



Using X-Loader with the OMAP35x Linux BSP

Application Note 460

Logic PD // Products
Published: December 2010
Last revised: September 2013

Abstract

This application note explains how to replace the standard LogicLoader bootloader with X-Loader on the Zoom™ OMAP35x Development Kit or Zoom OMAP35x Torpedo Development Kit.

This document contains valuable proprietary and confidential information and the attached file contains source code, ideas, and techniques that are owned by Logic PD, Inc. (collectively "Logic PD's Proprietary Information"). Logic PD's Proprietary Information may not be used by or disclosed to any third party except under written license from Logic PD, Inc.

Logic PD, Inc. makes no representation or warranties of any nature or kind regarding Logic PD's Proprietary Information or any products offered by Logic PD, Inc. Logic PD's Proprietary Information is disclosed herein pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase or lease equipment. The only warranties made by Logic PD, Inc., if any, with respect to any products described in this document are set forth in such license or agreement. Logic PD, Inc. shall have no liability of any kind, express or implied, arising out of the use of the Information in this document, including direct, indirect, special or consequential damages.

Logic PD, Inc. may have patents, patent applications, trademarks, copyrights, trade secrets, or other intellectual property rights pertaining to Logic PD's Proprietary Information and products described in this document (collectively "Logic PD's Intellectual Property"). Except as expressly provided in any written license or agreement from Logic PD, Inc., this document and the information contained therein does not create any license to Logic PD's Intellectual Property.

The Information contained herein is subject to change without notice. Revisions may be issued regarding changes and/or additions.

© Copyright 2013, Logic PD, Inc. All Rights Reserved.

Revision History

REV	EDITOR	DESCRIPTION	APPROVAL	DATE
A	SMC, JCA	-Initial Release	RGL	12/09/10
B	HJS	-Throughout: Updated template; updated links for new support site; -Added Section 1.1	SO	09/19/13

Table of Contents

1	Introduction	1
1.1	Nomenclature.....	1
2	Prerequisites	1
2.1	Connect Kit to Host PC	1
2.2	Download OMAP35x Linux BSP	1
3	Upload X-Loader to OMAP35x SOM	2
3.1	Copy Boot Files to SD Card	2
3.2	Identify PoP Memory Chip ECC Method	2
3.3	Load Boot Files to NAND	3
3.4	Boot the Linux Kernel using X-Loader	5
4	Summary	5

1 Introduction

This document explains how to replace LogicLoader with X-Loader on the Zoom™ OMAP35x Development Kit or Zoom OMAP35x Torpedo Development Kit. Logic PD provides a custom version of X-Loader that will work with the Logic PD OMAP35x Linux BSP; this X-Loader is provided as part of the OMAP35x Linux BSP download.

1.1 Nomenclature

- This document covers the OMAP35x SOM-LV and OMAP35x Torpedo SOM. Use of “OMAP35x SOM” suggests text that applies to both platforms; information specific to one platform will call out the precise name.
- Use of “Zoom OMAP35x Development Kit” suggests text that applies to both the OMAP35x SOM-LV Development Kit and OMAP35x Torpedo Development Kit; information specific to one development kit will call out the precise name.

2 Prerequisites

The following items are necessary to complete the procedures in this document.

- Zoom OMAP35x Development Kit [registered on Logic PD's support site](#)¹
- Host PC
- Available serial port or USB-to-serial adapter
- Terminal emulation program
- Active Internet connection
- SD card
- SD card reader

Before continuing, it is also necessary to complete Section 2.1 and Section 2.2 below.

2.1 Connect Kit to Host PC

1. Use the gray serial cable included in the OMAP35x Development Kit to connect the serial port on the kit to the appropriate port on your host PC.
2. Launch your terminal emulation program and set the following port settings:
 - a. Baud rate: **115200**
 - b. Data: **8 bit**
 - c. Parity: **None**
 - d. Stop: **1 bit**
 - e. Flow control: **None**

2.2 Download OMAP35x Linux BSP

To get the X-Loader version for your OMAP35x SOM, it is necessary to download the OMAP35x Linux BSP from Logic PD's website.

1. Log into Logic PD's [support site](#)² using your username and password.
2. Navigate to the product page for your product and locate the *Linux* section heading.
3. Click on the *OMAP35x SOM Linux Demo Image* to download the BSP.
4. Save the ZIP file to your host PC and extract the file.

¹ <http://support.logicpd.com/TechnicalSupport/RegisterProduct.aspx>

² <http://support.logicpd.com/Home.aspx>

3 Upload X-Loader to OMAP35x SOM

This section will explain the steps to replace the LogicLoader bootloader on your OMAP35x SOM with the Logic PD custom version of X-Loader.

3.1 Copy Boot Files to SD Card

1. Copy the following files from the Linux BSP download to a blank SD card.
 - *x-load.bin.ift* (X-Loader first stage bootloader binary image)
 - *u-boot.bin* (U-Boot bootloader binary image)
 - *u-boot* (ELF file)
 - *uImage* (Linux kernel)
 - *rootfs.yaffs2* (YAFFS root filesystem)

NOTE: These files may have a 10xxxxx number preceding the file name (e.g., *1016980_x-load.bin.ift*). Once copied to the SD card, feel free to remove the *10xxxxx_* characters from the file names so that they will be easier to use in the instructions that follow.

2. Insert the SD card into the SD card slot on the OMAP35x Development Kit baseboard.

3.2 Identify PoP Memory Chip ECC Method

Before beginning, it is important to know what Package-on-Package (PoP) memory chip is on the SOM; this is necessary because the error-correcting code (ECC) method is different depending upon the chip manufacturer and die revision. Knowing what PoP memory chip is on your SOM will tell you whether a 4-bit ECC or 1-bit ECC method is used.

The following steps provide the best method to identify the ECC method.

1. Launch a terminal emulation program.
2. Power on your development kit and wait until you see the LogicLoader `losh>` prompt.
3. Use the `load elf` command to load the *u-boot* file from the SD card; then use the `exec` command to launch U-Boot.

Below is an example of how you can load the *u-boot* file.

```
losh> mount fatfs /dev/sdmmc0a /sd
losh> cd sd
losh> ls
      :                x-load.bin.ift      20684
      :                rootfs.yaffs2 31699008
      :                U-BOOT      852644
      :                U-BOOT.BIN   217228
      :                uImage    2500648
losh> load elf /sd/U-BOOT
loading from /sd/U-BOOT:
.....
.....
ELF section 0: download address: 0x80808000 load address: 0x80e80000
loaded 217228 @ 0x80e80000 Ram
file loaded
losh> exec
```

NOTE: If you see the *Please enter your LCD display number* message, enter the number of the display connected to your development kit (this will most likely be 15 which corresponds to the 4.3" WQVGA LCD included with your development kit).

4. When you see the *Hit any key to stop auto-boot* message and a countdown, press any key to stop the auto-boot process.
5. Scroll up in the terminal window and locate the message *NAND: XX ECC selected* (where *XX* is either *HW* or *Internal to NAND*), see highlighted line below for an example.

```
U-Boot 2009.08-rc2/bsp-2.1 (Oct 07 2010 - 11:03:29)

OMAP3530-GP ES2.1, CPU-OPP2 L3-165MHz
OMAP3 Logic + LPDDR/NAND
DRAM: 128 MB
NAND: 256 MB
NAND: HW ECC selected
*** Warning - bad CRC or NAND, using default environment
```

6. If *HW* appears, then 1-bit ECC is being used; if *Internal to NAND* appears, then 4-bit ECC is being used (the example above has memory that uses 1-bit ECC). This information will be necessary for steps in Section 3.3.

3.3 Load Boot Files to NAND

In the instructions that follow, all commands following the terminal prompt => should be typed exactly as shown followed by then pressing the **Enter** key. Any deviation from this process will be called out in the **IMPORTANT NOTE** for that step.

1. Initialize the SD card.

```
=> mmc init
mmc1 is available
```

2. Verify the contents of the SD card.

```
=> fatls mmc 0
xxxxxxx rootfs.yaffs2
xxxxxxx u-boot.bin
xxxxxxx uImage
xxxxxx x-load.bin.ift

4 file(s), 0 dir(s)
```

3. In order to load X-Loader to the OMAP35x SOM, the entire NAND chip must be erased.

```
=> nand erase

NAND erase: device 0 while chip
Erasing at 0xxxxxxx -- 100% complete.
OK
```

4. Load and burn X-Loader into NAND starting at offset 0x00000, as well as three duplicate locations in case of bad blocks, by entering the commands below.

```
=> fatload mmc 0 $loadaddr x-load.bin.ift
reading x-load.bin.ift

xxxxx bytes read
=> nandecch hw
=> nand write $loadaddr 0x00000 0x20000
=> nand write $loadaddr 0x20000 0x20000
=> nand write $loadaddr 0x40000 0x20000
=> nand write $loadaddr 0x60000 0x20000
```

5. Load and burn U-Boot into NAND starting at offset 0x80000.

IMPORTANT NOTE: If your memory chip uses 1-bit ECC, replace `<ECC1>` in the command below with `sw`; if your memory chip uses 4-bit ECC, replace `<ECC1>` with `chip`. (See Section 3.2 to determine which memory chip you have.)

```
=> fatload mmc 0 $loadaddr u-boot.bin
reading u-boot.bin

xxxxxxx bytes read
=> nandecch <ECC1>
=> nand write $loadaddr 0x80000 0x40000
```

6. Load and burn the Linux kernel into NAND starting at 0x100000 (assuming the kernel is 0x280000 bytes or smaller).

IMPORTANT NOTE: If your memory chip uses 1-bit ECC, replace `<ECC2>` in the command below with `hw`; if your memory chip uses 4-bit ECC, replace `<ECC2>` with `chip`.

```
=> fatload mmc 0 $loadaddr uImage
=> nandecch <ECC2>
=> setenv kernelsize 0x280000
=> nand write $loadaddr 0x100000 $kernelsize
```

7. Load and burn the YAFFS2 root filesystem into NAND starting at offset 0xec0000.

```
=> fatload mmc 0 $loadaddr rootfs.yaffs2
reading rootfs.yaffs2

xxxxxxxxx bytes read
=> nand write.yaffs $loadaddr 0xec0000 $filesize

NAND write: device 0 offset 0xec0000, size 0xxxxxxxxx

Writing data at 0xxxxxxxxx -- 100% complete.
xxxxxxxxx bytes written: OK
```

3.4 Boot the Linux Kernel using X-Loader

1. Reboot the development kit by pressing the **POWER ON/OFF** button on the baseboard.
Note that since we erased the entire NAND flash, all content was lost on this memory. This includes the U-Boot environment, so U-Boot will use the default environment.
2. You will see the *Please enter your LCD display number* message. Enter the number of the display connected to your development kit (this will most likely be **15** which corresponds to the 4.3" WQVGA LCD included with your development kit).
3. Interrupt the auto-boot sequence while it is counting down by pressing any key.
4. Set up the NAND boot method and variables; then save these variables to the U-Boot environment.

```
=> setenv kernelsize 0x280000
=> setenv nandboot 'nand read $loadaddr 0x100000 $kernelsize; run
mtdboot'
=> setenv bootcmd run nandboot
=> saveenv
```

5. Reboot the development kit by pressing the **POWER ON/OFF** button on the baseboard. Allow the auto-boot to continue and your development kit should boot into the Linux kernel. A portion of your terminal output will look similar to what you see below.

```
Texas Instruments X-Loader 1.4.2 (Oct 19 2010 - 10:40:17)
DRAM: H8KCS0SJ0AER_MT29C2G24MAKLAJG6 128MB
timed out in wait_for_bb: I2C_STAT=1000
Loading u-boot.bin from nand (offset 00080000 to 000bffff)
NAND: reading data with HW ECC method
.
.
.
Starting the dropbear ssh server:
Setting up touch handler and default /etc/pointercal

Welcome to the LTIB Embedded Linux Environment

!!!!!! WARNING !!!!!!!

The default password for the root account is: root
please change this password using the 'passwd' command
and then edit this message (/etc/issue) to remove this message

OMAP-35x login:
```

6. X-Loader and the Linux kernel have successfully been loaded onto your OMAP35x SOM. You can now use X-Loader to boot your OMAP35x Development Kit.

4 Summary

This application note provided the steps necessary to replace the LogicLoader bootloader on the OMAP35x SOM with X-Loader and the Linux kernel provided by Logic PD.