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-gnueabihf-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-gnueabihf-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-gnueabi-
```

```
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
```

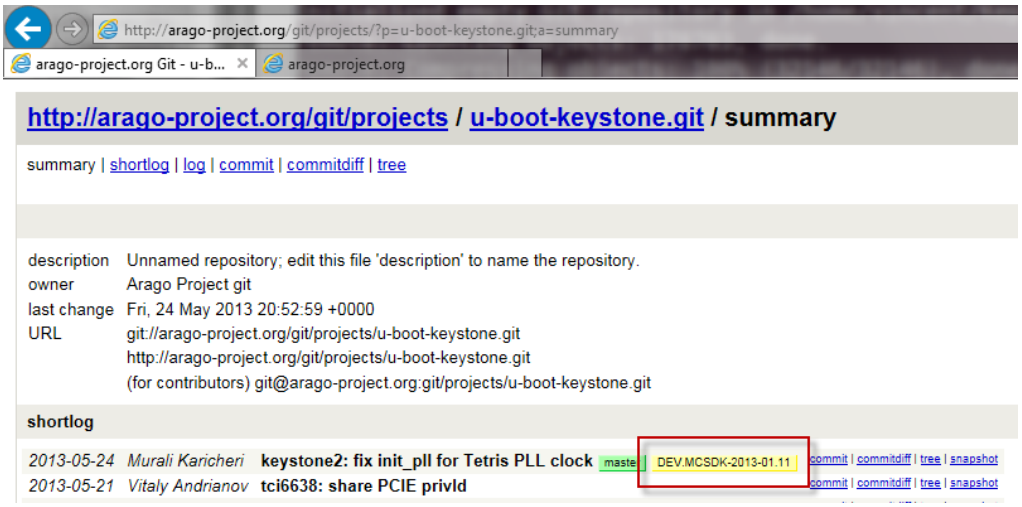
```
git reset --hard <Release tag>
```

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

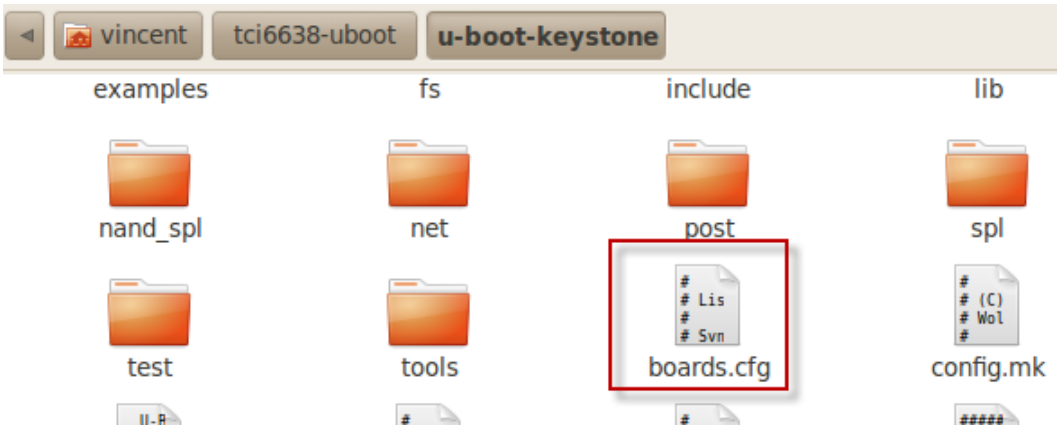


You will see the following info:

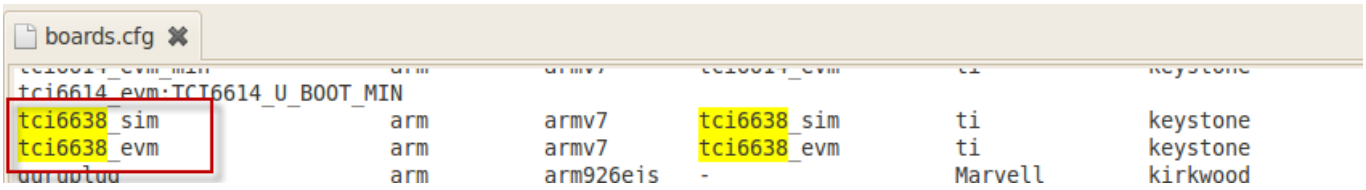
```
vincent@vincent-desktop:~/tci6638-uboot$ git reset --hard DEV.MCSDK-2013-01.11
fatal: Not a git repository (or any of the parent directories): .git
vincent@vincent-desktop:~/tci6638-uboot$ cd u-boot-keystone/
```

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$
```

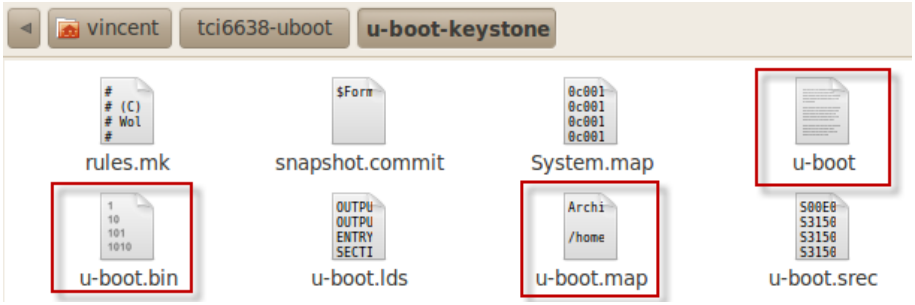
Then we will make the U-boot image by using:

```
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
```

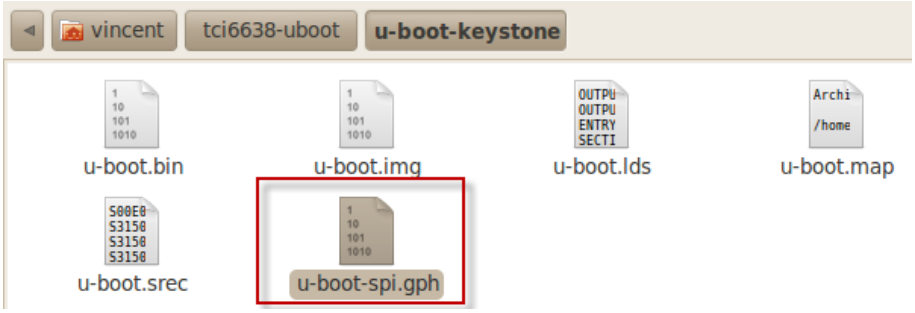
```
make tci6638_evm_config
```

```
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>
```

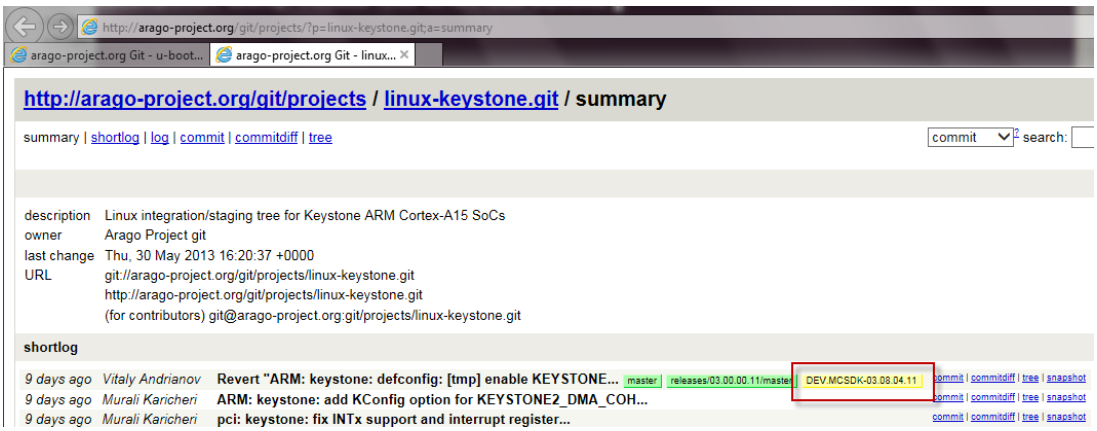
For example, we can use the MCSDK alpha 11 tags:

```
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:



You'll see the following info:

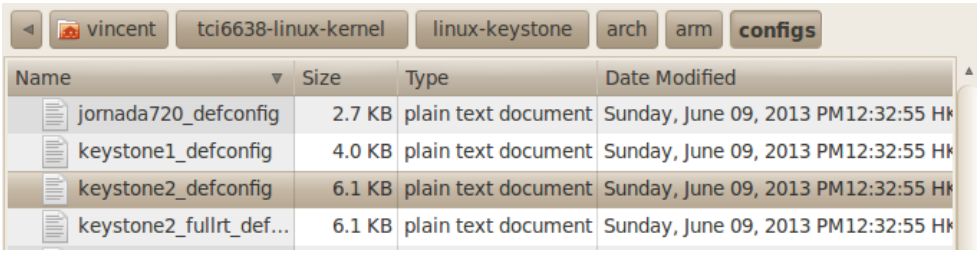
```
vincent@vincent-desktop:~/tci6638-linux-kernel/linux-keystone$ git reset --hard DEV.MCSDK-03.08.04.10
HEAD is now at a5fb562 uio: increasing MAX_UIO_MAPS to 7
```

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/



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/bitsperlong.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”, “uImage” and “zImage” for different usage scenarios:

Image: the built original Linux kernel image

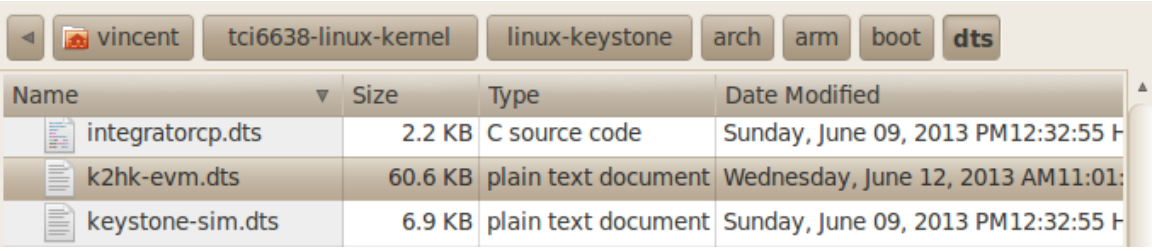
uImage: the built image which uses co-works with u-boot

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/:



Then you’ll see the following info:

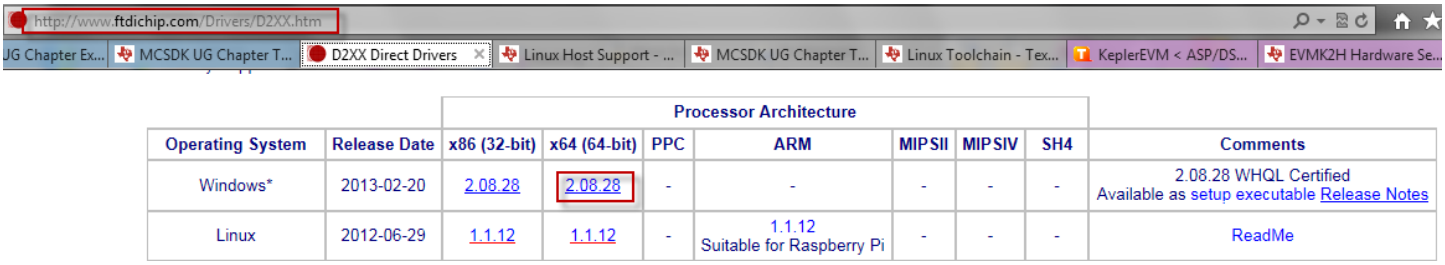
```
vincent@vincent-desktop:~/tci6638-linux-kernel/linux-keystone$ make k2hk-evm.dtb
  DTC     arch/arm/boot/dts/k2hk-evm.dtb
vincent@vincent-desktop:~/tci6638-linux-kernel/linux-keystone$
```

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:

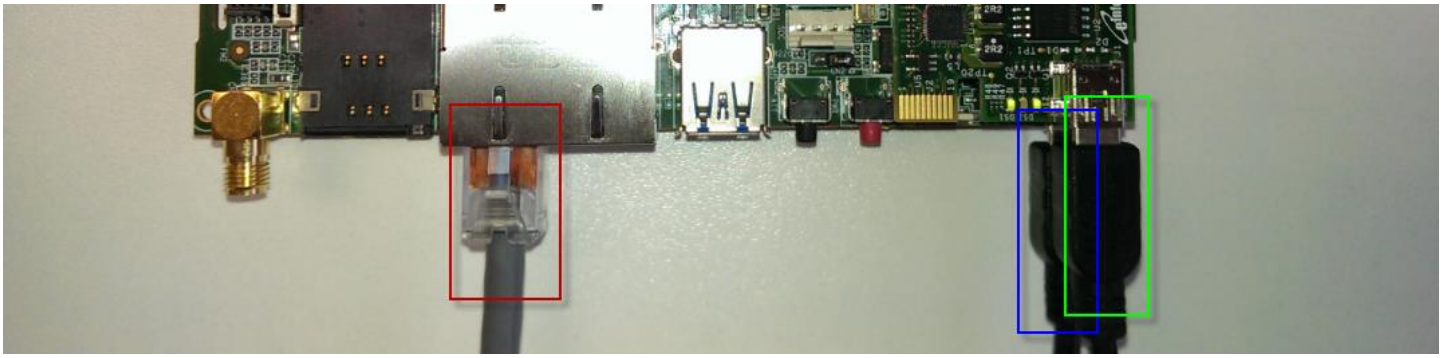
http://www.ftdichip.com/Drivers/D2XX.htm



		Processor Architecture							
Operating System	Release Date	x86 (32-bit)	x64 (64-bit)	PPC	ARM	MIPSII	MIPSIV	SH4	Comments
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

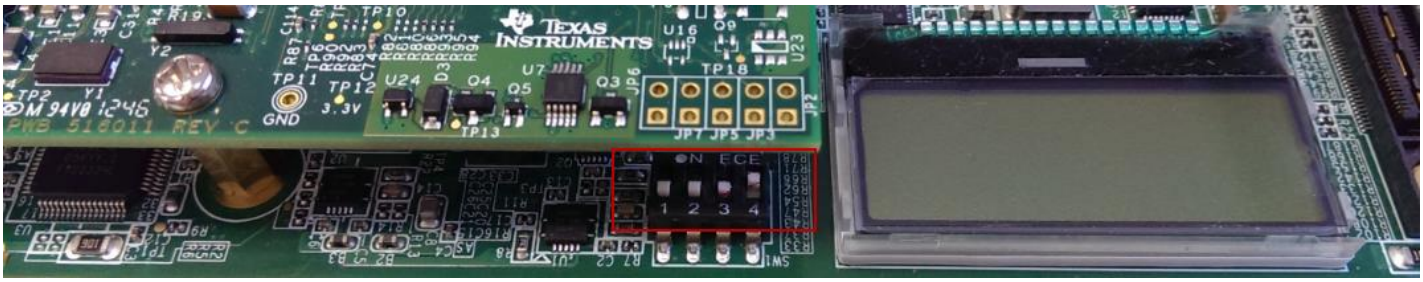
Choose the latest Windows version to download and install, e.g: “CDM 2.08.28 WHQL Certified.7z”

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.

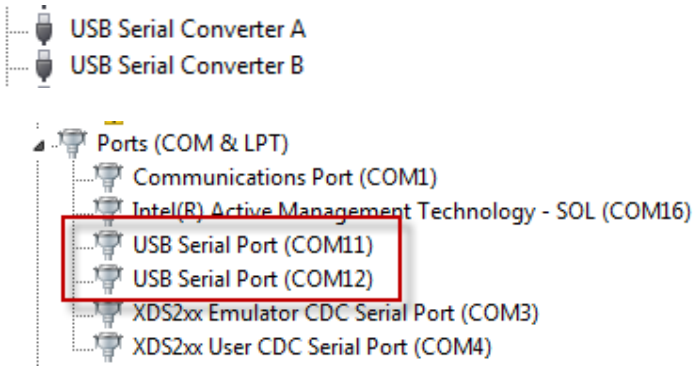
Before we power on the EVM, please make sure the boot DIP pins on EVM in the position like:



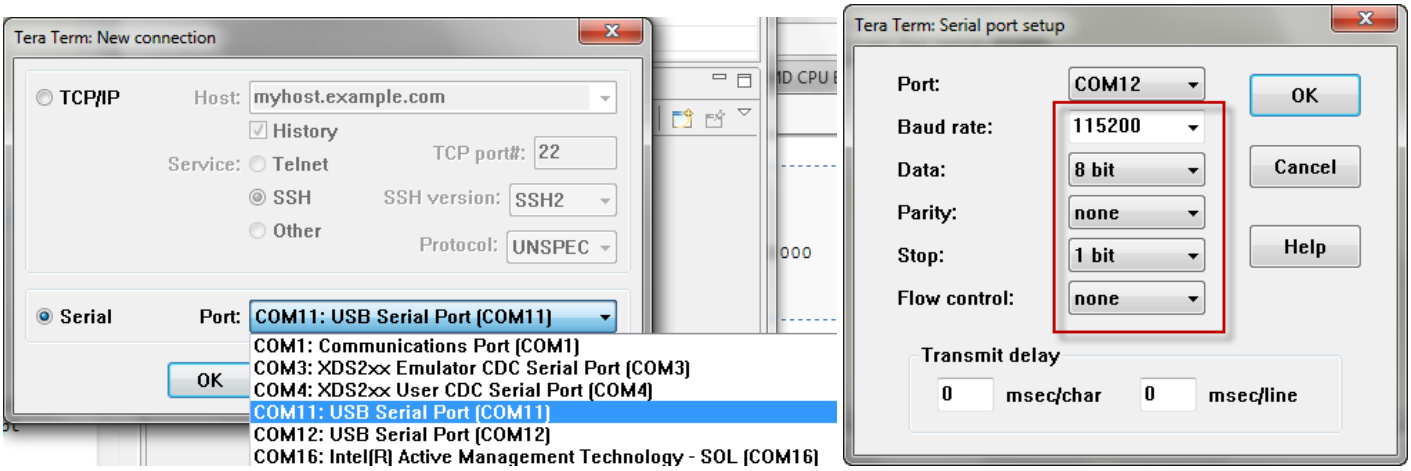
This makes the board in “no boot” mode for running U-boot through CCS.

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

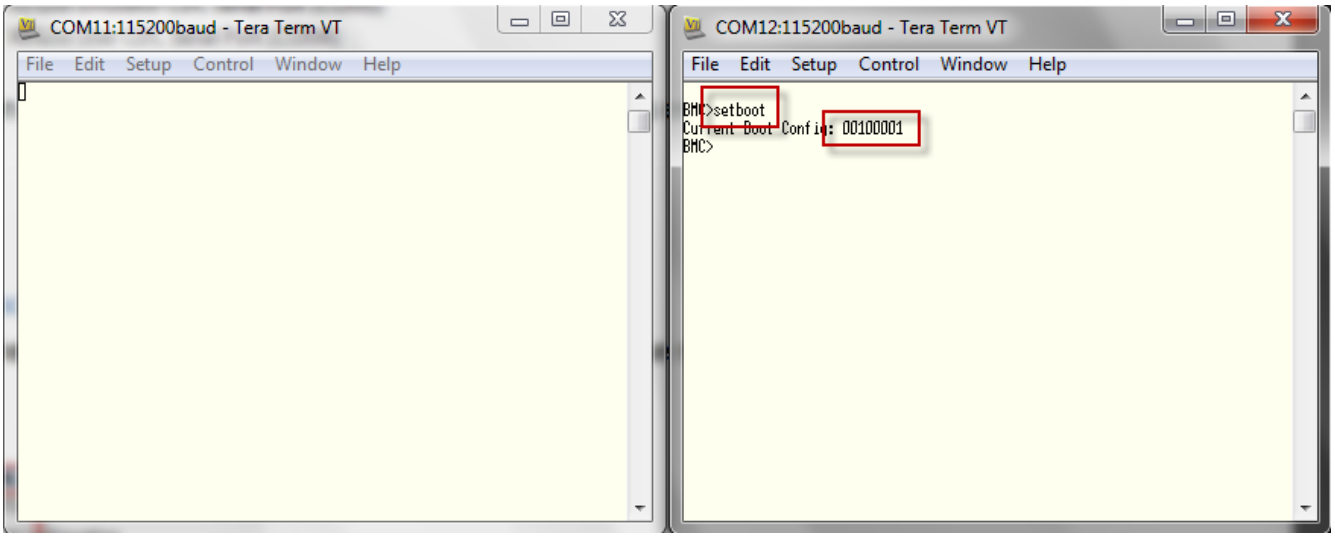
If the USB-COM drivers are correctly installed, you should found 2 more items were showed in your device manager:



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.

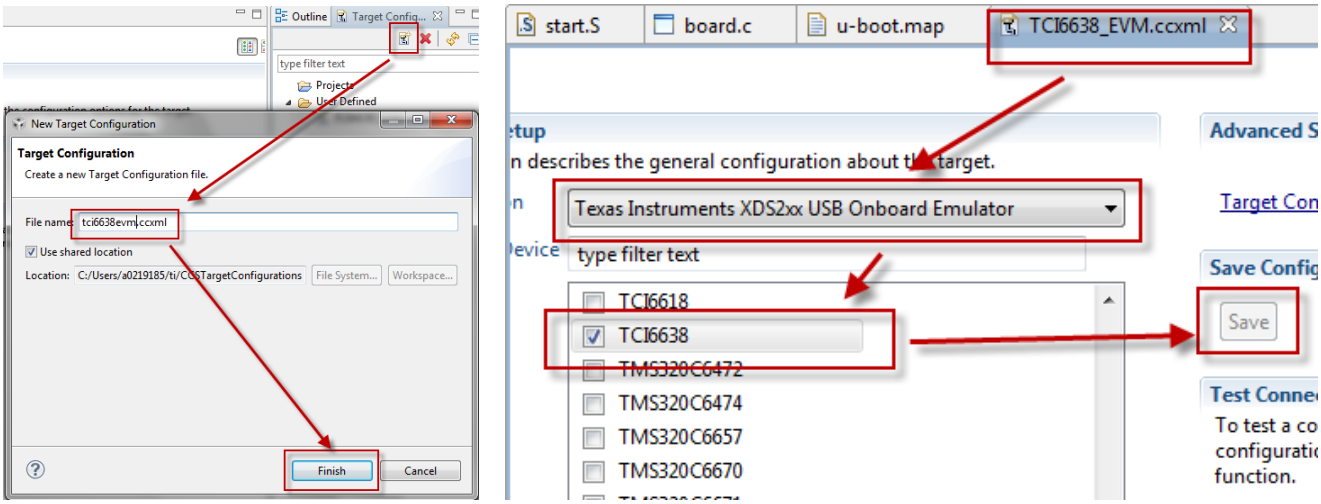
Please note that the DIP switch only takes effect on the first power on when connecting the power cable, other boot modes can be decided by the “setboot” commands on BMC and take effect after the command “fullrst”.

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.

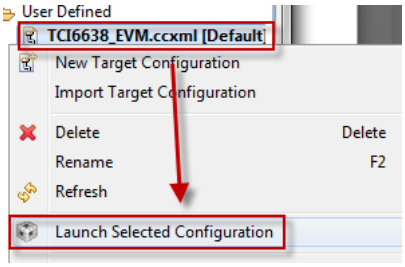
For more information about the boot mode DIP switch and the BMC encoding, please refer to the section “Appendix”

4.2 Create and Launch Debug Session and Connect Target

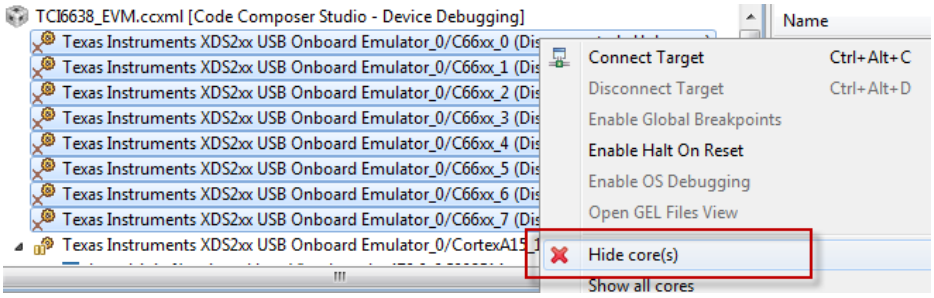
Create the .ccxml file in CCS as the follow steps:



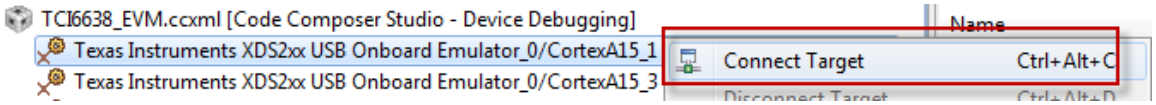
Launch the TCI6638 EVM debug session on CCSv5, please make sure do not use any ARM core GEL files.



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

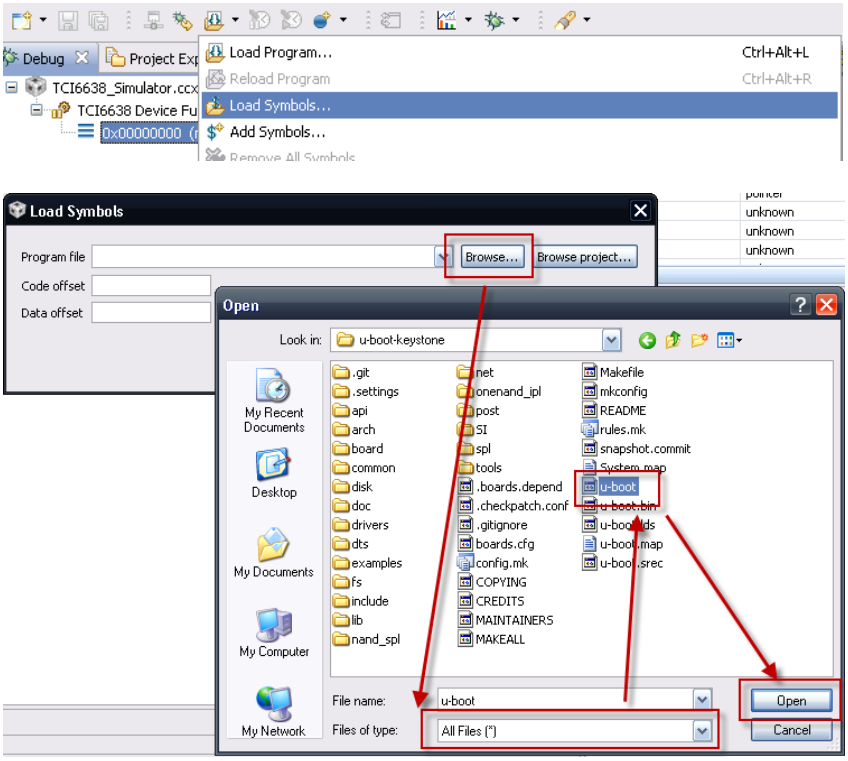


Then connect to the first ARM Core:

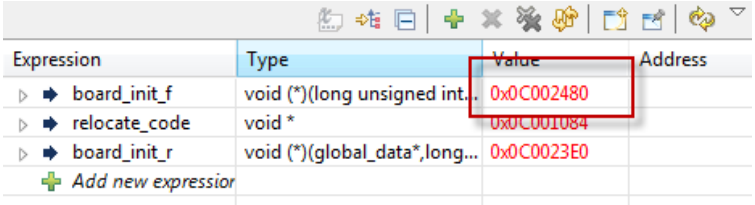


4.3 Load Image on the Target and Run

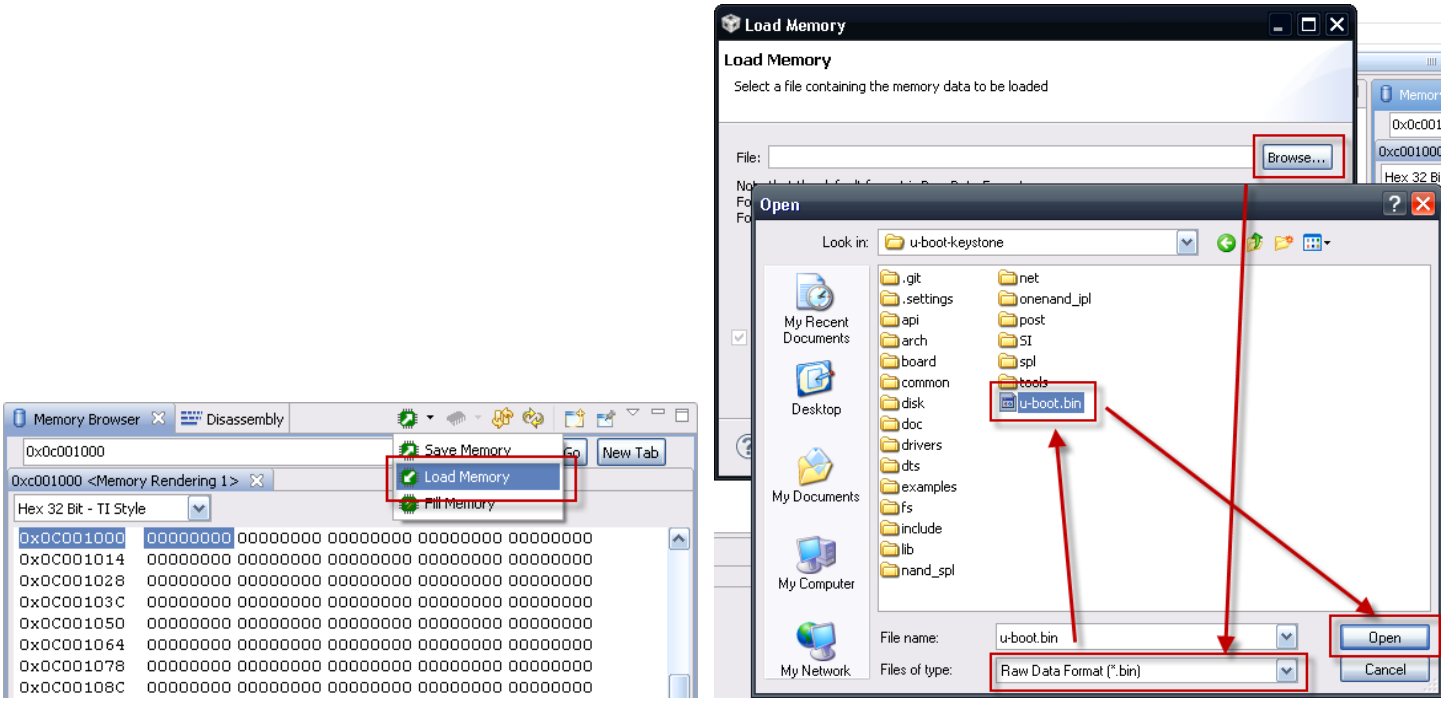
Launch the Load the file with no extension “u-boot” on target which can let the CCS recognize the symbol table:

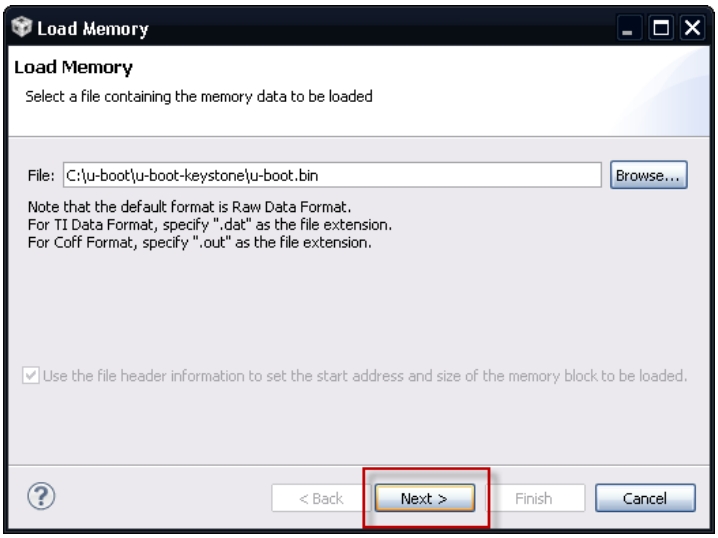


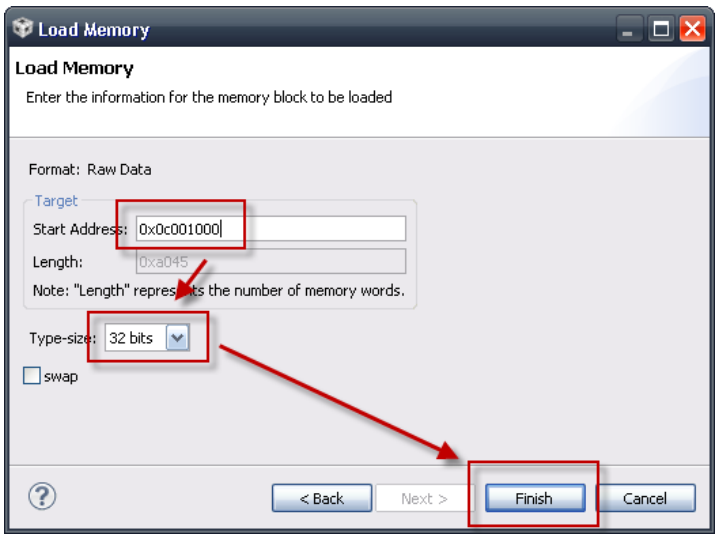
After you load the symbol table, some of the symbols/functions which defined in U-boot image can be recognized by CCS:



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

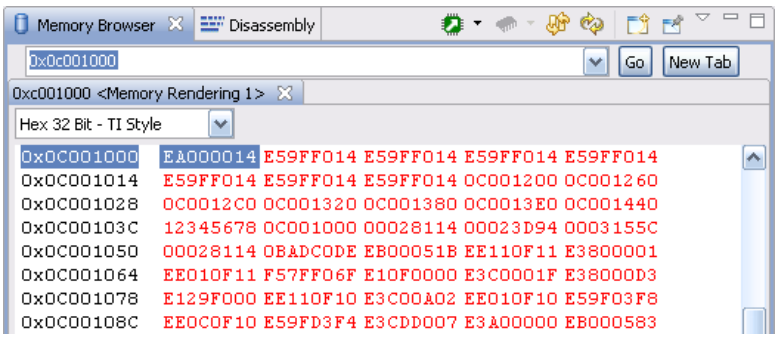




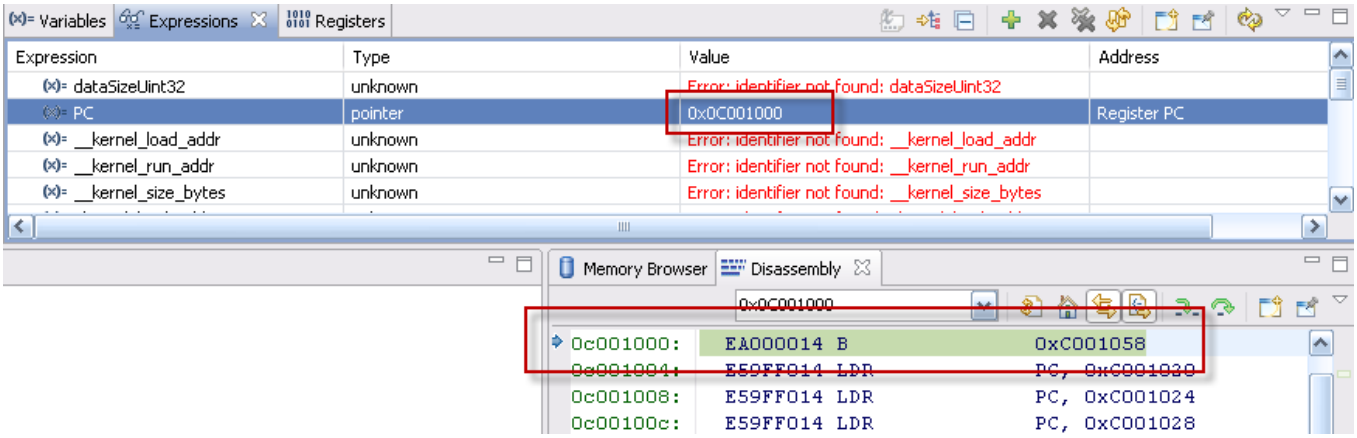


Then choose the load address “0x0C00_1000” (this address is the default relocate address which on RAM of SoC) and choose the 32-bit load mode.

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.

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

5.1 Burn U-boot Image on TCI6638 EVM SPI Flash

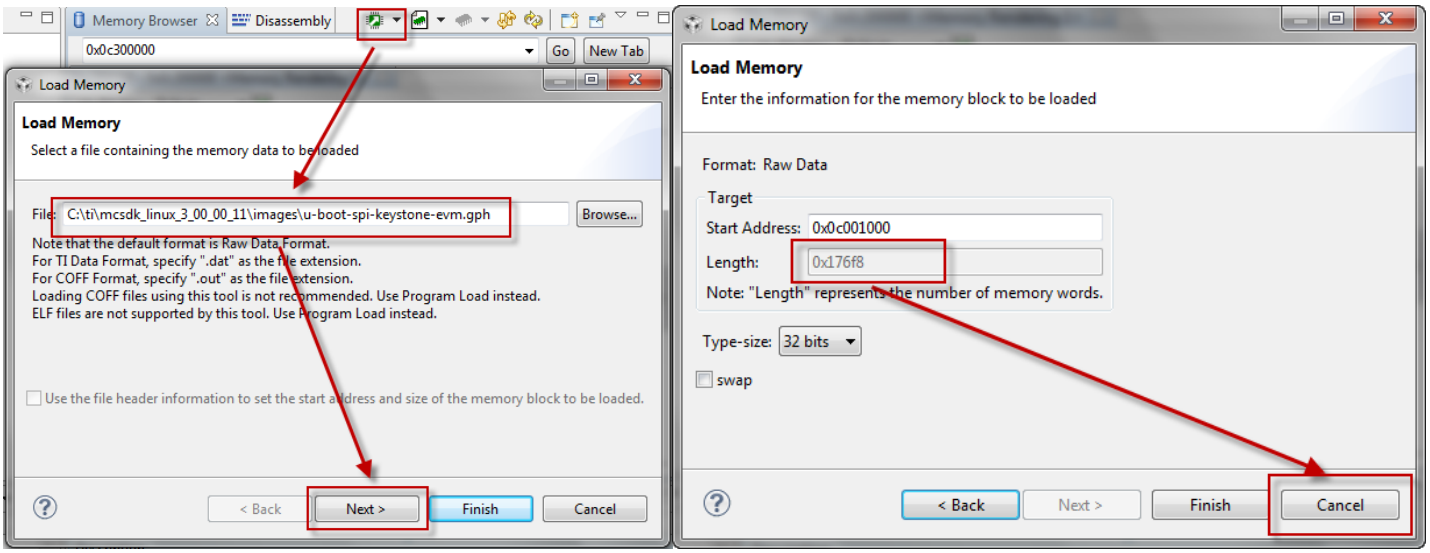
Before execute any steps in this section, please make sure that the U-boot already running on TCI6638 EVM.

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

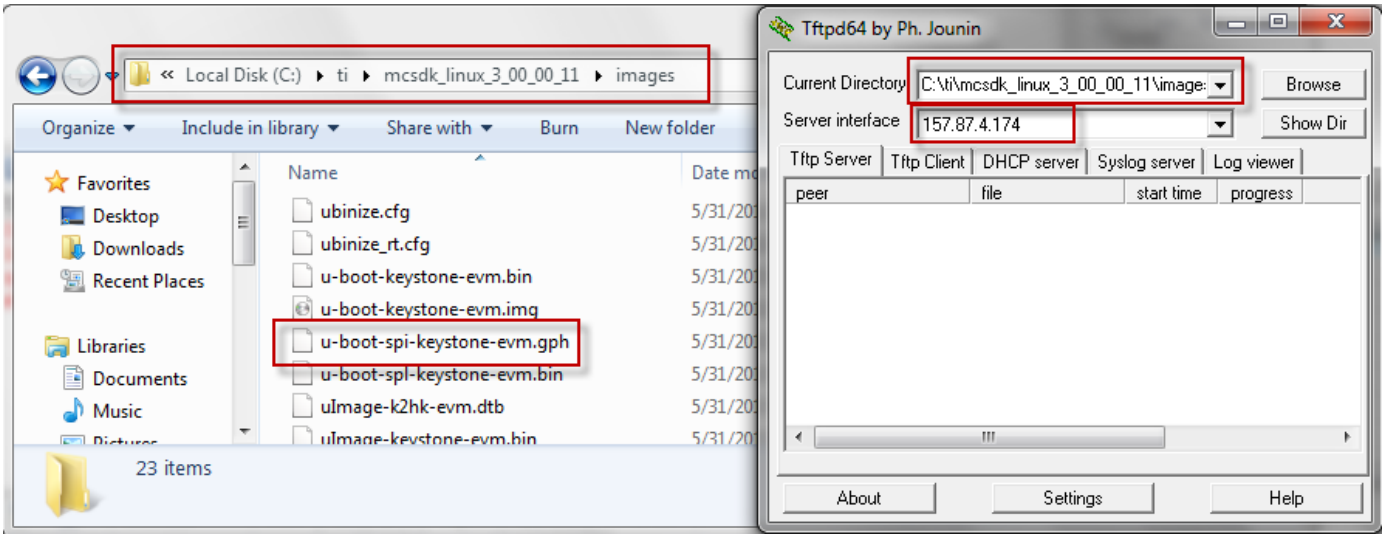
http://tftpd32.jounin.net/tftpd32_download.html

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.

Before we burn the gph file, we need to get the HEX size of the file by using the CCS tools:



Then get the .gph file path and your PC IP address and configure the tftp tools:



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

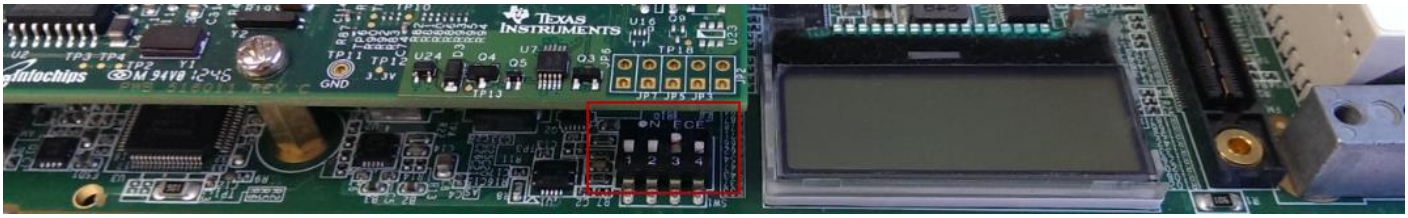
```
setenv serverip <your tftp server IP set in the tftp SW>

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 (5dbe0 hex)
TCI6638 EVM # run burn_uboot
SF: Detected N25Q128A 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:

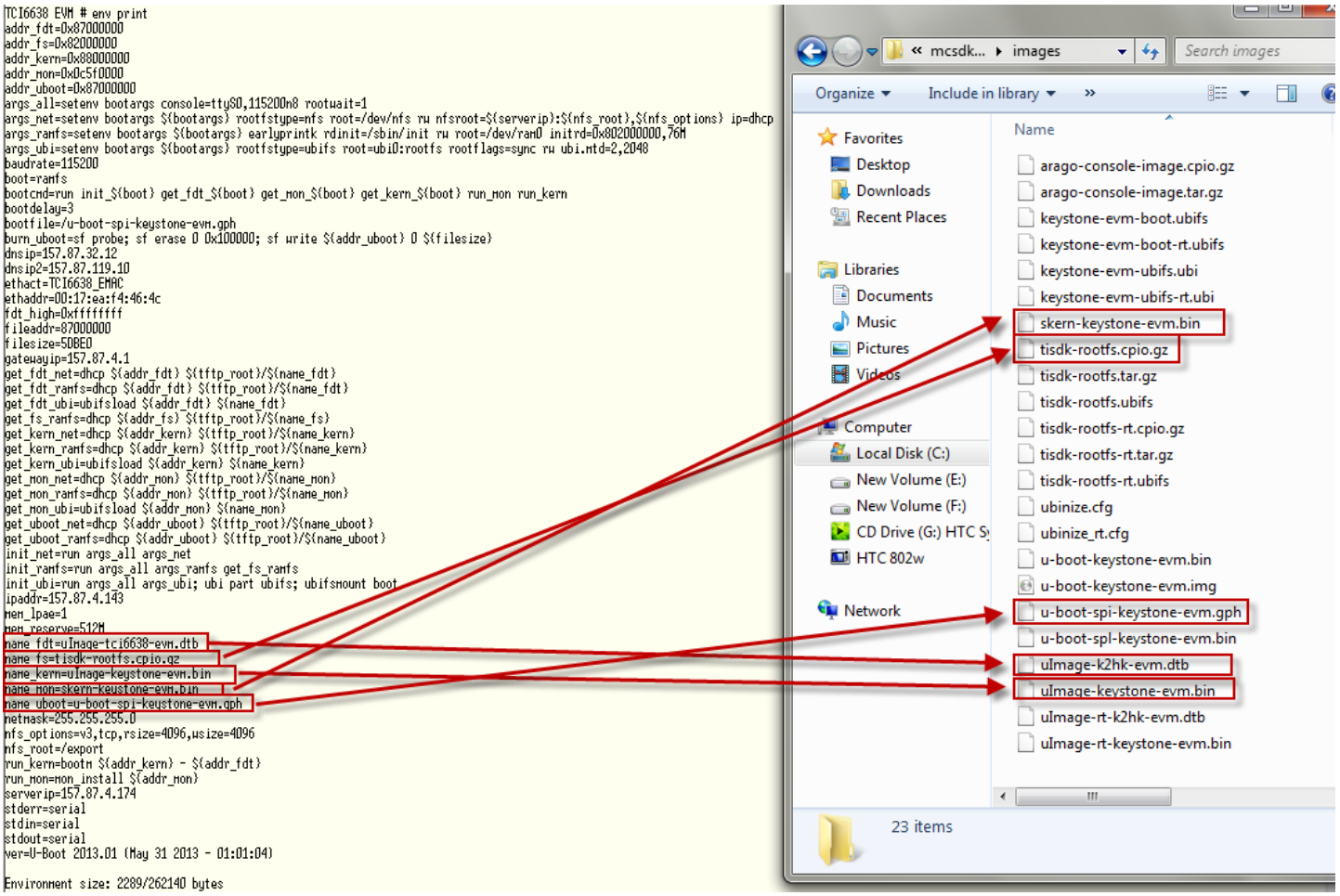
6. Load and Run Linux kernel on TCI6638 EVM through U-boot

6.1 U-boot Environment Settings for Run Linux Kernel

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 files.

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

```
env print
```

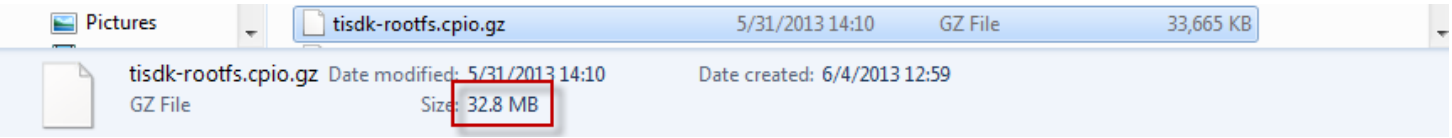


It'll print all the environment variables which current u-boot recorded; we need to modify the unmatched item to the corresponding values, such as:

```
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=0x802000000,9M", 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=0x802000000,33M'
```



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

```
saveevn
```

```
TCI6638 EVM # saveenv
Saving Environment to NAND...
Erasing Nand...
Erasing at 0x120000 -- 100% complete.
Writing to Nand... done
TCI6638 EVM #
```

6.2 Load and Run Linux Kernel by U-boot

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


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
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
Loading: #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #################################################################
         #########
         918 KiB/s
done
Bytes transferred = 34472931 (20e03e3 hex)
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 157.87.4.143
Using TCI6638_EMAC device
TFTP from server 157.87.4.174; our IP address is 157.87.4.143
Filename '/uImage-k2hk-evm.dtb'.
Load address: 0x87000000
Loading: ####
         710 KiB/s
done
Bytes transferred = 54537 (d509 hex)
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 '/skern-keystone-evm.bin'.
Load address: 0xc5f0000
Loading: ####
         757.8 KiB/s
done
Bytes transferred = 45056 (b000 hex)
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 157.87.4.143
Using TCI6638_EMAC device
TFTP from server 157.87.4.174; our IP address is 157.87.4.143
Filename '/uImage-keystone-evm.bin'.
Load address: 0x88000000
Loading: ##################################################################
         ##################################################################
         ##################################################################
         ###############################################
         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  : 0xffff0000 - 0xffff1000   (   4 kB)
[    0.000000]     fixmap  : 0xfff00000 - 0xfffe0000   ( 896 kB)
[    0.000000]     vmalloc : 0xe0800000 - 0xff000000   ( 488 MB)
[    0.000000]     lowmem  : 0xc0000000 - 0xe0000000   ( 512 MB)
[    0.000000]     pkmap   : 0xbfe00000 - 0xc0000000   (   2 MB)
[    0.000000]     modules : 0xbf000000 - 0xbfe00000   (  14 MB)
[    0.000000]       .text : 0xc0008000 - 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
128
[    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
MT29F4G08ABBDAHC), 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] 0x000000000000-0x000000100000 : "u-boot"
[    4.566832] 0x000000100000-0x000000180000 : "params"
[    4.572804] 0x000000180000-0x000008000000 : "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] 0x000000000000-0x000000080000 : "u-boot-spl"
[    4.625422] 0x000000080000-0x000001000000 : "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) SGMII 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
udhcpc (v1.20.2) started
Sending discover...
Sending select for 157.87.4.143...
Lease of 157.87.4.143 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
[    7.906739] keystone-netcp 2090000.netcp: initializing cpsw version 1.3 (1) SGMII
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
udhcpc (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-rltraceroute6.
Configuring ndisc6-rdisc6.
INIT: Entering runlevel: 5
Starting system message bus: dbus.
Starting Dropbear SSH server: Will output 1024 bit rsa secret key to
'/etc/dropbear/dropbear_rsa_host_key'
Generating key, this may take a while...
Public key portion is:
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAAAgn2RM75uXHDi+KkwLV2nQepQuXQuV9KbC+ZyNpdjbBRu1Y5tmqEX7FTsTA049
0grnuM8bMra1Yxy5vvvH5w24THU3M1eXofDtbP0vq5opw6/XUAcxnB3qk/gXeC17VofJkwriRJPRHrBpvNkR0tR6H
lMz1GtKT2b1xk0lORcEFvgUeU= 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:
```

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])

DIPSW	Title	Boot Value(hex)	Comments
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.

DIPSW	Title	Boot Value(hex)	Comments
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
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

DIPSW	Title	Boot Value(hex)	Comments
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	
23				Reserved for future boot config latched values
22		AVSIFSEL1	TIMI1	Reserved: EVM forces these bits to strap values during reset
21		AVSIFSEL0	TIMI0	""
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	

Bit	Devstat Bit	Config Pin Function	Normal Pin Function	Comments
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
```