



Linux/Android/WinCE 対応マルチ・メディア ARM11
ボード Idea6410+LCD4.3
の Android 1.5 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

2010/2/5

[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	Ver0.1	新規作成	2010/2/5
2	Ver0.2	ビルド操作手順の追加	2010/02/16

- ※ 使用されたソースコードは <http://www.csun.co.jp/>からダウンロードできます。
- ※ この文書の情報は、事前の通知なく変更されることがあります。
- ※ (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。



第一章	Cross-compile のインストール	4
1.1	コンパイラをインストール	4
1.2	インストール成功したか確認する	4
第二章	u-boot のコンパイル	5
2.1	u-boot-movi.bin のコンパイル	5
2.2	u-boot-nand.bin のコンパイル	5
第三章	カーネルのコンパイル	6
3.1	zImage-fix-debug のコンパイル	6
3.2	zImage-fix-nand のコンパイル	6
第四章	cupcake のコンパイル	7
4.1	準備作業	7
4.2	コンパイル	7
第五章	イメージファイルを書き込む	9
5.1	書き込む条件	9
5.2	書き込む手順	9
付録	ネットワーク設定	19



第一章 Cross-compile のインストール

Version:Android-1.5_v0.10

Linux 環境: ubuntu-9.04

1.1 コンパイラをインストール

toolchains フォルダの arm-none-linux-gnueabi-arm-2008q3-72-for-linux.tar.bz2 を /usr/local/arm/フォルダにインストールする。

下記コマンドを実行する:

```
# tar jxvf arm-none-linux-gnueabi-arm-2008q3-72-for-linux.tar.bz2 -C /
```

※デフォルトのフォルダが /usr/local/arm/

1.2 インストール成功したか確認する

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

arm-none-linux-gnueabiが /usr/local/arm/ にインストールされている。



第二章 u-boot のコンパイル

u-bootは二つある：u-boot-movi.binとu-boot-nand.bin

u-boot-movi.bin：SDカードに書き込んで、SDカードからの起動を実現する。ソース
bootloader/u-boot-1.1.6-ut-s3c6410-moviをコンパイルして出来上がる。

u-boot-nand.bin：nand flashに書き込んで、nandからの起動を実現する。ソース
bootloader/u-boot-1.1.6-ut-s3c6410-nandをコンパイルして出来上がる。

2.1 u-boot-movi.bin のコンパイル

bootloader/u-boot-1.1.6-ut-s3c6410-moviフォルダで下記コマンドを実行する：

```
# make clean  
# make smdk6410_config  
# make  
# ./mkmovi
```

実行後、カレントフォルダにu-boot-movi.binファイルが生成される。

2.2 u-boot-nand.bin のコンパイル

bootloader/u-boot-1.1.6-ut-s3c6410-nandフォルダで下記コマンドを実行する：

```
# make clean  
# make smdk6410_config  
# make  
# ./mknand
```

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

第三章 カーネルのコンパイル

カーネルは二つある：zImage-fix-debugとzImage-fix-nand

zImage-fix-debug：SDカードに書き込んで、SDカードから起動して他のImageファイルを書き込むに使う。

zImage-fix-nand：nand flashに書き込んで、nandから起動した時のzImage。

■使用ソースコード：下記 URL からダウンロード

http://www.dragonwake.com/download/idea6410/en/urbetter-android-1.5_v1.0.tgz

3.1 zImage-fix-debug のコンパイル

kernel/urbetter-android-1.5-2.6.27/フォルダで下記コマンドを実行する：

```
# make clean
# cp urbetter-android-debug-430.config .config
 7" のLCDを使っている場合は cp urbetter-android-debug-700.config .config
# make menuconfig
# make
# ./fix-image
```

実行後生成されたzImage-fixをカレントフォルダにコピーして下記コマンドでファイル名をzImage-fix-debugに変更する。

```
# cp arch/arm/boot/zImage-fix ./zImage-fix-debug
```

3.2 zImage-fix-nand のコンパイル

下記コマンドを実行する：

```
# make clean
# cp urbetter-android-nand-430.config .config
 7" のLCDを使っている場合は cp urbetter-android-nand-700.config .config
# make menuconfig
# make
# ./fix-image
```

実行後生成された zImage-fix をカレントフォルダにコピーして下記コマンドでファイル名を zImage-fix-nand に変更する。

```
# cp arch/arm/boot/zImage-fix ./zImage-fix-nand
```

第四章 cupcake のコンパイル

4.1 準備作業

まず、cupcakeフォルダにあるソースコードをカレントフォルダに解凍する。

```
# sudo tar xvf urbetter-android-cupcake.tar.bz2
```

cupcakeフォルダに入る。

```
# cd android_cupcake
```

環境変数を設定する。

```
# export ANDROID_JAVA_HOME=/usr/lib/jvm/java-6-sun/
```

4.2 コンパイル

```
# make -j4 TARGET_PRODUCT=sec_smdk6410
```

少し時間がかかる。

コンパイル完了後、カレントフォルダ (android_cupcake) に「out」フォルダが生成される。

Out/target/product/smdk6410/フォルダの中身を見るとandroidのルートファイルシステム関連のすべてのファイルが生成されている。

```
└─1.5_v0.10\cupcake\android_cupcake\out\target\product\smdk6410
```

名称	大小	タイプ
data		フォルダ
obj		フォルダ
root		フォルダ
symbols		フォルダ
system		フォルダ
android-info.txt	1 KB	テキストファイル
clean_steps.mk	1 KB	Makefile
ramdisk.img	142 KB	IMG ファイル
system.img	54,228 KB	IMG ファイル
userdata.img	3 KB	IMG ファイル

◆ systemフォルダにあるすべてのファイルをroot/systemフォルダにコピーする。

ユーザーのニーズに応じて、独自で作成したアプリやWEBからダウンロードしたアプリをroot/system/appフォルダに追加して利用できる。弊社からも一部のアプリとゲームを (apkファイル) をtest file/apk/softwareとgamesフォルダで提供している。

ユーザーから変更あるいは追加があった場合は、全コンパイルが必要。その場合、「out」フォルダを丸ごと削除して再コンパイルする必要。



◆ ルートファイルシステムの圧縮

Out/target/product/smdk6410/root/で下記コマンドを実行する:

```
# sudo tar cvf ../urbetter-android-1.5-rootfs.tar *
```

カレントフォルダに生成されたurbetter-android-1.5-rootfs.tarファイルがルートファイルシステムになる。

第五章 イメージファイルを書き込む

5.1 書き込む条件

5-1-1.使用マニュアル: 上記の環境を解凍後、下記マニュアルを使用した。

`urbetter-android-1.5_v0.10¥doc¥android-1.5-UserManual_v0.10.pdf`

5-1-2.使用 SD カード: Panasonic1GB × 2

5-1-3.手順:

下記イメージを使用した。あるいは第五章前にコンパイルされたファイルを利用してください。

`¥urbetter-android-1.5_v0.10¥image¥image-480_272`

`u-boot-movi.bin`

`u-boot-nand.bin`

`zImage-fix-debug`

`zImage-fix-nand`

5.2 書き込む手順

5-2-1. Eboot 用の SD カードの作成

マニュアルに記載されている Eboot は使用環境に存在しないため、idea6410

付属の DVD より入手(下記ファイル)した。

`DVD¥Tools¥SDboot¥IROM_Fusing_Tool.exe`

`IROM_SD_EBOOT.nb0`

`SD boot.txt`

- ・WindowsPC 上で SD カードを Format する Format は FAT32 で実施。
- ・Image file to fuse に「IROM_SD_EBOOT.nb0」を指定。
- ・SD/MMC Driver に SD カードのドライブを指定。
- ・START キー押下。

5-2-2. Eboot でブロックの削除

- ・idea6410 に SD カードを挿入し、電源を投入。
- ・起動され、3 秒以内にスペースキーを押下。
- ・メニューより「A」を選択。
- ・「Erasing block」を確認。

5-2-3. boot 用の SD カード作成

Eboot 用とは別に用意した SD カードに対して実施した。

使用ツールはイメージファイルと同じフォルダのものを使用した。

`urbetter-android-1.5_v0.10¥image¥image-800_400¥moviNAND_Fusing_Tool.exe`

- WindowsPC 上で SD カードを Format する Format は FAT32 で実施。
- WindowsPC 上で moviNAND_Fusing_Tool.exe を実行。
- SD/MMC Driver に SD カードのドライブを指定。
- Bootloader の Image file に「u-boot-movi.bin」を指定。
- Specific Sector の Sector に 32 を指定。
- Specific Sector の Image file に「zImage-fix-debug」を指定。
- START キー押下。

5-2-4. urbetter-android-1.5-rootfs.tar のコピー

- Eboot と同じ SD にコピー。

5-2-5. idea6410 の設定:SD ブート:SW1 を「1111」に設定

5-2-6. Flash 内容をクリア

1. 【操作条件】

- ① シリアルケーブルを PC と接続(シリアルケーブルは USB アダプタにて変換)、Serial Port を Connect とする。
- ② SD ブート状態:1111
- ③ SD1 カードを ARM11 ボードに入れ、電源を投入、Linux を起動します。

2. 【ハイパーターミナル上に結果画面】

```
Done.  
chvt: can't open console  
modprobe: chdir(2.6.29): No such file or directory  
s3c-nand: ECC uncorrectable error detected  
modprobe: chdir(2.6.29): No such file or directory  
Spawning shell within the initramfs  
/bin/sh: can't access tty; job control turned off  
(initramfs):/# █
```



3. 【操作内容】

```
flash_eraseall /dev/mtd0  
flash_eraseall /dev/mtd1  
flash_eraseall /dev/mtd2  
flash_eraseall /dev/mtd3
```

```
(initramfs):/#  
(initramfs):/# flash_eraseall /dev/mtd0  
Erasing 128 Kibyte @ 60000 -- 75 % complete.  
(initramfs):/#  
(initramfs):/#  
(initramfs):/# flash_eraseall /dev/mtd1  
Erasing 128 Kibyte @ 4e0000 -- 97 % complete.  
(initramfs):/#  
(initramfs):/#  
(initramfs):/# flash_eraseall /dev/mtd2  
Erasing 128 Kibyte @ 2e0000 -- 95 % complete.  
(initramfs):/#  
(initramfs):/#  
(initramfs):/# flash_eraseall /dev/mtd3  
Erasing 128 Kibyte @ 4ac0000 -- 30 % complete.  
Skipping bad block at 0x04ae0000  
Erasing 128 Kibyte @ 4ee0000 -- 31 % complete.  
Skipping bad block at 0x04f00000  
Erasing 128 Kibyte @ cb40000 -- 82 % complete.  
Skipping bad block at 0x0cb60000  
Erasing 128 Kibyte @ f6e0000 -- 99 % complete.  
(initramfs):/#  
(initramfs):/#
```

5-2-7. ルートファイルシステムを書き込む準備

- 1 ARM11 ボードを再起動
- 2 再起動できたら、下記操作を行います。

```
ubiattach /dev/ubi_ctrl -m 3  
ubimkvol /dev/ubi0 -N rootfs -s 200MiB  
mount -t ubifs ubi0:rootfs /mnt
```

```
(initramfs):/#  
(initramfs):/#  
(initramfs):/# ubiattach /dev/ubi_ctrl -m 3  
ubiattach: error!: cannot attach mtd3  
error 17 (File exists)  
(initramfs):/#  
(initramfs):/#  
(initramfs):/# ubimkvol /dev/ubi0 -N rootfs -s 200MiB  
Volume ID 0, size 1626 LEBs (209793024 bytes, 200.1 MiB), LEB size 129024 byt  
(initramfs):/#  
(initramfs):/#  
(initramfs):/# mount -t ubifs ubi0:rootfs /mnt  
UBIFS: default file-system created  
UBIFS: mounted UBI device 0, volume 0, name "rootfs"  
UBIFS: file system size: 208373760 bytes (203490 KiB, 198 MiB, 1615 LEBs)  
UBIFS: journal size: 10450944 bytes (10206 KiB, 9 MiB, 81 LEBs)  
UBIFS: media format: 4 (latest is 4)  
UBIFS: default compressor: lzo  
UBIFS: reserved for root: 5182151 bytes (5060 KiB)  
(initramfs):/#  
(initramfs):/#
```

5-2-8. ルートファイルシステムを書き込む

- 1 再起動せずに電源を入れたまま、SD1 を取り出し、SD2 を入れてください。

* SD2(4で「urbetter-android-1.5-rootfs.tar」をコピー済)

2 操作内容

```
mount -t vfat /dev/mmcblk0p1 /home  
tar xvf /home/urbetter-android-1.5-rootfs.tar -C /mnt  
umount /mnt  
umount /home
```



5-2-9. ブートを Nand Flash に書き込む

1. 【操作条件】

- ① SD ブート状態:1111
- ②SD1 カードを ARM11 ボードに入れてください。
- ③付属 USB ケーブルで PC と ARM11 を接続し、PC と接続したら、USB ドライバをインストールしてください。USB Driver 場所:¥tools¥usb-driver(secusb2.sys、secusb2.inf)

U-Boot を起動する(「Hit any key to stop autoboot」が出来たら、Enter キーを押す)

```
U-Boot 1.1.6 (Aug  3 2009 - 09:41:16) for SMDK6410
*****
**   UT-S3C6410/idea6410 SD boot v0.19
**   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

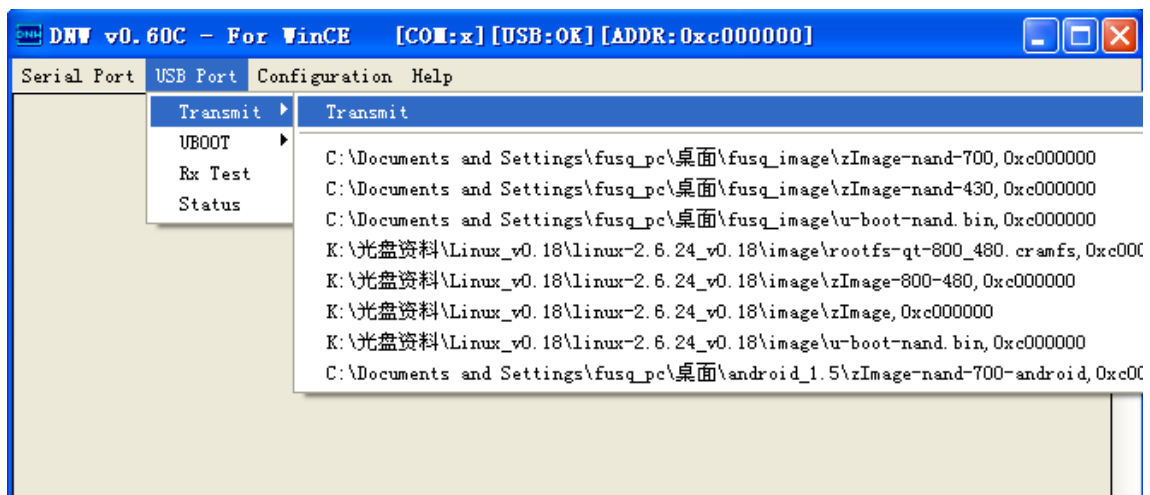
In:       serial
Out:      serial
Err:      serial
Net:      Not Found CS8900@0x18800300
Hit any key to stop autoboot:  0
SMDK6410 #
SMDK6410 #
SMDK6410 #
```

2. コマンド「dnw c0008000」を発行(最後 Enter キーを入力)

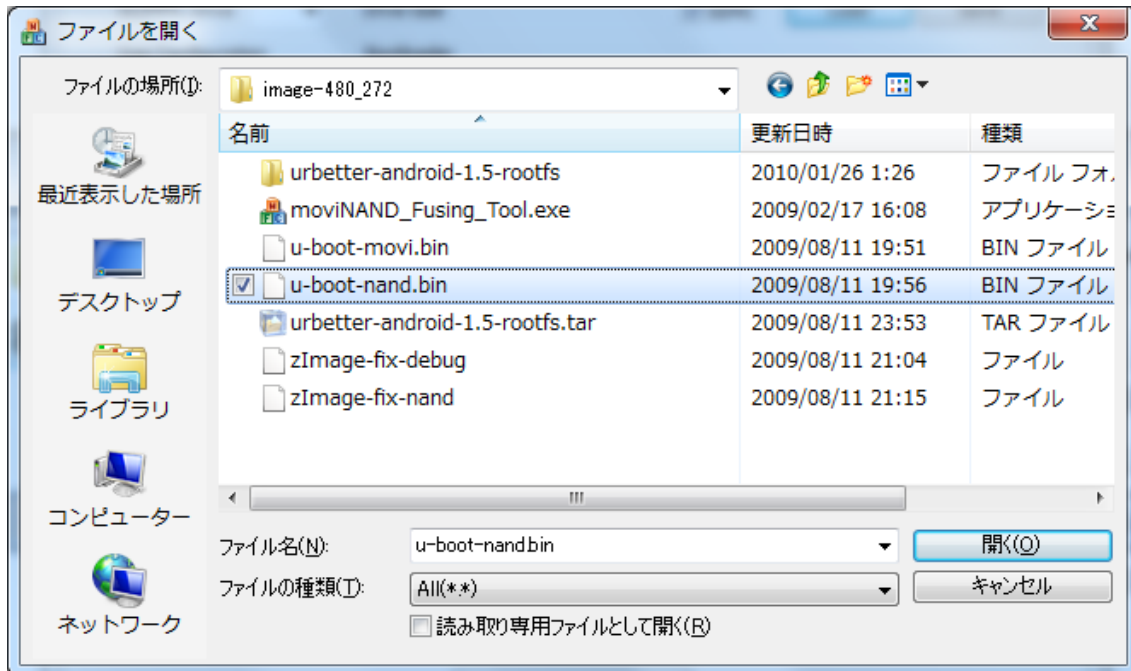
```
SMDK6410 #  
SMDK6410 #  
SMDK6410 # dnw c0008000  
OTG cable Connected!  
Now, Waiting for DNW to transmit data
```

3. dnw を起動

USB Port → Transmit → Transmit



4. u-boot-nand.binを開く



```
SMDK6410 #  
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 #  
SMDK6410 #
```

5. ubootをnand flash 0に書き込む

コマンド「nand write c0008000 0 100000」を発行(Enterを入力)

```
SMDK6410 #  
SMDK6410 #  
SMDK6410 # nand write c0008000 0 100000  
NAND write: device 0 offset 0x0, size 0x100000  
1048576 bytes written: OK  
SMDK6410 #
```

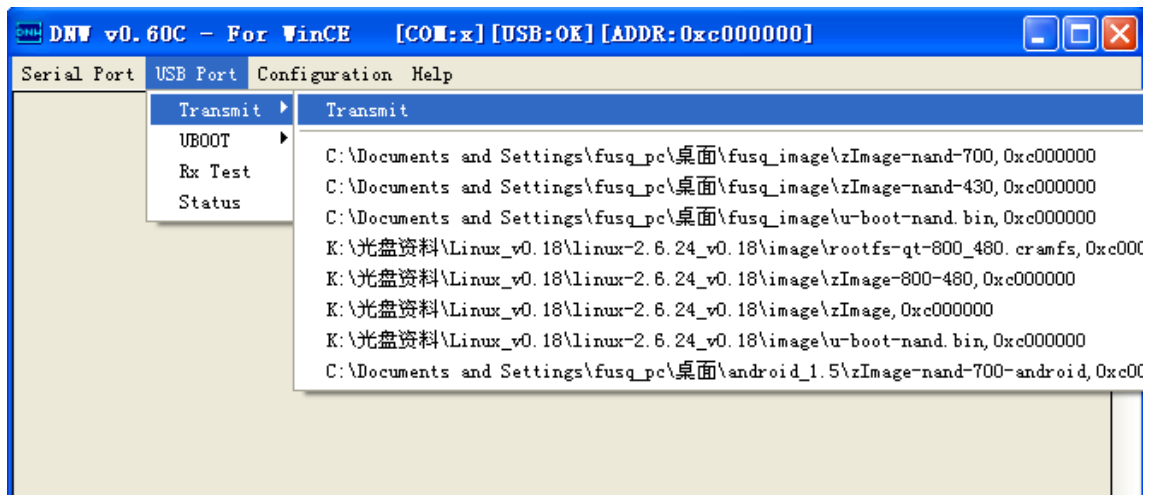
5-2-10 イメージファイルを書き込む(9-4の画面で続きまして)

1. コマンド「dnw c0008000」を発行(最後Enterキーを入力)

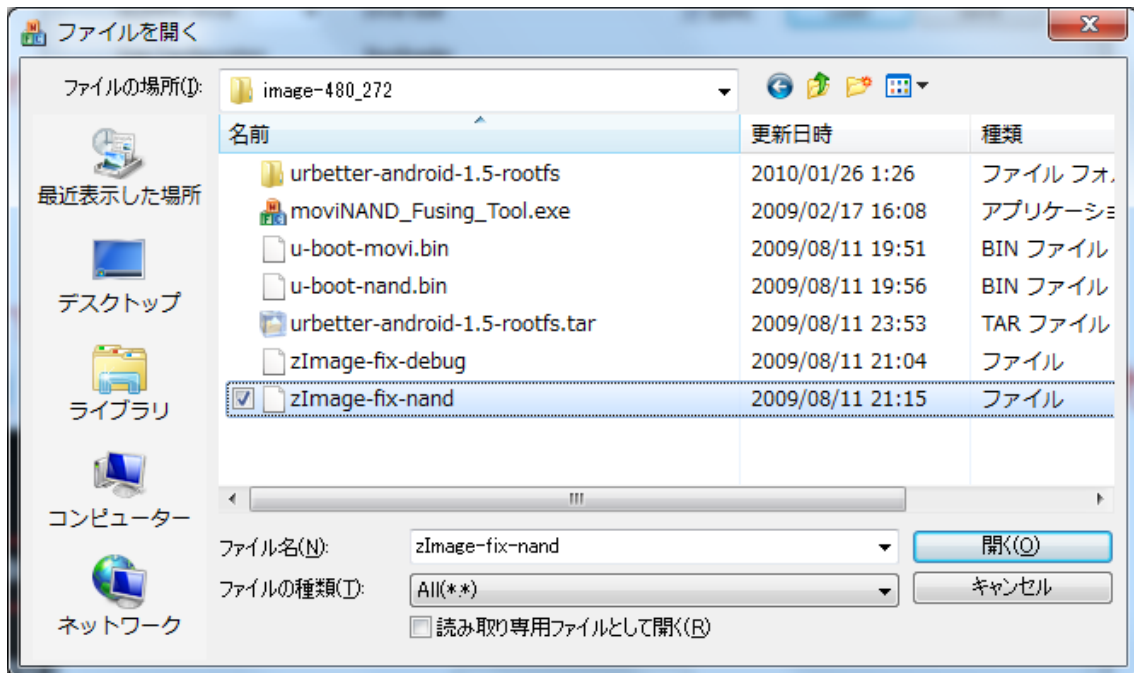
```
SMDK6410 #  
SMDK6410 # dnw c0008000  
Insert a OTG cable into the connector!  
OTG cable Connected!  
Now, Waiting for DNW to transmit data  
Download Done!! Download Address: 0xc0008000, Download Filesize:0x20  
Checksum is being calculated...  
Checksum O.K.  
SMDK6410 #
```

2. dnw を起動

USB Port → Transmit → Transmit



3. zImage-fix-nandを開く



4. zImage-fix-nandを「nand flash 0x600000」に書き込む

コマンド「nand write c0008000 600000 300000」を発行 (Enterを入力)

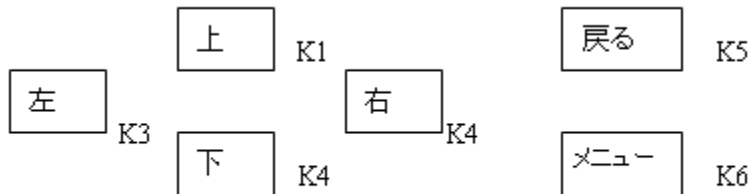
```
SHDK6410 #  
SHDK6410 #  
SHDK6410 # nand write c0008000 600000 300000  
  
NAND write: device 0 offset 0x600000, size 0x300000  
3145728 bytes written: OK  
SHDK6410 #
```

5-2-11 Nand Flash起動モードに設定(SW1:1100)、ARM11を再起動

Androidを初めて起動する場合、時間がかかります。起動後、1分間でスリープ状態に入ります。そのため、起動後、設定は変更が必要です。

Androidのメイン画面で、「K6」(メニューボタン)を押し、“Settings”を選択し、“Sound & Display”-》“Screen timeout”-》“Never timeout”

* ARM11ユーザーボタンはAndroidが起動の際に操作イメージを下記の図に参照



付録: ネットワーク設定

1. 起動前にLANケーブルを付ける場合、IPは自動取得されますので、特に設定必要がありません。
2. 起動後LANケーブルを付ける場合、手動で設定が必要です。
 - ① LANケーブルを接続する時、

```
#  
#  
# eth0: link up, 100Mbps, full-duplex, lpa 0x45E1  
#
```

- ② ネットワークの状況を調べ、コマンド「netcfg」

```
#  
# netcfg  
lo      UP      127.0.0.1      255.0.0.0      0x00000049  
eth0    UP      0.0.0.0        0.0.0.0        0x00001043  
tunl0   DOWN   0.0.0.0        0.0.0.0        0x00000080  
gre0    DOWN   0.0.0.0        0.0.0.0        0x00000080  
#
```

上記の結果により、IPは取れてない状況です。

- ③ コマンド「netcfg eth0 up」を発行
- ④ コマンド「netcfg eth0 dhcp」を発行
- ⑤ コマンド「netcfg」をもう一回発行

```
#  
# netcfg  
lo      UP      127.0.0.1      255.0.0.0      0x00000049  
eth0    UP      192.168.1.113  255.255.255.0  0x00001043  
tunl0   DOWN   0.0.0.0        0.0.0.0        0x00000080  
gre0    DOWN   0.0.0.0        0.0.0.0        0x00000080  
#
```

IPは割り当てられたことを明らかにした、ネットワークは接続できます。

最後:Android起動後の様子

