

MS104-SH4

MS104-USB H/S の使用方法

1版 2005年02月18日

目次

ダイジェスト版

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。

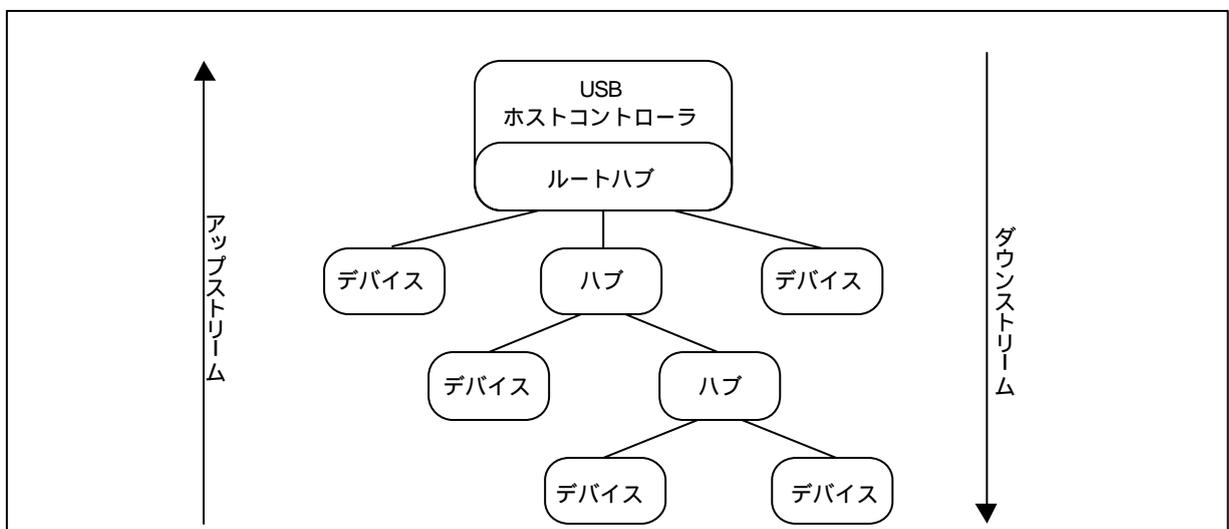


Fig1.2-1 USB のバスターミナル

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。

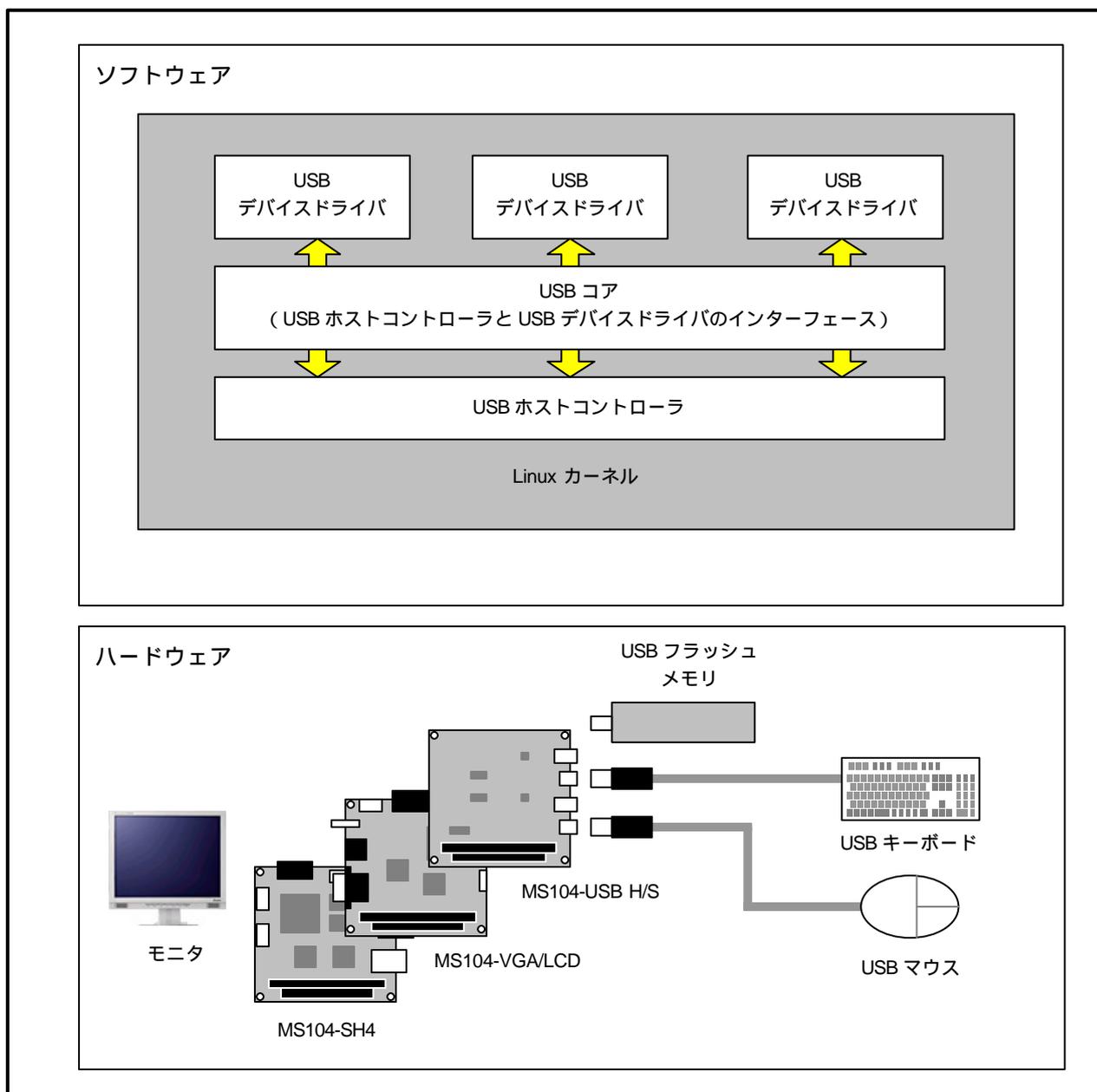


Fig 2.1-1 MS104-USB H/S の概要

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。

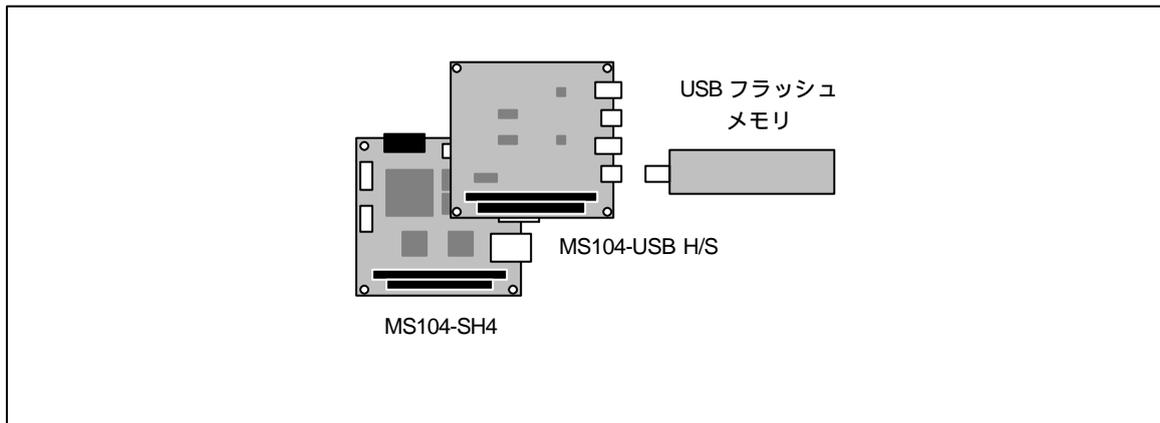


Fig 2.2-1 USB フラッシュメモリとの接続

下図に USB キーボード・マウスの接続例を示します。

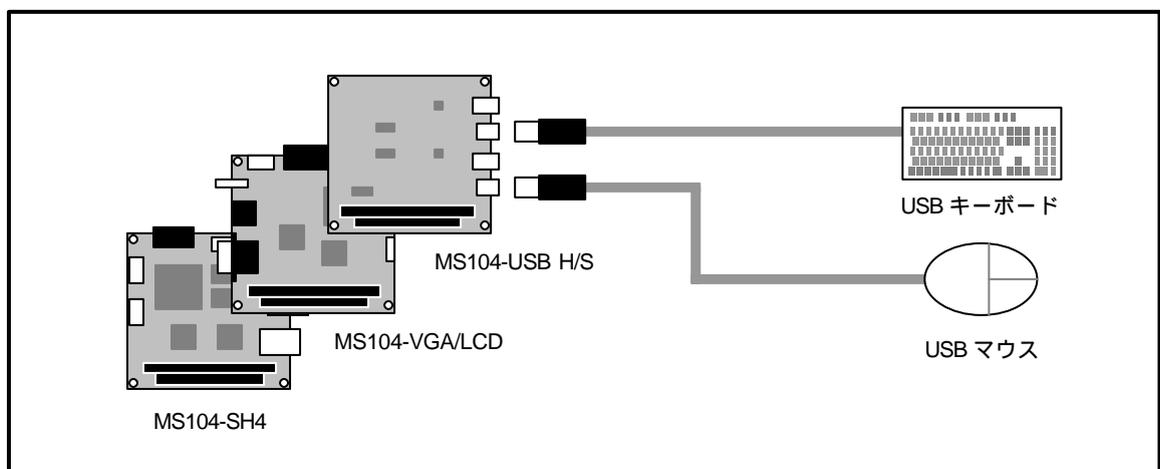


Fig 2.2-2 USB キーボード・マウスとの接続

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。

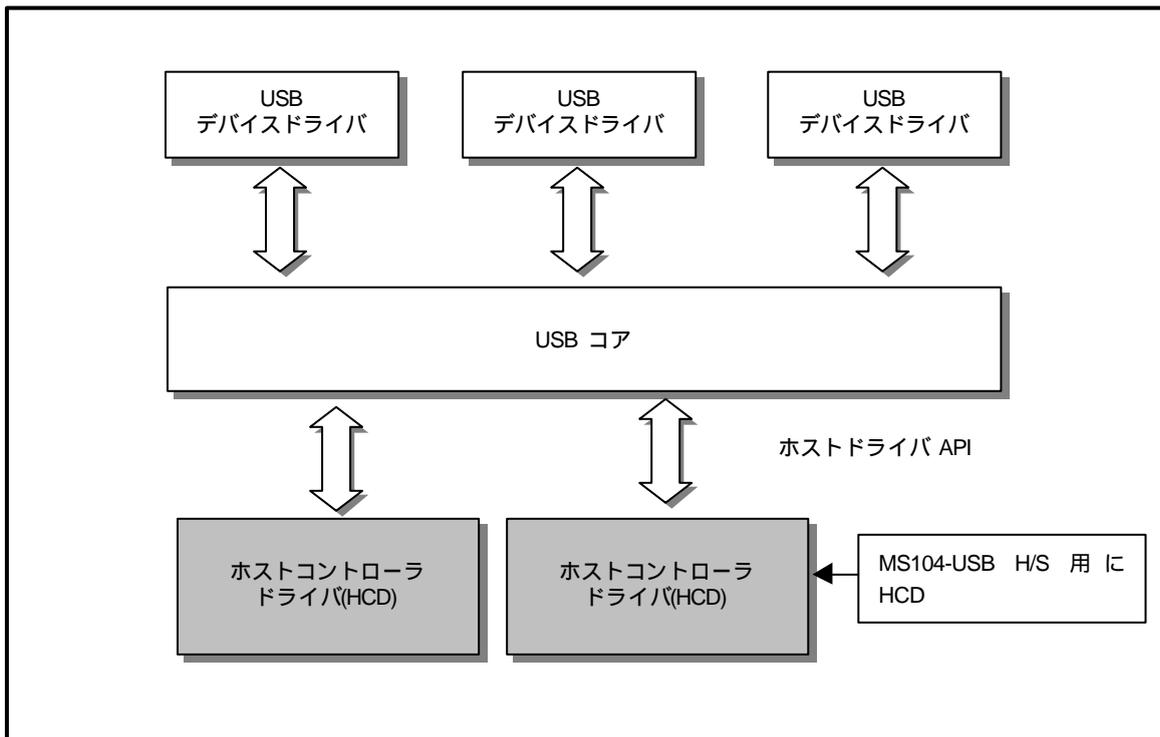


Fig 2.3-1 Linux USB ドライバ

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	割り込み番号
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り込み制御
    |-- hc_sl811.h : USB パケットの送受信と割り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 バスから電源の供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

- 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。

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。

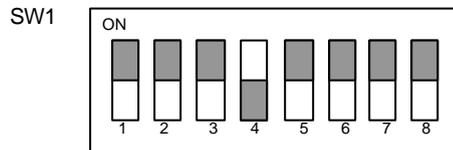


Fig 3.2-1 SW1 の設定

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。

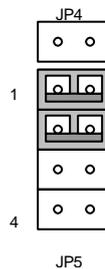


Fig 3.2-2 JP5 の設定

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。

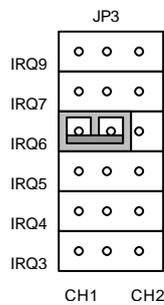


Fig 3.2-3 JP3 の設定 (その 1)

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。

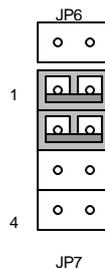


Fig 3.2-4 JP7 の設定

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。

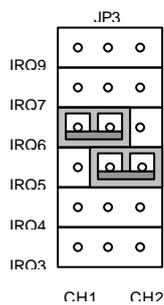


Fig 3.2-5 JP3 の設定 (その 2)

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) の使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。

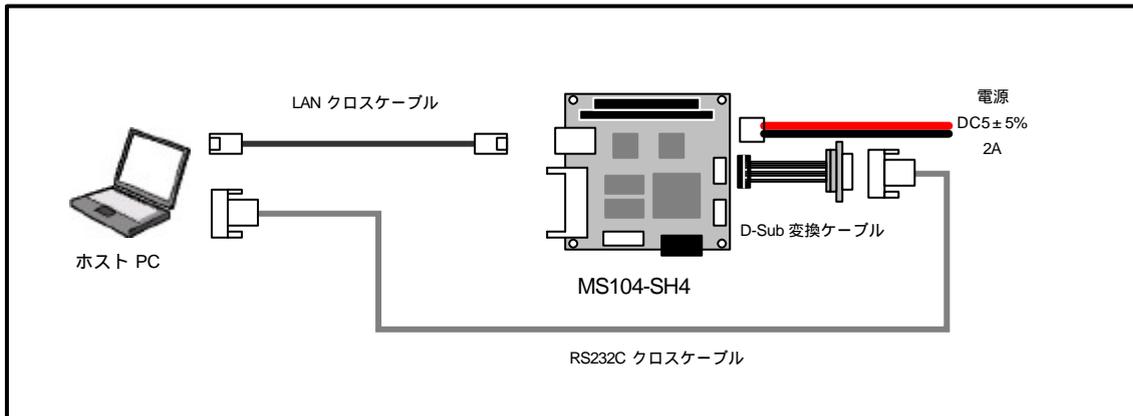


Fig 3.3-1 MS104-SH4 の接続

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
```

「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。

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>
```

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 0x8dfdf000-0x8dff0000 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。

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 0x8dfdf000-0x8dff0000 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。

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>
```

『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。

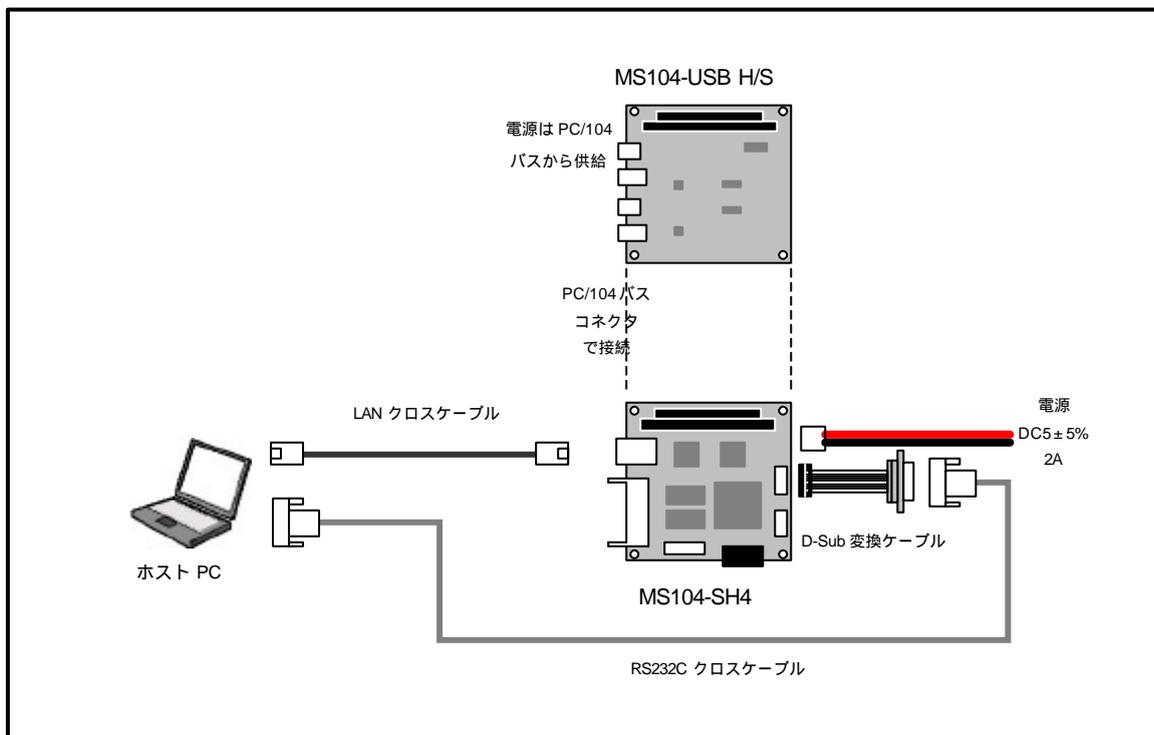


Fig 3.3-2 MS104-SH4 との接続

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>
```

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.
.
```

Linux の起動を確認し、root 権限でログインします。

```
MS104SH4 login: root
```

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
```

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
```

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> をご覧ください。

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。

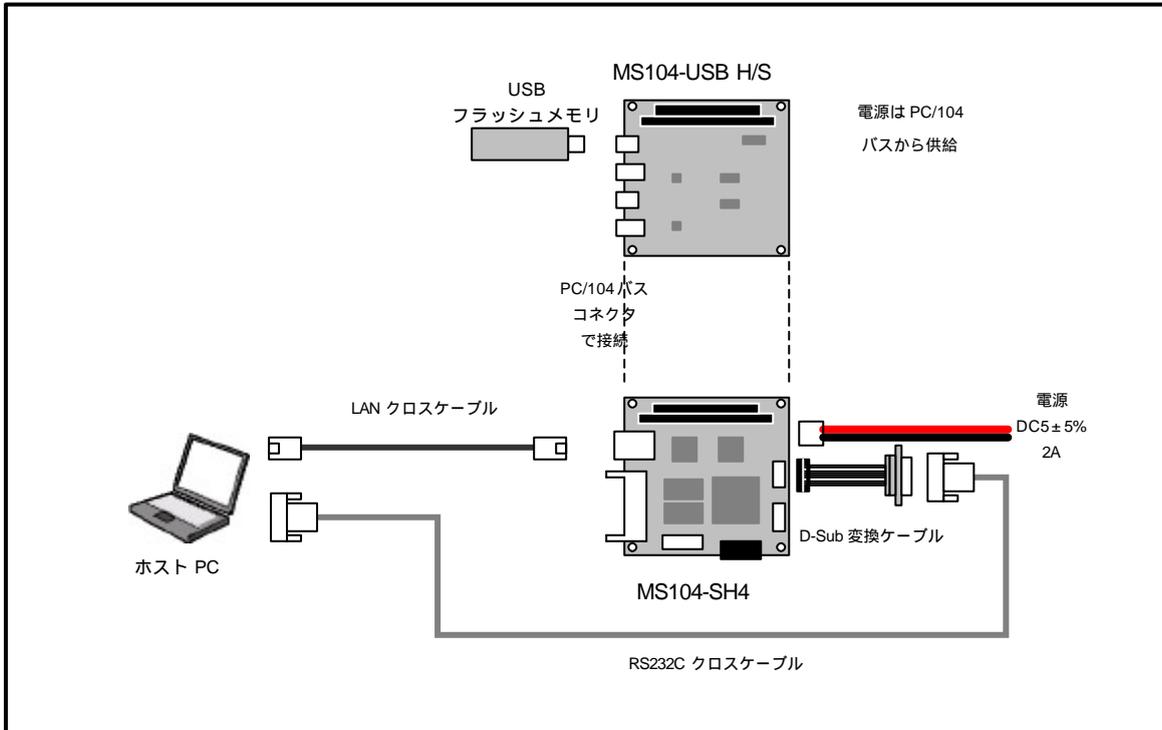


Fig 3.3-3 USB フラッシュメモリとの接続

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>
```

『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.
.
.
```

Linux の起動を確認し、root 権限でログインします。

```
MS104SH4 login: root
```

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

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
```

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。

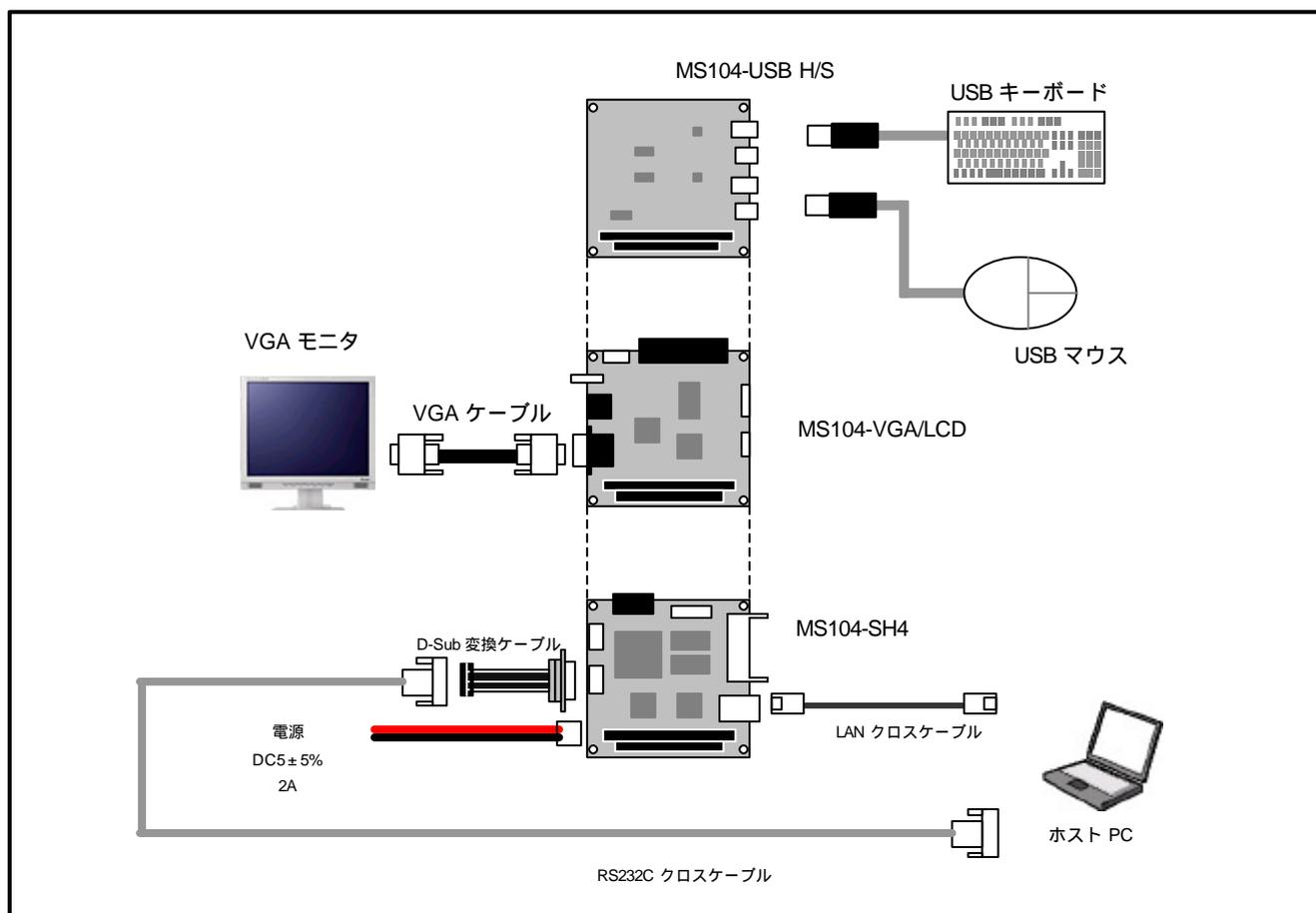


Fig 3.3-4 USB キーボード・USB マウスとの接続

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>
```

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

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 ←入力
.
RedBoot>
```

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 ←入力
.
RedBoot>
```

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 ←入力
.
RedBoot>
```

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 ←入力
.

RedBoot> fconfig ms104vga_output_lcd false ←入力
.

RedBoot> fconfig ms104vga_output_lcd_sim false ←入力
.

RedBoot> fconfig ms104vga_output_ntsc false ←入力
.

RedBoot> fconfig ms104vga_output_svideo false ←入力
.

RedBoot>
```

『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 ←入力
boot_script: false
.

kernel_para: false
ms104vga: true ← MS104-VGA/LCD サポート
ms104vga_bpp: 8 ← 色深度設定 (8 or 15 or 16)
ms104vga_custom_pc104: false ← カスタム PC/104 バスの使用
ms104vga_output_lcd: false ← LCD 出力サポート
ms104vga_output_lcd_sim: false ← LCD 同時出力サポート
ms104vga_output_ntsc: false ← NTSC コンポジット出力サポート
ms104vga_output_svideo: false ← S-Video 出力サポート
ms104vga_output_vga: true ← VGA 出力サポート
ms104vga_res: 800600 ← 解像度設定 (800600 or 640480)
net_debug: false
RedBoot>
```

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。

『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 ←入力
Now loading Linux kernel:
Linux kernel source address : 0x80040000
Linux kernel destination address : 0x8c210000
Linux kernel size : 0x00100000
.
.
Uncompressing Linux... Ok, booting the kernel.
.
.
```

Linux の起動を確認し、root 権限でログインします。

```
MS104SH4 login: root
```

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。

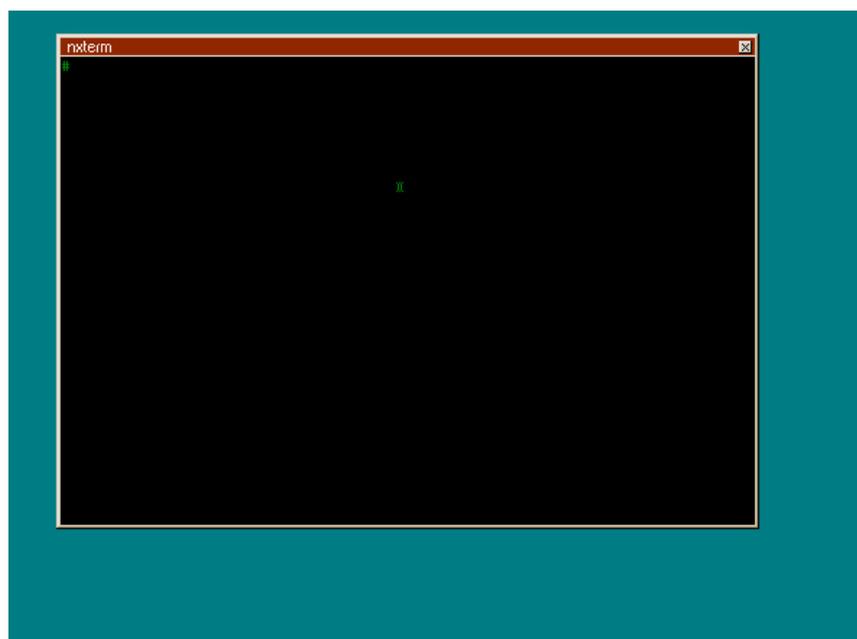


Fig 3.3-5 nxterm の起動

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。

Linux カーネルの作成

Microwindowsのインストール

ルートファイルシステムの作成

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。

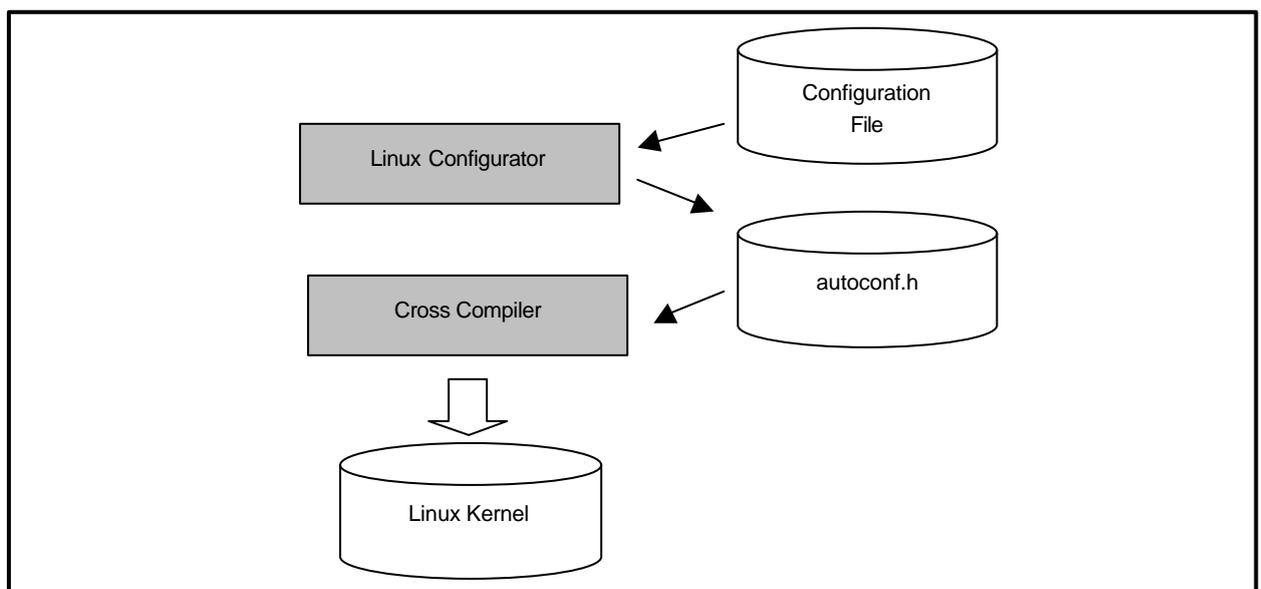


Fig 4.1-1 Linux の build イメージ

4.5 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。

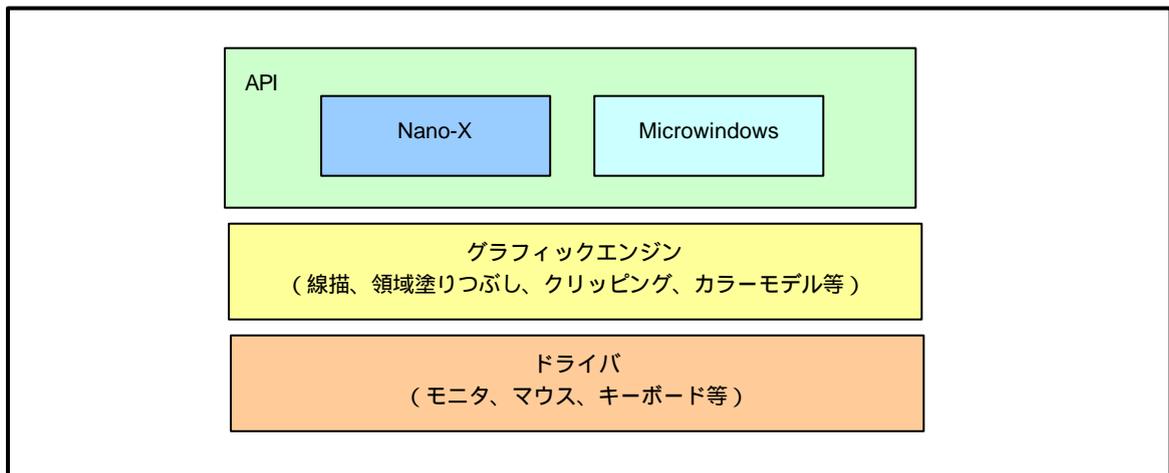


Fig 4.5-1 Microwindows アーキテクチャ

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。

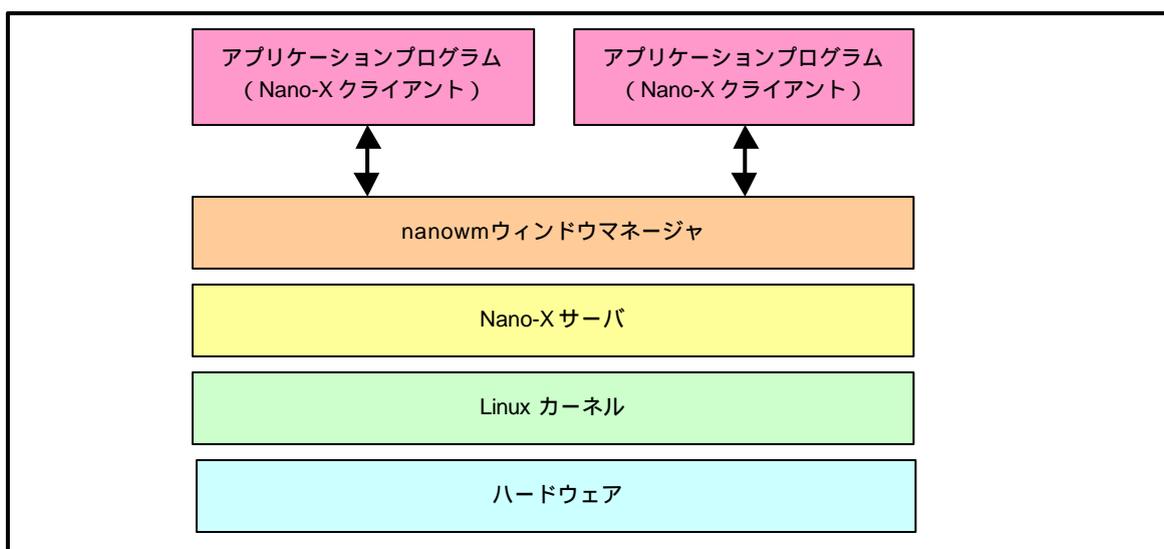


Fig 4.5-2 Microwindows システム構成

4.7 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。

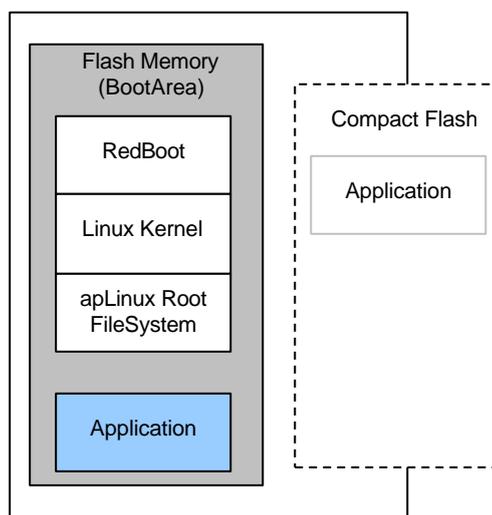


Fig 4.7-1 apLinux のシステム構成例

4.9 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。

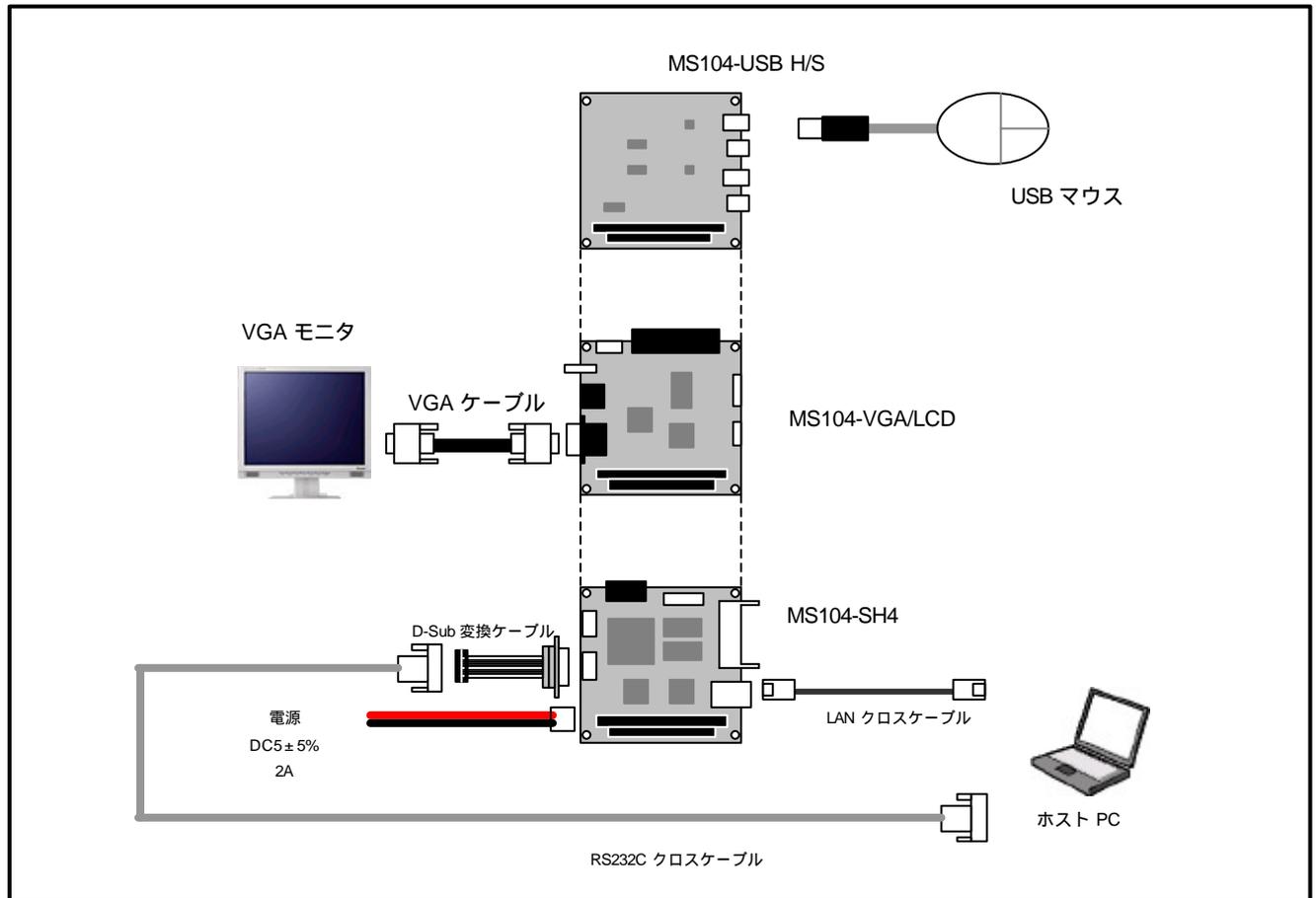


Fig 4.9-1 USB マウスとの接続

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