

基于VxWorks 的嵌入式实时系统设计

王金刚 宫霄霖 杨锡勋 编著
苏 淇 丁大尉 姜 平



清华大学出版社

基于 VxWorks 的嵌入式实时系统设计

王金刚 宫霄霖 杨锡劭

编著

苏 淇 丁大尉 姜 平

清华大学出版社

北 京

内 容 简 介

本书是一本介绍嵌入式系统原理和应用的教材,从介绍嵌入式概况入手,以 VxWorks 实时操作系统为例,讲述了 VxWorks 的原理及其运行机制、Tornado 开发工具的使用、嵌入式实时系统 BSP 结构、嵌入式实时系统网络编程以及以 SC4510B ARM 为硬件的开发系统。

本书用作通信工程、电子信息等专业的研究生学习嵌入式系统的教材,同时可以作为电子信息学科和航空控制类本科生高年级的教材,也可作为从事嵌入式开发的工程人员的参考书。

版权所有,翻印必究。举报电话:010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

基于 VxWorks 的嵌入式实时系统设计/王金刚等编著. —北京:清华大学出版社,2004.10

ISBN 7-302-08847-0

I. 基… II. 王… III. 实时操作系统, VxWorks-系统设计-教材 IV. TP316.2

中国版本图书馆 CIP 数据核字(2004)第 057397 号

出 版 者:清华大学出版社

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

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:曾 刚

文稿编辑:吕春龙

封面设计:秦 铭

版式设计:俞小红

印 装 者:北京鑫霸印务有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:19.25 字数:427 千字

版 次:2004 年 10 月第 1 版 2004 年 10 月第 1 次印刷

书 号:ISBN 7-302-08847-0/TP·6277

印 数:1~4000

定 价:29.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

前 言

VxWorks 操作系统是美国 WindRiver 公司于 1983 年设计开发的一种可以运行在目标机上的高性能、可裁剪的嵌入式实时操作系统 (RTOS)。它由于具有开放式的结构、支持广泛的工业标准、持续发展的能力、高性能的内核、友好的开发环境、卓越的可靠性和实时性,被广泛应用于通信、军事、航空、航天等实时性要求极高的领域中,在嵌入式实时操作系统领域占据了一席之地。

VxWorks 系统是专为嵌入式实时应用而设计的模块化的实时操作系统,为用户提供了大量的系统调用,作为 VxWorks 系统的主要组成部分的板级支持包 BSP (Board Support Package),对各种板子的硬件功能提供了统一的软件接口。对于用户来说,一个实时应用软件是由板级支持包 BSP、操作系统内核及用户选用组件、中断服务程序 ISR 组成,所以开发者可以根据需要进行系统模块的定制,只需做最少的工作即可设计出有效的适合于不同用户的产品,因而 VxWorks 系统越来越受到消费电子行业广大开发人员的青睐。

考虑到 VxWorks 操作系统是嵌入式开发环境的关键部分,以及 Tornado 是嵌入式实时领域里最新一代的开发调试环境,本教材是基于最新版本的 VxWorks5.5 和 Tornado2.2 开发环境进行编写的。本书第 1 章对嵌入式系统进行了简要的概述,随后的几章详细地讨论了 VxWorks 系统、Tornado 开发环境、BSP 的编写、网络编程以及在 S3C4510 微控制器开发平台上进行开发等几个部分。其中第 2 章简要介绍嵌入式 VxWorks 实时操作系统,包括系统的内核、任务的管理和通信、同步和互斥的机制、中断和定时机制以及文件系统等内容;第 3 章阐述了 Tornado 的集成开发环境,介绍了面向嵌入式系统的开发和调测工具;第 4 章讲述了如何创建 BSP 和组件,以及 BSP 驱动程序的编写;第 5 章讨论了网络编程的知识,包括 VxWorks 网络协议栈和网络服务等内容;最后,介绍了 S3C4510 微控制器的结构和开发平台,并详细讲解了如何移植 VxWorks 操作系统。总之,本书理论联系实际,既强调对嵌入式操作系统基本理论知识的理解,又非常注重实际的应用。所以本书不仅适合于初学者学习使用,而且对于进行 VxWorks 操作系统移植和网络编程的工作者也有非常大的帮助。

参加编写工作的还有毛战华、王达心、李蕊、孙效闻、秦承虎、王铮、毕晓东、左国辉等。

由于编写人员的水平有限,书中难免出现不足之处,欢迎广大读者提出批评和修改建议。

作 者

2004 年 1 月

目 录

第 1 章 绪论	1
1.1 嵌入式系统概述	1
1.2 嵌入式系统的发展历史	2
1.3 嵌入式系统的体系结构	3
1.3.1 嵌入式处理器	3
1.3.2 嵌入式外围设备	5
1.3.3 嵌入式应用软件	5
1.3.4 嵌入式操作系统	6
1.4 典型的嵌入式操作系统	7
1.5 嵌入式系统的应用领域	11
第 2 章 嵌入式实时操作系统 VxWorks	12
2.1 简介	12
2.2 VxWorks 任务及任务间通信	13
2.2.1 多任务机制	13
2.2.2 任务状态转变	14
2.2.3 Wind 任务调度	15
2.2.4 任务控制	17
2.2.5 任务扩展函数	20
2.2.6 任务异常处理	20
2.2.7 共享代码和重入	21
2.2.8 共享数据结构	22
2.2.9 互斥	22
2.2.10 信号量	23
2.2.11 消息队列	27
2.2.12 管道	28
2.2.13 任务间网络通信	29
2.2.14 信号 (signal)	30
2.3 VxWorks 事件	31
2.3.1 pSOS 事件	32
2.3.2 VxWorks 事件	33
2.3.3 API 比较	35

2.4	文件系统.....	36
2.4.1	文件、设备和驱动程序	36
2.4.2	基本 I/O 接口	37
2.4.3	缓冲型 I/O 设备: stdio	38
2.4.4	异步输入/输出操作	38
2.4.5	VxWorks 操作系统中的设备	39
2.4.6	与 MS-DOS 兼容的文件系统: dosFs 文件系统	41
2.4.7	使用 SCSI 设备从本地 dosFs 文件系统启动	43
2.4.8	原始文件系统: rawFs 文件系统	43
2.4.9	磁带文件系统: tapeFs 文件系统	44
2.4.10	CD-ROM 文件系统: cdromFs	45
2.4.11	目标服务器文件系统: TSFS	46
2.5	中断处理与定时机制	46
2.5.1	看门狗定时机制	46
2.5.2	中断服务程序	48
2.5.3	POSIX 时钟和计时器	51
2.5.4	POSIX 内存上锁接口	52
2.5.5	POSIX 与 Wind 的比较	53
2.6	目标机工具	54
2.6.1	基于目标机的 shell	54
2.6.2	基于目标机的加载器	55
2.6.3	基于目标机的符号表	56
2.7	基于 VxWorks 的 C++ 语言开发	59
2.7.1	VxWorks 系统下使用 C++ 语言	59
2.7.2	初始化和确定静态目标	60
2.7.3	使用 GNU C++ 编程	61
2.7.4	使用 Diab C++ 编程	65
2.7.5	使用 C++ 库	66
2.7.6	运行事例演示	67
第 3 章	基于图形用户界面的集成开发环境 Tornado 及开发实例	69
3.1	Tornado 简介	69
3.1.1	Tornado 交叉开发环境	70
3.1.2	VxWorks 目标机环境	71
3.1.3	Tornado 主机集成开发环境	71
3.1.4	主机与目标机的接口	73
3.2	Tornado 的安装和启动	74
3.2.1	Tornado 初步	74

3.2.2	建立主机环境	76
3.2.3	建立默认的目标机硬件环境	77
3.2.4	主机与目标机的通信配置	78
3.2.5	启动 VxWorks	79
3.2.6	启动 Tornado	82
3.2.7	启动目标服务器	84
3.2.8	显示目标机信息	85
3.3	Tornado 环境下的工程开发	86
3.3.1	设计用户工程	86
3.3.2	创建可下载的应用程序	89
3.3.3	创建自定义的 VxWorks 映像	89
3.3.4	创建可启动的应用程序	90
3.3.5	配置目标机与主机间通信接口	91
3.3.6	配置和构建 VxWorks 启动程序	92
3.4	Tornado 目标服务器	93
3.4.1	简介	94
3.4.2	配置和启动目标服务器	94
3.4.3	选择目标服务器	95
3.4.4	管理目标服务器	96
3.4.5	停止目标服务器	97
3.5	Tornado 主机工具	97
3.5.1	shell	97
3.5.2	调试器	101
3.5.3	浏览器	103
3.6	仿真器 VxSim	105
3.6.1	简介	105
3.6.2	集成仿真器	106
3.6.3	构建应用程序	107
3.6.4	配置 VxSim 全功能仿真器	107
第 4 章	板级支持包 BSP 基础	110
4.1	BSP 简介	110
4.2	BSP 组织结构	110
4.2.1	BSP 源文件及包含文件	111
4.2.2	生成文件	121
4.2.3	必要函数	122
4.2.4	可选函数	123
4.3	VxWorks 引导顺序	124

4.3.1	引导过程概述	124
4.3.2	相关文件及函数	124
4.4	创建一个 BSP	127
4.4.1	建立开发环境	128
4.4.2	编写 BSP 文件	129
4.4.3	编译和下载 VxWorks	132
4.4.4	调试初始化代码	133
4.4.5	启动 WDB 服务器	135
4.4.6	最小化内核	137
4.4.7	其他相关问题	138
4.5	组件	141
4.5.1	组件描述语言	142
4.5.2	创建组件	151
4.6	硬件设计指导	158
4.6.1	体系结构	158
4.6.2	内存	159
4.6.3	总线	160
4.6.4	设备	163
4.7	编写驱动程序	166
4.7.1	驱动程序规范及要求	166
4.7.2	驱动程序开发步骤	170
4.7.3	高速缓存相关问题	172
第 5 章	网络编程	176
5.1	引言	176
5.2	VxWorks 网络协议栈	176
5.2.1	支持的协议和工具	176
5.2.2	设定与网络任务相关的任务优先级	178
5.3	数据链路层组件	179
5.3.1	以太网驱动支持	179
5.3.2	共享内存的底板网络驱动	180
5.3.3	用户接口	190
5.4	网络协议栈的配置	190
5.4.1	配置设置摘要	190
5.4.2	创建时配置网络协议栈	191
5.4.3	配置 TCP/IP	201
5.4.4	网络配置协议	210
5.4.5	远程访问应用	216

5.5 VxWorks 套接字	222
5.5.1 BSD 套接字	223
5.5.2 ZBUF 套接字	236
第 6 章 S3C4510 微控制器开发平台	250
6.1 S3C4510 微控制器概述	250
6.1.1 S3C4510 微控制器特点及总体结构	250
6.1.2 S3C4510 微控制器内部结构	251
6.2 S3C4510B 微控制器开发平台	259
6.2.1 开发板总体结构	260
6.2.2 开发板各模块的功能介绍	260
6.3 开发板软/硬件调试	262
6.3.1 JTAG 调试接口	262
6.3.2 集成开发组件 ADS	264
6.3.3 开发板调试举例	267
6.4 VxWorks 操作系统移植	271
6.4.1 编写 BSP 软件	272
6.4.2 VxWorks 操作系统的启动过程	277
6.4.3 串行设备驱动程序	280
6.4.4 网络设备驱动程序	285
参考文献	295
结束语	296

第1章 绪 论

当今，在嵌入式领域，嵌入式技术（Embedded Technology）已经成为新的技术热点。嵌入式系统的最典型的特点是它同人们的日常生活紧密相关。小到 MP3、PDA 等微型数字化设备，大到信息家电、智能电器、车载 GPS 等形形色色运用了嵌入式技术的电子产品和各种新型嵌入式设备在数量上现已远远超过了通用计算机。在嵌入式设备发展的 30 多年的历史中，嵌入式技术从来没有像现在这样风靡过，人类也从来没有像现在这样享受嵌入式技术带来的便利。

1.1 嵌入式系统概述

嵌入式系统被定义为：以应用为中心，以计算机技术为基础，软硬件可裁剪，适合于应用系统对功能、可靠性、成本、体积、功耗方面要求严格的专用计算机系统。嵌入式计算机在应用数量上远远超过了一般的通用计算机，一台通用计算机的外部设备中就包含了 5~10 个嵌入式微处理器，键盘、鼠标、软驱、硬盘、显示卡、显示器、网卡、调制解调器、声卡、打印机、扫描仪、数字相机、USB 集线器等，它们均是由嵌入式处理器控制的。

嵌入式系统是将先进的计算机技术、半导体技术、电子技术和各个行业的具体应用相结合后的产物，这一特点决定了它必然是一个技术密集、资金密集、高度分散、不断创新的知识集成系统。由于嵌入式处理器的体系结构及其相关的片上外设、开发工具、库函数、嵌入式应用产品等是一套复杂的知识系统，用户和半导体厂商一般都不会轻易地放弃某种处理器，所以嵌入式处理器的发展也具有较强的稳定性。

嵌入式系统与具体应用是有机地结合在一起的，其升级换代也是与具体产品的开发同步进行的，因此嵌入式系统产品一旦进入市场，就具有较长的生命周期。嵌入式系统中的软件是实现嵌入式系统功能的关键，具体体现在系统软件的高实时性、软件固态化存储要求、软件代码高质量和高可靠性，以及多任务操作系统等方面。

嵌入式系统工业的基础是以应用为中心的“芯片”设计和面向应用的软件产品开发。嵌入式系统是面向用户、面向产品、面向应用的，如果独立于应用而自行发展则会失去市场。嵌入式处理器的功耗、体积、成本、可靠性、速度、处理能力、电磁兼容性等方面均受到应用要求的制约，这些也是各个半导体厂商之间竞争的热点。与通用计算机不同，嵌入式系统的硬件和软件必须高效率地设计，量体裁衣、去除冗余，力争在同样的硅片面积上实现更高的性能，这样才能在具体应用中对处理器的选择上更具竞争力。

从某种意义上来说,嵌入式系统工业是一个分散的工业,充满了竞争、机遇与创新,没有哪一个系列的处理器和操作系统能够垄断全部市场。即便在体系结构上存在着主流,但各自不同的应用领域决定了不可能少数公司、少数产品垄断全部市场。因此,嵌入式系统领域的产品和技术,必然高度分散,留给各个行业中小规模、高技术的公司创新余地很大。另外,社会上的各个应用领域在不断向前发展,必然要求嵌入式处理器核心同步发展,这也构成了推动嵌入式工业发展的强大动力。

1.2 嵌入式系统的发展历史

嵌入式系统的发展历史相当悠久,早在电子数字计算机出现之前就有了把计算装置嵌入到系统和设备中的嵌入式系统,那时的计算装置是由一些凸齿轮的机械量和电压电流的模拟量实现的。在 20 世纪 40 年代第一代电子计算机出现后的近 20 年里,这类计算机主要器件由电子管、晶体管、磁芯等组成,因体积庞大阻碍了它们在嵌入式系统中的发展。

计算机用于控制设备或嵌入式系统的历史几乎同电脑自身发展的历史一样长。在通信领域,20 世纪 60 年代晚期,计算机被用于电子电话交换机,称为“存储程序控制”系统。存储程序指内存装有程序和路由信息。用软件存储控制逻辑,而不是将其固化在硬件中,这在当时观念上是一种真正的突破,在今天看来,这种工作机理早已视为理所当然。按今天的标准来看,它们有着奇特的专用指令和特殊的 I/O 设备,但这已经足以标志嵌入式应用的诞生。

嵌入式应用的兴起是在 20 世纪 60 年代末集成电路化的第三代计算机时期,在这一时期计算机技术取得重大发展,例如,并行技术、先行控制、流水线、单总线结构、高速通用寄存器、强有力的中断系统、交叉存取、操作系统等。这些技术同时也应用到嵌入式计算机中来。

而嵌入式应用的真正发展却是在微处理器问世之后。在这个时期,嵌入式计算机的发展是伴随着 PC 的发展而发展的。微处理器的广泛应用形成了一个广阔的嵌入式应用市场,计算机厂家开始以插件的方式向用户大量提供 OEM 产品,再由用户根据自己的需要选择一套适合自己应用的 CPU 板、存储器板和各式 I/O 插件板构成专用的嵌入式计算机系统,并嵌入到自己的系统设备中。为实现其灵活性和兼容性的特点,已形成了系列化、模块化的单板计算机,此时人们开始不必从选择芯片开始来设计一台专用的嵌入式计算机了,只要选择各种不同的功能模块,就可以组建一台专用计算机系统。

20 世纪 80 年代可以说是各种总线层出不穷、群雄并起的时代。随着微电子工艺水平的提高,集成电路制造商开始把嵌入式应用所需要的微处理器、I/O 接口、A/D、D/A 转换器、串行接口以及 RAM、ROM 等都集成到一个超大规模集成电路(VLSI)中,制造出面向 I/O 设计的微控制器,即俗称的单片机。单片机成为嵌入式系统异军突起的一支新秀,而其后发展的 DSP 产品更快地提升了嵌入式系统的技术水平,迅速地渗入到消费电子、医

用电子、智能控制、通信电子、仪器仪表、交通运输等各种领域。此时，商业化的嵌入式操作系统开始得到蓬勃发展。

20 世纪 90 年代以来，在分布控制、柔性制造、数字通信和数字家电等巨大商业需求的推动下，嵌入式应用进一步得到了发展。在硬件方面，尤其是芯片方面的发展表现出三个特点。第一，面向实时信号处理算法的 DSP 向高速、高精度、低功耗方向发展。第二，32 位微控制器已成为市场主流，并向高速、智能化的方向发展。第三，嵌入式 CPU 的主频越来越高，同时在对多媒体的支持方面发展迅速，例如，Intel 基于 Xscale 的微架结构和富士通的 FR-V 等都是嵌入式 CPU 中支持多媒体的典范。在软件方面，嵌入式操作系统出现了进一步的分化，诞生了一些分别针对不同领域、不同应用的嵌入式操作系统家族。越来越多的嵌入式系统被连接到网络上。因而，在嵌入式系统中需要有网络协议栈的支持，把网络栈添加到一个仅用循环控制来实现的简单嵌入式系统所带来的复杂程度，足以唤起人们对一个嵌入式操作系统的渴望。

目前国内外已有几十种商业化操作系统可供选择，如 VxWorks、pSOS、Palm OS、Neculeus、Windows CE 和“女娲 Hopen”等。在我国，嵌入式操作系统可分为两大类型：一类是自主知识产权的操作系统，另一类是基于 Linux 的操作系统。自主知识产权的操作系统方面，国内有“女娲 Hopen”操作系统、桑夏 2000 操作系统和 DeltaOS 操作系统等。近年来，嵌入式 Linux 进展较快。在我国，以 Linux 为基础的嵌入式操作系统比较活跃，其中中软 Linux、红旗 Linux、东方 Linux 是业界的代表。

随着现代技术的发展，计算机工业逐渐向后 PC 时代发展，PC 工业慢慢地成为制造业的一部分。在后 PC 时代，嵌入式应用将成为计算机应用的主流，同时，嵌入式应用同无线应用的结合使嵌入式应用更具活力。应该指出，未来几年的嵌入式应用将能更灵活地与大型网络计算平台互动，形成新的应用模式，甚至构架出新的计算体系机构。因此，嵌入式应用和嵌入式软件将成为计算机工业中最具活力的部分。

1.3 嵌入式系统的体系结构

根据国际电气和电子工程师协会（IEEE）的定义，嵌入式系统是“控制、监视或者辅助设备、机器和车间运行的装置”（devices used to control, monitor, or assist the operation of equipment, machinery or plants）。一般而言，整个嵌入式系统的体系结构可以分成四个部分：嵌入式处理器、嵌入式外围设备、嵌入式操作系统和嵌入式应用软件。

1.3.1 嵌入式处理器

1. 嵌入式微处理器（EMPU）

嵌入式微处理器（Embedded Microprocessor Unit）的核心是通用计算机中的 CPU。在

实际应用中,为了大幅度地减小系统的体积和功耗,微处理器通常被装配在专门设计的电路板上,仅保留同嵌入式应用有关的功能。嵌入式微处理器虽然在功能上与标准微处理器基本上一样,但为了适应嵌入式应用的特殊要求和场合,在工作温度、抗电磁干扰,以及可靠性等方面一般都做了各种增强。与工业控制计算机相比,嵌入式微处理器具有体积小、重量轻、成本低、可靠性高的优点;但是由于在电路板上必须包含 ROM、RAM、总线接口、各种外设等器件,从而降低了系统的可靠性,并且技术保密性也较差。

2. 嵌入式微控制器 (MCU)

嵌入式微控制器 (Microcontroller Unit) 又称单片机。顾名思义,这就是将整个计算机系统集成到一块芯片中。嵌入式微控制器一般以某种微处理器内核为核心,芯片内部集成 ROM/EPROM、RAM、总线、总线逻辑、定时/计数器、看门狗定时器、I/O、串行口、脉宽调制输出、A/D、D/A、Flash、EEPROM 等各种必要的功能和外设。为适应不同的应用场合,一般某种系列的单片机都具有多种衍生产品,每种衍生产品的处理器内核都相同,不同的是存储器和外设的配置及封装。这样可以使单片机最大限度地同应用需求相匹配,使其功能齐全又不浪费,而且减少了功耗及产品成本。与嵌入式微处理器相比,微控制器的最大优点就是单片化,大大减小了体积,降低了功耗和成本,并提高了可靠性。微控制器是目前嵌入式系统工业的主流,微控制器的片上外设资源一般比较丰富,适合于控制,因此称之为微控制器。

3. 嵌入式 DSP 处理器

嵌入式数字信号处理器 (Embedded Digital Signal Processor, EDSP) 对系统结构和指令进行了特殊设计,使其适合于执行 DSP 算法,编译效率较高,指令执行速度快。DSP 算法正大量进入嵌入式领域。在数字滤波、FFT、频谱分析等方面,DSP 应用正从通用单片机中以普通指令实现 DSP 功能,过渡到采用嵌入式 DSP 处理器。由于智能化算法一般运算量较大,特别是向量运算、指针线性寻址等较多,而这些正是 DSP 处理器的长处所在,例如应用在带有智能逻辑的消费类产品,生物特征识别的终端,实时语音解压缩系统等方面。

4. 嵌入式片上系统

随着电子数据交换 (EDI) 的推广和 VLSI 设计的普及化,以及半导体工艺的迅速发展,在一个硅片上实现一个更为复杂的系统的时代已经来临,这就是嵌入式片上系统 (System On Chip, SOC)。各种通用处理器内核将作为 SOC 设计公司的标准库,与许多其他嵌入式系统外设一样,成为 VLSI 设计中一种标准的器件,它用标准的 VHDL 等语言进行描述,并存储在器件库中。用户只需定义出整个应用系统,仿真通过后就可以将设计图交给半导体工厂制作样品。这样除个别无法集成的器件外,整个嵌入式系统大部分均可集成到一块或几块芯片中,应用系统电路板将变得很简洁,这对于减小体积和功耗,以及提高可靠性都非常有利。

1.3.2 嵌入式外围设备

在嵌入式系统硬件系统中,除了中心控制部件(MCU、DSP、EMPU、SOC)以外,用于完成存储、通信、调试、显示等辅助功能的其他部件,事实上都可以算作嵌入式外围设备。目前常用的嵌入式外围设备按功能可以分为存储设备、通信设备和显示设备三类。

存储设备主要用于各类数据的存储,常用的有静态易失型存储器(RAM、SRAM)、动态存储器(DRAM)和非易失型存储器(ROM、EPROM、EEPROM、Flash)三种,其中Flash凭借其可擦写次数多、存储速度快、存储容量大、价格便宜等优点,在嵌入式领域内得到了广泛应用。

目前存在的绝大多数通信设备都可以直接在嵌入式系统中应用,包括RS-232接口(串行通信接口)、SPI(串行外围设备接口)、IrDA(红外线接口)、I²C(现场总线)、USB(通用串行总线接口)、Ethernet(以太网接口)等。

由于嵌入式应用场合的特殊性,通常使用的是阴极射线管(CRT)、液晶显示器(LCD)和触摸板(Touch Panel)等外围显示设备。

1.3.3 嵌入式应用软件

嵌入式应用软件是针对特定应用领域,基于某一固定的硬件平台,用来达到用户预期目标的计算机软件。由于用户任务可能有时间和精度上的要求,因此有些嵌入式应用软件需要特定嵌入式操作系统的支持。嵌入式应用软件和普通应用软件有一定的区别,它不仅要求其准确性、安全性和稳定性等方面能够满足实际应用的需要,而且还要尽可能地进行优化,以减少对系统资源的消耗,降低硬件成本。具体体现在如下几个方面:

1. 系统软件(OS)的高实时性

在多任务的嵌入式系统中,对重要性各不相同的任务进行合理调度是保证每个任务及时执行的关键,单纯通过提高处理器速度是不合理的。这种任务调度需要由优化编写的系统软件来完成,因此系统软件的高实时性是基本要求。

2. 软件要求固态化存储

为了提高执行速度和系统可靠性,嵌入式系统中的软件一般都固化在存储器芯片或单片机本身中,而不是存储于磁盘等载体中。

3. 软件代码高质量、高可靠性

尽管半导体技术的发展使处理器速度不断提高、片上存储器容量不断增加,但在大多数应用中,存储空间仍然是非常宝贵的,另外还存在实时性的要求。为此,对程序编写和编译工具的要求要高,其目的在于减少程序的二进制代码长度,并提高软件执行的速度。

4. 多任务操作系统是走向工业标准化道路的基础

由于嵌入式系统本身不具备自举开发能力,因此它不能像通用计算机一样,在具有完善的人机接口界面上增加一些开发的应用程序和环境进行对自身的开发,而必须具有一套开发工具和环境才能进行开发。

此外,通用计算机具有完善的操作系统和应用程序接口,应用程序的开发以及完成后的软件都可以在 OS 平台上运行,但通常它们不具有实时性。而嵌入式系统则不同,应用程序可以在没有操作系统的芯片上直接运行,但为了合理地调度多任务和利用系统资源,系统必须采用 RTOS 开发平台,这样才能保证程序执行的实时性和可靠性,并减少开发时间,保障软件质量。

1.3.4 嵌入式操作系统

为了使嵌入式系统的开发更加方便和快捷,需要有专门负责管理存储器分配、中断处理、任务调度等功能的软件模块,这就是嵌入式操作系统。嵌入式操作系统是用来支持嵌入式应用的系统软件,是嵌入式系统极为重要的组成部分,通常包括与硬件相关的底层驱动程序、系统内核、设备驱动接口、通信协议、图形用户界面(GUI)等。嵌入式操作系统具有通用操作系统的基本特点,例如:能够有效管理复杂的系统资源;对硬件进行抽象;提供库函数、驱动程序、开发工具集等。但与通用操作系统相比较,嵌入式操作系统在系统实时性、硬件依赖性、软件固化性以及应用专用性等方面,具有更加鲜明的特点。

从嵌入式操作系统能否满足实时性要求来分类,可把操作系统成分时操作系统和实时操作系统。分时操作系统按照相等的时间片轮流调度进程运行,由调度程序自动计算进程的优先级,并不由用户控制进程的优先级。这样的系统无法实时响应外部异步事件,分时系统主要应用于科学计算和一般实时性要求不高的场合。实时操作系统能够在限定的时间内执行所规定的功能,并能够在限定的时间内对外部的异步事件做出响应。在实时系统中,操作的正确性不仅依赖于逻辑设计的正确程度,而且与这些操作进行的时间有关。也就是说,实时系统对逻辑和时序的要求非常严格,如果逻辑和时序控制出现偏差将会产生严重后果。实时系统主要应用于过程控制、数据采集、通信、多媒体信息处理等对时间敏感的场所。

实时系统根据响应时间可以分为弱实时系统、一般实时系统和强实时系统三种。弱实时系统在设计时的宗旨是使各个任务运行得越快越好,但没有严格限定某一任务必须在多长时间内完成;弱实时系统更多关注的是程序运行结果的正确与否,以及系统安全性能等其他方面,对任务执行时间的要求相对来讲较为宽松,一般响应时间可以是数十秒或者更长。一般实时系统是弱实时系统和强实时系统的一种折衷,它的响应时间可以在秒的数量级上,广泛应用于消费电子设备中。强实时系统则要求各个任务不仅要保证执行过程和结果的正确性,同时还要保证在限定的时间内完成任务,响应时间通常要求在毫秒甚至微秒的数量级上,这对涉及到医疗、安全、军事的软硬件系统来说是至关重要的。

1.4 典型的嵌入式操作系统

嵌入式操作系统是嵌入式系统的核心，嵌入式操作系统根据应用场合可分为两大类：一类是面向消费电子产品的非实时系统，这类设备包括个人数字助理（PDA）、移动电话、机顶盒（STB）等；另一类则是面向控制、通信、医疗等领域的实时操作系统，如 WindRiver 公司的 VxWorks、QNX 系统软件公司的 QNX 等。

目前，市场上的嵌入式操作系统种类繁多，下面简要介绍常用的嵌入式操作系统。

1. VxWorks

VxWorks 操作系统是美国 WindRiver 公司于 1983 年设计开发的一种嵌入式实时操作系统（RTOS），因其良好的持续发展能力、高性能的内核、友好的开发环境、卓越的可靠性和实时性，被广泛应用于通信、航空、航天等实时性要求极高的领域中。

VxWorks 由一个体积很小的内核以及一些可以根据需要进行定制的系统模块组成。微内核支持实时系统的一系列特征，包括多任务、中断支持、任务抢占式调度和轮转调度。微内核设计使 VxWorks 缩减了系统开销并加速了对外部事件的反应。VxWorks 内核最小为 8KB，即便加上其他的必要模块，所占用的空间仍然很小且不失其实时性和多任务的系统特征。

VxWorks 实时操作系统由 400 多个相对独立的、短小精炼的目标模块组成，用户可根据需要选择适当模块来裁剪和配置系统，这可有效地保证系统的安全性和可靠性。系统的链接器可按应用的需要自动链接一些目标模块。这样，通过目标模块之间的按需组合，可得到许多满足功能需求的应用。

此外，VxWorks 支持广泛的工业标准，如 POSIX 1003.1b 实时扩展、ANSI C（浮点支持）和 TCP/IP 网络协议等。VxWorks 提供了强大的网络功能，能与其他许多主机系统进行通信。这种广泛的协议支持在主机和目标机之间提供了无缝的工作环境，任务可通过网络向其他系统的主机存取文件，即远程文件存取，也支持远程过程调用。通过以太网，采用 TCP/IP 和 UDP/IP 协议在不同主机之间传送数据。这些标准促进多种不同产品间的互用性，提升了可移植性。由于其高度的灵活性，用户可以很容易地对该操作系统重新进行定制或作适当开发，以满足自己的实际应用。

Tornado 是嵌入式实时领域里最新一代的开发调试环境，提供了高效明晰的图形化的实时应用开发平台，它包括一套完整的面向嵌入式系统的开发和调测工具。Tornado 环境采用主机—目标机交叉开发模型，应用程序在主机上 Windows 环境下编译链接生成可执行文件，下载到目标机，通过主机上的目标服务器与目标机上的目标代理程序的通信完成对应用程序的调测、分析。这些工具包括 C 和 C++ 远程级调试器、目标和工具管理、系统目标跟踪、内存使用分析和自动配置，所有工具能很方便地同时运行，很容易增加扩展和交互式开发。

板级支持包 BSP (Board Support Package) 作为 VxWorks 系统的主要组成部分, 对各种板子的硬件功能提供了统一的软件接口, 它包括硬件初始化、中断的产生和处理、硬件时钟和计时器管理、局域和总线内存地址映射、内存分配等。每个板级支持包包括一个 ROM 启动 (Boot ROM) 或其他启动机制。

总之, VxWorks 是可以根据用户需求进行组合的实时操作系统, 其开放式结构和对工业标准的支持使开发者只需做最少的工作即可设计出有效的适合于不同用户要求的实时操作系统。

本书将在下面几章中分别详细介绍 VxWorks 操作系统、Tornado 开发环境、BSP 以及网络编程的相关内容, 在此不再赘述。

2. Windows CE

Windows CE (以下简称 CE) 是美国微软公司专门为各种移动和便携电子设备、个人信息产品、消费类电子产品、嵌入式应用系统等非台式或笔记本电脑领域设计的一种 32 位高性能操作系统。它具有一个简捷、高效的完全抢先式多任务操作核心, 支持强大的通信和图形显示功能, 能够适应广泛的系统需求。

CE 操作系统由一些独立的模块组成, 每一个模块提供特定的系统功能, 大的模块又可以分成几个组件。这种组件式结构能使 CE 系统变得非常紧凑, 仅需使用很少的硬件资源就可运行。最主要的系统模块有四个, 即内核、持久性存储模块、绘图窗口事件子系统和通信模块。内核是整个操作系统的核心部分, 它负责最基本的操作系统功能, 包括内存管理、进程管理和必需的文件管理。持久性存储模块通过对 CE 系统中对象存储功能的支持, 为用户和应用程序提供数据存储服务。绘图、窗口、事件子系统是用户、应用程序和操作系统之间的图形化操作界面, 负责提供系统与用户之间的人机交互接口。通信模块为运行 CE 系统的设备提供对多种通信硬件和数据传输协议的支持, 包括串/并行数据端口、红外数据端口、电话应用程序接口以及网络通信协议。

这种基于模块和组件的结构使 CE 系统具有高度的可伸缩性, 它允许用户在设计自己的操作系统时自由选择需要的模块或组件。这样, 系统开发者就能够方便地根据具体的硬件要求定制出合适的操作系统, 使其中仅包含设备运行所必需的功能模块, 从而不必为支持其他不需要的硬件或功能付出额外代价。通过采用这种方式, CE 系统在支持目前市场上绝大多数硬件设备及接口标准的同时又保证了最大的灵活性和系统效率, 因此能够适应广泛的用户系统的要求。

CE 操作系统不需要任何特定的硬件结构, 实际的硬件系统完全由用户根据需要自由设计。从理论上说, 一台以 CE 系统为基础的设备必须有一个处理器, 有内存及内部时钟以便处理进度, 以及周边设备。为了进一步减少对内存资源的需求, CE 系统在采用模块化设计的同时还大量使用了动态链接库。许多功能函数都包含在动态链接库中, 仅当应用程序需要时才调入内存, 平时则以文件形式保存在存储器内。作为一个小而灵活的操作系统, 以 CE 为基础的系统所需内存大小完全依赖于系统选用的模块和组件。

此外, CE 兼容于微软公司的视窗 (Windows) PC 电脑操作系统, 支持超过 1000 个常用的 32 位视窗应用程序接口函数, 支持高分辨率真彩色显示, 为应用软件提供了强大的运