



**立宇泰电子**

Liyutai Elec. CO., Ltd.

**专注于做最好的嵌入式计算机系统供应商**

# ARMSYS6410 开发套件 Linux2.6.28

## **使用说明**

Rev 1.0

2009 年 10 月 27 日

**杭州立宇泰电子有限公司**

HangZhou LiYuTai Elec.Co.,Ltd

## 目录

1. 内容综述 .....	7
2. 需要准备的硬件设备及主机环境 .....	8
3. 安装 LINUX 到 ARMSYS6410 .....	9
3.1 安装 U-BOOT .....	9
3.1.1 将 u-boot-sd.bin 文件写入到 SD 卡 .....	9
3.1.2 设置开发板从 SD 卡启动 .....	10
3.1.3 将 u-boot-nand.bin 写入到 nandflash .....	12
3.1.4 设置开发板从 Nandflash 启动 .....	15
3.1.5 设置 u-boot 工作参数 .....	16
3.2 安装 KENERL .....	17
3.3 安装根文件系统 .....	23
3.3.1 根文件系统目录 .....	23
3.3.2 NFS 方式挂载根文件系统 .....	23
3.3.3 安装 Yaffs2 文件系统 .....	25
3.3.4 安装 cramfs 文件 .....	28
4. 安装交叉编译器 .....	28
5. U-BOOT 源码包的编译 .....	29
6. LINUX 内核源码包的编译 .....	29
6.1 配置内核 .....	29
6.2 编译内核 .....	32
7. 设备驱动与测试 .....	32
7.1 视频硬件解码测试 (H.263/H.264, MPEG4, VC-1 格式) .....	33
7.2 多种视频格式同时解码 (4 窗口) 测试 .....	37
7.3 JPEG 图像解码测试 .....	38
7.4 摄像头预览和 H.264 编码测试 .....	38
7.5 摄像头预览和 JPEG 编码测试 .....	39
7.7 摄像头预览和 MFC 解码 .....	41
7.7 视频解码 TVOUT 输出测试 .....	42
7.8 摄像头预览和 TVOUT 输出测试 .....	43
7.5 触摸屏设备驱动测试与 TSLIB 校准 .....	44
7.6 AC97 音频设备驱动测试 .....	45
7.7 键盘驱动测试 .....	45
7.1 以太网口驱动测试 (PING) .....	46
7.2 TTYSA0~3 串口驱动测试 .....	46
7.3 两路 SD/MMC 卡驱动测试 .....	48
7.4 主 USB 接口设备驱动测试 .....	48
7.4.1 USB 鼠标/键盘测试 .....	48
7.4.2 U 盘测试 .....	49

---

附录 A: 网络服务 .....	49
A.1 IFCONFIG 命令 .....	49
A.2 PING 命令 .....	50
A.3 TELNET 服务器 INETD.....	50
A.4 远程文件传送 FTP.....	51

## 版本记录

版本号	描述	日期
V1.0	最初版	2009-10-29

感谢使用立字泰电子研制的 ARMSYS6410 系列产品！

ARMSYS6410 采用了 Linux-2.6.28 作为标准版的 linux 内核，其中集成了丰富的驱动资源，充分展现 S3C6410 的各项特性，包括硬件编解码、2D/3D 加速、显示协处理、TVOUT 输出、视频采集和编码、4 路串口、2 路 SD/MMC 接口、1 路 10/100M 以太网接口、1 路 USB host 接口等等，使 ARMSYS6410 成为目前 linux 配置最为强劲和最完整的开发板之一。ARMSYS6410 支持 Yaffs2 文件系统，根文件系统提供了完整的常用命令和多种网络服务，结构清晰简明，使用灵活方便。图形界面采用 Qt/ia2.2.0 版本，集成丰富的应用程序，可以支持触摸屏输入和鼠标输入。同时提供丰富的测试程序，可以对 ARMSYS6410 的各项特色功能进行充分测试和展现。

详细配置，参考下表：

ARMSYS6410_Linux2.6.28 相关资源		
类别	功能特性	描述
引导装载 (u-boot1.1.6)	Bootloader	初始化并装载启动 linux 内核
	Download Binary by DNW	支持与 DNW 软件配合 USB 下载
	Set Kernel Booting Param.	支持设置内核启动参数
	Update Flash	支持映像文件的固化
	Nand Boot, SD Boot	提供从 Nandflash 启动和 SD 卡启动两个版本
内核及设备驱动	Linux 2.6.28	采用 2.6.28 版本内核
	System Clock: 533MHz	系统主频: 533MHz
	4 Port Serial device driver	<b>4 路串口驱动(ttySAC0~3)</b>
	256M NandFlash device driver	256M NandFlash 驱动
	10/100Base-T Ethernet device (DM9000) driver	<b>1 路 10/100M 以太网控制器驱动 (DM9000)</b>
	2×5 Keypad	<b>2 ×5 用户按键驱动</b>
	RTC (Real Time Clock) Driver	实时时钟驱动
	USB Host driver	1 路 USB Host 驱动 (USB1.1) 支持优盘、鼠标等
	2 SD/MMC Memory Card Driver	<b>2 路 SD/MMC 卡座驱动，支持热插拔</b>
	AC97 Audio driver (WM9714)	音频设备 (WM9714)，支持耳机和扬声器放音，音量可调
	Framebuffer	液晶(支持立字泰电子 TFT 型液晶屏系列) 驱动
	TouchScreen	支持 4 线触摸屏
	MFC(Multi-Format Video Codec)	支持 H.263/264, MPEG4, VC-1 编码和解码
	JPEG Codec	支持 JPEG 图片的编解码
	Post Processor	支持
	TV OUT	支持
	2D/3D	支持
	Camera (OV9650)	支持预览和编码
	Watchdog	支持
	support	

	/CRAMFS/EXT2/FAT/NFS/Initramfs /JFFS2/YAFFS2 filesystem	/CRAMFS/EXT2/FAT/NFS/InitRAMFS /JFFS2/YAFFS2 等文件系统 根文件系统采用 NFS、YAFFS2、CRAMFS
	NFS Client	NFS 网络文件系统客户端
编译器	cross-4.2.2-eabi.tar.bz2	
网络服务	TCP/IP	完整的 TCP/IP 协议
	Telnet Server	Telnet 服务器
	FTP Client	FTP 客户端服务
	Remote login (telnet)	Telnet 远程登录
	Ifconfig, inetd	提供 ifconfig,inetd 等网络配置和服务程序
	Network ping	使用 ping 检查系统
基本命令	Busybox	Linux 常用命令
触摸屏校准工具	Tslib	触摸屏校准工具
图形界面	Qtvia2.2.0	图形界面系统。Trolltech 开发的用于嵌入式 Linux 的图形用户界面 PDA 版系统，支持浏览器，Medialplay 播放器，文件管理器等。
测试程序	MFC test (H.263/H.264, MPEG4, VC-1) Multi-Format media codec (4-Windows) JPEG codec Camera preview and H.264 encode Camera preview and JPEG encode Camera preview and MFC decode MFC and TVOUT Camera preview and TVOUT Touchscreen test and tslib calibrate AC97 audio device test Keyboard test Ethernet interface test TtySAC0~3(4 port serial) test 2 channel SD/MMC socket test USB host driver test	视频硬件解码测试 (H.263/H.264, MPEG4, VC-1 格式) 多种视频格式同时解码 (4 窗口) 测试 JPEG 图像解码测试 摄像头预览和 H.264 编码测试 摄像头预览和 JPEG 编码测试 摄像头预览和 MFC 解码 视频解码 TVOUT 输出测试 摄像头预览和 TVOUT 输出测试 触摸屏设备驱动测试与 TSLIB 校准 AC97 音频设备驱动测试 键盘驱动测试 以太网口驱动测试 (PING) TTYsAC0~3 串口驱动测试 两路 SD/MMC 卡驱动测试 主 USB 接口设备驱动测试



# 1. 内容综述

本文将对以下内容进行详细说明，您会从中了解如何建立基于 ARMSYS6410 的嵌入式 Linux 开发环境及如何在这个环境下进行开发工作。

- (1) 需要准备的硬件设备及主机环境；
- (2) 如何安装 u-boot 到目标板；
- (3) 如何安装 kernel 映像文件到目标板；
- (4) 如何采用 NFS 挂载文件系统，安装 YAFFS2 文件系统和安装 CRamfs 文件系统；
- (5) 如何安装交叉编译工具；
- (6) 如何编译 u-boot 和 kernel，产生映像文件；
- (7) 测试 ARMSYS6410 的各个设备驱动；
- (8) 附录 A: ARMSYS6410 支持的网络服务。

关于 ARMSYS6410 其它相关内容，请参考文档《ARMSYS6410 开发套件硬件用户手册》、《ARMSYS6410 开发套件 WinCE6.0 用户手册》。

## 2. 需要准备的硬件设备及主机环境

### 硬件设备:

- ARMSYS6410 开发板;
  - DC+9V 直流电源, 为 ARMSYS6410 开发板供电, 插头为内正外负;
  - DB9 串口电缆线 (2-3 交叉), 两端分别连接 PC 机 COM 口和开发板的 UART0 口 (Console 接口);
  - USB 电缆(一端 Type-A 插头, 一端 miniUSB 插头), 两端分别连接 PC 机的 USB 口和开发板的 Mini-USB 口;
  - 交叉网线电缆, 两端分别连接 PC 机的 RJ45 网口和 ARMSYS6410 的 RJ45 网口;
- 可选设备有: SD 卡读卡器和 SD 卡、LCD 模块、摄像头小板、AV 视频连接线, 耳机/MIC、鼠标、U 盘等。

### 主机环境:

#### **Windows XP+VMware6.0+ubuntu 9.04**

本手册将以此为例进行讲解, 用户可以根据自身需要安装其他方式的主机环境。

## 3. 安装 Linux 到 ARMSYS6410

将 linux 系统安装到 ARMSYS6410 的 nandflash 中，需要用到的文件有：

- 从 nandflash 启动引导程序：u-boot-nand.bin；
- 内核：zImage；
- 根文件系统：armsys6410.cramfs、root\_armsys6410\_v1.0.tgz；

如果板上 nandflash 为空或原来的 u-boot 损坏，那么要借助从 SD 卡启动来恢复系统，这时除了以上映像文件，还要使用到：

- 从 SD 卡启动引导程序：u-boot-sd.bin。

预装 linux 的 ARMSYS6410 开发板在出厂时都已经安装好了完整的 Linux 系统。您需要更新哪个部分，就选择下面哪个小节来阅读。

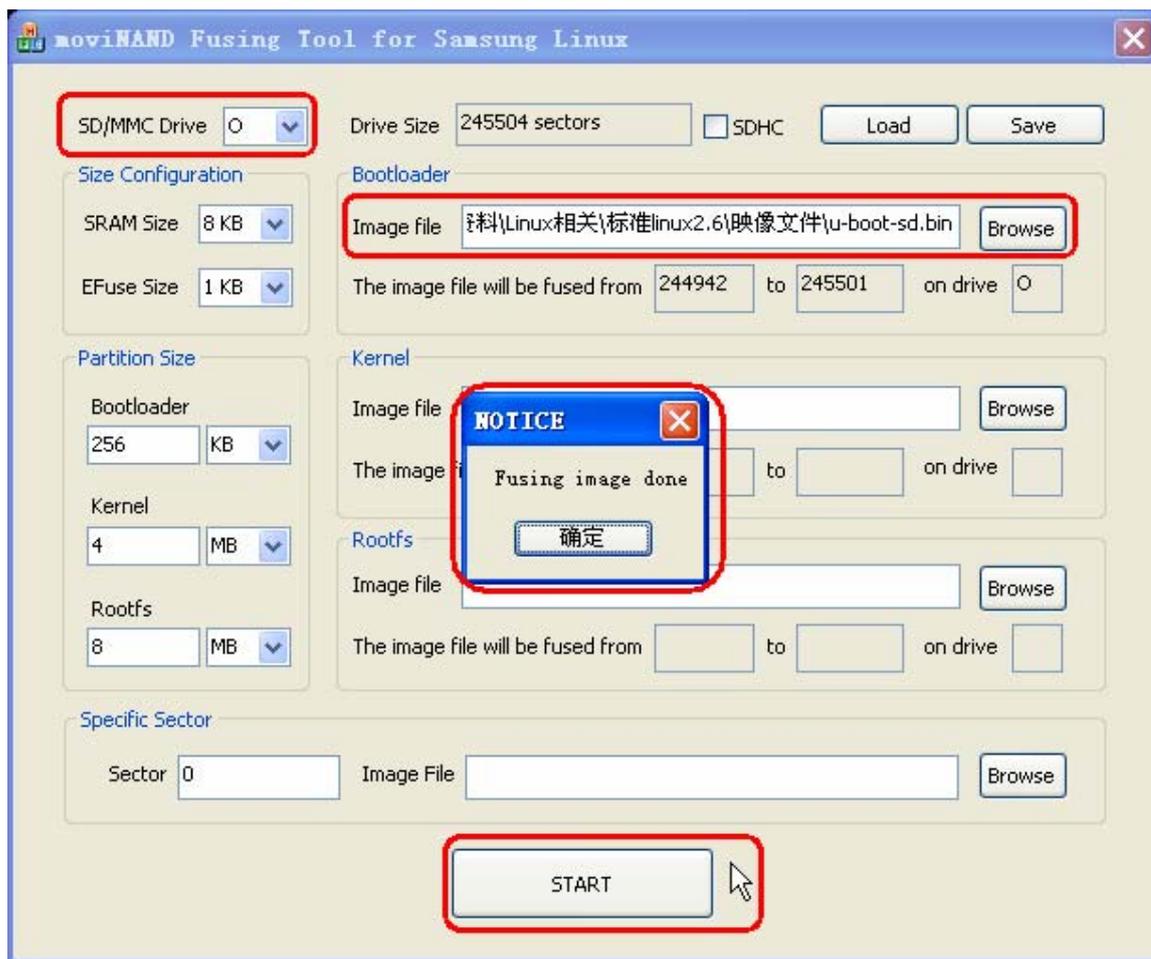
### 3. 1 安装 u-boot

下面介绍的是空板固化 u-boot 的方法：

#### 3.1.1 将 u-boot-sd.bin 文件写入到 SD 卡

(1) 将 SD 卡插入 SD 卡读卡器，并连接到 PC 机；

(2) 在 WindowsXP 环境下，打开“ARMSYS6410-B 基础配置光盘资料\SD 卡烧录工具\Linux 使用\SD\_Fusing\_Tool.exe”软件，如下图所示：



- (3) 在“Image file”框中，点击“Browse”，选择“ARMSYS6410-B 基础配置光盘资料\Linux 相关\ linux2.6 标准版\映像文件\ u-boot-sd.bin”文件；
- (4) 在“SD/MMC Driver”框中，选择在您计算机上 SD 卡读卡器映射的磁盘盘符。
- (5) 点击“START”，弹出“Fusing image done”信息，烧录完成。

### 3.1.2 设置开发板从 SD 卡启动

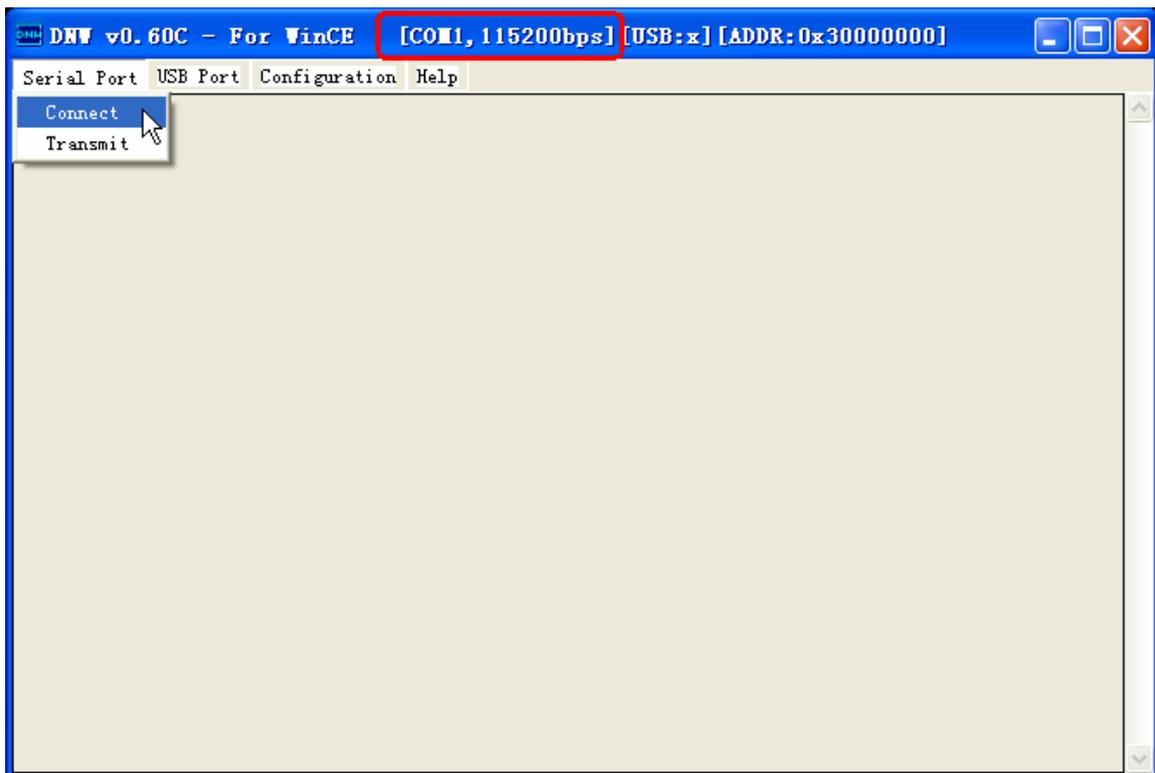
- (1) 将 SD 卡从 SD 卡读卡器拔出，插入到 ARMSYS6410 开发板的 SD1 插座中；
- (2) 设置 SW3，让开发板从 SD 卡启动：

拨码开关SW3 引脚号	Pin 8	Pin 7	Pin 6	Pin 5	Pin 4	Pin 3	Pin 2	Pin 1
引脚定义	OM1	OM2	OM3	OM4	SELNAND	GPN15	GPN14	GPN13
Nandflash启动	0	0	1	1	0	X	X	X
<b>SD卡启动</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>X</b>	<b>1</b>	<b>1</b>	<b>1</b>
Norflash启动	0	1	0	1	X	X	X	X

拨码开关拨到“ON”位置上表示“1”，拨到“OFF”位置上表示“0”

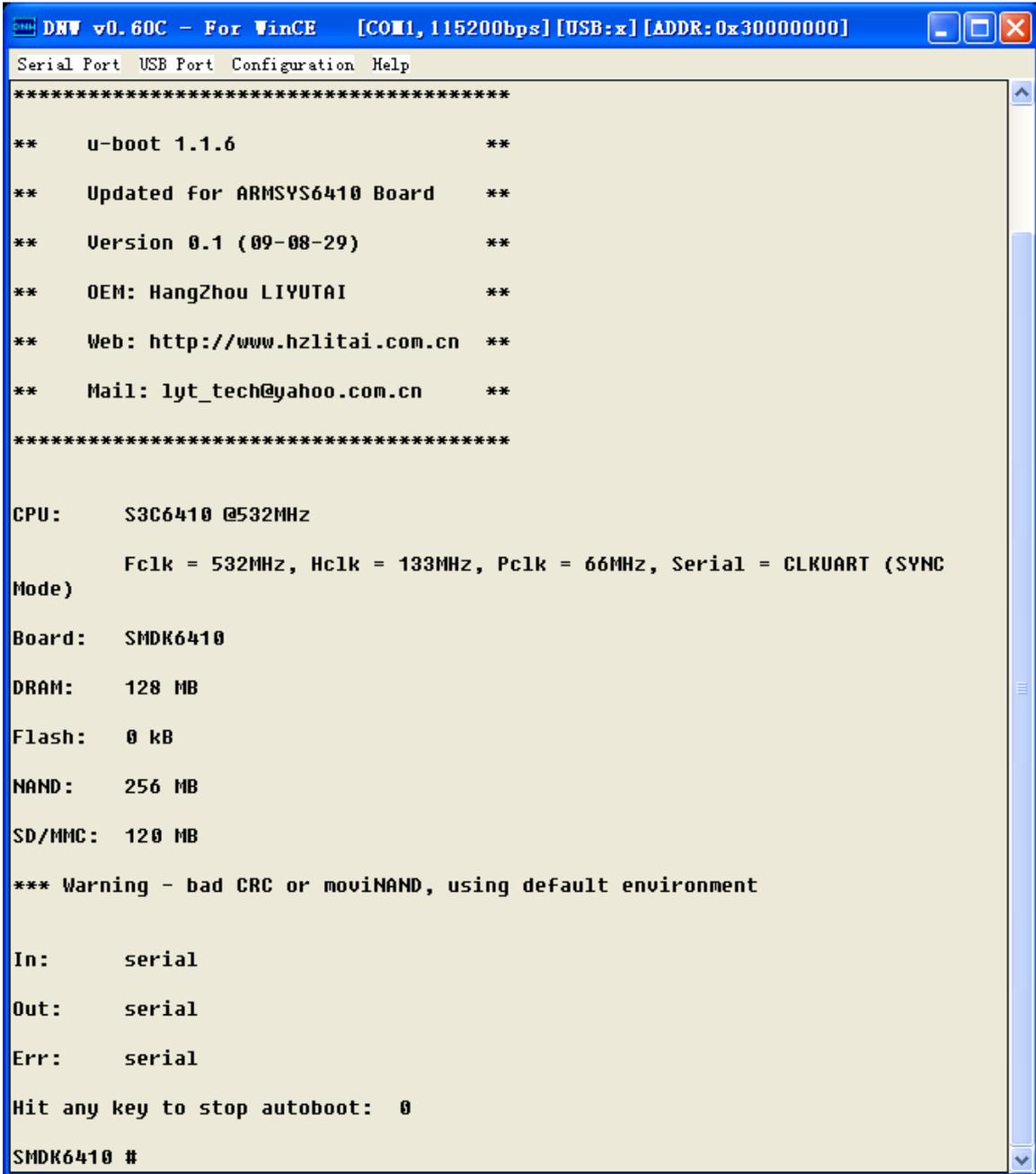
- (3) 连接好开发板的串口线，USB 线，电源线，打开 dnw0.60C 软件（“ARMSYS6410-B 杭州立字泰电子有限公司 第 10 页 共 51 页 电话：0571-56763523 56763526 网址：[www.hzlitai.com.cn](http://www.hzlitai.com.cn) [www.armssystem.com.cn](http://www.armssystem.com.cn) 传真：0571-56763523-808

基础配置光盘资料\DNW 和 USB 驱动\dnw\_v0.60c.exe”):



点击菜单“Serial Port→Connect”，显示连上串口。连接成功后，标题栏会显示 [COM1,115200bps]

(4) 打开开发板电源开关，在 DNW 窗体上看到打印信息时，按下空格键进入 u-boot 命令行：



```
DNW v0.60C - For WinCE [COM1, 115200bps] [USB:x] [ADDR: 0x30000000]
Serial Port  USB Port  Configuration  Help
*****
**   u-boot 1.1.6           **
**   Updated for ARMSYS6410 Board   **
**   Version 0.1 (09-08-29)         **
**   OEM: Hangzhou LIYUTAI         **
**   Web: http://www.hzlitai.com.cn **
**   Mail: lyt_tech@yahoo.com.cn   **
*****

CPU:      S3C6410 @532MHz

          Fclk = 532MHz, Hclk = 133MHz, Pclk = 66MHz, Serial = CLKUART (SYNC
Mode)

Board:    SMDK6410

DRAM:     128 MB

Flash:    0 kB

NAND:     256 MB

SD/MMC:   120 MB

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

In:       serial

Out:      serial

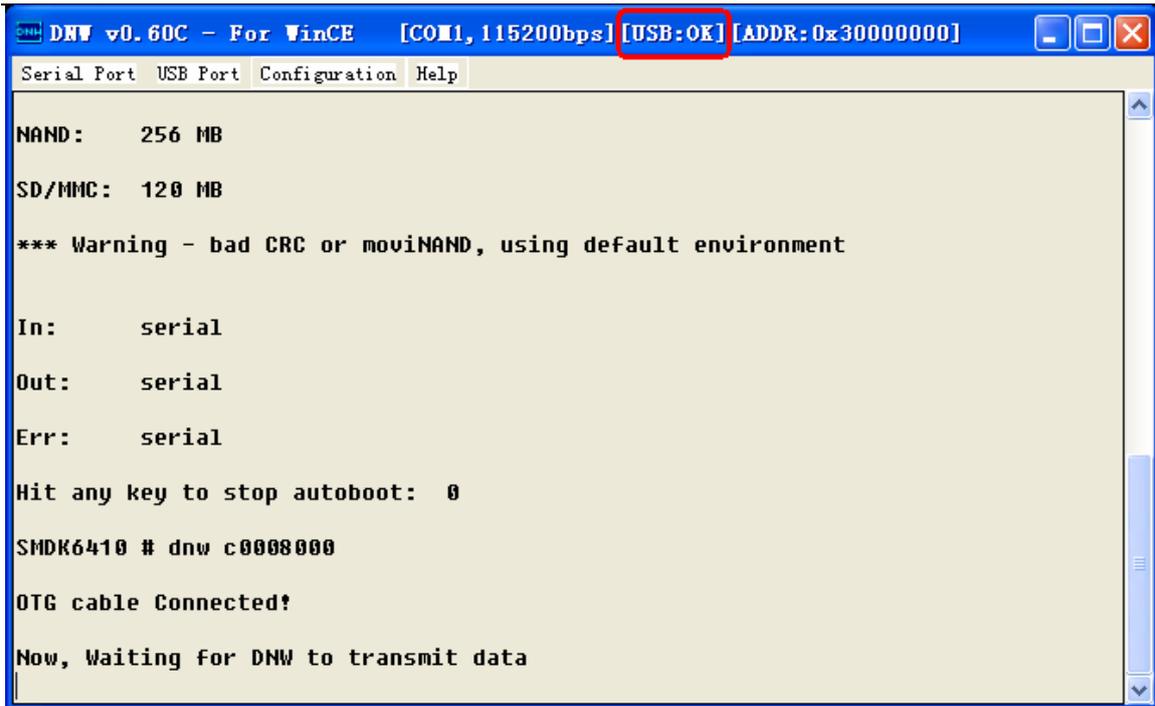
Err:      serial

Hit any key to stop autoboot:  0

SMDK6410 #
```

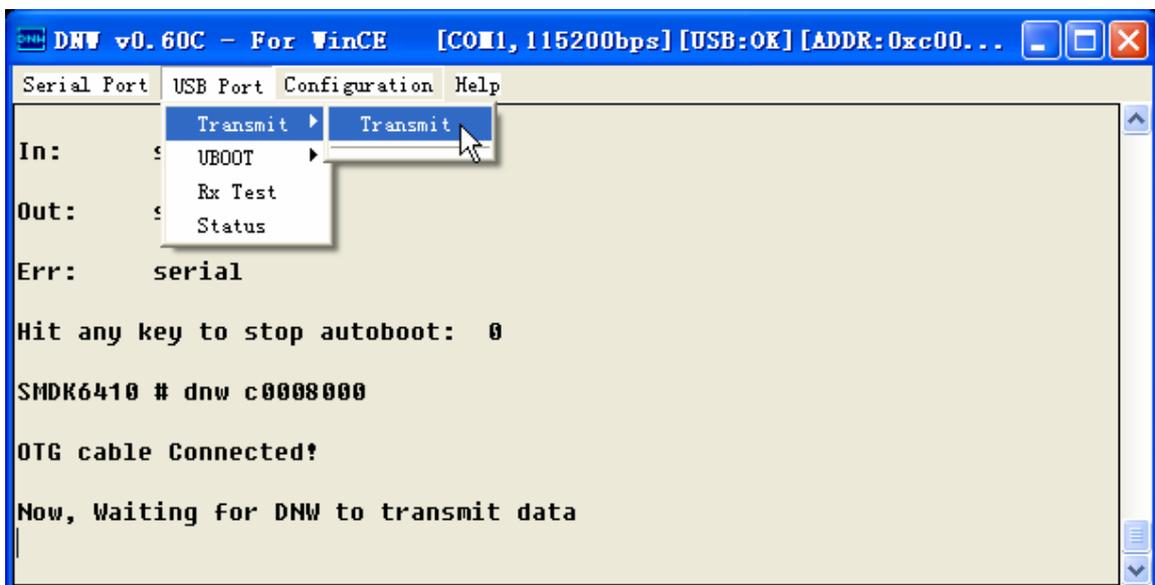
### 3.1.3 将 u-boot-nand.bin 写入到 nandflash

(1) 进入 u-boot 命令行后，输入 dnw c0008000，回车。

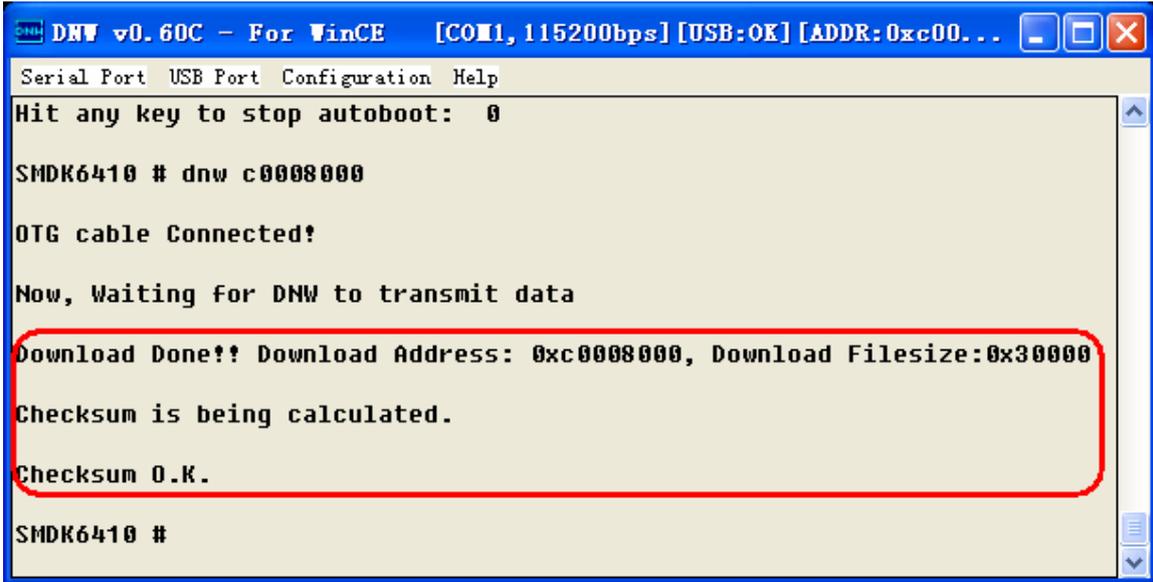


此时如果连好了开发板和 PC 机之间的 USB 线，dnw 的标题栏里面就会显示 “[USB:OK]”。

(2) 点击菜单 “USB Port→Transmit→Transmit”，选中 “ARMSYS6410-B 基础配置光盘资料\Linux 相关\linux2.6 标准版\映像文件\u-boot-nand.bin” 文件，下载映像文件到内存。



下载成功的信息：

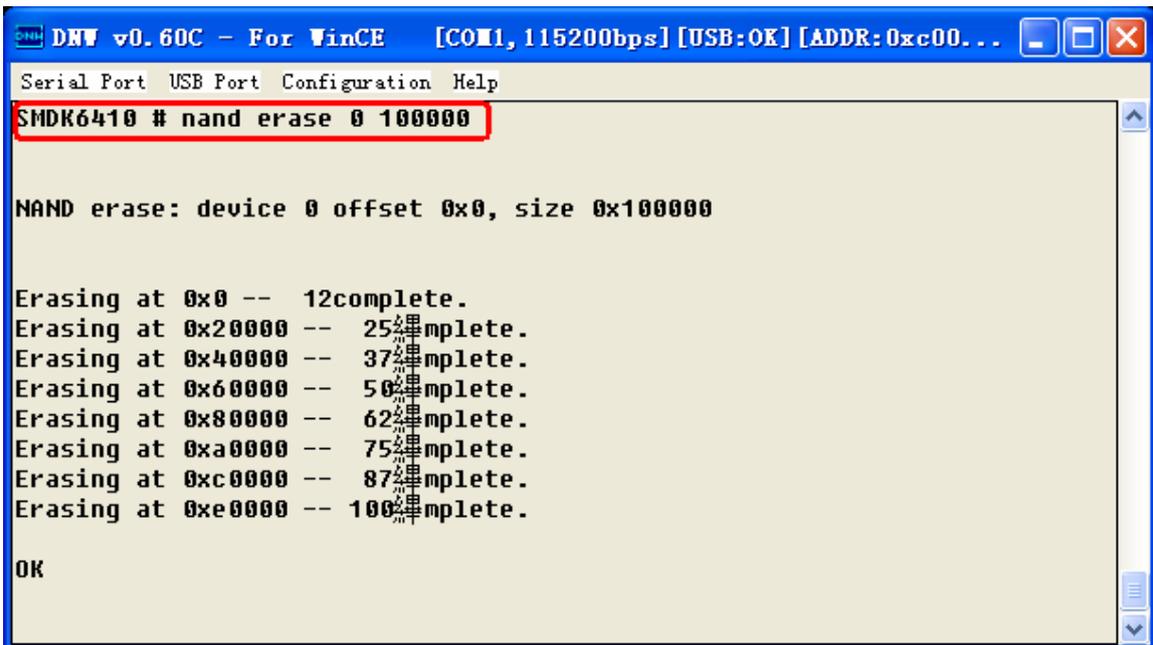


```
DNW v0.60C - For WinCE [COM1, 115200bps] [USB:OK] [ADDR:0xc00...
Serial Port USB Port Configuration Help
Hit any key to stop autoboot: 0
SMDK6410 # dnw c0008000
OTG cable Connected!
Now, Waiting for DNW to transmit data
Download Done!! Download Address: 0xc0008000, Download Filesize:0x30000
Checksum is being calculated.
Checksum O.K.
SMDK6410 #
```

(3) 执行擦除命令，命令格式：

nand erase <start block number> <image size>

输入：nand erase 0 100000，擦除 nandflash 从 0 开始的 1MB 大小的区域。

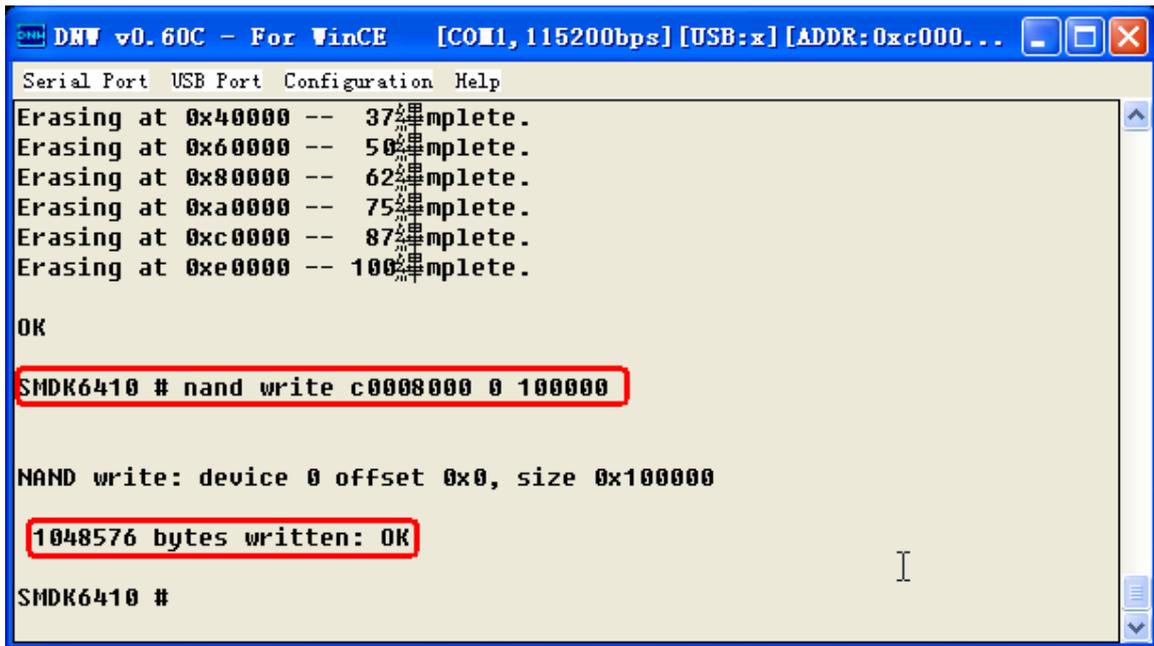


```
DNW v0.60C - For WinCE [COM1, 115200bps] [USB:OK] [ADDR:0xc00...
Serial Port USB Port Configuration Help
SMDK6410 # nand erase 0 100000
NAND erase: device 0 offset 0x0, size 0x100000
Erasing at 0x0 -- 12complete.
Erasing at 0x20000 -- 25complete.
Erasing at 0x40000 -- 37complete.
Erasing at 0x60000 -- 50complete.
Erasing at 0x80000 -- 62complete.
Erasing at 0xa0000 -- 75complete.
Erasing at 0xc0000 -- 87complete.
Erasing at 0xe0000 -- 100complete.
OK
```

(4) 执行写入命令，命令格式：

nand write <temporary address> <start block number> <image size>

输入：nand write c0008000 0 100000，将 u-boot 写入 nandflash。



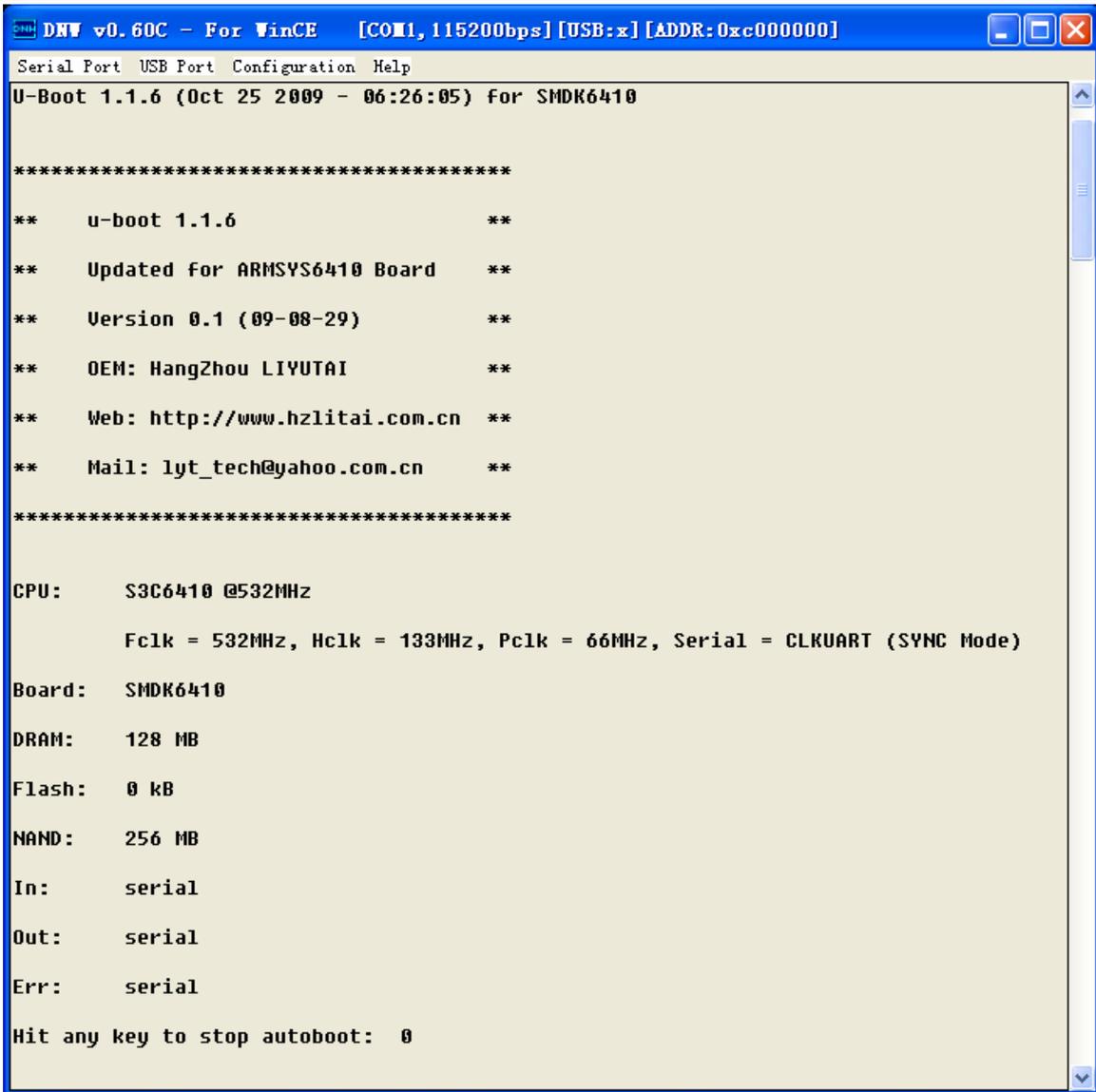
### 3.1.4 设置开发板从 Nandflash 启动

(1) 关闭开发板电源，设置 SW3，让开发板从 Nandflash 启动：

拨码开关SW3 引脚号	Pin 8	Pin 7	Pin 6	Pin 5	Pin 4	Pin 3	Pin 2	Pin 1
引脚定义	OM1	OM2	OM3	OM4	SELNAND	GPN15	GPN14	GPN13
Nandflash启动	0	0	1	1	0	X	X	X
SD卡启动	0	0	0	0	X	1	1	1
Norflash启动	0	1	0	1	X	X	X	X

拨码开关拨到“ON”位置上表示“1”，拨到“OFF”位置上表示“0”。

(2) 开启开发板电源，查看串口信息，烧录在 nandflash 中的 u-boot 已经启动。



```
DNW v0.60C - For WinCE [COM1, 115200bps] [USB:x] [ADDR: 0xc000000]
Serial Port USB Port Configuration Help
U-Boot 1.1.6 (Oct 25 2009 - 06:26:05) for SMDK6410

*****

**      u-boot 1.1.6                **
**      Updated for ARMSYS6410 Board  **
**      Version 0.1 (09-08-29)        **
**      OEM: HangZhou LIYUTAI        **
**      Web: http://www.hzlitai.com.cn **
**      Mail: lyt_tech@yahoo.com.cn   **
*****

CPU:      S3C6410 @532MHz
          Fclk = 532MHz, Hclk = 133MHz, Pclk = 66MHz, Serial = CLKUART (SYNC Mode)

Board:    SMDK6410
DRAM:     128 MB
Flash:    0 kB
NAND:     256 MB
In:       serial
Out:      serial
Err:      serial

Hit any key to stop autoboot: 0
```

### 3.1.5 设置 u-boot 工作参数

配置 ARMSYS6410 的 u-boot 包括以下项目，第一次使用时必须正确配置否则无法正常工作：

- 内核启动命令 **bootcmd**;
- 内核启动参数 **bootargs**。

采用命令 “**setenv**”来设置参数，“**saveenv**”来保存参数，例如：

```
SMDK6410 # bootargs "root=/dev/mtdblock2 rootfstype=yaffs2
console=ttySAC0,115200"
```

```
SMDK6410 # setenv bootcmd nand read 0xc0008000 0x100000 0x300000\;bootm
0xc0008000
```

```
SMDK6410 # saveenv
```

实际运行情况：

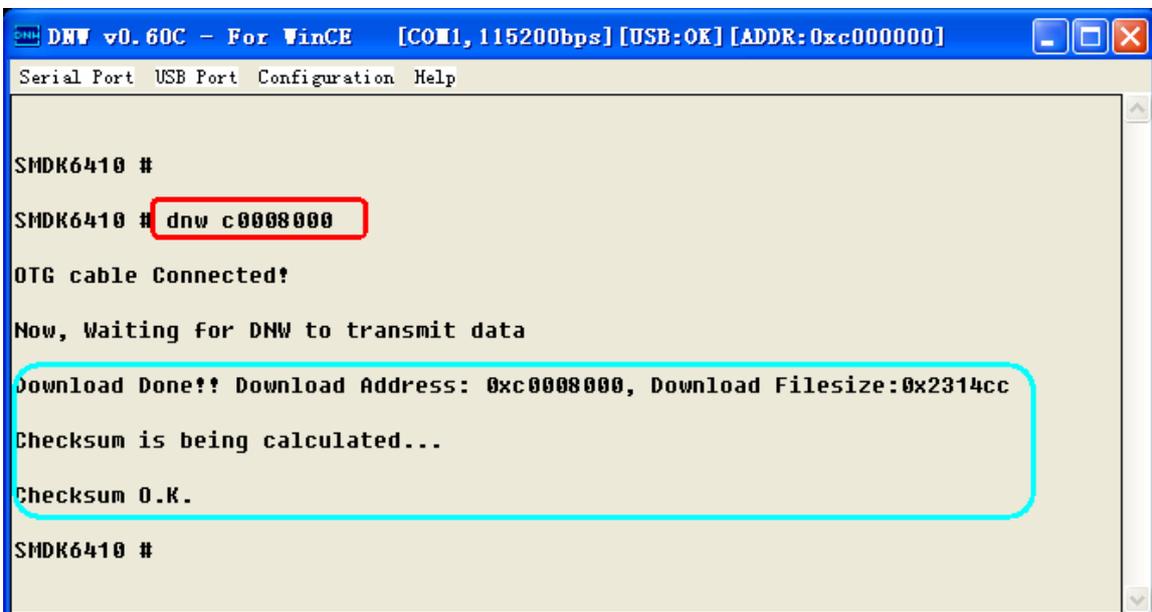
```
SMDK6410 # setenv bootargs "root=/dev/mtdblock2 rootfstype=yaffs2 console=ttySAC0.115200"  
SMDK6410 # setenv bootcmd nand read 0xc0008000 0x100000 0x300000\;bootm 0xc0008000  
SMDK6410 # saveenv  
Saving Environment to NAND...  
Erasing Nand...Writing to Nand... done  
SMDK6410 #
```

执行命令“printenv”，可以打印出您刚刚设置的参数。

### 3. 2 安装 kenerl

(1) 输入 dnw c0008000，并回车；

(2) 点击菜单“USB Port→Transmit→Transmit”，选中“ARMSYS6410-B 基础配置光盘资料\Linux 相关\linux2.6 标准版\映像文件\zImage”文件，下载映像文件到内存。



(3) 执行擦除命令：

```
nand erase 100000 300000
```

```
SMDK6410 # nand erase 100000 300000
```

```
NAND erase: device 0 offset 0x100000, size 0x300000  
Erasing at 0x3e0000 -- 100% complete.  
OK
```

(4) 执行写入命令：

```
nand write c0008000 100000 300000
```

```
SMDK6410 # nand write c0008000 100000 300000
```

```
NAND write: device 0 offset 0x100000, size 0x300000  
3145728 bytes written: OK  
SMDK6410 #
```

(5) 输入命令复位开发板：

```
reset
```

以下是完整的内核启动信息：

Uncompressing

Linux.....

..... done, booting the kernel.

Linux version 2.6.28.6 (lyt@lyt-desktop) (gcc version 4.2.2) #1 Thu Oct 29  
16:43:10 CST 2009

CPU: ARMv6-compatible processor [410fb766] revision 6 (ARMv7), cr=00c5387f

CPU: VIPT nonaliasing data cache, VIPT nonaliasing instruction cache

Machine: SMDK6410

Memory policy: ECC disabled, Data cache writeback

CPU S3C6410 (id 0x36410101)

S3C24XX Clocks, (c) 2004 Simtec Electronics

S3C64XX: PLL settings, A=532000000, M=532000000, E=24000000

S3C64XX: HCLKx2=266000000, HCLK=133000000, PCLK=66500000

div1: 00000555

mout\_apll: source is fout\_apll (1), rate is 532000000

mout\_epll: source is fout\_epll (1), rate is 24000000

mout\_mppll: source is mppll (1), rate is 532000000

mmc\_bus: source is dout\_mppll (1), rate is 44333333

mmc\_bus: source is dout\_mppll (1), rate is 44333333

mmc\_bus: source is dout\_mppll (1), rate is 44333333

usb-host-bus: source is mout\_epll (0), rate is 24000000

uclk1: source is dout\_mppll (1), rate is 66500000

spi-bus: source is mout\_epll (0), rate is 24000000

spi-bus: source is mout\_epll (0), rate is 24000000

audio-bus0: source is mout\_epll (0), rate is 24000000

audio-bus1: source is mout\_epll (0), rate is 24000000

audio-bus2: source is mout\_epll (0), rate is 24000000

irda-bus: source is mout\_epll (0), rate is 24000000

s3c64xx: 15728640 bytes SDRAM reserved for fimc at 0x505db000

s3c64xx: 8388608 bytes SDRAM reserved for pp at 0x514db000

s3c64xx: 8388608 bytes SDRAM reserved for tv at 0x51cdb000

s3c64xx: 6291456 bytes SDRAM reserved for mfc at 0x524db000

s3c64xx: 8388608 bytes SDRAM reserved for jpeg at 0x52adb000

s3c64xx: 8388608 bytes SDRAM reserved for cmm at 0x532db000

Built 1 zonelists in Zone order, mobility grouping on. Total pages: 32512

Kernel command line: root=/dev/mtdblock2 rootfstype=yaffs2

console=ttySAC0,115200

PID hash table entries: 512 (order: 9, 2048 bytes)

Console: colour dummy device 80x30

s3c24xx\_serial\_init\_ports: initialising ports=4...

console [ttySAC0] enabled

Dentry cache hash table entries: 16384 (order: 4, 65536 bytes)

Inode-cache hash table entries: 8192 (order: 3, 32768 bytes)

Memory: 128MB = 128MB total

Memory: 70556KB available (3888K code, 494K data, 520K init)  
SLUB: Genslabs=12, HWalign=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1  
Calibrating delay loop... 530.84 BogoMIPS (lpj=1327104)  
Mount-cache hash table entries: 512  
CPU: Testing write buffer coherency: ok  
net\_namespace: 316 bytes  
NET: Registered protocol family 16  
S3C6410: Initialising architecture  
S3C DMA-pl080 Controller Driver, (c) 2006-2007 Samsung Electronics  
Total 32 DMA channels will be initialized.  
SCSI subsystem initialized  
usbcore: registered new interface driver usbfs  
usbcore: registered new interface driver hub  
usbcore: registered new device driver usb  
NET: Registered protocol family 2  
IP route cache hash table entries: 1024 (order: 0, 4096 bytes)  
TCP established hash table entries: 4096 (order: 3, 32768 bytes)  
TCP bind hash table entries: 4096 (order: 4, 81920 bytes)  
TCP: Hash tables configured (established 4096 bind 4096)  
TCP reno registered  
NET: Registered protocol family 1  
NetWinder Floating Point Emulator V0.97 (double precision)  
yaffs Oct 29 2009 16:36:37 Installing.  
msgmni has been set to 138  
alg: No test for stdrng (krng)  
io scheduler noop registered  
io scheduler anticipatory registered  
io scheduler deadline registered  
io scheduler cfq registered (default)  
S3C\_LCD clock got enabled :: 133.000 Mhz  
LCD TYPE :: L70T84 will be initialized  
Window[0] - FB1: map\_video\_memory: clear ff000000:00177000  
          FB1: map\_video\_memory: dma=57200000 cpu=ff000000  
size=00177000  
Window[0] - FB2: map\_video\_memory: clear ff0bb800:000bb800  
          FB2: map\_video\_memory: dma=572bb800 cpu=ff0bb800  
size=000bb800  
Console: switching to colour frame buffer device 100x30  
fb0: s3cfb frame buffer device  
Window[1] - FB1: map\_video\_memory: clear ff177000:00177000  
          FB1: map\_video\_memory: dma=57400000 cpu=ff177000  
size=00177000  
Window[1] - FB2: map\_video\_memory: clear ff232800:000bb800  
          FB2: map\_video\_memory: dma=574bb800 cpu=ff232800

```
size=000bb800
fb1: s3cfb frame buffer device
Window[2] - FB1: map_video_memory: clear ff2ee000:000bb800
      FB1:      map_video_memory:      dma=57600000      cpu=ff2ee000
size=000bb800
fb2: s3cfb frame buffer device
Window[3] - FB1: map_video_memory: clear ff3aa000:000bb800
      FB1:      map_video_memory:      dma=57700000      cpu=ff3aa000
size=000bb800
fb3: s3cfb frame buffer device
s3c6400-uart.0: s3c2410_serial0 at MMIO 0x7f005000 (irq = 16) is a S3C6400/10
s3c6400-uart.1: s3c2410_serial1 at MMIO 0x7f005400 (irq = 20) is a S3C6400/10
s3c6400-uart.2: s3c2410_serial2 at MMIO 0x7f005800 (irq = 24) is a S3C6400/10
s3c6400-uart.3: s3c2410_serial3 at MMIO 0x7f005c00 (irq = 28) is a S3C6400/10
brd: module loaded
loop: module loaded
PPP generic driver version 2.4.2
dm9000 Ethernet Driver
eth2090023871: con201 Invalid ethernet MAC address. using default config,
Please set using ifconfig
eth0: dm9000 at f7b00300,f7b00304 IRQ 111 MAC: 00:e0:4a:bc:15:e7
Linux video capture interface: v2.00
s3c-fimc: controller 0 registered successfully
s3c-fimc: controller 1 registered successfully
S3C6400 MFC Driver, (c) 2007 Samsung Electronics
S3C6400 MFC Driver, (c) 2007 Samsung Electronics
S3C PostProcessor Driver v3.12, (c) 2009 Samsung Electronics
S3C6410 TV encoder Driver, (c) 2008 Samsung Electronics
S3C6410 TV encoder Driver init OK.
S3C6410 TV scaler Driver, (c) 2008 Samsung Electronics
S3C6410 TV scaler Driver init OK.
S3C Rotator Driver, (c) 2008 Samsung Electronics
s3c_rotator_probe called
s3c_rotator_probe success
S3C JPEG Driver, (c) 2007 Samsung Electronics
s3c_g2d_probe called
s3c_g2d_probe Success
S3C G2D Init : Done
S3C G3D Driver, (c) 2007-2009 Samsung Electronics
s3c_g3d version : 0x1050000
S3C G3D Init : Done
S3C CMM Driver, (c) 2008 Samsung Electronics
Driver 'sd' needs updating - please use bus_type methods
S3C NAND Driver, (c) 2008 Samsung Electronics
杭州立宇泰电子有限公司          第 20 页 共 51 页          电话: 0571-56763523 56763526
网址: www.hzlitai.com.cn          传真: 0571-56763523-808
www.armsystem.com.cn
```

S3C NAND Driver is using software ECC.  
NAND device: Manufacturer ID: 0xec, Chip ID: 0xda (Samsung NAND 256MiB 3,3V 8-bit)  
Creating 4 MTD partitions on "NAND 256MiB 3,3V 8-bit":  
0x00000000-0x00040000 : "Bootloader"  
0x00040000-0x00400000 : "Kernel"  
0x00400000-0x05400000 : "Rootfs"  
0x05400000-0x10000000 : "File System"  
ohci\_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver  
s3c2410-ohci s3c2410-ohci: S3C24XX OHCI  
s3c2410-ohci s3c2410-ohci: new USB bus registered, assigned bus number 1  
s3c2410-ohci s3c2410-ohci: irq 79, io mem 0x74300000  
usb usb1: configuration #1 chosen from 1 choice  
hub 1-0:1.0: USB hub found  
hub 1-0:1.0: 2 ports detected  
Initializing USB Mass Storage driver...  
usbcore: registered new interface driver usb-storage  
USB Mass Storage support registered.  
s3c-udc : S3C HS USB OTG Device Driver, (c) 2008-2009 Samsung Electronics  
s3c-udc : version 15 March 2009 (DMA Mode)  
mice: PS/2 mouse device common for all mice  
[KEY]Initialize the GPIO for keybord of ARMSYS6410.<6>input: s3c-keypad as  
/class/input/input0  
s3c-keypad Initialized  
S3C Keypad Driver  
S3C Touchscreen driver, (c) 2008 Samsung Electronics  
S3C TouchScreen got loaded successfully : 12 bits  
input: S3C TouchScreen as /class/input/input1  
S3C24XX RTC, (c) 2004,2006 Simtec Electronics  
s3c2410\_rtc: tick irq 34, alarm irq 92  
s3c2410-rtc s3c2410-rtc: rtc disabled, re-enabling  
s3c2410-rtc s3c2410-rtc: rtc core: registered s3c as rtc0  
i2c /dev entries driver  
s3c2440-i2c s3c2440-i2c.0: slave address 0x10  
s3c2440-i2c s3c2440-i2c.0: bus frequency set to 377 KHz  
parent clock for camera: 266.000 MHz, divisor: 11  
[CAM]Reset and init reg!1cam->client=0  
[CAM]Reset and init reg!3  
s3c2440-i2c s3c2440-i2c.0: i2c-0: S3C I2C adapter  
s3c2440-i2c s3c2440-i2c.1: slave address 0x10  
s3c2440-i2c s3c2440-i2c.1: bus frequency set to 377 KHz  
parent clock for camera: 266.000 MHz, divisor: 11  
[CAM]Reset and init reg!1cam->client=0  
[CAM]Reset and init reg!3

```
s3c-fimc: OV965X attached successfully
s3c2440-i2c s3c2440-i2c.1: i2c-1: S3C I2C adapter
sdhci: Secure Digital Host Controller Interface driver
sdhci: Copyright(c) Pierre Ossman
s3c-sdhci s3c-sdhci.0: clock source 0: hsmmc (133000000 Hz)
s3c-sdhci s3c-sdhci.0: clock source 1: hsmmc (133000000 Hz)
s3c-sdhci s3c-sdhci.0: clock source 2: mmc_bus (44333333 Hz)
[SDHCI]to add external irq as a card detect signal.....
[SDHCI]if (pdata->cfg_ext_cd).....
mmc0: SDHCI controller on samsung-hsmmc [s3c-sdhci.0] using ADMA
[SDHCI]request_irq.....
sdhci: card inserted.
s3c-sdhci s3c-sdhci.1: clock source 0: hsmmc (133000000 Hz)
s3c-sdhci s3c-sdhci.1: clock source 1: hsmmc (133000000 Hz)
s3c-sdhci s3c-sdhci.1: clock source 2: mmc_bus (44333333 Hz)
[SDHCI]to add external irq as a card detect signal.....
mmc1: SDHCI controller on samsung-hsmmc [s3c-sdhci.1] using ADMA
usbcore: registered new interface driver usbhid
usbhid: v2.6:USB HID core driver
Advanced Linux Sound Architecture Driver Version 1.0.18rc3.
ASoC version 0.13.2
WM9713/WM9714 SoC Audio Codec 0.15
playback: 1, capture : 1
asoc: AC97 HiFi <-> s3c64xx-ac97 mapping ok
[WM9713]Open speaker volume.
ALSA device list:
  #0: SMDK6400 (WM9713)
TCP cubic registered
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
VFP 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
VFS: Mounted root (yaffs2 filesystem).
Freeing init memory: 520K
Warning: unable to open an initial console.
Kernel panic - not syncing: No init found.  Try passing init= option to kernel.
```

由于没有安装根文件系统，因此最后一行显示：Kernel panic - not syncing: No init found.  
下面我们来安装根文件系统。

## 3. 3 安装根文件系统

### 3.3.1 根文件系统目录

目前 ARMSYS6410 支持 YAFFS2 文件系统和 CRAMFS 文件系统。光盘资料中提供了两套文件系统目录，位于 rootfilesystem 目录下。

- root\_armsys6410\_v1.0.tgz: YAFFS2 根文件系统采用的目录压缩包。
- armsys6410.cramfs: cramfs 文件系统映象文件；

下面我们用挂载 NFS 的方式来更新 yaffs2 文件系统。先看如何建立 NFS 挂载根文件系统。

### 3.3.2 NFS 方式挂载根文件系统

下面针对 ARMSYS6410 开发板介绍建立 NFS 服务中的基本设置,如果您第一次使用 NFS 服务,请参考一些网文或书籍来了解如何建立 NFS 服务,及如何建立虚拟机的网卡桥接等。

#### 3.3.2.1 准备好 NFS 共享目录

启动 nfs 服务之前,必须在主机上准备好 NFS 共享目录。

例如我们采用 /home/localhost/armsys6410/root 作为 NFS 共享目录,将光盘中的 rootfilesystem/ root\_armsys6410\_v1.0.tgz 压缩文件拷贝到这个目录下,然后再解压缩,得到根文件系统所需要目录。将压缩文件 root\_armsys6410\_v1.0.tgz 压缩文件更名为 root\_armsys.tgz 以备烧录之用。

#### 3.3.2.2 主机上设置

(1) 修改主机上/etc/exports 文件的内容,添加一行:

```
/home/localhost/armsys6410/root *(rw, sync, no_root_squash)
```

注意: root和\*之间有空格!

这里/home/localhost/armsys6410/root 代表要共享输出的 root 文件系统的共享目录; \*代表所有的客户机都可以挂接此文件系统; rw 代表客户机可以读写许可来挂接它们的根文件系统; no\_root\_squash 选项允许客户机以主机上的 root 身份挂接根文件系统。

(2) 配置好主机 IP 地址

主机 IP 和开发板的 IP 要在同一个网段,例如:

开发板 IP: 192.168.253.12

主机 IP: 192.68.253.2

(3) 在主机上启动 NFS 服务:

```
Sudo /etc/init.d/portmap restart
```

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

### 3.3.2.3 在开发板上设定启动参数

#### (1) 通过 u-boot 设置启动参数

u-boot 启动时，在命令提示符下设置启动参数如下：

```
setenv bootargs "root=/dev/nfs
nfsroot=192.168.253.2:/home/localhost/armsys6410/root/
ip=192.168.253.12:192.168.253.2:192.168.253.2:255.255.255.0:armsys.hzlitai.com
:eth0:off console=ttySAC0,115200"
```

命令行必须在一行中输入完。因此我们提供了一个 u-boot/linux\_arg.txt 文本文件，其中已经写好了这行，只要将它拷贝并粘贴到 SMDK6410#后即可。

注意，以上命令行假定主机地址为 192.168.253.2，ARMSYS6410 开发板的 IP 地址为 192.168.253.12。如果你的主机 IP 与此不同，请用 ifconfig 重新设定，或者修改该命令行的内容。

#### (2) 保存参数

```
SMDK6410# saveenv
```

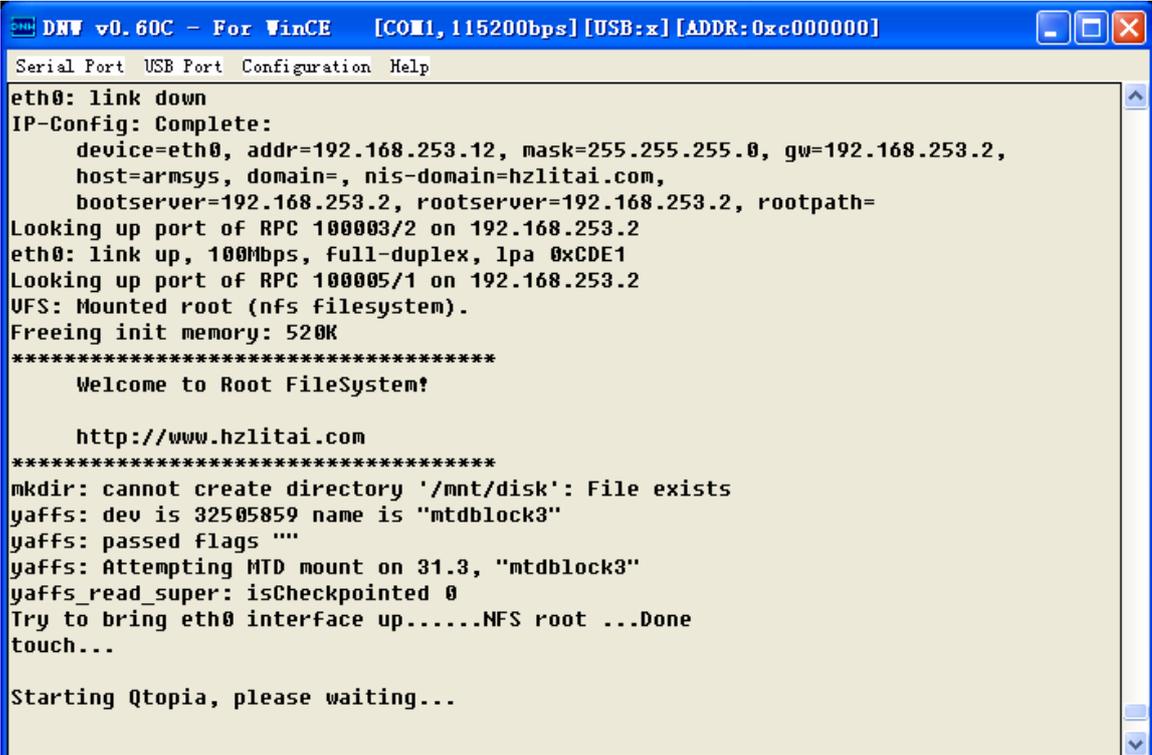
即可保存刚刚设定的参数，下一次启动就不需要重复输入上述命令行了。

### 3.3.2.4 启动并挂载

确保你已经连接好开发板和主机之间的网线，按照前面的步骤设置好 nfs 服务，现在可以在 SMDK6410#下输入：

```
reset
```

内核启动后将自动挂载上 nfs 文件系统，信息如下：

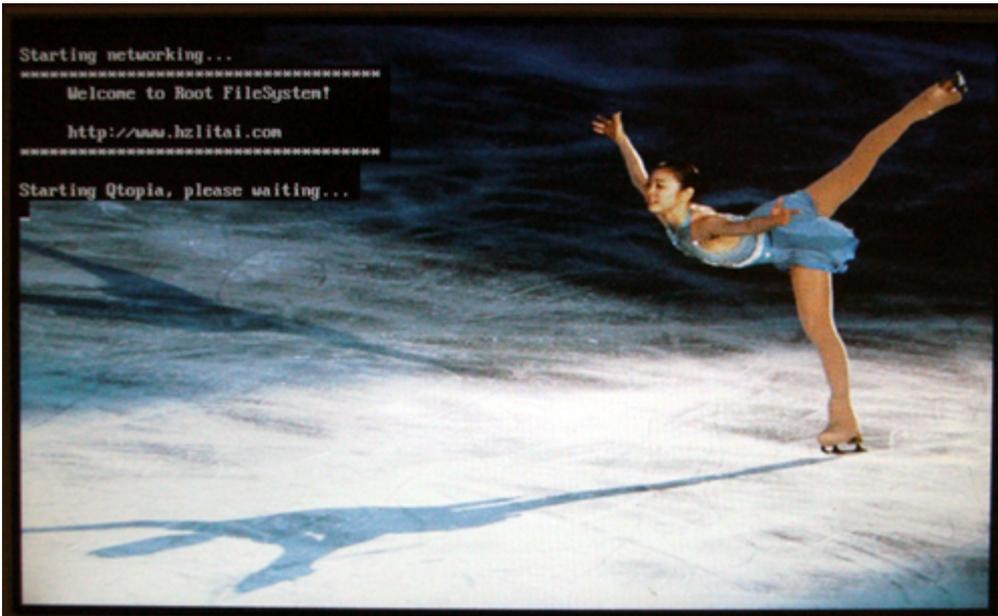


```
DHW v0.60C - For WinCE [COM1,115200bps] [USB:x] [ADDR:0xc000000]
Serial Port USB Port Configuration Help
eth0: link down
IP-Config: Complete:
  device=eth0, addr=192.168.253.12, mask=255.255.255.0, gw=192.168.253.2,
  host=armsys, domain=, nis-domain=hzlitai.com,
  bootserver=192.168.253.2, rootserver=192.168.253.2, rootpath=
Looking up port of RPC 100003/2 on 192.168.253.2
eth0: link up, 100Mbps, full-duplex, lpa 0xCDE1
Looking up port of RPC 100005/1 on 192.168.253.2
UFS: Mounted root (nfs filesystem).
Freeing init memory: 520K
*****
Welcome to Root FileSystem!

http://www.hzlitai.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"
yaffs_read_super: isCheckpointed 0
Try to bring eth0 interface up.....NFS root ...Done
touch...

Starting Qtopia, please waiting...
```

最后部分的信息,说明已经成功地 mount 上了主机的 root 文件系统。液晶屏上显示 LOGO 图片效果如下图:



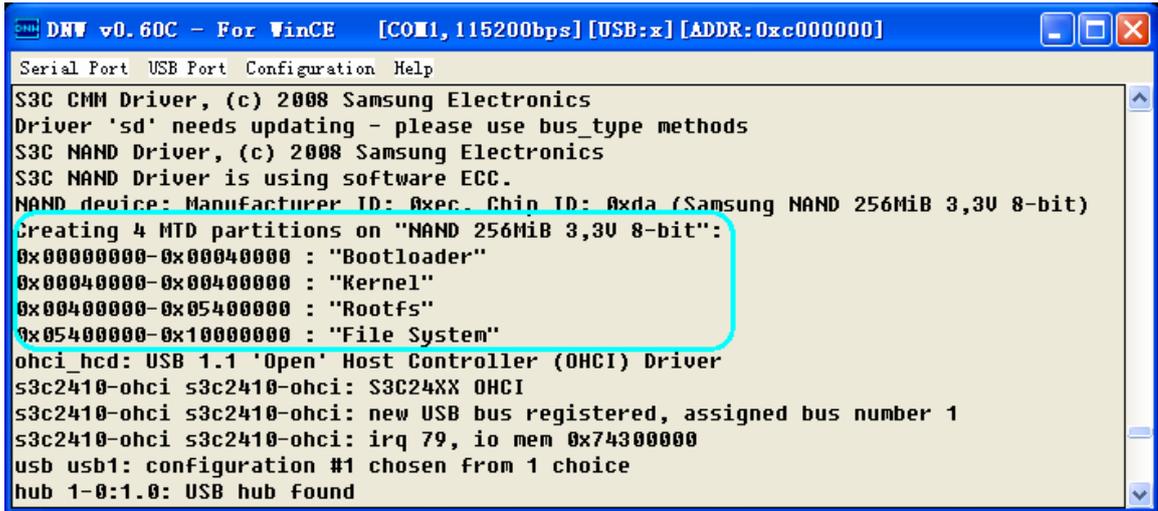
显示 qtopia 桌面的效果:



### 3.3.3 安装 Yaffs2 文件系统

#### 3.3.3.1 擦除 nandflash 分区

在 linux 内核启动信息中可以注意到以下信息:



```
DHW v0.60C - For WinCE [COM1,115200bps] [USB:x] [ADDR:0xc000000]
Serial Port USB Port Configuration Help
S3C CMM Driver, (c) 2008 Samsung Electronics
Driver 'sd' needs updating - please use bus_type methods
S3C NAND Driver, (c) 2008 Samsung Electronics
S3C NAND Driver is using software ECC.
NAND device: Manufacturer ID: 0xec, Chip ID: 0xda (Samsung NAND 256MiB 3,3V 8-bit)
Creating 4 MTD partitions on "NAND 256MiB 3,3V 8-bit":
0x00000000-0x00040000 : "Bootloader"
0x00040000-0x00400000 : "Kernel"
0x00400000-0x05400000 : "Rootfs"
0x05400000-0x10000000 : "File System"
ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
s3c2410-ohci s3c2410-ohci: S3C24XX OHCI
s3c2410-ohci s3c2410-ohci: new USB bus registered, assigned bus number 1
s3c2410-ohci s3c2410-ohci: irq 79, io mem 0x74300000
usb usb1: configuration #1 chosen from 1 choice
hub 1-0:1.0: USB hub found
```

系统建立了 4 个 MTD 分区，前面 3 个分别用来存放 bootloader、kernel 和 rootfs。后面的 File System 分区由用户自定义使用。

我们采用第 3 个分区，也就是 mtdblock2 作为存放根文件系统分区。

首先采用 u-boot 的擦除功能对它进行擦除。进入 u-boot 的提示符，输入擦除命令：

```
nand erase 400000 500000
```

这样就擦除了根文件系统所需要的空间。

### 3.3.3.2 烧录 YAFFS2 文件系统

启动内核挂载上 NFS 文件系统。我们通过运行 root/bin/下的 Writerootfs 脚本文件，可以一次性完成 Yaffs2 根文件系统在 mtdblock2 分区的固化。

Writerootfs 的实际步骤是：

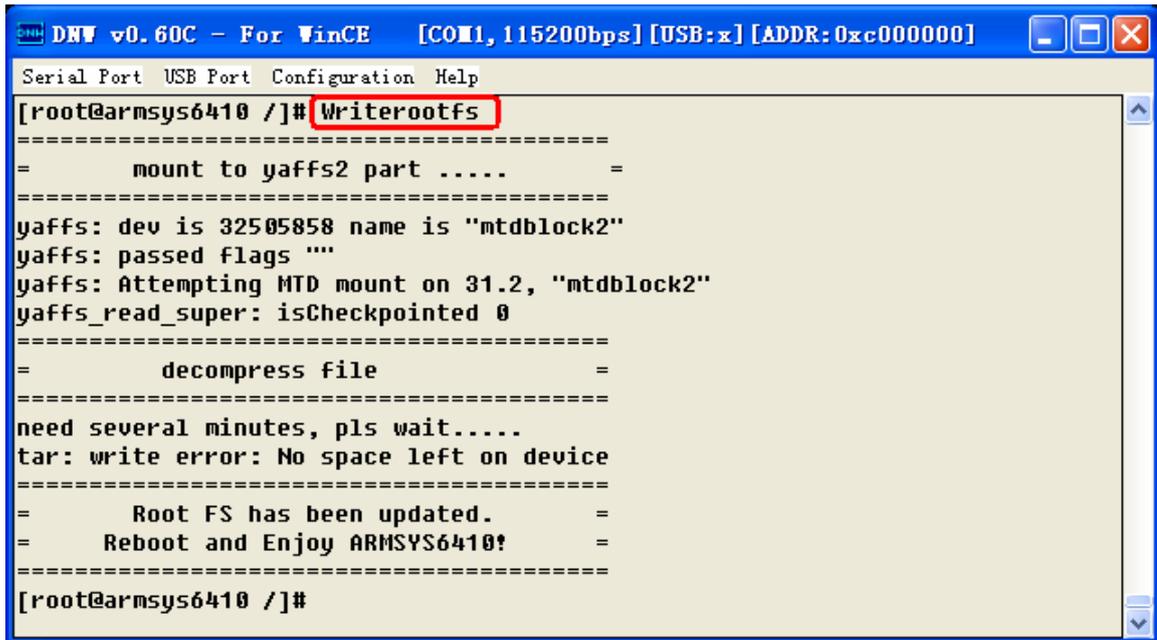
```
mount -t yaffs2 /dev/mtdblock2 /mnt
```

```
cd /mnt
```

```
tar xzf /root_armsys.tgz
```

所以一定要保证固化用的 root\_armsys.tgz 放在根目录下（参考 3.3.2.1 节）。

在控制台上输入 Writerootfs，显示结果如下：



```
DNW v0.60C - For WinCE [COM1, 115200bps] [USB:x] [ADDR:0xc000000]
Serial Port USB Port Configuration Help
[root@armsys6410 /]# 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.....
tar: write error: No space left on device
=====
=          Root FS has been updated.            =
=          Reboot and Enjoy ARMSYS6410!        =
=====
[root@armsys6410 /]#
```

说明已经完成了 root 文件系统的更新，下面我们就可以从更新的文件系统启动了。

### 3.3.3.3 从 YAFFS2 文件系统启动

在 u-boot 中重新设置系统启动参数，同样可以从 u-boot/linux\_arg.txt 文本文件中拷贝获得：

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

这里我们从 mtdblock2 分区启动 root 文件系统。

可以用以下命令行保存该设置：

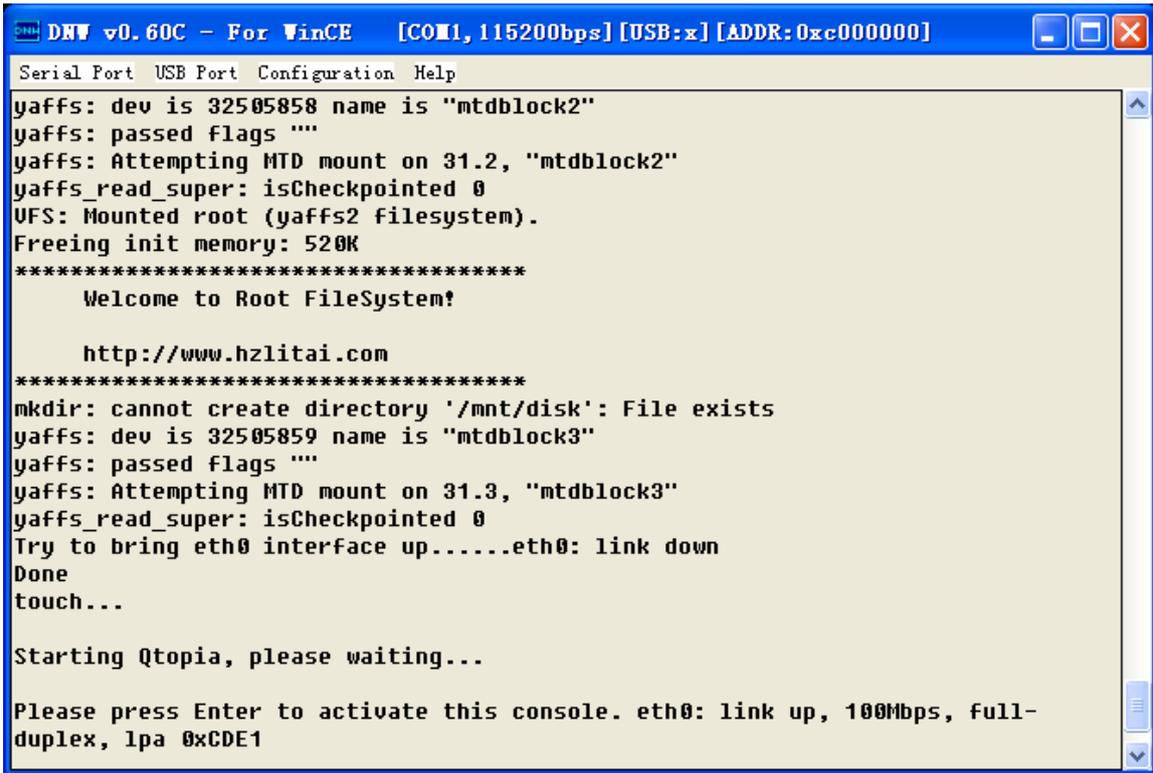
**saveenv**

重新启动系统：

**reset**

由于采用了可读写的 YAFFS2 文件系统，你所保存的文件和设置在掉电后都不会丢失。

以下是 yaffs2 文件系统启动信息：



```
Serial Port USB Port Configuration Help
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: 520K
*****
Welcome to Root FileSystem!

http://www.hzlitai.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"
yaffs_read_super: isCheckpointed 0
Try to bring eth0 interface up.....eth0: link down
Done
touch...

Starting Qtopia, please waiting...

Please press Enter to activate this console. eth0: link up, 100Mbps, full-
duplex, lpa 0xC0E1
```

### 3.3.4 安装 cramfs 文件

a) 进入 u-boot 的提示符，下载根文件系统的映像：

```
SMDK6410 # dnw c0008000
```

通过 dnw，下载“rootfilesystem/armsys6410.cramfs”文件。

b) 擦除 mtdblock2 所在分区：

```
SMDK6410 # nand erase 400000 5000000
```

c) 再通过以下命令将 cramfs 文件系统写入 Nandflash：

```
SMDK6410 # nand write c0008000 400000 5000000
```

d) 重新设置启动参数，从 cramfs 文件系统启动：

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

e) 保存参数：

```
saveenv
```

重新复位开发板，系统就自动从 cramfs 文件系统启动了。

## 4. 安装交叉编译器

交叉编译工具链的安装文件为“ARMSYS6410-B基础配置光盘资料\Linux相关\linux2.6 标准版\交叉编译器”目录下的 cross-4.2.2-eabi.tar.bz2。

执行解包：

```
tar xjvf cross-4.2.2-eabi.tar.bz2 -C /usr/local/arm
```

解包后将在/usr/local/arm/目录下生成“4.2.2-eabi”目录。

这样，内核或其他应用程序均可用/usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-来指定使用该交叉编译器。

## 5. u-boot 源码包的编译

u-boot 源码包位于“ARMSYS6410-B 基础配置光盘资料\Linux 相关\linux2.6 标准版\uboot\”目录下，将压缩包 uboot1.1.6\_armsys6410\_v1.0.tgz 拷贝到你的工作目录下，解压缩并编译：

```
tar xzf uboot1.1.6_armsys6410_v1.0.tgz
```

```
cd uboot1.1.6_armsys6410_v1.0
```

```
make smdk6410_config
```

```
make
```

如果编译成功，将在 **uboot1.1.6\_armsys6410\_v1.0** 目录下产生名为 u-boot.bin 的二进制文件。为了和 SD 卡启动的 uboot 区别，我们重命名为 u-boot-nand.bin。

u-boot-sd.bin 文件为从 SD 卡启动的二进制文件，不提供源码，请用户直接使用映像文件。

## 6. Linux 内核源码包的编译

内核源码包位于“ARMSYS6410-B 基础配置光盘资料\Linux 相关\linux2.6 标准版\kernel\”目录下，将压缩包 linux2.6.28\_armsys6410\_v1.0.tgz 拷贝到你的工作目录下，解压缩：

```
tar xzf linux2.6.28_armsys6410_v1.0.tgz
```

解压成功后产生 linux2.6.28\_armsys6410 目录。

### 6. 1 配置内核

您可能需要安装 libncurses5，方便使用 make menuconfig 命令，采用以下命令行安装：

```
sudo apt-get install libncurses5-dev
```

首先恢复内核的默认配置：

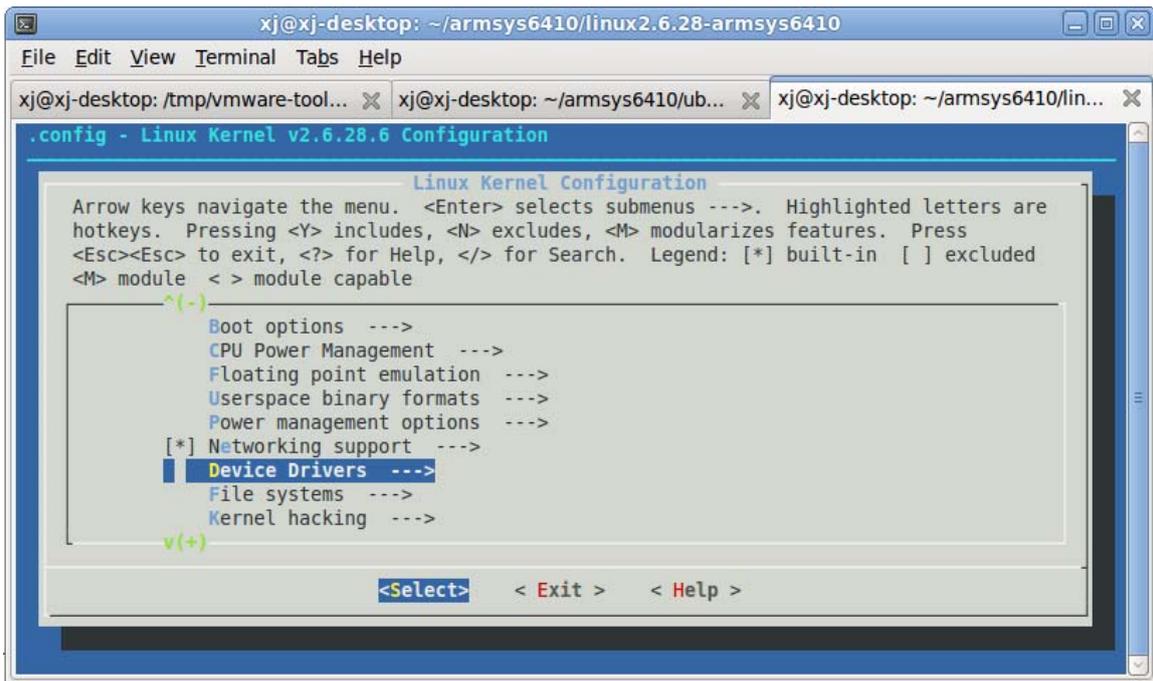
```
cd linux2.6.28_armsys6410
```

```
cp ARMSYS640_L70T84_CFG .config
```

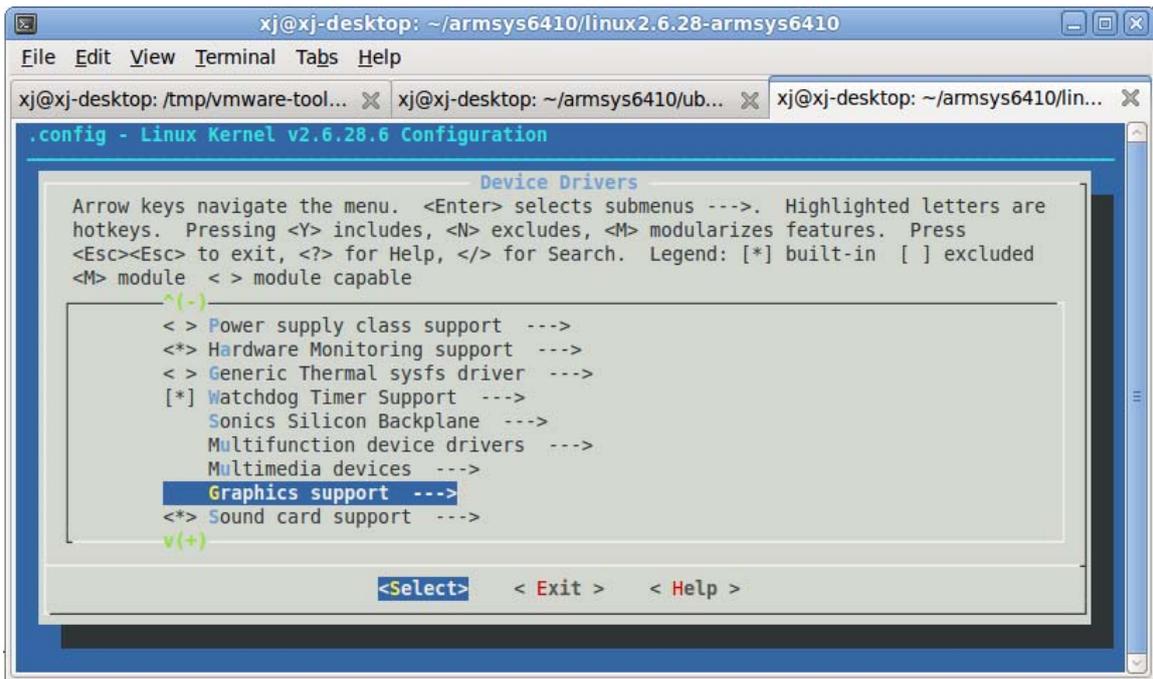
选择液晶屏类型：

```
make menuconfig
```

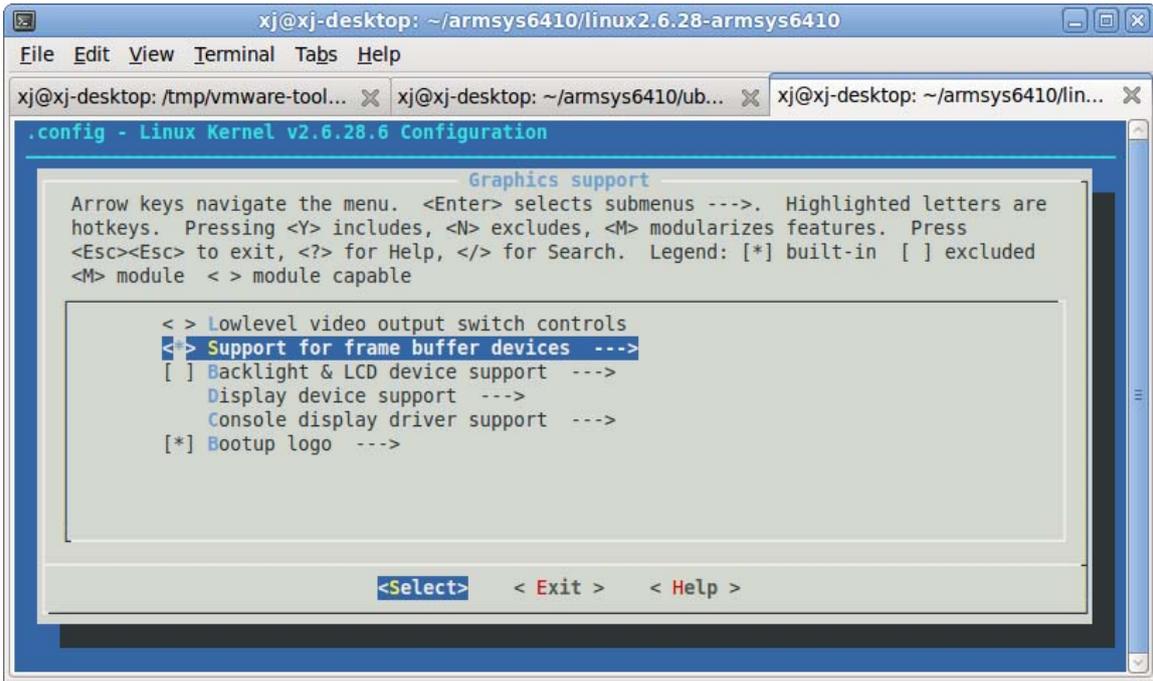
进入设备驱动配置，Device Drivers→



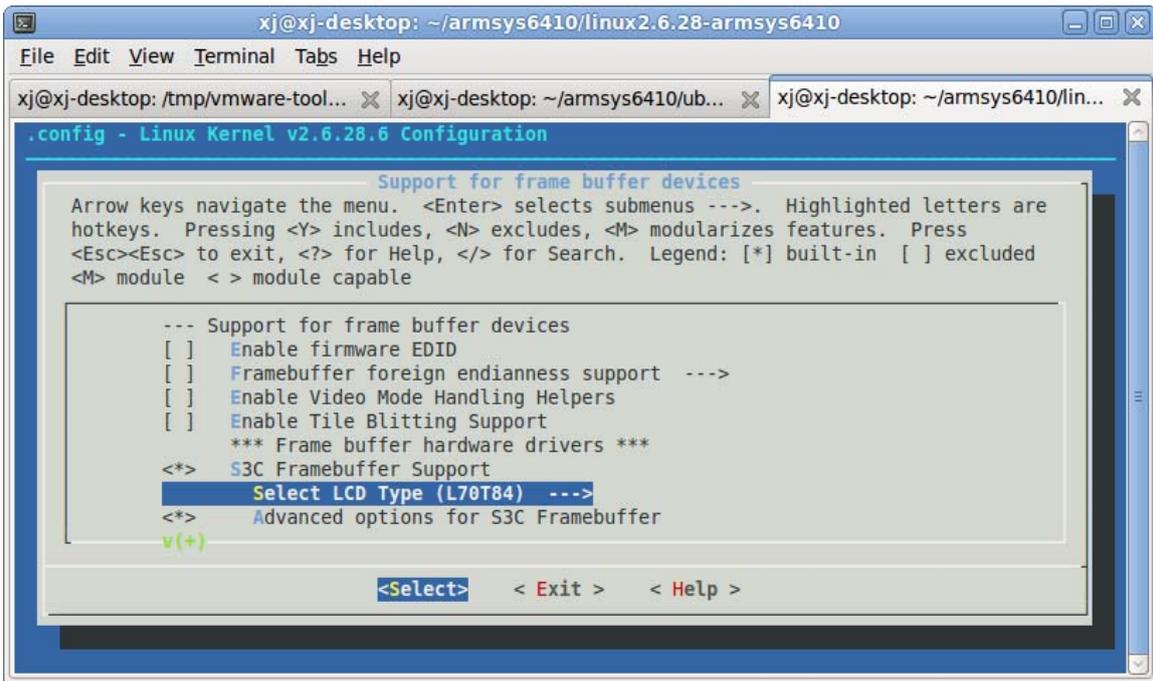
进入图形设备配置，Graphics support →



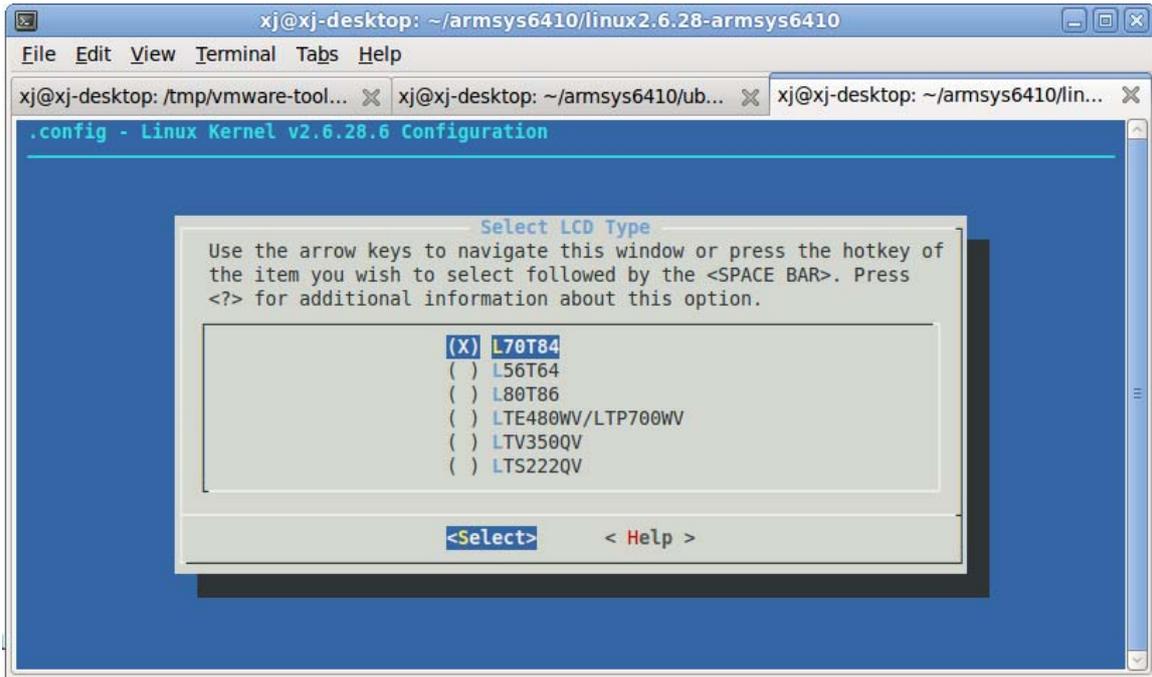
进入 framebuffer 设备配置，Support for frame buffer devices →



选择 LCD 类型:



根据你配套的液晶屏类型，选择液晶屏:



这样就完成了对内核的基本配置。

## 6. 2 编译内核

命令行如下：

**make**

编译结束后将在 arch/arm/boot 中得到 linux 内核压缩映像文件：zImage。

## 7. 设备驱动与测试

首先我们使用立字泰提供的一个综合测试程序“ARMSYS6410-B 基础配置光盘资料\Linux 相关\linux2.6 标准版\测试实例\multimediatest\_armsys6410\_v1.0.tgz”来进行硬件编解码、显示协处理、TVOUT 输出、摄像头视频采集和编码的功能。该工具包包含一个测试程序 mediatest 和一组媒体文件，开发板出厂时，已经保存在 mtdblock3 分区中，即目录/mnt/disk/下。

进入该目录运行测试程序：

**cd /mnt/disk/multimediatest\_armsys6410\_v1.0**

**./Mediatest**

显示如下界面：

```
===== ARMSYS6410 Media Demo Application =====
Modify by liyutai, 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 -->
```

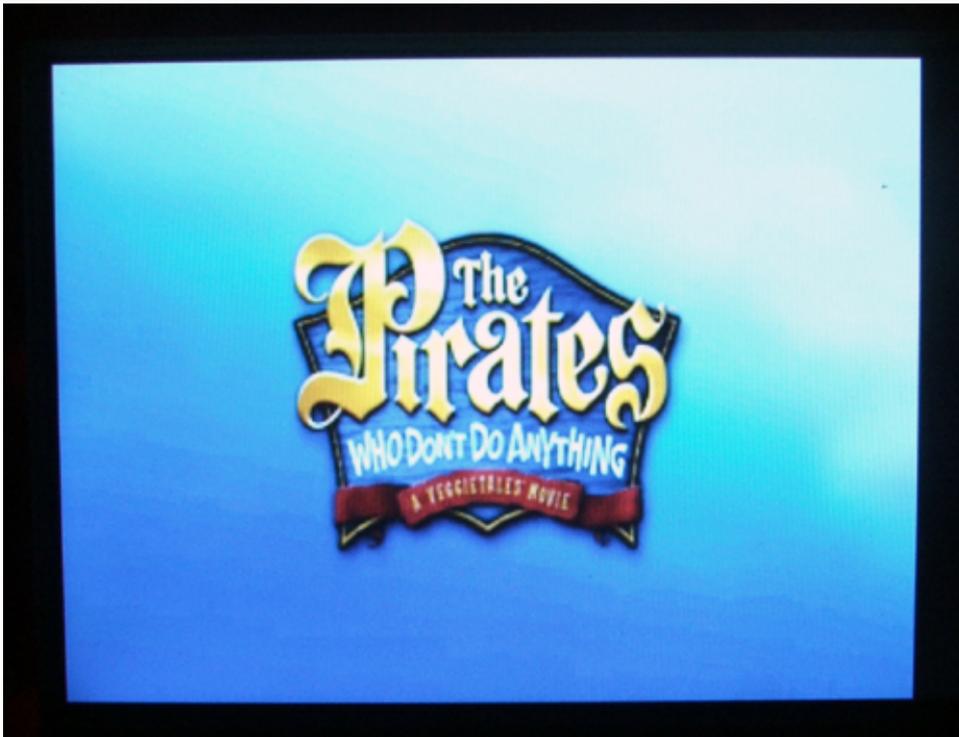
## 7. 1 视频硬件解码测试 (H.263/H.264, MPEG4, VC-1 格式)

输入 1, 开始 H.264 格式视频测试:

```
-----
Select number --> 1
===== H.264 File Decodec Test =====
liyutai, 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
```

播放效果:



【注意】如何进行过 Camera 视频编码测试，则这里会自动播放 Cam\_encoding\_320x240-1.264 视频编码文件。

输入 2，开始 MPEG4 格式视频播放测试：

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

播放效果如下：



输入 3，开始 H.263 格式视频播放测试：

```
Select number --> 3
[3. H.263 display]
Using IP      : MFC, Post processor, LCD
Input filename : iron.263
Input vector size : QVGA(320x240)
Display size  : WVGA(800x480)
Bitrate      : 460 Kbps
FPS          : 30
```

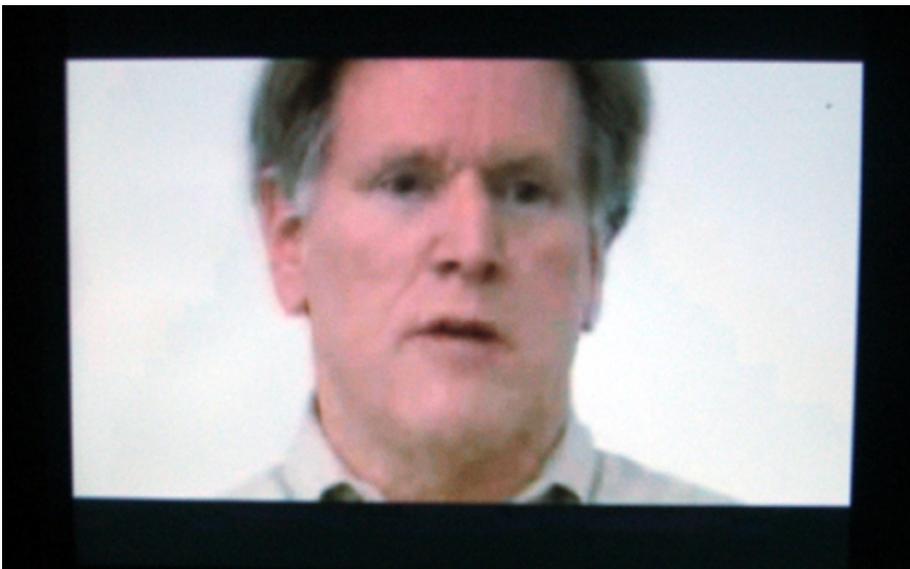
播放效果如下图所示：



输入 4，开始 VC-1 格式视频播放测试：

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

播放效果如下图：

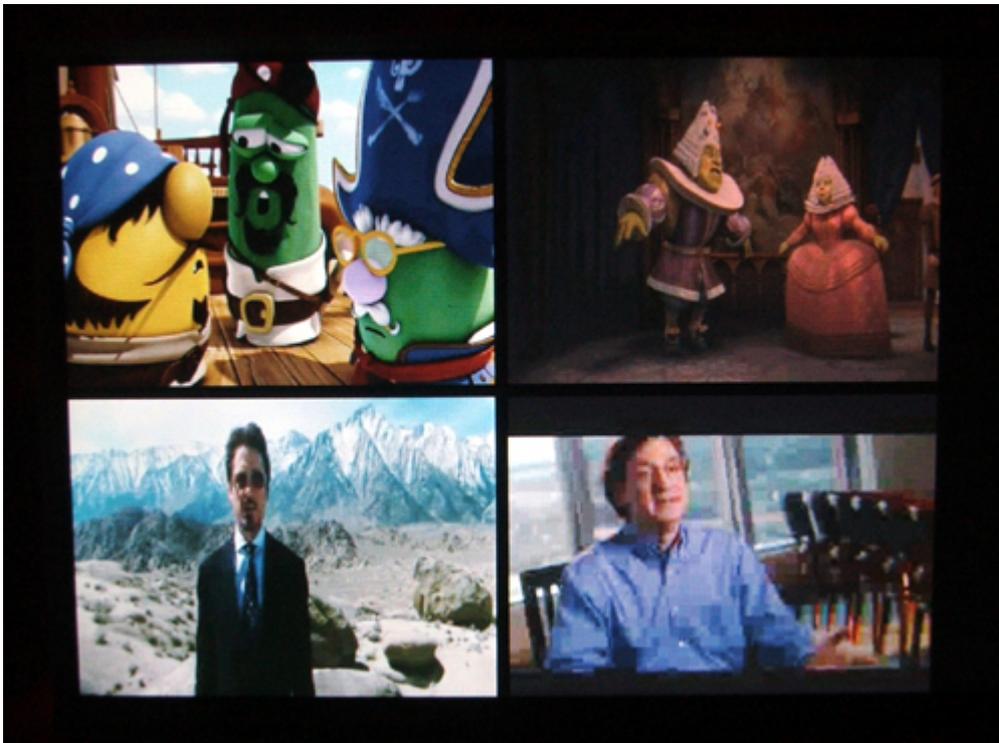


## 7. 2 多种视频格式同时解码（4 窗口）测试

在主菜单下，输入 5，开始本项测试：

```
[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 : QVGA       * Input vector size : QVGA       *
* 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             : VC-1        *
* Input filename    : iron.263   * Input filename    : test2_0.rcv  *
* Input vector size : QVGA       * Input vector size : QVGA       *
* Display size      : 400x240    * Display size      : 400x240    *
* Bitrate           : 460 Kbps   * Bitrate           : 460 Kbps   *
* FPS               : 30         * FPS               : 30         *
*
*****
```

播放效果如下图所示：



【注意】如何进行过 Camera 视频编码测试，则这里会自动播放视频编码文件。

## 7. 3 JPEG 图像解码测试

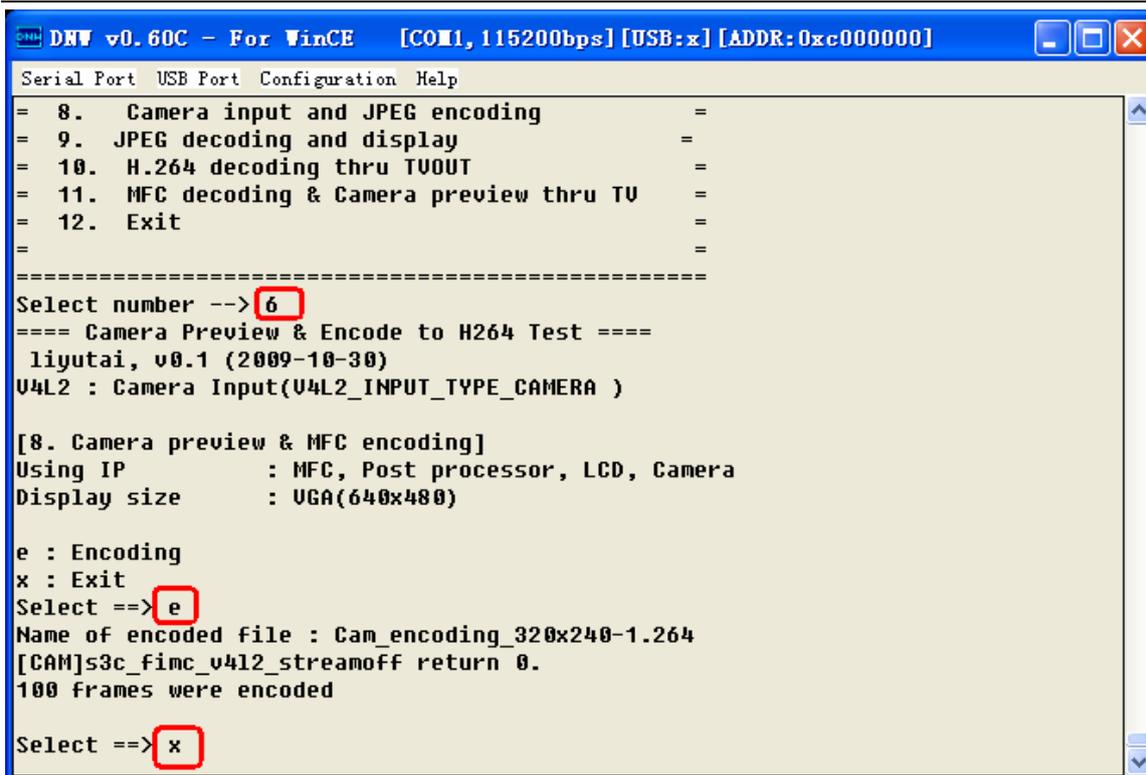
在主菜单下输入 9，进行 JPEG 图像解码测试，液晶屏上显示效果如下图：



**【注意】** 如何进行过摄像头 JPEG 编码测试，则这里会自动显示编码文件 `Cam_capture_640x480-1.jpg`。

## 7. 4 摄像头预览和 H.264 编码测试

用排线连接好 OV9650 摄像头和开发板 CAM 接口，然后启动开发板，进入 Mediatest 测试，在主菜单下输入 6，开始本项测试：



液晶屏上的预览效果如下图：



输入 e，开始 H.264 编码测试，可以尝试变换拍摄内容，直到 100 帧编码完成：

输入 x，退出本项测试。

在主菜单下输入 1，可以播放刚才拍摄并编码完成的 H.264 格式的视频文件 Cam\_encoding\_320x240-1.264。

## 7. 5 摄像头预览和 JPEG 编码测试

用排线连接好 OV9650 摄像头和开发板 CAM 接口，然后启动开发板，进入 Mediatest 测试，在主菜单下输入 8，开始本项测试：

```
DNV v0.60C - For WinCE [COM1,115200bps] [USB:x] [ADDR:0xc000000]
Serial Port USB Port Configuration Help
= 8. Camera input and JPEG encoding =
= 9. JPEG decoding and display =
= 10. H.264 decoding thru TUOUT =
= 11. MFC decoding & Camera preview thru TU =
= 12. Exit =
=====
Select number --> 8
===== Camera Preview & JPEG Encode =====
liyutai, 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
Name of photo file : Cam_capture_640x480-1.jpg
CAPTURE SUCCESS

Select ==> x
```

液晶屏上的预览效果如下图:

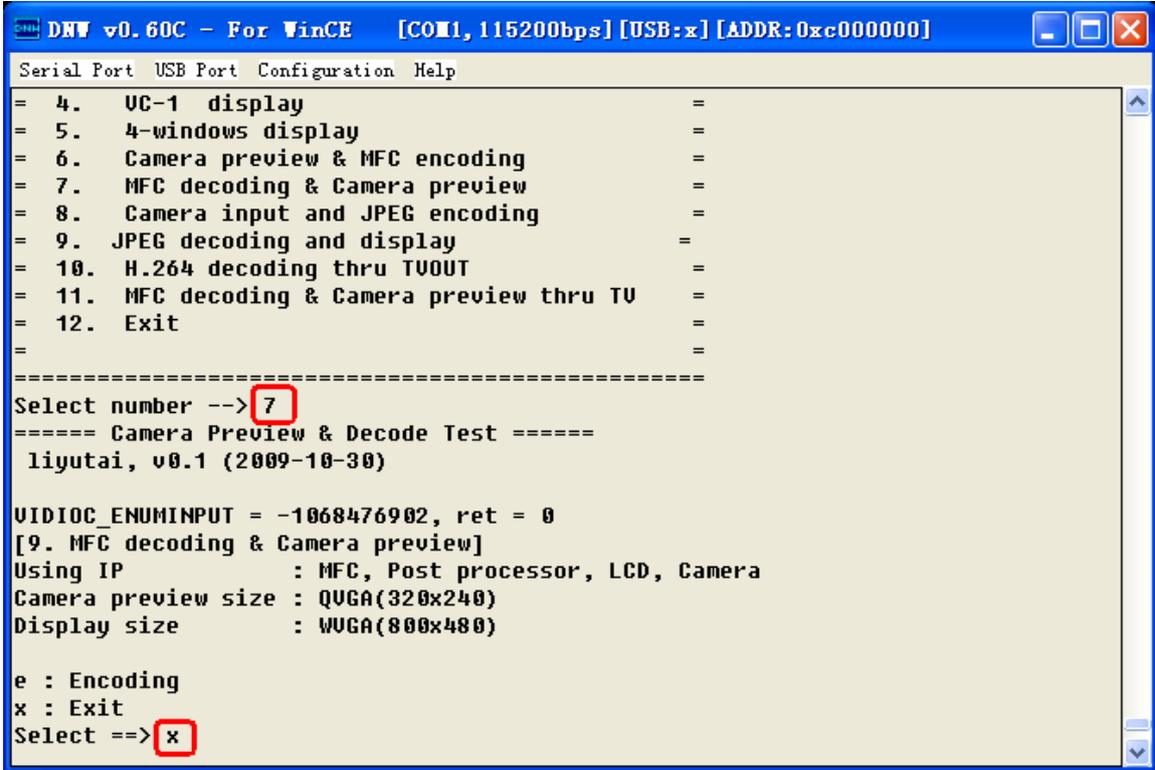


输入 c, 开始获取一张 JPEG 图片, 输入 x, 退出本项测试。

在主菜单下输入 9, 可以显示刚才拍摄并编码完成的 JPEG 格式的图像文件 **Cam\_capture\_640x480-1.jpg**。

## 7. 6 摄像头预览和 MFC 解码

本项测试实现了摄像头预览同时 MFC 解码，操作如下：

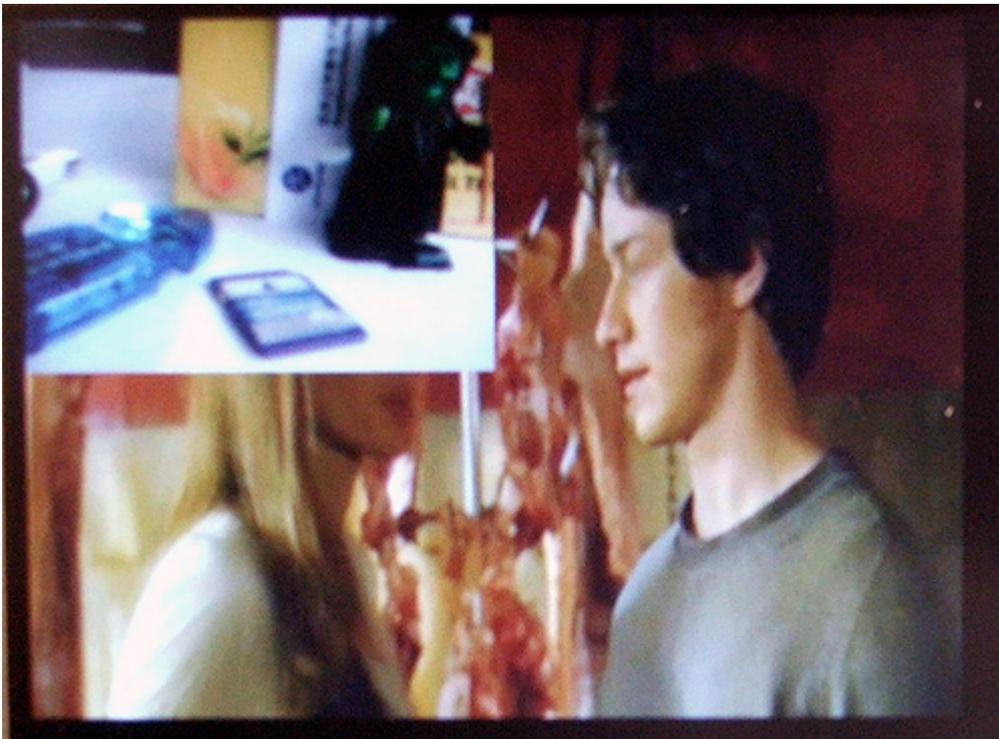


```
DNV v0.60C - For WinCE [COM1,115200bps] [USB:x] [ADDR:0xc000000]
Serial Port USB Port Configuration Help
= 4. UC-1 display =
= 5. 4-windows display =
= 6. Camera preview & MFC encoding =
= 7. MFC decoding & Camera preview =
= 8. Camera input and JPEG encoding =
= 9. JPEG decoding and display =
= 10. H.264 decoding thru TUOUT =
= 11. MFC decoding & Camera preview thru TU =
= 12. Exit =
=====
Select number --> 7
===== Camera Preview & Decode Test =====
liyutai, v0.1 (2009-10-30)

VIDIOC_ENUMINPUT = -1068476902, ret = 0
[9. MFC decoding & Camera preview]
Using IP : MFC, Post processor, LCD, Camera
Camera preview size : QVGA(320x240)
Display size : WUGA(800x480)

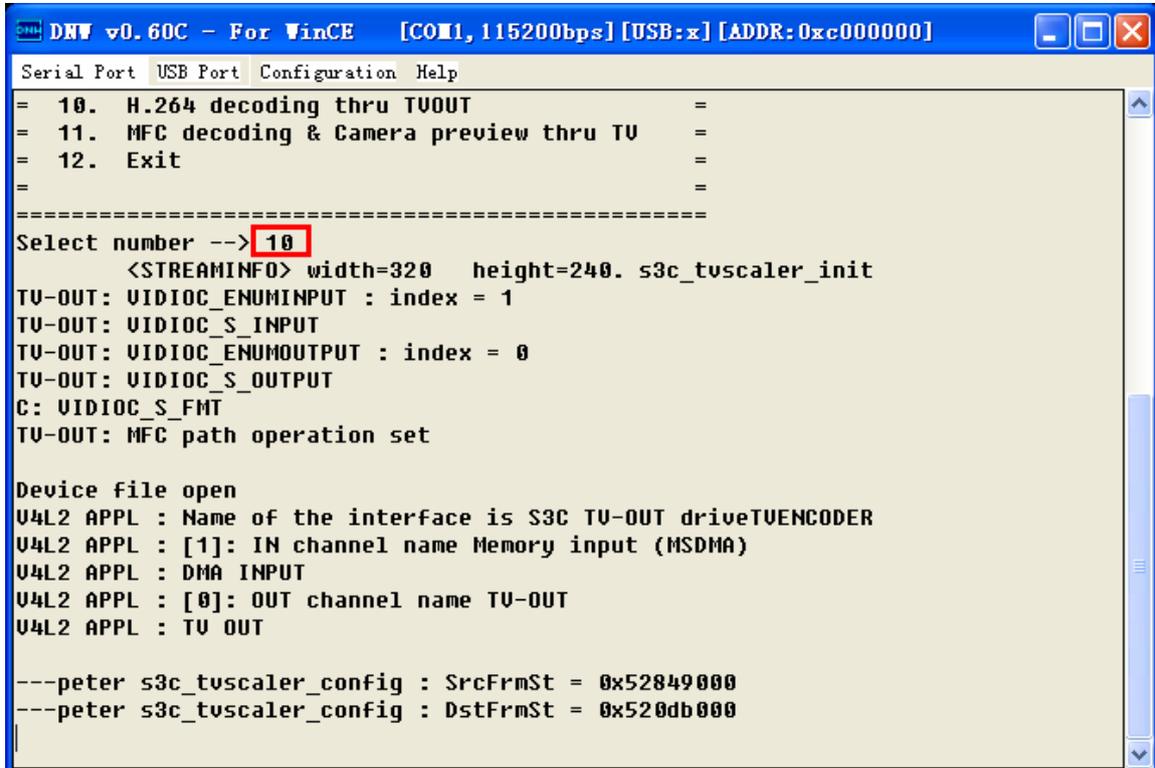
e : Encoding
x : Exit
Select ==> x
```

液晶屏上显示效果如下图：



## 7. 7 视频解码 TVOUT 输出测试

用 AV 连接线连接好 ARMSYS6410 开发板的 RCA 接口和视频播放设备，启动开发板，进入 Mediatest 测试，在主菜单下输入 10，开始本项测试：



```
DNV v0.60C - For WinCE [COM1, 115200bps] [USB:x] [ADDR:0xc000000]
Serial Port USB Port Configuration Help
= 10. H.264 decoding thru TVOUT =
= 11. MFC decoding & Camera preview thru TU =
= 12. Exit =
=====
Select number --> 10
      <STREAMINFO> width=320 height=240. s3c_tvscaler_init
TV-OUT: VIDIOC_ENUMINPUT : index = 1
TV-OUT: VIDIOC_S_INPUT
TV-OUT: VIDIOC_ENUMOUTPUT : index = 0
TV-OUT: VIDIOC_S_OUTPUT
C: VIDIOC_S_FMT
TV-OUT: MFC path operation set

Device file open
U4L2 APPL : Name of the interface is S3C TU-OUT driveTUENCODER
U4L2 APPL : [1]: IN channel name Memory input (MSDMA)
U4L2 APPL : DMA INPUT
U4L2 APPL : [0]: OUT channel name TU-OUT
U4L2 APPL : TU OUT

---peter s3c_tvscaler_config : SrcFrmSt = 0x52849000
---peter s3c_tvscaler_config : DstFrmSt = 0x520db000
```

视频播放结束后自动退出测试。

显示效果如下图：



## 7. 8 摄像头预览和 TVOUT 输出测试

用排线连接好 OV9650 摄像头和开发板 CAM 接口，用 AV 连接线连接好 ARMSYS6410 开发板的 RCA 接口和视频播放设备，启动开发板，进入 Mediatest 测试，在主菜单下输入 11，开始本项测试：

```

DNW v0.60C - For WinCE [COM1,115200bps] [USB:x] [ADDR:0xc000000]
Serial Port USB Port Configuration Help
=====
Select number --> 11
=====Camera Preview & TVOUT Test =====
liyutai, v0.1 (2009-10-30)
s3c_tuscaler_init
TV-OUT: VIDIOC_ENUMINPUT : index = 1
TV-OUT: VIDIOC_S_INPUT
TV-OUT: VIDIOC_ENUMOUTPUT : index = 0
TV-OUT: VIDIOC_S_OUTPUT
C: VIDIOC_S_FMT
TV-OUT: LCD path operation set
U4L2 APPL : Name of the interface is S3C TU-OUT
---peter s3c_tuscaler_config : SrcFrmSt = 0x51cd9000
---peter s3c_tuscaler_config : DstFrmSt = 0x520db000
driveTUENCODER
U4L2 APPL : [1]: IN channel name Memory input (MSDMA)
U4L2 APPL : DMA INPUT
U4L2 APPL : [0]: OUT channel name TU-OUT
U4L2 APPL : TU OUT

[13. MFC decoding & Camera preview]
Using IP : MFC, Post processor, LCD, Camera, TU
scaler/encoder
Camera preview size thru TU : SD(720x480)
Display size thru LCD : WUGA(800x480)

e : Encoding
x : Exit
    
```

## 7. 9 触摸屏设备驱动测试与 tslib 校准

进入 Qtopia 桌面，选择“设置”页面，点击“重校正”项目，选择“yes”重启 Qtopia。这时自动切换进入 tslib 校准界面：



准确点击 5 个校准点，tslib 程序将自动产生校准数据，您可以通过 Qtopia 自带的手写输入法等，

来测试校准效果。

## 7. 10 AC97 音频设备驱动测试

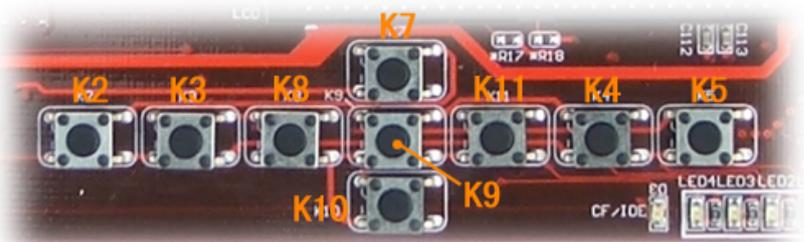
可以使用 Qtopia 自带的播放器进行音频文件（例如 MP3 文件）的播放，进入 Qtopia 桌面，选择“Documents”页面，双击“New Soul”自动打开播放器进入播放状态。ARMSYS6410 开发板默认同时打开了扬声器和耳机，因此您如果购买套餐可以直接听到扬声器播放的效果，如果您购买的是单板，请插入耳机来收听效果。



## 7. 11 键盘驱动测试

在 Qtopia 下可以进行部分按键功能的测试。也可以采用 showkey 工具进行测试：  
按键的定义是：

- K2--F1
- K3--F2
- K4--F3
- K5--取消
- K7--上
- K8--左
- K9--回车
- K10--下
- K11--右



## 7. 12 以太网口驱动测试 (ping)

确保用交叉网线连接好了开发板和主机，在开发板的控制台中设置好 IP 地址（与主机处于同一个网段），然后 ping 主机：

```
[root@armsys6410 /]# ifconfig eth0 192.168.253.12 up
[root@armsys6410 /]# ping 192.168.253.5
PING 192.168.253.5 (192.168.253.5): 56 data bytes
64 bytes from 192.168.253.5: seq=0 ttl=128 time=2.001 ms
64 bytes from 192.168.253.5: seq=1 ttl=128 time=0.334 ms
64 bytes from 192.168.253.5: seq=2 ttl=128 time=0.300 ms
64 bytes from 192.168.253.5: seq=3 ttl=128 time=0.298 ms
^C
--- 192.168.253.5 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.298/0.733/2.001 ms
[root@armsys6410 /]#
```

## 7. 13 ttySAC0~3 串口驱动测试

采用“ARMSYS6410-B 基础配置光盘资料\Linux 相关\linux2.6 标准版\测试实例\uarttest\_v1.0.tgz”进行该项测试，开发板出厂时，已经保存在 mtdblock3 分区中，即目录/mnt/disk/下。

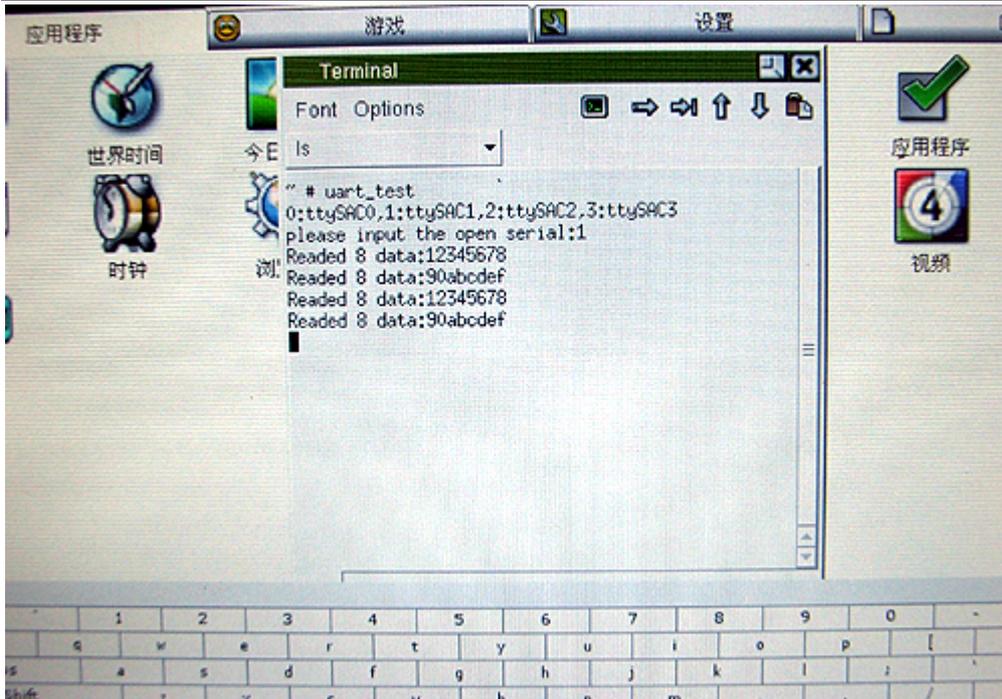
由于现在大部分主机只有一个串行口，为了方便测试，我们使用 Qtopia 中自带的终端软件作为控制台，输入串口测试命令 uart\_test。



点击 Terminal 软件菜单“Font”，选择“Medium Fixed”。

在 Qtopia 桌面左下角选择输入法，选择 Keyboard 方式输入。

输入 uart\_test，并且输入 0~3，选择您要测试的串口，如下图所示：



然后在 PC 端打开串口测试软件（推荐采用 SSCOM3.2），在“字符串输入框”中输入若干字符，并点击发送；开发板的终端上将显示接到的字符串，同时 PC 端测试软件上也会收到程序返回的字符：



## 7. 14 两路 SD/MMC 卡驱动测试

我们采用两张卡来同时测试 ARMSYS6410 的两路 SD 卡通道。一张 Kingston 的 128MB 卡和一张 Kingston 的 2GB 卡，分别插入 SD1 卡座和 SD2 卡座，串口上将显示出两张卡的插入信息：

```
sdhci: card inserted.  
sdhci: card inserted.  
mmc0: new SD card at address 0001  
mmcblk0: mmc0:0001 128MB 119 MiB  
mmcblk0:  
mmc1: new high speed SD card at address dd23  
mmcblk1: mmc1:dd23 SD02G 1.83 GiB  
mmcblk1: p1
```

下面分别挂载两张卡：

```
mkdir /mnt/sd1  
mount /dev/mmcblk0 /mnt/sd1  
mkdir /mnt/sd2  
mount /dev/mmcblk1p1 /mnt/sd2
```

挂载完成后，可以随意做一些读写操作来进行测试。

测试完成后，卸载 SD 卡：

```
umount /mnt/sd1  
umount /mnt/sd2
```

拔出卡片：

```
[root@armsys6410 /]# umount /mnt/sd1  
[root@armsys6410 /]# umount /mnt/sd2  
[root@armsys6410 /]# mmc1: card dd23 removed  
sdhci: card inserted.  
sdhci: card inserted.  
sdhci: card inserted.  
sdhci: card removed.  
mmc0: card 0001 removed
```

## 7. 15 主 USB 接口设备驱动测试

### 7.15.1 USB 鼠标/键盘测试

由于我们已经在内核中添加了对 USB 接口鼠标、键盘的支持，向 USB\_A 插座插入 USB 鼠标，ARMSYS6410 控制台上将出现枚举和驱动的信息，下图是鼠标插入和拔除时显示的信息：

```
usb 1-1: new low speed USB device using s3c2410-ohci and address 6
```

```
usb 1-1: configuration #1 chosen from 1 choice
input: USB Optical Mouse as /class/input/input2
generic-usb 0003:0461:4D15.0001: input: USB HID v1.11 Mouse [USB Optical
Mouse] on usb-s3c24xx-1/input0
拔除鼠标:
usb 1-1: USB disconnect, address 6
```

## 7.15.2U 盘测试

由于我们已经在内核中添加了对 USB 存储设备的支持，插入 U 盘到 USB\_A 接口，ARMSYS6410 控制台上将出现枚举和驱动的信息。下图信息显示了一个 1G 优盘插入的信息：

```
usb 1-1: new full speed USB device using s3c2410-ohci and address 7
usb 1-1: configuration #1 chosen from 1 choice
scsi0 : SCSI emulation for USB Mass Storage devices
scsi 0:0:0:0: Direct-Access      Kingston DataTraveler 2.0 PMAP PQ: 0 ANSI: 0
CCS
sd 0:0:0:0: [sda] 1953792 512-byte hardware sectors: (1.00 GB/954 MiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sd 0:0:0:0: [sda] 1953792 512-byte hardware sectors: (1.00 GB/954 MiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Assuming drive cache: write through
sda: sda1
sd 0:0:0:0: [sda] Attached SCSI removable disk
sd 0:0:0:0: Attached scsi generic sg0 type 0
```

U 盘自动挂载到/udisk 目录下；查看 U 盘内容：

**ls /udisk**

现在就可以开始读写优盘了。

存储设备拔出时，在控制台将看到如下提示：

```
usb 1-1: USB disconnect, address 7
```

# 附录 A：网络服务

## A.1 ifconfig 命令

用 ifconfig 设置 ARMSYS6410 网卡的 IP 地址，例如：

```
ifconfig eth0 192.168.253.12 up
```

## A.2 ping 命令

**ping** 命令可以用来测试本机和网络上的另一台计算机是否连通：

```
[@lyt /]#ping 192.168.253.5 -l 25500
```

表示向 192.168.253.1 发送大小为 25500 字节的数据包，以验证网络是否连接正常，如果连接正常，则结果如下图所示：

```
[root@armsys6410 /]# ping 192.168.253.5 -s 25500
PING 192.168.253.5 (192.168.253.5): 25500 data bytes
25508 bytes from 192.168.253.5: seq=1 ttl=128 time=8.584 ms
25508 bytes from 192.168.253.5: seq=2 ttl=128 time=8.349 ms
25508 bytes from 192.168.253.5: seq=3 ttl=128 time=8.343 ms
25508 bytes from 192.168.253.5: seq=4 ttl=128 time=8.343 ms
25508 bytes from 192.168.253.5: seq=5 ttl=128 time=8.313 ms
25508 bytes from 192.168.253.5: seq=6 ttl=128 time=8.360 ms
^C
--- 192.168.253.5 ping statistics ---
7 packets transmitted, 6 packets received, 14% packet loss
round-trip min/avg/max = 8.313/8.382/8.584 ms
[root@armsys6410 /]#
```

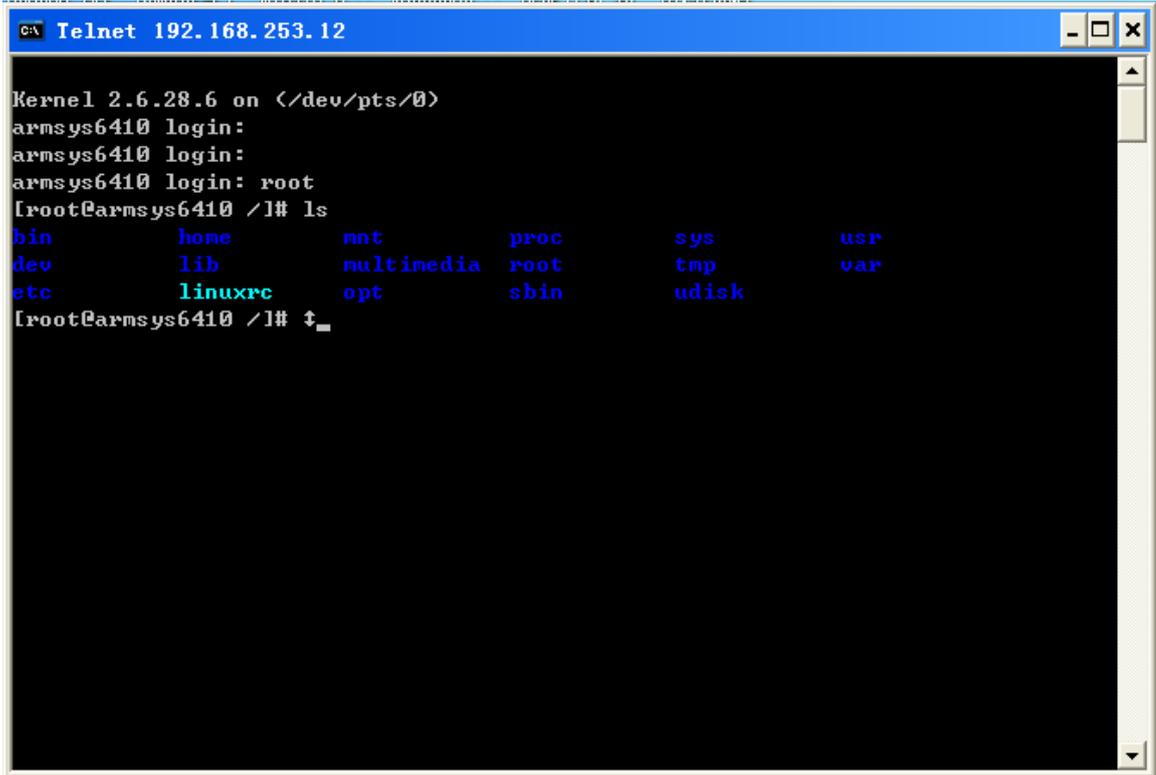
## A.3 Telnet 服务器 inetd

ARMSYS6410 的/etc/init.d/rcS 启动脚本默认已经启动该服务。

当 ARMSYS6410 已经被设置了 IP 地址后，它就可以作为一台 telnet 服务器了。例如 ARMSYS6410 的缺省 IP 地址被设置为 192.168.253.12，则在 PC 主机的终端中敲入以下命令：

```
telnet 192.168.253.12
```

输入用户名“root”，密码为空，即可登录 ARMSYS6410，如下图所示。



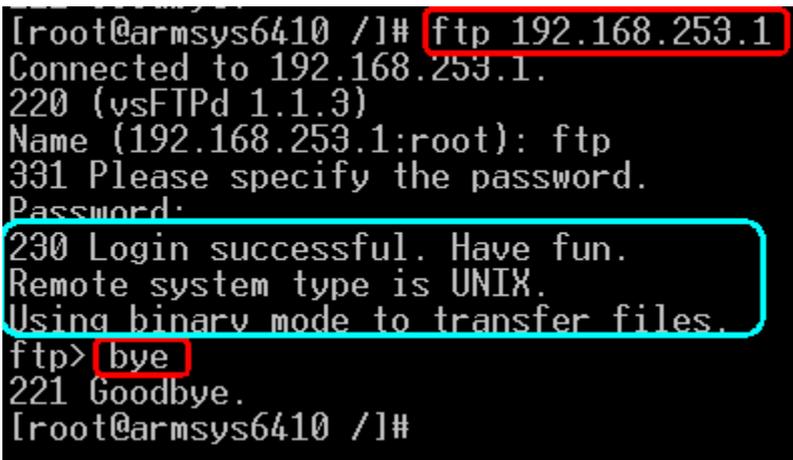
```
C:\ Telnet 192.168.253.12
Kernel 2.6.28.6 on </dev/pts/0>
armsys6410 login:
armsys6410 login:
armsys6410 login: root
[root@armsys6410 /]# ls
bin          home         mnt          proc         sys          usr
dev          lib          multimedia  root         tmp          var
etc          linuxrc     opt          sbin        udisk
```

## A.4 远程文件传送 ftp

ftp 的使用方法与标准 Linux 是相同的, 假定 ARMSYS6410 所在网络中 ftp 服务器 IP 地址是 192.168.253.1, 在 ARMSYS6410 的控制台命令提示符下输入:

```
[@lyt /]# ftp 192.168.253.1
```

按回车后将出现登录提示, 输入正确的用户名及密码后即登录该 ftp 服务器, 使用“ls”命令可以查看服务器当前目录下的文件列表; 使用“bye”命令退出 ftp 服务器, 如下图所示:



```
[root@armsys6410 /]# ftp 192.168.253.1
Connected to 192.168.253.1.
220 (vsFTPd 1.1.3)
Name (192.168.253.1:root): ftp
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bye
221 Goodbye.
[root@armsys6410 /]#
```