## AT91RM9200DK U-Boot Flash Programming Solutions

### Introduction

The AT91RM9200DK Development Board is provided with a default boot program in its external Flash memory. This boot program, called U-Boot, is used by the AT91RM9200DK to boot after reset. The U-Boot performs initialization of the AT91RM9200DK, provides user interface via the Serial Debug Port and allows application download through the Serial Debug Port or Ethernet.

This Application Note aims at providing solutions when the U-Boot program contained in the external Flash of AT91RM9200DK Development Board has been deleted. Two ways of reprogramming the Flash memory are described in this Application Note; the method selected depends on whether the user has a Slingshot<sup>™</sup> JTAG/ICE interface. If this interface is not being used, then the external Flash is reprogrammed using the internal BootROM (embedded on the AT91RM9200 version 58A07F).



AT91 ARM<sup>®</sup> Thumb<sup>®</sup> Microcontrollers

# **Application Note**

6041A-ATARM-01/04





## **U-Boot Software**

### Features

The main features of the U-Boot software embedded in the AT91RM9200DK Development Kit are:

- Standalone primary bootstrap
- Small footprint
- OS-independent
- Auto-boot and interactive modes
- Command line interface
- Non-volatile environment variables
- Flash programming capability
- DataFlash programming capability (only available in latest Open Source download)
- Download through serial interface (Kermit protocol)
- Download through Ethernet (tftp)
- Integrated bootp
- Scripting capability

### Organization

To optimize the use of resources, U-Boot is made up of three parts:

- A primary bootstrap
- A de-compression executable
- A gzip-compressed binary executable called the U-Boot Image

The primary bootstrap is concatenated to the de-compression executable into one single file called the boot image and must reside in the AT91RM9200DK Flash on NCS0 at address 0x1000\_0000.

The primary bootstrap is a simple assembly code routine that sets up the system and puts it into an operational state. More specifically, it starts the high-speed clocks of the CPU and programs the on-chip memory controller to define the memory layout. When these steps are completed, the primary bootstrap gives control to the de-compression executable.

The de-compression executable is an optimized version of the gunzip program. It decompresses the gzip-compressed image of U-Boot into RAM, and jumps to it. The location of the binary executable in RAM is defined at link time.

# AT91RM9200DK U-Boot Application Note

# 0x1000\_0000 Boot Image 24 Kbytes 0x1000\_6000 Free 32 Kbytes 0x1000\_E000 0x1001\_0000 U-Boot gzipped Image 64 Kbytes

### Figure 1. Flash Mapping





## Reprogramming the External Flash on the AT91RM9200DK Development Board

### For Users of Slingshot JTAG/ICE Interface and MULTI<sup>®</sup>2000

Hardware and Software

The procedure for reprogramming the Flash described here is applicable to users of the AT91RM9200DK Development Board who have the Green Hills<sup>®</sup> Slingshot JTAG/ICE interface and MULTI2000.

Requirements	• • •	Green Hills S Green Hills M A serial cable HyperTermin
Connecting the AT91RM9200DK to the Host PC	1.	Connect the Serial Debug PC. Open a rate, 8-bit dat
	2.	Connect the JTAG/ICE Int
Launching the MULTI2000	3.	Extract the AT91RM920

- UBootFlashProgramming.zip file provided with this Application Note
- Slingshot JTAG/ICE interface
- /IULTI2000 (V3.6.1 or higher) running on your PC
- e (null modem)
- al running on your PC application (1115200, 8, N, 1)
- AT91RM9200DK to the Host PC using the serial cable through the Port on the AT91RM9200DK and the serial COM port on the Host HyperTerminal session with the following configuration: 115200 bit ta length, No parity bit, 1 stop bit and No flux control.
  - Slingshot JTAG/ICE Interface to the AT91RM9200DK through the terface connector and to the PC through the USB Port.
- AT91RM9200 Getting Started Software Package file called 0-GettingStarted-GHS3 6-1 1.exe, available in the UBootFlashg.zip file provided with this Application Note.
  - 4. Launch the MULTI2000 Project Manager and open the project Getting-**Started.bld.** This is available in the software package, extracted to the following C:\UBootFlashProgramming\AT91RM9200-GettingStarteddirectory path GHS3\_6-1\_1\AT91RM9200-GettingStarted\compil.
  - 5. Select the project basic\_flash.bld by double-clicking on it. Build it and check if build was succesful.
  - Power-up and reset the AT91RM9200DK Development Board. 6

#### **Configuring the Connection to** This is one of the most important steps in the procedure. As the Flash has been comthe AT91RM9200DK pletely erased, a standard initialization that would normally configure the External Bus Interface, the Remap and the PLLs is not performed on the AT91RM9200DK. To carry out the initialization correctly, a Target Setup Script file (extension .dbs) must be used. The file, named mpserv\_dk9200\_PLL.dbs, is provided with this Application Note. You are strongly advised to use it.

To proceed:

7. Click on the Connect button to open the Connection Chooser window:

5	*	Z
ers Co	nnect	(F4)

In the Connection Chooser window, select the ARM<sup>®</sup>9 Green Hills Probe<sup>™</sup> Connection (mpserv):



9. Edit the selected Connection Method:

Connection Chooser		
Connect to a Target:		
ARM 9 Green Hills Probe Connection (repserv) [Green Hills Probe and Sing 💌 💰	3	
Custom	alacted (	Connection Method

10. In the Slingshot Connection Editor window, select the Target Setup Script file that initializes the AT91RM9200DK correctly. It is very important to use the file mpserv\_dk9200\_PLL.dbs provided with this Application Note:

Green Hills Probe and Slingshot (mpserv) Connection Editor
Name: ARM 9 Green Hills Probe Connection (mpserv)
Type: Green Hills Probe and Singshot Connection (mpserv)
Log Connection to file:
Connection Advanced Debug
C Ethernet / IP Connection
Target Name or IP Address:
USB Connection
Target Setup Script IVesource\mpserv_dk3200_PLL.dbs Choose
mpserv -setup C:\UBootFlashProgramming\AT91RM9200-GettingStarted-GHS3
Connect OK Cancel Revert Apply





- Launching the Debugger
- 11. Click on the Debug button to run the debugger:



12. In the debugger environment, click on the Step button to download the basic\_flash project:



13. In the debugger environment, switch to the command pane:

ownloading program text and data. Please Wait ownload complete. unning 'C:\UBootFlashFrogramming\&T91RM9200-GettingStarted-GHS3_6-1_1\&T91RM9200-GettingSta ULTI>	urted\
inning 'C:\UBootFlashProgramming\AT91RE9200-GettingStarted-GES3_6-1_1\AT91RE9200-GettingSta	urted\
	vrted\
JLTI>	

Loading the Loader Application	The next step is to download the <b>loader.bin</b> application. This application is used to download a binary file from a HyperTerminal session through the Serial Debug Port using the Xmodem protocol. The loader.bin file to download is available in the the software package extracted at the beginning of the procedure and available in the directory C:\UBootFlashProgramming\AT91RM9200-GettingStarted-GHS3_6-1_1\AT91RM9200-GettingStarted\Flash. See "Launching the MULTI2000" on page 4.
	<ol> <li>Download the loader.bin application in the internal RAM at address 0x200000 by writing the following command in the command pane window (prompt MULTI&gt;):</li> </ol>
	MULTI> memload raw C:\UBootFlashProgramming\AT91RM9200- GettingStarted-GHS3_6-1_1\AT91RM9200-GettingStarted\Flash\loader.bin 0x200000
	15. Set the Program Counter (pc) at the beginning of the downloaded loader.bin application and run from here:
	MULTI> target reg pc 0x200000
	MULTI> target run
Loading U-Boot	The loader application is now running on the AT91RM9200DK and is ready to download a binary application file through the Serial Debug Port using the Xmodem protocol on a HyperTerminal session.
	16. Download the u_boot.bin binary file using the Xmodem protocol. This file is available in the UBootFlashProgramming.zip file provided with this Application Note under the directory path C:\UBootFlashProgramming\binary.

# 6 AT91RM9200DK U-Boot Application Note

# AT91RM9200DK U-Boot Application Note

In the HyperTerminal session, send the file u\_boot.bin using the Xmodem protocol. At the end of the Xmodem transfer, a new prompt is displayed in the HyperTerminal window:

Uboot>

The U-Boot is now running on the AT91RM9200DK.

This Boot program can now be used to download files to the SDRAM through the Serial Debug Port using the Kermit protocol with a HyperTerminal session. It is then copied to the Flash.

Downloading U-Boot BinaryThe last step is to download the primary bootstrap, boot.bin, and the gzip-compressed<br/>U-Boot image, u-boot.gz. These two files are available in the UBootFlashProgram-<br/>ming.zip file provided with this Application Note in the directory<br/>C:\UBootFlashProgramming\binary.

17. The first file to load is the boot image "boot.bin".

Uboot> loadb 20000000
## Ready for binary (Kermit) download ...
## Start Addr =0x20000000

The boot image is now loaded in SDRAM. The following command copies the SDRAM (0x20000000) to Flash(0x10000000):

Uboot> protect off 1000000 10005fff Unprotect 3 sectors Uboot> erase 10000000 10005fff Erase Flash from 0x10000000 to 0x10005fff... Erasing sector 0 ... ok. Erasing sector 1 ... ok. Erasing sector 2 ... ok. done. Erased 3 sectors. Uboot> cp.b 20000000 10000000 5FFF Copy to flash... done. Uboot> protect on 10000000 10005FFF Protected 3 sectors Uboot>

18. Once the primary bootstrap has been loaded, the U-Boot gzipped image can be copied into Flash. The file to load is the uboot gzipped image "**u-boot.gz**":

Uboot> loadb 20000000
## Ready for binary (Kermit) download ...
## Start Addr =0x20000000
Uboot> protect off 10010000 1001ffff
Unprotect 1 sectors
Uboot> erase 10010000 1001ffff
Erase Flash from 0x10010000 to 0x1001ffff...





```
Erasing sector 8 ... ok.
done.
Erased 1 sectors.
Uboot> cp.b 20000000 10010000 FFFF
Copy to flash... done.
Uboot> protect on 10010000 1001FFFF
Protected 1 sectors
Uboot>
```

#### 19. The U-Boot has been upgraded. Reboot your board.

```
boot 1.0 (Aug 8 2003 - 12:29:00)
 Uncompressing image...
 U-Boot 0.3.2 (Jun 24 2003 - 17:04:08)
 U-Boot code: 21F00000 -> 21F14160 BSS: -> 21F1FAEC
 DRAM Configuration:
 Bank #0: 20000000 32 MB
 Atmel: AT49BV1614A (16Mbit)
 Flash: 2 MB
 DataFlash:AT45DB642
 Nb pages: 8192
Page Size: 105
              1056
 Size= 8650752 bytes
 Logical address: 0xC0000000
 *** Warning - bad CRC, using default environment
 Uboot>
            Détect auto 115200 B-N-1 Défit Mai Num Cap
00:28:13 connecté
```

# AT91RM9200DK U-Boot Application Note

Using the Internal BootROM	The procedure for reprogramming the Flash described here is applicable to users of the AT91RM9200DK Development Board who do not have a Green Hills Slingshot JTAG/ICE Interface. The solution presented here uses the Internal BootROM of the AT91RM9200.
	After reset, the AT91RM9200DK boots from the external Flash memory by default (BMS high during reset). To boot using the embedded Boot ROM program (on-chip boot mode with BMS low during reset), it is mandatory to remove the R159 resistor from the AT91RM9200DK development board temporarily.
Hardware and Software Requirements	<ul> <li>UBootFlashProgramming.zip file available with this Application Note</li> <li>A serial cable (Null modem)</li> </ul>
	<ul> <li>HyperTerminal running on your PC application (1115200,8,N,1)</li> </ul>
Connecting the AT91RM9200DK to the Host PC	<ol> <li>Connect the AT91RM9200DK to the Host PC using the serial cable through the Serial Debug Port on the AT91RM9200DK and the serial COM port on the Host PC. Open a HyperTerminal session with the following configuration, 115200 bit rate, 8-bit data length, No parity bit, 1 stop bit and No flux control.</li> </ol>
Loading the Loader Application	<ol> <li>Download the <b>loader.bin</b> application. This application is used to download a binary file from a HyperTerminal session through the Serial Debug Port using the Xmodem protocol.</li> </ol>
	<ol> <li>Extract the AT91RM9200 Getting Started Software Package file called AT91RM9200-GettingStarted-GHS3_6-1_1.exe, available on the UBootFlash- Programming.zip file provided with this Application Note. The loader.bin file to download is available in the software package extracted to the following directory path C:\UBootFlashProgramming\AT91RM9200-GettingStarted-GHS3_6- 1_1\AT91RM9200-GettingStarted\Flash.</li> </ol>
	4. Power-up and reset the AT91RM9200DK Development Board.
	The Internal Boot Program runs and waits for downloading an application code into the internal SRAM via Xmodem protocol through the Serial Debug Port (DBGU). It then branches to the application entry point at the first address of the SRAM.
	5. On the HyperTerminal session, send the file loader.bin using Xmodem protocol.
Loading U-Boot	The loader application is now running on the AT91RM9200DK and is ready to download a binary application file through the Serial Debug Port using the Xmodem protocol on a HyperTerminal session.
	<ol> <li>Download the u_boot.bin binary file using the Xmodem protocol. This file is available available in the UBootFlashProgramming.zip file provided with this Application Note under the following directory path C:\UBootFlashProgramming\binary.</li> </ol>
	In the HyperTerminal session, send the file u_boot.bin using Xmodem protocol. At the end of the Xmodem transfer, a new prompt is displayed in the HyperTerminal window:
	The UBoot is now running on the AT91RM9200DK.
	This Boot program can be used to download files to the SDRAM through the Serial Debug Port using the Kermit protocol with a HyperTerminal session. It is then copied to the Flash.





Downloading U-Boot Binary Files The last step is to download the primary bootstrap, boot.bin, and the gzip-compressed U-Boot image, u-boot.gz. These two files are also available in the UBootFlashProgramming.zip file provided with this Application Note in the directory C:\UBootFlashProgramming\binary.

7. The first file to load is the boot image "boot.bin".

Uboot> loadb 20000000
## Ready for binary (Kermit) download ...
## Start Addr =0x20000000

The boot image is now loaded in SDRAM. The following command copies the SDRAM (0x20000000) to Flash(0x10000000).

Uboot> protect off 1000000 10005fff Unprotect 3 sectors Uboot> erase 10000000 10005fff Erase Flash from 0x10000000 to 0x10005fff... Erasing sector 0 ... ok. Erasing sector 1 ... ok. Erasing sector 2 ... ok. done. Erased 3 sectors. Uboot> cp.b 20000000 10000000 5FFF Copy to flash... done. Uboot> protect on 10000000 10005FFF Protected 3 sectors Uboot>

8. Once the primary bootstrap has been loaded, the U-Boot gzipped image can be copied into Flash. The file to load is the uboot gzipped image "**u-boot.gz**":

Uboot> loadb 2000000
## Ready for binary (Kermit) download ...
## Start Addr =0x2000000
Uboot> protect off 10010000 1001ffff
Unprotect 1 sectors
Uboot> erase 10010000 1001ffff
Erase Flash from 0x10010000 to 0x1001ffff...
Erasing sector 8 ... ok.
done.
Erased 1 sectors.
Uboot> cp.b 20000000 10010000 FFFF
Copy to flash... done.
Uboot> protect on 10010000 1001FFFF
Protected 1 sectors
Uboot>

The U-Boot has been upgraded.

# 10 AT91RM9200DK U-Boot Application Note I

9. Set the R159 resistor on the AT91RM9200DK to boot from the external Flash memory. Reboot your board.

```
boot 1.0 (Aug 8 2003 - 12:29:00)
 Uncompressing image...
 U-Boot 0.3.2 (Jun 24 2003 - 17:04:08)
 U-Boot code: 21F00000 -> 21F14160 BSS: -> 21F1FAEC
 DRAM Configuration:
 Bank #0: 20000000 32 MB
 Atmel: AT49BV1614A (16Mbit)
 Flash: 2 MB
 DataFlash:AT45DB642
 Nb pages: 8192
 Page Size: 1056
 Size= 8650752 bytes
 Logical address: 0xC0000000
 *** Warning - bad CRC, using default environment
 Uboot>
00:28:13 connecté
              Détect auto 115200 8-N-1 Défil Mai Num Captur
```





### **Atmel Corporation**

2325 Orchard Parkway San Jose, CA 95131, USA Tel: 1(408) 441-0311 Fax: 1(408) 487-2600

### **Regional Headquarters**

#### Europe

Atmel Sarl Route des Arsenaux 41 Case Postale 80 CH-1705 Fribourg Switzerland Tel: (41) 26-426-5555 Fax: (41) 26-426-5500

#### Asia

Room 1219 Chinachem Golden Plaza 77 Mody Road Tsimshatsui East Kowloon Hong Kong Tel: (852) 2721-9778 Fax: (852) 2722-1369

#### Japan

9F, Tonetsu Shinkawa Bldg. 1-24-8 Shinkawa Chuo-ku, Tokyo 104-0033 Japan Tel: (81) 3-3523-3551 Fax: (81) 3-3523-7581

### **Atmel Operations**

Memory

2325 Orchard Parkway San Jose, CA 95131, USA Tel: 1(408) 441-0311 Fax: 1(408) 436-4314

#### Microcontrollers

2325 Orchard Parkway San Jose, CA 95131, USA Tel: 1(408) 441-0311 Fax: 1(408) 436-4314

La Chantrerie BP 70602 44306 Nantes Cedex 3, France Tel: (33) 2-40-18-18-18 Fax: (33) 2-40-18-19-60

#### ASIC/ASSP/Smart Cards

Zone Industrielle 13106 Rousset Cedex, France Tel: (33) 4-42-53-60-00 Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd. Colorado Springs, CO 80906, USA Tel: 1(719) 576-3300 Fax: 1(719) 540-1759

Scottish Enterprise Technology Park Maxwell Building East Kilbride G75 0QR, Scotland Tel: (44) 1355-803-000 Fax: (44) 1355-242-743

#### **RF**/Automotive

Theresienstrasse 2 Postfach 3535 74025 Heilbronn, Germany Tel: (49) 71-31-67-0 Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd. Colorado Springs, CO 80906, USA Tel: 1(719) 576-3300 Fax: 1(719) 540-1759

#### Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine BP 123 38521 Saint-Egreve Cedex, France Tel: (33) 4-76-58-30-00 Fax: (33) 4-76-58-34-80

*Literature Requests* www.atmel.com/literature

**Disclaimer:** Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

© Atmel Corporation 2004. All rights reserved. Atmel<sup>®</sup> and combinations thereof are the registered trademarks of Atmel Corporation or its subsidiaries. Green Hills<sup>®</sup> and MULTI<sup>®</sup> are the registered trademarks, and Slingshot<sup>™</sup> and Green Hills Probe<sup>™</sup> are the trademarks of Green Hills Software, Inc. ARM<sup>®</sup> is the registered trademark of ARM Ltd. Other terms and product names may be the trademarks of others.

