AT2440EVB-II 开发板 使用手册 v1.5

广州朗成电子科技有限公司

版权所有©Avantech 2007.9.2 (任何人未经书面同意不得用于商业行为,违者必究)

http://www.avantech.com.cn

 $020{-}87570973 \ 87566943$

内部学习资料 不作商业用途

日 录

<i>—</i> `,	AT2440EVB-II图解4
<u> </u>	
Ξ,	
四、	
	1、ATBOOT 的位置
	2、ITAG 下载板的使用
	3、拨码开关的设置
	4、串口的使用
	5、SJF2440环境设置
五、	ATBOOT启动烧写方法······18
六、	
	2. ATBOOT 菜单说明
七、	用ATBOOT下载WINCE5.0 内核NK.bin到Nand Flash中
八、	
九、	 Platform Builder 5.0 安装注意事项47
+、	
+-	
$+$ _	二、 <u>WINCE5.0使用指南(介绍相关例子使用)</u>
	1. MP4 播放器
	2. USB 摄像头测试
	3. Camera 摄像头测试
	4. SD 卡
	5. 保存文件
	6. USB HOST 接口
	7. 网络使用
	8. 触摸屏校正
	9. 录音
	10. 显示界面旋转 90 度
	11. GPS 接收器的使用
	12. USB 同步
	13. 添加你的程序到 NK 中
	14. 存储管理器
	15. 调整内存空间
	16. 增加启动时个性化 LOGO
	17. USB 无线网卡驱动安装编译
$\pm \Xi$	E、 <u>Linux2.6内核系统介绍</u> ······98
十四	4、Linux2. 6. 18. 2 光盘目录介绍······103
$+\Xi$	ī、 <u>用ATBOOT下载Linux2.6.18.2方法</u>
十六	K、LINUX2. 6. 18. 2 640*480 800*480 分辨率校正方法 ····································
++	117 <u>Linux开发环境 (NFS) 的快速搭建</u>
十八	(、 <u>基于Busybox的根文件系统制作</u>
十九	L、Linux内核的编译及烧写·······124

二十	、 <u>u-boot.bin启动文件的烧写方法</u> ······128
二十	一、 <u>内核用UB00T方式自动烧写方法</u>
二十	二、Linux2.6.18.2的部分应用······144
二十	三、 <u>Linux基本知识</u>
二十	四、 <u>各种应用介绍(WINCE与LINUX)</u> ······151
	24.1 WinCE 中如何向 image 中添加文件
	24.2 建立 HIVE 注册表
	24.3 应用程序中对注册表的编辑
	24.4 添加输入法
	24.5 删除一个注册表健值
	24.6 改变桌面背景图片
	24.7 WinCE 中支持多国语言切换
	24.8 实现开机后 WINCE 的任务栏为隐藏模式
	24.9 应用程序在 WinCE 启动后自动运行
	24.10 向 WinCE 内核添加文件
	24.11 应用程序在 WinCE 启动后自动运行
	24.12 WinCE 配置文件中的路径
	24.13 WinCE 字库与内核分离设置
	24.14 向 WinCE 的注册表添加其他注册表文件
	24.15 WinCE 内核 NK.nb0 大小设置
	24.16 WinCE 中如何编译生成 NK. nb0
	24.17 WinCE 电源管理
	24.18 TCP/MP 源代码编译
	24.19 WinCE 挂起和唤醒(suspend/wakeup)
	24.20 WinCE 应用程序开发(基于 EVC)
	24.21 QT 开发--Hello
	24.22 QT 开发--信号与插槽
	24.23 Linux 应用程序开发--串口通信

24.24 Linux 应用程序开发——Hello

一、AT2440EVB-II 图解

AT2440EVB-II 开发板是由核心板+底板结构方式的,核心板是六层板,尺寸为 74x31mm,集成了 S3C2440A 400MHz CPU、64M SDRAM、2M NOR FLASH、64M NAND FLASH,采用四排双行插针 1.27mm 间距扩 展方式,是目前公开结构最小的核心板之一,这种结构也是目前最可靠的扩展方式,对作为产品应用不可 缺少的一部分。

底板是四层板,尺寸为110x80mm,集成三个串口(一个五线制,两个三线制),100M网卡,USB主从口,音频输入输出,SD卡,红外线,引出I2C接口等,将三星S3C2440A CPU的接口标准完全扩展出来。 所有布线均采用专业设计,完全符合主板高速运行的生产工艺。



AT2440EVB-II 开发板整体图



AT2440EVB-II 开发板接口定义图 广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II



AT2440EVB-II 背面图

二、AT2440EVB-II 硬件说明

2.1 硬件配置

核心板硬件配置如下:

- ◆ 处理器: Samsung S3C2440A, ARM920T 内核, 主频 400MHz
- ◆ 2M Bytes NorFlash, 采用 AMD AM29LV160DB 芯片
- ◆ 64M Bytes NandFlash, 采用 Samsung K9F1208UOB 芯片
- ◆ 64M Bytes SDRAM,采用两片三星 K4S561632H-UC75 芯片 底板硬件配置如下:
- ◆ 一个 100Mbps 以太网接口,采用 Davicom DM9000AE 芯片
- ◆ 一个 USB HOST 接口,遵循 USB1.1 协议
- ◆ 一个 USB DEVICE 接口,遵循 USB1.1 协议
- ◆ 音频输入\输出接口各一个,采用 Philips UDA1341TS 芯片,遵循 IIS 总线协议,可同时放音录音
- ◆ 一个 SD 卡接口, 支持 1G 容量的 SD 卡
- ◆ 一个 5 线制串口,两个 3 线制串口,采用两片 Sipex SP3232EEA 芯片
- ◆ 一个红外接口,采用 IRMS6542 红外收发器
- ◆ 一个 50 针 LCD 和触摸屏接口,包含 4 线触摸屏信号和 LCD 控制器的所有信号
- ◆ 一个摄像头接口,支持 OV7660 摄像头模块
- ◆ 一个 64 针数据、地址总线接口
- ◆ 一个 44 针 GPIO 接口,包含 A/D 输入信号、SPI 总线及其它所有未使用的 I/O 接口
- ◆ 两个用户 LED
- ◆ 一个 6 针 JTAG 接口
- ◆ 四个用户自定义按键
- ◆ 一个复位按键,采用 MAX811T 复位芯片
- ◆ 一个电源开关,采用 AMS1084-3.3 和 AMS1085CD 电源芯片
- ◆ 一个 5V 电源接口

2.2硬件接口详细说明

2.2.1 以太网接口

板上集成一个 100M 以太网接口,如果和 PC 直接相连请用配件中的交叉网线,如果通过交换机或路 由器连到局域网里,请自备一条直连网线。在使用 AT244EVB-II 开发板进行开发的过程中,以太网接口可 以用来下载 WinCE 镜像、Linux 内核和根文件系统等。

2. 2. 2 USB HOST 接口

USB HOST 接口支持 USB1.1 协议,可以接 U 盘、USB 鼠标、USB 键盘、USB HUB、USB 摄像头、USB 无线网卡等等。

2. 2. 3 USB Device 接口

USB Device 接口支持 USB1.1 协议,可以在 WinCE 下与 PC 同步(前提是 PC 上要安装微软提供的同步软件 Activesync 与驱动),这样在 PC 端就可以直接通过 UBS 电缆和开发板传送数据。

2. 2. 4 音频输入\输出接口

音频输入和输出采用 IIS 总线的接口标准,音频输入接口可以支持录音,音频输出可以支持各种声音 广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II 6 的播放,支持同时录音放音。

2. 2. 5 SD 卡接口

SD卡接口可以支持1G容量的SD存储卡,WINCE5.0下即插即用。

2.2.6 串口

开发板上带有 3 个串口,一个 5 线制串口,两个 3 线制串口。通过两个标准 DB9 接口引出两个串口。 一个 10 针 2.0mm 间距双排插针引出了 3 个串口的所有信号,这些信号都是 RS232 电平。

2.2.7 红外接口

红外接口采用 IRMS6452 红外收发器,开发板可以通过红外收发器与外部进行数据传输,采用的是第3路串口。



2.2.8 LCD 和触摸屏接口

图 2.2.8 LCD 和触摸屏接口

LCD 接口引出了 AT2440EVB-II 开发板的 LCD 控制器的所有信号,可以支持 STN 液晶屏和 TFT 液晶 屏,同时还引出了 4 线的触摸屏控制信号。开发套件标准配置的 LCD 为 Sharp 3.5' TFT 液晶屏,分辨率为 240×320,该型号 LCD 带有 4 线触摸屏。LCD 的具体型号为 LQ035Q7DH01、LQ035Q7DH02 或 LQ035Q7DH04。同时还可选购 SHARP 8'TFT 液晶屏 (640*480),日立 7'TFT 液晶屏 (800*480),这几个 分辨率两个系统都配有相应的驱动,如用户要配置其它型号的 LCD,请自行修改相关驱动程序。

2.2.9 摄像头接口

WinCE 下的摄像头驱动支持 30 万像素的 OV7660 模块,如用户需要配置其它型号的摄像头模块,请 自行修改相关驱动程序。此部分的测试程序只对接口验证是否有效,对于所看到的测试效果不佳等不负责。 同时并没有这部分相关的源代码提供,只提供相关驱动作参考使用。

2. 2. 10 数据和地址总线接口





总线接口包括 16 位的数据总线和 27 位的地址总线,以及读写信号、片选信号和复位信号等,用户可 以通过总线接口方便地扩展各种外总线接口设备。

2. 2. 11 GPIO 接口



图 2.7 GPIO 接口

GPIO 接口引出了所有未使用的 I/O 信号,包括 A/D 输入信号、TTL 电平的串口信号、中断信号、时 钟输出信号、DMA 信号、PWM 信号等,用户可以通过 GPIO 接口方便地扩展各种外部 I/O 接口设备。

2. 2. 12 JTAG 接口



图 2.8 JTAG 接口

用户可以 JTAG 下载头通过 JTAG 接口来下载程序到 NorFlash 或 NandFlash,由于 JTAG 下载头下载 程序的速度很慢,所以只适合下载一些像 bootloader 这样的小程序或 ADS 编译出来的一些应用程序,操作

系统的镜像文件不适合通过 JTAG 下载头来下载,而要通过 USB 口或以太网接口来下载。JTAG 接口还可以接 ARM 仿真器,配合 ADS 开发环境来调试一些用户程序。

下载 u-boot.bin, ATBoot.bin 使用 JTAG 下载头,将配有一条 6PIN 的线接到 J1 的座上,请对好 1 脚插上开发板的 JTAG 位置。



开发板与仿真器连接,将 JTAG 下载头的 J3 位置四个短接帽全部取下来,即可使用于仿真器的转换使用。

2. 2. 13 拨码开关





开发板上有一个四位的拨码开关,OM0 和 OM1 的配置如下:

OM[1:0]	启动方式	
00:	Nand boot	——NandFlash 启动
01:	Halfword (16-bit)	——NorFlash 启动(默认启动方式)
10:	Word (32-bit)	
11:	Test Mode	
NADN_WP 信号	是 NandFlash 的写保护住	言号,Nand_CON 是 NandFlash 的配置信号。

ATBOOT 的 USB 下载方式,将开关1和2、4 拨到 "ON"方向的位置,开关3在 "OFF"的位置。

三、AT2440EVB-II 软件功能配置

- 配有 Linux2.6.18.2 版本 (QT:2.3.10)、Wince5.0 版本的系统,提供相应系统源代码;
- LCD 液晶驱动真彩屏分别有: 夏普 3.5'LCD, 夏普 8'LCD, 日立 7'LCD (导航专用宽频液晶屏);

3.1、WINCE5.0 软件功能介绍:

- ▶ 提供 USB 无线网卡驱动、USB 200 万像素摄像头驱动、CAM 手机专用 30 万像素摄像头驱动。(注: 此部分只提供驱动与测试 DEMO,没有源代码;只负责设备能正常使用,不负责二次开发)
- ▶ 三个串口可直接使用,COM0 己支持 GPS 接收器使用,特别提供 USB 下载方法,非常适合做产品者使用。
- 具有时钟、IP地址、PC同步选项修改后断电可保存的注册表问题;增加数据保存区 ResidentFlash, 断电后数据不会掉失。国内其它厂家的同类产品很少有提供这部分功能。
- ▶ 增加全新 ATBOOT.bin 的 USB 自动下载功能,增加系统分区功能。
- ▶ 增加 BSP 中个性 LOGO 显示,方法简单容易,可以动手制作自己的 LOGO。
- ▶ 音频支持放音与录音,可同时进行放录功能。
- ▶ USB HOST 支持外接 USB HUB 扩展,即插即用。
- ▶ USB DEV 支持 USB 下载系统功能。
- ▶ 网口支持 10/100M 接口,具有过滤功能,不需要在同一局域网中手动分配 MAC。
- ▶ SD 卡即插即用。

3.2、Linux2.6.18.2 软件功能介绍:

- ▶ 己推出 ATBOOT USB 口方式下载 LINUX2.6 内核到 Nand flash 的启动方法(可自动跳过 FLASH 坏道功能),可节省 Nor flash。
- ▶ 音频支持放音录音功能。
- ▶ USB HOST 支持外接 USB HUB 扩展,即插即用。
- ▶ USB DEV 支持 USB 下载系统功能。
- ▶ 网口支持 10/100M 接口,具有过滤功能,不需要在同一局域网中手动分配 MAC。
- ▶ SD 卡即插即可挂用。
- ▶ 四个按键提供测试驱动。

四、ATBOOT 环境设置

4.1、ATBOOT 的位置

ATBOOT.bin 目前己升级到 V4.02 版本,全面支持通过 USB 口方式下载 Wince5.0, Linux2.6 两个系 统的内核的 bootloader。路径位于光盘根目录下\ATBOOT\,分三种不同分辨率,根据自己的 LCD 来选择。

4.2、JTAG 下载板的使用

ATBoot.bin 文件烧写方法是使用我们配的 JTAG 下载板, JTAG 下载板并口接到 PC 电脑的并口,并 要安装并口驱动。J1 是六针的接口,用我们配好的 6PIN 白色线插到 AT2440EVB-II 开发板上 JTAG 位置, 请注意第一脚要一一对应,带有三角号的为第一脚。如果出现检查不到 CPU 的错误,请检查 JTAG 的 6PIN 线是否插好,一般就是接触不良引起的,请压紧开发板的 JTAG 部分。

4.3、拨码开关的设置

AT2440EVB-II 开发板上的 SW2 位置是四位拨码开关,分为 1、2、3、4 档,在上面还标有 ON 的方向。ATBoot.bin 是使用 JTAG 下载板来烧写到 AT2440EVB-II 开发板上 Nand flash(K9F1208U0B)中,是 Nand Flash 启动,请将 1,2 与 4 两档跳到 ON 位置;3 在反方向(OFF)才能看到启动信息。



4.4、串口的使用

ATBoot.bin 是使用 COM1 进行通讯,使用我们配的交叉串口线将开发板的 COM1 与 PC 电脑的串口 连接。开发板上带有 3 个串口,一个 5 线制串口,两个 3 线制串口。通过两个标准 DB9 接口引出两个串口。一个 10 针 2.0mm 间距双排插针引出了 3 个串口的所有信号,与 DB9 的两个是相同,这些信号都是 RS232 电平。在板上己有 COM0,COM1 是标准 DB9 接口。自行通过修改驱动与注册表可实现三个串口都可使用。

4.5、SJF2440 环境设置

位置:使用 SJF2440 下载板需要先安装其驱动,驱动路径:AT2440EVB-II\工具软件\GIVEIO。 SJF2440.exe 要在 WindowsNT/2000/XP 上直接把并口作为 I0 口访问,需要安装 GIVEI0.SYS,下面以 广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II 12 WindowsXP 为例介绍如何安装:

4.5.1、以 administrator 身份登录系统。

4.5.2、复制 giveio.sys 到 C:\WINDOWS\system32\drivers\的目录下。

4.5.3、打开"控制面板",选择"添加/删除硬件"。(注:需要切换到经典视图 才出现"添加/删除硬件" 选项)



4.5.4、 选择下一步



4.5.5、选择己连接硬件,下一步

添加硬件向导
硬件连接好了吗?
悠已经将此硬件连接到计算机了吗? ④ <u>是,我已经连接了此硬件(Y)</u> ④ 否,我尚未添加此硬件(<u>H</u>)
< 上一步 (2) 下一步 (2) > 取消
4.5.6、选择"添加新的硬件设备",下一步
漆加硬件向导 通常加速 通常加速 通常加速 通常 ないない しょう しょうしょう しょう
以下硬件已安装在您的计算机上
从以下列表,选择一个已安装的硬件设备,然后单击"下一步",检查属性或 解决您遇到的问题。

要添加列表中没有显示的硬件,请单击"添加新的硬件设备"。

已安装的硬件 (20):	
😋 USB Root Hub	<u> </u>
🗬 USB Root Hub	
🗬 USB Root Hub	
🗬 USB Root Hub	
添加新的硬件设备	
	_
	< 上一步(18) 下一步(18) > 取消
	· · · · · · · · · · · · · · · · · · ·

4.5.7、选择"安装我手动从列表选择的硬件(高级)",下一步

这个向导可以帮助您安装其他硬件 这个向导可以搜索其他硬件并为您自动安装。或者,如果您知道要安装哪个型号的硬件,您可以从列表选择。 您期望向导做什么? ① 搜索并自动安装硬件(推荐)(S) ④ 逻装我手动从列表选择的硬件(高级)(0) (上一步(B)) 取消

4.5.8、选择类型为"端口 (COM 和 LPT)",下一步

添加硬件向导	1.2.10103
从以下列表,选择要安装的硬件类型	
如果看不到想要的硬件类别,诸单击"显示所有设备"。 常见硬件类型(d):	
 ◆SCSI和 RAID 控制器 ●Windows 便携设备 ●磁带驱动器 ◆打印机 ●调制解调器 ●端口 (COM 和 LPT) >到多串ロ卡 ● 红外线设备 	
< 上一步 (B) 下一步	图) 取消

4.5.9、选择"从磁盘安装(H)…",在弹出的窗口中选择 GIVEIO. inf。



4.5.11、添加成功后,会出现下列型号选择,然后下一步

漆加硬件向导	五氢加加
选择要为此硬件安装的设备驱动程序	
子 诸选定硬件的厂商和型号,然后单击"下一步"。 对 程序的磁盘,诸单击"从磁盘安装"。	口果手头有包含要安装的驱动
型묵 giveio	
於 这个驱动程序没有经过数字签署! 告诉我为什么驱动程序签名很重要	从磁盘安装(H)
< 上一步 (2)	下一步(20) > 取消

4.5.12、两次下一步后,会弹出确认窗口,选择"仍然继续"

添加硬件向导		
向导正在安装软件,	请稍候	
	硬件安装	
正在安装支持软件	<u>.</u>	正在为此硬件安装的软件: giveio 没有通过 Windows 徽标测试,无法验证它同 Windows XP 的相容性。(<u>告诉我为什么这个测试很重要。</u>) 继续安装此软件会立即或在以后使系统变得不稳定。 Licrosoft 建议您现在停止此安装,并同硬件供应商 联系,以获得通过 Tindows 数标测试的软件 .
		仍然继续 (C) 「停止安装 (S)」
		<上一步(12)下一步(12)> 取消

4.5.13、按照提示操作直到安装完毕

4.5.14、安装是否成功,请到"系统属性--设备管理器--端口"中,展开+,会看到有 giveio(COMx),x 代表串口多少。

五、ATBOOT 启动烧写方法

硬件连接:

将并口线一头接到 PC 主机的打印口 (并口),另一头接到 JTAG 下载板上。JTAG 下载板上的 J1 位置是 6PIN 的接口,使用我们配的白色 6PIN 线按方面一头接上,另一头接在开发板上的 JTAG 口。请注意此条的 1 脚要一一对应,通常我们会在对应的 1 脚上画了一条黑色线了。

5.1 软件命令:

5.1.1 首先将 ATBoot. bin 拷贝到 SJF2440. exe (SJF2440. exe 文件己略作升级,但本文中的图片显示 的部分内容就不再更新,以实际文件为准。)所在目录中,将开发板上 SW2 拨码开关的 1、2、4 跳到 "ON" 的位置,3 跳到 "OFF" 位置。一般我们在出厂时己设置好了,然后按如下方式进行烧写,先将 ATBoot. bin 烧写到 Nand Flash 的 Block0.

在开始--运行 的窗口中输入 CMD, 打开命令提示符, 进入 SJF2440. exe 所在目录后, 输入"sjf2440 /f:ATBoot4.02-35. bin",回车; 目前我们配有三个不同分辨率的 ATBoot, 4.20 为 ATBOOT 版本号, 后面 数字主要是与分辨率显示有关, 如-35 代表 3.5' LCD 型号, 请根据你的 LCD 分辨率选择 ATBoot. bin。 ATBoot. axf 是我们提供给使用仿真器环境下的使用的文件。

ATBoot 文件的路径: AT2440EVB-II\ ATBOOT\ ATBOOT4.02-35.bin: 为 SHARP3.5'LCD 系统使用的 ATBoot。 ATBOOT4.02-7.bin: 为日立七英寸 LCD 系统使用的 ATBoot。 ATBOOT4.02-8.bin: 为 SHARP8'LCD 系统使用的 ATBoot。



回车后,下图显示 JTAG 小板与开发板正常连接的情况,如果连接不好或接触不良,将会显示找不到 CPU,请检查 JTAG 小板的第1脚与开发板 JTAG 位置上的第1脚是否一致,或者检查 JTAG 小板是否有问题。 同时注意 JTAG 小板上的 J3 位置上四个短接帽的情况,如果 J3 位置上的四个短接帽都插上,是用于 JTAG 下载使用;

如果 J3 位置上的四个短接帽已取下来,是用于仿真器与开发板的转换作用。仿真器的 20PIN 插在 J2 广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II 18

上,J1 接到开发板上 JTAG。

5.1.2 然后选择 "0", 回车; 再选择 "0", 回车

🔤 C:\WINDOWS\system32\cmd.exe - sjf2440 /f:atboot.bin - 🗆 × D:\AT2440\USB>sjf2440 /f:atboot.bin SEC JTAG FLASH(SJF) v 0.1 (\$3C2440X & \$MDK2440 B/D) Usage: SJF /f:<filename> /d=<delay> > S3C2440X(ID=0x0032409d) is detected. [SJF Main Menu] 0:K9S1208 prog 1:28F128J3A prog 2:AM29LV800 Prog 3:Memory Rd/Wr 4:Exit Select the function to test:0 [K9S1208 NAND Flash JTAG Programmer] K9S1208 is detected. ID=0xec76 0:K9S1208 Program 1:K9S1208 Pr BlkPage 2:Exit Select the function to test :0 [SMC(K9S1208V0M) NAND Flash Writing Program] Source size:0h~aa53h Available target block number: 0~4095 Input target block number:_

5.1.3 在提示"Input target block number:"时,输入"0",回车,下面开始烧写 ATBoot.bin,等待烧 写完成。

🛤 C:\WINDOWS\system32\cmd.exe - sjf2440 /f:atboot.bin	- 🗆	×
D:\AT2440\USB>sjf2440 /f:atboot.bin		
++		
SEC JTAG FLASH(SJF) v 0.1	-	
++		
Usage: SJF /f: <filename> /d=<delay></delay></filename>		
> S3C2440X(ID=0x0032409d) is detected.		
[SJF Main Menu]		
0:K9S1208 prog 1:28F128J3A prog 2:AM29LV800 Prog 3:Memory Rd/Wr 4:Exit		
Select the function to test:0		
[K9S1208 NAND Flash JTAG Programmer]		
K9S1208 is detected. ID=0xec76		
0:K9S1208 Program 1:K9S1208 Pr BlkPage 2:Exit Salact the function to test :0		
[SMC(K9S1208V0M) NAND Flash Writing Program]		
Source size:Oh~aa53h		
Aussishie tanget block surbant GM400E		
Input target block number: 0 4075 输入0,将程序烧到第0块位置		•

以下图显示的烧写是 ATBOOT.bin 不带 LOGO 部分,烧写只有三行,目前配的都带有 LOGO,烧录过程大概要几分钟时间,文件比较大些,请注意等待。 5.1.4 烧写完成后输入"2"退出。

C:\WINDOWS\system32\cmd.exe	- 🗆 🗙
[SJF Main Menu] Ø:K9S1208 prog 1:28F128J3A prog 2:AM29LV800 Prog 3:Memory Rd/Wr 4:Exit	
Select the function to test:0	
[K9S1208 NAND Flash JTAG Programmer] K9S1208 is detected. ID=0xec76 0:K9S1208 Program 1:K9S1208 Pr BlkPage 2:Exit Select the function to test :0	
[SMC(K9S1208V0M) NAND Flash Writing Program]	
Source size:Oh~aa53h	
Available target block number: 0~4095 Input target block number:0 target start block number =0	
target size (0x4000×n) =0xc000 STATUS:Eppppppppppppppppppppppp	
Еррррррррррррррррррррррррррррррр Ерррррр	
Select the function to test :2 选择2退出	
D:\AT2440\USB>	-

至此己烧写完毕。将交叉串口线接到 AT2440EVB-II 的 COM1 位置,打开 DNW,设置串口,波特率,连接后,打开电源即可看到启动信息。

六、ATBOOT 功能使用说明

6.1. DNW USB 驱动的安装

6.1.1 将开发板设置成 Nand Flash 启动方式(默认的启动方式)。连上串口线(注意我们配的为交叉线),串口线是连接到 COM1。

6.1.2 打开 DNW.exe(在工具软件目录中,需要复制到硬盘中使用),DNW 的使用方法见下面说明。首先 打开 Configuration>>Options,选择波特率与 COM 口,见下图。

	B:x] 🔤 🗖 🔀
Serial Port USB Port Configurat	ion Help
Options	
	2

6.1.3 选择 Baud Rate 是 <u>115200(</u>必选)与 COM Port(根据你 PC 主机的串口选择)后,按 OK。

Serial Fort USB Fort Configuration Help	-
UART/USB Options Serial Port Baud Rate COM Port © 115200 © COM 1 © 57600 © COM 2 © 38400 © COM 3 © 19200 © COM 4 © 9600 USB Port Download Address 0xc000000	

6.1.4 连接到开发板上, Serial Port>>Connect, 按下图进行。

DNW v0.50A [COM:x][USB:x]	
Serial Port USB Port Configuration Help	
Connect	<u>^</u>
Transmit	
	×

6.1.5 将 USB 线接到 USB 从口, 另一头接到你的电脑上, 连接到开发板成功后, 会出现相关参数, 见下图:

DNV v0.	50A [C	:0 1 3, 115200b	ps][USB:x]
Serial Port	USB Port	Configuration	Help

上图结果是 DNW 下的 USB 驱动还没有安装,如果此驱动己安装成功了,会出现下图的 USB:OK 提示。

I DRV v0.	50 <u>A [</u> (CO Ⅲ 3, 115200b	ps][USB:OK]
Serial Port	USB Port	Configuration	Help

6.1.6 DNW 驱动安装:

在第一次使用时必须先安装驱动,除非你使用过其它厂家的开发板已安装了此驱动,就不需要另安装。 DNW 的 USB 驱动路径: AT2440EVB-II\工具软件\USB 驱动\使用 DNW 的 USB 驱动 6.1.6.1、请将"使用 DNW 的 USB 驱动/secbulk.sys 复制到 windows/system32/driver 目录下"。

特别提示一:

如果没有安装过 DNW 下的 USB 驱动,同时需要在下列两种情况下操作,才会在电脑的任务栏中,出现找到新硬件设备的提示,然后按下面的步骤进行安装驱动。

A: 第一次自己烧写完 ATBOOT 时,启动时会自动停止在下图的"USB host is not connected yet.",此时按下键盘的空格键即可进入菜单选项,同时如果 USB 线己连接在开发板与 PC 主机,就会提示发现新硬件。 B: 如果你的开发板在我们出厂时己烧好系统,包括 WINCE5.0,LINUX2.6,此时打开开发板的电源,启动时不会自动进入到下面菜单选项,操作方法需要先"按着键盘的空格键一直不放开,再打开开发板的电源, 直到你在 DNW 窗口中看到下列菜单选项时,即可放开空格键"。此时 USB 线连接好的情况下,才会提示发现新硬件。

如下两个图的介绍。



此时插入 USB 电缆线(PC 端为 USB host, AT2440 端为 USB Device), PC 电脑端将会搜索到下图的硬件。

特别提示二:如果你的电脑此驱动没有安装成功,但同时又没有下图提示,请到 <u>windows/system32/driver</u>, 删除你己有的 <u>secbulk.sys</u>,再将光盘的 <u>secbulk.sys</u>复制到此目录。在插入 USB 线的同时将设备管理器中的 通用串行总线控制器列表中有黄色警告叹号的驱动删除,再重新插入 USB 线即可发现新硬件。



6.1.6.2、在弹出的窗口中,选择下面一项<u>否,暂时不(T)</u>,然后按<u>下一步</u>。



< 上一步(B) 下一步(B) >

要继续,请单击"下一步"。

取消

6.1.6.4、从浏览中选择 DNW 的 USB 驱动 所在位置,然后下一步。见下面两个图:

找到新的硬件向导
请选择您的搜索和安装选项。
 ● 在这些位置上搜索最佳驱动程序(S)。 使用下列的复选框限制或扩展默认搜索,包括本机路径和可移动媒体。会安装找到的最佳驱动程序。 ● 搜索可移动媒体(软盘、CD-ROM)(M) ● 在搜索中包括这个位置(D): D:\ce5\使用DNW的USB驱动 ● 浏览(B) ● 不要搜索。我要自己选择要安装的驱动程序(D)。 选择这个选项以便从列表中选择设备驱动程序。Windows 不能保证您所选择的驱动程序与您的硬件最匹配。
< 上一步 (B) 下一步 (B) > 取消

青选择您的搜索和安装选项。		
● 在这些位置上搜索最佳驱动程.	<mark>浏览文件夹</mark> 选择包含您的硬件的驱动程序的文件夹。	?
使用下列的复选性限制或扩展; 到的最佳驱动程序。	⊞ 🥪 SYSTEM (C:)	
✓ 在搜索中包括这个位置(D:\ce5\使用DXW的USB驱	□ → DATA (D:) □ 26 □ 12440evb	
 ○ 不要搜索。我要自己选择要安 选择这个选项以便从列表中选; 动程序与您的硬件最匹配。 	 	
	, ~ ~ ~ 要查看任何子文件夹,请单击上面的 + 号。	

6.1.6.5、在出现在驱动列表中,选择 <u>secbulk.inf</u>此项,然后 <u>下一步</u>,见下图:

找到新的硬件向导				
从下列表中选择与您的硬件(的最佳四	un: -		
SEC SOC Test Bo	ard			
描述	版本	制造商	位置	
SEC SOC Test Board	未知	SEC, SYSTEM LSI	c:\windows\inf\secbulk.inf	
SEC SOC Test Board	未知	SEC, SYSTEM LSI	c:\windows\inf\oem10.	
<		1111	>	
▲ 这个驱动程序没有名 告诉我驱动程序签署自	2 过数字 : 的重要性	签署!		
		〈上一步(B)下一步(B) > 取消	

6.1.6.6、此时弹出是否继续的窗口,选择 仍然继续,见下图。

找到新的硬件向导	
向导正在安装软件,请稍候	
硬件安 SEC SOC Test Bo	装 正在为此硬件安装的软件: SEC SOC Test Board 没有通过 Windows 徽标测试,无法验证它同 Windows XP 的相容性。(告诉我为什么这个测试很重要。) 继续安装此软件会立即或在以后使系统变得不稳定。 Licrosoft 建议您现在停止比安装,并同硬件供应商 联系,以获得通过 Tindows 谢标测试的软件。
	仍然继续 (2) 停止安装 (3)

6.1.6.7、有时会出现自动找不到 secbulk.sys 的文件,此时会弹出窗口,手动选择此文件的位置,见下图:

6.1.6.8、上图 按确定后,出现下面窗口:

新的硬件向]류
导正在安装	教件,请稍候
所需文件	+
	需要 USB Downloader Installation Disk for SEC SOC 上的文件 'SECBULK.sys'。 取消 输入文件所在的路径,然后单击"确定"。
	文件复制来源 (C): ☑
	[<上→步(26)][下→步(26) >] [取消

6.1.6.9、从浏览中选择 secbulk.sys 所在位置(此文件位于"使用 DNW 的 USB 驱动"中)。

找到新	的硬件向导					
向令	正在安装软件,	诸稍候				
	所需文件				×	
	查找文件					? 🔀
	查找范围(I):	🛅 使用มพหย่าน	SB马区运力	· 01	1 📂 🖽 -	
	 我最近的文档 夏面 没面 我的文档 我的文档 夏面 夏回 夏回	in usb. 2k				
		文件名 (M):	secbulk		~	打开(0)
		文件类型 (T):	SECBULK sys; SECBULK sy_		~	取消

6.1.6.10、打开"使用 DNW 的 USB 驱动>>secbulk.sys"。

所需文件	+
	需要 USB Downloader Installation Disk for SEC SOC 上的文件 'SECBULK.sys'。 和) 输入文件所在的路径,然后单击"确定"。 文件复制来源(C):
	D:\ce5\使用DNW的USB驱动 🔽 浏览(B))

6.1.6.11、从上图中 确定 后,会继续完成驱动的安装。



此时 USB 的驱动安装完成,从上面介绍的[USB:OK]可以确定安装成功。

如果 USB 电缆连接经常检测不了,请插拨多几次,有时接触不良会导致这个问题的。

6.2. ATBOOT 菜单说明

版本号: v4.02, 更新日期: 2007-12-3

新增内容:增加 LINUX 内核 USB 口下载方式,增加自动跳过 NAND FLASH 坏道功能。 V3.02 增加下载到 SDRAM 的调试功能。



6.2.1 功能简介

Select Menu

[0] Download & run to SDRAM WinCE5.0) //下载 NK.BIN 到 SDRAM 中运行,断电后不保存
[1] Download Only	//手动下载 NK.BIN
[2] Write File to NAND Flash	//将 NK.BIN 写入 NAND FLASH,手动
[3] Erase NAND Flash Partition	//擦除 NAND FLASH, 注意会删除 ATBOOT
[4] Auto Flash WinCE5.0	//自动引导下载 NK.BIN,是我们使用的下载方式
[5] MemoryTest	//内存测试,不建议自己使用
[6] Config parameters	//参数配置,不建议自己使用
[7] Start OS	//手动启动当前所下载系统
[8] Auto Flash Linux2.6	//自动引导下载 Linux2.6 内核与文件系统

6.2.2 在此我们只介绍以下三个选项:

[0] Download & run to SDRAM WinCE5.0	//下载 NK.BIN 到 SDRAM 中运行,断电后不保存
[4] Auto Flash WinCE5.0	//自动引导下载 NK.BIN,是我们使用的下载方式
[8] Auto Flash Linux2.6	//自动引导下载 Linux2.6 内核与文件系统

1)选择 0, [0] Download & run to SDRAM WinCE5.0,在调试编译好的 NK.BIN 时,为了测试是否正常但 又不想经常去擦除 FLASH,减少 FLASH 的寿命,同时此项不会擦除原来所烧录好的内核,可以选择这项。 输入 0,会出现: Download&Run is selected.

USB host is connected. Waiting a download.

然后从 DNW 中的 USB Port 选择你所要下载的 NK.BIN 文件即可,接着系统会自动下载,解压,运行 WINCE5.0 系统。注意此功能对 LINUX2.6 内核不支持。

2)选择 4, [4] Auto Flash WinCE5.0,此功能我们己设置好自动写入的地址,分配方式等。子菜单包括三个选项的内容:
0: offset 0x0 , size 0x40000 [ATBOOT] //可在线通过 USB 口更新 ATBOOT.BIN
1: offset 0x40000 , size 0x2300000 [WINCE] //下载 NK.BIN
2: offset 0x2340000 , size 0x1cc0000 [USER] //用户使用区,需要与 SMFLASH 驱动配合使用

各段地址分配如下: ATBOOT 0[~]0x40000 WINCE 0x40000[~]0x02340000 USER 0x02340000[~]0x04000000

3)选择 8, [8] Auto Flash Linux 2.6, 此功能我们已设置好自动写入的地址,分配方式等。子菜单包括三个 选项的内容:

0: offset 0x0 , size 0x40000 [ATBOOT] //可在线通过 USB 口更新 ATBOOT.BIN, 与 4 项相同。 1: offset 0x40000 , size 0x1c0000 [KERNEL] //下载 zimage 的内核, 分不同分辨率版本的内核

2: offset 0x200000 , size 0x3500000 [ROOTFS] //下载 rootfs.img 文件系统

各段地址分配如下:

ATBOOT 0~0x40000 KERNEL 0x40000 ~ 0x240000

ROOTFS 0x240000~0x4000000

6.2.3 USB 方式更新 ATBOOT

如果第一次使用 JTAG 下载成功 ATBOOT,以后需要更新或重烧 ATBOOT.BIN,可以通过 USB 下载方式快速更新不同分辨率的 ATBOOT.BIN 文件。

按空格键 进入下面菜单时,选择 4([4] Auto Flash),再选择 0 (0:offset 0x0 , size 0x20000 [ATB00T]), 然后确定选择 Y,即可通过 USB 口更新 ATB00T.BIN。

注: 4 与 8 项的 "0: offset 0x0 , size 0x40000 [ATB00T]" 这项功能是一样的,选择哪个都可以更新。

₩ DHT =0.50Å [CON4,115200bps][USB:0K]	
Serial Port USB Port Configuration Help	
http://www.avantech.com.cn	▲
<*************************************	
++	
AT2440EVB USB Downloader 2007-12-3	
++	
USB. IN ENDPOINT.1 OUT ENDPOINT.3	
FORMAT: <addr(data):4>+<size(n+10):4>+<data:n>+<cs:2></cs:2></data:n></size(n+10):4></addr(data):4>	
NOTE: 1. Power off/on or press the reset button for 1 sec	
in order to get a valid USB device address.	
2. For additional menu, Press any key.	
USB host is not connected yet.	
###### Select Menu ######	
[0] Download & run to SDRAM WinCE5.0	
[1] Download Only	
[2] Write File to NAND Flash	
[3] Erase NAND Flash Partition	
[4] Auto Flash WinCE5.0	
[5] MemoryTest	
[6] Config parameters	
[/] Start US	
[8] HULO FIASH LINUX2.0	
	Ţ

选择 4([4] Auto Flash)出现下列三个选项



再选择 0 (0: offset 0x0 , size 0x20000 [ATBOOT]) Are you sure to erase nand flash from page 0x0, block count 0x10 ? [y/n]

然后确定选择 Y,即可象下载 NK. BIN 方法一样,通过 USB 口更新 ATBOOT. BIN。Erase Nand partition completed success

The temporary download address is 0x31000000.

USB host is connected. Waiting a download.

- 🗆 🗵 Serial Port USB Port Configuration Help USB host is not connected yet. * ###### Select Menu ###### [0] Download & run to SDRAM WinCE5.0 [1] Download Only [2] Write File to NAND Flash [3] Erase NAND Flash Partition [4] Auto Flash WinCE5.0 [5] MemoryTest [6] Config parameters [7] Start OS [8] Auto Flash Linux2.6 Read chip id = ec76 Nand flash status = c0 Please select which region to erase : Esc to abort 1 : offset 0x0 , size 0x40000 2 : off [ATBOOT] 1 : offset 0x40000 , size 0x2300000 [WINCE] 2 : offset 0x2340000 , size 0x1cc0000 [USER] Are you sure to erase nand flash from paqe 0x0, block count 0x10 ? [y/n]Erase Nand partition completed success The temporary download address is 0x31000000. USB host is connected. Waiting a download.

通过 USB Port—Transmit。

™ DNT v0 . 50	IA [CO∎4, 11	5200bps][VSB:0K]	
Serial Port	USB Port Conf	iguration Help	
USB host i	Transmit	cted yet.	
	Rx Test		
###### 261	Status	#### - CDDAW W-055 A	
[[0] Dowur	ogg a ran c	O SDRHW WINCES.0	
נחשטע נרן ן	oad uniy		
[2] Write	FILE TO NH	NV Flash Reutitie	
[3] Erase	NHNV Flasn	Partition	
[4] AUTO	Flash WinCE	5.0	
[5] Memor	ylest .		
[6] Confi	g parameter	S	
[7] Start	05		
[8] Auto	Flash Linux	2.6	

打开所要升级的 ATBOOT. BIN

DHT 👓. 50A	[CO II 4, 115200b	ps][USB:OK]			
Serial Port USB	Port Configurat	tion Help			🗄 🖶 📮
USB host is r	at connected	unt			
	1177				<u> </u>
[0] Dompjosc	查找范围(I):	C USB		- 🗢 🖻 🖻	· 📰 -
[1] Download		LTDOOT4 OO T	-	_	
[2] Write Fi		ATBOOT4. 02-8			
[3] Erase Nf	我最近的文档	ATBOOT4.02-0	5		
[4] Auto Fla			•		
[5] Memoryle					
[7] Start OS	桌面				
[8] Auto Fla					
Read chip id					
Nand flash st	我的文档				
Please select					
1 : offset By	我的曲脑				
2 : offset 0					
Are you sure					
[y/n]	网上邻居				
The temperature					
The cemporary		1			
USB host is c		文件名 (M):	ATB00T4.02-7	-	打开(0)
		文件类型 (<u>T</u>):	BIN Files (*. bin; *. nb0)	•	
		_	,		

升级成功,内容如下:

- 🗆 × Serial Port USB Port Configuration Help Now, Downloading [ADDRESS:31000000h,TOTAL:43378] * RECEIVED FILE SIZE: 43378 (517.0KB/S,0.1S) Now, Checksum calculation Download O.K. Invalid BIN file signature! 45,0,0,ea,1b,0,0 Read chip id = ec76 Nand flash status = c0 Now write nand flash page 0x0 from ram address 0x31000000, filesize = 43378 ... Program nand flash partition success ####### Select Menu ####### [0] Download & run to SDRAM WinCE5.0 [1] Download Only [2] Write File to NAND Flash [3] Erase NAND Flash Partition [4] Auto Flash WinCE5.0 [5] MemoryTest [6] Config parameters [7] Start OS [8] Auto Flash Linux2.6

注意:有时会更新失败,失败后只能用 JTAG 下载板去重烧 ATBOOT. BIN,方法如上述介绍。

七、用 ATBOOT 下载 WINCE5.0 内核 NK.bin 到 Nand Flash 中

将内核下载到 Nand Flash 中固化了,断电后不会掉失,启动后系统运行很稳定,这正是其优点。 7.1 AT2440 开发板通电后在输出"*"时要及时按<u>空格键</u>才能启动 ATBOOT 的菜单功能,否则会直接启动 WINCE5.0 或 LINUX2.6 系统。第一次烧写 ATBOOT 后系统启动自动进入到下载窗口,此时按空格键即可进入菜单选项。

ATBOOT 启动后如下图所示:(以下为 V4.02 的图解,目前是 V4.02 版本,见后面相关图解。)

₩ DBT v0.50A [C0H4,115200bps][USB:x]	
Serial Port USB Port Configuration Help	
s3c2440x WinCE ATBOOTLOADER v4.02 Dec 03 2007 00:30:20	-
 在此位置按 空格键 才能进入下列可调试菜单状态 Loading s3c2440x WinCE BOOT 	
FCLK=399MHz, HCLK=133MHz, PCLK=66MHz, CPU is running at 399MHz UPLL=47MHz, UCLK=47MHz DTUN UPLL0	
MPLLVal [M:6eh,P:3h,S:1h] CLKDIVN:7h	
AVANTECH BOOTLOADER v4.02 , Model 240*320 LOGO http://www.avantech.com.cn	
<pre></pre>	
H12440EVB USB DOWN10ader 2007-12-3 ++	
FCLK=399.7MHz,DMA mode	
USB: IN_ENDPOINT:1 OUT_ENDPOINT:3	
FORMATE CADDR(DATA):47+CST2E(H+T0):47+CDATA:H7+CCS:27 NOTE: 1. Power off/on or press the reset button for 1 sec	
in order to get a valid USB device address.	
2. For additional menu, Press any key.	
USB host is connected. Waiting a download. 需要按 空格键 才出现下列菜单选I	<u> </u>
###### Select Menu ######	
[0] Download & run to SDRAM WinCE5.0	
[1] Download Only [2] Weite File to NOND Flach	
[2] WRITE FILE TO NHAD FLASH [3] Frase NAND Flash Partition	
[4] Auto Flash WinCE5.0	
[5] MemoryTest	
[6] Config parameters	
[7] Start OS	-

7.2 从上图中,选择"4"后即出现下图分区类型:

0 分区是分别保存 ATBOOT,由于我们是采用 JTAG 方法分别烧写到相应的分区,此处不需要进行操作。 如果你配有仿真器,可以通过仿真器进行不同的方法烧到此分区,在此不介绍具体的方法。 1 分区是下载 WINCE5.0 的内核 NK 文件保存区。

2 分区是用户自定义数据区,我们在 BSP 系统中己添加了此功能,在第一次启动系统时将会停止一分钟时 广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II 35 间进行分区处理,进入到 CE 界面后,在 WINDOWS 目录中会出现此分区。

Serial Port USB Port Configuration Help	
http://www.avantech.com.cn	
<*************************************	
++	
S3C2440A USB Downloader ver R0.03 2004 Jan	
FCLK=399.7MHz,DMA mode	
USB: IN ENDPOINT:1 OUT ENDPOINT:3	
FORMAT: <addr(data):4>+<size(n+10):4>+<data:n>+<cs:2></cs:2></data:n></size(n+10):4></addr(data):4>	
NOTE: 1. Power off/on or press the reset button for 1 sec	
in order to get a valid USB device address.	
2. For additional menu, Press any key.	
USB host is not connected yet.	
###### Select Menu ######	
[1] Download Only	
[2] Write File to NAND Flash	
[3] Erase NAND Flash Partition	
[4] Auto Flash	
[5] MemoryTest	
[6] Config parameters	
Read chip id = ec76	
Nand flash status = cV	
rlease select which region to erase : Esc to abort	
U: UTTSEL UXU , SIZE UXZUUUU [HIBUUI] 1 · offcot By200000 cizo By22000000 [HIBUUI]	
2 · offset 0x2320000 _ size 0x2300000 [WINGE] 匹件I烷与U	
	Ļ

7.3 在上图位置输入"1",出现提示是否清除分区内容,选择Y:


此时正在等待擦除分区的 FLASH,完成后见下图。

		×
Serial Port USB Port Configuration Help		
[3] Erase NAND Flash Partition		
[4] Auto Flash		
[5] MemoryTest		
[6] Config parameters		
Kead Chip id = ec/o Nand flach status - c0		
Please select which region to erase · Esc to abort		
0 : offset 0x0 . size 0x20000 [ATBOOT]		
1 : offset 0x20000 , size 0x2300000 [WINCE]		
2 : offset 0x2320000 , size 0x1ce0000 [USER]		
Are you sure to erase nand flash from page 0x100, block count 0x8c0 ? [y/n]		
Arte dels distativos a super sus escara ele		
等付消除NAND FLASH中		
		-

👥 DI	T vi	D. 5(DA	[C	0 84, 115200bp:	s][USB:OK]												. 🗆	×
Seria	l Po	ort	USB	Por	t Configurati	on Help													
														• • •	 	• • •		• • •	
			• • •			• • • • • • • • • •		• • • •				• • • •		• • •	 	• • •	• • •	• • •	
			• • •		• • • • • • • • • • • • • • • •	• • • • • • • • • •		• • • •	• • • •	• • • •		• • • •		• • •	 	• • •	• • •	• • •	
			• • •	'				• • • •	• • • •	• • • •			•••	•••	 	• • •	• • •	•••	
			• • •										• • •		 	• • •		• • •	
			• • •			• • • • • • • • • •		• • • •		• • • •		• • • •	• • •	• • •	 	• • •	• • •	• • •	
			• • •			• • • • • • • • • •		• • • •	• • • •	• • • •		• • • •		• • •	 	• • •	• • •	• • •	
													• • •		 				
			• • •										• • •	• • •	 	• • •	• • •	• • •	
			• • •		• • • • • • • • • • • • • • • •	• • • • • • • • • •		• • • •	• • • •	• • • •		• • • •	•••	• • •	 	• • •	• • •	• • •	
			• • •	'		• • • • • • • • • •		• • • •	• • • •	• • • •				•••	 	• • •	• • •	•••	
Eras	e N	land	t pa	arti	tion compl	eted succ	ess	#	察除	NAND) FL	ASH	[完]	龙					
The	tem	ipoi	rary	y da	wnload add	ress is O	x310000)00. [*]	411/414				101						
		_																	
USB	hos	ti	LS (conr	nected. Wai	ting a do	wnload.	- 4	爭待	ト载	NK.	BIN							
																			•

此时等待下载内核 NK.BIN 文件。

特别提示:

上述擦除 NAND FLASH 过程中,如果 NAND FLASH 有坏道,会在擦除过程中提示,要求按"Press any key to continue",按任何键跳过坏道后即可继续擦除。目前市场上的 NAND FLASH 都存在坏道的问题,这个是厂家允许的工作范围内的。

7.4 选择"USB Port"菜单下的 Transmit -> Transmit, 选择要烧写的 NK.bin, 注意是 NK. bin 而不是 NK. nb0。





打开后,会自动下载 NK 文件,此时出现下载进度,见下图:

■DBT =0.50A [CON4,115200bps][USB:0K] _ 🗆 🗙 Serial Port USB Port Configuration Help Downloading C:\TINCE500\PBWorkspaces\AT24CE5S35\Rel... 🗙 Erase Nand partition completed success The temporary download address is 0x31000000. USB host is connected. Waiting a download. Now, Downloading [ADDRESS:31000000h,TOTAL:28453929] RECEIVED FI 2555904 下载完成后,出现下面窗口: INV v0.50J - For WinCE [COM4, 115200bps] [USB:OK] [ADDR: 0xc000000] Serial Port USB Port Configuration Help The temporary download address is 0x31000000. ~ USB host is connected. Waiting a download. Now, Downloading [ADDRESS:31000000h,TOTAL:29178529] RECEIVED FILE SIZE:29178529 (893.9KB/S,32.6S) 下载完NK后,即自动出现此部分信息。如没有出现, Now, Checksum calculation 需要重新开始下载。 Download O.K. Found single BIN signature. Image start is 0x80200000, length is 0x01c2a28c REC start 0x80200000, length 0x4 REC start 0x80200000, length 0x4 REC start 0x80200040, length 0x8 REC start 0x80200048, length 0x4 REC start 0x80201000, length 0x6600c REC start 0x80268000, length 0x2098 REC start 0x8026b000, length 0x78d48 REC start 0x8026b000, length 0x666h 此部分自动解压NK REC start 0x802e3d48, length 0x6564 REC start 0x802eb000, length 0x34088 REC start 0x80320000, length 0xac8b4 REC start 0x803cc8b4, length 0x984c REC start 0x803d7000, length 0x8 REC start 0x803d8000, length 0x903

注意:此时只下载到 SDRAM 中,还没有直接写入到 NAND Flash 中,请等待系统自动将 NK 解压到 NAND

Flash 中, 然后回车, 见下图:

Serial Port USB Port Configuration Help	
dllfirst 0x01e001ea, dlllast 0x02000000	_
physfirst 0x80200000, physlast 0x81d74100	
REC start 0x81d72018, length 0x20e8	
real entry is uxauzzcuaa, pnysical address is uxauzzcuaa	
Nand flash status = c0	
.Save boot params success.	
Read chip id = ec76 解压后自动将NK烧到NAND FLASH中	
Nand flash status = c0	
Now write nand flash page 0x100 from ram address 0x30200000, filesize = 2878	6944
	· · · · ·
	·
	· · · · ·
	· • • • • •
从上图中需要等待几十秒时间,烧写成功后,会出现下面窗口:	

DH DH	v 0.5	A Oi	[CO	4, 115200bps][USB:OK]	
Seria	l Port	USB	Port	Configuration	Help	
						· · · · · ·
					• • • • • • • • • • • • • • • • • • • •	
				•••••		
••••						
					•••••••••••••••••••••••••••••••••••••••	•••••
				•••••		
	• • • • •	• • • •				
						Pr
ogra	m nan	d fl	ash	partition s	^{ICCESS} 烧写完毕,请按开发板的RESET键重启即	开始
####	## Se	lect	Men	u ######	启动WINCE5 0系统,在启动中系统全进行	元初
[1]	Down	load	l Onl	y		1 1/3
[2]	Writ	e Fi	le t	o NAND Flas	如化,分区等,有时会停止一分钟时间。 ————————————————————————————————————	
[3]	Eras	e NA	IND F	lash Partit	lon 请注意不要关电源。	
[4]	Auto	Fla	sh			
[5]	Memo	ryfe	st	atouc		
[0]	CONF	ra b	arali	leters		
						`

见上面提示部分,代表本次内核己烧写成功。

7.5 重启开发板,按 RESET 键,系统即可自动进入到 WINCE5.0 系统启动中,第一次烧写的系统,将会出现自动初始化,分区等操作,在"FMD_OEMIoControl: unrecognized IOCTL (0x71c24)."此步将停止几十秒之内,请不要关闭电源等,初始化完成后将会自动启动进入 WINCE5.0 界面。成功启动 WINCE5.0 的窗口界面如下:

很多新用户不没有详细看说明,以为停止在此处是系统出错,导致他关了电源或重启了,请注意。

Serial Port USB Port Configuration Help	
AVANTECH BOOTLOADER v1.0 Jul 17 2007 16:20:54 *****	
Read chip id = ec76 Nand flash status = c0 Load Kernel 系统启动中	
**************************************	****
run 0x3022c088 Windows CE Kernel for ARM (Thumb Enabled) Built on Jun 24 2004 at 18:25:00 ProcessorType=0920 Revision=0 sp_abt=ffff5000 sp_irq=ffff2800 sp_undef=ffffc800 OEMAddressTable = 8022bf44 DCache: 8 sets, 64 ways, 32 line size, 16384 size ICache: 8 sets, 64 ways, 32 line size, 16384 size Sp=ffffc7cc DEMIsControl: Unsupported Code 0x10100bk - device 0x0101 func 45	F
OEMICControl: Unsupported Code 0x101008c - device 0x0101 func 35	
FMD::FMD_Init FMD::FMD_Init - pBSPArgs->nfsblk = 0x0 FMD::FMD_Init - READ_REGISTER_BYTE(pNFSBLK) = 0x0 FMD::FMD_Init IPL FMD::FMD_Init Done	
MID = 0xec, DID = 0x76 4th Cycle : 0xa5	
MOMBLOCKS : 512(0x200), SECTORSPERBLOCK = 256(0x100), BYTESPERSECTOR = 512 (0x200) FMD_OEMIoControl: unrecognized IOCTL (0x71c24). 始化,分区等,请不要关电法	刃 _

DET v0.50A [COM4, 115200bps][USB:x] - 🗆 × Serial Port USB Port Configuration Help [dm9: Tries to map io space with 18000000 * [dm9: The mapped address is 2D0000 [dm9: Chip signature is 0A469000 <DM9:--MiniportInitialize> CamClockOn = 1 Camera::Block TX Ended... CAMERA.DLL::InterruptThread Initialized. CAMERA: CIS_Open CIS: IOCTL POWER CAPABILITIES CAMERA: CIS_Open CAMERA: CIS Close DeviceFolder::LoadDevice!Enumerate Found deprecated load instructions at (Drivers\BuiltIn\AFD). Driver cannot be unloaded. OEMIoControl: Unsupported Code 0x10100d0 - device 0x0101 func 52 OEMIoControl: Unsupported Code 0x10100f8 - device 0x0101 func 62 DeviceFolder::LoadDevice!Enumerate Found deprecated load instructions at (Drivers\BuiltIn\PPP). Driver cannot be unloaded. OEMIoControl: Unsupported Code 0x10100fc - device 0x0101 func 63 Lyg.p: Layout Manager successfully initialized to 2 Maximum Allowed Error 7: MSIM: IM ReadRegistry read KB 5 CamClockOn = 0 Explorer(V2.0) taskbar thread started. NDISPWR:: Found adapter [DM9CE1] 启动完成

以上代表系统己成功启动, USB 方法下载 NK 成功。

注:此时如果 USB 线还插在 AT2440 上,在 PC 电脑上会弹出一个找到新硬件设备的提示。此提示是 WINCE 窗口下 AT2440 作为一个移动设备来处理,如需要使用 USB 同步的方法,请安装 "wince 下与 PC 同步的 USB 驱动"。

八、用 ATBOOT 下载 WINCE5.0 内核 NK.bin 到 SDRAM 中调试

从 ATBOOT v3.02 起我们己完善了将 NK. bin 只下载到 SDRAM 调试的功能,方便各位在调试 NK 时,不 需要写入到 NAND Flash 中,减少对 NAND Flash 的擦除,提高其使用寿命,同时缩短了开发者的调试时间。

下载到 SDRAM 中运行的内核系统,不会对己固化好的内核造成冲突,也不会删除固化好的内核系统的, 但在 SDRAM 中运行的系统,在断电或重启后就掉失了,不会保存你所设置的内容,只方便调试使用。

8.1、首先烧好 ATBOOT. BIN 后,打开电源马上按空格键进入下面窗口。选择 0。

DHT ▼0.50Å [COH4,115200bps][VSB:0K]	
Serial Port USB Port Configuration Help	
http://www.avantech.com.cn	▲
<*************************************	
++ AT2440EVB USB Downloader 2007-12-3 ++	
FCLK=399.7MHz,DMA mode	
USB: IN_ENDPOINT:1 OUT_ENDPOINT:3	
FORMAT: <addr(data):4>+<size(n+10):4>+<data:n>+<cs:2></cs:2></data:n></size(n+10):4></addr(data):4>	
NOTE: 1. Power off/on or press the reset button for 1 sec	
in order to get a valid USB device address.	
2. For additional menu, Press any key.	
USB host is connected. Waiting a download.	
###### Select Menu ######	
[0] Download & run to SDRAM WinCE5.0	
[1] Download Only	
[2] Write File to NAND Flash	
[3] Erase NAND Flash Partition	
[4] Auto Flash WinCE5.0	
[5] Memorylest [6] Config powomotows	
[0] CONTLY PARAMETERS	
[8] Auto Flach Linux2 6	

8.2、 连接好 USB 线, 等待下载内核。

Serial Port USB Port Configuration Help	
++	
FCLK=399.7MHz,DMA mode	
USB: IN_ENDPOINT:1 OUT_ENDPOINT:3	
FORMAT: <addr(data):4>+<size(n+10):4>+<data:n>+<cs:2></cs:2></data:n></size(n+10):4></addr(data):4>	
NOTE: 1. Power off/on or press the reset button for 1 sec	
in order to get a valid USB device address.	
2. For additional menu, rress any key.	
Neofawoh e paiting a download	
osb host is connected. Naiting a downioda.	
###### Select Menu ######	
[0] Download & run to SDRAM WinCE5.0	
[1] Download Only	
[2] Write File to NAND Flash	
[3] Erase NAND Flash Partition	
[4] Auto Flash WinCE5.0	
[5] MemoryTest	
[6] Config parameters	
[/] Start US	
[8] HUCO FIASH LINUX2.0	
Download&Run is selected	
Pownroddunun ry Screeced.	
USB host is connected. Waiting a download.	
	-

8.3、通过 USB Port>>Transmit 选择 NK. bin 下载



8.4、下载完后系统在判断所下载的内核类型,因为目前的版本 v4.02 增加了 WINCE 与 LINUX 两个系统的 下载方式,比原来的多一个系统,所以多了一步验证方式。

🚾 DHT ₩0.50Å [COH4,115200bps]	[USB:OK]	<u>- 0 ×</u>
Serial Port USB Port Configuration	. Help	
in order t 2. For additiona	to get a valid USB device address. 1 menu, Press any key.	
USB host is connected. Wait:	ing a download.	
####### Select Menu #######		
[0] Download & run to SDRA	M WINCE5.0	
[1] VOWNICAO UNIY [2] Wwite Eile te NAND Elev	- h	
[2] WILLE FILE LU MHND FIA: [3] Eraco NOND Elach Parti	50 tion	
[4] Auto Flash WinCF5_A		
[5] MemoruTest		
[6] Config parameters		
[7] Start OS		
[8] Auto Flash Linux2.6		
Download&Run is selected.		
USB host is connected. Wait:	ing a download.	
Now, Downloading [ADDRESS:3	1000000h,TOTAL:32875389]	
RECEIVED FILE SIZE:32875389		
(870.4KB/5,37.55)	下载完后系统需要判断下载的内核类型因为目前增加	
now, checksum carculation	WINCE与LINUX两个系统,比原来多了一个系统,从而多了此步。	

8.5、自动对 NK. BIN 进行解压,需要等待些时间。

	[VSB:OK]	
Serial Port USB Port Configuration	Help	
REC start 0x8078e000, length	0x41a4	_
REC start 0x80793000, length	0x7178	
REC start 0x8079b000, length	0xd 0b8	
REC start 0x807a9000, length	0x1028	
REC start 0x807ab000, length	0x1038	
REC start 0x807ad000, length	0x1088	
REC start 0x807af000, length	0x3fffc	
REC start 0x807ef000, length	0x19fdc	
REC start 0x80809000, length	0x4278	
REC start 0x8080e000, length	0x2eac	
REC start 0x80811000, length	0x20b8	
REC start 0x80814000, length	0x13c4	
REC start 0x80816000, length	0x78	
REC start 0x80817000, length	0x1088	
REC start 0x80819000, length	0x1414	
REC start 0x8081b000, length	0x78	
REC start 0x8081c000, length	0x30f0	
REC start 0x80820000, length	0x5138	
REC start 0x80826000, length	0x2f20c	
REC start 0x80856000, length	0x38138	
REC start 0x8088f000, length	0x4000	
REC start 0x80893000, length	0x4b558	
REC start 0x808de558, length	0xb67c	
REC start 0x808ea000, length	0x632c	
		•

8.6、随后就进入到 WinCE5.0 界面,下图为启动完成的内容。

DNW v0.50A [COM3,115200bps][USB:x]	
Serial Port USB Port Configuration Help	
<pre>(66608640). I2C Init IIC IRQ mapping: [IRQ:27->sysIRQ:20]. <davicom 4.2="" 5.0="" dm9000="" driver="" for="" v3.2.7="" wince=""> <dm9:++miniportintialize> [dm9: Tries to map io space with 18000000 [dm9: The mapped address is 2D0000 [dm9: Chip signature is 0A469000 <dm9:miniportinitialize> CamClockOn = 1 Camera::Block TX Ended CAMERA.DLL::InterruptThread Initialized. CAMERA: CIS_Open CIS: IOCTL_POWER_CAPABILITIES</dm9:miniportinitialize></dm9:++miniportintialize></davicom></pre>	
CAMERA: CIS_Open CAMERA: CIS_Close OEMIoControl: Unsupported Code 0x10100d0 - device 0x0101 func 52 OEMIoControl: Unsupported Code 0x10100f8 - device 0x0101 func 62 HW_USBClocks::D0 HW_USBClocks::D4 OEMIoControl: Unsupported Code 0x10100fc - device 0x0101 func 63 CamClockOn = 0 HW_USBClocks::D0	111

九、Platform Builder 5.0 安装注意事项

路径: DVD 光盘\wince5.0 评估版\。可试用 120 天,此涉及到微软的版权注册问题,我们只提供试用版方 便各位免下载而己,请使用者向微软相关部分注册使用,对此所涉及的所有责任与我们没任何关系。 在安装过程中,注意下面两个内容的选择,其它都是默认的下一步。



上图选择 Custom(Tools and OS)。

Select the	tup program features you want installed.	
Click on an ic	 Platform Builder Shared Source for Windows CE 5.0 Windows CE 5.0 Test Kit Windows CE 5.0 Operating System Emulator × 86 × ARMV41 	s installed. Feature Description ARM 72X, ARM92X, and ARM102X, cores, Intel XScale, Motorola DragonBall i.MX, NeoMagic, NM572XX, Samsung S3C24XX, Texas Instruments OMAP This feature requires 0KB on
	This feature will be install This feature, and all subfeature This feature, and all subfeature	ed on local hard drive. Atures, will be installed on loca

上图窗口必须选择添加 ARMV4I,因为 AT2440EVB-II 的 BSP 是在此项下编译的。 广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II

十、WINCE5.0 BSP 安装编译说明

BSP 是板级支持包,是介于主板硬件和操作系统之间的一层,应该说是属于操作系统的一部分,主要目的是为了支持操作系统,使之能够更好的运行于硬件主板,意思为 BootStrap Processor, or the CPU which enables the other CPUs in an SMP system. BSP 是相对于操作系统而言的,不同的操作系统对应于不同定义形式的 BSP,象 WINCE5.0 的 BSP 都是在原厂提供的通用模板上进行修改编译的。目前各个厂家所设计的开发板中均提供 BSP 包源代码(某些公司或者不提供,只提供镜像文件),只有相对应的 BPS 包源代码 才能可以订制自己需要的系统。

路径: WinCE5.0 源代码\ 有三个文件夹,分不同分辨率版本的 BSP 源代码。

10.1.将 SMDK2440A 复制到 WINCE500\PLATFORM 目录下,并去掉只读属性。我们为了区别不同的源代码,一般会在 SMDK2440A 后加些标识或按分辨率来分开不同的源代码,如 SMDK2440A-Sharp3.5',复制后请记得将文件夹名改为"SMDK2440A"。或 240_320\.....

10.2.打开 Platform Builder, 进入窗口, 先添加 smdk2440a.cec 文件, 选择 File 菜单下的 Manage Catalog Items。如下图:



10.3. 选择 Import

nported .cec <u>f</u>	iles:				
File	Version	Vendor	Description		OK
generic.cec	5.00	Microsoft	Generic driver type declar		E.C.
audio.cec	5.00	Microsoft	Audio Catalog Items		Kemove
lisplay.cec	5.00	Microsoft	Display Catalog Items	1 A 4	
.394. cec	5.00	Microsoft	IEEE 1394 Driver Support	In	nport
eybmouse. cec	5.00	Microsoft	Keyboard and Mouse Catalog		-
etcard. cec	5.00	Microsoft	Ethernet Catalog Items	F	Refresh
rda. cec	5.00	Microsoft	IrDA Catalog Items		
arallel.cec	5.00	Microsoft	Parallel Port Catalog Items		
ci.cec	5.00	Microsoft	PCI Bus Catalog Items		
cmcia.cec	5.00	Microsoft	PCMCIA Catalog Items		
rinting. cec	5.00	Microsoft	Printing and Printer Drivers		
erial.cec	5.00	Microsoft	Serial Port Catalog Items		
martcard. cec	5.00	Microsoft	Smart Card Support and Dri		
storage.cec	5.00	Microsoft	Storage Device Support and	1000	
asb. cec	5.00	Microsoft	USB Catalog Items		

10.4. 添加 SMDK2440A 目录下的 smdk2440a.cec 文件

Inport Catal	log Itens				? 🔀
查找范围(L):	C SMDK2440A		00	🥲 🖽 -	
 表最近的文档 していたいで、 していたいたいで、 していたいたいで、 していたいたいたいたいで、 していたいたいで、 していたいたいたいで、 していたいたいたいたいたいたいたいたいたいで、 していたいたいたいたいたいたいたいたいたいたいたいたいたいたいたいたいたいたいた	Doc Files Iib Src smdk2440a				
网工物店	文件名 (2):	smdk2440a			打开(0)
	文件类型 (I):	Catalog Item Files (*.cec)			取消

10.5. 新建一个工程,选择 File 菜单下的 New Platform



Hey Platfor Workspace Choose	n Ubsard - Stap 2 Name And Location a friendly name for your workspace.	×
	Name: AT2440H8	
	Path: C:\WINCE500\PBWorkspaces\AT2440H8	
0	< Back Next > Einish	Cancel

10.8. 选择"SAMSUNG SMDK2440A:ARMV4I",然后点击"Next"

A BSP contains a set of device drivers that	are added to your OS design. 🤇
Available BSPs: AMD GEODE: X86 CEPC: X86 EMULATOR: X86 INTEL PXA27X DEV PLATFORM:ARMV4I SAMSUNG SMDK2410: ARMV4I SAMSUNG SMDK2440A: ARMV4I	Select one or more BSPs for your OS design. A BSP for the Samsung SMDK2440A Development Platform. The platform uses the OS that is built for the ARM v4 architecture and contains the ARM instruction set with Thumb Interworking enabled. Note: Only BSPs supported by installed CPUs are displayed in the list.

10.9. 选择"Mobile Handheld", 然后点击"Next"

Available design templates: Custom Device Digital Media Receiver Enterprise Terminal Enterprise Web Pad Gateway Industrial Controller Internet Appliance IP Phone Mobile Handheld Set-Top Box Tiny Kernel Windows Thin Client	Choose the design template that is most closely aligned with the purpose of your target device. Provides the starting point for a range of personal digital assistants (PDAs) or mobile devices with a clamshell-and-keyboard design.
---	---

10.10. 选上下面这些选项, 然后点击"Next"。Quarter VGA Resources-Portrait Mode 是必选, 其它项建议尽可能少选, 编译生成的 NK 大于 32M 通过 USB 方法下载后, 偶然会看不到 WINCE 界面的现象或编译出错或下载出错等。ActiveSync 是 USB 同步功能, 需要使用 USB 同步必须选择此项, 如需要添加 word、Excel 等请选择相关的选项。



Nav Platform Vizard - Stay 6	🔀
Networking & Communications Select items for networking and communic design.	ations to include in your OS 🌏
Items: TCP/IPv6 Support Local Area Network (LAN) Personal Area Network (PAN) Wide Area Network (WAN)	The Internet standard protocol, version 6.
(2) < Back	Estimated size of these items: 6601 KB Next > Einish Cancel

10.12.继续点击"Next"

Nev Platforn Vizard - St	ay 7			
Bluetooth				Â
Security Warning				
Under certain circumstances, the security of your platform. This ca	Object Exchange Pr stalog item poses the	otocol (OBEX) cata following potentia	log item can compro I security risks:	mise the
 If proper security and authen can be installed. 	tication techniques a	re not used, a serv	ice that interferes wi	th services.exe
 If proper encryption technique to third parties. 	es are not used, OBE	X running over Blu	etooth could expose	data packets
To learn more about potential Of more securely, see the following	BEX security risks, a topics:	s well as the best p	ractices for using thi	s catalog item
OBEX Security				
Enhancing the Security of a Devi	<u>ce</u>			
IrDA				
Security Warning				
Under certain circumstances, the security of your platform. This ca	Object Exchange Pr atalog item poses the	otocol (OBEX) cata following potentia	log item can compro I security risks:	mise the
 If proper security and authen can be installed. 	tication techniques a	re not used, a serv	ice that interferes wi	th services.exe
If proper encryption technique	es are not used. OBP	X running over Bli	ietooth could expose	data nackets
2	< Back	<u>N</u> ext >	Einish	Cancel
	162 (S	104	28 - 18	10 (M)

10.13. 选择 "Finish"

Platforn Uiza	and - Step 8	
Completing) the New Platform Wizard	
You have succes	ssfully completed the New Platform Wizard.	
You have create platform. By def configuration ar	ed an OS design for a Windows CE-based ault, Platform Builder provides a Debug nd a Release configuration of this OS design.	
Options: • Modify build design witho	options for the Debug and Release configurations of your OS ut closing this wizard.	
To close this wize	ard, click <i>Finish</i> .	
-1		
2)	< Back Next > Einish	Cancel

至此新工程文件己建立完成。

10.14. 一个基本的工程己建好了,下面需要设置一些编译参数,选择"Platform"菜单下的"Settings"。

amsung SMDK2440A: ARMV4 الالم	I Release		and the second s
지지		1 2 2 3 1 2 0 0 1 2 5 CE Device	22 58 年 第1四
Active Tr Cubrari Cubrari Cuppon Lightweige	Start	Name Modified	Cotaleg Cotaleg Core OS Core OS Cor
B Messap	Getting Starled	C:\WINCES00'PBWorkspaces\AT2440HS	a 🔄 Third Party
III MSM	Children Cathonianites	DATTAGEN	
8 Object E		New Flatform Open Workspace	
* SOAP Te		20 XI	
* XML			
🛞 🔁 Communicz			
Gre OS Se File System		O histofication History	
🗄 🔛 Fonts			
Graphics attem Internation		Notification Date	
* 🔛 Internet Cliv		Device Feature Pack 2006-	4-4
Bos. Pa stra		Windows CE 5.0 Platform Builder.: Yearly Updated Full Update Fackage 2006-	2-3
		(through 12/31/2005)	



10.15. 在 "Build Options" 中按如下设置。此页的选项涉及很多方面的使用,在此不作介绍。如需要其它 功能的使用,请根据需要自己添加,同时可参考相关教材了解这些使用方法。

511D g SMDK24404	ARMV4T Roless			
ISTATE DIIDIZ TION	Aun H_Refeas			
neral Locale	Build Options	Environment	Custom Build Action	s Image Settings
Buffer track	ed events in RAJ rget Control Su	M (IMGOSCAPT) pport (SYSGE)	JRE=1) N_SHELL=1)	
🖌 Enable Eboot	Space in Memory	<pre>y (IMGEBOOT=)</pre>	D.	
Enable Event	Tracking Durin;	g Boot (IMGC)	ELOGENABLE=1)	
_ Enable Full] _ Frable Karne	iernel Mode (no Debugger (no	IMGNUTALLKMU	JUE=I) 8=1)	
Enable KITL	(no IMGNOKITL=1))	a-1)	
 Enable Profil	ling (IMGPROFIL)	ER=1)		
Enable Ship 1	Build (WINCESHI)	P=1)		
🗌 Flush tracke	d events to Rel	ease Director	ry (IMGAUTOFLUSH=1)	
Run-time Ima	ge Can be Larger	r than 32 MB	(IMGRAM64=1)	
Use XCOPY in	stead of links	to populate	release directory (B)	JILDREL_USE_COPY=
Write Kun-tin	he image to fia:	sh Memory (11	MGFLASH=1)	
<]		- 111		5

10.16. 在"Locales"中添加中文,如果需要操作系统显示中文,可以将"Default language"设成中文,再点击"OK"。如果只调试,建议使用英文,编译出来的NK内核会小很多,下载调试快。

	Setti	ings				
<u>C</u> onfigura	tion:					
Samsung S	MDK2440A	A: ARMV4I_Relea	se			~
				6		2
General	Locale	Build Options	Environment	Custom Bui	ld Actions	Image Settings
Locales	:					
口中文	(台湾)					
口中文	(香港特别	别行政区)				
	(新加坡)	1				<u>C</u> lear All
De <u>f</u> ault	languag	ge:				
中文(中	围)					
and the second second						
Coderes						
codepag	;es:				2000	
U <u>o</u> aepag	es: (OEM -	United States)	6			
 Codepag ✓ 437 ○ 708 ○ 720 	es: (OEM - (Arabic	United States) 2 - ASMO 708)	4500)			
 Codepage 437 708 720 737 	es: (OEM - (Arabic (Arabic (OEM -	United States) 2 - ASMO 708) 2 - Transparent Greek 437G)	ASMO)			Clear All
Codepag	es: (OEM - (Arabic (Arabic (OEM -	United States) 2 - ASMO 708) 2 - Transparent Greek 437G)	ASMO)			Clear All
Codepag	es: (OEM - (Arabic (Arabic (OEM -	United States) 2 - ASMO 708) 2 - Transparent Greek 437G)	ASMO)			Clear All
 ✓ 437 ○ 708 ○ 720 ○ 737 ✓ Local 	(OEM - (Arabic (Arabic (Arabic (OEM -	United States) 2 - ASMO 708) 2 - Transparent Greek 437G) <u>b</u> uild	ASMO)			Cl <u>e</u> ar All
✓ Local	es: (OEM - (Arabic (Arabic (OEM -	United States) - ASMO 708) - Transparent Greek 437G) <u>b</u> uild	ASMO)			Clear All
✓ 437 708 720 737 ✓ Local	es: (OEM - (Arabic (Arabic (OEM - Lize the t local:	United States) 2 - ASMO 708) 2 - Transparent Greek 437G) <u>b</u> uild ization checkin	ASMO) g in the buil	a		Clear All
 Codepag 437 708 720 737 ✓ Local Stric 	(OEM - (Arabic (Arabic (OEM - Lize the t local:	United States) = - ASMO 708) = - Transparent Greek 437G) <u>b</u> uild ization checkin	ASMO) g in the buil	d		Clgar All

10.17. 下面需要在工程中添加一些必须驱动的组件。这些驱动组件与我们提供的 BSP 有相应关联的,需要使用到的功能必须一一对应添加才可以支持。但若编译一个简单的版本测试 BSP 是否正常,可以选择简单的几项即可。

添加"Catelog\Device Drivers\SDIO\SD Memory",方法见图。以下操作类似,不再提示。



10.18. 添加 "Catelog\Device Drivers\SDIO\SD Host\SDIO Standard Host Controller"



10.19. 添加 "Catelog\Device Drivers\USB Function\USB Function Clients\Mass Storage"



10.20. 添加 "Catelog\Device Drivers\USB Function\USB Function Clients\Serial"



10.21. 添加 "Catelog\Device Drivers\USB Host\USB Host Controllers\OHCI"



10.22. 添加 USB 驱动, "Catalog\Core OS\Windows CE devices\Core OS Services\USB Host Support\USB Human Input Device Class Driver\USB HID Keyboard and Mouse"



10.23. 添加 "Catalog\Core OS\Windows CE devices\Core OS Services\USB Host Support\USB Storage Class Driver"



10.24. 添加"Catalog\Core OS\Windows CE devices\Applications and Services Development\ Microsoft Foundation Classes (MFC)"



10.25. 添加 "Catalog\Core OS\Windows CE devices\Applications and Services Development\ Standard SDK for Windows CE"



10.26. 添加注册表驱动 "Catalog\Core OS\Windows CE devices\ File Systems and Data Store\ Registry Storage

(Choose 1)\ Hive-based Registry"



10.27. 添加分区 FAT 类型 "Catalog\Core OS\Windows CE devices\ File Systems and Data Store\ Storage Manager\ FAT File System"



10.28. 添加分区管理器驱动 "Catalog\Core OS\Windows CE devices\ File Systems and Data Store\ Storage Manager\ Partition Driver"



10.29. 添加控制面板 "Catalog\Core OS\Windows CE devices\ File Systems and Data Store\ Storage Manager\ Storage Manager Control Panel Applet"



10.30. 添加 "Catalog\Third Party\BSPs\Samsung SMDK2440A:ARMV4I\Storage Drivers\SmartMedia NAND Flash Driver (SMFLASH)"

(以上部分驱动是必需添加的,如果少添加某项,可能会导致编译不成功,目前有部分客户编译不成功都是



10.31. 如需要测试电影播放,请添加下面的 MS 播放器,同时要添加相关的解码才能正常播放的。不过你不需要这方面的话,建议你不添加这部分,添加后会使 NK 很大的。



10.32. Windows Media Play 播放器需要添加相关解码才能正常使用,根据你的需要,添加下面图上的内容。



10.33. 其它方面的驱动根据自己的需要去添加。同时要修改 IP 地址等适合你的网络需求的,或添加其它应用程序到 BSP 包一起编译,请参考 WINCE5.0 使用指南相关的方法。

10.34.下面就可以开始编译了,选择"Build OS"菜单下的"Build and Sysgen"。有时编译出非文件性错的话,可选上"Clean Before Building"再用"Sysgen"编译一次即可解决问题。



35.整个编译过程视你的电脑配置档次,大概需要几十分钟。编译完后,得到的 NK.nb0 和 NK.bin 的文件 在 WINCE500\PBWorkspaces\(你建的工程文件名)\RelDir\smdk2440a_ARMV4I_Release 下,请复制到设定 的文件夹中,使用 USB 方法或 EBOOT 引导方法进行烧录下载内核。

十一、WINCE5.0 EBOOT 引导方式的烧写方法

特别声明:此方法是 WINCE 最原始的下载方式,我们没做进一步的开发,建议不要使用,有时会出现启动不稳定的问题,提供给有需要的用户学习或修改为适合自己的方式,此方法不提供升级与技术支持。其源代码在光盘 BSP 的 SMDK2440A 相应目前中,但我们己很久没升级这部分了,目前不一这适合使用,可参考学习这种方法。

11.1. EBOOT 烧写步骤说明

这个 WINCE5.0 的版本采用的是直接用 eboot 来引导 NK 的方式。先通过 JTAG 板将 eboot 烧写到 NOR FLASH 的 0 地址, eboot 启动后再通过网口烧写 NK 就可以了。注意这里的 eboot 和 NK 都是从 COM1 输 出信息的。SW2 的拨码开关设置为 1 与 3 在 "OFF", 2 与 4 在 "ON", 即为 NOR FLASH 启动方式。

11.2.烧写 eboot.nb0

烧写 eboot 与烧写 uboot 的方法是一样的,如新建文件夹 AT2440CE5,将 eboot.nb0, sjf2440.exe, ESHELL.EXE, NK.bin 四个文件复制到此目录中。然后连上 JTAG 板按如下方式烧写就可以了。

先输入 sjf2440 /f:eboot.nb0 , 回车后, 选择 "2", 将 eboot 烧写入 AM29LV800 的 NOR Flash 中。

C:\VINDOWS\system32\cmd.exe	×
D:\tools>sjf2440 /f:eboot.nb0	
••	
SEC JTAG FLASH(SJF) v 0.1	
(\$3C2440X & \$MDK2440 B/D)	
++ Usage: SJF /f: <filename> /d=<delay></delay></filename>	
> S3C2440X(ID=0x0032409d) is detected.	
[SJF Main Menu]	
0:KYS1208 prog 1:28F128J3A prog 2:AM29L0800 Prog 3:Memory Rd/Wr 4:Exit	
Select the function to test:2	
[AM29F800 Writing Program]	
NOTE: AM29LV800BB needs 4 step sequences for 1 half-word data.	
So,the program time is twice of Starata flash(2 step sequences). [Check AM291JI800]	
Manufacture ID= 1(0x0001), Device ID(0x225B)=2249	
1mage 512e:0h~16000h	
Available Target Offset:	-
	//.

在下面的 Input target offset: 输入 "0",将 EBOOT 烧入 NOR Flash 的 0 地址位置。大概过几分钟后,烧 写完毕。

C:\VINDOWS\system32\cmd.exe	- 🗆 🗙
4:Exit	
Select the function to test:2	
[AM29F800 Writing Program]	
NOTE: AM29LV800BB needs 4 step sequences for 1 half-word data.	
So,the program time is twice of Starata Flash(2 step sequences). [Check AM29LU800]	
Manufacture ID= 1(0x0001), Device ID(0x225B)=2249	
Image Size:0h~16000h	
Available Target Offset:	
0x0, 0x4000, 0x6000, 0x8000,0x10000,0x20000,0x30000,0x40000, 0x50000 0x50000 0x20000 0x20000 0x20000 0x20000 0x20000	
0xd0000,0xe0000,0xf0000	
Input target offset:0	
SectorOffset=0x0	
SectorSize =0x4000	
Erase the sector:0x0.	
Sector Erase is started!	
Start of the sector data writing.	
0 100 200 300 400 500 600 700 800 900 a00 b00 c00 d00 e00 f00 1000 1100	1200 -
	► //

11.3.烧写 NK.bin

Eboot 启动后在进入倒计时的时候按空格键,如下图所示

```
DNV v0.50J - For VinCE
                            [COM1, 115200bps] [USB:x] [ADDR: 0x30000000]
Serial Port USB Port Configuration Help
Explorer(V2.0) taskbar thread started.
                                                                                  ^
Microsoft Windows CE Ethernet Bootloader Common Library Version 1.1 Built Mar
8 2007 01:10:19
Microsoft Windows CE Bootloader for the Samsung SMDK2440 Version 2.4 Built May
13 2007
BP Init
FMD::FMD Init
FMD::FMD_Init - pBSPArgs->nfsblk = 0x0
FMD::FMD Init - READ REGISTER BYTE(pNFSBLK) = 0x0
FMD::FMD Init IPL ....
FMD::FMD Init Done
MID = 0xec, DID = 0x76 4th Cycle : 0xa5
NUMBLOCKS : 512(0x200), SECTORSPERBLOCK = 256(0x100), BYTESPERSECTOR = 512
(0x200)
MID = 0xec, DID = 0x76 4th Cycle : 0xa5
NUMBLOCKS : 512(0x200), SECTORSPERBLOCK = 256(0x100), BYTESPERSECTOR = 512
(0x200)
wNUM_BLOCKS : 512(0x200)
Press [ENTER] to launch image stored on boot media, or [SPACE] to enter boot
monitor.
Initiating image launch in 5 seconds.
```

11.4 进入菜单后按如下设置,修改为如下图一致。特别注意第5点,如果这个没修改为下图一样,烧写完

后,断电NK就消失了,因为只是保存在SDRAM位置,没有真正写入FLASH中。

下图的第 7 点有些网络是使用: 00:11:22:33:44:55 (建议采用这个通用性大些)。输入方法是: 00.11.22.33.44.55 回车即可。

INT v0.50J - For WinCE [COM1,115200bps][USB:x][ADDR:0x30000000]	×
Serial Port USB Port Configuration Help	
Initiating image launch in 4 seconds. Ethernet Boot Loader Configuration:	^
0) IP address: <u>192.168.1.8</u> 1) Subnet mask: <u>255.255.255.0</u> 2) DHCP: <u>Disabled</u> 3) Boot delay: 5 seconds	
4) Reset to factory default configuration 5) Startup image: <u>LAUNCH EXISTING</u> 6) Program disk image into SmartMedia card: <u>Enabled</u> 7) Program DM9000A MAC address (<u>11:22:33:44:55:66</u>) 8) Kernel Debugger: ENABLED	
9) Format Boot Media for BinFS E) Erase Reserved Block B) Mark Bad Block at Reserved Block F) Low-level format the Smart Media card	
D) Download Image now L) LAUNCH existing Boot Media image R) Read Configuration U) DOWNLOAD image now(USB) W) Write Configuration Right Now	
Enter your selection:	

11.5 然后按 F 格式化 NAND FLASH

E DNW v0.50J - For WinCE [COM1,115200bps][USB:x][ADDR:0x30000000]	
Serial Port USB Port Configuration Help	
1) Subnet mask: 255.255.255.0	<u>^</u>
2) DHCP: Disabled	
3) Boot delay: 5 seconds	
4) Reset to factory default configuration	
5) Startup image: LAUNCH EXISTING	
6) Program disk image into SmartMedia card: Enabled	
7) Program DMY000H MHC address (11:22:33:44:55:00)	
8) Kernel Debuyyer: EMHBLED D) Format Boot Modia for BinES	
F) FURMAL DUUL MEUIA FUR DINES	
R) Mark Rad Rlock at Recorned Rlock	
F) Low-level format the Smart Media card	
D) Download image now	
L) LAUNCH existing Boot Media image	
R) Read Configuration	
U) DOWNLOAD image now(USB)	
W) Write Configuration Right Now	
Enter your selection: f	
Reserving Blocks [0x0 - 0x3]	
reserve complete.	
Low-level format Blocks [0x4 - 0x1ff]	
erase complete.	
	<u> </u>

11.6 接上网线, 然后按 D 开始下载。注意网线是我们标配的交叉网线。

DNW v0.50J - For WinCE [COM1, 115200bps] [USB:x] [ADDR:0x30000000]	
Serial Port USB Port Configuration Help	
D) Download image now L) LAUNCH existing Boot Media image R) Read Configuration U) DOWNLOAD image now(USB) W) Write Configuration Right Now	^
Enter your selection: <u>D</u>	
INFO: Probe: DM9000 is detected.	
DM9000: MAC Address: 11:22:33:44:55:66	
System ready: Deepowing for deeplood	
rreparing for uownioad	
Sent ROOTME to 255 255 255	
Sent BOOTNE to 255.255.255	
Locked Down Link 1	
Src IP 192.168.1.8 Port 0400	
EthDown::TFTPD_OPEN::boot.bin	
-EbootSendBootmeAndWaitForTftp	
Download BIN file information:	
[0]: Base Address=0x80200000	
TFTP: Desktop losing ACK, block number = 3522, Ack again	~

11.7 下载方式有两种:

11.7.1 不需要使用 Platform Builder 的菜单功能,直接使用我们提供的 ESHELL.EXE 工具(在工具软件目录中),选择 NK.bin 所在位置,打开后即可直接下载,此称为快速烧写方法。此方法不需要重新编译工程文件,可直接下载我们编译好的内核 NK 文件。

11.7.2 打开 ESHELL.EXE 时。	XP系统经常会弹出一个警告窗口,	选择"解除阻止"。
-------------------------	------------------	-----------

Select	Device X
Enter De SMDK2	₩ Tindows 安全警报 🛛 🔀
Or <u>S</u> elec	为帮助保护您的计算机,Windows 防火墙已经阻止此程序的部分功能。
	您想保持阻止此程序吗?
	名称 (E): EShell TIN32 Application 发行者 (E): Microsoft Corporation
	保持阻止 医 解除阻止 (1) 稍后询问 (4)
	Windows 防火墙已经阻止此程序接受来自 Internet 或网络的连接。如果您 了解该程序或信任发布者,您可以解除阻止。 <u>何时应该解除阻止程序?</u>
	广州朗成电子科技有限公司 http://www.avantech.com.cn AT2440EVB-II

11.7.3 当从 EBOOT 菜单中己按"D"后,会发出设备名,如下图收到相应的设备名后,按"OK"。

Select Device	
<u>Enter Device Name:</u> SMDK244017493 Or <u>S</u> elect From the List	OK Cancel
SMDK244017493	Reset devices to automatically put names on the list.

11.7.4 进入到窗口后,选择 NK.bin 所在的位置。打开后,窗口会收到相应的位置,即开始下载通讯中。

📑 Microsoft Windows CE Debug	Shell - SMD	1244017493 (D:\AT	2440CE5\NK.bin)		- • ×
Like Edit Icola Options Fice Hal	p.	and party			
	• n	50			
打开				?	3
査找范围(1) 支援范围(1) 支援范囲(1) 支援范的文档 () 支面 () 支面 () () () () () () () () () ()	AT2440CE5		• E	0-	
	文件名 (20):	ях	•	打开(1)	5
30	文件类型(1):	MK Images (*. bin)		取消	1
				DOWNLOAD	CLEAN

11.7.5 打开 NK.bin 文件后, 窗口会收到正确的下载信息, 如下图。出现这样的信息才代表开始下载。



大概需要等几分钟时间。

11.7.7 另一种方法是在编译内核的同时,通过 Platform Builder 直接可以下载。在 EBOOT 菜单中按"D" 后, EBOOT 开始通过网络下载操作系统的映像文件,点击 PB 的 Target 菜单中的 connectivity options…。

🖓 Attach Device	Ctrl+Shift+D
😓 Detach Device	
🎋 Reset Device	
<u>R</u> un Programs	Alt+Shift+R
CE Debug Zones	
CE Target Control	Ctrl+Alt+Shift+A
😻 CE Processes	
CE Inreads	
🔼 CE Modules and Sym	ols
CE Exceptions	
<u>C</u> onnectivity Options	Ctrl+Shift+C
Debug Message တိုင်းရ	ons
Release Directory Mo	dules

在弹出 Target〉〉 Device Connectivity Options 对话框,将 Download,Kernel 两项选择"Ethernet"。

Stanson and a start			
Divisis Configuration	Larget Januar		
ALLING	CI Junio		
DISCREDUCE			
	Inclusion		
Gence Configuration	Distant		- Det tigge
forced line rate from	(MARTINE)		1
OVER REPORT OF THE	Texasport.		
Secret 2064	Itherat		- ferting
	(MARTERS)		
	and the second		
	Algheight .		
	heat		A Detters
		1	
		R	

再点击 Download 的 Settings 按钮
🥑 Ethernet Download Settings 🗙
Device Boot Name:
SMDK244017493
IP Address: 192.168.1.8 Bootloader: 2.4
Active Devices:

在 Active Devices 文本框中会出现请求连接的设备名 SMDK244017493,双击使之出现在 Device Boot Name 选择框内,点击 OK 退出。

点击 platform builder 中的 target ---Attach Device 项。会弹出下载对话框,并且显示下载进度。下载完 后请不要关电或停止,因为系统还要将 NK 写入 FLASH 中。请跳到 DNW 窗口观察过程,直到下面提示 的己成功为止。





以上两种方法均可使用 EBOOT 通过网口下载 NK.bin 内核的方法。

11.8 下载过程中会 eboot 会对 NAND FALSH 进行检测,如果发现坏块会自动屏蔽掉,如下图。

🏧 DHV v0.50J - For VinCE [COM1,115200bps][USB:x][ADDR:0x30000000]	×
Serial Port USB Port Configuration Help	
Sent BOOTME to 255.255.255.255	^
Sent BUUIME to 255.255.255.255	
Src IP 192_168_1_8 Port 0400 Dest IP 192_168_1_6 Port 061D	
EthDown::TFTPD OPEN::boot.bin	
-EbootSendBootmeAndWaitForTftp	
Download BIN file information:	
[0]: Base Address=0x80200000 Length=0x982074	
TFTP: Desktop losing ACK, block number = 3522, Ack again	
TFTP: Desktop losing ACK, block number = 15531, Ack again	
rom_offset=0x0.	
KUMHUK at Hodress 80200044n Wwiting single region/multi-region undate dwDINESPartLength: 0060700	
IsUalidMRR: MRR sector = 0x400	
OpenPartition: Invalid MBR. Formatting flash.	
Enter LowLevelFormat [0x0, 0x1ff].	
Erasing flash block(s) [0x0, 0x1ff] (please wait): EraseBlocks: found a bad	
block (0x0) - skipping	
EraseBlocks: found a Dad Dlock (0x1) - skipping Ekseplaaks: found a bad block (0x2) - skipping	
FraseBlocks. found a bad block (0x2) - Skipping	
L'ascolocast round à bad bloch (ono) Shipping	~

11.9 烧写过程比较长,要等待几分钟,如果出现如下信息表示烧写完成,WINCE 会自启动

DNV v0.50J - For VinCE [COM1, 115200bps] [USB:x] [ADDR:0x30000000] Serial Port USB Port Configuration Help Enter LowLevelFormat [0x0, 0x1ff]. ~ Erasing flash block(s) [0x0, 0x1ff] (please wait): EraseBlocks: found a bad block (0x0) - skipping... EraseBlocks: found a bad block (0x1) - skipping... EraseBlocks: found a bad block (0x2) - skipping... EraseBlocks: found a bad block (0x3) - skipping... Done. WriteMBR: MBR block = 0x4. Done. OpenPartition: Partition Exists=0x0 for part 0x21. CreatePartition: Enter CreatePartition for 0x21. LastLogSector: Last log sector is: 0x1fbff. CreatePartition: Start = 0x500, Num = 0x4d00. Log2Phys: Logical 0x500 -> Physical 0x900 WriteMBR: MBR block = 0x4. BP_SetDataPointer at 0x0 WriteData: Start = 0x0, Length = 0x982074. Log2Phys: Logical 0x500 -> Physical 0x900 Updateded TOC! IsValidMBR: MBR sector = 0x400 OpenPartition: Partition Exists=0x0 for part 0xb. CreatePartition: Enter CreatePartition for Oxb. FindFreeSector: FreeSector is: 0x5200 after processing part 0x21.

11.10 这是 WINCE5.0 启动完成后的信息

DNV v0.50J - For WinCE [COM1,115200bps][USB:x][ADDR:0x30000000] Serial Port USB Port Configuration Help FMD::FMD_Init - READ_REGISTER_BYTE(pNFSBLK) = 0x0 ~ FMD::FMD_Init IPL FMD::FMD Init Done MID = 0xec, DID = 0x76 4th Cycle : 0xa5 NUMBLOCKS : 512(0x200), SECTORSPERBLOCK = 256(0x100), BYTESPERSECTOR = 512 (0x200) FMD OEMIoControl: unrecognized IOCTL (0x71c24). OEMIoControl: Unsupported Code 0x1010118 - device 0x0101 func 70 OEMIoControl: Unsupported Code 0x10100d0 - device 0x0101 func 52 OEMIoControl: Unsupported Code 0x10100f8 - device 0x0101 func 62 INFO: CReq2440Vart::CReq2440Vart using processor frequency reported by the OAL (66608640). INFO: CReq2440Uart::CReq2440Uart using processor frequency reported by the OAL (66608640). DeviceFolder::LoadDevice!Enumerate Found deprecated load instructions at (Drivers\BuiltIn\AFD). Driver cannot be unloaded. DeviceFolder::LoadDevice!Enumerate Found deprecated load instructions at (Drivers\BuiltIn\PPP). Driver cannot be unloaded. OEMIoControl: Unsupported Code 0x10100fc - device 0x0101 func 63 Lyg.p: Layout Manager successfully initialized to 2 Maximum Allowed Error 1280: MSIM: IM ReadRegistry read KB 5 Explorer(V2.0) taskbar thread started.

十二、WINCE5.0 使用指南

1. MP4 播放器

在"我的设备——Windows—player"可以播放 MP4(avi)格式的电影,经测试使用 800*480 分辨率全屏播放,效果非常流畅,图象清晰。可以使用光盘自带的电影测试,得到的参数如下: BetaPlayer Version STABLE.0.5 Benchmark Results

Average Speed	49.88%	(平均速度)			
Video Frames	31	(视频帧数)			
Audio Samples	84480	(音频采样)			
Amount of Data	123 KB	(数据总计)			
Bench. Time	0:02.591	(测试时间)			
Bench. Frame Rate	11.96	(测试帧速率)			
Bench. Sample Rate	32593	(测试音频采样)			
Bench. Data Rate	391 kbit/s	(测试数据传输率)			
Original Time	0:01.292	(原始时间)			
Original Frame Rate	23.98	(原始帧速率)			
Original Sample Rate	48000	(原始音频采样)			
Original Data Rate	784 Mbit/s	(原始数据传输率)			
URL	\Storage Car	·d∖A1_mp4.avi			
Size	67174400				
Platform	PLATFORM_TYP	ΡE			
OS Version	5.00				
OEM Info	PLATFORM_OEM	[
Clock speed	416 Mhz	416 Mhz			
Video output	Raw FrameBuf	fer 800x480 Lookup			
Audio output	Wave Output	Wave Output 48000Hz 16Bits 2Ch.			

或者可以使用微软自带的 Media Player,但这个不支持 MP4 格式。

2. USB 摄像头测试

采用中星微 301 系列摄象头驱动程序,最高支持 200 万有效像素,速度与清晰率非常好,可与 PC 常用的设备效果相比。驱动己添加到 BSP 包里,插入 USB 摄像头后,进入"我的设备--Windows—Testzc030x",打开 Testzc030x,会提示找到1个设备,按 0K 即可出现图像窗口。

本程序采用第三方驱动与测试程序,只有 DLL 驱动,不提供源代码等,需要咨询注册与购买请看说 明中第三方的资料与介绍。

3. Camera 摄像头测试

WinCE下的摄像头驱动支持 30 万像素的 OV7660 模块,如用户需要配置其它型号的摄像头模块,请 自行修改相关驱动程序。此部分的测试程序只对接口验证是否有效,对于所看到的测试效果不佳等不负责。 同时并没有这部分相关的源代码提供,只提供相关驱动作参考使用。

AT2440EVB 开发板提供这个接口,在 WINCE 系统中也添加了这个 OV7660 的驱动,在 WINDOWS/CameraTEST.exe 的测试 DEMO,可以用来测试摄像头与接口是否正常。但 DEMO 所看到的效 广州朗成电子科技有限公司 http://www.avantech.com.cn AT2440EVB-II 76

果中比较差的,并且只能看图,不能抓图,保存等。没有此部分源代码提供。

4. SD 卡

WINCE5.0 支持 SD 卡即插即用,插入 SD 卡后,会在"我的设备——Windows"找到"Storage Card" SD 卡文件夹。

目前最大支持1G容量,更大的容量需要自己修改文件系统,我们暂不提供此项服务。

5. 保存文件

在"我的设备——Windows"找到"ResidentFlash",将你的文件资料放在此目录下,可以永久保存,断电后数据不会掉失。如果你要将程序放在Windows目录下,那就需要在编译时将你的程序添加到BSP中,一起编译。

6. USB HOST 接口

支持 USB 鼠标、USB 键盘与 U 盘等设备即插即用。USB 是 1.1,有些 USB2.0 设备有可能不支持。 同时有些 USB 鼠标由于供电不足也会导致不用使用,建议使用 USB 的小鼠标。

7. 网络使用

进入"我的设备——控制面板——网络和拨号连接",打开"DM9CE1"进行设置 IP 地址、名称服务 器等参数,即可登录网络。此为 100M 的网卡。

上网需要设置 DNS,中国电信的 DNS 一般为 202.96.128.166.

WINCE5.0 支持局域网方式上网,其它方式如 ADSL 拨号上网暂不支持。同时支持 USB 无线网卡上 网方式。

8. 触摸屏校正

进入"我的设备--控制面板--笔针",可重新校正触摸屏,并可保存。

触摸屏的参数在注册表中可以查看到,在"我的设备——Windows——regedit",打开注册表,在"My Device——HKEY_LOCAL_MACHINE——HARDWARE—DEVICEMAP—TOUCH",CalibrationData的Data的Data的值就是触摸屏参数。

在 PB 中 Workspace——ParameterView——platform.reg,双击打开代码,在下面内容可以修改后再 Make Rum-Time Image.

[HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\TOUCH]

"MaxCalError"=dword:500

;800*480

"CalibrationData"="486,293 151,529 811,530 803,61 140,58 "

9. 录音

在"我的设备——Windows——recorder",通过话筒插入 MIC 音频录音口,点 recorder 窗口的红色 按钮,即可开始录音。支持同时录音,放音。

10. 显示界面旋转 90 度

目前我们配的 3.5 'LCD 分辨率显示为 240*320 竖屏,有些客户需要将显示为横屏效果 320*240,由于 SHARP 屏的驱动支持这个旋转显示,现在将修改的方法列出如下:

用记事本打开: SMDK2440A\Files\platform.reg 修改下面的注册表为如下内容,重新编译即可: [HKEY_LOCAL_MACHINE\System\GDI\Rotation]

"ANGLE"=dword:5a

"LandscapeMode"=dword:1

注: 请注意大小写要区分。

原来为 240*320 如下:

[HKEY_LOCAL_MACHINE\System\GDI\Rotation]

```
"Angle"=dword:0
```

为了演示效果,你也可以通过进入CE5.0/Windows/regedit.exe中

[HKEY_LOCAL_MACHINE\System\GDI\Rotation]修改 Angle 为 90, 然后重启即可。但断电后不会保存结果的。

11. GPS 接收器的使用

若选购我们配的 GPS 模块, 会配有一个 TTL 转 COM 的小板, 这个板上有两条线, 一条是焊有串口头(母头), 另一头只是两根, 分别为 3.3~5V 电源线与地线。

另一种方法是可从 IO 扩展口中直接引出 COMO 的 TTL 信号到 GPS 模块口,注意 RX, TX 交叉互换才可以。 将串口头接到开发板的 COMO,另那两根电源线可接到 UART 的第 1,2 脚。UART 的第 1 脚是 3.3V,第 2 脚是地线。关于 UART 的原理请查看上述硬件说明或查看原理图。

接线方法见下图:



GPS 测试软件在光盘 "GPS 软件"的目录中,有测试串口是否找到 GPS 设备的工具,还配有 GPS 地图 软件试用版。需要将目录下的内容复制到 SD 卡中,插入到开发板中直接使用。使用此功能完全可以作为 一台手持 GPS 导航仪。

12. USB 同步

在我们编译好的 NK 内核中, 己增加 USB 同步的选项, ActiveSync 是 USB 同步功能,选择后 CE 进入窗口会经常弹出检测 USB 连接,如果你的 2440 板 USB 从口没有连接到 PC 电脑上,会一直发出检测提示,可以在"我的设备——控制面板——PC 连接"中取消此"启用与台式计算机的直接连接"选项后 OK 即可保存此设置,我们己解决了注册表保存功能,就算断电后都不掉失。

使用此功能的用户请 USB 同步在 WINCE5.0 中不需要进行任何设置,在控制面板--PC 连接,打开 PC 连接属性,需要选择"启用与台式计算机的直接连接"然后 OK 即可。默认使用 USB 连接。。

要进行 USB 同步测试,首先要在 PC 机上安装同步软件 MicrosoftActiveSync_setup_cn.msi(在光盘的 工具软件/wince 同步软件/目录里面)。

第一次使用此功能时需要安装相关驱动,将 USB 打印线一头接在 AT2440 的 USB DEV 口,另一头插 在 PC 的 USB 口上,进入 WINCE5.0 系统后会提供发现新的硬件,照下面步骤进行安装 USB 同步的驱动。



选择"从列表或指定位置安装",点击"下一步"

找到新的硬件向导	
	这个向导帮助您安装软件: USB Device 如果您的硬件带有安装 CB 或效念,请现在将
	您期望向导做什么?
	 ○ 自动安装软件(推荐)(L) ● 以列表或指定位置安装(高级)(S)
	要继续,诸单击"下一步"。
	< 上一步 (8) 下一步 (8) > 取消

选择"在搜索中包括这个位置",找到光盘中的"工具软件\USB 驱动\WinCE 下与 PC 同步的 USB 驱动"目录,点击"下一步"

找到新的硬件向导
请选择您的搜索和安装选项。
 ● 在这些位置上搜索最佳驱动程序(S)。 使用下列的复选框限制或扩展默认搜索,包括本机路径和可移动媒体。会安装找到的最佳認动程序。 ■ 搜索可移动媒体(软盘、CD-ROM)(M) ● 在搜索中包括这个位置(D): D:\tools\USB驱动\wince下与PC同步的USB驱动
○不要搜索。我要自己选择要安装的驱动程序 (2)。 选择这个选项以便从列表中选择设备驱动程序。Windows 不能保证您所选择的驱动程序与您的硬件最匹配。
(< 上一步 (B) (下一步 (U) >) [取消]

选择"仍然继续"



点击"完成"



则 PC 端与 USB 设备同步的驱动安装完成。这时同步软件 Activesync 会自动打开,第一次使用时会提供 USB 同步是否选择同步,选择后要将"文件"部分选上,确定后将出现下面窗口进入。全部使用默认的配置,不需要作修改。

选择是, 然后下一步。

新建合作关系	
	建立合作关系
	要想在移动设备和此计算机之间同步信息,必须先在 二者之间建立合作关系。 要建立合作关系吗?
	 > 邊所 建立合作关系,以便在移动设备和此计算机之间 同步信息。 () 否(0) 无需同步信息。将移动设备设置为"来宾",以便 在移动设备和此计算机之间复制或移动信息。
	〈上一步 ⑧〉下一步 ⑭〉〉 取消 帮助

选择文件,弹出说明窗口,选择确定。

新建合作关系	×
选择同步设置 选择准备同步的信息类型。	
文件同步	
終在台式计算机上创建一个"同步文件"文件夹。请将您准备进行 同步的文件放在该文件夹中。 当移动设备和此台式计算机之间进行同步时,Microsoft ActiveSync可能需要转换文件。注意:转换后的文件可能无法包含 初始文件中的所有信息。	
要了解更多有关同步信息以及自定义同步内容,请在列表中单设置 <u>(S)</u> 设置(S)	
< 上一步 (B) 下一步 (U) > 取消 帮助	

选择了文件然后下一步。

新建合作关系 🔀
选择同步设置 选择准备同步的信息类型。
要同步特定类型的信息,诸选中其复选框。要停止同步,诸湷除其复选框。
移动设备 台式计算机
☑ 函 文件 同步文件
西了般西夕方光同步信有则还自会议同步中容。注方利吉中省
要了解更多有天向步信息以及自定义向步内容,诸在列表中单 设置(S)… 击相应的信息类型,然后单击"设置"按钮。
< 上一步 (2) 下一步 (2) > 取消 帮助 税 (1) 取消 (1) 取[1) 取[1] (1) 取[1] (1) 和 (1)

出现完成的窗口,点完成即可进入下面界面。

🔞 Licrosoft ActiveSync	
文件 (2) 视图 (2) 工具 (2) 帮助 (2)	
 ⑧ ⑧ ○ □ □ □<	
WindowsCE	
查找更改的内容	
(====	
信息类型状态	
🔁 文件	

更改成功后,即可完成,下图为同步成功的窗口。

🕲 Licrosoft Ac	tiveSync	
文件(27) 视图(2) [Ĺ具(Ľ) 帮助(H)	
 ② ③ 同步 停止 详细信 	息 浏览 选项	
WindowsCE		
己连接 己同步		
信息类型	状态	
🔁 文件	已同步	

此时点击工具栏的"浏览",就会弹出开发板上"我的电脑"中的内容。此时可将开发板作为一个 U 盘设备使用,我们开放了一个保存区"ResidentFlash",将文件复制到此目录即可永久保存,断电后也不掉 失。

Sake						26	×
文件(2) 編時(2) 査者(2) 收算	w IAD AND	_					ŧ,
Q at - 0 - 3 P	秋秋 🕑 文件美 🛄・						
18世(1) 🚦 都助会る						× 🗗	樹
末空牧田 ・ 夏 市的市路 ・ 夏 市的市路 ・ 日 市の文格 ・ ● 井手文格 ・ • 月上泉田 ・	Application Date Fib3	By Documents	Program Files	Deni dent	Teap	Tinders	

偶会有用户提到 USB 同步不稳定的问题,经过与部分用户了解情况,有部分与 PC 主机的 USB 接口 输出电压偏低有关系的。目前此功能是比较稳压的,除非驱动没有安装正确才会导致 USB 同步出错。如 果有时检测不了同步,请将 USB 线插拨多次即可。

13. 添加你的程序到 NK 中

如需要将你的程序添加到NK内核中一起编译烧录到开发板中,可参考我们做的USB摄像头程序一样, 作个简单的介绍。

将你的程序复制到"WINCE500\PLATFORM\SMDK2440A\Files"中,包括 exe、dll、reg 等必须的文件,然后在 PB 中 Workspace——ParameterView——platform.bib 打开代码,在下面内容添加链接后再 Make Rum-Time Image.

FILES

;	Name	Path	Memory Type		
;					
	Testzc030x.exe	$(_FLATRELEASEDIR) \setminus Testzc030x.exe$	NK	U	
	zc030x.dll	$(_FLATRELEASEDIR) \ge c030x.dll$	NK	U	
	zc030x.reg	\$(_FLATRELEASEDIR)\zc030x.dll	NK	U	

同时在 platform.reg 此页中添加注册表的链接。

[HKEY_LOCAL_MACHINE\Drivers\USB\LoadClients\2760\Default\Default\ZC030X] "DLL"="ZC030X.DLL"

[HKEY_LOCAL_MACHINE\Drivers\USB\ClientDrivers\ZC030X] "Prefix"="CAM" "Dll"="ZC030X.DLL"

在光盘的目录中"PB简易上手指南"有介绍如何添加复杂的程序到NK内核中的方法。

14. 存储管理器

在控制面板中增加了存储管理器功能,可以对 FLASH 进行格式化等功能。请注意,会造成数据掉失。

15. 调整内存空间

如果你的程序需要放在 WINCE 目录下测试,但发现提示空间不足,请到控制面板中,打开系统,在 内存这页,可以调整内存的。

16. 增加启动时个性化 LOGO

在我们配的 BSP 中, 己添加了 LOGO, 如果需要更换为你自己的个性化 LOGO, 可照下面方法更换, 再重新编译一次, 得到的 NK 就包括你的 LOGO。

进入到 "WINCE500\PLATFORM\SMDK2440A\Src\Kernel\Oal" 目录下,有一个 bitmap.c 的文件,此 文件就是包括了 LOGO 的转换代码。

在光盘"\AT2440EVB-II\工具软件"目录下,有一个 Bmp2HexPro.exe 工具,打开后将你的界面设置为如下图一致的配置。



通过 OPEN 打开你设计的 BMP 格式图片,记得使用 16BIT 的格式保存。图片大小要与屏的分辨率一致才保证不会变形。

文件格式 ⓒ Windows ⓒ 0S/2	 取消
深度	
C1位 C4位	
C 8位	
● 16 位	
C 24 1 <u>以</u> C 32 位	
□ 压缩(RLE) □ 翻转行序	高级模式

打开你的 BMP 图片时,软件会弹出可以编译的窗口,此时点下周围的位置即可退出。



然后点 Output,输出.c 的文件类型保存。建议直接保存为 bitmap.c 的文件名。

1211Bmp2Hex Pro - [240320]			(点袋かか) - ロ×
Tile View Window Help	_		X
New Open Close Edit Save Output	at Preview Print Help		
🔽 Color Mode 🔿 12BPP Mode 💽 16BPP	Mode 🔿 18BPP Mode	🔽 Output Freview 🔽 16	5-bit data output
🗖 Extend Width 🔽 Extend Height 🔽 Color	Statistic 🔽 Paged Output	🦳 Reverse Output	
	♥ 明成科 领取未 创造未	技 来 来	
Bmp2Hex Professional Edition			240 X 320 X24bpp //

用编辑器或记事本打开你所保存的 bitmap.c 文件,在第一行位置你将会见到下面函数: unsigned short []={

现在你需要将以下的内容替代刚才的函数:

#include <windows.h>
const USHORT ScreenBitmap[] = {

保存后得到新的 bitmap.c 文件。

此时将新的 bitmap.c 复制到 "WINCE500\PLATFORM\SMDK2440A\Src\Kernel\Oal"目录下,覆盖原有的相同文件,再重新编译 NK,即可添加你的个性化 LOGO。

下面的内容为己修改完成的前面部分,仅供参考对比。



17. USB 无线网卡驱动安装编译

必须先按照上述方法将我们提供的 BSP 源代码编译成功后,再去添加 USB 无线网卡驱动,然后再执行一次编译才能正常使用。

一、驱动位置

本系统使用的 USB 无线网卡为 VIA Networking VT6656 USB-Wireless LAN Adapter,我们提供 WINCE4.2/5.0 系统的驱动,目录在光盘的 wince5.0-源代码\USB 无线网卡,请将"3rdParty"文件夹复制 到你安装的 WINCE500 目录下,并去掉只读属性。请注意必须为相应的 WINCE5.0 驱动才可以,USB 无 线网卡的驱动 WINCE4.2 与 WINCE5.0 是不兼容的。见图 1

[-c-] 🖌 [system] 906.6 MB 在 14,991.3 MB 中
c:\WINCE500*.*
文件名
€ []
🔁 [3rdparty]
[PBWorkspaces]
E [CRC]
E [PLATFORM]
E [OTHERS]
pass1
pass2
E build
build

二、添加驱动到 BSP 中

- 1. 首先确保你的开发板或产品有 USB 接口驱动并能正常使用。
- 2. 打开 Platform Builder,在你己编译好的 BSP 中,打开原有的 BSP。
- 3. 从菜单 File 点击 Manage Catalog Features , 见图 2

	Fil	e <u>E</u> dit <u>V</u> iew <u>P</u> roject Pl <u>a</u> tform Ta <u>r</u> get <u>B</u>	duild	Tools
2		New Platform Ctrl+N		F E
		New Project or File Ctrl+Shift+N		
	2	Open Ctrl+O	V4I V	Vin32
-		<u>C</u> lose		;
E		Open Workspace		-
		Sava Workspace	de∨	
		Close Workspace		;
	_	CIOSC #01mspace	elo	;
		Save Ctrl+S	let	-
	_	Save <u>A</u> s		,
	œ	Save A <u>l</u> l		II
		Page Setup		
	6	Print Ctrl+P		
		Recent Files		EI
		Recent Wo <u>r</u> kspaces		
		Manage Catalog Features		
		License <u>B</u> inary Image		
		E <u>x</u> it		F
_				;
	_		_	;
<				
63	Fe	atureView 🕸 Parameter 🔳 FileVie	w	
				山

4. 进入 Manage Catalog Features 窗口,见图 3

			1 .		OF
File	Version	Vendor	Description		AU
aspen, cec	4.20	Microsoft	Aspen BSP Features		
rockhopper	4.20	Microsoft	Rockhopper BSP Features		<u>R</u> emove
cepc. cec	4.20	Microsoft	CEPC BSP Features		
eagle.cec	4.20	Microsoft	Eagle BSP Features		Import
emulator.cec	4.20	Microsoft	Emulator BSP Features		
geode. cec	4.20	Microsoft	Geode BSP Features		Refresh
integrator	4.20	Microsoft	ARM Integrator BSP Features	_	
keywest. cec	4.20	Microsoft	Keywest BSP Features		
lubbock. cec	4.20	Microsoft	Lubbock BSP Features		
smdk2410. cec	4.20	Microsoft	Samsung SMDK2410 Developme		
i abase. cec	4.20	Microsoft	Display based and Headless		
wcetk. cec	4.20	Microsoft	Windows CE Test Kit Client		
platmgr.cec	4.20	Microsoft	Platform Manager Features		
sourcetags	4.20	Microsoft	Source code information fo		
smdk2440. cec	4.20	Microsoft	Samsung smdk2440 Developme	~	

图 3

5. 点 Import,选择 VNUWLC5.CEC 。见图 4

nage,	Import Cata	log Items				2
File	查找范围(g):	🔁 Vauxle5	•	01	• 🖽 •	
generi sodio displa 1394.c keybmo- netour irda.c parall pci.ce pcmcia printi serial imartc storag usb.ce	表現近的文档 変面 変加文档 変加文档 変加电路 阿上等居	Construction				
Free		文件名 (1):	VIRVATICS		• [打开①
		文件类型(I):	Catalog Ites Files (*. cec)		~	取消

图 4

6. 选择打开后,添加 VNUWLC5.CEC 到 Manage Catalog Features 窗口。见图 5

ported . cec f	iles:				
File	Version	Vendor	Description	^	OK
baul100. cec	5.00	Hicrosoft	DBAu1100 ESP Features	1	-
bau1500.cec	5.00	Microsoft	DBAu1500 ESP Cutalog Items		Eenove
mulator.cec	5.00	Microsoft	Emulator ESP Catalog Items		and the second second
eode. cec	5.00	Hicrosoft	Geode ESP Catalog Items		Import.
ainstonei	5.00	Microsoft .	Intel MainstoneII BSP Cata		-
\$2_vr4131	5.00	Hicrosoft	SG2_VR4131 BSP Catalog Items		Refresh
\$2_vr5500	5.00	Hicrosoft	SG2_VR5500 BSP Catalog Items		
adk2410. cec	5.00	Microsoft	Sansung SMDE2410 Developme	1.00	
latnan. cec	5.00	Microsoft	Platform Manager Features	1.0	
oreos. cec	5.00	Microsoft	Windows CE Core OS components		
cetk. cec	5.00	Hicrosoft.	Windows CE Test Kit Client	19	
ourcetags	5.00	Microsoft	Source code information fo		
lonemodul	5.00	Hicrosoft.	Catalog items that can be		
ndk2440a.cec	5.00	Sansung	Sansung SMDE2440A Developm		
muwlc5.cec	5.00	VIA Netw	Ethernet Catalog Items	v	

图 5

7. 点 OK,即添加了驱动到 BSP 中,从 Catalog 窗口中即可查到看。

"Catalog\Third Party\Device Drivers\Networking\Local Area Networking (LAN) devices\VNT VNWLC5

Wireless LAN"...见图 6



图 7

8. 点鼠标右键选菜单 Add to Platform 将此驱动添加到 BSP 中,见图 7



9. 从下图中可以确定无线网卡驱动添加成功。见图 8

"Catalog\Device Drivers\Networking\Local Area Networking (LAN) devices\CISCO Aironet 340/350 PCMCIA WLAN Card"



10. 修改 IP 协议地址,符合你的局域网等参数要求。请用记事本打开 VNUWLC4.REG,见图 9,10。

c:\WINCE500\3rdparty\Vnt\Vnuwlc5*.*	
文件名	扩展:大
€ []	<
(obj)	<
🔁 [Target]	<
📳 Build	log
Build	dat
@ VNUWLC5	reg
🗐 ReadMe	txt
E YNUWLC5	CEC
S VNUWLC5	bib
sources	
i postlink	bat
VNUWLC5	pbpxml
VNUWLC5	def
makefile	
prelink 🛛	bat
ProjSysgen	bat

图 9



图 10

11. 从图 10 中,我们选择局域网固定分配 IP 地址的选项,将 IP 地址,网关,子网掩码,DNS 等信息 修改好。如果你不了解这些应该填写什么,请与你的网管联系确定。 如果你选择 DHCP 的方式,请将屏蔽的符号去掉,再屏蔽分配 IP 地址这部分。 修改好后保存,关闭 VNUWLC5.REG。

12. 从菜单 Build OS 选择"Sysgen"编译内核。见图 11

如果你编译成功了,下次想修改 IP 地址这部分,请重复第 9,10 两步,然后再 Make Run-Time Image 即可。

Bu	ild OS <u>T</u> ools <u>W</u> indow <u>H</u> elp
	Sysgen
	Build and Sysgen
	Build and Sysgen Current BSP
2	<u>C</u> lean Before Building
-	Copy Files to Release Directory After Build
~	Make Run-Time Image After Build
	Copy Files to <u>R</u> elease Directory
	Make Run-Time Image

图 11

13. 编译得到的 NK.bin 内核文件,请照 CE 使用说明烧写方法,将 NK 下载到 AT2440EVB-II 开发板的, 方法请参考 CE 快速烧写方法 介绍。

14. 等 CE 系统启动成功后,将 USB 无线网卡插入 USB 口,即会弹出网络联接窗口。如果你配好无线 HUB, 从 CE 系统的无线网卡窗口的,无线信息此页(图 14),会搜索到相关的无线 HUB 信息。我们使用的是 TP-LINK TL-WR541G+ 54M 无线宽带路由器,系统会找到 TP-LINK,选择此列,点"连接",在弹出 的无线属性 窗口中,按 OK,返回到无线信息窗口时,会自动搜索相关信息,最后看到

状态: 己连接到 TP-LINK

信号强度:非常好

代表己成功连接,按OK确定后,CE即可上网等操作。

同时在"我的电脑一控制面板一网络和拨号连接"窗口中会看到 VNUWLC51 的图标,打开后可修改 IP 地 址与 DNS 等参数。不过在此修改,断电后参数不能保存。

见图 12, 13(注: 此图的天线是我们自己另配的, 原厂并没有配相应的天线)



图 12

VNUWLC41		OK	×
IP 信息 IPv6 信息	息 无线信息	Sale J.	
选择网络并按"连 络,请双击"添加	接"或右键单击查看更多选项。)新网络"。	要添加新网	
A FRANCE PROS			
▲ 添加新网络			-
↑添加新网络 ↑ TP-LINK (首)	选)		
↑添加新网络 ・ TP-LINK (首)	走)		
★ TP-LINK (首)	选) 已连接到 TP-LINK		
 ★ 添加新网络. ★ TP-LINK (首) 状态: 信号强度: ✓ 新的无线网络 	选) 已连接到 TP-LINK 非常好 可用时通知我		

图 13

15. 同时我们测试发现一个问题,就是 USB 无线网卡插入在开发板上,再开机,此时找不到无线网卡, 需要先进入 CE 系统,再插入 USB 无线网卡,才能正常找到此设备。有待研究此问题。

16. 关于 TP-LINK TL-WR541G+ 54M 无线宽带路由器 的安装使用,在此顺便提及下,以便没有使用 过的朋友方便参考。

可以将 PC 机上网线直接接到 HUB 上,在 IE 地址中输入 192.168.1.1,输入用户名,密码进入到管理窗口。 默认的用户名与密码都是 admin。

关于相关的设置可参考 HUB 的使用说明,以选择符合你的宽带参考要求。

确保你的网络正常后,你只需要"开启无线功能"选上,即可使用无线设备。见图 14



17. USB 无线网卡 VIA Networking VT6656 USB-Wireless LAN Adapter 参数说明。



Embedded System/IA Solution for VT6656

WLAN 802.11B/G embedded module enables high-speed 54Mbps wireless connections to LAN services. Integrated WLAN into embedded module devices makes it more convenient to get on the net everywhere you are.

WLAN embedded system/IA solution is the innovation design, fully compliant with IEEE's security and 802.11g standards. Enabling data rates between wireless devices of up to 54Mbps, it can deliver the bandwidth necessary for real-time streaming of high-definition digital multimedia content in home networks, and provide high-speed email, web and LAN access to multiple mobile users in corporate environments or public hotpots.

Specifications:

Radio	
Den de	•2400-2483.5MHz(for US, Canada)
Frequency Band:	•2400-2483.5MHz(for Europe)
	IEEE 802.1b, Direct Sequence Spread Spectrum,
	IEEE 802.11g,Orthogonal Frequency Division
Operating Channel:	Multiplexing
	•11 Channels(US, Canada)
	•13 Channels(Europe, Japan)
Modulation Type:	QPSK, BPSK, CCK, OFDM
Data Rate:	Up to 54Mbps
Antenna:	To support two external antenna via connector
Radiation Power:	13-17dBm at maximum radiation direction
Voltage Supply:	5V
Dowon Consumptions	①384uA@5V, Suspend/high speed mode
Power Consumption:	②259uA@5V, Active/high speed mode
Dessiver Consistivity	①Min-71dBm@54Mbps,PER<8%
Receiver Sensitivity	②Min-84dBm@11Mbps, PER<8%
Compliant:	FCC Part 15 C(USA); CE ETSI 300-328(Europe);USB-IF
Media Access Controlle	r/Baseband Controller
Chip:	VT6656
Media Access Protocol:	CSMA/ CA(Collision Avoidance) with ACK
USB Interface	
Host Interface:	Support USB2.0/1.1 standard
Driver Supporting	
	Windows: 98SE/ Me/ NT/ 2000/ XP/ 2003 Ser.,
Operating System:	Linux(2.4.x/2.6.x),
	CE(4.2/5.0b)
Mechanical	
Physical Dimension:	60x25x5.52mm(L*W*H)
Weight:	10g
Operating	•0° C to 55° C in Operating
Temperature:	•-20° C to 70° C in Storage
Operating Humidity.	•0% to 70% in Operating
operating numbers:	•0% to 95% Non-condensing in Storage
Warranty	1Year

注: USB 无线网卡驱动部分内容为 VIA 公司提供英文版本,本文中文介绍说明是广州朗成电子所编 写的,如有编译相差之处,以 VIA 公司的英文版本为准。

18. 三个串口使用

AT2440EVB-II 己引出三个串口,取消了红外线功能的功能。其中两个是用 DB9 标准头,分别为 COM0,COM1, COM0 支持 GPS 接收器功能,COM1 用于 WINCE5.0 调试输出,还有一个是 UART 位置的 7,8 脚,可以用 UART 的 2、7、8 三脚引线扩展为 COM2 使用。具体定义请查看原理图。

十三、Linux2.6 内核系统介绍

众所周知, Linux 在嵌入式系统中的应用已经非常普遍。为了进一步促进这方面的应用,在 Linux 2.6 中,引入了很多非常有利于嵌入式应用的功能。这些新功能包括实时性能的增强、更方便的移植性、对大容量内存的支持、支持微控制器和 I/O 系统的改进等。

嵌入式计算通常涉及到各种尺寸的计算机,其中包括从小型手持设备(比如手表、照相机)到包含成千 上万个节点的分布式系统(比如通信交换机)。嵌入式系统可能简单得只需要一个小的微控制器,也可能使 用的是大量的并行处理器和海量的内存。Linux 2.6 的改进,则对这一系列需求提供了支持。

响应时间的改进

嵌入式系统通常需要稳定的时间限制。虽然 Linux 2.6 还不是一个真正的实时操作系统,但说到其改进后的响应能力更加适合于这一领域的需求。

在 2.6 内核以前,要想让 Linux 获得更好的响应能力,就需要一些特殊的补丁。通常情况下,需要用 户从厂商处购买补丁来改进中断性能和调度反应时间。如今,2.6 内核把这些改进加入到了主流的内核当 中,因此无需再对其进行特殊的配置。

Linux 2.6 提供了一些功能,可以改进整体响应能力。其中两个改变更值得注意,首先现在使用的是 抢占式内核,其次使用的调度算法也更加高效。

抢占式内核

和其它大多数通用操作系统一样,当一个进程被系统调用并处于运行状态时,以前版本的 Linux 是不 允许进程进行调度的。这就意味着一旦系统调用中有某个任务正在执行,那么该任务就会控制处理器,直 到系统调用结束,而不管其使用处理器时间的长短。这种设计显然要简单得多,但是在很多时候这会导致 一些更重要的任务在等待系统调用完成的过程中被耽搁。

现在,内核在一定程度上使用了可抢占的模式。因此,在一些时效性比较强的事件中,Linux 2.6 要 比 2.4 具有更好的响应能力。当然了,它实际上并不是一个真正的 RTOS,但是与以前的内核相比较,"停 顿"的感觉要少得多。在Linux 2.6 内核中,代码被设置了抢占点,这就意味着调度程序会中止现在正在运 行的进程而来执行优先级更高的进程。在系统调用过程中,Linux 2.6 会定时地检查抢占点,以避免不合理 的延迟发生。而在检查过程中,调度进程很可能就会中止当前的进程来让另外一个进程运行。

有执行时间限制的软件和虚拟内存请求页面调度是不兼容的,因为这种方法处理页面慢的缺点会破坏 程序的响应能力。而 2.6 内核可以被编译成没有虚拟内存的系统来消除这个问题。当然,这就要求软件设 计人员必须考虑要有足够的实内存来运行应用程序。

图 1 和图 2 显示了 2.4.1 内核和 2.6 内核在平均情况下和最坏情况下任务的响应时间。数据来自通过 一台 PIII 1.0GHz 处理器采集的 310 万个样本。测试使用的是带有超过 5 个中断的 LynuxWorks 的实时测 试设备。系统处在由持续地磁盘数据传送、网络通信、控制台输入、图像处理和一个定时卡组成的高负载 下运行。







图 2 最差响应时间

高效的调度程序

在 2.6 版本中,进程调度经过重新编写,去掉了以前版本中效率不高的算法。以前,为了决定下一步 要运行哪一个任务,进程调度程序要查看每一个准备好的任务,并且经过计算来决定哪一个任务相对来说 更为重要。所有的计算完成之后,得分最高的任务将被选中。因为这种算法中大量的任务需要的时间一般 都不相同,一些复杂的多任务应用程序往往得不到及时的调度。

在 2.6 内核中,调度程序每次不再扫描所有的任务,而是在一个任务变成就绪状态时将其放到一个名为"当前队列"的队列之中。当进程调度程序运行时,它只选择队列中最有利的任务来执行。这样,调度就可以在一个恒定的时间里完成。当任务执行时,它就会得到一个时间段,或在其转到另外一个线程之前得到一段时间的处理器使用权。当它的时间段用完之后,任务就会被移到另外一个名为"过期"的队列中。而在该队列中,任务会根据其优先级进行排序。

从某种意义上讲,所有位于当前队列的任务都将被执行并且都将被移到"过期"队列之中。当这种事情 发生时,情况就会有所变化,队列就会被进行切换,原来的"过期"队列成为当前队列,而空的当前队列也 就变成了过期队列。由于在新的当前队列中的任务已经被排列好了,调度程序现在也可以使用简单的队列 算法,即总是取当前队列的第一个任务进行执行。不管任务是多是少,这个新的过程实际上要比老的过程 快得多。

新的同步措施

多进程应用程序有时需要共享一些资源,比如共享内存或设备。为了避免竞争的出现,程序员会使用 一个名为互斥的功能来确保同一时刻只有一个任务在使用资源。到目前为止,Linux 还是通过一个包含在 内核中的系统调用来完成互斥的实现,并由该系统调用来决定一个线程是等待还是继续执行。但当决定继 续执行时,这个耗时的系统调用就不需要了。Linux 2.6 内核支持所谓的 FUSM(Fast User-Space Mutex)。这 个新功能会检查用户的空间,查看是否会有等待的情况出现,并且只有在线程需要等待时才进行系统调用。 而当等待不需要时,就会避免不必要的系统调用以节约时间。该功能也使用优先级调度,以便在出现竞争 时决定哪一个线程可以被执行。

共享内存的改进

嵌入式系统有时也是一个有很多处理器的设备,比如在电信网络或大型存储系统中就是如此。而不论 是均衡或是松散连接的多处理器,一般都是共享内存的。均衡多进程的设计是所有的处理器都有对内存有 均等使用权,而限制使用内存的决定性因素是进程的效率。Linux 2.6 为多程序提供了一种不同的途径,即 所谓的 NUMA(Non Uniform Memory Access)。这种方法中,内存和处理器是相互连接的,但是对于每一个 处理器,一些内存是"关闭"的,而有的内存则是"更远"的。这就意味着当内存竞争出现时,"更近"的处理 器对就近的内存有更高的使用权。2.6 内核提供了一套功能来定义内存和处理器之间的拓扑关系。调度程 序可以利用这些信息来为任务分配本地内存。这减少了内存竞争造成的瓶颈,因而提高了吞吐量。

POSIX 线程、信号和计时器

POSIX 标准描述的是一套用于创建和管理 POSIX 线程的功能。这些经过精心定义的系统功能在以前的 Linux 版本中就可以用了,不过在 2.6 内核中,它得到了很大的改进。相比较而言,NPTL(Native POSIX Thread Library)的改进更为明显,甚至都已经超过了现在一些补丁中可用的高性能选择方案。

与 POSIX 线程一起,2.6 把 POSIX 信号和 POSIX 高精度计时器作为了主流内核的一个组成部分。 POSIX 信号比以前 Linux 版本中使用的 Unix 模式的信号有了很大的改进。新的 POSIX 信号不能被丢失, 并且可以携带信息作为参数。此外,POSIX 信号也可以从一个 POSIX 线程传送至另外一个线程,而不是 像 Unix 信号一样,只能从一个进程至另外一个进程。 嵌入式系统通常要求硬件能够在固定的时间安排下来运行任务。POSIX 计时器可以轻松地让任何一个任务都可以周期性地得到预定安排的时间。计时器的时钟可以达到很高的精度,从而可以让软件工程师更加精确地控制任务的调度。

支持通用设计

嵌入式世界里的硬件设计通常都要经过定制,以满足特定的应用程序。因此,设计人员经常需要使用 原始的方式来解决设计上的问题。比如,为特定目的制造的主板可能使用不同的 IRQ 管理器而不是使用类 似的设计。为了能在一块新的主板上运行,Linux 就要进行更改(迁移)以支持新的硬件。如果操作系统 是由独立的组件组成的,那么这种更改应该说是比较简单的,因为只需要更改那些需要更改的代码。在 2.6 内核中,就引入了一个名为子框架的概念。在新的定义中,各组件被清晰地分开,并且可以独立进行更改 或替换,而不会对其它的组件或软件包造成影响,或者影响非常小。

设备、总线和 I/O

现在Linux 正在变成行业用户的第一选择。2.6 内核包含了 ALSA(Advanced Linux Sound Architecture), 该体系结构可以安全地使用 USB 和 MIDI 设备。通过使用 ALSA,系统可以同时播放和记录音频。

用于支持视频的 Video4Linux 系统,在 2.6 中也焕然一新。虽然其不能向后兼容,但却可用于最新的 广播、电视、数码相机和其它的多媒体。

Linux 2.6 使用的是 USB 2.0, 它要比一般的 USB 快 40 倍。可以预见,在不久的将来,高速设备将非常普及,而在对 USB 2.0 支持方面,Linux 可以说是一个先行者。

对 64 位处理器的支持

在一些嵌入式系统中,计算机要提供非常多的资源,比如很大的内存和高吞吐量的多处理器。这些大型系统有数量众多的嵌入式应用程序,比如大规模存储系统和特殊计算引擎。

使用 2.6 内核,对于那些需要大量内存的嵌入式 Linux 开发人员就可以选择 64 位的处理器。Intel 安 腾 64 位处理器在以前的 Linux 版本中就已经被支持,而在新版本中,更是包含了 AMD64 体系结构,开始 提供对 AMD Opteron 处理器的支持。当然 PowerPC 也没有被忽略,现在 PPC64 也已经获得了支持。很显 然, Linux 社区对于大容量总线、大容量内存计算的创新问题充满了激情。

支持微控制器

现在主流的 Linux 2.6 内核中,也提供了对微处理控制器的支持。以前,大多数情况下 Linux 都需要 一个包含内存管理的全功能微处理器。然而,在嵌入式市场领域,更简单的微控制器更适于低价和简单的 应用。

Linux 的微控制项目成为了一个 Linux 应用于小型系统中非常重要的分支。这个名为 uClinux 的分支 已经受到了小型处理器开发人员的广泛关注。2.6 版本中把 uClinux 中一个非常重要的部分整合到了内核之 中,使得微控制器被主流的 Linux 所支持。

2.6 内核支持一些没有内存管理单元的微控制器,比如它支持摩托罗拉m68k的Dragonball和ColdFire、 日立的H8/300、NEC v850,以及Axis 通信公司的ETRAX网络微控制器家族。

广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II

32 位处理器上的大容量内存

有时,嵌入式系统可能需要使用通用的 Intel 架构处理器,但是却需要超出一般 32 位地址空间所能访问的内存。对此,Intel 引入了一个名为 PAE(物理地址扩展)的概念,这使得 32 位的处理器可以访问达 64GB的内存。通过 PAE, Linux 2.6 运行在更新的 x86 系统上时,可以通过一个可移动的"窗口"来查看内存。Linux 2.6 对 PAE 的支持使得它尤其适于那些需要快速处理大量数据的应用程序之中。

十四、Linux2.6.18.2 光盘目录介绍

光盘目录说明:

1.busybox-1.6.1.tar.gz 为根文件系统的源代码

- 2.linux-2.6.18.2.tar.gz 为内核源码包
- 3.root.tar 为文件系统的源代码,带QT
- 3.root_no_qt.tar.gz 为文件系统的源代码,不带 QT
- 4.opie.tar 为基于 QT 上的 OPIE 源代码
- 5.autoflash 为我们自动烧写引导的源代码
- 6.mkfs.jffs2、.config 是系统需要使用的环境,如果你编译出来的不同,可以使用此环境文件代替你的。
- 7.tools 目录下的为交叉编译工具与 FTP
- 8.应用程序 目录下的内容为一些应用软件,可移植到系统上使用
- 9.烧写文件 目录中有不同的分辨率内核与烧写文件,分 USB 与网络两种下载方式。

十五、用 ATBOOT 下载 Linux2.6.18.2 方法

ATBOOT 路径:光盘\ATBOOT\

对应烧写文件路径: Linux2.6 源代码\内核烧写文件\USB\

15.1 用前面的介绍方法烧好 ATBOOT4.02-35.bin, 在 DNW 启动进入 ATBOOT 菜单后,关于 ATBOOT 的 升级更新方法见前面的介绍,在此只介绍下载 LINUX2.6 内核的方法。见下图。

DHT v0.50A [CON4,115200bps][USB:0K]	
Serial Port USB Port Configuration Help	
http://www.avantech.com.cn	^
<*************************************	
++	
AT2440EVB USB Downloader 2007-12-3	
++	
IISB. IN ENDPOINT.1 OUT ENDPOINT.3	
FORMAT: <addr(data):4>+<size(n+10):4>+<data:n>+<cs:2></cs:2></data:n></size(n+10):4></addr(data):4>	
NOTE: 1. Power off/on or press the reset button for 1 sec	
in order to get a valid USB device address.	
2. For additional menu, Press any key.	
USB host is connected. Waiting a download.	
###### Select Menu ######	
[0] Download & run to SDRAM WinCE5.0	
[1] Download Only	
[2] Write File to NAND Flash	
[3] Erase NAND Flash Partition	
[4] HUTO FIASN WINGES.0	
[6] Config parameters	
[7] Start OS	
[8] Auto Flash Linux2.6	
	-

15.2 从菜单中选择 8,进入其子级菜单,见下图。

Serial Port USB Port Configuration Help

■ DHT v0.50Å [CON4,115200bps][USB:0K]

* Invalid BIN file signature! 45,0,0,ea,1b,0,0 Read chip id = ec76 Nand flash status = c0 Now write nand flash page 0x0 from ram address 0x31000000, filesize = 196662Program nand flash partition success ###### Select Menu ###### [0] Download & run to SDRAM WinCE5.0 [1] Download Only [2] Write File to NAND Flash [3] Erase NAND Flash Partition [4] Auto Flash WinCE5.0 [5] MemoryTest [6] Config parameters [7] Start OS [8] Auto Flash Linux2.6 Read chip id = ec76 Nand flash status = c0 Please select which region to erase : Esc to abort , size 0x40000 [ATBOOT] 0 : offset 0x0 1 : offset 0x40000 , size 0x1c0000 [KERNEL] 2 : offset 0x200000 , size 0x3500000 [ROOTFS]

15.3 选择1准备下载内核文件 zImage,见下图。

```
DET v0.50Å [COM4, 115200bps][USB:OK]
                                                                            - 🗆 ×
Serial Port USB Port Configuration Help
Invalid BIN file signature!
                                                                                 ▲|
45,0,0,ea,1b,0,0
Read chip id = ec76
Nand flash status = c0
Now write nand flash page 0x0 from ram address 0x31000000, filesize = 196662
.....Program nand flash partition success
####### Select Menu #######
 [0] Download & run to SDRAM WinCE5.0
 [1] Download Only
 [2] Write File to NAND Flash
 [3] Erase NAND Flash Partition
 [4] Auto Flash WinCE5.0
 [5] MemoryTest
 [6] Config parameters
 [7] Start OS
[8] Auto Flash Linux2.6
Read chip id = ec76
Nand flash status = c0
Please select which region to erase : Esc to abort
                  , size 0x40000
0 : offset 0x0
                                       [ATBOOT]
                     , size 0x1c0000
1 : offset 0x40000
                                       [KERNEL]
2 : offset 0x200000 , size 0x3500000 [ROOTFS]
Are you sure to erase nand flash from page 0x200, block count 0x70 ? [[]/n]
```

15.4 选择 Y, 确定擦除 nand flash。见下图。

•••• DHT ▼0.50Å [COH4,115200bps][USB:0K]	
Serial Port USB Port Configuration Help	
USB host is connected. Waiting a download.	▲
###### Select Menu ######	
[0] Download & run to SDRAM WinCE5.0	
[1] Download Uniy	
[2] Write File to NAND Flash	
[3] Erase NHNV Flash Partition	
[4] HULU FIASH WINCES.0	
[6] Config parameters	
[7] Start OS	
[8] Auto Flash Linux2.6	
Read chip id = ec76	
Nand flash status = c0	
Please select which region to erase : Esc to abort	
0 : offset 0x0 , size 0x40000 [ATBOOT]	
1 : offset 0x40000 , size 0x1c0000 [KERNEL]	
2 : offset 0x200000 , size 0x3500000 [ROOTFS]	
Are you sure to erase nand flash from page 0x200, block count 0x70 ? [y/n]	
	•••••
The transmission developed address for Linew is gradegeeee	
The temporary download address for LINUX 15 0X31000000.	
lise bost is connected Waiting a download	
USD HOST IS CONNECTED. WAITING & DOWNIDAD.	H

15.5 从 USB Port——Transmit, 打开 zImage 文件。zImage35 是 3.5'LCD 内核, zImage7,8 分别对应相应 LCD 内核文件。

•••• DHT +0.50A [CON4,115200bps][USB:0K]	
Serial Port USB Port Configuration Help	
USB host i Transmit . Waiting a download.	
###### Se] Status ####	
[0] Download a ran do SDRAM WinCE5.0	
[1] Download Only	
[2] Write File to NAND Flash	
[3] Erase NAND Flash Partition	
[4] Auto Flash WinCES.0	
[5] Memorylest	
[0] CONFIG Parameters	
[7] Start US [9] Auto Elach Liouw2 6	
$\begin{bmatrix} 0 \end{bmatrix} Huto Fiash Linux2.0$ Read chip id = ec76	
Nand flash status = c0	
Please select which region to erase : Esc to abort	
0 : offset 0x0 . size 0x40000 [ATBOOT]	
1 : offset 0x40000 , size 0x1c0000 [KERNEL]	
2 : offset 0x200000 , size 0x3500000 [ROOTFS]	
Are you sure to erase nand flash from page 0x200, block count 0x70 ? [y/n]	
Erase Nand partition completed success	
The temporary download address for Linux is 0x31000000.	
USB host is connected. Waiting a download.	

15.6 在弹出的窗口中,将文件类型选择为 "All Files(*.*)",才能看到 zImage 的文件。选择你的 LCD 相应 zImage 文件,打开即下载。

打开					? ×
查找范围(<u>I</u>):	🗀 USB		•	- 🗈 💣 🖩	-
ました 我最近的文档 () 泉面 () () () () () () () () () ()	TIMAGES	5			
	文件名(M):	zImage35		-	打开(0)
	文件类型 (I):	All Files (*	*)		取消

15.7 下载中。若只看到进度条在变化,下面的 FILE SIZE: 数字不动或看不到,代表下载失败,需要重来。

Serial Port USB Port Configuration Help	
###### Select Menu ######	_
[0] Download & run to SDRAM WinCE5.0	
[1] Download Only	
[2] Write File to NAND Flash	
[3] Erase NAND Flash Partition	
[4] Auto Flash WinCE5.0	
[5] MemoryTest	
[6] Config parameters	
[7] Start OS	
[8] Auto Flash Linux2.6	
Read chip id = Downloading D:\AT2440\Linux2.6\USB\zImage35	
Nand flash stat	
Please select w	
0 : offset 0x0	
1 : Offset UX40000 , Size UX1C0000 [KERNEL]	
2 : Offset 0x200000 , S120 0x3500000 [RUUIFS]	
Hre you sure to erase nand flash from page 0x200, block count 0x/0 ? [y/n]	
Funce Wind poutition completed success	
The temperary deveload address for Linux is 0x21000000	
The cemporary domitoad address for clinux is 0x31000000.	
Declawob a principal download	
osb nost is connected. Watcing a downioad.	
Now, Downloading [ADDRESS:31000000h.TOTAL:1308502]	
RECEIVED FILE SIZE: 327680 若此数据不动或看不到,需要重新下载。	-
	107
/ 們閉放电丁科拉有限公司 http://www.avantech.com.cn AT2440EVB-II	107

15.8 下载完后,自动跳到菜单状态。

	×
Serial Port USB Port Configuration Help	
Now, Checksum calculation	┣
Invalid BIN file signature!	
0,0,a0,e1,0,0,a0	
Read chip id = ec76	
Nand flash status = c0	
.Save boot params success.	
Read chip id = ec76	
Nand flash status = c0	
Now write nand flash page 0x200 from ram address 0x31000000, filesize = 1308502	
	-
Program nand flash partition success	
###### Select Menu ######	
[0] Download & run to SDRAM WinCE5.0	
[1] Download Only	
[2] Write File to NAND Flash	
[3] Erase NAND Flash Partition	
[4] Auto Flash WinCE5.0	
[5] MemoryTest	
[6] Config parameters	
[7] Start OS	
[8] Auto Flash Linux2.6	

15.9 继续下载文件系统 rootfs.img。选择 8,进入二级菜单。

•••• DHT +0.50Å [COH4,115200bps][USB:0K]	
Serial Port USB Port Configuration Help	
Nand flash status = c0	<u> </u>
.Save boot params success.	
Read chip id = ec76	
Nand flash status = c0	
Now write nand flash page 0x200 from ram address 0x31000000, filesize = 130	35 02
Program nand flash partition success	•••••
###### Select Menu ######	
[0] Download & run to SDRAM WinCE5.0	
[1] Download Only	
[2] Write File to NAND Flash	
[3] Erase NAND Flash Partition	
[4] Auto Flash WinCE5.0	
[5] MemoryTest	
[6] Config parameters	
[7] Start OS	
[8] Auto Flash Linux2.6	
Read chip id = ec76	
Nand flash status = c0	
Please select which region to erase : Esc to abort	
0 : offset 0x0 , size 0x40000 [ATBOOT]	
1 : offset 0x40000 , size 0x1c0000 [KERNEL]	
2 : offset 0x200000 , size 0x3500000 [ROOTFS]	
	-
15.10 选择 2,再选择 Y,擦除相应地址的 NAND FLASH。

DHT =0.50A [COM4, 115200bps][USB:0K] Serial Port USB Port Configuration Help .Save boot params success. • Read chip id = ec76 Nand flash status = c0 Now write nand flash page 0x200 from ram address 0x31000000, filesize = 1308502 Program nand flash partition success ####### Select Menu ###### [0] Download & run to SDRAM WinCE5.0 [1] Download Only [2] Write File to NAND Flash [3] Erase NAND Flash Partition [4] Auto Flash WinCE5.0 [5] MemoryTest [6] Config parameters [7] Start OS [8] Auto Flash Linux2.6 Read chip id = ec76 Nand flash status = c0 Please select which region to erase : Esc to abort 0 : offset 0x0 , size 0x40000 [ATBOOT] 1 : offset 0x40000 , size 0x1c0000 [KERNEL] 2 : offset 0x200000 , size 0x3500000 [ROOTFS] Are you sure to erase nand flash from page 0x1000, block count 0xd40 ? [IJ/n]

15.11 擦除过程需要一段时间,请稍等。

DHT v0.50A [COM4, 115200bps][USB:OK] _ 🗆 🗵 Serial Port USB Port Configuration Help [1] Download Only ▲| [2] Write File to NAND Flash [3] Erase NAND Flash Partition [4] Auto Flash WinCE5.0 [5] MemoryTest [6] Config parameters [7] Start OS [8] Auto Flash Linux2.6 Read chip id = ec76 Nand flash status = c0 Please select which region to erase : Esc to abort 0 : offset 0x0 , size 0x40000 [ATBOOT] 1 : offset 0x40000 , size 0x1c0000 [KERNEL] 2 : offset 0x200000 , size 0x3500000 [ROOTFS] Are you sure to erase nand flash from page 0x1000, block count 0xd40 ? [y/n]

15.12 擦除完成,等待下载 rootfs.img 文件系统中。

	[USB:0K]	
Serial Port USB Port Configuration	Help	
		····· •
		•••••
	Erase Nand partition completed success	
The temporary download addre	ess for Linux is 0x31000000.	
USB host is connected. Waiti	ing a download.	
		

15.13 从 USB Port——Transmit, 打开 rootfs.img。文件系统对所有分辨率的 LCD 都是一样的。

打开					<u>? ×</u>
查找范围(<u>I</u>):	🗀 USB		•	🗕 🗈 💣 🎟	
ました 我最近的文档 でした 東面 美面 教的文档 変更 我的电脑 変更 和 一 一 一 一 一 一 一 一 一 一 一 一 一	ATBOOT4.02-7 ATBOOT4.02-8 ATBOOT4.02-3 Tootfs.img zImage7 zImage8 zImage8	7 3 35			
	文件名(M):	rootfs		•	打开(0)
	文件类型 (I):	All Files (*.*)		•	取消



15.14 下载中,如出现下面的 FILE SIZE: 数字不动或看不到,代表下载失败,需要重来。

15.15 下载完后即开始检测系统类型,停顿几秒时间后即将文件写入到 nand flash 中。

DHT =0.50A [COM4, 115200bps][USB:0K] _ 🗆 🗵 Serial Port USB Port Configuration Help •Erase Nand partition completed success The temporary download address for Linux is 0x31000000. USB host is connected. Waiting a download. Now, Downloading [ADDRESS:31000000h,TOTAL:32718858] RECEIVED FILE SIZE:32718858 (652.1KB/S,50.2S) Now, Checksum calculation Download O.K. Invalid BIN file signature! 85,19,1,e0,2d,0,0 Read chip id = ec76 Nand flash status = c0 Now write nand flash page 0x1000 from ram address 0x31000000, filesize = 32718858

15.16 写入 FLASH 中需要一段时间,请等待完成的提示,见下图。



15.17	按开发板上的 RESET 重启,	启动过程需要些时间,	进入下图停止。	此时按回车进入	入命令窗口。
					· · · · · · · · · · · · · · · ·

•••• DHT v0.50A [COH4,115200bps][VSB:x]	
Serial Port USB Port Configuration Help	
Empty flash at 0x01f3405c ends at 0x01f34064	▲
Empty flash at 0x01f3406c ends at 0x01f34074	
Empty flash at 0x01f34080 ends at 0x01f34084	
Empty flash at 0x01f3408c ends at 0x01f34090	
Empty flash at 0x01f340b4 ends at 0x01f340b8	
Empty flash at 0x01f340c0 ends at 0x01f340c4	
Empty flash at 0x01f340d8 ends at 0x01f340dc	
Empty flash at 0x01f340ec ends at 0x01f340f4	
Empty flash at 0x01f34100 ends at 0x01f3410c	
Empty flash at 0x01f34134 ends at 0x01f34138	
Empty flash at 0x01f3414c ends at 0x01f34154	
Empty flash at 0x01f34158 ends at 0x01f34160	
Empty flash at 0x01f34170 ends at 0x01f34174	
Empty flash at 0x01f34188 ends at 0x01f3418c	
Empty flash at 0x01f341a0 ends at 0x01f341ac	
Empty flash at 0x01f341d8 ends at 0x01f341dc	
Empty flash at 0x01f341e4 ends at 0x01f341e8	
Empty flash at 0x01f341f0 ends at 0x01f341f4	
VFS: Mounted root (jffs2 filesystem).	
Freeing init memory: 112K	
mount /etc as ramfs	
mount /dev as tmpfs	
mounting all	
Please press Enter to activate this console.	-

15.18 进入命令行后见下图。

•••• DHT ▼0.50Å [COH4,115200bps][USB:x]					
Serial Port USB Port Configuration Help					
Empty flash at 0x01f340d8 ends at 0x01f340dc					
Empty flash at 0x01f340ec ends at 0x01f340f4					
Empty flash at 0x01f34100 ends at 0x01f3410c					
Empty flash at 0x01f34134 ends at 0x01f34138					
Empty flash at 0x01f3414c ends at 0x01f34154					
Empty flash at 0x01f34158 ends at 0x01f34160					
Empty flash at 0x01f34170 ends at 0x01f34174					
Empty flash at 0x01f34188 ends at 0x01f3418c					
Empty flash at 0x01f341a0 ends at 0x01f341ac					
Empty flash at 0x01f341d8 ends at 0x01f341dc					
Empty flash at 0x01f341e4 ends at 0x01f341e8					
Empty flash at 0x01f341f0 ends at 0x01f341f4					
VFS: Mounted root (jffs2 filesystem).					
Freeing init memory: 112K					
mount /etc as ramfs					
mount /dev as tmpfs					
mounting all					
Please press Enter to activate this console.					
BusyBox v1.4.1 (2007-03-25 03:24:25 CST) Built-in shell (ash) Enter 'help' for a list of built-in commands.					
[root@avantech /]#	•				

15.19 此时内核与文件系统己启动了,但可视界面 QT-OPIE 还没有运行,需要输入下面命令运行 QT。 [root@avantech /]# echo ''./etc/init.d/opie start'' >> /mnt/etc/init.d/rcS

Serial Port USB Port Configuration Help	
Empty flash at 0x01f340d8 ends at 0x01f340dc	▲
Empty flash at 0x01f340ec ends at 0x01f340f4	
Empty flash at 0x01f34100 ends at 0x01f3410c	
Empty flash at 0x01f34134 ends at 0x01f34138	
Empty flash at 0x01f3414c ends at 0x01f34154	
Empty flash at 0x01f34158 ends at 0x01f34160	
Empty flash at 0x01f34170 ends at 0x01f34174	
Empty flash at 0x01f34188 ends at 0x01f3418c	
Empty flash at 0x01f341a0 ends at 0x01f341ac	
Empty flash at 0x01f341d8 ends at 0x01f341dc	
Empty flash at 0x01f341e4 ends at 0x01f341e8	
Empty flash at 0x01f341f0 ends at 0x01f341f4	
VFS: Mounted root (jffs2 filesystem).	
Freeing init memory: 112K	
mount /etc as ramts	
mount /dev as tmp+s	
mounting all	
Please press Enter to activate this console.	
BusyBox v1.4.1 (2007-03-25 03:24:25 CST) Built-in shell (ash) Enter 'help' for a list of built-in commands.	
<pre>[root@avantech /]#]echo "./etc/init.d/opie start" >> /mnt/etc/init.d/rcS</pre>	•

15.20 再在命令行中输入[root@avantech /]# mnt/etc/init.d/rcS 即可运行。见下图运行内容。

Serial Port USB Port Configuration Help	
Empty flash at 0x01f341a0 ends at 0x01f341ac	_
Empty flash at 0x01f341d8 ends at 0x01f341dc	
Empty flash at 0x01f341e4 ends at 0x01f341e8	
Empty flash at 0x01f341f0 ends at 0x01f341f4	
VFS: Mounted root (jffs2 filesystem).	
Freeing init memory: 112K	
mount /etc as ramfs	
mount /dev as tmpfs	
mounting all	
Please press Enter to activate this console.	
BusyBox v1.4.1 (2007-03-25 03:24:25 CST) Built-in shell (ash)	
Enter 'help' for a list of built-in commands.	
[root@avantech /]# echo "./etc/init.d/opie start" >> /mnt/etc/init.d/rcS	
[root@avantech /]# mnt/etc/init.d/rcS	
mount /dev as tmpfs	
mounting all	
Scanning hardware	
You seem to already have a /home/root/Applications directory.	
Assuming it is the Opie Applications directory. Exiting.	
Starting Opie	
	-

15.21 启动到下面内容,一会即可见到漂亮的 OPIE 界面, 3.5 'LCD 的触摸屏即可使用了。如果你是使用 7'或 8'LCD,请照下面方法校正触摸屏。

•••• DHT ▼0.50Å [COH4,115200bps][VSB:x]	
Serial Port USB Port Configuration Help	
mount /dev as tmpfs mounting all	
Please press Enter to activate this console.	
BusyBox v1.4.1 (2007-03-25 03:24:25 CST) Built-in shell (ash)	
Enter 'help' for a list of built-in commands.	
[root@avantech /]# echo "./etc/init.d/opie start" >> /mnt/etc/init.d/rcS	
[root@avantech /]# mnt/etc/init.d/rcS	
mount /dev as tmpfs	
mounting all	
Scanning hardware	
You seem to already have a /home/root/Applications directory.	
Assuming it is the Upie Applications directory. Exiting.	
Starting Upie	
UDevice() - Found Hardware : SB22440	
UDevice() - unknown nardware - using default.	
UGlobal::creating global configuration instance.	
<pre>Kunknown>: UDevice reports transformation to be 0</pre>	
<pre><unknown>: setting UWS_DISPLAY to 'Transformed:Rotu:u'</unknown></pre>	
qt_init() - starting in daemon mode	
[root@avantech /]# ⊥	•

如果你要输入命令操作,在 DNW 中会出现些字符影响效果的,建议使用其它的工具或超级终端来进行。

十六、LINUX2.6.18.2 640*480 800*480 分辨率校正方法

由于原来的系统使用是 240*320 分辨率的触摸屏参数,在烧写 640*480 800*480 分辨率的内核后,触 摸屏需要重新校正。当然方法对任何分辨率都适合的,也可重新校正原来不准的。具体方法如下: 注:**红色字体部分是需要输入的命令**

16.1.在超级终端里,系统启动停止,出现下面信息时,按回车。

Please press Enter to activate this console.

16.2. 按下面步骤进行,删除原来的触摸屏校正参数。

BusyBox v1.4.1 (2007-03-25 03:24:25 CST) Built-in shell (ash) Enter 'help' for a list of built-in commands.

[root@avante	ch /]# <mark>ls</mark>	(查看列表)			
ls					
arm	etc		opt	sys	var
bin	home		proc	tmp	
dev	lib	mnt	sbin	usr	
[root@avante	ch /]# cd etc	(进入 ETC	C 目录)		
cd etc					
[root@avante	ch /etc]# <mark>ls</mark>	(查看列表	pointercal	触摸校正文件	F是否存在)
ls					
familiar-versio	on inetd.cont	f poi	intercal	tmp	
fstab	init.d]	profile	var	
group	inittab)	protocols		
hosts	issue		resolv.com	nf	
hosts.allow	passwd		services		
[root@avante	ch /etc]# <mark>rm</mark>	pointercal	(删除 po	intercal 文件)	
rm pointercal					
[root@avante	ch /etc]# <mark>cd /</mark>	mnt/etc	进入 mnt/e	etc 目录)	
cd /mnt/etc					
[root@avante	ch etc]# <mark>ls</mark>	(查看列表	pointercal	触摸校正文件	是否存在)
ls					
familiar-versio	on inetd.cont	f poi	intercal	tmp	
fstab	init.d]	profile	var	
group	inittab)	protocols		
hosts	issue		resolv.com	nf	
hosts.allow	passwd		services		
[root@avante	ch etc]# <mark>rm p</mark>	oointercal	(删除 po	ointercal 文件)	
rm pointercal					

此时按 RESET 键 重启开发板,进入系统后,SHARP 8'LCD 窗口会出现一个新的校正界面。四点校 正方法,然后填写其它相关的信息后,一直按 OK,最后一步按 Finish 即校正成功。在这些界面中点 OK 键有时不响应,按多几次。 校正成功后,得到的校正参数在 etc 目录中,需要将这个参数复制到 mnt/etc 下才能断电后保存,以 便下次开机时不需要再校正,具体如下:

[root@avantech	/]# <mark>ls etc</mark>	(查看生成的校正参	数文件 pointercal	是否成功)
ls etc				
familiar-version	inetd.conf	pointercal	tmp	
fstab	init.d	profile	var	
group	inittab	protocols		
hosts	issue	resolv.co	nf	
hosts.allow	passwd	services		
[root@avantech	/]# cp /etc/	pointercal /mnt/etc	(复制 pointercal	到 mnt/etc 下保存)
cp /etc/pointercal	/mnt/etc			
[root@avantech	/]# ls /mnt/	etc (查看复制 poin	ntercal 是否成功,	存在即保存生效了)
ls /mnt/etc				
familiar-version	inetd.conf	pointercal	tmp	
fstab	init.d	profile	var	
group	inittab	protocols		
hosts	issue	resolv.co	nf	
hosts.allow	passwd	services		
[root@avantech /] #			
此时己复制成功	。你可以重	自后试下是否成功。		

十七、Linux 开发环境(NFS)的快速搭建

17.1.建立文件系统目录

假定文件系统的目录为/rootfs,将 root_no_qt.tar.gz 解压到该目录。 mkdir /rootfs tar zxf root_no_qt.tar.gz -C /rootfs

17.2.配置 nfs 文件系统

修改/etc/exports 文件,格式为 共享目录 开发板 ip 地址(权限设置) cat <<EOM >/etc/exports /rootfs 192.168.1.120(rw,sync,insecure,no_root_squash) EOM 重启 nfs 服务器 /etc/init.d/nfs restart

17.3.烧写可启动内核

将待烧写的 uImage 放到 tftp 服务器指定的目录下,一般是/tftpboot(假定其 ip 为 192.168.1.100),/arm/rootfs 是服务器上存放根文件系统的 nfs 目录。进入 u-boot 的命令行状态,顺序执行下列命令: set ipaddr 192.168.1.120 set serverip 192.168.1.100 erase 0x00040000 0x001dffff tftp 0x30000000 uImage cp.b 0x30000000 0x00040000 \$(filesize) set bootargs root=/dev/nfs nfsroot=192.168.1.100:/arm/rootfs rw noinitrd init=/linuxrc ip=192.168.1.120:192.168.1.100:192.168.1.100:255.255.255.0:avantech:eth0:off console=ttySAC0,115200 set bootcmd cp.b 0x00040000 0x33000000 \$(filesize)\;bootm 0x33000000 saveenv

reset

注: 本机的 ip 要与开发板的设置一致,修改 ip 可以使用以下命令

ifconfig eth0 192.168.1.100

或者修改 bootargs,将 192.168.1.100 的主机地址换成你的主机地址。开发板要与主机设在同一网段。

17.4. 重新启动即可进入开发环境。在 linux 服务器端将编译好的程序放到/rootfs 目录下,通过 telnet 连入 开发板即可。

17.5. 执行新的程序。下面以 usb 摄像头的采集程序为例,简要说明开发环境的使用方法。

首先,将摄像头驱动 usbcam.ko 和采集程序 spcaserv 复制到/rootfs/opt/usbcam 目录下

mkdir /rootfs/opt/usbcam -p

cp usbcam.ko spcaserv /rootfs/opt/usbcam

然后进入开发板,执行

insmod /opt/usbcam/usbcam.ko

./rootfs/opt/usbcam/spcaserv -w 192.168.1.120:7070 -s 320x240 -d /dev/video0 最后可以在 windows 下执行 play.bat 即可看到从开发板采集到的图象。

十八、基于 Busybox 的根文件系统制作

制作必须的文件夹及设备文件

#mkdir rootfs
#cd rootfs
#mkdir bin dev etc lib mnt proc sbin sys root
#mkdir /etc/var
#mkdir /etc/tmp
#ln -s /etc/var /var
#ln -s /etc/tmp /tmp

从 Linux-2.6.18 开始,负责旧版本的设备管理系统 devfs 已被废除,但新版本的 udev 是一个基于用户 空间的设备管理系统。在内核启动时并不能自动创建设备节点,固需手动创建 console 和 null 两个启动过 程必须的设备节点。

#mknod -m 660 dev/console c 5 1
#mknod -m 660 dev/null c 1 3
#touch linuxrc

拷贝经过交叉编译的库文件,参数-d,保持库文件的链接关系。进入 rootfs/lib 目录下,执行下面命令: #cp -rfd /usr/local/arm/3.4.1/arm-linux/lib/*./

注: /usr/local/arm/3.4.1/ 为交叉编译工具链的存放目录,如果是网上下载的交叉编译工具链,必须将 其存放在该目录或建立相应的软链接。

补:为方便大家,我们已生成一个简易的根文件系统 root_no_qt.tar.gz

执行 tar zxf root_no_qt.tar.gz 即可,内容如下:

🛃 rootQRHAS	:/arn/sbc	2440/min	ni_rooti	s [80x	24]	
<u>C</u> onnection <u>E</u> di	t <u>V</u> iew <u>W</u> in	dow <u>O</u> ptio	n <u>H</u> elp			
[root@RHAS s	3bc2440]#	cd mini	_rootfs,	/		
[root@RHAS n	aini_rootf	s]# 11				
total 44						
drwxr-xr-x	2 root ro	ot 4096	Jul 24	17:05		
drwxr-xr-x	5 root ro	ot 4096	May 31	00:59		
drwxr-xr-x	5 root ro	ot 4096	Jul 24	19:14		
drwxr-xr-x	2 root ro	ot 4096	May 31	01:03		
drwxr-xr-x	3 root ro	ot 4096	May 31	01:03		
-rwxr-xr-x	1 root ro	ot 159	Jul 24	16:57	linuxrc	
drwx	2 root ro	ot 4096	May 31	00:59		
drwxr-xr-x	2 root ro	ot 4096	May 31	00:59		
drwxr-xr-x	2 root ro	ot 4096	Jul 24	17:05		
drwxr-xr-x	2 root ro	ot 4096	May 31	00:59		
lrwxrwxrwx	1 root ro	ot 8	May 31	00:59	tmp -> /var/tmp	
drwxr-xr-x	3 root ro	ot 4096	Jul 24	19:12	usr	
lrwxrwxrwx	1 root ro	ot 8	May 31	00:59	var -> /etc/var	
[root@RHAS r	aini_rootf	s]#			<u>x</u>	

交叉编译 BusyBox

下载最新的稳定版的 BusyBox 1.6.1 (stable), 解压缩

tar zxf busybox-1.6.1.tar.tar

解压后修改 Makefile, 指明交叉编译器:

ARCH ?= arm

CROSS_COMPILE ?= /usr/local/arm/3.4.1/bin/arm-linux-

在解压后的 BusyBox 1.6.1 目录里执行 make menuconfig ,选择系统需要用到的命令或工具。值得注意的是 shell 的选择,在 Choose your default shell (none) 这一项回车后选择 ash,否则,编译后会提示找不到 shell,使系统无法启动。如下:

Starting pid 739, console /dev/console: '/bin/sh'

sh: applet not found

Process '-/bin/sh' (pid 739) exited. Scheduling it for restart.

执行 make install 进行编译,编译完后,会出现_install 目录,包含 bin、sbin 两个目录和一个 linuxrc 文件。linuxrc 是 linux 启动后第一执行的文件(由 u-boot 传给 linux 的启动参数决定),其作用是执行相应的初始化工作。但 u-boot 编译生成的仅是一个指向/bin/busybox 的链接文件。由于我们需要作一些初始化工作,如启动 web 服务器,加载特定驱动程序等,我们将自行编写 linuxrc 这个文件。bin、sbin 这两个目录里除了 bin/busybox 一个文件外,其余都是指向 busybox 的链接文件。我们将 bin、sbin 这两个目录拷贝到 rootfs 的相应目录,基本的文件系统制作完成。

如果找不到 Linuxrc 文件,可以自建一个:

ln -s /bin/busybox linuxrc

补:为方便大家,我们已打包一个通过编译的 busybox busybox-1.6.1.tar.gz

执行 tar zxf busybox-1.6.1.tar.gz 即可,内容如下:

[root@RHAS busybox-1.6.1]# ls						
applets						
arch			README			
archival		LICENSE				
AUTHORS						
busybox		Makefile				
busybox.links		Makefile.custom				
busybox.map		Makefile.flags				
busybox_unstripped		Makefile.help		≣		
Config.in			TODO			
console-tools	INSTALL		TODO_config_nommu			
coreutils						
[root@RHAS busybox-	1.6.1]#			$\mathbf{\nabla}$		

_install 已包含编译后的 busybox 文件。执行以下命令将删除_install 目录并重新编译 busybox:

rm _install -rf

make clean

make && make install

生成镜像文件

因为这个文件系统是存放在开发版的 nand flash 中,还需经先转换为 jffs2 镜像文件,再通过 u-boot 烧写进 nand flash。(mkfs.jffs2 文件需要复制到与 rootfs 同目录下)

./mkfs.jffs2 -d rootfs -p -s 0x200 -e 0x4000 -n -l -U -o root_no_qt.img

./mkfs.jffs2 -d rootfs -p -s 0x200 -e 0x4000 -n -l -U -o rootfs.img (我们配的为带 QT)

-d 指定待转换的目录,-p 指定将所有的已擦除但未使用的块填充为 0xFF,-s 设定页的大小为 528 字节,-e 设定擦除块的大小为 16KB,-n 指明不添加清楚标记(nand flash 有自己的校检块,存放相关的 信息。),-l 指定使用小端格式,-U 将所有文件的拥有者设为 root 用户,最后-o 指明输出的镜像文件名。 生成的 root_no_qt.img 或 rootfs.img 可通过 u-boot 直接烧写至 nand flash:

nand erase 0x200000 0x3e00000

tftp 0x3000000 root_no_qt.img

nand write.jffs2 0x3000000 0x200000 \$(filesize)

要想使用 jffs2 文件系统作为根文件系统,还需给 linux 传入以下参数: root=/dev/mtdblock1 rw rootfstype=jffs2,可以通过设定 u-boot 的 bootargs 完成这项工作。

setenv bootargs root=/dev/mtdblock1 rw rootfstype=jffs2 noinitrd init=/linuxrc console=ttySAC0,115200

save

过程如下图:

```
NAND erase: device 0 offset 2097152, size 65011712 ... OK
SBC2440A# tftp 0x30000000 root_no_qt.img
<DM9000A> REG R/W TEST ===> REG R/W PASS
<DM9000A> FIFO TEST R: 0000 W: 0000 ===> FIFO TEST R/W PASS
<DM9000A> FIFO TEST R: 1000 W: 0010 ===> FIFO TEST R/W PASS
<DM9000A> FIFO TEST R: 2000 W: 0020 ===> FIFO TEST R/W PASS
<DM9000A> FIFO TEST R: 2000 W: 0020 ===> FIFO TEST R/W PASS
<DM9000A> FIFO TEST R: 3000 W: 0030 ===> FIFO TEST R/W PASS
TFTP from server 192.168.1.101; our IP address is 192.168.1.8
Filename 'root_no_qt.img'.
Load address: 0x30000000
```

擦除完毕,开始下载。



下载完毕。

SBC2440A# nand write.jffs2 0x30000000 0x200000 \$(filesize)	
NAND write: device 0 offset 2097152, size 3883008 3883008 bytes written: OK	
SBC2440A# setenv bootargs root=/dev/mtdblock1 rw rootfstype=jffs2 noinitrd init=/linuxrc console=ttySAC0,115200	
SBC2440A# save	
Saving Environment to Flash	
Un-Protected 1 sectors]	
Erasing Flash	
Erasing sector 001f0000	
done	
Erased 1 sectors	
Writing to Flashdone	
Protected 1 sectors	
SBC2440A#	

烧写完毕,一切 OK, 输入 reset 重启。

TCP bic registered NET: Registered protocol family 1 NET: Registered protocol family 17 s3c2410-rtc s3c2410-rtc: setting the system clock to 2002-01-02 12:09:25 (1009973365) VFS: Mounted root (jffs2 filesystem). Freeing init memory: 112K command='/etc/init.d/rcS' action=1 tty='' command='-/bin/sh' action=4 tty='' mount /dev as tmpfs mount /dev as tmpfs mounting all Please press Enter to activate this console.

正常启动的画面。

烧写中有一个自动引导的文件 autoflash.img, 生成方法如下:

./mkimage -A arm -O Linux -T script -C none -n autoflash -d autoflash autoflash.img

补:为方便大家,我们已生成一个简易的文件系统 root_no_qt.img

十九、Linux 内核的编译及烧写

19.1 安装交叉编译工具

在 linux 主机(Redhat/Fedora)上挂载随板光盘,将 2.6-QT/tools/arm-linux-gcc-3.4.1.tar.bz2 复制到系统根目录(/)下;

在系统根目录下,将 arm-linux-gcc-3.4.1. tar. bz2 解包,交叉编译工具生成在/usr/local/arm/目录下,解包命令:

tar vxjf arm-linux-gcc-3.4.1.tar.bz2

打开\$HOME 目录下的.bash_profile 文件,修改 PATH 脚本行为: PATH=\$PATH:/usr/local/arm/3.4.1/bin 保存.bash_profile 文件后,注销用户,再以同一用户登录,PATH 环境变量生效.

(小提示: 在终端中, . 的文件为隐藏, 需要进行下面操作才可以修改, 对新手来说可能不懂。

cd

1s -a

vi ./bash_profile 用 VI 打开进行编辑

将光标上下移动到 PATH=\$PATH:此列,按 A 进入编辑状态,按 ESC 退出,按 SHIFT +ZZ 保存退出。)

19.2 内核编译

19.2.1. 解压内核源码包

tar zxf linux-2.6.18.2.tar.gz cd linux-2.6.18.2

19.2.2. 修改交叉编译器

vi Makefile

输入 / arm-linux, 找到

CROSS_COMPILE ?= arm-linux-

这一行,将 arm-linux-修改为你交叉编译器的路径,如

/usr/local/arm/3.4.1/bin/arm-linux-

输入:wq 保存并退出。

19.2.3. 修改 LCD 驱动, 默认为 7 寸屏的驱动

.lcdcon1

vi sbz2410.c

输入 / TFT7, 找到

#define TFT7

这一行,如果是3.5寸屏,修改为TFT3,如下图;如果是8寸,则修改为TFT8。

如果是 8'LCD,因为频率对某些批次的 LCD 有影响,以前的屏参数如下:

在 sbc2440.c 里面

#ifdef TFT8

.regs={

= S3C2410_LCDCON1_TFT16BPP |

S3C2410_LCDCON1_TFT |

S3C2410_LCDCON1_CLKVAL(0x1), /* HCLK/4 */

如果编译出来用 ATBOOT 下载后,运行一段时间出现白屏或看不到显示界面,原来是 S3C2410_LCDCON1_CLKVAL(0x1)的,如果遇到这种情况可以改为 S3C2410_LCDCON1_CLKVAL(0x3) 增大频率试下。



19.2.4.需要手工更新下面内容

请照下面文件修改: 在 arch/arm/mach-s3c2410/common-smdk.c 里的 static struct s3c2410_platform_nand smdk_nand_info = { .tacls = 10, .twrph0 = 30, .twrph1 = 10, 原来是 20,60,20,但这个参数会找不到 nand flash,需要修改为上面几个。否则得到的 uImage 文件会下 广州朗成电子科技有限公司 http://www.avantech.com.cn AT2440EVB-II 125 载运行不了。

19.2.5. 编译

make clean

make uImage (uImage 内核文件为 UBOOT 方式下载使用的。)

另 ATBOOT 下载是使用 zImage,也可直接 make zImage

接着,是漫长的等待...第一次编译所花的时间较长。

编译结果如下图:

make -f scripts/Makefile.build obj=arch/arm/boot MACHINE=arch/arm/mach-s3c2410/						
arch/arm/boot/uImage						
Kernel: arch/arm/boot/Image is ready						
make -f scripts/Makefile.build obj=arch/arm/boot/compressed arch/arm/boot/compr	e					
ssed/vmlinux						
Kernel: arch/arm/boot/zImage is ready						
/bin/sh /arm/chain/linux-2.6.18.2/scripts/mkuboot.sh -A arm -O linux -T kerne	1					
-C none -a 0x30008000 -e 0x30008000 -n 'Linux-2.6.18.2' -d arch/arm/boot/zImag	2					
arch/arm/boot/uImage						
Image Name: Linux-2.6.18.2						
Created: Tue Jul 24 21:55:07 2007						
Image Type: ARM Linux Kernel Image (uncompressed)						
Data Size: 1307284 Bytes = 1276.64 kB = 1.25 MB						
Load Address: 0x30008000						
Entry Point: 0x30008000						
Image arch/arm/boot/uImage is ready						
cp -rf arch/arm/boot/uImage /arm/app						
[root@RHAS linux-2.6.18.2]#						

生成的 uImage 存放在 arch/arm/boot/uImage 目录下。将它拷贝到 tftp 的目录下准备烧写。tftp 指定的 目录, 一般是/tftpboot.

19.2.6. 内核烧写

(注:此处介绍的下载方法为旧版本的 U-BOOT.BIN 从 NOR FLASH 启动方式使用的,旧版本的 U-BOOT.BIN 的对应的源代码有些出错,目前修正不了,己放弃使用,从看到有此说明的版本起不再提供 旧的 UBOOT 源代码。目前新的 U-BOOT.BIN 版本己作大部分修改,版本号为 1.3,是从 NAND FLASH 中启动,并提供完整的 UBOOT 1.3 版本的源代码,编译工具为 3.4.1,在源代码中,上述有介绍。新的下载方式见 UBOOT 启动说明部分。)

下载内核

t 0x3000000 uImage

```
SBC2440A# t 0x30000000 uImage
<DM9000A> REG R/W TEST ===> REG R/W PASS
<DM9000A> FIF0 TEST R: 0000 W: 0000 ===> FIF0 TEST R/W PASS
<DM9000A> FIF0 TEST R: 1000 W: 0010 ===> FIF0 TEST R/W PASS
<DM9000A> FIF0 TEST R: 2000 W: 0020 ===> FIF0 TEST R/W PASS
<DM9000A> FIF0 TEST R: 3000 W: 0030 ===> FIF0 TEST R/W PASS DM9000 work in 16 bus width
Link on ethernet at:100 Mbps
Filename 'uImage'.
Load address: 0x30000000
done
Bytes transferred = 1291248 (13b3f0 hex)
SBC2440A# 🔳
```

19.2.6. 擦除 Nor Flash

erase 0x40000 0x1dffff 将内核烧写至 Nor Flash Erasing sector 00190000

```
Erasing sector 001a0000
Erasing sector 001b0000
Erasing sector 001c0000
Erasing sector 001d0000
done
Erased 26 sectors
SBC2440A# cp.b 0x30000000 0x400000 $(filesize)
Copy to Flas/done
SBC2440A#
```

19.2.7. 设置启动命令并保存

set bootcmd cp.b 0x00040000 0x30000000 \$(filesize)\;bootm 0x30000000 saveenv

至此,内核烧写完毕,请参考《基于 Busybox 的根文件系统制作》或《Linux 开发环境(NFS)的快速 搭建》设置根文件系统。

二十、u-boot.bin 启动文件的烧写方法

注: Linux2.6.18.2 的引导文件 u-boot.bin 下载方式,需要将开关1、2和4拨到"ON"方向的位置, 开关3在"OFF"方向的位置的 NAND FLASH 启动方式。COM0 输出信息。

更新说明:最新的 u-boot.bin 是 2007.12.17 日期的,与旧版本的完全不同,启动方式是从 NAND FLASH 中启动,同时源代码为 u-boot-1.3.0-rc3.tar.gz。编译工具为 arm-linux-gcc-3.4.1.tar.bz2,方法见上述。

旧版本的 u-boot. bin 是从 NOR FLASH 中启动,但所对应的源代码编译起来总存在些出错,目前己放 弃了修正与升级,并不再提供此版本的源代码。但旧的 u-boot. bin 会放在光盘中让有需要用 NOR FLASH 启动的用户直接使用,文件名为 u-boot-norflash. bin。不过对应的下载内核文件我们不再提供在光盘中 了,需要的用户可邮件向我们索取。

20.1、SJF2440环境设置(见前面部分图文介绍)

使用 SJF2440 下载板需要先安装其驱动,驱动路径: AT2440EVB-II\工具软件\GIVEIO。

SJF2440.exe 要在 WindowsNT/2000/XP 上直接把并口作为 IO 口访问, 需要安装 GIVEIO.SYS, 下面以 WindowsXP 为例介绍如何安装:

(1)以 administrator 身份登录系统

(2)复制 giveio.sys 到% systemroot% \system32 \drivers, 其中% systemroot% 是 Windows XP 的系统目录

(3)打开"控制面板",选择"添加/删除硬件" (注:需要切换到经典视图 才出现"添加/删除硬件"选项)

(4)系统将会自动搜索添加硬件,之后选择"手动添加新硬件"

(5)选择"从列表或指定位置添加"

(6)选择'浏览...',找到 GIVEIO.inf 所在的位置

(7)按照提示操作直到安装完毕

安装是否成功,请到"系统属性--设备管理器--端口"中,展开+,会看到有 giveio(COMx),x 代表串口多少。

20.2、烧写 u-boot.bin 到 Nand flash 中

u-boot.bin 路径:光盘\uboot 及源代码\

- (1) 点击 XP 开始菜单,单击运行,输入 cmd 按确定即打开命令行,输入如下图命令(红线部分)进入 sjf2440.exe 所在目录(注: sjf2440.exe 和 u-boot.bin 必须在同一目录下)。
- (2) Sjf2440 /f:u-boot.bin (红线部分,下同) 回车若出现如下画面即表示成功检测到 2440 目标板。



(3)照下图三步,都是选择 0,第一二步选择 0 为写入 NAND FLASH 中;第三步选择 0 是烧入到 FLASH 的第 0 块位置。



注:若出现提示找不到 CPU,大部分问题是在 6PIN 的线接触不良引起的,请压紧。或者是你的并口(打印口)有问题或驱动没有安装正常。

(4)用 JTAG 下载的时间需要比较长,请耐心等待。当出现下面菜单时,己代表完成。见下图:

C:\VINDOWS\system32\cmd.exe	- 🗆 X
[K9S1208 NAND Flash JTAG Programmer] K9S1208 is detected. ID=0xec76	-
0:K9S1208 Program 1:K9S1208 Pr BlkPage 2:Exit	
Select the function to test :0	
[SMC <k9s1208v0m> NAND Flash Writing Program]</k9s1208v0m>	
Source size:Oh~1da9fh	
Available target block number: 0~4095	
Input target block number:0	
target start block number =Ø	
target size (0x4000×n) =0x20000	
STATUS : Ερρρορορορορορορορορορορορορορο	
Ερορορορορορορορορορορορορορορο	
Ερορορορορορορορορορορορορορορο	
Ερορορορορορορορορορορορορορο	
0:K9S1208 Program 1:K9S1208 Pr BlkPage 2:Exit	
Select the function to test :4	
D:\atboot>	-

注:若提示找不到 AM29LV800,原因可能是 SW2 开关没有设置正确,另一原因可能是 JTAG 线接触 不良引起的。

选择4退出即己完成。

此时取出 JTAG 接口转接板线,将串口线插到开发板上的 COM0,打开超级终端,设置相关参数再 打开开发板的电源,即可看到 UBOOT 的启动信息。从超级终端可以看到如下画面(如果串口没有数据输 出,并且检测硬件没有别的问题时,则可能是 u-boot 烧写失败,需要重烧新烧写)。

毎秒位数 (B):	115200	*
数据位 @):	8	~
奇偶校验(£):	无	*
停止位 (S):	1	~
数据流控制 (2):	无	*
	还原为默认	值(图)

```
    ▲ AT2440 - 超级终端
    文件で) 編辑で) 查看 (火) 呼叫で) 传送で) 帮助(水)
    □ 逆 (※) ③ □ 凸 (ご)
    □ (□) ③ ③ □ 凸 (ご)
    □ □ ○ (□) ③ (□) □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○ (□)
    □ ○
```

此时在 Hit any key to stop autoboot:3 这时倒计时 3 秒时按下 SPACE 空格键,即出现上面的 AVANTECH # 的内容。

▲ AT2440 - 提级终端
 文件 (2) 編輯 (2) 查看 (2) 呼叫 (2) 传送 (1) 帮助 (2)
 ○ ② ③ □ □ □ □
 □ □ □
 □ □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □
 □ □

二十一、内核用 UBOOT 方式自动烧写方法

我们提供己完成的内核路径: AT2440EVB-II\Linux2.6 源代码\内核烧写文件\Network\, uImage 默认为夏普 3.5 'LCD 的内核。有分三种不同分辨率的内核,如 uImage7 代表是日立 7'LCD 的内核, uImage8 为夏普 8'LCD 的内核,请根据你手上的 LCD 选择不同的内核文件。

方法:将默认 uImage 更名为 uImage35,将你需要下载的内核取消后面的数字,修改为下载要使用的 uImage 文件名。

21.1. 将相应的 autoflash.img, rootfs.img, uImage, tftp32.exe 这四个文件复制到 D:\AT2440QT26(目录为自己新建的,但不能有中文目录)下,打开 XP 系统自带的超级终端(推荐使用超级终端,LINUX 系统烧写不建议使用 DNW.EXE, DNW 不能粘贴所复制的内容。),路径为:开始一一程序——附件——通讯——超级终端。



将超级终端的属性配置设置为下面的选项的一致: 115200-8-1-无

C0■4 尾性 端口设置			<u>্</u> যুস
每秒位数 (B):	115200		.
数据位 (L):	8		•
寄偶校验 (E):	无		•
停止位 (S):	1		•
数据流控制(E):	无		-
		还原为默い	(ā (s)
	确定	取消	应用 (A)

21.2.准备工作:将我们配的交叉串口线、交叉网线连接好开发板与 PC 电脑, PC 电脑的网络属性一定要设置相应的 IP 地址与网关。可根据你的网络来设置,开发板的 IP 到时设置为与你的网络为同一段。

注:不能使用"自动获得 IP 地址"这个选项。

Internet 协议(TCP/IP) 雇性	🗶 🎘 🍋 🕙 🗶
常规	
如果网络支持此功能,则可以获 您需要从网络系统管理员处获得	取自动指派的 IP 设置。否则, 适当的 IP 设置。
○ 自动获得 IP 地址 @)	
┌● 使用下面的 IP 地址(S): —	
IP 地址(L):	192 . 168 . 1 . 21
子网掩码 (U):	255 . 255 . 255 . 0
默认网关 (2):	192 .168 . 1 . 1
C 自动获得 DNS 服务器地址(B)
┌── 使用下面的 DNS 服务器地t	ut (E):
首选 DNS 服务器(P):	· · ·
备用 DNS 服务器(A):	· · ·
	高级 (٧)
	确定

21.3.将开发板的电源开关打开,即可在超级终端中看到 UBOOT 的启动信息,在 Hit any key to stop autoboot:3 这时倒计时 3 秒时按下 SPACE 空格键,即出现下面的 AVANTECH #的内容。

21.4.很多第一次使用开发板的用户还没有看清楚此行,就跳过去了,如果没有及时按空格键进入到命令行,请重启后再操作,或者一直按着 SPACE 空格键,再开机都可以。

🥙 AT2440 - 超级终端 文件 (2) 编辑 (2) 查看 (2) 呼叫 (2) 传送 (2) 帮助 (2)

D 😅 🍘 🌋 🗈 🎦 😭

U-Boot 1.3.0-rc3 (Dec 15 2007 - 22:10:35) DRAM: 64 MB dm9000 i/o: 0x18000300, id: 0x90000a46 MAC: 00:0c:20:02:0a:5b NAND: 64 MiB In: serial Out: serial Err: serial Hit any key to stop autoboot: 0 AVANTECH #_

注:如果开发板己烧过 UBOOT.BIN 的启动文件,不需要再重烧 UBOOT,开机后即可自动出现上述的启动信息。

21.5. 此时再打开下面的 FTP 工具,tftp32.exe,设置好内核所在路径,下面的 IP 地址是自动读取你 PC 电脑的 IP,如果显示为 127.0.0.1,代表你的 PC 电脑没有设置成功,同时检查下开发板是否己通电,网 卡绿灯是否亮。

👋 Iftpd32 by Ph. Jounin	
Current Directory D:\AT2440\QT26-Net Server interfaces 192.168.1.21	<u>B</u> rowse Show <u>D</u> ir
Current Action Listening on port 69	
About <u>S</u> ettings	<u>H</u> elp

21.6.UBOOT 的烧写方法:

步骤一:

设置开发板 IP 地址: AVANTECH# set ipaddr 192.168.1.7 (回车)

设置 TFTP 服务器 IP 地址: AVANTECH# set serverip 192.168.1.100 (此为连接开发板的 PC 电脑 IP, 设置后回车)

清除 nand flash: AVANTECH# nand erase 0x50000 0x3fb0000 (回车) 广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II 🏶 AT2440 - 超级终端

文件 (E)编辑 (E)查看 (V)呼叫 (E)传送 (E)帮助 (E) D 🚅 📨 🌋 🗈 💾 😭

U-Boot 1.3.0-rc3 (Dec 15 2007 - 22:10:35) DRAM: 64 MB dm9000 i/o: 0x18000300, id: 0x90000a46 MAC: 00:0c:20:02:0a:5b NAND: 64 MiB In: serial serial Out: serial Err: Hit any key to stop autoboot: Й 设置开发板IP地址 AVANTECH # set ipaddr 192.168.1.7 AVANTECH # set serverip 192.168.1.21 设置TFTP服务器IP地址 AVANTECH # nand erase 0x50000 0x3fb0000_清除要保存的部分 nand flash区域 已连接 0:07:20 自动检测 115200 8-N-1 SCROLL CAPS NUM | 捕 | 打印

在 nand erase 0x50000 0x3fb0000 回车后,会出现下面的信息,OK 后即代表擦除成功。或出现 ERROR 的错误,代表 NAND FLASH 中有坏块,如果坏块出现在前面位置部分,这样就没有办法下载内核的,只能更换 FLASH。我们曾写完跳过坏块的代码,但成功率不高。这就是 UBOOT 的缺点之一,建议使用 ATBOOT 下载的方法进行。

特别声明:

使用 UBOOT 下载使用时,如果在 nand erase clean 命令中出现 ERROR 的错误提示,会导致内核下载 不成功的。对此我们不负责更换 NAND FLASH,不会对 UBOOT 跳过坏道的功能进行修改升级,对 UBOOT 的修改我们能力有限,解决不了跳过坏道的问题。因为三星公司出厂 NAND FLASH 就规定允许有坏道存 在的,对第三方的硬件质量允许范围内,我们也只能接受这个事实的存在。

如遇到这个情况,建议使用 ATBOOT 下载, ATBOOT 中增加了可以自动识别跳过坏道的功能。

🌺 AT2440 - 超级终端 文件 (2) 编辑 (2) 查看 (2) 呼叫 (2) 传送 (1) 帮助 (3)

D 🚅 🍏 🔏 🗈 🎦 😭

U-Boot 1.3.0-rc3 (Dec 15 2007 - 22:10:35) DRAM: 64 MB dm9000 i/o: 0x18000300, id: 0x90000a46 MAC: 00:0c:20:02:0a:5b NAND: 64 MiB In: serial Out: serial Err: serial Hit any key to stop autoboot: 0 AVANTECH # set ipaddr 192.168.1.7 AVANTECH # set serverip 192.168.1.21 AVANTECH # nand erase 0x50000 0x3fb0000 NAND erase: device 0 offset 0x50000, size 0x3fb0000 Erasing at 0xbc4000 -- 18% complete._

步骤二:

下载自动引导文件 autoflash.img。

AVANTECH# t 33000000 autoflash.img (回车)

▲12440 - 超级终端
 文件(12) 編辑(12) 查看(12) 呼叫(12) 传送(12) 帮助(12)
 □ □ □ □ □ □ □ □
 □ □ □ □
 □ □ □
 □ □

U-Boot 1.3.0-rc3 (Dec 15 2007 - 22:10:35) DRAM: 64 MB dm9000 i/o: 0x18000300, id: 0x90000a46 MAC: 00:0c:20:02:0a:5b NAND: 64 MiB In: serial Out: serial Err: serial Hit any key to stop autoboot: 0 AVANTECH # set ipaddr 192.168.1.7 AVANTECH # set serverip 192.168.1.21 AVANTECH # nand erase 0x50000 0x3fb0000 NAND erase: device 0 offset 0x50000, size 0x3fb0000 Erasing at 0x3ffc000 -- 100% complete. 0K AVANTECH # t 33000000 autoflash.img_ 已连接 0:09:28 自动检测 115200 8-N-1 SCROLL | 捕 | 打印 NUM

在上图的 t 33000000 autoflash.img 位置回车后,网络连接正常时会出现下载成功,见下图的位置。如果一直出现 T T T T,代表网络没有连接成功,解决办法:

一是看上面的一行 TFTP from server 192.168.1.21; our IP address is 192.168.1.7 这部分显示的 IP 与你的 PC 电脑上 IP 是否一致,开发板上设置的是否一致。

二是网线是否插上,绿灯是否亮,请检查并插好水晶头。

三是上述所讲到的四个文件是否在同一目录中,所建的文件夹是否有中文,不允许建立中文的目录。

🏶 AT2440 - 超级终端 文件 (E) 编辑 (E) 查看 (Y) 呼叫 (C) 传送 (T) 帮助 (H) 0 🗃 🗑 🕉 🗈 🗃 dm9000 i/o: 0x18000300, id: 0x90000a46 MAC: 00:0c:20:02:0a:5b NAND: 64 MiB In: serial serial Out: Err: serial Hit any key to stop autoboot: 0 AVANTECH # set ipaddr 192.168.1.7 AVANTECH # set serverip 192.168.1.21 AVANTECH # nand erase 0x50000 0x3fb0000 NAND erase: device 0 offset 0x50000, size 0x3fb0000 Erasing at 0x3ffc000 -- 100% complete. 0K AVANTECH # t 33000000 autoflash.img dm9000 i/o: 0x18000300, id: 0x90000a46 MAC: 00:0c:20:02:0a:5b TFTP from server 192.168.1.21; our IP address is 192.168.1.7 Filename 'autoflash.img'. Load address: 0x33000000 Loading: # done Bytes transferred = 735 (2df hex) AVANTECH # 已连接 0:10:00 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印

步骤三:

烧写内核,以下为自动烧写内核与文件系统,期间还没见到界面时请不要断电。

1) AVANTECH# autoscr 33000000 (回车)

```
🏶 🗛 🗛 🕹
                                                                                                              文件 (2) 编辑 (2) 查看 (2) 呼叫 (2) 传送 (2) 帮助 (4)
0 🗃 👘 🐉 👘 🖓
  dm9000 i/o: 0x18000300, id: 0x90000a46
MAC: 00:0c:20:02:0a:5b
  NAND: 64 MiB
           serial
  In:
           serial
  Out:
  Err:
           serial
  Hit any key to stop autoboot: 0
AVANTECH # set ipaddr 192.168.1.7
AVANTECH # set serverip 192.168.1.21
AVANTECH # nand erase 0x50000 0x3fb0000
  NAND erase: device 0 offset 0x50000, size 0x3fb0000
Erasing at 0x3ffc000 -- 100% complete.
  UK
  ÄVANTECH # t 33000000 autoflash.img
  dm9000 i/o: 0x18000300, id: 0x90000a46
  MAC: 00:0c:20:02:0a:5b
TFTP from server 192.168.1.21; our IP address is 192.168.1.7
Filename 'autoflash.img'.
  Load address: 0x33000000
  Loading: #
  done
  Bytes transferred = 735 (2df hex)
  AVANTECH # autoscr 33000000_
已连接 0:10:34 自动检测 115200 8-14-1 SCROLL CAPS NOM 捕 打印
```

中间需要等待一段时间,当出现下图的内容后,正在将内核写到 NAND FLASH 中,还需要等一段时间才可以。

🏶 AT2440 - 超级终端 文件 @) 编辑 @) 查看 (Y) 呼叫 @) 传送 @) 帮助 @) 🗅 🗃 🏐 🔏 👘 🚰 Hit any key to stop autoboot: AVANTECH # set ipaddr 192.168.1.7 AVANTECH # set serverip 192.168.1.21 AVANTECH # nand erase 0x50000 0x3fb0000 NAND erase: device 0 offset 0x50000, size 0x3fb0000 Erasing at 0x3ffc000 -- 100% complete. **UK** AVANTECH # t 33000000 autoflash.img dm9000 i/o: 0x18000300, id: 0x90000a46 MAC: 00:0c:20:02:0a:5b TFTP from server 192.168.1.21; our IP address is 192.168.1.7 Filename 'autoflash.img'. Load address: 0x33000000 Loading: # done Bytes transferred = 735 (2df hex) AVANTECH # autoscr 33000000 ## Executing script at 33000000 ===== Power by Weibing ===== ===== erasing nand flash ===== NAND erase: device 0 offset 0x50000, size 0x3fb0000 Erasing at 0xef4000 -- 23% complete._ 已连接 0:02:13 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕打印

2) 自动下载 uImage 内核文件,见下图。

```
🍓 AT2440 - 超级终端
                                                                               文件 (E) 编辑 (E) 查看 (Y) 呼叫 (C) 传送 (E) 帮助 (H)
🗅 🖆 📨 🕈 👘 😭
 done
 Bytes transferred = 735 (2df hex)
 AVANTECH # autoscr 33000000
 ## Executing script at 33000000
 ===== Power by Weibing =====
 ===== erasing nand flash =====
 NAND erase: device 0 offset 0x50000, size 0x3fb0000
Erasing at 0x3ffc000 -- 100% complete.
 0K
 ===== writing Linux Kernel =====
 dm9000 i/o: 0x18000300, id: 0x90000a46
 MAC: 00:0c:20:02:0a:5b
TFTP from server 192.168.1.21; our IP address is 192.168.1.7
Filename 'uImage'.
 Load address: 0x30000000
 *********
 done
 Butes transferred = 1310552 (13ff58 hex)
 NAND write: device 0 offset 0x50000, size 0x1b0000
 Writing data at 0xc0400 -- 26% complete.
已连接 0:02:21 自动检测 115200 8-14-1 SCROLL
                              CAPS
                                       | 捕 | 打印
                                   NUM
```

3) 自动下载 rootfs.img 文件系统。



4)下图显示的是己下载完成,开始写入 NAND FLASH 中。

4	AT2	140 - 超	级终端					
Ĵ	件便	编辑(E)	查看(V)	呼叫(C)	传送 (I)	帮助(H)		
۵) 🖻	🐡 🏅	=D 🎦	P				
Г								[🔼
			#####	****	****	₩₩₩₩₩₩		
			****	*######	******	******	***************************************	
			######		*****			
			#####	#####	#####	######	*****	
			#####	#####	****	₩₩₩₩₩₩		
			#####	*######	*#####	******	***************************************	
			######		*****	******	***************************************	
			#####	#####	#####	######	*****	
			#####	#####		######		
			#####	*######	*****	*****		
			****	******	!###### !######	******	***************************************	
			#####	***	****	######	****	
			#####	#####	#####	######	*****	
			#####	***	****	₩₩₩₩₩₩		
	don	0	HHHH	*****	ннанн	*****	****	
	Bvt	es tra	nsfer	red =	32718	848 (1	f34000 hex)	
	NAN	D writ	te: de	vice () offs	et Øx2	00000, size 0x2000000	
	hleri	tina d	lata a	+ 0.24	5-00	3'	% complete	
	mi 1	ting t			5000	J.	w comprete.	
2	连接 0	:02:50 自著	动检测	115200 8-1	-1 SCRO	LL CAPS	s xum 捕打印	

5) 上步写入完成后,系统会自动重启,请不要断开电源,重启后开始读文件运行。

AT2440 - 超级终端				
文件(22) 編輯(22) 查看(2) 呼叫(2) 传送(2) 帮助(2)				
Bytes transferred = 32718848 (1f34000 hex)				
NAND write: device 0 offset 0x200000, size 0x2000000				
Writing data at 0x21ffe00 100% complete. 33554432 bytes written: OK ===== setting bootargs ===== Saving Environment to NAND Erasing NandWriting to Nand done ===== flash complete! reboot now =====? U-Boot 1.3.0-rc3 (Dec 15 2007 - 22:10:35)				
DRAM: 64 MB dm9000 i/o: 0x18000300, id: 0x90000a46 MAC: 00:0c:20:02:0a:5b NAND: 64 MiB In: serial Out: serial Err: serial Hit any key to stop autoboot: 0				
NAND read: device 0 offset 0x50000, size 0x13ff58				
Reading data from 0xd3200 4 <u>0</u> % complete.				
已连接 0:03:42 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印	.:			

6) 启动过程中会进行网络配置等, 会停留一段时间。

🏶 AT2440 - 超级终端
文件(2) 編辑(2) 査看(2) 呼叫(2) 传送(2) 帮助(2)
USB Mass Storage support registered. usbcore: registered new driver usbhid drivers/usb/input/hid-core.c: v2.6:USB HID core driver mice: PS/2 mouse device common for all mice ts: Compaq touchscreen protocol output s3c-buttons s3c-buttons: successfully loaded input: s3c-buttons as /class/input/input0 s3c2410 TouchScreen successfully loaded input: s3c2410 TouchScreen as /class/input/input1 S3c24XX RTC, (c) 2004,2006 Simtec Electronics s3c2410-rtc s3c2410-rtc: rtc disabled, re-enabling s3c2410-rtc s3c2410-rtc: rtc core: registered s3c as rtc0 s3c2410-rtc s3c2410-rtc: rtc core: registered s3c as rtc0 s3c2410-rtc s3c2410-rtc: initialisation done. UDA1341 audio driver initialized TCP bic registered NEI: Registered protocol family 1 NEI: Registered protocol family 17 s3c2410-rtc s3c2410-rtc: hctosys: invalid date/time IP-Config: Complete:

7) 接着进行第一次扫描 NAND FLASH 的漫长过程,第一次烧写后,系统要全部扫描过 FLASH 进行检测 才停止。大概等待几十秒。

🌯 🗛 🕹 🕹	终端			
文件(E)编辑(E)查: 	看(V) 呼叫(C)	传送(T) 帮助(H)		
D 🖻 💮 🌋 🗈	<mark>ĕ</mark> 🗳			
Empty flash	at 0x01f	45dac ends	at 0x01f45db0	
Empty flash	at 0x01f	43abo enas 65dcc ends	at 0x011450C8 at 0x01f65dd0	
Empty flash	at 0x01f	45dd4 ends	at 0x01f45de0	
Empty flash	at 0x01f	45de8 ends	at 0x01f45df0	
Empty flash	at 0x01f	45df4 ends	at 0x01f45e00	
Emply flash	at 0x011	4Jevo ends 45e2c ends	at 0x0114Je20 at 0x01f45e34	
Empty flash	at 0x01f	45e38 ends	at 0x01f45e44	
Empty flash	at 0x01f	45e48 ends	at 0x01f45e50	
Empty flash	at 0x01f	45e54 ends	at 0x01f45e60	
Empty flash	at 0x011	4Jeoc ends 45e7c ends	at 0x01145e74 at 0x01f45e80	
Empty flash	at 0x01f	45e84 ends	at 0x01f45e88	
Empty flash	at 0x01f	45e8c ends	at 0x01f45e94	
Empty flash	at 0x01f	45e98 ends	at 0x01145ea0 at 0x01165ca8	
Empty flash	at 0x01f	45eac ends	at 0x01f45ebc	
Empty flash	at 0x01f	45ec4 ends	at 0x01f45ee8	
Empty flash	at 0x01f	45eec ends	at 0x01f45ef8	
Empty flash	at UXUIT	43erc ends 65f06 ends	at 0x01145100 at 0x01f65f0c	
Empty flash	at 0x01f	45f18 ends	at 0x01f45f30	
Empty flash	at 0x01f	45f34 ends	_	
已连接 0:03:51 目动检社	씨 115200 8-N-	-1 SUKULL CAP	S NOW TH STEP	
	广州朗成	电子科技有限公	可 <u>http://www.avantech.com.cn</u>	AT2440EVB-II 1

8)当出现下图的部分,内核己烧写完成,但还没有将图形界面启动。回车后,继续下一步。

🌯 🗛 🗤 🕹 🖉	
文件 (E) 编辑 (E) 查看 (Y) 呼叫 (C) 传送 (I) 帮助 (H)	
D 🚔 🍘 🖧 🗈 🖰 📸	
	_1^
Empty flash at 0x01fffefc ends at 0x01ffff00	
Empty flash at 0x01ffff04 ends at 0x01ffff0c	
Empty flash at 0x01ffff14 ends at 0x01ffff1c	
Empty flash at 0×01ffff20 ends at 0×01ffff44	
Empty flash at 0x01ffff48 ends at 0x01ffff50	
Empty flash at 0x01ffff5 ands at 0x01ffff5(
Empty flash at 0x011113C ends at 0x0111104	
Empty flash at 0x01fff70, ends at 0x01fff78	
Empty flash at 0x01ffff7c ends at 0x01ffff80	
Empty flash at 0x01ffff84 ends at 0x01ffff88	
Empty flash at 0x01ffff90 ends at 0x01ffffa0	
Empty flash at 0x01ffffa4 ends at 0x01ffffac	
Empty flash at 0x01ffffb0 ends at 0x01ffffc4	
Empty flash at 0×01ffffc8 ends at 0×01ffffcc	
Empty flash at 0×01ffffd0 ends at 0×01ffffd8	
Empty flash at 0x01ffffdc ends at 0x01ffffe0	
Eroping init momentus 112V	
mount / otc as ranfs	
mount / deu as tents	
mounting all	
Please press Enter to activate this console	
已连接 0:12:15 自动检测 115200 8-N-1 SCROLL CAPS JUN	

步骤四:

输入下面两条命令,将 OPIE 图形界面跑起来。

[root@avantech/]# echo "./etc/init.d/opie start" >> /mnt/etc/init.d/rcS

[root@avantech /]# mnt/etc/init.d/rcS

回车后即可,请注意大小写,空格等。建议直接复制到超级终端中粘贴即可。

```
🏶 AT2440 - 超级终端
文件 @) 编辑 @) 查看 (V) 呼叫 @) 传送 @) 帮助 (H)
D 🛩 💿 🌋 🗈 🗃 😭
  Empty flash at 0x01ffffc8 ends at 0x01ffffcc
 Empty flash at 0x01ffffd0 ends at 0x01ffffd8
Empty flash at 0x01ffffd0 ends at 0x01ffffd8
Empty flash at 0x01ffffdc ends at 0x01ffffe0
VFS: Mounted root (jffs2 filesystem).
Freeing init memory: 112K
mount /etc as ramfs
mount /dou as tmpfc
  mount /dev as tmpfs
  mounting all
  Please press Enter to activate this console.
  BusyBox v1.4.1 (2007-03-25 03:24:25 CST) Built-in shell (ash)
            'help' for a list of built-in commands.
  Enter
  [root@avantech /]# echo "./etc/init.d/opie start" >> /mnt/etc/init.d/rcs
[root@avantech /]# mnt/etc/init.d/rcs
 mount /dev as tmpfs
mounting all
Scanning hardware
 You seem to already have a /home/root/Applications directory.
Assuming it is the Opie Applications directory. Exiting.
  Starting Opie...
                                                              捕|打印
已连接 0:10:03 自动检测
                       115200 8-N-1 SCROLL
                                                CAPS
                                                       NUM
```

在上图位置回车后,出现下面的信息,等待几秒时间,然后从 LCD 上可以看到 OPIE 的图形界面。可以超级终端中直接输入命令进行管理与操作。



以上步骤己将 LINUX2.6 的内核烧成功。

如果你要更换为其它不同分辨率的内核,方法与240*320分辨率的一样。只是内核与文件系统不同而 己。但需要注意的一点是其它两个分辨率(640*480,800*480)的内核烧好后,触摸屏的参数需要重新校

正,因为系统默认的是240*320的触摸屏参数。请参考上面介绍如何校正触摸屏的参数的内容。

二十二、Linux2.6.18.2 的部分应用

路径: Linux2.6 源代码\应用程序\

1、OPIE 源代码的使用:

开发所需的文件已打包为一个叫 opie 的压缩包。 qt/embedded 在 opie/qt-2.3.10 的文件夹里, tmake 也在其中。 qt-x11 一般不用在嵌入式环境中,用的就是 qt/embedded。 我们采用开源的手机图形系统 opie 来取代 qtopia-2 这个比较老的图形系统。

解开 opie 压缩包后,在 x-windows 环境中,输入 source opie_env ./qvfb-qt2 & ./bin/qpe 即可在 x86 平台下进行仿真测试。

2、新的文件系统带有一个 telnet 服务器,连入开发板并不需要串口线。在 windows 的命令提示符下输入: telnet 客户机的 ip 99 会弹出提示: avantech login:

输入用户名 root 即可连入开发板。如下:

BusyBox v1.4.1 (2007-03-25 03:24:25 CST) Built-in shell (ash)

Enter 'help' for a list of built-in commands.

[root@avantech /]#

刚烧写完的 Linux 默认是不会进入到 opie 的图形界面,确保 3.5 寸显示屏连接正确后,键入: [root@avantech /]# /etc/init.d/opie start

如需开发板启动后自动载入该图形界面,可执行

[root@avantech /]# echo "/etc/init.d/opie start" >> /mnt/etc/init.d/rcS 新的文件系统存放在 64M 的 nand flash 中,为保证文件系统的完整性和 flash 的寿命,请在关机前先执行 halt 命令。

3、新的文件系统还带有一个 mp3 播放器,可执行下面的命令来测试音频驱动是否正常,注意是使用网线来传输的。

/opt/madplay /opt/happy.mp3

4、SD 卡读写命令:(注:需要先插入 SD 卡再开机,U 盘等一样操作)
 Mount /dev/mmcblk0p1 /mnt
 Ls /mnt

5、关于网络配置的设置

由于 Qt 的设置文件还没设好,故不能在 Qt 的界面下设置网络配置。 在 shell 下使用下面两条命令可以修改网络配置。

ifconfig eth0 192.168.1.100 #设置开发板的 ip 地址
route add default gw 192.168.1.1 #设置网关 ip

若需将此设置固定,把这两条命令添加到/linuxrc或/etc/init.d/rcS即可:

echo "ifconfig eth0 192.168.1.100 #设置开发板的 ip 地址" >> /etc/init.d/rcS echo "route add default gw 192.168.1.1 #设置网关 ip" >> /etc/init.d/rcS

为防止 flash 的频繁写操作,我将/mnt/etc 目录在启动时复制到/etc 目录。(参见 linuxrc 文件) 所以,最后应将修改后的配置文件/etc/init.d/rcS 替换原来的/mnt/etc/init.d/rcS

cp /etc/init.d/rcS /mnt/etc/init.d/rcS

注:本系统采用保护 NAND FLASH 的做法,一般修改只保存在 SDRAM 中,断电后会掉失。如果你要将数据或修改的参数保存到系统中,请参考上面的方法,将参数也写入到 mnt/etc.

6、usb 键盘作标准输入

烧写新的内核和文件系统后,开发板可独立运行,不一定需要串口或者网络。前提是有一个液晶屏和 usb 键盘。将标准输出显示到液晶屏可以通过以下方法。 1.进入 U-BOOT,保存原来的启动参数 set bootargs_b \$(bootargs) 2.添加标准输出 set bootargs \$(bootargs) console=tty0 3.保存重启 saveenv;reset 至此,可以看到 LCD 上的启动信息了。 最后,插上 usb 键盘,按一下回车键即可进入 shell。

7、视频服务器应用

spcaserv 视频服务器, 请先执行 insmod usbcam.ko 加载摄像头驱动,执行./spcaserv -d /dev/video0 -s 320x240 -w 192.168.1.11:7070

play 目录下的 play.bat 为视频客户端,用记事本打开修改里面的 ip 地址。如 spcaview -w 192.168.1.11:7070 -s 320x240

注: 192.168.1.11 为开发版的 ip 地址

thttpd 为 web 服务器, thttpd-php 是带 php 支持的版本,执行./thttpd-php -C thttpd.conf 启动。thttpd.conf 为 配置文件,里面的 dir=/指明 web 目录为根目录,可以修改为你喜欢的目录,如/tmp。index.cgi 为 shell 版本 的 测 试 页, 输 入 所 有 cgi 变 量。可以将 其 复 制 到 / 根 目 录 下 测 试,在 浏 览 器 里 输入 http://192.168.1.11/index.cgi 即可。

usbcam.rar 开发板采集和 pc 端显示的源代码

9、sqlite 为嵌入式的数据库, test.c 是一个在 c 语言下与 sqlite 沟通的简单例子。

10、tslib 为触摸屏的校正、测试工具。auto.sh 是一个自动配置的脚步,执行前请先修改里面 export TSLIB_ROOT=/opt/tslib 这一变量,改为 tslib 所在的目录。./auto.sh 执行完后可进入 tslib/bin 目录下, ts_calibrate 是一个校正工具(我们的系统上已自带校正工具,其校正数据仅供 te_test 使用), ts_test 是一个触摸屏测试工具。

ad.ko 和 ts.ko 是 ad 转换和触摸屏驱动,详细使用请留意后续说明,请暂时不要使用。

11、vntwusb.ko 是 USB 无线网卡的驱动:通过 insmod vntwusb.ko 加载,执行 ifconfig 应该会多一个无线网卡出来。

wetctl 是 USB 无线网卡的控制软件:执行./wetctl 就应该看到..(我们暂时调不通)

12、adc ad 检测的驱动来的,加载了后可以运行./ad 来测电压的

13.、SD卡更新、使用说明

将光盘中文件 new_mmc.patch 拷贝到 Linux-2.6.18.2 目录下,执行 patch -p1 < new_mmc.patch 即可。 执行 make clean && make uImage 重新编译内核。启动后如果没有插入 SD 卡会看到以下两行信息: s3c2410-sdi s3c2410-sdi: powered down. s3c2410-sdi s3c2410-sdi: initialisation done.

当 SD 卡插入后会提示: (我的是 1G 的 SD 卡) [root@avantech /]# mmcblk0: mmc0:0d50 SD01G 1006080KiB mmcblk0: p1

同时/dev 目录下会出现 mmcblk0p1 和 mmcblk0 。如下: [root@avantech /]# ls /dev console mmcblk0p1 ptmx shm mmcblk0 null pts tty0

mmcblk0p1 即为 SD 卡的第一个分区,也是唯一的一个分区,挂载该分区即可读写 SD 卡。执行 mount /dev/mmcblk0p1 /mnt 将 SD 卡挂载到/mnt 目录下。下面是读写的测试过程: [root@avantech /]# cd /mnt [root@avantech /mnt]# ls etc [root@avantech /mnt]# mount /dev/mmcblk0p1 /mnt/ [root@avantech /mnt]# cd /mnt/ [root@avantech /mnt]# ls happy.mp3 [root@avantech /mnt]# rm happy.mp3 [root@avantech /mnt]# ls [root@avantech /mnt]# cp /opt/happy.mp3 ./ [root@avantech /mnt]# ls happy.mp3 [root@avantech /mnt]# cd .. [root@avantech /]# umount /mnt/ [root@avantech /]# ls mnt/

etc

[root@avantech /]#

如果 Linux 启动时 SD 已插进去,/dev 目录下是不会出现 mmcblk0p1 和 mmcblk0,请执行 mdev –s 重新扫描硬件即可。

二十三、Linux 基本知识

23.1 Linux 常用命令

下面中 Linux 常用命令,也适合于嵌入式 Linux 系统。

23.1.1 文件列表 -ls

ls #以默认方式显示当前目录文件列表

ls-a #显示所有文件包括隐藏文件

ls-l #显示文件属性,包括大小,日期,符号连接,是否可读写是否可执行

23.1.2 目录切换-cd

cd dir #切换到当前目录下的 dir 目录

cd/ #切换到根目录

cd.. #切换到上一级目录

23.1.3 复制 -cp

cp source target	#将文件 source 复制为 target
------------------	------------------------

cp/root/source. #将/root下的文件 source 复制到当前目录

cp-av source_dir target_dir #将整个目录复制到,两目录完全一样

cp-fr source__dir target_dir #将整个目录复制,并且是以非法链接方式复制,当 source 目录带有符号 链接时,两个目录不相同

23.1.4 删除 -rmrm file #删除某一个文件rm-fr dir #删除当前目录下叫 dir 的整个目录

23.1.5 移动 -mv

mv source target #将文件 source 更名为 target

23.1.6 比较 -- diff

diff dir l dir2 #比较目录 1 与目录 2 的文件列表是否相同,但不比较文件的实际内容,不同则列出 diff dir l dir2 #比较文件 1 与文件 2 的内容是否相同,如果是文本格式的文件,则将不相同的内容 显示,如果是二进制代码则只表示两文件是不同的

23.1.7 回显 -echo

echo message #显示一串字符

echo"message message2" #显示不连续的字符串

23.1.8 容量查看 -du

du #计算当前目录的容量

du-sm/root #计算/root 目录的容量并以 M 为单位

23.1.9 文件内容查看 --cat

cat file #显示文件的内容,和 DOS 的 type 相同

cat file | more #显示文件的内容并传输到 more 程序实现分页显示,使用命令 less file 可实现相同的功能

23.1.10 分页查看 -more

more #分页命令,一般通过管道将内容传给它,如ls | more

23.1.11 时间日期 -date

date #显示当前日期时间

date -s 20:30:30 #设置系统时间为 20: 30: 30

date -s 2002-3-5 #设置系统时期为 2002-3-5

23.1.12 查找 -find

find -name/path file # 在/path 目录下查找看是否有文件 file

23.1.13 搜索 -grep

grep --ir"chars" #在当前目录的所有文件查找字串 chars,并忽略大小写,-i为大小写,-r为下一级目录

23.1.14 设置环境变量 -export

export LC_ALL=zh_CN.GB2312 #将环境变量 LC_ALL 的值设为 zh_CN.GB2312

23.1.15 编辑 --vi

vi file #编辑文件 file

vi 原基本使用及命令:

输入命令的方式为先按 ctrl+c, 然后输入: x (退出),: x!(退出并保存): w (写入文件), :w!(不询

广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II

问方式写入文件),: r file(读文件 file), %s/oldchars/newchars/g(将所有字符串 oldchars 换成 newchars) 这一类的命令进行操作

23.1.16 压缩与解压 -tar tar xfzv file.tgz #将文件 file.tgz 解压 tar cfzv file.tgz source_path #将文件 source_path 压缩为 file.tgz

23.1.17 挂接 -mount

mount -t yaffs/dev/mtdblock/0/mnt#把/dev/mtdblcock/0 装载到/mnt 目录mount -t nfs 192.168.0.1:/ root/mnt#将 nfs 服务的共享目录 root 挂接到/mnt 目录

23.1.18 启动信息显示 -dmesg

dmesg #显示 kernle 启动及驱动装载信息

23.1.19 -chmod

chmod a+x file #将 file 文件设置为可执行,脚本类文件一定要这样设置一个,否则得用 bash file 才能执行

chmod 666 file #将文件 file 设置为可读写

23.1.20 创建节点 -mknod

Mknod/dev/ttyl c 4 1 # 创建字符设备 ttyl,主设备呈为 4,从设备呈为 1,即第一个 tty 终端

23.1.21 进程查看 -ps

ps #显示当前系统进程信息

ps-ef #显示系统所有进程信息

23.1.22 杀死进程 -- kill

kill -9500 #将进程编号为 500 的程序杀死

二十四、各种应用介绍(WINCE 与 LINUX)

注: 以下内容不作解释与技术支持,仅以提供方便自学研究。

24.1 WinCE 中如何向 image 中添加文件

这里分3种情况来说明如何添加

第一种,就是 image 已经在 CEPC 或者终端上跑起来了,那么这个时候要想添加文件可以通过 PB 或者 EVC 提供的 remote file viewer,这个比较简单,看着界面操作就行了。

第二种,就是对定制的 image 已经 build 过了,现在想往里面添加文件的话,可以在你对应平台的 release 文件夹里面直接添加文件,然后修改平台配置文件 project.bib,然后再 make image,也可以将文件添加到 image 中去,将 image 启动后,会出现在 windows 文件夹下面

比如你想将 test.txt 添加到 image 中,则首先需要将此文件拷贝到平台 release 目录下面

(平台 release 目录也就是环境变量_FLATRELEASEDIR 的值, _FLATRELEASEDIR 的 = %_WINCEROOT%\PBWorkspaces\%_TGTPROJ%\RelDir\%_TGTPLAT%\%_TGTCPUFAMILY%_Release, 也就是 realease 目录, (_TGTPLAT 为平台名, _TGTCPUFAMILY 为 CPU 名)。我新建的平台为 shellTest, 其值为 E:\WINCE500\PBWorkspaces\ShellTest\RelDir\MyCEPC_x86_Release)

然后按照以下修改 project.bib:

在其中添加类似这样一行

test.txt \$(_FLATRELEASEDIR)\test.txt NK S

这句话的意思是说将平台 release 文件夹下面的 test.txt 文件添加到 image 中,文件属性为系统文件,关于 bib 文件的格式,请查阅 WinCE 的帮助。

修改好 project.bib 后,保存,然后在 pb 的 bulid 菜单下选择 make image,成功后下载到终端或 CEPC,就可以看到添加的文件了。

下面这种方法适合在 PB5 中处理:

第三种,就是平台定制都没做好,或者做好了需要修改,那么如果你直接按照第二种来做,然后选择 build 菜单的 sysgen and build 的话,你会发现根本不会讲 test.txt 拷贝到 image 中,这个也是我的惨痛教训, 花了好些时间才知道原因。

为什么呢?从 build image 时的 output 窗口,我每次都可以看到 clean up 项目 release 目录的输出,看

来我直接把文件复制到 release 目录是不行的,因为在 sysgen and build 的过程中此文件夹会被清空,自然 我的 test.txt 也被清掉了。那该怎么办呢?

这里缺少一个步骤,那就是要在平台设置中,作一些配置,从而让 Release 目录在被清空以后能将目标文件从本地硬盘动态复制到 release 目录

修改配置以便拷贝文件到 Release 目录的主要步骤如下:

1 pb 中从 platform 菜单选 setting

2 在弹出对话框中 Configuration 一项确保正确,一般默认就是正确的。

3 Custom Build Actions 选项卡中的 Build step 下拉框,选择 Pre-Make Image (有四个选项,分别为 Pre-Sysgen, Post-Sysgen, Pre-Make Image, Post-Make Image, 意思如其名),然后 New,在弹出的 Custom

copy \ %_FLATRELEASEDIR% \

Build Action 对话框中输入类似以下的语句:

比如 test.txt 放在我电脑上的 e 盘根目录下,那么语句是这样的:

copy E:\test.txt %_FLATRELEASEDIR%\test.txt

加上这个步骤后,再按第二种方法就可以达到目的了。

24.2 建立 HIVE 注册表

To use the hive-based registry

- 1. Add the Hive-based Registry Catalog item to your OS design.
- 2. Verify the following registry settings in the Platform.reg file for your OS design.

[HKEY_LOCAL_MACHINE\init\BootVars]

"SystemHive"="<your system hive location>"

"Start DevMgr"=dword:<your value>

For more information about configuring the registry to support the hive-based registry, see Hive-based Registry Setup.

hive to load.

[HKEY_LOCAL_MACHINE\init\BootVars]

"DefaultUser"="<username>"

4. Verify that all registry entries necessary for starting drivers in the first boot phase are wrapped in the comments.

The following code example shows typical comments.

; HIVE BOOT SECTION

<your registry settings>

; END HIVE BOOT SECTION

These tags are commands that tell the ROM registry builder to add the entries to the boot hive.

5. Set the following flag bit on each driver that is loaded during the first boot phase.

[HKEY_LOCAL_MACHINE\Drivers\...]

"Flags"=dword:1000

The flags are a bitmask to OR with any existing settings. This flag tells the Device Manager to load your driver in the first boot phase with the boot registry, and not to load it a second time in the second boot phase with the system registry. It prevents the driver from being started twice.

6. Set the MountFlags registry value equal to DWORD:2.

Set this value on the Storage Manager profile of the file system driver for the medium that contains the registry. This indicates that the file system contains the following registry key.

"MountFlags"=dword:2

7. Decide whether a registry flushing mechanism is necessary.

Outstanding registry data will be flushed on a suspend or resume cycle and any time the system goes through a software shutdown. However, data may be lost if power is suddenly removed. If a software shutdown is not an option, you may need to create a thread that periodically flushes data. To ensure that data is not lost, call RegFlushKey. This will flush any unsaved changes in the hive to the persistent file. It will not damage anything to call RegFlushKey when no data has changed; in that case, the file will not be touched. RegFlushKey should be called on both the system hive HKEY_LOCAL_MACHINE and the user hive HKEY_CURRENT_USER.

注:完成第六步后,需选择编译选项的 Clean Before Building,然后再编译(Build and Sysgen)。

24.3 应用程序中对注册表的编辑

在 WinCE 中的应用程序如需对注册表编辑,可使用如下几个函数:

Programming element Description

RegCloseKey This function releases the handle of the specified key.

RegCreateKeyEx This function creates the specified key. If the key already exists in the registry, the function opens it.

RegDeleteKey This function deletes a named subkey from the specified registry key.

RegDeleteValue This function removes a named value from the specified registry key.

RegEnumKeyEx This function enumerates subkeys of the specified open registry key.

RegEnumValue This function enumerates the values for the specified open registry key.

RegFlushKey This function writes all the attributes of the specified open registry key into the registry.

RegNotifyChangeKeyValue This function notifies the caller about changes to the attributes or contents of a specified registry key.

RegOpenKeyEx This function opens the specified key.

RegQueryInfoKey This function retrieves information about a specified registry key.

RegQueryValueEx This function retrieves the type and data for a specified value name associated with an open registry key.

RegSetValueEx This function stores data in the value field of an open registry key.

我们重点使用 RegOpenKeyEx, RegSetValueEx, RegQueryValueEx 和 RegCloseKey 这几个函数,其功能如 上面所介绍。对一个注册表操作,需先用 RegOpenKeyEx 打开,再用 RegSetValueEx, RegQueryValueEx 查询或设置,最后用 RegCloseKey 关闭。

下面是一个对注册表设置的例子:

void RegSwitch(LPCWSTR ImKey, LPCWSTR ImIconKey,

```
WCHAR * Im240X320, WCHAR * Im320X240, WCHAR * ImIcon)
```

```
{
```

LONG lret;

HKEY hKEY;

if(GetSystemMetrics(SM_CYSCREEN) > 240)

{

```
lret = RegOpenKeyEx(RESKEY,ImKey,0,KEY_READ,&hKEY);
if(lret==ERROR_SUCCESS)
```

{

RegSetValueEx(hKEY,VALUENAME,0,REG_SZ,

(LPBYTE)Im240X320,

sizeof(WCHAR)*(wcslen(Im240X320)+1));

lret = RegOpenKeyEx(RESKEY,ImIconKey,0,KEY_READ,&hKEY);

```
if(lret==ERROR_SUCCESS)
```

{

RegSetValueEx(hKEY,VALUENAME,0,REG_SZ,

(LPBYTE)ImIcon,

sizeof(WCHAR)*(wcslen(ImIcon)+1));

}

```
}
```

RegCloseKey(hKEY);

```
}
```

else

```
{
   lret = RegOpenKeyEx(RESKEY,ImKey,0,KEY_READ,&hKEY);
   if(lret==ERROR_SUCCESS)
   {
       RegSetValueEx(hKEY,VALUENAME,0,REG_SZ,
               (LPBYTE)Im320X240,
               sizeof(WCHAR)*(wcslen(Im320X240)+1));
       lret = RegOpenKeyEx(RESKEY,ImIconKey,0,KEY_READ,&hKEY);
       if(lret==ERROR_SUCCESS)
       {
           RegSetValueEx(hKEY,VALUENAME,0,REG_SZ,
                   (LPBYTE)ImIcon,
                   sizeof(WCHAR)*(wcslen(ImIcon)+1));
       }
   }
   RegCloseKey(hKEY);
}
```

```
下面是一个对注册表查询的例子,该例子实现了查询注册表中的屏幕旋转键值,并使屏幕旋转:
```

#include ''stdafx.h''

#include "windows.h"

int WINAPI WinMain(HINSTANCE hInstance,

HINSTANCE hPrevInstance,

LPTSTR lpCmdLine,

int nCmdShow)

{

}

DEVMODE devMode;

```
LONG lret;
```

LONG regStatus;

HKEY hKEY;

INT keyValue;

LPCWSTR subKey = L''System\\GDI\\Rotation'';

LPCWSTR valueName = L''Angle'';

DWORD pRegType,pRegKey,pRegKeySize;

lret = RegOpenKeyEx(HKEY_LOCAL_MACHINE,subKey,0,KEY_READ,&hKEY);

if(lret==ERROR_SUCCESS)

{

regStatus

=

RegQueryValueEx(hKEY,valueName,0,&pRegType,(LPBYTE)&pRegKey,&pRegKeySize);

```
if(regStatus == ERROR_SUCCESS && pRegType == REG_DWORD)
```

{

memset(&devMode,0x00,sizeof(devMode)); devMode.dmSize = sizeof(devMode);

devMode.dmFields = DM_DISPLAYORIENTATION;

 $Change Display Settings Ex (NULL, \& dev Mode, NULL, CDS_TEST, NULL);$

```
keyValue = (INT)pRegKey;
```

switch(keyValue)

{

```
case 90:
```

devMode.dmDisplayOrientation = DMDO_90;

break;

case 180:

devMode.dmDisplayOrientation = DMDO_180;

break;

case 270:

devMode.dmDisplayOrientation = DMDO_270;

break;

default:

```
devMode.dmDisplayOrientation = DMDO_0;
```

break;

}

devMode.dmFields = DM_DISPLAYORIENTATION;

```
if (DISP_CHANGE_SUCCESSFUL == ChangeDisplaySettingsEx(NULL,
```

```
&devMode, NULL, CDS_RESET, NULL))
```

{

```
RETAILMSG(1, (L"ChangeDisplaySettingsEx changed rotation angle to %d", keyValue));
```

} else

{

RETAILMSG(1, (L"ChangeDisplaySettingsEx failed to change the rotation angle to %d", keyValue));

```
}
RegCloseKey(hKEY);
}
return 0;
```

```
}
```

函数中的参数具体意义可以参考 PB 自带的帮助。

24.4 添加输入法

```
1、在 PB 中选择一个输入法。例如: Catalog\\Core OS\\Display based devices\\International\\Locale Specific Support\\Chinese(Simplofied)\\Input Method Editor\\.... 右键点击其中一个,选择"add to platform" 这是 添加输入法。
```

如要手动添加输入法,则需将相应的输入法 DLL 文件打包进内核,即添加进 platform.bib 或 project.bib 文件中,如下例所示:

 并在注册表添加如下例键值:

$[HKEY_CLASSES_ROOT\CLSID\{7DD4CC5E-4EB7-4f86-AA8F-D74C5F8360DA\}]$

@="English"

;"KBMode"=dword:0

[HKEY_CLASSES_ROOT\CLSID\{7DD4CC5E-4EB7-4f86-AA8F-D74C5F8360DA}\InprocServer32] @=''\\windows\\englishim.dll''

[HKEY_CLASSES_ROOT\CLSID\{7DD4CC5E-4EB7-4f86-AA8F-D74C5F8360DA}\IsSIPInputMethod] @="1"

[HKEY_CLASSES_ROOT\CLSID\{7DD4CC5E-4EB7-4f86-AA8F-D74C5F8360DA}\DefaultIcon] @=''\\windows\\englishim.dll,0''

如需将输入法设置成默认输入,可以设置如下键值:

[HKEY_CURRENT_USER\ControlPanel\Sip]

"DefaultIm"="{7DD4CC5E-4EB7-4f86-AA8F-D74C5F8360DA}"

2、加入环境变量 SIPSELECT=1。这是添加输入法选择器,就是在界面上多了个输入法的图标,点它后 会让你选择输入法。即选择菜单的 Platform->Setting…,在弹出的对话框中的 Environment 页添加环境变量。

24.5 删除一个注册表健值

注册表文件(platform.reg 或 project.reg 等)中在要删除的键值前加-号。例如:

[-HKEY_CLASSES_ROOT\CLSID\{42429667-ae04-11d0-a4f8-00aa00a749b9}]

表示删除 HKEY_CLASSES_ROOT\CLSID\{42429667-ae04-11d0-a4f8-00aa00a749b9}这个键值。

24.6 改变桌面背景图片

设置如下键值:

[HKEY_CURRENT_USER\ControlPanel\Desktop]

"Wallpaper"="\\Windows\\Wall.bmp"

"Tile"=dword:0

其中\\Windows\\Wall.bmp为在桌面上显示的背景图片

24.7 WinCE 中支持多国语言切换

1. 选择菜单的 Platform->Setting…, 在弹出的对话框中的 Custom Build Actions 页的下拉框中选择 Pre-Make Image

2. 点击 New…加入语句 createmui XXXX,此处的 XXXX 表示要支持切换的语言,例如:

createmui 0409,0804,0404

此处 0409 为英语 (美国), 0804 为简体中文, 0404 为繁体中文。以","分隔。

3. 在 platform.bib 中添加如下语句:

#include \$(_FLATRELEASEDIR)\MultiUI.bib

4. 重新编译(Build and Sysgen Current BSP)。

24.8 实现开机后 WINCE 的任务栏为隐藏模式

在 PLATFORM.REG 中添加

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Shell\AutoHide]

"Default"=dword:1

24.9 应用程序在 WinCE 启动后自动运行

在注册表(platform.reg 或 project.reg)的[HKEY_LOCAL_MACHINE\init]键值添加要启动的程序,例如:

[HKEY_LOCAL_MACHINE\init]

"Launch80"="TouchCalibrate.exe"

"Launch90"="ScrRotationChecker.exe"

"Depend80"=hex:1e,00

键值中"Launch80"="TouchCalibrate.exe"的 TouchCalibrate.exe 为要启动的程序, "Depend80"=hex:1e,00 表明 要 Launch30(30=0x1E)的程序启动成功才会启动 TouchCalibrate.exe。

然后重新编译即可。

24.10 向 WinCE 内核添加文件

在 platform.bib 或 project.bib 里添加。 格式:模块名称 路径 内存块 类型

具体类型如下: S: 系统文件, H: 隐藏文件, U: 不压缩文件(默认是压缩的) 例如:

s3c2440a_lcd.dll \$(_FLATRELEASEDIR)\s3c2440a_lcd.dll NK SH TouchCalibrate.exe \$(_TARGETPLATROOT)\Files\TouchCalibrate.exe NK U

24.11 应用程序在 WinCE 启动后自动运行

在注册表(platform.reg 或 project.reg)的[HKEY_LOCAL_MACHINE\init]键值添加要启动的程序,例如:

[HKEY_LOCAL_MACHINE\init]

"Launch80"="TouchCalibrate.exe"

"Launch90"="ScrRotationChecker.exe"

"Depend80"=hex:1e,00

键值中"Launch80"="TouchCalibrate.exe"的 TouchCalibrate.exe 为要启动的程序, "Depend80"=hex:1e,00 表明 要 Launch30(30=0x1E)的程序启动成功才会启动 TouchCalibrate.exe。 然后重新编译即可。

24.12 WinCE 配置文件中的路径

1. \$(PBWORKSPACEROOT),指项目文件所在目录。例如 D:\WINCE500\PBWorkSpaces\GSLPDA

2. \$(_TARGETPLATROOT),指平台文件所在目录。例如 D:\WINCE500\PLATFORM\GSL2440A

3. \$(_FLATRELEASEDIR), 指编译后文件所在目录,也就是 nk.nb0 所在目录,例如
D:\WINCE500\PBWorkSpaces\POS35\RelDir\GSL2440A_ARMV4I_Release
4. \$(_WINCEROOT),指WinCE 安装目录。例如 D:\WINCE500

24.13 WinCE 字库与内核分离设置

 1. 设置项目的注册表 project.reg 的 [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\FontPath] 和
 [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\FontLink\SystemLink]键值,如: 广州朗成电子科技有限公司 http://www.avantech.com.cn AT2440EVB-II

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\FontPath]

"FontPath"="\\Boot Disk"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\FontLink\SystemLink]

"Tahoma''=''\\Boot Disk\\simsun.ttc,NSimSun''
"Courier New''=''\\Boot Disk\\simsun.ttc,NSimSun''
"Arial''=''\\Boot Disk\\simsun.ttc,NSimSun''
"Times New Roman''=''\\Boot Disk\\simsun.ttc,NSimSun.ttc,NSimSun''
"Symbol''=\\Boot Disk\\simsun.ttc,NSimSun

其中\Boot Disk 为字库文件所在目录。

2.将 D:\WINCE500\PUBLIC\COMMON\OAK\FILES 下的 simsun.ttc 复制到相应分区(即\Boot Disk),多个 字库,可使用;号隔开。例如:

"Tahoma"="\\Boot Disk \\Windows\\simsun.ttc,NSimSun; \\Boot Disk \\Windows\\mingliu.ttc,PMingLiU"

24.14 向 WinCE 的注册表添加其他注册表文件

在要添加其他注册表文件的注册表内,加上如下例语句:

#include ''\$(PBWORKSPACEROOT)\Files\hwime\ime.reg''

其中 ime.reg 为要添加的注册表文件, \$(PBWORKSPACEROOT)\Files\hwime\为该注册表文件所在的路径。

24.15 WinCE 内核 NK.nb0 大小设置

设置 config.bib 文件中 ROMSIZE 的大小,例如:

ROMSTART = 8000000

ROMWIDTH = 32

$\mathbf{ROMSIZE} = \mathbf{001800000}$

该处设置 nk.nb0 为 24M

24.16 WinCE 中如何编译生成 NK.nb0

要产生 NK.NBx 文件,必须在 config.bib (例如\$ (WINCEROOT)\PLATFORM\SMDK2440A\Files,以后的的 config.bib 如无特别说明,均指该文件)中设置 ROMSIZE 参数。如 ROMSIZE=02000000,设置 NK.NB0 的大小为 32M,若 Image 的大小超过 32M,则还会产生 NK.NB1, NK.NB2.....,设置例子如下(加在 config.bib 文件末端):

ROMSTART = 80000000 ROMWIDTH = 32 ROMSIZE = 001800000

24.17 WinCE 电源管理

Windows CE 是典型的使用电池供电的系统。这使得正确操作系统十分关键,应用程序大多数时间都不需要关注 Windows CE 设备的电源损耗,但是在某些时候,你可能要注意这些损耗。

当用户关闭了一个使用电池的 Windows CE 设备,电源系统不会关闭 PC 电源,事实上,只是系统被 挂起(译者注:这里就像有些 PocketPC 把关闭电源放在拔 SIM 卡的位置,拔出 SIM 卡才真正关闭电源。 但是,目前包括 Smartphone 在内,因为硬件设备,比如 CPU 无法进入低功耗,所以为了省电,需要做到 关闭应用处理器及大部分设备供电,然后需要唤醒时,再通过定时器或无线模块唤醒。所以不关闭电源的 情况不是绝对的。)当用户打开设备电源,设备不会像 PC 一样重新启动,而是被唤醒,返回到与系统挂起 前一样的状态。这样导致一个应用程序在唤醒后会像挂起前一样运行。事实上,应用程序根本不知道它被 挂起,除非它明确地请求当系统挂起时通知它。从应用程序的角度看,电源管理有三种方式,查询电源状态,改变电源状态,和防止电源状态改变。

查询电源状态

要查询系统当前的电源状态,你必须调用

DWORD GetSystemPowerStatusEx2 (PSYSTEM_POWER_STATUS_EX2 pSystemPowerStatusEx2, DWORD dwLen, BOOL fUpdate);

函数带了三个参数:一个指向 SYSTEM_POWER_STATUS_EX2 结构的指针,结构的长度,和一个布尔值, 表示告诉操作系统是否应该查询电池驱动来得到最后的信息或者直接返回电池缓存中的信息。系统大约每 5 秒查询一次电池状态,因此,如果第三个差数是 FALSE,得到的数据不会太旧。结构 SYSTEM_POWER_STATUS_EX2 被定义为 typedef struct _SYSTEM_POWER_STATUS_EX2 {

BYTE ACLineStatus;

BYTE BatteryFlag;

BYTE BatteryLifePercent;

BYTE Reserved1;

DWORD BatteryLifeTime;

DWORD BatteryFullLifeTime;

BYTE Reserved2;

BYTE BackupBatteryFlag;

BYTE BackupBatteryLifePercent;

BYTE Reserved3;

DWORD BackupBatteryLifeTime;

DWORD BackupBatteryFullLifeTime;

WORD BatteryVoltage;

DWORD BatteryCurrent;

DWORD BatteryAverageCurrent;

DWORD BatteryAverageInterval;

DWORD BatterymAHourConsumed;

DWORD BatteryTemperature;

DWORD BackupBatteryVoltage;

BYTE BatteryChemistry;

} SYSTEM_POWER_STATUS_EX2;

在我描述的这个巨大的结构之前,我必须告诫你,这个结构返回的数据精确程度和电池驱动一样。同 样的结构被传给电池驱动来查询它的状态。Windows CE 不验证电池驱动返回的数据。这个函数返回来的 数据依赖于电池驱动,因此不同的系统有不同的变化。举个例子,许多系统在使用 AC 电源时不报告精确 的电源级数;另一些系统则相反。应用程序使用 GetSystemPowerStatusEx2 来自动预防和检测系统是否可 能运行应用程序。

第一个区域, ACLineStatus, 包含一个标志, 表示系统是否连接到 AC 电源。如果值是 AC_LINE_OFFLINE, 表示系统没有使用 AC 电源; AC_LINE_ONLINE, 表示系统使用了 AC 电源; AC_LINE_BACKUP_POWER 和 AC_LINE_UNKNOWN, 表示备用电源和未知电源。BatteryFlag 区域, 提 广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II 164 供了一个总的标识, 表示当前系统的电池状态, 可以有以下值:

BATTERY_FLAG_HIGH

电池被充满或接近充满。

BATTERY_FLAG_LOW

电池还有一点剩余。

BATTERY_FLAG_CRITICAL

电池电量处在一个临界状态。

BATTERY_FLAG_CHARGING

电池当前正在充电。

BATTERY_FLAG_NO_BATTERY

系统无电池

BATTERY_FLAG_UNKNOWN

电池状态未知

BatteryLifePercent 区域包含估计的电池电量能够维持的百分比。数值可能是 0 到 100 之间的一个,或用 255 表示百分比未知。BatteryLifeTime 区域表示电池耗尽之前可以维持的秒数。如果该值不能估计,区域填入 BATTERY_LIFE_UNKNOWN。BatteryFullLifeTime 区域包含完全充满电池需要的时间。如果该值不能估计,填入 BATTERY_LIFE_UNKNOWN。注意,在许多系统中,这些值可能难以测量。大多数 OEM 厂商简单 地在每个区域内填入 BATTERY_LIFE_UNKNOWN。

接下来的第四个区域(不计算保留区域)重复了前面的表述,只不过是对系统备份电池来说。因为这些值 广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II 165 大多数难以测量,许多系统简单地返回"unknown"给这些区域。

剩下的区域描述了电池和备用电池的电力状态,因为许多系统缺少测量这些值的能力,这些区域也被简单地默认为"unknown"。最后一个区域,BatteryChemistry,包含一个标志,表示系统中电池的类型。当前已 定义的值包括

- BATTERY_CHEMISTRY_ALKALINE
- BATTERY_CHEMISTRY_NICD
- BATTERY_CHEMISTRY_NIMH
- BATTERY_CHEMISTRY_LION
- BATTERY_CHEMISTRY_LIPOLY
- BATTERY_CHEMISTRY_UNKNOWN

改变电源状态

应用程序能通过一系列的方式改变系统的电源状态。在基于 Windows CE.NET 系统的较新系统中,首选的方式是使用电源管理程序,在之后的章节将会讨论。可是无论如何,还有大量的基于早期 Windows CE 版本的系统以及 Windows CE.NET 不包含电源管理程序版本。对这些系统来说,下面的技术会很方便。

关闭电源

应用程序可以通过调用一个少有资料的 GwesPowerOffSystem 函数挂起系统。这个函数可以在大多数 版本 Windows CE 中使用,但是最近才被公开。事实上,大多数 SDK 没有包含这个函数的原型,你可能要 提供原型。这个函数定义为

void GwesPowerOffSystem(void);

GwesPowerOffSystem 的使用很简单:简单调用,系统就会挂起。

如果你想避免使用很少资料的函数,你可以通过简单地模拟用户按关闭按钮来关闭系统。你可以通过使用 keybd_event 函数很容易地允许你的应用程序挂起系统,如下:

keybd_event (VK_OFF, 0, KEYEVENTF_SILENT, 0); keybd_event (VK_OFF, 0, KEYEVENTF_SILENT | KEYEVENTF_KEYUP, 0);

这两个 keybd_event 调用模拟了按和释放电源按钮,电源按钮的虚拟键值是 VK_OFF。执行前面的两行代 码将挂起系统。因为虚拟键代码在执行时会由 GWES 表现,两个函数可能在系统挂起前有一些状态的表现 (译者注:屏幕上会有关闭对话框之类的图像,和真实按下按钮的画面一样)。如果你的程序无法在 keybd_event 函数之前停止工作,添加一个 Sleep 调用来使应用程序暂停一些毫秒来让 GWES 真实地挂起 系统。

关闭屏幕

如果系统有有色背光显示,主要的电源消耗不是 CPU 而是背光。在一些环境下,一个应用程序需要运行却不需要显示在屏幕上。一个例子是音乐播放器应用程序,当用户听音乐的时候,不关注屏幕。在这些情形下,有能力关闭背光将意味着提高电池寿命。

当然,当用户想看屏幕时,任何关闭背光应用程序的需要一个简单的用户友好的方式来重新打开屏幕。 同样,记得用户典型的想法是屏幕变黑时会认为被关闭了,因此要考虑这点。举个例子,一个用户可能在 系统已经运行时试图打开系统电源,并且这样做了,却很意外地发现,设备电源被关闭了。同样,当系统 在这种情况下关闭显示,它同时也关闭了触摸屏。这意味着你不能告诉用户敲击屏幕来打开。而是,你需 要使用一些其他的事件,比如设置时间,任务完成,或用户按了一个按钮。最后,这里讨论的方式对大多 数基于 Windows CE 3.0 或更新的版本比较有用,并且被 Windows CE .NET 4.0 中的电源管理程序所替代。 对于较新的系统,先看看是否电源管理程序可用,然后通过它来控制屏幕。如果失败了,ExtEscape 方式 也许能行。

在 Windows CE 中,显示的控制是通过 Ext-Escape 函数。这是一个显示和打印机驱动的后门。Windows CE 显示驱动支持许多设备转义代码(escape codes),这些被公布在 Platform Builder 中。对于我们的目的 来说,只有两个转义代码被用到: SETPOWERMANAGEMENT 来设置显示的电源状态和 QUERYESCSUPPORT 来查询是否 SETPOWERMANAGEMENT 被驱动支持。下面的例子打开或关闭系统 显示通过显示驱动,并且支持完全的转义代码:

 \parallel

// Defines and structures taken from pwingdi.h in the Platform Builder

//

#define QUERYESCSUPPORT	8	
#define SETPOWERMANAGEMENT		6147
#define GETPOWERMANAGEMENT		6148

typedef enum _VIDEO_POWER_STATE {

VideoPowerOn = 1,

VideoPowerStandBy,

VideoPowerSuspend,

VideoPowerOff

} VIDEO_POWER_STATE, *PVIDEO_POWER_STATE;

typedef struct _VIDEO_POWER_MANAGEMENT {

ULONG Length;

ULONG DPMSVersion;

ULONG PowerState;

} VIDEO_POWER_MANAGEMENT, *PVIDEO_POWER_MANAGEMENT;

//-----

// SetVideoPower - Turns on or off the display

//

int SetVideoPower (BOOL fOn) {

VIDEO_POWER_MANAGEMENT vpm;

int rc, fQueryEsc;

HDC hdc;

// Get the display dc.

hdc = GetDC (NULL);

// See if supported.

fQueryEsc = SETPOWERMANAGEMENT;

rc = ExtEscape (hdc, QUERYESCSUPPORT, sizeof (fQueryEsc),

(LPSTR)&fQueryEsc, 0, 0);

if (rc == 0) {

// No support, fail.

ReleaseDC (NULL, hdc);

return -1;

}

// Fill in the power management structure.

```
vpm.Length = sizeof (vpm);
vpm.DPMSVersion = 1;
if (fOn)
```

vpm.PowerState = VideoPowerOn;

else

vpm.PowerState = VideoPowerOff;

// Tell the driver to turn on or off the display.

rc = ExtEscape (hdc, SETPOWERMANAGEMENT, sizeof (vpm),

(LPSTR)&vpm, 0, 0);

// Always release what you get.

ReleaseDC (NULL, hdc);

return 0;

}

前面的代码通过调用 ExtEscape 和 QUERYESCSUPPORT 命令来查询是否支持转移代码。被查询的命令 首先交给输入缓冲,如果 SETPOWERMANAGEMENT 命令被支持,程序就填充 VIDEO_POWER_MANAGEMENT 结构并再次调用 ExtEscape 设置电源状态。

虽然这些转义代码允许应用程序打开或关闭显示,Windows CE 没有一个统一的方式来控制背光的亮度。每个系统都有它自己的 OEM 特有方式来控制背光亮度。如果将来有一种标准的背光亮度控制方式, 它将很可能放在 ExtEscape 函数中。

打开系统电源

当系统被挂起,应用程序将不再运行,因此当系统唤醒时,应用程序看起来没有被控制。然而,有一些方式来唤醒一个挂起的设备。首先,一个应用程序通过给定一个时间,并使用 11 章提到的消息 API (Notification API)做系统被唤醒的计划。在一般情况下,OEM 厂商会分配一些中断条件,以便管理系统电源打开,或唤醒。这种方式的一个例子是一个系统当防止了一个同步架 (synchronization cradle)时被唤醒。

防止系统关闭电源

就自动挂起。要防止自动挂起,一个应用程序可以周期性地调用一下函数:

void WINAPI SystemIdleTimerReset (void);

这个函数重设 Windows CE 用来监视用户输入的定时器。如果定时器到达预先的没有用户输入的间隔,系 统会自动挂起。因为挂起超时值可以被改变,一个应用程序需要知道超时值,这样就要多一点调用 SystemIdleTimerReset。系统维护三个超时值,这些都能够使用 SystemParametersInfo 来查询。传递给 SystemParametersInfo 的常量的不同表现,显示如下:

SPI_GETBATTERYIDLETIMEOUT

当系统运行在电池电源状态下,离用户最后输入的时间

SPI_GETEXTERNALIDLETIMEOUT

当系统运行在 AC 电源状态下,离用户最后输入的时间

SPI_GETWAKEUPIDLETIMEOUT

在系统再次挂起时离系统被自动唤醒的时间

要防止电源被自动挂起,你需要查询这三个值,并在最短时间内返回之前调用 SystemIdleTimerReset。如果 超时值被设置为 0,表示超时值被禁止。

电源管理程序

一个新的,独立的电源管理组件在 Windows CE .NET 4.0 中被引入了。这个电源管理程序替代了许多 GWES 以前完成的函数。电源管理程序定义了一系列的电源状态,如 D0,D1,D2,和 D3。这些看起来 神秘的名字被对应于一些友好的系统级别名称。

对嵌入式系统来说,OEM 厂商定义了系统的电源状态。例如,电源状态可能是打开(On),空闲(Idle) 和挂起(Suspend)。其他电源状态也被定义了,像 ScreenOff, InCradle,和 OnBattery。

从应用程序的观点看,新的电源管理程序提供了通知电源状态改变的能力以及通过一系列的函数统一 改变电源状态的能力。

系统的电源状态被定义在注册表中,SDK 定义了 PWRMGR_REG_KEY,以致你不得不知道注册表的 字符串,但是当常量没定义的时间,电源管理程序注册数据被保留在 HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Power。电源状态被定义作为子键,位于 Key State。

电源通知

电源管理程序一个十分受欢迎的特点是,可以在系统电源状态改变时通知应用程序。这可以让应用程序从手动检测电源状态中解脱出来。一个应用程序可以通过调用 RequestPowerNotifications 请求电源管理程序当电源状态改变的时候发送一个通知给应用程序。电源管理程序会通过一个由应用程序前面建立的消息队列发送通知。

RequestPowerNotifications 原型如下。

HANDLE RequestPowerNotifications (HANDLE hMsgQ, DWORD Flags);

第一个参数是一个应用程序在之前建立的消息队列的句柄。第二个参数是一系列参数,表示应用程序想接 收的通知。

PBT_TRANSITION

接受系统电源状态改变的通知。例如,当系统从 On 到 Suspend。

PBT_RESUME

当系统 resume 的时候接收通知。

PBT_POWERSTATUSCHANGE

PBT_POWERINFOCHANGE

当系统电池级数变化时接收通知。

POWER_NOTIFY_ALL

接收所有的通知。

RequestPowerNotifications 函数返回一个电源通知的句柄,失败返回 NULL。消息队列建立的时候必须使应用程序有读权限,因为应用程序将从消息队列中读取电源通知。

要接收通知,应用程序必须使用 WaitForSingleObject 来阻塞消息句柄。像第 10 章所讨论的,当通知被放 在队列中时,句柄将被 signaled。实际的通知将由结构 POWER_BROADCAST 表中被接收到。

typedef struct _POWER_BROADCAST {

DWORD Message; DWORD Flags; DWORD Length; WCHAR SystemPowerState[1]; } POWER_BROADCAST, *PPOWER_BROADCAST;

第一个要注意的是,这个结构长度是可变的。最后一个字段,SystemPowerState,是被定义为 WCHARs 类型,但是可以填上非字符串数据。第一个字段是通知自己的标识,这个字段可以填前面 PBT_标志列表之

一。Flags 区可以包括以下标志,依赖于被接收的通知:

POWER_STATE_ON

系统处于 on 状态。

POWER_STATE_OFF

系统处于 off 状态。

POWER_STATE_CRITICAL

系统进入了一个临界 off 状态。

POWER_STATE_BOOT

系统正在启动。

POWER_STATE_IDLE

系统进入 idle 状态。

POWER_STATE_SUSPEND

系统被挂起。

POWER_STATE_RESET

系统被复位。

最后两个字段是相互关联的。Length 字段是 SystemPowerState 字段数据的长度。SystemPowerState 中包含的数据依赖于被发送的通知。对于 PBT_TRANSITION 通知来说,SystemPowerState 字段包含一个新电源状态的标识字符串。这个字符串是以非 0 结尾的。为了结束字符串,使用 Length 字段来指出字符串的长度。注意,Length 字段是以字节为单位的,当字符是双字节的 Uncode 字符时,需要获得字符串字符的长度,就需要用 Length 字段去除 TCHAR 的 size。

对于 PBT_POWERINFOCHANGE 通知来说, SystemPowerState 字段包含一个 PPOWER_BROADCAST_POWER_INFO 结构:

typedef struct _POWER_BROADCAST_POWER_INFO {

- DWORD dwBatteryLifeTime;
- DWORD dwBatteryFullLifeTime;
- DWORD dwBackupBatteryLifeTime;
- DWORD dwBackupBatteryFullLifeTime;
- BYTE bACLineStatus;
- BYTE bBatteryFlag;
- BYTE bBatteryLifePercent;
- BYTE bBackupBatteryFlag;
- BYTE bBackupBatteryLifePercent;

} POWER_BROADCAST_POWER_INFO, *PPOWER_BROADCAST_POWER_INFO;

注意,这里有一些字段的名字和函数十分相似于前面讨论的 SYSTEM_POWER_STATUS_EX2 结构。

设置电源状态

电源管理程序提供的函数也允许应用程序来控制电源状态。有两个方式来控制电源。第一个方式是应 用程序给定一个电源设定。第二个方式是应用程序请求电源状态不要低于给定的级别。

一个应用程序通过调用函数 SetSystemPowerState 可以请求特定的电源状态。这个函数原型如下。

DWORD SetSystemPowerState (LPCWSTR psState, DWORD StateFlags, DWORD Options);

电源状态可以被请求通过指定前两个参数。如果第一个参数是非零值,它指向一个字符串标识被请求的状态。这个字符串必须和注册表中列出的电源状态之一相匹配。

如果 psState 为 NULL,第二个参数 StateFlags,定义了请求的电源状态。这个参数是从 POWER_STATE_ON 直到 POWER_STATE_RESET 状态其中之一,这些在前面提到的 POWER_BROADCAST 结构有描述。

比较特别的是 POWER_STATE_RESET 标志。这个标志请求系统重起,使用 SetSystemPowerState 的方法重起 比 通 过 直 接 使 用 IOCTL_HAL_REBOOT 命 令 来 调 用 KernelIoControl 的 方 法 更 好 。 调 用 SetSystemPowerState 会让系统在重起设备之前任何还在缓冲中的数据保存到文件系统。

调用 SetSystemPowerState 是一个直接改变电源状态的方法。更巧妙的方法是通过调用 SetPowerRequirement 来请求系统维持应用程序所需最低限度的电源状态。SetSystemPowerState 是假定应用程序知道所需状态, 而调用 SetPowerRequirement 是允许系统对电源设定做优化以满足应用程序的需要。一个使用 SetPowerRequirement 会比较方便的例子是,一个使用串口的应用程序需要串口在进行通信时保持住电源状态。SetPowerRequirement 被定义如下。

HANDLE SetPowerRequirement (PVOID pvDevice,

CEDEVICE_POWER_STATE DeviceState, ULONG DeviceFlags, PVOID pvSystemState, ULONG StateFlags);

第一个参数指定了应用程序需要维护电源状态的设备。DeviceState 参数定义了设备的电源状态。 CEDEVICE_POWER_STATE 指定了状态范围是从 D0(意味着设备是处于最大功耗状态)到 D4(表示设 备被关闭)(译者注:其实 D0 到 D4 的状态的具体表现,完全是由 OEM 厂商可自定义的,对应用程序开 广州朗成电子科技有限公司 http://www.avantech.com.cn AT2440EVB-II 174 发者来说,比如是在 D1 关 LCD 背光还是在 D2,都是不确定的,微软只给出标准定义,而不是实际定义)。 DeviceFlags 参数由两个标志合并而成: POWER_NAME,表示设备名有效; POWER_FORCE,表示设备 应当维持当前状态甚至当系统挂起时。如果 pvSystemState 不为 NULL,它表示只有对于在 pvSystemState 中已命名的电源请求才是有效的。设备可能无法更改请求的状态。

应用程序应当注销通过调用 ReleasePowerRequirement 来注销请求, 原型如下。

DWORD ReleasePowerRequirement (HANDLE hPowerReq);

这里唯一的参数是从 SetPowerRequirement 里返回的句柄。

24.18 TCP/MP 源代码编译

TCP/MP 是一个开源的播放器,现在好像商业化了,改名叫 core player.据说目前支持最多媒体格式的播放器. 除了 rm,rmvb 等少数几种格式不支持外.可以在这个网址获得开源代码 http://picard.exceed.hu/tcpmp/. 我昨天下载了一个 0.72RC1 的源代码.期望能自己编译定制一个播放器.好东西搞起来果然很费劲!花费一个晚上终于在今天中午编译成功,在我的 arm9+wince 上跑起来了.我是用 evc4.2 编译的,网络上搜索,很多人都是碰到许多编译问题和显示问题.但都没有提供一个完整的解决办法.

1.从 http://picard.exceed.hu/tcpmp/下载源代码.我下载的始 0.72RC1.

2.编译环境.我安装的是 evc4.2+standard sdk+Win32(WCE ARMV4) Release.更低版本的 evc 应该也是可以. 如果是编译 x86 的.要下载一个 nasm 汇编工具.这点在 readme.txt 里面提到.nasm 的下载地址 http://nasm.sourceforge.net/.否则的话会报错.网上不少人碰到这个错误没法继续了,不先看 readme.txt 实在不 应该.

 3.从下面 2 个网址下载 AMR 的解码器的源代码.并且分别拷贝到 AMR 目录下的 26104 和 26204 中.同样,

 这
 个
 信
 息
 在
 readme.txt
 中
 提
 到

 http://www.3gpp.org/ftp/Specs/archive/26_series/26.104/26104-610.zip

4.将 player_ce3 项目 set as active project.这个项目将编译一个 player_ce3.exe 就是最终的程序.但编译这个程序依赖许多的库.这些库就是其他的 project 编译后提供的.

5.如果这时候点 rebuild all 会出现错误.我在 evc 左边工作区窗口的 FileView 标签下一个个的 project 编译过 去.player_ce2, sample_ce3 这 2 个 project 先忽略过去.(因为我们选择了 player_ce3.如果搞定了一个其它的也 没有问题.)大部分可库文件都可以正确编译.除了 asap 和 flac 这两个项目会报错.剩下的事情主要是解决这 2 个项目.

6.编译 asap项目.因为反复改动,具体的过程我记忆不清了.我把成功设置的结果写下来.要编译 asap 这个项目, 需要修改的内容:右键点 asap files -->settings-->c/c++-->Category-->Preprocessor.在 Additional include directories:中增加项目路径,注意这个是相对路径".,asap,atari800/src".不然一堆头文件会找不到.在 Preprocessor definitions:中增加一个宏定义",ASAP".上面的描述都是加入引号内的内容不包括引号.当然这 2 步动作也可以在源代码中修改.如此设置完毕后,asap files project 就可以正确编译了.

7.编译 flac 项目.同 6,打开 flac 的 settings 到相同界面.在 Additional include directories:中增加路径 "flac/include,flac/src/libFLAC/include",不然一堆头文件找不到. 然后,在 Preprocessor definitions:"FLAC_NO_DLL".增加这个定义避免使用_declspec(dllexport)定义函数造成的一大堆 c2491 错误.如此设置后,flac project 应该可以正确编译.

8. 同 理 修 改 player_ce3 项 目 , 在 Additional include directories 中 增 加 路 径 "../asap/asap,../asap,../asap/atari800/src,flac/include,flac/src/libFLAC/include".然后在 Preprocessor difinitions:增 加 ASAP.最后,成功编译 player_ce3.exe

9.拷贝包含 player_ce3.exe 在内所有的生成文件到目标板上.可以运行!但是菜单没有显示正确.

10.将 lang 目录下面的多国语言支持文件拷贝到目标板同一个目录下面.根据需要,我只拷贝了 lang_std.txt lang_en.txt,lang_chs.txt,lang_ca.def 也可以了.

完成上面内容.TCP/MP 就可以正常在我机器上运行了.(我的目标板报了个错误,没有声音.应该是我音频驱动还没有调试的原因.我的板子是自己定做的,驱动正在陆续增加.)赶紧下载了个 wmv 来测试.速度不错!

24.19 WinCE 挂起和唤醒(suspend/wakeup)

不管任何方式的系统挂起,最终都会调用 OEMPowerOff()函数来实现.OEMPowerOff()函数由 OEM 来完成, 这个函数也许会位于 power.c 或者 off.c 的文件中.OEMPowerOff()是 OEM 来实现的,代码和流程也许不同, 但基本按照下面的方法来完成.

挂起的过程:

1.先进行平台相关的动作,比如清屏,设置 AD,usb 等.

2.保存芯片所有的寄存器值到一个静态数组(就是堆栈中)

3.设置 io,关闭 kitl 等

4.呼叫 OALCPUPowerOff()进行挂起.

OALCPUPowerOff()是一个位于 startup.s 中的汇编函数,它按照下面的流程实现挂起功能

5.保存通用寄存器 r4-r12, lr 到堆栈

6. 保存 wakeup 后的地址,MMU 寄存器,进入各模式将 sp 和 lr 寄存器保存到内存 RAM 的某一个位置,这个 位置是由 config.bib 指定保留的.为什么不象之前一样保存到堆栈呢?因为系统唤醒后跳转到 reset 开始执行, 这时候堆栈还没有初始化.这也是 poweroff 过程复杂的原因.

7.计算刚才保存的数据块的检验和并保存到 GSTATUS3 寄存器.(GSTATUS3 和 GSTATUS4 是状态寄存器, 挂起直到唤醒过程都会保存里面的值)

8.禁止中断.

9.清 cache

10.使能唤醒中断,能唤醒可以是外部中断 0,1,2,或者 RTC 中断

11.设置 sdram 进入自刷新模式,最终 cpu 进入 power off 状态

唤醒的过程:

cpu 的 power off 模式和其他睡眠模式不同,其他的睡眠模式唤醒后会从睡眠处继续运行,而 power off 模 式唤醒后是从 reset 处执行.reset 有 3 种可能情况,1.正常的上电冷启动,包括 reset 信号线有效造成的 reset.2. 看门狗失效造成的 reset.3.power off 之后被外部中断或者 rtc 中断唤醒的 reset.在 reset 之后可以根据 GSTATUS2 寄存器来判断是否从 power off 唤醒.还有一个问题,不论何种方式 reset,都是先执行 bootloader 的 代码,所以唤醒过程需要 bootloader 的参与配合.具体流程:

1.外部中断或者 rtc 中断唤醒 cpu 进入 bootloader

2.bootloader 中停止 sdram 的自刷新模式,然后跳到内核开始地址.有些 bootloader 会做的更多,因为前面我们 把数据都保存到了 ram 中的某处,事实上只要知道这个 ram 地址就可以取得数据进入唤醒过程.所以有些 bootloader 会直接去唤醒.我认为这并不好,增加了 bootloader 的依赖性,层次间的耦合性也高了.

3.检查 checksum,因为之前设置 sdram 处于自刷新状态,在 poweroff 期间 sdram 里面的数据会保持,增加 广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II 177 checksum 是有必要的安全措施.

4.从 RAM 取得之前保存的参数,其中包含了唤醒后应该跳转的地址,和 MMU 的配置数据以及各个模式的 sp 和 lr.

5.启动 mmu

6.跳到唤醒后的新地址.

7.进入各个模式恢复 sp 和 lr.

8.恢复 r4-r12,lr

9.跳转到 lr,即相当于 OALCPUPowerOff()返回,返回到 OEMPowerOff()中.

10.打开 kitl,恢复所有寄存器,恢复平台之前状态.

唤醒过程实际是一个挂起的逆过程.如此,系统成功唤醒,所有运行的应用程序不知道自己被系统挂起过而继续运行.

24.20 WinCE 应用程序开发(基于 EVC)

1、得到当前进程的 HINSTANCE

HMODULE hModule = ::GetModuleHandle(NULL);

HINSTANCE hInst = hModule;

2、得到当前进程的绝对路径

GetModuleFileName()函数,该函数的第一个参数如果为 NULL,则返回最后一个发出 WM_CREAT 消息的文件的绝对路径。

3、手工创建状态栏上应用程序的图标

CreateStatusIcon 函数就是调用 CreateIconIndirect 用来手工创建 16x16 图标

4、用点对点消息队列进行进程间通信

CreateMsgQueue();

OpenMsgQueue();

GetMsgQueueInfo();

ReadMsgQueue();

WriteMsgQueue();

CloseMsgQueue();

RequestDeviceNotifications();

5.隐藏 taskbar 的几种方法:

5.1 全屏时隐藏,退出全屏时显示,在这种情况下可以先 Findwindow ("HHTASKBAR"),然后用 movewindow 移出全屏范围之外,需要时再移入

5.2 一直隐藏(就是从系统起来之后就不希望看到 taskbar),可以通过 Findwindow,和 showwindow 来实现,也可以通过修改 taskbar.cpp (public/shell/oak/hpc/explorer/taskbar/)中的 CTaskBar::Register 函数,让其直接返回 true

5.3 动 态 显 示 和 隐 藏 , 可 以 修 改 platform.reg 文 件 HKEY_LOCAL_MACHINE\Software\Microsoft\Shell\AutoHide("default"=dword:1) 和 HKEY_LOCAL_MACHINE\Software\Microsoft\Shell\OnTop("default"=dword:0) 以 及 HKEY_LOCAL_MACHINE\Software\Microsoft\Shell\SlideTaskBar("default"=dword:0)

6.对软件盘操作的函数:

SipGetInfo(&si)

SipSetInfo(&si)

SipShowIM(SIPF_ON)

7.增加或删除"开始"启动程序:

在 stmenu.cpp (public/shell/oak/hpc/explorer/taskbar)中对 stmenuData 这个结构体数组增加或删除成员

8.流驱动注册表中的 index:

大家都知道驱动设备的前缀 prefix 要用 3 个大写字母表示,而 index 则是驱动设备的索引。当在应用程序 中用 createfile 打开设备的时候,该函数的第一个参数就是 prefix 和 index 的组合

9.EVC 中打开网页程序

SHELLEXECUTEINI	SO ShExecInfo = $\{0\}$;		
ShExecInfo.cbSize =	= sizeof(SHELLEXECUTEINFO);		
ShExecInfo.fMask =	= SEE_MASK_NOCLOSEPROCESS;		
ShExecInfo.hwnd =	NULL;		
ShExecInfo.lpVerb	= NULL;		
ShExecInfo.lpFile =	TEXT("http://www.163.com");		
ShExecInfo.lpParameters =NULL;			
广步	h朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II		

ShExecInfo.lpDirectory = NULL; ShExecInfo.nShow = SW_SHOWNORMAL; ShellExecuteEx(&ShExecInfo);

10.取得设备信息

取得设备 OEM 信息:

TCHAR wszMachineName[128];

SystemParametersInfo(SPI_GETOEMINFO, sizeof(wszMachineName),&wszMachineName, 0);

取得平台信息:

TCHAR wszVersion[256];

SystemParametersInfo(SPI_GETPLATFORMTYPE, sizeof(wszVersion),&wszVersion, 0);

11.wince 屏幕的旋转

从 Windows CE 4.0 开始, DEVMODE 结构多了一个属性 dmDisplayOrientation, 你可以通过这个属性来 取得或设置屏幕旋转模式, 它对应的值如下。

DMDO_0 不旋转

DMDO_90 旋转 90 度

DMDO_180 旋转 180 度

DMDO_270 旋转 270 度

切换屏幕旋转模式通过调用 ChangeDisplaySettingsEx 来实现,如:

DEVMODE devmode = {0};

devmode.dmSize = sizeof(DEVMODE);

devmode.dmDisplayOrientation = DMDO_90; //水平模式

devmode.dmFields = DM_DISPLAYORIENTATION;

ChangeDisplaySettingsEx(NULL, &devmode, NULL, 0, NULL);

取得当前模式的代码如下:

DEVMODE devmode = {0};

devmode.dmSize = sizeof(DEVMODE);

devmode.dmFields = DM_DISPLAYORIENTATION;

ChangeDisplaySettingsEx(NULL, &devmode, 0, CDS_TEST, NULL);

通过 devmode.dmDisplayOrientation 的值可以判断当前模式。

值得注意的是,这样切换模式并不是永久的,当设备重新起动后,就会还原为初始状态。

如果你需要在保留切换的屏幕模式,可以通过设置注册表中 广州朗成电子科技有限公司 <u>http://www.avantech.com.cn</u> AT2440EVB-II 180
HKEY_LOCAL_MACHINE\System\GDI\Rotation 下的两个键值实现

Angle - DWORD (0, 90, 180, 270)

LandscapeMode - DWORD (0,1)

12.拷屏并保存为 bmp 图片

void SaveDCToBMP(HDC hDC,int nStartX,int nStartY,

int nWidth, int nHeight, LPCTSTR szFileName)

{

//定义图形色深

int iPixel = 32;

//位图信息头

LPBITMAPINFO lpbmih = new BITMAPINFO;

lpbmih->bmiHeader.biSize = sizeof(BITMAPINFOHEADER);

lpbmih->bmiHeader.biWidth = nWidth;

lpbmih->bmiHeader.biHeight = nHeight;

lpbmih->bmiHeader.biPlanes = 1;

lpbmih->bmiHeader.biBitCount = iPixel;

lpbmih->bmiHeader.biCompression = BI_RGB;

lpbmih->bmiHeader.biSizeImage = 0;

lpbmih->bmiHeader.biXPelsPerMeter = 0;

lpbmih->bmiHeader.biYPelsPerMeter = 0;

lpbmih->bmiHeader.biClrUsed = 0;

lpbmih->bmiHeader.biClrImportant = 0;

//创建位图数据

HDC hdcMem;

HBITMAP hBitMap = NULL;

CBitmap *pBitMap = NULL;

CDC *pMemDC = NULL;

BYTE *pBits;

hdcMem = CreateCompatibleDC(hDC);

hBitMap = CreateDIBSection(hdcMem,lpbmih,DIB_PAL_COLORS,(void **)&pBits,NULL,0);

pBitMap = new CBitmap;

pBitMap->Attach(hBitMap);

```
pMemDC = new CDC;
```

pMemDC->Attach(hdcMem);

pMemDC->SelectObject(pBitMap);

```
pMemDC->BitBlt(0,0,nWidth,nHeight,CDC::FromHandle(hDC),nStartX,nStartY,SRCCOPY);
BITMAPFILEHEADER bmfh;//位图文件头
ZeroMemory(&bmfh,sizeof(BITMAPFILEHEADER));
*((char *)&bmfh.bfType) = 'B';
*(((char *)&bmfh.bfType) + 1) = 'M';
bmfh.bfOffBits = sizeof(BITMAPFILEHEADER) + sizeof(BITMAPINFOHEADER);
bmfh.bfSize = bmfh.bfOffBits + (nWidth * nHeight) * iPixel / 8;
int iBMPBytes = nWidth * nHeight * iPixel / 8;
CFile file;
if(file.Open(szFileName,CFile::modeWrite | CFile::modeCreate))
{
file.Write(&bmfh,sizeof(BITMAPFILEHEADER));
file.Write(&(lpbmih->bmiHeader),sizeof(BITMAPINFOHEADER));
file.Write(pBits,iBMPBytes);
file.Close();
}
DeleteObject(hBitMap);
```

DeleteObject(hDC);
pMemDC->DeleteDC();
delete pMemDC; pMemDC = NULL;
delete pBitMap; pBitMap = NULL;
delete lpbmih; lpbmih = NULL;
}

```
13.识别当前系统语言
一般系统语言的表示是
0409 // Enaglish US
0404 // 繁体中文
```

0804 // 简体中文
可以用 GetSystemDefaultLangID()) 或 GetUserDefaultLangID()来获取
<pre>switch(GetSystemDefaultLangID()/*GetUserDefaultLangID()*/) {</pre>
case 0x0804:
break;
case 0x0404:
break;
case 0x0409:
break;

}

以下为 LINUX 基础的部分学习知识

24.21 QT 开发--Hello

第一步:

建立 hello.pro 文件,该文件的作用是为建立 Makefile 文件,如下例所示:

TEMPLATE = app

CONFIG += qt staticlib thread warn_on release

HEADERS += hello.h

SOURCES += hello.cpp $\$

main.cpp

TARGET = hello

其中 SOURCES += hello.cpp 识别源码中的.cpp 源文件。可以添加多个.cpp 文件

如有多个头文件,可以如下例所示添加:

HEADERS $+= += hello.h \setminus$

main.h

```
第二步:
使用 qmake 工具生成 Makefile。如下例所示:
qmake -o Makefile hello.pro
此时可看到在 hello.pro 所在目录生成了一个 Makefile 文件。
```

第三步:

有了 Makefile 后,现在开始编程了。建立 hello.cpp 文件,其内容如下: #include <qlayout.h> #include <qpushbutton.h> #include "hello.h" Hello::Hello(QWidget * parent,const char * name) : QDialog(parent,name) {

setCaption(tr("Hello"));

resize(240,300);

```
}
```

这是一个继承了 QDialog 的 Hello 类。

再建立 hello.h 头文件,其内容如下:

#ifndef HELLO_H

#define HELLO_H

#include <qdialog.h>

class Hello : public QDialog

{

public:

Hello(QWidget * parent = 0,const char * name = 0);

};

#endif

最后建立 main.cpp 文件:

#include <qapplication.h>

#include <qfont.h>

#include "hello.h"

QFont font("",11,QFont::Bold);

int main(int argc,char ** argv)

{

```
QApplication app(argc,argv);
app.setFont(font,TRUE,0);
Hello * dlg = new Hello();
app.setMainWidget(dlg);
dlg->show();
return app.exec();
```

```
}
```

其作用是调用刚才我们写好的 Hello 类,并在屏幕显示。

第四步:

编译上面的几个文件,如无意外,可以得到一个 hello 文件。在终端上输入./hello 运行该程序,可以看到屏 幕上显示一个 240*320 大小的空对话框。

后记: 在本例中,我们并没有涉及到信号与插槽,也没有涉及到交叉编译的问题,所以非常简单。目的是为了让大家了解一下 QT 开发的简单流程。

24.22 QT 开发--信号与插槽

在"QT开发--Hello"的基础上示范一下怎样应用信号与插槽。

第一步**:**

在 "QT 开发--Hello" 的基础上对 hello.h 作如下修改:

#ifndef HELLO_H

#define HELLO_H

#include <qdialog.h>

class QPushButton;

class Hello : public QDialog

```
{
```

```
Q_OBJECT
```

public:

Hello(QWidget * parent = 0,const char * name = 0);

private slots:

```
void btnClicked();
```

private:

QPushButton * btn;

};

```
#endif
```

```
在 "QT 开发--Hello"的基础上对 hello.cpp 作如下修改:
```

#include <qlayout.h>

#include <qpushbutton.h>

#include "hello.h"

Hello::Hello(QWidget * parent,const char * name)

: QDialog(parent,name)

{

```
setCaption(tr("Hello"));
```

resize(240,300);

QVBoxLayout * layout = new QVBoxLayout(this);

btn = new QPushButton(tr("Quit"),this);

layout->addWidget(btn);

connect(btn, SIGNAL(clicked()), this, SLOT(btnClicked()));

}

void Hello::btnClicked()

{

```
printf("You Clicked the Button!\n");
this->close();
```

```
}
```

第二步**:**

编译程序。如无意外,可生成 hello,运行 hello,可以在屏幕上看到现在的程序比之前的多了一个按钮, 点击按键,可以在终端上看到 You Clicked the Button!这行字。

现在我们来解释一下刚才的程序。在 hello.h 文件中,我们添加了一个按钮 QPushButton * btn,所以需要添加 class QPushButton 按钮类。另外,我们添加了一个插槽:

private slots:

是为了响应按钮的点击事件 clicked()),。连接信号与插槽需要用 connect 函数,即在 hello.cpp 中的 connect(btn,SIGNAL(clicked()),this,SLOT(btnClicked()))语句。使用信号与插槽,需要 QT 的 moc 工具对源 文件进行处理及在头文件中加入 Q_OBJECT,这个在先前的 Makefile 中已经包括了 moc 处理的命令。 信号与插槽可以理解成把一些事件的响应连接到相应的处理函数上。上例我们就是把按钮的 clicked()事件 连接到我们自己定义的 btnClicked()函数上进行处理。故在程序运行后,点击了按钮,就会处理 btnClicked() 中的程序。

信号与插槽的详细说明,建议大家到网上查看一下。

24.23 Linux 应用程序开发--串口通信

第一步:

参考"Linux应用程序开发--Hello"的第一步。

第二步:

示例程序如下:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <termios.h>

#include <errno.h>

#include <pthread.h>

int main(void)

{

```
int fd;
char rxBuffer[1024];
int len = 1024;
int rxByte;
struct termios opt;
fd = open("/dev/ttyS1",O_RDWR);
```

```
if(fd==-1)
{
    perror("error");
}
//set bps
tcgetattr(fd,&opt);
cfsetispeed(&opt,B9600);
cfsetospeed(&opt,B9600);
//set raw input and output
opt.c_lflag &= ~(ICANON|ECHO|ECHOE|ISIG);
opt.c_oflag &= ~OPOST;
tcsetattr(fd,TCSANOW,&opt);
//print tips
printf("Please enter any data to the COM1, Enter key to confirm\n");
write(fd,"READY!\n",7);
//receive bytes and send back
while(1)
{
    rxByte = 0;
    rxByte = read(fd,rxBuffer,len);
    if(rxByte>0)
     {
        rxBuffer[rxByte]='\0';
         write(fd,"RE:",3);
         write(fd,rxBuffer,rxByte);
         write(fd,"\n",1);
        printf("%s\n",rxBuffer);
     }
}
close(fd);
return 0;
```

}

第三步**:**

参考"Linux应用程序开发——Hello"的第三步。

第四步:

参考"Linux应用程序开发——Hello"的第四步。

第五步:

参考"Linux应用程序开发——Hello"的第五步。

24.24 Linux 应用程序开发——Hello

第一步:

在 Linux 平台(我用的是 Redhat9.2)上安装好 gcc 编译工具,例如编译 ARM 平台的应用程序可用 arm-linux-gcc,如用在 PC 平台下运行,可用 PC 自带的 gcc。

第二步**:**

编写 Hello 程序,可以使用 Redhat 自带的 gedit。

程序如下:

#include <stdio.h>

int main()

{

printf("Hello!");

}

并将程序保存为 hello.c。

第三步:

编写 Makefile,用一个文本编辑器(例如 gedit),新建一个文件。输入如下内容:

CC = gcc

EXEC = hello

OBJS = hello.o

CFLAGS +=

LDFLAGS +=

all: \$(EXEC)

\$(EXEC): \$(OBJS)

\$(CC) \$(LDFLAGS)-o \$@ \$(OBJS) \$(LDLIBS \$(LDLIBS_\$@))

clean:

-rm -f \$(EXEC) *.elf *.gdb *.o

并保存为 Makefile。注意 Makefile 跟 hello.c 放在同一目录下。

上面的例子中, CC = gcc 指所用的 gcc, 如果使用其他平台的 gcc, 可以修改此处, 例如 CC=arm-linux-gcc。

第四步:

编译。打开终端,进入 Makefile 跟 hello.c 所以目录。键入 make 并回车。如无意外,可以看到该目录下有一个 hello 的程序。

第五步:

运行程序。在相应平台的终端进入上一步所编译出来的 hello 所在目录,键入./hello,并回车。

修改记录表:

日期	修改内容	备注
2007-12-10	升级 ATBOOT 为 v4.02 版本,增加 USB 口下载 LINUX 内核的 方法,整合了使用文档。	
2007-12-26	增加 UBOOT v1.3 版本,提供新的 UBOOT 源代码,新的版本 是从 NAND FLASH 启动方式,取消旧的 UBOOT 使用方式。更 新使用文档的新 UBOOT 使用说明。	
2008-2-26	修改 USB 同步功能说明	
2008-2-26	添加各种应用的例子	