

嵌入式软件设计之

思想与方法

张邦术 编著



北京航空航天大学出版社

嵌入式软件设计之 思想与方法

张邦术 编著

北京航空航天大学出版社

内 容 简 介

本书从教学的角度出发,全面讨论了嵌入式软件设计的思想与方法。在编排上循序渐进,从基础准备,到驱动模型,再深入到整个系统及系统的构建。在讲解上通过建立模型来帮助读者系统掌握嵌入式软件设计的普遍原理与编程接口。内容包括:高效、稳定和规范的程序基础,多任务环境,I/O 系统的内部结构,驱动模型,BSP 设计要素,嵌入式软件设计的经验技巧;在硬件基础方面讨论了总线与设备的模型,基于 MIPS 和 ARM SoC 在多个系统平台 VxWorks, Linux 及 WinCE 下的系统资源的操控。

本书可作为在校学生学习嵌入式软件设计原理的教学参考用书,也可作为嵌入式软件开发工程人员深入掌握系统软件设计的指南,以及嵌入式软件培训的参考教材。

图书在版编目(CIP)数据

嵌入式软件设计之思想与方法/张邦术编著. —北京:
北京航空航天大学出版社, 2009. 1
ISBN 978-7-81124-420-5

I. 嵌… II. 张… III. 软件设计 IV. TP311.5

中国版本图书馆 CIP 数据核字(2008)第 159679 号

© 2009, 北京航空航天大学出版社, 版权所有。

未经本书出版者书面许可,任何单位和个人不得以任何形式或手段复制本书内容。
侵权必究。

嵌入式软件设计之思想与方法

张邦术 编著

责任编辑 史海文 杨 波

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100191) 发行部电话:010-82317024 传真:010-82328026

<http://www.buaapress.com.cn> E-mail:emsbook@gmail.com

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787 mm×960 mm 1/16 印张:18.25 字数:409 千字

2009 年 1 月第 1 版 2009 年 1 月第 1 次印刷 印数:5 000 册

ISBN 978-7-81124-420-5 定价:32.00 元

前言

硬件技术的飞速发展,使硬件的性能显著提高,并且成本极速降低。微处理器已经深入到人们生活和生产的各个领域,各种产品和设备都逐渐增加了复杂的智能化功能,使得消费类电子产品、个人媒体产品、个人数字助理以及工业控制等领域得以快速发展。随着这些产品的高度智能化和复杂化,嵌入式软件的需求得到迅猛发展。从单片机的控制软件,到功能强大的多任务实时操作系统平台,产品的智能化程度越来越高,易用性越来越好,嵌入式软件及其应用领域越来越广泛,从而对嵌入式软件的要求也变得越来越复杂。

本书旨在为嵌入式软件开发爱好者提供一个入门的引导。面对复杂的嵌入式系统软件,作为一位初学者,如何清楚把握嵌入式软件的设计对象与目标,如何寻找一个很好的切入点,尽快参与到嵌入式软件的设计当中,对于这些问题,希望通过本书的讲解,能够为读者提供一些有益的启示。

笔者多年来一直在嵌入式软件领域从事实际项目的开发工作,出于对软件设计的执著与偏爱,笔者把这些年从事嵌入式软件设计的经验点滴整理出来,与更多的嵌入式软件设计爱好者分享。目前,尽管介绍嵌入式软件设计方面的书籍较多,但全面、系统地讨论如何从头开始设计嵌入式系统软件的书籍却很少。很多嵌入式软件设计方面的书籍都是一些诸如百科全书的参考手册,由于体系过于庞大,或讨论过于专业,初学者很难在短时间内把握其中有用的部分,因而更难将庞大体系里各书籍中的精华串到一起,而本书正是这些书籍精华的一种提炼。本书以讲述的方式,深入剖析嵌入式系统软件设计的各个层面,以及设计实践中的各个关键之事,以帮助读者轻松地领会嵌入式软件设计的方法,掌握嵌入式软件的核心架构。

书中通过对嵌入式系统的分解,重点讲述嵌入式系统软件的层次结构。通过对目前多个主流系统(VxWorks, Linux, WinCE)内核进行深入浅出的剖析与对比,帮助读者建立起正确的驱动设计模型;通过对不同硬件平台(MIPS, ARM)所开发的板级支持包(BSP)的深入讨论,帮助读者掌握硬件适配层(OAL)设计的核心概念,使读者清楚理解系统环境的上下文,前因后果,从而更好地把握各个软件模块设计的分界与接口,把握设计的对象与目标,在设计中做到心中有数,目标明确,从而更好、更快地解决问题。

要想成为一名成功的嵌入式软件程序员,程序的设计能力是首要的技能。如何打好程序设计基础,如何编写工程化的程序,如何在设计与团队协作开发、在后续开发中有效地升级

前言

与维护,如何编写规范的文档等,这些都是工程化软件设计中非常关键的环节,本书花费大量篇幅进行介绍,以帮助读者提高程序设计能力。

书中从各种复杂的软件系统中抽象出驱动模型和板级支持包的设计模型;对于硬件基础,也通过模型化的方法讲述了总线的一般概念与作用,抽象出输入/输出设备的模型。通过这些模型化的讲解,便于读者掌握嵌入式软件设计的目标与内容,从而提高软件设计能力。

1. 读者对象

本书的读者对象为嵌入式程序设计的初学者,本书也可作为大中专学生学习嵌入式软件设计的入门参考。对于那些已从事嵌入式软件设计一段时间,但是在设计实践中感觉力不从心,需要全面掌握嵌入式软件设计内容与目标,掌握一些新的技巧与方法的读者,相信本书将会起到良师益友的作用。

本书也可以作为嵌入式软件培训的教材。

2. 题材与组织

本书共分为四篇,其中第一篇着重讨论作为一名优秀的嵌入式软件设计人员所必备的知识 and 技能。需要说明的是:限于时间和精力,本书没能全面囊括嵌入式软件设计的所有知识点和技术面,但希望本书能让读者掌握基本的框架,使读者在今后的学习和工作实践中,更好地结合优秀读物和参考资料,不断学习和实践,从而提高自身的软件设计能力和水平。

(1) 基础方法篇

第一篇包含三部分内容:程序基础、多任务系统和硬件基础。要想成为一名优秀的嵌入式软件设计人员,概括地讲,需要熟练掌握以下4方面的知识技能:

- 程序基础;
- 操作系统原理;
- 硬件知识;
- 调试能力与学习能力。

在此篇中,简要分析了嵌入式软件设计的特殊性及其要求,讨论接口、代码及文档的要求规范,通过各种例证,以点带面,引导读者一步步设计出高性能、稳定可靠、维护性强和可读性好的嵌入式程序。

在“多任务操作系统”一章中,着重讨论了与嵌入式软件设计相关的多任务环境、模块间同步与通信协同、驱动设计以及动态库设计中可重入性等问题。为了避免重复,与驱动设计密切相关的 I/O 系统则放在驱动模型一章中详细讨论。与其他书籍不同,本书没有展开那些与嵌入式软件设计关系不大的内核机制,例如进度调度、内存管理的实现等,除非在涉及操作系统核心开发时才有所展开。在讨论多任务话题时,本书也是从程序实现的角度来分析系统需求以及软件的实现,而不单从理论上加以分析。去除了那些既复杂,又与 OEM 硬件平台开发不相关的内容,增加了实践中的理论相关性,使本书非常简明、实用。

在“硬件基础”一章中简要地讨论了 MIPS 和 ARM 的汇编程序设计基础,以及与板级支持包(BSP)开发紧密相关的 CPU 中断与异常的处理。通过对 MIPS 与 ARM 两种 RISC 架构内部机制的比较,帮助读者理解硬件的工作机制,以及软件与硬件交互所要实现的任务。

除此之外,还简要地分析作为硬件设备基础的总线模型,并通过两种常用的总线 I²C 和 PCI 的相关例子,帮助读者学习和理解总线协议。总线是设备的命脉,设备通过总线传达信息,交互数据。理解了这些原理和硬件的工作行为后,就能做到在驱动实现和板级支持包的开发中得心应手,游刃有余。

对于嵌入式软件开发的工具包,以及一些软件的调试技能和技巧,由于篇幅所限,在此未作讨论,将在后续的版本中不断丰富和完善。

(2) 驱动模型篇

在第二篇中深入讨论了嵌入式软件设计中重要任务之一的驱动程序的模型架构。从抽象、一般的概念到多个实际嵌入式操作系统的驱动架构的分析,读者可以深入学习驱动设计的层次模型,清晰地掌握驱动设计过程中各种接口及其相互关系,从而准确地把握各个模块的设计任务。

第二篇从第 4 章到第 7 章,共分 4 章,第 4 章讨论驱动的通用模型,从驱动层次结构,驱动与应用及与系统的交互,驱动种类,驱动的系统初始化、挂接以及内部实现例程的各种接口来讨论驱动设计的一般概念。

后面三章分别就前一章所讨论的一般概念,结合 VxWorks, Linux 和 WinCE 等实际操作系统,深入分析 I/O 系统的内部运行机制、内部数据结构、应用程序打开、关闭设备文件以及进行各种读、写、控制操作所关联的驱动实现,并通过一个简单的例子,介绍如何编写一个完整、复杂的驱动程序。

(3) BSP/OAL 篇

第三篇是板级支持包(BSP)的开发。板级支持包的开发比驱动的开发更复杂,涉及的问题更多,要求的知识更专业。从编排上将这一更复杂的系统驱动放在第三部分,体现了由易到难的编排方式。本篇从中断、异常、硬件 I/O 访问等核心要素,讲解如何设计硬件适配层的支持软件包,详细讨论了中断处理的完整架构,异常处理的向量表安装和分发机理,并通过大量例子说明如何进行实际代码实现。

(4) 扩展篇

作为本书的结尾篇“扩展篇”,进一步介绍了作者在嵌入式软件设计中的一些心得体会。另外,再次讨论作为软件表现形式——“程序”的内部结构。深入理解程序的内部结构是开发板级支持包以及开发系统程序的核心基础。

本书虽然在最后一章才提到思想,事实上,前面的章节都通过嵌入式软件开发的实践阐述了嵌入式软件设计的方法与设计理念,从而将嵌入式软件的设计思想贯穿于全书之中。

本书通过实际例子,从驱动模型和操作系统底层设计模型来讲述嵌入式软件设计的一般

前言

原理与方法。在体系的安排上也采用循序渐进、由简到难的方式。虽然本书不能作为某个平台开发的完整参考,但是对每个问题的讨论,都力求深入、完整。

3. 如何理解本书内容

对于驱动的开发,本书不是针对一些特定设备讨论具体的驱动编程开发,而是讲述驱动开发必备的知识与技能,其逻辑关系如图 0-1 所示。

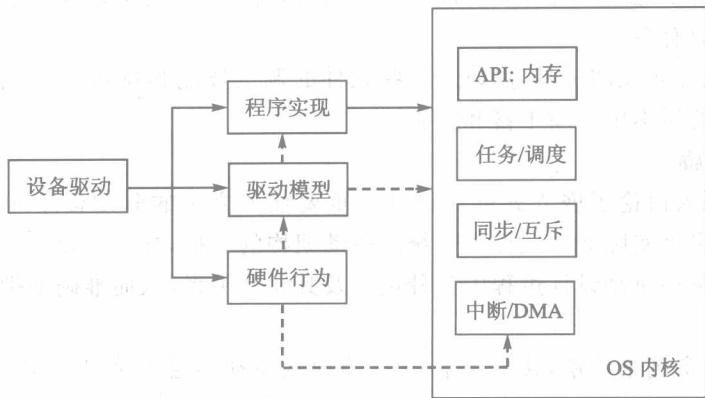


图 0-1 设备驱动开发的知识与技能

同样,对于 BSP,本书也讲述了所有必需的知识与技能,其逻辑关系如图 0-2 所示。

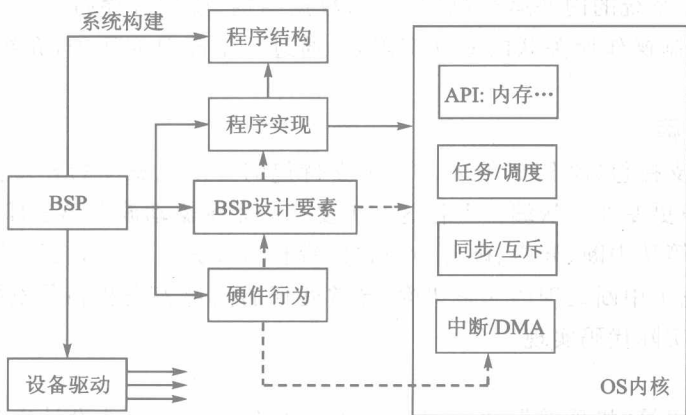


图 0-2 BSP 开发的知识与技能

这些内容的选材与组织编排体现了本书“思想与方法”的宗旨。本书详细地讨论所有这些相关知识与技能的各个方面,着重从能力方面培养读者设计嵌入式系统软件的各个部件乃至整个嵌入式产品。

为何要谈思想与方法？生活中这样的例证很多，若做一件事情不得法或不得要领，往往收效甚微。嵌入式软件是一项复杂的设计工作，如果方法不当，往往顾此失彼，或只见树木，不见森林。本书之所以谓之思想与方法，是因为从一个程序员的角度来看问题，力求讲清楚问题的实质，从整体到局部，从个性到共性，从理论到实践，都有比较深入的讨论。

历史上曾经有一位射影几何学大师帕斯卡，在研究圈圈棒棒的过程中，于16岁就发表了射影几何中最重要的定理之一——帕斯卡定理。以下摘引一个“圈圈棒棒非欧几何”的典故(<http://www.6edu.org.cn/news/ReadNews.asp?NewsID=6518>)，以提起读者兴趣。

帕斯卡的神秘六边形

差不多与笛沙格同时，对射影几何作出重要贡献的是数学天才帕斯卡。1623年，他出生在法国克勒芒，小时候虽体弱多病，却很早就显出非凡的数学才能。父亲不让小帕斯卡过早接触数学，怕过度紧张的思考损害他的健康，将所有的数学书籍都藏起来。

严格的禁令反而激发了小帕斯卡的好奇心，12岁时他问父亲：“几何学究竟是什么？”

父亲回答说：“几何学是一门提供正确作图，并找出各图形之间存在的关系的学科。”说完，马上强调以后再不能谈论数学问题了。

然而帕斯卡听了父亲的谈话后，激动的心情不能自己。他自立定义，把欧氏几何中的线段叫“棒棒”，圆叫“圈圈”，整日迷恋着棒棒和圈圈组成的图形。当父亲知道他自行证明，独立地发现了三角形的内角和定理时，不禁惊喜交加，叹服他的几何才能，从此不再阻止他学习数学了，还送给他一部《几何原本》。

从14岁起，帕斯卡经常随父亲参加巴黎一群数学家的每周聚会（法国科学院就是从这发展起来的），耳濡目染，使帕斯卡在科学之路上迅速成长。1639年，当笛沙格构造的射影空间遭非议、受排斥时，只有帕斯卡为其新思想所吸引。他用笛沙格的射影观点研究圆锥曲线，得到许多令人欣喜的新发现。

1640年，16岁的帕斯卡发表了《试论圆锥曲线》的8页论文，文中包含了3条定义，3个推理和一些定理。其中一个定理被认为是射影几何中最重要的定理：“圆锥曲线的内接六边形，延长相对的边得到3个交点，这3点必共线”。该定理命名为帕斯卡定理，定理中的六边形叫做“神秘六边形”。据说帕斯卡从这个定理导出了400多条推论。帕斯卡定理向人们展示了射影几何深刻、优美的直观魅力，其宏伟壮观的气势令人惊叹！

作为笛卡儿的学生，在解析法风靡一时，同时代人都不愿意接受射影观点的潮流下，帕斯卡独树一帜，用纯几何的方法发现了神秘六边形，取得了自古希腊阿波罗尼斯以来研究圆锥曲线的最佳成果，为射影几何大厦奠定了基石。帕斯卡的精神难能可贵。据说笛卡儿读了他的著作后大为叹服，竟不相信是出自自己的学生，一位16岁少年之手。

从上面这个例子中可以看出，一旦对于系统设计的原理有了深刻的掌握，对于某一个模块的设计有了系统、完整的实现，那么对于其他系统或模块的设计就会触类旁通，以至于凭直觉就可以推想在一个未知领域应该如何设计，而不在于它的形式上是“棒棒”还是“直线”，是 Vx -

前言

Works 还是 WinCE,其实质都是一样的。

本书重在讲解嵌入式操作系统软件设计的一些原理和方法。讨论思想虽然是一个哲学的范畴,而不是设计的问题。但一个软件产品的设计好坏,归根结底是在于一个人的品性,一个人的品性决定一个人能够把任何事情做到什么样的水平。所以在设计实践中,除了要学习专业知识,培养专业技能,更多地,还要培养自己认真负责的态度,锻炼程序设计的意志! 中国有句“宝剑锋从磨砺出,梅花香自苦寒来”的名言。程序的设计、嵌入式软件的设计、操作系统原理的理解以及对总线协议的理解,都需要设计人员艰苦的劳动。

4. 感谢

首先要感谢我读研究生时的导师谢文楷教授,笔者的成长离不开导师的指导和辛勤的培育。在我的研究生学习和研究生涯中,导师不但教给我学习和研究的方法,更教育了我应该如何做人,如何在人生道路上去接受挑战,如何去面对挫折。导师宽广的胸怀,诲人不倦的风格,严谨的学术作风,深刻地影响着我以后的学习和工作,也体现了本书的精神与实质。

感谢我的母校电子科技大学,云阳师范学校! 歧阳小学! 谨以此书献给培育我成长的所有母校。感谢我的中学数学老师邓世文、李必信,他们在数学思维方面对我的影响极其深刻和重要。感谢我的启蒙老师刘丙章老师! 感谢泰克(Tektronix)科技的 Michael Flaherty 和 Jeff Yost 对我工作的帮助与支持。

感谢北京航空航天大学出版社马广云博士、胡晓柏主任对本书的支持与关怀,他们的耐心鼓励与支持,使得本书才能够成功问世。非常感谢本书的编辑史海文先生,史先生认真地阅读了书稿,从书中的错字、别字、语句表达,到专业知识的理论与术语都提出了细致的修改和指导意见。

感谢我的家人、父母、兄弟姐妹张定寿、张邦松、张淑琼、刘文宣、陈瑶、张毅祥、刘仁芳和张佳婉,是父母和家人培育了我的智慧,激发了我工作中的灵感,锻炼了我坚韧顽强的品性! 感谢我的朋友刘滢、方荣、戴慰如、贾如和卓志伟对本书直接和间接的帮助与启发!

感谢我的同事,联想集团的万长青、刘洪、郭建军、李建辉、黄平、柴治;泰鼎多媒体的薛强、张立人、唐国峰、Simon Hong、汪永宁;杰得微电子总裁欧阳合博士、张泉、杨海;微开半导体的技术总裁艾保罗(Paul van der AREND)先生、李文华总经理、徐世彰(Ray Hsu)、管华亮等。同事永远是学习和工作中的良师益友。

限于笔者水平有限,加之时间仓促,错误不当之处,诚望读者批评指正,联系方式是 E-mail: MOL@263.NET。

目 录

第一篇 基础方法篇

第1章 程序基础

1.1 设计高性能程序的必要性	3
1.1.1 设计高性能程序的必要性	3
1.1.2 嵌入式软件的设计范畴	3
1.1.3 嵌入式软件的分层结构	6
1.2 嵌入式软件的程序设计要求	8
1.2.1 代码结果的要求	9
1.2.2 代码形式的要求	10
1.3 嵌入式软件开发的基本思路 and 原则	10
1.3.1 系统分析, 定义接口	11
1.3.2 函数实现, 优化算法	12
1.3.3 清理代码, 补充注释	14
1.3.4 测试修订, 完善文档	14
1.4 程序实例剖析	14
1.4.1 正确理解栈	14
1.4.2 内存泄漏	18
1.4.3 消除编译依赖	18
1.4.4 消除潜在隐患	20
1.4.5 规范实现范例	21
1.4.6 性能优化	23
1.5 程序设计其他注意点	30

目 录

1.5.1	谨慎使用“宏”	30
1.5.2	正确理解预定义宏	34
1.5.3	避免歧义	37
第2章 多任务操作系统		
2.1	板级支持包	40
2.2	嵌入式操作系统与实时性	40
2.2.1	嵌入式操作系统	41
2.2.2	实时操作系统	42
2.3	多任务概述	42
2.3.1	进程、线程与任务	43
2.3.2	何时需要多任务	44
2.3.3	任务状态的转换	50
2.3.4	进程调度与调试算法	51
2.3.5	任务相关的 API	51
2.4	进程间共享代码与可重入性	53
2.4.1	共享代码	53
2.4.2	共享代码可重入性问题	53
2.4.3	使用私有数据	55
2.4.4	使用临界区数据	57
2.5	线程间通信	57
2.5.1	共享数据结构	57
2.5.2	互 斥	59
2.5.3	信号量	60
2.5.4	临界区与信号量的实现实例	63
第3章 硬件基础		
3.1	ARM	74
3.1.1	ARM 编程模式	75
3.1.2	ARM 指令概述	78
3.1.3	ARM 异常及处理	80
3.2	MIPS	86
3.2.1	MIPS 编程模式	87
3.2.2	MIPS 指令概述	90

3.2.3 MIPS 中断与异常	95
3.3 接口基础	98
3.3.1 总线概述	99
3.3.2 I ² C 总线	105
3.3.3 PCI 总线	108
3.3.4 设备模型	115
3.3.5 一个 IDE 控制器设备实例	117

第二篇 驱动模型篇

第 4 章 驱动的通用模型

4.1 设备驱动的作用	121
4.2 驱动类型	123
4.2.1 Linux 中的驱动类型	123
4.2.2 WinCE 中的驱动类型	125
4.2.3 VxWorks 中的驱动类型	125
4.3 设备驱动的通用模型	126
4.3.1 模块部分的驱动	126
4.3.2 设备的驱动例程	127

第 5 章 VxWorks 的驱动模型

5.1 VxWorks 的 I/O 系统	131
5.1.1 I/O 系统概述	131
5.1.2 文件名与设备	133
5.1.3 基本 I/O	134
5.1.4 缓冲 I/O	136
5.1.5 格式化 I/O	136
5.2 VxWorks 的驱动及其内部结构	137
5.2.1 驱动的安装、驱动表	138
5.2.2 设备的创建、设备链表	140
5.2.3 文件的打开、文件描述符表	142
5.2.4 文件的读、写、控制和关闭操作	143

第 6 章 Linux 的驱动模型

6.1 Linux 的驱动加载方式	145
-------------------------	-----

目 录

6.1.1	内核驱动模块与模块化驱动	145
6.1.2	模块化驱动的加载与卸载	146
6.2	Linux 的驱动架构	147
6.2.1	一个最简单的内核驱动	148
6.2.2	一个最简单的模块驱动	151
6.2.3	Linux 驱动中注册驱动	153
6.2.4	Linux 系统中的设备文件	154
6.3	Linux 字符型设备驱动	155
6.3.1	驱动的加载与清理	155
6.3.2	中断的申请与释放	156
第 7 章 WinCE 的驱动模型		
7.1	WinCE 驱动类型	158
7.2	设备管理器及其驱动模型	159
第三篇 BSP/OAL 篇		
第 8 章 BSP 的基本概念		
8.1	BSP 与驱动	161
8.2	BSP 开发的目标任务	162
第 9 章 BSP 的设计要素		
9.1	中断处理	163
9.1.1	物理中断号与逻辑中断号	163
9.1.2	CPU 中断与中断控制器扩展	164
9.1.3	中断源的查找	165
9.1.4	中断处理线程	166
9.2	CPU 异常	166
9.2.1	异常向量表	167
9.2.2	向量表的安装	173
9.2.3	异常处理代码实例	177
9.3	硬件 I/O 的访问	188
9.3.1	避免使用绝对物理地址	188
9.3.2	内存一致性问题	192

9.3.3 I/O 访问的刷新	198
-----------------------	-----

第 10 章 Linux 的启动过程

10.1 Linux 的启动流程	199
10.2 Linux 的启动过程简介	201
10.2.1 _stext 函数	201
10.2.2 start_kernel 函数	203
10.2.3 setup_arch 函数	204
10.2.4 trap_init 函数	204
10.2.5 init_IRQ 函数	205
10.2.6 sched_init 函数	205
10.2.7 do_initcalls 函数	205
10.2.8 init 函数	206
10.2.9 init 程序	207

第 11 章 WinCE 的设计

11.1 WinCE OS 平台开发简介	209
11.1.1 WinCE 平台的开发流程	209
11.1.2 WinCE 内核结构	211
11.1.3 WinCE 设计中的一些名词术语	212
11.2 WinCE BSP 开发	213
11.2.1 启动装载器	213
11.2.2 OAL 开发	215
11.2.3 WinCE 配置文件	219
11.3 WinCE 设备驱动的开发流程	221
11.3.1 设备驱动源代码	221
11.3.2 修改配置文件	222
11.3.3 向 OS 平台注入驱动	223

第四篇 扩展篇

第 12 章 理解程序的内部结构

12.1 x86 汇编及其程序结构	226
12.1.1 x86 程序段定义	227

目 录

12.1.2	关联段寄存器、确定段的种类	230
12.1.3	段组伪指令	230
12.2	嵌入式系统中的程序结构	231
12.2.1	嵌入式系统中执行程序的映像	231
12.2.2	链接器与命令脚本	236
12.3	ELF 文件格式	241
12.3.1	ELF 文件格式概述	241
12.3.2	ELF 文件格式分析器	248
第 13 章 嵌入式系统的设计思想		
13.1	直截了当的思想	262
13.2	层次化的思想	267
13.3	循序渐进的思想	269
13.4	实践是最好的老师	269
13.5	团队协作意识	270
13.6	大胆尝试与积极创新	270
结 束 语		272
参考文献		273

插图索引

图 1-1 嵌入式软件的分层结构	7
图 2-1 VxWorks 中的任务状态转换图	50
图 2-2 驱动中的可重入性问题 1	54
图 2-3 驱动中的可重入性问题 2	56
图 2-4 使用共享数据区访问临界区的例子	58
图 3-1 ARM 程序状态寄存器格式	77
图 3-2 MIPS CPU 寄存器	88
图 3-3 MIPS FPU 寄存器	90
图 3-4 I ² C 数据位的传输	106
图 3-5 I ² C 起始条件和停止条件	106
图 3-6 I ² C 总线数据传输时序图	107
图 3-7 PCI CONFIG - ADDRESS 寄存器格式	113
图 3-8 PCI 类型 0 配置空间头部	114
图 3-9 ITE8172 IDE 控制器框图	118
图 5-1 驱动在系统中的层次结构	132
图 5-2 VxWorks I/O 系统的调用关系	133
图 5-3 VxWorks 驱动安装	140
图 5-4 VxWorks 设备添加	141
图 5-5 VxWorks 文件打开	142
图 5-6 文件读操作的 I/O 控制流程	143
图 6-1 Linux 驱动与操作系统核心之间的关系	147
图 7-1 WinCE 驱动内部框图	158
图 7-2 WinCE 系统中应用程序与设备驱动的交互	160
图 9-1 驱动程序中完整的中断处理架构	164
图 9-2 IT8172G 中断控制器内部框图	177

图 10-1	Linux 启动流程框图	200
图 10-2	Linux 启动执行过程细节	201
图 11-1	WinCE OS 开发的工作流程	210
图 11-2	WinCE 的内部层次结构	211
图 11-3	WinCE BSP 框图	214
图 12-1	x86 汇编段结构	228
图 12-2	宏汇编中的段链接映像	230
图 12-3	x86 段组定义	231
图 12-4	节的简单格式	237
图 12-5	节的完整定义	239
图 12-6	口(ENTRY)的定义	240
图 12-7	ELF 目标文件格式	242