

MS104-SH4

MS104-USB H/S の使用方法

1版 2005年02月18日

目次

1. 製品紹介	1
1.1 Linux について.....	1
1.2 USB について.....	1
1.3 Microwindows について	2
2. MS104-USB H/S について	3
2.1 MS104-USB H/S の概要	3
2.2 MS104-USB H/S の接続	4
2.4 デバイスドライバ	5
3. MS104-USB H/S の起動	6
3.1 MS104-USB H/S の動作環境	6
3.2 MS104-USB H/S の設定	8
3.3 USB デバイスの使用	9
4. Linux	20
4.1 Linux カーネルの概要	20
4.2 Linux カーネルのコンフィグレーション	21
4.3 アドレスの設定	24
4.4 Linux カーネルのコンパイル.....	25
4.5 Microwindows 概要	29
4.6 Microwindows のインストール	30
4.7 ルートファイルシステムの概要	34
4.8 ルートファイルシステムの作成	35
4.9 サンプルプログラムの作成	37
5. 保証とサポート	42

1. 概要

本アプリケーションノートは USB ホスト・スレーブコントローラボード「MS104-USB H/S」を MS104-SH4 用 Linux で使用方法について述べます。

MS104-USB H/S は、SL811HST (Cypress 社) USB ホスト・スレーブコントローラを 2 機搭載した PC/104 規格準拠周辺ボードです。MS104-USB H/S は、USB ホスト・スレーブデバイス両対応で、フルスピード (12M) とロースピード (1.5M) デバイスに対応しています。

MS104-SH4 と MS104-USB H/S を組み合わせることにより、LinuxUSB ホストコントローラとして動作させることができます。本アプリケーションノートでは、MS104-SH4 用 Linux を使用して、USB フラッシュメモリ、USB キーボード、USB マウスの動作方法について説明します。

本アプリケーションノートを実行するには、必ず「MS104-SH4 Linux 開発環境キット Linux-KIT-A02」がインストールされている必要があります。

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。そのため、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 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 USB について

USB (Universal Serial Bus) は Compaq (現 Hewlett-Packard)、Intel、Microsoft、NEC の四社により策定された PC と周辺機器とのデータ転送方式の一つです。USB は現在バージョン 2.0 が最新版となります。USB2.0 では、従来の USB1.1 の Full Speed (12Mbps) と Low Speed (1.5Mbps) に加え、High Speed (480Mbps) が追加されました。USB2.0 では上位互換が確保されており、USB1.1 準拠の製品は USB2.0 の環境でも使用することができます。USB は通信をすべてホスト側で制御するようになっており、ホストと各デバイス間の通信では必ずホストから各デバイスに対して転送要求を行います。USB は、1 台のホストからハブを介することによって、最大 127 個のデバイスを接続することができます。USB の詳細については、インターネット上に規格書が公開されておりますので、そちらをご覧ください。

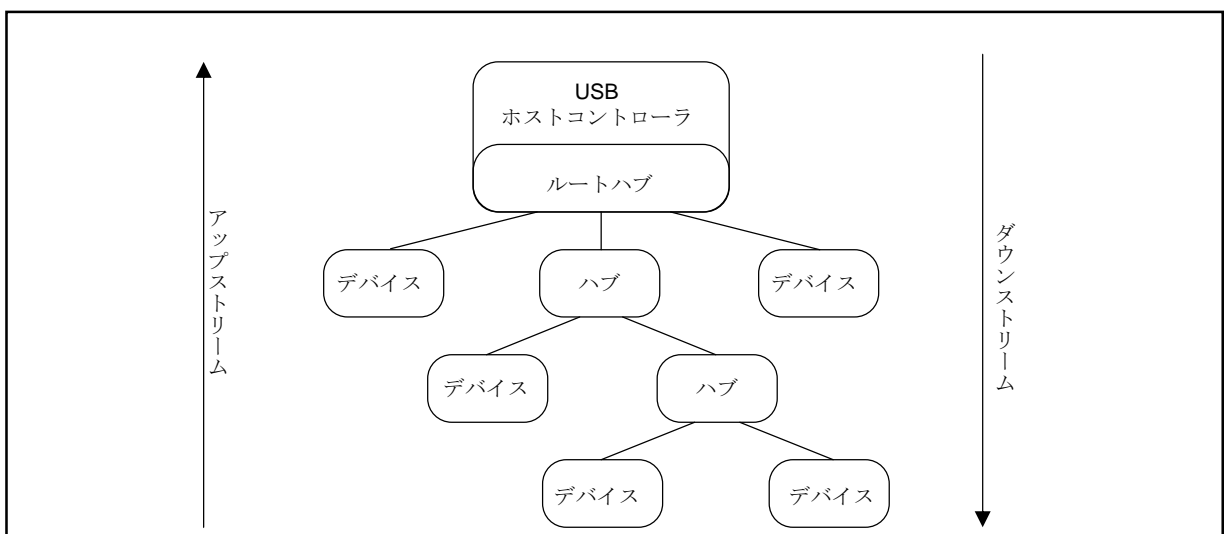


Fig1.2-1 USB のバスポジ

1.3 Microwindows について

Microwindows は Century Software 社のオープンソースのプロジェクトで、組み込み機器向けに設計されたウィンドウシステムです。Microwindows は 3 階層で設計されており、最下層はモニターやマウス、キーボードなどの入出力デバイスのドライバです。中間層は線描、領域塗りつぶし、クリッピング、カラーモデルなどのグラフィックエンジン、最上位層は 2 つの API をサポートしています。

API は Microwindows と呼ばれる Win32 API を模した API と Nano-X と呼ばれる X Window System の API を模した API の 2 種類です。Nano-X は Linux で採用されています。

Microwindows はフレームバッファ（グラフィックスイメージを保持するメモリ領域）をサポートしているため、非常に高速に動作し、PC 用の X Window System に比べ、Nano-X サーバのサイズが 100~200KB 程度と非常にサイズが小さくなります。

2. MS104-USB H/S について

2.1 MS104-USB H/S の概要

MS104-USB H/S は USB2.0 の High Speed (12Mbps) と Low Speed (1.5Mbps) に対応した、PC/104 バス準拠の USB ホスト・スレーブコントローラボードです。

MS104-USB H/S は弊社 CPU ボード『MS104-SH4』と組み合わせることにより、LinuxUSB ホストコントローラとして動作することができます。

MS104-SH4 用 Linux (apLinux) には MS104-USB H/S 用ホストデバイスドライバが組み込まれており、USB マスストレージ規格に準拠する USB フラッシュメモリや、弊社 PC/104 バス準拠グラフィックボード『MS104-VGA/LCD』と組み合わせることにより USB キーボード、組込み向けウィンドウシステム『Microwindows』と組み合わせれば USB マウスを動作させることができます。

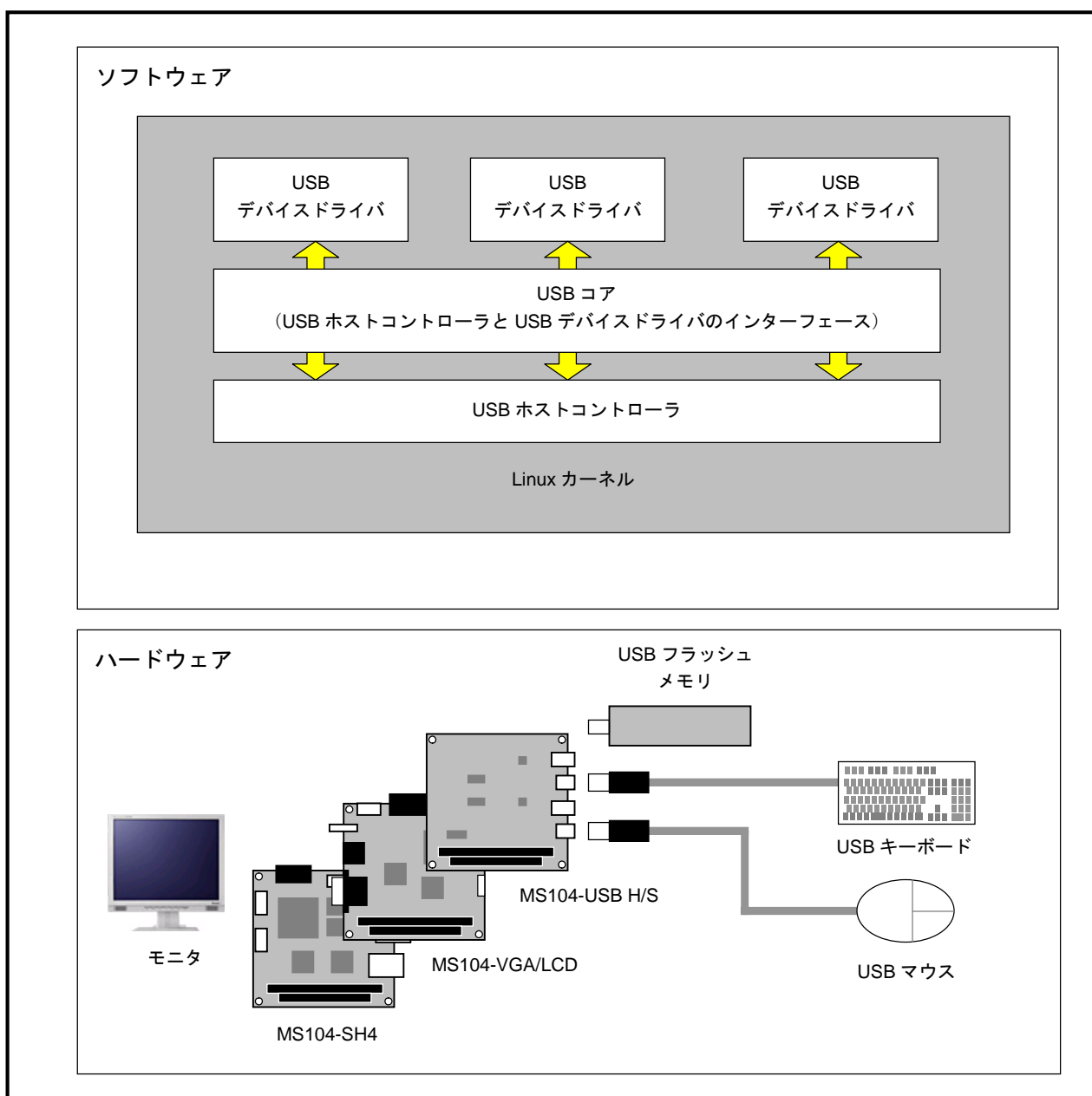


Fig 2.1-1 MS104-USB H/S の概要

2.2 MS104-USB H/S の接続

MS104-USB H/S は MS104-SH4 と組み合わせることにより、LinuxUSB ホストコントローラとして動作させることができます。また、MS104-VGA/LCD と組み合わせることにより、USB キーボード・マウスといった入力デバイスの動作を確認することができます。

下図に MS104-USB H/S および MS104-SH4 を使用したときの USB フラッシュメモリとの接続例を示します。

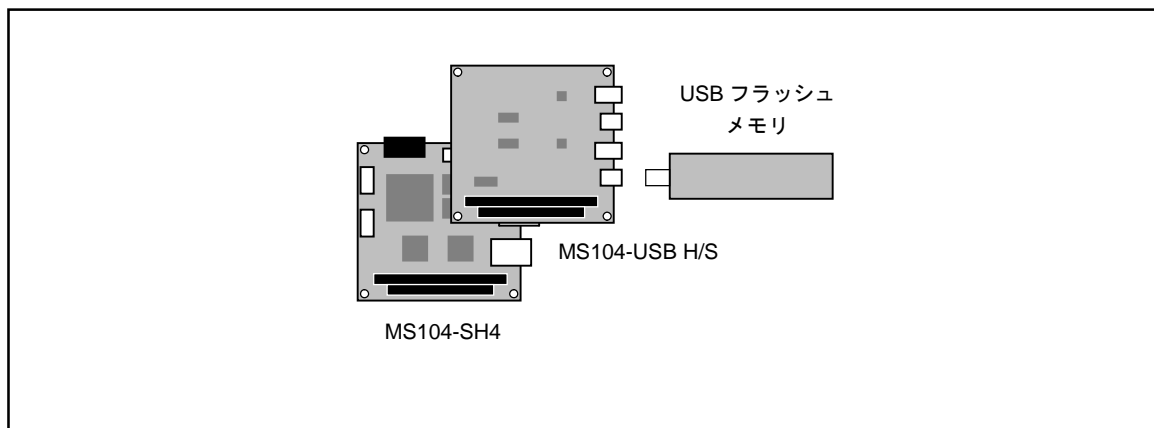


Fig 2.2-1 USB フラッシュメモリとの接続

下図に USB キーボード・マウスの接続例を示します。

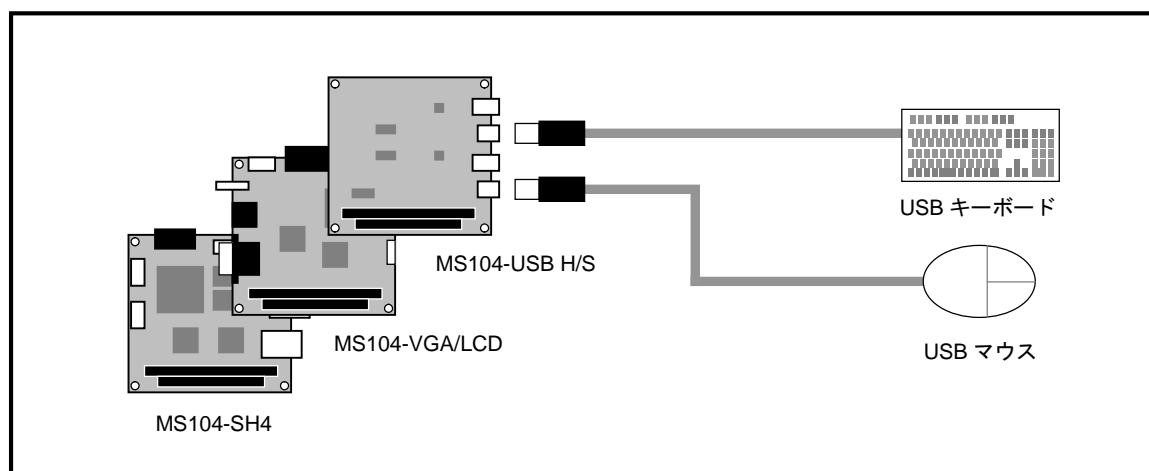


Fig 2.2-2 USB キーボード・マウスとの接続

2.3 デバイスドライバ

Linux

Linux の USB ドライバは3階層に分かれています。最上位層は各 USB ドライバを制御する USB デバイスドライバ、中間層は各 USB デバイスドライバのロード・アンロードや USB デバイスドライバと最下位層の HCD との仲介を行う API を提供する USB コア、最下位層は実際 USB 転送などのハードウェアの制御を行う HCD (ホストコントローラドライバ) となります。

MS104-USB H/S は、最下位層の HCD のみ実装し、USB ホストを動作させています。MS104-USB H/S は USB ホストコントローラ「SL811HST」が2機搭載されており、apLinux 上からは USB ホストコントローラが二つ存在していることとなります。

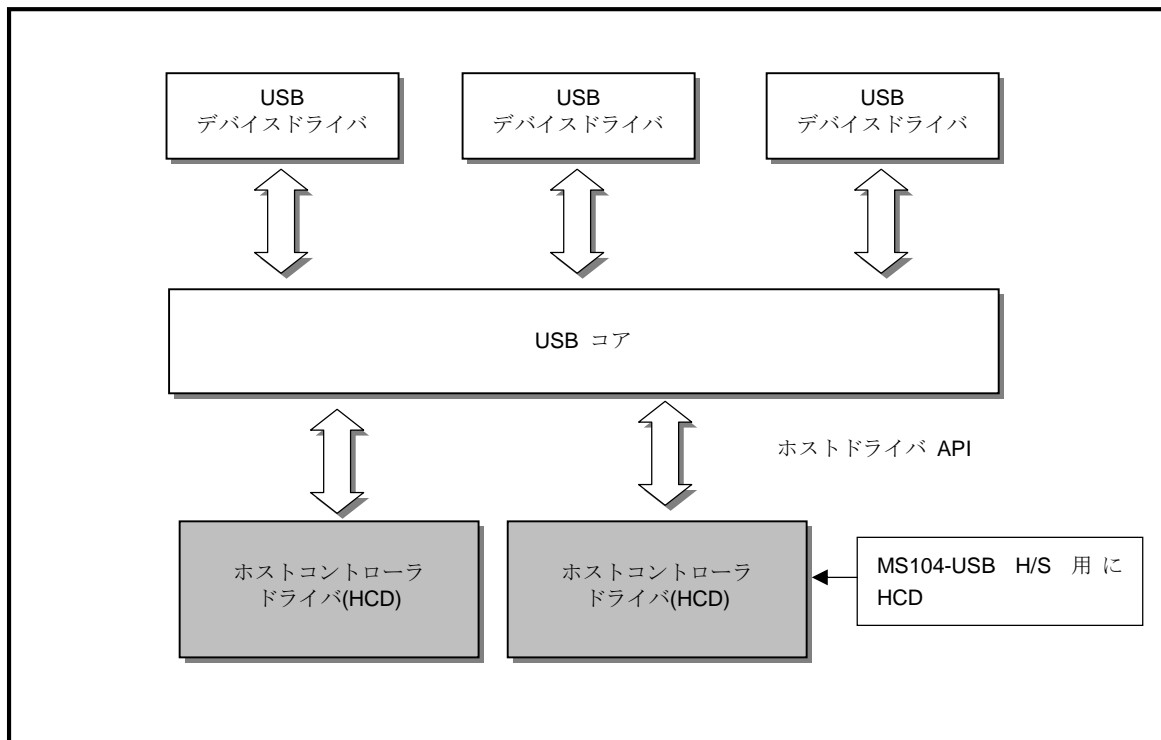


Fig 2.3-1 Linux USB ドライバ

Table 2.3-1 USB ホストコントローラデバイス

チャンネル	アドレス	割込み番号
CH1	0x1000	9
	0x1001	
CH2	0x1002	10
	0x1003	

Linux カーネルソースディレクトリ

```

|--drivers
  |--usb
    |-- hc_simple.c      :   URB ハンドリングと USB コアへのインターフェース
    |-- hc_simple.h     :   URB ハンドリングと USB コアへのインターフェースヘッダファイル
    |-- hc_sl811.c      :   USB パケットの送受信と割込み制御
    |-- hc_sl811.h     :   USB パケットの送受信と割込み制御用ヘッダファイル
    |-- hc_sl811rh.c   :   ルートハブルーチン
  
```

Fig 2.3-2 LinuxUSB ホストコントローラデバイスソースファイル構成

3. MS104-USB H/S の起動

『MS104-USB H/S』と『MS104-SH4』を使用して、USB デバイスを動作させる手順について説明します。

3.1 MS104-USB H/S の動作環境

apLinux が動作する MS104-SH4 ボードをご用意ください。

●ホスト PC

RedBoot/Linux のコンソール、および、TFTP、NFS サーバとして使用します。シリアルポート、ネットワーク、TFTP、NFS サーバが使用可能な PC をご用意ください。

●電源

MS104-USB H/S は PC/104 バスから電源の配給を受けることができます。MS104-SH4 に必要な電源は DC5V±5%です。MS104- USB H/S と合わせて使用するため、2A 程度の電源をご用意ください。

※USB に接続するデバイスにより、最大 1A まで電源を消費します。接続する USB デバイスに合わせて電源をお選びください。

●LAN

NFS を使用してホスト PC と MS104-SH4 でデータのやり取りを行います。MS104-SH4 をネットワークに接続できる LAN ケーブルをご用意ください。

●USB フラッシュメモリ

Linux 上での USB ホストの機能を検証するために USB フラッシュメモリを使用します。USB フラッシュメモリは USB マスストレージクラス対応で 256Mbyte 以下のものをご用意ください。

Table 3.1-1 MS104-USB H/S、MS104-SH4 の推奨動作環境

使用機器等	環 境
PC/104 USB ボード	MS104-USB H/S
CPU ボード	MS104-SH4
HOST PC	PC/AT 互換機
OS	Linux(推奨 FedoraCore1)
メモリ	使用 OS による
ソフトウェア	ターミナルソフト TFTP サーバ NFS サーバ
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
USB フラッシュメモリ	256Mbyte 以下の容量 ※1 USB マスストレージクラス対応
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
VGA モニタ & ケーブル	垂直周波数 60.1Hz 水平周波数 29.5kHz 対応モニタ
電源	DC5V±5% 2A 程度

※1 現在 Linux 上では USB ハードディスクなど容量が大きいストレージデバイスを接続するとディレクトリ探索 (ls 等のコマンド) が非常に遅いという現象が見られます。現在、弊社で動作確認しているのは 256Mbyte 以下の USB フラッシュメモリです。

3.2 MS104-USB H/S の設定

Linux 用に MS104-USB H/S ボードの設定を行います。

- ① I/O アドレスを H'1000 に設定します。SW1 の 4 番のみ OFF にして下さい。

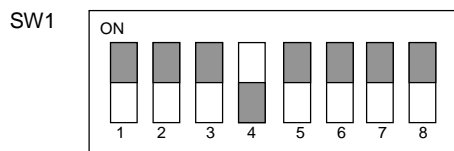


Fig 3.2-1 SW1 の設定

- ② CH1 をホストの設定にします。JP5 の 1 番と 2 番にジャンパピンを接続します。

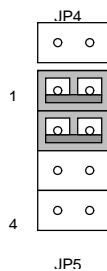


Fig 3.2-2 JP5 の設定

- ③ CH1 の割り込みを PC/104 バス IRQ6 (Linux 上での割り込み番号 9) に設定します。JP3 の CH1 を IRQ6 にジャンパピンで接続します。

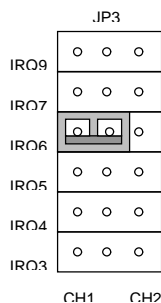


Fig 3.2-3 JP3 の設定 (その 1)

- ④ CH2 をホストの設定にします。JP7 の 1 番と 2 番にジャンパピンを接続します。

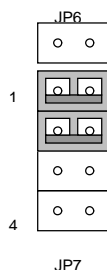


Fig 3.2-4 JP7 の設定

- ⑤ CH2 の割込みを PC/104 バス IRQ5 (Linux 上での割込み番号 10) に設定します。JP3 の CH2 を IRQ5 にジャンパピンで接続します。

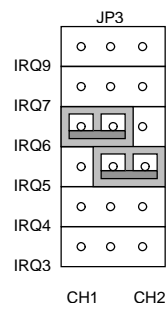


Fig 3.2-5 JP3 の設定 (その 2)

3.3 USB デバイスの使用

apLinux では USB デバイスとして HID (Human Interface Device) とストレージデバイスを扱うことができます。以下では、各 USB デバイス (USB マウス・キーボード、USB フラッシュメモリ・ハードディスク) の用法について説明します。

MS104-USB H/S ホストコントローラ対応 Linux カーネル

各種 USB デバイスを動作させる前に、MS104-USB H/S ホストコントローラに対応した Linux カーネルを MS104-SH4 搭載のフラッシュROMに格納しておきます。

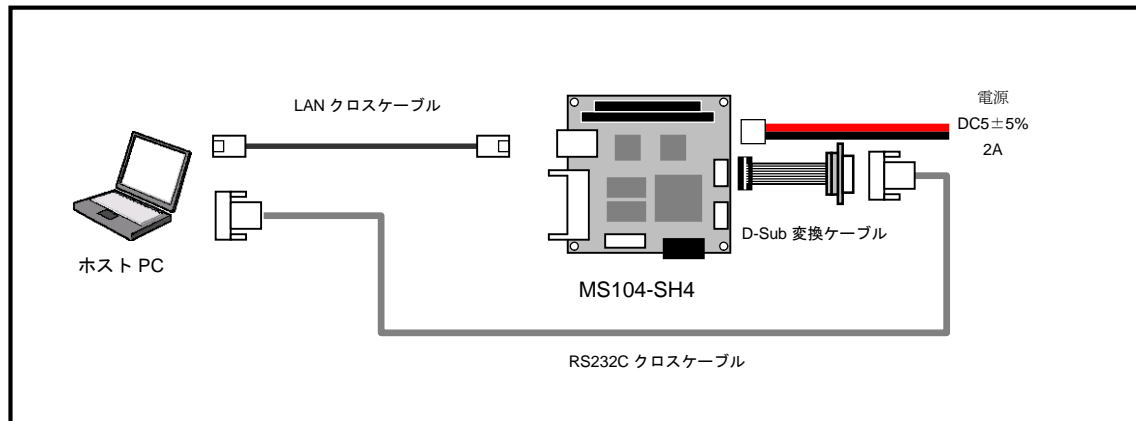


Fig 3.3-1 MS104-SH4 の接続

- ① Linux 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

- ② 「MS104-SH4」の添付 CD-ROM、もしくは、「MS104-SH4 ファイルセンター」から MS104-USB H/S ホストコントローラ用 Linux カーネルと RAM ディスクイメージ (ルートファイルシステム) を取得し、TFTP サーバディレクトリ『/tftpboot』にコピーします。

下記のコマンドは「MS104-SH4」の添付 CD-ROM から Linux カーネルと RAM ディスクイメージ (ルートファイルシステム) をコピーする場合は。

```
[alpha@ap_linux root]# mount /dev/cdrom /mnt/cdrom 
[alpha@ap_linux root]# cp /mnt/cdrom/linux/ms104usb/vmlinuz-ms104sh4-x.x-usb /tftpboot/ 
[alpha@ap_linux root]# cp /mnt/cdrom/linux/ms104usb/ramdisk-ms104sh4-x.x-usb.gz /tftpboot/ 
```

- ※ MS104-SH4 Linux導入マニュアル『MS104-SH4 Linuxstart.pdf』添付CD-ROMもしくは弊社MS104-SH4 ファイルセンター『<http://www.apnet.co.jp/e-linux/dl/filecenter/index.html>』から入手することができます。
- ※ MS104-SH4 ファイルセンター『<http://www.apnet.co.jp/e-linux/dl/filecenter/index.html>』にアクセスするには、ユーザ登録『https://www.apnet.co.jp/cgi-bin/ms104_reg/index.html』が必要です。
- ※ 「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 MS104-USB H/S と MS104-SH4 を「Fig 3.3-1 MS104-SH4 の接続」を参考に接続します。
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、自動起動を停止するため『Ctrl+C』とタイ
プします。

```
+Ethernet eth0: MAC address 00:0c:7b:xx:xx:xx
IP: 192.168.1.200/255.255.255.0, Gateway: 0.0.0.0
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 20:25:22, Oct  3 2003

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

RAM: 0x8c000000-0x8e000000, 0x8c00f0b8-0x8dfdd000 available
FLASH: 0x80000000 - 0x81000000, 128 blocks of 0x00020000 bytes each.
== Executing boot script in 5.000 seconds - enter ^C to abort
^C
RedBoot>
```

- ④ RedBoot に Linux カーネルイメージをダウンロードし、フラッシュROMに格納します。

TFTP サーバの IP アドレスは「192.168.1.201」と仮定します。

```
RedBoot> load -r -b 0x8c210000 -m tftp -h 192.168.1.201 vmlinuz-ms104sh4-x.x-usb
Raw file loaded 0x8c210000-0x8c30c120, assumed entry at 0x8c210000
RedBoot> fis create -b 0x8c210000 -l 0x100000 -r 0x8c210000 -e 0x8c210000 -f 0x80040000 vmlinuz
An image named 'vmlinuz' exists - continue (y/n)? y
... Erase from 0x80040000-0x80140000: .....
... Program from 0x8c210000-0x8c310000 at 0x80040000: .....
... Unlock from 0x80fe0000-0x81000000: .
... Erase from 0x80fe0000-0x81000000: .
... Program from 0x8dfdf000-0x8dfff000 at 0x80fe0000: .
... Lock from 0x80fe0000-0x81000000: .
RedBoot>
```

※ 「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。

- ⑤ RedBoot に RAM ディスクイメージをダウンロードし、フラッシュROMに格納します。

TFTP サーバの IP アドレスは「192.168.1.201」と仮定します。

```
RedBoot> load -r -b 0x8c360000 -m tftp -h 192.168.1.201 ramdisk-ms104sh4-x.x-usb.gz
Raw file loaded 0x8c360000-0x8c708320, assumed entry at 0x8c360000
RedBoot> fis create -b 0x8c360000 -l 0x400000 -r 0x8c360000 -e 0x8c360000 -f 0x80140000 ramdisk.gz
An image named 'ramdisk.gz' exists - continue (y/n)? y
... Erase from 0x80140000-0x80540000: .....
... Program from 0x8c360000-0x8c760000 at 0x80140000: .....
... Unlock from 0x80fe0000-0x81000000: .
... Erase from 0x80fe0000-0x81000000: .
... Program from 0x8dfdf000-0x8dfff000 at 0x80fe0000: .
... Lock from 0x80fe0000-0x81000000: .
RedBoot>
```

※ 「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。

- ⑥ RedBoot の『**fconfig**』コマンドを使用して、『**libboot**』コマンドで Linux カーネルが起動するよう RedBoot のコンフィグレーションデータを変更します。

```
RedBoot> fconfig
Run script at boot: false
Use BOOTP for network configuration: false
Gateway IP address: 0.0.0.0
Local IP address: 192.168.1.200
Local IP address mask: 255.255.255.0
Default server IP address: 0.0.0.0
Console baud rate: 38400
DNS server IP address: 0.0.0.0
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
Modify Linux Kernel Parameter: false
Use VGA/LCD monitor(MS104-VGA/LCD): false
Network debug at boot time: false
Update RedBoot non-volatile configuration - continue (y/n)? y
... Unlock from 0x80fc0000-0x80fd0000: .
... Erase from 0x80fc0000-0x80fd0000: .
... Program from 0x8dfcf000-0x8dfdf000 at 0x80fc0000: .
... Lock from 0x80fc0000-0x80fd0000: .
RedBoot>
```

- ⑦ 『**libboot**』コマンドを使用して、Linux が起動するか確認します。

```
RedBoot> libboot
Now loading Linux kernel:
Linux kernel source address : 0x80040000
Linux kernel destination address : 0x8c210000
Linux kernel size : 0x00100000
.
.
Uncompressing Linux... Ok, booting the kernel.
.
.
```

USB デバイスファイルシステム

USB デバイスファイルシステムは USB の状態を `/proc` ファイルに出力します。主にデバッグ用ツールとして用いられます。下記に、MS104-USB H/S と MS104-SH4 の接続を示します。

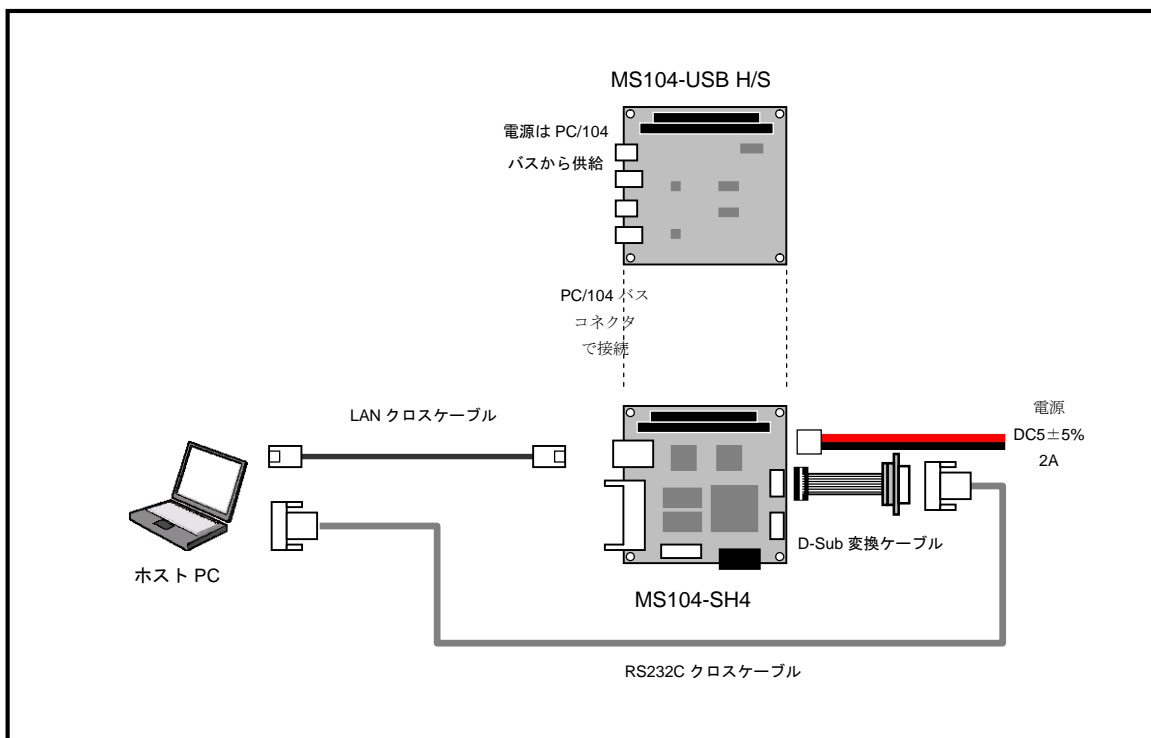


Fig 3.3-2 MS104-SH4 との接続

- ① MS104-USB H/S と MS104-SH4 を「Fig 3.3-2 MS104-SH4 との接続」を参考に接続します。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ることを確認します。

```
+Ethernet eth0: MAC address 00:0c:7b:xx:xx:xx
IP: 192.168.1.200/255.255.255.0, Gateway: 0.0.0.0
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 20:25:22, Oct  3 2003

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

RAM: 0x8c000000-0x8e000000, 0x8c00f0b8-0x8dfdd000 available
FLASH: 0x80000000 - 0x81000000, 128 blocks of 0x00020000 bytes each.
RedBoot>
```

- ② MS104-SH4 上で『libboot』コマンドを使用して、MS104-USB H/S ホストコントローラ対応 Linux カーネルを起動します。

```
RedBoot> libboot
Now loading Linux kernel:
Linux kernel source address      : 0x80040000
Linux kernel destination address : 0x8c210000
Linux kernel size                 : 0x00100000
.
.
Uncompressing Linux... Ok, booting the kernel.
.
```

- ③ Linux の起動を確認し、root 権限でログインします。

```
MS104SH4 login: root
```

- ④ USB デバイスファイルシステムは起動スクリプト内で『/proc/bus/usb』ディレクトリマウントされます。
/proc/bus/usb ディレクトリの中身を確認します。

```
[MS104SH4@root]# cd /proc/bus/usb
MS104SH4@root]# ls
001/  002/  devices  drivers
```

- ⑤ Linux カーネルに組み込まれている USB デバイスドライバを確認します。

```
[MS104SH4@root]# cat drivers
usbdevfs
hub
usb_mouse
keyboard
usb-storage
serial
```

- ⑥ USB ホストコントローラデバイスを確認します。

```
[MS104SH4@root]# cat devices
T: Bus=02 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 1
B: Alloc= 0/900 us (0%), #Int= 0, #Iso= 0
D: Ver= 1.10 Cls=09(hub ) Sub=00 Prot=00 MxPS= 8 #Cfgs= 1
P: Vendor=0000 ProdID=0000 Rev= 0.00
S: Product=USB SL811HS Root Hub
S: SerialNumber=0
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 2 Ivl=255ms
T: Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh= 1
B: Alloc= 0/900 us (0%), #Int= 0, #Iso= 0
D: Ver= 1.10 Cls=09(hub ) Sub=00 Prot=00 MxPS= 8 #Cfgs= 1
P: Vendor=0000 ProdID=0000 Rev= 0.00
S: Product=USB SL811HS Root Hub
S: SerialNumber=0
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
I: If#= 0 Alt= 0 #EPs= 1 Cls=09(hub ) Sub=00 Prot=00 Driver=hub
E: Ad=81(I) Atr=03(Int.) MxPS= 2 Ivl=255ms
```

『/proc/bus/usb/devices』ファイルの書式の詳細については <http://www.linux-usb.org/USB-guide/book1.html> をご覧下さい。

USB マスストレージデバイス

USB マスストレージデバイスは FDD、HDD や CD-ROM など外部記憶装置をサポートします。USB マスストレージクラスは外部記憶装置をサポートするための USB 規格として制定されています。

Linux 上では USB マスストレージデバイスを SCSI デバイスとして認識します。

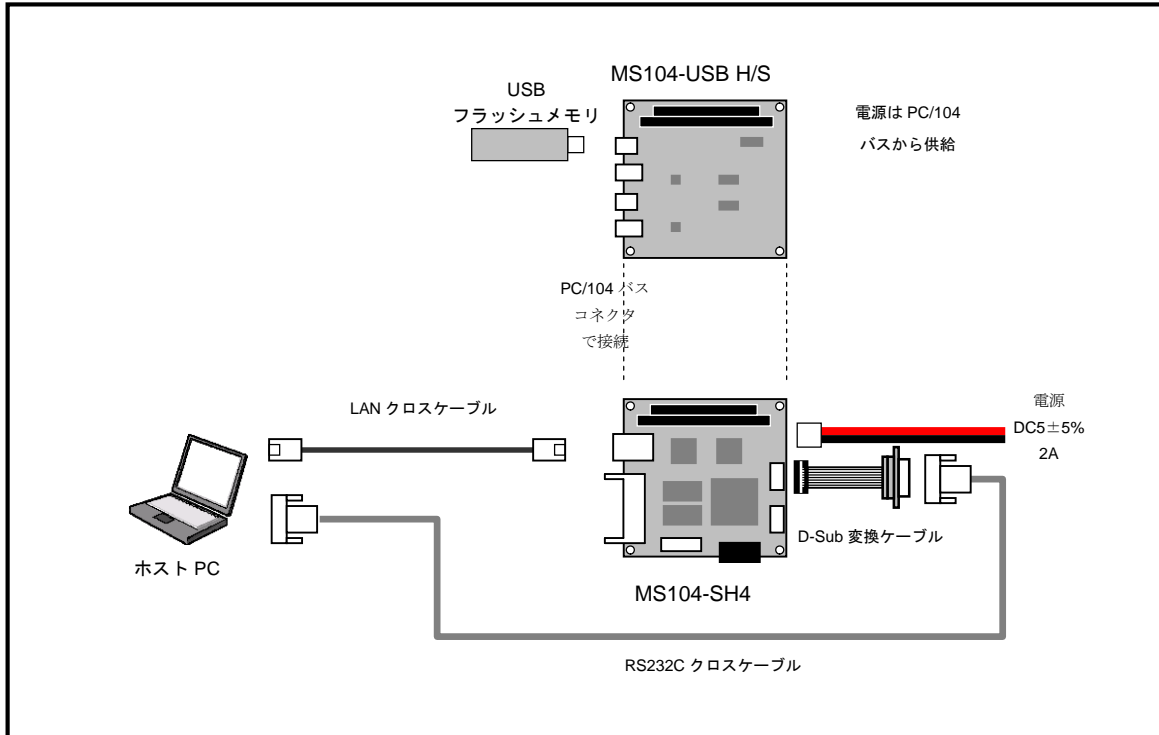


Fig 3.3-3 USB フラッシュメモリとの接続

- ① MS104-USB H/S と MS104-SH4 を「Fig 3.3-3 USB フラッシュメモリとの接続」を参考に接続します。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ることを確認します。

```
+Ethernet eth0: MAC address 00:0c:7b:xx:xx:xx
IP: 192.168.1.200/255.255.255.0, Gateway: 0.0.0.0
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 20:25:22, Oct  3 2003

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

RAM: 0x8c000000-0x8e000000, 0x8c00f0b8-0x8dfdd000 available
FLASH: 0x80000000 - 0x81000000, 128 blocks of 0x00020000 bytes each.
RedBoot>
```

- ② 『libboot』 コマンドを使用して、MS104-USB H/S ホストコントローラ対応 Linux カーネルを起動します。

```
RedBoot> libboot
Now loading Linux kernel:
Linux kernel source address      : 0x80040000
Linux kernel destination address : 0x8c210000
Linux kernel size                 : 0x00100000
.
.
Uncompressing Linux... Ok, booting the kernel.
.
.
```

- ③ Linux の起動を確認し、root 権限でログインします。

```
MS104SH4 login: root
```

- ④ MS104-USB H/S のコネクタシリーズ A に FAT 形式でフォーマットされた USB フラッシュメモリを挿入してください。

接続されたデバイスファイルが 『/dev/sda1』 であることを確認します。

```
[MS104SH4@root]# hub.c: USB new device connect on bus2, assigned device number 2
scsi0 : SCSI emulation for USB Mass Storage devices
  Vendor: HAGIWARA   Model: UD-Pure       Rev: 1.00
  Type:   Direct-Access          ANSI SCSI revision: 02
Attached scsi removable disk sda at scsi0, channel 0, id 0, lun 0
SCSI device sda: 243712 512-byte hdwr sectors (125 MB)
sda: Write Protect is off
Partition check:
sda: sda1
```

USB フラッシュ
メモリ挿入時の
メッセージ

- ⑤ FAT 形式で USB フラッシュメモリをマウントします。

マウント時にエラーメッセージが出力されなければ、通常のディスクとして扱うことができます。

```
[MS104SH4@root]# mkdir /mnt/usbfs1
[MS104SH4@root]# mount -t vfat /dev/sda1 /mnt/usbfs1
```

※ USB フラッシュメモリは活栓挿抜に対応しています。USB フラッシュメモリを抜く際は必ずアンマウントを実行してください。

```
[MS104SH4@root]# umount /dev/sda1
[MS104SH4@root]# usb.c: USB disconnect on device 2 ← USB フラッシュメモリ抜いた場合のメッセージ
```

- ⑥ USB フラッシュメモリを挿したまま、USB ハードディスクを MS104-USB H/S に接続します。

USB マスストレージデバイスを二個接続すると次に接続したデバイスは 『/dev/sdb1』 になります。

```
[MS104SH4@root]# hub.c: USB new device connect on bus1/1, assigned device number
3
scsi1 : SCSI emulation for USB Mass Storage devices
  Vendor: USB-FS     Model: SAMSUNG SV8004H   Rev: 0.01
  Type:   Direct-Access          ANSI SCSI revision: 02
Attached scsi disk sdb at scsi1, channel 0, id 0, lun 0
SCSI device sdb: 156368016 512-byte hdwr sectors (80060 MB)
sdb: sdb1
```


USB キーボード・USB マウス

USB キーボード・USB マウスは USB クラスのヒューマン インターフェイス デバイス (HID) クラスに属しています。HID はキーボードやマウスといった、人が直接操作するデバイスについての規格を定義しています。

Linux はもともと、英語圏での使用が前提となっているため、通常キーボードは英語 101 キーボードを使用します。

USB キーボード・USB マウスの動作を確認するため、組み込み向けウィンドウシステム「Microwindows」を使用します。

Microwindows 上では USB マウスを PS/2 マウスとして扱います。そのため、USB マウスには『/dev/psaux』デバイスファイルを通じてアクセスします。『/dev/psaux』デバイスファイルのメジャー番号は 13、マイナー番号は 63 になります。

※ 現在、「Microwindows」にはキーボードの使用に際して、不具合があり、方向キーおよびファンクションキーの使用、「Ctrl+C」を使用できません。

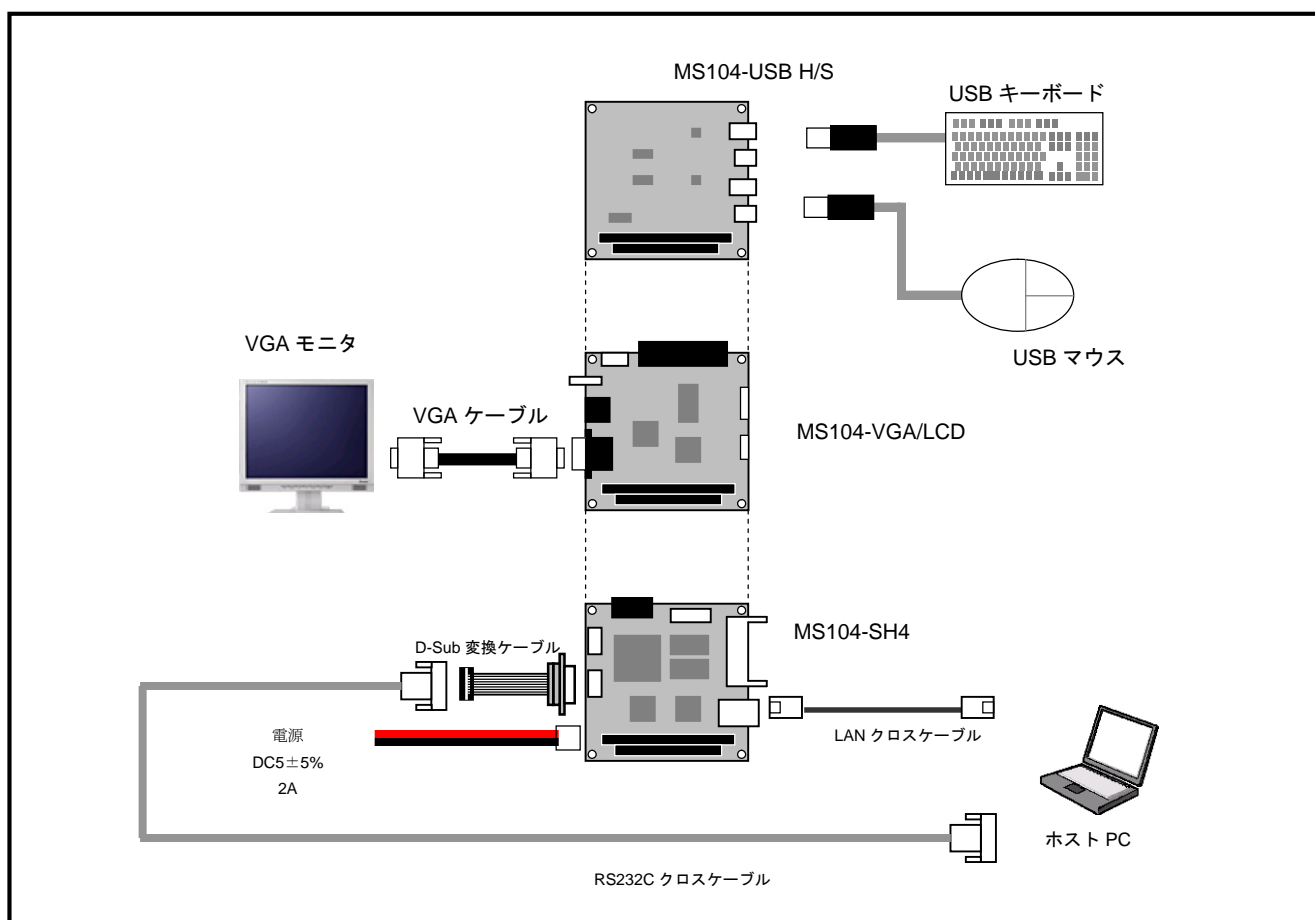


Fig 3.3-4 USB キーボード・USB マウスとの接続

- ① MS104-USB H/S と MS104-SH4 を「Fig 3.3-4 USB キーボード・USB マウスとの接続」を参考に接続します。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ることを確認します。

```
+Ethernet eth0: MAC address 00:0c:7b:xx:xx:xx
IP: 192.168.1.200/255.255.255.0, Gateway: 0.0.0.0
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 20:25:22, Oct  3 2003

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

RAM: 0x8c000000-0x8e000000, 0x8c00f0b8-0x8dfdd000 available
FLASH: 0x80000000 - 0x81000000, 128 blocks of 0x00020000 bytes each.
RedBoot>
```

- ② RedBoot の『fconfig』コマンドを使用して、MS104-VGA/LCD の VGA 出力に対応するよう RedBoot のコンフィグレーションデータを変更します。

『fconfig ms104vga true』を実行して、RedBoot の MS104-VGA/LCD サポート機能を有効にします。

```
RedBoot> fconfig ms104vga true 
ms104vga: Setting to true
Update RedBoot non-volatile configuration - continue (y/n)? y 
... Unlock from 0x80fc0000-0x80fd0000: .
... Erase from 0x80fc0000-0x80fd0000: .
... Program from 0x8dfcf000-0x8dfdf000 at 0x80fc0000: .
... Lock from 0x80fc0000-0x80fd0000: .
RedBoot>
```

- ③ fconfig ms104vga_bpp 8 を実行して、色深度を 8bpp に変更します。

```
RedBoot> fconfig ms104vga_bpp 8 
.
RedBoot>
```

- ④ fconfig ms104vga_res 800600 を実行して、解像度を 800×600 に変更します。

```
RedBoot> fconfig ms104vga_res 800600 
.
RedBoot>
```

- ⑤ fconfig ms104vga_output_vga true を実行して、VGA 出力を有効にします。

```
RedBoot> fconfig ms104vga_output_vga true 
.
RedBoot>
```

- ⑥ VGA 以外の出力とその他の機能を無効にします。

```
RedBoot> fconfig ms104vga_custom_pc104 false
.
RedBoot> fconfig ms104vga_output_lcd false
.
RedBoot> fconfig ms104vga_output_lcd_sim false
.
RedBoot> fconfig ms104vga_output_ntsc false
.
RedBoot> fconfig ms104vga_output_svideo false
.
RedBoot>
```

- ⑦ 『fconfig -l -n』で MS104-VGA/LCD サポート機能の設定を確認します。

```
RedBoot> fconfig -l -n
boot_script: false
.
kernel_para: false
ms104vga: true
ms104vga_bpp: 8
ms104vga_custom_pc104: false
ms104vga_output_lcd: false
ms104vga_output_lcd_sim: false
ms104vga_output_ntsc: false
ms104vga_output_svideo: false
ms104vga_output_vga: true
ms104vga_res: 800600
net_debug: false
RedBoot>
```

← MS104-VGA/LCD サポート
← 色深度設定 (8 or 15 or 16)
← カスタム PC/104 バスの使用
← LCD 出力サポート
← LCD 同時出力サポート
← NTSC コンポジット出力サポート
← S-Video 出力サポート
← VGA 出力サポート
← 解像度設定 (800600 or 640480)

※ LCD、NTSC、S-Video、VGA 出力のサポート設定は同時に有効にすると正しく動作しない恐れがあります。必ずいずれか一つのみを有効にしてください。

- ⑧ 『liboot』コマンドを使用して、MS104-USB H/S ホストコントローラ対応 Linux カーネルを起動します。

```
RedBoot> liboot
Now loading Linux kernel:
Linux kernel source address      : 0x80040000
Linux kernel destination address : 0x8c210000
Linux kernel size                 : 0x00100000
.
.
Uncompressing Linux... Ok, booting the kernel.
.
.
```

- ⑨ Linux の起動を確認し、root 権限でログインします。

```
MS104SH4 login: root
```

- ⑩ Microwindows 用端末『**nxterm**』を起動します。

```
[MS104SH4@root]# nano-X | nanowm | nxterm
```

下記のように画面が立ち上がります。USB キーボード・USB マウスの動作を確認してください。

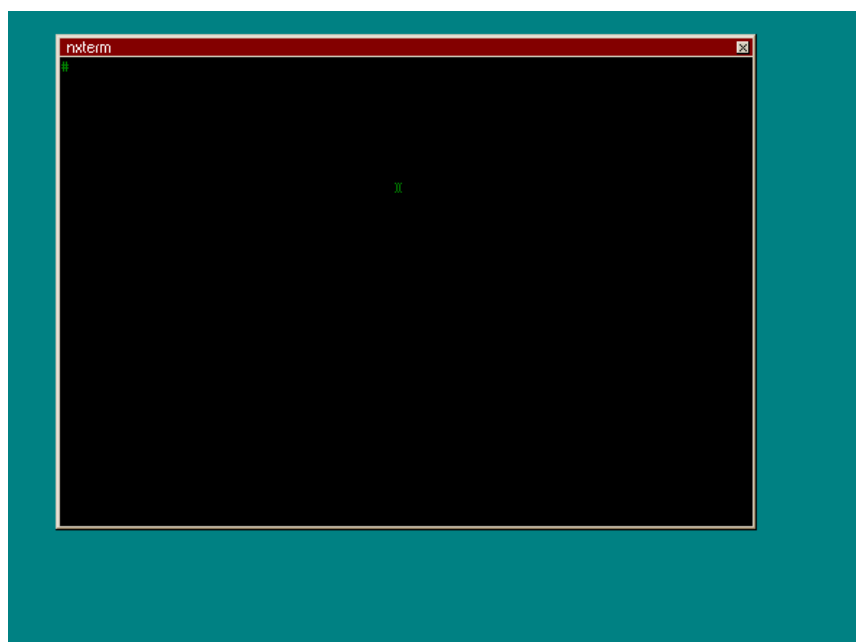


Fig 3.3-5 nxterm の起動

4. Linux

この章では、USB ホストに対応した Linux カーネルの作成から、USB マウスに対応した GUI アプリケーションの作成までの手順を説明します。GUI 環境として『Microwindows』を使用します。

GUI アプリケーションの作成までの流れは以下のようになります。

- ① Linux カーネルの作成
- ↓
- ② Microwindows のインストール
- ↓
- ③ ルートファイルシステムの作成
- ↓
- ④ サンプルプログラムの作成

4.1 Linux カーネルの概要

MS104-USB H/S を使用するためには、Linux カーネルに各 USB デバイスに対応したドライバを追加する必要があります。デバイスドライバの追加は X-Window もしくはテキストベースのコンフィグレータにより行います。また、Linux のコンフィグレータでは以前行ったコンフィグレーションを記録することができます。記録したコンフィグレーションファイル呼び出すことにより、再度コンフィグレーションをやり直すことができます。

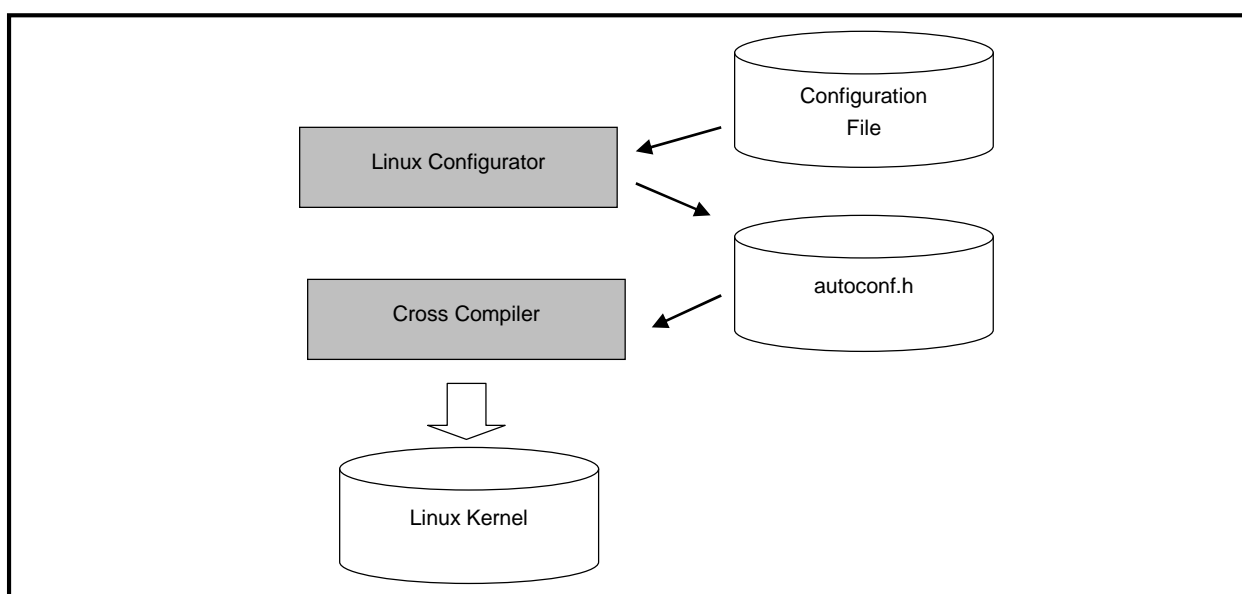


Fig 4.1-1 Linux の build イメージ

4.2 カーネルのコンフィグレーション

Linux コンフィグレータの起動

Linux カーネルは X-Window もしくはテキストベースのコンフィグレータにより、コンフィグレーションを行うことができます。

MS104-SH4 用 Linux のコンフィグレータを起動するには、『/opt/alpha_project/linux-kit-a02/linux/kernel/celinux-040503-alp』ディレクトリに移動し、『make menuconfig』を実行してください。

```
[root@ap_linux root]# cd /opt/alpha_project/linux-kit-a02/linux/kernel/celinux-040503-alp
[root@ap_linux celinux-040503-alp]# make menuconfig
```

● Linux コンフィグレーション画面



Linux コンフィグレーション項目について

MS104-SH4 用 Linux カーネルのコンフィグレーションを下記に示します。

『/opt/alpha_project/linux-kit-a02/linux/kernel/celinux-040503-alp/arch/sh/def-configs/ms104-sh4/ms104usb.config』

ディレクトリ下にコンフィグレーションファイルが保存されています。

Table 4.2-1 MS104-USB H/S 用 Linux コンフィグレーション

設定項目	設定	説明
SCSI support		SCSI に関するカテゴリ
SCSI support	M ※1	SCSI のサポート
SCSI disk support	M	SCSI ディスク (USB IDE ドライバ) のサポート
SCSI tape support	M	SCSI テープデバイスのサポート
SCSI CD-ROM support	M	SCSI CD-ROM のサポート
SCSI generic support	M	SCSI スキャナ、CD ライター等のサポート
Probe all LUNs on each SCSI device	* ※2	複数 LUN (Logical Unit Number) のサポート
Input core support		入力デバイスに関するカテゴリ
Input core support	M	USB Human Interface Device (HID) のサポート
Keyboard support	M	USB キーボードをシステムキーボードとしてサポート
Mouse support	M	USB マウスを疑似 PS/2 マウスとしてサポート
Horizontal screen resolution	(800)	マウスの水平スクリーン分解能
Vertical screen resolution	(600)	マウスの垂直スクリーン分解能
Character devices		シリアルポート、マウス等キャラクタ型デバイスに関するカテゴリ
Virtual terminal	*	ディスプレイ、キーボードデバイスをターミナルとしてサポート
Support for console on virtual terminal	*	ディスプレイをコンソールとして使用
Console drivers		コンソールデバイスに関するカテゴリ
Frame-buffer support	-	フレームバッファのサポート
Support for frame buffer devices	*	フレームバッファデバイスのサポート
Epson S1D13506/13806 framebuffer support	*	EPSON グラフィックコントローラのサポート
Use CRT display	*	VGA モニタ出力
Use Composite NTSC (TV) display	*	TV モニタ出力
Use TFT Panel display	*	TFT 型 LCD モニタ出力
Advanced low level driver options	*	フレームバッファデバイスのハードウェアに関する設定
8 bpp packed pixels support	*	8 ビット色深度のサポート
16 bpp packed pixels support	*	16 ビット色深度のサポート
Select compiled-in fonts	*	フォントの組込みのサポート
VGA 8x8 font	*	8×8 フォントサポート
VGA 8x16 font	*	8×16 フォントサポート
USB support		USB デバイスに関するカテゴリ
Support for USB		
Miscellaneous USB options	-	各種 USB オプション
Preliminary USB device filesystem	M	/proc/bus/usb ファイルシステムのサポート

USB Host Controller Drivers	-	USB ホストコントローラデバイスの選択
SL811HS support	M	SL811HS USB ホストコントローラのサポート
USB host support for MS104-USB H/S CH1	*	MS104-USB H/S のチャンネル 1 (CH1) をホストコントローラとして使用
USB host support for MS104-USB H/S CH2	*	MS104-USB H/S のチャンネル 2 (CH2) をホストコントローラとして使用
USB Device Class drivers	-	USB デバイスクラスドライバのサポート
USB Mass Storage support	M	USB マスストレージクラスのサポート
Datafab MDCFEB Compact Flash Reader support	*	Datafab 社製 MDCFEB CF リーダサポート
Freecom USB/ATAPI Bridge support	*	Freecom 社製 USB/ATAPI ブリッジアダプタサポート
ISD-200 USB/ATA Bridge support	*	In-Systems Design 社製 ISD-200 USB/ATA ブリッジサポート
Microtech CompactFlash /SmartMedia support	*	Microtech 社製 CompactFlash/SmartMedia リーダサポート
HP CD-Writer 82xx support	*	HP 社製 CD-Writer サポート
SanDisk SDDR-09 (and other SmartMedia) support	*	SanDisk 社製 SDDR-09 SmartMedia リーダサポート
Lexar Jumpshot Compact Flash Reader	*	Lexar 社製 Jumpshot Compact Flash リーダサポート
USB Human Interface Devices (HID)	-	USB HID のサポート
USB HIDBP Keyboard (basic) support	M	USB キーボードサポート
Japanese 106 keyboard or English 101 keyboard	JP106 or US101	日本語 106 キーボードまたは英語 101 キーボードの選択
USB HIDBP Mouse (basic) support	M	USB マウスサポート
USB Serial Converter support	-	USB シリアルコンバータサポート
USB Serial Converter support	M	USB シリアルコンバータサポート
USB FTDI Single Port Serial Driver	M	FTDI 社製 USB シリアルポートドライバのサポート

※1 「M」はコンフィグレーションで選択した機能を Linux カーネルに静的に組み込むことを示しています。

※2 「*」はコンフィグレーションで選択した機能をモジュールとしてコンパイルすることを示しています。

4.3 アドレス・割込み番号の設定

MS104-USB H/S の CH1 と CH2 は LinuxUSB ホストドライバとしてそれぞれアドレス、割込み番号が設定されます。各チャネルのアドレスと割込み番号を変更するには Linux ソースファイルを変更する必要があります。以下に USB ホストドライバのアドレス・割込み番号の設定箇所を示します。

■ USB ホストドライバ「drivers/usb/hc_sl811.c L98」

```
#ifndef CONFIG_SH_MS104SH4

# ifdef CONFIG_USB_MS104USBHS_CH1_HOST
static int base_io = 0x1000;
static int data_reg_io = 0x1001; } ← CH1 アドレス
static int irq = 9; ← CH1 割込み番号
# endif

# ifdef CONFIG_USB_MS104USBHS_CH2_HOST
static int base_io2 = 0x1002;
static int data_reg_io2 = 0x1003; } ← CH2 アドレス
static int irq2 = 10; ← CH2 割込み番号
# endif

#else
```

■ I/O アクセス関数「arch/sh/kernel/io_ms104sh4.c L232」

```
unsigned long ms104sh4_isa_port2addr(unsigned long offset){
    /* Ether (NIC) */
    if((offset & ~0x1f) == 0x1300)
        return NIC_OFFSET + offset;
    /* CompactFlash (IDE) */
    else if((offset & ~7) == 0x1f0 || offset == 0x3f6)
        return (unsigned long)cf_io_base + offset;
    /* USB */
    else if((offset & ~0x3) == 0x1000) ← アドレス
        return PC104_IO_8_ADDR + offset;
    /* PC104 I/O bus */
    return PC104_IO_ADDR + offset;
}
```

4.4 Linux カーネルのコンパイル

MS104-USB H/S USB ホスト対応 Linux カーネルのコンパイル方法について説明します。

- ① 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
```

- ② LinuxPC 上で MS104-SH4 用 Linux カーネルソースディレクトリに移動します。

```
[root@ap_linux root]# cd /opt/alpha_project/linux-kit-a02/linux/kernel/celinux-040503-alp/ 
[root@ap_linux celinux-040503-alp]#
```

- ③ Linux カーネルのコンフィグレータを起動し、コンフィグレーションを行います。

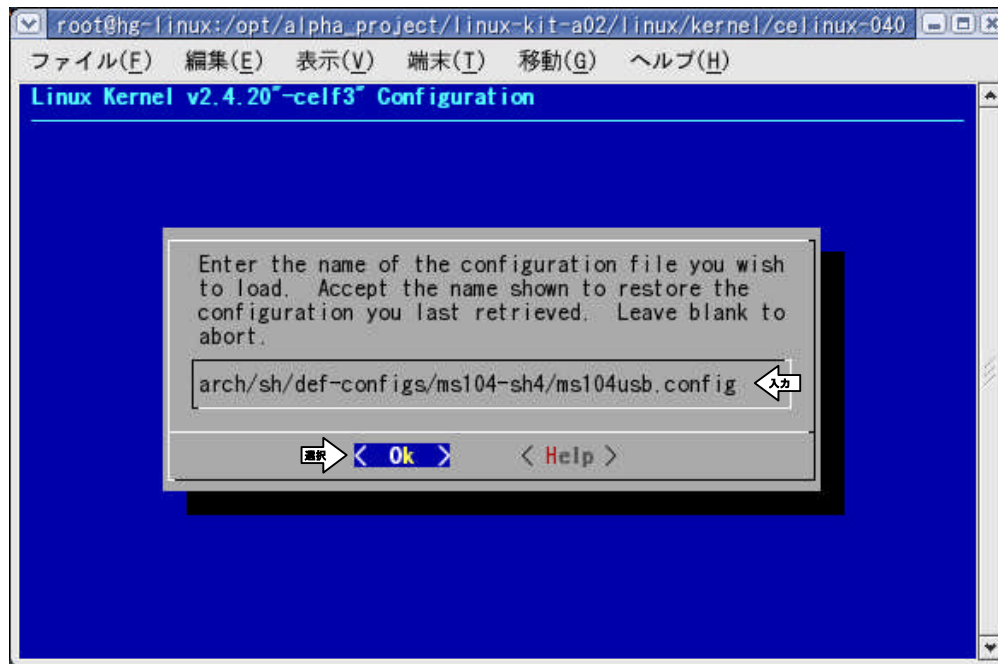
LinuxPC 上で『make menuconfig』を実行してください。

```
[root@ap_linux celinux-040503-alp]# make menuconfig 
```

- ④ MS104-USB H/S ホスト対応 Linux カーネルのコンフィグレーションを行います。。

『Load an Alternate Configuration File』を選択し、『arch/sh/def-configs/ms104-sh4/ms104usb.config』と入力した後、<OK> を選択します。

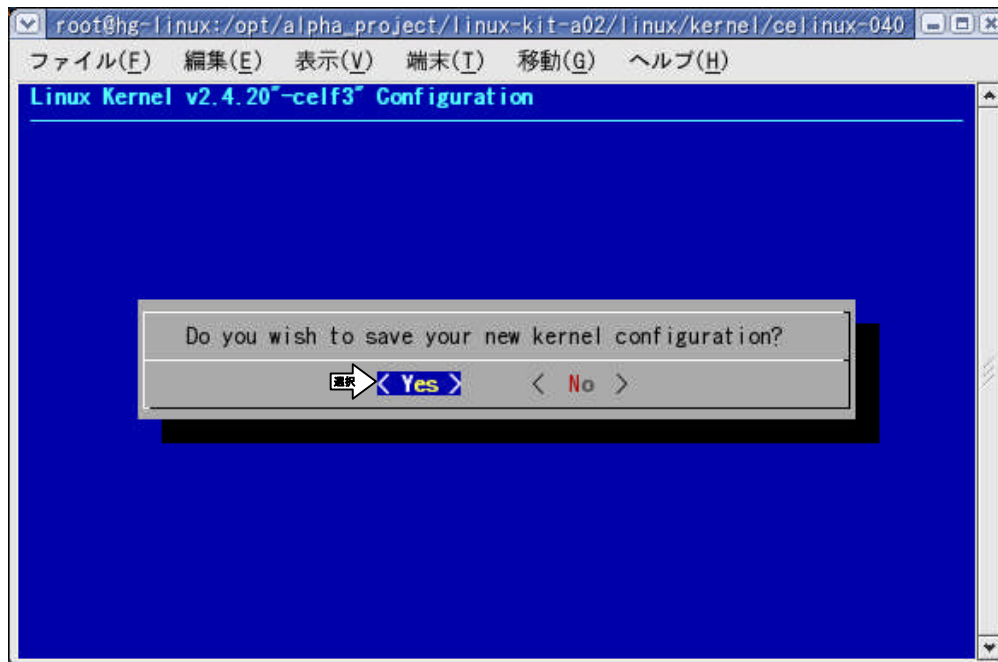




- ⑤ MS104-USB H/S ホスト対応 Linux カーネルのコンフィグレーションが完了したら、< Exit > を選択します。



『Do you wish to save your new kernel configuration?』と表示されるので < Yes > を選択します。



⑥ Linux カーネルの依存関係記述ファイルを更新します。

LinuxPC 上で『make dep』を実行してください。

```
[root@ap_linux celinux-040503-alp]# make dep
```

⑦ 前回コンパイルを実行したときの Linux カーネル中間ファイル等を削除する場合は『make clean』を実行します。

```
[root@ap_linux celinux-040503-alp]# make clean
```

⑧ Linux カーネルをコンパイルします。

LinuxPC 上で『make zimage』を実行してください。

```
[root@ap_linux celinux-040503-alp]# make zimage
```

※ コンパイルが正常終了すれば『arch/sh/boot』ディレクトリ下に Linux 圧縮カーネルイメージ『zimage』が作成されます。

⑨ モジュールのコンパイルを行います。

LinuxPC 上の『make modules』を実行してください。

```
[root@ap_linux celinux-040503-alp]# make modules
```

⑩ モジュールのインストールを行います。

LinuxPC 上の『make modules_install』を実行してください。

```
[root@ap_linux celinux-040503-alp]# make modules_install
```

- ⑩ インストール先ディレクトリを確認してください。
下記のようなディレクトリ構成になります。

Fig 4.4-1 モジュールインストールディレクトリ

```
/opt/alpha_project/linux-kit-a02/linux/filesystems/module(インストール先)ディレクトリ
|-- lib
    |-- modules
        |-- 2.4.20-celf3
            |-- build
            |-- kernel
                |-- drivers
                    |-- cdrom
                    |   |-- cdrom.o
                    |-- input
                    |   |-- input.o
                    |   |-- keybdev.o
                    |   |-- mousedev.o
                    |-- scsi
                    |   |-- osst.o
                    |   |-- scsi_mod.o
                    |   |-- sd_mod.o
                    |   |-- sg.o
                    |   |-- sr_mod.o
                    |-- usb
                    |   |-- hc_sl811.o
                    |   |-- serial
                    |       |-- ftdi_sio.o
                    |       |-- usbserial.o
                    |-- storage
                    |   |-- usb-storage.o
                    |-- usbcore.o
                    |-- usbkbd.o
                    |-- usbmouse.o
                |-- pcmcia
```

4.5 Microwindows の概要

Microwindows は Century Software 社のオープンソースのプロジェクトで、組み込み機器向けに設計されたウィンドウシステムです。Microwindows は 3 階層で設計されており、最下層はモニタやマウス、キーボードなどの入出力デバイスのドライバです。中間層は線描、領域塗りつぶし、クリッピング、カラーモデルなどのグラフィックエンジン、最上位層は 2 つの API をサポートしています。

API は Microwindows と呼ばれる Win32 API を模した API と Nano-X と呼ばれる X Window System の API を模した API の 2 種類です。Nano-X は Linux で採用されています。

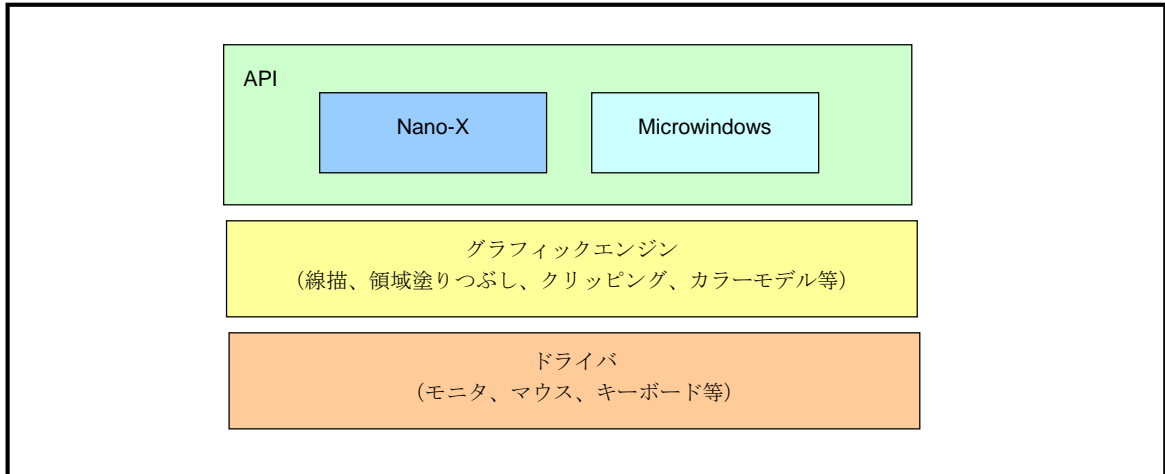


Fig 4.5-1 Microwindows アーキテクチャ

Nano-X は X Window System と同様にクライアント・サーバ型をとっています。そのため、アプリケーションプログラムを動作させる場合、Nano-X サーバを立ち上げ、アプリケーションプログラムとして Nano-X クライアントを立ち上げます。

Nano-X サーバはキーボードやモニター、マウスなどの入出力デバイスの処理を行い、クライアントは各アプリケーションの処理、つまり、ウィンドウの表示をするプログラムの処理を担当します。

Nano-X ではモニタ上でウィンドウがどのように見えるかや、マウスがウィンドウをどのように動かすかなど、見栄えや操作方法については定義されていません。そのため、ウィンドウマネージャと呼ばれるアプリケーションプログラムが見栄えや操作方法を定義、管理します。Nano-X には標準のウィンドウマネージャとして「nanowm」が添付されています。

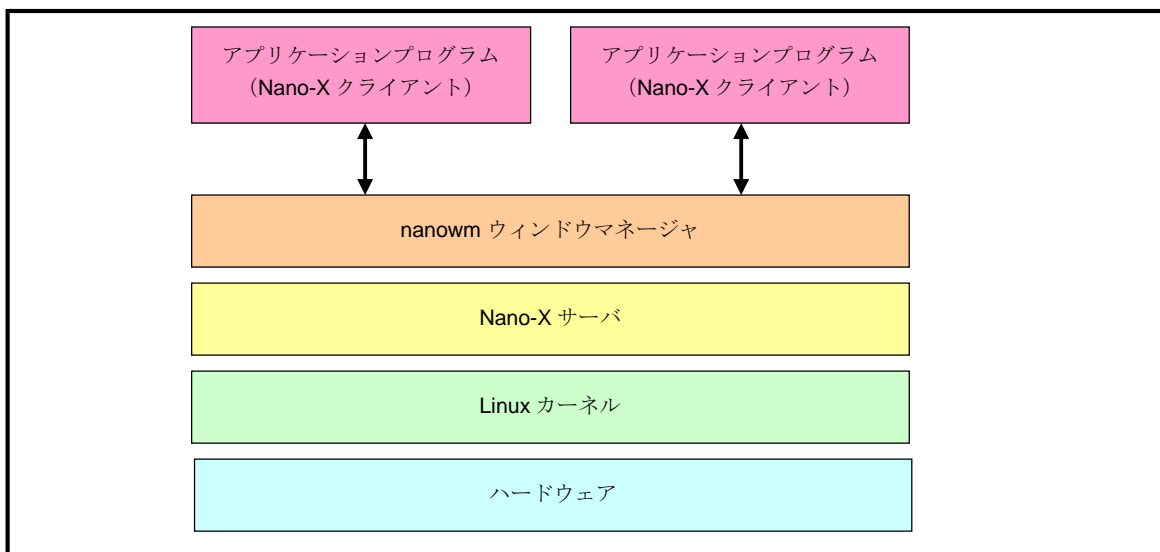


Fig 4.5-2 Microwindows システム構成

4.6 Microwindows のインストール

Microwindows はサポートデバイスや解像度・色深度により、クロス開発環境用ライブラリや MS104-SH4 用バイナリファイル等を変更する必要があります。USB マウス・キーボードを使用するには、USB マウス・キーボード対応の Microwindows をインストールしなければなりません。以前の Microwindows RPM ファイルをアンインストールし、USB マウス・キーボード対応の Microwindows RPM ファイルをインストールするのは非常に手間がかかります。そこで、この章では、Microwindows の開発環境構築をサポートするユーティリティプログラムを使用して、USB マウス・キーボード対応 Microwindows のインストールを行います。

Microwindows RPM ファイル

以下に USB 対応 Microwindows RPM ファイルの一覧を示します。

Table 4.6-1 USB 対応 Microwindows RPM ファイル一覧

サポート デバイス	MS104 シリーズ	解像 度	色深 度	MS104-SH4 用 バイナリ RPM	クロス開発環境用 ライブラリ RPM	PC 用 バイナリ・ ライブラリ RPM
USB キーボード	MS104-SH 4	640 × 480	8	alp-microwindows- 0.90-ms104-bpp16- usb-1.1-1.sh4.rpm	alp-microwindows-0.90- ms104-bpp16-usb-devel -sh4-1.1-1.sh4.rpm	alp-microwindows-0.90- ms104-bpp8-640x480-1. 1-1.i686.rpm
			16	alp-microwindows- 0.90-ms104-bpp16- usb-1.1-1.sh4.rpm	alp-microwindows-0.90- ms104-bpp16-usb-devel -sh4-1.1-1.sh4.rpm	alp-microwindows-0.90- ms104-bpp16-640x480- 1.1-1.i686.rpm
USB マウス	MS104-US B H/S	800 × 600	8	alp-microwindows- 0.90-ms104-bpp16- usb-1.1-1.sh4.rpm	alp-microwindows-0.90- ms104-bpp16-usb-devel -sh4-1.1-1.sh4.rpm	alp-microwindows-0.90- ms104-bpp8-800x600-1. 1-1.i686.rpm
			16	alp-microwindows- 0.90-ms104-bpp16- usb-1.1-1.sh4.rpm	alp-microwindows-0.90- ms104-bpp16-usb-devel -sh4-1.1-1.sh4.rpm	alp-microwindows-0.90- ms104-bpp16-800x600- 1.1-1.i686.rpm

※ 各種RPMは『MS104-SH4 Linux開発環境 Linux-KIT-A02』添付CD-ROMもしくは弊社MS104-SH4 ファイルセンター『<http://www.apnet.co.jp/e-linux/dl/filecenter/index.html>』から入手することができます。

※ MS104-SH4 ファイルセンター『<http://www.apnet.co.jp/e-linux/dl/filecenter/index.html>』にアクセスするには、ユーザ登録『https://www.apnet.co.jp/cgi-bin/ms104_reg/index.html』が必要です。

USB マウス・キーボード対応 Microwindows のインストール

MS104-SH4 クロス開発環境構築をサポートするユーティリティプログラム『**chdevelop_ms104sh4.sh**』を使用して、Microwindows RPM ファイルのインストール、および、RAM ディスク（ルートファイルシステム）の作成を行うことができます。

以下では、USB マウス・キーボードサポート、解像度 800×600、色深度 8bpp 対応の Microwindows 開発環境を構築します。

※ 『**chdevelop_ms104sh4.sh**』は『**MS104-SH4 Linux開発環境 Linux-KIT-A02**』添付CD-ROMもしくは弊社MS104-SH4 ファイルセンター『<http://www.apnet.co.jp/e-linux/dl/filecenter/index.html>』から入手することができます。

※ MS104-SH4 ファイルセンター『<http://www.apnet.co.jp/e-linux/dl/filecenter/index.html>』にアクセスするには、ユーザ登録『https://www.apnet.co.jp/cgi-bin/ms104_reg/index.html』が必要です。

① 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

② LinuxPC 上で Microwindows の開発環境構築をサポートするユーティリティプログラム『**chdevelop_ms104sh4.sh**』を実行します。

```
[root@ap_linux root]# chdevelop_ms104sh4.sh
```

③ ルートファイルシステムの作成先ディレクトリを問われます。

ルートファイルシステムの作成先ディレクトリに『**/opt/alpha_project/linux-kit-a02/linux/filesystems/microwindows**』ディレクトリを指定します。

```
[root@ap-linux root]# chdevelop_ms104sh4.sh
-----
Please select a directory for installation
[Default /opt/alpha_project/linux-kit-a02/linux/filesystems/work]:
/opt/alpha_project/linux-kit-a02/linux/filesystems/microwindows
```

④ ディレクトリを削除するか問われます。

ディレクトリを削除するため、『**yes**』を入力します。

```
[Delete root filesystems? Yes/No]
yes
DELETED root filesystems!
[Deleted /opt/alpha_project/linux-kit-a02/linux/filesystems/microwindows ]
```


- ⑤ MS104-VGA/LCD ボードを使用するか選択します。
MS104-VGA/LCD ボードを使用するため、**[1]** を選択します。

```
-----  
  
Please choose whether MS104-VGA/LCD board is used  
[1] used  
[2] unused  
  
[Default " unused"]:  
1 ←  
  
MS104-VGA/LCD board used
```

- ⑥ 色深度を選択します。
色深度 8bpp 対応の開発環境を構築するため、**[2]** を選択します。

```
-----  
  
Please select MS104-VGA/LCD board color depth  
[1] 16 bit per pixel  
[2] 8 bit per pixel  
  
[Default "8 bit per pixel"]:  
2 ←  
  
MS104-VGA/LCD board color depth is 8 bit per pixel
```

- ⑦ 解像度を選択します。
色深度 800×600 対応の開発環境を構築するため、**[2]** を選択します。

```
-----  
  
Please select MS104-VGA/LCD resolution  
[1] 640x480  
[2] 800x600  
  
[Default "800x600"]:  
2 ←  
  
MS104-VGA/LCD board resolution is 800x600
```

- ⑧ ヒューマンインターフェースデバイスを選択します。
USB キーボード・マウスをサポートするため、**[1]** を選択します。

```
-----  
  
Please choose which shall be used Touch Panel or USB Keyboard & Mouse.  
[1] USB keyboard & mouse  
[2] Touch panel  
  
[Default " Touch panel"]:  
1 ←  
  
USB mouse & keyboard usbhid
```

- ⑨ 選択した項目の一覧が表示され、インストールの可否に『yes』を選択します。

```
-----
Your select or automatically installation setting
Installation directory is          "/opt/alpha_project/linux-kit-a02/linux/filesystems/microwindows"
MS104-VGA/LCD board              "used"
MS104-VGA/LCD board color depth  "8"
MS104-VGA/LCD board resolution   "800x600"
USB mouse & keyboard              "used"
MS104-USB H/S board              "used"
MS104-USB H/S board is used as    "host"

[start install? Yes/No]:
yes ←入力
```

- ⑩ 新たな MS104-SH4 クロス開発環境用 RPM がインストールされ、『/opt/alpha_project/linux-kit-a02/linux/filesystems/microwindows』ディレクトリに RAM ディスク（ルートファイルシステム）が作成されます。

```
Uninstall old RPM package
Please wait : done.

Install new RPM package
Please wait : done.

Create ramdisk root filesystem for MS104-SH4
Please wait : done.

Created new MS104-SH4 Development Environment
```

- ⑪ LinuxPC 上でインストールされた RPM パッケージを確認します。

```
[root@ap_linux root]# rpm -qa | grep alp-microwindows ←入力
alp-microwindows-0.90-ms104-bpp8-800x600-1.1-1
alp-microwindows-0.90-ms104-bpp8-usb-devel-sh4-1.1-1
```

- ⑫ LinuxPC 上で作成されたルートファイルシステムを確認します。

```
[root@ap-linux bin]# cd /opt/alpha_project/linux-kit-a02/linux/filesystems/microwindows ←入力
[root@ap-linux microwindows]# ls ←入力
ramdisk
```

4.7 ルートファイルシステムの概要

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。

- apLinux MS104-SH4 専用のルートファイルシステム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。なお CF からの起動も可能になっています。

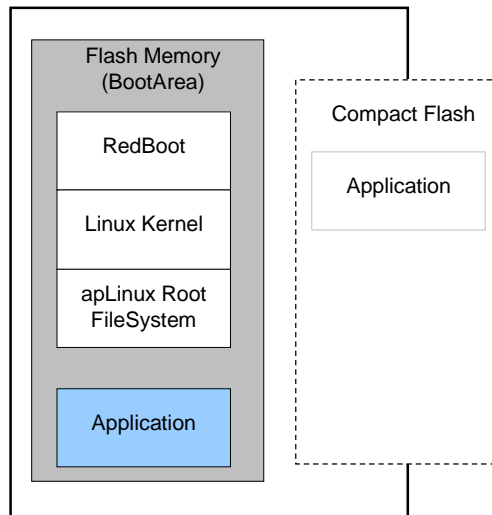


Fig 4.7-1 apLinux のシステム構成例

4.8 ルートファイルシステムの作成

USB ホストとして MS104-USB H/S を動作させるには USB ホスト用 Linux モジュールを Linux カーネルに組み込む必要があります。Linux カーネルにモジュールを組み込むために、ルートファイルシステムにモジュールをインストールし、設定ファイルを編集しなければなりません。

以下では、USB ホスト用ルートファイルシステムを作成する方法について説明します。

USB 設定ファイル

apLinux では USB ホスト用 Linux モジュールを組み込む際、『`/etc/sysconfig/usb`』ファイルを編集する必要があります。USB ホスト用 Linux モジュールを Linux カーネル起動時に組み込むには『`/etc/sysconfig/usb`』ファイルの『`USB`』を『`on`』にします。モジュールを組み込まない場合は、『`USB`』を『`off`』にします。各、USB デバイスについては、HID ストレージ、USB・シリアルコンバータを設定することができます。各 USB デバイスのモジュールを組み込む場合は、『`yes`』、組み込まない場合は『`no`』を設定します。

※ 上記の設定は USB ホスト用 Linux モジュールがルートファイルシステムにインストールされていることが前提です。



● `/etc/sysconfig/usb` ファイル

<code>USB=on</code>	: モジュールの組み込み
<code>HID=yes</code>	: HID デバイスのサポート
<code>MOUSE=yes</code>	: USB マウスのサポート
<code>KEYBOARD=yes</code>	: USB キーボードのサポート
<code>STORAGE=yes</code>	: ストレージデバイスのサポート
<code>DISK=yes</code>	: USB ハードディスクのサポート
<code>CDROM=yes</code>	: USB CD-ROM のサポート
<code>GENERIC=yes</code>	: USB スキャナーのサポート
<code>SERIAL=yes</code>	: FTDI 社製 USB・シリアルコンバータのサポート

USB マウス対応 Linux カーネル・RAM ディスクの作成

『4.4 Linux カーネルのコンパイル』を参考に MS104-USB H/S USB ホスト対応 Linux カーネル・モジュールを作成し、『4.6 Microwindows インストール』で作成した RAM ディスクルートファイルシステムに Linux モジュールをインストールします。

① 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
```

② 『4.4 カーネルのコンパイル』を実行し、Linux カーネルイメージ『`/opt/alpha_project/linux-kit-a02/linux/kernel/celinux-040503-alp/arch/sh/boot/zimage`』と Linux モジュールディレクトリ『`/opt/alpha_project/linux-kit-a02/linux/filesystems/module/lib/modules`』を作成します。

- ③ LinuxPC 上で『4.6 Microwindows インストール』で作成したルートファイルシステムの作成先ディレクトリに『/opt/alpha_project/linux-kit-a02/linux/filesystems/microwindows』に移動し、RAM ディスクのマウントポイントを作成します。

```
[root@ap_linux root]# cd /opt/alpha_project/linux-kit-a02/linux/filesystems/microwindows/ramdisk
[root@ap_linux ramdisk]# mkdir -p mnt
```

- ④ LinuxPC 上で RAM ディスクイメージを解凍します。

```
[root@ap_linux ramdisk]# gzip -d ramdisk-ms104sh4-x.x.gz
```

※ ファイル名の x.x はバージョン番号を示します。

- ⑤ LinuxPC 上でルートファイルシステム RAM ディスクイメージをマウントします。

```
[root@ap_linux ramdisk]# mount -t ext2 -o loop ramdisk-ms104sh4-x.x mnt
```

- ⑥ USB デバイスドライバモジュールを RAM ディスクイメージにコピーします。

LinuxPC 上で『cp -r /opt/alpha_project/linux-kit-a02/linux/filesystems/module/lib/modules/mnt/lib/』を実行してください。

```
[root@ap_linux ramdisk]# cp -r /opt/alpha_project/linux-kit-a02/linux/filesystems/module/lib/modules/mnt/lib/
```

- ⑦ USB デバイスシステムファイル『/etc/sysconfig/usb』の確認をします。

LinuxPC 上で『vi mnt/etc/sysconfig/usb』を実行します。

```
[root@ap_linux ramdisk]# vi mnt/etc/sysconfig/usb
```

● /etc/sysconfig/usb ファイル

```
USB=on
HID=yes
MOUSE=yes
KEYBOARD=yes

STORAGE=yes
DISK=yes
CDROM=yes
GENERIC=yes

SERIAL=yes
```

“on” になっていることを確認する。

“yes” になっていることを確認する。

- ⑧ LinuxPC 上で RAM ディスクイメージをアンマウントします。

```
[root@ap_linux ramdisk]# umount ramdisk-ms104sh4-x.x
```

- ⑨ LinuxPC 上で RAM ディスクイメージを圧縮します。

```
[root@ap_linux ramdisk]# gzip ramdisk-ms104sh4-x.x
```

4.9 サンプルプログラム作成

この章では、Microwindows のサンプルプログラムを作成し、MS104-SH4、MS104-VGA/LCD、MS104-USB H/S、USB マウスを組み合わせたボード上で動作させます。

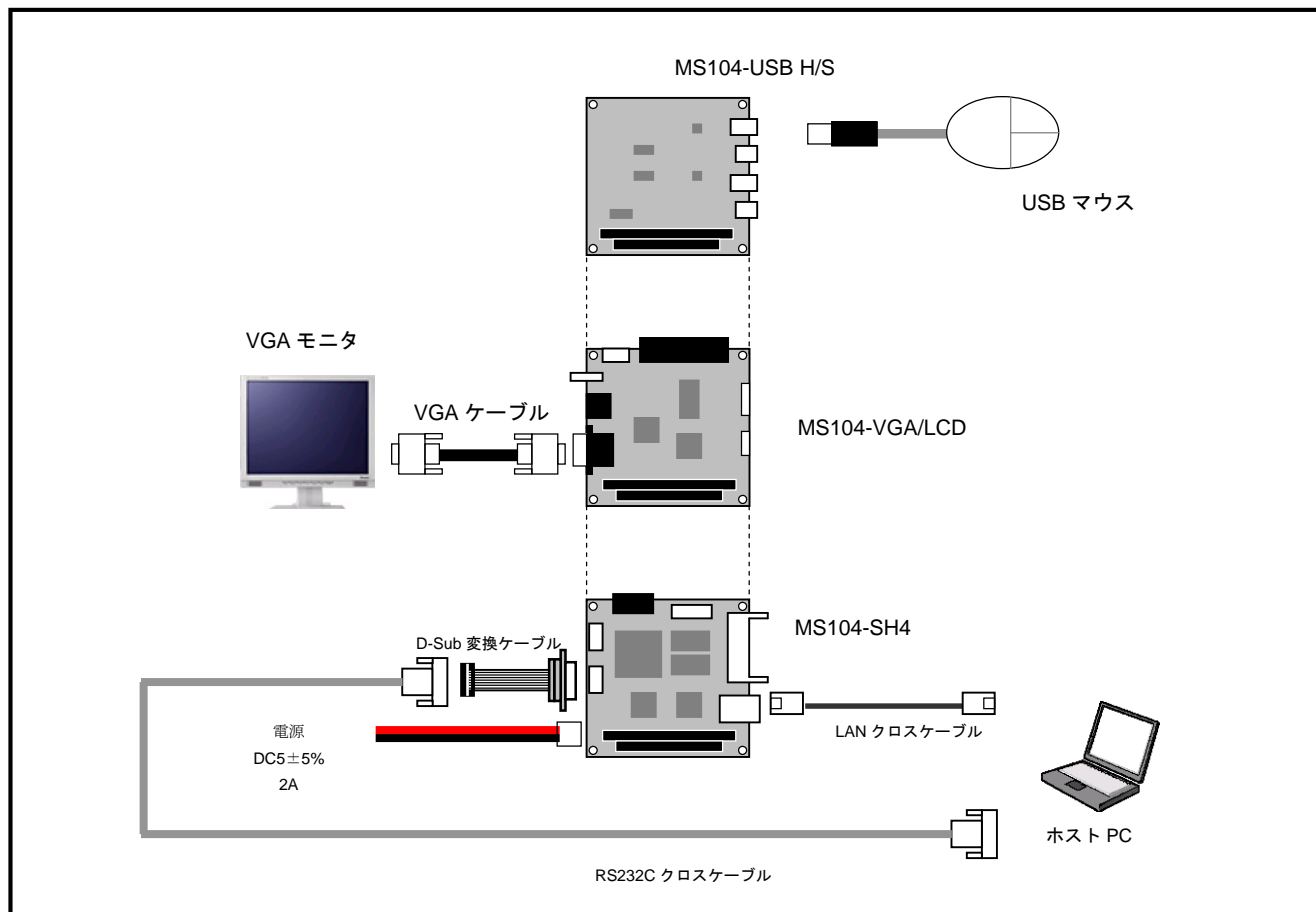


Fig 4.9-1 USB マウスとの接続

MS104-SH4 上での動作

sh4-linux-gcc で Microwindows サンプルプログラムをクロスコンパイルします。
出力先として VGA モニタを使用します。

● microwindows_sample.c

```
#include <stdio.h>
#include "nano-X.h"

// 色定義
#define WHITE MWRGB(255, 255, 255)
#define BLACK MWRGB(0, 0, 0)
#define RED MWRGB(255, 0, 0)
#define GREEN MWRGB(0, 255, 0)
#define BLUE MWRGB(0, 0, 255)

void errorhandler(GR_EVENT *ep); // イベントハンドラ

int main(void)
{
    GR_WINDOW_ID wid1;
    GR_GC_ID gc;
    GR_EVENT event;
    GR_WM_PROPERTIES props;
    GR_POINT point[3];

    if(GrOpen() < 0) // Nano-X サーバへの接続を開始する
    {
        printf("Can't open graphics\n");
        exit(1);
    }

    GrSetErrorHandler(errorhandler); // エラーハンドラの登録

    // 新規ウィンドウの作成
    wid1 = GrNewWindow(GR_ROOT_WINDOW_ID, 0, 0, 260, 230, 0, WHITE, BLACK);

    // ウィンドウタイトル設定
    props.flags = GR_WM_FLAGS_TITLE;
    props.title = "Sample";
    GrSetWMPProperties(wid1, &props); // ウィンドウ情報のコピー

    gc = GrNewGC(); // 新規グラフィックコンテキストの作成

    // イベントマスク設定
    GrSelectEvents(wid1, GR_EVENT_MASK_EXPOSURE | GR_EVENT_MASK_CLOSE_REQ);
    GrMapWindow(wid1); // ウィンドウの再帰表示
    GrSetFocus(wid1); // ウィンドウフォーカスの設定

    // 三角形のための座標
    point[0].x = 10;
    point[0].y = 140;
    point[1].x = 10;
    point[1].y = 220;
    point[2].x = 250;
    point[2].y = 180;
```

```

// メッセージループ
for(;;)
{
    GrGetNextEvent(&event); // イベントキューからイベントを取り出す
    switch(event.type)
    {
        // 描画
        case GR_EVENT_TYPE_EXPOSURE:
            GrSetGCForeground(gc, BLUE); // 前景色の指定
            GrFillRect(wid1, gc, 10, 10, 100, 100); // 塗りつぶし四角形を描画
            GrSetGCForeground(gc, RED); // 前景色の指定
            GrFillEllipse(wid1, gc, 200, 60, 50, 50); // 塗りつぶし楕円を描画
            GrSetGCForeground(gc, GREEN); // 前景色の指定
            GrFillPoly(wid1, gc, 3, point); // 塗りつぶしポリゴンを描画
            break;
        // クローズ
        case GR_EVENT_TYPE_CLOSE_REQ:
            GrClose(); // Nano-X との接続を終了
            exit(0);
            break;
    }
}

GrClose();

return 0;
}

// エラーハンドラー
void errorhandler(GR_EVENT *ep)
{
    printf("Error (%s) code %d id %d", ep->error.name, ep->error.code,
        ep->error.id);
    exit(1);
}

```

- ① LinuxPC に root 権限でログインします。

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

- ② LinuxPC 上で Microwindows サンプルプログラム格納ディレクトリに移動します。

```
[root@ap_linux root]# cd /opt/alpha_project/linux-kit-a02/linux/sample/microwindows ←入力
```

- ③ LinuxPC 上でソースファイルをクロスコンパイルし、MS104-SH4 用 Nano-x クライアントアプリケーションプログラム『microwindows_sample-usb.out』を作成します。

```
[root@ap_linux microwindows]# sh4-linux-gcc microwindows_sample.c -I/usr/local/sh4-linux/include/microwin -L/usr/local/sh4-linux/lib -lnano-X -o microwindows_sample-usb.out ←入力
```


- ④ LinuxPC 上で作成したサンプルアプリケーションプログラムを NFS 共有ディレクトリ 『/nfs』 にコピーし、NFS を起動します。

NFS サーバの設定・起動については『MS104-SH4 Linux 導入マニュアル』をご覧ください。

```
[root@ap_linux microwindows]# cp microwindows_sample-usb.out /nfs/
[root@ap_linux microwindows]# /etc/init.d/nfs start
```

- ⑤ LinuxPC 上で『4.8 ルートファイルシステムの作成』で作成した Linux カーネルと RAM ディスクルートファイルシステムを TFTP サーバディレクトリ 『/tftpboot』 にリネームして、コピーします。

※ TFTP サーバの起動については『MS104-SH4 Linux 導入マニュアル』をご覧ください。

```
[root@ap_linux microwindows]# cd /opt/alpha_project/linux-kit-a02/linux/filesystems/microwindows/ramdisk
[root@ap_linux ramdisk]# cp ramdisk-ms104sh4-x.x.gz /tftpboot/ramdisk.gz
[root@ap_linux ramdisk]# cp /opt/alpha_project/linux-kit-a02/linux/kernel/celinux-040503-alp/arch/sh/boot/zImage /tftpboot/vmlinuz
```

※ ファイル名の x.x はバージョン番号を示します。

- ⑥ MS104-USB H/S、MS104-VGA/LCD と MS104-SH4 を『Fig 4.9-1 USB マウスとの接続』を参考に接続します。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、自動起動を停止するために『Ctrl+C』とタイプします。

```
+Ethernet eth0: MAC address 00:0c:7b:xx:xx:xx
IP: 192.168.1.200/255.255.255.0, Gateway: 0.0.0.0
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 20:25:22, Oct 3 2003

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

RAM: 0x8c000000-0x8e000000, 0x8c00f0b8-0x8dfdd000 available
FLASH: 0x80000000 - 0x81000000, 128 blocks of 0x00020000 bytes each.
== Executing boot script in 5.000 seconds - enter ^C to abort
^C
RedBoot>
```

- ⑦ RedBoot に Linux カーネルイメージと RAM ディスクイメージをダウンロードします。

TFTP サーバの IP アドレスは「192.168.1.201」と仮定します。

```
RedBoot> load -r -b 0x8c210000 -m tftp -h 192.168.1.201 vmlinuz
Raw file loaded 0x8c210000-0x8c3072e1, assumed entry at 0x8c210000
RedBoot> load -r -b 0x8c360000 -m tftp -h 192.168.1.201 ramdisk.gz
Raw file loaded 0x8c360000-0x8c6b9da4, assumed entry at 0x8c360000
```

- ⑧ VGA モニタ、解像度 800×600、色深度 8bpp 対応 apLinux を起動します。

```
RedBoot> exec -m 1 -f 0 -r 0x0100 -l 1 -i 0x360000 -j 0xc00000 -c "console=ttySC1,3840
0,N,8 mem=32M ether=8,0x1300,0,0,eth0 video=e1356fb:system:ms104sh4,display:crt,bpp:8,800x600@6
0"
```

- ⑨ apLinux の起動を確認し、root 権限でログインします。

```
MS104SH4 login: root
```

- ⑩ apLinux から NFS マウントを実行します。

NFS サーバの IP アドレスは「192.168.1.201」と仮定します。

```
[MS104SH4@root]# mount -t nfs -o nolock -o rsize=2048 192.168.1.201:/nfs /mnt/nfs
```

- ⑪ Nano-X サーバを起動します。

```
[root@ap_linux microwindows]# nano-X &
```

- ⑫ nanowm ウィンドウマネージャを起動します。

```
[root@ap_linux microwindows]# nanowm &
```

- ⑬ 作成したサンプルアプリケーションプログラムを起動します。

```
[root@ap_linux microwindows]# /mnt/nfs/ microwindows_sample-usb.out
```

- ⑭ VGA モニタに下記の出力がされていることを確認し、USB マウスでウィンドウが移動することを確認します。

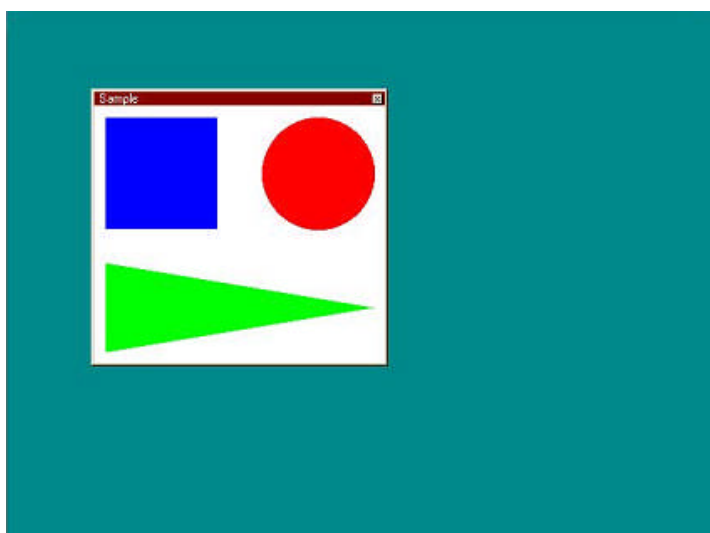


Fig 4.9-2 サンプルアプリケーションプログラム

5. 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 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。

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 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。

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
「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」	Daniel P. Bovet,Marco Cesati 著 高橋弘和、早川仁 監訳 岡島順治朗、田宮まや、三浦広志 訳／オライリージャパン
「USB ハード&ソフト開発のすべて」	CQ 出版社

その他 各社データシート

<http://www.usb.org/>

USB Implementers Forum ホームページ。各種 USB 規格書をダウンロードできます。

<http://www.linux-usb.org>

Linux USB プロジェクト ホームページ。USB の Linux への移植を行っています。

<http://embedded.centurysoftware.com/>

Century Software 株式会社ホームページ。本製品に収録されている Microwindows の開発元。

<http://www.embedded.jp/>

アップウィンドテクノロジー・インコーポレイテッドが主催している組込み向け情報サイト。

謝辞

Linux、SH-Linux、eCos/Redboot、Microwindows の開発に関わった多くの貢献者に深い敬意と感謝の意を示します。

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。
- ・ Microwindows は Century Software の登録商標、商標または商品名称です。
- ・ eCos™ および RedBoot™ は RedHat™ 社の商標です。
- ・ Windows® の正式名称は Microsoft® Windows® Operating System です。
Microsoft、Windows、Windows NT は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。
Windows® XP、Windows® 2000 Professional、Windows® Millennium Edition、Windows® 98 は、米国 Microsoft Corporation の商品名称です。
本文書では下記のように省略して記載している場合がございます。ご了承ください。
Windows® XP は Windows XP もしくは WinXP
Windows® 2000 Professional は Windows 2000 もしくは Win2000
Windows® Millennium Edition は Windows Me もしくは WinMe
Windows® 98 は Windows 98 もしくは Win98
- ・ 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。



ALPHA PROJECT Co.,LTD.

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