

ARMSYS44B0-C嵌入式系统开发板 使用说明书

(Ver. 20061011)



杭州立宇泰电子有限公司

<http://www.hzlitai.com.cn>

前 言

感谢您使用杭州立泰电子有限公司的 ARMSYS44B0-C 开发板！

随着计算机技术逐渐渗透到各类电子产品当中，一种实用、高效的计算机系统——嵌入式系统不断展现出它独特的魅力。与桌面计算机不同，嵌入式计算机系统专门服务于特定需求，一般要求低成本、低功耗、轻型、高性能、高可靠性及可配置性。嵌入式系统日益广泛的应用也让人们看到了这项蕴涵的巨大的市场潜力。市场的需求带动了对技术人才的需求，今天，大批的技术人员和学者的目光都被吸引到嵌入式系统的设计与开发这门技术上。拥有一套好的嵌入式系统开发平台是每位技术人员的心愿，而 ARMSYS44B0-C 开发板就满足了您的需求。

ARMSYS44B0-C 开发板使用建议：

1. 了解 ARMSYS44B0-C 的硬件组成，你可以从附带光盘中提供电路原理图中找到更详细的电路资料。
2. 了解 ARMSYS44B0-C 提供的开发环境，并熟悉各种开发工具的使用。在本说明书的第四章有关于开发工具的使用例子，一步步指导用户如何使用各种开发工具。
3. 了解 ARMSYS44B0-C 提供的实例项目，我们提供 source（源代码包）其中包含了完整的，经过验证的实例代码。用户可以首先读懂这些代码，然后编译、仿真调试，观察实验现象。最后可根据自身需要修改这些代码来满足特定的应用需求。
4. 可以采用配套的书籍配合学习。我们采用的配套教材为《嵌入式系统设计与开发实例详解》。其中有实例分析、运行结果等。
5. 用户拿到开发板套件后请先仔细阅读该说明书，然后运行开发板测试程序，了解开发板的基本使用方法，同时验证开发板的各项功能。

目 录

前 言.....	2 -
目 录.....	3 -
第一章 ARMSYS44B0-C开发板概述.....	6 -
一、ARMSYS44B0-C嵌入式开发板主要由以下部分组成.....	6 -
二、ARMSYS44B0-C嵌入式开发板配套光盘资料介绍.....	6 -
1.2.1 “BIOS” 文件夹.....	6 -
1.2.2 “source” 文件夹.....	6 -
1.2.3 “uC/OS-II” 文件夹.....	7 -
1.2.4 “uClinux” 文件夹.....	7 -
1.2.5 “电路图” 文件夹.....	7 -
1.2.6 “开发工具” 文件夹.....	7 -
1.2.7 “芯片资料” 文件夹.....	7 -
1.2.8 ARMSYS44B0-C开发板使用说明书.....	8 -
1.2.9 uclinux在armsys上的使用说明书.....	8 -
三、ARMSys44B0-C的启动程序和地址空间分配.....	8 -
四、ARMSYS44B0-C开发板接口及其他.....	9 -
1.4.1 系统基本参数设定.....	9 -
1.4.2 开发板中断分配.....	10 -
1.4.3 开发板对外接口与指示说明.....	10 -
五、JTAG调试板的接口说明.....	11 -
第二章 ARMSYS44B0-C开发板电路描述.....	12 -
2.1 电源电路.....	12 -
2.2 复位电路.....	12 -
2.3 CPU单元电路.....	12 -
2.4 SDRAM电路.....	13 -
2.5 NorFlash电路.....	14 -
2.6 异步串行接口电路.....	14 -
2.7 RTL8019 网络接口电路.....	14 -
2.8 LCD和触摸屏接口电路.....	15 -
2.9 USB接口电路.....	16 -
2.10 ATA (IDE) 接口电路.....	16 -
2.11 按键、蜂鸣器和LED电路.....	17 -
2.12 IIC接口电路.....	18 -
第三章 BOOTLOADER使用说明.....	19 -
3.1 ARMSYS44B0-C开发板工作环境的建立.....	19 -
3.1.1 开发板工作电源.....	19 -
3.1.2 建立和设置超级终端.....	19 -
3.1.3 开发板接入以太网.....	21 -
3.2 BOOTLOADER介绍.....	21 -
3.2.1 USB口下载工具.....	21 -
3.2.2 采用XMODEM下载（通过串行口）.....	25 -

第四章 ARM开发环境介绍	28 -
4.1 ADS1.2 集成开发环境简介与安装	28 -
4.2 JTAG调试代理软件的安装与使用	28 -
4.3 使用CodeWarrior 建立工程并进行编译	30 -
4.3.1 调入模板或重新建立项目	31 -
4.3.2 在工程中添加源文件	35 -
4.3.3 工程进行编译和连接	36 -
4.4 使用AXD 进行仿真调试	37 -
4.4.1 调试前的准备	37 -
4.4.2 AXD调试器的设置	37 -
4.4.3 AXD调试器的使用	39 -
4.4.4 AXD观测窗口	40 -
4.4.5 程序全速运行	41 -
第五章 程序代码下载与存储	42 -
5.1 程序文件下载至开发板的途径	42 -
5.1.1 通过USB下载程序文件	42 -
5.1.2 通过串口下载程序文件	44 -
5.2 采用bootloader烧写程序代码到NorFlash中	45 -
5.3 启动代码的烧写（系统恢复）	45 -
5.3.1 使用FlashPGM 快速烧写Flash	45 -
5.3.2 用programmer工程烧写	51 -
第六章 ARMSYS44B0-C测试程序	52 -
6.1 测试前的准备	52 -
6.2 利用bootloader测试开发板的各项功能	52 -
开发板的测试步骤如下:	52 -
Step1: 正确连接开发板与计算机的接口电缆	52 -
Step2: 打开电源	52 -
Step3: 系统复位	52 -
Step4: Nandflash格式化	53 -
Step5: 运行Nandflash中已经保存为文件的程序	53 -
Step6: ARMSYS功能部件测试	53 -
第七章 uCLinux的使用说明	59 -
7.1 uCLinux简介	59 -
7.2 下载和运行uCLinux	59 -
7.2.1 uCLinux运行环境的建立	59 -
7.2.2 uCLinux内核的存储	59 -
7.2.3 运行uCLinux	60 -
7.3 配置和编译uCLinux	60 -
7.3.1 解压uCLinux 移植包	60 -
7.3.2 安装编译环境	61 -
7.3.3 配置和裁减uCLinux	61 -
7.3.4 编译uCLinux	61 -
7.4 uCLinux 的应用开发	62 -
7.4.1 开发模式	62 -

7.4.2 调试方法	- 62 -
7.5 在Linux操作系统下程序代码的下载	- 62 -
使用minicom终端仿真程序	- 62 -

第一章 ARMSYS44B0-C 开发板概述

ARMSYS44B0-C 嵌入式系统开发板（基于 S3C44B0X 处理器）提供做工精良、资源丰富的硬件电路板，和完整全面、极具价值的源代码包和开发工具，是价廉物美、物超所值的 ARM7 开发板。**本说明书详细讲解**，系统、完整、开放的技术内容，使您能够快速将 ARM 技术应用到新产品的开发中去。

ARMSYS44B0-C 嵌入式开发板采用 Samsung（三星）公司推出的 16/32 位 RISC 处理器 S3C44B0X（ARM7TDMI 内核），为手持设备和一般类型应用提供了高性价比和高性能的微控制器解决方案，为了使应用系统降低成本，S3C44B0X 提供了丰富的内置资源，包括：8KB cache，可选内部 SRAM，LCD 控制器（支持到 256 色 DSTN），带自动握手的 2 通道 UART（兼容 IrDA1.0 标准，具有 16-byte FIFO），4 通道 DMA，存储系统管理器（片选逻辑，FP/EDO/SDRAM 控制器），PWM 功能的 6 通道定时器，71 个通用 I/O 端口，实时时钟 RTC，8 通道 10 位 ADC，IIC-BUS 接口，同步 SIO 接口和 PLL 倍频器。

一、ARMSYS44B0-C 嵌入式开发板主要由以下部分组成

- **CPU**: S3C44B0X, 8MHz 外部钟振，内部倍频至 64MHz；
- **RAM**: 64Mbit (4×1M×16bit) PC100/PC133 兼容 SDRAM；
- **Flash ROM**: SST39VF160 16Mbit (1M×16bit) 程序 Flash；
- **串行 EEPROM**: AT24C04 4Kbit EEPROM，IIC 接口；
- **UART 接口**: 2 通道 UART 接口，波特率高达 115200bps，具有 RS232 电平转换电路，可直接连接 PC 机；
- **LCD 接口**: 20 脚双排插针的 LCD 接口，可接 1600*1600 以下分辨率的单色或 256 色 STN/DSTN 型各种 LCD 屏，具有对比度调节电位器；
- **USB 接口模块**: 采用 NS 公司的高性能全速 USB 设备接口芯片 USBN9603。该器件符合 USB1.1 规范，2MB/s 的全速传输速率，共具有 6 个 endpoints
- **以太网接口模块**: 采用 10M 以太网接口控制器 RTL8019 的以太网卡模块。
- **IDE 接口**: 可挂硬盘或符合 IDE 接口标准的其他设备
- 20PIN 的 **JTAG 标准接口**
- 4 个用户可编程 **LED 指示灯**，1 个 **蜂鸣器**
- **复位**和外部 **中断**输入按键
- **3V 锂电池**，提供 CPU 内置 RTC 保持电源

二、ARMSYS44B0-C 嵌入式开发板配套光盘资料介绍

1.2.1 “BIOS” 文件夹

此文件夹中包含 ARMSYS44B0-C 开发板的启动工程文件 (ADS1.2 工程) 及源程序。

1.2.2 “source” 文件夹

Source 文件夹中包含 Ads_source 和 Sdt_source 两个文件夹。

其中 Ads_source 文件夹和 Sdt_source 文件夹中分别包含 12 个非常实用的应用实例源码（按照字母排序），说明如下：

- adctest** S3C44B0X 内置 A/D 转换器应用实例；
- einttest** S3C44B0X 外部中断实例；
- flashtest** flash 读写实例；
- Helloworld** Helloworld 程序；
- iictest** IIC 接口运用实例；
- iotest** I/O 应用实例——矩阵式键盘扫描；
- rtctest** S3C44B0X 内置 RTC 应用实例；
- Timertest** 定时器应用实例；
- uarttest** UART 串行口应用实例；
- ucos_ex1** uc/os-II 运用之一：多任务；
- ucos_ex2** uc/os-II 运用之二：任务间通讯；
- ucos_ex3** uc/os-II 运用之三：中断服务程序；

template.apj 工程模板

光盘资料中各项内容更为细致的说明，参考每个目录下的 Readme.txt 文件。

1.2.3 “uC/OS-II” 文件夹

此文件夹中包含有 uC/OS-II 在 ARMSYS44B0-C 上的移植源程序包。

1.2.4 “uClinux” 文件夹

此文件夹中包含有 uClinux 移植在 ARMSYS44B0-C 上的源码包和 Bootloader 源程序包。

1.2.5 “电路图” 文件夹

此文件夹中包含有 ARMSYS44B0-C 开发板的原理图（pdf 文件格式）。

1.2.6 “开发工具” 文件夹

此文件夹中包含有开发板使用到的工具软件或应用程序。其中包括：

- 1) 集成开发环境软件安装包（ADS1.2&SDT2.5）。
- 2) “ARMJtagDebugFinal” 文件夹中包含有 ARM7 和 ARM9 调试代理软件及驱动安装相关文件。
- 3) “programmer” 文件夹中包含有快速烧录工程程序 ADS_programmer 和 SDT_programmer。烧录用批处理文件及二进制（bin）代码文件。
- 4) “flashpgm2.2.4” 文件夹中包含有 “flashpgm” 快速烧录工具安装包及配置文件 44B0.OCD。
- 5) “usb” 文件夹中包含非常有用的 USB 代码下载工具（下载*.BIN 文件），也可以用来做 USB 下载实验（下载*.txt 文件）。
- 6) “下载器用 bin 文件” 文件中包括出厂时使用的 bin 文件。

1.2.7 “芯片资料” 文件夹

此文件夹中包含有开发板相关的主要芯片及部件的数据资料。

1.2.8 ARMSYS44B0-C 开发板使用说明书

此文件为 ARMSYS44B0-C 开发板使用说明书 (pdf 文件格式)。

1.2.9 uclinux 在 armsys 上的使用说明书

此文件为 uclinux 在 ARMSYS44B0-C 上的使用说明书 (pdf 文件格式)。

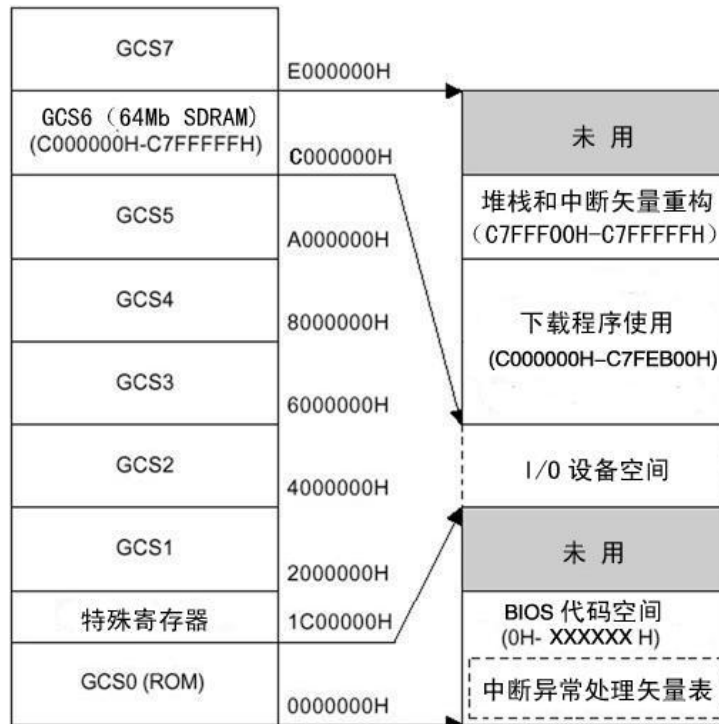
光盘资料中各项内容更为细致的说明, 参考每个目录下的 Readme.txt 文件。

三、ARMSys44B0-C 的启动程序和地址空间分配

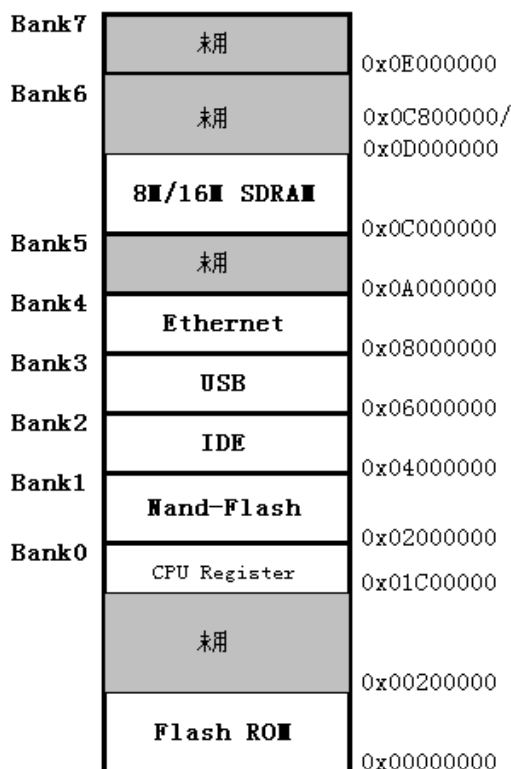
S3C44B0X 处理器可以对 8 个 bank 进行寻址, 每 bank 最大空间为 32M; 为了使处理器对各个设备的访问互不干扰, 我们将不同类的设备映射到不同的 bank 内。在 ARMSys 平台上, S3C44B0X 的 bank 空间的分配如图 3 所示。

图 3(a) 是 ARMSys44B0 的 bootloader 对程序空间和数据空间的分配。在程序空间 flash ROM 内 (在主板上对应 2M 字节大小的 SST39VF160 器件), 已经固化了一段启动系统并对系统进行初始化的程序——bootloader 程序。在图中, 可以看到 flashROM 存储器映射在了系统的 bank0 上, 也就是说, 系统上电时处理器即从 flashROM 的 0x00000000 地址处取得指令开始运行。这个地址上的 bootloader 程序完成了时钟设置初始化, 中断矢量的定义, 存储器的参数设置、堆栈地址定义等工作, 这些设置对于系统正常启动是非常重要的。由于 flash ROM 是非易失性的存储器, 因此程序就算掉电也不会丢失。但是如果由于某个误操作覆盖了 flashROM 中启动程序的内容, 系统就将无法正常启动, 这时就需要重新烧录 flashROM。烧录用的 44BAPP.bin 文件和烧录工具都可以在光盘中的“开发工具\programmer”目录下找到。

主板上的 SDRAM 器件映射在 bank6 上, 也就是 0xC000000 地址处。SDRAM 是易失性的可快速擦写的存储器, 因此它通常作为系统的数据空间。



(a) 程序和数据空间分配



(b) 外设空间分配

图3 ARMSys 对寻址空间的安排

如何设置系统的程序空间和数据空间呢？在 SDT/ADS 中通过对链接器的参数设置来完成。将只读基地址（程序空间的开始地址）-ro-base 设置为 0x00000000，读写基地址（数据空间的开始地址）-rw-base 设置为 0x0c000000 以上地址处，因此，这样就完成了程序空间和数据空间的设置（参考 7.2.3 节）。在图 3（a）中，0x0C000000~0x0C7FEB00 作为下载的程序使用，就是指在仿真调试程序时，将-ro-base 设置为 0x0c000000（在 bootloader 中，0x0C000000~0x0C008000 存放了二级中断矢量，因此-ro-base 必须设置在 0x0C008000 处）这样程序空间定位在 SDRAM 中，程序就可以随时下载、更新了。同时，数据空间往往定义在更高地址的位置上，例如将-rw-base 设置为 0x0c5f0000，只要为数据空间留下足够的空间，同时不与程序空间发生冲突即可。

图 3（b）中，显示了 ARMSys 的外部设备在处理器 bank 空间上的映射分布。Flash ROM 映射在 bank0 上，但它没有使用全部 bank0 的空间。Bank0 的高位空间地址，由 S3C44B0X 的内部特殊寄存器占用。Bank1, bank2, bank3, bank4, bank6 分别作为 Nand-flash, IDE, USB, 以太网接口设备, SDRAM 的映射空间。与 Bank x 对应的片选引脚 nGCS x 将作为外部设备的使能脚使用（参考电路原理图）。

四、ARMSYS44B0-C 开发板接口及其他

1.4.1 系统基本参数设定

- BANK0 总线宽度为 16 bit。
- Little Endian (小端) 模式。
- ARM 内核工作频率默认为 64MHz。

- 拨动开关 SW1 设定开发板供电方式，当拨动开关拨到 USB 端表示 USB 接口供电，当拨动开关拨到 EXT 时表示外部电源供电（采用 5V 1A 直流电源）

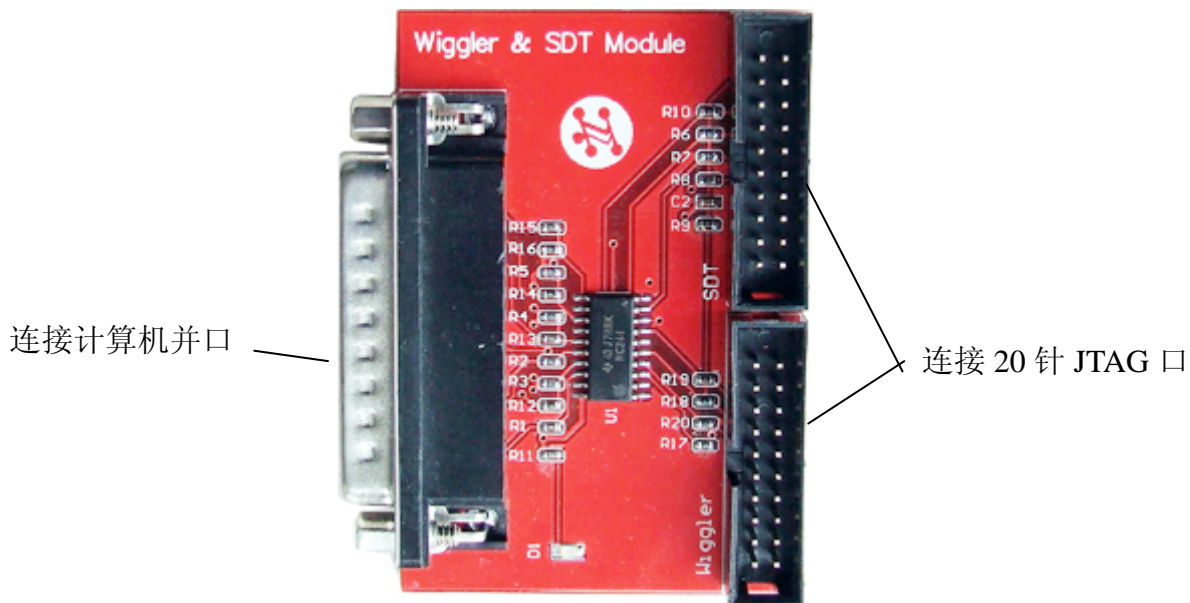
1.4.2 开发板中断分配

- EINT0: USB USBN9603
- EINT1: IDE/ATA
- EINT2: IDE
- EINT3: RTL8019
- EINT4: 按键 Exint4
- EINT5: 按键 Exint5
- EINT6: 按键 Exint6
- EINT7: 触摸屏 TSP(与按键 Exint7 共用)

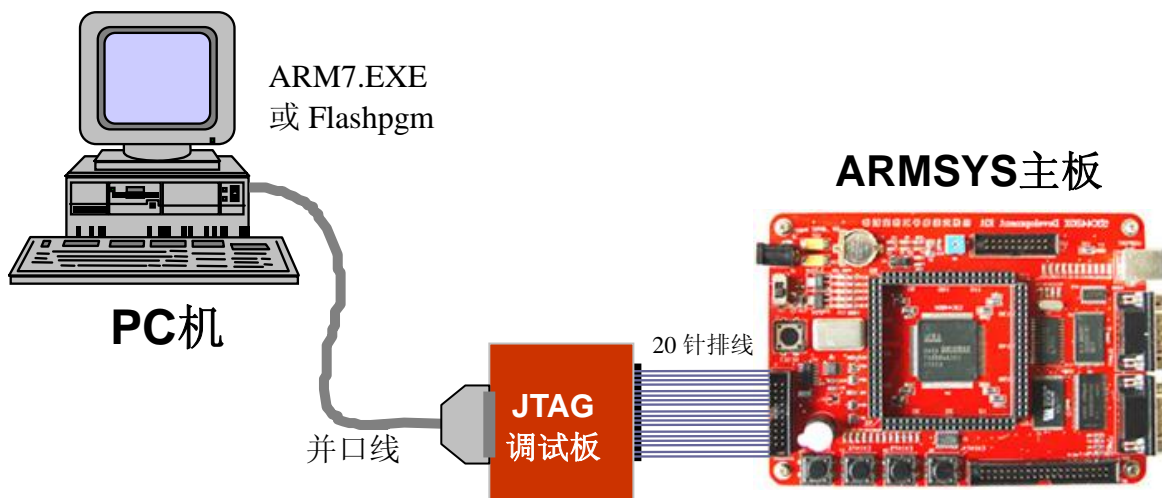
1.4.3 开发板对外接口与指示说明

- D8: 5V 直流电源指示灯。
- D51 和 D50: RTL8019 工作状态指示灯(未焊上)。
- D3: 由 I/O 口 GPE7 控制的指示灯。
- D4: 由 I/O 口 GPE6 控制的指示灯。
- D5: 由 I/O 口 GPE5 控制的指示灯。
- D6: 由 I/O 口 GPE4 控制的指示灯。
- D7: IDE/ATA 接口指示灯(未焊上)。
- 5V input: +9V 直流电源输入接口(内正外负)。
- SW1: usb 与外部电源切换开关。
- RESET: 开发板复位按键。
- JTAG20: 20 针 JTAG 标准接口。
- LCD&TSP: 26Pin 排线座, LCD 和触摸屏接口。
- USB-Slave: USB-B 型接口, USB DEVICE 接口。
- RJ45: 以太网口插座。
- UART0: 串口 0 接口座。
- UART1: 串口 1 接口座。
- EIDE Interface: 40Pin 排线座, IDE/ATA 硬盘接口。
- Exint4: 外部中断 4 按键。
- Exint5: 外部中断 5 按键。
- Exint6: 外部中断 6 按键。
- Exint7: 外部中断 7 按键。

五、JTAG 调试板的接口说明



JTAG 调试板的作用是程序下载和调试，一边采用并口线与 PC 机相连，一边 20 针排线与主板的连接，构成 ARMSYS 的调试系统。调试时的连接方式如下图所示：



使用 Wiggler 还是 SDT 口？

当用 ADS 或者 SDT 调试目标板时，可以采用 JTAG 调试板上 SDT 或 Wiggler 接口与目标板的 JTAG 接口相联接，只需要在调试代理软件上选择对应的接口即可。

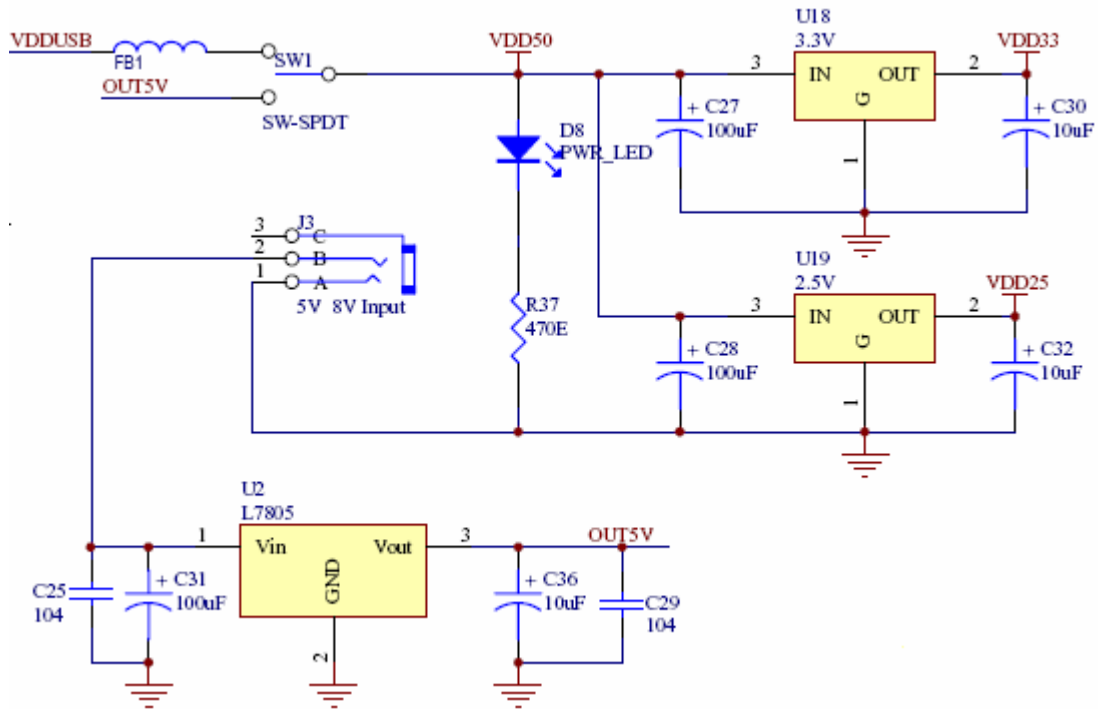
当用 Flashpgm 软件烧录程序时，必须采用 Wiggler 接口与目标板的 JTAG 接口相联接。

当用 Fluted 软件烧录程序时，必须采用 SDT 接口与目标板的 JTAG 接口相联接。

第二章 ARMSYS44B0-C 开发板电路描述

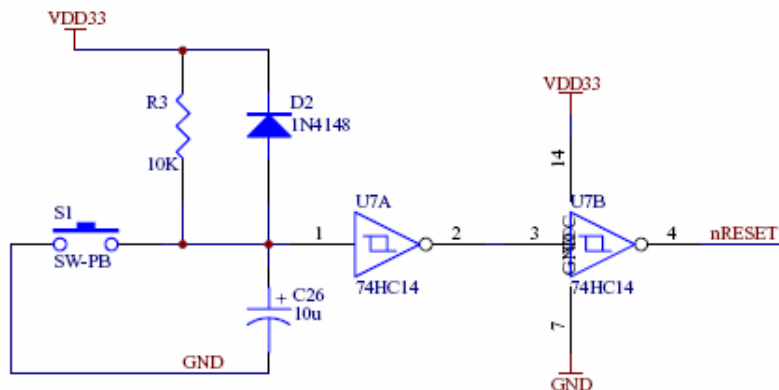
2.1 电源电路

开发板的供电可由 USB 接口或者外部直流稳压电源提供(L7805 未焊, 采用 5V 供电), 为了提高供电的稳定性和可靠性, 开发板采用集成线性稳压电源对 CPU 和外围电路供电。CPU 内核工作需要供给的 2.5V 电源采用 LM1117-2.5 产生。CPU 的 IO 和外设工作所需要的 3.3V 电源是采用 LM1117-3.3 产生, 相关电路如下:



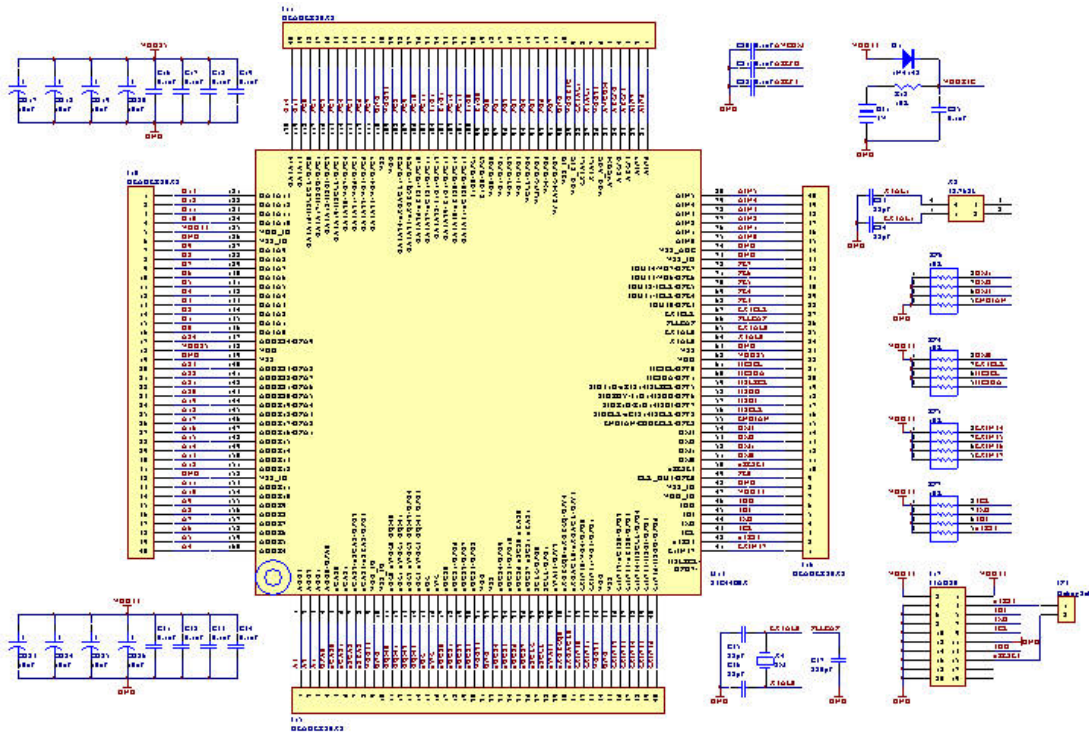
2.2 复位电路

采用较常用的 RC 加施密特反相器组成可靠的复位电路, 如下所示:



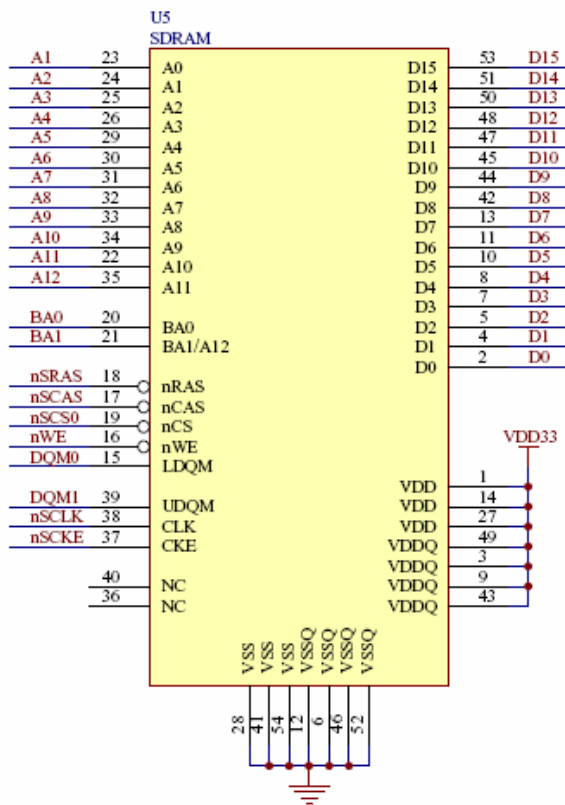
2.3 CPU 单元电路

CPU 单元电路采用开放式设计, 用户可任意扩展。该电路包括有时钟电路、RTC 实时时钟调电电源电路和 JTAG 接口电路。如下所示:



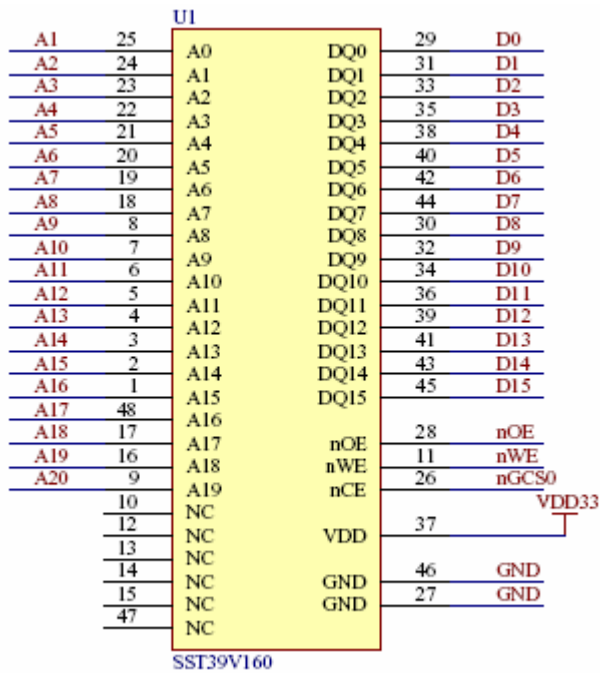
2.4 SDRAM 电路

SDRAM 电路采用了兼容性设计，考虑到用户的实际需要，支持 8M 字节、16M 字节和 32M 字节的 SDRAM，由三组六个电阻进行选择。如果是 8M 的 SDRAM（如 HY57V281620）则只焊接电阻 R5 和 R7，其他悬空不用。如果是 16M 的 SDRAM 则只焊接电阻 R6 和 R8，其他悬空不用。电路如下所示：



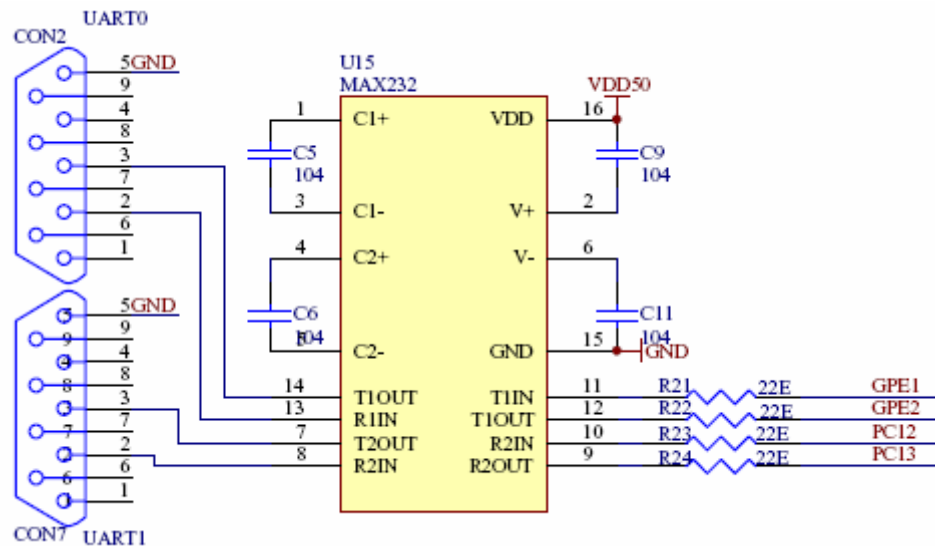
2.5 NorFlash 电路

NorFlash 电路采用 2M 字节 Flash。电路如下所示：



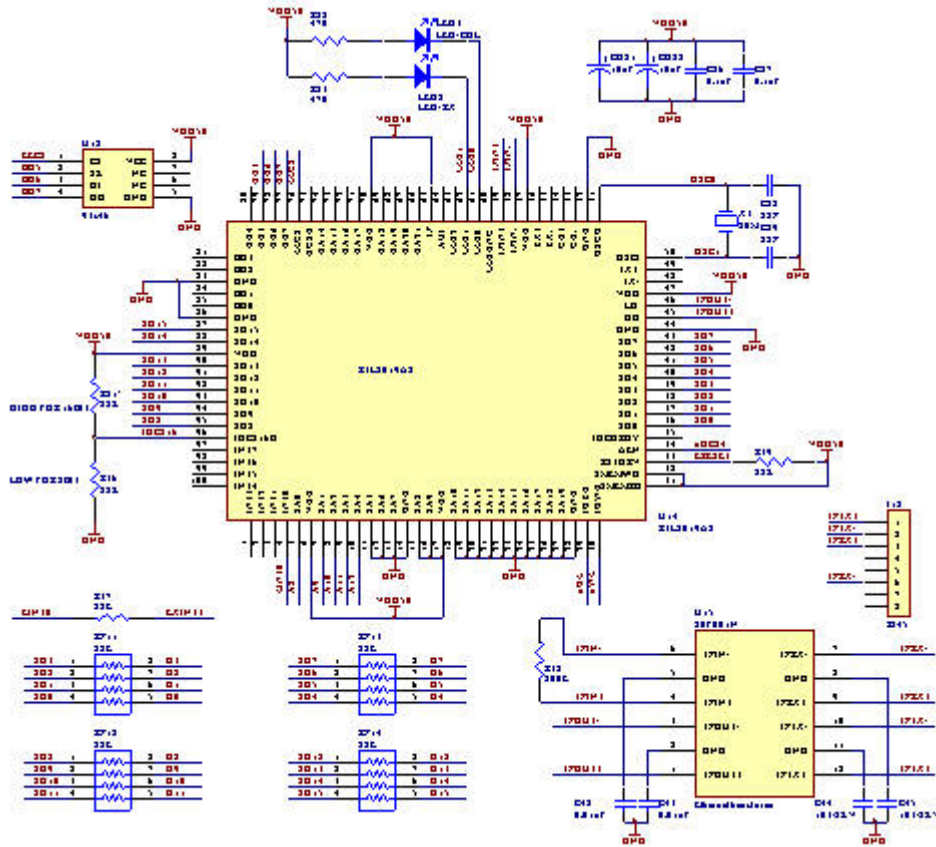
2.6 异步串行接口电路

该电路把 CPU 内部的两个串行口转换为标准的 RS232 接口。电路如下所示：



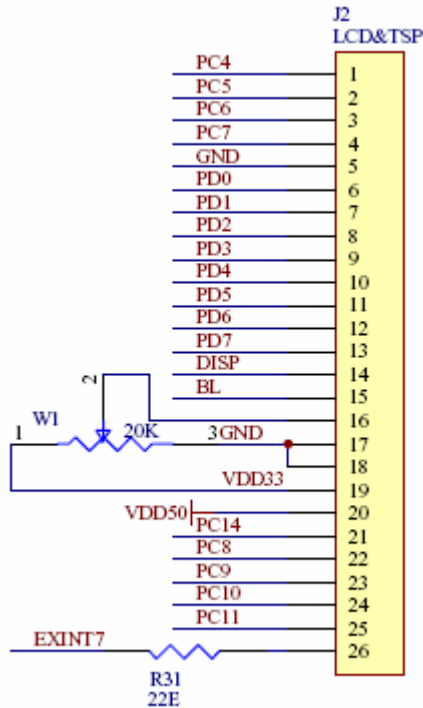
2.7 RTL8019 网络接口电路

该电路采用 10M 以太网接口控制器 RTL8019 为核心组成。与 CPU 的数据接口可 8 位或 16 位选择，默认数据位宽为 8 位。电路如下所示：



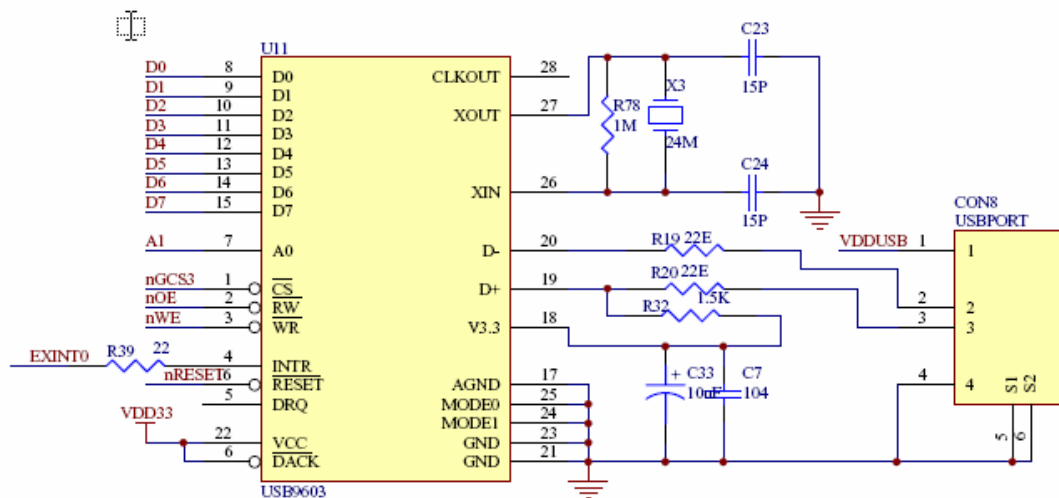
2.8 LCD 和触摸屏接口电路

该电路将 LCD 的接口和触摸屏的接口整合在一起，通过排线可连接至液晶电路板。电路如下所示：



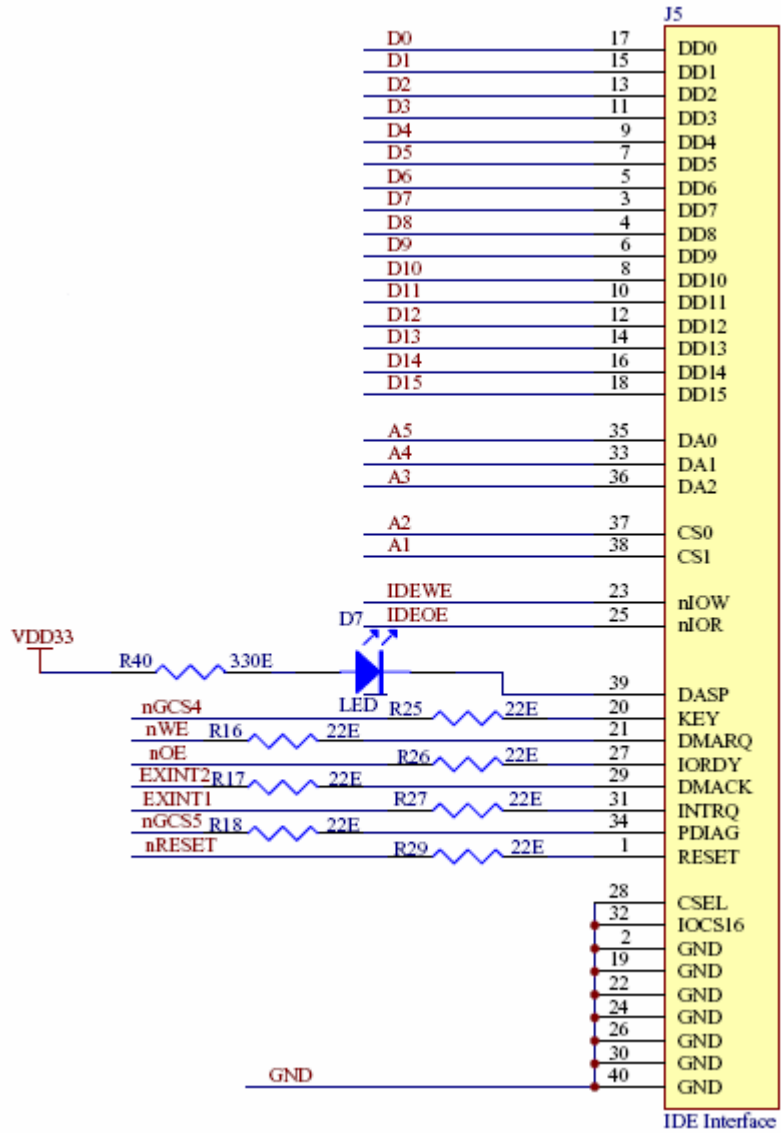
2.9 USB 接口电路

USB 接口电路采用 USBN9603 芯片，该接口也可为开发板供电。



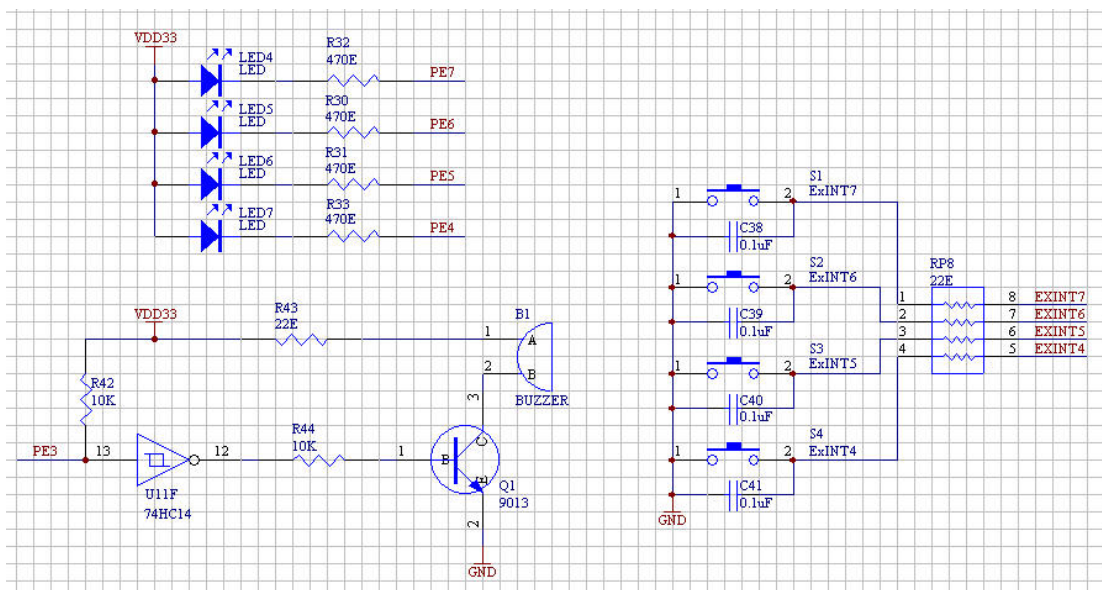
2.10 ATA (IDE) 接口电路

ATA 接口电路如下所示，可挂接 IDE 硬盘或连接 CF 卡接口电路。



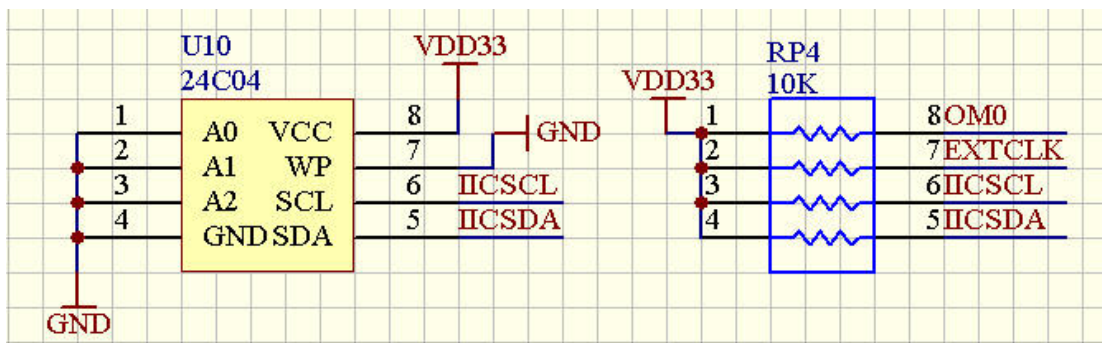
2.11 按键、蜂鸣器和 LED 电路

按键电路为 4 个外部中断的引出，分别对应外部中断 ExINT4、ExINT5、ExINT6 和 ExINT7。蜂鸣器和 LED 直接由 CPU 的 I/O 来控制。电路如下所示：



2.12 IIC 接口电路

IIC 电路采用存储器 AT24C04 芯片来实现，电路如下所示：



第三章 BOOTLOADER 使用说明

3.1 ARMSYS44B0-C 开发板工作环境的建立

3.1.1 开发板工作电源

将开发板 USB 接口通过 USB 电缆线与计算机的 USB 接口相接, 拨动开关 SW1 拨到“USB”位置, 可使开发板上电开始工作, 此时开发板上的红色电源指示 LED 点亮。

3.1.2 建立和设置超级终端

首先用开发板配套的串行电缆线连接计算机的串行口与开发板上的 UART0 串行口。然后在计算机上点击[开始 | 程序 | 附件 | 通讯 | 超级终端]如下图。

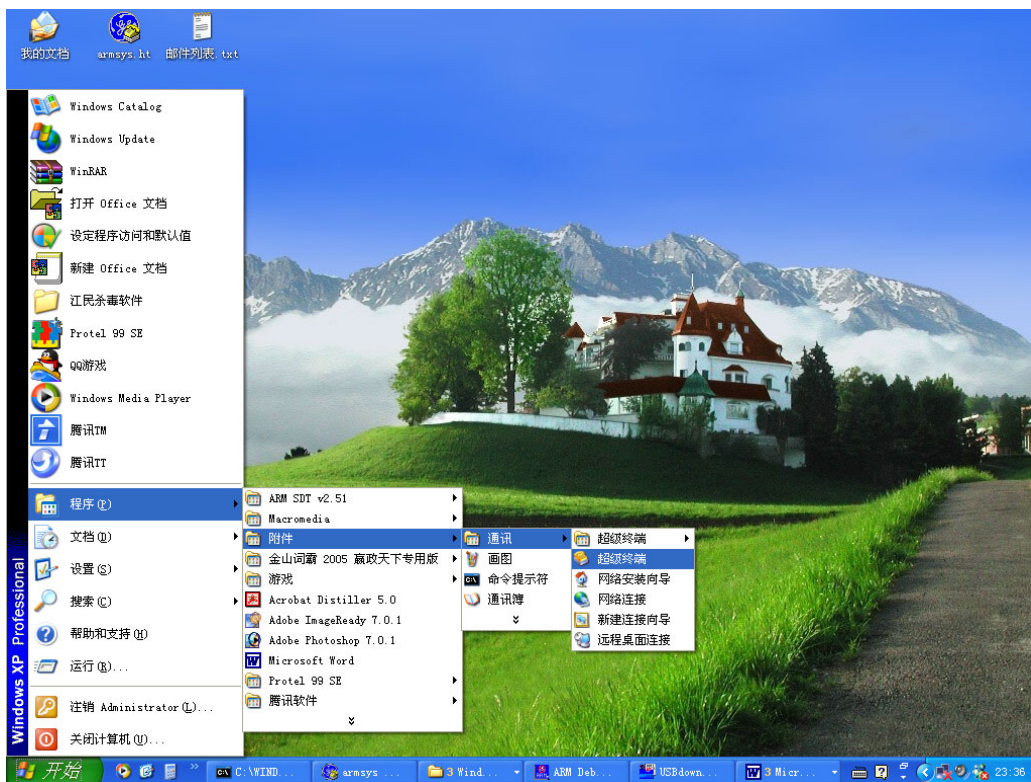


图 3-1 新建超级终端

新建一个超级终端项目, 将其命名为 ARMSYS, 点击“确定”, 弹出以下对话框:



图 3-2 超级终端属性

在“连接时使用”项中选择你所使用的串口号，点击“确定”按钮。按照下图配置该串口：



图 3-3 串口属性设置

点击确定，超级终端就配置好了。最后打开开发板电源即可看到如下启动信息：

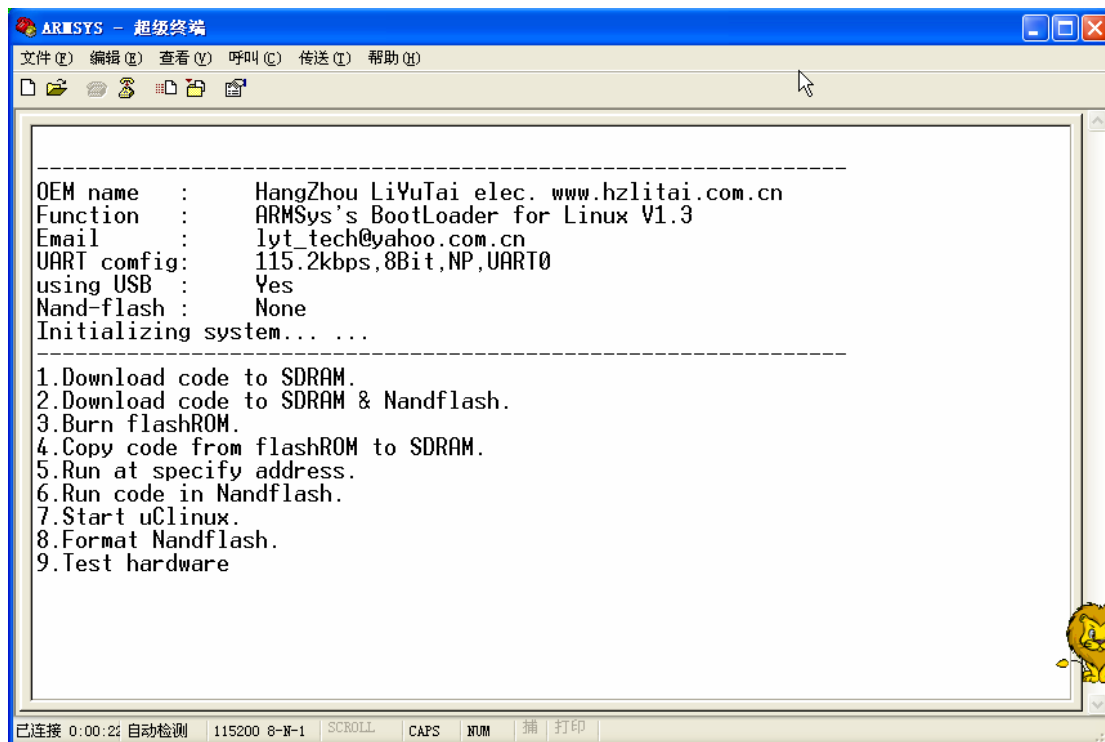


图 3-4 开发板启动信息

3.1.3 开发板接入以太网

用开发板配套的对等网线一端接计算机另一端接开发板，或者用 HUB 集线器引出的网线连接到开发板的 RJ45 接口座，打开电源计算机“本地连接”将处于“连接”状态。如下图所示：



3.2 BOOTLOADER 介绍

ARMSYS 开发板提供的 bootloader 是一个功能强大，使用方便的软件。它被烧写在 FLASH 中，上电即可运行。bootloader 可以方便的实现：

1. 开发板的硬件测试（详见第六章）
2. USB 下载
3. XMODEM 下载

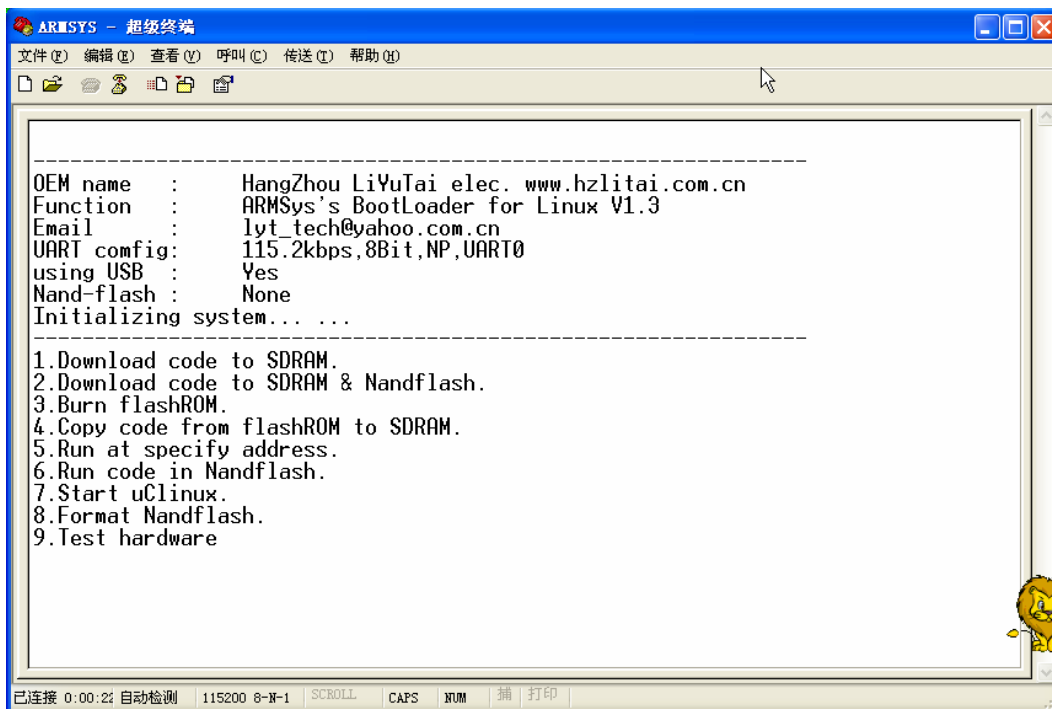
3.2.1 USB 口下载工具

USB 口下载工具能够将二进制代码快速下载到 ARMSys 上并运行。用户能够立即观察到程序运行的效果。USB 下载器工具放在光盘的开发工具\usb 目录下。安装驱动程序步骤如下：

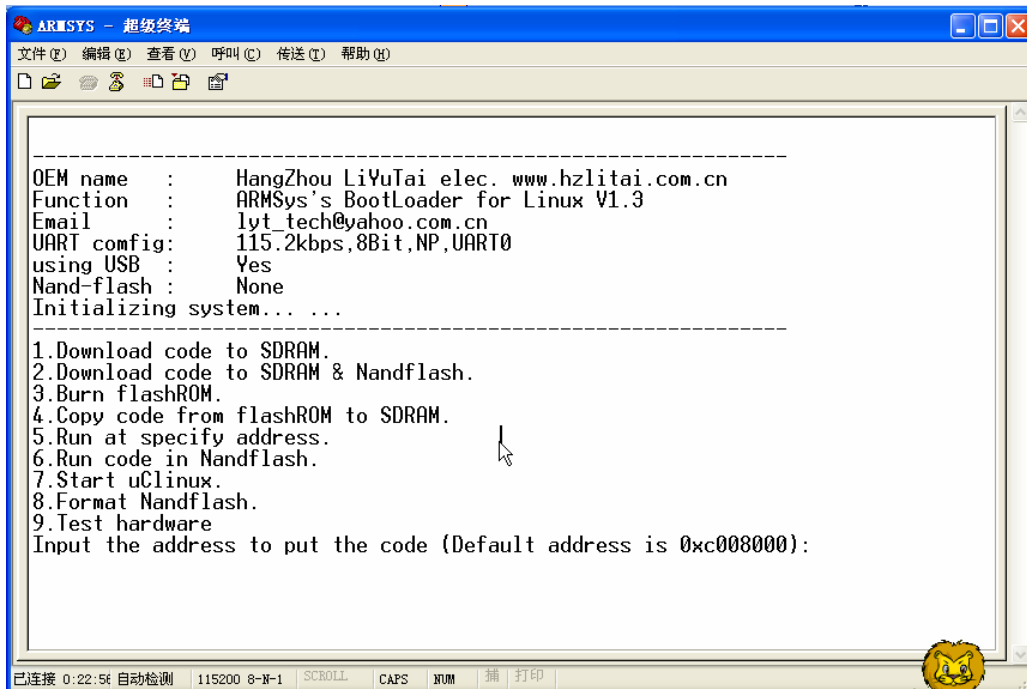
- 步骤 1，将 usbininstall 目录，整个拷贝到 C:\ 下（注意必须是 C:\）；
- 步骤 2，双击 C:\usbininstall\install.bat，等待一会儿，直到整个批处理程序执行完毕（注

意：弹出的控制台窗口自动消失之前不要手动关闭！在 98 下标题栏中最后显示“已完成”）；

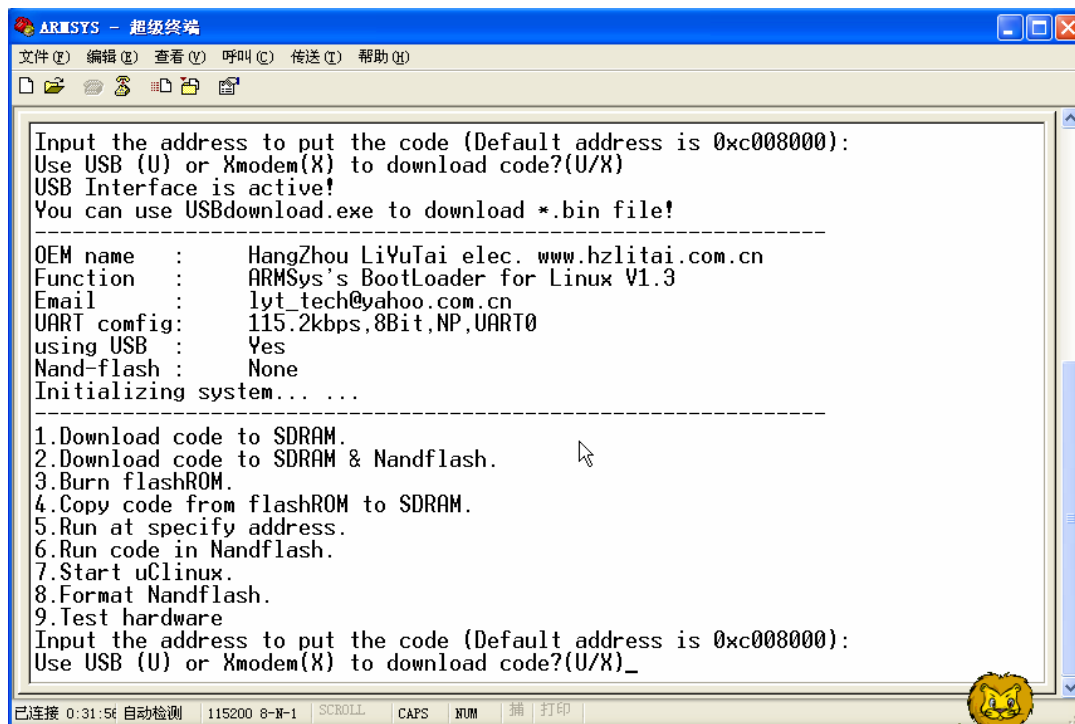
- 步骤 3，用 USB 口线连接 ARMSys 核心板和 PC 机 USB 口，打开电源；开发板正确复位后超级终端显示如下：



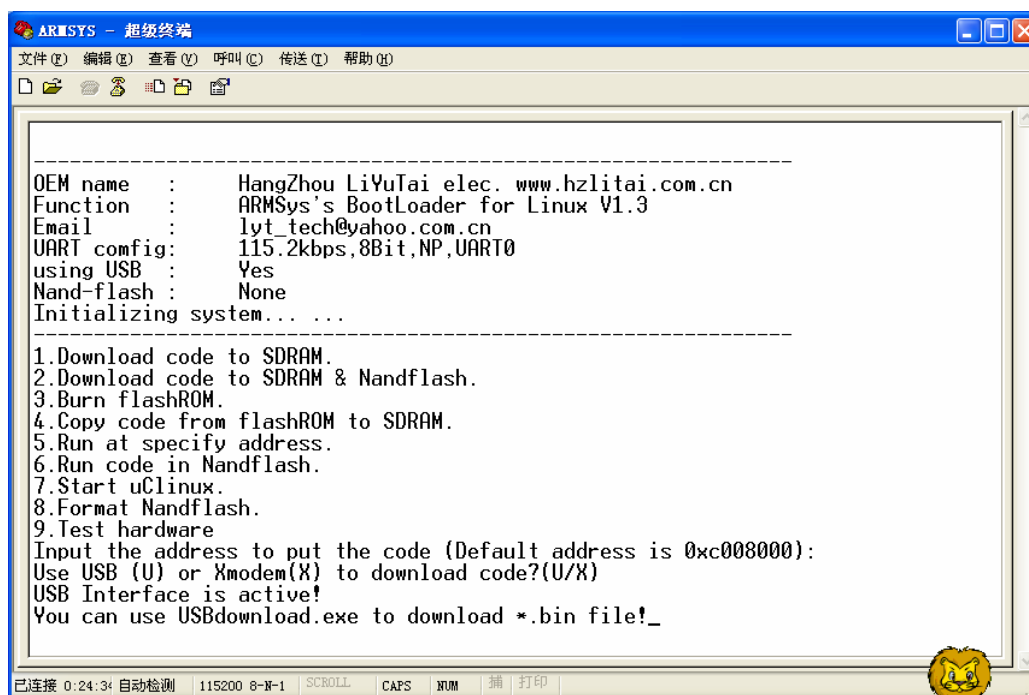
按下键盘上的‘1’提示你输入地址，如下图所示：



这个地址就是你下载到 SDRAM 中的地址，也是链接器代码定位的地址，直接敲回车则使用括号中的缺省地址。输入地址并回车后出现：

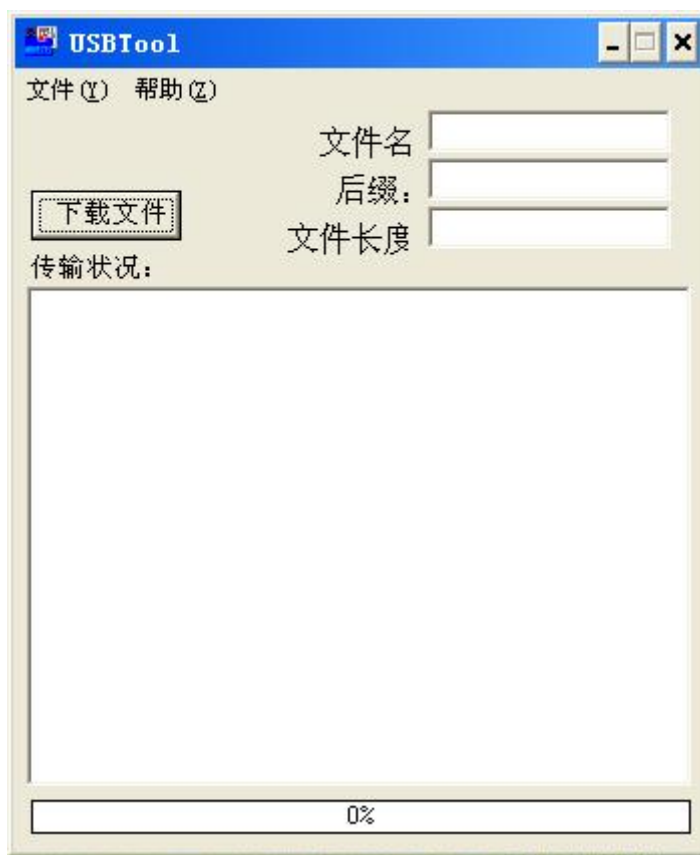


按下键盘上的‘U’或者按回车键，将采用usb下载方式，下图所示：



这样 USB 接口就成功地被使能了；这时打开“设备管理器”会看到 Jungo 下面自动识别出了 USB Tool Device 设备，说明驱动安装成功了。

- 步骤 4，双击运行 C:\usbinstall\Usbdownload.exe，出现如下对话框界面：



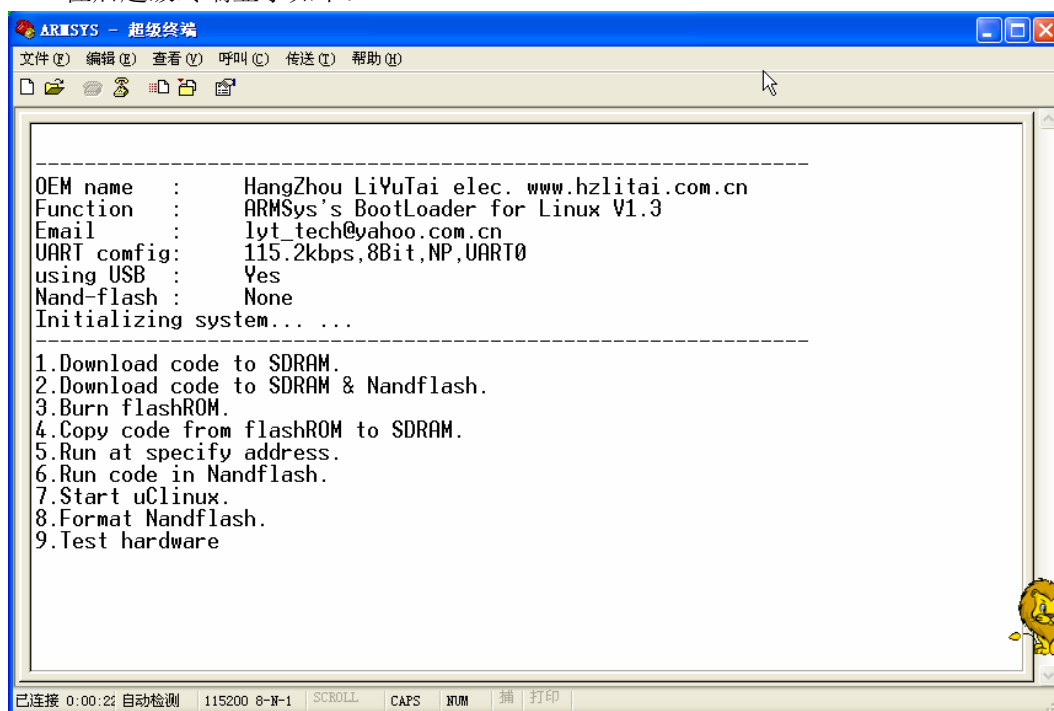
- 步骤 6, 点击 USBTool 工具的菜单项[文件 | 打开文件], 然后在“文件类型”中下拉选择*.bin 类型文件, 选择好要下载的文件后 (例如刚刚产生的 `helloworld.bin` 文件), 点击“打开”。这时可以看到, 对话框的文件名、后缀、文件长度编辑框中即出现所选文件的相关信息。点击“下载文件”按钮, 二进制文件被下载到 ARMSys 中。等待一会儿, 系统自动开始运行刚刚下载的程序。通过超级终端上显示的内容, 可以看到程序的运行情况。



3.2.2 采用 XMODEM 下载（通过串行口）

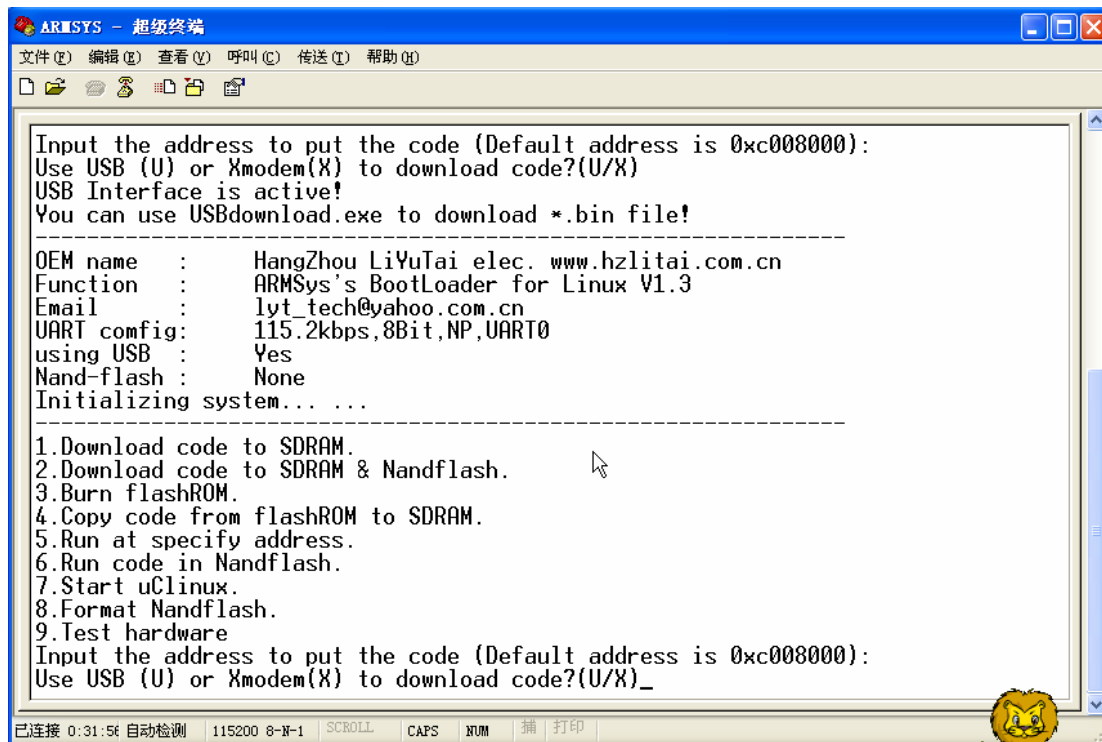
采用 xmodem 协议下载 bin 文件的方法如下步骤：

- 步骤 1、用串行电缆线连接 ARMSys 核心板和 PC 机串行口，打开电源；开发板正确复位后超级终端显示如下：



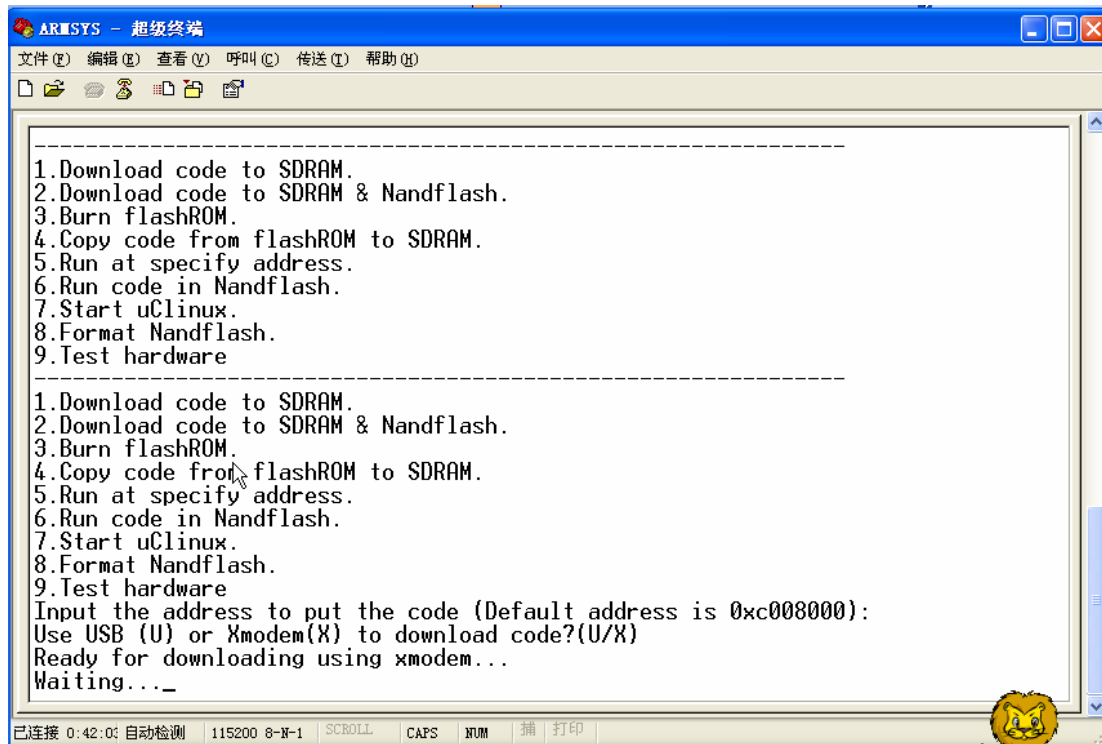
这个地址就是你下载到 SDRAM 中的地址，也是链接器代码定位的地址，直接敲回车则使

用括号中的缺省地址。输入地址并回车后出现：



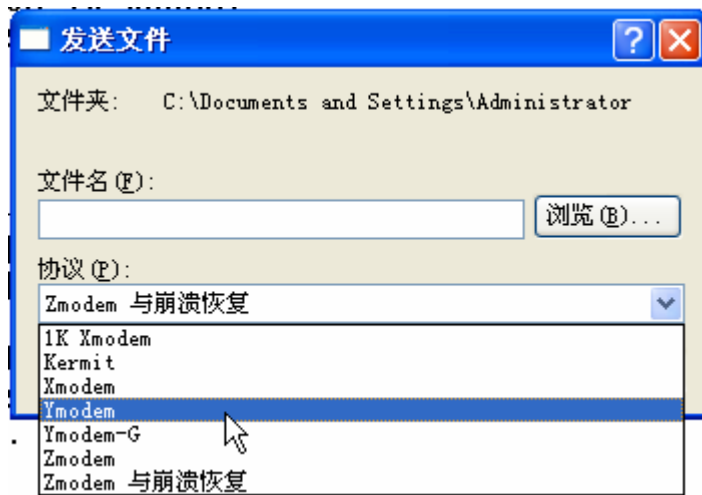
```
ARMSYS - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
[Icons]
-----
Input the address to put the code (Default address is 0xc008000):
Use USB (U) or Xmodem(X) to download code?(U/X)
USB Interface is active!
You can use USBdownload.exe to download *.bin file!
-----
OEM name   :   HangZhou LiYuTai elec. www.hzlitai.com.cn
Function   :   ARMSys's BootLoader for Linux V1.3
Email      :   lyt_tech@yahoo.com.cn
UART config: 115.2kbps,8Bit,NP,UART0
using USB  :   Yes
Nand-flash :   None
Initializing system... ..
-----
1.Download code to SDRAM.
2.Download code to SDRAM & Nandflash.
3.Burn flashROM.
4.Copy code from flashROM to SDRAM.
5.Run at specify address.
6.Run code in Nandflash.
7.Start uClinux.
8.Format Nandflash.
9.Test hardware
Input the address to put the code (Default address is 0xc008000):
Use USB (U) or Xmodem(X) to download code?(U/X)_
-----
已连接 0:31:56 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印
```

按下键盘上的‘X’或者按回车键，将采用usb下载方式，下图所示：



```
ARMSYS - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
[Icons]
-----
1.Download code to SDRAM.
2.Download code to SDRAM & Nandflash.
3.Burn flashROM.
4.Copy code from flashROM to SDRAM.
5.Run at specify address.
6.Run code in Nandflash.
7.Start uClinux.
8.Format Nandflash.
9.Test hardware
-----
1.Download code to SDRAM.
2.Download code to SDRAM & Nandflash.
3.Burn flashROM.
4.Copy code from flashROM to SDRAM.
5.Run at specify address.
6.Run code in Nandflash.
7.Start uClinux.
8.Format Nandflash.
9.Test hardware
Input the address to put the code (Default address is 0xc008000):
Use USB (U) or Xmodem(X) to download code?(U/X)
Ready for downloading using xmodem...
Waiting..._
-----
已连接 0:42:03 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印
```

步骤 2：点击超级终端的菜单【传送】—【发送文件】，选择协议为“xmodem”，如下图所示：



步骤 3: 点击“浏览”，选择下载文件即可。

第四章 ARM 开发环境介绍

4.1 ADS1.2 集成开发环境简介与安装

ADS1.2 是一个使用方便的集成开发环境，全称是 ARM Developer Suite v1.2。它是由 ARM 公司提供的专门用于 ARM 相关应用开发和调试的综合性软件。在功能和易用性上比较 SDT 都有提高，是一款功能强大又易于使用的开发工具。下面就我们对 ADS1.2 进行一些简要的介绍。

ADS 囊括了一系列的应用，并有相关的文档和实例的支持。使用者可以用它来编写和调试各种基于 ARM 家族 RISC 处理器的应用。你可以用 ADS 来开发、编译、调试采用包括 C、C++和 ARM 汇编语言编写的程序。

ADS 主要由以下部件构成：

- ◇ 命令行开发工具；
- ◇ 图形界面开发工具；
- ◇ 各种辅助工具；
- ◇ 支持软件。

其中重点介绍一下图形界面开发工具。

◇ **AXD** 提供给基于 Windows 和 UNIX 使用的 ARM 调试器。它提供了一个完全的 Windows 和 UNIX 环境来调试你的 C、C++和汇编语言级的代码。

◇ **Code Warrior IDE** 提供基于 Windows 使用的工程管理工具。它的使用使源码文件的管理和编译工程变得非常方便。但 CodeWarrior IDE 在 UNIX 下不能使用。

运行开发板配套光盘中的开发工具下的 ADS1.2 文件夹下面的可执行安装程序 setup.exe，然后就像安装其他 Windows 应用程序一样一步一步向下执行，最后装好 license.dat 文件，编译调试环境就安装好了。

4.2 JTAG 调试代理软件的安装与使用

4.2.1 首次使用时先安装驱动（以后不用再安装），将配套光盘中 [开发工具 \ARMJtagDebugFinal] 文件夹拷贝到硬盘一个目录下，去掉只读属性（**注意：ARMJtagDebugFinal 所在的路径不要带有中文字符或特殊字符，或者处于太深的目录下**）。然后进入 ARMJtagDebugFinal 目录，双击下的“安装驱动.exe”安装好并口驱动和 OCX。安装过程如图所示：





出现以上窗口后就说明 ARM 调试代理软件已经安装成功，点击[确定][Quit]退出即可。

4.2.2 安装完毕后即可运行 ARM7.exe 调试 ARM7 系统。

4.2.3 如果在以后的使用过程中发现程序无法启动，重新执行“安装驱动.exe”即可。

4.2.4 用 JTAG 仿真调试之前，首先将开发板电源接上，JTAG 复位同步跳线帽插上，用开发板配套的并口电缆连接计算机和 JTAG 仿真小板，再用 20 针排线连接 JTAG 仿真小板到开发板的 20 针 JTAG 接口座。

硬件连接好以后，打开开发板电源，运行 ARM7.exe 调试代理软件，可以看到以下结果，说明 JTAG 调试代理软件运行成功。



4.3 使用 CodeWarrior 建立工程并进行编译

首先我们学习如何使用 ADS 中的 CodeWarrior —— 项目管理器来管理源代码。一个嵌入式系统项目通常是由多个文件构成的，这其中包括用不同的语言（如汇编或 C）、不同的类型（源文件，或库文件）的文件。CodeWarrior 通过“工程（Project）”来管理一个项

目相关的所有文件。因此，在我们正确编译这个项目代码以前，首先要建立“工程”，并加入必要的源文件、库文件等。

4.3.1 调入模板或重新建立项目

我们通常采用工程模板来建立新的工程，工程模板已经针对目标系统对编译选项进行了设置，为避免重复设置，我们提供了一个在 ARMSYS44B0-C 上使用的通用工程模板——template.mcp。

点击 CodeWarrior 菜单[File | open...], 找到 source\template.mcp (配套光盘中的源程序要拷贝到计算机的硬盘上，并去掉只读属性)，选中并打开。

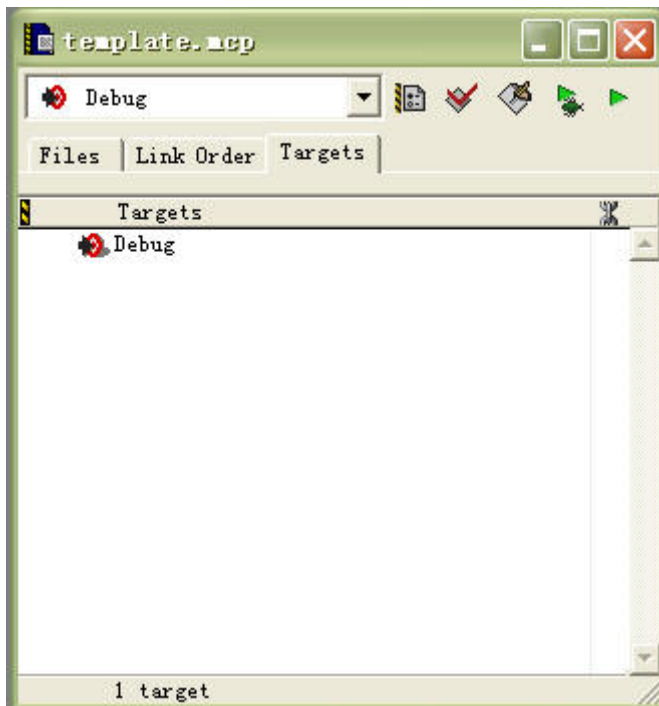



图 4-1 打开工程模板

点击[File | Save as...], 将它另存为: source\Myhelloworld.mcp, (或者是自定义的其它目录)\Myhelloworld.mcp。然后, 关闭当前的工程, 重新调入 Myhelloworld.mcp, 就可以向工程中添加文件了。

如果你不想利用模板, 也可以按照以下步骤来新建一个工程:

- (1) 选择 File 菜单下的 new 选项, 或直接单击 , 出现以下对话框:

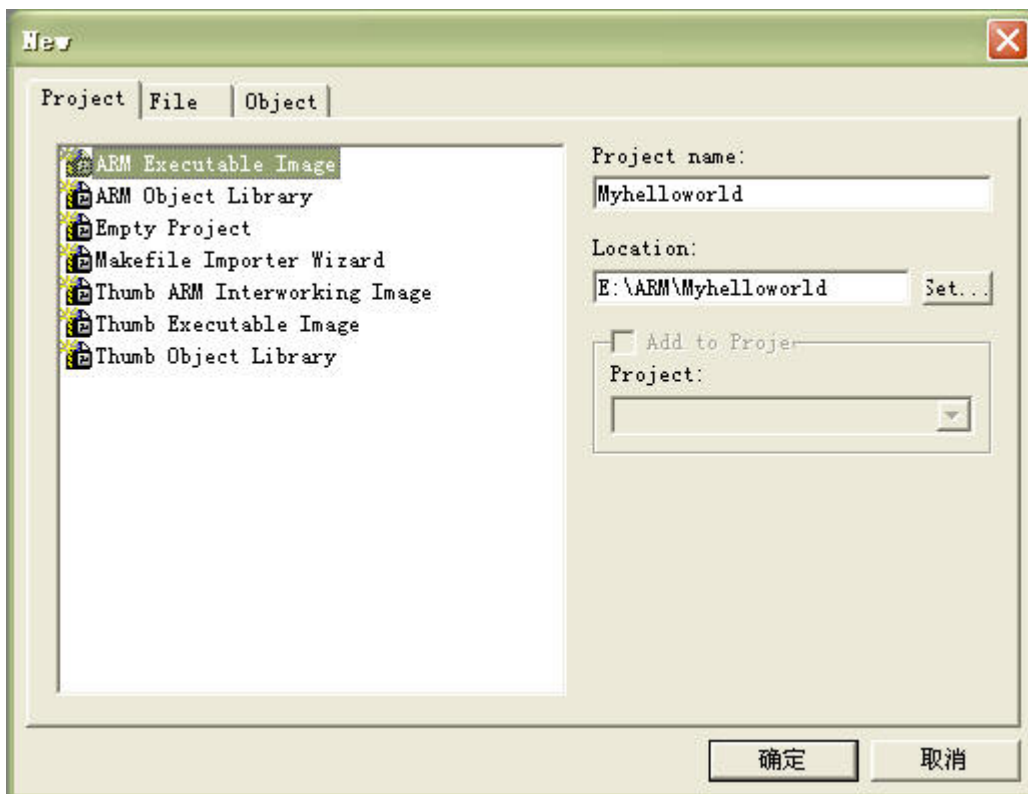


图 4-2 新建工程对话框

(2) 选中“ARM Executable Image”选项，在右边的编辑框中输入工程名（例如 Myhelloworld），在下面的 Location 栏中，点击“Set...”，选择放置工程的路径。ADS1.20 不支持中文的目录名字，所以新建工程的文件夹向上一直到根目录的所有文件夹的名字都是英文的。

(3) 点击[确定]后工程被建立。

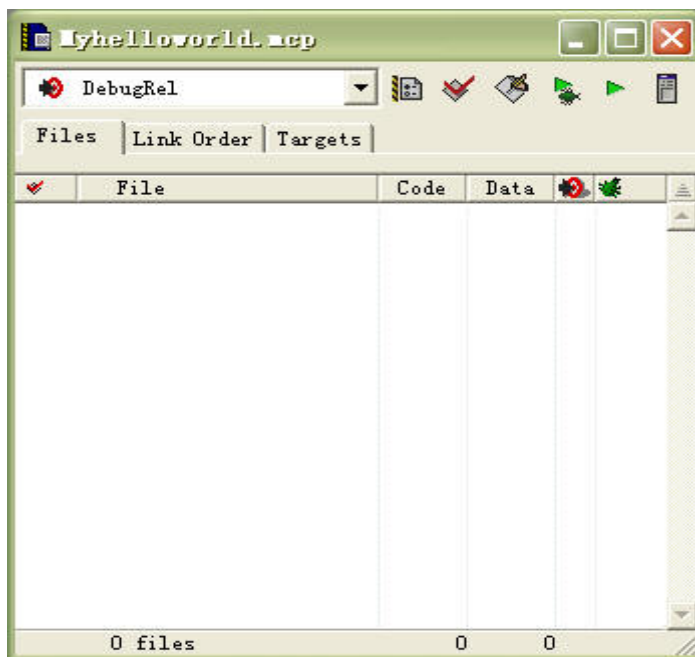


图 4-3 新建工程

但这样的工程还并不能正确地编译，还需要对工程的编译选项进行适当配置。为了设置方便，先点选 Targets 页面，选中 DebugRel 和 Release 变量，按下 Del 键将它们删除，

仅留下供调试使用的 Debug 变量。点击菜单 [Edit | Debug Setting...], 弹出配置对话框:

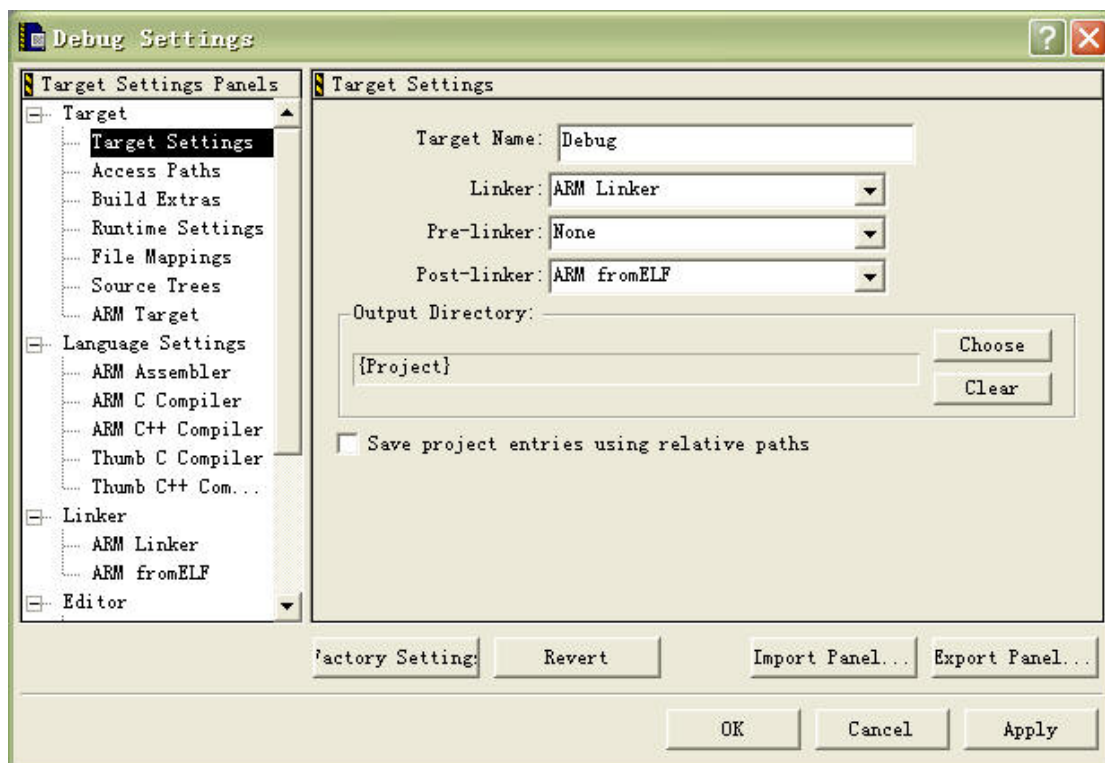


图 4-4 工程配置对话框——目标设置

首先选中 Target Setting，将其中的 Post-linker 设置为 ARM fromELF，使得工程在链接后再通过 fromELF 产生二进制代码。

然后选中 ARM Linker，对链接器进行设置:

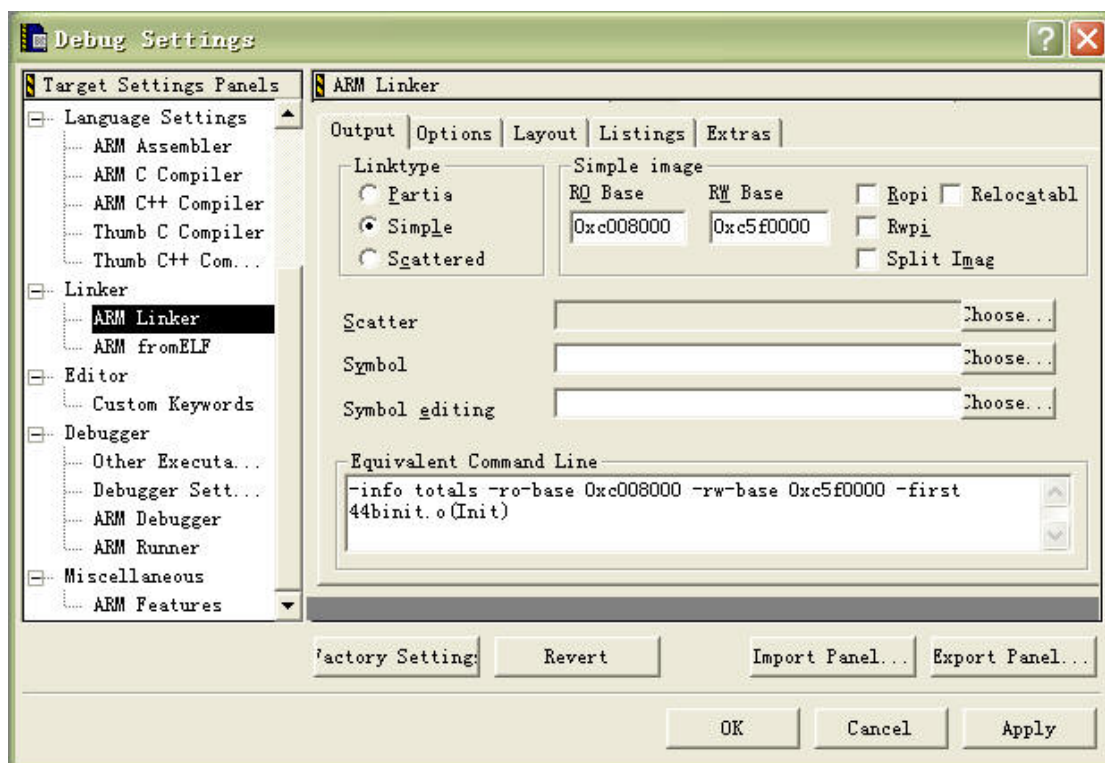


图 4-5 (a) ARM Linker 的设置

注意,在调试时,-RO-Base 的设置应当大于 0xC000000。我们为了与 uLinux 的 memory map 保持一致,采用了 0xC008000 这个地址。

选取 Layout 页面进行设置:

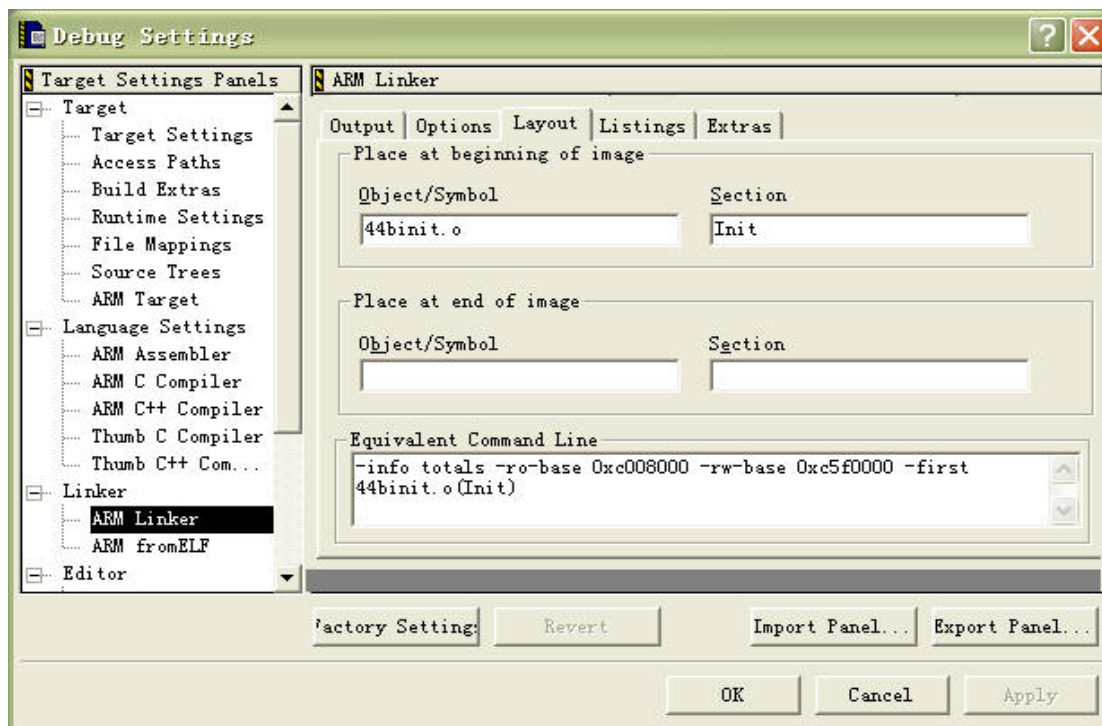


图 4-5 (b) ARM Linker 的设置

将 44binit.o 放在映象文件的最前面,它的区域名是 Init。

最后,如果你希望编译的最后生成二进制文件,就要设置 ARM fromELF:

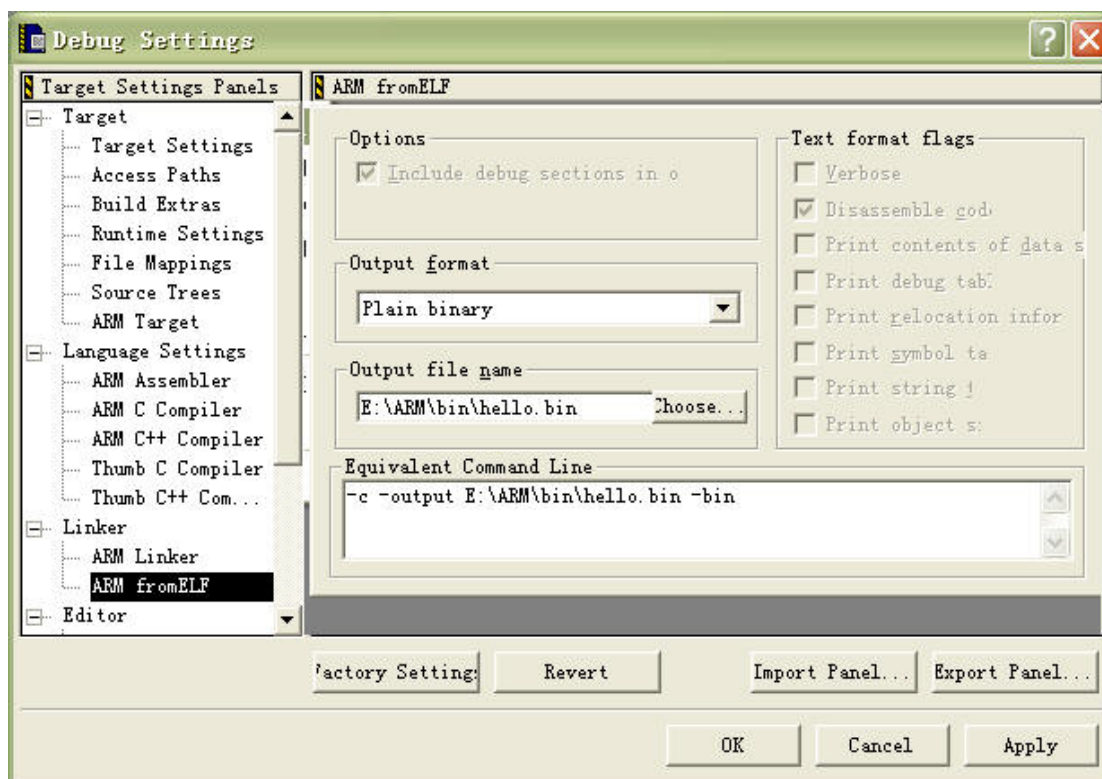


图 4-6 ARM fromELF 的设置

在 Output format 栏中选择 Plain binary，在 Output file name 栏中，点击“Choose...”选择你要输出的二进制文件的文件名和路径。

这样，对于 Debug 变量的基本设置都完成了。点击“OK”键退出。

4.3.2 在工程中添加源文件

在图 4-2 的对话框中，点选 File 页面，选中 Text File，并设置好文件名和路径，点击确定，CodeWarrior 就会为你新建一个源文件，并可以开始编辑该空文件。CodeWarrior 与 SDT 中的 APM 不同，它具有一个很不错的源代码编辑器，因此，大多数时候，我们可以直接采用它的代码编辑器来编写好程序，然后再添加到工程中。

添加源文件的步骤如下：例如添加 main.c 文件，在图 4-3 窗口中点选 Files 页面，在空白处单击鼠标右键，点选“Add Files”项，从目录中选取 main.c 文件（Myhelloworld\main.c），点击“打开”，main.c 文件就被加入了工程中。

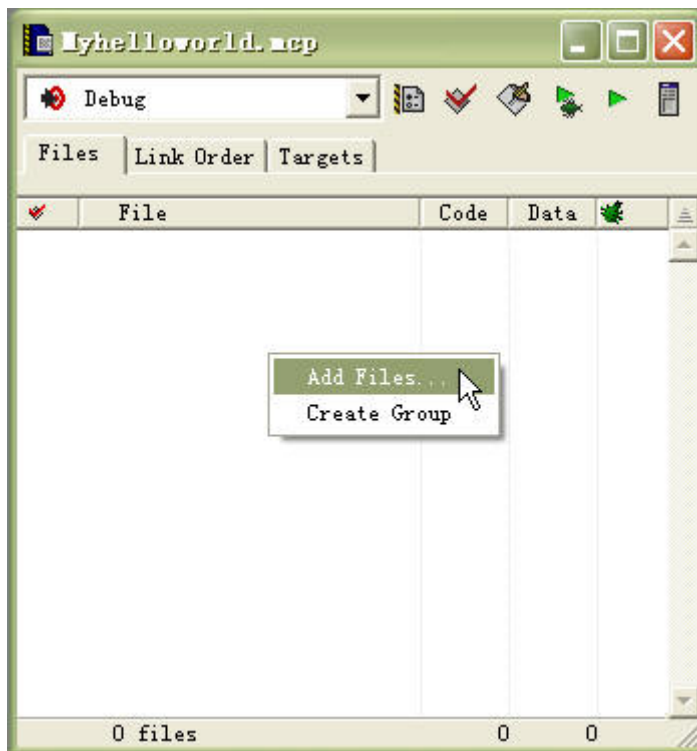


图 4-7 添加源文件

用同样的方法，将 Myhelloworld\下所有的*.C 和*.S 源文件文件都添加到 source 中去(包括 Target 目录下的源文件)。

Target 目录下还有一个 44b1ib.c 文件，这是一个库文件，其中提供了一些常用函数的定义，这些函数在 44b1ib.h 进行了声明。这个文件也必须添加到工程中。同样的方法，按鼠标右键，Add files ...，将 44b1ib.c 文件添加到工程中。所有必须的文件添加完成后如图 4-8 所示。

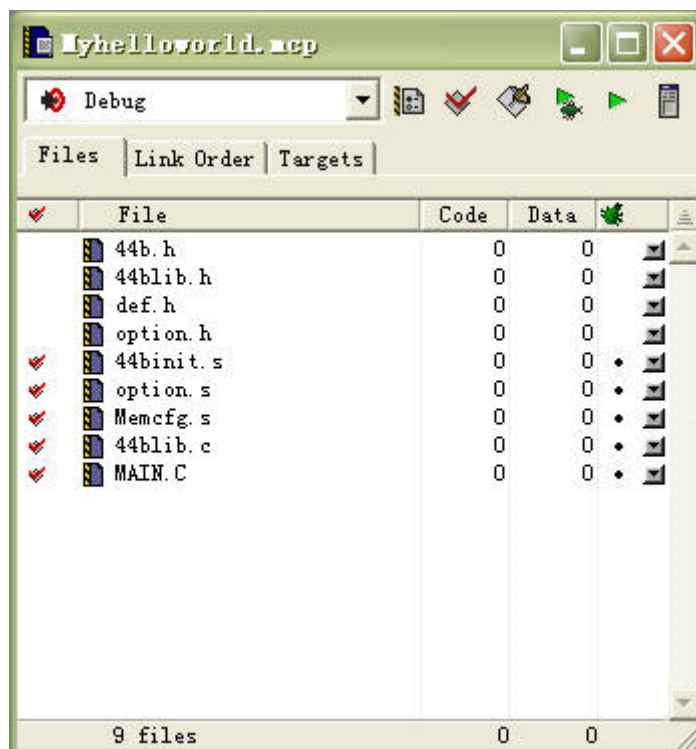





图 4-8 源文件添加完成

4.3.3 工程进行编译和连接

注意到在上图 4-8 中新加入的文件前面有个红色的“钩”，说明这个文件还没有被编译过。在进行编译之前，你必须正确设置该工程的工具配置选项。如果前面采用的是直接调入工程模板，有些选项已经在模板中保存了下来，可以不再进行设置。如果是新建工程，则必须按照上文中所述的步骤进行设置。

- 选中所有的文件，点击  图标进行文件数据同步；
- 然后点击  图标，对文件进行编译（compile）；
- 点击  按钮，对工程进行 Make，Make 的行为包括以下过程：
 - 编译和汇编源程序文件，产生*.o 对象文件；
 - 链接对象文件和库产生可执行映像文件；
 - 产生二进制代码。

Make 之后将弹出“Errors & Warnings”对话框，来报告出错和警告情况。编译成功后的显示如下。注意到左上角标示的错误和警告数目都是“0”，如图所示：

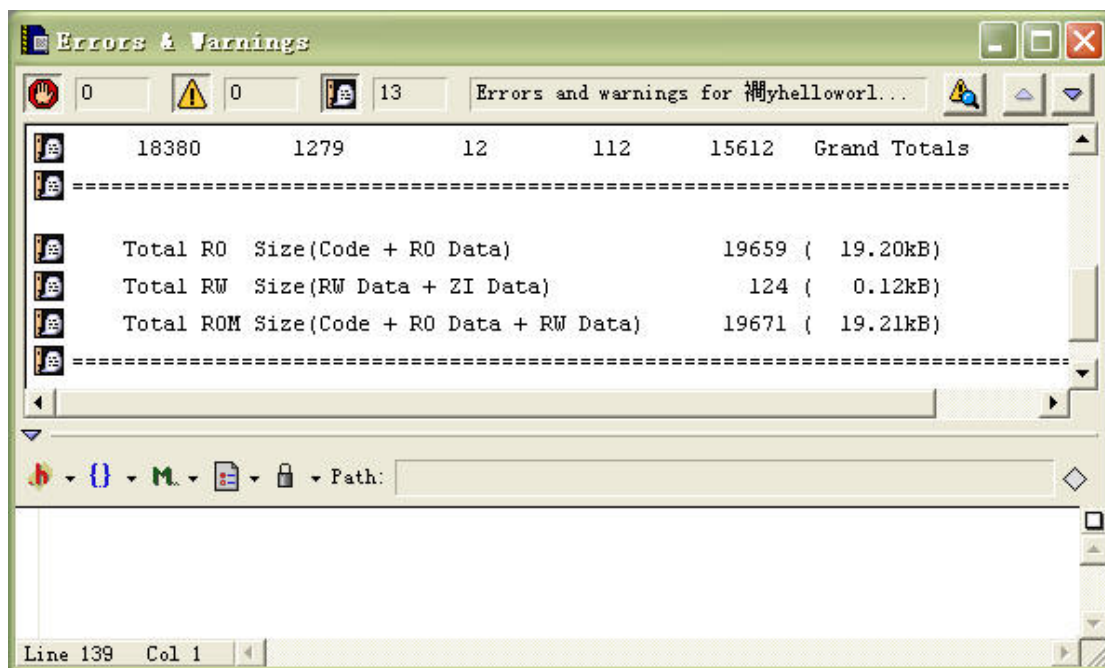


图 4-9 编译后的结果

Make 结束后产生了可执行映像文件 Myhelloworld.axf 文件，这个文件可以载入 AXD 进行仿真调试了。并且还通过 fromelf 工具将 ELF 文件转换为二进制格式文件 hello.bin。它可以用来最终固化到 flash ROM 中（但链接选项中的 -ro-base 要修改为 0x0），也可以下载到 -ro-base 地址中运行。

4.4 使用 AXD 进行仿真调试

4.4.1 调试前的准备


在调试之前，我们先用并口电缆将计算机并口和 JTAG 仿真小板连接起来，并把 JTAG 仿真小板和开发板的 JTAG 接口用 20 芯排线连接，用串口线将计算机串口和主板的 UART0 口连接起来。然后检查 JTAG 同步复位跳线帽是否插上，所有连线连接正确、接触良好后就可以拨动电源开关上电了。

在打开电源之前，要先打开按上文介绍建立的“超级终端”才能观察到开发板的启动输出信息。如图 3-4 中所示。

电源打开之后，可以听到主板发出一声蜂鸣器的“嘀——”声，看到绿色发光管点亮后熄灭，这说明开发板启动正常。此时 JTAG 仿真小板上只有红色电源指示发光管点亮，说明并口已经连接好了。

在进行调试之前，要先建立好 AXD 与目标系统之间的通讯。如果采用简易 JTAG 调试器进行调试，则首先要运行 JTAG 调试代理软件。**注意，在 AXD 调试器在线仿真期间，不要关闭 JTAG 调试代理软件！**如果采用 Multi-ICE 仿真器来调试，则首先要运行 Multi-ICE Server（具体请查看仿真器的使用说明）。

4.4.2 AXD 调试器的设置

在 CodeWarrior 编译环境中，工程经过编译成功，产生了 *.axf 文件之后，就可以进行调试了。点击  按钮，进入了 AXD 视窗界面。点击菜单项 [Option | Configure Target...]，对调试目标进行配置：

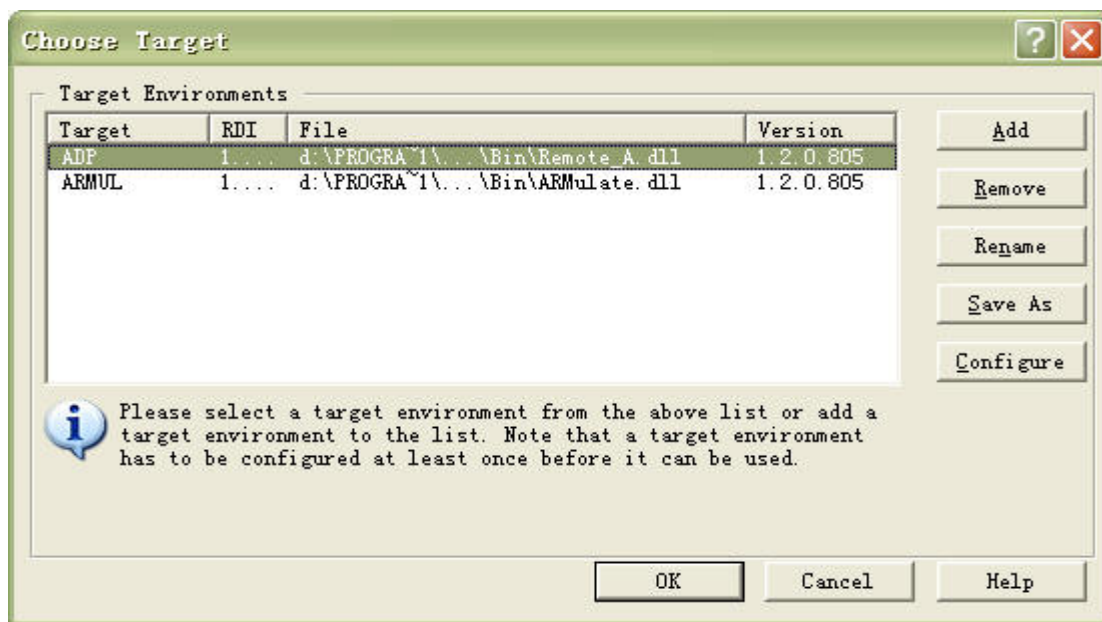


图 4-10 调试目标设置对话框

在 Target Environment 栏中选中“ADP”选项，注意到下面的注释的说明，“ADP 是直接连接 ARM Debugger 到目标板或者到目标板上的 EmbeddedICE 单元的一种方式。直接连接目标板需要 Angel 调试监视器软件的支持。参考 RDI 1.51”，点击“Configure”按钮，进入到下面设置：

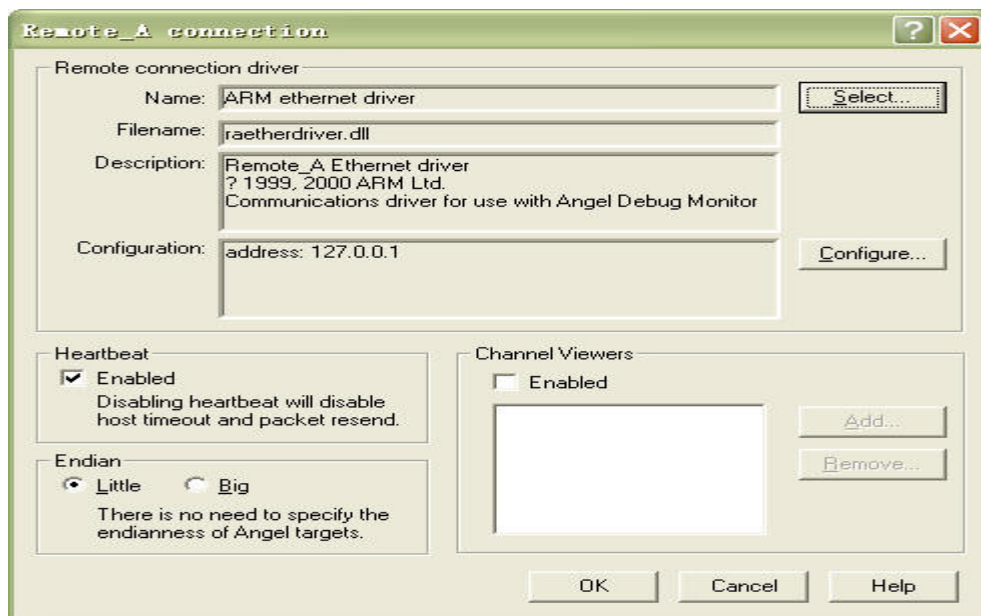


图 4-11 Remote_A 连接设置

其中“Remote Connection driver”栏中，点击右边的“Select...”按钮，选择“ARM ethernet driver”。点击右边的“Configure...”按钮，在编辑栏中输入本机的 IP 地址或者 127.0.0.1。其它设置如上图所示，保持不变。点击“OK”退出调试目标的设置。这时会弹出：



图 4-12 重新载入对话框

点击“是”按钮，如果目标系统正确链接了，会看到程序下载的进度条显示。进度消息框消失后，显示当前执行代码视窗，蓝色指针指向第一条执行的语句：

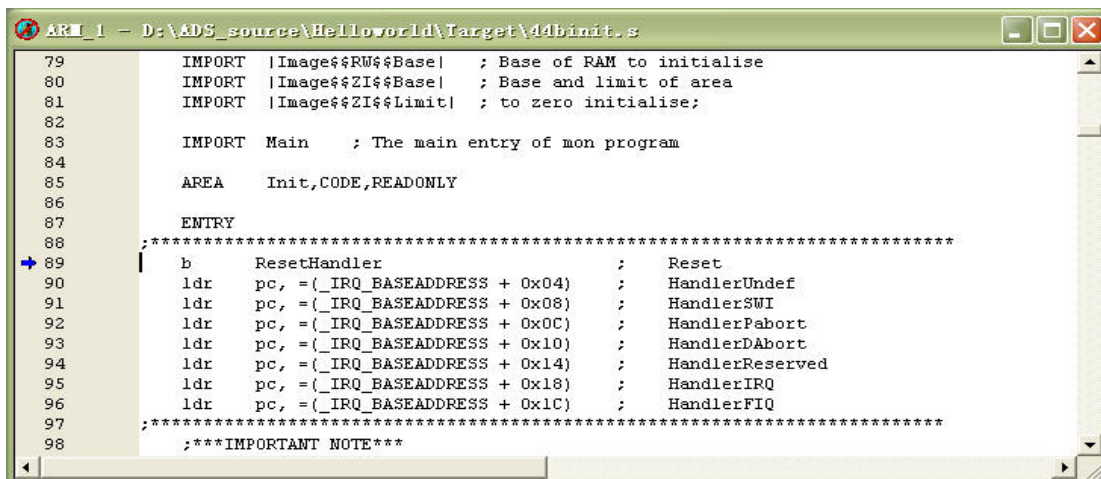






图 4-13 当前执行代码窗口

这时，先点击  按钮，尝试进行单步运行，如果程序立即正确地跳转到“ResetHandler”处执行，而没有跑飞或顺序执行，则说明程序的下载成功了，可以进行调试了。

4.4.3 AXD 调试器的使用

我们首先来熟悉一下断点的设置。下拉滚动条至 377 行，在 BL Main 语句处点击按钮  设置一个断点，如图 4-14。然后点击按钮  (GO)，令程序自动执行到断点。当程序执行到 BL Main 语句处，自动停止，点击按钮 ，程序跳转到 main.c 文件的 Main() 处程序开始运行，如图 4-15。

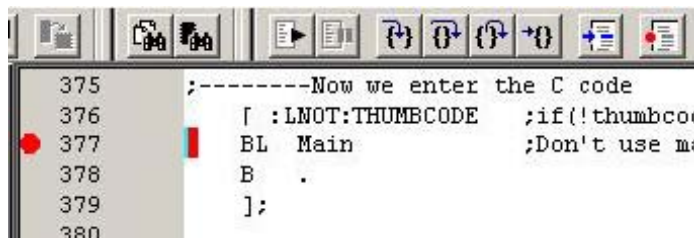


图 4-14 放置断点

```

1  #include "Target\44blib.h"
2  #include "Target\44b.h"
3
4  /*****
5  //  ARMSYS实验一: HelloWorld
6  //  描述: 连接PC机串口, 观察超级终端输出
7  *****/
8
9  void Main(void)
10
11     char aa;
12     rSYSCFG=CACHECFG;
13     Port_Init();
14     Uart_Init(0,115200);
15     Led_Display(0xf);
16     Uart_Select(0);
17     Reep(0x1);

```

图 4-15 进入主函数运行

通过上面的操作，我们了解到，44binit.s 程序中的 BL Main 语句就是跳转到 C 语言 main () 函数的入口语句。AXD 也会自动在 Main() 函数的入口处放置一个断点，因此程序下载后，立即全速运行的话，就会首先跳到该断点停下来。

读者可以继续对一些单步操作，了解每条语句的作用。

4.4.4 AXD 观测窗口

AXD 提供了许多有用的观察窗口，点击菜单项中的 Processor View ，可以从它的下拉菜单项中了解可观察的项目。

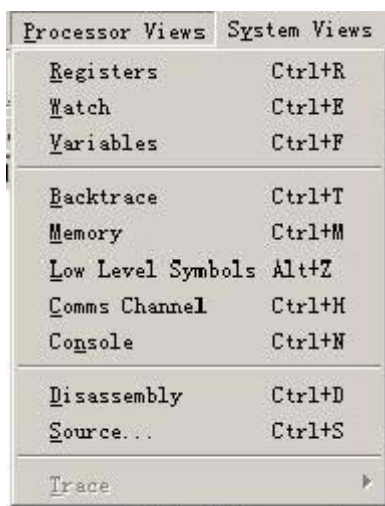


图 4-16 观测窗口菜单

这里说明一下其中常用的项目：

Registers: 可以查看 CPU 在各个工作模式下内部寄存器的值；


Variables : 查看变量，本地变量、全局变量、类变量；

Watch : 可以用表达式查看变量的值；

Backtrace: 函数调用情况（堆栈）查看；

Memory: 查看存储器内容。输入地址，即可查看这个地址开始的存储单元的值。

4.4.5 程序全速运行

在 AXD 中点击  ‘GO’ 图标，可以全速地运行程序，注意观察超级终端窗体，上面将显示如下信息。

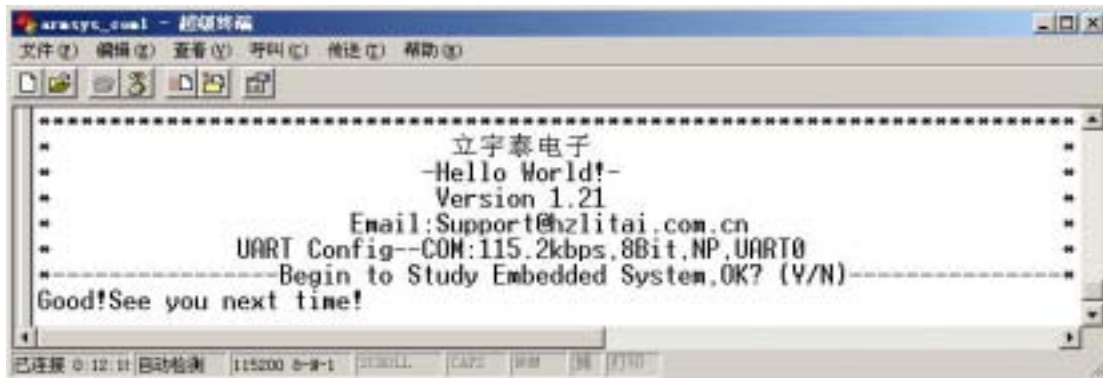


图 4-17 Hello World 运行后超级终端的显示

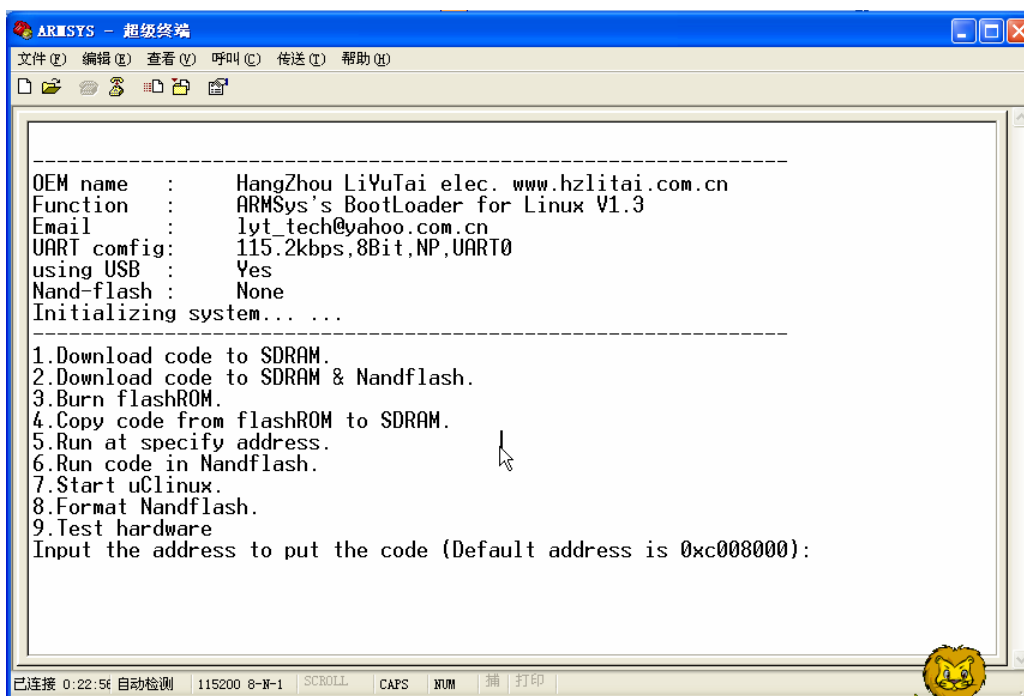
在超级终端上显示 “Begin to Study Embedded System, OK? (Y/N)” 后，在计算机键盘上键入 Y，超级终端上出现 “Good! See you next time!”。

第五章 程序代码下载与存储

5.1 程序文件下载至开发板的途径

5.1.1 通过 USB 下载程序文件

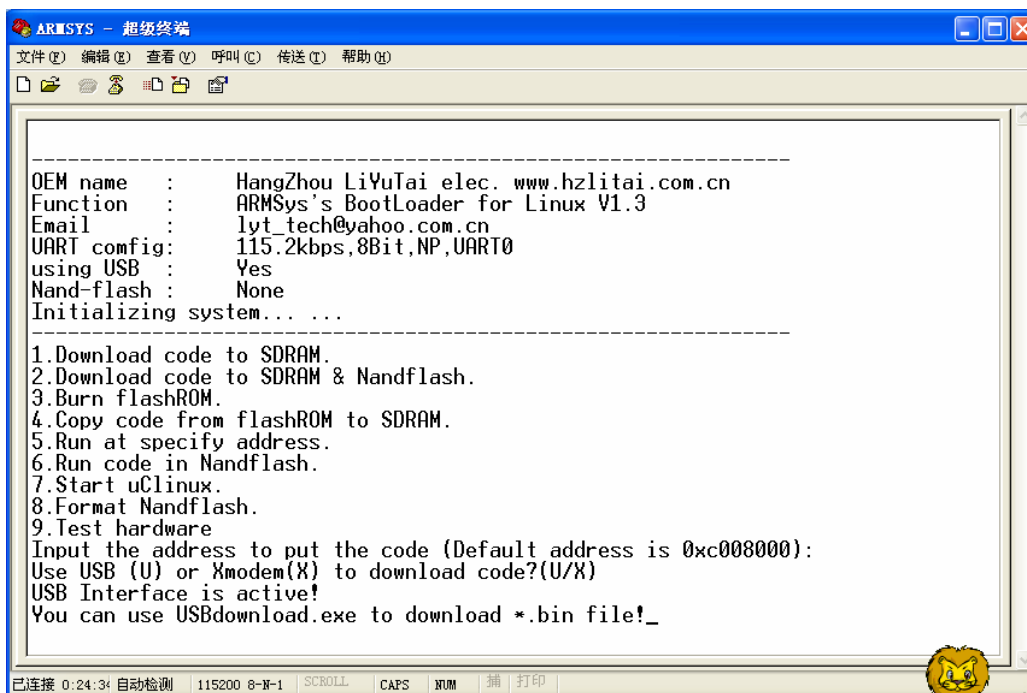
首先将开发板的电源、串口和 usb 连接好，然后打开电源开关，开发板 bootloader 启动完成以后，在超级终端中输入 BOOTLOADER 命令“1”，如下图所示：



```
ARMSYS - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
-----
OEM name : HangZhou LiYuTai elec. www.hzlitai.com.cn
Function : ARMSys's BootLoader for Linux V1.3
Email : lyt_tech@yahoo.com.cn
UART comfig: 115.2kbps,8Bit,NP,UART0
using USB : Yes
Nand-flash : None
Initializing system... ..
-----
1.Download code to SDRAM.
2.Download code to SDRAM & Nandflash.
3.Burn flashROM.
4.Copy code from flashROM to SDRAM.
5.Run at specify address.
6.Run code in Nandflash.
7.Start uClinux.
8.Format Nandflash.
9.Test hardware
Input the address to put the code (Default address is 0xc008000):
-----
已连接 0:22:56 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印
```

“1”命令是启动开发板下载程序，可以在命令行中加入下载目的地址，没有输入时默认下载地址为“0xC008000”，如要将程序下载到目的地址为“0xC200000”时，输入命令“0xC200000”即可。

再按下键盘上的‘U’或者按回车键，将采用 usb 下载方式，下图所示：



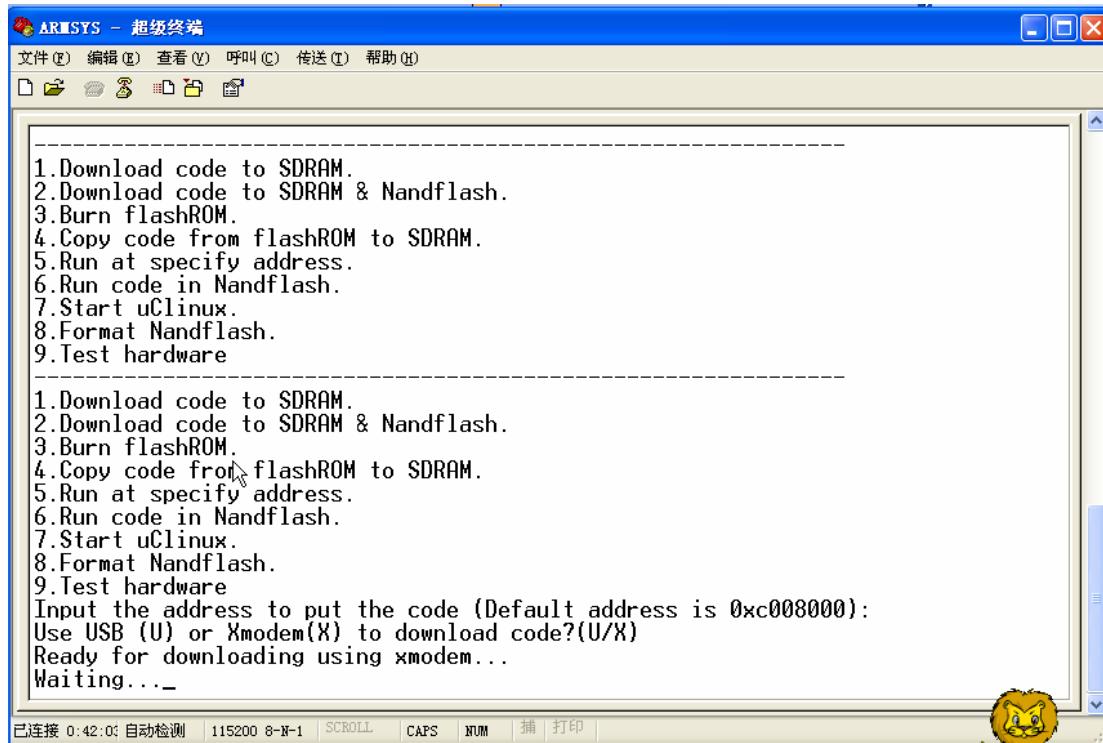
这样 USB 接口就成功地被使能了；这时打开“设备管理器”会看到 Jungo 下面自动识别出了 USB Tool Device 设备，说明驱动安装成功了。

点击 USBTool 工具的菜单项[文件 | 打开文件]，然后在“文件类型”中下拉选择*.bin 类型文件，选择好要下载的文件后（例如刚刚产生的 helloworld.bin 文件），点击“打开”。这时可以看到，对话框的文件名、后缀、文件长度编辑框中即出现所选文件的相关信息。点击“下载文件”按钮，二进制文件被下载到 ARMSys 中。等待一会儿，系统自动开始运行刚刚下载的程序。通过超级终端上显示的内容，可以看到程序的运行情况。



5.1.2 通过串口下载程序文件

将开发板接上电源和串口电缆到计算机，打开计算机上的超级终端和开发板电源，在超级终端中输入命令“1”，下载目的地址采用默认的地址“0xC008000”，也可改变下载的目的地址，如“c200000”。然后再终端中输入“X”，如下图所示：



启动开发板串口 Xmodem 传输程序后，等待计算机运行 Xmodem 程序下载程序文件到开发板。然后在超级终端窗口点击[传送|发送文件]，出现如下窗口：

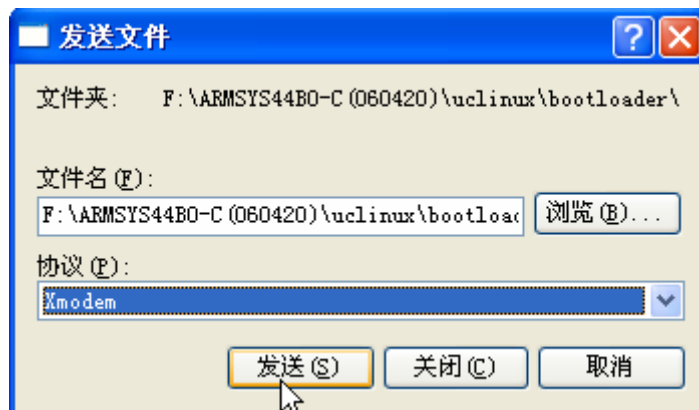


图 6-5 Xmodem 发送文件

点击[浏览]选择要下载的程序文件，协议选择“Xmodem”或“1K Xmodem”然后点击[发送]即可开始传输程序文件了。可以看到如下图所示的传输过程：



图 6-6 串口 Xmodem 传输过程

当程序文件传输完了，该窗口关闭，通过串口传输完成。

5.2 采用 bootloader 烧写程序代码到 NorFlash 中

ARMSYS44B0-C 开发板的 Bank0 接至 NorFlash 芯片，在开发板启动时就从 NorFlash 的零地址开始度取程序执行。因此，NorFlash 的从零地址开始要存放开发板的启动程序代码，烧写时要注意这一点。采用 bootloader 烧写程序代码到 NorFlash 时要用到 44bapp.bin 烧写工具，先把该烧写工具下载到开发板的 SDRAM 中，再把被烧写的程序代码下载到地址为“0xC200000”中，代码大小不能超出 NorFlash 的容量，具体步骤如下（以下为系统出厂时的烧录步骤）：

(1) 利用 USB 或串口先将 44bapp.bin 程序代码文件下载到目的地址为“0xC008000”处，等到下面步骤执行完后运行该代码。

(2) 下载 bootloader.bin 代码文件到“0xC200000”地址中。

(3) 下载 imagerom.bin 代码文件到“0xC220000”地址中。

(4) 在超级终端中输入命令“5”回车，开始执行烧写。

烧写完成后，绿色的发光管会全亮。然后复位即执行刚才烧写的程序代码。

5.3 启动代码的烧写（系统恢复）

5.3.1 使用 FlashPGM 快速烧写 Flash

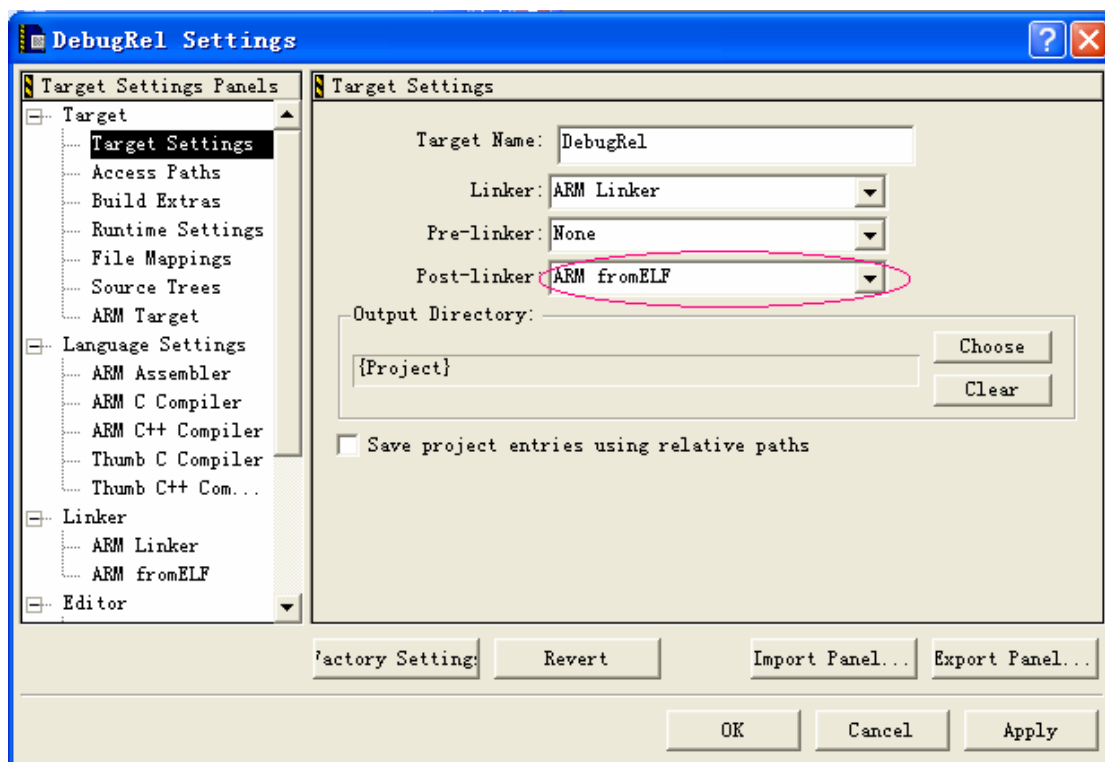
5.3.1.1 计算机的设置

请在PC 机的BIOS 里面将并口设置成 EPP 方式；

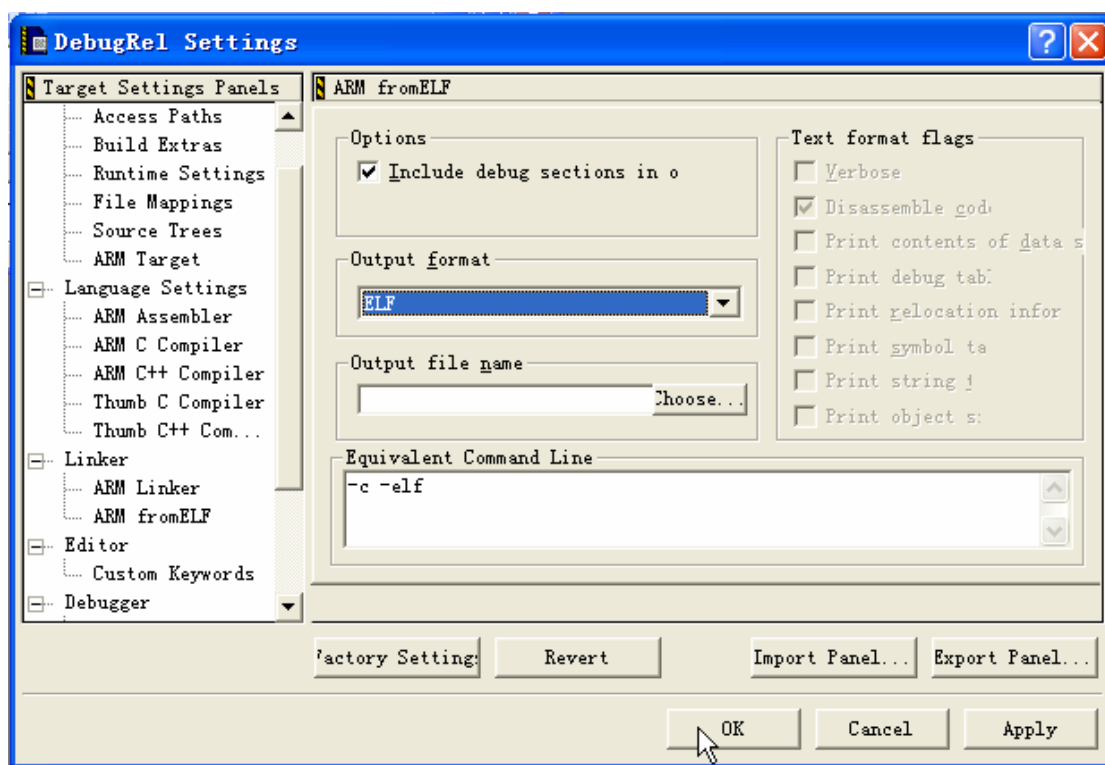
ELF 文件的准备

flashpgm 不能烧写*.bin 格式的文件，但可以烧写*.elf 或者是*.hex 格式的文件，一般都使用*.elf 格式的文件进行烧写。

以ADS1.2 为例，在其项目编译界面下按ALT+F7 进入下面的设置界面，按照下面画红色圆圈的画面在Target Settings 目录下选择ARM fromELF ，以生成目标代码：



然后在Linker目录下选择ELF。

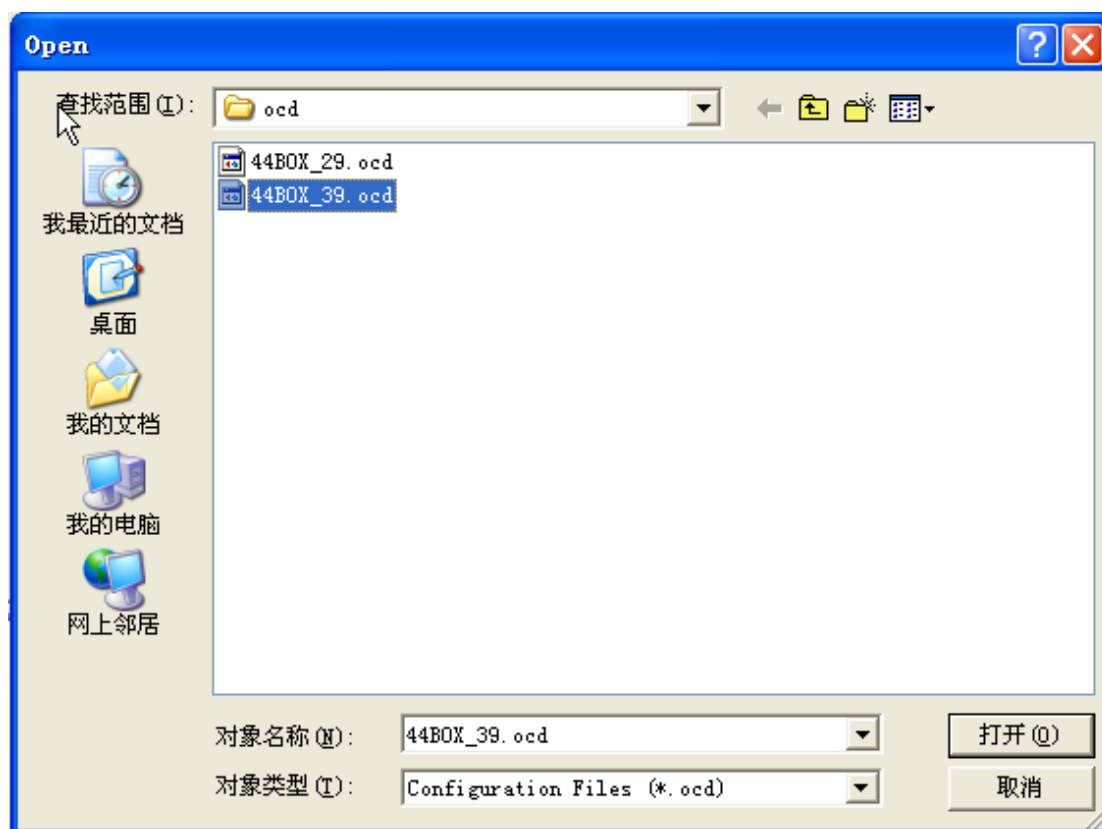
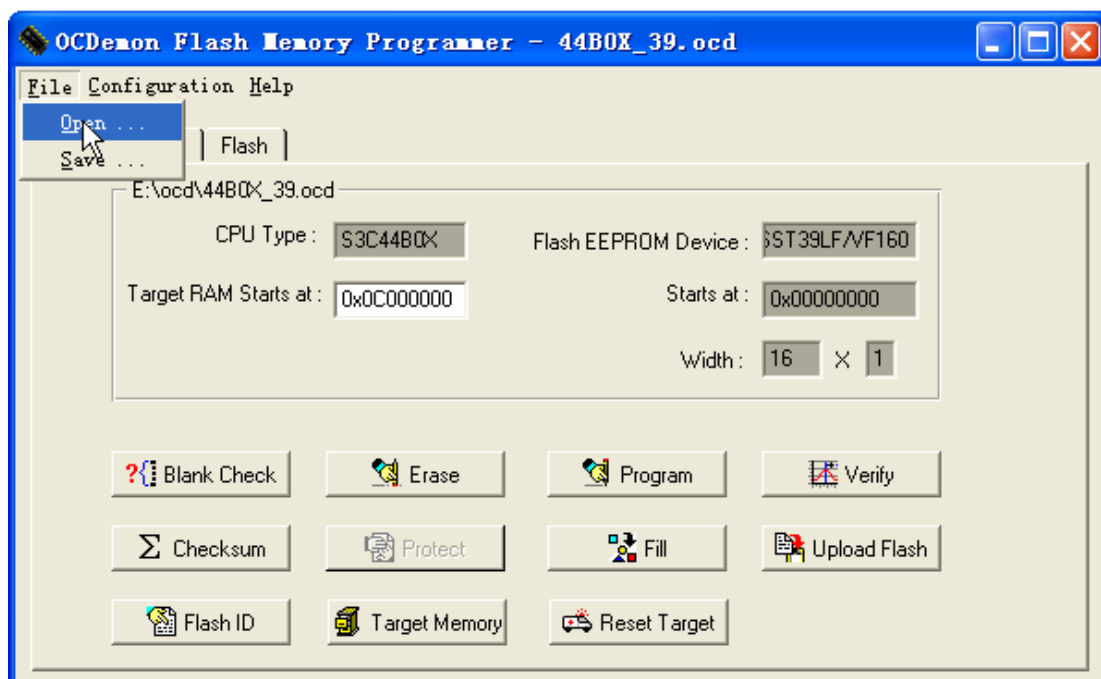


然后编译就会生成你所设置文件名的ELF格式文件了。

注意：Flashpgm 烧写时Jtag 的接法。（Jtag小板上选择wiggler接口）

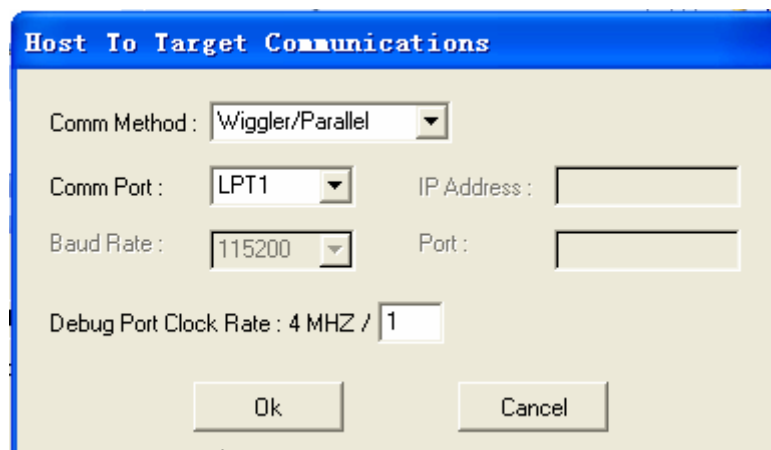
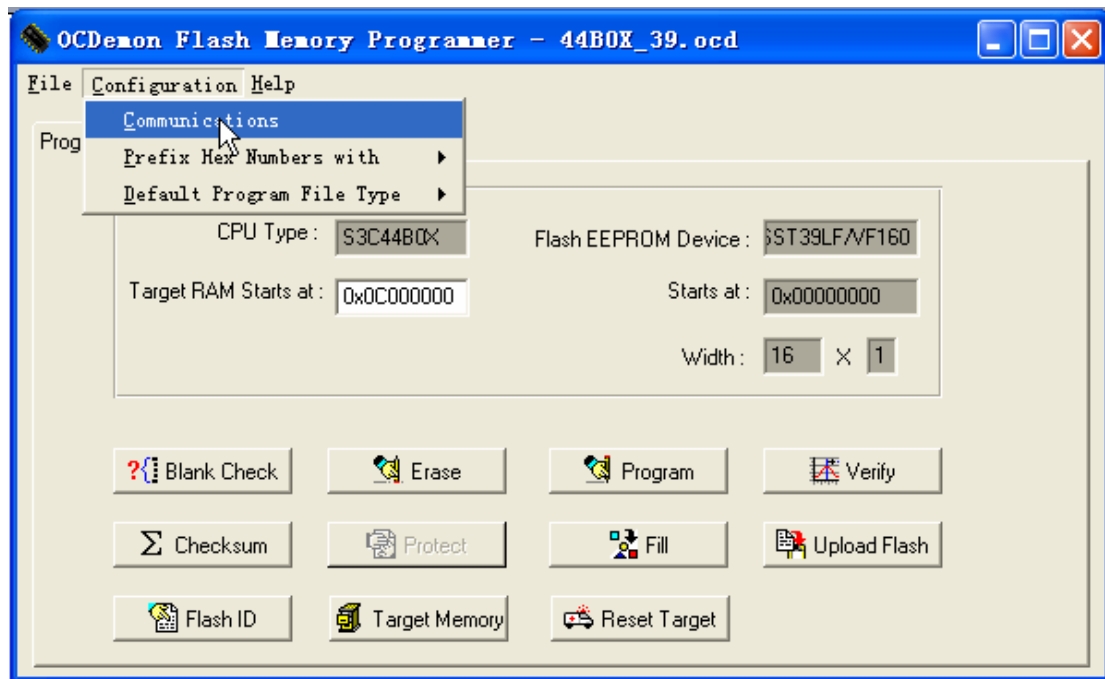
5.3.1.2 Flashpgm 的设置

调入 44BOX_39.ocd 文件

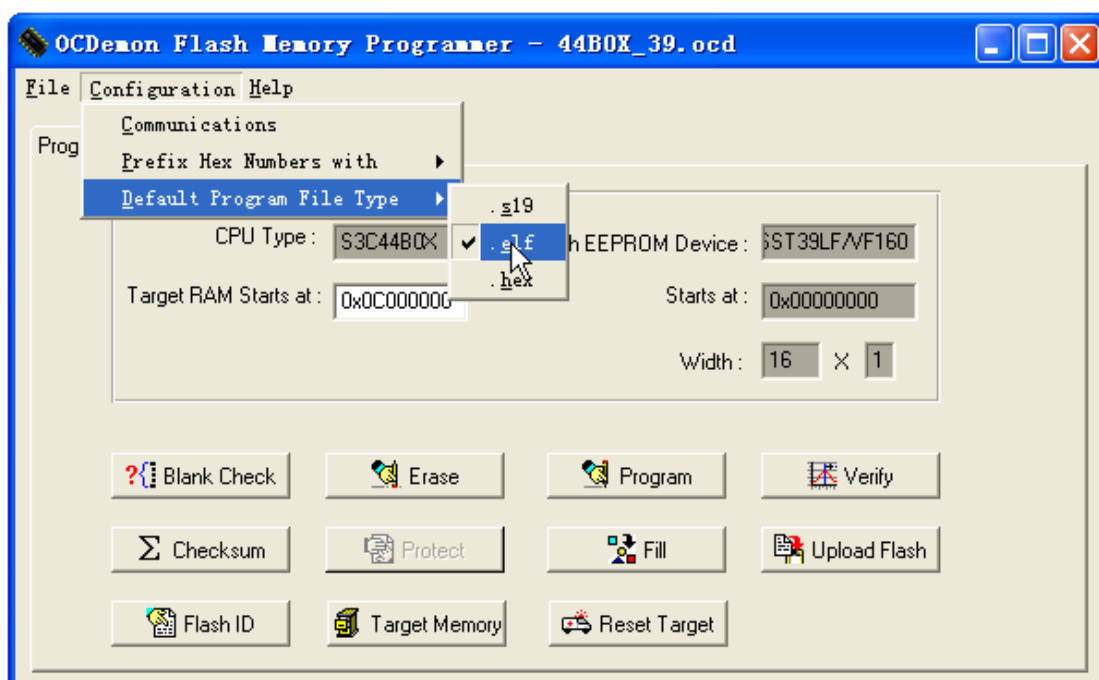


5.3.1.3 设置通信端口

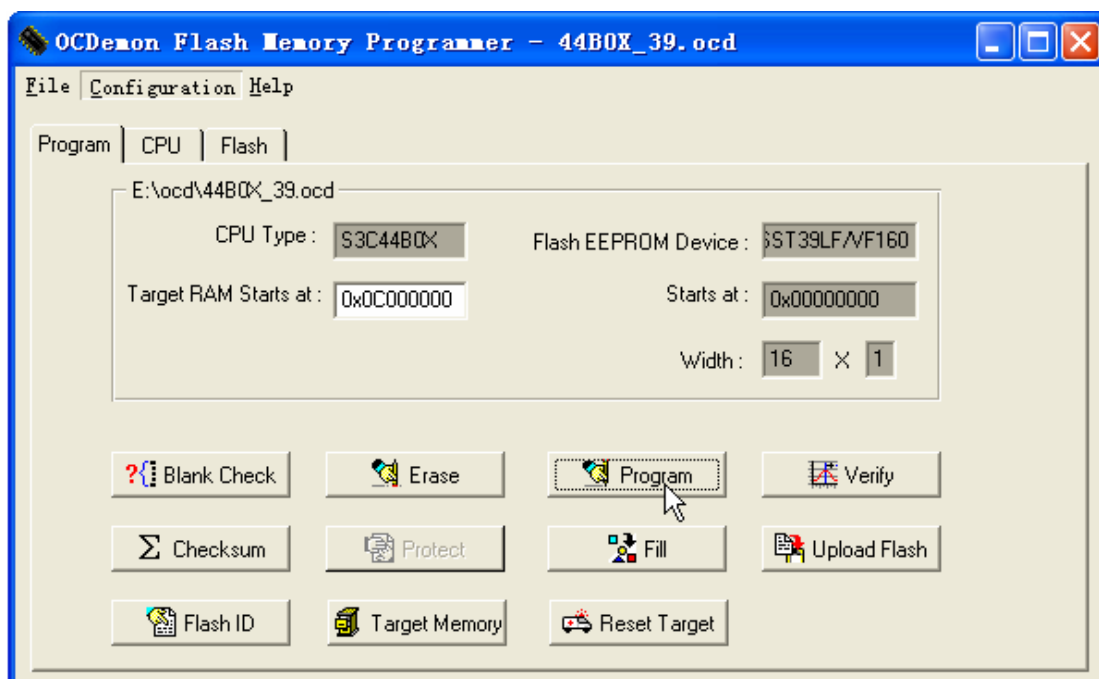
选择Wiggler Parallel



5.3.1.4 选择编程文件类型



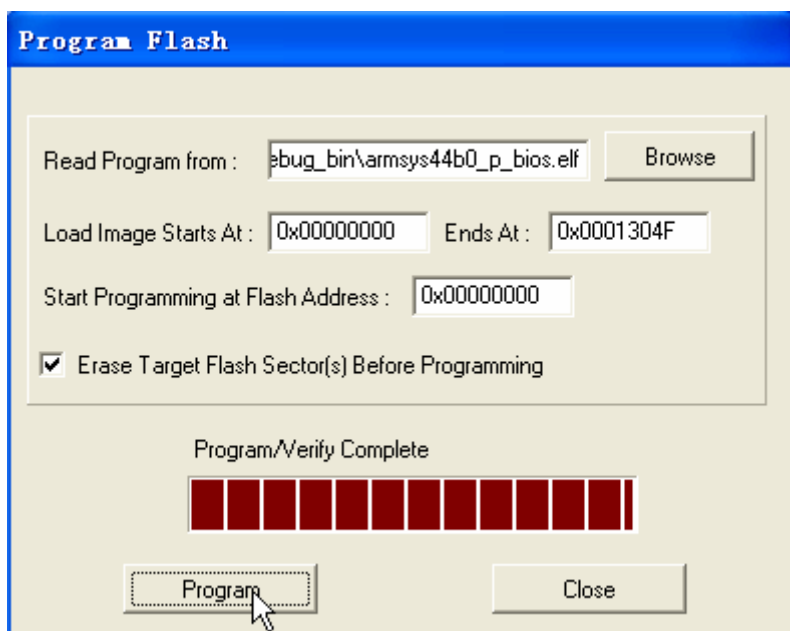
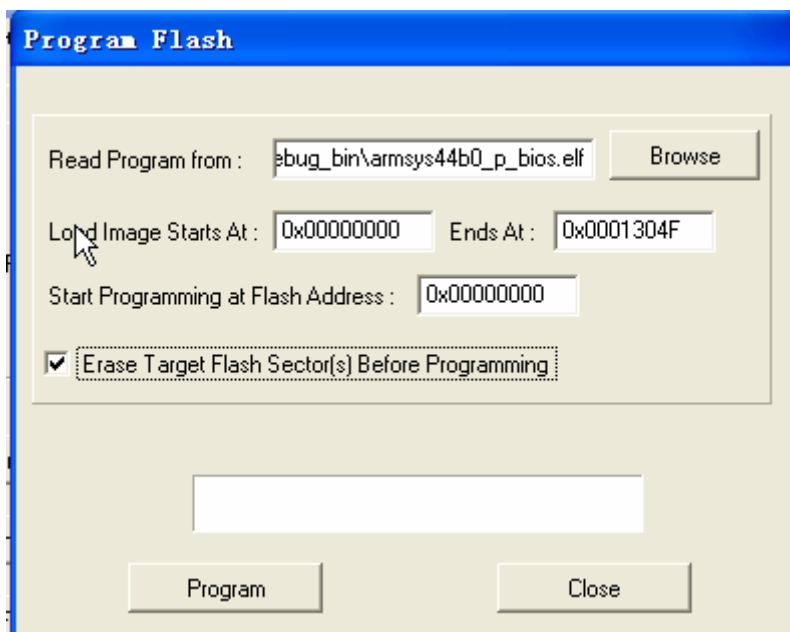
5.3.1.5 点击编程按钮



如果目标板连接不正常将显示如下对话框，请检查目标板电源是否打开，20 芯排线是否连接正常。



如果连接正常，将显示如下对话框。点击Browse 按钮，找到要烧写的文件的位置，如下图，另外，我们也要选中 Erase Target Flash Sector(s) Before Program




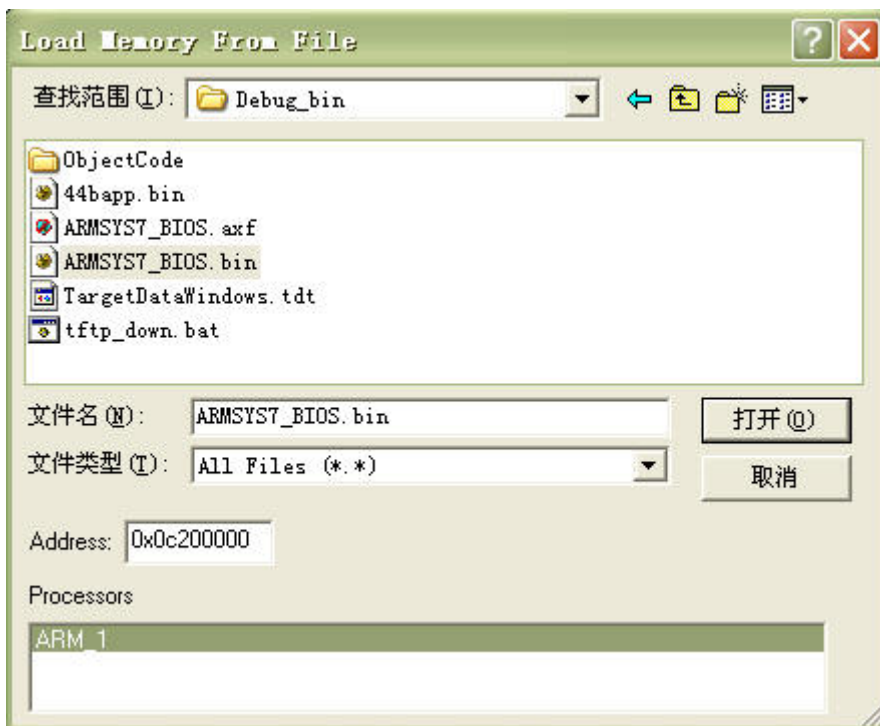
如果在上面的画面中，你发现按钮 Program 始终为灰色，无论如何也不行。请点击 Close 按钮关闭上面的画面，然后按一下板子上的复位按钮，再点击 Flashpgm 界面中的 Program 按钮，重新进入上面的画面，就会发现 Program 不在是灰色的了，这样就可以编程了。这

实际上是 Flashgpm 的一个 bug，但没有更好的办法，不过这种情况不多见而已。

5.3.2 用 programmer 工程烧写

programmer 是一个烧写工程，用它进行烧录要使用到 ADS1.2 的 AXD 调试功能。programmer 烧录的优点是速度较 Fluted 快。操作步骤如下：

将开发板配套光盘中“开发工具\programmer\ads_programmer”文件夹复制到计算机硬盘中并去掉只读属性。编译该工程进入 AXD 调试环境，然后加载要烧写的启动代码，在 AXD 调试窗口下点击菜单“File-> Load Memory From File...”或者点击图标出现下图所示窗口：



选择要烧写的代码文件（如 bootloader.bin），在 Address 地址栏中输入“0x0c200000”，然后点击打开，可以看 JTAG 模块加载代码时的绿色工作发光管被点亮，AXD 调试环境中也可看到加载进度调条。

这样启动代码 bin 文件很快被下载到相应的地址。全速运行烧录程序。可以通过超级终端观察到烧录的过程，烧录过程分为片擦除、空检查、写入和校验，烧录成功的输出信息为：

```
Chip erased!  
Blank check OK!  
Begin to Write flash...  
Write OK! Begin to Verify...  
Verify OK!
```

如果终端输出以上信息，其间没有报告错误，说明烧录已经成功了。

第六章 ARMSYS44B0-C 测试程序

6.1 测试前的准备

首先请仔细阅读以上第三章第五章的说明书,并在全面测试开发板的各项功能之前进行如下准备工作:

- (1) 用配套的 USB 电缆线将计算机 USB 接口和开发板的 USB 接口相连接。
- (2) 用串口电缆连接开发板的 UART0 接口到计算机串口。
- (3) 用交叉对等网线连接开发板和计算机的 RJ45 接口,或者用普通直连网线将开发板连接到计算机所在的局域网中。
- (4) 如有液晶屏和触摸屏模块时,用排线接至 LCD&TSP 接口。
- (5) 拨动电源转换开关 SW1 至“EXT”位置,此时开发板接通电源开始工作,红色电源指示发光管点亮,按下复位键,开发板上的蜂鸣器将发出“滴”的响声,四个绿色发光二极管将点亮后熄灭。

6.2 利用 bootloader 测试开发板的各项功能

ARMSYS44b0-c 开发板提供的 bootloader 是一个功能强大,使用方便的软件。它被烧写在 FLASH 中,上电即可运行。bootloader 可以方便的实现:

4. 开发板的硬件测试
5. USB 下载

关于 USB 下载功能的使用请查看 5.1 节。本节仅对开发板的测试功能的相关操作进行说明。

【注意】如果您的开发板烧录的是可用于启动 linux 的 bootloader,那么它同样具备以下测试功能,只是人机界面有些不同。例如你同样可以用选项 9 来进行各个功能那个项的测试。

开发板的测试步骤如下:

Step1: 正确连接开发板与计算机的接口电缆

首先用串行电缆线连接计算机的串行口与开发板上的 UART0 串行口、使用 USB 电缆连接计算机的 USB 口与开发板主板上的 USBPORT 端口。

如果需要进行 JTAG 调试,使用并口电缆线连接计算机的并行口与开发板的 JTAG 调试模块的并口,再通过排线连接开发板的 JTAG 调试模块与开发板主板上的 JTAG20 端口。

一般情况下,开发板可以直接通过 USB 口取电工作,但是计算机 USB 口提供的最大电流有限制 ($\leq 500\text{mA}$),因此在调试消耗电流比较大的控制对象时,注意使用合适的外接电源。例如,在使用 LCD 显示时,最好使用供电大于 500mA 的外接直流电源。

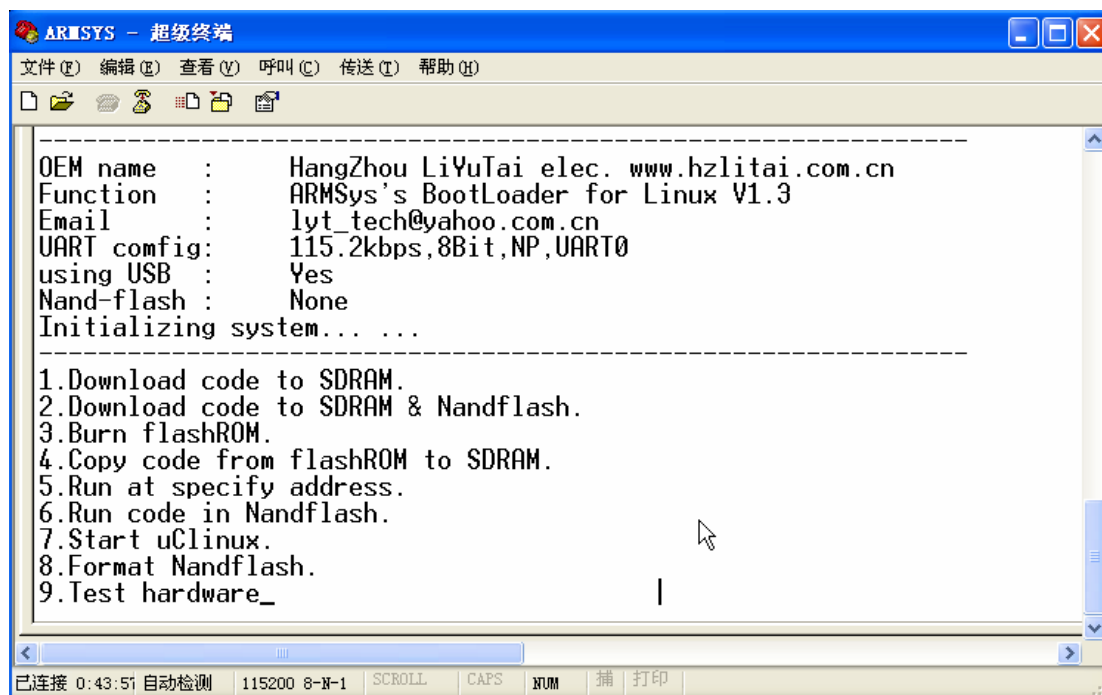
(注:如果采用 usb 下程序,请不要在开发板上插 jtag 调试模块。)

Step2: 打开电源

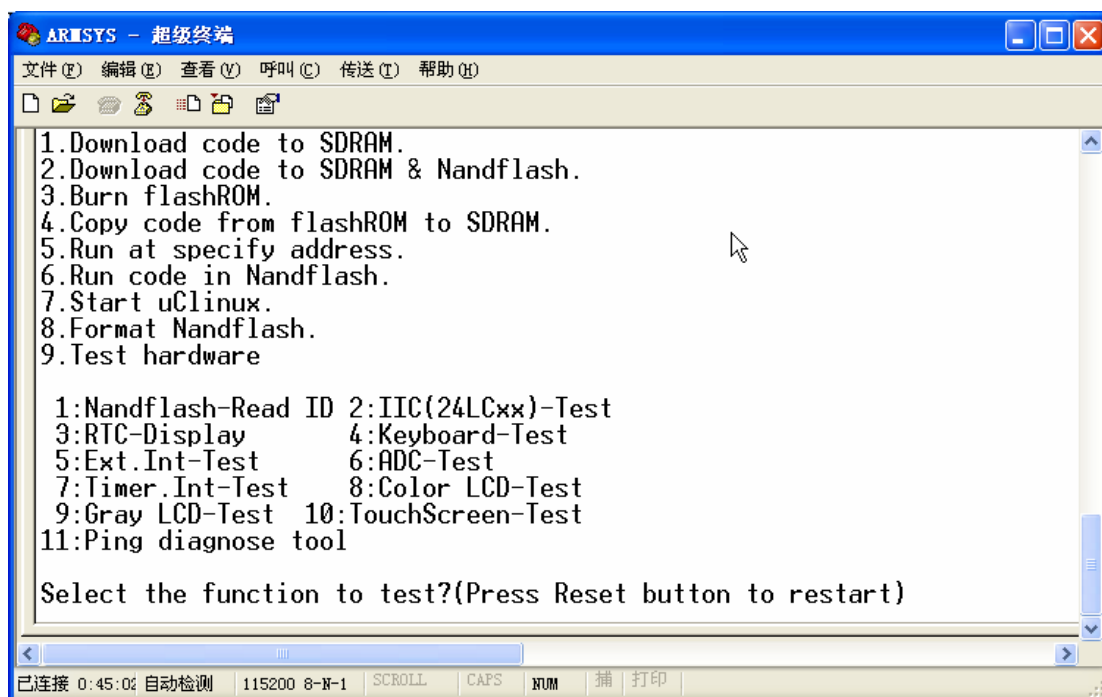
把主板上 SW1 的拨动开关拨到 USB 位置(如果是外接电源则拨到 EXT 位置),此时主板上的红色 LED 点亮。

Step3: 系统复位

按下主板上的复位键,确保系统复位,此时超级终端中可以看到如下图的内容:



如果按下“9”键，超级终端输出如下图的信息：



Step4: Nandflash 格式化

由于经济型开发板不具备 Nandflash 部件，因此这里省略相关说明。

Step5: 运行 Nandflash 中已经保存为文件的程序

由于经济型开发板不具备 Nandflash 部件，因此这里省略相关说明。

Step6: ARMSYS 功能部件测试

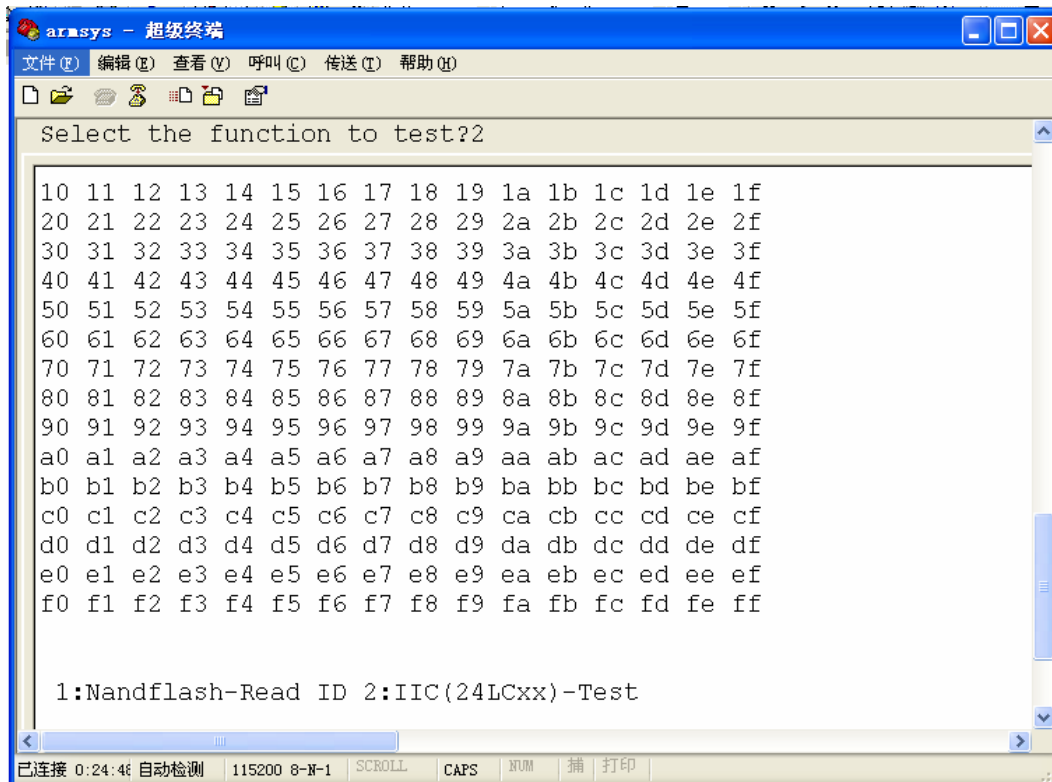
按下计算机“1~11”数字键，超级终端中显示如下图，这时，开发板的硬件测试就开始了。下面对其中的几个选项进行依次说明：

(1) Nandflash-Read ID 的测试

由于经济型开发板不具备 Nandflash 部件，因此这里省略相关说明。

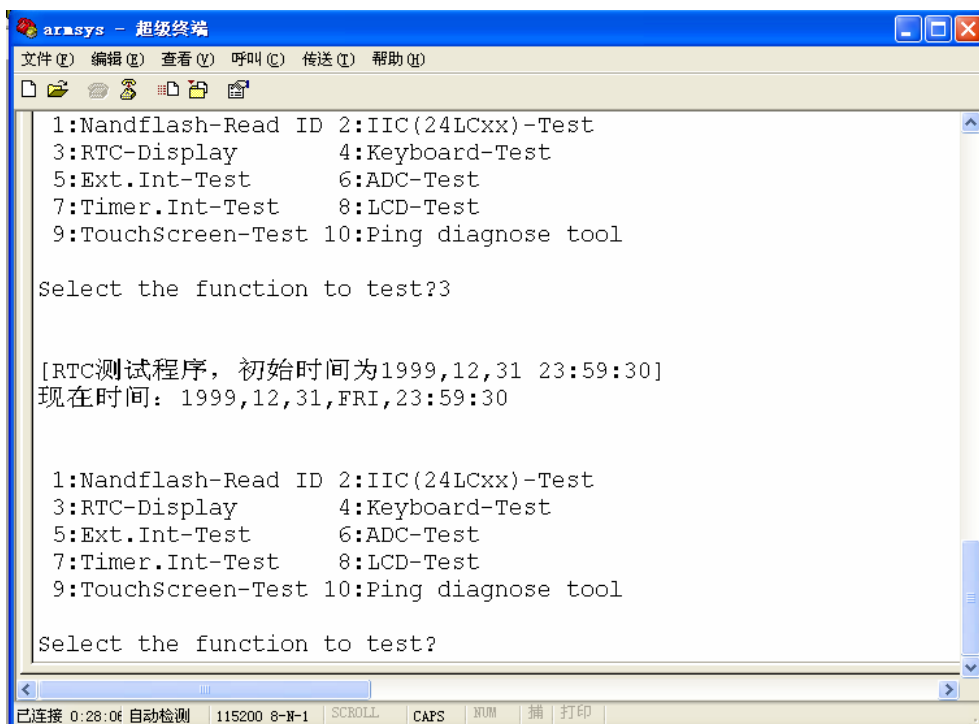
(2) IIC (24LCXX) 的测试

按下计算机键盘的“2”键，然后按回车键，超级终端将输出如下图所示的信息：



(3) RTC-Display 的测试

按下计算机键盘的“3”键，然后按回车键，超级终端将输出如下图所示的信息：



超级终端中显示的信息：

```
“[RTC 测试程序，初始时间为 1999,12,31 23:59:30]
现在时间： 1999,12,31,FRI,23:59:30”
```

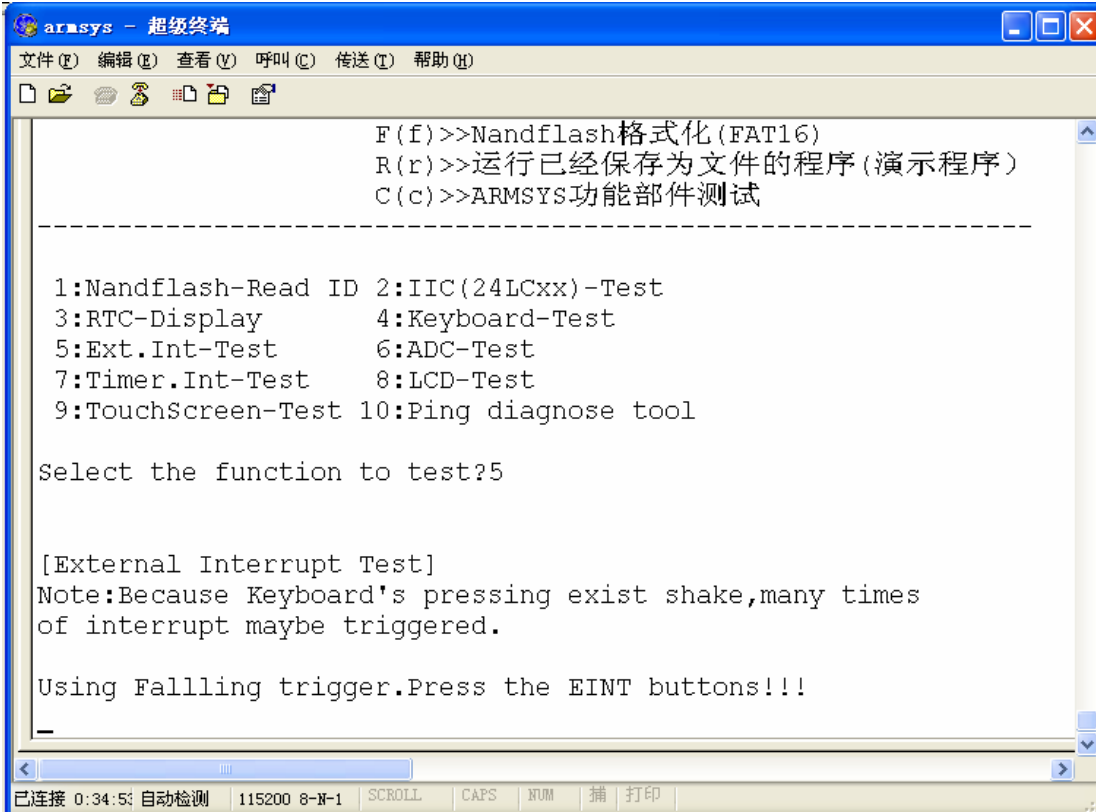
这个时间也是测试的起始时间。再多次按下“3”键，时间会从这个起始时间开始自动计时更新。

(4) KEYBOARD 的测试

该项提供给实验箱使用，开发板不使用，本文档就不加以说明了。

(5) Ext.Int (中断输入) 的测试

按下计算机键盘的“5”键，然后按回车键，超级终端将输出如下图所示的信息：



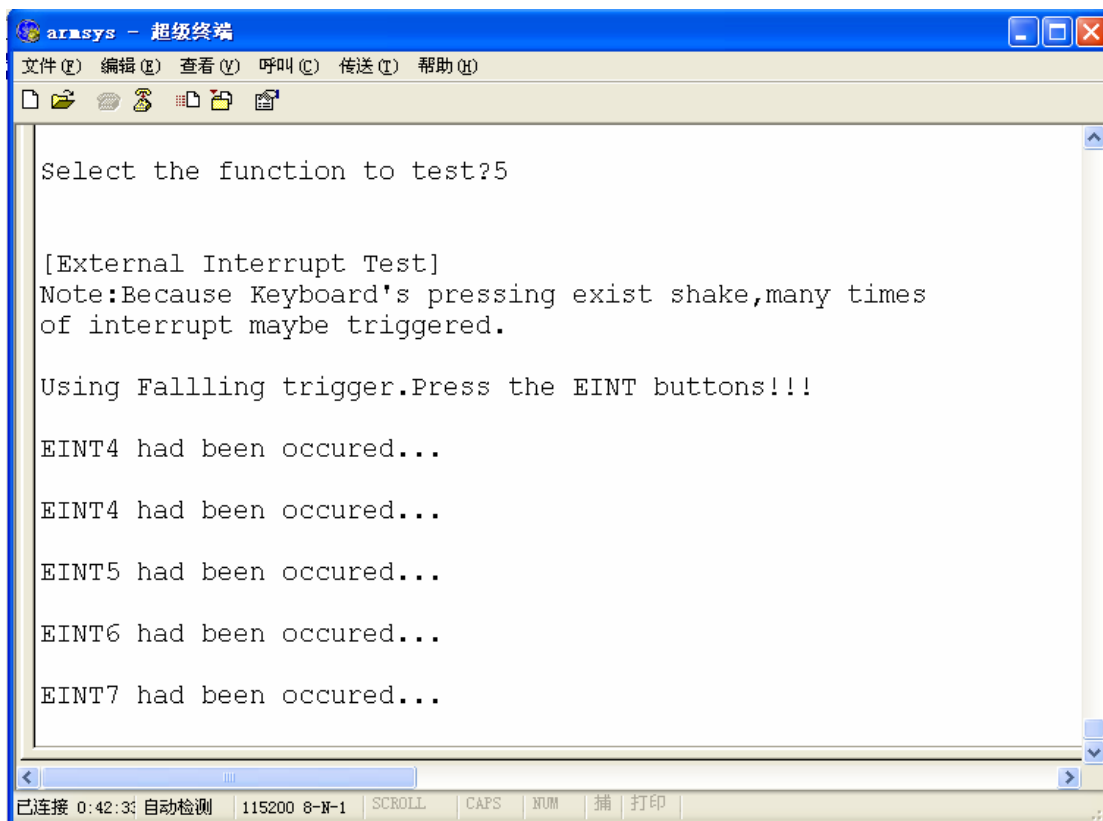
```
armsys - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
F(f)>>Nandflash格式化(FAT16)
R(r)>>运行已经保存为文件的程序(演示程序)
C(c)>>ARMSYS功能部件测试
-----
1:Nandflash-Read ID 2:IIC(24LCxx)-Test
3:RTC-Display      4:Keyboard-Test
5:Ext.Int-Test     6:ADC-Test
7:Timer.Int-Test  8:LCD-Test
9:TouchScreen-Test 10:Ping diagnose tool

Select the function to test?5

[External Interrupt Test]
Note:Because Keyboard's pressing exist shake,many times
of interrupt maybe triggered.

Using Fallling trigger.Press the EINT buttons!!!
```

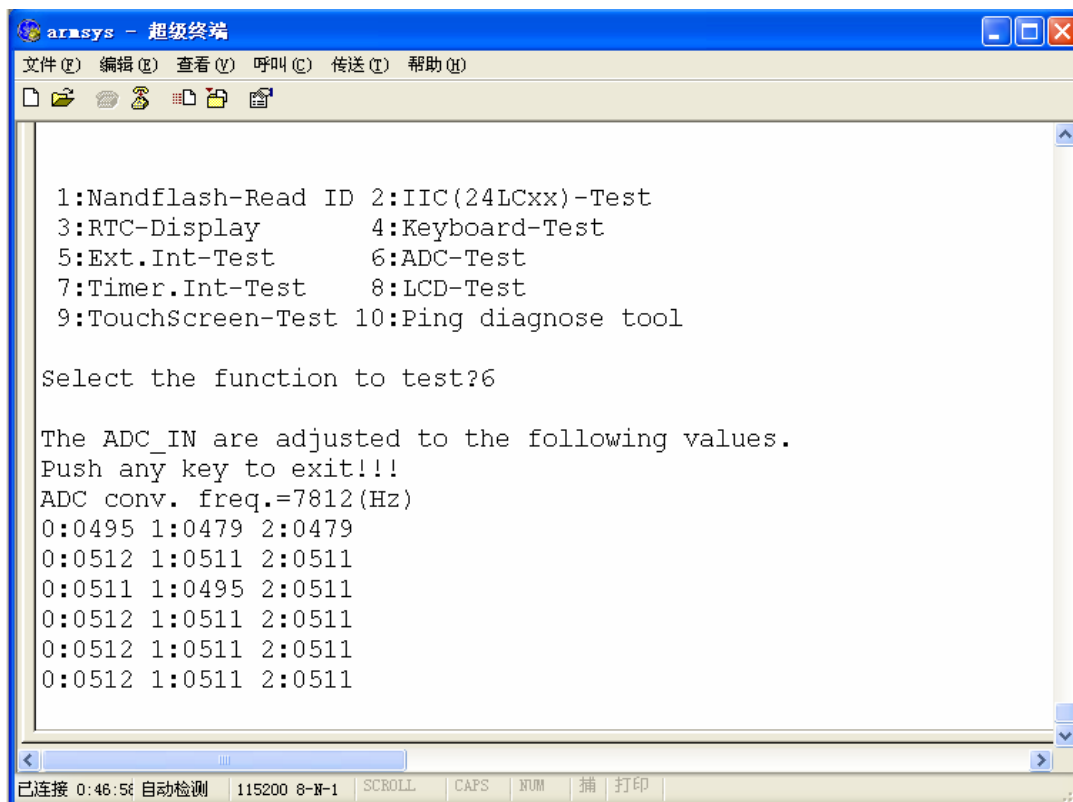
超级终端输出的信息：“Using Fallling trigger.Press the EINT buttons!!!”，然后，依次按下主板上的四个中断按键“ExInt4”、“ExInt5”、“ExInt6”、“ExInt7”。超级终端将输出以下信息：



按下 PC 机键盘上的任意键，退出中断测试。

(6) ADC 输出测试

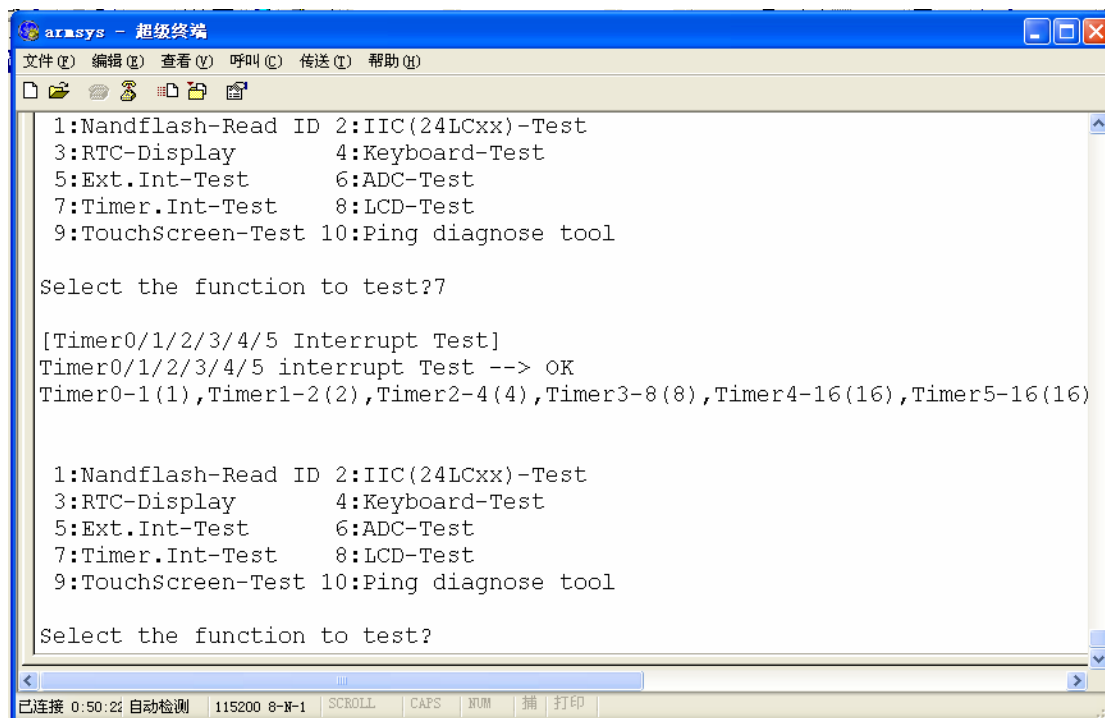
按下计算机键盘的“6”键，然后按回车键，超级终端将输出如下图所示的信息：



按下回车键，退出 ADC 输出测试。

(7) Timer.Int 定时器中断测试

按下计算机键盘的“7”键，然后按回车键，超级终端将输出如下图所示的信息：



```
armsys - 超级终端
文件(F) 编辑(E) 查看(V) 呼叫(C) 传送(T) 帮助(H)
1:Nandflash-Read ID 2:IIC(24LCxx)-Test
3:RTC-Display      4:Keyboard-Test
5:Ext.Int-Test     6:ADC-Test
7:Timer.Int-Test  8:LCD-Test
9:TouchScreen-Test 10:Ping diagnose tool

Select the function to test?7

[Timer0/1/2/3/4/5 Interrupt Test]
Timer0/1/2/3/4/5 interrupt Test --> OK
Timer0-1(1),Timer1-2(2),Timer2-4(4),Timer3-8(8),Timer4-16(16),Timer5-16(16)

1:Nandflash-Read ID 2:IIC(24LCxx)-Test
3:RTC-Display      4:Keyboard-Test
5:Ext.Int-Test     6:ADC-Test
7:Timer.Int-Test  8:LCD-Test
9:TouchScreen-Test 10:Ping diagnose tool

Select the function to test?

已连接 0:50:22 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印
```

(8) LCD 的测试

由于经济型不配套 LCD 屏，因此这里省略该项测试内容。

(9) TouchScreen（触摸屏）的测试

由于经济型不配套 LCD 屏，因此这里省略该项测试内容。

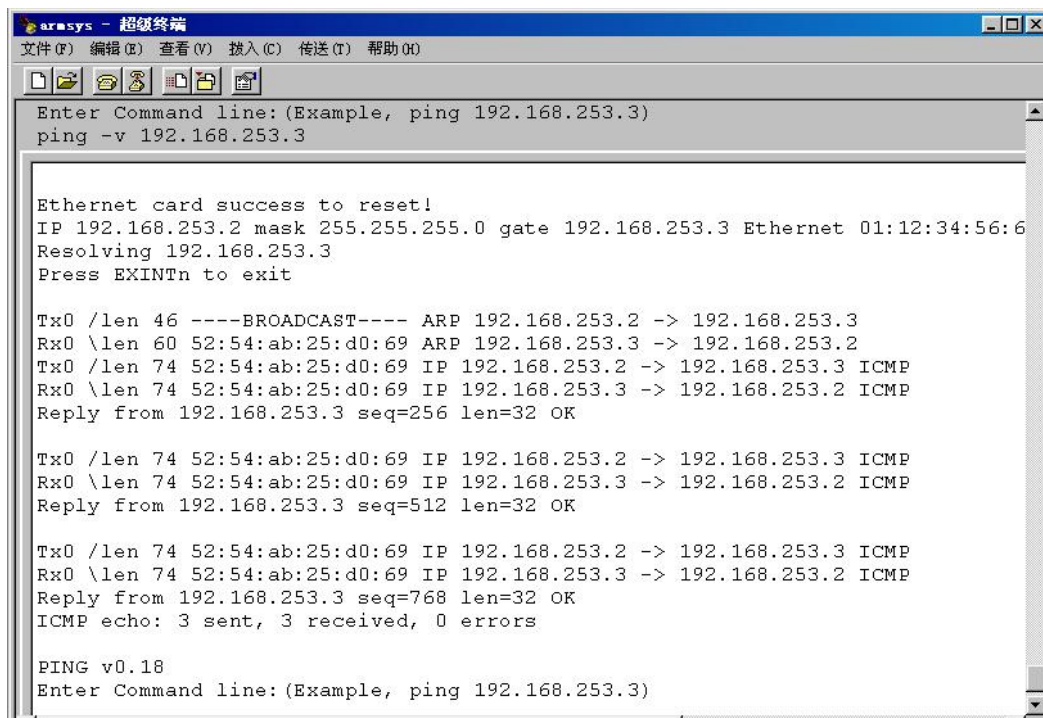
(10) 以太网卡测试

断开开发板电源。首先拔下刚刚插入的触摸屏控制板，正确插入以太网控制板。用以太网交叉线连接计算机的以太网卡和以太网控制板。将计算机的 IP 地址设置为：192.168.253.x (x 不能为 2)，然后打开开发板电源，按下 N, C 重新进入测试界面。

按下计算机键盘的“10”键，然后按回车键。超级终端上出现：Enter Command line: (Example, ping 192.168.253.3)。

输入命令行：ping -v 192.168.253.x (其中 -v 表示解析整个 ping 过程，192.168.253.X 表示你的 PC 机的 IP 地址) 敲回车。如果一切顺利，超级终端上应出现如下显示 (这里我们使用的 PC 机的 IP 地址是 192.168.253.3，可以看到开发板的 IP 地址是 192.168.253.2)。

【注意】改命令行不可重复输入，需要重新复位开发板才能再次正确地 ping 通主机。



```
arSYS - 超级终端
文件(F) 编辑(E) 查看(V) 拨入(C) 传送(T) 帮助(O)

Enter Command line: (Example, ping 192.168.253.3)
ping -v 192.168.253.3

Ethernet card success to reset!
IP 192.168.253.2 mask 255.255.255.0 gate 192.168.253.3 Ethernet 01:12:34:56:6
Resolving 192.168.253.3
Press EXINTn to exit

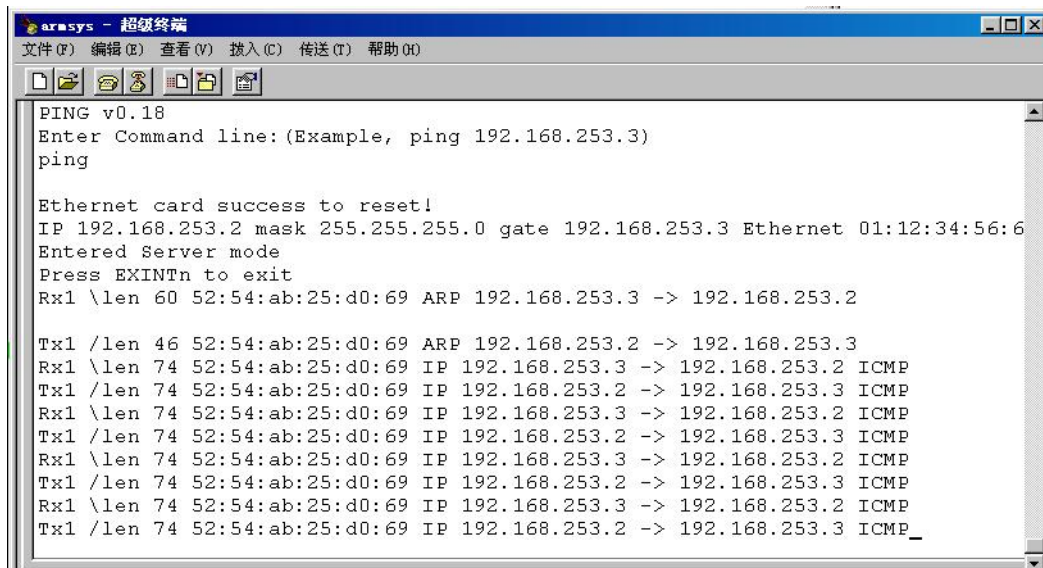
Tx0 /len 46 ----BROADCAST---- ARP 192.168.253.2 -> 192.168.253.3
Rx0 \len 60 52:54:ab:25:d0:69 ARP 192.168.253.3 -> 192.168.253.2
Tx0 /len 74 52:54:ab:25:d0:69 IP 192.168.253.2 -> 192.168.253.3 ICMP
Rx0 \len 74 52:54:ab:25:d0:69 IP 192.168.253.3 -> 192.168.253.2 ICMP
Reply from 192.168.253.3 seq=256 len=32 OK

Tx0 /len 74 52:54:ab:25:d0:69 IP 192.168.253.2 -> 192.168.253.3 ICMP
Rx0 \len 74 52:54:ab:25:d0:69 IP 192.168.253.3 -> 192.168.253.2 ICMP
Reply from 192.168.253.3 seq=512 len=32 OK

Tx0 /len 74 52:54:ab:25:d0:69 IP 192.168.253.2 -> 192.168.253.3 ICMP
Rx0 \len 74 52:54:ab:25:d0:69 IP 192.168.253.3 -> 192.168.253.2 ICMP
Reply from 192.168.253.3 seq=768 len=32 OK
ICMP echo: 3 sent, 3 received, 0 errors

PING v0.18
Enter Command line: (Example, ping 192.168.253.3)
```

重新出现 Enter Command line: (Example, ping 192.168.253.3)时，输入 ping 并回车。再到 PC 机端，点击[开始 | 运行]在命令框中输入“ping 192.168.253.2”回车。如果一切顺利，超级终端上应出现如下显示：



```
arSYS - 超级终端
文件(F) 编辑(E) 查看(V) 拨入(C) 传送(T) 帮助(O)

PING v0.18
Enter Command line: (Example, ping 192.168.253.3)
ping

Ethernet card success to reset!
IP 192.168.253.2 mask 255.255.255.0 gate 192.168.253.3 Ethernet 01:12:34:56:6
Entered Server mode
Press EXINTn to exit
Rx1 \len 60 52:54:ab:25:d0:69 ARP 192.168.253.3 -> 192.168.253.2

Tx1 /len 46 52:54:ab:25:d0:69 ARP 192.168.253.2 -> 192.168.253.3
Rx1 \len 74 52:54:ab:25:d0:69 IP 192.168.253.3 -> 192.168.253.2 ICMP
Tx1 /len 74 52:54:ab:25:d0:69 IP 192.168.253.2 -> 192.168.253.3 ICMP
Rx1 \len 74 52:54:ab:25:d0:69 IP 192.168.253.3 -> 192.168.253.2 ICMP
Tx1 /len 74 52:54:ab:25:d0:69 IP 192.168.253.2 -> 192.168.253.3 ICMP
Rx1 \len 74 52:54:ab:25:d0:69 IP 192.168.253.3 -> 192.168.253.2 ICMP
Tx1 /len 74 52:54:ab:25:d0:69 IP 192.168.253.2 -> 192.168.253.3 ICMP
Rx1 \len 74 52:54:ab:25:d0:69 IP 192.168.253.3 -> 192.168.253.2 ICMP
Tx1 /len 74 52:54:ab:25:d0:69 IP 192.168.253.2 -> 192.168.253.3 ICMP
```

在 PC 机的控制台窗口中可以看到 3 次 ping 尝试都成功了。这样，就说明在开发板作为服务器和客户机的模式下，工作都是正常的。按下开发板上的 ExintX 键退出这个测试。

到此，我们已经进行完了 bootloader 所提供的功能部件的测试。

第七章 uCLinux 的使用说明

7.1 uCLinux 简介

uCLinux 是专门用于微控制领域的嵌入式 Linux 操作系统，它已被成功移植到多种平台上运行，它是针对没有 MMU 的处理器而设计的，uCLinux 符合 GNU/GPL 公约完全开放源代码，它采用 romfs 文件系统体积小精巧，深受广大微控制领域的用户喜爱，uCLinux 的官方网站：<http://www.uClinux.org>。

7.2 下载和运行 uCLinux

7.2.1 uCLinux 运行环境的建立

运行 uCLinux 之前要有程序来加载其内核，bootloader 就是完成在操作系统内核运行之前运行的一段小程序。通过这段小程序，我们可以初始化硬件设备、建立内存空间的映射图，从而将系统的软硬件环境带到一个合适的状态，以便为最终调用操作系统内核准备好正确的环境。ARMSYS44B0-C 开发板用的 BIOS 中已经包含了 bootloader 代码，在开发板 bootloader 运行后可直接将 uCLinux 内核下载运行，或者直接在开发板启动后输入命令“7”即可加载 uCLinux 到内存并运行。

编译好的 uCLinux 内核可以存放在 NorFlash 中。

7.2.2 uCLinux 内核的存储

如果要存储 uCLinux 内核到 NorFlash 中，ARMSYS44B0-C 开发板提供了烧写工具，使写入存储变简化些。烧写工具是前面提到过的 44bapp.bin 程序代码，它可以完成同时将开发板的 bootloader 代码、测试程序代码和 uCLinux 内核烧写到 NorFlash 中，具体步骤如下：

- (1) 将烧写工具 44bapp.bin 通过网口或串口下载到地址 0xC008000 中。
- (2) 将开发板配套的 bootloader.bin 代码文件下载到地址 0xC200000 起始位置。（作为 bootloader 使用）
- (3) 将编译好的 uCLinux 内核文件代码 imagerom.bin 下载到起始地址为 0xC220000 处。
- (5) 以上步骤顺利完成后就可以开始烧写了，在超级终端中输入运行命令“5”，这时程序会提示用户是否“运行指定地址中的程序”输入“y”，下面还会有提示判断“确认烧录吗？(y/n)”，输入“y”烧写工具开始运行，四个绿色用户指示灯会循环点亮，说明烧写程序还在进行，直到四个绿色发光管全部点亮或超级终端上显示烧写完成，这时说明烧写工具已经将 bootloader、测试程序和 uCLinux 内核烧写到了 NorFlash 中。然后按下复位按键复位开发板，只要 bootloader 顺利启动开发板，说明以上烧写基本完成，在超级终端中输入命令“7”即可启动 uCLinux，启动信息如下图所示：

将 uClinux-ARMSYS-20050101.tar.gz 拷贝到/home/ 下，运行解压命令：
“tar xvzf uClinux-ARMSYS-20051111.tar.gz”
解压结束后会在/home/ 下生成 uClinux-dist 目录。

7.3.2 安装编译环境

将 arm-elf-tools-20030314.sh 拷贝到根目录，运行安装：
sh arm-elf-tools-20030314.sh

7.3.3 配置和裁减 uClinux

下面开始配置 uClinux 的内核和用户选项。打开终端。

```
# cd /home/uClinux-dist (进入/home/uClinux-dist 目录下)
# make menuconfig (运行 uClinux 配置环境)
```

进入 uClinux 配置 (uClinux v3.1.0 Configuration) ，选中 “Kernel/Library/Defaults Selection ->” 敲空格进入。其中有两个选项需要选取：“内核设置和用户选项设置”：

```
[*] Customize Kernel Settings
[*] Customize Vendor/User Settings
```

选中这两项，按下 ESC 键退出，在询问是否保存时，选择 Yes 并回车。终端将首先进入内核配置选单。我们在配置 uClinux 内核时，就可以通过对这些选项的选择和取消选择来设定内核所具有的功能项。这也是裁减 uClinux 内核的基本方法。

按下 ESC 退出后将进入用户选项选单。用户选单中所列举的是一些常用工具和用户程序，我们自己编写的应用程序也可以放入用户选单。例如在选单中的：

```
My new application --->
```

中就有一个最简单的 Helloworld 应用程序。

用户选项中选中的内容，将交叉编译成压缩格式的可执行二进制文件，放入到 rom 文件系统中供用户调用运行。

了解这些选项之后，可以暂时不对这些选项进行任何修改。按 ESC 键退出配置界面。

7.3.4 编译 uClinux

在/opt/目录下，按下面的步骤对 uClinux 源码包进行编译：

```
make dep.....建立依赖关系
make clean (非必要) .....清除旧的编译结果
make lib_only .....编译库
make user_only.....编译用户程序
make romfs.....产生 rom 文件系统
make image.....产生映像文件
make.....编译内核
```

编译成功后，在 uClinux-dist/ 目录下将产生 images 目录，其中包含的 2 个文件：image.ram, image.rom 就是我们可以用来下载和烧录的映像文件。然后下载或烧录这些二

进制文件，并启动运行 uClinux。

7.4 uClinux 的应用开发

开发应用程序的简单例子：HelloWorld 和跑马灯实例，请查看《参考资料》中的 helloworld_armsys.pdf 文档。

7.4.1 开发模式

ARMSYS 上 uClinux 应用的开发目前支持的调试模式，一种是编译产生了二进制映像文件后，通过 BIOS 所带有的 TFTP 下载到目标板上运行观察运行结果，在开发应用程序部分时，可以采用 NFS，mount 上宿主机的硬盘的某个目录，运行其中的可执行程序。因为 mount 的宿主机硬盘上的应用程序会自动覆盖更新，再重新执行的就是更改后的新版本。这样反复调试、更改、编译再调试，而不必切换操作系统或下载、烧写板子，直至程序工作正常，是一种十分方便的方法。关于 NFS 的移植和使用的详细内容请参考文档：

基于 ARMSYS 的 uClinux 应用之一—NFS 开发环境的建立。

注意，uClinux-ARMSYS-20051111.tar.gz 中已经移植好 NFS 了，只需要在配置时添加即可。用户自己开发应用程序一定要放在 uClinux/user 目录下，否则，若放在其它目录下将有许多宏要自己定义，很是繁琐。

7.4.2 调试方法

直接在目标板上调试应用程序有两种办法：

(1) 打印串口：

这是嵌入式系统中最常用的调试手段，虽然简单但却有效实用。其实几种方法相比之下，最有效便捷的方法还是 printf。

(2) gdb 调试

目前暂不支持。用户也可以自己在板子上移植 gdbserver。

7.5 在 Linux 操作系统下程序代码的下载

使用 minicom 终端仿真程序

在 Linux 操作系统下，minicom 是一个友好易用的串口通信程序。通过它可以监视并控制串行口的信息。在终端下输入命令：“# minicom ” 启动该程序。minicom 的操作使用是基于窗口的。要弹出所需功能的窗口，可按 Ctrl+A 键，然后再按各功能键（a-z 或 A-Z）。先按 Ctrl+A 键，再按 z 键，将出现一个帮助窗口，它提供了所有命令的简述。按 Ctrl+A 键、O 键，就进入了 setup 菜单。改变其中的波特率为 115200bps，不要硬件或软件流控（8N1）。设置调制解调器的 init 和 reset 的项（用 minicom -s 命令，使用其提供的菜单就可以设置）为空项。确保 ARMSYS44B0-C 开发板已经通过串口同 PC 相连，上电开发板，就可在终端窗口中观测到开发板从串口发出的启动信息。如果没有显示请检查 minicom 有关设置是否正确。以上步骤正确完成后，一个终端仿真程序就可正常运行了，可以通过串口对开发板进行各种操作了。

下面就可在串口终端通过 XMODEM 方式下载程序代码至开发板了。具体操作是这样的：

1、在开发板上执行 XMODEM 下载命令“1”，选择“X”（地址可以改变）。

2、在 minicom 终端窗口按 Ctrl+A 键、S 键，就进入发送文件状态。

3、选择传输方式为 XMODEM，然后选择要发送的文件，确认后开始发送文件到串口，直到发送完毕。