_image   ubinize
flashcp         ftl_format    nandtest      sumtool       ubirename
flash_erase     jffs2dump     nandwrite     ubiattach     ubirmvol
flash_eraseall  jffs2reader   nftldump      ubicrc32      ubirsvol
flash_lock      mkfs.jffs2    nftl_format   ubidetach     ubiupdatevol
flash_otp_dump  mkfs.ubifs    recv_image    ubiformat
flash_otp_info  mtd_debug     rfddump       ubimkvol
damon@damon-laptop:~/mtd/install/home/damon/mtd/install/sbin$
```

Reference

[1] How do I create an UBIFS image: http://www.linux-mtd.infradead.org/faq/ubifs.html#L_mkfubifs

[2] UBI – Unsorted Block Images: http://www.linux-mtd.infradead.org/doc/ubi.html

[3] UBIFS – UBI File-System: http://www.linux-mtd.infradead.org/doc/ubifs.html

[4] MCSDK User Guide for KeyStone II: http://processors.wiki.ti.com/index.php/MCSDK_User_Guide_for_KeyStone_II

[5] Build and Run U-boot and Linux Kernel on TCI6638 EVM-V2: http://www.deyisupport.com/question_answer/dsp_arm/c6000_multicore/f/53/t/23812.aspx

[6] MTD-Utils Compilation: http://processors.wiki.ti.com/index.php/MTD_Utilities#MTD-Utils_Compilation