LPC313x Linux Quick Start Guide

1 Introduction

This quick-start guide explains the steps necessary to get Linux up and running on the Embedded Artist's LPC313x board using LTIB or ELDK frameworks. This guide is meant for users new to Linux or as a method to get a Linux image up and running fast. The current LPC313x Linux BSP supports all the following chips:

- LPC313x series (LPC3131 / LPC3130) : Tested on EA3131 board.
- LPC314x series (LPC3143 / LPC3141) : Tested on NXP internal VAL3153 board.
- LPC315x series (LPC3154 / LPC3152) : Tested on NXP internal VAL3154 board.

As per http://www.bitshrine.org/, LTIB is explained as:

"The LTIB (Linux Target Image Builder) project is a simple tool that can be used to develop and deploy BSPs (Board Support Packages) for various target platforms. Using this tool a user will be able to develop a GNU/Linux image for their target platform."

LTIB handles a lot of the work of building a complete Linux system, such as setting up the root filesystem, package selection, system init configuration, etc. Advanced users that prefer not to use LTIB may be interested in just getting the necessary kernel and bootloader patches and starting from there – please see the ELDK section of this guide for the locations of those patches.

As per http://www.denx.de/wiki/DULG/ELDK, ELDK is explained as:

The Embedded Linux Development Kit (ELDK) includes the GNU cross development tools, such as the compilers, binutils, gdb, etc., and a number of pre-built target tools and libraries necessary to provide some functionality on the target system. It is provided for free with full source code, including all patches, extensions, programs and scripts used to build the tools. Some versions of ELDK are available in two versions, which use Glibc or uClibc as the main C library for the target packages. Packaging and installation is based on the RPM package manager.

Note: The description in this document assumes using ELDK version 4.2 for ARM.

The steps covered include the following:

- Using LTIB framework
  - Getting and installing LTIB
  - Configuring LTIB and the build process
- Using ELDK framework
  - Getting and installing ELDK
  - Configuring & building Apex
  - Configuring and building the Linux kernel
- Boot process

- Loading Apex from
  - UART
  - SD/MMC card
  - On board SPI-NOR flash
  - On board NAND flash
  - USB DFU
- Loading kernel & ramdisk using Apex boot-loader from
  - On board NAND flash
  - SD/MMC card (EXT2 formatted)
  - TFTP boot (available for EA LPC31xx Version 2.0 base boards only)
  - UART (Xmodem protocol)
- Porting Apex to LPC313x custom board
- Porting Linux 2.6.x to LPC313x custom board

From start to finish, the process can take anywhere from a few hours to a day depending on factors such as host machine speed, internet connection speed, current host system configuration, etc.

1.1 Copyrights and limitations

The LPC313x BSP is provided free of charge and with no support from NXP. Portions of the BSP are copyrighted by NXP Semiconductors.

1.2 Where to start

The sections in this guide are meant to be followed in sequential order starting with Section 2. Prior to that, the requirements for the host, target, and development environment should be reviewed in Sections 1.2 to 1.4.

1.3 Host system requirements

To develop Linux for the LPC313x, a host PC running the Linux operating system is needed. Because of the many variations of Linux releases and supported packages, it is unknown if the tools included with the BSP will work correctly on a specific release of Linux.

LTIB and the supporting tools have been tested with the Fedora 9 Linux release. Although other releases may work fine, they are currently untested. Your host machine will also need a connection to the internet to download packages for the target system. Packages are downloaded and cached on the host machine as needed.

ELDK and the supporting tools have been tested with the Ubuntu 8.04 (Hardy Heron) and Fedora 9 Linux release. Although other releases may work fine, they are currently untested.

1.3.1 Additional host machine software requirements

As LTIB is installing itself, it may fail due to missing host system packages. If this occurs, carefully read the LTIB error messages and install any other packages on your host machine required by LTIB.

Similarly, as ELDK is installing itself, it may fail due to missing host system packages. If this occurs, carefully read the ELDK error messages and install any other packages on your host machine required by ELDK. For additional help refer to http://www.denx.de/wiki/DULG/ELDK.

1.4 Target board requirements

Embedded Artists' LPC3131 OEM Board (mounted on the LPC31xx Base Board) is required to run the generated bootloader and kernel images. The board is available from Embedded Artists' at http://www.embeddedartists.com/products/uclinux/oem_lpc3131_bundle.php.

A newer version (2.0) of base boards is also available with on board DM9000 Ethernet controller for Linux development.

1.5 Development environment requirements

The target boot configuration is setup to support RARP, TFTP and DHCP. A RARP, TFTP & DHCP servers should be available on the network where your Linux target board is plugged in. If a DHCP or RARP server isn't available, enter a manual IP configuration for the target network configuration in step 4.2.3.

An SD or MMC card is also needed to initially setup the bootloader on the EA313x board.

1.6 Basic required Linux skills

Skills such as TFTP boot configuration, basic network setup, mounting and using a storage card (such as SD/MMC), and installing new host system packages are a few of the skills assumed to be known by the user in this guide.

There are already a lot of resources on the internet explaining these types of things so they won't be covered here in this guide.

2 Using LTIB framework

2.1 Getting and installing LTIB

Getting and installing LTIB is easy! Open a web browser on your host machine and go to the http://www.bitshrine.org website. Look for the section for LTIB installation. Follow the directions there to install LTIB. It is highly recommended to use the netinstall script to install LTIB or use a direct CVS download (explained in the LTIB FAQ). Using a fixed snapshot image may not get you the latest files and will make updates harder to manage.

LTIB will begin installing by downloading files and packages over the internet. If any errors occurred during installation, correct the error before continuing. Most errors are due to missing or out-of-date packages on the host machine. If you do see missing packages, they can be added or updated with the host machine's software management tools.

Once LTIB has finished installing, run LTIB per the website instructions. A menu should appear requesting selection of the platform. Continue to the next section to setup LTIB to build Linux for the EA3131 board.

2.2 Configuring LTIB

Once LTIB has been installed and executed for the first time, a menu will appear requesting which platform to use with LTIB, Select the Embedded Artists board with the NXP LPC3131 SoC option. Choose the Exit option and save the configuration to continue the setup process.

Another menu should shortly appear that allows you to customize the LTIB build options for the EA3131 platform. If this is the first time running LTIB, a number of default options are already selected for you that can be used to build a working Linux system.

Although the default options are fine for the first build, you will need to verify that your board version matches the supported version in the kernel. You can change or add build options by selecting LTIB menu items. For example, if you select the "Configure Apex" or "Configure the kernel" options from the LTIB menu, the Apex and Linux kernel menus will appear allowing customization of their settings during the build cycle.

2.3 LTIB build process

The LTIB build process for the EA3131 board builds the Apex bootloader, Linux kernel image, and root filesystem. The Apex bootloader and kernel image are built with the toolchain selected in the LTIB main menu and the default configurations provided with LTIB or selected by the user from the Apex and kernel configuration menus.

As the Linux system is being built, the directory called "rootfs" will be built under the ./ltib directory. This directory contains the raw images of the root filesystem. The Apex bootloader (apex.bin) and the kernel image (zImage) will be placed in the ./rootfs/boot directory.

Files and packages will be built and placed in the rootfs directory's subdirectories based on the package selections in the LTIB package selection menu. These include packages such as busybox, mp3play, mtd-utils, etc.

Based on the "Target System Configuration" LTIB menu item, the startup process of the Linux kernel may be altered. This section of the LTIB menu can be used to adjust network settings or starting services.

Once the rootfs directory has been completely built, the "Target Image Generation" LTIB menu item specifies the type of deployment the Linux system will use on the EA3131 board. If NFS is selected, the root filesystem directory at ./ltib/rootfs is used as the root filesystem mount point. If another option is used such as JFFS2 or ramdisk, another file will be created to be used for deployment.

To save time rebuilding the kernel and Apex, check the “Leave the Apex sources after building” and “Leave the kernel sources after building” options.

3 Using ELDK framework

For ease of development following directory structure is suggested. Rest of the section assumes this directory structure in its command illustrations.

Create the following structure in your development area:

- /home/xxx/projects/lpc313x
  - /home/xxx/projects/lpc313x/eldk42
  - /home/xxx/projects/lpc313x/apex
    - /home/xxx/projects/lpc313x/apex/work_1.6.8 <-- this will hold the current working sources for apex
  - /home/xxx/projects/lpc313x/kernel
    - /home/xxx/projects/lpc313x/kernel/work_2.6.28.2 <-- this will hold the current working sources for Linux
  - /home/xxx/projects/lpc313x/patches
  - /home/xxx/projects/lpc313x/temp_dir
  - /home/xxx/projects/lpc313x/downloads <- put all tarballs etc you download from internet here

3.1 Getting and installing ELDK

- Download the latest ISO image of ELDK4.2 for ARM at ftp://ftp.denx.de/pub/eldk/4.2/arm-linux-x86/iso/arm-2008-11-24.iso.
- Follow the instruction listed in ftp://ftp.denx.de/pub/eldk/4.2/arm-linux-x86/distribution/README.html. It is pretty simple. Mount the .iso file using

```
>sudo mount -o loop downloads/arm-2008-11-24.iso temp_dir
```

- CD to temp_dir and run the install script and pass the directory where you want to install the tools. Note, the iso contains both pre-built GCC4.2.2 for ARM and also pre-built root file system for ARM target. We will use the pre-built rootfs during our development.

```
>cd temp_dir
>sudo ./install -d /home/xxx/projects/lpc313x/eldk42
```

- Run the ELDK script described in section "1.6. Mounting Target Components via NFS" of README.html file for future NFS root development.

3.2 Building Apex boot-loader

- Download the Apex1.6.8 tarball from ftp://ftp.buici.com/pub/apex/apex-1.6.8.tar.gz.

- Untar the apex the sources.

```
>cd apex
>tar -xzf ../downloads/apex-1.6.8.tar.gz
>mv apex-1.6.8 work_1.6.8
```

- Apply LPC313x apex patch present as part of the release tarball.

```
>cat ../patches/apex-1.6.8_lpc313x.patch | (cd work_1.6.8; patch -p1)
```

- Prior to building apex add ELDK tools to your path. From bash shell do

```
>export ARCH=arm
>export CROSS_COMPILE=arm-linux-
>export
PATH=/home/xxx/projects/lpc313x/eldk42/usr/bin:/home/xxx/projects/lpc313
x/eldk42/bin:$PATH
```

- To build apex assuming pwd = /home/xxx/projects/lpc313x

```
>make -C apex/work_1.6.8 ea313x_v1_config apex.bin
```

- Once build completes the apex binary image for deployment can be found at apex/work_1.6.8/apex.bin
- For systems with version 2.0 I/O boards use ea313x_v2_config.
- To change apex configuration issue the following command.

```
>make -C apex/work_1.6.8 ea313x_v1_config menuconfig
```

- Don't forget to backup the modified config.

```
>cp apex/work_1.6.8/.config apex/work_1.6.8/src/mach-lpc313x/myconfig

>make -C apex/work_1.6.8 myconfig apex.bin
```

3.3 Building kernel

- Download the kernel sources tarball from ftp://ftp.kernel.org/pub/linux/kernel/v2.6/linux-2.6.28.2.tar.bz2
- Un-tar the kernel the sources.

```
>cd kernel
>tar -xjf ../downloads/linux-2.6.28.2.tar.bz2
>mv linux-2.6.28.2 work_2.6.28.2
```

- Apply LPC313x kernel patch present as part of the release tarball.

```
>cat ../patches/linux-2.6.28.2_lpc313x.patch | (cd work_2.6.28.2; patch
-p1)
```

- Prior to building kernel add ELDK tools to your path. From bash shell do

```
>export ARCH=arm
>export CROSS_COMPILE=arm-linux-
>export
PATH=/home/xxx/projects/lpc313x/eldk42/usr/bin:/home/xxx/projects/lpc313
x/eldk42/bin:$PATH
```

- To build kernel assuming pwd = /home/xxx/projects/lpc313x

```
> make -C kernel/work_2.6.28.2 ARCH=arm CROSS_COMPILE=arm-linux-
ea313x_defconfig zImage
```

- Once build completes the kernel binary image for deployment can be found at kernel/work_2.6.28.2/arch/arm/boot/zImage
- To change kernel configuration issue the following command.

```
>make -C kernel/work_2.6.28.2 ARCH=arm CROSS_COMPILE=arm-linux-
ea313x_defconfig menuconfig
```

- Don't forget to backup the modified config.

```
>cp kernel/work_2.6.28.2/.config
kernel/work_2.6.28.2/arch/arm/configs/myconfig

>make -C kernel/work_2.6.28.2 ARCH=arm CROSS_COMPILE=arm-linux- myconfig
zImage
```

4 Boot process

Booting Linux on Embedded Artists' LPC313x boards is a two step process.

1 Loading Apex: LPC313x has on chip bootrom which loads properly formatted images from multiple sources, including SPI Flash, NOR Flash, UART, USB, SD Card, and NAND Flash. The boot interface is selected based on the states of GPIO0, GPIO1 and

GPIO2 pins at reset. See LPC313x user manual for more details. The apex.bin image built as part of LTIB and ELDK frameworks has proper image header for bootROM to load the image. Section 4.1 gives more details on how to prepare the interface to load Apex.

2 Loading kernel and ramdisk: Once Apex boots, Apex is used to load kernel and boot the kernel. Apex boot-loader supports loading kernel from UART, NAND, SD/MMC card, Ethernet and USB interfaces. See section 4.2 for more details on how to load kernel & ramdisk from various interface.

4.1 Loading Apex

4.1.1 UART boot

Set the EA LPC3131 board in UART boot mode by setting jumpers – BOOT0(H), BOOT1(H) and BOOT2(L). Configure the serial port on EA3131 board as described in section 4.1.1.1.

4.1.1.1 Configuring the serial port in EA3131 board

Two different modes can be used for the serial port:

a) UART (DB9 connector):

In this case, J27/J29 (see 2 in Fig 1) must be set to the Upper position (RS232 position), while J28/J30/J31/J32 (see 3 in Fig 1) must be set to the Left position (RS232 position). A serial cable must be connected between the EA3131 board (DB9 connector) and the PC (serial connector).

b) USB-to-UART bridge (mini-USB connector):

In this case, J27/J29 (see 2 in Fig 1) must be set to the Lower position (USB position). A USB cable must be connected between the EA3131 board (mini-USB connector) and the PC (USB connector). A Virtual COM driver must be installed in this case, so please refer to the EA3131 board's User Manual for specifics instructions.

Note: independently of the above configuration chosen, the EA3131 board can be powered via the mini-USB or the External Power Supply connectors (or even from both connectors at the same time).

4.1.1.2 Start a PC Terminal application program

Configure a terminal application (which should be able to transfer files in binary mode, such as TeraTerm Pro) with 115200-8-n settings. If using USB-to-serial bridge port on EA board, the appropriate Virtual COM port has to be selected. By the time USB-to-serial enumerates, the bootROM of LPC313x would have transmitted the initial string. Hence reset the board using the "reset" button after opening the terminal application.

Note: the default installation of TeraTerm Pro allows only up to COM4 ports. To increase the number of COM ports accessible by TeraTerm Pro, change the following line in TERATERM.INI (C:\Program Files\TTERMPRO):

MaxComPort=4 to MaxComPort=10

4.1.1.3 Load the programmer code

Once the EA313x board is powered and connected to the PC, reset the board. The following message should appear in the terminal window:

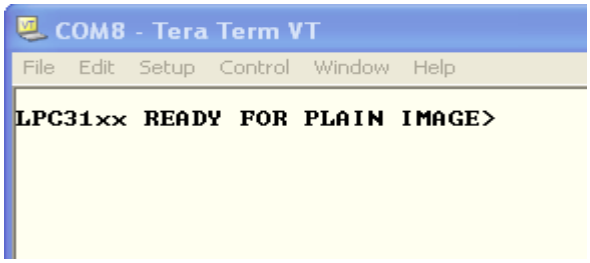
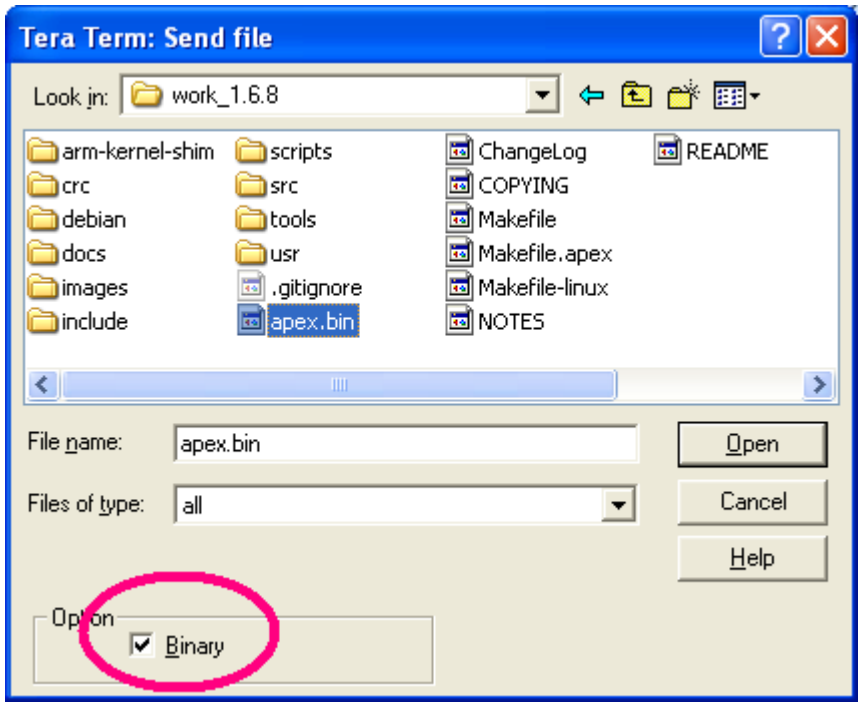


Fig 1. Initial prompt after resetting the board

Select *File -> Send* file option from the terminal application's menu and the next screen will appear:



Don't forget to check the Binary option, as the file to send is a binary image.

Fig 2. Send file selection: apex.bin

Choose the "apex.bin" file, check the Binary option, and press Open to start the file transfer. After download, the following message will appear:

```
LPC31xx READY FOR PLAIN IMAGE>
Download finished

* NAND flash 256 MiB total, 128 KiB erase, 2048 B page
(0x2c/0xaa/0x80/0x15)


APEX Boot Loader 1.6.8 -- Copyright (c) 2004-2008 Marc Singer
  compiled for Unspecified target on 2009.May.15-14:50:24

    APEX comes with ABSOLUTELY NO WARRANTY.  It is free software and
    you are welcome to redistribute it under certain circumstances.
    For details, refer to the file COPYING in the program source.

  apex => mem:0x11029000+0xfafc    (64252 bytes)
  env  => lnand:512k+256k          (empty)

    Use the command 'help help' to get started.

# wait 10
```

```
# copy $kernelsrc $bootaddr
# copy ext2://1/zImage 0x30008000
1113872 bytes transferred
# copy $ramdisksrc $ramdiskaddr
# copy ext2://1/ramdisk_image.gz 0x32000000
1659961 bytes transferred
# boot
ARCH_ID: 9998 (0x270e)
ATAG_HEADER
ATAG_MEM: start 0x30000000  size 0x04000000
ATAG_CMDLINE: (52 bytes) 'console=ttyS0,115200n8 root=/dev/ram0 rw
loglevel=7'
ATAG_INITRD2: start 0x32000000  size 0x00300000
ATAG_END
Booting kernel at 0x30008000...
Uncompressing
Linux.....................................................................
.......... done, booting the kernel
```

4.1.2 SD/MMC card

Set the EA LPC3131 board in SD/MMC boot mode by setting jumpers – BOOT0(L), BOOT1(H) and BOOT2(H). Load apex.bin file to SD/MMC card as described in section 4.1.2.1. Configure the serial port on EA3131 board as described in section 4.1.1.1. Configure the terminal window on PC as described in section 4.1.1.2. Insert SD/MMC card into the slot on EA LPC3131 board. Now power-up the board, the Apex boot screen should appear on the terminal window.

4.1.2.1 Formatting the card

This section gives the step-by-step instructions in creating LPC3130/31 bootable partition on SD/MMC cards using “fdisk” utility available on Linux PC.

- Invoke fdisk on the device node associated with SD card. Use ‘dmesg’ command to figure out “/dev/sdxx” device Linux used for the current USB card reader. The “/dev/sdxx” log entries appear at the very end of the dmesg output.

```
$ sudo fdisk /dev/sde
[sudo] password for xxx_user:
```

- Print the current partition table entries.

```
Command (m for help): p
Disk /dev/sde: 32 MB, 32112640 bytes
1 heads, 62 sectors/track, 1011 cylinders
Units = cylinders of 62 * 512 = 31744 bytes
Disk identifier: 0xde283a86
Device Boot Start End Blocks Id System
/dev/sde1 2 899 27838 6 FAT16
/dev/sde2 900 1011 3472 df BootIt
Command (m for help):
```

- Note, always create "bootit" (partition type 0xDF) partition as second partition. So that when the card is plugged back into a Windows PC it doesn't format “bootit” partition. Windows will not complain as long as the first partition is either FAT or NTFS partition.
- You could use 'm' command under “fdisk” to get help on other “fdisk” commands.
- Delete all existing partitions on the card one at a time

```
Command (m for help): d
Partition number (1-4): 1
Command (m for help): d
Partition number (1-4): 2
```

- Now create new partitions. To specify the amount of space you need to specify start block and end block for each partition. This is usually the cylinders numbers. Since they vary from card to card it is little confusing what to specify. So we create the second partition first with +1M (1 MB size).

```
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (1-1011, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-1011, default 1011): +1M
Command (m for help): t
Selected partition 2
Hex code (type L to list codes): df
Changed system type of partition 2 to df (BootIt)
Command (m for help):
```

- Now create first partition of type FAT16 or FAT32. The card used in illustration is 32MB only so we will create FAT16 in this example.

```
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (34-1011, default 34):
Using default value 34
Last cylinder or +size or +sizeM or +sizeK (34-1011, default 1011):
Using default value 1011
Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): 6
Changed system type of partition 1 to 6 (FAT16)
Command (m for help): p
Disk /dev/sde: 32 MB, 32112640 bytes
1 heads, 62 sectors/track, 1011 cylinders
Units = cylinders of 62 * 512 = 31744 bytes
Disk identifier: 0xde283a86
Device Boot Start End Blocks Id System
/dev/sde1 34 1011 30318 6 FAT16
/dev/sde2 1 33 1022+ df BootIt
Partition table entries are not in disk order
```

```
Command (m for help):
```

- Now write the table and exit from fdisk

```
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: If you have created or modified any DOS 6.x
partitions, please see the fdisk manual page for additional
information.
Syncing disks.
$
```

- Now dump the boot image to /dev/sde2 partition using "dd" command as follows. If you are using latest LPC313x CDL, the bin files generated by make system can be written directly to the card. If not then you need to create the image in the format described in LPC313x User manual Section 6–4.2.

```
$ sudo dd if=./apex.bin of=/dev/sde2 bs=512
[sudo] password for xxxuser:
102+1 records in
102+1 records out
52528 bytes (53 kB) copied, 0.186911 s, 281 kB/s
$
```

- Now the card is ready for booting. Don't forget to "sync" the card before ejecting. Also don't forget to put LPC3130/31 in SD/MMC boot mode.
- The apex.bin file can be found at:
    a. LTIB framework: ./ltib/rootfs/boot
    b. ELDK framework: /home/xxx/projects/lpc313x/apex/work_1.6.8/apex.bin

4.1.3 On board SPI-NOR flash

Set the EA LPC3131 board in SPI boot mode by setting jumpers – BOOT0(L), BOOT1(L) and BOOT2(H). Program “apex.bin” into on-board SPI-NOR flash. Refer AN10811 Programming SPI flash on EA3131 boards (with software), V1 (May 1, 2009) application note available on http://www.standardics.nxp.com/support/documents/microcontrollers/?scope=LPC3131 website for SPI-flash programming instructions. Configure the serial port on EA3131 board as described in section 4.1.1.1. Configure the terminal window on PC as described in section 4.1.1.2. Now power-up the board, the Apex boot screen should appear on the terminal window.

4.1.4 On board NAND flash

LPC313x patch adds NAND based block device support Apex 1.6.8. LPC313x has NAND controller with HW ECC. A new command called "lpcnand" is added to Apex to format NAND flash device as defined by boot ROM. This command writes block0-page0 with params and

block0-page1 with bad block list. The bad block indicator programmed by the NAND vendor will be removed if an erase command is issued to the block. Hence the bad block list should be created before programming any other part of the device. Hence this command should only be executed only once on a board fresh from factory.

```
apex> lpcnand format
```

To program nand flash with apex.bin image,

- Copy the image an ext2 formatted SD card.
- Load apex on the board using UART boot or SD boot method.
- At apex prompt copy the image from SD card to SDRAM .

```
apex> copy ext2://1/apex.bin 0x30008000
62664 bytes transferred
apex>
```

- Now copy the image from SDRAM to NAND flash. Always copy the image to SDRAM as intermediate step since the nand driver in apex always operate on pages boundary. Copying images directly from SD card to NAND is not supported.

```
apex>copy 0x30008000+62664 lnand:128k
```

- Note, the number 62664 is size of the apex.bin image. This number will vary depending on your build image size. But always program apex.bin at offset 128k onwards.

4.1.5 USB DFU

TBD.

4.2 Loading kernel & ramdisk using Apex boot-loader from

4.2.1 SD/MMC card (EXT2 formatted)

Apex 1.6.8 with LPC313x patch supports loading images from ext2 formatted SD/MMC cards.

4.2.1.1 Preparing SD/MMC card

To format the SD/MMC card do the following:

- Insert card into USB reader/SD slot connected to your Linux PC.
- Open shell terminal and find out the device node SD card is associated. Use ‘dmesg’ command to figure out “/dev/sdxx” device Linux used for the current USB card reader. The “/dev/sdxx” log entries appear at the very end of the dmesg output.

- Now issue format command on the partition to which you would like to copy the zImage and ramdisk.

```
$ sudo mke2fs /dev/sde1
[sudo] password for xxxx:
mke2fs 1.40.8 (13-Mar-2008)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
7584 inodes, 30284 blocks
1514 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=31195136
4 block groups
8192 blocks per group, 8192 fragments per group
1896 inodes per group
Superblock backups stored on blocks:
      8193, 24577

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 31 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
$ sync
$
```

- Mount the formatted partition and copy the zImage and ramdisk files.

```
$ sudo mount /dev/sde1 /media/disk/
$ sudo cp kernel/work_2.6.28.2/arch/arm/boot/zImage /media/disk/
$ sync
$ sudo cp eldk/arm/images/ramdisk_image.gz /media/disk/
$ sync
$
```

- Note, the above commands show the path of the files when ELDK framework is used to build kernel. For users using LTIB framework the image are available at:
  - Apex bootloader image ---> ./ltib/rootfs/boot/apex.bin
  - Linux kernel image ---> ./ltib/rootfs/zImage
  - Compressed ramdisk (EXT2) ---> ./ltib/rootfs.ext2.gz

4.2.1.2 Configuring Apex to boot from SD/MMC

Once Apex boots to load the kernel and ramdisk issue the following commands at Apex prompt:

```
apex> copy ext2://1/zImage $bootaddr
apex> copy ext2://1/ramdisk_image.gz $ramdiskaddr
```

Note, when LTIB framework is used the ramdisk filename rootfs.ext2.gz should be used in the above command. Also to list all the files present on the SD/MMC card use the following command at apex prompt.

```
apex> info ext2://1/
./
../
lost+found/
apex>
```

As part of Apex configuration the default startup command can be configured to autoboot from SD card. The ea313x_v1_config configuration file present as part of the patch already sets the startup command to boot from SD card. To modify the startup command, change the value of CONFIG_ENV_DEFAULT_STARTUP in ea313x_v1_config file. The following config variables affect the SD boot.

For the LTIB environment, this can be done by checking the “Configure Apex” option from the LTIB main configuration menu (./ltib –config) and then rebuilding. If Apex doesn’t rebuild, use the following command to force Apex to rebuild the next time Apex is run.

```
touch ./ltib/dist/lfs-5.1/apex/apex-common.tmpl
```

```
CONFIG_ENV_REGION_KERNEL="ext2://1/zImage"
CONFIG_ENV_REGION_RAMDISK="ext2://1/ramdisk_image.gz"
CONFIG_ENV_DEFAULT_STARTUP="wait 10;copy $kernelsrc $bootaddr;copy
$ramdisksrc $ramdiskaddr; boot"
```

The apex environment variables can be modified using ‘setenv’ command. The current EA313x implementation of Apex saves the environment on onboard NAND flash. To save the environment use “saveenv” command.

4.2.2 On board NAND flash

The LPC313x patch adds NAND based block device support Apex 1.6.8. Please format the NAND device as per LPC313x boot specification before programming the device. Check section 4.1.4 for more details.

To program kernel image into nand flash do the following. Always copy kernel image at 768k offset so that enough block are left for apex and its environment. See apex\work_1.6.8\src\mach-lpc313x\drv-nandc.c file for more details.

```
apex> copy ext2://1/zImage 0x30008000
1113872 bytes transferred
apex>copy 0x30008000+1113872 lnand:768k
```

You can also copy kernel using UART download mechanism built in Apex. To do that issue "xreceive" command as shown below. On terminal application window initiate the file transfer using XMODEM protocol with checksum option.

```
apex> xreceive 0x30008000
C77952 bytes received
apex> copy 0x30008000+77952 lnand:128k
77952 bytes transferred
apex>
```

To load kernel image from NAND do the following.

```
apex> copy lnand:768k+2m 0x30008000
2097152 bytes transferred
apex>
```

Similarly program the ramdisk into "lpcnand:4m" region (ie. at 4 Mbytes offset in NAND device).

JFFS2 based root file system is not tested at the release of this package. But all the need drivers and tools are made available in this release.

4.2.3 Ethernet using TFTP boot (available for EA LPC31xx Version 2.0 base boards only)

Embedded Artist's LPC313x version 2.0 IO boards have DM9000 Ethernet controller onboard. The board doesn't have MAC-EEPROM populated by default, hence the MAC address has to be stored in on board NAND flash or SPI-flash. Current implementation of Apex provides "eth mac" command to set the mac address. Users could use apex environment variables to save different MAC address for different boards. The following apex screen dump provides command sequence to boot using TFTP. The following commands assume the IP address for server and the target are setting using "rarp" protocol.

```
apex>  eth mac 00:08:ee:00:80:43
apex>  ipconfig rarp
hostip 192.168.1.77
serverip 192.168.1.30
gatewayip 192.168.1.30
apex>  copy tftp://$serverip/zImage $bootaddr
# copy tftp://192.168.1.30/zImage 0x30008000
1114524 bytes transferred
apex>   copy tftp://$serverip/ramdisk_image.gz $ramdiskaddr
# copy tftp://192.168.1.30/ramdisk_image.gz 0x32000000
1659961 bytes transferred
# boot
```

To configure rarp setting on your development Linux pc do the following:

- Add the following entry to /etc/ethers file. If file is not present create it.

```
00:08:ee:00:80:43 192.168.1.77
```

- Then execute "$sudo arp -f". This will update the arp entries from the "/etc/ethers" file.

Configuring TFTP server on Linux PC is outside the scope of this document.

4.2.4 UART using Xmodem protocol

Apex 1.6.8 when enabled supports XMODEM protocol for serial file transfers. The file transfers are achieved using “xreceive” command.

- Download kernel image by issue following command. On PC terminal application select file transfer using XMODEM protocol and select the zImage file.

```
apex> xreceive $bootaddr
# xreceive 0x30008000
C1114624 bytes received
apex>
```

- Now download the ramdisk image using the method as above.

```
apex> xreceive $ramdiskaddr
# xreceive 0x32000000
C1114624 bytes received
apex>
```

- After both files are downloaded enter ‘boot’ command to boot linux.

4.2.5 USB using mass storage class

Current version (1.6.8) of Apex doesn’t support USB boot mechanism. Since USB interface is the only high-speed communication interface on Embedded Artists’ version 1.0 boards, NXP has created a USB boot mechanism with a 3rd party stack as an external linked library. Due to licensing issue the USB code can’t be distributed under GPL. Hence customers could download the pre-built binary module and do the following to integrate this feature for development purpose only.

- Download usbmsc.d module from NXP microcontroller website (also available in lpc313x_linux_r1a.tar.bz2 tarball).
- Copy usb/usbmsc.d file to apex working directory at work_1.6.8\src\mach-lpc313x\usb
- Edit the apex configuration to enable “USB boot” using “make menuconfig”. Check ELDK and LTIB configuration sections for more detail on how to edit apex configuration in individual framework environments.
- Build apex for EA313x version 1 board.

- For LTIB framework users the source code is located at ./ltib/rpm/BUILD/apex
  - LTIB deletes the sources after build hence make sure the “Leave the Apex sources after building” option is checked in the LTIB main menu (./ltib –config).

To use USB boot mechanisms do the following.

- A new command "usb" has been added to Apex command list. When you use this command the device enumerates as mass storage drive to the PC/Linux host it is connected to. On PC you could copy only 1 file to the drive as soon as copy finishes within 2 secs the apex will disconnect the drive from PC. The command takes single parameter to specify the type of file (kernel image or ramdisk) you plan to download.

```
            apex> usb k    <---- Download zImage. The board will
enumerate to PC with volume label as "zImage Disk"
            apex> usb r    <---- Download ramdisk_image.gz. The board
will enumerate to PC with volume label as "rootfs Disk"
```

If no parameter is specified the command assume kernel download.

- After Apex boots, issue commands in following sequence to boot Linux.

```
apex> usb
Downloading Kernel to address 0x30008000
done
apex> usb r
Downloading Ramdisk to address 0x32000000
done
apex> boot
```

5 Porting Apex to LPC313x custom board

T.B.D

6 Porting Linux 2.6.x to LPC313x custom board

T.B.D

7 Build differences between LTIB and ELDK

LTIB and ELDK take different approaches to providing a complete Linux system. The ELDK approach provides pre-built binary packages from the root filesystem image. LTIB builds all necessary packages and the root filesystem from the configuration options selected by the user.

For the current version of the ELDK and LTIB releases, the following differences apply.

- ELDK and LTIB use similar, but different toolchains
- The generated LTIB ramdisk image is bigger than the pre-packaged ELDK ramdisk image
- The Apex default configuration has a change to the Linux kernel command line to allow for increased ramdisk size
    - If packages are added to the system via the LTIB package menu, this ramdisk size may need to be increased. LTIB will give a recommended value to use for the new ramdisk size.
    - The ramdisk size can be edited in the Apex menu using the “Configure Apex” option and then the environment menu in Apex.