

Build and Run U-boot and Linux Kernel on TCI6638 EVM

1. Environment preparation

This document based on the following environment:

- Ubuntu 10.04 LTS 32-bit (virtual machine or installation machine) or higher LTS version (can get from <http://www.ubuntu.com/download/desktop>)
- “Linaro” cross compile tool chain (can get from https://launchpad.net/linaro-toolchain-binaries/trunk/2013.03/+download/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux.tar.bz2) or higher version
- CCS v5.4.0.00091 or higher version, (can get from http://software-dl.ti.com/dsps/forms/self_cert_export.html?prod_no=ccs_setup_5.4.0.00091.exe&ref_url=http://software-dl.ti.com/dsps/dsps_public_sw/sdo_ccstudio/CCSv5/CCS_5_4_0/)
- MCSDK 3.0 GA (Alpha11) or higher version packet (can get from http://software-dl.ti.com/sdoemb/sdoemb_public_sw/mcsdk/03_00_00_11/index_FDS.html)
- TCI6638 EVM USB-COM converter driver (can get from <http://www.ftdichip.com/Drivers/D2XX.htm>)
- TFTP tools (can get from http://tftpd32.jounin.net/tftpd32_download.html)

1.1 Proxy settings (optional, example with TI intranet configurations)

Configure apt-get by adding following to /etc/apt/apt.conf (create the file if does not exist). Root access is required to modify or create this file. A simple text editor, “gedit”, can be used to perform these changes:

```
sudo gedit /etc/apt/apt.conf
```

Add following part to make the apt-get can get Ubuntu components through proxy (red part should be your proxy server and port):

```
ACQUIRE
{
    http::proxy "http://wwwgate.ti.com:80"
}
```

Open and edit the `$HOME/.bashrc` file (create it if does not exist), root access is required to modify this file. A simple text editor, “gedit”, can be used to perform these changes:

```
sudo gedit $HOME/.bashrc
```

Add the following items in this file to setup proxy, then save and exit it:

```
export http_proxy="http://webproxy.ext.ti.com:80"  
export ftp_proxy="http://webproxy.ext.ti.com:80"  
export https_proxy="http://webproxy.ext.ti.com:80"  
export no_proxy="ti.com"  
export GIT_PROXY_COMMAND=$HOME/git-proxy.sh
```

Source `.bashrc` file after updating it:

```
source ~/.bashrc
```

Add following in `$HOME/git-proxy.sh` (create the file if it does not exist) and save the file (red part should be your proxy server and port):

```
#!/bin/sh  
if [ $(getent hosts intranet.ti.com | cut -d' ' -f 1) x = "127.0.0.1x" ]  
then  
    # this machine is inside intranet network  
    if echo $1 | grep ti.com > /dev/null  
    then  
        # ... and so is the remote machine  
        socat - tcp:$1:$2  
    else  
        socat - proxy:wwwgate.ti.com:$1:$2,proxyport=80  
    fi  
else  
    socat - tcp:$1:$2  
fi
```

1.2 Necessary Ubuntu components

Install relative necessary Ubuntu components by using apt-get install:

```
sudo apt-get install -y build-essential git-core expect automake socat  
uboot-mkimage
```

Then perform the following command:

```
echo "no" | sudo dpkg-reconfigure -f teletype dash
```

1.3 Cross compile tool chain

Please download the cross compile tool chain in the following link:

https://launchpad.net/linaro-toolchain-binaries/trunk/2013.03/+download/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux.tar.bz2

Download the Linux binary packet, locate it in your \$HOME folder, then untar it by using:

```
tar xjf gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux.tar.bz2
```

Open and edit the \$HOME/.bashrc file by using:

```
sudo gedit $HOME/.bashrc
```

Add the following items in this file, then save and exit it:

```
export PATH=<path-to-linaro-bin>:$PATH
```

```
export CROSS_COMPILE=arm-linux-gnueabihf-
```

```
export ARCH=arm
```

For example, the PATH can be: “export PATH=/home/vincent/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux/bin:\$PATH”

After updating the .bashrc file and check the value of the PATH environment variable to make sure it was updated:

```
source ~/.bashrc
```

```
echo $PATH
```

You should see the path you set in the print info, which means the environment settings take effect.

Tips: If the path settings cannot take effect or cannot show out when echo it, please restart your virtual machine (or machine) to make it takes effect.

2. Build U-boot

2.1 Download the TCI6638 U-boot from arago

Create a new folder for the U-boot and enter it:

```
mkdir ~/tci6638-uboot
```

```
cd tci6638-uboot/
```

Git clone the TCI6638 U-boot from arago:

```
git clone git://arago-project.org/git/projects/u-boot-keystone.git
```

After git clone, you'll see the following print info:

```
vincent@vincent-desktop:~/tci6638-uboot$ git clone git://arago-project.org/git/projects/u-boot-keystone.git
Initialized empty Git repository in /home/vincent/tci6638-uboot/u-boot-keystone/.git/
remote: Counting objects: 198647, done.
remote: Compressing objects: 100% (37694/37694), done.
remote: Total 198647 (delta 160131), reused 196576 (delta 158155)
Receiving objects: 100% (198647/198647), 51.43 MiB | 429 KiB/s, done.
Resolving deltas: 100% (160131/160131), done.
vincent@vincent-desktop:~/tci6638-uboot$
```

The TCI6638 U-boot has many branches, reset to your preferred tags by using:

```
cd tci6638-uboot
```

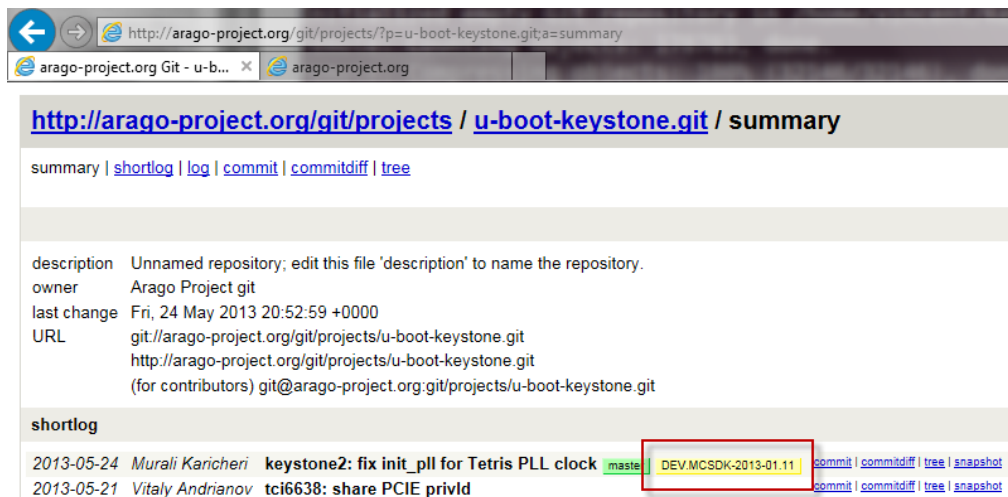
```
git reset --hard <Release tag>
```

For example, we can use the MCSDK alpha 11 tags:

```
git reset --hard DEV.MCSDK-2013-01.11
```

The “Release tag” here means the released version you want to checkout, you can find this info on the arago u-boot-keystone site for the latest update tags:

<http://arago-project.org/git/projects/?p=u-boot-keystone.git;a=summary>



The screenshot shows a web browser displaying the summary page of the Arago Project Git repository for u-boot-keystone. The page title is "http://arago-project.org/git/projects / u-boot-keystone.git / summary". Below the title, there are links for "summary", "shortlog", "log", "commit", "commitdiff", and "tree". The main content area displays repository metadata: description (Unnamed repository), owner (Arago Project git), last change (Fri, 24 May 2013 20:52:59 +0000), and URL (git://arago-project.org/git/projects/u-boot-keystone.git). Below this, there is a "shortlog" section showing a list of commits. The first commit is dated 2013-05-24 by Murali Karicheri, titled "keystone2: fix init_pll for Tetris PLL clock", and is marked as "master". The second commit is dated 2013-05-21 by Vitaly Andrianov, titled "tci6638: share PCIE privld". The "DEV.MCSDK-2013-01.11" tag is highlighted with a red box in the shortlog table.

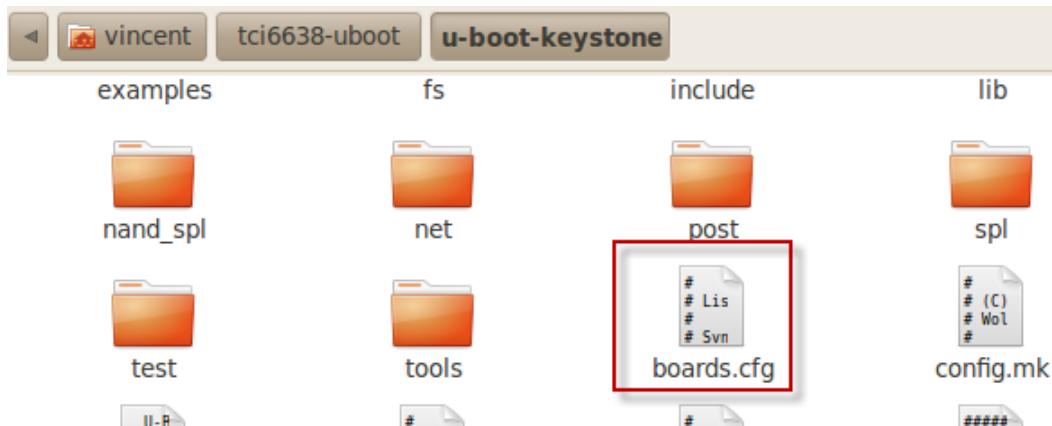
date	author	commit	message	branch	tags
2013-05-24	Murali Karicheri	keystone2: fix init_pll for Tetris PLL clock	master	DEV.MCSDK-2013-01.11	
2013-05-21	Vitaly Andrianov	tci6638: share PCIE privld			

You will see the following info:

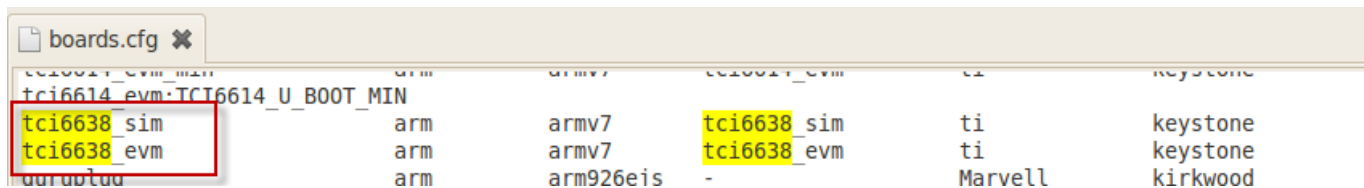
```
vincent@vincent-desktop:~/tci6638-uboot$ cd u-boot-keystone/  
vincent@vincent-desktop:~/tci6638-uboot/u-boot-keystone$ git reset --hard DEV.MCSDK-2013-01.11  
HEAD is now at 1a0354e keystone2: fix init_pll for Tetris PLL clock  
vincent@vincent-desktop:~/tci6638-uboot/u-boot-keystone$
```

2.2 Build the TCI6638 U-boot

Enter the git folder, you can see a file named “boards.cfg”:



This file shows the supported configuration types in this U-boot version:



We need to choose and build the preferred target first by using (**type twice**):

```
make tci6638_evm_config
```

```
make tci6638_evm_config
```

You'll see the following print info:

```
vincent@vincent-desktop:~/tci6638-uboot/u-boot-keystone$ make tci6638_evm_config  
Configuring for tci6638_evm board...  
vincent@vincent-desktop:~/tci6638-uboot/u-boot-keystone$ make tci6638_evm_config  
Generating include/autoconf.mk  
Generating include/autoconf.mk.dep  
Configuring for tci6638_evm board...  
vincent@vincent-desktop:~/tci6638-uboot/u-boot-keystone$
```

Then we will make the U-boot image by using:

```
make
```

You'll see the following info after build complete:

```

arm-linux-gnueabi-ld.bfd -r -o libstubs.o stubs.o
arm-linux-gnueabi-ld.bfd -g -Ttext 0xc100000 \
-o hello_world -e hello_world hello_world.o libstubs.o \
-L/home/vincent/gcc-linaro-arm-linux-gnueabi-2012.03-20120326_linux/bin/../lib/gcc/arm-linux-gn
ueabi/4.6.3 -lgcc
arm-linux-gnueabi-objcopy -O srec hello_world hello_world.srec 2>/dev/null
arm-linux-gnueabi-objcopy -O binary hello_world hello_world.bin 2>/dev/null
make[1]: Leaving directory `/home/vincent/tci6638-uboot/u-boot-keystone/examples/standalone'
make -C examples/api all
make[1]: Entering directory `/home/vincent/tci6638-uboot/u-boot-keystone/examples/api'
make[1]: Nothing to be done for `all'.
make[1]: Leaving directory `/home/vincent/tci6638-uboot/u-boot-keystone/examples/api'
vincent@vincent-desktop:~/tci6638-uboot/u-boot-keystone$

```

The built u-boot.bin will be located in the same folder:



Then built the u-boot image “u-boot-spi.gph” with SPL which uses for SPI NOR flash boot by using:

```
make tci6638_evm_config
```

```
make tci6638_evm_config
```

```
make u-boot-spi.gph
```

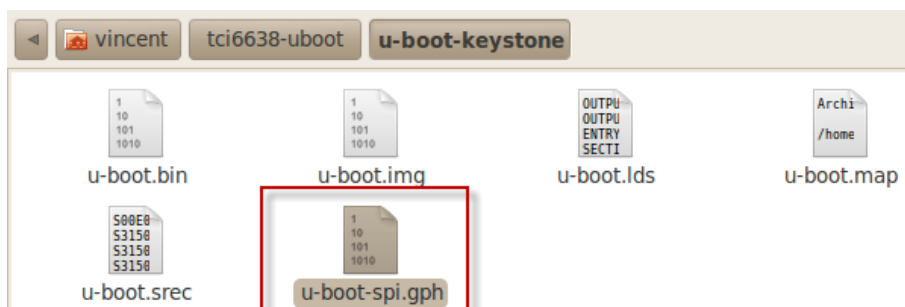
You’ll see the following info after build complete:

```

arm-linux-gnueabi-objcopy --gap-fill=0xff -O binary /home/vincent/tci6638-uboot/u-boot-keystone/spl/u-boot-spl /home/vi
ncent/tci6638-uboot/u-boot-keystone/spl/u-boot-spl.bin
make[1]: Leaving directory `/home/vincent/tci6638-uboot/u-boot-keystone/spl'
tools/mkimage -A arm -T gpimage -C none \
-a 0x0c200000 -e 0x0c200000 \
-n SPL -d spl/u-boot-spl.bin spl/u-boot-spl.gph
GP Header: Size 2ad0 LoadAddr c200000
arm-linux-gnueabi-objcopy --gap-fill=0xff -I binary \
--pad-to=65536 --gap-fill=0 -O binary \
spl/u-boot-spl.gph spl/u-boot-spl-pad.gph
cat spl/u-boot-spl-pad.gph u-boot.img > u-boot-spi.gph
vincent@vincent-desktop:~/tci6638-uboot/u-boot-keystone$

```

The built “u-boot-spi.gph” will be located in the same folder:



3. Build Linux kernel

3.1 Download the TCI6638 Linux kernel from arago

Create a new folder for the Linux kernel and enter it:

```
mkdir ~/tci6638-linux-kernel
```

```
cd tci6638-linux-kernel/
```

Git clone the TCI6638 Linux kernel from arago:

```
git clone git://arago-project.org/git/projects/linux-keystone.git
```

After git clone, you'll see the following print info:

```
vincent@vincent-desktop:~/tci6638-linux-kernel$ git clone git://arago-project.org/git/projects/linux-keystone.git
Initialized empty Git repository in /home/vincent/tci6638-linux-kernel/linux-keystone/.git/
remote: Counting objects: 2864439, done.
remote: Compressing objects: 100% (473004/473004), done.
Receiving objects: 100% (2864439/2864439), 612.08 MiB | 706 KiB/s, done.
remote: Total 2864439 (delta 2400977), reused 2827195 (delta 2363831)
Resolving deltas: 100% (2400977/2400977), done.
Checking out files: 100% (41630/41630), done.
vincent@vincent-desktop:~/tci6638-linux-kernel$
```

Created your local master branch and checkout with master branch in git created folders:

```
cd linux-keystone/
```

```
git reset --hard <Release tag>
```

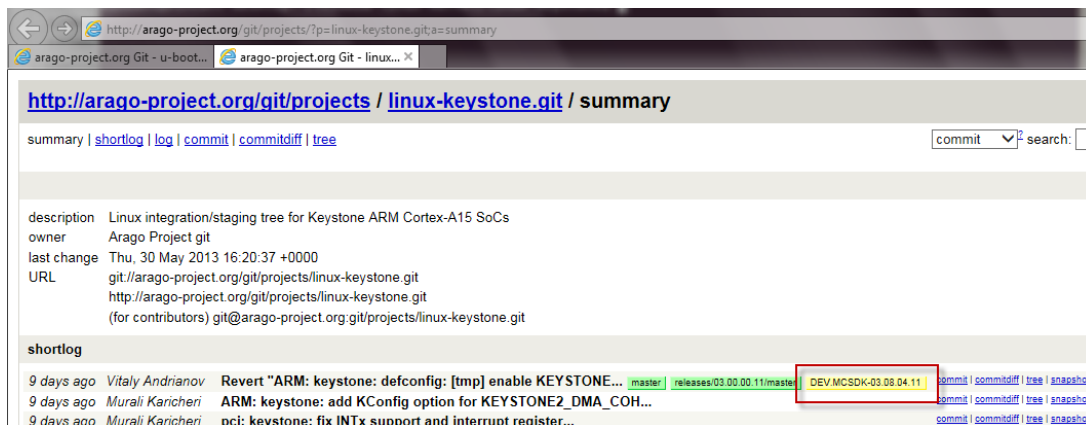
For example, we can use the MCSDK alpha 11 tags:

```
git reset --hard DEV.MCSDK-03.08.04.11
```

The “Release tag” here means the released version you want to checkout, you can find this info on the arago linux-keystone site for the latest update tags:

<http://arago-project.org/git/projects/?p=linux-keystone.git;a=summary>

You'll see the latest branch information is like:



The screenshot shows a web browser displaying the Arago Project Git repository page for linux-keystone. The page title is "http://arago-project.org/git/projects / linux-keystone.git / summary". Below the title, there is a navigation bar with links: "summary | shortlog | log | commit | commitdiff | tree". A search bar is also present. The main content area shows the repository description: "Linux integration/staging tree for Keystone ARM Cortex-A15 SoCs", owner "Arago Project git", last change "Thu, 30 May 2013 16:20:37 +0000", and URL "git://arago-project.org/git/projects/linux-keystone.git". Below this, the "shortlog" section lists recent commits. The first commit is "9 days ago Vitaly Andrianov Revert 'ARM: keystone: defconfig: [tmp] enable KEYSTONE...'", followed by "9 days ago Murali Karicheri ARM: keystone: add KConfig option for KEYSTONE2_DMA_COH...", and "9 days ago Murali Karicheri pci: keystone: fix INTx support and interrupt register...". The commit hashes are visible, and the "DEV.MCSDK-03.08.04.11" tag is highlighted in red.

You'll see the following info:

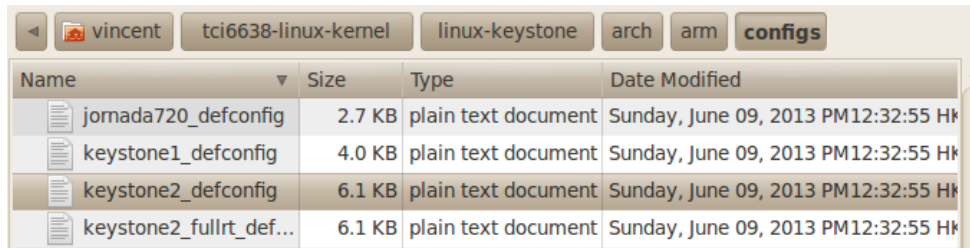
```
vincent@vincent-desktop:~/tci6638-linux-kernel/linux-keystone$ git reset --hard DEV.MCSDK-03.08.04.10
HEAD is now at a5fb562 uio: increasing MAX_UIO_MAPS to 7
```

3.2 Build the TCI6638 Linux kernel

In the git folder, we need to choose and build the target first by using:

```
make keystone2_defconfig
```

The supported targets can be found in path of /linux-keystone/arch/arm/configs/



Name	Size	Type	Date Modified
jornada720_defconfig	2.7 KB	plain text document	Sunday, June 09, 2013 PM12:32:55 Hk
keystone1_defconfig	4.0 KB	plain text document	Sunday, June 09, 2013 PM12:32:55 Hk
keystone2_defconfig	6.1 KB	plain text document	Sunday, June 09, 2013 PM12:32:55 Hk
keystone2_fullrt_def...	6.1 KB	plain text document	Sunday, June 09, 2013 PM12:32:55 Hk

You'll see the following info:

```
vincent@vincent-desktop:~/tci6638-linux-kernel/linux-keystone$ make keystone2_defconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/zconf.lex.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTLD scripts/kconfig/conf
#
# configuration written to .config
#
```

Then we build the “uImage” by using:

```
make uImage
```

You'll see the following info:

```
vincent@vincent-desktop:~/tci6638-linux-kernel/linux-keystone$ make uImage
scripts/kconfig/conf --silentoldconfig Kconfig
WRAP arch/arm/include/generated/asm/auxvec.h
WRAP arch/arm/include/generated/asm/bitperlong.h
WRAP arch/arm/include/generated/asm/cputime.h
WRAP arch/arm/include/generated/asm/current.h
```

It will take a while to build the Linux kernel “uImage”. You will see the following after build:

```
LD arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready
UIMAGE arch/arm/boot/uImage
Image Name: Linux-3.8.4
Created: Wed Jun 12 11:22:06 2013
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 3527088 Bytes = 3444.42 kB = 3.36 MB
Load Address: 0x80008000
Entry Point: 0x80008000
Image arch/arm/boot/uImage is ready
vincent@vincent-desktop:~/tci6638-linux-kernel/linux-keystone$
```

You can choose to build “Image”, “ulmage” and “zImage” for different usage scenarios:

Image: the built original Linux kernel image

ulmage: the built image which uses co-works with u-boot

zImage: the built zipped Linux kernel image

Then build the .dtb file which will be used on EVM by using

```
make k2hk-evm.dtb
```

You can find the supported DTS files from the path of /linux-keystone/arch/arm/boot/dts/:

◀

 vincent

tc16638-linux-kernel

linux-keystone

arch

arm

boot

dts

Name	Size	Type	Date Modified
 integratorcp.dts	2.2 KB	C source code	Sunday, June 09, 2013 PM12:32:55 H
 k2hk-evm.dts	60.6 KB	plain text document	Wednesday, June 12, 2013 AM11:01:
 keystone-sim.dts	6.9 KB	plain text document	Sunday, June 09, 2013 PM12:32:55 H

Then you’ll see the following info:

```
vincent@vincent-desktop:~/tc16638-linux-kernel/linux-keystone$ make k2hk-evm.dtb
DTC      arch/arm/boot/dts/k2hk-evm.dtb
vincent@vincent-desktop:~/tc16638-linux-kernel/linux-keystone$
```

4. Run U-boot on TCI6638 EVM with CCSv5

4.1 Install TCI6638 EVM USB to COM Driver and Hardware Configuration

Due to the TCI6638 EVM have the ftdi chip to convert the COM to USB interface, we need to download and install the ftdi drivers on the PC side to use the COM port. You should download and install the driver from the following link:

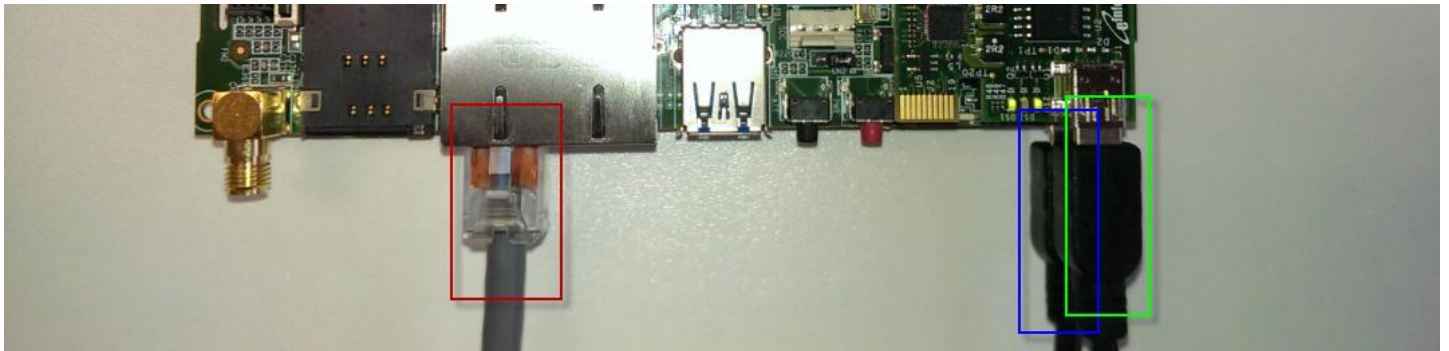
<http://www.ftdichip.com/Drivers/D2XX.htm>



Operating System	Release Date	Processor Architecture							Comments
		x86 (32-bit)	x64 (64-bit)	PPC	ARM	MIPSII	MIPSIV	SH4	
Windows*	2013-02-20	2.08.28	2.08.28	-	-	-	-	-	2.08.28 WHQL Certified Available as setup executable Release Notes
Linux	2012-06-29	1.1.12	1.1.12	-	1.1.12 Suitable for Raspberry Pi	-	-	-	ReadMe

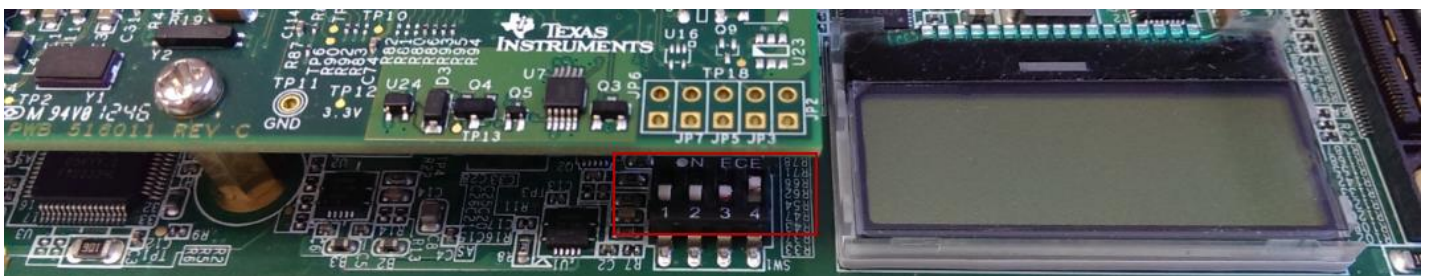
Choose the latest Windows version to download and install, e.g: “CDM 2.08.28 WHQL Certified.7z”

Connect the USB and LAN cable before power on the EVM board:



There are 2 LAN ports on the EVM, make sure the LAN cable connect with the left one, there are also 2 mini-USB ports, the one on the mezzanine board is the USB XDS200 Emulator cable, the one on the main EVM board is the USB-COM cable. Make sure the LAN cable connect in the same subnet with your PC and also connect 2 USB cables on your PC.

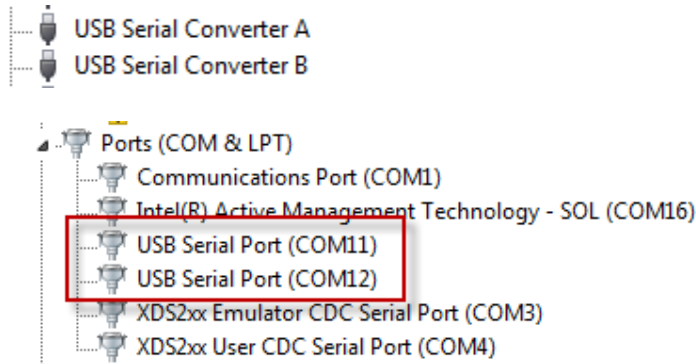
Before we power on the EVM, please make sure the boot DIP pins on EVM in the position like:



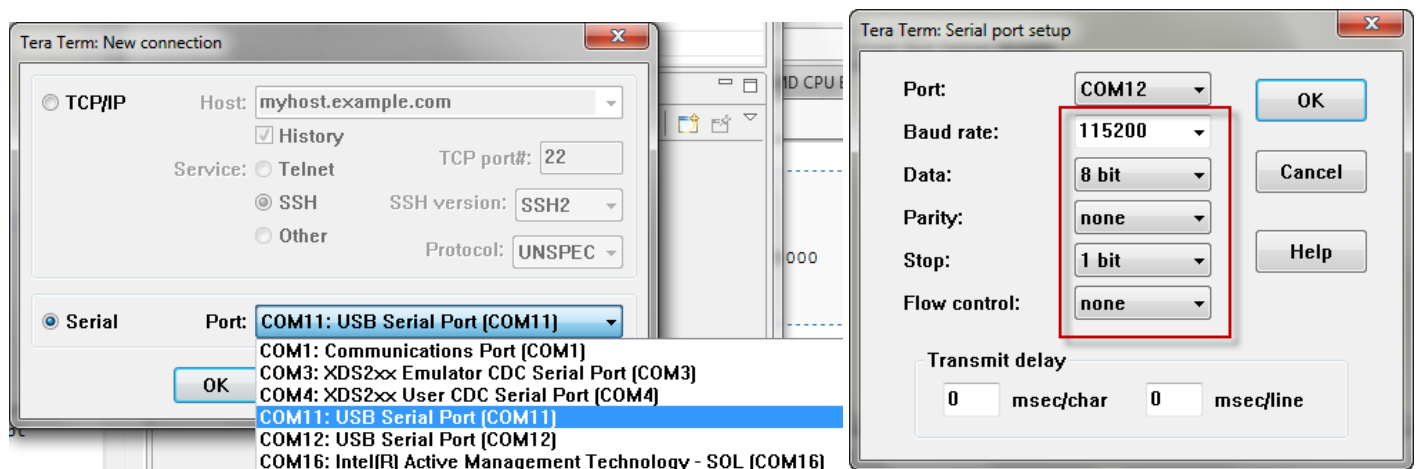
This makes the board in “no boot” mode for running U-boot through CCS.

Power on the EVM and the driver of the Emulator and USB-COM will be installed on your PC.

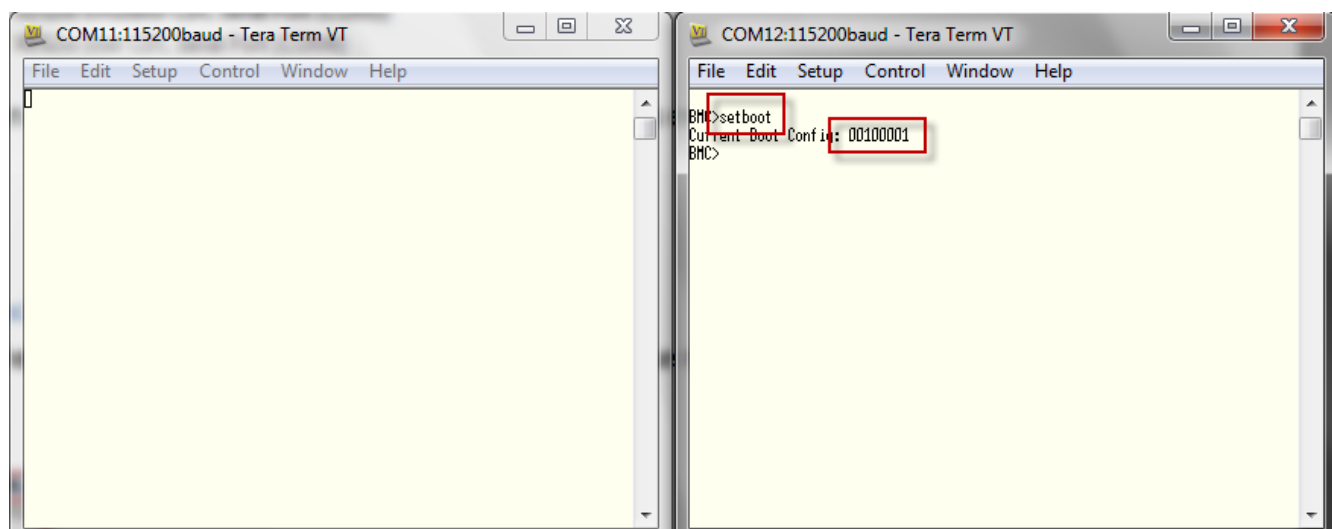
If the USB-COM drivers are correctly installed, you should find 2 more items were showed in your device manager:



Then open the hyper-terminal tool (we use TeraTerm as example) to create 2 sessions on your PC:



After we created 2 sessions, you will see the terminals such as:



The one which started with “BMC>” is the control port to configure the boot mode pins of TCI6638, the most used command of BMC is listed as follow:

`setboot`: print the current boot mode

`setboot 00100001` or `00112005`: set the boot mode, 00100001: no boot (for running u-boot on CCS); 00112005: SPI flash boot (for booting u-boot on SPI flash)

`fullrst`: full reset to make the boot mode change take effect.

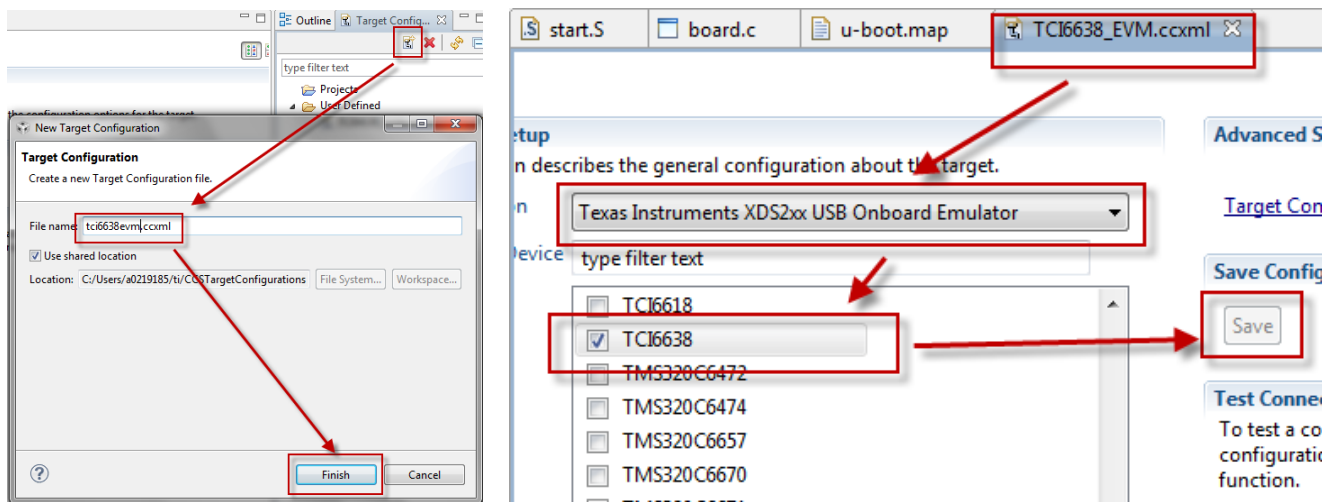
Please note that the DIP switch only takes effect on the first power on when connecting the power cable, other boot modes can be decided by the “setboot” commands on BMC and take effect after the command “fullrst”.

If you change the boot mode by using the BMC command, the boot mode will change back to the value of the DIP switch after you disconnect then connect the power cable.

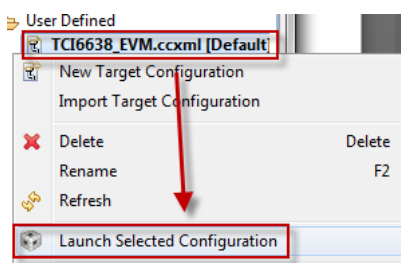
For more information about the boot mode DIP switch and the BMC encoding, please refer to the section “Appendix”

4.2 Create and Launch Debug Session and Connect Target

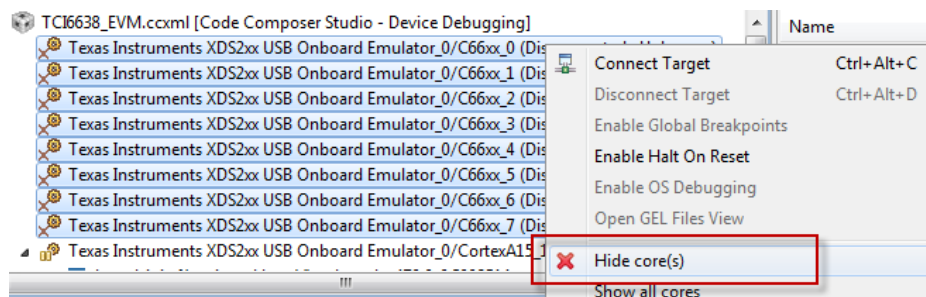
Create the .ccxml file in CCS as the follow steps:



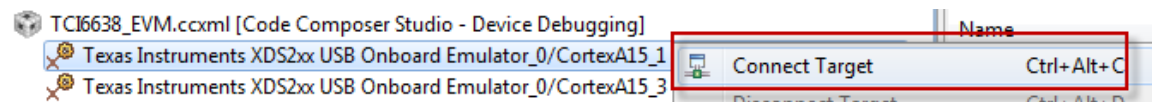
Launch the TCI6638 EVM debug session on CCSv5, please make sure do not use any ARM core GEL files.



After launched the debug session, please hide the unused cores to give you a clear view:

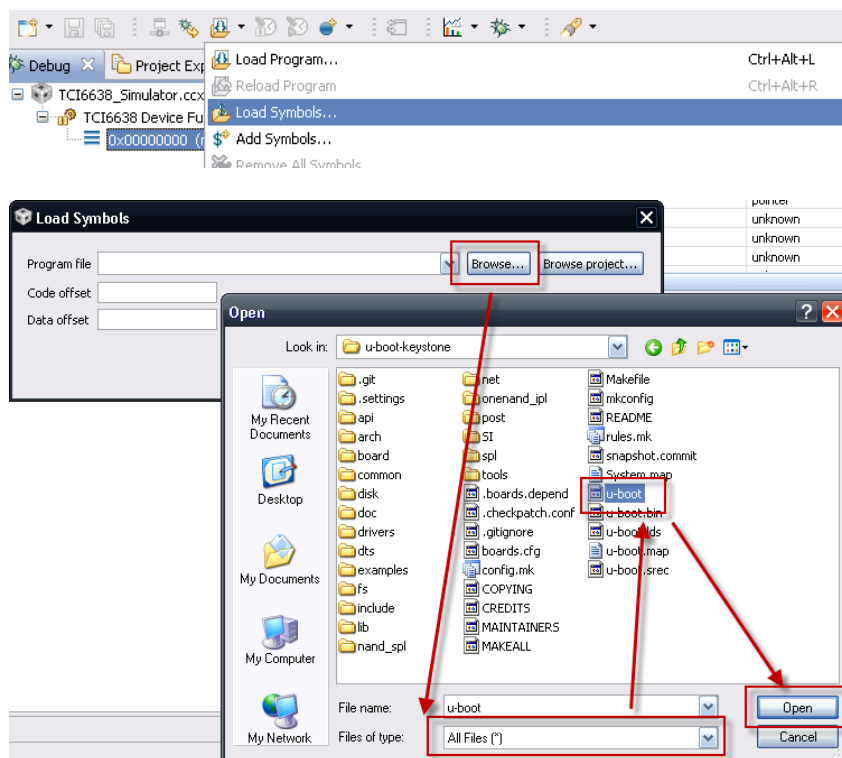


Then connect to the first ARM Core:



4.3 Load Image on the Target and Run

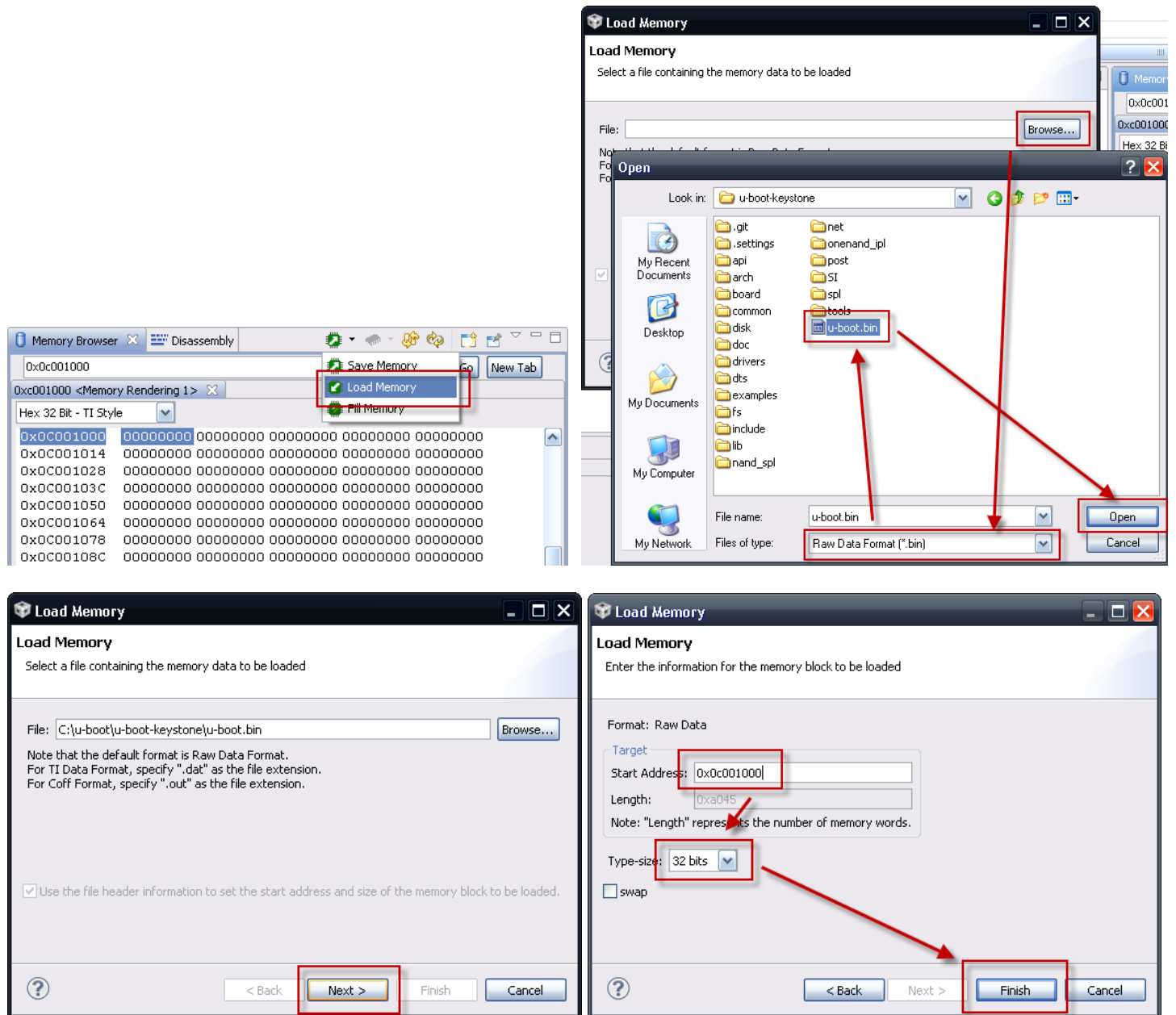
Launch the Load the file with no extension “u-boot” on target which can let the CCS recognize the symbol table:



After you load the symbol table, some of the symbols/functions which defined in U-boot image can be recognized by CCS:

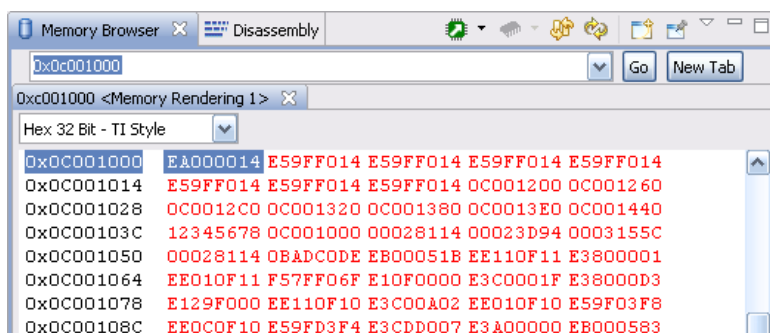
Expression	Type	Value	Address
board_init_f	void (*)(long unsigned int...	0x0C002480	
relocate_code	void *	0x0C001084	
board_init_r	void (*)(global_data*, long...	0x0C0023E0	
+ Add new expression			

Then manually load U-boot bin file on RAM of SoC:

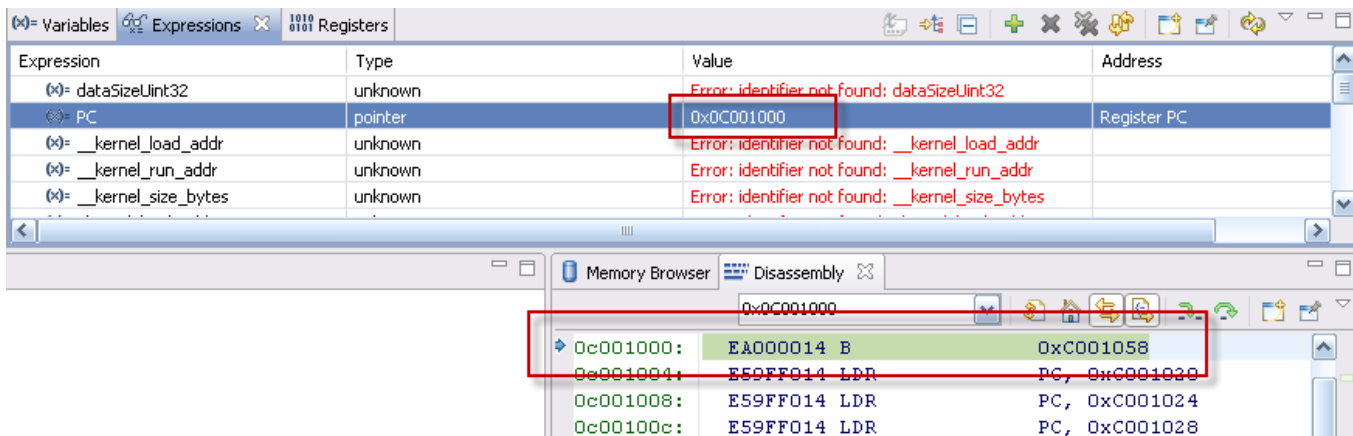


Then choose the load address "0x0C00_1000" (this address is the default relocate address which on RAM of SoC) and choose the 32-bit load mode.

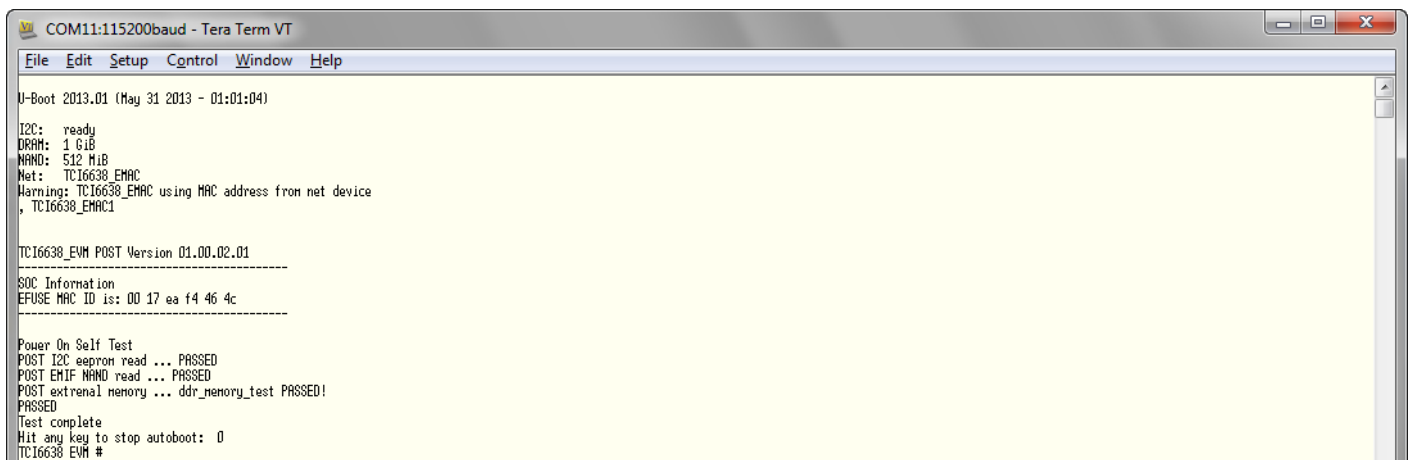
After U-boot.bin was loaded, you will see the data filled in memory browser:



Add PC pointer in expression view and force PC pointer point to “0x0C00_1000”, then the pointer in disassembly view will switch to the relative assembly code:



Then click the “Resume” button in CCS to run the u-boot, you will see the printed info similar as follow:



Please note that in the MCSDK Alpha11 (GA), there is the POST test executes during the u-boot booting procedures including the DDR tests by default, which need about 1-2 minutes for the booting procedures, please be patient to wait for the boot procedures.

If you don't want to do the POST test, you can pause at the U-boot auto prompt point and input the following U-boot commands to disable it:

```
setenv no_post 1
```

```
saveenv
```

This will take effect in the next U-boot booting procedures.

5. Burn and Run U-boot on TCI6638 EVM SPI Flash

5.1 Burn U-boot Image on TCI6638 EVM SPI Flash

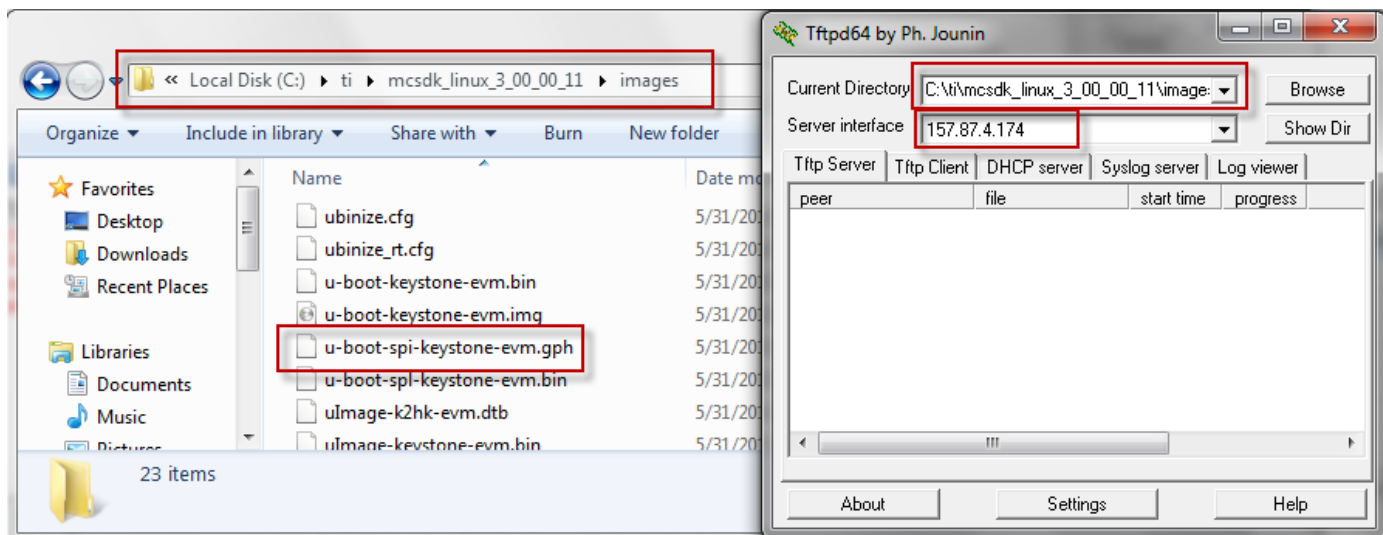
Before execute any steps in this section, please make sure that the U-boot already running on TCI6638 EVM.

We also need to download the tftp software and install it the following path:

http://tftpd32.jounin.net/tftpd32_download.html

In this step, we will burn the .gph file on the SPI flash, the .gph file includes the U-boot image and the SPI boot parameter header which generated by the steps describes in section 2.2.

Get the .gph file path and your PC IP address and configure the TFTP tools:



Please note that the LAN cable connected with EVM must be in the same subnet with your TFTP server (or PC with TFTP software), this keeps the EVM obtains the correct IP address with DHCP and can get the images from the TFTP server (or PC with TFTP software).

We use the PC with TFTP software as the example in this document, the IP address in the TFTP software configuration should be the IP address of the PC which installed the TFTP software.

If you don't want to modify anything, you can just use the prebuilt images and files which provided by MCSDK, these files are in path of `${MCSDK_LINUX_INSTALL_PATH}/images`.

If you want to add/remove some features in U-boot or Linux Kernel, you will need to use the built files which generated in the previous sections.

Either way is feasible and you need to make sure the files' name and path are correct with the U-boot environment variables and TFTP path configurations. We used the MCSDK prebuilt files as example in this document.

Now we can input the following command in the U-boot session:

```
setenv serverip <your tftp server IP set in the tftp software>
```

```
setenv name_uboot <the actual entire .gph file name>
```

```
saveenv
```

```
run get_uboot_net
```

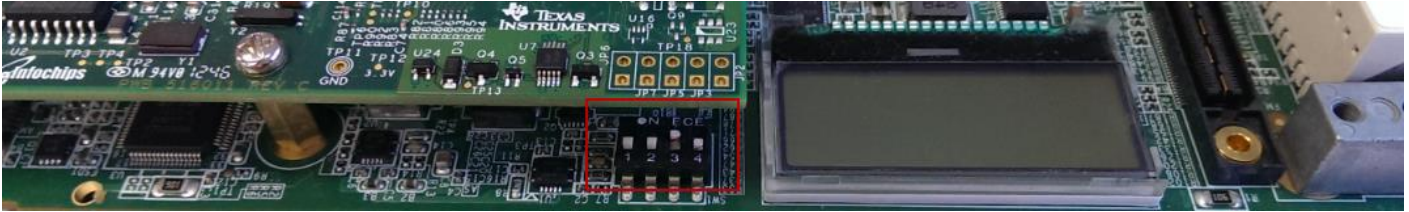
```
run burn_uboot
```

```
TCI6638 EVM # setenv serverip 157.87.4.174
TCI6638 EVM # setenv name_uboot 'u-boot-spi-keystone-evm.gph'
TCI6638 EVM # saveenv
Saving Environment to NAND...
Erasing NAND...
Erasing at 0x120000 -- 100% complete.
Writing to NAND... done
TCI6638 EVM # run get_uboot_net
BOOTP broadcast 1
*** Unhandled DHCP Option in OFFER/ACK: 44
*** Unhandled DHCP Option in OFFER/ACK: 46
*** Unhandled DHCP Option in OFFER/ACK: 150
*** Unhandled DHCP Option in OFFER/ACK: 44
*** Unhandled DHCP Option in OFFER/ACK: 46
*** Unhandled DHCP Option in OFFER/ACK: 150
DHCP client bound to address 157.87.4.143
Using TCI6638 EMAC device
TFTP from server 157.87.4.174; our IP address is 157.87.4.143
Filename 'u-boot-spi-keystone-evm.gph'.
Load address: 0x87000000
Loading: #####
          944.3 KiB/s
done
Bytes transferred = 383968 (5dba0 hex)
TCI6638 EVM # run burn_uboot
SF: Detected M25Q128A with page size 64 KiB, total 16 MiB
TCI6638 EVM #
```

After this step, the .gph image is burnt on the TCI6638 EVM on-board SPI Flash.

5.2 Boot U-boot Image on TCI6638 EVM SPI Flash

After the burn the image on the flash, please power off the EVM and change the boot mode pins as follow:



Then power on the EVM, you can see the ARM core can boot u-boot image from the SPI flash:

```
COM11:115200baud - Tera Term VT
File Edit Setup Control Window Help

U-Boot 2013.01 (May 31 2013 - 01:01:04)

I2C: ready
DRAM: 1 GiB
NAND: 512 MiB
Net: TCI6638_EMAC
Warning: TCI6638_EMAC using MAC address from net device
       TCI6638_EMAC1

TCI6638_EVM POST Version 01.00.02.01
-----
SOC Information
EFUSE MAC ID is: 00 17 ea f4 46 4c
-----

Power On Self Test
POST I2C eeprom read ... PASSED
POST EMIF NAND read ... PASSED
POST extrenal memory ... ddr_memory_test PASSED!
PASSED
Test complete
Hit any key to stop autoboot: 0
TCI6638 EVM #
```

6. Load and Run Linux kernel on TCI6638 EVM – RAM File System

6.1 U-boot Environment Settings for Running Linux Kernel with RAM File System

Before we boot the Linux kernel, we need to make sure the U-boot environment variables' settings are the same with the actual Linux kernel related files you want to use.

If you don't want to modify anything, you can just use the prebuilt images and files which provided by MCSDK, these files are in path of `${MCSDK_LINUX_INSTALL_PATH}/images`.

If you want to add/remove some features in U-boot or Linux Kernel, you will need to use the built files which generated in the previous sections.

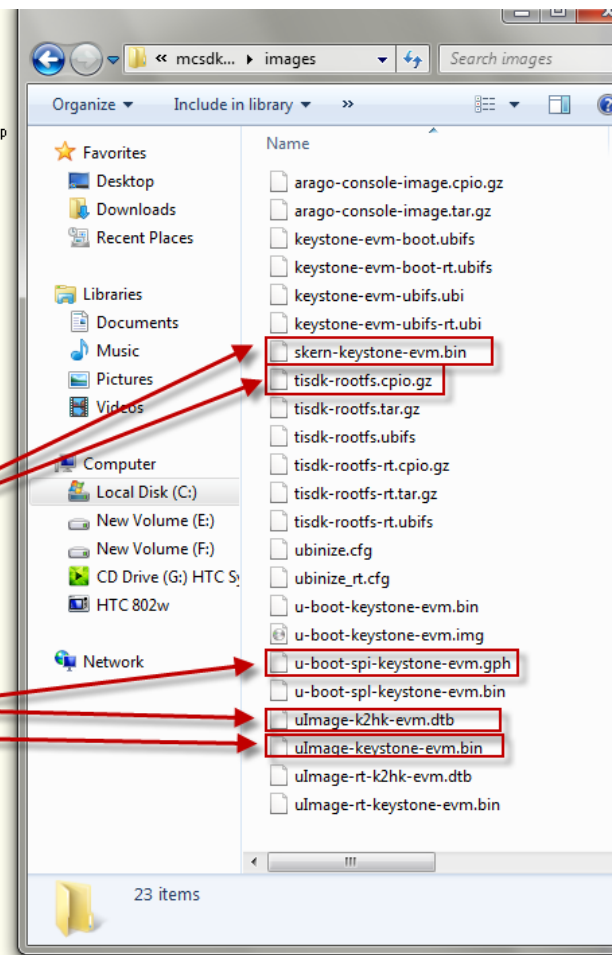
Either way is feasible and you need to make sure the files' name and path are correct with the U-boot environment variables and TFTP path configurations. We used the MCSDK prebuilt files as example in this document.

We can use the following command to check these environment variables:

```
env print
```

```
TCI6638 EVM # env print
addr_fdt=0x87000000
addr_fs=0x82000000
addr_kern=0x88000000
addr_mon=0x0c5f0000
addr_uboot=0x87000000
args_all=setenv bootargs console=ttyS0,115200n8 rootwait=1
args_net=setenv bootargs ${bootargs} rootfstype=nfs root=/dev/nfs ru nfsroot=${serverip}:${nfs_root},${nfs_options} ip=dhcp
args_ranfs=setenv bootargs ${bootargs} earlyprintk rdinit=/sbin/init ru root=/dev/ram0 initrd=0x80200000,76h
args_ubi=setenv bootargs ${bootargs} rootfstype=ubifs root=ubi0:rootfs rootflags=sync ru ubi.mtd=2,2048
baudrate=115200
boot=ranfs
bootcmd=run init_${boot} get_fdt_${boot} get_mon_${boot} get_kern_${boot} run_mon run_kern
bootdelay=3
bootfile=u-boot-spi-keystone-evm.gph
burn_uboot=sf probe; sf erase 0 0x100000; sf write ${addr_uboot} 0 ${filesize}
dhcpip=157.87.32.42
dnssip2=157.87.119.10
ethact=TCI6638_EMAC
ethaddr=00:17:ea:f4:46:4c
fdt_high=0xffffffff
fileaddr=87000000
filesize=50BED
gatewayip=157.87.4.1
get_fdt_net=dhcp ${addr_fdt} ${tftp_root}/${name_fdt}
get_fdt_ranfs=dhcp ${addr_fdt} ${tftp_root}/${name_fdt}
get_fdt_ubi=ubifsload ${addr_fdt} ${name_fdt}
get_fs_ranfs=dhcp ${addr_fs} ${tftp_root}/${name_fs}
get_kern_net=dhcp ${addr_kern} ${tftp_root}/${name_kern}
get_kern_ranfs=dhcp ${addr_kern} ${tftp_root}/${name_kern}
get_kern_ubi=ubifsload ${addr_kern} ${name_kern}
get_mon_net=dhcp ${addr_mon} ${tftp_root}/${name_mon}
get_mon_ranfs=dhcp ${addr_mon} ${tftp_root}/${name_mon}
get_mon_ubi=ubifsload ${addr_mon} ${name_mon}
get_uboot_net=dhcp ${addr_uboot} ${tftp_root}/${name_uboot}
get_uboot_ranfs=dhcp ${addr_uboot} ${tftp_root}/${name_uboot}
init_net=run args_all args_net
init_ranfs=run args_all args_ranfs get_fs_ranfs
init_ubi=run args_ubi; ubi part ubifs; ubi mount boot
ipaddr=157.87.4.143
mem_lpae=1
mem_reserved=512M
name_fdt=uImage-tci6638-evm.dtb
name_fs=tisdk-rootfs.cpio.gz
name_kern=uImage-keystone-evm.bin
name_mon=skern-keystone-evm.bin
name_uboot=u-boot-spi-keystone-evm.gph
netmask=255.255.255.0
nfs_options=v3,tcp,rsize=4096,usize=4096
nfs_root=/export
run_kern=bootin ${addr_kern} - ${addr_fdt}
run_mon=non_install ${addr_mon}
serverip=157.87.4.174
stderr=serial
stdin=serial
stdout=serial
ver=U-Boot 2013.01 (May 31 2013 - 01:01:04)

Environment size: 2289/262140 bytes
```

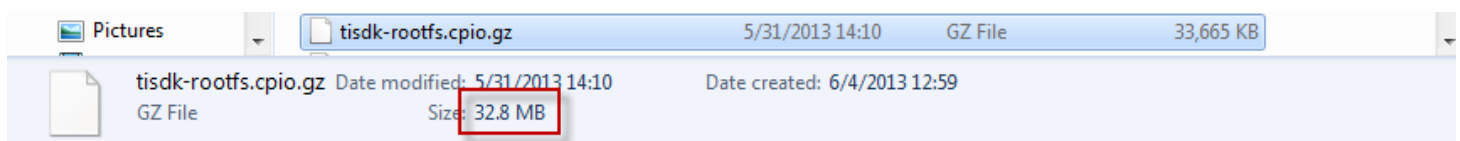


It'll print all the environment variables which current u-boot recorded; we need to modify the unmatched item to the corresponding values, such as:

```
setenv name_fdt 'uImage-k2hk-evm.dtb'
```

Please note that the default value of “args_ramfs” is “args_ramfs=setenv bootargs \${bootargs} earlyprintk rdinit=/sbin/init rw root=/dev/ram0 initrd=0x80200000,9M”, the “9M” here means the file system size, we need to modify it to the actual file system size ceiling for MByte, e.g.: the actual file system size is 32.8MByte, so we modify this value to 33M by using:

```
setenv args_ramfs 'setenv bootargs ${bootargs} earlyprintk  
rdinit=/sbin/init rw root=/dev/ram0 initrd=0x80200000,33M'
```



After the modification, we can use this command to save the environment variables back to Flash

```
saveenv
```

```
TC16638 EVN # saveenv  
Saving Environment to NAND...  
Erasing NAND...  
Erasing at 0x120000 -- 100% complete.  
Writing to NAND... done  
TC16638 EVN #
```

6.2 Load and Run Linux Kernel by U-boot with RAM File System

After configuring U-boot environment variables, we can use U-boot to load and run Linux Kernel, we use the “ramfs” mode for the procedures.

We can use 2 ways to boot the Linux Kernel:

- 1) Automatic boot from U-boot. If the U-boot environment variables configures correctly, U-boot can boot Linux Kernel Automatically.
- 2) Input any key to stop the U-boot when it prompts then manually boot Linux Kernel. This way usually used for debugging Linux Kernel.

You can see the similar print info for the typical SMP Linux Kernel system booting procedures:

```
U-Boot SPL 2013.01 (May 31 2013 - 01:01:04)
```

```
SF: Detected N25Q128A with page size 64 KiB, total 16 MiB
```

```
I2C:    ready
DRAM:   1 GiB
NAND:   512 MiB
Net:    TCI6638 EMAC, TCI6638 EMAC1
```

```
SOC Information
EFUSE MAC ID is: 00 17 ea f4 46 4c
```

```
Test complete
Hit any key to stop autoboot: 0
BOOTP broadcast 1
```

```
*** Unhandled DHCP Option in OFFER/ACK: 44
*** Unhandled DHCP Option in OFFER/ACK: 46
*** Unhandled DHCP Option in OFFER/ACK: 150
*** Unhandled DHCP Option in OFFER/ACK: 44
*** Unhandled DHCP Option in OFFER/ACK: 46
*** Unhandled DHCP Option in OFFER/ACK: 150
```

```
DHCP client bound to address 157.87.4.143
Using TCI6638_EMAC device
TFTP from server 157.87.4.174; our IP address is 157.87.4.143
Filename '/tisdk-rootfs.cpio.gz'.
Load address: 0x82000000
```

[illegible]


```
#####
#####
#####
908.2 KiB/s
```

done

Bytes transferred = 3529136 (35d9b0 hex)

installed monitor, freq [133120000], status 133120000

Booting kernel from Legacy Image at 88000000 ...

Image Name: Linux-3.8.4

Created: 2013-05-31 4:01:23 UTC

Image Type: ARM Linux Kernel Image (uncompressed)

Data Size: 3529072 Bytes = 3.4 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 87010508

Starting kernel ...

```
[ 0.000000] Booting Linux on physical CPU 0x0
[ 0.000000] Linux version 3.8.4 (gtbldadm@ubuntu-12) (gcc version 4.7.3 20130226
(prerelease) (crosstool-NG linaro-1.13.1-4.7-2013.03-20130313 - Linaro GCC 2013.03) ) #1
SMP Fri May 31 00:00:49 EDT 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 @c0af4000 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 earlyprintk
rdinit=/sbin/init rw root=/dev/ram0 initrd=0x802000000,33M
[ 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: 462264k/462264k available, 62024k reserved, 0K highmem
[ 0.000000] Virtual kernel memory layout:
[ 0.000000] vector : 0xfffff0000 - 0xfffff1000 ( 4 kB)
[ 0.000000] fixmap : 0xffff00000 - 0xffffe0000 ( 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 - 0xc06320e8 (6313 kB)
[ 0.000000] .init : 0xc0633000 - 0xc0673d80 ( 260 kB)
[ 0.000000] .data : 0xc0674000 - 0xc06b0d08 ( 244 kB)
[ 0.000000] .bss : 0xc06b0d08 - 0xc06db39c ( 170 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 798720000, postdiv = 2, mult = 12,prediv = 0
```

```

[    0.000000] pll_clk parent_rate(122880000 Hz), rate(327680000 Hz), postdiv = 6, mult =
15, prediv = 0
[    0.000000] tci6614-timer: no matching node
[    0.000000] Architected local timer running at 133.12MHz (phys).
[    0.000000] Switching to timer-based delay loop
[    0.000000] sched_clock: 32 bits at 133MHz, resolution 7ns, wraps every 32263ms
[    0.000000] Console: colour dummy device 80x30
[    0.000106] Calibrating delay loop (skipped), value calculated using timer frequency..
266.24 BogoMIPS (lpj=1331200)
[    0.000123] pid_max: default: 4096 minimum: 301
[    0.000357] Mount-cache hash table entries: 512
[    0.013446] CPU: Testing write buffer coherency: ok
[    0.013725] CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
[    0.013762] Setting up static identity map for 0x8047c8a0 - 0x8047c8d4
[    0.101124] CPU1: Booted secondary processor
[    0.101146] CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
[    0.187955] CPU2: Booted secondary processor
[    0.187976] CPU2: thread -1, cpu 2, socket 0, mpidr 80000002
[    0.274770] CPU3: Booted secondary processor
[    0.274792] CPU3: thread -1, cpu 3, socket 0, mpidr 80000003
[    0.274859] Brought up 4 CPUs
[    0.274894] SMP: Total of 4 processors activated (1064.96 BogoMIPS).
[    0.298152] NET: Registered protocol family 16
[    0.299918] DMA: preallocated 256 KiB pool for atomic coherent allocations
[    0.315505] keystone2_pcie_serdes_setup
[    0.317609] keystone2_pcie_serdes_setup done
[    0.317639] hw-breakpoint: found 5 (+1 reserved) breakpoint and 4 watchpoint registers.
[    0.317649] hw-breakpoint: maximum watchpoint size is 8 bytes.
[    0.336505] bio: create slab <bio-0> at 0
[    0.336815] keystone-pcie: keystone_pcie_rc_init - start
[    0.337130] MEM 0x0000000050000000..0x000000005fffffff -> 0x0000000050000000
[    0.337147] IO 0x0000000024000000..0x0000000024003fff -> 0x0000000000000000
[    0.337206] pcie - number of legacy irqs = 4
[    0.337289] pcie - number of MSI host irqs = 8, msi_irqs = 32
[    0.451475] keystone-pcie: Doing PCI Setup...Done
[    0.451485] keystone-pcie: Starting PCI scan...
[    0.451710] PCI host bridge to bus 0000:00
[    0.451731] pci_bus 0000:00: root bus resource [mem 0x50000000-0x5fffffff]
[    0.451746] pci_bus 0000:00: root bus resource [io 0x0000-0x3fff]
[    0.451760] pci_bus 0000:00: No busn resource found for root bus, will use [bus 00-ff]
[    0.451838] PCI: bus0: Fast back to back transfers enabled
[    0.451867] keystone-pcie: Ending PCI scan...
[    0.451880] keystone-pcie: keystone_pcie_rc_init - end
[    0.452201] vgaarb: loaded
[    0.452779] SCSI subsystem initialized
[    0.453490] usbcore: registered new interface driver usbfs
[    0.453666] usbcore: registered new interface driver hub
[    0.453855] usbcore: registered new device driver usb
[    0.455844] keystone-hwqueue hwqueue.2: qmgr start queue 0, number of queues 8192
[    0.455997] keystone-hwqueue hwqueue.2: 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
[    0.456015] keystone-hwqueue hwqueue.2: qmgr start queue 8192, number of queues 8192
[    0.456169] keystone-hwqueue hwqueue.2: 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
[    0.457322] keystone-hwqueue hwqueue.2: qos: sched port @8096, drop sched @8000
[    0.458918] keystone-hwqueue hwqueue.2: qos: sched port @6496, drop sched @6400
[    0.460495] keystone-hwqueue hwqueue.2: added pool pool-net: 2048 descriptors of size

```

```

[ 0.460514] keystone-hwqueue hwqueue.2: added pool pool-rio: 128 descriptors of size
256
[ 0.460530] keystone-hwqueue hwqueue.2: added pool pool-udma: 1636 descriptors of size
256
[ 0.460546] keystone-hwqueue hwqueue.2: added pool pool-xge: 2048 descriptors of size
128
[ 0.463934] keystone-hwqueue hwqueue.2: registered queues 0-16383
[ 0.464041] keystone-hwqueue hwqueue.2: qos version 0x2000105, magic valid
[ 0.464691] keystone-hwqueue hwqueue.2: qos version 0x2000105, magic valid
[ 0.475806] keystone-pktdma 2004000.pktdma: registered 20 logical channels, flows 32,
tx chans: 9, rx chans: 24
[ 0.480984] keystone-pktdma 2a08000.pktdma: registered 24 logical channels, flows 32,
tx chans: 32, rx chans: 32, loopback
[ 0.482007] keystone-pktdma 2fa1000.pktdma: registered 4 logical channels, flows 32,
tx chans: 16, rx chans: 16
[ 0.482252] Switching to clocksource arch_sys_counter
[ 0.506707] NET: Registered protocol family 2
[ 0.507320] TCP established hash table entries: 4096 (order: 3, 32768 bytes)
[ 0.507433] TCP bind hash table entries: 4096 (order: 3, 32768 bytes)
[ 0.507543] TCP: Hash tables configured (established 4096 bind 4096)
[ 0.507586] TCP: reno registered
[ 0.507602] UDP hash table entries: 256 (order: 1, 8192 bytes)
[ 0.507632] UDP-Lite hash table entries: 256 (order: 1, 8192 bytes)
[ 0.507910] NET: Registered protocol family 1
[ 0.508152] RPC: Registered named UNIX socket transport module.
[ 0.508163] RPC: Registered udp transport module.
[ 0.508173] RPC: Registered tcp transport module.
[ 0.508183] RPC: Registered tcp NFSv4.1 backchannel transport module.
[ 0.508383] Unpacking initramfs...
[ 3.450286] Initramfs unpacking failed: junk in compressed archive
[ 3.478281] Freeing initrd memory: 33792K
[ 3.625810] Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
[ 3.626271] NTFS driver 2.1.30 [Flags: R/O].
[ 3.626786] jffs2: version 2.2. (NAND) © 2001-2006 Red Hat, Inc.
[ 3.628930] NET: Registered protocol family 38
[ 3.629332] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 254)
[ 3.629345] io scheduler noop registered
[ 3.629356] io scheduler deadline registered
[ 3.629614] io scheduler cfq registered (default)
[ 3.631686] keystone-udma udma0.3: registered udma device udma0
[ 3.711112] Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
[ 3.713650] 2530c00.serial: ttyS0 at MMIO 0x2530c00 (irq = 309) is a 16550A
[ 4.466437] console [ttyS0] enabled
[ 4.470820] 2531000.serial: ttyS1 at MMIO 0x2531000 (irq = 312) is a 16550A
[ 4.483599] loop: module loaded
[ 4.486983] at24 0-0050: 131072 byte 24c1024 EEPROM, writable, 1 bytes/write
[ 4.495564] Generic platform RAM MTD, (c) 2004 Simtec Electronics
[ 4.512965] ONFI param page 0 valid
[ 4.516442] ONFI flash detected
[ 4.519577] NAND device: Manufacturer ID: 0x2c, Chip ID: 0xac (Micron
MT29F4G08ABDBAHC), 512MiB, page size: 2048, OOB size: 64
[ 4.532666] Bad block table found at page 262080, version 0x01
[ 4.541750] Bad block table found at page 262016, version 0x01
[ 4.549324] 3 ofpart partitions found on MTD device 30000000.nand
[ 4.555407] Creating 3 MTD partitions on "30000000.nand":
[ 4.560790] 0x0000000000000-0x0000000100000 : "u-boot"
[ 4.566832] 0x0000000100000-0x0000000180000 : "params"
[ 4.572804] 0x0000000180000-0x0000000800000 : "ubifs"
[ 4.578752] davinci_nand 30000000.nand: controller rev. 2.5
[ 4.585174] spi_davinci 21000400.spi: master is unqueued, this is deprecated

```

```
[ 4.596733] m25p80 spi32766.0: found n25q128a11, expected n25q128
[ 4.602830] m25p80 spi32766.0: n25q128a11 (16384 Kbytes)
[ 4.608134] 2 ofpart partitions found on MTD device spi32766.0
[ 4.613956] Creating 2 MTD partitions on "spi32766.0":
[ 4.619079] 0x0000000000000-0x0000000080000 : "u-boot-spl"
[ 4.625422] 0x0000000080000-0x0000001000000 : "test"
[ 4.635230] spi_davinci 21000400.spi: Controller at 0xe0878400
[ 4.643081] keystone-netcp 2f00000.netcp: No streaming regs defined
[ 4.650891] keystone-netcp 2090000.netcp: Created interface "eth0"
[ 4.657086] keystone-netcp 2090000.netcp: dma_chan_name nettx0
[ 4.664240] keystone-netcp 2090000.netcp: Created interface "eth1"
[ 4.670409] keystone-netcp 2090000.netcp: dma_chan_name nettx1
[ 4.678098] keystone-dwc3 2690000.dwc: usbss revision 47914300
[ 4.683972] keystone-dwc3 2690000.dwc: mapped irq 425 to virq 608
[ 4.893717] xhci-hcd xhci-hcd: xHCI Host Controller
[ 4.898598] xhci-hcd xhci-hcd: new USB bus registered, assigned bus number 1
[ 4.906959] xhci-hcd xhci-hcd: irq 608, io mem 0x02690000
[ 4.912473] usb usb1: New USB device found, idVendor=1d6b, idProduct=0002
[ 4.919239] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 4.926450] usb usb1: Product: xHCI Host Controller
[ 4.931311] usb usb1: Manufacturer: Linux 3.8.4 xhci-hcd
[ 4.936614] usb usb1: SerialNumber: xhci-hcd
[ 4.941486] hub 1-0:1.0: USB hub found
[ 4.945258] hub 1-0:1.0: 1 port detected
[ 4.949465] xhci-hcd xhci-hcd: xHCI Host Controller
[ 4.954355] xhci-hcd xhci-hcd: new USB bus registered, assigned bus number 2
[ 4.961509] usb usb2: New USB device found, idVendor=1d6b, idProduct=0003
[ 4.968289] usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 4.975497] usb usb2: Product: xHCI Host Controller
[ 4.980358] usb usb2: Manufacturer: Linux 3.8.4 xhci-hcd
[ 4.985663] usb usb2: SerialNumber: xhci-hcd
[ 4.990506] hub 2-0:1.0: USB hub found
[ 4.994274] hub 2-0:1.0: 1 port detected
[ 4.998585] Initializing USB Mass Storage driver...
[ 5.003636] usbcore: registered new interface driver usb-storage
[ 5.009620] USB Mass Storage support registered.
[ 5.014585] mousedev: PS/2 mouse device common for all mice
[ 5.020486] i2c /dev entries driver
[ 5.024738] watchdog 22f0080.wdt: heartbeat 60 sec
[ 5.029914] keystone-crypto 20c0000.crypto: crypto accelerator enabled
[ 5.036992] usbcore: registered new interface driver usbhid
[ 5.042565] usbhid: USB HID core driver
[ 5.046926] remoteproc0: 2620040.dsp0 is available
[ 5.051788] remoteproc0: Note: remoteproc is still under development and considered
experimental.
[ 5.060731] remoteproc0: THE BINARY FORMAT IS NOT YET FINALIZED, and backward
compatibility isn't yet guaranteed.
[ 5.071097] remoteproc0: no firmware found
[ 5.075756] remoteproc1: 2620044.dsp1 is available
[ 5.080617] remoteproc1: Note: remoteproc is still under development and considered
experimental.
[ 5.089565] remoteproc1: THE BINARY FORMAT IS NOT YET FINALIZED, and backward
compatibility isn't yet guaranteed.
[ 5.099946] remoteproc1: no firmware found
[ 5.104565] remoteproc2: 2620048.dsp2 is available
[ 5.109426] remoteproc2: Note: remoteproc is still under development and considered
experimental.
[ 5.118373] remoteproc2: THE BINARY FORMAT IS NOT YET FINALIZED, and backward
compatibility isn't yet guaranteed.
[ 5.128732] remoteproc2: no firmware found
```

[5.133330] remoteproc3: 262004c.dsp3 is available
[5.138190] remoteproc3: Note: remoteproc is still under development and considered experimental.
[5.147138] remoteproc3: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[5.157497] remoteproc3: no firmware found
[5.162086] remoteproc4: 2620050.dsp4 is available
[5.166964] remoteproc4: Note: remoteproc is still under development and considered experimental.
[5.175902] remoteproc4: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[5.186259] remoteproc4: no firmware found
[5.190840] remoteproc5: 2620054.dsp5 is available
[5.195717] remoteproc5: Note: remoteproc is still under development and considered experimental.
[5.204654] remoteproc5: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[5.215014] remoteproc5: no firmware found
[5.219600] remoteproc6: 2620058.dsp6 is available
[5.224478] remoteproc6: Note: remoteproc is still under development and considered experimental.
[5.233414] remoteproc6: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[5.243773] remoteproc6: no firmware found
[5.248361] remoteproc7: 262005c.dsp7 is available
[5.253238] remoteproc7: Note: remoteproc is still under development and considered experimental.
[5.262164] remoteproc7: THE BINARY FORMAT IS NOT YET FINALIZED, and backward compatibility isn't yet guaranteed.
[5.272536] remoteproc7: no firmware found
[5.277214] oprofile: no performance counters
[5.281837] oprofile: using timer interrupt.
[5.286419] GACT probability on
[5.289552] Mirror/redirect action on
[5.293222] Simple TC action Loaded
[5.297404] netem: version 1.3
[5.300452] u32 classifier
[5.303159] Performance counters on
[5.306978] input device check on
[5.310623] Actions configured
[5.314031] Netfilter messages via NETLINK v0.30.
[5.318730] nf_conntrack version 0.5.0 (8006 buckets, 32024 max)
[5.325399] ctnetlink v0.93: registering with nfnetlink.
[5.331117] IPv4 over IPv4 tunneling driver
[5.335979] gre: GRE over IPv4 demultiplexor driver
[5.340839] ip_gre: GRE over IPv4 tunneling driver
[5.346492] ip_tables: (C) 2000-2006 Netfilter Core Team
[5.351930] ipt_CLUSTERIP: ClusterIP Version 0.8 loaded successfully
[5.358315] arp_tables: (C) 2002 David S. Miller
[5.362995] TCP: cubic registered
[5.366296] Initializing XFRM netlink socket
[5.371472] NET: Registered protocol family 10
[5.377025] NET: Registered protocol family 17
[5.381477] NET: Registered protocol family 15
[5.385996] Bridge firewalling registered
[5.389998] Ebtables v2.0 registered
[5.393666] 8021q: 802.1Q VLAN Support v1.8
[5.398698] sctp: Hash tables configured (established 16384 bind 16384)
[5.405476] NET: Registered protocol family 40
[5.410126] VFP support v0.3: implementor 41 architecture 4 part 30 variant f rev 0

```
[ 5.417801] Registering SWP/SWPB emulation handler
[ 5.429061] Freeing init memory: 256K
INIT: version 2.88 booting
Starting udev
Starting Bootlog daemon: bootlogd.
Configuring network interfaces... [ 7.236317] keystone-netcp 2090000.netcp:
initializing cpsw version 1.3 (1) SGMIID identification value 0x4ed1
[ 7.247439] keystone-netcp 2090000.netcp: Created a cpsw ale engine
[ 7.253814] keystone-netcp 2090000.netcp: initialized cpsw ale revision 1.3
[ 7.313467] keystone-netcp 2090000.netcp: Using Packet Accelerator Firmware version
0x01030008
[ 7.322065] keystone-netcp 2090000.netcp: pa_clk_rate(163840000
HZ),mult(25000),shift(12)
[ 7.341576] net eth0: netcp device eth0 opened
[ 7.348542] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 7.354367] 8021q: adding VLAN 0 to HW filter on device eth0
[ 7.360011] net eth0: adding rx vlan id: 0
[ 7.364194] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
udhcpd (v1.20.2) started
Sending discover...
Sending select for 157.87.4.143...
Lease of 157.87.4.143 obtained, lease time 28800
/etc/udhcpd.d/50default: Adding DNS 157.87.32.12
/etc/udhcpd.d/50default: Adding DNS 157.87.119.10
/etc/udhcpd.d/50default: Adding DNS 157.170.147.7
[ 7.906739] keystone-netcp 2090000.netcp: initializing cpsw version 1.3 (1) SGMIID
identification value 0x4ed1
[ 7.929799] net eth1: netcp device eth1 opened
[ 7.935193] IPv6: ADDRCONF(NETDEV_UP): eth1: link is not ready
[ 7.941008] 8021q: adding VLAN 0 to HW filter on device eth1
[ 7.946662] net eth1: adding rx vlan id: 0
udhcpd (v1.20.2) started
[ 8.012597] 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
Fri May 31 05:08:00 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-rltracert6.
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
AAAAB3NzaC1yc2EAAAADAQABAAQAg2RM75uXHDi+KkwLV2nQepQuXQuV9KbC+ZyNpdjbBRu1Y5tmqEX7FTsTA049
0grnuM8bMralYxy5vvvH5w24THU3M1eXofDtbP0vq5opw6/XUAcxnB3qk/gXeCl7VofJkwriRJPRHrBpvNkR0tR6H
lMz1GtKT2blxk0lORcEFvgUeU= root@keystone-evm
Fingerprint: md5 18:6d:cf:48:45:0e:ec:20:e5:b0:55:ab:ac:2b:fb:68
dropbear.
Starting mpmsrv daemon.
Starting telnet daemon.
#>>>>> LCD 12
IP Address:
157.87.4.143
Starting tiipclad daemon.
Starting syslog-ng:.
Starting thttpd.
Starting Lighttpd Web Server: lighttpd.
2013-05-31 05:08:01: (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:
```

7. Rebuild and Run Root File System on TCI6638 EVM

7.1 Modify and Rebuild Root File System Packages

Usually, we modify the root file system from the existing one, so we use the file system in MCSDK alpha11 (GA) as the example to show how to rebuild the root file system.

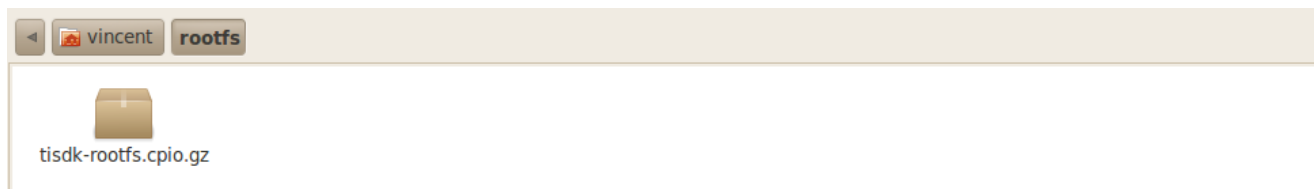
We can get the root file system package “tisdsk-rootfs.cpio.gz” in path of `${MCSDK_LINUX_INSTALL_PATH}/images`, and copy it in a created folder e.g. “rootfs” by using:

```
cd ~/
```

```
mkdir rootfs
```

```
cp tisdsk-rootfs.cpio.gz rootfs/
```

```
cd rootfs
```



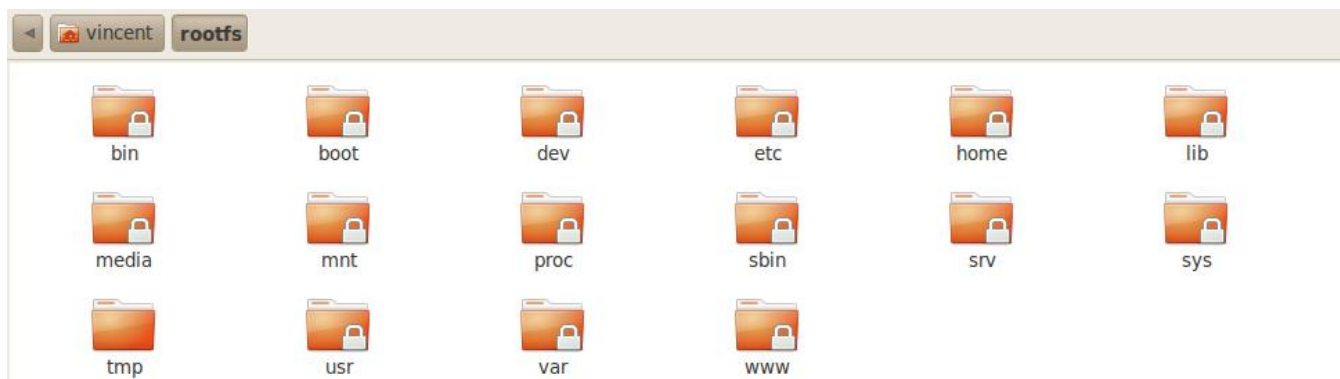
Use the following command to decompress the package:

```
sudo gunzip tisdsk-rootfs.cpio.gz
```

```
sudo cpio -idmv < tisdsk-rootfs.cpio
```

```
sudo rm tisdsk-rootfs.cpio
```

You'll see the following folders were uncompressed from the package:



We create folder named “12345678” and some files in this folder to show the modification:



Now we will rebuild the root file system by using:

```
sudo find . | sudo cpio -H new -o > ../rootfs.cpio
```

```
sudo gzip -c ../rootfs.cpio > ../rootfs.cpio.gz
```

You'll find these 2 files were created in your \$HOME folder

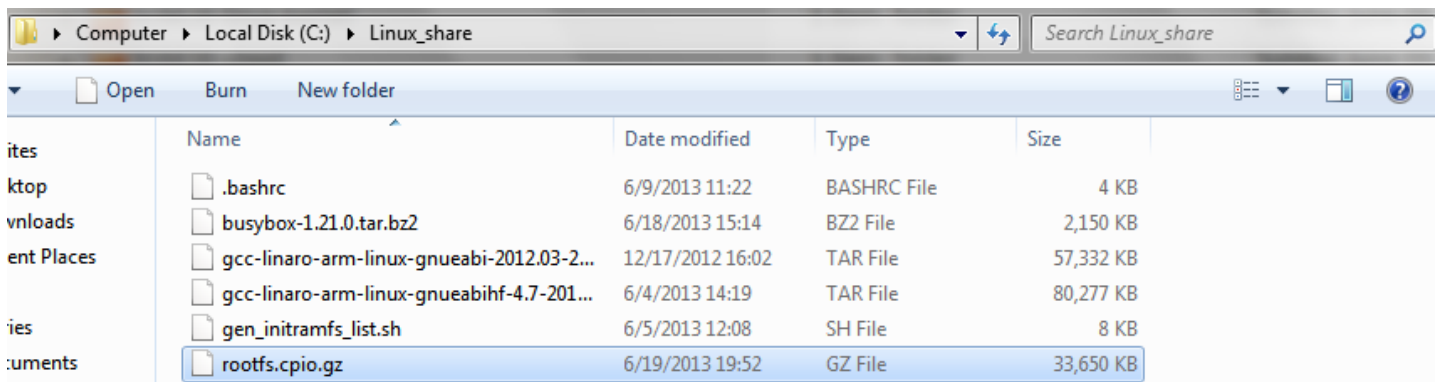
rootfs.cpio	84.0 MB	CPIO archive
rootfs.cpio.gz	32.9 MB	CPIO archive (gzip-compressed)

Copy it to the virtual machine shared folder which can be gotten in Windows environment by using:

```
cd ..
```

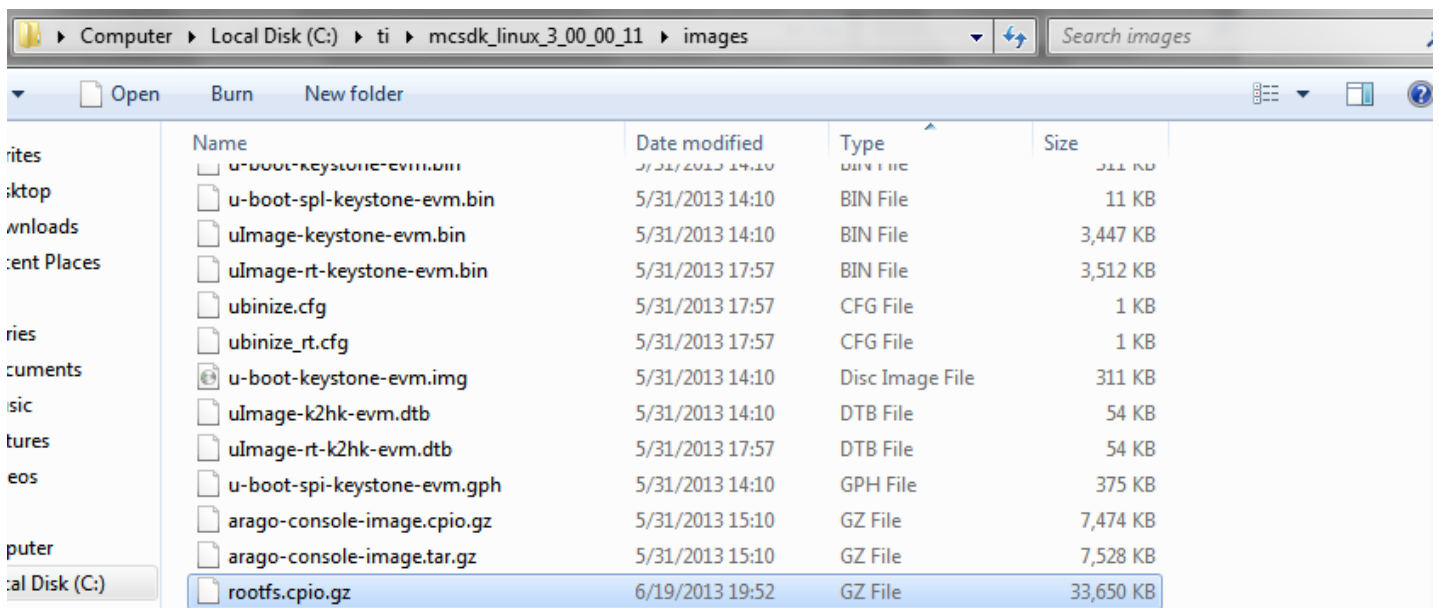
```
sudo cp rootfs.cpio.gz /media/sf_Linux_share/
```

You can get it in your Windows environment:



Name	Date modified	Type	Size
.bashrc	6/9/2013 11:22	BASHRC File	4 KB
busybox-1.21.0.tar.bz2	6/18/2013 15:14	BZ2 File	2,150 KB
gcc-linaro-arm-linux-gnueabi-2012.03-2...	12/17/2012 16:02	TAR File	57,332 KB
gcc-linaro-arm-linux-gnueabi-4.7-201...	6/4/2013 14:19	TAR File	80,277 KB
gen_initramfs_list.sh	6/5/2013 12:08	SH File	8 KB
rootfs.cpio.gz	6/19/2013 19:52	GZ File	33,650 KB

Copy the zipped file to the Windows environment which in the TFTP software configured path:



Name	Date modified	Type	Size
u-boot-keystone-evm.bin	5/31/2013 14:10	BIN File	311 KB
u-boot-spl-keystone-evm.bin	5/31/2013 14:10	BIN File	11 KB
ulmage-keystone-evm.bin	5/31/2013 14:10	BIN File	3,447 KB
ulmage-rt-keystone-evm.bin	5/31/2013 17:57	BIN File	3,512 KB
ubinize.cfg	5/31/2013 17:57	CFG File	1 KB
ubinize_rt.cfg	5/31/2013 17:57	CFG File	1 KB
u-boot-keystone-evm.img	5/31/2013 14:10	Disc Image File	311 KB
ulmage-k2hk-evm.dtb	5/31/2013 14:10	DTB File	54 KB
ulmage-rt-k2hk-evm.dtb	5/31/2013 17:57	DTB File	54 KB
u-boot-spi-keystone-evm.gph	5/31/2013 14:10	GPH File	375 KB
arago-console-image.cpio.gz	5/31/2013 15:10	GZ File	7,474 KB
arago-console-image.tar.gz	5/31/2013 15:10	GZ File	7,528 KB
rootfs.cpio.gz	6/19/2013 19:52	GZ File	33,650 KB

7.2 Boot and Test the Rebuilt Root File System on TCI6638EVM

Connect the cable on TCI6638 EVM and power on it, pause at the U-boot auto prompt point:

```
U-Boot SPL 2013.01 (May 31 2013 - 01:01:04)
SF: Detected M25Q128A with page size 64 KiB, total 16 MiB
```

```
U-Boot 2013.01 (May 31 2013 - 01:01:04)
```

```
I2C:   ready
DRAM:  1 GiB
NAND:  512 MiB
Net:    TCI6638_EMAC, TCI6638_EMAC1
```

```
TCI6638_EVM POST Version 01.00.02.01
```

```
-----
SOC Information
EFUSE MAC ID is: 00 17 ea f4 46 4c
-----
```

```
Power On Self Test
POST I2C eeprom read ... PASSED
POST EMIF NAND read ... PASSED
POST extrenal memory ... ddr_memory_test PASSED!
PASSED
Test complete
Hit any key to stop autoboot: 0
TCI6638 EVM #
```

Print out and modify the U-boot environment variables with the new root file system name and actual size (ceiling to MByte) then save it:

```
setenv name_fs 'rootfs.cpio.gz'
```

```
setenv args_ramfs 'setenv bootargs ${bootargs} earlyprintk
rdinit=/sbin/init rw root=/dev/ram0 initrd=0x80200000,34M'
```

```
setenv bootargs
```

```
saveenv
```

Reboot the EVM board by inputting the following commands in BMC windows:

```
fullrst
```



The press “Enter”, after the Linux Kernel booting, you’ll find the rebuilt root file system takes effect:

```
Arango Project http://arango-project.org keystone-evm ttyS0
Arango 2013.04 keystone-evm ttyS0
keystone-evm login: root
root@keystone-evm:~# cd ../../
root@keystone-evm:~# ls
12345678  debug   home    media   root     sys      var
bin       dev     ipaddr  mnt     sbin     tmp      www
boot     etc     lib     proc    srv      usr
root@keystone-evm:~# cd 12345678/
root@keystone-evm:12345678# ls
a.txt  b.c   c.h   d.dat  e.doc  f.hp3
root@keystone-evm:12345678#
```

8. Load and Run Linux kernel on TCI6638 EVM – UBI

8.1 U-boot Environment Settings for Running Linux Kernel with UBI

Before we boot the Linux kernel, we need to make sure the U-boot environment variables' settings are the same with the actual Linux kernel related files you want to use.

If you don't want to modify anything, you can just use the prebuilt images and files which provided by MCSDK, these files are in path of `${MCSDK_LINUX_INSTALL_PATH}/images`.

If you want to add/remove some features in U-boot or Linux Kernel, you will need to use the built files which generated in the previous sections.

Either way is feasible and you need to make sure the files' name and path are correct with the U-boot environment variables and TFTP path configurations. We used the MCSDK prebuilt files as example in this document.

If your used other boot methods (such as RAM file system) previously, you need to clean the previous U-boot setting which may have side effects for booting with UBI. You can use the following commands to clean the configuration to the default value then save the value:

```
env default -f -a
```

```
saveenv
```

We can use the following command to check these environment variables:

```
env print
```

It'll print all the environment variables which current u-boot recorded; we need to modify some items as follow:

1. If you want to skip the POST in U-boot, you should add:

```
setenv no_post 1
```

2. Change the boot method to UBI by using:

```
setenv boot 'ubi'
```

3. Set the TFTP Server IP which will be used to get the UBI file later:

```
setenv serverip '<your TFTP server IP>'
```

In this case, the server IP should be:

```
setenv serverip '157.87.4.174'
```

4. Modify the TFTP root path variable:

```
setenv tftp_root '<your TFTP root path>'
```

In this case, we don't need extra TFTP root path, the TFTP root path should be:

```
setenv tftp_root
```

5. Setup the MTD partition configurations:

```
setenv mtdparts 'mtdparts=  
davinci_nand.0:1024k(bootloader),512k(params)ro,129536k(ubifs)'
```

This configuration is the typical settings for the NAND flash on EVM, which contains 3 partitions:

- bootloader - Contains u-boot
- params - contains env variables
- ubifs - contains following UBI volumes:
 - boot volume - contains Kernel image (ulmage), device tree blob etc,
 - rootfs volume - contains the rootfs which is the primary filesystem

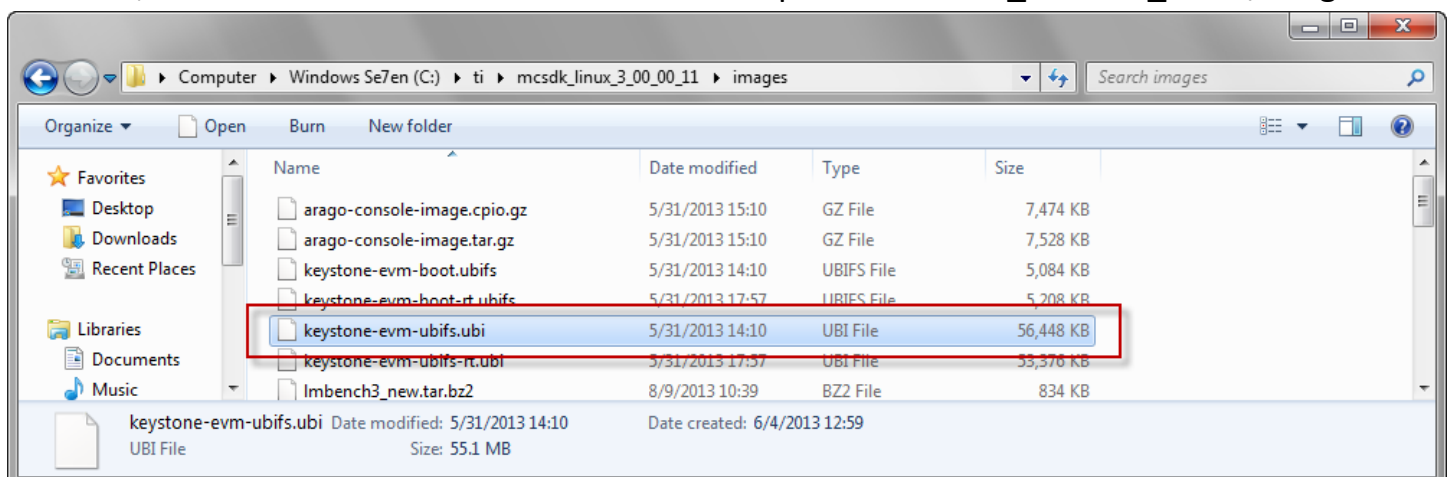
The bootloader partition is blank, u-boot is stored on SPI NOR flash for now. The names in brackets are partitions' name, which will be used in the following NAND partition operations.

6. Setup UBI related operation scripts:

Set the UBI file name:

```
setenv name_ubi '<the actual UBI file name you will use>'
```

This value should be the same with the UBI file which you will use for booting procedures, in this case, this value is the same with the UBI file in path of MCSDK_INSTALL_PATH/images



Which should be:

```
setenv name_ubi 'keystone-evm-ubifs.ubi'
```

Then set the UBI file size in byte:

```
setenv size_ubi '<the actual UBI file size you will use>'
```

In this case, the size value should be:

```
setenv size_ubi '0x3720000'
```

Set NAND Flash UBI partition erases script:

```
setenv erase_nand_ubi 'nand erase.part ubifs'
```

The bold partition name should be the same with your MTD partition configurations which configured earlier of the “mtdparts”

Set NAND Flash UBI partition writes script:

```
setenv write_nand_ubi 'nand write ${addr_fs} ubifs ${size_ubi}'
```

Please also keep the bold part is the valid value based on the “mtdparts” configuration.

Set UBI file fetches script:

```
setenv get_tftp_ubi 'dhcp ${addr_fs} ${tftp_root}/${name_ubi}'
```

After the modification, we can use this command to save the environment variables back to Flash

```
saveenv
```

```
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 # setenv no_post 1
TCI6638 EVM # setenv boot 'ubi'
TCI6638 EVM # setenv serverip '157.87.4.174'
TCI6638 EVM # setenv tftp_root
TCI6638 EVM # setenv mtdparts 'mtdparts=davinci_nand.0:1024k(bootloader),512k(params)ro,129536k(ubifs)'
TCI6638 EVM # setenv name_ubi 'keystone-evm-ubifs.ubi'
TCI6638 EVM # setenv size_ubi '0x3720000'
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 # saveenv
Saving Environment to NAND...
Erasing NAND...
Erasing at 0x120000 -- 100% complete.
Writing to NAND... done
TCI6638 EVM #
```

8.2 Burn the UBI on NAND Flash on EVM

After configuring U-boot environment variables, we can use U-boot to burn the UBI to the NAND Flash on EVM by using:

```
run erase_nand_ubi
```

```
run get_tftp_ubi
```

```
run write_nand_ubi
```

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

```
TCI6638 EVM # run erase_nand_ubi
NAND erase.part: device 0 offset 0x180000, size 0x7e80000
Erasing at 0x7fe0000 -- 100% complete.
OK
TCI6638 EVM # run get_tftp_ubi
BOOTP broadcast 1
BOOTP broadcast 2
*** Unhandled DHCP Option in OFFER/ACK: 44
*** Unhandled DHCP Option in OFFER/ACK: 46
*** Unhandled DHCP Option in OFFER/ACK: 150
*** Unhandled DHCP Option in OFFER/ACK: 44
*** Unhandled DHCP Option in OFFER/ACK: 46
*** Unhandled DHCP Option in OFFER/ACK: 150
DHCP client bound to address 157.87.4.196
Using TCI6638_ETH device
TFTP from server 157.87.4.174; our IP address is 157.87.4.196
Filename '/keystone-evm-ubifs.ubi'.
Load address: 0x82000000
Loading: #####
#####
```

... ..

```
#####
#####
906.3 KiB/s
done
Bytes transferred = 57802752 (3720000 hex)
TCI6638 EVM # run write_nand_ubi
NAND write: device 0 offset 0x180000, size 0x3720000
57802752 bytes written: OK
TCI6638 EVM #
```

8.3 Load and Run Linux Kernel by U-boot with UBI

After configuring U-boot environment variables, we can use U-boot to load and run Linux Kernel with UBI.

You can see the similar print info for the typical SMP Linux Kernel system booting procedures:

```
U-Boot SPL 2013.01 (May 31 2013 - 01:01:04)
SF: Detected N25Q128A with page size 64 KiB, total 16 MiB
```

```
U-Boot 2013.01 (May 31 2013 - 01:01:04)
```

```
I2C:   ready
DRAM:  1 GiB
```

```
NAND: 512 MiB
Net: TCI6638_EMAC
Warning: TCI6638_EMAC using MAC address from net device
, TCI6638_EMAC1
Hit any key to stop autoboot: 0
Creating 1 MTD partitions on "nand0":
0x0000000180000-0x000008000000 : "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: volume 1 ("rootfs") re-sized from 398 to 915 LEBs
UBI: attached mtd1 to ubi0
UBI: MTD device name: "mtd=2"
UBI: MTD device size: 126 MiB
UBI: number of good PEBs: 1012
UBI: number of bad PEBs: 0
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: 1012
UBI: number of PEBs reserved for bad PEB handling: 10
UBI: max/mean erase counter: 1/0
UBIFS: mounted UBI device 0, volume 0, name "boot"
UBIFS: mounted read-only
UBIFS: file system size: 5079040 bytes (4960 KiB, 4 MiB, 40 LEBs)
UBIFS: journal size: 1142785 bytes (1116 KiB, 1 MiB, 9 LEBs)
UBIFS: media format: w4/r0 (latest is w4/r0)
UBIFS: default compressor: LZO
UBIFS: reserved for root: 0 bytes (0 KiB)
Loading file 'uImage-k2hk-evm.dtb' to addr 0x87000000 with size 54537 (0x0000d509)...
Done
Loading file 'skern-keystone-evm.bin' to addr 0x0c5f0000 with size 45056 (0x0000b000)...
Done
Loading file 'uImage-keystone-evm.bin' to addr 0x88000000 with size 3529136
(0x0035d9b0)...
Done
## installed monitor, freq [133120000], status 133120000
## Booting kernel from Legacy Image at 88000000 ...
Image Name: Linux-3.8.4
Created: 2013-05-31 4:01:23 UTC
Image Type: ARM Linux Kernel Image (uncompressed)
Data Size: 3529072 Bytes = 3.4 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 87010508

Starting kernel ...

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

```

[ 0.000000] Linux version 3.8.4 (gtbldadm@ubuntu-12) (gcc version 4.7.3 20130226
(prerelease) (crosstool-NG linaro-1.13.1-4.7-2013.03-20130313 - Linaro GCC 2013.03) ) #1
SMP Fri May 31 00:00:49 EDT 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 @c0af4000 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: 496056k/496056k available, 28232k reserved, 0K highmem
[ 0.000000] Virtual kernel memory layout:
[ 0.000000] vector : 0xffff0000 - 0xffff1000 ( 4 kB)
[ 0.000000] fixmap : 0xffff0000 - 0xffffe000 ( 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 - 0xc06320e8 (6313 kB)
[ 0.000000] .init : 0xc0633000 - 0xc0673d80 ( 260 kB)
[ 0.000000] .data : 0xc0674000 - 0xc06b0d08 ( 244 kB)
[ 0.000000] .bss : 0xc06b0d08 - 0xc06db39c ( 170 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 798720000, postdiv = 2, mult = 12, prediv = 0
[ 0.000000] pll_clk parent_rate(122880000 Hz), rate(327680000 Hz), postdiv = 6, mult =
15, prediv = 0
[ 0.000000] tci6614-timer: no matching node
[ 0.000000] Architected local timer running at 133.12MHz (phys).
[ 0.000000] Switching to timer-based delay loop
[ 0.000000] sched_clock: 32 bits at 133MHz, resolution 7ns, wraps every 32263ms
[ 0.000000] Console: colour dummy device 80x30
[ 0.000110] Calibrating delay loop (skipped), value calculated using timer frequency..
266.24 BogoMIPS (lpj=1331200)
[ 0.000127] pid_max: default: 4096 minimum: 301
[ 0.000360] Mount-cache hash table entries: 512
[ 0.013450] CPU: Testing write buffer coherency: ok
[ 0.013722] CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
[ 0.013759] Setting up static identity map for 0x8047c8a0 - 0x8047c8d4
[ 0.101110] CPU1: Booted secondary processor
[ 0.101132] CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
[ 0.187935] CPU2: Booted secondary processor
[ 0.187957] CPU2: thread -1, cpu 2, socket 0, mpidr 80000002
[ 0.274751] CPU3: Booted secondary processor
[ 0.274774] CPU3: thread -1, cpu 3, socket 0, mpidr 80000003
[ 0.274838] Brought up 4 CPUs
[ 0.274874] SMP: Total of 4 processors activated (1064.96 BogoMIPS).
[ 0.298141] NET: Registered protocol family 16
[ 0.299913] DMA: preallocated 256 KiB pool for atomic coherent allocations

```

```

[ 0.315437] keystone2_pcie_serdes_setup
[ 0.317541] keystone2_pcie_serdes_setup done
[ 0.317571] hw-breakpoint: found 5 (+1 reserved) breakpoint and 4 watchpoint registers.
[ 0.317581] hw-breakpoint: maximum watchpoint size is 8 bytes.
[ 0.336363] bio: create slab <bio-0> at 0
[ 0.336673] keystone-pcie: keystone_pcie_rc_init - start
[ 0.336981] MEM 0x0000000050000000..0x000000005fffffff -> 0x0000000050000000
[ 0.336999] IO 0x0000000024000000..0x0000000024003fff -> 0x0000000000000000
[ 0.337058] pcie - number of legacy irqs = 4
[ 0.337141] pcie - number of MSI host irqs = 8, msi_irqs = 32
[ 0.451457] keystone-pcie: Doing PCI Setup...Done
[ 0.451468] keystone-pcie: Starting PCI scan...
[ 0.451689] PCI host bridge to bus 0000:00
[ 0.451708] pci_bus 0000:00: root bus resource [mem 0x50000000-0x5fffffff]
[ 0.451723] pci_bus 0000:00: root bus resource [io 0x0000-0x3fff]
[ 0.451737] pci_bus 0000:00: No busn resource found for root bus, will use [bus 00-ff]
[ 0.451815] PCI: bus0: Fast back to back transfers enabled
[ 0.451844] keystone-pcie: Ending PCI scan...
[ 0.451857] keystone-pcie: keystone_pcie_rc_init - end
[ 0.452178] vgaarb: loaded
[ 0.452755] SCSI subsystem initialized
[ 0.453465] usbcore: registered new interface driver usbfs
[ 0.453640] usbcore: registered new interface driver hub
[ 0.453828] usbcore: registered new device driver usb
[ 0.455811] keystone-hwqueue hwqueue.2: qmgr start queue 0, number of queues 8192
[ 0.455968] keystone-hwqueue hwqueue.2: 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
[ 0.455986] keystone-hwqueue hwqueue.2: qmgr start queue 8192, number of queues 8192
[ 0.456138] keystone-hwqueue hwqueue.2: 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
[ 0.457293] keystone-hwqueue hwqueue.2: qos: sched port @8096, drop sched @8000
[ 0.458888] keystone-hwqueue hwqueue.2: qos: sched port @6496, drop sched @6400
[ 0.460473] keystone-hwqueue hwqueue.2: added pool pool-net: 2048 descriptors of size
128
[ 0.460491] keystone-hwqueue hwqueue.2: added pool pool-rio: 128 descriptors of size
256
[ 0.460507] keystone-hwqueue hwqueue.2: added pool pool-udma: 1636 descriptors of size
256
[ 0.460523] keystone-hwqueue hwqueue.2: added pool pool-xge: 2048 descriptors of size
128
[ 0.463992] keystone-hwqueue hwqueue.2: registered queues 0-16383
[ 0.464110] keystone-hwqueue hwqueue.2: qos version 0x2000105, magic valid
[ 0.464741] keystone-hwqueue hwqueue.2: qos version 0x2000105, magic valid
[ 0.475997] keystone-pktdma 2004000.pktdma: registered 20 logical channels, flows 32,
tx chans: 9, rx chans: 24
[ 0.481173] keystone-pktdma 2a08000.pktdma: registered 24 logical channels, flows 32,
tx chans: 32, rx chans: 32, loopback
[ 0.482181] keystone-pktdma 2fa1000.pktdma: registered 4 logical channels, flows 32,
tx chans: 16, rx chans: 16
[ 0.482426] Switching to clocksource arch_sys_counter
[ 0.505453] NET: Registered protocol family 2
[ 0.506072] TCP established hash table entries: 4096 (order: 3, 32768 bytes)
[ 0.506192] TCP bind hash table entries: 4096 (order: 3, 32768 bytes)
[ 0.506302] TCP: Hash tables configured (established 4096 bind 4096)
[ 0.506345] TCP: reno registered
[ 0.506361] UDP hash table entries: 256 (order: 1, 8192 bytes)
[ 0.506391] UDP-Lite hash table entries: 256 (order: 1, 8192 bytes)
[ 0.506677] NET: Registered protocol family 1

```

```

[ 0.506921] RPC: Registered named UNIX socket transport module.
[ 0.506932] RPC: Registered udp transport module.
[ 0.506940] RPC: Registered tcp transport module.
[ 0.506949] RPC: Registered tcp NFSv4.1 backchannel transport module.
[ 0.654178] Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
[ 0.654637] NTFS driver 2.1.30 [Flags: R/O].
[ 0.655159] jffs2: version 2.2. (NAND) © 2001-2006 Red Hat, Inc.
[ 0.657310] NET: Registered protocol family 38
[ 0.657690] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 254)
[ 0.657704] io scheduler noop registered
[ 0.657714] io scheduler deadline registered
[ 0.657973] io scheduler cfq registered (default)
[ 0.659979] keystone-udma udma0.3: registered udma device udma0
[ 0.737626] Serial: 8250/16550 driver, 4 ports, IRQ sharing disabled
[ 0.739761] 2530c00.serial: ttyS0 at MMIO 0x2530c00 (irq = 309) is a 16550A
[ 1.478458] console [ttyS0] enabled
[ 1.482838] 2531000.serial: ttyS1 at MMIO 0x2531000 (irq = 312) is a 16550A
[ 1.499023] loop: module loaded
[ 1.502377] at24 0-0050: 131072 byte 24c1024 EEPROM, writable, 1 bytes/write
[ 1.510883] Generic platform RAM MTD, (c) 2004 Simtec Electronics
[ 1.528295] ONFI param page 0 valid
[ 1.531771] ONFI flash detected
[ 1.534918] NAND device: Manufacturer ID: 0x2c, Chip ID: 0xac (Micron
MT29F4G08ABDAHC), 512MiB, page size: 2048, OOB size: 64
[ 1.547984] Bad block table found at page 262080, version 0x01
[ 1.557079] Bad block table found at page 262016, version 0x01
[ 1.564647] 3 ofpart partitions found on MTD device 30000000.nand
[ 1.570718] Creating 3 MTD partitions on "30000000.nand":
[ 1.576108] 0x0000000000000-0x0000000100000 : "u-boot"
[ 1.582104] 0x0000000100000-0x0000000180000 : "params"
[ 1.588053] 0x0000000180000-0x0000000800000 : "ubifs"
[ 1.594011] davinci_nand 30000000.nand: controller rev. 2.5
[ 1.600394] spi_davinci 21000400.spi: master is unqueued, this is deprecated
[ 1.607872] m25p80 spi32766.0: found n25q128a11, expected n25q128
[ 1.613964] m25p80 spi32766.0: n25q128a11 (16384 Kbytes)
[ 1.619268] 2 ofpart partitions found on MTD device spi32766.0
[ 1.625087] Creating 2 MTD partitions on "spi32766.0":
[ 1.630209] 0x0000000000000-0x0000000080000 : "u-boot-spl"
[ 1.636529] 0x0000000080000-0x0000000100000 : "test"
[ 1.642854] spi_davinci 21000400.spi: Controller at 0xe0878400
[ 1.650583] keystone-netcp 2f00000.netcp: No streaming regs defined
[ 1.658382] keystone-netcp 2090000.netcp: Created interface "eth0"
[ 1.664571] keystone-netcp 2090000.netcp: dma_chan_name nettx0
[ 1.671705] keystone-netcp 2090000.netcp: Created interface "eth1"
[ 1.677888] keystone-netcp 2090000.netcp: dma_chan_name nettx1
[ 1.685521] keystone-dwc3 2690000.dwc: usbss revision 47914300
[ 1.691385] keystone-dwc3 2690000.dwc: mapped irq 425 to virq 608
[ 1.901098] xhci-hcd xhci-hcd: xHCI Host Controller
[ 1.905997] xhci-hcd xhci-hcd: new USB bus registered, assigned bus number 1
[ 1.914357] xhci-hcd xhci-hcd: irq 608, io mem 0x02690000
[ 1.919848] usb usb1: New USB device found, idVendor=1d6b, idProduct=0002
[ 1.926634] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 1.933844] usb usb1: Product: xHCI Host Controller
[ 1.938705] usb usb1: Manufacturer: Linux 3.8.4 xhci-hcd
[ 1.944005] usb usb1: SerialNumber: xhci-hcd
[ 1.948858] hub 1-0:1.0: USB hub found
[ 1.952624] hub 1-0:1.0: 1 port detected
[ 1.956836] xhci-hcd xhci-hcd: xHCI Host Controller
[ 1.961708] xhci-hcd xhci-hcd: new USB bus registered, assigned bus number 2
[ 1.968870] usb usb2: New USB device found, idVendor=1d6b, idProduct=0003

```

```
[ 1.975652] usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[ 1.982860] usb usb2: Product: xHCI Host Controller
[ 1.987721] usb usb2: Manufacturer: Linux 3.8.4 xhci-hcd
[ 1.993021] usb usb2: SerialNumber: xhci-hcd
[ 1.997853] hub 2-0:1.0: USB hub found
[ 2.001600] hub 2-0:1.0: 1 port detected
[ 2.005923] Initializing USB Mass Storage driver...
[ 2.010953] usbcore: registered new interface driver usb-storage
[ 2.016949] USB Mass Storage support registered.
[ 2.021894] mousedev: PS/2 mouse device common for all mice
[ 2.027795] i2c /dev entries driver
[ 2.032020] watchdog 22f0080.wdt: heartbeat 60 sec
[ 2.037207] keystone-crypto 20c0000.crypto: crypto accelerator enabled
[ 2.044271] usbcore: registered new interface driver usbhid
[ 2.049822] usbhid: USB HID core driver
[ 2.054169] remoteproc0: 2620040.dsp0 is available
[ 2.059031] remoteproc0: Note: remoteproc is still under development and considered
experimental.
[ 2.067983] remoteproc0: THE BINARY FORMAT IS NOT YET FINALIZED, and backward
compatibility isn't yet guaranteed.
[ 2.078343] remoteproc0: no firmware found
[ 2.082935] remoteproc1: 2620044.dsp1 is available
[ 2.087795] remoteproc1: Note: remoteproc is still under development and considered
experimental.
[ 2.096739] remoteproc1: THE BINARY FORMAT IS NOT YET FINALIZED, and backward
compatibility isn't yet guaranteed.
[ 2.107101] remoteproc1: no firmware found
[ 2.111684] remoteproc2: 2620048.dsp2 is available
[ 2.116563] remoteproc2: Note: remoteproc is still under development and considered
experimental.
[ 2.125501] remoteproc2: THE BINARY FORMAT IS NOT YET FINALIZED, and backward
compatibility isn't yet guaranteed.
[ 2.135867] remoteproc2: no firmware found
[ 2.140442] remoteproc3: 262004c.dsp3 is available
[ 2.145321] remoteproc3: Note: remoteproc is still under development and considered
experimental.
[ 2.154257] remoteproc3: THE BINARY FORMAT IS NOT YET FINALIZED, and backward
compatibility isn't yet guaranteed.
[ 2.164609] remoteproc3: no firmware found
[ 2.169194] remoteproc4: 2620050.dsp4 is available
[ 2.174072] remoteproc4: Note: remoteproc is still under development and considered
experimental.
[ 2.183009] remoteproc4: THE BINARY FORMAT IS NOT YET FINALIZED, and backward
compatibility isn't yet guaranteed.
[ 2.193361] remoteproc4: no firmware found
[ 2.197939] remoteproc5: 2620054.dsp5 is available
[ 2.202817] remoteproc5: Note: remoteproc is still under development and considered
experimental.
[ 2.211743] remoteproc5: THE BINARY FORMAT IS NOT YET FINALIZED, and backward
compatibility isn't yet guaranteed.
[ 2.222111] remoteproc5: no firmware found
[ 2.226696] remoteproc6: 2620058.dsp6 is available
[ 2.231556] remoteproc6: Note: remoteproc is still under development and considered
experimental.
[ 2.240499] remoteproc6: THE BINARY FORMAT IS NOT YET FINALIZED, and backward
compatibility isn't yet guaranteed.
[ 2.250855] remoteproc6: no firmware found
[ 2.255448] remoteproc7: 262005c.dsp7 is available
[ 2.260308] remoteproc7: Note: remoteproc is still under development and considered
experimental.
```

```

[ 2.269251] remoteproc7: THE BINARY FORMAT IS NOT YET FINALIZED, and backward
compatibility isn't yet guaranteed.
[ 2.279606] remoteproc7: no firmware found
[ 2.284283] oprofile: no performance counters
[ 2.288904] oprofile: using timer interrupt.
[ 2.293486] GACT probability on
[ 2.296618] Mirror/redirect action on
[ 2.300268] Simple TC action Loaded
[ 2.304461] netem: version 1.3
[ 2.307509] u32 classifier
[ 2.310202] Performance counters on
[ 2.314038] input device check on
[ 2.317685] Actions configured
[ 2.321081] Netfilter messages via NETLINK v0.30.
[ 2.325797] nf_conntrack version 0.5.0 (8006 buckets, 32024 max)
[ 2.332449] ctnetlink v0.93: registering with nfnetlink.
[ 2.338183] IPv4 over IPv4 tunneling driver
[ 2.343038] gre: GRE over IPv4 demultiplexor driver
[ 2.347899] ip_gre: GRE over IPv4 tunneling driver
[ 2.353553] ip_tables: (C) 2000-2006 Netfilter Core Team
[ 2.358990] ipt_CLUSTERIP: ClusterIP Version 0.8 loaded successfully
[ 2.365371] arp_tables: (C) 2002 David S. Miller
[ 2.370026] TCP: cubic registered
[ 2.373337] Initializing XFRM netlink socket
[ 2.378510] NET: Registered protocol family 10
[ 2.384066] NET: Registered protocol family 17
[ 2.388517] NET: Registered protocol family 15
[ 2.393039] Bridge firewalling registered
[ 2.397041] Ebtables v2.0 registered
[ 2.400693] 8021q: 802.1Q VLAN Support v1.8
[ 2.405740] sctp: Hash tables configured (established 16384 bind 16384)
[ 2.412518] NET: Registered protocol family 40
[ 2.417158] VFP support v0.3: implementor 41 architecture 4 part 30 variant f rev 0
[ 2.424828] Registering SWP/SWPB emulation handler
[ 2.431144] UBI: attaching mtd2 to ubi0
[ 5.747773] UBI: scanning is finished
[ 5.788875] UBI warning: print_rsvd_warning: cannot reserve enough PEBs for bad PEB
handling, reserved 10, need 80
[ 5.800773] UBI: attached mtd2 (name "ubifs", size 126 MiB) to ubi0
[ 5.807033] UBI: PEB size: 131072 bytes (128 KiB), LEB size: 126976 bytes
[ 5.813805] UBI: min./max. I/O unit sizes: 2048/2048, sub-page size 2048
[ 5.820482] UBI: VID header offset: 2048 (aligned 2048), data offset: 4096
[ 5.827338] UBI: good PEBs: 1012, bad PEBs: 0, corrupted PEBs: 0
[ 5.833328] UBI: user volume: 2, internal volumes: 1, max. volumes count: 128
[ 5.840438] UBI: max/mean erase counter: 1/0, WL threshold: 4096, image sequence
number: 1653276260
[ 5.849458] UBI: available PEBs: 0, total reserved PEBs: 1012, PEBs reserved for bad
PEB handling: 10
[ 5.858664] UBI: background thread "ubi_bgt0d" started, PID 1187
[ 5.865679] UBIFS: parse sync
[ 5.877439] UBIFS: background thread "ubifs_bgt0_1" started, PID 1189
[ 6.086463] UBIFS: start fixing up free space
[ 19.147261] UBIFS: free space fixup complete
[ 19.257460] UBIFS: mounted UBI device 0, volume 1, name "rootfs"(null)
[ 19.263982] UBIFS: LEB size: 126976 bytes (124 KiB), min./max. I/O unit sizes: 2048
bytes/2048 bytes
[ 19.273097] UBIFS: FS size: 107421696 bytes (102 MiB, 846 LEBs), journal size 9023488
bytes (8 MiB, 72 LEBs)
[ 19.282899] UBIFS: reserved for root: 0 bytes (0 KiB)

```

```
[ 19.287936] UBIFS: media format: w4/r0 (latest is w4/r0), UUID 8856CB9A-E7A9-47BF-B2FE-D05A155F5970, small LPT model
[ 19.308375] VFS: Mounted root (ubifs filesystem) on device 0:11.
[ 19.314603] Freeing init memory: 256K
INIT: version 2.88 booting
Starting udev
Starting Bootlog daemon: bootlogd.
Configuring network interfaces... [ 24.435878] keystone-netcp 2090000.netcp:
initializing cpsw version 1.3 (1) SGMIID identification value 0x4ed1
[ 24.447008] keystone-netcp 2090000.netcp: Created a cpsw ale engine
[ 24.453384] keystone-netcp 2090000.netcp: initialized cpsw ale revision 1.3
[ 24.512640] keystone-netcp 2090000.netcp: Using Packet Accelerator Firmware version
0x01030008
[ 24.521237] keystone-netcp 2090000.netcp: pa_clk_rate(163840000
HZ),mult(25000),shift(12)
[ 24.540740] net eth0: netcp device eth0 opened
[ 24.547712] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 24.553537] 8021q: adding VLAN 0 to HW filter on device eth0
[ 24.559182] net eth0: adding rx vlan id: 0
[ 24.563370] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
udhcpc (v1.20.2) started
Sending discover...
Sending select for 157.87.4.196...
Lease of 157.87.4.196 obtained, lease time 28800
/etc/udhcpc.d/50default: Adding DNS 157.87.32.12
/etc/udhcpc.d/50default: Adding DNS 157.87.119.10
/etc/udhcpc.d/50default: Adding DNS 157.170.147.7
[ 25.740368] keystone-netcp 2090000.netcp: initializing cpsw version 1.3 (1) SGMIID
identification value 0x4ed1
[ 25.763758] net eth1: netcp device eth1 opened
[ 25.769133] IPv6: ADDRCONF(NETDEV_UP): eth1: link is not ready
[ 25.774963] 8021q: adding VLAN 0 to HW filter on device eth1
[ 25.780607] net eth1: adding rx vlan id: 0
udhcpc (v1.20.2) started
[ 25.852538] 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
Fri May 31 05:08:00 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-rltracert6.
Configuring ndisc6-rdisc6.
```

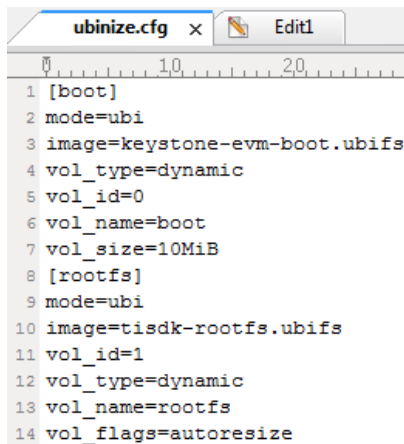

9. Rebuild and Update UBI on TCI6638 EVM

9.1 Modify and Rebuild UBI Image

Before we modify the UBI image, we need to know the partitions in UBI image.

When creating the UBI image, there is a configuration file needed during the image creation procedures to partition the UBI image into many parts.

Let's just use the prebuilt UBI image "keystone-evm-ubifs.ubi" in MCSDK as the example, the configuration file for this image is "ubinize.cfg" which is in the same path:



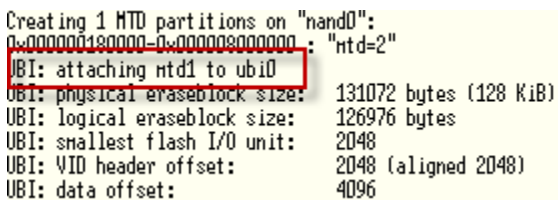
```
ubinize.cfg
1 [boot]
2 mode=ubi
3 image=keystone-evm-boot.ubifs
4 vol_type=dynamic
5 vol_id=0
6 vol_name=boot
7 vol_size=10MiB
8 [rootfs]
9 mode=ubi
10 image=tisdk-rootfs.ubifs
11 vol_id=1
12 vol_type=dynamic
13 vol_name=rootfs
14 vol_flags=autoresize
```

We can see this image is combined by 2 ubifs images:

Partition 0, name is "boot", image is "keystone-evm-boot.ubifs"

Partition 1, name is "rootfs", image is "tisdk-rootfs.ubifs"

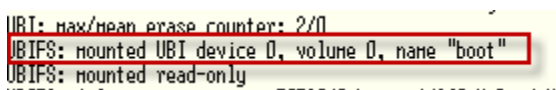
As this image is the first UBI image in system, the name of this image will be "ubi0" which we can tell from the Kernel print info:



```
Creating 1 MTD partitions on "nand0":
0x000000000000-0x000000000000 : "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
```

So the 2 partitions in this image can have alias as "ubi0_0" and "ubi0_1".

The partition "boot" will be mounted by "ubifsmount boot" for getting the DTB, boot_monitor, and ulmage in U-boot then will not use anymore in Linux Kernel. We can get the mount info from printed log:



```
UBI: max/mean erase counter: 2/0
UBIFS: mounted UBI device 0, volume 0, name "boot"
UBIFS: mounted read-only
```

The partition “rootfs” will be mounted during Linux Kernel starting procedures as the root file system, we can also get the mount info from printed log:

```
[ 5.869483] UBIFS: background thread "ubifs_bgt0_1" started, PID 1189
[ 6.349153] UBIFS: mounted UBI device 0, volume 1, name "rootfs"(null)
[ 6.355671] UBIFS: LEB size: 126976 bytes (124 KiB), min./max. I/O unit sizes: 2048 bytes/2048 bytes
```

For the partition “rootfs”, any modification will be saved in the UBIFS file on Flash because this partition already mounted as the root file system.

We can simply simulate the modification such as:

```
mkdir 12345678/
```

For the partition “boot”, due to not used in Kernel, you should mount it first before any modification by using:

```
mount -t ubifs ubi0_0 <the folder you want to mount>
```

E.g.: After the Linux Kernel was up, you can use the following commands to mount the “boot” partition to a folder:

```
root
```

```
cd /mnt
```

```
mkdir boot/
```

```
mount -t ubifs ubi0_0 boot/
```

```
cd boot/
```

```
ls
```

We can see the following output:

```
Arago Project http://arago-project.org keystone-evm ttyS0
Arago 2013.04 keystone-evm ttyS0

keystone-evm login: root
root@keystone-evm:~# cd /mnt/
root@keystone-evm:/mnt# mkdir boot/
root@keystone-evm:/mnt# mount -t ubifs ubi0_0 boot/
[ 3568.119680] UBIFS: background thread "ubifs_bgt0_0" started, PID 1652
[ 3568.455643] UBIFS: mounted UBI device 0, volume 0, name "boot"(null)
[ 3568.461980] UBIFS: LEB size: 126976 bytes (124 KiB), min./max. I/O unit sizes: 2048 bytes/2048 bytes
[ 3568.471152] UBIFS: FS size: 5079040 bytes (4 MiB, 40 LEBs), journal size 1142785 bytes (1 MiB, 9 LEBs)
[ 3568.480440] UBIFS: reserved for root: 0 bytes (0 KiB)
[ 3568.485491] UBIFS: media format: u4/r0 (latest is u4/r0), UUID CEFF56A4-5900-4FC4-98FD-021F9011CD54, small LPT model
root@keystone-evm:/mnt# cd boot/
root@keystone-evm:/mnt/boot# ls
skern-keystone-evm.bin  uImage-k2hk-evm.dtb      uImage-keystone-evm.bin
root@keystone-evm:/mnt/boot#
```

Files containing in this partition will in this folder, we can simply simulate the modification such as:

```
mkdir abcd/
```

Then you should un-mount this partition to make the modifications take effect by using:

```
cd ..
```

```
umount boot/
```

```
root@keystone-evm:/mnt/boot#  
root@keystone-evm:/mnt/boot# mkdir abcd  
root@keystone-evm:/mnt/boot# ls  
abcd                               uImage-k2hk-evm.dtb  
skern-keystone-evm.bin             uImage-keystone-evm.bin  
root@keystone-evm:/mnt/boot# cd ..  
root@keystone-evm:/mnt# umount boot/  
[ 3907.870683] UBIFS: un-mount UBI device 0, volume 0  
[ 3907.876319] UBIFS: background thread "ubifs_bgt0_0" stops  
root@keystone-evm:/mnt# █
```

Please note that you must un-mount the partition before any reboot or reset, or the file system will have unexpected error.

9.2 Boot and Test the Updated UBI on TCI6638EVM

After the UBI content modification, let's check whether these modifications take effect or not.

For the "rootfs" partition, after reboot, we can see there is extra folder named "12345678" in the root path:

```
Arago Project  
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  
12345678 dev ipaddr mnt srv usr  
bin etc lib proc sys var  
debug hone media sbin tmp www  
root@keystone-evm:/#
```

For the "boot" partition, we can see the updates after mount it to some folder:

```
root@keystone-evm:/#  
root@keystone-evm:/# cd /mnt/  
root@keystone-evm:/mnt# ls  
boot card cf net ram  
root@keystone-evm:/mnt# cd boot/  
root@keystone-evm:/mnt/boot# ls  
root@keystone-evm:/mnt/boot# cd ..  
root@keystone-evm:/mnt# mount -t ubifs ubi0_0 boot/  
[ 174.732782] UBIFS: background thread "ubifs_bgt0_0" started, PID 1641  
[ 175.065940] UBIFS: mounted UBI device 0, volume 0, name "boot"(null)  
[ 175.072277] UBIFS: LEB size: 126976 bytes (124 KiB), min./max. I/O unit sizes: 2048 bytes/2048 bytes  
[ 175.081401] UBIFS: FS size: 5079040 bytes (4 MiB, 40 LEBs), journal size 1142785 bytes (1 MiB, 9 LEBs)  
[ 175.090695] UBIFS: reserved for root: 0 bytes (0 KiB)  
[ 175.095745] UBIFS: media format: u4/r0 (latest is u4/r0), UUID CFFF56A4-59D0-4FC4-98F0-D21F9D11C054, small LPT model  
root@keystone-evm:/mnt# cd boot/  
root@keystone-evm:/mnt/boot# ls  
abcd                               uImage-k2hk-evm.dtb  
skern-keystone-evm.bin             uImage-keystone-evm.bin  
root@keystone-evm:/mnt/boot#  
root@keystone-evm:/mnt/boot#
```

Appendix. TCI6638 EVM Boot Mode DIP and BMC Encoding

A.1 TCI6638 EVM DIP Switch Reference

These are the bootmodes for the Phase 2 BMC (1.0.1.3[a])

<u>DIPSW</u>	<u>Title</u>	<u>Boot Value(hex)</u>	<u>Comments</u>
0000	Default	0x00110CE7	This setting should be the default bootmode for each board. For K2X, this is expected to be (and is currently) ARM NAND.
0001	JTAG	0x00100001	This setting should be the default JTAG mode for each mode. This mode is sleep w/o no PLLs.
0010	ARM SPI	0x00112005	(These will be DSP on a no-ARM device)
0011	ARM I2C	0x00100003	""
0100	ARM UART	0x00100DEF	"" NOTE: this value is currently incorrect in BMC v1.0.1.2 and will be fixed in v1.0.1.3 and up. (Correct value: 0x00000CEF)
0101	ARM RBL ENET	0x00111CEB	"" , SPI or NAND bootloader will also have ENET boot modes
0110	Reserved	0x001010E1	Right now this is Sleep with Max speed SYS PLL and ARM Bypass. We could use it for 2nd I2C or 2nd SPI config. We could also use it for ARM NAND even for boards where ARM NAND is not the default.
0111	PWR OFF	0x00103EE1	Wait with power off until told what to do by BMC console or AMC MMC command (Right now this is Sleep with Max speed PLLs)
1000	Reserved	0x001101E7	These should be used for alternate configurations of the default bootloader (in NAND or SPI) Right now this is DSP NAND
1001	Reserved	0x001010C1	"" Right now this is Sleep with Slow Sys PLL and Bypass ARM PLL
1010	Reserved	0x00112105	"" Right now this is DSP SPI

DIPSW	Title	Boot Value(hex)	Comments
1011	Reserved	0x00100103	"" Right now this is DSP I2C
1100	User Programmable	0x00100DEF	Right now this is DSP UART
1101	User Programmable	0x001111EB	Right now this is DSP RBL ENET
1110	User Programmable	0x00103CC1	Right now this is Sleep w/ slow SYS PLL and slow ARM PLL
1111	User Programmable	N/A	Right now this is HW DBG / BMC Phase 1 mode, this mode will go away to be replaced by commands to enter hardware debug mode.

A.2 TCI6638 EVM BMC Encoding Reference

The BMC uses an extended boot mode value to encode all the boot configuration values into one 32 bit value. The lower 17 bits of this value matches the 17bit values used in the boot mode description tables.

(Note: Bit 0 value of 0 gives big endian for DSP, and 1 gives little endian for DSP. Also note that the ARMENDIAN bit is opposite of this. ARMENDIAN should always be 0; if ARM big endian operation is desired, the processors endian bit should be changed in SW.)

Bit	Devstat Bit	Config Pin Function	Normal Pin Function	Comments
31	na	na	na	reserved for wait in power off
30	na	na	na	reserved for wait in reset
29	na	na	na	reserved for other BMC SW config
28	na	na	na	reserved for other BMC SW config
27				Reserved for future constant drive config bits
26				Reserved for future constant drive config bits
25		PACLKSEL	PACLKSEL	
24		CORECLKSEL	CORECLKSEL	

<u>Bit</u>	<u>Devstat Bit</u>	<u>Config Pin Function</u>	<u>Normal Pin Function</u>	<u>Comments</u>
23				Reserved for future boot config latched values
22		AVSIFSEL1	TIMI1	Reserved: EVM forces these bits to strap values during reset
21		AVSIFSEL0	TIMIO	""
20		DDR3_REMAP_EN	GPIO16	
19		ARM_LENDIAN	GPIO15	0 = little, 1 = is not supported; do in SW
18		MAINPLLODSEL	GPIO14	
17		ARMAVSSHARED	CORESEL3	
16	16	BOOTMODE15	CORESEL2	
15	15	BOOTMODE14	CORESEL1	
14	14	BOOTMODE13	CORESEL0	
13	13	BOOTMODE12	GPIO13	
12	12	BOOTMODE11	GPIO12	
11	11	BOOTMODE10	GPIO11	
10	10	BOOTMODE9	GPIO10	
9	9	BOOTMODE8	GPIO9	
8	8	BOOTMODE7	GPIO8	
7	7	BOOTMODE6	GPIO7	
6	6	BOOTMODE5	GPIO6	
5	5	BOOTMODE4	GPIO5	
4	4	BOOTMODE3	GPIO4	
3	3	BOOTMODE2	GPIO3	
2	2	BOOTMODE1	GPIO2	
1	1	BOOTMODE0	GPIO1	
0	0	LENDIAN	GPIO0	

To run a custom boot mode value not available on the dip switch, do the following:

```
BMC> setboot <relative boot mode encodings>
```

```
BMC> fullrst
```