

# at91sam9263ek-nor 1.00

---

Booting the at91sam9263ek from NOR flash (gnu)Linux  
October 2007

Alessandro Rubini (rubini@gnuudd.com)

---

## at91sam9263ek-nor

This document explains how to boot an AT91SAM9263 Evaluation kit from NOR flash memory. You need to mount a NOR chip on the board, though, as Atmed doesn't ship one.

I used a 8MB bottom-boot-block device, as suggested on the EK schematics. Some code must be adapted if you run a different chip.

### 1 The problem

The AT91SAM9263 CPU should be able to boot from NAND or SD, but as documented in the errata it can't boot from either device. So we have the obsolete DataFlash or the NOR flash, and my client chose to mount a NOR device in its production devices.

Given a bottom-boot-block device (8 sectors are 8kB wide, the rest is 64kB) I chose to allocate it in this way (NCS0 starts at 0x1000.0000):

```
0x1000.0000-0x1000.2000  at91boot
0x1000.2000-0x1000.4000  environment
0x1000.4000-0x1001.0000  available for app. configuration
0x1001.0000-0x1005.0000  u-boot
```

I modified *at91boot* instead of placing the copying code in *u-boot* itself like is done on most CPU cores for three reasons:

- My NOR-aware u-boot binary is the same you boot from dataflash
- The code that enables the SDRAM is already in *at91boot*
- Copying all of *u-boot* from NOR in slow-clock-mode takes time.

So I had to change both *at91boot* and *u-boot*. The patches are available in <ftp://gnudd.com/pub/patches/at91sam9263ek-nor/> and within `at91sam9263ek-nor=1.00.tar.gz`.

I used ELDK-4.1 to compile everything, and I assume in this document that `CROSS_COMPILE` is properly set in you environment.

### 2 at91boot

At boot time, the CPU runs very slowly, so we must switch the *master clock* at a higher frequency before we load *u-boot*. This switch can't be performed from NOR flash, as the access timings to the chip are different with the two clocks.

What I used in this project is `at91bootstrap-2.3` as available on [www.at91.com](http://www.at91.com). To the original source you need to apply my patches:

```
wget http://www.at91.com/repFichier/Project-209/at91bootstrap-2.3.tar.bz2
NORPATCH1=at91bootstrap-2.3-norboot1.patch.gz
wget http://gnudd.com/pub/patches/at91sam9263ek-nor/$NORPATCH1
tar xjf at91bootstrap-2.3.tar.bz2
cd at91bootstrap-2.3
zcat ../NORPATCH1 | patch -p1
make at91sam9263ek_defconfig MEMORY=norflash
make
```

You'll get a file called `at91sam9263ek-norflashboot-2.3.bin` in the `results/` subdir. This must be written to the first sector of the flash. The last chapter explains how to write it if you have a bootable DataFlash card.

A precompiled binary can be downloaded from <http://gnudd.com/pub/patches/at91sam9263ek-nor/>.

## 2.1 Description of the at91boot patch

I made all my changes trying not to break the other configurations, but I didn't re-check them. The patch changes the following files:

```

Makefile   For consistency, I use norflash instead of flash as device name; also a spurious tarball
           target is removed.

crt0_gnu.S
           The code now copies itself to SRAM, from _stext to _edata.

main.c     The source now calls load_norflash(), if CFG_NORGLASH

board/at91sam9263ek/at91sam9263ek.c
           I activated the I-cache, to make the copy loop faster.

board/at91sam9263ek/norflash/Makefile
board/at91sam9263ek/norflash/at91sam9263ek.h
           Ne filese, copied from ../dataflash and modified.

driver/driver.mk
           Added norflash.c

driver/norflash.c
           New file, it programs NCS0 timings and copies u-boot to SDRAM.

include/norflash.h
           Ne file, with the prototype.

```

## 3 U-Boot

For this project I used `u-boot-1.1.5` modified by Atmel. I tried their 1.2.0 version as well, but it didn't see the NAND flash, so I stucked to the older one.

To build from source, you need to run the following commands (there is a reject in the Atmel patch, but you can ignore it).

```

wget ftp://ftp.denix.de//pub/u-boot/u-boot-1.1.5.tar.bz2
ATPATCH="u-boot-1.1.5_atmel_1.4.diff.bz2"
wget http://www.at91.com/repFichier/Project-218/$ATPATCH
NORPATCH2=u-boot-1.1.5-atmel-1.4-norboot2.patch.gz
wget http://gnuudd.com/pub/patches/at91sam9263ek-nor/$NORPATCH1
tar xjf u-boot-1.1.5.tar.bz2
cd u-boot-1.1.5
bzcat ../$ATPATCH | patch -p1
zcat ../NORPATCH2 | patch -p1
make at91sam9263ek_config
make

```

The result is `u-boot.bin`, that you should place at offset `0x10000` (64kB) in the NOR flash. If you boot from DataFlash in order to program the NOR flash, you need to ran a NOR-aware U-boot binary, but you can't overwrite the one that is currently running.

Therefore, you'll need a binary compiled to run at a different address. I chose `0x23e0.0000` instead of the default `0x23f0.0000`.

To accomplish the task, after you compiled `u-boot.bin`, do the following:

```

FILES="board/at91sam9263ek/config.mk include/configs/at91sam9263ek.h"
perl -p -i -e 's/0x23f0/0x23e0/' $FILES
make && mv u-boot.bin u-boot-23e0.bin

```

```
perl -p -i -e 's/0x23f0/0x23e0/' $FILES
```

Both binaries are also available as precopiled files in <http://gnuudd.com/pub/patches/at91sam9263ek-nor/>.

### 3.1 Description of the patch to u-boot

The patch file `u-boot-1.1.5-atmel-1.4-norboot2.patch.gz` acts on the following files:

`board/at91sam9263ek/Makefile`

Use of local `flash.c` is removed, as I use `drivers/cfi_flash.c` instead.

`board/at91sam9263ek/at91sam9263ek.c`

It calls `hw_flash_init` to set up NCS0 timings, and it doesn't reset the PHY (it's unnedded, and I have an error in the debugger it is reset).

`common/lcd.c`

The strings are fixes to RAM, NOR and NAND sizes are printed as probed (not hardwired), and author attribution is fixed.

`cpu/arm926ejs/at91sam926x/ether.c`

The net device is initialized each time, as per sane u-boot policy. More over, an infinite while loop is broken, so ctrl-C works.

`cpu/arm926ejs/at91sam926x/interrupts.c`

Timer management has been rewritten, it didn't work. (This is not perfect, I had no time, but at least it works good enough).

`drivers/cfi_flash.c`

Fixed a bug when `CFG_HZ` is less than 1000.

`include/configs/at91sam9263ek.h`

I use NOR

`lib_arm/board.c`

Turn a debug print into a normal prints, so the run address is always shown

`tools/Makefile`

Changed to use my logo, and extract it from a text file, so it can be added by a textual patch.

`tools/logos/gnudd.gz.uu`

My log, as *bmp*, compressed and uuencodes.

## 4 Setting up for NOR boot

In order to program the NOR flash, you can either use a JTAG debugger or boot from DataFlash, for example with the binary-only *u-boot* that Atmel is releasing in blatant copyright violation.

I assume here you have the three binary files in a TFTP server.

First of all, you need to set up the boot loader for network access. For example like this:

```
setenv serverip 192.168.16.12
setenv ipaddr 192.168.16.200
setenv ethaddr 0:1:2:3:4:5
```

I'll assume in all of the following commands that the three variables are properly set (both when issuing commands to the blatantly-violating-copyright-binary-only-u-boot and when issuing them to the nor-aware 1.1.5 we rebuilt).

Initially, you boot from DataFlash and run *u-boot* at address `0x23f0.0000`. You therefore need to run the new one at a different address:

```
tftpboot 23e00000 uboot-23e0.bin
go 23e00000
```

At this point, you have a NOR-aware boot loader running and you can write to NOR flash:

```
tftpboot 23800000 at91sam9263ek-norflashboot-2.3.bin
protect off 10000000 10001fff
erase 10000000 10001fff
cp.b 23800000 10000000 00002000
```

```
tftpboot 23800000 u-boot.bin
protect off 10010000 1004ffff
erase 10010000 1004ffff
cp.b 23800000 10010000 40000
```

At this point, you can reboot with the BMS jumper in place, and enjoy a completely-freed system booting from NOR. You can also `saveenv` if you wish, keeping in mind the address choices outlined at the beginning of this document.

Accessing the NOR chip from GNU/Linux is easy, as you just need to configure the *physmap* driver with the proper addresses for NCS0 memory.

## Table of Contents

<b>at91sam9263ek-nor</b> .....	<b>1</b>
<b>1 The problem</b> .....	<b>1</b>
<b>2 at91boot</b> .....	<b>1</b>
2.1 Description of the at91boot patch.....	2
<b>3 U-Boot</b> .....	<b>2</b>
3.1 Description of the patch to u-boot .....	3
<b>4 Setting up for NOR boot</b> .....	<b>3</b>