

■ 高等学校理工科电子信息类规划教材

嵌入式系统设计实验教程

EMBEDDED SYSTEM DESIGN EXPERIMENT

马洪连 主编



大连理工大学出版社
DALIAN UNIVERSITY OF TECHNOLOGY PRESS

■ 清华大学出版社“十二五”重点图书



嵌入式系统设计实验教程

EMBEDDED SYSTEMS DESIGN EXPERIMENTAL COURSE

第二版

◎

◎

◎

◎

◎

◎

◎

◎

清华大学出版社

■ 高等学校理工科电子信息类规划教材

嵌入式系统设计实验教程

EMBEDDED SYSTEM DESIGN EXPERIMENT

主编 马洪连

编者 丁 男 张益嘉

宋嘉琳 许 侃



大连理工大学出版社
DALIAN UNIVERSITY OF TECHNOLOGY PRESS

林蓬成献类息言千串样工野对学等高

图书在版编目(CIP)数据

嵌入式系统设计实验教程/马洪连主编. —大连:大连理工大学出版社, 2007. 6

ISBN 978-7-5611-3568-6

I. 嵌… II. 马… III. 微型计算机—系统设计—高等学校—教材 IV. TP360.21

中国版本图书馆 CIP 数据核字(2007)第 055552 号

大连理工大学出版社出版

地址:大连市软件园路 80 号 邮政编码:116023

电话:0411-84708842 邮购:0411-84703636 传真:0411-84701466

E-mail:dutp@dutp.cn URL:<http://www.dutp.cn>

大连业发印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸:185mm×260mm 印张:15.5 字数:349 千字
2007 年 6 月第 1 版 2007 年 6 月第 1 次印刷

责任编辑:梁 锋 范业婷 责任校对:宜 呈
封面设计:宋 蕾

ISBN 978-7-5611-3568-6

定 价:24.80 元

大连理工大学出版社
DALIAN UNIVERSITY OF TECHNOLOGY PRESS



前 言

随着信息化、智能化、网络化的快速发展,嵌入式技术已经成为通信及电子类产品的共同发展方向,并在相关领域内得到了广泛应用。嵌入式系统具有软硬件紧密结合的特点,这使其与计算机领域单纯的软件设计及电子工程领域单纯的硬件设计都不同,它是两者的结合,是软硬件的综合设计。目前国内能进行这方面设计的人才还很缺乏。本书正是采用软硬件结合的方式介绍嵌入式系统设计,以帮助有志于从事相关研究及应用的学生和工程技术人员提高综合设计能力。

目前,各高校所开设的相关课程主要由理论和上机实践两部分组成。其中,上机实践是学习嵌入式系统设计的重要环节。通过课堂讲解,能让学生掌握嵌入式系统的基本知识;通过动手实践,才能让学生掌握嵌入式系统设计开发的方法。

本书正是针对嵌入式课程中的实验环节,从实用的角度出发,分别为嵌入式接口、实时操作系统 uCOS-II 和嵌入式 Linux 操作系统等理论课程提供相应的配套实验,同时附有部分原理图和实验代码,使得本书不仅可以作为上机实验的实验指导书,同时也可以作为相关理论学习的参考书。

本书分为 4 篇,共 10 章,主要实验 20 个,各个实验相对独立,针对不同要求和不同水平提供不同的实验内容。具体内容安排如下:

第 1 篇,实验准备,包括第 1 章和第 2 章。主要介绍实验过程中所使用的实验平台和在做嵌入式接口实验以及普通嵌入式系统开发中所使用的主要工具。

第 2 篇,嵌入式系统接口实验,包括第 3 章。主要介绍基于 SAMSUNG S3C2410 ARM9 嵌入式处理器的接口实验。

第 3 篇,基于 uCOS-II 实时操作系统的开发,包括第 4 章

和第 5 章。主要介绍嵌入式实时操作系统 uCOS-II 的相关实验。

第 4 篇,嵌入式 Linux 操作系统的实验与应用,包括第 6~10 章。主要介绍嵌入式 Linux 操作系统相关的实验。针对学生的实际水平和课时,又将实验进一步细化。

附录部分给出了实验过程中常用的 API 函数、ARM 编程指令和 Linux 操作系统的命令,以便于学生在实验过程中查找与使用。

编者从事计算机教学工作多年,同时一直负责基于 ARM 微处理器系列的科研项目的开发和设计工作。在编写本教材的过程中,精选实验内容,力求符合从事嵌入式系统开发初学者的需求特点,做到概念清晰,理论联系实际。在叙述方法上,力求由浅入深、通俗易懂,以使读者能在较短的时间内掌握相关知识,达到事半功倍的效果。

本书可作为高等院校相关专业的高年级学生和研究生专业实验教材,也可作为从事嵌入式系统开发和设计人员的参考用书。

为了便于本课程的教学需要,本书另配有多媒体教学课件,需要者可通过 E-mail:mhl@dlut.edu.cn 联系,或者在编者的网站 <http://esis.dlut.edu.cn> 上下载。

由于编者水平有限,书中难免有出错和疏漏之处,敬请读者批评指正。大家有任何意见或建议,请通过以下方式与我们联系:

邮箱 jcyj@dlut.cn

电话 0411-84707962;84708947

编者

2007 年 5 月

目 录

第 1 篇 实验准备

- | | |
|-----------------------|--------------------------|
| 第 1 章 嵌入式系统实验平台概述 / 1 | 2.2 超级终端设置及映像文件下载方法 / 11 |
| 第 2 章 嵌入式系统开发环境简介 / 6 | |
| 2.1 ADS1.2 开发环境 / 6 | |

第 2 篇 嵌入式系统接口实验

- | | |
|-------------------------|----------------------|
| 第 3 章 嵌入式系统驱动程序实验 / 13 | 3.3 LCD 的驱动控制实验 / 23 |
| 3.1 ARM 的串行口实验 / 13 | 3.4 触摸屏驱动实验 / 35 |
| 3.2 ARM 的 A/D 接口实验 / 18 | |

第 3 篇 基于 uCOS-II 实时操作系统的开发

- | | |
|------------------------------------|---------------------------------|
| 第 4 章 实时操作系统 uCOS-II 开发与应用 / 43 | 4.2 完善的 uCOS-II 开发框架 / 52 |
| 4.1 uCOS-II 在 ARM 微处理器上的移植及编译 / 43 | 第 5 章 基于 uCOS-II 操作系统的开发案例 / 54 |

第 4 篇 嵌入式 Linux 操作系统的实验与应用

- | | |
|-----------------------------|------------------------------------|
| 第 6 章 嵌入式 Linux 开发基础知识 / 57 | 第 8 章 MiniGUI 图形应用程序设计 / 82 |
| 6.1 嵌入式 Linux 简介 / 57 | 8.1 MiniGUI 的配置与安装 / 82 |
| 6.2 嵌入式 Linux 开发流程 / 59 | 8.2 MiniGUI 程序架构及helloworld程序 / 87 |
| 6.3 建立嵌入式 Linux 开发环境 / 60 | 8.3 MiniGUI——loadbmp 位图实验 / 92 |
| 第 7 章 基础实验 / 65 | 第 9 章 内核实验 / 95 |
| 7.1 熟悉 Linux 开发环境 / 65 | 9.1 Linux 内核移植与编译实验 / 95 |
| 7.2 多线程应用程序设计 / 66 | 9.2 内核、根文件系统烧写实验 / 114 |
| 7.3 串行端口程序设计 / 74 | |
| 7.4 AD 接口实验 / 79 | |

第 10 章 驱动模块实验 / 121

- 10.1 内核驱动设计入门——模块方式驱动实验 / 121
- 10.2 内核驱动设计实验——触摸屏

附录 / 186

- 附录 1 嵌入式系统应用编程 API 函数 / 186

参考文献 / 242

驱动 / 142

- 10.3 系统中断实验——键盘中断的实现 / 154
- 10.4 SD 驱动使用实验 / 166

附录 2 ARM 汇编指令集 / 210

附录 3 常用 Linux 命令的使用 / 233

第 8 章 嵌入式 Linux 系统应用 / 57

- 8.1 MiniGUI 的配置与安装 / 57
- 8.2 MiniGUI 控件结构及 HelloWorld 程序 / 87
- 8.3 MiniGUI——IUI 图形库 / 92

第 9 章 嵌入式 Linux 系统应用 / 92

- 9.1 Linux 内核移植 / 92
- 9.2 内核移植 / 92

第 6 章 嵌入式 Linux 系统应用 / 57

- 6.1 嵌入式 Linux 简介 / 57
- 6.2 嵌入式 Linux 开发流程 / 60
- 6.3 建立嵌入式 Linux 开发环境 / 60

第 7 章 基础实验 / 62

- 7.1 熟悉 Linux 开发环境 / 62
- 7.2 系统级应用程序设计 / 66
- 7.3 串行端口程序设计 / 74
- 7.4 AD 接口实验 / 78

第 1 篇

实验准备

第 1 章 嵌入式系统实验平台概述

嵌入式系统的硬件平台种类比较多,但是从目前国内应用情况来讲,基于 ARM 体系结构嵌入式系统占领了 32 位嵌入式微处理器市场较大的份额,并且在国内市场上很容易购买到 ARM 核的嵌入式处理器。正因为如此,本书具体的实验选用了基于 ARM 体系结构硬件平台,即由北京博创兴业科技有限公司开发生产的 UP-NetARM3000 (ARM7)和 UP-NetARM2410-S(ARM9)嵌入式系统实验仪。

UP-NetARM3000 的微处理器采用的是 ARM7TDMI 内核的三星 S3C44B0X01 芯片,因为其处理器内部没有 MMU(存储器管理单元),所以只能运行一些如 uCos、uClinux 等实时操作系统。UP-NetARM2410-S 的处理器是基于 ARM920T 内核的三星 S3C2410 芯片,其内部具有 MMU,可以运行一些较大的操作系统,如 Linux、Windows CE 等操作系统。通过使用这两个实验平台,我们可以学习嵌入式系统中的针对有 MMU 和无 MMU 的不同开发过程。有关 UP-NetARM3000 实验仪的相关资料可以访问博创公司的网站 www.up_tech.com 来获得。本书主要针对 UP-NetARM2410-S(ARM9)嵌入式系统实验仪作一些详细的介绍。

1. UP-NetARM2410-S 嵌入式系统实验仪概述

UP-NetARM2410-S 实验平台如图 1-1 所示,硬件配置详见表 1-1。

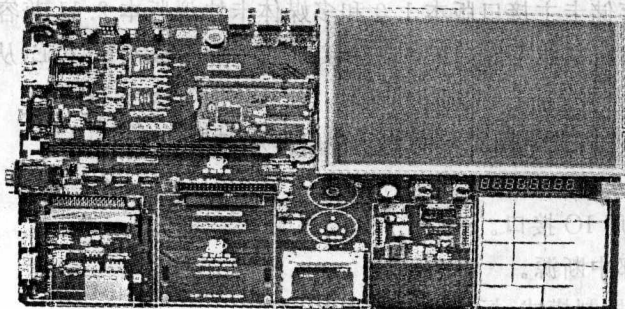


图 1-1 UP-NetARM2410-S 嵌入式系统实验仪正视图

表 1-1 UP-NetARM2410-S 的硬件配置

配置名称	型号	规格
CPU	ARM920T 结构芯片三星 S3C2410X	工作频率 203 MHz
以太网	AX88796	10/100 M 自适应
FLASH 盘	SAMSUNG K29F2808	16 MB(WinCE 64 M)
内存 SDRAM	HY57V561620BT-H	32 M×2=64 M
LCD	STN 240×320 256 色 TFT 640×480 24 位	
键盘 LED 数码管驱动器	ZLG7290	17 键数字小键盘
触摸屏	FM7843 驱动	
USB 主从接口	USB1.1	
串口	RS232	2 个
AD	3 个电位器控制输入	
调试接口	JTAG, 14 针、20 针	JTAG 烧写 Flash
音频输出	IIS 总线, UDA1341 芯片, 44.1 KHz 音频	
扩展插槽	三个 168Pin	总线直接扩展
DA 扩展板		D/A 转换
SD 卡扩展板		
GPS_GPRS 扩展板	SIMCOM 的 SIM100-E 模块	支持双道语音通信
PS/2&IC_CARD 扩展板		外接键盘鼠标和 IC 卡
CAN 扩展板		
IDE_CF 卡扩展板		

2. 嵌入式微处理器 S3C2410 功能介绍

基于 ARM9 的 S3C2410X 微处理器芯片是韩国三星公司生产的,其内部集成了大量的功能单元,具体包括:

- (1) 微处理器内核供电 1.8 V,存储器系统供电 3.3 V,外部 IO 供电 3.3 V。
- (2) 内含 16 kB 数据 Cache、16 kB 指令 Cache 和存储器管理单元 MMU。
- (3) 内置外部存储器控制器(SDRAM 控制和芯片选择逻辑)。
- (4) LCD 控制器(最高 4 K 色 STN 型和 256 K 彩色 TFT 型),一个 LCD 专用 DMA 单元。
- (5) 4 个带外部请求线的 DMA 单元。
- (6) 3 个通用异步串行端口(内含红外串口 IrDA1.0、16Byte FIFO 发送缓冲区、16Byte FIFO 接收缓冲区),2 通道 SPI(串行外设总线)。
- (7) 一个 IIC 串行总线,一个 IIS 嵌入式音频总线控制器。
- (8) 支持 SD 存储卡主接口版本 1.0 和多媒体卡协议版本 2.11 兼容。
- (9) 2 个通用串行总线 USB HOST(主接口),一个 USB DEVICE(从接口),支持 USB 版本 VER1.1。
- (10) 4 个 PWM 脉宽调制定时器和一个内部定时器。
- (11) 看门狗定时器。
- (12) 117 个通用 IO 接口。
- (13) 24 个外部中断源。
- (14) 4 种电源控制模式:标准、慢速、休眠、掉电。
- (15) 8 通道 10 位 ADC 转换器和触摸屏接口。

- (16)带日历功能的实时时钟。
- (17)芯片内置 PLL(锁相环)时钟电路。
- (18)设计用于手持设备和通用嵌入式系统。
- (19)16/32 位 RISC 体系结构,使用 ARM920T 处理器核的强大指令集。
- (20)微处理器内带有 MMU 的先进的体系结构,支持 Wince、EPOC32、Linux 等系统软件。
- (21)指令缓存(Cache)、数据缓存(Cache)、写缓冲和物理地址 TAG RAM,减小了对主存储器带宽和性能的影响。
- (22)ARM920T 处理器核支持 ARM 调试的体系结构。
- (23)内部具有先进的位控制器总线(AMBA)(AMBA2.0、AHB/APB)。
- (24)采用 FBGA 封装形式。

3. UP-NetARM2410-S 实验仪硬件资源

- (1)S3C2410 核心板:采用 S3C2410X 处理器、64MB NANDFLASH 和 64MB RAM。
- (2)双路 100 M 以太局部网卡:均由 AX88796 构成,但分配不同地址。
- (3)4 个 HOST(主)/1 个 DEVICE USB(从)接口:主 USB 口扩展为 4 个,由 AT43301 构成 USB HUB,其中电源管理用 MIC2525;从 USB 口 1 个。
- (4)3 个 UART/IrDA 接口:保持 2 个 RS232 串口,另外一个通用异步接收、发送串口可以作为增加的 1 个 RS485 串口或者 1 个 IrDA 红外收发器,它们均从处理器的 UART2 引出。
- (5)168Pin EXPORT:有一个 168Pin 扩展卡插槽,并去掉已经被主板上各模块占用的资源。网卡部分还在局部总线上,其余电路包括 168Pin 扩展槽都在外部总线上。
- (6)LCD 彩色显示器:兼容多种 LCD,可采用 5 英寸 256 色屏或 8 英寸 16bit 真彩屏,同时预留一个 24bit 接口。可以支持板外 8bit 或 24bit 屏。
- (7)TouchScr 触摸屏接口:采用 ADS7843,预备了直接用 2410 内部 ADC 构成的转换电路接口。
- (8)AUDIO:采用 UDA1341,具有放音、录音等功能。
- (9)PS2 KEYPAD:使用 ATMEGA8 单片机控制 2 个 PS2 接口和板载 17 键小键盘。2 个 PS2 可接 PC 键盘和鼠标。
- (10)LED:使用 ZLG7290 驱动 8 只小数码管。同时可作 IIC 总线实验。
- (11)POWER SUPPLY、RESET、RTC 等必需资源。
- (12)ADC:板载 3 个电位器和选择跳线,同时在板上设模拟电压输入专用接口。
- (13)IDE/CF 卡插座:支持 2.5 英尺的笔记本硬盘读写和 IDE 模式下的 CF 卡读写。
- (14)PCMCIA 和 SD 卡插座:由 EPM3128A100 CPLD 支持实现。
- (15)IC 卡插座:由 ATMEGA8 单片机控制。
- (16)DC/STEP 电机:步进电机采用 74HC573 扩展 IO,软件形成时序来控制。同时剩余 IO 可以控制 CAN 等电路,以节省 CPU 的 GPIO 资源。
- (17)CAN 总线接口:设置 1 个 CAN 口,采用 MCP2510 和 TJA1050。
- (18)Double DA:设置两个 DAC 端口,采用 MAX504 接 SPI 总线。

(19)GPRS/GPS 扩展板不做在主板上,单独设计扩展板。注意 GPS 的 RS232 需要增加 MAX3232 芯片来转换为 TTL 才能引到 168Pin 插座上。

4. 跳线设置参考

说明:PCB 上所有“EXPORT”丝印字符表示 168Pin 扩展槽。

- JP201 RESET-SEL:设置复位电路,位置在 JTAG20 插座右边。
 - 1-2:ICE 的 ICE-TRST 复位信号可以控制系统复位信号 RESET。
 - 2-3:ICE 的 ICE-TRST 复位信号不可以控制系统复位信号 RESET。
- JP501 LCDVDD-SEL:设置 LCD 的驱动芯片工作电压,与 LCD 电压有关,在逆变器上方。
 - 1-2:3.3 V。
 - 2-3:5.0 V。
- JP502 TS-SEL:选择 IO 口线 GPG12 的作用,用于选择触摸屏电路单元,在核心板右边。
 - 1-2:GPG12 作为 ADS7843 的片选信号,用于触摸屏第 1 单元。
 - 2-3:GPG12 作为第 2 单元的 XMON 信号。
- JP503 AIN5-SEL。
- JP504 AIN7-SEL:选择 AIN5 和 AIN7 的作用,用于选择触摸屏电路单元,在核心板右边。
 - 1-2:AIN5 和 AIN7 连接到扩展槽上作为 ADC 输入信号。
 - 2-3:连接到触摸屏第 2 单元使用。
- JP505、JP506、JP507、JP508 选择扩展的触摸屏电路单元,位置在逆变器下边。
 - 扩展触摸屏接口指在 24bit LCD 接口上的触摸屏信号,用于扩展外部触摸设备。
 - 4 个跳线分别对应:ETSXPON、ETSXMON、ETSYPON、ETSYMON。
 - 1-2:使用 ADS7843。
 - 2-3:使用第 2 单元(2410 内部 ADC)。
- JP901 MASTER-SEL:IDE 设备在线选模式下的主从选择,位置在 IDE 硬盘右边。
 - 1-2:HDD 硬盘为主盘,而 CF 卡为从 IDE 设备。
 - 2-3:HDD 硬盘为从盘,而 CF 卡为主 IDE 设备。
- JP1001、JP1002、JP1003 分别选择 AIN0、AIN1、AIN2 连接电位器或扩展槽,位置在各电位器下边。三个跳线分别对应电位器 POT0、POT1、POT2。
 - 1-2:ADC 输入端连接到扩展槽。
 - 2-3:ADC 输入端连接板载电位器。
- JP1104 TXD1-SEL。
- JP1105 RXD1-SEL:UART1 选择扩展槽或者 RS232 的 DB9 插座。
 - 1-2:UART1 连接 RS232-1,从 DB9 双串口插座输出。
 - 2-3:UART1 连接到扩展槽。
- JP1101 RS485VCC-SW:RS485 接口是否提供电源选择开关。

1-2:DB9 母座上不提供 5 V 电压。

2-3:DB9 母座上提供 5 V 直流电压,从第 6 脚上输出。

JP1102 TXD2-SEL。

JP1103 RXD2-SEL:UART2 选择跳线,分别为 RS485、IrDA、扩展槽。

1-2:UART2 连接到 RS485 总线上。

2-3:UART2 连接到扩展槽上。

3-4:UART2 连接到扩展槽上。

4-5:UART2 连接到 IrDA 红外线电路上。

5. 嵌入式系统开发流程

如图 1-2 所示为嵌入式系统开发的总体流程,在第 2 章的实验案例中给出了建立工程,设置编译环境和工程模板参数,编译,调试,下载等过程的详细说明。

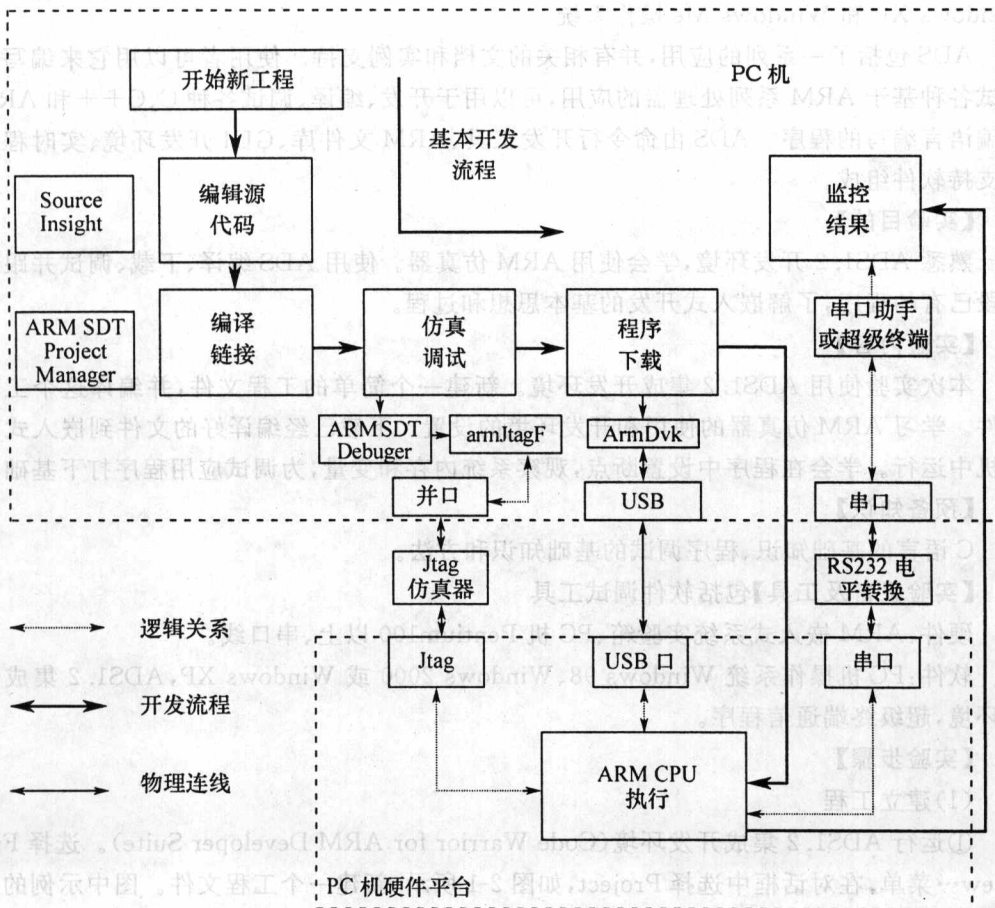


图 1-2 嵌入式系统开发流程框图

第 2 章 嵌入式系统开发环境简介

2.1 ADS1.2 开发环境

ARM ADS(ARM Developer Suite)是 ARM 公司提供的专门用于 ARM 相关的应用开发和调试的综合性软件工具。目前 ADS 的最新版本是 1.2,它是一款功能强大又易于使用的开发工具。它可以安装在 Windows NT4、Windows 2000 等操作系统下,还支持 Windows XP 和 Windows Me 操作系统。

ADS 包括了一系列的应用,并有相关的文档和实例支持。使用者可以用它来编写和调试各种基于 ARM 系列处理器的应用,可以用于开发、编译、调试各种 C、C++ 和 ARM 汇编语言编写的程序。ADS 由命令行开发工具、ARM 文件库、GUI 开发环境、实时程序和支持软件组成。

【实验目的】

熟悉 ADS1.2 开发环境,学会使用 ARM 仿真器。使用 ADS 编译、下载、调试并跟踪一段已有的程序,了解嵌入式开发的基本思想和过程。

【实验内容】

本次实验使用 ADS1.2 集成开发环境。新建一个简单的工程文件,并编译这个工程文件。学习 ARM 仿真器的使用和开发环境的设置。下载已经编译好的文件到嵌入式目标机中运行。学会在程序中设置断点,观察系统内存和变量,为调试应用程序打下基础。

【预备知识】

C 语言的基础知识、程序调试的基础知识和方法。

【实验设备及工具】

包括软件调试工具

硬件:ARM 嵌入式系统实验箱、PC 机 Pentium100 以上、串口线。

软件:PC 机操作系统 Windows 98、Windows 2000 或 Windows XP,ADS1.2 集成开发环境,超级终端通信程序。

【实验步骤】

(1) 建立工程

① 运行 ADS1.2 集成开发环境(Code Warrior for ARM Developer Suite)。选择 File | New... 菜单,在对话框中选择 Project,如图 2-1 所示,新建一个工程文件。图中示例的工程名为 Exp6.mcp。点 Set... 按钮可为该工程选择路径,如图 2-2 所示,选中 Creat Folder 选项后将以图 2-1 中的 Project Name 或图 2-2 中的文件名为名创建目录,这样可以将所有与该工程相关的文件放到该工程目录下,便于管理工程。

在图 2-1 中工程模板列表中的 2410 ARM Executable Image 是专为本嵌入式开发板

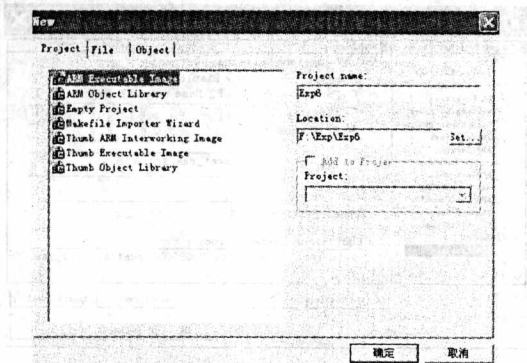


图 2-1 新建工程

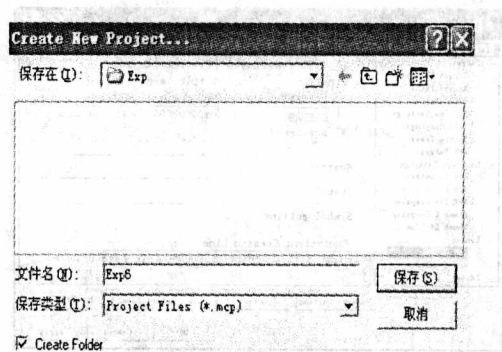


图 2-2 保存工程

设置的工程模板,后文有具体说明。在此也可选择 ARM Executable Image 通用模板。

②在新建的工程中,如图 2-3 所示,选择 Debug 版本,使用 Edit | Debug Settings 菜单对 Debug 版本进行参数设置。

③在 Debug Settings Panels 对话框中选择 Target Settings 项,如图 2-4 所示。在 Post-linker 一栏中选择 ARM fromELF。

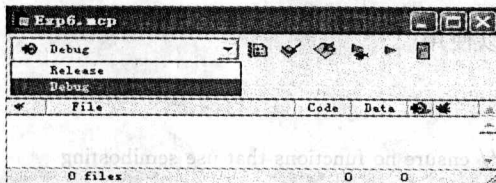


图 2-3 选择版本

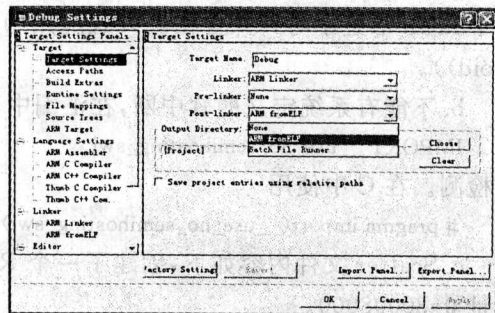


图 2-4 Target Settings

④在 Debug Settings Panels 对话框中选择 ARM Linker 项,如图 2-5 所示。在 Output选项卡的 Simple image 框中设置连接的 Read-Only(只读)和 Read-Write(读写)地址。地址 0x30008000 是开发板上 SDRAM 的真实地址,是由系统的硬件决定的;0x30200000 指的是系统可读写的内存地址。也就是说,在 0x30008000~0x30200000 之间是只读区域,存放程序的代码段,在 0x30200000 开始是程序的数据段。

如图 2-5 所示的设置只是一种简单设置,如果程序需要用到标准 C 库函数的话需要按图 2-6 进行连接地址的设置。

标准 C 中如果使用 malloc 及其相关的函数,需要使用系统的堆(Heap)空间,可以通过 Scatter 文件来描述系统 Heat 段的位置。针对 2410-S 开发板,把程序的入口定位在 0x30008000,并定义 Scatter 文件为 scat_ram.scf。在图 2-6 中选择 Linktype 为 Scattered,输入 Scatter 文件名 scat_ram.scf;然后切换到 Options 选项卡,在 Image Entry Point 框中输入 0x30008000。也可以在图 2-6 的 Equivalent Command Line 框中直接输入 -entry 0x30008000 -scatter scat_ram.scf 进行上述设置。

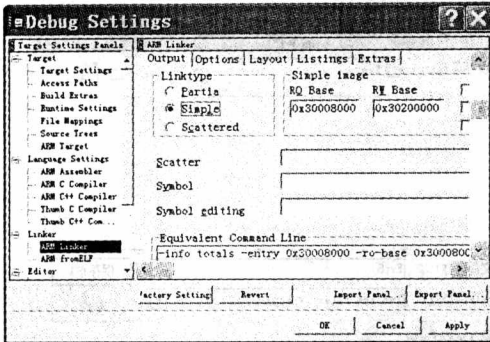


图 2-5 设置连接地址范围

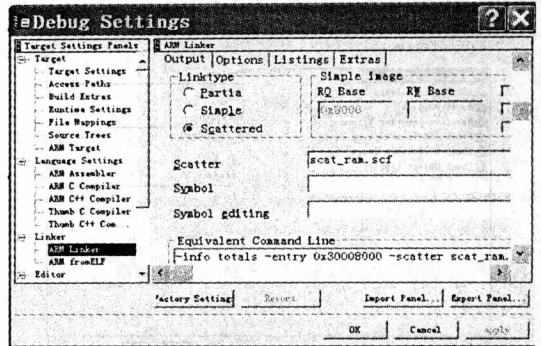


图 2-6 通过 Scatter 文件设置连接地址

提示:

a. 程序移植到 ADS 后,最开始首先执行用汇编写的初始化代码——包括中断向量和堆栈的初始化。在该段代码中使用如下两条语句,进入 C 语言代码。

```
IMPORT __main ;注意 main 前面是两个下划线
B __main
```

进行系统内部的标准 C 函数初始化,然后调用用户在 C 中定义的 main 函数(注意:两个 main 都是小写),并且在嵌入式应用中用户 C 的 main 函数中不能有参数(int main (void))。

b. 不能有系统定义的软中断,在汇编中可以使用

```
IMPORT __use_no_semihosting_swi
```

来检测。在 C 中使用

```
# pragma import(__use_no_semihosting_swi) // ensure no functions that use semihosting
```

c. Scatter 文件内容如下,创建了一个 RAM_LOAD 的程序和数据的装载区域,起始地址 0x30008000。

```
RAM_LOAD 0x30008000
{
  RAM_EXEC +0
  {
    startup.o (init, +First)
    * (+RO)
  }
  LOPAGETABLE 0x30200000 UNINIT ;about 2MByte offset SDRAM
  {
    pagetable.o (+ZI)
  }
  STACKS +0x100000 UNINIT ;64kByte under L0 pagetable
  {
    stack.o (+ZI)
  }
  RAM +0
  {
    * (+RW,+ZI)
  }
  HEAP +0 UNINIT
```



```

{
    heap.o (+ZI)
}
EXCEPTION_EXEC 0 OVERLAY ;exception region
{
    exception.o (+RO)
}
}

```

d. 定义 retarget.c 函数,重新定位标准 C 库中 stdio 的一些相关函数。主要有:

```

struct __FILE { int handle; /* Add whatever you need here */};
FILE __stdout; //文件的定义
int fputc(int ch, FILE * f) //fputc 函数
int ferror(FILE * f) //ferror 函数
void _sys_exit(int return_code) //系统退出函数
int __raise(int signal, int argument)
__value_in_regs struct __initial_stackheap__user_initial_stackheap( unsigned R0, unsigned SP,
unsigned R2, unsigned SL) //用户的堆空间和栈空间函数

```

具体定义可以参考 init/retarget.c。

⑤ 在第④步中如果不选择简单的连接地址设置,则需按图 2-7 所示设置 C 编译器。在 Debug Settings 对话框中选择 ARM C Compiler 项,在 ATPCS 选项卡中选择 ARM/Thumb interworkir,或者在命令行中添加-apcs /interwork。

⑥ 在第④步中选择简单的地址连接设置,在 Debug Settings 对话框中选择 ARM Linker 项,如图 2-8 所示。在 Layout 选项卡的 Place at beginning of image 框中设置程序的入口模块。指定在生成的代码中,程序是从 startup.o 开始运行的。Object 设为 startup.o,section 设为 init。

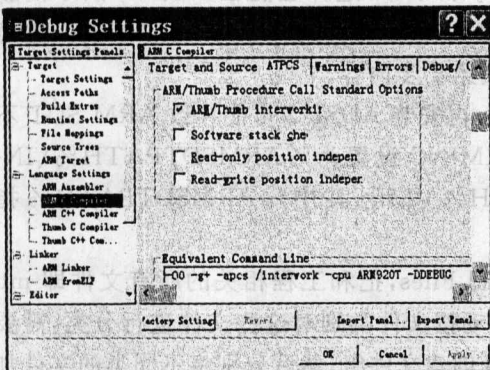


图 2-7 设置 ARM C Compiler

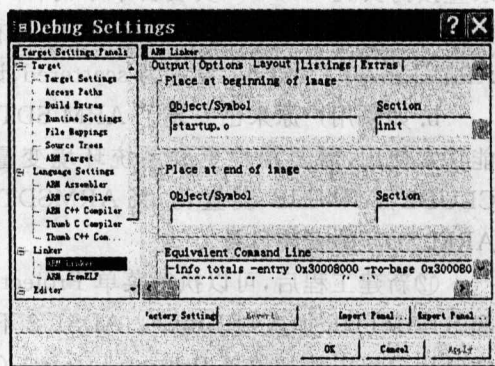


图 2-8 设置入口模块保存工程

⑦ 在 Debug Settings 对话框中选择 ARM fromELF 项,如图 2-9 所示。在 Output file name 框中设置输出文件名为 system.bin,这就是要下载到开发板的嵌入式应用程序文件。

⑧ 回到如图 2-3 所示的工程窗口中,选择 Release 版本,使用 Edit | Release Settings 菜单对 Release 版本进行参数设置。