

21世纪高等学校嵌入式系统专业规划教材

赖晓晨 原旭 孙宁 编著

嵌入式 系统程序设计



清华大学出版社

21世纪高等学校嵌入式系统专业规划教材

赖晓晨 原旭 孙宁 编著

嵌入式 系统程序设计



清华大学出版社
北京

内 容 简 介

本书突破了以各种低端单片机芯片为基础讲解 C 语言程序设计的传统模式,以主流的 32 位 ARM 嵌入式处理器/嵌入式 Linux 操作系统为平台,全面介绍嵌入式系统的 C 语言程序设计方法,内容涉及 C 语言适用于嵌入式系统的语法特性、Linux 系统的开发环境和工具链、交叉编译、uClinux 和 ARM Linux 的移植等基础内容,以及嵌入式系统程序框架、BootLoader、BusyBox、Linux 系统编程、嵌入式数据库、模块设计、驱动程序设计、Framebuffer 技术、图形界面移植等较为深入的内容。

本书的嵌入式编程环境为:宿主机采用稳定的 RedHat Linux 9 操作系统和 Cygwin 模拟环境,使用方便;目标机采用深圳英蓓特信息技术有限公司的 Embest EduKit-III 教学实验平台。考虑到读者的实际情况,作者力争对涉及到的硬件平台部分做详细的介绍,一些平台无关的代码,尽量在模拟器上执行,使没有条件购买硬件设备的读者也能得到全面的认识。

阅读本书的读者应该具有 C 语言的基础,对 ARM 处理器有初步了解。本书适合嵌入式专业的高年级本科学学生作为教材使用,也适用于希望转向嵌入式系统设计的 IT 从业人员,尤其适合学习了嵌入式硬件基础知识,但是还没有形成嵌入式系统设计完整概念的人员,作为学习主流嵌入式系统程序设计的指导教材使用。

本书电子资源内容包括完整的教学课件、书中所有的例程源代码、书中使用到的 GNU 编程工具等。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

嵌入式系统程序设计/赖晓晨等编著. —北京:清华大学出版社,2009.11

21 世纪高等学校嵌入式系统专业规划教材

ISBN 978-7-302-20253-0

I. 嵌… II. 赖… III. 微型计算机—系统设计—教材 IV. TP360.21

中国版本图书馆 CIP 数据核字(2009)第 082251 号

责任编辑:梁 颖 李玮琪

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

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

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京国马印刷厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:22.25 字 数:548 千字

版 次:2009 年 11 月第 1 版 印 次:2009 年 11 月第 1 次印刷

印 数:1~4000

定 价:32.00 元

本书如存在文字不清、漏印、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)62770177 转 3103 产品编号:029737-01

前 言

当前,嵌入式计算机已经形成一个规模庞大的产业,嵌入式产品遍及天空、陆地和海洋,市场对各类嵌入式产品提出了巨大的需求,一方面促进了嵌入式技术的进一步发展,另一方面也导致需求和供给之间的矛盾,体现为嵌入式人才的巨大缺口,这是呈现在高校计算机教学领域的严峻课题。

本书作者自 2005 年以来,一直在大连理工大学软件学院讲授“嵌入式系统程序设计”课程,教学内容为介绍基于 ARM 处理器/嵌入式 Linux 操作系统的 C 语言程序设计方法。开课伊始,遇到了教材缺乏的困难。当时图书市场中介绍嵌入式系统的书籍较多,大体可以分为 4 类:嵌入式系统概述性质的书籍;从硬件角度讲解 ARM 处理器体系结构及汇编语言程序设计的书籍;各种介绍其他体系结构嵌入式处理器的书籍,例如 8051、龙珠处理器等;介绍嵌入式操作系统的书籍,例如 uC/OS II、VxWorks 等。仅有的几种嵌入式程序设计类书籍,内容全部基于 8051 单片机,描述 C51 程序设计。在图书市场中,没有一本介绍基于 32 位 ARM 处理器/嵌入式 Linux 操作系统的 C 语言程序设计的书籍。不但如此,当时国内各个高校,也几乎没有开设“嵌入式系统程序设计”类似课程的先例。在这种情况下,笔者萌发了编写一本教材的想法。在各位同行的帮助下,一边学习,一边总结摸索,通过搜集大量资料,结合自身在实验设备上的各种尝试,逐步设计出了课程的教学大纲和教学课件,然后在授课过程中根据学生的反馈意见不断完善授课内容,经过几年的磨砺,最终形成了一套比较稳定的教学资源。2008 年春,清华大学出版社来我院开办书展,经过和出版社梁颖编辑协商,决定把几年来的教学资料加以整理,编写一本专门教材,这就是本书诞生的经过。

本书正文部分内容共 7 章:第 1 章介绍一些背景知识,力争使读者建立起嵌入式系统程序设计的完整概念;第 2 章介绍一些前导知识,包括 ARM 体系结构、Linux 操作系统、本书相关硬件平台等内容,有基础的读者可以跳过此章;第 3 章介绍 C 语言在嵌入式系统中的应用;第 4 章介绍 GNU 开发工具链,这部分知识适合于本地开发环境和交叉开发环境;第 5 章介绍基于 ARM7 处理器的 uClinux 系统的建立和程序设计步骤,包括驱动程序设计等;第 6 章介绍基于 ARM9 处理器的 ARM Linux 系统的建立以及图形用户界面的移植过程;第 7 章以一个运行于 ARM9/Linux 平台完整实例的介绍作为本书正文部分的结尾。本书还包含 5 个附录,介绍具体硬件平台相关的操作步骤、知识点的背景常识等内容。

本书的大部分内容来自于作者的教学实践,另有部分内容参考了其他相关书籍以及网络上的资料,作者已尽量在参考文献中列出来源。由于本书编写过程漫长,且初期资料整理缺乏条理,因此有部分资料,现已无法找到准确的出处,在此向各位作者致歉,并请相关作者和本联系。此外,本书的 4.7、6.6 小节部分内容分别来自于徐海兵和于强(大漠孤狼)两位网友,经他们允许编入本书,感谢这两位朋友及其他更多未曾谋面的老师和同行!

本书内容涉及的嵌入式平台为深圳英蓓特信息技术有限公司设计的 Embest EduKit-III 型教学实验平台。为了保持正文叙述的流畅性,在编写过程中,作者刻意把一些背景和细节内容安排到附录中,供没有基础的读者查询。书中的典型概念,第一次出现时都有对应的

英文缩写以及英文全称,便于读者与英文原版教材接轨;书中的所有代码,都经调试验证无误;所有操作步骤,都有详细解释;重要步骤有截图供参考;如果读者采用相同的开发环境,可以顺利完成书中介绍的每一项操作。

本书的编写,历时将近3年,在这个过程中,得到了很多人的帮助,在此向大家致以衷心的感谢!首先感谢我的爷爷奶奶和父母家人,是你们教导我做一个正直的人,并引导我走上今天的人生道路;其次感谢大连理工大学电信学院的王孝良、梁茵、王伟、刘全利老师,是你们把我带入了计算机知识的海洋;感谢清华大学出版社,让本书的面世成为可能;特别感谢深圳市英蓓特信息技术有限公司的黄日新、周麒、尹锐、梁华骥工程师为我提供的无私帮助;感谢大连理工大学软件学院的惠晓丽、郭禾、周宽久、吴国伟、张政凯、惠煌等老师给我提供了一个宽松的工作环境;感谢李守维、杨洋、陈铁仁、施元超、陈超卓、李丽坤等朋友在本书内容和校对方面的贡献;感谢陈洁为本书编写付出的种种努力;最后,也是最重要的,感谢我曾经的学生们和将来的学生们,是你们带给了我工作的乐趣和动力!

由于作者经验有限,加之时间仓促,书中不可避免会有很多不足之处,恳请读者不吝批评指正。所有关于本书的意见,请发送电子邮件到 armprogram@gmail.com 信箱,希望在和读者交流的过程中能有所裨益。

编 者

2009年3月于大连

目 录

第 1 章 绪论	1
1.1 嵌入式系统概述	1
1.1.1 计算机的发展和分类.....	1
1.1.2 嵌入式系统概述.....	1
1.1.3 嵌入式系统的组成.....	2
1.1.4 嵌入式系统的发展.....	5
1.2 嵌入式系统开发流程	6
1.3 嵌入式系统程序设计	8
1.3.1 嵌入式系统开发语言.....	8
1.3.2 嵌入式系统的开发工具	10
1.4 本书内容简介.....	12
第 2 章 预备知识	14
2.1 ARM 体系结构	14
2.1.1 ARM 体系结构简介	14
2.1.2 ARM 指令集	21
2.1.3 S3C44B0X 嵌入式处理器	28
2.1.4 S3C2410X 嵌入式处理器	30
2.2 Linux 操作系统	31
2.2.1 Linux 操作系统的历史	31
2.2.2 Linux 操作系统的特点	33
2.2.3 Linux 版本	34
2.2.4 Linux 的安装	37
2.3 Linux 操作系统基础操作	37
2.3.1 shell	37
2.3.2 常用 shell 命令	39
2.4 Embest EduKit-Ⅲ型教学实验平台简介.....	42
2.4.1 Embest EduKit-Ⅲ型教学实验平台	43
2.4.2 JTAG 仿真器	44
2.4.3 Flash 编程器	44
2.4.4 Embest 集成开发环境.....	45
2.4.5 RealView MDK	46
第 3 章 嵌入式 C 语言基础	48
3.1 嵌入式 C 语言概述	48

3.1.1	C 语言的历史和特点	48
3.1.2	预处理指令	50
3.1.3	位运算	58
3.1.4	函数指针	59
3.1.5	C 程序的移植	61
3.2	C 语言与 C++ 语言的区别	64
3.2.1	变量定义位置	64
3.2.2	结构体变量	64
3.2.3	数据类型	64
3.2.4	输入输出	65
3.2.5	动态内存分配	67
3.2.6	其他区别	68
3.3	GNU C 扩展	69
3.4	错误处理机制	73
3.4.1	C 语言的错误处理机制	73
3.4.2	系统日志文件	80
3.5	嵌入式 C 程序架构	83
3.5.1	嵌入式 C 程序	83
3.5.2	人机界面	85
3.5.3	指针的使用方法	87
3.6	ARM C 程序设计优化	88
3.6.1	编译器的特性	88
3.6.2	数据类型对程序效率的影响	89
3.6.3	循环方式对程序效率的影响	90
3.6.4	指针对程序效率的影响	91
3.6.5	边界对齐对程序效率的影响	91
3.7	硬件编程	92
3.8*	C 语言编程规范	95
3.8.1	排版	96
3.8.2	注释	98
3.8.3	命名规则	102
3.8.4	可读性	103
3.8.5	变量和结构	104
3.8.6	函数	104
3.8.7	程序效率	104
第 4 章	嵌入式 Linux 程序设计基础	106
4.1	交叉开发环境	106
4.2	Linux 开发工具链	107

4.2.1	Linux 开发工具链简介	107
4.2.2	GNU 交叉开发环境的建立	109
4.3	编辑器	111
4.4	编译器	114
4.4.1	gcc 简介	114
4.4.2	gcc 的基本用法	115
4.5	链接器	118
4.6	调试器	121
4.7	自动化编译配置文件	129
4.7.1	自动化编译配置文件简介	129
4.7.2	Makefile 规则	132
4.7.3	Makefile 的变量	141
4.7.4	Makefile 的执行	150
4.7.5	make 内嵌函数	154
4.7.6	make 的常见错误信息	163
4.8*	源码包配置工具	164
4.9	版本控制工具	167
第 5 章	嵌入式 uClinux 程序设计	172
5.1	uClinux 操作系统	172
5.1.1	uClinux 操作系统简介	172
5.1.2	uClinux 操作系统架构	175
5.1.3	BusyBox	176
5.2	BootLoader	180
5.2.1	BootLoader 概述	180
5.2.2	BootLoader 框架	182
5.2.3	BootLoader 代码分析	184
5.3	文件系统	186
5.4	uClinux 操作系统构建	188
5.5	uClinux 应用程序设计	190
5.5.1	交叉编译	191
5.5.2	执行程序	192
5.6	嵌入式 C 语言进阶	195
5.6.1	文件操作	195
5.6.2	目录操作	200
5.6.3	内存操作	203
5.6.4	进程控制	205
5.6.5	进程间通信	212
5.7	嵌入式数据库	215

5.7.1	SQLite	216
5.7.2	Berkeley DB	216
5.7.3	SQLite 移植	217
5.8	Linux 内核模块	218
5.8.1	Linux 内核模块概述	218
5.8.2	Linux 内核模块实例	219
5.8.3	Linux 内核模块实现机制	223
5.9	Linux 驱动程序工作原理	225
5.9.1	设备驱动程序概述	225
5.9.2	设备驱动程序接口	228
5.10	Linux 驱动程序实例	230
5.10.1	驱动实例(一)	230
5.10.2	驱动实例(二)	233
5.10.3	驱动实例(三)	235
第 6 章	ARM Linux 程序设计	246
6.1	Linux 库	246
6.1.1	Linux 库概述	246
6.1.2	库操作工具	247
6.1.3	静态库	248
6.1.4	共享库	250
6.1.5	动态链接库	251
6.2	ARM Linux 操作系统	253
6.2.1	ARM Linux 操作系统简介	253
6.2.2	启动引导程序 vivi	254
6.2.3	CRAMFS 文件系统	256
6.2.4	构建 ARM Linux 系统	258
6.3	ARM Linux 驱动程序设计	260
6.4	Framebuffer 技术	267
6.5	基于 Framebuffer 的图形界面	271
6.6	MiniGUI	288
6.6.1	MiniGUI 简介	288
6.6.2	MiniGUI 模拟环境	290
6.6.3	MiniGUI 移植	296
6.7	SkyEye	307
第 7 章	嵌入式系统程序设计实例	310
7.1	应用背景	310
7.2	系统架构	311

7.3 详细设计	314
7.3.1 数据结构描述.....	314
7.3.2 电梯运行算法设计.....	316
7.3.3 输入输出部分设计.....	319
7.4 系统函数说明	320
7.5 系统演示	321
附录 A 虚拟机配置	322
A.1 操作系统安装准备	322
A.2 配置文件共享目录	324
A.3 配置网络	325
附录 B 数字化温度传感器	327
附录 C Linux 模拟环境 Cygwin	331
附录 D uCLinux 内核选项	333
附录 E BMP 文件格式	336
参考文献.....	339

第1章 绪 论

随着计算机技术的日益发展和人们需求的不断增加,嵌入式系统渗入到了社会生活的各个方面,几乎每一个人每天都在使用嵌入式计算机,计算机和信息技术的发展正在逐渐步入以嵌入式为核心的普适计算时代。本章主要介绍嵌入式系统概念(Embedded System)的内涵和外延、嵌入式系统的开发流程、嵌入式系统程序设计的相关知识,希望通过本章的阅读,读者可以大致了解嵌入式系统的软硬件体系结构,建立起宏观的概念。

1.1 嵌入式系统概述

1.1.1 计算机的发展和分类

计算机作为20世纪最伟大的发明,给人们的生活带来了翻天覆地的变化。传统上,按照结构规模的不同,把计算机分为大型机、中型机、小型机和微型计算机,并以此来组织学科和产业分工,这种分类沿袭了约40年。近十几年来随着计算机技术的迅速发展,各种社会需求的不断涌现,使得实际情况产生了根本性的变化:例如,个人计算机占据了当前全球计算机工业90%的市场份额,个人计算机的处理速度,早已超过了几十年前大型计算机的运行速度;再如,十几年前的笔记本计算机曾被视为“阳春白雪”,价格十分昂贵,如今性能提升几十倍、价格只有当初十分之一的笔记本早已走入千家万户。此外,各种包含微处理器的消费类电子产品如mp4、个人数字助理(Personal Digital Assistant, PDA)、手机、数码相机等已成为人们生活中不可或缺的一部分。由此可见,计算机技术的发展已经使按照规模划分层次结构的方法落后于实际情况,计算机技术和产品在不断地对其他行业进行广泛地渗透,以应用为中心的分类方法变得更为切合实际,即按计算机的嵌入式应用和非嵌入式应用将其分为嵌入式计算机和通用计算机。

通用计算机具有计算机的标准形态,通过安装不同的应用软件,以相似面目出现并应用在社会各个方面,其典型产品为个人电脑(Personal Computer, PC);而嵌入式计算机则是以嵌入式系统的形式隐藏在各种装置、产品和系统内部,不显式地具有计算机的外观,没有通用计算机的标准输入和标准输出部分,功能比较单一,一般不具备通用计算机处理各种事务的综合能力。

1.1.2 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、软件硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。任何一台嵌入式计算机都强化了一个主要的应用目的,而弱化了通用计算机具有的其他应用功能,并因此而裁减了软硬件。例如,嵌入到导弹中的嵌入式计算机计算功能强大,强化了跟踪制导的功能,而

不具备其他的诸如多媒体之类的功能,所以也不必配备 PC 的标准外设,例如音箱、大屏幕显示器,同样也不需要一个庞大的 Windows 操作系统。再如,数码相机中的嵌入式计算机,强化了图像处理能力,在提高画面质量等方面投入了大部分计算能力,但是不具备 PC 的标准键盘,输入能力远远弱于 PC,数码相机使用的软件也紧紧围绕其图像处理的主要应用目的,裁减掉了一切不必要部分。由于嵌入式计算机要求体积小,使用时一般没有外接电源,而且可以应用于关键场所,所以在体积、功耗、可靠性等方面都有严格的要求。

在现实世界中,嵌入式计算机的数量远远超过了通用计算机,这是因为每一台通用计算机实际都包含了 5~10 个嵌入式微处理器,例如键盘、显示卡、显示器、鼠标、软驱、硬盘、网卡、Modem、声卡、打印机、扫描仪、数字相机、USB 集线器等均是由嵌入式处理器控制的。制造工业、过程控制、通信、仪器、仪表、汽车、船舶、航空、航天、军事装备、消费类产品等方面均是嵌入式计算机的应用领域。

嵌入式系统是一个综合的概念,它将计算机技术、半导体技术和电子技术和各个背景行业的具体应用结合起来,这就决定了它是一个技术密集、资金密集、高度分散、不断创新的知识集成系统。

嵌入式系统行业每年创造的工业年产值已超过了几万亿美元。1997 年美国嵌入式系统大会(Embedded System Conference,ESC)的报告指出,未来 5 年仅基于嵌入式计算机系统的全数字电视产品,就将在美国产生一个每年 1500 亿美元的新市场。美国汽车大王福特公司的高级经理也曾宣称“福特出售的‘计算能力’已超过了 IBM”,由此可以想象嵌入式计算机工业的规模和广度。2004 年之后,中国嵌入式系统市场步入快速增长时期,嵌入式系统的发展为几乎所有的电子设备注入了新的活力,由于迅速发展的 Internet 和非常廉价的微处理器的出现,嵌入式系统在日常生活中变得无处不在。2007 年,由 ARM(Advanced RISC Machines)公司的合作伙伴设计、制造的基于 ARM 处理器的芯片出货量达到了 29 亿;2008 年,这一数字突破了 100 亿,国内仅嵌入式微处理市场产值即逼近 2500 亿元人民币,中国政府已经开始高度重视嵌入式相关产业的发展,科技部将嵌入式软件列入国家重点发展专项课题,并正在积极打造嵌入式产业链。近年来,随着嵌入式行业的发展,人才与需求的矛盾日益突出,中国已经有超过 300 家大学开设了 ARM 课程;据权威部门统计,国内嵌入式人才缺口达到了每年 80 万人左右。

通用计算机的很多领域都是技术垄断的,例如大部分计算机的 CPU 都出自 Intel、AMD 等少数几个公司,在个人计算机的操作系统领域中,Windows 占据了很大市场份额。但是嵌入式计算机由于涉及方面非常广泛,没有哪个公司能够取得绝对垄断地位,因此嵌入式系统工业充满了竞争、机遇与创新,留给各个公司非常广阔的创新余地。此外,社会的经济生活的不断向前发展,也带动了嵌入式系统软件、硬件、开发工具、应用软件的同步发展,这也是推动嵌入式系统发展的原动力。

1.1.3 嵌入式系统的组成

嵌入式系统一般由嵌入式处理器、外围硬件设备、嵌入式操作系统以及用户应用程序 4 个部分组成。系统硬件部分包括嵌入式处理器、存储器、外围接口和其他硬件功能模块,软件部分包括嵌入式操作系统和用户应用程序,设计人员需要把操作系统与应用软件两者结合起来,操作系统负责控制与硬件的交互,应用程序负责实现用户的需求。

1. 嵌入式处理器

嵌入式处理器是嵌入式系统的核心部件,据不完全统计,目前全世界嵌入式处理器已经超过1000多种,流行的体系结构有30多个系列,8051体系结构在其中占据了很大比重,生产8051单片机的半导体厂家有20多个,共350多种衍生产品,仅飞利浦公司就有近100种。嵌入式处理器的寻址空间小于通用计算机,一般从64KB到16MB,处理速度从0.1 MIPS (Million Instructions Per Second)到2000 MIPS,常用封装从8个引脚到144个引脚。嵌入式处理器可以分为4个大类:嵌入式微处理器、嵌入式微控制器、数字信号处理器以及片上系统。

(1) 嵌入式微处理器(Embedded MicroProcessor Unit, EMPU)。

嵌入式微处理器是在通用计算机中央处理器的基础上设计而来的,它将微处理器安装到专门设计的电路板上,只保留和嵌入式应用有关的主板功能,大幅减小了系统的体积和功耗。为了满足嵌入式应用的特殊要求,嵌入式微处理器在工作温度、抗电磁干扰、可靠性等方面都做了增强。

嵌入式微处理器目前主要有 Am186/88、386EX、SC-400、Power PC、68000、MIPS 系列等。

(2) 嵌入式微控制器(MicroController Unit, MCU)。

嵌入式微控制器又称单片机,它以某种微处理器内核为核心,将计算机系统的各个部分集成在一块芯片中,包括 ROM/EPROM、RAM、总线、总线逻辑、定时/计数器、看门狗、I/O、串行口、脉宽调制输出(Pulse Width Modulation, PWM)、模数转换(A/D)、数模转换(D/A)、Flash、EEPROM等。各公司的微控制器一般都形成一个系列,同一系列内部各微控制器采用相同的内核,差别在于速度不同、内部集成的外设种类、特性不同,因而性能、功耗、价格不同,但是汇编语言基本相同,用户可以根据自己的实际需求选用适合的产品。

嵌入式微控制器的品种和数量最多,比较有代表性的通用系列包括 8051、P51XA、MCS-251、MCS-96/196/296、C166/167、MC68HC05/11/12/16、68300等。目前 MCU 占嵌入式系统约70%的市场份额。

(3) 数字信号处理器(Digital Signal Processor, DSP)。

DSP 处理器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,指令执行速度也较高。在数字滤波、FFT、谱分析等方面,DSP 算法正在大量进入嵌入式领域。嵌入式 DSP 处理器有两个发展来源,一个是 DSP 处理器通过增加外设成为嵌入式 DSP 处理器,另一个来源是普通嵌入式处理器在片内集成 DSP 协处理器,例如 Intel 的 MCS-296 和 Infineon(Siemens)的 TriCore。随着各种带有智能逻辑的消费类产品的不断发展,嵌入式 DSP 处理器的应用也愈加广泛。

在嵌入式 DSP 处理器中,比较有代表性的产品是德州仪器(Texas Instruments, TI)设计的 TMS320 系列和摩托罗拉公司设计的 DSP56000 系列。TMS320 系列处理器包括用于控制的 C2000 系列,移动通信的 C5000 系列,以及性能更高的 C6000 和 C8000 系列。DSP56000 目前已经发展成为 DSP56000、DSP56100、DSP56200 和 DSP56300 等几个不同系列的处理器。

(4) 嵌入式片上系统(System on Chip, SoC)。

随着超大规模集成电路(Very Large Scale Integration, VLSI)技术的发展,在一个芯片上设计一个复杂的系统已经成为可能,这就是片上系统技术。它是采用硬件描述语言来设

计各种处理器内核以及各种外设,把设计好的单元存储在器件库当中,用户只需根据系统要求选用这些器件,仿真通过后就可以将设计图交给半导体工厂制作样品。采用这种方式,可以把系统的大部分功能集成到一个芯片中,大大提高了系统可靠性,降低了系统功率,减少了开发时间,使系统设计变得更加简捷。

SoC 可以分为通用和专用两类。通用系列包括英飞凌(Infineon)的 TriCore,摩托罗拉的M-Core,某些 ARM 系列器件。专用 SoC 应用于某个特定领域中,一个代表性产品是 TI 公司出品的嵌入式片上系统 CC2430,这是一块真正的片上系统芯片,它以 8051 单片机为核心,包含高性能 2.4GHz 直接序列扩频射频收发器(DSSS),并集成了 zigbee 协议栈,向用户提供应用编程接口,可以直接进行无线网络应用的开发。

2. 外围硬件设备

视具体的应用目的不同,嵌入式系统具体的外围硬件设备各不相同,大体上可以包括以下几个部分。

(1) 电源部分:某些嵌入式设备一旦安装完毕,位置就固定下来,例如安装在工厂中的一些监测仪表。这部分设备的电源可以由交流电经过降压、整流、滤波、稳压来得到,相应的电源部分电路体积庞大。另外一些嵌入式设备的使用位置经常变换,例如手机、PDA、电子词典等,因此只能采用电池供电,相应的电路相对简单,体积较小。

(2) 输入部分:这部分电路属于人机接口部分,用来把外界信息转换成为计算机能够识别的信息格式,传送到计算机内部。例如:键盘、鼠标、触摸屏、拨码开关等。

(3) 输出部分:这部分电路同样属于人机接口,负责把经过计算得到的结果用人类能够接受的方式表现出来,例如发光二极管、LED(Light Emitting Diode)显示器、LCD 显示器、蜂鸣器等。

(4) 接口电路:这部分电路负责提供与其他电子设备的接口,例如 USB 接口、PS/2 接口、串口、IDE 接口、红外接口、1394 接口、CF 卡接口、网络接口、CAN 总线接口、RS422 接口、RS485 接口等,每一种接口一般都对应一个专用的控制芯片,例如串口一般是由 8250 芯片来控制。

(5) 存储部分:这部分电路用来进行信息的存储,包括永久性存储器以及易失性存储器。前者的特点是断电后信息不会丢失,例如 ROM、EPROM、EEPROM、Flash 等存储器,后者的特点是一旦断电,保存在其中的所有信息都会丢失,例如 RAM。

(6) 其他硬件逻辑电路:例如 AD 转换电路、电机驱动电路、时钟日期生成电路等。

3. 嵌入式操作系统

嵌入式操作系统属于系统软件,一般固化在硬件存储器当中。常见的嵌入式操作系统有 Linux、uClinux(实际应为 μ CLinux,为了书写印刷方便,用“u”替换“ μ ”)、Windows CE、PalmOS、Symbian、eCos、uC/OS II、VxWorks、pSOS、Nucleus、ThreadX、Rtems、QNX、INTEGRITY、OSE、C Executive。

嵌入式系统中一般使用实时操作系统(Real Time Operation System,RTOS),通俗地说,它是指针对外界激励,能够在一段确定的时间内及时给出响应的操作系统。这是因为嵌入式设备经常应用在生产生活中的关键场合,要求在最坏情况下,仍然能保持一定的反应速度。在常用的嵌入式操作系统中,除了 Windows CE 之外,大多数实时性都很强。从事嵌入式的研发人员至少须掌握一个嵌入式操作系统,这在嵌入式的所有技术中最为关键。

目前最重要的嵌入式操作系统主要有以下几个。

(1) VxWorks 操作系统: VxWorks 是一个非常优秀的嵌入式操作系统,它的实时性强,内核可裁剪得很小,可靠性较高。在北美,VxWorks 占据了嵌入式系统的多半市场,特别是在通信设备等实时性要求较高的系统中,VxWorks 占据了绝大部分市场份额。但是 VxWorks 及相关开发工具的价格很高,小公司往往难以承受。

(2) 嵌入式 Linux 操作系统: Linux 是一个成功的服务器操作系统,此外由于其开源、免费、可移植性好的特点,在嵌入式系统中也有广泛应用。嵌入式 Linux 可以分为两类,一类经过修改,专门在没有内存管理单元(Memory Management Unit,MMU)的硬件上运行,例如 uCLinux;另一类具有完全的 Linux 特性,运行于设有内存管理单元的硬件平台,例如 ARM Linux。本书的第 5 章和第 6 章会分别介绍这两种系统。学习 Linux 要注重 3 个方面: Linux 移植、Linux 驱动设计、Linux 内核裁减和优化。

(3) uC/OS II: 目前很多大学都以 uC/OS II 为教学使用的实时操作系统,这主要是由于 uC/OS II 较简单,源码公开,非常适合入门者学习实时操作系统原理。uC/OS II 的缺点是功能有限,实际使用较少。

(4) Windows CE: Windows CE 是微软公司推出的嵌入式操作系统,在短短几年时间内已经占据了很大的市场份额,特别是在 PDA、手机、显示仪表等界面要求较高或者要求快速开发的场合。Windows CE 的开发平台主要为 Windows CE Platform Builder,有时也用 EVC(Embedded Visual C++)环境开发一些较上层的应用,由于 Windows CE 开发环境秉承了微软一贯的软件风格,为大家所熟知,所以学习 Windows 程序设计课程相对较为简单,这也是 Windows CE 容易被人们接受的原因,开发环境方便快捷,兼之微软的强大技术支持,使得 Windows CE 开发难度远低于嵌入式 Linux。

本书描述的嵌入式程序开发主要基于 Linux/uCLinux。

4. 嵌入式应用程序

嵌入式处理器的应用软件是实现嵌入式系统功能的关键,对嵌入式处理器系统软件和应用软件的要求也和通用计算机有所不同。

(1) 固态化存储: 嵌入式系统为了提高执行速度和系统可靠性,一般不采用硬盘等存储设备,而是采用非易失性存储器,例如 EPROM 或者 Flash 存储器来保存操作系统和应用程序,软件固化在存储器芯片或单片机中。

(2) 高质量、高可靠性: 嵌入式系统资源的稀缺性和执行的实时性要求软件代码应该尽可能地短小、高效、快速,这对编程以及编译器(Compiler)的设计提出了更高的要求。

(3) 高实时性: 在多任务嵌入式系统中,存在重要性不同的多个任务,如何对多任务请求合理的响应是系统设计的关键,这个环节处理不当很可能导致灾难性结果。保证任务请求的迅速执行不能单纯通过提高处理器速度来实现,必须通过优化系统软件,合理设计任务调度来解决。

1.1.4 嵌入式系统的发展

计算机技术在主机计算模式下发展了 40 年,从 20 世纪 80 年代起,进入了桌面计算模式时代,这种计算模式极大地推动了计算机技术和计算机产业的发展,普及了计算机知识,使计算机与各个领域完美结合,推动了国防、工业、文教等各方面的发展。Internet 的发明,使计算模式自然过渡到网络和分布式计算(Distribute Computing)时代,信息技术的发展使

海量资源共享成为可能,各种手持终端(Handheld Terminal)已经具备了桌面计算机的性能,计算机的应用越来越广泛,计算机的形态也越来越多样化,甚至在不知不觉中,人们已经享受到了计算机提供的多种服务,计算模式正在经历向普适计算(Ubiquitous Computing, UC)的转变。

普适计算又称为无处不在的计算(Pervasive Computing, PVC),这一思想最早由 Mark Weiser 提出。1991 年,Mark Weiser 在《Scientific American》杂志发表了《The Computer for the 21st Century》这一文章,标志着普适计算概念的正式确立。在文章中,Mark Weiser 提出应该把计算机嵌入到环境当中,并通过各种有线、无线网络连接起来互相通信。计算机不再显式地出现在视野范围内,人们因此可以把注意力集中于问题本身,在不知不觉中接受计算机的服务。他认为,计算机技术发展到高级阶段,必将充满人们生活的各个角落,并可以任意形态存在。人们可以随身携带多个计算机,甚至可以把计算机当作衣服“穿”在身上,计算机就像空气一样,每时每刻都在起作用,但却很少被有意识地注意到,计算机的形态“消失”了。“最深奥的技术是那些消失了的技术,这些技术将它们的自身交织于日常生活中,直至不可区分”。

普适计算的研究基础是各种嵌入式技术,研究内容包括智能环境和不可见计算、无缝的可移动性、普遍的信息访问、上下文感知、可穿戴计算等。普适计算也面临着一些挑战,例如计算模式的转变、计算资源的共享、移动计算的实现、应用程序的相互沟通等。

普适计算概念出现以来,迅速成为学术界的研究热点,这以两个重要学术会议的创立为标志,分别是 1999 年开始的 Ubicomp 国际会议和 2000 年开始的 Pervasive Computing 国际会议。2002 年,IEEE Pervasive Computing 期刊创刊,进一步提高了普适计算的研究热度。许多国家已经开展了各种普适计算研究项目,例如 MIT 的 Oxygen、CMU 的 Aura、Illinois 大学的 Gaia、UC Berkeley 的 Endeavour、Washington 大学的 Portalano、IBM 的 DreamSpace、Microsoft 的 EasyLiving、欧盟资助的 Disappearing Computer 以及国内清华大学的 SmartClassroom 等。

1.2 嵌入式系统开发流程

一般来说,嵌入式系统的开发流程包括以下几个步骤:确定系统需求,根据需求来确定系统应该具有的功能;进行概要设计,画出系统硬件框图,明确系统各硬件组成部分之间的关系,对系统运行的软件进行初步规划;详细设计系统硬件,细化硬件框图,设计出系统硬件原理图,设计原理图的同时要考虑是否有利于软件的编写;根据原理图绘制印刷电路板图(Printed Circuit Board, PCB),制板,调试硬件;移植操作系统,进行软件设计;测试。

嵌入式系统的典型特征是面向用户、面向产品、面向应用的,市场应用是嵌入式系统开发的导向和前提。一个嵌入式系统的设计取决于系统的需求。与普通软件设计一样,进行嵌入式系统设计的前提是尽可能明确需求,包括明确系统要实现的功能、系统的性能指标、系统的稳定性和可靠性要求、通信方式、人机界面等,必要的时候可以撰写一个系统使用说明,作为将来设计的初步依据。

概要设计部分要解决的问题主要为系统各模块选型,包括处理器选型、操作系统选型、各功能模块选型等,并绘制系统框图。选型的一个原则是在保证系统性能指标的前提下选

取规格最低的模块或器件,这样可以降低成本。一个常见的误区是盲目选取性能最好、精度最高、技术最新的模块和器件,这样固然可以达到系统性能要求,但是从经济角度考虑却是下下之策。选型时要做市场分析,参考以往产品的优缺点,同时要考虑市场因素,包括模块使用的广泛程度、成本、质量、口碑,万不可闭门造车,避免完成设计之后才发现某模块已经停产的不利情况。要规划好哪些模块可以直接购买,哪些需要自己设计。直接购买模块能够大幅缩短开发时间,同时质量相对更可靠,自己设计模块耗时较长,可靠性需要自行解决,但是可以降低成本。确定系统的软硬件界面也是一个重要的工作,即确定哪些功能由硬件提供,哪些功能由软件实现。由硬件实现的功能开发速度快,成本高,由软件实现的功能需要较长的设计时间,但是大批量生产时可以降低成本。概要设计部分应该完成较为详细的系统框图。概要设计要解决的问题还包括确定是否需要操作系统,需要哪种类型的操作系统等。

在详细设计阶段,需要实例化系统框图,明确系统每个模块具体使用的硬件器件,并连接好系统各硬件的连线。使用各种计算机辅助设计(Computer Aided Design, CAD)软件绘制系统原理图,原理图是使用电路元件符号绘制的电子元件连线走向图,它详细地描绘了各个元件的连线和走向,各个引脚的说明。绘制原理图时要对软件模块有一个初步的计划,硬件设计会直接影响到软件设计,二者要协同考虑,硬件设计的一个原则是尽量方便将来的软件设计。原理图设计完成之后要设计 PCB 图,PCB 图是电路板的映射图纸,它详细描绘了电路板的走线,元件的位置等,是电路板的基本结构图。设计 PCB 图的技术性比较强,有很多设计原则,根据电路的复杂程度可以把印刷电路板设计为双层板、多层板等,PCB 图设计的好坏直接影响到系统的成本以及运行的稳定性和可靠性。设计 PCB 图之后的下一个步骤是制板,然后把各个元件焊接到电路板上,元件包括通孔插装型和表面安装型,采用什么型号的元件是在设计 PCB 图时确定的。接下来要调试硬件,验证各部分硬件是否能够正常工作。一般先调试显示部分,如果显示部分工作正确,再利用显示部分的输出反馈信息来调试其他部分。详细设计阶段还包括设计软件流程图,如果选择使用操作系统,可以简化软件设计,如果不使用操作系统,那么所有软件部分都需要自行设计。对软件流程设计得越详细,将来编码就越容易。

如果设备需要嵌入式 Linux 操作系统的支持,那么开发过程还需要以下步骤。首先,在 PC 上安装 Linux 操作系统,不需要安装最新的版本,只要相关版本成熟、稳定就可以,推荐使用 Red Hat Linux 9,也可以使用 Linux 系统的模拟环境;然后需要在 Linux 系统中安装能够为嵌入式设备编译源程序的编译器,例如 arm-elf-gcc、arm-linux-gcc 等;PC 与嵌入式设备之间需要建立必要的通信连接,一般使用串口通信,所以 PC 上还需要安装 Linux 的 minicom 程序,或者直接使用 Windows 的超级终端;嵌入式操作系统能够正确引导的前提是有系统引导装载程序的支持,因此接下来要移植一个引导装载程序;然后裁减嵌入式 Linux 系统,配置根文件系统,期间可以借助于 BusyBox 等工具;最后,需要为设备上的特定硬件编写驱动程序。

下一步进入到编码阶段,如果开发环境提供了模拟器,可以先在 PC 上编写代码,调试通过后再下载到目标机。编码的复杂程度直接依赖于详细设计阶段的详尽程度。编码时要遵循一定的编程规范,例如标识符命名规则、代码缩进、必要的注释等。

测试只能证明缺陷存在,而不能证明缺陷不存在。对于一个复杂的系统而言,无论采取