

MS104 series

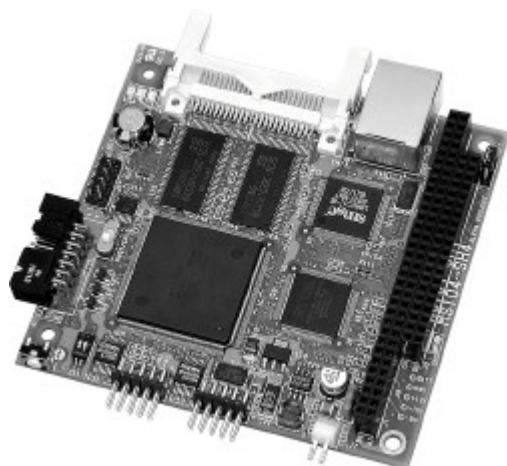
PC/104 規格準拠 SH-4 CPU BOARD

MS104-SH4

Linux 導入マニュアル

1 版

ダイジェスト版



ALPHA PROJECT co.,LTD

<http://www.apnet.co.jp>

参考資料

本製品に付属している CD-ROM には、下記の参考資料が収録されておりますので、本マニュアルと合わせてご覧ください。

SH7750 ハードウェアマニュアル	ルネサステクノロジ
PC/104 Specification	PC/104 Consortium
その他各社デバイスデータシート	

なお、Linux を初めて導入されるお客様は、巻末に掲載されております参考文献も是非ご覧ください。

参考URL

下記の URL に本製品に関連するデバイスおよび規格の情報が掲載されておりますので、参考にしてください。

株式会社ルネサステクノロジ	http://www.renesas.com/jpn/
スタンダードマイクロシステムズ株式会社	http://www.smsc.jp/
インテル株式会社	http://www.intel.co.jp/
日本アルテラ株式会社	http://www.altera.co.jp/
PC/104 Consortium	http://www.pc104.org
CompactFlash Association	http://www.compactflash.org/
eCos™ Home Page	http://sources.redhat.com/ecos/
CELinux フォーラム	http://www.celinuxforum.org/

目 次

1. はじめに	1
1.1 Linux について	1
1.2 RedBoot について	1
1.3 CE Linux について	1
1.4 GNU と FSF について	1
1.5 GPL と LGPL について	2
1.6 保証とサポート	2
2. MS104-SH4 の Linux システム	3
2.1 ブートローダ	3
2.2 Linux カーネル	3
2.3 ファイルシステム (ルートファイルシステム)	4
2.4 添付 CD の構成	5
3. apLinux の起動	6
3.1 動作環境	6
3.2 ターミナルと Ethernet の接続	7
3.3 ターミナルの設定	8
3.4 apLinux の起動と動作確認	13
3.5 Web サーバーの動作確認	17
4. RedBoot	18
4.1 RedBoot の概要	18
4.2 MS104-SH4 での機能	18
4.3 ネットワーク設定の確認・変更	19
4.4 カーネル・RAM ディスクの書き換え	23
4.5 起動スクリプトの編集	26
4.6 Linux カーネルの起動手順	28
4.7 Linux カーネルの高速起動	43
5. apLinux	45
5.1 apLinux (RAM ディスクルートファイルシステム) の概要	46
5.2 プログラム配置イメージ	47
5.3 RAM ディスクルートファイルシステムの編集	48
5.4 RAM ディスクルートファイルシステムのネットワーク設定の確認・変更	50
5.5 NFS(NetworkFileSystem)	53
5.6 apLinux (CompactFlash ルートファイルシステム) の概要	55
5.7 CompactFlash ルートファイルシステムの作成	56
5.8 CompactFlash ルートファイルシステムのネットワーク設定の確認・変更	60

6. ハードウェア依存設定**62**

6.1	レジスタ設定	62
6.2	MMU 設定	64
6.3	I/O マップ	67
6.4	IRQ 番号	67
6.5	LED 設定	68

付録 A	動作確認済み CompactFlash	69
付録 B	RedBoot コマンド一覧	70
付録 C	RedBoot コンフィグレーションデータ一覧	72
付録 D	RAM ディスクルートファイルシステム収録パッケージ一覧	74
付録 E	CompactFlash ルートファイルシステム収録パッケージ一覧	79

製品サポートのご案内

エンジニアリングサービスのご案内

1 はじめに

MS104-SH4 は、SH7750R(ルネサステクノロジ社)を搭載したボードコンピュータで、標準 OS に Linux を採用しています。Linux を採用することにより、高度なネットワーク機能やヒューマンインターフェースを備えた機器を容易に開発することができます。また、世界中のプログラマによって日々開発される膨大なオープンソースソフトウェア資産を全てロイヤリティフリーで利用することができます。

1.1 Linux について

Linux とは 1991 年に Linus Torvalds 氏によって開発された、オープンソースの UNIX 互換オペレーティングシステムです。Linux はオープンソース、ロイヤリティフリーという特性から、世界中のプログラマたちにより日々改良され、現在では Windows を脅かす存在にまで成長しました。今では大手企業のサーバーや、行政機関などにも広く採用されています。

また、Linux の特長として CPU アーキテクチャに依存しないということがあげられます。これは、GNU C コンパイラの恩恵にもよるものですが、数多くのターゲット(CPU)に移植されており、デジタル家電製品を中心に非 PC 系製品にも採用されるようになりました。

Linux は、カーネルと呼ばれる OS の核となる部分とコマンドやユーティリティなど多くのソフトウェアから構成されます。これらのソフトウェアの多くは FSF の GNU プロジェクトによるフリーソフトウェアです。

本書では MS104-SH4 ボードに付属する Linux のごく一部の機能と使い方のみを説明しています。

Linux の詳細については、一般書籍やインターネットから多くの情報を得られますので、それらを参考にしてください。

1.2 RedBoot™について

RedBoot とは RedHat™社で開発された組み込み用途向けリアルタイムオペレーティングシステム eCos™に含まれるマルチプラットフォームのブートローダ (ROM モニタ)です。RedBoot は eCos 以外の OS にも対応しており、Linux と同様にオープンソースソフトウェアとして公開されているので、さまざまなターゲットに移植されています。

現在は、RedHat 社の手を離れ、eCosCentric Limited により運営、保守が行われています。

RedBoot には、たくさんの機能及び、オプションがあります。本書では、RedBoot を使用した Linux の起動方法についてのみ説明しておりますので、詳細な情報につきましては、関連文書やインターネットサイトを参考にしてください。

1.3 CELinux について

CELinux とは、松下電器産業とソニーが中心となって設立した CE Linux フォーラムが開発した、AV 機器や携帯電話等のデジタル家電向け Linux です。

CELinux は Linux カーネル 2.4 系列ではカーネル 2.4.20 が採用された MontaVista Linux Consumer Electronics Edition version 3.x がベースになっており、起動・終了時間の短縮、応答時間などのリアルタイム性の向上、必要メモリー容量の削減などの改良が加えられています。

1.4 GNU と FSF について

GNU プロジェクトとは、UNIX ライクなソフトウェアの自由な流通を目的として 1980 年代半ばに Emacs や GCC の作者である Richard Stallman 氏により立ち上げられました。GNU プロジェクトは、非営利団体の FSF(Free Software Foundation)によって管理運営されており、世界中のボランティアと寄付から成り立っています。GNU ツールのオリジナルはこの FSF を通じて配布されています。

1.5 GPL と LGPL について

Linux や GCC、RedBoot を使用する前に GPL(GNU General Public License) と LGPL(GNU Lesser General Public License) について触れておく必要があります。GPL はソフトウェアの自由な流通 (コピーレフト) を保証するためのライセンスで、Linux や RedBoot および FSF から配布される全てのソフトウェア及び派生物に適用されています。FSF に関連しないソフトウェアでも適用しているものが数多くあります。以下に概要を記載します。

- ・ 頒布、複製、改造等が自由であること。
- ・ 頒布したソフトウェアにはソースコードを添付するか、提供手段を用意すること
- ・ GPL が適用されたソフトウェアからの派生物 (コードの一部を埋め込んだり、改変した著作物) についても GPL を適用すること
- ・ 著作権の表示義務
- ・ 無保証であること

なお、フリーソフトウェアだから必ず無償でなければいけないということではなく、頒布等のサービスに対して対価を請求することも認められています。ただし、有償で頒布されたものでも、所有者は複製を自由に頒布することができます。

GPL と LGPL の詳しい内容については、GNU プロジェクトのホームページ (<http://www.gnu.org>) をご覧ください。

本製品に付属している CD-ROM 内に GPL と LGPL の原文が収録されておりますので、ご利用になる前に必ずご一読ください。

1.6 保証とサポート

弊社では最低限の動作確認をしておりますが、Linux・RedBoot および付属ソフトウェアの性能や動作を保証するものではありません。また、これらのソフトウェアについての個別のお問い合わせ及び技術的な質問は一切受け付けておりませんのでご了承ください。なお、疑問点がある場合には、弊社ホームページに設置されております専用掲示板の利用をお勧めします。個別サポートをご希望されるお客様には、別途有償サポートプログラムをご用意しておりますので、弊社営業までご連絡ください。

Linux・RedBoot など、付属する GPL ソフトウェアのソースコードは弊社ホームページより全てダウンロードすることができます。また、これらのソフトウェアは不定期にバージョンアップをおこない、ホームページ上で公開する予定です。

2 MS104-SH4 の Linux システム

組み込み用の Linux システムは、通常、ブートローダと Linux カーネルとファイルシステムから構成されます。本製品ではブートローダに RedBoot バージョン 2.0、CELinux カーネルに、ファイルシステムには MS104-SH4 専用 RAM ディスクパッケージとコンパクトフラッシュパッケージの 2 種類を用意しています。

2.1 ブートローダ

Linux カーネルは RAM 上で動作しますが、カーネル自体は RAM 上に自身をロードする機能を有していません。そのため、Linux カーネルをロードする何らかの手段が必要となります。この手段を提供するのがブートローダです。ブートローダは CPU やメモリ、周辺デバイスの初期化も行い、OS カーネルを RAM 上に展開したあとに OS をブートさせます。

本製品で採用している RedBoot では、その他にデバッグ機能等も有しています。

RedBoot 以外の SH プロセッサ用のブートローダとして GNU/Linux on SuperH Project で開発された sh-ipl+g などがあります。

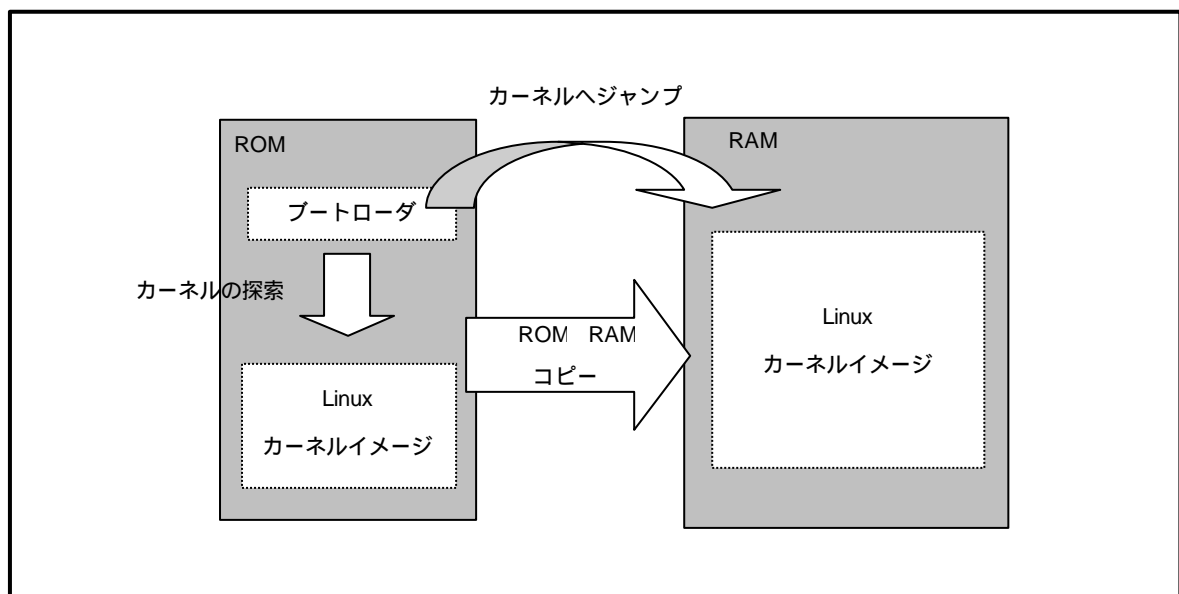


Fig 2.1-1 ブートローダの動作イメージ

2.2 Linux カーネル

Linux カーネルの機能はマルチタスク、メモリ管理、各種ファイルフォーマット、ネットワーク機能などがあり、デバイスドライバ自身もカーネルに組み込まれます。

本製品では CELinux カーネルを使用しています。本製品に添付される Linux カーネルは TCP/IP によるネットワーク機能、各種ファイルフォーマット(ext2、ext3、ramfs、cramfs、JFFS2、NFS)をサポートしています。

なお、本製品は CE Linux フォーラム(<http://www.celinuxforum.org>)の成果を中心に作成されています。

2.3 ファイルシステム（ルートファイルシステム）

Linux は、OSカーネルとファイルシステムという2つの要素から構成されます。

Linux では、全てのデータがファイルという形で管理されています。アプリケーションプログラムやデバイスドライバをはじめ、HDDやCOMポートなどの入出力デバイスもファイルとして扱われます。

Linux では全てのファイルがルートディレクトリを起点としたディレクトリ構造下に管理されており、これら全てのファイル構造のことをファイルシステムと呼びます。また、システム動作に必要なシステムファイル群のこともファイルシステムと呼びます。本書では、これらの意味を明確にするため、ファイル管理構造(ext2やext3)のことをファイルシステム、システム動作に必要なファイル群のことをルートファイルシステムと表現しています。

Linux のルートファイルシステムは、そのシステムが必要とする機能に合わせて構築する必要があります。

MS104-SH4 では、利用形態に合わせて2種類のルートファイルシステムからなる Linux パッケージを用意しています。

RAM ディスク	MS104-SH4 専用で構成されたオリジナル Linux パッケージです。サイズは約 8Mbyte で、オンボード FlashMemory に十分に収まるため、CF を使用する必要がなく、消費電力を抑えることができます。
CompactFlash	CELinux を参考にして MS104-SH4 専用に構成されたオリジナル Linux パッケージです。サイズは約 40Mbyte となっております。機能が豊富で、高機能なシステムを構築できます。

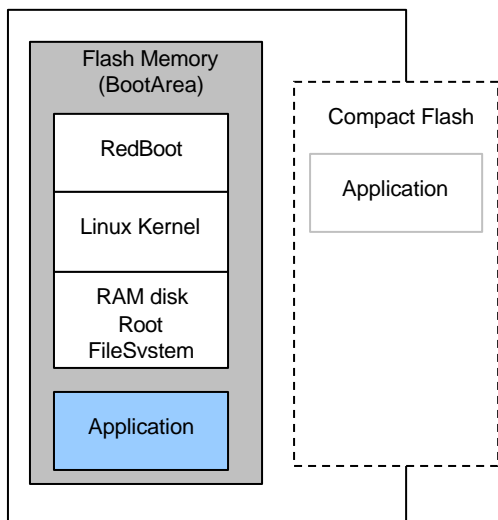


Fig 2.3-1 RAM ディスクのシステム構成例

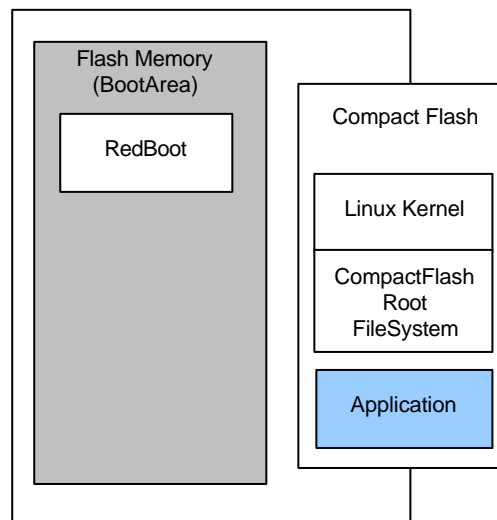


Fig 2.3-2 コンパクトフラッシュのシステム構成例

2.4 添付 CD の構成

MS104-SH4 に添付される CD-ROMには、RedBoot バイナリファイル、Linux カーネルバイナリファイル、RAM ディスク（ルートファイルシステム）、CompactFlash ルートファイルシステムが添付されます。

MS104_SH4	・添付 CD-ROM
-- CD-ROM.pdf	・本 CD-ROMについて
-- FAX 用ユーザー登録用紙.pdf	・FAX 用ユーザー登録用紙
-- GPL	・GPL 原文
-- MS104Update.pdf	・改変履歴
-- images	・CD-ROM インデックス HTML 用画像ファイル
-- index.html	・CD-ROM インデックス HTML
-- applicationnote	・各種アプリケーションノート
-- datasheet	・各種データシート
-- hdl	・ハードウェア関連
-- qdesigns 42.zip	・CPLD HDL ソースファイル
-- images	・本 CD-ROM インデックス HTML 用画像ファイル
-- index.html	・本 CD-ROM インデックス HTML
-- linux	・Linux 関連
-- ms104sh4	
-- cf-ms104sh4-x.x.tar.gz	・MS104-SH4 用 CompactFlash ルートファイルシステム
-- ramdisk-ms104sh4-x.x.gz	・MS104-SH4 用 RAM ディスクルートファイルシステム
-- vmlinuz-ms104sh4-x.x	・MS104-SH4 用 Linux カーネル
-- ms104vga	
-- mdemo	・Microwindows デモプログラム
-- ramdisk-ms104sh4-x.x-vga.gz	・MS104-VGA/LCD 対応 RAM ディスクルートファイルシステム
-- vmlinuz-ms104sh4-x.x-vga	・MS104-VGA/LCD 対応ホスト Linux カーネル
-- ms104usb	
-- ramdisk-ms104sh4-x.x-usb.gz	・MS104-USB H/S USB ホスト対応 RAM ディスクルートファイルシステム
-- vmlinuz-ms104sh4-x.x-usb	・MS104-USB H/S USB ホスト対応 Linux カーネル
-- manual	・マニュアル
-- MS104-SH4 Linuxstart.pdf	・MS104-SH4 Linux 導入マニュアル
-- MS104-SH4Hardware.pdf	・MS104-SH4 ハードウェアマニュアル
-- redboot	・RedBoot 関連
-- redboot-ms104sh4-x.x.bin	・MS104-SH4 用 RedBoot バイナリファイル
-- schematic	・回路図
-- ms104-sh4.pdf	・MS104-SH4 回路図

Fig 2.4-1 MS104-SH4 の CD-ROM 収録内容

「x.x」はバージョン番号を示します。バージョン 2.0 の場合は「2.0」になります。

3 apLinux (RAM ディスク) の起動

本章では、MS104-SH4 に搭載されているフラッシュロムから apLinux の起動する方法、および、簡単な動作確認までを説明します。

3.1 動作環境

MS104-SH4 の apLinux の起動を確認するためには、MS104-SH4 以外に以下の環境が必要です。

ホスト PC

MS104SH4 用 RedBoot/Linux では PC をコンソール端末として使用しますので、MS104-SH4 の Linux の起動を確認するためには、シリアルポートが使用可能な PC が必要となります。PC では、ハイパーターミナル・Minicom 等のターミナルソフトウェアを動作させます。

電源

MS104-SH4 本体に必要な電源は DC5V ±5% です。単体で動作させる場合には 1A 程度の電源を用意してください。

LAN

MS104-SH4 をネットワークに接続する場合は、LAN ケーブルを接続してください。直接ホスト PC と接続する際はクロスケーブル、ハブを介してネットワークに接続する際はストレートケーブルをご使用ください。

LAN ケーブルは、10/100BASE-TX 対応 (UTP カテゴリ 5) ケーブルをご利用ください。

Table 3.1-1 MS104-SH4 の推奨動作環境

使用機器等	環 境
Linux ボード	MS104-SH4
HOST PC	PC/AT 互換機
OS	Windows98 / NT4.0 / 2000 / XP or Linux (RedHatLinux9, FedoraCore1 等)
メモリ	使用 OS による
ソフトウェア	ターミナルソフト
ドライブ	CD-R 読み込み可能なドライブ
LAN ポート	10Base-T or 100Base-TX 1ポート
RS232C ケーブル	クロスケーブルを使用
シリアル変換コネクタ	MS104-SH4 付属品
LAN ケーブル	ホスト PC と接続時はクロスケーブルを使用 ハブと接続時はストレートケーブルを使用
電源	DC5V ±5% 1A 以上

3.2 ターミナルと Ethernet の接続

ホスト PC と MS104-SH4 を下図のように接続してください。

LAN ネットワークに直接接続する場合は、ネットワーク管理者と相談し、設定に注意して接続してください。

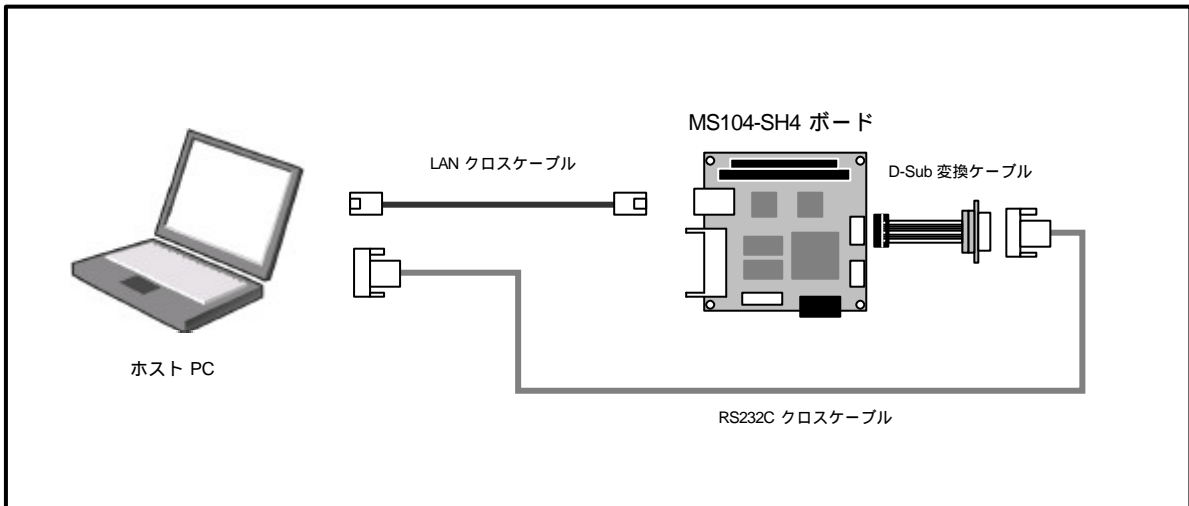


Fig 3.2-1 MS104-SH4 接続例 (PC に接続する場合)

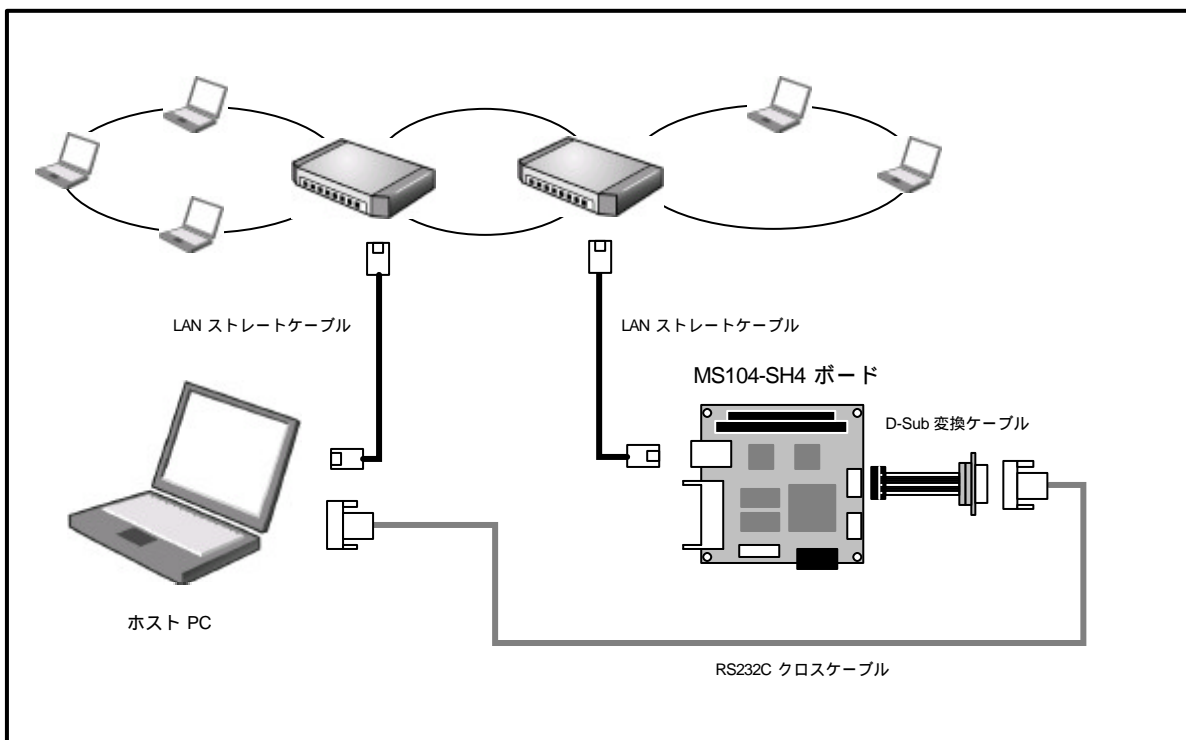


Fig 3.2-2 MS104-SH4 接続例 (HUB に接続する場合)

3.4 apLinux の起動と動作確認

『3.2 ターミナルとEthernetの接続』を参考にして、MS104-SH4 とホスト PC を接続してください。

ホスト PC のターミナルソフトを起動します。（設定は、『3.3 ターミナルの設定』を参照してください）

MS104-SH4 の電源を入れます。すると、ターミナル上に RedBoot の起動ログが表示され、5 秒後に Linux カーネルが自動起動します。（全ての起動までに約 20 秒前後かかります）

Table 3.4-1 Redboot 起動ログ

```
+Ethernet eth0: MAC address 00:0c:7b:20:**:**
IP: 192.168.1.200/255.255.255.0, Gateway: 0.0.0.0
Default server: 0.0.0.0, DNS server IP: 0.0.0.0

RedBoot(tm) bootstrap and debug environment [ROM]
Non-certified release, version v2_0 - built 22:28:09, Dec 21 2004

Platform: MS104-SH4 (SH7750R) Version x.x
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.

RAM: 0x8c000000-0x8e000000, 0x8c082f78-0x8dfbf000 available
FLASH: 0x80000000 - 0x81000000, 128 blocks of 0x00020000 bytes each.
== Executing boot script in 5.000 seconds - enter ^C to abort
RedBoot> fis load vmlinuz
RedBoot> fis load ramdisk.gz
RedBoot> exec -m 1 -f 0 -r 0x0100 -l 1 -i 0x360000 -j 0xc00000
Now booting linux kernel:
Base address 0x8c001000 Entry 0x8c210000
Cmdline : console=ttySC1,38400,N,8 mem=32M ether=8,0x300,0,0,eth0
MOUNT_RDONLY : 0x00000001
RAMDISK_FLAGS : 0x00000000
ORIG_ROOT_DEV : 0x00000100
LOADER_TYPE : 0x00000001
INITRD_START : 0x00360000
INITRD_SIZE : 0x00c00000
```

Table 3.4-2 apLinux (RAM ディスク) 起動ログ

```

Uncompressing Linux... Ok, booting the kernel.
Linux version 2.4.20-celf3 (root@hg-linux) (gcc version 3.2.3) #1 Tue Jan 11 19:
59:12 JST 2005
MS104-SH4 Setup...done
On node 0 totalpages: 8192
zone(0): 8192 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line: console=ttySC1,38400,N,8 mem=32M ether=8,0x1300,0,0,eth0
MS104SH4 IRQ setup ..<6> done .
CPU clock: 235.47MHz
Bus clock: 58.86MHz
Module clock: 58.86MHz
Interval = 147173
Calibrating delay loop... 235.11 BogoMIPS
Use 'Preset loops_per_jiffy=1175552 for preset lpi.
Memory: 18024k/32768k available (1830k kernel code, 14744k reserved, 56k data, 4
0k 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)
CPU: SH7750/SH7751/SH7750R/SH7751R/SH7760
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
PC/104 bus for I/O space : physical addr 0x18000000, Dynamic bus sizing addr 0xc
0102000, 8 Bit addr 0xc0113000, 16 Bit addr 0xc0124000
PC/104 bus for memory space :physical addr 0x18000000, 8 Bit addr 0xc1136000, 16
Bit addr 0xc0135000
Starting kswapd
Disabling the Out Of Memory Killer
VFS: Disk quotas vdfquot 6.5.1
Journaled Block Device driver loaded
Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
JFFS2 version 2.1. (C) 2001, 2002 Red Hat, Inc., designed by Axis Communications
AB.
SuperH SCI(F) driver initialized
ttySC0 at 0xffe00000 is a SCI
ttySC1 at 0xffe80000 is a SCIF
Real Time Clock Driver v1.10e
SH Real Time Clock Driver v0.01
Uniform Multi-Platform E-IDE driver Revision: 6.31
ide: Assuming 50MHz system bus speed for PIO modes; override with idebus=xx
RAMDISK driver initialized: 16 RAM disks of 12288K size 1024 blocksize
loop: loaded (max 8 devices)
smc91111.c:v1.0saa 08/22/01 by Daris A Nevil (dnevil@snmc.com)
eth0: SMC91C11xFD(r:1) at 0x1300 IRQ:8
MEM:8192b NOWAIT:0 ADDR_00:0c:7b:20:00:11
Probing for flash chips at 0x00000000:
cfi cmdset 0001: Erase suspend on write enabled
Using buffer write method
MS104-SH4: Flash at 0x00000000
6 RedBoot partitions found on MTD device MS104-SH4 FLASH
Creating 6 MTD partitions on "MS104-SH4 FLASH":
0x00000000-0x00040000 : "RedBoot"
0x00040000-0x00140000 : "vmlinuz"
0x00140000-0x00540000 : "ramdisk.az"
0x00540000-0x00fc0000 : "unallocated"
0x00fc0000-0x00fd0000 : "RedBoot config"
mtd: partition "RedBoot config" doesn't end on an erase block -- force read-only

0x00fe0000-0x01000000 : "FIS directory"
Tracer: Initialization complete
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 4096)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Compressed image found at block 0
Freeing initrd memory: 12288k freed
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 40k freed
init started: BusyBox v0.60.5 (2004.04.29-07:59+0000) multi-call binary

Alpha Project apLinux System

Mounting proc:Mounting '/' read-only: done.
Remounting '/' read/write: done.
Bringing up interface eth0: done.
Bringing up interface lo: done.
Starting klogd: done.
Starting syslogd: done.
Starting xinetd: done.
Starting tiny web server: done.

MS104SH4 login:

```

環境やバージョンによって若干異なる場合があります。

3.5 Web サーバーの動作確認

apLinux では組み込み用小型 Web サーバとして boa Web サーバがインストールされています。

boa Web サーバは apLinux 起動時に自動的に立ち上がりますので、Web ブラウザで、MS104-SH4 の IP アドレス（192.168.1.200：出荷時設定の場合）を参照すれば、MS104-SH4 に内蔵された apLinux のデモページを参照することができます。



Fig 3.5-1 MS104-SH4 デモページ

デモページが開けない場合は『ping』コマンドを実行し、正しくネットワークの設定が行われているかご確認ください。

Boa Web サーバを停止する場合は、『ps』で Boa Web サーバのプロセス ID を確認し、『kill プロセス ID』でプロセスを停止してください。

```
[MS104SH4@root]# ps
PID TTY      Uid        Size State Command
  1   root      root        1972  S    init
  2   root      root         0  S    [keventd]
  3   root      root         0  R    [ksoftirqd_CPU0]
  4   root      root         0  S    [events/0]
  5   root      root         0  S    [kswapd]
  6   root      root         0  S    [bdflush]
  7   root      root         0  S    [kupdated]
  8   root      root         0  S    [mtdblockd]
 52   root      root       1968  S    klogd
 62   root      root       1968  S    syslogd
 84   root      nobody     1880  S    /usr/sbin/boa -c /etc/boa
 97   root      root       1976  S    -sh
103   root      root       2220  R    ps
[MS104SH4@root]# kill 84
```

Boa Web サーバを再起動する際は、『/usr/sbin/boa -c /etc/boa』を実行してください。

```
[MS104SH4@root]# /usr/sbin/boa -c /etc/boa
```

4 RedBoot

4.1 RedBoot の概要

RedBoot とは Red Hat 社により開発された組み込み用ブートローダです。RedBoot は eCos の HAL (Hardware Abstraction Layer) を元に設定されており、以下の機能をサポートしています。

- ・起動スクリプト
- ・シリアル・イーサネットポートによる通信
- ・Flash Image System(FIS)をサポート
- ・HTTP/TFTPによるプログラムのダウンロードをサポート
- ・Command Line Interface(CLI)をサポート
- ・GDB をサポート
- ・XY モデムによるプログラムのダウンロードをサポート

MS104-SH4 用にカスタマイズされた RedBoot では以下の機能が追加されています。

- ・CompactFlashをサポート(活線挿抜非対応)
- ・Linux カーネルの高速起動
- ・Real Time Clock をサポート
- ・グラフィックボード (MS104-VGA/LCD) のサポート

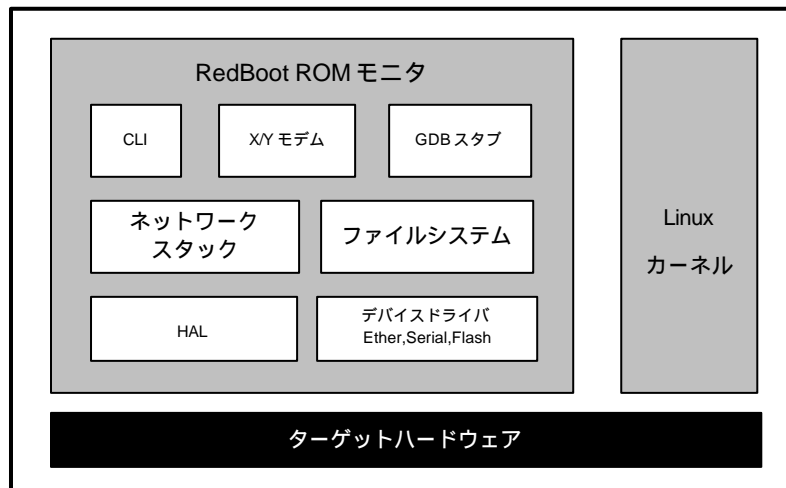


Fig 4.1-1 RedBoot アーキテクチャ

MS104-VGA/LCD はアルファプロジェクト社製グラフィック表示ボードです。

4.2 MS104-SH4 での機能

Redboot は、MS104-SH4 では次のような機能を実現しています。

Linux の起動

FlashROMの書き換え (カーネル、ファイルシステムの書き換え)
 ネットワーク経由でのカーネル、ファイルシステムのダウンロード
 シリアル経由でのカーネル、ファイルシステムのダウンロード
 CompactFlashからのカーネル、ファイルシステムのダウンロード
 デバッグ機能 (現在のところサポートしていません)

時計の設定

MACアドレスの設定

4.3 ネットワーク設定の確認・変更

Redboot は、ネットワークの設定をコマンド入力で簡単に変更することができます。ただし、これは Redboot の設定で、Linux の設定とは異なりますので注意してください。

MS104-SH4 の出荷時のネットワーク設定は次のようになっています。

Table 4.3-1 ネットワーク設定初期値

ネットワークの設定	
IP アドレス	192.168.1.200
サブネットマスク	255.255.255.0
ゲートウェイ	0.0.0.0
DNS サーバ	0.0.0.0

apLinux では Redboot のネットワーク設定と、Linux の設定は個別に設定することができますが、必要がない限り、同じ設定にしておくことをお勧めします。

ネットワーク設定の確認

MS104-SH4 の電源が OFF であることを確認し、MS104-SH4 ボードの COM2(SCIF) と Ethernet ポートをそれぞれ、ホスト PC のシリアルポートと Ethernet ポートに接続してください。

なお、ホスト PC の IP アドレスは「192.168.1.201」、MS104-SH4 の IP アドレスを「192.168.1.200」と仮定します。

MS104-SH4 の電源を ON にし、RedBoot の起動ログが表示されたら、『Ctrl+C』とタイプします。

そこで、RedBoot の起動時の表示ログによりネットワークの設定を確認することができます。

```
+Ethernet eth0: MAC address 00:0c:7b:xx:xx:xx
IP: 192.168.1.200/255.255.255.0, Gateway: 0.0.0.0 ← RedBoot ネットワーク設定
Default server: 0.0.0.0, DNS server IP: 0.0.0.0

RedBoot(tm) bootstrap and debug environment [ROM]
Non-certified release, version v2_0 - built 10:38:44, Jul 26 2003

Platform: MS104-SH4 (SH7750R) Version x.x
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.

RAM: 0x8c000000-0x8e000000, 0x8c00cc78-0x8dfdd000 available
FLASH: 0x80000000 - 0x81000000, 128 blocks of 0x00020000 bytes each.
== Executing boot script in 5.000 seconds - enter ^C to abort
^C
RedBoot>
```

また、『ip_address』コマンドでもネットワーク設定を確認することができます。

```
RedBoot> ip_address
IP: 192.168.1.200/255.255.255.0, Gateway: 0.0.0.0
Default server: 0.0.0.0, DNS server IP: 0.0.0.0
```


4.4 カーネル・RAM ディスクイメージの書き換え

RedBoot の機能により、FlashROM に格納された Linux カーネルイメージ、RAM ディスクイメージの書き換えを行うことができます。カーネル・RAM ディスクイメージは一度 RAM 上にダウンロードした後、FlashROM に保存することができます。カーネル・RAM ディスクイメージのダウンロード方法は CompactFlash、ネットワーク、シリアル の 3 つの手段があります。この章では、TFTP サーバ（ネットワーク）を使用した場合の Linux カーネル・RAM ディスクイメージの書き換え方法について説明します。

TFTP サーバの起動については『4.6 Linux カーネルの起動手順 ネットワーク (tftp/http) からの起動』をご覧ください。

書き換えを行う Linux カーネルイメージと RAM ディスクイメージはホスト PC の TFTP サーバディレクトリに『**vmlinuz**』と『**ramdisk.gz**』というファイル名で保存されていると仮定します。

MS104-SH4 の電源が OFF であることを確認し、MS104-SH4 ボードの COM2(SCIF) と Ethernet ポートをそれぞれ、ホスト PC のシリアルポートと Ethernet ポートに接続してください。

なお、ホスト PC の IP アドレスは「**192.168.1.201**」、MS104-SH4 の IP アドレスを「**192.168.1.200**」と仮定します。

MS104-SH4 の電源を ON にし、RedBoot の起動ログが表示されてから、設定時間内（デフォルト 5 秒）に『**Ctrl+C**』とタイプしてください。

```
+Ethernet eth0: MAC address 00:0c:7b:xx:xx:xx
IP: 192.168.1.1/255.255.255.0, Gateway: 0.0.0.0
Default server: 0.0.0.0, DNS server IP: 0.0.0.0

RedBoot(tm) bootstrap and debug environment [ROM]
Non-certified release, version v2_0 - built 10:38:44, Jul 26 2003

Platform: MS104-SH4 (SH7750R) Version x.x
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.

RAM: 0x8c000000-0x8e000000, 0x8c00cc78-0x8dfdd000 available
FLASH: 0x80000000 - 0x81000000, 128 blocks of 0x00020000 bytes each.
== Executing boot script in 5.000 seconds - enter ^C to abort
^C
RedBoot>
```

FlashROM の内容を確認します。『**fis list**』を実行してください。

```
RedBoot> fis list
Name          FLASH addr  Mem addr    Length      Entry point
RedBoot       0x80000000  0x80000000  0x00020000  0x00000000
RedBoot config 0x80FC0000  0x80FC0000  0x00001000  0x00000000
FIS directory 0x80FE0000  0x80FE0000  0x00020000  0x00000000
vmlinuz      0x80040000  0x8C210000  0x00100000  0x8C210000
ramdisk.gz  0x80140000  0x8C360000  0x00400000  0x8C360000

↑
RAM ディスクイメージ名

Linuxカーネルイメージ名
```

4.5 起動スクリプト

RedBoot は電源投入後に指定されたコマンドを自動実行することができます。このコマンドを起動スクリプトと呼びます。起動スクリプトは 256 文字までのコマンドを実行することができます。編集は『fconfig』コマンドを用いて行います。

Table 4.5-1 起動スクリプト初期値

	起動スクリプト	概要
1	fis load vmlinuz	FlashROMから Linux カーネルのダウンロード
2	fis load ramdisk.gz	FlashROMから RAMディスクイメージのダウンロード
3	exec -m 1 -f 0 -r 0x0100 -l 1 -i 0x360000 -j 0xc00000	Linux カーネルの起動

MS104-SH4 の電源が OFF であることを確認し、MS104-SH4 ボードの COM2(SCIF)と Ethernet ポートをそれぞれ、ホスト PC のシリアルポートと Ethernet ポートに接続してください。

MS104-SH4 の電源を ON にし、RedBoot の起動ログが表示されたら、『Ctrl+C』とタイプします。

```
+Ethernet eth0: MAC address 00:0c:7b:xx:xx:xx
IP: 192.168.1.1/255.255.255.0, Gateway: 0.0.0.0
Default server: 0.0.0.0, DNS server IP: 0.0.0.0

RedBoot(tm) bootstrap and debug environment [ROM]
Non-certified release, version v2_0 - built 10:38:44, Jul 26 2003

Platform: MS104-SH4 (SH7750R) Version x.x
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.

RAM: 0x8c000000-0x8e000000, 0x8c00cc78-0x8dfdd000 available
FLASH: 0x80000000 - 0x81000000, 128 blocks of 0x00020000 bytes each.
== Executing boot script in 5.000 seconds - enter ^C to abort
^C
RedBoot>
```

起動スクリプトの編集を行います。『fconfig』を実行してください。

```
RedBoot> fconfig
Run script at boot: true "true"または"false"を指定
```

『fconfig』コマンドを実行すると、上記のメッセージが出力されます。

ブートスクリプトを実行する場合は『true』、ブートスクリプトを実行しない場合は『false』を入力します。

true を選択した場合、ブートスクリプトの変更を行います。「>>」に続いて実行するコマンドを入力します。

```
RedBoot> fconfig
Run script at boot: true
Boot script:
.. fis load vmlinuz
.. fis load ramdisk.gz
.. exec -m 1 -f 0 -r 0x0100 -l 1 -i 0x360000 -j 0xc00000
Enter script, terminate with empty line
>> load -r -b 0x8c210000 -m tftp -h 192.168.1.201 vmlinuz
>> load -r -b 0x8c360000 -m tftp -h 192.168.1.201 ramdisk.gz
>> exec -m 1 -f 0 -r 0x0100 -l 1 -i 0x360000 -j 0xc00000
>> リターン
```

← 前回の起動スクリプト

← 新たな起動スクリプト

4.6 Linux カーネルの起動手順

MS104-SH4 は RedBoot を使用することにより、さまざまな Linux カーネルの起動方法を実現できます。本章では、各起動方法の手順を簡単に説明します。

起動方法の種類

RedBoot は、次の Linux カーネル起動方法をサポートしています。

- FlashROMからの起動
- CompactFlashからの起動
- ネットワークダウンロード (HTTP/TFTP) による起動
- シリアル (X/Modem) からの起動

自動起動について

MS104-SH4 は Linux を自動で起動することができます。MS104-SH4 の電源を ON にすると、RedBoot が立ち上がり、あらかじめ設定された時間の経過後（初期値 5 秒）に FlashROM から Linux カーネルと RAM ディスクイメージを RAM に展開し、Linux を起動させます。

RedBoot の自動起動をキャンセルするには、起動ログが表示されてから、設定時間内に『Ctrl+C』をタイプしてください。

RedBoot の起動から Linux の起動まで、約 20 秒程度かかります。

正常に起動ログが表示されない場合、MS104-SH4 の電源を OFF し、電源・シリアルケーブルの接続、ターミナルソフトの設定、RedBoot 起動スクリプトを再確認してください。

```
+Ethernet eth0: MAC address 00:0c:7b:xx:xx:xx
IP: 192.168.1.1/255.255.255.0, Gateway: 0.0.0.0
Default server: 0.0.0.0, DNS server IP: 0.0.0.0

RedBoot(tm) bootstrap and debug environment [ROM]
Non-certified release, version v2_0 - built 22:28:09, Dec 21 2004

Platform: MS104-SH4 (SH7750R) Version x.x
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.

RAM: 0x8c000000-0x8e000000, 0x8c00cc78-0x8dfdd000 available
FLASH: 0x80000000 - 0x81000000, 128 blocks of 0x00020000 bytes each.
== Executing boot script in 5.000 seconds - enter ^C to abort
RedBoot> fis load vmlinuz
RedBoot> fis load ramdisk.gz
RedBoot> exec -m 1 -f 0 -r 0x0100 -l 1 -i 0x360000 -j 0xc00000
Now booting linux kernel:
Base address 0x8c001000 Entry 0x8c210000
```

```
MS104SH4 login: root
```

```
BusyBox v0.60.5 (2003.07.03-06:22+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.
```

```
[MS104SH4@root]#
```

4.7 Linux カーネルの高速起動

RedBoot は起動後に各種デバイスを初期化を行い、Linux カーネルを起動します。Linux カーネルも RedBoot と同様に起動後各種デバイスの初期化を行います。そのため、RedBoot でのデバイスの初期化を省き、起動時間の短縮を図ります。Linux カーネルの高速起動は MS104-SH4 に搭載されているフラッシュロムからの起動のみをサポートします。

MS104-SH4 の電源が OFF であることを確認し、MS104-SH4 ボードの COM2(SCIF)と Ethernet ポートをそれぞれ、ホスト PC のシリアルポートと Ethernet ポートに接続してください。

MS104-SH4 の電源を ON にし、RedBoot の起動ログが表示されたら、『Ctrl+C』とタイプします。

```
+Ethernet eth0: MAC address 00:0c:7b:xx:xx:xx
IP: 192.168.1.1/255.255.255.0, Gateway: 0.0.0.0
Default server: 0.0.0.0, DNS server IP: 0.0.0.0

RedBoot(tm) bootstrap and debug environment [ROM]
Non-certified release, version v2_0 - built 10:38:44, Jul 26 2003

Platform: MS104-SH4 (SH7750R) Version x.x
Copyright (C) 2000, 2001, 2002, Red Hat, Inc.

RAM: 0x8c000000-0x8e000000, 0x8c00cc78-0x8dfdd000 available
FLASH: 0x80000000 - 0x81000000, 128 blocks of 0x00020000 bytes each.
== Executing boot script in 5.000 seconds - enter ^C to abort
^C
RedBoot>
```

Linux カーネル・ルートファイルシステム (RAMディスク) のフラッシュロムロード設定を行います。『fconfig』を実行してください。

```
RedBoot> fconfig
Run script at boot: false
Use BOOTP for network configuration: false
Gateway IP address: 192.168.1.254
Local IP address: 192.168.1.200
Local IP address mask: 255.255.255.0
Default server IP address: 192.168.1.1
Console baud rate: 38400
DNS server IP address:
GDB connection port: 9000
Force console for special debug messages: false
Booting Linux Kernel at Power ON: false
Load Linux Kernel & File System: true
Linux Kernel Destination address: 0x8c210000
Linux Kernel Source address: 0x80040000
Linux Kernel Size: 0x100000
RAM disk Destination address: 0x8c360000
RAM disk Source address: 0x80140000
RAM disk Size: 0x400000
```

“false” から “true” へ

Linux カーネルロード設定

ルートファイルシステム (RAMディスク) ロード設定

フラッシュロムロード設定をするには MS104-SH4 のフラッシュロムに Linux カーネルとルートファイルシステム (RAM ディスク) が格納されている必要があります。

フラッシュロムロード設定はフラッシュロムのデータ格納状況により異なるため、必ず『fis list』コマンドでフラッシュロムの内容を確認してから設定してください。上記はデフォルト設定です。

5 apLinux

apLinux は、MS104-SH4 用 Linux カーネルと RAM ディスク・CompactFlash 用ルートファイルシステムから構成される Linux システムです。

RAM ディスクルートファイルシステムは MS104-SH4 用に最適化されており、サイズは約 8Mbyte で、オンボード FlashMemory に十分に収まるため、CompactFlash を使用する必要がなく、消費電力を抑えることができます。

CompactFlash 用ルートファイルシステムは CELinux をベースに構成されており、サイズは約 40Mbyte で、RAM ディスクルートファイルシステムよりも機能が豊富で、高性能なシステムを構築できます。

5.1 apLinux (RAM ディスクルートファイルシステム) の概要

apLinux は Ethernet、CompactFlash (IDE/ATA)、Memory Technology Devices (MTD)、シリアルデバイスをサポートしており、ファイルシステムは ext2、ext3、JFFS2、cramfs、ramfs、NFS をサポートしています。

収録アプリケーションはコマンドユーティリティ群「busybox」、ログイン・ユーザ管理ツール群「tinylogin」、EXT2 ファイルユーティリティ「fdisk、mke2fs 等」、バックグラウンドプロセス管理サービス「cron」、Web サーバ「boa」などを収録しています。

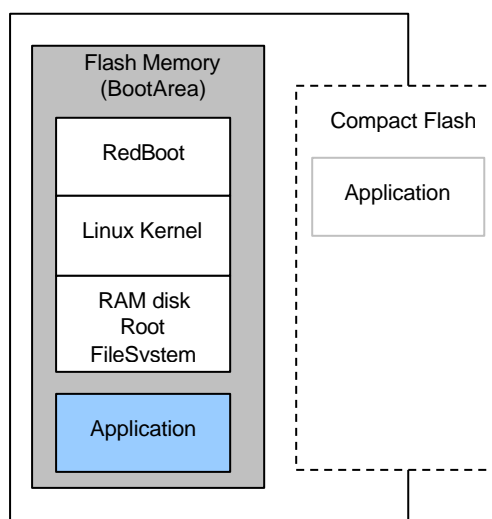


Fig 5.1-1 RAM ディスクのシステム構成例

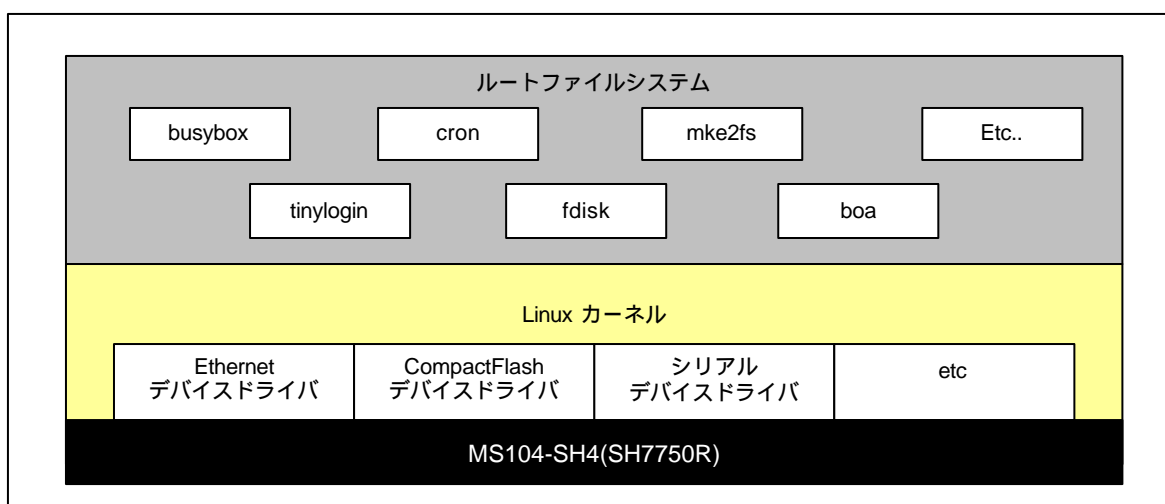


Fig 5.1-2 RAM ディスクシステムイメージルートファイルシステム

5.2 プログラム配置イメージ

RAMディスクルートファイルシステムを利用した apLinux は、RAM上にカーネル領域と仮想ディスク領域を確保し、それぞれカーネルとルートファイルシステムを展開して動作します。仮想ディスク領域は RAMディスクと呼び、フラッシュメモリに保存されているファイルシステムの圧縮イメージのことを RAMディスク圧縮イメージと呼びます。

なお、apLinux は RAMディスクルートファイルシステムを再構築することにより、CompactFlash や FlashROM 上にもルートファイルシステムを置くことができます。

MS104-SH4 のオンボードフラッシュメモリには、出荷時に apLinux が書き込まれています。

apLinux のフラッシュメモリの配置イメージと RAMへの展開イメージは次のようになっています。



Fig 5.2-1 メモリ展開イメージ

- 1 アドレスは、SH7750R の P1 領域アドレスを示します。
- 2 ブートローダの領域と Linux カーネルは重複していますが、Linux カーネルをロードする時点でブートローダは、0x8C001000 以降の領域を開放していますので、問題ありません。

5.3 RAM ディスクルートファイルシステムの編集

ルートファイルシステムは、使用する環境やアプリケーションにより編集する必要があります。本章では、RAM ディスクルートファイルシステムの編集方法について簡単に説明します。

開発環境

RAM ディスクルートファイルシステムの作成・変更には、Linux を搭載した開発用ホスト PC が必要です。

Table 5.3-1 推奨開発環境

パーソナルコンピュータ	PC/AT 互換機
OS	Fedora Core1

Fedora Core1 は Fedora プロジェクトのホームページ(<http://fedora.redhat.com/>)からダウンロードすることができます。

クロスコンパイラを使用する開発環境には、Fedora Core1 を推奨します。他のバージョンまたはディストリビューションの場合、コンパイルが正常にできない可能性があります。

ルートファイルシステムのディレクトリ構成

apLinux のルートファイルシステムのディレクトリ構成は下記のようになります。

Fig 5.3-1 ルートファイルシステム

/	: ルートディレクトリ
-- bin	: 基本コマンド
-- boot	: ブートファイル格納ディレクトリ
-- dev	: デバイスファイル
-- etc	: システムの設定ファイル
-- home	: 一般ユーザ用のホームディレクトリ
-- lib	: 共有ライブラリ
-- linuxrc -> bin/busybox	: 起動スクリプト
-- lost+found	: 破損ファイルの格納
-- mnt	: 一時的なマウントポイント
-- proc	: プロセス・デバイス情報などを格納するためのディレクトリ
-- root	: root 用ホームディレクトリ
-- sbin	: システム管理用コマンド
-- tmp	: テンポラリディレクトリ
-- usr	: ユーザ用各種プログラム・ディレクトリ格納領域
-- var	: ログ等の変更が多いデータを格納するためのディレクトリ

5.5 NFS (NetworkFileSystem)

NFS(Network File System)とはネットワークを介した分散ファイルシステムです。NFSを使用すれば、LinuxPC 上にある共有ディレクトリに存在するファイルを apLinux から共有することができます。例えば、下図のように MS104-SH4 のルートファイルシステムをマウントしておけば、変更のたびに MS104-SH4 のフラッシュメモリに書き込む必要はなくなります。

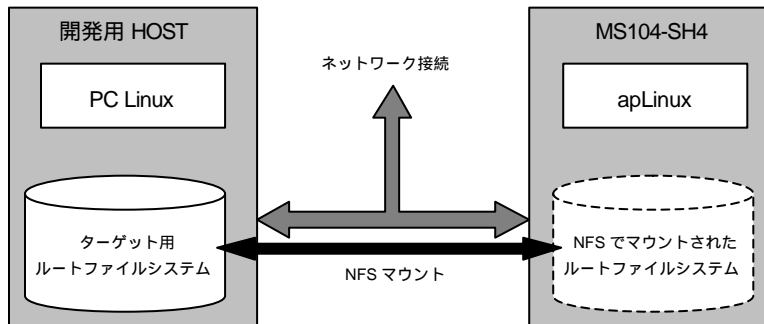


Fig 5.5-1 NFS の使用概念

NFS の使用方法

NFS を使用するには LinuxPC 上でサーバを設定・起動する必要があります。以下で Fedora Core1 を使用した NFS サーバの起動と apLinux から NFS をマウントする方法を説明します。

ホスト PC の IP アドレスは「192.168.1.201」、MS104-SH4 の IP アドレスを「192.168.1.200」と仮定します。

NFS サーバの起動

ホスト PC (LinuxPC) に root 権限でログインします。

```
[alpha@ap_linux ms104sh4]$ su -
Password:***** パスワードを入力してください
```

ホスト PC 上に共有ディレクトリを作成します。

ホスト PC 上で『mkdir /nfs』を実行してください。

```
[root@ap_linux root]# mkdir /nfs
```

共有ディレクトリの属性と所有者・グループを変更します。

ディレクトリの属性は実行・読み書きのすべての権限をすべてのユーザに与える属性にします。ディレクトリの所有者とグループは『nobody』に設定します。

```
[root@ap_linux root]# chmod a+rw /nfs
[root@ap_linux root]# chown nobody /nfs
[root@ap_linux root]# chgrp nobody /nfs
```


5.6 apLinux (CompactFlash ルートファイルシステム) の概要

CompactFlash 用ルートファイルシステムは CELinux をベースに構成されています。RAMディスク用ルートファイルシステムより豊富な機能が組み込まれており、WEB、FTP、TELNET といったネットワークサーバなどが動作します。

CompactFlash 用ルートファイルシステムは RAMディスク用ルートファイルシステムの CompactFlash フォーマット機能が利用し、セルフ環境で作成することができます。

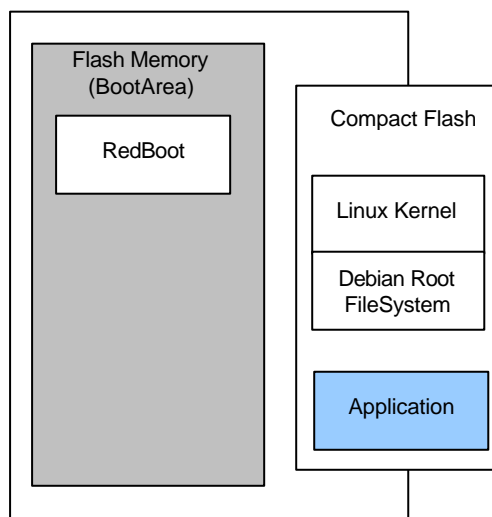


Fig 5.6-1 コンパクトフラッシュのシステム構成例

5.7 CompactFlash ルートファイルシステムの作成

CompactFlash ルートファイルシステムの作成は、RAM ディスクルートファイルシステムを使用した apLinux 上から NFS を用いて行います。LinuxPC 上で NFS サーバを起動し、適切な容量 (64MByte 以上) の CompactFlash を MS104-SH4 に挿し apLinux を起動してください。

ホスト PC の IP アドレスは「192.168.1.201」と仮定します。

MS104-SH4 で CompactFlash にパーティションを作成します。

apLinux 上で『fdisk /dev/hda』を実行してください。

```
[MS104SH4@root]# fdisk /dev/hda
hda: hda1

Command (m for help): d
Partition number (1-4): 1
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-782, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-782, default 782):
Using default value 782

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
```

すでにパーティションができている場合は『Selected partition 1』の表示のみ

fdisk コマンド

d : パーティションの削除 n : 新規パーティションの追加 w : 設定の書き込み、終了する

「fdisk」の詳細な使用については「fdisk」のマニュアルを参照するか、関連資料をご覧ください。

MS104-SH4 上で EXT2 フォーマットでファイルシステムを作成します。

```
[MS104SH4@root]# mke2fs /dev/hda1
```

MS104-SH4 上で CompactFlash のマウントポイントを作成します。

```
[MS104SH4@root]# mkdir -p /mnt/cf
```

MS104-SH4 上で CompactFlash をマウントします。

```
[MS104SH4@root]# mount /dev/hda1 /mnt/cf
```

ホスト PC (LinuxPC) に root 権限でログインします。

```
[alpha@ap_linux ms104sh4]$ su -
Password:*****
```

パスワードを入力してください

Table 5.7-1 CompactFlash ルートファイルシステム起動ログ

```

Uncompressing Linux... Ok, booting the kernel.
Linux version 2.4.20-celf3 (root@hg-linux) (gcc version 3.2.3) #5 Tue May 25 15:
10:14 JST 2004
kernel BUG at bootmem.c:86!
MS104-SH4 Setup...done
On node 0 totalpages: 8192
zone(0): 8192 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line: console=ttySC1,38400,N,8 mem=32M ether=8,0x300,0,0,eth0
MS104SH4 IRQ setup ..<6> done .
CPU clock: 236.25MHz
Bus clock: 59.06MHz
Module clock: 59.06MHz
Interval = 147662
sci_set_baud: t=47
Calibrating delay loop... 235.92 BogoMIPS
Use 'Preset loops_per_jiffy=1179648 for preset lpj.
Memory: 30444k/32768k available (1694k kernel code, 2324k reserved, 55k data, 40
k 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)
CPU: SH7750/SH7751/SH7750R/SH7751R/SH7760
POSIX conformance testing by UNIFIX
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
PC/104 bus for I/O space : physical addr 0x18000000, Dynamic bus sizing addr 0xc
0102000, 8 Bit addr 0xc0113000, 16 Bit addr 0xc0124000
PC/104 bus for memory space :physical addr 0x18000000, 8 Bit addr 0xc1136000, 16
Bit addr 0xc0135000
Starting kswapd
Disabling the Out Of Memory Killer
VFS: Disk quotas vdfquot 6.5.1
Installing knfsd (copyright (C) 1996 okir@monad.swb.de).
JFFS2 version 2.1. (C) 2001, 2002 Red Hat, Inc., designed by Axis Communications
AB.
SuperH SCI(F) driver initialized
ttySC0 at 0xffe00000 is a SCI
ttySC1 at 0xffe80000 is a SCIF
Real Time Clock Driver v1.10e
SH Real Time Clock Driver v0.01
Uniform Multi-Platform E-IDE driver Revision: 6.31
ide: Assuming 50MHz system bus speed for PIO modes; override with idebus=xx
hda: Hitachi XXM2.3.0, ATA DISK drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 12
hda: 125184 sectors (64 MB) w/1KiB Cache, CHS=978/4/32
Partition check:
hda: hda1
RAMDISK driver initialized: 16 RAM disks of 12288K size 1024 blocksize
loop: loaded (max 8 devices)
smc91111.c:v1.0saa 08/22/01 by Daris A Nevil (dnevil@snmc.com)
eth0: SMC91C11xFD(r:1) at 0x300 IRQ:8
MEM:8192b NOWAIT:0 ADDR 00:0c:7b:20:00:11
Probing for flash chips at 0x00000000:
cfi_cmdset_0001: Erase suspend on write enabled
Using buffer write method
MS104-SH4: Flash at 0x00000000
8 RedBoot partitions found on MTD device MS104-SH4 FLASH
Creating 8 MTD partitions on "MS104-SH4 FLASH":
0x00000000-0x00020000 : "RedBoot"
0x00020000-0x00040000 : "unallocated"
0x00040000-0x00140000 : "vmlinuz"
0x00140000-0x00540000 : "ramdisk.gz"
0x00540000-0x00640000 : "vmlinuz-vga"
0x00640000-0x00a40000 : "ramdisk-vga.gz"
0x00a40000-0x00fc0000 : "unallocated"
0x00fc0000-0x00fc1000 : "RedBoot config"
mtd: partition "RedBoot config" doesn't end on an erase block -- force read-only

Tracer: Initialization complete
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP, IGMP
IP: routing cache hash table of 512 buckets, 4Kbytes
TCP: Hash tables configured (established 2048 bind 4096)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Couldn't find valid RAM disk image starting at 0.
Freeing initrd memory: 0k freed
hda: hda1
hda: hda1
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 40k freed

```

```
modprobe: modprobe: Can't locate module char-major-4
INIT: version 2.84 booting
Mounting proc filesystem: [ OK ]
Configuring kernel parameters: [ OK ]
Setting clock (localtime): Tue May 25 15:46:11 UTC 2004 [ OK ]
Setting hostname MS104SH4: [ OK ]
Remounting root filesystem in read-write mode: [ OK ]
Activating swap partitions: [ OK ]
Finding module dependencies: [ OK ]
Checking filesystems
Checking all file systems.
[ OK ]
Mounting local filesystems: [FAILED]
INIT: Entering runlevel: 3
Setting network parameters: [ OK ]
Bringing up loopback interface: [ OK ]
Bringing up interface eth0: [ OK ]
Starting system logger: [ OK ]
Starting kernel logger: [ OK ]
Starting xinetd: [ OK ]
Starting vsftpd for vsftpd: [ OK ]
Starting boa: [ OK ]
Starting crond: [ OK ]

Alpha Project apLinux System
Kernel 2.4.20-celf3 on an sh4
MS104SH4 login:
```

バージョンやパッケージにより、起動メッセージが異なる場合があります。

CELinux ベースの起動スクリプトしているため、ローカルファイルシステムマウント時にネットワークファイルシステム (smbfs、nfsfs 等) もマウントしようとするため、表示は [FAILED] になりますが動作には影響ありません。

6 ハードウェア依存設定

本章では、MS104-SH4 のハードウェアに依存する設定について説明します。ハードウェアの詳細については、「MS104-SH4 ハードウェアマニュアル」をご覧ください。

6.1 レジスタ設定

MS104-SH4 に搭載されている SH7750R の BSC(Bus State Controller)と I/O ポートのレジスタは次のように設定されています。

Table 6.1-1 レジスタ設定

レジスタ名	データ	内容
周波数制御(FRQCR)	0x0E1A	クロック設定 CPU クロック 235.9296MHz バスクロック 58.9824MHz モジュールクロック 58.9824MHz
バスコントロール 1 (BCR1)	0x80000809	エンディアン設定 LittleEndian メモリアイプ設定 エリア 0: バースト ROM エリア 1: 通常モード エリア 2: SRAM エリア 3: SDRAM エリア 4: 通常モード エリア 5: PCMCIA エリア 6: PCMCIA
バスコントロール 2 (BCR2)	0xAAE5	バス幅設定(ビット) エリア 0: 16 エリア 1: 8 エリア 2: 16 エリア 3: 32 エリア 4: 16 エリア 5: 16 エリア 6: 16
バスコントロール 3 (BCR3)	0x0000	エリア設定 エリア 1 を SRAM インターフェースに設定 エリア 4 を SRAM インターフェースに設定
バスコントロール 4 (BCR4)	0x00000013	非同期入力設定 ISIS16、BREQ、RDY を非同期入力
ウェイトコントロール 1 (WCR1)	0x71111111	サイクル間アイドル指定(アイドルサイクル数) エリア 0: 1 エリア 1: 1 エリア 2: 1 エリア 3: 1 エリア 4: 1 エリア 5: 1 エリア 6: 1 DMA: 15
ウェイトコントロール 2 (WCR2)	0xDFF2466A 1	ウェイトステート数(ウェイトサイクル) エリア 0: 9 エリア 1: 1 エリア 2: 3 エリア 3: 2 エリア 4: 1 エリア 5: 15 エリア 6: 12
		SDRAM ウェイト(エリア 3) CAS レイテンシ: 2
ウェイトコントロール 3 (WCR3)	0x07717515	ライトストロープセットアップ時間 エリア 0: 1 エリア 1: 0 エリア 2: 1 エリア 4: 0 データホールド時間 エリア 0: 1 エリア 1: 1 エリア 2: 1 エリア 4: 1
個別メモリコントロール (MCR)	0x1009219C	SDRAM モードレジスタ設定 リフレッシュ後 RAS プリチャージ期間: 6 サイクル CAS ネゲート期間: 1 サイクル RAS プリチャージ期間: 2 サイクル RAS-CAS 遅延: 2 サイクル 書き込みプリチャージ遅延: 2 サイクル リフレッシュ後のコマンド間隔: 4+TRC サイクル データサイズ: 32

1 エリア 5 のウェイトは CF カードの CIS 情報に基づいて再設定します。

PCMCIA コントロール (PCR)	0xCF3B 1	ウェイト数 エリア 5 : 50 エリア 6 : 0 アドレス - OE/WEアサート遅延 エリア 5 : 15 エリア 6 : 6 OE/WE - アドレスネゲート遅延 エリア 5 : 15 エリア 6 : 6
リフレッシュタイムコントロール (RTCSR)	0xA508	クロックセレクト CKIO/4
リフレッシュタイムカウンタ (RTCNT)	0xA500	タイムカウンタ 初期値 : 0
リフレッシュタイムコンスタント (RTCOR)	0xA5BC	リフレッシュサイクル 4kCycle/64msec
リフレッシュカウンタ (RFCR)	0xA400	リフレッシュカウンタ 初期値 : 0
シンクロナス DRAM モード (SDMR3)	0xFF94008C (アドレス) 0x00 (データ)	SDRAM設定 バス幅 : 32 バースト長 : 8 CAS レイテンシ : 2
ポートコントロール A	0x54540000	I/O ポート A 入出力設定 P15 : 出力 P14 : 出力 P13 : 出力 P12 : 入力 P11 : 出力 P10 : 出力 P9 : 出力 P8 : 入力 P7-0 : 入力 : ユーザ開放 J5 に接続
ポートデータ A	0x0E00	I/O ポート A コントロール
ポートコントロール B	0x000000BF	I/O ポート B 入出力設定 P19 : 入力 P18 : 出力 P17 : 出力 P16 : 出力
ポートデータ B	0x0004	I/O ポート B コントロール

6.2 MMU 設定

MS104-SH4 に搭載されている SH7750R は MMU(Memory Management Unit)を内蔵しています。

MMU とは、物理メモリ空間に仮想メモリ空間を効率よく割り当てるための機能で、Linux 等の仮想メモリを使用する OS には不可欠な機能です。MS104-SH4 では、Linux カーネルによるプロセスのメモリ割り当てのほかに、CompactFlash と PC104 バスのアクセスにも MMU を使用します。

RedBoot では MMU を固定で使用します。Linux ではページング機構を利用し、MMU の設定を動的に切り換えながら使用します。MS104-SH4 における MMU の RedBoot での設定は「Table6.2.1 RedBoot での MMU の設定」のようになっています。

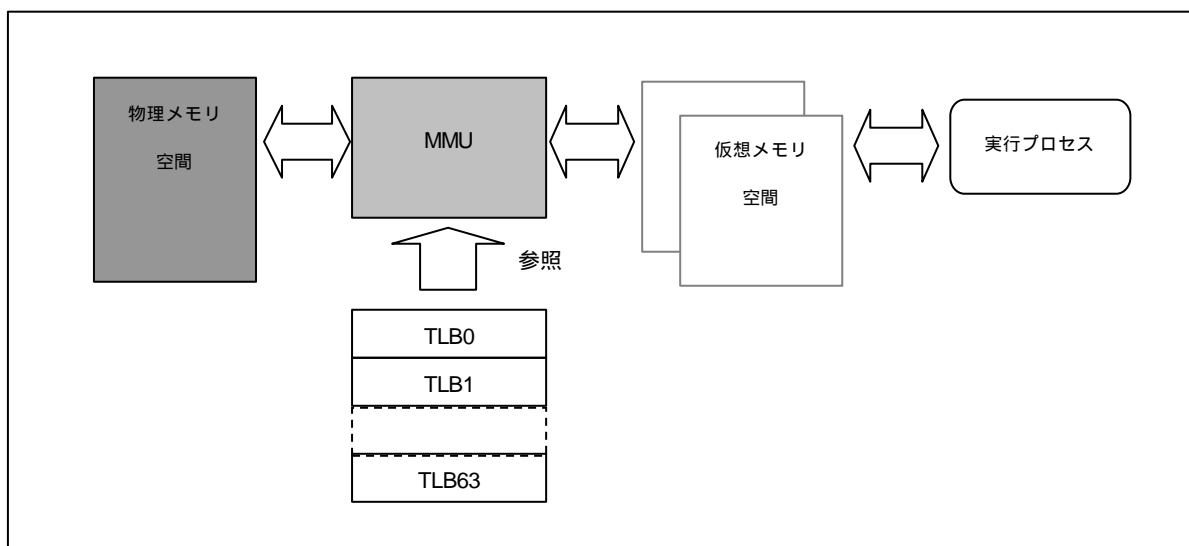


Fig 6.2-1 MMUの動作イメージ

RedBoot 上での MMU の設定は『`ecos-2.0/packages/sh/ms104sh4/v2_0/include/plf_mmu.h`』ファイルで定義しています。実際に MMU の設定を行っているのは『`ecos-2.0/packages/sh/ms104sh4/v2_0/src/plf_mmu.c`』ファイルになります。RedBoot ソースファイルは弊社 Linux 開発環境セット「Linux-KIT-A02」、もしくは、弊社組込み Linux ホームページ <http://www.apnet.co.jp/e-linux/index.html> から入手することができます。

MMU の設定方法については『SH-4 SH7750 シリーズ ハードウェアマニュアル』をご覧ください。

Table 6.2-1 RedBoot での MMU設定

物理アドレス	サイズ	ページ番号	仮想アドレス	用途
-	-	0 ~ 25	-	未使用
0x14000000	4kbyte	26	0x14000000	CompactFlash 16bit アトリビュートメモリ
0x14000000	4kbyte	27	0x14010000	CompactFlash 16bit 共有メモリ
0x14000000	4kbyte	28	0x14020000	CompactFlash 16bit I/O 空間
0x18000000	64kbyte	29	0x19000000	PC104 ダイナミックバスサイジング I/O 空間
0x18000000	64kbyte	30	0x19100000	PC104 8bit I/O 空間
0x18000000	64kbyte	31	0x19200000	PC104 8bit I/O 空間
0x18000000	1Mbyte	32	0x20000000	PC104 8bit メモリ空間
0x18100000	1Mbyte	33	0x20100000	PC104 8bit メモリ空間
0x18200000	1Mbyte	34	0x20200000	PC104 8bit メモリ空間
0x18300000	1Mbyte	35	0x20300000	PC104 8bit メモリ空間
0x18400000	1Mbyte	36	0x20400000	PC104 8bit メモリ空間
0x18500000	1Mbyte	37	0x20500000	PC104 8bit メモリ空間
0x18600000	1Mbyte	38	0x20600000	PC104 8bit メモリ空間
0x18700000	1Mbyte	39	0x20700000	PC104 8bit メモリ空間
0x18800000	1Mbyte	40	0x20800000	PC104 8bit メモリ空間
0x18900000	1Mbyte	41	0x20900000	PC104 8bit メモリ空間
0x18a00000	1Mbyte	42	0x20a00000	PC104 8bit メモリ空間
0x18b00000	1Mbyte	43	0x20b00000	PC104 8bit メモリ空間
0x18c00000	1Mbyte	44	0x20c00000	PC104 8bit メモリ空間
0x18d00000	1Mbyte	45	0x20d00000	PC104 8bit メモリ空間
0x18e00000	1Mbyte	46	0x20e00000	PC104 8bit メモリ空間
0x18f00000	1Mbyte	47	0x20f00000	PC104 8bit メモリ空間
0x18000000	1Mbyte	48	0x18000000	PC104 16bit メモリ空間
0x18100000	1Mbyte	49	0x18100000	PC104 16bit メモリ空間
0x18200000	1Mbyte	50	0x18200000	PC104 16bit メモリ空間
0x18300000	1Mbyte	51	0x18300000	PC104 16bit メモリ空間
0x18400000	1Mbyte	52	0x18400000	PC104 16bit メモリ空間
0x18500000	1Mbyte	53	0x18500000	PC104 16bit メモリ空間
0x18600000	1Mbyte	54	0x18600000	PC104 16bit メモリ空間
0x18700000	1Mbyte	55	0x18700000	PC104 16bit メモリ空間
0x18800000	1Mbyte	56	0x18800000	PC104 16bit メモリ空間
0x18900000	1Mbyte	57	0x18900000	PC104 16bit メモリ空間
0x18a00000	1Mbyte	58	0x18a00000	PC104 16bit メモリ空間
0x18b00000	1Mbyte	59	0x18b00000	PC104 16bit メモリ空間
0x18c00000	1Mbyte	60	0x18c00000	PC104 16bit メモリ空間
0x18d00000	1Mbyte	61	0x18d00000	PC104 16bit メモリ空間
0x18e00000	1Mbyte	62	0x18e00000	PC104 16bit メモリ空間
0x18f00000	1Mbyte	63	0x18f00000	PC104 16bit メモリ空間

Linux は SH の持つ MMU 機能を利用し、物理アドレスと仮想アドレスの管理を行います。Linux のアドレス変換は各種 CPU に対応するため 3 段引きページテーブルを採用しています。それぞれページテーブルは上位から PGD、PMD、PTE となります。SH CPU は 2 段引きページであるため、PMD は PGD と同一となり、Linux 上でも 2 段階のページングが行われます。MS104-SH4 用 Linux では `remap_area_pages` 関数を使用して、PTE に PC/104 バスの物理アドレスを登録しています。PTE への登録は `arch/sh/kernel/pc104_ms104sh4.c` ファイルで行っています。CompactFlash バスでは `arch/sh/kernel/cf-enabler.c` ファイル中で PTE への登録を行っています。

物理アドレスを PTE へ登録が成功すると、割り当てられた仮想アドレスの先頭番地を変数に登録します。この変数は `include/asm-sh/io_ms104sh4.c` でマクロとして定義されています。各 PC/104 バスと変数名の対応は「Table 6.2.2 Linux での MMU の設定」に示します。

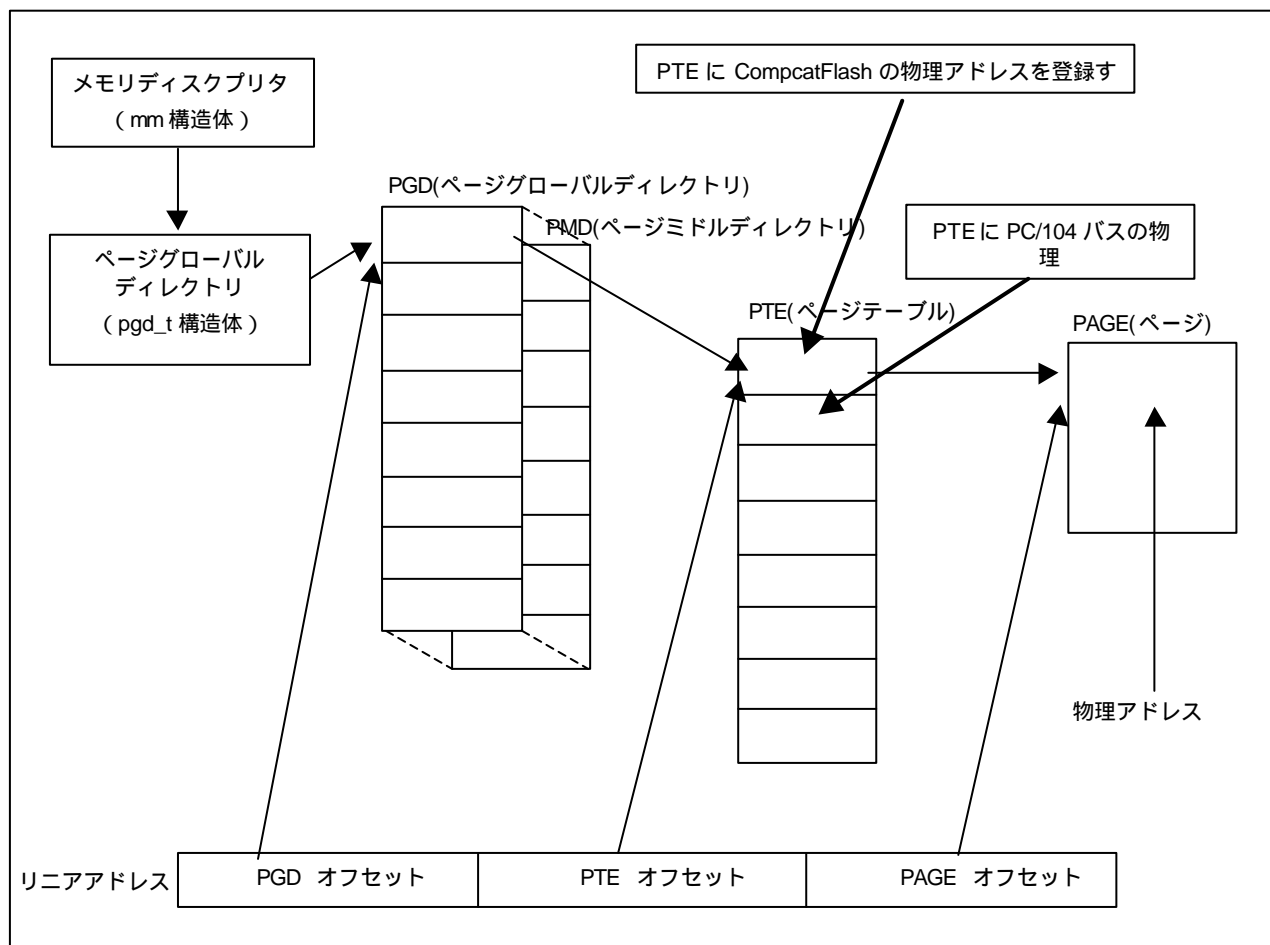


Fig 6.2-2 Linux ページング機構

Table 6.2-2 Linux での MMU 設定

用途	変数(マクロ)名
CompactFlash 16bit I/O 空間	cf_io_base
PC104 ダイナミックバスサイジング I/O 空間	PC104_IO_ADDR
PC104 8bit I/O 空間	PC104_IO_8_ADDR
PC104 8bit I/O 空間	PC104_IO_16_ADDR
PC104 8bit メモリ空間	PC104_MEM_8_ADDR
PC104 16bit メモリ空間	PC104_MEM_ADDR

Linux ソースファイルは弊社 Linux 開発環境セット「Linux-KIT-A02」、もしくは、弊社組込み Linux ホームページ <http://www.apnet.co.jp/e-linux/index.html> から入手することができます。

6.3 I/O マップ

Linux は元々 PC/AT 互換機用に開発された OS です。そのため、デバイスドライバの I/O ポートアドレス (ISA アドレス) や IRQ 番号は、PC/AT アーキテクチャに従ったアドレスや番号になっています。

SH プロセッサは I/O 空間の概念がなく、メモリマップド I/O アーキテクチャです。したがって、Linux のデバイスドライバで使用される I/O ポートアドレスを MS104-SH4 に割り当てられた物理メモリまたは仮想メモリアドレスに変換する必要があります。apLinux におけるデバイスドライバの I/O ポートアドレスと SH CPU のメモリアドレスの対応は以下のとおりです。これら I/O マップの変換は `arch/sh/kernel/io_ms104sh4.c` の `ms104sh4_isa_port2addr` 関数で行っています。

Table 6.3-1 I/O マップ

デバイス	ISA アドレス(Linux)	メモリアドレス(SH)
IDE(CompactFlash)	0x1F0 ~ 0x1F7 0x3F6	Linux が任意に選択
NIC(イーサネット)	0x1300 ~ 0x131F	0xA8001300 ~ 0xA800131F
USB (MS104-USB H/S)	0x1000 ~ 0x1004	PC104_IO_8_ADDR (Linux が任意に選択)
PC104 I/O 空間	0x00000 ~ 0x10000 * 上記の ISA アドレス以外	PC104_IO_ADDR (Linux が任意に選択)

6.4 IRQ 番号

MS104-SH4 で使用される外部割り込みは Linux 上で以下の IRQ 番号が割り当てられています。

Table 6.4-1 IRQ 番号

IRQ 番号	割り込み要因
1	PC104 IRQ15
2	PC104 IRQ14
3	PC104 IRQ12
4	PC104 IRQ11
5	PC104 IRQ10
6	PC104 IRQ9
7	PC104 IRQ7 (MS104-USB H/S スレーブ使用時)
8	NIC(Ethernet)
9	PC104 IRQ6 (MS104-USB H/S ホスト使用時)
10	PC104 IRQ5 (MS104-USB H/S ホスト使用時)
11	PC104 IRQ4 (MS104-VGA/LCD 使用時 Touch panel)
12	IDE(CompactFlash)
13	PC104 IRQ3
14	RTC

6.5 LED 設定

MS104-SH4 の LED は電源の投入、CompactFlash の認識、GDB スタブの認識のために RedBoot が使用します。

Fig 6.5-1 LED 設定

LED 番号	説明
LED1	GDB スタブの認識時に点灯
LED2	CompactFlash の認識時に点灯
LED3	電源投入時に点灯

付録 A 動作確認済み CompactFlash

TableA-1 動作確認済み CF一覧

メーカー名	シリーズ名	確認カード
I・O DATA	CFS シリーズ	CFS-iV64 CFS-iV128
	PCCF シリーズ	PCCF-64MS PCCF-128MS PCCF-192MS
BUFFALO	RCF-X シリーズ	RCF-X 255MB
ハギワラシスコム	V シリーズ	HPC-CF256V
SanDisk	SDCFB シリーズ	SDCFB-128-801 SDCFB-256-801 SDCFB-512-801
LEXAR MEDEA	8 倍速	MCF-64Mx8

付録 D RAM ディスクルートファイルシステム収録パッケージ一覧

パッケージ一覧

Table D-1 RAM ディスクルートファイルシステム収録パッケージ一覧

パッケージ名	説明
dhclient	DCHP クライアント
e2fsprogs	Linux ファイルシステムユーティリティコマンド・ライブラリ群
glibc	標準ライブラリ群
iptables	IP パケットフィルタ管理コマンド群
libgcc_s	GCC 用ランタイムライブラリ
libstdc++	C++ランタイムライブラリ
libtermcap	端末管理用ライブラリ群
ncurses	スクリーン制御用ライブラリ群
tcp_wrappers	TCP 監視、フィルタリングデーモンプログラム
util-linux	システムユーティリティコマンド群 (fdisk・hwclock収録)
vixie-cron	プロセス管理デーモン
xinetd	インターネットサービスデーモン
(alp-)busybox	組み込み機器向けユーティリティコマンド群 (MS104-SH4 用に修正)
(alp-)tinylogin	組み込み機器向けログイン・ユーザ管理コマンド群 (MS104-SH4 用に修正)
(alp-)boa	組み込み用小型 Web サーバ (MS104-SH4 用に修正)
alp-initscripts	MS104-SH4 用設定ファイル群
alp-MAKEDEV	MS104-SH4 用デバイスファイル群

パッケージ名の先頭に「alp-」がないパッケージは、CELinux フォーラムで公開されているパッケージを使用しています。
http://tree.celinuxforum.org/toolchains/Userland_sh4_RPMS.tar.gz

コマンド一覧

Table D-2 /bin ディレクトリ(基本コマンド)

コマンド名	パッケージ	説明
addgroup	tinylogin	グループの追加
adduser	tinylogin	ユーザの追加
ash	busybox	Bash ライクなシェル
busybox	busybox	Linux 基本ツール群
cat	busybox	ファイルを連結して出力
chgrp	busybox	ファイルグループの変更
chmod	busybox	ファイル属性の変更
chown	busybox	ファイル所有者の変更
cp	busybox	ファイルのコピー
cpio	busybox	cpio アーカイブファイルの抽出
date	busybox	システム時間の表示・変更
dd	busybox	ファイルのコピー(ファイル形式を選択)
delgroup	tinylogin	グループの削除
deluser	tinylogin	ユーザの削除
df	busybox	ファイルシステム情報の表示
dmseg	busybox	カーネルのリングバッファの表示・制御
dumpkmap	busybox	キーボード変換テーブルの出力
echo	busybox	1 行のテキストを表示する
false	busybox	終了ステータスとして 1 以外を返す
getopt	busybox	コマンドの引き数を解析する
grep	busybox	パターンにマッチする行を表示する
gunzip	busybox	ファイルの圧縮・伸長を行う

付録 E CompactFlash ルートファイルシステム収録パッケージ一覧

パッケージ一覧

Table E-1 CompactFlash ルートファイルシステム収録パッケージ一覧

パッケージ名	説明
bash	Bourne シェル
bzip2	品質ブロックソーティング圧縮プログラムユーティリティ
cpio	ファイルアーカイブ管理プログラム
cracklib	順向抑制的なパスワードチェックライブラリ
devfsd	デバイスファイルシステム用デーモン
dhclient	DHCP クライアント
diffutils	ファイル比較ユーティリティ
e2fsprogs	ext2 ファイルシステムユーティリティ・ライブラリ
filesystem	CELinux 基本ファイルシステム
fileutils	GNU ファイル管理ユーティリティ
findutils	ファイル検索ユーティリティ
ftp	FTPクライアント
gawk	GNU awkパターン検索 & 処理言語
gdb	GNU デバッガ
glibc	GNU C ライブラリ (共有ライブラリ・タイムゾーンデータ)
grep	GNU grep、egrep、fgrep
gzip	GNU 圧縮ユーティリティ
hdparm	高性能発揮用にハードディスクのパラメータを調整
initscripts	CELinux 基本システムスクリプト
iproute	Linux カーネルのネットワーク機能の制御
iptables	IP パケットフィルタ管理ツール群
iputils	IP ネットワークユーティリティ
libcap1	POSIX.1e サポート
libgcc_s	GCC サポートライブラリ
libstdc++	GNU 標準 C++ライブラリ
libtermcap	termcap ライブラリ
mingetty	コンソールのための getty プログラム
minicom	シリアルコンソール端末
modutils	Linux モジュールユーティリティ
ncurses	共通のターミナルファイル
net-tools	ネットワークツール
pam	認証モジュール
portmap	RPC ボートマッパー
procps	/proc ファイルシステムユーティリティ
psmisc	proc ファイルシステム用ユーティリティファイル群
readline	コマンド行インターフェース用ユーザインターフェースライブラリ
sed	GNU sed ストリームエディタ
setup	コンフィグレーション・設定ファイル群
sh-utils	GNU シェルプログラミングユーティリティ
shadow-utils	アカウント管理・シャドウパスワードユーティリティ
strace	システムコール分析ツール
sysklogd	システムログデーモン

Sysvinit	SystemV 風 init
tar	GNU tar
tcp_wrappers	TCP ラッパーユーティリティ
tcpdump	ネットワーク監視ツール
telnet	TELNET クライアント
telnet-server	TELNET サーバ
termcap	ターミナル用データベース
textutils	GNU テキストファイル処理ユーティリティ
tftp	TFTPクライアント
tftp-server	TFTPサーバ
util-linux	システムユーティリティ
vixie-cron	バックグラウンドプロセス
vsftpd	FTPサーバ
xinetd	ネットワークスーパーサーバ
zlib	圧縮ライブラリ
alp-MAKEDEV	MS104-SH4 用デバイスファイル
alp-boa-cf	CompactFlash ルートファイルシステム用小型 Web サーバ
alp-busybox-cf	vi と rpm2cpio コマンド
alp-setup-ms104sh4-cf	CompactFlash ルートファイルシステム用設定ファイル群

パッケージ名の先頭に「alp-」がないパッケージは、CELinux フォーラムで公開されているパッケージを使用しています。
http://tree.celinuxforum.org/toolchains/Userland_sh4_RPMS.tar.gz

製品サポートのご案内

● ユーザ登録

ユーザ登録は弊社ホームページにて受け付けております。ユーザ登録をしていただきますと、ユーザ専用ページにアクセスすることができます。ユーザ専用ページでは、最新版のマニュアルやソフトウェア、またアプリケーションノート等、お客様にお役立ていただける情報を掲載しておりますので是非ご利用ください。

弊社ホームページアドレス <http://www.apnet.co.jp>

● ハードウェアのサポート

万が一、製作上の不具合や回路の機能的な問題を発見された場合には、お手数ですが弊社サポートまでご連絡ください。以下の内容に該当するお問い合わせにつきましては受け付けておりませんのであらかじめご了承ください。

- 本製品の回路動作及び CPU および周辺デバイスの使用方法に関するご質問
- ユーザ回路の設計方法やその動作についてのご質問
- 関連ツールの操作指導
- その他、製品の仕様範囲外の質問やお客様の技術によって解決されるべき問題

● ソフトウェアのサポート

ソフトウェアに関する技術的な質問は、受け付けておりませんのでご了承ください。

サポートをご希望されるお客様には、別途有償サポートプログラムをご用意しておりますので、弊社営業までご連絡ください。

● バージョンアップ

本製品に付属するソフトウェアは、不定期で更新されます。それらは全て弊社ホームページよりダウンロードできます。FD や CD-ROM などの物理媒体での提供をご希望される場合には、実費にて承りますので弊社営業までご連絡ください。

● 修理の依頼

修理をご依頼いただく場合には、お名前、製品名、シリアル番号、詳しい故障状況を弊社製品サポートへご連絡ください。弊社にて故障状況を確認のうえ、修理の可否、修理費用等をご連絡いたします。ただし、過電圧印加や高熱等により製品全体がダメージを受けていると判断される場合には、修理をお断りする場合もございますのでご了承ください。なお、弊社までの送料はお客様ご負担となります。

● 製品サポートの方法

製品サポートについては、FAX もしくは E-MAIL でのみ受け付けております。お電話でのお問い合わせは受け付けておりませんのでご了承ください。なお、お問い合わせの際には、製品名、使用環境、使用方法等、問題点を詳細に記載してください。

製品サポート窓口

■ FAX	053-401-0035
■ E-MAIL	query@apnet.co.jp

エンジニアリングサービスのご案内

弊社製品をベースとしたカスタム品やシステム開発を承っております。

お客様の仕様に合わせて、設計から OEM 供給まで一貫したサービスを提供いたします。

詳しくは、弊社営業窓口までお問い合わせください。

営業案内窓口

■ TEL	053-401-0033 (代表)
■ E-MAIL	sales@apnet.co.jp

改定履歴

版数	日付	改定内容
1 版	2005/01/20	新規作成

謝辞

Linux、GNU プロジェクト、SH-Linux、eCos/Redboot、CE-Linux の開発に関する多くの貢献者の方々に深い敬意と感謝の意を表します。

参考文献

- | | |
|---|--|
| 「SH7750 シリーズハードウェアマニュアル」 | ルネサステクノロジ |
| 「PC/104 Specification」 | PC/104 Consortium |
| 「SH-3&SH-4 プロセッサ入門」 | CQ 出版 |
| 「技術者のための UNIX 系 OS 入門」 | CQ 出版 |
| 「エンジニアリング Linux 応用技法」 | CQ 出版 |
| 「組み込み Linux 入門」 | 日本エンベデッド リナックス コンソーシアム監修 / CQ 出版 |
| 「LINUX デバイスドライバ 第2版」 | Alessandro rubini, Jonathan corbet 著
山崎康宏、山崎邦子、長原宏治、長原陽子 訳 / オライリージャパン |
| 「詳解 LINUX カーネル 第2版」 | Daniel P. Bovet, Marco Cesati 著 高橋弘和 監訳
杉田真由美、高杉昌督、畑崎恵介、平松正巳、安井隆宏 訳 / オライリージャパン |
| 「Embedded Software Development with eCos」 | Anthony J. Massa / Prentice Hall |

<http://www.celinuxforum.org/>
CELinux フォーラム。

<http://www.jp.redhat.com/>
レッドハット株式会社ホームページ。本製品に収録されている eCos/RedBoot の開発元。

<http://www.ecoscentric.com/index.shtml>
eCosCentric Ltd. ホームページ。現在レッドハット株式会社代わり eCos/RedBoot の保守・サポートをしています。

<http://sources.redhat.com/redboot/>
RedBoot 公式ホームページ

<http://www.gnu.org/japan/>
GNU プロジェクトホームページ(日本国内向け)

<http://www.busybox.net/>
Busybox 公式ホームページ

<http://tinylogin.busybox.net/>
Tinylogin 公式ホームページ

<http://www.boa.org/>
Boa 公式ホームページ

<http://www.m17n.org/linux-sh/>
GNU/Linux on SuperH Project のホームページ。

著作権について

- ・本文書の著作権は（株）アルファプロジェクトが保有します。
- ・本文書の内容を無断で転載することは一切禁止します。
- ・本文書の内容は、将来予告なしに変更されることがあります。
- ・本文書の内容については、万全を期して作成いたしました。万が一不審な点、誤りなどお気づきの点がありましたら弊社までご連絡下さい。
- ・本文書の内容に基づき、アプリケーションを運用した結果、万が一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。

商標について

- ・ SuperH は、（株）ルネサステクノロジーの登録商標、商標または商品名称です。
 - ・ Linux は、Linus Torvalds の米国およびその他の国における登録商標または商標です。
 - ・ eCos™ および RedBoot™ は RedHat™ 社の商標です。
 - ・ コンパクトフラッシュはサンディスク社の商標です。
 - ・ Windows™ はマイクロソフト社の商標です。
 - ・ RedHatLinux™ はレッドハット社の商標です。
- ・ その他の会社名、製品名は、各社の登録商標または商標です。



株式会社アルファプロジェクト
〒431-3114
静岡県浜松市東区積志町 834
<http://www.apnet.co.jp>
E-MAIL : sales@apnet.co.jp
