



U-Boot for Toradex Colibri module with NVIDIA Tegra 2 Manual

Piotr Gięda, Piotr Skrzypek

April 27, 2011

CONTENTS

1	Introduction	1
1.1	Contents of the release package	1
2	Build procedure	2
2.1	Preparing the working environment	2
2.2	Patching the U-Boot source	3
2.3	Compiling U-Boot	3
3	Running U-Boot	5
3.1	Running from RAM	5
3.2	Flashing U-Boot to NAND	5
4	Usage and tweaking	7
5	Changelog	8

INTRODUCTION

This document accompanies the release of a U-Boot port to the Toradex Colibri NVIDIA Tegra CPU module prepared by Ant Micro (<http://antmicro.com>). It explains how to patch, build, run and use the U-Boot 2011.03-rc2 on the Toradex Colibri NVIDIA Tegra2 Module.

In the section entitled *Build procedure* you will find the steps necessary to start your work and build the necessary files. If you want to use the precompiled image, you can skip this section entirely.

Running U-Boot shows how to run U-Boot from NAND and RAM using NVIDIA's *nvflash* utility.

Usage and tweaking gives some info on how to do minor tweaks to the release and points to a user manual for the bootloader.

Changelog contains a brief list of features added to the generic U-Boot 2011.03-rc2 sourcecode.

1.1 Contents of the release package

The release consists of the following files:

- *colibri_patches.tar.gz*, file containing the Ant Micro patches for u-boot-2011.03-rc2 sourcecode
- *apply_nv_patches.sh*, script for applying the NVIDIA patches
- *apply_colibri_patches.sh*, script for applying the Ant Micro patches
- *do_flash.sh*, script for flashing u-boot.bin to NAND Flash using *nvflash*
- *do_memory.sh*, script for running u-boot.bin from RAM using *nvflash*
- *nvflash.cfg*, a config file for the *nvflash* utility (for *do_flash.sh*)
- *nand.bct*, a bct file for the *nvflash* utility (for *do_flash.sh*)
- *gpl.txt*, a copy of the GNU General Public License (GPL).

Additionally, the following files will be available from [Ant Micro's weblog](#):

- *U-Boot_Colibri_Manual.pdf*, this manual
- *u-boot-2011.03-rc2-toradex-colibri-tegra.bin*, a precompiled image that can be used to skip the source compilation process

BUILD PROCEDURE

This chapter describes how to patch and compile U-Boot for the Toradex Colibri Tegra module. The U-Boot binary image can afterwards be loaded to RAM memory or to NAND flash, using the NVIDIA *nvflash* utility. Here we will focus only on how to produce the .bin file. Refer to *Running U-Boot* for more details concerning the flashing/bootup procedure.

We will proceed as follows:

- prepare the working environment and necessary files
- patch the sources
- configure and build the U-Boot image

2.1 Preparing the working environment

Any Linux installation is sufficient for creating the boot image. This manual was tested on *Gentoo 11* distribution and *Codesourcery arm-2010.09-50* GCC cross-compiler available from the [Codesourcery website](#).

2.1.1 Unpacking the release

The file *u-boot-2011.03-rc2-toradex-colibri-tegra.tar.gz* includes all the necessary files to patch U-Boot.

Unpacking the package:

```
tar xzvf u-boot-2011.03-rc2-toradex-colibri-tegra.tar.gz
```

2.1.2 Getting U-Boot

The unpatched sources can be downloaded from the [denx.de](#) FTP.

2.2. Patching the U-Boot source

Getting the source:

```
wget ftp://ftp.denx.de/pub/u-boot/u-boot-2011.03-rc2.tar.bz2
```

2.2 Patching the U-Boot source

2.2.1 Running the NVIDIA patches

Additional patches provided by NVIDIA are needed for the Colibri patch to work.

Applying the NVIDIA patches:

```
cd /path/to/u-boot-colibri-release/  
sh apply_nv_patches.sh /path/to/u-boot-2011.03-rc2/source
```

2.2.2 Running Ant Micro patches for the Toradex Colibri Tegra 2 Module

After applying the NVIDIA patches, the main patches provided by Ant Micro can be applied.

Applying the main patch:

```
cd /path/to/u-boot-colibri-release/  
sh apply_colibri_patches.sh /path/to/u-boot-2011.03-rc2/source
```

The source code should now be ready for cross-compilation.

2.3 Compiling U-Boot

The compilation process should now be fairly straightforward:

Compiling U-Boot from source:

```
cd /path/to/u-boot-colibri-release/source  
PATH="$PATH:/path/to/crosscompiler" CROSS_COMPILE="arm-none-linux-gnueabi" \  
TARGET_ARCH=arm make colibri_tegra2_config  
PATH="$PATH:/path/to/crosscompiler" CROSS_COMPILE="arm-none-linux-gnueabi" \  
TARGET_ARCH=arm make all
```

2.3. Compiling U-Boot

The result should be a u-boot.bin file located in the main U-Boot sources directory. This binary can be now uploaded to the Tegra module and run, which is covered in the next section.

RUNNING U-BOOT

U-Boot for Tegra can be run either from RAM or from NAND flash. Refer to *Running from RAM* and *Flashing U-Boot to NAND* respectively.

3.1 Running from RAM

To run U-Boot from RAM, without flashing it to NAND flash, NVIDIA's utility *nvflash* (tool available from NVIDIA) can be used.

To do so one must put Toradex Colibri Module to *recovery mode* (refer to: http://wiki.toradex.com/index.php/Colibri_Recovery_Mode).

evoking do_memory.sh script:

```
sh do_memory.sh /path/to/u-boot.bin
```

3.2 Flashing U-Boot to NAND

NVIDIA's utility *nvflash* can also be used to flash U-Boot to the device's NAND. The prerequisite is the original Colibri Tegra 2 bootloader (eboot), available from Toradex's website (*EBOOT.NB0* - put it in the same directory as the *nvflash* tool), as the *nvflash* utility is not able to modify the flash on its own, without the help of a cooperating bootloader.

flashing u-boot.bin with the do_flash.sh script:

```
sh do_flash.sh /path/to/u-boot.bin
```

The binary file to be booted cannot be smaller than 128 kB. Therefore if the resulting u-boot.bin image is smaller than 128 kB, the following command can be used:

3.2. Flashing U-Boot to NAND

Padding the u-boot.bin image:

```
dd if=u-boot.bin.original of=u-boot.bin bs=1k count=128 conv=notrunc
```

and then the result can be flashed to NAND.

After flashing to NAND, the U-Boot should run automatically every time the device is powered up.

USAGE AND TWEAKING

The configuration header file (`colibri_tegra2.h`) is located in the *include/configs/* directory. It can be tweaked to add additional commands, customize environmental variables or switch the USB used in U-Boot (internal ASIX Ethernet or external connector).

A manual describing the usage of U-Boot (The DENX U-Boot and Linux Guide - DULG) can be found at <http://www.denx.de/wiki/DULG/WebHome>.

CHANGELOG

The core content of this release resides in the *colibri_patches.tar.gz* archive, which contains the following features:

- Separate configuration layout files for Toradex/Colibri
- Correct PLL, Clock and pinmux configuration
- cores initialization
- UART_A initialization, serial console
- Initialization for USB ULPI (SMSC USB3340)
- Asix AX88772B USB 2.0 Ethernet support

The bootloader was tested on a Toradex Colibri Tegra module on both the Iris and Orchid Carrier Boards. However, no warranty is given - use this package at your own risk. This software is released under *GNU General Public License (GPL)*, a copy of which is included in the software package.