# 前 言

本文档为JXARM-2410系列教学实验系统的用户手册,用于指导JXARM9-2410的硬件安装和软件安装以及基本开发过程,在阅读本文档之前,请简单浏览本文的书写约定:

1、正文:

如下字体所示为普通正文:

JXARM9-2410-3教学系统主板上有几个跳线。具体的意义说明如下:

2、特别提示:

对于文中特别需要注意的地方,以下面的字体表示:

如果将JXARM9-2410-3直接与计算机连接时,请使用计算机直连网线,如果将JXARM9-2410-3接入局域网,请使用普通网线。

3、宿主机Redhat终端输入输出信息

如下所示,\$表示Redhat控制台,紧接其后的以黑体和斜体字表示的为在Redhat控制台 上输入的命令行,并以回车键结束。后面的为命令输入后的提示信息。

\$/etc/init.d/nfs restart		
关闭 NFS mountd:	[ 确定 ]	
关闭 NFS 守护进程:	[ 确定 ]	
Shutting down NFS quotas:	[ 确定 ]	
关闭 NFS 服务:	[ 确定 ]	
启动 NFS 服务:	[ 确定 ]	
Starting NFS quotas:	[ 确定 ]	
启动 NFS 守护进程:	[ 确定 ]	
启动 NFS mountd:	[ 确定 ]	

4、u-boot终端输入输出信息

如下所示,JX2410 #表示u-boot控制台,紧接其后的以黑体和斜体字表示的为在u-boot 控制台上输入的命令行,并以回车键结束。后面的为命令输入后的提示信息。

JX2410 # erase 1:4-7

```
Erase Flash Sectors 4-7 in Bank # 1
Erasing sector 4 ... done
Erasing sector 5 ... done
Erasing sector 6 ... done
Erasing sector 7 ... done
```

5、目标机Linux终端输入输出信息

如下所示, #表示目标机Linux控制台, 紧接其后的以黑体和斜体字表示的为在Linux控制台上输入的命令行, 并以回车键结束。后面的为命令输入后的提示信息。

#./helloworld Hello, World!

# 第一章 系统组成

### 1.1 硬件组成

JXARM9-2410-3教学系统采用统一的豪华实验箱包装,除ADT1000仿真器外,所有配件均 放置于实验箱之内,所包含配件如下:

- 1、JXARM9-2410-3主板一块(内置简易JTAG仿真器);
- 2、ADT1000仿真器一套(为可选配置);
- 3、320 X 240 256色彩色LCD显示器一块(带触摸屏);
- 4、4 X 4键盘一套;
- 5、USB连接电缆一条;
- 6、PC并口延长电缆一条;
- 7、RS-232串口通讯线一条;
- 8、直连网线一根;
- 9、软件光盘一张;
- 10、实验指导书一本;
- 11、GPRS天线一根;
- 12、GPS天线一根;

### 1.2 软件组成

JXARM9-2410-3软件均保存在随箱附带的软件光盘中,包含如下软件:

- 1、实验源程序;
- 2、实验程序模板;
- 3、 ADT IDE for ARM;
- 4、实验用工具软件及驱动程序。

# 1.3 光盘文件说明

目录	说明
\adt1000	ADT IDE for ARM
\document	本用户手册
\examples	实验源程序
\modules	实验程序模板

\tools	实验用工具软件以及驱动程序
\linux	Linux源代码、例程
\datasheet	硬件芯片数据手册
\flashupdate	Flash中数据的二进制代码

# 1.4 实验清单

Windows 下实验清单:

实验	说明
Leddemo	ARM开发环境实验
Asse1	ARM汇编语言编程实验
Asse2	C语言与汇编语言编程实验
uart	串口通讯实验
Interrupt	中断实验
Dma	DMA实验
Pwm	PWM实验
Rtc	实时时钟实验
Watchdog	看门狗实验
Rs485	RS485接口实验
lic	IIC总线实验
Da	D/A实验
Ad	A/D实验
Stepper	步进电机驱动实验实验
Keyboard	键盘输入与10实验
Lcd	LCD显示实验
Touch	触摸屏驱动实验
Iis	数字音频接口实验
Tftp	TFTP以太网通讯实验
can	CAN总线实验
Ps2	PS/2接口实验(键盘和鼠标)
cf	CF卡实验
ide	IDE硬盘接口驱动实验
parallel	并行打印机接口通讯实验
Ucos-ii	UCOS-II实验: Eg1, Eg2, Eg3, Eg4

gprs	GPRS相关实验(主叫实验,被叫实验,收短
	信实验和发短信实验)
gps	GPS实验
Linux 下实验清单:	
实验	说明
/home/cvtech/jx2410/linux	Linux内核移植实验
/home/cvtech/jx2410/linux	Linux文件系统实验(JFFS2/CRAMFS/RAMDISK)
/home/cvtech/jx2410/examples/helloworld	Linux基本应用程序编写实验
/home/cvtech/jx2410/examples/thread	Linux多线程应用程序设计实验
/home/cvtech/jx2410/examples/drv	Linux驱动程序编写实验
/home/cvtech/jx2410/examples/serial	Linux串口通信实验
/home/cvtech/jx2410/examples/interrupt	Linux中断处理实验
/home/cvtech/jx2410/examples/watchdog	Linux端口编程及看门狗实验
/home/cvtech/jx2410/examples/timer	Linux下的定时器编程实验
/home/cvtech/jx2410/linux	Linux下的以太网驱动实验
/home/cvtech/jx2410/examples/socket	Linux下的SOCKET通信实验
/home/cvtech/jx2410/linux	Linux下的USB接口实验
/home/cvtech/jx2410/examples/framebuffer	Linux下显示驱动及应用实验
/home/cvtech/jx2410/examples/touch	Linux下触摸屏驱动及应用实验

### 注:对于高校实验室建设方案,另外提供如下实验以及教案(PowerPoint文档):

实验	说明
/home/cvtech/jx2410/u-boot-1.1.1	u-boot移植实验
/home/cvtech/jx2410/examples/minigui	MiniGUI图形系统移植实验
/home/cvtech/jx2410/examples/minigui	MiniGUI图形应用程序编写实验

# 1.5 硬件检测

JXARM9-2410出厂时已经安装了四个操作系统: uC/OS-II、Linux、Windows CE、 vxWorks。在系统安装之前请运行它们以便检测硬件是否正常。下面请安装下面的提示一步 一步进行:

该节动作均在Windows环境下进行。

1、配置超级终端:

请参照"超级终端设置"一节进行配置超级终端,并连接串口线。

2、系统上电

然后将系统上电,正常将出现如下界面:

U-Boot 1.1.1 (Dec 14 2004 - 11:51:57) U-Boot code: 33F80000 -> 33F9F394 BSS: -> 33FD3A54

```
RAM Configuration:
Bank #0: 30000000 64 MB
Flash: 32 MB
In: serial
Out: serial
Err: serial
Hit any key to stop autoboot: 0
JX2410 #
```

如果不能出现上述界面,请检查您的超级终端设置,如果仍然没有出现请不要再继续往 下进行,请与我们联系。

Mail:support@cvtech.com.cn

Tel:027-87522625

3、运行Linux

在上述界面输入run bootlinux然后回车,或者按下实验箱小键盘的"7"键将运行固化 到flash中的linux操作系统,超级终端中显示如下所示:

JAZ410 # <b>run bootiinux</b>
## Starting application at 0x30008000
Uncompressing Linux done,
booting the kernel.
Linux version 2.4.18-rmk7-pxa1 (root@localhost.localdomain) (gcc version 2.95.3 20010315
(release)) #148 二 12 月 14 10:52:36 CST 2004
CPU: ARM/CIRRUS Arm920Tsid(wb) revision 0
Machine: Samsung-SMDK2410
Warning: bad configuration page, trying to continue
On node O totalpages: 16384
zone(0): 16384 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line: root=/dev/ram0 console=ttyS0

4、运行uC/OS-II demo程序

重新将系统上电,并在第3步的界面输入run bootdemo然后回车,或者按下实验箱小键 盘的 "8"键将运行固化到flash中的uC/OS-II操作系统,超级终端中显示如下所示:

JX2410 # *run bootdemo* ## Starting application at 0x30000000 ... uC/OS-II, The Real-Time Kernel ARM9 (S3C2410X) Ported version

(Ported by) CVTECH Ltd., www.cvtech.com.cn 1:数码管 2:蜂鸣器 0:跑马灯 3:放音测试 4:录音测试 5:IIC 测试 6:串口 0 测试 7:串口1测试 8:RS485 测试 9:AD 测试 10:DA 测试 11:键盘测试 12:实时时钟 13:USB FIFO 测试 14:步进电机驱动 15:SD/MMC 卡读写 16:CF卡读写 17: IDE 硬盘读写 18: 打印机驱? 19: GPS 测试 21:GPRS-短信 22:GPRS-连接 PC 20:GPRS-电话

Select the function to test :

此时在LCD显示器上将显示一个菜单,请使用小键盘的上下键进行选择并使用Enter键确 认以进行各项功能的测试。 5、运行vxWorks程序

重新将系统上电,并在第3步的界面输入bootm 800000然后回车,或者按下实验箱小键 盘的 "1"键将运行固化到flash中的vxWorks操作系统,超级终端中显示如下所示:

JX2410 # bootm 800000 ## Booting image at 00800000 ... Image Name: vxworks Created: 2005-08-30 12:17:45 UTC Image Type: ARM Linux Kernel Image (gzip compressed) Data Size: 816772 Bytes = 797.6 kB Load Address: 30010000 Entry Point: 30010000 Verifying Checksum ... OK Uncompressing Kernel Image ... OK Starting kernel ... OAttached TCP/IP interface to rtl unit 0 Attaching interface lo0...done Adding 3910 symbols for standalone. 11 (R) וווו ווווווווווווו ] ]]]]]]]]]]]] ]] 11 ווווווו וווווו ]]] 11111 1 111 1 1111 111 ]]]] ]] ]]]] ווווווווו 11 ווווו ]]] ]] ]]] 1111 1 11111 1 1111 11111 וווו וווווווו 11 1111 וווווו 1111 ]] ]]]] ]]]]]] ] ]]]]] ]]]] ]]]]]]]]] 111 1111 11 1111 1111 1111 1111 ן ווווו 111111 1 ווווווו 11111 ]]] ]]]]]]]]]] ] Development System VxWorks version 5.5.1 KERNEL: WIND version 2.6 Copyright Wind River Systems, Inc., 1984-2003 CPU: CVTECH JX2410-X ARM920T (ARM). Processor #0. Memory Size: 0x37f0000. BSP version 1.2/5. WDB Comm Type: WDB\_COMM\_END WDB: Ready. tffsDevCreate failed.  $\rightarrow$ 

在实验箱的LCD显示器上同时将显示一个图形化的demo程序。

6、运行Windows CE程序

重新将系统上电,并在第3步的界面输入run bootwince然后回车,或者按下实验箱小键 盘的 "9"键将运行固化到flash中的Windows CE操作系统,超级终端中显示如下所示:

JX2410 # run bootwince

## Starting application at 0x30200000 ...

```
Uncompressing
Linux....
  .....
   ..... done, booting the
kernel.
  Windows CE Kernel for ARM
                      Built on Mar 13 2003 at 22:54:19
  ProcessorType=0920 Revision=0
  sp abt=ffff5000 sp irq=ffff2800 sp undef=ffffc800 OEMAddressTable = 8c201280
  Windows CE Firmware Init
  INFO: Initializing system interrupts...
  INFO: Initializing system clock(s)...
  INFO: Initializing driver globals area...
  SDMMC config current rGPGCON: ff55fdba
  SDMMC config set rGPGCON: ff65fdba
  SDMMC config Init Done.
  OEMInit Done...
  Sp=ffffc7cc
  OEMSetRealTime: Year: 2003, Month: 9, Day: 1, Hour: 12, Minute: 0, second: 0
rcnr=1h
  OEMSetRealTime(register): Year: 3, Month: 9, Day: 1, Hour: 12, Minute: 0,
second: 0 rcnr=1h
  FMD::FMD Init
  Prescaler:2
  384 clock
  USB:OhcdPdd Init
  ++InitializeOHCI
  USB:*pIrg=11, *pioPortBase=0x2D0000
  OHCD: MapIrq2SysIntr(11): 27
  OHCD: Memory Object
  --InitializeOHCI
  SL Open 0x31760 (0 opens)
   ::: PBT InitializeAddresses - Success
     charlie::SDIO::SDHOST::SDCSDCardDllEntry::DLL_PROCESS_ATTACH
   ::: SYSINTR SDMMC
                 OEMInterruptDisable
   ::: SYSINTR DMA0
                OEMInterruptDisable
```

::: SYSINTR_SDMMC_CARD_DETECT	0EMInterruptEnable			
::: SYSINTR_SDMMC_SDIO_INTERRUPT	OEMInterruptEnable			
S3C2410DISP::InitializeHardware				
SL_Open 0x31760 (0 opens)				
+SL_GetStatus 0x31760				
在实验箱的LCD显示器上同时将显示标准Windows界面。				

# 第二章 硬件安装

### 2.1 JXARM9-2410-3 硬件安装步骤:

- 1、打开JXARM9-2410-3实验箱顶盖,将顶盖放倒或者直接取出;
- 2、安装下图所示进行硬件连线;



注:

如果将JXARM9-2410-3直接与计算机连接时,请使用计算机直连网线,如果将 JXARM9-2410-3接入局域网,请使用普通网线。 请不要带电拔插实验箱中的各种线以及模块。 请使用JXARM9-2410-3实验箱提供的电源,不要使用其他电源,对于使用其他电源

武汉创维特 All rights reserved 9

导致的实验系统损坏,本公司概不负责。

### 2.2 硬件跳线说明

JXARM9-2410-3教学系统主板上有几个跳线。具体的意义说明如下:

JP5:为调试模块选择跳线,当使用JXARM9-2410-3内置简易调试模块时,请将该跳线用 跳线帽短接起来,如果使用ADT1000高级JTAG仿真器时,请去掉该跳线帽。

LCD对比度调节旋钮:该旋钮调节LCD对比度,通过它调节LCD的对比度。

### 2.3 调试器安装

1、当使用JXARM9-2410-3內置简易调试模块时,请将计算机并口与实验箱左上角的并口 通过并口延长线实现连接,并将JP5跳线通过*跳线帽短接*。

2、当使用ADT1000高级仿真器时,请将计算机并口与仿真器实现连接,然后通过14Pin的JTAG对连线与JXARM9-2410-3的JTAG口实现连接,并将JP5跳线上的*跳线帽去掉*。连接如下 图所示:

注意:请不要同时连接简易调试模块和ADT1000仿真器。



# 第三章 JXARM9-2410 教学实验系统硬件资源

### 3.1 JXARM9-2410 教学实验系统硬件模块

JXARM9-2410 教学实验系统的硬件部分包括:基本模块、调试模块、通讯模块、人机交 互模块、A/D D/A 模块、工业控制模块、IDE/CF/SD/MMC 接口模块、GPRS 模块、GPS 模块。

### 1. 基本模块:

◇ SDRAM 存储器: 主板包含 64MB SDRAM。由两片 16 位数据宽度的 SDRAM 存储器组成,地址从 0x30000000<sup>~</sup>0x33fffffff。

◇ FLASH存储器: 主板包含 32MB NOR FLASH存储器和 8MB NAND FLASH, NOR FLASH 内部存放启动代码 Bootloader、Linux 内核映象、IIS 测试声音文件等。其数据宽度为 32 位,地址从: 0x00000000<sup>~</sup>0x01ffffff; NAND FLASH 中包含一个 cramfs 文件系统,在Linux 中使用。

◆ 串行通讯口: 主板包含 3 个 UART 接口: UARTO 和 UART1 用作 RS232 串行接口, UART2 用作 RS485 接口。UART0 在 Bootloader、演示程序、Linux 和多个实验中用于人机交 互(通过超级终端)以及文件传输;

◆ IIS 录放音接口: 主板有一个可以基于 DMA 操作的 IIS 总线接口,可进行立体声录 放音;

- ◆ I<sup>2</sup>C 总线接口: 与 24C08 芯片接口,可以存放一些固定的配置数据;
- ◆ 4个 LED 跑马灯:可独立软件编程;
- ◆ 6个七段数码管:6个共阳七段数码管;
- ◆ 外部中断测试: 一个按键用于外部中断0的测试;
- ◆ 复位按键:按键用于 CPU 复位;
- ♦ 两通道通用 DMA,两通道具有外部请求引脚的外设 DMA;
- ◇ 五个 PWM 定时器和一个内部定时器;
- ◆ 看门狗定时器;
- ◆ 8通道10-bit ADC。

#### 2. 调试模块:

- ◇ 标准 JTAG 接口: 20 针标准 JTAG 接口, 该接口用于高速仿真调试;
- ◆ 简易 JTAG 调试接口: 直连标准计算机并口,调试接口,该接口用于简易仿真调
  试。

#### 3. 通讯模块:

♦ 以太网通讯接口: 10M 以太网卡;

- ◆ USB 接口:两个 USB HOST 接口,可以挂接 U 盘、USB 鼠标、USB 摄像头等 USB 设备。遵循 USB1.1 标准;
- ◆ 标准计算机打印口(并口)。

### 4. 人机交互模块:

- ◆ 显示器/触摸屏: 5.7 英寸, STN 256 色 LCD 显示器, 320X240;
- ♦ 按键: 4X4 按键;
- ◆ PS/2 键盘和鼠标接口;
- ◆ USB 鼠标和键盘接口。

### 5. A/D D/A 模块:

◆ 10bit A/D 模块。

### 6. 工业控制模块:

- ◆ 两相步进电机驱动;
- ♦ RS485 总线接口;
- ◆ CAN 总线接口。

### 7. IDE/CF/SD/MMC 接口模块:

- ◆ 标准 IDE 硬盘接口;
- ◆ 标准 CF 卡接口;
- ♦ SD/MMC 卡接口。
- 8. GPRS 模块:
- ♦ GPRS 无线通讯模块。
- 9. GPS 模块:
- ◆ GPS 全球定位系统模块。

### 3.2 JXARM9-2410 教学实验系统硬件资源分配

1. 地址空间分配以及片选信号定义

表 3-1 JXARM9-2410 地址空间分配表

带杆区印	3份 明	数据宽度	读/写
地址区内	193	(位)	属性
	FLASH 存储器地址:	32	R/W
0x00000000~0x001ffffff	TE28F128: 16M 字节*2		
0x10000000	控制键盘扫描及跑马灯	8	W
0x10000002	读入键盘扫描值	8	R
0x10000004	数码管数据寄存器		W
0x10000006	数码管扫描控制寄存器	8	W
0x10080000~0x100807ff	CF 卡 MEMORY 模式属性寄存器	8/16	R/W
0x10080800~0x100808ff	CF 卡 MEMORY 模式公共寄存器	8/16	R/W
0x100c0000~0x100c07ff	CF 卡 I/0 模式	8/16	R/W
0x100c0000~0x100c00ff	IDE 读写地址空间		R/W

0x18000300~0x1800031f	网络接口芯片 RTL8019AS 读写	16	R/W
0x28000000	并口控制寄存器	8	W
0x28000000	并口状态寄存器	8	R
0x28000002	并口数据缓存器	8	R/W
0x28000004	控制寄存器 0	8	W
0x28000006	控制寄存器 1	8	W
0x28080000	控制寄存器 2	8	W
0x20000000	读密码	8	R
	SDRAM 存储器地址空间:	32	R/W
0x30000000~0x33FFFFFF	HY57V561620:32M 字节*2		

### 2. 外部中断分配

表	3-	-2	TXARM9-	-2410	外部中	中断分	配表
1	U.	-	Januas	<b>2110</b>	ZI HP I		HUN

中断口	说明
INTO	IDE 中断
INT1	扩充口中断
INT2	中断测试
INT3	中断测试
INT4	网口中断
INT5	CAN 中断
INT7	PS2 键盘中断
INT8	PS2 鼠标中断

### 3. A/D 端口分配

#### 表 3-3 JXARM9-2410 A/D 端口分配表

A/D 口	说明	备注
AINO	A/D 采集口测试 0	用于 A/D 测试
AIN1	A/D采集口测试1	用于 A/D 测试
AIN5	采集触摸屏的 Y 坐标	用于触摸屏
AIN7	采集触摸屏的X坐标	用于触摸屏

### 4. RAM 空间组织

存储空间的定制是嵌入式系统开发的一个特色,我们可以根据自己的需要自行规定内存的使用。在本教学实验系统中,SDRAM地址范围从 0x30000000 - 0x33ffffff 共 64Mbytes。 在不同的程序中,SDRAM 空间范围被分配成不同的区域用于不同的用途,下表是实验测试程 序所用的一个默认分配方式,它的代码空间和数据空间分配如下:

开始地址	结束地址	用途
0x30000000	0x33ff0000	程序区
0x33ff0000	0x33ff8000	栈
0x33ffff00	0x33ffffff	中断向量表

### 5. FLASH 空间组织

FLASH 地址范围从 0x00000000 - 0x01ffffff 共 32Mbytes。其空间组织如下表:

表 3-5 FLASH 空间分配表

开始地址	结束地址	用途
0x00000000	0x00040000	BOOTLOADER (u-boot) 代码
0x00040000	0x00080000	u-boot 环境变量
0x00080000	0x00100000	用户程序区
0x00100000	0x00200000	Linux 内核映象文件 zImage
0x00200000	0x00600000	Linux Ramdisk 映象文件 ramdisk.gz
0x00700000	0x10800000	用户程序/数据区
0x01080000	0x01800000	JFFS2/CRAMFS 文件系统区
0x01800000	0x01ffffff	用户程序/数据区

# 第四章 Windows 软件安装与设置

### 4.1 系统配置需求

Microsoft Windows98, Windows NT, Windows 2000, Windows XP。
 486以上CPU,建议采用Pentium II及更高级的处理器;
 64M以上内存,建议采用128M以上;
 200M空间的可用硬盘空间;
 CD-ROM驱动器;
 并行打印机端口。

# 4.2 ADT IDE 安装

注意:安装和使用ADT IDE请在管理员权限用户下进行。

- 1、关闭当前Win32环境下运行的所有应用程序;
- 2、将ADT IDE安装光盘放置在光驱中,运行光盘中adt1000\Setup.exe
- 3、在Welcome对话框中选择Next按钮;



4、在License对话框中,列出了有关ADT IDE系统发布和使用时,用户必须遵守的有关协议,选择Yes按钮,同意该协议,安装程序继续运行;

InstallShield Wizard			×
License Agreement Please read the following license agreement caref	ully.		
Press the PAGE DOWN key to see the rest of the a	greement.		
END-USER LICENSE AGREEMENT			<u> </u>
IMPORTANT-READ CAREFULLY: This End-User License Agreement ("EULA") is a li individual or a single entity) and Wuhan CVTECH i IDE, which includes computer software and associ include electronic documentation ("SOFTWARE P copying, or otherwise using the SOFTWARE PROI this EULA. If you do not agree to the terms of this EU Do you accent all the terms of the preceding Licens	egal agreement nfo & tech Co Ltech ated media and RODUCT' or "S DUCT, you agre ULA, do not use 1 se Anreement? 1	between you (eithe for the software pri printed materials, a OFTWARE''). By ir e to be bound by th the SOFTWARE Pl fyrou choose No. t	ran oduct ADT ind may istalling, reterms of RODUCT
close. To install ADT IDE, you must accept this ag	e Agreement		No
	N Dack	Tes	

5、在User信息对话框中输入有关用户信息,包含ADT1000仿真器配置的用户请在 "Serial Number"编辑框中输入*仿真器背面的序列号,没有包含ADT1000仿真器的用户请随* 便输入,选择Next按钮,继续安装;

ustomer Information	
Please enter your information.	
Please enter your name, the name of number.	t the company for whom you work and the product serial
<u>U</u> ser Name:	
UserName	
<u>C</u> ompany Name:	
Company Name	
Joompany Hamo	
Serial Number:	
Serial Number:	
Serial Number:	
Serial Number: 1111-1111-1111-1111 JIShield	

6、在Destination对话框中,用户需要选择ADT IDE系统的安装目录,请*选择 C:\ADT1000*,选择Next按钮继续安装,如果用户需要改变安装目录,选择Browse按钮,将出 现Choose Folder对话框,用户可以改变ADT IDE的安装位置,注意,改变的目录中不能包含 空格字符:

nstallShield Wizard	×
Choose Destination Location Select folder where Setup will install f	iles.
Setup will install ADT1000 in the follow	wing folder.
To install to this folder, click Next. To folder.	install to a different folder, click Browse and select another
Destination Folder	Browse
stallShield	< <u>B</u> ack Next> Cancel

7、 在Program Folder对话框中输入系统保存的系统名称,选择Next按钮继续安装;

Choose Folder	×
Please choose the installation for	lder.
<u>P</u> ath:	
c:\ADT1000	
Directories:	
● → win98 ● → WINDOWS ● → WINNT ● → PART_2 (D:) ● → PART_3 (E:) ● → PART_4 (F:) ● → 光盘 (G:) ● → 光盘 (K:) ● → 光盘 (K:) ● → ペー和店	
确定	

8、选择安装类型,用户可根据自己需要选择典型安装、小型安装以及定制安装;典型 安装包括应用程序、文档以及一些例程,小型安装只安装了应用程序和文档,另外用户可选 择定制安装,去掉一些安装项目;

InstallShield Wizar	d 🔀		
Setup Type Select the Setup	o Type to install.		
Click the type of Setup you prefer, then click Next.			
• Typical	Program will be installed with the most common options. Recommended for most users.		
© <u>C</u> ompact	Program will be installed with minimum required options.		
C C <u>u</u> stom	You may choose the options you want to install. Recommended for advanced users.		
InstallShield	< <u>B</u> ack <u>N</u> ext> Cancel		

9、在Information对话框中,列除了当前的有关安装信息和系统信息,用户如果确认,选择Next按钮,继续安装,安装程序进入拷贝文件阶段;

Start Conving Files	
Review settings before copying files.	
<ul> <li>Setup has enough information to start settings, click Back. If you are satisfie</li> </ul>	copying the program files. If you want to review or change any d with the settings, click Next to begin copying files.
Current Settinge:	
Current Settings:	
Semp Type	<u> </u>
Typical	
Destination Folder	
c:\ADT1000	
Liser Information	
	<b>▼</b>
•	
tallShield	
	(Basta Nasta Correct
	Allerel Nevel

10、文件拷贝结束后,系统提示用户重新启动计算机,用户需要在安装程序执行完后,重新启动计算机;

InstallShield Wizard	
	InstallShield Wizard Complete Setup has finished installing ADT1000 on your computer.
	< Back Finish Cancel

# 4.3 实验教程安装

将光盘中的如下目录拷贝到d:\cvtech\jx2410目录下。

光盘目录	硬盘目录	说明
\document	D:\cvtech\jx2410\document	本用户手册
\modules	D:\cvtech\jx2410\modules	所有实验程序模板
\tools	D:\cvtech\jx2410\tools	实验用工具软件以及驱动程序

\datasheet D:\cvtech\jx2410\datasheet 硬件芯片数据手册

在学生做实验时, examples示例供老师参考, 请不要拷贝

### 4.4 超级终端设置

1、运行Windows操作系统下的超级终端(HyperTerminal)应用程序,如下图所示,新建 一个终端通信,取名为JXARM9-2410。单击"确定"按钮。

	×
注接描述       ? ×         ●       新建连接         输入名称井为该连接选择图标:       各称(图):         [] jx44b0        國标(I):         國标(I):       ●         ●       ●       ●       ●         ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●         ●       ●       ●       ●       ●       ●         ●       ●       ●       ●       ●       ●         ●       ●       ●       ●       ●       ●         ●       ●       ●       ●       ●       ●         ● <th></th>	
	//,

2、选择终端连接串口,如串口2

🧞 jx44b0 - 超级终端	<u> </u>
文件(E) 編辑(E) 查看(V) 呼叫(C) 传送(E) 帮助(E)	
注接到 ? ×	
🧞 jx44b0	
输入待拨电话的详细信息:	
国家(地区)( <u>C</u> ):	
区号(E):	
电话号码(P):	
连接时使用(1):	
确定 取消	
斯开 自动检测 自动检测 SCROLL CAPS NUM 捕打印	

3、设置通信格式和协议

🍣 jx44b0 - 超级终端		
文件(E) 编辑(E) 查看(⊻) 呼叫(C)	COM2 属性	? ×
	端口设置	
	每秒位数(E): 115200	
	数据位(D):  8	
	奇偶校验(P): 无 ▼	
	停止位(S): 1	
	数据流控制(F): 无	
	还原为默认值(图)	
	确定 取消 应用	
断开 自动检测 自动检测	SCROLL CAPS NUM 捕 打印	

4、设置完成超级终端后,使用串口连接线连接设定的计算机串口与JXARM9-2410-3的串

口0,将JXARM9-2410-3上电,看超级终端中是否有打印信息,如果有,则表示配置正确。

# 第五章 ADT IDE 开发流程

本章将以 leddemo 程序为例,讲述在 ADT IDE 集成开发环境下,怎样编写、编译和调试程序。

### 5.1 检查硬件连接

检查实验箱配件是否齐全,包括主板,核心板和 LCD 等; 检查 JP5 设置,确保跳线帽设置正确,参照"硬件跳线说明";

### 5.2 连接调试器

如第一章所述,连接调试器,然后接通电源,包括 JXARM9-2410-3 的电源以及 ADT1000 的电源。

注意:

实验箱和 ADT1000 分别使用各自装置配套电源,不要混用。 不要同时连接 ADT1000 仿真器和简易调试模块,同时只能选择其中一种方式。

### 5.3 编辑、编译、调试

1、建立工程:

打开 ADT1000,点击"File->New 菜单",弹出 New 对话框,如下图所示,选择 Project 页,在 Project 页中选择调试设备,对于包含 ADT 1000 仿真器的用户,请选择 ARM9LPT,对于简易调试器的用户,请选择 ARM9SIMPLE,在工程名和位置中输入如下图所 示,可以修改该目录,但注意目录和工程名中不要包含空格。在工程类型中选择 EXEC。

New					×
File	Project	Workspace			
Targe	t devices		Pro	ject name	
SoftS ARM71	im PT		le	ddemo	
ARM91 ARM75 ARM95 ARM95 ARM95	PT imple imple asy		Loc D: Pro EX	ation \leddemo gect type EC Create new wo Add to currer	→ vrkspace at workspace
				Ok	Cancel

```
2、新建一个文件并保存为 d: \leddemo \leddemo. c,并编辑该文件,添加如下所示代
码:
/*文件名称: LEDSEG7.C
                                             */
/*实验现象: 数码管依次显示出 0、1, 2、……9、a、b、C、d、E、F
                                             */
/* LED DEMO CODE for MBA-44B0(S3C2410X)
                                             */
#define U8 unsigned char
unsigned char seg7table [16] =
{
  /* 0
             2
                            5
         1
                  3
                       4
                                6
                                     7*/
     0xf9.
          0xa4, 0xb0, 0x99,
  0xc0.
                          0x92.
                               0x82.
                                   0xf8.
  /* 8
        9
                                     F*/
             А
                  В
                       С
                            D
                                E
  0x80.
     0x90,
          0x88,
                0x83.
                     0xc6.
                          0xa1.
                               0x86.
                                   0x8e.
}:
void Delay(int time);
/* 函数说明: EB44B0 7 段构共阳数码管测试
                                             */
/* 功能描述: 依次在7 段数码管上显示 0123456789ABCDEF
                                              */
/* 返回代码:无
                                             */
/* 参数说明: 无
                                             */
void Test Seg7(void)
{
  int i:
  *((U8*) 0x1000006) = 0x00;
  for(;;)
  {
    /* 数码管从 0 到 F 依次将字符显示出来 */
    for(i=0:i<0x10:i++)
```

```
/* 杳表并输出数据 */
        *((U8*) 0x10000004) = seg7table[i];
        Delay (1000);
     }
     /* 数码管从 F 到 0 依次将字符显示出来 */
     for (i=0xf;i>=0x0;i--)
     {
        /* 查表并输出数据 */
        *((U8*) 0x10000004) = seg7table[i];
        Delay (1000);
     }
   }
}
/* Function name : 循环延时子程序
                                                     */
/* Description : 循环 'time' 次
                                                    */
/* Return type : void
                                                    */
        : 循环延时计数器
/* Argument
                                                     */
void Delay(int time)
{
  int i;
  int delayLoopCount=1000;
  for(;time>0;time--)
     for(i=0;i<delayLoopCount;i++);</pre>
}
  3、将 leddemo.c 文件加入到工程中,如下图所示:
```

🗳 leddemo - Arm Development Tools
Eile Edit View Project Build Debug Tools Window Help
🛛 🏠 🖾 🕼 🎒 🍐 🌾 🖻 💼 🗠 🗠 🖪 🎘 😽 🙀 Ram_demo
■ 巻 (巻 /* // #f    🕅 📕 🖬 🖬 🖬 🖬 🖬 🖬 🖬 🖬 🖬 🖬 🖬 🖬 🖬
ぬ 谷   ★ 冬 冬 ★    目 国 国   → 円 円 円   ① 物  ;
Workspace 'eddemo': 1 proj i i leddemo C hee Add Files to Folder MISC Docking View Hide Properties Alt+Enter Workspace
弹出文件选择对话框,选择 d:\leddemo\leddemo.c 文件,并点击打开
打开
查找范围( <u>I</u> ): 🔁 ledemo 🔽 🗢 🗈 📸 💷 🔻

查找范围( <u>I</u> ):	🔁 ledemo 💌	🗢 🔁	≝
c]leddemo.c			
' 文件名(N):	leddemo		打开(0)
_ 文件类型( <u>T</u> ):	Source Files (*.c,*.h,*.s)	-	THE WH
	,		AX.7H

? ×

```
4、新建一个文件并保存为 d:\leddemo\ldscript,并编辑该文件,按照如下进行编
辑:
SECTIONS
{
   = 0x30000000;
   .text : { *(.text) }
   .data : { *(.data) }
   .rodata : { *(.rodata) }
   .bss : { *(.bss) }
   EH FRAME BEGIN = .;
   \_EH_FRAME\_END\_ = .;
PROVIDE ( stack = .);
   .debug info
                0 : { *(.debug info) }
   .debug_line
                 0 : { *(.debug_line)
                                     }
   . debug abbrev 0 : { *(. debug abbrev) }
   . debug frame 0 : { *(. debug frame) }
}
   5、如下图所示,在工作区窗口中的 leddemo 工程名上右键点击并选择 Setting 菜单项
                                     - - -
                  🖃 🜃 Workspace "leddemo": 1 proj
                    🖻 🔝 leddomo
```



弹出工程设置对话框,选择Link页,在Link script中输入.\ldscript,然后点击OK

Project Settings	<u>د</u>	<
Setting for: DEBUG	General Debug Compile Assemble Link Directory	1
⊞-∰ leddemo	Category: Genernal Output file type © Execute file C Library Link script	
	Output .\debug\\$(TargetName) Link options	
	-o.\debug\\$(TargetName) \$(ObjFiles)	
	OK	-

6、如下图所示,在工作区窗口中的 leddemo 工程名上右键点击并选择 Build 菜单项。



编译成功结果如下图所示。

실 leddemo - Arm Development Tools	_ <mark>_ </mark>
Elle Edit View Project Build Debug Tools Window Help	
) 🖆 🚅 🖬 🕼 🕹 🕺 🛍 🛍 🕰 🕰 📴 🌠 🚰 🎇 RAM_DEMO 🔄 💟	
] 🗟 約 昭 /* // #f    🕅 📕 ddemo 🔽 DEBUG 🔽 🥙 🗃 🖉 👗	
●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●	
Workspace 'leddemo': 1 proj     G Source files     C head files     Assemble files     MISC files      Workspace	
≚ Configuration : leddemo - DEBUG Configuration : leddemo - DEBUG leddemo.c Link	<u></u>
leddemo.elf - build successful.	V
Build (Debug) Command Find in files 1 Find in files 2	Þ
Ready	NUM READ

7、连接仿真器到 JXARM9-2410-3 JTAG 连接。并将调试器和 JXARM9-2410-3 上电,然后 点击 Debug 菜单进行连接,如下图所示。

🗳 leddemo - Arm Development Tools						
∬ Eile Edit ⊻iew Project Build	De	bug	Tools	<u>W</u> indow	v <u>H</u> elp	
12 🚅 🖬 🗿 🕹 X 🖻 🖬		Rem	iote Coj	nect	F8	
	_	Dow	nload		Shift+F8	
A & 2 & 4 % % K   E	Ēļ.	<u>G</u> o			F5	55
	E.	Stop	)		Shift+F5	0.8-1
	Ef	<u>R</u> ese	et		Ctrl+R	
Superior State Sta	Ð	Re <u>s</u> t	art			
	{+}	Step	Into		F11	
i 🔛 leddemo.c	{} <sup>}</sup>	Step	<u>o</u> ver		F10	
- C head files	{}	Step	o <u>ou</u> t		Shift+F11	
	<b>?</b> {}	Run	To <u>C</u> ur	sor	Ctrl+F10	
	_	Mem	iory U <u>p</u>	Joad		
		Mem	iory Do	wnload		
		Тод	gle <u>B</u> rea	kpoint	F9	
		Enat	ole All Br	eakpoint	S	
		Disa	bļe Alī B	reakpoint	3	
		Dele	te All Br	eakpoints	3	
•		Breg	kpoints			
🚯 Workspace	-	Elash	n Progra	amer		
<pre>▲Information[0018] :</pre>	La	Set /	Active C	BR Confi	iguration	

正常连接结果如下图所示。

실 leddemo - Arm Development Tools [Run]	_ & ×
Eile Edit View Project Build Debug Tools Window Help	
昆 牧 将 /* // #    ♥   leddemo ▼ DEBUG ▼ 参 晋 晋 Ø Ă	
Ma A A   A % % % M    B    B    B    B    C    C (P (P (P ()    C (M    B    D    D    C (D () (D (	
	Reos
E R Workspace 'eddemo': 1 proj	E Current mode
	⊞User mode
Beddemo.c	⊕ Fiq mode
C head files	
Assemble ries	⊕ S∨c mode
	Abort mode
	⊞ Undet mode
tter workspace	
Information[0018] : target connected	<b>A</b>
	~
Build Debug Command Find in files 1 Find in files 2	
Ready	NUM READ

如果连接正确 Debug 菜单项将如下图所示



8、点击 Debug->Download 菜单下载程序到 SDRAM 中

🍕 leddemo - Arm Development	t To	ols [l	Run]				
Eile Edit View Project Build	De	bug	Tools	<u>W</u> indo	W	Help	
웥 🚅 🖬 🎒 🍜 🕺 🖻 🔒		Rem	ote Dis	co <u>n</u> nect	:	F8	
		Dow	nload			Shift+F8	Ī
M & 2 / A 2 2 K   E	ΞĻ	<u>G</u> o				F5	0.0
	E.	Stop	)			Shift+F5	
	ĒŤ	<u>R</u> ese	t			Ctrl+R	
eddemo	E	Re <u>s</u> t	art				
E C source files	<del>{</del> })	Step	Into			F11	
eddemo.c	<b>₽</b>	Step	<u>O</u> ver			F10	
C head files	{}•	Step	O <u>u</u> t		S	hift+F11	
	<b>?{}</b>	Run	To <u>C</u> ur	sor		Ctrl+F10	
		Mem	iory U <u>p</u> l	load			
		Mem	iory Do	<u>w</u> nload.			
	1	Тод	gle <u>B</u> rea	kpoint		F9	
		Enat	ole All Br	eakpoin	ts		
		Disat	ole All Br	reakpoir	nts		
		Dele	te All Br	eakpoint	ts		
•		Bre <u>a</u>	kpoints				
🚯 Workspace		Elash	n Progra	amer			
Information[0012]		Set /	Active C	BR Con	ifigu	ration	
Information[0018] :	τα	ger	. con	песте	u		

下载成功后,将显示入口点的源代码,如下图所示:



9、运行程序,点击 Debug->Go 菜单项,运行该程序,如果运行正常,JXARM9-2410-3 上的六个七段数码管将显示十六进制数据。



10、点击 Debug->Stop 停止程序运行。

# 5.4 运行 examples

在 JXARM9-2410 安装目录的 examples 目录下有本实验系统提供的所有实验源代码,如下图所示:



有两个工作区文件 examples. aws 和 uCOS-II. aws,在 ADT IDE 安装后可以通过鼠标双 击直接打开它们。如下图所示为打开 examples. aws 后的界面:

Bie Edit View Project Buld Debug Tock Window Help         Image: State in the	🔄 examples - Arm Developmen	nt Tools			_ 7 🗙
Image: Second	Eile Edit <u>V</u> iew Project Build D	ebug <u>T</u> ools <u>W</u> indow <u>H</u> elp			
Vorkspace' exceptes': 24 pro Vorkspace' exceptes': 24 pro Vorksp	<b>1 1 1 1 1 1 1 1 1 1</b>	🖞 🐀   💭 🖾 🛱 🖗 🖾 🗍 leddemo	T DEBUG	\$ # # <i>\$</i> 1	
	Vorkspace 'examples': 24 pro Vorkspace 'examples': 24 pro Vorksp				
	Link				▲ ■
leddemo.elf - build successful.	leddemo.elf - build succes	ssful. Find in files 1 \ Find in files 2 /			×
					: U .

在左边的窗口为工作区窗口,表示 examples. aws 工作区中包含多个工程,而每一个工作几乎都是对应于一个实验的源代码,如果要对其中的某个工程进行编译调试,先将该工程设置为活动工程,方法为在工程名的地方点击鼠标右键,并选择 "Set As Active Project"。所选择的工程名将变为黑体字,如下图所示:
🔄 examples - Arm Developmen	nt Tools				X
∬Eile Edit ⊻iew Project Build De	ebug <u>T</u> ools <u>W</u> indow <u>H</u> elp				
		asse1 💌	DEBUG	• 🕸 🎬 🎜 👗	
Vorkspace 'examples': 24 pro Passel	k				
<pre> 1eddemo.c Link </pre>					<b>^</b>
leddemo.elf - build succes	ssful.				
Build Debug Command	Find in files 1 $ angle$ Find in files 2 $/$				: 🖸 🛋

🔄 examples - Arm Development Tools	_ 7 🗙
Ele Edit View Broject Build Debug Tools Window Help	
E E E E → 7 7 6 4 ∞ 🖉 🛛 E 🖗 4 0 1 E Eddemo 🔽 DEBUG 🔽 🕸 🎬 🗸 👗	
Image: State of S	
× leddemo.c ↓ Link	<b>^</b>
leddemo.elf - build successful.	
Build (Debug Command ), Find in files 1 ), Find in files 2 /	: 🗊 🗂
然后和上节一样可以开始编译和调试,具体方法,请参考"编辑、编译、调试"进行	ŕ.

# 第六章 Linux 交叉开发流程

## 6.1 Linux 交叉开发模式

Linux 交叉开发采用宿主机和目标机的模式进行。宿主机是一台运行 Linux (如: RedHat)的 PC 机,目标机即 JXARM9-2410 教学实验系统。开发时使用宿主机上的交叉编 译、汇编及连接工具形成可执行的二进制代码,这种代码并不能在宿主机上执行,而只能在 目标机上执行。然后把可执行文件下载到目标机上运行。调试时的方法很多,可以使用串 口,以太网口等,具体使用哪种调试方法可以根据目标机处理器所提供的支持作出选择。宿 主机和目标板的处理器一般都不相同,宿主机为 Intel 处理器,而目标板如 JXARM9-2410 为 SAMSUNG S3C2410。

Linux 交叉开发包括 Linux 内核的开发和 Linux 应用程序的开发应用程序的开发。有两种方式:

1、Linux内核的开发:

下图所示为本实验系统所采用的实验环境以及开发流程。在主机的 RedHat Linux 操作 系统下安装 Linux 发行包以及交叉编译器 arm-linux-gcc。然后对 Linux 进行配置 (make menuconfig)并选择适合本实验系统的相关配置,配置完成后进行编译生成 Linux 映像文件 zImage。然后通过 u-boot 的 tftp 命令将该文件下载到目标板并执行。



2、Linux应用程序的开发:

Linux 应用程序的开发在宿主机一端的操作与内核开发类似,都使用 arm-linux-gcc 编译器应用程序,不同的是,编译生成 elf 格式的文件而不是 Linux 映像文件。编译生成的应用程序可以通过多种方式进行调试,下面介绍几种我们的系统中可以使用的方式:

1、Linux应用程序以文件的方式直接将 elf 格式的文件加入到 Linux 文件系统中,然 后更新整个文件系统,然后运行目标机 Linux 系统并运行应用程序。

2、运行目标机 Linux 系统,然后在目标机 Linux 系统中通过一些工具软件将宿主机中的应用程序通过以太网下载到目标机 Linux 系统,如 NFS、FTP、TFTP 等。

3、在宿主机端将 Linux 应用程序拷贝到 U 盘,然后运行目标机 Linux 系统,然后在目标机 Linux 系统,就后在 U 盘运行应用程序。

## 6.2 Linux 宿主机环境的建立与安装

在进行嵌入式开发前第一步的工作就是要安装一台装有指定操作系统的 PC 机作宿主开发机,对于嵌入式 Linux,宿主机上的操作系统一般使用 Redhat Linux。本系统推荐使用 Redhat 9.0。

## 6.2.1 安装 Linux

嵌入式开发通常要求宿主机配置有网络,支持NFS(为Linux应用程序开发时使用),支持TFTP服务器(为内核下载和烧写使用)。然后要在宿主机上建立交叉编译调试的开发环境。因此在宿主机上安装的Redhat 9.0必须具备以上条件。

安装Redhat 9.0的方法比较简单,只是在安装之前必须准备一定的空间给Linux。

1、建议首先在window下使用patrionmagic工具软件分配出大约4G的自由磁盘空间。过程如下:

🜲 PowerQuest PartitionMagic 8.	.0		- 日 ×
General View Disk Partition To	ools T <u>a</u> sks <u>H</u> elp		
🖆 🖆   🌭 🌋 🏀   🗟 🔏   🔤	<u>र</u> 🛃 🚺 💰 । 🗠 🕾 🔕		Ø
Pick a Task 🔕	🏶 Disk 1 - 78159 MB		
Create a new partition Create a backup partition	C: 10,001.4 MB D: WINXP	Browse	F: BAK G:
Install another operating system		Resize / Move	
A Resize a partition	Partition Turne	Copy	MR Status Drillion
Redistribute free space Morgo pertitions	Disk 1	Create	MB Status Fil/Log
Copy a partition		Delete	02.0 Arthur Drivery
	(*) FAT32	Undelete	0.0 None Primary
Partition Operations 🔕	WINXP (D:) FAT32	Label 9	04.4 None Logical
Browse partition	BAK (F:)	Eormat 2	99.4 None Logical
Delete partition	Local Disk (G:) 🛛 NTFS	Con <u>v</u> ert 20	62.6 None Logical
Presize/Move partition		Merge	
Convert partition		Split	
Split partition		Properties	
Properties		Check for Errors	
		Windows CheckDisk	
0 Operations Pending 🔗		windows Detragmenter	
		<u>A</u> dvanced	
🖄 Undo 🛛 🖉 Apply			
FAT FAT32 NTFS Lin	ux Ext2/3 📕 Linux Swap 🧧 Extended 📕 🛙	Unallocated 📕 Unformatted	Other Used Unused
Resize and/or move the selected parti	ition		0 operations pending
▓▋开始 🛛 🥭 🗹 🎾 🔄 🎽 🥔	ଞ   🔕 e:   🛥 e:   🧀 eb   🖏 d:   🗄	אנ 🕅 ad 🖓 xt 🕅 אנ	🥶 Pa 🚮 📢 🖓 🗔 CH 14:32

PowerQuest PartitionMagic 8.0		547 _ 🗗 🗙
General <u>V</u> iew <u>D</u> isk <u>Partition</u> <u>T</u> ools	T <u>a</u> sks <u>H</u> elp	
🖆 🖄 🖄 🖘   🔧 🔏 👌 🚮 🦉	Y   O &   ち & Ø	<u> </u>
Image: Solution of the second seco	Image: Second	F: BAK G: Ve Primary he Logical he Logical he Logical he Logical he Logical
FAT FAT32 NTFS Linux B	xt2/3 Linux Swap Extended Unallocated Unformatted	ther Used Unused
		0 operations pending
ﷺ日开始 🛛 🥭 🗹 🎾 🕒 🎽 🥑 搜	🔯 E:   🔄 E:   🥝 EB   💁 D:   🚍 30   🖄 Ad   😋 \\   📾 JX	😂 Pa 🚮 📢 🕼 🔽 🗔 CH 14:33

点击Ok, 然后点击"Apply the changes that are pending"按钮, 按照提示操作, 并重新启动计算机。

2. 将RedHat光盘插入光驱中,本例中以RedHat 9.0为例进行说明。

安装Redhat 9.0,安装时请选择完全安装,即选择Custom定制安装,在配置Firewall时请选择No Firewall,在选择软件Package时选择最后一项: everything,即完全安装。

## 6.2.2 配置以太网

前提必须是以太网卡已经配置正确,对于一般常见的RTL8139网卡,REDHAT 7.2及更高版本可以自动识别并自动装好,不要用户参与,因此建议使用该网卡。

然后配置宿主机IP,本用户手册下面的使用均假设宿主机的IP地址为192.168.1.180:

\$ifconfig eth0 192.168.1.180

为了验证网络是否配置成功,请按照如下提示进行:

首先连接网络,将宿主机使用普通网线接入局域网或者使用直连网线与另外一台计算机 进行连接。

在局域网内另外一台计算机上,保证其IP地址为192.168.1.xxx,其中xxx不能为180。 如果该计算机为Windows操作系统,在该计算机上输入如下命令:

- 🗆 🗙

٠

```
C: VIIBOUS/System32/cmd.exe
Microsoft Windows XP 1版本 5.1.26001
(C) 版权所有 1985-2001 Microsoft Corp.
C: \Documents and Settings\cvtech>ping 192.168.1.180
Pinging 192.168.1.180 with 32 bytes of data:
Reply from 192.168.1.180: bytes=32 time<1ms ITL=64</p>
Ping statistics for 192.168.1.180:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Documents and Settings\cvtech>_
```

如果为Linux操作系统,在终端中输入:

\$*ping 192.168.1.180* 如果ping通,表示网络连接已经建立。

# 6.2.3 配置 TFTP 服务器

检查宿主机端的TFTP服务是否已经开通,此步骤只在第一次时使用,以后可忽略。 开通宿主机上的TFTP服务,对于Redhat 9.0,在宿主机上执行:

```
$setup
选择System services:
```



将其中的tftp一项选中,并去掉ipchains和iptables两项服务。然后还要选择Firewall configuration,选中No firewall。

您想自	─────────────────────────────────────	务 £服务?	
	<ul> <li>[*] sshd</li> <li>[*] syslog</li> <li>[] talk</li> <li>[] telne</li> <li>[] tftp</li> <li>[] time</li> <li>[] time</li> <li>[] time</li> </ul>	g t udp ≇	
	确定	取消	



最后,退出setup,执行如下命令以启动TFTP服务:

```
$service xinetd restart
```

配置完成后,简单测试一下TFTP服务器是否可用,即自己tftp自己,例如在宿主机上执行:

```
$cp /home/cvtech/jx2410/images/zImage /tftpboot/
$tftp 192.168.1.180
tftp>get zImage
```

正确的结果如下图所示:



此时表示TFTP服务器配置成功了。若弹出信息说: Timed out,则表明未成功,或者用如下命令查看tftp服务是否开通:

\$netstat -a/grep tftp

若TFTP服务器没有配置成功,需要按照上述步骤重新检查一遍。

# 6.2.4 配置 NFS 服务器

大家可以通过主机系统的图形界面来配置NFS服务,命令行的配置如下:

编辑/etc/exports文件,设置如下:

\$vi /etc/exports

在该空文件中添加如下行:

/ (rw)

其中"/tftpboot"是共享的目录,也是目标系统的根文件系统目录;"192.168.1.46" 表示该服务提供给所有的主机和网络。

然后重新启动nfs服务,刚才的设置就生效了

\$/etc/init.d/nfs restart			
关闭 NFS mountd:	[	确定	]
关闭 NFS 守护进程:	[	确定	]
Shutting down NFS quotas:	[	确定	]

关闭 NFS 服务:	[	确定		
启动 NFS 服务:	[	确定		
Starting NFS quotas:	[	确定		
启动 NFS 守护进程:	[	确定		
启动 NFS mountd:	[	确定		
		<u>)</u> / H	1	

注意,在使用nfs服务的请大家确认防火墙没有限制nfs服务,并且,在系统服务中 portmap服务是启动运行的。

现在我们测试一下nfs服务,假设有nfs服务的主机的IP为192.168.1.180,我们在 192.168.1.180上测试,首先在192.168.1.180上建立目录/home/tmp,然后用超级用户执行 下列命令

\$mount 192.168.1.180:/tftpboot /home/tmp

这样,将192.168.1.180上的目录"/tftpboot"通过NFS挂载到了192.168.1.180的目录 "/home/tmp"上,在192.168.1.180上查看一下目录"/home/tmp"里面的内容是否和 192.168.1.180 上目录"/tftpboot"里面的内容完全一样,确定NFS服务可用。

# 6.2.5 设置 minicom

在 Linux 操作系统下,使用 minicom 作为超级终端,因此首先必须正确配置超级终端。 在宿主机端 Linux 下执行:



第一次使用时需要对串口进行配置,在该菜单下选择"Serial port setup",然后回 车将出现如下配置菜单:



参照上面的配置,相应输入左边的"A"、"B"等字符将出现配置各自参数的界面,分别配置成上述设置。

设置完成后回车退出,并选择保存,如下所示:



然后选择 Exit 菜单,将启动 minicom 程序,此时请将 JXARM9-2410 的 UARTO 连接到 PC 机串口并将 JXARM9-2410 重新上电,在 minicom 中,将出现如下所示界面:



此时表示 minicom 已经配置正确。

当下次使用 minicom - s 进入 minicom 时,如果已经将设置保存,请直接选择 Exit 菜 单进入到 minicom 即可。

# 6.3 安装 Linux 编译环境和源程序

在宿主机端,插入 JXARM9-2410 光盘到光驱中,在 Linux 下执行:

\$mount /dev/cdrom /mnt

若系统不识别/dev/cdrom的话,使用如下命令,假设 CDROM 为第二个 IDE 口的主驱,即为/dev/hdc,则:

\$mount -t iso9660 /dev/hdc /mnt

然后读取光盘内容并安装:

\$cd /mnt

\$cd linux

然后执行脚本文件 linuxinstall, 该程序将自动安装编译器, Linux 源代码, 在安装之前请确保已经建立 cvtech 用户名,并保证以 root 登陆。

\$./linuxinstall

如果一切正常,将得到编译结果,在/tftpboot/目录下将产生zImage文件,该文件为 Linux 映像,然后通过 bootloader 下载到 JXARM9-2410 中运行。

# 6.4 Linux 内核编译

Linux 安装环境正确以后,就可以开始编译内核:



加载配置文件 config-jx2410



Enter 键进入,并输入 config-jx2410



然后保存



然后执行

\$make zImage

将编译生成自己定制的内核映像文件,并自动被复制到/tftpboot/目录。

## 6.5 Linux 映像下载并运行

首先将需要下载的内核映像或者 ramdisk 文件拷贝到宿主机的/tftpboot 目录, 然后启动 minicom:

#### \$minicom -s

将 JXARM9-2410 的串口 0 与 PC 的串口 0 连接,将 JXARM9-2410 的网卡和 PC 的网卡使用 直连网线进行连接,或者将 JXARM9-2410 和 PC 机使用普通网线接入同一局域网。

然后将 JXARM9-2410 上电,正常情况下将在 minicom 中看到 bootloader 菜单界面。然 后在 bootloader 中按照如下提示进行:

```
_____
     done
  Bytes transferred = 4632504 (46afb8 hex)
  JX2410 # tftp 30008000 zlmage
  RTL8019AS Founded!
  MAC : 0x8-0x0-0x3e-0x26-0xa-0x5b
  MAC: 0x0:0x0:0x0:0x0:0x0:0x0
  TFTP from server 192.168.1.180; our IP address is 192.168.1.45
  Filename 'zImage'.
  Load address: 0x30008000
  done
  Bytes transferred = 871740 (d4d3c hex)
  JX2410 #go 30008000
  Uncompressing Linux.....
                 ..... done.
booting the kernel.
  Linux version 2.4.18-rmk7-pxal (root@Linux-Lizm) (gcc version 2.95.3 20010315 (release))
#133 四 11 月 18 11:07:55 CST 2004
  CPU: ARM/CIRRUS Arm920Tsid(wb) revision 0
  Machine: Samsung-SMDK2410
  .....
```

如果 tftp 出现失败,则可能存在一下几种问题:

1、网络没有正确连接:使用网线不对,或者宿主机和目标机的 IP 地址不在同一子网, 此处我们要求,宿主机和目标机都在 192.168.1.xxx 网段,而且,最后一位不能相同。

2、所传送的文件在宿主机的/tftpboot 目录下不存在。

# 6.6 Linux 应用程序编写以及执行

本节将以一个 helloworld 程序为例说明如何在 JXARM9-2410 中编写应用程序

# 6.6.1 编译 helloworld 应用程序

1. 建立工作目录

注:本实验以及后续的所有实验中用"\$"符号表示在主机的 Linux 控制台上输入的命令行。用"#"符号表示在目标机的 Linux 控制台上输入的命令行。

<ul> <li>2. 编写程序源代码</li> <li>实际的源代码较简单,如下:</li> <li>#include <stdio.h></stdio.h></li> </ul>	\$cd /home/cvtech/jx2410/examples \$mkdir helloworld \$cd helloworld
<pre>#include <stdio.h> </stdio.h></pre>	2. 编写程序源代码 实际的源代码较简单,如下:
<pre>int main() {     printf("Hello, World!\n");</pre>	<pre>#include <stdio.h> int main() {     printf("Hello, World!\n");</stdio.h></pre>

3. 编写 Makefile 文件

使用 vi 编辑工具编辑 Makefile,请注意文件名的 M 必须大写,其余为小写,如下所示,。注意其中每行前面的空格位置必须使用'Tab'键。

```
CC = /usr/local/arm/2.95.3/bin/arm-linux-gcc
LD = /usr/local/arm/2.95.3/bin/arm-linux-ld
EXEC = helloworld
OBJS = helloworld.o
CFLAGS +=
LDFLAGS +=
all: $(EXEC)
$(EXEC): $(0BJS)
       $(CC) $(LDFLAGS) -o $@ $(OBJS) $(LDLIBS$(LDLIBS $@))
clean:
       -rm -f $(EXEC) *.elf *.gdb *.o
4. 使用 make 进行编译
使用如下命令编译 ARM 平台的 helloworld 程序。
$make clean
$make
/usr/local/arm/2.95.3/bin/arm-linux-gcc -c -o helloworld.o helloworld.c
/usr/local/arm/2.95.3/bin/arm-linux-gcc -o helloworld helloworld.o
```

编译成功将生成 helloworld 文件,这个文件就是能够在 JXARM9-2410 中运行的应用程序,然后通过下面几种方法将该应用程序下载到 JXARM9-2410 的 Linux 中运行。

## 6.6.2 通过 NFS 将 helloworld 应用程序下载到 JXARM9-

## 2410

JXARM9-2410 通过 NFS 将主机的/tftpboot/目录挂接到目标机的/mnt/nfs 目录中,因此,需要将上面编译生成的程序 helloworld 拷贝到主机的/tftpboot/目录或其子目录下,将该例程放置到/tftpboot/examples/目录下。

\$cp helloworld /tftpboot/examples/

在目标机端启动Linux,并在Linux命令行下输入如下命令将主机端/tftpboot/目录挂接到/mnt/nfs/目录下:

#mount 192.168.1.180:/tftpboot/ /mnt/nfs

然后就可以运行 helloworld 程序:

```
#cd /mnt/nfs/examples/
#. /helloworld
正确的结果将在 MiniCom 上打印如下字符串:
Hello, World!
```

# 6.6.3 将 helloworld 应用程序添加到 ramdisk 并重新下载

# 内核运行

还有一种方法就是将新生成的 helloworld 文件添加到 ramdisk,并重新制作 ramdisk, 然后下载内核和文件系统运行。

/home/cvtech/jx2410/root/ramdisk.gz

ramdisk.gz为Linux的文件系统映像压缩文件。用户可以在文件系统中加入自己的应用,例如可以将 ramdisk.gz 拷贝到根目录下,新建一个 ramdisk 目录并解开 ramdisk.gz:

\$cd /home/cvtech/jx2410/root/
\$mkdir rd

\$gunzip ramdisk.gz

在/home/cvtech/jx2410/root/目录下会生成 ramdisk, ramdisk 为解开后的 Linux 的文件系统映像文件。

再将 ramdisk 文件系统映像文件 mount 到新建目录 rd 中:

\$mount -o loop ramdisk rd/ \$cd rd/ \$ls

bin dev etc lib linuette mnt proc qt sbin tmp usr var 这个就是Linux的文件系统,与目标板启动后的文件系统完全一样。

此时用户可以加入自己的应用程序。

下面举例说明将上一节生成的 hello 加入到文件系统:

\$cd rd

```
$cp /home/cvtech/jx2410/examples/helloworld/helloworld.
$cd /home/cvtech/jx2410/root/
$umount rd
压缩新生成的 ramdisk 文件系统映像文件:
```

\$gzip ramdisk ramdisk.gz

# 6.6.4 将 helloworld 应用程序使用 u 盘下载运行

JXARM9-2410 支持两个 USB HOST,可以挂接 u 盘,因此可以将应用程序先拷贝到 u 盘, 然后将 u 盘挂接到 JXARM9-2410 中,然后运行。

在宿主机端:插入u盘

```
$mount /dev/sda1 /mnt
$cd /home/cvtech/jx2410/examples/helloworld
$cp helloworld /mnt
$umount /mnt
```

从宿主机端拔下u盘,然后将u盘插入到HOST1或者HOST2,然后启动JXARM9-2410的 Linux,在其命令行中输入:

#cd /mnt/jffs2 #mount /dev/sda1 /mnt/udisk #cd /mnt/udisk #./helloworld

## 6.7 Linux JFFS2 文件系统

在 JXARM9-2410 NOR flash 的 0x1080000-0x1800000 的位置上是 JFFS2 文件系统,其制 作和烧写方法请参考"<u>u-boot 烧写 JFFS2 文件系统</u>"。

# 6.8 Linux CRAMFS 文件系统

在 PC 中制作 CRAMFS 文件系统使用 mkfs. cramfs 工具,该工具在 /home/cvtech/jx2410/root/目录下,下面的示例将/home/cvtech/jx2410/root/cramfs/目 录下的内容制作成 CRAMFS 文件系统,并保存为 prog. cramfs 文件,这个文件就是制作后生

成的文件系统映像文件。

```
$cd /home/cvtech/jx2410/root
$/home/cvtech/jx2410/root/mkfs.cramfs cramfs/ prog.cramfs
$/s prog.cramfs -/
-rw-r--r-- 1 root root 5574656 11 月 18 09:58 prog.cramfs
$cp /home/cvtech/jx2410/root/prog.cramfs /tftpboot
```

然后将 prog. cramfs 映像文件通过 imagewrite 烧写到 NAND FLASH 中。这是在 JXARM9-

2410的Linux下进行的。运行 JXARM9-2410的Linux。

```
# imagewrite /dev/mtd/4 -part 0
# imagewrite /dev/mtd/4 /mnt/nfs/prog.cramfs:0
使用如下命令将 CRAMFS 文件系统挂载上去:
# mount -t cramfs /dev/mtdblock/4 /mnt/cramfs
# ls /mnt/cramfs
```

# 6.9 Linux MP3 测试

在系统提供的 CRAMFS 文件系统中有 mp3 播放器和一个 mp3 测试文件 tsyj.mp3,可以使用它们测试 mp3:

#/mnt/cramfs/bin/madplay /mnt/cramfs/mp3/tsyj.mp3

另外,也可以播放 u 盘中的 mp3 文件:

在宿主机端: 插入u盘

```
$mount /dev/sda1 /mnt
$cd /home/cvtech/jx2410/examples/helloworld
$cp /home/cvtech/jx2410/root/cramfs/mp3/tsyj.mp3 /mnt
$umount /mnt
```

从宿主机端拔下u盘,然后将u盘插入到HOST1或者HOST2,然后启动JXARM9-2410的 Linux,在其命令行中输入:

#mount /dev/sda1 /mnt/udisk #cd /mnt/udisk #/mnt/cramfs/bin/madplay tsyj.mp3

# 第七章 u-boot 使用说明

JXARM9-2410 使用 u-boot 作为 bootloader, 通过它可以实现如下功能:

1、烧写 flash;

2、引导 Linux 操作系统及其它程序;

3、通过以太网下载 Linux 操作系统或其它程序并引导。

首先必须将 u-boot 烧写到 flash 的起始扇区, JXARM9-2410 中, u-boot 使用第 1-2 共两个扇区,其中第一个扇区为 u-boot 映象,第二个扇区保存 u-boot 环境变量。

因此在使用之前必须烧写 u-boot 映象到第一个扇区。烧写方法有两种:

1、在 Windows 环境下使用 ADT IDE 的 flash programmer 工具进行烧写,在当前 flash 中的 bootloader 没有正确烧写之前必须使用这种方法进行烧写。

2、使用 u-boot 的 flash 烧写功能进行烧写,这必须在已经有 u-boot 映象已经烧写到 flash 中并且能够正常运行的前提下。

## 7.1 使用 ADT IDE 烧写 u-boot

在 JXARM9-2410 安装目录的 flashupdate 目录下有 u-boot 映象文件:u-boot.bin。另外 还有一个命令脚本文件 JX2410\_load.csf, 在使用 ADT IDE 烧写 u-boot 时需要用到。

下面介绍怎样在 ADT IDE 集成开发环境下烧写 u-boot。

1、硬件连接

参照"第二章 硬件安装"进行硬件连接,然后将仿真器和 JXARM9-2410 电源打开。

2、打开 flash programmer 工具

打开 ADT IDE 集成开发环境,并选择其 Debug 菜单的 Flash Programmer 菜单项启动 flash programmer 工具。



3、设置 flash programmer

如下图所示对 flash programmer 进行设置。其中 Image 编辑框中为待烧写的 u-boot 映 象, Command Script 编辑框中为命令脚本。Sector From 和 To 两个编辑框表示烧写到 flash 的位置,此处由于 u-boot 映象必须烧写到第一个扇区,因此选择 1。在 Device 中选 择 Arm9Lpt 或者 Arm9Simple,如果使用 ADT 1000 或者 ADT 1000A 仿真器请选择 Arm9Lpt, 如果使用 JXARM9-2410 内置的简易仿真器,请选择 Arm9Simple。

🖌 Arn Development Tools				- 2 🛛
]] Eile Edit View Project Build Debug	<u>; T</u> ools <u>W</u> indow <u>H</u> elp			
● (* 4) 6 (* *   1 1 1 1 1			~	
	Flash Programmer File Image d:\cvtech\flashupdate\u Command Script d\\cvtech\flashupdate\u Flash Maker: intel Port: 16x2 Start: 0 Target RAM From: 0x3000000 to: 0x3200000	flat bin v -boot.bin ×2410_load.csf × Device: 28F128J3 v Sector: Sector Info Sector From: 1 To 1 Endian Device device: Arm9Lv Dovice Info Endian Device Arm9Lv 0 C Big port: Ipt1 v	Check Blank Erase Program Verify Cancel Auto V 1. Erase 2. Blank 3. Program 4. Verify Auto	
Duild (Dobug) Compand) Find	lin files 1 \ Find in files 2 /	1.1.1		×
	r in nies 1 λ Fina in nies 2 /			; 🖮 🕄 🗧

4、擦除扇区

设置完毕后点击右边的 Erase 按钮进行擦除操作,擦除结果将在对话框的下面的提示框中显示,如果显示 Erase OK 则表示擦除成功。否则请检查设置并重新擦除。

5、编程

在擦除之后请点击右边的 Program 按钮进行编程操作,编程结果将在对话框的下面的提示框中显示,如果显示 Program OK 则表示编程成功。否则请检查设置并重新编程。

6、重新启动

擦除和编程成功后,请关闭 flash programmer 对话框,然后将 JXARM9-2410 的 UARTO 与 PC 串口进行连接并启动超级终端软件,然后将 JXARM9-2410 重新上电,如果烧写正确,将在超级终端中打印类似如下启动信息:

```
U-Boot 1.1.1 (Sep 6 2004 - 12:12:06)
U-Boot code: 33F80000 -> 33F9CED0 BSS: -> 33FA116C
RAM Configuration:
Bank #0: 30000000 64 MB
Flash: 32 MB
In: serial
Out: serial
Err: serial
```

```
Hit any key to stop autoboot: 0
JX2410 #
```

# 7.2 Windows 环境下使用 u-boot

u-boot 使用串口作为输出和命令行输入界面,在 Windows 环境下,可以使用两个工具:

1、超级终端:

请参照"<u>第四章 Windows 软件安装与设置</u>"的"<u>超级终端设置</u>"一节对 windows 超级 终端进行设置,并启动超级终端。

2, dnw:

在 JXARM9-2410 安装目录的 tools 目录下有一个 dnw. exe 文件,运行该程序,并选择 Configuration 的 Options 菜单项进行配置。



参考下图进行设置,关键在于 Baud Rate 以及 COM Port 的设置,设置完成后点击 OK 退出。

UART/USB Optio	ns				
Serial Port Baud Rate	COM Port-	ОК			
115200	⊙ COM 1	Cancel			
C 57600	C COM 2				
C 38400	C COM 3				
C 14400	, com 4				
° 9600					
USB Port					
Download Address 0xc000000					

然后选择 Serial Port 的 Connect 菜单项,如果配置正确,此时 dnw 就开始连接。

	[COH:x] [USB:x]	_ 2	
Serial Port USB Por	rt Configuration Help		
Connec			_
Transmit			
		1 hun 17	
		j 🖮 🕻	) I
		1	

无论使用 Windows 的超级终端或者 dnw,在正确设置后,将 JXARM9-2410 的 UART0 与 PC 串口进行连接,然后将 JXARM9-2410 重新上电,如果此时 u-boot 已经被烧写到 flash 中,

将在超级终端或者 dnw 中显示上节描述的 u-boot 启动信息。

# 7.3 Linux 环境下使用 u-boot

Linux 环境下使用 minicom 作为超级终端工具,设置方法请参考"<u>第六章 Linux 交叉开</u> <u>发流程</u>"的"<u>设置 minicom</u>"一节。和 Windows 环境下一样,设置完成后启动 minicom 并启 动目标系统的 u-boot,看 minicom 中是否有启动信息。

# 7.4 u-boot 烧写 u-boot

如果当前系统已经有 u-boot 烧写到 flash,我们也可以通过 u-boot 烧写自身。

```
JX2410 # protect off all
Un-Protect Flash Bank # 1
IX2410 # erase 1:0
Erase Flash Sectors 0-0 in Bank # 1
Erasing sector 0 ... done
JX2410 # tftp 30000000 u-boot.bin
RTL8019AS Founded!
MAC : 0x8-0x0-0x3e-0x26-0xa-0x5b
MAC: 0x0:0x0:0x0:0x0:0x0:0x0
TFTP from server 192.168.1.188; our IP address is 192.168.1.45
Filename 'u-boot.bin'.
Load address: 0x3000000
done
Bytes transferred = 126976 (1f000 hex)
JX2410 # cp. b 30000000 0 1f000
Copy to Flash...\done
JX2410 #reset
```

# 7.5 u-boot 烧写 Linux 映像文件

在 JXARM9-2410 安装目录的 flashupdate 子目录下有 JXARM9-2410 的 Linux 映象文件以及 ramdisk 文件系统映象文件,分别为 zImage 和 ramdisk.gz。

我们可以将这两个文件或者用户自己编译的Linux 映象文件以及 ramdisk 文件系统映象 文件烧写到 flash 中,并由 u-boot 进行引导,下面将对这个过程进行介绍。

JXARM9-2410 的 flash (Intel 128J3A) 共 32Mbytes,分为 128个扇区,每个扇区 256Kbytes,u-boot 已经使用了 flash 的第一、二个扇区,地址为 0x0-0x7ffff,因此其它 的文件不能烧写到这两个扇区。

下面的例子假设将 zImage 烧写到 0x100000-0x1fffff,将 ramdisk.gz 烧写到 0x200000-0x6fffff,且 zImage 大小小于 1Mbytes,而 ramdisk.gz 的大小小于 5Mbytes。

另外,烧写时需要用到 tftp,因此,建议宿主机在 Redhat 操作系统下进行,安装和配置方法请参考"<u>第六章 Linux 交叉开发流程</u>",此时必须保证 TFTP 服务器正确启动。

然后需要启动 u-boot, 启动方法参见上节"<u>Linux 环境下使用 u-boot</u>"。下面的提示 信息以及输入均是在 minicom 程序中进行的,且在 u-boot 正确启动后在其命令行中进行。

1、拷贝文件到宿主机的 tftp 目录

拷贝 zImage 和 ramdisk.gz 文件到宿主机的/tftpboot 目录下。可以使用 u 盘或者直接 从光盘拷贝,下面介绍从光盘拷贝的方法:

\$mount /dev/cdrom /mnt/cdrom
\$cd /mnt/cdrom/flashupdate
\$cp zlmage /tftpboot
\$cp ramdisk.gz /tftpboot
2、烧写 zlmage

先使用 erase 命令擦除 zImage 所使用扇区,如果将 zImage 烧写到 0x100000-0x1fffff,即 flash 的 4-7 扇区,请注意 flash 扇区的号码是从 0 开始的。

```
JX2410 # erase 1:4-7
Erase Flash Sectors 4-7 in Bank # 1
Erasing sector 4 ... done
Erasing sector 5 ... done
Erasing sector 6 ... done
Erasing sector 7 ... done
JX2410 # tftp 30000000 zlmage
RTL8019AS Founded!
MAC : 0x8-0x0-0x3e-0x26-0xa-0x5b
MAC: 0x0:0x0:0x0:0x0:0x0:0x0
TFTP from server 192.168.1.180; our IP address is 192.168.1.45
Filename 'zImage'.
Load address: 0x3000000
done
Bytes transferred = 757874 (b9072 hex)
JX2410 # cp. b 30000000 100000 b9072
Copy to Flash...\done
也可以使用缺省的环境变量 burnlinuxzImage 完成
JX2410 # run burnlinuxzlmage
```

3、烧写 ramdisk.gz

然后使用 erase 命令擦除 ramdisk.gz 所使用扇区,如果将 ramdisk.gz 烧写到 0x200000-0x6fffff,即 flash 的 8-27 扇区,请注意 flash 扇区的号码是从 0 开始的。

```
IX2410 # erase 1:8-27
  Erase Flash Sectors 8-27 in Bank # 1
  Erasing sector 8 ... done
  Erasing sector 9 ... done
  Erasing sector 26 ... done
  Erasing sector 27 ... done
  JX2410 # tftp 3000000 ramdisk.gz
  RTL8019AS Founded!
  MAC : 0x8-0x0-0x3e-0x26-0xa-0x5b
  MAC: 0x0:0x0:0x0:0x0:0x0:0x0
  TFTP from server 192.168.1.180; our IP address is 192.168.1.45
  Filename 'zImage'.
  Load address: 0x3000000
  done
  Bytes transferred = 3222512 (312bf0 hex)
  JX2410 # cp. b 3000000 200000 312bf0
  Copy to Flash...\done
  也可以使用缺省的环境变量 burnlinuxramdisk 完成
  JX2410 # run burnlinuxramdisk
  4、从 flash 中启动 Linux
  Linux 内核映象和 ramdisk 都烧写到 flash 以后,就可以通过 u-boot 将 Linux 启动起
来,方法如下:
   首先将 ramdisk.gz 拷贝到 sdram 的 0x30800000 地址处:
  JX2410 # cp. b 200000 30800000 500000
  然后将 zImage 拷贝到 sdram 的 0x30008000 地址处:
```

JX2410 # *cp. b 100000 30008000 100000* 然后运行 zImage:

JX2410 # *go 30008000* 

也可以使用缺省的环境变量 bootlinux 完成

JX2410 # run bootlinux

## 7.6 u-boot 烧写应用程序

本节介绍如何将在 ADT IDE 中编写的应用程序烧写到 flash 中运行。

1、在 ADT IDE 环境下编译程序,如下图所示为编译成功后的提示信息:



在工程上点击鼠标右键将可以看到该文件所在的工程路径

යි examples - Arm Development Tools	_ 7 🗙
Ele Edit <u>V</u> iew Project Build Debug <u>T</u> ools <u>Wi</u> ndow <u>H</u> elp	
E I I I → ひ ひ か か 知 = 日 日 日 I I I I I I I I I I I I I I I I	
Image: Star Star Star Star Star Star Star Star	
Aleddemo.c Link	•
leddemo.elf - build successful.	
Build / Debug / Command / Find in files 1 / Find in files 2 /	) 🖮 🛛 🗘
本例中为:	

Project Properties	
File name:	2410\examples\leddemo\leddemo.ap)
Last modified	13:59:58 2004-12-01
Size:	3571 Bytes
	ОК

如果使用缺省设置,在相应工程目录下的 debug 目录下将生成 leddemo. elf 文件。

#### F:\cooljet\EB2410\examples\leddemo\debug \_ & × 文件 (E) 编辑 (E) 查看 (Y) 收藏 (A) 工具 (E) 帮助 (H) **-**🔇 后退 🔹 🌍 🔹 🏂 🔎 搜索 📂 文件夹 🛄 🕶 地址 @) d:\cvtech\jx2410\examples\leddemo\debug 🖌 芛 转到 main.o 0 文件 3 KB **leddemo.elf** ELF 文件 35 KB leddemo.o 0 文件 3 KB 文件和文件夹任务 \$ 💋 创建一个新文件夹 ▲ 将这个文件夹发布到 Web 😂 共享此文件夹 $\mathbb{Q}$ 其它位置 ۲ 🛅 leddemo 📋 我的文档 □ 共享文档 🚽 我的电脑 🧐 网上邻居 详细信息 \* ) 🖮 🕐 😳

#### JXARM9-2410-3 用户手册

2、生成 binary 文件

然后在 ADT IDE 环境下,使用 tools 下的 elf to binary 功能生成 binary 文件。请注 意,一定要将所需要生成 binary 文件的工程设置为当前活动工程,即在工作区窗口中的工 程列表中必须为黑体字表示,如下图所示的 leddemo 工程。



此时再去查看 debug 目录将多了一个文件 leddemo. bin,如下图所示:

#### F:\cooljet\EB2410\examples\leddemo\debug \_ @ 🛛 文件 (E) 编辑 (E) 查看 (Y) 收藏 (A) 工具 (E) 帮助 (H) 🔇 后退 🔹 🌍 🔹 🏂 🔎 搜索 📂 文件夹 🛄 🗸 地址 (1) d:\cvtech\jx2410\examples\leddemo\debug 🖌 🄁 转到 **leddemo.elf** ELF 文件 35 KB leddemo.o 0 文件 3 KB main.o 0 文件 3 KB leddemo BIN 文件 1 KB 文件和文件夹任务 2 💋 创建一个新文件夹 🔕 将这个文件夹发布到 Web 😂 共享此文件夹 其它位置 \* 🛅 leddemo 📋 我的文档 🛅 共享文档 🔵 我的电脑 🧐 网上邻居 详细信息 \* ) 🖮 🛛 🙄

### JXARM9-2410-3 用户手册

这个文件就是可以直接烧写到 flash 中运行的文件。

3、烧写 leddemo. bin 文件到 flash

接下来的工作就是要将 leddemo. bin 文件烧写到 flash 中,在 JXARM9-2410 中,我们将 从 0x80000-0xfffff 的 flash 空间,共 512Kbytes 作为用户程序使用,小于 512Kbytes 的程 序都可以烧写到该区间。

可以通过像烧写 Linux 映象一样通过 tftp 将 leddemo. bin 烧写到 flash,对于小的应 用程序,也可以通通过串口线进行下载,当然,速度会比较慢。

在 u-boot 中使用 loadb 命令将程序下载到 sdram 中,如下所示:

JX2410 # loadb 3000000
将程序下载到 0x30000000 地址处,或者使用
JX2410 # <i>loadb</i>

将程序下载到缺省内存地址,如下图所示:
💑 JX44b0 - 超级终端							
文件(E) 编辑(E) 查看(Y) 呼叫(C) 传送(I) 帮助(H)							
	) 🖆 🚿 💼 🎦 🖆				N N N N N N N N N N N N N N N N N N N		
	/Tx DMAbank # 64:UART0 Rx/Tx FIFO protect off start end	65:UART0 AFC Tx	66:UARTØ AFC R×	67:UART1 Rx			
	U-B /Tx Intmake FL Wait0 Eth 84:Stone Test SMDK2410 # tftpboottin	85:ETC NEC Int me	86:nBATT_FAULT int	87:Core Cac	I		
	88:NAND View Bad Block	k 89:NAND View Page	90:NAND Write	91:NAND Era			
	se 92:NAND ECC	93:NOR Flash Program	94:Flash Programmer				
	Select the function to	o test :					
	U-Boot 1.1.1 (Aug 14 2	-Boot 1.1.1 (Aug 14 2004 - 09:06:37)					
	U-Boot code: 33F80000 RAM Configuration: Bank #0: 30000000 64 I Flash: 512 kB *** Warning - bad CRC						
	In: serial Out: serial Err: serial SMDK2410 # SMDK2410 # SMDK2410 # SMDK2410 #						
	## Ready for binary (	kermit) download to 0x33	3000000 at 115200 bps			<b>.</b>	
 已连接 2:32:52 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印							
■开始 @ ♀ ♀ ∽ » @ E © 昱 S @ S S S & S @ S & C & S S & C & 12:58							
	然后选择"传送	"菜单:					

7. TX44b0 - 親毎終端						
文件(方) 编辑(方) 杳看(八) 呼叫(C) 传送(石) 帮助(日)						
	- 🔰					
00000000: ea0 protect off start end	<u> </u>					
U-B /Tx Intmake FL 68:UART1 Rx/Tx DMA protect off N:SF[-SL] SMDK2	ntmake FL RT1 Rx/Tx DMA 69:UART1 Rx/Tx FIFO 70:UART1 AFC Tx 71:UART1 AF ct off N:SF[-SL]					
88:NAND View Bad Block 89:1 #发送文件 ? × HAND Era						
Se 92:NAND ECC 93:I 文件夹: D:\EB2410\Diag\debug						
Select the function to tes 文件名(E):						
U-Boot 1.1.1 (Aug 14 2004 世後) [D:\EB2410\Diag\debug\Diag.bin 测览(图)]						
U-Boot code: 33F80000 -> 3: RAM Configuration: Bank #0: 30000000 64 MB Flash: 512 kB **** Warning - bad CRC, using our source on the summers						
In: serial Out: serial Err: serial SMDK2410 # SMDK2410 # SMDK2410 # SMDK2410 # SMDK2410 #						
## Ready for binary (kermit) download to 0x33000000 at 115200 bps						
	•					
<b>已连接 2:33:41 自动检测 115200 8-N-1</b> SCROLL CAPS NUM 捕 打印						
III开始 / 《 · · · · · · · · · · · · · · · · · ·						
在"文件名"编辑框中选择 leddemo.bin 文件, "协议"中选择"Kermit"	,然后点击					

"发送"按钮。然后将弹出一个对话框,并提示传送进度,如下图所示:

😵 JX44b0 - 超级终端	<u>_ 8 ×</u>					
文件(E) 编辑(E) 查看(⊻) 呼叫(C) 传送(工) 帮助(出)						
00000000: ea000012 ? Onstant addr 00000000: ea0 protect off start end	<u> </u>					
U-B /Tx Intmake FL 68:UART1 Rx/Tx DMA protect off N:SF[-SL SMDK2 正在发送: D:\EB2410\Diag\debug\Diag.bin	NF					
88:NAND View Bad Blo	a					
Se 数据也: 102 文件数: 1771						
重试次数: 0 重试总次数: 1						
Select the function						
U-Boot 1.1.1 (Aug 14						
U-Boot code: 33F8000 文件: 本任: 本任: 本任: 本任: 本任: 本任: 本任: 本任: 本任: 本任						
Bank #0: 30000000 64 Flash: 512 kB						
*** Warning - bad CR						
In: serial 取消						
Err: serial	-					
SMUK2410 #   SMUK2410 #						
SMDK2410 #						
SMDK2410 # loadb						
## Ready for binary (kermit) download to 0x33000000 at 115200 bps						
	·					
已達接 2:33:55 自动检测 115200 8-N-1  SCROLL  CAPS  NUM  捕  打印						
[1977년] 🧶 🍳 🥗 "] (환) 🔄 🖾 🔜 🔄 11 🔄 🔄 🔄 11 🕼 🕄 🖉 11 📑 🕵	E 🕮 🛃 🥝 🎬 🔯 🏷 🕒 🛛 12:59					
传送完毕后,将打印传送文件的大小以及下载地址,如下图所示,	leddemo.bin 文件被					

传送到 0x33000000 地址处,大小为 0x49d10 字节。

```
🌯 JX44b0 - 超级终端
                                                                                                    _ & ×
文件(E) 编辑(E) 查看(⊻) 呼叫(C) 传送(T) 帮助(H)
                                                                                               V
0 🖻 🙍 🔏 🖻 🗃
 protect off start end
                                                                                                         U-B
 /Tx Intmake FL
68:UART1 Rx/Tx DMA
                         69:UART1 Rx/Tx FIF0
                                                 70:UART1 AFC Tx
                                                                         71:UART1 AF
 protect off N:SF[-SL]
 SMDK2
 ech
     - make secto
 92:NAND ECC
                         93:NOR Flash Program
                                                 94:Flash Programmer
 Select the function to test :
 U-Boot 1.1.1 (Aug 14 2004 - 09:06:37)
 U-Boot code: 33F80000 -> 33F96C34 BSS: -> 33F9AD04
 RAM Configuration:
Bank #0: 30000000 64 MB
 Flash: 512 kB
 *** Warning – bad CRC, using default environment
 In:
        serial
                                                      L
 Out:
        serial
        serial
 Err
 ŠMDK2410 #
 SMDK2410 #
 SMDK2410 #
 SMDK2410 #
 SMDK2410 # loadb
 ## Ready for binary (kermit) download to 0x33000000 at 115200 bps...
## Total Size = 0x00049d10 = 302352 Bytes
## Start Addr = 0x33000000
 SMDK2410 #
                                                                                                         -
已连接 2:35:52 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印
MATHA 🥔 🍳 🥙 ») 🔄 💽 🛅 🔜 🔄 🔄 🔄 🔄 🔄 🚭 🛃 🐑
                                                                              😹 🕀 🕮 🗖 🧔 🍊 🎦 ڭ 🖽 🛛 13:01
     然后进行烧写:
     JX2410 # erase 1:2-3
     Erase Flash Sectors 2-3 in Bank # 1
     Erasing sector 2 ... done
     Erasing sector 3 ...
                               done
     JX2410 # cp. b 33000000 80000 49d10
     Copy to Flash...\done
```

4、从 flash 中启动该程序

首先将 leddemo. bin 拷贝到 sdram 的 0x30000000 地址(这个地址可以通过修改连接脚本进行修改,对于 JXARM9-2410,如果不修改连接脚本,我们缺省使用的地址是 0x30000000 地址)处:

```
JX2410 # cp. b 80000 30000000 100000
然后运行 leddemo:
```

```
JX2410 # go 3000000
```

将运行你所编译的程序,可以观察结果看效果,对于 leddemo 的效果就是数码管将闪烁。

# 7.7 u-boot 烧写 JFFS2 文件系统

在 JXARM9-2410 NOR flash 的 0x1080000-0x1800000 的位置上是 JFFS2 文件系统,在宿

主机中制作 JFFS2 文件系统使用 mkfs. jffs2 工具,该工具在/home/cvtech/jx2410/root/目录下,/home/cvtech/jx2410/root/jffs2/目录下的内容为 JFFS2 文件系统的内容,其中包含 Linux 操作系统下示例程序代码以及内核模块,下面将该目录下的内容制作称为 jffs2 文件系统映像 prog. jffs2。

\$cd /home/cvtech/ix2410/root \$/home/cvtech/jx2410/root/mkfs.jffs2 -d jffs2 -o prog.jffs2 \$ Is prog. jffs2 -1 -rw-r--r--1 root root 5574656 11 月 18 09:58 prog. jffs2 \$cp /home/cvtech/jx2410/root/prog.jffs2 /tftpboot 然后将生成的 prog. jffs2 文件烧写到 flash 中。 JX2410 # erase 1:66-95 Erase Flash Sectors 66-95 in Bank # 1 Erasing sector 66 ... done Erasing sector 67 ... done Erasing sector 68 ... done Erasing sector 69 ... done Erasing sector 94 ... done Erasing sector 95 ... done JX2410 # tftp 3000000 prog. jffs2 RTL8019AS Founded! MAC : 0x8-0x0-0x3e-0x26-0xa-0x5b MAC: 0x0:0x0:0x0:0x0:0x0:0x0 TFTP from server 192.168.1.180; our IP address is 192.168.1.45 Filename 'prog. jffs2'. Load address: 0x3000000 done Bytes transferred = 5574656 (551000 hex) IX2410 # cp. b 30000000 1080000 551000 Copy to Flash...\done 请注意,最后的 cp. b 命令的最后一个参数是烧写的字节数,是 tftp 命令的最后一行中 括号里面的数据。

JX2410 # **/s** 

看其中内容与/home/cvtech/jx2410/root/jffs2 目录下的内容是否一致,如果一致表示烧写正确。

也可以使用缺省的环境变量 burn jffs2 完成

JX2410 # run burnjffs2

# 7.8 u-boot 烧写 uCOS-II 演示程序

JXARM9-2410 教学系统中还包含一个 uCOS-II 的演示程序,在

/home/cvtech/jx2410/flashupdate/demo.bin 文件中。

可以使用缺省的环境变量 burndemo 完成该程序的烧写:

JX2410 # run burndemo

烧写完成后通过如下命令运行该程序:

JX2410 # run bootdemo

# 7.9 恢复 u-boot 缺省环境变量

在 JXARM9-2410 中使用 NOR flash 的 0x1080000-0x1800000 的位置上是 JFFS2 文件系 统,在宿

JX2410 # protect off all Un-Protect Flash Bank # 1 JX2410 # erase 1:1 Erase Flash Sectors 1-1 in Bank # 1 Erasing sector 1 ... done JX2410 #reset

# 第八章 uCOS-II 演示程序 GPRS、GPS 功能测试

JXARM9-2410 教学系统中还包含一个 uCOS-II 的演示程序,采用全菜单界面通过小键盘的"UP"、"DOWN"以及"ENTER"键进行操作。

另外,JXARM9-2410-3还包含一个完整的手机解决方案和GPS全球定位模块,可以在其 上实现电话、短信以及全球定位等功能。在这个演示程序中均可以进行测试。

### 8.1 硬件连接

1. 插入 SIM 卡到 SIM 卡座;

2. 如果要使用电话功能需要接上耳机和麦克风,短信功能可以不接;

3. 接上 GPRS 天线;

4. 连接 GPS 天线。

然后上电并运行如下命令:

JX2410 # *run bootdemo* 以启动 uCOS-II 演示程序。

## 8.2 电话功能

在系统测试程序中选择"20. GPRS-电话"选项,并按"Enter"键确定,将进入电话功能主界面。

### 打电话:

在上一界面中输入"1"键,将进入电话主叫功能。

然后输入对方电话号码并以"Enter"键确认并开始拨号。也可以按下"Cancel"键退 回到电话功能主界面。

拨号后,如果对方接听,便可以使用耳机和麦克风与对方通话,同时在此过程中也可以 按下"Cancel"键挂机。

### 接电话:

在电话功能主界面中,如果有电话输入,屏幕将提示来话:

按下"Enter"键接听电话,使用耳机和麦克风可以开始和对方通话。

## 8.3 短信功能

短信功能可以不连接耳机和麦克风。

在系统测试程序中选择"21. GPRS-短信"选项,并按"Enter"键确定,将进入短信功能主界面.

### 设置短信中心号码:

进入短信功能后首先将进入短信中心设置界面,在正确收发短信之前必须设置正确的短 信中心号码,系统提供了中国移动武汉局的短信中心号码为缺省的设置,使用武汉移动 SIM 卡可以不修改,直接按 "Cancel"键保持设置,如果是其他的卡需要修改该设置,按下 "Enter"键进入修改界面。 输入类似"8613800270500"的号码,并按下"Enter"键修改,并提示修改成功,注意 一般手机中短信中心的设置86前面有个"+"号,在此处设置中不需要输入该符号,直接输 入后面的号码即可。

修改成功后或者在短信功能主界面按下"Cancel"跳过设置均进入提示输入号码界面。 发短信:

输入对方手机号码并以"Enter"键开始发送,或者按下"Cancel"键回到短信功能主界面。

可以接着输入号码进行另外一次发送或者"Cancel"键退出。

收短信:

在短信中心设置完毕、网络连接成功后,就可以接收短消息,如果有短消息,系统将自 动读出并显示出来。

## 8.4 GPS 功能

在使用 GPS 功能时,请按照 8.1 节进行硬件连接,然后将系统通电。

在系统测试程序中选择"19. GPS 测试"选项,并按"Enter"键确定,将进入 GPS 功能主界面,系统将自动定时获取当前全球定位信息(格林威治时间、北京时间、经度和纬度等)并打印出来。