

Quick start with 3.14 linux

Contents

Kernel

- Getting Source
- Building Kernel

U-Boot

- Getting Source
- Building U-boot

Bootting the EVM

- Preparing SD card
- Filesystem
- Copy Images
- Modifying U-Boot environment variables
 - choosing the dtb file
 - Using NFS as Root filesystem
 - Using TFTP for downloading Images
 - Backup of boot environment
- Building Modules

Kernel

Getting Source

Clone the kernel git repo from git.ti.com

Please note here i'm using the LCPD kernel repo,

```
git clone git://git.ti.com/ti-linux-kernel/ti-linux-kernel.git 3.14
cd 3.14
git branch -r
git checkout -b ti-linux-3.14.y origin/ti-linux-3.14.y
```

To build the GLSDK kernel, you should clone [git://git.omapzoom.org/kernel/omap.git](https://git.omapzoom.org/kernel/omap.git) and checkout p-ti-linux-3.14.y

Building Kernel

Use the following command to first build the config file, by applying all required config fragments [NEW in 3.14]

```
./scripts/kconfig/merge_config.sh -m arch/arm/configs/omap2plus_defconfig ti_config_fragments/ipc.cfg ti_config_fragments/power.cfg
ti_config_fragments/audio_display.cfg ti_config_fragments/system_test.cfg ti_config_fragments/baseport.cfg ti_config_fragments/wlan.cfg
ti_config_fragments/connectivity.cfg ti_config_fragments/auto.cfg ti_config_fragments/dra7_only.cfg
```

make olddefconfig ARCH=arm

Note: ti_config_fragments directory is present in the top level directory. You may chose not to apply any of the config fragments if you dont want it, for ex: ti_config_fragments/system_test.cfg, enables KMEMLEAK AND RTC DEBUG, which may not be required if you are not running any debug tests.

It is good to export the ARCH and CROSS_COMPILE variable to environment or pass it in every command

```
make -j4 ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- zImage
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- dra7-evm.dtb
```

U-Boot

Getting Source

Get teh U-boot source form the lcpd git repo

```
git clone git://git.ti.com/ti-u-boot/ti-u-boot.git 2014.07
cd 2014.07
git branch -r
git checkout -b ti-u-boot-2014.07 origin/ti-u-boot-2014.07
```

Building U-boot

Use the default config for dra7xx evm

```
make dra7xx_evm_config
make
```

(This will create MLO and u-boot.img in current directory - 2014.07)

Booting the EVM

Preparing SD card

Format sd card (>= 2GB) to have two partitions, one fat32 of size ~64MB and name "boot" another ext3/4 partition of size atleast 1.5G with name "rootfs"

If you have trouble partitioning the card use the *mksdboot.sh* script taht comes with GLSDK installer. Refer to step 4 of [1] (http://processors.wiki.ti.com/index.php/DRA7xx_GLSDK_Software_Developers_Guide#Starting_your_software_development)

Filesystem

```
wget http://swubn01.india.ti.com/publish/GLSDK/6_10_02/exports/arago-glsdk-multimedia-image-dra7xx-evm.tar.gz
```

Extract to sd card

```
sudo tar -zxf arago-glsdk-multimedia-image-dra7xx-evm.tar.gz -C /media/rootfs
```

Copy Images

Copy the u-boot binaries (MLO and u-boot.img) to "boot" partition of SD card

```
cp MLO u-boot.img /media/boot
```

and copy the kernel binaries to filesystem direcorey "/boot" (boot directory in the rootfs)

```
sudo cp arch/arm/boot/zImage arch/arm/boot/dts/dra7-evm.dtb /media/rootfs/boot
sync
umount /media/rootfs /media/boot
```

Insert the sd card to mmc card slot and Power on **Weclome to Linux 3.14.. Happy De-bugging..**

Modifying U-Boot environment variables

U-boot boots up with default environment variables hardcoded in teh U-boot source code. To see what is the configured environment type "print" at the U-Boot prompt.

```
U-Boot#printenv
or
U-Boot#print
for a particluar variable
U-Boot#printenv mmcargs
replace mmcargs by the variable name you want to see.
```

To modify any of the environment variables such as loadaddr, fdtfile etc you can create a "uenv.txt" and place it in the "boot" partition of the SD card.

choosing the dtb file

- To use the device tree blob (dtb) for 7" lcd panel on Revision <= F, create a file "uenv.txt"

```
vi uenv.txt
```

Add following line to "uenv.txt"

```
fdtfile=dra7-evm-lcd7.dtb
```

save the file.

- For 10" lcd panel on Revision >= G

```
fdtfile=dra7-evm-lcd10.dtb
```

save the file.

Using NFS as Root filesystem

- To update the bootargs to use with *nfs*

```
mmcargs=setenv bootargs 'console=tty00,115200n8 vram=16M root=/dev/nfs rw nfsroot=<ip>:/srv/rootfs_path rootwait ip=dhcp'
```

Using TFTP for downloading Images

serverip=<172.24.*.*, ip address of the tftp server>
ipaddr=<172.24.*.*, give a temp ip address to use, should be the same subnet as the actual ip you get>
loadfdt=tftp ox88000000 dra7-evm-lcd10.dtb
loadimage=tftp ox82000000 zImage

Copy this file to the boot partition of the SD card

```
cp uenv.txt /media/boot
```

10' panel looks like a Tablet surface.. try it sometime

Backup of boot environment

```
U-Boot 2014.07 (Feb 03 2015 - 23:38:08)

CPU   : DRA752 ES1.0
Board: DRA7xx
I2C:   ready
DRAM:  1.5 GiB
NAND:  0 MiB
MMC:   OMAP SD/MMC: 0, OMAP SD/MMC: 1
Using default environment

SATA link 0 timeout.
AHCI 0001.0300 32 slots 1 ports 3 Gbps 0x1 impl SATA mode
Flags: 64bit ncq stag pm led clo only pmp pio slum part ccc apst
scanning bus for devices...
Found 0 device(s).
SCSI:  Net:  <ethaddr> not set. Validating first E-fuse MAC
cpsw
Hit any key to stop autoboot:  0
U-Boot# pr
arch=arm
baudrate=115200
board=dra7xx
board_name=dra7xx
boot_os=0
bootcmd=run findfdt; run mmcboot;setenv mmcdev 1; setenv bootpart 1:2; setenv mmcroot /dev/mmcblk0p2 rw; run mmcboot;run nandboot;
bootdelay=1
bootdir=/boot
bootfile=zImage
bootm_size=0x10000000
bootpart=0:2
bootscrip=echo Running bootscrip from mmc${mmcdev} ...; source ${loadaddr}
console=tty00,115200n8
cpu=armv7
dfu_alt_info_emmc=rawemmc raw 0 3751936;boot part 1 1;rootfs part 1 2;MLO fat 1 1;MLO.raw raw 0x100 0x100;u-boot.img.raw raw 0x300 0x400;spl-os-args.raw raw 0x80 0x80;spl-os-image.raw raw 0x900 0x2000;spl1
dfu_alt_info_mmc=boot part 0 1;rootfs part 0 2;MLO fat 0 1;MLO.raw raw 0x100 0x100;u-boot.img.raw raw 0x300 0x400;spl-os-args.raw raw 0x80 0x80;spl-os-image.raw raw 0x900 0x2000;spl-os-
args fat 0 1;spl-os1
dfu_alt_info_ram=kernel ram 0x80200000 0x4000000;fdt ram 0x80f80000 0x800000;ramdisk ram 0x81000000 0x4000000
dfu_buflsiz=0x10000
eth1addr=20:cd:39:2b:e4:1f
ethact=cpsw
ethaddr=20:cd:39:2b:e4:1e
fdt_addr_r=0x88000000
fdtaddr=0x88000000
fdtfile=undefined
findfdt=if test ${board_name} = omap5-uevm; then setenv fdtfile omap5-uevm.dtb; fi; if test ${board_name} = dra7xx; then setenv fdtfile dra7-evm-lcd7.dtb; fi;if test ${board_name} = dra72x;
then setenv fdtfile
importbootenv=echo Importing environment from mmc${mmcdev} ...; env import -t ${loadaddr} ${filesize}
kernel_addr_r=0x82000000
loadaddr=0x82000000
loadbootenv=fatload mmc ${mmcdev} ${loadaddr} uEnv.txt
loadbootscrip=fatload mmc ${mmcdev} ${loadaddr} boot.scr
loadfdt=load mmc ${bootpart} ${fdtaddr} ${bootdir}/${fdtfile};
loadimage=load mmc ${bootpart} ${loadaddr} ${bootdir}/${bootfile}
mmcargs=setenv bootargs console=${console} ${optargs} vram=${vram} root=${mmcroot} rootfstype=${mmcrootfstype}
mmcboot=mmc dev ${mmcdev}; if mmc rescan; then echo SD/MMC found on device ${mmcdev};if run loadbootenv; then echo Loaded environment from ${bootenv};run importbootenv;fi;if test -n
$uenvcmd; then echo Ru;
mmcdev=0
mmcroot=/dev/mmcblk0p2 rw
mmcrootfstype=ext4 rootwait
mtdds=nand0=nand.0
mtdparts=mtdparts=nand.0:128k(NAND.SPL),128k(NAND.SPL.backup1),128k(NAND.SPL.backup2),128k(NAND.SPL.backup3),256k(NAND.u-boot-spl-os),1m(NAND.u-boot),128k(NAND.u-boot-env),128k(NAND.u-
boot-env.backup1),8m)
nandargs=setenv bootargs console=${console} ${optargs} root=${nandroot} rootfstype=${nandrootfstype}
nandboot=echo Booting from nand ...; run nandargs; nand read ${fdtaddr} NAND.u-boot-spl-os; nand read ${loadaddr} NAND.kernel; bootz ${loadaddr} - ${fdtaddr}
nandroot=ubi0:rootfs rw ubi.mtd=NAND.file-system,2048
nandrootfstype=ubifs rootwait=1
partitions=uuid_disk=${(uuid_gpt_disk)};name=rootfs,start=2MiB,size=-,uuid=${(uuid_gpt_rootfs)}
ramdisk_addr_r=0x88000000
rdaddr=0x88000000
scsides=0
soc=omap5
stderr=serial
stdin=serial
stdout=serial
usbttty=cdc_acm
vendor=ti
ver=U-Boot 2014.07 (Feb 03 2015 - 23:38:08)
vram=16M
```

Building Modules

To build and use modules

```
mkdir ./modules
make modules
make modules_install INSTALL_MOD_PATH=./modules

cp -rf ./modules/lib/modules/3.14-* /media/rootfs/lib/modules/
```

Keystone=

{{

1. switchcategory:MultiCore=

■ For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum

■ For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **Quick start with 3.14 linux** here.

■ For technical support on MultiCore devices, please post your questions in the C6000 MultiCore Forum

■ For questions related to the BIOS MultiCore SDK (MCSDK), please use the BIOS Forum

Please post only comments related to the article **Quick start with 3.14 linux** here.

Please post only comments related to the article **Quick start with 3.14 linux** here.

C2000=For technical support on the C2000 please post your questions on The C2000 Forum. Please post only comments about the article **Quick start with 3.14 linux** here.

DaVinci=For technical support on DaVinciplease post your questions on The DaVinci Forum. Please post only comments about the article **Quick start with 3.14 linux** here.

MSP430=For technical support on MSP430 please post your questions on The MSP430 Forum. Please post only comments about the article **Quick start with 3.14 linux** here.

OMAP35x=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **Quick start with 3.14 linux** here.

OMAPL1=For technical support on OMAP please post your questions on The OMAP Forum. Please post only comments about the article **Quick start with 3.14 linux** here.

MAVRK=For technical support on MAVRK please post your questions on The MAVRK Toolbox Forum. Please post only comments about the article **Quick start with 3.14 linux** here.

For technical support on please post your questions at <http://e2e.ti.com>. Please post on comments about article **Quick start with 3.14 linux** here.

Links



[Amplifiers & Linear Audio](#)
[Broadband RF/IF & Digital Radio](#)
[Clocks & Timers](#)
[Data Converters](#)

[DLP & MEMS High-Reliability Interface](#)
[Logic](#)
[Power Management](#)

Processors

- [ARM Processors](#)
- [Digital Signal Processors \(DSP\)](#)
- [Microcontrollers \(MCU\)](#)
- [OMAP Applications Processors](#)

[Switches & Multiplexers](#)
[Temperature Sensors & Control ICs](#)
[Wireless Connectivity](#)

Retrieved from "https://processors.wiki.ti.com/index.php?title=Quick_start_with_3.14_linux&oldid=203972"

This page was last edited on 27 July 2015, at 03:43.
Content is available under [Creative Commons Attribution-ShareAlike](#) unless otherwise noted.