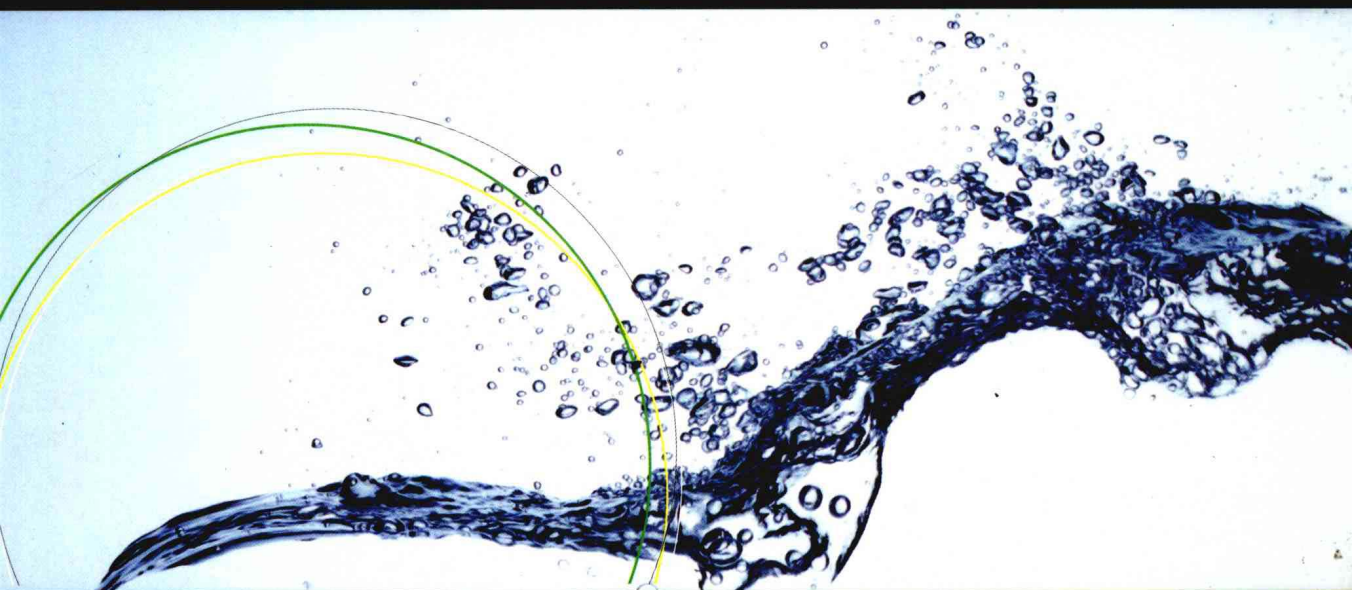


DESIGN



基于底层硬件的 软件设计

怯肇乾 编著
(Kai Zhaoqian)



北京航空航天大学出版社

基于底层硬件的软件设计

怯肇乾 编著
(Kai Zhaoqian)

北京航空航天大学出版社

内 容 简 介

介绍基于底层硬件的软件设计,涉及了设备驱动程序的设计、嵌入式实时操作系统的定制/移植、基于底层硬件的软件体系架构等实用技术。主要包括两个方面的内容:一是通用计算机在 Windows、Linux 和 VxWorks 等常见操作系统下的串/并/网络通信实现和 USB、ISA、PCI 设备/板卡的驱动程序设计;二是嵌入式应用体系的直接基本软件架构与基于 μ C/OS、DSP/BIOS、WinCE/EXP、 μ CLinux 及 VxWorks 等常见嵌入式实时操作系统下的基本软件架构及各类常见嵌入式软件体系下的 UART、SPI、CAN、EMAC、ADC、DAC、存储器件等外设/接口的驱动软件设计。书中还介绍了如何使用 CPLD/FPGA/PAC 等器件进行可编程数字/模拟逻辑软件的设计,进而实现所需的特定外设/接口及其连接与 FPGA-SoPC 软硬件协同的设计。

本书特别适合于从事嵌入式应用系统设计的广大工程技术人员,也是高等学校/职业学校嵌入式系统软硬件设计与机电一体化教育培训的理想教材和参考书。

图书在版编目(CIP)数据

基于底层硬件的软件设计/怯肇乾编著. —北京:北京
航空航天大学出版社,2008.7

ISBN 978-7-81124-350-5

I. 基… II. 层… III. 软件设计 IV. TP311.5

中国版本图书馆 CIP 数据核字(2008)第 098503 号

基于底层硬件的软件设计

怯肇乾 编著

(Kai Zhaoqian)

责任编辑 李青 李冠咏 李徐心

*

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: bhp@263.net

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:41 字数:1050 千字

2008 年 8 月第 1 版 2008 年 8 月第 1 次印刷 印数:5 000 册

ISBN 978-7-81124-350-5 定价:68.00 元

前 言

怎样在一个硬件平台上建立并运行一个最小的基本软件体系? 怎样通过软件与系统的各种外围设备打交道? 怎样通过系统的各种接口在软件上实现与外部通信? 怎样使构建的整个基本软件体系在操作系统的各种硬件外设或接口既稳定可靠又实时高效? ……本书系统地介绍了这些令人关注的具体设计应用中的敏感焦点技术。

基于底层硬件的软件设计主要是嵌入式软件体系的架构和能够对嵌入式应用系统进行监控的通用计算机设备驱动程序设计。本书介绍了两大类型的技术: 一是通用计算机在 Windows、Linux、VxWorks 等常见操作系统下的串/并/网络通信实现和 USB、ISA、PCI 设备/板卡的驱动程序设计; 二是嵌入式应用体系的直接基本软件架构与基于 $\mu\text{C}/\text{OS}$ 、DSP/BIOS、WinCE/EXP、 μCLinux 、VxWorks 等常见嵌入式实时操作系统下的基本软件架构及各类常见嵌入式软件体系下的 UART、SPI、CAN、EMAC、ADC、DAC、存储器件等外设/接口的驱动软件设计, 这两方面通过数据传输通信紧密地联系在一起。本书还介绍了如何使用 CPLD/FPGA/PAC 等器件进行可编程数字/模拟逻辑软件设计, 进而实现所需的特定外设/接口及其连接与 FPGA-SoPC 软硬件协同设计。书中既有对设备驱动程序模型、嵌入式实时操作系统的定制/移植、嵌入式体系外设/接口的驱动软件架构、可编程器件软件设计等的理论介绍和实践经验汇总, 又列举了大量的项目开发和实际问题解决处理的软件设计实例, 是作者多年来从事软硬件项目产品开发和高校应用技术课程讲解的经验总结与资料积累。

本书共有 12 章。第 1 章概括描述了基于底层硬件的软件设计所涉及的技术范畴; 第 2~4 章介绍通用计算机在常用操作系统下的设备驱动软件模型和串/并/网络的通信实现及其 USB、ISA、PCI 设备/板卡的驱动程序设计; 第 5 章介绍常见微控制/处理器的基本软件体系及其外设/接口驱动软件的直接软件架构; 第 6~10 章介绍常用嵌入式实时操作系统的定制/移植及其基本软件体系与外设/接口驱动的软件架构; 第 11 章介绍嵌入式应用体系外设/接口的可编程逻辑设计和 FPGA-SoPC 软硬件协同设计; 第 12 章归纳总结了基于底层硬件的软件的基本特点和设计规则并通过若干个典型的项目设计实例说明了这些设计规律的综合实践应用。

需要特别说明的是, 相关 VxWorks 操作系统的底层硬件的软件设计, 根据实际应用, 在书中把它分为两章进行介绍: 一章为通用 X86 及其 Pentium 系列计算机下的数据传输通信实现与设备驱动程序设计, 另一章为像 ARM 等的嵌入式应用软件体系架构。在工业数据采集和控制应用中, VxWorks 在 X86 及其 Pentium 系列计算机中应用普遍, 几乎直接安装 Tornado for X86/Pentium 就可运行, 所做的只是特殊的数据传输通信实现和设备驱动程序设计; 而在 ARM 等嵌入式应用体系中, 则需要做特别的操作系统移植和 BSP 设计及其外设/接口驱动软件设计。

本书具有以下 4 个特点:



1. 理论浅显易懂,偏重实用,项目设计实例丰富;
2. 知识涉及面广,现代新技术应用处处可见;
3. 软硬件嵌接紧密,重点讲述了软件如何稳定可靠、高效快速地操作硬件;
4. 结构组成上力求条理清晰、重点突出、目的明确,循序渐进、由浅入深、由抽象到具体、由理论到实践的思想贯穿于每一章节。

该书是本人“嵌入式软硬件及其系统设计”系列应用技术丛书中的第二本。本人计划写作4本书:《嵌入式系统硬件体系设计》、《基于底层硬件的软件设计》、《嵌入式应用程序及其监控软件设计》和《嵌入式系统工程规划设计》。第一本书《嵌入式系统硬件体系设计》已经出版,它是本书的基础,相关硬件的设计和操作可以参考该书。《嵌入式应用程序及其监控软件设计》是本书的后续技术书籍,本书是它的基础。

本书的读者群可以是从事工业检测控制、语音/图像处理与解压缩、航空航天、军事、移动通信及便携式PDA产品设计等行业的各级软硬件设计人员,也可以作为高等学校/职业学校嵌入式系统软、硬件设计与机电一体化教育培训的教材。本书的很多内容曾经被编成系列讲义,在高等学校本科生和专职/在职研究生中讲解,收到很好的效果。愿本书的出版能够给从事或欲从事软、硬件设计的广大工程技术人员开发设计出稳定可靠、简捷便利、经济实用的嵌入式系统产品带来更多的帮助。

基于底层硬件软件设计的目标和方向有两个:一是嵌入式基本体系及其外设与接口的软件架构;二是通用计算机常规操作系统下的设备驱动程序设计。两者都屏蔽了所有相关硬件的操作,只留有相关硬件操作的API函数、功能性软件设计。或者说,留给应用软件工程师的任务就是在基于硬件的程序架构下编写功能代码。

基于底层硬件的软件设计是一门综合性软硬件协同设计技术,涉及很多应用领域及方法和技巧。由于个人知识水平和认识能力的局限,书中存在的不当或错误之处,敬请广大读者批评指正。

法肇乾(Kai Zhaoqian)

2007年5月28日于上海

目 录

第 1 章 基于底层硬件的软件设计概述

1.1 底层硬件操作软件及设计的总体阐述	1
1.1.1 底层硬件操作软件的综合阐述	1
1.1.2 底层硬件操作软件的层次组织	2
1.1.3 基于底层硬件的操作软件设计	3
1.1.4 硬件操作软件设计的目的和要求	4
1.2 通用计算机底层硬件操作软件及设计	4
1.2.1 通用计算机的底层硬件软件概述	4
1.2.2 常用操作系统及其设备驱动介绍	5
1.3 嵌入式体系底层硬件操作软件及设计	8
1.3.1 嵌入式体系的底层硬件软件概述	8
1.3.2 常用 E-RTOS 及其软件体系设计	10
1.3.3 嵌入式体系中的可编程逻辑设计	14
1.3.4 嵌入式软件体系架构的考虑要素	14
本章小结	15

第 2 章 Windows 底层硬件的软件设计

2.1 Windows 底层硬件驱动及其软件开发设计概述	18
2.1.1 Windows 系统构造及其底层硬件驱动概述	18
2.1.2 Windows 底层硬件设备驱动软件开发综述	22
2.2 用 WinDDK 开发设计 Windows 设备驱动软件	27
2.2.1 WinDDK 设备驱动程序的软件编写	27
2.2.2 WinDDK 设备驱动程序的编译构建	30
2.2.3 WinDDK 设备驱动程序的检查验证	32
2.2.4 WinDDK 设备驱动程序的安装/调试	33
2.2.5 WinDDK 设备驱动程序的测试/使用	34
2.3 用 DriverStudio 开发设计 Windows 设备驱动软件	35
2.3.1 DriverStudio 设备驱动软件开发设计概述	35
2.3.2 DriverStudio 设备驱动程序的编译与装载	35
2.3.3 使用 DriverStudio 快速开发设备驱动软件	36
2.4 用 WinDriver 开发设计 Windows 设备驱动软件	37
2.4.1 WinDriver 设备驱动程序开发工具概述	37



2.4.2	主要 WinDriver 数据结构和 API 函数介绍	38
2.4.3	用 WinDriver 编程向导快速开发驱动程序	40
2.4.4	直接利用 WinDriver 的 API 函数开发驱动程序	41
2.4.5	WinDriver 开发的驱动程序的分发与应用	41
2.5	通过常见 Windows 通信接口进行数据传输设计	42
2.5.1	在 Windows 下通过异步串行口传输数据	42
2.5.2	在 Windows 下通过并行接口传输数据	54
2.5.3	通过 Winsock 编程接口实现以太网网络通信	60
2.6	USB 接口硬件设备的 Windows 驱动软件设计	68
2.6.1	USB 体系及其 WDM 型驱动程序结构	68
2.6.2	USB 硬件设备驱动程序应用设计举例	70
2.7	ISA/PC104 接口板卡的 Windows 驱动软件设计	72
2.7.1	ISA/PC104 接口板卡及其驱动程序设计概述	72
2.7.2	ISA/PC104 板卡硬件驱动程序设计举例	72
2.8	PCI/CPCI 接口板卡的 Windows 驱动软件设计	75
2.8.1	PCI/CPCI 板卡硬件设备驱动程序的特点	75
2.8.2	常见 PCI/CPCI 板卡驱动程序的开发设计	76
2.8.3	DMA 传输的 PCI/CPCI 板卡驱动程序设计	82
2.8.4	PCI/CPCI 板卡驱动程序的调用与调试	86
	本章小结	87

第 3 章 基于 Linux 操作系统底层硬件的软件设计

3.1	Linux 硬件驱动及其软件开发设计概述	89
3.1.1	Linux 下的硬件设备驱动概述	89
3.1.2	Linux 硬件驱动软件开发设计基础	93
3.2	字符型硬件设备的驱动程序软件设计	101
3.2.1	字符型硬件设备驱动综述	101
3.2.2	向系统中添加字符型设备	101
3.2.3	字符型设备驱动软件设计举例	102
3.3	块型硬件设备的驱动程序软件设计	106
3.3.1	块型硬件设备驱动综述	106
3.3.2	向系统中添加块型设备	107
3.3.3	块型设备驱动程序的设计	109
3.4	网络型硬件设备的驱动程序软件设计	114
3.4.1	网络设备驱动程序的运行机理概述	114
3.4.2	网络型设备驱动程序的具体实现	115
3.4.3	网络设备驱动程序的应用设计举例	116
3.5	常见硬件的 Linux 硬件驱动软件设计	118
3.5.1	在 Linux 下进行异步串行数据传输	118

3.5.2 在 Linux 下通过并行接口传输数据	121
3.5.3 Socket 接口的以太网数据传输	125
3.5.4 USB 外设的 Linux 驱动软件设计	131
3.5.5 ISA/PC104 板卡的 Linux 驱动设计	140
3.5.6 PCI/CPCI 板卡的 Linux 驱动设计	146
3.6 用 WinDriver 开发 Linux 设备驱动程序	153
3.6.1 WinDriver for Linux 开发工具简介	153
3.6.2 应用 WinDriver 快速开发驱动程序	154
3.6.3 WinDriver 驱动程序的分发与应用	155
本章小结	156

第 4 章 VxWorks 底层硬件的软件设计

4.1 VxWorks 底层硬件驱动及其开发设计概述	158
4.1.1 VxWorks 操作系统及其体系结构	158
4.1.2 VxWorks 的 BSP 及其开发设计	160
4.1.3 VxWorks 设备驱动程序及其开发设计	163
4.1.4 Tornado IDE 及其 VxWorks 程序设计	166
4.2 字符型硬件设备的驱动程序软件设计	168
4.2.1 字符型硬件设备及其驱动综述	168
4.2.2 字符型设备驱动程序的访问过程	171
4.3 块型设备驱动程序设计及其文件系统操作	171
4.3.1 块型硬件设备及其驱动程序综述	171
4.3.2 块型硬件设备支持的文件系统概述	174
4.3.3 块型设备驱动编写举例——电子盘操作	174
4.4 常见通信接口的 VxWorks 数据传输实现	177
4.4.1 在 VxWorks 下通过异步串口传输数据	177
4.4.2 在 VxWorks 下通过并行接口传输数据	180
4.4.3 以 Socket 编程接口实现网络传输数据	182
4.5 USB 接口设备的 VxWorks 驱动软件设计	186
4.5.1 USB 协议栈及其驱动层次结构概述	186
4.5.2 VxWorks 下的核心驱动 USBD 详解	187
4.5.3 VxWorks 下的 USB 设备驱动及应用	191
4.6 ISA/PC104 板卡的 VxWorks 驱动软件设计	192
4.6.1 ISA 接口设备 VxWorks 驱动设计概述	192
4.6.2 ISA/PC104 板卡设备的驱动设计举例	192
4.7 PCI/CPCI 板卡的 VxWorks 驱动软件设计	194
4.7.1 PCI/CPCI 板卡的驱动程序设计综述	194
4.7.2 PCI/CPCI 板卡的驱动程序设计举例	196
4.8 用 WinDriver 开发 VxWorks 设备驱动程序	197



4.8.1 WinDriver for VxWorks 开发工具介绍	198
4.8.2 用 WinDriver 开发 VxWorks 驱动程序	199
本章小结	199

第5章 嵌入式基本体系及外设接口的直接软件架构

5.1 嵌入式应用系统的直接软件架构概述	201
5.1.1 嵌入式应用系统的直接软件架构	201
5.1.2 嵌入式系统直接软件架构的特点	202
5.2 嵌入式单片机基本体系的软件架构设计	203
5.2.1 嵌入式单片机体系的软件架构综述	203
5.2.2 嵌入式单片机体系的直接软件架构	203
5.3 嵌入式 DSPs 基本体系的软件架构设计	218
5.3.1 嵌入式 DSPs 体系的软件架构综述	218
5.3.2 嵌入式 DSPs 体系的直接软件架构	218
5.4 嵌入式体系中的接口直接驱动软件设计	224
5.4.1 嵌入式体系硬件接口及其驱动概述	224
5.4.2 常见嵌入式接口的直接驱动软件设计	225
5.5 嵌入式体系中的外设直接驱动软件设计	245
5.5.1 嵌入式体系硬件外设及其驱动概述	245
5.5.2 常见嵌入式外设的直接驱动软件设计	245
5.6 嵌入式体系外设与接口的驱动程序测试	266
5.6.1 外设与接口驱动程序测试概述	266
5.6.2 外设与接口驱动测试软件编制	267
5.7 使用软件架构工具快速构建应用软件平台	270
5.7.1 常用嵌入式体系软件架构工具介绍	270
5.7.2 嵌入式体系软件架构工具应用举例	270
本章小结	279

第6章 嵌入式 μ C/OS 基本体系及外设接口的软件架构

6.1 μ C/OS 嵌入式实时操作系统概述	280
6.1.1 μ C/OS 操作系统简要介绍	280
6.1.2 μ C/OS 下的多任务信息流	281
6.1.3 μ C/OS 的任务调度与切换	284
6.1.4 μ C/OS 的中断处理与优化	285
6.1.5 μ C/OS 软件体系的利弊分析	286
6.2 嵌入式 μ C/OS 基本软件体系架构	288
6.2.1 μ C/OS 基本软件体系综述	288
6.2.2 μ C/OS 下的 C 语言编程	289
6.2.3 μ C/OS 移植的方法技巧	291

6.2.3 μ C/OS 移植的关键技术阐述	295
6.3 常见嵌入式体系的 μ C/OS 移植	300
6.3.1 SCM 体系的 μ C/OS 移植	300
6.3.2 DSPs 体系的 μ C/OS 移植	314
6.4 μ C/OS 下的外设/接口驱动设计	315
6.4.1 外设接口驱动设计综述	315
6.4.2 典型外设接口驱动设计	315
6.5 μ C/OS 下的文件系统及存取访问	322
6.5.1 μ C/FS 文件系统及其应用	322
6.5.2 EMFS 文件系统及其应用	323
6.6 μ C/OS 嵌入式软件体系架构应用	325
6.6.1 数据采集/传输系统软件架构	325
6.6.2 总线式数据采集软件体系架构	327
本章小结	329

第 7 章 嵌入式 DRTOS 基本体系及外设接口的软件架构

7.1 DRTOS 嵌入式实时操作系统综述	330
7.1.1 DRTOS 嵌入式操作系统概述	330
7.1.2 嵌入式 DSP/BIOS 体系综述	332
7.2 嵌入式 DSP/BIOS 基本软件体系架构	338
7.2.1 嵌入式 DSP/BIOS 软件体系开发	338
7.2.2 DSP/BIOS 的配置工具及其使用	338
7.2.3 DSP/BIOS 文件及其编译与链接	340
7.2.4 DSP/BIOS 启动序列及自举引导	341
7.2.5 DSP/BIOS 软件的调试与监测	343
7.3 DSP/BIOS 下的外设/接口驱动软件设计	344
7.3.1 DSP/BIOS 外设接口驱动设计概述	344
7.3.2 DSP/BIOS 典型 I/O 数据传输设计	345
7.3.3 DSP/BIOS 典型网络通信操作设计	348
7.3.4 DSP/BIOS 类/微型驱动程序设计	351
7.4 DSP/BIOS 嵌入式软件体系架构应用	358
7.4.1 DSP/BIOS 数据采集体系软件架构	358
7.4.2 DSP/BIOS 图像处理体系软件架构	359
7.4.3 DSP/BIOS 机顶盒多任务调度架构	361
本章小结	364

第 8 章 嵌入式 WinCE/XPE 基本体系及外设接口的软件架构

8.1 WinCE/XPE 嵌入式操作系统综述	366
8.1.1 WinXPE 及软件体系开发概述	367



8.1.2	WinCE 及软件体系开发简介	367
8.1.3	WinCE 体系结构与功能综述	369
8.1.4	WinCE 下应用软件开发总览	371
8.2	定制 WinCE 嵌入式基本软件体系	372
8.2.1	WinCE 定制的一般设计流程	372
8.2.2	PB/组件/WinCE 及构建详述	373
8.2.3	简单示例:定制并运行 CEPC	374
8.3	移植 WinCE 嵌入式实时操作系统	375
8.3.1	WinCE 运行的硬件需求	375
8.3.2	WinCE BSP 及开发设计	375
8.3.3	WinCE 引导程序的编写	376
8.3.4	WinCE OAL 程序的编制	378
8.4	WinCE 的设备驱动程序及其设计	380
8.4.1	WinCE 设备驱动程序综述	380
8.4.2	WinCE 设备驱动程序设计	384
8.4.3	WinCE 设备驱动设计举例	385
8.4.4	开发与测试设备驱动程序	393
8.5	WinCE USB 设备驱动程序及设计	394
8.5.1	WinCE USB 软件体系综述	394
8.5.2	编写 WinCE USB 驱动程序	395
8.5.3	简单示例:USB 鼠标驱动	396
8.6	WinCE NDIS 网络设备驱动及设计	401
8.6.1	WinCE NDIS 网络驱动概述	401
8.6.2	WinCE 微端口驱动及其实现	402
8.7	WinCE 块型设备驱动及文件系统操作	407
8.7.1	WinCE 的块型设备驱动综述	407
8.7.2	块型设备系统体系及文件系统	408
8.7.3	实现 WinCE 块型设备驱动程序	409
8.8	常用的 WinCE 数据通信及其实现	411
8.8.1	WinCE 下的通信模型综述	411
8.8.2	WinCE 串行数据通信实现	412
8.8.3	WinCE 网络数据通信实现	415
	本章小结	421

第 9 章 嵌入式 Linux 基本体系及外设接口的软件架构

9.1	Linux 嵌入式实时操作系统综述	422
9.1.1	Linux 嵌入式操作系统概述	422
9.1.2	嵌入式 μ CLinux 体系综述	423
9.2	μ CLinux 开发环境的建立及其移植	425

9.2.1	μ CLinux 开发环境简介	425
9.2.2	建立 μ CLinux 开发环境	427
9.2.3	μ CLinux 的芯片级移植	428
9.3	μ CLinux 设备驱动程序及设计综述	435
9.3.1	μ CLinux 设备驱动程序概述	435
9.3.2	μ CLinux 内核模块基本框架	436
9.3.3	Makefile 文件及其基本框架	436
9.4	μ CLinux 字符型设备驱动程序设计	437
9.4.1	字符型设备驱动的整体架构设计	437
9.4.2	相关接口操作的函数代码编写	438
9.4.3	底层中断及其处理程序的设计	441
9.4.4	编译指导文件 Makefile 的编制	442
9.4.5	字符型设备驱动的应用程序调用	443
9.5	μ CLinux 块型设备驱动与闪存文件操作	444
9.5.1	嵌入式块驱动及文件操作概述	444
9.5.2	μ CLinux 的块型设备驱动程序设计	444
9.5.3	闪存 Flash 驱动及文件系统操作	448
9.6	μ CLinux 的网络设备驱动及网络通信	450
9.6.1	μ CLinux 网络设备驱动程序设计	450
9.6.2	基于 μ CLinux 的 Socket 网络通信	459
	本章小结	460

第 10 章 嵌入式 VxWorks 基本体系及外设接口的软件架构

10.1	嵌入式 VxWorks 软件体系架构基础	461
10.1.1	VxWorks 体系结构及设备驱动	461
10.1.2	VxWorks 的 BSP 及其开发设计	463
10.1.3	Tornado 开发工具及其 IDE 简介	463
10.2	VxWorks 内核移植及 BSP 软件编写	464
10.2.1	VxWorks 操作系统的移植过程	464
10.2.2	S3C4510B VxWorks BSP 开发	464
10.2.3	LPC2104 VxWorks BSP 设计	467
10.3	VxWorks 下字符型设备驱动软件设计	472
10.3.1	字符型设备驱动及其设计简述	472
10.3.2	字符型设备驱动程序软件框架	473
10.3.3	字符型设备驱动设计应用举例	477
10.4	VxWorks 下块型设备驱动及文件系统架构	479
10.4.1	块型设备驱动与文件系统操作概述	479
10.4.2	闪存介质 CF 卡及 TFFS 操作	480
10.4.3	TFFS 构建与大容量闪存操作	482



10.5 VxWorks 下的异步串口驱动程序设计	486
10.5.1 VxWorks 异步串口驱动概述	486
10.5.2 串口驱动程序设计流程分析	487
10.5.3 示例:编写 S3C2410 串口驱动	490
10.6 VxWorks 下的网络设备驱动及其实现	496
10.6.1 VxWorks 网络设备驱动综述	496
10.6.2 END 设备驱动程序及其编写	499
10.6.3 示例:RT8139C 网络接口驱动	502
本章小结	504

第 11 章 硬件外设/接口及其片上系统的可编程软件实现

11.1 外设/接口及其片上系统软件实现综述	505
11.1.1 软件实现外设/接口及其片上系统	505
11.1.2 硬件设施软件实现应用技术简介	506
11.2 可编程实现常见外设/接口及简易系统	515
11.2.1 嵌入式应用体系的外存模块设计	515
11.2.2 总线接口的时序逻辑变换实现	518
11.2.3 常见外设/接口的 PLD 简易实现	521
11.2.4 专用外设/接口的 PLD 简易实现	523
11.2.5 简单测量/控制体系的可编程实现	529
11.3 外设/接口的片上可编程软件配置实现	538
11.3.1 PSD 外设/接口的灵活软件实现	538
11.3.2 μ PSD 及其片内外设/接口的应用	543
11.3.3 PSoC 及其片内外设/接口的应用	544
11.4 模拟硬件外设/接口的可编程软件设计	547
11.4.1 ispPAC 系列器件及应用设计简介	547
11.4.2 用 ispPAC 器件设计模拟外设/接口	548
11.5 特定 DSP 算法的 FPGA 可编程设计	553
11.5.1 DSP Builder 及其 DSP 设计简介	554
11.5.2 System Generater 及 DSP 实现综述	554
11.5.3 典型 DSP 算法的 FPGA 实现举例	556
11.6 嵌入式体系的 FPGA - SoPC 实现技术	560
11.6.1 常用 FPGA - SoPC 实现技术综述	561
11.6.2 FPGA - SoPC 技术应用设计实践	565
本章小结	575

第 12 章 基于底层硬件的软件设计实践

12.1 在项目设计中规划基于底层硬件的软件架构	577
12.1.1 基于底层硬件体系软件架构的总体考虑	577

12.1.2	嵌入式应用体系软件架构的规划设计·····	578
12.1.3	通用计算机通信相关的设备驱动设计·····	579
12.1.4	特定应用系统的数据通信规约及其制订·····	580
12.2	铁路道岔运行状况监控系统的软件体系架构·····	581
12.2.1	项目构成及软件架构的主要环节综述·····	581
12.2.2	关键性子系统的软件体系架构及实现·····	583
12.3	交流电机伺服驱动监控系统的软件体系架构·····	599
12.3.1	项目系统组成及其需要架构的软件体系·····	599
12.3.2	上/下位软件体系之间的通信及其规约·····	599
12.3.3	交流电机伺服控制器系统的软件架构·····	600
12.3.4	上位机数据传输通信软件体系的构造·····	615
12.4	μ Linux 下的 ARM 与 DSPs 的数据通信实现·····	620
12.4.1	项目体系的构造及关键硬件电路组成·····	620
12.4.2	ARM - Linux 下的 HPI 接口驱动设计·····	621
12.5	嵌入式 RTOS 下跨平台通信体系的软件架构·····	624
12.5.1	E - RTOS 体系跨平台通信的整体设计·····	624
12.5.2	E - RTOS 跨平台通信的部分代码示例·····	626
12.6	基于 FPGA - SoPC 的 MP3 播放器及软件架构·····	628
12.6.1	系统的总体框架设计及其功能描述·····	628
12.6.2	FPGA - SoPC 的软硬件协同设计实现·····	629
12.7	基于底层硬件的软件设计参考书籍推荐·····	632
	本章小结·····	633
	参考文献·····	634

第1章 基于底层硬件的软件设计概述

怎样在硬件基础上搭建基本的软件平台？怎样通过软件设计去操作、访问并控制硬件？怎样为实现测量、控制、监控、通信、数字助理等应用程序设计构造一个稳定、高速、实用的最小基本软件环境？阅读本章，将会有有一个全面的了解和初步的体验。

本章的主要内容如下：

- ▶ 底层硬件操作软件及设计的总体阐述；
- ▶ 通用计算机底层硬件操作软件及设计；
- ▶ 嵌入式体系底层硬件操作软件及设计。

1.1 底层硬件操作软件及设计的总体阐述

1.1.1 底层硬件操作软件的综合阐述

基于底层硬件的操作软件，泛指能够控制相关硬件的功能行为、对其进行读/写访问或数据接收/发送传输操作的程序代码集，通常也称为硬件操作软件或者基于硬件的软件。

一般来说，硬件操作软件所基于的硬件通常是软硬件系统的核心中央处理器 CPU(Central Processing Unit)及其外围设备或各类接口器件。CPU 外围设备，如数据存储器 RAM、程序存储器 ROM、模/数转换器 ADC、数/模转换器 DAC、定时/计数器及各类总线控制器等，在通用计算机或嵌入式系统中，简称为外设；一些常用外设如定时/计数器、看门狗定时器、快速 RAM 等与 CPU 集成在一块芯片中，称为片内外设，其余没有与 CPU 集成在一起的需要在设计时进行外围扩展的外设称为片外设备。接口器件，如串口控制转换器、PCI 桥件及 USB 控制器等，通常用于实现系统的人机界面、与外界的各种通信等。从软件角度对这些外设或接口的控制与操作就是对其进行驱动，基于 CPU 外设或接口的软件设计主要是 CPU 外设或接口的设备驱动程序设计。另外，在嵌入式应用系统中还常常涉及用于实现各种接口逻辑变换的可编程器件逻辑设计，这种可编程数字或模拟逻辑器件 PLD(Programmable Logic Device)或 PAC(Programmable Analog Device)设计也是基于底层硬件的软件设计中不可缺少的部分，进行必要的 PLD 或 PAC 设计可以有效地减少系统软硬件设计的难度，做到系统资源更加合理的支配。

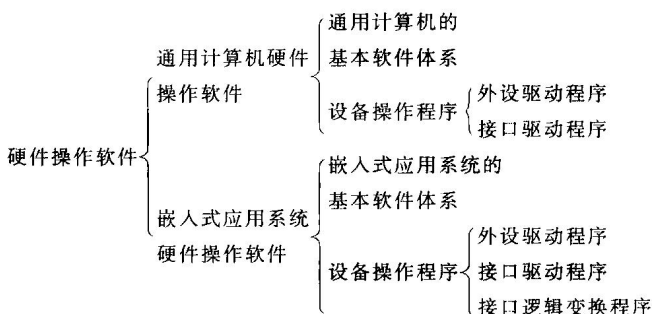
基于底层硬件的软件设计主要是设备驱动程序设计，但是基于底层硬件的软件并不完全是设备驱动程序，还包括基于 CPU 结构的基本软件体系架构。基于 CPU 结构的基本软件体系架构，就是建立能够进行系统应用软件设计所需的最小的可靠软件运行环境，一般包括时钟管理、工作模式设置、外设配置、接口初始化、多任务分配及中断设置等。应用软件是能够实现一定功能的程序代码集。为了系统的安全可靠和软件维护的方便，通常应用软件不直接控制或操作硬件，而是通过设备驱动程序来控制或操作硬件。实际应用中的 CPU 体系主要有用



户监控的通用计算机系统和嵌入式应用系统两种,常见的通用计算机主要是个人计算机 PC (Personal Computer)、工控机(industrial control computer)等。在工业数据采集和控制应用领域以工控机最为常见;常见的嵌入式应用系统主要是以单片机 SCM(Single Chip Micro-computer)或/和信号处理器 DSPs(Digital Signal Processor)为核心微控制/处理器的系统,也可以是它们的“缩形”——基于现场可编程门阵列(FPGA Field Programmable Gate Array)的可编程片上系统 SoPC。通用计算机的基本软件体系可以采用常见的 Windows、Linux、Vx-Works 等操作系统以最小系统配置构成,嵌入式应用系统的基本软件架构可以采用基于相应 CPU 指令体系的直接软件架构或采用基于 $\mu\text{C}/\text{OS}$ 、DSP/BIOS、WinCE/XP、 μCLinux 、Vx-Works 等嵌入式实时操作系统以最小系统配置构成。

综上所述,基于底层硬件的软件结构组成可以用表 1.1 所列结构进行简单的概括。

表 1.1 基于底层硬件的软件组成



在各类软硬件应用体系产品中,硬件体系是整个系统赖以存在的基础,应用软件或程序使整个系统具有特定的功能和特征,基于硬件的软件是联系底层硬件体系和顶层应用软件的纽带和桥梁。有了底层硬件操作软件,整个软硬件体系才是一个有机整体。

1.1.2 底层硬件操作软件的层次组织

实际应用及软硬件设计实践中,各类基于底层硬件的软件的层次组织构成可以简单地用图 1.1 表示。图中的软硬件应用系统,可以是工控机、个人计算机、便携式计算机等,也可以是工业板卡、智能终端、个人数字助理 PDA(Personal Digital Assistant)、便携式仪器仪表等嵌入式应用系统;图中的数据传输通道,可以是异步串口、并行口、局域网接口、USB 总线接口、ISA/PC104 总线接口、PCI/CPCI 接口及各类现场总线接口等。

这种基于底层硬件的软件层次组织的活动过程可以简单描述为:基于系统核心 CPU 的基本软件体系的应用程序通过各类外设或接口的逻辑变换并驱动进行数据存取操作,然后再通过选定的接口逻辑变换并驱动与其他的软硬件系统,进行数据交换。

有很多这样的层次组织,如:数码相机捕捉感觉影像,再通过 USB(Universal Serial Bus)接口把图像传送到个人计算机,个人计算机做图像处理,最后存储到非易失设备或送打印机输出照片;其中,数码相机、打印机是嵌入式应用体系,个人计算机是计

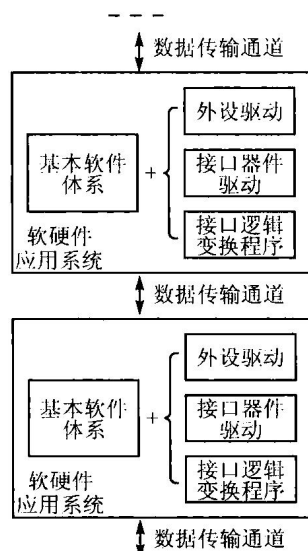


图 1.1 基于底层硬件的软件层次组织图

算机系统,个人计算机与数码相机之间的数据通道是 USB 接口,个人计算机与打印机之间的数据通道是并行口。又如:CPCI 数据采集/控制卡的采集数据,通过 PCI 总线,传送给工控机进行处理和显示,工控机再通过 PCI 总线,负责对 CPCI 卡进行初始配置和控制;其中,CPCI 卡是一个独立的嵌入式应用系统,工控机是计算机监控系统,PCI 总线是两者之间的快速数据传输通道。

1.1.3 基于底层硬件的操作软件设计

基于底层硬件的软件设计,主要有 3 种类型:基本软件体系构建、底层外设或接口驱动程序设计和硬件接口逻辑变换程序设计。此外,还可以是这三者的有机整合——可编程片上系统设计 SoPC。

1. 基本软件体系的构建

基本软件体系的构建,主要是针对特定的 CPU 而构建的系统,能够初步运行和完成要求功能的基本软件平台,包括系统启动、时钟配置、所用操作系统的构建、所需外设或接口的初始化及中断与任务的分配和启动等。对于通用计算机系统,如常用的 Intel 公司的 X86 与 Pentium 系列计算机,由于有足够容量的内存和硬盘,直接自定义安装最小化的操作系统即可,操作系统如 Windows、Linux 等。对于嵌入式应用系统,当前主流应用是单片机体系和 DSPs 体系,主要是启动文件、主程序文件、外设与接口初始化配置文件、中断处理文件及实时监控文件等的基本架构;采用嵌入式操作系统的,如 μ CLinux、VxWorks 等,还包括操作系统内核的最小选配或嵌入移植。

基本软件体系的构建,主要是嵌入式应用系统的基本软件体系构建。

2. 底层外设或接口驱动程序设计

底层外设或接口的驱动程序设计,主要是相关外设或接口器件的初始化配置、行为控制、基本的“读/写”访问或“收/发”操作等,及面向应用程序提供界面友好的 API(Application Programming Interface)调用接口。

初始化配置和行为控制主要用于设置硬件的工作模式、数据传输格式、通信速率及返回状态信息规定等。初始化配置只在系统启动时由系统基本软件体系调用,行为控制通常在执行过程中用以改变硬件的行为规则。

“读/写”访问或“收/发”操作的方式一般有两种:查询方式和中断方式。查询方式需要软件不断地进行状态字或状态位的查询,直到有效事件发生为止;中断方式无需软件干预,只要有效事件发生,就通过硬件通知软件。“写”访问或发送操作通常是主动的,只要硬件空闲,就可以使其执行,通常采用查询方式;“读”访问或接收操作是被动的,硬件何时得到或产生有效数据是不确定的,通常采用中断方式。

3. 硬件接口逻辑变换程序设计

硬件接口的逻辑变换程序设计,就是对完成硬件接口逻辑转换的 PLD 或 PAC 器件进行可编程逻辑程序的设计。

PLD 器件主要用于以软件编程形式灵活地实现数字信号的接口逻辑变换,如操作时序变换、状态等待插入、总线形式转换、工况查询、统计分析、PCI 接口逻辑变换、脉冲计数转速分析、屏幕扫描显示及简易键盘扫描编码等。PLD 编程语言多采用 VHDL 或接近 C 语言的 Verilog。近年来的 PLD 设计,还常常借助于 Matlab 数学运算工具和特定的 DSP 逻辑设计工