Armadillo-210

Software Manual

Version 1.0.2 Pre1

August 11, 2006

Atmark Techno, Inc.
http://www.atmark-techno.com/

Armadillo Official Web Site
http://armadillo.atmark-techno.com/

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. About This Manual

This manual provides information necessary for using Armadillo-210. Outline of contents is as follows:

- Rewriting Flash Memory
- Basic Operations
- Building Kernel and Userland
- Application Software Development

This document is described assuming that the base image of Armadillo-201 is configured with the default settings. For information about the recover image of Armadillo-210, refer to Armadillo-210 Startup Guide.

We hope the information in this document will help you to get the best functionality out of the Armadillo-210.

1.2. Typographical Conversions

The following typographical conversions are used in this manual.

Table 1-1 Fonts

Fonts	Description
Fonts in text	Text
[PC ~]$ **ls**	Prompt and user input character strings

1.3. Conventions in Command Input Examples

The command input examples contained in this manual are based on the execution environment associated with the respective display prompt. The directory part "/" differs depending on the current directory. The home directory of each user is represented by "~".

Table 1-2 Relationship between Display Prompt and Execution Environment

Prompt	Command Execution Environment
[PC /]#	To be executed by a privileged user on Work PC
[PC /]$	To be executed by a general user on Work PC
[a210 /]#	To be executed by a privileged user on Armadillo-210
[a210 /]$	To be executed by a general user on Armadillo-210

1.4. Acknowledgements

The software used in the Armadillo-210 is composed from Free Software / Open Source Software. This Free Software / Open Source Software are the achievements of many developers from all over the world. We would like to take this opportunity to thank all these developers.

1.5. Precautions

The software and documentation contained in this product is provided “AS IS” without warranty of any kind including warranty of merchantability or fitness for a particular purpose, reliability, correctness or accuracy. Furthermore, we do not guarantee any outcomes resulting from the use of this product.

2. Getting Started

2.1. Preparation

Please prepare the following hardware and software before using the Armadillo-210.

- Work PC
  A PC that runs either Linux or Windows and has at least one serial port

- Serial Cross Cable and RS232C Level Conversion Adapter
  A D-Sub9 pin (female-to-female) cable for cross connections. Connect the yellow cable of RS232C level conversion adapter to the pin-1 of CON4.

- Supplied CD-ROM (hereafter called “CD-ROM”)
  This CD-ROM contains various manuals and source code related to Armadillo-210.

- Serial Console Software
  Please install serial console software such as minicom or Tera Term on the work PC (Software for Linux is contained in the CD-ROM (the “tools” directory).

2.2. Connecting Cables

Connect the serial cross cable, RS232C level conversion adapter, AC adapter and LAN cable to the Armadillo-210 as shown in Figure 2-1.

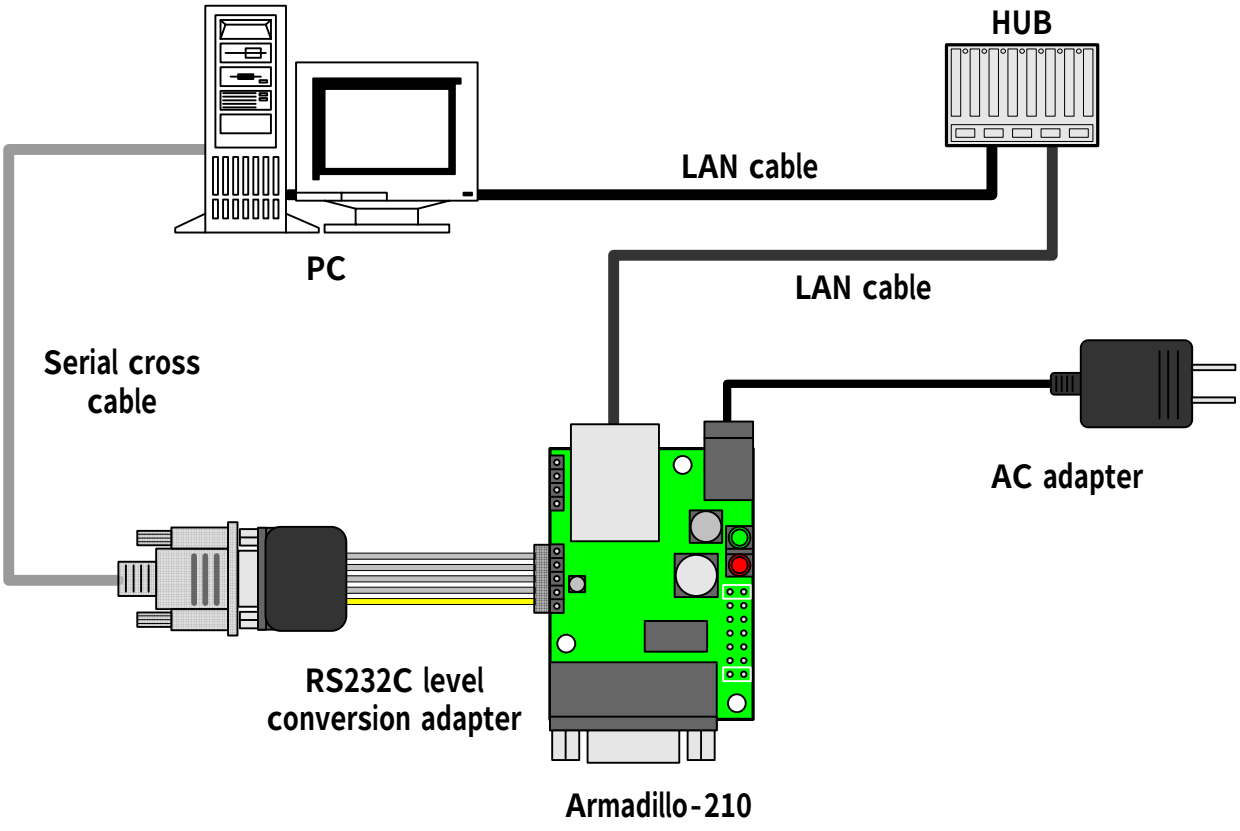


Figure 2-1 Example of Armadillo-210 Cable Connections

2.3. Jumper Pin Setting

The boot operation can be set with the jumpers on the Armadillo-210. The jumper setting and its corresponding operation are shown in Table 2-1.

Table 2-1 Jumper Settings and Boot Operation

JP1	JP2	Operation at Booting
OPEN	OPEN	**Linux kernel will boot up.**
OPEN	SHORT	**Hermit command prompt will boot up.**
SHORT	–	**CPU on-chip Boot ROM (*1) will boot up.**

*1: This is used for restoring the bootloader etc.

TIPS

Definition of OPEN and SHORT

- OPEN : A jumper socket is not placed in the jumper
- SHORT : A jumper socket is placed in the jumper

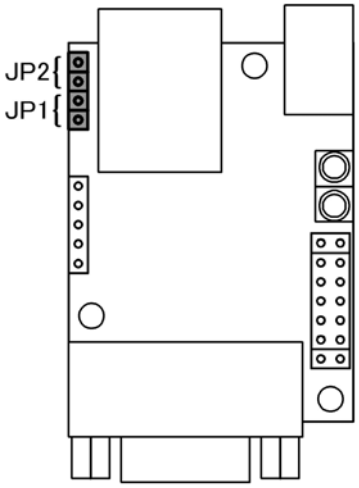


Figure 2-2 Jumper Position

3. Development Environment

The software cross-development for Armadillo-210 can be carried out on the work PC.

3.1. Installing Cross Development Environment Packages

Install the cross development environment packages that are contained in the CD-ROM ("cross-dev" directory). This installation must be done under root privilege. The following packages are available.

Table 3-1 Cross Development Environment Packages

Package Name	Version	Description
binutils-arm-linux	2.15-6	The GNU Binary utilities
cpp-3.4-arm-linux	3.4.3-13	The GNU C preprocessor
g++-3.4-arm-linux	3.4.3-13	The GNU C++ compiler
gcc-3.4-arm-linux	3.4.3-13	The GNU C compiler
libc6-arm-cross	2.3.2.ds1-22	GNU C Library: Shared libraries and Timezone data
libc6-dev-arm-cross	2.3.2.ds1-22	GNU C Library: Development Libraries and Header Files
libc6-pic-arm-cross	2.3.2.ds1-22	GNU C Library: PIC archive library
libc6-prof-arm-cross	2.3.2.ds1-22	GNU C Library: Profiling Libraries
libdb1-compat-arm-cross	2.1.3-7	The Berkeley database routines
libgcc1-arm-cross	3.4.3-13	GCC support library
libstdc++6-0-arm-cross	3.4.3-13	The GNU Standard C++ Library v3
libstdc++6-0-dbg-arm-cross	3.4.3-13	The GNU Standard C++ Library v3 (debugging files)
libstdc++6-0-dev-arm-cross	3.4.3-13	The GNU Standard C++ Library v3 (development files)
libstdc++6-0-pic-arm-cross	3.4.3-13	The GNU Standard C++ Library v3 (shared library subset kit)
linux-kernel-headers-arm-cross	2.5.999-test7-bk-17	Linux Kernel Headers for development
libncurses5-arm-cross	5.4-9	Shared libraries for terminal handling
libncurses5-dev-arm-cross	5.4-9	Developer's libraries and docs for ncurses
libssl0.9.7-arm-cross	0.9.7g-1	SSL shared libraries
libssl-dev-arm-cross	0.9.7g-1	SSL development libraries, header files and documentation
zlib1g-arm-cross	1.2.3-3	compression library - runtime
zlib1g-dev-arm-cross	1.2.3-3	compression library - development

The package files included are deb for Debian distributions, rpm for Red Hat distributions and tgz (non-installer). Choose any one appropriate for your operating system in use.

```
[PC ~]# dpkg -i binutils-arm-linux_2.14.90.0.7-8_i386.deb ←when deb package is used

[PC ~]# rpm -i binutils-arm-linux-2.14.90.0.7-8.i386.rpm  ←when rpm package is used

[PC ~]# tar zxf binutils-arm-linux-2.14.90.0.7.tgz -C /   ←when tgz is used
```

Figure 3-1 Example of Extracting a Development Package

Attempt the following if the installation fails due to the dependency

```
[PC ~]# dpkg -i xxx.deb yyy.deb ←when two or more deb packages are installed
                                                    simultaneously
```

Figure 3-2 Example of Extracting Packages

3.2. Packages Required for Building atmark-dist

The packages shown in Table 3-2 must be installed to build atmark-dist in the work PC. Install them to meet the environment of the work PC in use.

Table 3-2 Packages Required for Building atmark-dist

Package Name	Version	Description
genext2fs	1.3-7.1-cvs20050225	ext2 filesystem generator for embedded systems
file	4.12-1 or later	Determines file type using "magic" numbers
sed	4.1.2-8 or later	The GNU sed stream editor
perl	5.8.4-8 or later	Larry Wall's Practical Extraction and Report Language

* The genext2fs package file is contained in the CD-ROM (“tools” directory).

3.3. Creating a Cross Development Package

The packages not contained in the supplied CD may be required when you develop application software. This section describes how to create ARM cross development packages.

First, obtain the package that can be a source of the cross package to be created. For instance, the "libncurses5_x.x-x_arm.deb" package is obtained for libncurses5.

The obtained package is converted for cross development with the following command.

```
[PC ~]$ dpkg-cross --build --arch arm libncurses5_x.x-x_arm.deb
[PC ~]$ ls
libncurses5-arm-cross_x.x-x_all.deb libncurses5_x.x-x_arm.deb
```

Figure 3-3 Creating a Cross Development Package (deb)

The cross package named "libncurses5-arm-cross_x.x-x_all.deb" is created. This is a package for deb. If necessary, create a package for rpm or tgz. The method to create a package for rpm and tgz is shown in Table 3-4.

```
[PC ~]# alien -r -k libncurses5-arm-cross_x.x-x_all.deb     ←Create rpm package
[PC ~]# alien -t -k libncurses5-arm-cross_x.x-x_all.deb     ←Create tgz
[PC ~]$ ls
libncurses5-arm-cross_x.x-x_all.deb libncurses5_x.x-x_arm.deb
libncurses5-arm-cross-x.x-x.noarch.rpm libncurses5-arm-cross_x.x.tgz
```

Figure 3-4 Creating a Cross Development Package (rpm, tgz)

4. How to Use Armadillo-210

This chapter focuses on how to use the Armadillo-210.

4.1. Before Boot-up

Connect the serial port 2 (CON4) on the Armadillo-210 and the work PC with a serial cable, and activate serial console software. Set values for serial communication parameters as shown in Table 4-1.

Table 4-1 Serial Communication Settings

Parameter	Value
Transfer Rate	115,200bps
Data Length	8bit
Stop Bit	1bit
Parity	None
Flow Control	None

4.2. Boot-up

Linux will boot up when JP1 and JP2 are set to OPEN and the power is turned on. The following log is output on the serial port 2 when Linux boots up successfully.

```
Uncompressing kernel.....done.
Uncompressing ramdisk..... done.
Doing console=ttyAM1,115200
Doing mtdparts=armadillo210-nor:0x10000(bootloader)ro,0x170000(kernel),0x270000(userland),-(config)
Linux version 2.6.12.3-a9-2 (atmark@pc-nsx) (gcc version 3.4.4 20050314 (prerelease) (Debian 3.4.3-13)) #2 Wed Dec 14 14:57:48 JST 2005
CPU: ARM920Tid(wb) [41129200] revision 0 (ARMv4T)
CPU0: D VIVT write-back cache
CPU0: I cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
CPU0: D cache: 16384 bytes, associativity 64, 32 byte lines, 8 sets
Machine: Armadillo-210
ATAG_INITRD is deprecated; please update your bootloader.
Memory policy: ECC disabled, Data cache writeback
Built 1 zonelists
Kernel command line: console=ttyAM1,115200 mtdparts=armadillo210-nor:0x10000(bootloader)ro,0x170000(kernel),0x270000(userland),-(config)
PID hash table entries: 256 (order: 8, 4096 bytes)
Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)
Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)
Memory: 8MB 8MB 16MB = 32MB total
Memory: 23072KB available (2166K code, 429K data, 96K init)
Mount-cache hash table entries: 512
CPU: Testing write buffer coherency: ok
checking if image is initramfs...it isn't (bad gzip magic numbers); looks like an initrd
Freeing initrd memory: 6592K
NET: Registered protocol family 16
NetWinder Floating Point Emulator V0.97 (double precision)
Initializing Cryptographic API
gpio: Armadillo-210 GPIO driver, (C) 2005 Atmark Techno, Inc.
led: Armadillo-210 LED driver, (C) 2005 Atmark Techno, Inc.
ttyAM0 at MMIO 0x808c0000 (irq = 52) is a EP93XX
ttyAM1 at MMIO 0x808d0000 (irq = 54) is a EP93XX
ttyAM2 at MMIO 0x808e0000 (irq = 55) is a EP93XX
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered
RAMDISK driver initialized: 16 RAM disks of 16384K size 1024 blocksize
loop: loaded (max 8 devices)
i2c /dev entries driver
i2c-armadillo9: i2c Armadillo-9 driver, (C) 2004-2005 Atmark Techno, Inc.
i2c-at24cxx: i2c at24cxx eeprom driver, (C) 2003-2005 Atmark Techno, Inc.
armadillo210-nor: Found 1 x16 devices at 0x0 in 16-bit bank
  Amd/Fujitsu Extended Query Table at 0x0040
armadillo210-nor: CFI does not contain boot bank location. Assuming top.
number of CFI chips: 1
cfi_cmdset_0002: Disabling erase-suspend-program due to code brokenness.
4 cmdlinepart partitions found on MTD device armadillo210-nor
parse_mtd_partitions:4
Creating 4 MTD partitions on "armadillo210-nor":
0x00000000-0x00010000 : "bootloader"
0x00010000-0x00180000 : "kernel"
0x00180000-0x003f0000 : "userland"
0x003f0000-0x00400000 : "config"
NET: Registered protocol family 2
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP established hash table entries: 2048 (order: 2, 16384 bytes)
TCP bind hash table entries: 2048 (order: 1, 8192 bytes)
TCP: Hash tables configured (established 2048 bind 2048)
IPv4 over IPv4 tunneling driver
ip_tables: (C) 2000-2002 Netfilter core team
Initializing IPsec netlink socket
NET: Registered protocol family 1
NET: Registered protocol family 10
Disabled Privacy Extensions on device c025bab0(lo)
IPv6 over IPv4 tunneling driver
NET: Registered protocol family 17
```

```
NET: Registered protocol family 15
SCTP: Hash tables configured (established 1024 bind 2048)
RAMDISK: ext2 filesystem found at block 0
RAMDISK: Loading 6592KiB [1 disk] into ram disk... done.
VFS: Mounted root (ext2 filesystem).
Freeing init memory: 96K
init started:  BusyBox v1.00 (2005.12.14-06:32+0000) multi-call binary
Starting fsck for root filesystem.
fsck 1.25 (20-Sep-2001)
ext2fs_check_if_mount: No such file or directory while determining whether /dev/ram0 is mounted.
/dev/ram0: clean, 556/1024 files, 4897/6592 blocks
Checking root filesystem:                                   done
Remounting root rw:                                        done
Mounting proc:                                              done
Setting hostname:                                           done
Cleaning up system:                                        done
Running local start scripts.
Changing file permissions:                                  done
Starting syslogd:                                           done
Starting klogd:                                             done
Starting basic firewall:                                   done
Loading /etc/config:                                       done
Configuring network interfaces:                             done
Starting inetd:                                            done
Stating sshd:                                              done
Starting thttpd:                                            done

atmark-dist v1.4.0 (AtmarkTechno/Armadillo-210.Base)
Linux 2.6.12.3-a9-2 [armv4tl arch]

A210 login:
```

Figure 4-1 Boot Log

For the base image of Userland, the login prompt is displayed on both the serial port 1 (CON2) and the serial port 2 (CON4).

The following two types of login users can be defined.

Table 4-2 User Name and Password for Console Login

User Name	Password	Privilege
root	root	Privileged user
guest	(None)	General user

4.3. Directory Structure

The directory structure on the Armadillo-210 is shown in Table 4-3.

Table 4-3 Directory

Directory Name	Description
/bin	Application
/dev	Device nodes
/etc	System settings
/etc/network	Network settings
/lib	Common libraries
/mnt	Mount points
/proc	Process information
/root	root home directory
/sbin	System management commands
/usr	Common user data
/home	User home directories
/home/ftp/pub	ftp data transfer
/tmp	Temporary backup
/var	Modified data

4.4. Shutdown

The Armadillo-210 can be shutdown by turning the power switch off.

4.5. Network Setting

Network setting can be altered by editing the /etc/network/interfaces file on the Armadillo-210.

4.5.1. Using a Fixed IP Address

The table 4-4 shows an example of network settings when a fixed IP address is used.

Table 4-4 Network Settings

Parameter	Value
IP Address	192.168.10.10
Net Mask	255.255.255.0
Broadcast Address	192.168.10.255
Default Gateway	192.168.10.1

```
# /etc/network/interfaces – configuration file for ifup(8), ifdown(8)

auto lo eth0

iface lo inet loopback

iface eth0 inet static
        address 192.168.10.10
        netmask 255.255.255.0
        network 192.168.10.0
        broadcast 192.168.10.255
        gateway 192.168.10.1
```

Figure 4-2 Example of Network Settings (Using Fixed IP Address)

If a gateway is not used, delete the whole line for specifying a gateway or comment out.

```
#       gateway 192.168.10.1 ←Commented out by adding # at the head of the line.
```

Figure 4-3 Example of Network Settings (Disabling Gateway)

4.5.2. Using DHCP

Figure 4-4 shows an example where DHCP is used to obtain an IP address.

```
# /etc/network/interfaces – configuration file for ifup(8), ifdown(8)

auto lo eth0

iface lo inet loopback

iface eth0 inet dhcp
```

Figure 4-4 Example of Network Settings (Using DHCP)

4.5.3. Enabling Network Settings

Once networking setting is completed, you can establish a network connection with that setting by executing the “/etc/init.d/networking” script.

If a network connection is already established, it is needed to close that connection. For a fixed IP network connection you can close it with the ifconfig command and for a DHCP connection with the dhcpcd command with -k option.

```
[a210 /]# ifconfig eth0 down ←Closing a fixed IP network connection

[a210 /]# dhcpcd -k          ←Closing a DHCP connection
```

Figure 4-5 Closing a Network Connection

```
[a210 /]# /etc/init.d/networking
```

Figure 4-6 Enabling a New Network Connection

4.5.4. How to Set a Fixed IP Network Connection Using DHCP Default Settings

The default setting of the Armadillo-210 network connection is DHCP. This section describes how to set a fixed IP network connection using this default setting. For information about how to set it after the boot-up of the Armadillo-210, refer to Section 4.5.1, "Using a Fixed IP Address".

To set it automatically when the Armadillo-210 boots up, it is required to recreate an image (romfs.img) to be written into the Armadillo-210.
Before creating an image, edit the atmark-dist/romfs/etc/network/interfaces according to the procedure described in Section 4.5.1, "Using a Fixed IP Address".

```
[PC ~/atmark-dist]$ vi romfs/etc/network/interfaces
//--- File to edit
# /etc/network/interfaces – configuration file for ifup(8), ifdown(8)

auto lo eth0

iface lo inet loopback

iface eth0 inet static
        address 192.168.10.10
        netmask 255.255.255.0
        network 192.168.10.0
        broadcast 192.168.10.255
        gateway 192.168.10.1
//--- End of File
[PC ~/atmark-dist]$
```

Then, create a writing image.

```
[PC ~/atmark-dist]$ make image
[PC ~/atmark-dist]$ ls images
linux.bin   linux.bin.gz   romfs.img   romfs.img.gz
```

Write the created romfs.img.gz into the Armadillo-210.

Caution

In this method, interface files are initialized if make or make romfs is carried out.
If you want a fixed IP network connection to be always set, edit the atmark-dist/vendors/AtmarkTechno/ Armadillo-210/etc/network/interfaces.

4.6. telnet Login

You can log into the system with the following user name and password. If you need root privileges for a task, log into the system as guest first and then change to root with the su command.

Table 4-5 User Name and Password for telnet Login

User Name	Password
guest	None

4.7. File Transfer

The Armadillo-210 supports FTP file transfer. Login the system with the following user name and password. The home directory is /home/ftp. Go to the /home/ftp/pub directory to upload data.

Table 4-6 User Name and Password for ftp

User Name	Password
ftp	None

4.8. Web Server

A small HTTP server called thttpd is run which allows the user to browse the Armadillo-210 from a Web browser.

Data directory: /home/www-data
URL: http://(IP address for Armadillo-210)/ (Example, http://192.168.10.10/)

4.9. ssh Login

You can log into the system with the following user name and password. If you need root privileges for a task, log into the system as guest first and then change to root with the su command.

Table 4-7 User Name and Password for ssh Login

User Name	Password
guest	None

5. Rewriting Flash Memory

The functionality of Armadillo-210 can be altered by rewriting the Flash memory. This chapter describes how to rewrite the Flash memory.

Caution

If the downloading of the image fails for any reason, the Armadillo-210 may not boot normally. Be careful about the following points when performing a rewriting.

- Do not power off the Armadillo-210.
- Do not disconnect the serial cable connecting the Armadillo-210 to the development PC.

5.1. Installing the Downloader

Install the Downloader (hermit) on the work PC. The downloader is used to rewrite the Armadillo-210 flash memory.

1) Linux:

Install the package files which are contained in the CD-ROM. This must be done by a user with root privileges.
Three package files are available; deb (Debian), rpm (Red Hat) and tgz (non-installer). Select any one suitable for your operating system in use.

```
[PC ~]# dpkg -i hermit-at-1.0.7_i386.deb ←when the deb package is used.

[PC ~]# rpm -i hermit-at-1.0.7-1.rpm      ←when the rpm package is used.

[PC ~]# tar zxf hermit-at-1.0.7-source.tgz -C /      ←when tgz is used.
```

Figure 5-1 Example of Expand Commands

2) Windows

Expand the Hermit-At WIN32 (downloader/win32/hermit-at-win_xxxxxxxx.zip) contained in the CD-ROM to an appropriate folder.

5.2. Specifying Memory Region

You can specify by region name the address of the Flash memory into which data is written. Three types of region names are available as shown below.

- **bootloader**
  This is a region to store an image of software that is first executed when the power switch is turned on. This region or bootloader provides the function to rewrite the Flash memory via a serial or boot up the operating system.

- **kernel**
  This is a region to store an image of Linux kernel. The kernel stored in this region is booted by the bootloader.

- **userland**
  This is a region to store an image of systems including applications such as telnet, ftp and Web server, various configuration files and user data.

The image file for each of the above region is contained in the CD-ROM (*image directory*).

Table 5-1 Image File Name for Each Region

Region	File Name
bootloader	loader-armadillo2x0-x.bin
kernel	linux-a210-x.xx.bin.gz
userland	romfs-a210-recover-x.xx.img.gz romfs-a210-base-x.xx.img.gz

For the Flash memory map, refer to Section 8, "Memory Maps".

5.3. Rewriting Procedure

Rewriting of the Flash memory is carried out according to the following procedure.

5.3.1. Jumper Pin Setting

Before the Armadillo-210 power switch is turned on, set jumper pins as follows.

- JP1：OPEN
- JP2：SHORT

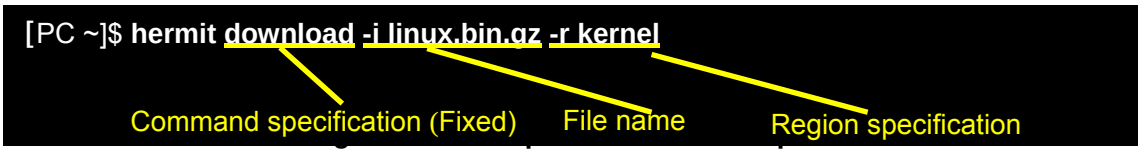
For more details on jumper setting, refer to Section 2.3, “Jumper Pin Setting”.

5.3.2. Transferring a Rewrite Image

First, connect the work PC to the serial port (CON2) on the Armadillo-210 with a serial cable and turn on the power switch on the Armadillo-210.
The procedure after that differs depending on the operating system in use

1) Linux

Boot up a terminal with the work PC that runs Linux. Then, specify a kernel image file and a region and enter the hermit command.
In the following example, the kernel image (linux.bin.gz) is specified as a file name (three options are available; bootloader, kernel and userland).



If the serial port to be used on the work PC is not “ttyS0”, add option “--port “port name”.

TIPS

The option --force-locked must be added when rewriting the bootloader region (rgion:bootloader / adress:0x60000000-0x6000ffff). If it is not added, a warning message is displayed and writing to the bootloader region is not carried out.

Caution

If a wrong image is written to the bootloader region, you can not boot up the system from the onboard Flash memory. In this case, restore the bootloader according to the procedure described in Section 6.4.1, “Resetting the Bootloader to a Default State”.

Once rewriting is completed and JP2 is set to OPEN, the Armadillo-210 will boot up with new image.

2) Windows

Activate the Hermit-At WIN32 (hermit.exe) contained in the folder where the file was extracted according to the procedure described in Section 5.1, “Installing the Downloader”.

Click the Download button. A screen as shown in Figure 5-3 is displayed.

On the Serial Port field, set the serial port that is connected to the Armadillo-210.
Specify the image file to be written in the Image field. You can specify it with the file dialog.

On the Region field, specify the region or the address to which an image file is written.

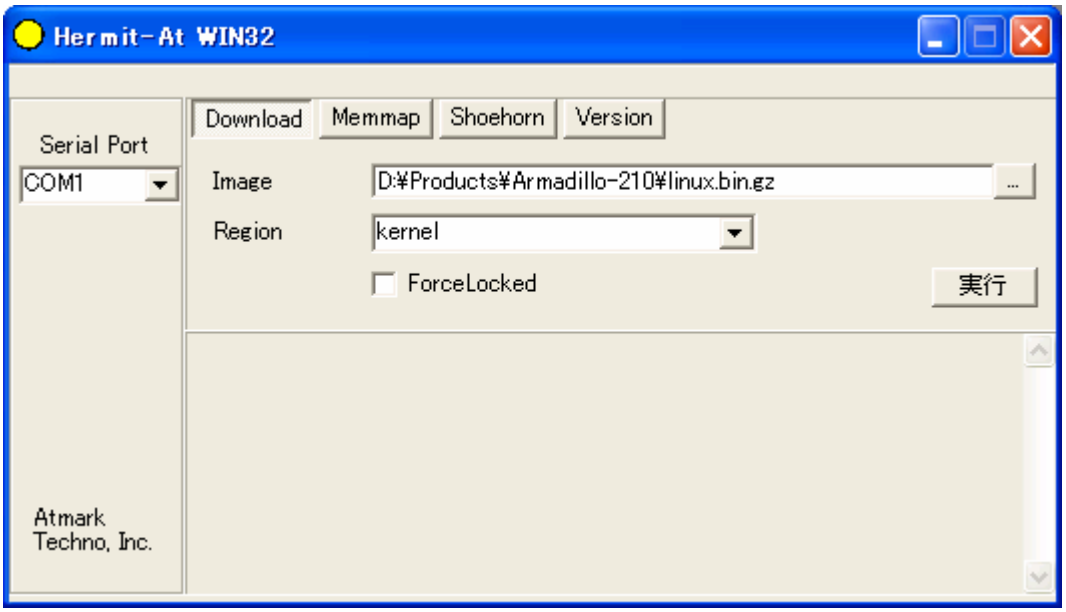


Figure 5-3 Download Screen

Click the Execute button to start writing or downloading to the Flash memory. A progress dialog appears as shown in Figure 5-4. The dialog is closed automatically when rewriting is completed.

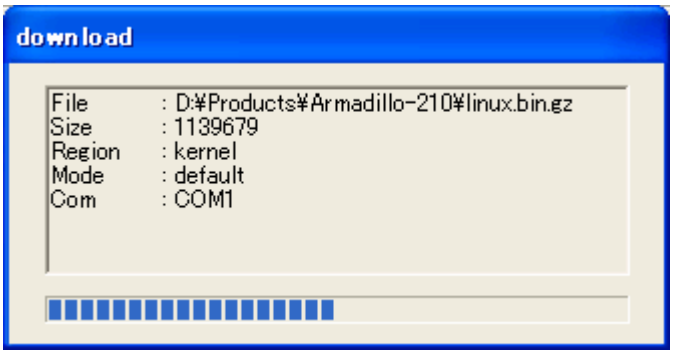


Figure 5-4 A Progress Dialog

TIPS

The ForceLocked box must be checked when rewriting the bootloader region (rgion:bootloader / adress:0x60000000-0x6000ffff). If it is not, a warning message is displayed and writing to the bootloader region is not carried out.

Caution

If a wrong image is written to the bootloader region, you can not boot up the system from the onboard Flash memory. In this case, restore the bootloader according to the procedure described in Section 6.4.1, "Resetting the Bootloader to a Default State".

Once rewriting is completed and JP2 is set to OPEN, the Armadillo-210 will boot up with new image.

5.4. Rewriting the Flash Memory with netflash

You can use the user application netflash to rewrite the Flash memory. Here, we introduce a method to rewrite the Flash memory with the netflash.

Caution

If rewriting the Flash memory fails for any reason, the Armadillo-210 may not boot normally. Be careful not to turn off the power switch on the Armadillo-210 during rewrite processing.

The netflash obtains a file from the HTTP or FTP server to write it into the Flash memory. First, the image file must be placed on the HTTP or FTP server.

A command example to alter the kernel image on the Armadillo-210 is shown in Figure 5-5.

```
[a210 ~]# netflash   -k    -n   -r /dev/flash/kernel
                              Option      Region specification
            http://download.atmark-techno.com/armadillo-210/image/linux-a210-1.02.bin.gz
                                        File name

*It is generally one line command.
```

Figure 5-5 netflash Command Example

Specify a region with the option “-r /dev/flash/kernel” according to the following table.

Kernel	/dev/flash/kernel
Userland	/dev/flash/userland

The netflash HELP can be viewed with the following command.

```
[a210 ~]# netflash -h
```

Figure 5-6 netflash Help Command

6. Bootloader

This chapter focuses on the bootloader of the Armadillo-210.

6.1. Preparing Packages

Copy the following packages from the downloader directory in the CD-ROM to the work PC.

Table 6-1 List of Bootloader Related Packages

Package Name	Description
hermit-at-x.x.x	A downloader that operates with the Armadillo-210 boot program （Inclusive of the Armadillo-210 boot program）
shoehorn-at-x.x.x	A downloader that operates with the CPU on-chip boot ROM

For information about how to install these packages, refer to Section 3.1, "Installing Cross Development Environment Packages".

6.2. Type of Bootloaders

Four types of bootloaders are available for the Armadillow-210 as shown in Table 6-2.

Table 6-2 Bootloader List

Bootloader Name	Description
loader-armadillo2x0	The serial port 1 is used for the hermit console.
loader-armadillo2x0-eth	The standard bootloader written in the Flash memory at shipment. The serial port 1 is used for the hermit console. The bootloder allows rewriting the Flash memory via TFTP
loader-armadillo2x0-ttyAM1	The serial port 2 is used for the hermit console.
loader-armadillo2x0-notty	The hermit console is not used.

6.3. Creating a Bootloader

The CD-ROM contains all these bootloaders which have been listed in the previous section. Besides them, you can also create an original version of bootloader by building from the source code.

6.3.1. Preparing the Source Code

From the source/bootloader directory contained in the CD-ROM copy the hermit-at-x.x.x-source.tar.gz onto the work PC.

```
[PC ~]$ tar zxf hermit-at-x.x.x-source.tar.gz
```

6.3.2. Build

Go to the directory which has been created as a result of extracting the above file. Then, enter the make command.

```
[PC ~]$ cd hermit-at-x.x.x
[PC ~]$ make TARGET=armadillo2x0
```

When the make is completed, the bootloader image file is created in the hermit-at-x.x.x/src/target/armadillo2x0 directory.

6.4. CPU with an On-chip Boot ROM

This section describes how to rewrite the Armadillo-210 bootloader into which the loader-armadillo2x0-notty is written. It also provides countermeasures to be implemented when the Armadillo-210 does not boot normally due to incorrect writing to the bootloader.

The Armadillo-210 has a CPU with an on-chip boot ROM. Resetting the bootloader to a default state can be performed using the software contained in this ROM. The following subsection describes how to do this.

6.4.1. Resetting the Bootloader to a Default State

1) Linux:
   1. Make sure that the Armadillo-210 power switch is turned off. Then, connect the serial port 1 on the Armadillo-210 to the serial port on the work PC with a cross (reverse) serial cable.
   2. Set the jumper JP1 on the Armadillo-210 to SHORT.
   3. Activate the shoehorn on the work PC.

```
[PC ~]$ shoehorn --boot --terminal --initrd /dev/null
            --kernel /usr/lib/hermit/loader-armadillo2x0-boot.bin
            --loader /usr/lib/shoehorn/shoehorn-armadillo2x0.bin
            --initfile /usr/lib/shoehorn/shoehorn-armadillo2x0.init
            --postfile /usr/lib/shoehorn/shoehorn-armadillo2x0.post
```

Figure 6-1 shoehorn Command Example

The above figure shows an example when the Armadillo-210 is connected to the serial port "/dev/ttyS0" on the work PC.
Add the following option to the shoehorn command when the Armadillo-210 is connected to another serial port.

--port [Serial port name]

The command input must be one line.

   4. Turn on the Armadillo-210 power switch.

      A command message is displayed at once. If it is not displayed normally, turn off the power switch on the Armadillo-210 and make sure that the serial cable and the Armadillo-210 jumper (JP1) connections are correct.

   5. When a "hermit>" message appears, enter Ctrl+C.

Now you are ready to download the bootloader to the Armadillo-210 from the work PC using hermit. Rewrite the bootloader according to the procedure described in Section 5, "Rewriting Flash Memory" without changing current jumper settings and turning off the power switch on the Armadillo-210.

2) Windows:
   1. Make sure that the Armadillo-210 is turned off. Then, connect the serial port 1 on the Armadillo-210 to the serial port on the work PC with a cross (reverse) serial cable.
   2. Set the jumper JP1 on the Armadillo-210 to SHORT.
   3. Activate the Hermit-At-WIN32 on the work PC.
   4. Click the Shoehorn button.

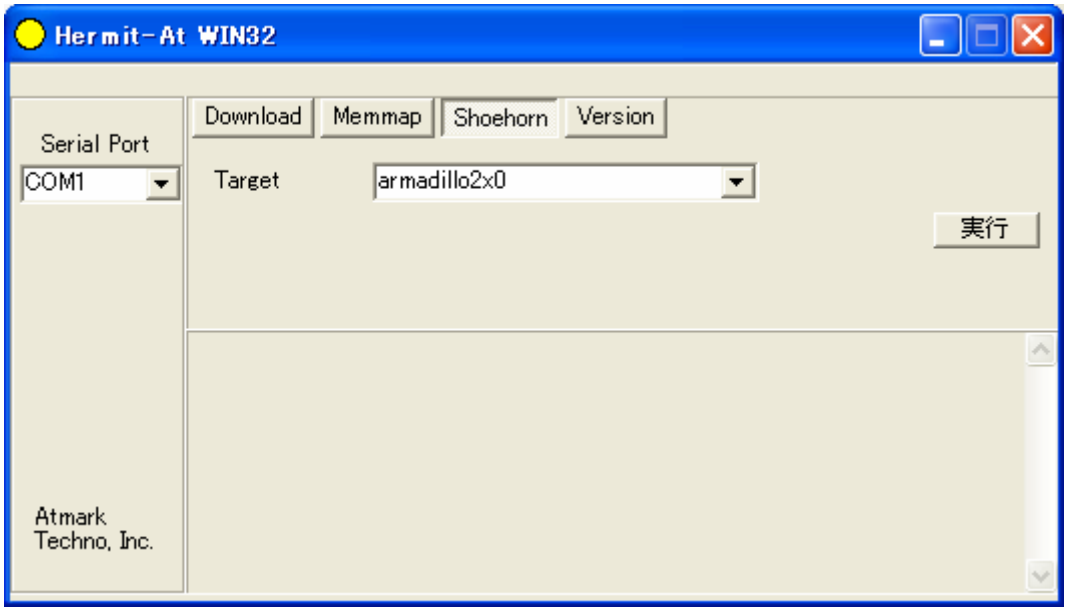


Figure 6-2 Shoehorn Screen

   5. On the Target field, specify “armadillo2x0”.
   6. Click the Execute button. A shoehorn dialog as shown in Figure 6-3 is displayed.

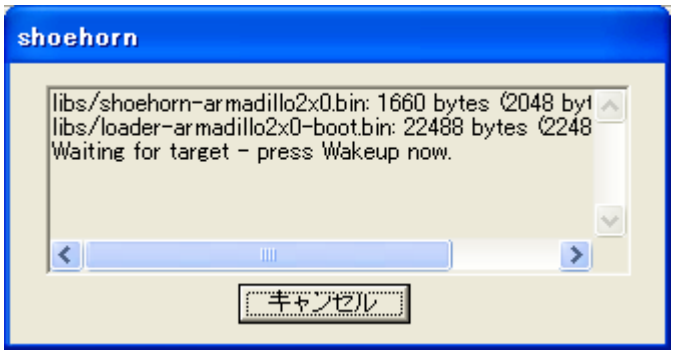


Figure 6-3 shoehorn Dialog

   7. Turn on the power switch on the Armadillo-210.

A command message is displayed at once. If it is not displayed normally, turn off the power switch on the Armadillo-210 and make sure that the serial cable and the Armadillo-210 jumper (JP1) connections are correct.

Now you are ready to download the bootloader to the Armadillo-210 from the work PC using hermit. Rewrite the bootloader according to the procedure described in Section 5, “Rewriting Flash Memory” without changing current jumper settings and turning off the power switch on the Armadillo-210.

6.5. Linux Boot Options

The Armadillo-210 provides an option to automatically boot Linux. The settings are stored on the Flash memory and will be enabled when Linux starts next time.

These Linux boot options can be set using the Hermit command prompt.

TIPS

To determine the appropriate Linux boot option to be set, you need accurate knowledge about Linux kernel in use. For more information about these options and effect, refer to the documents associated with Linux and those supplied with the source file.

6.5.1. Activating Hermit Command Prompt

1. Activating Serial Console Software
   Connect the serial port 1 on the Armadillo-210 to the work PC with a serial cable, activate the serial console software and set the serial communication parameters as shown in Table 6-3.

Table 6-3 Serial Communication Settings

Parameter	Setting
Transfer Rate	115,200bps
Data Length	8bit
Stop Bit	1bit
Parity	None
Flow Control	None

2. Setting Jumper Pins
   Before turning on the Armadillo-210 the power switch, set the jumper pins as follows:
   - JP1: OPEN
   - JP2: SHORT

   For more information about the jumper pin settings, refer to Section 2.3, "Jumper Pin Setting".

3. Booting the Armadillo-210
   Turn on the Armadillo-210power switch. The following Hermit command prompt will be displayed.

```
Hermit-At v1.0.7 (Armadillo-210C/eth) compiled at 00:00:00, Jan 1 2005
hermit>
```

6.5.2. Setting Linux Boot Options

The setenv command is used to set the Linux boot options from the Hermit command prompt. Enter the Linux boot option to be set following the setenv command.

```
hermit> setenv console=ttyAM1,115200
```

Caution

If the Linux boot options are not set (default), the bootloader will automatically set the serial port 2 (ttyAM1) to “CONSOLE” using the option “console=ttyAM1,115200” when Linux is activated. However, if any appropriate option is set using seteny command, this option is not automatically enabled.
When you want to use the serial console even in seteny mode of operation, include the “console=ttyAM1,115200” in the option settings.

To activate Linux using the set boot option, turn off the power switch on the Armidillo-210, set the jumpers properly and turn on the power switch again.

6.5.3. Identifying Linux Boot Option Settings

To identify the current Linux boot option setting, enter the setenv command without parameters to display it.

```
hermit> setenv
1: console=ttyAM1,115200
```

6.5.4. Initializing Linux Boot Options

To clear the current Linux option boot option settings and initialize them to the default state, enter the clearenv command.

```
hermit> clearenv
```

Caution

Rewriting the bootloader might cause a collapse of the Linux boot option region and prevent a normal activation. If this happens, it is required to execute the clearenv command to initialize the Linux boot option region.

6.5.5. Examples of Linux Boot Options

Examples of Linux boot option settings are shown below:

ex.1) An example to display the Linux boot log on the serial port 1 (ttyAM0) using the serial console

```
hermit> setenv console=ttyAM0,115200
```

ex.2) An example not to display the Linux boot log.

```
hermit> setenv console=null
```

7. Creating an Image With atmark-dist

This chapter describes how to create kernel and userland images with the atmark-dist. For information about how to use the atmark-dist, refer to atmark-dist Developers Guide.

Caution

Development with the atmark-dist involves the creation and allocating of the basic libraries applications and the system configuration files. All files are created and allocated under the atmark-dist directory. To avoid any damages to the operating system on the work PC due to wrong operations, be sure to carry out all operations with a general user, not a root user.

7.1. Extracting a Source Codes Archive

There is a source code archive named atmark-dist-YYYYMMDD.tar.gz in the CD-ROM (*source/dist directory*). Extract this file to an appropriate directory. Here, we extract it to the user's home directory (~/) .

```
[PC ~]$ tar zxvf atmark-dist.tar.gz
```

Next, extract the Linux kernel source code and create a symbolic link named linux-2.6x in the atmark-dist directory. The kernel source code file named linux-2.6.x-a9-x.tar.gz is contained in the CD-ROM (*source/kernel directory*).

```
[PC ~]$ tar zxvf linux-2.6.x-a9-x.tar.gz
  :
  :
[PC ~]$ cd atmark-dist
[PC ~/atmark-dist]$ ln -s ../linux-2.6.x-a9-x ./linux-2.6.x
```

7.2. Configuration

Let's configure the dist for the target board. Start configuration by entering the command as shown in the following example.

```
[PC ~/atmark-dist]$ make config
```

You will be prompted to enter the vendor name of the board to be used. Enter AtmarkTechno.

```
[PC ~/atmark-dist]$ make config
config/mkconfig > config.in
#
# Using defaults found in .config
#
*
* Vendor/Product Selection
*
*
* Select the Vendor you wish to target
*
Vendor (3com, ADI, Akizuki, Apple, Arcturus, Arnewsh, AtmarkTechno, Atmel, Avnet, Cirrus, Cogent, Conexant, Cwlinux, CyberGuard, Cytek, Exys, Feith, Future, GDB, Hitachi, Imt, Insight, Intel, KendinMicrel, LEOX, Mecel, Midas, Motorola, NEC, NetSilicon, Netburner, Nintendo, OPENcores, Promise, SNEHA, SSV, SWARM, Samsung, SecureEdge, Signal, SnapGear, Soekris, Sony, StrawberryLinux, TI, TeleIP, Triscend, Via, Weiss, Xilinx, senTec) [SnapGear] (NEW) AtmarkTechno
```

Then, you will be prompted to enter the board name. Enter Armadillo-210.Base.

```
*
* Select the Product you wish to target
*
AtmarkTechno Products (Armadillo, Armadillo-210.Base, Armadillo-210.Recover, Armadillo-9, Armadillo-9.PCMCIA, Armadillo-J.Base, Armadillo-J.Jffs2, Armadillo-J.Recover, SUZAKU, SUZAKU-UQ-XUP) [Armadillo] (NEW) Armadillo-210.Base
```

Specify the C library to be used. Supported libraries differ depending on the board being used. Select None for Armadillo-210.

```
*
* Kernel/Library/Defaults Selection
*
*
* Kernel is linux-2.4.x
*
Libc Version (None, glibc, uC-libc, uClibc) [uClibc] (NEW) None
```

You will be asked whether to set default settings. Select y (Yes).

```
Default all settings (lose changes) (CONFIG_DEFAULTS_OVERRIDE) [N/y/?] (NEW) y
```

Select n (No) for the last three questions.

```
Customize Kernel Settings (CONFIG_DEFAULTS_KERNEL) [N/y/?] n
Customize Vendor/User Settings (CONFIG_DEFAULTS_VENDOR) [N/y/?] n
Update Default Vendor Settings (CONFIG_DEFAULTS_VENDOR_UPDATE) [N/y/?] n
```

Once all questions are answered, configuration of the build system is initiated. You will return to the prompt when configuration is completed.

7.3. Build

Enter the following command to carry out the build.

```
[PC ~/atmark-dist]$ make dep all
```

Depending on the version of dist, build may stop during make processing and a message prompting to input to the undefined parameters appear. Generally, just hit the return key to proceed with operation, leaving default settings as they are.

If the build process is successfully completed, the kernel image linux.bin.gz and the userland image romfs.img.gz are created in the atmark-dist/image directory. For more information about how to rewrite the created images to the Armadillo-210, refer to Section 5, “Rewriting Flash Memory”.

8. Memory Maps

Table 8-1 Memory Map (Flash Memory)

Address	Region	Size	Description
0x60000000 0x6000ffff	bootloader	64KB	Hermit bootloader image *loader-armadillo210.bin*
0x60010000 0x6017ffff	kernel	approx. 1.44MB	Linux kernel image *linux.bin.gz* (Uncompressed and gz compressed images supported.)
0x60180000 0x603effff	userland	approx. 2.44MB	Userland image *romfs.img* (Uncompressed and gz compressed images supported.)
0x603f0000 0x603fffff	config	64KB	Configuration region

* Only the kernel and userland are extracted and copied to RAM before linux is booted.

Table 8-2 Memory Map (RAM)

Address	Content	File System	Description
0xc0018000	kernel	—	Extracted and copied from the Flash memory before linux is booted.
0xc0800000	userland	EXT2	Extracted and copied from the Flash memory before linux is booted.

9. Device Driver Specifications

9.1. GPIO Port

The parameters of the device node corresponding to the GPIO port are shown in Table 9-1.

Table 9-1 GPIO Node

Type	Major Number	Minor Number	Node Name (/dev/xxx)
Character Device	10	185	gpio

The Armadillo-210 GPIO can be controlled directly with the ioctl command.

The file descriptor of the device file is specified to the first argument.
The command to control the GPIO is specified to the second argument.

Table 9-2 GPIO Operation Command

Command	Description	Type of the third argument
PARAM_SET	This command sets the GPIO state by the third argument	struct gpio_param
PARAM_GET	This command obtains the GPIO state by the third argument	struct gpio_param
INTERRUPT_WAIT	This command WAITs the GPIO interruption by the third argument	struct wait_param

The structure *struct gpio_param* and *struct wait_param* that are defined in the (kernel source) /include/asm-arm/arch-ep93xx/armadillo2x0_gpio.h is used for the third argument. The *struct gpio_param* is defined as a single direction list. So, if two or more GPIO are controlled at the same time, use the *next* member. Furthermore, specify *0(NULL)* to the last *next* member in the list.

For more information about how to use the GPIO device driver, refer to the source code of the sample GPIO control application (atmark-dist/vendors/AtmarkTechno/Armadillo-210.Common/gpiod).

9.2. LED

The parameters of the device node corresponding to the LED are shown in Table 9-3.

Table 9-3 LED Node

Type	Major Number	Minor Number	Node Name (/dev/xxx)
Character Device	10	215	led

The LEDs on the Armadillo-210 can be controlled directly with the ioctl command.

The file descriptor of the device file is specified to the first argument.
The command to control the LED is specified to the second argument.

Table 9-4 LED Control Command

Command	Description	Type of the third argument
LED_RED_ON	This command lights on LED (RED).	None
LED_RED_OFF	This command lights off LED (RED).	None
LED_RED_STATUS	This command obtains LED (RED) state.	Buffer to store the state (Min. 1 Byte)
LED_GREEN_ON	This command lights on LED (GREEN).	None
LED_GREEN_OFF	This command lights off LED (GREEN).	None
LED_GREEN_STATUS	This command obtains LED (GREEN) state.	Buffer to store the state (Min. 1 Byte)

For information about how to use the LED device driver, refer to the source code of the sample LED control application (atmark-dist/vendors/AtmarkTechno/Armadillo-210.Common/ledctrl).

9.3. Onboard Flash Memory

The onboard Flash memory is handled by region unit as a Memory Technology Device (MTD). For information on the onboard Flash memory region, refer to Section 8, "Memory Maps".

The parameters of the device node corresponding to the regions are shown in Table 9-5.

Table 9-5 MTD Nodes

Type	Major Number	Minor Number	Node Name (/dev/xxx)	Device Name
Character Device	90	0	mtd0	bootloader
		1	mtdr0	bootloader (read only)
		2	mtd1	kernel
		3	mtdr1	kernel (read only)
		4	mtd2	userland
		5	mtdr2	userland (read only)
		6	mtd3	config
		7	mtdr3	config (read only)
Block Device	31	0	mtdblock0	bootloader
		1	mtdblock1	kernel
		2	mtdblock2	userland
		3	mtdblock3	config

10. Appendix

10.1. Building a Development Environment in Windows

A cross development environment for the Armadillo-210 can be built in Windows using coLinux(http://www.colinux.org/) that realizes the Linux environment. WindowsXP and Windows2000 are supported by coLinux.

10.1.1. Installing coLinux

1) Activate the coLinux-0.6.2.exe from the colinux directory contained in the CD-ROM.
2) Specify c:¥coLinux as the directory to which coLinux is installed, with all other settings at their defaults.

TIPS

If a directory other than c:¥coLinux is specified as the directory to which coLinux is installed, it is required to edit the default.colinux.xml file that is obtained under the procedure described in the following subsection and change the directory name appropriately.

10.1.2. Preparing Files for Building Environments

Obtain the following files from the CD-ROM (*colinux directory*) and extract them to the coLinux installation folder (c:¥coLinux).

- root_fs.zip (root file system)
- swap_device_256M.zip (swap file system)
- home_fs_2G.zip (file system mounted to /home)
- default.colinux.xml.zip (device information configuration file)

TIPS

The numbers in the file names swap_device_..., home_fs_... etc. represent the file size after extraction. Files with other sizes are also available. Choose and extract the most appropriate one from them.
Extraction may fail depending on the extraction software. We have confirmed that extraction will work properly with the WindowsXP standard function.

10.1.3. Starting coLinux

1) Activate a DOS prompt and go to the installation folder (c:¥coLinux).
2) Enter the colinux-daemon.exe -c default.colinux.xml command.
3) Login with root" privilege after colinux login: is displayed following the boot log.

10.1.4. Network Settings

coLinux has an IP addresses that is different from Windows and accesses the network through Windows, thus it is required to alter the network settings

There are two types of configurations methods, router connections and bridge connections. Here, we explain how to configure the router connections.

(WindowsXP)

1) Open Network Connections from Control Panel.
2) Right-click the externally connected network and open Properties.
3) Select the Sharing tab and enable internet connection sharing.

(Windows2000)

1) Open Network and Dial-up Connections from Control Panel.
2) Right-click the externally connected network and open Properties.
3) Select the Sharing tab and enable internet connection sharing.

Next, execute the following command on coLinux to enable the network settings.

```
colinux:~# /etc/init.d/networking restart
Reconfiguring network interfaces: done.
colinux:~#
```

Example 10-1 Network Configuration Command

The network address 192.168.0.0/24 is automatically used for router connections. Thus, if the network address for external connection uses the same address of 192.168.0.0/24, network setting is not successfully completed. In this case, change the network address for external connection.
If the network address for external connection can not be changed, refer to Section 10.1.8, “Windows Network Settings under Special Circumstances”.

10.1.5. Creating a coLinux User

Enter commands on coLinux screen as shown in Example 10-2 to create a work user. Specify a password if necessary.

Example 10-2 Adding a User "Somebody" as Work User

```
colinux:~# adduser somebody
Adding user somebody...
Adding new group somebody (1000).
Adding new user somebody (1000) with group somebody.
Creating home directory /home/somebody.
Copying files from /etc/skel
Enter new UNIX password:
```

10.1.6. File Sharing between Windows and coLinux

Example 10-3 shows a method to exchange files between coLinux and Windows using a Windows shared folder. Enter the smbmount command on coLinux screen as shown, and then enter the password for the shared folder.

Example 10-3 Windows IP Address:192.168.0.100, Shared Folder Name:shared

```
colinux:~# mkdir /mnt/smb
colinux:~# smbmount //192.168.0.100/shared /mnt/smb
212: session request to 192.168.0.100 failed (Called name not present)
212: session request to 192 failed (Called name not present)
Password:
```

If the coLinux user name differs from that on the Windows side, specify the user name as a command option. For more information, refer to HELP by entering man smbmout.
Thereafter, the data in the Windows' shared folder "shared" will be identical as that in the coLinux's /mnt/smb directory.

10.1.7. Installing the Cross Development Environment

Install the cross development environment onto coLinux according to the procedure described in Section 3, "Development Environment".

All files necessary for building the development environment can be obtained from the shared folder by accessing from coLinux as described in the previous section.

Now you have completed your preparations for developing on Windows. The instructions in the following subsections are applied only to the special circumstances.

10.1.8. Windows Network Settings under Special Circumstances

The following network configuration method is to be used when the network address for external connection is set to 192.168.0.0/24.

(WindowsXP)

Here, the bridge connections method is used.

1) Open Network Connections from Control Panel.
2) Select both the externally connected network and the network with device name TAP-Win32 adapter.
3) Select Bridge Connections from Advanced in Menu.

(Windows2000)

In Windows2000, network address other than 192.168.0.0/24 is used for a private network. Here, we use 192.168.1.0/24.

1) Open Network and Dial-up Connections from Control Panel.
2) Right-click the externally connected network and disable it.
3) Right-click the externally connected network and open Properties.
4) Select the Internet Protocol (TCP/IP) from the General tab and click the Properties button.
5) Select Use the following IP address and set 192.168.100.100 in there.
6) Open the Sharing tab and enable internet connection sharing.
7) Right-click the network connection with device name TAP-Win32 adapter and open Properties.
8) Select Internet Protocol (TCP/IP) in the General tab and click the Properties button.
9) Select Use the following IP address and set 192.168.1.1 in there.
10) Right-click the externally connected network and open Properties.
11) Select Internet Protocol (TCP/IP) in the General tab and click the Properties button.
12) Set the IP address back to its original settings.
13) Right-click the externally connected network and enable it.

10.1.9. coLinux Network Settings

While DHCP is used at the time of installation, a fixed IP address must be set in an environment where no DHCP servers are installed.

The network settings can be viewed with the ifconfig command as shown in Example 10-4.

Example 10-4 Executing ifconfig Command

```
colinux:~# ifconfig
eth0      Link encap:Ethernet  HWaddr XX:XX:XX:XX:XX:XX
          inet addr:192.168.0.151  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:189 errors:0 dropped:0 overruns:0 frame:0
          TX packets:115 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:24472 (23.8 KiB)  TX bytes:9776 (9.5 KiB)
          Interrupt:2

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

colinux:~#
```

A fixed IP address must be set when the IP address of the eth0 device is not displayed. The IP address to be set should match that of the TAP-Win32 adapter network when the router connections is selected, or should match that of the externally connected network when the bridge connections method is selected.

Here, we set network connection parameters as shown in Table 10-1.

Table 10-1 Network Settings

Parameter	Setting
IP Address	192.168.1.100
Net Mask	255.255.255.0
Gateway	192.168.1.1
DNS Server	192.168.1.1

1) Edit /etc/network/interfaces on coLinux as shown in Example 10-5.

Example 10-5 Example of Editing /etc/network/interfaces File

```
auto lo eth0
iface lo inet loopback
iface eth0 inet static
   address 192.168.1.100
   gateway 192.168.1.1
   netmask 255.255.255.0
```

2) Edit /etc/resolv.conf on coLinux as shown in Example 10-6.

Example 10-6 Example of Editing /etc/resolve.conf File

```
nameserver 192.168.1.1
```

3) Execute the following command to update the network settings with the edited content as shown in Example 10-7.

```
colinux:~# /etc/init.d/networking restart
Reconfiguring network interfaces: done.
colinux:~#
```

Example 10-7 Network Re-Setting Command

10.2. LED Status at Booting

The status of the Armadillo-210 can be checked by the LED lighting pattern at the booting of the Armadillo-210.

There are two types of status, CPU boot status and software start status. The CPU boot status represents the status to be displayed before the bootloader of the Armadillo-210 boots up.

Table 10-2 CPU Boot Status List

【Rev.A, Rev.B】

LED(GREEN)	LED(RED)	Status
Light Off	Flashing	A bootable image is not written in the Flash memory
Light On	Light On	Reset state
Light Off	Light On	Booting with CPU on-chip boot ROM

【Rev.C or later】

LED(GREEN)	LED(REN)	Status
Flashing		A bootable image is not written in the Flash memory
Light On		Reset state
Light On		Booting with CPU on-chip boot ROM

Table 10-3 Software Start Status List

LED(GREEN)	LED(RED)	Status
Light On	Light On	Hermit or Linux is booting.
Light On	Light Off	Hermit command prompt mode or booting Linux completed.

Revision History

Ver	Date	Revisions
1.0.0	2005.12.15	・Initial release
1.0.1	2006.3.7	・Corrected the description about the serial port.
1.0.2	2006.08.11	・Corrected the description in Section 10.2, "LED Status at Booting" due to the additional support of Rev.C board.

Armadillo-210 Software Manual August 11, 2006 version 1.0.2 Pre1

Atmark Techno, Inc.
AFT Building, 6F, East 2 North 5, Chuo-ku, Sapporo, Japan 060-0035
TEL:011-207-6550 FAX:011-207-6570