MS104-SH4
MS104-VGA/LCD の使用方法

第２版　２００９年３月２７日

目　次

１．概要

本アプリケーションノートはグラフィックコントローラボード「MS104-VGA/LCD」を MS104-SH4 用 Linux で使用する方法について述べます。
MS104-VGA/LCD は、S1D13506（セイコーエプソン社）グラフィックコントローラを搭載した PC/104 準拠周辺ボードです。
MS104-VGA/LCD は、VGA、LCD、NTSC、S-Video 出力をサポートし、タッチパネルコントローラ、ブザーデバイスをボード上に実装しています。
MS104-SH4 と MS104-VGA/LCD を組み合わせることにより高度な GUI を備えたシステムを容易に構築することができます。
本アプリケーションノートでは、MS104-SH4 を使用して U-Boot、Linux 及び DirectFB 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-A03」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下さい。

１．２　U-Boot について

U-Boot は DENX Software Engineering 社の Wolfgang Denk 氏が保守を行っているオープンソフトウェアの汎用ブートローダです。多くの開発者によって支援され、現在最も機能が豊富で柔軟性に富み、開発が活発に行われています。対応しているアーキテクチャは、SuperH の他に、PPC、ARM、AVR32、MIPS、x86、68k、Nios、MicroBlaze などです。またプログラムのダウンロードに関しても、ネットワークを介した TFTP の他に、CF カードなどのストレージデバイスからのダウンロードにも対応しています。

１．３　DirectFB について

DirectFB は Linux 上のフレームバッファデバイスだけではなく、ハードウェアグラフィックスアクセラレーション、入力デバイスの操作と抽象化、及び半透明のウィンドウと複数の表示レイヤをサポートした高度なウィンドウシステムを提供します。k 下層のハードウェアによってサポートされない描画処理を、ソフトウェアによって代替する完成されたハードウェアの抽象化レイヤです。

２．MS104-VGA/LCD について

２．１　MS104-VGA/LCD の概要

MS104-VGA/LCD は VGA、LCD、NTSC、S-Video 出力をサポートし、タッチパネルコントローラ、ブザーデバイスも実装した PC/104 周辺ボードです。
MS104-VGA/LCD は MS104-SH4 CPU ボードと組み合わせることにより、高度な GUI を構築することが可能です。
また、弊社 LVDS/DVI ボード『MS104-LVDS/DVI』と LCD キット『LCD-KIT-A03』を組み合わせることにより、LCD を使用した開発環境を構築することができます。

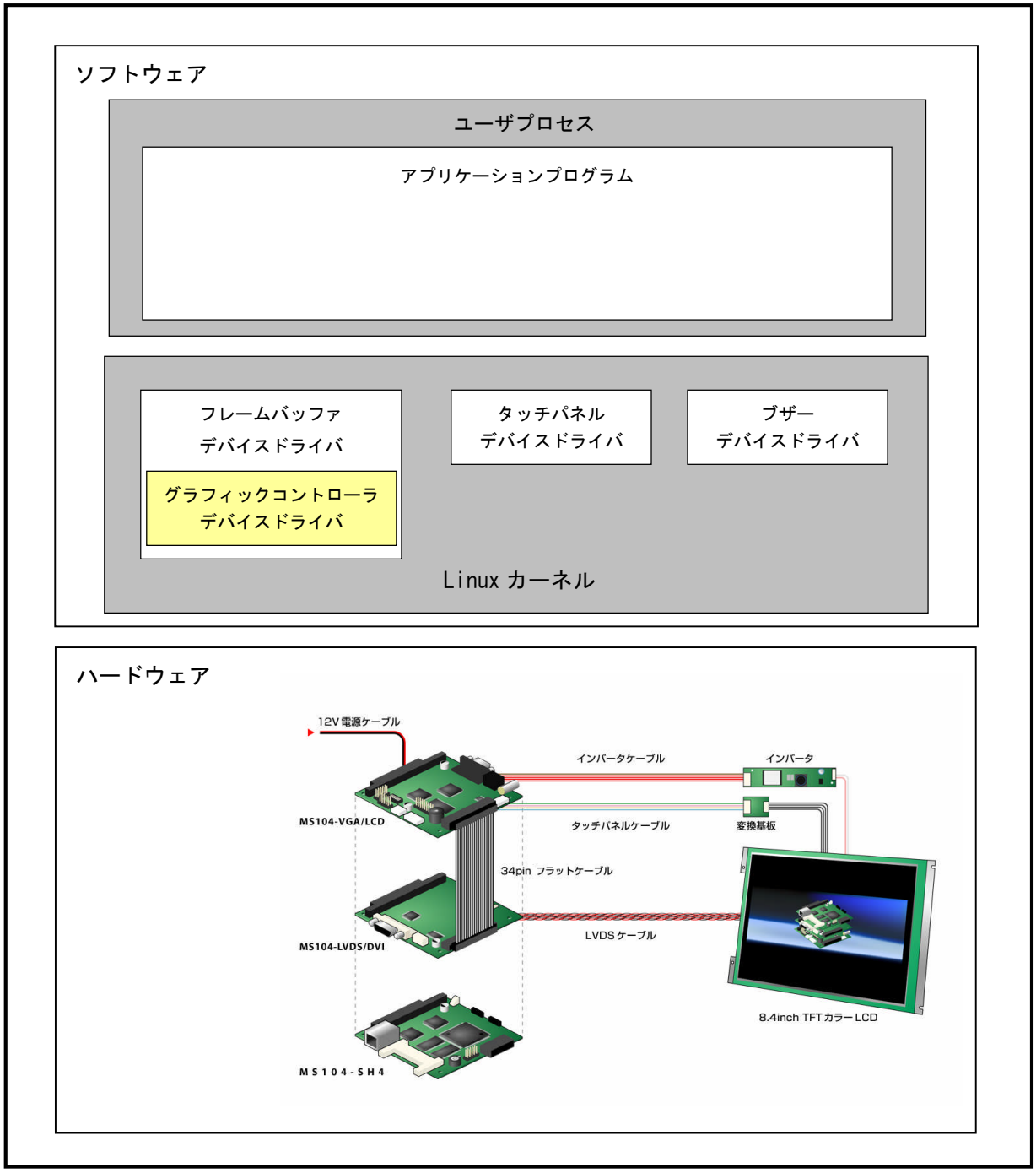


Fig 2.1-1　MS104-VGA/LCD の概要

２．２　MS104-VGA/LCD とモニタの接続

MS104-VGA/LCD を VGA、LCD、NTSC、S-Video 出力をサポートしています。各モニタと接続する際は下図を参考にして下さい。

下図に LCD 及び DVI モニタに接続例を示します。LCD 及び DVI モニタの接続にはディスプレイインターフェースボード『MS104-LVDS/DVI』とタッチパネル付 TFT-LCD キット『LCD-KIT-A03』が必要となります。
LCD、DVI は同時に出力することができます。

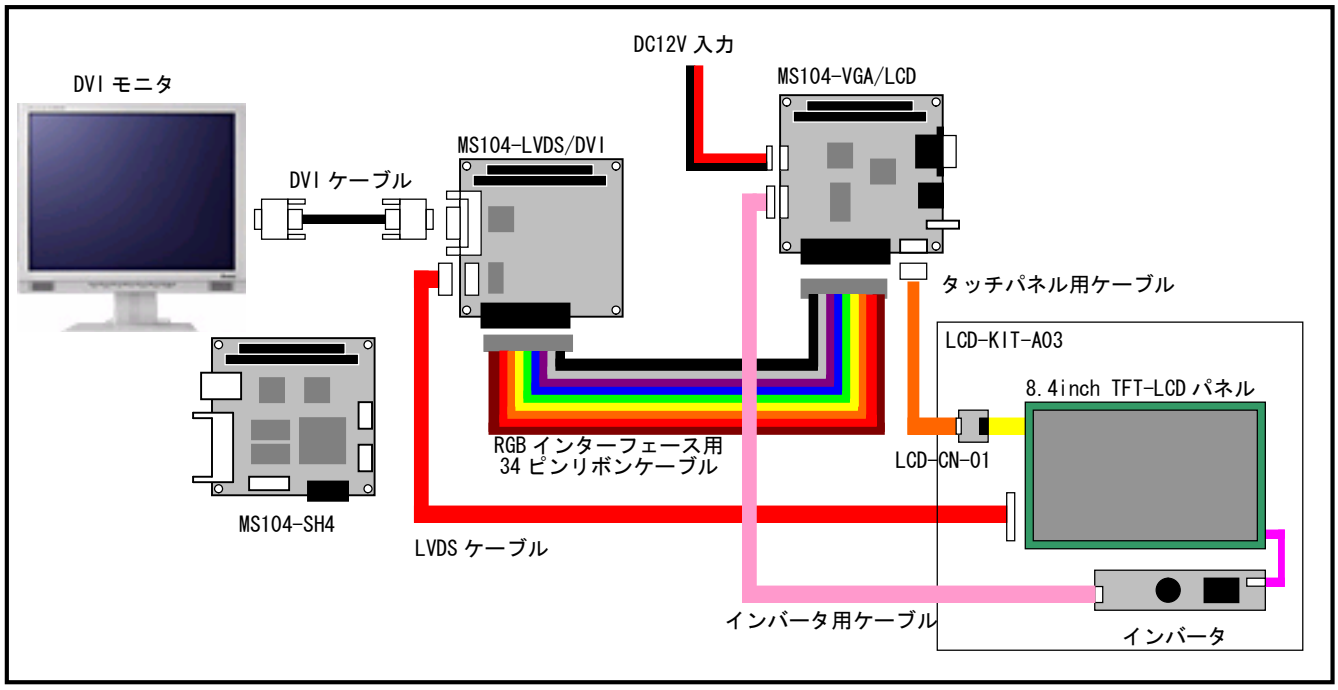


Fig 2.2-1　LCD、DVI との接続

下図に VGA、NTSC、S-Video モニタの接続例を示します。
VGA、NTSC、S-Video は同時に出力することができません。出力先はソフトウェア（U-Boot、Linux）により選択されます。

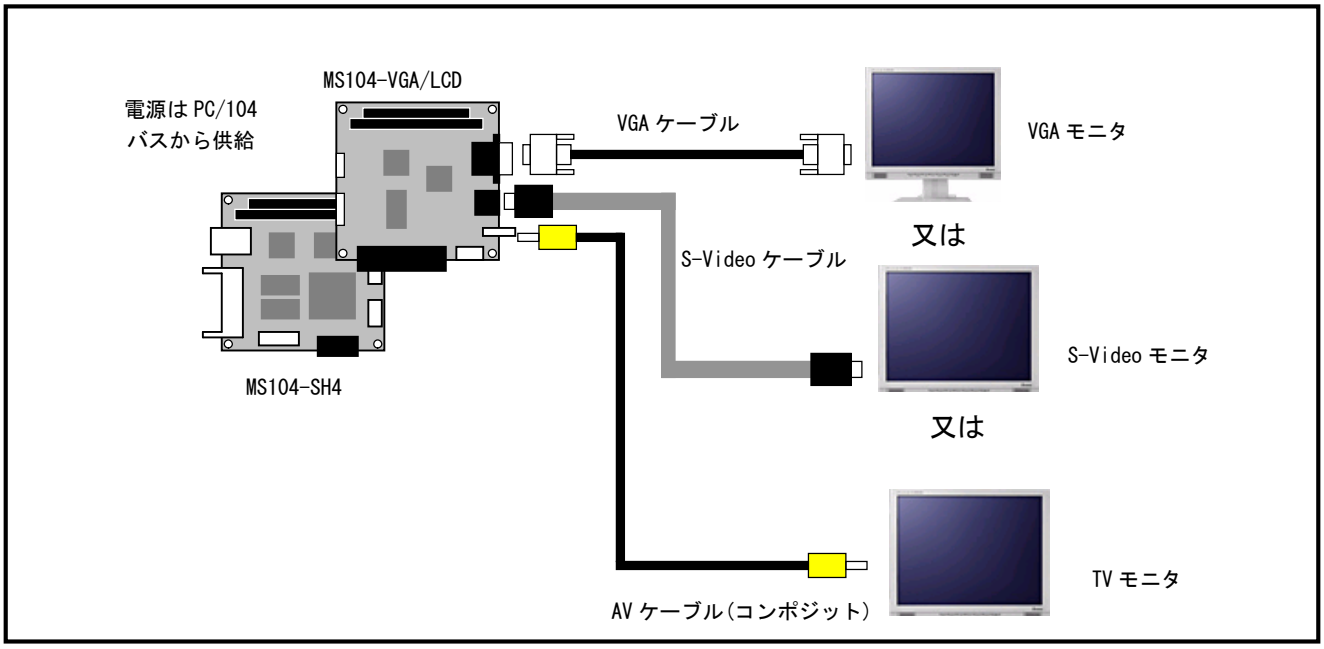


Fig 2.2-2　VGA、NTSC、S-Video との接続

２．３　MS104-VGA/LCD の出力対応

MS104-SH4 用 U-Boot・Linux で対応している出力の一覧を示します。

出力先	解像度	色深度	U-Boot		Linux	
			垂直 周波数 (Hz)	水平 周波数 (kHz)	垂直 周波数 (Hz)	水平 周波数 (kHz)
LCD	640×480 ※1	8	47	24.7	47	24.7
		15	47	24.7	-	-
		16	47	24.7	47	24.7
	800×600	8	60.3	37.9	60.3	37.9
		15	60.3	37.9	-	-
		16	60.3	37.9	60.3	37.9
VGA	640×480	8	60.1	29.5	60.1	29.5
		15	60.1	29.5	-	-
		16	60.1	29.5	60.1	29.5
	800×600	8	60.3	37.9	60.3	37.9
		15	60.3	37.9	-	-
		16	60.3	37.9	60.3	37.9
NTSC	640×480	8	62	15.7	62	15.7
		15	62	15.7		-
		16	62	15.7	62	15.7
S-Video	640×480	8	62	15.7	62	15.7
		15	62	15.7	-	-
		16	62	15.7	62	15.7
LCD & VGA	640×480 ※1	8	31	16.5	31	16.5
			60.1	29.5	60.1	29.5
	800×600	8	20.1	12.6	20.1	12.6
			60.3	37.9	60.3	37.9
LCD & NTSC	640×480 ※1	8	31	16.5	31	16.5
			62	15.7	62	15.7
LCD & S-Video	640×480 ※1	8	31	16.5	31	16.5
			62	15.7	62	15.7

Table 2.3-1　MS104-VGA/LCD 出力対応

※1　弊社「LCD-KIT-A03」では解像度が異なるため、ご使用いただけません。

２．４　デバイスドライバについて

MS104-VGA/LCD ボードに対して、U-Boot ではビデオドライバを追加し、Linux ではフレームバッファ、タッチパネル、ブザーデバイスドライバを追加しています。この章では、デバイスドライバの概要について説明します。

MS104-VGA/LCD の色深度、解像度はソフトウェア(U-Boot、Linux、DirectFB 等)に依存します。そのため、各設定に合わせてソフトウェアを変更する必要があります。

２．４．１　U-Boot

U-Boot は MS104-VGA/LCD ボードに対し、ロゴの出力を行います。解像度、色深度、出力先といった設定はコンパイル時に設定されたコンフィグレーションデータによって決まります。また、ロゴは BMP 形式の画像ファイルのみ使用することができます。

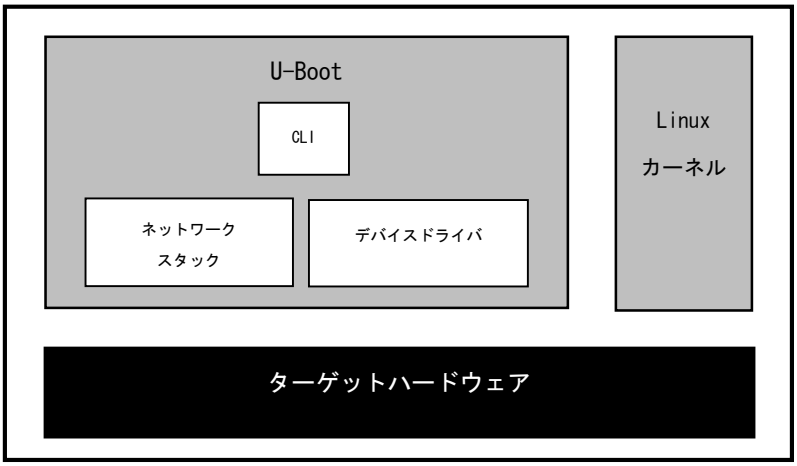


Fig 2.4-1　U-Boot アーキテクチャ

設定			U-Boot コンフィグレーションデータ						
ディスプレイ	解像度	色深度	CONFIG_MS104VGA_BPP	CONFIG_MS104VGA_RES	CONFIG_MS104VGA_OUTPUT_LCD	CONFIG_MS104VGA_OUTPUT_LCD_SIM	CONFIG_MS104VGA_OUTPUT_NTSC	CONFIG_MS104VGA_OUTPUT_SVIDEO	CONFIG_MS104VGA_OUTPUT_VGA
VGA	640x480	8	8	640480	0	0	0	0	1
		15	15	640480	0	0	0	0	1
		16	16	640480	0	0	0	0	1
	800x600	8	8	800600	0	0	0	0	1
		15	15	800600	0	0	0	0	1
		16	16	800600	0	0	0	0	1
LCD	640x480 ※1	8	8	640480	1	0	0	0	0
		15	15	640480	1	0	0	0	0
		16	16	640480	1	0	0	0	0
	800x600	8	8	800600	1	0	0	0	0
		15	15	800600	1	0	0	0	0
		16	16	800600	1	0	0	0	0
NTSC	640x480	8	8	640480	0	0	1	0	0
		15	15	640480	0	0	1	0	0
		16	16	640480	0	0	1	0	0
S-Video	640x480	8	8	640480	0	0	0	1	0
		15	15	640480	0	0	0	1	0
		16	16	640480	0	0	0	1	0
LCD & VGA	640×480	8	8	640480	0	1	0	0	1
	800×600	8	8	800600	0	1	0	0	1
LCD& NTSC	640×480	8	8	640480	0	1	1	0	0
LCD& S-Video	640×480	8	8	640480	0	1	0	1	0

Table 2.4-1　U-Boot コンフィグレーションデータ （include/configs/ms104sh4.h）

２．４．２　Linux

MS104-VGA/LCD 用に加えられたデバイスドライバはフレームバッファデバイスドライバ、タッチパネルデバイスドライバ、ブザーデバイスドライバがあります。
また、各モニタ出力の設定は、Linux 起動時にカーネルパラメータによって変更することができます。

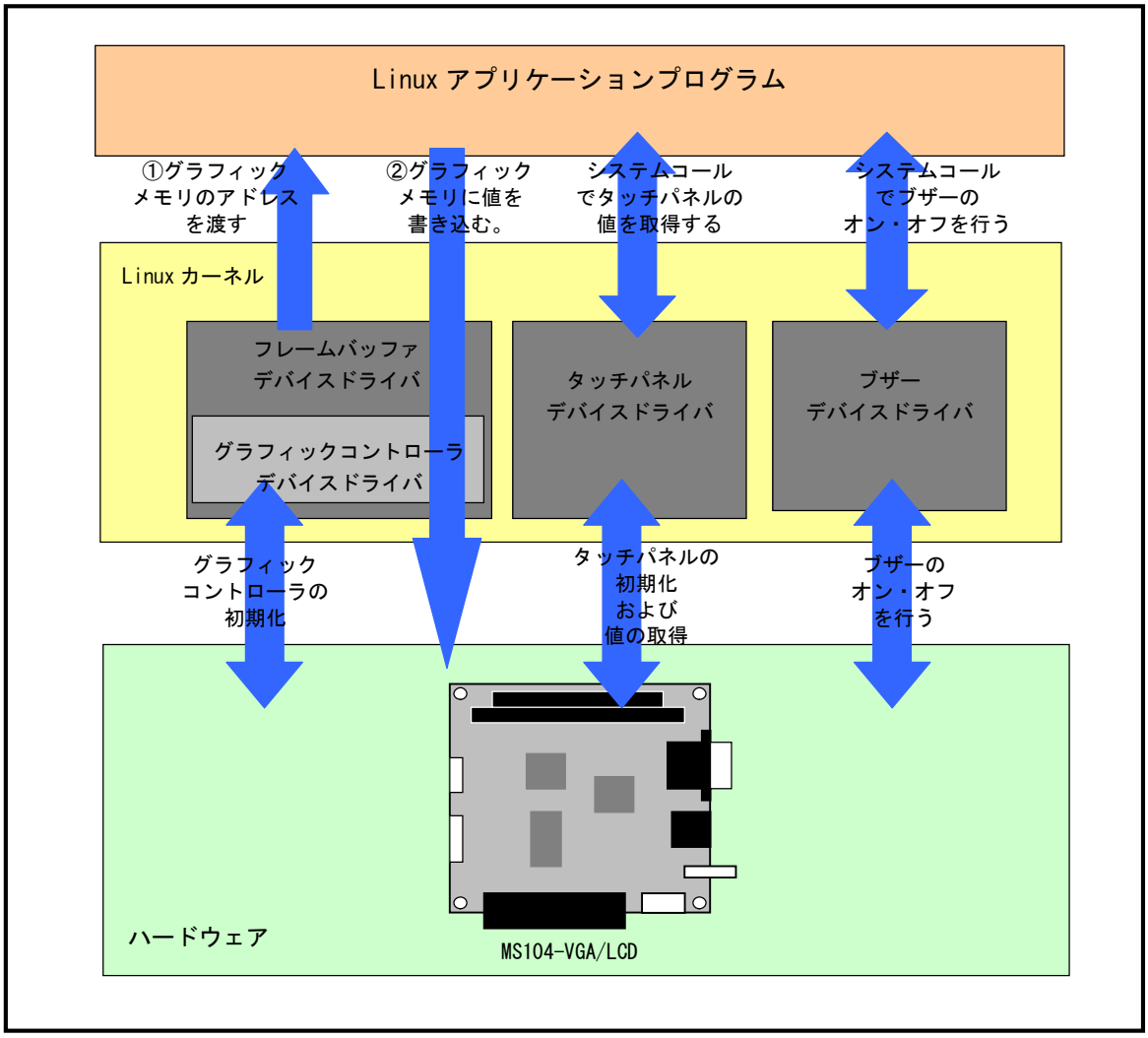


Fig 2.4-2　Linux デバイスドライバ

デバイス名	デバイスファイル	メジャー番号	マイナー番号	アドレス	割込み番号
フレームバッファ	/dev/fb0	29	0	0x400000	-
タッチパネル	/dev/input/event0	13	64	0xa00	11
ブザー	/dev/ms104vga_buzzer	10	63	0xa08	-

Table 2.4-2　Linux デバイスドライバ

各出力に対するカーネルパラメータの一覧を示します。

設定			Linux コンフィグレーションデータ						
ディスプレイ	解像度	色深度	CONFIG_MS1 04VGA_BPP	CONFIG_MS1 04VGA_RES	CONFIG_MS1 04VGA_OUTP UT_LCD	CONFIG_MS1 04VGA_OUTP UT_LCD_SIM	CONFIG_MS1 04VGA_OUTP UT_NTSC	CONFIG_MS1 04VGA_OUTP UT_SVIDEO	CONFIG_MS1 04VGA_OUTP UT_VGA
VGA	640x480	8	8	640x480	0	0	0	0	1
		16	16	640x480	0	0	0	0	1
	800x600	8	8	800x600	0	0	0	0	1
		16	16	800x600	0	0	0	0	1
LCD	640x480 ※1	8	8	640x480	1	0	0	0	0
		16	16	640x480	1	0	0	0	0
	800x600	8	8	800x600	1	0	0	0	0
		16	16	800x600	1	0	0	0	0
NTSC	640x480	8	8	640x480	0	0	1	0	0
		16	16	640x480	0	0	1	0	0
S-Video	640x480	8	8	640x480	0	0	0	1	0
		16	16	640x480	0	0	0	1	0
LCD & VGA	640×480	8	8	640x480	0	1	0	0	1
	800×600	8	8	800x600	0	1	0	0	1
LCD& NTSC	640×480	8	8	640x480	0	1	1	0	0
LCD& S-Video	640×480	8	8	640x480	0	1	0	1	0

Table 2.4-3　Linux デバイスドライバ出力対応

※1　弊社タッチパネル付 TFT-LCD モニタ「LCD-KIT-A03」では解像度が異なるため、ご使用いただけません。

３．MS104-VGA/LCD の起動

この章では、「MS104-VGA/LCD」と「MS104-SH4」を使用して、各モニタに DirectFB のデモプログラムを出力する手順を説明します。

３．１　MS104-VGA/LCD の動作環境

●ホスト PC

U-Boot/Linux のコンソール、及び、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 をご用意下さい。

●電源

MS104-VGA/LCD 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-VGA/LCD と合わせて使用するため、2A 程度の電源をご用意下さい。

●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 ケーブルをご用意下さい。

使用機器等		環　　境
PC/104 グラフィックボード		MS104-VGA/LCD
CPU ボード		MS104-SH4
HOST PC		PC/AT 互換機
	OS	Linux(推奨　Fedora)
	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	DVD-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
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

Table 3.3-1　MS104-VGA/LCD、MS104-SH4 の推奨動作環境

シリアル・ネットワークの設定は MS104-SH4 ソフトウェアマニュアルをご参照ください。

３．２　MS104-VGA/LCD の設定

MS104-SH4 用に MS104-VGA/LCD ボードの設定を行います。

① メモリアドレスを H’400000 に設定します。SW1 の 1 番を ON、2 番を OFF にして下さい。

② I/O アドレスを H’0a00 に設定します。SW1 の 3 番を OFF、4 番を ON、5 番を OFF、6 番を ON にして下さい。

③ PC/104 バスの BALE を有効にします。SW1 の 7 番を ON にして下さい。

④ インターフェースモードの設定を PC/104 互換モードにします。SW1 の 8 番を OFF にして下さい。

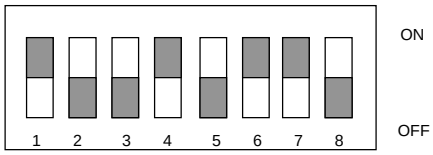


Fig 3.2-1　SW1 の設定

⑤ タッチパネルの割込みに IRQ4 を利用します。JP1 の 4 番にジャンパピンを接続して下さい。

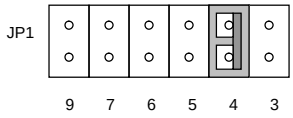


Fig 3.2-2　JP1 の設定

⑥ LCD コントローラ（EPSON 社製 S1D13506）を ISA バス（PC/104 バス）モードにします。JP3 の 1 番、3 番、4 番 15 番にジャンパピンを接続して下さい。

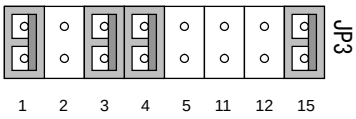


Fig 3.2-3　JP3 の設定

３．３　MS104-VGA/LCD の動作

この章では、各モニタに対し、色深度 16bpp 対応の DirectFB デモプログラムを起動する手順を示します。

３．３．１　VGA

解像度 800×600、色深度 16bpp 対応の Linux カーネルを起動し、DirectFB のデモプログラムを動作させます。
以下に MS104-SH4、MS104-VGA/LCD 及び VGA モニタの接続を示します。

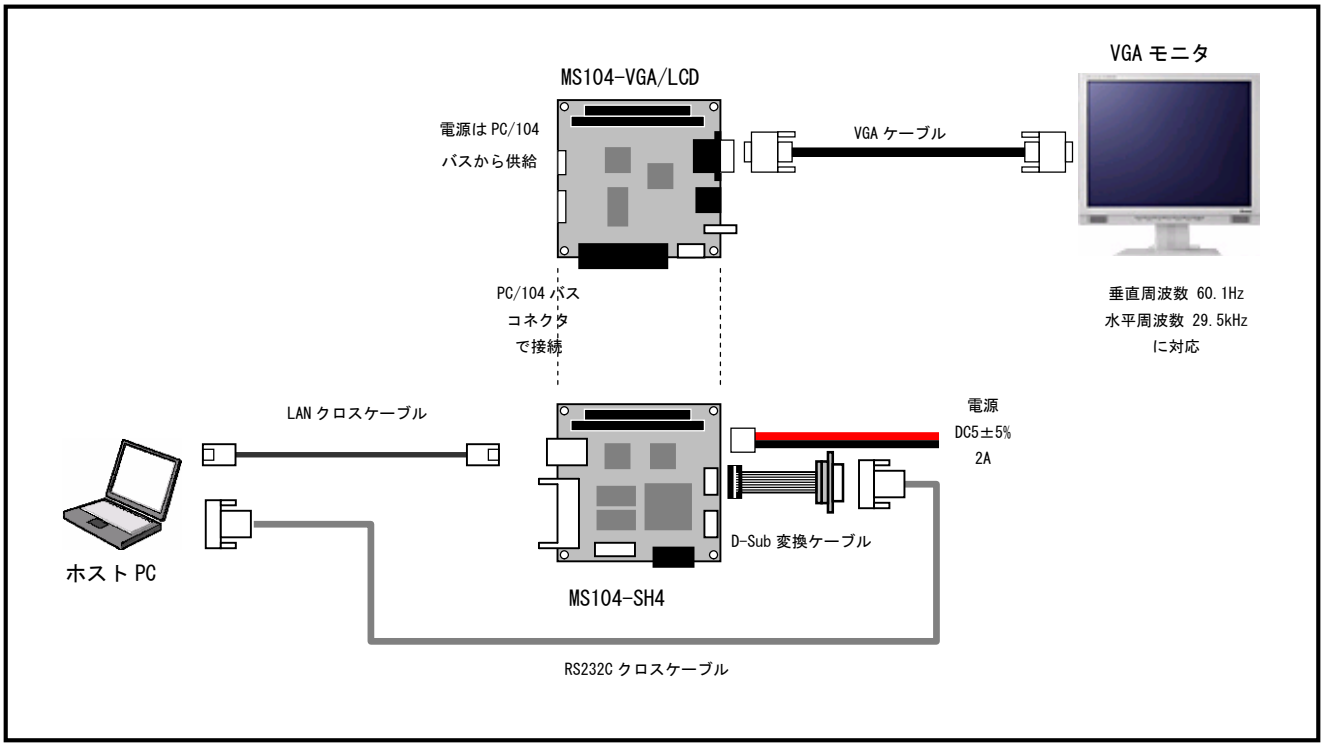


Fig 3.3-1　VGA モニタとの接続

①　「ディスプレイ=>VGA、解像度=>800x600、色深度=>16」の設定にコンフィグレーションされた U-Boot をフラッシュ ROM に書き込み、同じ設定の Linux カーネルイメージをホスト PC の TFTP サーバディレクトリにコピーして下さい。BMP ファイル ms104.bmp も TFTP サーバディレクトリにコピーして下さい。

② MS104- VGA/LCD と MS104-SH4 を「Fig 3.3-2　VGA モニタとの接続」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 ポートに接続して下さい。MS104-SH4 の電源を ON し、U-Boot の起動ログが表示されることを確認します。

```
U-Boot 2009.01 ( 3 月 27 2009 - 10:30:59)

CPU: SH4
BOARD: SH7750R ALPHAPROJECT MS104-SH4
DRAM:  32M
FLASH: 16MB
In:    serial
Out:   serial
Err:   serial
Net:   Hit any key to utoboot:  0 <-- 何かキーを入力する
=>
```

③ BMP ファイルを TFTP でダウンロードし表示します。

```
=> tftpboot 8c800000 ms104.bmp
Using MAC Address 00:0c:7b:20:xx:xx
TFTP from server 192.168.128.201; our IP address is 192.168.128.200
Filename 'ms104.bmp'.
Load address: 0x8c800000
Loading: #########
done
Bytes transferred = 132054 (203d6 hex)
=> bmp display
```

U-Boot のコンフィグレーション設定が正しければ、下記のロゴが VGA モニタに出力されます。

Fig 3.3-2　U-Boot ロゴ

④　Linux を TFTP でダウンロードし起動します。

```
=> tftpboot 8c800000 uImage-ms104sh4-vga
Using MAC Address 00:0c:7b:20:xx:xx
TFTP from server 192.168.128.201; our IP address is 192.168.128.200
Filename 'uImage-ms104sh4-vga'.
Load address: 0x8c800000
Loading: #################################################################
         #################################################################
         #################################################################
         ################################
done
Bytes transferred = 3337804 (32ee4c hex)
=> icache on ; bootm
Instruction Cache is ON
## Booting kernel from Legacy Image at 8c800000 ...
... 中略 ...

Welcome to Buildroot
uclibc login:
```

⑤　ルートでログインし、DirectFB のサンプルプログラム『df_andi』を実行します。

```
uclibc login: root
# df_andi

   ~~~~~~~~~~~~~~~~~~~~~~~~~~| DirectFB 1.2.7 |~~~~~~~~~~~~~~~~~~~~~~~~~~
        (c) 2001-2008  The world wide DirectFB Open Source Community
        (c) 2000-2004  Convergence (integrated media) GmbH
      ------------------------------------------------------------


(*) DirectFB/Core: Single Application Core. (2009-03-23 08:08)
(*) Direct/Thread: Started 'VT Switcher' (877) [CRITICAL OTHER/OTHER 0/0] <2093056>...
(*) DirectFB/Graphics: Generic Software Rasterizer 0.6 (directfb.org)
(*) DirectFB/Core/WM: Default 0.3 (directfb.org)
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 0) at offset 0 and pitch 1600.
 EINVAL
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 0) at offset 0 and pitch 1600.
(*) FBDev/Mode: Setting 800x600 RGB16
(*) FBDev/Mode: Switched to 800x600 (virtual 800x1200) at 16 bit (RGB16), pitch 1600
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 1) at offset 960000 and pitch
1600.
(*) Direct/Interface: Loaded 'FT2' implementation of 'IDirectFBFont'.
(*) Direct/Interface: Loaded 'PNG' implementation of 'IDirectFBImageProvider'.
(*) Direct/Interface: Loaded 'JPEG' implementation of 'IDirectFBImageProvider'.
```

VGA 上に以下の画面が出力されます。

Fig 3.3-3　DirectFB デモプログラム

３．３．２　LCD

解像度 800×600、色深度 16bpp 対応の Linux カーネルを起動し、DirectFB のデモプログラムを動作させます。
以下に MS104-SH4、MS104-VGA/LCD、MS104-LVDS/DVI 及び LCD-KIT-A03 の接続を示します。

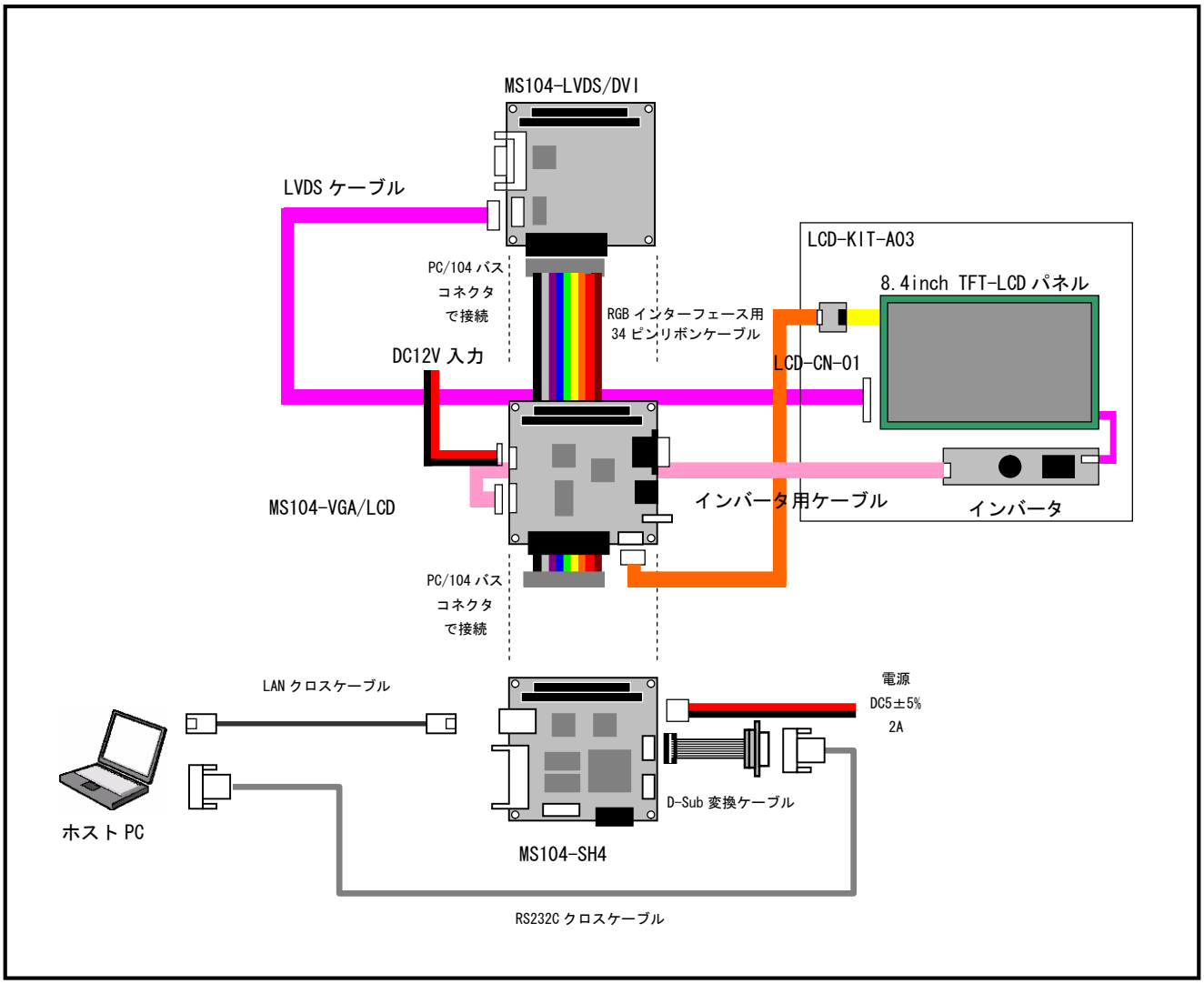


Fig 3.3-4　LCD-KIT-A03 との接続

①　「ディスプレイ=>LCD、解像度=>800x600、色深度=>16」の設定にコンフィグレーションされた U-Boot をフラッシュ ROM に書き込み、同じ設定の Linux カーネルイメージをホスト PC の TFTP サーバディレクトリにコピーして下さい。BMP ファイル ms104.bmp も TFTP サーバディレクトリにコピーして下さい。

② MS104- VGA/LCD と MS104-SH4 を「Fig 3.3-4　LCD-KIT-A03 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 ボ下さ 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 ポートに接続して下さい。MS104-SH4 の電源を ON し、U-Boot の起動ログが表示されることを確認します。

```
U-Boot 2009.01 ( 3 月 27 2009 - 10:30:59)

CPU: SH4
BOARD: SH7750R ALPHAPROJECT MS104-SH4
DRAM:  32M
FLASH: 16MB
In:    serial
Out:   serial
Err:   serial
Net:   Hit any key to utoboot:  0 <-- 何かキーを入力する
=>
```

③ BMP ファイルを TFTP でダウンロードし表示します。

```
=> tftpboot 8c800000 ms104.bmp
Using MAC Address 00:0c:7b:20:xx:xx
TFTP from server 192.168.128.201; our IP address is 192.168.128.200
Filename 'ms104.bmp'.
Load address: 0x8c800000
Loading: #########
done
Bytes transferred = 132054 (203d6 hex)
=> bmp display
```

U-Boot のコンフィグレーション設定が正しければ、LCD モニタ上に「Fig 3.3-3　U-Boot ロゴ」が出力されます。

④　Linux を TFTP でダウンロードし起動します。

```
=> tftpboot 8c800000 uImage-ms104sh4-vga
Using MAC Address 00:0c:7b:20:xx:xx
TFTP from server 192.168.128.201; our IP address is 192.168.128.200
Filename 'uImage-ms104sh4-vga'.
Load address: 0x8c800000
Loading: #################################################################
         #################################################################
         #################################################################
         ###############################
done
Bytes transferred = 3337804 (32ee4c hex)
=> icache on ; bootm
Instruction Cache is ON
## Booting kernel from Legacy Image at 8c800000 ...
... 中略 ...

Welcome to Buildroot
uclibc login:
```

⑤　ルートでログインし、DirectFB のサンプルプログラム『df_andi』を実行します。

```
uclibc login: root
# df_andi

   ~~~~~~~~~~~~~~~~~~~~~~~~~| DirectFB 1.2.7 |~~~~~~~~~~~~~~~~~~~~~~~~~
        (c) 2001-2008  The world wide DirectFB Open Source Community
        (c) 2000-2004  Convergence (integrated media) GmbH
      ------------------------------------------------------------------


(*) DirectFB/Core: Single Application Core. (2009-03-23 08:08)
(*) Direct/Thread: Started 'VT Switcher' (877) [CRITICAL OTHER/OTHER 0/0] <2093056>...
(*) DirectFB/Graphics: Generic Software Rasterizer 0.6 (directfb.org)
(*) DirectFB/Core/WM: Default 0.3 (directfb.org)
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 0) at offset 0 and pitch 1600.
 EINVAL
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 0) at offset 0 and pitch 1600.
(*) FBDev/Mode: Setting 800x600 RGB16
(*) FBDev/Mode: Switched to 800x600 (virtual 800x1200) at 16 bit (RGB16), pitch 1600
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 1) at offset 960000 and pitch
1600.
(*) Direct/Interface: Loaded 'FT2' implementation of 'IDirectFBFont'.
(*) Direct/Interface: Loaded 'PNG' implementation of 'IDirectFBImageProvider'.
(*) Direct/Interface: Loaded 'JPEG' implementation of 'IDirectFBImageProvider'.
```

⑤　LCD モニタ上に「Fig 3.3-3　DirectFB デモプログラム」が出力されます。

３．３．３　NTSC

解像度 640×480、色深度 16bpp 対応の Linux カーネルを起動し、DirectFB のデモプログラムを動作させます。
以下に MS104-SH4、MS104-VGA/LCD 及び TV モニタの接続を示します。

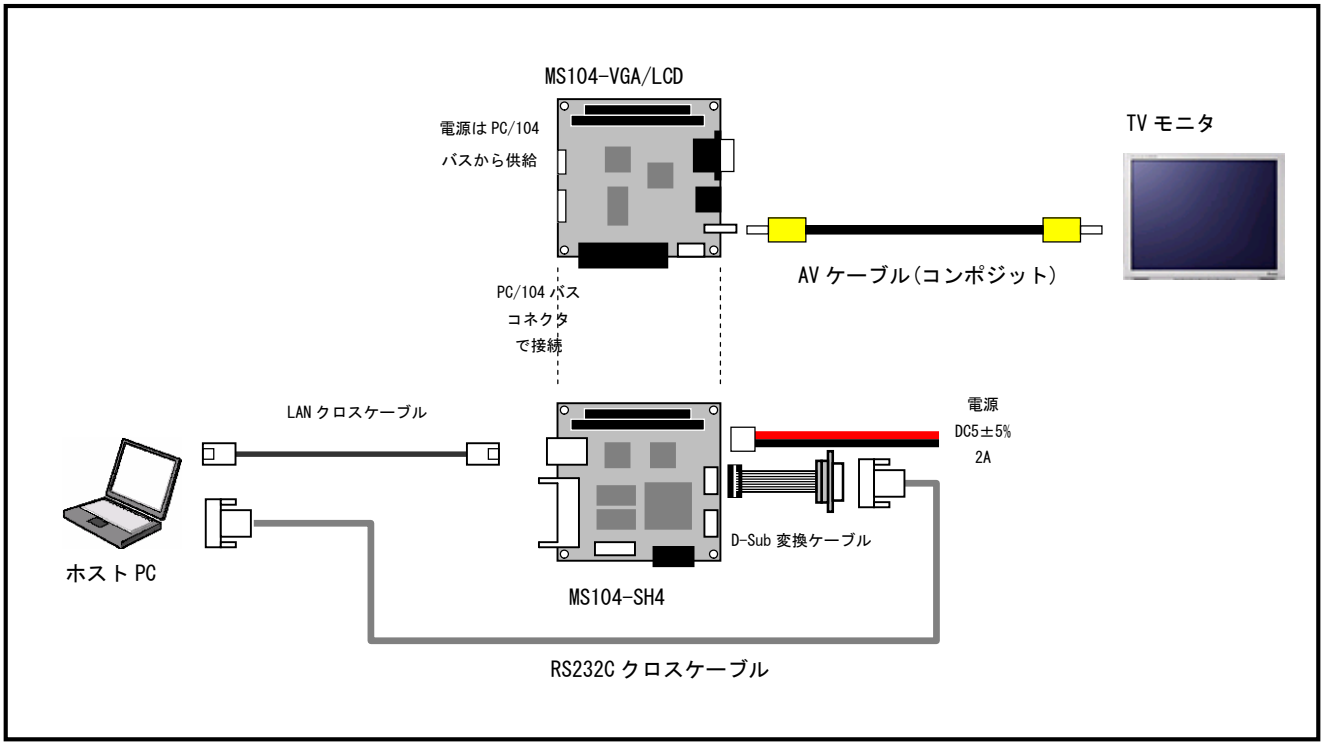


Fig 3.3-5　TV モニタとの接続

① 「ディスプレイ=>NTSC、解像度=>640x480、色深度=>16」の設定にコンフィグレーションされ下さ Boot をフラッシュ ROM に書き込み、同じ設定の Linux カーネルイメージをホスト PC の TFTP サーバディレクトリにコピーして下さい。BMP ファイル ms104.bmp も TFTP サーバディレクトリにコピーして下さい。

② MS104- VGA/LCD と MS104-SH4 を「Fig 3.3-5　TV モニタとの接続」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 ポートに接続して下さい。MS104-SH4 の電源を ON し、U-Boot の起動ログが表示されることを確認します。

```
U-Boot 2009.01 ( 3 月 27 2009 - 10:30:59)

CPU: SH4
BOARD: SH7750R ALPHAPROJECT MS104-SH4
DRAM:  32M
FLASH: 16MB
In:    serial
Out:   serial
Err:   serial
Net:   Hit any key to utoboot:  0 <-- 何かキーを入力する
=>
```

③　BMP ファイルを TFTP でダウンロードし表示します。

```
=> tftpboot 8c800000 ms104-640.bmp
Using MAC Address 00:0c:7b:20:xx:xx
TFTP from server 192.168.128.201; our IP address is 192.168.128.200
Filename 'ms104-640.bmp'.
Load address: 0x8c800000
Loading: #########
done
Bytes transferred = 132054 (203d6 hex)
=> bmp display
```

U-Boot のコンフィグレーション設定が正しければ、TV モニタ上に「Fig 3.3-3　U-Boot ロゴ」が出力されます。

④　Linux を TFTP でダウンロードし起動します。

```
=> tftpboot 8c800000 uImage-ms104sh4-vga
Using MAC Address 00:0c:7b:20:xx:xx
TFTP from server 192.168.128.201; our IP address is 192.168.128.200
Filename 'uImage-ms104sh4-vga'.
Load address: 0x8c800000
Loading: ################################################################
         ################################################################
         ################################################################
         ##############################
done
Bytes transferred = 3337804 (32ee4c hex)
=> icache on ; bootm
Instruction Cache is ON
## Booting kernel from Legacy Image at 8c800000 ...
... 中略 ...

Welcome to Buildroot
uclibc login:
```

⑤　ルートでログインし、DirectFB のサンプルプログラム『df_andi』を実行します。

```
uclibc login: root
# df_andi

   ~~~~~~~~~~~~~~~~~~~~~~~~~~| DirectFB 1.2.7 |~~~~~~~~~~~~~~~~~~~~~~~~~~
        (c) 2001-2008  The world wide DirectFB Open Source Community
        (c) 2000-2004  Convergence (integrated media) GmbH
      ------------------------------------------------------------


(*) DirectFB/Core: Single Application Core. (2009-03-23 08:08)
(*) Direct/Thread: Started 'VT Switcher' (877) [CRITICAL OTHER/OTHER 0/0] <2093056>...
(*) DirectFB/Graphics: Generic Software Rasterizer 0.6 (directfb.org)
(*) DirectFB/Core/WM: Default 0.3 (directfb.org)
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 0) at offset 0 and pitch 1600.
 EINVAL
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 0) at offset 0 and pitch 1600.
(*) FBDev/Mode: Setting 800x600 RGB16
(*) FBDev/Mode: Switched to 800x600 (virtual 800x1200) at 16 bit (RGB16), pitch 1600
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 1) at offset 960000 and pitch
1600.
(*) Direct/Interface: Loaded 'FT2' implementation of 'IDirectFBFont'.
(*) Direct/Interface: Loaded 'PNG' implementation of 'IDirectFBImageProvider'.
(*) Direct/Interface: Loaded 'JPEG' implementation of 'IDirectFBImageProvider'.
```

TV モニタ上に「Fig 3.3-3　DirectFB デモプログラム」が出力されます。

３．３．４　S-Video

解像度 640×480、色深度 16bpp 対応の Linux カーネルを起動し、DirectFB のサンプルプログラムを動作させます。
以下に MS104-SH4、MS104-VGA/LCD 及び S-Video モニタの接続を示します。

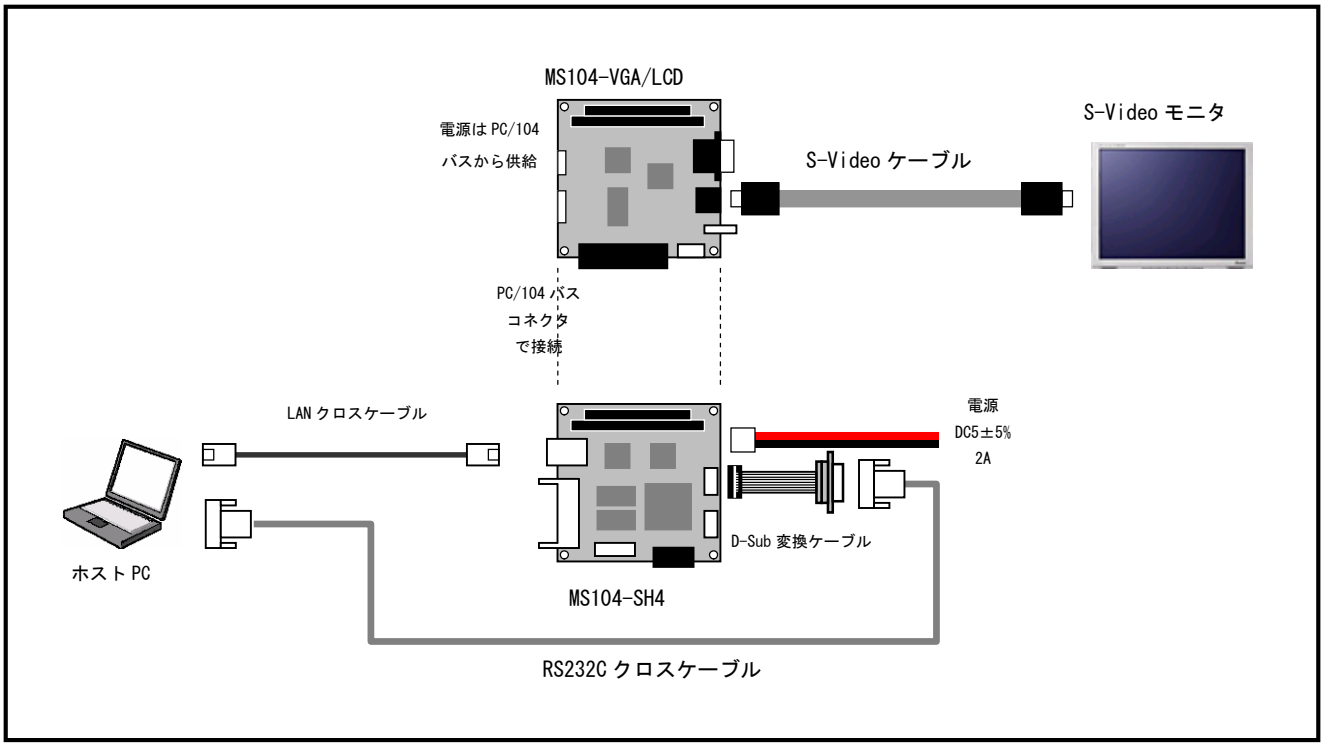


Fig 3.3-6　S-Video モニタとの接続

① 「ディスプレイ=>S-Video、解像度=>640x480、色深度=>16」の設定にコンフィグレーションされた U-Boot をフラッシュ ROM に書き込み、同じ設定の Linux カーネルイメージをホスト PC の TFTP サーバディレクトリにコピーして下さい。BMP ファイル ms104.bmp も TFTP サーバディレクトリにコピーして下さい。

② MS104- VGA/LCD と MS104-SH4 を「Fig 3.3-7　S-Video モニタとの接続」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 ポートに接続して下さい。MS104-SH4 の電源を ON し、U-Boot の起動ログが表示されることを確認します。

```
U-Boot 2009.01 ( 3 月 27 2009 - 10:30:59)

CPU: SH4
BOARD: SH7750R ALPHAPROJECT MS104-SH4
DRAM:  32M
FLASH: 16MB
In:    serial
Out:   serial
Err:   serial
Net:   Hit any key to utoboot:  0 <-- 何かキーを入力する
=>
```

③　BMP ファイルを TFTP でダウンロードし表示します。

```
=> tftpboot 8c800000 ms104-640.bmp
Using MAC Address 00:0c:7b:20:xx:xx
TFTP from server 192.168.128.201; our IP address is 192.168.128.200
Filename 'ms104-640.bmp'.
Load address: 0x8c800000
Loading: #########
done
Bytes transferred = 132054 (203d6 hex)
=> bmp display
```

U-Boot のコンフィグレーション設定が正しければ、S-Video モニタ上に「Fig 3.3-3　U-Boot ロゴ」が出力されます。

④　Linux を TFTP でダウンロードし起動します。

```
=> tftpboot 8c800000 uImage-ms104sh4-vga
Using MAC Address 00:0c:7b:20:xx:xx
TFTP from server 192.168.128.201; our IP address is 192.168.128.200
Filename 'uImage-ms104sh4-vga'.
Load address: 0x8c800000
Loading: ################################################################
         ################################################################
         ################################################################
         ##############################
done
Bytes transferred = 3337804 (32ee4c hex)
=> icache on ; bootm
Instruction Cache is ON
## Booting kernel from Legacy Image at 8c800000 ...
... 中略 ...

Welcome to Buildroot
uclibc login:
```

⑤　ルートでログインし、DirectFB のサンプルプログラム『df_andi』を実行します。

```
uclibc login: root
# df_andi

   ~~~~~~~~~~~~~~~~~~~~~~~~~~| DirectFB 1.2.7 |~~~~~~~~~~~~~~~~~~~~~~~~~~
        (c) 2001-2008  The world wide DirectFB Open Source Community
        (c) 2000-2004  Convergence (integrated media) GmbH
      ------------------------------------------------------------


(*) DirectFB/Core: Single Application Core. (2009-03-23 08:08)
(*) Direct/Thread: Started 'VT Switcher' (877) [CRITICAL OTHER/OTHER 0/0] <2093056>...
(*) DirectFB/Graphics: Generic Software Rasterizer 0.6 (directfb.org)
(*) DirectFB/Core/WM: Default 0.3 (directfb.org)
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 0) at offset 0 and pitch 1600.
 EINVAL
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 0) at offset 0 and pitch 1600.
(*) FBDev/Mode: Setting 800x600 RGB16
(*) FBDev/Mode: Switched to 800x600 (virtual 800x1200) at 16 bit (RGB16), pitch 1600
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 1) at offset 960000 and pitch
1600.
(*) Direct/Interface: Loaded 'FT2' implementation of 'IDirectFBFont'.
(*) Direct/Interface: Loaded 'PNG' implementation of 'IDirectFBImageProvider'.
(*) Direct/Interface: Loaded 'JPEG' implementation of 'IDirectFBImageProvider'.
```

S-Video モニタ上に「Fig 3.3-3　DirectFB デモプログラム」が出力されます。

３．３．５　LCD・VGA 同時出力

解像度 800×600、色深度 8bpp 対応の Linux カーネルを起動し、DirectFB のデモプログラムを動作させます。
以下に MS104-SH4、MS104-VGA/LCD、MS104-LVDS/DVI、LCD-KIT-A03 及び VGA モニタの接続を示します。

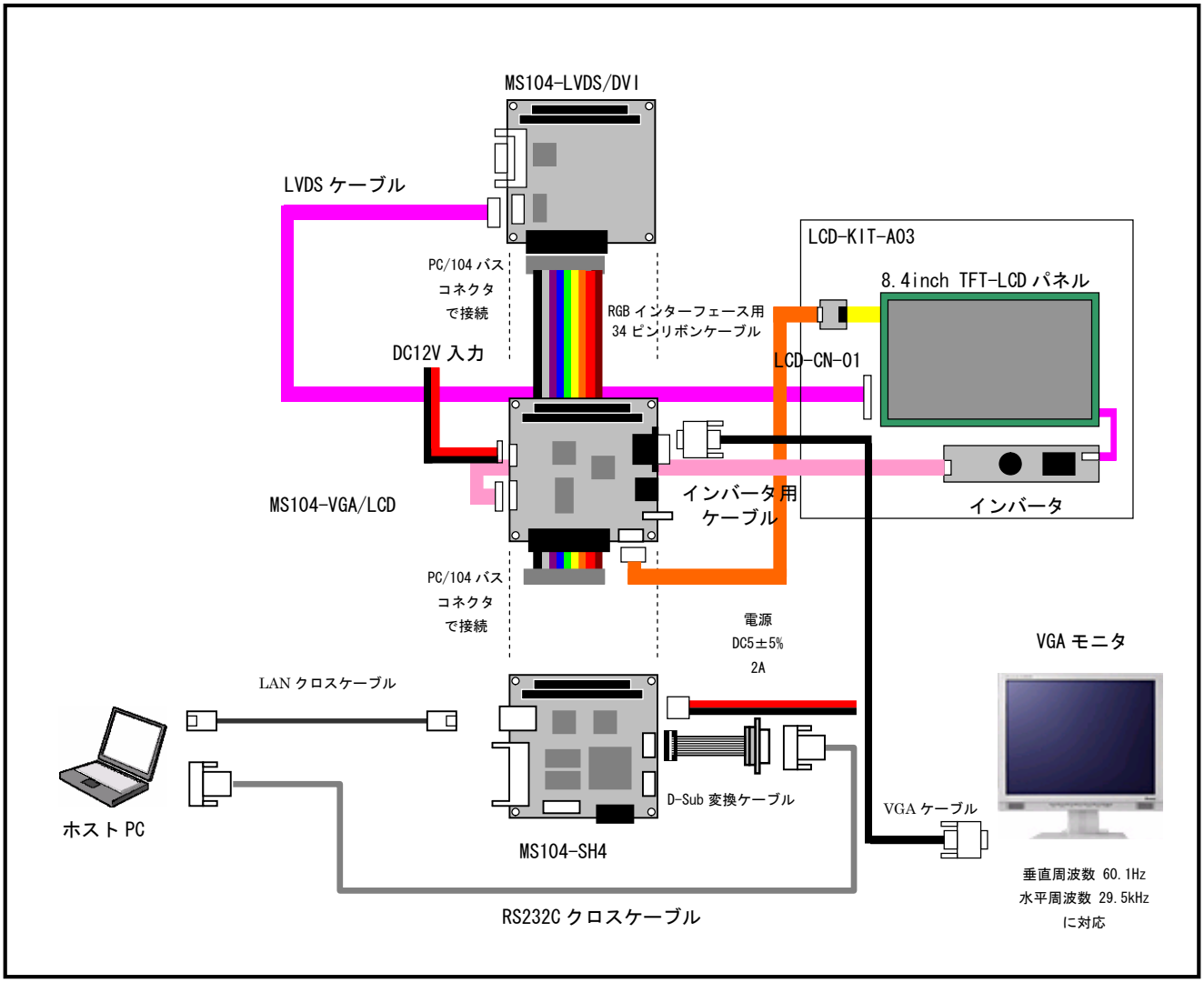


Fig 3.3-7　LCD-KIT-A03・VGA モニタとの接続

①　「ディスプレイ=>LCD&VGA、解像度=>800x600、色深度=>8」の設定にコンフィグレーションされた U-Boot をフラッシュ ROM に書き込み、同じ設定の Linux カーネルイメージをホスト PC の TFTP サーバディレクトリにコピーして下さい。BMP ファイル ms104.bmp も TFTP サーバディレクトリにコピーして下さい。

② MS104- VGA/LCD と MS104-SH4 を「Fig 3.3-7　LCD-KIT-A03・VGA モニタとの接続」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 ポートに接続して下さい。MS104-SH4 の電源を ON し、U-Boot の起動ログが表示されることを確認します。

```
U-Boot 2009.01 ( 3 月 27 2009 - 10:30:59)

CPU: SH4
BOARD: SH7750R ALPHAPROJECT MS104-SH4
DRAM:  32M
FLASH: 16MB
In:    serial
Out:   serial
Err:   serial
Net:   Hit any key to utoboot:  0 <-- 何かキーを入力する
=>
```

③ BMP ファイルを TFTP でダウンロードし表示します。

```
=> tftpboot 8c800000 ms104-640.bmp
Using MAC Address 00:0c:7b:20:xx:xx
TFTP from server 192.168.128.201; our IP address is 192.168.128.200
Filename 'ms104-640.bmp'.
Load address: 0x8c800000
Loading: #########
done
Bytes transferred = 132054 (203d6 hex)
=> bmp display
```

U-Boot のコンフィグレーション設定が正しければ、LCD モニタ及び VGA モニタ上に「Fig 3.3－2　U－Boot ロゴ」が表示されます。

④　Linux を TFTP でダウンロードし起動します。

```
=> tftpboot 8c800000 uImage-ms104sh4-vga
Using MAC Address 00:0c:7b:20:xx:xx
TFTP from server 192.168.128.201; our IP address is 192.168.128.200
Filename 'uImage-ms104sh4-vga'.
Load address: 0x8c800000
Loading: #################################################################
         #################################################################
         #################################################################
         ###############################
done
Bytes transferred = 3337804 (32ee4c hex)
=> icache on ; bootm
Instruction Cache is ON
## Booting kernel from Legacy Image at 8c800000 ...
... 中略 ...

Welcome to Buildroot
uclibc login:
```

⑤　ルートでログインし、DirectFB のサンプルプログラム『df_andi』を実行します。

```
uclibc login: root
# df_andi

   ~~~~~~~~~~~~~~~~~~~~~~~~~| DirectFB 1.2.7 |~~~~~~~~~~~~~~~~~~~~~~~~~
        (c) 2001-2008  The world wide DirectFB Open Source Community
        (c) 2000-2004  Convergence (integrated media) GmbH
      ------------------------------------------------------------------


(*) DirectFB/Core: Single Application Core. (2009-03-23 08:08)
(*) Direct/Thread: Started 'VT Switcher' (877) [CRITICAL OTHER/OTHER 0/0] <2093056>...
(*) DirectFB/Graphics: Generic Software Rasterizer 0.6 (directfb.org)
(*) DirectFB/Core/WM: Default 0.3 (directfb.org)
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 0) at offset 0 and pitch 1600.
 EINVAL
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 0) at offset 0 and pitch 1600.
(*) FBDev/Mode: Setting 800x600 RGB16
(*) FBDev/Mode: Switched to 800x600 (virtual 800x1200) at 16 bit (RGB16), pitch 1600
(*) FBDev/Surface: Allocated 800x600 16 bit RGB16 buffer (index 1) at offset 960000 and pitch
1600.
(*) Direct/Interface: Loaded 'FT2' implementation of 'IDirectFBFont'.
(*) Direct/Interface: Loaded 'PNG' implementation of 'IDirectFBImageProvider'.
(*) Direct/Interface: Loaded 'JPEG' implementation of 'IDirectFBImageProvider'.
```

LCD モニタ及び VGA モニタ上に「Fig 3.3-3　DirectFB デモプログラム」が出力されます。

4．U-Boot

4．1　U-Boot のコンフィグレーション

U-Boot のコンフィグレーションは U-Boot ソースディレクトリ以下の、『include/configs/ms104sh4.h』ファイルを編集し、再コンパイルをすることにより行います。MS104-VGA/LCD 用の設定は 120 行目付近にあります。

```
... 中略 ...
#define CONFIG_MS104VGA

#ifdef CONFIG_MS104VGA
 #define CONFIG_CMD_BMP
 #define CONFIG_VIDEO
 #define CONFIG_VIDEO_SED13806
 #define CONFIG_VIDEO_SED13806_16BPP

 #define CONFIG_MS104VGA_BPP 16      /* 8 or 15 or 16 */
 #define CONFIG_MS104VGA_RES 800600 /* 640480 or 800600  */
 #define CONFIG_MS104VGA_OUTPUT_LCD      0
 #define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
 #define CONFIG_MS104VGA_OUTPUT_NTSC     0
 #define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
 #define CONFIG_MS104VGA_OUTPUT_VGA      1

 #if CONFIG_MS104VGA_RES != 640480 && CONFIG_MS104VGA_RES != 800600
  #error CONFIG_MS104VGA must be 640480 or 800600
 #endif

 #if CONFIG_MS104VGA_OUTPUT_NTSC + ¥
     CONFIG_MS104VGA_OUTPUT_SVIDEO + ¥
     CONFIG_MS104VGA_OUTPUT_VGA + ¥
     CONFIG_MS104VGA_OUTPUT_LCD != 1
  #error CONFIG_MS104VGA LCD or NTSC or SVIDEO or RGB select 1.
 #endif

 #if CONFIG_MS104VGA_RES == 800600 && ¥
     (CONFIG_MS104VGA_OUTPUT_NTSC || CONFIG_MS104VGA_OUTPUT_SVIDEO)
  #error NSTC, SVIDEO must be 640x480
 #endif

 #if CONFIG_MS104VGA_OUTPUT_LCD_SIM && CONFIG_MS104VGA_BPP != 8
  #error Simul output support 8 bpp only
 #endif
#endif
```

『CONFIG_MS104VGA』のディレクティブを有効にします。

```
#define CONFIG_MS104VGA
```

以下のマクロを編集することにより設定をします。

```
#define CONFIG_MS104VGA_BPP 16      /* 8 or 15 or 16 */
#define CONFIG_MS104VGA_RES 800600 /* 640480 or 800600  */
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      1
```

４．２　U-Boot 設定例

以下に可能な組み合わせの一覧を示します。これ以外の設定をした場合の動作は保証できません。

「ディスプレイ=>VGA、解像度=>640x480、色深度=>8」

```
#define CONFIG_MS104VGA_BPP 8
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      1
```

「ディスプレイ=>VGA、解像度=>640x480、色深度=>15」

```
#define CONFIG_MS104VGA_BPP 15
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      1
```

「ディスプレイ=>VGA、解像度=>640x480、色深度=>16」

```
#define CONFIG_MS104VGA_BPP 16
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      1
```

「ディスプレイ=>VGA、解像度=>800x600、色深度=>8」

```
#define CONFIG_MS104VGA_BPP 8
#define CONFIG_MS104VGA_RES 800600
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      1
```

「ディスプレイ=>VGA、解像度=>800x600、色深度=>15」

```
#define CONFIG_MS104VGA_BPP 15
#define CONFIG_MS104VGA_RES 800600
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      1
```

「ディスプレイ=>VGA、解像度=>800x600、色深度=>16」

```
#define CONFIG_MS104VGA_BPP 16
#define CONFIG_MS104VGA_RES 800600
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      1
```

「ディスプレイ=>LCD、解像度=>640x480、色深度=>8」

```
#define CONFIG_MS104VGA_BPP 8
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      1
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>LCD、解像度=>640x480、色深度=>15」

```
#define CONFIG_MS104VGA_BPP 15
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      1
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>LCD、解像度=>640x480、色深度=>16」

```
#define CONFIG_MS104VGA_BPP 16
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      1
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>LCD、解像度=>800x600、色深度=>8」

```
#define CONFIG_MS104VGA_BPP 8
#define CONFIG_MS104VGA_RES 800600
#define CONFIG_MS104VGA_OUTPUT_LCD      1
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>LCD、解像度=>800x600、色深度=>15」

```
#define CONFIG_MS104VGA_BPP 15
#define CONFIG_MS104VGA_RES 800600
#define CONFIG_MS104VGA_OUTPUT_LCD      1
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>LCD、解像度=>800x600、色深度=>16」

```
#define CONFIG_MS104VGA_BPP 16
#define CONFIG_MS104VGA_RES 800600
#define CONFIG_MS104VGA_OUTPUT_LCD      1
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>NTSC、解像度=>640x480、色深度=>8」

```
#define CONFIG_MS104VGA_BPP 8
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     1
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>NTSC、解像度=>640x480、色深度=>15」

```
#define CONFIG_MS104VGA_BPP 15
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     1
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>NTSC、解像度=>640x480、色深度=>16」

```
#define CONFIG_MS104VGA_BPP 16
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     1
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>S-Video、解像度=>640x480、色深度=>8」

```
#define CONFIG_MS104VGA_BPP 8
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   1
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>S-Video、解像度=>640x480、色深度=>15」

```
#define CONFIG_MS104VGA_BPP 15
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   1
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>S-Video、解像度=>640x480、色深度=>16」

```
#define CONFIG_MS104VGA_BPP 16
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  0
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   1
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>LCD&VGA、解像度=>640x480、色深度=>8」

```
#define CONFIG_MS104VGA_BPP 8
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  1
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      1
```

「ディスプレイ=>LCD&VGA、解像度=>800x600、色深度=>8」

```
#define CONFIG_MS104VGA_BPP 8
#define CONFIG_MS104VGA_RES 800600
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  1
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      1
```

「ディスプレイ=>LCD&NTSC、解像度=>640x480、色深度=>8」

```
#define CONFIG_MS104VGA_BPP 8
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  1
#define CONFIG_MS104VGA_OUTPUT_NTSC     1
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   0
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

「ディスプレイ=>LCD&S-Video、解像度=>640x480、色深度=>8」

```
#define CONFIG_MS104VGA_BPP 8
#define CONFIG_MS104VGA_RES 640480
#define CONFIG_MS104VGA_OUTPUT_LCD      0
#define CONFIG_MS104VGA_OUTPUT_LCD_SIM  1
#define CONFIG_MS104VGA_OUTPUT_NTSC     0
#define CONFIG_MS104VGA_OUTPUT_SVIDEO   1
#define CONFIG_MS104VGA_OUTPUT_VGA      0
```

４．３　U-Boot のコンパイル

① U-Boot のソースディレクトリに移動します。

```
[guest@LinuxKit ~]$ cd linuxkit-ms104sh4/u-boot-2009.03-alp/
[guest@LinuxKit u-boot-2009.03-alp]$
```

② 『4.2U-Boot 設定例』を参照し『include/configs/ms104sh4.h』ファイルを編集します。

③ make します。

```
[guest@LinuxKit u-boot-2009.03-alp]$ make ms104sh4_config
Configuring for ms104sh4 board...
[guest@LinuxKit u-boot-2009.03-alp]$ make
Generating include/autoconf.mk.dep
... 中略 ...
sh4-linux-objcopy --gap-fill=0xff -O binary u-boot u-boot.bin
[guest@LinuxKit u-boot-2009.03-alp]$
```

④ 生成されたバイナリファイルを確認します。

```
[guest@LinuxKit u-boot-2009.01]$ ls u-boot.bin
u-boot.bin
[guest@LinuxKit u-boot-2009.01]$
```

U-Boot のフラッシュ ROM への書き込み方法は「MS104-SH4 ソフトウェアマニュアル」をご覧下さい。

５．Linux

５．１　Linux カーネルのコンフィグレーション

Linux カーネル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。
Linux カーネルのコンフィグレーションにつきましては『MS104-SH4 ソフトウェアマニュアル』をご参照下さい。

① Linux カーネルのソースディレクトリに移動します。

```
[guest@LinuxKit ~]$ cd linuxkit-ms104sh4/linux-2.6.28.8-alp/
[guest@LinuxKit linux-2.6.28.8-alp]$
```

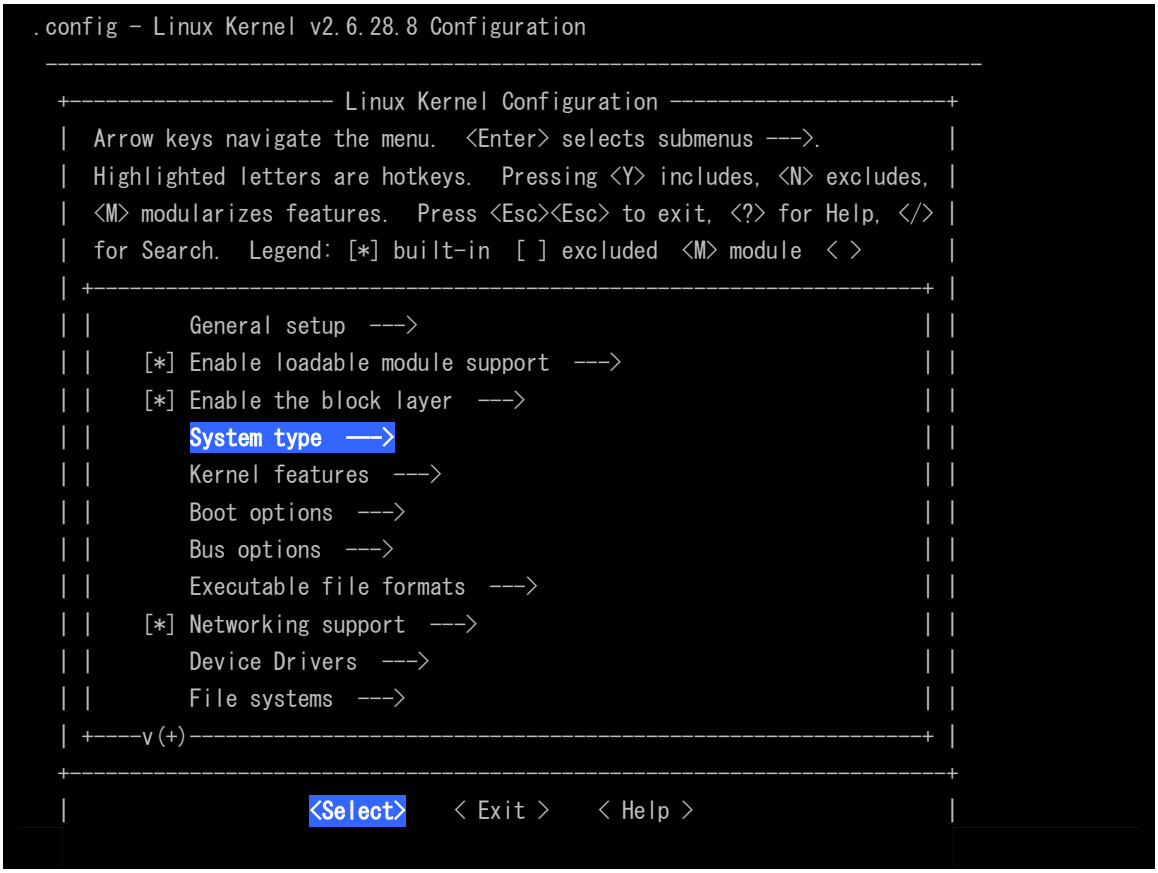
② MS104-VGA/LCD 用の設定を呼び出します。

```
[guest@LinuxKit linux-2.6.28.8-alp]$ ARCH=sh make ms104sh4_vga_defconfig
#
# configuration written to .config
#
[guest@LinuxKit linux-2.6.28.8-alp]$
```

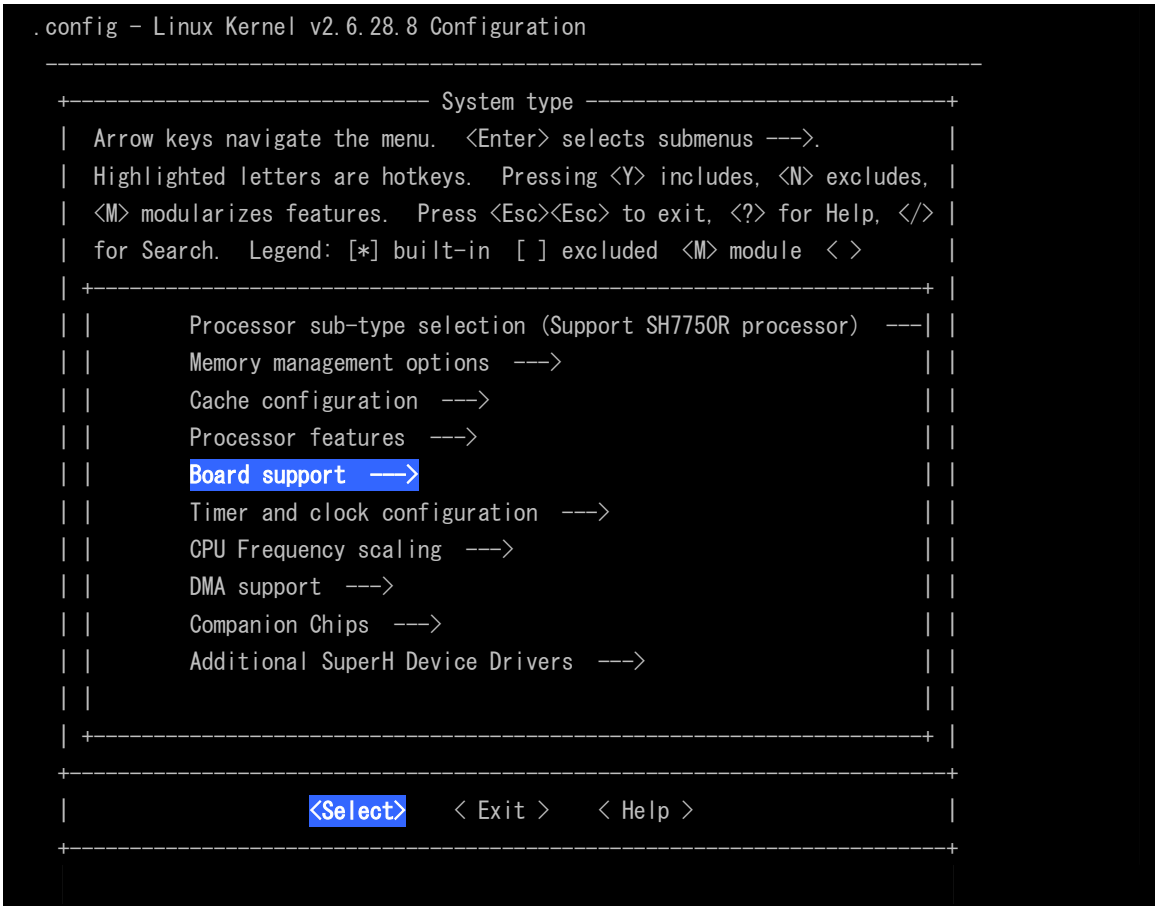
③ MS104-VGA/LCD 用の更なるコンフィグレーションは 『make menuconfig』を実行します。

```
[guest@LinuxKit linux-2.6.28.8-alp]$ ARCH=sh make menuconfig
```

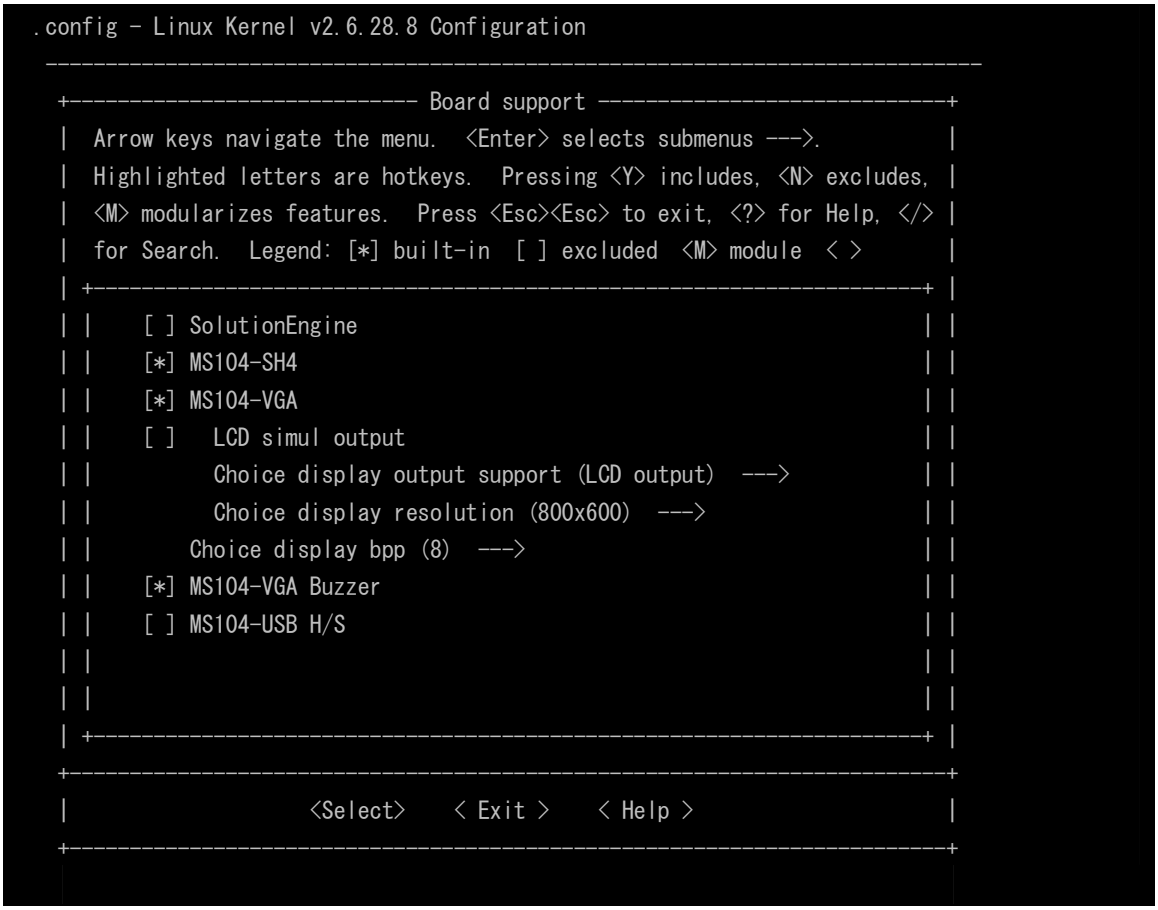
④　『System type』を選択してください。



⑤　『Board support』を選択します。



MS104-VGA/LCD のコンフィグレーションが表示されます。



５．２　Linux カーネル設定例

以下に可能な組み合わせの一覧を示します。これ以外の設定をした場合の動作は保証できません。

「ディスプレイ=>VGA、解像度=>640x480、色深度=>8」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (VGA output)  --->  | |
    | |          Choice display resolution (640x480)  --->        | |
    | |        Choice display bpp (8)  --->                       | |
```

「ディスプレイ=>VGA、解像度=>640x480、色深度=>16」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (VGA output)  --->  | |
    | |          Choice display resolution (640x480)  --->        | |
    | |        Choice display bpp (16)  --->                      | |
```

「ディスプレイ=>VGA、解像度=>800x600、色深度=>8」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (VGA output)  --->  | |
    | |          Choice display resolution (640x480)  --->        | |
    | |        Choice display bpp (8)  --->                       | |
```

「ディスプレイ=>VGA、解像度=>800x600、色深度=>16」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (VGA output)  --->  | |
    | |          Choice display resolution (800x600)  --->        | |
    | |        Choice display bpp (86)  --->                      | |
```

「ディスプレイ=>LCD、解像度=>640x480、色深度=>8」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (LCD output)  --->  | |
    | |          Choice display resolution (640x480)  --->        | |
    | |        Choice display bpp (8)  --->                       | |
```

「ディスプレイ=>LCD、解像度=>640x480、色深度=>16」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (LCD output)  --->  | |
    | |          Choice display resolution (640x480)  --->        | |
    | |        Choice display bpp (16)  --->                      | |
```

「ディスプレイ=>LCD、解像度=>800x600、色深度=>8」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (LCD output)  --->  | |
    | |          Choice display resolution (800x600)  --->        | |
    | |        Choice display bpp (8)  --->                       | |
```

「ディスプレイ=>LCD、解像度=>800x600、色深度=>16」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (LCD output)  --->  | |
    | |          Choice display resolution (800x600)  --->        | |
    | |        Choice display bpp (16)  --->                      | |
```

「ディスプレイ=>NTSC、解像度=>640x480、色深度=>8」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (NTSC output)  --->  | |
    | |          Choice display resolution (640x480)  --->        | |
    | |        Choice display bpp (8)  --->                       | |
```

「ディスプレイ=>NTSC、解像度=>640x480、色深度=>16」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (NTSC output)  --->  | |
    | |          Choice display resolution (640x480)  --->        | |
    | |        Choice display bpp (16)  --->                      | |
```

「ディスプレイ=>S-Video、解像度=>640x480、色深度=>8」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (S-Video output)  --->  | |
    | |          Choice display resolution (640x480)  --->        | |
    | |        Choice display bpp (8)  --->                       | |
```

「ディスプレイ=>S-Video、解像度=>640x480、色深度=>16」

```
    | |    [*] MS104-VGA                                          | |
    | |    [ ]   LCD simul output                                 | |
    | |          Choice display output support (S-Video output)  --->  | |
    | |          Choice display resolution (640x480)  --->        | |
    | |        Choice display bpp (16)  --->                      | |
```

「ディスプレイ=>LCD&VGA、解像度=>640x480、色深度=>8」

```
    | |    [*] MS104-VGA                                              | |
    | |    [*]   LCD simul output                                     | |
    | |          Choice display output support (VGA output)  --->     | |
    | |          Choice display resolution (640x480)  --->            | |
    | |        Choice display bpp (8)  --->                           | |
```

「ディスプレイ=>LCD&VGA、解像度=>800x600、色深度=>8」

```
    | |    [*] MS104-VGA                                              | |
    | |    [*]   LCD simul output                                     | |
    | |          Choice display output support (VGA output)  --->     | |
    | |          Choice display resolution (800x600)  --->            | |
    | |        Choice display bpp (8)  --->                           | |
```

「ディスプレイ=>LCD&NTSC、解像度=>640x480、色深度=>8」

```
    | |    [*] MS104-VGA                                              | |
    | |    [*]   LCD simul output                                     | |
    | |          Choice display output support (NTSC output)  --->    | |
    | |          Choice display resolution (640x480)  --->            | |
    | |        Choice display bpp (8)  --->                           | |
```

「ディスプレイ=>LCD&S-Video、解像度=>640x480、色深度=>8」

```
    | |    [*] MS104-VGA                                              | |
    | |    [*]   LCD simul output                                     | |
    | |          Choice display output support (S-Video output)  --->  | |
    | |          Choice display resolution (640x480)  --->            | |
    | |        Choice display bpp (8)  --->                           | |
```

５．３　Linux カーネルのコンパイル

① Linux カーネルのソースディレクトリに移動します。

```
[guest@LinuxKit ~]$ cd linuxkit-ms104sh4/linux-2.6.28.8-alp/
[guest@LinuxKit linux-2.6.28.8-alp]$
```

② 『5.2Linux カーネル設定例』を参照し設定します。

③ make します。

```
[guest@LinuxKit linux-2.6.28.8-alp]$ ARCH=sh CROSS_COMPILE=sh4-linux- make uImage
  HOSTLD  scripts/kconfig/conf
... 中略 ...
  Image arch/sh/boot/uImage is ready
[guest@LinuxKit linux-2.6.28.8-alp]$
```

④ 生成されたバイナリファイルを確認します。

```
[guest@LinuxKit linux-2.6.28.8-alp]$ ls arch/sh/boot/uImage
arch/sh/boot/uImage
[guest@LinuxKit linux-2.6.28.8-alp]$
```

６．タッチパネルのキャリブレーション

タッチパネル付 LCD モニタを使用する場合、タッチパネルで得られる値と LCD モニタの位置を関連付けなければなりません。タッチパネルで得られる値はアナログ値で、経年変化による劣化や使用環境による値の増減を考慮する必要があります。そのため、タッチパネル付 LCD モニタを使用する際はタッチパネルの位置合わせ（キャリブレーション）をする必要があります。
MS104-VGA/LCD を弊社 LCD キット「LCD-KIT-A03」と接続した際に Linux 上からキャリブレーションを行うアプリケーションプログラム『ts_calibrate』がルートファイルシステムに含まれています。
以下ではタッチパネルキャリブレーションとキャリブレーションの実行について説明します。

タッチパネルは 12 ビットの分解能、つまり、0～4095 までの値でタッチパネルのどの座標を押したかを示します。
解像度 800×600 の LCD モニタを使用した場合、LCD モニタの（800、600）という座標はタッチパネルでは（4095、4095）という座標になります。実際には LCD モニタとタッチパネルの Y 座標の値は最大値と最小値が逆になるため、（800、600）は（4095、0）という値で表されます。

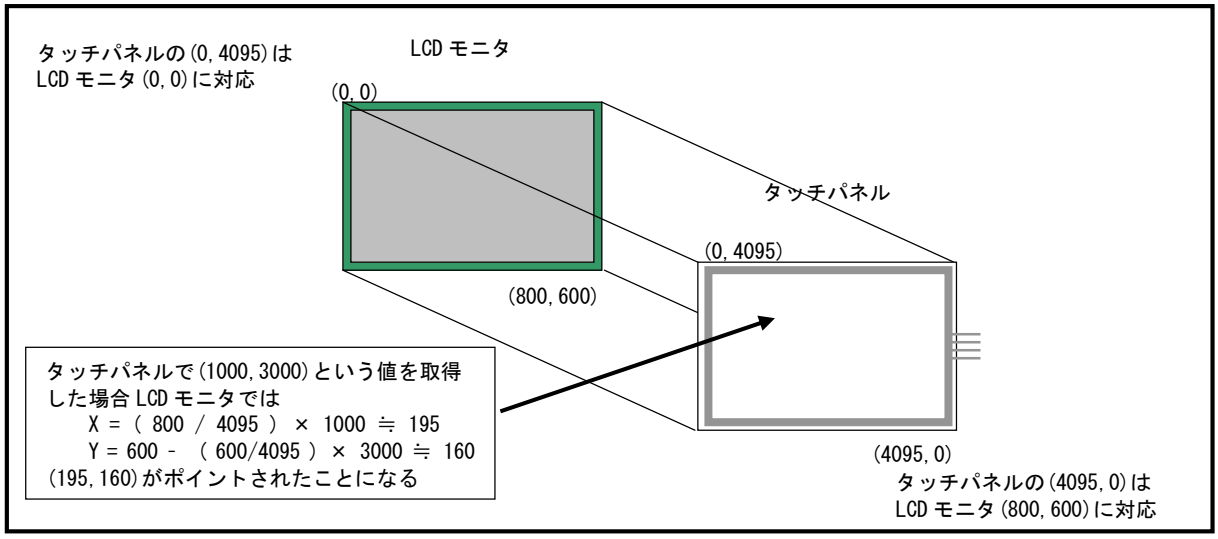


Fig 6.1-1　タッチパネルと LCD モニタの座標

しかし、上記は理想的な場合で実際には LCD モニタの（0、0）や（800、600）という座標がタッチパネルの（0、4095）や（4095、0）といった値にはなりません。タッチパネルの 0～4095 という値はあくまで分解能であり、LCD モニタの 0～800 という座標がそのまま 0～4095 に対応しません。したがって、LCD モニタの（0、0）や（800、600）という座標に対して、タッチパネルはどのような値をとるか測定し、その測定値を元に LCD モニタとタッチパネルの座標の変換を行う必要があります。

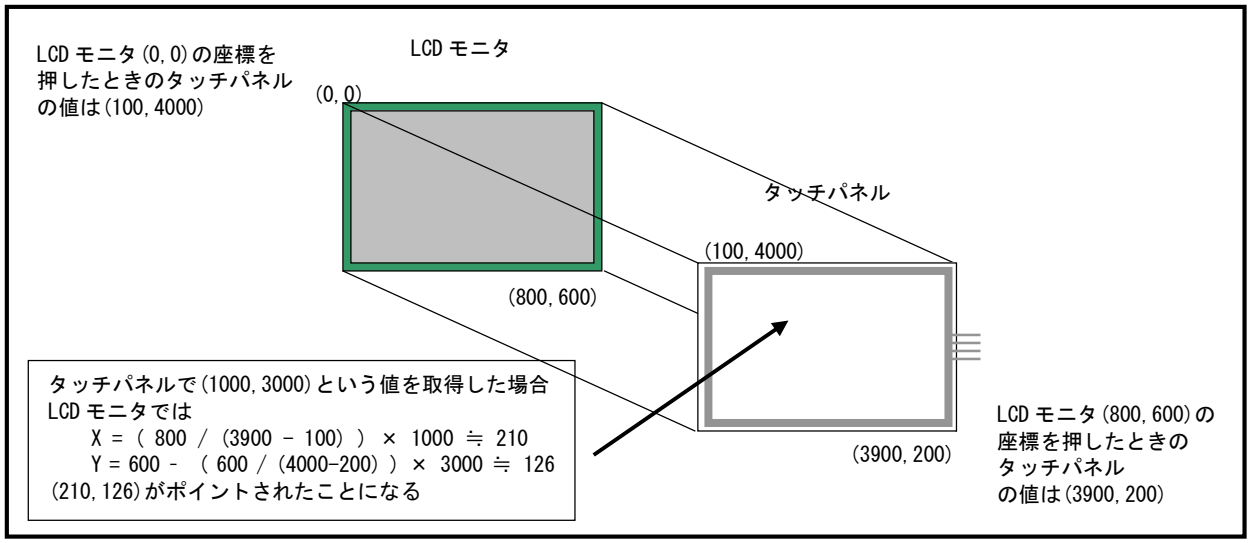


Fig 6.1-2　タッチパネルのキャリブレーション

上図のタッチパネルと LCD モニタの位置変換は tslib と『/etc/ts.conf』ファイルで行っています。

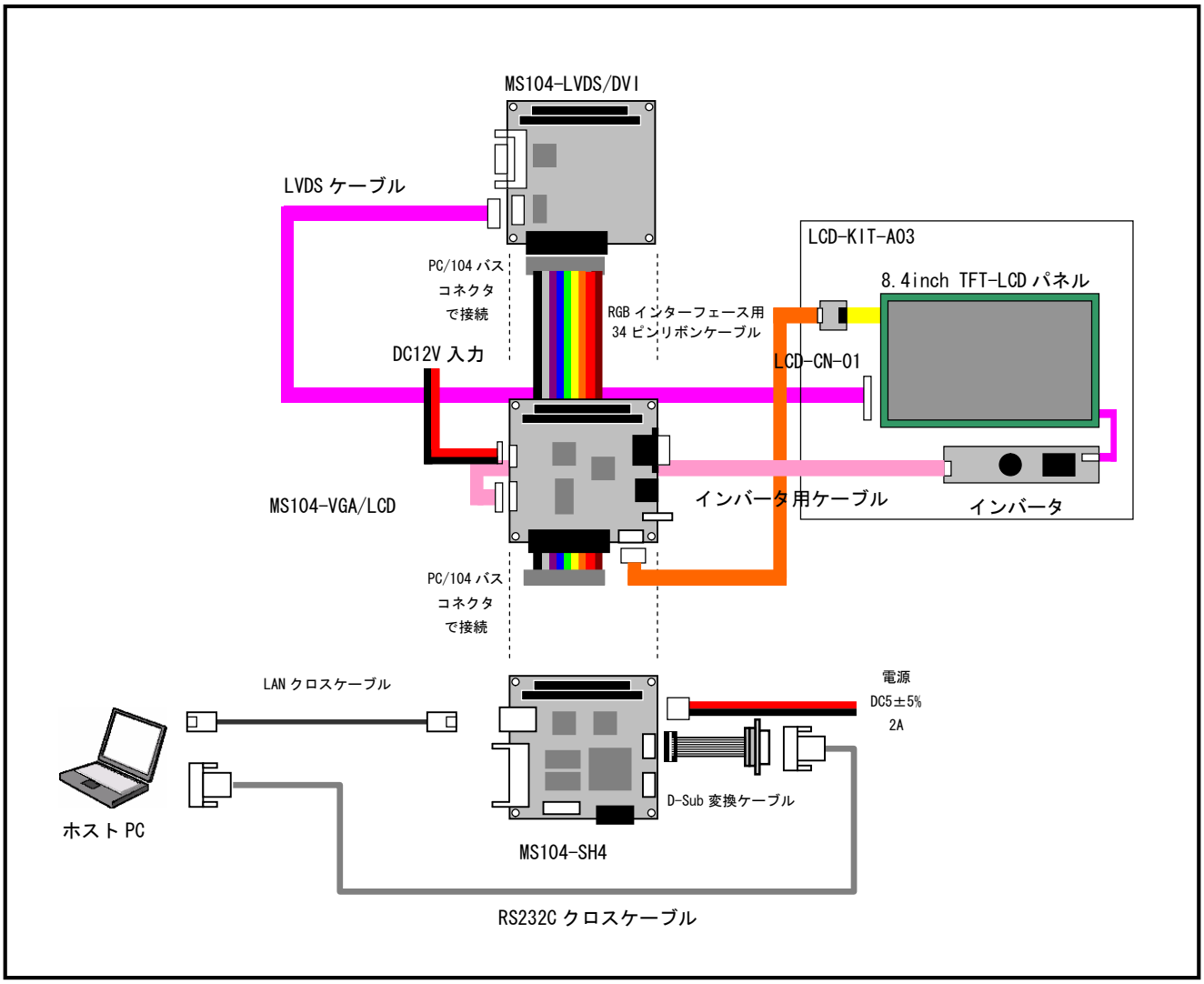


Fig 6.1-3　LCD-KIT-A03 との接続

① MS104-VGA/LCD と MS104-SH4、LCD-KIT-A03 を接続します。「Fig 6.1-3 LCD-KIT-A03 との接続」を参考に接続をして下さい。

② 「3.4　MS104-VGA/LCD の動作」を参考に Linux を起動し、ルート権限でログインします。

```
uclibc login: root
#
```

③ キャリブレーションプログラム『ts_calibrate』を起動します。

```
# ts_calibrate
xres = 800, yres = 600
```

④ LCD モニタ上に表示される左上のカーソルをタッチして下さい。カーソルをタッチすると、右上、右下、左下、中央とカーソルが移動していきます。中央のカーソルをタッチするとキャリブレーションは終了です。

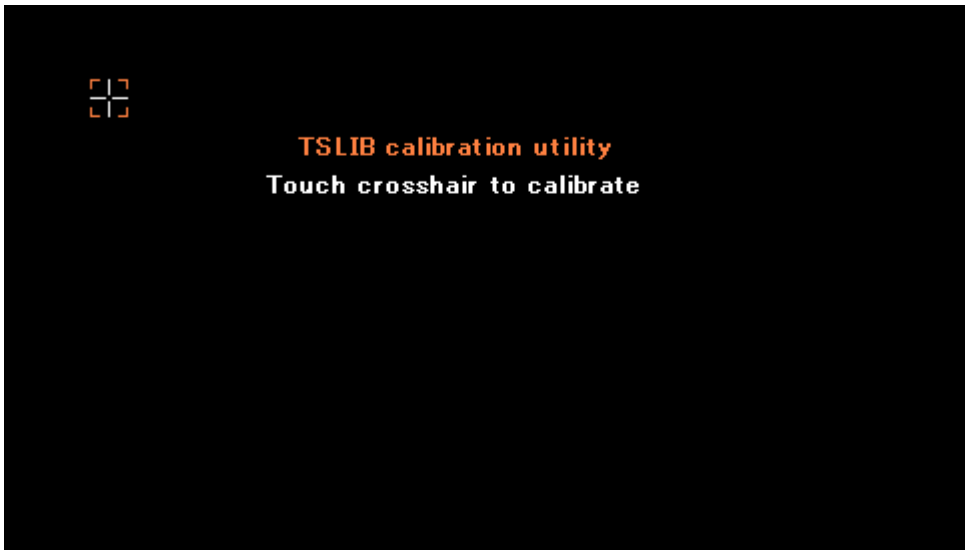


Fig 6.1-4　キャリブレーション画面

７．ブザーの再生

Linux でブザーを再生する際は、/dev/ms104vga_buzzer デバイスファイルにアクセスします。

７．１　シェル上からのアクセス

① MS104-VGA/LCD と MS104-SH4、LCD-KIT-A03 を接続します。「Fig 6.1-3 LCD-KIT-A03 との接続」を参考に接続をして下さい。

② 「3.3　MS104-VGA/LCD の動作」を参考に Linux を起動し、ルート権限でログインします。

```
uclibc login: root
#
```

③ デバイスファイルに 0 以外の値を書き込むとブザーが再生されます。

```
# echo -e "¥x1" > /dev/ms104vga_buzzer
#
```

④ デバイスファイルに 1 を書き込むとブザーの再生が停止します。

```
# echo -e "¥x0" > /dev/ms104vga_buzzer
#
```

７．２　Ｃ言語でのアクセス

Ｃ言語でアクセスする場合のサンプルは以下のようになります。アプリケーションのコンパイル及び実行につきましては、「MS104-SH4 ソフトウェアマニュアル」をご参照下さい。

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>

int main(int argc, char **argv)
{
  char *device = "/dev/ms104vga_buzzer";
  int fd;
  int on = 1, off = 0;

  /* デバイスファイルをオープン */
  fd = open(device, O_WRONLY);

  /* 1 を書き込みブザーを再生 */
  write(fd, &on, 1);

  /* 1 秒間待つ */
  sleep(1);

  /* 0 を書き込みブザーを停止 */
  write(fd, &off, 1);

  /* デバイスファイルをクローズ */
  close(fd);
}
```

８．サンプルプログラム

８．１　サンプルプログラムの動作

DirectFB のサンプルプログラムをコンパイルし、実行します。

●directfb-sample.c

```
#include <stdio.h>
#include <unistd.h>

#include <directfb.h>

static IDirectFB *dfb = NULL;

static IDirectFBSurface *primary = NULL;

static IDirectFBFont *font = NULL;

static int screen_width  = 0;
static int screen_height = 0;

#define DFBCHECK(x...)                                         ¥
  {                                                            ¥
    DFBResult err = x;                                         ¥
                                                               ¥
    if(err != DFB_OK)                                          ¥
      {                                                        ¥
        fprintf( stderr, "%s <%d>:¥n¥t", __FILE__, __LINE__ ); ¥
        DirectFBErrorFatal( #x, err );                         ¥
      }                                                        ¥
  }

int main(int argc, char **argv)
{
  DFBSurfaceDescription dsc;
  DFBFontDescription font_dsc;

  /* DirectFB の初期化 */
  DFBCHECK(DirectFBInit(&argc, &argv));

  /* メインインターフェースの取得 */
  DFBCHECK(DirectFBCreate(&dfb));

  /* フルスクリーン指定 */
  DFBCHECK(dfb->SetCooperativeLevel(dfb, DFSCL_FULLSCREEN));

  /* 画面設定 */
  dsc.flags = DSDESC_CAPS;
  dsc.caps  = DSCAPS_PRIMARY | DSCAPS_FLIPPING;

  /* 描画画面の取得 */
```

```
DFBCHECK(dfb->CreateSurface( dfb, &dsc, &primary ));

/* 画面サイズの取得 */
DFBCHECK(primary->GetSize(primary, &screen_width, &screen_height));

/* バックグランド色での塗潰し */
DFBCHECK(primary->FillRectangle(primary, 0, 0, screen_width, screen_height));

/* 描画色の指定：赤 */
DFBCHECK(primary->SetColor(primary, 255, 0, 0, 255));

/* 線描画 */
DFBCHECK(primary->DrawLine(primary, 0, 136, 799, 136));

/* 四角描画 */
DFBCHECK(primary->DrawRectangle(primary, 100, 100, 250, 250));

/* 描画色の指定：緑 */
DFBCHECK(primary->SetColor(primary, 0, 255, 0, 255));

/* 四角描画（塗潰し） */
DFBCHECK(primary->FillRectangle(primary, 200, 200, 500, 300));

/* 描画色の指定：青 */
DFBCHECK(primary->SetColor(primary, 0, 0, 255, 255));

/* 三角描画（塗潰し） */
DFBCHECK(primary->FillTriangle(primary, 100, 10, 250, 300, 550, 300));

/* フォントサイズの指定 */
font_dsc.flags = DFDESC_HEIGHT;
font_dsc.height = 48;

/* フォントデータの作成 */
DFBCHECK (dfb->CreateFont (dfb, "/usr/share/directfb-examples/fonts/decker.ttf", &font_dsc, &font));

/* フォント設定 */
DFBCHECK (primary->SetFont (primary, font));

/* フォント色の指定：赤 */
DFBCHECK (primary->SetColor (primary, 255, 0, 0, 0xFF));

/* 文字描画 */
DFBCHECK (primary->DrawString (primary, "DirectFB Sample", -1, 240, 136, DSTF_BOTTOMCENTER));

/* 画面表示の実行 */
DFBCHECK(primary->Flip(primary, NULL, 0));

getchar();

/* フォントの破棄 */
DFBCHECK(font->Release (font));
```

```
    /* 描画画面の破棄 */
    DFBCHECK(primary->Release( primary ));

    /* メインインターフェースの破棄 */
    DFBCHECK(dfb->Release( dfb ));

    return 0;
}
```

① ホスト PC 上のサンプルプログラムのソースディレクトリに移動します。

```
[guest@LinuxKit ~]$ cd linuxkit-ms104sh4/samples/ms104vga-sample/
[guest@LinuxKit ms104vga-sample]$
```

② サンプルプログラムをコンパイルします。

```
[guest@LinuxKit ms104vga-sample]$ sh4-linux-gcc `PKG_CONFIG_SYSROOT_DIR=/home/guest/linuxk
it-ms104sh4/staging_dir pkg-config --cflags --libs directfb` -o directfb-sample directfb-s
ample.c
[guest@LinuxKit ms104vga-sample]$
```

③ ホスト PC 上で作成したサンプルプログラムを NFS 共有ディレクトリ『/nfs』にコピーします。

```
[guest@LinuxKit ms104vga-sample]$ cp -a directfb-sample /nfs/
[guest@LinuxKit ms104vga-sample]$
```

④ MS104-VGA/LCD と MS104-SH4、LCD-KIT-A03 を接続します。「Fig 6.1-3 LCD-KIT-A03 との接続」を参考に接続をして下さい。

⑤ 「3.4　MS104-VGA/LCD の動作」を参考に Linux を起動し、ルート権限でログインします。

```
uclibc login: root
#
```

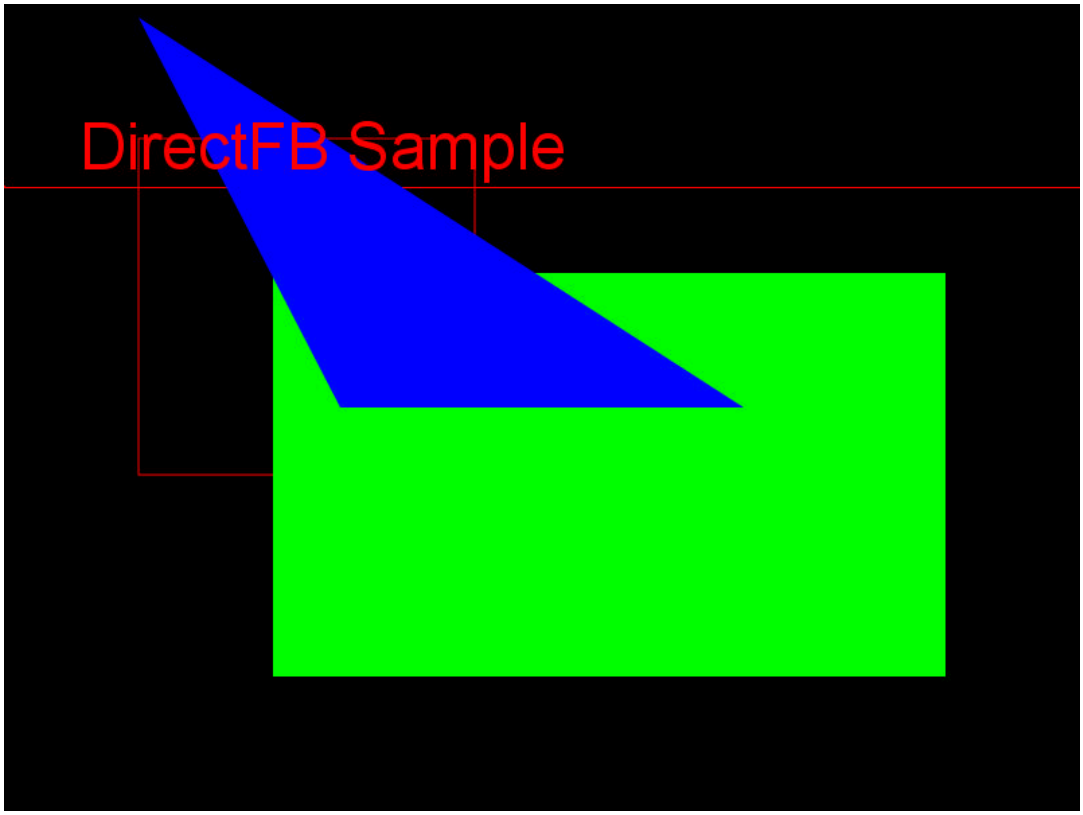
⑥ ホスト PC の NFS ディレクトリをマウントします。

```
# mount -t nfs -o nolock,rsize=2048 192.168.4.112:/nfs /mnt/nfs/
#
```

⑦　サンプルプログラムを起動します。

```
# /mnt/nfs/directfb-sample
```

LCD モニタ上に以下の画面が表示されます。



９．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下さい。
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下さい。

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。

謝辞

Linux、SH-Linux、U-Boot、DirectFB 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

商標について

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 : query@apnet.co.jp