



Idea6410 ボード Linux_2.6.28 用 のマニュアル

株式会社日昇テクノロジー

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

info@csun.co.jp

2010/3/2

[copyright@2010](http://www.csun.co.jp)



・ 修正履歴

NO	バージョン	修正内容	修正日
1	Ver1.0	新規作成	2010/3/2

※ この文書の情報は、文書を改善するため、事前の通知なく変更されることがあります。最新版は弊社ホームページからご参照ください。

「<http://www.csun.co.jp>」

※ (株)日昇テクノロジーの書面による許可のない複製は、いかなる形態においても厳重に禁じられています。



目次

第一章	Linux2. 6. 28 バージョンの特徴	5
1.1	仕様	5
1.2	機能概要	5
第二章	開発環境要求	6
2.1	開発ボードの準備	6
2.2	ホスト環境	6
3.1	主なイメージファイル	7
3.1.1	SDboot	7
3.1.2	u-boot-nand	7
3.1.3	zImage	7
3.1.4	rootfs	7
3.2	開発ボード書き込み時の注意事項	7
3.2.1	注意事項一	7
3.2.2	注意事項二	9
3.2.3	注意事項三	10
3.3	起動用 SD カードの作成	10
3.4	u-boot の書き込む	12
3.4.1	起動モードの設定	12
3.4.2	SDboot の起動	13
3.4.3	Nand Flash のフォーマット	15
3.4.4	u-boot-nand.bin を Nand Flash に書き込む	16
3.4.5	u-boot-nand.bin の書き込みが成功したかの確認	18
3.5	zImage の書き込み	20
3.6	rootfs の書き込む	23
3.6.1	NFS ルートファイルシステムのマウント	23
3.6.2	YAFFS2 ファイルシステムの書き込み	25
第四章	イメージファイルのコンパイル	28
4.1	クロスコンパイラのインストール	28
4.2	u-boot ソースパケットのコンパイル	28
4.3	カーネルソースパケットのコンパイル	28
4.3.1	カーネルコンフィグファイルの設定	29
4.3.2	カーネルのコンパイル	31
第五章	ドライバとインタフェースのテスト	32
5.1	LCD テスト	32



5.2	タッチパネルのテスト及び訂正	33
5.3	SDカードのテスト	33
5.3.1	SDカードの読み出しテスト	34
5.3.2	SDカード書き込みテスト	34
5.4	USBテスト	35
5.4.1	USBマウステスト	35
5.4.2	USBキーボードテスト	35
5.4.3	USBメモリディスクテスト	36
5.5	LANテスト	36
第六章	マルチメディアハードウェアエンコード・デコード機能のテスト	38
6.1	H264ビデオハードウェアデコード機能テスト	38
6.2	MPEG-4ビデオハードウェアデコード機能テスト	39
6.3	H263ビデオハードウェアデコード機能テスト	40
6.4	VC-1ビデオハードウェアデコード機能テスト	41
6.5	複数チャンネル同時ハードウェアデコード機能テスト	42
6.6	JPEG画像のハードウェアデコード機能テスト	44
6.7	カメラプレビューとH264ビデオハードウェアエンコード機能テスト	45
6.8	カメラプレビューとJPEG画像ハードウェアエンコード機能テスト	47
6.9	カメラプレビューとMFCハードウェアデコード機能テスト	48



第一章 Linux2.6.28 バージョンの特徴

Idea6410 ARM11 ボードの Linux 最新バージョン Linux2.6.28 は以前のバージョン linux2.6.24 と比較して機能はもっと多くなり、豊かなデバイスドライバとサンプルソースが増加されて、S3C6410 開発ボードの優れ点を示している。

1.1 仕様

- 1、カーネルのバージョン : linux2.6.28
- 2、グラフィカルインターフェイス : Qtpia 2.2.0
- 3、ルートファイルシステム : yaffs2
- 3、ブートローダのバージョン : u-boot-1.1.6
- 4、クロスコンパイラ : cross-4.2.2-eabi

1.2 機能概要

- 1、MPEG4/H264/H263/VC-1 マルチメディアのハードウェアコーデック
- 2、ダブル、四分画面のビデオハードウェアのデコード
- 3、JPEG ハードウェアエンコードとデコード
- 4、カメラのキャプチャ、ビデオのエンコード、JPEG エンコーディング
- 5、2D/3D のハードウェア加速
- 6、2.8~10.4 の幅広いサイズの LCD の対応
- 7、TVOUT 出力
- 8、USB マウス、キーボード
- 9、タッチスクリーンのキャリブレーション
- 10、mediaplayer のビデオ再生
- 11、画像ブラウザ、ファイル管理、カレンダー、時計、電卓
- 12、手書き入力、キーボード入力、端末ターミナル
- 13、プラグインの管理
- 14、様々な内蔵ゲーム
- 15、Nand Flash、USB、SD/HSMCMC、オーディオ、イーサネット、UART、LCD、TOUCH、カメラ、SPI、IIC、DMA、KEYPAD、RTC、MFC、JPEG などのドライバ



第二章 開発環境要求

2.1 開発ボードの準備

- ◆idea6410 開発ボード (Ethernet IC は DM9000AEP)
- ◆クロスシリアルケーブル (メス・メス) : 開発ボード COM0 と PC の COM ポートを接続してデバッグするために使用
- ◆LAN ケーブル : ネットワークのテストに使用
- ◆USB ケーブル : イメージファイルをダウンロードするために使用
- ◆12V/2A DC 電源アダプタ : 開発ボードに電源提供
- ◆SD カード : 初回書き込む時、SD カードから起動する (2GB 未満容量の SD カードを使用してください)。
- ◆USB インタフェースの SD カードリーダー : PC から起動ファイルを SD カードにコピーする

2.2 ホスト環境

二種類ある。

1 : 二台の PC を用意する (お勧め)。一台は **ubuntu9.04** をインストールして、ブートローダ、カーネル、アプリ、NFS サーバーなどをコンパイルするために使用する。もう一台は Windows XP をインストールし、コンパイル済みのイメージファイルを開発ボードに書き込むために使用。

2 : PC が一台しかない場合は Windows XP+ VMware6.0 (以上) + ubuntu9.04 の環境でも良い。

このマニュアルでは、上記 1 を前提で説明する。



第三章 イメージファイルの書き込む

3.1 主なイメージファイル

Linux2.6.28 のイメージファイル：

- ①SDboot.bin
- ②u-boot-nand.bin
- ③zImage
- ④rootfs

3.1.1 SDboot

SDboot は開発ボードの SD カードから起動時用のイメージファイルで、主に初回書き込みの際に使用のブートとなる。Nand Flash が空の時、SD カードを介して SDboot を起動する。SDboot は USB OTG で高速ダウンロードし Nand Flash に書き込む事をサポートする。

3.1.2 u-boot-nand

u-boot-nand は Nand Flash に書き込まれて、Nand Flash から起動する際に使われるイメージファイルである。u-boot-nand は USB OTG でダウンロード、Nand Flash のリード、書き込み、消去などをサポートする。u-boot-nand のベースは u-boot-1.1.6 バージョンで、このバージョンで対応している全てのコマンドと機能（ネットワーク機能以外）を実現している。

3.1.3 zImage

zImage はカーネルイメージファイルで、ソースコードの urbetter-linux2.6.28-v1.0 バージョンに基づいている。

3.1.4 rootfs

rootfs はルートファイルシステムで、ここで提供しているルートファイルシステムのタイプは YAFFS2 である。下記 2 つの起動方法をサポートしている：

- 1、NFS ネットワークルートファイルシステム
- 2、Nand Flash から YAFFS2 のファイルシステムを起動する

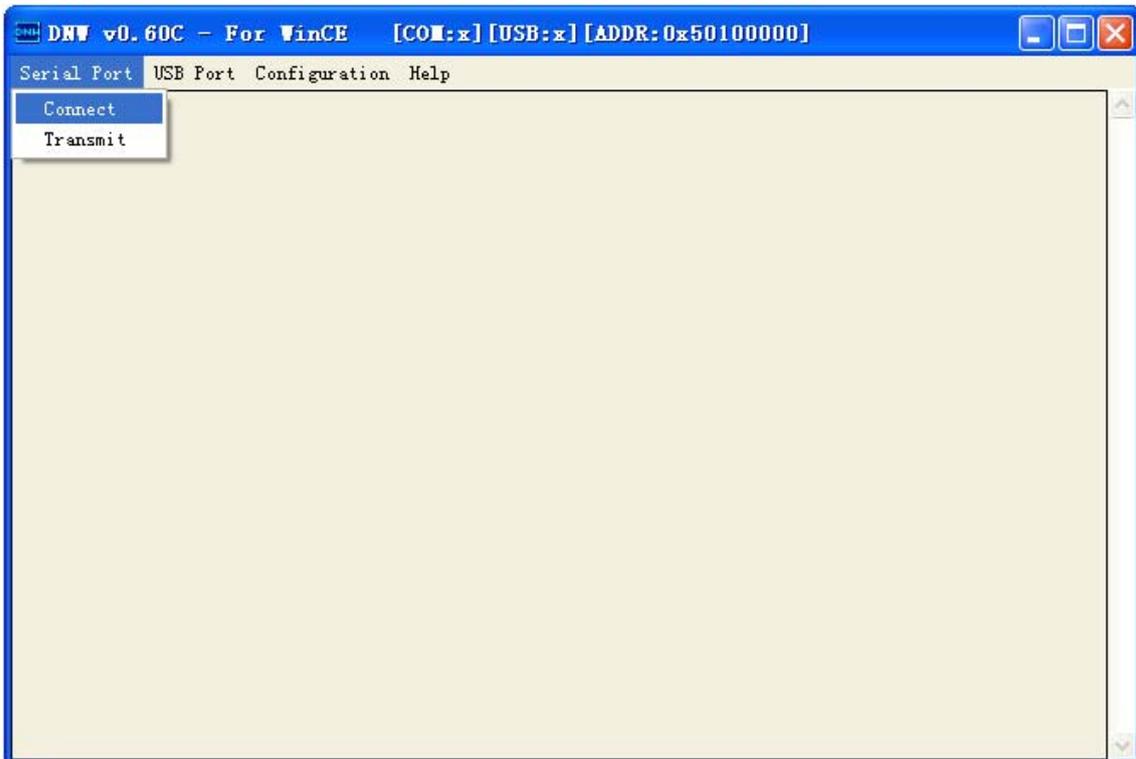
3.2 開発ボード書き込み時の注意事項

3.2.1 注意事項一

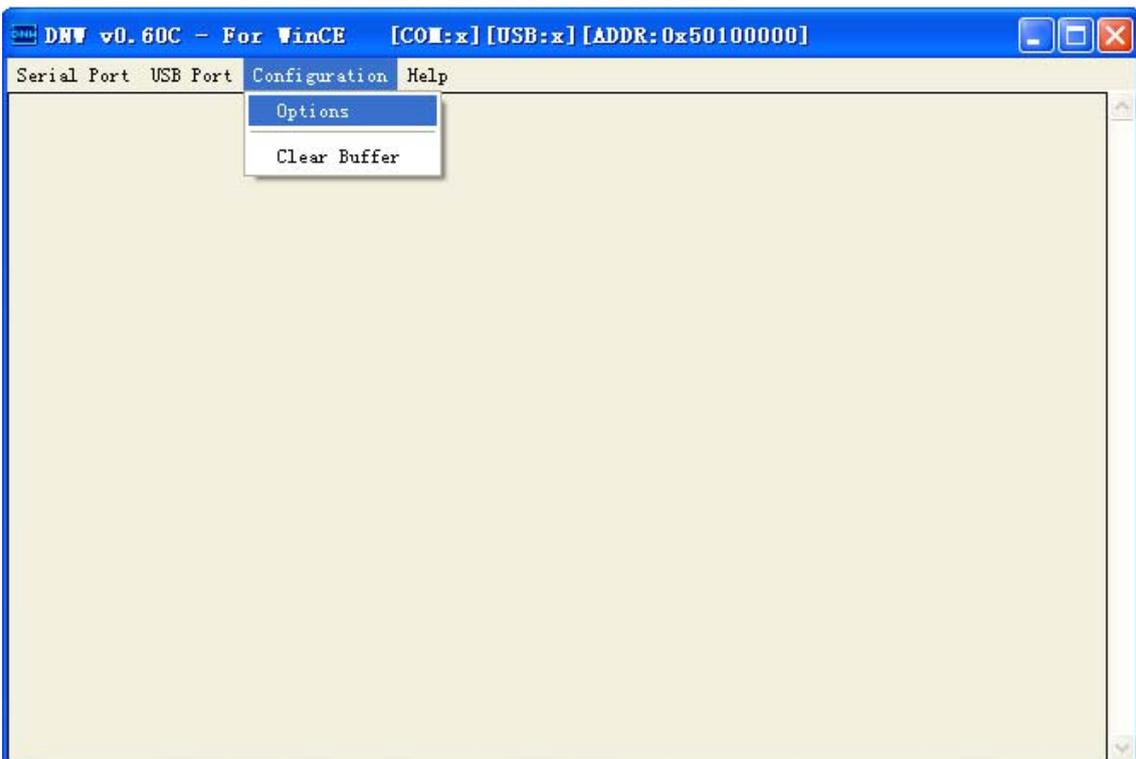
デフォルトでは wince6.0 がインストールされている。PC とのシリアル通信を確認ください。次の注意点がある：

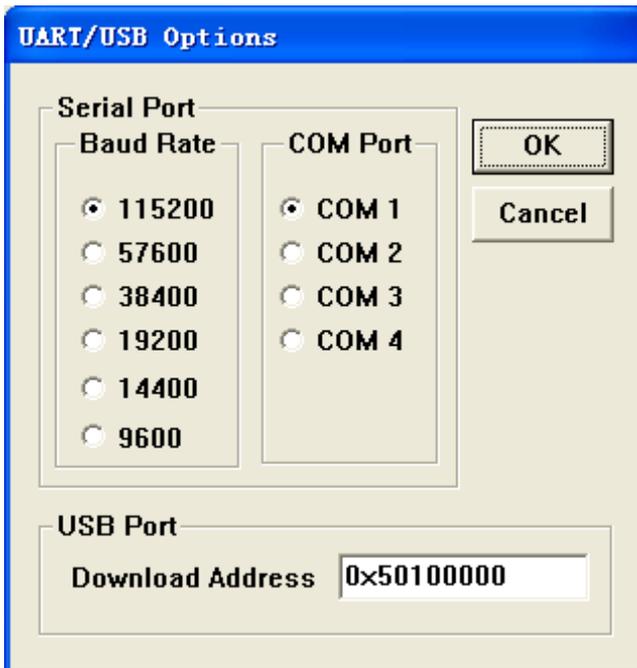
- 1、弊社提供しているクロスシリアル線を使ってください。USB-COM 変換ケーブルはちょっと不安定なので避けて欲しい。
- 2、windows xp 環境で DNW 或いは他のシリアルデバッグターミナルを起動して、DNW を設定する。設定手順は下記です：

- ◆ Serial Port -> Connect



◆ Configuration -> Options





3、電源入れると、シリアルターミナルに起動情報が表示される（下記は EBOOT 情報である）：

```
DNW v0.60C - For WinCE [COM1, 115200bps] [USB:x] [ADDR: 0x50100000]
Serial Port USB Port Configuration Help
[Eboot] --InitializeDisplay()
+OALArgsInit()
SocID:0x36410101
Arguments area is initialized
-OALArgsInit()
INFO: (unsigned)C_IsrHandler : 0x80064A38
INFO: (unsigned)ASM_IsrHandler : 0x80066EA8
INFO: (unsigned)pISR : 0xEA019BA2
BP_Init
[FMD] ++FMD_Init() ****
[FMD:INF] FMD_Init() : Read ID = 0x0000ecda
[FMD] FMD_Init() : NUM_OF_BLOCKS = 2048
[FMD] FMD_Init() : PAGES_PER_BLOCK = 64
[FMD] FMD_Init() : SECTORS_PER_PAGE = 4
[FMD] --FMD_Init()
[FMD] FMD_GetInfo() : NUMBLOCKS = 2048(0x800), SECTORSPERBLOCK = 64(0x40),
BYTESPERSECTOR = 2048(0x800)
[FMD] FMD_GetInfo() : NUMBLOCKS = 2048(0x800), SECTORSPERBLOCK = 64(0x40),
BYTESPERSECTOR = 2048(0x800)
wNUM_BLOCKS : 2048(0x800)
Press [ENTER] to launch image stored on boot media, or [SPACE] to enter boot
monitor.

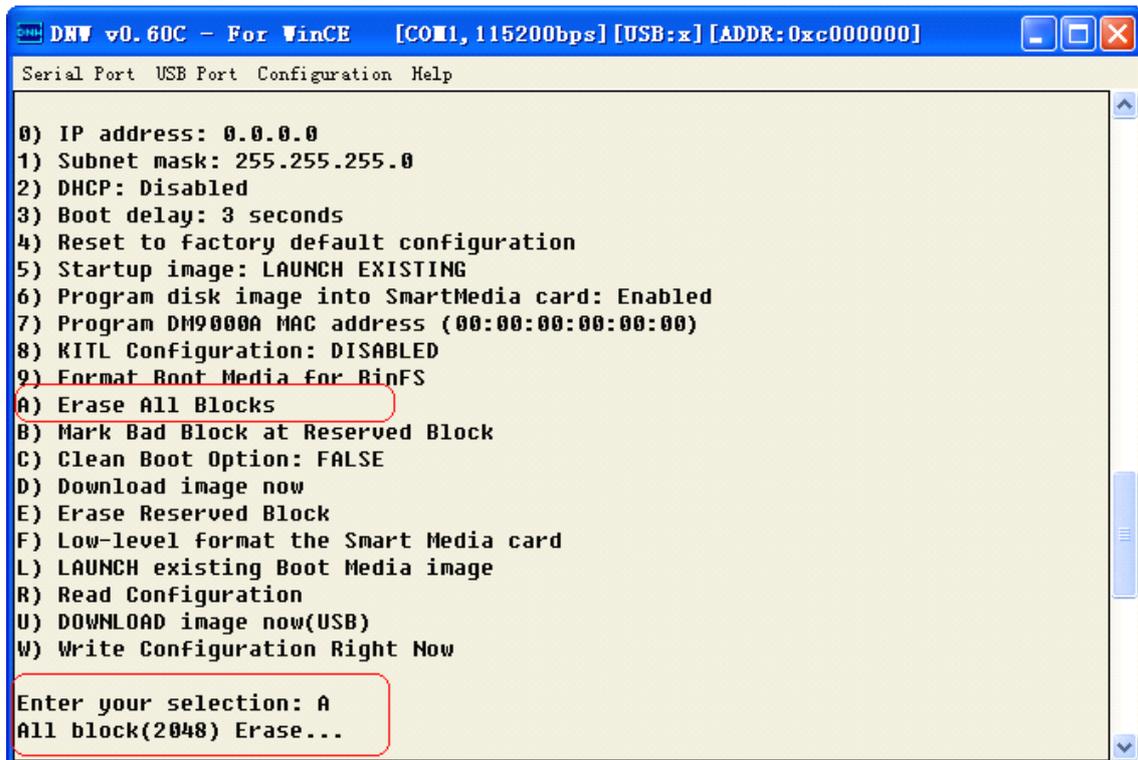
Initiating image launch in 1 seconds.
```

3.2.2 注意事項二

出荷時のデフォルト OS は Wince6.0 なので、Linux のイメージファイルを書き込む前に、Wince の Eboot で「A」を選択して、Nand Flash をフォーマットする必要。しないと、正常に u-boot を書き込みあるいは実行できない。

Nand Flash をフォーマットする手順：

- 1、シリアル線でボードの Com0 と PC を接続する。
- 2、Windows xp で DNW を起動する。
- 3、電源をいれて、3 秒以内に space key 押して eboot 画面に入る。
- 4、「A」を選択して、フォーマットする。

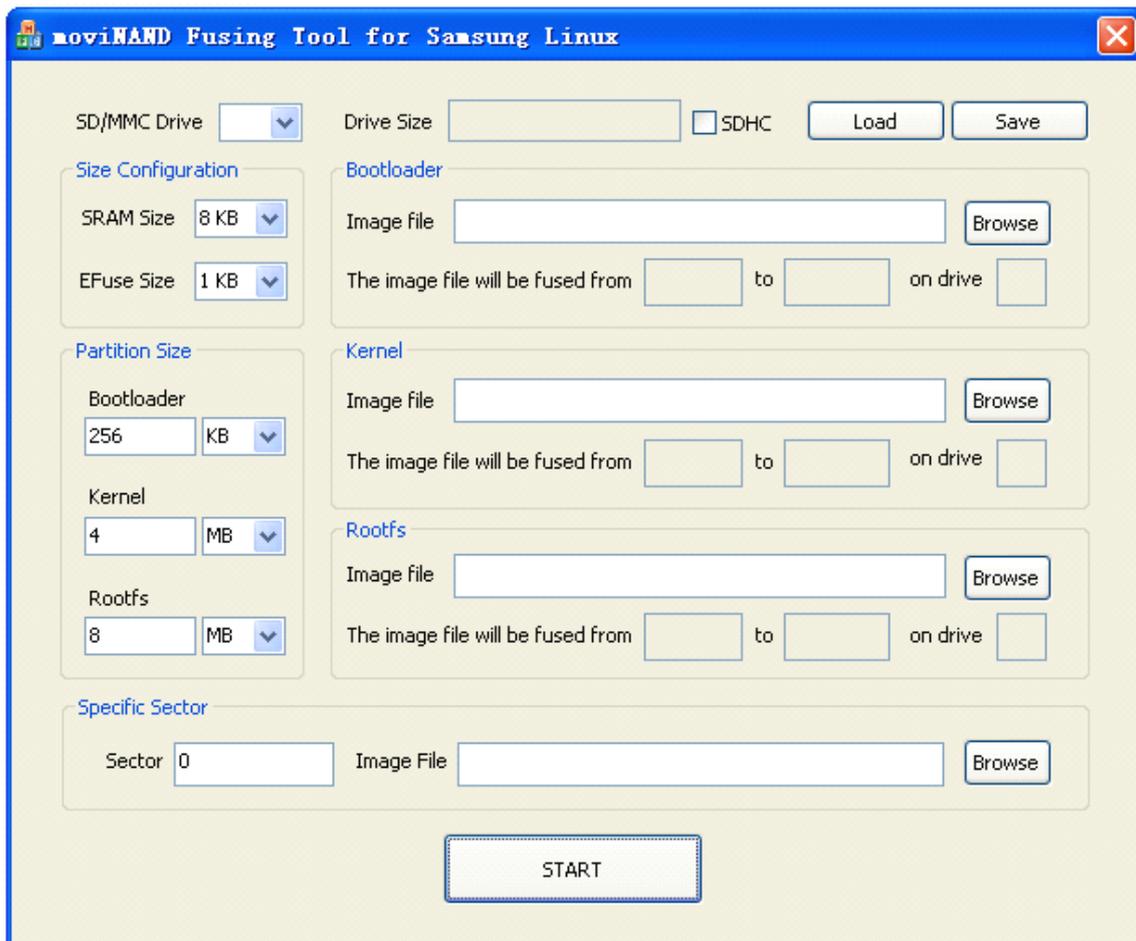


3.2.3 注意事項三

- 1、起動用の SD カードを作る。標準の SD カードで 2GB 以内のをお勧め。
- 2、SD カードに SDboot を書き込む時、SD カードリーダーを使ってください。ノートパソコン内蔵のカードリーダーで書き込む場合は正常に書けない或いは起動できない場合がある。

3.3 起動用 SD カードの作成

- 1、SD カードを USB カードリーダーに挿して、Windows xp 環境で FAT32 ファイルシステムでフォーマットする。
- 2、Windows xp 環境で、Linux2.6.28¥tools¥moviNAND_Fusing_Tool.exe を実行する。

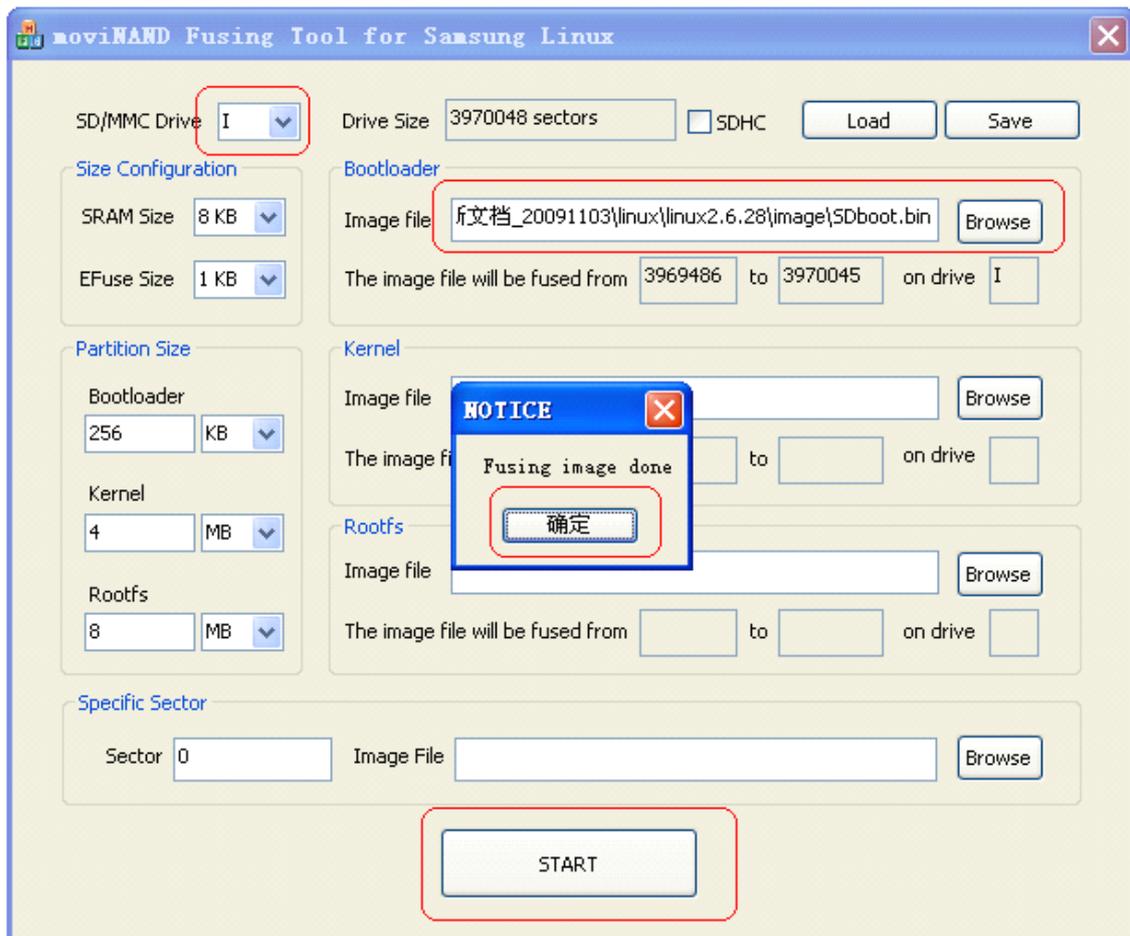


3、SDboot.bin を SD カードに書き込む。

SD/MMC Driver の所は SD カードの Windows xp 上のドライブ名を選択する。

Image file の所は“Browse”をクリックして¥linux2.6.28¥image¥SDboot.bin を選択する。

“START” をクリックする。



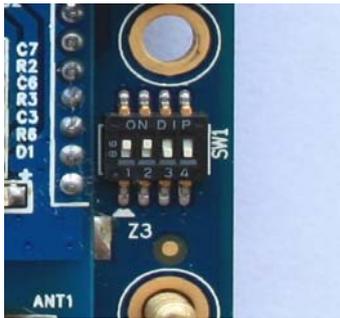
※書き込み成功しても、SD カードから書き込んだデータは見えない、また容量も変わらない。

3.4 u-boot の書き込む

本節では SD カードから SDboot を起動して、SDboot のコマンドで USB OTG で u-boot-nand を Nand Flash に書き込む。初めて書き込む時あるいは Nand Flash 中の u-boot が壊れた場合、この方法で u-boot を Nand Flash に再書き込むできる。

3.4.1 起動モードの設定

起動モード	OM1	OM2	OM3	OM4
SW1 の Pin	1	2	3	4
SD カード起動	1	1	1	1
Nand Flash 起動	1	1	0	0



デフォルトは Nand Flash 起動なので、SD カード起動モードに設定するには、SW1 の 3,4 Pin を ON にする必要。

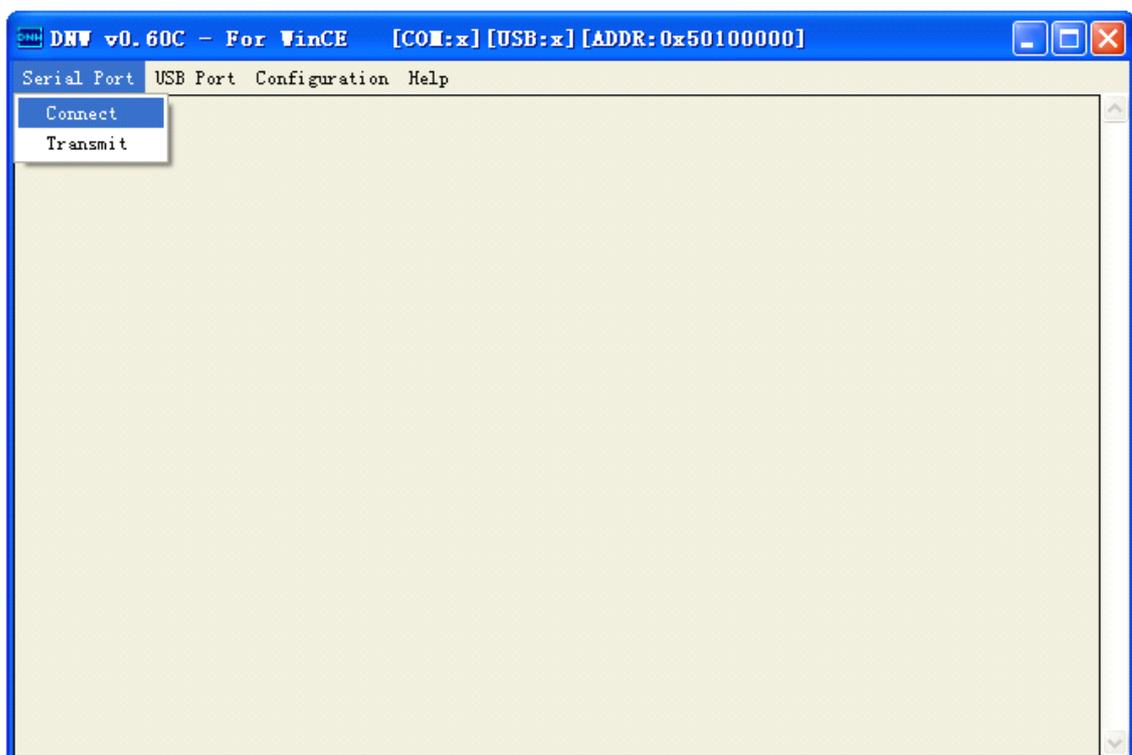
※1、SW1 の ON の場合は “1” で、OFF の場合は “0”、“X” は High 或いは Low 電流。

2、idea6410 の出荷時の設定は Nand Flash 起動モード。

SD カードで SDboot を起動する時は、起動モード設定 SW1 を SD カードモードに設定する必要。つまり、SW1 の 1 から 4 が : 1 1 1 1

3.4.2 SDboot の起動

- 1、起動用 SD カードをボードの SD 口に挿入する。
- 2、ボードと PC の間にシリアル線と USB 線で接続する。
- 3、Windows xp で DNW を起動し、接続 COM を設定する。

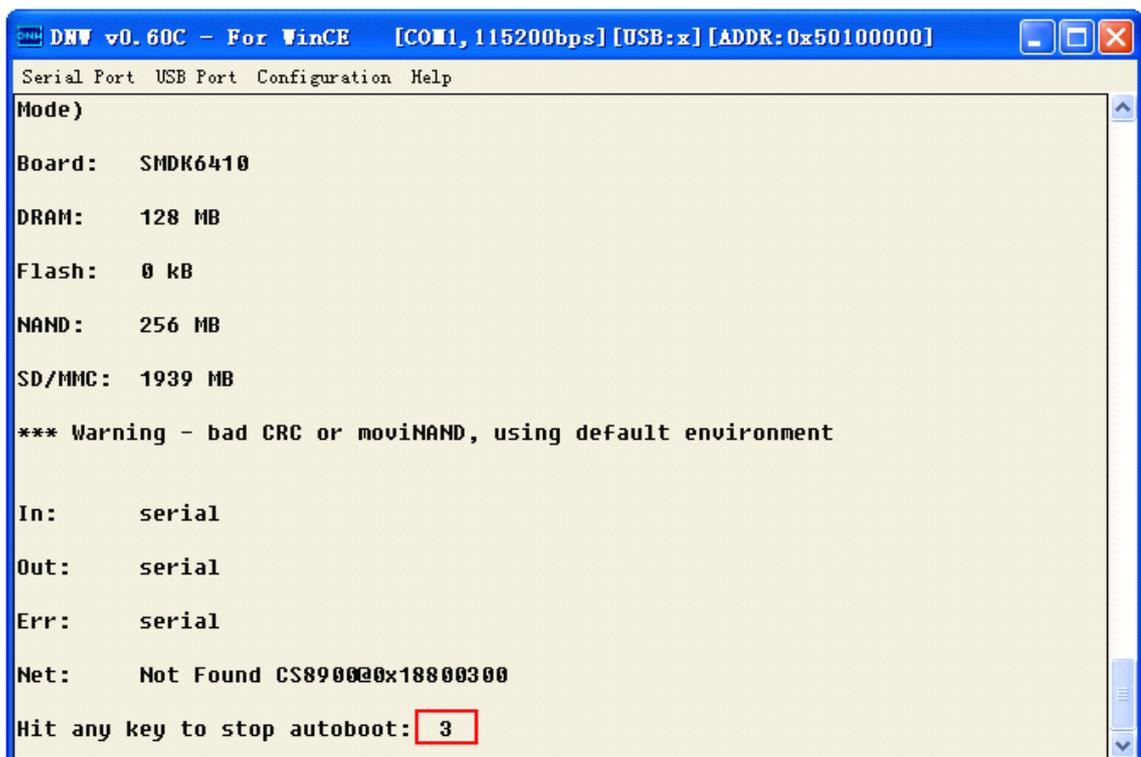


◆ DNW のメニューの “Serial Port” -> “Connet”、接続成功したら、ステータス欄

に[COM1,115200bps]が表示される。



4、ボードの電源入れると、DNW から情報が出てくる、bootdelay はデフォルトで 3 秒。



◆ 3 秒以内に space key を押すと、SDboot コマンドラインになる。



```
DNW v0.60C - For WinCE [COM1,115200bps] [USB:x] [ADDR:0x50100000]
Serial Port USB Port Configuration Help
Board: SMDK6410
DRAM: 128 MB
Flash: 0 kB
NAND: 256 MB
SD/MMC: 1939 MB
*** Warning - bad CRC or moviNAND, using default environment
In: serial
Out: serial
Err: serial
Net: Not Found CS8900@0x18800300
Hit any key to stop autoboot: 0
SMDK6410 #
```

3.4.3 Nand Flash のフォーマット

- ◆ “nand erase 0 10000000” コマンドを実行する。

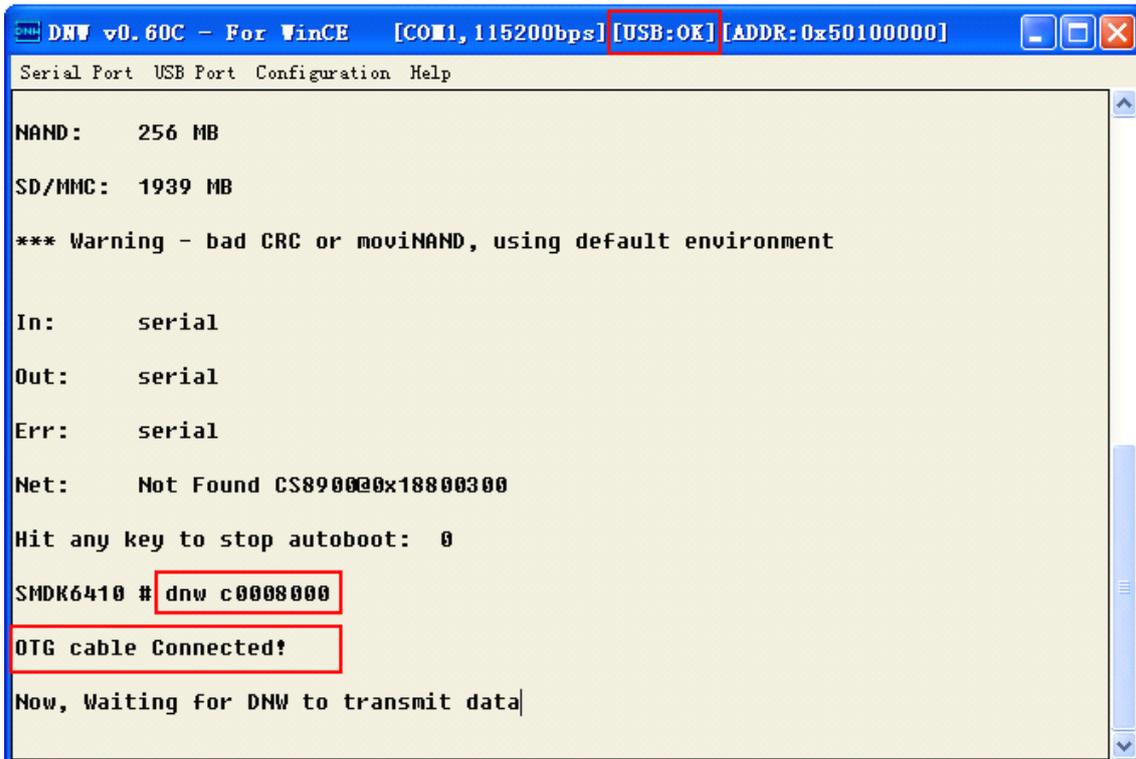
```
DNW v0.60C - For WinCE [COM1,115200bps] [USB:x] [ADDR:0x50100000]
Serial Port USB Port Configuration Help
Board: SMDK6410
DRAM: 128 MB
Flash: 0 kB
NAND: 256 MB
SD/MMC: 1939 MB
*** Warning - bad CRC or moviNAND, using default environment
In: serial
Out: serial
Err: serial
Net: Not Found CS8900@0x18800300
Hit any key to stop autoboot: 0
SMDK6410 # nand erase 0 10000000
```

3.4.4 u-boot-nand.bin を Nand Flash に書き込む

- ◆ “dnw c0008000” コマンドを実行する。

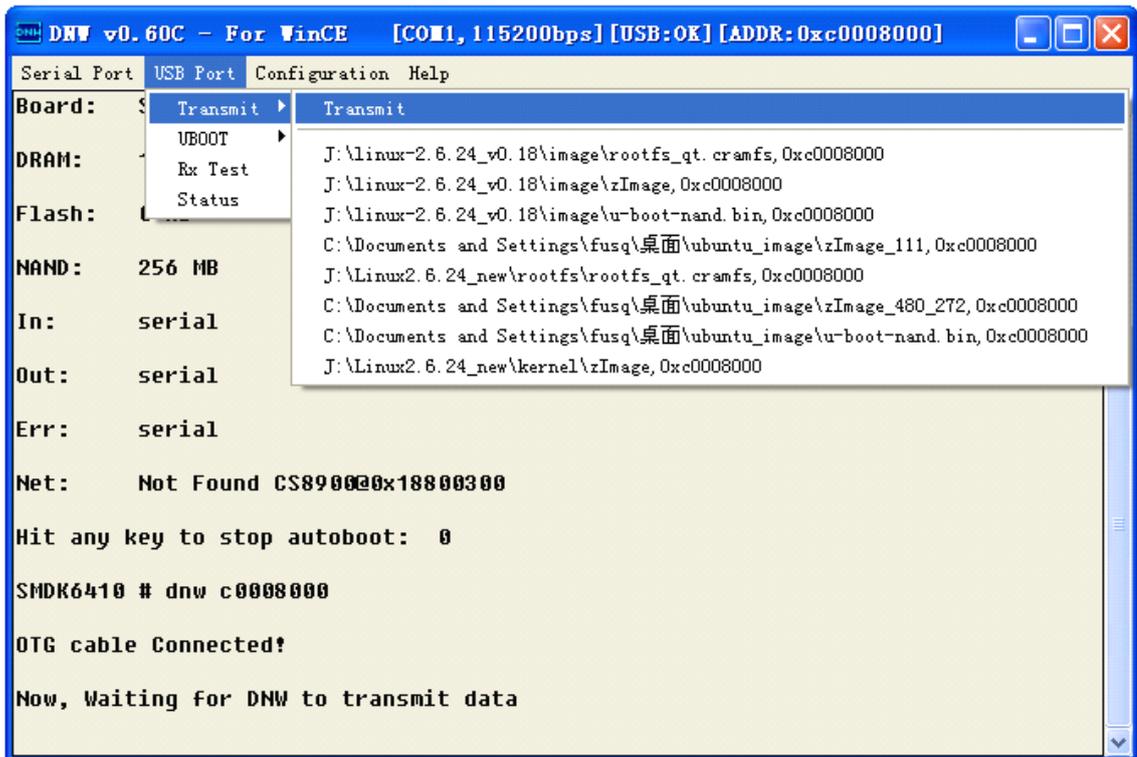
PC 上にまだ USB ドライバをインストールしていない場合は、自動で新しいハードウェアを発見されて、ドライバをインストールしてから、DNW でイメージファイルをダウンロードできる。USB ドライバは¥linux2.6.28¥tools¥USBdriver を参照する事。

既に USB ドライバがインストールされている場合は、ステータス欄に[USB:OK]の情報が表示されて、「OTG cable Connected!」の様なメッセージが表示される。

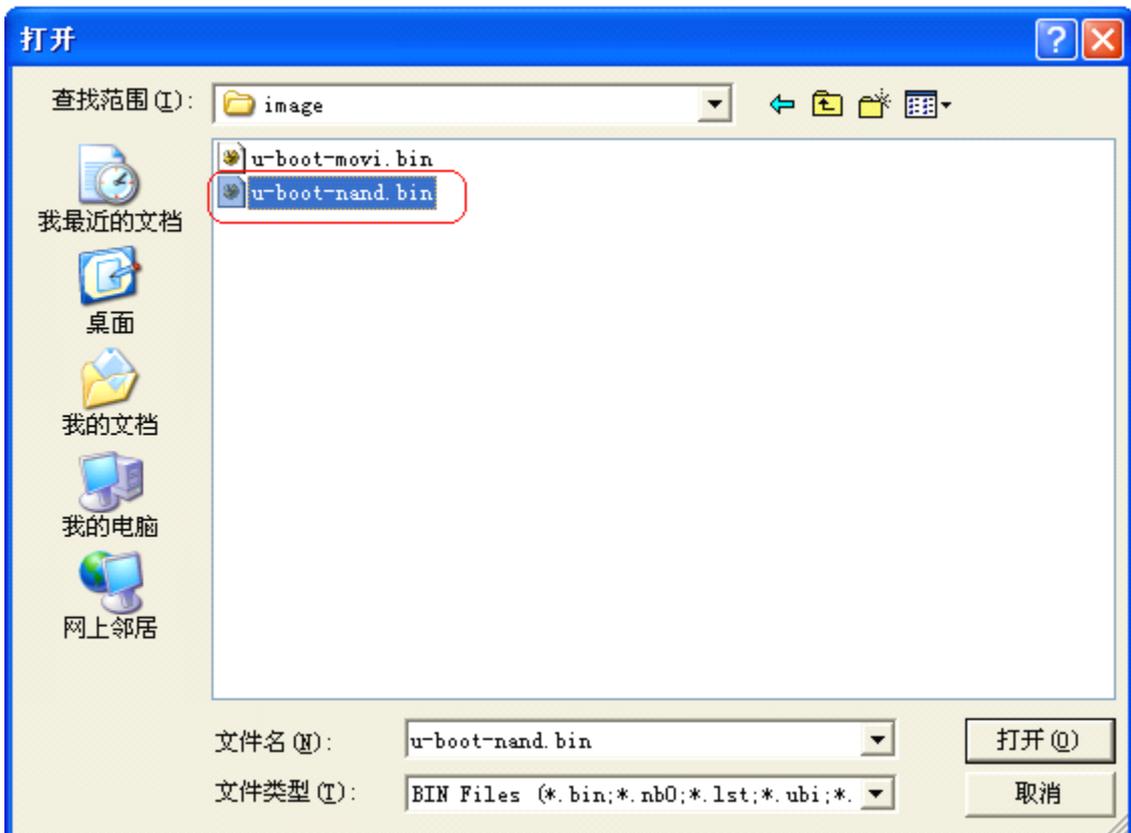


```
DNW v0.60C - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0x50100000]
Serial Port USB Port Configuration Help
NAND: 256 MB
SD/MMC: 1939 MB
*** Warning - bad CRC or moviNAND, using default environment
In: serial
Out: serial
Err: serial
Net: Not Found CS8900@0x18800300
Hit any key to stop autoboot: 0
SMDK6410 # dnw c0008000
OTG cable Connected!
Now, Waiting for DNW to transmit data|
```

- ◆ USB Port ->Transmit -> Transmit の順にクリックする。



- ◆ ¥linux2.6.28¥image¥u-boot-nand.bin を ram 0xc0008000 アドレスに書き込む。



```
DNW v0.60C - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0xc0008000]
Serial Port USB Port Configuration Help
In: serial
Out: serial
Err: serial
Net: Not Found CS89000@0x18800300
Hit any key to stop autoboot: 0
SMDK6410 # dnw c0008000
OTG cable Connected!
Now, Waiting for DNW to transmit data
Download Done!! Download Address: 0xc0008000, Download Filesize:0x30000
Checksum is being calculated.
Checksum O.K.
SMDK6410 #
```

- ◆ “nand write c0008000 0 100000” コマンドを実行する。

```
DNW v0.60C - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0x50100000]
Serial Port USB Port Configuration Help
Erasing at 0xffe0000 -- 100% complete.
OK
SMDK6410 # dnw c0008000
OTG cable Connected!
Now, Waiting for DNW to transmit data
Download Done!! Download Address: 0xc0008000, Download Filesize:0x30000
Checksum is being calculated.
Checksum O.K.
SMDK6410 # nand write c0008000 0 100000
NAND write: device 0 offset 0x0, size 0x100000
1048576 bytes written: OK
SMDK6410 #
```

u-boot-nand.bin を nand flash 0x0 アドレスに書き込む。

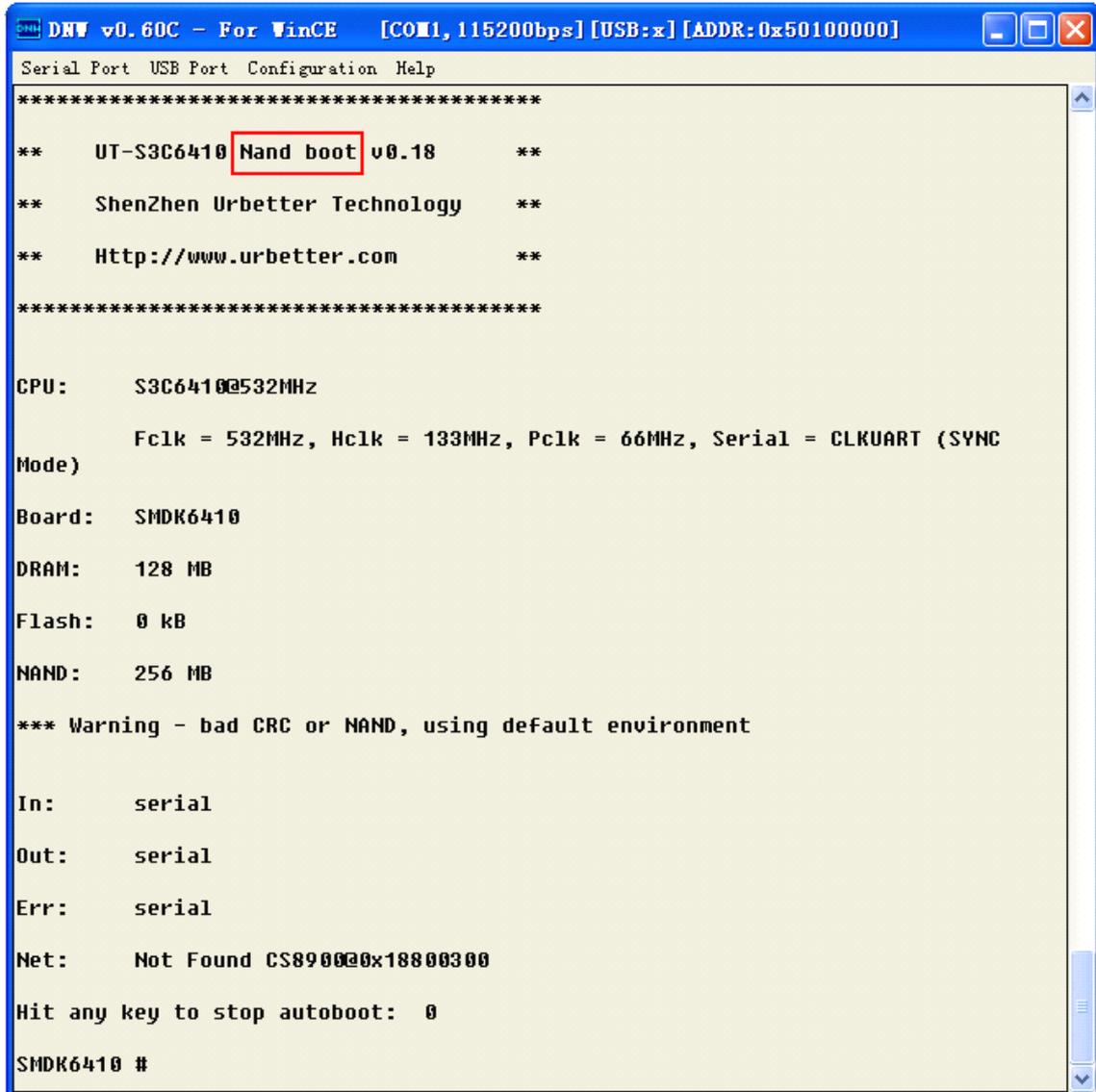
3.4.5 u-boot-nand.bin の書き込みが成功したかの確認

ボードの起動モードを nand flash モードに設定して、起動する。U-boot-nand が正常に起

動できれば、正常に書き込まれている。

具体的には：

- ◆SW1 を nand 起動コードに設定する、1 から 4 が 1 1 0 0。
- ◆SD カードを抜き出して、起動する。



```
Serial Port USB Port Configuration Help
*****
**   UT-S3C6410 Nand boot v0.18   **
**   ShenZhen Urbetter Technology   **
**   Http://www.urbetter.com   **
*****

CPU:      S3C6410@532MHz
          Fclk = 532MHz, Hclk = 133MHz, Pclk = 66MHz, Serial = CLKUART (SYNC
Mode)
Board:    SMDK6410
DRAM:    128 MB
Flash:    0 kB
NAND:    256 MB
*** Warning - bad CRC or NAND, using default environment

In:      serial
Out:     serial
Err:     serial
Net:     Not Found CS8900@0x18800300

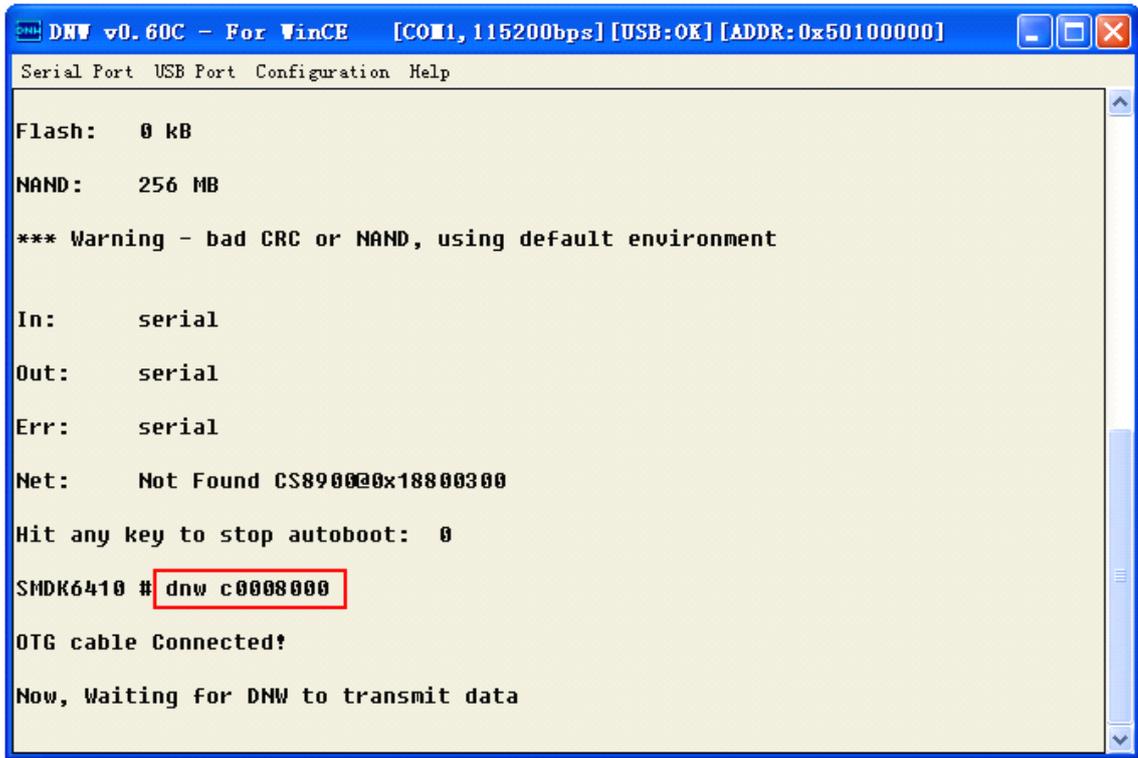
Hit any key to stop autoboot:  0

SMDK6410 #
```

ボードが正常に nand flash から u-boot-nand を起動すれば、u-boot-nand が正常に nand flash に書き込まれて動作する。この後のイメージファイルの書き込みは nand flash の u-boot で実現して、起動 SD カードはもう必要ない。

3.5 zImage の書き込み

- ◆ u-boot コマンドラインに “dnw c0008000” コマンドを実行する。



```
DNW v0.60C - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0x50100000]
Serial Port USB Port Configuration Help

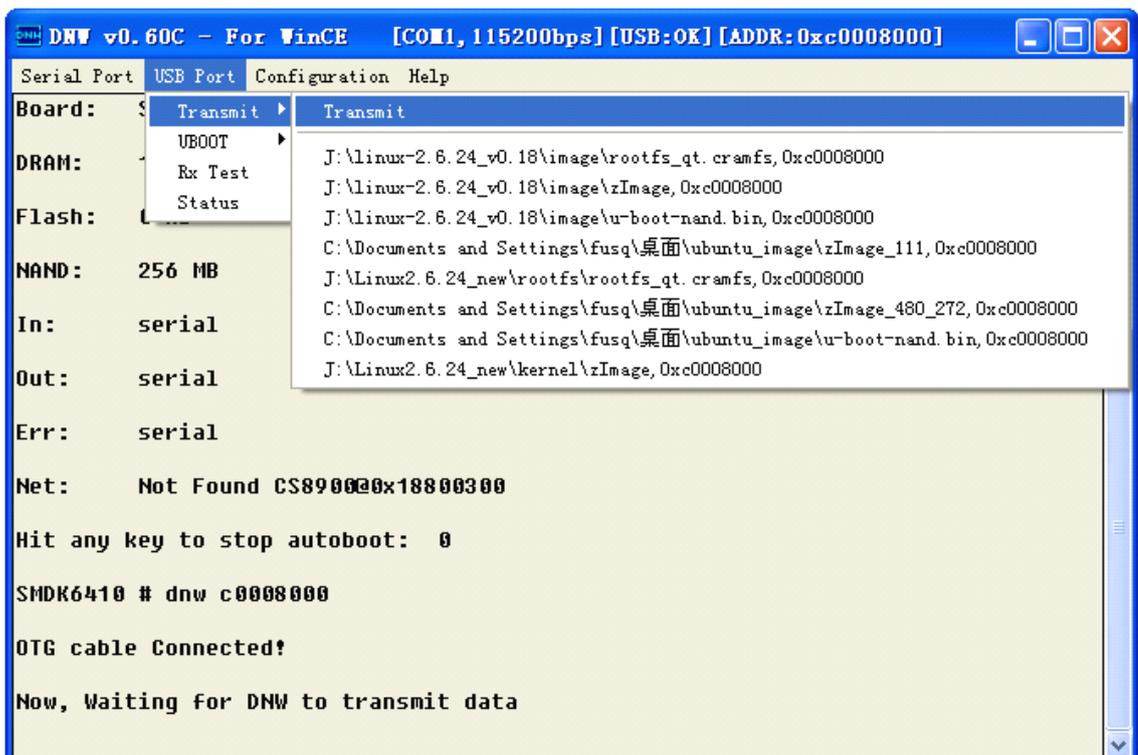
Flash: 0 kB
NAND: 256 MB

*** Warning - bad CRC or NAND, using default environment

In: serial
Out: serial
Err: serial
Net: Not Found CS8900@0x18800300

Hit any key to stop autoboot: 0
SMDK6410 # dnw c0008000
OTG cable Connected!
Now, Waiting for DNW to transmit data
```

- ◆ USB Port → Transmit → Transmit の順にクリックする。

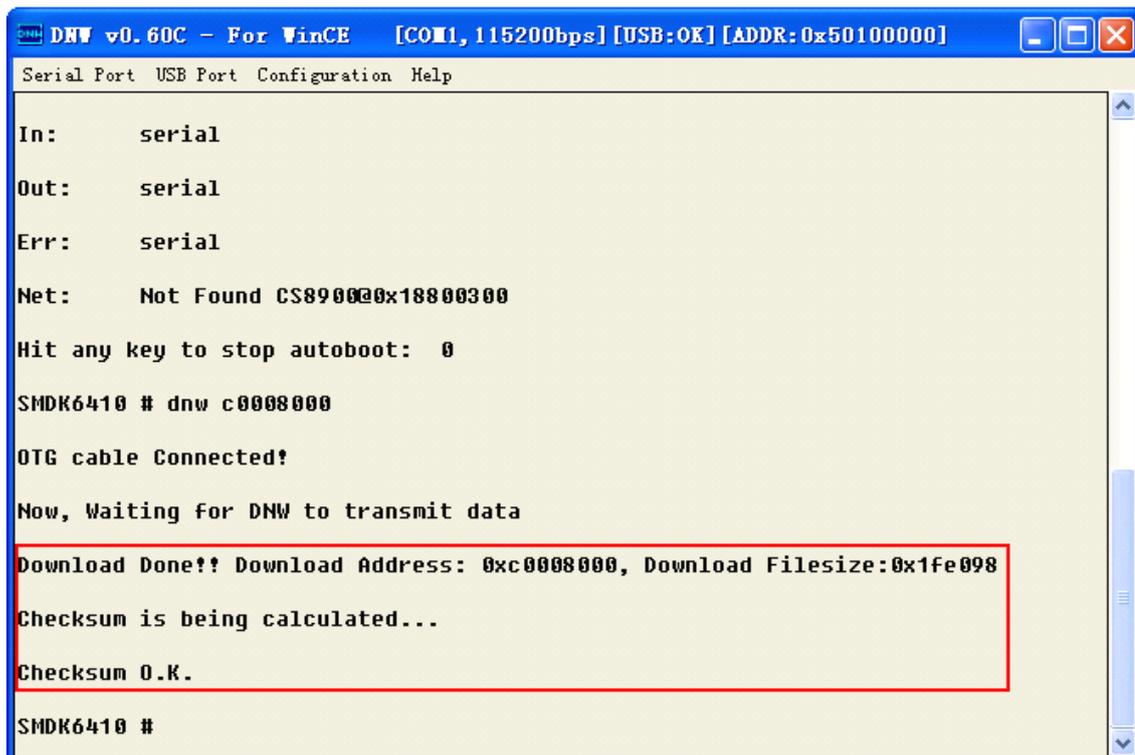
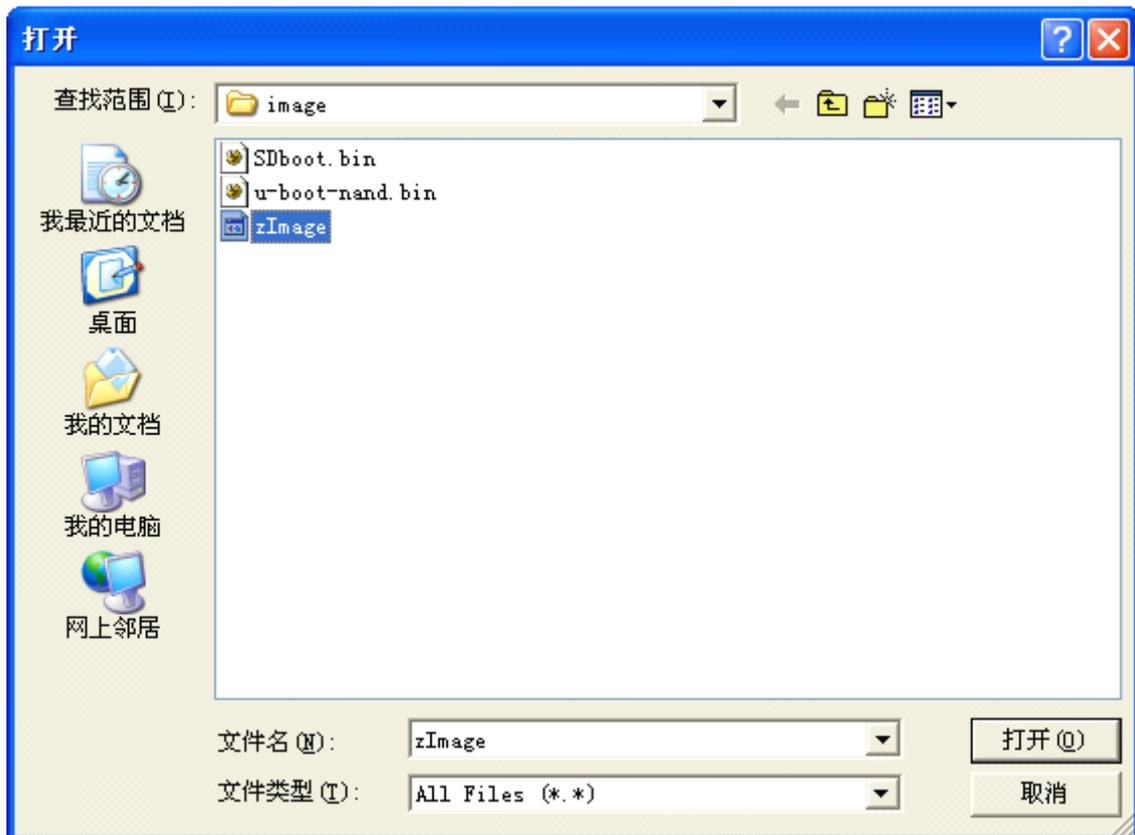


```
DNW v0.60C - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0xc0008000]
Serial Port USB Port Configuration Help

Board: Transmit Transmit
        UBOOT
        Rx Test
        Status
DRAM: J:\linux-2.6.24_v0.18\image\rootfs_qt.cramfs,0xc0008000
Flash: J:\linux-2.6.24_v0.18\image\zImage,0xc0008000
        J:\linux-2.6.24_v0.18\image\u-boot-nand.bin,0xc0008000
        C:\Documents and Settings\fusq\桌面\ubuntu_image\zImage_111,0xc0008000
NAND: 256 MB
        J:\Linux2.6.24_new\rootfs\rootfs_qt.cramfs,0xc0008000
In: serial
        C:\Documents and Settings\fusq\桌面\ubuntu_image\zImage_480_272,0xc0008000
Out: serial
        C:\Documents and Settings\fusq\桌面\ubuntu_image\u-boot-nand.bin,0xc0008000
Err: serial
        J:\Linux2.6.24_new\kernel\zImage,0xc0008000
Net: Not Found CS8900@0x18800300

Hit any key to stop autoboot: 0
SMDK6410 # dnw c0008000
OTG cable Connected!
Now, Waiting for DNW to transmit data
```

- ◆ ¥linux2.6.24¥image¥zImage を ram 0xc0008000 に書き込む。



- ◆ “nand write c0008000 100000 300000” コマンドを実行する。
もし nand flash 0x100000~0x400000 に何か書き込まれている場合は、先にクリアしてか



ら書き込まれる。クリアコマンドは：nand erase 100000 300000。この前に既に0x0~0x10000000をクリアしているので、ここではクリアしなくて書き込む事が出来る。

```
DNW v0.60C - For WinCE [COM1,115200bps] [USB:OK] [ADDR:0x50100000]
Serial Port USB Port Configuration Help
Net:      Not Found CS89000@0x18800300
Hit any key to stop autoboot:  0
SMDK6410 # dnw c0008000
OTG cable Connected!
Now, Waiting for DNW to transmit data
Download Done!! Download Address: 0xc0008000, Download Filesize:0x1fe098
Checksum is being calculated...
Checksum O.K.
SMDK6410 # nand write c0008000 100000 300000
NAND write: device 0 offset 0x100000, size 0x300000
3145728 bytes written: OK
SMDK6410 #
```

3.6 rootfs の書き込む

Yaffs2 ファイルシステムの書き込む流れは次：

Linux ホスト NFS サーバー環境を作る

↓

u-boot の bootargs を NFS 起動に設定する

↓

起動する。システムが起動して自動的に NFS ネットファイルシステムにマウントする。

↓

NFS ファイルシステム起動後、/bin フォルダにある Writerootfs を実行して YAFFS2 ルートファイルシステムを nand に書き込む。

↓

再起動する。U-boot に入って、u-boot の bootargs を nand flash 起動の Yaffs2 ルートファイルシステムに設定する。

↓

書き込み完了後、再起動する。Nand flash から Yaffs2 ファイルシステムを起動する。

3.6.1 NFS ルートファイルシステムのマウント

1、Linux ホスト (ubuntu 9.04) 上に NFS サーバーをインストールする。

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

2、Linux ホスト (ubuntu 9.04) 上に nfs_share 共有フォルダーを作成する。

3、Linux ホスト (ubuntu 9.04) 上の NFS コンフィグファイル (/etc/exports) を修正する。

```
## /srv/nfs4      gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
## /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
##
/home/fusq/nfs_share 192.168.1.128(rw,sync,no_root_squash)
~
~
~
```

Fusq はホストのユーザー名。

/home/fusq/nfs_share はサーバーの共有フォルダー。

192.168.1.128 はボードの IP

4、¥linux2.6.28¥filesystem¥urbetter-rootfs-qt-2.2.0.tgz を Linux ホスト共有フォルダー/home/fusq/nfs_share にコピーする。

5、nfs_share フォルダーで urbetter-rootfs-qt-2.2.0.tgz を解凍する。

```
# sudo tar xvf urbetter-rootfs-qt-2.2.0.tgz
```

これでルートファイルシステムのフォルダーが作成される。



urbetter-rootfs-qt-2.2.0.tgz はそのまま保持する、後でまた使う。

6、Linux ホスト NFS サーバーを再起動する。

```
#sudo /etc/init.d/portmap restart
```

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

7、ボードに LAN ケーブルを挿して、Linux ホストのローカル LAN に接続する。

8、電源入れて、u-boot コマンドラインに入る。

```
U-Boot 1.1.6 (May 11 2009 - 10:23:24) for SMDK6410

*****
**      UT-S3C6410 Nand boot v0.18      **
**      ShenZhen Urbetter Technology    **
**      Http://www.urbetter.com         **
*****

CPU:      S3C6410@532MHz
          Fclk = 532MHz, Hclk = 133MHz, Pclk = 66MHz, Serial = CLQUART (SYNC Mode)
Board:    SMDK6410
DRAM:     128 MB
Flash:    0 kB
NAND:     256 MB
*** Warning - bad CRC or NAND, using default environment

In:       serial
Out:      serial
Err:      serial
Net:      Not Found CS8900@0x18800300
Hit any key to stop autoboot:  0
SMDK6410 #
SMDK6410 #
SMDK6410 # █
```

9、u-boot 起動パラメータを設定する。

◆ bootcmd を設定する。

```
#setenv bootcmd nand read 0xc0008000 0x100000 0x300000¥;bootm 0xc0008000
```

◆ bootargs を設定する。

```
#setenv bootargs "root=/dev/nfs nfsroot=192.168.1.110:/home/fusq/nfs_share
ip=192.168.1.128:192.168.1.110:192.168.1.1:255.255.255.0:www.urbetter.com:eth0:off
console=ttySAC0,115200"
```

◆ パラメータを保存する。

```
#saveenv
```

192.168.1.110 : NFS サーバーホスト IP

注意点：パラメータを設定する時は必ず一行で入力ください。
¥linux2.6.28¥u-boot¥linux-boot.txt ファイルに上記内容を保存しているので、コピーして使う方をお勧め。

10、Reset キーを押して、ボードを再起動する。



起動したら、自動的に NFS ファイルシステムをマウントする。

```
TCP cubic registered
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
UFP support v0.3: implementor 41 architecture 1 part 20 variant b rev 5
s3c2410-rtc s3c2410-rtc: hctosys: invalid date/time
eth0: link down
IP-Config: Complete:
    device=eth0, addr=192.168.1.128, mask=255.255.255.0, gw=192.168.1.1,
    host=www, domain=, nis-domain=urbetter.com,
    bootserver=192.168.1.110, rootserver=192.168.1.110, rootpath=
Looking up port of RPC 100003/2 on 192.168.1.110
eth0: link up, 100Mbps, full-duplex, lpa 0x45E1
Looking up port of RPC 100005/1 on 192.168.1.110
UFS: Mounted root (nfs filesystem).
Freeing init memory: 148K
*****
Welcome to Root FileSystem!

http://www.urbetter.com
*****
mkdir: cannot create directory '/mnt/disk': File exists
yaffs: dev is 32505859 name is "mtdblock3"
yaffs: passed flags ""
yaffs: Attempting MTD mount on 31.3, "mtdblock3"
block 1323 is bad
yaffs_read_super: isCheckpointed 0
Try to bring eth0 interface up.....NFS root ...Done
touch...

Starting Qtopia, please waiting...

Please press Enter to activate this console.
[root@urbetter /]#
[root@urbetter /]# █
```

LCD 上は QT 画面が出て来る。

3.6.2 YAFFS2 ファイルシステムの書き込み

1、NFS ファイルシステム正常に起動後、urbetter-rootfs-qt-2.2.0.tgz はルートフォルダにある事を確認ください。

```
[root@urbetter /]# ls
application_test      proc
bin                   root
dev                   sbin
etc                   sdcard
home                  sys
lib                   tmp
linuxrc               urbetter-rootfs-qt-2.2.0.tgz
mnt                   usr
opt                   var
[root@urbetter /]# █
```

2、./bin/Writerootfs を実行すると、自動的に YAFFS2 のファイルシステム

urbetter-rootfs-qt-2.2.0.tgz を nand flash (mtdblock2) に解凍する。

```
[root@urbetter /]# ./bin/Writerootfs
=====
=          mount to yaffs2 part .....          =
=====
yaffs: dev is 32505858 name is "mtdblock2"
yaffs: passed flags ""
yaffs: Attempting MTD mount on 31.2, "mtdblock2"
yaffs_read_super: isCheckpointed 0
=====
=          decompress file                      =
=====
need several minutes, pls wait.....
=====
=          Root FS has been updated.            =
=          Reboot and Enjoy idea6410!          =
=====
```

/bin/Writerootfs 実行の流れは次の様に確認できる。

```
opt                               var
[root@urbetter /]# vi bin/Writerootfs
#!/bin/sh
echo "=====
echo "=          mount to yaffs2 part .....          ="
echo "=====
mount -t yaffs2 /dev/mtdblock2 /mnt
echo "=====
echo "=          decompress file                      ="
echo "=====
echo "need several minutes, pls wait....."
cd /mnt
tar xzf /urbetter-rootfs-qt-2.2.0.tgz
echo "=====
echo "=          Root FS has been updated.            ="
echo "=          Reboot and Enjoy idea6410!          ="
echo "=====
```

3、YAFFS2 ルートファイルシステムを nand flash に書き込んだ後、u-boot の起動パラメータを設定して、ボードを nand flash から起動する。

◆ ボードを再起動して、u-boot コマンドラインに入る。

◆ bootargs を設定する

```
# setenv bootargs "root=/dev/mtdblock2 rootfstype=yaffs2 console=ttySAC0,115200"
```

◆ パラメータを保存する。

```
#saveenv
```



```
SMDK6410 #  
SMDK6410 # saveenv  
Saving Environment to NAND...  
Erasing Nand...Writing to Nand... done  
SMDK6410 #
```

ボードを再起動すると、nand flash から yaffs2 ルートファイルシステムを起動する。

```
TCP cubic registered  
RPC: Registered udp transport module.  
RPC: Registered tcp transport module.  
UFP support v0.3: implementor 41 architecture 1 part 20 variant b rev 5  
s3c2410-rtc s3c2410-rtc: hctosys: invalid date/time  
yaffs: dev is 32505858 name is "mtdblock2"  
yaffs: passed flags ""  
yaffs: Attempting MTD mount on 31.2, "mtdblock2"  
yaffs_read_super: isCheckpointed 0  
UFS: Mounted root (yaffs2 filesystem).  
Freeing init memory: 148K  
*****  
Welcome to Root FileSystem!  
  
http://www.urbetter.com  
*****  
mkdir: cannot create directory '/mnt/disk': File exists  
yaffs: dev is 32505859 name is "mtdblock3"  
yaffs: passed flags ""  
yaffs: Attempting MTD mount on 31.3, "mtdblock3"  
block 1323 is bad  
yaffs_read_super: isCheckpointed 0  
Try to bring eth0 interface up.....eth0: link down  
Done  
touch...  
  
Starting Qtopia, please waiting...
```

第四章 イメージファイルのコンパイル

4.1 クロスコンパイラーのインストール

1、linux2.6.28¥cross_compile¥cross-4.2.2-eabi.tar.bz2 を Linux ホストワークフォルダにコピーする。

2、Linux ホストで下記コマンドを実行して、クロスコンパイラーをインストールする。

```
# sudo tar jxvf cross-4.2.2-eabi.tar.bz2 -C /user/local/arm
```

3、確認する。

```
# ls /user/local/arm
```

```
fusq@fusq-urbetter:~$ ls /usr/local/arm/  
3.3.2-test 3.4.1-test 4.2.2-eabi arm-none-linux-gnueabi  
fusq@fusq-urbetter:~$  
fusq@fusq-urbetter:~$
```

```
fusq@fusq-urbetter:~$ ls /usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-  
arm-linux-addr2line arm-linux-cpp arm-linux-gprof arm-linux-ranlib  
arm-linux-ar arm-linux-g++ arm-linux-ld arm-linux-readelf  
arm-linux-as arm-linux-gcc arm-linux-ldconfig arm-linux-size  
arm-linux-c++ arm-linux-gcc-4.2.2 arm-linux-nm arm-linux-strings  
arm-linux-cc arm-linux-gccbug arm-linux-objcopy arm-linux-strip  
arm-linux-c++filt arm-linux-gcov arm-linux-objdump  
fusq@fusq-urbetter:~$
```

4.2 u-boot ソースパケットのコンパイル

1、¥linux2.6.28¥u-boot¥urbetter-u-boot-1.1.6-v1.0.tgz ソースパケットを Linux ホストワークフォルダにコピーする。

2、urbetter-u-boot-1.1.6-v1.0.tgz を解凍する。

```
# tar zxvf urbetter-u-boot-1.1.6-v1.0.tgz
```

3、urbetter-u-boot-1.1.6-v1.0 フォルダで下記コマンドを実行する。

```
# make clean
```

```
# make smdk6410_config
```

```
# make
```

実行後、カレントフォルダに u-boot.bin ファイルが生成される。ファイル名を u-boot-nand.bin に変更する。

4.3 カーネルソースパケットのコンパイル

1、¥linux2.6.28¥kernel ¥ urbetter-linux2.6.28-v1.0.tgz ソースパケットを Linux ホストワークフォルダにコピーする。

2、urbetter-linux2.6.28-v1.0.tgz を解凍する。

```
# tar zxvf urbetter-linux2.6.28-v1.0.tgz
```

3、urbetter-linux2.6.28-v1.0 フォルダで、下記コマンドを実行する。

```
# make clean
```

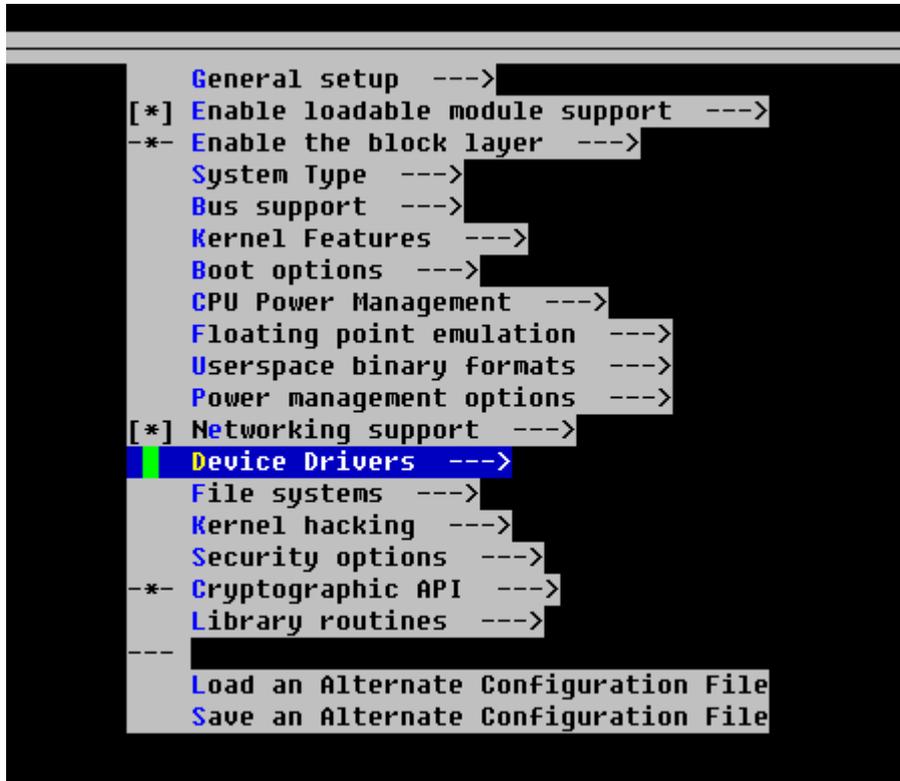
4.3.1 カーネルコンフィグファイルの設定

1、コンフィグファイルを修正する。

```
# cp urbetter-LCD7A.config .config
```

```
# make menuconfig
```

```
Device Driver ---->
```



```
General setup ---->
[*] Enable loadable module support ---->
--* Enable the block layer ---->
System Type ---->
Bus support ---->
Kernel Features ---->
Boot options ---->
CPU Power Management ---->
Floating point emulation ---->
Userspace binary formats ---->
Power management options ---->
[*] Networking support ---->
Device Drivers ---->
File systems ---->
Kernel hacking ---->
Security options ---->
--* Cryptographic API ---->
Library routines ---->
-----
Load an Alternate Configuration File
Save an Alternate Configuration File
```

```
Graphics support ---->
```



```
^(-)
SCSI device support --->
< > Serial ATA (prod) and Parallel ATA (experiment.
[ ] Multiple devices driver support (RAID and LUM)
[*] Network device support --->
[ ] ISDN support --->
Input device support --->
Character devices --->
<*> I2C support --->
[ ] SPI support --->
-* GPIO Support --->
< > Dallas's 1-wire support --->
< > Power supply class support --->
<*> Hardware Monitoring support --->
< > Generic Thermal sysfs driver --->
[*] Watchdog Timer Support --->
Sonics Silicon Backplane --->
Multifunction device drivers --->
Multimedia devices --->
Graphics support --->
<*> Sound card support --->
[*] HID Devices --->
[*] USB support --->
<*> MMC/SD/SDIO card support --->
< > Sony MemoryStick card support (EXPERIMENTAL)
[ ] Accessibility support --->
```

Support for frame buffer devices --->

```
< > Lowlevel video output switch controls
<*> Support for frame buffer devices --->
[ ] Backlight & LCD device support --->
Display device support --->
Console display driver support --->
[*] Bootup logo --->
```

Select LCD Type (UT_LCDxxxxxx) --->

```
--- Support for frame buffer devices
[ ] Enable firmware EDID
[ ] Framebuffer foreign endianness support --->
[ ] Enable Video Mode Handling Helpers
[ ] Enable Tile Blitting Support
*** Frame buffer hardware drivers ***
<*> S3C Framebuffer Support
  Select LCD Type (UT_LCD7A 800*480) --->
<*> Advanced options for S3C Framebuffer
  Select BPP(Bits Per Pixel) (16 BPP) --->
(4) Number of Framebuffers
[ ] Enable Virtual Screen
[*] Enable Double Buffering
< > Epson S1D13XXX framebuffer support
< > Virtual Frame Buffer support (ONLY FOR TESTING!)
< > E-Ink Metronome/8track controller support
< > Fujitsu MB862xx GDC support
```

該当する LCD タイプを選択する。

```
----- Select LCD Type -----
Use the arrow keys to navigate this window or press the hotkey of
the item you wish to select followed by the <SPACE BAR>. Press
<?> for additional information about this option.

( ) UT_LCD43C_D 480*272
( ) UT_LCD5A 640*480
(X) UT_LCD7A 800*480
( ) UT_LCD104B 800*600
( ) UT_VGA 640*480
( ) UT_SUGA 800*600
u(+)
```

<Select> < Help >

保存して Exit。

4.3.2 カーネルのコンパイル

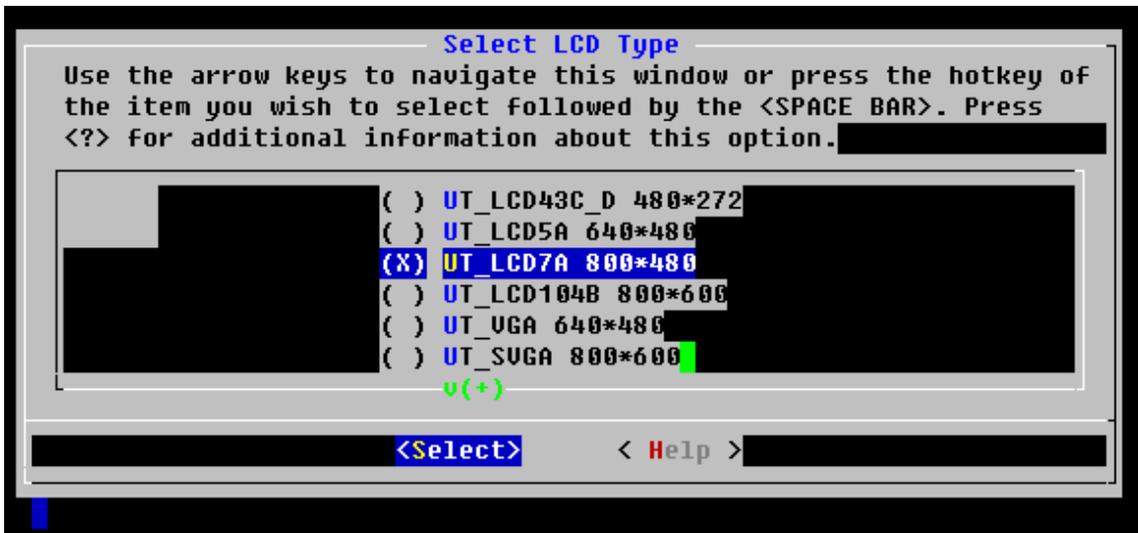
「make」コマンドを実行する。

コンパイル後は、arch/arm/boot/フォルダに zImage ファイルが生成される。

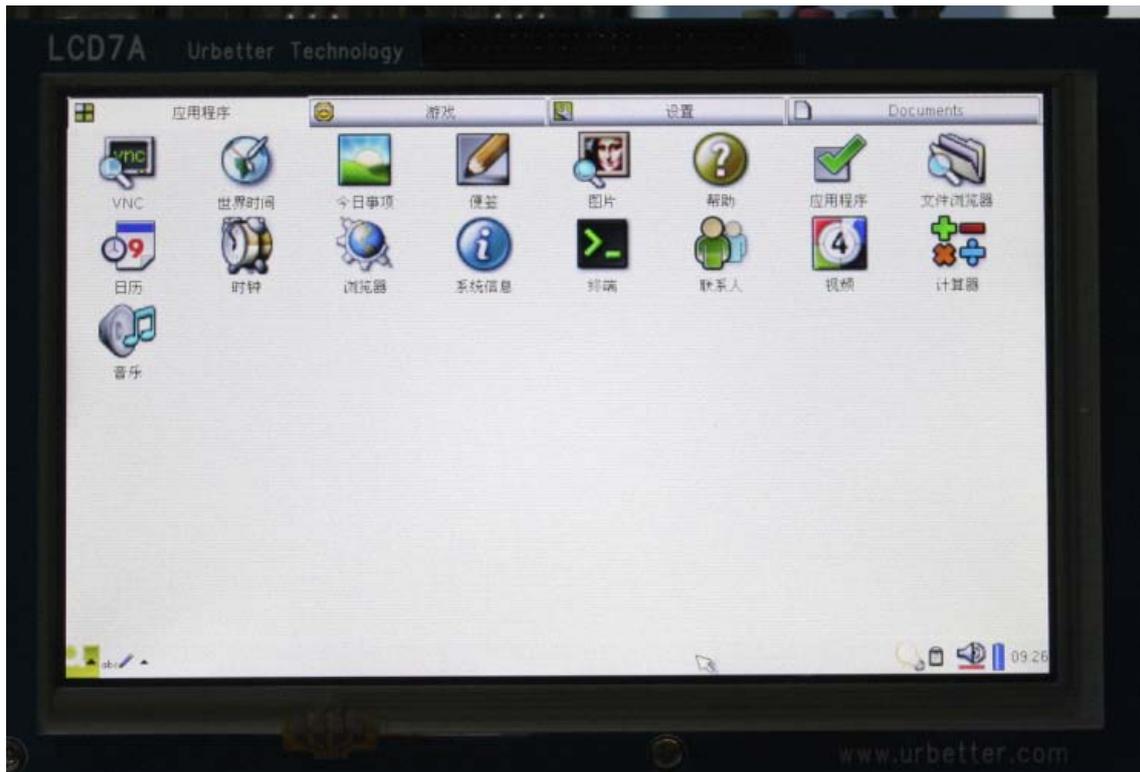
第五章 ドライバとインタフェースのテスト

5.1 LCD テスト

make menuconfig で該当する LCD タイプを選択して、コンパイルしてできた zImage をボードに書き込むと、起動時に該当するドライバが搭載される。



下記画面は UT_LCD7A の表示結果：



5.2 タッチパネルのテスト及び訂正

初めて起動時はUSB マウスで訂正する必要。USB マウス機能はデフォルトではOFFですが、OPEN する方法は「5.41 USB マウスのテスト」を参照ください。



5.3 SD カードのテスト

SD カードをボードに挿入する。

```
[root@urbetter /]#  
[root@urbetter /]# mmc0: new high speed SD card at address 8bde  
mmcblk0: mmc0:8bde SD02G 1.83 GiB  
mmcblk0: p1
```

自動的にルートフォルダの sdcard フォルダにマウントされる。「ls」コマンドでマウントが成功したか確認できる。

```
[root@urbetter /]# ls sdcard/  
3d_test          android-test  
3g               autorun.inf  
Bluetooth       block0.nb0  
CEFlash         camera
```

SD カードを拔出す。



```
[root@urbetter /]#  
[root@urbetter /]#  
[root@urbetter /]# mmc0: card 8bde removed  
[root@urbetter /]#
```

5.3.1 SDカードの読み出しテスト

SDカード上の logo.bin ファイルを/tmp/フォルダにコピーする。

```
[root@urbetter /]#  
[root@urbetter /]#  
[root@urbetter /]# cp /sdcard/logo.bin /tmp/  
[root@urbetter /]#
```

「ls」コマンドで確認する。

```
[root@urbetter /]#  
[root@urbetter /]# ls tmp/  
logo.bin      qtembedded-0  qtopia-0  
[root@urbetter /]#  
[root@urbetter /]#
```

読み出す結果 OK です。

5.3.2 SDカード書き込みテスト

上記でコピーした tmp/logo.bin ファイルを SD カードに logo1.bin にネーム変更してコピーする。

```
[root@urbetter /]#  
[root@urbetter /]#  
[root@urbetter /]# mv tmp/logo.bin /sdcard/logo1.bin  
[root@urbetter /]#  
[root@urbetter /]#
```

「ls」コマンドで確認する。

```
[root@urbetter /]#  
[root@urbetter /]# ls sdcard/  
3d_test          autorun.inf  
3g               block0.nb0  
Bluetooth        camera  
CEFlash          cmda_e  
CameraDemo.rar  dcim  
EvdoUsbcdcDriver.dll  denis_wei  
Flash            eboot.bin  
GPS_test         exe  
HelloWorld       firefox  
IROM_Fusing_Tool.exe  game  
IROM_SD_EBOOT_update.nb0  jetcet  
JPEG_TEST        logo.bin  
M7_Init.exe      logo1.bin  
MultiMediaDemo  nk.bin  
Navigator         npopuk
```

5.4 USB テスト

5.4.1 USB マウステスト

デフォルトは OFF ですが、下記手順で ON にする。

- 1、シリアルターミナル側で `vi ./bin/qtopia` コマンドを実行する。

```
[root@urbetter /]#  
[root@urbetter /]# vi ./bin/qtopia █
```

- 2、マウスとタッチ機能をオープンする（デフォルトではコメントアウトされている）。

```
unset TS_INFO_FILE  
  
export QWS_MOUSE_PROTO="TPanel:/dev/input/event0 USB:/dev/input/mice"  
  
export QWS_KEYBOARD=TTY:/dev/tty1  
export KDEDIR=/opt/kde
```

保存して Exit。

- 3、ボードを再起動する。
- 4、USB マウスを USB Host インタフェースに挿入すると、ターミナル側で下記メッセージが表示される：

```
[root@urbetter /]#  
[root@urbetter /]# usb 1-1: new low speed USB device using s3c241  
0-ohci and address 8  
usb 1-1: configuration #1 chosen from 1 choice  
input: USB Optical Mouse as /class/input/input4  
generic-usb 0003:15CA:00C3.0003: input: USB HID v1.10 Mouse [USB  
Optical Mouse] on usb-s3c24xx-1/input0  
[root@urbetter /]# █
```

USB マウスを拔出すと

```
[root@urbetter /]#  
[root@urbetter /]# usb 1-1: USB disconnect, address 8  
[root@urbetter /]#
```

5.4.2 USB キーボードテスト

USB キーボードを USB Host インタフェースに挿入すると、ターミナル側で下記メッセージが表示される：



```
[root@urbetter /]#
[root@urbetter /]# usb 1-1: new low speed USB device using s3c241
0-ohci and address 5
usb 1-1: configuration #1 chosen from 1 choice
input: HID 04d9:1203 as /class/input/input7
generic-usb 0003:04D9:1203.0006: input: USB HID v1.11 Keyboard [H
ID 04d9:1203] on usb-s3c24xx-1/input0
input: HID 04d9:1203 as /class/input/input8
generic-usb 0003:04D9:1203.0007: input: USB HID v1.11 Device [HID
04d9:1203] on usb-s3c24xx-1/input1

[root@urbetter /]#
[root@urbetter /]# █
```

拔出すと、

```
[root@urbetter /]#
[root@urbetter /]# usb 1-1: USB disconnect, address 5

[root@urbetter /]# █
```

5.4.3 USB メモリディスクテスト

USB メモリディスクを USB Host インタフェースに挿入すると、ターミナル側で下記メッセージが表示される：

```
[root@urbetter /]#
[root@urbetter /]# usb 1-1: new full speed USB device using s3c2410-ohci and address 3
usb 1-1: configuration #1 chosen from 1 choice
scsi1 : SCSI emulation for USB Mass Storage devices
scsi 1:0:0:0: Direct-Access    Generic    Mass Storage    8.07 PQ: 0 ANSI: 2
sd 1:0:0:0: [sda] 1019904 512-byte hardware sectors: (522 MB/498 MiB)
sd 1:0:0:0: [sda] Write Protect is off
sd 1:0:0:0: [sda] Assuming drive cache: write through
sd 1:0:0:0: [sda] 1019904 512-byte hardware sectors: (522 MB/498 MiB)
sd 1:0:0:0: [sda] Write Protect is off
sd 1:0:0:0: [sda] Assuming drive cache: write through
sda: sda1
sd 1:0:0:0: [sda] Attached SCSI removable disk
sd 1:0:0:0: Attached scsi generic sg0 type 0
```

USB メモリディスクは自動的に/udisk にマウントされる。「ls」コマンドで確認する。

```
[root@urbetter /]#
[root@urbetter /]# ls udisk/
20090219                               WinHex-v15.1sr-8H.rar
8688 driver                             app_tv
?? Text File Type.txt                  autorun.inf
????                                    comtest_xp.exe
?????.exe                               constants.const
??????                                  djyos
???????                                funy2.flv
??????WinCE?????????.exe              iicdata.txt
??B??D                                  iron_269
```

5.5 LAN テスト

電源入れて、LAN ケーブルを挿入する。



```
[root@urbetter /]#  
[root@urbetter /]# eth0: link up, 100Mbps, full-duplex, lpa 0x45E1  
[root@urbetter /]#
```

ボードからホストを Ping する : ping 192.168.1.110

```
[root@urbetter /]# ping 192.168.1.110  
PING 192.168.1.110 (192.168.1.110): 56 data bytes  
64 bytes from 192.168.1.110: seq=0 ttl=64 time=2.540 ms  
64 bytes from 192.168.1.110: seq=1 ttl=64 time=0.440 ms  
64 bytes from 192.168.1.110: seq=2 ttl=64 time=0.439 ms  
64 bytes from 192.168.1.110: seq=3 ttl=64 time=0.430 ms  
64 bytes from 192.168.1.110: seq=4 ttl=64 time=0.422 ms  
64 bytes from 192.168.1.110: seq=5 ttl=64 time=0.426 ms  
64 bytes from 192.168.1.110: seq=6 ttl=64 time=0.424 ms  
64 bytes from 192.168.1.110: seq=7 ttl=64 time=0.437 ms  
64 bytes from 192.168.1.110: seq=8 ttl=64 time=0.421 ms  
^C  
--- 192.168.1.110 ping statistics ---  
9 packets transmitted, 9 packets received, 0% packet loss  
round-trip min/avg/max = 0.421/0.664/2.540 ms  
[root@urbetter /]#
```

第六章 マルティメディアハードウェアエンコード・デコード機能のテスト

マルチメディアハードウェアエンコード・デコードテストアプリの実行テストアプリでテスト項目を選択する。

- 1、Linux 2.6.28 test/UT_MultiMedia_test フォルダを全て SD カードにコピーする。このフォルダにテストアプリ UT_MultiMedia_test とビデオテストアプリ (TestVectors フォルダ中) が含めている。
- 2、SD カードをボードに挿す。ターミナルから UT_MultiMedia_test フォルダに入る。

```
[root@urbetter /]#  
[root@urbetter /]#  
[root@urbetter /]#  
[root@urbetter /]# cd sdcard/UT_MultiMedia_test/  
[root@urbetter UT_MultiMedia_test]#  
[root@urbetter UT_MultiMedia_test]#  
[root@urbetter UT_MultiMedia_test]#  
[root@urbetter UT_MultiMedia_test]# ls  
TestVectors UT_MultiMedia_test uart_test  
[root@urbetter UT_MultiMedia_test]#  
[root@urbetter UT_MultiMedia_test]#
```

- 3、UT_MultiMedia_test アプリを実行する。

```
# ./ UT_MultiMedia_test
```

```
===== UT-S3C6410 Media Demo Application =====  
www.urbetter.com, v0.1 (2009-10-30)  
=  
= 1. H.264 display =  
= 2. MPEG4 display =  
= 3. H.263 display =  
= 4. VC-1 display =  
= 5. 4-windows display =  
= 6. Camera preview & MFC encoding =  
= 7. MFC decoding & Camera preview =  
= 8. Camera input and JPEG encoding =  
= 9. JPEG decoding and display =  
= 10. H.264 decoding thru TVOUT =  
= 11. MFC decoding & Camera preview thru TV =  
= 12. Exit =  
=  
=====
```

Select number --> █

6.1 H264 ビデオハードウェアデコード機能テスト

「1」 入力して Enter キーを押す。

```
Select number --> 1
===== H.264 File Decodec Test =====
urbetter, v0.1 (2009-10-30)
#####<STREAMINFO> width=320 height=240.

[1. H.264 display]
Using IP      : MFC, Post processor, LCD
Input filename : wanted.264
Input vector size : VGA(640x480)
Display size   : WVGA(800x480)
Bitrate       : 971 Kbps
FPS           : 30
```



6.2 MPEG-4 ビデオハードウェアディコード機能テスト

「2」 入力して Enter キーを押す。

```
=====
Select number --> 2

[2. MPEG4 display]
Using IP      : MFC, Post processor, LCD
Input filename : shrek.m4v
Input vector size : QVGA(320x240)
Display size  : WUGA(800x480)
Bitrate      : 482 Kbps
FPS          : 24
█
```



6.3 H263 ビデオハードウェアディコード機能テスト

「3」 入力して Enter キーを押す。

```
=====
Select number --> 3

[3. H.263 display]
Using IP      : MFC, Post processor, LCD
Input filename : iron.263
Input vector size : QUGA(320x240)
Display size  : WUGA(800x480)
Bitrate      : 460 Kbps
FPS          : 30
█
```

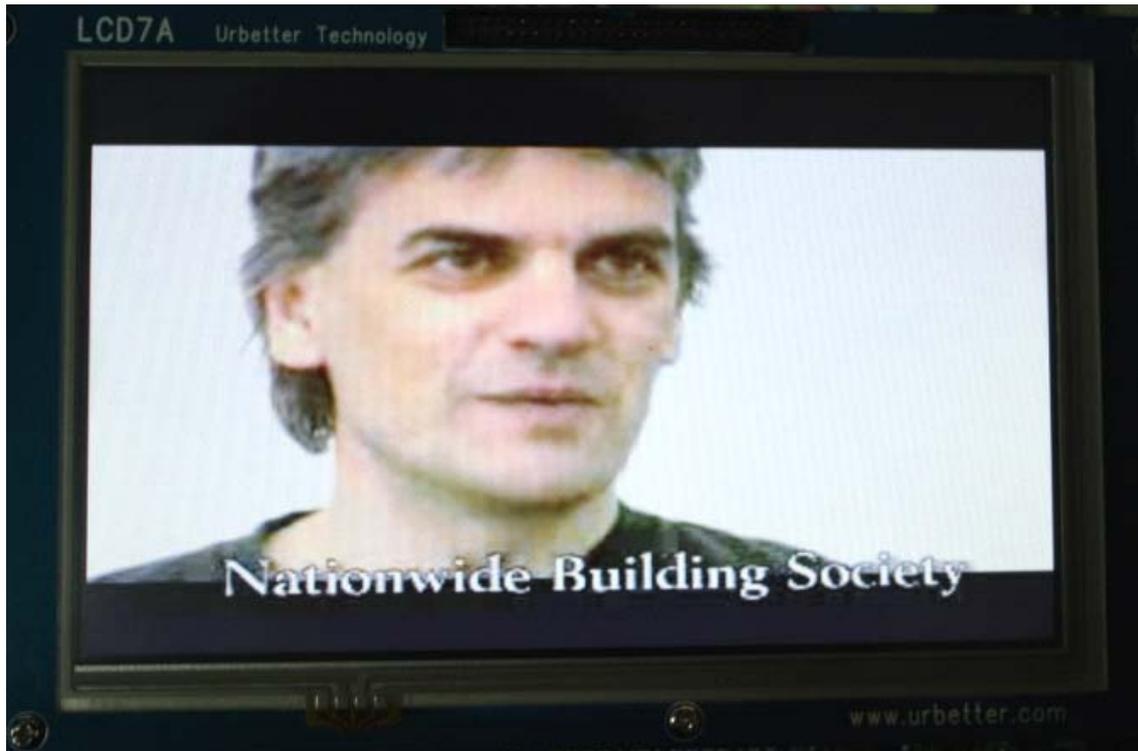


6.4 VC-1 ビデオハードウェアディコード機能テスト

「4」 入力して Enter キーを押す。

```
=====
Select number --> 4

[4. VC-1 display]
Using IP      : MFC, Post processor, LCD
Input filename : test2_0.rcv
Input vector size : QUGA(320x240)
Display size  : WUGA(800x480)
Bitrate      : 460 Kbps
FPS          : 30
█
```



6.5 複数チャネル同時ハードウェアディコード機能テスト

「5」入力してEnterキーを押す。

四つのウィンドーで同時にH264、MPEG4、H263、VC-1のハードウェアディコードする。



```
=====
Select number --> 5
===== 4 Windows multi-Format decode Test =====
urbetter, v0.1 (2009-10-30)

[4-windows display]
Using IP : MFC, Post processor, LCD
*****
*
* Frame buffer      : 0                * Frame buffer      : 1                *
* Codec            : H.264             * Codec            : MPEG4             *
* Input filename   : veggie.264        * Input filename   : shrek.m4v         *
* Input vector size : QUGA             * Input vector size : QUGA             *
* Display size     : 400x240          * Display size     : 400x240          *
* Bitrate          : 460 Kbps         * Bitrate          : 482 Kbps         *
* FPS              : 30               * FPS              : 24               *
*
*
*****
*
* Frame buffer      : 2                * Frame buffer      : 3                *
* Codec            : H.263             * Codec            : UC-1              *
* Input filename   : iron.263         * Input filename   : test2_0.rcv      *
* Input vector size : QUGA             * Input vector size : QUGA             *
* Display size     : 400x240          * Display size     : 400x240          *
* Bitrate          : 460 Kbps         * Bitrate          : 460 Kbps         *
* FPS              : 30               * FPS              : 30               *
*
*
*****
```



6.6 JPEG 画像のハードウェアディコード機能テスト

「9」入力して Enter キーを押す。



6.7 カメラプレビューと H264 ビデオハードウェアエンコード機能テスト

「6」入力して Enter キーを押す。

```
=====
Select number --> 6
==== Camera Preview & Encode to H264 Test ====
urbetter, v0.1 (2009-10-30)
U4L2 : Camera Input(U4L2_INPUT_TYPE_CAMERA )

[8. Camera preview & MFC encoding]
Using IP      : MFC, Post processor, LCD, Camera
Display size  : UGA(640x480)

e : Encoding
x : Exit
Select ==> █
```

「e」入力して Enter キーを押す。

エンコードが始まる。100 フレーム満たすと自動で停止する。エンコードした H264 ファイルはカレントフォルダに保存される。ファイル名は：[Cam_encoding_320x240-1.264](#)



```
=====
Select number --> 6
==== Camera Preview & Encode to H264 Test ====
urbetter, v0.1 (2009-10-30)
U4L2 : Camera Input(U4L2_INPUT_TYPE_CAMERA )

[8. Camera preview & MFC encoding]
Using IP      : MFC, Post processor, LCD, Camera
Display size  : UGA(640x480)

e : Encoding
x : Exit
Select ==> e
Name of encoded file : Cam_encoding_320x240-1.264
█
```



「x」を入力して、本テスト終了する。



```
=====
Select number --> 6
==== Camera Preview & Encode to H264 Test ====
  urbetter, v0.1 (2009-10-30)
U4L2 : Camera Input(U4L2_INPUT_TYPE_CAMERA )

[8. Camera preview & MFC encoding]
Using IP      : MFC, Post processor, LCD, Camera
Display size  : UGA(640x480)

e : Encoding
x : Exit
Select ==> e
Name of encoded file : Cam_encoding_320x240-2.264
[CAM]s3c_fimc_v4l2_streamoff return 0.
100 frames were encoded

Select ==> x
```

6.8 カメラプレビューと JPEG 画像ハードウェアエンコード機能テスト

「8」入力して Enter キーを押す。

```
=====
Select number --> 8
===== Camera Preview & JPEG Encode =====
  urbetter, v0.1 (2009-10-30)

VIDIOC_ENUMINPUT = 0
[11. Camera input & JPEG encoding]
Using IP      : Post processor, LCD, Camera, JPEG
Camera preview size : UGA(640x480)
Capture size   : UGA(640x480)

c : Capture
x : Exit
Select ==> █
```

「c」を選択する。1フレームをキャッチして JPEG 画像ファイルで保存される。
場所はカレントフォルダ、ファイル名は `Cam_capture_640x480-1.jpg`

```
=====
Select number --> 8
===== Camera Preview & JPEG Encode =====
urbetter, v0.1 (2009-10-30)

VIDIOC_ENUMINPUT = 0
[11. Camera input & JPEG encoding]
Using IP : Post processor, LCD, Camera, JPEG
Camera preview size : UGA(640x480)
Capture size : UGA(640x480)

c : Capture
x : Exit
Select ==> █
```



6.9 カメラプレビューと MFC ハードウェアディコード機能テスト

「7」入力して Enter キーを押す。

カメラからリアルタイムでキャッチし、ビデオディコードも同時に行って表示される。



不可能への挑戦

株式会社日昇テクノロジー

低価格、高品質が不可能？
日昇テクノロジーなら可能にする



——以上。