Software Manual

Version 1.3.1

August 11, 2006

株式会社アットマークテクノ
http://www.atmark-techno.com/

SUZAKU Official Web Site
http://suzaku.atmark-techno.com/

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. About This Manual

This manual provides the information necessary for using SUZAKU.

Outline of contents is as follows:

- Basic operations
- Writing the Flash memory

- Building Kernel and Userland
- Customizing

- Application software development

We hope the information in this document will help you get the best functionality out of the SUZAKU.

1.2. About the Fonts

In this manual fronts are used according to the following conventions.

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 assumed 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
[SUZAKU /]#	To be executed by a privileged user on SUZAKU
[SUZAKU /]$	To be executed by a general user on SUZAKU

1.4. Acknowledgements

The software used in the SUZAKU is composed from Free Software / Open Source Software. These 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 of 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 before using the SUZAKU.

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

- Serial cross cable
  A D-Sub9-pin (female-to-female) cable for cross connections

- D-Sub9-pin – 10-pin conversion cable
  The D-Sub9-pin - 10-pin conversion cable is required to connect the D-Sub9-pin to the pin header (10pins) of the board

- Development Kit CD-ROM (hereafter called CD-ROM)
  This CD-ROM contains various manuals and source code related to the SUZAKU.

- Serial console software
  Please install serial console software such as minicom or Tera Term on the work PC.

- DC3.3V Power Supply
  Please prepare a power supply that has a DC3.3V output.

2.2. Connecting Cables

Connect the serial cross cable, the power cable, and the LAN cable to the SUZAKU as shown in Figure 2-1.

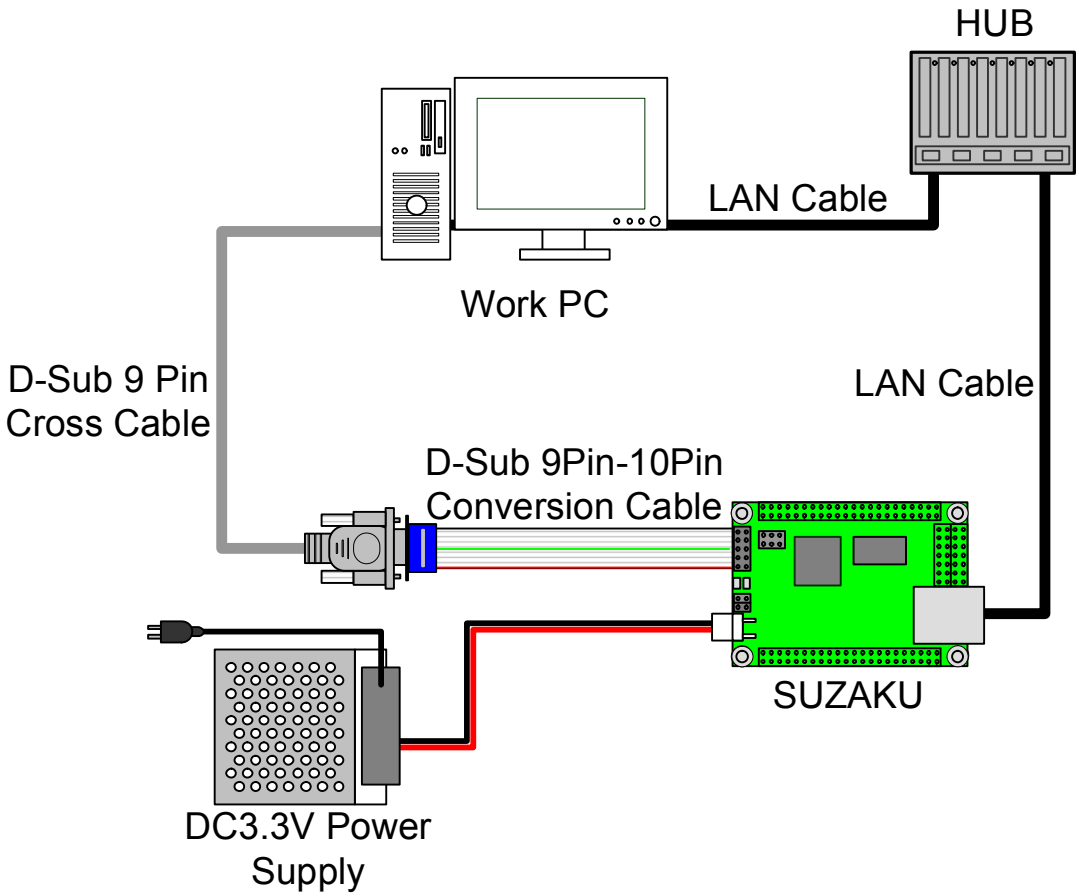

Figure 2-1 Example of SUZAKU Cable Connections

2.3. Jumper Pin Setting

The operation at booting can be set by setting jumpers on the SUZAKU. The jumper setting and its corresponding operation are shown in Table 2-1. For more information, refer to Section 4.2, Boot Mode.

Table 2-1 Jumper Setting and Operation at Booting

JP1	JP2	Operation at Booting	Boot Mode
OPEN	OPEN	Boot Linux Kernel	Auto Boot Mode
SHORT	OPEN	Boot First Stage Boot Loader	Boot Loader Mode
–	SHORT	FPGA Configuration (*1)	–

*1 For more information, refer to the SUZAKU Hardware Manual.

TIPS

Definition of “OPEN” and “SHORT”

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

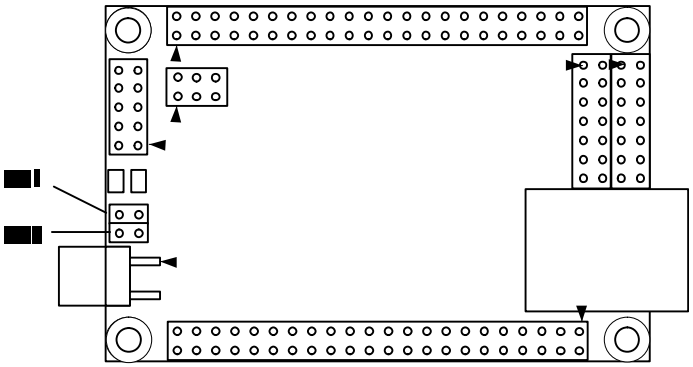

Figure 2-2 Jumper Position

3. Development Environment

Applications for the SUZAKU can be cross-developed on the work PC.

3.1. Installing Cross Development Environment Packages

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

3.1.1. SUZAKU-S

The cross development environment package for SUZAKU-S is a tar archive file compressed using gzip. Extract it from the /usr/local/microblaze-elf-tools/ directory.

```
[PC ~]$ su –
[PC ~]# mkdir -p /usr/local/microblaze-elf-tools/
[PC ~]# cd /usr/local/microblaze-elf-tools/
[PC microblaze-elf-tools]# tar zxvf microblaze-elf-tools-20060213.tar.gz
[PC microblaze-elf-tools]# ls
bin include info lib libexec microblaze share
[PC microblaze-elf-tools]# exit
[PC ~]$
```

Figure 3-1 Installing a Cross Development Package

Then add the directory that contains an executable file to the PATH environment variables so that you can easily use the Cross Development Environment. The setting method differs depending on type of shell. Refer to the manual associated to the shell you are using.

The following shows an example of bash setting. If these settings are described in the .bashrc file, these settings can be available automatically when you login the system next time.

```
[PC ~]$ export PATH=$PATH:/usr/local/microblaze-elf-tools/bin
[PC ~]$
```

Figure 3-2 Environment Variable PATH Setting Example

3.1.2. SUZAKU-V

The cross development environment packages for SUZAKU-V are contained in the CD-ROM at cross-dev/powerpc directory. Included package formats are, deb (for Debian system distribution), rpm (for Red Hat system distribution) and tgz (non-installer). Select any one that is appropriate for the operating system in use.

Table 3-1 List of Cross Development Environment Packages

Package name	Version	Description
binutils-powerpc-linux	2.15-5	The GNU Binary utilities
cpp-3.3-powerpc-linux	3.3.5-13	The GNU C preprocessor
g++-3.3-powerpc-linux	3.3.5-13	The GNU C++ compiler
gcc-3.3-powerpc-linux	3.3.5-13	The GNU C compiler
genromfs	0.5.1-3	The make equivalent for romfs file system
libc6-powerpc-cross	2.3.2.ds1-20	GNU C Library: Shared libraries and Time zone data
libc6-dev-powerpc-cross	2.3.2.ds1-20	GNU C Library: Development Libraries and Header Files
libc6-pic-powerpc-cross	2.3.2.ds1-20	GNU C Library: PIC archive library
libc6-prof-powerpc-cross	2.3.2.ds1-20	GNU C Library: Profiling Libraries
libdb1-compat-powerpc-cross	2.1.3-7	The Berkeley database routines
libgcc1-powerpc-cross	3.3.5-13	GCC support library
libstdc++5-powerpc-cross	3.3.5-13	The GNU Standard C++ Library v3
libstdc++5-3.3-dbg-powerpc-cross	3.3.5-13	The GNU Standard C++ Library v3 (debugging files)
libstdc++5-3.3-dev-powerpc-cross	3.3.5-13	The GNU Standard C++ Library v3 (development files)
libstdc++5-3.3-pic-powerpc-cross	3.3.5-13	The GNU Standard C++ Library v3 (shared library subset kit)
linux-kernel-headers-powerpc-cross	2.5.999-test7-bk-16	Linux Kernel Headers for development

An example of installation for the cross development packages is shown in Figure 3-3.

```
[PC ~]# dpkg -i binutils-powerpc-linux_2.15-5_i386.deb ←when “deb” package is used
[PC ~]# rpm -i binutils-powerpc-linux-2.15-5_i386.rpm  ←when “rpm” package is used
[PC ~]# tar zxf binutils-powerpc-linux-2.15.tgz -C /   ←when “tgz” is used
```

Figure 3-3 Installing a Cross Development Package

Specify multiple packages simultaneously as follows if the installation fails due to the dependency.

```
[PC ~]# dpkg -i xxx.deb yyy.deb zzz.deb
```

Figure 3-4 Installing multiple Packages

3.2. Packages Required for Building uClinux-dist

The packages shown in Table 3-2 are required to build uClinux-dist on the work PC. Please install them to meet the environment of the work PC in use.

Table 3-2 List of Packages Required for Building uClinux-dist

Package Name	Version	Description
libncurses-dev	5.4-4 or later	Developer’s libraries and docs for ncurses
zlibg-dev	1.2.2-4 or later	compression library - development
genromfs	0.5.1-3 or later	This is the mkfs equivalent for romfs filesystem
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

4. How to Use SUZAKU

This chapter describes how to use the SUZAKU.

4.1. Before Boot-up

Connect the serial port 1 (CON1) of the SUZAKU to the work PC using a serial cable and activate the serial console software. Then set the communication parameters as follows.

Table 4-1 Serial Communication Setting

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

4.2. Boot Mode

The SUZAKU provides three boot modes, Auto Boot Mode, Boot Loader Mode and Motorola S-Format Download Mode. The selection of the mode can be done with Jumper Pin (JP1) and BBoot Menu Screen. For the Mode Selection Jumper, refer to Section 2.3, Jumper Pin Settings.

4.2.1. Auto Boot Mode

In Auto Boot mode, Linux starts automatically. To select this mode, set the jumper pins (JP1/JP2) to OPEN and then power on the SUZAKU.

4.2.2. Boot Loader Mode

The boot loader mode is used to start the boot loader. The SUZAKU with factory default setting has a high performance boot loader + downloader called Hermit in the Flash memory. Once the Hermit is started, a prompt to accept user input is displayed. Command functions available include the memory dump function as well as the function to write the Linux image into the Flash memory.

To select the boot loader mode, set the jumper pin JP1 to SHORT and JP2 to OPEN, then power on the SUZAKU. The BBoot boot log of the first stage boot loader and the instruction to go into the BBoot Menu is displayed on the serial console.

```
BBoot v2.x (microblaze)
Press ‘z’ or ‘Z’ for BBoot Menu.
```

Figure 4-1 BBoot Startup Screen

Then press the z key for the BBoot Menu screen.

Caution

If the “Z” key is pressed after ”Press ‘z’ or ‘Z’ for BBoot Menu” is displayed, the key input may not be detected. In this case, power on the SUZAKU while holding down the z key.

The following BBoot Menu screen is displayed on the serial console.

```
Please choose one of the following and hit enter.
a:  activate second stage boot loader (default)
s:  download a s-record file
```

Figure 4-2 BBoot Menu Screen

Then, hit either the “a” key or “Enter” key here. The SUZAKU’s default boot loader “Hermit” will boot up and the following prompt is displayed.

```
Hermit-At v1.1.3 (suzaku/microblaze) compiled at 15:39:32, Aug 10 2006
hermit>
```

Figure 4-3 Hermit Startup Screen

To write data into the Flash memory using Hermit, you must once boot the SUZAKU in this mode. For more information to write the Flash memory, refer to Section 5.2, Writing Flash Memory with Hermit.

TIPS

If the BBoot Menu screen is not displayed, the Hermit Startup screen automatically appears.

4.2.3. Motorola S-Format Download Mode

This mode is used to write the boot loader on the Flash memory when it is malfunctioning. To select this mode, you can type in “s” from the BBoot Menu screen. For more information about how to go to the BBoot Menu Screen, refer to Section 4.2.2, Boot Loader Mode.

The BBoot is stored in the area called Block RAM in FPGA. Therefore, even if Linux or the area for Hermit on the Flash memory is collapsed by mistake, it still can be used as far as the FPGA is in a normal running. Moreover, since BBoot is the first program that runs after the SUZAKU is booted, it is also called the first stage boot loader.

For more information about how to write in Motorola S-Format, refer to Section 5.4, Writing the Flash Memory in Motorola S-Format.

4.3. Boot-up

Linux will boot up when the SUZAKU is powered on in the auto boot mode. The following log will be output on the serial port 1 if it boots up normally. (SUZAKU-S Starter Kit, Distribution:uClinux-dist-20051110-suzaku1)

```
Linux version 2.4.32-uc0 (atmark@build) (gcc version 3.4.1 ( Xilinx EDK 8.1 Build EDK_I.17 090206 )) #1 Mon Jul 10 19:22:07 JST
2006
On node 0 totalpages: 8192
zone(0): 8192 pages.
zone(1): 0 pages.
zone(2): 0 pages.
CPU: MICROBLAZE
Kernel command line:
Console: xmbserial on UARTLite
Calibrating delay loop... 25.60 BogoMIPS
Memory: 32MB = 32MB total
Memory: 29744KB available (957K code, 1703K data, 44K init)
Dentry cache hash table entries: 4096 (order: 3, 32768 bytes)
Inode cache hash table entries: 2048 (order: 2, 16384 bytes)
Mount cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer cache hash table entries: 1024 (order: 0, 4096 bytes)
Page-cache hash table entries: 8192 (order: 3, 32768 bytes)
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
Microblaze UARTlite serial driver version 1.00
ttyS0 at 0xffff2000 (irq = 1) is a Microblaze UARTlite
Starting kswapd
xgpio #0 at 0xFFFFA000 mapped to 0xFFFFA000
Xilinx GPIO registered
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
eth0: LAN9115 (rev 1150001) at ffe00000 IRQ 2
uclinux[mtd]: RAM probe address=0x80125a30 size=0x174000
uclinux[mtd]: root filesystem index=0
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 4096)
VFS: Mounted root (romfs filesystem) readonly.
Freeing init memory: 44K
Mounting proc:
Mounting var:
Populating /var:
Running local start scripts.
Setting hostname:
Setting up interface lo:
Mounting /etc/dhcpc:
Starting DHCP client:
Starting inetd:
Starting thttpd:

SUZAKU-S.STARTER-KIT login:
```

Figure 4-4 Boot Log

Both the login user and the default password are “root”.

Table 4-2 User Name and Password for Console Login

User name	Password	Privilege
root	root	Privileged user

4.4. Structure of a Directory

The structure of a directory is shown in Table 4-3.

Table 4-3 Structure of a Directory

Directory name	Description
/bin	Applications
/dev	Device nodes
/etc	System settings
—/etc/config	flatfsd settings
/lib	Common libraries
/mnt	Mount points
/proc	Process information
/sbin	System management commands
/usr	Common user data
/home	User home directory
/home/httpd	Web Server home directory
/tmp	Temporary backup
/var	Modified data

4.5. Shutdown

Turn the SUZAKU off for shutdown.

4.6. Network Setting

The SUZAKU is predefined to obtain an IP address via DHCP. To change the network settings, you must re-create an image. For details, refer to Section 7, Creating an Image in uClinux-dist and uClinux-dist Developers Guide.

4.6.1. Confirming Network Settings

You can view network settings with the ifconfig command. For more information, refer to the “man” of the “ifconfig” command. The following screen appears if an IP address has been successfully obtained via DHCP.

```
# ifconfig
eth0      Link encap:Ethernet  HWaddr XX:XX:XX:XX:XX:XX
          inet addr:192.168.1.xx  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
                Interrupt:2 Base address:0x300

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
```

Figure 4-5 ifconfig Execution Example

4.6.2. Using a Fixed IP Address

First of all, remove “dhcpcd-new” (2.0/2.4).

```
[PC ~/uClinux-dist]$ make menuconfig
Main Menu
   Kernel/Library/Defaults Selection     --->
      [*] Customize Vendor/User Settings

Main Menu
   Network Applications     --->
      [ ] dhcpcd-new (2.0/2.4)                    ← Remove a check.
```

Figure 4-6 Removing “dhcpcd –new”

Next, set a fixed IP address by editing the “vendors/AtmarkTechno/SUZAKU-S/etc/rc/ifconfig” file. Specify the IP address to be set in the IP_ADDRESS described at upper part of the ifconfig file.

```
[PC ~/uClinux-dsit]$ vi vendors/AtmarkTechno/SUZAKU-S/etc/rc/ifconfig
#!/bin/sh

IP_ADDRESS=192.168.10.54                    ← Rewrite IP Address

PATH=/bin:/sbin:/usr/bin:/usr/sbin

echo “Setting up interface lo: “
ifconfig lo up 127.0.0.1

echo “Setting up interface eth0: “
ifconfig eth0 $IP_ADDRESS
```

Figure 4-7 Editing a ifconfig” File

Recreate an image file and rewrite the Flash memory in the end.

4.6.3. Using DHCP

To return to a DHCP setting after a fixed IP address has been changed, add a DHCP client daemon (dhcpcd-new) as follows. Then, recreate an image file and rewrite the Flash memory.

```
[PC ~/uClinux-dist]$ make menuconfig
Main Menu
   Kernel/Library/Defaults Selection     --->
      [*] Customize Vendor/User Settings

Main Menu
   Network Applications     --->
      [*] dhcpcd-new (2.0/2.4)                  ← Add a check.
```

Figure 4-8 Adding ”dhcpcd –new”

4.7. telnet Login

You can log into the system via telnet with the following user name and password.

Table 4-4 User Name and Password for telnet Login

User name	Password
root	root

4.8. File Transfer

You can transfer files with ftp by logging into the system with the following user name and password. The home directory is “/”. To write data, go to the “/var/tmp” directory.

Table 4-5 User Name and Password for ftp

User name	Password
root	root

4.9. Web Server

A small HTTP server named thttpd is running. The SUZAKU can be browsed from a Web server. The data directory is “/home/httpd”. The URL is “http://(IP address for SUZAKU)/”. (ex. http://192.168.1.10/)

Figure 4-9 Web Server

5. Rewriting the Flash Memory

The functionality of the SUZAKU can be modified by rewriting the Flash memory. This chapter focuses on how to rewrite the Flash memory.

Caution

If the downloading of an image fails for any reason, the SUZAKU may not boot normally. Be careful about the following points while executing a writing.

- Do not power off the SUZAKU.
- Do not disconnect the serial cable connecting the SUZAKU to a development PC.

5.1. Specifying a Memory Region

You can specify by region name the address to which data is written into the Flash memory. There are three region names. Details of these regions and associated images are explained below.

- fpga
  This region stores the configuration data for FPGA.

- bootloader
  This region stores the software image called bootloader that is first executed after power is turned on. The bootloader provides the function to rewrite the Flash Memory via a serial, boot the operating system and so on

- image
  This region stores the Linux kernel and Userland images. The kernel stored in this region is started with the bootloader. This region stores the system image including each application.

The CD-ROM contains image files for each region in the image directory.

Table 5-1 Image File Name for Each Region

Region	File name	Remarks
fpga	fpga-sz###.bin	###: Model number
bootloader	loader-suzaku-microblaze-v#.#.#.bin loader-suzaku-powerpc-v#.#.#.bin	A file differs depending on the CPU in use. ###: Version number
image	image-sz###.bin	###: Model number

In the following subsections, the file name for each region is abbreviated as fpga.bin, loader-suzaku.bin and image.bin for simplification.

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

5.2. Rewriting a Flash Memory Using Hermit

Rewrite the Flash memory with Hermit according to the following procedure. As data is transferred from the work PC to the SUZAKU, this operation is also referred to as “download”.
If Hermit is not installed, first install it according to the procedure described in Section 10.2, Installing the Downloader (Hermit).

5.2.1. Booting in Boot Loader Mode

Connect the work PC to the serial port 1 (CON1) of the SUZAKU with a serial cable and set the jumper pins of the SUZAKU as follows.

- JP1 : SHORT
- JP2 : OPEN

Go to the boot loader mode from the BBoot Menu screen after power-on. For more information, refer to Section 4.2.2, Boot loader Mode.

5.2.2. Downloading

The following downloading procedure differs depending on the operating system on the work PC.

Caution
If the serial communication software such as Minicom uses the serial port, the Hermit cannot use that port. As a result, downloading always fails. Be sure to disable the serial communication software before starting Hermit.

Caution
If a wrong image is written in the boot loader region, you cannot boot from the onboard Flash memory. Restore the boot loader according to the procedure described in Rewriting the Flash Memory in Motorola S-Format.

1) Linux

Activate a terminal on the work PC that runs Linux and enter the hermit command specifying the image file and region.
In the following example, the uClinux image (image.bin) and the Hermit image (loader-suzaku.bin) are downloaded. Specify a file name with i-option and a region name with r-option.

```
[PC ~]$ hermit download -i fpga.bin -r fpga --force-locked
[PC ~]$ hermit download -i loader-suzaku.bin -r bootloader --force-locked
[PC ~]$ hermit download -i image.bin -r image
```

Figure 5-1 hermit Command Input Example

If the serial port used for the work PC is not “ttys0”, add the option “—port ‘port name’”.

TIPS

The —force-locked option must be added when writing the FPGA and the boot loader area. If it is not specified, a warning message appears and writing to the boot loader area is not executed.

When rewriting is completed, set JP1 and JP2 to OPEN and reboot the SUZAKU to load the new write image.

2) Windows

Activate Hermit-At WIN32 (hermit.exe) in the folder from which the file was extracted based on the procedure described in Section 10.2, Installing Downloader (Hermit)

The Download Screen appears when the Download button is clicked.

On the Serial Port field, set the serial port connecting to the SUZAKU.
Specify write image file in the Image field. It can also be specified from the file dialog.
On the Region field, specify write region or address.

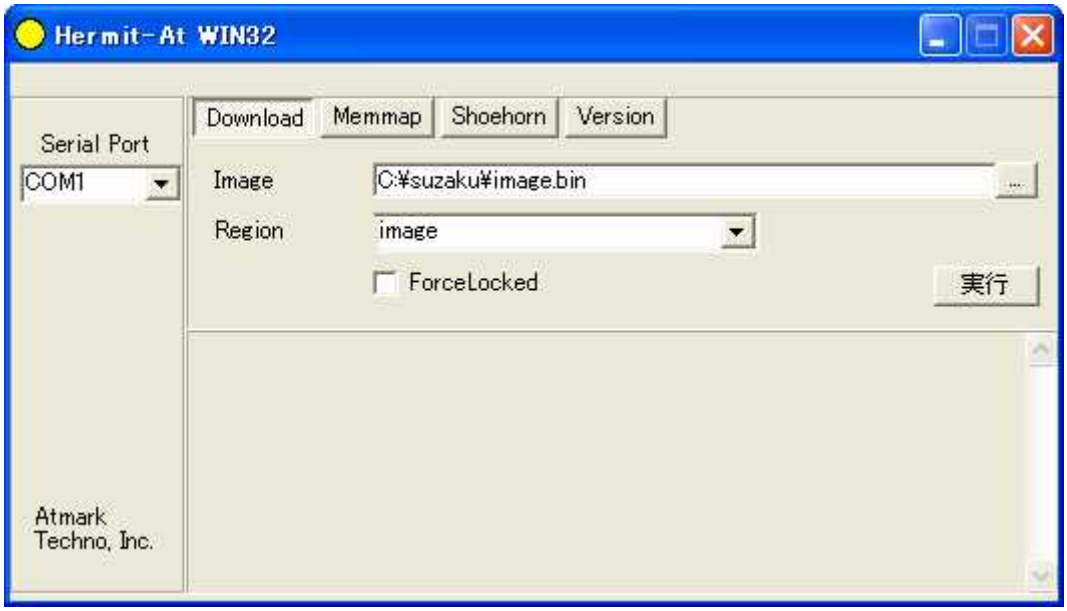

Figure 5-2 Download Screen

Click the Execution button to start downloading of the Flash memory. During downloading, the status of progress is shown in Figure 5-3. The dialog is closed automatically when downloading is finished and the Download is completed screen appears as shown in Figure 5-4.

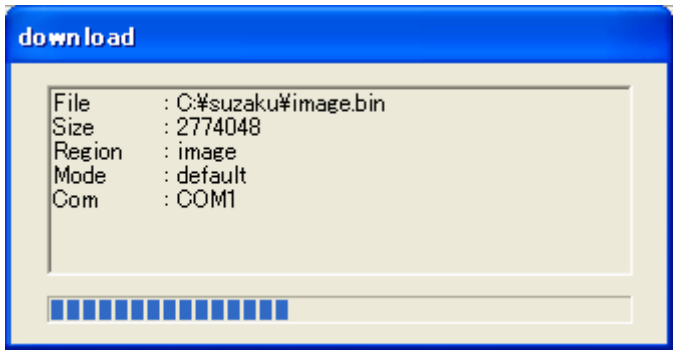

Figure 5-3 Status of Downloading dialog

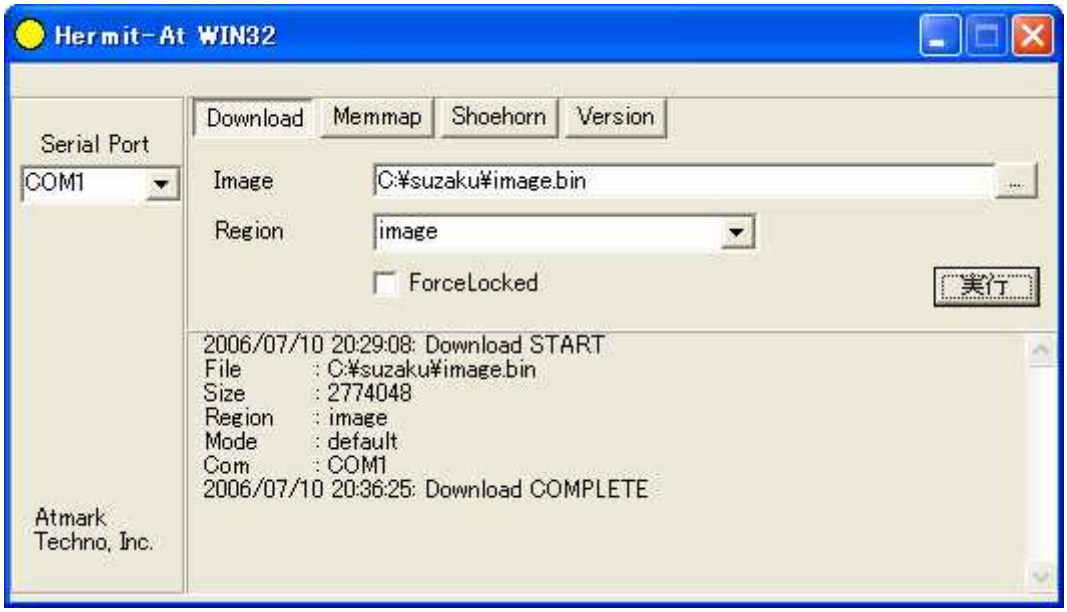

Figure 5-4 Download is Completed screen

TIPS

The check box “ForceLocked” must be checked to rewrite the FPGA and boot loader region. Otherwise, a warning message appears and writing to the boot loader region is not executed.

When writing is completed, set JP1 and JP2 to OPEN and reboot the SUZAKU to load the new write image.

5.3. Writing into the Flash Memory with netflash

Another method to write into the Flash memory is the use of the “netflash” user application. This section focuses on how to write into the Flash memory with netflash.

The netflash obtains an image file from the HTTP or FTP Server and then writes it into the Flash memory. First, place the image into the HTTP or FTP Server.

A command example to change the image (image.bin) on the SUZAKU is shown in Figure 5-5.

```
[SUZAKU ]# netflash -r image http://local.server.name/suzaku/image.bin
```

Figure 5-5 netflash Execution Example

Specify a region with the -r /dev/flash/image option based on the following table.

Table 5-2 Flash Device Name for Region

Region	Flash Device Name
fpga	/dev/flash/fpga
image	/dev/flash/image
config	/dev/flash/config

The netflash Help is available. You can view it with the following command.

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

Figure 5-6 netflash Help Command

Caution

The current version of SZ130-U00 does not support “netflash” in the distribution (uClinux-dis-20051110-suzaku2). Please be patient with us to wait for the update.

5.4. Writing into the Flash Memory in Motorola S-Format

Downloading in the Motorola S-Format is the final method to update the SUZAKU Flash memory by software. It can be executed without booting Hermit or Linux. This section focuses on how to download the Motorola S-Format Hermit to the SUZAKU with minicom.
The CD-ROM contains Motorola S-Format Hermit in the bootloader/s-record directory under the name of loader-suzaku-microblaze-v1.x.x-4M.srec (SZ010-U00), loader-suzaku-microblaze-v1.x.x-8M.srec (SZ030-U00, SZ130-U00) and loader-suzaku-powerpc-v1.x.x-8M.srec (SZ310-U00). Assuming that this file is copied into the home directory, we proceed with taking a 4MB Flash memory as an example.

Caution

The address where Hermit is written differs depending on the size of the Flash memory. In the Motorola S-Format, the writing address is written in the file. Thus, in the same SUZAKU-S, files exist separately depending on the size of the Flash memory.

First, connect the SUZAKU to the work PC with a serial cross cable and then activate “minicom” on the PC. Then short the boot mode jumper on the SUZAKU, power on the SUZAKU and go to the BBoot menu screen. The Boot Mode Selection Screen should then appear on the minicom screen.
Enter “s” on the BBoot Menu screen. The following message appears.

```
Start sending S-Record!
```

Figure 5-7 Start Downloading in Motorola S-Format Screen

Then, the SUZAKU waits for Motorola S-Format files.

Select “Send files” from the minicom Menu screen. Then select “ASCII” from the Upload submenu and go to the File Selection screen. Then, select “bootloader-suzaku.srec” from the File Selection screen to initiate file transfer. During file transfer is being executed, a small screen appears in minicom. Once file transfer is completed, the following screen is displayed,

```
Erasing SPI ...
Programming SPI ...
done.
Reboot.
```

Figure 5-8 Terminate Motorola S-Format Downloading Screen

The SUZAKU will reboot. Then select the boot loader mode to boot up Hermit.

6. Boot Loader

This chapter focuses on the SUZAKU boot loader and the boot sequence.

6.1. Types of Boot Loaders

The SUZAKU provides the following two types of boot loaders:

Table 6-1 Boot Loaders

Boot Loader Name	Description
BBoot	First Stage Boot Loader
Hermit	Second Stage Boot Loader

6.2. Boot Sequence

6.2.1. First Stage (BBoot)

When the SUZAKU is reset, the program counter is initialized to “0”. The FPGA internal memory is mapped to the address “0” and the first stage boot loader called BBoot is stored there.
The BBoot provides the function to boot up the second stage boot loader and download a Motorola S-Format file to the Flash memory. Because the FPGA internal memory is extremely expensive resource, the BBoot is designed to be as a small program as possible and the general high performance boot loader is placed in the Flash memory.
The BBoot checks status of the SUZAKU’s boot mode selection jumpers and displays the result on the BBoot Menu screen so that a user can determine whether to download the Motorola S-Format or boot up the second stage boot loader.

6.2.2. Second Stage (Hermit)

The SUZAKU uses “Hermit” as the second stage boot loader. The Hermit is a high performance boot loader and downloader. It is also used by the ARM CPU board Armadillo.
The main role of the Hermit is to boot the kernel. The Hermit checks the boot mode selection jumper settings and determines whether to boot the kernel or boot the SUZAKU in boot loader mode. In auto boot mode, the Hermit copies the kernel image from the Flash memory to the SDARM and gives control to the kernel.

When Hermit is set to boot up in boot loader mode (this mode can be selected from the BBoot Menu screen), the Hermit writes the uClinux image to the Flash memory with the proprietary binary transfer method.

6.2.3. Third Stage (Kernel)

When the kernel is given control from the boot loader, it performs system initialization. Most of system initializations are the same as those employed by the general Linux, including initialization of a timer necessary for scheduling, initialization of an interrupt vector and initialization of a valid RAM area by the memory management subsystem. The kernel mounts the root file system and gives control to the userland at the end.

As a unique function to uClinux, it can integrate the root file system at the last position of the kernel image. In the SUZAKU, to suppress the utilization of a ROM, the root file system is integrated to rewrite the BSS section of the kernel. The kernel initializes the BSS section by 0 when it finds the image of the root file system at initialization

6.2.4. Fourth Stage (Userland)

The kernel first executes the /sbin/init by default. The /sbin/init will boot up “getty” for login from the serial console, system settings and daemons according to the /etc/inittab. While the implementation of the /sbin/init in use is different from the desktop or server distribution, the basic operation to execute the shell script in order is the same.

7. Creating an Image with uClinux-dist

This chapter focuses on how to create kernel and userland images with uClinux-dist. For details, refer to uClinux-dist Developers Guide.

Caution

Development with uClinux-dist involves creation and handling of basic libraries, applications and system configuration files. While all files are created and handled under the uClinux–dist directory, make sure that all operations are performed under a general user privilege, not a root user privilege so as to avoid any damage to the operating system of the work PC that may result from a wrong operation.

7.1. Extracting a Source Code Archive

The CD-ROM provides a source code archive named uClinux-dist-YYYYMMDD-suzakuX.tar.gz in the dist directory. Extract this file to an appropriate directory. Here, we are to extract it to the user's home directory (~/).

```
[PC ~]$ tar zxvf uClinux-dist-YYYYMMDD-suzakuX.tar.gz
```

Figure 7-1 Extracting a dist Archive

7.2. Configuration

First, we carry out the configuration of the dist for the target board. Start configuration by entering the command as shown in the following example.

```
[PC ~/uClinux-dist-YYYYMMDD-suzakuX]$ make config
```

Figure 7-2 dist Configuration

You will be prompted to choose the vendor name of the board. Enter **AtmarkTechno**.

```
*
* 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, EMAC, EPSD, Exys, Feith, Future, GDB,
Hitachi, Imt, Insight, Intel, KendinMicrel, LEOX, Mecel, Midas, Motorola, NEC, NetSilicon,
Netburner, Nintendo, OPENcores, OpenGear, Philips, Promise, SNEHA, SSV, SWARM,
Samsung, SecureEdge, Signal, SnapGear, Soekris, Sony, StrawberryLinux, TI, TeleIP,
Triscend, Triscend, Via, Weiss, Xilinx, senTec) [SnapGear] (NEW) AtmarkTechno
```

Figure 7-3 Selection of a Vendor

You will then be prompted to choose the board name. Enter **SUZAKU-S** for SUZAKU-S series, **SUZAKU-S.STARTER-KIT** for the SUZAKU-S starter kit or **SUZAKU-V** for the SUZAKU-V series.

```
*
* Select the Product you wish to target
*
AtmarkTechno Products (SUZAKU-S, SUZAKU-S.STARTER-KIT, SUZAKU-UQ-XIP, SUZAKU-V) [SUZAKU-S] (NEW) SUZAKU-S.STARTER-KIT
```

Figure 7-4 Selecting a Product

Specify the C library to be used. Supported libraries differ depending on the board to be used. For SUZAKU, select **uClibc**.

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

Figure 7-5 Selecting a Library

You will be prompted to select default settings or not. Select **y** (Yes).

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

Figure 7-6 Selecting Default Settings

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
```

Figure 7-7 Selection of Customize and Update

When all questions have been answered, the configuration of the build system will start. Once the configuration has been completed, you will return to the prompt screen.

7.3. Build

Enter the following command to carry out the build.

```
[PC ~/uClinux-dist-YYYYMMDD-suzakuX]$ make dep all
```

Figure 7-8 Build

Depending on the version of dist, the make processing may pause and a prompting screen appears for unset

items. Normally, you don't need to change default settings. Just hit the return key to continue.

An image file containing both the kernel image and the userland (image.bin) image is created in the uClinux-dist/images directory once the build has completed. For information on how to write the created image to the SUZAKU, refer to Section 5, Writing into the Flash Memory.

TIPS

The dep target resolves the dependency. The build system for Linux kernel ver.2.4 or earlier needs to resolve the dependency before make. Linux ver.2.6 does not need this.

8. Flash Memory

The SUZAKU Flash memory is divided into several areas called regions. The category of these regions differs depending on products.

The SZ130-U00 Flash memory map used in the SUZAKU-S starter kit is shown in Table 8-1.

Table 8-1 Flash Memory Map (SZ130-U00, Flash: 8MB)

Address	Region		Size	Description
0x00000000 0x000FFFFF	fpga		1MB	
0x00100000 0x0011FFFF	bootloader		128KB	Hermit boot loader
0x00120000 0x0041FFFF	image	kernel	3MB	Linux kernel
0x00420000 0x007EFFFF		user	Approx. 3.81MB	Userland
0x007F0000 0x007FFFFF	config		64KB	Configuration area

The SZ010-U00 Flash memory map used in the SUZAKU-S series is shown in Table 8-2.

Table 8-2 Flash Memory Map (4MB)

Address	Region		Size	Description
0x00000000 0x0007FFFF	fpga		512KB	
0x00080000 0x0009FFFF	bootloader		128KB	Hermit boot loader
0x000A0000 0x0020FFFF	image	kernel	Approx. 1.44MB	Linux kernel
0x00210000 0x003EFFFF		user	1.875MB	Userland
0x003F0000 0x003FFFFF	config		64KB	Configuration area

The SZ030-U00 Flash memory map for the SUZAKU-S series and the SZ310-U00 Flash memory map for the SUZAKU-V series are shown in Table 8-2.

Table 8-3 Flash Memory Map (8MB)

Address	Region		Size	Description
0x00000000 0x0000FFFF	free1		64KB	
0x00010000 0x0007FFFF	free2		448KB	
0x00080000 0x000FFFFF	fpga		512KB	
0x00100000 0x0011FFFF	bootloader		128KB	Hermit bootloader
0x00120000 0x0041FFFF	image	kernel	3MB	Linux kernel
0x00420000 0x007EFFFF		user	Approx. 3.81MB	Userland
0x007F0000 0x007FFFFF	config		64KB	Configuration area

Caution

The fpga region stores the FPGA configuration data. Writing improper data to this area can cause heat generation, degradation or damage to the SUZAKU or peripheral equipment due to the malfunction of the SUZAKU. To make the SUZAKU functioning properly, it is required to re-program FPGA configuration data. For more information, refer to the Hardware Manual.

9. Flat Binary Format

The general Linux that requires MMU does not operate properly when the SUAKU is equipped with the soft-core processor Microblaze without MMU. Therefore, the SUZAKU-S with factory default setting uses uClinux.
The size of executable file represents a critical problem in the embedded systems targeted by uClinux. The ELF used by generic Linux provides a flexible format, but the size is too large. So, uClinux uses a new binary format similar to the traditional a.out format. This chapter focuses on Flat Binary Format.
We first explain the features of the Flat Binary Format and the method to compress executable files and change the stack size.

9.1. Features of Flat Binary Format

The Flat Binary Format has the following features.

- Simplicity:
  This simple design contributes to the execution speed and the size of a binary file. Although it is less flexible than ELF, it is a necessary tradeoff in embedded systems.

- Compressible
  The Flat Binary Format is a compressible format. There are two compression types. One is the entire file compression and the other is data area only compression. Since the executable file is uncompressed when it is loaded, the boot speed is slow compared to a non-compressed executable file. Since there is no difference between a compressed and a non-compressed executable file if once it is booted up, it is suitable for programs such as residential processes which do not repeat start-and-stop.

- Stack Size Filed
  The Flat Binary Format has a stack size field that can be changed without re-compiling. The CPU without a MMU has a difficulty in dynamically extending the stack area, so it provides a fixed size stack area. This field can be changed using a tool called flthdr. Moreover, it is also possible to specify the stack size at compiling.

- XIP Compatible
  The Flat Binary Format is also compatible with XIP. XIP is an abbreviation of eXecute Place (spot execution), which generally refers to the function of executable binary files to run on stored ROM without being copied to RAM.

9.2. Compressing an Executable File

The following example shows how to compress Hello World program which is created in uClinux-dis Developers Guide. The method to specify compression at compiling is also explained at the later part of this section.

Caution

With the default kernel, you can not execute a compressed Flat Binary Format (ZFLAT) file. To make the kernel compatible to ZFLAT, refer to Section 9.4 “Creating ZFLAT-based Kernel”.

9.2.1. Compressing a Compiled Binary File

This section focuses on the method to compress a compiled binary file with flthdr. "flthdr" is a program that is used to edit or view Flat Binary Format files. Since it is contained in the Microblaze tool chain with name of mb-flthdr, it is referred to as mb-flthdr hereinafter.

The following example shows an executable file created in an ordinal compiling.

```
[PC ~/hello]$ make
[PC ~/hello]$ mb-flthdr hello                                    --- (2)
hello
        Magic:          bFLT
        Rev:            4
        Build Date:     Fri Jun 30 18:33:00 2006
        Entry:          0x50
        Data Start:     0x4f40
        Data End:       0x5cf0
        BSS End:        0x7d10
        Stack Size:     0x1000
        Reloc Start:    0x5cf0
        Reloc Count:    0x51
        Flags:          0x1 (Load-to-Ram)
[PC ~/hello]$
```

Figure 9-1 Ordinary Flat Binary Format

Then, compile it with mb-flthdr.

```
[PC ~/hello]$ mb-flthdr -z hello                                 --- (1)
zflat hello --> hello
[PC ~/hello]$ mb-flthdr hello                                    --- (2)
        Magic:          bFLT
        Rev:            4
        Build Date:     Fri Jun 30 18:35:00 2006
        Entry:          0x50
        Data Start:     0x4f40
        Data End:       0x5cf0
        BSS End:        0x7d10
        Stack Size:     0x1000
        Reloc Start:    0x5cf0
        Reloc Count:    0x51
        Flags:          0x5 ( Load-to-Ram Gzip-Compressed )      --- (3)
[PC ~/hello]$
```

(1) Pass the compression option '-z' to mb-flthdr
(2) Display the header of the executable file with mb-flthdr command.
(3) The Gzip-Compressed flag can be identified.

Figure 9-2 Compressed Flat Binary Format

9.3. Specifying Stack Size

This section introduces two methods to specify stack size.

9.3.1. Changing the Stack Size of a Compiled Binary File

The stack size can be specified with mb-flthdr ”s”. The efault stack size of 4096 (0x1000) is mostly used but this may depend on architecture.

```
[PC ~/hello]$ mb-flthdr -s 8192 hello                    --- (1)
[PC ~/hello]$ mb-flthdr hello                            --- (2)
hello
      Magic:          bFLT
      Rev:            4
      Build Date:     Fri Jun 30 18:34:00 2006
      Entry:          0x50
      Data Start:     0x4f40
      Data End:       0x5cf0
      BSS End:        0x7d10
      Stack Size:     0x2000                             --- (3)
      Reloc Start:    0x5cf0
      Reloc Count:    0x51
      Flags:          0x1 (Load-to-Ram )
[PC ~/hello]$
```

(1) Pass the stack size (Decimal) with stack size change option ‘-s to mb-flthdr’
(2) Display the header of the executable file created with the mb-flthdr command
(3) The change to 8192byte can be identified.

Figure 9-3 Changing Stack Size

9.3.2. Specifying Stack Size at Compiling

The FLTFLAGS environment variable is used to specify stack size at compiling. The following shows an example of Hello World.

```
[PC ~/hello]$ make FLTFLAGS='-s 8192'                    --- (1)
[PC ~/hello]$ mb-flthdr hello                            --- (2)
hello
      Magic:          bFLT
      Rev:            4
      Build Date:     Fri Jun 30 18:34:00 2006
      Entry:          0x50
      Data Start:     0x4f40
      Data End:       0x5cf0
      BSS End:        0x7d10
      Stack Size:     0x2000                             --- (3)
      Reloc Start:    0x5cf0
      Reloc Count:    0x51
      Flags:          0x1 (Load-to-Ram )
 [PC ~/hello]$
```

- (1) Execute make by setting '-s 8192' to the FLTFLAGS environment variable.
- (2) Display the header of the executable file created with the mb-flthdr command.
- (3) The change to 8192byte can be identified.

Figure 9-4 Specifying Stack Size by FLTFLAGS

9.4. Creating ZFLAT-based Kernel

ZFLAT represents an executable file in compressed Flat Binary Format. Let' try to create a ZFLAT-based kernel here.

Execute a make menuconfig command in the uClinux-dist directory and then select Cutomize Kernel Settings from the Kernel/Library/Defaults Selection menu. Select Exit from the Main menu to save the settings and quit.

Go to General setup from the Linux Kernel Configuration Main menu. Select Enable ZFLAT support beneath Kernel support for flat binaries. Select Exit to save the settings and quit.

Execute a make clean && make dep && make command to create an image.bin file. ZFLAT files can be executed in the created kernel image..

10. Appendix

10.1. Installing the Serial Console Software (minicom)

Install the “Serial Console Software (minicom)” on the work PC.

Install the package files contained in the CD-RO. This must be done by a user with root privileges.
The supplied CD contains the package files, deb (for Debian distributions) and rpm (for Red Hat distributions). Select either one suitable for the operating system in use.

```
[PC ~]# dpkg -i minicom_2.1-4.woody.1_i386.deb      ←when deb package is used.

[PC ~]# prm -i minicom_2.1-1-rh7.3.i386.prm         ←when rpm package is used.
```

Figure 10-1 Installing minicom

Activate minicom with -s option. Specifying -s option takes you to the Configuration screen. Set serial port communication parameters.

TIPS

It appears that in many default minicom setups modem initialization is performed at start-up. You can omit the modem initialization by removing the AT command for initialization or by adding the -o option to minicom.
If you do not have read/write privilege from/to the serial port in use, minicom fails to start-up. Be sure to check read/write privilege for the serial port in use. For more information, refer to the minicom manual or operating system manual in use.

10.2. Installing the Downloader (Hermit)

Install the Downloader (hermit) on the work PC. The downloader is used to write into the SUZAKU Flash memory.

1) Linux
   Install the package files contained in the CD-ROM. This must be done by a user with root privileges.
   The CD-ROM contains the package files, deb (for Debian distributions) and rpm (for Red Hat distributions). Select either one suitable for the operating system in use. The tar.gz (non-installer, compile required) is also contained.

```
[PC ~]# dpkg -i hermit-at_x.x.x_i386.deb          ←when deb package is used.
[PC ~]# rpm -i hermit-at-x.x.x-x.i386.rpm         ←when rpm package is used.
[PC ~]$ tar zxf hermit-at-x.x.x-source.tar.gz -C /        ←when tar.gz is used.
```

Figure 10-2 installing Hermit

2) Windows
Extract Hermit-At Win32 (hermit-at-win-vx.x.xx.zip) contained in the CD-ROM to an appropriate folder.

10.3. Building a Development Environment on Windows

A cross development environment can be built on Windows by using coLinux (http://www.colinux.org/) which enables a Linux environment. Supported operating systems are WindowsXP and Windows2000.

10.3.1. Installing coLinux

1) Execute colLinux-0.6.4.exe which is contained in the colinux directory of the CD-ROM.
2) Specify c:¥coLinux as the install folder. All other settings should be defaults.

Figure 10-3 Specifying the Install Folder

TIPS

If another directory is specified as the install folder, you will need to edit the file prepared under the following procedure (default.colinux.xml) and change the directory name as appropriate.

10.3.2. Preparing Files for Building Environments

Prepare the following files from the colinux directory in the CD-ROM and extract 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. As other file sizes are also available, extract the file with appropriate size.
You may fail to extract these files when using some particular extraction programs. Extraction has been confirmed to work properly with WindowsXP standard function.

10.3.3. Start coLinux

1) Boot the DOS prompt and go to the installation folder (c:¥coLinux).
2) Enter “colinux-daemon.exe –c default default.colinux.xml” at the command line.
3) When the “colinux login:” message is displayed, login as “root”.

TIPS

If the blue screen (DRIVER_IRQL_NOT_LESS_OR_EQUAL) is displayed during execution of coLinux, try to rewrite “/noexecute=option” with “/noexecute=AlwaysOff” in c:¥boot.ini file.

10.3.4. Network Settings

As coLinux has a different IP address from Windows and accesses the network through Windows, the Windows network settings must be changed.
Two configuration methods are available, router connections and bridge connections. The following explains the procedure for setting the router connections.

(For WindowsXP)
1) Open Network Connections from the Control Panel.
2) Right-click the network connected externally and open Property.
3) Select the Advanced tab, and enable internet connection sharing.

(For Windows2000)
1) Open Network and Dial-up Connections from Control Panel.
2) Right-click the network connected externally, and open Property.
3) Select the Sharing tab, and enable internet connection sharing.

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

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

Figure 10-4 Network Configuration Command

The network address 192.168.0.0/24 is automatically used for “router connections”. So, if the same network address is used for external connection, you cannot establish a connection successfully. In this case, change the network address for external connection.
Refer to the information in Section 10.3.8, Windows Network Settings under Special Circumstances, if the network address for external connection can not be changed.

10.3.5. Creating a coLinux User

Enter the command on the coLinux screen as shown below to create a work user. Specify the password if necessary.

```
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:
```

Figure 10-5 Adding User “somebody” as a Work User

10.3.6. File Sharing between Windows and coLinux

This method allows exchanging files between coLinux and Windows using a Windows shared folder. Enter the smbmount command on coLinux screen as follows, and then enter the password for the shared folder.

```
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:
```

Figure 10-6 Windows IP Address:192.168.0.100, Shared Folder Name:shared

If the coLinux user name differs from that on Windows, specify the user name with command option. For more information, refer to the help information by executing “man smbmout”.
Thereafter, the data in the shared folder “shared” on Windows will be identical as that in the “/mnt/smb” directory on coLinux.

10.3.7. Installing Cross Development Environment

Install the cross development environment on coLinux according to the procedure described in Section 3, Preparation of Development Environment.
The necessary files for building the environment can be obtained from coLinux through the shared folder described in the previous section.

Development of the SUZAKU can be carried out on Windows. The subsequent instructions are for special circumstances.

10.3.8. Windows Network Settings under Special Circumstances

The following network setting is applied when the network address for external connection is 192.168.0.0/24.

(For WindowsXP)

The following method uses the bridge connections.

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

(For Windows2000)

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

1) Open Network and Dial-up Connections from Control Panel.
2) Right-click the network connected externally to disable it.
3) Right-click the network connected externally to open “Properties”.
4) Select 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 a 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 network connected externally 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 the original setting.
13) Right-click the network connected externally to enable it.

10.3.9. coLinux Network Setting

While DHCP is selected by default at installation, a fixed IP address must be set in an environment where DHCP server is not operating.
The network settings can be displayed with the 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:~#
```

Figure 10-7 Executing ifconfig Command

If the IP address of the eth0 device is not displayed, a fixed IP address must be set. The IP address to be set should match that of “TAP-Win32 adapter” network for router connections or that of the external network for bridge connections.
The following table shows how to change the network settings.

Table 10-1 Network Settings

Item	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) On coLinux, edit /etc/network/interfaces as follows.

```
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
```

Figure 10-8 /etc/network/interfaces File Editing Example

2) Eon coLinux, edit /etc/resolv.conf as follows.

```
nameserver 192.168.1.1
```

Figure 10-9 /etc/resolve.conf File Editing Example

3) Execute the following command to update the network settings based on the above editions.

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

Figure 10-10 Network Update Command

Revision History

Ver	Date	Revisions
1.0.0	2004/04/29	・First release
1.0.1	2004/06/04	・Changed installation of Minicom from apt to dpkg ・Corrected the misprint for tool chain name. ・generalized OTC Makefile
1.0.2	2004/12/15	・Changed company address
1.1.0	2005/01/31	・Additional description of 8MB Flash
1.2.0	2005/03/01	・Additional description of SUZAKU-V
1.3.0	2006/07/14	・Reviewed the whole composition ・Added network related information (Settings, telnet, ftp) ・Additional description of a downloader (Hermit) for Windows ・Additional description of SZ130-U00 Flash memory map ・Added a new chapter “Building Development Environment on Windows (coLinux)”
1.3.1	2006/08/11	・Additional description of fpga region in Rewriting Flash memory

SUZAKU Software Manual August 11, 2006 version 1.3.1

Atmark Techno, Inc.

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