ARMSYS 经济型嵌入式系统开发板

使用说明书 (Vol. 20041101)

杭州立宇泰电子有限公司

http:// www. Hzlitai.com.cn

目 录	2
前言	3
ARMSYS 开发板使用建议:	3
1. 硬件配置	3
1.1 主板配置说明	3
1. 2 JTAG 调试板	4
1. 3 附件	5
2. 光盘资料	5
3. ARMSys 的启动程序和地址空间分配	6
4. ARMSys 提供的开发环境	8
4. 1 JTAG 仿真调试	8
4.2串口动态观察	8
4.3 USB 下载器	9
5. 应用开发步骤	9
6. SDT2.5 的介绍	10
7. 利用 Helloworld.apj 工程来学习使用 ARMSYS	11
7.1 编写好源程序代码	11
7.2使用 APM 对工程进行编译	12
7.3 使用 ADW 进行仿真调试	
7. 4 在 Windows2000/XP 下的使用 JTAG.exe	23
7.5 USB 口下载工具	24
7.6代码固化	27
8. 利用 BIOS 测试开发板的各项功能	
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	
Step2: 打开电源	29
Step3: 建立和设置超级终端	29
Step4: 系统复位	
Step5: ARMSYS 功能部件测试	

目 录

欢迎使用杭州立泰电子有限公司的 ARMSYS 嵌入式系统开发板产品!

ARMSYS 开发板使用建议:

- 1. 了解 ARMSYS 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;
- 了解 ARMSYS 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7节有 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。
- 了解 ARMSYS 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。
- 可以采用配套的书籍配合学习。我们采用的配套的教材书名为《嵌入式系统设计与开发 实例详解》。其中有实例分析,和运行结果等,该书在 2004 年 11 月下旬出版。

1. 硬件配置

经济系统共具有2块电路板,包括主板、JTAG调试板。另外还配有多条电缆线。



1.1 主板配置说明

- 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 4KB 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
- 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
- 以太网接口:采用 10M 以太网控制器 RTL8019,提供标准的 RJ45 接口座,可接对等网线。

1. 2 JTAG 调试板



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:



图 1 主板与 JTAG 板连接图

1.3 附件

经济型中还提供了如下的附件:



2. 光盘资料

内容及说明如下图所示:



图 2 光盘资料树型图

其中 source 文件夹中包含 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。

3. ARMSys 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 ARMSys 的 BIOS 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——BIOS 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 BIOS 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。

GCS7	E000000H	
GCS6 (64Mb SDRAM) (C000000H-C7FFFFH)	сооооон	未用
GCS5	А000000Н	堆栈和中断矢量重构 (C7FFF00H-C7FFFFFH)
GCS4	8000000н	下载程序使用
GCS3	6000000н	(C000000H-C7FEB00H)
GCS2	4000000H	Ⅰ/0 设备空间
GCS1	2000000н	未用
特殊寄存器	1С00000н	BIOS 代码空间 (0H- XXXXXX H)
GCS0 (ROM)	0000000н	中断异常处理矢量表

(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 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 0x0000000,读写基地址(数据 空间的开始地址) -rw-base 设置为 0x0c000000,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,0xC000000~0xC7FEB00 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,这样程序空间定位在 SDRAM 中,程序就可以随时下载、更新了。这时,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 0x0c400000,只要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)。

4. ARMSys 提供的开发环境



在 ARMSys 平台下进行开发十分方便,如下图所示:

图 4 ARMSys 的开发环境

4. 1 JTAG 仿真调试

ARMSys 的主板上提供了标准的 20 针 JTAG 接口,在 JTAG 板上具有 JTAG 口转并口 电路。仿真调试软件主要采用 SDT2.5 中的 ADW (ARM Debugger for Windows)。只需用并 口线将 JTAG 板与主机的并口相连 (如图 4),同时在主机上运行 JTAG exe 这个协议转换软件,就能够在进行代码下载、仿真调试和 flash 固化等工作了 (具体操作指导参考 7.3 节)。

4.2串口动态观察

仅仅依靠通过调试器进行单步、断点等调试方法来观察程序的运行是比较局限的,因为 采用这些方法我们都要使程序停止下来,这在一些要求实时性的、连贯性的工作中是不能起 到有效的作用的。而通过在程序中加入串口输出,可以在程序动态运行过程观察程序运行情况。因此我们可以在程序中加入向串口发送信息的语句,采用串口线连接起主板上的 UART0 口和 PC 机的串口,并利用 Windows 操作系统自带的软件:超级终端来观察串口的输出。在 ARMSYS 的 BIOS 中,就加入了串口调试语句,因此你可以观察到主板启动时的情况。如 果配置好了超级终端,按下主板上的复位键,超级终端上出现如下图所示的信息。

🍓 armsys - 超级终端	×
文件 (E) 編輯 (E) 查看 (V) 呼叫 (C) 传送 (E) 帮助 (B) D 🚅 📨 🌋 💷 🎦 😭	
OEM name : 杭州立泰电子 Function : ARMSys's BootLoader Email : rao_dali@263.net UART配置 : 115.2kbps,8Bit,NP,UARTO 采用USB : 是 Nand-flash : K9F2808 初始化系统	
已连接 2:53:44 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印	4

图 5 超级终端使用示例图

4. 3 USB 下载器

在图 5 中,我们可以看到最后一行的使用 USB 下载器吗? (Y/N) 这一行询问用户是 否开始 USB 下载。

简易 JTAG 口下载的速度通常是比较慢的,在调试一个较大的程序时,我们常常为了程序的下载要等待很长时间。因此 ARMSys 特别提供一个快速的 USB 口下载器,可以将编译 生成的二进制代码通过 USB 口下载到系统 SDRAM 中运行。具体使用指导参考下面 7.5 节详细说明。

如果你不想使用 USB 口,则输入 N,启动程序将进入开发板的功能测试项目(参考第 8 节)。

5. 应用开发步骤

从零开始设计一个系统通常是很花费时间和精力的;而 ARMSYS 平台为使用者提供了 成熟的、开放式的软、硬件的资源,并提供界面友好、功能强大的开发工具环境。这些平台 资源,不仅可以用于教学、实验活动,还能够方便地用于各种嵌入式系统科研、应用开发、 产品二次开发等用途。

在基于平台环境的应用开发中,通常需要做的就是利用平台资源,在平台之上编写用户 代码,下载到 ARMSYS 上仿真运行。经过反复地调试除错,最终实现各种应用目标。归纳 起来,在 ARMSYS 上进行开发可按以下步骤进行:

Step1: 了解 ARMSys 硬件电路结构和系统基本配置;

Step2: 阅读实例代码,了解各种软件包功能和结构;

- Step3: 自行编写代码或者修改实例代码,用编译器对代码进行初步除错;
- Step4: 通过 JTAG 口将可执行映像文件下载到目标系统,进行仿真调试,观察目标系统的运行情况,不断改善源代码,直到满足要求(或者通过 USB 下载器将代码下载到开发板上,观察运行情况)。
- Step5: 最后,利用代码固化工具,将编译生成的二进制代码固化到 flash 中,使系统能够在脱离主机的情况下独立运行。

6. SDT2.5 的介绍

SDT2.5 是一个使用方便的集成开发环境,集成开发环境实际上是一种工具软件,这种工具软件一般由处理器或处理器内核厂商提供。例如在本系统中采用了 S3C44B0X,它采用 了 ARM7TDMI 内核;因此我们采用了 ARM 公司提供的 SDT (ARM Software Development Toolkit)来完成代码编译的工作。

以下就我们对 SDT2.5 进行一些介绍。

SDT 工具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 SDT 来开发、编译、调试采用包括 C、C++和 ARM 汇编语言编写的程序。

SDT 主要包括两个视窗开发工具:

ADW 它为使用者提供了一个完全的 Windows 视窗环境来进行 C、C++、和 ARM 汇编 语言源程序的调试;

APM ARM 项目管理器。这是一个用户图形界面的项目管理工具,用来管理源文件, 并且集成了编译、链接器,可以直接在 APM 中同时编译、链接你的项目文件。 APM 在 build 命令中集成了命令行工具中的编译和链接命令。

这里所说的命令行工具有:

armcc ARM 的 C 语言编译器。它能够将 ANSI 或 PCC 源程序转换为 32 位的 ARM 代 码。

tcc Thumb C 编译器。它能够将 ANSI 或 PCC 源程序转换位 16 位的 Thumb 代码。

armasm ARM 和 Thumb 的汇编器。能够汇编包括 ARM 和 Thumb 的汇编程序

armlink ARM 的链接器。它将一个或多个 object 文件和选择的部分 object 库进行链接, 产生可执行映像文件,也就是 ELF 格式的文件(参考下面对 fromELF 的说明)。

armsd ARM 和 Thumb 的符号调试器。使 SDT 可以进行源代码级的调试。

SDT 还提供一些实用工具,例如:

FromELF ARM 文件转换工具。它将 ELF 格式的输入文件转换为各种格式的输出文件, 包括 AIF 格式,二进制代码格式,IHF 格式,Motorola32 位 S 记录格式,Intel Hex 32 格式。

••••

SDT 还支持软件仿真调试,主要有两个功能部件:

ARMulator 软件模拟调试器;

Angel ARM 调试监控器。运行在基于 ARM 处理器的目标系统之上,与主机通 讯进行调试。

7. 利用 Helloworld.apj 工程来学习使用 ARMSYS

本实验利用光盘中 source\Helloworld\下的源代码进行实验,要求完成以下工作:

- 1) 利用工程模板 source\template.apj, 在 SDT2.5 的 APM (项目管理器)中建立新的工程, 在工程中加入 source\Helloworld\Target 目录中提供的文件;
- 2) 编写主程序文件 main.c, 并将文件加入到工程中;
- 3) 正确设置编译器的编译选项,并对工程进行编译、除错,最终产生可执行的映像文件 (*.axf)和二进制代码(*.bin)文件;
- 4) 打开超级终端,正确配置串口参数;
- 5) 采用 ADW (ARM 视窗调试器)通过 JTAG 模块下载可执行程序,并仿真调试,观察实验现象;
- 6) 采用 ARMSys 提供的 USB 下载器下载二进制代码,并观察运行情况。

将 source 目录整个拷贝到硬盘中,例如拷贝到 D:\source 处,方便进行实验。

首先,安装 SDT2.5,在光盘的\开发工具\集成开发环境\Windows\目录下有 SDT2.5 的安装文件,双击 setup.exe 进行安装,具体的安装过程这里就不赘述了。

7.1编写好源程序代码

用编辑器打开 source\HelloWorld\main.C 文件,读懂其中的代码。其中 main()函数的定义是: void Main(void)

{

```
char aa;
```

```
Uart Init(0,115200);//初始化 UARTO 口,设置波特率为 115200bps
Led Display(0xf); //点亮绿色发光二极管
Uart Select(0);
               //选中 UARTO
                //点响蜂鸣器
Beep(0x1);
//向串口输出字符串
Beep(0x0);//
Uart Printf("\n*
                立泰电子
                                                *");
Uart_Printf("\n* -Hello World!-
                                                 *");
Uart Printf("\n* Version 1.10
                                                 *");
Uart Printf("\n* Email:rao dali@263.net
                                                *");
Uart Printf("\n* UART Config--COM:115.2kbps,8Bit,NP,UART0 *");
Uart Printf("\n*Begin to Study Embedded System, OK?(Y/N) --- *");
Led Display(0x0);
aa = Uart Getch();//等待并从串口获得一个字符
if((aa=='Y')||(aa=='y'))
Uart Printf("\nGood!See you next time!");
else
Uart Printf("\nByeBye!");
```

}

```
这是一个简单的例子,因此代码比较简单。这里你可以尝试修改一下代码,将:
Uart_Printf("\nGood!See you next time!");
修改为:
Uart_Printf("\n好!我们马上开始学习嵌入式系统!");
然后保存文件。
除了 main.C 文件以外,我们还要具备几个必不可少的源文件:初始化程序 44binit.S,
```

和它引用的 Memcfg.a 和 Option.a,完成 I/O 设备初始化和定义 Uart_XXXX 函数的库文件 44blib.alf(或源文件 44blib.c)和头文件 44blib.h。这些文件几乎在以后所有的工程中都要使 用到。

7.2 使用 APM 对工程进行编译

首先我们学习如何使用 SDT 中的 APM——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(例如汇编或 C)、不同的类型(源 文件,或库文件)的文件。APM 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",并加入源文件、库文件。

7.2.1 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 ARMSys 上使用的通用工程模板——template.apj。 点击 APM 菜单[File | open...],找到 source\template.apj,选中并打开。依次点击 rom 和 Debug 前面的+号,展开后如下图所示:



图 6 模板工程展开树型图

点击[File | Save as...],将它另存为: source\helloworld(或者是自定义的某个目录)\ MyHelloworld.apj。

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:

选择 File 菜单下的 new 选项,或直接按下 ,出现以下对话框:



图 7 新建对话框

- 2). 选中"Project"选项。
- 3). 点击"OK"则出现以下对话框:

Template Description	OK
Build and debug an ARM executable image made from C and ARM Assembly	Cancel
Language source files and ARM Object Libraries. The DebugRel	Help
build your object libraries using	
sub-projects.	
sub-projects. [ype: ARM Executable Image	
sub-projects. [ype: ARM Executable Image	
sub-projects. Cype: ARM Executable Image roject Name: roject1	
sub-projects. Type: ARM Executable Image oject Name: roject1 oject Directory:	

图 8 新建工程对话框

在这里我们选择 Type 的第一项: **ARM Executable Image**,在下面的编辑框中分别输入工程名称和路径。完成后点击"OK"。出现如下图示,工程建立完成:



图 9 新建工程树型图

实例中仅使用 Debug(调试)变量,它包含了调试信息。因此,分别点选 DebugRel 变量和 Release 变量,按下"Delete"键,将这两个变量删除。

7.2.2在工程中添加源文件

在图 7 的对话框中,点选源文件的类型,点击确定, APM 就会为你新建一个源文件,并

开始编辑该空文件。但 APM 并不是一个很好的源代码编辑器,因此,大多数时候,我们采用 用其它的代码编辑器(例如 UtraEdit、SourceInsight)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,在工程展开树型图中,点选"source" 项,点击[Project | Add Files to Source…]项,从目录中选取 main.c文件 (source\helloworld\main.c),点击"打开",main.c文件就被加入了工程中。



图 10 工程树型图

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

APM 用图标来表示源程序 (source 和 include file) 文件的几种类型:

.C	C 或 C++源文件;
s	ARM 汇编语言文件;
.h]	头文件;
T.	子工程(sub-project)
— 3	

未知类型文件

导出文件的图标和类型:

- 唐文件
- 🔜 🛛 🔤 ARM 执行文件

符号 : 🤲 说明该文件还未成功产生。

Target 目录下还有一个 44blib. alf 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 44blib.h 进行了声明。这个文件也必须添加到工程中。点选树型图中的 Libraries 项目,再点击菜单[Project | Add Files to Libraries…]项,将 44blib.alf 文件添加到工程中。所有必须的文件添加完成后的树型图如图 10 所示。

7.2.3 编译选项设置

在进行编译之前,你必须正确设置该工程的工具配置选项。因为我们在建立工程时,采 用的是调入工程模板,有些选项已经在模板中保存了下来,因此可以不再进行设置。但我们 仍然需要确认几个重要选项的设置。

在工程的树形图中,点选"Debug"项,然后点击 Project 菜单,鼠标移至下拉菜单中的 Tool Configuration for "Debug"项目,其中有三个支项:

- <cc> = armcc C语言编译器设置
- <asm> = armasm ARM 汇编器设置
- O armlink 链接器设置

对 armcc 和 armasm 保持默认设置,点击 armlink 项目(链接器设置),对其中的几个选 项页进行设置:

(1) 设置 General 选项页。按照下图设置 General 选项页,主要的是设置好 General 页中 库文件的路径,使它指向 SDT 安装目录中的 1 ib 目录,例如 C:\ARM251\LIB.

inker Configuration		
General Listings Output	Entry and Base ImageLayo	1
General		ï
 ✓ Include debugging in ✓ Give progress inform ✓ Search standard libration 	format ation while	
-Give Information on Totals T Sizes T	Interwork 🦵 Unused	
-Search path for librarie	25	
D:\ARM251\LIB		100 00 000
Equivalent Command Line	1	
-info total -ro-base Ox OxOc5f0000 -first 44bin	Oc200000 -rw-base it.o(Init)	

(2) 选择 Entry and Base 选项页。在 Read-Only 下面的编辑框中,填入: 0x0c000000, Read-Write 下面的编辑框中,填入: 0x0c5f0000(在仿真调试情况下,将程序空间和变量 空间都映射到 SDRAM 上)。此时,在 Equivalent Command Line 框中会自动增加-ro-base 0x0c000000 - rw-base 0x0c5f0000的字符串。

Entry Poi	nt			
	Base of Read-O OxOcO	f Image		
	Read-W	rit. £0000		
-info to 0x0c5f000	t Command Line tal -ro-base O DO -first 44bi	: xOc000000 -r nit.o(Init)	w-base	

图 11 armlink 设置对话框

Listings	Output	Entry and Base	ImageLayout	Areas 4
Place	at begin	ning of image —]
Object 44bis	t File	Are	ea Name	

图 12 ImageLayout 选项

(3) 选择 ImageLayout 选项页。在"Place at beginning of image"中的"Object File" 中填写: 44binit.o; 44binit.o是由系统初始化程序 44binit.s汇编产生的 object 对象文 件。将 44binit 放在整个工程 image 的最前面,使得系统刚开始运行时首先进行系统初始化。 这一步是必须的。然后在 Area name 下填入"Init"即可。在 Equivalent Command Line 框中会增加-first 44binit.o(Init)字符串。

其它选项页保持缺省状态,不做修改。以上设置进行完毕以后,就可以进行工程的编译 了。

7.2.4 进行编译和链接

- □ 选中工程的树型视窗中的一个 source 文件,这时可以点击 按钮,对文件 进行编译 (compile);
- □ 点击 按钮,对工程进行 build, build 的行为是由工程模板中定义的 build 步骤确定的。典型的 build 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:
- 编译子工程。

接钮,对工程进行强制 build,强制 build 与 build 的区别在于, build □ 点击 只针对上一次 build 之后修改的内容, 而 force build 却无论是否修改, 将工程中所有文件 都再 build 一次。

按下 build 或 force build 按钮后,在工程树型窗体下面的 build 日志视窗内, 会显示 出 build 的过程和最终结果。APM 用一些符号表示诊断信息的类型:

i Information: 编译或链接信息; Warning:警告: Error: 错误; 严重错误: 致命错误。

工程成功编译后的树型视窗变为:



图 13 编译成功的树型图

Build 日志视窗(在树型视窗下面)内显示出编译过程信息:

```
----- Build Debug variant (Force Build) ------
 armasm -o E:\Helloworld\Debug\44binit.o -MD- -IE:\Helloworld\Target
         E:\Helloworld\Target\44binit.s
 armcc -o E:\Helloworld\Debug\Main.o -c -MD- E:\Helloworld\Main.c
 armlink -o E:\Helloworld\Debug\helloworld.axf E:\Helloworld\Debug\44binit.o
         E:\Helloworld\Debug\Main.o E:\Helloworld\Target\44blib.alf
 fromelf -nozeropad E:\Helloworld\Debug\helloworld.axf -bin E:\Helloworld\Debug\helloworld.bin
 Translation to BIN format successful.
i Information: "E:\Helloworld\helloworld.apj"; Project is up to date
 O Warnings, O Errors
```

图 14 编译成功的日志视窗

从中可以看出,源程序经过 armasm 汇编、armcc 编译和 armlink 链接之后,最终产生 了 helloworld.axf 文件,也就是 ARM 的 ELF 可执行映像文件;这个文件可以载入 ADW 进行 仿真调试了。最后通过 fromelf 工具将 ELF 文件转换为二进制格式文件 helloworld.bin。 它可以用来最终固化到 ROM 中(但链接选项中的一ro-base 要修改),也可以通过 USB 口下 载运行。它可以用来最终固化到 ROM 中(但链接选项中的一ro-base 要修改),也可以通过 USB 口下载运行。

7.3 使用 ADW 进行仿真调试

7.3.1硬件准备

如图 4 所示,在调试之前,我们先用并口电缆将 PC 机并口和 JTAG 调试模块连接起来,用串口线将 PC 机串口和主板的 UART0 口连接起来(当然还要将主板和 JTAG 板连接起来)。然后,就可以上电了。如下图所示,ARMSys 开发板有两种供电的方式:

- USB □取电;
- 外部 5~8V 直流电源输入;

如果使用 USB 口取电,通过主板上的拨动开关拨到 USB 档;如果使用外部直流电源,则将拨动开关拨到 EXT 档。这里的外部直流电源是指外部的+5~8V 直流稳压电源,稳压电源插头的极性为外圈负,内圈正。

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,这说明主板启动正常。此时 JTAG 模块上只有指示电源的红灯点亮,说明并口已经连接好了。

7.3.2 使用 UART 串口和超级终端进行系统调试

在 Windows 操作系统下,点击[开始 | 程序 | 附件|通讯 | 超级终端]。新建一个超级终端项目,将其命名为 ARMSYS,点击"属性"图标,出现以下对话框:

armsys	Lixeexternery	<u>Adaman ing s</u>
国家(地区) 代码(C):	中国 (86)	-
输入不带长途	前缀的区号。	
区号(E):	0571	
电话号码(P):	Γ	
连接时使 用 00:	直接连接到串口 1	•
	配置 (C)	
▼ 使用国家(地区)代码和区号(II)	

图 15 超级终端属性

在属性对话框的"连接时使用"项中选好你所使用的串口,点击"配置"按钮。按照下 图配置该串口:

:0∎1 届性	?)
端口设置	
波特率(B):	115200
数据位 (0):	8
奇偶校检 (P):	元 👤
停止位 (S):	1
流量控制(F):	无
高级(ム)	还原默认值 (R)
确定	取消 应用 (A)

图 16 串口属性配置

点击确定,超级终端就配置好了。

7.3.3 运行 JTAG.exe 程序

将光盘中的 Tools\JTAG\目录拷贝至硬盘某个目录下,其中的 JTAG exe 是一个协议转换 程序,在运行 ADW 调试器之前必须首先运行它。在 Win98/ME 下,双击 JTAG exe 图标运行。如果在 Win2000/XP 下,JTAG exe 需要协同一个工具一起工作,具体配置过程请跳到 7.4 节查看。注意,在 ADW 调试器在线仿真期间,不要关闭 JTAG exe !

在主机上启动 JTAG.exe 之后,即可进入仿真调试阶段。

7.3.4调试器设置

在 APM 中,工程经过编译成功,产生了*.axf 文件之后,就可以开始调试了。点击 经 按钮,进入了 ARM Debugger 视窗界面。点击菜单项 [Option | Configure Debugger...],对 调试器进行配置:

gger Lonfiguration	
rget Debugger Memory Ma	ıps
Target Environment	
ARMulate	•
ARMulate	
Lemote_A	Configure
? Use the ARM Debugger	with the 'ARMulator'
to execute ARM progr ARM hardware, by sim	nator. Inis allows you rams without physical mulating the ARM

图 17 调试器设置对话框

在 Target Environment 栏中选择 "Remote_A" 选项。

如果选择"ARMulate"则 Debugger 将在不与目标硬件连接的情况下,进行软件模拟调试。

然后点击下面的"Configure…"按钮。 出现以下对话框:

Density Converting	
C Serial C Serial C Serial / Parallel	Heartbeat Disabling heartbeat will disable hos timeout and packet resend.
• Ethernet 192	168.253. 3
Ports Serial: COM1 Parallel: LPT1	Serial Line Speed Baud Rate:
Channel Viewers	
ThumbCV.dll	Add Remove

图 18 远程调试器设置

在 Remote Connection 一栏中,我们采用网络通讯的方式,在 IP 地址栏中填入本机的 IP, 或者填入 127.0.0.1。

在 Heartbeat 前面打上钩。点选该选项,使 ADW 将通过发送 heartbeat 信号来保证传输的可靠性,用以防止主机与目标系统之间,由于同时等待数据包而发生死锁的情况。

设置完毕点击 OK。点击"确定"退出 Debugger Configuration 的设置。

此时 ADW 开始初始化目标系统,下面是有可能出现的错误消息框,如果查找目标系统 失败,会弹出:



图 19 连接错误对话框

这是通常是由于 JTAG.exe 还没有运行,或并口线没有连接好的原故。运行 JTAG.EXE,或者重新连接好主机与目标系统之间的并口线。点击上面的"确定"按钮,弹出:

Initial	lise Error			×
	The Debugger failed image: "E:\example\Debug\j	d to initialise project2.axf″	RDI target "Remote_/	A" with
	Retry	Configure	Quit Nehuzzer	

图 20 重试对话框

点击 "Retry" 按钮,如果目标系统正确链接了,会看到程序下载的进度的显示。进度消息 框消失后,显示当前执行代码视窗:

👗 ARII – E	xecuting 44binit. s 📃 🗖 🔀	这时
76	MEND	
77 78 79 80 81 82	IMPORT Image\$\$RO\$\$Limit ; I IMPORT Image\$\$RW\$\$Base ; I IMPORT Image\$\$ZI\$\$Base ; I IMPORT Image\$\$ZI\$\$Limit ;	试进 立 "Re
83	IMPORT Main ; The main entry of	而没
85	AREA Init, CODE, READONLY	说明
86 87	ENTRY	以进
88 99 91 92 93 94 95 96 97 98	<pre>b ResetHandler ;for debug b HandlerUndef ;handlerUndef b HandlerSWI ;SWI interrupt hand b HandlerPabort ;handlerPAbort b HandlerDabort ;handlerDAbort b . ;handlerReserve b HandlerIRQ b HandlerFIQ ;***IMPORTANT NOTE*** ;If the H/W vectored interrutp ; b changed like below to work</pre>	
99	; be changed like below, to work · h HandlerTRO → subs no lr	图 /

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。

图 21 当前执行代码视窗

7.3.5 调试器的使用





图 23 放置断点

👗 ABI	– Executing Nain. c 📃 🗖 🔀
20	void Main(void)
21 22	{ char aa;
23	Vart_Init(0,115200);
25	Uart_Select(0); //Select UAR
27 28	Beep(Ux1); Uart_Printf("\n************************************
29 30	Beep(0x0); Uart_Printf("\n*
31 32	Uart_Printf("\n*

图 24 进入主函数运行

3.3.6 观察窗口

ADW 提供了许多有用的观察窗口,点击菜单项中的 View,可以从它的下拉菜单项中了解可观察的项目。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、通过表达式来查看变量的值; Source Files: 源文件索引;

Memory...: 查看存储器内容。输入地址,即可查看这个地址上存储单元的值。

3.3.7 全速运行

在 ADW 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。

🗞 arnsys - 超级终端		
文件(F) 编辑(E) 查看	(V) 拨入(C) 传送(T) 帮助(H)	
02 28 0		
*****	* * * * * * * * * * * * * * * * * * * *	* * * * * * * * * * * * * * * * * * * *
*	立泰电子工作室	
*	-Hello World!-	*
*	Version 1.10	*
*	Email:rao dali@263.net	*
*	ART ConfigCOM: 115.2kbps,8Bit,NP,UART	0 *
*	Begin to Study Embedded System, OK? ()	Y/N)*
Good!See you no	xt time!_	
4		<u>•</u>
三注接 0:03:03 ANSI	115200 8-N-1 SCROLL CAPS NUM 捕 打印]

图 25 HelloWorld运行后超级终端的显示

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 文件的的 Main() 处程序开始运行,如图 24。

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。

读者可以继续进行一些单步操 作,了解每条语句的作用。 在 "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!"。如果你按照 7.1 节修改了源程序,则应当出现"好!我们 马上开始学习嵌入式系统!"字符串。

7. 4 在 Windows2000/XP 下的使用 JTAG.exe

JTAG.exe 是一个控制台程序,它直接对计算机的并口进行操作。只能在 Windows98 下运行,无法直接在 Win2000 和 XP 下运行。要使它能够在 Windows2000 及 XP 下运行需要 Porttalk 工具的配合。按照以下步骤进行安装(注意:这里以 Windows2000 为例,系统目录 为 Winnt,如果是 WindowsXP 下则系统目录为 Windows):

1) 将光盘中\JTAG\目录的内容全部拷贝到 C:\下(本说明以 C:\JTAG 为例进行说明,你也可以拷贝到别的目录)。

2) 该目录中的 giveio.sys 和 porttalk.sys 文件拷贝到 c:\winnt\system32\drivers\ 下,双击 loaddrv.exe,出现以下界面:

:\windows\s	ystem32\drive	rs\	
Install	Start	Stop	Remove
atus:			

程序启动后,编辑框中的路径是"c:\windows\system32\drivers\",把它改为:C:\winnt\system32\drivers\giveio.sys。

3) 点击 Install 按钮, 状态显示: "Operation was successful"

c:\winnt\syste	m32\drivers\c	iveio.sys	
Install	Start	Stop	Remove
Status: One	eration was su	ccessful	

4) 再执行Start, 同样看到"Operation was successful", 再次执行则会看到"Service already running"

c:\winnt\sush	em32\drivers\c		
en ministre region	onioz iani olo ig	gi t ele. ey e	B
Install	Start	Stop	Remove
Status: Se	rvice already ru	Inning	

5) 点击"OK"退出loaddrv.exe。

6)如果需要确认,可以进入设备管理器查看,右击"我的电脑",点击"属性",点击"硬件"标签页;点击"设备管理器"按钮,再点击菜单项中[查看 | 显示隐藏设备],然后在"非既插即用驱动程序"中可以看到giveio 项。如果不想每次计算机启动后都运行LOADDRV.EXE,将giveio 改为"请求"运行。

giveio 尾性				? ×
常规 驱动程	序			
🔷 givei	0			
服务领	3:	giveio		
显示很	3称(11):	giveio		
- 当前状态 - 状态: E - 启动	己启动		日动②	₱止(I)
类型(匠):	[
	循來 系仍 已停用		JCANETTFE	паже (27
			确定	取消

7) 双击本目录下的porttalk.reg,将porttalk的信息加入注册表。

8) 在"开始" → "运行"中输入: c:\jtag\allowio jtag.exe /a, jtag.exe 就 开始运行了。这样就可以在 Win2000 下进行 ADW 的调试和 flash 烧录等操作了。我们在该 目录下放置了一个*.BAT 批处理文件,可以将它的快捷方式放到桌面上,以后只要双击它就 可以启动 JTAG.exe 程序了。

7.5 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 usbinstall 目录, 整个拷贝到 C:\下(注意必须是 C:\);
- 步骤 2,双击 C:\usbinstall\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:

1	armsys - 超级终端	
х Г	[件 (E) 编辑 (E) 查看 (Y) 呼叫 (C) 传送 (E) 帮助 (H)	
	○ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	^
-< こ	」 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	>

按下键盘上的'Y'键,会提示你输入地址,如下图所示:

🧐 armsys - 超级终端	
文件 (E) 编辑 (E) 查看 (Y) 呼叫 (C) 传送 (E) 帮助 (H)	
D 🚅 🍘 🕉 🗈 🖰 😭	
OEM name : 杭州立泰电子 Function : ARMSys's BootLoader Email : rao_dali@263.net UART配置 : 115.2kbps,8Bit,NP,UARTO 采用USB : 是 Nand-flash : K9F2808 初始化系统	• • • •
▲ W A A A A A A A A A A A A A A A A A A	>

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 (E) 编辑 (E) 查看 (Y) 呼叫 (E) 传送 (E) 帮助 (H)	
UART配置 : 115.2kbps,8Bit,NP,UART0 采用USB : 是 Nand-flash : K9F2808 初始化系统 : · · · · · · · · · · · · · · · · · · ·	 الج
	~
已连接 3:17:32 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印	

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:

SBTool		_ 🗆 🗙
文件(2) 帮助(2)		
下载文件] 传输状况:	文件名 后缀: 文件长度	
	0%	

图 26 USB 下载工具界面

● 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。

📲 USBTool		- 🗆 ×
文件(Y) 帮助(Z)		
	文件名	helloworld.bin
	后缀,	bin
下载文件	文件长度	27416
传输状况:		
Send command to tran Send command to tran Trans the file to buffer Transferreing filePle. Compeleted! Transferr	ieramekilielen. nsfer the whole file. ase wait ed 27416 bytes	
	100%	

图 27 下载成功界面

7.6代码固化

请注意,将工程编译为烧入二进制代码时,需要将链接器重新设置,将调试时程序空间 定位地址(Read-Only地址)由 0x0c000000 改为 0x00000000,如下图所示。

General	Listings	Output	Entry	and Base	ImageLayo 🔺
Entry	Point				
	[Base of	Image -		
		Read-Or	վյ		
		0x0000	0000		
		Read-Wr	rito		
		0x0c00	0000		
-Fauire	lent Comm	and Line	-	19	

图 28 Entry and Base 设置页

可以使用 fluted.exe 烧录 flash,它同样要求主机上运行一个 JTAG.EXE。在 Win2000 下,则运行 allowio "fluted 命令行" /a 即可,在目录下也有批处理文件可以直接双击运行。Fluted 的工具放在光盘\开发工具\programmer\下。Fluted 的缺点是烧写速度很慢。

在光盘的开发工具\programmer 目录有一个 flashprgm 程序,该程序可以在目标系统已经 具有烧录启动代码的情况下,实现 flash 程序的快速覆盖烧录。Flashprgm 的使用方法请参考 同一目录下的使用说明。

请使用光盘目录下\开发工具\programmer下的 44bapp.bin 文件进行 flash 的烧录。请不要直接使用实例代码编译出来的*.bin 文件进行烧录,防止目标系统不能得到正确的初始化。

8. 利用 BIOS 测试开发板的各项功能

ARMSYS 开发板提供的 BIOS 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。BIOS 可以方便的实现:

1. 开发板的硬件测试

2. USB 下载

关于 USB 下载功能的使用请查看 7.5 节。本节仅对开发板的测试功能的相关操作进行 说明。

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 UARTO 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 (<500mA),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。

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: 建立和设置超级终端



点击[开始 | 程序 | 附件 | 通讯 | 超级终端]如下图。

设置如下图:

COTE4 属性	?×
端口设置	
	~
毎秒位数 (B): 115200 💌	
数据位 @): 8	
奇偶校验 (P): 无 💙	
停止位 (2): 1	
数据流控制 (E): 无	
还原为默认值 ®)	
一 确定 取消 应	Ħ(<u>k</u>)

Step4: 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"N"键,超级终端输出如下图的信息:

🧞 armsys - 超级终端 📃	
文件 (2)编辑 (2) 查看 (2) 呼叫 (2) 传送 (2) 帮助 (2) □ 22 22 23 ⊪□ 12 13 23 	
EM name : 杭州立泰电子	
unction : ARMSys's BootLoader	
mall : rao_dall0263.net 和田町安 · 115_2kbpg Apit ND UADWO	
ARTALLE : IIS.2KDpS,0BIC,MP,0ARTO	
and-flash : K9F2808	
7始化系统	
史用USB下载器吗?(Y/N)	
SB按口木被激活!	
在PC机键盘上按下字母,选择对应功能项:	
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-切尼台的行 视机	
–	
	~
C在接 0:08:06 自动检测 115200 8-N-1 SCROLL CAPS NUM 捕 打印	.:

Step5: ARMSYS 功能部件测试

按下计算机 "C"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:

🍓 armsys - 超级终端	
文件 (2) 编辑 (2) 查看 (Y) 呼叫 (2) 传送 (2) 帮助 (H)	
C(c)>>ARMSYS功能部件测试	^
保存在Nandflash中的内容都将被擦除,你确定吗?(Y/N) 格式化Nand-flash成功!	
在PC机键盘上按下字母,选择对应功能项: 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	:)
 USB接口失效。	
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?	~
	>

(1) Nandflash-Read ID 的测试

因经济型开发板不配置 Nandflash,因此这里省略该项测试内容的说明。

(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:

🧠 arı	isys	- 超	级终端	5														_	
文件 @)编辑	₿ (E)	查看(<u>v</u>) P	የጣ (C)	传送	(I)	帮助()	Ð										
🗅 🖻		3	•D 2) 🗈	7														
Se	lect	t tł	ne i	fund	ctic	on t	:o 1	test	:22										~
																			_
10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f				
20	21	22	23	24	25	26	27	28	29	2a	2b	2C	2d	2e	21 25				
30	31 41	32	33 12	34 44	35 45	36 46	31	38 10	39	Ja An	3D 4h	3C Aa	3a 4a	Je ∕∖o	JI Af				
50	41 51	42 52	40 52	44 57	40 55	40 56	47	40 58	49 59	4a 55	4D 5b	40 50	4u 5d	40 50	41 5f				
60	61	62	63	54 64	55	50	67	50	69	Jа Ба	Бр Бр	БС БС	6d	Jе Бе	51 6f				
70	71	72	73	74	75	76	77	78	79	7a	7b	7c	7d	7e	7f				
80	81	82	83	84	85	86	87	88	89	8a	8b	8c	8d	8e	8f				
90	91	92	93	94	95	96	97	98	99	9a	9b	9c	9d	9e	9f				
a0	a1	a2	a3	a4	a5	аб	a7	a8	a9	aa	ab	ac	ad	ae	af				
b0	b1	b2	b3	b4	b5	bб	b7	b8	b9	ba	bb	bc	bd	be	bf				
C0	с1	c2	с3	с4	с5	сб	с7	с8	с9	са	сb	CC	cd	се	cf				
d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	da	db	dc	dd	de	df				
e0	e1	e2	e3	e4	e5 Er	e6 £C	e7	e8	e9	ea E-	eb	ec	ed	ee £-	ei				=
ΙIU	ΤI	IΖ	ЦЗ	Ι4	Ι5	IЮ	I/	Ι8	Ι9	га	αı	IC	Ια	Iе	ΓI				
1	:Nat	ndf ⁻	last	h−Re	bee	тр	2:	ITC	(24)	l C X 3	ر) – ر	Test	-						
	- 14 - 14		Lan		Juu	10	.		(•/ •		-						~
<]					1. 400.00			_	_	_	_	>
已连接	0:24:4	18 自动	检测	1152	200 8-	N-1	SCROL	L	CAPS	NUM	捕	打印	,						

(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:

```
🌯 arasys - 超级终端
                                                           文件(E) 编辑(E) 查看(Y) 呼叫(C) 传送(T) 帮助(H)
D 🗃 💮 🔏 🗈 🛅 😭
  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?3
 [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
  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?
                                                               >
                          CAPS NUM 捕 打印
已连接 0:28:0€ 自动检测 115200 8-N-1 SCROLL
```

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 (2) 编辑 (2) 查看 (Y) 呼叫 (2) 传送 (1) 帮助 (H)	
D 🖻 🛞 🔏 🗈 🎦 🖆	
F(f)>>Nandflash格式化(FAT16) R(r)>>运行已经保存为文件的程序(演示程 C(c)>>ARMSYS功能部件测试	序)
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!!! _	
	Notes that the second secon
已连接 0:34:55 自动检测 115200 8-N-1 SCROLL CAPS NUM 猫 打印	

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:

```
🧐 arnsys - 超级终端
                                                            文件 (P) 编辑 (B) 查看 (Y) 呼叫 (C) 传送 (E) 帮助 (H)
0 🗃 🚳 🖏 👘
                                                                   ~
  1:Nandflash-Read ID 2:IIC(24LCxx)-Test
  3:RTC-Display 4:Keyboard-Test
                     6:ADC-Test
  5:Ext.Int-Test
  7:Timer.Int-Test 8:LCD-Test
  9:TouchScreen-Test 10:Ping diagnose tool
 Select the function to test?6
 The ADC IN are adjusted to the following values.
 Push any key to exit!!!
 ADC conv. freq.=7812(Hz)
 0:0495 1:0479 2:0479
 0:0512 1:0511 2:0511
 0:0511 1:0495 2:0511
 0:0512 1:0511 2:0511
 0:0512 1:0511 2:0511
 0:0512 1:0511 2:0511
                                                                  >
                           CAPS NUM 捕 打印
已连接 0:46:58 自动检测 115200 8-N-1 SCROLL
```

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 (E) 编辑 (E) 查看 (Y) 呼叫 (C) 传送 (E) 帮助 (H)
D 🖆 🏐 🖏 🕒 🚰 🗍
  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
                                     描
```

(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。

```
🍢 arasys - 超级终端
                                                                           - 0 ×
文件(F) 编辑(E) 查看(V) 拨入(C) 传送(T) 帮助(H)
02 28 28
 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
 TxO /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:

```
🗞 armsys - 超级终端
                                                                           _ 🗆 ×
文件(F) 编辑(E) 查看(V) 拨入(C) 传送(T) 帮助(H)
 02 38 08
                                                                               .
 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 TP 192.168.253.2 -> 192.168.253.3 TCMP
 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 BIOS 所提供的功能部件的测试。

关于该文档,有任何问题,欢迎来信交流:<u>office@hzlitai.com.cn</u>、<u>support@hzlitai.com.cn</u>, 或垂询电话: 0571-88331446。