組み込み Linux ガイド

シリコンリナックス株式会社

目次

1 はじめに

本書は組み込み Linux ボード CAT724 シリーズ向けの 組み込み Linux ガイドです。本書をお客様のアプリケーション作成にお役立てください。

1.1 開発キット CDROM 構成

開発キット CDROM は以下の構成です。

```
|-- VirtualBox                開発用 PC にインストールする仮想 PC ソフト
|-- bootloader                CAT724 ブートローダ (/dev/mtdblock0)
|-- cross-tools               開発用 PC にインストールするクロスコンパイラ
|   |-- toolchain-amd64             64bit OS 用
|   `-- toolchain-i386            32bit OS 用
|-- debian-sh                 SH4 向け debian システム
|   `-- squeeze-sh4
|       |-- base
|       `-- packages
|-- kernel                    CAT724 カーネル、ソースコード及びバイナリ
|-- rootfs                    CAT724 内蔵 FLASH ルートファイルシステム
`-- sample_driver                本ガイドで解説しているサンプルドライバ
      |-- countdrv
      |-- hellodrv
      |-- kmalloc
      |-- leddipswdriver1
      |-- leddipswdriver2
      |-- lock-sample
      |-- poll_test
      |-- software_timer
      |-- software_timer_tasklet
      |-- software_timer_workqueue
      |-- timer_warikomi
      |-- waittest
      `-- waittest2
```

1.2 開発環境について

本機のプログラムを開発するためにパソコンを1台ご用意ください。表に示す環境で動作確認しています。本書では debian 6.0 (通称 squeeze)システムを奨励しています。PC に debian 6.0 squeeze をインストールするか、もしくは VirtualBox などの仮想 PC 環境を活用して debian 6.0 squeeze をインストールしてください。

奨励環境	Windows 7 32bit/64bit	+ VirtualBox debian 6.0 squeeze
動作可能環境	Windows VISTA, XP, debian 6.0 squeeze	

1.2.1 本キット付属の VirtualBox

VirtualBox は GNU GPL Version2 ライセンスで配布されるフリーな仮想PCソフトウェアです。本キットには VirtualBox 実行環境と、クロスコンパイラをインストール済みの debian 6.0 squeeze ディスクイメージが添付されています。本キットを使用することにより SH4 クロスコンパイラを素早く導入する事が出来ます。

詳しくは別紙 VirtualBox のインストールをご覧ください。

1.2.2 Debian 6.0 squeeze を使用する場合

開発に必要なユーティリティをインストールします

```
# apt-get update
# apt-get install build-essential nfs-kernel-server samba dpkg-dev dpkg-cross
# apt-get install libncurses5-dev
```

32bit または 64bit の debian 6.0 squeeze が使用できます。PC に debian 6.0 squeeze をインストールした場合は以下の操作によりクロス環境の導入ができます。

CDROM 内の cross-tools/ に保存されています。

32bit PC 版 ディレクトリ名 toolchain-i386

64bit PC 版 ディレクトリ名 toolchain-amd64

上記ディレクトリを PC 内にコピーします。

例) 32bitPC で /home/kaihatsu にコピーしたとします。

root ユーザになります

```
$ su -
Passwd:
```

/etc/apt/sources.list に 1 行追加します。最後の ./ も必要です。

```
deb file:/home/kaihatsu/toolchain-i386 ./
```

クロスツールのインストール

```
# apt-get update
# apt-get install gcc-4.4-sh4-linux-gnu
# apt-get install g++-4.4-sh4-linux-gnu
```

2　開発PCの環境を整える

2.1　USB シリアルドライバ

CAT724 のコンソールは USB シリアルとなっています。

EB724A ベース基板 CN10　　USBminiB 端子

FDSI 社 シリアル変換チップ FT232RL

Windows7 であればドライバ不要です（自動的にインストールされます）

WindowsVISTA, XP をご利用の方は FTDI 社のサイトから VCP ドライバを入手しインストールしてください。

```
http://www.ftdichip.com/
  → Drivers を選択する
    → VCP Drivers を選択する
      → Windows を選択する
```

2.2　COMポート通信ソフト

CAT724 はシリアルポートをコンソールとして利用します。開発PCには何かしらのCOMポート通信ソフトが必要です。本書では TeraTerm を推奨します。

```
http://sourceforge.jp/projects/ttssh2/
```

CAT724 コンソールボーレート

115200bps, 8bit, 1stopbit, ノンパリティ, 文字コード UTF-8

2.3　NFS サーバのセットアップ

開発環境PCで以下のコマンドをタイプしてください。

NFS サーバのインストール（本キット付属 CDROM の debian では既にインストールされています）

```
開発 PC の root ユーザコマンドラインでタイプします

# apt-get install nfs-kernel-server
```

エクスポート（共有）ディレクトリの設定

```
開発 PC の root ユーザコマンドラインでタイプします

# vi /etc/exports
```

以下ファイルの中身

```
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
/home    192.168.1.0/255.255.255.0(ro,no_root_squash,no_subtree_check)
```

上の例では /home 位置を、192.168.1.0/255.255.255.0 LAN に接続された全てのホストに対し、「ReadOnly」、

「root 権限でのマウントを許可」の条件で共有を許しています。

255.255.255.0 と“(”（括弧）の間にはスペースを入れてはいけません。

2.3.1 NFS サーバの再起動

開発 PC の root ユーザコマンドラインでタイプします

```
# /etc/init.d/nfs-kernel-server restart
```

2.3.2 NFSクライアント側の操作方法

CAT724 はNFSクライアント側になります。CAT724 側からは mount コマンドにて サーバの共有ディレクトリのマウントを行います。

CAT724 の root ユーザコマンドラインでタイプします

```
# mount   192.168.1.2:/home   /mnt   -o   ro,tcp
```

書式は

mount サーバの IP アドレスもしくはホスト名:/サーバのディレクトリ　マウント先 -o オプション
オプションの例 -o ro リードオンリ、 tcp TCP 接続を利用する

になります。

3 CAT724の操作方法

3.1 初めての立ち上げ

1. CAT724 の miniUSB ポートとPCを接続します
2. TeraTerm でシリアルポート COM x を開きます。
3. ボーレートを 115200bps とします
4. CAT724 の電源を投入します。

CAT724 の電源を投入するとブートローダが立ち上がります。

```
CATBOOT for CAT724 Ver 1.94 build Sep  9 2011 18:23:37
>>
```

```
>>help
              admin: administrator mode
               boot: boot linux kernel
             dipsw: display dipsw
                led: led on/off
            eeprom: dump eeprom
               help: show command list
            setmac: set mac address
          setparam: set kernel paramater
  setkernelsize: set kernel size
              timer: timer interrupt on/off
              reset: hardware reset
              debug: show debug info
            sddump: SD raw dump
                dir: print directry
                cat: print file
                md5: print md5sum
       flashwrite: flash write
       flasherase: flash erase
```

3.2 ブートローダの主要コマンド

```
boot      Linux 起動
dir       マイクロ SD メモリのファイル表示
                  マイクロ SD メモリは、第一パーティションを FAT フォーマットとしてください
flasherase        開始番地(HEX)    長さ(HEX)        FlashRom の消去
```

flashwrite	ファイル名	開始番地(HEX)　FlashRom の書き込み
setparam		
	-d	デフォルト値
	-s	SD メモリの第２パーティションを rootfs にする設定
	-u	USB メモリの第２パーティションを rootfs にする設定

3.3 Linux の起動とログイン

ブートローダで boot とタイプすると Linux が起動します。login: プロンプトに対して root ユーザ、パスワード初期値 root でログインできます。

初期値			
	管理ユーザ名	root	パスワード　root
	一般ユーザ名	kaihatsu	パスワード　kaihatsu

3.4 rommode

本装置は電源の即断に対応するため root ファイルシステムはリード・オンリーになっています。ファイルを編集する際は rommode コマンドでリード・ライトに変更してください。

```
CAT724 の root ユーザで操作
# rommode rw        リード・ライトモード　ファイルの編集可能
# rommode ro        リード・オンリーモード　ファイルの編集不可能（電源即断可能）
```

3.5 ネットワークの設定

/etc/network/interfaces に記載します。vi エディタで編集してください。

```
CAT724 の root ユーザで操作
# rommode rw　(リード・ライトモードにする)
# vi  /etc/network/interfaces
```

dhcp による自動割り当てとする場合

```
# The primary network interface
auto eth0
iface eth0 inet dhcp
#iface eth0 inet static
#     address 172.16.0.52
#     netmask 255.255.0.0
#     gateway 172.16.0.1
```

固定 IP 割り当てとする場合（例）

```
# The primary network interface
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
        address 192.168.1.3
        netmask 255.255.255.0
        gateway 192.168.1.1
```

```
# rommode ro  (リード・オンリモードにする)
```

反映させるために再起動してください。

3.6 時計のセット

```
# date -s "2011/9/12 18:14:00"
```

″年/月/日 時:分:秒″ の書式で入力します。OS の時間(ソフトウェア時計)がセットされます。ソフト時計はOS内部の変数です。電源を切ると消えてしまいます。

```
# hwclock --systohc
```

date コマンドでセットしたソフトウェア時計を RTC(ハードウェア時計)に転送します。

3.7 マイクロ SD メモリーカードのパーティション

マイクロ SD メモリーカード(以下SDカードと略記)は第一パーティションをFATフォーマットとすることを推奨します。

- Windows 系OSは、リムーバブルメディアは第一パーティションしか認識しない
- Windows 系OSとのデータの交換はFATで行う
- CAT724 のブートローダも第一パーティションがFATであることを前提としている

といった理由があります。

4G バイトの SD メモリーカードをのパーティションを作成する例

```
CAT724 の root ユーザで操作
# cfdisk /dev/mmcblk0            (USB メモリを使用するときは # cfdisk /dev/sda とします)
```

```
                        cfdisk (util-linux-ng 2.17.2)

                              Disk Drive: /dev/mmcblk0
                        Size: 3965190144 bytes, 3965 MB
                Heads: 122   Sectors per Track: 62   Cylinders: 1023

    Name        Flags      Part Type  FS Type          [Label]        Size (MB)
------------------------------------------------------------------------------
```

```
mmcblk0p1          Primary   FAT16        [        ]     127.81
mmcblk0p2          Primary   Linux ext3                   3500.99
mmcblk0p3          Primary   Linux swap / Solaris           333.06

 [ Bootable ] [  Delete  ] [   Help   ] [ Maximize ] [  Print   ]
 [   Quit   ] [   Type   ] [  Units   ] [  Write   ]

                    Quit program without writing partition table
```

cfdisk ユーティリティでは

- カーソルキーの左右で下段のメニューを選択
- カーソルキーの上下でパーティション位置を選択

となります。パーティション ID は

FAT16	0x06
Linux	0x83
LinuxSwap	0x82

を使用してください。

3.8 SDカードのフォーマット

```
CAT724 の root ユーザで操作
# mkdosfs  -F16  /dev/mmcblk0p1          第一パーティションを FAT でフォーマット
# mkfs.ext3  /dev/mmcblk0p2              第二パーティションを ext3 でフォーマット
# mkswap  /dev/mmcblk0p3                 第三パーティションを swap 領域としてフォーマット
```

3.9 USB メモリのフォーマット

```
CAT724 の root ユーザで操作
# mkdosfs  -F16  /dev/sda1               第一パーティションを FAT でフォーマット
# mkfs.ext3  /dev/sda2                   第二パーティションを ext3 でフォーマット
# mkswap  /dev/sda3                      第三パーティションを swap 領域としてフォーマット
```

4 クロスコンパイル

4.1 簡単なプログラムのクロスコンパイル

以下のような簡単なプログラムを書き、クロスコンパイルを行って実行させてみましょう。

開発 PC の一般ユーザコマンドラインでタイプします

```
$ vi hello.c
```

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

int main(){
        int i;
        char *p;

        printf("hello sh-linux world\n");

        // 1M バイトメモリ確保
        p=malloc(1024*1024);

        while(1){
                printf("main=%08x, printf=%08x, malloc=%08x, i=%08x\n",
                        main, printf, p, &i);
                sleep(1); // 1 秒スリープ
        }
}
```

普通にコンパイルを行います。

開発 PC の一般ユーザコマンドラインでタイプします

```
$ gcc hello.c
```

開発PCで実行

開発 PC の一般ユーザコマンドラインでタイプします

```
$ ./a.out
hello sh-linux world
main=004005d4, printf=00400498, malloc=7ce39010, i=7e749d14
main=004005d4, printf=00400498, malloc=7ce39010, i=7e749d14
main=004005d4, printf=00400498, malloc=7ce39010, i=7e749d14
main=004005d4, printf=00400498, malloc=7ce39010, i=7e749d14
```

```
:略　CTRL+C で停止します
```

gcc の出力ファイル名のデフォルトは a.out です。これは-o オプションで変更することが出来ます。また UNIX ではカレントディレクトリに実行パスが通っていませんので、カレントディレクトリを示す ./ を頭に付けて実行します。

クロスコンパイルを行う

CAT724 では

```
開発 PC の一般ユーザコマンドラインでタイプします
$ sh4-linux-gnu-gcc hello.c
```

とします。

結果を確かめる

```
開発 PC の一般ユーザコマンドラインでタイプします
$ file a.out
a.out: ELF 32-bit LSB executable, Renesas SH, version 1 (SYSV), dynamically linked (uses shared libs), for
GNU/Linux 2.6.26, not stripped
```

開発PCで実行してみる

```
開発 PC の一般ユーザコマンドラインでタイプします
$ ./a.out
-bash: ./a.out: cannot execute binary file
```

実行できません。

4.2　CAT上でプログラムを実行する

コンパイルしたプログラムを CAT 機で実行しましょう。CAT 上で以下のコマンドをタイプし、開発 PC のディレクトリを NFS マウントします。

```
CAT724 の root ユーザコマンドラインでタイプします
# mount  192.168.1.2:/home  /mnt  –o  ro,tcp
```

書式 mount IP アドレスもしくはホスト名:/共有ディレクトリ　/マウント先

-o オプション ro は ReadOnly，tcp は TCP 接続を利用する（デフォルトは UDP）

コンパイルした a.out の実行

```
CAT724 の root ユーザコマンドラインでタイプします
# cd /mnt/kaihatsu　　(a.out ファイルのある場所に移動)

CAT724 の root ユーザコマンドラインでタイプします
```

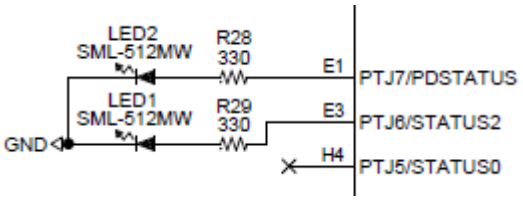
```
# ./a.out
hello sh-linux world
main=004005a0, printf=004003c0, malloc=296e2008, i=7bbaed8c
main=004005a0, printf=004003c0, malloc=296e2008, i=7bbaed8c
main=004005a0, printf=004003c0, malloc=296e2008, i=7bbaed8c
main=004005a0, printf=004003c0, malloc=296e2008, i=7bbaed8c
: 略 CTRL+C で停止
```

5 IO ポート(LED, DIPSW)

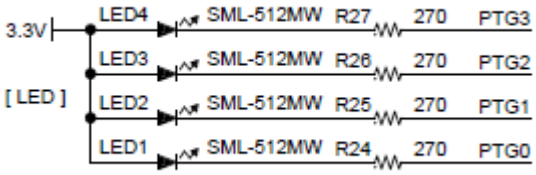
5.1 LED, DIPSWポートについて

Linuxとは離れますが、CAT724 のハードウェア的なことをまとめます。CAT724 の LED と DIPSW の回路を抜粋して記載します。基板上のシルク印刷は LED1, LED2 となっていますが、ソフトウェア上は 0 始まりとします。LED0 は CAT724 CPU ボード上の「LED1」とします。

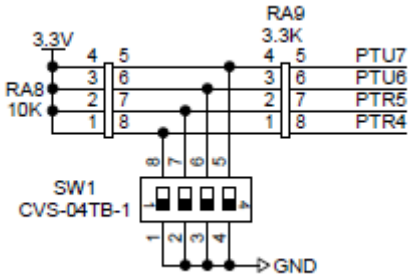
- LED0, 1



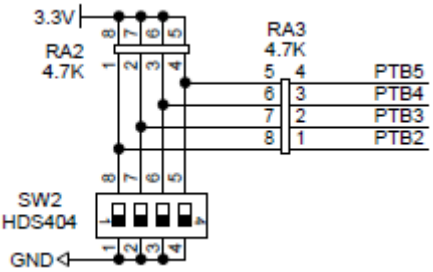
- LED2, 3, 4, 5



- DIPSW0, 1, 2, 3



- DIPSW4, 5, 6, 7



LED0 は 回路図上の PTJ6 に接続されています。PTJ6 は「ポート J の第6ビット」の意味です。ポート J 第6ビットを論理レベル1にすると電圧がHレベルとなり、LED に向かって電流が流れ LED が点灯します。論理レベル0にすると消灯します。

5.2 Linux 汎用 gpio ドライバ

CAT724 では Linux 標準の gpio ドライバが使用できます。

```
# cat /sys/kernel/debug/gpio
```

を実行すると各 GPIO の接続先が確認できます。

```
gpio-0   (GPIO_PTA7         ) in  hi
gpio-1   (GPIO_PTA6         ) in  hi
gpio-2   (GPIO_PTA5         ) in  hi
gpio-3   (GPIO_PTA4         ) in  hi
gpio-4   (GPIO_PTA3         ) in  hi
gpio-5   (GPIO_PTA2         ) in  hi
gpio-6   (GPIO_PTA1         ) in  hi
gpio-7   (GPIO_PTA0         ) in  hi
gpio-8   (GPIO_PTB7         ) in  hi
gpio-9   (GPIO_PTB6         ) in  hi
gpio-10  (GPIO_PTB5         ) in  hi          (SW2-4 /dev/dipsw7)
gpio-11  (GPIO_PTB4         ) in  hi          (SW2-3 /dev/dipsw6)
gpio-12  (GPIO_PTB3         ) in  hi          (SW2-2 /dev/dipsw5)
gpio-13  (GPIO_PTB2         ) in  hi          (SW2-1 /dev/dipsw4)
gpio-14  (GPIO_PTB1         ) in  hi
gpio-15  (GPIO_PTB0         ) in  hi
gpio-16  (GPIO_PTC7         ) in  lo
gpio-17  (GPIO_PTC6         ) in  lo
gpio-18  (GPIO_PTC5         ) in  lo
gpio-19  (GPIO_PTC4         ) in  lo
gpio-20  (GPIO_PTC3         ) in  lo
gpio-21  (GPIO_PTC2         ) in  lo
gpio-22  (GPIO_PTC1         ) in  lo
gpio-23  (GPIO_PTC0         ) in  lo
gpio-24  (GPIO_PTD7         ) in  lo
gpio-25  (GPIO_PTD6         ) in  lo
gpio-26  (GPIO_PTD5         ) in  lo
gpio-27  (GPIO_PTD4         ) in  lo
gpio-28  (GPIO_PTD3         ) in  lo
gpio-29  (GPIO_PTD2         ) in  lo
gpio-30  (GPIO_PTD1         ) in  lo
gpio-31  (GPIO_PTD0         ) in  lo
gpio-32  (GPIO_PTE7         ) in  hi
```

```
gpio-33  (GPIO_PTE6       ) in  lo
gpio-34  (GPIO_PTE5       ) in  lo
gpio-35  (GPIO_PTE4       ) in  lo
gpio-36  (GPIO_PTE3       ) in  lo
gpio-37  (GPIO_PTE2       ) in  lo
gpio-38  (GPIO_PTE1       ) in  lo
gpio-39  (GPIO_PTE0       ) in  lo
gpio-40  (GPIO_PTF7       ) in  lo
gpio-41  (GPIO_PTF6       ) in  lo
gpio-42  (GPIO_PTF5       ) in  lo
gpio-43  (GPIO_PTF4       ) in  lo
gpio-44  (GPIO_PTF3       ) in  lo
gpio-46  (GPIO_PTF1       ) in  lo
gpio-47  (GPIO_PTF0       ) in  lo
gpio-48  (GPIO_PTG5       ) in  lo
gpio-49  (GPIO_PTG4       ) in  lo
gpio-50  (GPIO_PTG3           ) out hi          (LED4  /dev/led5)
gpio-51  (GPIO_PTG2           ) out hi          (LED3  /dev/led4)
gpio-52  (GPIO_PTG1           ) out hi          (LED2  /dev/led3)
gpio-53  (GPIO_PTG0           ) out hi          (LED1  /dev/led2)
gpio-54  (GPIO_PTH7       ) in  hi
gpio-55  (GPIO_PTH6       ) in  hi
gpio-56  (GPIO_PTH5       ) in  hi
gpio-57  (GPIO_PTH4       ) in  hi
gpio-58  (GPIO_PTH3       ) in  hi
gpio-59  (GPIO_PTH2       ) in  hi
gpio-60  (GPIO_PTH1       ) in  hi
gpio-61  (GPIO_PTH0       ) in  hi
gpio-62  (GPIO_PTJ7           ) out lo          (LED2  /dev/led1)
gpio-63  (GPIO_PTJ6           ) out hi          (LED1  /dev/led0)
gpio-69  (GPIO_PTK7       ) in  hi
gpio-70  (GPIO_PTK6       ) in  hi
gpio-71  (GPIO_PTK5       ) in  hi
gpio-72  (GPIO_PTK4       ) in  hi
gpio-73  (GPIO_PTK3       ) in  hi
gpio-74  (GPIO_PTK2       ) in  hi
gpio-75  (GPIO_PTK1       ) in  hi
gpio-76  (GPIO_PTK0       ) in  hi
gpio-77  (GPIO_PTL7       ) in  lo
gpio-78  (GPIO_PTL6       ) in  lo
gpio-79  (GPIO_PTL5       ) in  lo
gpio-80  (GPIO_PTL4       ) in  lo
gpio-81  (GPIO_PTL3       ) in  lo
```

```
gpio-82  (GPIO_PTL2        ) in  lo
gpio-83  (GPIO_PTL1        ) in  hi
gpio-84  (GPIO_PTL0        ) in  hi
gpio-85  (GPIO_PTM7        ) in  hi
gpio-86  (GPIO_PTM6        ) in  hi
gpio-87  (GPIO_PTM5        ) in  hi
gpio-88  (GPIO_PTM4        ) in  hi
gpio-89  (GPIO_PTM3        ) in  hi
gpio-90  (GPIO_PTM2        ) in  hi
gpio-91  (GPIO_PTM1        ) in  lo
gpio-92  (GPIO_PTM0        ) out hi
gpio-93  (GPIO_PTN7        ) in  hi
gpio-94  (GPIO_PTN6        ) in  hi
gpio-95  (GPIO_PTN5        ) in  lo
gpio-96  (GPIO_PTN4        ) in  hi
gpio-97  (GPIO_PTN3        ) in  hi
gpio-98  (GPIO_PTN2        ) in  hi
gpio-99  (GPIO_PTN1        ) in  lo
gpio-100 (GPIO_PTN0        ) in  lo
gpio-111 (GPIO_PTR5        ) in  hi      (SW1-2 /dev/dipsw1)
gpio-112 (GPIO_PTR4        ) in  lo      (SW1-1 /dev/dipsw0)
gpio-117 (GPIO_PTS6        ) in  hi
gpio-118 (GPIO_PTS5        ) in  hi
gpio-119 (GPIO_PTS4        ) in  hi
gpio-120 (GPIO_PTS3        ) in  hi
gpio-121 (GPIO_PTS2        ) in  hi
gpio-122 (GPIO_PTS1        ) in  hi
gpio-123 (GPIO_PTS0        ) in  hi
gpio-132 (GPIO_PTU7        ) in  hi      (SW1-4 /dev/dipsw3)
gpio-133 (GPIO_PTU6        ) in  hi      (SW1-3 /dev/dipsw2)
gpio-134 (GPIO_PTU5        ) in  lo
gpio-135 (GPIO_PTU4        ) in  hi
gpio-136 (GPIO_PTU3        ) in  hi
gpio-137 (GPIO_PTU2        ) in  lo
gpio-138 (GPIO_PTU1        ) in  lo
gpio-139 (GPIO_PTU0        ) in  hi
gpio-140 (GPIO_PTV7        ) in  hi
gpio-141 (GPIO_PTV6        ) in  hi
gpio-142 (GPIO_PTV5        ) in  hi
gpio-143 (GPIO_PTV4        ) in  hi
gpio-144 (GPIO_PTV3        ) in  hi
gpio-145 (GPIO_PTV2        ) in  hi
```

```
gpio-146 (GPIO_PTV1       ) in  hi
gpio-147 (GPIO_PTV0       ) in  hi
gpio-156 (GPIO_PTX7       ) out lo
gpio-157 (GPIO_PTX6       ) in  hi
gpio-158 (GPIO_PTX5       ) in  lo
gpio-159 (GPIO_PTX4       ) in  lo
gpio-160 (GPIO_PTX3       ) in  lo
gpio-161 (GPIO_PTX2       ) in  lo
gpio-162 (GPIO_PTX1       ) out lo
gpio-163 (GPIO_PTX0       ) in  hi
```

これにより LED0 が接続されている PTJ6 は gpio-63 であることがわかります。

5.2.1 LED0 の制御

ポートを出力にする

```
# echo out > /sys/class/gpio/gpio63/direction
```

ポートを High にする（LED0 が点灯する）

```
# echo 1 > /sys/class/gpio/gpio63/value
```

ポートを Low にする（LED0 が消灯する）

```
# echo 0 > /sys/class/gpio/gpio63/value
```

5.2.2 LED2 の制御

LED2 は PTG0 (gpio-53) です。LED2 から 5 は負論理です。

ポートを出力にする

```
# echo out > /sys/class/gpio/gpio53/direction
```

ポートを High にする（LED0 が消灯する）

```
# echo 1 > /sys/class/gpio/gpio53/value
```

ポートを Low にする（LED0 が点灯する）

```
# echo 0 > /sys/class/gpio/gpio53/value
```

5.2.3 DIPSW4 の読み込み

SW1 (dipsw0 から 3) は CAT724 CPU ボード上にある小さな DIPSW です。小さくて変更しにくいため、ベースボード EB724 上にある SW2 (dipsw4 から 7)で実験します。SW2-1 は PTB2 (gpio-13)です。

ポートを入力にする

```
# echo in > /sys/class/gpio/gpio13/direction
```

スイッチの論理地を読む

ポートを出力にする

```
# cat /sys/class/gpio/gpio13/value
1
```

5.3 led 制御プログラム

C 言語で記述すると次のようになります。LED を ON/OFF させるプログラムです。

CDROM の（sample_program/gpio_led/led.c）

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

#define LED0_DEVICE     "/sys/class/gpio/gpio63/value"

int led(int value){
        int fd;
        char c;

        fd = open(LED0_DEVICE, O_RDWR);
        if(fd<0){
                perror(LED0_DEVICE);
                exit(1);
        }

        if(value)
                c = '1';
        else
                c = '0';

        write(fd, &c, 1);

        close(fd);
}

int main(){
```

```
        while(1){
                printf("点灯\n");
                led(1);
                sleep(1);

                printf("消灯\n");
                led(0);
                sleep(1);
        }
}
```

コンパイル（開発PCにて）

```
開発 PC の一般ユーザコマンドラインでタイプします
$ sh4-linux-gnu-gcc led.c -O2 -g -o led
```

実行（CAT724 にて）

```
CAT724 の root ユーザコマンドラインでタイプします
# ./led
消灯
点灯
消灯
点灯
```

6　カーネル

カーネルはオペレーションシステムの中心部で、プロセスのスケジューリングやメモリ、IOの管理を行っています。またネットワークやファイルシステムもカーネルの仕事です。カーネルをコンフィグレーションすることで、デバイスドライバの追加削除や、読み書きできるファイルシステム形式の追加削除、ネットワーク機能の追加削除が出来ます。

6.1　カーネルの入手と展開

CDROM 内にあるカーネルを展開してください。

開発 PC の一般ユーザコマンドラインでタイプします

```
$ mkdir ~/cat-kernel
$ cd ~/cat-kernel
$ tar xzfv /CDROM をコピーしたディレクトリ/kernel/linux-3.0.4_cat724_日付.tgz
$ cd linux-3.0.4_cat724/
```

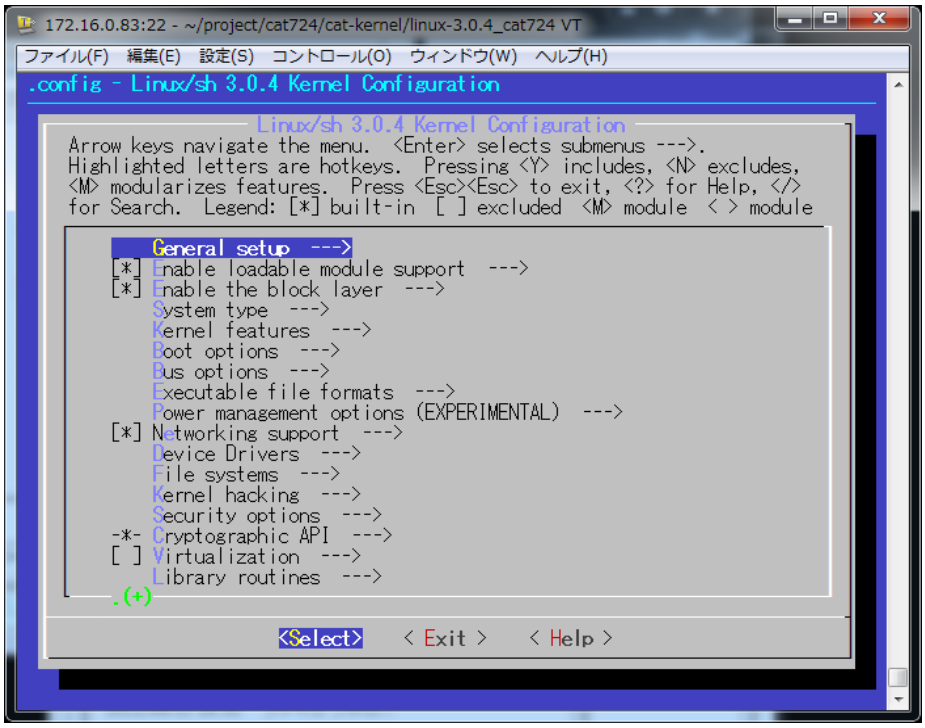
6.2　コンフィグレーション

以下のコマンドをタイプし、カーネルを展開してください。

開発 PC の一般ユーザコマンドラインでタイプします

```
$ make cat724_defconfig          CAT724 デフォルトのコンフィグレーション
$ make menuconfig
```

ここでは上下左右のカーソル、スペースキーで選択が出来ます。<*>印はカーネルに組み込む機能、<M>印は外部モジュールファイルとしてビルドする事を示します。



コンフィグレーションは特に指定のない限り変更しないでください。

6.3 config 変更の例（必要が無い限りは不要です）

CAT724 出荷時のカーネルは IPv6 サポートが無効になっています。これを有効にする例を示します。

開発 PC の一般ユーザコマンドラインでタイプし、config メニューを起動します。

```
$ make menuconfig

Networking support  --->            [enter を押す]
   Networking options  --->         [enter を押す]
      <*>   The IPv6 protocol  --->    [Y を押す]
```

カーソルキーの右を押して <Exit> を選択(enter を押す)
何度か繰り返してコンフィグレーションメニューを終了する

Do you wish to save your new configuration?
to continue.
の質問に <YES> を選択して保存終了する（ <NO> を選択すると保存せず終了する ）。

6.4 ビルド

以下のコマンドをタイプし、カーネルをビルドしてください。

開発 PC の一般ユーザコマンドラインでタイプします

```
$ make -j4
```

コンフィグレーションした機能および開発環境PCのスペックによりますが、1GHz の機械でおよそ 5～8 分程度でコンパイルが終了します。-j はジョブ数（並列コンパイル数）のオプションで、開発PCがマルチコア CPU の場合、CPU 数 x1.5 倍ほどにします。

コンパイル後にできあがった zImage ファイルがカーネルです。zImage ファイルは以下の場所にできあがります。

開発 PC の一般ユーザコマンドラインでタイプします

```
$ ls –l arch/sh/boot/zImage
```

ROMエリアのカーネル保存領域はデフォルトで 0x2E0000 (3,014,656) バイトです。zImage ファイルはこのサイズ以下に収まるよう、コンフィグレーションを行ってください。なお、ROM エリアのカーネル保存領域は bootloader のメニューにて変更することが出来ます。詳しくはブートローダーの章を参照してください。

6.5 コンパイルしたカーネルのインストール

6.5.1 Linux 上でのカーネル書き込み

Linux が動作中であればシェル上で zImage ファイルを /dev/mtdblock1 にコピーし、カーネルのアップデートが出来ます。

CAT724 の root ユーザコマンドラインでタイプします

```
# cp zImage /dev/mtdblock1
```

Linux 上で、CAT724 でのメモリーは以下のデバイスとしてアクセスできます

/dev/mtdblock0	FLASH ブートローダ
/dev/mtdblock1	FLASH カーネル(zImage)
/dev/mtdblock2	FLASH ルートファイルシステム(rootfs)
/dev/mtdblock3	SRAM (バッテリーバックアップ 512K)

6.5.2 ブートローダでのカーネル(zImage)の書き込み

本機のブートローダからカーネル(zImage)を FLASHROM に書き込むことができます。

・注意: SDカードの第一パーティションが FAT であること

本機にてSDカードにパーティションを作成したり FAT でフォーマットできます。詳しくは3章に記載があります。

SDカードの第一パーティションが FAT フォーマット済みであるとして、以下のように zImage ファイルを書き込みます。

```
# mount  /dev/mmcblk0p1  /media/sd/
# cp  /mnt/kaihatsu/cat-kernel/linux-3.0.4_cat724/arch/sh/boot/zImage  /media/sd/
# sync
# umount  /media/sd/
```

再起動しブートローダで次のように作業します

```
CATBOOT for CAT724 Ver 1.94 build Sep  9 2011 18:23:37
>>dir                （ファイル名の確認）
>>admin
password:*******        パスワードは  silinux です
ok.
#>flashwrite zImage kernel
```

6.6 カーネル起動パラメータ

カーネルには種々の起動パラメータがあります。起動パラメータを変更することでカーネルの動作を変更することが出来ます。ブートローダメニューの setparam コマンドで変更することが出来ます。カーネル起動パラメータは EEPROM に記録されます。最大文字数は 240 文字です。

主要なパラメータを紹介します。

```
console=ttySC0,115200    コンソール出力先の指定とボーレート（ほぼ必須）
root=/dev/mtdblock2      root としてマウントするデバイスの指定
                                 /dev/mtdblock2 などを指定してください。
                                 /dev/mtdblock2 は内蔵 FLASH です
ro                   起動時にルートをリードオンリでマウントします（必須）
rootfstype=jffs2         Flash メモリをマウントするときには
                                 ファイルシステム形式の指定が必要です（必須）
```

```
panic=10　　カーネルパニック発生時に（CPU が動いていれば）10 秒後にリブートします
```

例1　CAT 内蔵 FLASH ROM(/dev/mtdblock2)を root としてマウントする場合（標準）

```
CAT724 のブートローダでタイプします
>> admin
password:silinux
#> setparam console=ttySC0,115200 root=/dev/mtdblock2 ro rootfstype=jffs2
```

例3　エイリアスとして -d（デフォルト）、-s（SD メモリカード）、-u（USB ストレージ)を rootfs とする設定ができます。

```
#>setparam -d　　（内蔵 FLASH　工場出荷時デフォルト）
  save kernel command line = console=ttySC0,115200 root=/dev/mtdblock2 rootfstype=jffs2 ro

#>setparam -s　　（SD カードの第２パーティションを rootfs とする）
  save kernel command line = console=ttySC0,115200 root=b302　　rootdelay=3 rootwait ro

#>setparam -u　　（USB ストレージの第２パーティションを rootfs とする）
  save kernel command line = console=ttySC0,115200 root=/dev/sda2 rootdelay=5 rootwait ro
```

6.7　カーネルモジュールのインストールテクニック

カーネルコンフィグレーション時にモジュールとしてビルドしたファイルのインストールには少しこつが必要です。
カーネルのソースツリートップのMakefileの INSTALL_MOD_PATH を修正します。

```
#
# INSTALL_MOD_PATH specifies a prefix to MODLIB for module directory
# relocations required by build roots.  This is not defined in the
# makefile but the arguement can be passed to make if needed.
#
INSTALL_MOD_PATH=~/cat-module
```

モジュールは INSTALL_MOD_PATH で指定したディレクトリ下にインストールされます。

```
開発 PC の一般ユーザコマンドラインでタイプします
$ make modules_install
$ tree -d ~/cat-module
/home/ebihara/cat-module/
`-- lib
    `-- modules
        `-- 3.0.4
            |-- build -> /home/kaihatsu/
            |-- kernel
            |   |-- crypto
            |   |-- drivers
```

```
|  |  |-- base
|  |  |-- block
|  |  |-- ide
|  |  |-- net
|  |  `-- pcmcia
|  |-- fs
|  |  |-- fat
|  |  |-- lockd
|  |  |-- msdos
|  |  |-- nfs
|  |  |-- nls
|  |  `-- vfat
|  `-- net
|     |-- packet
|     `-- sunrpc
`-- source -> /home/kaihatsu/
```

従って、~/cat-module/ を CAT のルートに転送します。CATにて以下のコマンドをタイプしてください。

```
CAT724 の root ユーザコマンドラインでタイプします
# cp /NFS サーバ/ebihara/cat-module/* / -av
# depmod -a
```

注意： depmod -a 実行時は時計が合っている必要があります。電池切れなどで時計が過去の値（2000 年など）になっていると、ドライバモジュールが組み込まれなくなってしまいます。時計を合わせてから depmod -a を実行してください。